

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



Diseño de un módulo de medición de cilindros para una fresadora vertical CNC

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en
Electrónica con el grado académico de Licenciatura**

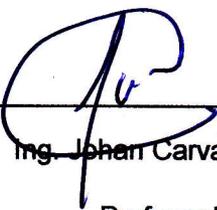
Miguel Rodríguez Escudé

Cartago, Noviembre de 2012

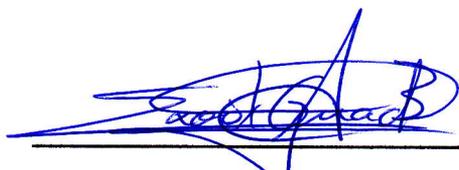
INSTITUTO TECNOLOGICO DE COSTA RICA
ESCUELA DE INGENIERIA ELECTRONICA
PROYECTO DE GRADUACIÓN
TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



Ing. Jehan Carvajal Godínez
Profesor lector



Dr. Saúl Guadamuz Brenes
Profesor lector



Lic. Marvin Hernández Cisneros
Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, Miércoles 28 de Noviembre de 2012

Declaratoria de autenticidad

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, Noviembre 2012

Miguel Rodríguez Escudé

Miguel Rodríguez Escudé

Céd: 114080364

Resumen

En este documento se propone un diseño de un módulo de medición de cilindros para una fresadora vertical CNC.

La empresa Ad Astra Rocket Company (AARC) tiene como objetivo el desarrollo y construcción del Motor de Magnetoplasma de Impulso Específico Variable (VASIMR®). Su sede en el país (AARC-CR) se encarga del análisis y optimización de las características termodinámicas de dicho motor, así como explorar la optimización de una fuente de plasma para la tecnología VASIMR®.

Siguiendo estos objetivos, el Ing. Juan Ignacio Del Valle Gamboa, empleado de AARC-CR, trabaja actualmente en su proyecto de doctorado titulado "*Modelo experimental de las interacciones entre el plasma y sus superficies de confinamiento en fuentes helicoidales de alta potencia*". El confinamiento del plasma ocurre dentro de un cilindro interno mediante el campo magnético producido por electroimanes de alta potencia. El cilindro sufre una erosión a raíz de las variaciones del campo magnético que permiten que partículas de plasma choquen contra él.

Dicho doctorado depende de múltiples mediciones de la superficie interna del cilindro interno, para determinar el comportamiento del plasma con el campo magnético, y así hacer las correcciones para aumentar la vida útil de los componentes del motor.

AARC-CR no cuenta con los instrumentos para hacer dichas mediciones, por lo que dependen de la disponibilidad de la maquinaria de otras instituciones. Por esta razón, se pretende añadir un módulo de medición a una fresadora vertical CNC disponible en AARC-CR, para que sea capaz de realizar las mediciones necesarias sin depender de otras instituciones.

Abstract

This paper describes the design of a cylinder measurement module for a vertical CNC milling machine.

Ad Astra Rocket Company (AARC) focuses in the development of the Variable Specific Impulse Magnetoplasma Rocket (VASIMR®). Its subsidiary, Ad Astra Rocket Company Costa Rica (AARC-CR), investigates the thermo dynamical characteristics of the rocket.

With this goal in mind, Ing. Juan Ignacio Del Valle is working on his doctoral thesis titled “Experimental model of the interactions between plasma and its confinement walls in high power helicoidal sources”. The confinement of the plasma occurs inside a cylinder in the rocket, caused by the magnetic field generated by a high power magnetic coil. The cylinder gets eroded because of the plasma leaks due to the variation of the magnetic confinement field.

This doctoral thesis depends of a series of measurements of the inner surface of the cylinder, to determine the behavior of the plasma with in the magnetic field, allowing optimizing the life span of the components of the rocket.

AARC-CR doesn't have the proper instruments to do these measurements, so they depend on the availability of the measurement machines of other companies. This is why this project attempts to design a measurement module that can be added to a vertical CNC milling machine, so that AARC-CR can perform the measurements in their facilities without any inconveniences.

Índice de contenido

Capítulo 1. Introducción	10
1.1. Problema existente e importancia	10
1.2. Solución planteada	12
Capítulo 2. Meta y objetivos.....	14
2.1. Meta	14
2.2. Objetivo general	14
2.3. Objetivos específicos.....	14
Capítulo 3. Marco teórico.....	15
3.1. Control Numérico Computarizado (CNC).....	15
3.1.1. Fresadora vertical CNC CENTROID™ Retrofit	15
3.1.2. G-Code.....	19
3.2. Dispositivo HID USB.....	20
3.3. Triangulación láser	20
3.4. Coordenadas cilíndricas	21
3.5. Dispositivos disponibles en el mercado	22
Capítulo 4. Procedimiento metodológico.....	24
4.1 Reconocimiento y definición del problema.....	24
4.2 Obtención y análisis de información	24
4.3 Evaluación de las alternativas y síntesis de la solución	25
4.4 Implementación de la solución.....	26
4.5 Revaluación y rediseño.....	27
Capítulo 5. Descripción de la solución	28
5.1 Análisis de soluciones y selección final.....	28
5.2 Descripción del hardware	29
5.2.1. Sistema de sujeción del sensor de medición y el cilindro.....	29
5.2.2. Sistema para la calibración de la posición de inicio	31
5.2.3. Pantalla LCD	31
5.2.4. Módulo RTC	32
5.2.5. Sensor Riftek RF-603	33
5.2.6. Simulador de sensor Riftek RF-603.....	34
5.2.7. Almacenamiento en tarjeta microSD.....	35

5.2.8.	Plataforma principal	36
5.3	Descripción del software	38
5.3.1.	Módulos de calibración	38
5.3.1.1.	Cercanía al cilindro	38
5.3.1.2.	Ángulo de inicio	39
5.3.2.	Simulación del sensor Riftek RF-603	40
5.3.3.	Módulo de medición	41
5.3.4.	Módulo de obtención de datos de RTC	42
5.3.5.	Flujo completo de medición	43
5.3.6.	Graficado de datos	45
Capítulo 6.	Análisis de resultados	48
6.1.	Resultados	48
6.2.	Análisis	49
Capítulo 7.	Conclusiones y recomendaciones	51
7.1.	Conclusiones	51
7.2.	Recomendaciones	51
Capítulo 8.	Bibliografía	52
Capítulo 9.	Anexos	53
A.1.	Código de las funciones relevantes	53
A.1.1.	Acercar a cilindro	53
A.1.2.	Angulo de inicio	54
A.1.3.	Medición	54
A.1.4.	Simulación de sensor	55
A.2.	Códigos de programación y solicitudes para el sensor	56
A.3.	Comandos de teclado de la CNC	56
A.4.	Comandos de teclado de Arduino Leonardo	57

Índice de figuras

Figura 1. Esquema del motor de plasma VASIMR.....	10
Figura 2. Cilindros de cuarzo (SiO ₂) y alúmina (Al ₂ O ₃) con signos de desgaste.	11
Figura 3. Diagrama de bloques general.	13
Figura 4. Fresadora vertical CNC.....	16
Figura 5. Sistema de coordenadas de fresadora vertical CNC.	16
Figura 6. Módulo de rotación (Eje A).	17
Figura 7. Pantalla inicial del controlador la fresadora vertical CNC.	18
Figura 8. Puertos de controlador de la fresadora vertical CNC.....	18
Figura 9. Principio de operación por triangulación láser.	20
Figura 10. Sistema de coordenadas cilíndricas.	21
Figura 11. Mitutoyo Crysta Apex	22
Figura 12. Mitutoyo RoundTest.	23
Figura 13. Pieza de sujeción del cilindro con el eje A.	29
Figura 14. Pin interno del émbolo (Sujeción de sensor)	30
Figura 15. Parte externa del émbolo (Sujeción a eje Z).....	30
Figura 16. Pieza de calibración de cercanía del sensor y ángulo de inicio.	31
Figura 17. Esquemático de conexión de LCD.	32
Figura 18. Módulo RTC con chip DS1307.	32
Figura 19. Esquemático de conexión de RTC.	33
Figura 20. Sensor Rifftek RF-603.....	34
Figura 21. Esquemático de conexión de comunicación serie.	34
Figura 22. Esquemático de conexión de MAX-232.....	35
Figura 23. Arduino Ethernet Shield con SD.	36
Figura 24. Arduino Leonardo.....	37
Figura 25. Esquemático de conexión del circuito completo.	37
Figura 26. Diagrama de flujo de acercamiento al cilindro.	39
Figura 27. Diagrama de flujo de movimiento al ángulo de inicio.....	40
Figura 28. Diagrama de flujo de simulador de sensor.....	41
Figura 29. Diagrama de flujo de toma de datos del sensor.....	42
Figura 30. Diagrama de flujo de toma de datos de RTC.....	43
Figura 31. Diagrama de flujo de proceso de inicialización.	44
Figura 32. Diagrama de flujo de proceso completo de medición.	44
Figura 33. Diagrama de flujo del graficador de datos.	46
Figura 34. Vista interna del cilindro graficado.....	46
Figura 35. Vista externa del cilindro graficado.....	47
Figura 36. Captura del archivo de datos creado.....	48
Figura 37. Applet del programa graficador.....	49

Índice de tablas

Tabla 1. Códigos M y G comunes.....	19
Tabla 2. Registros y variables relevantes de la fresadora.....	19
Tabla 3. Códigos de solicitudes para el sensor Riftek RF-603.....	56
Tabla 4. Comandos de teclado para la fresadora CNC.....	56
Tabla 5. Comandos de teclado para Arduino Leonardo.....	57

Capítulo 1. Introducción

1.1. Problema existente e importancia

Este proyecto se desarrolló en las instalaciones de la empresa Ad Astra Rocket Company Costa Rica (AARC-CR), ubicada en Liberia, Guanacaste. El objetivo principal de la empresa engloba todo lo relacionado al análisis térmico y de desgastes del motor de plasma VASIMR®. Igualmente, se involucran en el desarrollo completo del sistema, en términos de eficiencia y optimización.

El desarrollo del motor de plasma VASIMR®, realizado por AARC-CR, es muy dependiente de sistemas de medición exactos y precisos, razón por la cual enfocan sus recursos a asegurarse de que todos los componentes que integran dicho motor sean de la más alta calidad, asegurando la confiabilidad y durabilidad del mismo.

El motor contiene un cilindro interno en la etapa de entrada, encargado de contener el haz de plasma que impulsa al motor. El plasma está confinado mediante unos electroimanes que se encargan de generar el campo magnético que limita al plasma al centro del cilindro. Por las variaciones del confinamiento, partículas de plasma se desvían y chocan contra la superficie interna del cilindro, erosionándolo.

Este cilindro se encuentra alrededor de la sección marcada con el número 3, en la figura 1. La antena helicoidal que ioniza el gas se encuentra dentro del cilindro.

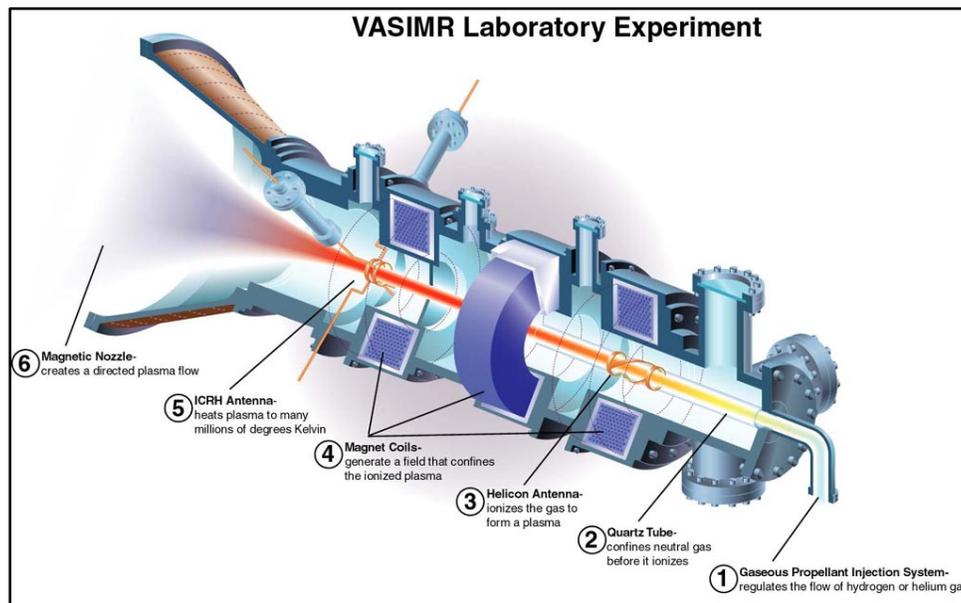


Figura 1. Esquema del motor de plasma VASIMR.¹

¹ <http://upload.wikimedia.org/wikipedia/commons/1/13/Vasimr.jpg>

La figura 2 muestra una imagen de los cilindros a medir.

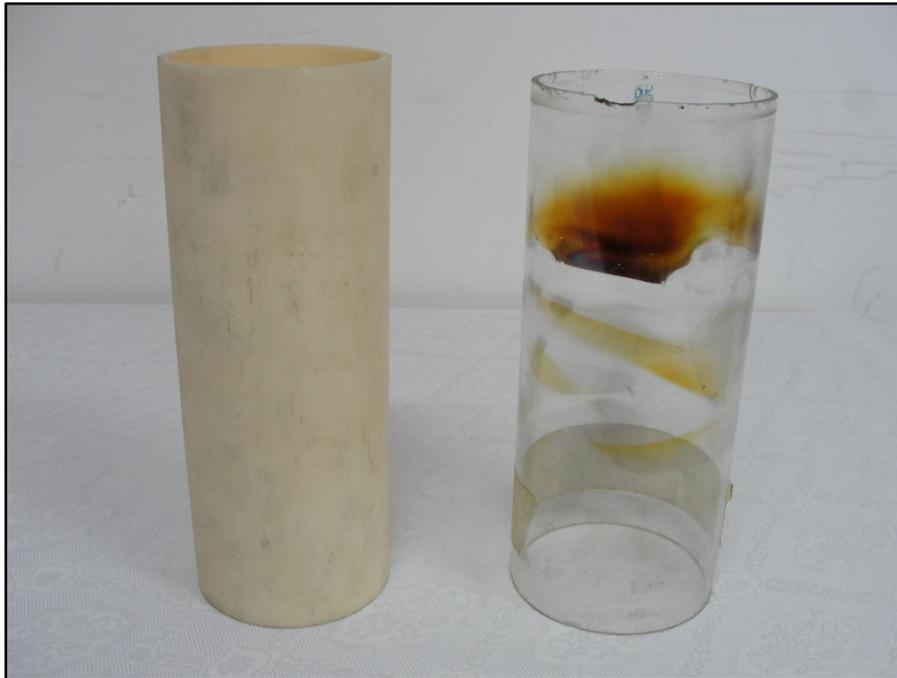


Figura 2. Cilindros de cuarzo (SiO_2) y alúmina (Al_2O_3) con signos de desgaste.

