



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO PARA EL
REGISTRO TRIDIMENSIONAL AUTOMATIZADO
DE CAMPOS MAGNÉTICOS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

FÍSICO

P R E S E N T A :

JUAN RANGEL GONZÁLEZ

TUTOR

DR. RAÚL ARTURO ESPEJEL MORALES



FACULTAD DE CIENCIAS
UNAM

Cd. Universitaria, Cd. de México, 2023



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Hoja de datos del jurado

1. Datos del alumno

Rangel

González

Juan

5554509674

Universidad Nacional Autónoma de México

Facultad de Ciencias

Física

306089925

2. Datos del tutor

Dr

Raúl Arturo

Espejel

Morales

3. Datos del sinodal 1

Dr

Alipio Gustavo

Calles

Martínez

4. Datos del sinodal 2

Dr

José Ignacio

Jiménez

Mier y Terán

5. Datos del sinodal 3

Dr

Daniel José

Marín

Lámbarri

6. Datos del sinodal 4

Dr

Naser

Qureshi

7. Datos del trabajo escrito

DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO PARA EL REGISTRO TRIDIMENSIONAL AUTOMATIZADO DE CAMPOS MAGNÉTICOS

74 p

2023

AGRADECIMIENTOS

Quiero agradecer a mis padres; Concepción y Jorge por sus enseñanzas, sus consejos, su paciencia, su amor y por todo su apoyo incondicional que me han brindado para poder cumplir todos mis objetivos personales y académicos.

Agradezco a mis sinodales por tomarse su tiempo para leer mi trabajo escrito y por contribuir con sus comentarios e ideas para mejorar este proyecto.

Al Dr. Raúl Arturo Espejel Morales le agradezco muy profundamente por su apoyo, paciencia, amistad y guía para la culminación de este proyecto.

A los miembros del Laboratorio Interdisciplinario por su amistad y sus consejos.

A todas las personas, amigos y familiares que me han dado su apoyo para salir adelante y sus comentarios motivacionales.

Gracias, a la Universidad Nacional Autónoma de México (UNAM), por la beca PAPIME con número PE106423 y PAPIIT con número IN114318.

Índice general

Índice general	IV
1. Capítulo 1. Introducción	1
1.1. Motivación	2
1.2. Marco teórico	3
1.2.1. Campo magnético y líneas de campo magnético	3
1.2.2. Campo magnético de un alambre largo y recto portador de corriente.	4
1.3. Campo magnético de una espira en cualquier punto del espacio	8
1.4. Campo magnético de una bobina.	10
1.5. Planteamiento del Dispositivo	11
1.6. Control Numérico Computarizado.	12
1.7. Arduino	14
1.8. Sensor de Campo Magnético	17
1.9. Motores paso a paso	18
1.10. Driver DRV8825	19
1.11. Objetivos	21
1.11.1. Objetivo General	21
1.11.2. Objetivos Particulares	21
1.12. Hipótesis	21
2. Capítulo 2. Desarrollo experimental	23
2.1. Diseño de la Estructura del Dispositivo	23
2.2. Circuitos	30
2.3. Control	33
2.4. Automatización	35

3. Capitulo 3. Resultados **37**

3.1. Funcionamiento 37

3.2. Características 39

3.3. Mediciones 39

3.4. Conclusiones 46

A. Efecto Hall **49**

B. Código Python **51**

C. Código Arduino **63**

Capítulo 1

Introducción

Los efectos del campo magnético se han conocido desde la antigüedad, cuando se observaron por primera vez los efectos de la magnetita (imán permanente que se encuentra en forma natural) con objetos de metal. El descubrimiento de la propiedad que presenta este material tuvo una gran influencia en la navegación pues dio lugar a la invención de la brújula, que inicialmente consistía en una aguja imantada que, al orientarse con el campo magnético de la Tierra, podía indicar su norte magnético (éste difiere en aproximadamente 500 km del polo norte geográfico, lo cual no le resta utilidad).

La electricidad y el magnetismo eran dos fenómenos totalmente distintos hasta 1820 cuando Oersted descubrió que una corriente eléctrica producía un campo magnético, este descubrimiento unió ambos fenómenos. Seis años más tarde André-Marie Ampère desarrolló la teoría que explica la interacción entre el magnetismo y la electricidad.

En 1831 Michael Faraday descubrió la inducción electromagnética e inventó los precursores del actual motor eléctrico.

En 1865 James Clerk Maxwell formuló las ecuaciones que llevan su nombre, las cuales unifican la electricidad, el magnetismo y la luz como manifestaciones distintas de un mismo fenómeno.

El estudio de estos fenómenos magnéticos permitieron el desarrollo de maquinaria eléctrica, como equipos de comunicaciones, motores eléctricos, computadoras, etc. que desempeñan un papel fundamental en nuestra vida diaria.

1.1. Motivación

Existen diversos métodos para obtener una visualización del campo magnético, desde una muy común y sencilla que se usa en los laboratorios de enseñanza, que consiste en esparcir limaduras de hierro alrededor de una fuente magnética, o mediante el uso de una brújula colocada en diferentes posiciones alrededor de la fuente de campo magnético, registrando la dirección de la aguja en cada posición, hasta arreglos de sensores de efecto Hall (Apéndice A) que se usan en dispositivos comerciales, como el que se muestra en la figura 1.1.

Así pues, la caracterización de campos magnéticos generados por imanes permanentes o arreglos de bobinas, cobra importancia en diversas disciplinas como en la medicina y la ingeniería, por ejemplo:

- + Los trenes de levitación magnética, los cuales emplean un gran número de imanes para realizar la levitación, guía y propulsión del tren [9].
- + Las cocinas de inducción, las cuales calientan directamente el recipiente contenedor (material ferromagnético), mediante un campo magnético alternante [10].
- + IRM (Imagen por Resonancia Magnética), la cual produce imágenes anatómicas tridimensionales detalladas, empleando imanes poderosos que producen un potente campo magnético que obliga a los protones en el cuerpo a alinearse con ese campo [11].
- + Dispositivos médicos implantados, los cuales requieren de micro bobinas planas para la transferencia inalámbrica de energía y señales [2].

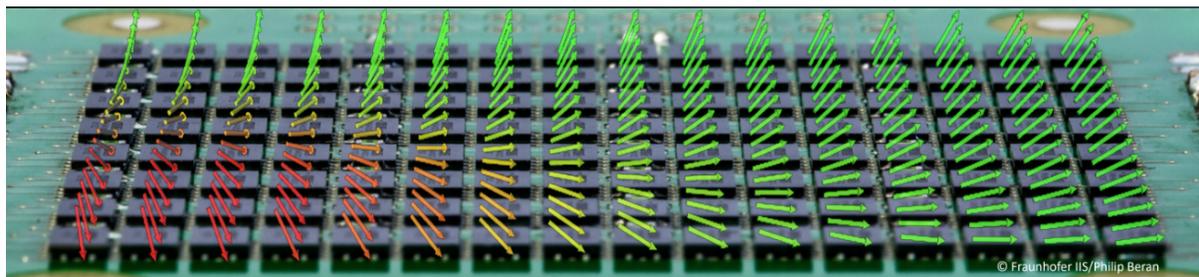


Figura 1.1: Arreglo de sensores Hall en el “HALLINSIGH”, Hall magnetic field camera [12].

Cada ejemplo antes mencionado tiene un campo magnético con una geometría muy específica, por lo cual es indispensable conocer esta geometría para aprovechar al máximo sus características.

Por lo anterior, en este proyecto, se diseñó un dispositivo de bajo costo que permite visualizar campos magnéticos en un mallado tridimensional para la caracterización de fuentes, de manera precisa y reproducible, el cual, entre otras aplicaciones, puede de ser un apoyo didáctico en los laboratorios de electromagnetismo, para ayudar en la comprensión de los conceptos de campo magnético y líneas de campo magnético.

1.2. Marco teórico

1.2.1. Campo magnético y líneas de campo magnético

Un campo magnético es una descripción matemática del efecto producido por corrientes eléctricas o materiales magnéticos. El campo magnético es un campo vectorial, por ello está especificado por una dirección y una magnitud en cada punto del espacio.

Los campos magnéticos pueden ser generados por corrientes eléctricas o por los momentos magnéticos de las partículas constituyentes de los materiales que comúnmente llamamos imanes, por ejemplo la magnetita, las aleaciones de hierro y níquel, y el neodimio.

Las líneas de un campo magnético son una forma de representar gráficamente la acción de las fuerzas magnéticas en el espacio. Son líneas que nunca se cruzan y que se acumulan en las regiones en que el campo es más intenso.

Comúnmente, para poder visualizar las líneas de campo magnético, se esparce limadura de hierro alrededor sobre una superficie cercana a una fuente de campo magnético (figura 1.2). El resultado es un patrón semejante a la geometría del campo magnético, sin embargo, con este método, aunque el patrón general para una fuente específica será siempre el mismo, la posición exacta y la densidad de las líneas formadas por la limadura, dependen de su distribución inicial, de su tamaño, etc, lo cual le resta reproducibilidad a este método.

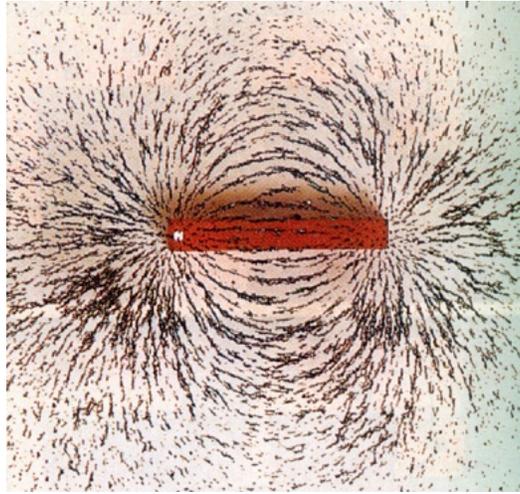


Figura 1.2: Líneas de campo magnético representadas con limaduras de hierro, imagen obtenida de internet [19].

A continuación se obtienen las ecuaciones de algunas fuentes magnéticas, para comparar los resultados teóricos con los datos obtenidos por el dispositivo.

1.2.2. Campo magnético de un alambre largo y recto portador de corriente.

Campo magnético producido por un segmento de alambre horizontal de largo L sobre el eje x centrado en el origen con corriente I medido en el punto (x, y, z) .

Consideremos, por ahora, un punto sobre el plano $z = 0$ y que la corriente va en dirección x , figura 1.3.

Aplicando la ley de Biot-Savart y tomando un segmento diferencial dh del conductor en el punto $(h, 0, 0)$:

$$\vec{B} = \frac{\mu_0}{4\pi} I \int_{-\frac{L}{2}}^{\frac{L}{2}} \frac{\hat{i} \times (\vec{r}_2 - \vec{r}_1)}{|\vec{r}_2 - \vec{r}_1|^3} dh \quad (1.1)$$

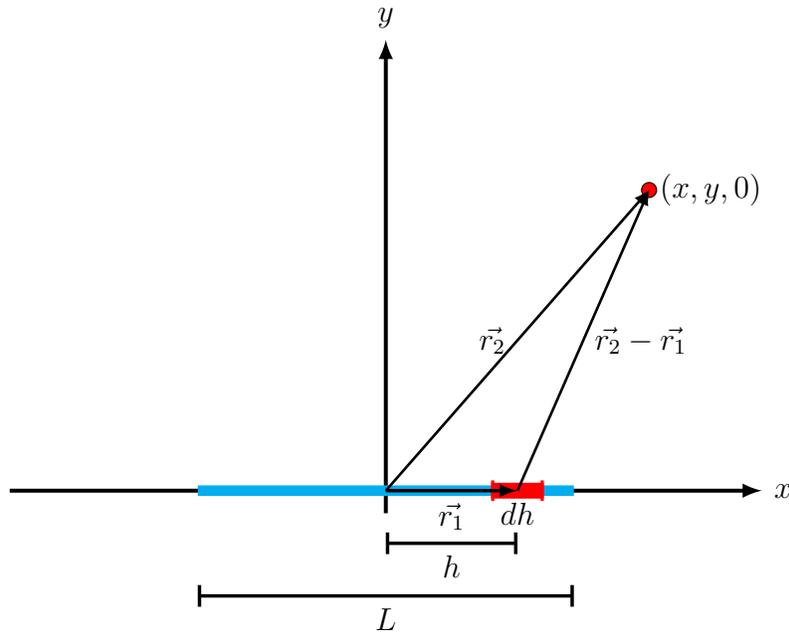


Figura 1.3: Esquema de un alambre conductor de largo L y un segmento diferencial dh , se muestra un punto sobre el plano xy .

notamos que:

$$\hat{i} \times (\vec{r}_2 - \vec{r}_1) = (1, 0, 0) \times (x - h, y, 0) = y \hat{k} \quad (1.2)$$

$$|\vec{r}_2 - \vec{r}_1|^3 = ((x - h)^2 + y^2)^{\frac{3}{2}} \quad (1.3)$$

por lo que el campo magnético puede calcularse como:

$$\vec{B} = \frac{\mu_0}{4\pi} I \int_{-\frac{L}{2}}^{\frac{L}{2}} \frac{y}{[(x - h)^2 + y^2]^{\frac{3}{2}}} dh \hat{k}. \quad (1.4)$$

Resolviendo la integral obtenemos:

$$\vec{B} = -\frac{\mu_0}{4\pi y} I \left(\frac{x - L/2}{\sqrt{(x - L/2)^2 + y^2}} - \frac{x + L/2}{\sqrt{(x + L/2)^2 + y^2}} \right) \hat{k}. \quad (1.5)$$

Ahora, aprovechando la simetría del sistema, es posible extender el cálculo a cualquier punto P fuera del plano (x, y) considerando, en vez de la coordenada y del punto donde se requiere conocer \vec{B} , la distancia ρ de este punto al eje x , como se muestra en la figura (1.4):

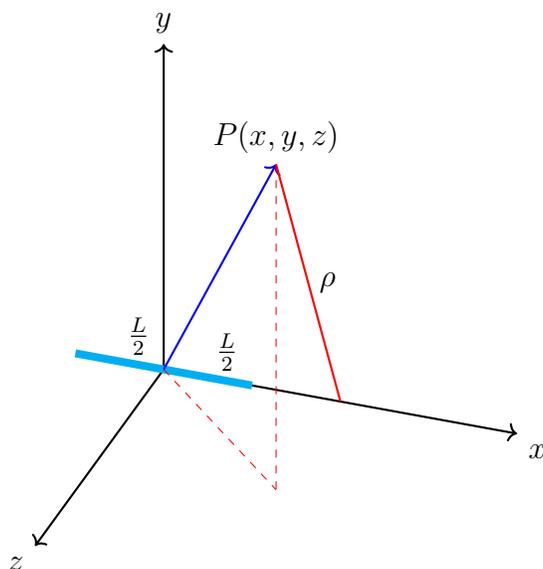


Figura 1.4: Esquema de un alambre conductor recto de longitud L sobre el eje x , se muestra un punto (x, y, z)

De este modo, es posible conocer la magnitud del campo magnético de la siguiente manera:

$$B = -\frac{\mu_0}{4\pi\rho} I \left(\frac{x - L/2}{\sqrt{(x - L/2)^2 + \rho^2}} - \frac{x + L/2}{\sqrt{(x + L/2)^2 + \rho^2}} \right), \quad (1.6)$$

y, dado que, por la simetría axial, el vector \vec{B} será siempre perpendicular al plano que forma el eje x con el punto P , podemos calcular sus componentes como:

$$\begin{aligned}
 B_x &= 0, \\
 B_y &= -B \left(\frac{z}{\rho} \right), \\
 B_z &= B \left(\frac{y}{\rho} \right),
 \end{aligned}
 \tag{1.7}$$

con $\rho = \sqrt{y^2 + z^2}$.

Con las ecuaciones anteriores obtenemos una representación del campo vectorial para el alambre recto que se muestra en la figura 1.5.

De estas imágenes observamos que las líneas de campo magnético son circulares y concéntricas al eje del alambre [5]; la dirección del campo magnético es tangente a las líneas de campo magnético y la magnitud del campo magnético disminuye al alejarse del alambre (figura 1.5a).

Los planos en los cuales se encuentran las líneas de campo magnético son perpendiculares a la dirección de la corriente I y por lo tanto al alambre (figura 1.5b).

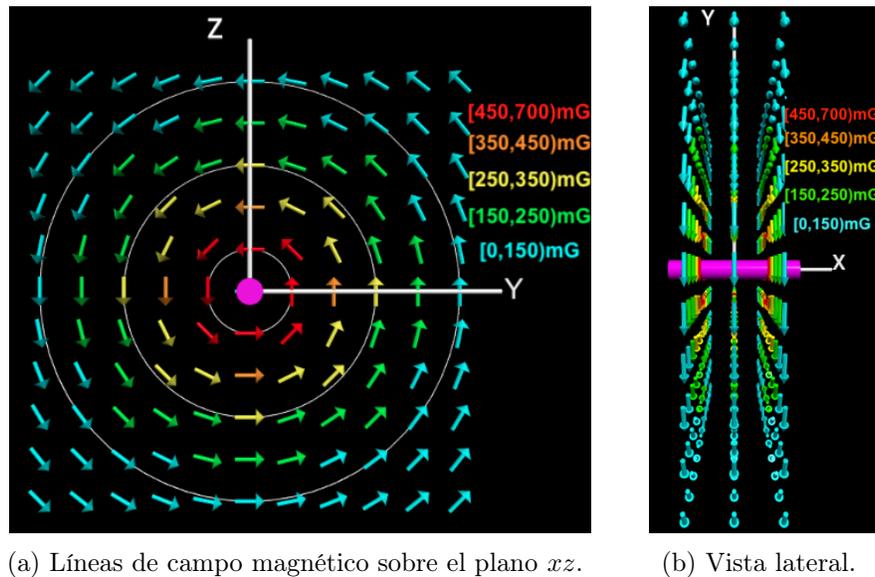


Figura 1.5: Líneas de campo magnético producidas por un alambre recto con corriente.

