



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

MAESTRÍA EN INGENIERÍA ELÉCTRICA



TESIS

DISEÑO Y CONSTRUCCIÓN DE UN MEDIDOR AMI PARA CARGAS TRIFASICAS
UTILIZANDO MICROCONTROLADOR XBee-ARDUINO

Que para obtener el Grado de:
Maestro en Ingeniería Eléctrica

Presenta:
Ing. Carlos Antonio Rangel Brambilla
G14071347
CVU:1106179

Director de tesis:
M.C. Rafael Castillo Gutiérrez

Codirector de tesis:
M.C. Aarón González Rodríguez

Diciembre 2023

Cd. Madero, Tamaulipas

Ciudad Madero, Tamaulipas, 27/noviembre/2023

Oficio No.: U.141/2023
Asunto: Autorización de impresión de tesis

C. CARLOS ANTONIO RANGEL BRAMBILLA
No. DE CONTROL G14071347

PRESENTE

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su Examen de Grado de Maestría en Ingeniería Eléctrica, se acordó autorizar la impresión de su tesis titulada:

"DISEÑO Y CONSTRUCCIÓN DE UN MEDIDOR AMI PARA CARGAS TRIFASICAS UTILIZANDO MICROCONTROLADOR XBee-ARDUINO"

El Jurado está integrado por los siguientes catedráticos:

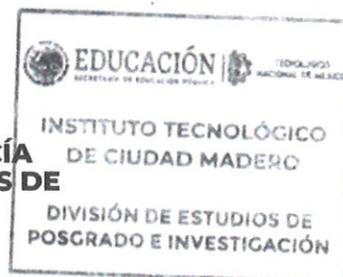
PRESIDENTE:	M.C.	RAFAEL CASTILLO GUTIERREZ
SECRETARIO:	M.C.	EDUARDO NACÚ SALAS CABRERA
VOCAL:	M.C.	PEDRO MARTÍN GARCÍA VITE
SUPLENTE:	M.C.	AARÓN GONZÁLEZ RODRÍGUEZ
DIRECTOR DE TESIS:	M.C.	RAFAEL CASTILLO GUTIERREZ
CO-DIRECTOR:	M.C.	EDUARDO NACÚ SALAS CABRERA

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

ATENTAMENTE

Excelencia en Educación Tecnológica
"Por mi patria y por mi bien"

MARCO ANTONIO CORONEL GARCÍA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



ccp. Archivo
MACG 'RCC'



INDICE

INDICE.....	2
LISTA DE FIGURAS.....	5
LISTA DE TABLAS.....	7
AGRADECIMIENTOS.....	8
DEDICATORIA	9
RESUMEN	10
ABSTRACT	11
CAPITULO 2	12
ANTECEDENTES DE MEDIDORES AMI	12
2.9 Introducción	13
2.10 Antecedentes del Problema	13
2.11 Justificación	14
2.12 Objetivos	15
2.12.1 Objetivos específicos.....	15
2.13 Hipótesis.....	16
2.14 Alcances y Limitaciones.....	16
2.14.1 Alcances.....	16
2.14.2 Limitaciones.....	16
CAPITULO 2	17
MARCO TEORICO.....	17
2.3 Infraestructura de medición Avanzada (AMI)	18
2.3.1 Características	18
2.3.2 Beneficios	20
2.3.3 Elementos clave	20
2.1 Microcontrolador Arduino	23
2.1.1 Tipos de microcontroladores	23
2.1.2 Características	23
2.1.3 Selección de microcontrolador Arduino para el desarrollo del proyecto	24
2.2 Zigbee	26
2.2.2 Estructura de una red ZigBee	27

INDICE

2.2.3	Aplicaciones de ZigBee	28
2.2.4	Características	28
2.3	XBee.....	29
2.3.1	Introducción	29
2.3.2	Definición	30
2.3.3	Infraestructura clave	30
2.4	Software XCTU.....	30
2.4.1	Primera aplicación. Chat Básico.	32
2.5	Parámetros eléctricos	35
2.5.1	Potencia Activa.....	36
2.5.2	Potencia Reactiva	36
2.5.3	Potencia Aparente.....	36
2.5.4	Factor de Potencia.....	36
2.5.5	Triangulo de potencias	37
2.6	Sensor para corriente de efecto Hall ACS712	38
2.7	Sensor para voltaje ZMPT101B	39
2.8	Relevador 5V	40
2.9	Contactador trifásico con bobina auxiliar	41
CAPITULO 3		42
DISEÑO E IMPLEMENTACION DEL PROTOTIPO		42
3.1	Medición de Voltaje trifásico	43
3.2	Medición de Corriente trifásica.....	44
3.3	Programación Arduino-IDE.....	45
3.4	Transmisión inalámbrica	49
3.5	Interfaz grafica	50
3.5.1	Instalación de Software Visual Studio 2019	50
3.5.1	USB-SERIAL COM	51
3.5.2	CONTROL DE SUMINISTRO ELÉCTRICO	52
CAPITULO 4		55
PRUEBAS Y RESULTADOS DEL PROTOTIPO.....		55
4.9	Modelado del suministro eléctrico trifásico	56
4.9.1	Motor de inducción.....	56
4.10	Pruebas de Voltaje con Arduino-IDE	57

INDICE

4.10.1	Pruebas físicas de voltaje monofásicas	58
4.10.2	Pruebas de voltaje trifásicas.....	60
4.11	Pruebas de Corriente con Arduino-IDE	61
4.11.1	Pruebas físicas de corriente monofásicas	62
4.11.2	Pruebas físicas de corriente trifásicas	63
4.12	Pruebas del relevador de 5V con Arduino-IDE.....	65
4.12.1	Pruebas de funcionamiento	65
4.13	Pruebas de relevador de 5V con contactor trifásico mediante Arduino-IDE.....	66
4.13.1	Pruebas de funcionamiento	66
4.14	Pruebas con radiofrecuencia utilizando Arduino-IDE/XBee.....	67
4.14.1	Pruebas físicas para lectura de voltaje y corriente trifásico mediante protocolo ZigBee	67
4.15	Pruebas de relevador de 5V con contactor trifásico mediante protocolo ZigBee utilizando Arduino-IDE/XBee	69
4.15.1	Pruebas de funcionamiento	69
CAPITULO 5	73
CONCLUSIONES Y TRABAJOS FUTUROS	73
5.1	Conclusiones.....	74
5.2	Recomendaciones para trabajos futuros.	74
CÓDIGO ARDUINO SENSORE EMISOR DE LAS VARIABLES	76
CÓDIGO ARDUINO SENSORE RECEPTOR DE LAS VARIABLES	99
CÓDIGO VISUAL PARA CREAR SOFTWARE	101
CÓDIGO VISUAL PARA LA DEFICNICION DE EVENTOS EN SOFTWARE	112
BIBLIOGRAFIA	115

LISTA DE FIGURAS

CAPITULO 2	12
ANTECEDENTES DE MEDIDORES AMI	12
CAPITULO 2	17
MARCO TEORICO.....	17
Fig. 2.1 Diagrama esquemático general de un sistema AMI.	19
Fig. 2.2 Placa Arduino 2560	24
Fig. 2.3 Logotipo de Arduino.....	25
Fig. 2.4 Aplicación en domótica de ZigBee	26
Fig. 2.17 Estructura de una Red ZigBee	27
Fig. 2.6 Dispositivo XBee.....	29
Fig. 2.7 XBee conectado a adaptador Shield.	30
Fig. 2.8 Interfaz gráfica XCTU.....	31
Fig. 2.9 Prueba uno. Chat básico entre dispositivos.....	32
Fig. 2.2 Comunicación XBee-XBee.	33
Fig. 2.11 Direccionamiento de módulos XBee.	33
Fig. 2.12 Red de multipuntos conectados a un coordinador.....	34
Fig. 2.13 Conexión Arduino-XBee	35
Fig. 2.14 Triángulo de Potencias y Factor de Potencia	37
Fig. 2.15 Diagrama de conexiones para conectar Arduino con sensor ASC712	38
Fig. 2.16 Diagrama de conexión para sensor ZMP101B	39
Fig. 2.17 Diagrama de conexiones para relevador de 5V	40
Fig. 2.18 Diagrama de conexiones para circuito de fuerza.....	41
CAPITULO 3	42
DISEÑO E IMPLEMENTACION DEL PROTOTIPO	42
Fig. 3.1 Conexión de tres sensores de voltaje ZMPT101B con Arduino MEGA	43
Fig. 3.3 Ejemplo declaración de variable de Voltaje 1.....	45
Fig. 3.4 Muestra tomada por segundo para corriente Fase 1	46
Fig. 3.5 Calculo de potencia real de Fase 2.....	46
Fig. 3.6 Ejemplo de lectura de potencia aparente de Fase 3 c.....	47
Fig. 3.7 Calculo de factor de potencia de la Fase 1.....	47

LISTA DE FIGURAS

Fig. 3.8	Señales de salidas analógicas de Arduino para enviarlas a interfaz grafica.....	48
Fig. 3.9	Diagrama de Conexiones del Módulo XBee-Arduino.....	49
Fig. 3.10	Visual Studio Community 2019.....	50
Fig. 3.11	Creación de nuevo proyecto con lenguaje C#.....	50
Fig. 3.12	Carpetas Creadas por Visual Studio 2019.....	51
Fig. 3.13	Todo Windows Forms o Cuadro de herramientas.....	51
Fig. 3.14	GroupBox 1 para establecer comunicación serial.....	52
Fig. 3.15	GroupBox 1 para establecer baudios por segundo a 9600.....	52
Fig. 3.16	GroupBox 2 para control de suministro para eventos de aplicación.....	53
Fig. 3.17	Interfaz Gráfica en Visual Estudio completa.....	54
CAPITULO 4	55
PRUEBAS Y RESULTADOS DEL PROTOTIPO	55
Fig. 4.1	Motor de inducción jaula de ardilla.....	56
Fig. 4.2	Distribución de las bobinas del motor.....	57
Fig. 4.3	medición de Voltaje monofásico.....	58
Fig. 4.4	Variador de voltaje al 70%.....	59
Fig. 4.5	Medición de voltaje de voltímetro Klein Tools.....	59
Fig. 4.6	Datos obtenidos por sensor ZMPT101B con variador de voltaje al 70%.....	60
Fig. 4.7	Variador de voltaje al 100%.....	60
Fig. 4.8	Medición de voltaje con variador al 100%.....	61
Fig. 4.9	Medición de voltaje Arduino-IDE con variador al 100%.....	61
Fig. 4.10	Pruebas de corriente con cargas resistivas.....	62
Fig. 4.11	Corriente medida por amperímetro Klein Tools.....	62
Fig. 4.12	Corriente medida por sensor ASC712.....	63
Fig. 4.13	Conexión para pruebas de corriente trifásicas.....	63
Fig. 4.14	Corriente tomada por amperímetro de efecto Hall.....	64
Fig. 4.15	Corrientes mostradas en Arduino-IDE.....	64
Fig. 4.16	Pruebas de funcionamiento de relevador de 5V.....	65
Fig. 4.17	Conexión de circuito de control con circuito de fuerza.....	67
Fig. 4.18	Conexiones para lectura de parámetros eléctricos Arduino/XBee.....	67
Fig. 4.19	Valores obtenidos por el dispositivo receptor de las variables.....	68
Fig. 4.20	Valores de corriente reales con multímetro Klein Tools.....	69
Fig. 4.22	Valores de Voltaje reales con multímetro Klein Tools.....	69

LISTA DE TABLAS

INDICE.....	2
LISTA DE FIGURAS.....	5
LISTA DE TABLAS.....	7
AGRADECIMIENTOS.....	8
DEDICATORIA	9
RESUMEN	10
ABSTRACT	11
CAPITULO 1	12
ANTECEDENTES DE MEDIDORES AMI	12
CAPITULO 2	17
MARCO TEORICO.....	17
CAPITULO 3	42
DISEÑO E IMPLEMENTACION DEL PROTOTIPO	42
CAPITULO 4	55
PRUEBAS Y RESULTADOS DEL PROTOTIPO.....	55

AGRADECIMIENTOS

Quiero agradecer a la División de Estudios de Posgrado e Investigación del ITCM, por otorgarme la oportunidad de cursar mi Maestría, agradecer al Consejo Nacional de Ciencia y Tecnología por su ininterrumpido apoyo durante estos 2 años, a todo el cuerpo docente de la Maestría en Ingeniería Eléctrica, por compartir sus conocimientos, apoyo y aliento para la culminación del programa.

A agradecer inmensamente a mi asesor y coordinador el M.C. Rafael Castillo Gutiérrez por su apoyo y enseñanzas en la realización de esta tesis y a su excelente equipo de profesores. Agradecer sus esfuerzos en el desarrollo del proyecto del medidor tipo AMI. Agradezco sus esfuerzos en ayudarnos a continuar mejorando este prototipo.

DEDICATORIA

Quiero dedicar este proyecto por la ayuda que recibí de todos los miembros de mi familia Rangel Brambilla por sus alientos en momentos difíciles y las personas que estuvieron cerca de mí en este camino y me brindaron todo su apoyo.

A mi madre Antonia Brambilla Reyes y a mi padre Juan Carlos Rangel por su apoyo incondicional, consejos y guías a través de toda mi vida escolar y personal.

RESUMEN

En los últimos años, se ha registrado un aumento notorio de consumo de energía eléctrica por parte de los usuarios en el área residencial e industrial, lo cual exige la modernización del sistema eléctrico nacional, permitiendo así una gestión eficiente de los recursos. Con el objetivo de tener una mejor administración del consumo de energía, la automatización de la red eléctrica, y además la utilización de tecnología automatizada es inminente.

Para la solución a este problema se propone el uso de medidores de infraestructura avanzada (AMI) son redes eléctricas inteligentes que combinan las tecnologías de la información y, también la comunicación remota con la automatización y el control. Esta tecnología esta aplicada sobre la generación, transmisión, distribución, y el consumo eléctrico. Los medidores de infraestructura avanzada son un sistema integrado capaz de medir redes eléctricas, comunicación y sistemas de gestión de datos que permiten la comunicación bidireccional entre el consumidor y los servicios de suministro.

Los parámetros del medidor se envían al centro de control de la empresa suministradora a través de una red de datos. El centro de control de los servicios puede utilizar los parámetros para mejorar la eficiencia operativa, y, asimismo, la sostenibilidad al controlar el uso de energía y la eficiencia del sistema de manera efectiva, detectar fallas y reconocer irregularidades. Es así que, su finalidad principal es asegurar un suministro eléctrico fiable y continuo. Además, ayudan a mejorar la calidad del servicio y a convertirlo en algo más seguro y sostenible.

Para la construcción de este prototipo se simulo una carga residencial trifásica mediante un motor trifásico del equipo de LabVolt variando el consumo eléctrico con un reóstato. Las antenas XBee son soluciones que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Estos módulos utilizan un protocolo llamado IEEE 802.15.4 para crear redes POINT-TO-MULTIPOINT (punto o multipunto); o para redes PEER-TO-PEER (punto a punto). Considerando que el Arduino que se utiliza conectado al motor trifásico es encargado de acondicionar y digitalizar las variables eléctricas simulando el consumo de un usuario del tipo residencial, el segundo Arduino recibe conectado a una laptop las variables monitoreadas por el primer Arduino como voltaje y corriente.

Una vez implementado este objetivo se podrá reducir los gastos para la empresa suministradora de electricidad, logrando reducir la mano de obra para la lectura del consumo eléctrico, evitando la lectura eléctrica medidor por medidor como actualmente se implementa el México.

ABSTRACT

In recent years, there has been a notable increase in electricity consumption by users in the residential and industrial area, which requires the improvement of the system, thus allowing efficient management of resources. In order to have a superior management of energy consumption, the automation of the electrical network and the use of automated technology is imminent. In recent years, there has been a notable increase in electricity consumption by users in the residential and industrial area, which requires the improvement of the system, thus allowing efficient management of resources. In order to have a superior management of energy consumption, the automation of the electrical network and the use of automated technology is imminent.

Advanced infrastructure meters are smart electricity networks that combine information and remote communication technologies with automation and control.

This technology is applied to generation, transmission, distribution, and electricity consumption. Currently, diver's corporations have installed devices, capable of measuring and storing information related to consumption per hour in order to know the demand for electrical energy throughout the day in more detail.

CAPITULO 2

ANTECEDENTES DE MEDIDORES AMI

Es de gran importancia mencionar, que para el completo entendimiento y la construcción del medidor de infraestructura avanzada es necesario tener conocimiento de los logros, necesidades y aplicaciones del proyecto “Diseño y construcción de un medidor tipo AMI con comunicación bidireccional utilizando microcontrolador Arduino” elaborado por el Ingeniero Darwin Sifuentes Hernández y los antecesores al presente proyecto.

2.9 Introducción

Los medidores de infraestructura avanzada (AMI) son un sistema inteligente integrado capaz de medir redes eléctricas, sistemas de comunicación de y sistemas de gestión de datos, que permiten la comunicación bidireccional entre el consumidor y los servicios de suministro.

Los medidores AMI es un dispositivo construido para la lectura de parámetros eléctricos enviados desde el medidor del usuario al centro de control de la empresa suministradora a través de una red de datos. El centro de control de los servicios puede utilizar los parámetros para mejorar la eficiencia operativa, y, asimismo, la sostenibilidad al controlar el uso de energía y la eficiencia del sistema de manera efectiva, detectar fallas y reconocer irregularidades en el sistema.

2.10 Antecedentes del Problema

Hoy en día el sistema de facturación por parte de la Comisión Federal de Electricidad (CFE) se puede mejorar con la incorporación de medidores AMI, esto pese a que, para la recopilación de consumo eléctrico de los usuarios, personal de CFE se ve en la necesidad de acudir a los domicilios personalmente para la facturación del servicio. Una de las situaciones que más se presentan actualmente es el corte del suministro eléctrico a los usuarios por falta de pago, y de igual manera, el personal de CFE se ve en la necesidad de acudir a los domicilios para realizar el corte de suministro eléctrico. Una vez que los usuarios realizan el pago correspondiente de su consumo eléctrico, personal de CFE se ve en la necesidad de acudir una vez más para hacer la conexión del suministro.

Este proceso afecta económicamente a la compañía. Los gastos de facturación para un servicio continuo se ven afectados y los procesos aplicados son tardados. Es así como, su finalidad principal es asegurar un suministro eléctrico fiable y continuo. Además, ayudar a mejorar la calidad del servicio y convertirlo en algo más que seguro y sostenible. Actualmente en México, no hay ningún tipo de Red Eléctrica inteligente (REI) aunque existe el interés por parte de CFE en modernizar sus sistemas de distribución, ya que en el año 2017 la Secretaría de Energía (SENER) publicó el programa de Red Inteligente en México con la finalidad de hacer al Sistema Eléctrico más eficiente. Para ello, se realizó una licitación para la instalación de 250 mil medidores AMI en casas-habitación, la cual fue ganada por la empresa SIEMENS en ese mismo año y planea que en los próximos 14 años se modernicen 5.7 millones de medidores con un costo estimado de 8mil 413 pesos, siendo este gasto exorbitante; el cual podría reducirse con la creación de un prototipo de manufactura mexicana que realice la misma acción y a un costo menor.

El estudio presentado en este artículo para la construcción de un medidor tipo AMI se simuló para una carga residencial trifásica mediante un motor trifásico del equipo de LabVolt realizando un estudio comparativo con diferentes tipos de limitadores para evaluar el comportamiento de la corriente, tensión aplicada, la respuesta de los relevadores y contactores.

El cálculo del consumo de corriente se realizó con la ayuda de un sensor de corriente “ACS712” basado en efecto Hall, se cuantificó el voltaje de operación con un sensor “ZMPT101B”. Con la ayuda de microcontroladores Arduino se convirtieron los datos analógicos a salidas digitales interpretados en el monitor Serie de la interfaz gráfica “Arduino IDE”. La comunicación bidireccional se implementó con equipos de radiocomunicación inalámbrica “XBee” basados en un protocolo de comunicación IEE 802.15.4 para la creación de una red de punto a punto. Una vez implementado este objetivo se podrán reducir los gastos de facturación del servicio eléctrico por parte de CFE, logrando reducir la mano de obra para la lectura del consumo eléctrico, evitando la lectura eléctrica medidor por medidor y aplicando la conexión o desconexión remota del suministro eléctrico.

2.11 Justificación

Hoy en día, las computadoras están presentes en todas las áreas de la actividad humana; en el hogar, en la oficina, en los bancos, en las escuelas, en la industria etc.

Aunque en algunos casos las computadoras realizan en forma aislada la función para la cual fueron diseñadas, en otros es necesario que haya intercambio de información con otros dispositivos. Eso significa que para el diseño de la mayor parte del equipo de comunicación que se instalan en la actualidad es necesario tomar en cuenta la clase de transmisión de datos y recepción de estos que permita comunicarse con otras computadoras o dispositivos.

La importancia de tener una buena base de datos en todos los aspectos tanto en los medios de transmisión como los diferentes protocolos de comunicación para así poder recabar el óptimo desempeño de ésta, siempre buscando que nuestra transmisión y recepción de datos sea lo mejor posible.

Los beneficios que se quieren lograr con la creación de este medidor principalmente es la reducción de los costos de medición y facturación, ahorro de mano de obra, facturación más precisa y oportuna, menos enfrentamientos con los clientes, mejoras en eficiencia operacional, más control sobre los consumos eléctricos de cada usuario ya sea residencial, comercial o industrial, reducir los gastos dirigidos hacia el servicio público.

Es importante recalcar que para lograr todos estos beneficios con el medidor tipo AMI se realizará con dispositivos y componentes de bajo costo para que esté al alcance de cualquier tipo de usuario, todo esto sin disminuir la calidad y la eficiencia del medidor.

2.12 Objetivos

El objetivo principal de esta aplicación es recabar y ordenar información de los parámetros eléctricos con el tema para contribuir en una mejor técnica para la distribución eléctrica mediante transmisión de datos.

Para la construcción de un medidor de medición avanzada debemos tener como visión la aplicación del AMI para buscar el registro y control del suministro de energía eléctrica de manera remota y la visualización en tiempo real del consumo del usuario para tener un control y acceso al servicio de los usuarios de manera remota.

2.12.1 *Objetivos específicos*

Diseñar y construir un medidor de medición avanzada (AMI) que realice lecturas eléctricas en tiempo real que entregue al elemento sensor primario y la convierta en una señal digital adecuada para que el dispositivo realice la presentación del dato a medir y sea capaz de realizar una comunicación bidireccional entre el punto de control y el dispositivo en campo.

La transmisión de las señales y las variables eléctricas serán transmitidas por señales de ondas de radio frecuencias.

- [1] Implementar una red de comunicación inalámbrica.
- [2] Acceso y control de parámetros eléctricos de manera remota.
- [3] Facturación del consumo de usuarios residenciales, comerciales e industriales.
- [4] Conocer variables eléctricas en forma digital mediante un elemento conversor de la variable.
- [5] Evaluación precisa de las variables eléctricas en tiempo real.
- [6] Mostrar un análisis de la red eléctrica suministrada.
- [7] Utilizarse en cualquier sector que requiera de energía para su operación.
- [8] Descubrir y evitar irregularidades o excesos de consumo.
- [9] Análisis de curvas de carga para localizar los puntos de máxima demanda energética.
- [10] Prevención de riesgos en la red eléctrica.
- [11] Identificación de problemas en la red.
- [12] Apertura y cierre del suministro de la energía eléctrica mediante la medición de la salida de impulsos de parámetros eléctricos.

- [13] Se trata de un medidor perfecto para el análisis, en especial la diferencia de tensión entre los inicios y finales de consumo.
- [14] Como características a destacar, la sencillez en su instalación y la facilidad de uso del software.

2.13 Hipótesis

El medidor de infraestructura de medición avanzada (AMI) será capaz de tener una comunicación bidireccional entre el usuario y el centro de control, con la finalidad de tener un suministro eléctrico controlado de manera exacta y en tiempo real, monitoreando toda la red eléctrica del usuario mostrando las mediciones de voltaje, corriente, potencia activa, potencia reactiva, potencia aparente, factor de potencia, triángulo de potencias y kilowatt-Hora.

2.14 Alcances y Limitaciones

Con la creación del presente medidor, se debe tomar en cuenta las características requeridas para lograr el objetivo final del mismo, para así evitar un resultado deficiente o equivocado. De igual manera, se debe tomar en cuenta los aspectos que no forman parte de la cobertura de el objetivo final.

2.14.1 *Alcances*

El medidor será apto para realizar las mediciones de voltaje y de corriente, de acondicionar estas variables para el cálculo del consumo que será mostrado en tiempo real al usuario, a su vez transmitirá esta información de manera remota a un punto de control donde podrán visualizarlo y controlar el suministro de energía eléctrica.

2.14.2 *Limitaciones*

El prototipo será desarrollado y modelado para un consumo de energía trifásica mediante un motor de inducción, para realizar una medición trifásica de 380 Volts. Por otra parte, la capacidad máxima de medición de los sensores de corriente es hasta 30A, los sensores de Voltaje tendrán la capacidad de medir un Voltaje por línea de hasta 180V y 280 entre líneas. La transmisión remota tendrá un radio de visión de 100 metros en espacio de interior o urbanos y hasta 1000 metros en espacios rurales. Este primer prototipo no tendrá la capacidad de almacenamiento de datos eléctricos, pero la posibilidad de agregar una memoria externa es posible para futuras mejoras.

CAPITULO 2

MARCO TEORICO

Para el desarrollo de este proyecto, se requiere tener el conocimiento principal de los componentes y elementos con los cuales se construirá el medidor tipo AMI. Así también conocer los parámetros eléctricos, los cuales se estarán monitoreando con el medidor tipo AMI.