Dadas las condiciones de operación del motor, es muy importante caracterizar y dimensionar el desgaste producido por el funcionamiento del motor y poder determinar las opciones para contrarrestar o disminuir dicho desgaste. El desgaste se puede caracterizar midiendo el perfil de la superficie interna del cilindro.

La empresa no cuenta con una máquina para hacer dichas mediciones, por lo que depende de otras instituciones para estos estudios. Actualmente, estas mediciones se realizan en el Laboratorio de Metrología del Campus Mario Echandi del Instituto Nacional de Aprendizaje, localizado en El Coyol, Alajuela.

El laboratorio de metrología cuenta con una Máquina de Medición por Coordenadas (MMC) con Control Numérico Computarizado (CNC). Esta permite hacer mediciones en tres dimensiones con una precisión de 0.1 micrómetros, utilizando una punta de prueba por contacto. Este tipo de medición, aunque es muy precisa, es muy lenta, ya que por la cantidad de mediciones que se deben realizar, medir un cilindro de 220 milímetros de largo y 90 milímetros de diámetro puede tomar hasta 18 horas.

Esto conlleva gastos en viáticos, recursos humanos y tiempo de investigación, debido a que el investigador debe estar presente al momento de las mediciones en caso de algún imprevisto, así como desplazarse desde AARC-CR hasta el laboratorio de metrología.

De contar con la maquinaria adecuada con AARC-CR, la empresa se podría ahorrar mucho tiempo y recursos al realizar las mediciones en su propio laboratorio, ya que aunque la medición tome la misma cantidad de tiempo, se puede continuar el flujo de trabajo, se reducen los gastos en viáticos y se evitan contratiempos que puedan interrumpir la medición.

1.2. Solución planteada

Según las estimaciones de AARC-CR, los cambios en las dimensiones del cilindro pueden variar cada 2 ó 3 años, por lo que el diseño del modulo de medición se enfocó en el cilindro actual, cuyas dimensiones son 226 milímetros de largo, y 90 milímetros de diámetro interno.

Con el fin de aprovechar recursos, se utilizó una fresadora vertical CNC, que cuenta con un sistema de control desarrollado por la empresa CENTROID, de Estados Unidos. Esta cuenta con un software que permite la manipulación completa de la maquina a partir de comandos de teclado y programación en códigos G.

El diseño planteado utiliza un sensor láser de alta precisión y corto rango, que provee una resolución muy alta, así como altas frecuencias de muestreo. Este sensor se situó en la prensa universal de la fresadora, que equivale al eje Z. De esta manera se controló el movimiento vertical del sensor.

El cilindro cuenta con una base de aluminio atornillada al eje A de la fresadora, para evitar que cambie de posición mientras se realiza la medición, además de utilizar un pin de anclaje, para garantizar que el cilindro siempre se comience a medir en el mismo punto. Esto permitió girar el cilindro para poder realizar la medición interna completa.

El sistema esta diseñado para que mida el radio del cilindro en diferentes puntos en la circunferencia, y a diferentes alturas, de modo que se puede determinar la erosión de un punto comparando el radio medido luego de una prueba del motor con su medida original.

El diseño desarrollado cuenta con varios módulos, para controlar la posición del sensor, el giro del cilindro, posiciones de inicio, adquisición y almacenamiento de datos.

La figura 3 muestra un diagrama de bloques general, para representar la comunicación entre los diferentes módulos.

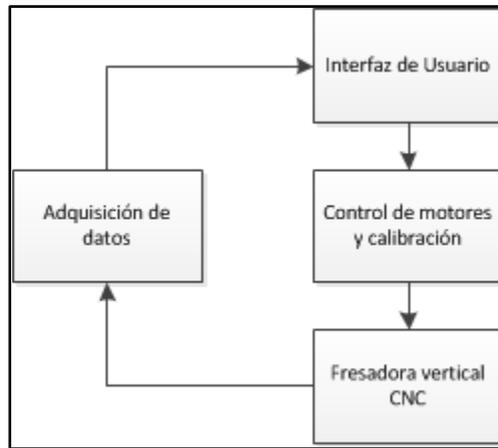


Figura 3. Diagrama de bloques general.

Al terminar la medición, los datos se almacenan en una tarjeta de memoria SD. Se crea un archivo de texto en formato *Comma Separated Values* (CSV) cuyo nombre corresponde al día, mes y año de la medición. Por ejemplo, el archivo 06112012.txt correspondería a la medición realizada el 06 de noviembre de 2012.

Adicionalmente, se desarrolló un programa de graficado básico, para crear un modelo en tres dimensiones del cilindro medido, representando el nivel de erosión mediante una escala de colores. Esto permite al investigador reconocer las secciones del cilindro que presentan mayor desgaste de forma más rápida e intuitiva.

Capítulo 2. Meta y objetivos

2.1. Meta

Elaborar un módulo de medición para una fresadora vertical CNC que permita a AARC-CR realizar las mediciones de erosión de un cilindro en sus propias instalaciones sin depender de otras instituciones ni incurrir en los altos costos de adquirir instrumentación de metrología.

2.2. Objetivo general

Diseñar un módulo de medición de la superficie interna de cilindros para una fresadora vertical CNC.

2.3. Objetivos específicos

- Diseñar un modulo de medición para cilindros entre 120 y 240 milímetros de largo.
- Crear una base de datos que contenga archivos en formato CSV con los radios medidos del cilindro.
- Desarrollar una aplicación de software que permita modelar en tres dimensiones el arreglo de datos medido.

Capítulo 3. Marco teórico

3.1. Control Numérico Computarizado (CNC)

Dada la necesidad de aumentar la precisión, repetitividad y versatilidad de los diferentes tipos de maquinaria industrial, se desarrollo un sistema computarizado para controlar la maquinaria y, en su mayor parte, sustituir a los operadores para eliminar el factor humano de la producción. [1]

El uso de esta tecnología se da por el aumento en la flexibilidad, seguridad, productividad y precisión en la manipulación de materiales, además de disminuir los costos de producción y herramientas, al poder realizar trabajos de alta complejidad sin intervención del usuario.

El CNC se basa en un sistema controlado con servo motores, controlado por una computadora o sistema embebido, el cual ejecuta un programa basado en instrucciones específicas. Este programa puede tener como objetivo crear una pieza (en el caso de un torno, fresadora o dobladora), o crear un modelo digital de una pieza utilizando sensores de posición por coordenadas (conocidas como Coordinate Measuring Machine o CMM).

Mediante el uso de un sistema computarizado, se ejerce control completo sobre el movimiento y ordenes de ejecución de la máquina sin tener que contrarrestar el efecto de la inercia o frenado de los servo motores. Esto hace posible obtener resoluciones en el rango de los micrómetros o nanómetros, dependiendo del sistema.

3.1.1. Fresadora vertical CNC CENTROID™ Retrofit

La compañía AARC-CR cuenta con un laboratorio de mecánica de precisión, y entre las herramientas de las que disponen, se encuentra una fresadora vertical CNC, automatizada por la empresa CENTROID™. Esta se puede apreciar en la figura 4.

Este controlador es el M-400S de la compañía CENTROID™, con el software CNC10 Versión 2.36. Este fue instalado en el año 2005, según informó AARC-CR.



Figura 4. Fresadora vertical CNC

Esta cuenta con el sistema de coordenadas representado en la figura 5. Sobre la mesa de trabajo (plano XY), se puede colocar un cuarto eje rotacional, denotado por el sistema como eje A. El módulo del eje A se puede observar en la figura 6.

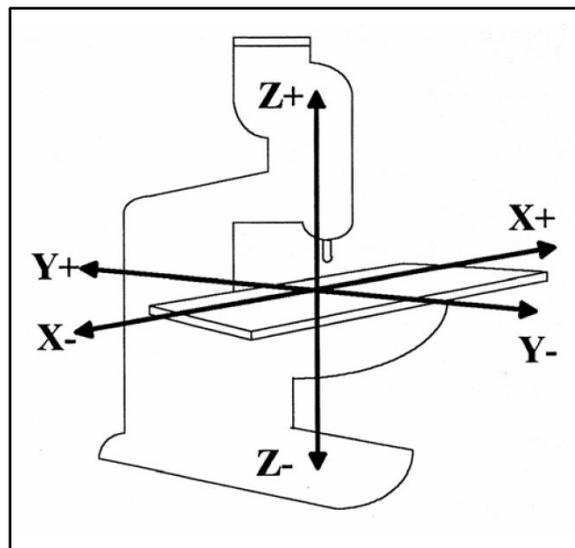


Figura 5. Sistema de coordenadas de fresadora vertical CNC.



Figura 6. Módulo de rotación (Eje A).

Este sistema tiene un rango de movimiento de 750mm en el eje X, 370mm en el eje Y, 120mm en el eje Z y 360° en el eje A. Además, la mesa de trabajo cuenta con un ajuste de altura manual en el caso de que se necesiten maquinarse piezas de altas dimensiones. [2]

La fresadora está controlada con una computadora, que es la encargada de controlar el controlador lógico programable (PLC) que acciona los servomotores. Esta computadora ejecuta un programa basado en Linux, que se encarga de realizar las funciones que indique el usuario.

Cuenta con opciones como cargar y editar programas, un modo Manual Data Input (MDI) y cambiar parámetros del sistema. El modo MDI permite digitar comandos individualmente de forma manual, únicamente se debe digitar el código correcto, presionar la tecla Enter del teclado y presionar CycleStart en el panel de control. Dicho modo hizo posible la manipulación de la máquina en este proyecto.

Esta máquina cuenta con múltiples limitaciones, ya que es una fresadora vertical antigua, que fue automatizada por la compañía CENTROID™ en Estados Unidos. En el año en el que se realizó la automatización de la máquina, las opciones de enviar información, como la posición actual de cada eje, mediante un puerto USB o serie con el protocolo RS-232, son nulas.

La figura 7 muestra la pantalla de inicio del controlador de la fresadora, que incluye las opciones disponibles en la parte inferior, con las teclas de acceso indicadas. En la figura 7, se encuentran los puertos disponibles en el sistema. Los puertos RJ45 (Ethernet) y DB9 (RS-232) están dedicados (mediante el software), al envío de programas de maquinado, con la extensión “.cnc” o “.txt”. El puerto USB es utilizado comúnmente para el acceso a sistemas de almacenamiento masivo. Por último, un conector hembra de 18 pines es el utilizado para la conexión del generador manual de pulsos, o MPG por sus siglas en inglés.



Figura 7. Pantalla inicial del controlador la fresadora vertical CNC.



Figura 8. Puertos de controlador de la fresadora vertical CNC.

En la figura 8 también se puede apreciar el control MPG, que consiste de una perilla giratoria, que indica los incrementos o decrementos de posición que debe tomar el eje seleccionado (perilla intermedia) según el multiplicador seleccionado (perilla inferior).

3.1.2. G-Code

Este es lenguaje de programación utilizado para la mayoría de los tipos de maquinaria CNC. Cuenta con códigos de movimiento o acción llamados códigos G, ya que empiezan con una G y un número acorde a la acción a ejecutar. [2]

También se encuentran los códigos M, que se refieren a rutinas de software, como crear archivos, enviar información, guardar datos o leer una entrada del PLC del controlador.

En la tabla 1 se detallan algunos de los códigos G y códigos M que son reconocidos por la fresadora vertical CNC de AARC-CR. No todas las máquinas reconocen todos los códigos, debido a disponibilidad de opciones y compatibilidad de versiones. Además, se incluyen algunos registros importantes en la tabla 2, que contienen variables del sistema.

Tabla 1. Códigos M y G comunes.

Código	Función
M00	Espera una acción del operador
M01	Espera opcional del operador
M02	Reiniciar programa
M03	Rotación de herramienta CW
M04	Rotación de herramienta CCW
M05	Apagado de rotación de herramienta
G00	Posicionamiento rápido
G01	Interpolación lineal
G02	Circular or Helical Interpolation CW
G03	Circular or Helical Interpolation CCW
G09	Chequeo de parada

Tabla 2. Registros y variables relevantes de la fresadora.

Número de registro	Dato
5021-5024	Posición de la máquina (X=5021, Y=5022, etc.). Retorna valor flotante.
5041-5044	Posición actual (X=5041, Y=5042, etc.). Retorna valor flotante.
6001-6080	Entradas de PLC 1 - 80
6900-6909	Entradas de PLC, en arreglos de 8.
7001-6080	Salidas de PLC 1-80

En este proyecto, se utilizó el código G "G00", para el posicionamiento del sensor, y los registros #5041 y #5044, para obtener las posiciones actuales de los ejes X y A, respectivamente.

3.2. Dispositivo HID USB

Este tipo de dispositivos nace por la iniciativa de fomentar y facilitar la innovación en dispositivos de entrada de computadoras y simplificar su proceso de instalación. Se utiliza el puerto Universal Serial Bus (USB) para la comunicación entre el dispositivo y la computadora.

Las siglas HID significan “Human Interface Device” o, en español, dispositivo de interfaz humana. Esto por tratarse de dispositivos con interacción bidireccional entre el usuario y el equipo.

El desarrollo de los controladores para dispositivos HID permite al usuario conectar el dispositivo al puerto USB y este, automáticamente, lo reconocerá para hacer uso de la información enviada, ya que estos soportan la tecnología Plug and Play (PnP).

Los objetivos principales de la clase HID son: ser lo más compacto posible para ahorrar espacio de datos del dispositivo; Permitir la aplicación de software para omitir información desconocida; Ser extensible y robusto; Que soporte la anidación y las colecciones y ser auto descriptivo para permitir aplicaciones de software genéricas. [3]

Los dispositivos mayormente utilizados son teclados, ratones, touchpads, joysticks. También existen otros buses de datos con el protocolo HID, como Bluetooth HID y Serial HID.