1.3. Campo magnético de una espira en cualquier punto del espacio

Para calcular el campo magnético producido por una espira circular situada en el origen en un punto fuera del eje de la espira (figura 1.6) en coordenadas cartesianas, consideramos las ecuaciones polares de la referencia [4]:

$$B_\rho = \frac{\mu I}{2\pi} \frac{z}{\rho[(a+\rho)^2+z^2]^{1/2}} \left[-K + \frac{a^2+\rho^2+z^2}{(a-\rho)^2+z^2} E \right] \quad (1.8)$$

$$B_z = \frac{\mu I}{2\pi} \frac{1}{[(a+\rho)^2+z^2]^{1/2}} \left[K + \frac{a^2-\rho^2-z^2}{(a-\rho)^2+z^2} E \right] \quad (1.9)$$

Renombramos algunas variables, teniendo en cuenta que:

$$B_x = |B| \cos \theta = |B| \frac{x}{\rho} \quad (1.10)$$

$$B_z = |B| \sin \theta = |B| \frac{z}{\rho} \quad (1.11)$$

donde θ es el ángulo que forma el vector de posición del punto donde se mide el campo con el plano x, z . Ya que el problema tiene geometría axial, tenemos que de la Ref. [1]:

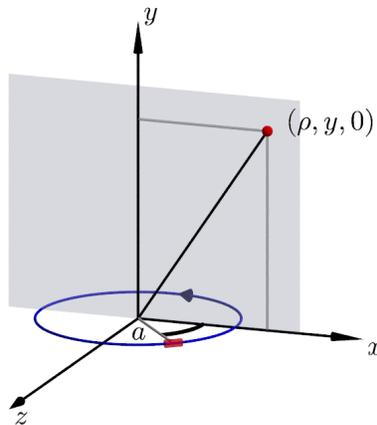


Figura 1.6: Esquema para el cálculo del campo magnético de un espira de corriente.

$$B_x = \frac{x}{\rho} I \frac{\gamma}{2\pi a \sqrt{Q}} \left[E(k) \frac{1 + \alpha^2 + \beta^2}{Q - 4\alpha} - K(k) \right] \quad (1.12)$$

$$B_y = I \frac{1}{2\pi a \sqrt{Q}} \left[E(k) \frac{1 - \alpha^2 - \beta^2}{Q - 4\alpha} + K(k) \right] \quad (1.13)$$

$$B_z = \frac{z}{\rho} I \frac{\gamma}{2\pi a \sqrt{Q}} \left[E(k) \frac{1 + \alpha^2 + \beta^2}{Q - 4\alpha} - K(k) \right] \quad (1.14)$$

con:

$$\rho = \sqrt{x^2 + z^2}, \quad \alpha = \frac{\rho}{a}, \quad \beta = \frac{y}{a}, \quad \gamma = \frac{y}{\rho}, \quad Q = (1 + \alpha)^2 + \beta^2, \quad k = \frac{4\alpha}{Q}, \quad (1.15)$$

donde a es el radio de la espira, I la corriente que circula por la espira, $K(k)$ es la integral elíptica de primera especie y $E(k)$ es la integral elíptica de segunda especie.

En la figura 1.7 se muestra el campo vectorial resultado de evaluar las ecuaciones 1.12, 1.13 y 1.14 en un mallado sobre un plano paralelo a su eje.

Si consideramos medir el campo magnético de la espira sobre su eje obtenemos que las componentes $x = 0$ y $z = 0$ por lo que las ecuaciones anteriores se reducen a:

$$B_y = \frac{\mu_0 I a^2}{2(a^2 + y^2)^{3/2}} \quad (1.16)$$

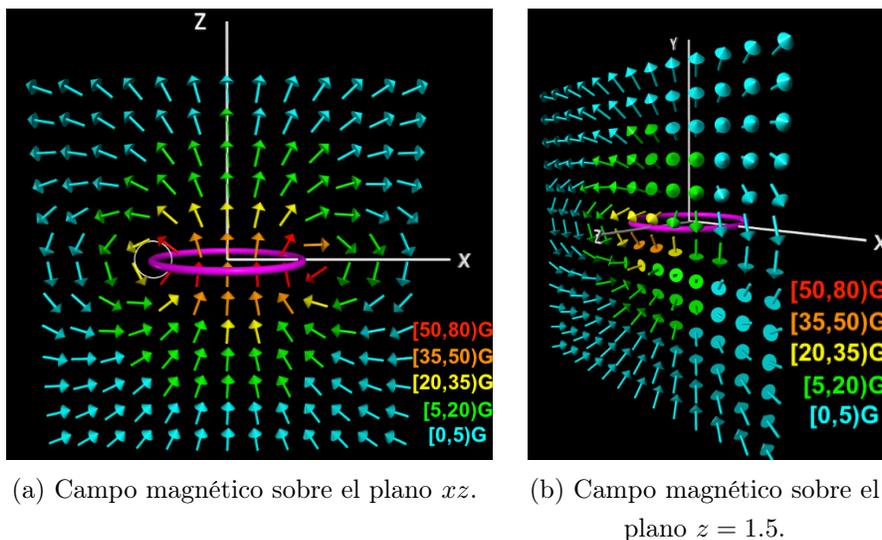


Figura 1.7: Campo magnético de una espira, visto sobre distintos planos.

1.4. Campo magnético de una bobina.

Una bobina es el resultado de enrollar un alambre sobre un eje de tal manera que tenga un número N de vueltas y en consecuencia genere un campo magnético cuando por esta circule una corriente I .

Empleando el principio de superposición y tomando en cuenta que una bobina se puede considerar como la adición de varias espiras [5], tenemos entonces que el campo magnético de una bobina en cualquier punto del espacio se puede calcular como la suma total del campo magnético de cada espira. Es decir si la bobina tiene N vueltas, entonces el campo magnético de la bobina es:

$$\vec{B}_{\text{bobina}} = \vec{B}_{\text{espira}_1} + \vec{B}_{\text{espira}_2} + \cdots + \vec{B}_{\text{espira}_N} \quad (1.17)$$

Considerando las ecuaciones 1.12, 1.13, 1.14 y 1.17 obtenemos que el campo magnético de una bobina ubicada en el origen sobre un plano paralelo a su eje tiene la forma de las figuras 1.8.

Observamos que las líneas de campo magnético entrarían por un extremo de la bobina y saldrían por el extremo opuesto, lo que se asemeja al campo magnético de un imán de barra como el de la figura 1.2. Además, la magnitud del campo en el interior de la bobina es aproximadamente constante y menos intenso que en el exterior [6].

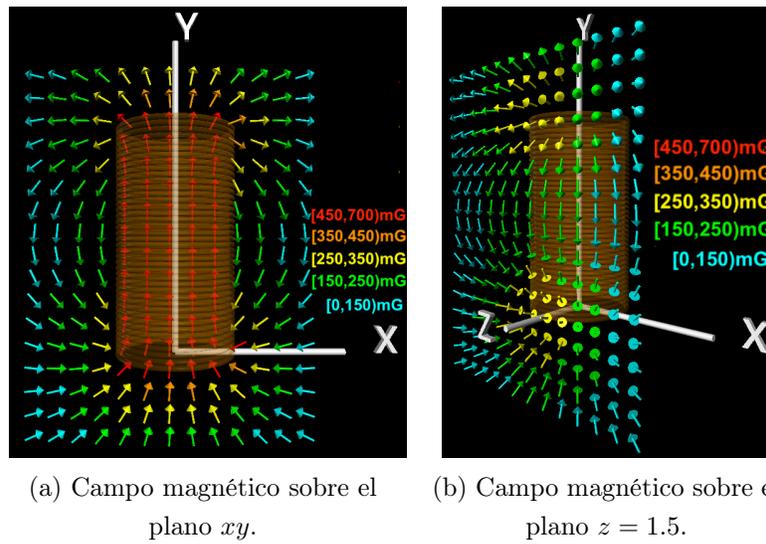


Figura 1.8: Campo magnético de una bobina de radio 1 calculado sobre distintos planos.

1.5. Planteamiento del Dispositivo

En la sección 1.2.1, mencionamos que podemos medir un campo magnético en el espacio generado por un objeto con propiedades magnéticas o por cargas en movimiento. Por tal motivo se necesita un dispositivo que cumpla con las siguientes características:

- ◇ Incorpore un sensor de campo magnético en 3 ejes ($\vec{B} = B_x, B_y, B_z$)
- ◇ Sea capaz de mover el sensor alrededor de la muestra en los 3 ejes.
- ◇ El movimiento debe ser preciso y repetible.
- ◇ Que pueda almacenar los datos obtenidos en la medición.
- ◇ Interprete y permita la visualización de los datos almacenados.
- ◇ Evitar el uso de materiales ferromagnéticos.

Para cubrir las características antes mencionadas se realizó una investigación de elementos que fueran capaces de satisfacerlas, y en las siguientes secciones se describen los elementos empleados para construir el dispositivo mapeador de campo magnético.

1.6. Control Numérico Computarizado.

El término *Control Numérico Computarizado* (CNC) se refiere a los instrumentos o maquinaria que emplean una computadora para controlar los movimientos de herramientas de corte, como brocas o buriles para maquilar objetos.

Estos instrumentos se desarrollaron inicialmente en el Instituto Tecnológico de Massachusetts (MIT) en la década de 1950 [20], en donde se construyó una fresadora de control numérico.

Actualmente, y con el avance de la tecnología, las computadoras son cada vez más accesibles, de modo que ha sido posible automatizar el funcionamiento de diversos tipos de aparatos, como las impresoras 3-D (figura 1.9) las cuales usan motores a pasos para controlar el movimiento de un extrusor (calefactor que funde un filamento plástico para formar un sólido), o en los *routers*, donde se controla el movimiento de una broca para desbastar madera o algún otro material (figura 1.10).

Estas máquinas que incorporan control numérico aumentan enormemente la calidad de los productos que generan respecto a las que son controladas manualmente, así como la eficiencia en la producción, de modo que en la actualidad son parte fundamental de la industria.

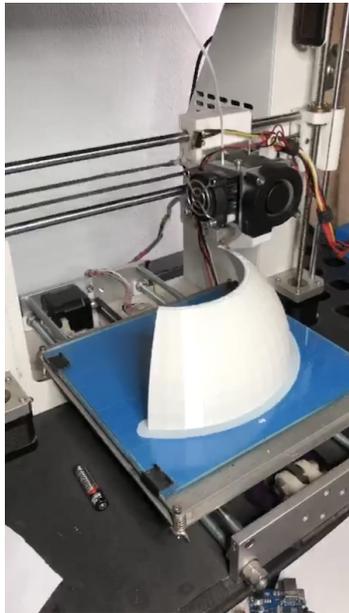


Figura 1.9: Impresora 3D



Figura 1.10: Router CNC

Algunas de las funciones específicas que pueden programarse en una CNC son las siguientes:

- * **Control de movimiento.** Las máquinas CNC tienen dos o más ejes de movimiento, ya sea lineales o rotativos (como en el caso de los tornos). El movimiento de cada uno de estos ejes es controlado por el procesador, el cual recibe a su vez información de los sensores que registran la posición del mismo en un o varios puntos.

- * **Programa CNC.** El programa de la máquina CNC consiste en un listado de comandos que representan una acción en la máquina. Un tipo de código muy usado en la actualidad es el llamado *G-code*, el cual es empleado por las impresoras 3D.
- * **Controlador CNC.** Este controlador incorpora un procesador que interpreta el programa CNC y genera las señales necesarias para accionar los motores de los ejes, asimismo recibe la información de los sensores. De este modo, en el controlador se implementa tanto el Control de movimiento como el Programa CNC.

En este proyecto construiremos en su totalidad una máquina tipo CNC para tomar medidas del campo magnético en posiciones precisas en el entorno de una muestra (la fuente de campo magnético a caracterizar). Esta máquina consta de tres ejes mutuamente perpendiculares a lo largo de los cuales se mueve un sensor que entrega la magnitud de las componentes x , y y z del campo magnético. El movimiento de los ejes se efectúa mediante motores de pasos y tienen un switch como sensor de posición en sus extremos, de este modo es posible asignar una posición precisa al punto donde se mide \vec{B} .

1.7. Arduino

A continuación daremos una breve explicación de algunos de los términos más importantes relacionados con el desarrollo de dispositivos empleando las tarjetas Arduino.

Arduino. Es una plataforma de desarrollo de código abierto, la cual está basada en **hardware y software gratuito**, flexible y fácil de utilizar. Esta plataforma permite crear diferentes tipos de controladores de una sola tarjeta para diferentes tipos de uso que incorpora la electrónica necesaria para controlar y/o leer señales lógicas[15].

Hardware libre. Son los dispositivos construidos de acuerdo a especificaciones y diagramas de acceso público, de manera que cualquier fabricante puede replicarlos legalmente. De este modo, Arduino ofrece la información necesaria para que cualquier persona o empresa pueda maquilar sus propias tarjetas, incorporando diferentes características, pero manteniendo la compatibilidad con la plataforma.

Software gratuito. Son los programas cuyo código es cedido al público para ser usado o modificado. Arduino tiene su plataforma **IDE** (Entorno de Desarrollo Integrado), que

es un entorno de programación en C para el desarrollo de aplicaciones en las tarjetas Arduino, las cuales definen su funcionalidad.

Por lo anterior trabajar con la plataforma Arduino resulta ser **accesible** y de **bajo costo**, de modo que la elección de esta plataforma para nuestro trabajo resulta obvia.

Arduino Nano. Es una pequeña y completa placa basada en el microcontrolador ATmega328 (figura 1.11).

Tiene 14 entradas digitales, (6 de ellas proveen, además, salidas moduladas por ancho de pulsos (PWM) que operan a 5 Volts), 8 entradas analógicas que operan con una corriente máxima de 40 mA, memoria Flash de 32 kB, SRAM de 2 kB y EEPROM de 1kB. Se alimenta con un cable micro USB conectado a la PC o con 5 volts de una fuente externa. Las dimensiones de esta placa son: 4.5 cm x 1.8 cm.

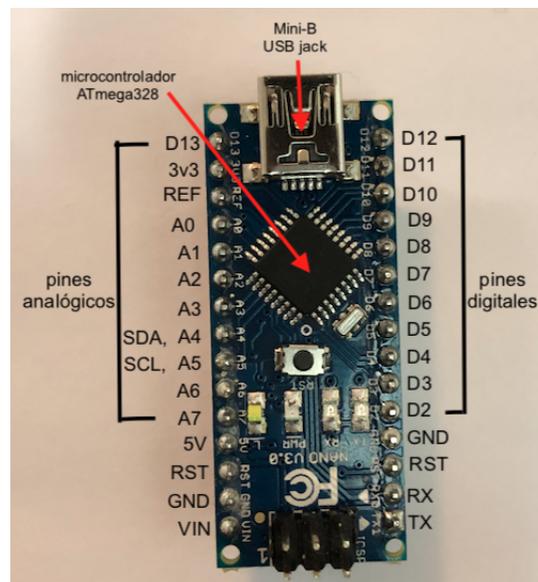


Figura 1.11: Arduino Nano, fotografía de la placa arduino empleada en este proyecto.

Programación de Arduino. Programar una tarjeta Arduino consiste en incluir las líneas de código necesarias para realizar las tareas que queremos llevar a cabo. Dentro del programa es posible leer los valores entregados por los sensores y, en función de estas condiciones, asignar valores lógicos a las salidas, lo que permite la interacción con los actuadores.

Arduino cuenta con un entorno de programación en *C*, junto con las utilerías que permiten compilar el programa y cargar el resultado de la compilación en la memoria de la tarjeta. Este entorno de programación permite incluir librerías específicas para los periféricos y sensores que existen en el mercado, compatibles con las diferentes variantes de las tarjetas Arduino.

Un programa de Arduino se denomina *sketch* y tiene la extensión **.ino**, el cual se crea en una carpeta con el mismo nombre que el *sketch*. Un *sketch* puede requerir de librerías, de modo que éstas deben instalarse utilizando las herramientas propias del del entorno.

La **estructura** de un *sketch* de Arduino consta de dos partes; la función `setup()` y la función `loop()`; la definición de cada una de ellas puede contener declaraciones de variables locales, ciclos e instrucciones. Adicional e independientemente, se puede incluir un encabezado con los comentarios que describen el programa y la declaración de las variables globales y las llamadas a librerías.

