



Diseño y construcción de un refractómetro de fibra
óptica con alta repetibilidad para control de calidad
de líquidos



Como requisito para la obtención de grado en:
Maestría en Optomecatrónica

Asesor:
Dr. Rodolfo Martínez Manuel

Estudiante:
Ing. María del Rosario Bautista Morales

Diciembre de 2017

Ags., Aguascalientes, México

“Ahora bien, parece extraño, pero las cosas que es bueno tener y los días buenos para disfrutar se cuentan muy pronto y no se les presta demasiada atención; en cambio, las cosas incómodas, estremecedoras, y aun horribles, pueden hacer un buen relato, y además lleva tiempo contarlas.”

J. R. R. Tolkien

Dedicatoria

A mi familia:

*Leopoldo y Guadalupe, mis hermanos,
y sobre todo a Leopoldo y Rita, mis padres.*

Gracias...

Agradecimientos

A mi asesor y amigo el Dr. Rodolfo Martínez Manuel por su orientación, paciencia, por aceptar mi forma de trabajo y confiar en mí.

Al Centro de Investigaciones en Óptica Unidad Aguascalientes por permitirme labo-
rar dentro de sus instalaciones y el uso del equipo necesario para el proyecto.

Al proyecto de Problemas Nacionales: 2015-01-1651, Diseño y Construcción del Po-
tabilizador Integral Solar de Agua para Comunidades Rurales, por el financiamiento
proporcionado para el desarrollo de este trabajo.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca de posgrado
429767 que se me fue otorgada para sustentarme.

Al Dr. Daniel López Cortés por su apoyo en la elaboración de los capilares para el
sensor.

A mis revisores, el Dr. Daniel Alberto May - Arrijoja y el Dr. Carlos Antonio Pine-
da Arellano, por sus valiosas observaciones.

A mi mejor amigo, el Ing. Jonathan Esquivel Hernández, por su apoyo durante el
desarrollo del software del prototipo.

Al Ing. Ricardo Valdivia Hernández por facilitarme las herramientas necesarias para
los diferentes procesos del proyecto.

A mis nuevos y viejos amigos, compañeros y profesores por formar parte de esta gran
etapa de mi vida.

Resumen

En este trabajo se presenta el diseño y construcción del prototipo de un refractómetro de fibra óptica portátil. Por sus características este sistema puede ser empleado para aplicaciones de control de calidad en la industria, o en la investigación como sensor multipunto de índice de refracción.

El sistema óptico para la medición del índice de refracción consta de un arreglo de fibra óptica de tipo interferométrico. Debido a su diseño y montaje, se logró disminuir el ruido y los efectos ambientales en la medición (ver CAP. 3). Con el uso de la técnica de reflectometría en el dominio de frecuencia óptica coherente, un diodo laser y un fotodiodo estándar son empleados como fuente óptica y fotodetector respectivamente permitiendo que el sistema sea de un costo accesible y de dimensiones pequeñas.

El sistema óptico además es multi-punto, tiene más de un sensor por lo que se puede analizar más de un índice de refracción al mismo tiempo. Cada sensor genera una señal resonante a una determinada frecuencia. Al aplicar el algoritmo de transformada rápida de Fourier, las señales se posicionan en diferentes puntos permitiendo separarlas y localizar la componente correspondiente para cada sensor (ver CAP. 3).

El prototipo está construido con láminas de acero, que por sus dimensiones, peso y robustez permite que su traslado sea fácil, (ver CAP. 4. Sección 3). El posicionamiento de las muestras es semi-automático, formado por un mecanismo compuesto de un soporte para muestras accionado por motores DC. Este cuenta con un circuito de control propio (ver CAP. 4, Sección 3).

El procesamiento de la señal se realiza con una tarjeta de desarrollo de tecnología ARM, que nos permite realizar operaciones de punto flotante. Además, el prototipo posee una interfaz de humano-maquina, hecha con una pantalla táctil de tipo capacitiva. La interfaz es amigable e intuitiva para el usuario (ver CAP. 4 Sección 1).

Los resultados obtenidos en el presente trabajo de investigación muestran que el prototipo es funcional, con la capacidad de medir y discriminar muestras líquidas (ver CAP. 5).

Publicaciones

Overlapped fiber optic Michelson interferometers for simultaneous refractive index measurement at two sensing points

Ma. del Rosario Bautista-Morales, Rodolfo Martínez-Manuel*, Carlos A. Pineda-Arellano.

Journal: Applied optics

Received 25 September 2017; revised 26 October 2017; accepted 30 October 2017; posted 30 October 2017 (Doc. ID 307810); published 30 November 2017.

We present a fiber refractometer based on the implementation of overlapped Michelson interferometers; the refractometer allows simultaneous refractive index measurement at two-sensing-points for samples discrimination. The fiber refractometer uses the Fresnel reflection in each fiber tip of the overlapped interferometers to generate the interference signal. Experimental results, implementing the two sensing points, for discrimination between non-contaminated and contaminated distilled water are presented. Despite the simplicity of the presented system, resolution and repeatability of 3×10^{-4} and 5×10^{-4} are obtained in a dip and read experiments using both sensing points simultaneously for refractive index measurement.

Multi-point fiber refractometer implementing coherent optical frequency domain reflectometry

Ma. del Rosario Bautista-Morales, Rodolfo Martínez-Manuel*, Daniel López-Cortés, Carlos A. Pineda-Arellano, Mikhail G. Shlyagin, Óscar Esteban.

Journal: Applied optics

Forthcoming for sending

We demonstrate a multi-point fiber refractometer using the Fresnel reflection at the fiber tip of each sensor. The refractometer is based on the coherent optical frequency domain reflectometry technique; which permits implementing a standard continuous-wave distributed feedback diode laser, tuned over 68 pico-meters, and a standard photodetector for simultaneous sensing signals detection. Signal demultiplexing in the frequency domain allows for multi-point refractive index measurement. Each sensor is made of a Fabry-Perot interferometer formed by a hollow optical fiber mirror and the fiber tip where Fresnel reflection is implemented. Experimental results, using binary liquid mixture of ethylene glycol with distilled water, for multi-point refractive index measurement are presented.

Conferencias

Presentación de poster:

- XIV Encuentro Participación de la Mujer en la Ciencia.

“Sensor interferométrico de fibra óptica tipo Michelson para el monitoreo multi-punto del índice de refracción en muestras de agua contaminadas.”

Ma. del Rosario Bautista-Morales, Rodolfo Martínez-Manuel, Carlos A. Pineda-Arellano.
CIO, 17-19 mayo 2017, León Guanajuato.

Seminario:

- Taller de óptica - CIO Unidad Aguascalientes.

“Refractómetros de fibra óptica para control de calidad de líquidos”

Ma. del Rosario Bautista-Morales, Rodolfo Martínez-Manuel.
CIO, 9-13 octubre, Ags., Aguascalientes.

Taller:

- Congreso Internacional en Sistemas Mecatrónicos.

“Unidad portátil de monitoreo y detección de muestras líquidas contaminadas, como agua o alcohol.”

Ma. del Rosario Bautista-Morales, Rodolfo Martínez-Manuel.
Instituto Tecnológico de Celaya, 18-20 octubre 2017, Ciudad de Celaya, Guanajuato.

Indice

	Página
Dedicatoria	III
Agradecimientos	IV
Resumen	V
Publicaciones	VI
Conferencias	VII
Indice de Figuras	XII
Glosario de acrónimos	XIII
1. Propuesta de trabajo	1
1.1. Contexto	1
1.2. Justificación	2
1.3. Objetivos de la investigación	2
1.4. Especificaciones del diseño	2
1.5. Descripción global del manejo del equipo	5
1.6. Perfil de los usuarios	8
1.7. Descripción del contenido de la tesis	8
2. Marco Teórico	10
2.1. Refractómetros de fibra óptica	10
2.2. Refractómetros interferométricos tipo Michelson	11
2.3. Refractómetros interferométricos de tipo Fabry-Perot	16
2.4. Reflectometría de dominio de frecuencia óptica	19
2.5. Interferencia de onda continua modulada en frecuencia	19
2.6. Índice de refracción como método analítico de control de calidad	21
2.7. Refractómetros comerciales	22
2.8. Conclusión	22
3. Diseño y desarrollo del sistema óptico	23
3.1. Diseño del refractometro multipunto	23
3.2. Simulación del sistema	25

3.3.	Principio de funcionamiento del refractometro multipunto	26
3.4.	Prueba de concepto	29
3.5.	Conclusión	32
4.	Instrumentación del refractómetro portatil	33
4.1.	Tarjeta de adquisición y procesamiento de datos	33
4.1.1.	Tarjeta de adquisicion STM32F767ZI NUCLEO 144	33
4.1.2.	Generación de la señal de modulación	39
4.1.3.	Lectura de datos con el ADC	42
4.1.4.	Transformada de Fourier	44
4.1.5.	Conclusión	49
4.2.	Interfaz de humano - máquina: Programación y funcionamiento	50
4.2.1.	Pantalla GEN4-ULCD-43DCT-CLB	50
4.2.2.	4D SYSTEMS - WORKSHOP 4 IDE	50
4.2.3.	Programación de la pantalla con ViSi - Genie	51
4.2.4.	Compilar y cargar el programa	56
4.2.5.	Comunicación entre la pantalla y la tarjeta de desarrollo: Protocolos de comunicación de ViSi - Genie	59
4.2.6.	Conclusión	68
4.3.	Diseño y construcción del sistema mecánico	69
4.3.1.	Sistema de posicionamiento	69
4.3.2.	Estructura de desplazamiento horizontal	69
4.3.3.	Estructura de desplazamiento vertical	71
4.3.4.	Sistema de control para el posicionamiento de las muestras	73
4.3.5.	Diseño del armazón del prototipo	79
4.3.6.	Conclusión	87
4.4.	Conclusión	87
5.	Integración del sistema y resultados finales	88
5.1.	Circuitos de acoplamiento de señales	88
5.2.	Fuente de alimentación	91
5.3.	Integración del prototipo	92
5.4.	Discusión	94
5.5.	Conclusión	95
5.6.	Trabajo a futuro	96
	Bibliografía	97
6.	Apéndices	101
	Artículo	102
6.1.	Código para la tarjeta STM32F767ZI Núcleo 144	107

Índice de Figuras

1.1.	Vista preliminar del prototipo y distribución de cada uno de sus componentes.	3
1.2.	Relación de los componentes que integran el sistema	4
1.3.	Pantallas de inicio y de medición	5
1.4.	Panel de control e indicadores	6
1.5.	Vista del soporte y colocación de las muestras	6
1.6.	Pantalla de medición mostrando el valor calculado	7
1.7.	Pantalla de medición con los indicadores inicializados o reseteados	7
1.8.	Botón Salir	8
2.1.	Diagrama esquemático de un refractómetro interferométrico tipo Michelson . .	11
2.2.	Espectro de interferencia normalizado simulado del sistema de detección	13
2.3.	Variación simulada del contraste de franjas	14
2.4.	Diagrama esquemático de un refractómetro interferométrico multipunto tipo Michelson [26].	14
2.5.	(a) Espectros de los interferómetros superpuestos, (b) componentes separados de frecuencia de estos interferómetros [26].	15
2.6.	Valor de amplitud de referencia A_{ref}	17
2.7.	a) Imagen del microscopio del sensor de fibra F-P de dos cavidades, (b) espectro de reflexión [28].	18
2.8.	Espectros de Fourier medidos con aire y con varias soluciones de IR	18
2.9.	Principio de funcionamiento de C-OFDR [8].	20
3.1.	Diagrama esquemático del refractómetro multipunto, empleando la técnica C-OFDR	23
3.2.	Cubierta protectora de los sensores F-P.	24
3.3.	Fotografía de la cubierta protectora de los sensores.	24
3.4.	Señal simulada para los interferómetros F-P; (a) espectros de los interferómetros, (b) componentes de frecuencia de cada interferómetro F-P.	26
3.5.	Codificación para la simulación del sistema óptico.	26
3.6.	Fibras HOF	27
3.7.	Espectros de las fibras HOF, utilizadas como espejos dentro de las cavidades interferométricas, la línea verde simboliza la posición y ancho del barrido del diodo láser.	27
3.8.	Espectro de reflexión del arreglo experimental	29
3.9.	Espectro en frecuencia de los interferómetros F-P, con ambos sensores simultáneamente sumergidos en cada una de las muestras.	30

3.10.	Valores de IR calculados usando los dos sensores F-P para medir los diferentes solventes: metanol, acetona, alcohol isopropílico y cloroformo.	30
3.11.	Valores medidos de soluciones de agua destilada con etilenglicol, la línea continua representa el valor promedio de cada ronda de mediciones.	31
3.12.	Relación de la concentración de la solución <i>vs</i> el IR calculado.	31
3.13.	Fotografía de las muestras homogénea utilizadas para realizar la prueba de control de calidad.	32
4.1.	Tarjeta de desarrollo STM32F767ZI, apariencia y distribución de módulos. . .	33
4.2.	Ventana de inicio de STM32CubeMX.	34
4.3.	Generación de un nuevo proyecto.	35
4.4.	Entorno de STM32CubeMX.	35
4.5.	Generación del código de inicialización en C.	36
4.6.	Project Explorer de AtollicTrueStudio	37
4.7.	Entorno de programación de AtollicTrueStudio.	38
4.8.	Entorno de la ventana de Debugger de AtollicTrueStudio.	38
4.9.	Activación de la terminal PA4 de la tarjeta	39
4.10.	Activación del Timer 2	39
4.11.	Sección de la configuración de reloj dentro de la tarjeta	40
4.12.	Sección de código para la inicialización y activación del DAC dentro de la tarjeta.	40
4.13.	Sección de código para la inicialización y activación del TIM2.	41
4.14.	Señal de modulación generada por la tarjeta (amarillo), comparada con la señal del generador de funciones (verde).	42
4.15.	Señal de interferencia obtenida por el sistema óptico, con la señal de modulación generada con la tarjeta (azul). Transformada de Fourier de la señal (rosa).	42
4.16.	Activación del ADC en la tarjeta con STM32CubeMX.	43
4.17.	Parámetros del ADC.	43
4.18.	Detección del inicio y el final de la señal generada con el DAC.	44
4.19.	Función de adquisición de la señal por el ADC.	44
4.20.	<i>Paths</i> dentro de las propiedades del proyecto donde se colocan todas las direcciones desde donde el programa buscara las librerías que necesiten para funcionar [45].	45
4.21.	Archivos complementarios necesarios para el cálculo de la FFT.	46
4.22.	Ejemplo del código para implementar la FFT.	47
4.23.	Algoritmo de selección de funciones implementado que rige el comportamiento descrito en el diagrama de flujo anterior.	49
4.24.	Dimensiones y aspecto de la pantalla GEN4-ULCD-43DCT-CLB implementada [47].	50
4.25.	Entorno de Workshop4IDE	51
4.26.	Pasos para la creación de un nuevo proyecto.	52
4.27.	Entorno de programación de Visi-Genie.	53
4.28.	Aspecto y objetos que componen la pantalla de inicio del prototipo.	54
4.29.	Pantalla de medición.	55
4.30.	Cable de programación de la compañía 4D System	56
4.31.	Conexión del módulo de programación de la pantalla con el cable 4D.	56
4.32.	Diferentes estados posibles entre la conexión de la pantalla con la PC.	57

4.33. Tarjeta de memoria micro - SD	57
4.34. Botón de compilar y cargar programa.	58
4.35. Consola del entorno	58
4.36. Colocación de la tarjeta micro - SD	58
4.37. Pantalla en funcionamiento con el programa desarrollado.	59
4.38. Estructura del mensaje para la comunicación serial.	59
4.39. Aspecto de la ventana de la herramienta GTX.	60
4.40. Comandos para diferentes objetos disponibles para la programación de la pantalla [51].	62
4.41. Ventana de la terminal a 9600 baudios.	62
4.42. Habilitación del USART6 y ubicación de las terminales de conexión.	65
4.43. Conexión entre la tarjeta de desarrollo y la pantalla.	65
4.44. Parámetros establecidos en la opción de configuración del STM32CubeMx para la comunicación serial.	66
4.45. Vista lateral del sistema de posicionamiento	69
4.46. Estructura de desplazamiento horizontal	70
4.47. Representación y nombre de las terminales del motor a pasos.	70
4.48. a) Sentido de giro en el motor. b) Sistema cremallera-piñón.	71
4.49. Estructura de desplazamiento vertical.	72
4.50. a) Estructuras de posicionamiento unidas con la descripción de los movimientos realizados por cada una. b) Sistema de posicionamiento armado.	73
4.51. a) Componentes del sistema porta-muestras. b) Desplazamientos del soporte.	73
4.52. a) DIP estándar de 16 pines del L293D. b) Conexión del integrado.	74
4.53. DIP estándar de 18 pines del PIC16F84A con distribución de terminales.	75
4.54. Asas del prototipo para su traslado.	79
4.55. Elementos para el ajuste de nivel del prototipo.	79
4.56. Estructura interna del prototipo para soporte de elementos.	86
4.57. Ensamble del prototipo.	86
5.1. Circuito de acondicionamiento de la señal de modulación. OPAM Restador No Inversor	89
5.2. Circuito de acondicionamiento de la señal de interferencia. OPAM Amplificador no inversor y filtraje tipo pasa bajos (recuadro rojo).	89
5.3. Configuración de pines para los integrados OP07 y OP177. [52], [53]	90
5.4. Circuitos de acondicionamiento armados y soldados dentro del prototipo.	90
5.5. Señal de modulación desplazada con el offset requerido por el sistema	91
5.6. Fuente de alimentación señalando el calibre de cables utilizado para cada sección del prototipo.	92
5.7. Integración del prototipo, vista preliminar de componentes internos.	93
5.8. Pantalla de medición en funcionamiento. Muestra el valor de IR de los sensores en aire.	94
5.9. Prototipo final ensamblado.	94

Glosario de acrónimos

Acrónimo

ARM	<i>(Advanced RISC Machine)</i>
BBS	Fuente de espectro amplio <i>(Broadband source)</i>
CMRR	Factor de rechazo al modo común <i>(Common Mode Rejection Ratio)</i>
C-OFDR	Reflectometría de dominio de frecuencia óptica coherente (<i>Coherent- optical frequency domain reflectometry</i>)
DAC	Convertidor digital-analógico <i>(Digital-To-Analog converter)</i>
DFB	Láser de retroalimentación distribuida <i>(Distributed feedback laser)</i>
DSP	Procesamiento de señales digitales <i>(Digital signal processing)</i>
DUT	Dispositivo bajo prueba (<i>Device under test</i>)
FBG	Rejillas de Bragg <i>(Fiber Bragg grating)</i>
FFT	Transformada rápida de Fourier <i>(Fast Fourier Transform)</i>
F-P	Interferómetro Fabry-Perot <i>(Fabry-Perot interferometer)</i>
FPU	Unidad de punto flotante <i>(Floating point unit)</i>
GUI	Interfaz gráfica de usuario <i>(Graphical user interface)</i>
HOF	Fibra óptica hueca <i>(Hollow optical fiber)</i>
IR	Índice de refracción <i>(Refractive index)</i>
LCD	Pantalla de cristal líquido <i>(Liquid Cristal Display)</i>
MMF	Fibra multimodo <i>(Multimode fiber)</i>
MPU	Unidad de procesamiento MIDI <i>(MIDI Processing Unit)</i>
OPAM	Amplificador operacional <i>(Operational Amplifier)</i>
OSA	Analizador de espectros ópticos <i>(Optical analyzer spectrum)</i>
PIC	Controlador de interfaz periférica <i>(Peripheral Interface Controller)</i>
PWM	Modulación por ancho de pulso <i>(Pulse-width modulation)</i>
RISC	Set de instrucciones reducido para computadoras <i>(Reduced Instruction Set Computer)</i> .
SMF	Fibra monomodo <i>(Single mode fiber)</i>
TLS	Fuente láser sintonizable <i>(Tunable laser source)</i>
TFT	Transistor de película delgada <i>(Thin-film transistor)</i> .

CAPÍTULO 1

Propuesta de trabajo

1.1. Contexto

El índice de refracción (IR) es una de las propiedades fundamentales fisicoquímicas de los materiales. Medir el IR permite identificar una sustancia particular, confirmar su pureza o medir su concentración; esto conlleva a que su estudio y medición sean útiles en diversos campos (tanto industriales como académicos) [1].

El equipo que mide el IR es conocido como refractómetro. Existen diversas configuraciones comerciales de refractómetros en las que destacan: refractómetros de mano tradicionales y digitales, refractómetros de Abbe, refractómetros de proceso en línea, entre otros [2]. Sin embargo, la principal desventaja de estos es que utilizan sistemas ópticos de compleja fabricación, como prismas. Además, debido a su susceptibilidad a la temperatura suelen incluir sistemas de control de temperatura que incrementan su costo.

En los últimos años, los refractómetros de fibra óptica han sido intensamente investigados y presentan grandes ventajas [3]. Algunos de estos refractómetros son, aquellos que se basan en la interacción del campo evanescente [4], de estructuras de resonancia tales como rejillas de período largo [5], rejillas de Bragg [6], y estructuras de resonancia de plasmón superficial [7]. A pesar de que todos estos refractómetros muestran buen rendimiento presentan ciertas desventajas pues requieren de equipos precisos y costosos para la interrogación del IR. Además estos refractómetros han sido desarrollados para medir una sola muestra a la vez.

En la presente tesis presentamos un sistema de refractómetro de fibra óptica de tipo interferométrico. La punta de la fibra se usa como sensor y no requiere ser modificada. Utiliza fibra óptica estándar (SMF28). Al variar la longitud de los interferómetros tipo Fabry-Perot que componen el sistema; el refractómetro puede ser multi-punto, es decir se pueden evaluar muestras en diferentes puntos de detección.

Por otra parte, los costos de fabricación del sistema propuesto disminuyen al utilizar un diodo láser como fuente de luz, y un solo fotodetector estándar para la señal de todos los sensores. Esto es posible gracias a la implementación de la técnica de reflectometría de dominio de frecuencia óptica coherente (C-OFDR, Coherence Optical Frequency Domain Reflectometry) [8].

El refractómetro construido tiene aplicaciones de control de calidad en la industria.

1.2. Justificación

El resultado esperado de este trabajo de investigación es la construcción de un prototipo funcional de un refractómetro de fibra óptica, cuyas mediciones sean repetibles, precisas y exactas, para que pueda ser implementado como un sistema de control de calidad.

Los tres aspectos tecnológicos más importantes en este proyecto son el desarrollo de sensores de fibras para la medición multipunto del IR, el desarrollo de un algoritmo capaz de desenvolverse dentro de una tarjeta de desarrollo que logre calcular la transformada de Fourier y procesar datos de tipo flotante. Y, finalmente lograr que el prototipo sea portátil.

Se exponen el diseño, el desarrollo y construcción del sistema, y se explican las limitantes y posibles mejoras que pueden realizarse. Este proyecto busca tener aplicaciones en la industria y ser una herramienta de medición en el campo por lo que se busca la robustez del prototipo y se definen las condiciones de las muestras óptimas para emplear el sistema; así como el tipo de clientes a los que se les ofrece este equipo.

1.3. Objetivos de la investigación

Objetivo general:

- Desarrollar el prototipo de un refractómetro de fibra óptica, para ser implementado como control de calidad de líquidos.

Objetivos Especificos:

- Diseñar e implementar sistemas ópticos que permitan mediciones repetibles y precisas.
- Desarrollar algoritmos digitales para el procesamiento de señales y obtención del índice de refracción.
- Diseñar y construir los elementos mecánicos del prototipo, que sea robusto capaz de ser transportable.
- Realizar pruebas en diferentes muestras con diferentes condiciones ambientales, para determinar la confiabilidad en las mediciones.
- Generar una interfaz humano-máquina de fácil utilización para el prototipo.
- Analizar los resultados obtenidos en la implementación y viabilidad de la utilización del dispositivo.

1.4. Especificaciones del diseño

La Figura 1.1 describe de manera general los bloques que componen el prototipo, en cuanto distribución, dimensiones y posición dentro de la carcasa del mismo, (en los siguientes capítulos se describen cada una de las secciones a detalle).

El sistema óptico de fibra es de tipo interferométrico y posee 2 sensores de detección para la medición del IR. Cada sensor es un Interferómetro Fabry-Perot. Una fibra óptica de hueco (HOF, Hollow Optical Fiber) de 0,5 mm de longitud empalmada en ambos lados con una fibra monomodo estándar (SMF, Single-Mode Optical Fiber) es el primer espejo del interferómetro y el segundo espejo se encuentra en la punta de fibra. En esta punta se genera la

reflexión de Fresnel, la cual cambia de acuerdo con el valor de IR medido. El sistema de interrogación se basa en la técnica de reflectometría de dominio de frecuencia óptica coherente (C-OFDR, Coherence Optical Frequency Domain Refelctometry). Esta técnica requiere de un láser sintonizable, por lo que se implementó como fuente óptica un diodo láser de realimentación distribuida (DFB, Distributed Feedback Laser) sintonizado, y un fotodetector estándar para detección de la señal de los sensores.

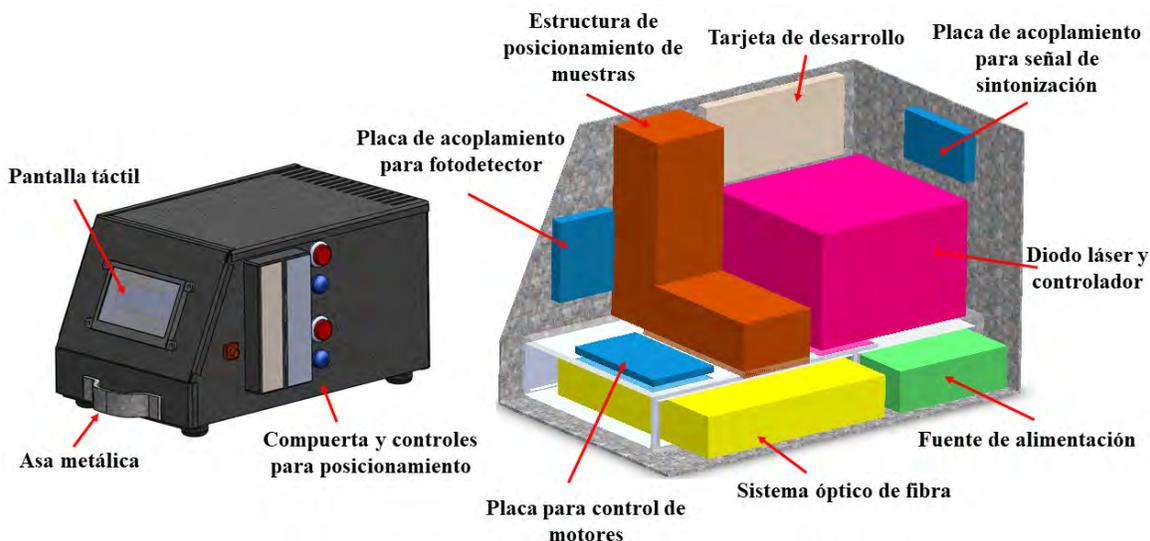


Figura 1.1: Vista preliminar del prototipo y distribución de cada uno de sus componentes.

La señal generada por cada uno de los sensores llega simultáneamente al fotodetector. Una vez adquirida la señal se le aplica la transformada de Fourier, con ello, ésta pasa del dominio del tiempo al dominio de la frecuencia y permite separar y discriminar entre sensores individuales.

Debido a que cada sensor posee una componente en frecuencia determinada por la resonancia del interferómetro que lo compone, el modelo matemático que rige el cálculo del IR se basa principalmente en la razón de amplitudes de los picos de frecuencia de cada interferómetro (ver Cap. III). Los resultados experimentales del sistema se comparan con los valores de la literatura reportados de un equipo calibrado que trabaja a la longitud de 1550nm [9], obteniendo un error promedio de 1×10^{-4} .

La Figura 1.2 muestra la relación de los componentes que conforman el funcionamiento general del prototipo. Para realizar el procesamiento de la señal proveniente de los sensores, se utilizó la tarjeta de desarrollo STM32F767ZI Núcleo 144, con procesador Diablo16 de la tecnología ARM, programada en el entorno de STM32CubeMx y AtollicTrueSTUDIO. La tarjeta además se encarga de generar la señal para modular la corriente del láser, y así obtener la sintonización en longitud de onda.

La interfaz de usuario es una pantalla táctil capacitiva de la familia de 4D System con matrícula: GEN4-ULCD-4DCT-CLB. La programación de la pantalla se realizó en WorkShop4IDE. Se desarrollo el algoritmo de comunicación entre la pantalla y la tarjeta de desarrollo con el

protocolo de comunicación UART a 9600 baudios.

El diseño de la carcasa del prototipo se realizó en el entorno de SolidWorks considerando los requisitos de espacio y aislamiento. Se manufacturó en láminas de acero siendo montado y desmontado con facilidad. Las medidas finales del prototipo fueron 31cmx19cmx19cm. Cuenta con un sistema semi-automático de posicionamiento de las muestras que ejecuta dos movimientos básicos de trayectoria horizontal y vertical, empleando un motor a pasos bipolar y un motor DC a +6V que se controlan con un PIC16F84A programado en MikroC, se emplean puentes H para permitir el giro del motor en ambos sentidos.

Cabe señalar que con la finalidad de tener una estructura modular (ver Figura 1.2), y operar apropiadamente el prototipo, cada sección se realizó por separado para posteriormente integrar todo de forma óptima.

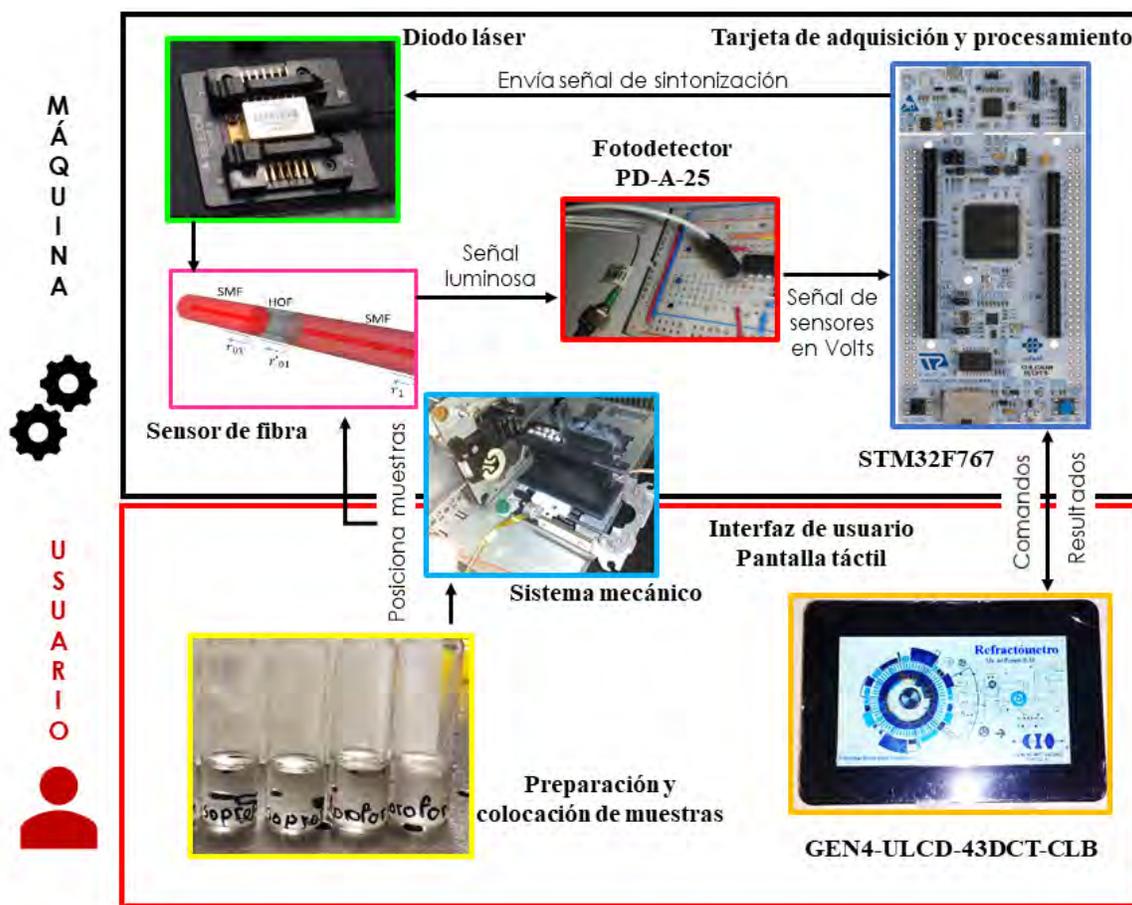


Figura 1.2: Relación de los componentes que integran el sistema, muestra las diferencias entre el trabajo automatizado de la máquina (refractómetro), y del usuario.

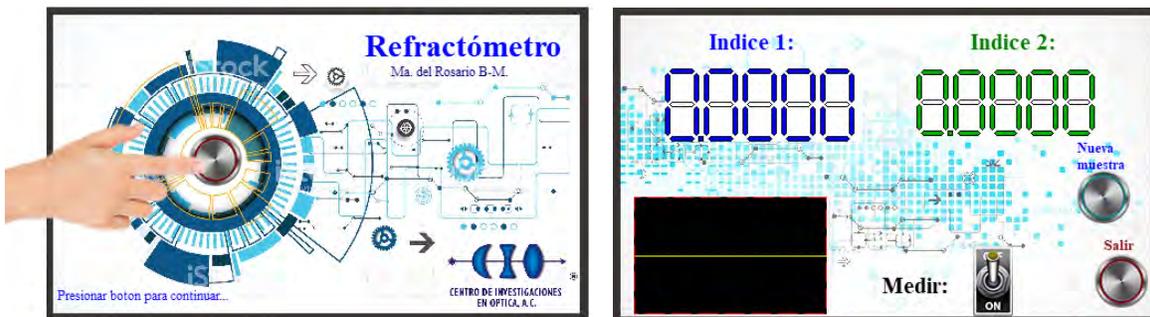


Figura 1.3: a) Pantalla de inicio del sistema, b) Pantalla de medición con valores inicializados.

1.5. Descripción global del manejo del equipo

A continuación, se describe el proceso para la medición del IR con el prototipo. Como ya se ha mencionado anteriormente, el equipo posee una pantalla de interfaz humano - máquina, con la que el usuario interactúa y ejecuta funciones.

La Figura 1.3(a) muestra la pantalla de inicio, para comenzar, se presiona el botón central. La pantalla cambiará como se muestra en la Figura 1.3(b), esto tiene la finalidad de que el algoritmo procese y obtenga los parámetros de referencia necesarios para proceder con la medición de las muestras. El algoritmo usado para el cálculo de IR se describe a detalle en el Cap. III.

Una alarma sonará cuando el equipo este listo para proceder con la colocación de las muestras. La Figura 1.4 muestra la compuerta y controles de posicionamiento, dos botones, “**Posicionar**” y “**Retirar**” con un LED indicador cada uno. El protocolo básico para operar el prototipo y poder relizar mediciones de IR en las muestras es el siguiente:

1. Se presiona “**Retirar**” para que el soporte que sostiene las muestras salga.
2. El LED indicador 2 se mantiene encendido cuando el soporte se encuentra fuera, (Figura 1.4(c)). En este momento las muestras deben ser colocadas en el soporte por parte del usuario (Figura 1.5).
3. Al colocar las muestras se debe considerar:
 - La muestra no sea viscosa.
 - La muestra sea líquida y transparente.
 - Llenar las celdas aproximadamente hasta 3/4 del volumen.
4. Una vez colocadas las muestras adecuadamente, presionar el botón “**Posicionar**”. El soporte con las muestras se desplazará hasta la posición donde los sensores del sistema queden introducidos dentro de la celda y hagan contacto adecuado con la muestra. El LED indicador 1 encenderá cuando estas se encuentren correctamente posicionadas (Figura 1.4(b)).

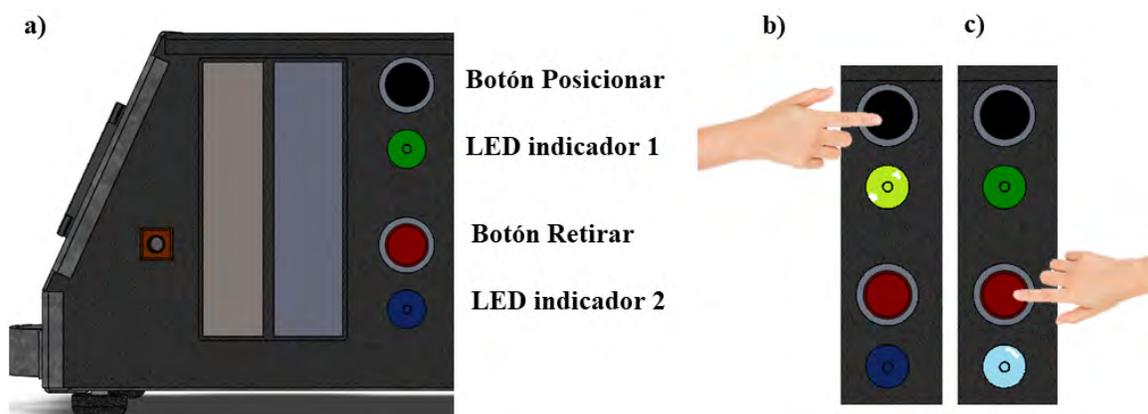


Figura 1.4: a) Panel de control para colocar las muestras en el prototipo. b) Botón "**Posicionar**" presionado **LED indicador 1 Encendido.** c) Botón "**Retirar**" presionado, **LED indicador 2 encendido.**

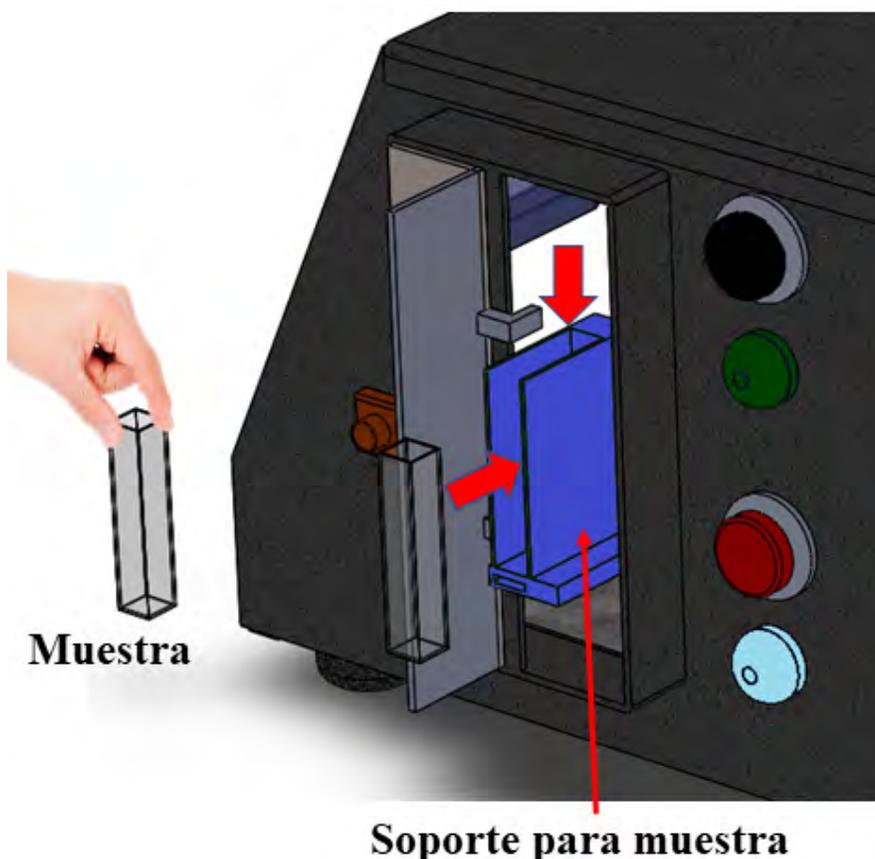


Figura 1.5: Vista del soporte y colocación de las muestras.

5. Presionar el botón de "**Medir**" en la pantalla táctil. En la pantalla, los displays de **Índice 1** e **Índice 2** se mostrarán los valores obtenidos del IR en los sensores 1 y 2 respectivamente (Figura 1.6).

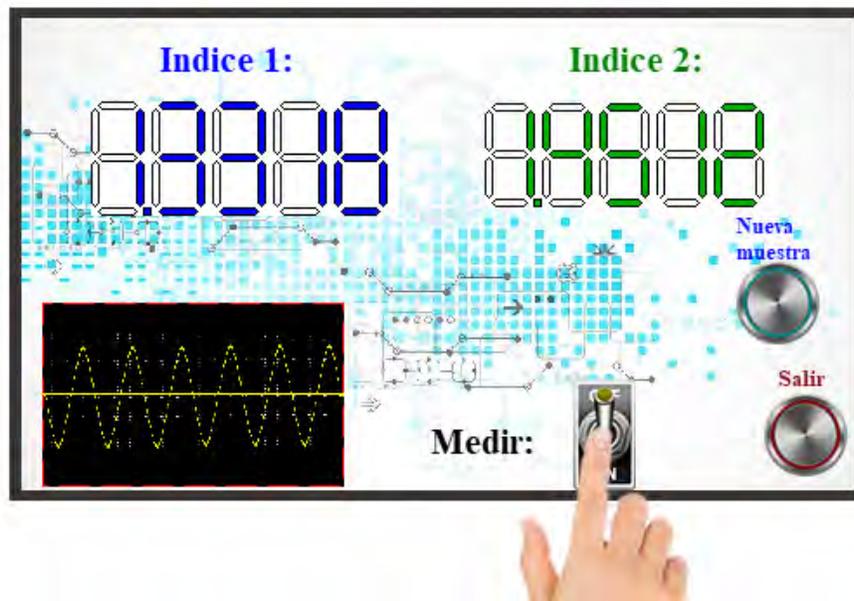


Figura 1.6: Pantalla de medición mostrando el valor calculado.

6. Para una nueva muestra se repite el procedimiento antes descrito para la colocación de muestras. Entre cada prueba las puntas sensoras se limpian empleando las soluciones adecuadas y se dejan dentro aproximadamente 3 seg. Se colocan las nuevas muestras siguiendo los mismos pasos.

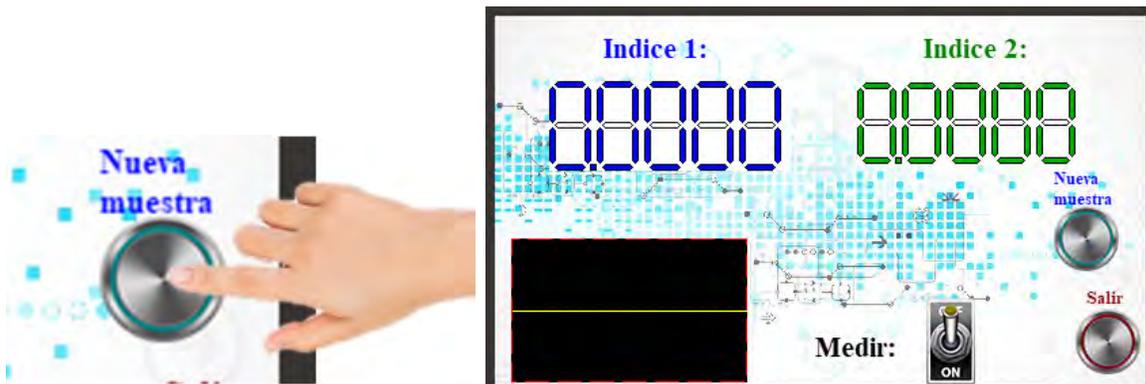


Figura 1.7: Pantalla de medición con los indicadores inicializados o reseteados.

7. Se presiona el botón “**Nueva muestra**”, para inicializar los valores de cálculo necesarios para la obtención del nuevo valor del IR (Figura 1.7). Se presiona el botón “**Medir**” para mostrar en los displays los nuevos valores.
8. Para finalizar se limpian por última vez los sensores y se presiona “**Salir**” (Figura 1.8). Se inicializan los valores de referencia y se finaliza el proceso.



Figura 1.8: Botón salir, resetea el sistema y regresa a la pantalla de inicio.

1.6. Perfil de los usuarios

Los clientes potenciales de este producto abarcan varios campos de aplicación, tales como:

- Control de calidad en productos químicos.
- Detección de bebidas adulteradas.
- Comparación de índices entre productos líquidos comerciales.
- Detección de medicamentos falsos.
- Investigación como sensor multipunto de medición de IR.

Se recomienda que los operadores de este equipo cumplan con los siguientes requerimientos:

- Conocimiento básico del área.
- Experiencia técnica.
- Nivel educacional básico.
- Seguir las indicaciones para el manejo óptimo del prototipo.
- Respetar las características y limitación en los tipos de muestra a medir.

El sistema es de fácil manejo por la robustez del mismo, sin embargo, es importante seguir y atender las instrucciones de manejo, así como interpretar adecuadamente el tipo de muestras permisibles para detectar.

1.7. Descripción del contenido de la tesis

CAPITULO 1: PROPUESTA DE TRABAJO

En este capítulo se describe de manera general el prototipo a desarrollar, los objetivos alcanzados y la justificación del estudio.

CAPITULO 2: MARCO TEORICO

En este capítulo se introducen los conceptos necesarios para comprender el funcionamiento del sistema, se muestran ejemplos de tipos de refractómetros y la forma en que operan.

Se describe la técnica de sintonización empleada en el sistema óptico y se explica el modo de interrogación del sensor. Finalmente se muestra el estado del arte y se mencionan algunas de las aplicaciones para el sensor.

CAPITULO 3: DISEÑO Y DESARROLLO DEL SISTEMA OPTICO

En este capítulo se describe a detalle el diseño del refractómetro implementado en el prototipo, así como el modelo matemático que lo rige. Los experimentos de la prueba de concepto con los resultados del sistema y caracterización del sensor. Finalmente se muestra la simulación con los valores calculados.

CAPITULO 4: DISEÑO Y CONSTRUCCION DEL SISTEMA MECATRONICO

En este capítulo se describe a detalle cada uno de los módulos que operan dentro del prototipo, la tarjeta, la pantalla, y los sistemas mecatrónicos, se explican los algoritmos de programación y se muestran los programas.

Finalmente se muestran los planos de la carcasa del dispositivo y la manufacturación del mismo.

CAPITULO 5: INTEGRACION DEL SISTEMA Y RESULTADOS FINALES

Se muestra el sistema armado con pruebas de su implementación en diferentes productos, se discuten los resultados y se mencionan las limitaciones y alcances.