3.3. Triangulación láser

Este es un principio de medición sin contacto, que permite altas frecuencias de muestreo, con alta resolución o rangos de medición, dependiendo del modelo del sensor. La figura 9 muestra un esquema del funcionamiento de un sensor láser con este método de medición. [4]

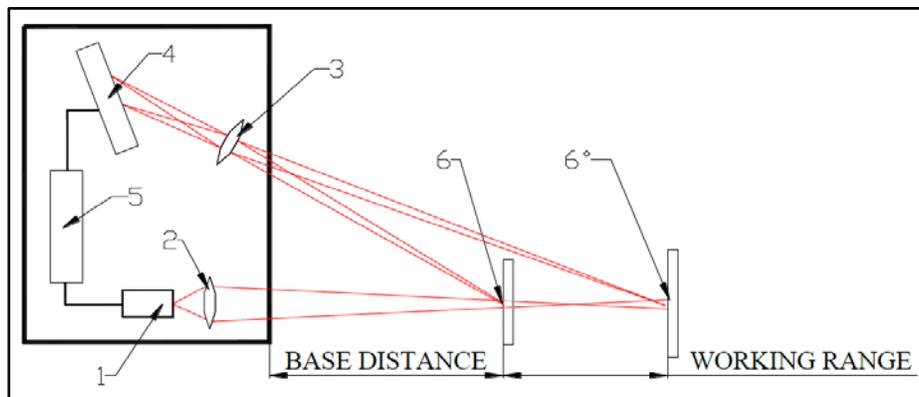


Figura 9. Principio de operación por triangulación láser.

Consiste en un generador láser (1) que pasa a través de un lente (2). Este se refleja en el objeto a medir (Entre el punto 6 y 6°), pasa a través de un lente receptor (3) y luego un modulo detector (5) realiza el calculo de medición según el punto enfocado en la matriz receptora (4). Para que la medición sea correcta, el objeto se debe mantener en el rango de medición (6-6°). [4]

3.4. Coordenadas cilíndricas

En este sistema de coordenadas, un punto P del espacio queda determinado por la intersección de 3 superficies: un cilindro cuyo eje es el eje Z y su radio r; un semiplano limitado por el eje z y que forma un ángulo φ con el eje X (CCW); un plano paralelo al plano X-Y y a una distancia Z del mismo. [5]

A dicho punto se le asignan 3 coordenadas: r, φ y z (z es la misma que en coordenadas cartesianas). Los rangos de cada una de ellas son:

- $r < 0$;
- $0 < \varphi < 2\pi$ y
- $-\infty < z < +\infty$.

La figura 10 representa dicho sistema de coordenadas.

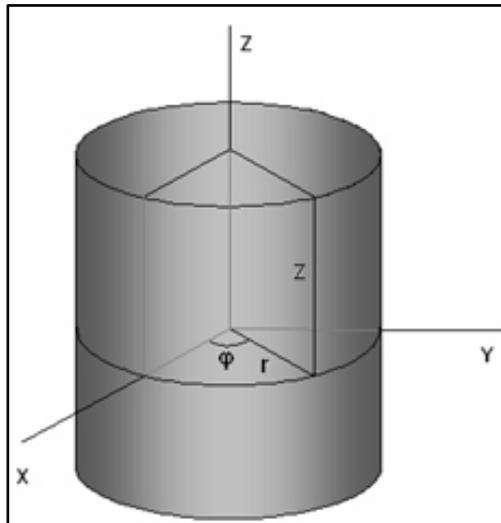


Figura 10. Sistema de coordenadas cilíndricas.²

² <http://portales.puj.edu.co/objetosdeaprendizaje/Online/OA04/Imagenes/Sistemas%20de%20coordenadas/09.jpg>

3.5. Dispositivos disponibles en el mercado

En el mercado se encuentran diferentes dispositivos que permiten realizar las mismas mediciones a las que apunta este proyecto. La mayoría de ellos no están disponibles en el país, y adquirirlas requiere una gran inversión de dinero.

Una de ellas es la Mitutoyo Crysta Apex, una CMM con resolución de hasta $1.7 \mu\text{m}$ por eje y altos rangos de medición, además de contar con un sistema de compensación de temperatura. Esta es la que se emplea en el laboratorio de metrología del INA, donde AARC-CR hace las mediciones del cilindro en la actualidad. Estas pueden llegar a costar hasta \$100,000, dependiendo del modelo seleccionado. Se pueden encontrar modelos similares usados, que rondan los desde los \$10,000 o más, tomando en cuenta que se trata de modelos de los años 1985 en adelante.

Otra opción en el mercado es el instrumento Mitutoyo RoundTest, que está diseñada específicamente para la medición de objetos cilíndricos. Los modelos encontrados no contaban con los requerimientos de dimensiones necesarias para medir el cilindro, además de no estar disponibles en el país. Igualmente, se encontró que su precio ronda los \$20,000 dólares con algunos años de uso, ya que este es un instrumento mucho más reciente.

Además, se encontraron investigaciones de medición de cilindros con instrumentos ópticos, pero estos necesitan convertidores óptico-digitales y demás, por lo que su uso es muy limitado, dado a los altos costos.

Las figuras 11 y 12, muestran los instrumentos Mitutoyo Crysta Apex y Mitutoyo RoundTest, respectivamente.

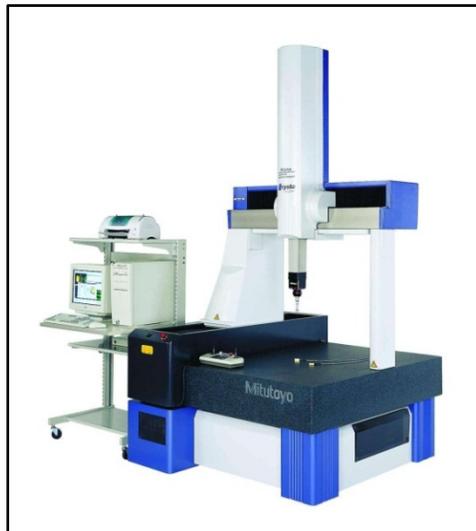


Figura 11. Mitutoyo Crysta Apex³

³ <http://cfnewsads.thomasnet.com/images/large/023/23165.jpg>



Figura 12. Mitutoyo RoundTest.⁴

⁴ http://www.inspection.ie/shopping_admin/images/prod_img/RA-400.jpg

Capítulo 4. Procedimiento metodológico

4.1 Reconocimiento y definición del problema

El problema se determinó luego de una entrevista con el Ing. Juan Ignacio Del Valle, empleado de AARC-CR, ya que la solución planteada llega a resolver muchos inconvenientes de su investigación doctoral.

La mejora del proceso de medición del cilindro interno del motor es muy importante, ya que permitiría determinar fallas en otros componentes del motor, como la antena helicoidal o las bobinas, con mayor rapidez. Igualmente, ayudaría a acelerar el proceso investigativo del doctorado del Ing. Del Valle.

Básicamente, la meta del proyecto es obtener una medición de la superficie interna del cilindro con rangos similares a los obtenidos por la CMM del Laboratorio de Metrología del INA. El requisito mínimo consistió en obtener mediciones que reflejaran correctamente la erosión interna del cilindro.

Como el objeto de estudio es un cilindro de dimensiones fijas, y estas no van a variar en un tiempo prolongado (2 años), se acordó que el módulo de medición y la aplicación de graficado solo deben estar diseñados para tomar los datos del cilindro utilizado.

Además, dado que la empresa cuenta con presupuesto reducido para este proyecto, se optó por utilizar la fresadora vertical CNC disponible en el taller de mecánica de precisión de la empresa. De esta manera, se aprovechó todo el sistema de control que tiene instalado y se reducen los costos de utilizar un sistema nuevo completo.

4.2 Obtención y análisis de información

Con las condiciones y especificaciones del proyecto establecidas, se inició por investigar varias alternativas de distintos problemas.

Inicialmente se investigó el modo de medición que utilizaría la fresadora, si este sería por contacto o sin contacto. Se descartó el modo de medición por contacto, ya que este involucra un estimado de la medición, así como regulación de la inercia de la máquina y otras variables que hacen más complejo el objetivo de controlar con un agente externo a la máquina. Las demás investigaciones respecto a los métodos de medición sin contacto, como sensores ultrasónicos, ópticos o láser, se hicieron utilizando medios de Internet, como hojas de datos, foros y demás fuentes de información.

Seguidamente se investigó como controlar de forma externa la fresadora. Para esto se consultaron los manuales disponibles en la web del fabricante, así como contactar al servicio técnico de la empresa para obtener datos relevantes al proyecto. Ellos

colaboraron al señalar las limitaciones del sistema, por su antigüedad, y otros detalles importantes del software en que se basa el control de la misma.

Finalmente, se investigaron diferentes módulos de control, para implementar el sistema encargado de controlar la fresadora, adquirir los datos provenientes del sensor y almacenarlos en el formato correcto. Además se tomó en cuenta el presupuesto de la empresa, que debía ser el menor posible.

Para todas las etapas de este proyecto se utilizó en su mayor parte la información de fuentes confiables de Internet, bases de datos académicas y manuales de fabricantes.

4.3 Evaluación de las alternativas y síntesis de la solución

Mediante una serie de pruebas de desplazamiento de la fresadora, se determinó que su recorrido máximo en el eje Z era de 120 milímetros, haciendo necesaria una extensión retráctil para el sensor y realizar el proceso de medición en dos etapas, ya que el cilindro a medir mide 226 milímetros.

En lo que respecta al sensor utilizado, se analizaron varias opciones, como lo son los sensores ópticos, ultrasónicos o láser. Mediante el estudio de hojas de datos y principios de operación, se descartaron los sensores ultrasónicos por tener un punto de medición muy amplio, ya que la onda incrementa su arco conforme aumenta la distancia a medir. Igualmente, los sensores ópticos no entregaban la resolución esperada ni en los rangos de medición necesarios. Por esta razón se optó por un sensor de distancia de corto alcance que utiliza el método de triangulación láser.

Ya que este proyecto solo contempla el diseño del sistema, se eligió simular el funcionamiento del sensor seleccionado mediante un microcontrolador, siguiendo el mismo flujo de datos y demás requerimientos.

En lo referente al control de la fresadora, se tenía la opción de manejarla mediante el puerto MPG y generar las mismas señales y seleccionar los ejes mediante conexiones a tierra, o utilizar un dispositivo USB-HID que actuara como un teclado y utilizara el modo MDI del sistema de control de la fresadora. Se decidió utilizar el modo MDI, dado que permite digitar los comandos de movimiento directamente con las coordenadas deseadas, asegurando la precisión máxima del sistema. Esta última opción fue sugerida por uno de los ingenieros de soporte de la empresa CENTROID, ya que era la única manera en que la fresadora podía ser controlada con precisión sin requerir retroalimentación de la posición.

Para el sistema de control, se seleccionó un Arduino Leonardo, el cual cuenta con un controlador USB que le permite actuar como un dispositivo HID, y cuenta con bibliotecas desarrolladas para actuar como un teclado completo. También se seleccionó un módulo Ethernet/SD Shield, el cual contiene el hardware para tarjetas microSD. Se incluyó un

módulo Real Time Clock (RTC) para guardar los archivos utilizando la fecha del día de la medición. Estos módulos fueron seleccionados dada su confiabilidad, bajo costo y facilidad de desarrollo, debido a que forman parte de una plataforma abierta de desarrollo de sistemas electrónicos, que cuenta con una comunidad muy participativa, facilitando el depurado de los proyectos desarrollados.

4.4 Implementación de la solución

Inicialmente, el trabajo se concentró en obtener control sobre la fresadora, por lo que se estudió la primera opción de manejarla mediante el puerto MPG. Este utiliza un encoder incremental rotativo, que tiene 2 líneas de pulsos desfasadas, así como la alimentación, tierra y varias señales que al ser conectadas a tierra seleccionan el eje a mover y el multiplicador de pulsos (x1, x10 y x100). Las señales de pulsos se generan basadas en el protocolo RS-422.

Dado el inconveniente de generar las señales con el desfase correcto, así como la incapacidad del control de la fresadora de proveer la posición actual mediante algún puerto, generó que se tomara la opción del dispositivo USB-HID.

Tomada esta decisión, se iniciaron pruebas de movimiento básico de la máquina, como mover el eje X o Z 1 milímetro, o variar en varios grados el eje A. Con estas pruebas se depuraron las funciones de movimiento de la máquina.

Seguidamente, se trabajó en la sección de almacenamiento de datos, corroborando que datos al azar se almacenaran en el formato correcto. Luego se agregó el RTC que crea el archivo con la fecha correcta.

Ya que el sensor debía ser simulado, se implementó y comprobó que ante la instrucción de entrada correcta, generara un dato al azar en un rango estimado posible, y lo enviara en el formato correcto.

Seguidamente, se agregó una pantalla Liquid Crystal Display (LCD) al sistema para alertar al usuario del progreso del programa, y de los pasos necesarios para la inicialización, calibración e inicio del sistema.

Para culminar con el módulo de medición de la fresadora, se integraron todas las secciones del programa y se hicieron las pruebas correspondientes con todo el sistema.

Finalmente, se creó la herramienta de graficado de datos. Esa viene a ser un apoyo básico para el usuario, sin llegar a sustituir a un software matemático como MATLAB. Esta aplicación se desarrolló con la ayuda del profesor Dr. Franklin Hernández Castro, de la Escuela de Diseño Industrial del ITCR.

Para todas las secciones mencionadas, se hizo una serie de pruebas, dependiendo de la función, utilizando mensajes desplegados en el monitor serie para determinar el

funcionamiento correcto y poder depurar la programación de cada módulo. Se corrigieron los errores encontrados y se procedió a ejecutar las pruebas finales.

4.5 Revaluación y rediseño

Dado que este es un diseño del cual se implementaron y probaron solo ciertas secciones, es imperativo el uso del sensor recomendado para depurar el sistema completamente.

De igual forma, de encontrarse una mejor opción en cuanto al sensor, deben hacerse las correcciones pertinentes al sistema, par que este lo reconozca correctamente y la medición mantenga la mayor confiabilidad posible.

También se debe analizar el funcionamiento de la aplicación gráfica, en caso de que se pueda optimizar o mejorar la forma de interpretar los datos, ya sea por secciones del cilindro u otra opción que permita utilizar de mejor manera los recursos de la computadora utilizada.

El sistema debe ser probado extensivamente para determinar algún error de programación en las rutinas de calibración, ya que estas fueron programadas de forma lógica, pero existe la posibilidad de que se presenten acciones inesperadas, ya que no se contó con el sensor para realizar la depuración correspondiente del sistema.

