

ESCUELA DE INGENIERÍA DE
TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE MÁSTER

**Sistema de control y monitorización de oclusión de
flujo sanguíneo para la mejora del rendimiento físico y
rehabilitación**

Titulación: Máster Universitario en Ingeniería de Telecomunicación

Autor: Francisco Carlos Felipe Rodríguez

Tutor: Dr. Antonio Gabriel Ravelo García

Cotutor: Dr. Víctor Guerra Yáñez

Fecha: julio de 2024

RESUMEN

Durante los últimos años, el grupo de investigación de “Rendimiento Humano, Ejercicio Físico y Salud” del Instituto Universitario de Investigaciones Biomédicas y Sanitarias (iUIBS) de la Universidad de Las Palmas de Gran Canaria ha estado investigando acerca de los posibles beneficios del entrenamiento con restricción de flujo sanguíneo. Sin embargo, a lo largo del proyecto se han encontrado con una serie de limitaciones que dificultan el progreso de la investigación. Por ello se han puesto en contacto tanto con el Grupo de Investigación IDeTIC: División de Procesado Digital de Señales del Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones como con el Grupo de Investigación de “Fabricación Integrada y Avanzada”, adscrito al departamento de Ingeniería Mecánica, para su incorporación al proyecto TRAINIRS, ayudando así tanto al desarrollo *hardware* como *software* de un sistema de oclusión del flujo que permita continuar con la investigación en curso.

Teniendo todo lo anterior en cuenta, este TFM se enfoca en el desarrollo del *hardware* y *software* necesarios para manejar un sistema de oclusión controlada del flujo sanguíneo en las extremidades, con el objetivo de mejorar tanto el rendimiento deportivo como aprovechar su potencial terapéutico. El sistema desarrollado debe controlar un complejo sistema mecánico que administra mangas de oclusión y debe adquirir lecturas de presión de las electroválvulas involucradas. Adicionalmente también se contempla la incorporación de una tecnología NIRS, que permita lecturas de la saturación de oxígeno en la zona ocluida mediante oximetría óptica cercana al infrarrojo. Y, por último, una interfaz gráfica mejorada que incremente las funcionalidades para la gestión integral del sistema.

Tras ser diseñado y fabricado el sistema propuesto se validó. Llegando a la conclusión que la integración de nuevas funcionalidades a través de la PCB de control, la interfaz manual y gráfica, y el botón de algometría, junto con el diseño del subsistema NIRS, ha incrementado notablemente sus posibilidades de uso y su eficiencia en la captura de datos. Dichas pruebas de validación han confirmado la funcionalidad y efectividad del sistema en condiciones reales de trabajo, sentando las bases para futuras mejoras. Aunque quedan limitaciones por superar, las perspectivas futuras son

optimistas y sugieren que este dispositivo podría convertirse en una herramienta de gran valor en los campos de los deportes de alto rendimiento y la salud, contribuyendo a la innovación y al bienestar de la sociedad.

ABSTRACT

Over the past years, the research group on "Human Performance, Physical Exercise, and Health" at the University Institute for Biomedical and Health Research (iUIBS) of the University of Las Palmas de Gran Canaria has been investigating the potential benefits of blood flow restriction training. However, throughout the project, they have encountered several limitations that hindered the progress of their research. Consequently, they have reached out to both the Research Group IDETIC: Division of Digital Signal Processing of the University Institute for Technological Development and Innovation in Communications, and the Research Group on "Integrated and Advanced Manufacturing," affiliated with the Department of Mechanical Engineering, to join the TRAINIRS project. This collaboration aims to assist in the hardware and software development of a blood flow occlusion system, enabling continued research.

Taking all the above into consideration, this project focuses on developing the necessary hardware and software to manage a controlled blood flow occlusion system in limbs, aiming to enhance athletic performance and leverage its therapeutic potential. The developed system must control a complex mechanical setup that manages occlusion cuffs and acquire pressure readings from the involved solenoid valves. Additionally, it includes incorporating NIRS technology for near-infrared spectroscopy readings of oxygen saturation in the occluded area. Lastly, an enhanced graphical interface is planned to increase system management functionalities.

After designing and fabricating the proposed system, validation tests confirmed that integrating new features through the control PCB, the manual and graphical interface, and the algometer button, along with the NIRS subsystem design, significantly increased its usability and data capture efficiency. These validation tests have affirmed the system's functionality and effectiveness under real working conditions, laying the groundwork for future improvements and applications. Despite remaining challenges, future prospects are optimistic, suggesting that this device could become a valuable tool in high-performance sports and healthcare, contributing to innovation and societal well-being.

Tabla de contenido

1.Introducción	1
1.1Antecedentes.....	2
1.2 Objetivos	4
1.3 Estructura de la memoria	5
2.Análisis del problema.....	7
2.1 Entrenamiento con restricción de flujo	8
2.2 Idea básica del sistema propuesto	15
2.3 Necesidades.....	17
3.Principios metodológicos de la propuesta.....	20
3.1 Metodología de diseño hardware de control	22
3.1.1 Requisitos fundamentales.....	22
3.1.2 Procedimiento	23
3.1.3 Métricas.....	23
3.2 Metodología de NIRS.....	24
3.2.1 Requisitos fundamentales.....	25
3.2.2 Procedimientos.....	26
3.2.3 Métricas.....	28
3.3 Metodología aplicada a la interfaz gráfica	29
3.3.1 Requisitos fundamentales.....	29
3.3.2 Procedimientos.....	30
4.Recursos y Herramientas	33
4.1 Hardware de control	34
4.1.1 KiCad	34
4.1.2 Lenguaje de programación C++.....	35
4.2 Hardware NIRS	36
4.2.1 LEDs infrarrojos	36
4.2.2 Fotorreceptor	37
4.2.3 Arduino Mega 2560.....	38
4.3 Interfaz Gráfica de Usuario	39
4.3.1 Python	39
5.Diseño	41
5.1 Diseño del hardware de control	42
5.1.1 Diseño hardware.....	42
5.1.2 Diseño software.....	56
5.2 Diseño del hardware NIRS	60
5.2.1 Diseño hardware.....	61
5.2.2 Diseño software.....	68
6.Implementación	77
6.1 Implementación del hardware de control	78
6.2 Implementación del hardware NIRS.....	80
6.3 Implementación de la GUI	89

7. Validación	92
7.1 Validación del hardware de control	93
7.2 Validación del hardware NIRS	100
7.3. Validación de la GUI	106
8. Conclusión	110
8.1 Conclusiones y posibles ampliaciones.....	111
9. Bibliografía	114
9.1 Bibliografía	115
10. Pliego de Condiciones	119
PL1. Condiciones Hardware	119
PL2. Condiciones Software	120
PL3. Condiciones de licencia	120
PL4. Derechos de autor	120
PL5. Restricciones	121
PL6. Garantías	121
PL7. Limitación de responsabilidad	121
PL8. Otras consideraciones	121
11. Presupuesto	123
P.1 Componentes del presupuesto.....	123
P.2 Recursos Materiales	124
P.3 Trabajo tarificado por tiempo empleado	127
P.3 Aplicación de impuestos y coste total.....	127
ANEXO	129
Software de la PCB de control	129
Software de control NIRS	132

Índice de Figuras

Figura 2.1. Ejemplo de rutina de entrenamiento pasivo [1].....	9
Figura 2.2. Proporción de los rangos de edad de los sujetos del estudio [9]	10
Figura 2.3. Número de pacientes registrados por edad [9]	11
Figura 2.4. Gráfico que muestra la aplicación de oclusiones según el entrenamiento [4]	11
Figura 2.5. Esquema de las posibles adaptaciones cardiovasculares y pulmonares que experimenta el cuerpo humano con el uso de entrenamientos de restricción de flujo [4].....	13
Figura 2.6. Estadística de los efectos secundarios sufridos por los sujetos [9]	14
Figura 2.7. Porcentaje de satisfacción de los usuarios que se sometieron al entrenamiento con restricción de flujo sanguíneo [9]	14
Figura 2.8. Mangas de oclusión del dispositivo Hokanson [51]	15
Figura 2.9. Parte mecánica del dispositivo implementado	16
Figura 2.10. Dispositivo comercial Hokanson E20/AG101 [52].....	18
Figura 3.1. Ciclo del principio metodológico de la propuesta	22
Figura 3.2. Reflexión de los haces de los LEDs [53].....	27
Figura 3.3. Representación de la señal NIRS [28]	28
Figura 4.1. Interfaz gráfica de KiCad [54]	35
Figura 4.2. Fotorreceptor S6775 [38]	37
Figura 4.3. Arduino Mega 2560 [33]	38
Figura 4.4. Comparación entre Tkinter y CustomTkinter [55]	40
Figura 5.1. PCB v0.2 fabricada.....	44
Figura 5.2. Diseño y conexionado de la red de alimentación en KiCad.....	45
Figura 5.3. Conexionado del DAC en el entorno de KiCad	46
Figura 5.4. Divisor de tensión diseñado en KiCad.....	48
Figura 5.5. Enrutado de la red de alimentación	49
Figura 5.6. Enrutado del DAC.....	49
Figura 5.7. Esquema de diseño de la interfaz manual.....	50
Figura 5.8. Esquema de conexión del encoder en KiCad	51
Figura 5.9. Enrutamiento correspondiente al encoder	52
Figura 5.10. Conexión del botón para configuración pull-down [56]	53
Figura 5.11. Conexionado y enrutamiento final de los botones.	54
Figura 5.12. Enrutado final de la placa v0.3	54
Figura 5.13. Conexionado final de la PCB v0.3.....	55
Figura 5.14. Ejemplo de funcionamiento de scheduler con tareas de diferentes periodos	56
Figura 5.15. Funcionalidad de la interfaz manual diseñada	60
Figura 5.16. Conexionado del espejo de corriente implementado	62
Figura 5.17. Gráfico corriente-potencia óptica del LED de 735 nm [34]	63
Figura 5.18. Gráfico corriente-potencia relativa del LED de 810 nm [35].....	63
Figura 5.19. Gráfico corriente-potencia relativa del LED de 850 nm [36]	63
Figura 5.20. Esquema diseñado para el fotorreceptor.....	66
Figura 5.21. Diagrama de funcionamiento del software de control NIRS	69
Figura 5.22. Diagrama de flujo del procotolo "Algometer Gradual".....	72
Figura 5.23. Diagrama de flujo del protocolo "Algometer Square"	73