Una vez descrito grosso modo el sistema desarrollado se procede a explicar a detalle en los siguientes capítulos el diseño y construcción del refractómetro.

CAPÍTULO 2

Marco Teórico

En este capítulo se introducen los conceptos necesarios para entender el principio de funcionamiento de un refractómetro de fibra óptica de tipo interferométrico.

Iniciaremos explicando las ventajas que poseen los sensores de fibra, posteriormente se mostrarán ejemplos de diferentes tipos de refractómetros y la forma en que operan. Al final se describirá la técnica empleada en el prototipo para la medición del índice de refracción (IR) y el método de interrogación del sensor.

Se mencionan algunas de las pruebas de calidad utilizadas en la industria y se describen refractómetros comerciales de uso vigente.

2.1. Refractómetros de fibra óptica

Los sensores de fibra óptica ofrecen muchas ventajas, incluida la inmunidad a la interferencia electromagnética, la capacidad de multiplexación de varios sensores, tamaño pequeño, capacidad de detección en tiempo real y resistencia a entornos hostiles, como altas temperaturas y grandes presiones [10]. Debido a este tipo de ventajas los sensores de fibra han recibido mucha atención, ya que además pueden ofrecer la capacidad de medir simultáneamente parámetros múltiples incluyendo tensión, presión, temperatura y el que concierne a este trabajo de investigación, el IR [11].

La medición del IR en líquidos es importante para aplicaciones médicas, químicas, industriales y biológicas [12]. En los últimos años se han desarrollado diferentes tipos de refractómetros en una gran variedad de configuraciones.

Entre las configuraciones más conocidas tenemos estructuras periódicas como rejillas de Bragg (FBG, Fiber Bragg Gratings) [13] o rejillas de período largo (LPG, Long Period Gratings) [14]. Sin embargo, estos sensores tienen un problema de sensibilidad cruzada [15], y también requieren costosos equipos de fabricación, que restringen sus aplicaciones prácticas.

El refractómetro Abbe es otro método clásico para medir el IR basado en detecciones de transmisión cerca de ángulos críticos relacionados con la reflexión total interna [16]. La desventaja que presentan es un alto grado de susceptibilidad a la temperatura [17].

Otro tipo de sensores de IR se basan en nanocapas metálicas. Existen diversos tipos de mecanismos para esta clase de sensores, que se clasifican dependiendo de las propiedades de la película delgada que lo conforma. Los de resonancia de plasmón superficial (SPR, Surface Plasmon Resonance) [18] y las resonancias de modos con pérdidas (LMR, Lossy modes resonances) [19]. Este tipo de sensores son muy sensibles al IR por lo que son muy atractivos para diversas aplicaciones. Empero, normalmente funcionan en modo de transmisión y presentan resonancias espectrales muy amplias que hacen que la multiplexación sea una tarea muy difícil [20].

Otra técnica empleada para medir el IR es utilizar la reflexión de Fresnel [21] en la punta de la fibra. Cuando ésta se sumerge en la muestra de interés, el IR de la muestra puede calcularse a partir del coeficiente de reflexión medido utilizando la fórmula de Fresnel para la incidencia normal. Sin embargo, la posible inestabilidad de la fuente de luz aumenta la incertidumbre para determinar el coeficiente de reflexión. Para lograr una mejor precisión con este método, se debe implementar una referencia dinámica de la potencia de la luz en la punta de la fibra. Comúnmente se utilizan dos fibras diferentes: una fibra sirve como sensor para la medición de IR y la otra se usa para referencia de intensidad [22].

Una buena alternativa para medir IR, son los refractómetros interferométricos debido a su alta sensibilidad, configuración simple, y a que la sensibilidad cruzada es pequeña.

2.2. Refractómetros interferométricos tipo Michelson

A continuación, se describen ejemplos de refractómetros de fibra óptica de tipo interferométrico.

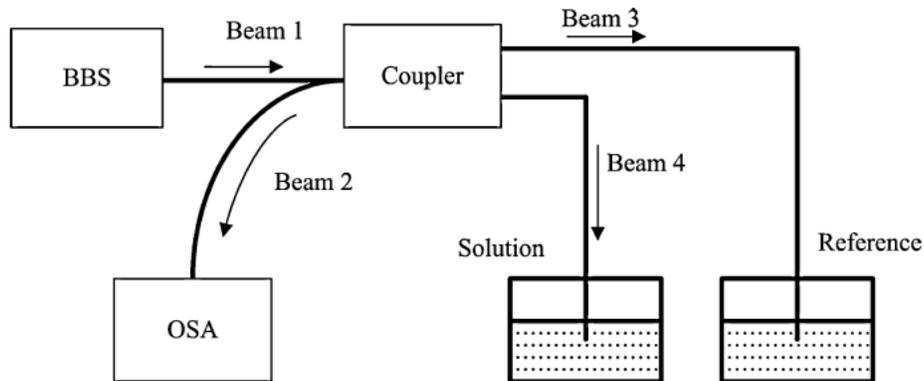


Figura 2.1: Diagrama esquemático de un refractómetro interferométrico tipo Michelson que emplea reflexión de Fresnel en la punta de la fibra para la medición de IR [23].

En este ejemplo se presenta un refractómetro interferométrico tipo Michelson [23] que utiliza la visibilidad del espectro de interferencia generado para calcular el valor de IR.

La Fig. 2.1 muestra el diagrama esquemático del refractómetro. Éste tiene una fuente óptica de emisión espontánea amplificada (ASE, Amplified spontaneous emission) que actúa como una fuente óptica de banda ancha (BBS, Broadband source), un acoplador de fibra, un analizador de espectro óptico (OSA, Optical spectrum analyzer) y dos fibras ópticas cuyos extremos actúan como sensores. La luz enviada desde el BBS se divide en dos haces (haz 3 y haz 4) mediante el acoplador y se refleja en la interfaz fibra-líquido. Los dos haces reflejados interfieren en el acoplador y son detectadas por el OSA.

La reflexión de Fresnel obtenida de la interfaz fibra-líquido cambiará de acuerdo con el IR de la muestra. La intensidad reflejada del haz 3 y del haz 4 pueden obtenerse mediante la ecuación de Fresnel:

$$I_{3R} = \frac{1}{2} \left(\frac{n_f - n_{x0}}{n_f + n_{x0}} \right)^2 I_0(\lambda) \quad (2.1)$$

$$I_{4R} = \frac{1}{2} \left(\frac{n_f - n_x}{n_f + n_x} \right)^2 I_0(\lambda) \quad (2.2)$$

donde $I_0(\lambda)$ es la intensidad de salida del BBS, n_f es el IR efectivo del núcleo de la fibra, n_{x0} y n_x son los IR del disolvente y la solución, respectivamente. Mediante la teoría de interferencia de dos haces, se obtiene la intensidad del haz 2:

$$I_2 = \frac{1}{2} I_{3R}(\lambda) + \frac{1}{2} I_{4R}(\lambda) + 2 \sqrt{\frac{1}{2} I_{3R}(\lambda) * \frac{1}{2} I_{4R}(\lambda) \cos(\Delta\phi)}, \quad (2.3)$$

donde $\Delta\phi$ es la diferencia de fase entre el haz 3 y 4 dado por:

$$\Delta\phi = \frac{2\pi}{\lambda} 2 n_f \Delta l, \quad (2.4)$$

con Δl como la diferencia de camino entre los haces. En este ejemplo para mostrar cómo varía el patrón de interferencia con el IR de la muestra, se calcularon los espectros de reflexión del sensor para soluciones de azúcar a diferentes concentraciones (Fig. 2.2).

La variación del contraste de franja contra el IR de la solución, n_x , se muestra en la Fig. 2.3. Con los resultados de la Fig. 2.3 se concluye que la resolución de este sistema aumenta con la disminución de la concentración de la solución. Además, puede proporcionar una medición de prácticamente cualquier IR excepto la del núcleo de la fibra, en el que el contraste de franja es igual a cero.

En este ejemplo se emplea la visibilidad de la señal del espectro de interferencia para calcular el IR. Sin embargo, el desempeño en la detección de los cambios en visibilidad depende del OSA que se utilice, del tal forma que existe una relación directamente proporcional entre la resolución del sistema y el costo. Además, esta clase de sensor tiene problemas debido a los cambios de fase en la señal [24].

Para evitar lidiar con los problemas de trabajar directamente con el espectro de la señal, otro método de análisis es trabajar en el dominio de la frecuencia, lo que permite además

demultiplexar la señal para tener un sistema multipunto (tener más de un punto de detección). A continuación, se describe este tipo de análisis [25].

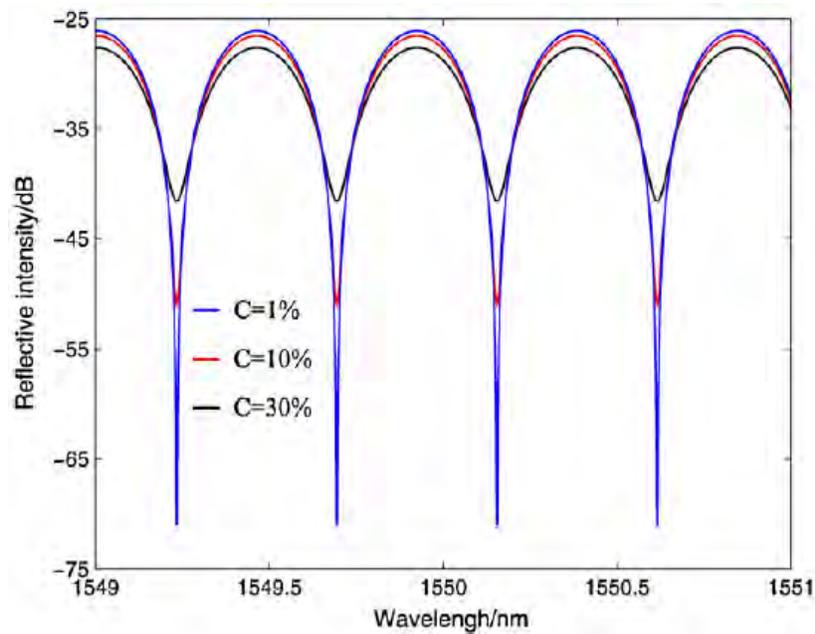


Figura 2.2: Espectro de interferencia normalizado simulado del sistema de detección inmerso en soluciones de azúcar con diferentes concentraciones de 1%, 10% y 30%. El contraste de franjas aumenta a medida que disminuye la concentración. Los parámetros utilizados fueron $l = 1.8 \text{ mm}$, $n_f = 1.44961$ y $n_{x0} = 1.3119$ (agua destilada). [23]

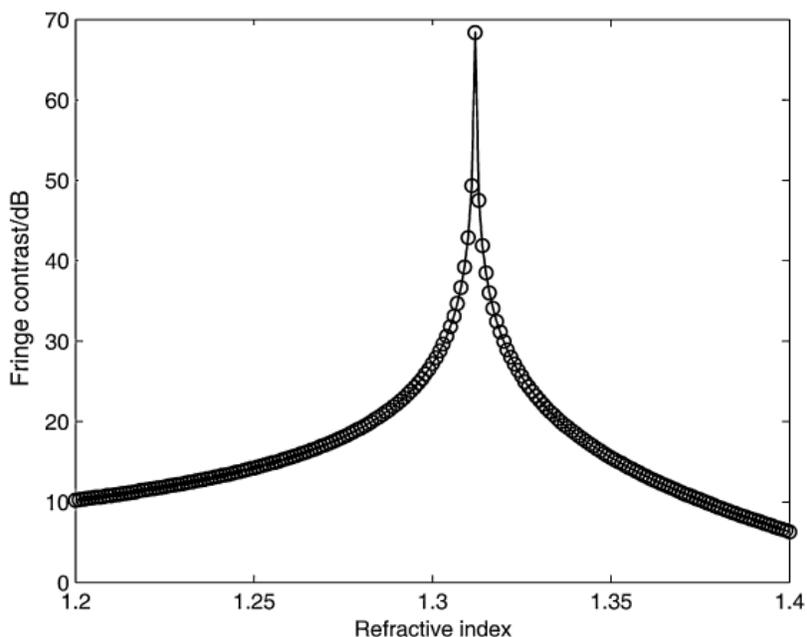


Figura 2.3: Variación simulada del contraste de franjas del sistema al variar el IR de la solución. El contraste de franjas tiene un máximo cuando el IR detectado es igual al IR del solvente. [23]

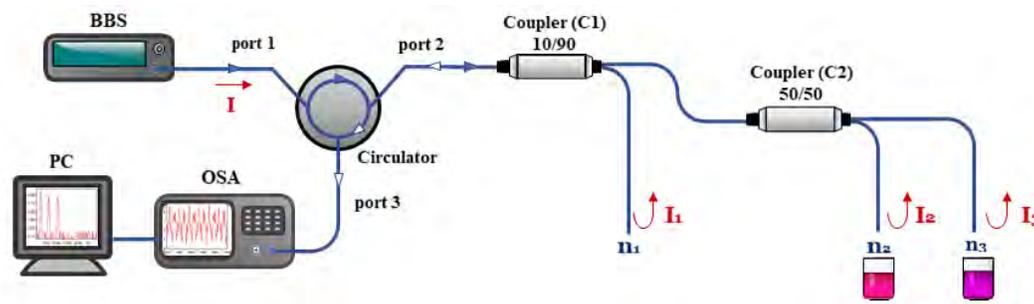


Figura 2.4: Diagrama esquemático de un refractómetro interferométrico multipunto tipo Michelson [26].

En la configuración que se muestra esquemáticamente en la Fig. 2.4 se tiene un diodo superluminiscente que se usa como BBS, la luz de la fuente óptica pasa a través del puerto uno de un circulator. El puerto dos del circulator se conecta a un acoplador estándar 10/90 (C1). El brazo del acoplador del 10 por ciento, actúa como una fibra de referencia (f1). Y el brazo de 90 por ciento del acoplador está conectado a un segundo acoplador 50/50 (C2) para dividir el haz lo que permite obtener dos puntos de detección en cada extremo de fibra (f2 y f3).

La luz reflejada desde estas tres puntas de fibra (f1, f2 y f3) genera la señal de interferencia en C1, que sale por el puerto 3 del circulator y es detectada por un OSA. La intensidad de la luz se refleja es una proporción dependiendo del coeficiente de reflexión de Fresnel, correspondiente al cambio de IR experimentado en cada punta de la fibra. Se puede representar como:

$$I_1 = \frac{1}{10} I \left(\frac{n_f - n_1}{n_f + n_1} \right)^2, \quad (2.5)$$

$$I_2 = \frac{9}{20} I \left(\frac{n_f - n_2}{n_f + n_2} \right)^2, \quad (2.6)$$

$$I_3 = \frac{9}{20} I \left(\frac{n_f - n_3}{n_f + n_3} \right)^2, \quad (2.7)$$

donde I es la intensidad de luz proporcionada por la fuente óptica; I_1 , I_2 e I_3 son la intensidad de los haces reflejados en las puntas de las fibras, f_1 , f_2 y f_3 , respectivamente; n_1 , n_2 y n_3 son los IR que rodean las puntas.

Los haces reflejados desde las puntas de las fibras generan interferómetros superpuestos. La señal de interferencia generada en estos interferómetros se puede escribir de la siguiente manera:

$$I_{12} = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\Delta\phi_{12}), \quad (2.8)$$

$$I_{13} = I_1 + I_3 + 2\sqrt{I_1 I_3} \cos(\Delta\phi_{13}), \quad (2.9)$$

$$I_{23} = I_2 + I_3 + 2\sqrt{I_2 I_3} \cos(\Delta\phi_{23}), \quad (2.10)$$

donde $\Delta\phi_{12} = \frac{2\pi}{\lambda} 2n_f l_{12}$, $\Delta\phi_{13} = \frac{2\pi}{\lambda} 2n_f l_{13}$ y $\Delta\phi_{23} = \frac{2\pi}{\lambda} 2n_f l_{23}$ son la diferencia de fase de cada interferómetro, y l_{12} , l_{13} y l_{23} son la diferencia de longitud entre cada uno. Las señales de estos interferómetros llegan al OSA. Se obtiene la transformada de Fourier para pasarlas al dominio de frecuencia y así poder separar cada señal.

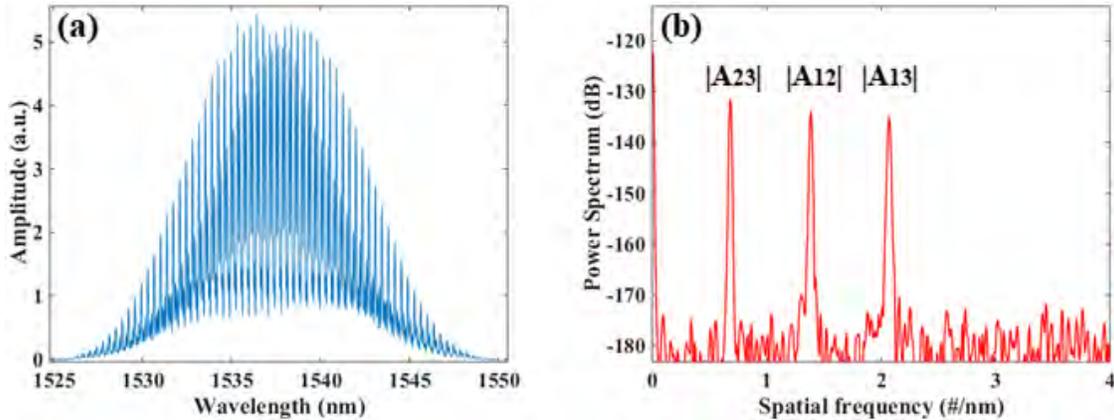


Figura 2.5: (a) Espectros de los interferómetros superpuestos, (b) componentes separados de frecuencia de estos interferómetros [26].

Los componentes que se observan en la Fig. 2.5, $|A_{12}|$ y $|A_{13}|$, se usan para el cálculo del IR. Las amplitudes de los componentes en frecuencia se definen a partir de la ecuación:

$$|A_{12}| = 2\sqrt{\frac{1}{10} I \left(\frac{n_f - n_1}{n_f + n_1} \right)^2 * I \left(\frac{n_f - n_2}{n_f + n_2} \right)^2}, \quad (2.11)$$

simplificando para cada interferómetro se tiene:

$$|A_{12}| = \frac{6}{5\sqrt{8}} \left(\frac{n_f - n_1}{n_f + n_1} \right) \left(\frac{n_f - n_2}{n_f + n_2} \right) I \quad (2.12)$$

$$|A_{13}| = \frac{6}{5\sqrt{8}} \left(\frac{n_f - n_1}{n_f + n_1} \right) \left(\frac{n_f - n_3}{n_f + n_3} \right) I \quad (2.13)$$

Para calcular el valor de IR de cada muestra, primero se toma una medición de referencia con las tres puntas en un medio de IR bien conocido n_0 (A_{ref12} y A_{ref13}).

Posteriormente, las puntas de fibra de detección (f2 y f3) se sumergen en la muestra a medir. Los espectros obtenidos se transforman en el dominio de frecuencia para identificar los componentes de frecuencia de cada interferómetro. Se toma una relación entre las amplitudes de la medición de referencia y la medición de la muestra y se calcula el valor de IR, esto se puede expresar de la siguiente forma:

$$\frac{|A_{ref12}|}{|A_{12}|} = \frac{\frac{6}{5\sqrt{8}} \left(\frac{n_f - n_0}{n_f + n_0} \right)^2 I}{\frac{6}{5\sqrt{8}} \left(\frac{n_f - n_0}{n_f + n_0} \right) \left(\frac{n_f - n_2}{n_f + n_2} \right) I} = \frac{\frac{n_f - n_0}{n_f + n_0}}{\frac{n_f - n_2}{n_f + n_2}}, \quad (2.14)$$

despejando n_2 de 2.14 se tiene,

$$n_2 = n_f \left[\frac{\frac{|A_{ref12}|}{|A_{12}|} (n_f + n_0) - n_f + n_0}{\frac{|A_{ref12}|}{|A_{12}|} (n_f + n_0) + n_f - n_0} \right] \quad (2.15)$$

Lo mismo ocurre con la punta de medición f3. Además, este procedimiento permite mediciones independientes de la intensidad y también se elimina la relación de acoplamiento de los acopladores [27]. La Fig. 2.6(a) ejemplifica el valor de amplitud de la señal cuando el valor de IR externo en la fibra cambia. Los resultados obtenidos al utilizar este método se muestran en la Fig. 2.6(b).

En los ejemplos descritos, a pesar de tener un buen rendimiento con valores calculados de IR adecuados, requieren de equipos costos para funcionar, tal es el caso de OSA. Además, este tipo de refractómetros son inadecuados si se busca disminuir en tamaño o emplearlos en aplicaciones prácticas en la industria.

2.3. Refractómetros interferométricos de tipo Fabry-Perot

Un interferómetro Fabry-Perot (F-P) está compuesto por dos superficies reflectoras paralelas entre sí y separadas por una distancia determinada [28]. La interferencia ocurre debido a las superposiciones múltiples de los haces reflejados y transmitidos en las dos superficies reflejantes.

El espectro del interferómetro F-P puede describirse como la modulación de intensidad dependiente de la longitud de onda del espectro de la luz de entrada. Este está determinado por la

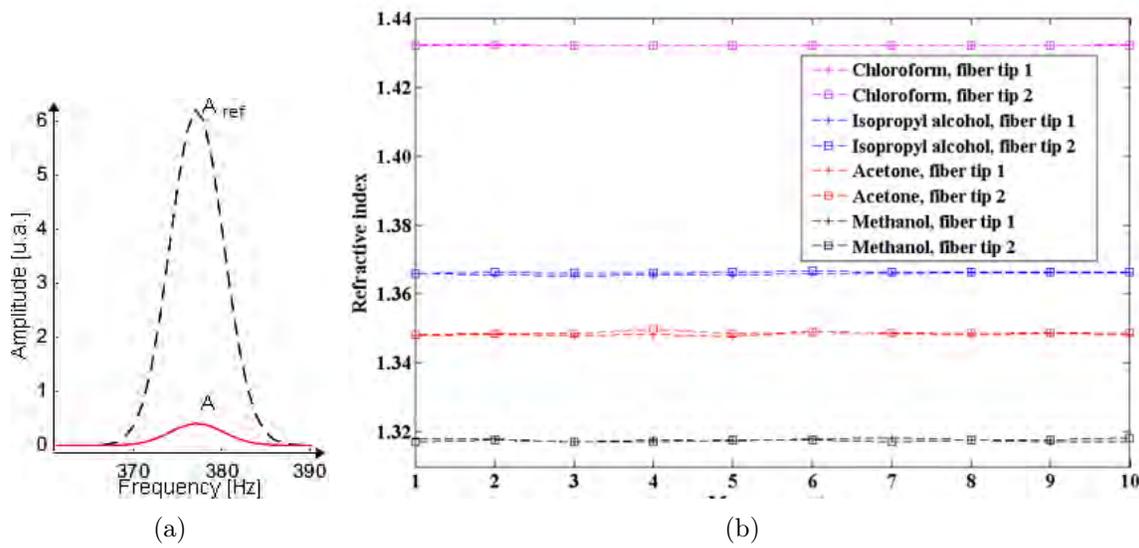


Figura 2.6: Valor de amplitud de referencia (A_{ref} del componente en frecuencia de un interferómetro cuando la punta de la fibra esta en un IR conocido (línea punteada), amplitud (A) del mismo pico cuando se sumerge en la muestra a ser medida (rojo). b) Resultados obtenidos con este arreglo experimental [26].

diferencia de fase óptica entre los dos haces reflejados. La diferencia de fase del F-P se expresa como:

$$\delta_{F-P} = \frac{2\pi}{\lambda} n 2L, \quad (2.16)$$

donde λ es la longitud de onda de la luz incidente, n es el IR del material de la cavidad, y L es la longitud física de la cavidad. Cuando ocurre una perturbación en el sensor, la diferencia de fase se ve influida por la variación en la diferencia de camino óptico (OPD, Optical path difference).

Existe una gran variedad de sensores F-P de fibra óptica [29]. Un claro de ejemplo es el que se muestra en la Fig. 2.7(a), el cual está formado por dos cavidades. Las cavidades se forman al fusionar un fragmento de HOF [30] entre una SMF y una fibra multimodo (MMF, Multimode fiber). El espectro de reflexión de este sensor F-P, se muestra en la Fig. 2.7(b). El patrón de interferencia es bastante complejo debido a la superposición de dos cavidades. Sin embargo, al igual que con el interferómetro de Michelson, al aplicar la FFT se observa como hay tres picos dominantes correspondientes a la interferencia producida entre las superficies reflectoras (Fig. 2.8(a)). Cuando el extremo del MMF se sumerge en una solución líquida, las intensidades del segundo y del tercer pico se reducen, pero el primer pico no se ve afectado (Fig. 2.8(a)) [28]. Esto se debe a que la reflexión de Fresnel solo afecta en la superficie del extremo de la MMF. El IR de la solución líquida calculado a partir de la variación de intensidad del pico de Fourier se representa gráficamente en la Fig. 2.8(b).

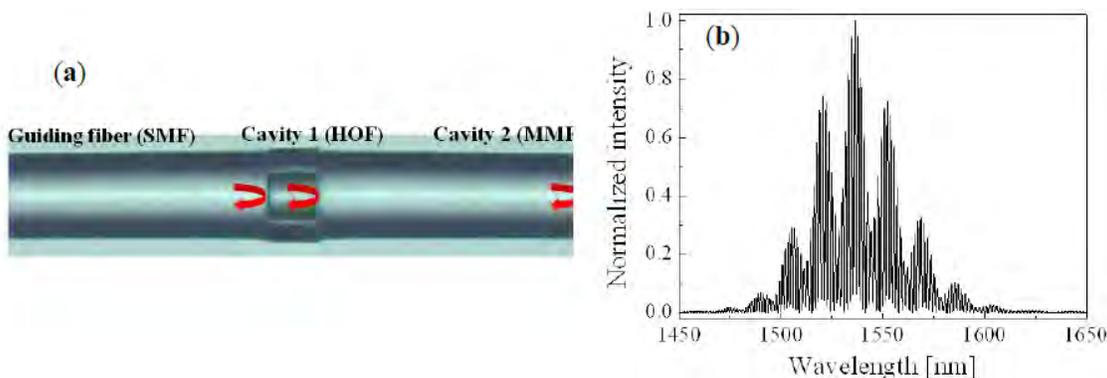


Figura 2.7: a) Imagen del microscopio del sensor de fibra F-P de dos cavidades, (b) espectro de reflexión [28].

Una ventaja que tiene la implementación de cavidades HOF intrínseca es que no se ve afectada por la variación de los entornos químicos o físicos externos [31]. Por lo tanto, los errores de medición del sensor basados intensidad, como la fluctuación de la fuente de potencia o la perturbación externa, se pueden compensar utilizando el primer pico de la señal.

Las cavidades intrínsecas han sido implementadas por muchos tipos de dispositivos especializados de fibra óptica, como fibras ópticas microestructuradas (MOF, Micro-structured optical fibers) o FBG [32].

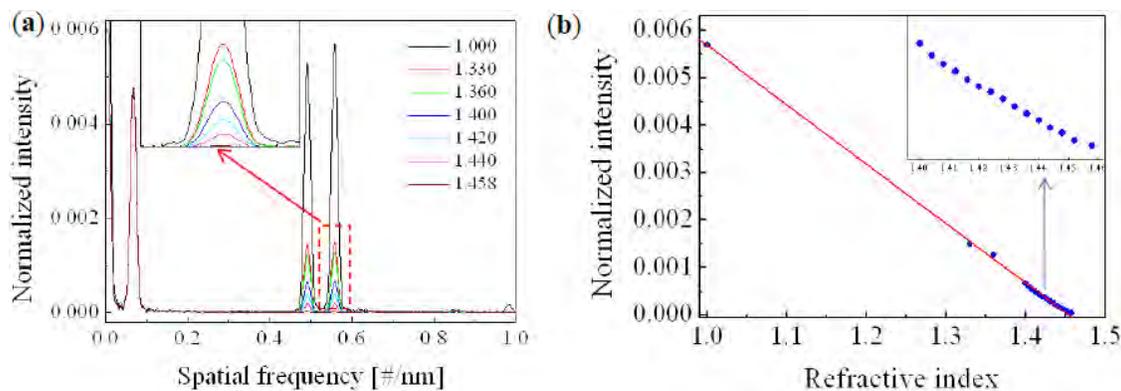


Figura 2.8: a) Espectros de Fourier medidos con aire (curva negra) y con varias soluciones de IR (curvas de color). El recuadro es la imagen ampliada de la región punteada roja; b) Variación de intensidad del tercer pico con respecto al IR de las soluciones [32].

En los ejemplos descritos, a pesar de tener un buen rendimiento, requieren de equipos costosos para funcionar, como el OSA. Además, este tipo de refractómetros son inadecuados si se busca disminuir en tamaño o emplearlos en aplicaciones prácticas en la industria. A continuación, se describe la técnica empleada en el prototipo para la interrogación del sensor. Esto es ya que permite implementar otro tipo de fuentes óptica y detectores.

2.4. Reflectometría de dominio de frecuencia óptica

La reflectometría de dominio de frecuencia óptica (OFDR, Optical frequency-domain reflectometry) tiene varias aplicaciones. Se utiliza típicamente como una herramienta de diagnóstico y caracterización para fibras ópticas, y también se emplea para la realización de sensores distribuidos de temperatura y deformación [33].

El fenómeno básico detrás de la OFDR coherente es la interferencia de onda continua modulada en frecuencia (FMCW, Frequency-modulated continuouswave interference) [34].

En la configuración básica, el OFDR coherente (C-OFDR) consiste en una fuente de láser sintonizable (TLS, Tunable laser source) que se barre linealmente. Luego, la señal óptica modulada en frecuencia se divide en dos haces, uno de los cuales pasa por el DUT (Device under test), mientras que el otro se usa como señal de referencia (Fig. 2.9). La señal de referencia que regresa desde el espejo de referencia y la señal del DUT interfieren en el acoplador. Esta señal de interferencia contiene las frecuencias de batimiento que aparecen como picos después de aplicar la transformada de Fourier.

El principio de funcionamiento de C-OFDR en la Fig.2.9 posee varias ventajas tales como: el ancho de banda del receptor es menor en comparación con las técnicas OTDR (Optical Time Domain Reflectometer) [35] que reducen el nivel de ruido y aumentan el rango dinámico. Además, los sistemas C-OFDR tienen la capacidad de medir dispositivos activos sin saturación ya que solo se utilizan señales de baja potencia [36]. La distancia de medición está limitada por la longitud de coherencia de la fuente de luz sintonizable.

2.5. Interferencia de onda continua modulada en frecuencia

La interferencia óptica tipo FMCW puede ofrecer una mayor precisión y un rango de medición más largo que los interferómetros ópticos tradicionales. Comparativamente, la interferencia homodina clásica difícilmente podría realizar una medición de largo alcance debido a la ambigüedad del cambio de fase. Además, debido a que la frecuencia de la señal de batimiento está relacionada con el OPD, la interferencia óptica de FMCW también puede realizar mediciones absolutas de OPD y de velocidad.

Este tipo de interferencia consta de señales ópticas que se modulan continuamente. Si dos haces ópticos, que se derivan de la misma fuente óptica coherente cuya frecuencia angular está modulada con una onda de diente de sierra pero han viajado por caminos diferentes, se recombinan en un punto en el espacio. La intensidad $I(\tau, t)$ del campo eléctrico resultante se puede escribir como:

$$I(\tau, t) = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\alpha\tau\omega_0\tau) = I_0 + V \cos(\omega_b t + \Phi_{b0}), \quad (2.17)$$

donde I_1 e I_2 son las intensidades de los haces de referencia y medición respectivamente α es la tasa de modulación de frecuencia angular óptica, τ es el tiempo de retardo de la onda de la señal de medición con respecto a la señal de referencia, t es el tiempo, ω_0 es la frecuencia angular óptica en el centro del período de modulación, I_0 es la intensidad media de la señal, V

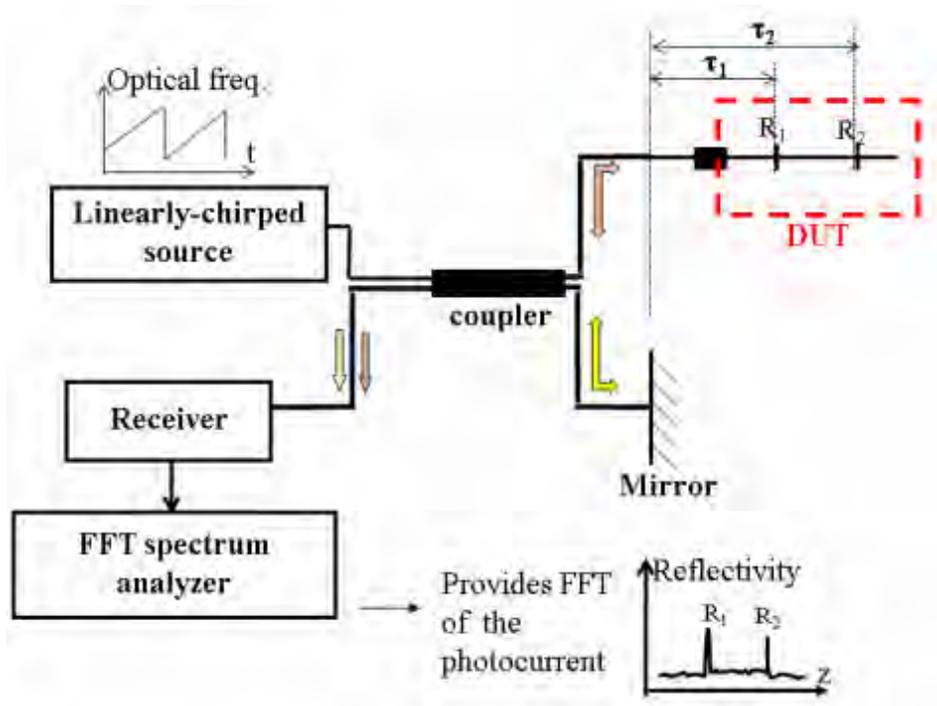


Figura 2.9: Principio de funcionamiento de C-OFDR [8].

es el contraste de la señal de batimiento $V = \frac{\sqrt{I_1 I_2}}{I_1 + I_2}$, y ω_b y Φ_{b0} son la frecuencia angular y la fase inicial, respectivamente, de la señal de batimientos. La tasa de modulación de frecuencia angular óptica esta dada por:

$$\alpha = \Delta\omega/T, \quad (2.18)$$

donde $\Delta\omega$ es la excursión de modulación de frecuencia angular óptica y T es el período de la señal de modulación. ω_b y Φ_{b0} , se definen como $\omega_b = \alpha\tau$, y, $\Phi_{b0} = \omega_0\tau$.

Por lo tanto, en términos de frecuencia, longitud de onda (λ) y la diferencia de camino óptico (OPD, optical Path Diference), la Eq. 2.17 de la intensidad puede ser reescrita como:

$$\begin{aligned} I(OPD, t) &= I_0 \left[1 + V \cos \left(\frac{2\pi\Delta\nu\nu_m}{c} OPD t + \frac{2\pi}{\lambda_0} OPD \right) \right] \\ &= I_0 [1 + V \cos(2\pi\nu_b t + \Phi_{b0})] \end{aligned} \quad (2.19)$$

donde $\Delta\nu$ es la excursión de modulación en frecuencia óptica, ν_m es la frecuencia de la señal de modulación, c es la velocidad de la luz en el vacío, λ_0 es la longitud de onda central, ν_b y Φ_{b0} son la frecuencia y la fase inicial de la señal de batimiento, respectivamente.

Tanto un interferómetro FMCW tipo Michelson o un interferómetro FMCW tipo Fabry-Perot son adecuados para medir el desplazamiento, la distancia y la velocidad de un objeto, así como para medir el índice de refracción óptica de un gas o líquido, siempre y cuando las muestras

se coloquen de tal forma que modifiquen el OPD del interferómetro [34].

Hasta ahora, se cree que los láseres semiconductores son las mejores fuentes ópticas para los interferómetros ópticos FMCW. Los láseres de semiconductores tienen las ventajas de la modulación de corriente continua, respuesta lineal aceptable, gran excursión de modulación (hasta 100 GHz), alta velocidad de modulación (hasta 10 MHz o en el rango de GHz en el caso de semiconductores), tamaño compacto, peso ligero y costo relativamente bajo [37]. Sin embargo, los láseres semiconductores se ven fácilmente afectados por la luz de retroalimentación de los interferómetros y por los cambios en la temperatura circundante. Por lo tanto, los interferómetros FMCW ópticos generalmente usan un aislador óptico para evitar la luz de retroalimentación y usan un sistema de control de temperatura o un sistema de compensación de deriva de fase para minimizar los efectos de la deriva de frecuencia del láser.

En el Cap. III se describirá el sistema óptico propuesto, así como el modelo matemático que lo rige.

2.6. Índice de refracción como método analítico de control de calidad

En la actualidad, un refractómetro óptico Abbe o un refractómetro digital moderno son parte del equipamiento estándar de prácticamente todos los laboratorios. La razón es porque la determinación del IR suele ser el método más simple y rápido para caracterizar muestras líquidas. En comparación con otras técnicas de caracterización demoran considerablemente más tiempo que la determinación del IR [36].

Incluso la introducción de medidores modernos de densidad digital apenas ha cambiado la situación: los refractómetros digitales de alta resolución son usualmente más económicos y fáciles de operar que los medidores digitales de densidad.

Las determinaciones de concentración usando índice de refracción con instrumentos digitales de alta resolución generalmente son muy precisas. Por ejemplo, la determinación del contenido de agua en una solución medida con el IR tiene las siguientes ventajas:

1. La determinación del contenido de agua lleva menos de dos minutos.
2. No se requieren productos químicos, a diferencia de otros métodos para la determinación del contenido de agua (por ejemplo, valoraciones de Karl Fischer) [38].
3. El método de índice de refracción es, por lo tanto, más ecológico y económico que la mayoría de los otros métodos.
4. La determinación no es destructiva. La misma muestra también se puede usar para otros análisis.

Un ejemplo de la aplicación de IR en el control de calidad industrial es que en cierta clase de productos líquidos, lo que interesa es comprobar que el producto no contiene otras sustancias.

Un posible método para el control de calidad de los solventes es una determinación del contenido de agua de Karl Fischer y un análisis de la misma muestra con cromatografía de gases [39].

Por razones económicas, sin embargo, un método tan lento puede difícilmente justificarse para el control de los productos en los disolventes orgánicos. Si se sabe el IR de un solvente con una resolución aceptable, si se miden más muestras y los resultados están dentro de un rango de aceptabilidad del producto, entonces se consideran confiables.

En muchas industrias, el control de calidad de muestras líquidas con medición de IR ya es un método estándar. Los fabricantes de perfumes y fragancias, por ejemplo, especifican estas dos cantidades en sus fichas técnicas. El usuario final solo necesita determinar el índice de refracción de los productos para garantizar que se mantenga la calidad.

La disponibilidad de instrumentos digitales de alta resolución ha dado lugar a que la determinación del índice de refracción se utilice cada vez más frecuentemente para el control de calidad de líquidos homogéneos de composición claramente definida.

2.7. Refractómetros comerciales

En el mercado existen diferentes tipos de refractómetros tales como: Refractómetros tradicionales como el refractómetro de Abbe [40], refractómetros manuales desarrollados por Brix Baume [41]. Esta clase de sistemas pueden llevar a errores en su medición, además son de uso normalmente académico.

Existen refractómetros digitales como METTLER TOLED [42] y refractómetros de mesa para control de procesos, desarrollados por Bellingham + Stanley Ltd. A pesar de que estos tienen un excelente desempeño de estos equipos no cumplen con la función que busca el prototipo desarrollado ya que estos instrumentos son de uso en laboratorio, de compleja fabricación y alto costo. Y finalmente refractómetros multiparamétricos como los que maneja LiquidPhysics [43], que presenta las desventajas antes mencionadas.

2.8. Conclusión

Una vez introducidos los conceptos necesarios para comprender el principio de funcionamiento del sensor, en el siguiente capítulo se explica el sistema óptico empleado. El prototipo desarrollado no tiene como objetivo competir contra los refractómetros comerciales existentes en el mercado, su misión es explotar un nuevo campo de aplicaciones en la industria al medir IR y actuar como un filtro de control de calidad en el campo.

CAPÍTULO 3

Diseño y desarrollo del sistema óptico

En esta capítulo se explicará a detalle el sistema óptico empleado en el prototipo, el modelo matemático y la simulación del concepto. Los experimentos realizados para la prueba concepto y los resultados se muestran al final.

3.1. Diseño del refractómetro multipunto

El refractómetro emplea la técnica de C-OFDR para su funcionamiento. Como se mencionó en el capítulo II, esta técnica utiliza una TLS lo que permite la implementación de un diodo láser estándar como fuente óptica y un fotodiodo como detector (Fig. 3.1).

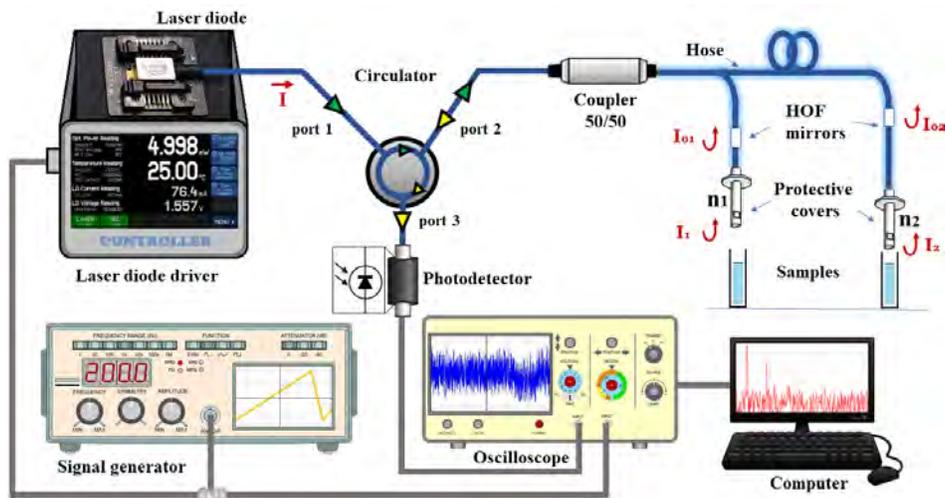


Figura 3.1: Diagrama esquemático del refractómetro multipunto, empleando la técnica C-OFDR

La Fig. 3.1, muestra el esquema del refractómetro propuesto. Se utiliza un diodo láser DFB como fuente de luz, este es sintonizado en longitud de onda con una señal de diente de sierra. La sintonización se logra modulando la corriente de bombeo del diodo láser. La señal proveniente del láser entra por el puerto 1 del circulator y se direcciona al puerto 2 del mismo. El puerto 2 del circulator está conectado a un acoplador de fibra 50/50 que divide el haz en dos porciones iguales y las dirige a cada una a los sensores.

Cada sensor está formado por una cavidad tipo Fabry-Perot (F-P). Las superficies reflejantes (o espejos) que conforman cada cavidad son: un fragmento de fibra tipo HOF y la punta de fibra donde se implementa la reflexión de Fresnel. La intensidad de la luz reflejada en las puntas es en proporción al cambio de IR experimentado que depende del coeficiente de reflexión de Fresnel. Estas hacen interferencia con el espejo tipo HOF que corresponde para cada una de las cavidades y posteriormente interfieren en el acoplador. La señal total sale simultáneamente a través del puerto 3 del circulador y es detectada por un fotodetector estándar. Finalmente se realiza el procesamiento de la señal para su demultiplexación e identificar la señal de resonancia de cada sensor.

Para facilitar la manipulación, y eliminar el ruido en las cavidades, se implementó una cubierta protectora sobre cada sensor (Fig. 3.2); la cubierta protectora consiste en una aguja de jeringa que pasa a través de un barril de plástico. La fibra óptica se inserta a través de la aguja de la jeringa hasta que la punta quede expuesta, además tienen un soporte para sostener la aguja de la jeringa y un conducto al extremo del barril de plástico para facilitar el flujo de las muestras dentro de la cubierta.

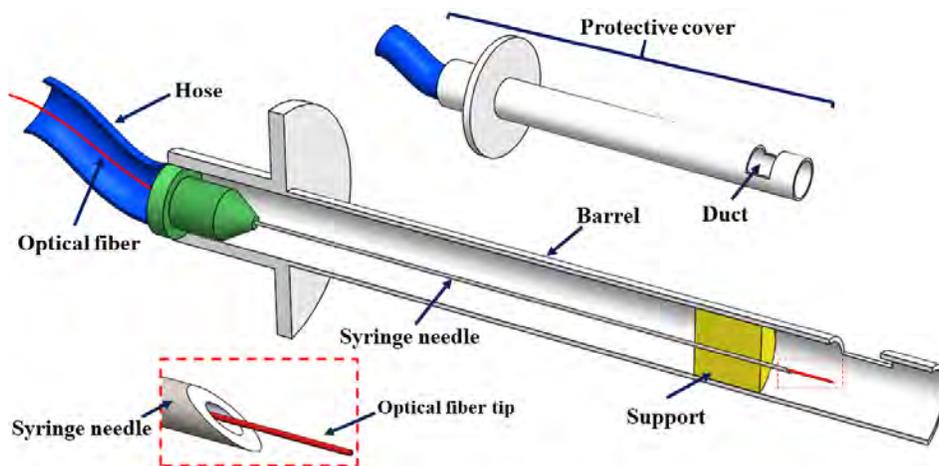


Figura 3.2: *Cubierta protectora de los sensores F-P.*



Figura 3.3: *Fotografía de la cubierta protectora de los sensores.*

3.2. Simulación del sistema

Para la simulación del sistema se sigue con el análisis descrito a continuación. En cada espejo de los interferómetros F-P, la intensidad de la luz reflejada depende del cambio de IR que se presenta en el medio externo; guiándose en la Fig. 3.1 la intensidad de la luz reflejada se puede escribir como:

$$I_{01} = I_{02} = \frac{1}{2} I \left(\frac{n_f - n_0}{n_f + n_0} \right)^2 \quad (3.1)$$

$$I_1 = \frac{1}{2} I \left(\frac{n_f - n_1}{n_f + n_1} \right)^2 * \left[1 - \frac{n_f - n_0}{n_f + n_0} \right]^2 \quad (3.2)$$

$$I_2 = \frac{1}{2} I \left(\frac{n_f - n_2}{n_f + n_2} \right)^2 * \left[1 - \frac{n_f - n_0}{n_f + n_0} \right]^2 \quad (3.3)$$

donde I es la intensidad de luz proporcionada por el diodo láser (fuente óptica); I_{01} e I_{02} son la intensidad de los haces reflejados en las fibras tipo HOF (primer espejo de cada interferómetro F-P), I_1 e I_2 son la intensidad del haz en las puntas de las fibras 1 y 2 (segundo espejo para cada interferómetro F-P, respectivamente); n_f es el índice efectivo del núcleo de la fibra; n_1 y n_2 son los índices de refracción que rodean la punta de fibra 1 y 2.