Capítulo 5. Descripción de la solución

5.1 Análisis de soluciones y selección final

Como se mencionó en las secciones anteriores, se utilizó una fresadora vertical CNC, la cual, por limitaciones de software de su sistema de control, solo puede ser controlada por un dispositivo USB-HID actuando como teclado.

Por facilidad y confiabilidad se decidió utilizar el dispositivo HID, ya que es mucho más fácil de implementar, la respuesta de la maquinaria es inmediata y no se requiere retroalimentación de la posición actual, ya que el mismo sistema cuenta con su propio sistema de control.

Se eligió utilizar un sensor láser de corto rango, de marca y modelo Riftek RF-603, ya que este cuenta con los requerimientos de resolución, frecuencia de muestreo, rangos de medición, dimensiones y método de comunicación. Además, era el más económico de los disponibles en el mercado, según las cotizaciones a diferentes empresas.

Por las limitaciones presupuestarias, se empleó la plataforma Arduino UNO para simular el funcionamiento del sensor y enviar los datos al sistema de control.

Igualmente, el sistema de control se implementó en la plataforma Arduino Leonardo, por contar con reconocimiento automático como HID con solo una instrucción, haciendo el diseño e implementación del sistema mucho más sencilla. Además, cuenta con múltiples facilidades para utilizar módulos como el LCD, RTC, comunicación serial y almacenamiento en tarjetas SD.

Ambas plataformas Arduino son de muy bajo costo y tienen un gran soporte en la comunidad de software libre, haciendo posible solucionar muchos problemas al depurar o agregar funciones al sistema en el caso de una mejora o adición al sistema.

Finalmente, para el programa graficador de datos, se utilizó el lenguaje de programación y entorno Processing, el cual es una herramienta de diseño gráfico muy versátil, fácil de usar y con mucho soporte en línea. Además, se contó con el apoyo de la Escuela de Diseño Industrial del ITCR para realizar la aplicación.

En general, se seleccionaron los componentes que proveyeran la mayor confiabilidad, facilidad de implementación, apego a los requerimientos y bajo costo.

5.2 Descripción del hardware

5.2.1. Sistema de sujeción del sensor de medición y el cilindro

El cilindro a medir debe estar sujeto a la base rotativa de la fresadora vertical, para asegurar que no se mueva y se el orden de los datos, llevando a un resultado erróneo. Por esta razón, se diseñó una base que sujeta el cilindro de forma confiable, y además, aprovechando que el cilindro tiene una muesca para marca la posición de inicio de las mediciones, se incluyó un pin para asegurar que la posición en la que se inserta el cilindro siempre sea la misma para cada medición.

La figura 13 muestra el diseño de la pieza necesaria para anclar el cilindro al módulo del eje A (Figura 6). Esta base debe estar atornillada a la base del eje A.

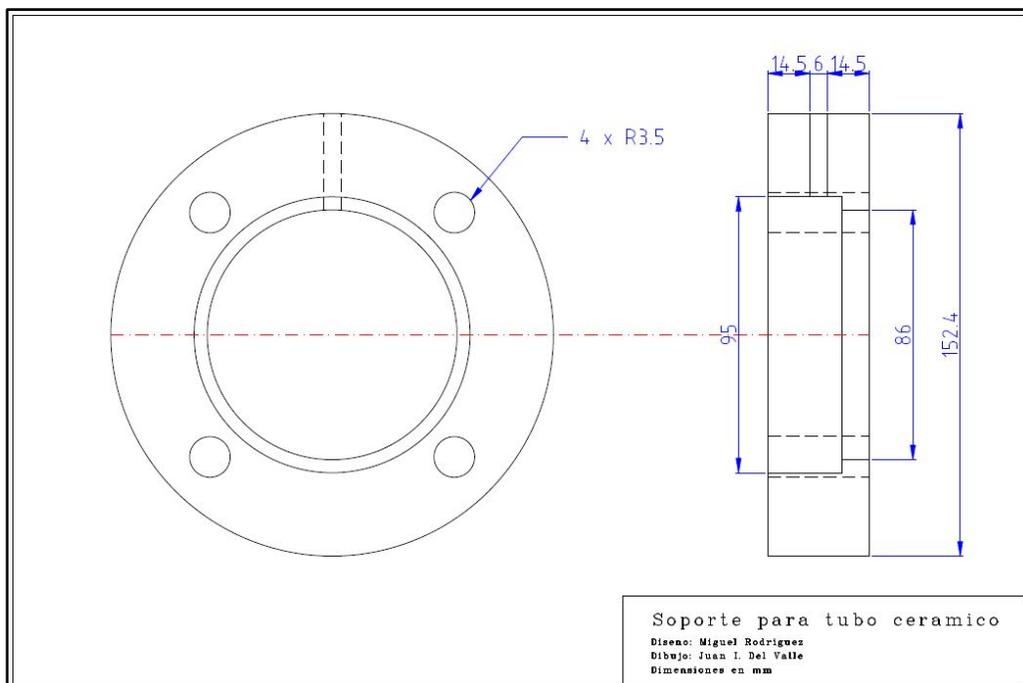


Figura 13. Pieza de sujeción del cilindro con el eje A.

Como se mencionó en las secciones anteriores, dado que el rango de movimiento del eje Z es de 120 mm y es necesario un mínimo de 226mm, se diseñó un émbolo con 2 posiciones fijas, una en posición mínima y otra la extendida, tomando en cuenta que la extendida añade 120mm al brazo del eje Z. Esto permite al usuario medir 120mm de la

parte baja de cilindro, luego ajustar el brazo del sensor a la posición base e iniciar la medición de nuevo, midiendo la sección siguiente del cilindro.

Las figuras 14 y 15 muestran los modelos interno y externo de las piezas del émbolo, respectivamente. El sensor esta acoplado en la parte inferior de la pieza interna, y la parte superior de la pieza exterior esta acoplada al eje Z de la fresadora. Se debe tomar en cuenta que la sección para el acople del sensor debe ser revisada al tener el aparato, ya que este diseño se basó en las dimensiones detalladas en la hoja de datos de dicho sensor. [4]

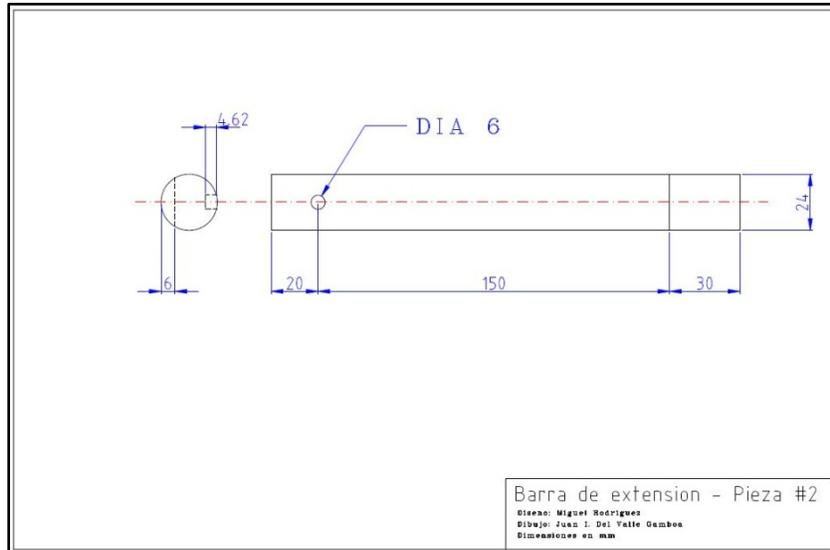


Figura 14. Pin interno del émbolo (Sujeción de sensor)

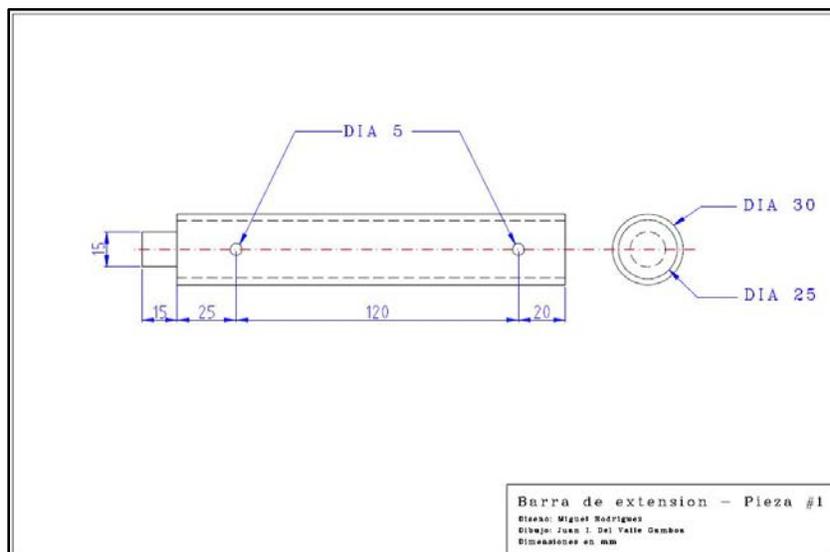


Figura 15. Parte externa del émbolo (Sujeción a eje Z)

5.2.2. Sistema para la calibración de la posición de inicio

Para calibrar la posición inicial de la medición, se diseñó una sección del cilindro con un corte vertical situado en la misma posición que el pin de anclaje del cilindro. Esta sección del cilindro consiste en un arco de metal con las mismas dimensiones de espesor y radio interno que el cilindro a medir.

La idea de este sistema es que el sensor primero se acerque hacia el cilindro hasta que lo detecta en su rango de medición, y luego gira el eje A hasta encontrar el corte vertical (pérdida de objeto en el rango), que indica la posición de inicio.

La figura 16 muestra el diseño de dicha pieza.

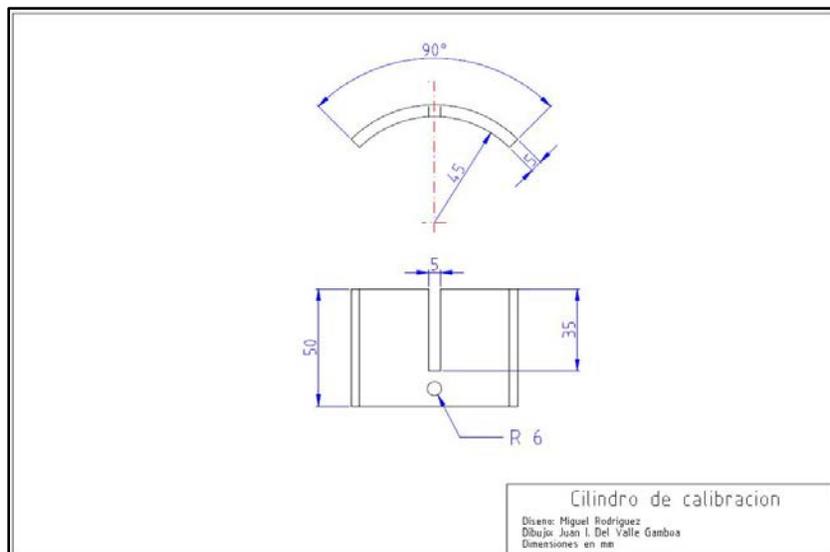


Figura 16. Pieza de calibración de cercanía del sensor y ángulo de inicio.

5.2.3. Pantalla LCD

Este módulo utiliza el controlador Hitachi HD44780, ampliamente utilizado en los sistemas LCD sencillos, comúnmente de 16x2 y 16x4. Utiliza un sistema paralelo, que puede ser de 8 bits o 4 bits que alternan entre los MSB y LSB. En este caso se utilizó el esquema de 4 bits para reducir el cableado. También se agregaron las señales de Enable y R/W, así como un potenciómetro para regular la intensidad de la luz de fondo. [6]

Las conexiones se realizaron siguiendo el esquemático de la figura 17.

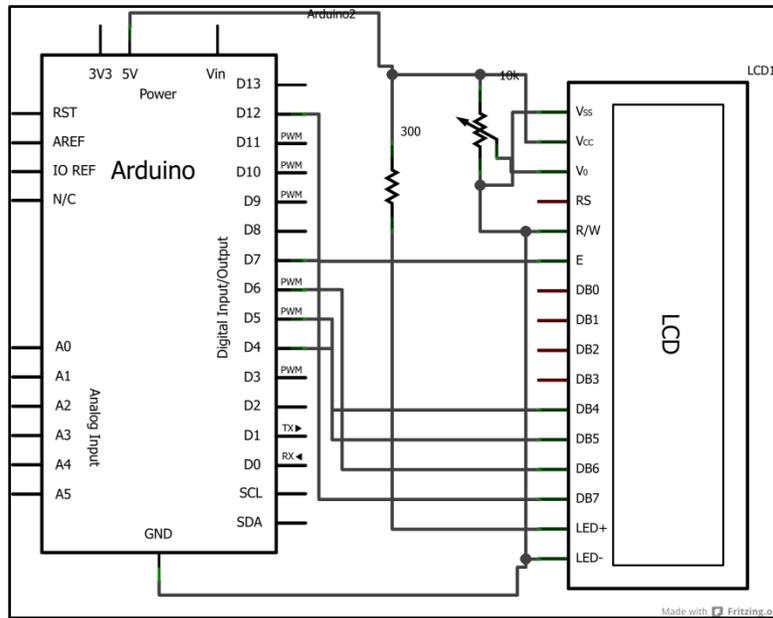


Figura 17. Esquemático de conexión de LCD.

5.2.4. Módulo RTC

Este módulo consta de un chip DS1307, un cristal de 32,768Hz y una batería de 3V CR1225 41mAh con una duración mínima de 9 años, con una variación temporal no significativa para la aplicación de este proyecto. También cuenta con un pin de alimentación de 5V y el mismo chip se encarga de seleccionar la fuente de alimentación, para conservar la batería.