Figura 5.24. Diagrama de flujo del protocolo "Basic Protocol"	75
Figura 6.1. Ventana de selección de capas para el archivo gerber	78
Figura 6.2. PCB que se recibió del fabricante	79
Figura 6.3. PCB v0.3 totalmente implementada	80
Figura 6.4. Esquema para montaje del sistema NIRS	81
Figura 6.5 Esquema de diseño para la sonda de medida	82
Figura 6.6. Resultado de la implementación de los componentes de las sondas	82
Figura 6.7. Caja de protección para la sonda	83
Figura 6.8. Separador de cables	84
Figura 6.9. Pieza de unión para ambas cajas	84
Figura 6.10. Montaje final del transmisor y el receptor	85
Figura 6.11. Conector DIN de 8 hilos utilizado	85
Figura 6.12. Resultado final del montaje del conector hembra	86
Figura 6.13. Resultado final del montaje de la sonda	86
Figura 6.14. Conector DIN macho implementado	87
Figura 6.15. Esquema de conexionado	88
Figura 6.16. Montaje final de los espejos de corriente	88
Figura 6.17. Sistema NIRS totalmente implementado	89
Figura 6.18. Secciones de la GUI básica	90
Figura 6.19. Implementación del protocolo Algometer Square	91
Figura 6.20. Implementación del protocolo básico de funcionamiento	91
Figura 6.21. Implementación del protocolo Algometer Gradual	91
Figura 7.1. Esquema de conexión de la entrada ICSP	93
Figura 7.2. Montaje para la programación del microprocesador	94
Figura 7.3. Conexión de la fuente de alimentación de 24 V	95
Figura 7.4. Medida de 7 V a la salida de la red de conversión	95
Figura 7.5. Implementación del botón de algometría	97
Figura 7.6. Modificación del flag al pulsar el botón	98
Figura 7.7. Interfaz manual completamente implementada	99
Figura 7.8. LCD correctamente programada	99
Figura 7.9. Visualización del inflado de cada manga	100
Figura 7.10. Montaje del sistema NIRS completo	101
Figura 7.11. Colocación de las sondas NIRS 1	103
Figura 7.12. Colocación de las sondas NIRS 2	104
Figura 7.13. Fichero de los datos NIRS capturados	104
Figura 7.14. Sujeto listo para la realización del protocolo gradual	107
Figura 7.15. Lectura en tiempo real de la prueba que se está realizando	108
Figura 7.16. Fichero de datos generado para uno de los sujetos	109
Figura 7.17. Momento en el cual se registra cuando el usuario pulsa el botón durante la prueba	109
Figura A.1. Constantes del software de la PCB de control	129
Figura A.2. Tareas del TaskScheduler y sus períodos	130
Figura A.3. Funciones del programa	130
Figura A.4. Setup del software de la PCB de control	131
Figura A.5. Constantes, variables e inicialización del software de control NIRS	132
Figura A.6. Programa Python para la captura de datos NIRS	133

Índice de Tablas

Tabla 4.1. Listado de LEDs utilizados para las sondas.	36
Tabla 5.1. Conexionado de los pines del DAC	47
Tabla 5.2. Conexionado de los pines del encoder	52
Tabla 5.3. Resumen de tareas del TaskScheduler	59
Tabla 5.4. Puntos de trabajo conocidos del LED-735	64
Tabla 5.5. Resumen de los valores obtenidos para el diseño de los espejos de corriente	66
Tabla 5.6. Comandos de operación de la GUI	71
Tabla 5.7. Parámetros a configurar en el protocolo "Basic Protocol"	74
Tabla 6.1. Correspondencia entre color del cable y componente	83
Tabla 7.1. Valores medidos para la sonda A	102
Tabla 7.2. Valores medidos para la sonda B	102
Tabla 10.1. Condiciones del Hardware	120
Tabla 10.2. Condiciones del Software	120
Tabla 11.1. Recursos Materiales	126

Acrónimos

BFR	<i>Blood Flow Restriction</i>	SPI	<i>Serial Peripheral Interface</i>
BJT	<i>Bipolar Junction Transistor</i>	TFM	Trabajo Fin de Máster
CPU	<i>Central Processing Unit</i>	THT	<i>Thru-Hole Technology</i>
CSV	Valores Separados por Coma	UART	<i>Universal Asynchronous Receiver/Transmitter</i>
DAC	<i>Digital-to-Analog Converter</i>	USB	<i>Universal Serial Bus</i>
DIN	<i>Deutshes Institut fur Normung</i>		
GH	Hormona del crecimiento		
GUI	<i>Graphical User Interface</i>		
HHb	Desoxihemoglobina		
ICSP	<i>In Chip Serial Programmer</i>		
I2C	<i>Inter-Integrated Circuit</i>		
LCD	<i>Liquid Crystal Display</i>		
LED	<i>Light Emitting Diode</i>		
MISO	<i>Master In Slave Out</i>		
MOSI	<i>Master Out Slave In</i>		
NIRS	<i>Near InfraRed Spectroscopy</i>		
O2Hb	Oxihemoglobina		
PCB	<i>Printed Circuit Board</i>		
PWM	<i>Pulse Width Modulation</i>		
SCLK	<i>Serial Clock</i>		

1. Introducción

En este capítulo se presenta la idea general del sistema desarrollado y su entorno, los objetivos perseguidos, algunos conceptos básicos relacionados con las tecnologías usadas y la estructura de la memoria.

1.1 Antecedentes

Actualmente, son muchos los estudios que demuestran los beneficios que pueden obtenerse mediante la oclusión sanguínea en extremidades, generando lo que se conoce como isquemia, llegando incluso a observarse una mejora en la respuesta adaptativa de los músculos [1] [2]. Esto resulta de gran interés tanto en el ámbito del entrenamiento deportivo, permitiendo investigar nuevas técnicas que mejoren el rendimiento y la recuperación de los deportistas de alto rendimiento, como en el ámbito de la salud, buscando nuevas formas de combatir o prevenir posibles enfermedades [3] [4].

Durante los últimos años, el grupo de investigación de “Rendimiento Humano, Ejercicio Físico y Salud” del Instituto Universitario de Investigaciones Biomédicas y Sanitarias (iUIBS) de la Universidad de Las Palmas de Gran Canaria ha estado investigando acerca de este innovador método. Sin embargo, a lo largo del proyecto se han encontrado con una serie de limitaciones que dificultan el progreso de la investigación. Por ello se han puesto en contacto tanto con el Grupo de Investigación IDeTIC: División de Procesado Digital de Señales del Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones como con el Grupo de Investigación de “Fabricación Integrada y Avanzada”, adscrito al departamento de Ingeniería Mecánica, para su incorporación al proyecto TRAINIRS, ayudando así tanto al desarrollo *hardware* como *software* de un sistema de oclusión del flujo que permita continuar con la investigación en curso.

El proyecto TRAINIRS representa un avance significativo en el campo de la mejora del rendimiento en actividades físicas y la terapia rehabilitadora. Este proyecto innovador tiene como objetivo el desarrollo *hardware* y *software* necesario para implementar un sistema de oclusión controlada del flujo sanguíneo en las extremidades, con el objetivo de mejorar tanto el rendimiento deportivo como potencial terapéutico. El sistema debe disponer de una serie de subsistemas integrados, incluyendo un complejo sistema hidráulico que administre mangas de oclusión. Este sistema hidráulico debe estar gestionado por un sistema empotrado que también permita la adquisición de lecturas de presión de las electroválvulas involucradas. Adicionalmente, incorporará

una tecnología NIRS (*Near InfraRed Spectroscopy*) de vanguardia, que permita lecturas en tiempo real de la saturación de oxígeno en la zona ocluida mediante oximetría óptica.

Al comienzo de este TFM (Trabajo Fin de Máster), toda la parte mecánica ya había sido implementada por el grupo de ingenieros mecánicos y el sistema empotrado de control se encontraba en las primeras etapas de desarrollo. Se había fabricado una primera PCB (*Printed Circuit Board*), la cual denominaremos PCB v0.1, que proporcionó cierta funcionalidad al equipo ya que permitió realizar las primeras pruebas de inflado de mangas y lecturas de datos. Tenía capacidad de gestionar hasta dos mangas y tres pines de lectura de datos. Sin embargo, los requisitos de interfaces entrada/salida fueron aumentado y, además, contaba con una interfaz manual muy limitada y no incluía la posibilidad de incorporar la tecnología NIRS aun por desarrollar. Por ello, se decidió rediseñar la placa y fabricar una segunda versión, que a lo largo de este TFM se denominará PCB v0.2.

La PCB v0.2 es una placa mucho más completa que permite aumentar las posibilidades de diseño y de uso del sistema de forma considerable. No obstante, presentaba una serie de limitaciones, que se comentarán con mayor detalle posteriormente y que debían ser subsanadas para el correcto funcionamiento del equipo. Por lo tanto, en este TFM se plantea el diseño, implementación y fabricación de la PCB v0.3 que permita optimizar y mejorar el funcionamiento del dispositivo TRAINIRS eliminando los errores que limitan su uso dentro de las investigaciones en curso.

