

UNIVERSIDAD NACIONAL DE JAÉN

**CARRERA PROFESIONAL DE INGENIERÍA MECÁNICA Y
ELÉCTRICA**



**UNIVERSIDAD NACIONAL
DE JAÉN**

**SIMULACIÓN DE MOTOR DE CORRIENTE DIRECTA
MEDIANTE MODELO MATEMÁTICO Y
COMPARACIÓN CON MEDICIONES REALES**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
MECÁNICO ELECTRICISTA.**

Autores:

- Bach. Jhon Darwin Rimapa Roncal.
- Bach. Eliseo Montoya Pintado.

Asesores:

- Mg. Henry Oswaldo Pinedo Nava.
- Dr. Nilthon Arce Fernández.

Línea de investigación:

- LI_IME_03_Automatización y control.

JAÉN – PERÚ, Noviembre de 2024.

NOMBRE DEL TRABAJO

**SIMULACIÓN DE MOTOR DE CORRIENTE
DIRECTA MEDIANTE MODELO MATEMÁ
TICO Y COMPARACIÓN CON MEDICIONE
S REA**

AUTOR

**Jhon Darwin Rimapa Roncal y Eliseo M
ontoya Pintado**

RECUENTO DE PALABRAS

14909 Words

RECUENTO DE CARACTERES

80580 Characters

RECUENTO DE PÁGINAS

84 Pages

TAMAÑO DEL ARCHIVO

3.6MB

FECHA DE ENTREGA

Oct 3, 2024 11:13 AM GMT-5

FECHA DEL INFORME

Oct 3, 2024 11:14 AM GMT-5

● **12% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 10% Base de datos de Internet
- Base de datos de Crossref
- 10% Base de datos de trabajos entregados
- 2% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● **Excluir del Reporte de Similitud**

- Material bibliográfico
- Coincidencia baja (menos de 15 palabras)

 UNIVERSIDAD NACIONAL DE JAÉN

Dr. Alexander Huamán Mera
Responsable de la Unidad de Investigación de la Facultad de Ingeniería



LEY DE CREACIÓN No. 29304

Universidad Licenciada con Resolución del Consejo Directivo No. 002-2018-SUNEDU/CD

ACTA DE SUSTENTACIÓN

En la Ciudad de Jaén el día 10 de Octubre del año 2024, siendo las 17:00 horas se reunieron los Integrantes del jurado:

Presidente: **OSCAR NÚÑEZ MORI**

Secretario: **JAIME ODAR HONORIO ACOSTA**

Vocal: **FRANS FUENTES MAZA**

Para evaluar la sustentación del Informe Final

() Trabajo de Investigación (X) Tesis () Trabajo de Suficiencia Profesional

Titulado "SIMULACIÓN DE MOTOR DE CORRIENTE DIRECTA MEDIANTE MODELO MATEMÁTICO Y COMPARACIÓN CON MEDICIONES REALES"

Presentado por el/(los) Estudiante(s) / egresado(s) O Bachiller(es) **Jhon Darwin Rimapa Roncal y Eliseo Montoya Pintado**, de la Escuela Profesional de Ingeniería Mecánica y Eléctrica.

Después de la Sustentación y Defensa el Jurado Acuerda:

(X) Aprobar () Desaprobar (X) Unanimidad () Mayoría

Con la siguiente mención:

Excelente	18, 19, 20	()
Muy bueno	16, 17	()
Bueno	14, <u>15</u>	(X)
Regular	13	()
Desaprobado	12 o menos	()

Siendo las 18:00 horas del mismo día el jurado concluye el acto de sustentación confirmando su participación con la suscripción de la presente.



Presidente



Secretario



Vocal

ÍNDICE

I. INTRODUCCIÓN	1
1.1. Objetivo general:	2
1.2. Objetivos específicos:	3
II. MATERIAL Y MÉTODOS	3
2.1. Materiales	3
2.1.1. Motor DC	3
2.1.2. Placa Arduino Mega 2560 R3	5
2.1.3. Sensor de corriente ACS712T-05A	6
2.1.4. Driver puente H BTS7960 43A	7
2.1.5. Micro sd card 74LVC125A	8
2.1.6. Módulo RTC DS3231 + EEPROM AT24C32	9
2.1.7. Multitester Digital Profesional Truper Mut-39	10
2.1.8. Multímetro LCR UNI-T UT612.	11
2.1.9. Osciloscopio Digital GDS2202E	11
2.1.10. Fuente de alimentación.....	12
2.1.11. Licencias de software.	12
2.1.12. Prototipo para las mediciones(maqueta)	12
2.2. Metodología	14
III. RESULTADOS	15
OE1: Identificar el modelo matemático que representará al motor real.	
3.1. Modelamiento matemático.	15
3.2. Modelado matemático con ecuaciones diferenciales	17
3.3. Modelación matemática con funciones de transferencia.....	18

OE2: Implementar un prototipo a nivel de pruebas de laboratorio del sistema para realizar las mediciones y obtener experimentalmente los parámetros del motor para la simulación.

3.4.	Adquisición experimental de parámetros	21
3.4.1.	Resistencia de armadura (R)	22
3.4.2.	Voltaje (V)	23
3.4.3.	Inductancia de armadura (L)	25
3.4.4.	Corriente sin carga (I)	25
3.4.5.	Velocidad (ω).....	28
3.4.6.	Constante electromotriz y de torque (C_c, C_{tq})	29
3.4.7.	Constante de tiempo mecánico (t_m)	30
3.4.8.	Momento de inercia (J)	32
3.4.9.	Corriente de arranque (i_{arr}).....	32
3.4.10.	Torque de fricción (T_f).....	33
3.4.11.	Torque del motor (T_m).....	34
3.4.12.	Constante de fricción de Coulomb (b).....	34
3.4.13.	Valores obtenidos de las mediciones.	35
3.5.	Implementación en Matlab/Simulink	36

OE3: Validación del modelo matemático con los parámetros obtenidos del experimento.

3.6.	Validación del modelado con un motor real	39
3.6.1.	Correlación de Pearson.	39
3.6.2.	Validación de la corriente.	40
3.6.3.	Validación de la velocidad.....	42

IV. DISCUSIÓN

V. CONCLUSIONES Y RECOMENDACIONES.....

VI. REFERENCIAS BIBLIOGRÁFICAS.....

VII. ANEXOS.....

ÍNDICE DE TABLAS

ATabla 1.	<i>Especificaciones técnicas del motor DC JGA25-370 de 12V/100RPM</i>	4
Tabla 2.	<i>Especificaciones técnicas de la Placa Arduino Mega 2560 R3</i>	5
Tabla 3.	<i>Especificaciones técnicas del ACS712T-20A</i>	6
Tabla 4.	<i>Ficha técnica del driver puente H BTS7960 43A</i>	7
Tabla 5.	<i>Especificaciones técnicas de la micro SD CARD 74LVC125A</i>	8
Tabla 6.	<i>Especificaciones técnicas del RTC DS3231 + EEPROM AT24C32 (I2C)</i>	9
Tabla 7.	<i>Especificaciones técnicas del Multitester Digital Profesional Truper Mut-39</i>	10
Tabla 8.	<i>Parámetros físicos de un motor.</i>	16
Tabla 9.	<i>Conexiones de pines.</i>	27
Tabla 10.	<i>Parámetros del motor DC 12V-100RPM</i>	35
Tabla 11.	<i>Leyenda de diagrama de bloques</i>	37
Tabla 12.	<i>Coeficiente de correlación de la escala “r” de Pearson</i>	40
Tabla 13	<i>Datos de los parámetros experimentales y simulados de la corriente y velocidad</i>	

ÍNDICE DE FIGURAS

Figura 1.	<i>Motor DC 12V 100rpm</i>	4
Figura 2.	<i>Pines del motor</i>	4
Figura 3.	<i>Arduino Mega 2560 R3</i>	5
Figura 4.	<i>Sensor de corriente ACS712T-05A</i>	6
Figura 5.	<i>Driver puente H BTS7960 43A.</i>	7
Figura 6.	<i>Micro sd card 74LVC24A</i>	8
Figura 7.	<i>Módulo RTC</i>	9
Figura 8.	<i>Multitester Digital Profesional Truper Mut-39</i>	10
Figura 9.	<i>Multímetro LCR UNI-T UT612.</i>	11
Figura 10.	<i>Osciloscopio Digital GDS2202E</i>	11
Figura 11.	<i>Fuente de alimentación 12V-5V</i>	12
Figura 12.	<i>Diagrama de conexiones para la maqueta</i>	13
Figura 13.	<i>Esquema para un motor de corriente directa.</i>	16
Figura 14.	<i>Diagrama de conexiones para medir la resistencia de armadura</i>	22
Figura 15.	<i>Circuito de divisor de voltaje</i>	23
Figura 16.	<i>Diagrama de conexión del circuito divisor de voltaje</i>	24
Figura 17.	<i>Resultados de la medición de voltaje</i>	24
Figura 18.	<i>Diagrama de conexión para medir la inductancia</i>	25
Figura 19.	<i>Diagrama de conexiones para la medición de la corriente</i>	26
Figura 20.	<i>Conexiones del circuito para medir corriente</i>	26
Figura 21.	<i>Resultados de la programación</i>	27
Figura 22.	<i>Diagrama de conexiones para la medición de RPM</i>	28
Figura 23.	<i>Resultados de la velocidad</i>	29
Figura 24.	<i>Circuito de un motor DC simplificado</i>	29
Figura 25.	<i>Conexiones de osciloscopio</i>	31
Figura 26.	<i>Medición de la constante del tiempo mecánico</i>	31

Figura 27. Diagrama de conexiones para la corriente de arranque	33
Figura 28. Diagrama de bloques de las Ecuaciones Diferenciales	36
Figura 29. Implementación del modelo matemático en Matlab/Simulink	37
Figura 30.....	38
Comparación de la corriente respecto al tiempo	38
Figura 31. Comparación de la velocidad respecto al tiempo	38
Figura 32. Validación gráfica de la corriente	41
Figura 33. Validación de la corriente mediante la correlación de Pearson.....	42
Figura 34. Validación gráfica de la velocidad.....	43
Figura 35 Validación de la velocidad mediante la correlación de Pearson	43

RESUMEN

La simulación de motores es una técnica inestimable para mejorar el rendimiento y la eficiencia, examinar el comportamiento de los motores en diversas condiciones y agilizar el diseño y el control de sistemas electromecánico. El presente estudio tiene como objetivo simular un motor de corriente directa mediante modelo matemático y compararlo con mediciones reales respectivas. En el proceso de validación del modelo, el valor del coeficiente de correlación de Pearson obtenido para la variable de corriente resultó ser medio, mientras que el obtenido para la variable de velocidad fue alto, no obstante, ambos valores se consideraron aceptables para los fines de la investigación. Los resultados de esta investigación sientan las bases para futuras investigaciones sobre la simulación y el análisis de motores de corriente directa, permitiendo la realización de experimentos virtuales que pueden suponer un ahorro sustancial de tiempo y recursos. Para cualquier sistema o mecanismo físico auténtico se puede desarrollar su correspondiente modelo matemático y posteriormente su simulación. Por tanto, este trabajo pretende avanzar en la estandarización de este ejercicio de modelización y simulación matemática para ser implementado como soporte didáctico de cursos impartidos en el laboratorio de Ingeniería mecánica y Eléctrica (Laboratorio de Medición de Recursos Energéticos Renovables).

Palabras claves: Motor de corriente directa, modelo matemático, simulación, programación.

ABSTRACT

Engine simulation is an invaluable technique for improving performance and efficiency, examining engine behavior under various conditions, and streamlining the design and control of electromechanical systems. The present study aims to simulate a direct current motor using a mathematical model and compare it with respective real measurements. In the model validation process, the value of the Pearson correlation coefficient obtained for the current variable turned out to be medium, while that obtained for the speed variable was high, however, both values were considered acceptable for the purposes of the investigation. The results of this research lay the foundation for future research on the simulation and analysis of direct current motors, allowing virtual experiments to be carried out that can lead to substantial savings in time and resources. For any authentic physical system or mechanism, its corresponding mathematical model can be developed and subsequently simulated. Therefore, this work aims to advance the standardization of this mathematical modeling and simulation exercise to be implemented as didactic support for courses taught in the Mechanical and Electrical Engineering laboratory (Renewable Energy Resources Measurement Laboratory).

Keywords: Direct current motor, mathematical model, simulation, programming.

I. INTRODUCCIÓN

La aplicación de motores DC abarca una amplia variedad de sectores en las actividades humanas, que van desde la industria, el ámbito académico, la juguetería, entre otros. Presentan movimiento angular, lineal, asimismo permiten el traslado de objetos o de personas de un punto a otro.(Valdivia, 2021)

Los motores de corriente directa (DC) son dispositivos fundamentales en numerosas aplicaciones industriales y sistemas de control, debido a su eficiencia, alta capacidad de control y respuesta rápida. Sin embargo, la falta de conocimiento profundo sobre sus bases teóricas y dificultades en su dimensionamiento, diseño y control han llevado a la necesidad de investigaciones que aborden estos desafíos. Por lo que, en este contexto, el presente proyecto tiene como objetivo principal simular y comparar un motor de corriente directa mediante un modelo matemático, con el fin de mejorar la comprensión de su funcionamiento y optimizar su rendimiento en diversas aplicaciones.

Quispe (2022), en su tesis plantea que los motores DC son utilizados en diferentes aplicaciones, debido a que su control es sencillo por eso es importante conocer su funcionamiento. La modelación de motores DC es esencial para optimizar su rendimiento y eficiencia, predice el comportamiento de los motores bajo diversas condiciones lo cual facilita el diseño y control en sistemas mecánicos y eléctricos. Aumenta la precisión en aplicaciones industriales y de automatización (Choe et al. 2021). Según Sheng et al. (2023), el control preciso de la velocidad facilita la integración en diversos sistemas. Hoy en día, los investigadores se dedican a desarrollar modelos que puedan comunicar un sistema físico en tiempo real. Sun (2022), destaca la importancia de la simulación para obtener información precisa sobre la corriente y la velocidad del rotor, lo que facilita el control. La simulación es esencial para el desarrollo y validación de modelos dinámicos de motores que permite la integración precisa de

múltiples parámetros. El modelamiento y simulación en un entorno controlado es esencial para mejorar el rendimiento y la eficiencia del motor (Wang et al. 2022)

El poco conocimiento de los principios básicos, funcionamiento y operación de motores DC se puede catalogar como un problema debido a que generalmente los motores eléctricos son operados con accionamientos ON/OFF (llaves interruptoras termomagnéticas en el mejor de los casos) lo cual conlleva a desarrollar enormes corrientes de arranque que no solo pueden dañar el motor sino a los dispositivos conectados en la misma red eléctrica. Por lo que es importante lograr el control analógico o digital de un motor DC en cualquier entorno que se utilice.

En este trabajo de investigación se presentan modelos matemáticos que nos ayudan a entender el comportamiento de los motores DC, para lo cual se hizo uso del software MATLAB, con sus respectivas herramientas, dando como resultado el desarrollo de un modelo matemático preciso y confiable para la simulación de motores de corriente directa. El modelo se basó en la descripción de las interacciones electromagnéticas y mecánicas que tienen lugar en el motor, teniendo en cuenta los principales parámetros que influyen en su rendimiento como la corriente de excitación, el par de carga, la inductancia y la resistencia del devanado entre otros.

Se espera que este modelo matemático sirva como base para futuras investigaciones en el campo de la simulación y el análisis de motores de corriente directa asimismo que facilite la experimentación virtual, lo que implicaría un ahorro significativo de tiempo y recursos.

La presente investigación tiene como objetivos:

1.1. Objetivo general:

- Simular un motor de corriente directa mediante modelo matemático y compararlo con mediciones reales respectivas.

1.2. Objetivos específicos:

- OE1: Identificar el modelo matemático que representará al motor real.
- OE2: Implementar un prototipo a nivel de pruebas de laboratorio del sistema para realizar las mediciones y obtener experimentalmente los parámetros del motor para la simulación.
- OE3: Validación del modelo matemático con los parámetros obtenidos del experimento.

II. MATERIAL Y MÉTODOS

2.1. Materiales

El presente proyecto forma parte del proyecto de investigación docente “Simulador de motor eléctrico mediante modelos matemáticos para el laboratorio de la carrera profesional de Ingeniería Mecánica y Eléctrica de la UNJ” aprobado con resolución N° 303-2022-UNJ, por lo que para el equipamiento de materiales se ha seguido las recomendaciones de dicho proyecto.

2.1.1. Motor DC

El motor que se utilizó para este proyecto fue un motor DC 12V 100RPM, que trae incluido un encoder de efecto hall el cual nos facilitó la medición de las revoluciones por minuto (RPM). Dicho motor se aprecia en la Figura 1. En la Tabla 1 se muestra las características más relevantes proporcionadas por la ficha técnica del fabricante y en la Figura 2 se aprecia los pines para la conexión del motor y de su encoder.

Figura 1.

Motor DC 12V 100rpm



Tabla 1.

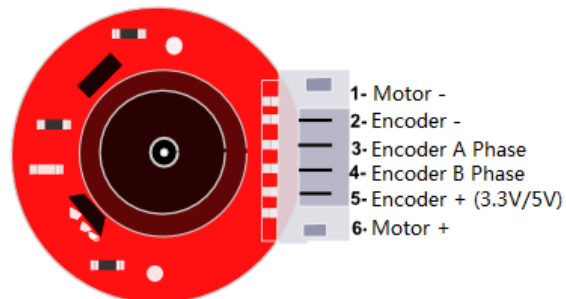
Especificaciones técnicas del motor DC JGA25-370 de 12V/100RPM

Características	Valores
Voltaje de alimentación del motor	5V - 12V DC
Voltaje de alimentación del encoder	3.3V - 5V DC
Consumo de corriente sin carga	100mA
Consumo de corriente nominal	300mA
Consumo de corriente eje detenido	1000mA (Pico)
Velocidad de rotación	100 RPM @ 12V
Relación en engranes	1:119
Encoder	Hall de dos fases
Resolución Hall	Hall x Ratio 119 = 1194 PPR
Diámetro del eje	4 mm, eje en forma de D
Diámetro exterior de la carcasa	25 mm
Dimensiones	D25mm*L71mm
Peso	95 gramos

Nota: Estos datos han sido tomados de la ficha técnica del motor.

Figura 2.

Pines del motor



Nota: Esta figura se ha tomado de la ficha técnica del motor.

2.1.2. Placa Arduino Mega 2560 R3

Para el control de la programación se utilizó una tarjeta de desarrollo Arduino Mega 2560 R3 la cual se puede apreciar en la Figura 3. En la Tabla 2 se muestra sus características de dicha placa. Esta placa fue necesaria por su mayor capacidad de memoria para soportar el código de programación necesario.

Figura 3.
Arduino Mega 2560 R3

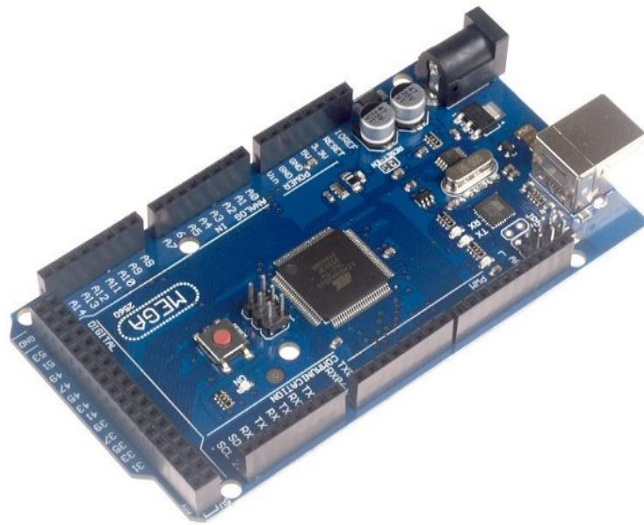


Tabla 2.
Especificaciones técnicas de la Placa Arduino Mega 2560 R3

Características	Valores
Microcontrolador	ATmega2560 (8-bit)
Chip USB	ATmega16U2
Conector USB	Tipo B
Voltaje de operación	5V DC
Voltaje de alimentación	6V - 20V DC (7-12V recomendado)
Pines digitales I/O	54 (15 salidas PWM)
Entradas analógicas	16 (ADC 10-bit)
Corriente entrada/salida por pin	40mA máx
Memoria FLASH	256KB
Memoria SRAM	8KB
Memoria EEPROM	4KB
Frecuencia de reloj	16MHz

Nota: Adaptado de datos del fabricante.

2.1.3. Sensor de corriente ACS712T-05A

Para medir la corriente que consume el motor se usó el sensor ACS712T-05A que a través de la programación en Arduino nos facilitó medir el consumo de corriente del motor DC, este sensor se puede apreciar en la Figura 4, y en la Tabla 3 se especifica sus datos más relevantes.