Fue módulo adquirido en la compañía SparkFun, con el código BOB-00099 y su esquemático esta disponible en línea. Este módulo se puede apreciar en la figura 18, así como su esquemático de conexión en la figura 19. [7]

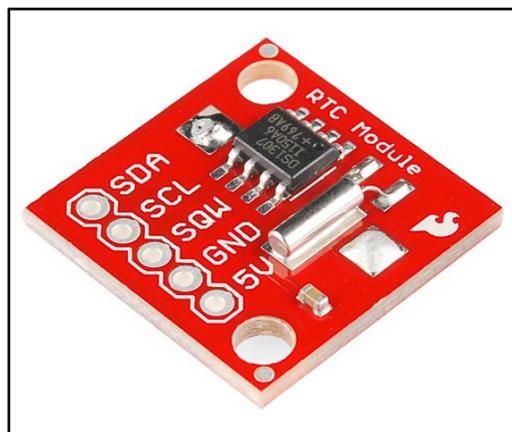


Figura 18. Módulo RTC con chip DS1307.⁵

⁵ <https://dlnmh9ip6v2uc.cloudfront.net/images/products/9/9/00099-02.jpg>

Utiliza el protocolo de comunicación I2C, la cual solo requiere la conexión directa de las señales “Serial Clock” (SCL) y “Serial Data” (SDA) con los pines del sistema del mismo protocolo. Estas pueden ser conectadas el en Arduino Leonardo a los pines marcados SDA y SCL.

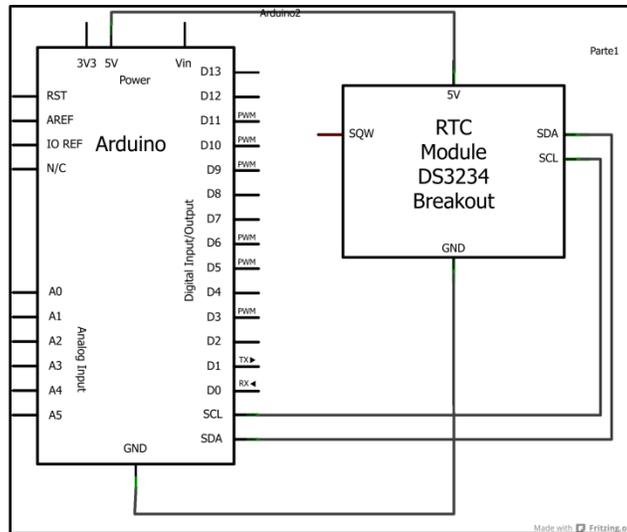


Figura 19. Esquemático de conexión de RTC.

5.2.5. Sensor Riftek RF-603

El sensor laser Riftek Rf-603 es un sensor laser de corto alcance. El modelo seleccionado cuenta con una distancia base de 10 milímetros y un rango de medición de 5 milímetros. Cuenta con una señal AL que indica si el sensor presencia algún objeto en su rango de medición, por lo que esta señal será utilizada durante todo el ciclo de medición. También cuenta con comunicación serial utilizando el protocolo RS-232.

Permite obtener datos mediante un flujo constante de mediciones o por polling. En este caso se utilizara polling, enviándole los bytes 0x01 y 0x86 desde el dispositivo principal.

El sensor toma la medición, y el número generado por el la matriz receptora se divide en 4 bytes. Estos son enviados utilizando el puerto serial.

Los demás detalles sobre el funcionamiento de este sensor, se pueden encontrar en la hoja de datos. [4]

La figura 20 muestra el sensor Riftek Rf-603.



Figura 20. Sensor Riftek RF-603.

5.2.6. Simulador de sensor Riftek RF-603

El sensor recomendado es un sensor de medición láser de corto rango, de marca Riftek y modelo RF-603, con una distancia base de 10 milímetros y una distancia de medición (rango) de 5 milímetros. Además cuenta con comunicación serial mediante el protocolo RS-232 y una señal AL que esta programada para indicar si el sensor tiene algún objeto en el rango de medición.

Para este módulo de simulación se utilizó un Arduino UNO R3. Este módulo no tiene ninguna modificación, y solo debe de ser conectadas las líneas TX/RX y RX/TX con el módulo de control, respectivamente. La figura 21 muestra la conexión necesaria.

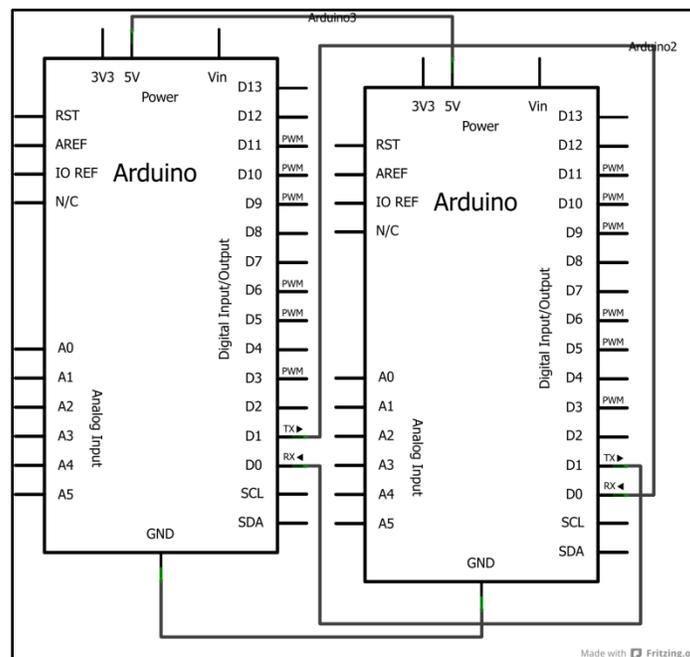


Figura 21. Esquemático de conexión de comunicación serie.

El sensor real, contiene una señal AL que indica si detecta un objeto en su rango de medición. Esto se simuló mediante un botón para realizar las pruebas respectivas, pero debe ser conectado al módulo principal.

Además, la figura 22 detalla las conexiones necesarias para convertir una señal Rs-232 a una señal serie TTL, ya que esta etapa será necesaria al añadir el sensor real. Esto es porque los niveles de voltaje del protocolo Rs-232 varían entre +15V y -15V, para representar un 0 y un 1 lógico, respectivamente. Las señales serie TTL corresponden a 0V y 5V para representar un 0 y un 1 lógico. Para esta modificación se debe utilizar un circuito integrado MAX-232. [8]

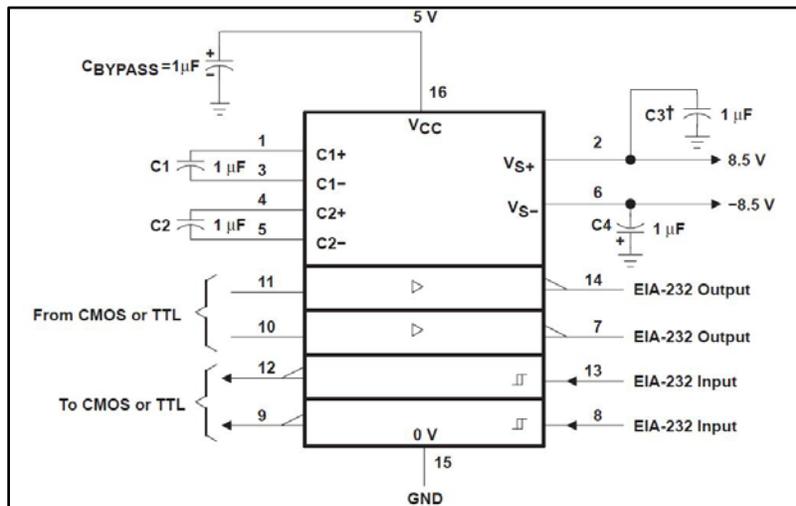


Figura 22. Esquemático de conexión de MAX-232.

5.2.7. Almacenamiento en tarjeta microSD

Para almacenar los datos en una tarjeta microSD, se utilizó el Arduino Ethernet /SD Shield. Este cuenta con un conector RJ45 para conectarlo a una red y un socket para una tarjeta microSD. Este módulo se conecta directamente al Arduino Leonardo y ya tiene todas las conexiones necesarias para la transmisión de datos y pines de selección (Chip Select). La figura 23 muestra dicho dispositivo.

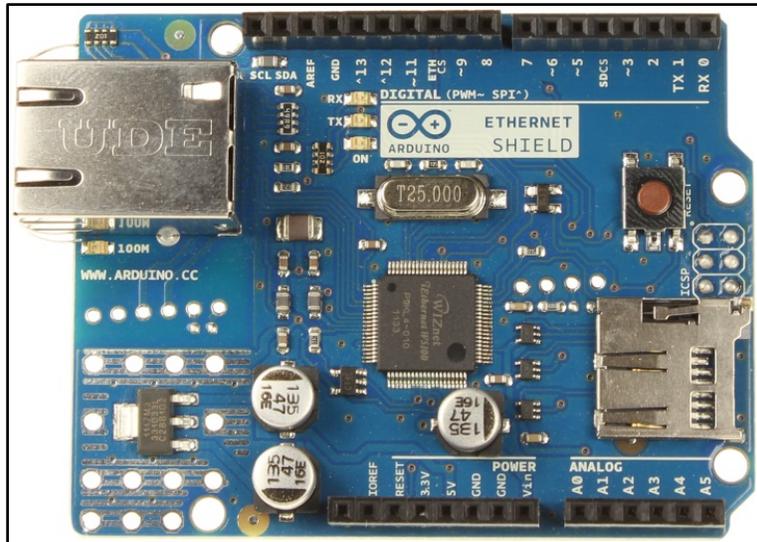


Figura 23. Arduino Ethernet Shield con SD.

Para utilizar este modulo de hardware solo se utilizaron algunas funciones de la biblioteca SdFat desarrollada para Arduino, por lo que no se detalla un módulo específico desarrollado para el almacenamiento de datos.

5.2.8. Plataforma principal

Como plataforma central del sistema, se utilizó un Arduino Leonardo. A este dispositivo se conectan el RTC, LCD y Ethernet Shield. Además, se debe conectar el puerto micro USB al puerto USB de la fresadora vertical CNC, para que sea capaz de actuar como un dispositivo USB-HID. La figura 24 muestra el Arduino Leonardo.

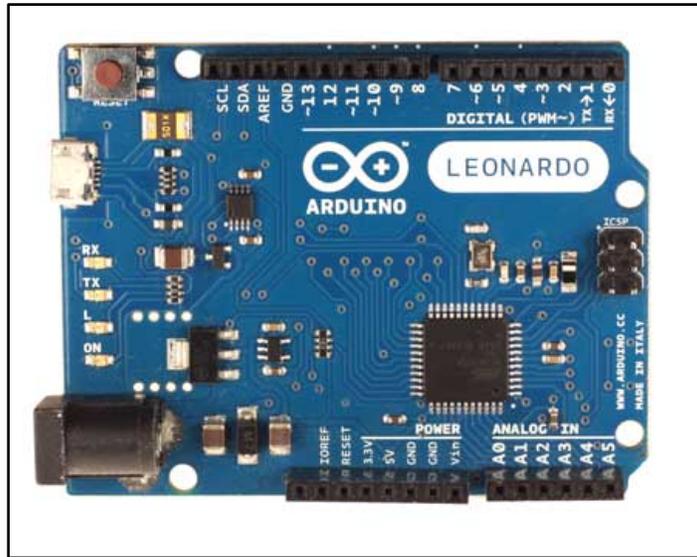


Figura 24. Arduino Leonardo.

La figura 25 detalla la conexión completa del sistema.

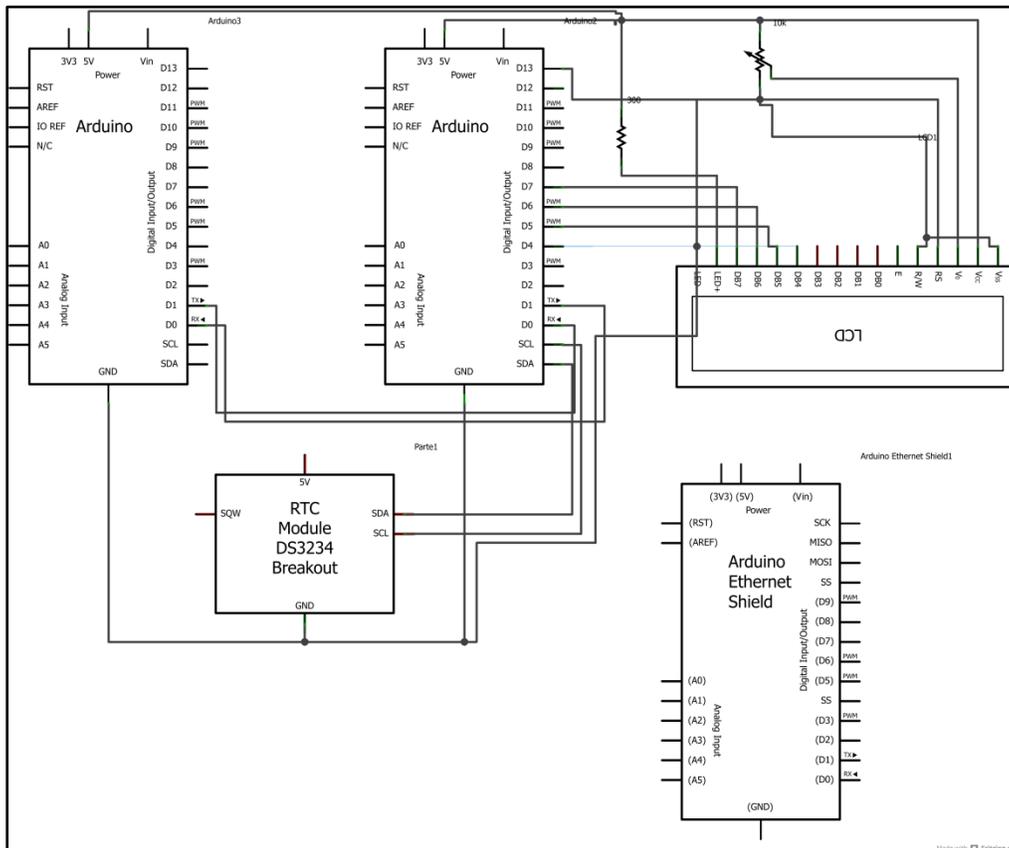


Figura 25. Esquemático de conexión del circuito completo.

5.3 Descripción del software

En esta sección se detallan cada uno de los módulos programados, con su respectiva explicación de funcionamiento y uso.

5.3.1. Módulos de calibración

Estas secuencias de software ocurren al inicio del programa, sus funciones son acercar el sensor al cilindro, para que pueda realizar las mediciones correctamente, y encontrar el ángulo correcto de inicio, que coincida con el pin de anclaje del cilindro, para mantener el mismo patrón de medición.