Las cavidades interferométricas generan las señales de interferencia I_{011} e I_{022} . La señal de interferencia generada en estos interferómetros se puede escribir de la siguiente manera:

$$I_{011}(ODP, t) = I_{01} + I_1 + 2\sqrt{I_{01}I_1} \cos \left(\frac{2\pi\Delta\nu\nu_m}{c} OPD t + \frac{2\pi}{\lambda_0} OPD \right) \quad (3.4)$$

$$I_{02}(ODP, t) = I_{02} + I_2 + 2\sqrt{I_{02}I_2} \cos \left(\frac{2\pi\Delta\nu\nu_m}{c} OPD t + \frac{2\pi}{\lambda_0} OPD \right) \quad (3.5)$$

donde $\Delta\nu$ es la excursión de modulación de frecuencia óptica, ν_m es la frecuencia de modulación, OPD es la diferencia de camino óptico entre los dos haces de interferencia de cada interferómetro, c es la velocidad de luz en el espacio, λ_0 es la longitud de onda óptica central, y t es el tiempo. Posteriormente la interferencia total que se realiza entre estas dos intensidades queda representada como:

$$I_T(OPD_1, OPD_2, t) = R * \left[I_{01} + I_1 + \sqrt{I_{01}I_1} \cos \left(\frac{2\pi\Delta\nu\nu_m}{c} OPD_1 t + \frac{2\pi}{\lambda_0} OPD_1 \right) \right] \\ + R * \left[I_{02} + I_2 + \sqrt{I_{02}I_2} \cos \left(\frac{2\pi\Delta\nu\nu_m}{c} OPD_2 t + \frac{2\pi}{\lambda_0} OPD_2 \right) \right] \quad (3.6)$$

donde R es la capacidad de respuesta del fotodiodo.

Los datos de simulación son los mismos que se usan para realizar los experimentos:

- El barrido de la señal en λ es de 68 pm , lo que equivale en frecuencia a $\Delta\nu$ de 123.43 GHz
- Longitud de las cavidades F-P de 48cm y 75cm
- λ_0 de 1550 nm

- Frecuencia de modulación de 200 mHz En la Fig. 3.4 se muestran los resultados de la simulación

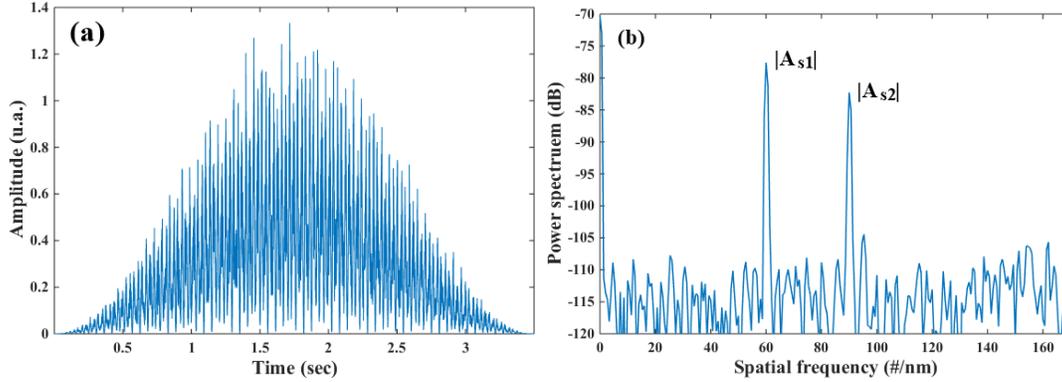


Figura 3.4: Señal simulada para los interferómetros F-P; (a) espectros de los interferómetros, (b) componentes de frecuencia de cada interferómetro F-P.

```

% Creación del patrón de interferencia
I01 = (0.5)*R0*I;
I02 = (0.5)*R0*I;
I1 = (0.5)*R1*I*(1-R0);
I2 = (0.5)*R2*I*(1-R0);

Deltav=6.8e-5;
vm=200e-3;
OPD1=2*1.4446*L011;
OPD2=2*1.4446*L022;
c=3e14;
t= linspace(0,3.5,length(Lm));
Delta0=1550;
If011 = I01 + I1 + 2 * sqrt(I01.*I1).*cos(((2*pi*Deltav*vm*OPD1)/(c)).*t+((2*pi./Lm)*OPD1));
If022 = ( I02 + I2 + 2 * sqrt(I02.*I2).*cos(((2*pi*Deltav*vm*OPD2)/(c)).*t+((2*pi./Lm)*OPD2)))*0.6;
IfFi=If011+If022;

% Obtención de la transformada
w=hanning(length(IfFi));
figure(2);
plot(Lm,IfFi,'b');
IfFi=If;
Fs=10000;
L=length(IfFi);
f = Fs*(0:(L/2))/L;
IfFi=IfFi.*w';
S1 = abs(fft(IfFi).*conj(fft(IfFi)));
T1 = abs(S1/L);
H1 = T1(1:L/2+1);
H1(2:end-1) = 2*H1(2:end-1);
figure(5);
plot(f,10*log10(H1),'LineWidth',1); hold on;
axis([0 170 -120 -70]);
xlabel('Spatial frequency (#/nm)'); ylabel('Power spectrum (dB)');

% Definición de constantes
clc; clear all; close all;

nf = 1.4446;
n0 = 1.0029;
n1 = 1.0029;
n2 = 1.0029;
L011 = 480000;
L022 = 720000;
R0 = ((nf-n0)/(nf+n0))^2;
R1 = ((nf-n1)/(nf+n1))^2;
R2 = ((nf-n2)/(nf+n2))^2;

Lm = 1.550:1.1e-8:1.5502;

Phi011 = ((2*pi)/Lm)*nf*2*L011;
Phi022 = ((2*pi)/Lm)*nf*2*L022;

I = normpdf(Lm,1.5501,0.002);
figure(1);
plot(Lm,I);

```

Figura 3.5: Codificación para la simulación del sistema óptico.

3.3. Principio de funcionamiento del refractómetro multipunto

Una característica importante del refractómetro es que se utiliza fibras HOF, con diámetros interior y exterior de $56 \mu\text{m}$ y $125 \mu\text{m}$ respectivamente, empalmados en ambos lados con una SMF como espejos que forman parte de las cavidades F-P (Fig. 3.6). La fibra HOF forma en sí un interferómetro, por lo que genera su propio interferograma, sin embargo éste posee un

periodo de varios nanómetros. Por lo tanto, al tener un barrido en sintonización de 68 pm (picometros), da como resultado que el valor de reflexión en la HOF sea casi constante; generando un espejo pseudo-estable, debido a que la señal de sintonización se posiciona en un valor que se considera constante dentro del interferograma de los capilares, Fig. 3.7. La cavidad HOF no se ve afectada por cambios de temperatura.

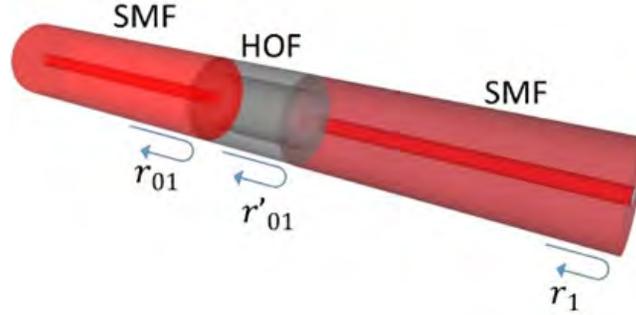


Figura 3.6: *Fibras HOF*

Teniendo en cuenta las interfaces fibra-aire-fibra generadas en el HOF, y siguiendo el esquema mostrado en la Fig. 3.6, para el sensor F-P 1, la luz reflejada desde el espejo HOF se puede escribir como:

$$I_{01} = \frac{1}{2} I t_{01}^2 \gamma_{01}^2 \alpha_{01}^2 r_{01}^2 r'_{01}{}^2 \quad (3.7)$$

donde I es la intensidad de luz proporcionada por la fuente óptica, t_{01} y γ_{01} son el coeficiente de transmisión y el coeficiente de atenuación en la interfaz fibra-aire del HOF, respectivamente; este coeficiente de atenuación tiene en cuenta cualquier pérdida generada debido a la naturaleza de un empalme no perfecto entre la fibra SMF y el HOF. α_{01} corresponde a las pérdidas de la cavidad a lo largo de la longitud de la fibra HOF. r_{01} y r'_{01} son el coeficiente de reflexión de amplitud en las interfaces fibra-aire y aire-fibra de la HOF.

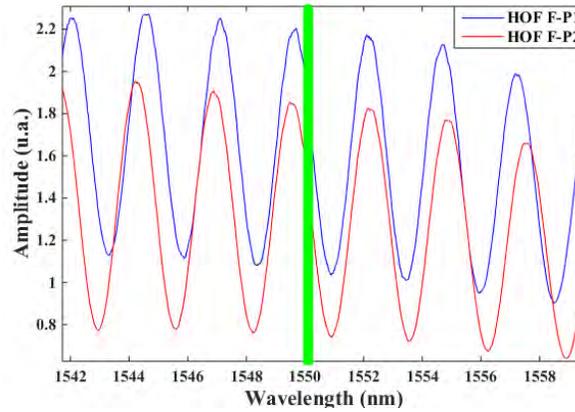


Figura 3.7: *Espectros de las fibras HOF, utilizadas como espejos dentro de las cavidades interferométricas, la línea verde simboliza la posición y ancho del barrido del diodo láser.*

Haciendo $m_{01} = t_{01}^2 \gamma_{01}^2$ y $r_{01}^2 = r'_{01}{}^2 = \left| \frac{n_f - n_0}{n_f + n_0} \right|^2$ donde n_f es el índice efectivo del núcleo de fibra y n_0 es el índice de refracción del aire en la HOF, se puede escribir:

$$I_{01} = \frac{1}{2} I m_{01} r_{01}^4 \quad (3.8)$$

La luz reflejada en la punta de la fibra que interfiere con I_{01} en la HOF se puede expresar como:

$$I_1 = \frac{1}{2} I t'_{01}{}^2 \gamma'_{01}{}^2 r_1^2 = \frac{1}{2} I m'_{01} r_1^2 \quad (3.9)$$

donde t'_{01} y γ'_{01} son el coeficiente de transmisión y el coeficiente de atenuación en la interfaz de aire-fibra de la HOF, respectivamente; r_1 es el coeficiente de reflexión de amplitud en la punta de la fibra; y $m'_{01} = t'_{01}{}^2 \gamma'_{01}{}^2$. La atenuación en la fibra no se considera en la ecuación. Por lo tanto para el sensor dos, tenemos:

$$I_{02} = \frac{1}{2} I m_{02} r_{02}^4 \quad I_2 = \frac{1}{2} I m'_{02} r_2^2$$

La luz reflejada desde cada sensor llega al fotodetector, generando una señal que incluye todas las señales dentro del sistema. Obteniendo nuevamente la Ec. 3.6 de la simulación. Al obtener la transformada de Fourier de la señal se obtienen los picos en frecuencias de cada componente de los interferómetros F-P, Fig. 3.4(b).

El componente de amplitud $|A_{s1}|$, corresponde al sensor F-P 1 y se puede escribir como:

$$\begin{aligned} |A_{s1}| &= 2R \sqrt{\frac{1}{2} I m_{01} r_{01}^4 * \frac{1}{2} I m'_{01} r_1^2} \\ &= R I r_{01}^2 r_1 \sqrt{m_{01} m'_{01}} \end{aligned} \quad (3.10)$$

Donde $r_1 = \frac{n_f - n_1}{n_f + n_1}$, y n_1 es el IR que rodea la punta de la fibra. Lo mismo ocurre para el sensor F-P cuyo componente en amplitud es $|A_{s2}|$:

$$|A_{s2}| = R I r_{02}^2 r_2 \sqrt{m_{02} m'_{02}} \quad (3.11)$$

El procedimiento para calcular el valor de IR en cada una de las puntas es el siguiente:

1. Se obtienen las amplitudes de referencia tomando una medición con un medio IR bien conocido ($n_1 = n_2 = \text{IR de aire} = n_0$).
2. Las puntas de los sensores se sumergen en la muestra de IR desconocido (n_1 y n_2) y se miden nuevamente las amplitudes.
3. Se obtiene la relación de las amplitudes y se calcula el IR desconocido que rodea las puntas de la siguiente manera:

$$\frac{|A_{s1ref}|}{|A_{s1}|} = \frac{R I r_{01}^2 r_{1ref} \sqrt{m_{01} m'_{01}}}{R I r_{01}^2 r_1 \sqrt{m_{01} m'_{01}}} = \frac{\frac{n_f - n_0}{n_f + n_0}}{\frac{n_f - n_1}{n_f + n_1}} \quad (3.12)$$

donde r_{1ref} es el coeficiente de reflexión obtenido con la punta de la fibra en el aire. Haciendo $D = \frac{A_{s1ref}}{|A_{s1}|}(n_f + n_0)$ y $\Delta n = n_f - n_0$, se tiene:

$$n_1 = n_f \frac{D - \Delta n}{D + \Delta n} \quad (3.13)$$

Con el procedimiento descrito la medición y cálculo de IR es independiente de la intensidad y también se eliminan los efectos de empalmes.

3.4. Prueba de concepto

El refractómetro propuesto se probó con varias muestras líquidas simultáneamente en los dos puntos de detección (sensores), de las puntas de fibra 1 y 2. Un espectro típico de la señal detectada se muestra en la Fig. 3.8.

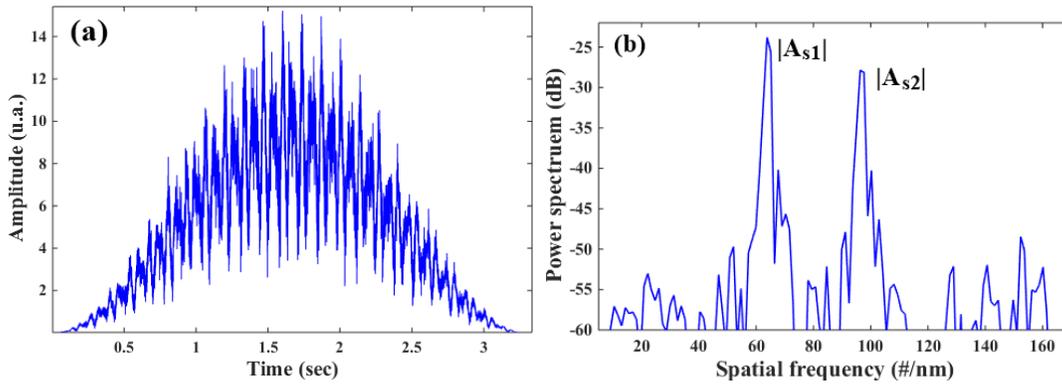


Figura 3.8: Espectro de reflexión del arreglo experimental (a) interferograma obtenido en el dominio del tiempo después de procesar la señal aplicando una ventana de Hanning; (b) espectro en el dominio de frecuencia del interferograma medido

Cada una de las puntas de los sensores se probó con diferentes solventes tales como metanol, acetona, alcohol isopropílico y cloroformo. Para calcular el valor de IR de cada muestra, como lo indica el procedimiento antes explicado, primero se toma una medida de referencia sumergiendo las puntas del sensor en un medio de IR conocido (como el aire). Posteriormente, las puntas del sensor se sumergen en la muestra a medir. Cada medición se transforma en el dominio de la frecuencia para identificar los picos de frecuencia de cada interferómetro F-P (1 y 2). Luego, para cada una de las mediciones, se obtiene la relación entre las amplitudes; una vez que se completó el proceso de calibración, calculamos los valores de IR Fig. 3.10. Se tomaron 8 mediciones para cada solvente.

El error promedio obtenido con los valores de IR calculados fue de 7×10^{-5} en el sensor 1, y de 1.3×10^{-4} en el sensor 2. En el rango de IR de 1.000 a 1.432, la variación de la amplitud del pico de frecuencia es de 32 dB, que corresponde a una sensibilidad de RI de 74 dB/RIU. El valor calculado de IR es muy similar para ambos sensores.

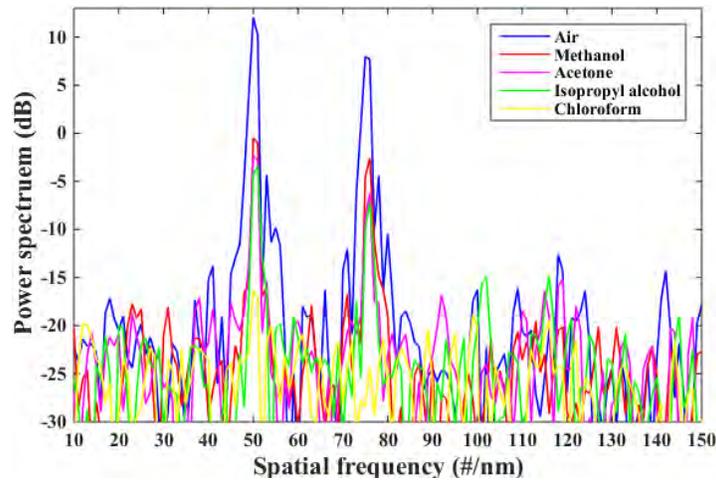


Figura 3.9: Espectro en frecuencia de los interferómetros F-P, con ambos sensores simultáneamente sumergidos en cada una de las muestras.

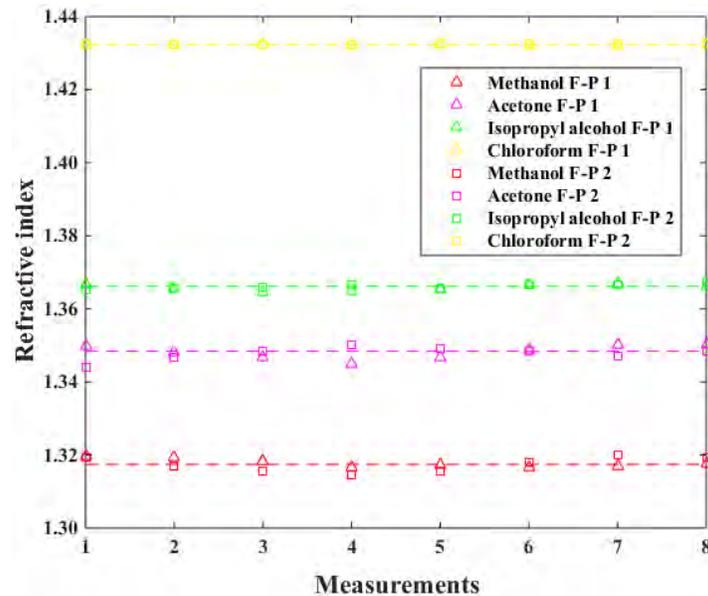


Figura 3.10: Valores de IR calculados usando los dos sensores F-P para medir los diferentes solventes: metanol, acetona, alcohol isopropílico y cloroformo.

La Tabla 3.1, muestra la comparación entre los valores de IR calculados con el sistema óptico, contra los valores de la literatura reportados a 1550 nm

Otro experimento realizado fue medir soluciones de agua destilada con etilenglicol; aumentando la concentración en 10 % de etilenglicol en el agua destilada hasta alcanzar etilenglicol puro; cubriendo el rango de RI de 1.315 a 1.4176. La Fig. 3.11 muestra los valores obtenidos después de la calibración.

Tabla 3.1: Comparación entre los valores de IR calculados contra los valores registrados en la literatura [9]

Muestra	Valores calculados de IR con el sistema óptico diseñado	Valores de IR en la literatura
Metanol	1.3174	1.3174
Acetona	1.3484	1.3483
Alcohol Isopropílico	1.3661	1.3661
Cloroformo	1.4323	1.4321

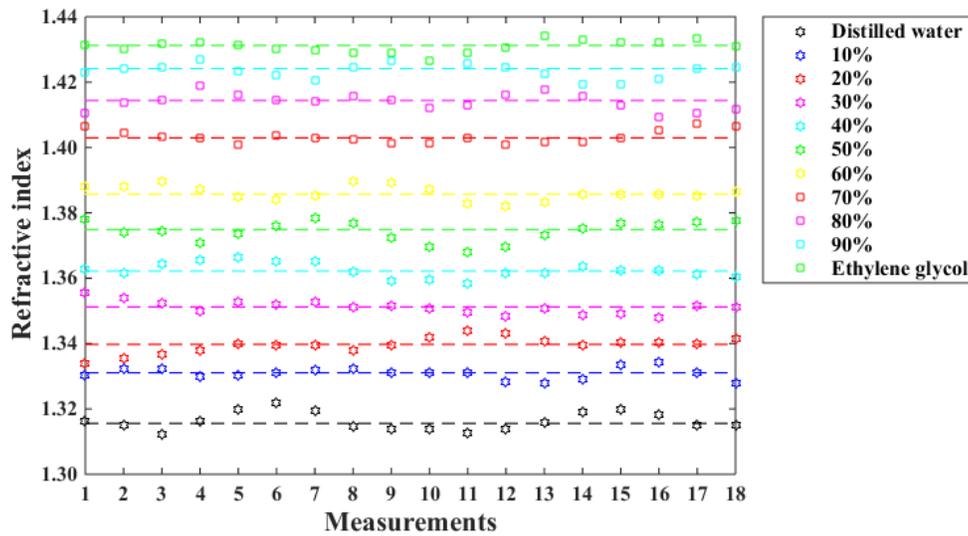


Figura 3.11: Valores medidos de soluciones de agua destilada con etilenglicol, la línea continua representa el valor promedio de cada ronda de mediciones.

La relación de la variación de la concentración contra el valor de IR muestra cómo la pendiente disminuye a medida que se aproxima al valor del índice efectivo de la fibra (Fig. 3.12).

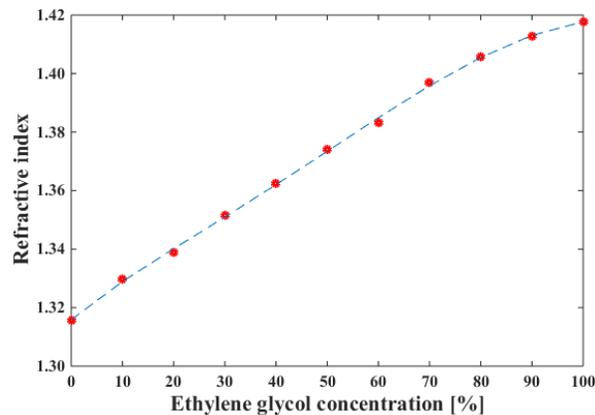


Figura 3.12: Relación de la concentración de la solución vs el IR calculado.

Una prueba realizada para implementar el sistema óptico como una prueba de control calidad, fue la discriminación de muestras contaminadas. Se midieron muestras de tequila adulterado con diferentes concentraciones de alcohol de caña. Los valores de IR calculados confirmaron que si es posible discriminar si la muestra está adulterada o no.



Figura 3.13: Fotografía de las muestras homogénea utilizadas para realizar la prueba de control de calidad.

3.5. Conclusión

Los resultados experimentales confirman la funcionalidad del sistema multipunto propuesto; aunque las componentes de frecuencia se ensancharon debido al efecto de no linealidad, en el refractómetro se pueden agregar más de 10 sensores usando cavidades de longitud dentro de los dos metros de fibra. La implementación de una fibra HOF como espejo permite usar el sistema a diferentes valores de longitud de onda.

La relación costo-beneficio del sistema es buena gracias a su simplicidad y la implementación de componentes estándar.

En los siguientes capítulos se describirá la construcción del prototipo utilizando como sensor el refractómetro descrito en este capítulo.

CAPÍTULO 4

Instrumentación del refractómetro portátil

4.1. Tarjeta de adquisición y procesamiento de datos

En esta sección se presenta la tarjeta de desarrollo STM32F767ZI implementada en el prototipo para la adquisición y procesamiento de la señal procedente del sensor. Se detalla el algoritmo de procesamiento, y se explican los entornos de trabajo de STM32CubeMX, y AtollicTrueSTUDIO; programas empleados para la manipulación de la tarjeta.

4.1.1. Tarjeta de adquisición STM32F767ZI NUCLEO 144

La tarjeta de desarrollo seleccionada para la adquisición de las señales y el procesamiento de los datos es la STM32F767ZI Núcleo 144 mostrada en la Fig. 4.1, la cual permite ser programada en el entorno AtollicTrueSTUDIO complementada con STM32CubeMX.

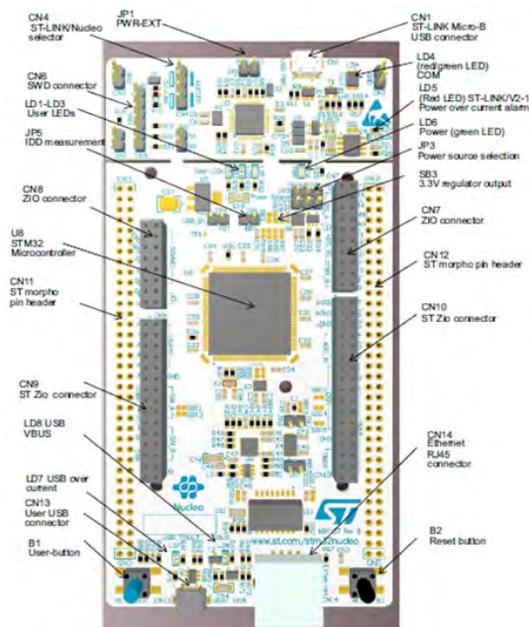


Figura 4.1: Tarjeta de desarrollo STM32F767ZI, apariencia y distribución de módulos.

Este tipo de tarjetas tienen un núcleo tipo RISC (Reduced Instruction Set Computer) de alto rendimiento tipo Arm®Cortex®-M7 de 32 bits y operan a una frecuencia de hasta 216 MHz.

El núcleo cuenta con una unidad de punto flotante (FPU, Floating point unit) que admite las instrucciones de procesamiento para *double-precision* & *single-precision*. También implementa un set completo de instrucciones para procesamiento digital de imágenes (DSP, Digital signal processing) y una unidad de protección de memoria (MPU, MIDI Processing Unit) que mejora la seguridad de la aplicación.

Además, posee tres ADC de 12 bits, dos convertidores digitales a analógico (DAC, Digital-To-Analog converter), doce temporizadores de 16 bits de uso general que incluyen dos temporizadores de modulación de ancho de pulso (PWM, Pulse-width modulation) para el control de motores, dos temporizadores de 32 bits de uso general. También cuentan con interfaces de comunicación estándar y avanzadas las cuales son aplicadas para la comunicación con la pantalla táctil.

La tarjeta se encargará de recolectar la información proveniente del fotodetector, procesarla y calcular el valor de índice de refracción que posteriormente será mostrado en la pantalla de interfaz usuario.

El primer paso para programar la tarjeta es definir las terminales que se usarán de la misma para realizar las funciones requeridas, así como adjuntar las librerías necesarias y configurar la velocidad del reloj para el funcionamiento de la tarjeta. El software de STM32CubeMX es una herramienta gráfica de configuración de software que permite la generación del código de inicialización C utilizando asistentes gráficos. Además, STM32CubeMX cubre todo el portafolio de tarjetas STM32.

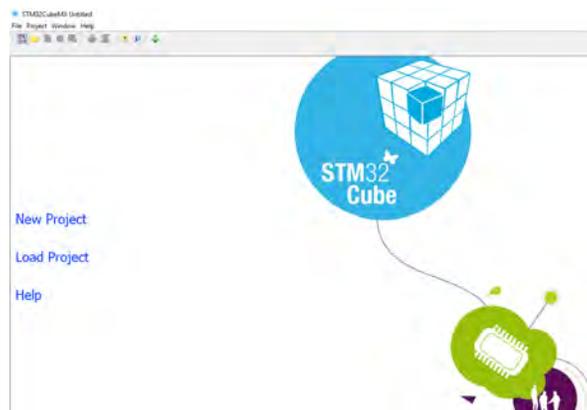


Figura 4.2: *Ventana de inicio de STM32CubeMX.*

El primer paso para crear un nuevo proyecto consiste en seleccionar el microcontrolador STM32 que se utilizará en el prototipo, ver Fig. 4.3, en este caso es la tarjeta de desarrollo STM32F767ZI, se presiona continuar.

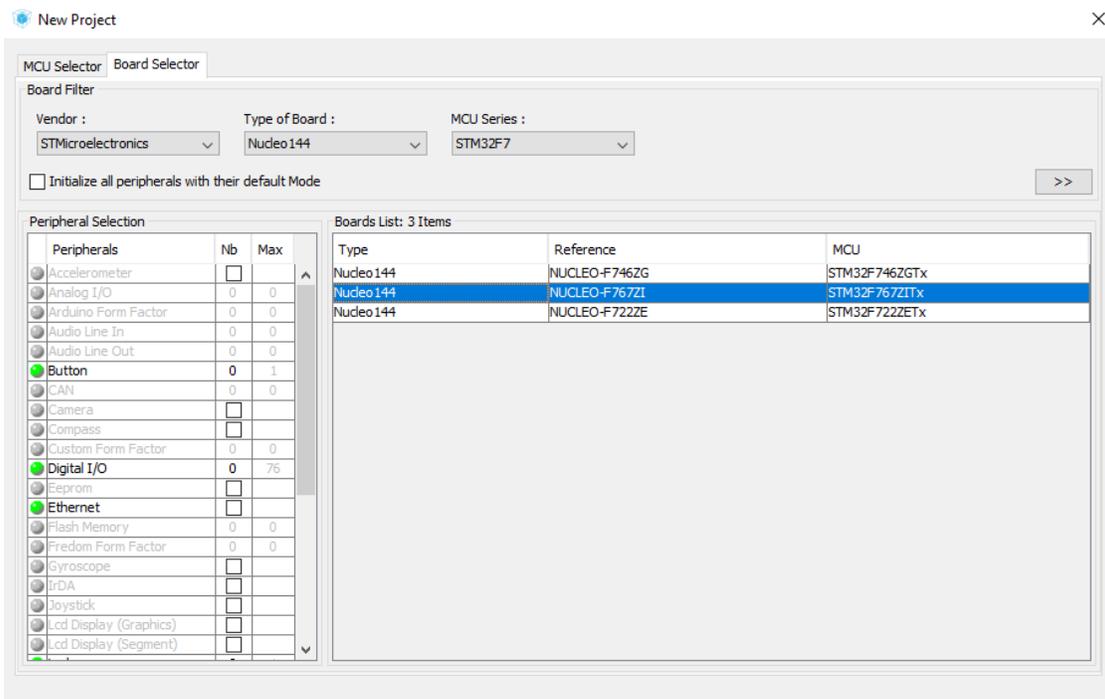


Figura 4.3: Generación de un nuevo proyecto. Dentro del Broad Selector, se utilizan los menús desplegados: “Vendor”, “Type of board” and “MCU Series”, hasta seleccionar la tarjeta a implementar, una vez localizada se da doble clic para crear el proyecto.

Una vez que el nuevo proyecto se crea el usuario puede configurar cada terminal del integrado ajustándose a las necesidades del programa a desarrollar (Fig.4.4).

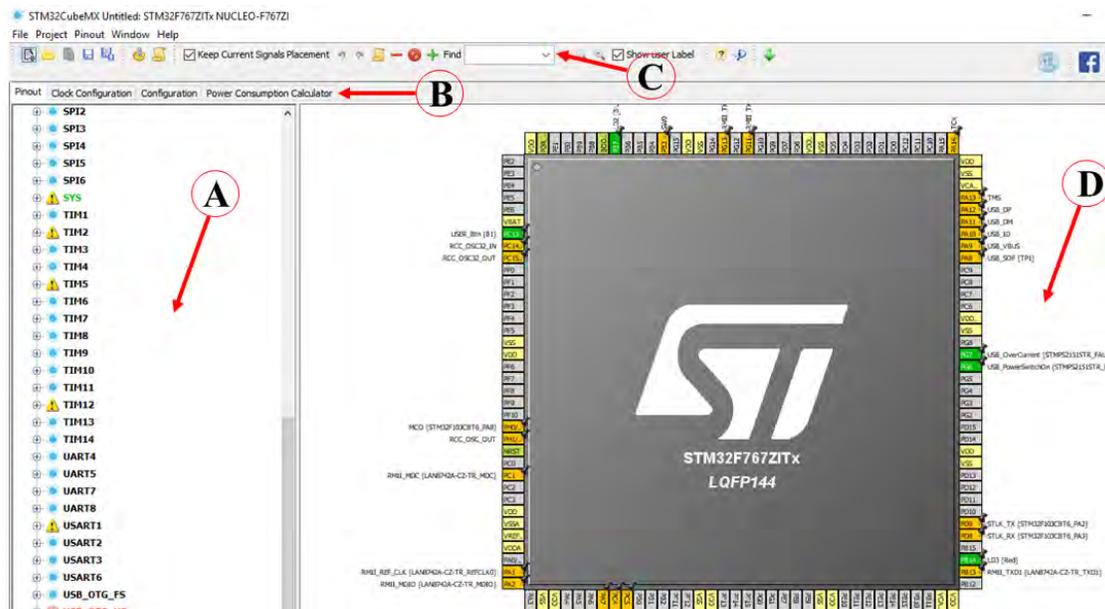


Figura 4.4: Entorno de STM32CubeMX.

Dentro del entorno del software (Fig. 4.4), se pueden realizar las siguientes acciones:

- Seleccionar las librerías complementarias para el programa a desarrollar, activarlas y establecer parámetros iniciales.
- Pestañas de navegación que permite configurar los parámetros de las librerías, del reloj entre otros parámetros.
- Buscar terminales y funciones de forma rápida.
- Configuración gráfica del PinOut del microcontrolador.

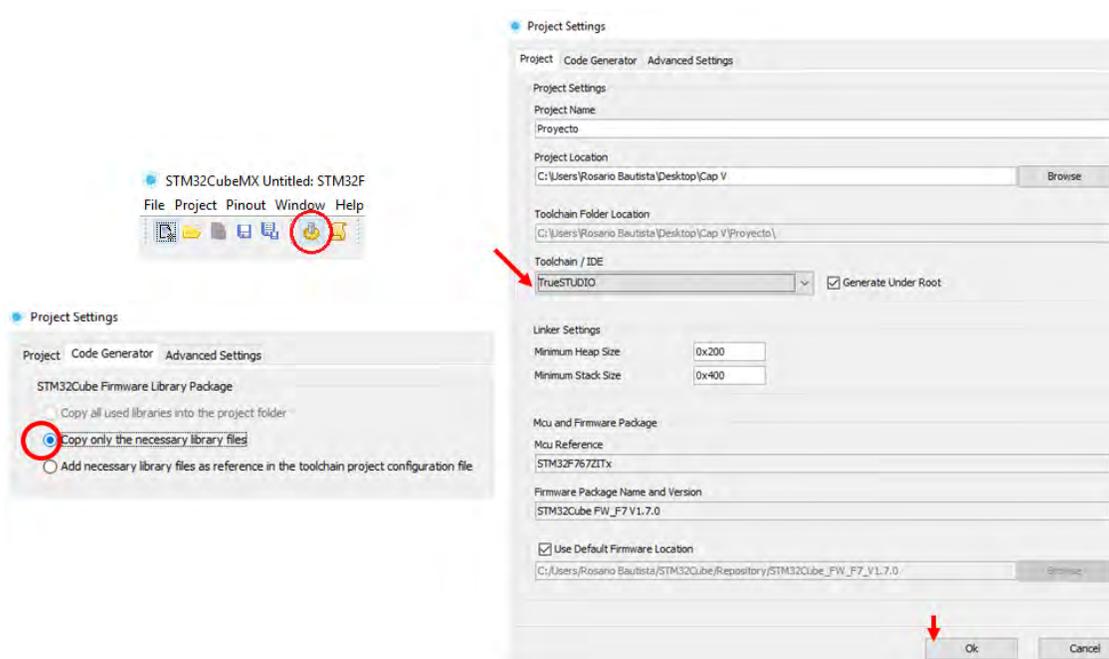


Figura 4.5: Generación del código de inicialización en C.

Finalmente, para generar el código de inicialización se presiona el botón con forma de engrane de la Fig. 4.5, dentro de los parámetros de proyecto (*Project Settings*) se asigna el nombre del proyecto y la ruta donde se guardarán todos los archivos que se generaran del mismo. Otro parámetro que configurar es la herramienta de programación (*ToolchainIDE*), en este caso se utiliza AtollicTrueStudio. Estos 2 softwares quedarán enlazados entre sí, es decir si se modifican los pines del microcontrolador o se selecciona la activación de nuevas librerías en STM32CubeMX, al generar el código nuevamente, solo se actualizará y agregaran las líneas nuevas de código solicitadas dentro del programa ya existente sin afectar el código previo del usuario. Para evitar saturar la memoria del microcontrolador se recomienda seleccionar “Copy only the necessary files” en la pestaña de *Code Generator*. Al dar clic en “ok” en automático correrá el programa de AtollicTruStudio.

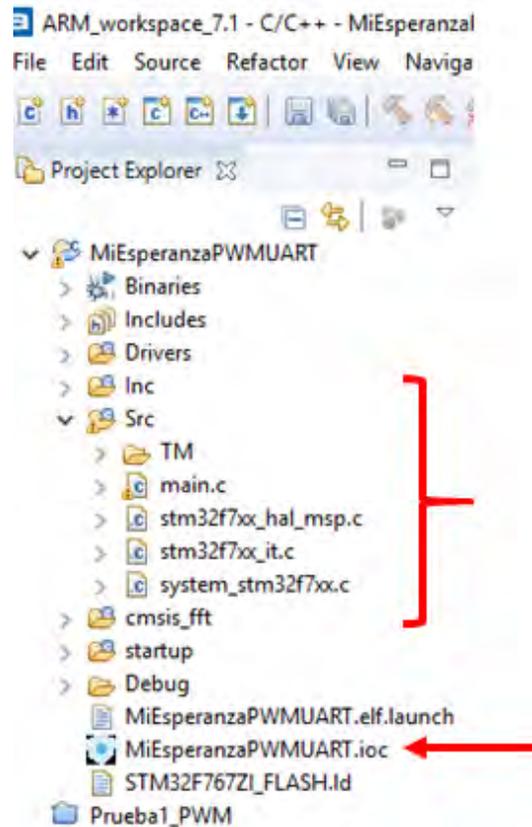


Figura 4.6: *Project Explorer* de *AtollicTrueStudio*, se muestra como el código generado de *STM32CubeMX*, esta enlazado al programa principal, la llave muestra parte de las librerías cargadas gracias a este software.

AtollicTrueStudio es una herramienta de desarrollo C/C++ para tarjetas STM32, que permite programar, compilar y cargar el código dentro de la tarjeta, además ofrece una interfaz de “*Debugger*” con la capacidad de visualizar valores de variables, agregar breakpoints, o comunicarse con la tarjeta para evaluar el funcionamiento del código.

Una vez que *STM32CubeMX* genera el código de inicialización, *AtollicTrueStudio* en automático creará el *workspace* apropiado para continuar con la programación de la tarjeta, el entorno de *AtollicTrueStudio* se muestra en la Fig. 4.6, donde:

- (a) Directorio del proyecto, donde se generan las carpetas necesarias para ordenar y distribuir todos los archivos que formen parte del mismo.
- (b) Editor de texto, en esta sección se escribe el código que controlará la tarjeta.
- (c) Consola de compilación, una vez creado el programa, al compilarlo desplegará el estado general del programa. Revisa que todas las librerías se detecten, que la tarjeta se encuentre correctamente conectada.
- (d) Si existe un problema con la compilación del programa los *warnings* y errores se detallarán en esta sección.

- (e) Botones para arrancar la compilación del programa.
- (f) Botón para cargar el programa en la tarjeta, además abre la ventana “Debugger”.
- (g) Esta sección enumera todas las constantes, variables, y funciones que existen dentro del programa. Sirven como atajos dentro del programa.

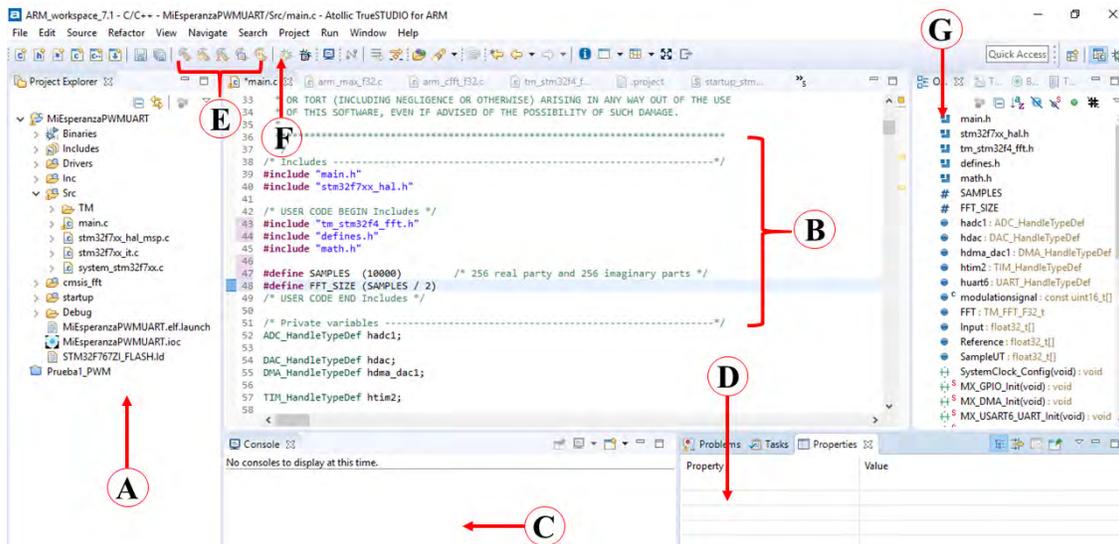


Figura 4.7: Entorno de programación de AtollicTrueStudio.

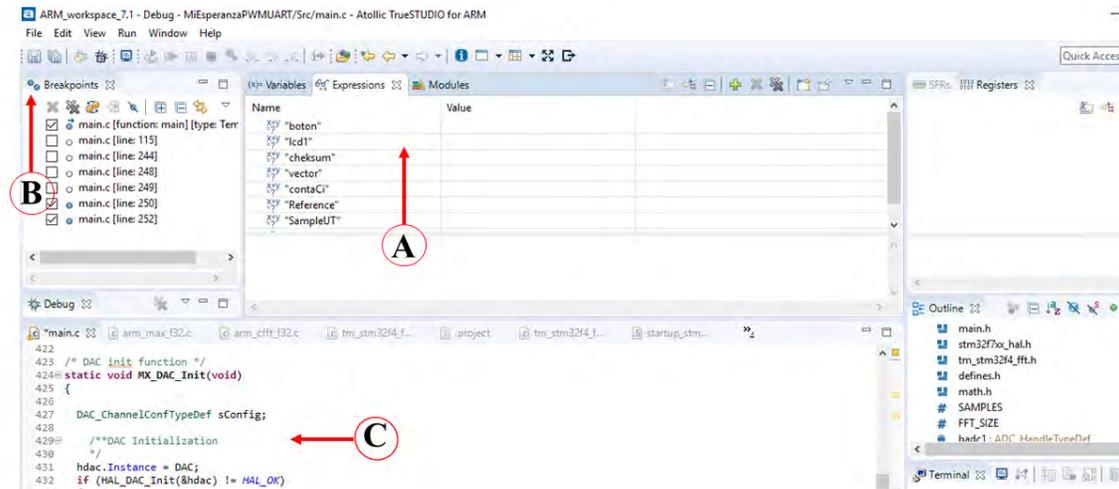


Figura 4.8: Entorno de la ventana de Debugger de AtollicTrueStudio.

En la Fig. 4.8 se observan algunas de las secciones de la ventana Debugger, estas son:

- (a) *Expressions*, permite visualizar el valor de las variables dentro del código, conforme corre el programa.

- (b) *Breakpoints*, ayudan a crear puntos de “paro” que permiten el análisis del código paso a paso.
- (c) Visualización del código para analizar como avanza el programa mientras este corre dentro de la tarjeta.

4.1.2. Generación de la señal de modulación

Para generar la señal de modulación que requiere el sistema para sintonizar el diodo láser, se consideraron los parámetros ya definidos en el sistema óptico (Cap. III), estos son:

- Frecuencia: 200 *mHz*
- Amplitud: 80 *mVpp*
- Offset: +50 *mV*
- Duty cycle: 80 %

La obtención de una señal analógica utilizando la tarjeta de desarrollo se hace con la activación de un DAC en STM32CubeMX, como se muestra en la Fig. 4.9.

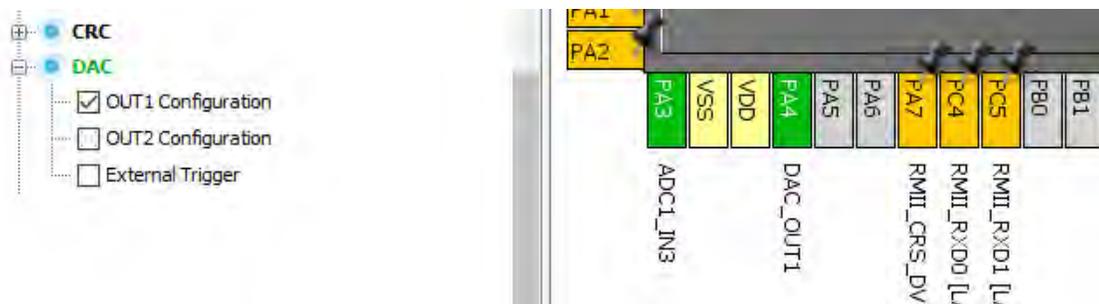


Figura 4.9: Activación de la terminal PA4 de la tarjeta para obtener una señal por parte del DAC, que selecciona la salida 1 (OUT1) para arrancar.

Otro parámetro importante para lograr obtener una señal como la que se requiere es la activación de un *timer* que sirva como interrupción para controlar el flujo de datos. Y para obtener el valor de la frecuencia requerida.