Figura 4.
Sensor de corriente ACS712T-05A

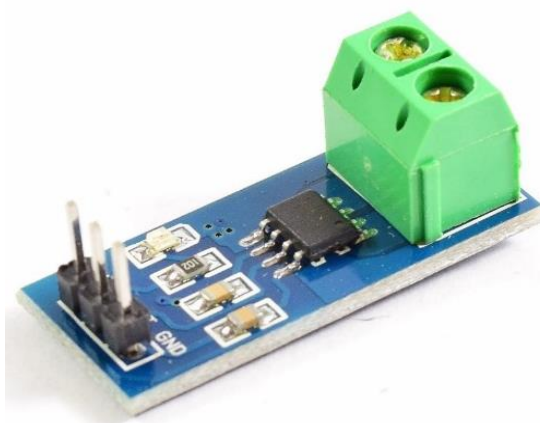


Tabla 3.
Especificaciones técnicas del ACS712T-20A

Características	Valores
Voltaje de alimentación	5VDC
Rango de sensado de corriente	- 05A a +05A
Rango de salida analógica	0.5V a 4.5V
Sensibilidad	185mV/A
Señal analógica de bajo ruido	---
Ancho de banda configurable mediante el pin FILTER	---
Tiempo de respuesta de la salida	5us
Ancho de banda máximo	80kHz
Error	±1.5% (@25 °C)
Resistencia del conductor	1.2mOhm
Mide tanto corrientes AC como DC	---
Offset de salida sumamente estable	2.5V

Nota: Adaptada de datos del fabricante.

2.1.4. Driver puente H BTS7960 43A

Sirve para controlar motores DC (Voltaje de ingreso, sentido de giro, entre otros), capaz de proporcionar hasta 43A de corriente a una tensión de alimentación de entre 6-27 V. En este proyecto se utilizó para realizar cambios en el voltaje que ingresa al motor, para así ver las variaciones que sufre este al estar sometido a distintas tensiones. En la Figura 5 se puede apreciar dicho driver y en la Tabla 4 se especifica sus datos técnicos.

Figura 5.

Driver puente H BTS7960 43A.

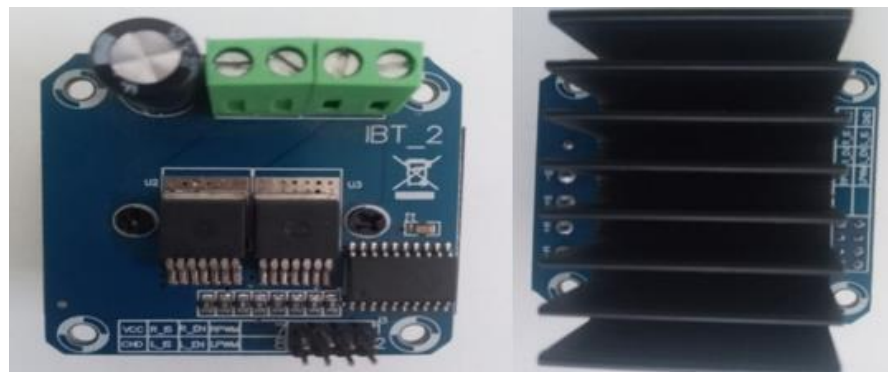


Tabla 4.

Ficha técnica del driver puente H BTS7960 43A

Características	Valores
Voltaje de alimentación (potencia)	5.5V-27V DC
Voltaje de control	TTL 3.3V/5V
Capacidad de corriente	43A (picos de hasta 60A)
Máxima frecuencia PWM	25 KHz
Consumo de corriente inactivo	7uA a 25°C
Modelo	HW-039
Chip Driver	BTS7960B
Dimensiones	50*50*43 mm
Peso	66 gramos

Nota: Adaptada de datos del fabricante.

2.1.5. Micro sd card 74LVC125A

La micro sd se utilizó para el registro de los datos. La cual nos permite grabar datos en memorias sd formato fat y así poder tener un registro de los parámetros obtenidos del motor DC. En la Figura 6 se aprecia la micro sd card y en la Tabla 5 sus especificaciones técnicas.

Figura 6.

Micro sd card 74LVC24A

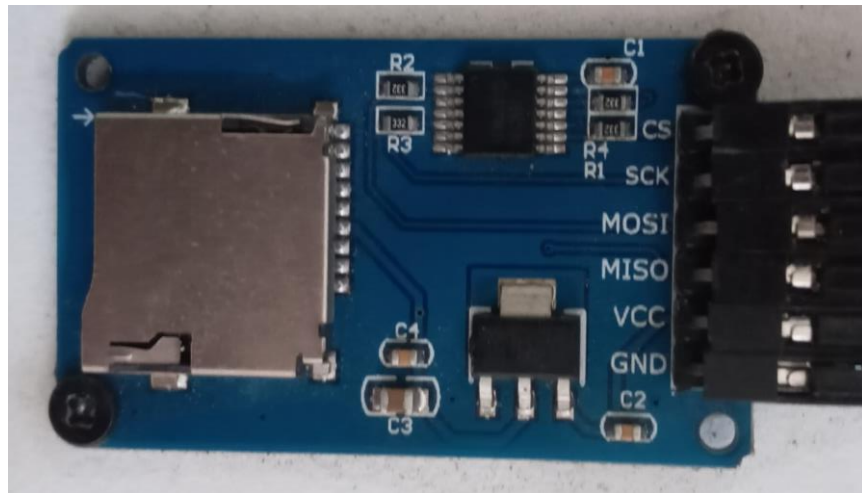


Tabla 5.

Especificaciones técnicas de la micro SD CARD 74LVC125A

Características	Valores
Interfaz	SPI
Voltaje de control	TTL 3.3V/5V
Cuenta con todos los pines SPI de la tarjeta SD	MOSI, MISO, SCK, CS
Chip conversor lógico	MC74VHCT125A/74LVC125A
Dimensiones	42*24 mm

Nota: Adaptada de datos del fabricante.

2.1.6. Módulo RTC DS3231 + EEPROM AT24C32

Este módulo se utilizó para la integración de mediciones de tiempo a nuestro proyecto. En la figura 7 se puede apreciar dicho módulo y en la tabla 6 se muestra su ficha técnica.

Figura 7.
Módulo RTC

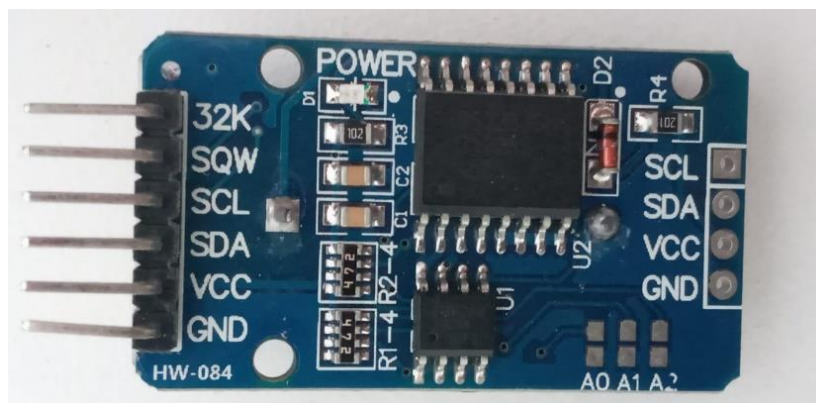


Tabla 6.
Especificaciones técnicas del RTC DS3231 + EEPROM AT24C32 (I2C)

Características	Valores
Voltaje de alimentación	3.3V - 5V DC
Exactitud Reloj	2ppm
Interfaz de comunicación digital	I2C
Dirección I2C del DS3132	Read(11010001) Write(11010000)
Memoria EEPROM AT24C32	(4K * 8bit = 32Kbit = 4KByte)
La batería puede mantener al RTC funcionando por 10 años	HW-039
Dimensiones	38*22 mm
Peso	5 gramos

Nota: Adaptada de datos del fabricante.

2.1.7. Multitester Digital Profesional Truper Mut-39

En primera instancia este equipo se utilizó para medir los parámetros del motor y para así poder validarlos con los resultados obtenidos de la programación. En la Figura 8 se puede apreciar dicho instrumento y en la Tabla 7 se detalla sus especificaciones técnicas.

Figura 8.
Multitester Digital Profesional Truper Mut-39



Tabla 7.
Especificaciones técnicas del Multitester Digital Profesional Truper Mut-39

Características	Valores
Tensión CC / CA	200 mV – 1,000 V / 2V – 750 V
Corriente CC / CA	2 mA – 20 A / 200 mA – 20 A
Resistencia	200 Ohms – 20 M Ohms
Temperatura (°C)	-40° C a 1,000° C
Capacitancia	20 nF – 20 mF
Frecuencia	0 – 60 MHz
Modelo	MUT-39

Nota: Adaptada de datos del fabricante.

2.1.8. Multímetro LCR UNI-T UT612.

Este instrumento se utilizó para medir la inductancia de los devanados del motor.

La Figura 9 muestra dicho instrumento:

Figura 9.
Multímetro LCR UNI-T UT612.



2.1.9. Osciloscopio Digital GDS2202E

Este instrumento se utilizó para medir la constante de tiempo mecánica. En la figura 10 se aprecia dicho instrumento.

Figura 10.
Osciloscopio Digital GDS2202E



Nota: Adaptado de la ficha del fabricante.

2.1.10. Fuente de alimentación

Se hizo uso de una fuente de poder de una PC que nos proporciona 12V-5V en DC. en la Figura 11 se puede apreciar dicha fuente.

Figura 11.

Fuente de alimentación 12V-5V



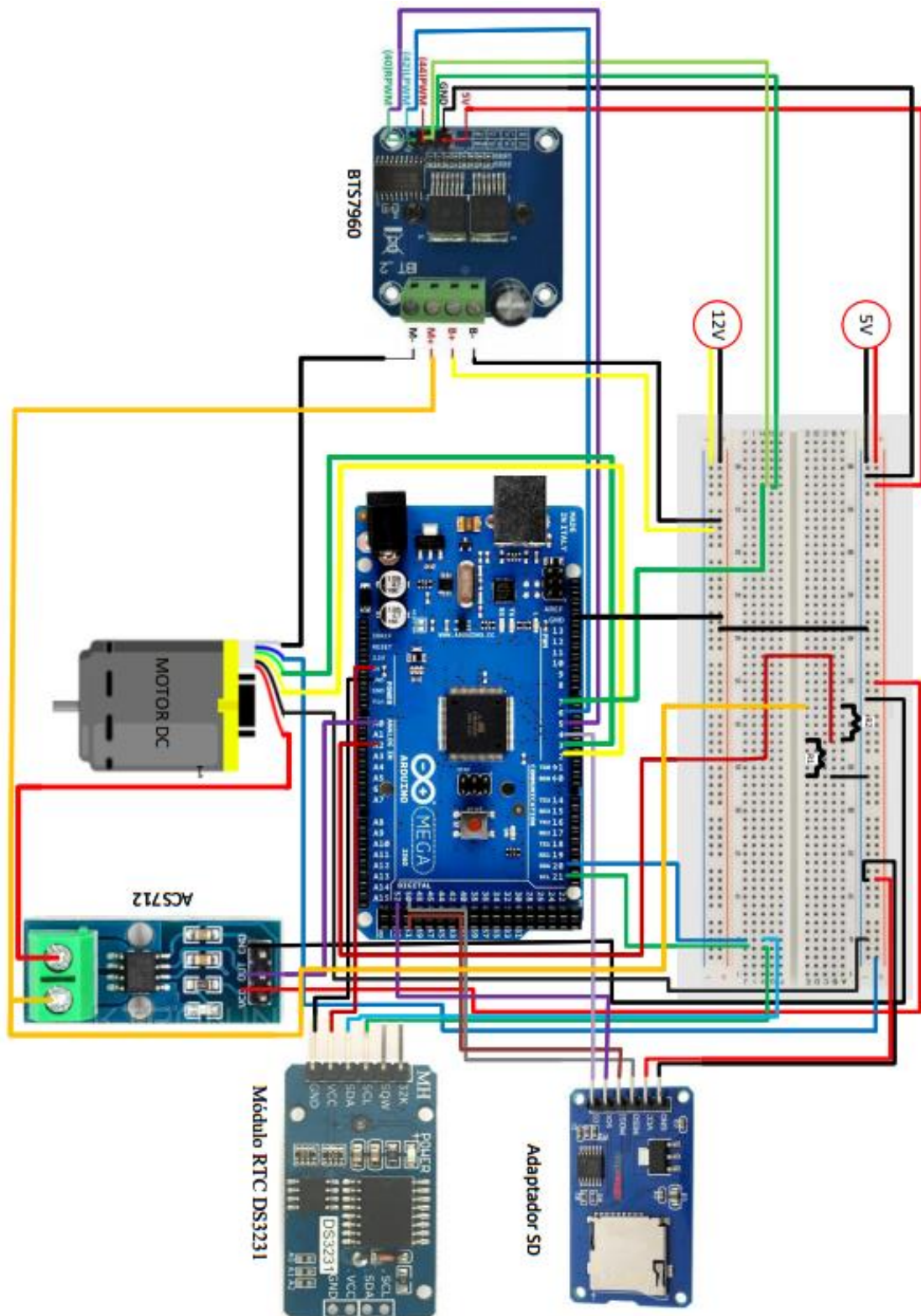
2.1.11. Licencias de software.

Para la redacción del informe final se utilizó el software Microsoft Office del Instituto de Investigación en Energías Renovables, para la programación se utilizó los softwares: MATLAB con licencia versión de prueba estudiantil, Arduino y Python con licencia libre.

2.1.12. Prototipo para las mediciones(maqueta)

Para las mediciones que se realizaron implementamos una maqueta con todas las conexiones que nos permitieron obtener los parámetros del motor. Las conexiones para la maqueta se aprecian en la figura 12 y el prototipo junto con el circuito eléctrico electrónico se aprecia en anexos.

Figura 12.
Diagrama de conexiones para la maqueta



2.2. Metodología

En la presente investigación se utilizó el método analítico y el método experimental para definir algunos parámetros internos del proyecto global que se ha orientado principalmente para desarrollo tecnológico.

Para la identificación del modelo matemático que representó al motor real se siguió los pasos:

- Definición del modelo matemático base factible del motor DC real seleccionado, el cual nos ayudó a cumplir con nuestros objetivos propuestos, procedimiento y conceptos presentados en la sección de resultados más adelante (Ver Figura 13 para detalle del modelo considerado en el presente trabajo).

Para la implementación de un prototipo a nivel de pruebas de laboratorio que permita realizar las mediciones y obtener los parámetros del motor para la simulación, se siguió los pasos:

- Selección de equipos para la medición que nos ayudó en la recolección de los datos requeridos (una placa Arduino Mega 2560, presentado en la sección 2.1.2; un sensor de corriente, presentado en la sección 2.1.3; un driver puente H, presentado en la sección 2.1.4; una micro SD, presentado en la sección 2.1.5; un módulo RTC, presentado en la sección 2.1.6; un multímetro, presentado en la sección 2.1.7, un multímetro LCR, presentado en la sección 2.1.8 y un osciloscopio digital, presentado en la sección 2.1.9)
- Diseño e implementación del sistema de medición de corriente y voltaje de entrada del motor con sensores y controlador Arduino mediante programación, dicho diseño se detalla en anexos.
- Luego de la selección de los equipos y el diseño del sistema de medición (maqueta) procedimos a realizar el experimento y pruebas de laboratorio donde se obtuvo los parámetros que sirvieron para la implementación en el modelo matemático.
- Los datos obtenidos en laboratorio fueron codificados, preparados y archivados en una tabla Excel, quedando todo listo para su procesamiento mediante el software Matlab/Simulink (Ver sección de resultados para mayor detalle).
- También el procedimiento y detalles de la implementación serán presentados en la sección de resultados.

Para la validación del modelo matemático con los parámetros obtenidos del experimento, se realizó la comparación de resultados (modelo matemático VS motor real). Para ello, se utilizó validación gráfica y validación usando el coeficiente de correlación lineal de Pearson implementado en el lenguaje de programación Python. Se usó dicha correlación, puesto que nos permite medir la fuerza de la relación lineal entre dos variables.

III. RESULTADOS

OE1: Identificar el modelo matemático que representará al motor real

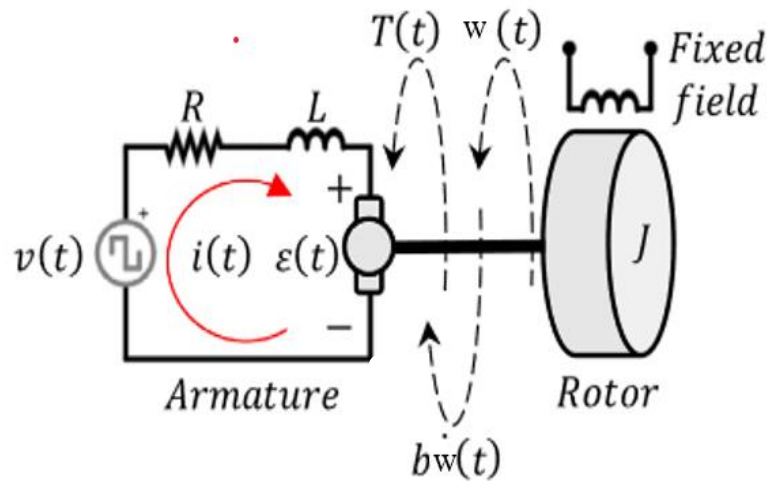
3.1. Modelamiento matemático.

El modelamiento matemático de cualquier sistema físico como el de un motor DC permite analizar el comportamiento de componentes físicos interconectados en un circuito eléctrico con el propósito de evaluar su funcionamiento. Los componentes físicos de un circuito, como una resistencia, una inductancia, y capacitor, tienen expresiones continuas matemáticas que son evaluadas en función del tiempo haciendo uso de ecuaciones diferenciales (Paulo et al., 2017).

En el modelamiento matemático del motor DC se consideró parámetros eléctricos (voltaje, resistencia, inductancia de armadura, fuerza electromotriz y corriente), parámetros mecánicos (Torque del motor, velocidad angular, coeficiente de fricción y momento de inercia) y parámetros electromecánicos (constante de fuerza electromotriz y constante del torque). En la Figura 13 se muestra un esquema general de un motor DC y en la Tabla 8 sus parámetros físicos.

Figura 13.

Esquema para un motor de corriente directa.



Nota. Esta figura ha sido tomada de “Didactic platform for DC motor speed and position control in Z-plane”, por Cabré et al. (2021). ISA Transactions, 118, 116-132.

Tabla 8.

Parámetros físicos de un motor.

Símbolo	Descripción	Unidades
$w(t)$	Posición angular del eje	rad
$T(t)$	Torque	Nm
$v(t)$	Voltaje de entrada	V
$i(t)$	Corriente	A
$\varepsilon(t)$	Fuerza electromotriz	V
R	Resistencia equivalente	Ω
L	Inductancia equivalente	H
j	Momento de inercia	$\text{Kg } m^2$
b	Constante de fricción viscosa del motor	N m s
C_c	Constante de fuerza electromotriz	Vrad/s
C_{tq}	Constante de par del motor	N m/A

Nota. Esta tabla ha sido tomada de “Didactic platform for DC motor speed and position control in Z-plane”, por Cabré et al. (2021). ISA Transactions, 118, 116-132.

3.2. Modelado matemático con ecuaciones diferenciales

Mediante un análisis de malla del circuito realizada a la Figura 13. se obtuvo la primera ecuación para el modelado matemático:

$$v(t) = Ri(t) + L \frac{di(t)}{dt} + E_a(t) \quad (1)$$

$$L \frac{di(t)}{dt} = v(t) - Ri(t) - E_a(t) \quad (2)$$

Donde:

L : Inductancia

$v(t)$: Voltaje

R : Resistencia

$i(t)$: Corriente

$E_a(t)$: Fuerza contraelectromotriz

La potencia mecánica que desarrolla el rotor es entregada a la carga mecánica conectado al eje del motor DC, pero parte de esta potencia desarrollada se pierde a través de la resistencia de la bobina del rotor, la fricción, por histéresis y perdidas por corrientes de Foucault en el hierro del rotor. Por lo que la ecuación para la sección mecánica del motor, está dada por:

$$T_m(t) = j \frac{dw(t)}{dt} + bw(t) \quad (3)$$

$$j \frac{dw(t)}{dt} = T_m(t) - bw(t) \quad (4)$$

Donde:

j : Momento de inercia

$T_m(t)$: Torque del motor

b : Coeficiente de fricción

$w(t)$: Velocidad angular

$\frac{dw(t)}{dt}$: Aceleración angular

Con la finalidad de interaccionar las ecuaciones (2) y (4) asumimos que existe una relación proporcional C_c : constante contraelectromotriz [$v / rad * s$], que relaciona el voltaje inducido en la armadura $E_a(t)$ y la velocidad angular del eje del motor $w(t)$.

$$E_a(t) = C_c w(t) \quad (5)$$

Por otro lado, asumimos una relación electromecánica C_{τ} : constante de torque [Nm / A], entre el torque mecánico $T_m(t)$ y la corriente eléctrica $i(t)$

$$T_m(t) = C_{\tau} i(t) \quad (6)$$

3.3. Modelación matemática con funciones de transferencia

Meneses & Zafra (2013), describen la función de transferencia de un sistema, modelado por una ecuación diferencial lineal e invariante en el tiempo, como la relación entre la transformada de Laplace de la salida y la transformada de Laplace de la entrada, asumiendo que todas las condiciones iniciales son nulas.

