



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

TEMA:

**Diseño de un dispositivo de control de microaspiración para biopsias
cerebrales estereotáxicas mediante el uso de microcontroladores**

AUTOR:

Peña Gaibor, Miguel Andres

Trabajo de Titulación previo a la obtención del título de
INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO

TUTOR:

M. Sc. Córdova Rivadeneira, Luis Silvio

Guayaquil, Ecuador

28 de febrero del 2020



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr.
Peña Gaibor, Miguel Andres como requerimiento para la obtención del
título de **INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO**.

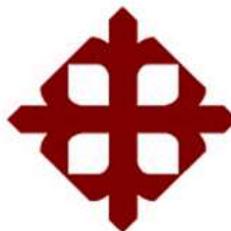
TUTOR

M. Sc. Córdova Rivadeneira, Luis Silvio

DIRECTOR DE CARRERA

M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, 28 de febrero del 2020



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Peña Gaibor, Miguel Andres**

DECLARÓ QUE:

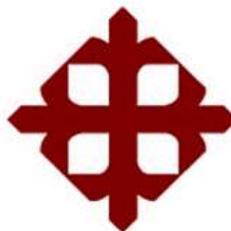
El trabajo de titulación: **“Diseño de un dispositivo de control de microaspiración para biopsias cerebrales estereotáxicas mediante el uso de microcontroladores”**, previo a la obtención del Título de **Ingeniero Electrónico en Control y Automatismo**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, 28 de febrero del 2020

EL AUTOR

PEÑA GAIBOR, MIGUEL ANDRES



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

AUTORIZACIÓN

Yo, **Peña Gaibor, Miguel Andres**

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **“Diseño de un dispositivo de control de microaspiración para biopsias cerebrales estereotáxicas mediante el uso de microcontroladores”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 28 de febrero del 2020

EL AUTOR

PEÑA GAIBOR, MIGUEL ANDRES

REPORTE DE URKUND

The screenshot shows the URKUND interface. On the left, document details are listed: 'Documento: TT Andres Peña.docx (D63919950)', 'Presentado: 2020-02-14 13:53 (-05:00)', 'Presentado por: fernandopm23@hotmail.com', 'Recibido: edwin.palacios.ucsg@analysis.orkund.com', and 'Mensaje: Revisión TT Andres Peña. 1% de estas 22 páginas, se componen de texto presente en 3 fuentes.' On the right, there is a 'Lista de fuentes' section with a table of sources and a 'Fuentes alternativas' section with a link to a wiki page. The bottom of the interface shows navigation icons and a status bar with '1 Advertencias'.

Categoría	Enlace/nombre de archivo
	naranjo_gregory_final.docx
	https://docplayer.es/33098861-Instituto-politecn...
	https://profesivandor.com/files.wordpress.com/20...
	https://www.allaboutcircuits.com/news/resource...
	https://www.wiki.cch.unam.mx/Microcontrolado...

Fuentes alternativas

- <https://wiki2.org/es/Microcontrolador>

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA

ELECTRÓNICA EN CONTROL Y AUTOMATISMO

TEMA: Diseño de un dispositivo de control de micro aspiración para biopsias cerebrales estereotáxicas mediante el uso de microcontroladores.

AUTOR: Peña Galbor, Miguel Andres

Trabajo de Titulación previo a la obtención del título de INGENIERO ELECTRONICO EN CONTROL Y AUTOMATISMO

TUTOR: M. Sc. Córdova Rivadeneira, Luis Silvio

Guayaquil, Ecuador

1 de marzo del 2020

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE

DEDICATORIA

Este trabajo y todo el proceso es para mi Madre, Beatriz Lucía de quien siempre admiré su tenacidad para enfrentar las adversidades de la vida, para mi padre, Miguel Ángel quien me ha enseñado valiosas lecciones de vida y de quien admiro la creatividad, Jordan David mi hermano quien ha sido mi motor y he tratado de ser el mejor ejemplo para el cada día de esta corta vida. Lilena mi inspiración, mi felicidad, mi todo.

EL AUTOR

PEÑA GAIBOR, MIGUEL ANDRES

AGRADECIMIENTO

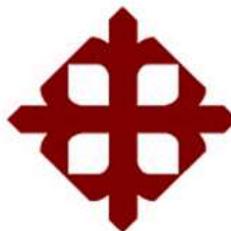
Agradezco a cada una de las personas que formaron parte de este pequeño proceso de la vida, a mi tutor el Ing. Luis Córdova por su disposición y enseñanza en materias cruciales de nuestra carrera, así mismo al Ing. Fernando Palacios por su arduo servicio y guía en las adversidades que se han presentado a lo largo de esta carrera.

Agradezco al Dr. Fidel Villamar por haber confiado en mí y abrir las puertas desde el principio permitiéndome colaborar y aprendiendo mucho de él.

Agradezco al Dr. Boris Zurita por su invaluable apoyo al permitirme formar parte de su equipo de trabajo y creer en mí, además de apostar por las ideas y proyectos presentado para mejor la calidad del trabajo.

EL AUTOR

PEÑA GAIBOR, MIGUEL ANDRES



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

TRIBUNAL DE SUSTENTACIÓN

f. _____

M. Sc. ROMERO PAZ, MANUEL DE JESUS
DECANO

f. _____

M. Sc. HERAS SÁNCHEZ, MIGUEL ARMANDO
DIRECTOR DE CARRERA

f. _____

M. Sc. PALACIOS MELÉNDEZ, EDWIN FERNANDO
OPONENTE

Índice General

Capítulo 1: Descripción General del Trabajo de Titulación	2
1.1. Introducción.....	2
1.2. Antecedentes.....	3
1.3. Definición del Problema.....	4
1.4. Justificación del Problema.....	5
1.5. Objetivos del Problema de Investigación.....	5
1.5.1. Objetivo General.....	5
1.5.2. Objetivos Específicos.....	6
1.6. Hipótesis.....	6
1.7. Metodología de Investigación.....	6
Capítulo 2: Fundamentación Teórica	7
2.1. Conceptualización general de microcontroladores.	7
2.2. Microcontroladores PIC.....	9
2.3. Arquitectura de los microcontroladores PIC	10
2.4. Arquitectura básica de una CPU.....	11
2.4.1. Arquitectura basada en acumuladores.....	12
2.4.2. Arquitectura basada en registros.....	12
2.4.3. Arquitectura basada en pila (Stack).....	13
2.4.4. Arquitectura de pipeline.....	13
2.5. Conjunto de instrucciones RISC contra CISC.....	14
2.6. Memoria.....	15
2.6.1. RAM:.....	16
2.6.2. EEPROM direccionable por byte:.....	17
2.6.3. Flash EEPROM:.....	17
2.7. Recursos especiales	17
2.7.1. Temporizadores.....	18
2.7.2. Watchdog.....	18

2.7.3.	Estado de bajo consumo.....	19
2.7.4.	Convertor A/D.....	19
2.8.	Sistemas de comunicaciones en microcontroladores.	19
2.8.1.	Comunicaciones en serie	20
2.9.	Lenguaje de programación.....	22
2.9.1.	C y ensamblador.....	22
2.9.2.	MicroPython.....	24
2.9.3.	Blockly	25
2.9.4.	Arduino IDE basado en C ++	26
2.10.	Introducción a los motores paso a paso.	27
2.11.	Ventajas y desventajas de los motores paso a paso.....	28
2.12.	Tipos de motor paso a paso.....	29
2.12.1.	Reluctancia variable.....	29
2.12.2.	Imán permanente.....	30
2.12.3.	Híbrido	31
2.13.	Modelo del motor paso a paso.	31
2.14.	Controlador de motor paso a paso.	35
	Capítulo 3: Diseño, Implementación y resultados.....	37
3.1.	Descripción general de sistema estereotáxicos.	37
3.2.	Diseño de la base del sistema de microaspiración estereotáxica.	42
3.3.	Diseño del riel.....	43
3.4.	Materiales utilizados en la implementación del sistema de microaspiración.....	44
3.5.	Diseño electrónico que opera al sistema de microaspiración estereotáxico controlado.	45
3.6.	Programación del sistema de microaspersión en CCS C Compiler.	50
3.7.	Implementación y validación del equipo diseñado.	55
	Conclusiones.....	59
	Recomendaciones.....	60
	Bibliografía	61

Índice de Figuras

Capítulo 2

Figura 2. 1: Diagrama de bloques de un microcontrolador.	7
Figura 2. 2: Diagrama de bloques del IC de un microcontrolador PIC.	9
Figura 2. 3: Representación del IC de un microcontrolador PIC 16F887.	9
Figura 2. 4: Diagrama de bloques de la arquitectura Von Neumann.	10
Figura 2. 5: Representación del IC de un microcontrolador PIC 16F887.	11
Figura 2. 6: Representación de la arquitectura de CPU basada en acumuladores.	12
Figura 2. 7: Representación de la arquitectura de CPU basada en registros.	13
Figura 2. 8: Representación de la arquitectura de CPU basada en pila (Stack).	13
Figura 2. 9: Representación del IC de un microcontrolador PIC 16F887.	14
Figura 2. 10: Representación del IC de un microcontrolador PIC 16F887. ...	16
Figura 2. 11: Circuito esquemático para el reinicio externo de borrado maestro.	18
Figura 2. 12: Ventana principal de plataforma de programación CCS C Compiler para lenguaje de alto nivel C.	23
Figura 2. 13: Ventana principal de plataforma de programación MPLAB para lenguaje de máquina.	24
Figura 2. 14: Ventana principal de plataforma de programación Python.	25
Figura 2. 15: Ventana principal de plataforma de programación Blockly.	26
Figura 2. 16: Ventana principal de plataforma de programación Arduino IDE basado en C++.	27
Figura 2. 17: Representación del motor paso a paso de reluctancia variable.	30
Figura 2. 18: Representación de un motor paso a paso de imán permanente.	30

Figura 2. 19: Representación de un motor paso a paso híbrido.	31
Figura 2. 20: Representación de un motor paso a paso bipolar híbrido genérico. A la izquierda se muestra el estator con los ocho polos, mientras que a la derecha se representa el rotor dentado.	33

Capítulo 3

Figura 3. 1: Ejemplo de una imagen de un tumor cerebral	37
Figura 3. 2: Imagen del sistema de obtención estereotáxica controlada.	38
Figura 3. 3: Imagen de una tomografía estereotáxica.....	38
Figura 3. 4: Imagen de selección de blanco quirúrgico.....	39
Figura 3. 5: Imagen de las coordenadas del blanco quirúrgico.	39
Figura 3. 6: Imagen triplanar.....	40
Figura 3. 7: Imagen de reconstrucción volumétrica y trayectoria	40
Figura 3. 8: Cánula de biopsia cerebral.....	40
Figura 3. 9: Imagen de un procedimiento quirúrgico estereotáxico.	41
Figura 3. 10: Diseño 3D en AutoCAD en diferentes posiciones de la base y riel del sistema de microaspiración estereotáxica controlada.	43
Figura 3. 11: Bujes o rodamientos para que se deslice la barra acerada de 12 mm.	44
Figura 3. 12: Barra acerada de 12 mm para el desplazamiento del riel.....	44
Figura 3. 13: Motor de paso NEMA 17 bipolar.	45
Figura 3. 14: Tornillo sin fin para ejecución del movimiento del riel.	45
Figura 3. 15: Tuerca para el tornillo sin fin-.....	45
Figura 3. 16: Diagrama esquemático en Proteus del diseño electrónico para sistema microaspiración estereotáxica controlado.	46
Figura 3. 17: Mensaje inicial del sistema propuesto.....	47
Figura 3. 18: Menú principal del sistema propuesto.....	48
Figura 3. 19: Inicio de biopsia para obtención de muestra cerebral.	48
Figura 3. 20: Mensaje de recorrido del émbolo en la base y riel del sistema implementado.....	49

Figura 3. 21: Mensaje de finalización de biopsia mediante microaspiración estereotáxica controlada.	49
Figura 3. 22: Mensaje de expulsión del tejido o muestra cerebral.	49
Figura 3. 23: Diseño de la PCB del sistema de microaspersión estereotáxica controlada.	50
Figura 3. 24: Diagrama ASM del sistema de microaspersión estereotáxica.	51
Figura 3. 25: Declaración de librería del sistema propuesto en CCS C Compiler.	52
Figura 3. 26: Declaración de funciones del sistema propuesto en CCS C Compiler.	52
Figura 3. 27: Función que ejecuta el avance del embolo.	52
Figura 3. 28: Función de presentación inicial.	53
Figura 3. 29: Función para tomar muestra del cerebro.	53
Figura 3. 30: Función para tomar muestra del hueso.	54
Figura 3. 31: Función para tomar muestra del hígado.	54
Figura 3. 32: Función para tomar muestra del pulmón.	55
Figura 3. 33: Función para avance y retroceso del embolo.	56
Figura 3. 34: Función para la operación del teclado matricial.	56
Figura 3. 35: Implementación del equipo que controla al sistema de microaspersión estereotáxica.	57
Figura 3. 36: Implementación final del equipo de microaspersión estereotáxica.	57
Figura 3. 37: Validación del equipo implementado.	58

Resumen

Las biopsias cerebrales estereotáxicas es una técnica quirúrgica que permite la extracción de una muestra diminuta de tejido anómalo para analizarlo y poder tener un diagnóstico certero, representan uno de los procedimientos como medio de diagnóstico ante un tumor, una de las complicaciones dentro del procedimiento es el riesgo de sangrado y/o muerte. Una complicación principal asociada con una biopsia cerebral estereotáxica es un hematoma intratumoral, que también es la causa más común de morbilidad relacionada con parálisis permanente y mortalidad en una biopsia cerebral estereotáxica. El riesgo de hemorragia perioperatoria es generalmente de entre 1% y 10%, pero esto podría ser una subestimación, ya que no es común para muchos neurocirujanos para realizar tomografías computarizadas muestras de tumores cerebrales para el diagnóstico después una biopsia cerebral estereotáxica sin complicaciones. El motivo de esta investigación es porque no existe un mecanismo de aspiración controlado o automático, ya que se lo realiza tradicionalmente a “pulso” del neurocirujano. Para esto se ha propuesto desarrollar e implementar un sistema de microaspiración electrónico controlado usando tecnología como la impresión 3D y el diseño de un circuito electrónico.

Palabras claves: ESTEREOTAXIA, TUMOR, BIOPSIA, MICROASPIRACIÓN, MICROCONTROLADORES, AUTOMÁTICO

Capítulo 1: Descripción General del Trabajo de Titulación

1.1. Introducción.

El presente trabajo de diseño e implementación de un sistema de micro aspiración controlada para biopsias cerebrales aborda un vacío en la técnica del procedimiento neuroquirúrgico ya que de manera tradicional se lo realiza a pulso del neurocirujano, sin aplicar una fuerza uniforme en el momento de la extracción. La biopsia estereotáxica cerebral es el procedimiento de primera elección ya que el método de selección del blanco quirúrgico es muy preciso, pero en el momento de la toma de la muestra el procedimiento aumenta su entropía ya que no existe un mecanismo de control mecánico ni mucho menos electrónico.

Las estrategias terapéuticas para lesiones de masa intracraneales son más eficaces cuando se basan en diagnósticos histopatológicos, en lugar de en los hallazgos clínicos y radiográficos. Cuando se sospecha fuertemente que las lesiones son tumores malignos, la resección quirúrgica directa debe ser considerada, porque la escisión tumoral masiva tiene un impacto positivo y significativo sobre el pronóstico y facilita el análisis histopatológico. (Mizobuchi et al., 2019)

A veces, la resección quirúrgica directa es difícil debido a lesiones múltiples, lesiones cerebrales profundas, la edad o condición general del paciente. Sin embargo, la obtención de tejido para el análisis es crítico para guiar la terapia y la prevención de intervenciones innecesarias. Ahora, biopsia estereotáctica guiada se utiliza comúnmente para obtener muestras para el diagnóstico histopatológico de tumores localizados en zonas profundas o elocuentes dentro del cerebro. (Mizobuchi et al., 2019)

La Biopsia cerebral estereotáctica ha reducido la morbilidad asociada a la cirugía craneal abierto y técnicas no estereotacticas y ha reducido al mínimo la interrupción de las estructuras neurales que funcionan normalmente (1, 5). De hecho, la tomografía computarizada (CT) / imágenes de resonancia magnética (MRI) – biopsias cerebrales asistidas, ahora se utiliza

rutinariamente para el diagnóstico y se considera un procedimiento preciso y relativamente seguro. Tales procedimientos a base de neuroimagen son menos invasivos, especialmente para las lesiones quirúrgicamente inaccesibles y para los pacientes que no pueden someterse craneotomía debido a su condición general. (Mizobuchi et al., 2019)

1.2. Antecedentes.

Definiendo la palabra estereotáxica en el año de 1908, Sir Victor Horsley y Robert Clarke crearon su propio sistema para tomar muestras en el cerebro de los monos, esto fue de uso experimental. Ellos empezaron a utilizar los planos cartesianos de coordenadas haciendo posible la localización “exacta” dentro del sistema estereotáxico. Dentro del procedimiento estereotáxico existen complicaciones que están ligadas entre sí, tales como la hemorragia intracerebral, el error en la selección del blanco quirúrgico y la obtención de una muestra insuficiente.

Con los recientes avances en la tomografía computarizada de alta resolución (TC) y Resonancia magnética en las últimas dos décadas, el uso de biopsias estereotáxica se convierte en rutina en el diagnóstico y tratamiento de muchas lesiones cerebrales. El tratamiento de las lesiones intracraneales debe basarse en un diagnóstico preciso. A pesar de su sofisticación, las nuevas modalidades de imagen siguen sin poder establecer el diagnóstico histológico.

La biopsia cerebral estereotáxica oferta un método muy preciso de acercarse a las lesiones cerebrales situados profundamente, lesiones localizadas las zonas cercanas elocuentes del cerebro o lesiones múltiples. La biopsia cerebral estereotáctica es hoy en día un procedimiento que se realizan con mucha frecuencia. Desde su introducción, un gran número de series se han publicado documentando su relativa seguridad.

La tasa de mortalidad registrada es inferior al 1% y la tasa de morbilidad es inferior al 5%. Es posible que los neurocirujanos y otros médicos, así como a los pacientes, a subestimar las posibles complicaciones de la biopsia

estereotáxica porque el procedimiento se realiza generalmente en anestesia local utilizando sólo una pequeña abertura en el cráneo. Sin embargo, el procedimiento es “ciego” y uno debe anticipar una significativa incidencia de la morbilidad.

1.3. Definición del Problema.

La biopsia tumoral estereotáctica es la más básica de todas las intervenciones estereotácticas. Representa un punto de partida ideal para sistemas de estereotaxia recientemente desarrollados. Una gran diversidad de los tumores cerebrales, los más comunes son los gliomas (50.3%), los meningiomas (20.9%), Adenomas hipofisarios (15%) y tumores de la vaina nerviosa (8%). Aproximadamente 1.5% de todas las enfermedades cancerosas son tumores cerebrales. Para establecer un diagnóstico, el primer método de elección son las herramientas de imagen, por ejemplo, tomografías computadas y resonancia nuclear magnética.

A veces, sin embargo, existe la necesidad de una caracterización histopatológica de una muestra de tejido ya que algunos tumores no pueden caracterizarse con la certeza necesaria. Hay evidencia de que más del 10% de los diagnósticos basados en muestras histológicas fueron insospechados basado en los estudios de imágenes convencionales. Además, la muestra del tejido de biopsia se puede utilizar para complementar información, por ejemplo, ensayos genéticos y moleculares.

