



TECNOLÓGICO NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO DE CIUDAD MADERO
DIVISION DE ESTUDIOS DE POSGRADO E INVESTIGACION
MAESTRIA EN INGENIERIA ELÉCTRICA

TESIS

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA
DE ADQUISICIÓN DE PARAMETROS DE UN
MOTOR DE INDUCCIÓN”**

**Que para obtener el grado de
Maestro en Ingeniería Eléctrica**

**Presenta:
Rey David Reyes Sosa
G08071310**

**Director de Tesis:
M.C. Rafael Castillo Gutiérrez**

Cd. Madero, Tamps., 08 de Junio de 2018

OFICIO No.: U2.059/18

ÁREA: DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

C. ING. REY DAVID REYES SOSA
NO. DE CONTROL G08071310
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, el cual está integrado por los siguientes catedráticos:

PRESIDENTE :	M.C. RAFAEL CASTILLO GUTIÉRREZ
SECRETARIO :	M.C. HERMENEGILDO CISNEROS VILLEGAS
VOCAL :	DR. PEDRO MARTÍN GARCÍA VITE
SUPLENTE :	M.C. AARÓN GONZÁLEZ RODRÍGUEZ
DIRECTOR DE TESIS :	M.C. RAFAEL CASTILLO GUTIÉRREZ

Se acordó autorizar la impresión de su tesis titulada:

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISIÓN DE PARÁMETROS DE UN MOTOR DE INDUCCIÓN”

Es muy satisfactorio para esta División compartir con Usted el logro de esta meta, esperando 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™



DR. JOSÉ AARÓN MELO BANDA
ENCARGADO DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO
DIVISIÓN DE ESTUDIOS DE POSGRADO
E INVESTIGACIÓN

c.c.p.- Archivo
Minuta

JAMB 'RCG 'mdcoa*



Contenido

	<i>Página</i>
Lista de Figuras	v
Lista de Tablas	ix
Agradecimientos	x
Dedicatorias	xi
Resumen	xii
Abstract	xiii
Capítulo 1: Introducción	1
1.1. ANTECEDENTES DEL PROBLEMA	2
1.2. PLANTEAMIENTO DEL PROBLEMA	2
1.3. JUSTIFICACIÓN	3
1.4. OBJETIVOS	4
1.5. HIPOTESIS	4
1.6. ALCANCES	5
1.7. LIMITACIONES	5
Capítulo 2: MATERIALES UTILIZADOS	
2.1. ARDUINO MEGA 2560	8
2.1.1. Alimentación	9
2.1.2. Memoria	10
2.1.3. Entradas y Salidas	10
2.1.4. Comunicaciones	12
2.1.5. Programación	12
2.1.6. Reinicio Automático por Software	13
2.1.7. Protección Contra Sobrecorrientes en USB	14
2.1.8. Características físicas y compatibilidad de shields	14

	<i>Página</i>
2.2. MOTOR DE CORRIENTE ALTERNA	14
2.2.1. Motores de rotor de jaula de ardilla	18
2.2.2. Conexión de los bobinados de un motor eléctrico trifásico	19
2.2.3. Sistema de arranque directo de los motores trifásicos	20
2.3. POTENCIA EN SISTEMAS TRIFÁSICOS	22
2.3.1. Conexión en estrella	23
2.4. ANÁLISIS DE UN MOTOR TRIFÁSICO POR MEDIO DE SU PLACA DE CARACTERÍSTICAS	24
2.5. EFECTO HALL	27
2.5.1. Sensores de efecto Hall con salida lineal	28
2.5.2. Sensor Hall de salida digital	30
2.5.3. Sensor Hall ACS712	31
2.6. SENSOR DE VOLTAJE ZMPT101B	35
2.7. SENSOR INFRARROJO DE TEMPERATURA	36
2.7.1. Funcionamiento	38
2.7.2. El bus i2C	40
2.7.2.1. Funcionamiento	42
2.7.2.2. Ventajas y desventajas del protocolo i2C	43
2.8. SERIE DE FOURIER EN FORMA COMPLEJA	44
2.8.1. Desarrollo en serie de Fourier de $f(t)$ en forma compleja	46
2.8.2. Transformada Discreta de Fourier	48
2.8.3. Uso de los coeficientes de Fourier para calcular el valor RMS de una señal periódica	49
2.8.3.1. Cálculo de valores RMS	50
2.8.3.2. Cálculo de los valores RMS de una señal periódica construida por series de Fourier	50

Capítulo 3: DISEÑO E IMPLEMENTACIÓN

3.1.	CONSTRUCCIÓN DEL PROTOTIPO	54
3.2.	PROGRAMACIÓN DEL ARDUINO	55
3.2.1.	Medición de Temperatura	57
3.2.2.	Funcionamiento del algoritmo basado en la T.D.F.	57
3.2.3.	Transmisión de datos obtenidos a la plataforma MATLAB 2015® .	60

Capítulo 4: PRUEBAS Y RESULTADOS

4.1.	PRUEBA CON MOTOR EN VACIO	63
4.2.	PRUEBA CON MOTOR CON CARGA	68
4.3	PRUEBA DE ERROR DE CONEXIÓN	74
4.4	CONEXIÓN A MATLAB® 2015 ^a	79

Capítulo 5: CONCLUSIÓN Y TRABAJOS FUTUROS

5.1.	CONCLUSIONES	89
5.2.	RECOMENDACIONES PARA TRABAJOS FUTUROS	90

Referencias	91
--------------------	----

Apéndice A: Programa del prototipo	93
---	----

Apéndice B: Código de MATLAB	108
-------------------------------------	-----

Lista de Figuras

	<i>Página</i>
Figura 2.1 Estructura de la Placa Arduino Mega 2560	9
Figura 2.2 Estructura de un motor asíncrono de inducción	15
Figura 2.3 Tipos de motores asíncronos basados en el devanado del estator	17
Figura 2.4 Rotor de jaula de ardilla	18
Figura 2.5 Estructura de un Motor con Rotor de Jaula de ardilla	18
Figura 2.6 Conexión estrella y delta respectivamente	19
Figura 2.7 Conexión para el cambio de giro de un motor	20
Figura 2.8 a) Circuito de fuerza arranque directo. b) Curva Corriente/Velocidad arranque directo. c) Curva Par/Velocidad arranque directo.....	22
Figura 2.9 Conexión en estrella de un sistema trifásico	23
Figura 2.10 Ejemplo de placa de características de un Motor	24
Figura 2.11 Triangulo de potencias	26
Figura 2.12 Diagrama del funcionamiento del efecto Hall	27
Figura 2.13 Estructura simplificada de un sensor Hall de salida lineal	28
Figura 2.14 Características de transferencia de un sensor Hall lineal	29
Figura 2.15 Estructura de un sensor de efecto Hall de salida digital	30
Figura 2.16 Placa con sensor ACS712 para Arduino	32
Figura 2.17 Equivalencia del voltaje de salida del sensor ACS712.....	34
Figura 2.18 Diagrama de conexión del sensor ACS712 a una carga	34
Figura 2.19 Sensor de Voltaje ZMPT101B	35
Figura 2.20 Diagrama de conexión del sensor de voltaje	36
Figura 2.21 Sensor infrarrojo de temperatura MLX90614	37
Figura 2.22 Espectro de objetos negros	38
Figura 2.23 Composición del MLX90614	39

	<i>Página</i>
Figura 2.24	Gráfica de error de medición del MLX90614 40
Figura 2.25	Diagrama de conexión del protocolo I2C 41
Figura 2.26	Funcionamiento del protocolo I2C 43
Figura 3.1	Diagrama de conexión del prototipo 56
Figura 3.2	Diagrama de flujo de funcionamiento del prototipo 59
Figura 3.3	Modelo en Simulink para la comunicación entre Arduino y MATLAB 60
Figura 3.4	Bloque submódulo del programa en Simulink 61
Figura 4.1	Conexión del prototipo y el analizador de redes FLUKE para pruebas con el motor..... 63
Figura 4.2	Resultados de las mediciones de voltaje y corriente tomados por el analizador de redes FLUKE (en vacío) 64
Figura 4.3	Resultados de las mediciones de voltaje y corriente tomados por el Prototipo (en vacío) 64
Figura 4.4	Resultados de las mediciones de potencias y factor de potencia tomados por el analizador de redes FLUKE (en vacío) 65
Figura 4.5	Resultados de las mediciones de potencia activa tomados por el prototipo (en vacío) 65
Figura 4.6	Resultados de las mediciones de potencia reactiva tomados por el prototipo (en vacío) 66
Figura 4.7	Resultados de las mediciones de potencia aparente tomados por el prototipo (en vacío) 66
Figura 4.8	Resultados de las mediciones de Factor de Potencia tomados por el prototipo (en vacío) 66
Figura 4.9	Resultados de las mediciones de ángulo de fase tomados por el analizador de redes FLUKE (en vacío) 67

	<i>Página</i>
Figura 4.10	Resultados de las mediciones de ángulo de fase tomados por el prototipo (en vacío) 67
Figura 4.11	Resultados de las mediciones de voltaje y corriente tomados por el analizador de redes FLUKE (con carga)..... 68
Figura 4.12	Resultados de las mediciones de voltaje y corriente tomados por el prototipo (con carga) 68
Figura 4.13	Resultados de las mediciones de potencias y factor de potencia tomados por el analizador de redes FLUKE (con carga) 69
Figura 4.14	Resultados de las mediciones de potencia activa tomados por el prototipo (con carga) 69
Figura 4.15	Resultados de las mediciones de potencia reactiva tomados por el prototipo (con carga) 70
Figura 4.16	Resultados de las mediciones de potencia activa tomados por el prototipo (con carga) 70
Figura 4.17	Resultados de las mediciones del factor de potencia tomados por el prototipo (con carga) 70
Figura 4.18	Resultados de las mediciones de ángulo de fase tomados por el analizador de redes FLUKE (con carga) 71
Figura 4.19	Resultados de las mediciones de ángulo de fase tomados por el prototipo (con carga) 71
Figura 4.20	Resultados de las mediciones de voltaje y corriente tomados por el analizador de redes FLUKE (conexión errónea) 75
Figura 4.21	Resultados de las mediciones de voltajes y corriente tomados por el prototipo (conexión errónea) 75
Figura 4.22	Resultados de las mediciones de potencias y factor de potencia tomados por el analizador de redes FLUKE (conexión errónea) 76

	<i>Página</i>
Figura 4.23 Resultados de las mediciones de potencia activa tomados por el prototipo (conexión errónea)	76
Figura 4.24 Resultados de las mediciones de potencia reactiva tomados por el prototipo (conexión errónea)	77
Figura 4.25 Resultados de las mediciones de potencia aparente tomados por el prototipo (conexión errónea)	77
Figura 4.26 Resultados de las mediciones del factor de potencia tomados por el prototipo (conexión errónea)	77
Figura 4.27 Resultados de las mediciones de ángulo de fase tomados por el analizador de redes FLUKE (conexión errónea)	78
Figura 4.28 Resultados de las mediciones de ángulo de fase tomados por el prototipo (conexión errónea)	78
Figura 4.29 Gráficas de voltaje por fase prueba 1	79
Figura 4.30 Gráficas de corriente por fase prueba 1	80
Figura 4.31 Gráficas de potencia activa por fase prueba 1	80
Figura 4.32 Gráficas de potencia reactiva por fase prueba 1	81
Figura 4.33 Gráficas de potencia aparente por fase prueba 1	81
Figura 4.34 Gráficas de Factor de potencia por fase prueba 1	82
Figura 4.35 Gráfica de temperatura del rotor prueba 1	82
Figura 4.36 Gráficas de voltaje por fase prueba 2	83
Figura 4.37 Gráficas de corriente por fase prueba 2	84
Figura 4.38 Gráficas de potencia activa por fase prueba 2	84
Figura 4.39 Gráficas de potencia reactiva por fase prueba 2	85
Figura 4.40 Gráficas de potencia aparente por fase prueba 2	85
Figura 4.41 Gráficas de factor de potencia por fase prueba 2	86
Figura 4.42 Gráfica de temperatura del rotor prueba 2	86

Lista de Tablas

		<i>Página</i>
Tabla 2.1	Aplicaciones de los sensores de efecto Hall	31
Tabla 2.1	Tabla de valores de sensibilidad de los sensores ACS712	33
Tabla 2.2	Entradas para la comunicación I2C por arduino	44
Tabla 4.1	Error entre mediciones del analizador de redes y el prototipo (en vacío)	72
Tabla 4.2	Error entre mediciones del analizador de redes y el prototipo (con carga)	73

AGRADECIMIENTOS

A la División de Estudios de Posgrado e Investigación (DEPI) del Instituto Tecnológico de Ciudad Madero, por otorgarme la oportunidad de terminar mi Maestría.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por su apoyo a través de la beca otorgada con número 737859.

A todo el cuerpo académico de la Maestría en Ingeniería Eléctrica, por brindarme apoyo en momentos difíciles.

Un agradecimiento especial al M.C. Joel Rodríguez Guillen por apreciada tutela y asistencia en el tema sobre la Transformada Discreta de Fourier.

Mi sincero y sentido agradecimiento a mi asesor, el M.C. Rafael Castillo Gutiérrez, por su apoyo para la realización de esta Tesis.

DEDICATORIA

A mi madre Rosa María Sosa Del Ángel que siempre me ha apoyado incondicionalmente durante toda mi vida escolar y personal.

A mi padre Jorge Serapio Reyes González q.e.p.d. que siempre estuvo orgulloso de mí.

A mis hermanos Jorge Alberto Reyes Sosa y Sandy Lizeth Reyes Sosa por su compañía y sus consejos en momentos difíciles.

A mis maestros: M.C. Aarón González R., el Dr. Rubén Salas C., la M. L. Gladis M. Galiana B., al M.C Rafael Castillo G., el M.C. Hermenegildo Cisneros V., el M.C. Eduardo Nacú Salas C., El Dr. Gasatón Hernández M., al M.C. José de Jesús Durón M., así como al Dr. Pedro M. García Vite, por sus enseñanzas y tutela que me brindaron durante mi estancia en la maestría.

A mis amigos: Joel Rodríguez, Arturo Álvarez, Lizbeth Laurens, José Inés, Anselmo Martínez, Edson Mares, Antonio Murillo, Eduardo Delgado, Cristian Abel, Eliel Wong, Rosario Rivera, Alfredo Alejandro, Ricardo Turrbiates, Luis Ong y muchos más por su amistad, consejos y buenos momentos que pasamos juntos.

Diseño e Implementación de un Sistema de Adquisición de Parámetros de un Motor De Inducción

Presenta: Ing. Rey David Reyes Sosa

Director de Tesis: Rafael Castillo Gutiérrez

Programa: Maestría en Ingeniería Eléctrica

Resumen

El siguiente trabajo propone el diseño y construcción de un sistema para adquirir los parámetros de las fases de un motor de inducción (corriente, voltaje, potencia activa, potencia reactiva, factor de potencia y ángulo de fase). Por medio de sensores de corriente de efecto Hall y sensores de voltaje, se obtiene las corrientes y voltajes de cada fase de un motor de inducción de cuatro polos rotor jaula de ardilla, de la marca LabVolt el cual tiene como corriente nominal 1.5 A RMS y un voltaje nominal de 120 V RMS por fase. Utilizando un algoritmo basado en la Transformada Discreta de Fourier programado en un Arduino Mega 2560, el cual es una plataforma de Hardware de desarrollo libre, se obtienen los valores RMS de Voltaje, Corriente, Potencia Aparente, Potencia Activa y Potencia Reactiva por fase, así como el Factor de Potencia y el Ángulo de fase. Un sensor de temperatura infrarrojo es utilizado para medir la temperatura del rotor. Todos estos datos son desplegados en una pantalla LCD para el correcto monitoreo de la condición operativa del motor al igual que crea gráficas del comportamiento de éstas variables por medio de comunicación en serie con MATLAB®.

Design and Implementation of a Parameter Acquisition System for an Induction Motor

Rey David Reyes Sosa

Abstract

The following paper proposes the design and construction of a system to acquire the parameters of the phases of an induction motor (current, voltage, active power, reactive power, power factor and phase angle). By means of Hall effect current sensors and voltage sensors, currents and voltages for each phase are obtained for a LabVolt four-pole squirrel cage induction motor, which has 1.5 A RMS nominal current and 120 V RMS nominal voltage per phase. Using an algorithm based on the Discrete Fourier transform programmed into an Arduino Mega 2560, which is a free development hardware platform, the RMS values of Voltage, Current, Apparent Power, Active Power and Reactive Power per phase are obtained as well as the Power Factor. An infrared temperature sensor is used to measure the rotor temperature. All this data is displayed on an LCD screen for the correct monitoring of the motor operating status as well as creating graphs of the behavior of these variables through serial communication with MATLAB®.

Capítulo

1

INTRODUCCIÓN

En este capítulo se presenta una introducción sobre el trabajo propuesto, los antecedentes, el planteamiento del problema, su justificación, así como los objetivos, la hipótesis, sus alcances y sus limitaciones.

1.1. ANTECEDENTES DEL PROBLEMA.

En la industria de la generación de la energía eléctrica, existen dos clases diferentes de generación de energía, estas son de corriente directa y de corriente alterna. El motor de inducción cumple un papel importante ya que puede ser utilizado como generador de corriente alterna.

El motor de inducción fue diseñado por Nikola Tesla, en 1882 con base en el descubrimiento del campo magnético rotatorio, un principio fundamental en la física y uno de los mayores descubrimientos en la historia. Este motor es utilizado en plantas de generación de energía eléctrica, la cual se busca que satisfaga ciertos estándares de calidad. Para esto se ha de mantener y asegurar la correcta operación del motor de inducción, lo cual se puede alcanzar por medio del monitoreo de las condiciones operativas del motor, lo cual involucra la medición de ciertos parámetros. [1]

La determinación de los parámetros del motor de inducción trifásico puede llevarse a cabo mediante mediciones hechas a través de ciertas configuraciones de conexión de los sensores [2]. La conexión de estos se realiza frecuentemente por medio de instrumentos para acondicionar las señales de tal forma que las magnitudes a sensar estén dentro del rango de medición del instrumento. Los cálculos requeridos para determinar los parámetros se pueden realizar de forma manual o utilizando un sistema digital.[3]

1.2. PLANTEAMIENTO DEL PROBLEMA.

Durante los últimos años el motor de inducción ha sido la máquina rotativa que más se ha empleado a nivel industrial, debido principalmente a su versatilidad. El desarrollo en la electrónica de potencia y la disminución en los costos de los componentes han hecho que dichos motores sean los más utilizados para cualquier condición de operación. Su sencillez, confiabilidad y eficiencia los han hecho de estos los preferidos para cualquier trabajo.

Es por esto que se requiere mantener un constante monitoreo de las condiciones de operación de un motor de inducción trifásico así como obtener mediciones de la energía eléctrica producida por éste, esto con el fin de establecer condiciones seguras de operación y de esta manera evitar que se agraven las condiciones de falla de existir las mismas. [4]

Con respecto a la calidad de la energía existen estándares que regulan las características del voltaje de generación las cuales deben satisfacerse con el fin de mantener un servicio de calidad. Para todo esto es necesario conocer el comportamiento de los parámetros del motor de inducción trifásico lo cual se logra mediante un análisis constante de estos. [5]

1.3. JUSTIFICACIÓN.

Dado la importancia del motor de inducción en las redes industriales por las pérdidas que conlleva tener parada una planta, es de vital importancia tenerlo monitoreado para evitar su daño total previniéndolo con mantenimientos preventivos.

Se pretende diseñar un prototipo portátil de adquisición de datos del motor trifásico de inducción, más específicamente la adquisición de datos de cada una de las fases (Voltaje, Corriente, Ángulo de desfasamiento, Potencia activa, Potencia reactiva, Potencia aparente y Factor de potencia), así como la temperatura del devanado de este, por medio de un sistema Arduino Mega 2560 y sensores de voltaje, corriente y temperatura.

Al ser este un sistema portátil se puede utilizar directamente en el área de maquinaria sin necesidad de acudir al cuarto de control para obtener los datos necesarios sobre un motor de inducción.

1.4. OBJETIVOS.

1.4.1. Objetivo general.

- Diseñar e implementar un prototipo de un circuito portátil basado en el microprocesador ArduinoMega2560 para medir los parámetros de cada fase de un motor de inducción al igual que su temperatura interna y desplegarlos en un LCD.

1.4.2. Objetivos específicos.

- Aprender a Programar el ArduinoMega2560.
- Llevar a cabo prácticas con el ArduinoMega2560.
- Diseñar y construir el circuito para la medición y monitoreo de las fases y temperatura del motor de inducción de Lab-Volt 8221.
- Obtener mediciones del motor de inducción y desplegarlas en un LCD.

1.5. HIPÓTESIS.

Es posible prevenir el daño o pérdida total de un motor de inducción trifásico mediante un sistema efectivo de monitoreo de variables.

El monitoreo de variables puede realizarse de tal forma que se obtengan los datos de cada una de las fases y la temperatura interna de un motor de inducción.

Es posible monitorear las variables por medio de un sistema de procesamiento digital.

El procesamiento digital se puede llevar a cabo por medio de un sistema Arduino Mega2560.

El monitoreo de las variables es posible mediante la implementación de sensores de voltaje, corriente y temperatura.

Es posible obtener por medio de programación interna la potencia activa, reactiva, aparente y factor potencia de cada fase a partir de las mediciones obtenidas en las fases del motor. [6]

Es posible desplegar los parámetros medidos y calculados en una pantalla LCD.

1.6. ALCANCES.

Al ser este un dispositivo portátil se podrá utilizar en motores de inducción en cualquier momento para un rápido análisis de sus fases y así poder monitorear el comportamiento de las mismas.

El dispositivo portátil puede modificarse para ser una herramienta útil y eficaz para la industria.

El diseño puede ser modificado con componentes más robustos para su implementación en diferentes tipos de motores de inducción.

1.7. LIMITACIONES.

Este proyecto solo se encargará de arrancar, parar y obtener las mediciones de las fases y temperatura del devanado del motor de inducción de Lab-Volt 8221 con un voltaje de línea de 120 V y una Corriente de línea de 1.5 A máximos.

Se limitará a medir por instrumentación voltaje, corriente y temperatura del devanado del motor, y se obtendrá por programación la Potencia Activa, Reactiva y Aparente así como su factor de potencia.

Capítulo

2

MATERIALES UTILIZADOS

En este capítulo se describen los materiales utilizados en la creación del prototipo así como la teoría en la que se basa el programa que se utiliza.

2.1. ARDUINO MEGA 2560. [6]

Arduino Mega 2560 está basado en el microcontrolador Atmega2560. Se dispone de 54 entradas/salidas digitales, 14 de las cuales se pueden utilizar como salidas PWM (modulación de ancho de pulso). Además dispone de 16 entradas analógicas, 4 UARTs (puertas series), un oscilador de 16MHz, una conexión USB, un conector de alimentación, un conector ICSP (In Circuit Serial Programming) y un pulsador para el reset. Para empezar a utilizar la placa sólo es necesario conectarla al ordenador a través de un cable USB, o alimentarla con un adaptador de corriente AC/DC. También puede alimentarse mediante una batería si se es requerido.

Debido a su gran poder es utilizada para grandes proyectos, entre los más importantes se encuentran los de DOMOTICA e IMPRESORAS 3D. El Arduino MEGA2560 es compatible con la mayoría de los shield o tarjetas de aplicación/ampliación disponibles para las tarjetas Arduino UNO original. Las características principales son:

- Microprocesador ATmega2560
- Voltaje de alimentación (recomendado) 7-12V
- Integra regulación y estabilización de +5Vcc
- 54 líneas de Entradas/Salidas Digitales (14 de ellas se pueden utilizar como salidas PWM)
- 16 Entradas Analógicas
- Máxima corriente continua para las entradas: 40 mA
- Salida de alimentación a 3.3V con 50 mA
- Memoria de programa de 256Kb (el bootloader ocupa 8Kb)
- Memoria SRAM de 8Kb para datos y variables del programa
- Memoria EEPROM para datos y variables no volátiles
- Velocidad del reloj de trabajo de 16MHz
- Reducidas dimensiones de 100 x 50 mm

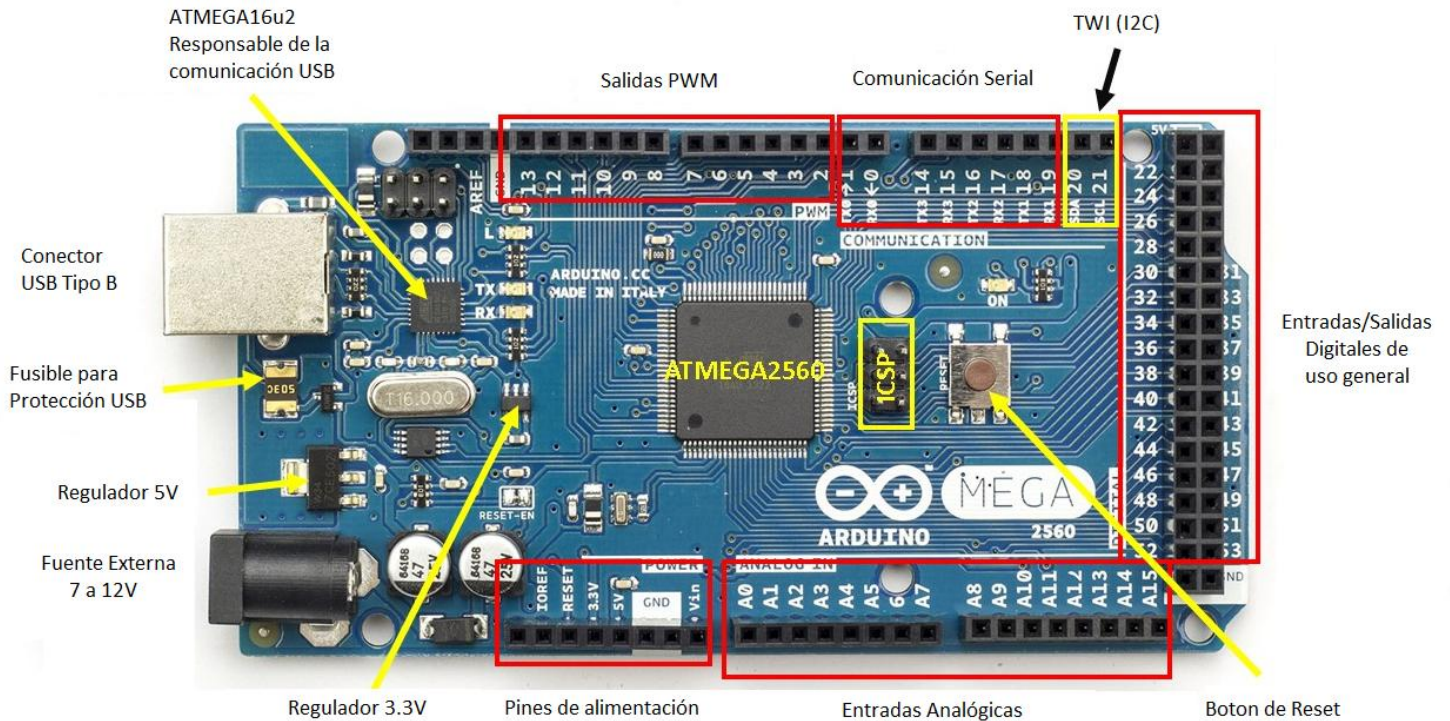


Figura 2.1 Estructura de la Placa Arduino Mega 2560.

2.1.1. Alimentación.

El Arduino Mega se puede alimentar vía conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente. Las fuentes de alimentación externas pueden ser tanto un transformador o una batería. El transformador se puede conectar usando un conector macho de 2.1mm con centro positivo en el conector hembra de la placa. Los cables de la batería pueden conectarse a los pines Gnd y Vin en los conectores de alimentación (POWER) La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V, el pin de 5V puede proporcionar menos de 5 voltios y la placa puede volverse inestable; si se usan más de 12V los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios. Los pines de alimentación son los siguientes:

- **VIN.** La entrada de voltaje a la placa Arduino cuando se está usando una fuente externa de alimentación (en opuesto a los 5 voltios de la conexión USB). Se puede proporcionar voltaje a través de este pin, o, si se está alimentando a través de la conexión de 2.1mm, acceder a ella a través de este pin.
- **5V.** La fuente de voltaje estabilizado usado para alimentar el microcontrolador y otros componentes de la placa. Esta puede provenir de VIN a través de un regulador integrado a la placa, o proporcionada directamente por el USB u otra fuente estabilizada de 5V.
- **3V3.** Una fuente de voltaje de 3.3 voltios generada por un regulador integrado a la placa. Su corriente máxima soportada es de 50mA.
- **GND.** Pines de toma de tierra.

2.1.2. Memoria.

El ATmega2560 cuenta con 256KB de memoria flash para almacenar código, 8KB son usados para el arranque del sistema o bootloader. El ATmega2560 tiene 8 KB de memoria SRAM y 4KB de EEPROM, se puede acceder a esta para leer o escribir con la librería EEPROM.

2.1.3. Entradas y Salidas.

Cada uno de los 54 pines digitales en el Mega pueden ser utilizados como entradas o como salidas usando las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()`. Las E/S operan a 5 voltios. Cada pin puede proporcionar o recibir una corriente máxima de 40mA y tiene una resistencia interna de pull-up (desconectada por default) de 20-50kOhms. Además, algunos de estos pines tienen funciones especializadas:

- **Serie: 0 (RX) y 1 (TX), Serie 1: 19 (RX) y 18 (TX); Serie 2: 17 (RX) y 16 (TX); Serie 3: 15 (RX) y 14 (TX).** Usados para recibir (RX) y transmitir (TX) datos a través de un puerto serie TTL. Los pines Serie: 0 (RX) y 1 (TX) están conectados a los pines correspondientes del chip FTDI USB-to-TTL.
- **Interrupciones Externas: 2 (interrupción 0), 3 (interrupción 1), 18 (interrupción 5), 19 (interrupción 4), 20 (interrupción 3), y 21 (interrupción 2).** Estos pines pueden ser configurados para lanzar una interrupción en un valor LOW(0V), en flancos de subida o bajada (cambio de LOW(0V) a HIGH(5V) o viceversa), o en cambios de valor. Con la función `attachInterrupt()`.
- **PWM: de 0 a 13.** Proporcionan una salida PWM (Pulse Wave Modulation, modulación de onda por pulsos) de 8 bits de resolución (valores de 0 a 255) por medio de la función `analogWrite()`.
- **SPI: 50 (SS), 51 (MOSI), 52 (MISO), 53 (SCK).** Estos pines proporcionan comunicación SPI (Serial Peripheral Interface), usando la librería SPI.
- **LED: 13.** Un LED integrado en la placa conectado al pin digital 13, cuando este pin tiene un valor HIGH(5V) el LED enciende y cuando este tiene un valor LOW(0V) este se apaga.