Para que ambas funciones resulten en un funcionamiento correcto, se debe tener el cilindro de calibración en la base del cilindro, esto para que se pueda obtener el punto inicial de la medición utilizando el corte vertical del mismo. Esta es la razón de que el sistema espera a que el usuario haya activado el switch de calibración, indicando que el cilindro de calibración está colocado y que el sensor se ubica en el centro de la base.

Esto se debe hacer de forma manual ya que el ajuste de la altura de la mesa de trabajo de la fresadora es manual y se reduce el tiempo de calibración al hacerlo de forma manual. Además se debe considerar que al no contar con el sensor real, la implementación de un método de calibración completo se complica, debido a las consideraciones de dimensiones del sensor.

5.3.1.1. Cercanía al cilindro

Esta rutina consiste en un ciclo que disminuye la posición del eje X en 2 milímetros, mientras la condición de no tener el cilindro en el rango de medición no se cumpla, utilizando la señal AL mencionada previamente. Esto es factible dado que el sensor recomendado tiene un rango de medición de 5 milímetros.

Al ser llamada la función, revisa el estado del pin de entrada de la señal AL, esto para saber si el sensor tiene el objeto en el rango de medición. Si no es así, inicializa el teclado virtual, entra al modo MDI del control de la fresadora, digita el código "G00 X[#5041-2]", presiona Enter y finalmente Ctrl+S, para iniciar el ciclo de movimiento. Por cada ejecución de ciclo, se aumenta una variable en 2 unidades, para añadir este acercamiento al radio real del cilindro.

El código G00 indica posicionamiento rápido, X indica el eje a desplazar y [#5041-2] toma el valor del registro 5041 y le resta 2, resultando en la posición actual del eje X menos 2. Esto funciona correctamente dado que los valores de las variables de posición están guardados en milímetros.

La figura 26 indica el diagrama de flujo de esta función.

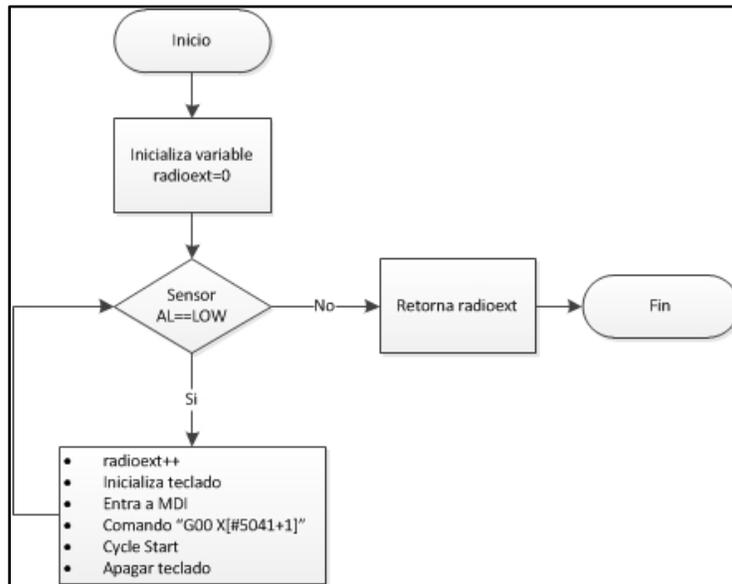


Figura 26. Diagrama de flujo de acercamiento al cilindro.

5.3.1.2. Ángulo de inicio

Ya que el sensor se encuentra posicionado con la pared del cilindro en su rango de medición, señal AL debe indicarlo. Seguidamente se gira el cilindro hasta que el sensor detecte el canal vertical, resultando en un cambio en la señal AL.

El giro del cilindro se hace mediante el comando “G00 A[#5044+1]”, esto gira el cilindro en la dirección contraria a las manecillas del reloj (CCW). Esta instrucción se incluye en un ciclo que revisa si el sensor mantiene el objeto en el rango de medición, si se mantiene, hace un giro de un grado, sino, termina el ciclo y la función, dado que ya se encontró la posición inicial indicada por el canal vertical en el cilindro de calibración.

El diagrama de flujo de esta función se puede observar en la figura 27.

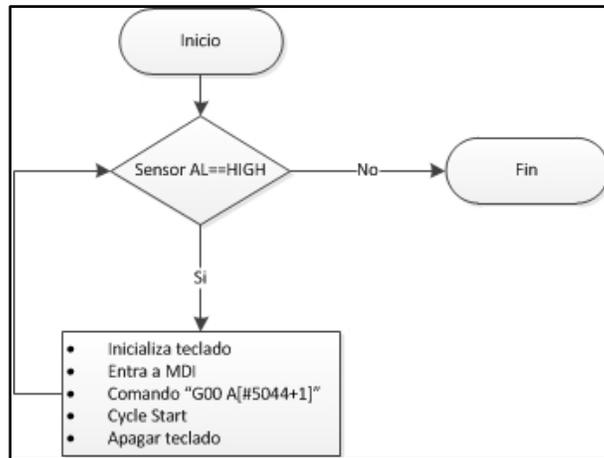


Figura 27. Diagrama de flujo de movimiento al ángulo de inicio.

5.3.2. Simulación del sensor Riftek RF-603

Se encarga de simular el comportamiento del sensor Riftek RF-603. Si recibe los bytes 0x01 y 0x86, debe generar un número al azar entre 7864 y 8520, que indican una medición entre 2.4 mm y 2.6 mm.

El resultado consta de 2 bytes o 4 nibbles. Cada uno de estos nibbles se debe concatenar con 0xB como MSB, ya que esto indica el funcionamiento del sensor y no es relevante para la medición, ya que no se está dando un flujo de datos constante. Este funcionamiento se explica con detalle en la sección 5.2.5.

Esto crea 4 bits, que se envían de LSB a MSB, para que el sistema de control principal obtenga el valor generado de manera correcta.

LA figura 28 muestra el diagrama de flujo de este simulador.

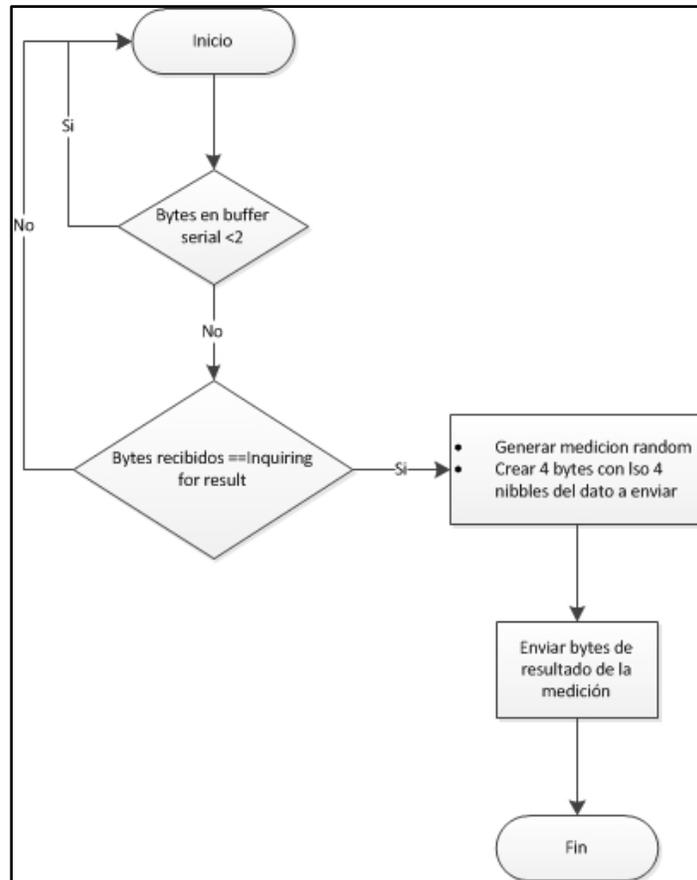


Figura 28. Diagrama de flujo de simulador de sensor.

5.3.3. Módulo de medición

Este módulo se encarga de obtener el dato medido por el sensor. Debe enviar los comandos 0x01 y 0x86, hexadecimal, los cuales equivalen a “Inquiring for result” según la hoja de datos del sensor. Esto genera la recepción de 4 bytes, a los que se les debe eliminar los 4 bits más significativos, y concatenarlos para obtener el resultado de la medición. [4]

Ya que el resultado equivale a un número entre 7864 y 8250, se debe realizar la conversión mediante regla de tres para obtener la distancia medida. Esta función retorna un número flotante con este último valor.

El diagrama de flujo de esta función se detalla en la figura 29.

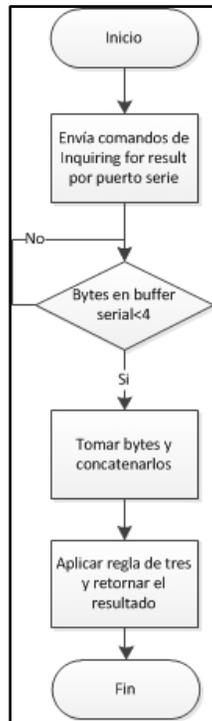


Figura 29. Diagrama de flujo de toma de datos del sensor.

5.3.4. Módulo de obtención de datos de RTC

El módulo RTC ya tiene la fecha de Costa Rica precargada, (UTC -6:00). Por esta razón, solo es necesario tomar la fecha del dispositivo. Este inicia la comunicación por el puerto I2C, con la dirección correspondiente, y toma los datos de segundos, minutos, hora, día del mes, día de la semana, mes y año. Luego, se toman los valores de día del mes, mes y año y se agregan a un arreglo de caracteres con la terminación “.txt”, para crear el nombre del archivo a crear y escribir los datos tomados por el sensor.

La figura 30 muestra el diagrama de flujo de esta sección del programa.

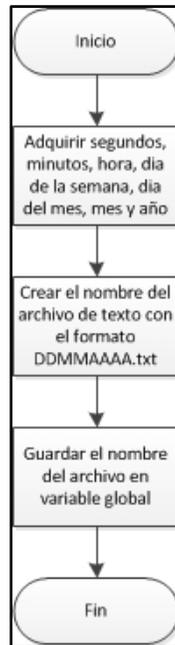


Figura 30. Diagrama de flujo de toma de datos de RTC.

5.3.5. Flujo completo de medición

El entorno Arduino IDE posee dos bloques básicos, `setup()` y `loop()`. El bloque `setup()` se ejecuta solo una vez al inicio del programa, luego el bloque `loop()` se ejecuta constantemente.

En este caso, se utilizó el bloque `setup()` para inicializar los puertos serie, LCD, memoria SD, adquirir la fecha desde el RTC y ejecutar las rutinas de calibración, mediante la intervención del usuario.

Seguidamente, en el bloque `loop()`, se ejecutan dos ciclos anidados. El ciclo interior se enfoca en la medición del cilindro y el almacenamiento del dato cada 4 grados hasta terminar la circunferencia completa. El ciclo externo se encarga de aumentar la altura en el eje Z, y de cambiar la línea de escritura en el archivo. La posición de medición y altura es constantemente mostrada en el LCD, para que usuario este consciente del progres de la medición.

Al finalizar ambos ciclos se indica al usuario mediante el LCD y se cierra el archivo.

La figura 31 muestra el diagrama de flujo de proceso de inicialización, y la figura 32 el diagrama de flujo del proceso de medición.

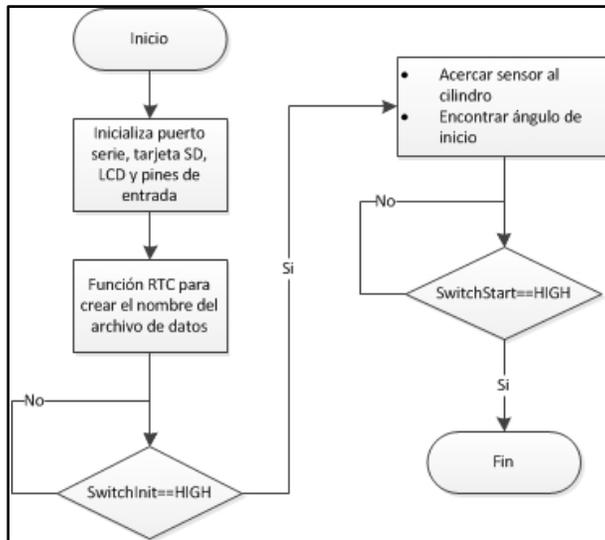


Figura 31. Diagrama de flujo de proceso de inicialización.

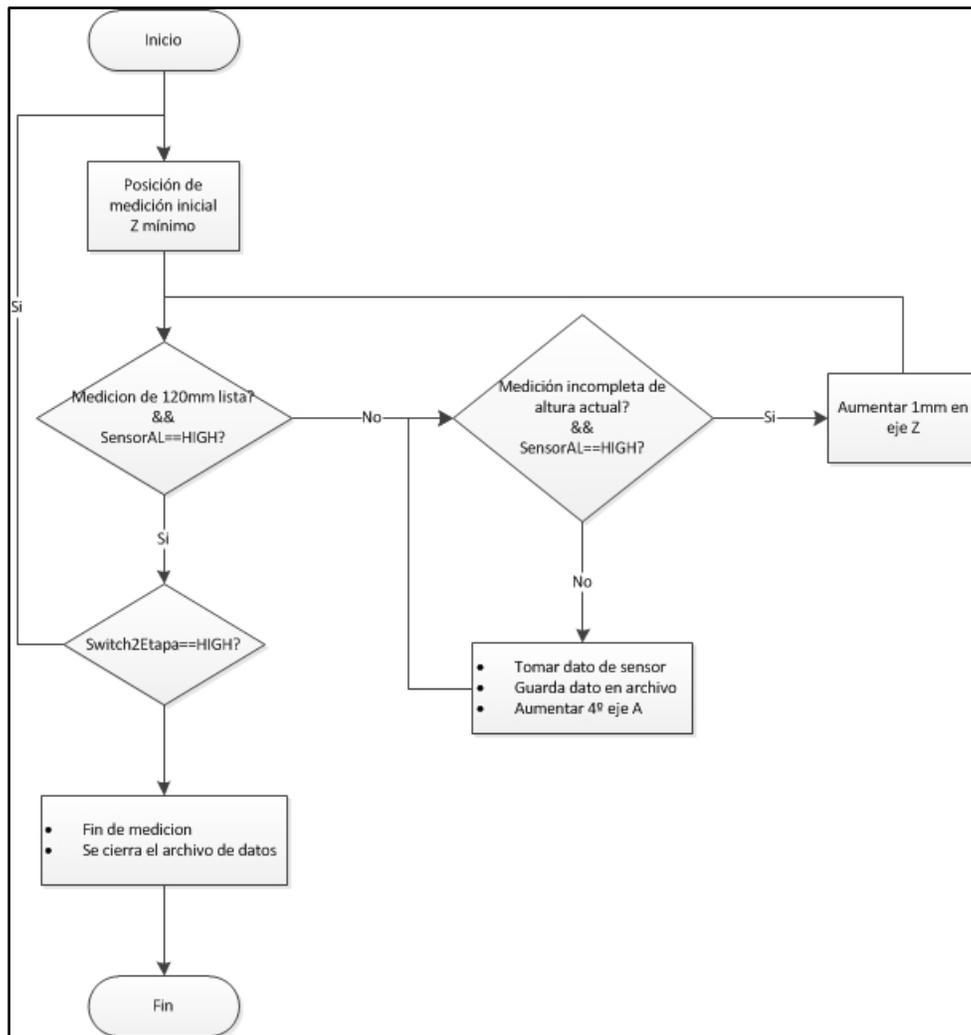


Figura 32. Diagrama de flujo de proceso completo de medición.