`setup()`. Esta se usa para configurar los sensores y la inicialización de los elementos, inicializar variables, etc. Ésta función sólo se ejecuta una vez al iniciar la tarjeta o después de presionar el botón de *reset*, también puede ser llamada desde la función `loop()` para reiniciar el sistema. Asimismo, es la primera función que se ejecuta al correr el programa

`loop()`. Es la función definida por el programa que se ejecuta cíclicamente, para la lectura de las entradas, la asignación de valores a las salidas, etc. Esta función es el control activo de la tarjeta. El primer ciclo de la función `loop()` se ejecuta justo después de haberse ejecutado `setup()`.

Algunos de los comandos que se emplearon en este proyecto se presentan en la Tabla 1.1.

Comando	Descripción
<code>pinMode(<i>pin</i>,<i>modo</i>)</code>	Esta instrucción es utilizada en la parte de configuración <code>setup()</code> y sirve para configurar el modo de trabajo del pin especificado, los valores pueden ser INPUT (entrada) u OUTPUT (salida).
<code>Serial.begin(<i>rate</i>)</code>	Activa el puerto serie y selecciona la velocidad de transmisión de datos en baudios (<i>rate</i>) a través del puerto USB (en serie). El valor típico es de 9600 baudios.
<code>digitalRead(<i>pin</i>)</code>	Lee el estado del <i>pin</i> digital especificado, ya sea HIGH (ALTO ó uno) ó LOW (BAJO o cero).
<code>analogRead(<i>pin</i>)</code>	Lee el valor en mV de la entrada analógica asociada al <i>pin</i> de entrada.
<code>Serial.print(<i>val</i>)</code>	Envía al puerto serie el mensaje especificado por <i>val</i> .
<code>delay(<i>t</i>)</code>	Detiene la ejecución del programa por la cantidad de tiempo (en milisegundos) especificada por <i>t</i> .

Cuadro 1.1: Comandos empleados en este proyecto.

1.8. Sensor de Campo Magnético

Un sensor es un dispositivo para medir o detectar alguna magnitud física o química por medio de un transductor, que recibe algún tipo de energía y genera una señal eléctrica para ser procesada.

En el caso de los sensores de campo magnético, el transductor funciona mediante el Efecto Hall (Apéndice A.1). Son dispositivos de estado sólido, y pueden usarse en combinación con otros circuitos analógicos y digitales. El sensor magnético que se empleó en este trabajo es el sensor HMC5883L, el cual se describe a continuación.

Sensor HMC5883L. Es un magnetómetro de 3 ejes (figura 1.12), cuenta con tres sensores de campo magnético (uno para cada eje X , Y y Z respectivamente) cada uno con un convertidor analógico-digital de 12 bits, de modo que tiene precisión de ± 2.175 miligauss. El voltaje de alimentación es de 1.8 a 3.3 V, sin embargo cuenta con un regulador para ser alimentado con 5 V o con 3.3 V [17]. Este sensor se comunica a través de un puerto



Figura 1.12: Fotografía del sensor HMC5883L empleado en este proyecto.

con protocolo I2C, por lo que es posible conectar varios dispositivos a una sola entrada de la tarjeta Arduino.

Este sensor tiene un rango de medición de ± 8 G, una resolución de 5 mG y un periodo de medición de 6 ms (desde que se recibe el comando hasta tener los datos listos). La placa en donde está montado el sensor tiene las dimensiones: 1.4 cm \times 1.5 cm \times 0.3 cm.

1.9. Motores paso a paso

El motor paso a paso (*stepper motor*) conocido también como motor de pasos, es un tipo de motor eléctrico cuyo rotor está acoplado con un imán permanente que se alinea secuencialmente con el campo magnético generado por bobinas dispuestas en una circunferencia, con lo cual se logran desplazamientos angulares discretos, de este modo, este motor es capaz de girar un cierto ángulo en función de sus entradas de control, figura 1.13.

Estos motores están conformados por dos partes: el estator y el rotor. El estator es la parte que permanece fija, y es donde el arreglo de bobinas es colocado simétricamente por pares. Por otro lado, el rotor es la parte que está montada en el eje y está acoplado a uno o más imanes permanentes colocados simétricamente en pares alternando su orientación.

Al aplicar corriente a una bobina, el rotor se alineará con el campo magnético producido, girando un cierto ángulo, al completar esta rotación, se corta la corriente y se aplica a la siguiente bobina, provocando un nuevo giro. Este proceso se realiza en secuencia en cada una de las bobinas para producir un número de giros específico. A menudo también se emplea la repulsión de los polos de los imanes del rotor, de esta manera algunas bobinas lo atraen y lo empujan, consiguiendo así una rotación en un ángulo intermedio.

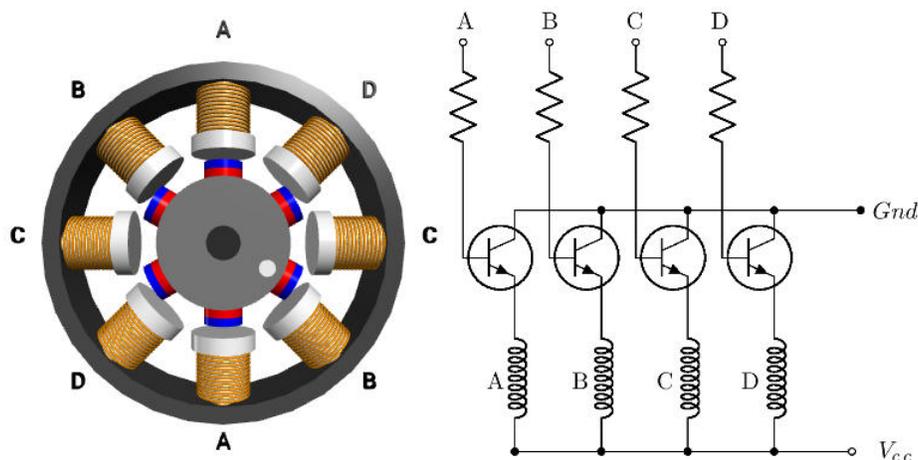


Figura 1.13: Esquema de la estructura interna de un motor paso a paso.

Así, es posible determinar de forma precisa la posición del rotor, al ser ésta un múltiplo del ángulo que existe entre una bobina y la siguiente (también es posible conseguir medios o cuartos de paso, con la electrónica adecuada). Este paso puede variar desde 90° hasta pequeños movimientos de 1.8° [13].

Estos motores poseen la habilidad de quedar fijos en una posición si una o más de sus bobinas está energizada o bien totalmente libres al dejar de aplicarles corriente.

Estos motores pueden ser controlados por señales digitales provenientes de un microcontrolador, como en nuestro caso, donde se requiere una buena precisión y alta repetibilidad en cuanto al posicionamiento, como en la mayoría de las máquinas CNC.

1.10. Driver DRV8825

Es un controlador (driver) que simplifica el manejo de motores paso a paso desde un procesador como Arduino, en la figura 1.14 se muestra el controlador con el nombre de sus respectivos pines.

Estos controladores permiten manejar los altos voltajes y corrientes que requieren estos motores, limitar la corriente que circula por el motor, y proporciona las protecciones para evitar que la electrónica pueda resultar dañada.

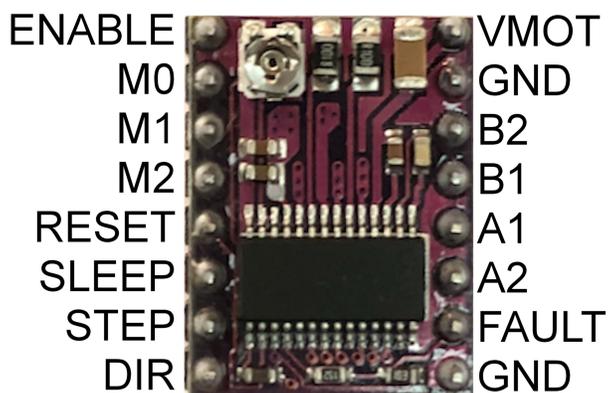


Figura 1.14: Fotografía del *driver* DRV8825 empleado en este proyecto.

El DRV8825 es un circuito que simplifica el control de motores paso a paso, consta, entre otras cosas, de un par de puentes H y controlador de corriente. Los puentes H sirven para energizar cada una de las bobinas del motor de manera independiente y en ambas polaridades, es decir, dan la secuencia adecuada de configuraciones de bobinas energizadas en el motor (Sección 1.9), por medio de un arreglo de transistores de potencia, para que el motor dé los pasos o fracciones de éstos en función de los pulsos de baja corriente en su pin *step* y del estado del pin de dirección (*dir*).

Por lo anterior, para su control únicamente se requieren dos salidas digitales, una para indicar el sentido de giro y otra para comunicar que queremos que el motor avance un paso. Además permite realizar *microstepping*, una técnica para conseguir precisiones superiores al paso nominal del motor al posicionar el rotor entre dos bobinas [14].

El limitador de corriente permite usar este driver para controlar motores de menor voltaje que el mínimo de la placa (8.2 V). Es decir, se puede usar una fuente de 10 V para manejar un motor paso a paso de 5 V.

1.11. Objetivos

1.11.1. Objetivo General

El presente trabajo describe la construcción de un dispositivo para la visualización del campo magnético de una fuente, como por ejemplo el de un alambre recto, de una bobina y de un imán permanente, de forma automatizada, de manera precisa y reproducible.

1.11.2. Objetivos Particulares

1. Diseñar y construir un dispositivo o estructura que nos permita el libre movimiento en las tres direcciones (x , y y z) alrededor de una fuente de campo magnético.
2. Crear un programa que sea capaz de mover y controlar el dispositivo antes mencionado y al sensor magnético para realizar mediciones y su respectivo registro.
3. Diseñar una interfaz gráfica para controlar el dispositivo y realizar el análisis de los datos obtenidos por el sensor.
4. Comparar los datos obtenidos contra los datos teóricos.

1.12. Hipótesis

Se puede construir un dispositivo mecánico de bajo costo y complementario a la forma práctica de visualización de campos magnéticos (con limadura de hierro) haciéndolo preciso y reproducible para un mejor análisis y entendimiento del concepto de campo magnético y así aprovechar sus características.

Capítulo 2

Desarrollo experimental

En este capítulo se describe el proceso de construcción del dispositivo mapeador de campo magnético, el cuál se dividió en las siguientes etapas:

- * Diseño y ensamblado.
- * Control.
- * Automatización.

2.1. Diseño de la Estructura del Dispositivo

Como se mencionó en la sección 1.5, el requisito para la estructura del dispositivo es que se pueda mover libremente en los ejes x , y y z , alrededor de una fuente de la cual se requieren realizar mediciones, como se muestra en la figura 2.4

Para lograr esto, se realizó un sistema similar al de una máquina CNC, pues en estas máquinas se puede controlar la posición y velocidad de los motores que accionan sus ejes, proporcionando así la precisión que se requiere.

Tras una revisión preliminar de los diseños disponibles, la máquina CNC que se tomó como base para nuestro diseño se muestra en la figura 2.2, pues su estructura es sencilla y ligera.

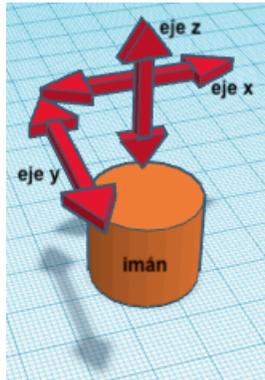


Figura 2.1: Movimientos requeridos para el dispositivo.



Figura 2.2: Máquina CNC de dos ejes, imagen obtenida de internet [18].

Esta máquina consiste de dos ejes perpendiculares uno fijo y uno móvil (figura 2.3), cuenta con una caja en la cual atraviesan y se sostienen las varillas que conforman estos ejes. La caja tiene cuatro poleas para la banda que rodea el dispositivo.

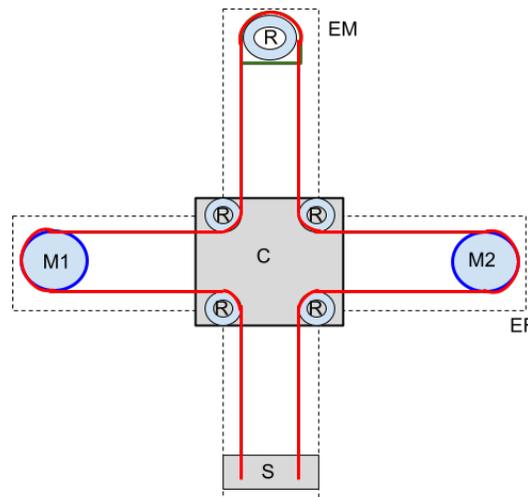
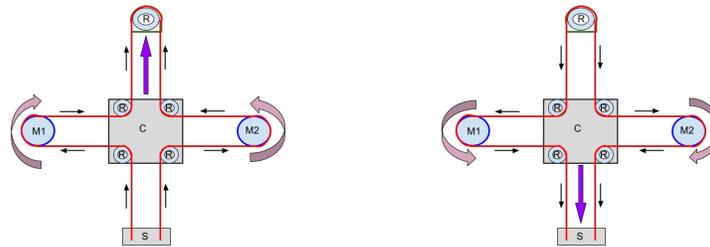


Figura 2.3: Diagrama estructural de la máquina CNC. EF=eje fijo; EM=eje móvil; C=caja soporte; R=poleas; M1, M2=motores

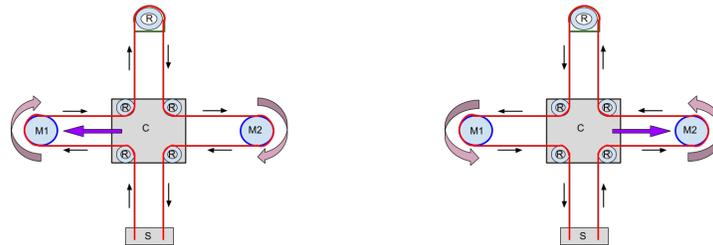
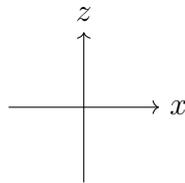
El eje fijo tiene un motor a pasos en cada extremo y, para generar el movimiento del eje móvil en las direcciones x , z , los motores deben girar de la siguiente forma:

- M1 en sentido horario y M2 en sentido antihorario producen movimiento en dirección negativa del eje z , Fig. 2.4a.
- M1 en sentido antihorario y M2 en sentido horario producen movimiento en dirección del eje z , Fig. 2.4b.
- M1 y M2 en sentido horario producen movimiento en dirección negativa del eje x , figura 2.4c.
- M1 y M2 en sentido antihorario producen movimiento en dirección del eje x , Fig. 2.4d.



(a) Movimiento en dirección $-z$.

(b) Movimiento en dirección z .



(c) Movimiento en dirección $-x$.

(d) Movimiento en dirección x .

Figura 2.4: Direcciones obtenidas por la combinación del sentido de giro de los motores.

Estos movimientos se deben a que la banda que rodea al dispositivo (línea roja figura 2.3) se encuentra fija en un extremo (S) del eje móvil (EM) y en su otro extremo pasa por una polea (R), y ya que la banda es inextensible, todo el movimiento de los motores se transmite a la banda y dependiendo del sentido de giro de los motores, y de la libertad de movimiento en las poleas y el soporte del sensor S se producirán los distintos movimientos antes mencionados.

De las imágenes de la figura 2.4 podemos observar que la caja sólo se mueve a lo largo del eje x , de forma que el extremo del eje móvil es el que tomará posiciones en puntos sobre el plano xz .

Éste dispositivo sólo nos permite movimientos en el plano xz , así que para tener movimientos sobre el eje y se diseñó otro eje, el cual consiste de un motor que sube y baja un portamuestras mediante una banda, como se muestra en la figura 2.5.

Debido a que los materiales ferromagnéticos modifican el campo magnético, se decidió utilizar material plástico para construir las piezas que conforman la estructura del dispositivo, por esa misma razón la máquina CNC que necesitamos tenía que ser ligera.

El diseño de las piezas se realizó con **Tinkercad**, la cual es una plataforma gratuita de modelado 3D en línea, y se imprimieron con una impresora 3D. En la figura 2.6 se muestran algunas de estas piezas.

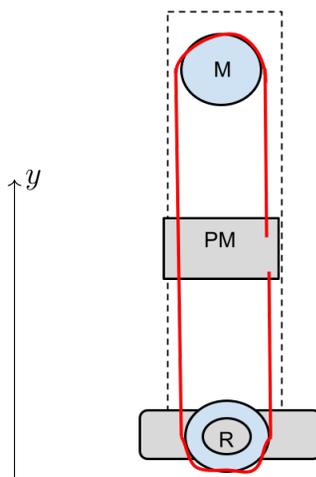


Figura 2.5: Diagrama estructural del eje y . M=motor, PM=portamuestras, R=punto de apoyo.