El Mega tiene 16 entradas analógicas, y cada una de ellas proporciona una resolución de 10bits (1024 valores). Tiene preestablecido un rango de medición de 0V a 5V, aunque es posible cambiar el límite superior de este rango usando el pin AREF y la función `analogReference()`.

- **I2C: 20 (SDA) y 21 (SCL).** Soporte para el protocolo de comunicaciones I2C (TWI), utilizado principalmente para la comunicación entre diferentes partes de un circuito, por ejemplo, entre un controlador y circuitos periféricos integrados, usando la librería `Wire`.
- **AREF.** Voltaje de referencia para las entradas analógicas dado por `analogReference()`.

- **Reset.** Se suministra un valor LOW (0V) para reiniciar el microcontrolador. Típicamente usado para añadir un botón de reset a los shields que no dejan acceso a este botón en la placa.

2.1.4. Comunicaciones.

EL Arduino Mega facilita en varios aspectos la comunicación con la PC, otro Arduino u otros microcontroladores. El ATmega2560 provee cuatro puertos de comunicación vía serie UART TTL (5V). Un ATmega16U2 integrado en la placa canaliza esta comunicación serie a través del puerto USB y los drivers (incluidos en el software de Arduino), estos proporcionan un puerto serie virtual en el ordenador. El software incluye un monitor de puerto serie que permite enviar y recibir información textual de la placa Arduino. Los LEDS RX y TX de la placa parpadearán cuando se detecte comunicación transmitida a través de la conexión USB (no parpadearán si se usa la comunicación serie a través de los pines 0 y 1). La librería SoftwareSerial permite comunicación serie por cualquier par de pines digitales del Mega. El ATmega2560 también soporta la comunicación I2C (TWI) y SPI. El software de Arduino incluye una librería Wire para simplificar el uso el bus I2C.

2.1.5. Programación.

El Arduino Mega se puede programar con el software Arduino. El ATmega2560 en el Arduino Mega viene precargado con un gestor de arranque (bootloader) que permite cargar nuevos códigos sin necesidad de un programador por hardware externo. Se comunica utilizando el protocolo STK500 original. También puede evitarse el gestor de arranque y programar directamente el microcontrolador a través del puerto ICSP (In Circuit Serial Programming) el cual es la habilidad de algunos dispositivos lógicos programables, microcontroladores y otros circuitos electrónicos, de ser programados mientras están instalados en un sistema completo, en lugar de requerir que el chip sea programado antes de ser instalado dentro del sistema.

2.1.6. Reinicio Automático por Software.

En vez de necesitar reiniciar presionando físicamente el botón de reset antes de cargar, el Arduino Mega está diseñado de manera que es posible reiniciarlo por software desde el ordenador donde esté conectado. Una de las líneas de control de flujo (DTR) del ATmega16U2 está conectada a la línea de reinicio del ATmega2560 a través de un condensador de 100 nanofaradios. Cuando la línea se pone en LOW(0V), la línea de reinicio también se pone en LOW el tiempo suficiente para reiniciar el chip. El software de Arduino utiliza esta característica para permitir cargar los programas con tan solo presionar un botón del entorno. Dado que el gestor de arranque tiene un lapso de tiempo para ello, la activación del DTR y la carga del programa están coordinados perfectamente.

Esta configuración tiene otras implicaciones. Cuando el Mega se conecta a un ordenador que utiliza Mac OS X o Linux, esto reinicia la placa cada vez que se realiza una conexión desde el software (via USB). Aproximadamente al medio segundo posterior, el gestor de arranque se está ejecutando. A pesar de estar programado para ignorar datos mal formateados, intercepta los primeros bytes que se envían a la placa justo después de que se abra la conexión. Si un programa ejecutándose en la placa recibe algún tipo de configuración inicial u otro tipo de información al inicio del programa, debe asegurarse de que el software con el cual se comunica se espera un segundo después de abrir la conexión antes de enviar los datos.

El Mega contiene una pista que puede ser cortada para deshabilitar el auto-reset. Las terminaciones a cada lado pueden ser soldadas entre ellas para rehabilitarlo. Están etiquetadas con “RESET-EN”. También se puede deshabilitar el auto-reset conectando una resistencia de 110 ohms desde el pin 5V al pin de reset.

2.1.7. Protección Contra Sobrecorrientes en USB.

El Arduino Mega tiene un multifusible reinicializable que protege la conexión USB de la PC de cortocircuitos y sobretensiones. Aparte de que la mayoría de ordenadores proporcionan su propia protección interna, el fusible proporciona una capa extra de protección. Si más de 500mA son detectados en el puerto USB, el fusible automáticamente corta la conexión hasta que el cortocircuito o la sobretensión desaparecen.

2.1.8. Características físicas y compatibilidad de Shields.

La longitud y amplitud máxima de la placa Mega 2560 son de 100 y 50 milímetros respectivamente, con el conector USB y la conexión de alimentación sobresaliendo de estas dimensiones. Tres agujeros para fijación con tornillos permiten colocar la placa en superficies y cajas. El Mega es compatible con la mayoría de los shield o tarjetas de aplicación/ampliación disponibles para las tarjetas Arduino UNO original.

2.2. MOTOR DE CORRIENTE ALTERNA. [7]

EL motor asíncrono consta de dos partes indispensables. 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 unas ranuras. 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 conectará 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 un bobinado eléctrico. Los más utilizados son el rotor de jaula de ardilla y el rotor bobinado.

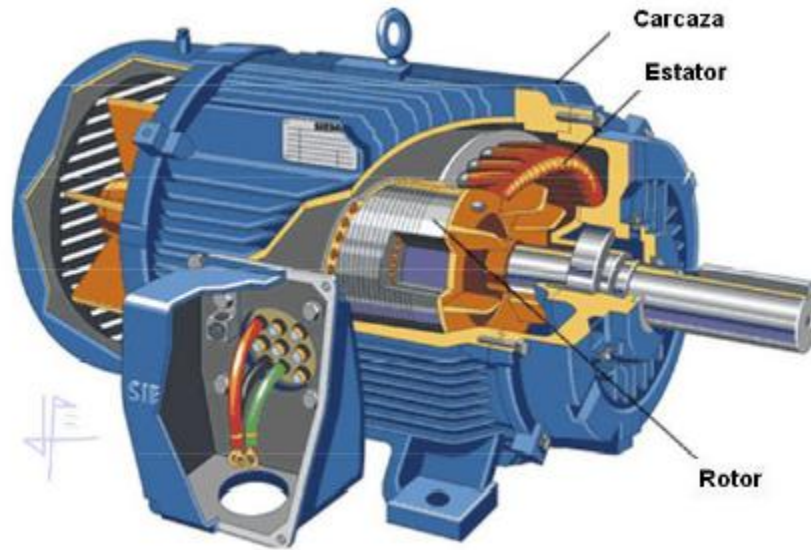


Figura 2.2 Estructura de un motor asíncrono de inducción.

A este tipo de motores se le conoce como motores de inducción debido a que su funcionamiento se basa en la interacción de campos magnéticos producidos por corrientes eléctricas. Las corrientes que circulan por el rotor son producidas por el fenómeno de inducción electromagnética, conocido comúnmente como ley de Faraday, que establece que si una espira es atravesada por un campo magnético variable en el tiempo se establece entre sus extremos una diferencia de potencial dado por la ecuación (3.1):

$$e = - \frac{d\Phi}{dt} \quad (2.1)$$

Donde:

e = Diferencia de potencial en la espira en Voltios

Φ = Flujo que corta a la espira en Weber

t = Tiempo en segundos

El signo negativo de la ecuación es una expresión de la ley de Lenz. Esta establece que la polaridad del voltaje inducido es tal que si sus extremos se pusieran en corto circuito, produciría una corriente que causaría un flujo para oponerse al cambio de flujo original.

Puesto que el voltaje inducido se opone al cambio que lo causa, se incluye el signo menos en la ecuación. Si se distribuye espacialmente alrededor del estator de un motor los bobinados de un sistema de tensiones trifásicas desfasadas 120° se genera un campo magnético giratorio. La velocidad de giro de este campo magnético, denominada velocidad de sincronismo, se da por la ecuación (2.2):

$$n = \frac{60*f}{p} \quad (2.2)$$

Donde:

n = Velocidad de giro del campo magnético en r.p.m.

f = Frecuencia de la corriente eléctrica de alimentación de la máquina.

p = Número de pares de polos magnéticos establecidos en el bobinado del estator.

En los motores eléctricos, la velocidad de giro del rotor es ligeramente inferior a la velocidad de giro del campo magnético del estator, debido a la fricción del rotor entre los cojinetes, rozamiento con el aire y la carga acoplada al eje del rotor, y es por tal motivo que se les conoce como motores asíncronos.

Se puede clasificar al motor asíncrono según el número de devanados en el estator:

- Monofásicos: tienen un sólo devanado en el estator. Se utilizan tanto en el hogar como en la industria (bombas, ventiladores, lavadoras, electrodomésticos en general, pequeñas máquinas-herramientas, etc.). [8]
- Bifásicos: tienen dos devanados en el estator los cuales están desfasados $\pi/(2p)$, siendo p el número de pares de polos de la máquina, en el espacio. Suelen ser utilizados en aplicaciones de control de posición. [8]

- Trifásicos: tienen tres devanados en el estator que se desfasan $2 \cdot \pi / (3p)$, siendo p el número de pares de polos de la máquina, en el espacio. Se suelen utilizar en aplicaciones industriales tales como: máquinas-herramientas (tornos, fresadoras, cepilladoras, etc.), grúas, bombas, compresores, ventiladores, etc. [8]

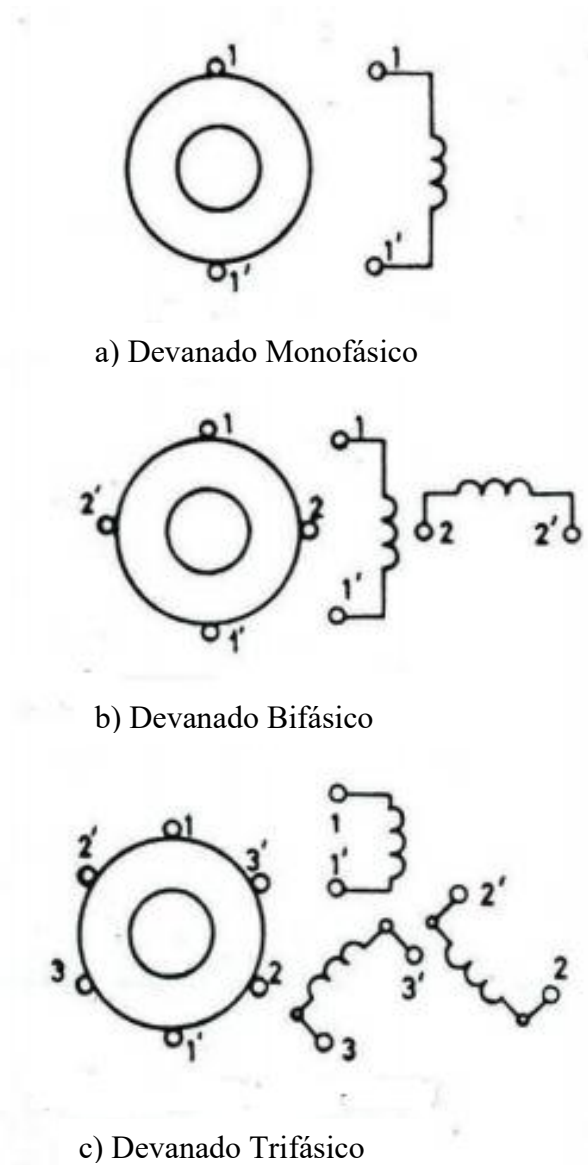


Figura 2.3 Tipos de motores asíncronos basados en el devanado del estator.

2.2.1. Motores de rotor de jaula de ardilla. [7]

El motor de rotor de jaula de ardilla, conocido como rotor en cortocircuito, es el más sencillo y más utilizado actualmente. El núcleo del rotor está construido de chapas estampadas de acero al silicio, en el interior de las cuales se disponen unas barras generalmente de aluminio moldeado a presión. Dichas barras van conectadas a unos anillos conductores denominados anillos extremos. El bobinado de esta manera asemeja la forma de una jaula de ardilla.

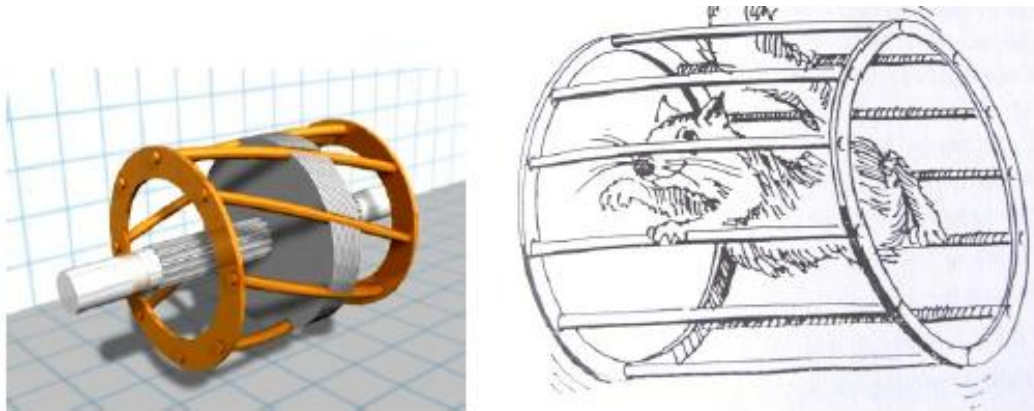


Figura 2.4 Rotor de Jaula de ardilla.

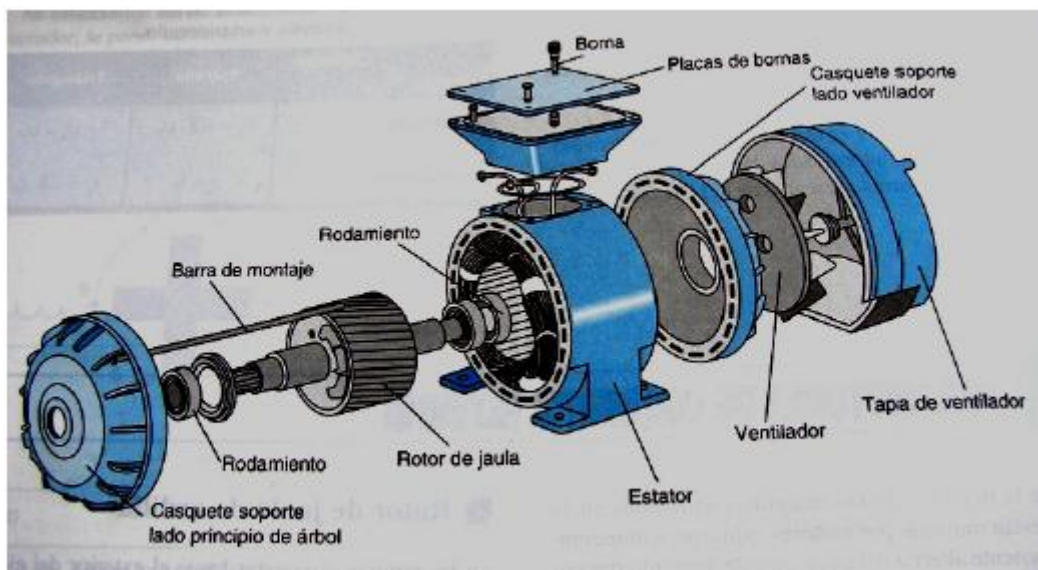


Figura 2.5 Estructura de un Motor con Rotor de Jaula de ardilla.

Las ranuras del rotor suelen construirse oblicuas respecto al eje para evitar puntos muertos en la inducción electromagnética. Uno de los inconvenientes de este tipo de motor es que en el arranque absorbe una corriente muy intensa (de 4 a 7 veces la nominal asignada), y lo hace con un bajo factor de potencia y su par suele ser bajo. Otro inconveniente de estos motores era que su velocidad no era regulable, pero actualmente con los variadores de velocidad electrónicos se puede conseguir un control perfecto de la práctica totalidad de parámetros del motor, entre los que destacan el par, la corriente absorbida y la velocidad de giro y la baja resistencia del rotor hace que tengan excelentes características para marchas a velocidades constantes.

2.2.2. Conexión de los bobinados de un motor eléctrico trifásico. [7]

El estator de un motor trifásico suele bobinarse con tres devanados distintos que corresponden a cada una de las fases con las que se les habrá de conectar en la red eléctrica. Según la forma de conectar las bobinas se pueden obtener dos conexiones: Delta y Estrella.

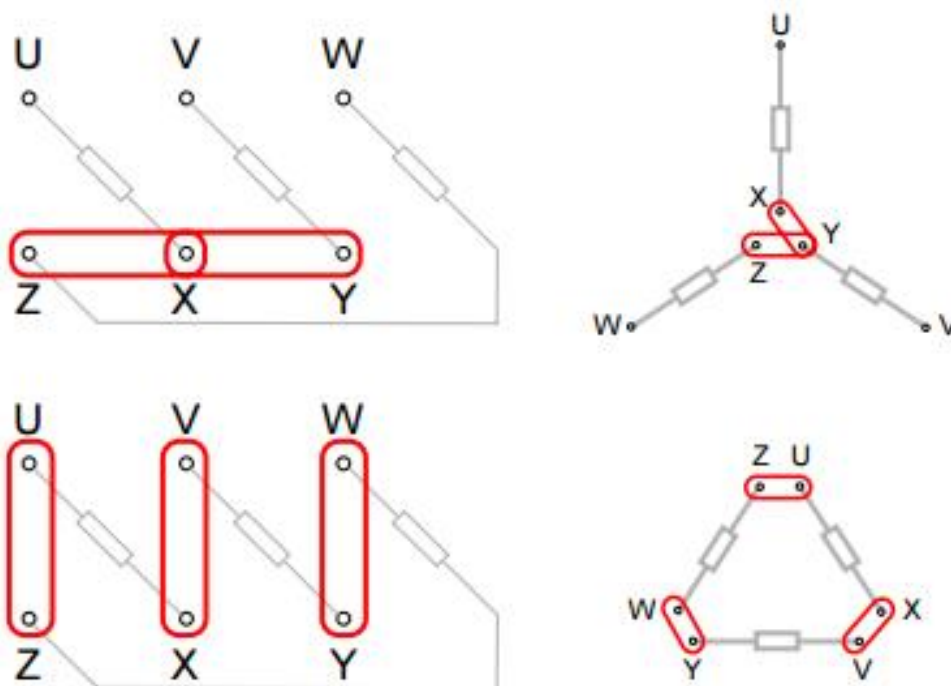


Figura 2.6 Conexión estrella y delta respectivamente.

Un motor conectado en estrella soporta el máximo voltaje que indica la placa de características del motor, mientras que en delta el voltaje que soporta es la más baja indicada por dicha placa. En el caso de las corrientes absorbidas ocurre justo lo contrario, correspondiéndole a la corriente más alta a la conexión delta. Si se desea cambiar el sentido de giro de un motor, basta simplemente con intercambiar dos de las fases de alimentación como se muestra en la figura 2.7.

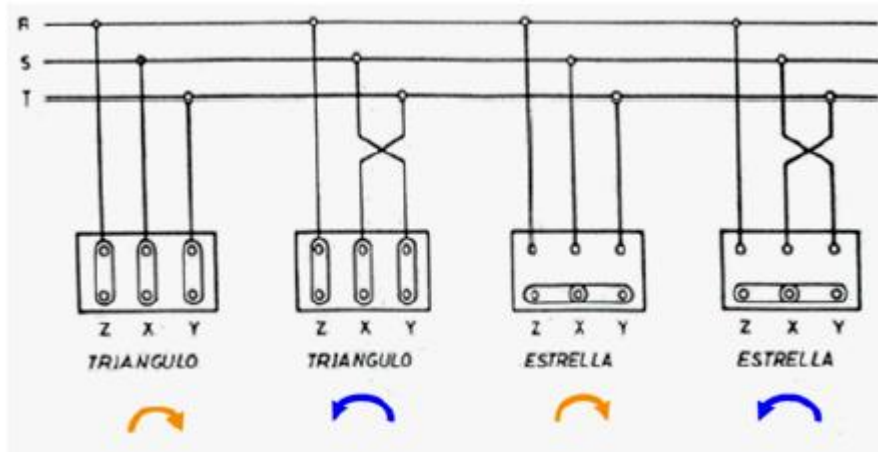


Figura 2.7 Conexión para el cambio de giro de un motor.

2.2.3. Sistema de arranque directo de los motores trifásicos. [10]

La forma más simple de arrancar un motor de jaula de ardilla es conectar el motor directamente a la red como se muestra en la figura (2.8). En el momento de la puesta bajo voltaje, el motor actúa como un transformador cuyo secundario, constituido por la jaula de poca resistencia del rotor, está en cortocircuito. La corriente primaria y la secundaria son prácticamente proporcionales.

Se obtiene un pico de corriente importante en la red:

$$I_{arr} = 5 \text{ a } 8 \text{ veces } I_n$$

Su par de arranque medio es:

$$M_{arr} = 0.5 \text{ a } 1.5 \text{ veces } M_n$$

El arranque directo tiene una serie de ventajas:

- Sencillez del equipo
- Par de arranque elevado
- Rápido arranque
- Bajo costo

A pesar de las ventajas que proporciona este tipo de arranque, sólo se puede utilizar en los casos siguientes:

- La potencia del motor es débil respecto a la de la red, para poder limitar las perturbaciones que provoca la corriente solicitada.
- El motor no requiere un aumento progresivo de velocidad y dispone de un dispositivo mecánico que impide el arranque brusco.
- El par de arranque tiene que ser elevado.

Por otro lado, será indispensable recurrir a algún procedimiento que disminuya la corriente solicitada o el par de arranque, siempre que:

- La caída de voltaje provocada por la corriente que solicita perturbe el buen funcionamiento de algún otro aparato conectado a la misma red.
- La máquina accionada no pueda soportar sacudidas mecánicas.
- La seguridad y/o comodidad de los usuarios se vea comprometida.

En estos casos, el método más utilizado consiste en arrancar el motor bajo un voltaje reducido. La variación del voltaje de alimentación posee las siguientes consecuencias:

- La corriente de arranque varía proporcionalmente al voltaje de alimentación.
- El par de arranque varía proporcionalmente al cuadrado del voltaje de alimentación.

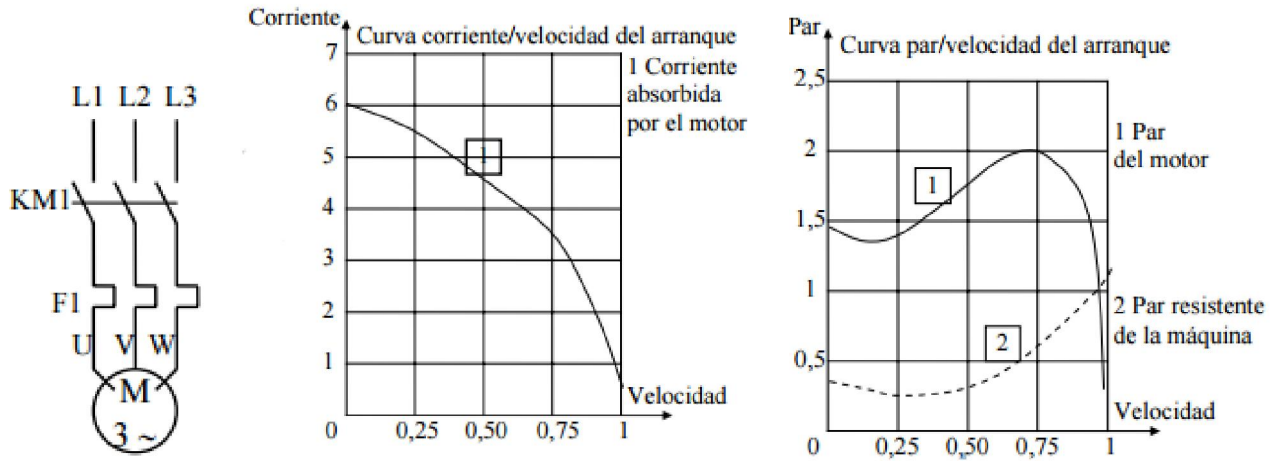


Figura 2.8. a) Circuito de fuerza arranque directo.

b) Curva Corriente/Velocidad arranque directo.

c) Curva Par/Velocidad arranque directo.

2.3. POTENCIA EN SISTEMAS TRIFÁSICO. [9]

Debido a que el motor de inducción representa un sistema trifásico equilibrado podremos obtener sus potencias por medio de las fórmulas (2.3), (2.4) y (2.5). Ya que un sistema trifásico se puede considerar como 3 circuitos monofásicos, por lo que la potencia total instantánea transferida a un circuito trifásico será igual a la suma de las potencias instantáneas transferidas a cada uno de los tres sistemas monofásicos por lo tanto la potencia activa será:

$$P = \sum_{k=1}^3 V_k I_k \cos \phi_k = V_1 I_1 \cos \phi_1 + V_2 I_2 \cos \phi_2 + V_3 I_3 \cos \phi_3 = 3 * V_F I_F \cos \phi_F \quad (2.3)$$

Donde:

V_k = Voltaje eficaz de las fases del motor.

I_k = Corriente eficaz de las fases del motor.

ϕ_k = Ángulo de desfase ($\phi_V - \phi_I$).

$\cos \phi_k$ = Factor de potencia de la carga.

Por otro lado la potencia reactiva se calcula así:

$$Q = \sum_1^3 V_k I_k \text{sen} \phi_k = V_1 I_1 \text{sen} \phi_1 + V_2 I_2 \text{sen} \phi_2 + V_3 I_3 \text{sen} \phi_3 = 3 * V_F I_F \text{sen} \phi_F \quad (2.4)$$

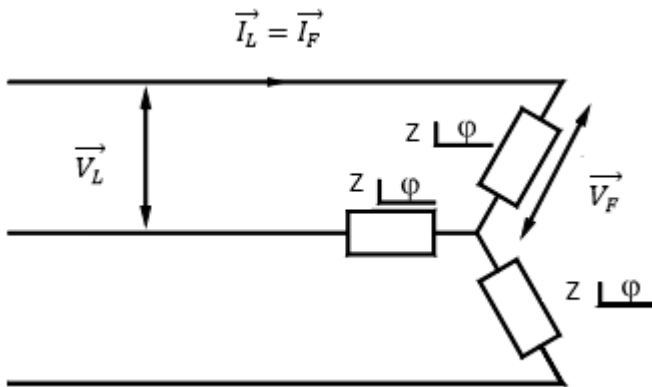
Y la potencia aparente será:

$$S = \sqrt{P_1^2 + Q_1^2} + \sqrt{P_2^2 + Q_2^2} + \sqrt{P_3^2 + Q_3^2} = S_1 + S_2 + S_3 \quad (2.5)$$

Una vez obtenidos la potencia activa (P) y la potencia aparente (S), se puede obtener el factor de potencia el cual se calcula de la siguiente manera:

$$f.p. = \frac{p}{S} \quad (2.6)$$

2.3.1. Conexión en estrella.



El voltaje y la corriente de fase en función de las líneas se obtienen de la siguiente manera:

$$V_L = \sqrt{3} V_F \rightarrow V_F = \frac{V_L}{\sqrt{3}} \quad (2.7)$$

$$I_L = I_F \quad (2.8)$$

Figura 2.9. Conexión en estrella de un sistema trifásico.

Entonces la potencia Activa, Reactiva y Aparente en función de las líneas resulta ser:

$$P = 3 * V_F I_F \cos\phi = 3 \frac{V_L}{\sqrt{3}} I_L \cos\phi = \sqrt{3} V_L I_L \cos\phi \quad (2.9)$$

$$Q = 3 * V_F I_F \sen\phi = 3 \frac{V_L}{\sqrt{3}} I_L \sen\phi = \sqrt{3} V_L I_L \sen\phi \quad (2.10)$$

$$S = 3 * V_F I_F = 3 \frac{V_L}{\sqrt{3}} I_L = \sqrt{3} V_L I_L \quad (2.11)$$

2.4. ANÁLISIS DE UN MOTOR TRIFÁSICO POR MEDIO DE SU PLACA DE CARACTERÍSTICAS. [7]

La figura (2.10) representa un ejemplo de una placa de características que corresponde a un motor trifásico. Enseguida se analizará con detalle los distintos datos y su significado.

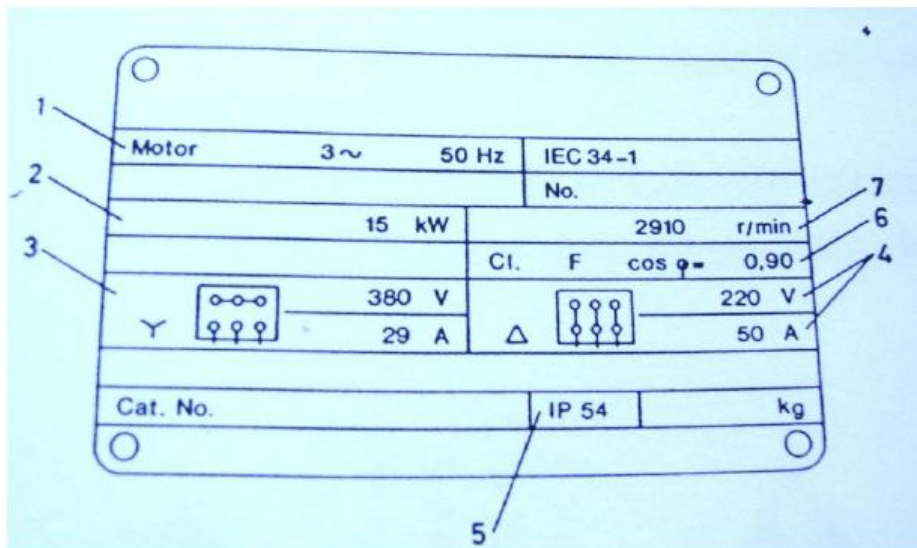


Figura 2.10 Ejemplo de placa de características de un Motor.

1. Se trata de un motor trifásico de corriente alterna a 50 Hz.
2. Su potencia nominal o asignada en el eje del motor es de 15 KW.
3. Los bobinados se pueden conectar en estrella hasta un voltaje máximo de 380 V, circulando una corriente por cada línea de alimentación de 29 A.
4. En conexión delta el voltaje compuesto entre las fases de alimentación podrá ser un máximo de 220 V, circulando por cada una de las líneas de alimentación una corriente de 50 A.
5. Indica el grado de protección de la carcasa del motor contra agentes externos, basándose en la clasificación establecida por la IEC (Comisión Electrotécnica Internacional), para este caso:
IP54: Carcasa protegida a prueba de polvo y proyecciones de agua
6. Cl F indica la clase del motor refiriéndose a la máxima temperatura a la que puede operar y al tipo de aislamiento, en este caso “clase F”, que indica que puede funcionar hasta una temperatura máxima de 155°C.