5.3.6. Graficado de datos

El programa de graficado de datos consta de una carpeta llamada “data”, donde se encuentra el archivo “.txt” a graficar. Por la simplicidad de la programación, es necesario tomar el archivo de la tarjeta micro SD del sistema de medición, y guardar el archivo de texto de interés bajo el nombre “datos.txt”.

El programa se encarga de leer el archivo “datos.txt” de la carpeta data, y guardarlo en un arreglo bidimensional. En este caso la tridimensionalidad esta dada por coordenadas cilíndricas, donde la altura esta dada por la fila utilizada, el número de columna indica el ángulo y el valor guardado en la posición [fila][columna] seleccionada indica el radio.

De esta forma, se hace el calculo de las variables “x” y “y” del sistema. Con estas variables se hace el traslado del eje de coordenadas para poder graficar un box(), que es un cubo graficado en la posición deseada.

Igualmente, se hace un mapeo utilizando el valor del radio y se compara con un rango máximo y mínimo, para obtener una representación en colores del desgaste que presenta el cilindro. El color azul representa un mínimo desgaste, y el color rojo representa un alto desgaste.

Finalmente, el sistema dibuja una esfera amarilla en el punto Z final, para mantener la interpretación correcta de las zonas de desgaste del cilindro.

También cuenta con una cámara, que permite mover el cilindro en cualquier dirección o aumentar o disminuir el cilindro, permitiendo diferentes vistas del lado interno y externo de cilindro. Para esto se utilizó la biblioteca PeasyCam.

El diagrama de flujo del programa se presenta en la figura 33.

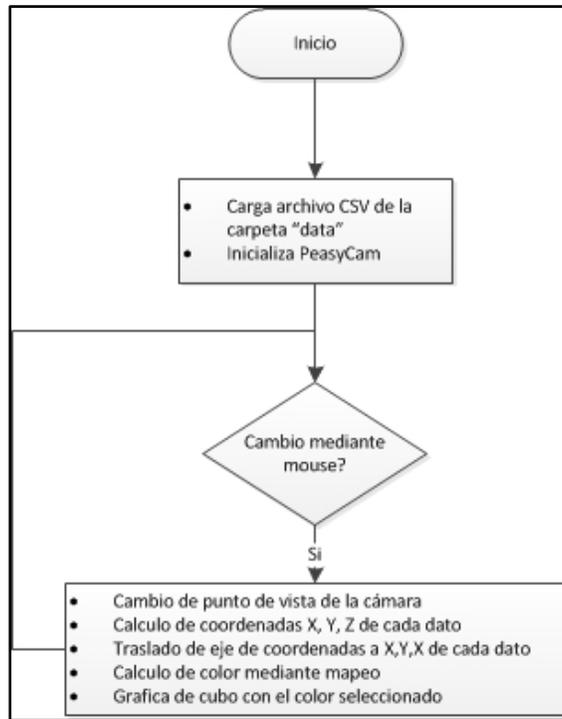


Figura 33. Diagrama de flujo del graficador de datos.

Las figuras 34 y 35 muestran diferentes vistas conseguidas por el programa.

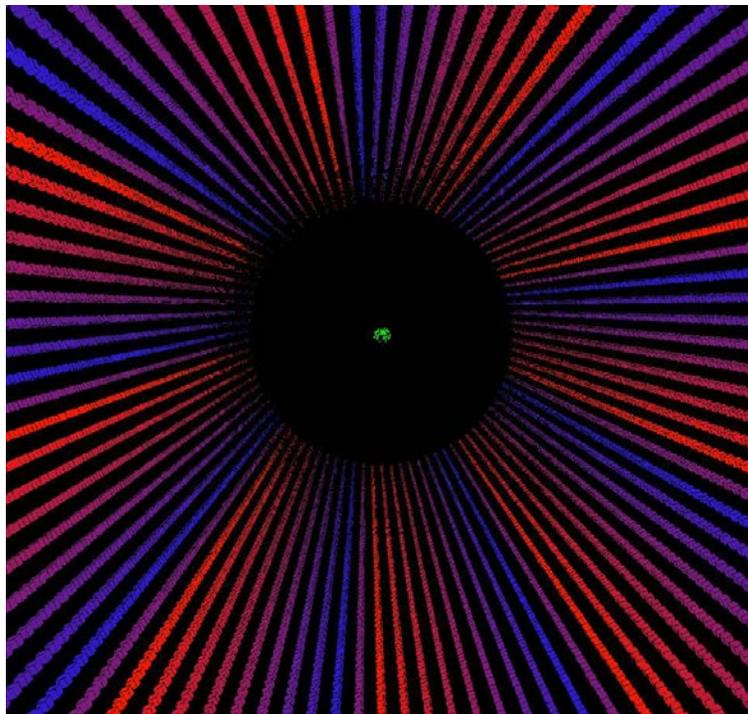


Figura 34. Vista interna del cilindro graficado.

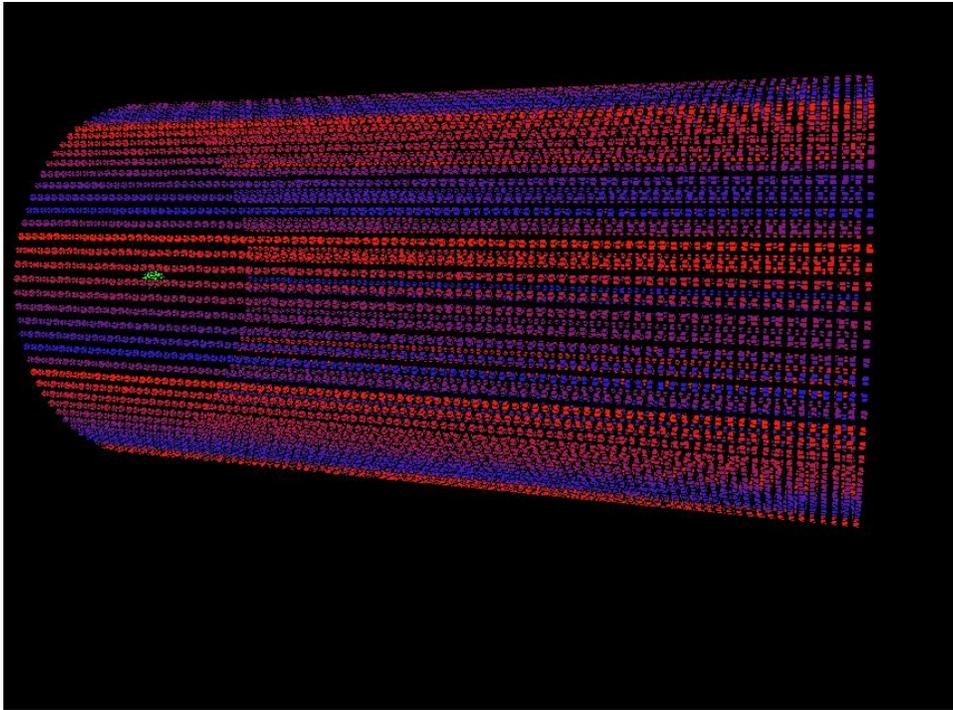


Figura 35. Vista externa del cilindro graficado.

Capítulo 6. Análisis de resultados

6.1. Resultados

En primera instancia, se consiguió obtener control de la maquinaria haciendo uso del modo MDI en el control de la fresadora CNC, permitiendo mover cualquier eje de la maquinaria a cualquier punto en el rango permitido.

Luego de la implementación de todos los módulos, se realizaron diferentes pruebas, observando el movimiento de la fresadora vertical CNC y corroborando las mediciones tomadas por el sistema. Luego de esto, se comprobó que el sistema de almacenamiento de datos funcionó correctamente, de manera que se obtuvo un archivo como el presentado en la figura 36.

Estos dos resultados indican el cumplimiento de los dos primeros objetivos específicos, ya que se logró adaptar una fresadora para que realizara mediciones, y se creó el sistema de almacenamiento de datos de forma correcta.

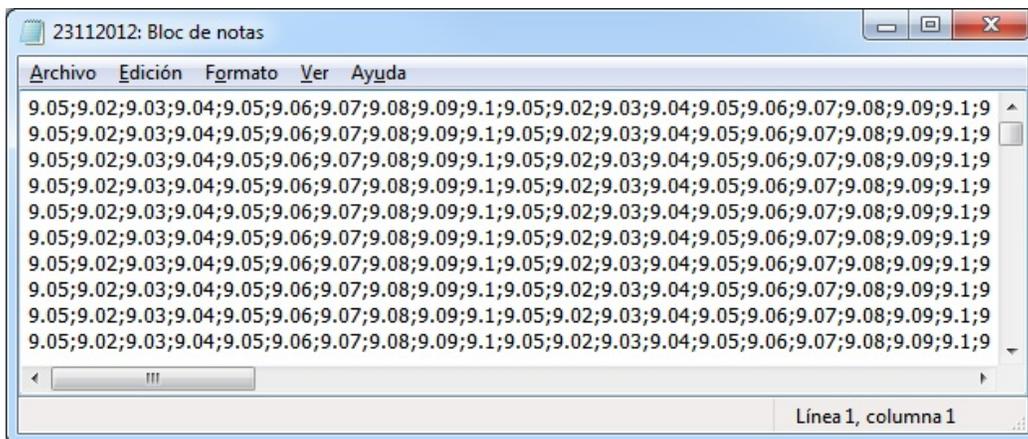


Figura 36. Captura del archivo de datos creado.

Dado que se obtuvo el archivo de datos de forma correcta, se comprobó el funcionamiento del sistema de graficado de datos, resultando en la figura 37.

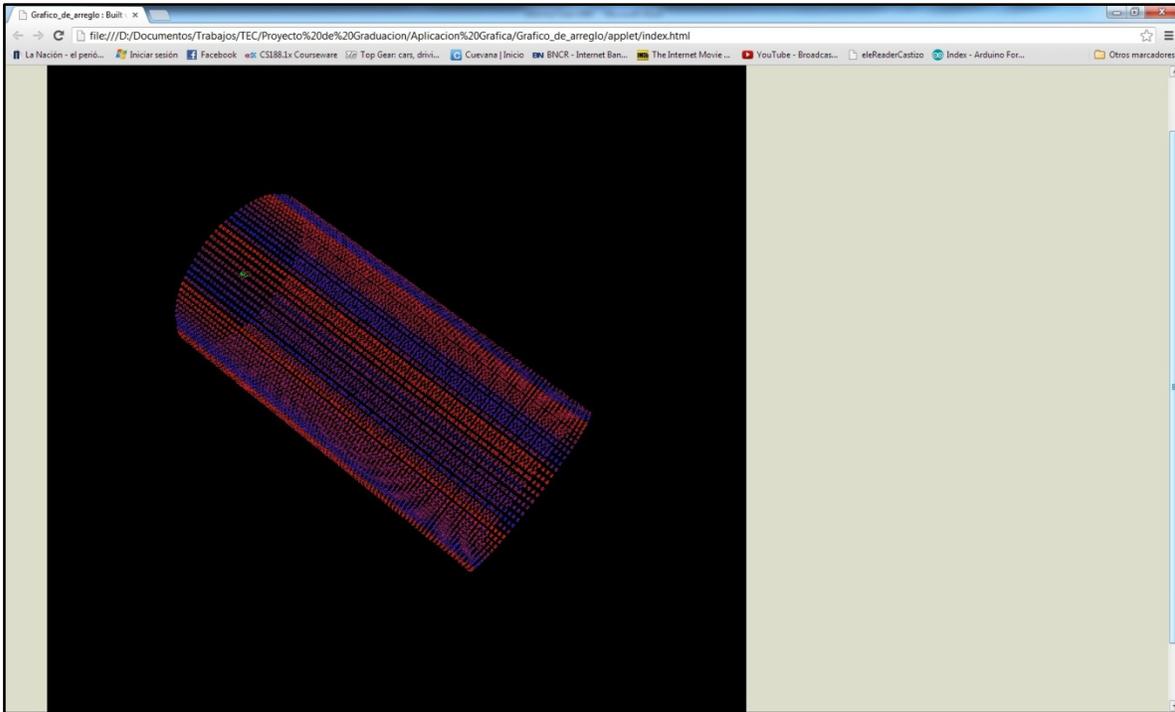


Figura 37. Applet del programa graficador.

Este último resultado comprueba el cumplimiento del último objetivo específico, al demostrar que se logró crear una herramienta sencilla para graficar los datos medidos y representar el nivel de erosión del cilindro.

Tomando en cuenta los posibles gastos de envío de todos los componentes necesarios, se estima que este proyecto tiene un costo total aproximado de \$1300, incluyendo el sensor Riftek RF-603 y demás dispositivos discutidos en la sección de hardware.

6.2. Análisis

En primera instancia, se debe reconocer que el movimiento de la máquina está limitado y por ende el proceso de medición se debe hacer en dos etapas.

Mediante diversas pruebas, se determinó que el movimiento de 1mm en el eje Z dura aproximadamente 250ms, mientras que el movimiento de 4 grados en el eje A puede durar alrededor de 4 segundos. De esta forma, una medición por líneas verticales en lugar de circunferencias resultaría en una medición mucho más rápida.

Con el método de medición implementado la duración de la medición, tomando datos cada 4 grados en cada circunferencia y cada milímetro de altura, se pueden durar aproximadamente 22 horas. Si se hicieran líneas verticales, como la cantidad de variaciones en el eje A es mínima, la duración estimada es de 5 horas y media.