Por otro lado, la tecnología NIRS que el dispositivo debe implementar aún no se había comenzado a diseñar, con lo cual el diseño, implementación y fabricación de la tecnología NIRS se contempla como parte de los objetivos de este TFM. Esta tecnología nos va a permitir dotar al dispositivo de unas características únicas en el mercado ya que no existe ningún dispositivo capaz de realizar oclusiones y lecturas de la saturación de oxígeno de forma simultánea.

Teniendo todo lo anterior en cuenta, este TFM se enfoca en el desarrollo del *hardware* y *software* necesarios para manejar un sistema de oclusión controlada del flujo sanguíneo en las extremidades, con el objetivo de mejorar tanto el rendimiento deportivo como aprovechar su potencial terapéutico. El sistema desarrollado debe

controlar un complejo sistema mecánico que administra mangas de oclusión y debe adquirir lecturas de presión de las electroválvulas involucradas.

Adicionalmente también se contempla la incorporación de una tecnología NIRS, que permita lecturas de la saturación de oxígeno en la zona ocluida mediante oximetría óptica cercana al infrarrojo. Y, por último, una interfaz gráfica mejorada que incremente las funcionalidades para la gestión integral del sistema.

1.2 Objetivos

Para asegurar la funcionalidad, eficiencia y escalabilidad de la solución en este proyecto, se han identificado varios objetivos técnicos cruciales que necesitan ser abordados:

- **O1. Diseñar y Fabricar una PCB para el Sistema Neumático:** Este objetivo implica crear una PCB específica que albergue todos los componentes electrónicos necesarios para el sistema neumático de oclusión de flujo planificando la distribución e integración de estos componentes. **Medio de verificación:** el firmware desarrollado para realizar las lecturas de los sensores correspondientes y la actuación sobre las válvulas garantiza el funcionamiento adecuado del sistema neumático.
- **O2. Caracterización del Subsistema NIRS:** Se requiere realizar un análisis detallado y la caracterización del subsistema NIRS y sus componentes. El objetivo es asegurar un buen rendimiento en la medición de la saturación de oxígeno mediante oximetría óptica. Para ello será necesario desarrollar una electrónica específica para el subsistema NIRS que permita su conexión e integración fluida con el sistema empotrado principal. **Medio de verificación:** el firmware desarrollado para realizar lecturas precisas de los niveles de saturación de oxígeno de las células musculares.
- **O3. Diseñar y Fabricar las Sondas de Medida:** Se trata de crear sondas de medida. Estas sondas pueden contar con diodos de diferentes longitudes de onda de emisión y uno o varios fotodiodos. Su función es captar las señales necesarias para la oximetría. **Medio de verificación:** pruebas de funcionamiento

de las sondas en condiciones simuladas y reales para validar su desempeño en la captación de señales de oximetría.

- **O4. Diseñar e Implementar una Interfaz Gráfica de Control:** Se busca desarrollar una nueva interfaz gráfica que no solo mejore la existente, sino que también incremente significativamente sus prestaciones. Con esta actualización se pretende ofrecer una experiencia de usuario más eficiente facilitando la interacción del usuario con el sistema y permitiendo una gestión más efectiva.
Medio de verificación: pruebas de usabilidad para validar la eficiencia y efectividad de la nueva interfaz gráfica de control.

1.3 Estructura de la memoria

La memoria de este TFM se ha estructurado en 8 capítulos, bibliografía y anexos. A continuación, se presenta la estructura de esta memoria y una breve descripción de los capítulos que serán desarrollados a lo largo de la misma.

En el primer capítulo hace de introducción al problema que se está tratando de dar solución en este TFM. Comenzando por una breve descripción del potencial que tiene este tipo entrenamientos de restricción de flujo y concluyendo con la historia y estado del proyecto al inicio de este TFM.

En el segundo capítulo se profundiza en la descripción teoría de los entrenamientos basados en isquemia, dando una visión más detallada de su potencial. Además, se comenta la idea básica del sistema a desarrollar, así como las principales necesidades que se deben solventar.

En el tercer capítulo se profundiza en la metodología general a seguir para el desarrollo del proyecto, especificándola para cada una de las partes diferenciadas del mismo. Esto permite conocer la metodología aplicada en cada sección del proyecto de manera detallada y los requisitos de diseño específicos de cada una.

En el cuarto capítulo se describen los recursos, tanto *hardware* como *software*, que se van a utilizar a lo largo del desarrollo de este TFM para llevar a cabo la consecución de los objetivos planteados.

En capítulo 5 se muestran las soluciones de diseño adoptadas para cada uno de los requisitos planteados en el capítulo anterior. De esta forma, se justifica cada una de las decisiones tomadas y se puede hacer un seguimiento de la línea de desarrollo del proyecto.

En el sexto capítulo se detalla cómo se llevó a cabo la implementación de cada una de las soluciones, mostrando en detalle las partes desarrolladas y los resultados de fabricación obtenidos.

En el capítulo 7 se recoge la validación de cada una de las partes del proyecto desarrolladas para poder así verificar el correcto funcionamiento y comprobar que se han podido completar los objetivos planteados al inicio de este TFM.

Por último, en el octavo capítulo, se comentan las conclusiones después del desarrollo de este TFM y se muestran posibles ampliaciones que se podrían llevar a cabo en futuras etapas del proyecto.

Para finalizar, se concluye con la bibliografía donde se enumeran las referencias empleadas para el desarrollo del TFM, el pliego de condiciones y el presupuesto donde se expone un análisis del coste económico de llevar a cabo el proyecto.

2. Análisis del problema

El análisis del problema se centra en identificar y comprender los fundamentos de los entrenamientos basados en la restricción del flujo sanguíneo, así como en examinar las limitaciones que presentan los dispositivos actuales para implementar esta técnica. Una vez analizados los problemas principales, se detallarán las necesidades esenciales que deben cubrirse para superar estas deficiencias y mejorar la eficacia de los dispositivos disponibles en el mercado.

2.1 Entrenamiento con restricción de flujo

El entrenamiento deportivo es un proceso sistemático y planificado que busca mejorar las cualidades físicas y/o mentales de la persona que desarrolla la actividad, contribuyendo así a su salud [5]. Este proceso incluye una serie de actividades y ejercicios que deben adaptarse a las cualidades de cada persona en particular, por lo tanto, el desarrollo de nuevos ejercicios y técnicas siempre ha sido un componente clave dentro del entrenamiento deportivo para tratar de alcanzar los mejores resultados posibles.

Debido a esta necesidad surgió lo que se conoce como entrenamientos con restricción de flujo sanguíneo o BFR (*Blood Flow Restriction*). Técnica que se basa en ocluir el flujo sanguíneo a través de las venas y restringir parcialmente el flujo arterial. Se debe tener en cuenta que el sistema circulatorio está compuesto por arterias, que transportan la sangre oxigenada desde el corazón a los diferentes órganos, y por venas, que llevan la sangre desoxigenada de vuelta al corazón. Para ello se aplica una presión externa, por ejemplo, haciendo uso de manguitos de compresión neumáticos, que restringe de forma total el paso de circulación sanguínea a través de las venas, las cuales se encuentran en capas más superficiales y de forma parcial el flujo a través de arterias que se encuentran en capas más profundas. Consiguiendo así que la sangre entre en el músculo pero que no salga de él [6].

Esta técnica puede combinarse con diferentes tipos de entrenamiento para potenciar así sus beneficios y obtener mejores resultados [1]:

- Entrenamientos aeróbicos. Este tipo de entrenamientos se llevan a cabo con una intensidad moderada durante un periodo prolongado, utilizando principalmente el oxígeno como como fuente de energía. Algunos ejemplos de estos ejercicios serían correr, nadar o saltar a la comba.
- Entrenamientos anaeróbicos. En este caso se reduce el periodo de tiempo de actividad, pero se aumenta la intensidad de los ejercicios, haciendo que la energía proceda principalmente de otras fuentes como puede ser la glucosa y no

el oxígeno, como sucedía anteriormente. Un ejemplo de entrenamiento sería el levantamiento de pesas.

- Entrenamientos pasivos. Este tipo de entrenamiento se caracterizan por la ausencia de ejercicios ya que está pensado para compensar la atrofia muscular que sufren las personas en periodos de reposo o inmovilización. Por ejemplo, en pacientes que han sufrido una operación de rodilla, ya que se reduce la cantidad de masa muscular que tendrían que recuperar en la posterior etapa de rehabilitación. En la Figura 2.1 se puede ver un ejemplo de rutina de entrenamiento para este tipo de caso.

Guidelines	
Frequency	1–2 times per day (duration of bed rest/immobilization)
Restriction time	5 min intervals
Type	Small and large muscle groups (arms and legs/uni or bilateral)
Sets	3–5
Cuff	5 (small), 10 or 12 (medium), 17 or 18 (large)
Rest between sets	3–5 min Uncertain – higher pressure may be needed (70–100% AOP)
Pressure	
Restriction form	Continuous

Figura 2.1. Ejemplo de rutina de entrenamiento pasivo [1]

El entrenamiento de fuerza ha demostrado ser efectivo para producir hipertrofia muscular cuando se realiza a intensidades superiores al 70%. Sin embargo, gracias al uso de técnicas de reducción de flujo se puede llegar a alcanzar niveles de hipertrofia con cargas de tan sólo el 20% [7]. Esta técnica puede ser especialmente relevante en ciertos grupos de población, como personas que se encuentran recuperándose de una lesión que ha implicado inmovilización en alguna extremidad o personas de avanzada edad con bajos niveles de masa y acondicionamiento muscular donde el movimiento de altas cargas puede conllevar riesgos de lesión.