Por otro lado, “ $\cos\phi = 0,9$ ” se refiere al factor de potencia, lo que permite hacer algunos cálculos básicos:

Se obtendrán los mismos datos operando con los datos de conexión estrella o delta.

- a) Potencia activa que es absorbida de la red: El voltaje (entre fases) de la línea de alimentación es 380 V, por tanto está conectado en estrella como se indica en la placa, entonces se obtiene:

$$P = \sqrt{3} * V * I * \cos\phi = 1.732 * 380 * 29 * 0.9 = 17178 W$$

b) Potencia reactiva de tipo inductivo que es absorbida:

$$\cos\phi = 0.9 \rightarrow \phi = 25^\circ 50'30.96'' \rightarrow \text{sen}\phi = 0.436$$

$$Q = \sqrt{3} * V * I * \text{sen}\phi = 1.732 * 380 * 29 * 0.436 = 8322 \text{ VAR}$$

c) Considerando el triángulo de potencias, se puede averiguar la potencia aparente que es demandada:

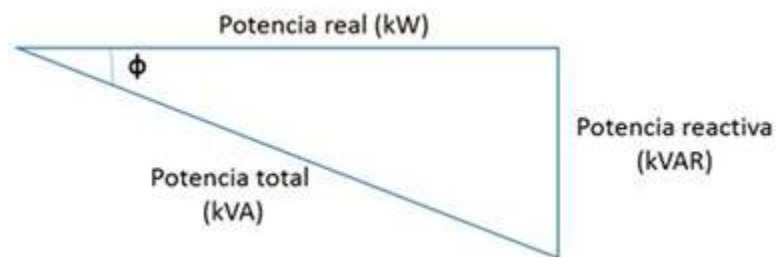


Figura 2.11 Triangulo de potencias.

$$S = \sqrt{P^2 + Q^2} \quad \cos\phi = \frac{P}{S} \quad \text{sen}\phi = \frac{Q}{S} \quad \tan\phi = \frac{Q}{P}$$

$$S = \sqrt{(17178)^2 + (8322)^2} \quad \text{ó} \quad S = \frac{17178 \text{ W}}{0.9} = 19087 \text{ KVA}$$

d) Una vez teniendo la potencia activa y la potencia aparente se puede obtener el factor de potencia de la siguiente forma:

$$f.p. = \frac{P}{S} = \frac{17178 \text{ W}}{19087 \text{ KVA}} = 0.9 \text{ en atraso}$$

e) El rendimiento, en porcentaje, del motor se obtiene de la siguiente forma:

$$\eta = \frac{\text{Potencia en el eje}}{\text{Potencia absorbida}} * 100 = \frac{15000 \text{ W}}{17178 \text{ W}} * 100 = 0.87 * 100 = 87\%$$

7. Velocidad del motor de inducción a plena carga 2910 r.p.m.

2.5. EFECTO HALL. [11]

El efecto Hall, descubierto por Edwin C. Hall en 1879, consiste en la producción de una caída de voltaje a través de un conductor o semiconductor con corriente, bajo la influencia de un campo magnético externo. Para esto es necesario que la dirección del campo magnético sea perpendicular a la dirección de flujo de la corriente.

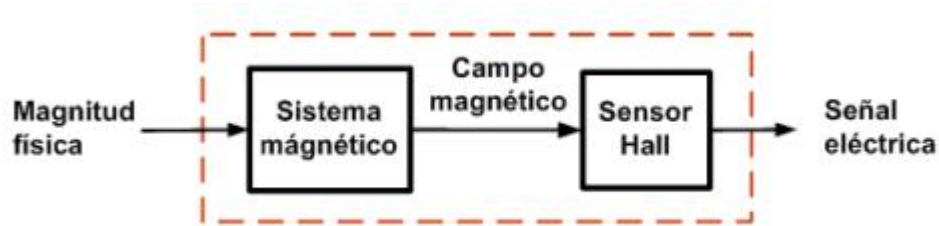


Figura 2.12. Diagrama del funcionamiento del efecto Hall.

El campo magnético transversal ejerce una fuerza desviadora (Fuerza de Lorentz) sobre el conductor o semiconductor. Esta fuerza causa la desviación de los portadores de carga que se mueven a través del material. Como resultado, aparece una diferencia de potencial V_{xy} (denominada voltaje de Hall) entre los extremos del conductor. Este voltaje es proporcional a la intensidad del campo magnético aplicado y su polaridad depende del signo de los portadores de carga.

El efecto Hall se presenta en conductores y en semiconductores. Las diferencias de potencial producidas en tiras metálicas son muy pequeñas, siendo a menudo enmascaradas por el ruido. Por esto, los dispositivos comerciales usan materiales semiconductores especiales, donde el efecto Hall es más notable.

En estos casos, el elemento básico es generalmente una tira de arseniuro de galio (GaAs) o de indio (InAs) la cual, cuando se polariza mediante una corriente constante y se sumerge en un campo magnético transversal a su superficie, genera un voltaje proporcional a la intensidad del campo. Este voltaje es reforzado por un amplificador operacional incorporado en el dispositivo y se procesa para proporcionar una señal de salida útil.

Dependiendo el tipo de salida, los sensores Hall se dividen en dos grupos:

- Sensores Hall de salida lineal
- Sensores Hall de salida digital (interruptores Hall)

2.5.1. Sensores de efecto Hall con salida lineal. [11]

En la figura (2.15) se puede ver la estructura simplificada de un sensor Hall de salida lineal. Debido a que la tensión Hall es muy pequeña (aproximadamente $30\mu\text{V}/\text{G}$), es necesaria una etapa de amplificación, esto se hace a través de un amplificador diferencial. Posee además una compensación en temperatura y un regulador de voltaje. La salida posee un transistor en emisor abierto para realizar la interfaz con cualquier dispositivo.

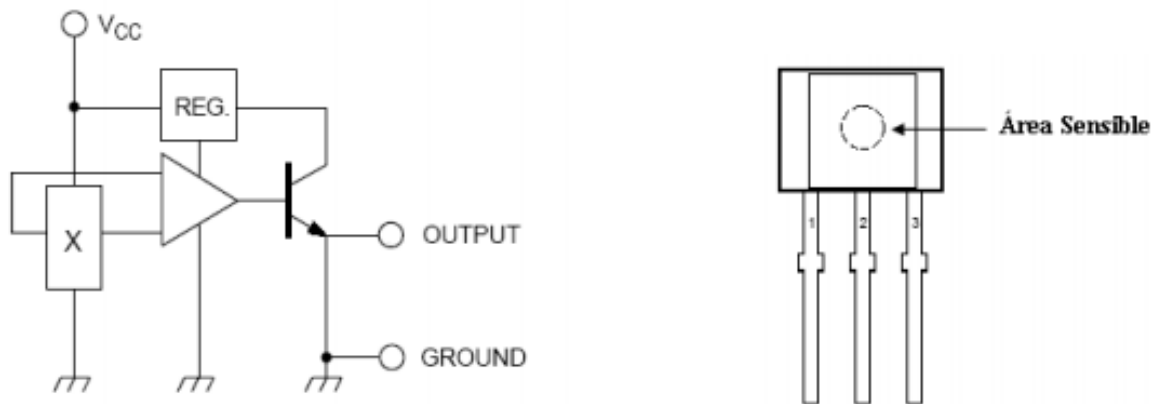


Figura 2.13 Estructura simplificada de un sensor Hall de salida lineal.

Según la polaridad del campo magnético que se le aplique al sensor, la tensión de salida será positiva o negativa, Pero el amplificador diferencial incorpora una tensión de polarización, de modo que con un campo magnético nulo se tiene en la salida una tensión

positiva. Esta tensión es la que se conoce como null offset o quiescent output voltaje. En la figura (2.15) se muestra la característica de transferencia de un sensor Hall lineal.

Se puede ver en dicha gráfica el null offset, la sensibilidad (en mV/G) y el alcance (span). Estos dispositivos generalmente poseen una salida en emisor abierto, permitiendo conectarlos fácilmente con casi cualquier circuito.

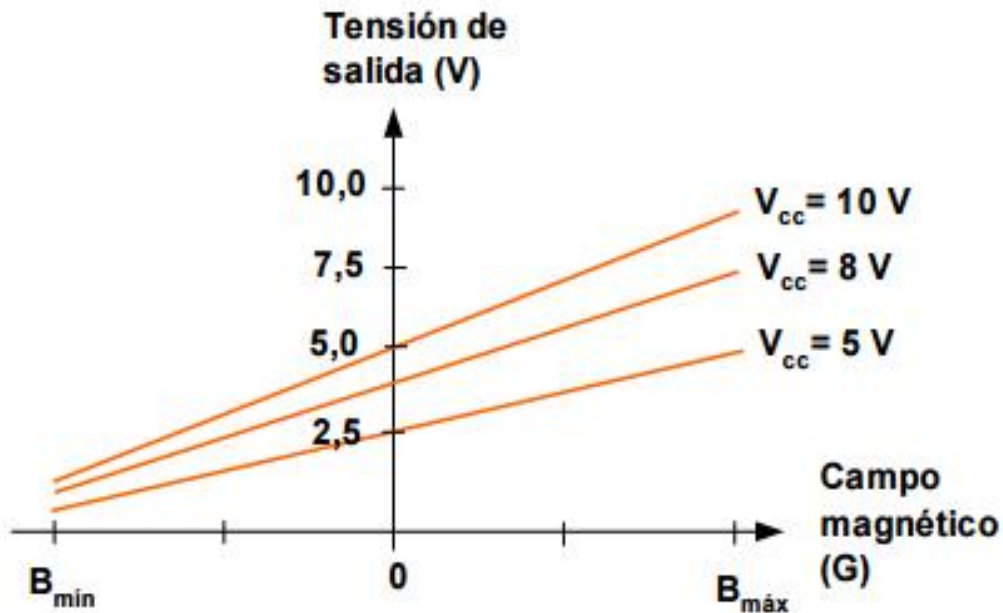


Figura 2.14 Características de transferencia de un sensor Hall lineal.

Otras características importantes de los sensores Hall lineales son:

- Sensibilidad (mV/G). Es la pendiente de la característica de transferencia. En la mayoría de los sensores Hall lineales la sensibilidad y el offset son proporcionales a la tensión de alimentación. Se dice que son sensores ratiométricos.
- Alcance (span). Define el margen de salida del sensor

2.5.2. Sensor Hall de salida digital.[11]

La estructura de este tipo de sensor es la misma que la del lineal pero se debe agregar un comparador Smitt trigger como se muestra en la figura (2.15).

Sin campo magnético aplicado al dispositivo, el transistor de salida permanece cortado (OFF). Al aplicar un campo magnético, perpendicular a la superficie del dispositivo, que está por encima de un valor umbral, el transistor de salida pasa a saturación (ON). Si se reduce ahora el campo magnético, el transistor conmutará para un valor menor que el umbral de conducción. Como se puede observar este dispositivo presenta cierta histéresis que previene disparos erróneos.

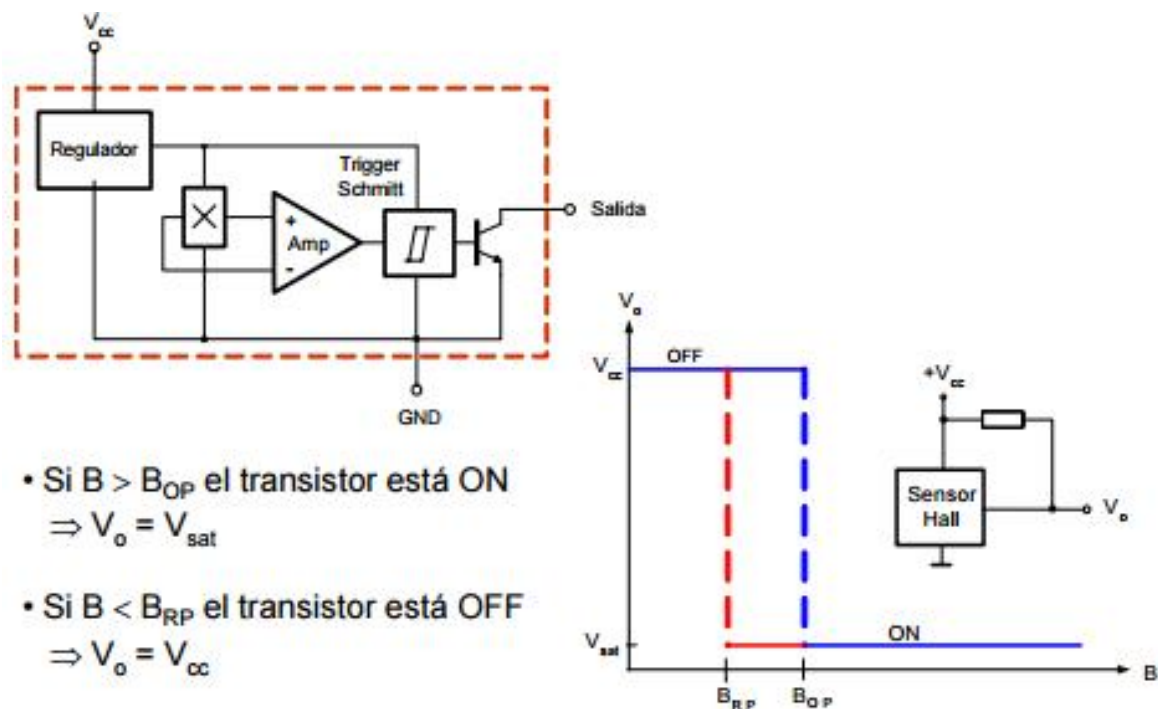


Figura 2.15 Estructura de un sensor de efecto Hall de salida digital.

Estos dispositivos generalmente poseen una salida en emisor abierto, permitiendo conectarlos fácilmente con casi cualquier circuito. Es común utilizar circuitos con comparadores o amplificadores operacionales. En la tabla 2.1 se muestran algunas aplicaciones más comunes de cada tipo de sensor.

Tabla 2.1 Aplicaciones de los sensores de efecto Hall

Aplicaciones Industriales	
Sensor Hall digital	Sensor Hall lineal
<ul style="list-style-type: none"> • Motor DC sin escobillas • Ventiladores • Medición de velocidad • Interruptor sin contacto • Medición de nivel de líquidos • Medición de flujo de líquidos • Interruptor de presión • Interruptor de rotación 	<ul style="list-style-type: none"> • Medición de corriente • Medición de posición • Medición de distancia • Medición de presión • Medición de fuerza • Medición de campo magnético • Medición de par

2.5.3. Sensor Hall ACS712.[12]

El ACS712 provee una solución precisa y económica para el sensado de corrientes alternas y directas industrialmente, comercialmente y en sistemas de comunicación. Sus aplicaciones típicas incluyen control de motores, detección y manejo de cargas, modo interruptor para fuentes de alimentación y protección contra sobrecorrientes de falla. Este dispositivo no está destinado para aplicaciones automotrices.

Consta de un circuito de efecto Hall lineal preciso, bajo offset, con una pista de cobre conductor ubicado cerca de la superficie del chip. La corriente que atraviesa esta pista de conducción genera un campo magnético que es detectado por el IC integrado Hall y convertida en una tensión proporcional. La exactitud de los dispositivos se optimiza a través de la proximidad de la señal magnética al transductor Hall. Una tensión precisa, proporcional es proporcionada por el bajo offset, chopper-estabilizado IC BiCMOS Hall, que está programado para la exactitud después del encapsulado.

Este sensor es un pequeña placa, que soporta un sensor de corriente de *Allegro*, el ACS715 20A está basado en el efecto Hall lineal, que ofrece una baja resistencia ($\sim 1,2$ m Ω) al paso de la corriente con un aislamiento eléctrico de hasta 2,1 kV RMS, según indica el fabricante.

Esta versión acepta una entrada de corriente continua de hasta 20 A y una salida de tensión analógica proporcional (66 mV/A) que mide 500 mV. Cuando la corriente de entrada es cero. El error de salida típico es de $\pm 1,5\%$. Opera desde 4,5 V a 5,5 V y está pensado para su uso en sistemas de 5 V.

Este dispositivo es muy sencillo, como se aprecia en la figura 2.19, dispone de tres pines, Vcc, GND y Salida en un extremo y en el otro, dispone de un conector con dos contactos para conectar la carga a la cual se le va a medir la corriente.

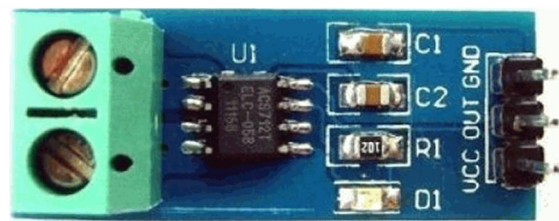


Figura 2.16. Placa con sensor ACS712 para Arduino.

El espesor de la pista de cobre permite la supervivencia del dispositivo en hasta $5 \times$ condiciones de sobre corriente. Los terminales de la pista conductora están eléctricamente aislados de los cables de los sensores (pines 5 a 8). Esto permite que el sensor de corriente ACS712 ser utilizado en aplicaciones que requieren aislamiento eléctrico sin el uso de aisladores ópticos u otras técnicas de aislamiento costosas.

El esquema práctico de conexionado es sencillo y fácil de entender. Se trata de conectar el sensor en serie con la carga que a la cual le se quiere conocer la corriente de consumo.

Tensión de salida en reposo ($V_{iout}(Q)$). La salida del sensor cuando la corriente primaria es cero. Para una tensión de alimentación unipolar, sigue siendo nominalmente $VCC / 2$. Por lo tanto, cuando no hay corriente la tensión de salida es $VCC / 2 = 5 \text{ V} / 2 = 2.5\text{V}$.

La variación en V_{out} (Q) se puede atribuir a la resolución de la compensación de tensión de reposo IC Allegro lineal y la deriva térmica. Cuando no hay corriente de paso por el sensor, leerá “cero”, pero esto no va a dar cero en las lecturas. Los puertos analógicos en Arduino utilizan un conversor A/D con una resolución de 10 bits que va de 0 a 1023, donde el 0 = 0 V y el 1023 = 5 V, como el sensor es bidireccional, cuando el sensor no detecta corriente el voltaje será de $2,5\text{ V} = (VCC / 2)$ y la lectura de Arduino será algo cercano a 512. Por lo tanto, si la corriente positiva va a aumentar la lectura y viceversa. En la tabla 2.2 se muestran las características de los diferentes modelos de sensor ACS712.

Tabla 2.2 *Tabla de valores de sensibilidad de los sensores ACS712*

Número de Parte	Empacado	T_A (°C)	Rango Optimizado, I_P (A)	Sensibilidad (mV/A)
ACS712ELCTR-05B-T	Tape and reel, 3000 pieces/reel	-40 a 85	±5	185
ACS712ELCTR-05B-T	Tape and reel, 3000 pieces/reel	-40 a 85	±20	100
ACS712ELCTR-05B-T	Tape and reel, 3000 pieces/reel	-40 a 85	±30	66

Para comprender mejor el funcionamiento del sensor, supongamos que este sensor mide corrientes entre -20 A y +20 A. Ya que la salida del sensor según la hoja de datos siempre entrega entre 0 y 5 voltios, cuyo valor de 0A se corresponde con los 2.5V en la salida, y por cada amperio la tensión varía 100 mV.

Este sensor puede medir tanto CD como CA, si vemos una senoide, el valor mínimo corresponde con los -20 A, de modo que los 2.5V de salida corresponden justo a 0 A y los +20 A con los 5 V, como se ve en la figura (2.17).

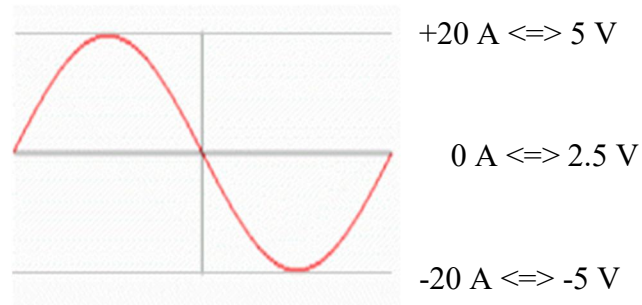


Figura 2.17. Equivalencia del voltaje de salida del sensor ACS712.

La forma de conectar este dispositivo se muestra en la figura (2.18)

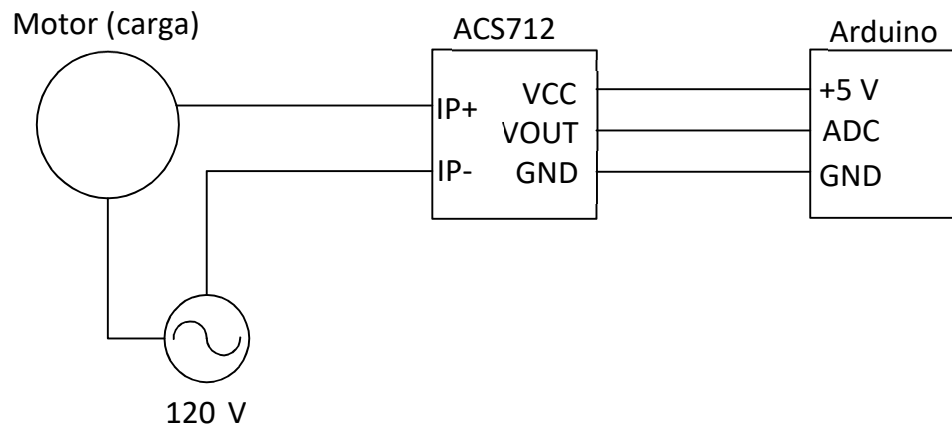


Figura 2.18 Diagrama de conexión del sensor ACS712 a una carga.

2.6. SENSOR DE VOLTAJE ZMPT101B. [13]

El sensor ZMPT101B es un módulo utilizado para medir la tensión de fase de corriente alterna. Está diseñado con un transformador de voltaje, por lo que solo se puede utilizar para leer el voltaje alterno. Éste se muestra en la figura (2.19).

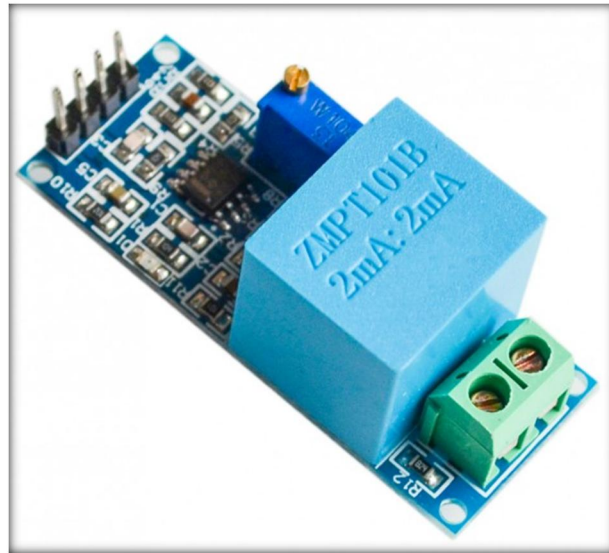


Figura 2.19 Sensor de Voltaje ZMPT101B.

Este es un módulo transformador de voltaje, tiene una salida monofásica activa. Contiene un circuito con amplificador operacional para compensar el offset de la salida analógica. Puede medir voltaje de baja tensión adecuado para voltaje como 220 V, tiene un potenciómetro en la placa de 0 a 10 k Ω para ajustar la amplitud de la señal de salida. Es un módulo ideal para aplicaciones de monitoreo y protección de equipos de CA, muy común en aplicaciones de domótica y retroalimentación para sistema de control de voltaje CA.

Puedes usar cualquier microcontrolador con entrada analógica para leer el voltaje instantáneo y así hacer cálculos de energía, necesitas calibrar la salida con un voltímetro para un resultado más preciso.

El sensor es alimentado por 5 V de CD, tiene una salida analógica proporcional a la señal de entrada, tiene un offset en la señal de salida la cual es equivalente a 2.5 V cuando está conectado a 0 V. El transformador tiene corriente nominal de entrada y salida de 2 mA, una relación de vueltas de 1000:1000 y una diferencia de fase $\leq 20^\circ$ (50Ω), 1% de linealidad y aislamiento eléctrico de hasta 3000 V. En la figura (2.18) se muestra como se conecta el sensor de voltaje.

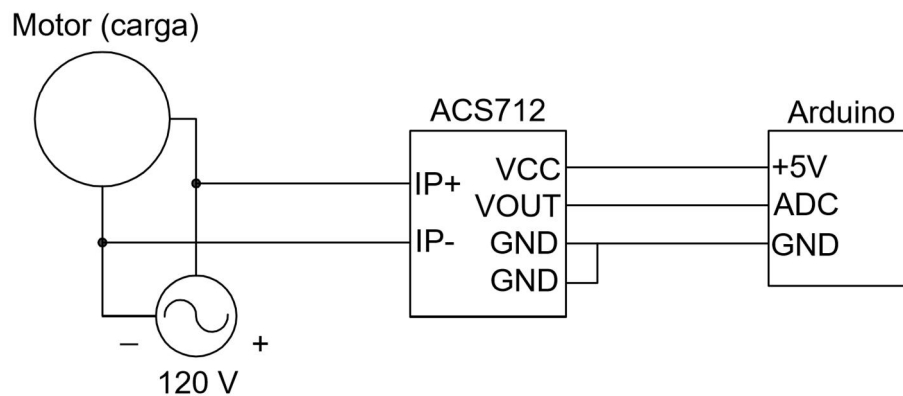


Figura 2.20 Diagrama de conexión del sensor de voltaje.

2.7. SENSOR INFRARROJO DE TEMPERATURA MLX90614. [14]

El MLX90614 es un sensor de temperatura infrarrojo sin de contacto, fabricado por la empresa Melexis. Es posible conectar estos sensores con un autómata o procesador como Arduino para medir la temperatura de un objeto a distancia. Existen distintos modelos del MLX90614 cada uno con un sufijo de tres letras. Los diferentes sensores difieren en el voltaje de operación, el número de sensores infrarrojos, y la posición del filtro.

La comunicación se realiza a través de SMBus, un subconjunto de bus I2C, por lo que resulta sencilla su lectura, y es posible conectar más de un sensor de forma simultáneamente. Frecuentemente se encuentran termómetros MLX90614 integrados en módulos como el GY-906 que incorporan la electrónica necesaria para conectarlo de forma sencilla a un Arduino.

En la mayoría de los módulos, esto incluye un regulador de voltaje que permite alimentar directamente a 5V Este tipo de termómetros infrarrojos tienen un gran número de aplicaciones, incluyendo sistemas de control de temperatura en instalaciones térmicas en edificios, control industrial de temperatura, detección de movimiento, y aplicaciones de salud. El modelo MLX90614ESF-BAA de Melexis mide temperaturas de entre -70 a $+380$ °C sin contacto. El elemento sensor es un chip de silicio con una fina membrana micromecanizada sensible a la radiación infrarroja de un objeto distante. Una etapa de adaptación de medida de la señal amplifica y digitaliza la señal procedente de la membrana y calcula la temperatura del objeto usando la calibración configurada en fábrica.

La temperatura de salida digital es linealizada y compensada completamente contra las variaciones de la temperatura ambiente. Incorpora amplificadores avanzados de bajo ruido, un convertidor A/D de 17 bits y un potente procesador digital de señales en su chip de acondicionamiento de señal, lo que permite un amplio rango de temperaturas de trabajo que van de -40 °C a 125 °C y un rango de temperaturas del objeto a medir que va de -70 °C a 380 °C, a la vez que mantiene una resolución de la temperatura de $0,02$ °C. Dispone de una salida SMBus (compatible con I2C) y puede ser configurado para una salida PWM de 10bits.

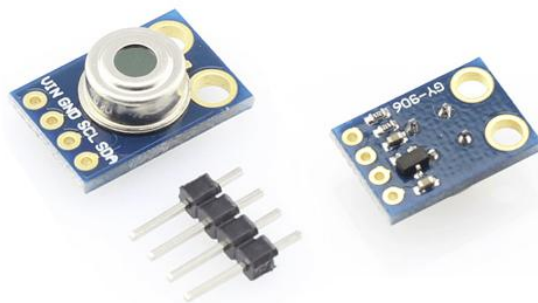


Figura 2.21 Sensor infrarrojo de temperatura MLX90614.

A diferencia de la mayoría de los sensores de temperatura, este sensor mide la luz infrarroja que rebota en los objetos remotos para que pueda detectar la temperatura sin tener que tocarlos físicamente. Simplemente se debe apuntar el sensor hacia lo que se desea medir y detectará la temperatura absorbiendo las ondas IR emitidas. Como no tiene que tocar el objeto que está midiendo, puede detectar un rango de temperaturas más amplio que la mayoría de los sensores. Toma la medida en un campo de visión de 90 grados, por lo que puede ser útil para determinar la temperatura promedio de un área.

2.7.1. Funcionamiento. [14]

Según la ley de Stefan-Boltzmann, todo objeto por encima del cero absoluto (°K) emite radiación cuyo espectro es proporcional a su temperatura. El MLX90614 recoge esta radiación y su salida es una señal eléctrica proporcional a la temperatura de todos los objetos en su campo de visión.

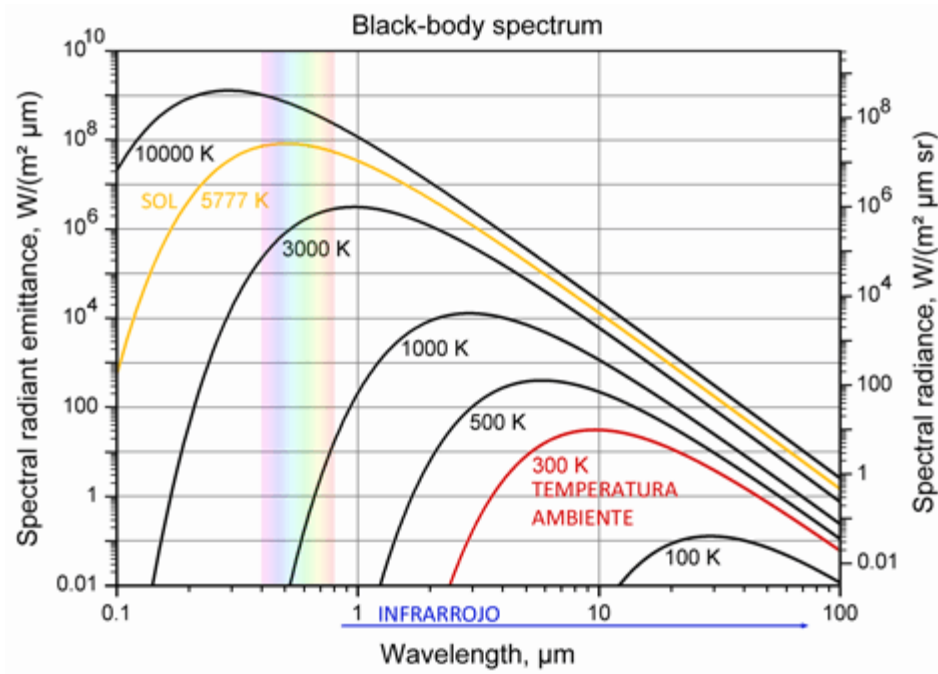


Figura 2.22 Espectro de objetos negros.