2.3 Infraestructura de medición Avanzada (AMI)

La infraestructura AMI (por sus siglas en inglés) integra hardware como medidores avanzados, centros de gestión de medida, enrutadores, concentradores y antenas. Además de software, arquitecturas y redes de comunicaciones, que permiten la operación de la infraestructura y la gestión de los datos del sistema de distribución de energía eléctrica y de los sistemas de medida del consumo de energía eléctrica. La medición es un proceso clave de los sistemas de energía eléctrica que permite, a las empresas prestadoras del servicio, cuantificar la cantidad de energía que se genera, transmite, distribuye y se factura.

Se conoce como sistemas avanzados de medición o sistemas con infraestructura de medición avanzada AMI, a los sistemas con capacidad de medir, registrar, recolectar y transferir remotamente, la información asociada al consumo, la demanda, los parámetros eléctricos y la forma de uso de la energía eléctrica, para su posterior presentación, análisis, gestión y toma de decisiones. Es decir, es una infraestructura moderna para la medición de los consumos de energía eléctrica, que aprovecha los sistemas de comunicaciones y avances tecnológicos.

2.3.1 Características

Un sistema AMI en general se compone de tres componentes principales: medidores inteligentes, redes de comunicaciones y el sistema de gestión de datos de medición (MDMS por sus siglas en inglés).

Uno de los elementos clave de estos sistemas es el medidor inteligente, que tiene integradas capacidades avanzadas de medición, registro de datos, análisis de uso de los servicios y comunicación bidireccional, para transferir remotamente la información a sistemas de procesamiento de datos, para fines de monitoreo remoto y facturación.

Otro elemento importante de los sistemas AMI son los concentradores o colectores de datos que realizan la recolección de datos de los medidores inteligentes, para transferirlos a través de redes domésticas (HAN), redes de medidores (NAN), redes de campo (FAN) y redes de área amplia (WAN), utilizando distintas tecnologías de comunicaciones (espectro disperso, radiofrecuencia, microondas, WiMax, Ethernet, Wi-Fi, ZigBee, celular, líneas eléctricas de potencia, fibras ópticas, etc. hasta alcanzar los sistemas de explotación y gestión de datos (MDM) en las oficinas de la empresa suministradora del servicio eléctrico de distribución, en donde se procesa la información para propósitos de facturación y el monitoreo de la calidad del servicio. La tecnología AMI es un componente clave para la reducción de pérdidas en las redes eléctricas inteligentes, originado del requerimiento del intercambio oportuno de información en tiempo real, del consumo de energía y de la demanda entre las empresas de electricidad y los consumidores.

El avance alcanzado por la tecnología de circuitos electrónicos especializados ha sido un factor fundamental para el desarrollo de las comunicaciones y de la medición inteligente que ha permitido desarrollar la tecnología AMI, la cual integra especialidades de medición, telecomunicaciones, informática y control.

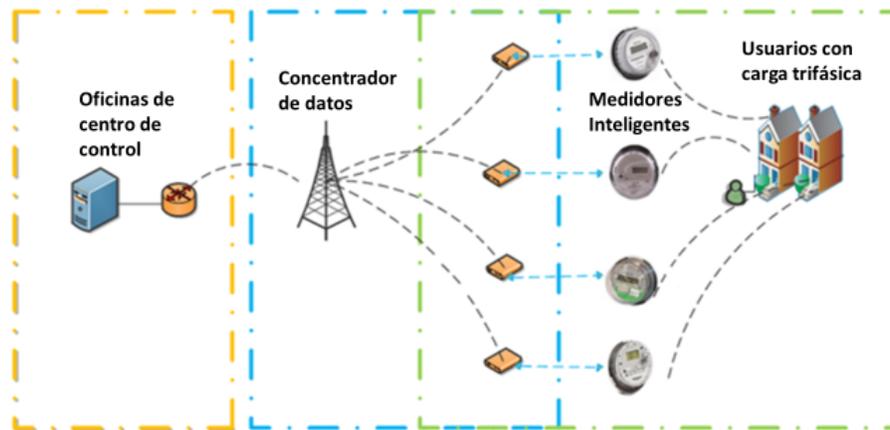


Fig. 2.1 Diagrama esquemático general de un sistema AMI.

Con estos sistemas se facilita la comunicación automatizada de dos vías entre dispositivos de medición inteligente y los procesos de las compañías de electricidad, para que éstas obtengan información de consumo, de demanda, de la forma y momentos de uso de la energía por parte de los consumidores, quienes obtienen información de gran utilidad para tomar decisiones respecto a la forma, períodos de uso y control de la energía que consumen. La tecnología AMI permite realizar mediciones remotas, detección de fallas, detección de pérdidas de energía, control de carga, generación distribuida, incorporación de vehículos híbridos eléctricos, entre otras, lo cual se traduce en diversos beneficios tanto para las empresas de electricidad, como para los usuarios del servicio eléctrico.

Esta tecnología también permite la generación de información necesaria para que los usuarios puedan modificar sus hábitos de consumo y a la empresa de electricidad le permite detectar fallas y pérdidas, lo que puede redundar en un mejor uso de la energía eléctrica, teniendo como consecuencia beneficios económicos y ambientales, tanto para la empresa como para el consumidor.

Las principales funciones de un sistema AMI son:

- La medición, recolección y análisis de datos de consumo de energía eléctrica.
- Manejar comandos y datos desde la oficina central hacia los dispositivos.
- Contribuye a proteger los ingresos de las empresas y hace más fácil y rápido su trabajo.

- Ayuda a incrementar la calidad de sus servicios, lo que mejora la calidad de vida de los usuarios.

Un sistema AMI es una poderosa herramienta para el monitoreo y diagnóstico de la red de distribución. El sistema permite la localización del punto de pérdida y la detección de eventos de manipulación sospechosos.

2.3.2 *Beneficios*

Los proyectos en los Estados Unidos demostraron que los sistemas tipo AMI pueden lograr importantes impactos y beneficios en la red para los clientes y las empresas de servicios públicos, entre ellos:

- Reducción de los costos de medición y facturación de menos camiones, ahorro de mano de obra, facturación más precisa y oportuna, menos disputas de clientes y mejoras en eficiencia operacional.
- Más control del cliente sobre el consumo de electricidad, los costos y las facturas gracias al mayor uso de las nuevas herramientas del cliente (por ejemplo, portales web y termostatos inteligentes) y técnicas (por ejemplo, cambio de la demanda a períodos de poca actividad).
- Reducir los gastos de capital de los servicios públicos y los ahorros en las facturas de los clientes como resultado de la reducción de la demanda máxima y las mejoras en la utilización y el mantenimiento de los activos.
- Reducir los costos de cortes y menos inconvenientes para los clientes debido a una restauración más rápida y un envío más preciso de los equipos de reparación a los lugares donde se necesitan.

AMI aplica tecnologías de comunicación y control inteligentes para automatizar las funciones de medición que se han realizado típicamente mediante operaciones intensivas manualmente, que incluyen lecturas del medidor de electricidad, conexión y desconexión del servicio, detección de robo y robo, identificación de fallas y cortes, y monitoreo de voltaje. Combinado con tecnologías avanzadas basadas en clientes, AMI también permite a las empresas de servicios públicos ofrecer nuevas opciones de tarifas que incentiven a los clientes a reducir la demanda máxima y el consumo de energía.

2.3.3 *Elementos clave*

Los parámetros eléctricos son de gran importancia en el desarrollo de las ciencias y de la humanidad, donde se involucran, voltajes, resistencias e intensidades de corriente eléctrica, que se encuentran presentes en todos aquellos dispositivos eléctricos de los cuales se comparten en la vida diaria, como lo son los celulares, computadoras, licuadoras, televisores, el consumo de la electricidad ya es algo común.

2.3.3.1 Sistema de infraestructura Avanzada de Medición (AMI)

Se considera un sistema de medición integrado por los siguientes componentes:

- a) Centro de gestión AMI.
- b) Medios de comunicación.
- c) Módulo o medios de recolección de datos y control.
- d) Gabinete de medidores.
- e) Medidor.
- f) Instrumento indicador de consumo.

a) Centro de gestión AMI

Se considera como centro de gestión de datos el hardware y el software que debe contener el sistema informático de gestión encargado de administrar la operación del sistema y de encuentra generalmente instalado en el centro de control por parte de la empresa suministradora de energía.

b) Medios de comunicación

La transmisión de los datos puede ser por radiofrecuencia, alámbrica o bien por fibra óptica.

c) Módulo o medios de Recolección de datos y control

Debe ser capaz de mantener una función de interfaz entre los medidores y el centro de control, puede llegar a estar integrada por más de un elemento.

d) Gabinetes o medidores

Encargado de mantener una comunicación con los indicadores de consumo y contiene módulos de adquisición y transmisión de datos hacia el gabinete concentrador de datos y control.

e) Medidor

Dispositivo capaz de informar al usuario y a la compañía suministradora del consumo en tiempo real.

f) Instrumento indicador de consumo.

Medio que presenta en forma visual el registro de los parámetros eléctricos e información al usuario.

2.3.3.2 Funciones del sistema AMI

El proceso de medición es clave en la reducción de pérdidas de energía eléctrica, por ello la tecnología AMI tiene un papel importante, ya que gracias a su oportuno intercambio de comunicación en tiempo real del consumo permite monitorear el consumo y lograr una comunicación entre los dispositivos de medición y las compañías proveedoras de electricidad.

La infraestructura de medición avanzada (AMI) permite una mejor eficiencia en el proceso de distribución de la empresa eléctrica. AMI incluye el hardware, software las comunicaciones, los sistemas asociados con la distribución de energía, los sistemas asociados a la red de consumidores y el software de gestión de datos de medidores inteligentes.

Con la infraestructura de medición (AMI) tanto como la empresa prestadora de servicio eléctrico y los usuarios, cuantifican la cantidad de energía que se genera, transmite, distribuye y se factura. Uno de los elementos importantes de los sistemas AMI son los concentradores o colectores de datos que realizan la recolección de datos de los medidores inteligentes, para transferirlos a través de redes domésticas (HAN), redes de medidores (NAN), redes de campo (FAN) y redes de área amplia (WAN). Utilizando distintas tecnologías de comunicaciones (espectro disperso, radiofrecuencia, microondas, WiMax, Ethernet, Wi-Fi, ZigBee, celular, líneas eléctricas de potencia, fibras ópticas, etc.).

Hasta alcanzar los sistemas de explotación y gestión de datos (MDM) en las oficinas de la empresa suministradora del servicio eléctrico de distribución, en donde se procesa la información para propósitos de facturación y el monitoreo de la calidad del servicio. La infraestructura de medición avanzada (AMI) es avance tecnológico que puede medir, registrar, recolectar y transferir la información del consumo, demanda, parámetros eléctricos y la forma de uso de la energía eléctrica. Están relacionado con dispositivos como los medidores inteligentes de electricidad, que poseen la capacidad de gestionar toda la información recolectada y a través de ella tomar decisiones.

AMI se compone de tres elementos principales:

- Medidores inteligentes.
- Redes de comunicaciones.
- Sistema de gestión de datos de medición.

Los avances tecnológicos AMI son componente clave para la reducción de pérdidas en las redes eléctricas de los usuarios, originado del requerimiento del intercambio inmediato de información en tiempo real, del consumo de energía y de la demanda entre las empresas de electricidad y los consumidores. El avance alcanzado por la tecnología de circuitos electrónicos especializados ha sido un factor fundamental para el desarrollo de las comunicaciones y de la medición inteligente, permite un desarrollo en la tecnología AMI, la cual integra especialidades de medición, telecomunicaciones, informática y control.

Estos sistemas hacen accesible la comunicación automatizada de dos vías entre dispositivos de medición inteligente y los procesos de las compañías de electricidad, para que estas adjudiquen información de consumo, de demanda, de la forma y momentos de uso de la energía por parte de los consumidores, quienes obtienen informe de gran utilidad para tomar decisiones respecto a la forma, períodos de uso y control de la energía que absorben del sistema de eléctrico.

2.1 Microcontrolador Arduino

Arduino se desarrolló para ser una plataforma libre soportada en una placa electrónica de hardware libre con diferentes pines hembra. Una placa electrónica mejor conocida como PCB (“Printed Circuit Board”, “Placa de Circuito Impreso” en español). Basado en la línea de microcontroladores Atmel AVR de 8 bits, la configuración de sus pines análogos, digitales y otros, se convirtió en un peculiar footprint que ha llegado a ser un estándar de la industria. El verdadero poderío de Arduino no está realmente en el hardware, sino en el software el IDE de Arduino. Sin duda Arduino es la tarjeta de desarrollo microcontrolador más utilizada en el mundo.

2.1.1 *Tipos de microcontroladores*

La mayoría de los microcontroladores inteligentes se pueden conectar a Internet, también son conocidos como MCU. Se puede creer que los microcontroladores son computadoras diminutas que se agregan a cualquier objeto físico o espacio para darle un “Cerebro”. Contiene uno o más procesadores de computadora, junto con memoria y periféricos de entrada / salidas programables, todo en un solo circuito integrado. Los MCU a comparación de una computadora estándar tienen menor capacidad de procesar datos, su bajo costo las convierte en una opción más práctica para agregar capacidades de computación a un objeto o dispositivo que no las tiene. Las MCU comúnmente van junto con lo que se conoce como una “placa de desarrollo”. Una placa de desarrollo proporciona todo lo necesario para programar la MCU. Las placas de desarrollo deben incluir cosas como fuente de alimentación, soporte para conectar sensores y, a veces, incluso sensores y actuadores a bordo. También permite a los usuarios conectar rápidamente y actuadores y el Software que los acompaña facilita la creación y despliegue del código.

2.1.2 *Características*

- **PRECIO ACCESIBLE:** Sin duda, su bajo costo es una de las principales particularidades de Arduino. Si se compara con otras plataformas microcontroladoras, se verá que las placas Arduino son relativamente baratas, además, existen módulos que pueden ensamblarse en forma manual o en versiones reducidas, lo que disminuye su precio en el mercado.

- **MULTIPLATAFORMA:** El paquete de software que acompaña a Arduino puede ser ejecutado en los sistemas operativos Windows, Mac OSX y GNU/Linux. En comparación, otros sistemas están limitados a un determinado sistema operativo.
- **ENTORNO DE PROGRAMACIÓN:** El entorno en el cual se puede programar a Arduino es fácil de usar y de aprender. Pero también es flexible, por lo que puede ser utilizado por usuarios avanzados o educadores.
- **SOFTWARE ABIERTO:** El software de Arduino está publicado como código abierto, por eso puede ser modificado y extendido por quien lo desee. Esto se realiza mediante el uso de librerías C++.
- **HARDWARE ABIERTO:** Arduino se basa en microcontroladores de Atmel; sus planos están disponibles bajo la licencia Creative Commons, por lo tanto, es posible crear versiones propias de los módulos, para extenderlos y mejorarlos. Además, se puede fabricar sus propias versiones de las placas, ya sea para entender su funcionamiento, para abaratar costos o para distribuirla.

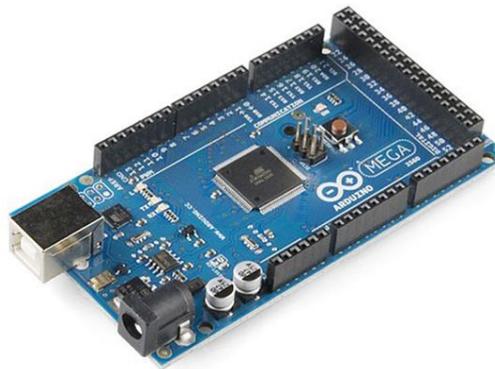


Fig. 2.2 Placa Arduino 2560

2.1.3 Selección de microcontrolador Arduino para el desarrollo del proyecto

Arduino fue creado con el propósito de ser una plataforma extremadamente fácil de usar en comparación con otras, lo que la hace ideal tanto para los desarrolladores más experimentados como para principiantes ya que ahora sus proyectos se pueden realizar mucho más rápido y son menos laboriosos.

Otro factor que hace que Arduino sea muy atractivo es su filosofía de hardware libre, es decir, la gente puede utilizarlo para crear varios proyectos sin coste alguno por los derechos de utilización de la plataforma y se puede distribuir de forma gratuita, si así lo desean.

Esto trae muchos beneficios; además de crear y distribuir varias bibliotecas nuevas y herramientas para ayudar al desarrollo de los proyectos todos los días, cuenta con una comunidad con miles de personas que revelan información y detalles acerca de lo que se crea y aportan documentación y tutoriales sobre el funcionamiento de Arduino.

Estas son también algunas de las razones por las que la popularidad de Arduino está creciendo entre los desarrolladores. Arduino es una herramienta de la electrónica que permite desde su diseño, montaje y uso con medios periféricos hacer prácticas pedagógicas en el área de mecatrónica, electrónica y eléctrica.

Una tarjeta de desarrollo es un circuito que contiene un microcontrolador o dispositivo lógico y elementos como puertos, conectores y reguladores que les permita a los usuarios acceder fácil y rápidamente a los periféricos para realizar aplicaciones y pruebas. Los elementos primordiales para el funcionamiento de la tarjeta de microcontrolador, puerto JTAG, conectores como USB, memoria MicroSD y un regulador.

El protocolo JTAG o “Joint Test Action Group”, nombrado por la norma IEEE 1149.1 utilizada para testear circuitos lógicos, probar los submódulos de circuitos integrados y como mecanismo para depuración de aplicaciones embebidas.

Los reguladores de voltaje se encargan de reducir la tensión de alimentación hasta los valores adecuados para los componentes, para una tarjeta de desarrollo pueden ser de voltaje de 5V (estándar TTL), 3.3V que es con el que operan los Arduino.



Fig. 2.3 Logotipo de Arduino.

Los conectores más empleados para las tarjetas de desarrollo son: USB significa “Universal Serial Bus”, permite transmitir datos a velocidades de 12 Mbps para conectar periféricos, para las tarjetas de desarrollo se usa el conector Mini-USB en modo Device y USB-A en Host. Conector para memoria MicroSD:

La mayoría de las placas de desarrollo tiene interfaces SPI, haciendo posible que soporten tarjetas de memoria SD, MMC o MicroSD, estas últimas son las más comunes porque son más sencillas de utilizar, ocupan menos espacio y son menos costosas, por lo tanto, para facilitar la comunicación entre el Arduino y la tarjeta se implementa un conector.

Arduino es una gama de circuitos electrónicos open source, basados la mayor parte en un microcontrolador del fabricante Atmel. Estos circuitos integran los componentes necesarios para permitir un uso rápido y sencillo del microcontrolador.

Esta simplificación está orientada a hacer accesible a todos la creación y la programación de objetos o dispositivos interactivos. Estos objetos pueden contener todo tipo de captadores, indicadores luminosos o interruptores que queramos.

Entre otros, las tarjetas Arduino están equipadas con conectores estandarizados para conectar módulos compatibles, llamados shields. Estos últimos son circuitos de un tamaño más o menos parecido al de Arduino y que se apilan sobre estos conectores. Ofrecen extensiones de hardware que permiten añadir funcionalidades originales a su proyecto.

2.2 Zigbee

ZigBee es una alianza para la definición de un protocolo de comunicación inalámbrico de alto nivel. Este protocolo está basado en el estándar 802.15.4 para las capas bajas de comunicación y tienen como principales rasgos diferenciales las características de bajo consumo, topología de red tipo mesh y tamaño reducido.



Fig. 2.4 Aplicación en domótica de ZigBee

Para usos industriales, científicos y médicos ZigBee utiliza la banda ISM. En Europa utiliza 868 MHz, en Estados Unidos 915 MHz y 2.4GHz en el resto del mundo. Gracias a estas características ZigBee se ha posicionado como una de las mejores soluciones inalámbricas para equipos de control desplegados en campo como equipos de metering o sensorial, ya que tienen la capacidad de operar mediante baterías.

El protocolo ZigBee también gestiona de forma automática la red, creando y manteniendo las rutas de forma dinámica. Esto permite que la red corrija automáticamente

situaciones de desconexión parcial de algunos de los nodos que la forman, evitando así que tener que acceder físicamente a los equipos para poder solventar este tipo de problemas.

ZigBee es comúnmente utilizado en las aplicaciones de domótica, esto debido a su tamaño reducido y costo, lo permite a ZigBee ser integrado en consumibles y equipos para el hogar, como bombillas, electrodomésticos, alarmas y otros elementos comunes en una casa domótica.

2.2.2 Estructura de una red ZigBee

Las redes ZigBee son accesibles para trabajar con una topología de red tipo mesh (o malla). Esto permite topológicamente a los nodos actualizar de forma dinámica la tabla de rutas, lo que aporta en gran manera la eficiencia de la red. Un nodo dentro de una red ZigBee puede tener los siguientes roles:

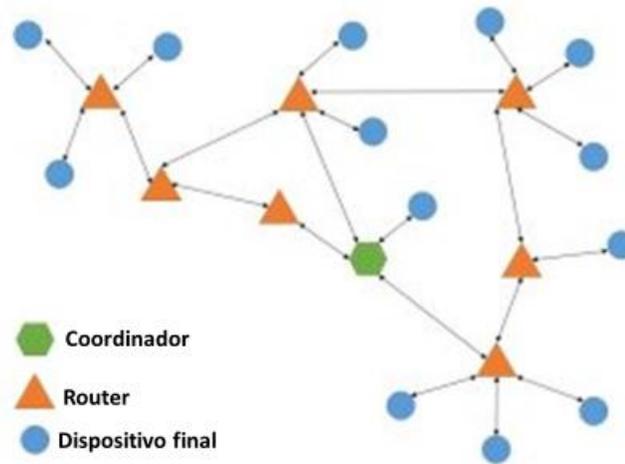


Fig. 2.17 Estructura de una Red ZigBee

- **Coordinador:** El coordinador de la red es el encargado de crear la red, enrutar y permitir a los paquetes las conexiones entrantes del resto de los nodos dentro de la red. Este nodo acostumbra a estar siempre conectado e integrado con un módulo de comunicación con conexión a internet, lo que a su vez permite enviar los datos recolectados por la red ZigBee a un servidor central.
- **Router:** De una manera parecida al trabajo del coordinador donde los nodos actúan como router, la red tiene la capacidad de enrutar paquetes, pero no pueden aceptar conexiones.
- **End Device:** Son los nodos más sencillos que representan la red. Capaces de recibir y enviar paquetes de datos dentro de la red, pero no cuentan con la capacidad de enrutar. Capaces de entrar en modo de bajo consumo, o sleep, para alargar la vida de la batería, conectando solo cuando sea necesario.

2.2.3 Aplicaciones de ZigBee

A continuación, se mencionan algunas de las aplicaciones más famosas:

- Comunicación inalámbrica
- Robot controlado inalámbricamente
- Sistema de monitoreo remoto
- Sistema inalámbrico de automatización para casas
- Sensores de temperatura inalámbrica y otros.

Los dispositivos ZigBee necesitan ser conectados con algún dispositivo inteligente como microcontroladores, Arduino y una computadora. Estos dispositivos darán la orden de que hacer o que no hacer a través de un programa ya precargado dentro de los microcontroladores Arduino.

2.2.4 Características

Las comunicaciones ZigBee se realizan en la banda libre de 2.4GHz. En comparación los dispositivos Bluetooth que no utilizan FHSS (Frequency hopping). Los XBee realizan la comunicación a través de una única frecuencia, es decir, de un canal. Se puede escoger un canal de 16 canales posibles.

El enlace depende de la potencia de emisión del dispositivo, así como las antenas utilizadas. Para la velocidad de transmisión de una red de datos ZigBee es de hasta 256kbps. En teoría las redes ZigBee pueden estar formadas hasta por 65535 equipos, es decir, el protocolo está preparado para tener la capacidad de controlar en la misma red de datos esta cantidad de dispositivos. Aunque en la aplicación es menor, siendo aun así capaz de controlar miles de equipos.

El estándar ZigBee tiene la capacidad de permitir la interoperabilidad entre diferentes dispositivos de diferentes fabricantes. Nos permite controlar luces, sistemas de climatización, riego, automático, robótico entre otros de diferentes marcas usando un mismo protocolo con productos certificados ZigBee.

Para lograr una interoperabilidad entre dispositivos, se debe tener por capas, separando los componentes y las funciones en módulos independientes.

Estas capas deben ser PHY y MAC que son las especificadas por el estándar IEEE 802.15.4 y las capas NWK que están especificadas por el estándar ZigBee que maneja la estructura de red, enrutamiento y seguridad.

La capa de aplicación es la que habilita la interoperabilidad y está compuesta por:

- **APS:** ofrece un interfaz entre la red y la capa de aplicación, define los mensajes estandarizados que permite la comunicación entre dispositivos de diferentes fabricantes.

- **Application Framework:** es el entorno donde se almacenan las aplicaciones en los dispositivos ZigBee.
- **ZDO (ZigBee Device Object):** ofrece funcionalidades de descubrimiento de dispositivos y gestión avanzada de red.

2.3 XBee

XBee es el nombre comercial del Digi de una gran familia de módulos son pequeñas radios capaz de comunicar información de forma inalámbrica unas con otras basados en el estándar ZigBee, pero Digi tiene muchos XBee Capaces de hacer cosas simples como reemplazar cables en una comunicación serial, lo cual es de gran utilidad cuando se desea crear, por ejemplo, una red de infraestructura avanzada.

2.3.1 Introducción

Los dispositivos XBee cuentan con entradas, salidas digitales y analógicas que son capaces de controlar e incluso se pueden conectar sensores y leer sus mediciones en forma remota. Una de las ventajas más importantes de estos dispositivos es su bajo consumo, además de su posibilidad de realizar conexiones de punto a punto y punto a multipunto.

Existen muchos tipos diferentes de módulos de comunicación, los cuales son revisador previamente antes de su aplicación.