Según Flores (2021), la transformada de Laplace es una herramienta muy poderosa en ingeniería especialmente en el análisis y diseño de sistemas lineales. Se utiliza para transformar ecuaciones diferenciales lineales ordinarias en ecuaciones algebraicas, lo cual simplifica su análisis y solución.

En el presente trabajo se utiliza la transformada de Laplace para obtener ecuaciones algebraicas simples que son mucho más fáciles de resolver, analizar y comprender.

Procedemos aplicando la transformada de Laplace a la ecuación (2)

$$l[L \frac{di(t)}{dt}] = l[v(t) - Ri(t) - E_a(t)] \quad (7)$$

Obtenemos la función de transferencia:

$$Lsi(s) = v(s) - Ri(s) - E_a(s) \quad (8)$$

Aplicando la transformada de Laplace a la ecuación (4):

$$l[J \frac{dw(t)}{dt}] = l[T_m(t) - bw(t)] \quad (9)$$

Obtenemos la función de transferencia:

$$Jsw(s) = T_m(s) - bw(s) \quad (10)$$

Aplicando la transformada de Laplace a la ecuación (5):

$$l[E_a(t)] = l[C_C W(t)] \quad (11)$$

Obtenemos la función de transferencia:

$$E_a(s) = C_C w(s) \quad (12)$$

Aplicando la transformada de Laplace a la ecuación (6):

$$l[t_m(t)] = l[C_{td} i(t)] \quad (13)$$

Obtenemos la función de transferencia:

$$T_m(s) = C_{iq}i(s) \quad (14)$$

Sustituyendo ecuaciones (12) y (14) en la ecuación (8) se obtuvo:

$$Ls \frac{T_m(s)}{C_{iq}} = v(s) - R \frac{T_m(s)}{C_{iq}} - C_c w(s) \quad (15)$$

De donde el voltaje de entrada es dado por:

$$v(s) = (Ls + R) \frac{T_m(s)}{C_{iq}} + C_c w(s) \quad (16)$$

Para obtener la velocidad angular despejamos $w(s)$ de la ecuación (10), es decir:

$$w(s) = \frac{T_m(s)}{Js + b} \quad (17)$$

Sustituyendo la ecuación (17) en la ecuación (16) se obtuvo:

$$v(s) = (Ls + R) \frac{T_m(s)}{C_{iq}} + C_c \left[\frac{T_m(s)}{Js + b} \right] \quad (18)$$

$$v(s) = \left[\frac{(Ls + R)}{C_{iq}} + \frac{C_c}{Js + b} \right] T_m(s) \quad (19)$$

$$v(s) = \frac{(Ls + R)(Js + b) + (C_c)(C_{iq})}{C_{iq}(Js + b)} T_m(s) \quad (20)$$

Luego, de la ecuación (20) se obtiene la función de transferencia que relaciona el torque del motor con el voltaje, es decir:

Función de transferencia: Torque del motor / Voltaje:

$$\frac{T_m(s)}{v(s)} = \frac{C_{iq}(Js + b)}{LJs^2 + (RJ + Lb)s + Rb + (C_c)(C_{iq})} \quad (21)$$

De manera análoga, usando las ecuaciones se obtuvo las funciones de transferencia que relacionan cualquier salida con la entrada que es el voltaje.

Función de transferencia: Fuerza contraelectromotriz / Voltaje:

$$\frac{E_a(s)}{v(s)} = \frac{(C_c)(C_{iq})}{LJs^2 + (RJ + Lb)s + Rb + (C_c)(C_{iq})} \quad (22)$$

Función de transferencia: corriente de armadura / Voltaje:

$$\frac{i(s)}{v(s)} = \frac{Js + b}{LJs^2 + (RJ + Lb)s + Rb + (C_c)(C_{iq})} \quad (23)$$

Función de transferencia: Velocidad angular / voltaje:

$$\frac{w(s)}{v(s)} = \frac{(C_{iq})}{LJs^2 + (RJ + Lb)s + Rb + (C_c)(C_{iq})} \quad (24)$$

Para conocer la posición del motor, se integró la velocidad angular, es decir, agregamos un integrador a la función de transferencia anterior:

Función de transferencia: posición del motor / voltaje:

$$\frac{\theta(s)}{v(s)} = \frac{(C_{iq})}{s(LJs^2 + (RJ + Lb)s + Rb + (C_c)(C_{iq}))} \quad (25)$$

OE2: Implementar un prototipo a nivel de pruebas de laboratorio del sistema para realizar las mediciones y obtener experimentalmente los parámetros del motor para la simulación

3.4. Adquisición experimental de parámetros

Los parámetros se han conseguido haciendo uso de equipos de medición de corriente, voltaje, sensores, placa de Arduino mega 2560, códigos de programación,

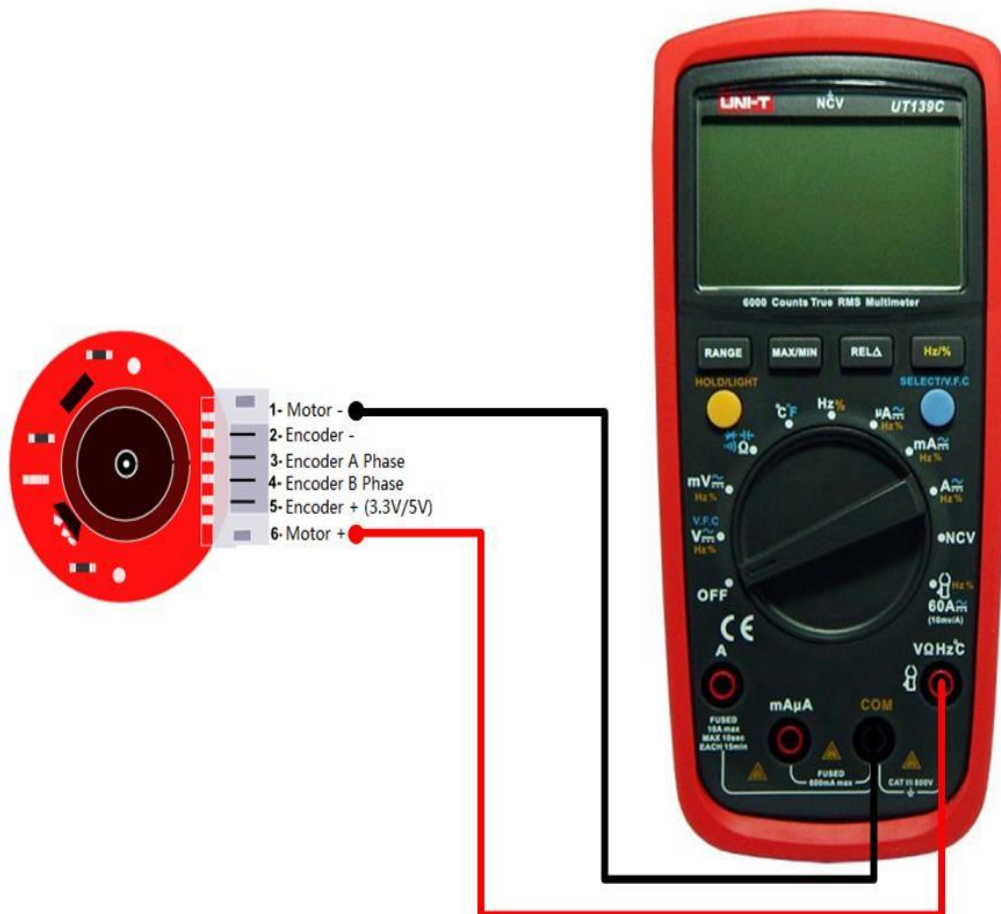
entre otros. La medición se realizó a un motor DC del tipo DC JGA25-370 de 12V/100RPM con encoder.

3.4.1. Resistencia de armadura (R)

La resistencia de armadura se obtuvo midiendo los devanados del motor con un multímetro, donde el negativo del motor debe ir conectado al común del multímetro y el positivo del motor debe ir conectado al puerto que tiene el símbolo de ohmios (Ω). Una vez realizada está conexión se enciende el instrumento y con el conmutador seleccionamos el valor de la medición de la resistencia. La Figura 14 muestra el diagrama de conexiones para realizar dicha medición.

Figura 14.

Diagrama de conexiones para medir la resistencia de armadura



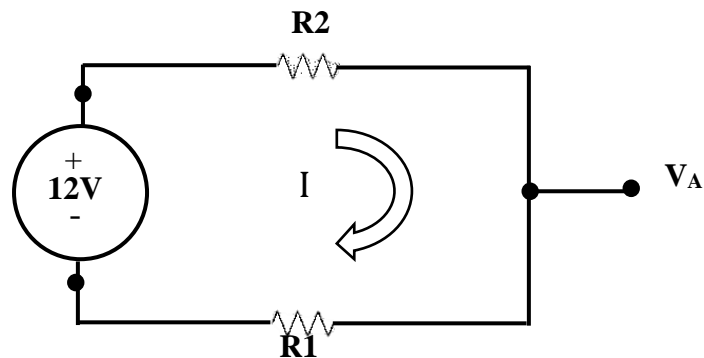
Entonces la resistencia de la armadura medida es de:

$$R = 4,2\Omega$$

3.4.2. Voltaje (V)

Para medir el voltaje que ingresa al motor haciendo uso del Arduino Mega 2560 se emplea un circuito de divisor de voltaje el cual nos ayudó a calcular las resistencias necesarias para reducir el voltaje de salida que va conectado al Arduino. En la Figura 15 se puede apreciar el circuito de divisor de voltaje.

Figura 15.
Circuito de divisor de voltaje



Analizando el circuito tenemos que:

$$\frac{R_2}{R_1} = 1.4$$

De donde:

$$R_2 = 560 \text{ ohm y } R_1 = 380 \text{ ohm}$$

Una vez calculada las resistencias procedemos con las conexiones y la programación para el cálculo del voltaje. En la Figura 16 se puede apreciar el diagrama de conexiones físicas para la medición de voltaje y en la Figura 17 se visualiza los resultados. La programación utilizada se detalla en anexos junto con su circuito eléctrico/electrónico.

Figura 16.

Diagrama de conexión del circuito divisor de voltaje

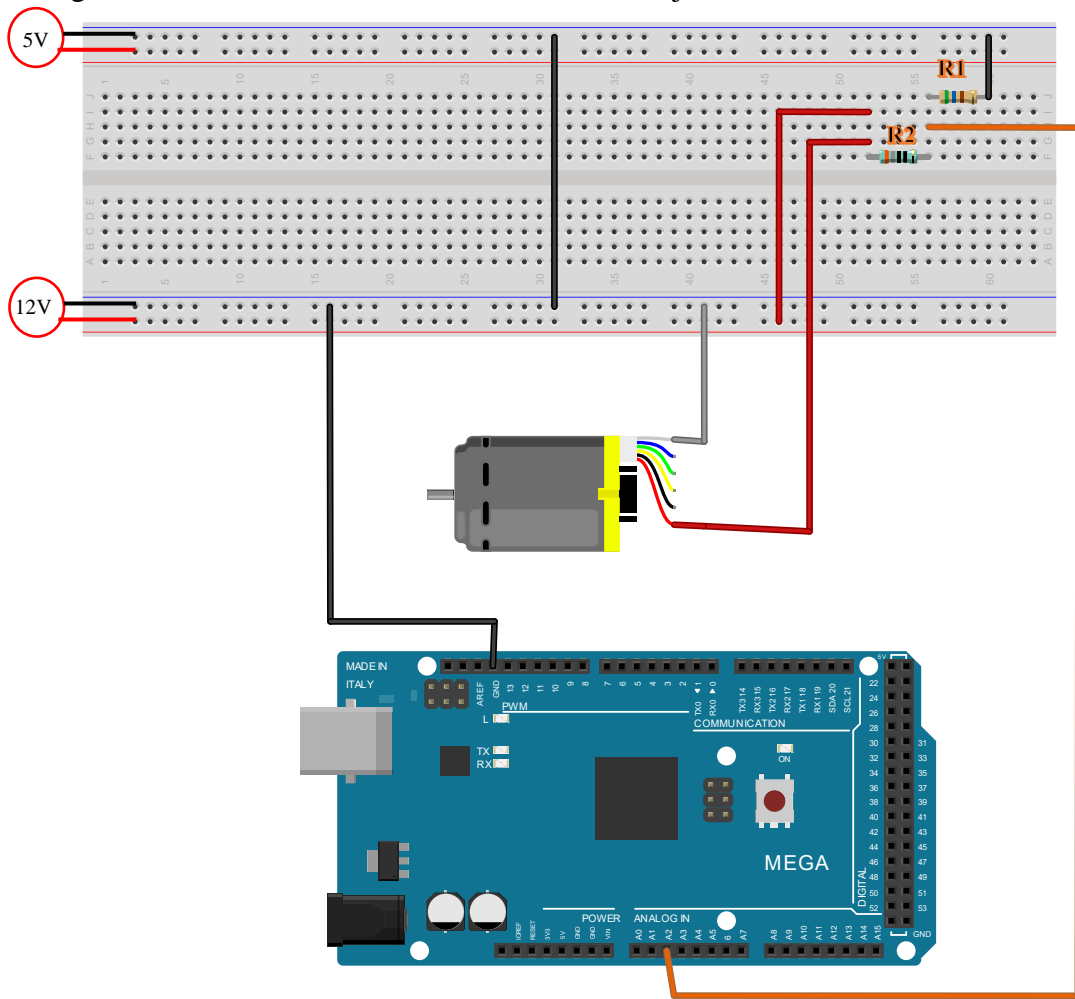


Figura 17.

Resultados de la medición de voltaje

```
Salida Monitor Serie x
Mensaje (Intro para mandar el mensaje d... Nueva línea 9600 baud
22:21:22.705 -> Midiendo voltage...
22:21:22.706 -> DV: 12.36 V,
22:21:23.736 -> DV: 12.34 V,
22:21:24.744 -> DV: 12.32 V,
22:21:25.772 -> DV: 12.32 V,
22:21:26.789 -> DV: 12.33 V,
22:21:27.817 -> DV: 12.33 V,
22:21:28.837 -> DV: 12.32 V,
22:21:29.853 -> DV: 12.32 V,
22:21:30.884 -> DV: 12.33 V,
22:21:31.909 -> DV: 12.32 V,
22:21:32.962 -> DV: 12.32 V,
22:21:33.943 -> DV: 12.32 V,
22:21:34.959 -> DV: 12.32 V,
22:21:35.979 -> DV: 12.32 V,
22:21:37.002 -> DV: 12.32 V,
22:21:38.062 -> DV: 12.33 V,
22:21:39.084 -> DV: 12.33 V,
22:21:40.063 -> DV: 12.33 V,
22:21:41.081 -> DV: 12.32 V,
```

3.4.3. Inductancia de armadura (L)

El valor de la inductancia de armadura se obtiene midiendo los devanados del motor con el multímetro LCR. En la Figura 18 se puede apreciar el diagrama de conexiones.

Figura 18.

Diagrama de conexión para medir la inductancia



El valor de la inductancia medido es de:

$$L = 3.427mH$$

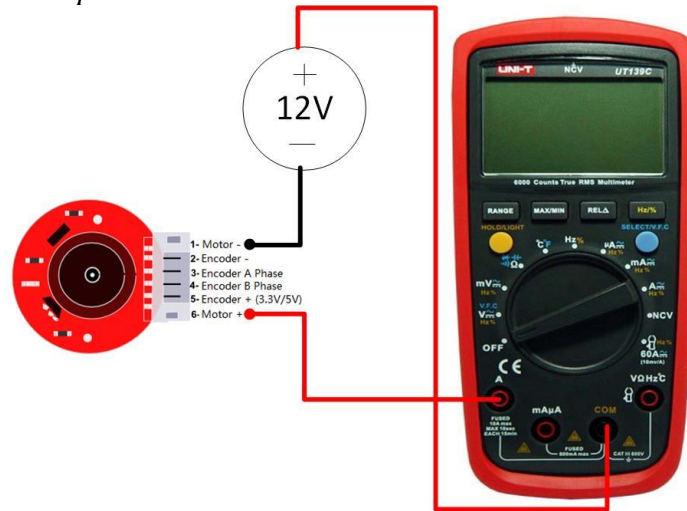
3.4.4. Corriente sin carga (I)

La medición del consumo de corriente para una alimentación de 12V, se obtuvo mediante dos métodos:

Método 1: Este método consistió en medir la corriente haciendo uso del multímetro. donde el negativo del motor debe ir conectado al negativo de la fuente, el común del multímetro debe ir conectado al positivo de la fuente y el puerto que tiene la unidad de la corriente (A) debe ir conectado al positivo del motor. Una vez realizada está conexión se enciende el instrumento y con el conmutador seleccionamos el valor de la medición de la corriente DC. La Figura 19 muestra el diagrama de conexiones.

Figura 19.

Diagrama de conexiones para la medición de la corriente



Método 2: Este método se basó en la programación haciendo uso del sensor de corriente ACS712 y la placa Arduino mega 2560. En la Figura 20 se aprecia las conexiones realizadas para la obtención de la corriente y la Figura 21 detalla los resultados obtenidos, mientras que en la Tabla 9 se detalla la conexión de los pines. El código de programación junto con su circuito eléctrico/electrónico se puede apreciar en los anexos.

Figura 20.

Conexiones del circuito para medir corriente

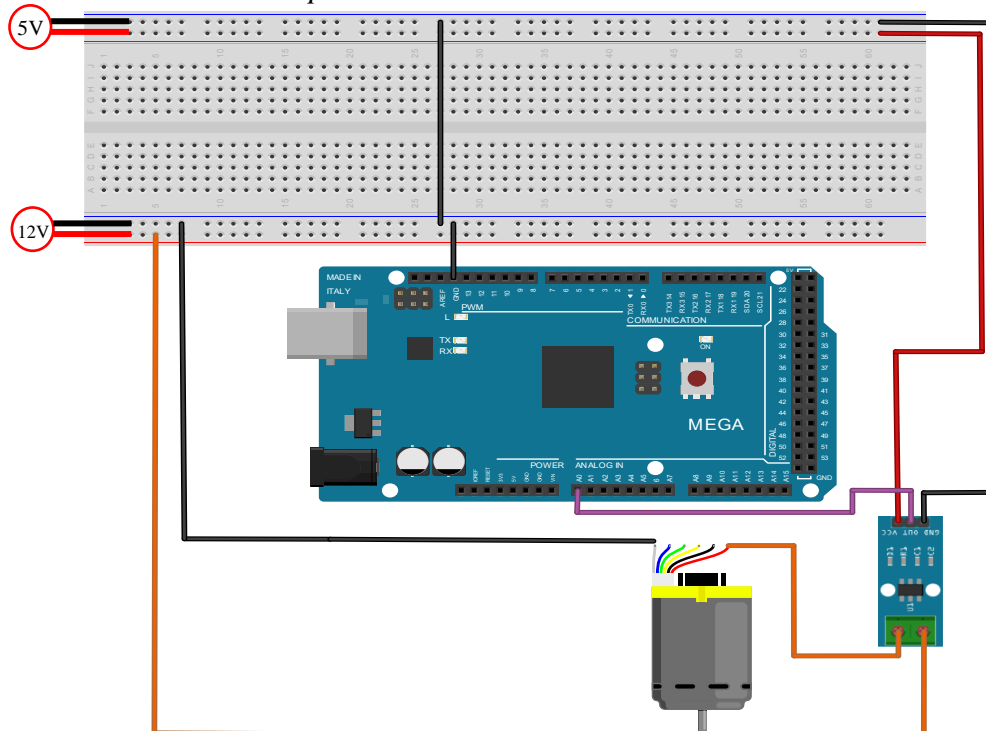


Tabla 9.
Conexiones de pines.

ACS712	Arduino mega 2560	Motor DC con encoder	Fuente de alimentación 12V
VCC	5V	-	-
OUT	A0	-	-
GND	GND	-	-
B1	-	-	12 V
B2	-	Motor +	-
-	-	Motor -	GND

Figura 21.
Resultados de la programación

```

COM7
Enviar

7:23:28.468 -> Corriente: 0.111
7:23:29.464 -> Corriente: 0.111
7:23:30.512 -> Corriente: 0.111
7:23:31.568 -> Corriente: 0.111
7:23:32.595 -> Corriente: 0.111
7:23:33.622 -> Corriente: 0.111
7:23:34.651 -> Corriente: 0.111
7:23:35.680 -> Corriente: 0.111
7:23:36.723 -> Corriente: 0.111
7:23:37.753 -> Corriente: 0.111
7:23:38.783 -> Corriente: 0.111

Autoscroll [x] Mostrar marca temporal [x] Sin ajuste de línea [v] 9600 baudio [v] Limpiar salida [b]

```

Los resultados de los dos métodos se asemejan, por lo que vamos a tomar la corriente como:

$$I = 0.11A$$

3.4.5. Velocidad (w)

Para el cálculo de la velocidad hacemos uso de la programación, utilizando el encóder acoplado al motor. La Figura 22 muestra el diagrama de conexiones realizadas para poder calcular la velocidad en RPM y la Figura 23 detalla los resultados obtenidos. El código de programación y el circuito eléctrico/electrónico se detalla en los anexos.