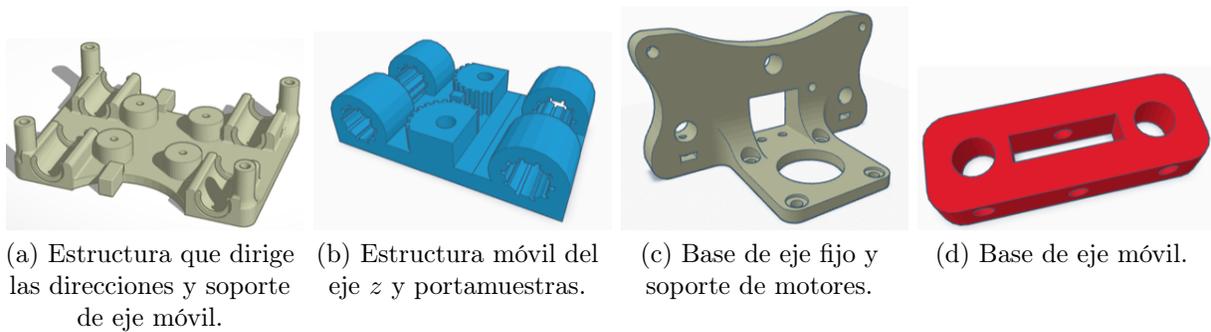


Figura 2.6: Diseño de piezas para la estructura del dispositivo realizados con Tinkercad.

Para ensamblar los ejes del plano xz se emplearon varillas de acrílico. La estructura que dirige las direcciones (figura 2.6a), las bases de eje móvil (figura 2.6c) y las bases de eje móvil (figura 2.6d); los motores se colocaron en los extremos del eje fijo (EF) y en las zonas donde son necesarios los puntos de apoyo y giro (R) se colocaron poleas para obtener un mejor deslizamiento (figuras 2.5, 2.3), el sensor (S) se coloca en el extremo del eje móvil (EM) donde se sujetan las puntas de la banda, esta estructura se muestra en la figura 2.7.

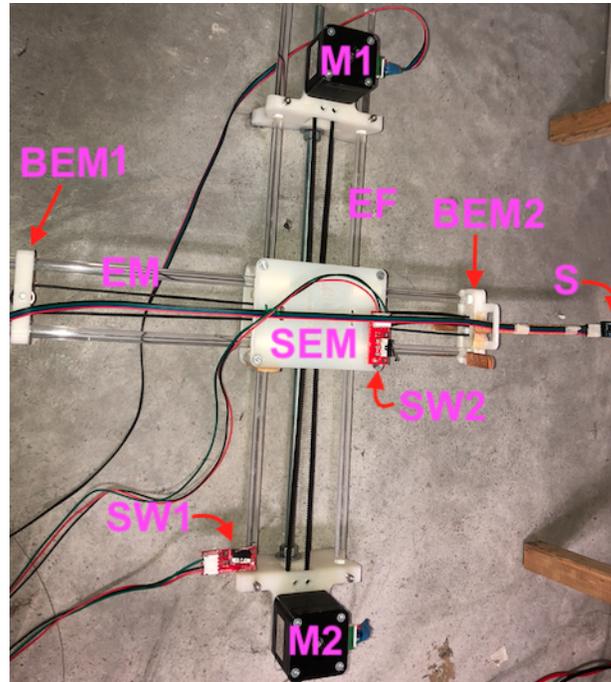


Figura 2.7: Estructura que permite al sensor moverse sobre el plano xz . EM: Eje móvil; SEM: Soporte de eje móvil; EF: Eje fijo; M1, M2: Motores Stepper; SW1, SW2: Switch de paro; BEM1, BEM2: Bases de eje móvil; S: Sensor magnético HMC5883L

Las dimensiones de esta estructura es de 46 cm x 45 cm, en donde el eje móvil realiza desplazamientos de 30 cm a lo largo el eje x y de 31 cm a la largo del eje z .

Para ensamblar el eje y se emplearon varillas de acrílico, una base de eje fijo (figura 2.6c), la base de eje móvil (figura 2.6d) y la estructura móvil del eje y (figura 2.6b). Esta estructura se muestra en la figura(2.8).

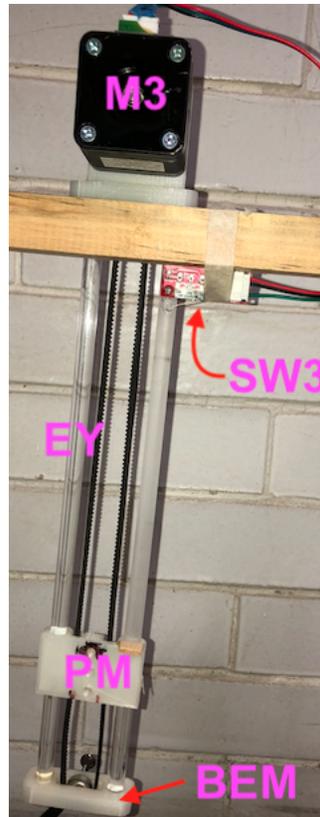
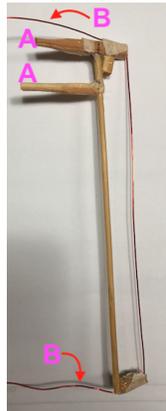


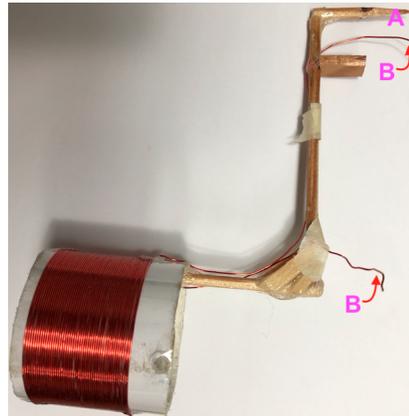
Figura 2.8: Estructura que permite al portamuestras moverse verticalmente (a lo largo del eje y). M3: Motor Stepper; SW3: Switch de paro; EY: Eje vertical; PM: Porta muestras; BEM: Base de eje móvil.

Esta estructura tiene una longitud de 35 cm y el portamuestras tiene desplazamientos de 24 cm a lo largo del eje vertical.

Las muestras a medir se colocaron sobre una estructura de madera rígida diseñada para encajar en el portamuestras, la figura 2.9 presenta un alambre y una bobina sobre la estructura antes mencionada.



(a) Alambre empleado en este proyecto.



(b) Bobina empleada en este proyecto.

Figura 2.9: Muestras colocadas sobre una estructura de madera. A: secciones que embonan en el portamuestras; B: contactos para la fuente de alimentación.

Finalmente la estructura del eje vertical se coloca sobre una estructura elevada de madera, la muestra a medir se coloca en el portamuestras y la estructura que mueve el sensor se alinea a la muestra; la figura 2.10 muestra el dispositivo ensamblado.

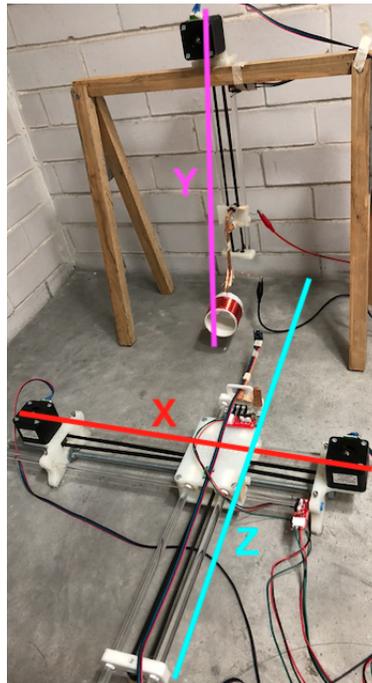


Figura 2.10: Dispositivo completo ensamblado.

Cabe mencionar que con esta estructura el sensor se mueve sobre el plano xz , y el portamuestras se mueve verticalmente. Con la combinación de estos movimientos se puede obtener una posición específica en el espacio alrededor de la fuente de campo magnético.

Las dimensiones del dispositivo final alineado son: 45 cm en el eje X , 74 cm en el eje Z y 50 cm del eje Y .

2.2. Circuitos

En este proyecto se empleó la placa *Arduino Nano* (Sección 1.7) como controlador de todo el dispositivo, el sensor magnético HMC5883l (Sección 1.8) para obtener datos sobre el campo magnético a medir, drivers drv8825 (Sección 1.10) para controlar los motores del dispositivo, motores bipolares Nema-17 para generar el movimiento del eje móvil del dispositivo e interruptores para limitar el recorrido del eje móvil. A continuación se muestran las respectivas conexiones entre esta placa con los distintos elementos.

Conexión de Arduino Nano, motores stepper y drv8825

Como se mencionó en la sección 1.10 el driver drv8825 utiliza dos salidas digitales del *Arduino Nano* para el control de un motor, además de una alimentación del Arduino de 5 V y una externa de 12 V. En la figura 2.11 se muestran estas conexiones y las correspondientes conexiones con los motores.

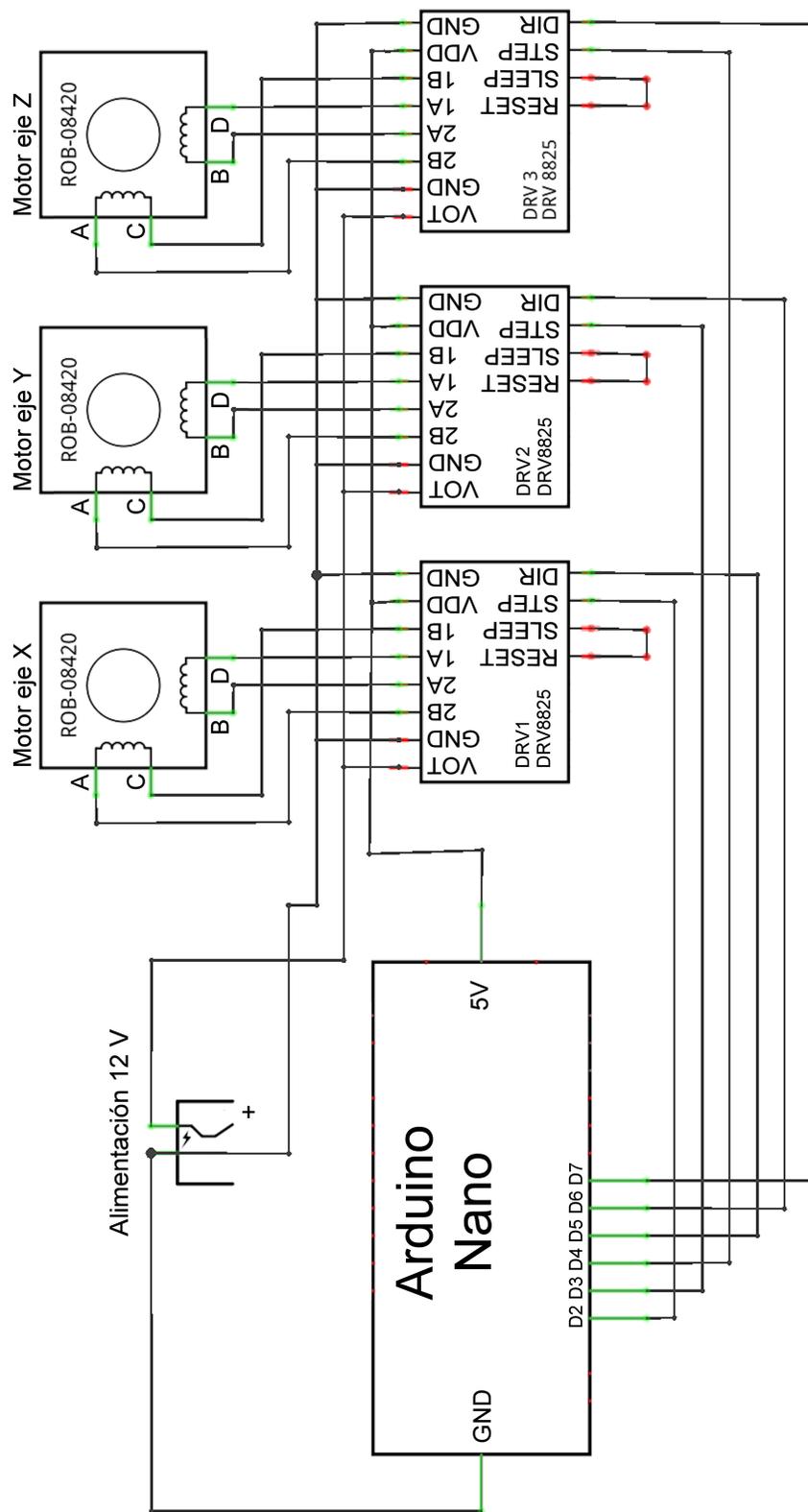


Figura 2.11: Esquema de conexiones del *Arduino Nano*, los drivers *drv8825* y los motores *Nema-17*.

Para definir los pines que serán las salidas del *Arduino Nano* para el control de un motor, se declaran en el *sketch* de la siguiente manera:

```
1  const int dirPin = 5; // PIN DE DIRECCION
2  const int stepPin= 2; // PIN DE PASOS
3
4  void setup(){
5      pinMode(dirPin, OUTPUT); // DEFINIMOS LOS PINES COMO
        SALIDA
6      pinMode(stepPin, OUTPUT);}
```

Conexión de Arduino Nano, sensor de campo magnético HMC5883L y botones

El elemento principal en este proyecto es el sensor de campo magnético HMC5883L, el cual necesita sólo dos conexiones y su respectiva alimentación del *Arduino Nano*.

Para tener un control de movimiento del dispositivo se agregaron botones normalmente abiertos, cada uno asignado a una entrada digital. En la figura 2.12 se muestran las conexiones antes mencionadas.

El código para activar el sensor HMC5883L y un botón es el siguiente:

```
1  #include <Wire.h> //LIBRERIA PARA COMUNICACION I2C
2  #include <HMC5883L.h> //LIBRERIA PARA EL HMC5883L
3  HMC5883L compass;
4
5  const int buttonPin0 = 8; //PIN PARA EL BOTON
6
7  void setup(){
8      while(!compass.begin()){delay(500);} //INICIALIZACION DEL
        HMC5883L
9      compass.setRange(HMC5883L_RANGE_1_3GA); //CALIBRACION DEL
        HMC5883L
10     compass.setMeasurementMode(HMC5883L_CONTINUOUS);
11     compass.setDataRate(HMC5883L_DATARATE_15HZ);
12     compass.setSamples(HMC5883L_SAMPLES_8);
```

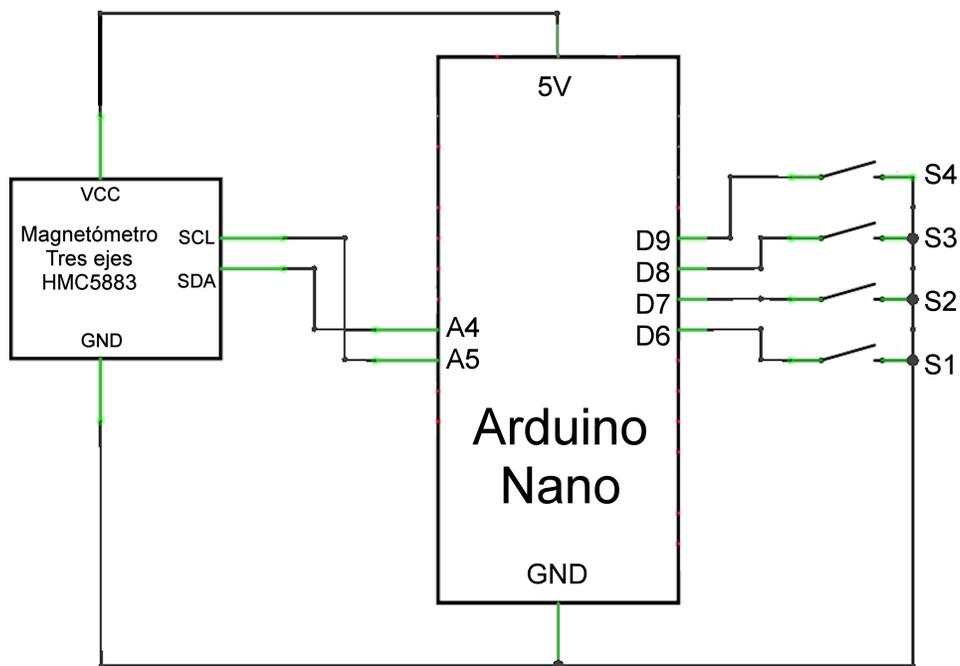


Figura 2.12: Esquema de conexiones del Arduino Nano, el sensor HMC5883L y los interruptores.

```

13
14 pinMode(buttonPin0 , INPUT_PULLUP);           //MODO DE PIN
15 }

```

Los interruptores para limitar el recorrido se asignan a entradas digitales, en particular los interruptores que se emplearon cuentan con tres pines, y conmutan $+V$ o tierra (GND) con un contacto central, de modo que al conectar éste a la entrada digital del Arduino, se mantendrá la señal en bajo a menos que el interruptor haga contacto.

2.3. Control

Una vez que se ensambló el dispositivo y se realizaron las conexiones correspondientes, se diseñó el programa para controlar los motores y con ello mover los ejes para posicionar el sensor de campo magnético alrededor de la muestra.