Desde la década de 1960, las biopsias tumorales son muy comunes. Después de la invención de la Tomografía Computarizada, las biopsias a menudo se basaron en datos de imágenes 3D individuales. Este desarrollo condujo a un mayor aumento en las aplicaciones clínicas de las intervenciones estereotácticas. En la década de 1980, el uso de la biopsia ha aumentado aún más y se ha convertido en un procedimiento estándar de neurocirugía. Para determinar qué tipo y el grado de un tumor, puede ser necesario conseguir una o más muestras del tejido. En tumores metastásicos, una sola muestra es suficiente para establecer un diagnóstico, en las

neoplasias gliales es mejor proporcionar múltiples muestras de diferentes regiones del tumor al patólogo.

Dependiendo del tipo y la etapa del tumor, el volumen del tumor y con él el objetivo estereotáctico pueden variar enormemente. Especialmente en tumores pequeños, la selección del blanco debe ser altamente precisa. El posicionamiento defectuoso de la herramienta de biopsia puede conducir a diagnósticos falsos. El dispositivo estereotáctico se ajusta a los parámetros determinados por el software. La hemorragia intracerebral puede ocurrir por la selección del blanco errónea en el momento del planeamiento prequirúrgico en un área muy vascularizada, además de esto, al no tener un mecanismo de control que aplique una fuerza uniforme en el momento de la micro aspiración y como consecuencia de esto además no se obtiene una muestra satisfactoria.

Por los motivos anteriormente expuestos realicé el diseño mediante un software de diseño asistido por computadora (CAD) el cual permite realizar un movimiento uniforme del embolo de la jeringuilla de insulina que genera un vacío para tomar la muestra del blanco quirúrgico seleccionado por el médico, así mismo se programó el microcontrolador PIC 16F887 en lenguaje C, se diseñó un circuito electrónico y se ensambló los elementos que componen el proyecto en la placa.

1.4. Justificación del Problema.

Este trabajo busca la reducción de los riesgos de la biopsia estereotáctica en el momento de la aspiración de la muestra y las complicaciones que esta conlleva como consecuencia. Esto puede ser útil para en principio para los neurocirujanos y en un futuro usarlo para los demás tipos de tejidos del cuerpo humano.

1.5. Objetivos del Problema de Investigación.

1.5.1. Objetivo General.

Diseño de un dispositivo de control de microaspiración para biopsias cerebrales estereotácticas mediante el uso de microcontrolador PIC 16F873A.

1.5.2. Objetivos Específicos.

- Describir las bases fundamentales de microcontroladores, software de simulación.
- Descripción general de sistemas estereotáxicos de biopsias cerebrales.
- Diseñar la estructura del sistema de microaspiración para biopsias cerebrales en 3D mediante uso de software CAD (diseño asistido por computadora)
- Implementar el sistema de microaspiración.

1.6. Hipótesis.

Con el desarrollo del sistema de microaspiración automatizado se reducirá el riesgo de sangrado, así como la toma de una muestra insuficiente en el momento de la biopsia.

1.7. Metodología de Investigación.

Para este proyecto se utiliza el método científico y se usa el método observacional.

Capítulo 2: Fundamentación Teórica

2.1. Conceptualización general de microcontroladores.

Un microcontrolador es un circuito integrado que incluye en su interior tres unidades funcionales de un computador: la unidad central de procesamiento, la memoria y periféricos de entrada y salida. La memoria del programa en forma de flash ROM a menudo se incluyen en el chip, así también una minúscula cantidad de RAM. Estos son diseñados para aplicaciones embebidas, en contraste con los microprocesadores utilizados en computadoras personales u otras aplicaciones de propósito general.

En la figura 2.1 se proporciona un diagrama de bloques básico de un microcontrolador genérico. En las siguientes secciones, se describen brevemente los sistemas que se encuentran comúnmente a bordo de un microcontrolador típico.

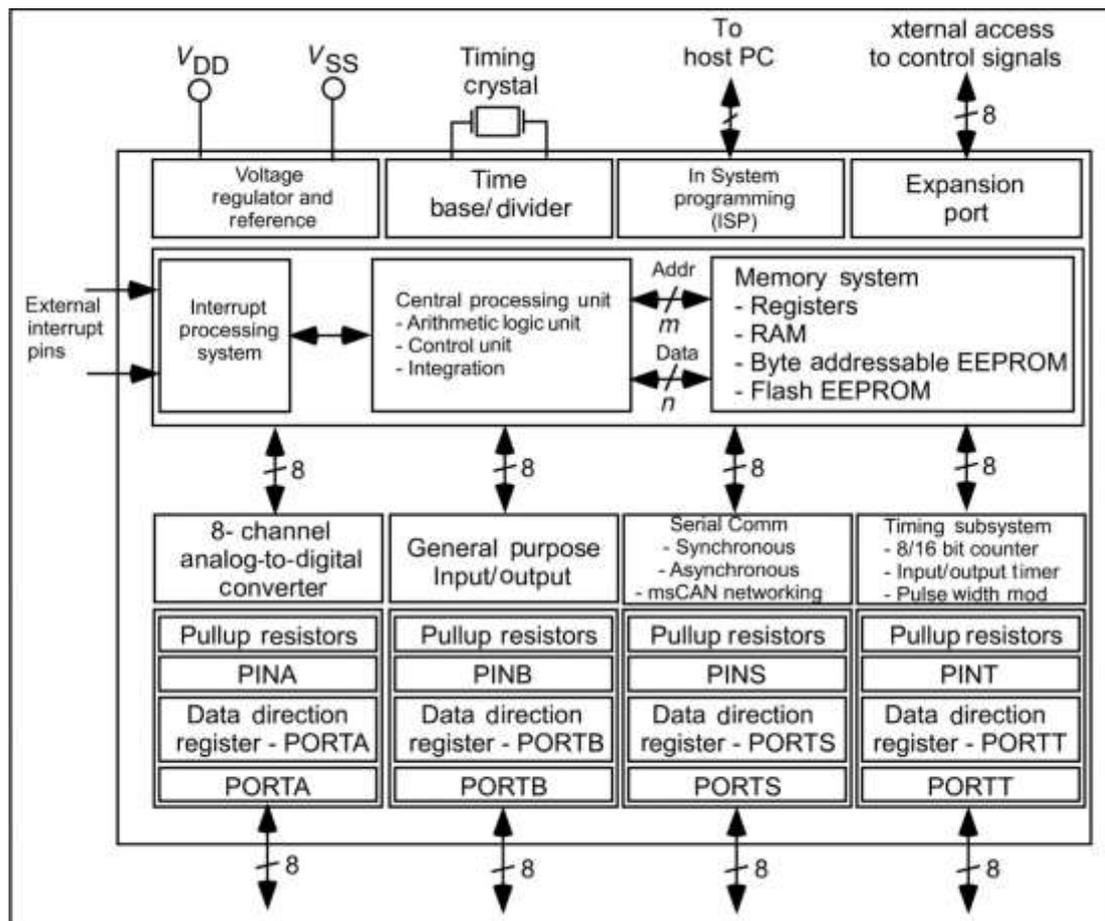


Figura 2. 1: Diagrama de bloques de un microcontrolador.

Fuente: (Verle, 2012a)

La mayoría de los microcontroladores están equipados con los subsistemas que se muestran en la figura 2.1. Los puertos se utilizan para proporcionar acceso al microcontrolador al mundo exterior. Por lo general, los puertos son bidireccionales y también tienen funciones alternativas, como conversión de analógico/digital, comunicaciones en serie y un sistema de temporización flexible.

Los microcontroladores también están equipados con un complemento de diferentes componentes de memoria. La operación normal del microcontrolador puede ser interrumpida por un evento externo usando los pines de interrupción externos. Esto permite que el microcontrolador responda a eventos de alta prioridad. El microcontrolador se programa mediante las funciones de Programación en el sistema (ISP) utilizando una computadora personal host. La base de tiempo para el microcontrolador es proporcionada por un oscilador o resonador de cristal externo

Los microcontroladores son muy utilizados en los sistemas controlados automáticamente y dispositivos, tales como los dispositivos médicos implantables, los sistemas de control del motor del automóvil, máquinas de oficina, herramientas eléctricas, electrodomésticos, etc. Al miniaturizar el tamaño y en consecuencia el costo en comparación con un diseño que emplea un microprocesador independiente, memoria y dispositivos de entrada/salida. Muchos microcontroladores suelen emplear bits de palabras de cuatro y funcionan a una velocidad de la frecuencia de un reloj incorporado tan bajas como 4 [kHz], de muy bajo consumo de energía.

Por lo general, tendrá la capacidad de conservar la funcionalidad a la espera de un evento como pulsar un botón u otra interrupción. Otros microcontroladores suelen ser útiles en situaciones críticas de rendimiento, donde necesitan que funcione más como un procesador de señal digital (DSP), con velocidades de reloj más alta y el consumo de energía. El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo se exteriorizan las líneas que dominan los periféricos. (Plaza Herrero, 2014)

2.2. Microcontroladores PIC.

Un microcontrolador PIC es un circuito integrado programable, en el cual se pueden encontrar todas las unidades para realizar y tener control de una determinada tarea. La figura 2.2 muestra el diagrama de bloques de la arquitectura interna de un microcontrolador PIC. Este emplea una variedad de instrucciones, las cuales van desde 35 hasta 70 para sistemas de gama baja y alta respectivamente.

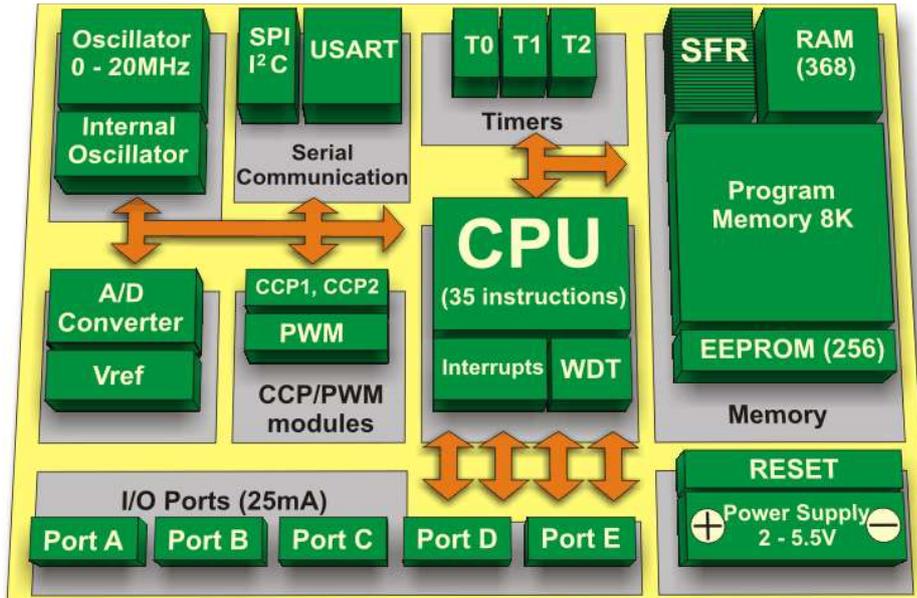


Figura 2. 2: Diagrama de bloques del IC de un microcontrolador PIC.
Fuente: (Verle, 2012a)

La empresa Microchip brinda un sistema de desarrollo freeware llamado MPLAB que incluye un software de simulación y ensamblador. Otras compañías desarrollan compiladores, tales como C y BASIC. La figura 2.3 muestra los pines de E/S correspondientes al microcontrolador PIC 16F887.

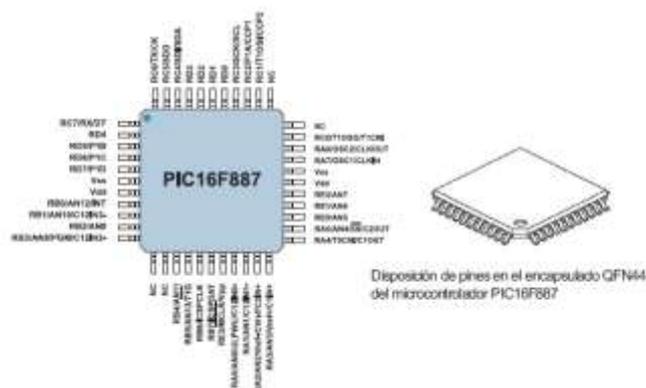


Figura 2. 3: Representación del IC de un microcontrolador PIC 16F887.
Fuente: (Verle, 2012a)

2.3. Arquitectura de los microcontroladores PIC

Actualmente existen dos tipos de arquitecturas que los fabricantes consideran al momento de diseñar los microcontroladores: Von Neumann y Harvard. Las dos arquitecturas funcionan para procesadores de tipo CISC o RISC. A continuación, se describen brevemente las arquitecturas ya mencionadas:

a) Arquitectura de Von Neuman

En la arquitectura clásica de Von Neumann, un procesador (CPU) está conectado a la memoria principal a través de un bus como se muestra en la figura 2.4. En los primeros sistemas informáticos, los tiempos para acceder a la memoria principal y para los cálculos computacionales estaban razonablemente bien equilibrados. Sin embargo, durante las últimas décadas, la velocidad de cómputo creció a un ritmo mucho más rápido en comparación con la velocidad de acceso a la memoria principal, lo que resultó en una brecha de rendimiento significativa. Esta discrepancia entre la velocidad de cálculo de la CPU y la velocidad de la memoria principal (DRAM) se conoce comúnmente como el cuello de botella de Von Neumann.

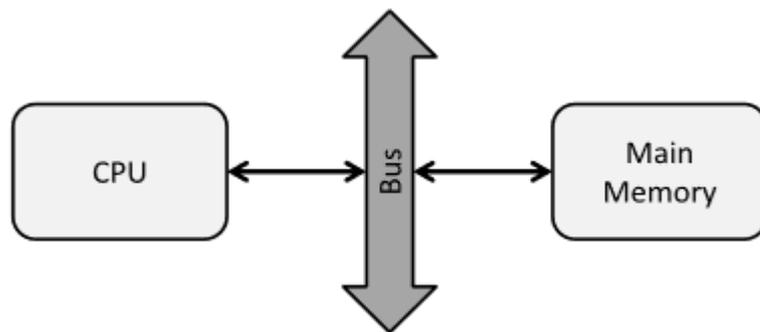


Figura 2. 4: Diagrama de bloques de la arquitectura Von Neumann.

Fuente: (Schmidt et al., 2018)

b) Arquitectura Harvard

Para Tan & Jiang, (2019) La arquitectura Harvard tiene dos espacios de memoria separados dedicados al código del programa y a los datos, respectivamente, dos buses de direcciones correspondientes y dos buses de datos para acceder a dos espacios de memoria, tal como se muestra en la figura 2.5. El procesador Harvard ofrece recuperación y ejecuciones en paralelo.

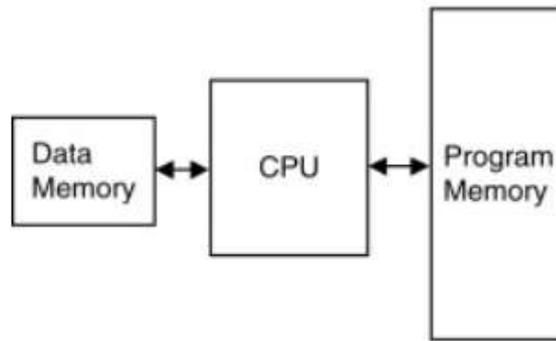


Figura 2. 5: Representación del IC de un microcontrolador PIC 16F887.
Fuente: (Schmidt et al., 2018)

2.4. Arquitectura básica de una CPU.

Un microcontrolador en su forma más fundamental es un sistema informático completo contenido en un solo circuito integrado. Uno de los principales desafíos en un diseño basado en microcontrolador es elegir el mejor controlador para un diseño específico. El objetivo es elegir el microcontrolador más económico que tenga los parámetros y características deseados para la aplicación en cuestión.

Los microcontroladores van desde pequeños procesadores de cuatro bits con características limitadas hasta procesadores de 32 bits de alta velocidad y funciones completas. Una vez restablecida, la CPU del microcontrolador buscará secuencialmente una instrucción de programa de la memoria, decodificará el contenido de la instrucción (lo que se supone que debe hacer) y ejecutará (ejecutará) la instrucción. La unidad central de procesamiento (CPU) es el principal centro de control para todo el microcontrolador.

Mientras responde a las diferentes instrucciones del programa, la CPU llamará a sus subsistemas residentes para que realicen sus tareas. La arquitectura básica de una CPU generalmente se puede colocar en una de las varias categorías generales (véanse las figuras 2.6 a 2.8). Cabe destacar que una arquitectura dada no es mejor que la otra.

A continuación, se indican cada una de las arquitecturas de CPU's con sus ventajas y desventajas inherentes.

2.4.1. Arquitectura basada en acumuladores.

La figura 2.6 muestra la arquitectura basada en acumuladores, las instrucciones comienzan y terminan en registros especialmente designados llamados acumuladores (A y B). Normalmente, se realiza una operación en la que un operando se encuentra en un acumulador y el otro se recupera de la memoria. El resultado se coloca en el acumulador.

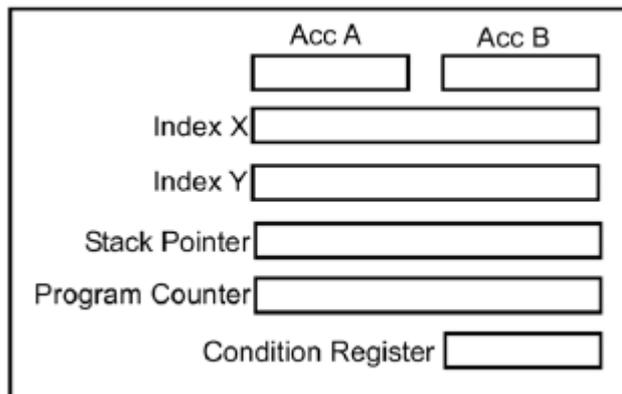


Figura 2. 6: Representación de la arquitectura de CPU basada en acumuladores.

Fuente: (Verle, 2012a)

Esta arquitectura tiende a ejecutarse más lentamente que las otras dos configuraciones, ya que los operandos deben recuperarse continuamente de la memoria. Por lo general, la memoria funciona a una velocidad más lenta que el procesador principal, por lo que el proceso debe ralentizarse para acomodar una búsqueda de operando desde la memoria. Una arquitectura basada en acumuladores tiene la capacidad de ejecutar instrucciones bastante complicadas. La arquitectura también puede modificarse de modo que un operando se encuentre en un registro y el otro se encuentre en la memoria.

2.4.2. Arquitectura basada en registros.

En una arquitectura basada en registros, ambos operandos se almacenan en registros que generalmente se colocan con la unidad central de procesamiento. El resultado de una operación dada también se almacena en un registro. Dado que la CPU y los registros funcionan a la misma velocidad, el procesador no tiene que reducir la velocidad para leer o escribir operandos. El contenido del registro se lee y se escribe en la memoria mediante una

operación en segundo plano. En la figura 2.7 se muestra el diagrama de bloques de la arquitectura CPU basada en registros.