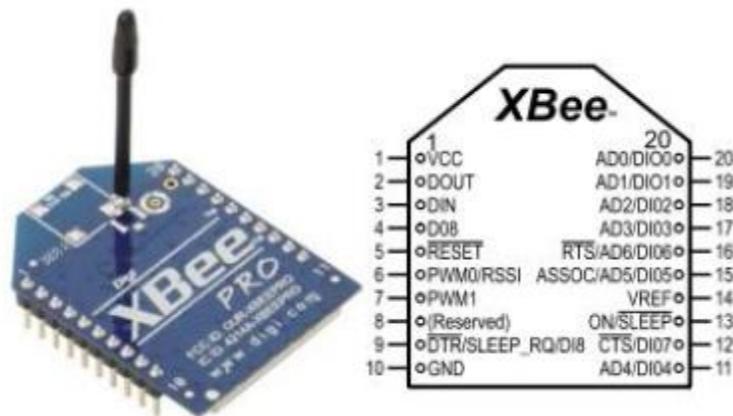


Fig. 2.6 Dispositivo XBee

Una de las ventajas de los dispositivos XBee es que todos, independientemente del modelo o serie, tienen los pines en la misma posición. Alimentación, tierra y los pines de comunicación TX Y RX se encuentran en la misma posición, haciendo que los dispositivos sean totalmente intercambiables, para la mayoría de las aplicaciones. Las características avanzadas más importantes en estos dispositivos son compatibles.

2.3.2 Definición

Los XBee de acuerdo con Digi, son soluciones integradas capaces de brindar un medio inalámbrico para la comunicación e interconexión entre dispositivos. Estos módulos utilizan un protocolo de comunicación llamado IEEE 802.15.4 para crear redes de POINT-TO-MULTIPOINT (punto a multipunto); también para redes PEER-TO PEER (punto a punto). Son diseñados para las aplicaciones que requieran un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible. Básicamente los XBee son una implementación de Digi basada en un protocolo Zigbee. En pocas palabras los XBee son módulos inalámbricos fáciles de usar.

2.3.3 Infraestructura clave

Para usar o configurar los dispositivos XBee con ordenador es necesario un adaptador o mejor conocido como Shield, este dispositivo cuenta con un chip FTDI que tiene como función interconectar el puerto serie y el USB con el software XCTU.



Fig. 2.7 XBee conectado a adaptador Shield.

2.4 Software XCTU

Para la configuración y uso de los módulos XBee es necesario descargar e instalar XBee Configuration and Test Utility mejor conocido como XCTU. Es un software multiplataforma que permite interactuar con los módulos mediante una interfaz gráfica. Esto con el fin de incluir una herramienta que haga muy sencilla la configuración y las pruebas de los módulos XBee.

Esta es la base sobre la que se define la especificación ZigBee, cuyo propósito es ofrecer una solución completa para este tipo de redes construyendo los niveles superiores de protocolos estándar.

XCTU nos sirve para configurar, inicializar, actualizar el firmware y testear los módulos XBee, comunicándose por puerto serie a los módulos. Este software nos permite ver de una manera rápida un resumen de los parámetros a los cuales está configurado el módulo y una descripción detallada de ellos. En la figura 2.7 se muestra la interfaz gráfica del software XCTU.

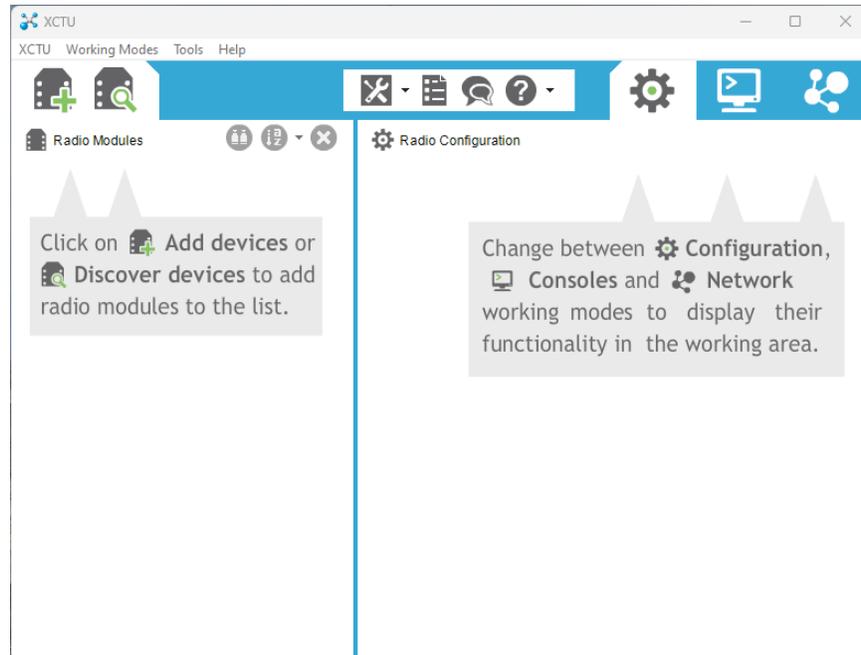


Fig. 2.8 Interfaz gráfica XCTU

Los aspectos más destacados de XCTU son las siguientes características:

- Pueden administrar y configurar varios dispositivos RF, incluso dispositivos conectados en forma remota (inalámbrico).
- El proceso de actualización del firmware y restaurar sin problemas la configuración de los módulos, manejando automáticamente el módulo y los baudios.
- Uso de dos consolas API y AT, fueron diseñadas desde cero para comunicarse con sus dispositivos de radio.
- Guardar sesiones de configuración y cargarlas en una PC diferente cuando se ejecute el XCTU.
- Integra un conjunto de herramientas que se pueden ejecutar sin tener ningún módulo RF conectado.
 - **Generador de marcos:** capaz de generar fácilmente cualquier tipo de marco API para guardar su valor.
 - **Interprete de marcos:** decodifique un marco API y vea su valor específico.
 - **Recuperación:** Recuperar los módulos de radio que tienen firmware dañado o están en modo de programación.

- **Cargar la sesión:** Cargar una sesión de consola guardada en cualquier PC que ejecute XCTU.
- **Prueba de alcance:** Realizar pruebas de alcance entre dos dispositivos en la misma red.
- **Explorador de firmware:** Conocer la biblioteca de Firmware de XCTU.

Para comenzar a utilizar el software XCTU, es necesario conectar el módulo RF a la placa de desarrollo, que tiene un módulo FTDI que saca una interfaz USB y luego enchufar el USB a la PC, es necesario poner el jumper en la posición UART en lugar de loopback.

2.4.1 Primera aplicación. Chat Básico.

La primera aplicación se utilizó con un chat básico para la transmisión de datos en tiempo real. El primer paso es poner el dispositivo en modo configuración y descubrir los módulos, seleccionando el puerto COM del USB al que se ha conectado la placa de desarrollo. Configurar los XBee's para la comunicación inalámbrica y luego comenzar a chatear. El texto que escriba un XBee se transmitirá de forma inalámbrica al otro XBee y viceversa, creando una comunicación bidireccional. En la figura 2.8 se muestra una imagen ilustrativa del funcionamiento básico que realiza el chat en tiempo real.



Fig. 2.9 Prueba uno. Chat básico entre dispositivos

Para la configuración de los módulos XBee con el XCTU, lo primero es poner el modo de configuración y descubrir los módulos, seleccionando el puerto COM del USB al que se ha conectado la placa de desarrollo.

Los dispositivos XBee transmiten por aire los datos que llegan al puerto serie del XCTU al segundo puerto serie del dispositivo XBee. Los datos son controlados desde el computador y gestionan los datos recibidos.

2.4.1.1 Comunicación entre dispositivos XBee.

Los módulos XBee se comunican entre ellos a través del aire enviado y recibiendo mensajes. Estos dispositivos no son capaces de gestionar los datos enviado o recibidos, sin embargo, pueden comunicarse con otros dispositivos a través de interfaz serie.

Los microcontroladores o los PCs pueden controlar que envían los dispositivos XBee y gestionan los mensajes inalámbricos entrantes.

Esta es la base sobre la que se define la especificación ZigBee, cuyo propósito es ofrecer una solución completa para este tipo de redes construyendo los niveles superiores de protocolos estándar.

Usando un Arduino/Xbee Shield como coordinador (maestro) y el segundo como router (esclavo) para lograr una configuración de punto a punto.

Cuentan con dos diferentes tipos de comunicación entre dispositivos XBee:

- **Comunicación inalámbrica:** Es la comunicación entre los módulos XBee.
- **Comunicación serie:** Es la comunicación entre módulos XBee y el microcontrolador o el PC a través de un puerto serie.



Fig. 2.2 Comunicación XBee-XBee.

Los módulos XBee transmiten y reciben información por medio de modulación de las ondas electromagnéticas. Para que la transmisión se realice en ambos módulos deben estar en la misma frecuencia y en la misma red. Esto se termina por dos parámetros:

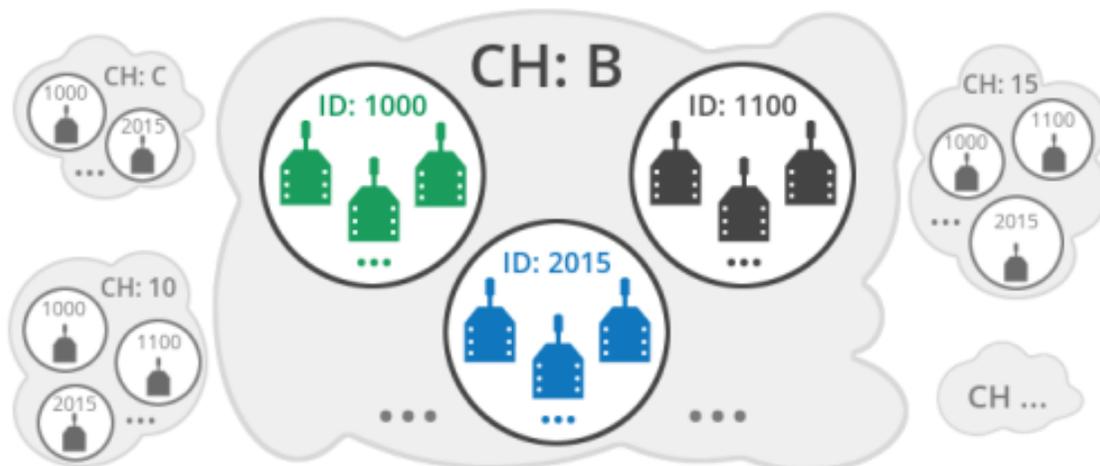


Fig. 2.11 Direccionamiento de módulos XBee.

- **Channel (CH):** Es la frecuencia usada para comunicar, es decir, el canal dentro de la red.
- **Personal Area Network Identifier (ID):** Es un identificador único que establece que los módulos están en la misma red.

Los modulos XBee solo reciben y transmiten datos a otros XBee dentro de la misma red (mismo ID) y usando el mismo canal (mismo CH).

2.4.1.2 Modelos de Comunicación

Existen dos tipos de comunicación y en ambos casos es bidireccional:

- **Punto a punto**
- **Punto a multipunto**

2.4.1.3 Punto a punto

La comunicación establecida en los módulos debe ser en el mismo canal (CH) y en la misma Network ID (ID), para lograr iniciar la comunicación es necesario saber la dirección MAC de 64-bit del módulo destinatario.

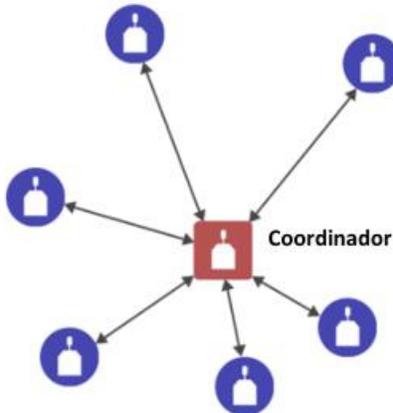


Fig. 2.12 Red de multipuntos conectados a un coordinador.

2.4.1.4 Punto a multipunto

En esta configuración el módulo puede comunicarse con un módulo o múltiples módulos que estén en la misma red. Para lograr esta comunicación debe haber un nodo central como coordinador con varios nodos remotos (end devices) conectados al nodo central.

La aplicación del protocolo IEEE 802.15.4 tiene dos roles:

- **Coordinador:** Es centro de la red o nodo principal. Inicia la red y permite a otros modulos conectarse, puede seleccionar la frecuencia del canal y hacer que la red este sincronizada. Para configurar el nodo principal hay que cambiar el parámetro CE (Coordinator Enable).
- **End device:** Es ek nodo remoto al coordinador. Capaz de comunicase con el coordinador y con otros end devices de la red. Se puede configurar modo sleep.

2.4.1.5 Conexión XBee con Arduino.

Las interconexiones es importante saber que dispositivo o modelo que se utiliza. En este ejemplo se presenta solamente las conexiones principales entre los dispositivos ZigBee con Arduino.

Estas conexiones representan las conexiones para la fuente de alimentación, el módulo debería estar conectado a un regulador de 3.3V con GND. Los pines TX y RX de ZigBee y Arduino están conectados entre sí. Arduino enviará algunas instrucciones a los módulos ZigBee y de acuerdo con estas instrucciones ZigBee responderá. De manera similar ZigBee recibe instrucciones de otros ZigBee a los que tiene dirección. Después de recibir instrucciones o datos de otros ZigBee.

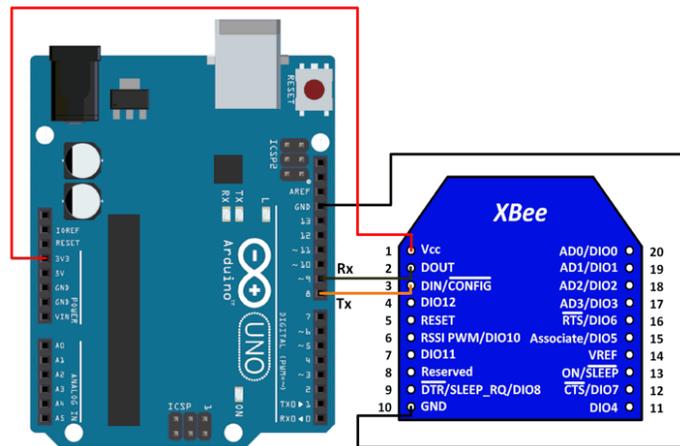


Fig. 2.13 Conexión Arduino-XBee

2.5 Parámetros eléctricos

Los parámetros eléctricos son de gran importancia en el desarrollo de las ciencias y de la humanidad, donde se involucran, voltajes, resistencias e intensidades de corriente eléctrica, que se encuentran presentes en todos aquellos dispositivos eléctricos de los cuales se comparten en la vida diaria, como lo son los celulares, computadoras, licuadoras, televisores, el consumo de la electricidad ya es algo común.

2.5.1 *Potencia Activa*

El voltaje y la corriente efectivos se representan con E e I, respectivamente, y los fasores E e I están en fase. Si se conecta un vatímetro (o wattmetro) a la línea, dará una lectura $P = EI$ watts, que siempre será positiva. El hecho de que la potencia siempre es positiva revela que siempre fluye del generador al resistor. Ésta es una de las propiedades básicas de la llamada potencia activa: aunque pulsa entre cero y un valor máximo, nunca cambia de dirección. El símbolo de la potencia activa es P y la unidad es el watt (W). El kilowatt (kW) y el megawatt (MW) son múltiplos del watt que se utilizan con frecuencia.

2.5.2 *Potencia Reactiva*

En un circuito con una reactancia XL la corriente I se retrasa 90° con respecto al voltaje E. Esta potencia consiste en una serie de pulsos positivos y negativos idénticos. La potencia que oscila de esta manera se llama potencia reactiva (símbolo Q), para distinguirla de la potencia activa. El producto EI también da la potencia reactiva. Sin embargo, para distinguir esta potencia de la potencia activa, se utiliza otra unidad: el Var. Sus múltiplos son el kilovar (kVAR) y el megavar (MVAR).

2.5.3 *Potencia Aparente*

Se puede considerar que las cargas que absorben tanto potencia activa P como reactiva Q están compuestas de una resistencia y una reactancia inductiva. Si se conecta un vatímetro y un varímetro al circuito, ambas lecturas serán positivas, es decir $P = EI$ watts y $Q = EI$ kVA, respectivamente. Si se conecta un amperímetro a la línea, indicará una corriente de I amperes. Como resultado, se podría pensar que la potencia suministrada a la carga es igual a EI watts. Pero, obviamente, esto es incorrecto porque la potencia consiste en un componente activo (watts) y un componente reactivo (vars). Por esta razón, el producto EI se llama potencia aparente. El símbolo de la potencia aparente es S. La potencia aparente no se expresa en watts ni en vars, sino en voltamperes. Los múltiplos son el kilovoltampere (kVA) y el megavoltampere (MVA).

2.5.4 *Factor de Potencia*

Se comenzará analizando el factor de potencia de un dispositivo o circuito de corriente alterna es la relación de la potencia activa P a la potencia aparente S, es decir factor de potencia = P/S donde:

- ❖ $P =$ potencia activa suministrada o absorbida por el circuito o dispositivo [Watts].
- ❖ $S =$ potencia aparente del circuito o dispositivo [kVA].

El factor de potencia se expresa como un número simple o como un porcentaje, es simplemente una forma de establecer qué fracción de su potencia aparente es potencia real, o activa. En un circuito monofásico el factor de potencia también mide el ángulo de desfase entre el voltaje y la corriente, aunque no siempre es real.

2.5.5 Triángulo de potencias

Debido a sus cambios, la corriente alterna produce en su circuito tres tipos de potencia en cuanto a la energía, que son la potencia aparente, potencia activa y potencia reactiva. La relación de estas tres potencias nos lleva a pensar en un triángulo rectángulo, siendo cada lado del triángulo un tipo de potencia. Se puede demostrar gráficamente la relación entre S, P y Q, mediante un triángulo de potencia. De acuerdo con lo mencionado, se aplican las siguientes reglas:

- ❖ La Potencia activa “P” absorbida por un circuito o dispositivo se considera positiva y se traza horizontalmente hacia la derecha.
- ❖ La Potencia activa P suministrada por un circuito o dispositivo se considera negativa y se traza horizontalmente hacia la izquierda.
- ❖ La Potencia reactiva “Q” absorbida por un circuito o dispositivo se considera positiva y se traza verticalmente hacia arriba.
- ❖ La Potencia reactiva “Q” suministrada por un circuito o dispositivo se considera negativa y se traza verticalmente hacia abajo.
- ❖ La Potencia aparente “S”, es la suma vectorial de las potencias reactivas y activas mencionadas anteriormente.

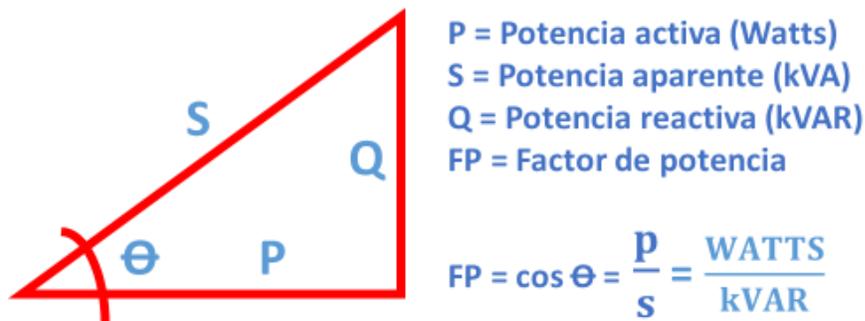


Fig. 2.14 Triángulo de Potencias y Factor de Potencia

2.6 Sensor para corriente de efecto Hall ACS712

El sensor de corriente ACS712 es capaz de soportar mediciones de corriente de hasta 30 A en AC O DC. Está compuesto por un sensor de efecto Hall que detecta el campo magnético producido del paso de la corriente por un alambre interno y convirtiendo este campo en un voltaje variable. Significa que cuando se tenga una corriente mayor, mayor será el voltaje producido en el pin. Esta diseñado idealmente para proyectos mecatrónicos aplicables en ambientes industriales o comerciales, el módulo cuenta con una conexión fácil de usar, tiene una bornera para conectar la línea que se necesita medir y tres pines más, dos para conectar la alimentación y un pin para la salida analógica.

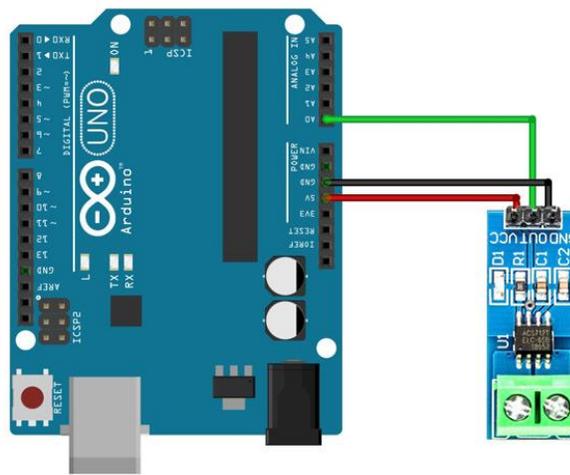


Fig. 2.15 Diagrama de conexiones para conectar Arduino con sensor ACS712

2.6.1.2 Especificación y características

- Modelo: ACS712-30A
- Sensor de corriente: ACS712
- Voltaje de Trabajo: 5V
- Rango Máximo (+/-) 30 A
- Salida análoga: 100mV/A
- Salida cuando no detecta corriente: 2 VCC
- Sensibilidad: 66 mV/A
- Resistencia interna: 1.2 mΩ
- Indicador LED
- Mínimo voltaje de aislamiento entre pines 1-4 a pines 5-8: 2.1 kVRM
- Dimensiones: 31mm x 13mm x 11.8 mm

2.7 Sensor para voltaje ZMPT101B

Modulo utilizado para la medición de tensión por fase. El sensor de voltaje AC esta diseñado a partir de un transformador, por lo que se puede utilizar para medir voltaje de corriente alterna. Las aplicaciones ideales es el monitoreo de energía eléctrica, muy común en aplicaciones domóticas como: Medidores de energía conectados a internet por Wifi, Bluetooth, ZigBee. Por los principios de funcionamiento de los transformadores solo puede medir voltaje AC. El módulo ZMPT101B soluciona problemas en base a la reducción de voltaje AC de entrada a un voltaje AC de amplitud menor que pueda ser leído por el Arduino o cualquier otro microcontrolador. Es recomendable calibrar a salida con un voltímetro previamente de conectar cualquier dispositivo a la salida.

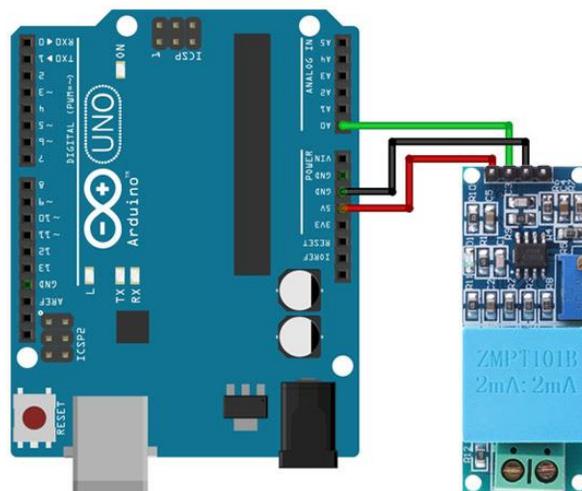


Fig. 2.16 Diagrama de conexión para sensor ZMP101B

2.7.1.2 Especificación y características

- Voltaje
 - De alimentación: 5V – 30V DC
 - Alterno de entrada máx: 250V AC
- Corriente nominal de entrada y salida: 2mA
- Voltaje de suministro: 5 VCD
- Diferencia de fase: $<30^\circ$ (a 50ohm)
- Rango lineal: 0-3mA (a 50ohm)
- Precisión: 0.2%
- Aislamiento eléctrico entrada: hasta 3000V

2.8 Relevador 5V

El relevador es un interruptor cuyo control corre por cuenta de un circuito eléctrico, a través de una bobina y un electroimán este incide sobre diversos contactos para la apertura o el cierre de otros circuitos, que funcionan de manera independiente.

Este relevador permite controlar dispositivos que trabajen máximo a 250V/10A / 125V/10A AC y 30V/10A DC por medio de una señal de control de 5V, ideales para implementarlos en placas de circuito impreso PCBs o tener el reemplazo para reparar tu módulo o proyecto electrónico.

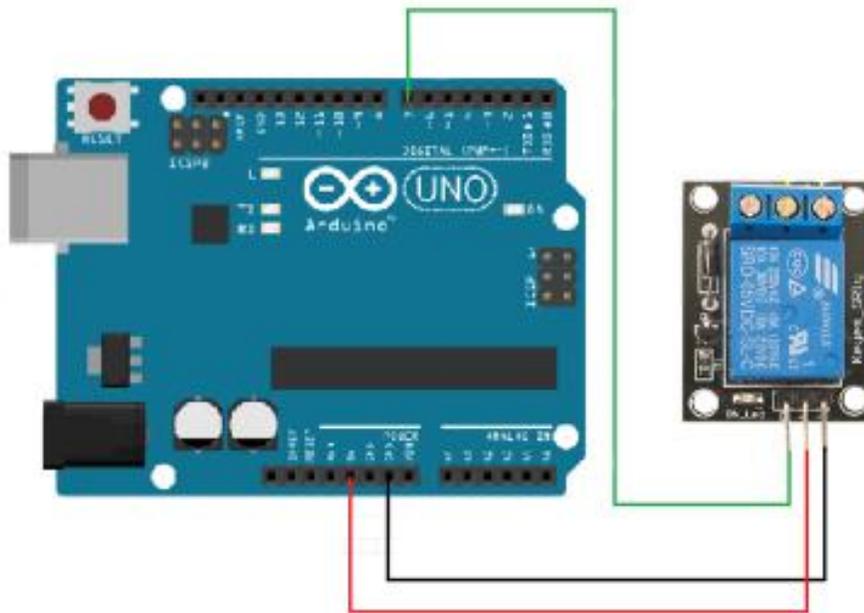


Fig. 2.17 Diagrama de conexiones para relevador de 5V

2.8.1.2 Especificación y características

- Voltaje nominal de la bobina: 5VDC
- Voltaje y Corriente Máx Carga:
 - AC: 240V / 10A ó 125V / 10A
 - DC: 30V / 10A ó 28V / 10A
- Corriente nominal en bobina 75mA
- Expectativa de vida: 100,000 accionamientos
- Temperatura de operación -25°C a 70 °C

2.9 Contactor trifásico con bobina auxiliar

El relevador de 5V cumple la misma función que en el punto 2.8. El relevador puede manejar la alta potencia necesaria para controlar directamente un motor eléctrico. Para una carga trifásica se necesita añadir un contactor de tres polos con una bobina auxiliar, siendo esta bobina auxiliar controlada por el relevador, manejando una carga más grande con tan solo una señal de 5V.