Esta duración, aunque es prolongada, mantiene las facilidades para AARC-CR de contar con el sistema de medición en sus instalaciones, facilitando el flujo constante de trabajo.

Esto no se puede implementar en esta fresadora dado que el rango de medición en el eje Z es de 120mm, mucho menor que la longitud del cilindro utilizado. Por esta razón es que se utiliza el émbolo con dos posiciones fijas en el diseño.

El método de utilizar el modo MDI para controlar la máquina es muy efectivo, ya que no se necesita un sistema de retroalimentación, dado que este está integrado al sistema controlador de la fresadora CNC. Además, provee la utilización de todas las instrucciones de G-Code, que pueden facilitar la implementación de futuras mejoras o cambios en el sistema.

Algo muy importante es tener en cuenta los tiempos que le toma a la fresadora realizar ciertos movimientos, así como la entrada al modo MDI.

El sistema de almacenamiento de datos, según las pruebas realizadas, funcionó correctamente en todas las ocasiones, y el archivo se generó de la misma manera, con el formato correcto.

De la misma manera, el sistema simulador del sensor y su recepción de datos funcionó sin fallas, otorgando una estimación precisa de como funcionaría el sistema en caso de contar con el sensor recomendado.

En general el sistema funcionó acorde a lo esperado. Las variaciones en las posiciones de los ejes tuvieron diferencias de máximo 3 micrómetros de la posición esperada, garantizando una correcta operación.

Respecto a la aplicación gráfica, se implementó de forma muy básica, ya que no se tenía mucha experiencia con dicha herramienta. Aunque se deben realizar ciertas operaciones antes de utilizarla, como guardar el archivo en la carpeta correcta y cambiar su nombre, la aplicación funcionó correctamente y provee al usuario con una representación sencilla pero muy intuitiva de los niveles de la erosión y su respectiva posición en la superficie interna del cilindro.

Al hacer una comparación de los costos entre maquinaria de medición y el sistema diseñado, se puede notar claramente una reducción significativa de la inversión monetaria necesaria. Esto hace la implementación completa del proyecto más factible, incrementando las posibilidades de realizar las mediciones del cilindro en las instalaciones del AARC-CR. De esta forma se permite continuar con el ritmo normal de trabajo, causando solo pequeñas intervenciones del usuario interesado.

Así, se evita la inversión de tiempo y fondos, así como otorgar mayor independencia a la institución para este tipo de mediciones.

Capítulo 7. Conclusiones y recomendaciones

7.1. Conclusiones.

- Según los métodos investigados, la mejor forma de controlar la fresadora vertical CNC es mediante el modo MDI de su sistema de control.
- Se comprobó que la duración al realizar movimientos es mucho mayor en el eje rotativo A que en los demás ejes lineales X, Y o Z, variando los tiempos de medición según la implementación.
- La implementación de medición de circunferencias incrementa a un aproximado de 4 veces el tiempo de medición (22 horas), en lugar de las 5,5 horas si se implementara la medición por líneas verticales.
- El sistema esta limitado a mediciones de cilindros de máximo 120mm de altura, o 240mm con el uso del émbolo diseñado.
- El sistema de almacenamiento implementado generó los archivos en el formato acordado para almacenar los datos de las mediciones.
- El graficador de datos implementado permite al usuario obtener una representación de las mediciones de forma rápida y confiable.
- El método de medición implementado resuelve el inconveniente de la empresa, pero involucra mayor participación del usuario para completar la medición.
- El sistema puede ser implementado por menos de una decima parte del costo de instrumentos de medición con características similares.

7.2. Recomendaciones

- Adquirir el sensor Riftek RF-603 recomendado, y depurar el sistema.
- Agregar facilidades al sistema para que se adapte a cilindros de diferentes dimensiones.
- Desarrollar una interfaz de usuario más intuitiva e informativa.
- Optimizar los tiempos de medición del sistema.
- Mejorar el programa graficador permitiendo seleccionar rangos de altura a graficar, así como la selección del archivo de datos a graficar.
- Calcular y certificar la precisión, resolución y exactitud del sistema.

Capítulo 8. Bibliografía

- [1] A. Kuttan, «Introduction to Mechatronics,» de *Introduction to Mechatronics*, Oxford University Press, 2007, pp. 216-247.
- [2] «CENTROID CNC M-400 Controls for mills,» 2011. [En línea]. Available: http://www.centroidcnc.com/cnc_control_m400.htm. [Último acceso: Setiembre 2012].
- [3] USB-Implementers'-Forum, «Device Class Definition HID,» 27 June 2001. [En línea]. Available: http://www.usb.org/developers/devclass_docs/HID1_11.pdf. [Último acceso: 10 2012].
- [4] Riftek, «RF-603 Laser triangulation Sensors,» 2012.
- [5] I. C. P. Salazar, «Coordenadas cilíndricas y esféricas,» Instituto Tecnológico Superior de Calkíni, Campeche, México, 2006.
- [6] XIAMEN OCULAR, «GDM1602K Specifications of LCD Module,» SparkFun, 2001.
- [7] Dallas Semiconductor, «DS1307 Real Time Clock,» Octubre 2001. [En línea]. Available: <http://www.sparkfun.com/datasheets/Components/DS1307.pdf>. [Último acceso: Setiembre 2012].
- [8] Texas Instruments, «MAX232 Dual EIA-232 Drivers/Receivers,» Texas Instruments, Dallas, Texas, 2004.
- [9] R. Jimenez, «Control Numérico por Computadora - UN Virtual,» 28 Junio 2005. [En línea]. Available: http://www.virtual.unal.edu.co/cursos/ingenieria/mecatronica/docs_curso/Anexos/TUTORIALcnc/DOCUMENTOS/otros/U4-control-numerico-por-computadora.pdf. [Último acceso: Agosto 2012].
- [10] F. Chang Diaz, «The VASIMR Rocket,» *Scientific American*, pp. 90-97, 2000.
- [11] INA, «Centros Regionales: INA,» [En línea]. Available: http://www.ina.ac.cr/regionales/regional_occidental/10.CDT%20Metalmechanica.html. [Último acceso: 10 Julio 2012].
- [12] AARC, «Ad Astra Costa Rica,» 2009. [En línea]. Available: http://www.adastrarocket.com/aarc/Ad_Astra_Costa_Rica. [Último acceso: 9 Julio 2012].
- [13] USB.org, «HID Tools,» [En línea]. Available: <http://www.usb.org/developers/hidpage>. [Último acceso: 10 10 2012].
- [14] Arduino, «Arduino Leonardo,» 2012. [En línea]. Available: <http://arduino.cc/en/Main/ArduinoBoardLeonardo>. [Último acceso: Setiembre 2012].

Capítulo 9. Anexos

A.1. Código de las funciones relevantes

Se presenta el código de las funciones mas relevantes del sistema, como lo son los métodos de calibración, medición y simulación del sensor.

A.1.1. Acercar a cilindro

```
int AcercarACilindro(){  
  
    int radioext=0;//Cuenta los milímetros de movimiento del sensor hacia el  
    cilindro, para tomarlos en cuenta al hacer el calculo del radio medido.  
    while(digitalRead(SensorA1)==LOW){ /*Si no se encuentra en el rango, hace  
    el decremento en X, para acercar al sensor a la superficie del cilindro. El  
    sensor siempre debe apuntar a la Izq. y ser inicializado correctamente. */  
        radioext+=1;  
        Keyboard.begin();  
        delay(500);  
        Keyboard.press(KEY_F3);  
        Keyboard.release(KEY_F3);  
        delay(250);  
        Keyboard.print("G00 X[#5041-1]");/*Comando de movimiento, moviendo X a la  
    posicion actual menos 1mm(Movimiento a la Izq)*/  
        delay(250);  
        Keyboard.press(KEY_RETURN);  
        Keyboard.release(KEY_RETURN);  
        delay(250);  
        Keyboard.press(KEY_LEFT_ALT);  
        Keyboard.print("S");  
        Keyboard.releaseAll();  
    }  
    return radioext;  
}
```

A.1.2. Angulo de inicio

```
void AnguloInicio(){  
  
    while(digitalRead(SensorA1)==HIGH){ /*Si no se encuentra en el rango, hace  
    el aumento en A, para girar el sensor a la ranura de inicio. El sensor  
    siempre debe apuntar a la Izq. i ser inicializado correctamente.  
        //Va a girar la base del cilindro hasta que esta este siempre en la misma  
    posicion.*/  
        Keyboard.begin();  
        delay(500);  
        Keyboard.press(KEY_F3);  
        Keyboard.release(KEY_F3);  
        delay(250);  
        Keyboard.print("G00 A[#5044+1]");/*Comando de movimiento, moviendo A a la  
    posicion actual mas 1º.*/  
        delay(250);  
        Keyboard.press(KEY_RETURN);  
        Keyboard.release(KEY_RETURN);  
        delay(250);  
        Keyboard.press(KEY_LEFT_ALT);  
        Keyboard.print("S");  
        Keyboard.releaseAll();  
        delay(1500);  
    }  
}
```

A.1.3. Medición

```
float MedicionSensor(){  
  
    int medsens=0;  
    float medicion=0.0;  
    byte comando0=0x01;  
    byte comando1=0x86;  
    Serial1.write(comando0);// manda los comandos de "Inquiring for result"  
    Serial1.write(comando1);  
    while(Serial1.available(<4){} //Lectura de los 4 bytes enviados por el  
    sensor  
    med[0]=Serial1.read();  
    med[1]=Serial1.read();  
    med[2]=Serial1.read();  
    med[3]=Serial1.read();  
    /*Se eliminan los primeros 4 bits de cada byte, y se concatenan de LSB a  
    MSB.*/  
    medsens=((med[3] & 0x0F)<<12)|((med[2] & 0x0F)<<8)|((med[1] &  
    0x0F)<<4)|((med[0] & 0x0F));  
    medicion=(medsens*5.0)/16384 + radioextra + AnchoSensor;  
    return medicion;  
}
```

A.1.4. Simulación de sensor

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  if(Serial.available(>1){
    command0=Serial.read();
    command1=Serial.read();
    if(command0==0x01 & command1==0x86){
      distancia=random(7864,8520); /*Calculo random de la distancia medida
desde 2,4mm hasta 2,6 mm.*/
      //Dividir la variable de distancia de 2 bytes en 4 bits (nibbles)
      dist1=(distancia & 0x000F); //4 bits menos significativos
      dist2=(distancia & 0x00F0)>>4;
      dist3=(distancia & 0x0F00)>>8;
      dist4=(distancia & 0xF000)>>12;
      /*Se les agrega 0xB, aunque se omite el aumentar el contador, ya que
por ser datos tomados por request y no en stream, se sabe que todos
pertenecen a la misma medición*/
      data1=0xB0|dist1;
      data2=0xB0|dist2;
      data3=0xB0|dist3;
      data4=0xB0|dist4;
      Serial.write(data1);
      Serial.write(data2);
      Serial.write(data3);
      Serial.write(data4);
    }
  }
  delay(25);
}
```

A.2. Códigos de programación y solicitudes para el sensor

Tabla 3. Códigos de solicitudes para el sensor Riftek RF-603.

Request code		Message (size in bytes)	Answer (size in bytes)
01h	Device identification	—	-device type -firmware release -serial number -base distance -range
02h	Reading of parameter	- code of parameter	- value of parameter
03h	Writing of parameter	- code of parameter - value of parameter	—
04h	Storing current parameters to FLASH-memory	- constant AAh	- constant AAh
04h	Recovery of parameter default values in	- constant 69h	- constant 69h
05h	Latching of current result	—	—
06h	Inquiring of result	—	- result
07h	Inquiring of a stream of results	—	- stream of results
08h	Stop data streaming	—	—

A.3. Comandos de teclado de la CNC

Tabla 4. Comandos de teclado para la fresadora CNC

Hot Key	Action
ALT A	Spindle auto/manual*
ALT B	Screen blanker on
ALT C	Flood coolant on/off*
ALT D	Switch between current position and machine position
CTRL D	Switch DRO between position and distance to go
ALT E	Mist coolant on/off*
ALT F	Displays available system memory
ALT H	Feed hold on/off*
ALT I	PLC diagnostics
ALT J	Enables keyboard jogging*
ALT K	Displays current ATC tool bin location
ALT M	MDI
ALT O	Tool check*
ALT P	Live PID display
CTRL P	Clear max and min error display
ALT Q	Spindle on/off counter-clockwise*
ALT R	Spindle on/off clockwise*
ALT S	Cycle start
ALT T	Displays current motor temperature estimates
ALT V	Displays current software version #
ALT W	MPG on/off*
ALT + / ALT -	Selects next/previous WCS, cycles through WCS 1-18**
ALT Tab	Cycle through currently running applications

A.4. Comandos de teclado de Arduino Leonardo

Tabla 5. Comandos de teclado para Arduino Leonardo

Key	Hexadecimal value	Decimal value
KEY_LEFT_CTRL	0x80	128
KEY_LEFT_SHIFT	0x81	129
KEY_LEFT_ALT	0x82	130
KEY_LEFT_GUI	0x83	131
KEY_RIGHT_CTRL	0x84	132
KEY_RIGHT_SHIFT	0x85	133
KEY_RIGHT_ALT	0x86	134
KEY_RIGHT_GUI	0x87	135
KEY_UP_ARROW	0xDA	218
KEY_DOWN_ARROW	0xD9	217
KEY_LEFT_ARROW	0xD8	216
KEY_RIGHT_ARROW	0xD7	215
KEY_BACKSPACE	0xB2	178
KEY_TAB	0xB3	179
KEY_RETURN	0xB0	176
KEY_ESC	0xB1	177
KEY_INSERT	0xD1	209
KEY_DELETE	0xD4	212
KEY_PAGE_UP	0xD3	211
KEY_PAGE_DOWN	0xD6	214
KEY_HOME	0xD2	210
KEY_END	0xD5	213
KEY_CAPS_LOCK	0xC1	193
KEY_F1	0xC2	194
KEY_F2	0xC3	195
KEY_F3	0xC4	196

KEY_F4	0xC5	197
KEY_F5	0xC6	198
KEY_F6	0xC7	199
KEY_F7	0xC8	200
KEY_F8	0xC9	201
KEY_F9	0xCA	202
KEY_F10	0xCB	203
KEY_F11	0xCC	204
KEY_F12	0xCD	205