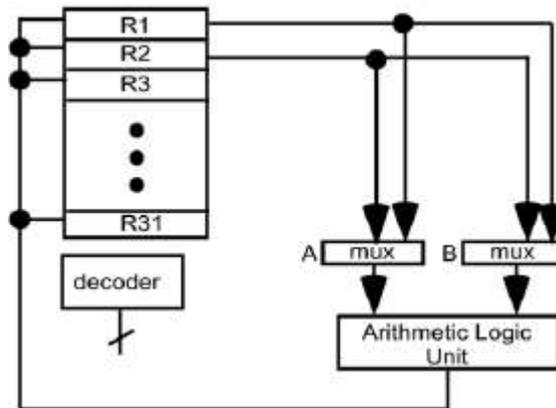


Figura 2. 7: Representación de la arquitectura de CPU basada en registros.
Fuente: (Verle, 2012a)

2.4.3. Arquitectura basada en pila (Stack).

En una arquitectura basada en pila, los operandos y la operación a realizar se almacenan en la pila. El resultado se vuelve a colocar en la pila. La pila puede basarse en registros dedicados o puede ser una parte especial de la memoria de acceso aleatorio. En la figura 2.8 se muestra el diagrama de bloques de la arquitectura CPU basada en pila.

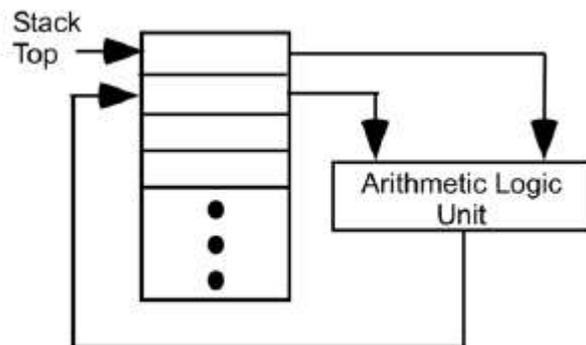


Figura 2. 8: Representación de la arquitectura de CPU basada en pila (Stack).
Fuente: (Verle, 2012a)

2.4.4. Arquitectura de pipeline.

Una arquitectura de microcontrolador basada en pipeline tiene la forma general ilustrada en la figura 2.9. La arquitectura consta de subsistemas de hardware separados llamados etapas para recuperar una instrucción de la memoria, decodificar la instrucción, recuperar operandos de la memoria o

registros, ejecutar la instrucción y luego volver a escribir los resultados en la memoria. Cada etapa procesa simultáneamente una instrucción diferente, de modo que el resultado general es que una instrucción completa la ejecución en cada ciclo de reloj.

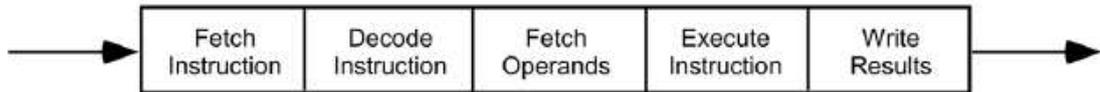


Figura 2. 9: Representación del IC de un microcontrolador PIC 16F887.

Fuente: (Schmidt et al., 2018)

Por ejemplo, en un pipeline de cinco etapas, cinco instrucciones se procesan simultáneamente a través de pipeline, cada una en una etapa diferente. Por lo general, las instrucciones en un sistema de procesamiento de tuberías son instrucciones simples que se implementan fácilmente en una sola etapa. Se construyen instrucciones más complejas a partir de estos pequeños bloques de construcción de instrucciones.

2.5. Conjunto de instrucciones RISC contra CISC.

Muy relacionada con la arquitectura de hardware del microcontrolador está la arquitectura del conjunto de instrucciones. Hay dos tipos básicos de arquitecturas de conjuntos de instrucciones: (a) Computadoras con un conjunto de instrucciones reducida (*Reduced Instruction Set Computer, RISC*) y computadoras con un conjunto de instrucciones complejas (*Complex Instruction Set Computer, CISC*).

Un procesador RISC, como su nombre lo indica, tiene un complemento de instrucciones simples de bloques de construcción. Las instrucciones más complejas se crean a partir de las instrucciones básicas del procesador RISC. Las arquitecturas de instrucción basadas en RISC se prestan a sistemas con arquitecturas de CPU menos complejas.

Una arquitectura basada en CISC tiene un complemento de funciones más completas, instrucciones más complejas que la arquitectura basada en RISC. Es difícil predecir si un programa dado se codificará de manera más eficiente con un conjunto de instrucciones basado en RISC o CISC. Depende

en gran medida de qué tan bien coincida el algoritmo específico con el conjunto de características de un procesador dado.

Como diseñador de sistemas, debe estar íntimamente familiarizado con la arquitectura de hardware y software de un microcontrolador dado, particularmente si codificará el sistema usando un lenguaje ensamblador. Sin embargo, si va a programar utilizando un lenguaje de alto nivel como C, no se requieren conocimientos de algunos de los detalles arquitectónicos de nivel inferior, siempre y cuando tenga un conocimiento profundo de los subsistemas de microcontroladores en el nivel de registro.

2.6. Memoria.

Como se mencionó anteriormente, el número de ubicaciones de memoria direccionables de manera exclusiva en un microcontrolador está determinado por el ancho del bus de direcciones. Este lapso de memoria direccionable generalmente contiene varios tipos diferentes de memoria, incluida la memoria estática de acceso aleatorio (*Static Random Access Memory, SRAM*), la memoria de solo lectura programable borrable eléctricamente (*Electrically Erasable Programmable Read Only Memory, EEPROM*) y la EEPROM Flash programable. Debe enfatizarse que estos componentes de memoria están contenidos dentro de los límites del chip del microcontrolador.

Para realizar un seguimiento de las ubicaciones de memoria en uso y el tipo de memoria presente dentro del sistema, se emplea una herramienta visual llamada mapa de memoria. El mapa de memoria proporciona el tamaño en bytes de cada componente de memoria y su dirección de inicio y parada dentro del sistema de memoria. Se proporciona un mapa de memoria de muestra en la figura 2.10. Tenga en cuenta que hay partes del mapa de memoria que no están en uso. Estos espacios abiertos se proporcionan para la expansión del sistema. Los siguientes componentes de memoria se muestran en la figura 2.10 y están comúnmente disponibles en la mayoría de los microcontroladores:

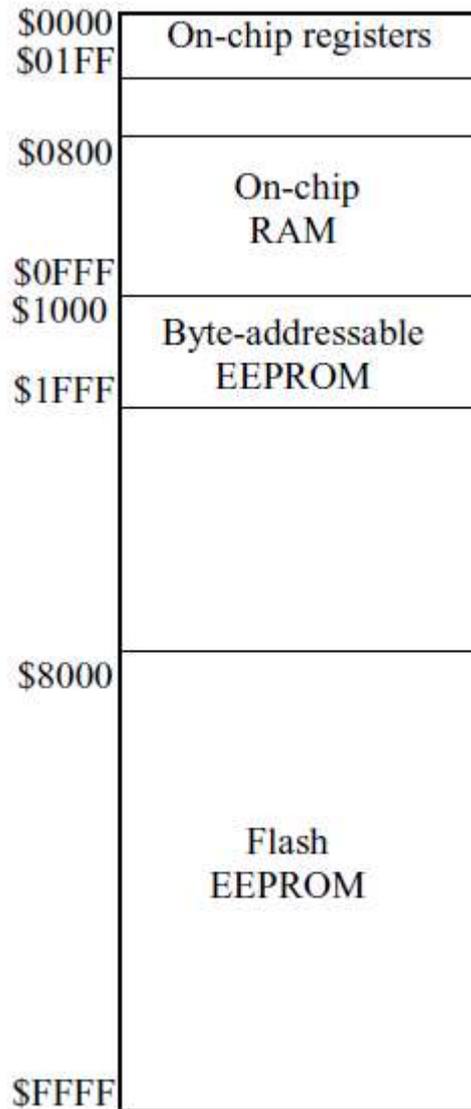


Figura 2. 10: Representación del IC de un microcontrolador PIC 16F887.
Fuente: (Schmidt et al., 2018)

2.6.1. RAM:

La memoria RAM es volátil, es decir, si el microcontrolador pierde energía, el contenido de la memoria RAM se pierde. Se puede escribir y leer durante la ejecución del programa. Normalmente se usa durante el desarrollo del sistema para almacenar un programa. Una vez que se completa el desarrollo, el programa completo se almacena en la memoria no volátil como Flash EEPROM. Durante la ejecución del programa, la RAM se usa para almacenar variables globales, admitir la asignación dinámica de memoria de las variables y proporcionar una ubicación para la pila (que se discutirá más adelante).

2.6.2. EEPROM direccionable por byte:

Este tipo de memoria se utiliza para almacenar y recuperar permanentemente variables durante la ejecución del programa. Es especialmente útil para registrar el mal funcionamiento del sistema y los datos de fallas durante la ejecución del programa. También es útil para almacenar datos que deben ser retenidos durante una falla de energía pero que deben cambiarse periódicamente. Los ejemplos en los que se usa este tipo de memoria se encuentran en aplicaciones para almacenar parámetros del sistema, combinaciones de cerraduras electrónicas y secuencias automáticas de desbloqueo electrónico de puertas de garaje.

2.6.3. Flash EEPROM:

Flash EEPROM programable se utiliza para almacenar programas. Se puede borrar y programar como un todo. Algunos sistemas de microcontroladores proporcionan un gran complemento de RAM y Flash EEPROM. Por lo tanto, se puede desarrollar un programa de sistema en RAM y luego transferirlo a Flash EEPROM cuando se complete.

Otros microcontroladores proporcionan solo una EEPROM Flash grande y un componente RAM más pequeño. Con esta configuración de memoria, el desarrollo del sistema se lleva a cabo en EEPROM Flash. Flash EEPROM se suele programar utilizando técnicas de programación en sistema (ISP). Es decir, una PC host está conectada mediante un cable a un microcontrolador mientras reside dentro de su circuito de aplicación. La PC host descarga el programa al microcontrolador.

2.7. Recursos especiales

Existen en el mercado mundial diferentes fabricantes de los microcontroladores y numerosas versiones para determinada arquitectura. Las diferencias entre cada fabricante radican en las capacidades de las memorias internas, en otros dispositivos disponen de nuevos recursos, aunque también existen reducciones al mínimo en las prestaciones para la implementación de aplicaciones muy simples. (García Chávida, 2001; Valverde Villarán, 2020)

El trabajo que debe realizar el diseñador es buscar el modelo mínimo para satisfacer todos los requerimientos de su aplicación. En consecuencia, se minimiza el coste, tanto del hardware y software. A continuación, en las secciones se describen los principales recursos específicos que incorporan los microcontroladores.

2.7.1. Temporizadores.

También conocido como Timers, son utilizados para el control de períodos de tiempo y para contar el número de acontecimientos que suceden en el interior. (Barra Zapata & Barra Zapata, 2011; Valverde Villarán, 2020)

2.7.2. Watchdog.

También conocido como perro guardián que en sí es un temporizador que cuando se desborda y pasa por cero provoca un reset automáticamente en el sistema. (Barra Zapata & Barra Zapata, 2011; Valverde Villarán, 2020) El WDT es similar al reinicio externo de borrado maestro (véase la figura 2.11), ya que mantendrá una MCU PIC® en reinicio cuando se active y colocará el contador de programa al comienzo del software del programa (org 0). El WDT se activa cuando se excede su tiempo de espera. Es un temporizador de funcionamiento libre independiente del reloj principal del sistema.

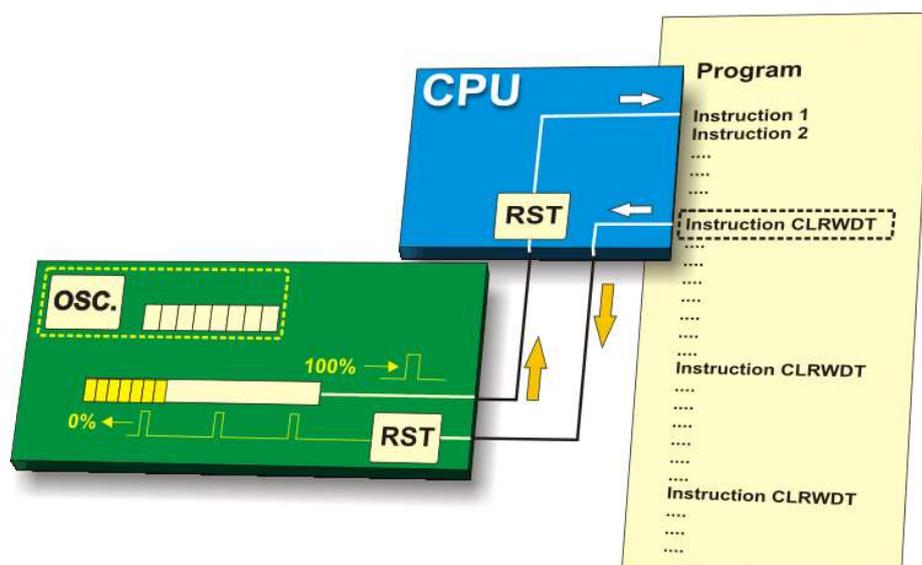


Figura 2. 11: Circuito esquemático para el reinicio externo de borrado maestro.

Fuente: (Verle, 2012b)

Si el funcionamiento del código en el dispositivo se ve afectado por una fuente externa que hace que el software se desvíe de su ruta esperada, el WDT puede reiniciar el dispositivo automáticamente dependiendo de cómo se use el reinicio en el diseño. (Microchip, 2020)

El diseñador de software puede colocar instrucciones para borrar el contador WDT y evitar que ocurra un reinicio de WDT. Es cuando esas instrucciones no se ejecutan, como cuando una interferencia externa afecta el funcionamiento correcto, que el WDT activará y reiniciará el dispositivo.

2.7.3. Estado de bajo consumo.

Estado de bajo consumo: es un estado del sistema donde se detiene el reloj principal y sus circuitos asociados con el objetivo de ahorrar energía en periodos de tiempo donde el microcontrolador se mantiene en espera de instrucciones. (Navarrete Navarrete & Sandoval Vásquez, 2011)

2.7.4. Conversor A/D.

Procesa señales analógicas convirtiéndolas en señales digitales. Muchos microcontroladores están equipados con subsistemas de conversión de analógico/digital (ADC). Este subsistema convierte continuamente señales analógicas variables del mundo exterior en una representación binaria adecuada para su uso por el microcontrolador. Estos convertidores suelen tener una resolución de 8-10 bits. Por lo tanto, una señal continua se convierte en una serie de instantáneas digitales de la señal analógica. Usted, como diseñador del sistema, debe determinar con qué frecuencia iniciar una conversión de ADC para una aplicación determinada.

2.8. Sistemas de comunicaciones en microcontroladores.

Cuando se considera la comunicación entre dos sistemas, la taxonomía consiste en comunicaciones paralelas y en serie. En pocas palabras, un método de comunicación paralela utiliza múltiples canales, cables de bus, para enviar y recibir múltiples flujos de datos simultáneamente, en comparación con un método de comunicación en serie donde solo se envía y recibe un solo flujo de datos a la vez. La ventaja más obvia de un método de

comunicación en paralelo sobre un método de comunicación en serie, siempre que la velocidad de comunicación sea igual, es la velocidad de transferencia de datos.

Usando el número múltiple de conexiones, se puede enviar y recibir la misma cantidad de datos más rápido, proporcional al número de conexiones, que una sola conexión utilizada en un método de comunicación en serie. La desventaja de un método de comunicación paralela es el costo de hardware y software para permitir la transferencia rápida de datos. Típicamente, las técnicas de comunicación paralela se utilizan para la comunicación a corta distancia dentro y fuera de un microcontrolador. Para largo alcance se utiliza una técnica de comunicación en serie para enviar y recibir datos.

2.8.1. Comunicaciones en serie

Hay dos tipos diferentes de comunicaciones en serie disponibles: comunicación sincrónica y asincrónica. El desafío clave en las comunicaciones en serie es mantener la sincronización entre el transmisor y receptor. El método de comunicación asíncrono utiliza un protocolo de bit de inicio y parada para sincronizar un transmisor y receptor. Los bits de sincronización de inicio y detención están integrados en cada byte de señal transmitida.

La ventaja de la comunicación asincrónica es que es económica, pero la desventaja es que las velocidades de transmisión de datos son típicamente más lentas que una comunicación serie síncrona debido a su sobrecarga (bits de inicio y parada). Los subsistemas de comunicación serie asíncrona se denominan transmisor receptor asíncrono universal (*Universal Asynchronous Receiver Transmitter, UART*) o interfaz de comunicación serie (*Serial Communications Interface, SCI*).

La comunicación serie sincrónica utiliza un reloj sincronizado para enviar y recibir cada bit. El receptor de comunicación en serie síncrono es mucho más simple en comparación con un módulo de receptor de comunicación en serie asíncrono. Puede verse como un registro de desplazamiento síncrono

de 16 bits con una mitad de ocho bits que reside en el transmisor y la otra mitad de ocho bits que reside en el receptor.

La desventaja de la comunicación serie sincrónica en comparación con el método de comunicación asincrónica es la tarea de sincronizar los relojes del transmisor y el receptor. Esto generalmente se logra con una línea de reloj adicional que une el transmisor y el receptor. Para una comunicación a larga distancia, no se recomienda la técnica de comunicación en serie síncrona. Para una comunicación de corta distancia, el sistema de comunicación síncrono puede mejorar considerablemente el rendimiento de bits sobre el sistema de comunicación asíncrono. Un sistema de comunicación en serie síncrono se conoce como SPI (interfaz periférica en serie).

Para comprender mejor los conceptos de comunicación en serie, primero debe introducirse alguna terminología. Las restricciones de espacio no permiten incluir una lista exhaustiva de términos asociados con las técnicas de comunicación en serie. Solo se presentan aquellos que uno encontraría fácilmente en la documentación técnica y la literatura.

- a) Modo simplex: en este modo, la comunicación en serie se logra transmitiendo datos en una dirección a la vez.
- b) Modo dúplex: en este modo de comunicación en serie, los datos se pueden transmitir y recibir desde ambos extremos del enlace de comunicación al mismo tiempo.
- c) Tasa de baudios: la tasa de bits enviados o recibidos. Describe el número de bits comunicados por segundo.
- d) Código ASCII: el código del código estándar estadounidense para el intercambio de información se utiliza en la comunicación para codificar alfabetos, números, signos de puntuación y caracteres de control mediante una representación de siete bits. ASCII es un subconjunto del estándar internacional Unicode.
- e) Tiempo de bit: el tiempo requerido para transmitir o recibir un solo bit.
- f) Código de línea en serie: mecanismo de codificación específico utilizado para transmitir y recibir información.

2.9. Lenguaje de programación

En la actualidad el mundo evoluciona cada día, la tecnología no solo se está volviendo cada vez más avanzada, sino también cada vez más común en nuestra vida cotidiana. La invención de productos "inteligentes" está revolucionando el proceso de diseño de casi todos los productos imaginables. Los zapatos que contienen chips que se emparejan con teléfonos inteligentes para realizar un seguimiento de nuestra actividad, los refrigeradores que pueden rastrear cuando caducan los comestibles, y ahora incluso los autos que son capaces de conducir por sí mismos son ejemplos de inventos modernos que utilizan microcontroladores para facilitarnos la vida.

Un microcontrolador es un IC programable, capaz de múltiples funciones dependiendo de cómo esté programado. Existen muchos tipos diferentes de microcontroladores que ofrecen una amplia gama de funcionalidades. La versatilidad del microcontrolador es lo que lo convierte en una de las herramientas más potentes en el diseño moderno. Esta sección explicará los lenguajes de programación de alto nivel.