El modelo este compuesto por dos circuitos:

- Circuito de control
- Circuito de fuerza

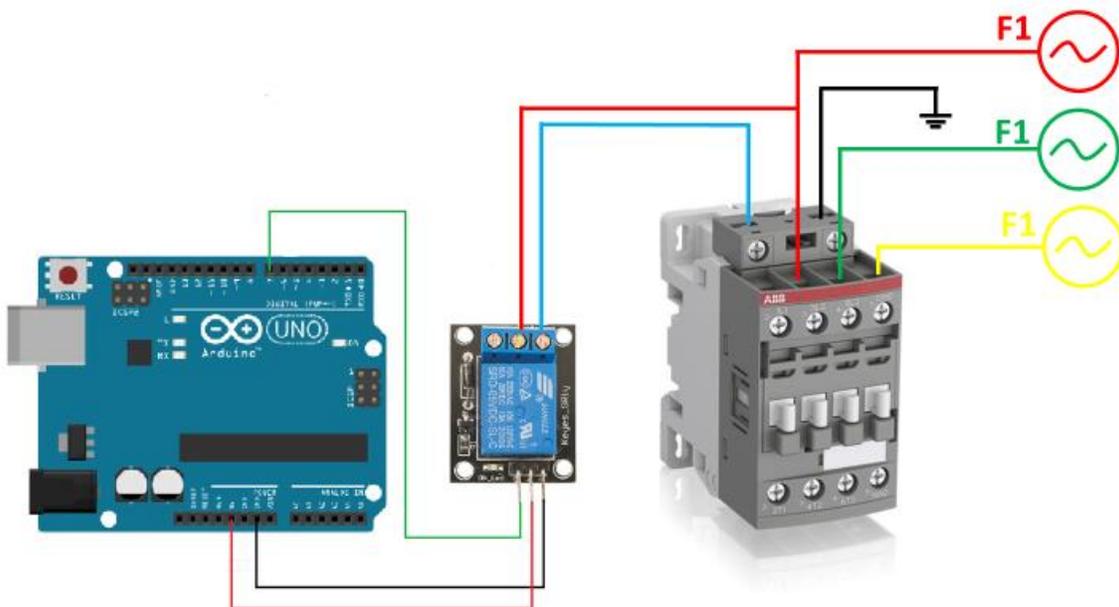


Fig. 2.18 Diagrama de conexiones para circuito de fuerza

CAPITULO 3

DISEÑO E IMPLEMENTACION DEL PROTOTIPO

El presente proyecto fue realizando, utilizando dos microcontroladores Arduino Mega 2560 conectados cada uno a una antena XBee para la transmisión de los parámetros eléctricos y funciones de accionamiento. El primer Arduino es el encargado de la parte de medición y transmisión; y el segundo es el encargado de la recepción. El cual está conectado internamente a una interfaz gráfica creada en Visual Estudio para una mejor visualización.

3.1 Medición de Voltaje trifásico

Para este prototipo se utilizaron tres sensores ZMPT101B, los cuales son sensores de voltaje para corriente alterna, y están formados internamente por un transformador que reduce el voltaje para que sean aceptado por el microcontrolador Arduino.

Los sensores están conectados a las entradas analógicas del Arduino, se puede observar en la figura 3.1 y mediante las librerías y programación del dispositivo en Arduino IDE se obtiene el valor de voltaje en tiempo real.

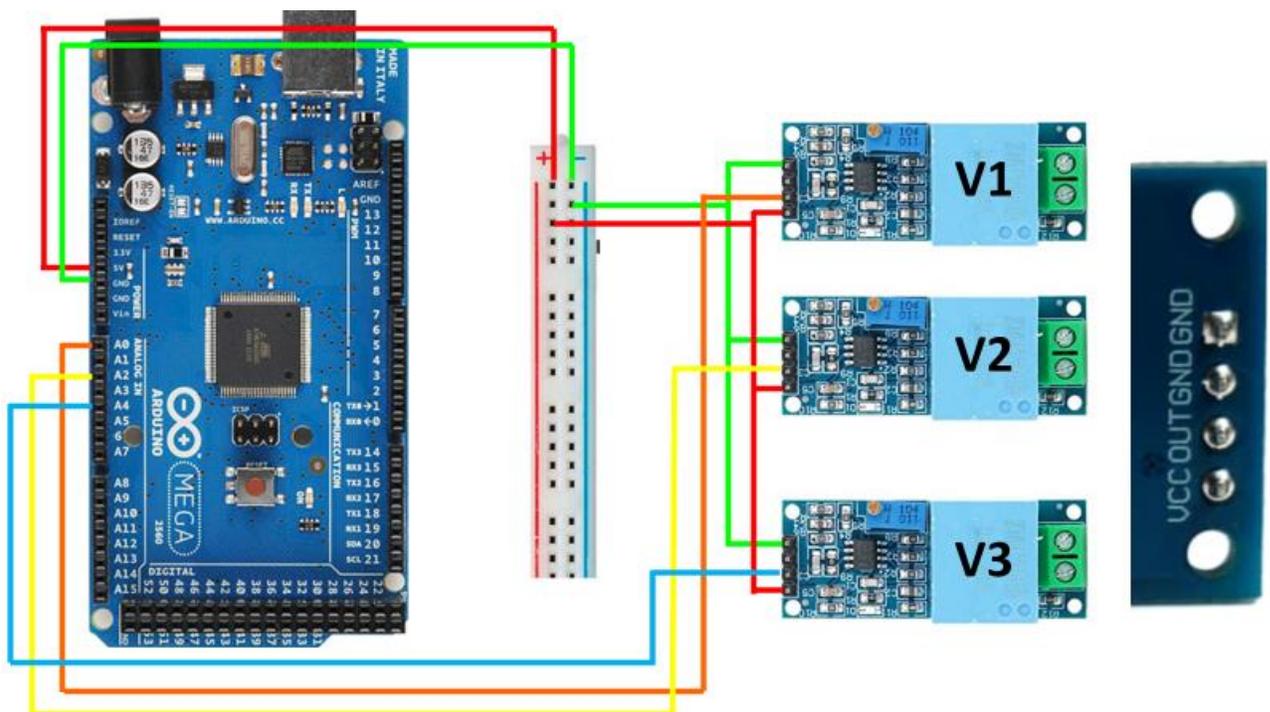


Fig. 3.1 Conexión de tres sensores de voltaje ZMPT101B con Arduino MEGA

Al conectar el módulo ZMPT101B AC al Arduino, se conectó las salidas analógicas OUT de los sensores a los pines analógicos A0, A2, A4 y luego conectar los pines de alimentación – GND y VCC a los pines GND y +5V del microcontrolador. Después se conectó la parte de alimentación, se conectaron los pines L y N en paralelo a la carga.

3.2 Medición de Corriente trifásica

Para este prototipo se utilizaron tres sensores ASC712, los cuales son sensores de corriente, los sensores de corriente funcionan por efecto Hall transformando la corriente en una señal de voltaje análoga que va de -5

Los sensores están conectados a las entradas analógicas del Arduino, se puede observar en la figura 3.2 y mediante las librerías y programación del dispositivo en Arduino IDE se obtiene el valor de voltaje en tiempo real.

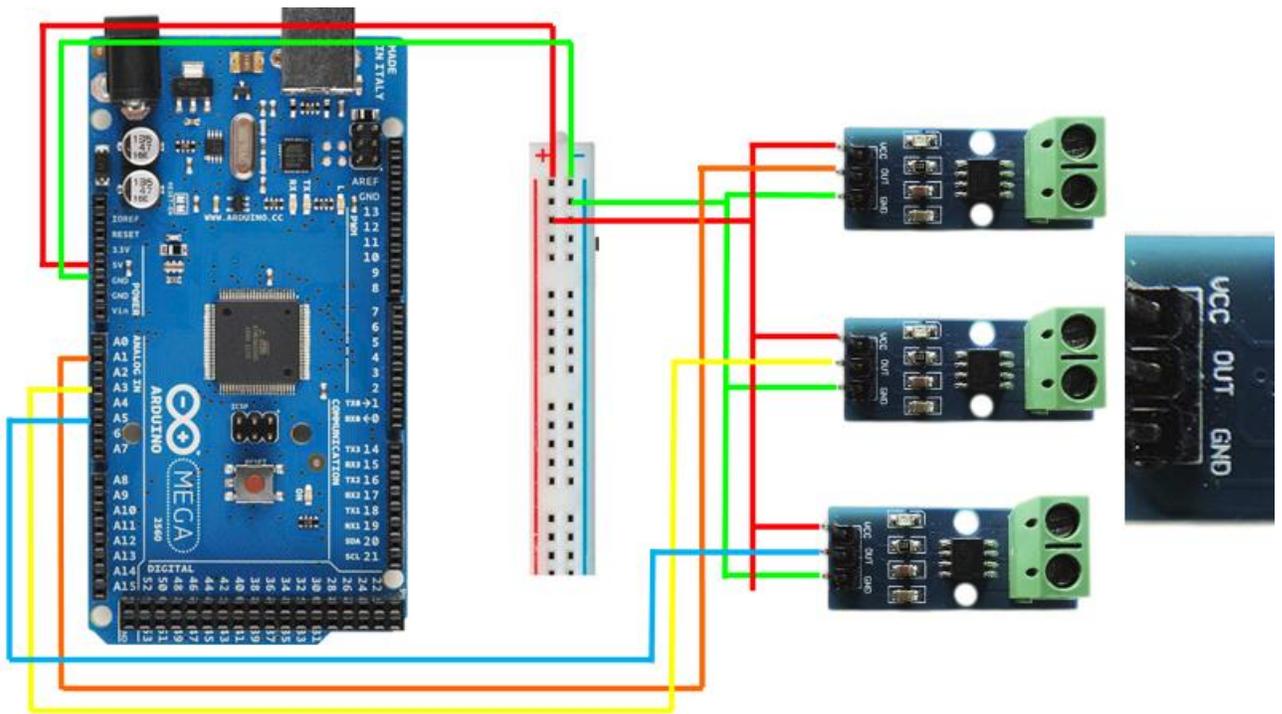


Fig. 3.2 *Conexión de tres sensores de corriente ASC712 con Arduino MEGA*

Al conectar el módulo ACS712 AC al Arduino, se conectó las salidas analógicas OUT de los sensores a los pines analógicos A1, A3, A5 y luego conectar los pines de alimentación – GND y VCC a los pines GND y +5V del microcontrolador. Después se conectó la parte de alimentación, se conectaron los pines de entrada y salida en serie con la carga.

3.3 Programación Arduino-IDE

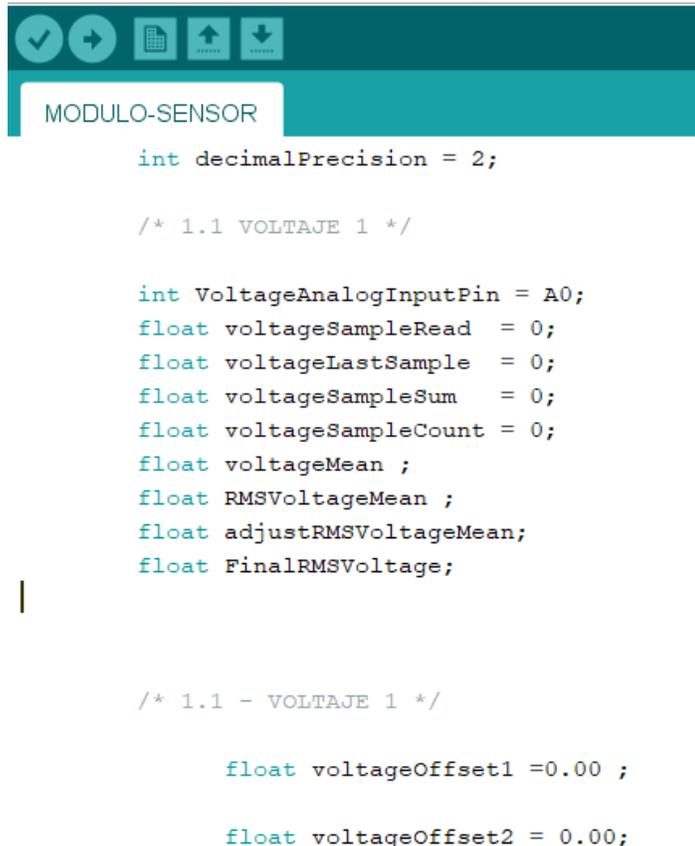
Para la programación de los sensores de voltaje y corriente se necesita una variable, lectura, de tipo float para guardar la lectura que haremos en el pin analógico A0 al A6, lo que nos dará un valor entre 0 y 1023. La segunda variable volt, también de tipo float, es para guardar la conversión de la medida en voltios. En la función setup() iniciamos el puerto serial con Serial.begin() e indicamos que el pin A0 de Arduino es un puerto de entrada. En la función loop() realizamos la lectura desde el pin analógico A0 al A6, realizamos la conversión en voltios y, por último, hacemos que se vea por el monitor serial.

La programación de la lectura de los parámetros eléctricos fueron los siguientes pasos:

1. Para el muestreo de las variables, se declararon diversas variables como; almacenamiento y cálculo de las condiciones de voltaje, corriente y potencias.

MODULO-SENSOR Arduino 1.8.19 (Windows Store 1.8.57.0)

Archivo Editar Programa Herramientas Ayuda



```

int decimalPrecision = 2;

/* 1.1 VOLTAJE 1 */

int VoltageAnalogInputPin = A0;
float voltageSampleRead = 0;
float voltageLastSample = 0;
float voltageSampleSum = 0;
float voltageSampleCount = 0;
float voltageMean ;
float RMSVoltageMean ;
float adjustRMSVoltageMean;
float FinalRMSVoltage;

/* 1.1 - VOLTAJE 1 */

float voltageOffset1 =0.00 ;

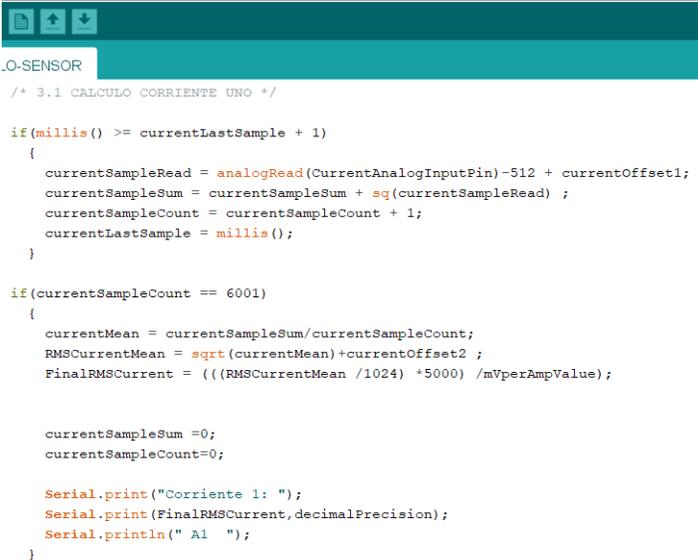
float voltageOffset2 = 0.00;

```

Fig. 3.3 Ejemplo declaración de variable de Voltaje 1.

2. Se procede a realizar el cálculo del voltaje RMS (Root Mean Square) de cada fase, tomando una muestra cada milisegundo, realizando las operaciones necesarias ejecutando el valor del voltaje cada segundo. Siguiendo el mismo principio se realizan las

operaciones para el cálculo de la corriente RMS por cada fase, mostrando ambos valores en el puerto serial del Arduino-IDE.



```

LO-SENSOR Arduino 1.8.19 (Windows Store 1.8.57.0)
editar Programa Herramientas Ayuda
.O-SENSOR
/* 3.1 CALCULO CORRIENTE UNO */

if(millis() >= currentLastSample + 1)
{
  currentSampleRead = analogRead(CurrentAnalogInputPin)-512 + currentOffset1;
  currentSampleSum = currentSampleSum + sq(currentSampleRead) ;
  currentSampleCount = currentSampleCount + 1;
  currentLastSample = millis();
}

if(currentSampleCount == 6001)
{
  currentMean = currentSampleSum/currentSampleCount;
  RMSCurrentMean = sqrt(currentMean)+currentOffset2 ;
  FinalRMSCurrent = (((RMSCurrentMean /1024) *5000) /mVperAmpValue);

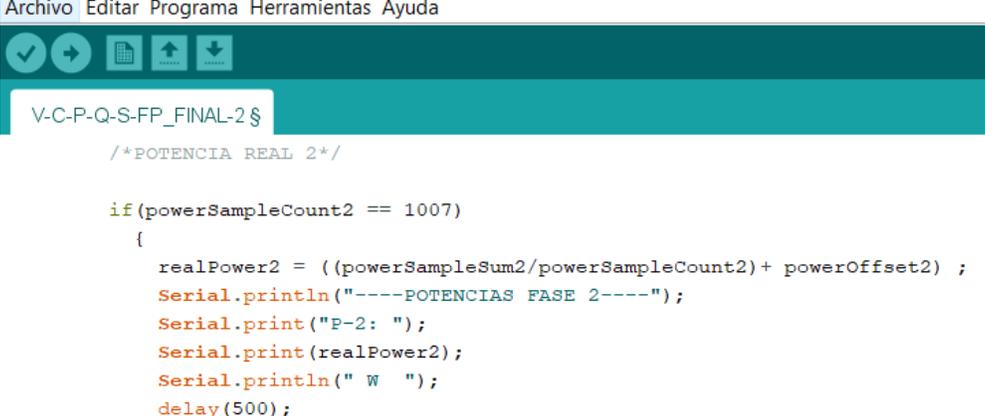
  currentSampleSum =0;
  currentSampleCount=0;

  Serial.print("Corriente 1: ");
  Serial.print(FinalRMSCurrent,decimalPrecision);
  Serial.println(" A1 ");
}

```

Fig. 3.4 Muestra tomada por segundo para corriente Fase 1

- El cálculo de la potencia real se lleva a cabo de acuerdo con la siguiente metodología: reside en medir la diferencia de tiempo para alcanzar el valor máximo ó sea el valor 0 entre ambas ondas mientras se usa el mismo punto o tiempo de inicio. Este método es el método más simple para obtener el Angulo de fase, basándose únicamente en 2 puntos de referencia entre el punto máximo y el punto inicio.



```

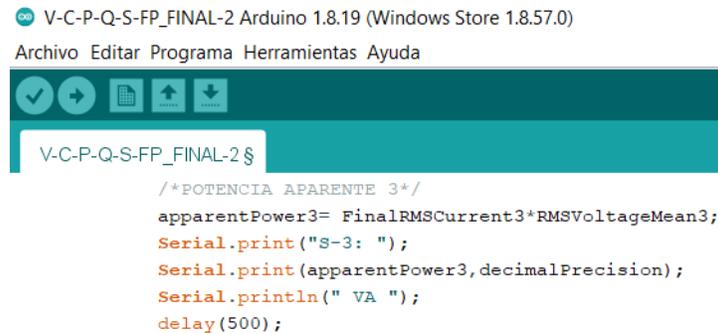
V-C-P-Q-S-FP_FINAL-2 Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda
V-C-P-Q-S-FP_FINAL-2 §
/*POTENCIA REAL 2*/

if(powerSampleCount2 == 1007)
{
  realPower2 = ((powerSampleSum2/powerSampleCount2)+ powerOffset2) ;
  Serial.println("----POTENCIAS FASE 2----");
  Serial.print("P-2: ");
  Serial.print(realPower2);
  Serial.println(" W ");
  delay(500);
}

```

Fig. 3.5 Calculo de potencia real de Fase 2

- El muestreo de la potencia aparente se calcula mediante la multiplicación de los valores del voltaje y corriente eficaz (RMS); a su vez la potencia reactiva se calcula despejando de la fórmula del triángulo de potencia quedando como diferencia la raíz cuadrada de la potencia aparente y real.



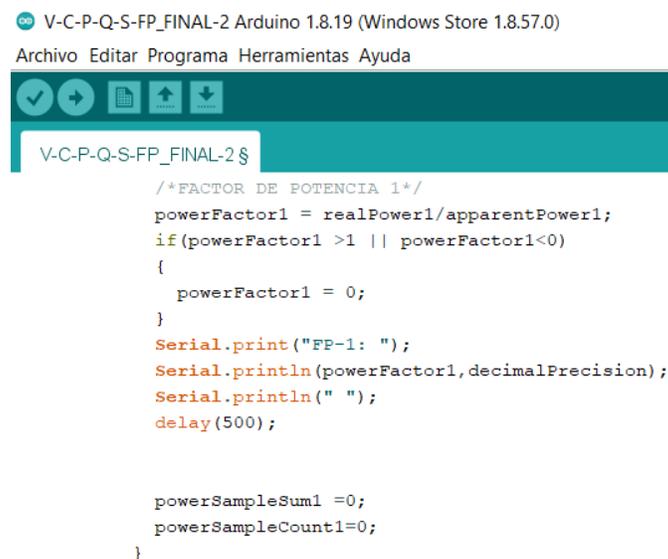
```

V-C-P-Q-S-FP_FINAL-2 Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda
V-C-P-Q-S-FP_FINAL-2 §
/*POTENCIA APARENTE 3*/
apparentPower3= FinalRMSCurrent3*RMSVoltageMean3;
Serial.print("S-3: ");
Serial.print(apparentPower3,decimalPrecision);
Serial.println(" VA ");
delay(500);

```

Fig. 3.6 Ejemplo de lectura de potencia aparente de Fase 3 c

- Como sabemos los valores máximos y mínimos que arrojan los sensores al aplicar tensión y corriente sobre estos, debemos diseñar con Arduino un programa que nos permita obtener el factor de potencia. obtener los valores instantáneos de tensión y corriente, calcular el valor medio y obtener la potencia activa. Con esta potencia y la potencia aparente calcularemos el factor de potencia.



```

V-C-P-Q-S-FP_FINAL-2 Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda
V-C-P-Q-S-FP_FINAL-2 §
/*FACTOR DE POTENCIA 1*/
powerFactor1 = realPower1/apparentPower1;
if(powerFactor1 >1 || powerFactor1<0)
{
  powerFactor1 = 0;
}
Serial.print("FP-1: ");
Serial.println(powerFactor1,decimalPrecision);
Serial.println(" ");
delay(500);

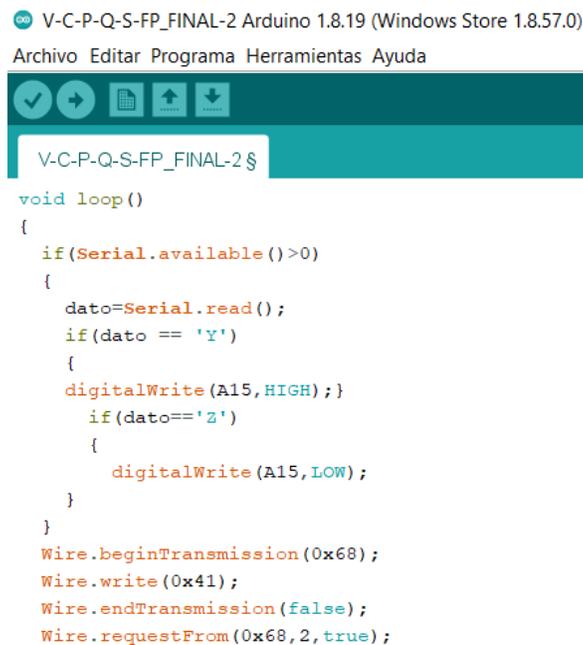
powerSampleSum1 =0;
powerSampleCount1=0;
}

```

Fig. 3.7 Calculo de factor de potencia de la Fase 1

Vamos a obtener el factor de potencia determinando el desfase que existe entre la tensión y corriente, para esto necesitamos establecer una referencia que servirá como punto inicial de la medición y punto final de la medición. Este punto tiene que estar presente en todo momento de una medición.

6. Las salidas analógicas del Arduino son enviadas a la interfaz gráfica el cual será la encargada de realizar el cambio del estado del relevador para interrumpir la corriente eléctrica y haciendo la interpretación de las variables a medir por medio de la interfaz gráfica, en caso de haber interrumpido la corriente eléctrica se dan como resultado mediciones nulas.



```
V-C-P-Q-S-FP_FINAL-2 Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda
V-C-P-Q-S-FP_FINAL-2$
void loop()
{
  if (Serial.available() > 0)
  {
    dato = Serial.read();
    if (dato == 'Y')
    {
      digitalWrite(A15, HIGH);
    }
    if (dato == 'Z')
    {
      digitalWrite(A15, LOW);
    }
  }
  Wire.beginTransaction(0x68);
  Wire.write(0x41);
  Wire.endTransmission(false);
  Wire.requestFrom(0x68, 2, true);
}
```

Fig. 3.8 Señales de salidas analógicas de Arduino para enviarlas a interfaz grafica

Al describir de manera resumida estos dos métodos parecen ser sencillos, fáciles de implementar, pero ejecutarlo tiene una gran cantidad de dificultades y de pasos que necesitan un buen conocimiento de esta plataforma. Hemos elegido el primer modelo para comenzar, ya que a simple razonamiento se lee como un modelo más sencillo y fácil, adecuándonos a nuestras necesidades que tenemos poco tiempo para diseñarlo e implementarlo.