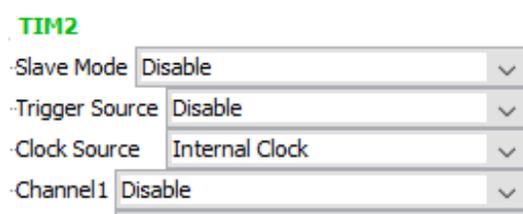


Figura 4.10: Activación del Timer 2 que funcionará con el reloj interno de la tarjeta.

Al trabajar con un timer, es importante conocer a que velocidad trabajan el reloj interno, si se observa el diagrama de la configuración del reloj de la tarjeta (Fig. 4.11). El TIM2 está

unido al reloj de APB1, por lo tanto, todos los cálculos realizados a continuación se harán considerando la frecuencia del reloj a 27MHz.

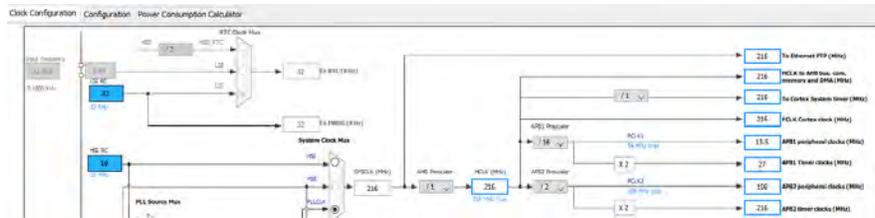


Figura 4.11: Sección de la configuración de reloj dentro de la tarjeta, se observa la frecuencia de trabajo para APB1, con la que el timer seleccionado trabajará.

Una vez generado el código de inicialización, dentro de Atollic se crearan las secciones de código correspondientes para cada parámetro:

```

/* DAC init function */
static void MX_DAC_Init(void)
{
    DAC_ChannelConfTypeDef sConfig;

    /**DAC Initialization
    */
    hdac.Instance = DAC;
    if (HAL_DAC_Init(&hdac) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    /**DAC channel OUT1 config
    */
    sConfig.DAC_Trigger = DAC_TRIGGER_T2_TRGO;
    sConfig.DAC_OutputBuffer = DAC_OUTPUTBUFFER_ENABLE;
    if (HAL_DAC_ConfigChannel(&hdac, &sConfig, DAC_CHANNEL_1) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}

```

Figura 4.12: Sección de código para la inicialización y activación del DAC dentro de la tarjeta.

```

static void MX_TIM2_Init(void)
{
    TIM_ClockConfigTypeDef sClockSourceConfig;
    TIM_MasterConfigTypeDef sMasterConfig;

    htim2.Instance = TIM2;
    htim2.Init.Prescaler = 320;
    htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim2.Init.Period = 843;
    htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    sMasterConfig.MasterOutputTrigger = TIM_TRGO_UPDATE;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}

```

Figura 4.13: Sección de código para la inicialización y activación del TIM2.

La tarjeta de desarrollo tiene un voltaje de operación de 3.3 V y el DAC una resolución de 12 bits, esto da como resultado que se tenga un total de 4095 posiciones, por lo tanto, el valor mínimo de voltaje generado por el DAC es de: $3.3 V/4095 = 805 \mu V$

Como la señal que se quiere es de 80 mV, ($80 mV/805 \mu V$) da como resultado 100, esto quiere decir que para obtener por el DAC una señal de 0 a 80 mV, a la salida del DAC deben enviarse un vector desde 0 a 100.

Para lograr que la frecuencia de la señal sea de 200 mHz, es necesario calcular el Prescaler y Period (divisores de frecuencia para el reloj interno), además de que el vector de valores de voltaje de la señal es de 500, por lo tanto:

$$((27 MHz / \text{Número de elementos del vector de la señal}) / \text{Prescaler}) / \text{Period} = 200 mHz$$

$$((27 MHz / 500) / 320) / 843 = 0.2001 Hz$$

Los valores obtenidos se pueden visualizar en la Fig. 4.13, en el código de inicialización del timer. La señal de sintonización y el código para obtenerla se muestran a continuación:

```

/* USER CODE BEGIN PV */
const uint16_t modulationSignal[500]={62,62,63,63,63,63,64,64,64,65,65,65,65,66,66,66,67,67,67,68,68,68...
HAL_TIM_Base_Start_IT(&htim2);
HAL_DAC_Start(&hdac,DAC_CHANNEL_1);
HAL_DAC_Start_DMA(&hdac, DAC_CHANNEL_1, (uint32_t*)modulationSignal, 500, DAC_ALIGN_12B_R);

```

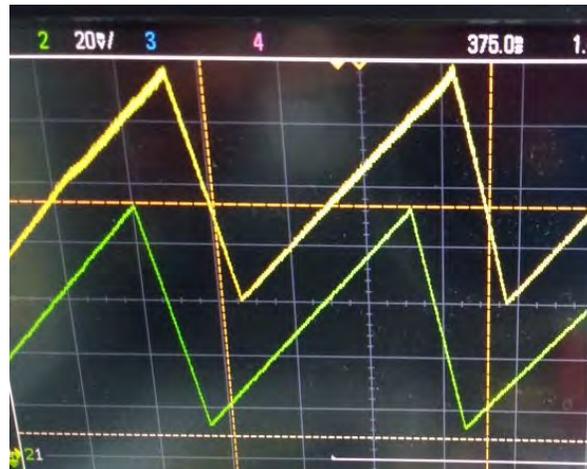


Figura 4.14: Señal de modulación generada por la tarjeta (amarillo), comparada con la señal del generador de funciones (verde).

4.1.3. Lectura de datos con el ADC

Una vez generada la señal de modulación esta se condiciona para entrar al controlador del diodo láser que posteriormente alimenta al sistema óptico. La señal de interferencia del sistema óptico obtenida se muestra en la Fig. 4.15:

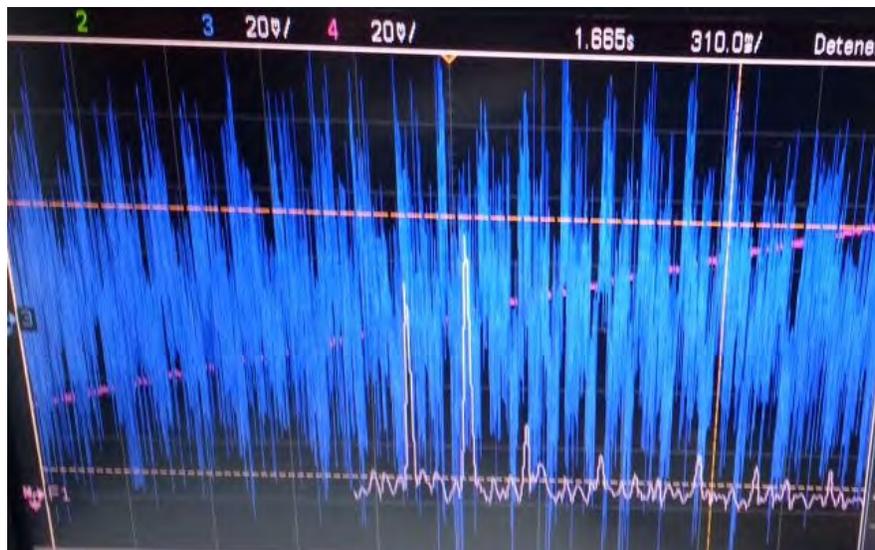


Figura 4.15: Señal de interferencia obtenida por el sistema óptico, con la señal de modulación generada con la tarjeta (azul). Transformada de Fourier de la señal (rosa).

Para leer la señal es necesario activar un ADC de la tarjeta con STM32CubeMX.

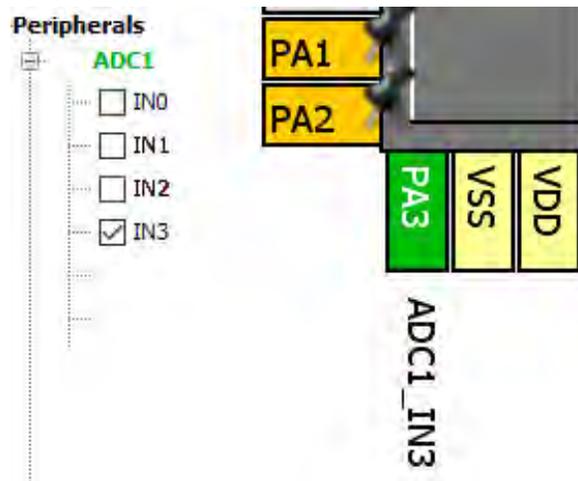


Figura 4.16: Activación del ADC en la tarjeta con STM32CubeMX.

Los parámetros del ADC se definen de acuerdo a la información que se observa en la figura 4.17:

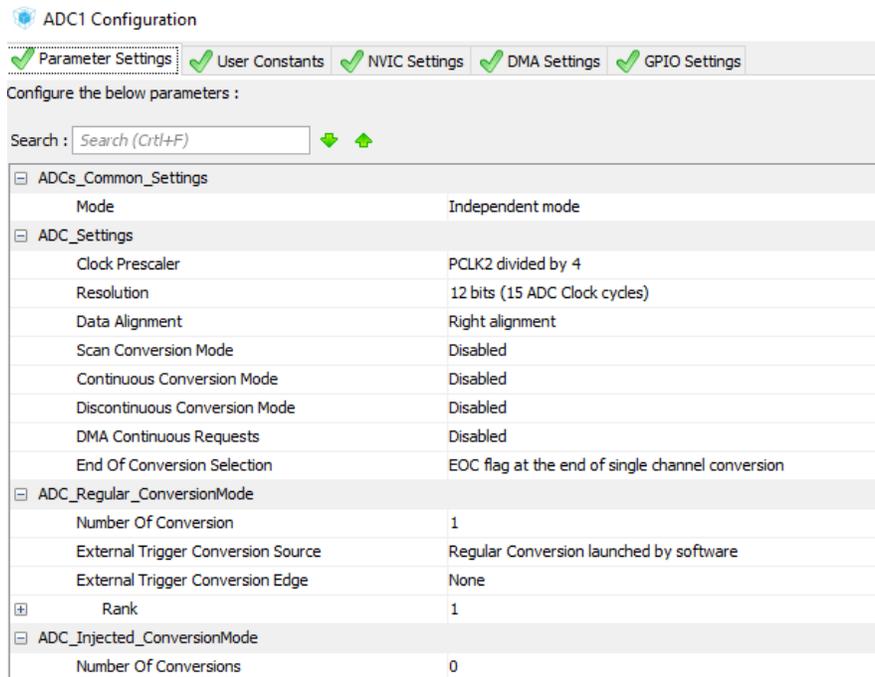


Figura 4.17: Parámetros del ADC.

Generado el código de inicialización una vez en AtollicTrueStudio. Cabe mencionar que dentro del algoritmo de procesamiento, para ahorrar espacio de memoria en la tarjeta; el ADC solo almacena un vector cuando se solicita por el usuario. Además, se ve restringido por el inicio y el final de DAC ver Fig. 4.18. Esto permite identificar el inicio y final de un periodo de la señal de modulación, de este modo se puede recortar y seleccionar el fragmento de la señal a la que se le aplicara la FFT.

```

void HAL_DAC_ConvCpltCallbackCh1 (DAC_HandleTypeDef *hdac)
{
    HAL_GPIO_TogglePin(GPIOB, LD2_Pin);
    if(bandinicio==1){
        contaCi++;
    }
}

```

Figura 4.18: Detección del inicio y el final de la señal generada con el DAC.

La lectura y almacenamiento de los datos se realiza cuando detecta el inicio de la señal de DAC, y cada uno de los valores son almacenados en un vector de 10 000 posiciones. La Fig. 4.19 muestra los códigos de las funciones que se encargan de realizar estas funciones.

```

uint16_t ADC_read(){
    uint16_t auxh=0;
    HAL_ADC_Start(&hadc1);
    HAL_ADC_PollForConversion(&hadc1, 10);
    auxh = HAL_ADC_GetValue(&hadc1);
    return auxh;
}
void obtenerV(void){
    uint16_t aux=0;
    long auxa=0;
    long u=0;
    bandinicio=1;
    while(1){
        if(contaCi==1){
            aux=ADC_read();
            if(u!=0){
                // if((aux<auxa+100)&&(aux>auxa-100)){
                vector[u]=aux;
                auxa=aux;
                //}
            }
            else{
                vector[0]=aux;
                auxa=aux;
            }
            delayUs(250);
            u++;

            if(u==SAMPLES) break;
        }else if(contaCi>1) {break;}
    }
    contaCi=0; bandinicio=0;
}

```

Figura 4.19: Función de adquisición de la señal por el ADC.

4.1.4. Transformada de Fourier

Para obtener la transformada de Fourier del vector almacenado con el ADC, se implementó y modifiqué una librería para las series M4 de los microcontroladores STM32. Por lo que dentro

de cada librería de apoyo fue necesario realizar el cambio de M4 a microcontroladores de la serie M7 [44].

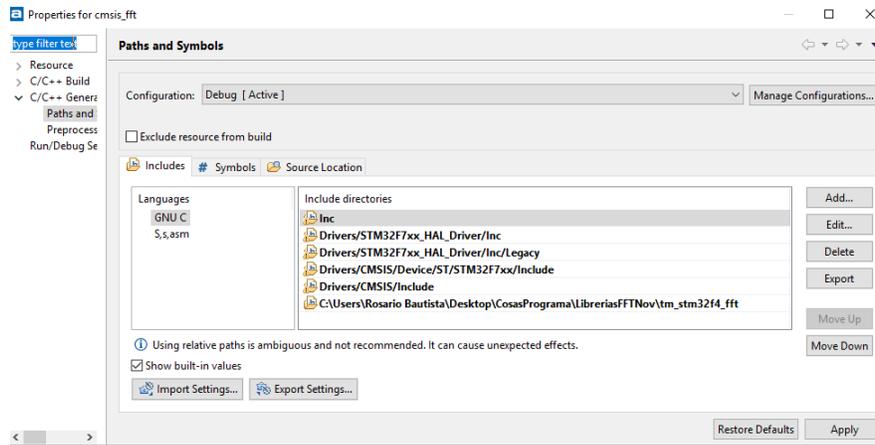


Figura 4.20: Paths dentro de las propiedades del proyecto donde se colocan todas las direcciones desde donde el programa buscare las librerías que necesiten para funcionar [45].

Cuando se aplica la FFT se debe de considerar siempre el vector obtenido es igual a la mitad de la longitud del vector de los datos de entrada para el cálculo, esto se debe a que la librería ARM DSP utiliza una parte real y una parte imaginaria del vector de entrada. Otra característica de esta librería es que maneja datos tipo float32; y además tiene funciones internas para detección de picos máximos. Para comenzar es necesario que en el compilador global se agregue el archivo `#define`, que contenga las siguientes 2 líneas:

```
/* FPU present on STM32F7xx device */
#define __FPU_PRESENT 1

/* Use ARM MATH for Cortex-M7 */
#define ARM_MATH_CM7
```

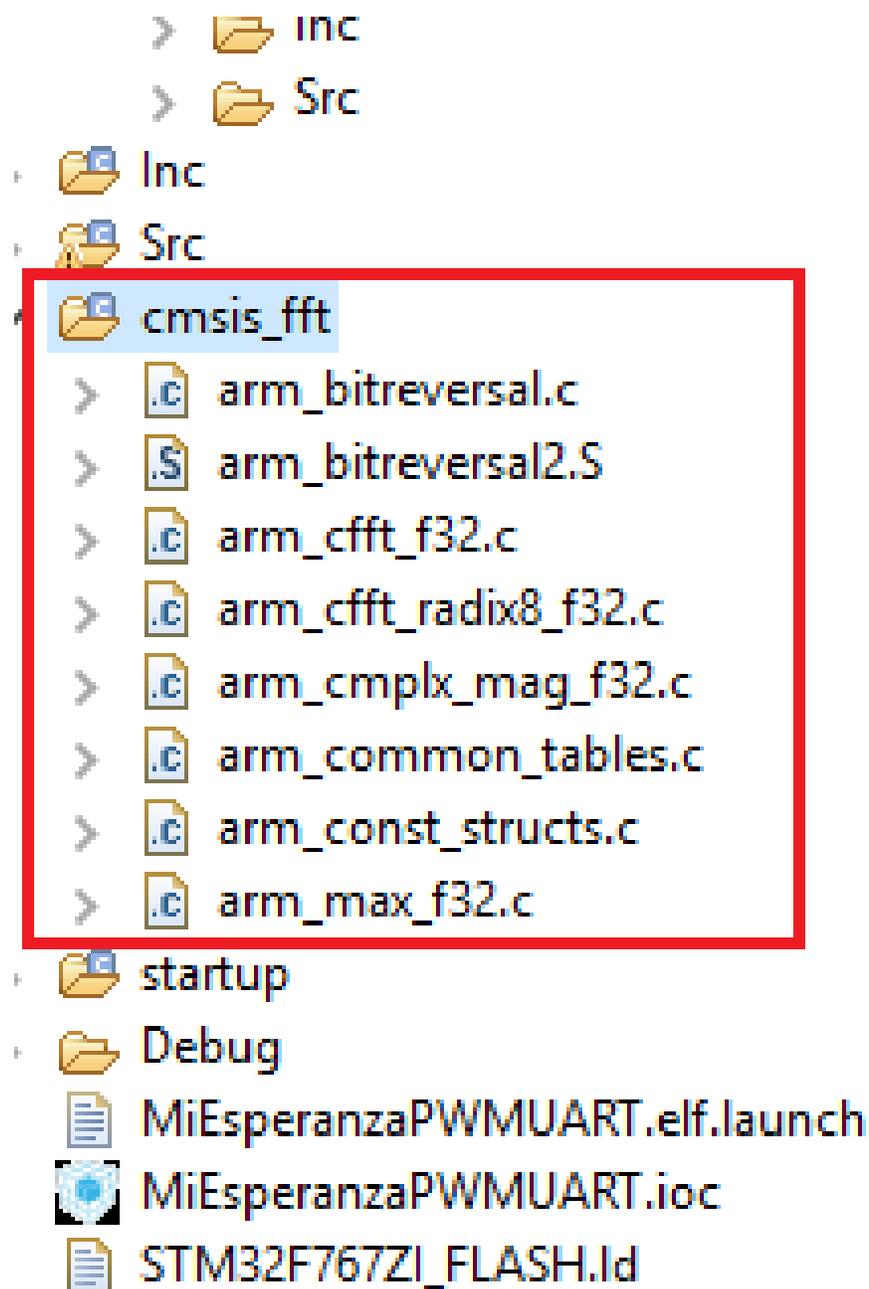


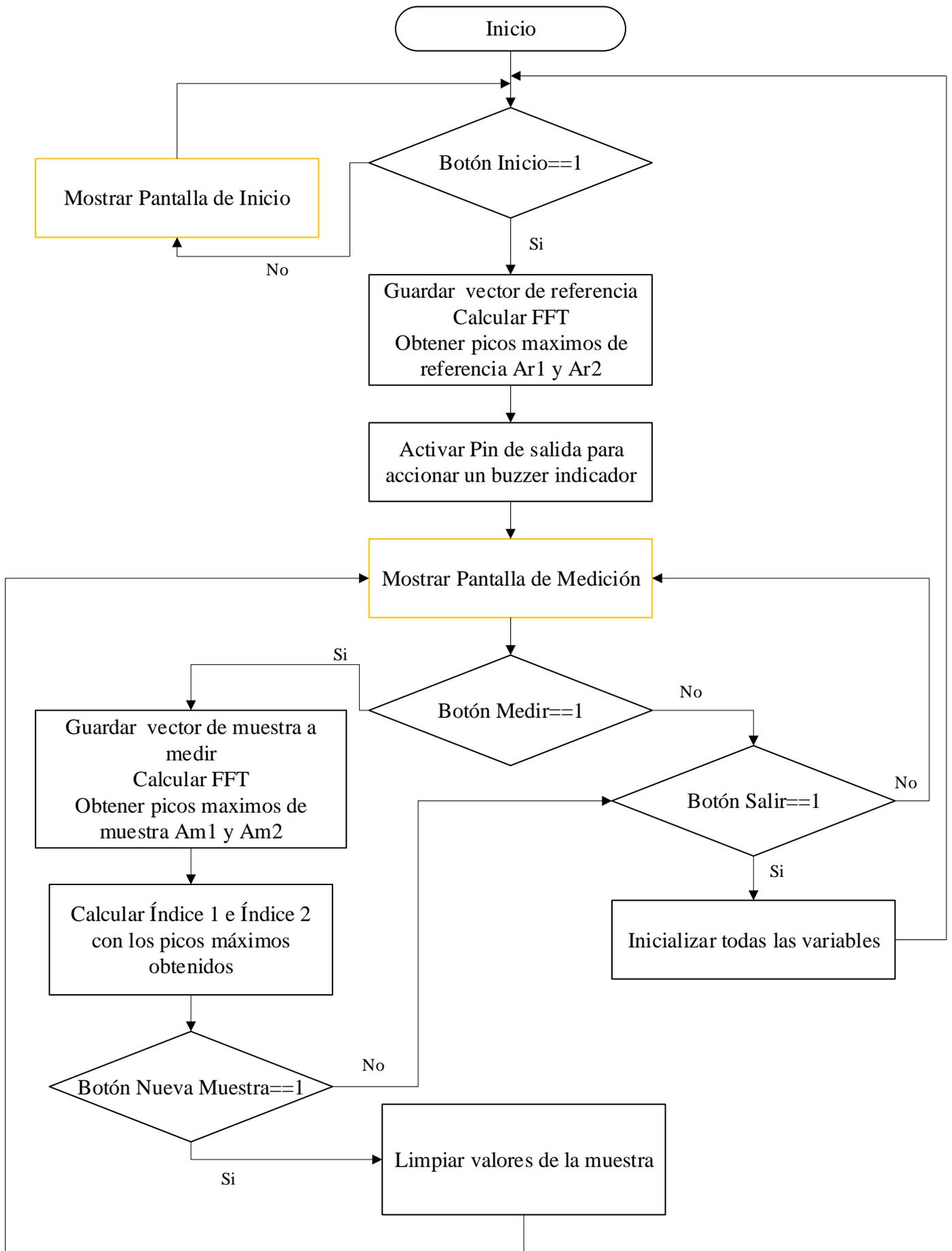
Figura 4.21: Archivos complementarios necesarios para el cálculo de la FFT.

El algoritmo de funcionamiento de esta librería de FFT, consiste en llenar primero el búfer de entrada después de procesarla se obtiene el búfer de salida. El buffer de entrada tiene que ser $2 * FFT_Size$ de longitud y el buffer de salida tiene que tener igual a FFT_Size . Por ejemplo, si tiene un tamaño de FFT de 512, entonces el búfer de entrada debe ser $2 * 512 = 1024$ muestras de float 32 y el búfer de salida es 512 muestras de float 32.

```
/* Private variables -----  
TM_FFT_F32_t FFT;  
  
float32_t Input[SAMPLES]; /*!< Input buffer is always 2 * FFT_SIZE */  
float32_t Reference[FFT_SIZE]; /*!< Output buffer is always FFT_SIZE */  
float32_t SampleUT[FFT_SIZE];  
  
/* USER CODE BEGIN 0 */  
uint8_t t=0, N=0;  
uint16_t vector[10000];  
uint8_t bandinicio=0, contaCi=0;  
  
void transformada(void){  
    long u=0;  
    do{  
        u++;  
    }while(!TM_FFT_AddToBuffer(&FFT, vector[u]/1.0));  
  
    TM_FFT_Process_F32(&FFT);  
}  
  
/* USER CODE BEGIN 2 */  
TM_FFT_Init_F32(&FFT, FFT_SIZE, 0);  
  
case 0x00:  
    TM_FFT_SetBuffers_F32(&FFT, Input, Reference);
```

Figura 4.22: Ejemplo del código para implementar la FFT.

A continuación se muestra el diagrama de flujo de como opera de modo general el algoritmo de la tarjeta:



```

while(boton[0]!=medir[0]&& boton[1]!=medir[1]){
HAL_UART_Receive(&huart6, boton, 6, 250);
}
bp=boton[2];
switch(bp){
case 0x00:
    TM_FFT_SetBuffers_F32(&FFT, Input, Reference);
    HAL_UART_Transmit(&huart6, Form1, 6, 250);
    obtenerV();
    transformada();
    break;
case 0x01: HAL_UART_Transmit(&huart6, Form0, 6, 250); break;
case 0x02: HAL_GPIO_WritePin(GPIOB, LD2_Pin, 1);
    obtenerV();
    transformada();
    HAL_GPIO_WritePin(GPIOB, LD2_Pin, 0);
    lcd1[3]=indice0/258;
    lcd1[4]=indice0&0xFF;
    for(int k=0;k<5;k++){
        if(k==0) checksum=lcd1[0];
        else checksum+=lcd1[k];
    }
    lcd1[5]=checksum;
    HAL_UART_Transmit(&huart6, lcd1, 6, 250);
    varc=0;
    while(varc!=0x06){
        HAL_UART_Receive(&huart6, &varc, 1, 250);
    }
    HAL_UART_Transmit(&huart6, lcd2, 6, 250);
    varc=0;
    while(varc!=0x06){
        HAL_UART_Receive(&huart6, &varc, 1, 250);
    }
    indice0++;
    break;
case 0x03: lcd0[2]=0x00;
    lcd0[5]=0x0E;
    HAL_UART_Transmit(&huart6, lcd0, 6, 250);
    varc=0;
    while(varc!=0x06){
        HAL_UART_Receive(&huart6, &varc, 1, 250);
    }
    lcd0[2]=0x01;
    lcd0[5]=0x0F;
    HAL_UART_Transmit(&huart6, lcd0, 6, 250);
    varc=0;
    while(varc!=0x06){
        HAL_UART_Receive(&huart6, &varc, 1, 250);
    }
    break;
}
for(int i=0;i<6;i++){
    boton[i]=0x00;
}

```

Figura 4.23: Algoritmo de selección de funciones implementado que rige el comportamiento descrito en el diagrama de flujo anterior.

4.1.5. Conclusión

El algoritmo desarrollado con la tarjeta permite controlar la pantalla táctil además de ejecutar el procesamiento adecuado para obtener el cálculo del IR, (el código completo se agrega en el apéndice). Debido a que la señal de modulación tiene un periodo de 5 segundos, y a que la tarjeta toma 10 barridos continuos para trabajar con valores promedio. El tiempo de adquisición, más el tiempo de cálculo de FFT y obtención de picos, hacen que la medición de IR sea aproximadamente de 1.5min (40 seg. para adquisición de la señal de referencia y 40 seg. para adquisición de la señal de la muestra).

El manejo de los softwares de STM32CubeMX y AtollicTrueSTUDIO para manipular la tarjeta permite explotar todas las herramientas que ésta facilita, debido a que son entornos desarrollados con el propósito de trabajar con los procesadores ARM de 32 bits.

Como trabajo a futuro, analizando los resultados obtenidos es posible agregar nuevas funciones dentro del prototipo para generar diferentes aplicaciones.

4.2. Interfaz de humano - máquina: Programación y funcionamiento

En esta sección se explica el manejo de la pantalla táctil GEN4-ULCD-43DCT-CLB, empleada como interfaz humano-máquina. El funcionamiento, entorno de programación, y comunicación con la tarjeta de desarrollo se detallan a continuación.

4.2.1. Pantalla GEN4-ULCD-43DCT-CLB

La pantalla implementada para el prototipo es de la gama Gen4 de 4.3" posee el microprocesador Diablo16 [46] fabricado por 4D Systems. La pantalla es de la tecnología LCD TFT (Thin-film transistor). Tiene la capacidad de detección táctil, almacenamiento en memoria micro - SD, comunicación serial al host deseado, así como generación de audio y reproducción de video.

La gama de la pantalla Gen4, es 100 % compatible con el software de Workshop4 IDE y sus 4 diferentes entornos de desarrollo (explicados en esta sección), proporcionando una gran cantidad de opciones para programar y controlar.

Características de la pantalla:

- Resolución 480 x 272 pixeles.
- Tecnología RGB de 65K colores.
- Panel táctil capacitivo (DCT, Display-Module Capacitive Touch).
- Pestillo incorporado para una tarjeta de memoria micro - SD para almacenamiento multimedia (formato FAT16).
- Admite todas las fuentes de Windows disponibles.
- Rango de voltaje de operación de 4.0V a 5.5V.

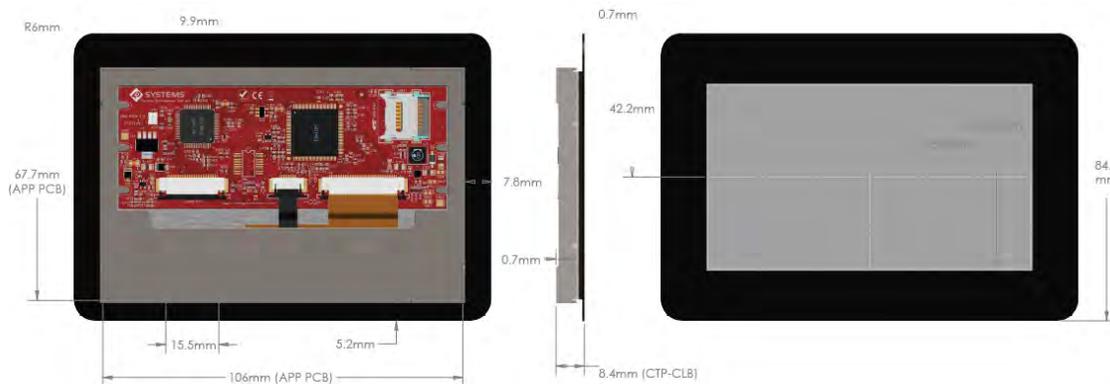


Figura 4.24: Dimensiones y aspecto de la pantalla GEN4-ULCD-43DCT-CLB implementada [47].

4.2.2. 4D SYSTEMS - WORKSHOP 4 IDE

WorkShop 4 IDE es el entorno de programación empleado para manipular la pantalla, esto es debido a que es un software integral que proporciona una plataforma para toda la familia de

procesadores y módulos 4D. Combina el *Editor*, el *Compiler*, el *Linker* y el *Downloader* para desarrollar el código de aplicación (4DGL, 4D Graphics Language [48]) completo.



Figura 4.25: a) Vista de la pantalla de inicio del software *Workshop4IDE*, con los atajos para la creación de proyectos. b) Diferentes entornos de programación para manipular la pantalla.

Workshop 4 IDE tiene múltiples entornos de programación para que el usuario seleccione la que más se ajuste a sus requisitos de diseño. A continuación, se describen brevemente:

Designer: es un entorno que permite al usuario escribir el código 4DGL en su forma natural (programación en línea).

ViSi: habilita la colocación de objetos de la forma “arrastrar” y “soltar” para ayudar con la generación de código 4DGL. Además, la vista previa de la pantalla se visualiza mientras se programa.

ViSi-Genie: es un entorno avanzado que no requiere ninguna codificación 4DGL, se colocan y despliegan los objetos deseados, con la opción de configurar los “eventos” para accionar una tarea determinada, así como modificar su apariencia.

Serial: permite al usuario controlar la pantalla desde cualquier microcontrolador o dispositivo host, usando la pantalla como un módulo esclavo.

4.2.3. Programación de la pantalla con ViSi - Genie

El entorno de programación seleccionado para manipular la pantalla es ViSi-Genie debido a su sencillez y fácil interpretación. Es un entorno visual que permite crear una aplicación GUI (Graphical User Interface) simple desde cero. A continuación, se describen los pasos para la programación de la pantalla, desde la creación del proyecto hasta el diseño final.

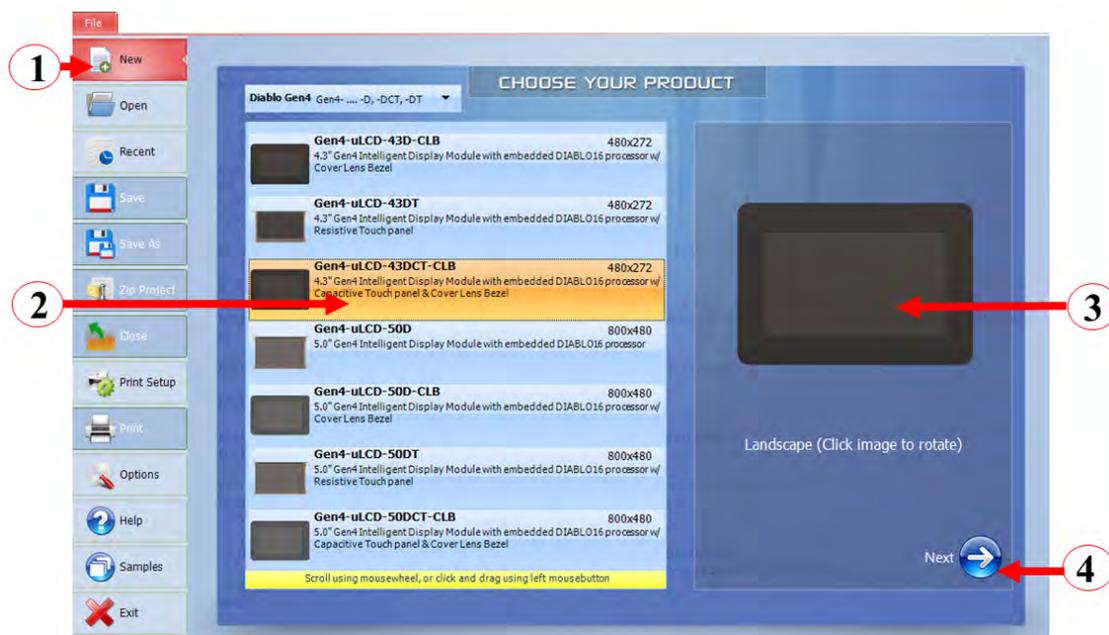


Figura 4.26: Pasos para la creación de un nuevo proyecto.

Tomando como referencia la Fig. 4.26, para la creación de un proyecto nuevo, se siguen los siguientes pasos:

1. Dar click sobre la pestaña “**New**” para crear un nuevo proyecto. En un costado se despliega una lista con los diferentes módulos compatibles con el software.
2. Seleccionar el módulo de pantalla con el que se trabajará.
3. Seleccionar la posición en la que se diseñará el GUI en la pantalla, (Landscape o Portrait).
4. Presionar “**Next**” para seleccionar el entorno de programación deseado, explicados anteriormente (Fig. 4.25(b)). Para este proyecto se utilizó en entorno de Visi-Genie.

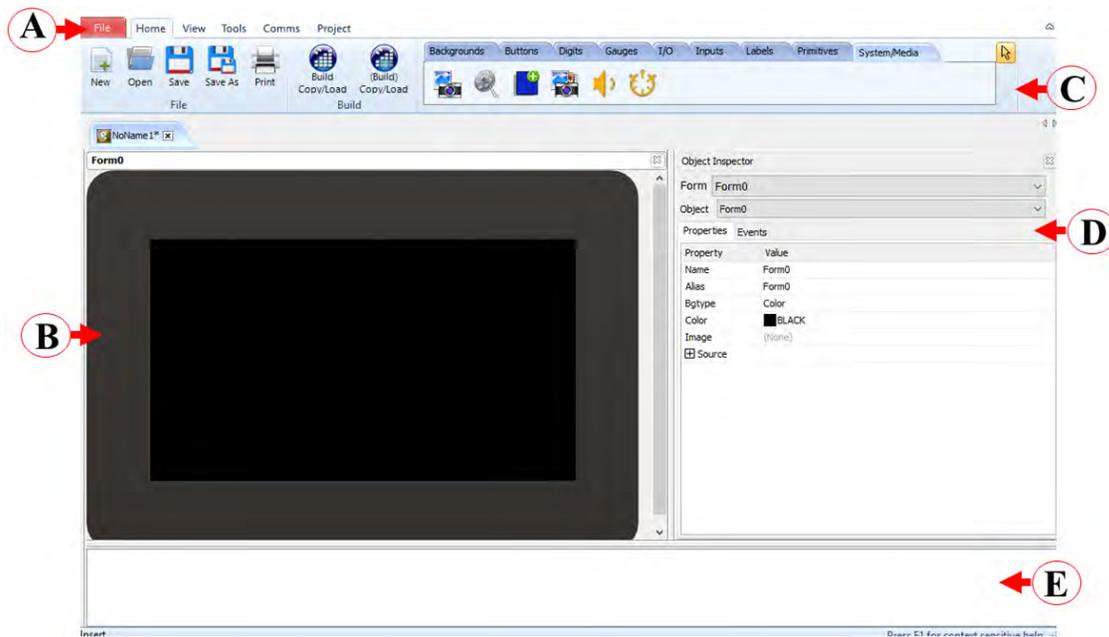


Figura 4.27: Entorno de programación de Visi-Genie.

El entorno de programación de Visi-Genie (Fig. 4.27) consta de diferentes paneles de control, descritos a continuación:

- Menús para acceder a funciones específicas (explicadas más adelante) para la programación de la pantalla.
- Vista previa del GUI en la pantalla. Del mismo modo corresponde al área de trabajo donde los objetos que requiera el diseño se arrastran y posicionan en la ubicación deseada.
- Pestañas de navegación de los tipos de objetos disponibles para agregar en el GUI.
- Object Inspector, permite modificar el aspecto de los objetos en la pantalla, establecer conexiones entre ellos y darles una tarea determinada.
- Consola del programa, una vez terminado el diseño del GUI de la pantalla, los detalles de la compilación se desplegarán en esta sección.

Para el prototipo se desarrollaron dos GUI de pantalla o “Form” como se denomina en el software, *Form0* corresponde a la pantalla de inicio y *Form1* a la pantalla de medición, ambas descritas en el capítulo I. Es importante aclarar que ViSi-Genie posee básicamente en tres grupos de objetos:

- Objetos de entrada: producen datos o estímulos para los objetos de tipo salida o pueden enviar datos en serie para accionar un evento. Ejemplo: un botón.
- Objetos de salida: solo reaccionan a los estímulos de entrada. Se emplean para desplegar valores. Ejemplo: un medidor.

- **Objetos combinados:** son objetos que necesitan tanto estímulos de entrada así como producir un evento de salida. Por ejemplo, una posición del control deslizante.

A continuación se describen los objetos que componen el GUI de la pantalla de inicio:

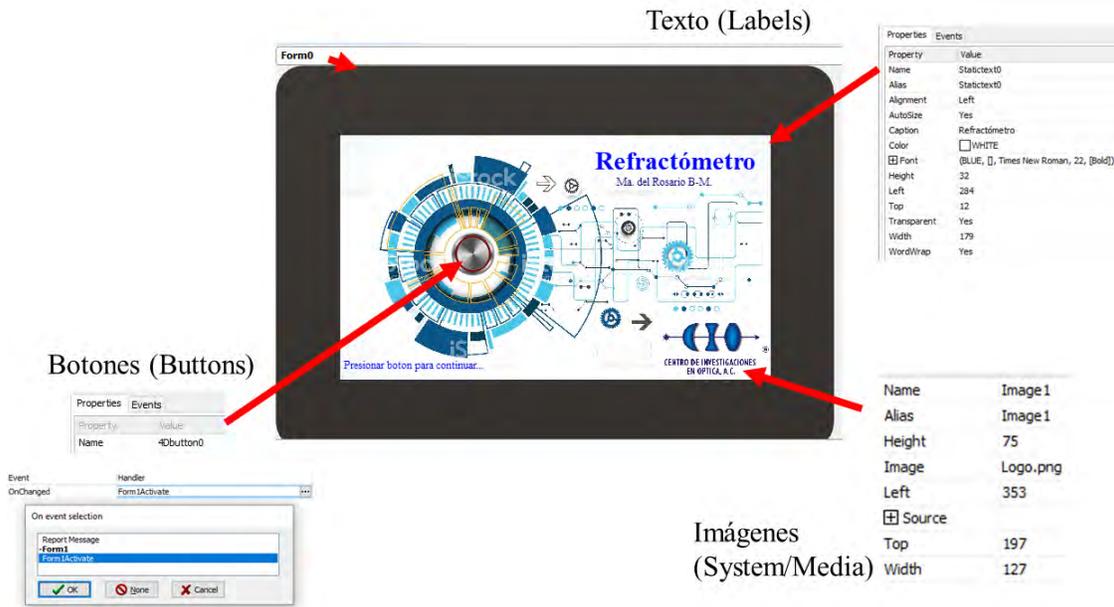


Figura 4.28: Aspecto y objetos que componen la pantalla de inicio del prototipo.

Para programar esta pantalla, se ocuparon elementos sencillos, como botones, imágenes y texto (Fig. 4.28). Cada objeto se modifica en el *Object Inspector* donde se le dan las propiedades requeridas para del diseño. A cada uno de estos se le asigna un nombre que sirve como puntero o indicador. En esta pantalla tenemos:

Botón Inicio: esta clase de objetos se encuentran en la pestaña de navegación llamada *Buttons* (Fig. 4.27(C)). Dentro de ella tenemos una gamma de diferentes botones que se pueden emplear. El aspecto de los botones puede ser modificado en cuanto tamaño y color. En la pestaña *Event* del *Object Inspector* se tienen dos opciones conocidas como: “*OnChanged*” y “*OnChanging*”.

Para el evento *OnChanged*, el botón enviará un valor hasta que el lápiz óptico o el dedo que lo presiona se levanten de la pantalla. La selección del evento *OnChanging*, por otro lado, hace que se envíen valores mientras un objeto se está presionando (el dedo o el lápiz en movimiento aún no se despegan), esta clase de eventos se usa principalmente en slides, barras de posicionamiento o potenciómetros. Para el prototipo ya que los botones actúan como indicadores para poder acceder o ejecutar diferentes tareas solo se usa el evento *OnChanged*. En la Fig. 4.28 el *Handler* del botón tiene seleccionado *Form1Active*; esto significa que al presionar y soltar el botón, la pantalla de inicio cambiara a la pantalla de medición.

Imágenes: se encuentran dentro de la pestaña de *System/Media*. La imagen seleccionada para desplegar puede ser un archivo del usuario, en este caso se colocó el logo del CIO y el

background del GUI. Las principales modificaciones son en cuanto tamaño y posición.

Cabe destacar que la capacidad de la pantalla permite, por medio de esta opción multimedia, anexar archivos de audio y video si así se requiere.

Texto: dentro de la pestaña *Labels*, la principal modificación para este es en cuanto a la posición y al aspecto. Cuenta con toda la gama de tipo de letras que posee Windows, y una amplia variedad en tamaños y colores. Esto brinda la posibilidad de personalizar el prototipo.

La pantalla de inicio tiene la función de dar una presentación del sistema al usuario, para acceder posteriormente a la pantalla de medición, la cuál, está conformada por los elementos que a continuación se describen.

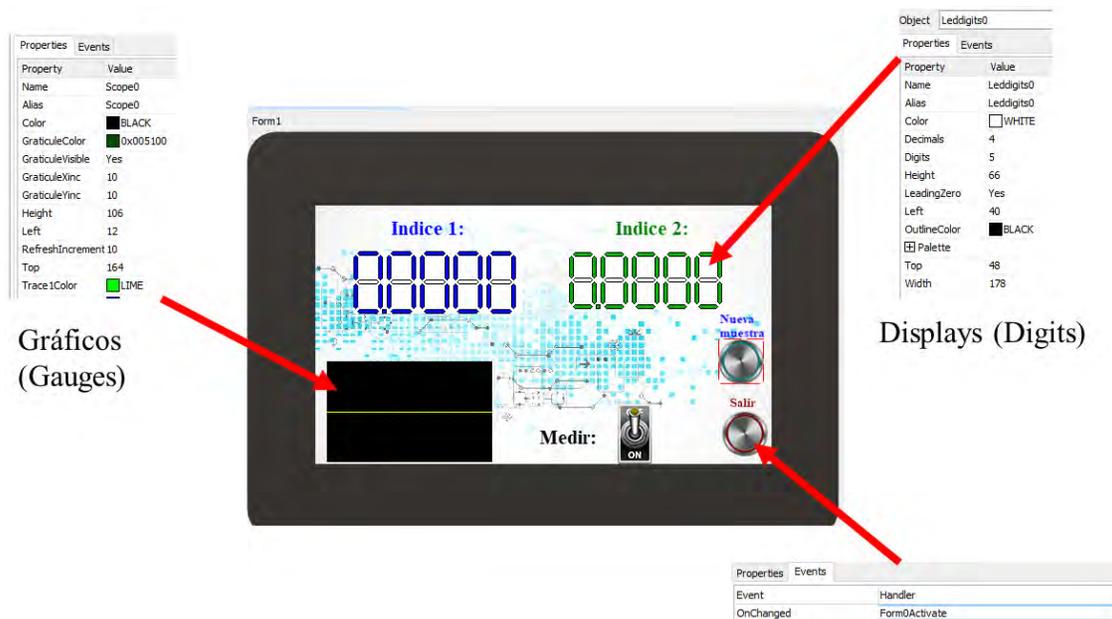


Figura 4.29: Pantalla de medición.

Se agregan nuevos objetos tales como:

Gráficos: se pueden encontrar dentro de la pestaña de *Gauges*. Además de poder realizar las modificaciones básicas de apariencia y posición también permite seleccionar el tipo de trazo, ya sea proveniente del host o desplegar la lectura de datos desde un archivo multimedia.

Displays: los LCD de 7 segmentos se encuentran en la pestaña de *Digits*. Son utilizados para desplegar los valores calculados de IR de la muestra. Es posible definir el número de dígitos a desplegar y dividirlo en enteros y decimales. De acuerdo a la resolución del sistema y rangos posibles de medición de IR, se estableció un entero y 4 decimales (Fig. 4.29). Para el prototipo simplemente se exhibe un valor cuando este se solicita, sin embargo, también puede actuar como un event tipo *OnChangeing* y mostrar los valores de forma dinámica.

Existen diferentes tipos de botones para seleccionar dentro de la pestaña de *Buttons*, para el prototipo se eligieron dos tipos de botones. Todos los botones se pueden modificar de la misma forma.

En la Fig. 4.29 se observa como el botón Salir, en la pestaña de *Event*, el *Handler* se encuentra direccionado a la pantalla de inicio “*Form0Active*”, al igual que el caso anterior esto sirve para realizar el cambio de pantalla al finalizar el proceso.

4.2.4. Compilar y cargar el programa

Una vez diseñado y establecido todas las funciones de los objetos del GUI de la pantalla se procede a compilar y cargar el programa en ésta.

Antes de realizar la compilación y carga del programa en la pantalla se debe considerar los siguientes aspectos:

Se necesita contar con el cable de programación 4D o un adaptador micro - USB-PA5, en este caso se utilizó el cable (se consigue dentro del catálogo de 4D System). Instalar previamente los controladores a la PC para poder usar el Cable de programación 4D o el adaptador [49].