Figura 22.
Diagrama de conexiones para la medición de RPM

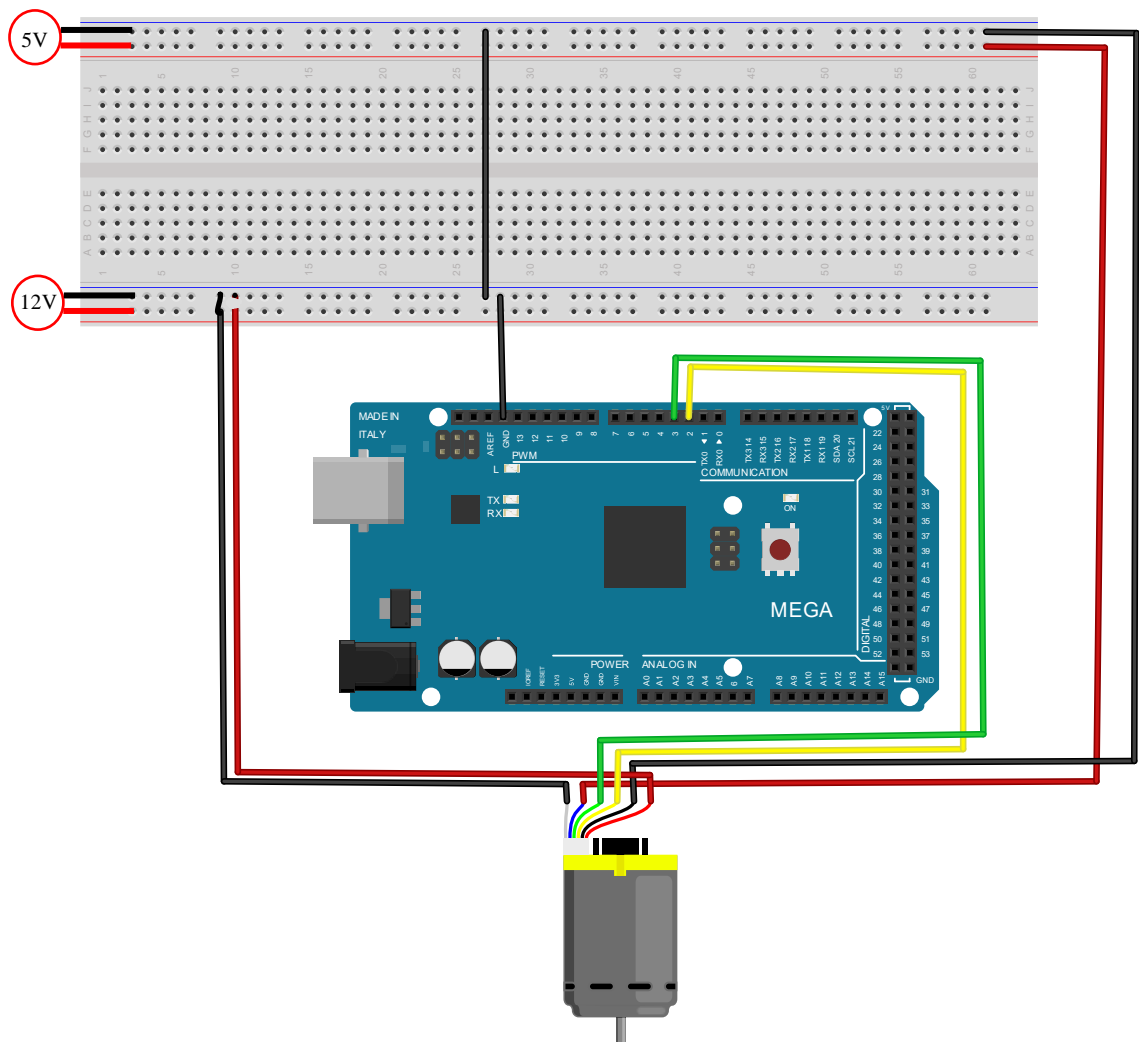
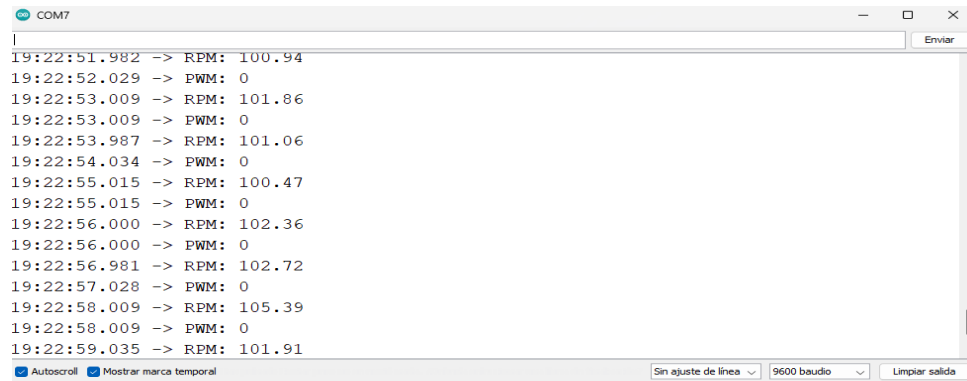


Figura 23.
Resultados de la velocidad



Entonces la velocidad promedio del motor DC sería de:

$$W = 101RPM = 10.577rad / s$$

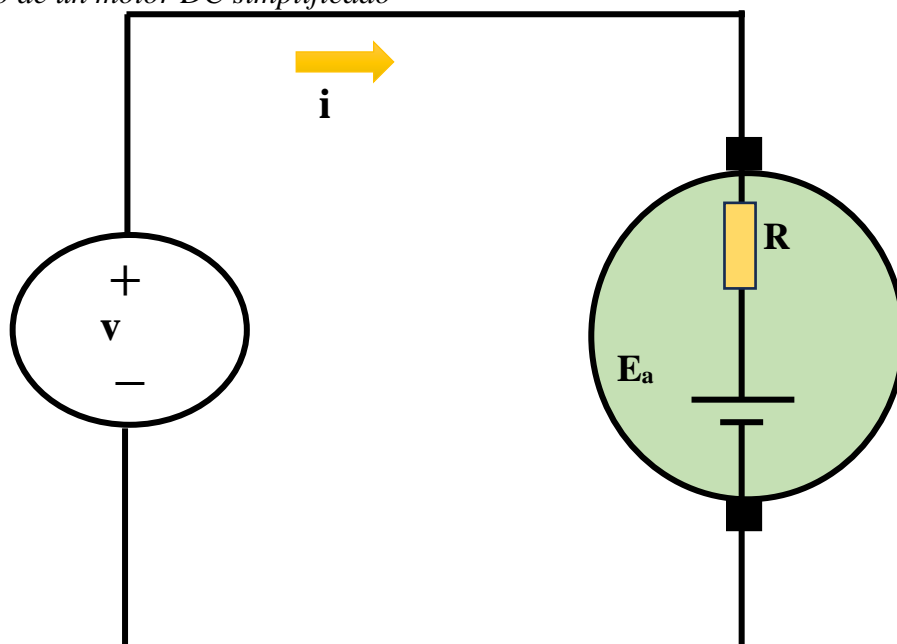
3.4.6. Constante electromotriz y de torque (C_c, C_{tq})

De la ecuación (5) se despeja la constante contraelectromotriz quedando así:

$$C_c = \frac{E_a}{w(t)} \quad (26)$$

Aplicando la ley de Kirchhoff al circuito presente en la Figura 24 se obtiene la ecuación de la fuerza contraelectromotriz.

Figura 24.
Circuito de un motor DC simplificado



Entonces la fuerza electromotriz es:

$$E_a = v - iR \quad (27)$$

Sustituyendo E_a en la ecuación (27), tenemos que:

$$C_c = \frac{v - (iR)}{w(t)} \quad (28)$$

Donde:

C_c : constante contraelectromotriz.

v : voltaje aplicado.

i : corriente de armadura que consume el motor.

R : resistencia de la armadura (Ω).

$w(t)$: velocidad (rad/seg).

Reemplazando valores en la ecuación (29):

$$C_c = \frac{(12V) - (0.11A)(4.2\Omega)}{10.577 \text{ rad} / s}$$

Obtenemos una constante electromotriz de:

$$C_c = 1.091 \text{ Vs} / \text{rad}$$

En unidades del sistema internacional, la constante de torque es igual a la constante contraelectromotriz de retorno, solo cambia las unidades, es decir:

$$C_{tq} = C_c = 1.091 \text{ Nm} / \text{A}$$

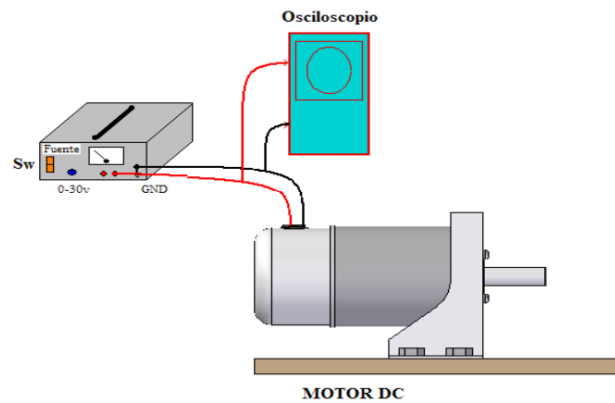
3.4.7. Constante de tiempo mecánico (t_m)

Ortega y Capacho (2009), nos dicen que el modelado matemático de un motor DC es la relación directa entre la velocidad en el eje y la tensión de armadura, que experimentalmente se observa, al aplicarse una variación de voltaje en los terminales hace generar en principio una característica transitoria de velocidad, seguida de una respuesta de régimen estable para cargas fijas, A partir de esta respuesta podemos

encontrar la constante de tiempo que es el 63.2 % del crecimiento de la dinámica hasta alcanzar el estado estable.

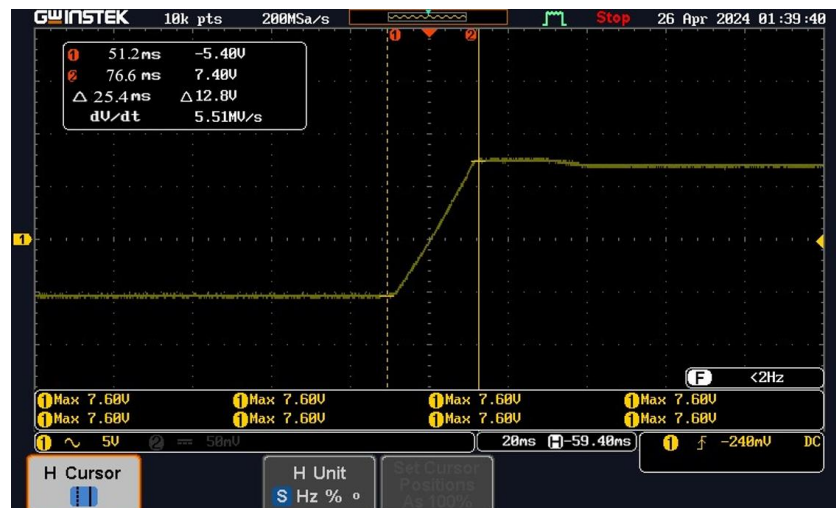
Para medir la constante de tiempo mecánica se hace uso del osciloscopio, para ello se conecta la sonda al borne rojo y el cable de tierra al borne negro como se puede apreciar en la Figura 25.

Figura 25.
Conexiones de osciloscopio



En la Figura 26 se ubica el valor del voltaje del 63.2% equivalente a (7.584 V)

Figura 26.
Medición de la constante del tiempo mecánico



De la figura 26 la constante de tiempo mecánico para 7.584 V es de:

$$t_m = 21ms$$

3.4.8. Momento de inercia (J)

Para el cálculo del momento de inercia con los parámetros obtenidos hasta ahora hacemos uso de la siguiente ecuación:

$$J = \frac{t_m * C_c * C_{tq}}{R} \quad (29)$$

Donde:

J : Momento de inercia.

t_m : Tiempo mecánico.

C_c : Constante electromotriz.

C_{tq} : Constante del par de motor.

R : resistencia.

Reemplazando datos en la ecuación (29), tenemos:

$$J = \frac{0.021s * 1.091Vs / rad * 1.091Nm / A}{4.2\Omega}$$

Así el momento de inercia tendría el siguiente valor:

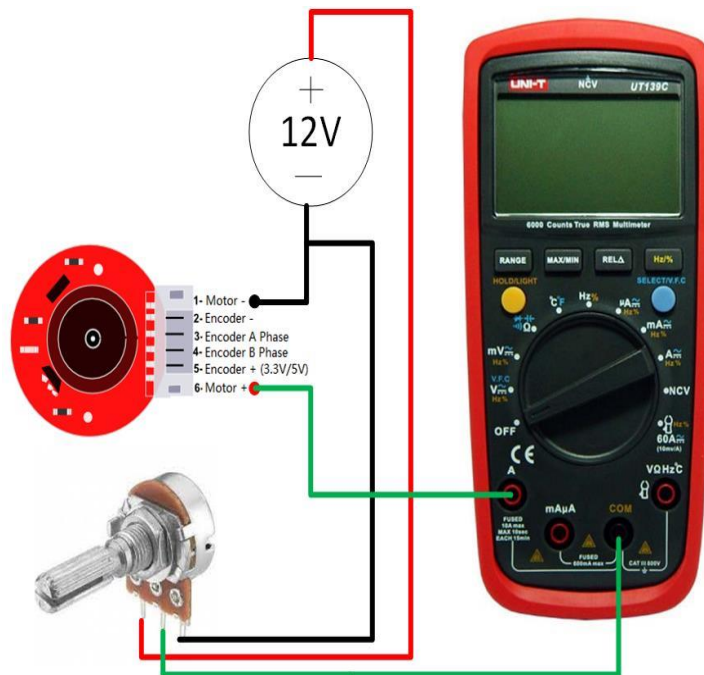
$$J = 0.006kg * m^2$$

3.4.9. Corriente de arranque (i_{arr})

Para la corriente de arranque se incrementó la tensión que se le aplica al motor haciendo uso de un potenciómetro hasta llegar a observar el giro del eje del motor, allí con la ayuda de un amperímetro se tomó lectura para así saber cuál es la corriente necesaria para que el motor arranque. El diagrama de conexiones se puede apreciar en la Figura 27.

Figura 27.

Diagrama de conexiones para la corriente de arranque



La corriente que hace girar al motor es de:

$$i_{arr} = 0.08A$$

3.4.10. Torque de fricción (T_f)

Para calcular el torque de fricción hacemos uso de la siguiente fórmula:

$$T_f = C_{iq} * i_{arr} \quad (30)$$

Donde:

T_f : Torque de fricción.

C_{iq} : Constante del par del motor.

i_{arr} : Corriente de arranque.

Reemplazando valores en la ecuación (30):

$$T_f = 1.096Nm / A * 0.08A$$

Tenemos que

$$T_f = 0.088Nm$$

3.4.11. Torque del motor (T_m)

Para calcular el torque del motor hacemos uso de la ecuación (4) descrita anteriormente:

$$T_m = C_{iq} i \quad (31)$$

Reemplazando datos en la ecuación (31):

$$T_m = 1.091 Nm / A * 0.110 A$$

Tenemos que el torque es de:

$$T_m = 0.120 Nm$$

3.4.12. Constante de fricción de Coulomb (b)

De acuerdo con López (2020), en su investigación establece que la constante de fricción de Coulomb se determina cuando el motor alcanza una velocidad constante. Este hallazgo subraya la importancia de comprender el comportamiento del motor en condiciones estables de funcionamiento.

Esta constante se encuentra cuando el motor está en estado estable, por lo que de la ecuación (2), sabiendo que el motor está en estado estacionario se toma las derivadas en cero, obteniendo así la fórmula de la fricción viscosa:

$$b = \frac{T_m - T_f}{w(t)} \quad (32)$$

b : Constante de fricción de Coulomb

T_m : Torque

T_f : Torque de fricción

$w(t)$: Velocidad del motor

Reemplazando valores en la ecuación (32):

$$b = \frac{0.120 Nm - 0.088 Nm}{10.577 rad / s}$$

Obtenemos un valor de:

$$b = 0.003 Nms$$

3.4.13. Valores obtenidos de las mediciones.

Los valores de los parámetros obtenidos del experimento se muestran en la Tabla 10.

Tabla 10.
Parámetros del motor DC 12V-100RPM

Parámetro	Símbolo	Valor	Unidad
Resistencia	R	4.200	Ω
Inductancia	L	0.003427	H
Corriente sin carga	I	0.110	A
Velocidad del motor	W	10.577	rad / s
Constante electromotriz	C_c	1.091	Vs / rad
Constante del Par del Motor	C_{τ}	1.091	Nm / A
Contante de tiempo mecánica	t_m	0.021	s
Momento de inercia	J	0.006	$kg * m^2$
Corriente de arranque	i_{arr}	0.080	A
Torque	T_m	0.120	Nm
Constante de Fricción viscosa	b	0.003	Nms

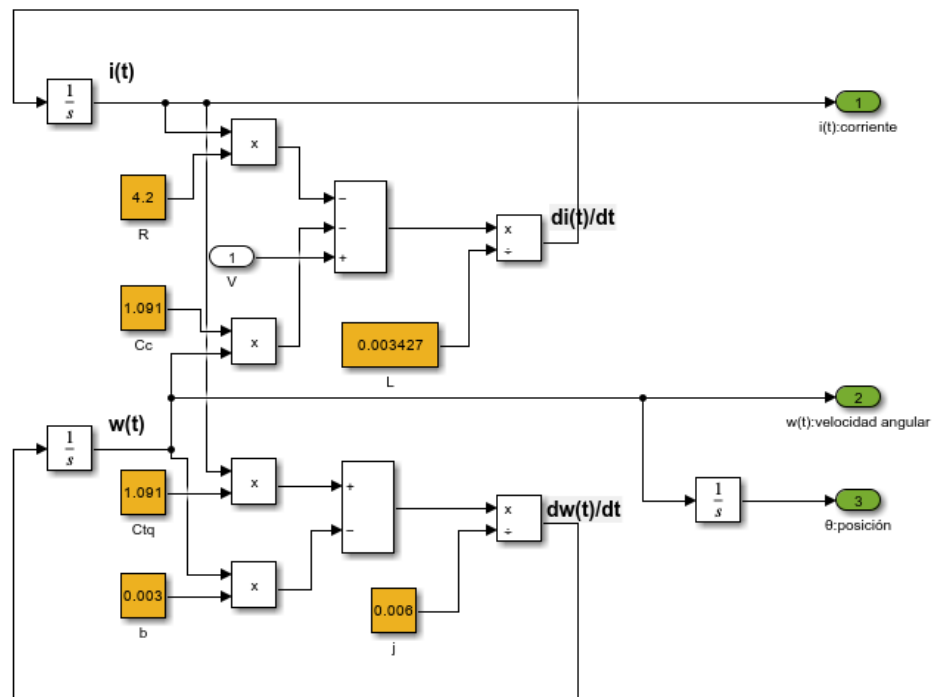
3.5. Implementación en Matlab/Simulink

Según Alvarez (2012), Matlab/Simulink proporciona una interfaz de usuario gráfica (GUI) para resolver modelos matemáticos mediante diagramas de bloques.

Por lo que para la implementación de las ecuaciones diferenciales en el software Simulink se hace uso de la lógica de bloques, como se puede apreciar en la Figura 28. Este diagrama de bloques abarca las ecuaciones diferenciales presentadas en el apartado “Modelado matemático con ecuaciones diferenciales”. Cada una de las fórmulas de este apartado se ve reflejada en el diagrama con sus tres variables de salidas que son: la corriente (i), la velocidad angular (w) y la posición (θ). En la Tabla 11 se describe los bloques utilizados en la Figura 28.

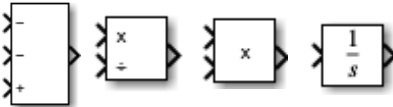



Figura 28.

Diagrama de bloques de las Ecuaciones Diferenciales



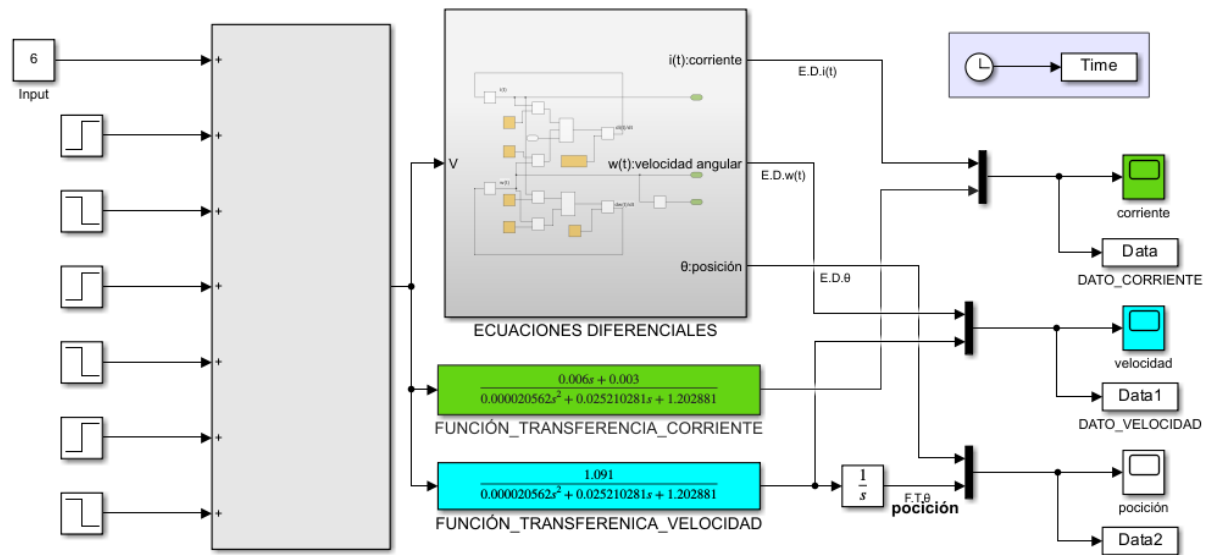
Nota: Esta imagen representa la solución de las Ecuaciones Diferenciales en Matlab/Simulink.