Como se mencionó en la sección 1.9, el motor de pasos es un dispositivo electromecánico que convierte impulsos eléctricos en desplazamientos angulares discretos, de modo que

para enviar estos impulsos del *Arduino Nano* al motor se implementó el siguiente código:

```

1  int delayTime=1200;          //TIEMPO ENTRE CADA PULSO
2
3  void loop(){
4  digitalWrite(dirPin,LOW);    //SENTIDO DE MOVIMIENTO DEL MOTOR
5  digitalWrite(stepPin,HIGH);
6  delayMicroseconds(delayTime);
7  digitalWrite(stepPin,LOW);
8  delayMicroseconds(delayTime);
9  }

```

Nótese que el pulso se crea al suministrar y quitar voltaje con su respectivo tiempo de ejecución a la salida de paso del motor (*stepPin*), ya que éste código se encuentra en la función **loop()** se repetirá una y otra vez hasta que se desconecte la placa Arduino o la fuente de alimentación externa.

Para cambiar el sentido de giro del motor, se coloca **HIGH** en la línea de escritura de **dirPin** (línea 4).

Se propone que los movimientos del eje móvil sean rectos a lo largo del eje *x* y del eje *y*, formando una malla, de modo que, en cada vértice, el sensor mida el campo magnético. Para lograr estos movimientos se emplean combinaciones del código anterior y combinaciones de giro de motor mostrados en la figura 2.4.

Ya que el sensor HMC5883L cuenta con tres sensores mutuamente perpendiculares, éste nos permite obtener las componentes del vector de campo magnético en coordenadas cartesianas. Para ello introducimos las siguientes instrucciones en la función **loop()** del código.

```

1  Vector norm=compass.readNormalize();          //TOMA LECTURA
2  Serial.print(norm.XAxis);Serial.print(",")    //IMPRIME EN EL
   PUERTO SERIAL
3  Serial.print(norm.YAxis);Serial.print(",")
4  Serial.println(norm.ZAxis);

```

Para controlar los motores de los movimientos del eje móvil, se incorporaron tres botones; cada botón con tres estados, el primer estado activa el movimiento en una dirección, el segundo estado invierte la dirección y el tercer estado desactiva el movimiento. El código para generar estos estados con cada pulsación de un botón y evitar rebotes o repeticiones es:

```
1  int buttonState0=0, contador0 =0, antes0=0;    //VARIABLES PARA
    CONTADOR
2
3  void loop(){
4      buttonState0=digitalRead(buttonPin0);    //LECTURA DE LA
    ENTRADA DIGITAL
5
6      if(antes0==0 && not buttonState0){
7          antes0=1;
8          contador0=(contador0+1) % 3;
9      }
10     if (buttonState0) antes0=0;
11 }
```

Finalmente todos los códigos anteriores se incorporaron en uno sólo para así lograr que el dispositivo se moviera en el mallado anteriormente mencionado, se controlara con los botones, tomara lecturas a través del puerto Serial y las mostrara en la pantalla. Apéndice C.

2.4. Automatización

Una vez que se logró el movimiento del dispositivo mediante botones físicos, se decidió sustituirlos por botones en una interfaz gráfica, uno para cada dirección del dispositivo, de modo que el movimiento en una dirección sea más precisa y no dependa del estado del botón. Para hacer esto se implementó un programa en Python, la cual mantiene comunicación con el puerto Serial del Arduino Nano, para así poder controlar los movimientos y rutinas del dispositivo en general, así como leer los datos del puerto Serial, almacenar los

datos en un archivo, analizar los datos obtenidos y, finalmente, mostrarlos gráficamente.

La automatización del dispositivo consistió en colocar botones los cuales comienzan una rutina en específico, como por ejemplo: el mallado horizontal sobre una fuente magnética. Para esto se emplearon los interruptores, que permiten tener un punto de referencia para el inicio de lecturas del sensor. El código para los interruptores es similar al de los botones normalmente abiertos.

La interfaz gráfica de usuario que incorpora todas las acciones antes mencionadas y otras más se muestran en la figura 2.13 y su respectivo código en Apéndice B.



Figura 2.13: GUI el dispositivo mapeador de CM.

Capítulo 3

Resultados

3.1. Funcionamiento

El dispositivo consta de dos programas esenciales para su funcionamiento, el primero es el código *Arduino* que contiene los algoritmos de control para el sensor de campo magnético, los sensores de posición y los motores del dispositivo (Apéndice C), el segundo es el código *Python* que genera la interfaz gráfica de usuario (Apéndice B), establece comunicación a través del puerto USB y genera la visualización de los datos obtenidos.

Para programar el procesador del dispositivo, se escribió el *sketch* con el código *Arduino* antes mencionado, el cual se compiló y cargó a la placa *Arduino Nano* a través de la plataforma oficial, de este modo, el programa queda almacenado en la memoria permanente de la placa y se ejecuta en cuanto se le suministra voltaje a la placa.

Posteriormente ejecutamos el código *Python* desde la computadora, el cual presenta la GUI (figura 2.13) y establece la comunicación entre ésta y el dispositivo. A través de la interfaz gráfica se puede cambiar la posición del sensor y la altura de la muestra con los botones que se encuentran en la sección del "MOVIMIENTOS DE MOTOR". Al presionar un botón de dirección comienza el movimiento, y se detiene al presionar el botón STOP. Estos mismos botones de dirección se pueden utilizar después de presionar el botón BRUJULA para observar la simulación de brújula (con tres grados de libertad) en cualquier punto del espacio alrededor de la muestra.

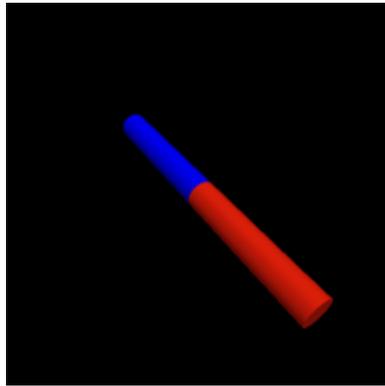


Figura 3.1: Simulación de la brújula.

Al seleccionar la opción “BARRIDO RAPIDO”, sólo se realiza la toma de datos en planos por debajo de la muestra, para esto, primero se selecciona el número de mediciones en el selector: “Dimensiones del mallado (x, y, z)” y la separación entre cada medición en el selector: “Tamaño del paso”; una vez que se seleccionaron estos campos se oprime el botón “Barrido” y una vez que el dispositivo termine su recorrido, se presiona el botón “Graficar”.

Cuando se selecciona la opción “BOBINA”, se realiza la toma de datos de una bobina, para esto primero se coloca una bobina de radio 2.5 cm en el portamuestra del eje y , se selecciona el número de planos y posteriormente se presiona el botón “mapear”, luego de que el dispositivo termine de tomar datos se presiona el botón “Graficar”.

En la sección “ALAMBRE” e “IMAN”, se realiza la toma de datos de un alambre y de un imán respectivamente, para esto se coloca el objeto a medir en el portamuestras del eje y , posteriormente se presiona el botón: mapear, luego de que el dispositivo termine de tomar datos se presiona el botón “Graficar”.

La sección “MAGNITUD VS LONGITUD” realiza mediciones de la magnitud del campo magnético a distintas distancias predeterminadas. Para su funcionamiento primero se coloca el sensor bajo la muestra empleando los botones de la sección “MOVIMIENTOS DE MOTOR”, y una vez que se coloca el sensor en la posición deseada, se presiona el botón “mapear”. Cuando el dispositivo termina de realizar las mediciones, se presiona el botón “graficar”.

Al presionar el botón “mapear” en cualquiera de las secciones, el sensor y el portamuestras se posicionan en el punto de inicio seleccionado. Después el sensor tomará mediciones de

las tres componentes del campo magnético en cada uno de los puntos de un mallado específico y los almacenará en un archivo. Una vez que el sensor terminó de recorrer todos los puntos del mallado, se repite el proceso para cancelar el campo magnético de fondo, para esto es importante quitar la fuente cuando el programa lo indique.

Para calibrar el origen de los ejes del dispositivo, el soporte del eje móvil, la base del eje móvil 2 y el portamuestras se desplazan cada uno en dirección de los switches hasta accionarlos, al hacerlo los motores se detienen. Esta calibración se realiza a través de la función `zero_bobina` del apéndice C .

3.2. Características

Desplazamiento: Como se mencionó anteriormente, los motores que se emplean en este proyecto son: Nema-17, los cuales dan 200 pasos para completar una vuelta, es decir, cada paso corresponde a $360^\circ/200 = 1.8^\circ$. El sistema de transmisión consta de una banda dentada y las poleas tienen un radio de 6.4 mm. La longitud de arco es el desplazamiento del eje móvil, entonces por cada paso se recorre:

$$LA = \frac{2\pi}{200}r = (0.0314)(6.4 \text{ mm}) = 0.20 \text{ mm}$$

En la sección “BARRIDO RAPIDO” el dispositivo da 10 y 20 pasos de separación entre cada medición, lo que corresponde a 2 mm y 4 mm respectivamente.

En la sección “BOBINA” el dispositivo da 30 pasos de separación entre cada medición, lo que corresponde a 6 mm en cada eje. Finalmente en la sección ”MAGNITUD VS LONGITUD” el dispositivo da 25 pasos de separación entre cada medición, lo que corresponde a 5mm.

El tiempo de respuesta al presionar un botón y comience cualquier movimiento es de aproximadamente 1 segundo, y el tiempo de respuesta al presionar el botón “STOP” es inmediato.

3.3. Mediciones

Como nuestro objetivo general es la visualización del campo magnético de una fuente magnética. Como ejemplos se consideraron las siguientes fuentes: una bobina, un alambre

recto y un imán cilíndrico, cada uno con las siguientes características:

- Bobina: radio = 2.5 cm, número de vueltas = 100, $I = 1.5$ A, $V = 6.0$ V, $L = 4.5$ cm
- Alambre: diámetro = 0.57 mm, largo = 10 cm, $I = 3.0$ A, $V = 7.0$ V
- Imán: diámetro = 1.00 cm, alto = 0.5 cm, $B_0 = 9.0$ Gauss

La distancia entre cada punto de medición a lo largo del eje x y del eje z es de 6 mm, y la distancia entre planos (a lo largo del eje y) es de 30 mm.

Mediciones de la bobina.

Una vez que se colocó la estructura de madera con la bobina (figura 2.9b) en el portamuestras del eje y y se siguieron los pasos mencionados en la Sección 3.1, se obtuvo el mapeo del campo magnético en un plano paralelo a su eje, figura 3.2 y en un plano paralelo a su eje por fuera de la bobina figura 3.4.

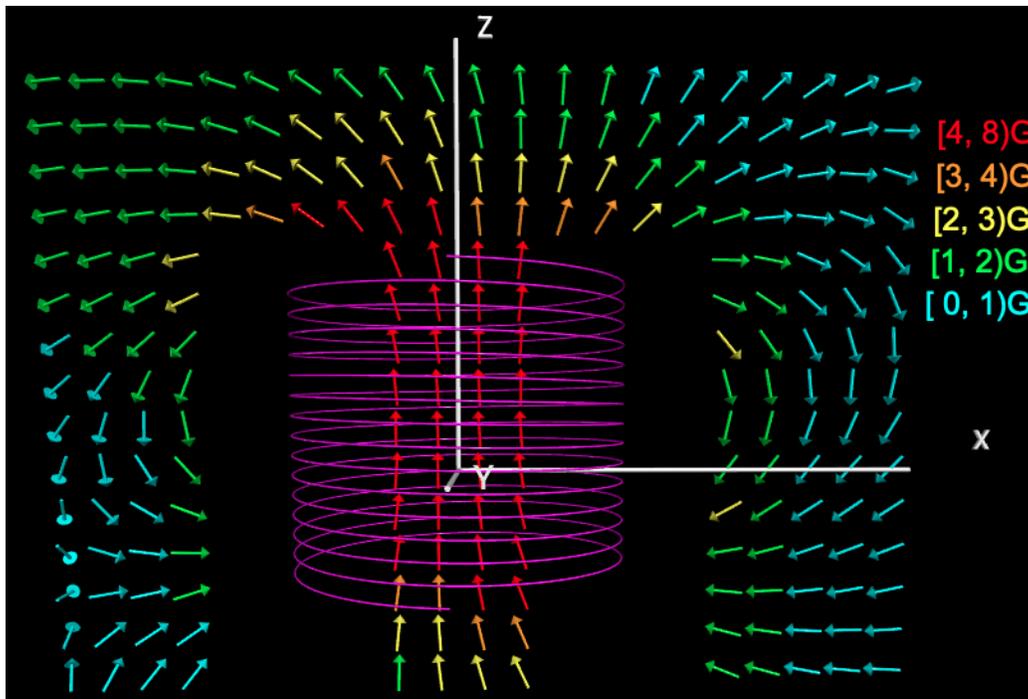


Figura 3.2: Mapeo del campo magnético sobre un plano paralelo al x, z .

De la figura 3.2, considerando simetría de la bobina observamos que:

- Las líneas de campo magnético son cerradas, pues rodean a las espiras, entrando por la parte superior de la bobina y saliendo por la parte inferior.
- La magnitud del campo magnético es mayor en el interior de la bobina a diferencia de su exterior.

Realizando un análisis de la magnitud del campo magnético sobre puntos a lo largo del eje x , para diferentes distancias sobre el eje z y para $y = \text{cte}$, notamos que, efectivamente, en el centro de la bobina y dentro de ella, la magnitud del campo magnético es mucho mayor comparado con su exterior y disminuye al alejarnos de la parte central al variar la posición en z (figura 3.3)[6].

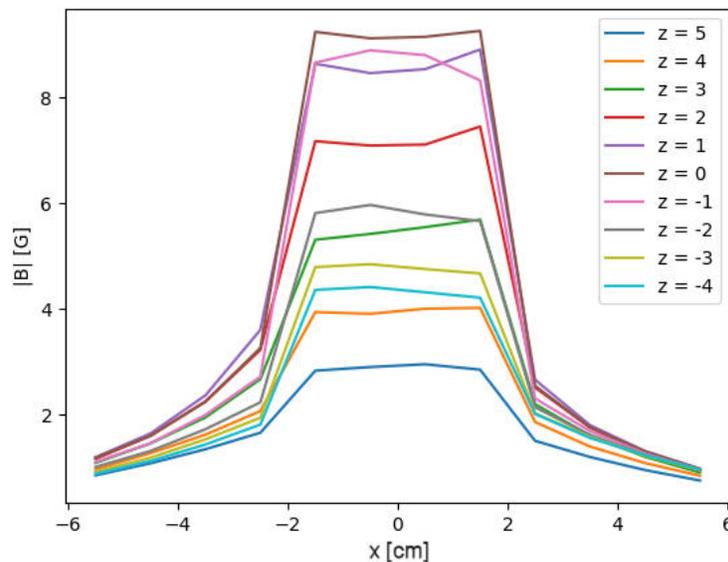


Figura 3.3: Magnitud del campo magnético para distintos planos del eje z .

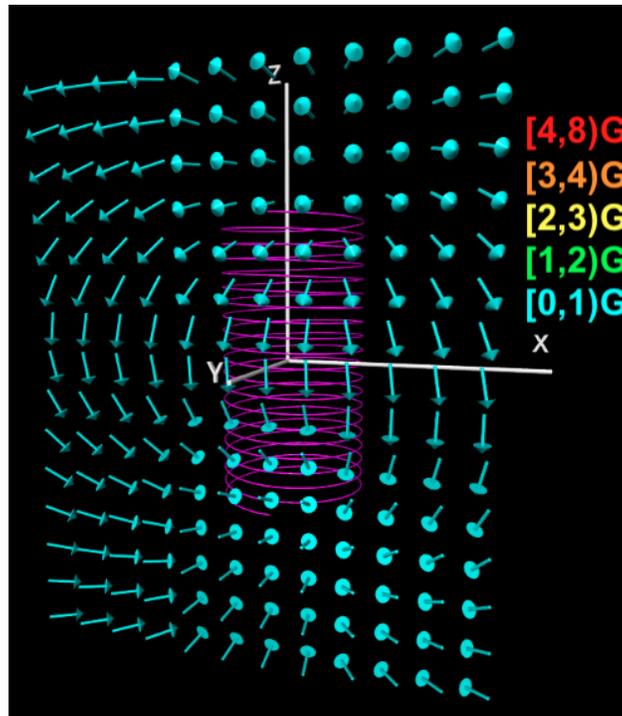


Figura 3.4: Mapeo del campo magnético sobre un plano exterior de la bobina empleada.

Por otro lado, de la figura 3.4, observamos que las líneas superiores apuntan en dirección del eje z y conforme descenden van inclinándose en dirección del eje y alcanzando esta dirección en el borde de la bobina, posterior a esto las líneas se van inclinando en dirección negativa del eje z , alcanzando esta dirección cuando están sobre la bobina y finalmente poco antes de llegar al extremo de la bobina, las líneas vuelven a inclinarse en dirección negativa del eje y ; de esto notamos que efectivamente, las líneas rodean la bobina.

Del análisis del comportamiento de las flechas de estas imágenes, consideramos que el campo magnético de la bobina es similar al que se presenta en la Sección 1.4.

Empleando la misma bobina pero ahora invirtiendo la dirección de la corriente que circula por la bobina obtenemos líneas de campo invertidas, es decir, las líneas dentro de la bobina apuntarán en dirección $-z$, figura 3.5.

Mediciones del alambre. Una vez que se colocó la estructura de madera con el alambre recto (figura 2.9a) en el portamuestras del eje y y se siguieron los pasos mencionados en la Sección 3.1, se obtuvo el mapeo del campo magnético en un plano perpendicular a su eje, figura 3.6.