Figura 4.30: Cable de programación de la compañía 4D System, con las terminales de +5V, Rx, Tx, GND y RES para conectar al módulo de programación de la pantalla.

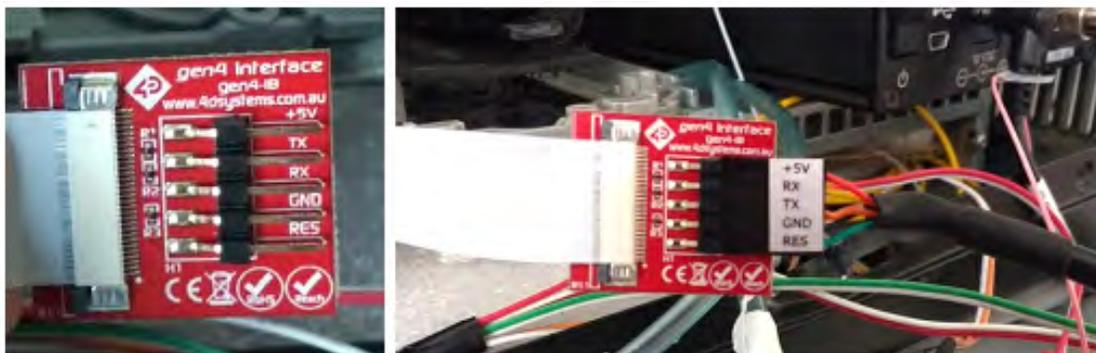


Figura 4.31: Conexión del módulo de programación de la pantalla con el cable 4D. Los pines señalados en ambos elementos deben coincidir. El módulo y cable plano vienen incluidos con la pantalla.

Se procede con la conexión del cable a la PC. Para comprobar si se detecta la pantalla, en el menú de *Comms*, se visualiza el estado de la conexión de la pantalla con respecto a la PC.

Existe un selector para indicar el puerto COM en el que la pantalla se encuentra conectada, a un costado la descripción del estado, y debajo un led indicador. La luz violeta indica que no hay ningún módulo de programación conectado. La luz se vuelve amarilla mientras se establece la conexión. La luz se pone azul cuando se establece correctamente la conexión. Finalmente cuando no se puede detectar la pantalla la luz se vuelve roja.

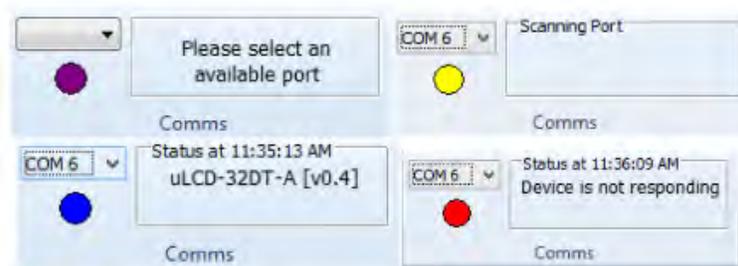


Figura 4.32: *Diferentes estados posibles entre la conexión de la pantalla con la PC.*

Para el tipo de procesador utilizado en esta pantalla, Diablo16, todo el contenido multimedia generado para el GUI se carga en una memoria micro - SD que debe tener el formato FAT16. La tarjeta de memoria además no puede exceder los 4 GB. Se utilizó una memoria 8 GB, por lo que fue necesario crear una partición para dar el formato necesario a la memoria.



Figura 4.33: *Tarjeta de memoria micro - SD, para cargar el contenido multimedia del programa.*

Después de asegurarse de que se detecta la pantalla correctamente, en el menú *Home* hacer clic en el botón *Build Copy/Load*. Workshop4IDE compilará y descargará el programa en la pantalla. Posteriormente se solicitará la tarjeta micro - SD para almacenar el contenido multimedia. Al seleccionar la tarjeta de memoria y dar clic en *Ok*; se mostrará una barra de progreso mientras se copian los archivos necesarios en la tarjeta micro - SD.

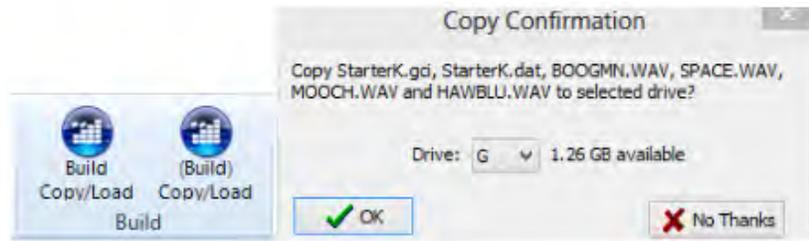


Figura 4.34: Botón de compilar y cargar programa. Selector de tarjeta de memoria micro - SD.

Un detalle importante, como se observa en la Fig. 4.34, es que existen dos botones para cargar el programa. El botón derecho crea los archivos gráficos sólo cuando Workshop detecta cualquier cambio realizado en la pantalla y/o en el *Object Inspector*. El botón izquierdo obliga a guardar los archivos gráficos todo el tiempo. Este botón es la mejor opción cuando se requiere asegurar que los archivos gráficos se actualicen todo el tiempo.

Workshop copia dos archivos en la tarjeta micro - SD: los archivos GCI y DAT. El archivo GCI contiene los gráficos y el archivo DAT contiene una lista de los objetos dentro del archivo GCI. El programa accederá a estos archivos cuando la pantalla esté encendida.

```

0 errors
0 warnings
0 notices
No Errors, code size = 217 bytes out of 9216 total
Initial RAM size = 0 bytes out of 510 total
Download successful.

```

Figura 4.35: Consola del entorno. Compilación y carga del programa completo y correcto.

Finalmente se desconecta la tarjeta de memoria micro - SD de la PC y conéctela a la ranura para tarjeta micro - SD de la pantalla. El proyecto ahora se ejecuta en la pantalla.



Figura 4.36: Colocación de la tarjeta micro - SD en la ranura del módulo de control de la pantalla.



Figura 4.37: Pantalla en funcionamiento con el programa desarrollado.

4.2.5. Comunicación entre la pantalla y la tarjeta de desarrollo: Protocolos de comunicación de ViSi - Genie

La plataforma de visualización ViSi-Genie ofrece un protocolo de comunicación en serie llamado *Genie Standard Protocol*. Éste proporciona acceso a la mayoría de las características de la pantalla y le da al host información detallada sobre el estado de todos los objetos utilizados.

El *Genie Standard Protocol* proporciona una interfaz simple pero efectiva entre la pantalla y el controlador del *host*. Todas las comunicaciones tienen un enlace bidireccional [50].

La configuración de datos serie es:

8 bits, sin paridad, 1 bit de paro.

La velocidad en baudios para el prototipo fue de 9600. Solo se utilizan las terminales de Rx/D (datos recibidos), Tx/D (datos transmitidos) y tierra de la señal para realizar la comunicación. Los comandos y parámetros que se envían y reciben usan una estructura de mensajería muy simple. El mensaje consiste de un byte de comando, parámetros de comando y un byte de suma de verificación. La siguiente figura muestra la organización del mensaje:

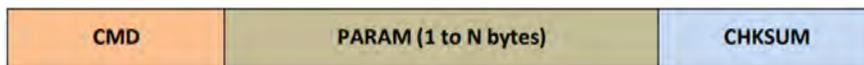


Figura 4.38: Estructura del mensaje para la comunicación serial.

- CMD: este byte indica el código de comando.

- PARAM: bytes de parámetros, contienen información perteneciente al objeto.
- CHKSUM: byte de suma de verificación; este byte se calcula tomando cada byte y realizando la operación XOR entre todos los bytes en el mensaje desde el byte CMD hasta el último byte de parámetro.

Workshop4 posee una herramienta llamada *Genie Test Executor* o GTX que nos permite visualizar los mensajes enviados y recibidos por los objetos de la pantalla. Para iniciar el depurador, nos dirigimos al menú de *Tools*, y damos clic en el botón GTX.

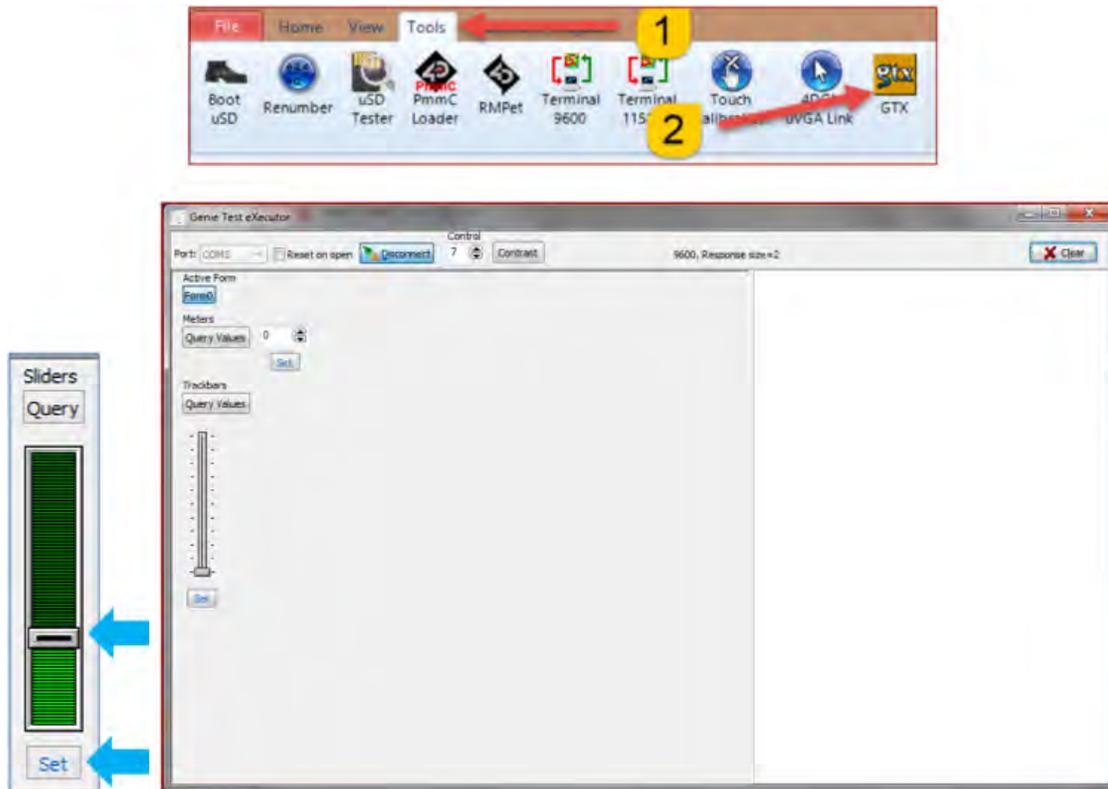


Figura 4.39: Aspecto de la ventana de la herramienta GTX.

Aparece una nueva pantalla, con los objetos que se colocaron previamente en el GUI de la pantalla, en este ejemplo se utiliza un *TrackBar*, para visualizar los cambios en los comandos. Al mover el puntero del *TrackBar* se visualizarán un conjunto de valores en hexadecimal como el siguiente:

```
Set Slider Value 08:54:50.541 [01 04 00 00 1F 1A]
ACK 08:54:50.574 [06]
```

- En verde, los mensajes enviados a la pantalla
- Y en rojo los mensajes recibidos de la pantalla:

Los bytes de mensaje reales son aquellos dentro de los corchetes.



El mensaje enviado se codifica de acuerdo con el siguiente patrón:

Command	Object Type	Object Index	Value MSB	Value LSB	Checksum
01	04	00	00	1F	1A
WRITE_OBJ	Slider	First	0x001F		

Traduciendo el mensaje significa: “Escribir en el primer objeto deslizando en la pantalla el valor 0x001F”. Al convertir el valor hexadecimal 0x001F a decimal se tiene el valor 31.

La suma de comprobación (*Checksum*) es un medio para que el host verifique si el mensaje recibido es correcto. Se agrega al final y si el mensaje es correcto, todos los bytes (incluido el byte de *Checksum*) dará un resultado de cero. Comprobando la integridad de un mensaje.

Antes de continuar hay que mencionar que para obtener todos los comandos necesarios para identificar cada uno de los objetos dentro del GUI, previamente en el *Object Inspector*, dentro del evento *Onchanged* se seleccionó el *Handler* de *Report Message*, esto con la finalidad de usar la herramienta de GTX. Y así se registraron todos los valores en hexadecimal.

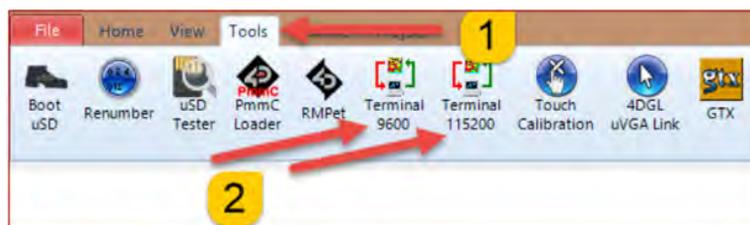
En la Fig. 4.40 se observan los diferentes comando en hexadecimal para varios tipos de objetos disponibles en WorkShop4IDE.

Object	ID
Dipswitch	0 (0x00)
Knob	1 (0x01)
Rockerswitch	2 (0x02)
Rotaryswitch	3 (0x03)
Slider	4 (0x04)
Trackbar	5 (0x05)
Winbutton	6 (0x06)
Angularmeter	7 (0x07)
Coolgauge	8 (0x08)
Customdigits	9 (0x09)
Form	10 (0x0A)
Gauge	11 (0x0B)
Image	12 (0x0C)
Keyboard	13 (0x0D)
Led	14 (0x0E)
Leddigits	15 (0x0F)

Meter	16 (0x10)
Strings	17 (0x11)
Thermometer	18 (0x12)
Userled	19 (0x13)
Video	20 (0x14)
Statictext	21 (0x15)
Sound	22 (0x16)
Timer	23 (0x17)
Spectrum	24 (0x18)
Scope	25 (0x19)
Tank	26 (0x1A)
Userimages	27 (0x1B)
PinOutput	28 (0x1C)
PinInput	29 (0x1D)
4Dbutton	30 (0x1E)
AniButton	31 (0x1F)
ColorPicker	32 (0x20)
UserButton	33 (0x21)

Figura 4.40: Comandos para diferentes objetos disponibles para la programación de la pantalla [51].

Una alternativa a la herramienta de GTX, es utilizar las terminales en el menú de *Tools*.



Se tienen 2 opciones:

- Terminal a 9600
- Terminal a 115200

Se abre una nueva ventana en la que se despliegan los valores en hexadecimal de los comandos correspondientes a los objetos que se encuentran siendo presionados en ese momento.

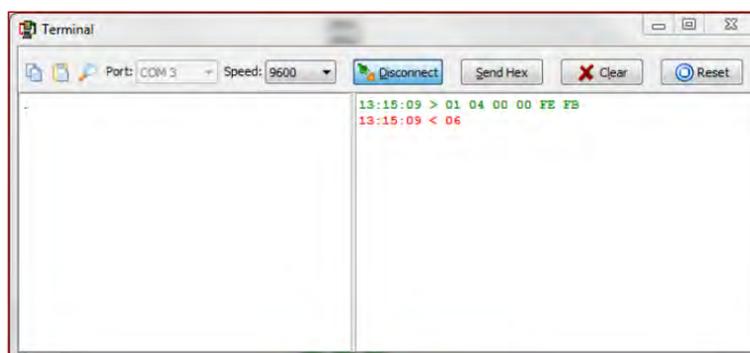


Figura 4.41: Ventana de la terminal a 9600 baudios.

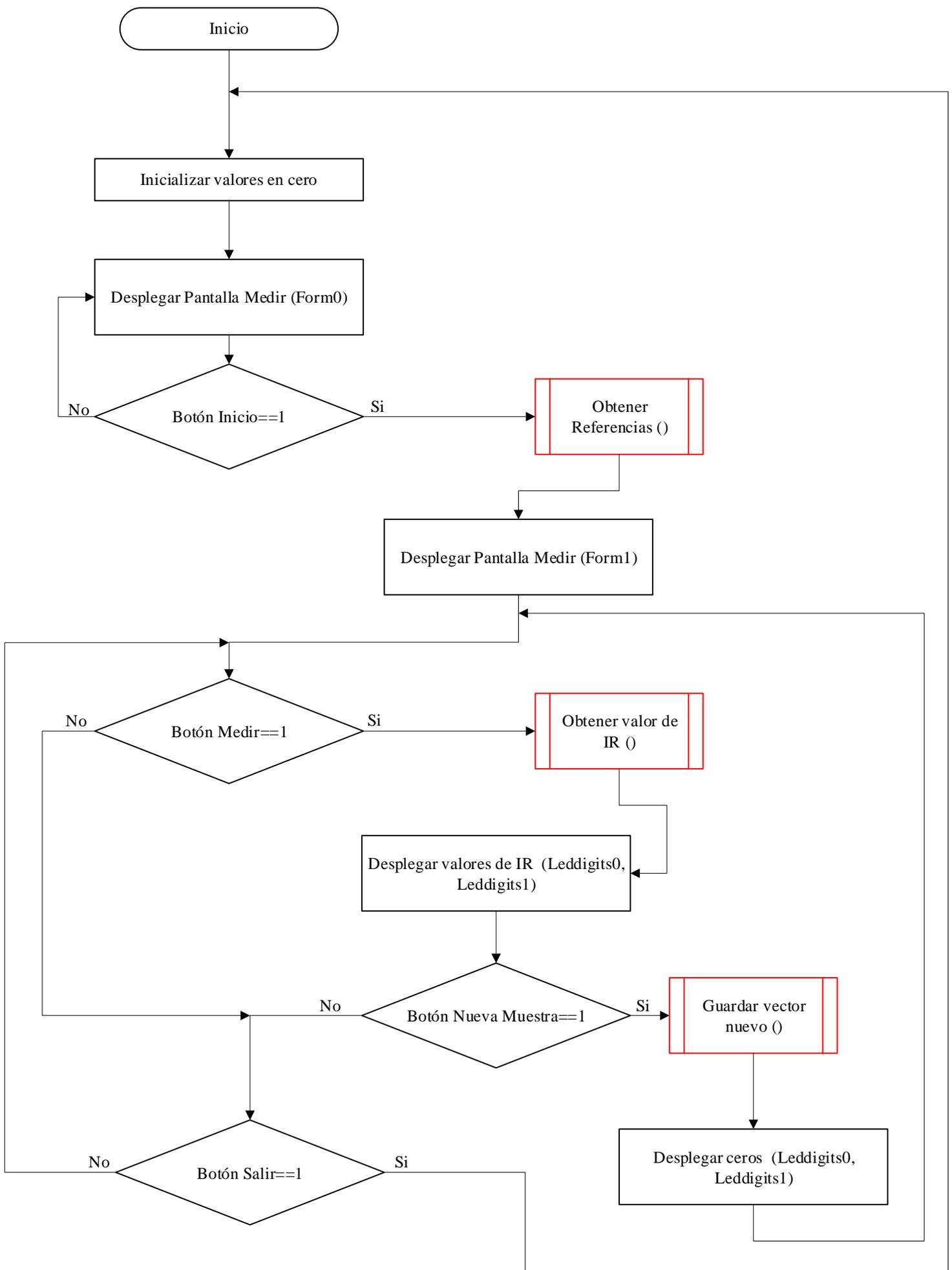
Los comandos enviados por el host y los mensajes enviados por la pantalla son los mismos que con el *Genie Test Executor*. De la misma forma tenemos:

- En verde, los mensajes enviados a la pantalla
- Y en rojo los mensajes recibidos de la pantalla

En este caso la pantalla responde con el acuse de recibo 0x06, que se muestra en rojo en la ventana de la terminal.

A diferencia de la terminal GTX, esta herramienta solo despliega los comandos recibidos y no hay una manipulación bidireccional. Sin embargo, es una herramienta útil para checar el correcto funcionamiento de la comunicación.

A continuación se muestra un diagrama de flujo de las funciones que desempeña la pantalla, y su relación con la tarjeta de desarrollo:



Manipulación de la pantalla con la tarjeta de desarrollo

Para lograr la comunicación entre la tarjeta de desarrollo y la pantalla, los parámetros de la tarjeta deben ser configurados de modo que sea capaz de leer los comandos que la pantalla mandara al presionar cualquier objeto del GUI.

En el software de STM32CubeMX accedemos al pinout y activamos los puertos para la comunicación serial USART y se habilita el modo asíncrono (Fig. 4.42).



Figura 4.42: Habilitación del USART6 y ubicación de las terminales de conexión.

Los pines a utilizar para la conexión son PC6 y PC7 del conector CN7 de la tarjeta. Gracias a que la tarjeta posee una terminal de 5V es posible alimentar y comunicarse con la pantalla sin la necesidad de agregar circuitos de acoplamiento. A continuación se muestra la conexión requerida:

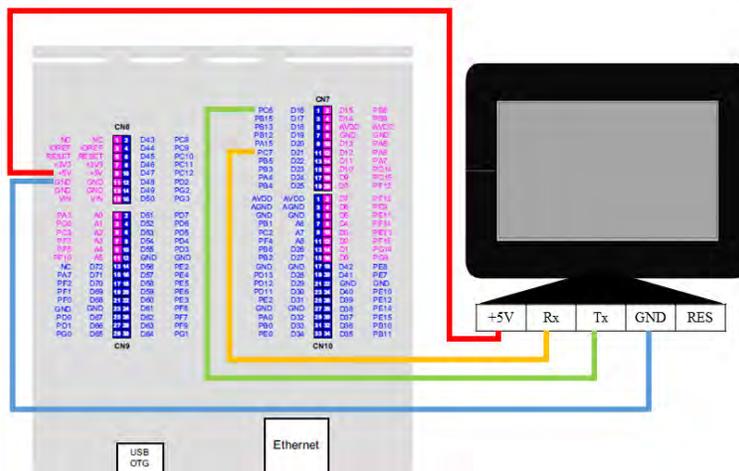


Figura 4.43: Conexión entre la tarjeta de desarrollo y la pantalla.

Es importante establecer los parámetros básicos para la comunicación serial, en la Fig. 4.44 se muestran los valores fijos para que la pantalla y la tarjeta estén sincronizadas.

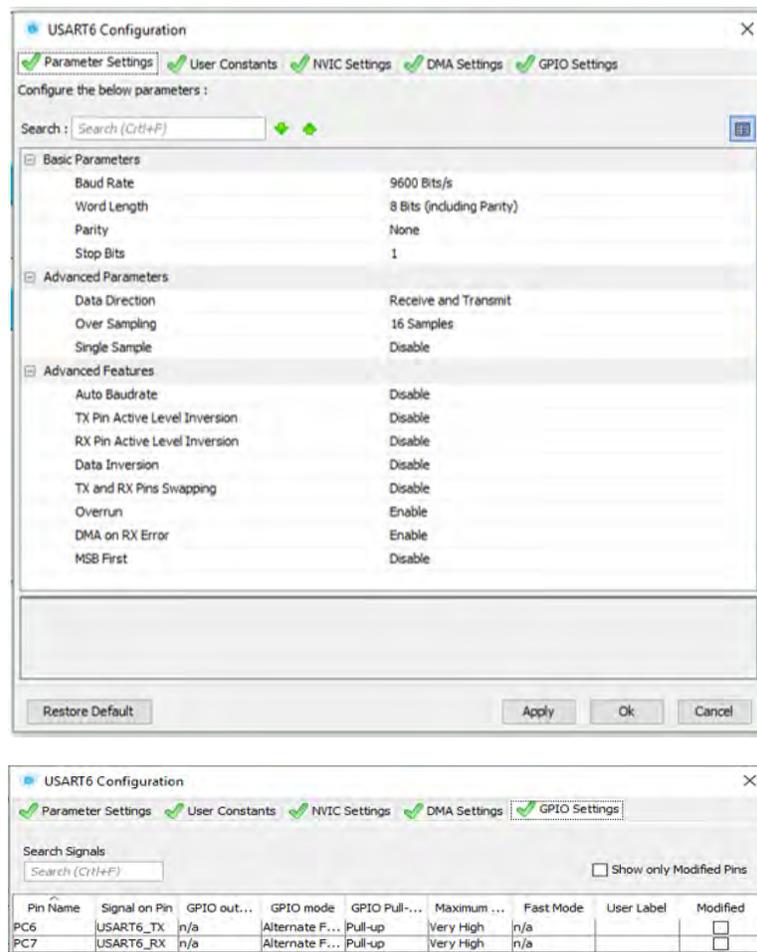


Figura 4.44: *Parámetros establecidos en la opción de configuración del STM32CubeMx para la comunicación serial.*

Al momento de compilar el STM32CubeMx, este generará las líneas de código dentro de main() de Atollic:

```

/* USART6 init function */
static void MX_USART6_UART_Init(void)
{
    huart6.Instance = USART6;
    huart6.Init.BaudRate = 9600;
    huart6.Init.WordLength = UART_WORDLENGTH_8B;
    huart6.Init.StopBits = UART_STOPBITS_1;
    huart6.Init.Parity = UART_PARITY_NONE;
    huart6.Init.Mode = UART_MODE_TX_RX;
    huart6.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart6.Init.OverSampling = UART_OVERSAMPLING_16;
    huart6.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
    huart6.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
    if (HAL_UART_Init(&huart6) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}

```

donde especificará lo parámetros previamente establecidos, agrega los archivos y/o librerías, dentro de la carpeta de proyecto, necesarias para ejecutar los comandos correspondientes a la comunicación serial.

Como se mencionó anteriormente, se guardaron los comandos correspondientes a los objetos dentro del GUI de la pantalla, para poder seleccionarlos se crean las variables correspondientes dentro del programa de la tarjeta:

```

167 int main(void)
168 {
169
170     /* USER CODE BEGIN 1 */
171     t8_t cdn[8]="John :)";
172     uint8_t varc=0;
173     uint8_t lcd1[6]={0x01,0x0F,0x00,0x33,0x6C,0x51};
174     uint8_t lcd2[6]={0x01,0x0F,0x01,0x38,0x6E,0x59};
175     uint8_t medir[6]={0x07,0x1E,0x02,0x00,0x00,0x1B};
176     //uint8_t nuevaM[6]={0x07,0x1E,0x03,0x00,0x00,0x1A};
177     uint8_t boton[6]={0x00,0x00,0x00,0x00,0x00,0x00};
178     uint8_t lcd0[6]={0x01,0x0F,0x00,0x00,0x00,0x0F};
179     uint16_t indice0=15000;
180     uint8_t checksum=0;

```

Se definen todos los elementos con lo que la tarjeta y la pantalla van a interactuar. En este caso es los displays de 7 segmentos nombrados como lcd1 y lcd2, los botones de medir, nueva muestras y salir.

La pantalla será quien manipule las acciones que realizara la tarjeta. A continuación se muestra el programa que ejemplifica como es que la tarjeta identifica cual es el botón presionado y hacia que objeto salida mandar un dato.

```

while(boton[0]!=medir[0]&& boton[1]!=medir[1]);
HAL_UART_Receive(&huart6, boton, 6, 250);
}
if(boton[2]==0x02){
    HAL_GPIO_WritePin(GPIOB, LD2_Pin, 1);
    obtenerV();
    transformada();
    HAL_GPIO_WritePin(GPIOB, LD2_Pin, 0);
    lcd1[3]=indice0/258;
    lcd1[4]=indice0&0xFF;
    for(int k=0; k<5; k++){
        if(k==0) cheksum=lcd1[0];
        else cheksum^=lcd1[k];
    }
    lcd1[5]=cheksum;
    HAL_UART_Transmit(&huart6, lcd1, 6, 250);
    varc=0;
    while(varc!=0x06){
        HAL_UART_Receive(&huart6, &varc, 1, 250);
    }
    HAL_UART_Transmit(&huart6, lcd2, 6, 250);
    varc=0;
    while(varc!=0x06){
        HAL_UART_Receive(&huart6, &varc, 1, 250);
    }
    indice0++;
}
else if(boton[2]==0x03){
    lcd0[2]=0x00;
    lcd0[5]=0x0E;
    HAL_UART_Transmit(&huart6, lcd0, 6, 250);
    varc=0;
    while(varc!=0x06){
        HAL_UART_Receive(&huart6, &varc, 1, 250);
    }
}
for(int i=0; i<6; i++){
    boton[i]=0x00;
}

```

4.2.6. Conclusión

Con las pruebas y el algoritmo desarrollado es posible manipular cualquier pantalla de 4DSytem, usando como host la tarjeta de desarrollo STM32F767. Lo mas importante es determinar los parámetros necesarios para la sincronización entre dispositivos, e identificar como se codifican los mensajes enviados por los objetos dentro del GUI.

WorkShop4ID es una herramienta de programación muy completa y permite realizar interfaces humano-máquina agradables al usuario. Explorar todas las funciones que brindan permite realizar proyectos mas complejos y crear GUI mas dinámicas.

4.3. Diseño y construcción del sistema mecánico

En esta sección se presenta el diseño de los elementos mecánicos que integran al prototipo: mecanismo de posicionamiento de muestras, sistema porta-muestra y armazón del sistema. Considerando los requisitos del sistema, se seleccionan materiales para su construcción. Finalmente se muestra el armado del sistema y la ubicación de componentes dentro del armazón.

4.3.1. Sistema de posicionamiento

Para la colocación de las muestras líquidas, se utiliza un sistema de posicionamiento que realiza dos recorridos base “**Posicionar**” y “**Retirar**”, ilustrados en color rojo y verde respectivamente en la figura 4.45.

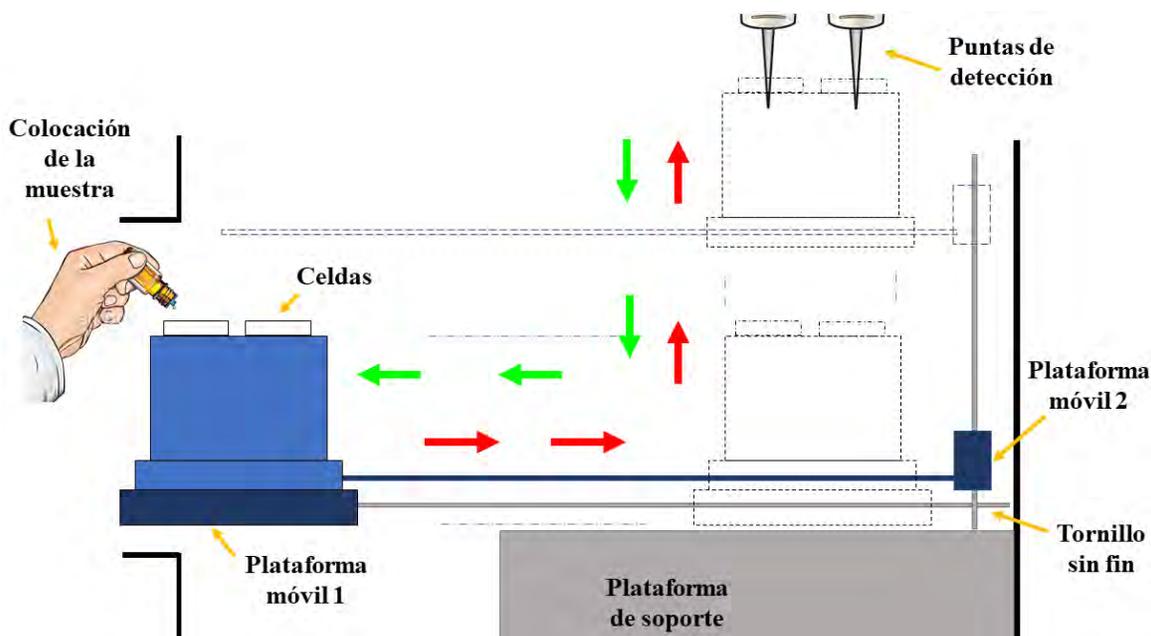


Figura 4.45: Vista lateral del sistema de posicionamiento señalando el desplazamiento de las muestras hacia los puntos de detección. Recorridos “**Posicionar**” y “**Retirar**” (rojo y verde) de la muestra.

Para completar ambos recorridos se tienen dos estructuras de desplazamiento. La estructura de desplazamiento horizontal realiza los movimientos de “**Salir**” y “**Entrar**”. La estructura de desplazamiento vertical que realiza los movimientos de “**Subir**” y “**Bajar**”.

4.3.2. Estructura de desplazamiento horizontal

Esta estructura tiene la finalidad de desplazar el porta-muestra hacia fuera y hacia dentro del equipo. Realiza los movimientos de Salir y Entrar, explicados más adelante en el algoritmo de control. La figura 4.46 describe los elementos que conforman la estructura.

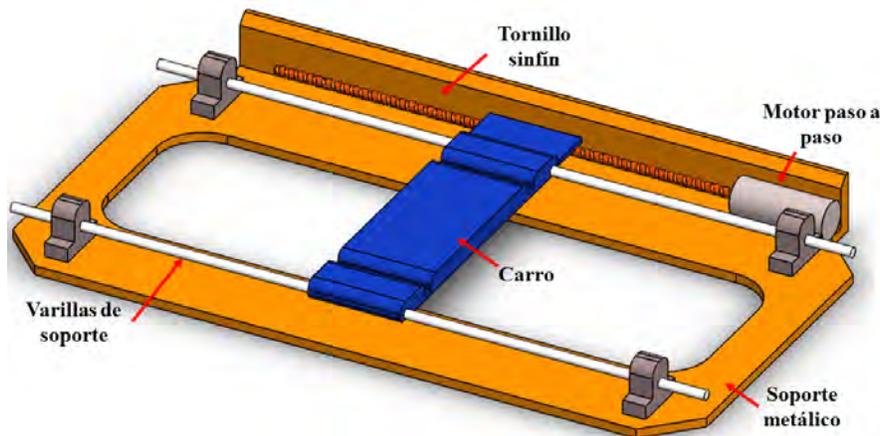


Figura 4.46: Estructura de desplazamiento horizontal. Posee un soporte metálico fijo que sostiene dos varillas que funcionan como guías para el carro móvil. Se emplea un motor a pasos para trasladar el carro móvil donde se colocará el sistema porta-muestra.

Un motor paso a paso bipolar se encarga de desplazar el carro móvil. Para realizar dicha acción el carro móvil está unido al eje del motor, compuesto por un tornillo sin fin. El sentido de giro del eje determinará hacia qué dirección avanza el carro. El carro es una pieza de fundición ligera, en donde se coloca la base del sistema porta-muestra.

Los motores paso a paso bipolares tienen generalmente cuatro cables de salida y necesitan ciertas manipulaciones para ser controlados (figura 4.47). Requieren del cambio de dirección del flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento. Además, es necesario un puente H por cada bobina del motor.

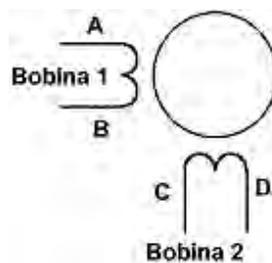


Figura 4.47: Representación y nombre de las terminales del motor a pasos.

La Tabla 4.1 muestra la secuencia implementada para controlar el motor. Dependiendo de la acción a realizar (Salir o Entrar), es la señal enviada a las terminales del motor.

Un punto importante que considerar sobre esta clase de motores es que al ser dispositivos mecánicos deben vencer ciertas inercias, hay que tener en cuenta el tiempo de duración y la frecuencia de los pulsos aplicados. Por esta razón se tiene una espera de 5 ms entre pasos y una de 100ms antes de enviar la siguiente secuencia de pasos. Esto para evitar cualquier movimiento indeseado, vibraciones, girar erráticamente o en sentidos opuestos.

Tabla 4.1: *Secuencia implementada para el desplazamiento del motor*

	Entrar				Salir			
	A	B	C	D	A	B	C	D
1	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0
3	0	0	1	0	0	1	0	0
4	0	0	0	1	1	0	0	0

El peso máximo soportado por la plataforma es de 300 gr, consumiendo una corriente de 500 mA.

4.3.3. Estructura de desplazamiento vertical

Los desplazamientos en vertical “Subir” y “Bajar” se realizan con un motor DC y un puente H para el sentido de giro. El giro del motor DC depende de los niveles de voltaje que existan en las terminales del mismo.

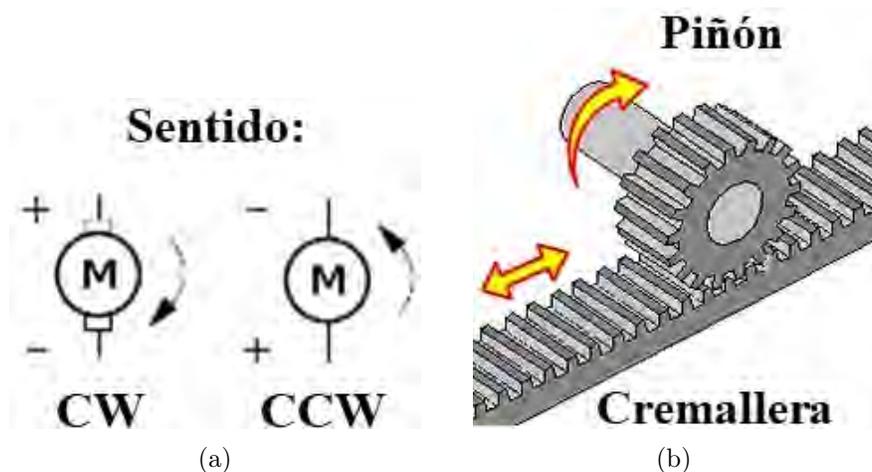


Figura 4.48: a) Sentido de giro en el motor. b) Sistema cremallera-piñón.

Para elevar la plataforma se usa un sistema cremallera-piñón (Fig. 4.48(b)). El piñón es una rueda dentada que describe un movimiento de rotación alrededor de su eje; éste se encuentra unido al eje del motor DC. La cremallera es una pieza dentada que se desplaza con movimiento rectilíneo en uno u otro sentido según la rotación del piñón, se encuentra unida al carro de desplazamiento que eleva la viga que sostiene la muestra.

Para el control de velocidad del motor DC se emplea un voltaje fijo para mantener el avance del motor constante y adecuado para el manejo de las muestras. El potencial que se le suministra

es de 5V. Llega a la posición inicial y final por medio de una rutina de tiempo, y ejecutando un paro de motor en sus terminales para mantener su posición.

La Fig. 4.49 describe los elementos que conforman esta estructura. La unión de ambas estructuras ayuda a realizar los movimientos conjuntos para el posicionamiento de la muestra (Fig. 4.50).

El sistema porta-muestra consta de 3 elementos. Una base unida al carro móvil de la estructura horizontal. Una viga unida al carro móvil de la estructura vertical. Y el porta-muestra que como su nombre lo indica sostiene las celdas donde se introducen las muestras líquidas a ser medidas. En la Fig. 4.51 se describen cada uno de los elementos.

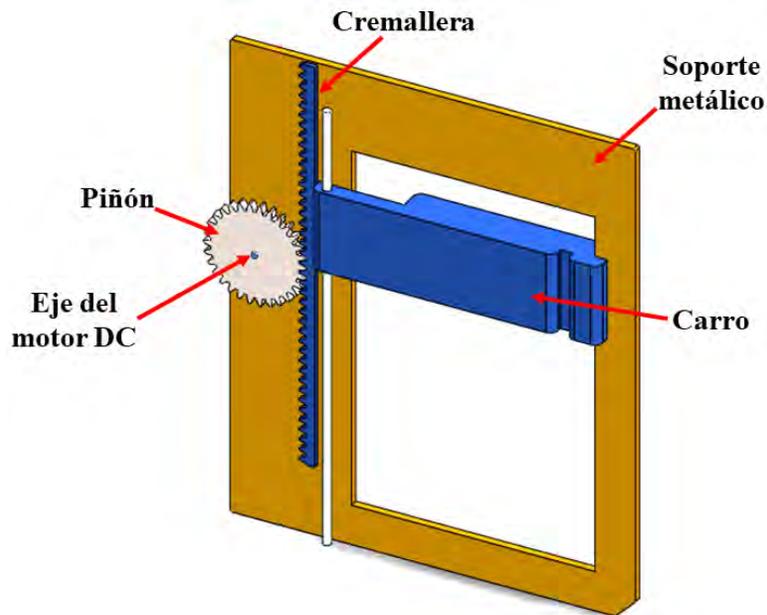


Figura 4.49: Estructura de desplazamiento vertical. Posee un soporte metálico fijo que sostiene la varilla guía para el carro móvil. Se emplea un motor de DC para trasladar el carro móvil.

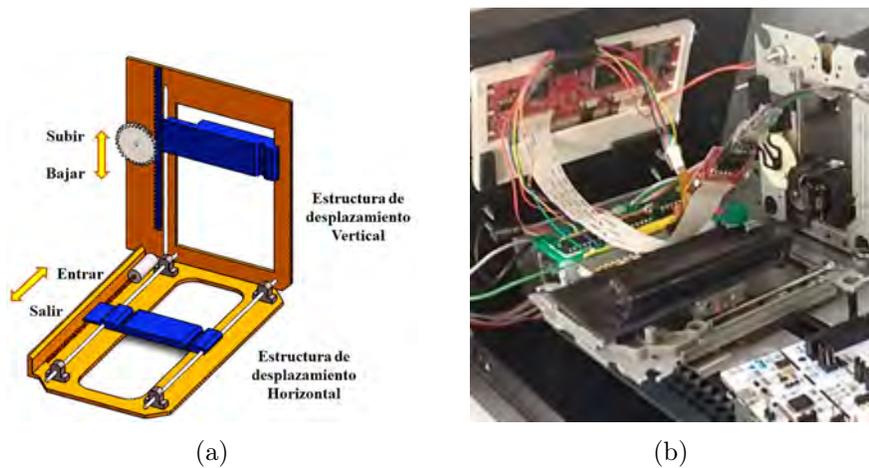


Figura 4.50: a) Estructuras de posicionamiento unidas con la descripción de los movimientos realizados por cada una. b) Sistema de posicionamiento armado.

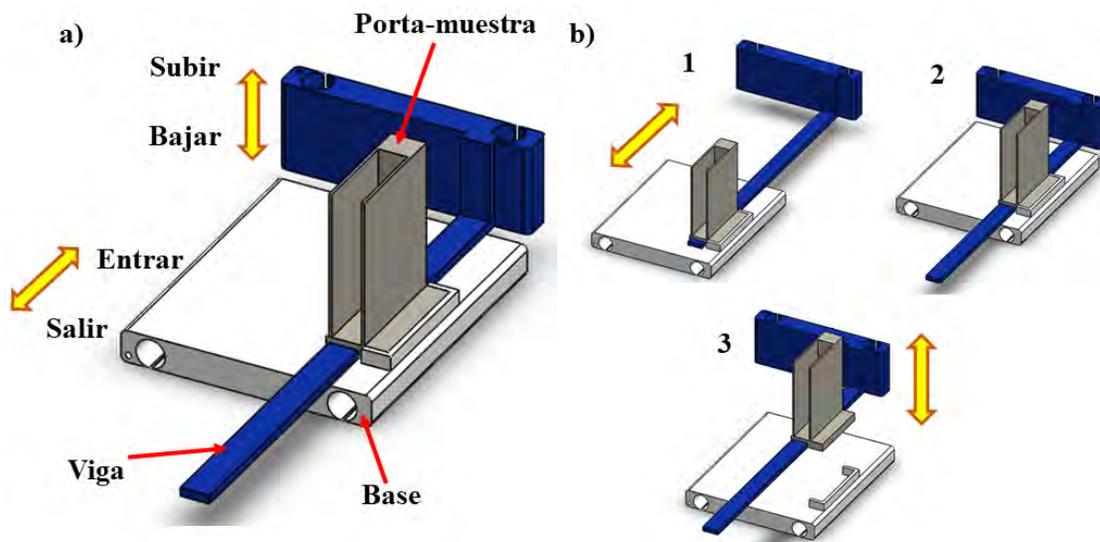


Figura 4.51: a) Componentes del sistema porta-muestras. b) Desplazamientos del soporte.

4.3.4. Sistema de control para el posicionamiento de las muestras

Como se mencionó anteriormente es necesario la implementación de un puente H para controlar el sentido de giro del motor, así como satisfacer las necesidades de corriente de los mismos.

El puente H utilizado fue el circuito integrado L293D (Fig. 4.52) cuyas entradas son compatibles con TTL, por lo que pueden manejarse con la mayoría de los microcontroladores y circuitos lógicos del mercado. La salida de los drivers pueden activarse y desactivarse en pares, mediante señales de habilitación. Cuando se presenta un 1 lógico en los pines de habilitación, las señales correspondientes de salida estarán activas y en fase con la señal de entrada. Cuando

las entradas de habilitación presentan un estado lógico 0, las salidas se encuentran en estado de alta impedancia.

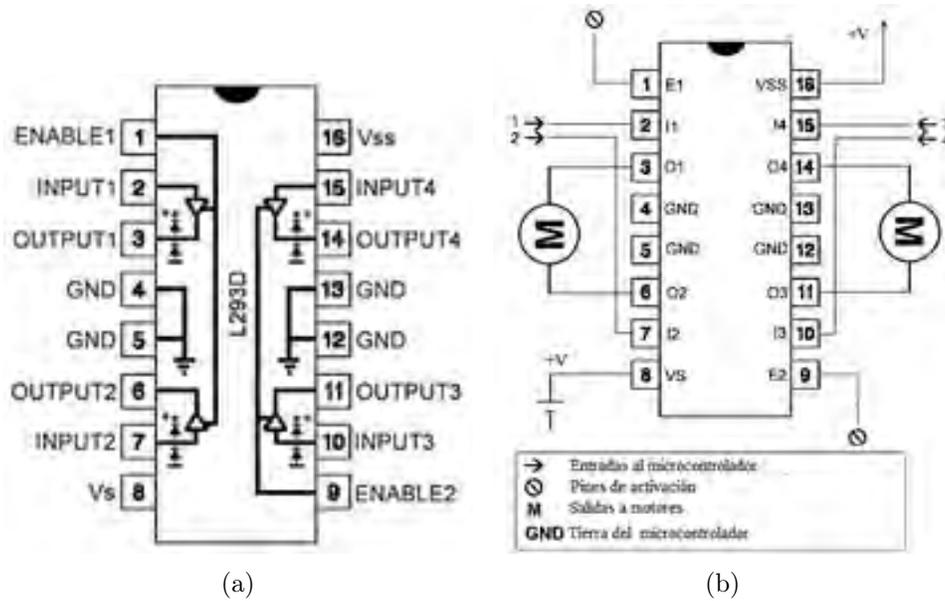


Figura 4.52: a) DIP estándar de 16 pines del L293D. b) Conexión del integrado.