Tabla 11.
Leyenda de diagrama de bloques

Nombre	Descripción	Bloque
Operadores matemáticos	Se utilizaron para sumar, restar, multiplicar, dividir e integrar.	
Constantes	Se utilizaron para agregar los valores de los parámetros obtenidos del experimento.	
Entrada	Sirve para ingresar el voltaje al modelo.	
Salidas	Permiten obtener las salidas del modelo matemático.	

La Figura 29 muestra el modelo matemático implementado en Matlab/Simulink, comparando las ecuaciones diferenciales con las funciones de transferencias respecto a la corriente y a la velocidad.

Figura 29.
Implementación del modelo matemático en Matlab/Simulink

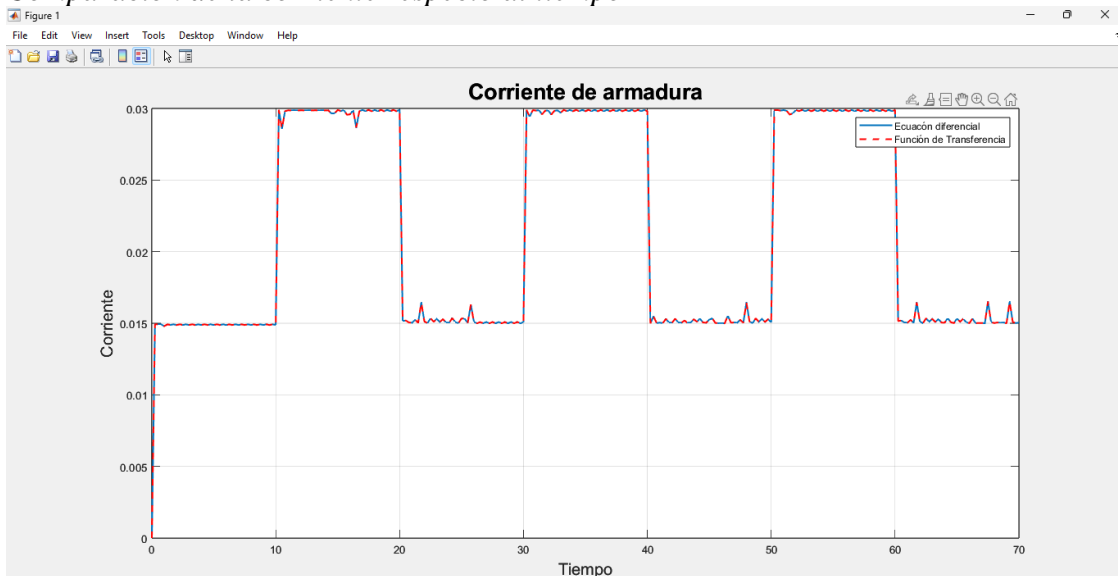


Nota: la imagen representa las E.D. y F.T. del modelo en Matlab/Simulink. Elaboración propia.

Se utilizó Matlab para verificar, exclusivamente para propósitos académicos descriptivos para los que precisamente se ha abocado el presente trabajo, que el modelo

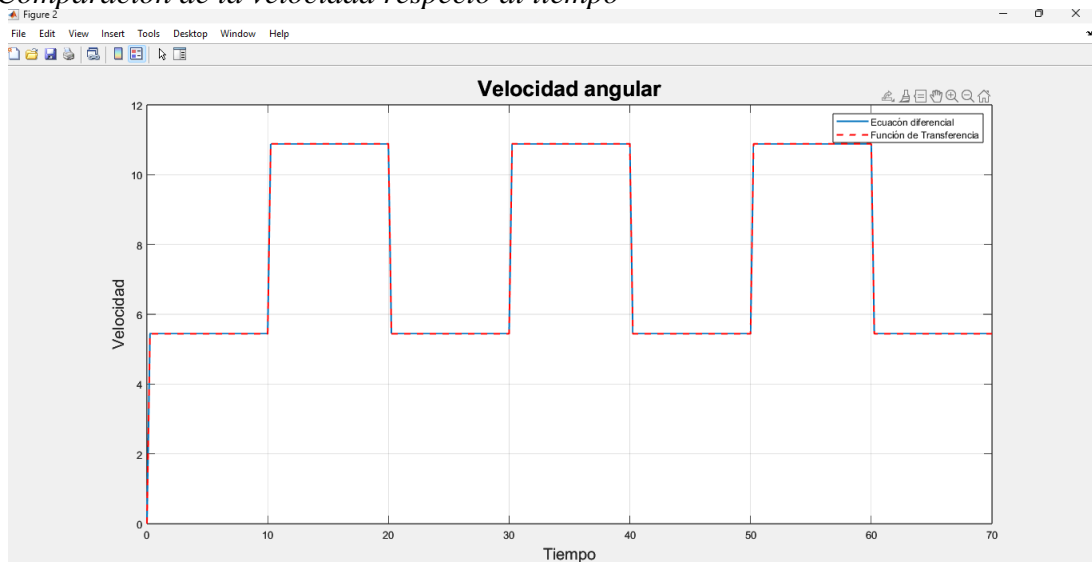
de ecuaciones diferenciales y el modelo de funciones de transferencias arrojan resultados iguales (ver figuras 30,31). Esto se debe a que el modelo fenomenológico obtenido del motor DC es Lineal.

Figura 30.
Comparación de la corriente respecto al tiempo



Nota: Esta imagen compara la ecuación diferencial con la función de transferencia de la corriente con respecto al tiempo.

Figura 31.
Comparación de la velocidad respecto al tiempo



Nota: Esta imagen compara la ecuación diferencial con la función de transferencia de la velocidad angular con respecto al tiempo.

Se ha considerado el modelo con funciones de transferencia porque está representado por ecuaciones algebraicas simples que son fáciles de entender por estudiantes o profesionales.

OE3: Validación del modelo matemático con los parámetros obtenidos del experimento

3.6. Validación del modelado con un motor real

La validación consiste en comparar los resultados reales con los obtenidos del modelo matemático, asegurando así su precisión y fiabilidad (Pulido, 2021).

La validación de la corriente y de la velocidad se realizó gráficamente y mediante el coeficiente de correlación de Pearson que cuantifica la asociación lineal entre dos variables numéricas. Los datos generados por el modelo y por el motor real se aprecian en anexos.

La programación en Arduino y el diagrama de conexiones para la obtención de los parámetros del motor DC se aprecia en anexos.

3.6.1. Correlación de Pearson.

Según Hernández et al. (2014), en su investigación detalla los valores de correlación de Pearson (r) y su valor oscilan entre -1 a +1. El valor -1 indica correlación negativa perfecta y el valor de +1 indica correlación positiva perfecta. Mientras más se acerque al +1 la relación de variables es más intensa, la ecuación que se utiliza para encontrar el valor de “ r ” es la siguiente. La Tabla 12 ayuda analizar e interpretar si existe una correlación entre las dos variables.

$$r = \frac{n \sum XY - \sum X \sum Y}{\sqrt{n \sum X^2 - (\sum X)^2} \sqrt{n \sum Y^2 - (\sum Y)^2}} \quad (33)$$

Tabla 12.*Coefficiente de correlación de la escala “r” de Pearson*

Niveles de correlación de Pearson	
-1,00	Correlación negativa perfecta: -1
-0,90	Correlación negativa muy fuerte: -0,90 a -0,99
-0,75	Correlación negativa fuerte: -0,75 a -0,89
-0,50	Correlación negativa media: -0,50 a -0,74
-0,25	Correlación negativa débil: -0,25 a -0,49
-0,10	Correlación negativa muy débil: -0,10 a -0,24
0,00	No existe correlación alguna: -0,09 a +0,09
+0,10	Correlación positiva muy débil: +0,10 a +0,24
+0,25	Correlación positiva débil: +0,25 a +0,49
+0,50	Correlación positiva media: +0,50 a +0,74
+0,75	Correlación positiva fuerte: +0,75 a +0,89
+0,90	Correlación positiva muy fuerte: +0,90 a +0,99
+1,00	Correlación positiva perfecta: +1

Nota: Esta tabla ha sido adaptada de “Metodología de la investigación”, por Hernández et al., 2014, p.305.

3.6.2. Validación de la corriente.

En la validación de la corriente se siguió los siguientes pasos:

PASO 1: Obtención de la corriente real.

En este paso a través de la programación en Arduino se obtuvo la corriente real que ingresa al motor haciendo variaciones en la tensión, dichas variaciones oscilan entre los 6 voltios y los 12 voltios. Los resultados obtenidos se aprecian en la Tabla 13 de anexos columna 3.

PASO 2: Obtención de la corriente modelo.

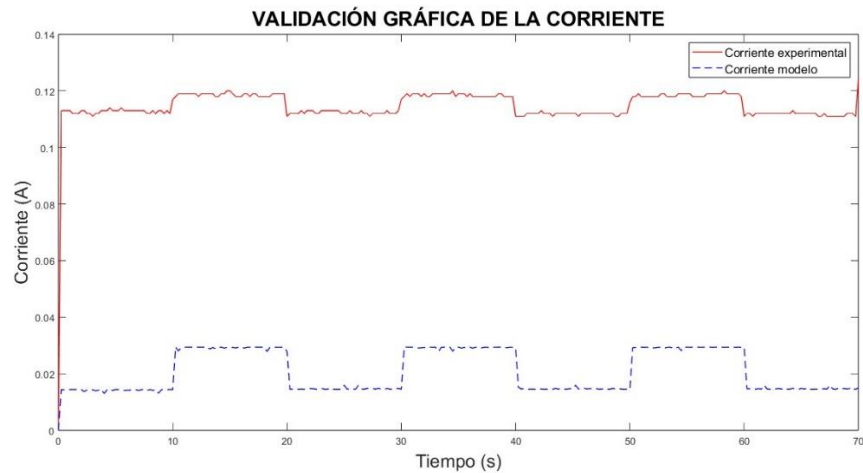
En este paso a través de la programación en Matlab se obtuvo la corriente modelo, aplicando las mismas tensiones que en el paso anterior. Los resultados se detallan en la Tabla 13 de anexos columna 4.

PASO 3: Validación gráfica.

En la Figura 32 se muestra la validación gráfica para la corriente experimental y modelada. Se aprecia que existe un “offset”, o variación del nivel de referencia, usualmente estas situaciones son normales ya que en las mediciones se incorporan inevitablemente voltajes DC que se consideran interferencias, y ello se corrige simplemente removiendo dicho nivel DC. En anexos se muestra la programación en Matlab para la obtención de las gráficas.

Figura 32.

Validación gráfica de la corriente



PASO 4: Validación aplicando la correlación de Pearson.

En este paso se aplica la correlación de Pearson a la corriente modelo y a la real, haciendo uso de la programación en Python. La Figura 33 detalla la programación realizada en Python y su resultado.

Figura 33.

Validación de la corriente mediante la correlación de Pearson

*Validación de la corriente *

```
95 # Importación de librerías
import pandas as pd
from google.colab import files

# Subir archivo Excel
uploaded = files.upload()

# Obtener el nombre del archivo subido
nombre_archivo = next(iter(uploaded))

# Leer el archivo Excel como DataFrame de pandas
datos_excel = pd.read_excel(nombre_archivo)

# Calcular la correlación entre las columnas "Current [A] (EXPERIMENT)" y "Current [A] (SIMULATION)"
correlacion = datos_excel['Current [A] (EXPERIMENT)'].corr(datos_excel['Current [A] (SIMULATION)'])

# Mostrar la correlación calculada
print(f"correlación Pearson: {correlacion}")
```

Elegir archivos PARAMETR...R_DC (1).xlsx

- PARAMETROS_FINALES_DEL_MOTOR_DC (1).xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet)

Saving PARAMETROS_FINALES_DEL_MOTOR_DC (1).xlsx to PARAMETROS_FINALES_DEL_MOTOR_DC (1) (13).xlsx
correlación Pearson: 0.5424021347108753

Nota: Este código ha sido adaptado de “Correlación lineal con Python”, por Amat (2020).

3.6.3. Validación de la velocidad.

En la validación de la velocidad se siguió los siguientes pasos:

PASO 1: Obtención de la velocidad real.

En este paso a través del sistema de medición desarrollado con la plataforma Arduino se obtuvo la velocidad real del motor para las mismas tensiones descritas en el apartado 3.6.2. Los resultados obtenidos se aprecian en la Tabla 13 de anexos columna 5.

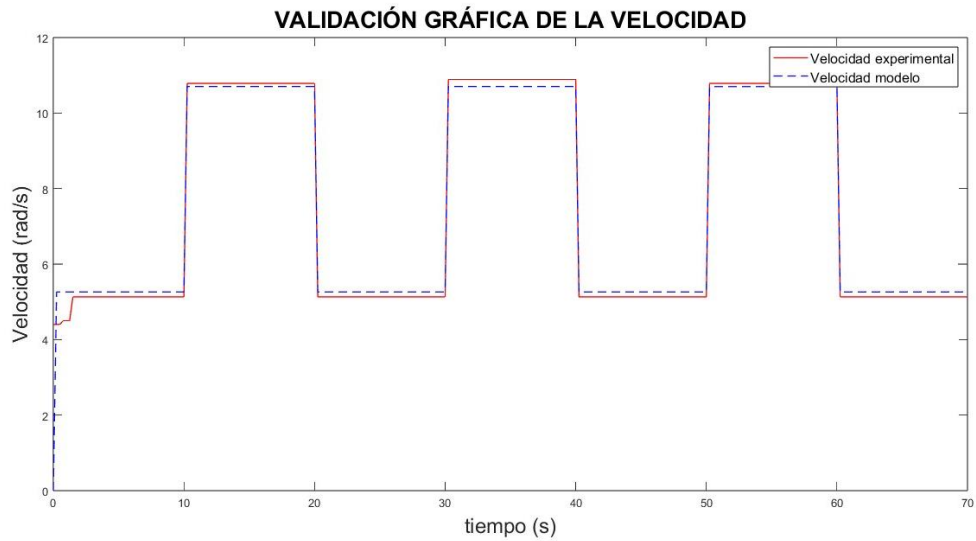
PASO 2: Obtención de la velocidad modelo.

En este paso a través del modelo matemático implementado con la programación en Matlab se obtuvo la velocidad modelo, aplicando las mismas tensiones que el apartado 3.6.2. dichos resultados se aprecian en la Tabla 13 de anexos columna 6.

PASO 3: Validación gráfica.

En la Figura 34 se muestra la validación gráfica para la corriente experimental y modelada. En anexos se muestra la programación en Matlab para la obtención de las gráficas.

Figura 34.
Validación gráfica de la velocidad



PASO 4: Validación usando la correlación de Pearson.

En este paso se aplica la correlación de Pearson a la velocidad modelo y a la real, haciendo uso de la programación en Python. La Figura 35 detalla la programación realizada en Python y su resultado.

Figura 35
Validación de la velocidad mediante la correlación de Pearson

```
Validación de la velocidad

# Importación de librerías
import pandas as pd
from google.colab import files

# Subir archivo Excel
uploaded = files.upload()

# Obtener el nombre del archivo subido
nombre_archivo = next(iter(uploaded))

# Leer el archivo Excel como DataFrame de pandas
datos_excel = pd.read_excel(nombre_archivo)

# Calcular la correlación entre las columnas "speed [rad/s] (EXPERIMENT)" y "speed [rad/s] (SIMULATION)"
correlacion = datos_excel['speed [rad/s] (EXPERIMENT)'].corr(datos_excel['speed [rad/s] (SIMULATION)'])

# Mostrar la correlación calculada
print(f"correlación Pearson' : {correlacion}")
```

Elegir archivos PARAMETR...R_DC (1).xlsx

- PARAMETROS_FINALES_DEL_MOTOR_DC (1).xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet)

Saving PARAMETROS_FINALES_DEL_MOTOR_DC (1).xlsx to PARAMETROS_FINALES_DEL_MOTOR_DC (1) (18).xlsx

correlación Pearson' : 0.9944448581256569

Nota: Este código ha sido adaptado de “Correlación lineal con Python”, por Amat (2020).

IV. DISCUSIÓN

En la validación del modelado matemático con los datos reales, la correlación de Pearson para la corriente fue de $r = 0.542$, aunque según la Tabla 12 quiere decir que existiría una correlación positiva media entre el modelo matemático y los datos reales del laboratorio para la variable de la corriente, lo cual se corrobora claramente en las mediciones donde la corriente es altamente variable. Estas variaciones se deben a que los cables de conexiones actúan como receptores de ruido e interferencias que al ingresar al sensor de corriente alteran la medición, esto se podría superar en trabajos futuros tanto con la incorporación de un sistema de filtrado antes del ingreso al sensor de corriente, como también reduciría el efecto del ruido e interferencias, realizar las conexiones a través de un circuito impreso PCB.

En la Figura 34 se aprecia que la similitud del comportamiento para la velocidad modelada y experimental son más similares respecto al tiempo que en la de la corriente. En la validación de la velocidad se obtuvo una correlación de $r = 0.994$, lo que según la Tabla 12 la correlación entre estas dos variables es positiva muy fuerte. En este caso, se muestra un índice de correlación alta por que la velocidad se mide de modo que no está sujeto a interferencias y ruido, ya que se realiza con un sistema de encoder óptico que ofrece mediciones más confiables.

En la Figura 32 se puede apreciar que la similitud sólo del comportamiento de la corriente modelada y experimental son similares respecto al tiempo. Sin embargo, se detalla una pequeña variación de nivel o “offset”. Este “offset”, o variación del nivel de referencia, usualmente estas situaciones son normales ya que en las mediciones se incorporan inevitablemente voltajes DC que se consideran interferencias, y ello se corrige simplemente removiendo dicho nivel DC. Este nivel DC que se inserta en las

mediciones se debe a que existe un nivel de ruido que se inserta a su vez en el sistema de medición desarrollado a través de los cables que funcionan como antenas de radiación electromagnética, proveniente principalmente de las bobinas del motor que se ubica cercanamente a los niveles de corriente usada por el sistema. Esta es también la razón de la alta variabilidad de las mediciones de corriente que como ya se ha reportado anteriormente tiene un índice de correlación medio.

El modelo del circuito eléctrico asumido, es un modelo básico que en futuras investigaciones se debe de mejorar, cabe recordar que este es uno de los primeros trabajos de aplicación de modelos matemáticos para propósitos de fortalecimiento de conocimientos en los talleres y/o ejercicios que se realicen en laboratorios de cursos vinculantes en la carrera de Ingeniería Mecánica y Eléctrica.

La utilidad de programas de simulación y programación son importantes en el ámbito académico, por la ayuda que estos brindan para comprender teoría y funcionamiento de los equipos.

El voltaje que ingresa al sistema experimental (maqueta) es un voltaje PWM, pero para fines académicos se realizó un nuevo experimento haciendo que el voltaje de entrada al sistema sea fijo. Al comparar estos voltajes (ver anexo figura comparación de voltajes) se aprecia que no existe mucha variación. Si la variable de entrada voltaje fijo no varía a gran escala del voltaje PWM, sucede lo mismo con las variables de salida corriente y velocidad. y es por ello que en esta tesis se ha optado por el voltaje PWM, ya que es más eficiente respecto al otro, esta alta eficiencia energética, es porque el sistema solo consume energía en los intervalos de encendido

V. CONCLUSIONES Y RECOMENDACIONES

1. A través de nuestra propia experiencia, este tipo de trabajos contribuyen a fortalecer las capacidades de desarrollo de hardware y software, además de consolidar conocimientos teóricos, y que en conjunto pueden fortalecer la experiencia de aprendizaje en los laboratorios IME-UNJ y el Laboratorio de Medición de Recursos Energéticos Renovables del Instituto de Investigación en Energías Renovables. Cabe resaltar que todo sistema o mecanismo físico real (mecánico y/o eléctrico) puede desarrollarse su respectivo modelo matemático, por lo tanto, con el presente trabajo se busca impulsar que ese ejercicio de modelamiento y simulación matemático se estandarice en cada curso de la carrera de Ingeniería Mecánica y Eléctrica, así como también sea incorporado en sus proyectos de investigación.
2. El valor del coeficiente de correlación obtenido para la corriente es medio y para la velocidad es alto, no obstante, ambos se consideran aceptables para los propósitos del presente proyecto. Cabe resaltar que en el caso de la velocidad (RPM) el sensor es un encoder basado en mecanismo óptico que ofrece mediciones más seguras y confiables, es por ello que al comparar con el modelo se obtiene un índice de correlación alto. Sin embargo para el caso de la corriente se aprecia que las mediciones son muy ruidosas y variables, y la interpretación que se brinda para ello es que el sistema de conexiones con cables que se ha usado es altamente sensible a las radiaciones electromagnéticas que existen hoy en día en cualquier ambiente, y que a través de los cables actuando como antenas reciben dichas radiaciones aportando a las mediciones ruido e interferencias que ingresan conjuntamente al sensor de corriente causando los efectos reportados, y su correspondiente índice de correlación medio respecto a los valores del modelo. No obstante, es parte del trabajo futuro que se recomienda realizar para continuar mejorando este proceso de definir cada vez mejores modelos matemáticos de sistemas reales.
3. Como en ambos casos la correlación es aceptable, queda demostrado que el modelo implementado ofrece resultados aceptables para el propósito principal de este trabajo que es impulsar a la comunidad IME-UNJ para que incursione en el

uso e interpretación de modelamiento matemático, simulaciones y ensayos experimentales de sistemas físicos como son los motores DC.