Internamente el MLX90614 está constituido con un chip de silicio con una fina membrana micromecanizada sensible a la radiación infrarroja, junto con la electrónica necesaria para amplificar y digitalizar la señal y calcular la temperatura. El conjunto incluye un amplificador de bajo ruido, un conversor ADC de 17 bits, un DSP (procesador digital de señal) y compensación de la temperatura ambiente.

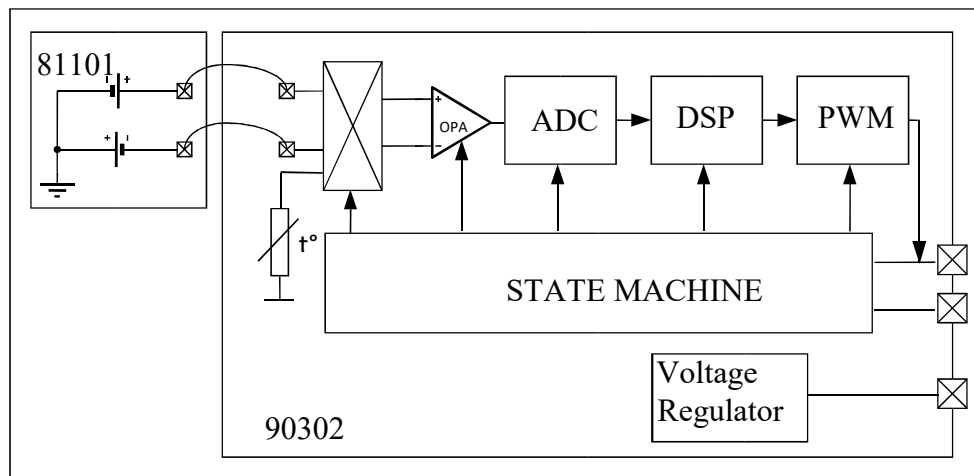


Figura 2.23 Composición del MLX90614.

El MLX90614 dispone de dos modos de salida. La estándar es SMBus, un conjunto del I2C, con una resolución de 0.02°C. También puede emplear una salida PWM de 10 bits para mediciones continuas, aunque con menor resolución 0.14°C. Es importante tener en cuenta la lectura del sensor solo es estable cuando el sensor se encuentra en equilibrio térmico con el ambiente. También puede afectarle la suciedad en la ventana del sensor.

También es importante entender que el MLX90614 es sensible a todos los objetos ubicados en su campo de visión. El ángulo de visión depende del modelo, y varía desde 5° a 80°C, el área de medición a 0.5 tiene un diámetro de 0.83 metros. Es decir, los modelos de menos ángulo son apropiados para medidas puntuales en frente del sensor. Los sensores de ángulo amplio están diseñados para detectar incrementos de temperatura en una gran zona, por ejemplo, para detección de fallas en maquinaria. La conexión es sencilla, simplemente se alimenta el módulo desde el Arduino mediante GND y 5V y se conecta el pin SDA y SCL de Arduino con los pines correspondientes del sensor.

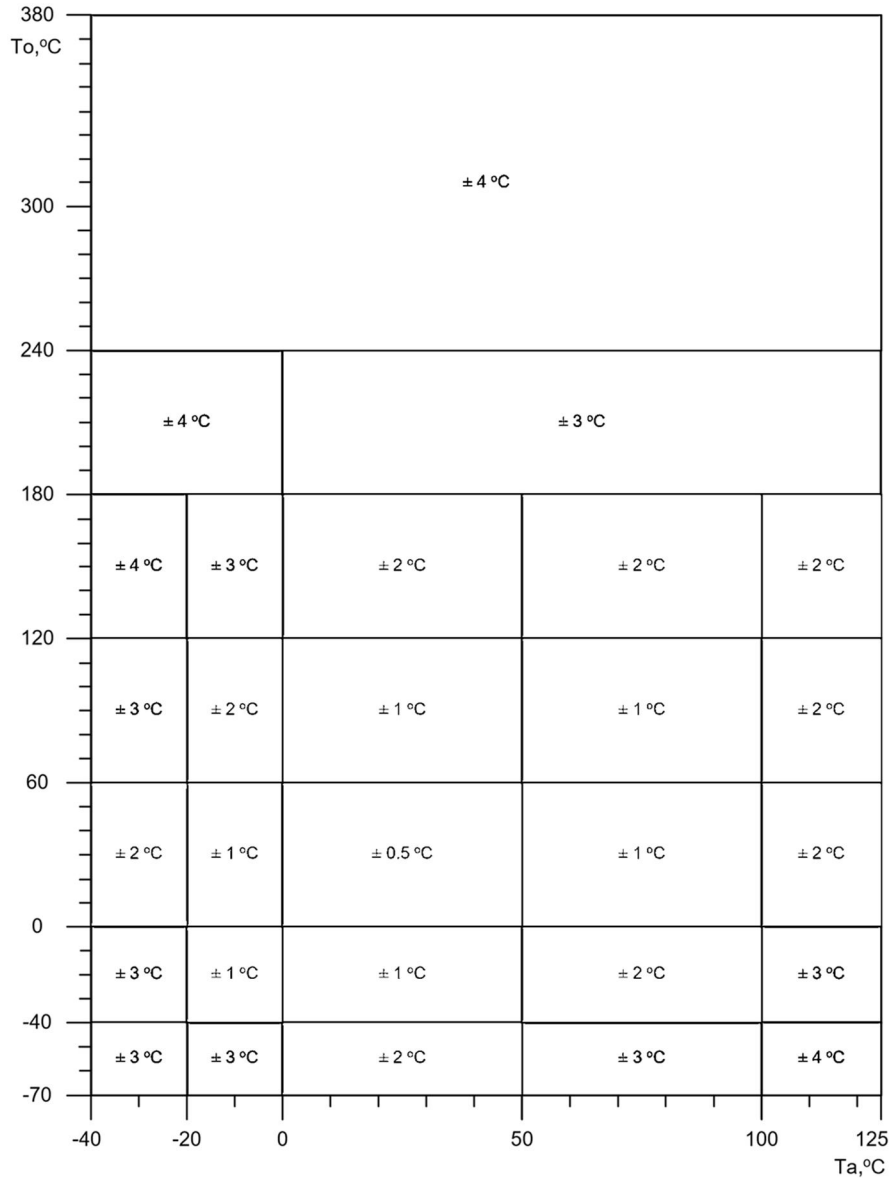


Figura 2.24 Gráfica de error de medición del MLX90614.

2.7.2. El bus i2C. [15]

El estándar I2C (Inter-Integrated Circuit) fue desarrollado por Philips en 1982 para la comunicación interna de dispositivos electrónicos en sus artículos. Posteriormente fue adoptado progresivamente por otros fabricantes hasta convertirse en un estándar del mercado.

También es denominado TWI (Two Wired Interface) únicamente por motivos de licencia. No obstante, la patente caducó en 2006, por lo que actualmente no hay restricción sobre el uso del término I2C. Requiere únicamente dos cables para su funcionamiento, uno para la señal de reloj (CLK) y otro para el envío de datos (SDA), lo cual es una ventaja frente al bus SPI. Por contra, su funcionamiento es un poco más complejo, así como la electrónica necesaria para implementarla.

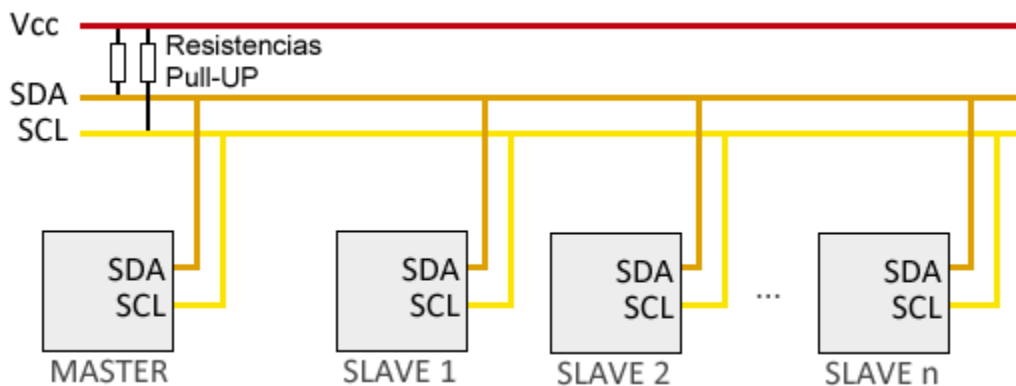


Figura 2.25 Diagrama de conexión del protocolo I2C.

En el bus Cada dispositivo dispone de una dirección, que se emplea para acceder a los dispositivos de forma individual. Esta dirección puede ser fijada por hardware (en cuyo caso, frecuentemente, se pueden modificar los últimos 3 bits mediante jumpers o interruptores) o totalmente por software. Cada dispositivo conectado al bus debe tener una dirección única. Si se tiene varios dispositivos similares tendremos que cambiar la dirección o, en caso de no ser posible, implementar un bus secundario.

Tiene una arquitectura de tipo maestro-esclavo, en el cual el dispositivo maestro inicia la comunicación con los esclavos, y puede mandar o recibir datos de los esclavos. Por el contrario los esclavos no pueden iniciar la comunicación ni hablar entre si directamente. Es posible disponer de más de un maestro, pero solo uno puede ser maestro cada vez. El cambio de maestro supone una alta complejidad, por lo que no es algo frecuente.

El bus I2C es síncrono. El maestro proporciona una señal de reloj que mantiene sincronizados a todos los dispositivos del bus. De esta forma, se elimina la necesidad de que cada dispositivo tenga su propio reloj, de tener que acordar una velocidad de transmisión y mecanismos para mantener la transmisión sincronizada (como en UART). El protocolo I2C prevé resistencias de Pull-UP de las líneas a Vcc. En Arduino frecuentemente no se instalan estas resistencias, ya que la librería Wire activa las resistencias internas de Pull-UP. Sin embargo las resistencias internas tienen un valor de entre 20-30kOhmios, por lo que son unas resistencias de Pull-UP muy blandas.

Usar resistencias blandas implica que los flancos de subida de la señal serán menos rápidas, lo que implica que se podrán usar velocidades menores y distancias de comunicación inferiores. Si se quieren emplear velocidades o distancias de transmisión superiores, se deberá colocar físicamente resistencias de Pull-UP de entre 1K a 4.7K.

2.7.2.1. Funcionamiento.

Para poder realizar la comunicación con solo un cable de datos, el bus I2C emplea una trama (el formato de los datos enviados) amplia. La comunicación consta de:

- 7 bits a la dirección del dispositivo esclavo con el que se quiere comunicar.
- Un bit restante que indica si se quiere enviar o recibir información.
- Un bit de validación
- Uno o más bytes son los datos enviados o recibidos del esclavo.
- Un bit de validación

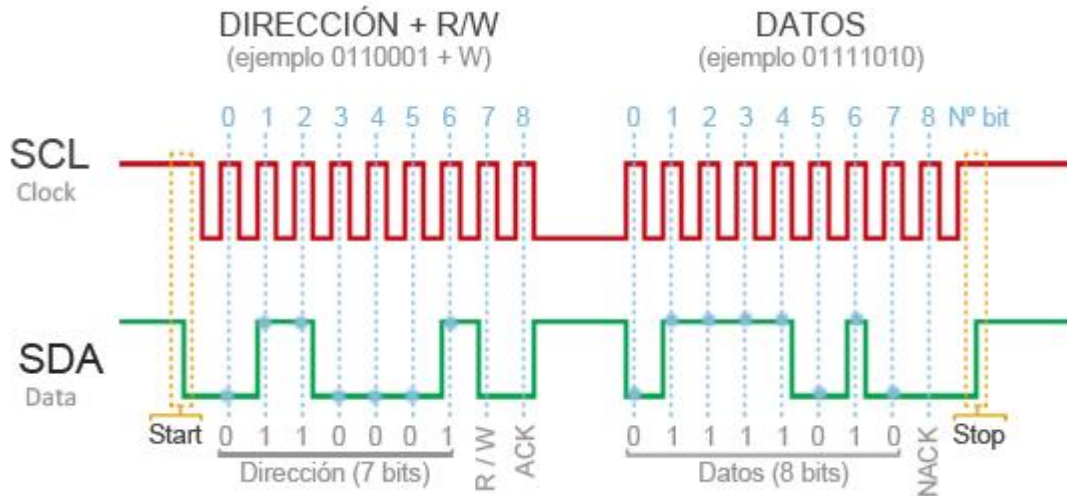


Figura 2.26 Funcionamiento del protocolo I2C.

Con estos 7 bits de dirección es posible acceder a 112 dispositivos en un mismo bus (16 direcciones de las 128 direcciones posibles están reservadas para usos especiales). Este incremento de los datos enviados (18bits por cada 8bits de datos) supone que, en general, la velocidad del bus I2C es reducida. La velocidad estándar de transmisión es de 100Mhz, con un modo de alta velocidad de 400Mhz.

El estándar I2C define otros modos de funcionamiento, como un envío de dirección de 8,10 y 12bits, o velocidades de transmisión de 1Mbit/s, 3.4Mbit/s y 5Mbit/s. No suelen ser empleados en Arduino.

2.7.2.2. Ventajas y desventajas del protocolo i2C

VENTAJAS

- Requiere pocos cables
- Dispone de mecanismos para verificar que la señal hay llegado

DESVENTAJAS

- Su velocidad es media-baja
- No es full duplex
- No hay verificación de que el contenido del mensaje es correcto

Arduino dispone de soporte I2C por hardware vinculado físicamente a ciertos pines. También es posible emplear cualquier otro grupo de pines como bus I2C a través de software, pero en ese caso la velocidad será mucho menor. Los pines a los que está asociado varían de un modelo a otro. La siguiente tabla 2.3 muestra la disposición en alguno de los principales modelos.

Tabla 2.3 Entradas para la comunicación I2C por arduino.

Modelo	SDA	SCL
Uno	A4	A5
Nano	A4	A5
Mini Pro	A4	A5
Mega	20	21

2.8. SERIE DE FOURIER EN FORMA COMPLEJA. [16]

Sea f una función periódica con periodo T , la cual puede desarrollarse en serie de Fourier en la forma

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 t + b_n \operatorname{sen} n\omega_0 t) \quad (2.12)$$

Con

$$a_0 = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) dt$$

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \cos n\omega_0 t dt$$

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \operatorname{sen} n\omega_0 t \, dt$$

Donde $\omega_0 = \frac{2\pi}{T}$

Y a_0 , a_n y b_n con $n = 1, 2, \dots$ son constantes.

Se sabe que,

$$e^{j\theta} = \cos \theta + j \operatorname{sen} \theta$$

$$e^{-j\theta} = \cos \theta - j \operatorname{sen} \theta$$

$$\cos \theta = \frac{1}{2}(e^{j\theta} + e^{-j\theta})$$

$$\operatorname{sen} \theta = \frac{1}{j2}(e^{j\theta} - e^{-j\theta}) = -j\frac{1}{2}(e^{j\theta} - e^{-j\theta})$$

Caso particular:

$$\theta = n\omega_0 t$$

$$\cos n\omega_0 t = \frac{1}{2}(e^{jn\omega_0 t} + e^{-jn\omega_0 t})$$

(2.13)

Y

$$\operatorname{sen} n\omega_0 t = -j\frac{1}{2}(e^{jn\omega_0 t} - e^{-jn\omega_0 t})$$

(2.14)

Se reemplaza (2.13) y (2.14) en (2.12), se tiene

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \frac{1}{2}(e^{jn\omega_0 t} + e^{-jn\omega_0 t}) - b_n j \frac{1}{2}(e^{jn\omega_0 t} - e^{-jn\omega_0 t}) \right)$$

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(\frac{1}{2}(a_n - jb_n)e^{jn\omega_0 t} + \frac{1}{2}(a_n + jb_n)e^{-jn\omega_0 t} \right)$$

Se hace $c_0 = \frac{a_0}{2}$; $c_{-n} = \frac{1}{2}(a_n + jb_n)$; $c_n = \frac{1}{2}(a_n - jb_n) = \overline{c_{-n}}$

$$f(t) = c_0 + \sum_{n=1}^{\infty} (c_n e^{jn\omega_0 t} + c_{-n} e^{-jn\omega_0 t})$$

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t}$$

(2.15)

2.8.1. Desarrollo en serie de Fourier de $f(t)$ en forma compleja. [16]

$$c_n = \frac{1}{2}(a_n - jb_n) = c_n = \frac{1}{2} \left(\frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \cos n\omega_0 t dt - j \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \sen n\omega_0 t dt \right)$$

$$c_n = \frac{1}{T} \left[\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) (\cos n\omega_0 t - j \sen n\omega_0 t) dt \right] = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-jn\omega_0 t} dt$$

$$c_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-jn\omega_0 t} dt$$

(2.16)

A los coeficientes a_n y b_n se les llama coeficientes de Euler-Fourier.

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t}$$

$$f(t) = \sum_{n \in \mathbb{Z}} c_n e^{jn\omega_0 t}$$

“Desarrollo en serie de Fourier de $f(t)$ en forma compleja”

Con

$$c_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-jn\omega_0 t} dt$$

$$f(t) = c_0 + \sum_{n=1}^{\infty} (c_n e^{jn\omega_0 t} + c_{-n} e^{-jn\omega_0 t})$$

$$c_{-n} = |c_{-n}| e^{j\angle c_{-n}} = |c_n| e^{-j\angle c_n}; c_n = \overline{c_{-n}} = |c_{-n}| e^{-j\angle c_{-n}} = |c_n| e^{j\angle c_n}$$

$$\begin{aligned} f(t) &= c_0 + \sum_{n=1}^{\infty} (|c_n| e^{j\angle c_n} e^{jn\omega_0 t} + |c_n| e^{-j\angle c_n} e^{-jn\omega_0 t}) \\ &= c_0 + \sum_{n=1}^{\infty} |c_n| (e^{j(n\omega_0 t + \phi_n)} + e^{-j(n\omega_0 t + \phi_n)}) \\ &= c_0 + \sum_{n=1}^{\infty} |c_n| 2 \cos(n\omega_0 t + \phi_n) \end{aligned}$$

Donde $\phi_n = \angle c_n$

$$f(t) = c_0 + 2 \sum_{n=1}^{\infty} |c_n| \cos(n\omega_0 t + \phi_n)$$

Con $\phi_n = \angle c_n$

$$\begin{aligned} f(t) &= c_0 + 2 \sum_{n=1}^{\infty} |c_n| \cos(n\omega_0 t + \phi_n) \\ c_n &= \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-jn\omega_0 t} dt \end{aligned}$$

Con $|c_n| = \frac{1}{2} |a_n - jb_n|$; $\phi_n = \angle c_n$

$$c_0 = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-j0\omega_0 t} dt = |c_0| \cos \phi_0$$

La amplitud de las componentes armónicas y de frecuencia fundamental es del doble de las calculadas a partir de $|c_n|$. La magnitud y signo de la componente “de corriente directa” es el correcto.

2.8.2. Transformada Discreta de Fourier. [17]

Considere una señal discreta $x(k)$ con N muestras de la forma

$$x_0, x_1, x_2, \dots, x_k, \dots, x_{N-1}$$

Donde x es un número complejo

$$x_i = x_{real} + jx_{imag}.$$

Suponga, que la serie fuera del rango $[0, N - 1]$ es periódica extendida con periodo N , es decir que $x_k = x_{k+N}$ para todo k . La transformada Discreta de Fourier de esta serie denotada por $X(k)$, también tendrá N muestras. La transformada (directa) de Fourier se define como:

$$X(n) = \frac{2}{N} \sum_{k=0}^{N-1} x(k) e^{-j\frac{2\pi nk}{N}}, \text{ para } n = 0, 1, 2, \dots, N - 1 \quad (2.17)$$

Aplicando la fórmula de Euler:

$$X(n) = \frac{2}{N} \sum_{k=0}^{N-1} x(k) \left(\cos \frac{2\pi nk}{N} - j \operatorname{sen} \frac{2\pi nk}{N} \right), \text{ para } n = 0, 1, 2, \dots, N - 1 \quad (2.18)$$

Aunque la función x es una serie compleja, las señales muestreadas reales pueden ser representadas mediante hacer la parte imaginaria igual a 0. En general, la transformada en el dominio de la frecuencia será una función compleja, es decir, con magnitud y fase.

$$magnitud = |X(n)| = \sqrt{x_{real}^2 + x_{imag}^2} \quad (2.19)$$

Para obtener la fase se usa la siguiente ecuación:

$$fase = \tan^{-1} \left(\frac{x_{imag}}{x_{real}} \right) * \quad (2.20)$$

La magnitud de la primera muestra $X(0)$ es el doble del valor promedio absoluto de la señal $x(n)$ también conocida como la “componente de corriente directa”. La Transformada Discreta de Fourier de una serie de muestras reales, resulta en una secuencia simétrica en torno a la frecuencia de Nyquist.

La frecuencia máxima posible se le llama frecuencia de Nyquist. Esta es la componente de frecuencia máxima que debería estar presente en la serie $x(k)$ a la cual se le aplica la transformada con el fin de obtener resultados correctos (sin contaminación).

2.8.3. Uso de los coeficientes de Fourier para calcular el valor RMS de una señal periódica. [18]

Para calcular los valores RMS de una señal periódica, siempre se seleccionan varios métodos como la integración, la regla de Simpson y la regla trapezoidal. Existe evidencia académica que demuestra que los valores RMS de una señal periódica también pueden ser obtenidos mediante el uso de los Coeficientes de Fourier.

2.8.3.1. Cálculo de valores RMS.

Aunque hay varios métodos para calcular los valores RMS, como ya se había mencionado antes, el método estándar utilizado es la técnica de Cálculo de integración la cual establece que:

$$f_{RMS} = \sqrt{\frac{1}{T} \int_{t_1}^{t_1+T} f^2(t) dt} \quad (2.21)$$

Donde:

f_{RMS} es el valor RMS de $f(t)$ entre el intervalo de dominio de t_1 a $t_1 + T$
 T es el periodo de $f(t)$.

2.8.3.2. Cálculos de los valores RMS de una señal periódica construida por series de Fourier.

Una señal periódica con periodo de $T = 2\pi$ puede expandirse en términos de la serie de Fourier como se muestra en la ecuación (2.1):

$$f(t) = a_0 + \sum_{n=1}^N (a_n \cos nt + b_n \sen nt) \quad (2.22)$$

$$= (a_0 + a_1 \cos t + a_2 \cos 2t + \dots + a_N \cos Nt) + (b_1 \sen t + b_2 \sen 2t + \dots + b_n \sen Nt)$$

Por lo tanto $f^2(t)$ puede expandirse de la siguiente manera:

$$\begin{aligned} f^2(t) = & [(a_0)^2 + (a_1 \cos t)^2 + (a_2 \cos 2t)^2 + \dots + (a_N \cos Nt)^2 + (b_1 \sen t)^2 + (b_2 \sen 2t)^2 \\ & + \dots + (b_N \sen Nt)^2 + 2(a_0 a_1 \cos t + \dots + a_M b_M \cos Nt \sen Nt \\ & + \dots + b_{N-1} b_N \sen(N-1t) \sen Nt)] \end{aligned}$$

Después, se define una integral entre el periodo (0 a 2π) de $f^2(t)$.

$$\int_0^{2\pi} f^2(t) dt = \int_0^{2\pi} [(a_0)^2 + (a_1 \cos t)^2 + (a_2 \cos 2t)^2 + \dots + (a_N \cos Nt)^2 + (b_1 \sin t)^2 + (b_2 \sin 2t)^2 + \dots + (b_N \sin Nt)^2 + 2(a_0 a_1 \cos t + \dots + a_M b_M \cos Nt \sin Nt + \dots + b_{N-1} b_N \sin(N-1t) \sin Nt)] dt \quad (2.23)$$

Al usar propiedades ortogonales de funciones sinusoidales, el resultado de la integración entre sus propios periodos será igual a cero o a π . Del razonamiento anterior algunos términos de la ecuación (2.25) serán eliminados, pero todos los coeficientes de Fourier elevados al cuadrado seguirán en la ecuación (2.26).

$$\int_0^{2\pi} f^2(t) dt = 2\pi(a_0)^2 + \pi(a_1)^2 + \pi(a_2)^2 + \dots + \pi(a_N)^2 + \dots + \pi(b_1)^2 + \pi(b_2)^2 + \dots + \pi(b_N)^2 \quad (2.24)$$

Si es necesario saber el valor RMS de la función periódica $f(t)$ todos los coeficientes de Fourier deben añadirse al cálculo como se muestra en la ecuación (2.27).

$$f_{RMS} = \sqrt{\frac{1}{2\pi} \int_0^{2\pi} f^2(t) dt} = \sqrt{(a_0)^2 + \frac{(a_1)^2 + (a_2)^2 + \dots + (a_N)^2 + \dots + (b_1)^2 + (b_2)^2 + \dots + (b_N)^2}{2}} \quad (2.25)$$

Esto indica que el valor RMS de una señal periódica no depende de las frecuencias armónicas. A continuación se muestra un ejemplo de cómo obtener el valor RMS de una señal de voltaje $v(t)$ con contenido armónico cuya ecuación es la siguiente:

$$v(t) = 7.50 - 5.00 \cos(2\omega t) + 0.500 \sin(100\omega t) + 3.00 \sin(50\omega t) + 2.50 \cos(174\omega t)$$

La cual se resolverá de la siguiente manera

$$V_{RMS} = \sqrt{(7.50)^2 + \frac{(5.00)^2 + (0.500)^2 + (3.00)^2 + (2.50)^2}{2}} \text{ volt} \approx 8.75 \text{ volts}$$

Como se pudo observar calcular el valor RMS de una señal periódica usando los coeficientes de Fourier es más conveniente que usar el método de Cálculo de integración si es que los coeficientes de Fourier se conocen. Además las diferentes frecuencias de los componentes armónicos de la señal no afectan el valor RMS. Esto quiere decir que los valores RMS de las señales periódicas no dependen de las frecuencias de las señales armónicas sino que dependen de su amplitud a las cuales se le llaman componentes de Fourier.

DISEÑO E IMPLEMENTACIÓN

En este capítulo se muestra como se construye el prototipo y como utilizando un algoritmo basado en Transformada Discreta de Fourier se obtienen los parámetros de un motor de inducción de jaula de ardilla de la marca LabVolt en conexión estrella en arranque sencillo.

3.1. CONSTRUCCIÓN DEL PROTOTIPO.

El Prototipo consta de 3 tipos de sensores, sensor de Corriente de efecto Hall, sensor de Voltaje de CA y un sensor infrarrojo de Temperatura conectados como se muestra en la figura 3.1.

Como este sistema se encarga de sensar las tres fases del motor de inducción, cuenta con un sensor de Voltaje y de Corriente por fase, ésta conexión es necesaria para poder obtener correctamente los parámetros de cada fase, ya que cada una de éstas tiene sus propias características. Cada sensor está conectado a una entrada analógica de la placa Arduino la cual se encarga convertir la información de la señal de salida que los sensores transmiten a este, por medio de una conversión de resolución para que se puedan interpretar. Por otra parte el sensor de temperatura se conecta a las entradas SCL y SDA que se encargan de la comunicación I2C la cual se vio en el capítulo anterior.

Se utiliza una pantalla LCD con una dimensión de 4x20 para mostrar los resultados de las mediciones de los parámetros en 6 diferentes pantallas, en la primera pantalla se muestra el Voltaje y la Corriente RMS de cada fase, en la segunda pantalla se muestra el Factor de potencia por fase, en la tercera se visualiza el ángulo de desfasamiento entre la onda de voltaje y la de corriente por fase, la cuarta muestra la Potencia Aparente RMS, la quinta y la sexta pantalla muestran la Potencia Activa y Reactiva RMS respectivamente. Del mismo modo es desplegada la medición de la temperatura del rotor en cada pantalla para tener un monitoreo constante de este parámetro.

Para poder navegar entre pantallas se utilizan dos PB conectados en los pines digitales 18 y 19, los cuales corresponden a interrupciones por hardware que tiene la placa arduino, cada uno con un circuito anti rebote para evitar saltos innecesarios entre pantallas y tener una navegación más eficiente.