Esta novedosa técnica surgió en Japón en el año 1966 y fue denominada como Kaatsu. Su creador fue el doctor Yoshiako Sato, quien descubrió que esta técnica tenía

grandes beneficios en la recuperación de lesiones y el rendimiento deportivo, cuyo principal objetivo es estimular la hipertrofia muscular, es decir, el aumento y la fuerza de los músculos [8]. El uso de esta técnica está ampliamente extendido en el país asiático, debido a los buenos resultados de su aplicación en situaciones reales. Se ha llevado cabo un estudio que permitió comprobar cómo se puede emplear dicha técnica y los beneficios que puede llegar a tener [9]. Para dicho estudio se consideran los datos y experiencias de un total de 12.642 sujetos, de los cuales el 45,4% eran hombres y el 54,6% mujeres, de diferentes edades. En la Figura 2.2 se puede ver la proporción de los rangos de edades de los sujetos que participaron en el estudio.

Esta técnica se introdujo en diferentes tipos de establecimientos relacionados con el ámbito de la salud y el tratamiento de dolencias físicas, como puede ser hospitales y clínicas, osteópatas, centros de acupuntura, centros de rehabilitación y gimnasios entre otros. Los pacientes acudían a este tipo de centros para tratar diferentes tipos de dolencias, en la Figura 2.3 se puede ver un recuento del número de pacientes y las diferentes enfermedades que sufrían.

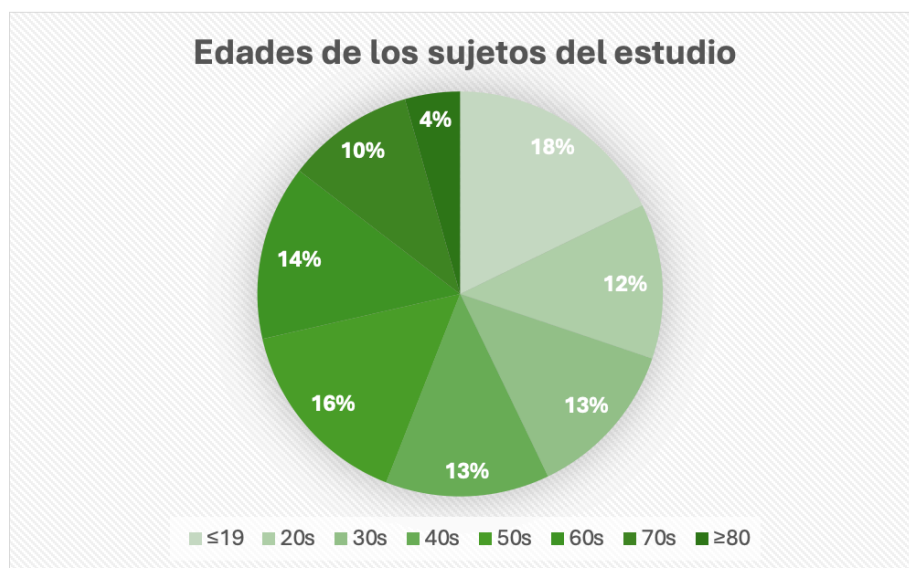


Figura 2.2. Proporción de los rangos de edad de los sujetos del estudio [9]

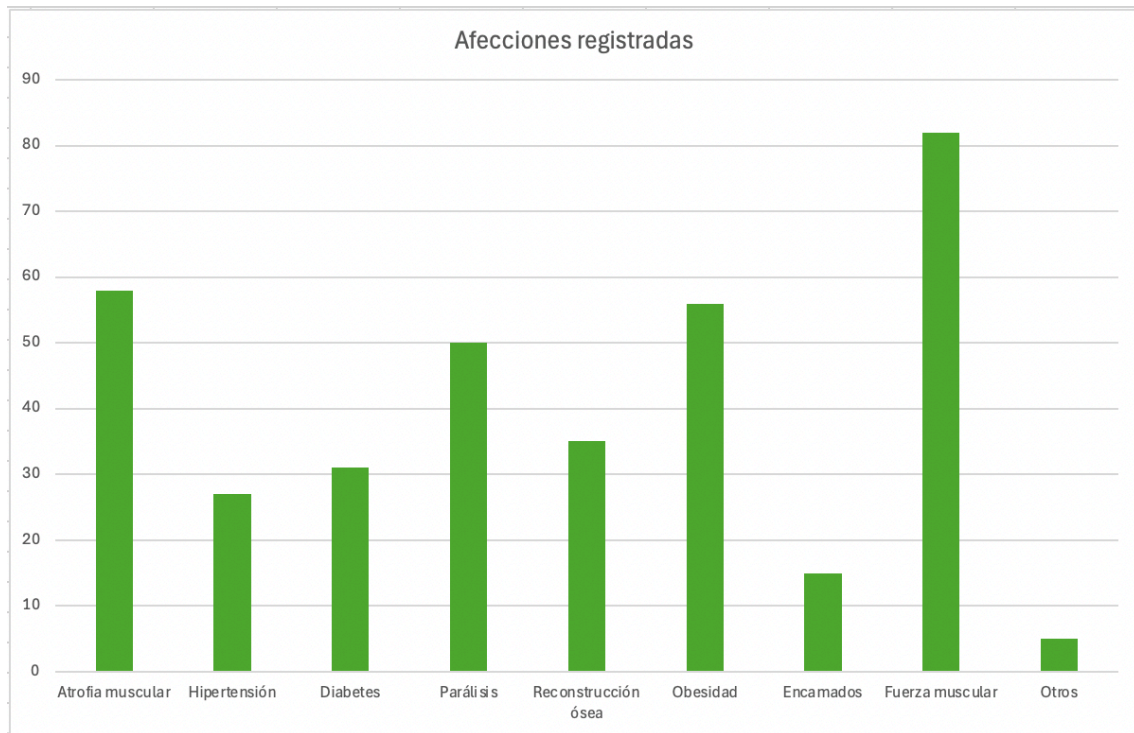


Figura 2.3. Número de pacientes registrados por edad [9]

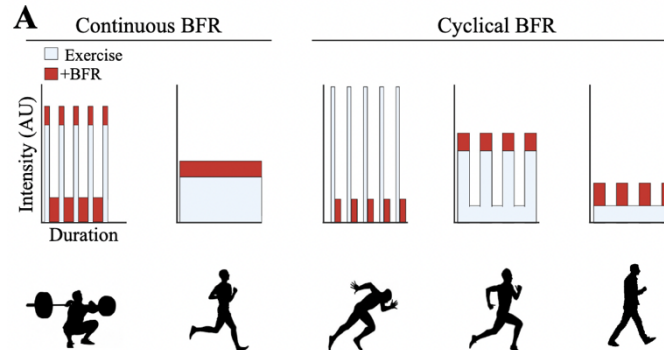


Figura 2.4. Gráfico que muestra la aplicación de oclusiones según el entrenamiento [4]

Según el estudio, la metodología Kaatsu puede ser combinada con otro tipo de actividades físicas, en este caso se combinó con sesiones de estiramientos, caminatas, bicicletas o incluso sesiones de pesas en el gimnasio. La sesión de entrenamiento final va a depender en gran medida de la dolencia que se quiera tratar y las características y condición física del paciente. Por ejemplo, según el tipo de trabajo que se vaya a realizar se pueden aplicar oclusiones continuas o cíclicas. Las oclusiones continuas son aquellas que se aplican de forma constante durante todo el entrenamiento independientemente del tipo de actividad o esfuerzo. Sin embargo, las oclusiones cíclicas se aplican de forma

temporal siguiendo un protocolo de trabajo predefinido antes de iniciar la sesión de trabajo. En la Figura 2.4 se puede ver un gráfico que muestra cómo sería la secuencia de oclusiones en un entrenamiento de cada tipo.

Son numerosos los beneficios reportados de la aplicación de los entrenamientos de restricción de flujo en sus diferentes variantes de implementación:

- El entrenamiento con oclusión permite a las personas entrenar a intensidades mucho más bajas con los beneficios del entrenamiento de alta intensidad, siendo muy útil, como se explicó anteriormente, para personas en situaciones de posoperación y para mejorar la función muscular en la población mayor postrada en cama [10].
- Se ha demostrado su utilidad dentro de la rehabilitación cardíaca ya que estimula significativamente las respuestas de diferentes tipos de hormonas, como puede ser, la hormona de crecimiento (GH) que ayuda al crecimiento y reparación de los tejidos, el factor de crecimiento similar a la insulina que desempeña un papel importante en el desarrollo y crecimiento del corazón y el factor de crecimiento endotelial vascular que permite la formación de nuevos vasos sanguíneos. Además, durante el ejercicio con oclusión, la carga de sangre que el corazón necesita bombear se reduce lo que permite a los pacientes con problemas cardíacos reducir el riesgo durante la actividad [11]. En la Figura 2.5 se puede ver las posibles adaptaciones cardiovasculares y del músculo que se experimentan con el uso de esta técnica.
- Permite la disminución del daño muscular, lo que permite reducir el riesgo de lesiones y el tiempo de recuperación [12].
- El desarrollo del músculo esquelético, que es un tipo de músculo estriado conectado al esqueleto, que permite el movimiento y mantiene unidos los huesos de las articulaciones a través de su contracción [4]. Esta técnica ayuda a mantener estos músculos en buen estado, ayudando a prevenir la artrosis (desgaste del cartílago), la sarcopenia (pérdida de masa y fuerza muscular debido al envejecimiento) y la atrofia muscular.

Además, un estudio [13] demuestra los beneficios de las técnicas de BFR dentro del programa de entrenamiento de los astronautas. Durante los vuelos espaciales, los astronautas sufren problemas de salud debido a los cambios en la función cardiovascular producida por la falta de gravedad, haciendo que el líquido intravascular de las piernas se desplace hacia la cabeza llegando a producir edema intracraneal y dolor de cabeza. Según el estudio, con la aplicación de oclusioniones en ambos muslos de los sujetos en posición supina se inducen alteraciones hemodinámicas y hormonales muy similares a las de estar de pie. Con lo cual se puede contrarrestar los síntomas que sufren los astronautas en estos periodos de ingravidez.