4. Se recomienda la adquisición de instrumentos de medición precisos y de buena calidad en los laboratorios de IME para obtener datos exactos y fiables en los experimentos. Por ejemplo, aquellos que son imprescindibles y urgentes podemos citar: Medidor multiparámetros LCR que incluya especialmente medición de Inductancia y Capacitancia con alto nivel de precisión; Tacómetro digital con capacidad de registro de mediciones; Fuentes de voltaje certificadas para ensayos de laboratorio, puntas de medición de osciloscopio que permita medir diversos niveles de voltaje tanto AC como DC. Asimismo, se recomienda la adquisición por parte de UNJ de la licencia del software MATLAB y varios de sus Toolboxes, ya que para este proyecto hemos usado una versión educativa de prueba que dura 30 días, pero hemos verificado la utilidad de dicho software en temas de programación y ayuda disponible en comunidades dentro de la web. De no ser este programa, se recomienda cualquier otro disponible en el mercado que preste similares funciones. Finalmente, con la finalidad de que este trabajo consolide su propósito de impulsar las capacidades de los alumnos de IME en modelización y simulación matemática, cuyas aplicaciones están en todos los campos dentro de la carrera, se recomienda apoyar la debida difusión del presente trabajo.

VI. REFERENCIAS BIBLIOGRÁFICAS

- Alvarez, M. (2012). Modelo matemático de un motor de corriente continua separadamente excitado: Control de velocidad por corriente de armadura. *LAPJE*. <https://www.researchgate.net/publication/266278997>
- Choe, T., Ryu, Z., Kang, S., & Ri, J. (2021). Mathematical modeling of the two-stroke crankcase scavenged spark ignition engine working process. *Energy Conversion and Management*: X, 10. <https://doi.org/10.1016/j.ecmx.2021.100086>
- Feliciano, G. (2019). *La inteligencia emocional y la relación con el clima organizacional en la institución educativa primaria N° 70168-Carumas.Acora-Puno-2018* [Tesis, Universidad Nacional de San Agustín de Arequipa]. <https://repositorio.unsa.edu.pe/server/api/core/bitstreams/2567d135-c3b8-4e93-9aaf-385030ee847c/content>
- Flores, D. (2021). *Un modelo praxeológico para el estudio de la transformada de laplace en ingeniería mecatrónica* [Tesis, Pontificia Universidad Católica del Perú]. https://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/21270/FLORES_GALLO_DIANA_CAROLINA%20%281%29.pdf?sequence=1
- Meneses, M., & Zafra, H. (2013). *Diseño e Implementación de un Módulo Educativo para el control del Sistema Bola y Varilla*. [Tesis, Pontificia Universidad Católica del Perú].

https://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/4895/MENESES_PAVEL_DISE%C3%91O_MODULO_EDUCATIVO_CONTROL_SISTEMA_BOLA_VARILLA.pdf?sequence=1&isAllowed=y

Ortega, M., & Capacho, C. (2009). *Obtención experimental de los parámetros del motor que se utilizará en el sistema de locomoción de una esfera rodante* [Tesis, Universidad Pontificia Bolivariana].

https://repository.upb.edu.co/bitstream/handle/20.500.11912/504/digital_17633.pdf?sequence=1&isAllowed=y

Paulo, I., Carlos, A., & Perez, M. (2017). Modelagem Matemática de Motor de Corrente Contínua e Análise Dinâmica. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 5(1), Article 1. <https://doi.org/10.5540/03.2017.005.01.0178>

Pulido, S. (2021). *Modelado Físico y Simulación usando Matlab-Simscape de Planta de Laboratorio* [Tesis, Universidad de Valladolid]. <https://uvadoc.uva.es/bitstream/handle/10324/47935/TFG-I-1916.pdf?sequence=1>

Quispe, J. (2022). *Obtención gráfica del control de velocidad para motores de corriente continua mediante el modelado y simulación utilizando PSIM* [Tesis, Universidad Nacional del Centro del Perú]. <http://repositorio.uncp.edu.pe/handle/20.500.12894/8543>

Sheng, H., Liu, T., Zhao, Y., Chen, Q., Yin, B., & Huang, R. (2023). New model-based method for aero-engine turbine blade tip clearance measurement.

Chinese Journal of Aeronautics, 36(8), 128-147.

<https://doi.org/10.1016/j.cja.2022.09.012>

Sun, L. (2022). Low speed sensorless control method of brushless DC motor based on pulse high frequency voltage injection. *Alexandria Engineering Journal*, 61(8), 6457-6463. <https://doi.org/10.1016/j.aej.2021.12.005>

Valdivia, R. (2021). *Modelamiento y control de velocidad de motor DC mediante algoritmo PID basado en reglas de Ziegler—Nichols* [Tesis, Universidad Católica de Santa María]. https://alicia.concytec.gob.pe/vufind/Record/UCSM_5b109ad64c4d6576bf991ce85cd27a9e

Wang, Y., Huang, J., Zhou, W., Lu, F., & Xu, W. (2022). Neural network-based model predictive control with fuzzy-SQP optimization for direct thrust control of turbofan engine. *Chinese Journal of Aeronautics*, 35(12), 59-71. <https://doi.org/10.1016/j.cja.2022.04.012>

VII. ANEXOS

Código de programación para la medición de voltaje.

```
// NOTA: Adaptado de: https://forum.arduino.cc/t/voltage-measurement/244718 //
/*=====
// Voltage rent setup measurements
//=====
*/

const float arduinoVCC      = 5.2; //Your Arduino voltage
unsigned long ValueR1       = 320; // Maxv = 5 (1 + ValueR2/ValueR1)
unsigned long ValueR2       = 450; // 1.4 =ValueR2/ValueR1
const int inputResolution   = 1023; //works with most Arduino boards
const int analogPinV       = A2; //the pin connecting the voltage.

/*=====
// Setup measurements and variables
//=====
*/

float voltage;
const float volt_n_aver = 140; // number of readings
volatile float voltage_V = 0.0;

void readVoltage()
{ volatile int A0Value = analogRead(analogPinV);
  volatile float voltage_sensed = A0Value * (arduinoVCC / (float)inputResolution);
  voltage_V = voltage_sensed * (1 + ((float)ValueR2 / (float)ValueR1));
}

void setup()
{ //=====
  // Setup monitoring
  Serial.begin(9600);
  while (!Serial)
  {
    // will pause Zero, Leonardo, etc until serial console opens
    delay(1);
  }
  Serial.println("Midiendo voltage...");
}

void loop()
{
  readVoltage();
  voltage_V = getVoltageAverage(volt_n_aver);
  Serial.print("DV: ");
  Serial.print(voltage_V);
  Serial.println(" V, ");
  delay(1000);

}

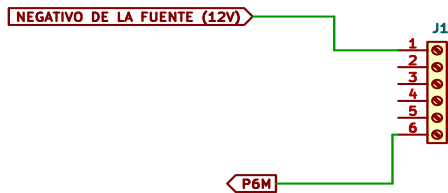
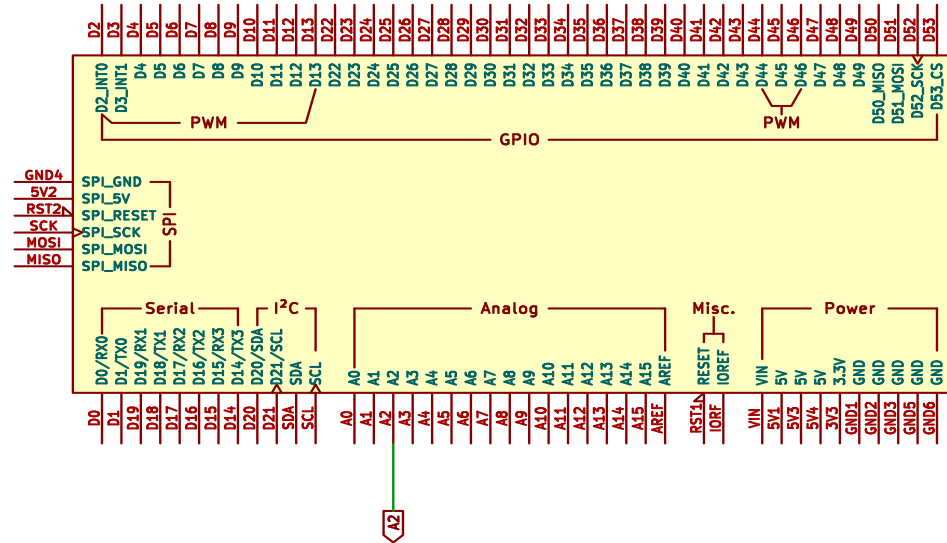
float getVoltageAverage(int volt_n_aver)
{
  volatile float voltage_temp_average = 0;
  for (int i = 0; i < volt_n_aver; i++)
  {
    readVoltage();
    voltage_temp_average += voltage_V;
  }
  return voltage_temp_average / volt_n_aver;
}
```

Circuito eléctrico/electrónico de conexiones para la medición de voltaje.

RESISTENCIAS



ARDUINO MEGA 2560



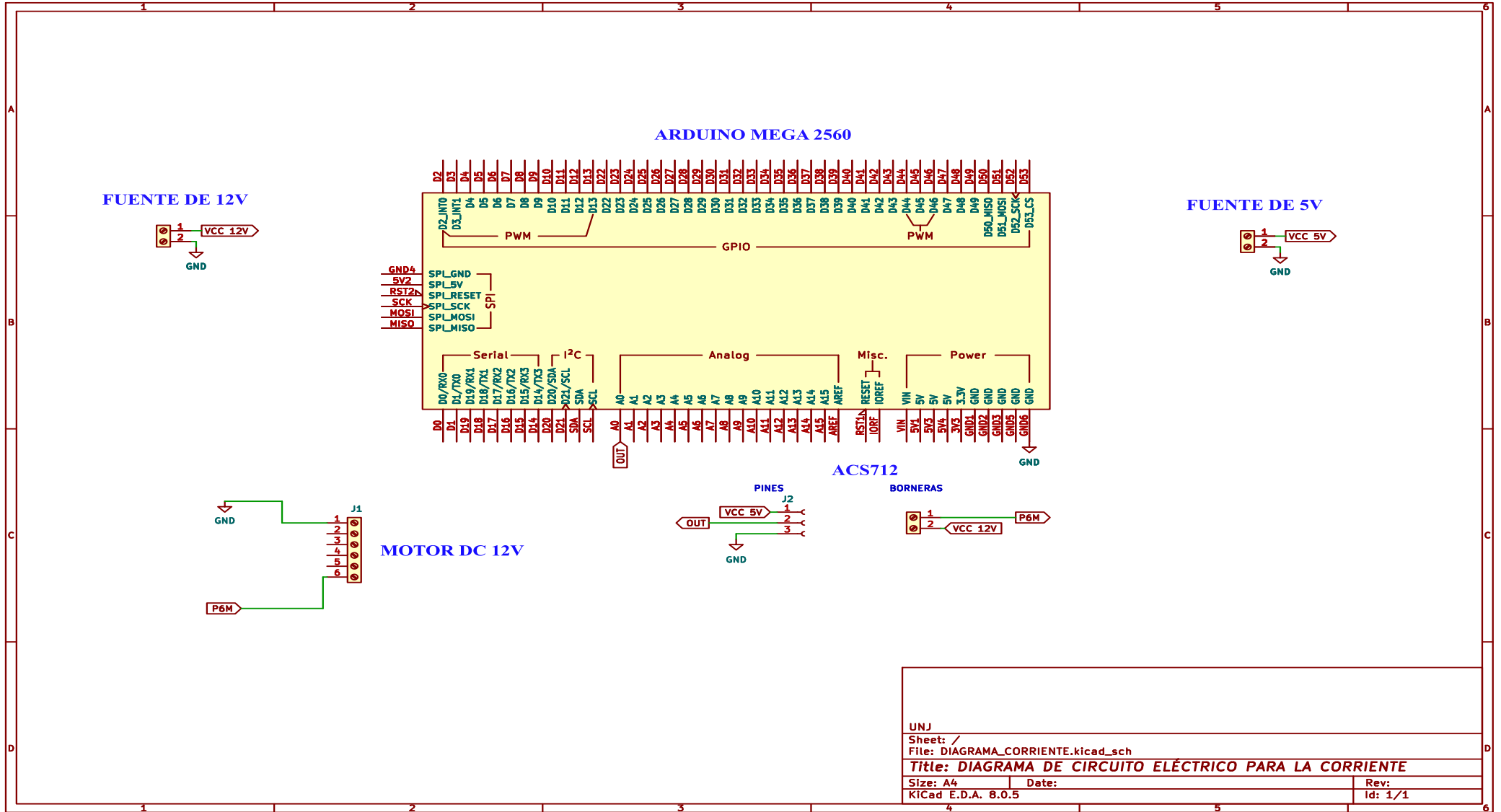
MOTOR DC 12V

UNJ	
Sheet: /	
File: DIAGRAMA_VOLTAJE.kicad_sch	
Title: CIRCUITO ELÉCTRICO PARA MEDIR VOLTAJE	
Size: A4	Date:
KiCad E.D.A. 8.0.5	Rev: 1/1

Código de programación para la medición de corriente

```
// NOTA: Adaptado de: https://naylampmechatronics.com/blog/48\_tutorial-sensor-de-corriente-acs712.html //  
// Medición de corriente haciendo uso del sensor ACS712ELCTR-05A-T.  
  
//Sensibilidad para ACS712ELCTR-05B-T = 180 mv/a  
//Sensibilidad para ACS712ELCTR-20A-T = 100 mv/a  
//Sensibilidad para ACS712ELCTR-30B-T = 66 mv/a  
  
float Sensibilidad=0.180; //sensibilidad en (Voltios/Amperio)para el sensor  
de 20A.  
  
void setup() {  
  Serial.begin(9600); //se inicia la comunicación serial con una velocidad  
de transmisión de 9600 baudios  
}  
  
void loop() {  
  
  float I=get_corriente(500); //obtenemos la corriente promedio de 500  
muestras  
  Serial.print("Corriente(A): "); //se imprime en el monitor serial la  
palabra corriente  
  Serial.println(I,3); // seguido de la palabra corriente se imprime el  
valor de I con 3 decimales.  
  delay(1000); //espera 1 segundo  
}  
  
//función para generar las 500 muestras  
  
float get_corriente(int n_muestras)  
{  
  float voltajeSensor;  
  float corriente=0;  
  for(int i=0;i<n_muestras;i++) //repite el bloque de sentencias encerrado  
entre las llaves un número determinado de veces:for (inicialización;  
condición; expresión)  
  {  
    voltajeSensor = analogRead(A0) * (5.2 / 1023.0); //lectura del sensor  
    corriente=corriente+((voltajeSensor-2.5)/Sensibilidad); //Ecuación para  
obtener la corriente  
  }  
  corriente=corriente/n_muestras;  
  return(corriente);  
}
```

Circuito eléctrico/electrónico de conexiones para la medición de corriente.



Código de programación para la medición de rpm.

// NOTA: Adaptado de: <https://www.joober.eu/calculo-de-las-revoluciones-por-minuto-rpm-de-un-motor-con-encoder-con-arduino/> //

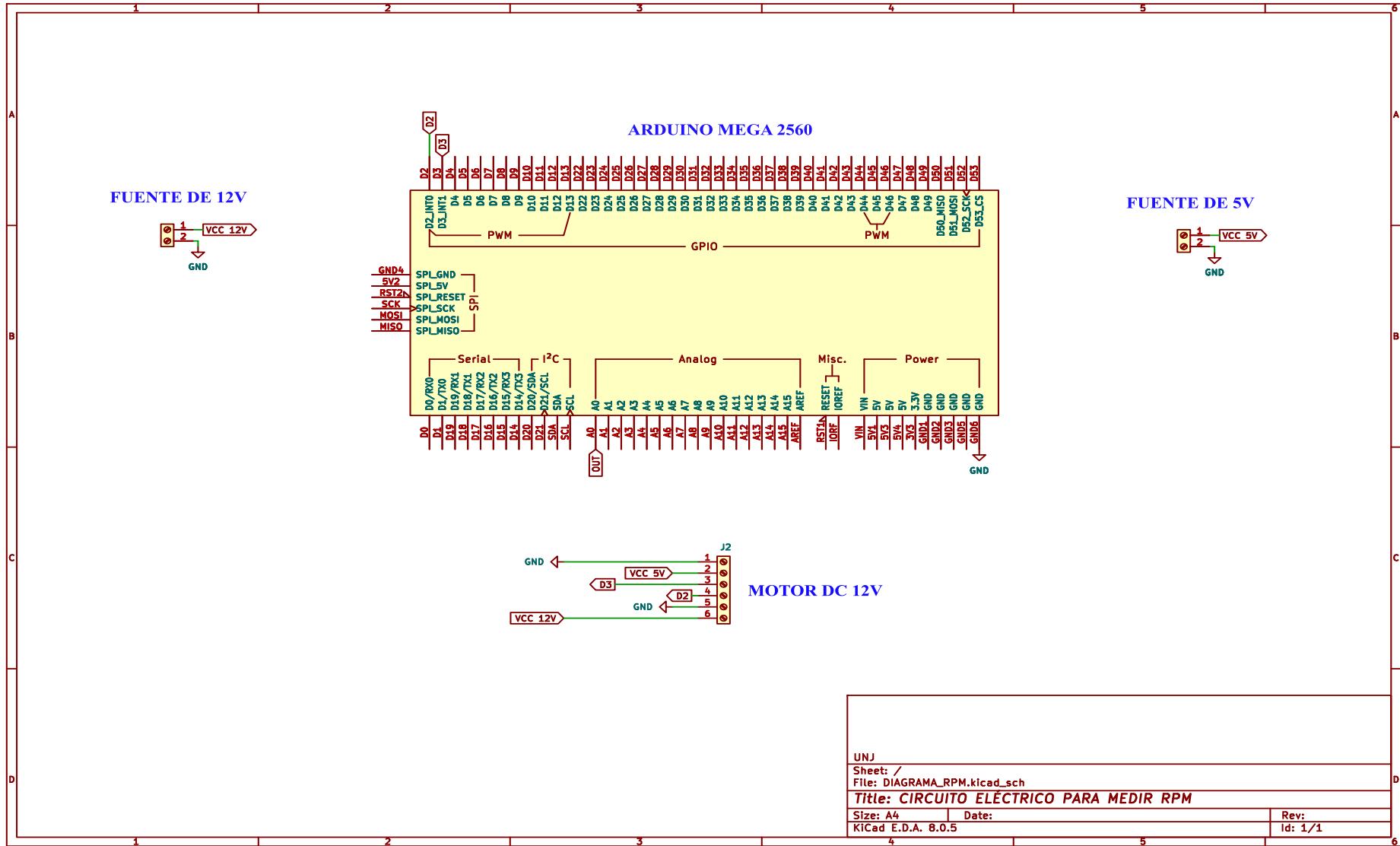
```
//Motor de CD con motorreductor, caja de engranes metálicos y encoder
cuadratura.
const byte Encoder_C1 = 3; // Cable amarillo pin 3 digital
const byte Encoder_C2 = 4; // Cable verde al pin 4 digital
byte Encoder_C1Last;
int paso = 0;
boolean direccion;
float rpm = 0;
unsigned long timeold = 0;

void setup()
{
  Serial.begin(57600);
  attachInterrupt(digitalPinToInterrupt(3), calculapulso, CHANGE);
}
void loop()
{
  pinMode(Encoder_C2, INPUT);
  /*Serial.print("Paso: ");
  Serial.print(paso);
  Serial.print(", ");
  Serial.print("Ángulo: ");
  Serial.println(paso * 0.481283422459893);*/
  /*Para por cada vuelta del rotor antes de la caja de engranes, se producen
  11 pulsos (o pasos) del encoder.
  El motor tiene una relación de reducción en su caja de engranes de
  1:119. Por tanto, se tienen 1309 ticks o pulsos
  del encoder por cada revolución del rotor después de la caja de
  engranes. Por lo que 1039/360 = 0.275...*/
  if (millis() - timeold >= 1000) {
    rpm = paso* (0.2/(millis() - timeold)) * (60000/360);
    Serial.print("RPM: ");
    Serial.println(rpm);
    timeold = millis();
    paso = 0;
  }

}

void calculapulso()
{
  int Lstate = digitalRead(Encoder_C1);
  if ((Encoder_C1Last == LOW) && Lstate == HIGH)
  {
    int val = digitalRead(Encoder_C2);
    if (val == LOW && direccion)
    {
      direccion = false; //Reverse
    }
    else if (val == HIGH && !direccion)
    {
      direccion = true; //Forward
    }
  }
  Encoder_C1Last = Lstate;
  if (!direccion) paso++;
  else paso--;
}
```