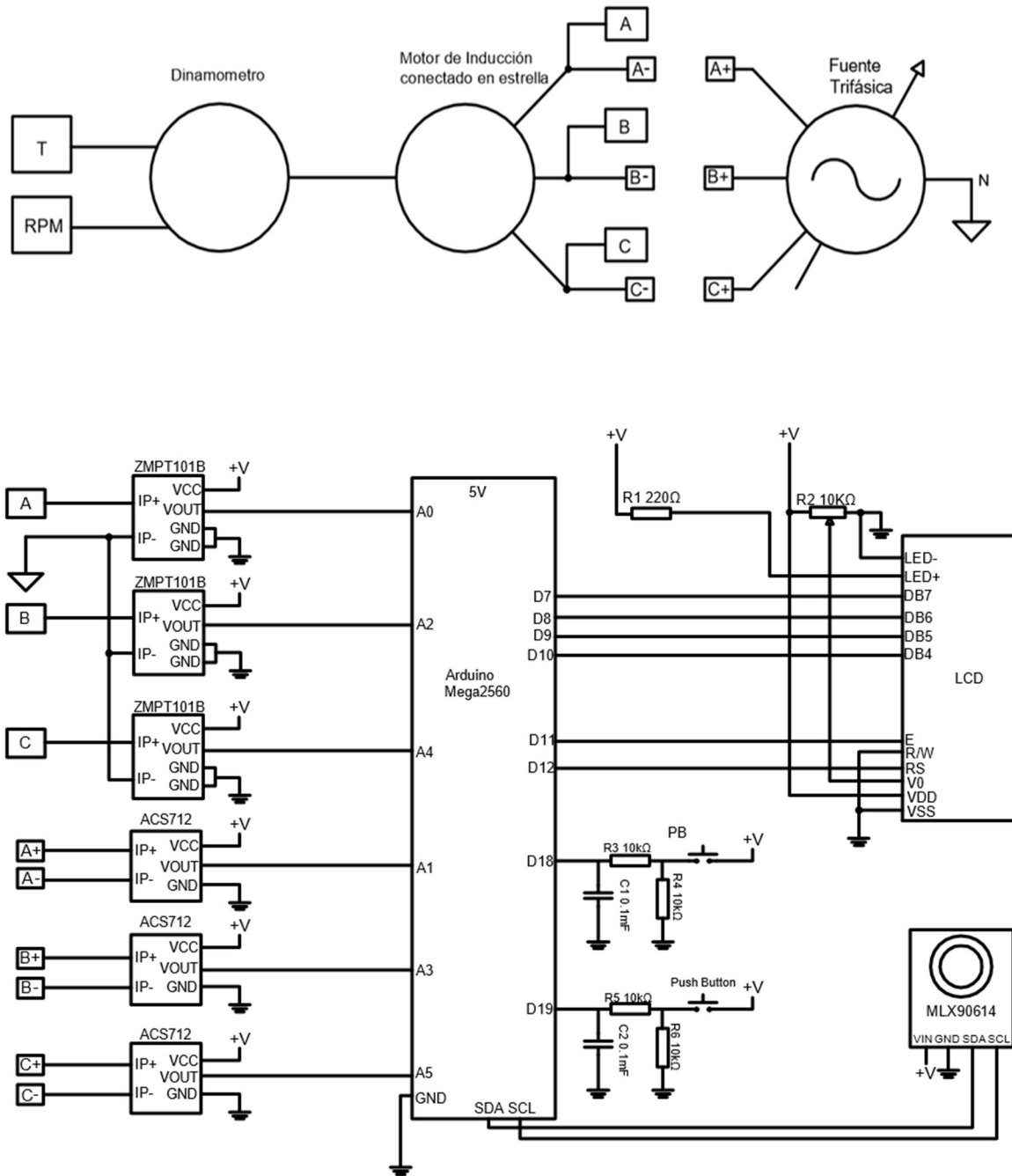


Figura 3.1 Diagrama de conexión del prototipo.

3.2. PROGRAMACIÓN DEL ARDUINO.

Para programar el Arduino se utilizó la plataforma Arduino IDE la cual es un software libre, de fácil instalación, amigable con el usuario, fácil de entender y gratuito, para crear programas o “Sketches”. Los sketches constan de 3 partes principales, una parte en donde se hacen comentarios se declaraba variables globales o se incluyen las librerías, incluso pueden crearse funciones (comúnmente van al final de los sketches). La siguiente parte del sketch es el “setup”, el cual solo se corre una vez cuando se inicia la placa arduino y donde, como su nombre lo dice, se declara un set up ósea se declaran pines como entradas o salidas digitales, se inicializa la comunicación en serie o por otros protocolos, se declaran interrupciones, etc. y se declara con la función void setup(). La última parte del sketch es la parte que se estará corriendo en un loop, aquí es donde se desarrolla la mayor parte del sketch, es donde se programan las actividades que realizará el Arduino y como procesará la información que se le aportará. Ésta parte está denominada por la instrucción void loop().

Los pasos que se siguieron para la realización del sketch fueron los siguientes:

1. Se incluyen las librerías necesarias para utilizar los dispositivos con los que se comunicará la placa de Arduino y declaran las variables globales que se utilizarán en el programa.
2. En la sección de void setup() se inicializa la comunicación serial, la comunicación con el sensor de temperatura MLX90614, se especifica las dimensiones de la LCD y se definen las interrupciones que se utilizarán.
3. En el bucle del programa se procede a llamar a una función para obtener los valores RMS de la corriente y voltaje a neutro por fase del motor e imprimirlos en una LCD. También se obtiene el dato de temperatura del sensor MLX90614.
4. Se crea una función basada en la transformada discreta de Fourier para obtener las componentes de frecuencia de Voltajes y Corrientes por fase, además de utilizar cálculos basados en estas dos variables para obtener la potencia activa, la potencia reactiva, la potencia aparente, el factor de potencia y el ángulo de fase.
5. Por último se crean dos interrupciones para la transición de una pantalla a otra.

3.2.1. Medición de Temperatura.

La comunicación con el sensor de temperatura se lleva a cabo mediante el protocolo I2C el cual permite utilizar diferentes sistemas esclavos conectados en un solo bus de comunicación. Dado que el resultado de la medición es provisto de forma digital, el ruido del ambiente no afecta el valor de la medición.

La comunicación con el sensor se logra mediante la librería 'i2cmaster.h' (Peter Fleury, 2005) la cual es de uso libre, se utiliza una dirección única para comunicarse con el Arduino la cual es accedida por medio de una instrucción en la librería 'i2cmaster.h' y así adquirir el dato de la medición del sensor.

Debido a que este dispositivo arroja los resultados en grados Kelvin, es necesario hacer una conversión digital ($\text{Grados Celsius} = \text{°K} - 273.15$).

3.2.2. Funcionamiento del algoritmo basado en la T.D.F.

La función sigue el algoritmo descrito por la T.D.F., el prototipo obtiene muestras de las corrientes y voltajes instantáneas de las fases del motor con un cierto periodo de muestreo uniforme. Se utiliza un ciclo "for" para poder hacer los cálculos necesarios individualmente para la parte real y la parte imaginaria de las sumatorias indicadas en (2.19).

Utilizando una estructura de control de flujo (ciclo while) se hace un muestreo uniforme de la señal de Voltaje y Corriente el cual consta de 40 muestras por ciclo con una ventana de tiempo de 416.66 micro segundos entre muestras.

Para asegurar que las muestras se tomen de manera uniforme se utiliza una estructura condicional 'if()' la cual permite que se tomen muestras solo si ha transcurrido el tiempo T_s como se muestra en la siguiente sección del código.

```
if (micros() - microsAnt >= Ts)
```

{...}

Donde 'micros()' es una función que devuelve el tiempo que ha transcurrido desde que se inició el sistema, 'microsAnt' es una variable que almacena el tiempo en que se tomó la última muestra y Ts es el periodo de muestreo en unidades de μs .

Una vez que se haya completado el muestreo de las señales, se procede a calcular las componentes de frecuencia en magnitud y fase utilizando la ecuación (2.19), pero solo la fundamental, la 3^a, 5^a y 7^a armónica ya que éstas son las que se encuentran más habitualmente en la red eléctrica industrial y también debido a que la velocidad de procesamiento del Arduino se ve comprometida con más cálculos matemáticos.[19]

El siguiente paso es calcular los valores RMS de el Voltaje y la Corriente utilizando la fórmula (2.25), las potencias activa y reactiva se obtuvo multiplicando las magnitudes de Corriente y Voltaje y a su vez multiplicarlas por el coseno y seno de las fases entre éstas respectivamente. Una vez obtenida la potencia Activa y pasiva se procede a calcular la potencia aparente y el factor de potencia. En la figura (3.2) se muestra de forma general el funcionamiento del programa con un diagrama de flujo.

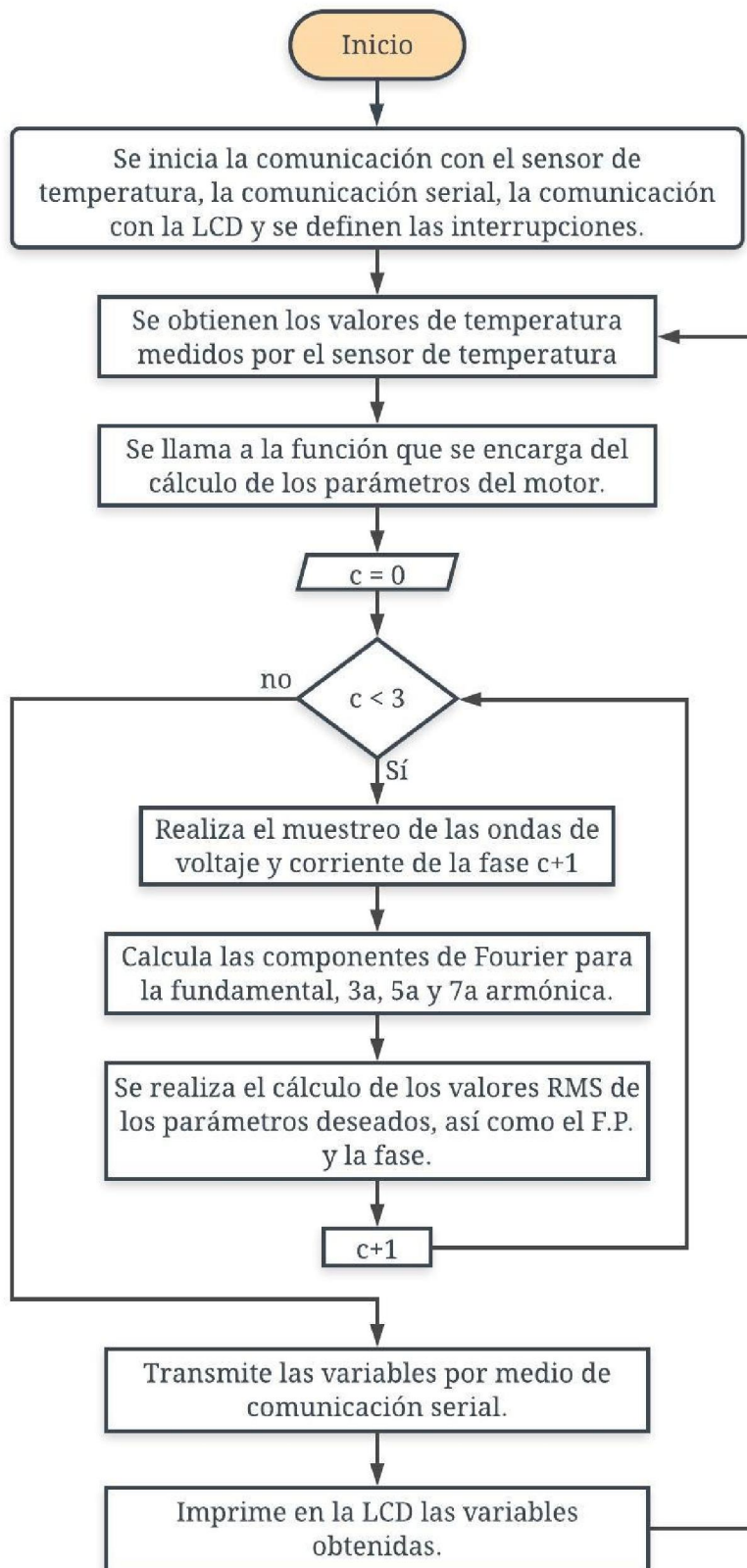


Figura 3.2 Diagrama de flujo de funcionamiento del prototipo.

3.2.3. Transmisión de datos obtenidos a la plataforma MATLAB® 2015^a.

Debido a que es necesario tener un historial para una mejor apreciación del comportamiento de los parámetros de las fases del motor de inducción, se utilizó la herramienta Simulink de la plataforma MATLAB® 2015a y la habilidad de la placa Arduino Mega 2560 para comunicarse de forma serial con el ordenador al que está conectado, para crear graficas sobre el comportamiento de estos. Para lograr esto se creó un programa en Simulink como se muestra en la fig (3.3) con el bloque Query Instrument el cual permite crear una comunicación serial entre MATLAB® 2015a y la placa Arduino. Este bloque se puede configurar con el puerto que se usará, la velocidad de transmisión, la cantidad de datos que se mandarán y su tipo (flotante o entero) y el formato de los datos.

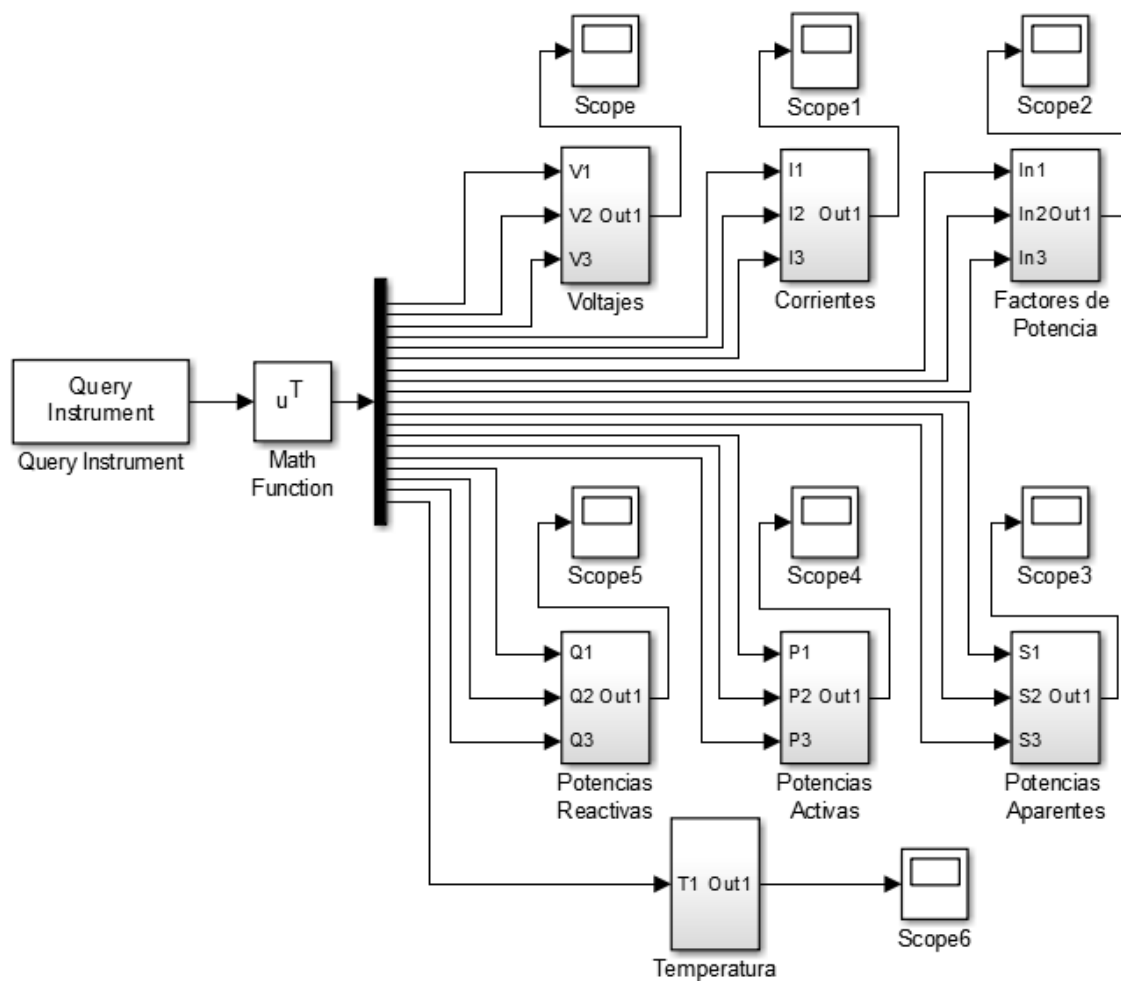


Figura 3.3 Modelo en Simulink para la comunicación entre Arduino y MATLAB.

Se utiliza un bloque de ‘transpuesta’ y un ‘demux’ para separar los datos en diferentes vías de comunicación las cuales van dirigidas a bloques y a un ‘scope’ para monitorear el comportamiento de los parámetros. Además se utiliza un bloque ‘to file’ para almacenar los datos en una extensión “.mat” y displays para poder monitorear los valores RMS de las variables como se muestra en la fig. (3.4). Por ultimo para poder obtener los datos y poderlos graficar en MATLAB® 2015a se hace uso de un pequeño programa el cual extrae los datos “.mat” y crea una gráfica con ellos.

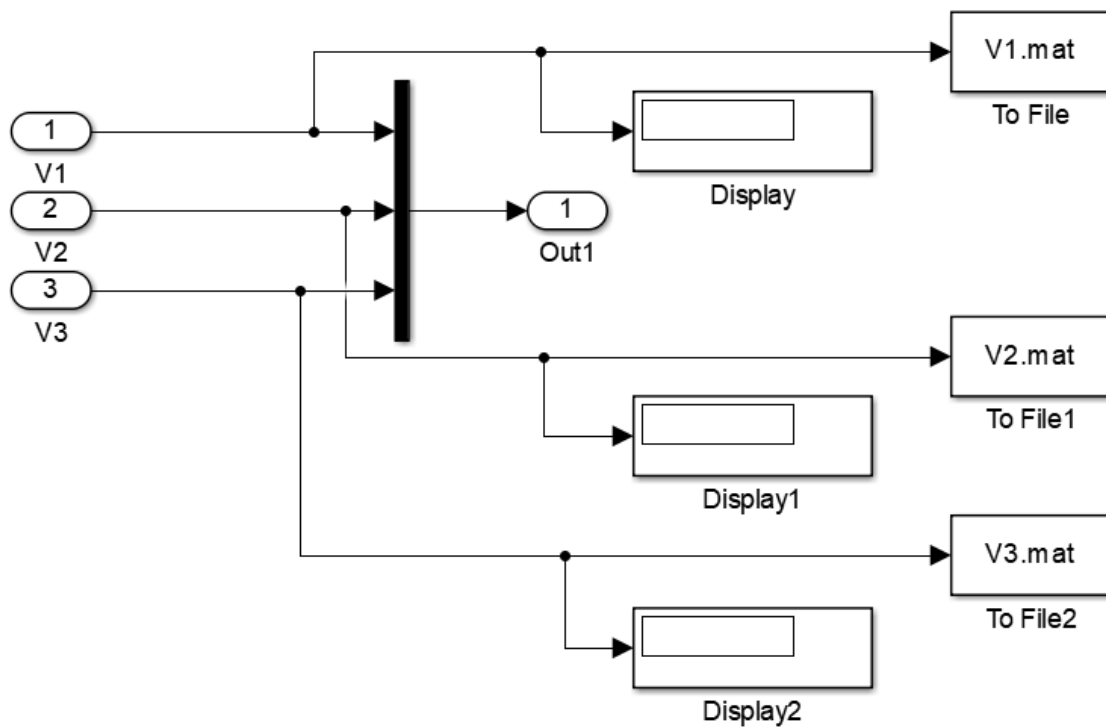


Figura 3.4 Bloque submódulo del programa en Simulink.

Capítulo

4

PRUEBAS Y RESULTADOS

En este capítulo se muestran las pruebas y los resultados que se le hicieron al prototipo utilizando el motor de inducción y un dinamómetro como carga y se analizan los resultados.

4.1. PRUBA CON MOTOR EN VACIO.

Al hacer pruebas con el prototipo se hizo una conexión en estrella de arranque sencillo al motor de inducción, los sensores de efecto hall se conectaron en serie con la línea del motor, los sensores de voltaje se conectaron en paralelo a la fuente de CA.

Al encender el motor se pudo observar que se registraban los picos de corriente que el motor tiene al encenderse de forma sencilla a voltaje pleno. Se utilizó un analizador de redes FLUKE para medir las corrientes de fase, se procedió a comparar con los valores que el prototipo arrojaba y se comprobó que los valores obtenidos por el prototipo y el analizador de redes eran parecidos como se muestra a continuación.



Figura 4.1 Conexión del prototipo y el analizador de redes FLUKE para pruebas con el motor.

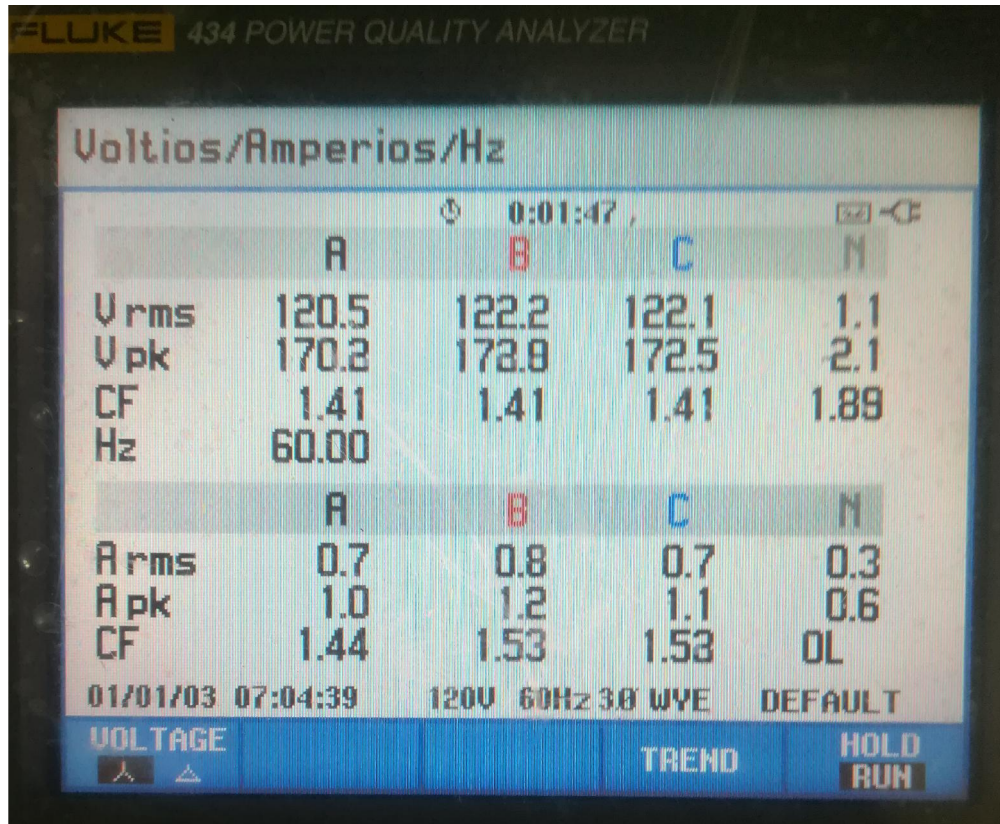


Figura 4.2 Resultados de las mediciones de voltaje y corriente tomados por el analizador de redes FLUKE (en vacío).

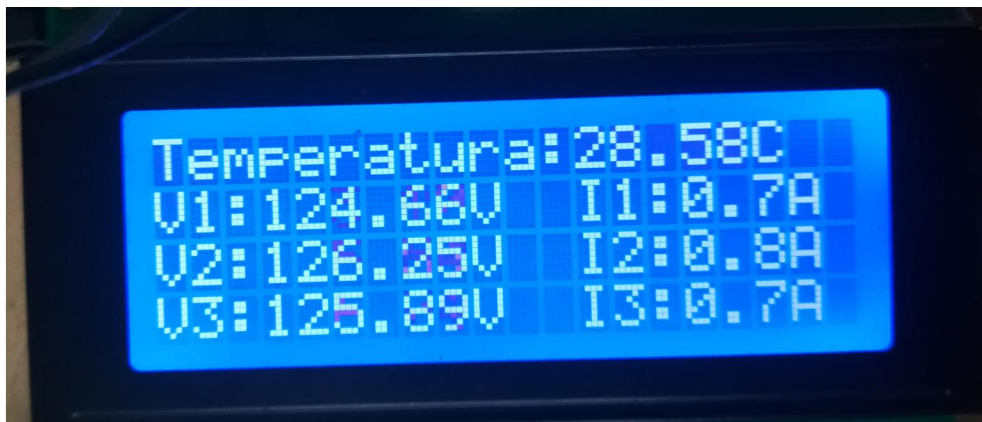


Figura 4.3 Resultados de las mediciones de voltaje y corriente tomados por el Prototipo (en vacío).

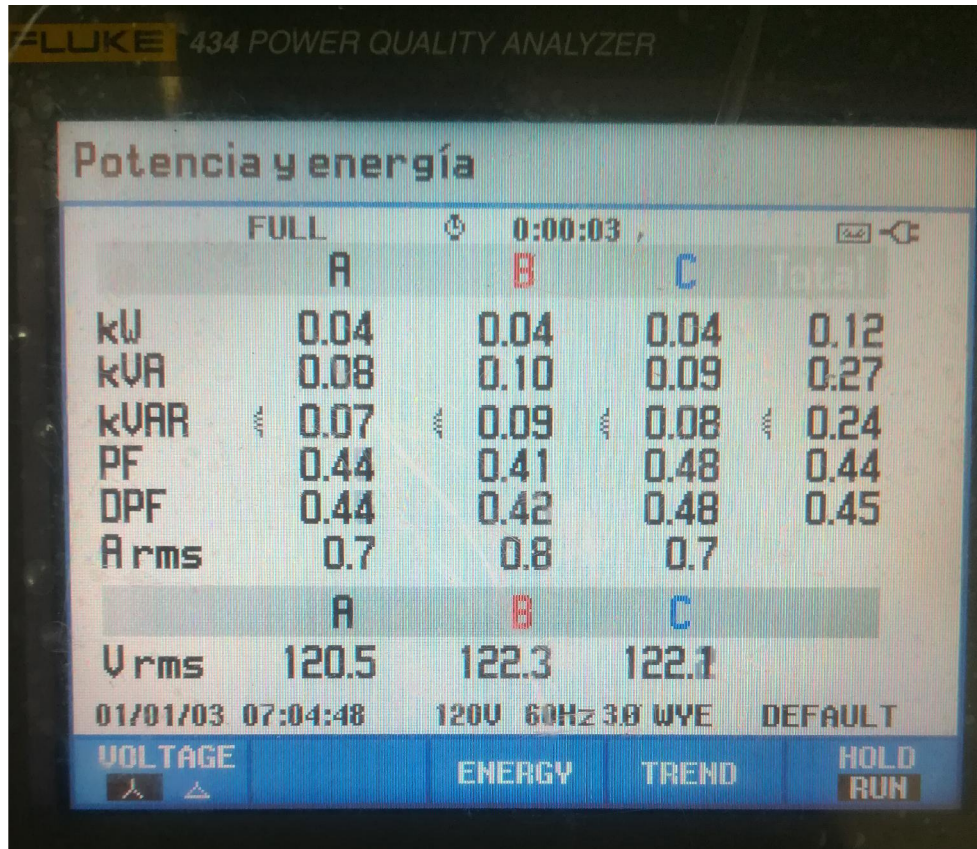


Figura 4.4 Resultados de las mediciones de potencias y factor de potencia tomados por el analizador de redes FLUKE (en vacío).

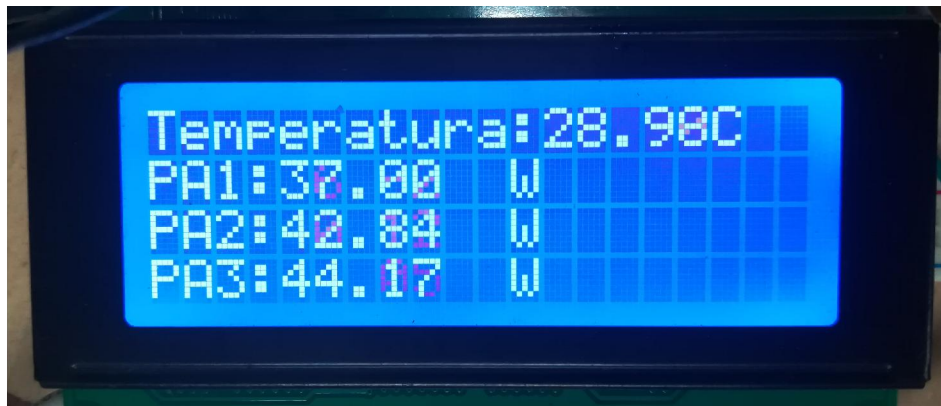


Figura 4.5 Resultados de las mediciones de potencia activa tomados por el prototipo (en vacío).



Figura 4.6 Resultados de las mediciones de potencia reactiva tomados por el prototipo (en vacío).



Figura 4.7 Resultados de las mediciones de potencia aparente tomados por el prototipo (en vacío).

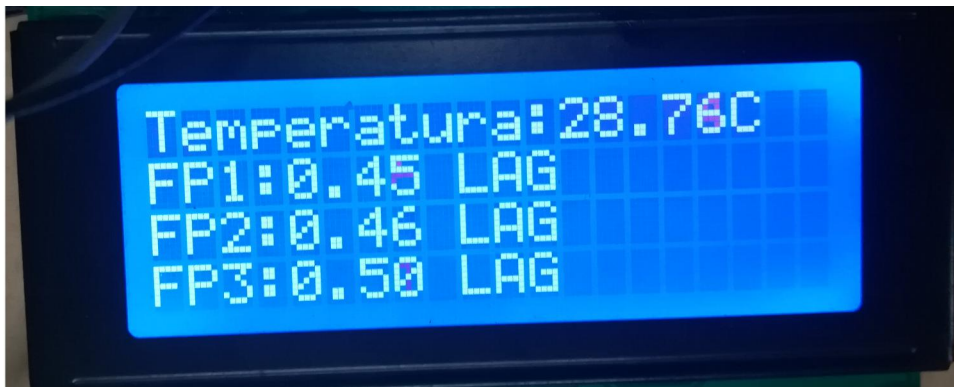


Figura 4.8 Resultados de las mediciones de Factor de Potencia tomados por el prototipo (en vacío).



Figura 4.9 Resultados de las mediciones de ángulo de fase tomados por el analizador de redes FLUKE (en vacío).

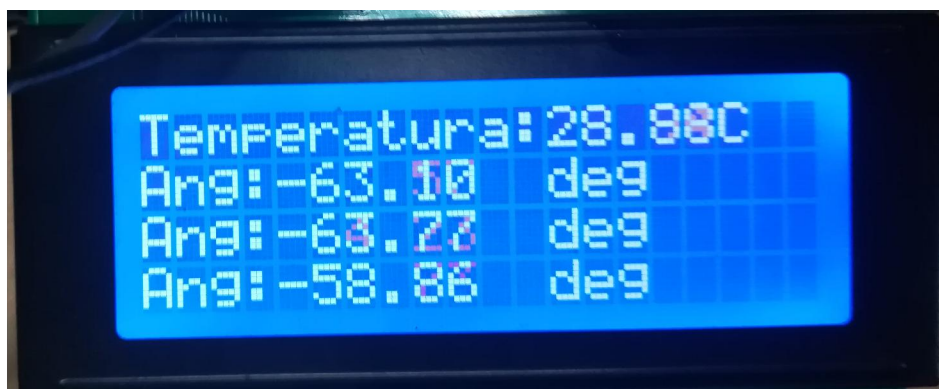


Figura 4.10 Resultados de las mediciones de ángulo de fase tomados por el prototipo (en vacío).

4.2. PRUBA CON MOTOR CON CARGA.

Esta prueba es como la anterior, con la diferencia de tener conectado un dinamómetro con el cual se llevó al motor de inducción a sus valores nominales por línea y como en la prueba anterior se procedió a tomar las mediciones de los parámetros con el analizador de redes y el prototipo. A continuación se muestran los resultados de ésta prueba.