Las especificaciones de microcontroladores han cambiado poco en las últimas décadas, pero la expansión de los idiomas y el software de soporte ha abierto muchos caminos para el desarrollo de aplicaciones. Durante años, la programación de microcontroladores implicó una elección entre dos lenguajes: C (lenguaje de alto nivel) y ensamblador (lenguaje de máquina). Pero con los nuevos lenguajes ahora disponibles para microprocesadores, los usuarios enfrentan más decisiones en el diseño de sus aplicaciones.

¿Qué idioma es el mejor? No hay una respuesta clara porque depende de la aplicación en la que se trabaje. Para ayudarlo a planificar su desarrollo, aquí hay una exploración de alto nivel de diferentes idiomas disponibles para el uso de microcontroladores, así como recursos para un mayor aprendizaje.

2.9.1. C y ensamblador.

Antes de que los nuevos lenguajes estuvieran disponibles para los microcontroladores, las dos opciones disponibles eran C y ensamblador. C es

un lenguaje de microcontrolador popular a pesar de que se desarrolló a principios de la década de 1970. Para su uso con microcontroladores, C funciona bien porque tiene la capacidad de manejar la asignación de memoria al tiempo que realiza funciones complejas como sentencias if, bucles y expresiones matemáticas.

La mayoría de los estudiantes y profesionales de electrónica utilizan la plataforma de programación CCS C Compiler, que permite realizar la programación C orientada a microcontroladores PIC. Se puede observar un pequeño código de programación.

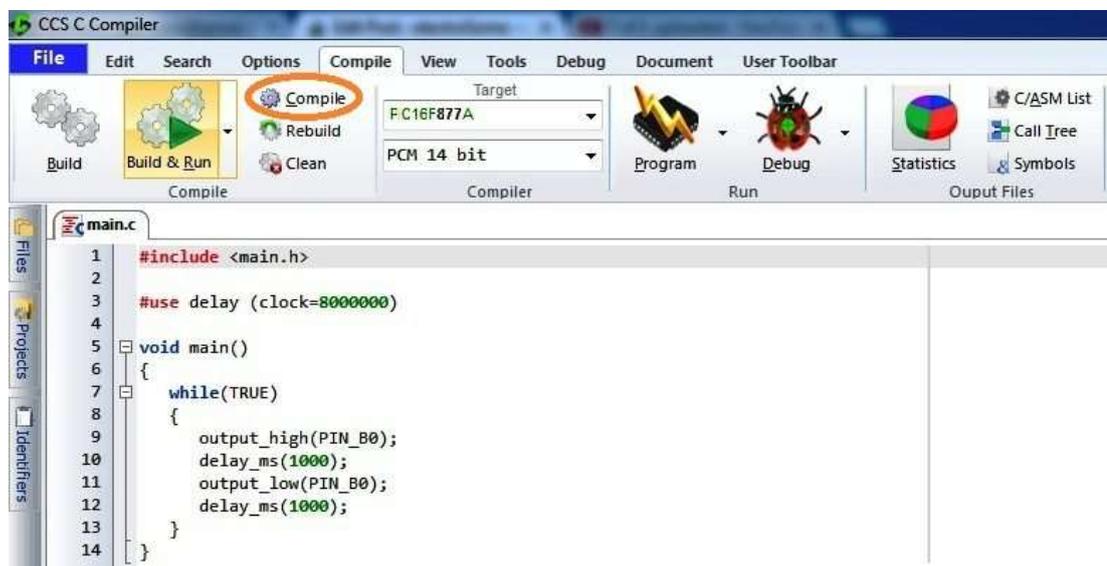


Figura 2. 12: Ventana principal de plataforma de programación CCS C Compiler para lenguaje de alto nivel C.

Fuente: (García Breijo, 2008)

Ensamblador ofrece una alternativa a C con algunas ventajas. Ensamblador es un lenguaje antiguo que a menudo utiliza código específico para el dispositivo, que se convierte mediante un programa de utilidad de ensamblador. Al usar este lenguaje, el usuario tiene acceso directo a la CPU, donde las rutinas se pueden ajustar cuidadosamente para aplicaciones de sincronización precisas. Ensamblador ofrece un mejor uso de la CPU, así como otros conjuntos de instrucciones que C no admite, como la manipulación de bits con instrucciones BIT específicas.

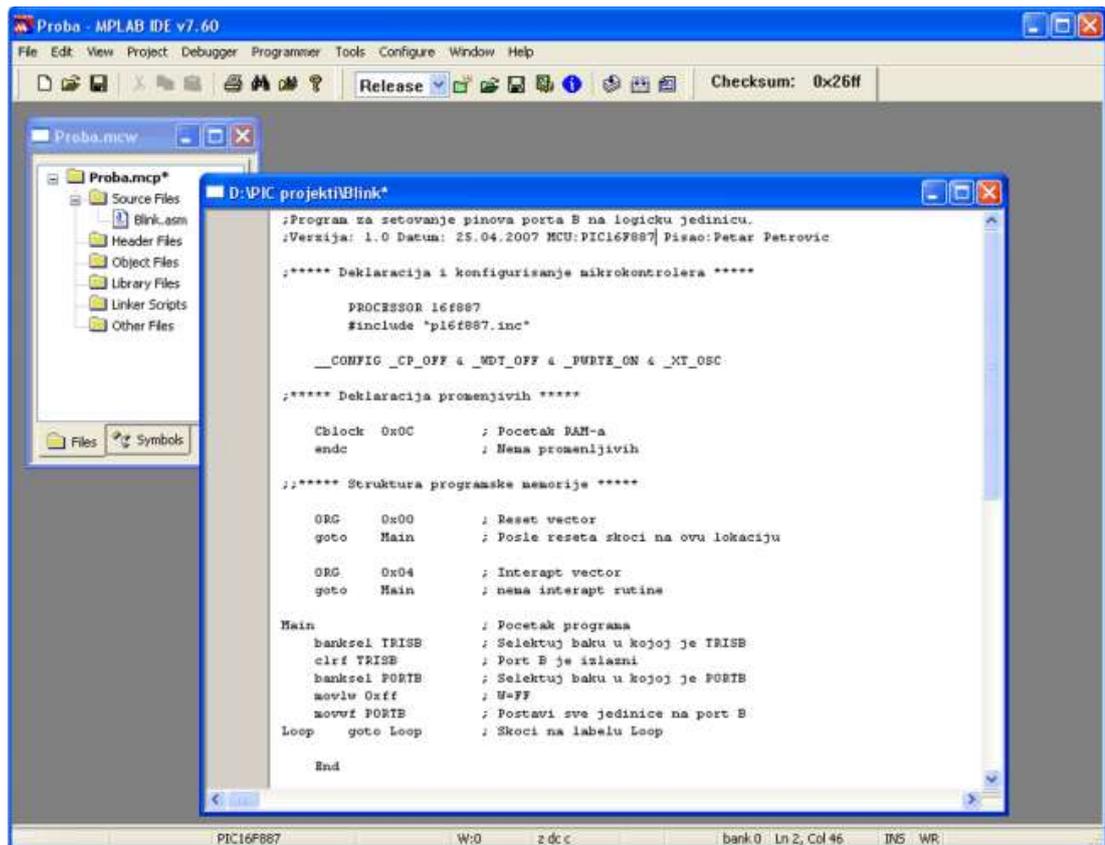


Figura 2. 13: Ventana principal de plataforma de programación MPLAB para lenguaje de máquina.
Fuente: (Verle, 2012b)

2.9.2. MicroPython

Python es un lenguaje interpretado rico en funciones, conocido por su simplicidad, claridad y adaptabilidad de uso general. Estas características también hacen que el lenguaje sea demasiado grande para los microprocesadores. Sin embargo, con la ayuda de una campaña de Kickstarter en 2013, el físico Damien George pudo adaptar Python a una versión más pequeña, MicroPython, que es adecuada para su uso en un microcontrolador.

El código de MicroPython es de código abierto bajo la licencia MIT. El lenguaje se ejecuta en el metal desnudo del microprocesador utilizando un pequeño subconjunto de la biblioteca estándar de Python. Puede funcionar con solo 256k de espacio y 16k RAM, pero está diseñado para ser lo más compatible posible con Python normal.

La figura 2.14 muestra la ventana principal del software de programación Python. Python de 64 bits es un lenguaje de programación dinámico orientado a objetos que se puede utilizar para muchos tipos de desarrollo de software. Ofrece un fuerte soporte para la integración con otros idiomas y herramientas, viene con amplias bibliotecas estándar y se puede aprender en pocos días.

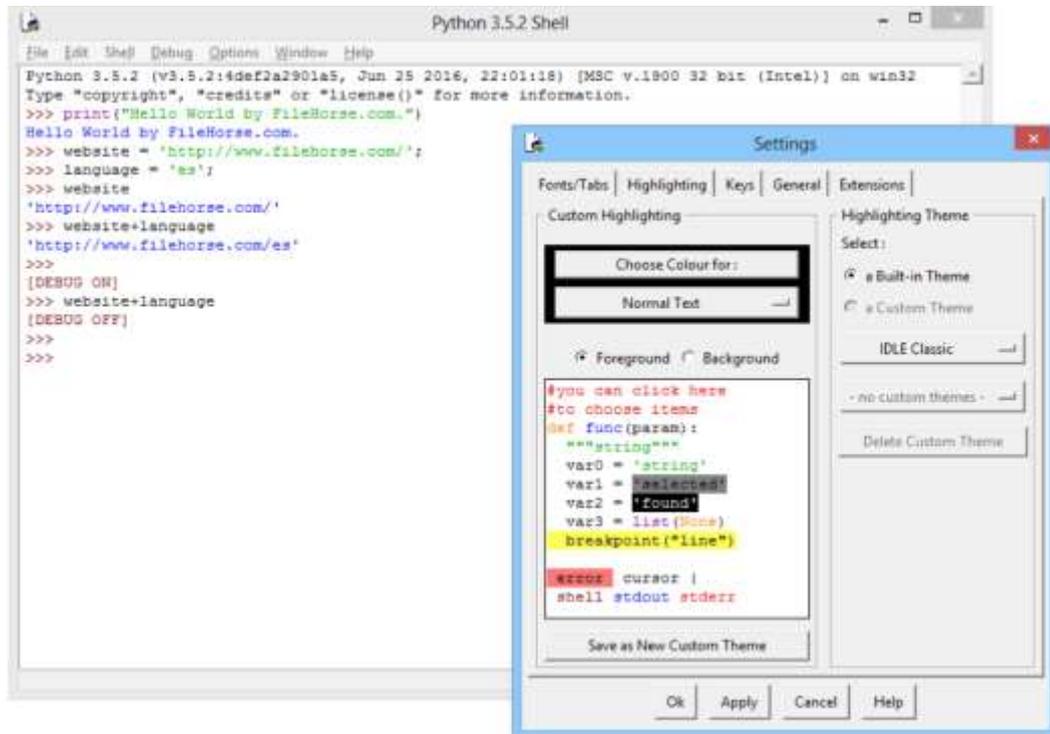


Figura 2. 14: Ventana principal de plataforma de programación Python.
Fuente: (MicroPython, 2020)

Originalmente diseñado para ejecutarse en sistemas basados en ARM, incluidos BBC micro: bit, Adafruit Circuit Playground Express y MicroPython PyBoard, también se ha puesto a disposición para otros procesadores, incluidos ESP8266 y ESP32. MicroPython brilla en la creación de prototipos de proyectos complejos. En el ESP8266, por ejemplo, la creación de proyectos de IoT se puede hacer con algunas llamadas a funciones simples para conectar un proyecto de MicroPython a un punto de Wi-Fi local y comenzar a transmitir información a un agente MQTT.

2.9.3. Blockly

Blockly es una biblioteca de programación de código abierto compatible con Google, cuyo objetivo es hacer que el código sea visualmente lógico mediante el uso de bloques de arrastrar y soltar.

En lugar de confiar en palabras y frases que no tienen sentido en una conversación normal, los usuarios colocan bloques en un entorno y los conectan entre sí para definir cómo fluye un programa tal como se muestra en la figura 2.15. Por ejemplo, en lugar de escribir una declaración IF, los usuarios arrastran el bloque IF al lugar de trabajo y las acciones resultantes también se arrastran dentro del bloque de la declaración IF.

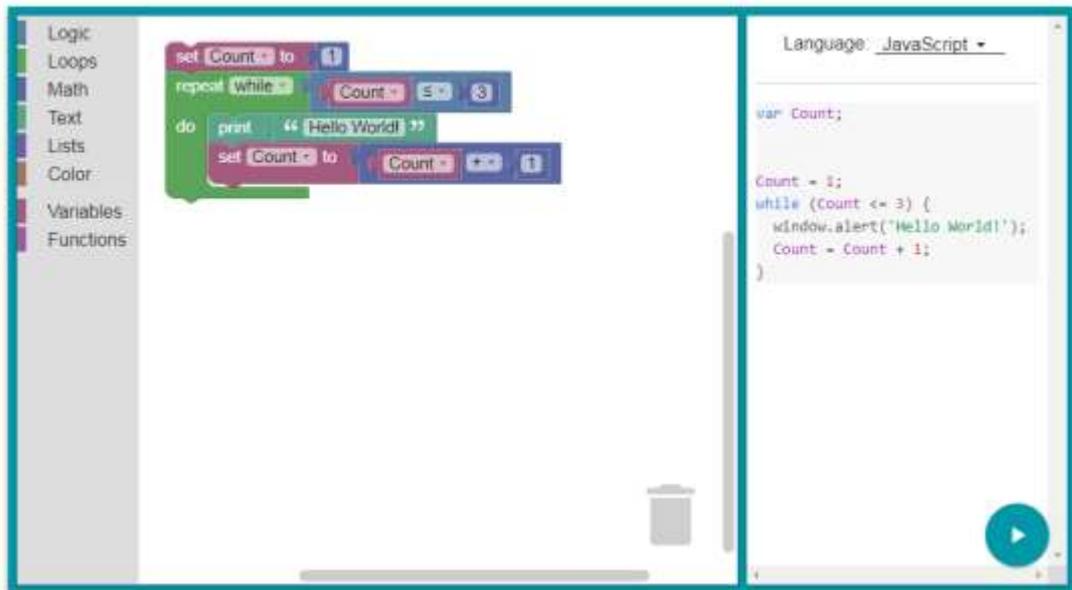


Figura 2. 15: Ventana principal de plataforma de programación Blockly.
Fuente: (All About Circuits, 2020)

2.9.4. Arduino IDE basado en C ++

Si bien a menudo se lo conoce como el "lenguaje de programación Arduino", Arduino en realidad está programado en un buen C ++ antiguo y utiliza un subconjunto de bibliotecas específicas de Arduino.

Las diferencias en Arduino C ++ con respecto a C ++ regular incluyen rutinas genéricas para microcontroladores, ya que las funciones estándar de C ++ de la biblioteca no pueden funcionar con su potencia y memoria limitadas.



Figura 2. 16: Ventana principal de plataforma de programación Arduino IDE basado en C++.

Fuente: (Basto et al., 2015)

2.10. Introducción a los motores paso a paso.

Un motor paso a paso es un motor eléctrico que gira en incrementos de pasos discretos. El movimiento de cada paso es preciso y repetible; por lo tanto, la posición del motor se puede controlar con precisión sin ningún mecanismo de retroalimentación, siempre que el motor esté cuidadosamente dimensionado para la aplicación. Este tipo de control elimina la necesidad de dispositivos caros de detección y retroalimentación, como codificadores ópticos.

La posición se conoce simplemente haciendo un seguimiento de los pulsos de paso de entrada. Es una de las formas más versátiles de sistemas de posicionamiento. Por lo general, se controlan digitalmente como parte de un sistema de circuito abierto, y son más simples y resistentes que los servosistemas de circuito cerrado.

Las aplicaciones industriales incluyen equipos de selección y colocación de alta velocidad y máquinas CNC de varios ejes, que a menudo conducen directamente tornillos de plomo o tornillos de bola. En el campo de la óptica, se utilizan con frecuencia en equipos de posicionamiento de precisión, como actuadores lineales, etapas lineales, etapas de rotación, goniómetros y montajes de espejo.

Otros usos son en maquinaria de envasado y posicionamiento de etapas piloto de válvula para sistemas de control de fluidos. Comercialmente, los motores paso a paso se utilizan en unidades de disquete, escáneres de superficie plana, impresoras de computadora, plotters, máquinas tragamonedas, escáneres de imágenes, unidades de discos compactos y muchos más dispositivos.

Según Jeff Keljik (2007), el motor paso a paso crea un paso de movimiento incrementado en lugar de una rotación suave y sin interrupciones. Concepto básico de motor paso a paso explicado con imán permanente en el rotor con dos conjuntos de polos. Después de que los pulsos de CC energicen el estator, el imán permanente será repelido o atraído para alinearse con los polos magnéticos del estator.

La característica importante del motor paso a paso es la forma en que gira. El motor girará a través de un ángulo fijo para cada pulso aplicado al secuenciador lógico. Cuando reciba el pulso de comando de paso, la secuencia lógica determinará la fase que se activará o desactivará. Luego, envía señales al conductor del motor paso a paso. Si el canal de salida del secuenciador lógico es "ALTO", la potencia funciona para excitar la fase correspondiente del devanado. En la fase del mismo número, no se excitará ni se desactivará si la salida es "BAJA"

2.11. Ventajas y desventajas de los motores paso a paso.

A continuación, se describen las ventajas de usar motores paso a paso:

- a) El ángulo de rotación del motor es proporcional al pulso de entrada.

- b) El motor tiene un par completo en reposo (si los devanados están energizados)
- c) Posicionamiento preciso y repetibilidad del movimiento, ya que los buenos motores paso a paso tienen una precisión del 3 al 5% de un paso y este error no es acumulativo de un paso al siguiente.
- d) Excelente respuesta al arranque / parada / marcha atrás.
- e) Muy confiable ya que no hay cepillos de contacto en el motor. Por lo tanto, la vida del motor depende simplemente de la vida del rodamiento.
- f) La respuesta de los motores a los pulsos de entrada digital proporciona un control de bucle abierto, lo que hace que el motor sea más simple y menos costoso de controlar.
- g) Es posible lograr una rotación síncrona de muy baja velocidad con una carga que está directamente acoplada al eje.
- h) Se puede realizar una amplia gama de velocidades de rotación, ya que la velocidad es proporcional a la frecuencia de los pulsos de entrada.

A continuación, se describen las desventajas de usar motores paso a paso:

- a) Las resonancias pueden ocurrir si no se controlan adecuadamente.
- b) No es fácil de operar a velocidades extremadamente altas.

2.12. Tipos de motor paso a paso

Existen tres tipos básicos de motores paso a paso, que son descritos en las siguientes subsecciones.

2.12.1. Reluctancia variable.

Este tipo de motor paso a paso ha existido durante mucho tiempo. Es probablemente el más fácil de entender desde un punto de vista estructural. La figura 2.17 muestra una sección transversal de un motor paso a paso de reluctancia variable. Este tipo de motor consta de un rotor de dientes múltiples de hierro blando y un estator bobinado. Cuando los devanados del estator se

energizan con corriente continua, los polos se magnetizan. La rotación ocurre cuando los dientes del rotor son atraídos a los polos del estator energizados.