Circuito eléctrico/electrónico de conexiones para la medición de rpm



Código de programación para obtener parámetros del motor corriente y velocidad.

```
//*****//
//*****//
//***** MEDICIÓN DE PARÁMETROS DE UN MOTOR DC *****//
//***** "JGA25-370 de 12V/100RPM" *****//
//*****//
//***** Versión 1 (2023) *****//
//***** AUTOR: HENRY PINEDO NAVA *****//
//***** Versión 2 (2024). *****//
//***** AUTORES: RIMAPA RONCAL JHON DARWIN *****//
//***** MONTOYA PINTADO ELISEO *****//
//**NOTA: Adaptado de múltiples recursos disponibles en web *****//
//*** https://forum.arduino.cc/t/speed-control-of-dc-motor-with-arduino/1238457 ***//
//*** https://github.com/adafruit/RTClib/blob/master/examples/ds3231/ds3231.ino ***//
//*** https://forum.arduino.cc/t/picos-en-la-senal-del-pulsein-al-calcular-los-hz/883192 ***//
//*** https://naylampmechatronics.com/blog/48\_tutorial-sensor-de-corriente-ac3712.html ***//
//*****//
//*****//
/*=====
// Librería para la comunicación I2C
//=====
*/
#include <Wire.h>

/*=====
// Conexión para controlar el motor DC usando la placa
// "Driver Puente H BTS7960 43A"
//=====
// BTS7960 Pin 1 (RPWM) to Arduino pin 5(PWM)
// BTS7960 Pin 2 (LPWM) to Arduino pin 6(PWM)
// BTS7960 Pin 3 (R_EN) to Arduino 5V == pin 7
// BTS7960 Pin 4 (L_EN) to Arduino 5V == pin 7
// BTS7960 Pin 5 (R_IS) not connected
// BTS7960 Pin 6 (L_IS) not connected
// BTS7960 Pin 7 (VCC) to Arduino 5V pin
// BTS7960 Pin 8 (GND) to Arduino GND
//=====
*/
#include "BTS7960.h"
const uint8_t EN_Output = 7;
const uint8_t RPWM_Output = 5; // Arduino PWM output pin 5; connect to IBT-2 pin 1 (RPWM)
const uint8_t LPWM_Output = 6; // Arduino PWM output pin 6; connect to IBT-2 pin 2 (LPWM)
BTS7960 motorController(EN_Output, LPWM_Output, RPWM_Output);

/*=====
// Librería para la "RTClib"
//=====
*/
#include <RTClib.h>
RTC_DS3231 rtc; // Declaramos un RTC DS3231

/*=====
// Librería para registro de datos en "SD"
//=====
*/
#include <SdFat.h>
SdFat sd; //Formato en la que trabaja la Sd=fat
File test;
```

```

File myFile;
int pinSD = 4; // PIN CS EN SD shield

/*=====
// ENCODER MOTOR DC
//=====
*/
const byte Encoder_C1 = 2; // Cable amarillo pin 2 digital
const byte Encoder_C2 = 3; // Cable verde al pin 3 digital
byte Encoder_C1Last; // Encoder_C1 al final
int paso = 0;
boolean direccion;
float rpm = 0;
unsigned long timeold = 0;

/*=====
// Variables configuracion velocidad
//=====
*/
int total;
int sensorValue = 0;

/*=====
// TIMING ESTIMATE
// 56 delay2rps 24 [ms]
// ]-----[ ]-----[ ]-----[
// ]-----[ Periodo entre muestras
//=====
*/
unsigned long t1 = 0;
unsigned long t2 = 0;
float dd = 0;

/*=====
// Medidas de configuración para la corriente "ACS712T-5A"
//=====
*/
const int analogPinA1 = A0; // ACS712 Pin (OUT) to Arduino pin analógico A0.

/*=====
// Configuración para la medición de Voltaje_Usando resistencias
//=====
*/
// max voltage 12, min voltage 0
const float arduinoVCC = 5.01; // Voltaje del Arduino
unsigned long ValueR1 = 380; // Maxv = 5 (1 + ValueR2/ValueR1)
unsigned long ValueR2 = 560; // 1.4 =ValueR2/ValueR1
const int inputResolution = 1023; //
float Sensibilidad1 = 0.8; // sensibilidad en Voltios/Amperio para sensor de
5A_Modificado
const int analogPinV = A2; // El pin que se conecta al arduino.

/*=====
// Tren de pulsos
//=====
*/
int tpu[] = { 30, 90, 150, 210 }; // Cadena de pulsos arriba
int tpd[] = { 60, 120, 180, 210 }; // Cadena de pulsos abajo
int top = 1023; //12V

```

```

int bottom = 1023/2;                //6V

/*=====
// Medidas y configuración de variables
//=====
*/
float voltage;

const float volt_n_aver = 660;      // número de lecturas
const float curr_n_aver = 660;      // número de lecturas
volatile float voltage_V = 0.0;
volatile float current_A1 = 0.0;
volatile float power_W1 = 0.0;

String filename;                    //cadena del nombre del archivo
String header;                      //cadena de encabezamiento

int report = 1;                     //reporte es igual a 1

/*=====
// Inicialización de la SD
//=====
*/
void initialization_SD() {

    if (report == 1) Serial.println("Iniciando SD ...");
    if (!sd.begin(pinSD)) {
        if (report == 1) Serial.println("No se pudo inicializar");
        return;
    }
    if (report == 1) Serial.println("inicializacion SD exitosa");
}

void setup(void) {

    //=====
    // Configuración del monitor serial
    //=====
    Serial.begin(115200);
    while (!Serial) {
        // will pause Zero, Leonardo, etc until serial console opens
        delay(1); // demora
    }

    Serial.println("Hello!");
    Serial.println("Medición de parametros usando el sensor, ACS712 y el circuito divisor de
Voltaje...");
    //=====
    // Comprobamos si tenemos el RTC conectado
    //=====
    if (report == 1) Serial.println("Verificando RTC");
    if (!rtc.begin()) {
        if (report == 1) Serial.println("No hay un módulo RTC");
        Serial.flush();
        while (1);
    }
    //=====
    // Ponemos en hora, solo la primera vez, luego comentar y volver a cargar.

```

```

//=====
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

//=====
//Encoders interrupt
//=====
attachInterrupt(digitalPinToInterrupt(2), calculapulso, CHANGE);

//=====
initialization_SD();

Serial.println("Finishing setup");
}

void loop(void) {
//=====
// Habilitando controladores del motor
//=====
motorController.Enable();
//=====
// Experimentos: x = números de experimentos
//=====
for (int x = 1; x <= 2; x++) {
  filename = "datalog" + String(x) + ".csv";
  if (report == 1) Serial.println("#####");
  if (report == 1) Serial.print("Archivo: ");
  if (report == 1) Serial.println(filename);
  header = "Experimento " + String(x);
  if (report == 1) Serial.print("Cabecera: ");
  if (report == 1) Serial.println(header);
  if (report == 1) Serial.println("#####");

  myFile = sd.open(filename, O_WRITE | O_CREAT);
  myFile.println(header);
  myFile.println("YYYY.MM.DD,HH:MM:SS,Relative Time [s],Voltage [V],Current [A],Power [W],
RPM, Set PWM(0-1023), PWM(0-255)");
  myFile.close();

  int experimentflag = 1;

  t1 = millis(); // Start time

  while (experimentflag == 1) {
    //if(report == 0) {
    int time1 = millis();
    //}

    //=====
    t2 = millis(); //tiempo actual
    dd = float(t2 - t1) / 1000; //tiempo de demora

    //=====
    sensorValue = bottom;
    if (dd >= tpu[0] and dd < tpd[0]) {
      sensorValue = top;
    }
  }
}

```

```

if (dd >= tpu[1] and dd < tpd[1]) {
    sensorValue = top;
}

if (dd >= tpu[2] and dd < tpd[2]) {
    sensorValue = top;
}

if (dd >= tpd[3]) {
    sensorValue = 0;
    if (report == 1) Serial.println("Apagando motor");
    experimentflag = 0;
}

DateTime now = rtc.now();
int total = map(sensorValue, 0, 1023, 0, 255); // sensor value is in the range 0 to
1023
motorController.TurnLeft(total);

//=====
if (report == 1) {
    Serial.println("=====");
    Serial.print("E" + String(x) + ": ");
    if (sensorValue == top) {
        Serial.println("ALTO=12V");
    }
    else {
        Serial.println("MEDIO=6V");
    }
    Serial.print("E" + String(x) + ": ");
    Serial.print(dd);
    Serial.println(" [s]");
    Serial.print("E" + String(x) + ": ");
    Serial.print(now.year());
    Serial.print(".");
    Serial.print(now.month());
    Serial.print(".");
    Serial.print(now.day());
    Serial.print("-");
    Serial.print(now.hour());
    Serial.print(':');
    Serial.print(now.minute());
    Serial.print(':');
    Serial.println(now.second());
}

//=====
//ACS712 Y DIVISOR DE VOLTAJE
readVoltage();
voltage_V = getVoltageAverage(volt_n_aver);
current_A1 = get_corriente(curr_n_aver, analogPinA1, Sensibilidad1); //obtenemos la
corriente promedio de 140 muestras
power_W1 = voltage_V * current_A1;

//=====
//ACS712.
if (report == 1) {
    //=====
    //ACS712 Y DIVISOR DE VOLTAJE

```

```

Serial.print("E" + String(x) + ": ");
Serial.print("ACS712-5A: ");
Serial.print("V_A: ");
Serial.print(voltage_V);
Serial.print(" V, ");
Serial.print("I_A: ");
Serial.print(current_A1, 3);
Serial.print(" A, ");
Serial.print("P_A: ");
Serial.print(power_W1);
Serial.println(" W");
}
//=====
//INICIO DE RPM
pinMode(Encoder_C2, INPUT);
/*Para por cada vuelta del rotor antes de la caja de engranes, se producen 11 pulsos
(o pasos) del encoder.
El motor tiene una relación de reducción en su caja de engranes de 1:119. Por tanto,
se tienen 2618 ticks o pulsos
del encoder por cada revolución del rotor después de la caja de engranes. Por lo que
1309/360 = 0.2...*/
if (millis() - timeold >= 1000) {
rpm = paso* (0.2/(millis() - timeold)) * (60000/360); //0.2750190985
timeold = millis();
paso = 0;
}
if (report == 1) {
Serial.print("E" + String(x) + ": ");
Serial.print("      RPM: ");
Serial.println(rpm);
Serial.println("=====");
}

//=====
// Grabación de datos en la SD
myFile = sd.open(filename, O_WRITE | O_APPEND);
if (report == 1) Serial.print("Actualizando datos en ");
if (report == 1) Serial.println(filename);
myFile.print(now.year());
myFile.print(".");
myFile.print(now.month());
myFile.print(".");
myFile.print(now.day());
myFile.print(",");
myFile.print(now.hour());
myFile.print(':');
myFile.print(now.minute());
myFile.print(':');
myFile.print(now.second());
myFile.print(",");
myFile.print(String(dd, 2));
myFile.print(",");
myFile.print(String(voltage_V, 2));
myFile.print(",");
myFile.print(String(current_A1, 3));
myFile.print(",");
myFile.print(String(power_W1, 2));
myFile.print(",");
myFile.print(int(rpm));

```

```

myFile.print(",");
myFile.print(sensorValue);
myFile.print(",");
myFile.println(total);
myFile.close();
int experimentflag = 1;

//=====
if (dd >= 210)
{
    if (report == 1) Serial.println("End of Experiment");
    experimentflag = 0;

    if (x == 2)
    {
        exit(0);
    }
}

//=====

if (report == 0) {
    int time2 = millis();
    Serial.println("*****");
    //Serial.print("Tiempo total de muestra : ");
    //Serial.println(time2-time1);
    Serial.print("exp: ");
    Serial.println(x);
    Serial.print("time: ");
    Serial.println(dd);
    Serial.println("*****");
}
}
}
}

void calculapulso() {
int Lstate = digitalRead(Encoder_C1);
if ((Encoder_C1Last == LOW) && Lstate == HIGH) {
    int val = digitalRead(Encoder_C2);
    if (val == LOW && direccion) {
        direccion = false; //Reverse
    }
    else if (val == HIGH && !direccion) {
        direccion = true; //Forward
    }
}
Encoder_C1Last = Lstate;
if (!direccion) paso++;
else paso--;
}

void readVoltage() {
//Robojax.com ARDVC-01 Measure any voltage with Arduino
//This code is explained at my Arduino Course at udemy.com http://robojax.com/L/?id=62
volatile int A0Value = analogRead(analogPinV);
volatile float voltage_sensed = A0Value * (arduinoVCC / (float)inputResolution);
//Serial.print("A0Value: ");
//Serial.print(A0Value,5);

```

```

//Serial.print(" voltage_sensed: ");
//Serial.print(voltage_sensed,5);
voltage_V = voltage_sensed * (1 + ((float)ValueR2 / (float)ValueR1));
//Serial.print(" voltage_V: ");
//Serial.println(voltage_V,5);
//Robojax.com ARDVC-01 Measure any voltage with Arduino
}

float getVoltageAverage(int volt_n_aver)
{
//Robojax.com ARDVC-01 Measure any voltage with Arduino
//This code is explained at my Arduino Course at udey.com http://robojax.com/L/?id=62
volatile float voltage_temp_average = 0;
for (int i = 0; i < volt_n_aver; i++)
{
    readVoltage();
    voltage_temp_average += voltage_V;
}

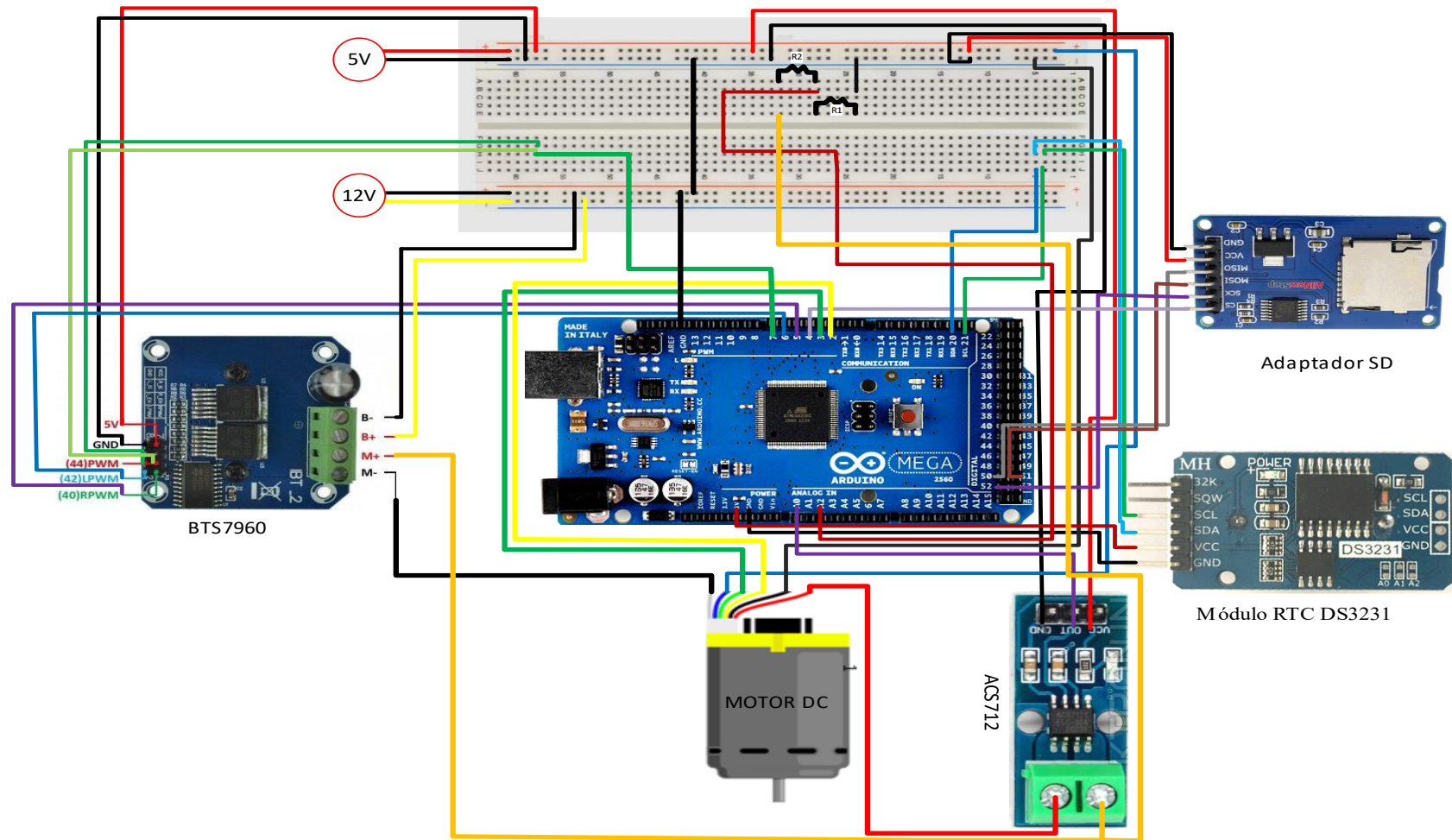
return voltage_temp_average / volt_n_aver;
}

float get_corriente(int n_muestras, int analogPinA, float Sensibilidad) {
    float voltajeSensor;
    float corriente = 0;
    for (int i = 0; i < n_muestras; i++) {

        voltajeSensor = analogRead(analogPinA) * (5.0 / 1023.0);        ////lectura del sensor
        corriente = corriente + (voltajeSensor - 2.5) / Sensibilidad; //Ecuación para obtener
la corriente
    }
    return corriente / n_muestras;
}

```

Diagrama de conexiones para obtener parámetros del motor corriente y velocidad.



Programación en Matlab para la validación gráfica.