Figura 4.11 Resultados de las mediciones de voltaje y corriente tomados por el analizador de redes FLUKE (con carga).



Figura 4.12 Resultados de las mediciones de voltaje y corriente tomados por el prototipo (con carga).

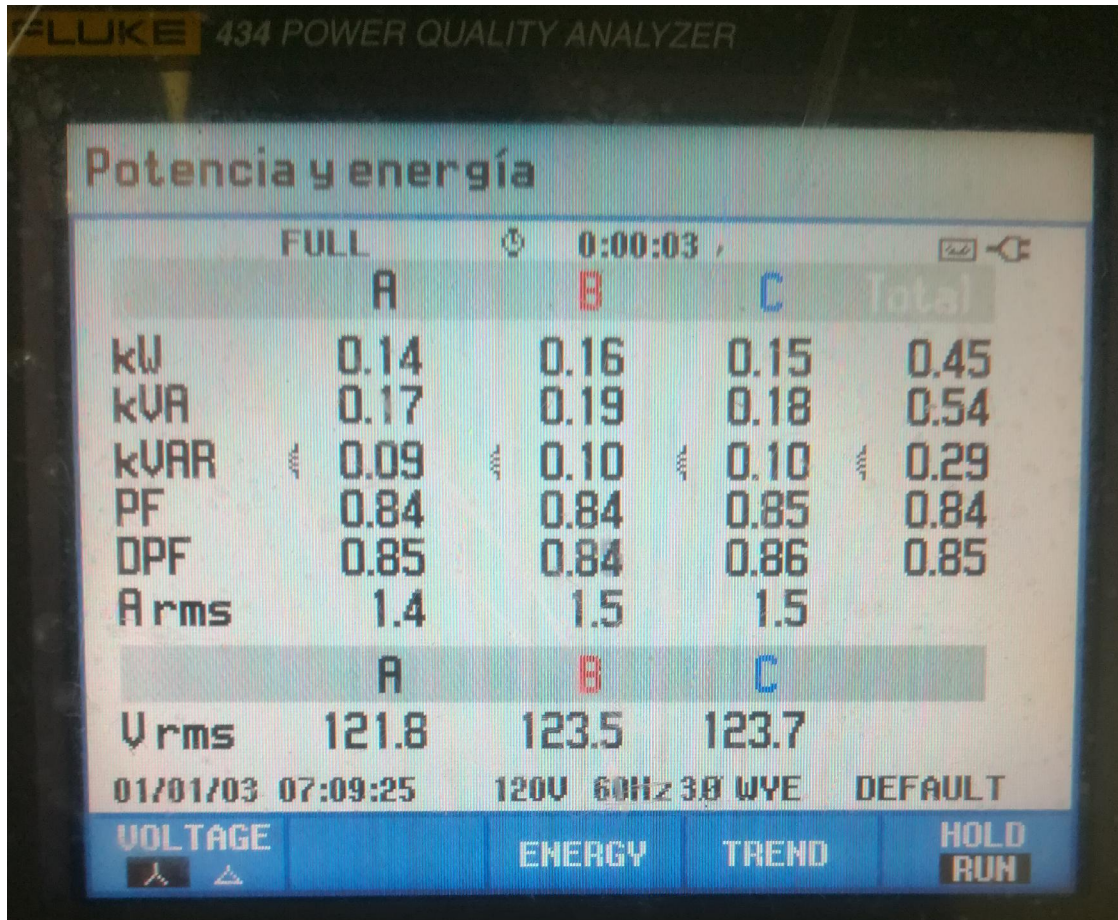


Figura 4.13 Resultados de las mediciones de potencias y factor de potencia tomados por el analizador de redes FLUKE (con carga).



Figura 4.14 Resultados de las mediciones de potencia activa tomados por el prototipo (con carga).

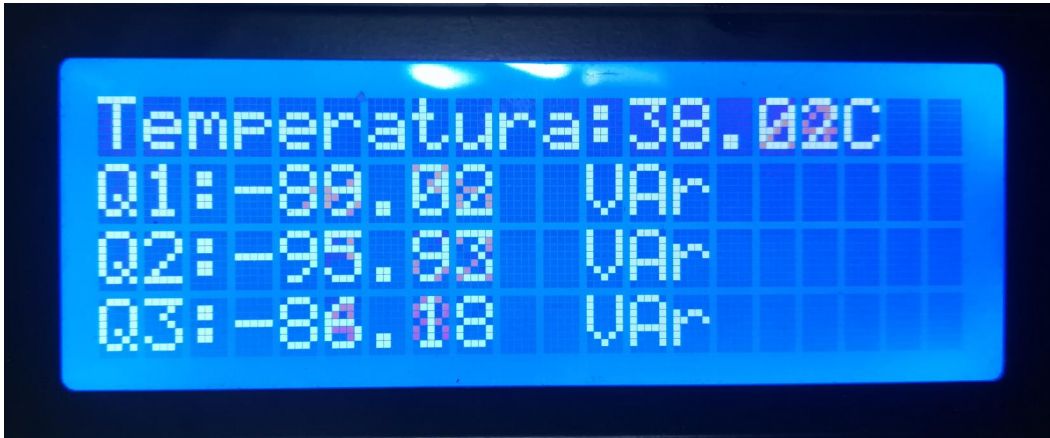


Figura 4.15 Resultados de las mediciones de potencia reactiva tomados por el prototipo (con carga).

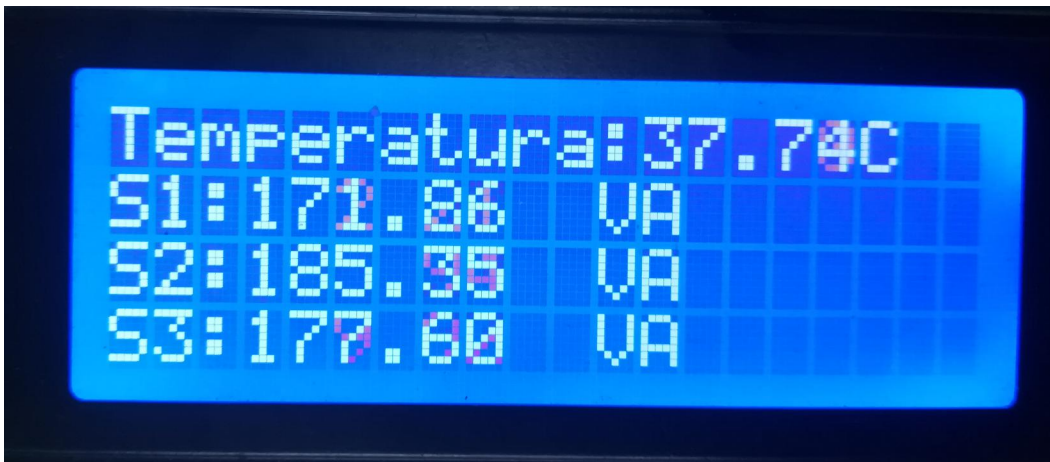


Figura 4.16 Resultados de las mediciones de potencia activa tomados por el prototipo (con carga).



Figura 4.17 Resultados de las mediciones del factor de potencia tomados por el prototipo (con carga).

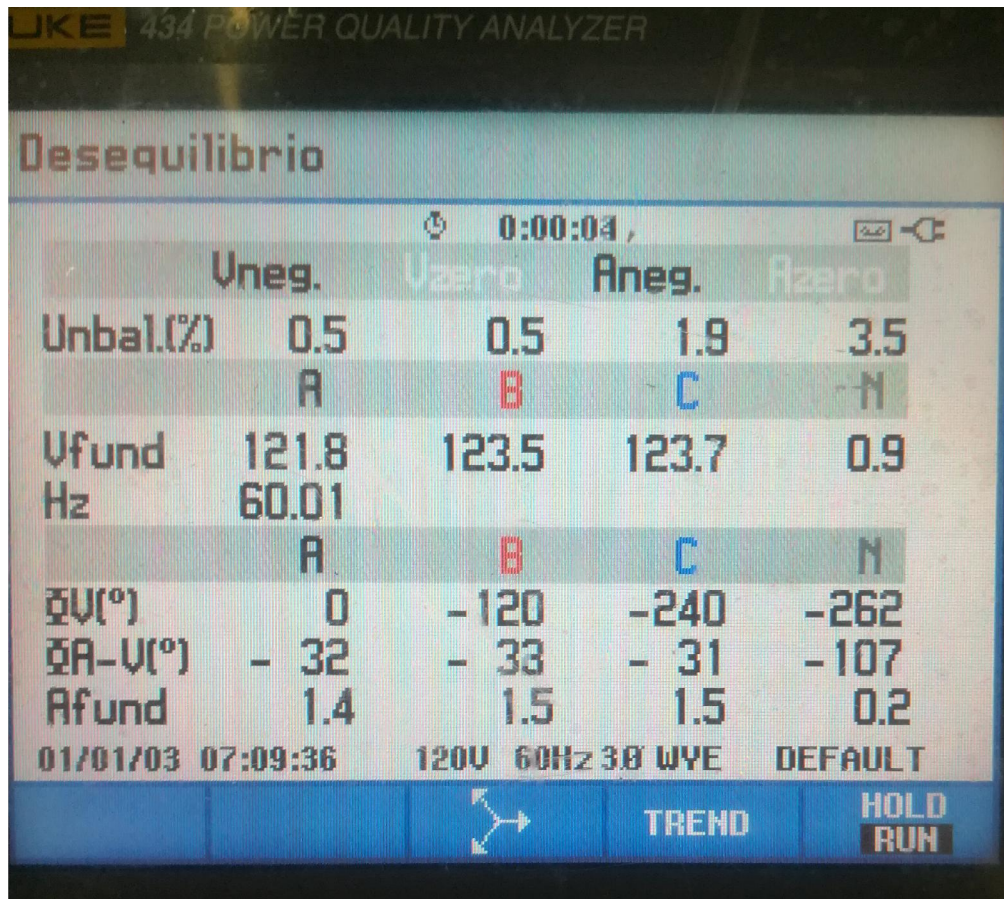


Figura 4.18 Resultados de las mediciones de ángulo de fase tomados por el analizador de redes FLUKE (con carga).

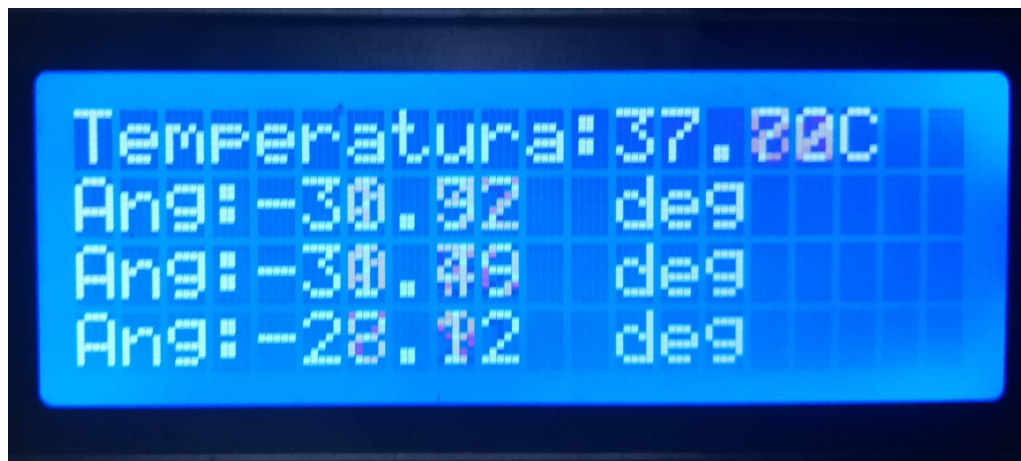


Figura 4.19 Resultados de las mediciones de ángulo de fase tomados por el prototipo (con carga).

Como se puede observar en ambas pruebas la medición de los Voltajes tiene de 3% a 5% de error (utilizando la fórmula $E\% = \frac{L_m - L_p}{L_p} * 100$) [20] entre los resultados del prototipo y el analizador de redes lo cual está dentro de los estándares de la norma IEC 600-44-1 lo cual se puede apreciar en las tablas 4.1 y 4.2. [21] Otro aspecto que se puede saber de estas pruebas es que en efecto la potencia reactiva del motor es negativa y su factor de potencia está en atraso ya que su ángulo de fase es negativo.

Tabla 4.1 Error entre mediciones del analizador de redes y el prototipo (en vacío).

Variable	Fluke	Prototipo	Error
Voltaje	120.5 V	124.66 V	3.45%
	122.2 V	126.25 V	3.31%
	122.1 V	125.89 V	3.10%
Corriente	0.7 A	0.7 A	0%
	0.8 A	0.8 A	0%
	0.7 A	0.7 A	0%
Potencia Activa	0.04 kW	37.00 W	N/A
	0.04 kW	42.84 W	N/A
	0.04 kW	44.17 W	N/A
Potencia Reactiva	-0.07 kVAr	-75.10 VAr	N/A
	-0.09 kVAr	-85.33 VAr	N/A
	-0.08 kVAr	-74.77 VAr	N/A
Potencia Aparente	0.08 kVA	85.27 VA	N/A
	0.10 kVA	96.10 VA	N/A
	0.09 kVA	87.86 VA	N/A
Factor de potencia	0.44 LAG	0.45 LAG	2.37%
	0.42 LAG	0.46 LAG	9.52%
	0.48 LAG	0.50 LAG	4.17%
Ángulo de fase	-64°	-63.10°	-1.40%
	-65°	-63.77°	-1.89%
	-61°	-58.86°	-3.50%

Tabla 4.2 Error entre mediciones del analizador de redes y el prototipo (con carga).

Variable	Fluke	Prototipo	Error (%)
Voltaje	121.7 V	126.43 V	3.74%
	123.4 V	127.05 V	2.95%
	123.6 V	128.33 V	3.82%
Corriente	1.4 A	1.4 A	0%
	1.5 A	1.5 A	0%
	1.5 A	1.4 A	6.66%
Potencia Activa	0.14 kW	146.26 W	N/A
	0.16 kW	158.62 W	N/A
	0.15 kW	155.01 W	N/A
Potencia Reactiva	-0.09 kVAr	-89.02 VAr	N/A
	-0.10 kVAr	-95.93 VAr	N/A
	-0.10 kVAr	-86.18 VAr	N/A
Potencia Aparente	0.17 kVA	171.86 VA	N/A
	0.19 kVA	185.35 VA	N/A
	0.15 kVA	177.60 VA	N/A
Factor de potencia	0.85 LAG	0.86 LAG	1.17%
	0.84 LAG	0.86 LAG	2.38%
	0.86 LAG	0.87 LAG	1.16%
Ángulo de fase	-32°	-31.92°	-0.25%
	-33°	-31.49°	-4.57%
	-31°	-28.12°	-9.29%

En las tablas se puede apreciar como algunas mediciones tienen errores de más del 5% esto se debe en parte a que los sensores de Voltaje no están calibrados de un modo correcto y a que el prototipo, como ya se mencionó antes solo obtiene los datos de la onda fundamental, la 3^a, 5^a y 7^a armónica por lo cual los resultados no son tan cercanos a los del analizador de redes. [

Otro factor que influye en el error de medición es que el analizador de redes estaba en una escala de kilos en cuanto a las potencias y el prototipo está en unidades por lo que el error de medición no se especifica en éstos parámetros, además también influyen los tiempos de procesamiento del Arduino y el número de muestras que se toman por ventana de tiempo debido a esto entre más muestras se obtenían más lento se volvía el prototipo lo que significa que sería más difícil poder obtener los datos de cambios rápidos en la red del motor..

También se puede apreciar que la temperatura aumentó 10°C en el tiempo que se realizó la prueba con el motor con carga debido al esfuerzo que realizaba el motor para mover la carga por lo cual se puede dar a conocer que el sensor funciona correctamente y se puede tener monitoreado no solo el rotor sino también otras partes del motor.

4.3. PRUEBA DE ERROR DE CONEXIÓN.

Ya que se comprobó que las mediciones son eficientes se procedió a hacer una prueba “de error de conexión” para corroborar que las fases son analizadas por separado. Lo que se hizo fue conectar el sensor de voltaje de la fase A en la fase B y viceversa, lo mismo con el analizador de redes, y los sensores de efecto Hall no fueron reubicados.

Debido a que las fases cambiaron de posición los ángulos de fase también cambiaron de valor y orientación mostrando medidas erráticas en cuanto a la potencia activa, reactiva y aparente, así como su factor de potencia.

Los resultados de esta prueba de conexión se muestran en las figuras siguientes. Al igual que la prueba anterior esta prueba se realizó con el motor en sus parámetros nominales para poder apreciar mejor el comportamiento de este.

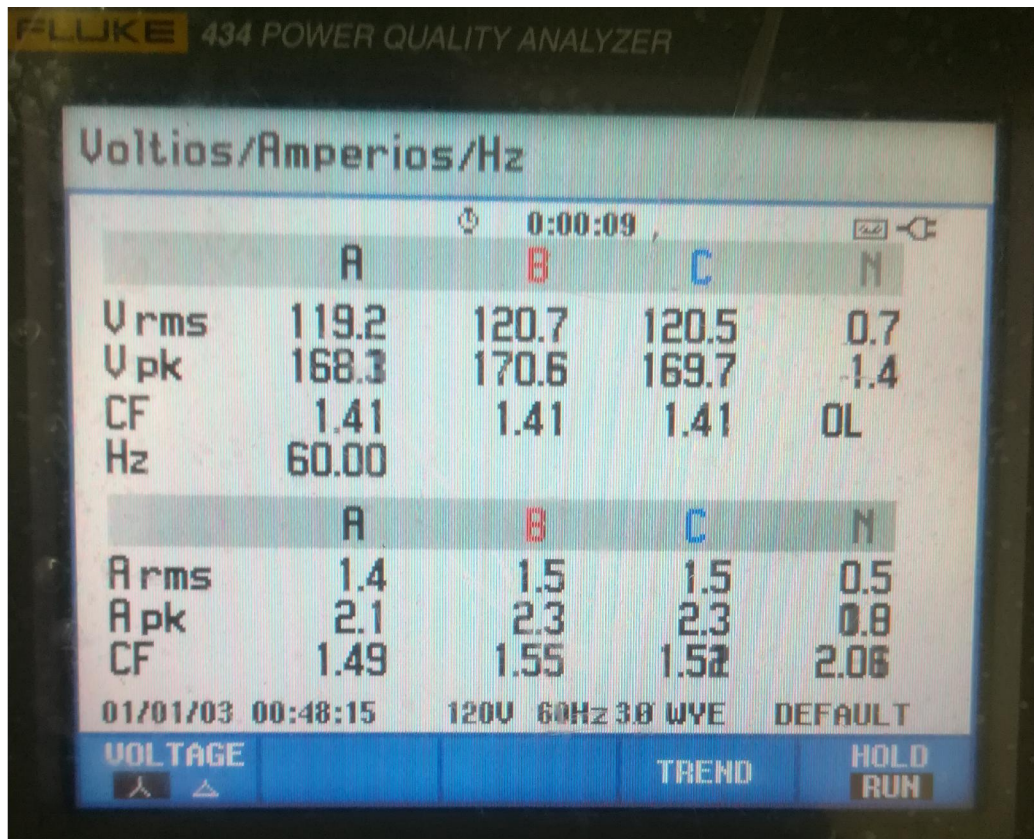


Figura 4.20 Resultados de las mediciones de voltaje y corriente tomados por el analizador de redes FLUKE (conexión errónea).



Figura 4.21 Resultados de las mediciones de voltajes y corriente tomados por el prototipo (conexión errónea).

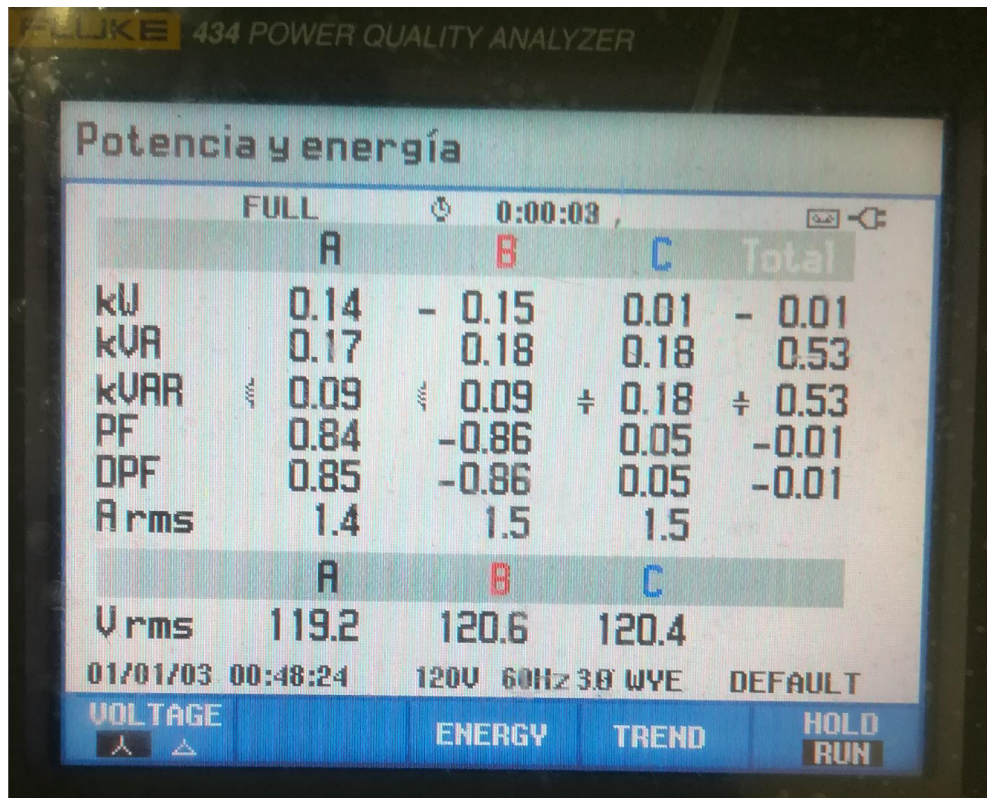


Figura 4.22 Resultados de las mediciones de potencias y factor de potencia tomados por el analizador de redes FLUKE (conexión errónea).



Figura 4.23 Resultados de las mediciones de potencia activa tomados por el prototipo (conexión errónea).

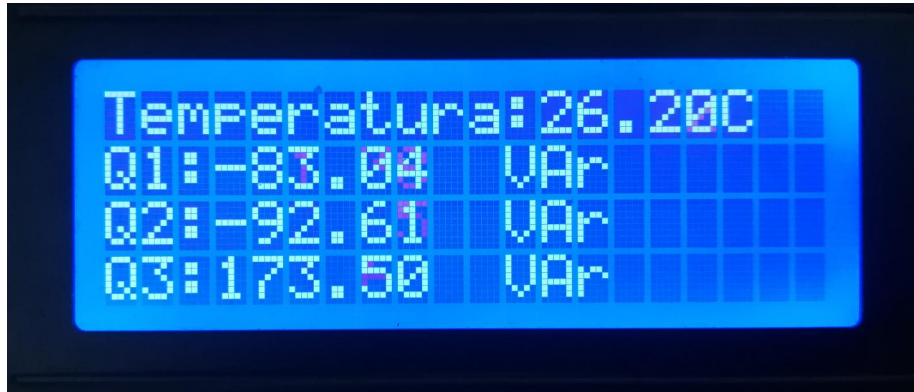


Figura 4.24 Resultados de las mediciones de potencia reactiva tomados por el prototipo (conexión errónea).

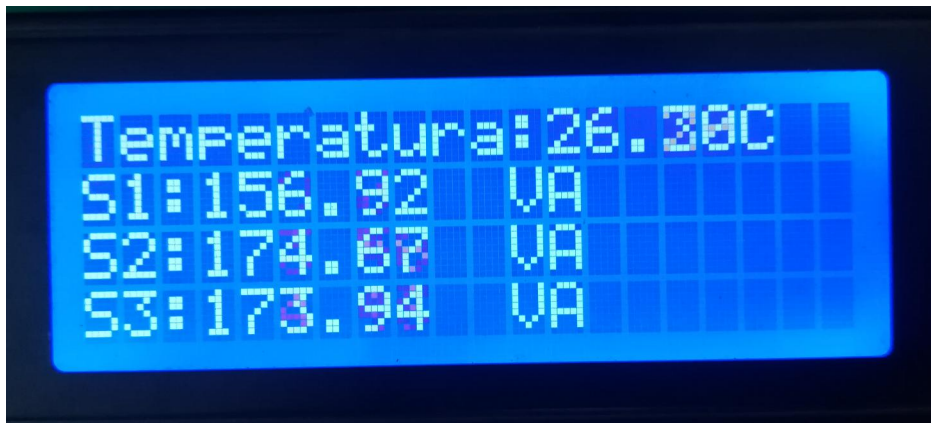


Figura 4.25 Resultados de las mediciones de potencia aparente tomados por el prototipo (conexión errónea).

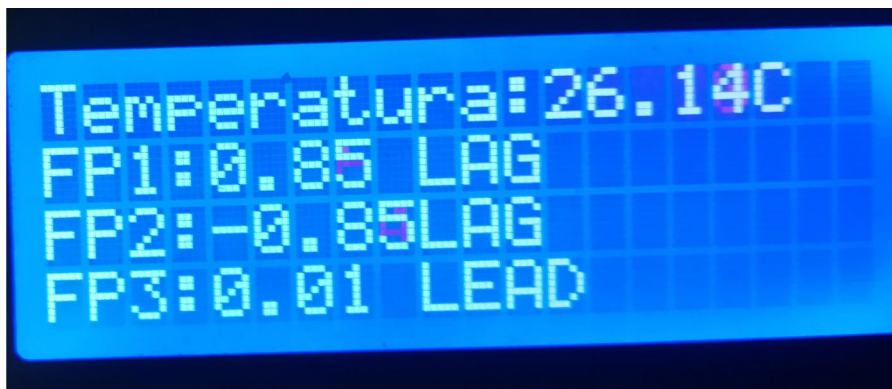


Figura 4.26 Resultados de las mediciones del factor de potencia tomados por el prototipo (conexión errónea).

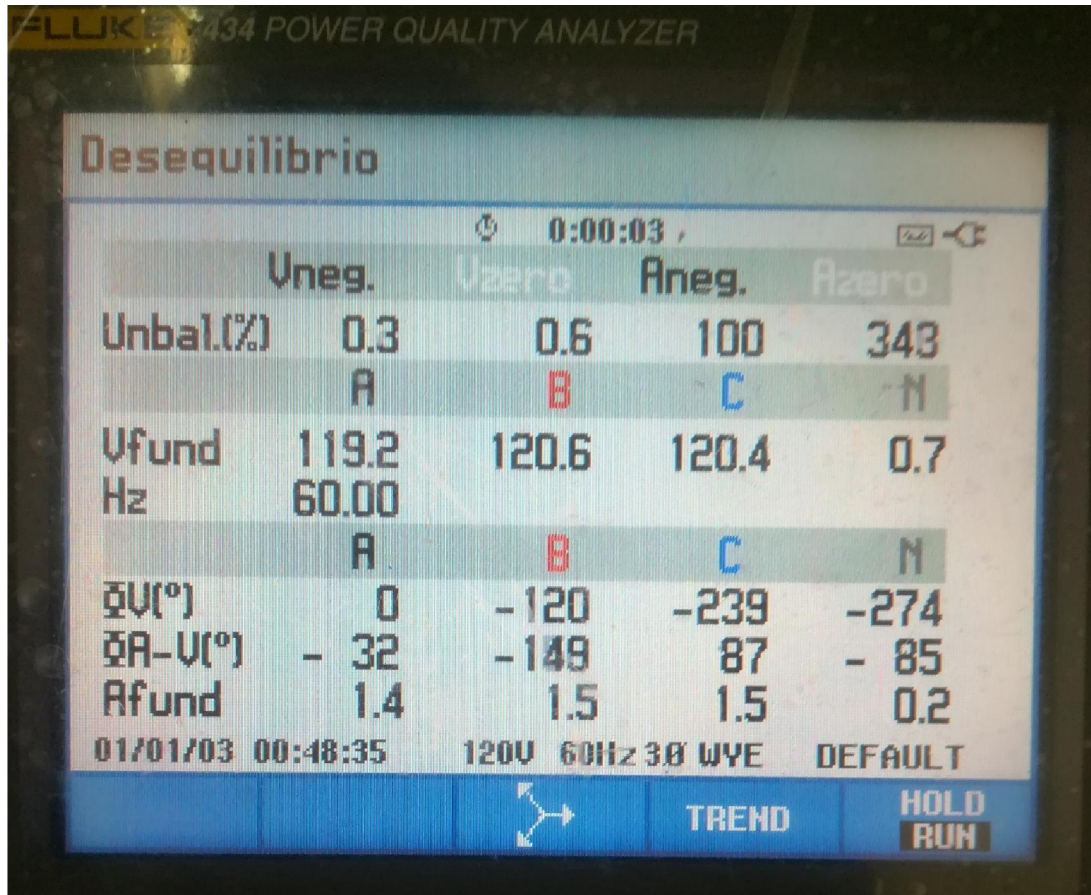


Figura 4.27 Resultados de las mediciones de ángulo de fase tomados por el analizador de redes FLUKE (conexión errónea).

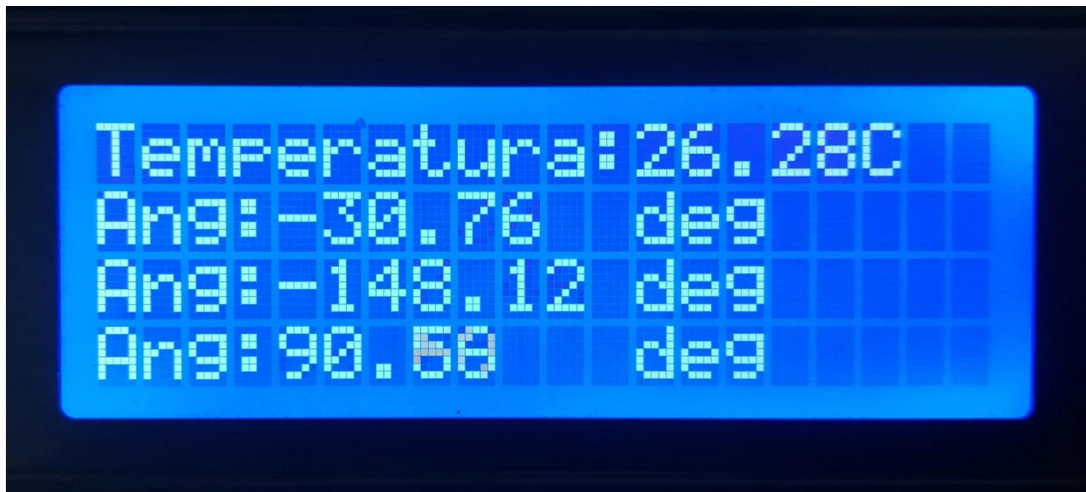


Figura 4.28 Resultados de las mediciones de ángulo de fase tomados por el prototipo (conexión errónea).