Por otro lado, se registró un caso [14] en el que este tipo de entrenamiento favoreció y agilizó la recuperación de un paciente que presentaba una fractura ósea del fémur, estimulando el crecimiento del hueso y la reparación del tejido ósea, así como evitando la atrofia muscular debido a la inmovilización de la pierna.

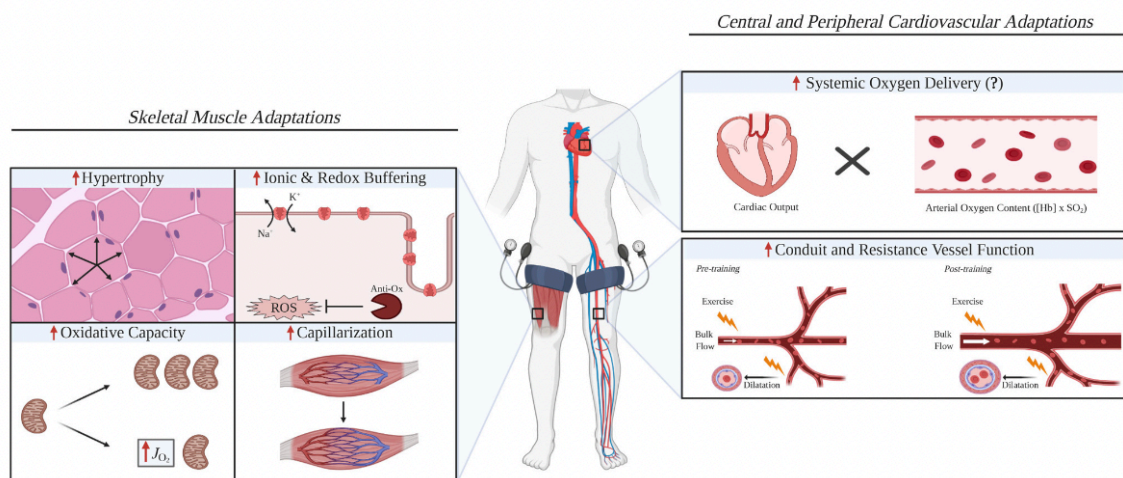


Figura 2.5. Esquema de las posibles adaptaciones cardiovasculares y pulmonares que experimenta el cuerpo humano con el uso de entrenamientos de restricción de flujo [4]

Sin embargo, este tipo de entrenamiento no está exento de riesgo y contraindicaciones, ya que en el estudio realizado se registró que el cuerpo de todos los pacientes no respondió de la misma forma al entrenamiento y experimentaron algún tipo de efecto adverso. En la Figura 2.6 se puede ver cuáles fueron las principales

consecuencias que sufrieron los pacientes a lo largo del entrenamiento. Por otro lado, en la Figura 2.7, se muestra que la gran mayoría de los pacientes quedaron muy satisfechos con los resultados obtenidos, consiguiendo una valoración positiva del más del 70%.

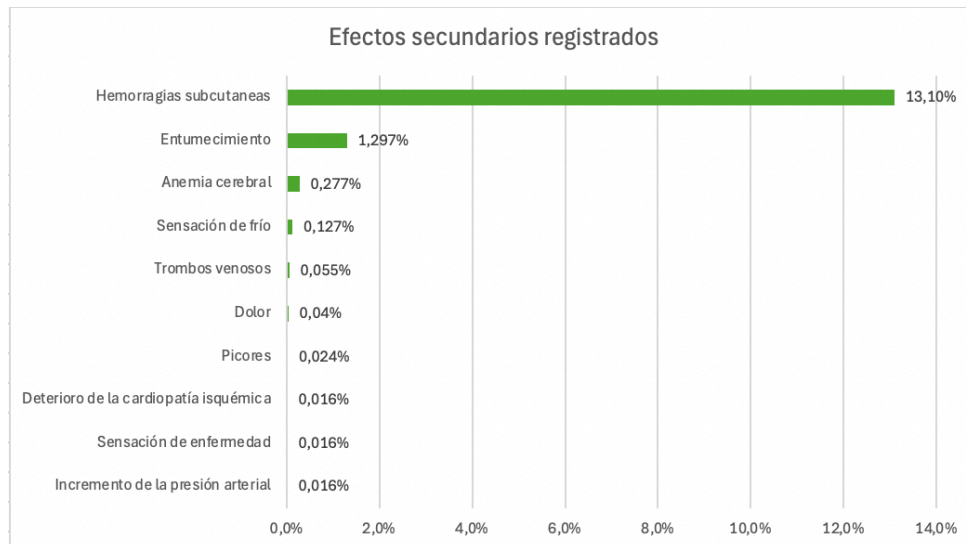


Figura 2.6. Estadística de los efectos secundarios sufridos por los sujetos [9]

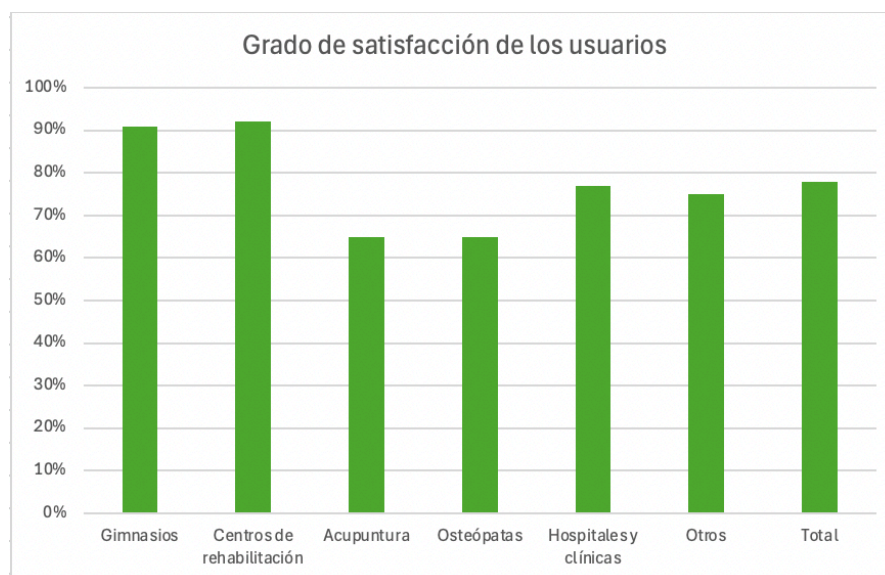


Figura 2.7. Porcentaje de satisfacción de los usuarios que se sometieron al entrenamiento con restricción de flujo sanguíneo [9]

2.2 Idea básica del sistema propuesto

Para poder llevar a cabo este tipo de entrenamiento de forma apropiada es necesario contar con un dispositivo neumático que permita comprimir las extremidades. Se requiere disponer de bandas neumáticas estrechas, como las de la Figura 2.8, que se coloquen alrededor de la extremidad con la que se quiera trabajar. Una vez colocada, se introduce aire dentro del manguito hinchable hasta alcanzar una presión que permita cortar la circulación sanguínea.

La presión necesaria para cortar la circulación en un brazo o una pierna depende de varios factores, como puede ser, la circunferencia del miembro del sujeto, la musculatura o la presión arterial del paciente. En general, se debe alcanzar una presión superior a la presión arterial sistólica del paciente, con lo cual, para alcanzar el proceso de isquemia en un brazo se debe aplicar una presión del orden de 150-250 mmHg mientras que en una pierna se deben alcanzar presiones de 250-300 mmHg [15].

En el mercado comercial actual, existen pocos dispositivos diseñados específicamente para esta finalidad con lo cual muchos de ellos presentan importantes limitaciones técnicas. Por lo tanto, puede ser que exista una demanda significativa de un dispositivo profesional e innovador que permita rediseñar los entrenamientos deportivos.



Figura 2.8. Mangas de oclusión del dispositivo Hokanson [51]

El objetivo del prototipo es la de producir isquemia, que es la reducción temporal o permanente del flujo sanguíneo de una parte del cuerpo, normalmente las extremidades, brazos o piernas [16]. Con este proceso lo que se busca es reducir la cantidad de oxígeno que recibe el músculo para poder estudiar cómo se comporta en situaciones de estrés deportivo y descubrir nuevos métodos que puedan aumentar el rendimiento del individuo, prevenir o revertir la atrofia muscular o incluso realizar ejercicios de menor intensidad con resultados similares o superiores a los de mayor carga.

El dispositivo cuenta con una base mecánica ya diseñada, por lo que lo que se aborda a lo largo de este TFM es el desarrollo de la parte electrónica del dispositivo. Dicha parte mecánica está formada principalmente por un compresor, encargado de suministrar el flujo de aire adecuado, las electroválvulas de llenado y vaciado de las mangas de presión, una serie de sensores de presión que permiten medir la presión alcanzada en cada manga durante el proceso de llenado y, por último, un depósito de presión que almacena aire comprimido para así poder realizar secuencias de llenado de mangas muchos más rápido ya que no hay que esperar que el compresor genere el aire. En la Figura 2.9 se puede ver el resultado final de la parte mecánica completamente montada dentro de la carcasa del dispositivo.

A continuación, en los siguientes apartados, se detallarán las principales necesidades de la parte electrónica, así como todo el proceso de desarrollo que ha sido necesario para poder llevarlo a cabo.



Figura 2.9. Parte mecánica del dispositivo implementado

2.3 Necesidades

Las necesidades que debe cubrir el dispositivo a diseñar fueron transmitidas por el equipo de investigación de “Rendimiento Humano, Ejercicio Físico y Salud” del Instituto Universitario de Investigaciones Biomédicas y Sanitarias de la Universidad de Las Palmas de Gran Canaria, que como se ha comentado anteriormente, está interesado en profundizar acerca de los posibles beneficios de la oclusión sanguínea en extremidades y cómo se puede llegar a amplificar la respuesta adaptativa de los músculos.