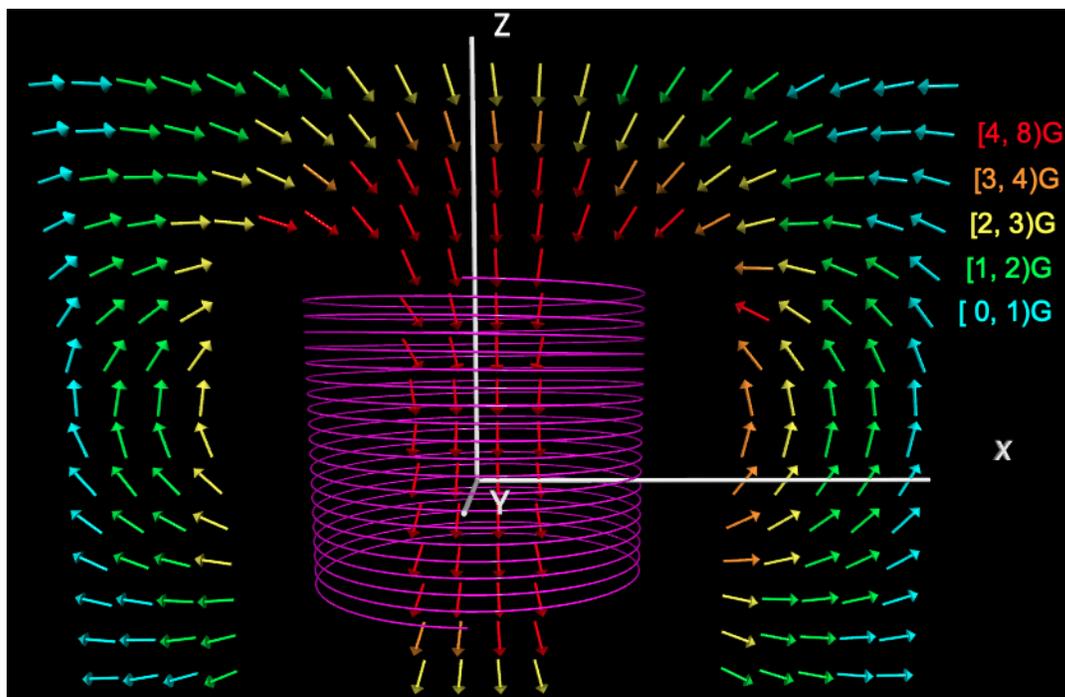


Figura 3.5: Mapeo del campo magnético sobre un plano paralelo al x, z con la dirección de la corriente invertida.

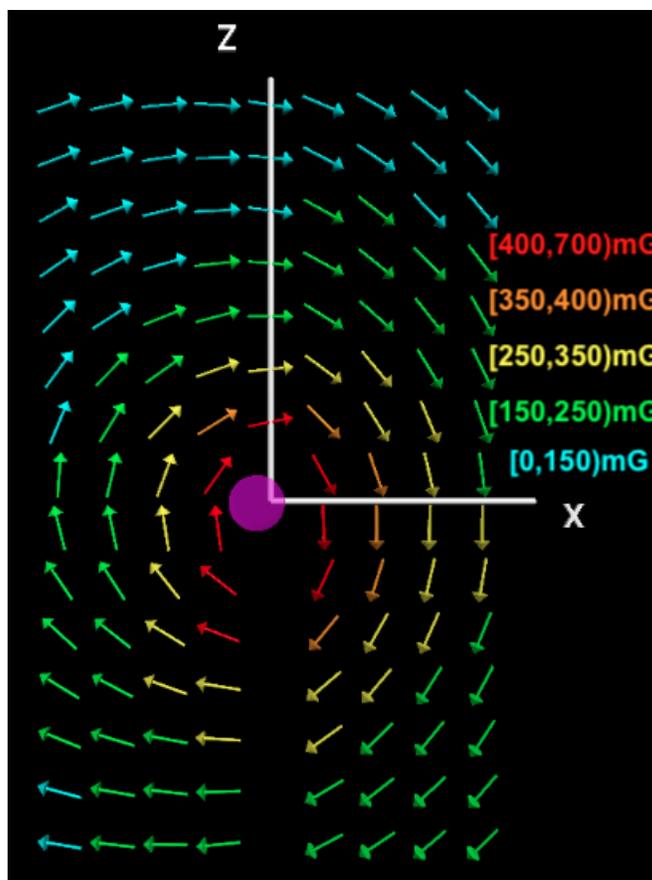


Figura 3.6: Mapeo del campo magnético de un alambre recto.

Observamos que el campo magnético rodea al alambre y, como las líneas de campo magnético son tangentes al campo magnético, tenemos que las líneas son circunferencias concéntricas al alambre (Sección 1.2.2), además la magnitud del campo magnético disminuye al incrementar la distancia y presenta simetría respecto a la posición del alambre (figura 3.7).

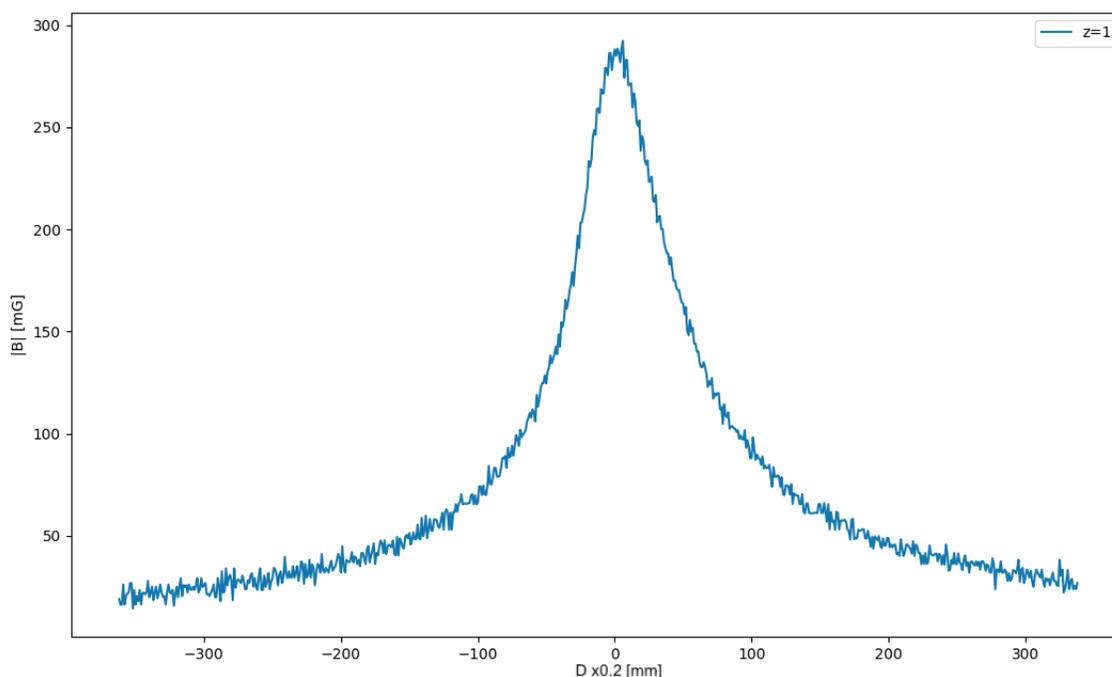


Figura 3.7: Magnitud del campo magnético a lo largo del eje x .

Mediciones del imán. Una vez que se colocó el imán en el porta-muestras y se siguieron los pasos mencionados en la Sección 3.1, se obtuvo el mapeo del campo magnético en un plano paralelo a su eje, figura 3.8.

De las figuras 3.8 y 3.9 observamos que las componentes del campo magnético centrales permanecen sobre el plano $z = 0$; del lado negativo del eje X apuntan hacia la dirección $-\hat{x}$ y conforme se acercan hacia el imán van girando hacia la dirección \hat{y} , sobre el imán las componentes apuntan hacia la dirección x y conforme se alejan del imán van girando hasta volver a tener la dirección $-x$. Lo que indica que las líneas de campo magnético salen por el lado izquierdo del imán, rodean al imán y entran por su otro extremo (Sección 1.2.1).

Las componentes del campo magnético fuera del centro siguen un patrón muy similar con la diferencia que para el plano $y = 0$ muestran componentes sobre el plano z y para el plano $y=1$ muestran componentes en las otras direcciones.

Todas las figuras antes presentadas se obtuvieron directamente de la interfaz gráfica que se planteó en la sección 2.4.

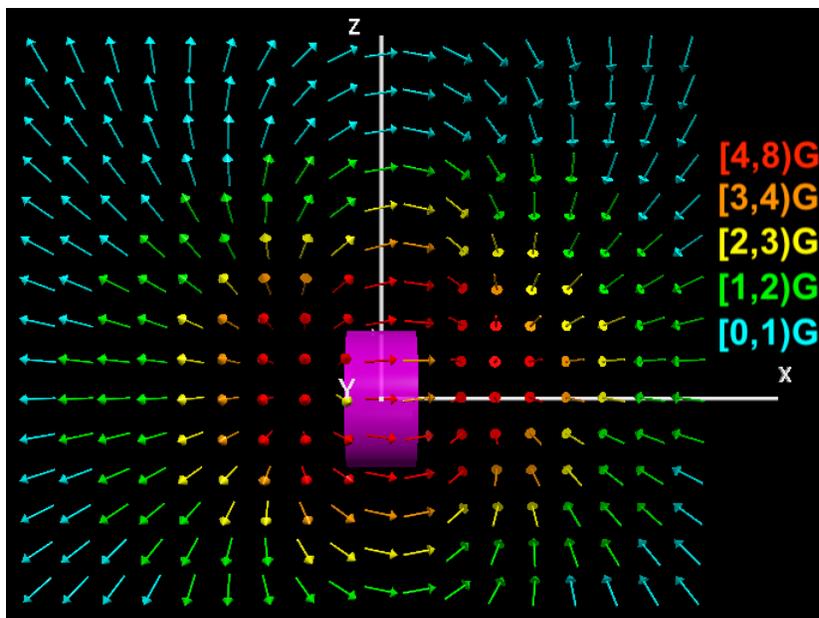


Figura 3.8: Mapeo del campo magnético de un imán cilíndrico sobre el plano $y=1$.

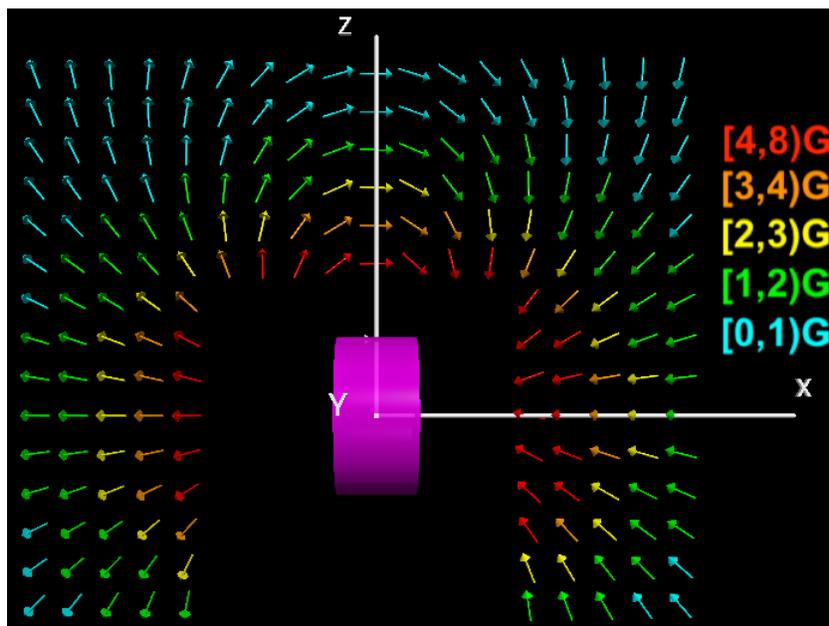


Figura 3.9: Mapeo del campo magnético de un imán cilíndrico sobre el plano $y=0$.

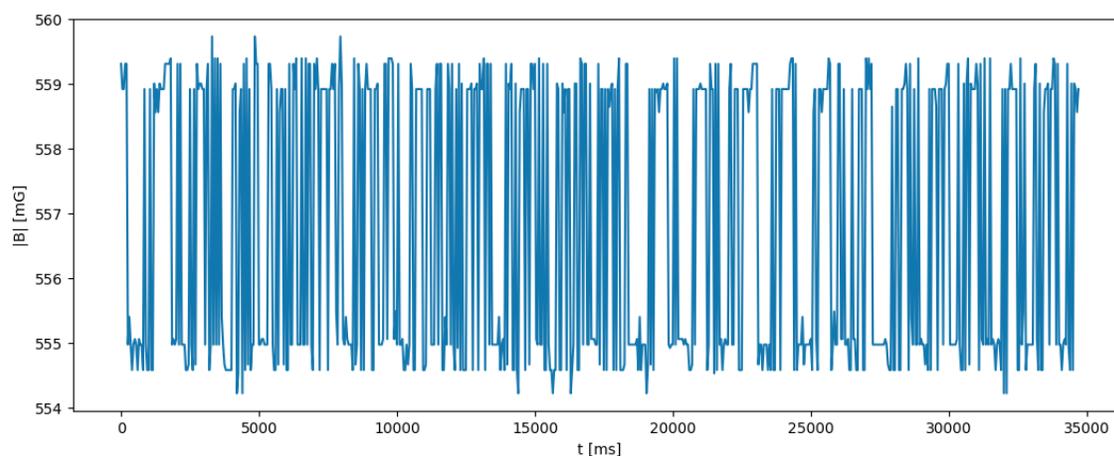


Figura 3.10: Magnitud del campo magnético terrestre.

Campo magnético terrestre. Finalmente, para comprobar la estabilidad del sensor, se midió el campo magnético terrestre durante un intervalo de tiempo (35 s). Para ello se colocó el sensor en un punto fijo, alejado de toda fuente magnética y se realizaron mediciones. Estos resultados se muestran en la figura 3.10.

De la figura 3.10 observamos que la magnitud del campo magnético se encuentra entre los valores de 554 mG y 560 mG, realizando un promedio de estos datos obtenemos que una magnitud para el campo terrestre de 556.8 ± 2.1 mG. También, de esta figura podemos ver que la resolución del sensor coincide con la nominal, que es de 5 mG.

3.4. Conclusiones

De los resultados anteriormente presentados, podemos decir que el dispositivo que se diseñó y construyó, cumplió satisfactoriamente con los objetivos mencionados de la Sección 1.11, pues nuestro dispositivo puede moverse libremente en 3 ejes alrededor de una fuente, realiza y almacena mediciones del campo magnético de una manera reproducible y, finalmente muestra los datos en una animación 3D, lo cual puede usarse como una herramienta didáctica.

Por otra parte, la interfaz gráfica es de fácil manejo y de mucha ayuda para el usuario, pues con ésta, sólo es necesario seleccionar algunos parámetros y el dispositivo comienza a realizar la toma de datos automáticamente y realiza su visualización.

Las ventajas destacables este dispositivo son que la mayor parte de su estructura está construida con material plástico y madera, lo que nos asegura que el campo magnético no se ve afectado por su estructura, al mismo tiempo que su construcción es de bajo costo y de fácil acceso a cualquier persona interesada en su reproducción. Además de que el *software* de la tarjeta *Arduino Nano* es fácilmente actualizable y el catalogo de sensores que son compatibles con esta tarjeta es muy amplio.

Apéndice A

Efecto Hall

El efecto Hall consiste en la aparición de un campo eléctrico (llamado campo Hall E_H) perpendicular a la dirección en la que circula una corriente eléctrica en un conductor en presencia de un campo magnético perpendicular a la corriente y normal al plano del conductor [7].

Para describir este efecto, consideramos un conductor rectangular, de espesor despreciable, de dimensiones transversales a y b , el cual lleva una corriente eléctrica I en la dirección positiva del eje x y a la cual se le aplica un campo magnético uniforme \vec{B} perpendicular al plano del conductor, el cual lo atraviesa en la dirección y , figura A.1.

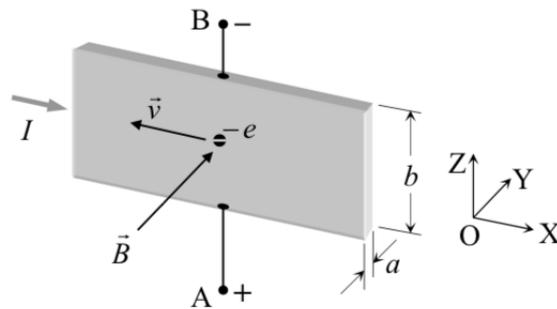


Figura A.1: Diagrama del efecto Hall.

De este modo, aparece la fuerza de Lorentz que actúa sobre los portadores de carga:

$$\vec{F} = q(\vec{v} \times \vec{B}) \quad (\text{A.1})$$

La cual provoca que los portadores de carga se distribuyan en uno de los extremos del conductor, generando una densidad superficial de carga, y en el otro extremo, aparecerá otra densidad de carga igual y opuesta.