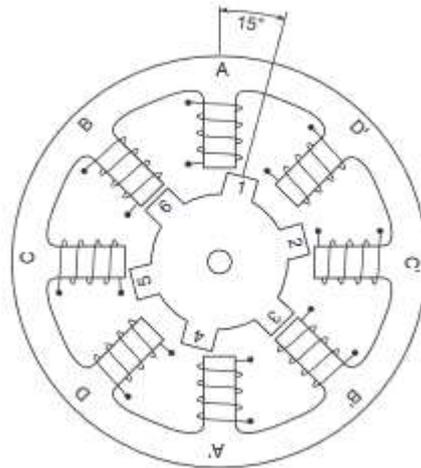


Figura 2. 17: Representación del motor paso a paso de reluctancia variable.
Fuente: (Schmidt et al., 2018)

2.12.2. Imán permanente

A menudo denominado motor de “lata” o “lata”, el motor de paso de imán permanente es un motor de bajo costo y baja resolución con ángulos de paso típicos de 7,5 ° a 15 °. (48 - 24 pasos/revolución). Los motores de imán permanente, como su nombre lo indica, tienen imanes permanentes agregados a la estructura del motor, tal como se muestra en la figura 2.18.

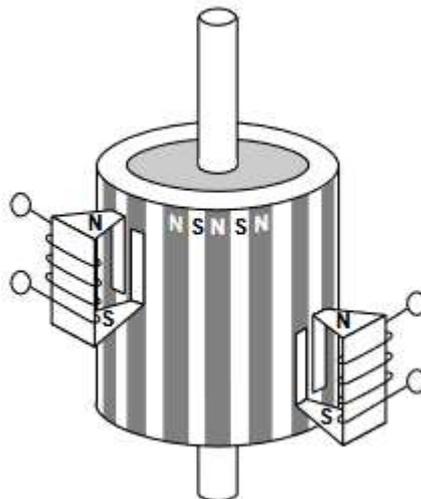


Figura 2. 18: Representación de un motor paso a paso de imán permanente.
Fuente: (Schmidt et al., 2018)

El rotor ya no tiene dientes como con el motor de reluctancia variable. En cambio, el rotor está magnetizado con polos alternos norte y sur situados

en una línea recta paralela al eje del rotor. Estos polos de rotor magnetizados proporcionan una mayor intensidad de flujo magnético y, debido a esto, el motor PM presenta características de par mejoradas en comparación con el tipo VR.

2.12.3. Híbrido

El motor paso a paso híbrido es más caro que el motor paso a paso de imán permanente, pero proporciona un mejor rendimiento con respecto a la resolución de paso, par y velocidad. Los ángulos de paso típicos para el motor paso a paso híbrido varían de 3.6° a 0.9° (100 - 400 pasos por revolución). El motor paso a paso híbrido combina las mejores características de los motores paso a paso de imán permanente y reluctancia variable.

El rotor tiene dientes múltiples como el motor de reluctancia variable y contiene un magneto concéntrico magnetizado axialmente alrededor de su eje, tal como se muestra en la figura 2.19. Los dientes en el rotor proporcionan una ruta aún mejor que ayuda a guiar el flujo magnético a las ubicaciones preferidas en el espacio de aire. Esto aumenta aún más las características de retención, retención y par dinámico del motor en comparación con los tipos de reluctancia variable e imán permanente.

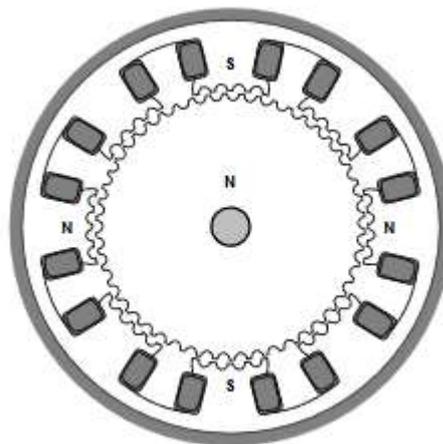


Figura 2. 19: Representación de un motor paso a paso híbrido.

Fuente: (Verle, 2012b)

2.13. Modelo del motor paso a paso.

Esta subsección presenta brevemente las características y el principio de funcionamiento del motor paso a paso, con el fin de obtener las ecuaciones

diferenciales que describen su comportamiento. Como se examinó en detalle en la sección anterior, el motor paso a paso es un actuador eléctrico alimentado por corriente continua, utilizado principalmente para las siguientes aplicaciones: posicionamiento punto a punto (donde se requiere alta aceleración, velocidad y precisión de posicionamiento); control de velocidad; y camino controlado. Hay tres categorías principales de ellos: imán permanente, reluctancia variable e híbrido.

El motor paso a paso de imán permanente presenta polos de rotor y estator que no tienen dientes. En cambio, el rotor tiene polos norte y sur alternativos paralelos al eje del eje del rotor. El motor paso a paso híbrido es una combinación de imán permanente y reluctancia variable. Tiene un rotor dentado de imán permanente, que guía bien el flujo magnético a través del entrehierro. El rotor magnético presenta dos conjuntos de cincuenta dientes cada uno, que están compensados por un cierto ángulo. Un anillo es todos los polos sur, y el otro anillo es todos los polos norte.

En el estator, en cambio, existe el circuito de excitación que consiste en devanados alrededor de los polos, que se utilizan para estimular o desalentar el flujo de flujo de imán a través de ciertos polos de acuerdo con la posición requerida del rotor. El motor paso a paso híbrido tiene dos fases, A y B, que están situadas en cuatro de los ocho polos del estator. Los devanados de cada fase pueden ser excitados por corriente positiva y negativa. Al excitar las fases en secuencia a través de impulsos eléctricos, se produce una fuerza electromagnética que provoca la realineación de los dientes del rotor, que giran en una cantidad fija. Luego, la rotación continua del motor se produce por excitación secuencial de los devanados de fase.

En la figura 2.20 se proporciona una representación genérica del estator y el rotor de un motor paso a paso híbrido. En cambio, en el motor de reluctancia variable, los dientes tienden a realinearse, por lo que se reduce la reticencia de los circuitos magnéticos de la pila (Stack). Cuando los dientes del rotor están alineados con el estator, la reticencia es mínima, mientras que el flujo magnético en la pila es máximo. El número de dientes del rotor es

diferente del del estator, por lo que no están alineados al inicio. Después de introducir el actuador y su principio de funcionamiento físico, se definen las ecuaciones diferenciales del modelo dinámico.

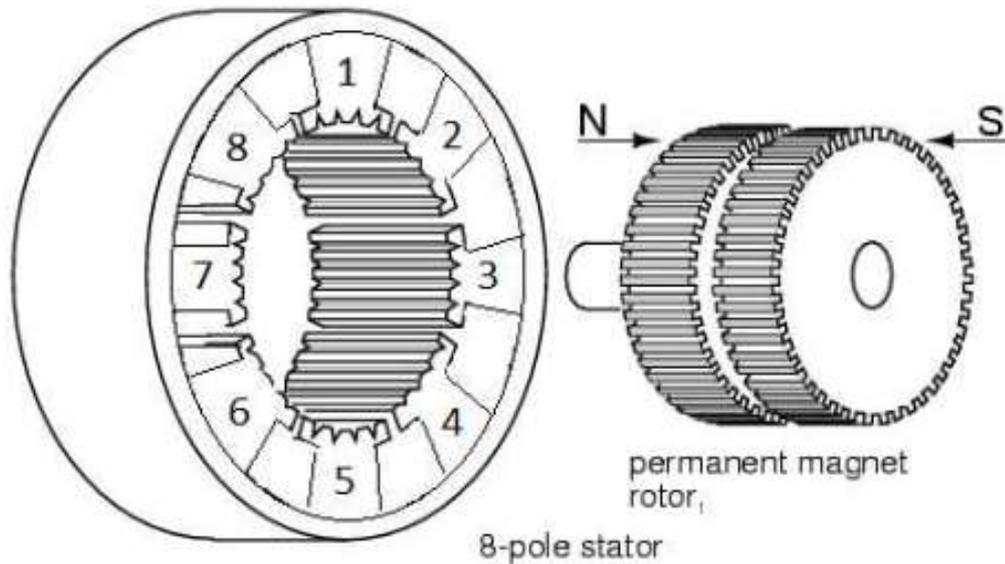


Figura 2. 20: Representación de un motor paso a paso bipolar híbrido genérico. A la izquierda se muestra el estator con los ocho polos, mientras que a la derecha se representa el rotor dentado.

Fuente: (Schmidt et al., 2018)

Primero, se considera el circuito de excitación equivalente representado por la siguiente ecuación:

$$V_{eq} = RI_{eq} + L \frac{dI_{eq}}{dt} - e_m$$

$$e_m = K_m \omega \sin(N\theta + \varphi)$$

Donde se enumeran los siguientes parámetros:

R , es la resistencia del circuito de excitación [ohm];

L , es la Inductancia del circuito de excitación [H];

e_m , es la fuerza electromotriz [V];

K_m , es la constante del motor [Nm/Arad];

ω , es la velocidad angular del motor; [rads];

θ , es la posición angular del motor [rad];

φ , es la fase;

N , es el número de pares de polos del rotor;

Como el circuito de excitación está compuesto por dos fases A y B, está representado por las siguientes ecuaciones diferenciales:

$$V_a = RI_a + L \frac{dI_a}{dt} - K_m \omega \sin(N\vartheta + \varphi)$$

$$V_b = RI_b + L \frac{dI_b}{dt} + K_m \omega \sin(N\vartheta + \varphi)$$

Una vez que se han determinado las ecuaciones del componente eléctrico, es posible definir la del par mecánico. Suponiendo que el circuito magnético es lineal y que la inductancia mutua entre las dos fases es insignificante, puede considerarse como la suma de dos componentes.

$$C_m = C_a + C_b$$

Donde, el par mecánico generado por las dos fases son:

$$C_a = -\frac{e_a I_a}{\omega} = -K_m I_a \sin(N\vartheta)$$

$$C_b = -\frac{e_b I_b}{\omega} = K_m I_b \cos(N\vartheta)$$

En consecuencia, la expresión final resultante es la siguiente:

$$C_m = -K_m I_a \sin(N\vartheta) + K_m I_b \cos(N\vartheta)$$

Conociendo el par mecánico, se encuentra la ecuación diferencial del motor paso a paso. Se obtiene de un equilibrio dinámico simple entre el lado del motor y el lado de la carga, suponiendo que este último esté conectado por una unión rígida.

$$J_m \left(\frac{d\omega}{dt} \right) + D \left(\frac{d\vartheta}{dt} \right) = C_m - C_r$$

Donde el término D (dθ/dt) representa una disipación de energía causada por el componente intrínseco de fricción viscosa en el lado del motor. Además, la estimación del parámetro D es compleja. El modelo matemático

que describe el comportamiento de un motor paso a paso híbrido bipolar está representado por el siguiente sistema de ecuación diferencial no lineal:

$$\begin{cases} \frac{dI_a}{dt} = \frac{[V_a - RI_a + K_m\omega \sin(N\vartheta)]}{L} \\ \frac{dI_b}{dt} = \frac{[V_b - RI_b + K_m\omega \cos(N\vartheta)]}{L} \\ \frac{d\omega}{dt} = \frac{[-K_m I_a \sin(N\vartheta) + K_m I_b \cos(N\vartheta) - D\omega - C_r]}{J} \\ \frac{d\vartheta}{dt} = \omega \end{cases}$$

2.14. Controlador de motor paso a paso.

La función del controlador del motor paso a paso es controlar el movimiento de este. El controlador del motor paso a paso recibirá pasos y señales de direcciones de un sistema de control usando una computadora. Según Giorgos Lazaridis (2010), la energía eléctrica se convertirá en energía mecánica utilizando un motor paso a paso. La diferencia entre el motor paso a paso y otros motores es la forma en que funcionan.

El motor paso a paso no gira continuamente y cada paso es una fracción de un círculo completo. Cada paso del eje del motor paso a paso necesita un pulso. Por lo general, un motor estándar de 200 pasos necesitará 200 pasos para completar una revolución del eje del motor paso a paso en su totalidad. La rotación y la velocidad del eje del motor paso a paso son directamente proporcionales a la frecuencia del pulso

Desde el controlador del motor paso a paso, la corriente fluirá hacia el devanado, por lo que se puede determinar la velocidad y el par de un motor paso a paso. La inductancia reducirá el tiempo que tarda la corriente durante la activación del proceso de bobinado. Básicamente, el circuito controlador del motor paso a paso está diseñado para suministrar una gran cantidad de voltaje en lugar del voltaje nominal del motor paso a paso.

Si el controlador del motor paso a paso proporciona un voltaje de salida más alto, esto conducirá a un mayor nivel de par versus velocidad. El voltaje

de salida del controlador del motor paso a paso también conocido como voltaje de bus. Además de eso, debe tener una clasificación de cinco a diez veces mayor que la clasificación de voltaje del motor paso a paso. La corriente del controlador del motor paso a paso debe limitarse a la clasificación de corriente del motor paso a paso para proteger el motor paso a paso.

Según Austin Hughes (1990), existen tres tipos de controladores que se utilizan comúnmente en el sector industrial, como el accionamiento de voltaje constante, el accionamiento forzado de corriente y el accionamiento chopper. Todos los tipos de controladores necesitan un transistor que actúe como interruptores que se enciendan por completo o se apaguen.

El controlador de voltaje constante es el tipo más simple de controlador. El controlador proporciona una aproximación razonablemente buena a una forma de onda de corriente rectangular a baja velocidad de avance. A una frecuencia más alta, el período de "encendido" es corto en comparación con la constante de tiempo de bobinado y la forma de onda actual se degenerará. Cuando la velocidad de extracción del par cae rápidamente, significa que el motor está limitado a una operación de baja velocidad.

Para el tipo de controlador de corriente forzada, necesitaban un voltaje de suministro más alto para aumentar la tasa de aumento de corriente durante el estado de encendido. Se debe agregar una resistencia adicional en serie con el devanado para evitar que la corriente exceda el valor nominal. El controlador Chopper utiliza un voltaje de alta potencia con dos transistores de conmutación conectados en cada fase. El primer transistor se enciende durante todo el período durante el cual se requiere corriente. Luego, el segundo transistor se activará cuando la corriente real caiga por debajo del umbral.

Capítulo 3: Diseño, Implementación y resultados

3.1. Descripción general de sistema estereotáxicos.

Antes de describir el procedimiento estereotáxico se debe saber que a partir de una tomografía y/o resonancia magnética cerebral se detectan los tumores que pueden ser benignos o malignos. Y van a depender del procedimiento quirúrgico estereotáxico para tener un diagnóstico certero mediante el estudio histopatológico de la presencia o no de un cáncer cerebral o también llamado glioblastoma. La figura 3.1 se muestra la imagen de una tomografía donde se puede apreciar la presencia de un tumor cerebral.

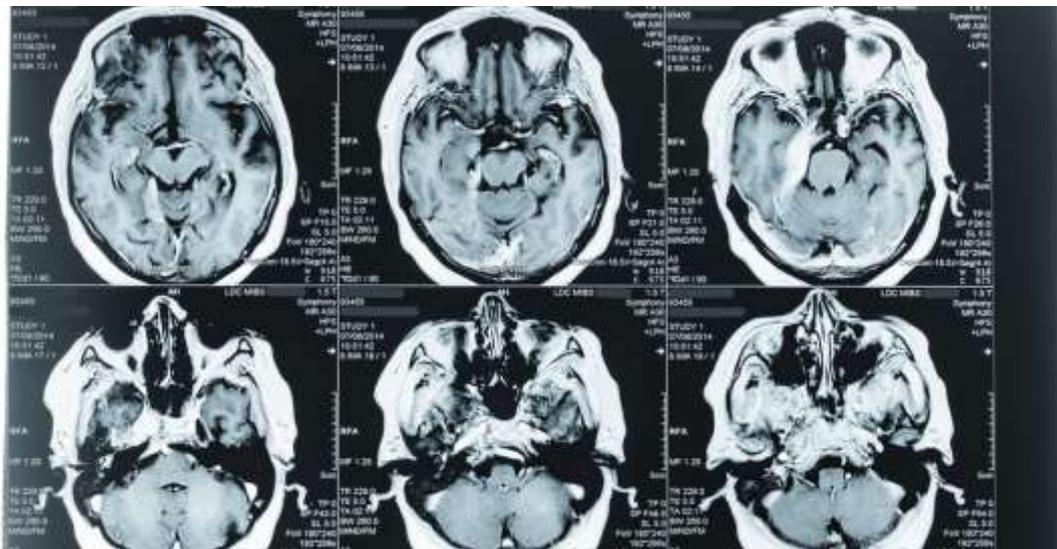


Figura 3. 1: Ejemplo de una imagen de un tumor cerebral
Fuente: (MSP, 2017).

La figura 3.2 muestra el equipo del sistema de obtención estereotáxica controlado fabricado por la empresa FI. ME de Argentina. A pesar de la seguridad comprobada de la biopsia cerebral estereotáctica, debe hacerse un intento razonable para diagnosticar una lesión antes se realiza un procedimiento invasivo. La consideración también debe ser dado a cómo un diagnóstico de biopsia puede influir en el curso de terapia para el paciente. Los procedimientos e indicaciones para la biopsia cerebral estereotáctica han sido bien establecida.

La figura 3.3 muestra a un paciente para proceder con la tomografía estereotáxica utilizando el equipo de obtención estereotáxica FI. ME.

La biopsia estereotáctica puede ir acompañada de ciertas complicaciones como hemorragia, déficit neurológico, ataques, e infección. La muerte después de una biopsia estereotáctica es muy rara. Como resultado de un aumento del edema cerebral o hemorragia intracraneal. La complicación más frecuente después de la biopsia estereotáctica es hemorragia. Post-biopsia reporta tasas de hemorragia que han oscilado entre 1.2 y 59.8%.



Figura 3. 2: Imagen del sistema de obtención estereotáctica controlada.
Fuente: (FI. ME., 2018)



Figura 3. 3: Imagen de una tomografía estereotáctica
Elaborado por: Autor.

La hemorragia después de la biopsia estereotáctica se ha clasificado en, por un lado, ya sea en hemorragia intratumoral (relacionada con biopsia) o hemorragia a lo largo de la trayectoria de la aguja (relacionada con la trayectoria). Por otro lado, se puede clasificar en sintomático cuando causa un nuevo déficit neurológico o provoca el deterioro de un déficit neurológico preexistente y en asintomático o silencioso cuando solo se descubre en

imágenes de tomografía computarizada de rutina después de la biopsia sin cambio clínico, Además, la hemorragia podría ser clasificado por localización en hemorragia intralesional, intraparenquimatosa hemorragia extralesional, hemorragia intraventricular, hemorragia subaracnoidea (SAH), subdural hemorragia (SDH) y hemorragia epidural (EDH).

La mayoría de los neurocirujanos no realizan imágenes Post-biopsia a menos que sus pacientes experimenten nuevos síntomas. La temporización de la tomografía computarizada posterior a la biopsia no es consistente en anteriores estudios. Todas las biopsias cerebrales estereotáxicas deben de constar con una serie de equipos como el sistema estereotáxico (véase figura 3.2), un software de visualización de imágenes médicas estereotáxico (véase las figuras 3.4 a 3.7) y la cánula de biopsia cerebral (véase la figura 3.8).