Con los resultados obtenidos se llega a la conclusión de que el prototipo si obtiene los parámetros de las fases independientemente y reacciona de la misma manera que el analizador de redes.

4.4. CONEXIÓN A MATLAB® 2015a.

Como el Arduino se alimenta y comunica al ordenador vía USB es posible crear una comunicación con MATLAB® y como se mencionó en el capítulo anterior es recomendable crear historiales para registrar el comportamiento de la máquina y así tener un buen monitoreo de ésta.

En ésta primera prueba se enciende el motor a voltaje pleno repetidas veces para observar el comportamiento de la corriente y por lo tanto de las potencias por línea del motor. Del mismo modo se varió la carga del motor hasta llevarlo a sus valores nominales por línea y regresándolo a un estado sin carga. A continuación se muestran los resultados de las pruebas que se hicieron al prototipo y las gráficas resultantes.

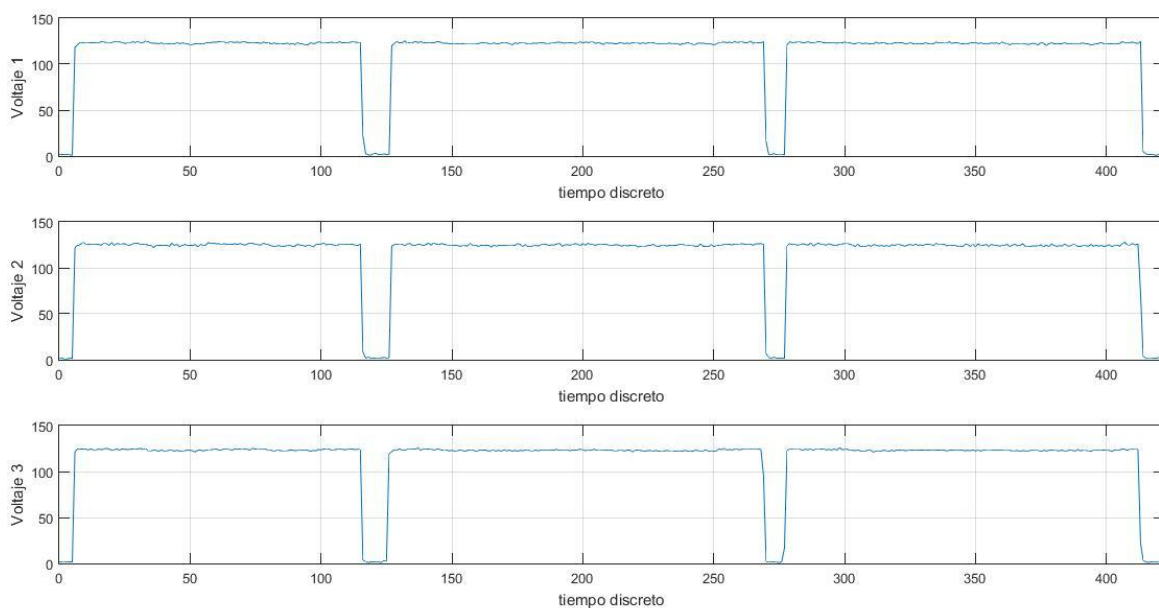


Figura 4.29 Gráficas de voltaje por fase prueba 1.

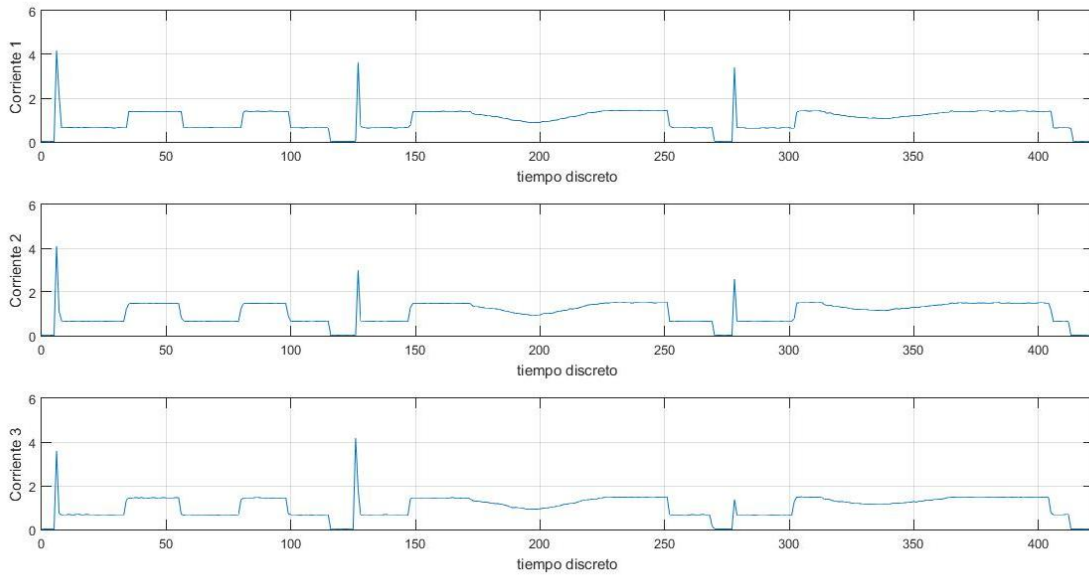


Figura 4.30 Gráficas de corriente por fase prueba 1.

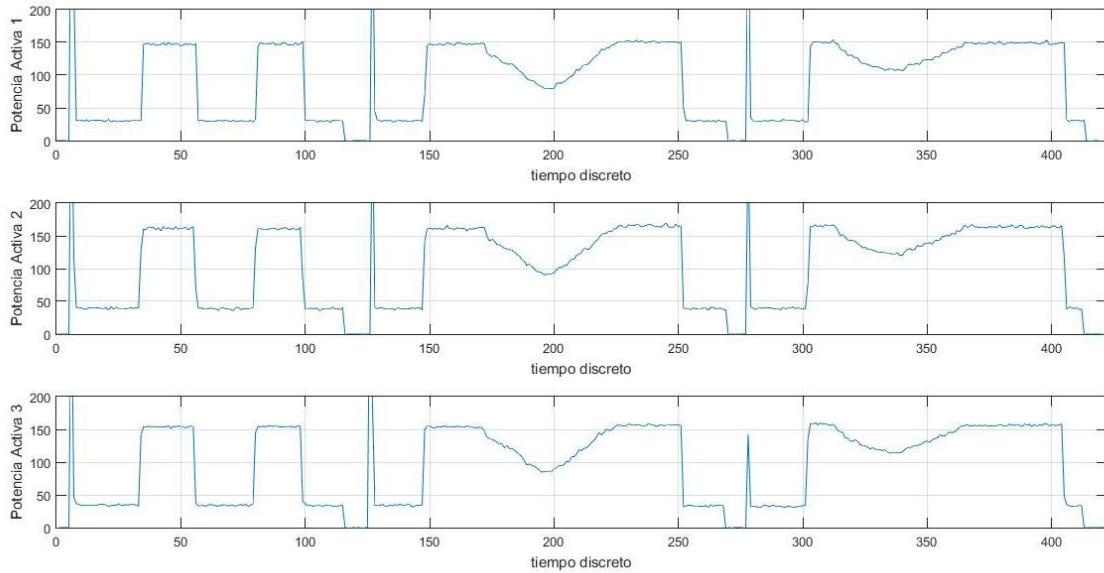


Figura 4.31 Gráficas de potencia activa por fase prueba 1.

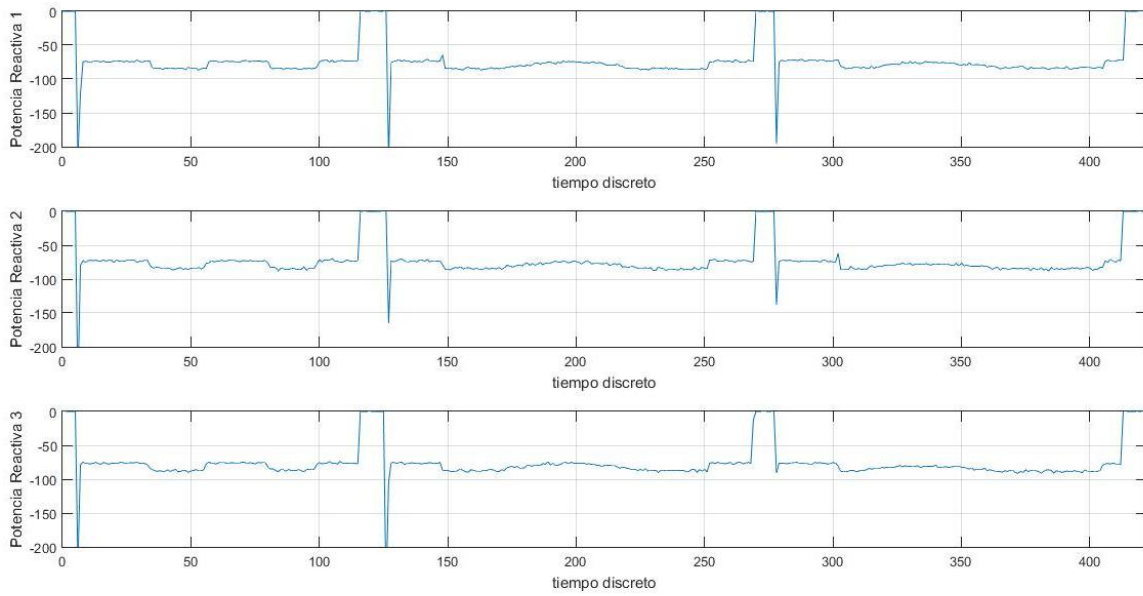


Figura 4.32 Gráficas de potencia reactiva por fase prueba 1.

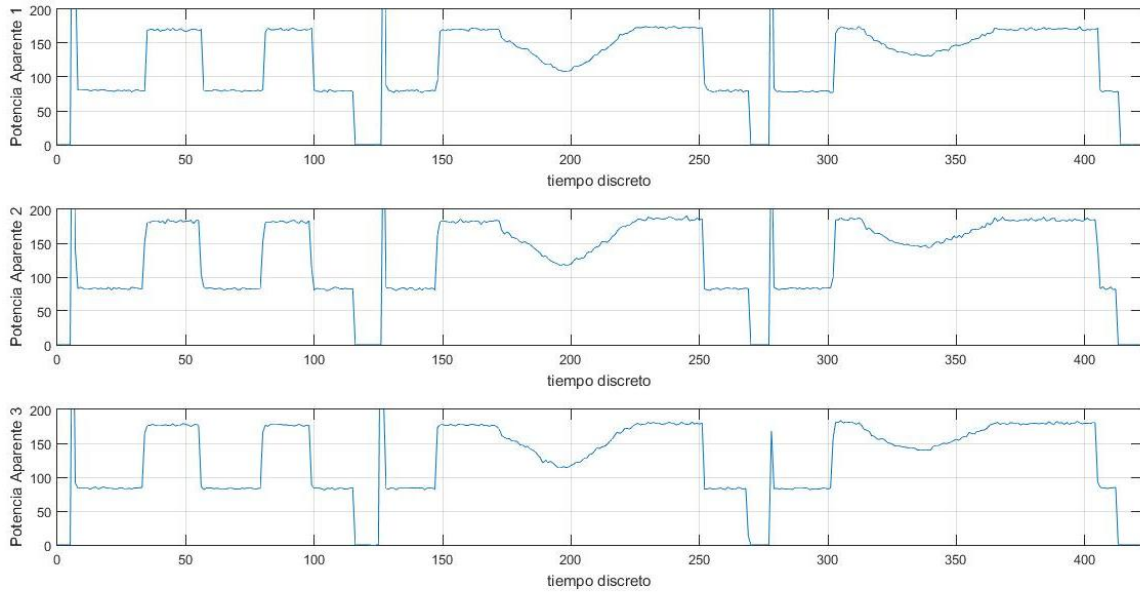


Figura 4.33 Gráficas de potencia aparente por fase prueba 1.

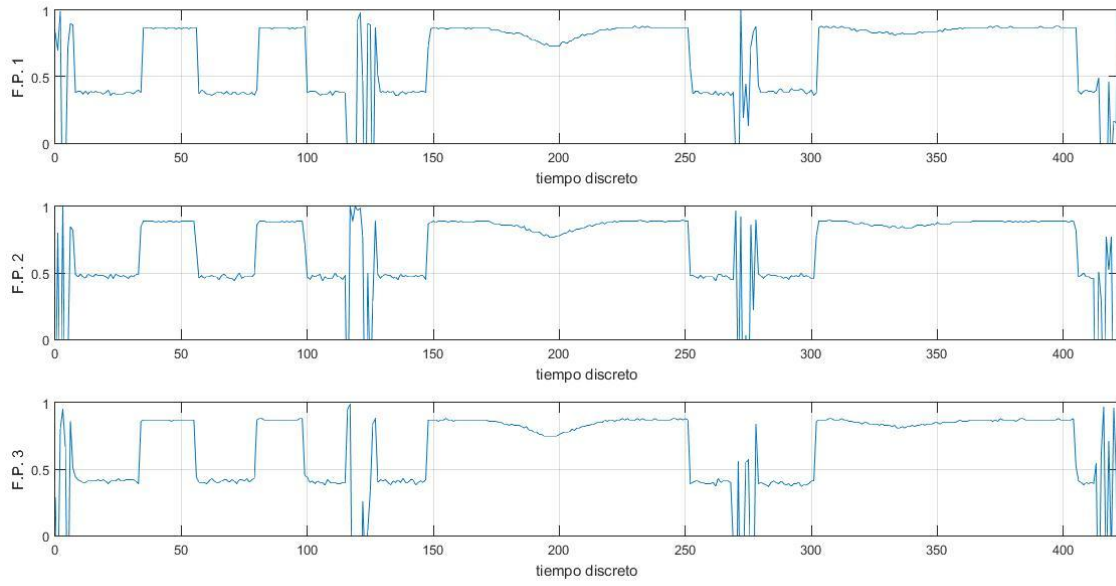


Figura 4.34 Gráficas de Factor de potencia por fase prueba 1.

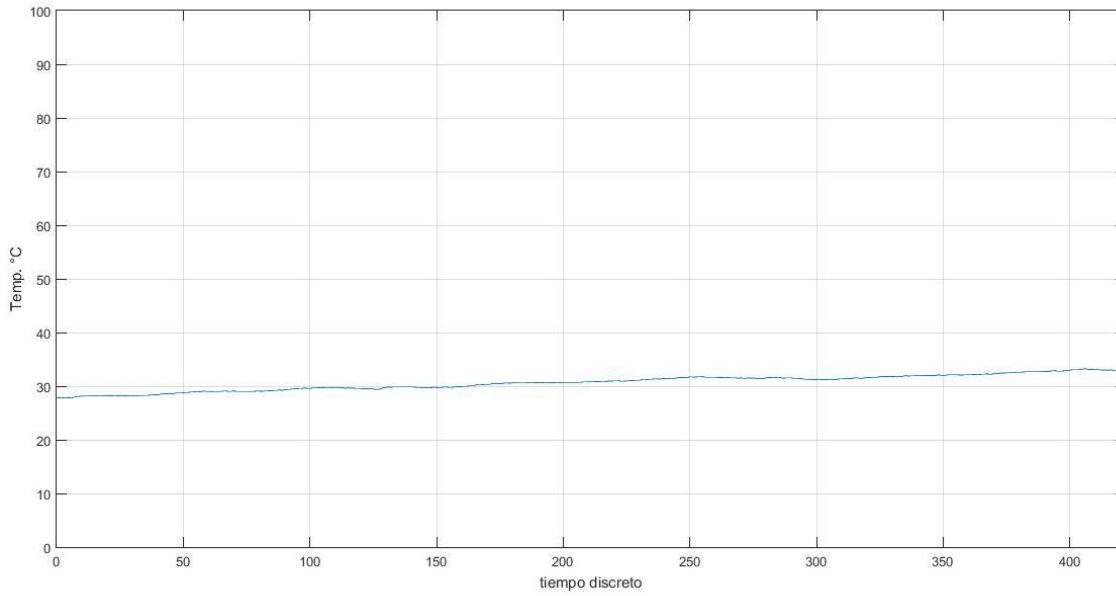


Figura 4.35 Gráfica de temperatura del rotor prueba 1.

Cómo se pudo observar el prototipo es capaz de registrar los picos de corriente que ocurren en el motor al encenderlo directamente a voltaje pleno. En la figura (4.28) se puede observar que el voltaje en las líneas se mantiene igual hasta que el motor es desenergizado. La figura (4.29) muestra el comportamiento de la corriente de línea conforme se enciende y apaga el motor, así como también con la conexión y desconexión de carga, en las figuras (4.30), (4.31), (4.32), se puede apreciar el cómo las potencias actúan directamente a los cambios en la operación del motor tal como se esperaría en un motor de inducción. Nótese que en la figura (4.33) el factor de potencia también cambia de acuerdo a los cambios en el motor, también se puede ver que el factor de potencia se comporta de forma errática cuando el motor es apagado, esto es debido a que cuando el motor está apagado los sensores leen solo la señal de ruido que existe en la señal de salida y por eso se crea esa distorsión. Y por último en la figura (4.34) manifiesta el cambio de temperatura que existe en el rotor.

La segunda prueba con la transmisión de datos a MATLAB® consiste en mantener encendido el motor de inducción con una carga conectada y a continuación variar la carga para mantener los parámetros del motor en constante cambio. El comportamiento del motor puede observarse a continuación.

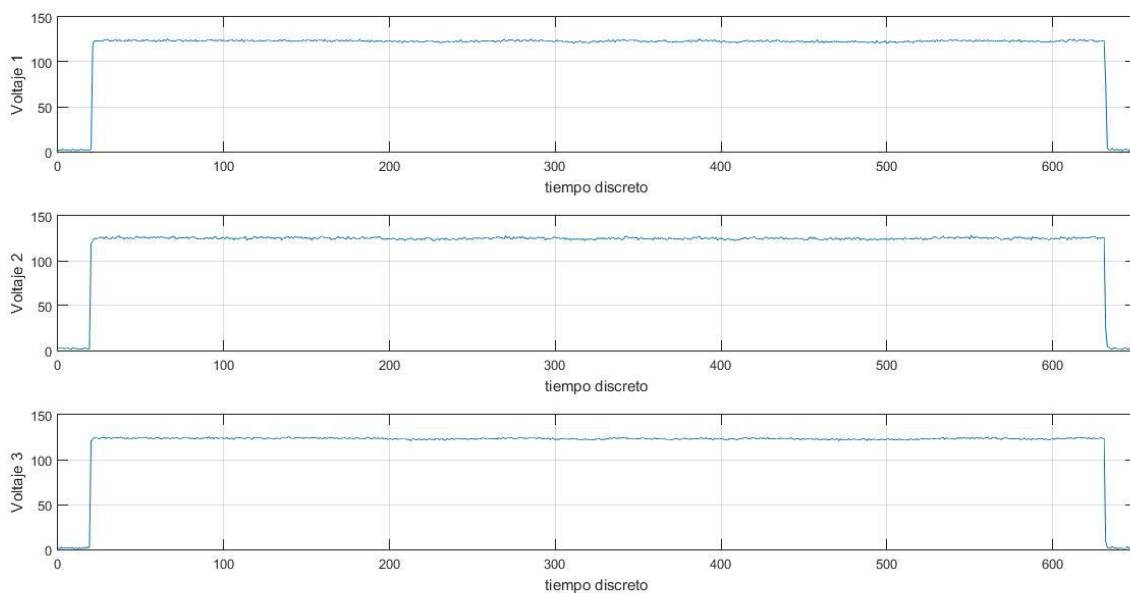


Figura 4.36 Gráficas de voltaje por fase prueba 2.

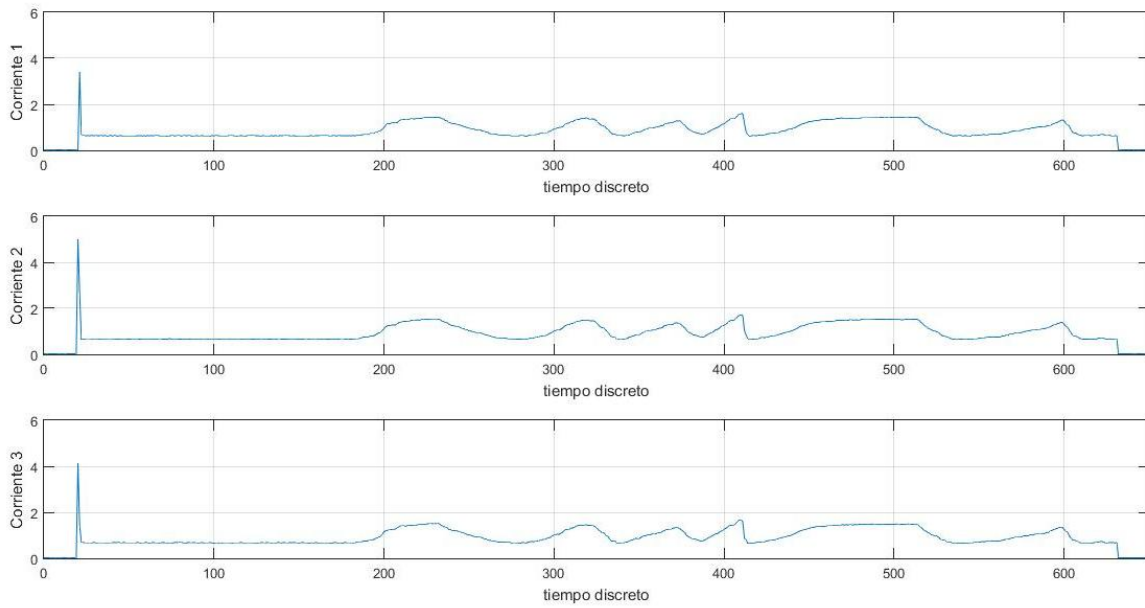


Figura 4.37 Gráficas de corriente por fase prueba 2.

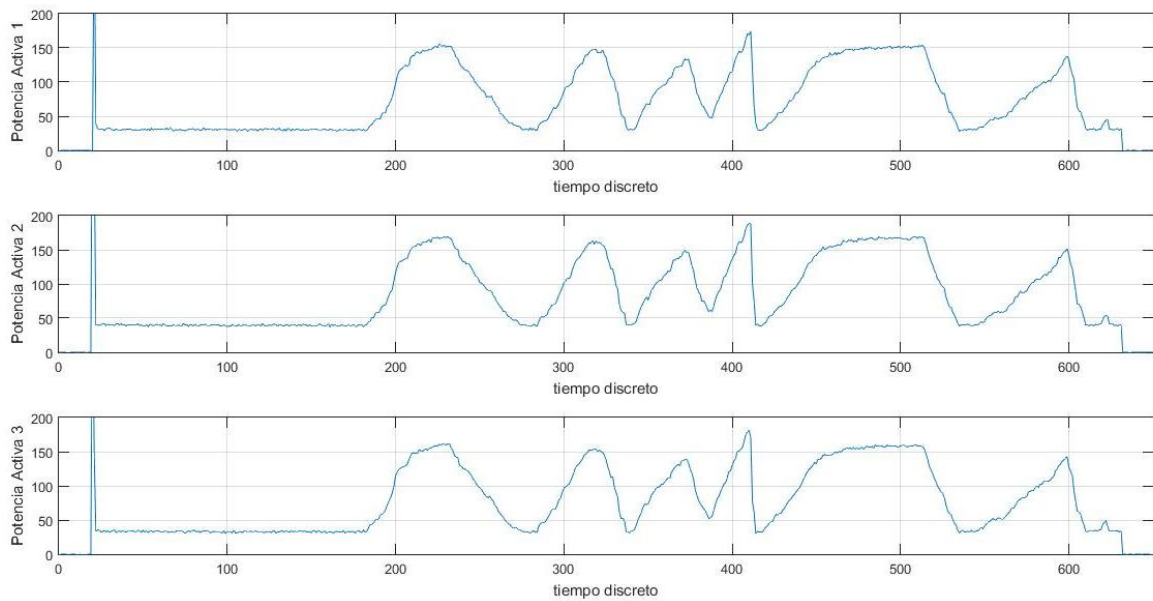


Figura 4.38 Gráficas de potencia activa por fase prueba 2.

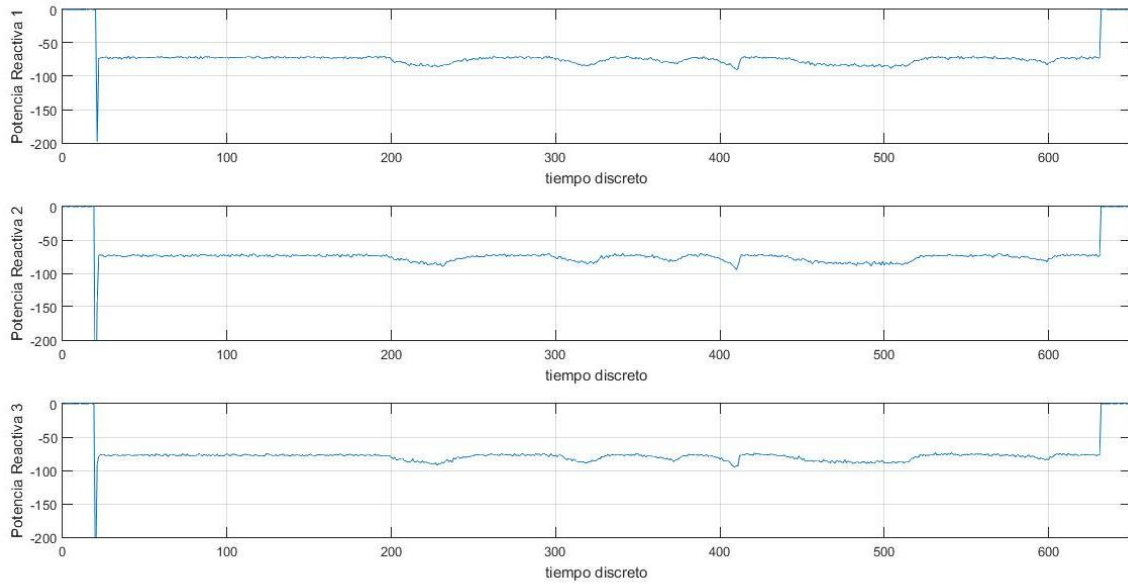


Figura 4.39 Gráficas de potencia reactiva por fase prueba 2.

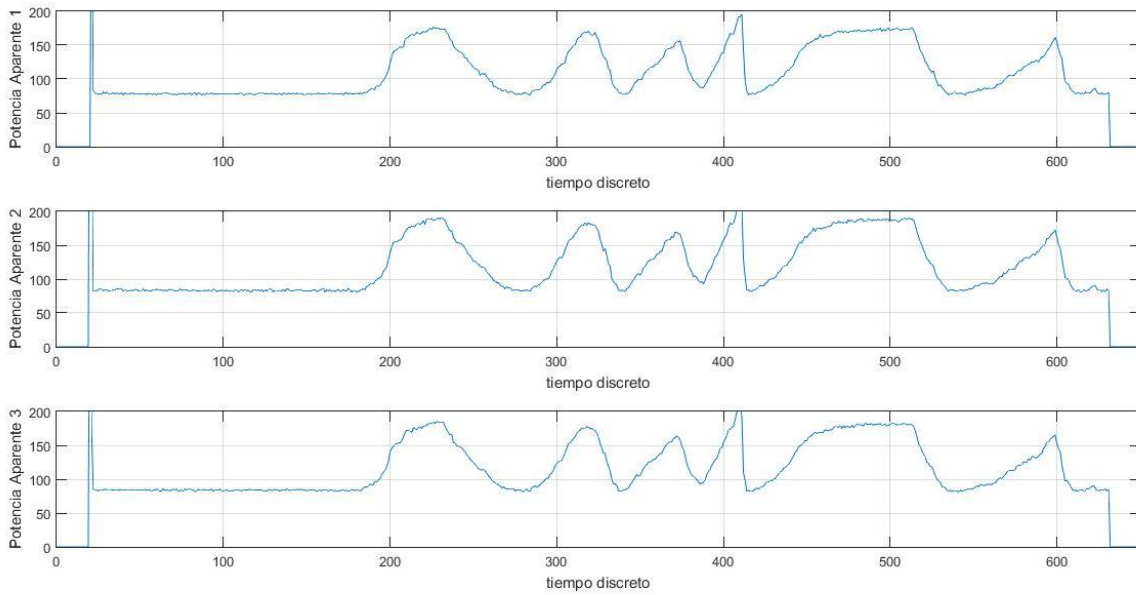


Figura 4.40 Gráficas de potencia aparente por fase prueba 2.

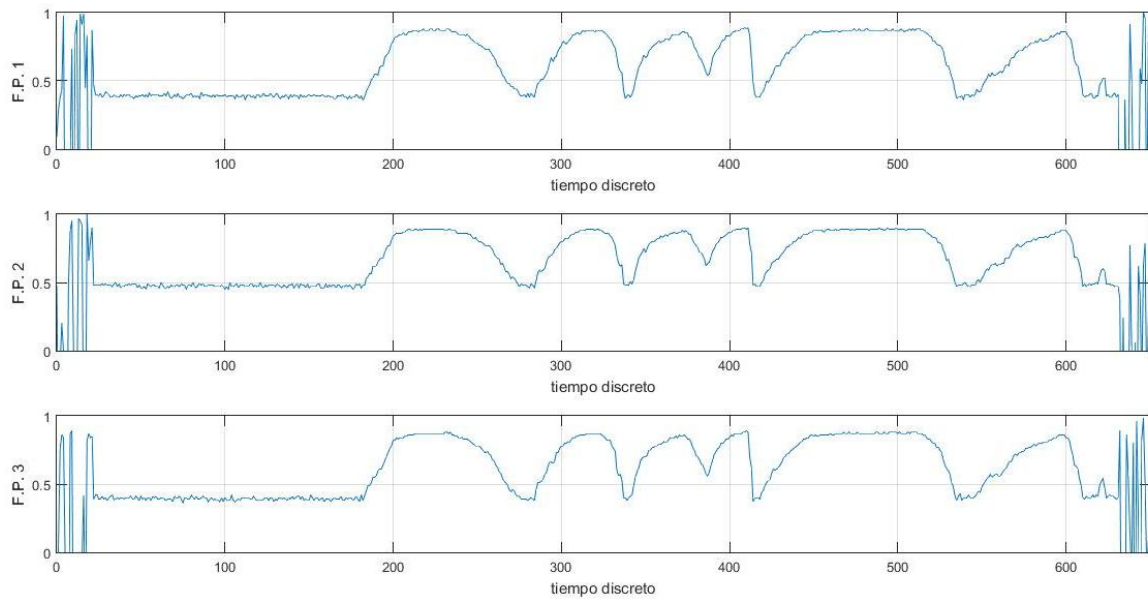


Figura 4.41 Gráficas de factor de potencia por fase prueba 2.