Por ejemplo para controlar el motor izquierdo se siguen los pasos de la Tabla 4.2. Está conectado al Pin3 (O1) y Pin6 (O2). Se necesitan tres pines para controlar este motor: Pin1 (E1), Pin2 (I1) y Pin7 (I2). La tabla 4.2 representa la funcionalidad. Con los pines análogos en el integrado; el control del motor derecho sigue el mismo comportamiento.

Tabla 4.2: Tabla de verdad para manipular el Motor izquierdo de la Figura 4.52(b)

Pin 1	Pin 2	Pin 7	Función
1	1	0	Gira en contra de las manecillas del reloj
1	0	1	Gira en el sentido de las manecillas del reloj
1	1	1	Paro
1	0	0	Paro
0	X	X	Paro

Entre las características principales de este integrado tenemos una corriente de salida de 600 mA hasta 1.2A, alta inmunidad al ruido electrónico, y diodos de protección incorporados.

Como se describió en el apartado anterior los motores requieren ya sea de una secuencia de pulsos en sus bobinas para generar su movimiento o de un control de tiempo para posicionarse adecuadamente. Es por ello por lo que con la finalidad de hacer que el prototipo quede integrado por módulos, y facilitar el acoplamiento de la fuente de alimentación; el control de los motores

se realizó con un PIC16F84A. El utilizar el microcontrolador PIC permite además tener el control de los motores separado de la tarjeta de desarrollo, esto evitará la acumulación de subrutinas en segundo plano, y permitirá tener un estado de reposo en los motores, bajando el consumo de energía.

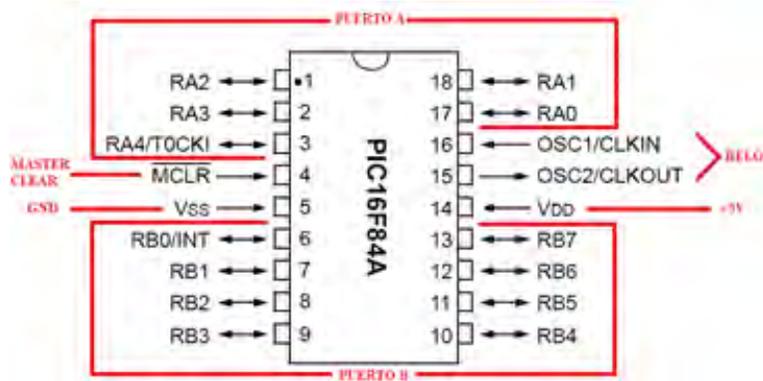


Figura 4.53: DIP estándar de 18 pines del PIC16F84A con distribución de terminales.

El PIC16F84A (Fig 4.53) ayuda a la simplicidad del sistema, se usa ya que es un microcontrolador estándar de gamma media, con una arquitectura de 8 bits, 18 pines, 13 pines en dos puertos, 5 pines el puerto A y 8 pines el puerto B. Utiliza una frecuencia de reloj externa de 4MHz a 20MHz (no posee convertidores analógicos-digital ni digital-analógicos).

Las dos secuencias requeridas para cada motor se programan en mikroC pro for PIC, en lenguaje C, el algoritmo que rige el desempeño de los motores se muestra a continuación:

Inicio

Definir Puerto B como salidas, Puerto A como entradas.
Definir variables i, bp,=0 y br=1 (banderas del sistema).

Si se presiona botón Posicionar y br=1 , entonces:

bp=1, br=0.
Llamar a subrutinas Entrar() y Subir().

Si se presiona botón Retirar y bp=1, entonces:

br=1, bp=0.
Llamar a subrutinas Bajar() y Salir().

Volver a Inicio

Definir funciones de las subrutinas.

A continuación se muestra el diagrama de flujo y el circuito para el control del sistema de posicionamiento de las muestras:

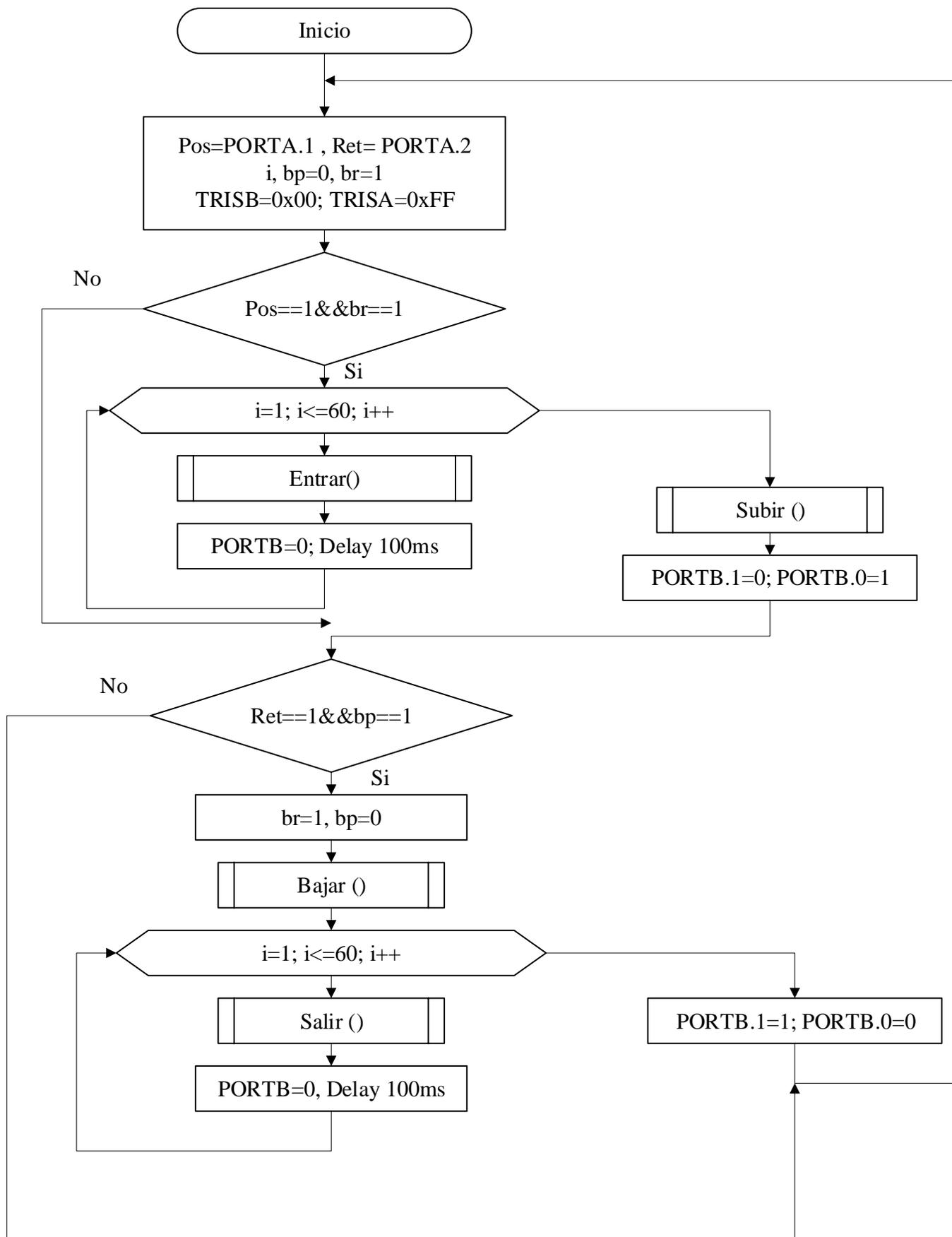


Diagrama de flujo del control del sistema de posicionamiento de las muestras.

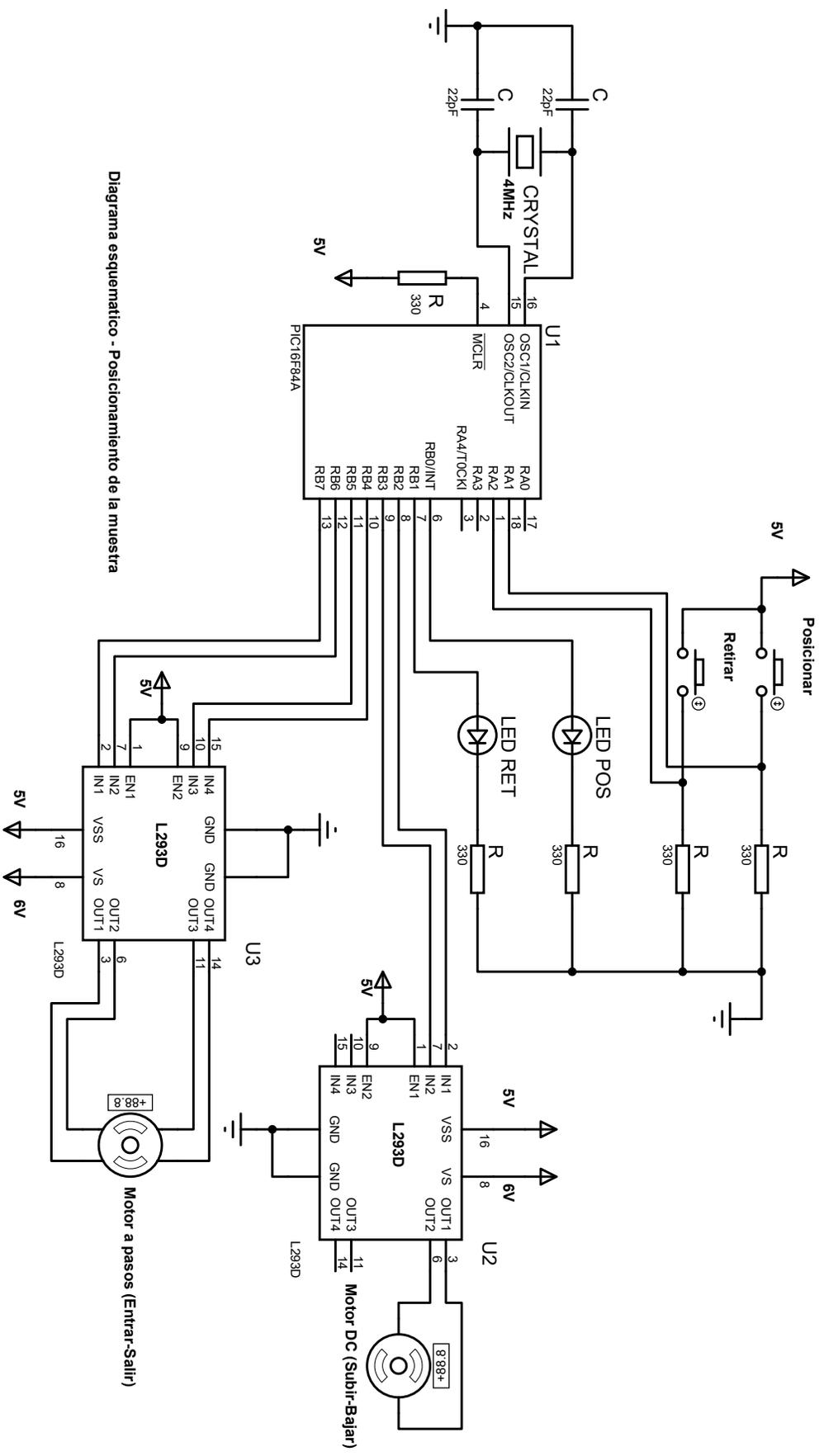


Diagrama esquemático - Posicionamiento de la muestra

Programa de control de los motores para el sistema de posicionamiento de las muestras.

```
#define Posicionar PORTA.F1
#define Retirar PORTA.F2
void Entrar();
void Salir();
void Subir();
void Bajar();
int i, bp=0, br=1;
```

Definición de:

- botones,
- funciones
- variables

```
void main() {
    TRISB = 0x00; //Makes PORTB0 or RB0 Output Pin
    PORTB = 0x00;
    TRISA = 0xFF;
    PORTA = 0x00;
```

Definición de entradas y salidas para los puertos del PIC

```
while(1) //Infinite Loop
```

```
{
```

```
if ((Posicionar==1) && (br==1)){
    bp=1; br=0;
    for (i=1;i<=60;i++) {
        Entrar();
        PORTB = 0x00;    //Apagar el motor.
        Delay_ms(100);  //Pausa de 500ms al final del movimiento.
    } // Fin del for entrar
    Subir();
    PORTB.F1=0;
    PORTB.F0=1;
} // Fin if posicionar
```

Rutina para el botón Posicionar

```
if ((Retirar==1) && (bp==1)){
    br=1; bp=0;
    Bajar();
    for (i=1;i<=60;i++) {
```

4.3.5. Diseño del armazón del prototipo

El diseño del armazón del prototipo se realizó en el software de SolidWorks. Este es un software CAD (diseño asistido por computadora) para modelado mecánico en 2D y 3D. Permite modelar piezas y ensambles, para posteriormente extraer de ellos tanto planos técnicos como otro tipo de información necesaria para la construcción.

Para el diseño el armazón se consideran los requisitos del sistema: dimensiones, resistencia, terminales para las conexiones, y las aberturas necesarias para la pantalla y botones externos para el encendido y apagado del equipo, así como el control del sistema de posicionamiento de las muestras.

Las dimensiones finales del equipo son $31\text{ cm} \times 22\text{ cm} \times 18\text{ cm}$. Para el traslado adecuado del prototipo se le colocan asas en la parte posterior y anterior.

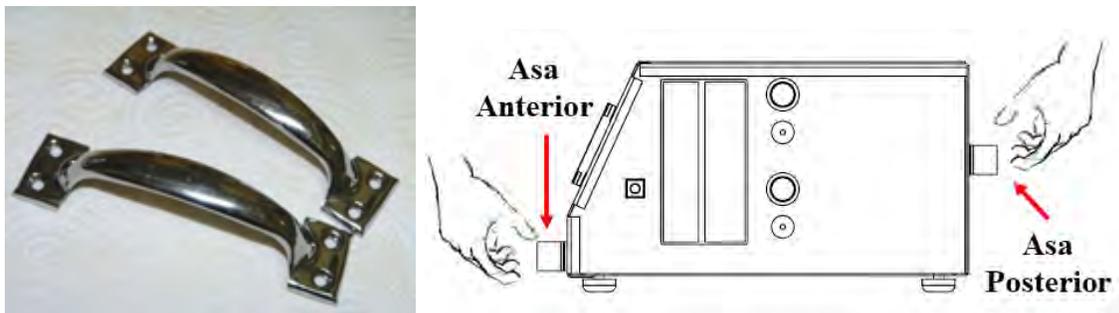


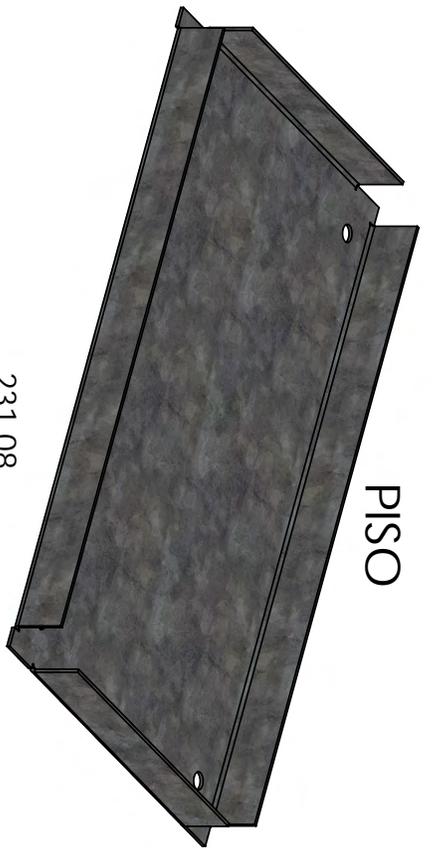
Figura 4.54: *Asas del prototipo para su traslado.*

En la base (piso) del prototipo se tiene un ajuste de nivel, hecho con tornillos con cabeza de caucho, que ayudan a amortiguar el posicionamiento del equipo y colocarlo en diferentes superficies.

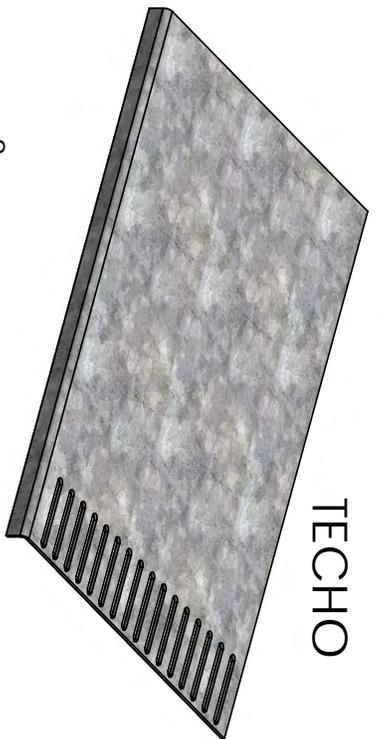
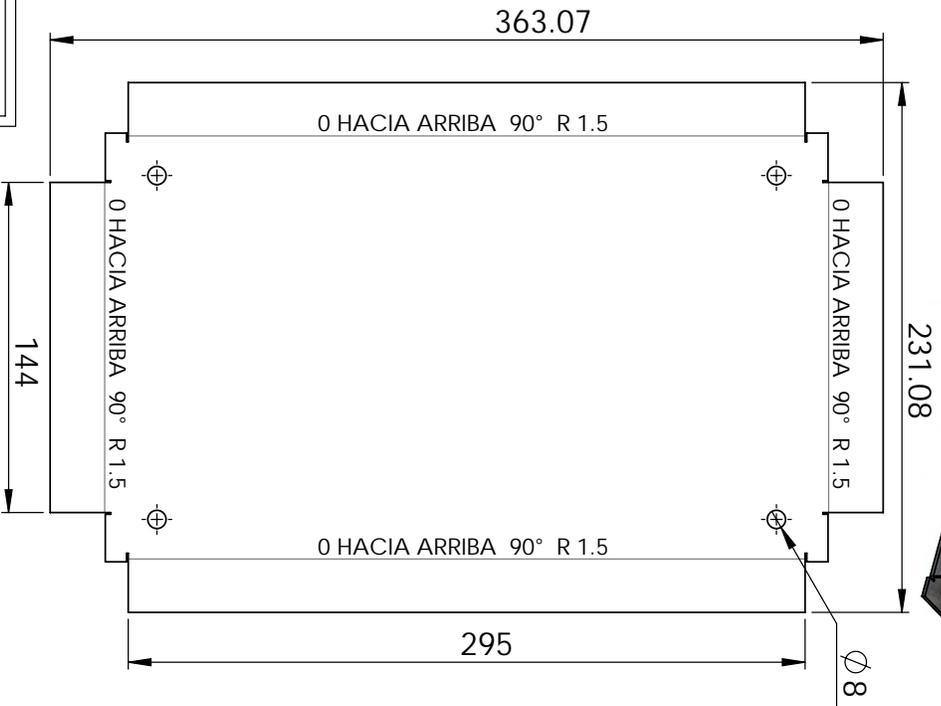


Figura 4.55: *Elementos para el ajuste de nivel del prototipo.*

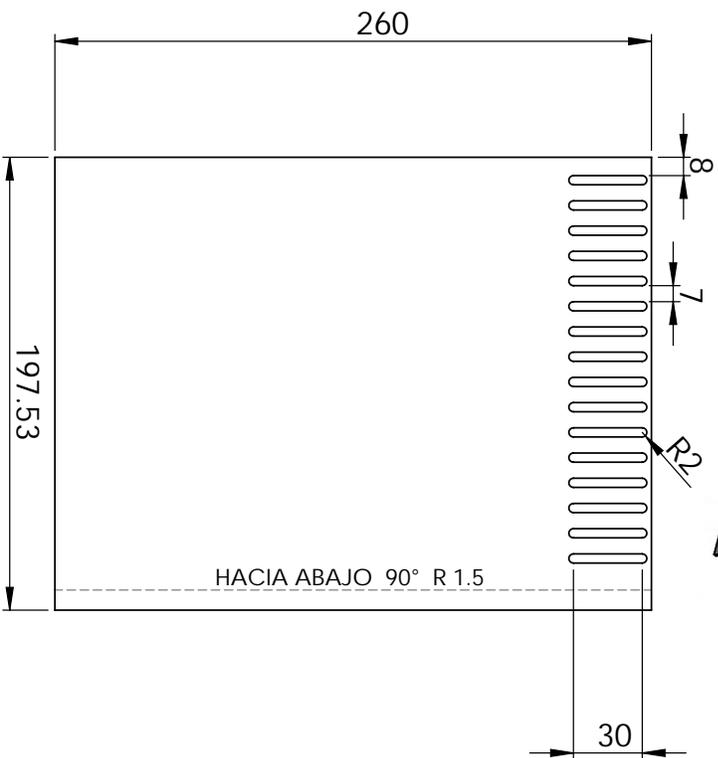
En las siguientes páginas se muestra el diseño del armazón del prototipo.



PISO



TECHO



Elaboró:
Ma. del Rosario
Bautista Morales

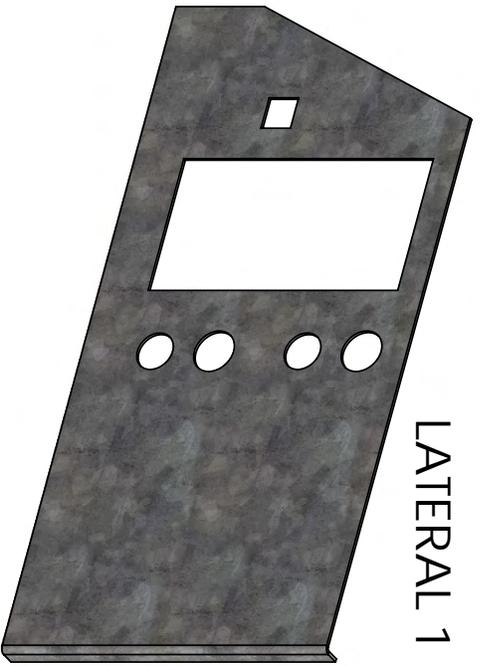
Centro de Investigaciones en Óptica, A.C.
Maestría en Optomecatrónica

PISO Y TECHO

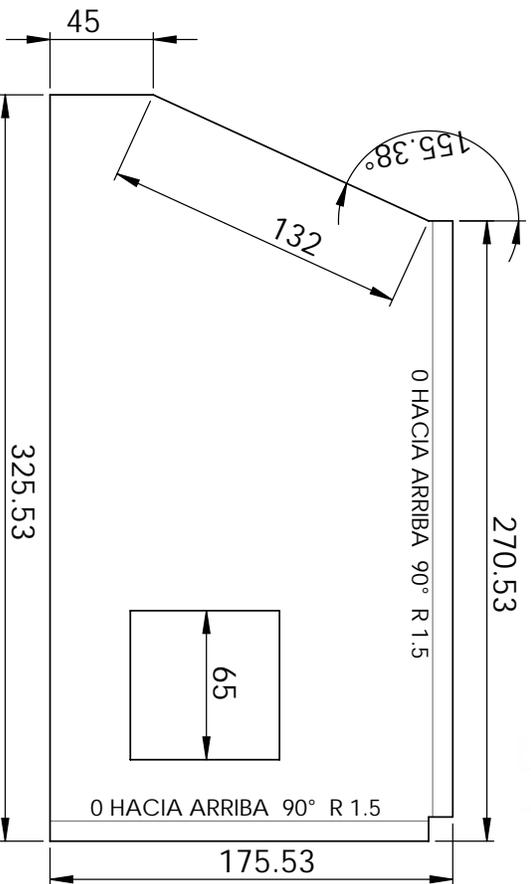
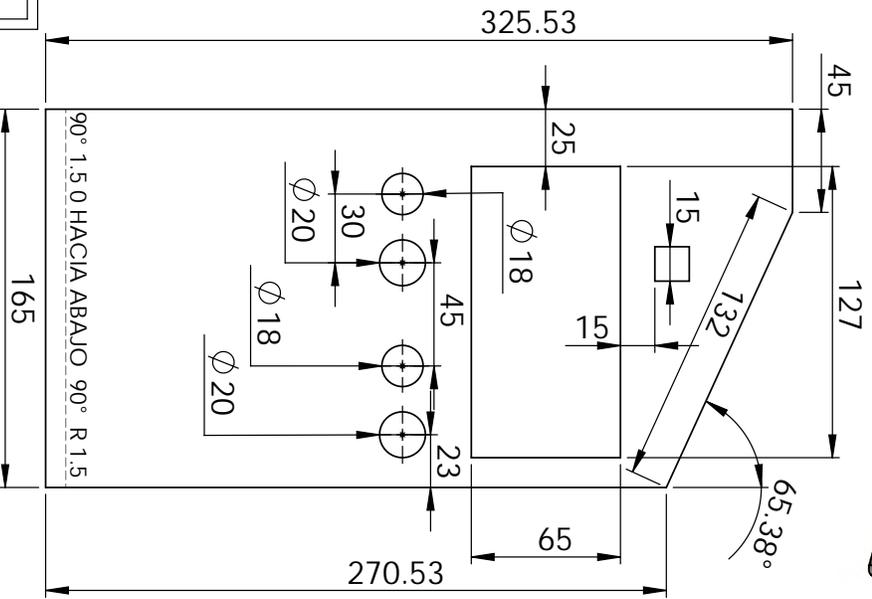
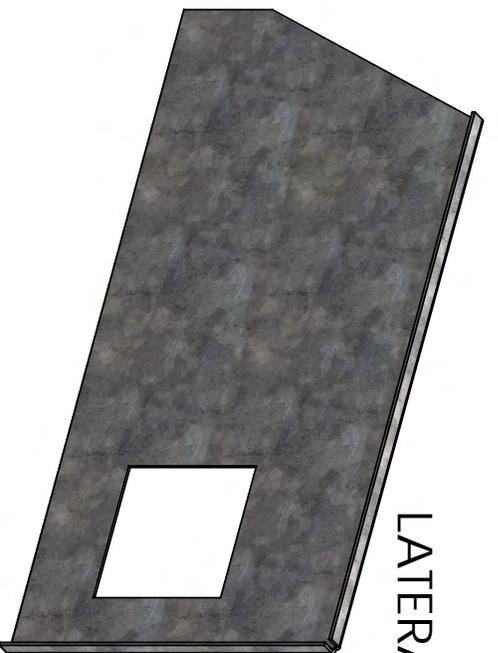
Acotación:
mm

Escala:
1:3

LATERAL 1



LATERAL 2



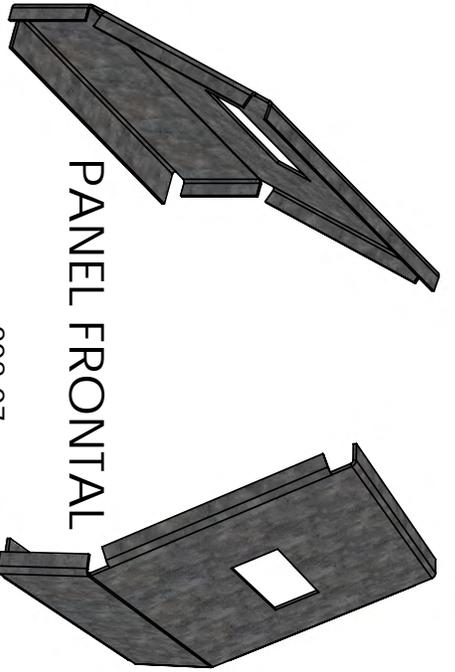
Elaboró:
Ma. del Rosario
Bautista Morales

LATERALES

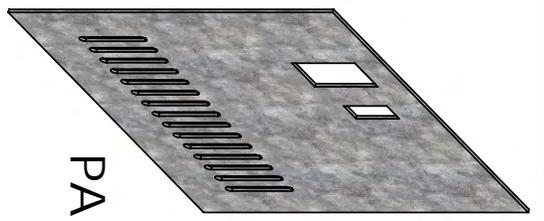
Acotación: mm
Escala: 1:3

Centro de Investigaciones en Óptica, A.C.

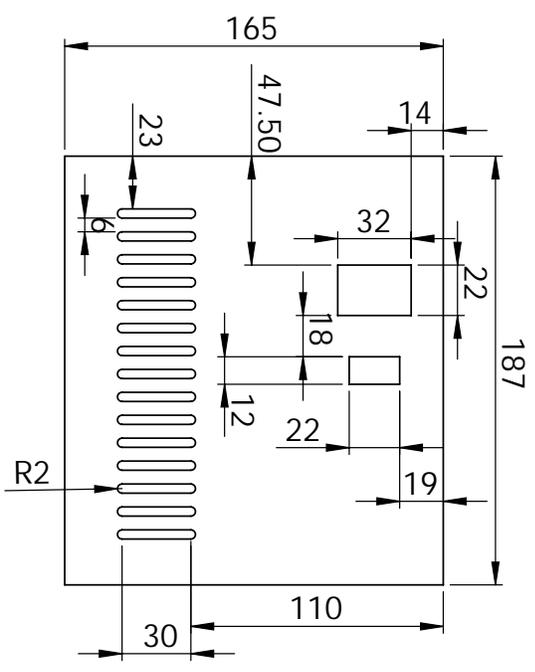
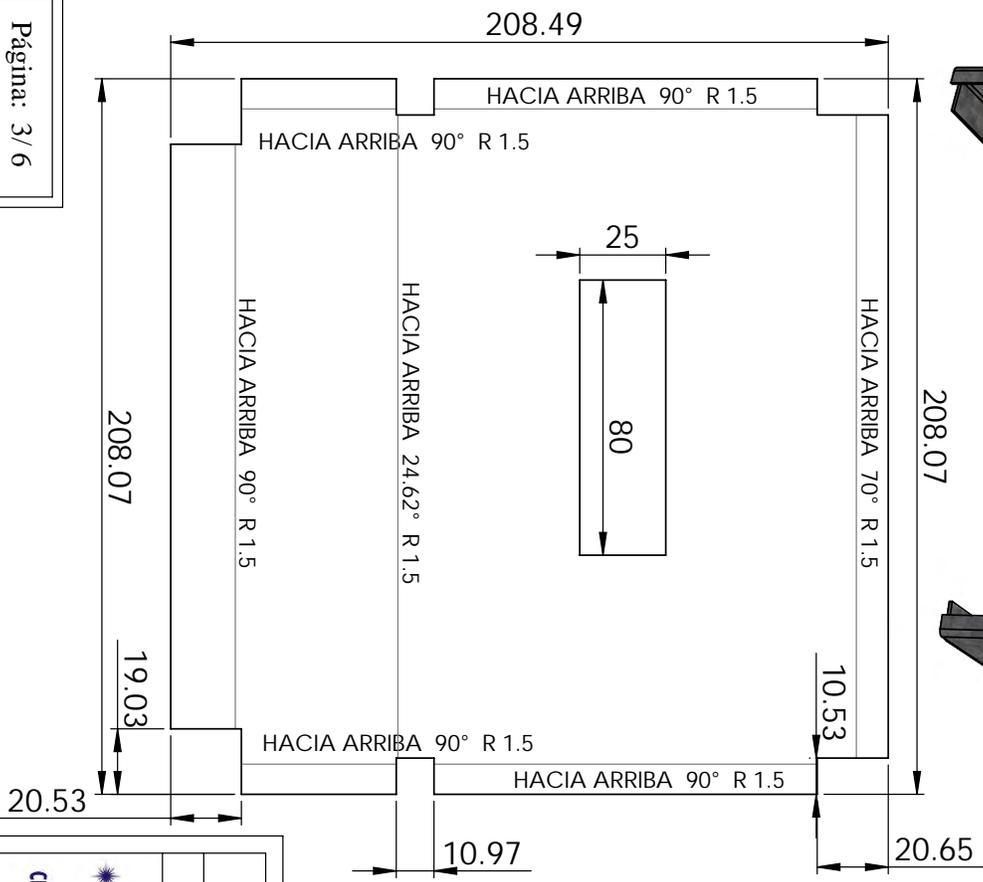
Maestría en Optomecatrónica



PANEL FRONTAL



PANEL POSTERIOR



Centro de Investigaciones en Óptica, A.C.

Maestría en Optomecatrónica

Elaboró:

Ma. del Rosario Bautista Morales

PANELES FRONTAL Y POSTERIOR

Acotación:

mm

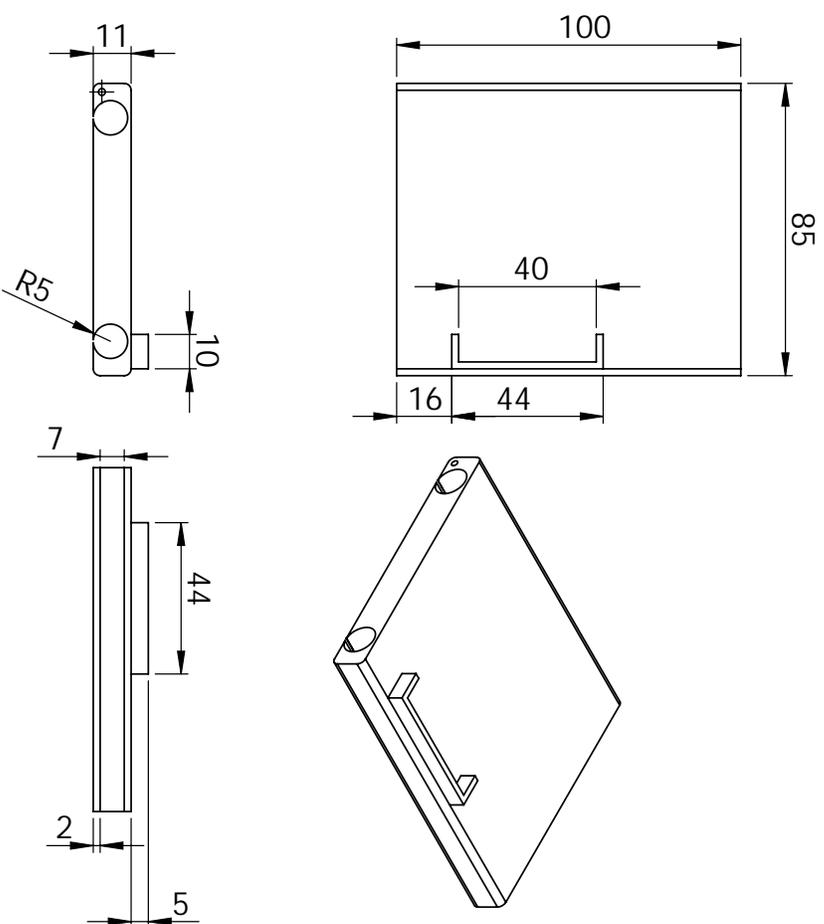
Escala:

1:3

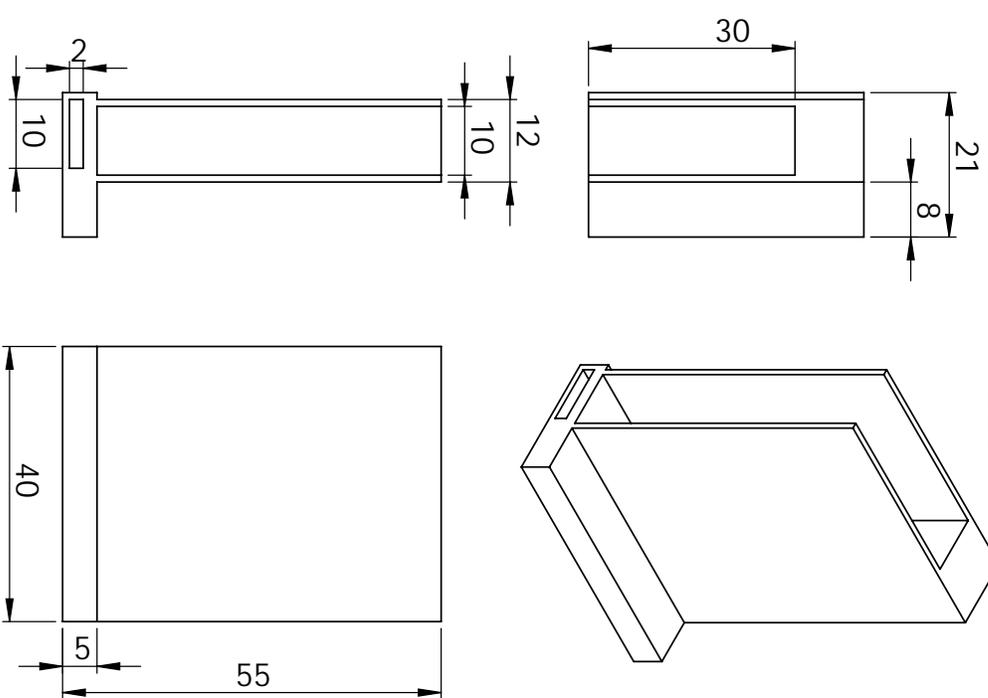


CENTRO DE INVESTIGACIONES EN ÓPTICA, A.C.

PLATAFORMA MÓVIL 1



SOPORTE PARA CELDAS



Centro de Investigaciones en Óptica, A.C.

Maestría en Optomecatrónica

Elaboró:

Ma. del Rosario
Bautista Morales

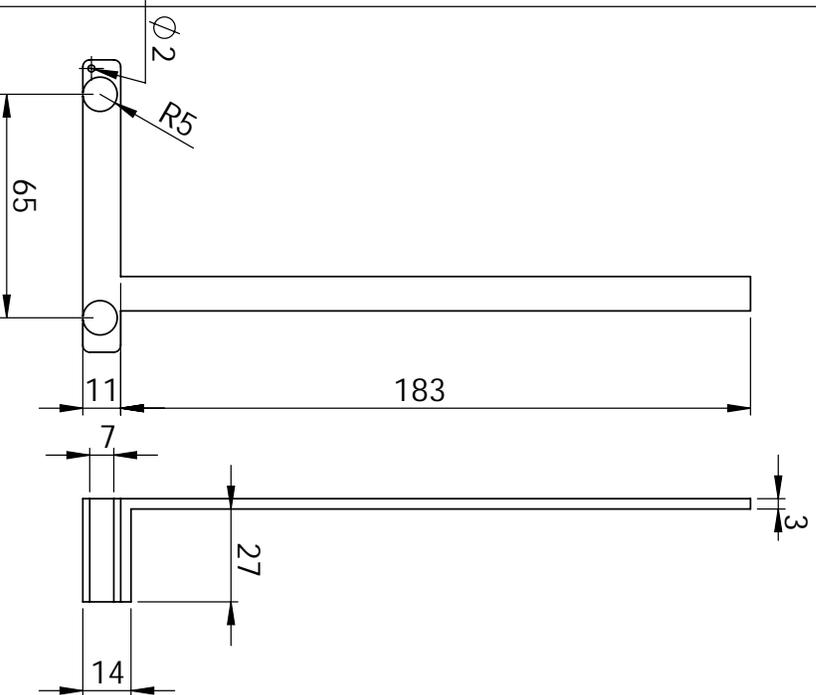
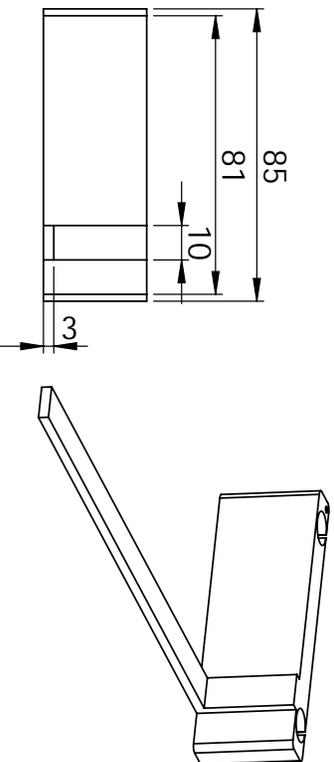
SOPORTE PARA CELDAS 1

Acotación:

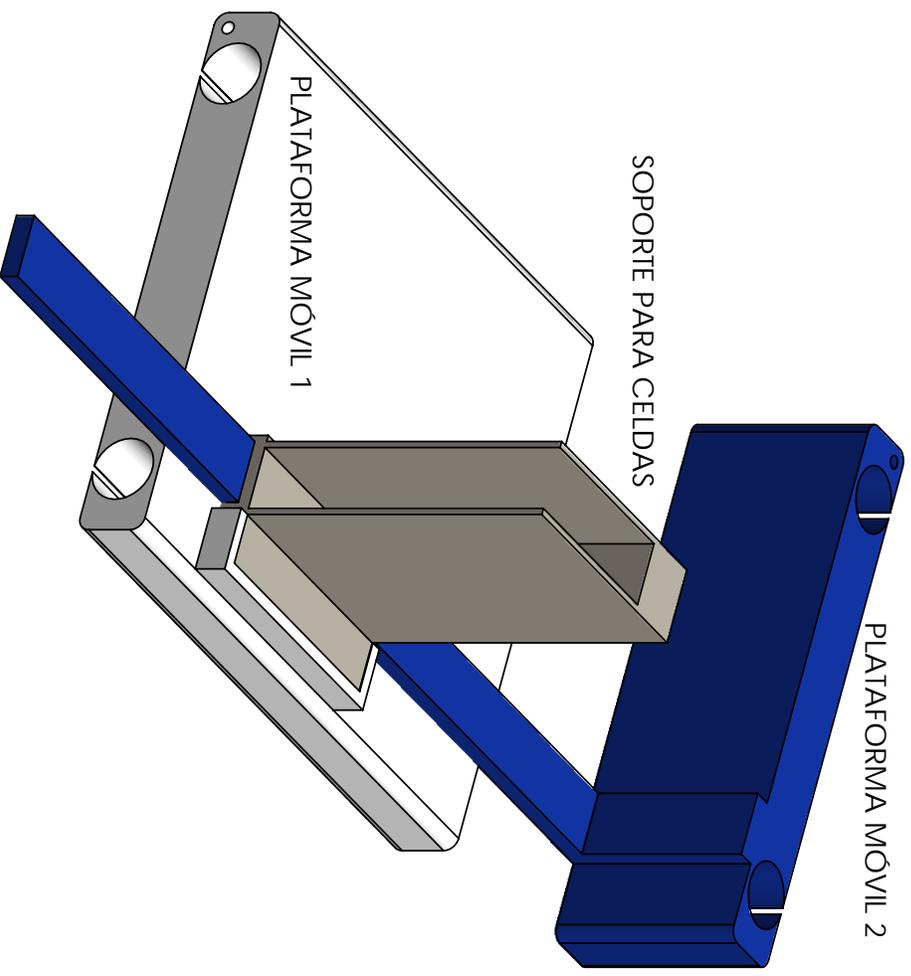
mm Escala: 1:2 & 1:1



PLATAFORMA MÓVIL 2



MECANISMO PARA COLOCACIÓN Y POSICIONAMIENTO DE LAS CELDAS



Centro de Investigaciones en Óptica, A.C.

Maestría en Optomecatrónica

SOPORTE PARA CELDAS 2

Elaboró:

Ma. del Rosario
Bautista Morales

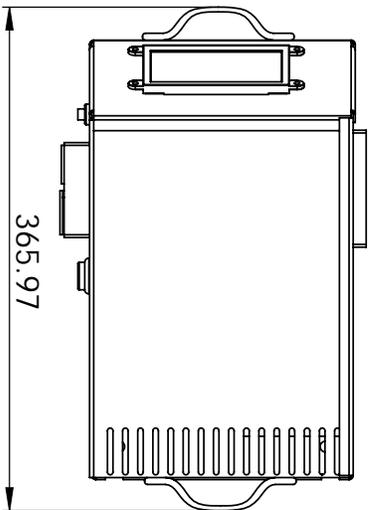


CENTRO DE INVESTIGACIONES
EN ÓPTICA, A.C.

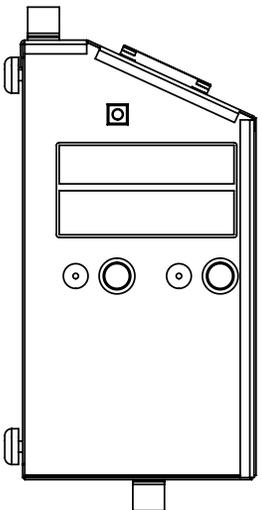
Acotación: Escala:

mm 1:2 & 1:1

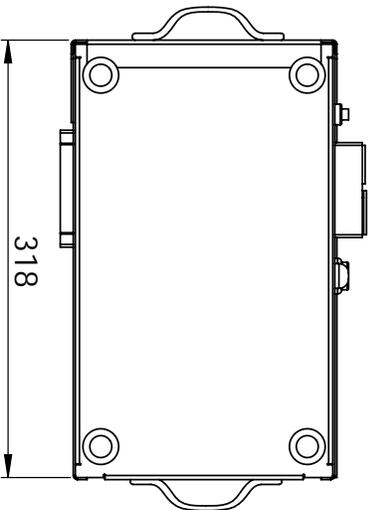
VISTA SUPERIOR



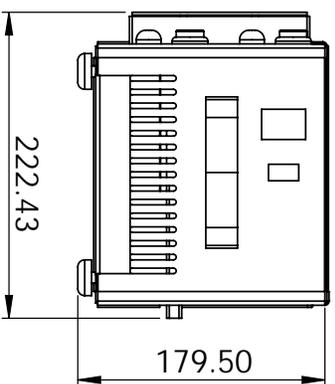
VISTA LATERAL



VISTA INFERIOR



VISTA POSTERIOR



Centro de Investigaciones en Óptica, A.C.

Maestría en Optomecatrónica

Elaboró:

Ma. del Rosario
Bautista Morales

PROTOTIPO

Acotación:

mm

Escala:

1:5



En la Fig. 4.56 se observa la estructura interna del prototipo que tiene la función de sostener, dar soporte y crear divisiones donde se colocarán todos los elementos internos para el funcionamiento del prototipo (circuitos de control, tarjeta de desarrollo, controlador del diodo láser, etc). Además se visualiza el montaje de las estructuras de posicionamiento de las muestras.

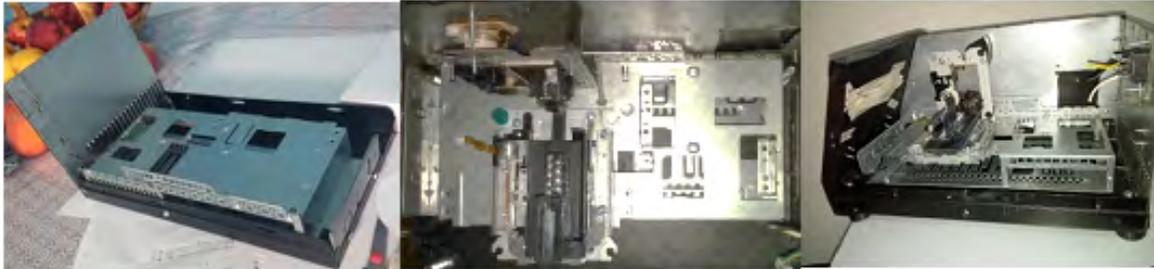


Figura 4.56: Estructura interna del prototipo para soporte de elementos. Y Sistema de posicionamiento de muestras.