Considerando que los portadores tengan carga $q = -e$, y que se mueven con una velocidad \vec{v} en la dirección negativa del eje x , dada la dirección y sentido del campo magnético, se tiene que la fuerza \vec{F}_m estará dirigida en la dirección z , de donde se obtiene que su magnitud es:

$$F_m = e v B \quad (\text{A.2})$$

Esta fuerza generará una distribución de carga negativa en la parte superior del conductor, mientras que en la parte inferior, se creará una distribución positiva [8]. De modo que tendremos un campo eléctrico, \vec{E}_H , y se alcanzará un estado de equilibrio cuando la fuerza neta sobre los portadores sea nula [7], es decir, cuando se verifique:

$$\vec{F}_m + (-e)\vec{E}_H = \vec{0} \implies E_H = v B \quad (\text{A.3})$$

Ya que el campo magnético es uniforme, el campo eléctrico también lo es y, entonces, el voltaje medido en los extremos del conductor será proporcional al campo magnético, de modo que al conocer la corriente, es posible obtener la magnitud del campo magnético:

$$V_H = E b \implies V_H = v B b \quad (\text{A.4})$$

Apéndice B

Código Python

```
1 import wx
2 import serial,time
3 from visual import *
4 from visual.graph import *
5
6 app = wx.App()
7 window = wx.Frame(None, title = "Ang_2000_2", size = (400,350))
8 panel=wx.Panel(window)
9 colors=(0.4,0.4,0.4)
10 opacidad=0.3
11
12 def barrid_2(self):
13     arduino=serial.Serial('/dev/cu.usbserial-AL033A8Y', 9600)
14     time.sleep(2)
15     arduino.write(b'7')
16     archivo=open('6_dic',"w")
17     a=1
18     while a<=20:
19         i = arduino.readline()
20         archivo.write(i)
21         for j in range(0,101):
22             if (a==j*5):
23                 print 'Llevas',a,'datos'
```

```
24     a=a+1
25     arduino.close()
26     archivo.close()
27
28 def grafic_2(self):
29     grafica=gdisplay(title='Magnitud de campo magnetico de una
30         bobina',xtitle='X',ytitle='B',x=0,y=0,width=500,height=300,
31         background=color.white,foreground=color.black)
32     posicion=gdots(gdisplay=grafica,color=color.red)
33     posicion2=gcurve(gdisplay=grafica,color=color.blue)
34     archivo=open('6_dic','r')
35     arc=archivo.readlines()
36     for i in arc:
37         arreglo=i.split()
38         posicion.plot(pos=(float(arreglo[3]),float(arreglo[1])))
39         posicion2.plot(pos=(float(arreglo[3]),float(arreglo[1])))
40     archivo.close()
41     sleep(0)
42
43 def brujul(self):
44     arduino=serial.Serial('/dev/cu.usbserial-AL033A8Y', 9600)
45     time.sleep(2)
46     arduino.write('8'.encode())
47     Norte = cylinder(pos=(0,0,0),axis=(0,0,1), radius=0.1, color=
48         color.red)
49     Sur = cylinder(pos=(0,0,0),axis=(0,0,-1), radius=0.1, color=
50         color.blue)
51     while(True):
52         line = arduino.readline()
53         if (len(line.split(" "))!=0):
54             B=line.split(" ")
55             Bx=float(B[1]);
56             By=float(B[4]);
57             Bz=float(B[7]);
58             r=sqrt(Bx*Bx+By*By+Bz*Bz);
59             Norte.axis=(Bx/r,By/r,Bz/r);
60             Sur.axis=(-Bx/r,-By/r,-Bz/r);
```

```
57         sleep(0.0);
58         arduino.close()
59
60     def izquierd(self):
61         arduino=serial.Serial('/dev/cu.usbserial-AL033A8Y', 9600)
62         time.sleep(2)
63         arduino.write('0'.encode())
64         arduino.close()
65
66     def derech(self):
67         arduino=serial.Serial('/dev/cu.usbserial-AL033A8Y', 9600)
68         time.sleep(2)
69         arduino.write('1'.encode())
70         arduino.close()
71
72     def enfrent(self):
73         arduino=serial.Serial('/dev/cu.usbserial-AL033A8Y', 9600)
74         time.sleep(2)
75         arduino.write('2'.encode())
76         arduino.close()
77
78     def atra(self):
79         arduino=serial.Serial('/dev/cu.usbserial-AL033A8Y', 9600)
80         time.sleep(2)
81         arduino.write('3'.encode())
82         arduino.close()
83
84     def sto(self):
85         arduino=serial.Serial('/dev/cu.usbserial-AL033A8Y', 9600)
86         time.sleep(2)
87         arduino.write('4'.encode())
88         arduino.close()
89
90     def arrib(self):
91         arduino=serial.Serial('/dev/cu.usbserial-AL033A8Y', 9600)
92         time.sleep(2)
93         arduino.write('5'.encode())
```

```

94     arduino.close()
95
96 def abaj(self):
97     arduino=serial.Serial('/dev/cu.usbserial-AL033A8Y', 9600)
98     time.sleep(2)
99     arduino.write('6'.encode())
100    arduino.close()
101
102 def EJES(coordx, coordy, coordz, a):
103     ejeX=cylinder(pos=(a,0,0), radius=0.06, axis=(7,0,0))
104     ejeY=cylinder(pos=(a,0,0), radius=0.06, axis=(0,6,0))
105     ejeZ=cylinder(pos=(a,0,0), radius=0.06, axis=(0,0,4))
106
107     X=text(text='%s'%coordx, pos=(6,0,0), height=0.5)
108     Y=text(text='%s'%coordy, pos=(-1,6,0), height=0.5)
109     Z=text(text='%s'%coordz, pos=(-1,0,4), height=0.5)
110
111 def FLECHAS(arc):
112     B=etiq()
113     particion=(B[1]-B[0])/5.0
114     archivo=open('%s'%arc, 'r')
115     lee = archivo.readlines()
116     for i in lee:
117         arreglo=i.split()
118         marco=frame()
119         norma=sqrt(float(arreglo[1])*float(arreglo[1])+float(
            arreglo[5])*float(arreglo[5])+float(arreglo[3])*float(
            arreglo[3]))
120         palito=cylinder(pos=(0,0,0), radius=0.04, axis=(0.5,0,0),
            frame=marco)
121         piquito=cone(pos=(0.5,0,0), radius=0.11, axis=(0.2,0,0), frame
            =marco)
122         marco.pos=(float(arreglo[6]), -float(arreglo[7]), float(
            arreglo[8]))
123         marco.axis=(float(arreglo[1]), float(arreglo[5]), float(
            arreglo[3]))
124         if norma>=B[0] and norma<B[0]+particion:

```

```
125         palito.color=(0,1,1)
126         piquito.color=(0,1,1)
127     elif norma>B[0]+particion and norma<B[0]+2*particion:
128         palito.color=(0,1,0)
129         piquito.color=(0,1,0)
130     elif norma>=B[0]+2*particion and norma<B[0]+3*particion:
131         palito.color=(1,1,0)
132         piquito.color=(1,1,0)
133     elif norma>=B[0]+3*particion and norma<B[0]+4*particion:
134         palito.color=(1,0.6,0)
135         piquito.color=(1,0.6,0)
136     elif norma>=B[0]+4*particion:
137         palito.color=(1,0,0)
138         piquito.color=(1,0,0)
139     sleep(0.1)
140     archivo.close()
141
142 def dos_archivos():
143     datos=[]
144     datos2=[]
145     with open('ar_0','r') as archivo, open('ar_1','r') as archivo2:
146         for lineas in archivo:
147             datos.append(lineas.split())
148         for lineas2 in archivo2:
149             datos2.append(lineas2.split())
150     suma=open('suma','w')
151     for i in range(len(datos)):
152         suma.write('X= %f Y= %f Z= %f %d %d %d \n'%(float(datos2[i]
153             ] [1]) -float(datos[i][1]),float(datos2[i][3]) -float(datos
154             [i][3]),float(datos2[i][5]) -float(datos[i][5]),int(datos
155             [i][6]),int(datos[i][7]),int(datos[i][8])) )
156     suma.close()
157
158 def grafic(self):
159     ventana=display(title='Ang',center=(0,0,0),width=600,height
160         =600,background=colors)
161     EJES('X','Z','Y',0)
```

```
158     iman=box(pos=(0,0,0),axis=(1,0,0),length=4,heigh=4,width=3,up
        =(1,0,0),color=color.magenta,opacity=opacidad)
159     dos_archivos()
160     FLECHAS('suma')
161
162     def selec(self):
163         mallado=str(combo.GetValue())
164         paso=float(combo_4.GetValue())
165         cadena=[mallado,paso]
166         return cadena
167
168     def pl(self):
169         numero=float(pla.GetValue())
170         return numero
171
172     def imp(self):
173         valores=selec(self)
174         print"Los valores del mallado ",valores,"son los siguientes:"
175         if valores==["15X15X1",0.5]:
176             lectura('a',225)
177         if valores==["15X15X1",1.0]:
178             lectura('b',225)
179         if valores==["15X15X2",0.5]:
180             lectura('c',450)
181         if valores==["15X15X2",1.0]:
182             lectura('d',450)
183         if valores==["15X15X3",0.5]:
184             lectura('e',675)
185         if valores==["15X15X3",1.0]:
186             lectura('f',675)
187
188     def lectura(letra,da):
189         arduino=serial.Serial('/dev/cu.usbserial-AL033A8Y', 9600)
190         time.sleep(2)
191         for a in range(2):
192             arduino.write(b'%s'%(letra))
193             if a==0:
```

```
194         archivo=open('ar_%d'%a,'w')
195     if a==1:
196         archivo=open('ar_%d'%a,'w')
197     c=1
198     while c<=da:
199         i = arduino.readline()
200         archivo.write(i)
201         for j in range(da+1):
202             if (c==j*15):
203                 print 'Llevas',c,'datos'
204                 label_7.SetLabel('Llevas %d datos'%c)
205             c=c+1
206         if a==0: print('*****ENCIENDE LA FUENTE*****')
207         if a==1: print('*****APAGA LA FUENTE*****')
208         time.sleep(30)
209     archivo.close()
210     arduino.close()
211
212 def bobina_dat(self):
213     plan=int(pla.GetValue())
214     if plan==1:
215         valor=plan*180
216         lectura('g',valor)
217     if plan==2:
218         valor=plan*180
219         lectura('h',valor)
220     if plan==3:
221         valor=plan*180
222         lectura('i',valor)
223     dos_archivos()
224
225 def etiq():
226     archivo=open('suma','r')
227     maximo=0.0
228     minimo=10000.0
229     lee = archivo.readlines()
230     for i in lee:
```

```

231     arreglo=i.split()
232     norma=sqrt(float(arreglo[1])*float(arreglo[1])+float(
        arreglo[5])*float(arreglo[5])+float(arreglo[3])*float(
        arreglo[3]))
233     if norma>maximo: maximo=norma
234     if norma<minimo: minimo=norma
235     B=[minimo,maximo]
236     archivo.close()
237     particion=(maximo-minimo)/5.0
238     for j in range(5):
239         inter=minimo
240         minimo=minimo+particion
241         for i in range(0,9,2):
242             etiqueta=label(pos=(9,i*0.4,0),box=False,visible=True)
243             if i==0 and j==0: etiqueta.text="|B|-->[%.1f,%.1f)mG"
                %(inter,minimo));etiqueta.color=(0,1,1)
244             elif i==2 and j==1: etiqueta.text="|B|-->[%.1f,%.1f)mG
                "%(inter,minimo));etiqueta.color=(0,1,0)
245             elif i==4 and j==2: etiqueta.text="|B|-->[%.1f,%.1f)mG
                "%(inter,minimo));etiqueta.color=(1,1,0)
246             elif i==6 and j==3: etiqueta.text="|B|-->[%.1f,%.1f)mG
                "%(inter,minimo));etiqueta.color=(1,0.6,0)
247             elif i==8 and j==4: etiqueta.text="|B|-->[%.1f,%.1f)mG
                "%(inter,minimo));etiqueta.color=(1,0,0)
248     return B
249
250 def bobin(self):
251     ventana=display(title='Ang',center=(0,0,0),width=600,height
        =600,background=colors)
252     EJES('X','Z','Y',-0.5)
253     bobina=helix(pos=(-0.5,-4,0),axis=(0,5,0),radius=2,color=color.
        magenta,coils=20,thickness=0.05,opacity=opacidad)
254     etiq()
255     FLECHAS('suma')
256
257 def alamb(self):

```

```
258     ventana=display(title='Ang',center=(0,0,0),width=750,height
259         =650,background=colors)
260     wire=cylinder(pos=(-0.25,-0.5,-1),radius=0.5,axis=(0,0,2),color
261         =color.magenta,opacity=opacidad)
262     etiq()
263     EJES('X','Z','Y',0)
264     FLECHAS('suma')
265
266 def alambre_dat(self):
267     lectura('p',127)
268     dos_archivos()
269
270 def ima_datos(self):
271     lectura('q',570)
272     dos_archivos()
273
274 def ima(self):
275     ventana=display(title='Ang',center=(0,0,0),width=600,height
276         =600,background=colors)
277     etiq()
278     iman=cylinder(pos=(-1,-2,0),axis=(2,0,0),radius=2,color=color.
279         magenta,opacity=0.4)
280     EJES('X','Z','Y',0)
281     FLECHAS('suma')
282
283 label_1 = wx.StaticText(panel, label = "MOVIMIENTOS DE MOTOR", pos
284     = (120,0))
285
286 izquierda = wx.Button(panel, label='izquierda', pos=(0,50))
287 izquierda.Bind(wx.EVT_BUTTON, izquierd )
288
289 derecha = wx.Button(panel, label='derecha', pos=(200,50))
290 derecha.Bind(wx.EVT_BUTTON, derech )
291
292 enfrente = wx.Button(panel, label='enfrente', pos=(100,20))
293 enfrente.Bind(wx.EVT_BUTTON, enfrent )
```

```
290  atras = wx.Button(panel, label='atras', pos=(100,80))
291  atras.Bind(wx.EVT_BUTTON, atra )
292
293  stop = wx.Button(panel, label='stop', pos=(100,50))
294  stop.Bind(wx.EVT_BUTTON, sto )
295
296  arriba = wx.Button(panel, label='arriba', pos=(300,20))
297  arriba.Bind(wx.EVT_BUTTON, arrib )
298
299  abajo = wx.Button(panel, label='abajo', pos=(300,80))
300  abajo.Bind(wx.EVT_BUTTON, abaj )
301
302  brujula=wx.Button(panel, label='brujula', pos=(200,100))
303  brujula.Bind(wx.EVT_BUTTON,brujul)
304
305  label_2 = wx.StaticText(panel, label = "BARRIDO RAPIDO", pos =
306          (140,130))
307
308  wx.StaticText(panel, label = "Dimensiones del mallado (x,y,z)", pos
309          = (30,150))
310  wx.StaticText(panel, label = "Tamano del paso", pos = (260,150))
311
312  size = ['15X15X1', '15X15X2', '15X15X3']
313  combo= wx.ComboBox(panel, -1, "15X15X1", (70, 170), wx.DefaultSize,
314          size, wx.CB_DROPDOWN)
315  combo.Bind(wx.EVT_COMBOBOX,selec)
316
317  size_2=['0.5', '1.0']
318  combo_4= wx.ComboBox(panel, -1, "0.5", (270, 170), wx.DefaultSize,
319          size_2, wx.CB_DROPDOWN)
320  combo_4.Bind(wx.EVT_COMBOBOX,selec)
321
322  sel = wx.Button(panel, label='Barrido', pos=(100,200))
323  sel.Bind(wx.EVT_BUTTON, imp )
324
325  wx.StaticText(panel, label = "cm", pos = (340,175))
```

```
323 grafica = wx.Button(panel, label='graficar', pos=(200,200))
324 grafica.Bind(wx.EVT_BUTTON, grafic )
325
326 label_3 = wx.StaticText(panel, label = "MAGNITUD VS LONGITUD", pos
    = (220,240))
327
328 barrido_2 = wx.Button(panel, label='mapear', pos=(200,260))
329 barrido_2.Bind(wx.EVT_BUTTON, barrid_2 )
330
331 grafica_2 = wx.Button(panel, label='graficar', pos=(290,260))
332 grafica_2.Bind(wx.EVT_BUTTON, grafic_2)
333
334 label_4 = wx.StaticText(panel, label = "BOBINA", pos = (0,240))
335 label_5 = wx.StaticText(panel, label = "plano", pos = (70,240))
336
337 plano=['1','2','3']
338 pla= wx.ComboBox(panel, -1, "1", (110, 236), wx.DefaultSize, plano,
    wx.CB_DROPDOWN)
339 pla.Bind(wx.EVT_COMBOBOX,pl)
340
341 bobina = wx.Button(panel, label='GRAFICAR', pos=(85,260))
342 bobina.Bind(wx.EVT_BUTTON, bobin )
343
344 bobina_datos = wx.Button(panel, label='MAPEAR', pos=(0,260))
345 bobina_datos.Bind(wx.EVT_BUTTON, bobina_dat )
346
347 label_6 = wx.StaticText(panel, label = "ALAMBRE", pos = (30,285))
348
349 alambre = wx.Button(panel, label='graficar', pos=(85,300))
350 alambre.Bind(wx.EVT_BUTTON, alamb )
351
352 alambre_datos = wx.Button(panel, label='mapear', pos=(0,300))
353 alambre_datos.Bind(wx.EVT_BUTTON, alambre_dat )
354
355 label_7 = wx.StaticText(panel,pos = (250,300))
356
357 label_8 = wx.StaticText(panel, label = "IMAN", pos = (275,285))
```

```
358
359 iman_datos = wx.Button(panel, label='mapear', pos=(200,300))
360 iman_datos.Bind(wx.EVT_BUTTON, ima_datos )
361
362 iman = wx.Button(panel, label='graficar', pos=(290,300))
363 iman.Bind(wx.EVT_BUTTON, ima )
364
365 window.Show(True)
366 app.MainLoop()
```