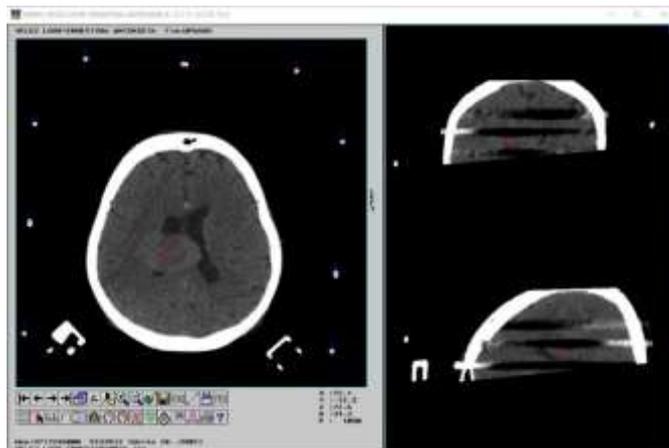


Figura 3. 4: Imagen de selección de blanco quirúrgico.
Elaborado por: Autor.

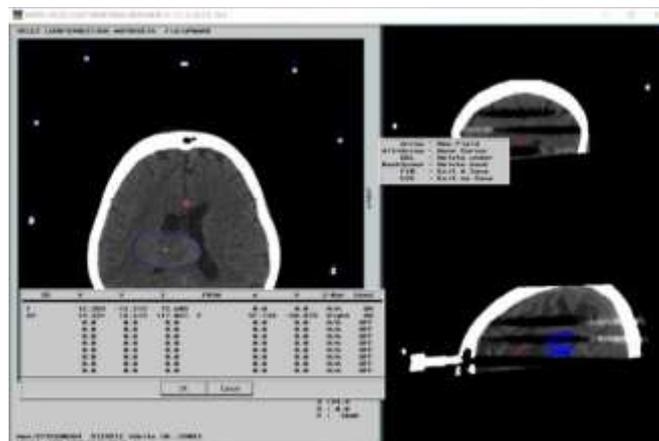


Figura 3. 5: Imagen de las coordenadas del blanco quirúrgico.
Elaborado por: Autor.

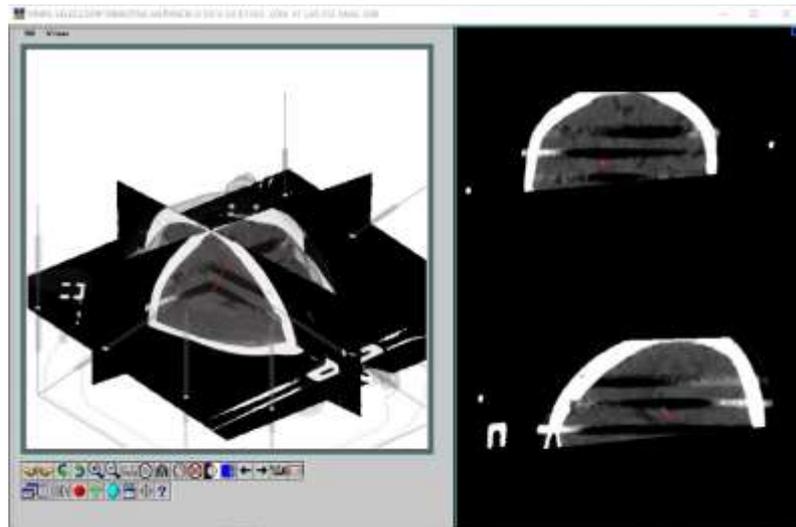


Figura 3. 6: Imagen triplanar
Elaborado por: Autor.

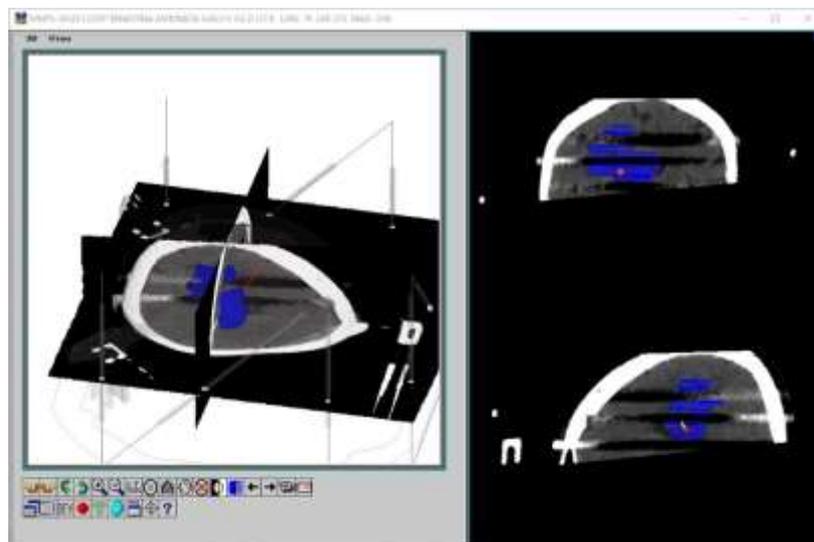


Figura 3. 7: Imagen de reconstrucción volumétrica y trayectoria
Elaborado por: Autor.



Figura 3. 8: Cánula de biopsia cerebral
Elaborado por: Autor.

Existen diversos tipos y modelos en el mercado mundial. El sistema referente para este estudio fue el sistema estereotáxico BLUE RIM (véase la figura 3.2), el software medico estereotáxico para planeamiento MEVIS (ver figuras 3.4 a 3.7) y una cánula de biopsias cerebrales MICROMAR (ver figura 3.8). Para realizar el procedimiento de biopsia cerebral estereotáxica (véase la figura 3.8), el medico neurocirujano tiene una imagen previa que muestra una tumoración u anomalía en el encéfalo, ya sea una resonancia nuclear magnética o una tomografía computarizada donde se observa una anomalía en el encéfalo y lo lleva a este a tomar la decisión de practicarle al paciente una biopsia cerebral.



Figura 3. 9: Imagen de un procedimiento quirúrgico estereotáxico.
Elaborado por: Autor.

Luego se procede a la preparación para la biopsia, donde se le coloca un marco estereotáxico (hardware) fijado a la teca externa de los huesos del cráneo, existen modelos y/o técnicas quirúrgicas de abordaje de 3 o 4 puntos de fijación dependiendo de las necesidades del médico. Todo esto bajo anestesia local en cada punto de fijación y sedación. Posterior se le procede a realizar una tomografía axial computarizada con cortes axiales milimétricos de 0.5mm de espesor, los cuales son exportados en formato DICOM (Digital Imaging and Communication On Medicine)

Una vez con las imágenes se las importa y procesa en el software MEVIS, de planeamiento estereotáxico donde se realiza la selección del blanco quirúrgico, así mismo la trayectoria, punto de entrada, selección de herramienta a utilizar para la biopsia. Este sistema permite además crear una reconstrucción volumétrica del tumor cerebral, lo cual es de mucha utilidad para el neurocirujano en el momento del procedimiento, ya que le permite visualizar las estructuras vecinas en áreas elocuentes o vasos cerebrales importantes.

Posterior a esto se procede a realizar, en un quirófano, la preparación para la operación como tal, se procede a realizar una incisión en la piel, se realiza un agujero de trepano de 14 mm de diámetro, se procede a realizar una incisión en la duramadre, el neurocirujano realiza una comprobación de las coordenadas y las traslada al sistema estereotáxico (hardware), una vez todo está confirmado se procede a introducir una cánula de la ventana lateral. Consiste en una cánula interna y una externa. Ambas cánulas contienen una ventana lateral de 10 mm de largo en su extremo distal.

La rotación de la cánula interna alinea las dos ventanas o las cierra. Una vez que se alcanza el objetivo estereotáxico, la jeringa se usa para aspirar el tejido vecino. A continuación, la cánula interna se gira para cortar el tejido. Después de eso, se retira la cánula interna y se obtiene la muestra. Si es necesario, la herramienta de biopsia se reinserta y se corta una muestra adicional fuera. Una vez obtenidas las muestras se procede a retirar la cánula de biopsia cerebrales, todo el procedimiento se realiza con el paciente despierto, ya que es necesario evaluar la función cerebral del paciente.

3.2. Diseño de la base del sistema de microaspiración estereotáxica.

Para diseñar la base del dispositivo, se utiliza un sistema de diseño asistido por computadora (CAD) el cual permite la creación de objetos en 3D y tener una visión general del prototipo. Este cuenta con una base y un sistema de riel. La base cuenta (véase la figura 3.10) con dos agujeros que atraviesan perpendicularmente y en el cual van las barras de acero inoxidable de 12mm, posee también un agujero por el cual se va a enlazar el tornillo sin

fin de 8 mm y el motor de pasos NEMA 17, para el motor se le realizan agujeros de 3 mm para fijar el mismo a la base mediante tornillos M 3x15mm de acero inoxidable. En la parte superior cuenta con el receptor de la jeringuilla de insulina.

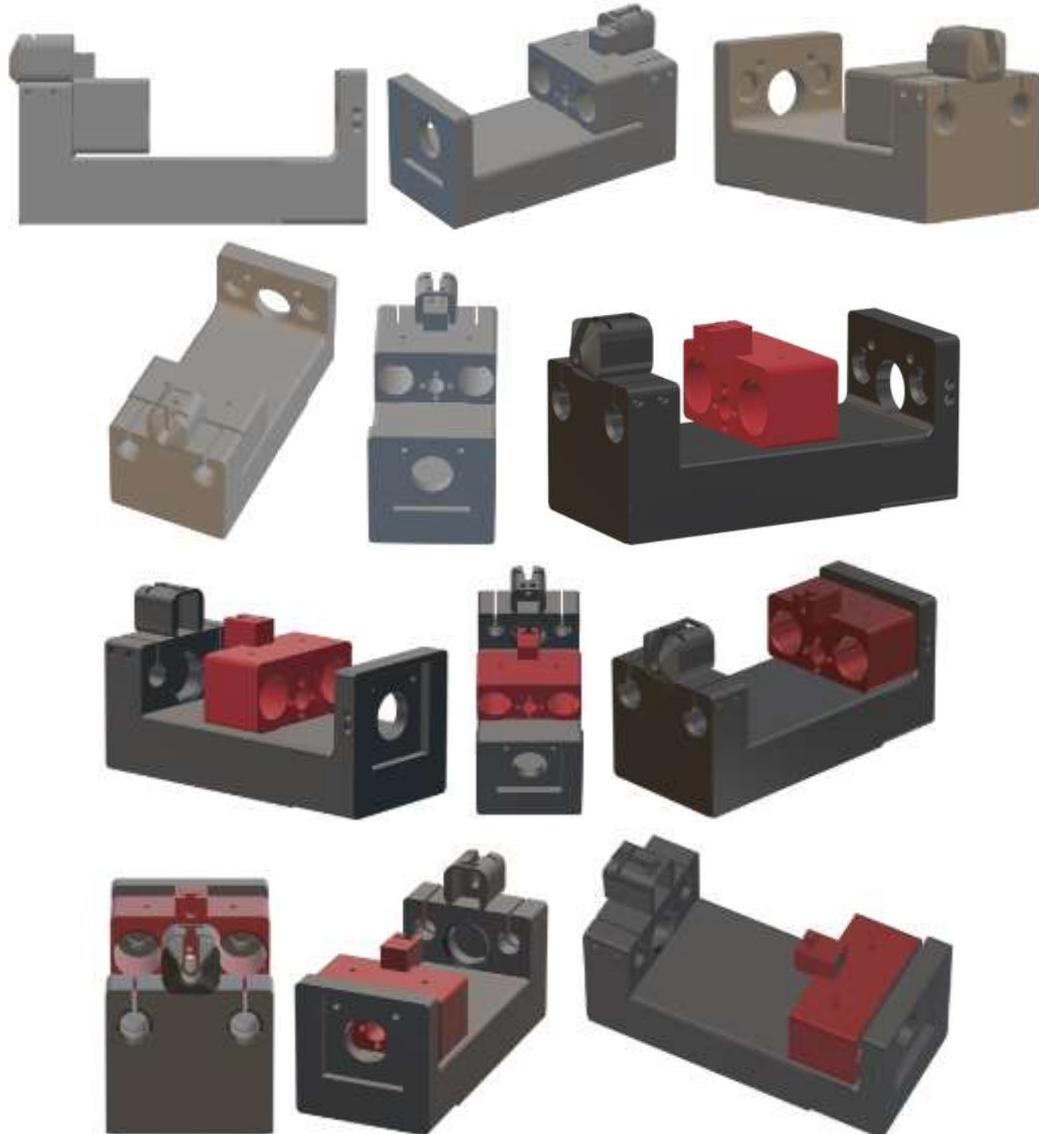


Figura 3. 10: Diseño 3D en AutoCAD en diferentes posiciones de la base y riel del sistema de microaspiración estereotóxica controlada.

Elaborado por: Autor.

3.3. Diseño del riel.

El sistema de riel cuenta también con dos agujeros (véase la figura 3.10) donde se colocarán los rodamientos para las barras de acero inoxidable y el acople del tornillo sin fin, el cual permitirá el desplazamiento de este. En la parte superior cuenta con un receptor para colocar el extremo del embolo de

la jeringuilla de insulina para tener control sobre la misma y realizar la aspiración.

3.4. Materiales utilizados en la implementación del sistema de microaspiración.

El diseño mostrado en las secciones 3.2 y 3.3 (base y riel) del sistema de microaspiración estereotáxica controlado no funcionaría sin los elementos que se muestran en las figuras 3.11 a 3.15. La figura 3.11 muestra los rodamientos para el deslizamiento de la barra acerada en el riel.



Figura 3. 11: Bujes o rodamientos para que se deslice la barra acerada de 12 mm.
Elaborado por: Autor.

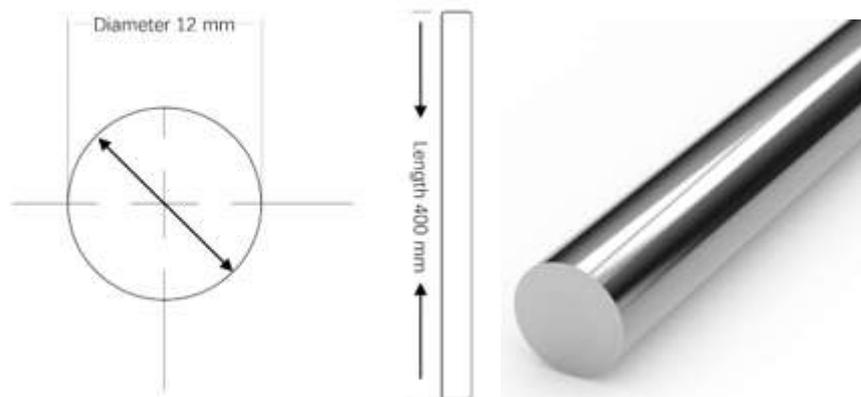


Figura 3. 12: Barra acerada de 12 mm para el desplazamiento del riel.
Elaborado por: Autor.

La figura 3.13 muestra el motor de paso bipolar que realiza el giro en sentido de las manecillas de reloj hasta llegar a una distancia aproximada de 5 cm para la toma de muestra controlada.

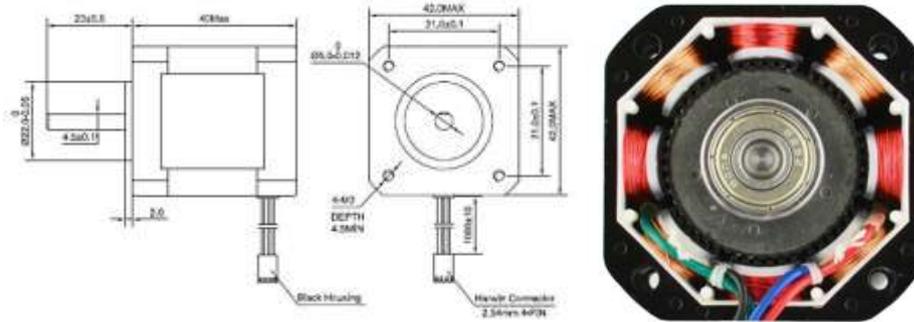


Figura 3. 13: Motor de paso NEMA 17 bipolar.
Elaborado por: Autor.

Las figuras 3.14 y 3.15 muestran al tornillo sin fin y a la tuerca que sujeta al mismo.

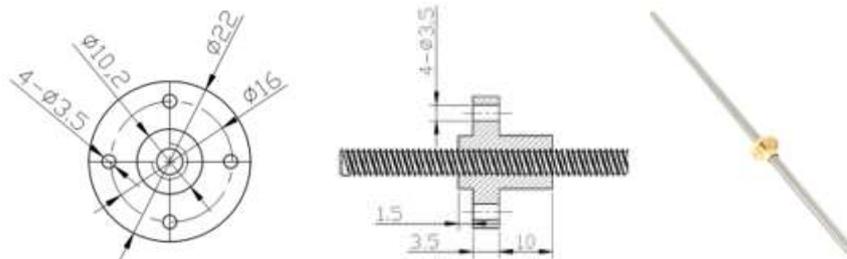


Figura 3. 14: Tornillo sin fin para ejecución del movimiento del riel.
Elaborado por: Autor.

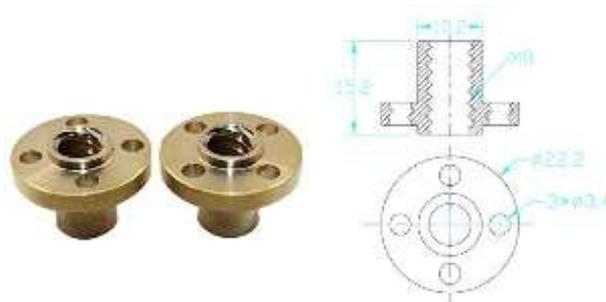


Figura 3. 15: Tuerca para el tornillo sin fin.
Elaborado por: Autor.

3.5. Diseño electrónico que opera al sistema de microaspiración estereotáxico controlado.

La figura 3.16 muestra el diagrama esquemático en Proteus del diseño electrónico (hardware) para el sistema de microaspiración estereotáxico controlado. Este diseño permitió ver el comportamiento del sistema y de la funcionalidad del código de programación. De la figura 3.16 se observan algunos dispositivos (PIC, LCD, controlador o speed control, motor DC, teclado matricial, juego de resistencias) que se describen uno a uno.

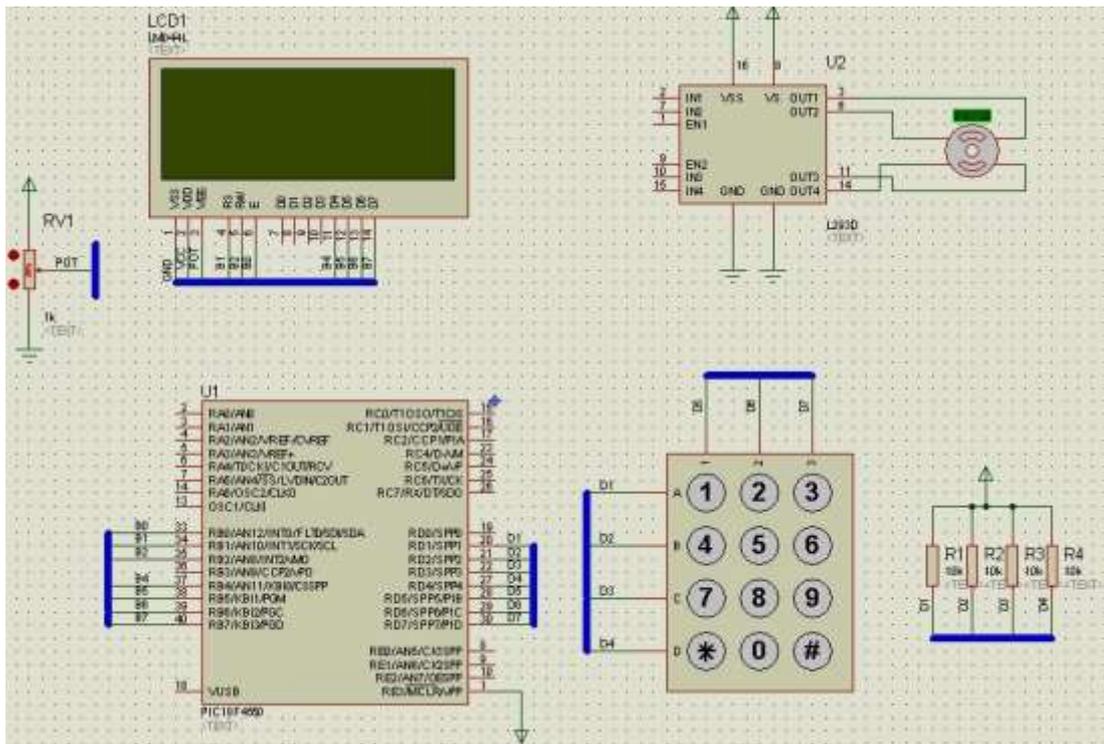


Figura 3. 16: Diagrama esquemático en Proteus del diseño electrónico para sistema microaspiración estereotáxica controlado.