Finalmente, la Fig. 4.57 muestra el ensamble de todos los elementos mecánicos. Una característica del prototipo es que tiene la capacidad de ser montado y desmontado con facilidad, gracias a que las piezas del ensamble están sujetas con tornillos (de 6mm de diámetro).



Figura 4.57: Ensamble del prototipo.

4.3.6. Conclusión

Se construyó y ensambló el armazón del prototipo guiándose en los planos diseñados en SolidWorks, se consideraron las dimensiones de los componentes que se posicionaran adentro del equipo. Además, se agregaron estructuras de soporte que permiten que cada elemento se mantenga en su posición designada. Se logró un prototipo robusto y capaz de ser transportado.

4.4. Conclusión

En este capítulo se describieron todas las partes mecánicas que conforman la instrumentación del prototipo. Se desarrolló el algoritmo de la adquisición y procesamiento de las señales para el cálculo de IR con la creación de varias funciones que permiten que el flujo del programa facilite para su interpretación.

Se diseñó la interfaz humano-maquina con la pantalla táctil de la familia 4D System, de fácil manejo para el usuario. Los softwares implementados para estas secciones son aquellos desarrollados precisamente para trabajar con el hardware seleccionado esto permite explotar todas las herramientas de los componentes y simplifica la integración del prototipo.

Finalmente se construyó la parte mecánica del prototipo con láminas de acero, esto le proporciona robustez y rigidez al equipo. Una vez explicadas las secciones que componen al prototipo y cómo funcionan cada una de ellas, en el siguiente capítulo se describe la integración de todas las partes y se muestra el funcionamiento total del sistema desarrollado.

CAPÍTULO 5

Integración del sistema y resultados finales

En este capítulo se muestra la integración del sistema, y el prototipo armado. En base a los resultados obtenidos se redacta una discusión que abarca las limitaciones y alcances del mismo. Finalmente se proponen mejoras específicas que potenciarían el desempeño logrado.

5.1. Circuitos de acoplamiento de señales

Debido a las características de la señal de modulación en cuanto a amplitud (+80mVpp, abarcando un rango de +10 mV a +90 mV) la tarjeta se ve limitada por la capacidad del DAC, ya que este no puede proporcionar voltajes menores a +50 mV.

Se diseñó y armó un circuito de acoplamiento que se encarga de reducir el offset de la señal generada por la tarjeta de desarrollo. Es decir, originalmente la tarjeta manda una señal de +80 mVpp, de +50 mV a +130 mV. El circuito implementado es un restador no inversor activo con un OPAM OP07, para desplazar -40 mV y alcanzar la señal de modulación óptima.

El OP07 es un amplificador operacional con un bajo nivel de ruido, y un amplio rango de voltaje de alimentación entre $\pm 3 V$ a $\pm 18 V$, también presenta una corriente de polarización de entrada baja y una ganancia de lazo abierto alta. El bajo offset que posee y la alta ganancia de lazo abierto hacen que el OP07 sea particularmente útil para aplicaciones de instrumentación de alta ganancia. Esto permite que sea implementado como compensador de offset de la señal de modulación en el prototipo.

El circuito restador se muestra en la Fig. 5.1. Los valores de R son de $33 k\Omega$, y R_0 de $10 k\Omega$; con un voltaje de alimentación de $\pm 12 V$. El valor de voltaje de salida está dado por:

$$V_{out} = V_2 - V_1 \quad (5.1)$$

Donde V_2 , se conecta a la señal de modulación generada por la tarjeta de desarrollo, y V_1 es conectado al valor de voltaje de DC que se requiere para desplazar el offset. V_{out} es la señal acondicionada que se conecta a la entrada del controlador del diodo láser para lograr la sintonización.

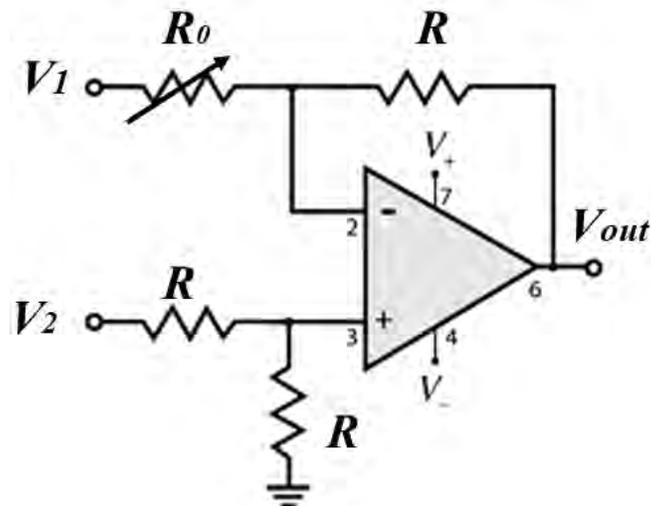


Figura 5.1: Circuito de acondicionamiento de la señal de modulación. OPAM Restador No Inversor

El siguiente circuito de acondicionamiento es para la señal proveniente del sistema óptico (la señal de interferencia). La finalidad de este circuito es amplificar la señal y facilitar la lectura por el ADC de la tarjeta, además de eliminar el ruido de alta frecuencia por lo que se implementa un circuito RC de filtrado tipo pasa bajos a 47 Hz .

El circuito de amplificación se construyó con un OPAM Amplificador No Inversor, se utiliza un OP177. El OP177 es un amplificador de ultrapresición con un bajo nivel de ruido y un CMRR de 130 dB .

La Fig. 5.2, muestra el diagrama esquemático del circuito de acondicionamiento.

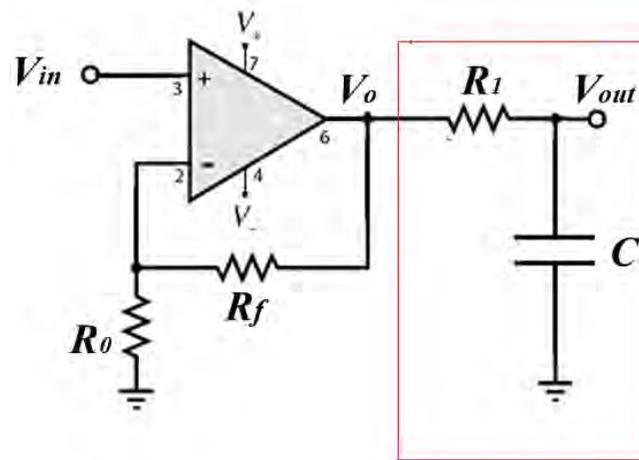


Figura 5.2: Circuito de acondicionamiento de la señal de interferencia. OPAM Amplificador no inversor y filtraje tipo pasa bajos (recuadro rojo).

Del análisis del amplificador operacional obtenemos el voltaje de salida V_0 :

$$V_0 = V_{in} \left(1 + \frac{R_f}{R_0} \right) \quad (5.2)$$

con $R_f = 330 \text{ k}\Omega$ y $R_0 = 18 \text{ k}\Omega$

En este caso V_{in} corresponde a la señal proveniente del fotodiodo donde la señal de las fibras es detectada, una vez amplificada entra al circuito de filtraje tipo pasa bajos RC, recuadro rojo Fig. 5.2. Donde $R_1 = 47 \text{ k}\Omega$ y $C = 82 \text{ nF}$

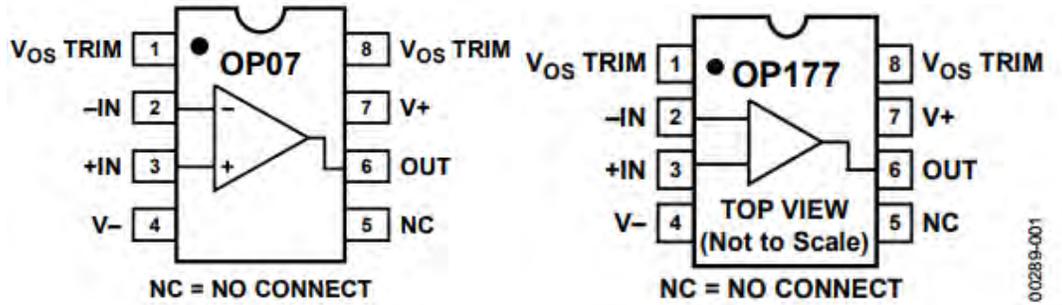


Figura 5.3: Configuración de pines para los integrados OP07 y OP177. [52], [53]

Los circuitos de acondicionamiento se construyeron sobre una placa fenólica perforada y se soldaron y aislaron. En la Fig. 5.4, se observan los circuitos armados y posicionados dentro del prototipo. La Fig. 5.5 muestra el resultado del rendimiento de los circuitos del acondicionamiento de las señales.

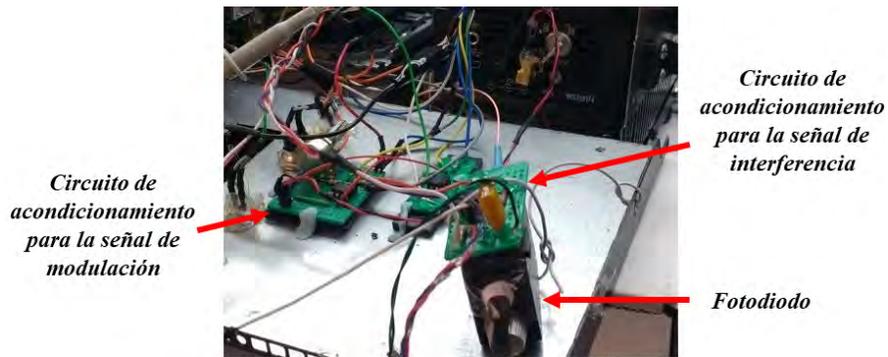


Figura 5.4: Circuitos de acondicionamiento armados y soldados dentro del prototipo.

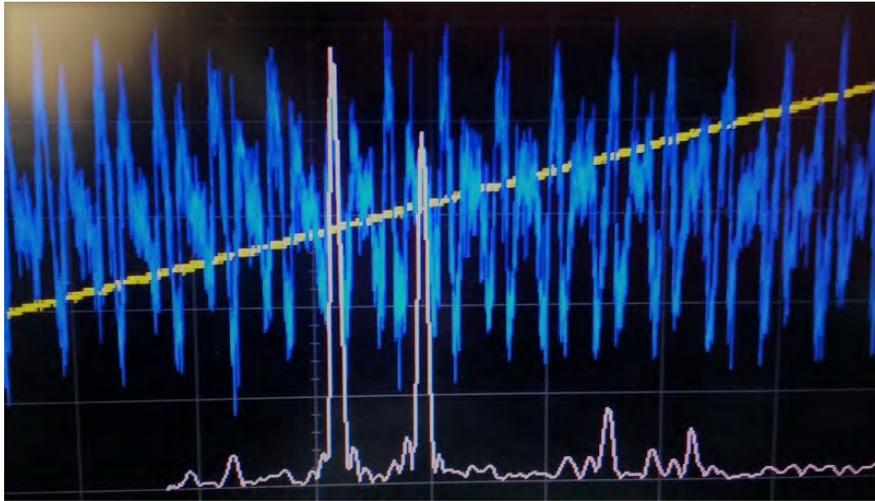


Figura 5.5: Señal de modulación desplazada con el offset requerido por el sistema (amarillo). Señal de interferencia por parte del sistema óptico, amplificada y filtrada (azul). Transformada de Fourier de la señal de interferencia, se observan los dos picos correspondientes a cada sensor del sistema, (rosa).

5.2. Fuente de alimentación

El sistema tiene varios requisitos en cuanto voltaje y corriente para cada sección del prototipo. Se utilizó una fuente conmutada de potencia para alimentar el sistema [54] Los voltajes de alimentación se muestran en la Tabla 5.1, junto con la demanda de corriente de cada sección:

Tabla 5.1: Distribución de potencia en el prototipo

	+5 V	± 12 V
200 mA	Circuito de control para el sistema de posicionamiento de las muestras. Tarjeta de desarrollo STM32F767ZI	Circuito de acondicionamiento para la señal de modulación Circuito de acondicionamiento para la señal de interferencia
> 500 mA	Pantalla táctil	+12 V Motores DC y motor a pasos para el posicionamiento de las muestras

Otro factor importante es el calibre de los cables para la conexión, en este caso se utilizaron 3 tipos, 5.2:

Tabla 5.2: Calibres de los cables para la conexión entre componentes del prototipo.

Calibre	Diámetro [mm]	Área [mm ²]	Conexión
12	2.050	3.310	Corriente alterna (85 V- 264 V)
22	0.644	0.325	Unión entre tarjetas de control, botones, alimentación corriente directa ($\pm 12 V$, +5 V)
28	0.321	0.080	Señales de modulación y de interferencia (mV)

La Fig. 5.6 muestra una fotografía con ejemplos de la implementación de los diferentes tipos de calibres en las conexiones dentro del prototipo.

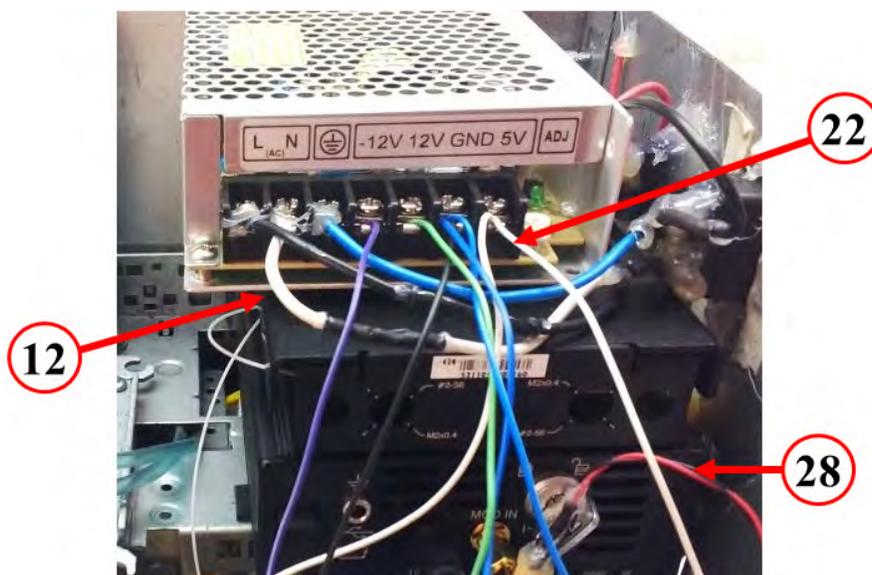


Figura 5.6: Fuente de alimentación señalando el calibre de cables utilizado para cada sección del prototipo.

5.3. Integración del prototipo

La Fig. 5.7 muestra los componentes que integran el prototipo, estos son:

1. Armazón metálico del prototipo.
2. Sistema de posicionamiento de las muestras.
3. Conexión de la pantalla a la tarjeta de desarrollo.
4. Circuito de control de motores para el posicionamiento de las muestras.
5. Puntas del refractómetro de fibra óptica

6. Fotodetector.
7. Fuente de alimentación del prototipo.
8. Controlador del diodo laser.
9. Tarjeta de desarrollo.
10. Circuito de acondicionamiento de la señal de modulación.

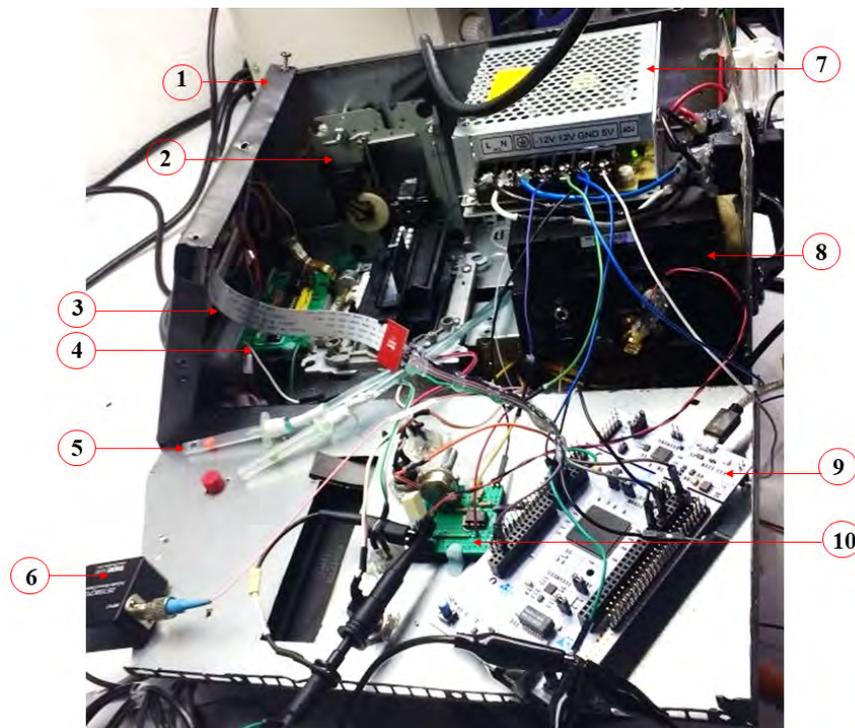


Figura 5.7: Integración del prototipo, vista preliminar de componentes internos.

Para probar el desempeño del prototipo se realizó se midió el IR del aire, (normalmente implementada como la señal de referencia) en diferentes momentos del día. El resultado reflejo la importancia del cálculo de una curva de calibración independiente para cada sensor del sistema óptico.

La Fig. 5.8, muestra la pantalla de medición en una ronda de medidas, se aprecia como la compensación que se necesita para cada sensor dentro del sistema óptico es independiente, esto es debido a que las cavidades HOF son diferentes entre sí, por lo que la cantidad de luz reflejada en cada una varía en intensidad.

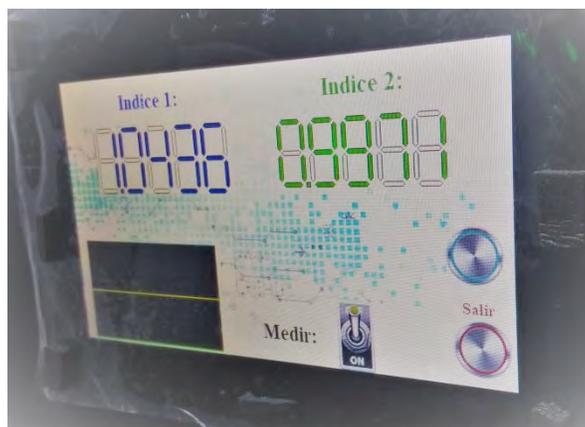


Figura 5.8: Pantalla de medición en funcionamiento. Muestra el valor de IR de los sensores en aire.

Una vez corroborado el funcionamiento del equipo, este se armó en su totalidad para poder transportarlo. Se realizaron mediciones finales con el sistema armado para comprobar que todas las secciones y componentes que constituyen al prototipo funcionaran como un conjunto. Los resultados fueron óptimos, no se observó pérdida de potencia en los componentes y los controles de manipulación actuaron con normalidad.



Figura 5.9: Prototipo final ensamblado.

5.4. Discusión

Existen varios puntos claves en los que es necesario poner énfasis en el prototipo, uno de ellos es que en la técnica de C-OFDR es un requisito tener fuentes ópticas de alto rendimiento que proporcionen un ajuste de frecuencia rápido y lineal. Sin embargo, los diodos láseres disponibles muestran en práctica fluctuaciones en su tasa de sintonización de frecuencia óptica.

Debido a estas características de sintonización no lineal, el muestreo de la señal de interferencia degrada la resolución espacial de la medición OFDR. Se recomienda que a futuro se implemente un algoritmo de corrección de no linealidad con lo cual la resolución y rango de medición del sistema pueda verse mejorado.

Las cavidades HOF del sistema óptico, tuvieron una pérdida de aproximadamente -10dB, la cual es muy alta. Para optimizar los resultados se recomienda utilizar cavidades HOF de menor longitud y que estas se encuentren en fase entre sí.

El algoritmo de procesamiento de la tarjeta de desarrollo es funcional. La adquisición de un barrido completo se realiza 10 veces seguidas, posteriormente el procesamiento de la señal y adquisición de la amplitud de picos en la transformada de Fourier se realiza con la media de todos picos leídos en una ronda. Esto lleva aproximadamente 40 seg., es decir se puede tomar la medición de IR en aproximadamente 1.5 min. Cabe destacar que tanto el algoritmo como el tiempo de procesamiento puede ser el mismo tanto para los 2 puntos de detección que tiene el prototipo, así como para los sensores requeridos a los que el sistema multipunto pueda expandirse.

Es posible mejorar los resultados dentro del algoritmo de procesamiento si en vez de considerar un solo punto como valor máximo en las componentes en frecuencia, se promedian 3 o más puntos del pico de la transformada de Fourier.

El software de desarrollo, la capacidad de procesamiento, la creación de funciones y subrutinas en segundo plano dan la pauta a la posibilidad de generar un menú de opciones que le den al prototipo diferentes funcionalidades y expandir las aplicaciones del mismo.

En el prototipo existen cambios de temperatura por lo que se recomienda que exista un control de temperatura que resguarde la zona de colocación de la muestra. Esto es debido a que a pesar de que los sistemas óptico y electrónico sean suficientemente estables a cambios de temperatura, las muestras si vanean sus propiedades con la temperatura pueden llegar a cambiar su valor de IR, y esto se vera reflejado con resultados no concordantes. Existe la posibilidad de explorar nuevas aplicaciones con el sistema. Probar con diferentes productos dentro del mercado.

5.5. Conclusión

Con el presente trabajo de investigación se lograron alcanzar los siguientes resultados:

- Se diseñaron diferentes tipos de refractómetros para la prueba de concepto y la familiarización de conceptos por parte del estudiante, esto llevo a la redacción de un artículo publicado.
- Se diseño el sistema óptico de un refractómetro de fibra con un error de 1×10^{-4} , el cual permite demultiplexar varios sensores para obtener un sistema multipunto, con la capacidad de extenderse hasta 10 sensores.
- Se realizaron pruebas de concepto con diferentes muestras para caracterización del sistema, así como pruebas con muestras para aplicaciones de control de calidad, con lo

que se concluye que si es posible discriminar entre una muestra control y una muestra contaminada.

- Se desarrollo una interfaz humano-maquina con la que el usuario puede interactuar de forma sencilla con el equipo.
- Se desarrollo un algoritmo de programación con una tarjeta de desarrollo de 32 bits lo que da pauta a la posibilidad de mejorar el procesamiento y crear funciones extras en el prototipo.
- Se diseño el armazón del sistema y se construyo con láminas de acero que ayudan a darle soporte y robustez.
- Se desarrollo un sistema semi-automático de posicionamiento de muestras.

Con todo lo antes mencionado se logró construir un prototipo de un refractómetro portátil funcional, esto sienta las bases para el desarrollo equipos tecnológicos mejorados en el CIO.

5.6. Trabajo a futuro

Lo siguientes puntos se proponen para mejorar los resultados obtenidos por el prototipo:

La señal de modulación es de una frecuencia muy baja se recomienda probar con diferentes varoles en frecuencia y analizar la respuesta del sistema. Esto además ayudará a que el tiempo de adquisición y procesamiento se optimice.

Además de realizar un filtraje externo, es posible la implementación de un filtraje digital de la señal considerando a que frecuencia resuenan los sensores del sistema óptico para evitar la pérdida de información importante.

Mejorar los resultados ya sea con un algoritmo para calcular el promedio de 3 posiciones de los picos de las componentes en frecuencia o incluir algoritmo de corrección de no linealidad para obtener las componentes en frecuencia menos anchas.

Implementar un control de temperatura dentro del prototipo que permite asegurar que las muestras no se ven alteradas con las posibles variaciones de temperatura dentro del mismo.

Bibliografía

- [1] Fardad Koohyar. «Refractive Index and Its Applications». En: *Journal of Thermodynamics & Catalysis* 4 (2013).
- [2] Jörg Rheims, Jan Köser y Thomas Wriedt. «Refractive-index measurements in the near-IR using an Abbe refractometer». En: *Measurement Science and Technology* 8.6 (1997), pág. 601.
- [3] Chun-Liu Zhao y col. «Simple fresnel reflection-based optical fiber sensor for multipoint refractive index measurement using an AWG». En: *IEEE Photonics Technology Letters* 25.6 (2013), págs. 606-608.
- [4] Pengfei Wang y col. «High-sensitivity, evanescent field refractometric sensor based on a tapered, multimode fiber interference». En: *Optics letters* 36.12 (2011), págs. 2233-2235.
- [5] T Allsop y col. «A high sensitivity refractometer based upon a long period grating Mach–Zehnder interferometer». En: *Review of scientific instruments* 73.4 (2002), págs. 1702-1705.
- [6] Kerstin Schroeder y col. «A fibre Bragg grating refractometer». En: *Measurement Science and technology* 12.7 (2001), pág. 757.
- [7] Agustín González-Cano y col. «Plasmonic sensors based on doubly-deposited tapered optical fibers». En: *Sensors* 14.3 (2014), págs. 4791-4805.
- [8] Kivilcim Yuksel y col. «Optical frequency domain reflectometry: A review». En: *Transparent Optical Networks, 2009. ICTON'09. 11th International Conference on*. IEEE. 2009, págs. 1-5.
- [9] John E Saunders y col. «Refractive indices of common solvents and solutions at 1550 nm». En: *Applied optics* 55.4 (2016), págs. 947-953.
- [10] Tao Zhu y col. «Fabry–Perot optical fiber tip sensor for high temperature measurement». En: *Optics Communications* 283.19 (2010), págs. 3683-3685.
- [11] Domenico Paladino y col. «Hybrid fiber grating cavity for multi-parametric sensing». En: *Optics express* 18.10 (2010), págs. 10473-10486.
- [12] C Gouveia y col. «Fabry–Pérot cavity based on a high-birefringent fiber Bragg grating for refractive index and temperature measurement». En: *IEEE Sensors Journal* 12.1 (2012), págs. 17-21.
- [13] A ASSEH S SANDGREN H AHLFELDT y B SAHLGREN R STUBBE G EDWALL. «Fiber optical Bragg grating refractometer». En: *Fiber & Integrated Optics* 17.1 (1998), págs. 51-62.

-
- [14] Vikram Bhatia. «Applications of long-period gratings to single and multi-parameter sensing». En: *Optics Express* 4.11 (1999), págs. 457-466.
- [15] Yi-Ping Wang y col. «Highly sensitive long-period fiber-grating strain sensor with low temperature sensitivity». En: *Optics letters* 31.23 (2006), págs. 3414-3416.
- [16] GH Meeten y AN North. «Refractive index measurement of turbid colloidal fluids by transmission near the critical angle». En: *Measurement Science and Technology* 2.5 (1991), pág. 441.
- [17] Xuewen Shu y col. «Sampled fiber Bragg grating for simultaneous refractive-index and temperature measurement». En: *Optics Letters* 26.11 (2001), págs. 774-776.
- [18] Sachin Kumar Srivastava y Banshi Dhar Gupta. «A multitapered fiber-optic SPR sensor with enhanced sensitivity». En: *IEEE Photonics Technology Letters* 23.13 (2011), pág. 923.
- [19] Ignacio Del Villar y col. «Lossy mode resonance generation with indium-tin-oxide-coated optical fibers for sensing applications». En: *Journal of Lightwave Technology* 28.1 (2010), págs. 111-117.
- [20] Alan D Kersey. «Multiplexed fiber optic sensors». En: *Proc. SPIE*. Vol. 1797. 1992, págs. 161-185.
- [21] Hui Su y Xu Guang Huang. «Fresnel-reflection-based fiber sensor for on-line measurement of solute concentration in solutions». En: *Sensors and Actuators B: Chemical* 126.2 (2007), págs. 579-582.
- [22] Rodolfo Martínez-Manuel, Óscar Esteban y Mikhail G Shlyagin. «Simple low-cost refractometer using a disposable optical fiber tip for measurements». En: *Optical Engineering* 55.11 (2016), págs. 116108-116108.
- [23] Hongyun Meng y col. «Michelson interferometer-based fiber-optic sensing of liquid refractive index». En: *Sensors and Actuators B: Chemical* 160.1 (2011), págs. 720-723.
- [24] T Kato y col. «Measurement of the nonlinear refractive index in optical fiber by the cross-phase-modulation method with depolarized pump light». En: *Optics letters* 20.9 (1995), págs. 988-990.
- [25] Hae Young Choi y col. «Cross-talk free and ultra-compact fiber optic sensor for simultaneous measurement of temperature and refractive index». En: *Optics express* 18.1 (2010), págs. 141-149.
- [26] Ma del Rosario Bautista-Morales, Rodolfo Martínez-Manuel y Carlos A Pineda-Arellano. «Overlapped fiber-optic Michelson interferometers for simultaneous refractive index measurement at two sensing points». En: *Applied Optics* 56.34 (2017), págs. 9515-9519.
- [27] ULPGC. *Capítulo 6: Conexiones de fibra óptica: uniones y acopladores*. 2017. URL: http://www.iuma.ulpgc.es/users/jrsendra/Docencia/Com_Opt_I/download/Com_Opt_I/Temario/conectores.pdf.

-
- [28] Byeong Ha Lee y col. «Interferometric fiber optic sensors». En: *Sensors* 12.3 (2012), págs. 2467-2486.
- [29] Pedro AS Jorge y col. «Fiber optic-based refractive index sensing at INESC Porto». En: *Sensors* 12.6 (2012), págs. 8371-8389.
- [30] Yue Ma y col. «Temperature-independent refractive index measurement based on Fabry-Perot fiber tip sensor modulated by Fresnel reflection». En: *Chinese Optics Letters* 10.5 (2012), pág. 050603.
- [31] Hae Young Choi y col. «Cross-talk free and ultra-compact fiber optic sensor for simultaneous measurement of temperature and refractive index». En: *Optics express* 18.1 (2010), págs. 141-149.
- [32] Ming Deng y col. «PCF-based Fabry-Pérot interferometric sensor for strain measurement at high temperatures». En: *IEEE Photon. Technol. Lett* 23.11 (2011), págs. 700-702.
- [33] Kivilcim Yüksel, Marc Wuilpart y Patrice Mégret. «Analysis and suppression of nonlinear frequency modulation in an optical frequency-domain reflectometer». En: *Optics express* 17.7 (2009), págs. 5845-5851.
- [34] Jesse Zheng. «Optical frequency-modulated continuous-wave interferometers». En: *Applied optics* 45.12 (2006), págs. 2723-2730.
- [35] Jianying Yuan y col. «A Fresnel reflection-based optical fiber sensor system for remote refractive index measurement using an OTDR». En: *Photonic sensors* 4.1 (2014), págs. 48-52.
- [36] P. Wyss. *Efficient quality control of liquids with density or refractive index determination*. 2017. URL: https://www.mt.com/mx/es/home/library/know-how/lab-analytical-instruments/Article_QC_UserCom7.html.
- [37] Inc. Wikimedia Foundation. *Laser diode*. 2017. URL: https://en.wikipedia.org/wiki/Laser_diode.
- [38] Grant Wernimont y F Hopkinson. «Dead-Stop End Point as Applied to Karl Fischer Method for Determining Moisture». En: *Industrial & Engineering Chemistry Analytical Edition* 15.4 (1943), págs. 272-274.
- [39] Sumin Kim y Hyun-Joong Kim. «Comparison of standard methods and gas chromatography method in determination of formaldehyde emission from MDF bonded with formaldehyde-based resins». En: *Bioresource Technology* 96.13 (2005), págs. 1457-1464.
- [40] LABOR material para laboratorio. *Refractómetros de Abbe*. 2017. URL: <http://comerciallabor.com/documents/ABBE.pdf>.
- [41] servoVendi. *Refractómetro BRIX*. 2017. URL: <https://www.servovendi.com/es/refractometro-brix-baume-para-vino-uvas-0-25-vol-atc-rhw-25-atc-be.html>.

-
- [42] Equipos & SolucionesMETTLER TOLEDO. *Refractómetro METTLER TOLEDO*. 2017. URL: <https://www.mt.com/mx/es/home.html>.
- [43] A Nova Analytics company B+S. *Refractómetros de mano & Instrumentación óptica*. 2017. URL: <https://es.scribd.com/document/288683448/670071095-tipos-de-refractometros-comerciales-docx>.
- [44] tilz0R. *HAL Library 14- Fast Fourier Transform for STM32Fxxx*. 2017. URL: <https://stm32f4-discovery.net/2015/07/hal-library-14-fast-fourier-transform-for-stm32fxxx/>.
- [45] tilz0R. *New library system built on STM32Fxxx HAL drivers*. 2017. URL: <https://stm32f4-discovery.net/2015/07/new-library-system-built-on-stm32fxxx-hal-drivers/>.
- [46] 4D Labs Semiconductors. *Diablo 16 Processor - Internal functions reference manual*. 2017.
- [47] 4D Systems. *gen4 HMI Display Modules*. 2017. URL: <http://www.4dsystems.com.au/products>.
- [48] 4D Labs Semiconductors. *Workshop4Pro-Smart Widgets Editor Manual*. 2017.
- [49] 4D Systems. *4D Programming Cable*. 2017. URL: <http://www.4dsystems.com.au/product/17/114/Accessories/4D-Programming-Cable/>.
- [50] 4D Labs Semiconductors. *Workshop4IDE Visi-Genie User Guide*. 2017.
- [51] 4D Labs Semiconductors. *Diablo 16 Processor - Serial command reference manual*. 2017.
- [52] Analog Devices. *OP07-Datasheet*. 2017. URL: <http://www.analog.com/media/en/technical-documentation/data-sheets/OP07.pdf>.
- [53] Analog Devices. *OP177-Datasheet*. 2017. URL: <http://www.analog.com/media/en/technical-documentation/data-sheets/OP177.pdf>.
- [54] agelectronica. *40W Triple Output Series HF40W-TL-A-Datasheet*. 2017. URL: <http://www.agspecinfo.com/pdfs/H/HF40W-TL.PDF>.

CAPÍTULO 6

Apéndices

Overlapped fiber-optic Michelson interferometers for simultaneous refractive index measurement at two sensing points

MA. DEL ROSARIO BAUTISTA-MORALES, RODOLFO MARTÍNEZ-MANUEL,* AND CARLOS A. PINEDA-ARELLANO

Centro de Investigaciones en Óptica, A.C., CIO, Aguascalientes, 20200, Mexico

*Corresponding author: rodolfom@cio.mx

Received 25 September 2017; revised 26 October 2017; accepted 30 October 2017; posted 30 October 2017 (Doc. ID 307810); published 30 November 2017

We present a fiber refractometer based on the implementation of overlapped Michelson interferometers; the refractometer allows simultaneous refractive index measurement at two sensing points for samples discrimination. The fiber refractometer uses the Fresnel reflection in each fiber tip of the overlapped interferometers to generate the interference signal. Experimental results, implementing the two sensing points, for discrimination between non-contaminated and contaminated distilled water are presented. Despite the simplicity of the presented system, resolution and repeatability of 3×10^{-4} and 5×10^{-4} are obtained in a dip and read experiment using both sensing points simultaneously for refractive index measurement. © 2017 Optical Society of America

OCIS codes: (060.2370) Fiber optics sensors; (060.4230) Multiplexing; (060.2300) Fiber measurements; (120.3180) Interferometry.

<https://doi.org/10.1364/AO.56.009515>

1. INTRODUCTION

Fiber-optic refractometers have recently been intensively investigated. They find applications in areas such as quality control in chemical and food industries, environmental monitoring, and biomedical sensing in health care applications [1–4]. They also present the advantages of small size, corrosion resistance, remote monitoring, high resolution, and, in some cases reported, multipoint capabilities.

A great number of fiber refractometers are based on the interaction of the evanescent field with the sample to be measured. In order to access the evanescent field, long-period gratings [5,6], tilted fiber Bragg gratings [7,8], and surface plasmon resonance sensing structures [9] have been employed; also, abrupt tapered fiber interferometers, such as Michelson [10], Mach–Zehnder [11–13], and Sagnac [14] configurations, have been employed to access the evanescent field. Micro-machined fiber cavities have been reported either to have the analyte to modify the optical path difference [15,16] or to have the cavity as a reference interferometer [17]. All these systems have proved good performance and can provide intensity-independent measurement. However, due to modification of the fiber in the sensing area, these refractometers are costly to manufacture and make the fiber fragile.

A simpler method to implement fiber-optic refractometers is the Fresnel reflection of the fiber tip. No modification in the sensing area is needed. However, measurements of back-reflected light intensity can be affected by either power fluctuation of the

light source or undesired curvature losses along the lead fiber. Proposed solutions to these drawbacks include the implementation of an extra fiber tip for intensity referencing [18,19], and also the use of in-line fiber interferometers, such as Fabry–Perot (F-P) interferometers, where one interferometer is used for refractive index (RI) measurements while another interferometer is not affected by the sample and therefore can be used for a light intensity referencing [16,20–22]; still in-line mirrors need to be manufactured.

For applications such as quality control and environmental monitoring, where RI measurement control is performed by comparing between samples, it is desirable to have the simplicity of a sensor at the fiber tip as well as the opportunity to perform multipoint measurements in real time.

Several configurations for multipoint fiber-optic refractometers, using Fresnel reflection at the fiber tip, have been reported, based on methods such as spatial frequency multiplexing [21,23], wavelength division multiplexing [24], short pulse optical time domain reflectometer (OTDR) using a multipoint coupler and reference channel [25], or multi-wavelength OTDR with an arrayed waveguide grating used as the wavelength division multiplexer [26]. However, for quality control, such as a go/no-go test where RI values of two samples are simultaneously measured and compared for a decision to be made, reject-accepted, these systems are underused.

For this kind of test, in this paper we present a simpler two-point RI sensor, based on overlapped Michelson interferometers.

The sensing heads are the fiber tips of standard fiber couplers, then the system is easy to implement, has low cost, and extra treatment to the fiber is not needed. Simultaneous measurements of two sensing points are achieved by implementing the frequency division multiplexing method.

Several tests for the two-sensing-point refractometer were performed, confirming the capability to discriminate between samples. Also, a test to identify distilled water from clean and contaminated samples was performed.

2. SENSOR PRINCIPLE

The proposed configuration is schematically shown in Fig. 1; a superluminescent diode is used as a broadband source (BBS); the optical source light goes through port one of a fiber circulator, which is used as an isolator and also to direct the beam through the sensing system. Port two of the circulator is then connected to a 10/90 standard fiber coupler (C1). The 10 percent coupler arm, arm 1, has the fiber tip that is left in air, named as fiber tip 1. And the 90 percent coupler arm is connected to a second coupler (C2) to split the light 50/50 for each sensing point, named fiber tip 2 and fiber tip 3, also generating arm 2 and arm 3 of the system, respectively. The light reflected from these three fiber tips generates the interference signal at C1. This signal exits through port 3 of the circulator and is detected by an optical spectrum analyzer (OSA). This signal as presented in the OSA cannot be used to discriminate between sensing points. Then, the corresponding signal processing is performed in the frequency domain to measure simultaneously the RI value at each of the two sensing points.

In this configuration, the light intensity reflects in a proportion corresponding to the index change experienced at each fiber tip, the Fresnel coefficient. From this experimental setup, and ignoring the losses in the fiber circulator and fiber couplers, the light intensity reflected at each fiber tip can be presented as

$$I_1 = \frac{1}{10} I \left(\frac{n_f - n_1}{n_f + n_1} \right)^2, \quad (1)$$

$$I_2 = \frac{9}{20} I \left(\frac{n_f - n_2}{n_f + n_2} \right)^2, \quad (2)$$

$$I_3 = \frac{9}{20} I \left(\frac{n_f - n_3}{n_f + n_3} \right)^2, \quad (3)$$

where I is the light intensity provided by the optical source; and I_1 , I_2 , and I_3 are the intensities of the beams reflected at fiber tips 1, 2, and 3, respectively. n_f is the effective index of the fiber core; and n_1 , n_2 , and n_3 are the RIs surrounding fiber tips 1, 2, and 3, respectively.

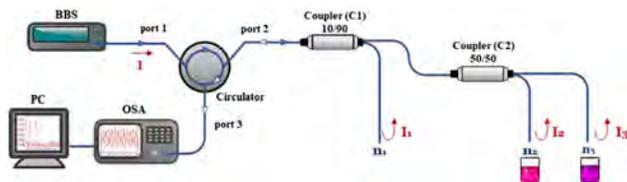


Fig. 1. Schematic diagram of the experimental setup to form overlapped fiber Michelson interferometers.

Also, due to the short length of the fiber system, 4–5 m of standard single-mode fiber (Corning SMF-28), attenuation in the fiber was not considered in the equations.

The beams reflected from the fiber tips generate overlapped fiber-optic Michelson interferometers: interferometer 1 between arms 1 and 2, interferometer 2 between arms 1 and 3, and interferometer 3, not used in this system, between arms 2 and 3. The first two overlapped Michelson interferometers have fiber tip 1 as a common tip, leaving tip 2 and tip 3 as the two sensing points, being independent between one another. The interference signal generated at these Michelson interferometers can be written as follows:

$$I_{12} = I_1 + I_2 + 2\sqrt{I_1 + I_2} \cos \Delta\phi_{12}, \quad (4)$$

$$I_{13} = I_1 + I_3 + 2\sqrt{I_1 + I_3} \cos \Delta\phi_{13}, \quad (5)$$

$$I_{23} = I_2 + I_3 + 2\sqrt{I_2 + I_3} \cos \Delta\phi_{23}, \quad (6)$$

where $\Delta\phi_{12} = \frac{2\pi}{\lambda} 2n_f l_{12}$, $\Delta\phi_{13} = \frac{2\pi}{\lambda} 2n_f l_{13}$, and $\Delta\phi_{23} = \frac{2\pi}{\lambda} 2n_f l_{23}$ are the phase differences at the Michelson interferometers 1, 2, and 3, respectively; and l_{12} , l_{13} , and l_{23} are the fiber length differences for each of these three overlapped interferometers, respectively. The signals from these interferometers arrive at the OSA and cannot be discriminated from one another. However, each of these three signals can be separated in the frequency domain if the optical path difference is different for each interferometer, as presented in the simulation results in Fig. 2.

The complex components $|A_{12}|$ and $|A_{13}|$, corresponding to interferometers 1 and 2, respectively, are used for the RI calculation. Fiber tips 2 and 3 are implemented as sensing points. These amplitudes of the frequency components are defined from Eqs. (1), (2), (3), (4), and (5) as

$$|A_{12}| = 2\sqrt{\left(\frac{1}{10}\right) I \left(\frac{n_f - n_1}{n_f + n_1}\right)^2 * \left(\frac{9}{20}\right) I \left(\frac{n_f - n_2}{n_f + n_2}\right)^2}, \quad (7)$$

then we can write

$$|A_{12}| = \frac{6}{5\sqrt{8}} \left(\frac{n_f - n_1}{n_f + n_1}\right) \left(\frac{n_f - n_2}{n_f + n_2}\right) I, \quad (8)$$

and in the same way,

$$|A_{13}| = \frac{6}{5\sqrt{8}} \left(\frac{n_f - n_1}{n_f + n_1}\right) \left(\frac{n_f - n_3}{n_f + n_3}\right) I. \quad (9)$$

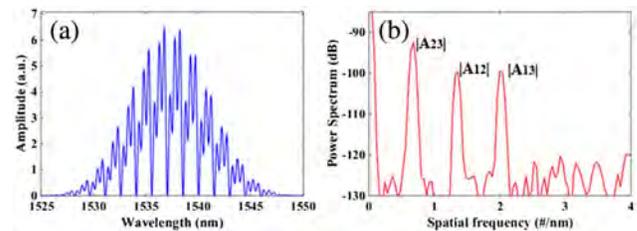


Fig. 2. Simulated signal for three overlapped Michelson interferometers, using air as the outer medium, and having $l_{12} = 1.10$ mm, $l_{13} = 1.65$ mm, and then $l_{23} = 0.55$ mm; (a) spectra of the overlapped interferometers and (b) frequency components of these interferometers, clearly separated.

Performing a measurement with a medium of a well-known RI n_o , such as air, we obtain the reference amplitudes, $A_{\text{ref}12}$ and $A_{\text{ref}13}$; then, dipping fiber tips 2 and 3 with the unknown RI samples n_2 and n_3 and leaving fiber tip 1 in air ($n_1 = n_o$), we can write

$$\frac{|A_{\text{ref}12}|}{|A_{12}|} = \frac{\frac{6}{5\sqrt{8}} \left(\frac{n_f - n_o}{n_f + n_o}\right)^2 I}{\frac{6}{5\sqrt{8}} \left(\frac{n_f - n_0}{n_f + n_0}\right) \left(\frac{n_f - n_2}{n_f + n_2}\right) I} = \frac{\frac{n_f - n_o}{n_f + n_o}}{\frac{n_f - n_2}{n_f + n_2}}, \quad (10)$$

and then,

$$n_2 = n_f \frac{\left[\frac{|A_{\text{ref}12}|}{|A_{12}|} (n_f + n_0) - n_f + n_0 \right]}{\left[\frac{|A_{\text{ref}12}|}{|A_{12}|} (n_f + n_0) + n_f - n_0 \right]}, \quad (11)$$

having the same process for interferometer 2:

$$n_3 = n_f \frac{\left[\frac{|A_{\text{ref}13}|}{|A_{13}|} (n_f + n_0) - n_f + n_0 \right]}{\left[\frac{|A_{\text{ref}13}|}{|A_{13}|} (n_f + n_0) + n_f - n_0 \right]}. \quad (12)$$

As can be seen from Eqs. (11) and (12), after a measurement is taken with all the fiber tips of the system in a medium of a well-known RI, and with the two sensing fiber tips submerged in the sample under test, we are able to calculate the unknown RI surrounding fiber tips 2 and 3. Also, this procedure allows for intensity-independent measurements, and the coupling ratio of the couplers is removed as well.

3. EXPERIMENTAL RESULTS

The proposed refractometer based on overlapped Michelson interferometers was tested by measuring RIs of several liquid samples simultaneously at the two sensing points, fiber tips 2 and 3. A typical spectrum of the detected signal is shown in Fig. 3(a) where isopropyl alcohol was used as the outer medium in both sensing points. In order to reduce boundary effects (side lobes in the frequency domain), the Blackman window was applied. The frequency spectrum of the detected signal is shown in Fig. 3(b). As it can be seen, three principal maxima appear that are associated with each Michelson interferometer, from lower to higher frequency: interferometer 3 ($l_{23} = 0.55$ mm), interferometer 1 ($l_{12} = 1.10$ mm), and interferometer 2 ($l_{13} = 1.65$ mm). These fiber length differences were calculated from the spectra generated in each interferometer after cleaving the fiber tips. These lengths allow working in low frequency range.