%Nota: Adaptado de:

https://la.mathworks.com/help/matlab/ref/xlsread.html?s_tid=doc_ta#responsive_offcanvas

https://la.mathworks.com/help/matlab/ref/plot.html?searchHighlight=plot&s_tid=srchtitle_support_results_1_plot

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
GRÁFICA DE LOS PARÁMETROS PARA EL MOTOR DC 12V
Autor: Rimapa Roncal Jhon Darwin
Montoya Pintado Eliseo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%limpiar
```

```
clc %limpiar command window.
```

```
clear all %limpiar workspace.
```

```
close all
```

```
%Leer el archivo.xlsx (datos de los parametros del excel).
```

```
[num,txt,row]=xlsread("PARAMETROS_FINALS_DEL_MOTOR_DC.xlsx");
```

```
t = num (:,1); %columna 1 = tiempo
```

```
i_exp = num (:,3); %columna 3 = corriente experimental
```

```
i_mod = num (:,4); %columna 4 = corriente modelo
```

```
w_exp = num (:,5); %columna 5 = velocidad experimental
```

```
w_mod = num (:,6); %columna 6 = velocidad modelo
```

```
%Graficar
```

```
figure(1)
```

```
%subplot (211) %divide la figura en 2 filas y 1 columna (selecciona sub1).
```

```
plot(t,i_exp,'r',t,i_mod,'b--','LineWidth',1) %grafica la salida t vs i.
```

```
title ('VALIDACIÓN GRÁFICA DE LA CORRIENTE','FontSize', 20) %título figura
```

```
ylabel('Corriente (A)', 'FontSize',15) %etiqueta 'y' como corriente.
```

```
xlabel('Tiempo (s)','FontSize',15) %etiqueta 'x' como tiempo.
```

```
legend('Corriente experimental','Corriente modelo','fontSize',12)
```

```
figure(2)
```

```
%subplot (212) %divide la figura en 2 filas y 1 columna (selecciona sub2).
```

```
plot(t,w_exp,'r',t,w_mod,'b--','LineWidth',1) %grafica la salida t vs i.
```

```
title ('VALIDACIÓN GRÁFICA DE LA VELOCIDAD','FontSize', 20) %título figura
```

```
ylabel('Velocidad (rad/s)', 'FontSize',15) %etiqueta 'y' como corriente.
```

```
xlabel('tiempo (s)','FontSize',15) %etiqueta 'x' como tiempo.
```

```
legend('Velocidad experimental','Velocidad modelo','fontSize',12)%leyenda corriente  
& tiempo.
```

Maqueta

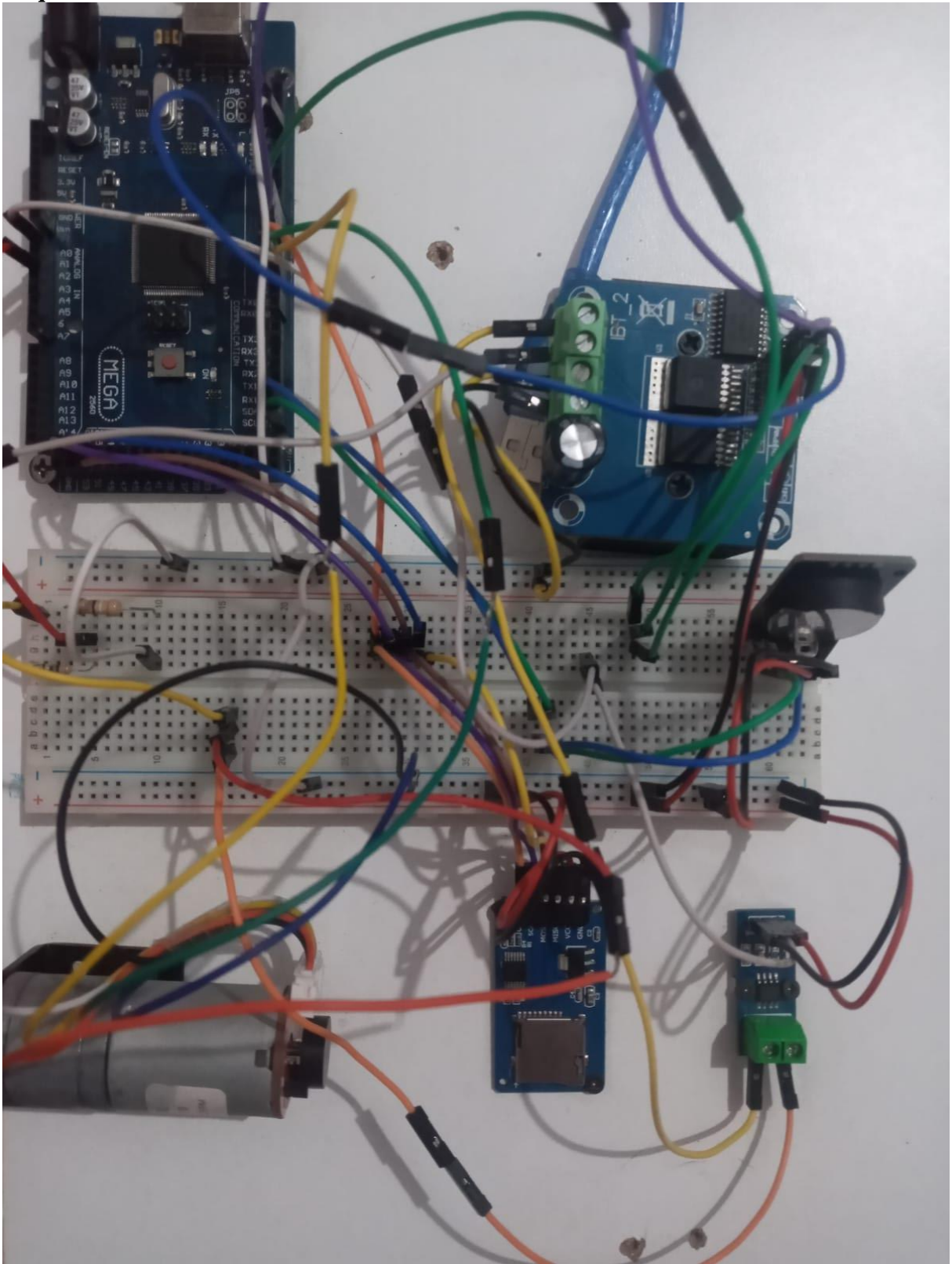


Tabla 13*Datos de los parámetros experimentales y simulados de la corriente y velocidad*

Time	Voltage [V]	Current [A] (EXPERIMENT)	Current [A] (SIMULATION)	speed [rad/s] (EXPERIMENT)	speed [rad/s] (SIMULATION)
0,00	0	0,000	0,000	0,000	0,000
0,25	6	0,113	0,014	5,131	5,261
0,50	6	0,113	0,014	5,131	5,261
0,75	6	0,113	0,014	5,131	5,261
1,00	6	0,113	0,014	5,131	5,261
1,25	6	0,112	0,014	5,131	5,261
1,50	6	0,112	0,014	5,131	5,261
1,75	6	0,112	0,014	5,131	5,261
2,00	6	0,113	0,014	5,131	5,261
2,25	6	0,113	0,014	5,131	5,261
2,50	6	0,112	0,014	5,131	5,261
2,75	6	0,112	0,014	5,131	5,261
3,00	6	0,111	0,014	5,131	5,261
3,25	6	0,112	0,014	5,131	5,261
3,50	6	0,112	0,014	5,131	5,261
3,75	6	0,113	0,014	5,131	5,261
4,00	6	0,113	0,013	5,131	5,261
4,25	6	0,113	0,014	5,131	5,261
4,50	6	0,114	0,014	5,131	5,261
4,75	6	0,113	0,014	5,131	5,261
5,00	6	0,113	0,014	5,131	5,261
5,25	6	0,113	0,014	5,131	5,261
5,50	6	0,114	0,014	5,131	5,261
5,75	6	0,113	0,014	5,131	5,261
6,00	6	0,113	0,014	5,131	5,261
6,25	6	0,113	0,014	5,131	5,261
6,50	6	0,113	0,014	5,131	5,261
6,75	6	0,113	0,014	5,131	5,261
7,00	6	0,113	0,014	5,131	5,261
7,25	6	0,113	0,014	5,131	5,261
7,50	6	0,113	0,014	5,131	5,261
7,75	6	0,112	0,014	5,131	5,261
8,00	6	0,112	0,014	5,131	5,261
8,25	6	0,113	0,014	5,131	5,261
8,50	6	0,112	0,014	5,131	5,261
8,75	6	0,113	0,013	5,131	5,261
9,00	6	0,113	0,014	5,131	5,261
9,25	6	0,112	0,014	5,131	5,261
9,50	6	0,113	0,014	5,131	5,261
9,75	6	0,112	0,014	5,131	5,261
10,00	6	0,117	0,014	5,131	5,261
10,25	12	0,118	0,029	10,786	10,703
10,50	12	0,119	0,028	10,786	10,703

10,75	12	0,119	0,029	10,786	10,703
11,00	12	0,119	0,029	10,786	10,703
11,25	12	0,119	0,029	10,786	10,703
11,50	12	0,119	0,029	10,786	10,703
11,75	12	0,119	0,029	10,786	10,703
12,00	12	0,119	0,029	10,786	10,703
12,25	12	0,118	0,029	10,786	10,703
12,50	12	0,119	0,029	10,786	10,703
12,75	12	0,119	0,029	10,786	10,703
13,00	12	0,119	0,029	10,786	10,703
13,25	12	0,119	0,029	10,786	10,703
13,50	12	0,119	0,029	10,786	10,703
13,75	12	0,118	0,029	10,786	10,703
14,00	12	0,118	0,029	10,786	10,703
14,25	12	0,119	0,029	10,786	10,703
14,50	12	0,119	0,029	10,786	10,703
14,75	12	0,120	0,029	10,786	10,703
15,00	12	0,120	0,029	10,786	10,703
15,25	12	0,119	0,029	10,786	10,703
15,50	12	0,118	0,029	10,786	10,703
15,75	12	0,118	0,029	10,786	10,703
16,00	12	0,118	0,029	10,786	10,703
16,25	12	0,119	0,029	10,786	10,703
16,50	12	0,119	0,029	10,786	10,703
16,75	12	0,118	0,029	10,786	10,703
17,00	12	0,119	0,029	10,786	10,703
17,25	12	0,119	0,029	10,786	10,703
17,50	12	0,118	0,029	10,786	10,703
17,75	12	0,118	0,029	10,786	10,703
18,00	12	0,118	0,029	10,786	10,703
18,25	12	0,118	0,028	10,786	10,703
18,50	12	0,118	0,029	10,786	10,703
18,75	12	0,119	0,029	10,786	10,703
19,00	12	0,119	0,029	10,786	10,703
19,25	12	0,119	0,029	10,786	10,703
19,50	12	0,119	0,029	10,786	10,703
19,75	12	0,119	0,029	10,786	10,703
20,00	12	0,111	0,028	10,786	10,703
20,25	6	0,112	0,015	5,131	5,261
20,50	6	0,112	0,015	5,131	5,261
20,75	6	0,112	0,015	5,131	5,261
21,00	6	0,112	0,014	5,131	5,261
21,25	6	0,113	0,015	5,131	5,261
21,50	6	0,112	0,015	5,131	5,261
21,75	6	0,113	0,015	5,131	5,261
22,00	6	0,113	0,015	5,131	5,261
22,25	6	0,113	0,015	5,131	5,261

22,50	6	0,112	0,015	5,131	5,261
22,75	6	0,112	0,015	5,131	5,261
23,00	6	0,113	0,015	5,131	5,261
23,25	6	0,113	0,015	5,131	5,261
23,50	6	0,113	0,015	5,131	5,261
23,75	6	0,113	0,015	5,131	5,261
24,00	6	0,113	0,015	5,131	5,261
24,25	6	0,113	0,015	5,131	5,261
24,50	6	0,113	0,015	5,131	5,261
24,75	6	0,112	0,014	5,131	5,261
25,00	6	0,112	0,016	5,131	5,261
25,25	6	0,112	0,015	5,131	5,261
25,50	6	0,113	0,015	5,131	5,261
25,75	6	0,112	0,015	5,131	5,261
26,00	6	0,112	0,015	5,131	5,261
26,25	6	0,112	0,016	5,131	5,261
26,50	6	0,113	0,015	5,131	5,261
26,75	6	0,112	0,015	5,131	5,261
27,00	6	0,112	0,015	5,131	5,261
27,25	6	0,111	0,015	5,131	5,261
27,50	6	0,112	0,015	5,131	5,261
27,75	6	0,112	0,015	5,131	5,261
28,00	6	0,112	0,015	5,131	5,261
28,25	6	0,112	0,015	5,131	5,261
28,50	6	0,112	0,015	5,131	5,261
28,75	6	0,113	0,015	5,131	5,261
29,00	6	0,112	0,015	5,131	5,261
29,25	6	0,112	0,015	5,131	5,261
29,50	6	0,112	0,015	5,131	5,261
29,75	6	0,113	0,015	5,131	5,261
30,00	6	0,117	0,015	5,131	5,261
30,25	12	0,118	0,029	10,891	10,703
30,50	12	0,119	0,029	10,891	10,703
30,75	12	0,118	0,029	10,891	10,703
31,00	12	0,119	0,029	10,891	10,703
31,25	12	0,119	0,029	10,891	10,703
31,50	12	0,119	0,029	10,891	10,703
31,75	12	0,118	0,029	10,891	10,703
32,00	12	0,119	0,029	10,891	10,703
32,25	12	0,118	0,029	10,891	10,703
32,50	12	0,118	0,029	10,891	10,703
32,75	12	0,119	0,029	10,891	10,703
33,00	12	0,119	0,029	10,891	10,703
33,25	12	0,119	0,028	10,891	10,703
33,50	12	0,119	0,029	10,891	10,703
33,75	12	0,119	0,029	10,891	10,703
34,00	12	0,119	0,029	10,891	10,703

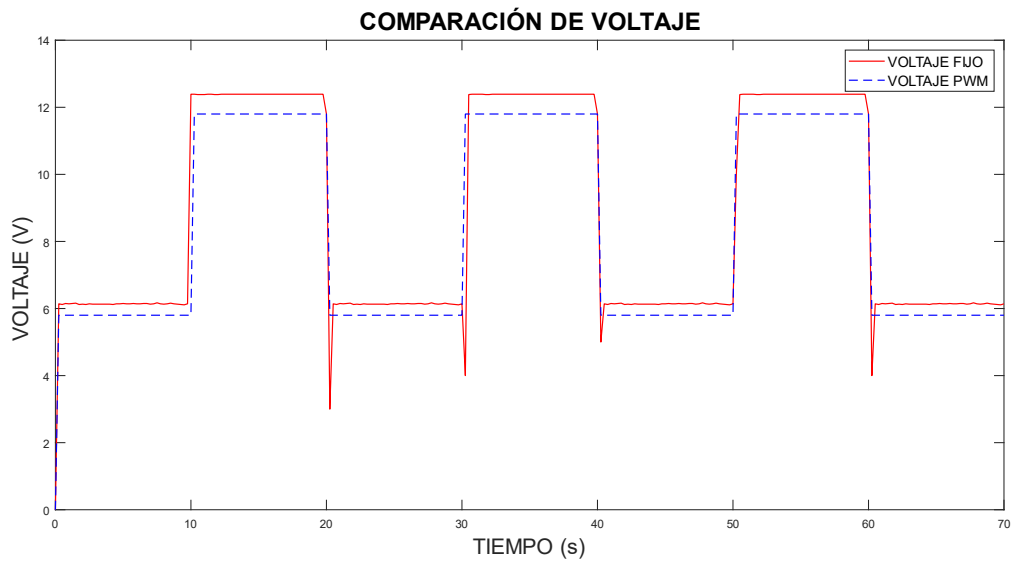
34,25	12	0,119	0,029	10,891	10,703
34,50	12	0,120	0,028	10,891	10,703
34,75	12	0,118	0,029	10,891	10,703
35,00	12	0,119	0,029	10,891	10,703
35,25	12	0,119	0,029	10,891	10,703
35,50	12	0,119	0,029	10,891	10,703
35,75	12	0,118	0,029	10,891	10,703
36,00	12	0,119	0,029	10,891	10,703
36,25	12	0,118	0,029	10,891	10,703
36,50	12	0,118	0,029	10,891	10,703
36,75	12	0,118	0,029	10,891	10,703
37,00	12	0,118	0,029	10,891	10,703
37,25	12	0,118	0,029	10,891	10,703
37,50	12	0,118	0,029	10,891	10,703
37,75	12	0,118	0,029	10,891	10,703
38,00	12	0,118	0,029	10,891	10,703
38,25	12	0,118	0,029	10,891	10,703
38,50	12	0,119	0,029	10,891	10,703
38,75	12	0,119	0,029	10,891	10,703
39,00	12	0,118	0,029	10,891	10,703
39,25	12	0,118	0,029	10,891	10,703
39,50	12	0,118	0,029	10,891	10,703
39,75	12	0,118	0,029	10,891	10,703
40,00	12	0,111	0,029	10,891	10,703
40,25	6	0,111	0,015	5,131	5,261
40,50	6	0,111	0,015	5,131	5,261
40,75	6	0,111	0,015	5,131	5,261
41,00	6	0,112	0,015	5,131	5,261
41,25	6	0,112	0,015	5,131	5,261
41,50	6	0,112	0,015	5,131	5,261
41,75	6	0,112	0,015	5,131	5,261
42,00	6	0,112	0,015	5,131	5,261
42,25	6	0,113	0,015	5,131	5,261
42,50	6	0,112	0,015	5,131	5,261
42,75	6	0,112	0,015	5,131	5,261
43,00	6	0,112	0,015	5,131	5,261
43,25	6	0,111	0,015	5,131	5,261
43,50	6	0,112	0,015	5,131	5,261
43,75	6	0,112	0,015	5,131	5,261
44,00	6	0,112	0,015	5,131	5,261
44,25	6	0,112	0,015	5,131	5,261
44,50	6	0,112	0,015	5,131	5,261
44,75	6	0,112	0,015	5,131	5,261
45,00	6	0,112	0,015	5,131	5,261
45,25	6	0,112	0,016	5,131	5,261
45,50	6	0,111	0,015	5,131	5,261
45,75	6	0,112	0,015	5,131	5,261

46,00	6	0,112	0,015	5,131	5,261
46,25	6	0,112	0,015	5,131	5,261
46,50	6	0,112	0,015	5,131	5,261
46,75	6	0,112	0,015	5,131	5,261
47,00	6	0,112	0,015	5,131	5,261
47,25	6	0,112	0,015	5,131	5,261
47,50	6	0,112	0,015	5,131	5,261
47,75	6	0,112	0,015	5,131	5,261
48,00	6	0,112	0,015	5,131	5,261
48,25	6	0,112	0,015	5,131	5,261
48,50	6	0,112	0,015	5,131	5,261
48,75	6	0,111	0,015	5,131	5,261
49,00	6	0,111	0,015	5,131	5,261
49,25	6	0,112	0,015	5,131	5,261
49,50	6	0,112	0,015	5,131	5,261
49,75	6	0,112	0,015	5,131	5,261
50,00	6	0,116	0,015	5,131	5,261
50,25	12	0,118	0,029	10,786	10,700
50,50	12	0,118	0,029	10,786	10,700
50,75	12	0,119	0,029	10,786	10,700
51,00	12	0,118	0,029	10,786	10,700
51,25	12	0,118	0,029	10,786	10,700
51,50	12	0,118	0,029	10,786	10,700
51,75	12	0,118	0,029	10,786	10,700
52,00	12	0,118	0,029	10,786	10,700
52,25	12	0,118	0,029	10,786	10,700
52,50	12	0,118	0,029	10,786	10,700
52,75	12	0,119	0,029	10,786	10,700
53,00	12	0,119	0,029	10,786	10,700
53,25	12	0,118	0,029	10,786	10,700
53,50	12	0,118	0,029	10,786	10,700
53,75	12	0,118	0,029	10,786	10,700
54,00	12	0,118	0,029	10,786	10,700
54,25	12	0,119	0,029	10,786	10,700
54,50	12	0,119	0,028	10,786	10,700
54,75	12	0,119	0,029	10,786	10,700
55,00	12	0,119	0,029	10,786	10,700
55,25	12	0,119	0,029	10,786	10,700
55,50	12	0,118	0,029	10,786	10,700
55,75	12	0,118	0,029	10,786	10,700
56,00	12	0,118	0,029	10,786	10,700
56,25	12	0,118	0,029	10,786	10,700
56,50	12	0,118	0,029	10,786	10,700
56,75	12	0,119	0,029	10,786	10,700
57,00	12	0,119	0,029	10,786	10,700
57,25	12	0,119	0,029	10,786	10,700
57,50	12	0,119	0,029	10,786	10,700

57,75	12	0,119	0,029	10,786	10,700
58,00	12	0,119	0,029	10,786	10,700
58,25	12	0,120	0,029	10,786	10,700
58,50	12	0,119	0,029	10,786	10,700
58,75	12	0,119	0,029	10,786	10,700
59,00	12	0,119	0,029	10,786	10,700
59,25	12	0,119	0,029	10,786	10,700
59,50	12	0,119	0,029	10,786	10,700
59,75	12	0,118	0,029	10,786	10,700
60,00	12	0,111	0,029	10,786	10,700
60,25	6	0,112	0,015	5,131	5,261
60,50	6	0,112	0,015	5,131	5,261
60,75	6	0,111	0,015	5,131	5,261
61,00	6	0,112	0,015	5,131	5,261
61,25	6	0,112	0,015	5,131	5,261
61,50	6	0,112	0,015	5,131	5,261
61,75	6	0,112	0,016	5,131	5,261
62,00	6	0,112	0,015	5,131	5,261
62,25	6	0,112	0,015	5,131	5,261
62,50	6	0,112	0,015	5,131	5,261
62,75	6	0,112	0,015	5,131	5,261
63,00	6	0,112	0,015	5,131	5,261
63,25	6	0,112	0,015	5,131	5,261
63,50	6	0,112	0,015	5,131	5,261
63,75	6	0,112	0,015	5,131	5,261
64,00	6	0,112	0,015	5,131	5,261
64,25	6	0,113	0,015	5,131	5,261
64,50	6	0,112	0,015	5,131	5,261
64,75	6	0,112	0,015	5,131	5,261
65,00	6	0,112	0,015	5,131	5,261
65,25	6	0,112	0,015	5,131	5,261
65,50	6	0,112	0,015	5,131	5,261
65,75	6	0,112	0,015	5,131	5,261
66,00	6	0,112	0,015	5,131	5,261
66,25	6	0,112	0,015	5,131	5,261
66,50	6	0,111	0,015	5,131	5,261
66,75	6	0,111	0,015	5,131	5,261
67,00	6	0,112	0,015	5,131	5,261
67,25	6	0,111	0,015	5,131	5,261
67,50	6	0,111	0,016	5,131	5,261
67,75	6	0,111	0,015	5,131	5,261
68,00	6	0,111	0,015	5,131	5,261
68,25	6	0,111	0,015	5,131	5,261
68,50	6	0,111	0,015	5,131	5,261
68,75	6	0,111	0,015	5,131	5,261
69,00	6	0,112	0,015	5,131	5,261
69,25	6	0,112	0,015	5,131	5,261

69,50	6	0,112	0,015	5,131	5,261
69,75	6	0,111	0,015	5,131	5,261
70,00	6	0,124	0,015	5,131	5,260

Comparación de Voltajes



Primera versión con conexiones a través de tarjeta PCB

A continuación, se presenta el resultado preliminar del trabajo futuro para la formalización de las conexiones a través de tarjeta PCB, cabe resaltar que es la primera versión que está recién en evaluación de sus conexiones y de los respectivos datos que genera. Esta tarjeta PCB es la formalización de las conexiones que se muestran en la figura denominada Maqueta en la página 68 del presente documento.