Hasta el momento, el equipo de investigación ha estado trabajando con equipos convencionales disponibles en el mercado, concretamente, con el dispositivo Hokanson E20/AG101 que se puede ver en la Figura 2.10. Sin embargo, han detectado una serie de limitaciones que obstaculizan el progreso de la investigación en curso, ya que el dispositivo no dispone de la precisión y las funcionalidades necesarias impidiendo así que el grupo de trabajo pueda recopilar los datos precisos y confiables que buscan. Algunas de estas limitaciones se detallan a continuación:

- Incapacidad de mantener una presión constante. El suministro de aire comprimido se agota gradualmente, lo que provoca una disminución de la presión en la manga.
- Imposibilidad de realizar ciclos consecutivos de manera eficiente. Requiere un intervalo entre cada carga para asegurar un llenado rápido de la manga.
- Falta de precisión en la determinación de la presión objetivo. Debido a que no cuenta con un indicador de presión.
- Dificultad en la recopilación y almacenamiento de datos de la prueba. Esto impide un análisis posterior de los resultados.
- Carencia de sistemas de seguridad específicos. No dispone de mecanismos que prevengan posibles accidentes al dispositivo o al sujeto.

Teniendo todo lo anterior en cuenta, la principal necesidad planteada es el desarrollo de un equipo innovador y fiable que permita el llenado y vaciado de mangas

con velocidad y precisión, así como la recopilación de datos necesarios que permitan obtener información de la prueba realizada. Con la parte mecánica del dispositivo ya implementada y funcionando correctamente, es necesario el diseño e implementación de una placa base que permita controlar y manejar toda la parte neumática. La PCB se debe diseñar de forma que pueda tanto actuar como recibir datos de la parte mecánica y que sea capaz de gestionar una interfaz manual que le permita el usuario poder trabajar fácilmente con el dispositivo y leer datos durante el entrenamiento.

Además de la interfaz manual comentada, resulta muy interesante poder contar con una interfaz gráfica o GUI (*Graphical User Interface*) de forma que se pueda utilizar el equipo desde el ordenador. Esto proporcionará al usuario la flexibilidad necesaria para usar diferentes métodos de trabajo, adaptándose a las necesidades específicas de la prueba que se vaya a realizar.



Figura 2.10. Dispositivo comercial Hokanson E20/AG101 [52]

Por último, es necesario disponer de unas sondas de medida basadas en tecnología NIRS. Esta tecnología se basa en aplicar una técnica no invasiva que permite medir la concentración de diversos componentes químicos en una muestra, en nuestro caso, el músculo [17]. Para ello, se basa en la transmisión y absorción de luz en la región del infrarrojo cercano del espectro electromagnético, abarcando longitudes de onda entre los 700-900 nanómetros. De esta forma se consigue medir la oxigenación del

tejido muscular durante pruebas físicas o en reposo para obtener valiosa información, acerca de la perfusión sanguínea y el metabolismo, que permita avanzar en la investigación llevada a cabo por el grupo.

En los siguientes apartados se profundizará en el diseño e implementación de cada una de las soluciones alcanzadas para tratar de cubrir todas las necesidades.

3. Principios metodológicos de la propuesta

En esta sección se detallan los principios metodológicos de nuestra propuesta, abordando los requisitos, procedimientos y métricas de las tres partes fundamentales que componen el proyecto: la PCB encargada del manejo de la parte neumática, las sondas de medida empleando tecnología NIRS y la interfaz gráfica del dispositivo.

Para llevar a cabo el desarrollo del proyecto se decidió implementar una metodología de trabajo basada en diseño, implementación, validación y realimentación, bajo un enfoque cíclico e iterativo como se puede ver en la Figura 3.1. Esta técnica es comúnmente utilizada en el desarrollo tanto de productos como de sistemas.

En la primera etapa de diseño se tuvo como objetivo identificar todas aquellas necesidades y requisitos del usuario. Para ello se realizó un análisis detallado de las necesidades de los posibles clientes para poder determinar claramente cuáles eran los requisitos de diseño. Una vez se tenían los requisitos claros, se planificó cómo se realizaría el diseño de cada una de las soluciones, qué tecnologías y recursos eran necesarios.

Terminada la etapa de diseño, dio comienzo la etapa de implementación, donde se llevó a cabo el desarrollo real del producto. Esto incluye la codificación de la parte *software*, la fabricación del producto o incluso la integración de las diferentes partes para asegurar que todos los componentes funcionaban como se esperaba.

Una vez se tuvo el producto montado, pudo dar comienzo la etapa de validación. Para ello se llevaron a cabo diferentes tipos de pruebas, funcionales, de rendimiento e incluso de seguridad, que permitieron garantizar que el dispositivo cumplía con todos los requisitos que se especificaron al inicio del proyecto.

Por último, durante todo el proceso de desarrollo se realizó una etapa de realimentación donde se recopilaban datos sobre el desempeño del producto para poder realizar una mejora continua. Esta retroalimentación nos llevó a realizar nuevas iteraciones del ciclo de diseño, implementación o validación, mejorando continuamente el producto.

Este enfoque iterativo permitió un desarrollo más flexible y adaptable, asegurando que el producto final no solo cumplía con los requisitos iniciales, sino que también se adaptaba a nuevas necesidades y mejoras constantes basándose en la retroalimentación. Además, facilitó la identificación temprana de problemas y la implementación de soluciones de manera oportuna.

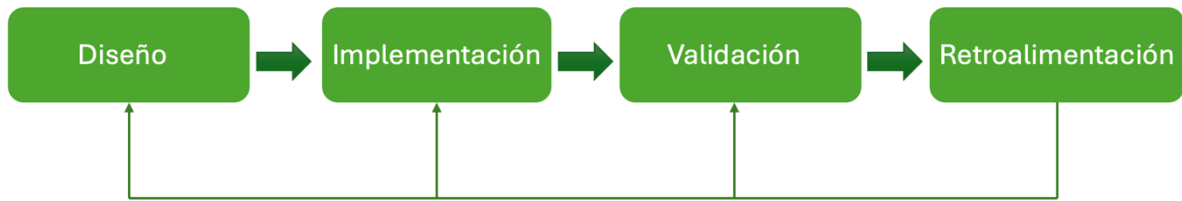


Figura 3.1. Ciclo del principio metodológico de la propuesta

3.1 Metodología de diseño hardware de control

En este apartado se va a profundizar en el diseño *hardware* de control, es decir, en el diseño de la PCB encargada de manejar toda la parte mecánica vista anteriormente. Para ello, se comienza detallando los requisitos que debe cumplir nuestro diseño. Una vez hecho esto, se procede a determinar que procedimientos se van a seguir para alcanzar el objetivo y qué métricas se van a tener en cuenta para poder considerar que la placa final está correctamente implementada.

3.1.1 Requisitos fundamentales

Cuando comienza el desarrollo de este TFM ya se había fabricado una PCB v0.2 que había sido probada detectándose una serie de deficiencias y limitaciones. Por lo tanto, era necesario el desarrollo de una nueva placa v0.3 que corrigiese y mejorase las prestaciones de la versión ya existente. A continuación, se recogen los requisitos que debía cumplir la PCB v0.3:

- Tras probar la v0.2 de la placa se detectó que existía una limitación importante en los niveles de presión que se podían alcanzar ya que nunca se alcanzaba la presión objetivo, sino que se quedaba en un nivel inferior. Por lo tanto, fue necesario amplificar el nivel de tensión de las señales de salida del microprocesador que llegan a las válvulas de presión encargadas de llenar las mangas para así poder alcanzar el nivel necesario.
- Se consideró necesaria la implementación de una interfaz manual más completa que permitiera utilizar el dispositivo directamente sin necesidad de utilizar el

ordenador. Con lo cual, fue imprescindible optimizar y ampliar las capacidades de la interfaz manual para aprovechar todas las funciones del equipo.

- Se determinó que sería de gran utilidad la posibilidad de disponer de un elemento extra que no estuviera instalado en el equipo, sino que lo pudiese utilizar el sujeto para realizar una marca y que esta quedase registrada en los datos que se estaban recopilando durante la prueba. Por ejemplo, esto iba a permitir al sujeto poder marcar a qué niveles de presión comenzaba a experimentar dolor en el transcurso del ejercicio.

3.1.2 Procedimiento

Para llevar a cabo la implementación de la PCB, se siguieron dos fases principales: el diseño del *hardware* y el desarrollo del *software*. En la fase de hardware, se utilizó KiCad [18], un programa de software libre que permitió diseñar la nueva PCB necesaria. Este proceso comenzó con la creación del diseño esquemático, incluyendo todos los componentes necesarios y asegurando que las conexiones estaban correctas. Luego, se procedió a diseñar la placa física, ubicando los componentes y trazando las rutas de las conexiones de manera óptima.

Una vez que se diseñó e implementó la placa v0.3, se procedió al desarrollo del *software*. En esta fase se obtuvo el código necesario a ejecutar en el microprocesador para permitir el adecuado funcionamiento del dispositivo. Para esta tarea, se empleó el entorno de desarrollo integrado de Arduino [19] y el lenguaje de programación C++.

3.1.3 Métricas

En este apartado se van a especificar las diferentes métricas que nos permitieron evaluar el rendimiento del dispositivo diseñado y comprobar que el resultado obtenido era satisfactorio. Por lo tanto, se relaciona cada métrica con cada uno de los requisitos comentados anteriormente, tratando así de reflejar con la mayor claridad posible cómo se consideró que cada requisito se había cumplido correctamente.