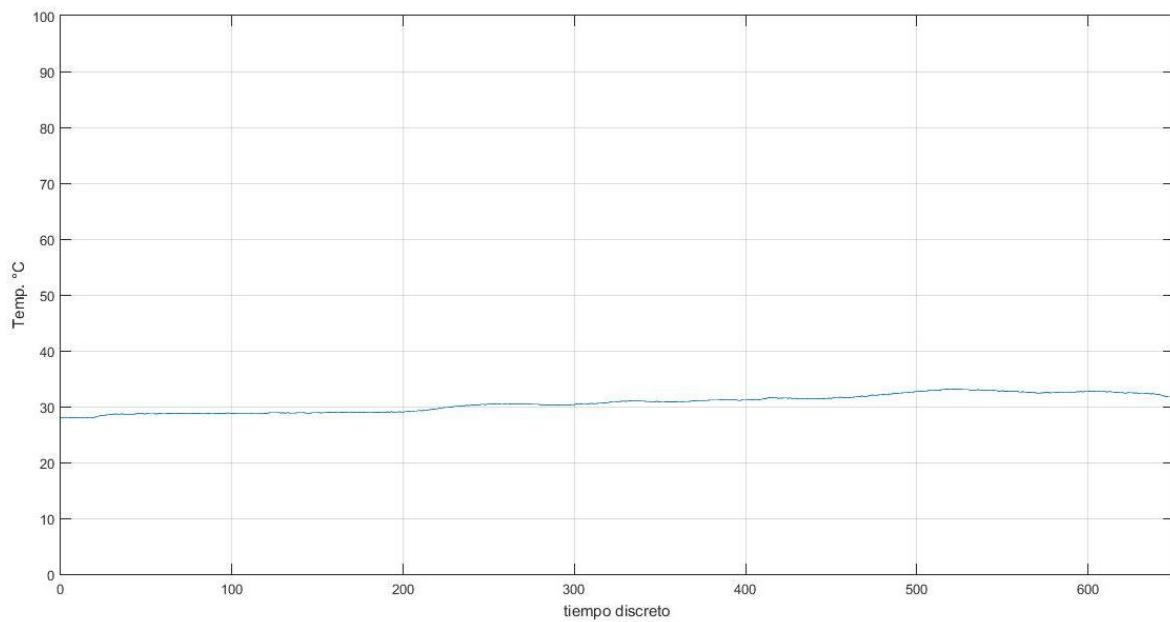


Figura 4.42 Gráfica de temperatura del rotor prueba 2.

En la figura (4.35) se puede apreciar que el voltaje entre fases no cambia en lo absoluto al variarle la carga ya que este está directamente relacionado con la fuente de alimentación. En la figura (4.36) se muestra el comportamiento de las Corrientes del motor por línea y en las figuras (4.37), (4.38), (4.39), se pueden observar las potencias, nótese como las variaciones en éstas están relacionadas directamente con el comportamiento de la corriente. La figura (4.40) muestra el factor de potencia de las líneas aquí se puede apreciar de manera evidente que los cambios también corresponden al incremento y decremento en la carga. Por último en la figura (4.41) se visualiza el cambio de la temperatura en el rotor y como éste va a aumentando conforme pasa el tiempo y se le agrega más carga al motor.

Gracias a estos resultados se puede decir que el prototipo puede registrar cambios relativamente rápidos y erráticos en el comportamiento de las líneas del motor y crearse un historial de estos eventos para tener un registro donde se puedan observar fallas cambios bruscos en la carga e incluso desconexiones o un desbalance en las cargas del motor y hasta cortos circuitos.

CONCLUSIÓN Y TRABAJOS FUTUROS

En éste apartado se realiza la conclusión sobre el proyecto planteado en la presente tesis, así como recomendaciones para mejorar el prototipo para un mejor desempeño en trabajos futuros.

5.1. CONCLUSIONES.

La adquisición de los parámetros del motor mediante el uso de una placa Arduino y sensores de Voltaje y Corriente fue posible gracias al uso de la Transformada Discreta de Fourier que permite, sin el uso de hardware extra, obtener los fasores de ondas sinusoidales, los valores RMS de Voltaje, Corriente, Potencia Activa, Potencia Reactiva y Potencia Aparente y el valor del Factor de Potencia que hay en cada una de las líneas del motor. La temperatura interna del motor fue posible obtenerse gracias al sensor no invasivo infrarrojo de temperatura y a la librería 'i2cmaster.h' creada por Peter Fleury, 2005. El monitoreo de las variables fue posible por el proceso digital del Arduino Mega 2560 y el uso de una pantalla LCD donde se despliegan los resultados de los cálculos de la T.D.F.

Debido al tamaño reducido de los sensores utilizados para este proyecto fue posible hacer de este un prototipo portátil que puede llevarse al área donde se le necesite. Aunque se obtuvieron los parámetros deseados para el monitoreo del motor de inducción estos presentan un porcentaje de error superando a veces el 5% lo cual ya es considerable lo cual es debido al despliegue de las variables en la LCD que en ocasiones suele variar considerablemente o por pequeñas fallas de conexión y a que el sensor de voltaje no está del todo bien calibrado y solo se obtienen los valores RMS de la onda fundamental, la 3^a, 5^a y 7^a armónica como se mencionó en el capítulo anterior. A pesar de estos percances el prototipo tiene una velocidad de transmisión y procesamiento considerablemente rápido.

La comunicación con MATLAB® por medio del puerto serial permitió una mejor apreciación gráfica del comportamiento del motor ante los cambios en sus variables, más específicamente, cuando se le agrega una carga o se enciende a voltaje pleno y gracias a ésta cualidad del prototipo se pudieron crear historiales gráficos y una tabla de valores de cada variable de las pruebas que se le realicen o las posibles fallas que pueda tener un motor de inducción.

5.2. RECOMENDACIONES.

Se recomienda investigar una mejor forma para calibrar el sensor de Voltaje zmp101b para un sensado más preciso.

Si se desea hacer un sensado no invasivo se pueden utilizar sensores de corriente de CA no invasivo del modelo SCT-013 el cual se puede utilizar con la placa Arduino. [22]

Para aumentar la velocidad de procesamiento del prototipo y poder tomar más muestras por fase y de forma simultánea, se recomienda utilizar un microprocesador con un sensor de Corriente y Voltaje por fase los cuales realicen un muestreo simultaneo y envíen sus resultados vía protocolo i2C hacia un maestro el cual podrá disponer de estos datos para procesarlos más rápido.

Investigar cómo crear un gestor gráfico que utilice los parámetros medidos para visualizar de forma más analítica el comportamiento de las fases del motor de inducción creando un diagrama fasorial y graficas donde se visualicen los parámetros.

Puede ser agregado un módulo Bluetooth hc-06 o de Wifi ESP8266 para crear servidores que proporcionen la información sobre los parámetros del motor vía inalámbrica a celulares o incluso otros equipos de cómputo según se necesite. [23] [24]

Referencias

- [1] “Tesla Memorial Society of New York Website” <http://bit.ly/25NvqdL>
- [2] Medición de potencia en sistemas eléctricos <http://bit.ly/1WGVHsH>
- [3] EQUIPOS DE MEDICION. <http://bit.ly/1PMB8bV>
- [4] W. Leonhard, “Control of Electrical Drives”, power systems series, 3a Edición, Springer, Nueva York, 2001.
- [5] Daniel Alberto Saucedo Martínez, et. al, “Factores que afectan la calidad de la energía y su solución”
- [6] <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>.
- [7] MOTORES DE CORRIENTE ALTERNA. Sistemas Automáticos de Producción Alimentaria, Andrés De Vandelvira Departamento De Sap ©J. Garrigós Noviembre 2011.
- [8] Motores Asíncronos Trifásicos, <http://www-app.etsit.upm.es/departamentos/teat/asignaturas/lab-ingel/motor%20asincrono%20trifasico.pdf>.
- [9] http://www.trifasica.net/pdf/TEMA_9._POTENCIA_EN_SISTEMAS_TRIFASICOS.pdf
- [10] Arranque de los motores de inducción, Departamento de Ingeniería Eléctrica – E.T.S.I.I. <http://www.die.eis.uva.es/~daniel/docencia/te/TEIQPractica9-2008.pdf>
- [11] Universidad Nacional de San Luis Facultad de Ciencias Físico Matemáticas y Naturales Ingeniería Electrónica con orientación en Sistemas Digitales Técnico Universitario en Microprocesadores Profesorado en Tecnología Electrónica Sensores de efecto Hall <http://www0.unsl.edu.ar/~interfases/labs/lab09.pdf>
- [12] Allegro MicroSystems, LLC, ACS712-Datasheet 2006-2017

-
- [13] <https://www.plexishop.it/pdf/ZMPT101B%20-%20Datasheet.jpg>
- [14] www.sparkfun.com/datasheets/Sensors/Temperature/MLX90614_rev001.pdf
- [15] <https://www.luisllamas.es/arduino-i2c/>
- [16] La Serie de Fourier en Forma Compleja, M. I. Joel Rodríguez Guillén, Instituto Tecnológico de Ciudad Madero, Diciembre 2016.
- [17] La Transformada Discreta de Fourier, M. I. Joel Rodríguez Guillén, Instituto Tecnológico de Ciudad Madero, Diciembre 2016.
- [18] Proof of Using Fourier Coefficients for Root Mean Square Calculations on Periodic Signals, Sompop Poomjan, et al, Department of Physics, Faculty of Science King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Rd. Ladkrabang, Bangkok Thailand 10520
- [19] Los Armónicos y la Calidad de la Energía Eléctrica, http://www.rtrenergia.es/downloads/armonicos_2012.pdf
- [20] Procedimiento para la verificación de medidores de energía eléctrica estáticos clases 1 y 2 y electromecánico clase 2, <http://www.inacal.gob.pe/inacal/files/metrologia/%20HCMEE/PV-001-INACAL.pdf>
- [21] Criterios de Selección de Transformadores de Corriente (TC's) En aplicaciones de Protección y Medición, Subgerencia de Protecciones Departamento de Ingeniería de Protecciones, Julio de 2008.
- [22] http://www.mcielectronics.cl/website_MCI/static/documents/Datasheet_SCT013.pdf
- [23] <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>
- [24] ESP8266EX Datasheet, Version 4.3 Espressif Systems IOT Team <http://bbs.espressif.com/> Copyright © 2015

Apéndice

A

Programa del prototipo

```
#include <i2cmaster.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

const unsigned long tiempoMuestreo = 16667UL;           // muestras sobre 100ms, es un
numero exacto de ciclos para fuentes de 60Hz igual a 6 ciclos
const unsigned long numMuestras = 40UL;                // escoge el número de muestras para
dividir exactamente el tiempo de muestreo, pero lo suficientemente bajo para mantener en
contacto el ADC
const unsigned long intervalosMuestra = tiempoMuestreo/numMuestras; // el intervalo de
muestreo igual a 400, tiene que ser más largo que el tiempo de conversión
int pantalla = 1; //variable para el cambio de pantallas

int dev = 0x5A<<1;
int data_low = 0;
int data_high = 0;
int pec = 0;

float pi = 3.141592654;
float Vrms[3]; //Vector donde se guardarán los valores de voltaje RMS
float V1=0;
float V2=0;
float V3=0;
float Irms[3]; //Vector donde se guardarán los valores de corriente RMS
```

```
float I1=0;
float I2=0;
float I3=0;
float Spower[3]; //Vector donde se guardarán los valores de potencia aparente RMS
float SP_1=0;
float SP_2=0;
float SP_3=0;
float Apower[3]; //Vector donde se guardarán los valores de potencia activa RMS
float AP_1=0;
float AP_2=0;
float AP_3=0;
float Rpower[3]; //Vector donde se guardarán los valores de potencia reactiva RMS
float RP_1=0;
float RP_2=0;
float RP_3=0;
float Theta[3]; //Vector donde se guardarán los fasores
float fase1=0;
float fase2=0;
float fase3=0;
float PowerF[3]; //Vector donde se guardarán los valores de factor de potencia
float PF_1=0;
float PF_2=0;
float PF_3=0;
int Cfp[3];
int red_1=0;
int red_2=0;
int red_3=0;

void setup() {
    // put your setup code here, to run once:

    i2c_init(); //Inicia el bus I2C
```

```
PORTC = (1 << PORTC4) | (1 << PORTC5); //activa la resistencia Pullup del arduino
Serial.begin(250000); //inicia la comunicación serial
lcd.begin(20,4); //establece el tamaño de la pantalla LCD
attachInterrupt(digitalPinToInterrupt(18),Pmas,FALLING); //interrupción para el cambio de
pantallas
attachInterrupt(digitalPinToInterrupt(19),Pmin,FALLING); //interrupción para el cambio de
pantallas
}

void loop() {
  // put your main code here, to run repeatedly:

  ////////////////////////////////////////
  ///          Sensado de Temperatura          ///
  ////////////////////////////////////////
  i2c_start_wait(dev+I2C_WRITE); //lee la dirección del sensor
  i2c_write(0x07);

  //lectura
  i2c_rep_start(dev+I2C_READ);
  data_low = i2c_readAck(); //Lee 1 byte y lo manda a ack
  data_high = i2c_readAck(); //Lee 1 byte y lo manda a ack
  pec = i2c_readNak();
  i2c_stop();

  //Esto convierte los byte altos y bajos juntos y procesa la temperatura, MSB es un bit de error y
es ignorado para la temp.
  double tempFactor = 0.02; // 0.02 grados por LSB (resolución de medición del MLX90614)
  double tempData = 0x0000; // Datos en Cero
  int frac; // datos pasado el punto decimal
  //Esto enmascara el bit de error del byte alto, luego lo mueve 8 bits a la izquierda y añade el byte
bajo
```

```
tempData = (double)((((data_high & 0x007F) << 8) + data_low));  
tempData = (tempData * tempFactor)-0.01;
```

```
float celcius = tempData - 273.15; //Hace la conversión de grados Kelvin a Celsius
```

```
VrmsFundamental(0); //instrucción para llamar la función de coeficientes de furier para obtener  
los parametros
```

```
//se procede a llenar las variables con los valores medidos de los parametros.
```

```
V1=Vrms[0];
```

```
I1=Irms[0];
```

```
fase1=Theta[0];
```

```
PF_1=PowerF[0];
```

```
SP_1=Spower[0];
```

```
AP_1=Apower[0];
```

```
RP_1=Rpower[0];
```

```
red_1=Cfp[0];
```

```
V2=Vrms[1];
```

```
I2=Irms[1];
```

```
fase2=Theta[1];
```

```
PF_2=PowerF[1];
```

```
SP_2=Spower[1];
```

```
AP_2=Apower[1];
```

```
RP_2=Rpower[1];
```

```
red_2=Cfp[1];
```

```
V3=Vrms[2];
```

```
I3=Irms[2];
```

```
fase3=Theta[2];
```

```
PF_3=PowerF[2];
```

```
SP_3=Spower[2];
```

```
AP_3=Apower[2];  
RP_3=Rpower[2];  
red_3=Cfp[2];
```

```
Serial.print(V1);  
Serial.print(",");  
Serial.print(V2);  
Serial.print(",");  
Serial.print(V3);  
Serial.print(",");
```

```
Serial.print(I1);  
Serial.print(",");  
Serial.print(I2);  
Serial.print(",");  
Serial.print(I3);  
Serial.print(",");
```

```
Serial.print(PF_1);  
Serial.print(",");  
Serial.print(PF_2);  
Serial.print(",");  
Serial.print(PF_3);  
Serial.print(",");
```

```
Serial.print(SP_1);  
Serial.print(",");  
Serial.print(SP_2);  
Serial.print(",");  
Serial.print(SP_3);  
Serial.print(",");
```

```
Serial.print(AP_1);
Serial.print(",");
Serial.print(AP_2);
Serial.print(",");
Serial.print(AP_3);
Serial.print(",");
```

```
Serial.print(RP_1);
Serial.print(",");
Serial.print(RP_2);
Serial.print(",");
Serial.print(RP_3);
Serial.print(",");
```

```
Serial.println(celcius);
```

```
////////////////////////////////////
///                               Impresion de pantallas                               ///
////////////////////////////////////
```

```
switch (pantalla) {
  case 1:
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temperatura:");
    lcd.print(celcius);
    lcd.print("C");
    lcd.setCursor(0, 1);
    lcd.print("V1:");
    lcd.print(V1);
    lcd.print("V");
    lcd.setCursor(12, 1);
```



```
lcd.print("I1:");  
lcd.print(I1,1);  
lcd.print("A");
```

```
lcd.setCursor(0, 2);  
lcd.print("V2:");  
lcd.print(V2);  
lcd.print("V");  
lcd.setCursor(12, 2);  
lcd.print("I2:");  
lcd.print(I2,1);  
lcd.print("A");
```

```
lcd.setCursor(0, 3);  
lcd.print("V3:");  
lcd.print(V3);  
lcd.print("V");  
lcd.setCursor(12, 3);  
lcd.print("I3:");  
lcd.print(I3,1);  
lcd.print("A");
```

```
break;
```

```
case 2:
```

```
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Temperatura:");  
lcd.print(celcius);  
lcd.print("C");  
lcd.setCursor(0, 1);  
lcd.print("FP1:");  
lcd.setCursor(4, 1);
```

```
lcd.print(PF_1);
switch(red_1){
  case 1:
    lcd.setCursor(9, 1);
    lcd.print("LAG");
  break;
  case 2:
    lcd.setCursor(9, 1);
    lcd.print("LEAD");
  break;
}
```

```
lcd.setCursor(0, 2);
lcd.print("FP2:");
lcd.setCursor(4, 2);
lcd.print(PF_2);
switch(red_2){
  case 1:
    lcd.setCursor(9, 2);
    lcd.print("LAG");
  break;
  case 2:
    lcd.setCursor(9, 2);
    lcd.print("LEAD");
}
```

```
lcd.setCursor(0, 3);
lcd.print("FP3:");
lcd.setCursor(4, 3);
lcd.print(PF_3);
switch(red_3){
  case 1:
```

```
        lcd.setCursor(9, 3);
        lcd.print("LAG");
    break;
    case 2:
        lcd.setCursor(9, 3);
        lcd.print("LEAD");
    break;
}
break;
case 3:
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temperatura:");
    lcd.print(celcius);
    lcd.print("C");
    lcd.setCursor(0, 1);
    lcd.print("Ang:");
    lcd.print(fase1);
    lcd.setCursor(12, 1);
    lcd.print("deg");

    lcd.setCursor(0, 2);
    lcd.print("Ang:");
    lcd.print(fase2);
    lcd.setCursor(12, 2);
    lcd.print("deg");

    lcd.setCursor(0, 3);
    lcd.print("Ang:");
    lcd.print(fase3);
    lcd.setCursor(12, 3);
    lcd.print("deg");
```

```
break;
case 4:
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temperatura:");
    lcd.print(celcius);
    lcd.print("C");
    lcd.setCursor(0, 1);
    lcd.print("S1:");
    lcd.print(SP_1);
    lcd.setCursor(11, 1);
    lcd.print("VA");

    lcd.setCursor(0, 2);
    lcd.print("S2:");
    lcd.print(SP_2);
    lcd.setCursor(11, 2);
    lcd.print("VA");

    lcd.setCursor(0, 3);
    lcd.print("S3:");
    lcd.print(SP_3);
    lcd.setCursor(11, 3);
    lcd.print("VA");
break;
case 5:
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temperatura:");
    lcd.print(celcius);
    lcd.print("C");
    lcd.setCursor(0, 1);
```

```
lcd.print("PA1:");
lcd.print(AP_1);
lcd.setCursor(11, 1);
lcd.print("W");

lcd.setCursor(0, 2);
lcd.print("PA2:");
lcd.print(AP_2);
lcd.setCursor(11, 2);
lcd.print("W");

lcd.setCursor(0, 3);
lcd.print("PA3:");
lcd.print(AP_3);
lcd.setCursor(11, 3);
lcd.print("W");
break;
case 6:
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temperatura:");
  lcd.print(celcius);
  lcd.print("C");
  lcd.setCursor(0, 1);
  lcd.print("Q1:");
  lcd.print(RP_1);
  lcd.setCursor(11, 1);
  lcd.print("VAr");

  lcd.setCursor(0, 2);
  lcd.print("Q2:");
  lcd.print(RP_2);
```

```

    lcd.setCursor(11, 2);
    lcd.print("VAr");

    lcd.setCursor(0, 3);
    lcd.print("Q3:");
    lcd.print(RP_3);
    lcd.setCursor(11, 3);
    lcd.print("VAr");
break;

}

}

////////////////////////////////////
///                               Toma de muestras                               ///
////////////////////////////////////

float VrmsFundamental(int PIN) // funcion que obtiene los parametros del motor por medio de
coeficientes de fourier
{
    for(int c=0;c<3;++c)
    {
        unsigned int contador = 0;
        unsigned long microsAnt = micros() - intervalosMuestra ;
        //int j=0;
        int k = 4;
        float XREAL[k];
        float XIMAG[k];
        float YREAL[k];
        float YIMAG[k];
        double ARGX[k];

```

```
double ARGY[k];
double f[k];
float CX[k];
float CY[k];
float xi[numMuestras]; // Vector donde se guardarán las muestras tomadas por el sensor
de voltaje
float yi[numMuestras]; // Vector donde se guardarán las muestras tomadas por el sensor
de corriente
```

```
while (contador < numMuestras)
{
    if (micros() - microsAnt >= intervalosMuestra)
    {
        xi[contador]= analogRead(2*c);
        yi[contador]= analogRead((2*c)+1);
        ++contador;
        microsAnt += intervalosMuestra;
    }
}

for (int j=0;j<k;++j){
    XREAL[j]=0;
    XIMAG[j]=0;
    YREAL[j]=0;
    YIMAG[j]=0;
    for(int i=0;i<numMuestras;++i)
    {
        XREAL[j] += ((float)xi[i])*cos((2*pi*(float)(2*j+1)*i)/numMuestras);
        XIMAG[j] += ((float)xi[i])*sin((2*pi*(float)(2*j+1)*i)/numMuestras);
        YREAL[j] += ((float)yi[i])*cos((2*pi*(float)(2*j+1)*i)/numMuestras);
        YIMAG[j] += ((float)yi[i])*sin((2*pi*(float)(2*j+1)*i)/numMuestras);
    }
}
```

```

    }
}
for(int j=0;j<k;++j){
    XREAL[j] = 2*(XREAL[j]/(float)numMuestras);
    XIMAG[j] = 2*(XIMAG[j]/(float)numMuestras);
    YREAL[j] = 2*(YREAL[j]/(float)numMuestras);
    YIMAG[j] = 2*(YIMAG[j]/(float)numMuestras);
    CX[j] = sqrt((float)(XREAL[j]*XREAL[j]+XIMAG[j]*XIMAG[j]))* ((5*201.0869565)/
1023.0); //201.0869565
    CY[j] = sqrt((float)(YREAL[j]*YREAL[j]+YIMAG[j]*YIMAG[j]))* ((5/0.100)/ 1023.0);
    //Vrms[j] = CX[j]/(float)sqrt(2) ;
    //Irms[j] = CY[j]/(float)sqrt(2) ;
    if (-XIMAG[j]>0){
        ARGX[j] = (atan2(-XIMAG[j],XREAL[j]))*(180/pi))-20; //-26.02
    }
    else{
        ARGX[j] = (atan2(-XIMAG[j],XREAL[j]))*(180/pi))+360-20;
    }
    if (-YIMAG[j]>0){
        ARGY[j] = (atan2(-YIMAG[j],YREAL[j]))*(180/pi)); //16.2
    }
    else{
        ARGY[j] = (atan2(-YIMAG[j],YREAL[j]))*(180/pi))+360;
    }
    f[j] = ARGY[j]-ARGX[j];
}

Vrms[c]=sqrt(((float)(CX[0]*CX[0])+(CX[1]*CX[1])+(CX[2]*CX[2])+(CX[3]*CX[3]))/2);
Irms[c]=sqrt(((float)(CY[0]*CY[0])+(CY[1]*CY[1])+(CY[2]*CY[2])+(CY[3]*CY[3]))/2);
Apower[c]=((float)(CX[0]*CY[0])*cos(f[0]*pi/180)/2)+((float)(CX[1]*CY[1])*cos(f[1]*pi/180)
/2)+((float)(CX[2]*CY[2])*cos(f[2]*pi/180)/2)+((float)(CX[3]*CY[3])*cos(f[3]*pi/180)/2);

```



```

    Rpower[c]=((float)(CX[0]*CY[0])*sin(f[0]*pi/180)/2)+((float)(CX[1]*CY[1])*sin(f[1]*pi/180)/
    2)+((float)(CX[2]*CY[2])*sin(f[2]*pi/180)/2)+((float)(CX[3]*CY[3])*sin(f[3]*pi/180)/2);
    Spower[c]=sqrt((float)(Apower[c]*Apower[c])+(Rpower[c]*Rpower[c]));
    PowerF[c]=(Apower[c]/Spower[c]);
    Theta[c]=acos((float)PowerF[c])*180/pi;

    if(Rpower[c]>0){
        Cfp[c]=2;
    }
    else{
        Cfp[c]=1;
    }

    Theta[c]=Theta[c]*pow(-1,Cfp[c]);
}
return 0;
}
void Pmas(){
    pantalla++;
    if (pantalla>6){
        pantalla=1;
    }
}
void Pmin(){
    pantalla--;
    if (pantalla<1){
        pantalla=6;
    }
}
}

```

Apéndice

B

Código de MATLAB

```
clc
clear all
close all

load V1.mat
load V2.mat
load V3.mat
load I1.mat
load I2.mat
load I3.mat
load PF1.mat
load PF2.mat
load PF3.mat
load S1.mat
load S2.mat
load S3.mat
load P1.mat
load P2.mat
load P3.mat
load Q1.mat
load Q2.mat
load Q3.mat
load temperatura.mat

figure(1)
title ('Voltajes RMS por línea')
subplot (3,1,1)
plot (V1(1,:),V1(2,:))
```

```
xlabel('tiempo discreto')
ylabel('Voltaje 1')
axis ([0,V1(1,length(V1(1,:))),0,150])
grid on
```

```
subplot (3,1,2)
plot (V2(1,:),V2(2,:))
xlabel('tiempo discreto')
ylabel('Voltaje 2')
axis ([0,V2(1,length(V2(1,:))),0,150])
grid on
```

```
subplot (3,1,3)
plot (V3(1,:),V3(2,:))
xlabel('tiempo discreto')
ylabel('Voltaje 3')
axis ([0,V3(1,length(V3(1,:))),0,150])
grid on
```

```
figure(2)
title ('Corrientes RMS por línea')
subplot (3,1,1)
plot (I1(1,:),I1(2,:))
xlabel('tiempo discreto')
ylabel('Corriente 1')
axis ([0,I1(1,length(I1(1,:))),0,6])
grid on
```

```
subplot (3,1,2)
plot (I2(1,:),I2(2,:))
xlabel('tiempo discreto')
ylabel('Corriente 2')
axis ([0,I2(1,length(I2(1,:))),0,6])
grid on
```

```
subplot (3,1,3)
plot (I3(1,:),I3(2,:))
```

```
xlabel('tiempo discreto')
ylabel('Corriente 3')
axis ([0,I3(1,length(I3(1,:))),0,6])
grid on

figure(3)
title ('Factor de potencia por línea')
subplot (3,1,1)
plot (PF1(1,:),PF1(2,:))
xlabel('tiempo discreto')
ylabel('F.P. 1')
axis ([0,PF1(1,length(PF1(1,:))),0,1])
grid on

subplot (3,1,2)
plot (PF2(1,:),PF2(2,:))
xlabel('tiempo discreto')
ylabel('F.P. 2')
axis ([0,PF2(1,length(PF2(1,:))),0,1])
grid on

subplot (3,1,3)
plot (PF3(1,:),PF3(2,:))
xlabel('tiempo discreto')
ylabel('F.P. 3')
axis ([0,PF3(1,length(PF3(1,:))),0,1])
grid on

figure(4)
title ('Potencia Aparente RMS por línea')
subplot (3,1,1)
plot (S1(1,:),S1(2,:))
xlabel('tiempo discreto')
ylabel('Potencia Aparente 1')
axis ([0,S1(1,length(S1(1,:))),0,200])
grid on
subplot (3,1,2)
```

```
plot (S2(1,:),S2(2,:))
xlabel('tiempo discreto')
ylabel('Potencia Aparente 2')
axis ([0,S2(1,length(S2(1,:))),0,200])
grid on

subplot (3,1,3)
plot (S3(1,:),S3(2,:))
xlabel('tiempo discreto')
ylabel('Potencia Aparente 3')
axis ([0,S3(1,length(S3(1,:))),0,200])
grid on

figure(5)
title ('Potencia Activa RMS')
subplot (3,1,1)
plot (P1(1,:),P1(2,:))
xlabel('tiempo discreto')
ylabel('Potencia Activa 1')
axis ([0,P1(1,length(P1(1,:))),0,200])
grid on

subplot (3,1,2)
plot (P2(1,:),P2(2,:))
xlabel('tiempo discreto')
ylabel('Potencia Activa 2')
axis ([0,P2(1,length(P2(1,:))),0,200])
grid on

subplot (3,1,3)
plot (P3(1,:),P3(2,:))
xlabel('tiempo discreto')
ylabel('Potencia Activa 3')
axis ([0,P3(1,length(P3(1,:))),0,200])
grid on
```

```
figure(6)
title ('Potencia Reactiva RMS')
subplot (3,1,1)
plot (Q1(1,:),Q1(2,:))
xlabel('tiempo discreto')
ylabel('Potencia Reactiva 1')
axis ([0,Q1(1,length(Q1(1,:))),-200,0])
grid on

subplot (3,1,2)
plot (Q2(1,:),Q2(2,:))
xlabel('tiempo discreto')
ylabel('Potencia Reactiva 2')
axis ([0,Q2(1,length(Q2(1,:))),-200,0])
grid on

subplot (3,1,3)
plot (Q3(1,:),Q3(2,:))
xlabel('tiempo discreto')
ylabel('Potencia Reactiva 3')
axis ([0,Q3(1,length(Q3(1,:))),-200,0])
grid on

figure (7)
plot (temperatura(1,:),temperatura(2,:))
xlabel('tiempo discreto')
ylabel('Temp. °C')
axis ([0,temperatura(1,length(temperatura(1,:))),0,100])
grid on
```