3.4 Transmisión inalámbrica

La comunicación inalámbrica se realizó con ayuda de los dispositivos Digi XBee3 ZigBee 3.0 teniendo como alcance hasta 1200 metros si se utiliza en espacios abiertos.

En ambos microcontroladores se conectaron los módulos los cuales pueden transmitir y recibir información de manera continua. El primer Arduino es encargado de hacer las mediciones y enviar los datos recolectados el segundo Arduino que a su vez se encarga de leer las mediciones recibidas e ilustrarlas en la interfaz gráfica.

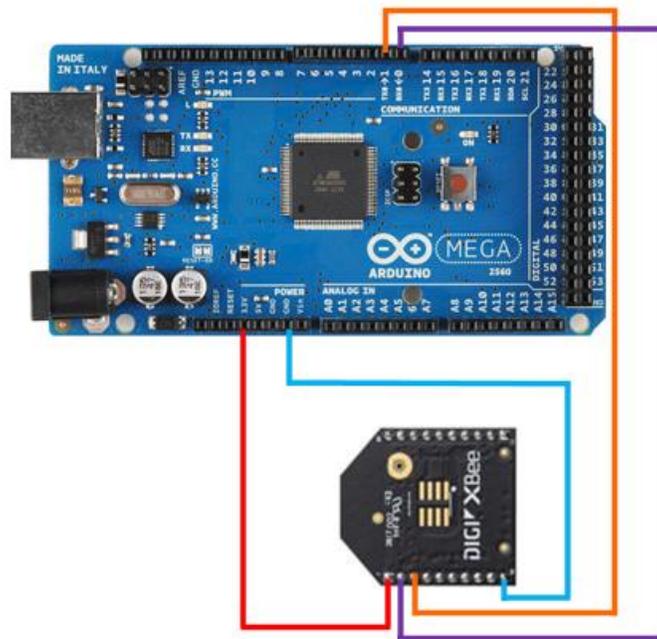


Fig. 3.9 Diagrama de Conexiones del Módulo XBee-Arduino

Al conectar el módulo XBee al Arduino, se conectó las salidas analógicas OUT de la antena a los pines analógicos TX y RX, luego conectar los pines de alimentación – GND y VCC a los pines GND y +3.3V del microcontrolador. Creando una comunicación serial entre dos dispositivos de manera inalámbrica.

Los módulos de radio XBee pueden ser usados con un número mínimo de conexiones: Power (3.3V), GND y TX/RX de la UART, junto con otras conexiones recomendables como reset y sleep. La mayoría de los módulos XBee tienen otras conexiones como flow control, input/output (I/O), analog-to-digital converter (A/D) and indicator. Los módulos XBee funcionan a 3.3V y los pines no son tolerantes a 5V. Desde Arduino podemos alimentar un módulo XBee, pero la comunicación serie en Arduino es a 5V y en el módulo XBee es a 3.3V.

3.5 Interfaz grafica

La interfaz gráfica, o mejor conocida como Graphical User Interface (GUI) es la forma en que podemos interactuar con los microcontroladores sin introducir comandos de texto en Arduini-IDE. Es un entorno visual amigable que nos permite realizar cualquier acción sin necesidad de tener conocimiento de programación.

Los GUI son los entornos Windows, MacOS o Android, gracias a los cuales pueden enviar comandos a través de gestos o movimientos de ratón, sin necesidad de introducir ningún código.

3.5.1 Instalación de Software Visual Studio 2019

La primera herramienta que se utilizó es el programa Visual Estudio 2019, descargando la versión gratuita de 1.3MB.

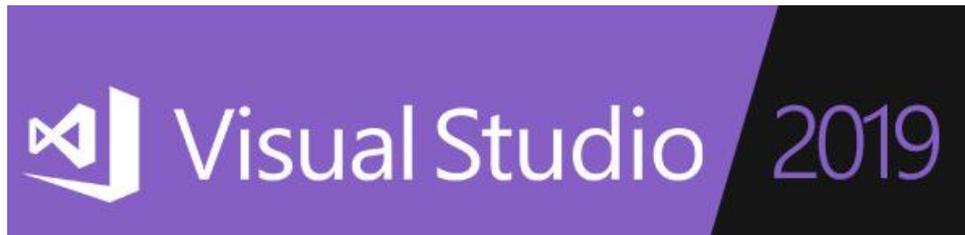


Fig. 3.10 Visual Studio Community 2019

Se realizó una interfaz gráfica con Visual Estudio en la opción de “Create a new project”, con lenguaje C#

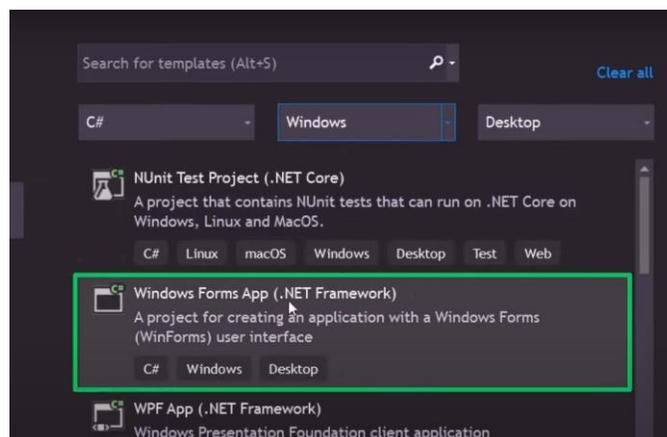


Fig. 3.11 Creación de nuevo proyecto con lenguaje C#

Se establece el nombre del proyecto, la localización para guardar el proyecto en el ordenador y estableciendo el Framework más reciente disponible, dando click en créate comenzara a crear una serie de carpetas para la creación del proyecto.

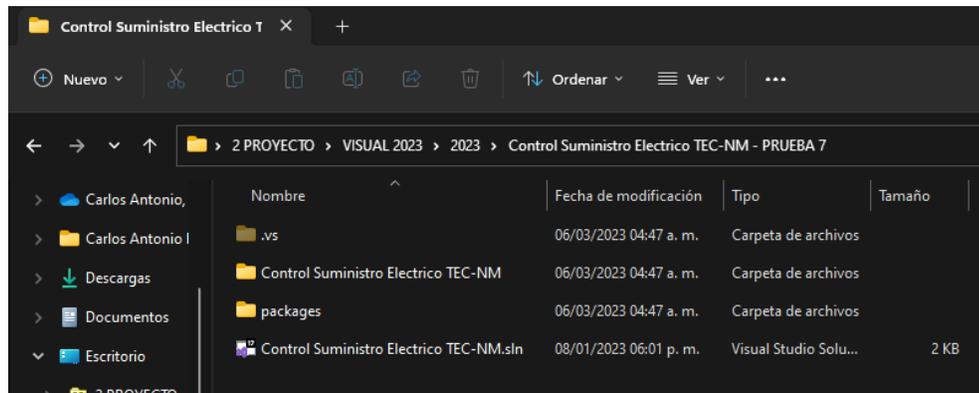


Fig. 3.12 Carpetas Creadas por Visual Studio 2019

3.5.1 USB-SERIAL COM

Se establecen propiedades del proyecto en la sección de “Todo Windows Forms” o “Cuadro de herramientas”; como colores de fondo y tamaño del Forms, del Toolbox se agregan GroupBox para separar la parte de los Puertos COM y la parte del Control de la carga, se establecen tamaños de letra y tamaños GroupBox.

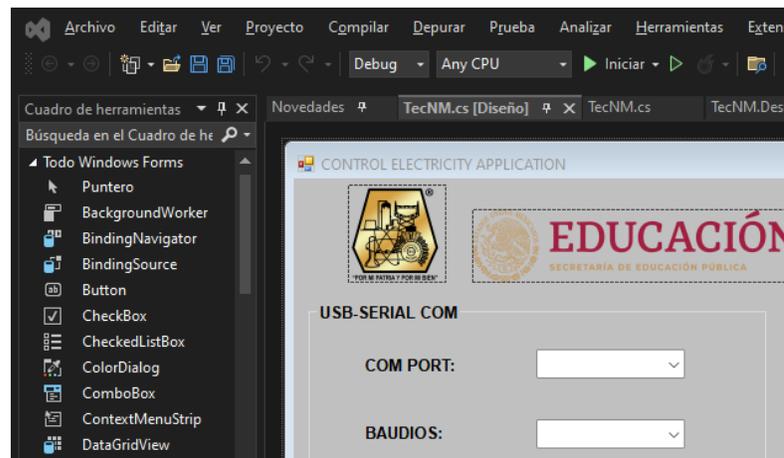


Fig. 3.13 Todo Windows Forms o Cuadro de herramientas

3.5.1.2 COM PORT

En el primer GroupBox se crea “USB - SERIAL COM” y se establecen etiquetas labelPort para controlar el puerto serie donde se conecta el microcontrolador y los baudios

por segundo a los cuales estaría la comunicación Serial, estas etiquetas se llaman “COM PORT” y “BAUDIOS” para tener identificados las funciones de los botones, con su respectivo taño, letra y color.

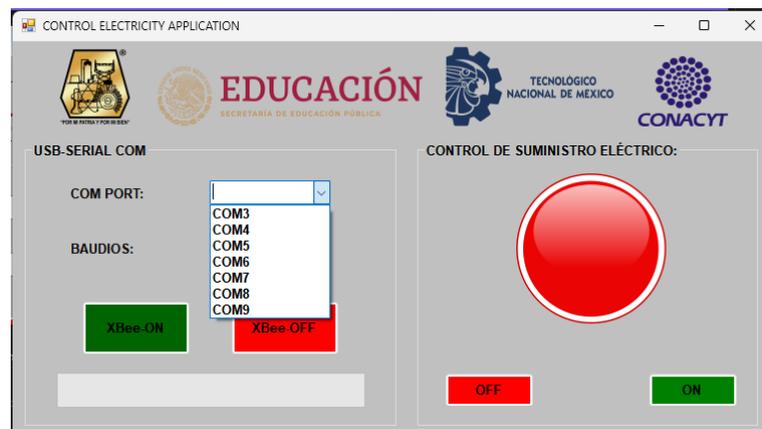


Fig. 3.14 GroupBox 1 para establecer comunicación serial

En este mismo GroupBox se insertan dos componentes “comboBox” llamados “comboBoxPort”, “comboBoxBaudRate” y dos botones “XBee On” y “XBee Off” para establecer o detener la comunicación Serial entre ambos dispositivos Arduino con comunicación Zigbee.

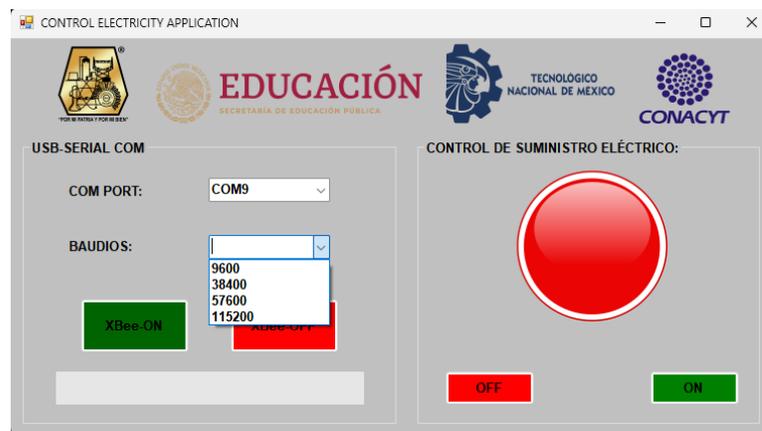


Fig. 3.15 GroupBox 1 para establecer baudios por segundo a 9600

3.5.2 CONTROL DE SUMINISTRO ELÉCTRICO

En el segundo GroupBox se crea “CONTROL DE SUMINISTRO ELÉCTRICO” y se establecen dos botones para el suministro eléctrico, “ON” y “OFF” para tener identificados las funciones de los botones, con su respectivo taño, letra y color, estas señales son representadas de forma dinámica, pero son el lenguaje binario que usa el código de configuración Arduino-IDE.

3.5.2.1 Eventos

Seleccionando en las propiedades del elemento también podemos acceder a los eventos que involucran a este. Haciendo doble click sobre alguno de ellos nos creara una función que ejecutara el código que nosotros proporcionemos cuando el evento suceda.

Cundo ejecutamos el programa y hacemos click en el botón el resultado será el siguiente;

3.5.2.2 Dibujo de figuras

Para dibujar o cambiar la imagen de las figuras sobre el form podemos hacer uso de los PictureBox, a la inserción de este similar al resultado de los elementos del ToolBox.

Haciendo click en el “button ON” se dibuja un o inserta una imagen circular de color verde haciendo referencia al suministro activo, cuando se haciendo click en el “button OFF” se dibuja una imagen circular color verde haciendo referencia al suministro suspendido.

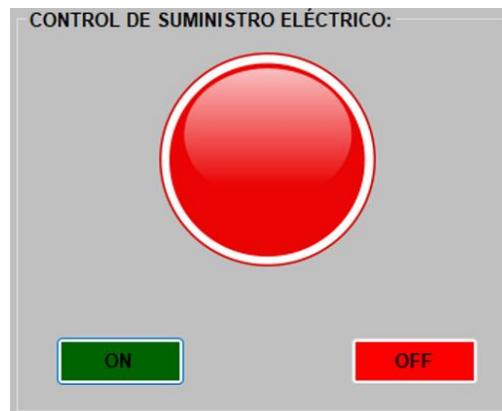


Fig. 3.16 GroupBox 2 para control de suministro para eventos de aplicación.

Empezamos por establecer una referencia en Arduino, que significa tomar el valor de un punto determinado dentro del espectro de valores que arrojan las entradas digitales en este caso, pero debe de estar presente cualquier medición que podamos hacer.

Es por esto que nuestro punto de referencia e inicio de medición es el cruce por cero de la función senoidal entregadas por los sensores, en Arduino este valor en tensión es de 2.5V y en las entradas análogas aproximadamente de un valor de 511.



Fig. 3.17 Interfaz Gráfica en Visual Estudio completa

Para la operación del contactor de fuerza es caracterizado porque solo puede adoptar uno de dos posibles estados o niveles: el estado de señal “Y” y el estado de señal “Z”.

Asimismo, estos estados cuando se relacionan, de acuerdo a su condición eléctrica, se dice: no existe tensión y existe tensión, la magnitud de la tensión no interesa, ya que dependerá del componente electrónico que pueda asumir esta tensión nominal y controlar bobina auxiliar del contactor trifásico.

Para obtener mayor capacidad de procesamiento a veces se agrupan los bytes, formando lo que se denomina la palabra. La palabra es una unidad mayor compuesta de 16 bits = 2 bytes. Los bits de una palabra se agrupan de derecha a izquierda, tomando como número de bit del 0 al 15. En una palabra, se pueden representar hasta 16 señales binarias, puede usarse para almacenar un número cuya magnitud como máximo sería:

CAPITULO 4

PRUEBAS Y RESULTADOS DEL PROTOTIPO

En este capítulo se concentra en forma general el proceso de los resultados del medido AMI. Se explican los procedimientos y características de los sensores para abordan los diferentes métodos de análisis que se implementaron en la realización del prototipo.

4.9 Modelado del suministro eléctrico trifásico

Para realizar el modelado de un consumidor con carga trifásico, se necesitó un maquina motriz, mejor conocido como motor de inducción.

4.9.1 Motor de inducción

Los motores se denominan motores eléctricos que funcionan con corriente alterna y funcionan con este tipo de alimentación eléctrica. Un motor es una máquina motriz, esto es, un aparato que convierte en forma determinada la energía suministrada en el equipo a una energía mecánica de rotación. Un motor eléctrico convierte la energía eléctrica en fuerzas de giro por medio de la acción mutua de los campos magnéticos. Uno de los casos, tales como las embarcaciones navales, donde su principal fuente de energía es de corriente continua, o donde se requiere un gran margen de velocidades de giro, pueden emplearse motores de corriente continua. Sin embargo, la gran parte de los motores modernos trabajan con fuentes de corriente alterna.

Un motor de inducción consta de partes indispensables, como el estator que es la parte fija del motor constituido por una carcasa en la que se fija una corona de chapas de acero al silicio provistas de ranuras, y los bobinados de sección apropiada están dispuestos en dichas ranuras formando las bobinas que se dispondrán en tantos circuitos como fases tenga la red a la que se conectara la máquina. El rotor es la parte móvil del motor, está situado en el interior del estator y consiste en un núcleo de chapas de acero al silicio apiladas que forman un cilindro, en el interior del cual se dispone una bobina eléctrica. Los más utilizados son el rotor de jaula de ardilla y el rotor bobinado. En la Figura 4.1 se muestra un motor de inducción jaula de ardilla y todos sus componentes tanto internos como externos.

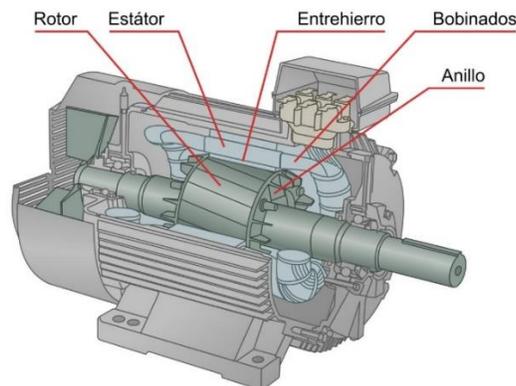


Fig. 4.1 Motor de inducción jaula de ardilla

A estos motores se les conoce como motores de inducción porque su funcionamiento principal se basa en la interacción de campos magnéticos producidos por corrientes eléctricas alternas.

Las corrientes que circulan por el rotor son producidas por el fenómeno de inducción electromagnética, conocido comúnmente como ley de Faraday como se muestra en la Figura 4.2, que establece que si una espira es atravesada por un campo magnético variable en el tiempo se crea entre sus extremos una diferencia de potencial.

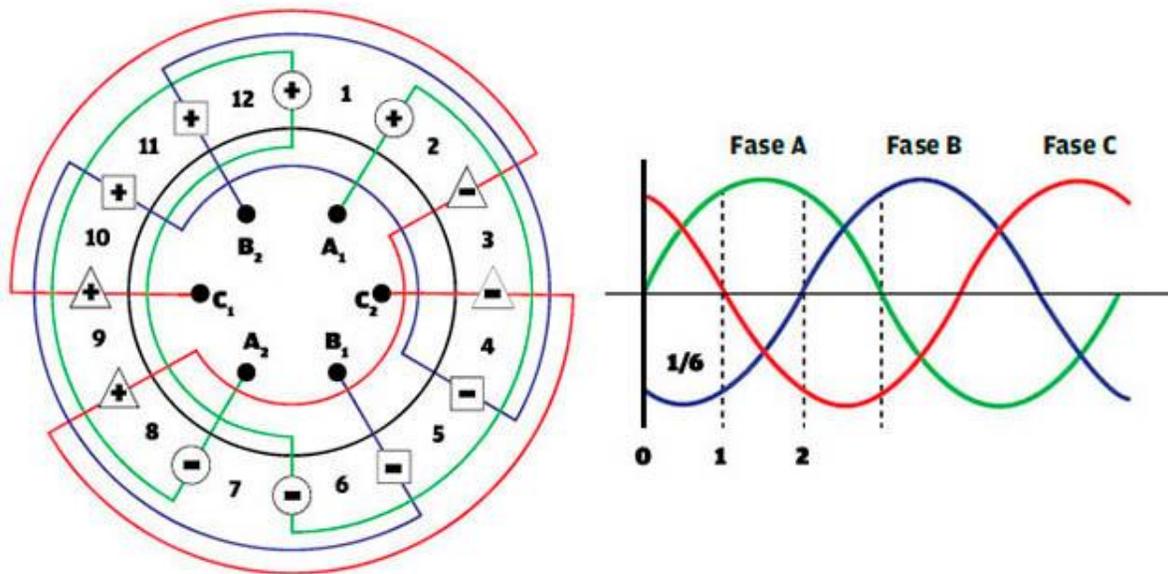


Fig. 4.2 Distribución de las bobinas del motor

Si se distribuye espacialmente alrededor del estator de un motor los bobinados de un sistema de tensiones trifásica desfasadas 120° se genera un campo magnético giratorio. La velocidad de giro de este campo magnético, denominada velocidad de sincronismo.

4.10 Pruebas de Voltaje con Arduino-IDE

Para las mediciones de voltaje se utilizaron sensores de Voltaje ZMPT101B conectado a las terminales analógicas del microcontrolador Arduino MEGA 2560. Se realizaron dos pruebas de funcionamiento, una prueba para un circuito monofásico utilizando un sensor de Voltaje y una prueba de Voltaje trifásico utilizando tres sensores de Voltaje como se muestra a continuación. La amplitud de la salida de voltaje se regula mediante el potenciómetro en el sensor dando un máximo de 5V, de modo que es importante realizar esta calibración utilizando otro instrumento de medición.

El siguiente circuito convierte la señal del sensor en información que puede procesar una tarjeta de desarrollo compatible con las tarjetas UNO y posteriormente el código convierte esta información en una lectura de voltaje AC que se muestra por el monitor serie.

4.10.1 Pruebas físicas de voltaje monofásicas

Para la primera prueba de Voltaje para una carga monofásica se conectó un Voltímetro Klein Tools conectado en serie con un variador de Voltaje. Se realizaron pruebas con una carga resistiva de 1000 ohms y una carga inductiva de 1.6 H.

El sensor nos da idealmente un valor de 2.5 V cuando el voltaje en alterna es de 0 V, de modo que la señal oscila alrededor del valor 512 de la lectura analógica, la amplitud de la salida de corriente se regula mediante el potenciómetro de precisión en el módulo.

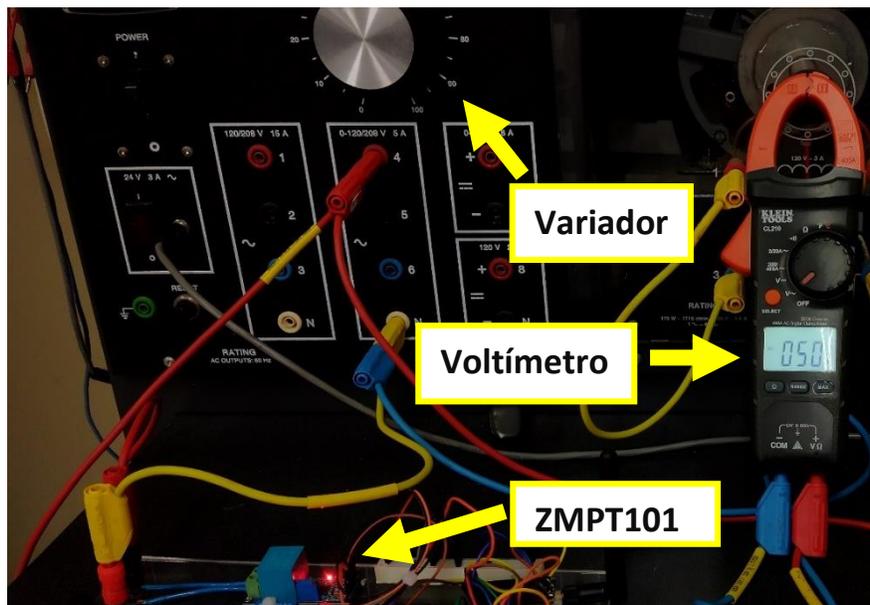


Fig. 4.3 medición de Voltaje monofásico

4.10.1.2 Variador de frecuencia al 70%

La primera prueba para la medición de Voltaje de una sola fase se coloca el variador de Voltaje a una capacidad del 70% de una línea monofásica de 127 V como se muestra en la imagen 4.5. conectado en paralelo un motor monofásico para el modelado de la carga.

El modulo está integrado por un transformador que cumple la función de aislamiento galvánico para mayor seguridad en el uso. El lado primario del transformador se conecta al voltaje alterno que deseamos medir, por ejemplo: la red eléctrica de nuestro hogar. En el lado secundario del transformador se encuentra un divisor de tensión y un circuito con amplificador operacional LM358 para adicionar un desplazamiento (offset) a la salida analógica.

Soporta voltajes de entrada de hasta 250VAC y entrega una onda senoidal de amplitud regulable por un potenciómetro en la placa. La onda senoidal de salida está desplazada positivamente para que la onda no tenga voltajes negativos y así poder leer la onda completamente con el ADC. El desplazamiento depende del voltaje con el que alimentemos el módulo: si el voltaje de alimentación es de 5V el desplazamiento será de 2.5V y si alimentamos el módulo con 3.3V el desplazamiento será de 1.65V. El circuito de acondicionamiento de señal permite que el voltaje de salida del módulo pueda ser leído por cualquier microcontrolador con entrada Análoga a Digital (ADC), de esta forma es posible leer el voltaje instantáneo y realizar cálculos de energía, como: Voltaje pico a pico (V_{pp}) y Voltaje Eficaz (V_{rms}).



Fig. 4.4 Variador de voltaje al 70%

Al colocar el variador de voltaje a un 70% con un voltaje de 127V de base se estaría suministrando solamente 86V aproximadamente, se hizo primero la medición real con un voltímetro Klein Tools para usar la medición como base para la calibración del sensor ZMPT101. En la figura 4.5 se muestra el voltaje detectado por el voltímetro.