En primer lugar, para verificar que el primero de los requisitos se cumplía, se determinó que se debía obtener una señal de salida hacia las mangas mayor de 5 V al tratar de inflar la manga a la máxima presión objetivo, 375 mmHg. Como se comentó anteriormente, la placa integra un microprocesador ATMEGA2560 y sus salidas se encuentran limitadas a 5 V, impidiendo que se puedan alcanzar las presiones deseadas. Por lo tanto, aumentado la tensión de salida de dicha señal se aumenta el nivel de presión aplicado.

Por otro lado, se consideró que el segundo requisito se cumplía siempre y cuando la interfaz manual permitiese al usuario poder operar y utilizar todas aquellas funciones básicas del dispositivo sin necesidad de utilizar el ordenador. Se consideran como funciones básicas la de llenado y vaciado de las mangas de forma independiente o ambas de forma simultánea.

Por último, se concluyó que el tercer requisito estaba totalmente alcanzado cuando el elemento externo implementado fuese capaz de mandar una señal al microprocesador en el momento que el paciente interactuase con él y, de esta forma, el microprocesador ser capaz de generar una marca en el archivo de datos que permita registrar el momento y el nivel de presión que el paciente marcó.

3.2 Metodología de NIRS

En esta sección se detalla la metodología seguida para la parte NIRS del proyecto. Como se hizo anteriormente, se especifican los requisitos de diseño que se tuvieron en cuenta. Luego, se explica la técnica en la que se basa la tecnología NIRS para la recopilación de los datos aprovechando la reflexión de la luz en las células del músculo. Teniendo esto en cuenta, se detalla el procedimiento de diseño que se siguió para poder llevar a cabo el montaje de las sondas de tecnología NIRS. Finalmente, se detallan aquellas métricas que se tuvieron en cuenta para validar el correcto funcionamiento del sistema.

3.2.1 Requisitos fundamentales

Disponer de este tipo de tecnología permite dotar al dispositivo de la capacidad de realizar mediciones no invasivas y en tiempo real. La tecnología NIRS permite medir la oxigenación del tejido muscular de manera no invasiva, es decir, mediante técnicas que no implican penetración o incisión en el cuerpo humano que es mucho más traumático para el sujeto en estudio [20]. Además, los sistemas basados en NIRS no solo mide la oxigenación del tejido muscular, sino que también proporciona información sobre cómo está siendo distribuido el flujo sanguíneo a través del músculo. Esto es de gran utilidad ya que permite entender cómo se está regulando el suministro de oxígeno y nutrientes durante el ejercicio y cómo podría verse afectado por la fatiga muscular [21] [22].

Si se consigue llegar a comprender la dinámica de la oxigenación muscular y la respuesta del músculo ante situaciones de estrés se podrían llegar a optimizar los programas de entrenamiento deportivo y rehabilitación pudiendo así aumentar los beneficios para la salud de este tipo de ejercicios.

Basándonos en lo citado con anterioridad, a continuación, se detallan los requisitos que se tuvieron en cuenta la hora de diseñar la parte NIRS de nuestro dispositivo:

- Longitudes de onda adecuadas. Se requiere operar en un rango de longitudes de onda de 700-900 nanómetros, con lo cual era fundamental que las sondas diseñadas estuvieran optimizadas para trabajar en este rango específico y empleando varios LEDs (*Light Emitting Diode*) que trabajasen a diferentes longitudes para así poder tener una mayor flexibilidad, precisión y capacidad de análisis de la medición de la oxigenación.
- Sensibilidad del fotorreceptor. Se necesitaba disponer de un fotorreceptor con la sensibilidad necesaria para poder detectar todos los rayos reflejados por las células musculares.
- Seguridad y normativas vigentes. Al tratarse de un equipo que va a estar en contacto con la piel humana es necesario cumplir con la estricta regulación

vigente para equipos médicos de este tipo. Debido a esto, era necesario que la potencia óptica emitida por los LEDs fuese menor a 10 mW para garantizar que no se producen quemaduras en la piel [17].

- Fiabilidad y precisión. Es necesario que el dispositivo sea totalmente fiable y con una precisión adecuada para garantizar la toma de medidas. Esta toma de medida se debe realizar cada 10 ms por longitud de onda.
- Compatibilidad con pruebas físicas. Las sondas debían ser diseñadas teniendo en cuenta que se utilizarán durante pruebas físicas, por lo tanto, debían tener un cierto nivel de robustez y ser capaces de soportar las condiciones típicas del ejercicio físico, incluyendo movimientos y posibles variaciones en la posición del músculo.
- Disponer de una señal de sincronismo. Con esto lo que se busca es que el dispositivo sea compatible con el equipo de análisis de datos existente en el laboratorio. De esta forma se consigue que el sistema de sondas pueda ser fácilmente integrado con sistemas de adquisición de datos y software de análisis.

3.2.2 Procedimientos

La tecnología NIRS se basa en el uso de la luz infrarroja, con un espectro de longitud de onda 700-900 nanómetros. Este tipo de luz puede atravesar el tejido biológico sin dificultad y permite llegar a iluminar de forma no invasiva varios centímetros del tejido interno [23] [22]. Esta capacidad de penetrar el tejido se debe a la transparencia característica de este rango de longitudes de onda. Por esta razón, el espectro de luz infrarroja cercana se conoce comúnmente como ventana óptica [24]. En general, mediante este procedimiento se puede medir tanto la oxihemoglobina (O₂Hb) como la desoxihemoglobina (HHb), permitiendo así cuantificar indirectamente la variación en la oxigenación del tejido durante la actividad [25].

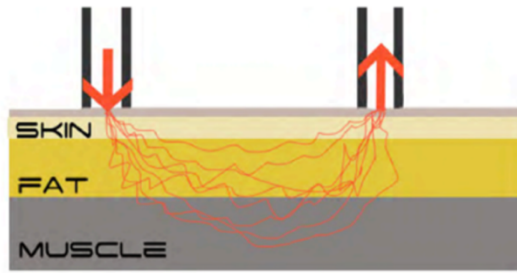


Figura 3.2. Reflexión de los haces de los LEDs [53]

Las fuentes de luz, es decir, los diodos emisores, se tiene que colocar directamente sobre la piel del sujeto, de esta forma los haces de luz emitidos penetrarán en ella y se reflejarán en las células del músculo siendo captados por el fotodetector, también llamado optodo [26]. En la Figura 3.2 se puede ver un resumen gráfico del funcionamiento explicado. Existe una relación de proporcionalidad entre la distancia LEDs-fotorreceptor y la profundidad de penetración de la luz. De modo que, a mayor distancia, más se dispersa la luz, por lo que el detector no debe colocarse a más de 3 cm de la fuente [23] [25].

Se debe tener en cuenta que para poder medir los cambios en la O₂Hb y en la HHb, el músculo necesita ser iluminado con varias longitudes de onda diferentes. Por lo tanto, en nuestro caso se seleccionaron hasta 3 LEDs de diferentes longitudes de onda insertándolos en la sonda, buscando así poder incrementar la precisión de los valores medidos [23]. Aunque no implementado en este punto, siempre es recomendable hacer uso de métodos adicionales de preprocesamiento como filtrados de la señal para así poder eliminar los efectos de la respiración o el latido del propio corazón [27].

Por ejemplo, en el siguiente estudio [28] se trató de determinar la saturación de oxígeno en el músculo esquelético empleando LEDs infrarrojos de 850 y 760 nanómetros. Las mediciones se tomaron en tres fases distintas: primero, mientras el sujeto estaba en reposo, luego, durante la aplicación de una isquemia al paciente, y finalmente, durante un periodo en el que se permitió nuevamente la entrada de sangre al músculo. En la Figura 3.3 se puede ver gráficamente el resultado obtenido. En este caso, se puede observar claramente como durante el proceso de isquemia la hemoglobina oxigenada (línea roja) se reduce y la desoxihemoglobina (línea azul) alcanza valores máximos ya que se está restringiendo la entrada de sangre oxigenada

en el músculo. Sin embargo, cuando se libera la presión y se permite el paso de la sangre oxigenada al músculo, se puede comprobar como la oxihemoglobina alcanza valores máximos.

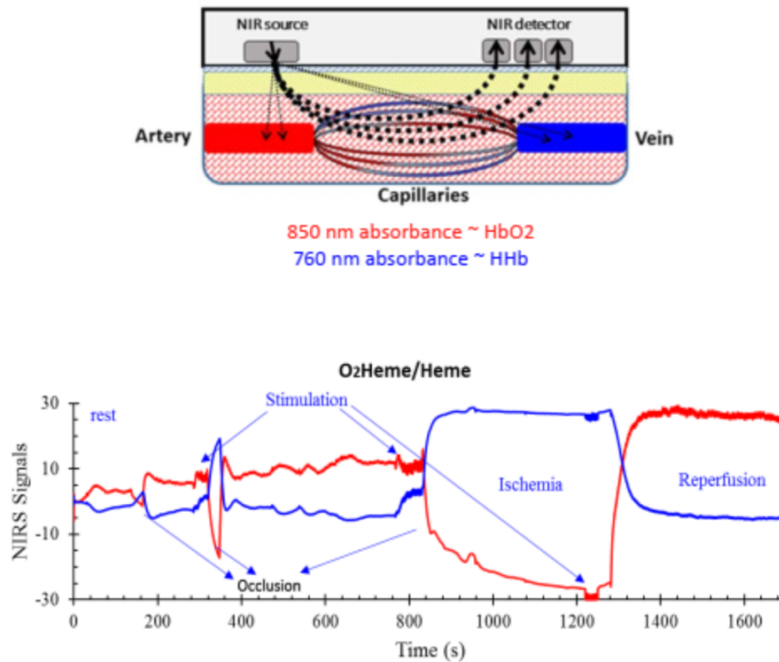


Figura 3.3. Representación de la señal NIRS [28]

3.2.3 Métricas

Como sucedió anteriormente, a continuación, se recogen las diferentes métricas que se emplearon para verificar el funcionamiento de los requisitos comentado en el apartado 3.2.1 Requisitos fundamentales.