Apéndice C

Código Arduino

```
1  #include <Wire.h>
2  #include <HMC5883L.h>
3  HMC5883L magnetometro;
4
5  const int X_DIR = 5;
6  const int Y_DIR = 6;
7  const int Z_DIR = 7;
8
9  const int X_STP = 2;
10 const int Y_STP = 3;
11 const int Z_STP = 4;
12
13 const int swit=11;
14 int estado_swit=0;
15 bool ya=false;
16
17 const int swit_3=10;
18 int estado_swit_3=0;
19 bool ya_3=false;
20
21 const int swit_2=12;
22 int estado_swit_2=0;
23 bool ya_2=false;
```

```
24
25  const int pin=13;
26
27  int delayTime=1200;
28
29  float Norma,x,y,z;
30
31  void zero_p(int a, int b, int c, int d, int e, int f){
32  a=digitalRead(d);
33  while(a==HIGH){
34      if(c==0){
35          stp(X_DIR, e, X_STP);
36          stp(Y_DIR, HIGH, Y_STP);}else{stp(Z_DIR, f, Z_STP);}
37          a=digitalRead(d);
38      }
39  digitalWrite(pin,HIGH);
40  for(int i=1;i<=10*16;i++){
41      if(c==0){
42          stp(X_DIR, f, X_STP);
43          stp(Y_DIR, LOW, Y_STP);}else{stp(Z_DIR, e, Z_STP);}
44      }
45  do{
46      if(c==0){
47          stp(X_DIR, e, X_STP);
48          stp(Y_DIR, HIGH, Y_STP);}else{ stp(Z_DIR, f, Z_STP);      }
49      a=digitalRead(d);
50      }while(a==HIGH);
51  b = true;
52  digitalWrite(pin,LOW);
53  }
54
55  void zero_bobina(){
56  zero_p(estado_swit_3,ya_3,0,swit_3,HIGH,LOW);delay(800);
57  Step(X_DIR, LOW, X_STP, Y_DIR, LOW, Y_STP, 140);delay(800);
58
59  zero_p(estado_swit_2,ya_2,1,swit_2,LOW,HIGH);delay(800);
60  for(int l=0;l<1250;l++){stp(Z_DIR, LOW, Z_STP);}
```

```
61
62 zero_p(estado_swit,ya,0,swit,LOW,HIGH);delay(800);
63 Step(X_DIR, HIGH, X_STP, Y_DIR, LOW, Y_STP, 910);delay(800);
64 delay(2000);
65 }
66
67 void Step(int motx, int est_motx, int stepperPinx,int moty, int
    est_moty, int stepperPiny, int steps){
68 digitalWrite(motx, est_motx);
69 digitalWrite(moty, est_moty);
70 for (int i = 0; i < steps; i++) {
71 digitalWrite(stepperPinx, HIGH);
72 delayMicroseconds(delayTime);
73 digitalWrite(stepperPinx, LOW);
74 delayMicroseconds(delayTime);
75 digitalWrite(stepperPiny, HIGH);
76 delayMicroseconds(delayTime);
77 digitalWrite(stepperPiny, LOW);
78 delayMicroseconds(delayTime);}
79 }
80
81 void stp(int MOT, int EST_MOT, int stepperPin){
82 digitalWrite(MOT, EST_MOT);
83 digitalWrite(stepperPin, HIGH);
84 delayMicroseconds(delayTime);
85 digitalWrite(stepperPin, LOW);
86 delayMicroseconds(delayTime);
87 }
88
89 void comp(float X,float Y, float Z){
90 Serial.print("X= ");Serial.print(X);
91 Serial.print(" Y= ");Serial.print(Y);
92 Serial.print(" Z= ");Serial.print(Z);delay(100);}
93
94 void mapeo_2(int A, int N,int x,int y, int z,int B){
95 int h; h=B;
96 for(int i=x;i<=y;i++){ /*veces a lo largo del eje X*/
```

```

97   for(int j=-7;j<=z;j++){ /*veces a lo largo del eje Y*/
98       Step(X_DIR, LOW, X_STP, Y_DIR, LOW, Y_STP, N);
99       Vector norm=magnetometro.readNormalize();comp(norm.XAxis,norm.
        YAxis,norm.ZAxis);Serial.print("  ");
100      Serial.print(i);Serial.print("  ");Serial.print(j);Serial.
        print("  ");Serial.println(h);
101      delay(500);}
102      for(int k=0;k<A;k++){ /*regreso en eje Y*/
103          Step(X_DIR, HIGH, X_STP, Y_DIR, HIGH, Y_STP, N);}delay(500);
104          Step(X_DIR, LOW, X_STP, Y_DIR, HIGH, Y_STP, N);delay(500);
105      }
106  }
107
108  void saltito(int ang, int ie){
109      Step(X_DIR, HIGH, X_STP, Y_DIR, LOW, Y_STP, 30*ang);delay(500);
110      for(int l=0;l<ie;l++){stp(Z_DIR, HIGH, Z_STP);}delay(250);
111  }
112
113  void alambre(int A){
114      for(int i=1;i<=A;i++){
115          mapeo_2(15,30,-4,-1,7,i);
116          mapeo_2(7,30,0,0,-1,i);
117          Step(X_DIR, LOW, X_STP, Y_DIR, HIGH, Y_STP, 65);delay(800);
118          mapeo_2(15,30,1,4,7,i);
119          if(A>1){
120              Step(X_DIR, HIGH, X_STP, Y_DIR, LOW, Y_STP, 30*9+65);delay(270);
121              for(int l=0;l<50;l++){stp(Z_DIR, HIGH, Z_STP);}delay(250);}
122          }
123      }
124
125  void planos(int A,int B){
126      for(int i=1;i<=A;i++){
127          mapeo_2(15,B,-7,7,7,i);
128          if(A>1){
129              Step(X_DIR, HIGH, X_STP, Y_DIR, LOW, Y_STP, B*15);delay(270);
130              for(int l=0;l<50;l++){stp(Z_DIR, HIGH, Z_STP);}delay(250);}
131          }

```

```
132     }
133
134 void bobina(int A){
135 if(A<=3){
136     mapeo_2(15,30,-10,10,7,3);
137     if(A<=2){
138         saltito(21,260);
139         mapeo_2(15,30,-10,-7,7,0);
140         mapeo_2(4,30,-6,-3,-4,0);
141         mapeo_2(15,30,-2,1,7,0);
142         mapeo_2(4,30,2,5,-4,0);
143         mapeo_2(15,30,6,10,7,0);
144         if(A<=1){
145             saltito(21,260);
146             mapeo_2(15,30,-10,10,7,-3);}
147         } }
148     }
149
150 float magnit(float x, float y, float z){
151 Norma=sqrt(x*x+y*y+z*z);
152 Serial.print(Norma);}
153
154 void maglon(){
155 for(int m=5;m<25;m++){
156 Vector norm=magnetometro.readNormalize();
157 Serial.print("Magnitud= ");magnit(norm.XAxis,norm.YAxis,norm.ZAxis
158 );
159 Serial.print(" ");
160 Serial.print("Distancia= ");Serial.println(m);
161 for(int n=0;n<25;n++){stp(Z_DIR, HIGH, Z_STP);}delay(400);
162 } }
163
164 void setup() {
165 while (!magnetometro.begin())
166 {delay(500);}
167 magnetometro.setRange(HMC5883L_RANGE_8_1GA);
168 magnetometro.setMeasurementMode(HMC5883L_CONTINUOUS);
```

```
168 magnetometro.setDataRate(HMC5883L_DATARATE_75HZ);
169 magnetometro.setSamples(HMC5883L_SAMPLES_8);
170 Serial.begin(9600);
171 }
172
173 void loop() {
174   if(Serial.available()>0){
175     delay(5);
176     char option=Serial.read();
177     while(option=='0'){
178       stp(X_DIR, HIGH, X_STP);
179       stp(Y_DIR, LOW, Y_STP);
180       delay(5);
181     }
182     while(option=='1'){
183       stp(X_DIR, LOW, X_STP);
184       stp(Y_DIR, HIGH, Y_STP);
185       delay(5);
186     }
187     while(option=='2'){
188       stp(X_DIR, LOW, X_STP);
189       stp(Y_DIR, LOW, Y_STP);
190       delay(5);
191     }
192     while(option=='3'){
193       stp(X_DIR, HIGH, X_STP);
194       stp(Y_DIR, HIGH, Y_STP);
195       delay(5);
196     }
197     if(option=='4'){
198     }
199
200     while(option=='5'){
201       stp(Z_DIR, HIGH, Z_STP);delay(5);
202     }
203
204     while(option=='6'){
```

```
205 stp(Z_DIR, LOW, Z_STP);delay(5);
206 }
207
208 if(option=='7'){
209     zero_p(estado_swit_3,ya_3,0,swit_3,HIGH,LOW);delay(800);
210     Step(X_DIR, LOW, X_STP, Y_DIR, LOW, Y_STP, 550);delay(800);
211
212     zero_p(estado_swit,ya,0,swit,LOW,HIGH);delay(800);
213     Step(X_DIR, HIGH, X_STP, Y_DIR, LOW, Y_STP, 580);delay(800);
214
215     zero_p(estado_swit_2,ya_2,1,swit_2,LOW,HIGH);delay(800);
216     for(int l=0;l<900;l++){stp(Z_DIR, LOW, Z_STP);}delay(800);
217
218     maglon();
219     }
220
221     while(option=='8'){
222         Vector norm=magnetometro.readNormalize();
223         comp(norm.XAxis,norm.YAxis,norm.ZAxis);Serial.println(" ");
224         delay(100);
225         if(Serial.available(>0){
226             delay(5);
227             char option_2=Serial.read();
228             if(option_2=='4'){option=4;}
229
230             if(option_2=='0'){
231                 Step(X_DIR, HIGH, X_STP, Y_DIR, LOW, Y_STP, 30);
232                 delay(5);}
233
234             if(option_2=='1'){
235                 Step(X_DIR, LOW, X_STP, Y_DIR, HIGH, Y_STP, 30);
236                 delay(5);}
237
238             if(option_2=='2'){
239                 Step(X_DIR, LOW, X_STP, Y_DIR, LOW, Y_STP, 30);
240                 delay(5);}
241
```

```
242     if(option_2=='3'){
243         Step(X_DIR, HIGH, X_STP, Y_DIR, HIGH, Y_STP, 30);
244         delay(5);}
245
246     if(option_2=='5'){
247     for(int an=1;an<=40;an++){ stp(Z_DIR, HIGH, Z_STP);delay(5);
248         delay(5);} }
249
250     if(option_2=='6'){
251     for(int gie=1;gie<=40;gie++){ stp(Z_DIR, LOW, Z_STP);delay(5);
252         delay(5);} }
253     }
254     }
255 if(option=='a'){
256     zero_bobina();delay(800);
257     planos(1,25); }
258
259 if(option=='b'){
260     zero_bobina();delay(800);
261     planos(1,50); }
262
263 if(option=='c'){
264     zero_bobina();delay(800);
265     planos(2,25);}
266
267 if(option=='d'){
268     zero_bobina();delay(800);
269     planos(2,50); }
270
271 if(option=='e'){
272     zero_bobina();delay(800);
273     planos(3,25); }
274
275 if(option=='f'){
276     zero_bobina();delay(800);
277     planos(3,50); }
278
```

```
279  if(option=='g'){
280      zero_bobina();
281      bobina(3);}
282
283  if(option=='h'){
284      zero_bobina();
285      bobina(2); }
286
287  if(option=='i'){
288      zero_bobina();
289      bobina(1); }
290
291  if(option=='p'){
292      zero_bobina();delay(100);
293      Step(X_DIR, LOW, X_STP, Y_DIR, LOW, Y_STP, 240);delay(100);
294      Step(X_DIR, LOW, X_STP, Y_DIR, HIGH, Y_STP, 160);delay(100);
295      delay(1000);
296      alambre(1);  }
297
298  if(option=='q'){
299      zero_bobina();delay(100);
300      delay(1000);
301      Step(X_DIR, HIGH, X_STP, Y_DIR, LOW, Y_STP, 15);
302      Step(X_DIR, LOW, X_STP, Y_DIR, LOW, Y_STP, 180);
303      mapeo_2(15,30,-8,9,7,-3);
304      Step(X_DIR, HIGH, X_STP, Y_DIR, LOW, Y_STP, 30*18);delay(500);
305      for(int l=0;l<150;l++){stp(Z_DIR, HIGH, Z_STP);}delay(250);
306          mapeo_2(15,30,-8,-4,7,0);
307          mapeo_2(6,30,-3,4,-2,0);
308          mapeo_2(15,30,5,9,7,0);
309          saltito(18,150);
310          mapeo_2(15,30,-8,9,7,3);
311          delay(1000);
312      }
313 }
```


Bibliografía

- [1] Espejel R A, Murguía G, Calles A, Cabrera E and Morán J L, Magnetic fields produced by rotating symmetrical bodies with homogeneous surface charge density, *Eur. J. Phys.* **37** (2016) 045204.
- [2] Amato M, Daleana F, Coviello C, De Vittorio M, Petroni S, Modeling, fabrication and characterization of micro-coils as magnetic inductors for wireless power transfer, *Microelectronic Engineering* **111** (2013) pp 143-148
- [3] Reitz J R, Milford F J, Christy R W, *Fundamentos de la Teoría Electromagnética*, 3^a ed., Alhambra Mexicana, México, 2001, pp 180-181.
- [4] Smythe W R, *Static and Dynamic Electricity*, 3^a ed., A summa book, USA, 1989, pp 290-291.
- [5] Alonso M, Finn E J, *Física volumen II: campos y ondas*, 1^a ed., Addison-wesley ibero-americana, México, 1987, pp 540, 547.
- [6] Serway R A, *Electricidad y magnetismo*, 3^a ed., McGraw-Hill, México, 1993, pp 215, 216, 218.
- [7] Halliday D, Resnick R and Krane K S, *Física Vol. 2*, 4^a ed., Cecsca, México, 1999, pp 169-170.
- [8] Sears, Zemansky, Young, Freedman, *Física Universitaria con Física Moderna*, volumen 2, 12^a ed., Pearson educación, México, 2009, pp 943
- [9] Morales E, *Análisis y diseño de un motor síncrono lineal como medio de propulsión para un tren de levitación magnética*. Tesis de licenciatura en ingeniería, 2013 IPN. Repositorio Dspace. Consultado el 21 de abril del 2023.
<https://tesis.ipn.mx/handle/123456789/12638>

- [10] Salazar J, Estudio técnico-comparativo para la introducción de cocinas eléctricas de inducción magnética en el ecuador. Tesis de licenciatura en ingeniería, 2010. Consultado el 25 de abril del 2023.
<http://bibdigital.epn.edu.ec/handle/15000/2165>
- [11] C Oswin, Aldana C, Hernandez C, , Resonancia magnética funcional: evolución y avances en clínica, *Tecnura*, **13** (2009) pp 88-103. Consultado el 21 de abril del 2023.
<https://www.redalyc.org/articulo.oa?id=257020617009>
- [12] Metrolab, Hallinsight Hall magnetic field camera, consultado el 21 de abril del 2023.
<https://www.metrolab.com/products/hallinsight-hall-magnetic-cameras/>
- [13] Schneider Electric, Stepper Motors: NEMA-17 [Datasheet], consultado el 21 de abril del 2023.
<https://datasheetspdf.com/pdf-file/1260602/Schneider/NEMA17/1>
- [14] Texas instruments, DRV8825 Steper Motor Controller IC, consultado el 21 de abril del 2023.
<https://datasheetspdf.com/pdf-file/1411598/etcTI/DRV8825/1>
- [15] Arduino, consultado el 21 de abril del 2023.
<https://www.arduino.cc/>
- [16] Arduino, placa Arduino nano, consultado el 21 de abril del 2023.
<https://store-usa.arduino.cc/products/arduino-nano?selectedStore=us>
- [17] Honeywell, 3-Axis Digital Compass IC HMC5883L, consultado el 21 de abril del 2023.
<https://www.farnell.com/datasheets/1683374.pdf>
- [18] Máquina plotter CNC Mini xy 2 ejes, consultado el 22 de abril del 2023.
<https://www.pinterest.com.mx/pin/433964114085852275/>
- [19] Partículas magnéticas, consultado el 22 de abril del 2023.
<https://sites.google.com/site/laboratorioensmateriales/particulas-magneticastmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>
- [20] CNC History: The Origination and Evolution of CNC Machining, consultado el 22 de abril del 2023.
<https://www.rapiddirect.com/blog/cnc-history/>