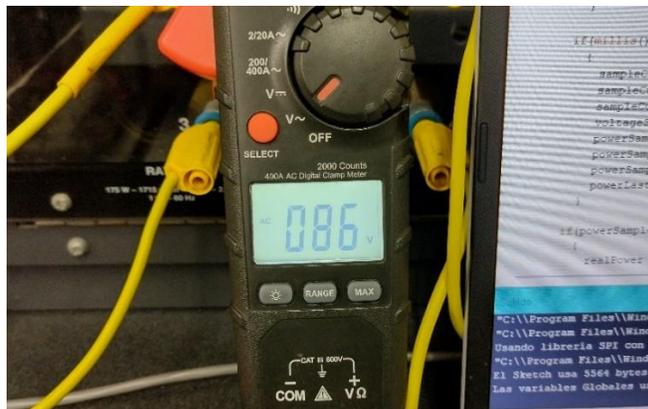


Fig. 4.5 Medición de voltaje de voltímetro Klein Tools.



Fig. 4.8 Medición de voltaje con variador al 100%

Al comparar las mediciones detectadas por el voltímetro, que se tomaron como valores reales, se realizó la calibración de sensor de voltaje ZMPT101B, los valores obtenidos de la fuente trifásica se pueden ver en la figura 4.9.

```

R_V_I_3F Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda
COM5
Corriente 1: 1.19 A
Corriente 2: 1.22 A
Corriente 3: 1.21 A
Voltaje 1 RMS: 127.68 V1
Voltaje 2 RMS: 127.65 V2
Voltaje 3 RMS: 127.61 V3

```

Fig. 4.9 Medición de voltaje Arduino-IDE con variador al 100%

4.11 Pruebas de Corriente con Arduino-IDE

Otro de los parámetros eléctricos principales para este proyecto es la medición de corriente, que para este proyecto representa el consumo eléctrico de un usuario. La primera prueba se realizó con ayuda de una carga resistiva.

4.11.1 Pruebas físicas de corriente monofásicas

Para las pruebas de corriente monofásicas se hizo la conexión con ayuda de tres cargas resistivas cada una representando un consumo eléctrico, logrando así el modelado del consumo de un usuario, como se muestra en la figura 4.10.

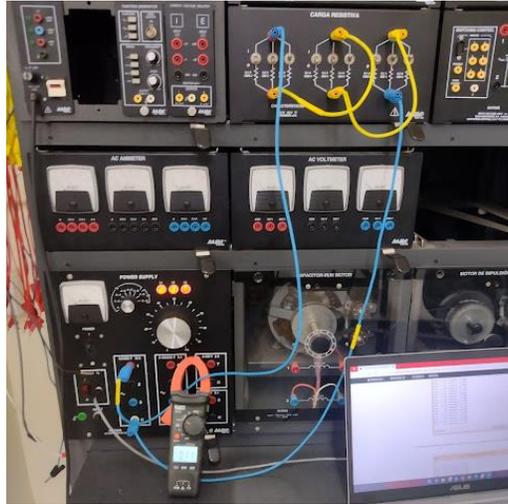


Fig. 4.10 Pruebas de corriente con cargas resistivas

A continuación, se presenta en la figura 4.11 la corriente medida con amperímetro de efecto Hall Klein Tools que dieron como resultado 0.129A.

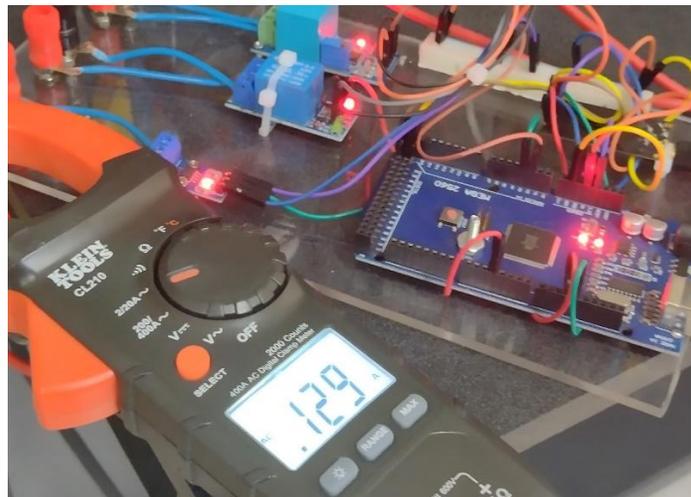


Fig. 4.11 Corriente medida por amperímetro Klein Tools

En la figura 4.12 se presenta la corriente detectada por el sensor de corriente ASC712 mediante Arduino IDE.

```
ComunicacionSensor COM5
if(powerSampleCount == 100)
{
  realPower = ((powerSampleSum / powerSampleCount) * 1000);
  Serial.print("VOLTAJE: ");
  Serial.print(VOLTAJE);
  Serial.print(" CORRIENTE: ");
  Serial.print(CORRIENTE);
  Serial.print(" POTENCIA: ");
  Serial.print(POTENCIA);
  Serial.println();
  powerSampleSum = 0;
  powerSampleCount = 0;
}
Serial.print("VOLTAJE: ");
Serial.print(VOLTAJE);
Serial.print(" CORRIENTE: ");
Serial.print(CORRIENTE);
Serial.println();
```

Fig. 4.12 Corriente medida por sensor ASC712

4.11.2 Pruebas físicas de corriente trifásicas

Para las pruebas de corriente trifásicas se aplicó una carga con ayuda de un motor de impulso o mejor conocido como dinamómetro del equipo de LabVolt representando un consumo eléctrico llevando el motor a plena carga para lograr el modelado del consumo de un usuario, como se muestra en la figura 4.13.

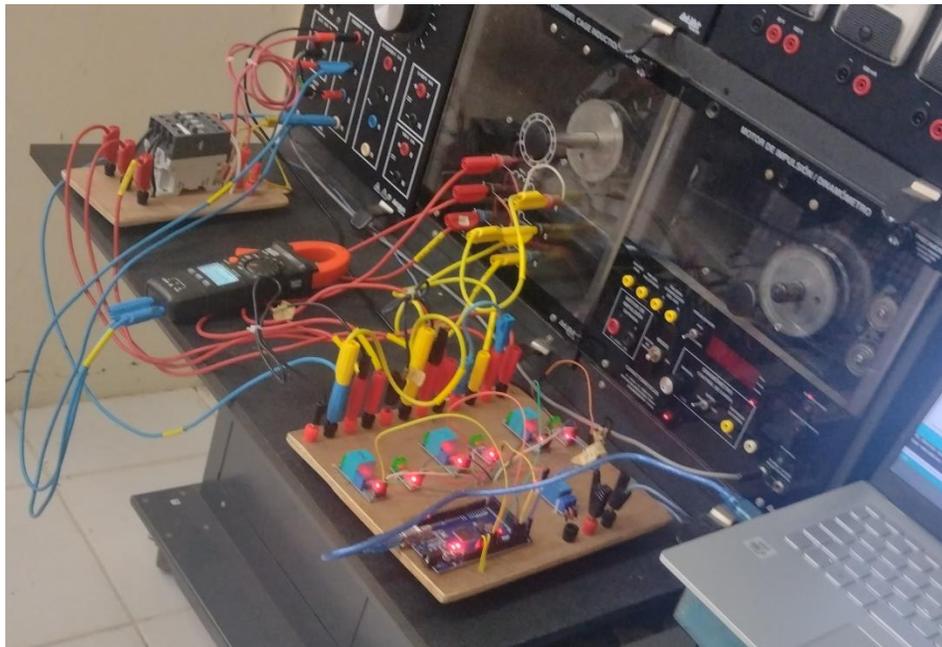


Fig. 4.13 Conexión para pruebas de corriente trifásicas

Las variables tomas como valores reales se hicieron en base a la medición de un amperímetro de gancho de efecto Hall marca Klein Tools como se muestra en la imagen 4.14.



Fig. 4.14 Corriente tomada por amperímetro de efecto Hall

Se observaron y analizaron las corrientes de las tres fases que alimentan al motor trifásico, una vez realizado el análisis de las tres fases se comparó con las mediciones detectadas por los tres amperímetros de ASC712 convertidas a valores digitales con ayuda del programa Arduino-IDE como se muestra en la imagen 4.15.

```
R_V_I_3F Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda
COM5
Corriente 1: 1.19 A
Corriente 2: 1.22 A
Corriente 3: 1.21 A
Voltaje 1 RMS: 127.68 V1
Voltaje 2 RMS: 127.65 V2
Voltaje 3 RMS: 127.61 V3
```

Fig. 4.15 Corrientes mostradas en Arduino-IDE

4.12 Pruebas del relevador de 5V con Arduino-IDE

La placa Arduino puede alimentar circuitos electrónicos como Leds, pequeños servomotores y adquirir lectura de diversos sensores. Para esta prueba se busca el control de grandes cargas eléctricas o grandes motores para apertura y cierre de grandes cargas con un Arduino.

4.12.1 Pruebas de funcionamiento

Las pruebas de funcionamiento para el circuito de control se realizaron mediante una comunicación Serial a través del puerto COM donde esté conectado el microcontrolador Arduino. Se utilizo un relevador capaz de activar su bobina con 5VCD y sus contactos sean compatibles con 11VCA. Se declara una salida digital EN EL PIN 15. El relevador se activa con un estado LOW.

Es importante desconectar el relevador antes de cargar el código al Arduino, ya que, en el proceso de carga del programa, los puertos se activan y desactivan; esto podría causar acciones no deseadas en el relevador. Se realizaron pruebas con ayuda de un motor monofásico para asegurarnos que la conversión de las señales digitales a señales analógicas sea correcta y al mismo tiempo el dispositivo nos respondas con un mensaje de Motor encendido o Motor apagado según sea el caso. En la figura 4.16 se muestra las conexiones del relevador, así como el mensaje de respuesta por parte del microcontrolador Arduino.

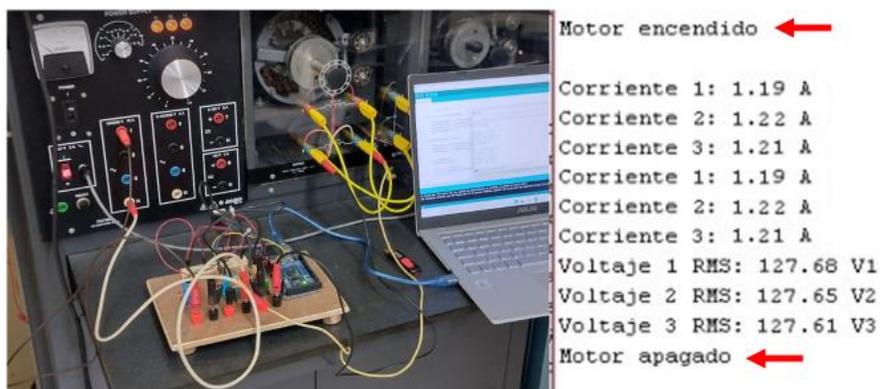


Fig. 4.16 Pruebas de funcionamiento de relevador de 5V

4.13 Pruebas de relevador de 5V con contactor trifásico mediante Arduino-IDE

Una vez realizadas las pruebas de funcionamiento del relevador de 5V se procede a agregar un circuito de fuerza con un contactor trifásico para el control de un suministro trifásico.

4.13.1 *Pruebas de funcionamiento*

Se controla un motor trifásico de potencia. El motor del equipo de LabVolt trifásico se alimenta de con tres fases de 110VCA, se utiliza un contactor de tres polos, que básicamente es un relevador que contiene una bobina auxiliar que activa tres contactos a la vez.

El contactor que se utiliza de ejemplo es de la marca ABB y su bobina auxiliar se activa con 110V. El relevador de 5V que se utilizó en la prueba anterior se utiliza para activar la bobina del contactor utilizando el mismo código de programación. Como se muestra en la figura 4.17.

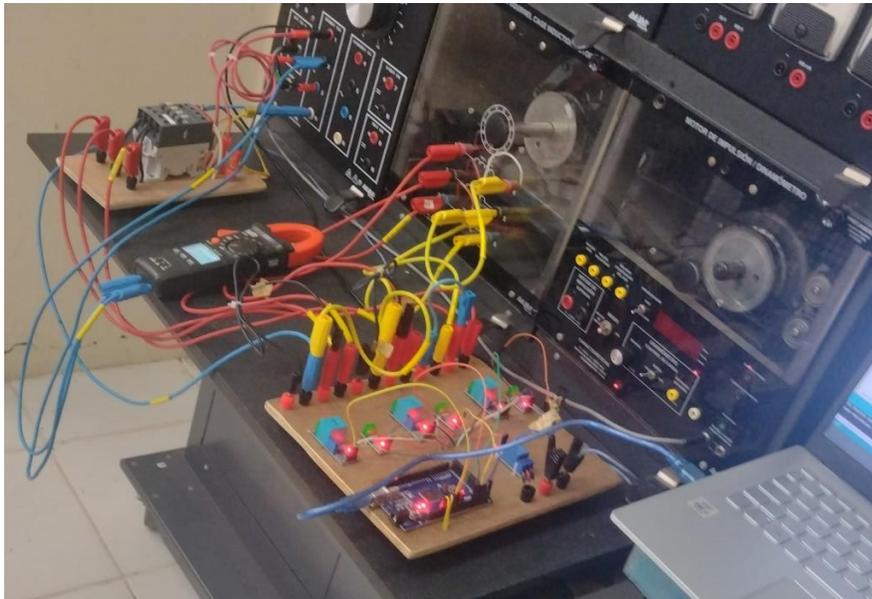


Fig. 4.17 Conexión de circuito de control con circuito de fuerza

4.14 Pruebas con radiofrecuencia utilizando Arduino-IDE/XBee

Para las pruebas de medición de parámetros eléctricos de manera inalámbrica se tomó como base la programación anterior, solo que esa vez se conectaron los dispositivos XBee a ambos microcontroladores Arduino.

4.14.1 Pruebas físicas para lectura de voltaje y corriente trifásico mediante protocolo ZigBee

Para la transferencia de datos inalámbricos se procedió a cargar la programación Arduino-IDE al primer microcontrolador para hacer la conversión de los valores analógicos a digitales, una vez cargado el programa al primer Arduino se declaran las variables en el segundo Arduino para que sea capaz de imprimir los valores digitales en el puerto serie de valores de voltaje y valores de corriente tal y como se muestra en las siguientes imágenes.

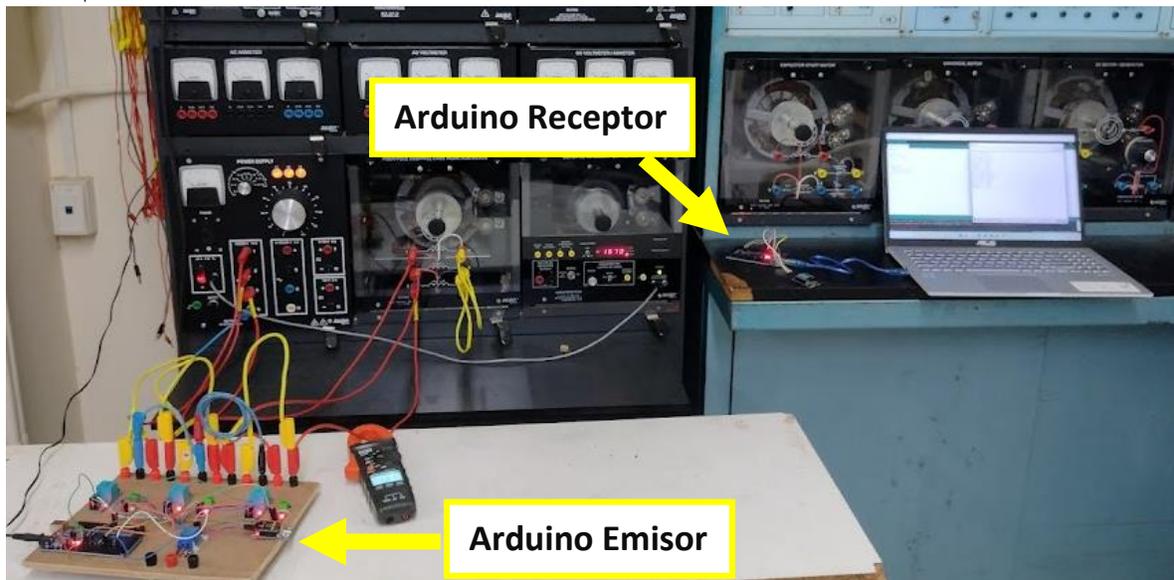


Fig. 4.18 Conexiones para lectura de parámetros eléctricos
Arduino/XBee

Las mediciones recibidas por el dispositivo receptor fueron comparadas con los valores reales detectados por el multímetro Klein Tools siendo el margen de error mínimo.

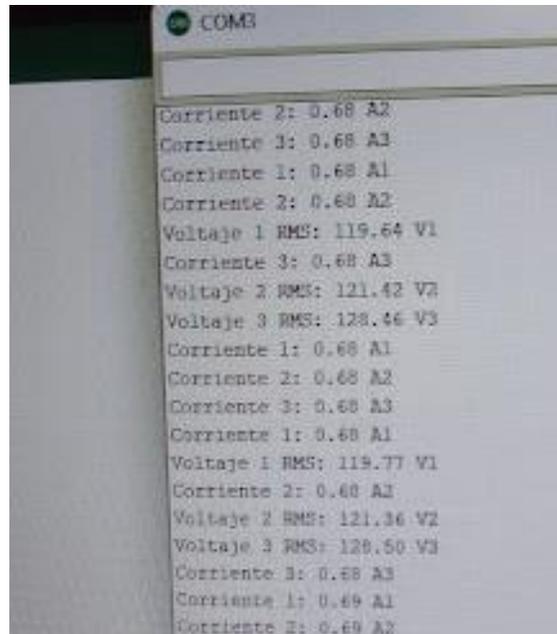


Fig. 4.19 Valores obtenidos por el dispositivo receptor de las variables

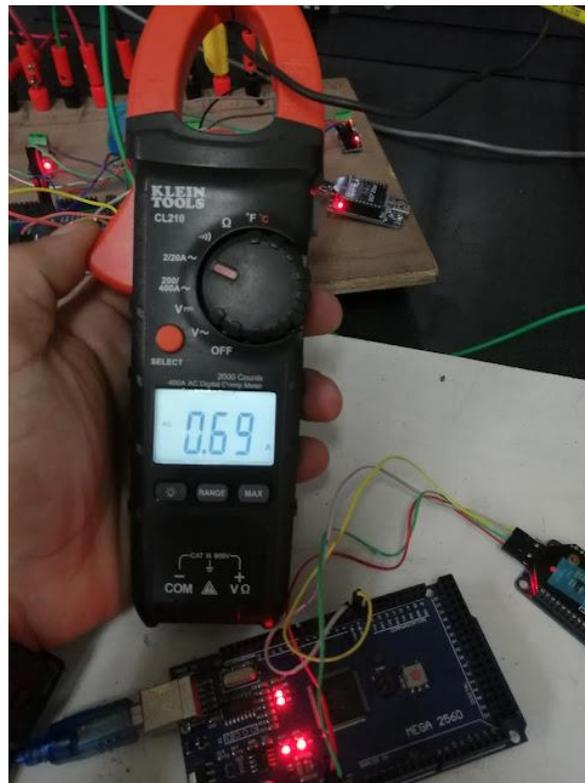


Fig. 4.20 Valores de corriente reales con multímetro Klein Tools



Fig. 4.22 Valores de Voltaje reales con multímetro Klein Tools

4.15 Pruebas de relevador de 5V con contactor trifásico mediante protocolo ZigBee utilizando Arduino-IDE/XBee

4.15.1 Pruebas de funcionamiento

Un relevador es un interruptor accionado eléctricamente. Para esta prueba de funcionamiento se utilizó un relevador con potencia necesaria para controlar directamente un motor eléctrico simulando una carga residencial. Relés de estado sólido de control son circuitos de potencia sin partes móviles, mediante la utilización de dispositivos semiconductores para realizar la conmutación.

Para esta prueba se declaró una instrucción al microcontrolador para cerrar o apertura una bobina auxiliar integrada al relevador de 5V en la cual cuando se mandaba una orden de; Encender, manda una señal analógica al dispositivo energizando la bobina auxiliar y permitiendo el paso de energía eléctrica que a su vez simula el suministro eléctrico, y este responde con una señal de “Motor encendido”.

Posteriormente se declara la variable de “Apagar” mandando una señal para negativa a la bobina auxiliar, des-energizando y abriendo el circuito eléctrico para simulación de corte de energía eléctrica, este responde con una señal de “Motor apagado”.

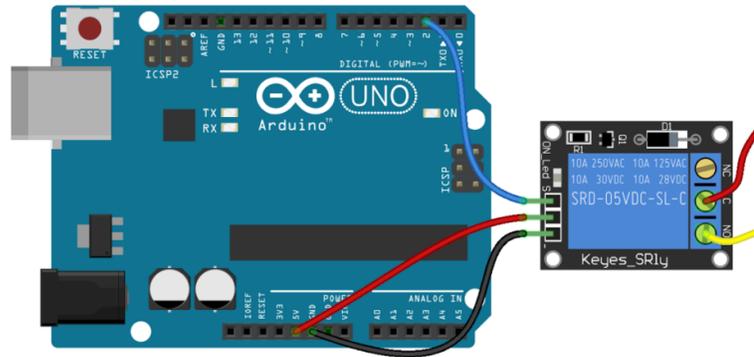


Fig. 4.23 Conexión Arduino Mega 2650 – Relevador de 5V

Estas pruebas se realizan con un motor monofásico del equipo de Lab-Volt a 127V y una serie de programación mediante la interfaz gráfica Arduino-IDE.

Una vez logrado controlar una carga monofásica con ayuda del relevador de 5V se conecta un contactor de 220V para manejar cargas más grandes y una fuente trifásica controlada con un microcontrolador de 5V.

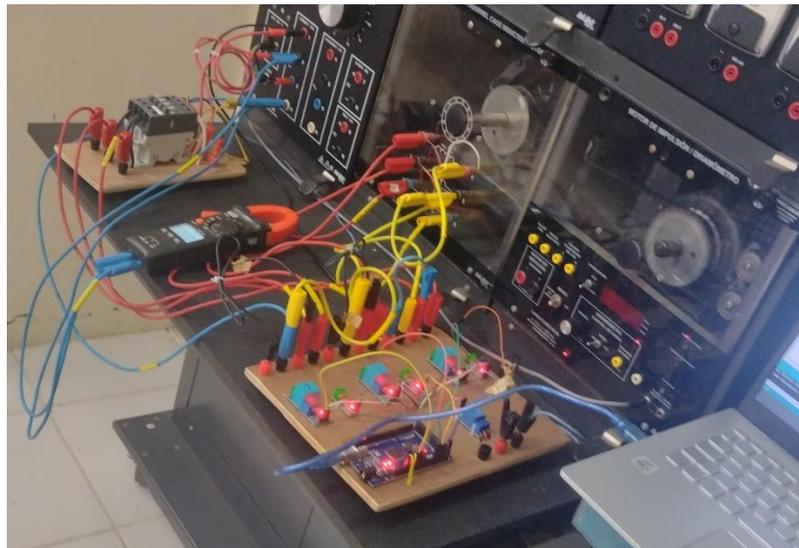


Fig. 4.24 Conexiones físicas de Relevador-Contactor y carga trifásica

Después se hicieron las respectivas conexiones de los sensores de voltaje, corriente y el relevador con la carga a medir, como se puede observar en la figura 4.18.

Esta prueba se realizó con un motor trifásico del equipo de Lab-Volt con conexión estrella, como se puede observar, el arduino solamente se encarga de activar o desactivar la bobina del relevador, esto activara o desactivara el circuito secundario o circuito de fuerza donde se encuentra el voltaje necesario para suministrar energía eléctrica a un usuario. Para el caso, el usuario podría ser sustituido por otro aparato eléctrico como una bomba de agua, una lámpara, entre muchas otras opciones.

```

R_V_L3F Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda
R_V_L3F
String entradaSerial =
bool entradaCompleta = f

/* 1 BLOQUE Gener
int decimalesPREC
/* 1.1 VOLTAJE
int VoltageAnalco
float voltageSan
float voltageLas
float voltageSan
float voltageSan
float voltageSan
float voltageMee
float RMSVoltage
float djustRMSV
float inalRMSV
  
```

```

Corriente 1: 1.19 A
Corriente 2: 1.22 A
Corriente 3: 1.21 A
Voltaje 1 RMS: 127.68 V1
Voltaje 2 RMS: 127.65 V2
Voltaje 3 RMS: 127.64 V3
Motor encendido
Corriente 1: 1.19 A
Corriente 2: 1.22 A
Corriente 3: 1.21 A
Corriente 1: 1.19 A
Corriente 2: 1.22 A
Corriente 3: 1.21 A
Voltaje 1 RMS: 127.68 V1
Voltaje 2 RMS: 127.65 V2
Voltaje 3 RMS: 127.64 V3
Motor apagado
  
```

Fig. 4.26 Respuesta de Arduino al recibir señales de cerrar o abrir bobina auxiliar.

La ventaja de utilizar el relevador es que, mediante un pequeño voltaje, suministrado por el Microcontrolador, podemos activar o desactivar un circuito de fuerza. Aquí diferenciamos dos tipos de circuitos: el circuito de control (con voltajes pequeños), y el circuito de potencia (con voltajes grandes). Esto nos permite que las corrientes eléctricas que circulan por el circuito de potencia no interfieran con las corrientes pequeñas del circuito de control. De esta manera, los circuitos se encontrar aislados eléctricamente.

Controlar algo realmente de potencia: un motor trifásico se alimenta con 220VCA, mediante tres líneas de energía. Por lo cual, se utiliza un contactor de tres polos, que es básicamente un relevador, contiene una bobina que activa tres contactos a la vez. El contactor que utilizamos su bobina se activa con 110VCA.

El diagrama de conexiones se presenta de la siguiente manera. El relevador utilizado para controlar el motor monofásico, nos ayuda para activar la bobina del contactor. Se utiliza el mismo código visto antes.