Elaborado por: Autor.

El PIC18F4550 (aquí denominado 4550) es un microcontrolador (MCU) de 8 bits con arquitectura Harvard, es decir, tiene códigos y espacios y vías de datos separados. Está comúnmente disponible en paquetes DIP de 40 pines, TQFP de 44 pines y QFN. A continuación, algunas de sus características:

- a) Alimentación de voltaje entre 2.5 - 5.5 V_{DC}, adecuado para aplicaciones de bastante baja potencia.
- b) La estructura flexible del oscilador tiene un generador de reloj en chip y también puede utilizar un oscilador de cristal externo.
- c) Tres interrupciones externas, cuatro temporizadores, comparadores analógicos duales con entradas multiplexadas, puerto serie síncrono maestro (MSSP) que admite comunicaciones SPI e I²C (todos los modos).
- d) Convertidor A/D de 10 bits y 13 canales en chip, captura/compara módulos PWM.
- e) Memoria de programa en chip de 32 KB (instrucciones), memoria de programa de 16 KB (bytes), con memoria de datos de 2 KB. Admite

la capacidad de programación en el sistema (ISP), con facilidad para la depuración en circuito.

Cabe recalcar que los elementos del diseño en Proteus asemejan a los dispositivos reales que se emplean en el presente proyecto. La pantalla de cristal líquido llamado LCD se utiliza para observar varios mensajes programados previamente en lenguaje de alto nivel, y el software de programación empleado es CCS C Compiler. También, en la figura 3.16 se incluye el controlador de velocidad (llamado speed-control) del motor DC (para efectos de simulación del motor de paso que Proteus no dispone en su librería). El teclado matricial permite ingresar los valores necesarios para la ejecución de las rutinas programada en CCS C Compiler.

A continuación, se muestran las visualizaciones en pantalla LCD de la simulación en Proteus para el sistema de microaspiración estereotáxica controlado. La figura 3.17 muestra el mensaje de presentación inicial del sistema propuesto.

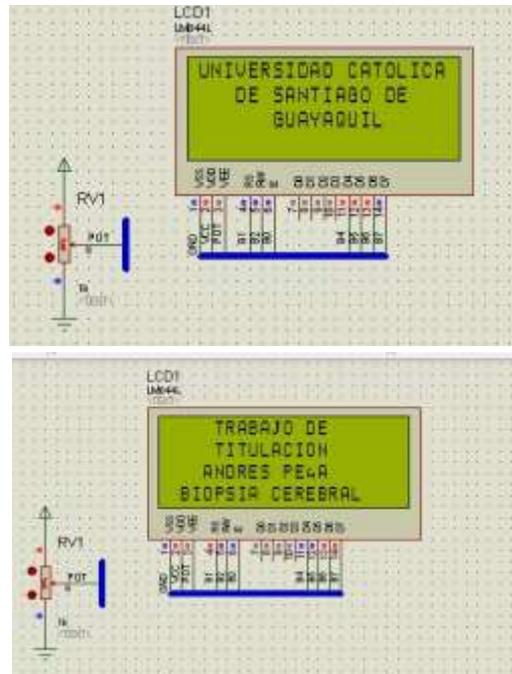


Figura 3. 17: Mensaje inicial del sistema propuesto.
Elaborado por: Autor.

La figura 3.18 muestra el mensaje del menú principal como sistema embebido para la microaspiración estereotáxica controlada. En el menú

principal se observa el tipo de tejido que se desea seleccionar para la microaspiración, que puede ser cerebral, huesos, hígado y pulmón (pero este proyecto se enfoca en cerebral, el resto se puede desplegar a futuros proyectos que analicen biopsias a huesos, hígado y pulmón).

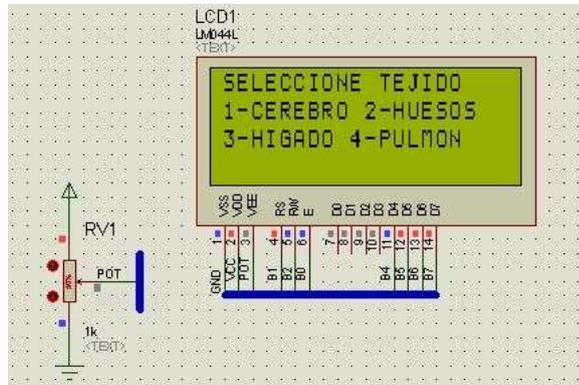


Figura 3. 18: Menú principal del sistema propuesto.
Elaborado por: Autor.

La figura 3.19 muestra la pantalla para el inicio de la biopsia que envía las señales de control necesarias al sistema de microaspiración estereotáxica controlado. El mensaje mostrado indica el inicio de la biopsia presionando una tecla para la toma de muestra cerebral.



Figura 3. 19: Inicio de biopsia para obtención de muestra cerebral.
Elaborado por: Autor.

La figura 3.20 muestra al LCD indicando el mensaje de retirando la muestra y el avance del movimiento del motor de paso en el que recorre el émbolo (parte de la jeringuilla) sobre la base y el riel del sistema de microaspiración estereotáxica controlada. En la simulación se observa que el émbolo ha recorrido 4.19 cm para extracción de la muestra del tumor cerebral.

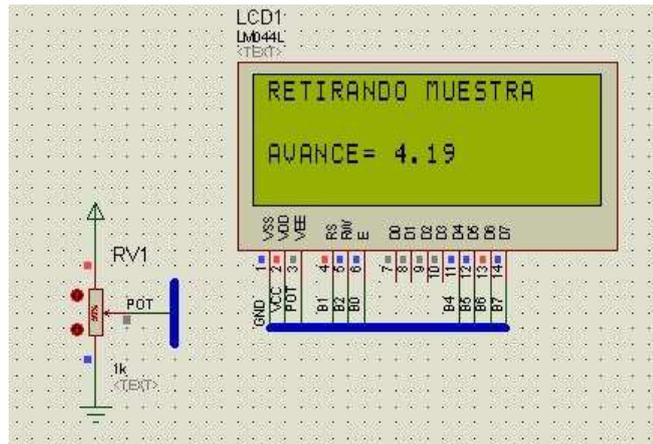


Figura 3. 20: Mensaje de recorrido del émbolo en la base y riel del sistema implementado.
Elaborado por: Autor.

Las figuras 3.21 y 3.22 muestran los mensajes de biopsia estereotáxica controlada que ha sido finalizada al presionar cualquier tecla del teclado matricial.

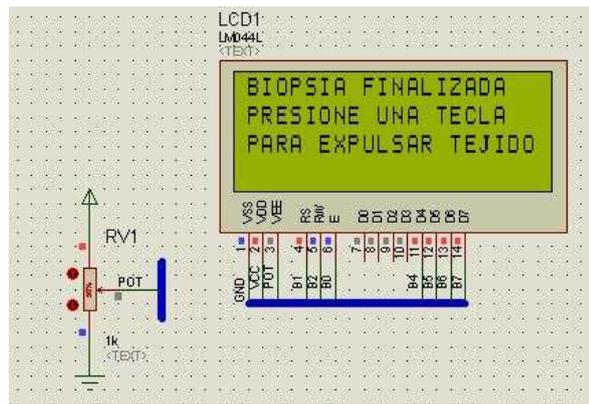


Figura 3. 21: Mensaje de finalización de biopsia mediante microaspiración estereotáxica controlada.
Elaborado por: Autor.

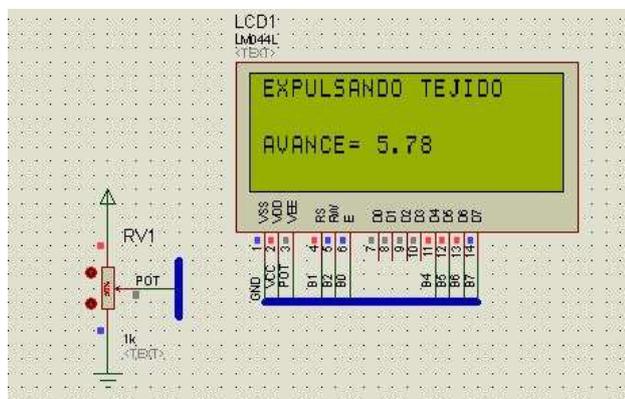


Figura 3. 22: Mensaje de expulsión del tejido o muestra cerebral.
Elaborado por: Autor.

La figura 3.23 muestra el diseño final de la PCB del sistema embebido para el control de microaspiración estereotáctica controlada.

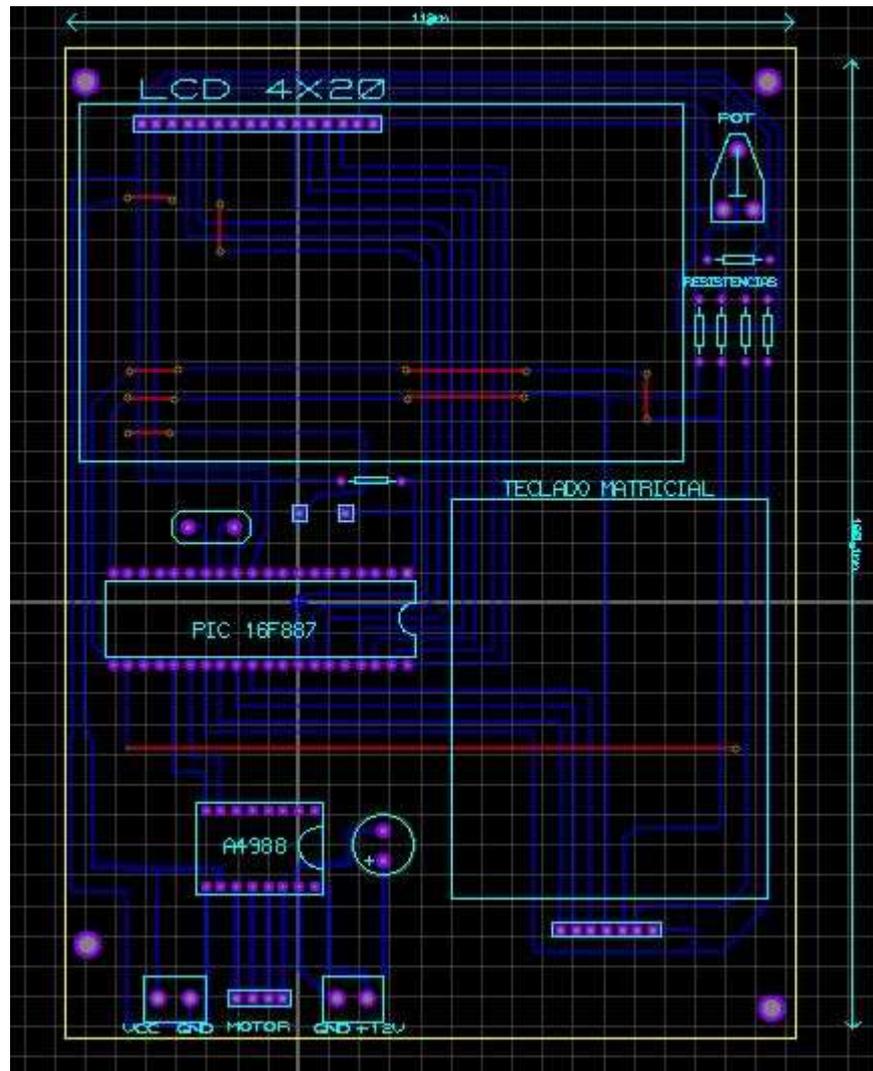


Figura 3. 23: Diseño de la PCB del sistema de microaspiración estereotáctica controlada.

Elaborado por: Autor.

3.6. Programación del sistema de microaspiración en CCS C Compiler.

En esta sección se presenta las líneas de programación para controlar al sistema de microaspiración estereotáctica. Antes de mostrar la programación se explica el diagrama ASM (véase la figura 3.24) del sistema propuesto. Primero se muestra la presentación general del sistema, después pide seleccionar el tipo de tejido, que puede ser: cerebral, huesos, hígado y pulmón. Posteriormente, para cada tipo de tejido se pulsa OK y se toma la muestra de este mediante el movimiento del motor de paso NEMA17. Si llega a 60 mm el microcontrolador envía la señal para la expulsión de la muestra

realizada por el motor NEMA17. En el resto de los tejidos deberá comportarse de manera similar, esto serviría para ampliar este trabajo a la obtención de muestras en huesos, hígado y pulmón.

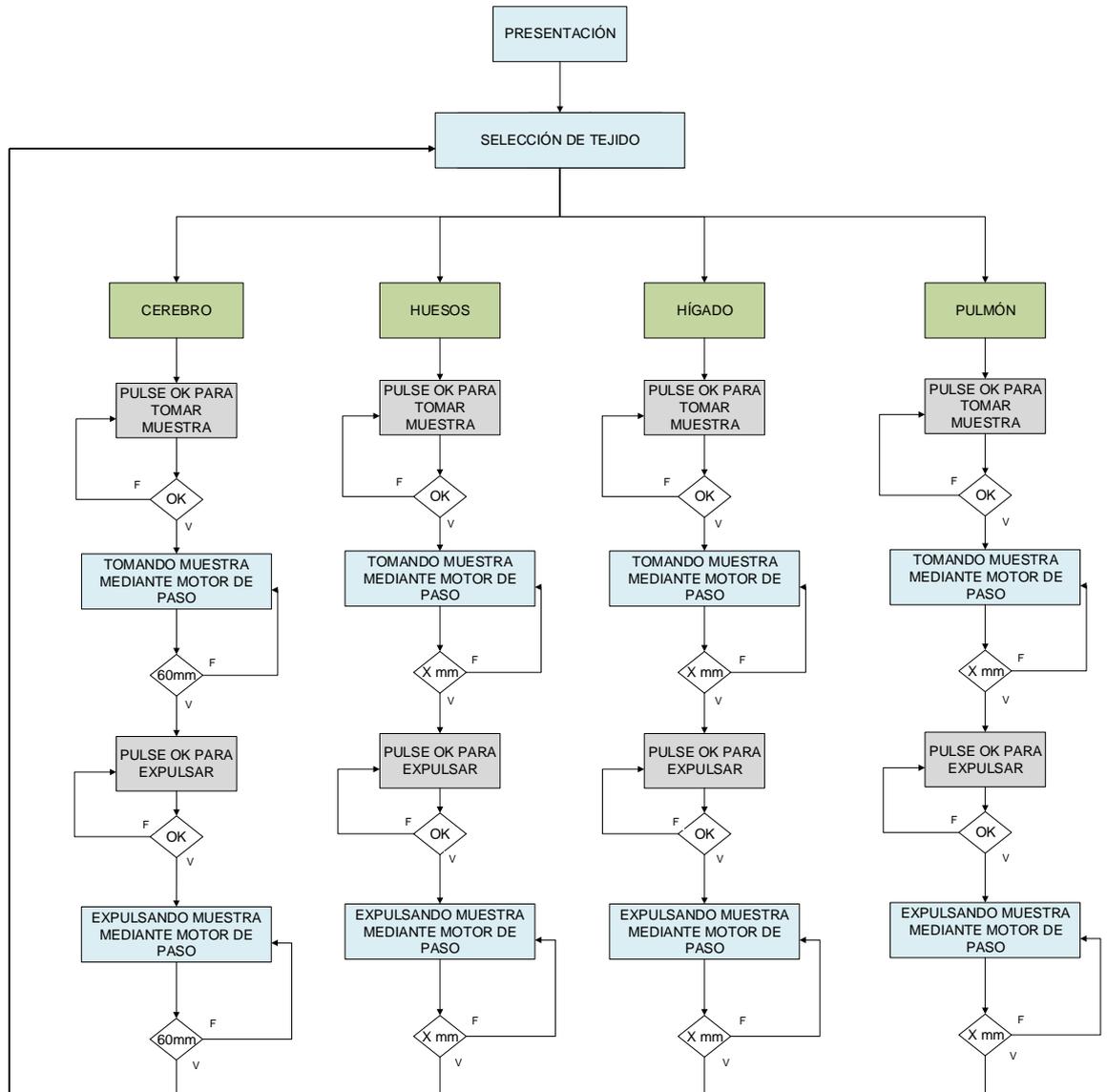


Figura 3. 24: Diagrama ASM del sistema de microaspiración estereotáxica.
Elaborado por: Autor.

En la figura 3.25 se muestra la declaración de librerías en CCS C Compiler, sin esto no se podría realizar la simulación y tampoco en la ejecución del sistema propuesto. La figura 3.26 muestra la declaración de las funciones void (), que es una variable de retorno e int16 (), que es una variable entera de 16 bits con signo. La figura 3.27 muestra las rutinas que permiten el movimiento de izquierda y derecha del embolo.

```

Motor de Paso.c
1  #include <18f4550.h>
2  #fuses xt
3  #use delay (clock=4000000)
4  // #include <stdlib.h>
5  // #byte MOTOR=0xf81 //PORTB con el nombre de MOTOR
6  // Salida de los Servos A3,A4,E0,E1,E2
7  #use fast_io(a)
8  #use fast_io(B)
9  #use fast_io(C)
10 #include <LCD420.c>
11 #include <KBD.c>
12

```

Figura 3. 25: Declaración de librería del sistema propuesto en CCS C Compiler.
Elaborado por: Autor.

```

12
13 void presentacion(void);
14 int16 teclado(int);
15 void cerebro(void);
16 void huesos(void);
17 void higado(void);
18 void pulmon(void);
19 void avanza(int16);
20 void regresa(int16);
21

```

Figura 3. 26: Declaración de funciones del sistema propuesto en CCS C Compiler.
Elaborado por: Autor.

```

22 float avance;
23 byte der[4]={0b1100, 0b0110, 0b0011, 0b1001};
24 byte izq[4]={0b1001, 0b0011, 0b0110, 0b1100};
25

```

Figura 3. 27: Función que ejecuta el avance del embolo.
Elaborado por: Autor.