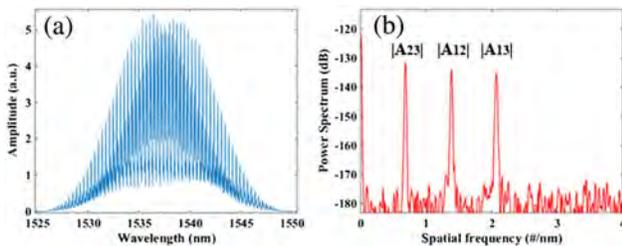


Fig. 3. (a) Measured interferogram generated by three overlapped Michelson interferometers, with isopropyl alcohol as the outer medium in the two sensing points, and a Blackman window used to reduce side lobes. (b) Power spectrum of the measured interferogram.

In our experiments, the liquid samples tested were distilled water, methanol, acetone, isopropyl alcohol, chloroform, and contaminated distilled water, covering the range of RIs between 1.315 and 1.432. In all the measurements, the OSA was set a resolution value of 0.03 nm. First, the two sensing fiber tips were submerged into each sample at the same time; the spectra of these measurements are presented in Fig. 4(a). As expected, the frequency-peak amplitude of each interferometer decreases as the RI value of the sample approaches the RI value of the core of the fiber. Over the RI range from 1 to 1.432, the frequency-peak amplitude variation is 33.1 dB, which corresponds to a RI sensitivity of ~ 76 dB/RIU. In order to prove the two sensing points work independently, the two sensing fiber tips were used to measure simultaneously two samples with different RIs and then were swapped around. The spectrum is shown in Fig. 4(b).

As presented in Eqs. (10), (11), and (12), in order to calculate the RI value of each sample, first a reference measurement with the three fiber tips surrounded by a well-known RI medium is taken; in this experiment, we used air as the outer medium. Then the sensing fiber tips are dipped into the sample under test, and 10 spectra measurements were taken. For each

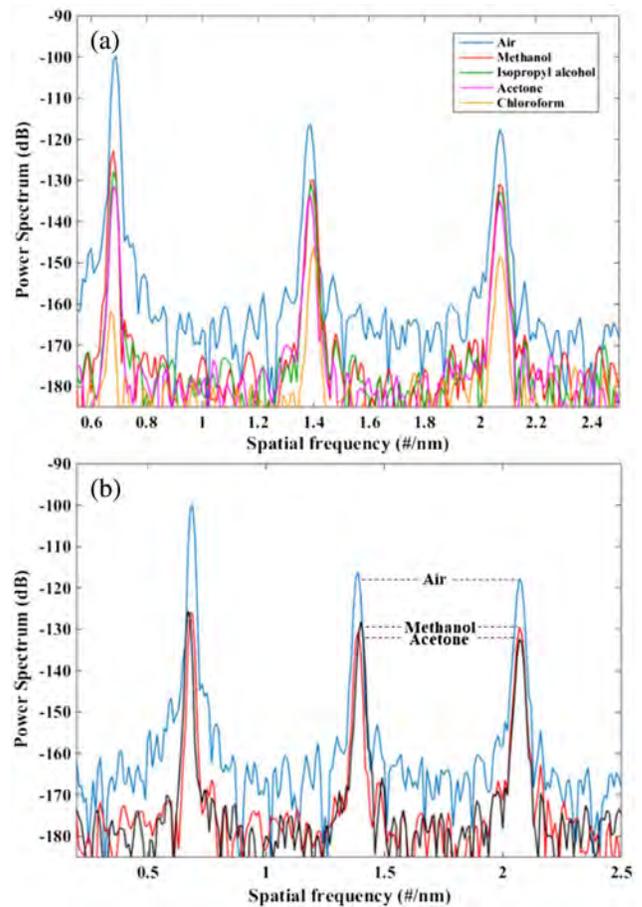


Fig. 4. Frequency spectrum of three overlapped Michelson interferometers, using fiber tip 2 and fiber tip 3 to run the measurements in the dip and read mode in different samples. (a) Two tips simultaneously submerged in each sample, chloroform, isopropyl alcohol, acetone, methanol, and distilled water. (b) Two tips submerged in two samples of different RIs, methanol and acetone, in one measurement and then swapped around in a second measurement.

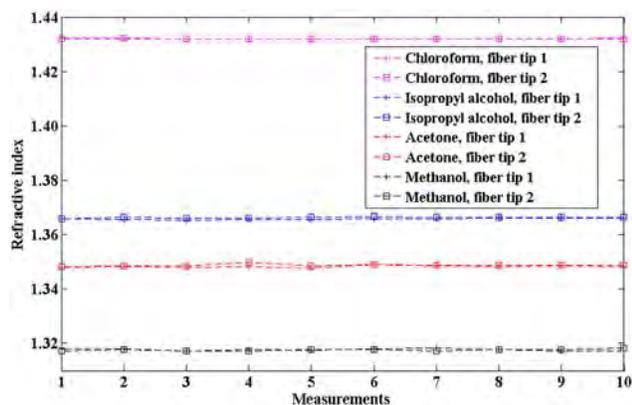


Fig. 5. Results of 10 independent simultaneous measurements; fiber tips submerged in each sample, chloroform, isopropyl alcohol, acetone, methanol, and distilled water; the standard error obtained was 8×10^{-5} .

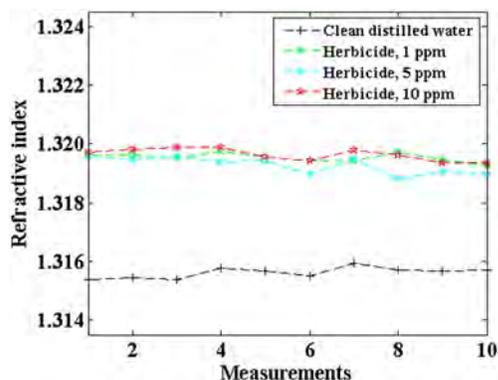


Fig. 6. Refractive index values of samples of distilled water contaminated with different atrazine herbicide concentrations.

measurement, the spectra obtained are transformed into the frequency domain to identify the frequency components for each Michelson interferometer, and a frequency component per sensing point, named fiber tip 2 and fiber tip 3, is defined. Then, per measurement, a ratio between the amplitudes of a reference measurement and a sensing measurement is taken as explained above; after the calibration process is completed, we calculated the RI values presented in Fig. 5. Using these values, a resolution of the 3×10^{-4} was obtained. The experiment was repeated five times using only isopropyl alcohol to calculate a repeatability of 5×10^{-4} .

An experiment was also performed to implement the system in a go/no-go test using clean and contaminated distilled water. For this quality control experiment with the two sensing points, we used one sensing point as a control sensor, keeping it in distilled water, while the other sensing point was used in the sample under test, distilled water with different concentrations of atrazine herbicide, 1, 5, and 10 ppm. The RI value obtained for samples with different concentrations of atrazine herbicide, after the signal processing, are presented in Fig. 6.

With these results, we confirm that it is possible to discriminate whether a sample is contaminated or not. The system was

able to measure RI change at small concentrations, with a resolution of 3×10^{-4} .

4. CONCLUSIONS

We presented a fiber refractometer capable to simultaneously measure in two sensing points. The refractometer is based on overlapped Michelson interferometers formed by two couplers connected in cascade; all components are standard SMF-28 fiber based. The system was tested in several samples of different RI values. Despite the simplicity of the system, resolution and repeatability of 3×10^{-4} and 5×10^{-4} were demonstrated in a dip and read experiment using both sensing points simultaneously. A further experiment was performed to successfully discriminate between non-contaminated and contaminated distilled water.

Funding. Consejo Nacional de Ciencia y Tecnología (CONACYT) (2015-01-1651, 429767).

Acknowledgment. This work was partially supported by the Problemas Nacionales project (Diseño y Construcción de Potabilizador Integral Solar de Agua para Comunidades Rurales), funded by the CONACYT of México. Authors also acknowledge and thank CONACYT for the postgraduate scholarship provided to María del Rosario Bautista-Morales.

REFERENCES

1. A. Ricciardi, A. Crescitelli, P. Vaiano, G. Quero, M. Consales, M. Pisco, E. Esposito, and A. Cusano, "Lab-on-fiber technology: a new vision for chemical and biological sensing," *Analyst* **140**, 8068–8079 (2015).
2. M. Pospíšilová, G. Kuncová, and J. Trögl, "Fiber-optic chemical sensors and fiber-optic bio-sensors," *Sensors* **15**, 25208–25259 (2015).
3. Q. Shi, L. Ying, L. Wang, B. Peng, and C. Ying, "A method of the detection of marine pollution based on the measurement of refractive index," *Appl. Mech. Mater.* **551**, 347–352 (2014).
4. R. Amin, A. Kulkarni, T. Kim, and S. H. Park, "DNA thin film coated optical fiber biosensor," *Curr. Appl. Phys.* **12**, 841–845 (2012).
5. G. Quero, A. Crescitelli, D. Paladino, M. Consales, A. Buosciolo, M. Giordano, A. Cutolo, and A. Cusano, "Evanescent wave long-period fiber grating within D-shaped optical fibers for high sensitivity refractive index detection," *Sens. Actuators B* **152**, 196–205 (2011).
6. D. W. Kim, Y. Zhang, K. L. Cooper, and A. Wang, "In-fiber reflection mode interferometer based on a long-period grating for external refractive-index measurement," *Appl. Phys.* **44**, 5368–5373 (2005).
7. J. Zheng, X. Dong, P. Zu, L. Y. Shao, C. C. Chan, Y. Cui, and P. P. Shum, "Magnetic field sensor using tilted fiber grating interacting with magnetic fluid," *Opt. Express* **21**, 17863–17868 (2013).
8. X. Chen, J. Xu, X. Zhang, T. Guo, and B.-O. Guan, "Wide range refractive index measurement using a multi-angle tilted fiber Bragg grating," *IEEE Photon. Technol. Lett.* **29**, 719–722 (2017).
9. A. González-Cano, M. Cruz-Navarrete, Ó. Esteban, and N. Díaz-Herrera, "Plasmonic sensors based on doubly-deposited tapered optical fibers," *Sensors* **14**, 4791–4805 (2014).
10. Z. Tian, S. S. H. Yam, and H. P. Loock, "Refractive index sensor based on an abrupt taper Michelson interferometer in a single-mode fiber," *Opt. Lett.* **33**, 1105–1107 (2008).
11. J. Yang, L. Jiang, S. Wang, B. Li, M. Wang, H. Xiao, Y. Lu, and H. Tsai, "High sensitivity of taper-based Mach-Zehnder interferometer embedded in a thinned optical fiber for refractive index sensing," *Appl. Opt.* **50**, 5503–5507 (2011).
12. P. Lu, J. Harris, Y. Xu, Y. Lu, L. Chen, and X. Bao, "Simultaneous refractive index and temperature measurements using a tapered bend-resistant fiber interferometer," *Opt. Lett.* **37**, 4567–4569 (2012).

13. Z. Tian, S. S. H. Yam, and H. P. Loock, "Single-mode fiber refractive index sensor based on core-offset attenuators," *IEEE Photon. Technol. Lett.* **20**, 1387–1389 (2008).
14. S. Pu, L. Luo, J. Tang, L. Mao, and X. Zeng, "Ultrasensitive refractive-index sensors based on tapered fiber coupler with Sagnac loop," *IEEE Photon. Technol. Lett.* **28**, 1073–1076 (2016).
15. C. R. Liao, T. Y. Hu, and D. N. Wang, "Optical fiber Fabry–Perot interferometer cavity fabricated by femtosecond laser micromachining and fusion splicing for refractive index sensing," *Opt. Express* **20**, 22813–22818 (2012).
16. S. Pevec and D. Donlagic, "High resolution, all-fiber, micro-machined sensor for simultaneous measurement of refractive index and temperature," *Opt. Express* **22**, 16241–16253 (2014).
17. Z. L. Ran, Y. J. Rao, W. J. Liu, X. Liao, and K. S. Chiang, "Laser-micromachined Fabry–Perot optical fiber tip sensor for high-resolution temperature independent measurement of refractive index," *Opt. Express* **16**, 2252–2263 (2008).
18. C.-B. Kim and C. B. Su, "Measurement of the refractive index of liquids at 1.3 and 1.5 micron using a fibre optic Fresnel ratio meter," *Meas. Sci. Technol.* **15**, 1683–1686 (2004).
19. H. Su and X. G. Huang, "Fresnel-reflection-based fiber sensor for online measurement of solute concentration in solutions," *Sens. Actuators B Chem.* **126**, 579–582 (2007).
20. H. Y. Choi, G. Mudhana, K. S. Park, U.-C. Paek, and B. H. Lee, "Cross-talk free and ultra-compact fiber optic sensor for simultaneous measurement of temperature and refractive index," *Opt. Express* **18**, 141–149 (2010).
21. K. S. Park, Y. H. Kim, J. B. Eom, S. J. Park, M.-S. Park, J.-H. Jang, and B. H. Lee, "Compact and multiplexible hydrogen gas sensor assisted by self-referencing technique," *Opt. Express* **19**, 18190–18198 (2011).
22. M. G. Shlyagin, R. M. Manuel, and Ó. Esteban, "Optical-fiber self-referenced refractometer based on Fresnel reflection at the fiber tip," *Sens. Actuators B Chem.* **178**, 263–269 (2013).
23. L. Liu, Y. Gong, Y. Wu, T. Zhao, H.-J. Wu, and Y.-J. Rao, "Spatial frequency multiplexing of fiber-optic interferometric refractive index sensors based on graded-index multimode fibers," *Sensors* **12**, 12377–12385 (2012).
24. C.-L. Zhao, J. Li, S. Zhang, Z. Zhang, and S. Jin, "Simple fresnel reflection-based optical fiber sensor for multipoint refractive index measurement using an AWG," *IEEE Photon. Technol. Lett.* **25**, 606–608 (2013).
25. A. Basgumus, F. E. Durak, A. Altuncu, and G. Yılmaz, "A universal and stable all-fiber refractive index sensor system," *IEEE Photon. Technol. Lett.* **28**, 171–174 (2016).
26. K. Yüksel, "Optical fiber sensor system for remote and multi-point refractive index measurement," *Sens. Actuators A Phys.* **250**, 29–34 (2016).

6.1. Código para la tarjeta STM32F767ZI Núcleo 144

```

1
2 *****
3 * File Name      : main.c
4 * Description    : Program for the STM32F767ZI board
5 * Author        : Maria del Rosario Bautista Morales
6 *****
7 *
8 * COPYRIGHT(c) 2017 STMicroelectronics
9 *
10 *****
11
12 /* Includes -----*/
13 #include "main.h"
14 #include "stm32f7xx_hal.h"
15 /* USER CODE BEGIN Includes */
16 #include "tm_stm32f4_fft.h"
17 #include "defines.h"
18 #include "math.h"
19 /* -----Definicion de Variables para la Transformada de Fourier
20  ----- */
21 #define SAMPLES (8192)          /* 256 real party and 256 imaginary
22  parts */
23 #define FFT_SIZE (SAMPLES / 2)
24 /* USER CODE END Includes */
25 /* Private variables -----*/
26 ADC_HandleTypeDef hadc1;
27 DAC_HandleTypeDef hdac;
28 DMA_HandleTypeDef hdma_dac1;
29 TIM_HandleTypeDef htim2;
30 UART_HandleTypeDef huart6;
31 /* USER CODE BEGIN PV */
32 /*-----Vector que contiene valores de la rampa de la senial de modulacion
33  -----*/
34 const uint16_t modulationsignal[500]={62, 62, 63, 63, 63, 63, 64, 64, 64,
35 65, 65, 65, 65, 66, 66, 66, 67, 67, 67, 67, 68, 68, 68, 68, 69, 69,
36 69, 70, 70, 70, 70, 71, 71, 71, 72, 72, 72, 72, 73, 73, 73, 74, 74,
37 74, 74, 75, 75, 75, 75, 76, 76, 76, 77, 77, 77, 77, 78, 78, 78, 79,
38 79, 79, 79, 80, 80, 80, 81, 81, 81, 81, 82, 82, 82, 82, 83, 83, 83,
39 84, 84, 84, 84, 85, 85, 85, 86, 86, 86, 86, 87, 87, 87, 88, 88, 88,
40 88, 89, 89, 89, 89, 90, 90, 90, 91, 91, 91, 91, 92, 92, 92, 93, 93,
41 93, 93, 94, 94, 94, 94, 95, 95, 95, 96, 96, 96, 96, 97, 97, 97, 98,
42 98, 98, 98, 99, 99, 99, 100, 100, 100, 100, 101, 101, 101, 101, 102,
43 102, 102, 103, 103, 103, 103, 104, 104, 104, 105, 105, 105, 105, 106,
44 106, 106, 107, 107, 107, 107, 108, 108, 108, 108, 109, 109, 109,
45 110, 110, 110, 110, 111, 111, 111, 112, 112, 112, 112, 113, 113, 113,
46 114, 114, 114, 114, 115, 115, 115, 115, 116, 116, 116, 117, 117, 117,
47 117, 118, 118, 118, 119, 119, 119, 119, 120, 120, 120, 121, 121,
48 121, 121, 122, 122, 122, 122, 123, 123, 123, 124, 124, 124, 124, 125,
49 125, 125, 126, 126, 126, 126, 127, 127, 127, 128, 128, 128, 128, 129,

```

```

129, 129, 129, 130, 130, 130, 131, 131, 131, 131, 132, 132, 132,
133, 133, 133, 133, 134, 134, 134, 135, 135, 135, 135, 136, 136, 136,
136, 137, 137, 137, 138, 138, 138, 138, 139, 139, 139, 140, 140, 140,
140, 141, 141, 141, 141, 142, 142, 142, 143, 143, 143, 143, 144,
144, 144, 145, 145, 145, 145, 146, 146, 146, 147, 147, 147, 147, 148,
148, 148, 148, 149, 149, 149, 150, 150, 150, 150, 151, 151, 151, 152,
152, 152, 152, 153, 153, 153, 154, 154, 154, 154, 155, 155, 155,
155, 156, 156, 156, 157, 157, 157, 157, 158, 158, 158, 159, 159, 159,
159, 160, 160, 160, 161, 161, 161, 161, 162, 162, 162, 162, 163, 163,
163, 164, 164, 164, 164, 165, 165, 165, 166, 166, 166, 166, 167,
167, 167, 168, 168, 168, 168, 169, 169, 169, 169, 170, 170, 170, 171,
171, 171, 171, 172, 172, 172, 173, 173, 173, 173, 174, 173, 172, 171,
169, 168, 167, 166, 165, 164, 163, 162, 161, 159, 158, 157, 156,
155, 154, 153, 152, 150, 149, 148, 147, 146, 145, 144, 143, 141, 140,
139, 138, 137, 136, 135, 134, 133, 131, 130, 129, 128, 127, 126, 125,
124, 122, 121, 120, 119, 118, 117, 116, 115, 114, 112, 111, 110,
109, 108, 107, 106, 105, 103, 102, 101, 100, 99, 98, 97, 96, 94, 93,
92, 91, 90, 89, 88, 87, 86, 84, 83, 82, 81, 80, 79, 78, 77, 75, 74,
73, 72, 71, 70, 69, 68, 67, 65, 64, 63, 62};
32 const float32_t nf=1.4446;
33 const float32_t nA=1.00029;
34 /* Private variables
-----*/
35 TM_FFT_F32_t FFT;
36 float32_t Input[SAMPLES]; /*!< Input buffer is always 2 * FFT_SIZE */
37 float32_t Reference[FFT_SIZE]; /*!< Output buffer is always FFT_SIZE */
38 float32_t SampleUT[FFT_SIZE];
39 float32_t MaxP[2];
40 int16_t PicP[2];
41 /* USER CODE END PV */
42 /* Private function prototypes
-----*/
43 void SystemClock_Config(void);
44 static void MX_GPIO_Init(void);
45 static void MX_DMA_Init(void);
46 static void MX_USART6_UART_Init(void);
47 static void MX_ADC1_Init(void);
48 static void MX_DAC_Init(void);
49 static void MX_TIM2_Init(void);
50 /* USER CODE BEGIN PFP */
51 /* Private function prototypes
-----*/
52 /* USER CODE END PFP */
53 /* USER CODE BEGIN 0 */
54 uint8_t t=0, N=0;
55 uint16_t vector[8192];
56 uint8_t bandinicio=0, contaCi=0;
57 uint32_t getUs(void) {
58 uint32_t usTicks = HAL_RCC_GetSysClockFreq() / 1000000;
59 register uint32_t ms, cycle_cnt;
60 do {
61 ms = HAL_GetTick();

```

```

62 cycle_cnt = SysTick->VAL;
63 } while (ms != HAL_GetTick());
64 return (ms * 1000) + (usTicks * 1000 - cycle_cnt) / usTicks;
65 }
66 void delayUs(uint16_t micros) {
67 uint32_t start = getUs();
68 while (getUs()-start < (uint32_t) micros) {
69 asm("nop");
70 }
71 }
72 /*Adquisicion del vector que almacena la senial de los sensores */
73 uint16_t ADC_read(){
74 uint16_t auxh=0;
75 HAL_ADC_Start(&hadc1);
76 HAL_ADC_PollForConversion(&hadc1, 10);
77 auxh = HAL_ADC_GetValue(&hadc1);
78 return auxh;
79 }
80 void obtenerV(void){
81 uint16_t aux=0;
82 //long auxa=0;
83 long u=0;
84 bandinicio=1;
85 while(1){
86 if (contaCi==1){
87 aux=ADC_read();
88 if (u!=0){
89 // if ((aux<auxa+50)&&(aux>auxa-50)){
90 vector[u]=aux;
91 // auxa=aux;
92 //}
93 //else{vector[u]=auxa;
94 //}
95 }
96 else{
97 vector[0]=aux;
98 // auxa=aux;
99 }
100 delayUs(488);
101 u++;
102 if (u==SAMPLES) break;
103 } else if (contaCi>1) {break;}
104 }
105 contaCi=0; bandinicio=0;
106 }
107 /* Funcion que determina puntos maximos en la transformada de Fourier*/
108 void maxiPicos(){
109 float32_t auss=0, M1=0, M2=0;
110 int16_t p1=0,p2=0;
111 for (int16_t p=5; p< 20;p++){
112 auss=FFT.Output[p];
113 if (auss>M1){

```

```

114     M1=auss;
115     p1=p;
116 }
117 }
118 for(int16_t p=21; p< 40;p++){
119     auss=FFT.Output[p];
120     if(auss>M2){
121         M2=auss;
122         p2=p;
123     }
124 }
125 MaxP[0]=M1;
126 MaxP[1]=M2;
127 PicP[0]=p1;
128 PicP[1]=p2;
129 }
130 void transformada(void){
131     long u=0;
132     float32_t conv=0.0;
133     do{
134         conv=(vector[u]*3.3)/4095;
135         u++;
136     }while(!TM_FFT_AddToBuffer(&FFT, conv));
137     TM_FFT_Process_F32(&FFT);
138     HAL_Delay(10);
139 }
140 void obtenerPicos(float32_t *muestra, uint8_t ref){
141     float32_t max=0, max1=0;
142     for(int j=0;j<10;j++){
143         obtenerV();
144         transformada();
145         //////////////////////////////////////
146         if(ref==1){
147             maxiPicos();
148         }
149         else{
150             max=FFT.Output[PicP[0]];
151             max1=FFT.Output[PicP[1]];
152         }
153         //////////////////////////////////////
154         if(j>0){
155             max=(max+MaxP[0])/2;
156             max1=(max1+MaxP[1])/2;
157         }
158         else{
159             max=MaxP[0];
160             max1=MaxP[1];
161         }
162     }
163     muestra[0]=max;
164     muestra[1]=max1;
165 }

```

```

166 void HAL_DAC_ConvCpltCallbackCh1 (DAC_HandleTypeDef *hdac)
167 {
168     HAL_GPIO_TogglePin(GPIOB, LD2_Pin);
169     if (bandinicio==1){
170         contaCi++;
171         // if (contaCi==2){
172         //     contaCi=0;
173         // }
174     }
175 }
176 /* USER CODE END 0 */
177 int main(void)
178 {
179     /* USER CODE BEGIN 1 */
180     uint8_t varc=0;
181     /*Definicion de comandos para la comunicacion pantalla tarjeta (seleccion
182     de componentes)*/
183     uint8_t lcd1[6]={0x01,0x0F,0x00,0x33,0x6C,0x51};
184     uint8_t lcd2[6]={0x01,0x0F,0x01,0x38,0x6E,0x59};
185     uint8_t medir[6]={0x07,0x1E,0x02,0x00,0x00,0x1B};
186     //uint8_t inicio[6]={0x07, 0x1E, 0x00, 0x00, 0x00, 0x19};
187     //uint8_t salir[6]={0x07, 0x1E, 0x01, 0x00, 0x00, 0x18};
188     //uint8_t nuevaM[6]={0x07,0x1E,0x03,0x00,0x00,0x1A};
189     uint8_t boton[6]={0x00,0x00,0x00,0x00,0x00,0x00};
190     uint8_t lcd0[6]={0x01,0x0F,0x00,0x00,0x00,0x0F};
191     uint8_t Form0[6]={0x01, 0x0A, 0x00, 0x00, 0x00, 0x0B};
192     uint8_t Form1[6]={0x01, 0x0A, 0x01, 0x00, 0x00,0x0A};
193     uint16_t indice0=15000;
194     uint16_t indice1=15000;
195     uint8_t cheksum=0;
196     uint8_t bp=0;
197     uint8_t status=0;
198     float32_t maxiA[2]={0.0,0.0};
199     float32_t maxiM[2]={0.0,0.0};
200     float32_t C=0.0;
201     float32_t indiceAux=0.0;
202     uint8_t bandera=0;
203     /* USER CODE END 1 */
204     /* MCU Configuration
205     _____*/
206     /* Reset of all peripherals, Initializes the Flash interface and the
207     SysTick. */
208     HAL_Init();
209     /* USER CODE BEGIN Init */
210     /* USER CODE END Init */
211     /* Configure the system clock */
212     SystemClock_Config();
213     /* USER CODE BEGIN SysInit */
214     /* USER CODE END SysInit */
215     /* Initialize all configured peripherals */
216     MX_GPIO_Init();
217     MX_DMA_Init();

```

```

215 MX_USART6_UART_Init();
216 MX_ADC1_Init();
217 MX_DAC_Init();
218 MX_TIM2_Init();
219 /* USER CODE BEGIN 2 */
220 status=TM_FFT_Init_F32(&FFT, FFT_SIZE, 0);
221 if (status==0) HAL_GPIO_WritePin(LD3_GPIO_Port, LD3_Pin,1);
222 else HAL_GPIO_WritePin(LD3_GPIO_Port, LD3_Pin,0);
223 HAL_TIM_Base_Start_IT(&htim2);
224 HAL_DAC_Start(&hdac,DAC_CHANNEL_1);
225 HAL_DAC_Start_DMA(&hdac, DAC_CHANNEL_1, (uint32_t*)modulationsignal,
226 500, DAC_ALIGN_12B_R);
227 /* USER CODE END 2 */
228 /* Infinite loop */
229 /* USER CODE BEGIN WHILE */
230 while (1)
231 {
232 /* USER CODE END WHILE */
233 /* USER CODE BEGIN 3 */
234 // voltaje=ADC_read();
235 // vector[0]= voltaje;
236 /*Select Case Principal de Control para la ejecucion del algoritmo de la
237 tarjeta*/
238 while (boton[0]!=medir[0]&& boton[1]!=medir[1]) {
239 HAL_UART_Receive(&huart6 , boton ,6 ,250);
240 }
241 bp=boton [2];
242 switch (bp) {
243 /*Boton inicio , captura de valores de referencia*/
244 case 0x00 :
245 TM_FFT_SetBuffers_F32(&FFT, Input , Reference);
246 HAL_UART_Transmit(&huart6 , Form1 ,6 ,250);
247 HAL_Delay(10);
248 lcd0 [2]=0x00;
249 lcd0 [5]=0x0E;
250 HAL_UART_Transmit(&huart6 , lcd0 ,6 ,250);
251 varc=0;
252 while ( varc!=0x06) {
253 HAL_UART_Receive(&huart6 ,&varc ,1 ,250);
254 }
255 lcd0 [2]=0x01;
256 lcd0 [5]=0x0F;
257 HAL_UART_Transmit(&huart6 , lcd0 ,6 ,250);
258 varc=0;
259 while ( varc!=0x06) {
260 HAL_UART_Receive(&huart6 ,&varc ,1 ,250);
261 }
262 HAL_GPIO_WritePin(Buzzer_GPIO_Port ,Buzzer_Pin ,1);
263 obtenerPicos (maxiA ,1);
264 HAL_GPIO_WritePin(Buzzer_GPIO_Port ,Buzzer_Pin ,0);
265 bandera=0;
266 break;

```

```

265 /* Boton Salir , regresa a la pantalla inicio*/
266 case 0x01: HAL_UART_Transmit(&huart6 ,Form0,6,250); break;
267 /*Boton medir , captura los valores necesarios para el calculo del IR y lo
muestra en pantalla*/
268 case 0x02: if (bandera==0){
269     HAL_GPIO_WritePin(GPIOB, LD2_Pin,1);
270     HAL_GPIO_WritePin(Buzzer_GPIO_Port, Buzzer_Pin ,1);
271     obtenerPicos (maxiM,0);
272     HAL_GPIO_WritePin(Buzzer_GPIO_Port, Buzzer_Pin ,0);
273     C =((nf-nA)/(nf+nA))/(maxiA[0]/maxiM[0]);
274     indiceAux=nf*((1-C)/(1+C));
275     indice0=indiceAux*10000;
276     lcd1[3]=indice0/258;
277     lcd1[4]=indice0&0xFF;
278     for (int k=0;k<5;k++){
279         if (k==0) cheksum=lcd1[0];
280         else cheksum^=lcd1[k];
281     }
282     lcd1[5]=cheksum;
283     HAL_UART_Transmit(&huart6 , lcd1 ,6 ,250);
284     varc=0;
285     while (varc!=0x06){
286         HAL_UART_Receive(&huart6 ,&varc ,1 ,250);
287     }
288     C =((nf-nA)/(nf+nA))/(maxiA[1]/maxiM[1]);
289     indiceAux=nf*((1-C)/(1+C));
290     indice1=indiceAux*10000;
291     lcd2[3]=indice1/258;
292     lcd2[4]=indice1&0xFF;
293     for (int k=0;k<5;k++){
294         if (k==0) cheksum=lcd2[0];
295         else cheksum^=lcd2[k];
296     }
297     lcd2[5]=cheksum;
298     HAL_UART_Transmit(&huart6 , lcd2 ,6 ,250);
299     varc=0;
300     while (varc!=0x06){
301         HAL_UART_Receive(&huart6 ,&varc ,1 ,250);
302     }
303     bandera=1;
304 }
305 break;
306 /*Boton de nueva muestra , permite repetir la medicion conservando la
misma referencia*/
307 case 0x03: lcd0[2]=0x00;
308     lcd0[5]=0x0E;
309     HAL_UART_Transmit(&huart6 , lcd0 ,6 ,250);
310     varc=0;
311     while (varc!=0x06){
312         HAL_UART_Receive(&huart6 ,&varc ,1 ,250);
313     }
314     lcd0[2]=0x01;

```

```

315     lcd0[5]=0x0F;
316     HAL_UART_Transmit(&huart6 , lcd0 , 6 , 250);
317     varc=0;
318     while (varc!=0x06) {
319         HAL_UART_Receive(&huart6 ,&varc , 1 , 250);
320     }
321     bandera=0;
322     break;
323 }
324 /*Inicializacion del comando auxiliar para deteccion de la pantalla*/
325 for (int i=0;i<6;i++){
326     boton[i]=0x00;
327 }
328     /*     varc=0;
329     while (varc!='o') {
330         HAL_UART_Receive(&huart6 ,&varc , 1 , 250);
331     }
332     HAL_UART_Transmit(&huart6 , cdn , 8 , 250);
333 HAL_Delay(500);
334 */
335 }
336 /* USER CODE END 3 */
337 }
338 /** System Clock Configuration
339 */
340 /* — CODIGO DE INICIALIZACION DE TODAS LAS LIBRERIAS HAL EMPLEADAS EN LA
341     TARJETA — */
342 void SystemClock_Config (void)
343 {
344     RCC_OscInitTypeDef RCC_OscInitStruct;
345     RCC_ClkInitTypeDef RCC_ClkInitStruct;
346     RCC_PeriphCLKInitTypeDef PeriphClkInitStruct;
347     /**Configure the main internal regulator output voltage
348     */
349     __HAL_RCC_PWR_CLK_ENABLE();
350     __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
351     /**Initializes the CPU, AHB and APB busses clocks
352     */
353     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
354     RCC_OscInitStruct.HSISState = RCC_HSI_ON;
355     RCC_OscInitStruct.HSICalibrationValue = 16;
356     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
357     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
358     RCC_OscInitStruct.PLL.PLLM = 8;
359     RCC_OscInitStruct.PLL.PLLN = 216;
360     RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
361     RCC_OscInitStruct.PLL.PLLQ = 2;
362     if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
363     {
364         _Error_Handler(__FILE__, __LINE__);
365     }
366     /**Activate the Over-Drive mode

```

```

366  */
367  if (HAL_PWREx_EnableOverDrive() != HAL_OK)
368  {
369      _Error_Handler(__FILE__, __LINE__);
370  }
371  /**Initializes the CPU, AHB and APB busses clocks
372  */
373  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
374                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
375  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
376  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
377  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV16;
378  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
379  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_7) != HAL_OK)
380  {
381      _Error_Handler(__FILE__, __LINE__);
382  }
383  PeriphClkInitStruct.PeriphClockSelection = RCC_PERIPHCLK_USART6;
384  PeriphClkInitStruct.Usart6ClockSelection = RCC_USART6CLKSOURCE_PCLK2;
385  if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInitStruct) != HAL_OK)
386  {
387      _Error_Handler(__FILE__, __LINE__);
388  }
389  /**Configure the SysTick interrupt time
390  */
391  HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);
392  /**Configure the SysTick
393  */
394  HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);
395  /* SysTick_IRQn interrupt configuration */
396  HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
397 }
398 /* ADC1 init function */
399 static void MX_ADC1_Init(void)
400 {
401     ADC_ChannelConfTypeDef sConfig;
402     /**Configure the global features of the ADC (Clock, Resolution, Data
403     Alignment and number of conversion)
404     */
405     hadc1.Instance = ADC1;
406     hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV4;
407     hadc1.Init.Resolution = ADC_RESOLUTION_12B;
408     hadc1.Init.ScanConvMode = DISABLE;
409     hadc1.Init.ContinuousConvMode = DISABLE;
410     hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
411     hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
412     hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
413     hadc1.Init.NbrOfConversion = 1;
414     hadc1.Init.DMAContinuousRequests = DISABLE;
415     hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
416     if (HAL_ADC_Init(&hadc1) != HAL_OK)

```

```

417 {
418     _Error_Handler(__FILE__, __LINE__);
419 }
420 /**Configure for the selected ADC regular channel its corresponding
421     rank in the sequencer and its sample time.
422     */
423 sConfig.Channel = ADC_CHANNEL_3;
424 sConfig.Rank = 1;
425 sConfig.SamplingTime = ADC_SAMPLETIME_3CYCLES;
426 if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
427 {
428     _Error_Handler(__FILE__, __LINE__);
429 }
430 /* DAC init function */
431 static void MX_DAC_Init(void)
432 {
433     DAC_ChannelConfTypeDef sConfig;
434     /**DAC Initialization
435     */
436     hdac.Instance = DAC;
437     if (HAL_DAC_Init(&hdac) != HAL_OK)
438     {
439         _Error_Handler(__FILE__, __LINE__);
440     }
441     /**DAC channel OUT1 config
442     */
443     sConfig.DAC_Trigger = DAC_TRIGGER_T2_TRGO;
444     sConfig.DAC_OutputBuffer = DAC_OUTPUTBUFFER_ENABLE;
445     if (HAL_DAC_ConfigChannel(&hdac, &sConfig, DAC_CHANNEL_1) != HAL_OK)
446     {
447         _Error_Handler(__FILE__, __LINE__);
448     }
449 }
450 /* TIM2 init function */
451 static void MX_TIM2_Init(void)
452 {
453     TIM_ClockConfigTypeDef sClockSourceConfig;
454     TIM_MasterConfigTypeDef sMasterConfig;
455     htim2.Instance = TIM2;
456     htim2.Init.Prescaler = 320;
457     htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
458     htim2.Init.Period = 843;
459     htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
460     htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
461     if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
462     {
463         _Error_Handler(__FILE__, __LINE__);
464     }
465     sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
466     if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
467     {

```

```

468     _Error_Handler(__FILE__, __LINE__);
469 }
470 sMasterConfig.MasterOutputTrigger = TIM_TRGO_UPDATE;
471 sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
472 if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) !=
    HAL_OK)
473 {
474     _Error_Handler(__FILE__, __LINE__);
475 }
476 }
477 /* USART6 init function */
478 static void MX_USART6_UART_Init(void)
479 {
480     huart6.Instance = USART6;
481     huart6.Init.BaudRate = 9600;
482     huart6.Init.WordLength = UART_WORDLENGTH_8B;
483     huart6.Init.StopBits = UART_STOPBITS_1;
484     huart6.Init.Parity = UART_PARITY_NONE;
485     huart6.Init.Mode = UART_MODE_TX_RX;
486     huart6.Init.HwFlowCtl = UART_HWCONTROL_NONE;
487     huart6.Init.OverSampling = UART_OVERSAMPLING_16;
488     huart6.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
489     huart6.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
490     if (HAL_UART_Init(&huart6) != HAL_OK)
491     {
492         _Error_Handler(__FILE__, __LINE__);
493     }
494 }
495 /**
496  * Enable DMA controller clock
497  */
498 static void MX_DMA_Init(void)
499 {
500     /* DMA controller clock enable */
501     HAL_RCC_DMA1_CLK_ENABLE();
502     /* DMA interrupt init */
503     /* DMA1_Stream5_IRQn interrupt configuration */
504     HAL_NVIC_SetPriority(DMA1_Stream5_IRQn, 0, 0);
505     HAL_NVIC_EnableIRQ(DMA1_Stream5_IRQn);
506 }
507 /** Configure pins as
508     * Analog
509     * Input
510     * Output
511     * EVENT_OUT
512     * EXTI
513     PC1  —————> ETH_MDC
514     PA1  —————> ETH_REF_CLK
515     PA2  —————> ETH_MDIO
516     PA7  —————> ETH_CRS_DV
517     PC4  —————> ETH_RXD0
518     PC5  —————> ETH_RXD1

```

```

519     PB13  —————> ETH_TXD1
520     PD8   —————> USART3_TX
521     PD9   —————> USART3_RX
522     PA8   —————> USB_OTG_FS_SOF
523     PA9   —————> USB_OTG_FS_VBUS
524     PA10  —————> USB_OTG_FS_ID
525     PA11  —————> USB_OTG_FS_DM
526     PA12  —————> USB_OTG_FS_DP
527     PG11  —————> ETH_TX_EN
528     PG13  —————> ETH_TXD0
529 */
530 static void MX_GPIO_Init(void)
531 {
532     GPIO_InitTypeDef GPIO_InitStructure;
533     /* GPIO Ports Clock Enable */
534     __HAL_RCC_GPIOC_CLK_ENABLE();
535     __HAL_RCC_GPIOH_CLK_ENABLE();
536     __HAL_RCC_GPIOA_CLK_ENABLE();
537     __HAL_RCC_GPIOF_CLK_ENABLE();
538     __HAL_RCC_GPIOB_CLK_ENABLE();
539     __HAL_RCC_GPIOD_CLK_ENABLE();
540     __HAL_RCC_GPIOG_CLK_ENABLE();
541     /*Configure GPIO pin Output Level */
542     HAL_GPIO_WritePin(SignalPWM_GPIO_Port, SignalPWM_Pin, GPIO_PIN_RESET);
543     /*Configure GPIO pin Output Level */
544     HAL_GPIO_WritePin(GPIOB, LD3_Pin|Buzzer_Pin|LD2_Pin, GPIO_PIN_RESET);
545     /*Configure GPIO pin Output Level */
546     HAL_GPIO_WritePin(USB_PowerSwitchOn_GPIO_Port, USB_PowerSwitchOn_Pin,
547                       GPIO_PIN_RESET);
548     /*Configure GPIO pin : USER_Btn_Pin */
549     GPIO_InitStructure.Pin = USER_Btn_Pin;
550     GPIO_InitStructure.Mode = GPIO_MODE_IT_RISING;
551     GPIO_InitStructure.Pull = GPIO_NOPULL;
552     HAL_GPIO_Init(USER_Btn_GPIO_Port, &GPIO_InitStructure);
553     /*Configure GPIO pins : RMII_MDC_Pin RMII_RXD0_Pin RMII_RXD1_Pin */
554     GPIO_InitStructure.Pin = RMII_MDC_Pin|RMII_RXD0_Pin|RMII_RXD1_Pin;
555     GPIO_InitStructure.Mode = GPIO_MODE_AF_PP;
556     GPIO_InitStructure.Pull = GPIO_NOPULL;
557     GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
558     GPIO_InitStructure.Alternate = GPIO_AF11_ETH;
559     HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);
560     /*Configure GPIO pins : RMII_REF_CLK_Pin RMII_MDIO_Pin RMII_CRS_DV_Pin
561     */
562     GPIO_InitStructure.Pin = RMII_REF_CLK_Pin|RMII_MDIO_Pin|RMII_CRS_DV_Pin;
563     GPIO_InitStructure.Mode = GPIO_MODE_AF_PP;
564     GPIO_InitStructure.Pull = GPIO_NOPULL;
565     GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
566     GPIO_InitStructure.Alternate = GPIO_AF11_ETH;
567     HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
568     /*Configure GPIO pin : SignalPWM_Pin */
569     GPIO_InitStructure.Pin = SignalPWM_Pin;
570     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;

```

```

569 GPIO_InitStruct.Pull = GPIO_NOPULL;
570 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
571 HAL_GPIO_Init(SignalPWM_GPIO_Port, &GPIO_InitStruct);
572 /*Configure GPIO pin : RMII_TXD1_Pin */
573 GPIO_InitStruct.Pin = RMII_TXD1_Pin;
574 GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
575 GPIO_InitStruct.Pull = GPIO_NOPULL;
576 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
577 GPIO_InitStruct.Alternate = GPIO_AF11_ETH;
578 HAL_GPIO_Init(RMII_TXD1_GPIO_Port, &GPIO_InitStruct);
579 /*Configure GPIO pins : LD3_Pin Buzzer_Pin LD2_Pin */
580 GPIO_InitStruct.Pin = LD3_Pin|Buzzer_Pin|LD2_Pin;
581 GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
582 GPIO_InitStruct.Pull = GPIO_NOPULL;
583 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
584 HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
585 /*Configure GPIO pins : STLK_RX_Pin STLK_TX_Pin */
586 GPIO_InitStruct.Pin = STLK_RX_Pin|STLK_TX_Pin;
587 GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
588 GPIO_InitStruct.Pull = GPIO_NOPULL;
589 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
590 GPIO_InitStruct.Alternate = GPIO_AF7_USART3;
591 HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
592 /*Configure GPIO pin : USB_PowerSwitchOn_Pin */
593 GPIO_InitStruct.Pin = USB_PowerSwitchOn_Pin;
594 GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
595 GPIO_InitStruct.Pull = GPIO_NOPULL;
596 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
597 HAL_GPIO_Init(USB_PowerSwitchOn_GPIO_Port, &GPIO_InitStruct);
598 /*Configure GPIO pin : USB_OverCurrent_Pin */
599 GPIO_InitStruct.Pin = USB_OverCurrent_Pin;
600 GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
601 GPIO_InitStruct.Pull = GPIO_NOPULL;
602 HAL_GPIO_Init(USB_OverCurrent_GPIO_Port, &GPIO_InitStruct);
603 /*Configure GPIO pins : USB_SOF_Pin USB_ID_Pin USB_DM_Pin USB_DP_Pin */
604 GPIO_InitStruct.Pin = USB_SOF_Pin|USB_ID_Pin|USB_DM_Pin|USB_DP_Pin;
605 GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
606 GPIO_InitStruct.Pull = GPIO_NOPULL;
607 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
608 GPIO_InitStruct.Alternate = GPIO_AF10_OTG_FS;
609 HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
610 /*Configure GPIO pin : USB_VBUS_Pin */
611 GPIO_InitStruct.Pin = USB_VBUS_Pin;
612 GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
613 GPIO_InitStruct.Pull = GPIO_NOPULL;
614 HAL_GPIO_Init(USB_VBUS_GPIO_Port, &GPIO_InitStruct);
615 /*Configure GPIO pins : RMII_TX_EN_Pin RMII_TXD0_Pin */
616 GPIO_InitStruct.Pin = RMII_TX_EN_Pin|RMII_TXD0_Pin;
617 GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
618 GPIO_InitStruct.Pull = GPIO_NOPULL;
619 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
620 GPIO_InitStruct.Alternate = GPIO_AF11_ETH;

```

```

621 HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
622 }
623 /* USER CODE BEGIN 4 */
624 /* USER CODE END 4 */
625 /**
626  * @brief This function is executed in case of error occurrence.
627  * @param None
628  * @retval None
629  */
630 void _Error_Handler(char * file , int line)
631 {
632     /* USER CODE BEGIN Error_Handler_Debug */
633     /* User can add his own implementation to report the HAL error return
634        state */
634     while(1)
635     {
636     }
637     /* USER CODE END Error_Handler_Debug */
638 }
639 #ifdef USE_FULL_ASSERT
640 /**
641  * @brief Reports the name of the source file and the source line
642  *        number
643  *        where the assert_param error has occurred.
644  * @param file: pointer to the source file name
645  * @param line: assert_param error line source number
646  * @retval None
647  */
647 void assert_failed(uint8_t* file , uint32_t line)
648 {
649     /* USER CODE BEGIN 6 */
650     /* User can add his own implementation to report the file name and line
651        number,
652        ex: printf("Wrong parameters value: file %s on line %d\r\n", file ,
653        line) */
652     /* USER CODE END 6 */
653 }
654 #endif
655 /**
656  * @}
657  */
658 /**
659  * @}
660  */
661 /***** (C) COPYRIGHT STMicroelectronics *****/
        FILE*****/

```