Solo que para controlar el encendido y apagado del motor utilizaremos señales como “Z” para encender e “Y” para apagar, ya que al juntar los códigos se puede percibir cierta interferencia entre la señal inalámbrica.

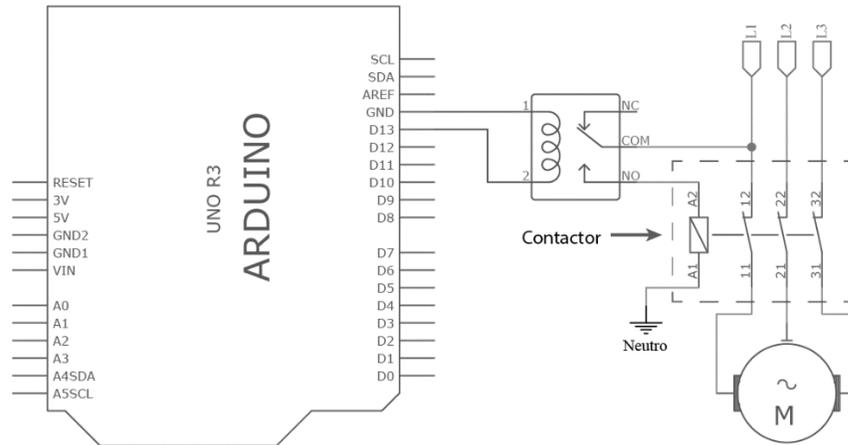


Fig. 4.27 Diagrama para el control de una carga trifásica con Arduino

CAPITULO 5

CONCLUSIONES Y TRABAJOS FUTUROS

La adquisición de los parámetros eléctricos de una carga trifásica con placa Arduino fue posible gracias a los sensores de voltaje, corriente y Arduino Mega 2560 facilitó el monitoreo de las variables.

5.1 Conclusiones.

Con este proyecto de investigación se logró desarrollar un sistema a de adquisición de datos con Arduino-XBee el cual puede realizar las funciones más significativas de sus contrapartes comerciales, esto es, la medición de voltaje, corriente control de suministro eléctrico y la creación de una interfaz gráfica moderna.

Las mediciones obtenidas en tiempo real de manera inalámbrica tienen la posibilidad de almacenarse en un archivo de texto previamente configurado de donde se pueden consultar los valores de cada una de las variables obtenida con su respectiva fecha y hora.

Los resultados obtenidos fueron satisfactorios presentando un porcentaje de error relativamente bajo con lo cual se puede decir que su aplicación cotidiana es posible pudiendo realizar mediciones de motores durante tiempos prolongados sin afectar las mediciones. Aunado a esto, el prototipo también puede funcionar como un medidor de inteligencia avanzada.

También se puede notar que las mediciones obtenidas de los sensores de 30^a tienen un porcentaje ligeramente más bajo, esto debido a que estos sensores poseen una sensibilidad más baja que sus contrapartes de 5^a y por esto son menos susceptibles al ruido que causan interferencias en las mediciones.

Un aspecto importante es la poder realizar la comparación entre uno de los métodos de análisis de motores de inducción como lo es el diagrama de circuito, con la medición obtenida de los sensores la cual puede darnos una referencia de la efectividad de los métodos que se utilizaban antes de proliferación de los instrumentos digitales.

Gracias al reducido tamaño de los sensores como del Arduino fue posible hacer de este proyecto un dispositivo portátil que puede llevarse al área donde se necesite instalar sin esfuerzo alguno.

5.2 Recomendaciones para trabajos futuros.

- Una de las recomendaciones que puede darse es recopilar información o realizar experimentos acerca del sensor de voltaje ZMPT101B con lo cual se espera obtener los valores precisos de la relación entrada y salida de las señales para poder mejorar la medición.
- Aumentar el rango de medición utilizando sensores de corriente con capacidades más elevadas y a su vez intentar realizar una medición no invasiva utilizando sensores en forma de gancho.
- Realizar modificaciones, ya sea por software o hardware, con la cual puedan mejorarse las mediciones de los sensores de 5^a y reduciendo la afectación que causa el ruido de sus mediciones.

- Realizar mediciones más seguras o implementar un sistema de seguridad para que las señales sean de un código seguro y agentes extraños no pueda tener acceso a las mediciones o puedan controlar el suministro eléctrico.
- Trabajar con una interfaz gráfica para que se puedan visualizar de una mejor manera los parámetros obtenidos por la medición.

CÓDIGO ARDUINO SENSORE EMISOR DE LAS VARIABLES

```
int dato;
```

```
#include <Wire.h>
```

```
/* %%%%%%%%%%%%%%% BLOQUE VOLTAJE */
```

```
int decimalPrecision = 2;          /*lugares decimales para todos los valores que se  
muestran en el monitor en serie*/
```

```
/* %%%%%%%%%%%%%%% 1.1 VOLTAJE 1 */
```

```
int VoltageAnalogInputPin = A0;    /*Qué pin medir el valor del voltaje*/  
float voltageSampleRead = 0;       /*Leer el valor de una muestra en analógico, incluido  
voltageOffset1*/  
float voltageLastSample = 0;       /*Contar el tiempo de cada muestra. Técnicamente se  
toma 1 milisegundo 1 muestra*/  
float voltageSampleSum = 0;        /*Acumulación de lecturas de muestra*/  
float voltageSampleCount = 0;     /*Contar el número de muestra*/  
float voltageMean ;               /*Calcular el valor promedio de todas las muestras, en  
valores analógicos*/  
float RMSVoltageMean ;           /*Calcular el valor promedio de todas las muestras, en  
valores analogicos*/  
float adjustRMSVoltageMean;  
float FinalRMSVoltage;           /*Valor de tensión final con valor de compensación*/  
  
/*VOLTAJE 1 */
```

```
float voltageOffset1 =0.00 ;    /*Para compensar la desviación y la precisión. Compensa cualquier corriente falsa cuando no funciona ninguna corriente*/
```

```
float voltageOffset2 = 0.00;    /*Valor demasiado compensado debido a un error de cálculo del cuadrado y la raíz cuadrada*/
```

```
/* %%%%%%%%%%%%%%% 1.2 - VOLTAJE 2 */
```

```
int VoltageAnalogInputPin2 = A2;    /*Qué pin medir el valor del voltaje*/
```

```
float voltageSampleRead2 = 0;    /*Leer el valor de una muestra en analógico, incluido voltageOffset1*/
```

```
float voltageLastSample2 = 0;    /*Contar el tiempo de cada muestra. Técnicamente se toma 1 milisegundo 1 muestra*/
```

```
float voltageSampleSum2 = 0;    /*Acumulación de lecturas de muestra*/
```

```
float voltageSampleCount2 = 0;    /*Contar el número de muestra*/
```

```
float voltageMean2 ;    /*Calcular el valor promedio de todas las muestras, en valores analógicos*/
```

```
float RMSVoltageMean2 ;    /*Calcular el valor promedio de todas las muestras, en valores analogicos*/
```

```
float adjustRMSVoltageMean2;
```

```
float FinalRMSVoltage2;    /*Valor de tensión final con valor de compensación*/
```

```
/*VOLTAJE 2 */
```

```
float voltageOffset3 =0.00 ;    /*Para compensar la desviación y la precisión. Compensa cualquier corriente falsa cuando no funciona ninguna corriente*/
```

```
float voltageOffset4 = 0.00;    /*Valor demasiado compensado debido a un error de cálculo del cuadrado y la raíz cuadrada*/
```

```
/* %%%%%%%%%%%%%%% 1.3 VOLTAJE 3 */
```

```
int VoltageAnalogInputPin3 = A4;    /*Qué pin medir el valor del voltaje*/
```

```
float voltageSampleRead3 = 0;    /*Leer el valor de una muestra en analógico, incluido voltageOffset1*/
```

```
float voltageLastSample3 = 0;      /*Contar el tiempo de cada muestra. Técnicamente se
toma 1 milisegundo 1 muestra*/

float voltageSampleSum3 = 0;      /*Acumulación de lecturas de muestra*/

float voltageSampleCount3 = 0;    /*Contar el número de muestra*/

float voltageMean3 ;              /*Calcular el valor promedio de todas las muestras, en
valores analógicos*/

float RMSVoltageMean3 ;          /*Calcular el valor promedio de todas las muestras, en
valores analógicos*/

float adjustRMSVoltageMean3;

float FinalRMSVoltage3;          /*Valor de tensión final con valor de compensación*/

/*VOLTAJE 3 */

float voltageOffset5 =0.00 ;      /*Para compensar la desviación y la precisión. Compensa
cualquier corriente falsa cuando no funciona ninguna corriente*/

float voltageOffset6 = 0.00;     /*Valor demasiado compensado debido a un error de
cálculo del cuadrado y la raíz cuadrada*/
```

```
/* %%% BLOQUE CORRIENTE */
```

```
/* %%% 1.1 CORRIENTE 1 */
```

```
int currentOffset1 = 0;          /* introducir valor de compensación*/

int currentOffset2 = -2;        /* introducir valor de compensación*/

int CurrentAnalogInputPin = A8; /* Qué pin medira el valor actual*/

float mVperAmpValue = 100;      /* Se utiliza el módulo de corriente ACS712: para el
módulo de 5A clave en 185, para el módulo de 20A clave en 100, para el módulo de 30A clave en
66*/
```

/ Si usa un transformador de corriente de "efecto Hall", ingrese el valor usando esta fórmula: $mVperAmp = \text{rango de voltaje máximo (en milivoltios)} / \text{corriente nominal de CT}$ */*

/ Por ejemplo, un transformador de corriente de efecto Hall de 20 A clasificado en 20 A, 2,5 V +/- 0,625 V, mV por amperio será $625 \text{ mV} / 20 \text{ A} = 31,25 \text{ mV/A}$ */*

```
float currentSampleRead = 0;      /* Para leer el valor de una muestra */
float currentLastSample = 0;      /* Para contar el tiempo de cada muestra.
Técnicamente se toma 1 milisegundo 1 muestra*/
float currentSampleSum = 0;       /* Acumulación de lecturas de muestra */
float currentSampleCount = 0;     /* Para contar el número de muestra */
float currentMean ;              /* Para calcular el valor promedio de todas las muestras*/
float RMSCurrentMean =0 ;        /* Square roof de currentMean*/
float FinalRMSCurrent ;         /* La lectura actual RMS final*/
```

/ %%% %%% %%% %%% %%% %%% %%% %%% %%% %%% 1.2 CORRIENTE 2 */*

```
int currentOffset3 = 0;          /* introducir valor de compensación*/
int currentOffset4 = -2;        /* introducir valor de compensación*/
int CurrentAnalogInputPin2 = A10; /* Qué pin medira el valor actual*/
float mVperAmpValue2 = 100;     /* Se utiliza el módulo de corriente ACS712: para el
módulo de 5A clave en 185, para el módulo de 20A clave en 100, para el módulo de 30A clave en
66*/
```

/ Si usa un transformador de corriente de "efecto Hall", ingrese el valor usando esta fórmula: $mVperAmp = \text{rango de voltaje máximo (en milivoltios)} / \text{corriente nominal de CT}$ */*

/ Por ejemplo, un transformador de corriente de efecto Hall de 20 A clasificado en 20 A, 2,5 V +/- 0,625 V, mV por amperio será $625 \text{ mV} / 20 \text{ A} = 31,25 \text{ mV/A}$ */*

```
float currentSampleRead2 = 0;    /* Para leer el valor de una muestra */
```

```

float currentLastSample2 = 0;      /* Para contar el tiempo de cada muestra.
Técnicamente se toma 1 milisegundo 1 muestra*/

float currentSampleSum2 = 0;      /* Acumulación de lecturas de muestra */

float currentSampleCount2 = 0;    /* Para contar el número de muestra */

float currentMean2 ;              /* Para calcular el valor promedio de todas las muestras*/

float RMSCurrentMean2 =0 ;        /* Square roof de currentMean*/

float FinalRMSCurrent2 ;         /* La lectura actual RMS final*/

```

```

/* %%%%%%%%%%%%%%% 1.3 CORRIENTE 3 */

```

```

int currentOffset5 = 0;           /* introducir valor de compensación*/

int currentOffset6 = -2;         /* introducir valor de compensación*/

int CurrentAnalogInputPin3 = A12; /* Qué pin medira el valor actual*/

float mVperAmpValue3 = 100;      /* Se utiliza el módulo de corriente ACS712: para el
módulo de 5A clave en 185, para el módulo de 20A clave en 100, para el módulo de 30A clave en
66*/

```

```

/* Si usa un transformador de corriente de "efecto Hall", ingrese el
valor usando esta fórmula: mVperAmp = rango de voltaje máximo (en milivoltios) / corriente
nominal de CT*/

```

```

/* Por ejemplo, un transformador de corriente de efecto Hall de 20
A clasificado en 20 A, 2,5 V +/- 0,625 V, mV por amperio será 625 mV / 20 A = 31,25 mV/A*/

```

```

float currentSampleRead3 = 0;    /* Para leer el valor de una muestra */

float currentLastSample3 = 0;    /* Para contar el tiempo de cada muestra.
Técnicamente se toma 1 milisegundo 1 muestra*/

float currentSampleSum3 = 0;    /* Acumulación de lecturas de muestra */

float currentSampleCount3 = 0;  /* Para contar el número de muestra */

float currentMean3 ;            /* Para calcular el valor promedio de todas las muestras*/

```

```

float RMSCurrentMean3 =0;      /* Square roof de currentMean*/
float FinalRMSCurrent3 ;      /* La lectura actual RMS final*/

/* %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AC Power Measurement F1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% */

int powerOffset1 =0;          // key in offset value
float sampleCurrent1 ;        /* use to calculate current*/
float sampleCurrent2 ;        /* use to calculate current*/
float sampleCurrent3 ;        /* use to calculate current*/
float apparentPower1;         /* the apparent power reading (VA) */
float realPower1 = 0;         /* the real power reading (W) */
float powerSampleRead1 = 0;   /* to read the current X voltage value of a
sample*/
float powerLastSample1 = 0;   /* to count time for each sample. Technically 1 milli
second 1 sample is taken */
float powerSampleCount1 = 0;  /* to count number of sample. */
float powerSampleSum1 = 0;    /* accumulation of sample readings */
float powerFactor1 = 0;      /* to display power factor value*/
float reactivePower1;

/* %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AC Power Measurement F2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% */

int powerOffset2 =0;          // key in offset value
float sampleCurrent4 ;        /* use to calculate current*/
float sampleCurrent5 ;        /* use to calculate current*/
float sampleCurrent6 ;        /* use to calculate current*/
float apparentPower2;         /* the apparent power reading (VA) */
float realPower2 = 0;         /* the real power reading (W) */
float powerSampleRead2 = 0;   /* to read the current X voltage value of a
sample*/

```

```
float powerLastSample2 = 0;          /* to count time for each sample. Technically 1 milli
second 1 sample is taken */
float powerSampleCount2 = 0;        /* to count number of sample. */
float powerSampleSum2 = 0;          /* accumulation of sample readings */
float powerFactor2 = 0;             /* to display power factor value*/
float reactivePower2;
```

```
/* %%%%%%%%%%%%%%% AC Power Measurement F3 %%%%%%%%%%%%%%% */
```

```
int powerOffset3 =0;                // key in offset value
float sampleCurrent7 ;              /* use to calculate current*/
float sampleCurrent8 ;              /* use to calculate current*/
float sampleCurrent9 ;              /* use to calculate current*/
float apparentPower3;               /* the apparent power reading (VA) */
float realPower3 = 0;                /* the real power reading (W) */
float powerSampleRead3 = 0;          /* to read the current X voltage value of a
sample*/
float powerLastSample3 = 0;          /* to count time for each sample. Technically 1 milli
second 1 sample is taken */
float powerSampleCount3 = 0;        /* to count number of sample. */
float powerSampleSum3 = 0;          /* accumulation of sample readings */
float powerFactor3 = 0;             /* to display power factor value*/
float reactivePower3;
```

```
/* 2 BLOQUE Monitor serie */
```

```
void setup()      /*Códigos para ejecutar una vez en monitor serie */
```

```
{  
    Serial.begin(9600);           /* Para mostrar lecturas en Serial Monitor a velocidades de  
    9600 baudios */  
    pinMode(A15,OUTPUT);  
  
    Wire.begin();  
    Wire.beginTransmission(0x68);  
    Wire.write(0x68);  
    Wire.write(0x00);  
    Wire.endTransmission(true);  
}
```

```
void loop()  
{  
    if(Serial.available(>0)  
    {  
        dato=Serial.read();  
        if(dato == 'Y')  
        {  
            digitalWrite(A15,HIGH);}  
        if(dato=='Z')  
        {  
            digitalWrite(A15,LOW);  
        }  
    }  
}
```



```

    if(FinalRMSVoltage <= 2.5)                                /* para eliminar
cualquier posible valor fantasma*/
    {FinalRMSVoltage = 0;}

    Serial.println("----VOLTAJES POR FASE----");

    Serial.print("V-1: ");

    Serial.print(FinalRMSVoltage,decimalPrecision);

    Serial.println(" V ");

    delay(500);

    voltageSampleSum =0;                                     /* Restablecer los
valores de muestra acumulados para el siguiente ciclo */

    voltageSampleCount=0;                                   /* Restablecer el
número de muestras para el siguiente ciclo*/
    }

/* %%% AC Voltage 2 Measurement %%%*/

    if(micros() >= voltageLastSample2 + 1001 )              /*cada 0,2
milisegundos tomando 1 lectura*/
    {

        voltageSampleRead2 = (analogRead(VoltageAnalogInputPin2)- 512)+ voltageOffset3;
/* lea el valor de muestra, incluido el valor de compensación*/

        voltageSampleSum2 = voltageSampleSum2 + sq(voltageSampleRead2) ;
/* acumular valores analógicos totales para cada lectura de muestra*/

        voltageSampleCount2 = voltageSampleCount2 + 1;      /*
para pasar al siguiente conteo siguiente*/

        voltageLastSample2 = micros() ;                       /*restablecer el
tiempo de nuevo para que el próximo ciclo pueda comenzar*/
    }

    if(voltageSampleCount2 == 1001)                          /* después de
4000 cuentas o 800 milisegundos (0,8 segundos), haga el cálculo y muestre el valor*/
    {

```

```

    voltageMean2 = voltageSampleSum2/voltageSampleCount2;
    /* calcular el valor promedio de todas las lecturas de muestra tomadas*/

    RMSVoltageMean2 = (sqrt(voltageMean2))*1.5;                               /*El
valor X 1.5 significa la relación hacia la amplificación del módulo*/

    adjustRMSVoltageMean2 = RMSVoltageMean2 + voltageOffset4;
    /* raíz cuadrada del valor promedio, incluido el valor de compensación*/
    /* square root of the average value*/

    FinalRMSVoltage2 = RMSVoltageMean2 + voltageOffset4;                       /*
este es el voltaje RMS final*/

    if(FinalRMSVoltage2 <= 2.5)                                               /* para eliminar
cualquier posible valor fantasma*/

        {FinalRMSVoltage2 = 0;}

    Serial.print("V-2: ");

    Serial.print(FinalRMSVoltage2,decimalPrecision);

    Serial.println(" V ");

    delay(500);

    voltageSampleSum2 =0;                                                       /* Restablecer los
valores de muestra acumulados para el siguiente ciclo */

    voltageSampleCount2=0;                                                     /* Restablecer el
número de muestras para el siguiente ciclo*/

    }

    /* %%%AC Voltage 3 Measurement %%%*/

    if(micros() >= voltageLastSample3 + 1002 )                               /*cada 0,2
milisegundos tomando 1 lectura*/

    {

        voltageSampleRead3 = (analogRead(VoltageAnalogInputPin3)- 512)+ voltageOffset3;
        /* lea el valor de muestra, incluido el valor de compensación*/

        voltageSampleSum3 = voltageSampleSum3 + sq(voltageSampleRead3) ;
        /* acumular valores analógicos totales para cada lectura de muestra*/

        voltageSampleCount3 = voltageSampleCount3 + 1;                       /*
para pasar al siguiente conteo siguiente*/

```



```

    if(millis() >= currentLastSample + 1)                /* Cada 1 milisegundo
tomando 1 lectura */
    {
        currentSampleRead = analogRead(CurrentAnalogInputPin)-512 + currentOffset1;    /*
Leer el valor de la muestra */

        currentSampleSum = currentSampleSum + sq(currentSampleRead) ;                /*
Acumular valor con lecturas de muestra anteriores*/

        currentSampleCount = currentSampleCount + 1;                                /* Para pasar al
siguiente conteo siguiente */

        currentLastSample = millis();                                                /* Para restablecer el tiempo
nuevamente para que el próximo ciclo pueda comenzar de nuevo*/
    }

    if(currentSampleCount == 1003)                                                    /* Después de 1000 cuentas
o 1000 milisegundos (1 segundo), haga el cálculo y muestre el valor*/
    {
        currentMean = currentSampleSum/currentSampleCount;                          /* Calcular el
valor promedio de todas las lecturas de muestra tomadas*/

        RMSCurrentMean = sqrt(currentMean)+currentOffset2 ;                          /* Raíz
cuadrada del valor promedio*/

        FinalRMSCurrent = (((RMSCurrentMean /1024) *5000) /mVperAmpValue);          /*
Calcula el valor RMS de la corriente final*/

        currentSampleSum =0;                                                         /* Para restablecer los valores
de muestra acumulados para el siguiente ciclo */

        currentSampleCount=0;                                                       /* para restablecer el número
de muestras para el próximo ciclo */

        Serial.println("----CORRIENTE POR FASE----");

        Serial.print("I-1:");

        Serial.print(FinalRMSCurrent,decimalPrecision);

```



```

    currentMean3 = currentSampleSum3/currentSampleCount3;           /* Calcular
el valor promedio de todas las lecturas de muestra tomadas*/

```

```

    RMSCurrentMean3 = sqrt(currentMean3)+currentOffset6 ;          /* Raíz
cuadrada del valor promedio*/

```

```

    FinalRMSCurrent3 = (((RMSCurrentMean3 /1024) *5000) /mVperAmpValue3); /*
Calcula el valor RMS de la corriente final*/

```

```

    currentSampleSum3 =0;                                           /* Para restablecer los valores
de muestra acumulados para el siguiente ciclo */

```

```

    currentSampleCount3=0;                                          /* para restablecer el número
de muestras para el próximo ciclo */

```

```

    Serial.print("I-3:");

```

```

    Serial.print(FinalRMSCurrent,decimalPrecision);

```

```

    Serial.println(" A ");

```

```

    Serial.println(" ");

```

```

    delay(500);

```

```

}

```

```

/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AC Power with Direction F1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/

```

```

    if(millis() >= powerLastSample1 + 1)                            /* every 1 milli second
taking 1 reading */

```

```

{

```

```

    sampleCurrent1 = analogRead(CurrentAnalogInputPin)-512+ currentOffset1;

```

```

    sampleCurrent2 = (sampleCurrent1/1024)*5000;

```

```

    sampleCurrent3 = sampleCurrent2/mVperAmpValue;

```

```

    voltageSampleRead = 2*(analogRead(VoltageAnalogInputPin)- 512)+ voltageOffset1 ;

```

```

    powerSampleRead1 = voltageSampleRead * sampleCurrent1 ;           /* real
power sample value */

    powerSampleSum1 = powerSampleSum1 + powerSampleRead1 ;           /*
accumulate value with older sample readings*/

    powerSampleCount1 = powerSampleCount1 + 1;                       /* to move on to
the next following count */

    powerLastSample1 = millis();                                       /* to reset the time again so
that next cycle can start again*/

}

/*POTENCIA REAL*/

if(powerSampleCount1 == 1006)                                         /* after 1000 count or
1000 milli seconds (1 second), do the calculation and display value*/
{
    realPower1 = ((powerSampleSum1/powerSampleCount1)+ powerOffset1) ; /*
calculate average value of all sample readings */

    Serial.println("----POTENCIAS FASE 1----");

    Serial.print("KW H-1: ");

    Serial.print(realPower1);

    Serial.println(" W ");

    delay(500);

/*POTENCIA APARENTE 1*/

    apparentPower1= FinalRMSCurrent*RMSVoltageMean;                   /* Apparent
power do not need to recount as RMS current and RMS voltage values available*/

    Serial.print("S-1: ");

    Serial.print(apparentPower1,decimalPrecision);

    Serial.println(" VA ");

    delay(500);

/*POTENCIA REACTIVA 1*/

```

if(apparentPower1 > realPower1){ //se aplica la formula correspondiente dependiendo que es mas grande si la potencia aparente o la potencia activa para calcular la reactiva

```
    reactivePower1= sqrt(sq(apparentPower1) - sq(realPower1));
```

```
  }
```

```
  else{
```

```
    reactivePower1= sqrt(sq(realPower1) - sq(apparentPower1));
```

```
  }
```

```
  Serial.print("Q-1: ");
```

```
  Serial.print(reactivePower1,decimalPrecision);
```

```
  Serial.println(" VAR ");
```

```
  delay(500);
```

```
  /*FACTOR DE POTENCIA 1*/
```

```
  powerFactor1 = realPower1/apparentPower1;
```

```
  if(powerFactor1 >1 || powerFactor1<0)
```

```
  {
```

```
    powerFactor1 = 0;
```

```
  }
```

```
  Serial.print("FP-1: ");
```

```
  Serial.println(powerFactor1,decimalPrecision);
```

```
  Serial.println(" ");
```

```
  delay(500);
```

```
    powerSampleSum1 =0; /* to reset accumulate sample values for the next cycle */
```

```
    powerSampleCount1=0; /* to reset number of sample for the next cycle */
```

```

}

/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AC Power with Direction F2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/

    if(millis() >= powerLastSample2 + 1)                                /* every 1 milli second
taking 1 reading */
    {
        sampleCurrent4 = analogRead(CurrentAnalogInputPin2)-512+ currentOffset3;
        sampleCurrent5 = (sampleCurrent3/1024)*5000;
        sampleCurrent6 = sampleCurrent4/mVperAmpValue2;
        voltageSampleRead2 = 2*(analogRead(VoltageAnalogInputPin2)- 512)+ voltageOffset3 ;
        powerSampleRead2 = voltageSampleRead2 * sampleCurrent2 ;          /* real
power sample value */
        powerSampleSum2 = powerSampleSum2 + powerSampleRead2 ;          /*
accumulate value with older sample readings*/
        powerSampleCount2 = powerSampleCount2 + 1;                      /* to move on to
the next following count */
        powerLastSample2 = millis();                                     /* to reset the time again so
that next cycle can start again*/
    }

/*POTENCIA REAL 2*/

    if(powerSampleCount2 == 1007)                                       /* after 1000 count or
1000 milli seconds (1 second), do the calculation and display value*/
    {
        realPower2 = ((powerSampleSum2/powerSampleCount2)+ powerOffset2) ;          /*
calculate average value of all sample readings */
        Serial.println("----POTENCIAS FASE 2----");
        Serial.print("P-2: ");
    }