La figura 3.28 muestra la programación de la presentación inicial y que se visualiza en un LCD (ver figura 3.17). Se puede ver que los comandos utilizados sirven para enviar los datos a visualizarse en la pantalla de cristal líquido. En la figura 3.29 muestra la programación para el inicio de la toma de muestra de la biopsia mediante microaspersión estereotáctica controlada. Consiste en retirar la muestra mediante el uso del motor NEMA17 y en el LCD se mostraría el mensaje de biopsia finalizada, y después expulsa la muestra para su posterior análisis. En las figuras 3.30 a 3.32 se muestra una estructura de programación idéntica al de la figura 3.29 y que puede servir para el desarrollo de futuros trabajos de titulación.

```

25
26 void presentacion(void) {
27     lcd_putc("\fUNIVERSIDAD CATOLICA");
28     lcd_gotoxy(4,2);
29     lcd_putc("DE SANTIAGO DE");
30     lcd_gotoxy(7,3);
31     lcd_putc("GUAYAQUIL");
32     lcd_gotoxy(4,4);
33     lcd_putc("");
34     delay_ms(2000);
35
36     lcd_gotoxy(5,1);
37     lcd_putc("\f    TRABAJO DE");
38     lcd_gotoxy(5,2);
39     lcd_putc("TITULACION");
40     lcd_gotoxy(4,3);
41     lcd_putc("ANDRES PEÑA");
42     lcd_gotoxy(2,4);
43     lcd_putc("BIOPSIA CEREBRAL");
44     delay_ms(2000);
45 }
46

```

Figura 3. 28: Función de presentación inicial.
Elaborado por: Autor.

```

47 void cerebro(void)
48 {
49     int k;
50     lcd_putc("\fINICIO DE BIOPSIA");
51     lcd_gotoxy(1,2);
52     lcd_putc("\fPRESIONE UNA TECLA");
53     lcd_gotoxy(1,3);
54     lcd_putc("PARA TOMAR MUESTRA");
55     k=0;
56     while (k==0)
57     k=kbd_getc();
58     lcd_putc("\fRETIRANDO MUESTRA"); // Retirando Muestra:
59     // lcd_gotoxy(1,2);
60     avanza(170); //aquí se retira el motor de pasos 60 mm
61                 // y poner el retiro en mm en la pantalla
62     lcd_putc("\fBIOPSIA FINALIZADA"); // mensaje de biopsia finalizada
63     lcd_gotoxy(1,2);
64     lcd_putc("\fPRESIONE UNA TECLA");
65     lcd_gotoxy(1,3);
66     lcd_putc("PARA EXPULSAR TEJIDO");
67     k=0;
68     while (k==0)
69     k=kbd_getc();
70     lcd_putc("\fEXPULSANDO TEJIDO"); // Retirando Muestra:
71     // lcd_gotoxy(1,2);
72     regresa(170); //aquí avanza el motor de pasos 60 mm
73                 // y poner el retiro en mm en la pantalla
74     lcd_putc("\fEXPULSION FINALIZADA"); // expulsión finalizada
75     delay_ms(1000);
76 }
77

```

Figura 3. 29: Función para tomar muestra del cerebro.
Elaborado por: Autor.

```

78 void huesos(void)
79 {
80     int k;
81     lcd_putc("\fINICIO DE BIOPSIA");
82     lcd_gotoxy(1,2);
83     lcd_putc("\PRESIONE UNA TECLA");
84     lcd_gotoxy(1,3);
85     lcd_putc("PARA TOMAR MUESTRA");
86     k=0;
87     while (k==0)
88     k=kbd_getc();
89     lcd_putc("\fRETIRANDO MUESTRA"); // Retirando Muestra
90     lcd_gotoxy(1,2);
91     regresa(100); //aquí se retira el motor de pasos 60 mm
92                     // y poner el retiro en mm en la pantalla
93     lcd_putc("\fBIOPSIA FINALIZADA"); // mensaje de biopsia finalizada
94     lcd_gotoxy(1,2);
95     lcd_putc("\PRESIONE UNA TECLA");
96     lcd_gotoxy(1,3);
97     lcd_putc("PARA EXPULSAR TEJIDO");
98     k=0;
99     while (k==0)
100     k=kbd_getc();
101     lcd_putc("\fEXPULSANDO TEJIDO"); // Retirando Muestra
102     lcd_gotoxy(1,2);
103     avanza(100); //aquí avanza el motor de pasos 60 mm
104                     // y poner el retiro en mm en la pantalla
105     lcd_putc("\fEXPULSION FINALIZADA"); // expulsión finalizada
106     delay_ms(1000);
107 }
108

```

Figura 3. 30: Función para tomar muestra del hueso.

Elaborado por: Autor.

```

108
109 void higado(void)
110 {
111     int k;
112     lcd_putc("\fINICIO DE BIOPSIA");
113     lcd_gotoxy(1,2);
114     lcd_putc("\PRESIONE UNA TECLA");
115     lcd_gotoxy(1,3);
116     lcd_putc("PARA TOMAR MUESTRA");
117     k=0;
118     while (k==0)
119     k=kbd_getc();
120     lcd_putc("\fRETIRANDO MUESTRA"); // Retirando Muestra
121     lcd_gotoxy(1,2);
122     regresa(100); //aquí se retira el motor de pasos 60 mm
123                     // y poner el retiro en mm en la pantalla
124     lcd_putc("\fBIOPSIA FINALIZADA"); // mensaje de biopsia finalizada
125     lcd_gotoxy(1,2);
126     lcd_putc("\PRESIONE UNA TECLA");
127     lcd_gotoxy(1,3);
128     lcd_putc("PARA EXPULSAR TEJIDO");
129     k=0;
130     while (k==0)
131     k=kbd_getc();
132     lcd_putc("\fEXPULSANDO TEJIDO"); // Retirando Muestra
133     lcd_gotoxy(1,2);
134     avanza(100); //aquí avanza el motor de pasos 60 mm
135                     // y poner el retiro en mm en la pantalla
136     lcd_putc("\fEXPULSION FINALIZADA"); // expulsión finalizada
137     delay_ms(1000);
138 }
139

```

Figura 3. 31: Función para tomar muestra del hígado.

Elaborado por: Autor.

```

139
140 void pulmon(void)
141 {
142     int k;
143     lcd_putc("\fINICIO DE BIOPSIA");
144     lcd_gotoxy(1,2);
145     lcd_putc("\PRESIONE UNA TECLA");
146     lcd_gotoxy(1,3);
147     lcd_putc("PARA TOMAR MUESTRA");
148     k=0;
149     while (k==0)
150     k=kbd_getc();
151     lcd_putc("\fRETIRANDO MUESTRA"); // Retirando Muestra
152     lcd_gotoxy(1,2);
153     regresa(100); //aquí se retira el motor de pasos 60 mm
154                     // y poner el retiro en mm en la pantalla
155     lcd_putc("\fBIOPSIA FINALIZADA"); // mensaje de biopsia finalizada
156     lcd_gotoxy(1,2);
157     lcd_putc("\PRESIONE UNA TECLA");
158     lcd_gotoxy(1,3);
159     lcd_putc("PARA EXPULSAR TEJIDO");
160     k=0;
161     while (k==0)
162     k=kbd_getc();
163     lcd_putc("\fEXPULSANDO TEJIDO"); // Retirando Muestra
164     lcd_gotoxy(1,2);
165     avanza(100); //aquí avanza el motor de pasos 60 mm
166                     // y poner el retiro en mm en la pantalla
167     lcd_putc("\fEXPULSION FINALIZADA"); // expulsión finalizada
168     delay_ms(1000);
169 }
170

```

Figura 3. 32: Función para tomar muestra del pulmón.
Elaborado por: Autor.

La figura 3.33 muestra la secuencia de avance y retroceso del motor de paso NEMA17, es decir, ejecutar la movilidad a la derecha e izquierda del embolo. La figura 3.34 muestra la programación para el teclado matricial que se encarga de selecciona los modos de operatividad del sistema propuesto.

3.7. Implementación y validación del equipo diseñado.

En esta parte final del capítulo se muestra el diseño ya implementado, así como una carta de validación de este. La figura 3.35 muestra la tarjeta PCB implementada y que es similar al diagrama de bloques visto en la figura 3.23, y este se encarga de manipular al sistema de microaspersión estereotáctica controlada. La figura 3.36 muestra el sistema propuesto y que fue validado (ver figura 3.37) por el Dr. Boris Zurita Cueva, especialista en Neurocirugía del OmniHospital.

```

169     }
170 }
171 void avanza(int16 secuencia){
172     int i,j;
173     avance=0;
174     for (i=0; i<secuencia; i++){
175         for (j=0; j<4; j++){
176             // MOTOR=der[j];
177             avance=avance+0.01;
178             delay_ms(30);
179             lcd_gotoxy(1,3);
180             printf(lcd_putc,"AVANCE= %f",avance);
181             lcd_gotoxy(1,3);
182         }
183     }
184 }
185
186 void regresa(int16 secuencia){
187     int i,j;
188     for (i=0; i<secuencia; i++){
189         for (j=0; j<4; j++){
190             // MOTOR=izq[j];
191             avance=avance-0.01;
192             delay_ms(30);
193             lcd_gotoxy(1,3);
194             printf(lcd_putc,"AVANCE= %f",avance);
195             lcd_gotoxy(1,3);
196         }
197     }
198 }
199 }

```

Figura 3. 33: Función para avance y retroceso del embolo.
Elaborado por: Autor.

```

199
200 int16 teclado(int s){
201     int16 numero=0;
202     int cuent,k;
203     k=0;
204     cuent=0;
205     while(k!=-13)
206     {
207         k=0;
208         while(k==0)
209             k=kbd_getc();
210         k=k-48;
211
212         if(k!=-13){
213             numero=numero*10+k;
214             cuent++;
215             if (cuent==s)
216             {
217
218                 return(numero);
219             }
220         }
221     }
222     return(numero);
223 }
224

```

Figura 3. 34: Función para la operación del teclado matricial.
Elaborado por: Autor.



Figura 3. 35: Implementación del equipo que controla al sistema de microaspersión estereotáctica.
Elaborado por: Autor.

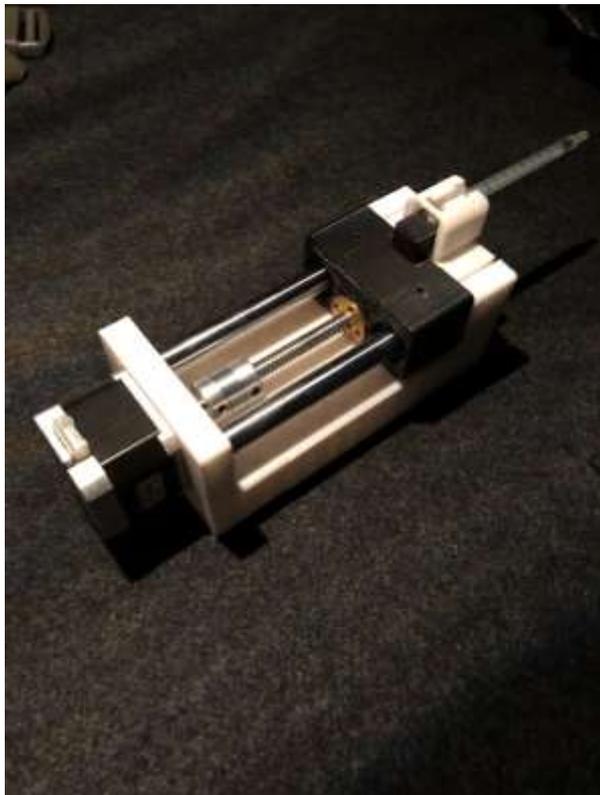


Figura 3. 36: Implementación final del equipo de microaspersión estereotáctica.
Elaborado por: Autor.

Guayaquil, 30 de Enero de 2020.

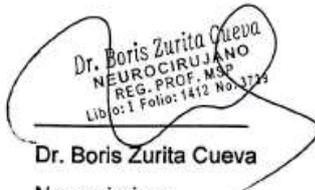
A quien interese,

De mis consideraciones:

Certifico haber aprobado el diseño del sistema desarrollado por el señor Miguel Andres Peña Gaibor con C.I.: 0930421664, validando posteriormente el funcionamiento del equipo en mis procedimientos quirúrgicos a mis pacientes con tumores cerebrales para la obtención de biopsias, así mismo, destaco la importancia del equipo para la calidad de estudio histopatológico.

Es todo lo que puedo decir respecto al caso.

Atentamente,



Dr. Boris Zurita Cueva
NEUROCIRUJANO
REG. PROF. MSP
Libro: I Folio: 1412 No. 3724

Dr. Boris Zurita Cueva
Neurocirujano

Figura 3. 37: Validación del equipo implementado.
Elaborado por: Autor.

Conclusiones.

- La creación de un sistema de microaspersión controlada para biopsias cerebrales estereotáxicas propone el inicio de la incursión del control y automatismo en el área de la medicina específicamente en el ámbito de la neurocirugía del país.
- El sistema desarrollado muestra una eficacia al momento de tomar la muestra de tejido cerebral brindando seguridad al neurocirujano y al paciente durante el procedimiento.
- Las funciones programadas se desempeñaron de acuerdo a lo esperado.

Recomendaciones.

- Realizar estudios más avanzados microscópicos respecto al comportamiento del tejido cerebral utilizando el dispositivo desarrollado y la lesión mecánica que produce este en comparación con la muestra sin el uso del mismo.
- Realizar más estudios con los diferentes tejidos y así mismo el comportamiento del dispositivo frente a los mismos.

Bibliografía

- All About Circuits. (2020). *Resources for Prototyping: The Basics of New Languages for Programming Microcontrollers*. <https://www.allaboutcircuits.com/news/resources-for-prototyping-basics-new-languages-programming-microcontrollers/>
- Barra Zapata, O. E., & Barra Zapata, F. (2011). *Microcontroladores PIC con programación PBP*. RA-MA.
- Basto, C., Pelá, L., & Chacón, R. (2015). *Study on Possibilities for Low-Cost Monitoring of Historical Structures*. <https://doi.org/10.13140/RG.2.1.2190.1284>
- FI. ME. (2018). *Estereotaxia*. <http://www.fime.com.ar/estereotaxia/>
- García Breijo, E. (2008). *Compilador C CCS y simulador PROTEUS para microcontroladores PIC*. Alfaomega.
- García Chávida, H. (2001). *Desarrollo de un mando a distancia reprogramable*. Página personal de Jesús M. Hernández Mangas. <https://www.ele.uva.es/~jesman/>
- Microchip. (2020). *Watchdog Timer Reset (WDT)—Developer Help*. Microchip Developer Help. <https://microchipdeveloper.com/8bit:wdt>
- MicroPython. (2020). *MicroPython—Python for microcontrollers*. <http://micropython.org/>
- Mizobuchi, Y., Nakajima, K., Fujihara, T., Matsuzaki, K., Mure, H., Nagahiro, S., & Takagi, Y. (2019). The risk of hemorrhage in stereotactic biopsy for brain tumors. *The Journal of Medical Investigation: JMI*, 66(3.4), 314–318. <https://doi.org/10.2152/jmi.66.314>
- MSP. (2017). Mayor la tasa de supervivencia de mujeres a los tumores cerebrales no malignos. *Medicina y Salud Pública*.

<https://medicinaysaludpublica.com/mayor-la-tasa-de-supervivencia-de-mujeres-a-los-tumores-cerebrales-no-malignos/>

Navarrete Navarrete, D. J., & Sandoval Vàsquez, C. A. (2011). *Modelo de referencia para el desarrollo de aplicaciones monitoreo, utilizando microcontroladores que administran el ciclo de vida de las aplicaciones* [Trabajo de Titulación, Universidad Técnica del Norte]. <http://repositorio.utn.edu.ec/handle/123456789/626>

Plaza Herrero, J. V. (2014). *Diseño y desarrollo de un sistema POV (Persistence Of Vision) mediante PIC* [Trabajo Final de Carrera, Universidad Politécnica de Valencia]. <https://riunet.upv.es/handle/10251/35887>

Schmidt, B., González-Domínguez, J., Hundt, C., & Schlarb, M. (2018). Modern Architectures. En *Parallel Programming* (pp. 47–75). Elsevier. <https://doi.org/10.1016/B978-0-12-849890-3.00003-4>

Tan, L., & Jiang, J. (2019). Hardware and Software for Digital Signal Processors. En *Digital Signal Processing* (pp. 727–784). Elsevier. <https://doi.org/10.1016/B978-0-12-815071-9.00014-2>

Valverde Villarán, A. (2020). *Sistema de desarrollo para el microcontrolador PIC18F452*. e-REdING. Biblioteca de la Escuela Superior de Ingenieros de Sevilla. <http://bibing.us.es/proyectos/abreproy/11301/direccion/Memoria%252F>

Verle, M. (2012a). *PIC Microcontrollers programming in Basic with examples: Complete guide to the PIC Microcontrollers*. Mikro Electronika.

Verle, M. (2012b). *PIC Microcontrollers—Programming in Assembly*. Mikroe. <http://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-assembly/introduction>



DECLARACIÓN Y AUTORIZACIÓN

Yo, **Peña Gaibor, Miguel Andres** con C.C: # 093042166-4 autor del Trabajo de Titulación: Diseño de un dispositivo de control de microaspiración para biopsias cerebrales estereotáxicas mediante el uso de microcontroladores, previo a la obtención del título de **INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 28 de febrero del 2020

f. _____

Nombre: Peña Gaibor, Miguel Andres

C.C: 093042166-4



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Diseño de un dispositivo de control de micro aspiración para biopsias cerebrales estereotáxicas mediante el uso de microcontroladores.		
AUTOR(ES)	Peña Gaibor, Miguel Andres		
REVISOR(ES)/TUTOR(ES)	M. Sc. Córdova Rivadeneira, Luis Silvio		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería Electrónica en Control y Automatismo		
TITULO OBTENIDO:	Ingeniero Electrónico en Control y Automatismo		
FECHA DE PUBLICACIÓN:	28 de febrero del 2020	No. DE PÁGINAS:	62
ÁREAS TEMÁTICAS:	Sistemas Microcontroladores, Diseño Electrónico		
PALABRAS CLAVES/ KEYWORDS:	Estereotaxia, Tumor, biopsia, Microaspiración, Microcontroladores, Automático		
RESUMEN/ABSTRACT:	<p>Las biopsias cerebrales estereotáxicas es una técnica quirúrgica que permite la extracción de una muestra diminuta de tejido anómalo para analizarlo y poder tener un diagnóstico certero, representan uno de los procedimientos como medio de diagnóstico ante un tumor, una de las complicaciones dentro del procedimiento es el riesgo de sangrado y/o muerte. Una complicación principal asociada con una biopsia cerebral estereotáxica es un hematoma intratumoral, que también es la causa más común de morbilidad relacionada con parálisis permanente y mortalidad en una biopsia cerebral estereotáxica.</p> <p>El riesgo de hemorragia perioperatoria es generalmente de entre 1% y 10%, pero esto podría ser una subestimación, ya que no es común para muchos neurocirujanos para realizar tomografías computarizadas muestras de tumores cerebrales para el diagnóstico después una biopsia cerebral estereotáxica sin complicaciones. El motivo de esta investigación es porque no existe un mecanismo de aspiración controlado o automático, ya que se lo realiza tradicionalmente a "pulso" del neurocirujano. Para esto se ha propuesto desarrollar e implementar un sistema de micro aspiración electrónico controlado usando tecnología como la impresión 3D y el diseño de un circuito electrónico.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593993093175	E-mail: andres122182@hotmail.com	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Mendoza Merchán, Eduardo Vicente		
	Teléfono: +593-9-85086815		
	E-mail: eduardo.mendoza@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			