```

```
Serial.print(realPower2);
```

```
Serial.println(" W ");
```

```
delay(500);
```

```
/*POTENCIA APARENTE 2*/
```

```
apparentPower2= FinalRMSCurrent2*RMSVoltageMean2;          /*Apparent  
power do not need to recount as RMS current and RMS voltage values available*/
```

```
Serial.print("S-2: ");
```

```
Serial.print(apparentPower2,decimalPrecision);
```

```
Serial.println(" VA ");
```

```
delay(500);
```

```
/*POTENCIA REACTIVA 2*/
```

```
if(apparentPower2 > realPower2){ //se aplica la formula correspondiente dependiendo  
que es mas grande si la potencia aparente o la potencia activa para calcular la reactiva
```

```
reactivePower2= sqrt(sq(apparentPower2) - sq(realPower2));
```

```
}
```

```
else{
```

```
reactivePower1= sqrt(sq(realPower2) - sq(apparentPower2));
```

```
}
```

```
Serial.print("Q-2: ");
```

```
Serial.print(reactivePower2,decimalPrecision);
```

```
Serial.println(" VAR ");
```

```
delay(500);
```

```
/*FACTOR DE POTENCIA 2*/
```

```
powerFactor2 = realPower2/apparentPower2;
```

```

if(powerFactor2 >1 || powerFactor2<0)
{
  powerFactor2 = 0;
}
Serial.print("FP-2: ");
Serial.println(powerFactor2,decimalPrecision);
Serial.println(" ");
delay(500);

    powerSampleSum2 =0;                                     /* to reset accumulate sample
values for the next cycle */
    powerSampleCount2=0;                                    /* to reset number of sample
for the next cycle */
}

/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AC Power with Direction F3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/

    if(millis() >= powerLastSample1 + 1)                   /* every 1 milli second
taking 1 reading */
    {
        sampleCurrent7 = analogRead(CurrentAnalogInputPin3)-512+ currentOffset5;
        sampleCurrent8 = (sampleCurrent5/1024)*5000;
        sampleCurrent9 = sampleCurrent6/mVperAmpValue3;
        voltageSampleRead3 = 2*(analogRead(VoltageAnalogInputPin3)- 512)+ voltageOffset5 ;
        powerSampleRead3 = voltageSampleRead * sampleCurrent3 ;           /* real
power sample value */
        powerSampleSum3 = powerSampleSum3 + powerSampleRead3 ;           /*
accumulate value with older sample readings*/

```

```

    powerSampleCount3 = powerSampleCount3 + 1;           /* to move on to
the next following count */

    powerLastSample3 = millis();                          /* to reset the time again so
that next cycle can start again*/

}

/*POTENCIA REAL 3*/

if(powerSampleCount3 == 1008)                            /* after 1000 count or
1000 milli seconds (1 second), do the calculation and display value*/
{
    realPower3 = ((powerSampleSum3/powerSampleCount3)+ powerOffset3); /*
calculate average value of all sample readings */
    Serial.println("----POTENCIAS FASE 3----");
    Serial.print("P-3: ");
    Serial.print(realPower3);
    Serial.println(" W ");
    delay(500);

    /*POTENCIA APARENTE 3*/

    apparentPower3= FinalRMSCurrent3*RMSVoltageMean3;    /*Apparent
power do not need to recount as RMS current and RMS voltage values available*/

    Serial.print("S-3: ");
    Serial.print(apparentPower3,decimalPrecision);
    Serial.println(" VA ");
    delay(500);

    /*POTENCIA REACTIVA 3*/

    if(apparentPower3 > realPower3){ //se aplica la formula correspondiente dependiendo
que es mas grande si la potencia aparente o la potencia activa para calcular la reactiva

        reactivePower3= sqrt(sq(apparentPower3) - sq(realPower3));

```

```
    }  
    else{  
        reactivePower3= sqrt(sq(realPower3) - sq(apparentPower3));  
    }  
    Serial.print("Q-3: ");  
    Serial.print(reactivePower3,decimalPrecision);  
    Serial.println(" VAR ");  
    delay(500);  
  
    /*FACTOR DE POTENCIA 3*/  
    powerFactor3 = realPower3/apparentPower3;  
    if(powerFactor3 >1 || powerFactor3<0)  
    {  
        powerFactor3 = 0;  
    }  
    Serial.print("FP-3: ");  
    Serial.println(powerFactor3,decimalPrecision);  
    Serial.println(" ");  
    delay(500);  
  
        powerSampleSum3 =0;                                /* to reset accumulate sample  
values for the next cycle */  
        powerSampleCount3=0;                                /* to reset number of sample  
for the next cycle */  
    }  
}
```

CÓDIGO ARDUINO SENSORE RECEPTOR DE LAS VARIABLES

```
//RECEPTOR

#include <Wire.h> //librería encargada de la comunicación con dispositivos i2c(xbee
//Explorer)

//#include <DallasTemperature.h> //librería encargada del sensor utilizado
#include <SoftwareSerial.h> // librería encargada de la comunicación serial
//SoftwareSerial mySerial(50, 51); // RX, TX pines de Arduino mega

char data;

String entradaSerial = ""; // a String to hold incoming data
bool entradaCompleta = false; // whether the string is complete

void setup() {

    //mySerial.begin(9600);

    Wire.begin();

    Serial.begin(9600);
    pinMode(A15, OUTPUT);

}

void loop() {
```

```
if (entradaCompleta) {  
  if (entradaSerial == "Y /n") {  
    digitalWrite(A15, LOW);  
  
  }  
  
  else if (entradaSerial == "Z /n") {  
    digitalWrite(A15, HIGH);  
  
  }  
  
  entradaSerial = "";  
  entradaCompleta = false;  
}  
if(Serial.available(>0){  
  data = Serial.read();  
  Serial.print(data);  
}  
}
```

CÓDIGO VISUAL PARA CREAR SOFTWARE

```
namespace Control_Suministro_Electrico_TEC_NM
{
    partial class Form1
    {
        /// <summary>
        /// Variable del diseñador necesaria.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Limpiar los recursos que se estén usando.
        /// </summary>
        /// <param name="disposing">true si los recursos administrados se deben desechar; false en
        caso contrario.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Código generado por el Diseñador de Windows Forms
```

```
/// <summary>
/// Método necesario para admitir el Diseñador. No se puede modificar
/// el contenido de este método con el editor de código.
/// </summary>
private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();
    this.groupBoxElec = new System.Windows.Forms.GroupBox();
    this.buttonLedOff = new System.Windows.Forms.Button();
    this.buttonLedOn = new System.Windows.Forms.Button();
    this.pictureBoxLed = new System.Windows.Forms.PictureBox();
    this.groupBoxBaudRate = new System.Windows.Forms.GroupBox();
    this.progressBarConnection = new System.Windows.Forms.ProgressBar();
    this.buttonClose = new System.Windows.Forms.Button();
    this.buttonOpen = new System.Windows.Forms.Button();
    this.comboBoxPort = new System.Windows.Forms.ComboBox();
    this.comboBoxBaudRate = new System.Windows.Forms.ComboBox();
    this.labelLbaud = new System.Windows.Forms.Label();
    this.labelPort = new System.Windows.Forms.Label();
    this.serialPort1 = new System.IO.Ports.SerialPort(this.components);
    this.pictureBox4 = new System.Windows.Forms.PictureBox();
    this.pictureBox3 = new System.Windows.Forms.PictureBox();
    this.pictureBox2 = new System.Windows.Forms.PictureBox();
    this.pictureBox1 = new System.Windows.Forms.PictureBox();
    this.contextMenuStrip1 = new
System.Windows.Forms.ContextMenuStrip(this.components);
    this.groupBoxElec.SuspendLayout();
    ((System.ComponentModel.ISupportInitialize)(this.pictureBoxLed)).BeginInit();
    this.groupBoxBaudRate.SuspendLayout();
```

```
((System.ComponentModel.ISupportInitialize)(this.pictureBox4)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
this.SuspendLayout();

//
// groupBoxElec
//
this.groupBoxElec.Controls.Add(this.buttonLedOff);
this.groupBoxElec.Controls.Add(this.buttonLedOn);
this.groupBoxElec.Controls.Add(this.pictureBoxLed);
this.groupBoxElec.Font = new System.Drawing.Font("Arial", 9.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.groupBoxElec.ForeColor = System.Drawing.SystemColors.Desktop;
this.groupBoxElec.Location = new System.Drawing.Point(405, 101);
this.groupBoxElec.Name = "groupBoxElec";
this.groupBoxElec.Size = new System.Drawing.Size(341, 284);
this.groupBoxElec.TabIndex = 0;
this.groupBoxElec.TabStop = false;
this.groupBoxElec.Text = "CONTROL DE SUMINISTRO ELÉCTRICO:";

//
// buttonLedOff
//
this.buttonLedOff.BackColor = System.Drawing.Color.Green;
this.buttonLedOff.Location = new System.Drawing.Point(231, 230);
this.buttonLedOff.Name = "buttonLedOff";
this.buttonLedOff.Size = new System.Drawing.Size(90, 35);
this.buttonLedOff.TabIndex = 2;
this.buttonLedOff.Text = "ON";
```

```
this.buttonLedOff.UseVisualStyleBackColor = false;

this.buttonLedOff.Click += new System.EventHandler(this.buttonLedOff_Click);

//

// buttonLedOn

//

this.buttonLedOn.BackColor = System.Drawing.Color.Red;

this.buttonLedOn.ForeColor = System.Drawing.SystemColors.Desktop;

this.buttonLedOn.Location = new System.Drawing.Point(27, 230);

this.buttonLedOn.Name = "buttonLedOn";

this.buttonLedOn.Size = new System.Drawing.Size(90, 35);

this.buttonLedOn.TabIndex = 1;

this.buttonLedOn.Text = "OFF";

this.buttonLedOn.UseVisualStyleBackColor = false;

this.buttonLedOn.Click += new System.EventHandler(this.buttonLedOn_Click);

//

// pictureBoxLed

//

this.pictureBoxLed.Image =
global::Control_Suministro_Electrico_TEC_NM.Properties.Resources.button_icon_png_21047;

this.pictureBoxLed.Location = new System.Drawing.Point(99, 31);

this.pictureBoxLed.Name = "pictureBoxLed";

this.pictureBoxLed.Size = new System.Drawing.Size(150, 150);

this.pictureBoxLed.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;

this.pictureBoxLed.TabIndex = 0;

this.pictureBoxLed.TabStop = false;

//

// groupBoxBaudRate

//

this.groupBoxBaudRate.Controls.Add(this.progressBarConnection);
```

```
this.groupBoxBaudRate.Controls.Add(this.buttonClose);
this.groupBoxBaudRate.Controls.Add(this.buttonOpen);
this.groupBoxBaudRate.Controls.Add(this.comboBoxPort);
this.groupBoxBaudRate.Controls.Add(this.comboBoxBaudRate);
this.groupBoxBaudRate.Controls.Add(this.labelBaud);
this.groupBoxBaudRate.Controls.Add(this.labelPort);
this.groupBoxBaudRate.Font = new System.Drawing.Font("Arial", 9.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.groupBoxBaudRate.ForeColor = System.Drawing.SystemColors.Desktop;
this.groupBoxBaudRate.Location = new System.Drawing.Point(12, 101);
this.groupBoxBaudRate.Name = "groupBoxBaudRate";
this.groupBoxBaudRate.Size = new System.Drawing.Size(372, 284);
this.groupBoxBaudRate.TabIndex = 1;
this.groupBoxBaudRate.TabStop = false;
this.groupBoxBaudRate.Text = "USB-SERIAL COM";
//
// progressBarConnection
//
this.progressBarConnection.Location = new System.Drawing.Point(32, 230);
this.progressBarConnection.Name = "progressBarConnection";
this.progressBarConnection.Size = new System.Drawing.Size(309, 35);
this.progressBarConnection.TabIndex = 2;
//
// buttonClose
//
this.buttonClose.BackColor = System.Drawing.Color.Red;
this.buttonClose.Location = new System.Drawing.Point(206, 158);
this.buttonClose.Name = "buttonClose";
this.buttonClose.Size = new System.Drawing.Size(109, 55);
```

```
this.buttonClose.TabIndex = 5;

this.buttonClose.Text = "XBee-OFF";

this.buttonClose.UseVisualStyleBackColor = false;

this.buttonClose.Click += new System.EventHandler(this.buttonClose_Click);

//

// buttonOpen

//

this.buttonOpen.BackColor = System.Drawing.Color.DarkGreen;

this.buttonOpen.Location = new System.Drawing.Point(57, 158);

this.buttonOpen.Name = "buttonOpen";

this.buttonOpen.Size = new System.Drawing.Size(109, 55);

this.buttonOpen.TabIndex = 4;

this.buttonOpen.Text = "XBee-ON";

this.buttonOpen.UseVisualStyleBackColor = false;

this.buttonOpen.Click += new System.EventHandler(this.buttonOpen_Click);

//

// comboBoxPort

//

this.comboBoxPort.FormattingEnabled = true;

this.comboBoxPort.Items.AddRange(new object[] {

"COM3",

"COM4",

"COM5",

"COM6",

"COM7",

"COM8",

"COM9"});

this.comboBoxPort.Location = new System.Drawing.Point(185, 38);

this.comboBoxPort.Name = "comboBoxPort";
```

```
this.comboBoxPort.Size = new System.Drawing.Size(121, 24);

this.comboBoxPort.TabIndex = 7;

//

// comboBoxBaudRate

//

this.comboBoxBaudRate.FormattingEnabled = true;

this.comboBoxBaudRate.Items.AddRange(new object[] {

"9600",

"38400",

"57600",

"115200"});

this.comboBoxBaudRate.Location = new System.Drawing.Point(185, 95);

this.comboBoxBaudRate.Name = "comboBoxBaudRate";

this.comboBoxBaudRate.Size = new System.Drawing.Size(121, 24);

this.comboBoxBaudRate.TabIndex = 2;

//

// labelBaud

//

this.labelBaud.AutoSize = true;

this.labelBaud.Location = new System.Drawing.Point(43, 98);

this.labelBaud.Name = "labelBaud";

this.labelBaud.Size = new System.Drawing.Size(70, 16);

this.labelBaud.TabIndex = 1;

this.labelBaud.Text = "BAUDIOS:";

//

// labelPort

//

this.labelPort.AutoSize = true;

this.labelPort.Location = new System.Drawing.Point(43, 43);
```

```
this.labelPort.Name = "labelPort";

this.labelPort.Size = new System.Drawing.Size(80, 16);

this.labelPort.TabIndex = 0;

this.labelPort.Text = "COM PORT:";

//

// pictureBox4

//

this.pictureBox4.Image =
global::Control_Suministro_Electrico_TEC_NM.Properties.Resources.Logotipo_de_la_CONACYT;

this.pictureBox4.Location = new System.Drawing.Point(626, 15);

this.pictureBox4.Name = "pictureBox4";

this.pictureBox4.Size = new System.Drawing.Size(90, 70);

this.pictureBox4.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;

this.pictureBox4.TabIndex = 5;

this.pictureBox4.TabStop = false;

//

// pictureBox3

//

this.pictureBox3.Image =
global::Control_Suministro_Electrico_TEC_NM.Properties.Resources.Educacion;

this.pictureBox3.Location = new System.Drawing.Point(145, 25);

this.pictureBox3.Name = "pictureBox3";

this.pictureBox3.Size = new System.Drawing.Size(265, 60);

this.pictureBox3.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;

this.pictureBox3.TabIndex = 4;

this.pictureBox3.TabStop = false;

//

// pictureBox2

//
```

```
this.pictureBox2.Image =
global::Control_Suministro_Electrico_TEC_NM.Properties.Resources.logo_tecnm_2018_orig;
this.pictureBox2.Location = new System.Drawing.Point(432, 5);
this.pictureBox2.Name = "pictureBox2";
this.pictureBox2.Size = new System.Drawing.Size(170, 80);
this.pictureBox2.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;
this.pictureBox2.TabIndex = 3;
this.pictureBox2.TabStop = false;
//
// pictureBox1
//
this.pictureBox1.Image =
global::Control_Suministro_Electrico_TEC_NM.Properties.Resources.logo_itcm_v2;
this.pictureBox1.Location = new System.Drawing.Point(44, 5);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(80, 80);
this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;
this.pictureBox1.TabIndex = 2;
this.pictureBox1.TabStop = false;
//
// contextMenuStrip1
//
this.contextMenuStrip1.Name = "contextMenuStrip1";
this.contextMenuStrip1.Size = new System.Drawing.Size(61, 4);
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.Color.Silver;
```

```
this.ClientSize = new System.Drawing.Size(761, 395);  
  
this.Controls.Add(this.pictureBox4);  
  
this.Controls.Add(this.pictureBox3);  
  
this.Controls.Add(this.pictureBox2);  
  
this.Controls.Add(this.pictureBox1);  
  
this.Controls.Add(this.groupBoxBaudRate);  
  
this.Controls.Add(this.groupBoxElec);  
  
this.ForeColor = System.Drawing.SystemColors.ControlText;  
  
this.Name = "Form1";  
  
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;  
  
this.Text = "CONTROL ELECTRICITY APPLICATION";  
  
this.groupBoxElec.ResumeLayout(false);  
  
((System.ComponentModel.ISupportInitialize)(this.pictureBoxLed)).EndInit();  
  
this.groupBoxBaudRate.ResumeLayout(false);  
  
this.groupBoxBaudRate.PerformLayout();  
  
((System.ComponentModel.ISupportInitialize)(this.pictureBox4)).EndInit();  
  
((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).EndInit();  
  
((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).EndInit();  
  
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();  
  
this.ResumeLayout(false);  
  
}
```

#endregion

```
private System.Windows.Forms.GroupBox groupBoxElec;  
  
private System.Windows.Forms.GroupBox groupBoxBaudRate;  
  
private System.Windows.Forms.Button buttonClose;  
  
private System.Windows.Forms.Button buttonOpen;
```

```
private System.Windows.Forms.ComboBox comboBoxPort;
private System.Windows.Forms.ComboBox comboBoxBaudRate;
private System.Windows.Forms.Label labelBaud;
private System.Windows.Forms.Label labelPort;
private System.Windows.Forms.Button buttonLedOff;
private System.Windows.Forms.Button buttonLedOn;
private System.Windows.Forms.PictureBox pictureBoxLed;
private System.Windows.Forms.ProgressBar progressBarConnection;
private System.IO.Ports.SerialPort serialPort1;
private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.PictureBox pictureBox2;
private System.Windows.Forms.PictureBox pictureBox3;
private System.Windows.Forms.PictureBox pictureBox4;
private System.Windows.Forms.ContextMenuStrip contextMenuStrip1;
}
}
```

CÓDIGO VISUAL PARA LA DEFINICION DE EVENTOS EN SOFTWARE

```
using Microsoft.Build.Tasks;
using System;
using System.IO.Ports;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net.NetworkInformation;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Control_Suministro_Electrico_TEC_NM
{
    public partial class Form1 : Form
    {
        public Form1 ()
        {
            InitializeComponent ();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            buttonOpen.Enabled = true;
            buttonClose.Enabled = false;
            buttonLedOn.Enabled = false;
            buttonLedOff.Enabled = false;
            comboBoxBaudRate.Text = "9600";
            progressBarConnection.Value = 0;
            progressBarConnection.ForeColor = Color.Red;
            pictureBoxLed.Image = Properties.Resources.button_icon_png_21047;

            string[] portList = SerialPort.GetPortNames ();
            comboBoxPort.Items.AddRange (portList);
        }

        private void buttonOpen_Click(object sender, EventArgs e)
        {
            try
            {
                serialPort1.PortName = comboBoxPort.Text;
            }
        }
    }
}
```

```

        serialPort1.BaudRate =
Convert.ToInt32 (comboBoxBaudRate.Text);
        serialPort1.Open();

        buttonOpen.Enabled = false;
        buttonClose.Enabled = true;
        buttonLedOn.Enabled = true;
        buttonLedOff.Enabled = true;
        progressBarConnection.Value =100;
        comboBoxBaudRate.Enabled = false;
        comboBoxPort.Enabled = false;
    }
    catch (Exception error)
    {
        MessageBox.Show(error.Message);
    }
}

private void buttonClose_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        try
        {
            //Primero apaga (OFF) si esta prendido (ON)
            serialPort1.WriteLine("Z");
            pictureBoxLed.Image =
Properties.Resources.button_icon_png_21047;

            serialPort1.Close();

            buttonOpen.Enabled = true;
            buttonClose.Enabled = false;
            buttonLedOn.Enabled = false;
            buttonLedOff.Enabled = false;
            progressBarConnection.Value = 0;
            comboBoxBaudRate.Enabled = true;
            comboBoxPort.Enabled = true;

        }
        catch (Exception error)
        {
            MessageBox.Show(error.Message);
        }
    }
}

private void buttonLedOn_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        try
        {
            //Cambio la imagen para prender
            pictureBoxLed.Image =
Properties.Resources.button_icon_png_21071;

            //Muestra el comando al Arduino para suministro de
servicio
            serialPort1.WriteLine("Y");

```

```

    }
    catch (Exception error)
    {
        MessageBox.Show(error.Message);
    }
}

}
private void buttonLedOff_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        try
        {
            pictureBoxLed.Image =
Properties.Resources.button_icon_png_21047;

            //Muestra el comando al arduino para el corte del
suministro
            serialPort1.WriteLine("Z");
        }
        catch (Exception error)
        {
            MessageBox.Show(error.Message);
        }
    }

}
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if (serialPort1.IsOpen)
    {
        try
        {
            //Primero corta el suministro si hay servicio
            serialPort1.WriteLine("Z");
            pictureBoxLed.Image =
Properties.Resources.button_icon_png_21047;

            serialPort1.Close();
        }
        catch (Exception error)
        {
            MessageBox.Show(error.Message);
        }
    }
}
}
}

```

BIBLIOGRAFIA

- [1] Boylestad, Robert, *Análisis introductorio de circuitos tercera edición*. Trillas 2005.
- [2] Bryan Jim, *Motor nameplate: What information it provides?* Disponible en: www.easa.com/
- [3] Chapman, Stephen, *Máquinas eléctricas tercera edición*, Mc Graw Hill 2000.
- [4] Daniel Lang, *Induction Motor circle diagram: The equivalent circuit graphical solution as function of slip*, GP Technologies Ltd, Disponible en: http://www.gptechnologies.com/wp-content/uploads/2016/04/Induction-Motor-Circle-Diagrams_GP-Technologies-White-Paper_Rev0_19-October-2016.pdf
- [5] David W. Knight, *The RMS Average*, Disponible en: http://g3ynh.info/zdocs/AC_theory/RMS.pdf
- [6] *Fluke 433/433 Three Phase Power Quality Analyzer user guide*, Disponible en: https://www.ujen.es/departamentos/ingele/sites/departamento_ingele/files/uploads/manualfluke434.pdf
- [7] <http://5nrorwxhmqjijik.leadongcdn.com/attachment/kjilKBmoioSRqlkjoipSR7ww7fgzb73m/ZMPT101B-specification.pdf>
- [8] <https://www.sparkfun.com/datasheets/BreakoutBoards/0712.pdf>
- [9] IEEE Standard 112, IEEE Standard Test Procedures for polyphase Induction Motors and Generator.
- [10] International Electrotechnical Commission, IEC 60034-1 standard Rotating electrical machines.
- [11] International Electrotechnical Commission, International standard 61000-4-30 Electromagnetic compatibility (EMC) Testing and measurement techniques – Power quality measurement methods, 2003-02.
- [12] J. Fessler, *The Fourier Discrete Transform*. Disponible en: <https://bit.ly/2JHT9KH>.
- [13] J. Manuel M. Vicencio, F. Javier A. Hernández, *Cálculo de funcionamiento de una máquina eléctrica rotatoria de inducción*. Ingeniería ITCM.
- [14] Julián Zapotitla Román, *Análisis de Fourier*, Ingeniería UNAM.
- [15] Labvolt Data acquisition and control interface description. Disponible en:

https://www.labvolt.com/solutions/6_electricity_and_new_energy/98-9063-00_data_acquisition_and_control_interface

[16] León Ledesma Carolina, et al, *Motores de corriente alterna síncronos y corriente directa*, 2015, Disponible en: <http://www.itcelaya.edu.mx/ojs/index.php/pistas/article/download/355/344>

[17] National Electrical Manufacturer Association Standard, NEMA standards publication MG 1-2019 Motors and Generators.

[18] R.W. Erickson, *Power and RMS Values of Fourier Series*, Disponible en: <http://www.g3ynh.info/>

[19] Sompop Poomjan, et al, *Proof of using Fourier coefficients for root mean square calculation on periodic signals*, Advanced Studies in Theoretical Physics, Vol 8, 2014, No. 1, 21-25.