- Respuesta del fotorreceptor. Se debe comprobar que el fotodetector sea capaz de captar eficazmente los rayos reflejados por las células musculares en el rango de longitudes de onda utilizadas y bajo las condiciones de entrenamiento físico.
- Seguridad y cumplimiento de las normativas. Se midió y garantizó que la potencia emitida por los LEDs fuese menor a 10 mW para cumplir con las regulaciones de seguridad y evitar posibles daños en la piel.

- Fiabilidad y precisión. Es necesario verificar que la frecuencia de muestreo de trabajo es la adecuada, es decir, que se realizase una medición por cada longitud de onda cada 10 ms.
- Integración con sistemas existentes. Se comprobó que la señal de sincronismo funcionase correctamente haciendo a nuestro equipo totalmente compatible con los equipos existentes en el laboratorio de trabajo.

3.3 Metodología aplicada a la interfaz gráfica

A continuación, se va a detallar la metodología de la interfaz gráfica necesaria para poder interactuar con el equipo mediante el ordenador personal. En primer lugar, se especifican los requisitos que debe cumplir nuestro diseño. Una vez definidos, se profundiza en el procedimiento seguido para el desarrollo de la interfaz, detallando aquellas tecnologías o medios que vayan a ser utilizados. Por último, se detalla las métricas que se utilizan para verificar que el diseño está correctamente implementado.

3.3.1 Requisitos fundamentales

Es necesario disponer de una interfaz gráfica que cuente con los controles básico de funcionamiento. Es decir, que pueda realizar llenados y vaciados de las mangas de presión, en una sola o en las dos, de forma independiente. De esta forma, lo que se consigue es poder manejar el equipo tanto desde el ordenador mediante la interfaz gráfica como desde la interfaz manual comentada anteriormente, permitiendo así que el dispositivo se pueda adaptar mejor a las necesidades del usuario en ese momento.

Además, uno de los requisitos es contar con un sistema que permita el uso de la algometría de presión como test sensorial cuantitativo. La algometría de presión es un método válido que se utiliza para la evaluación del dolor en seres humanos mediante diferentes tipos de pruebas, en nuestro caso, empleando técnicas de presión [29]. De esta forma se busca poder registrar el momento en el que el sujeto experimenta dolor durante el proceso de isquemia para poder determinar el comportamiento del músculo en diferentes escenarios.

Para ello se diseñaron tres protocolos de trabajo, que deben manejarse y configurarse desde la interfaz gráfica. El funcionamiento de estos protocolos se detalla a continuación, permitiendo registrar el dolor del sujeto mediante diferentes secuencias de procesos de isquemia:

- *“Algometer Gradual”*. En este protocolo se aumenta la presión de la manga de forma progresiva hasta que el sujeto sienta dolor. En ese momento, debe interactuar con algún elemento que permita detener el llenado, vaciar la manga y almacenar que presión se alcanzó. Este proceso se repite en una pierna primero, luego en la otra y finalmente en ambas.
- *“Algometer Square”*. Para este protocolo se realizan una serie de isquemias consecutivas en las que sonará una alarma. Cuando se produzca este sonido, el sujeto debe comunicar el nivel de dolor que está sintiendo para su registro. Se debe tener en cuenta que todos los valores se tienen que poder configurar para poder adaptar mejor la prueba al paciente y sus condiciones, por lo tanto, se debe recibir como parámetros el tiempo que va a durar cada isquemia, el tiempo de descanso entre isquemias, la presión objetivo de cada isquemia y el tiempo exacto en que sonará la alarma durante el proceso de isquemia.
- *“Basic Protocol”*. Este protocolo permite configurar toda la sesión de trabajo de la prueba que se vaya a realizar. Se le permite configurar al usuario los tiempos que va a durar la isquemia, los diferentes tiempos de descanso entre isquemias, el número de isquemias que se quiere realizar y la presión que se quiere alcanzar.

Por último, se necesita que la interfaz permita el análisis de datos tanto en tiempo real durante la realización de la prueba como una vez finalizada. Por lo tanto, se debe almacenar los datos de la prueba en algún tipo de fichero e implementar el código necesario para la visualización en pantalla de los datos recogidos por los sensores del equipo.

3.3.2 Procedimientos

Para el desarrollo de la interfaz gráfica se empleó Python y la biblioteca CustomTkinter, que facilita la creación de interfaces modernas y personalizadas. Como

se mostró anteriormente, primero se definieron los requisitos funcionales y de diseño de la interfaz, asegurando que fuera intuitiva y fácil de utilizar. Luego, se instalaron las dependencias necesarias, incluyendo Python y CustomTkinter.

En el momento de mi incorporación al proyecto se encontraba implementada una primera versión muy básica de la GUI que contaba con los controles básico de inflado y desinflado de mangas. Sin embargo, toda la interfaz tuvo que ser reestructurada, separando de forma clara los componentes visuales de la aplicación para mantener un código limpio y manejable.

Durante el desarrollo, se creó un *frame* principal que permite al usuario actuar sobre el equipo donde se agregaron varios *widgets* como botones, etiquetas, campos de entrada y cuadros de selección. Estos elementos se personalizaron para alinearse con el diseño deseado, empleando las opciones de estilo y configuración de CustomTkinter. Además, se implementaron las funciones de control de eventos para manejar las interacciones del usuario, asegurando que cada acción tuviera una respuesta adecuada. Una vez diseñado el *frame* principal, se creó otro destinado a la visualización de los datos enviados por el equipo. Pudiendo así obtener información de la prueba en tiempo real.

3.3.3 Métricas

En esta sección, se detallarán las diversas métricas que se utilizaron para evaluar el funcionamiento de la GUI diseñada y asegurar que cumple con los requisitos establecidos anteriormente. Para facilitar la verificación de la interfaz esta fue dividida en tres grandes bloques, la GUI básica, que corresponde con los controles de la funcionalidad básica del dispositivo, los protocolos de algometría y, por último, el análisis de datos, que incluye tanto el almacenamiento de datos como la visualización de estos en tiempo real.

A continuación, se recogen las métricas empleadas para la verificación de cada una de las partes basándonos en los requisitos previamente comentados:

- GUI básica. En este caso es necesaria la verificación de controles y la independencia de estos, es decir, se tiene que comprobar que todos los controles básicos funcionan según lo esperado y que los llenados y vaciados de las mangas de presión pueden realizarse tanto en una sola manga como en ambas.
- Protocolos de algometría de presión. Para llevar a cabo la validación de los protocolos implementados se decidió que la mejor opción era la realización de una serie de pruebas en un entorno real. Para ello, con ayuda del grupo de investigación de “Rendimiento Humano, Ejercicio Físico y Salud”, se organizaron una serie de pruebas en las que se emplearon los protocolos de algometría. Esto permitió al grupo recopilar los datos necesarios para sus estudios y a nosotros validar nuestra interfaz gráfica.
- Análisis de los datos. Es necesario verificar la integridad y formato de los datos que se almacenan. Para ello se comprobó que el fichero se creaba correctamente y se almacenaban los datos necesarios en el formato predefinido. Además, se verificó que los datos se graficaban en tiempo real durante la prueba permitiendo así poder realizar un seguimiento del paciente a lo largo del ejercicio.

4. Recursos y Herramientas

En este TFM se han empleado diversos recursos y herramientas para el desarrollo y fabricación de las diferentes partes que lo componen. A continuación, se muestra cada una de ellas, así como una breve explicación para comprender mejor el uso que se le han dado dentro del ámbito del proyecto.

4.1 Hardware de control

A continuación, se detalla el uso de las herramientas empleadas en el diseño e implementación de la PCB v0.3. Para el diseño esquemático y el enrutado de todos los componentes de la placa, se utilizó KiCad, como se mencionó anteriormente. Una vez fabricada la PCB, fue necesario programarla para dotarla de funcionalidad. Para ello, se utilizó el lenguaje de programación C++.

4.1.1 KiCad

KiCad es una aplicación de software libre y multiplataforma especializada en el diseño electrónico automatizado. Fue un proyecto que comenzó Jean-Pierre Charras en el año 1992 y que continúa en desarrollo gracias a una comunidad activa de usuarios y desarrolladores que contribuyen a su desarrollo y mejora continua [30].

Este programa permite a los usuarios crear diagramas esquemáticos de circuitos electrónicos que representan gráficamente los componentes y sus conexiones. Además, facilita el diseño de placas de circuito impreso, permitiendo elegir la distribución de los componentes y hacer el enrutamiento de las conexiones eléctricas entre ellos, destacando las capacidades de modelado, representación en 3 dimensiones y depurado de trazado de una forma muy completa. Debido a todo esto, se decidió trabajar con este programa para realizar el diseño de la placa v0.3 que se contempla en este TFM. En la Figura 4.1 se puede ver la interfaz del programa, en la ventana de la izquierda se muestra gráficamente los componentes y sus conexiones, y en la ventana de la derecha se puede ver el diseño de la placa, tanto sus componentes como el enrutamiento entre ellos.

Por otro lado, cuenta con una extensa cantidad de bibliotecas que pueden ser utilizadas por los diseñadores de forma gratuita. Sin embargo, también ofrece la oportunidad de crear y personalizar tus propias bibliotecas de componentes. Esto es de gran valor ya que con su herramienta de simulación se podrá simular el circuito antes de fabricarlo y comprobar así que los componentes son los adecuados y todo funciona correctamente. Por último, permite generar fácilmente los archivos *Gerber* necesarios

para la fabricación de la PCB. Estos son los archivos que se le deben enviar al fabricante para que pueda construir nuestra placa.

En nuestro caso la versión descargada y con la que se va a trabajar a lo largo del desarrollo del TFM es la 7.0.10.

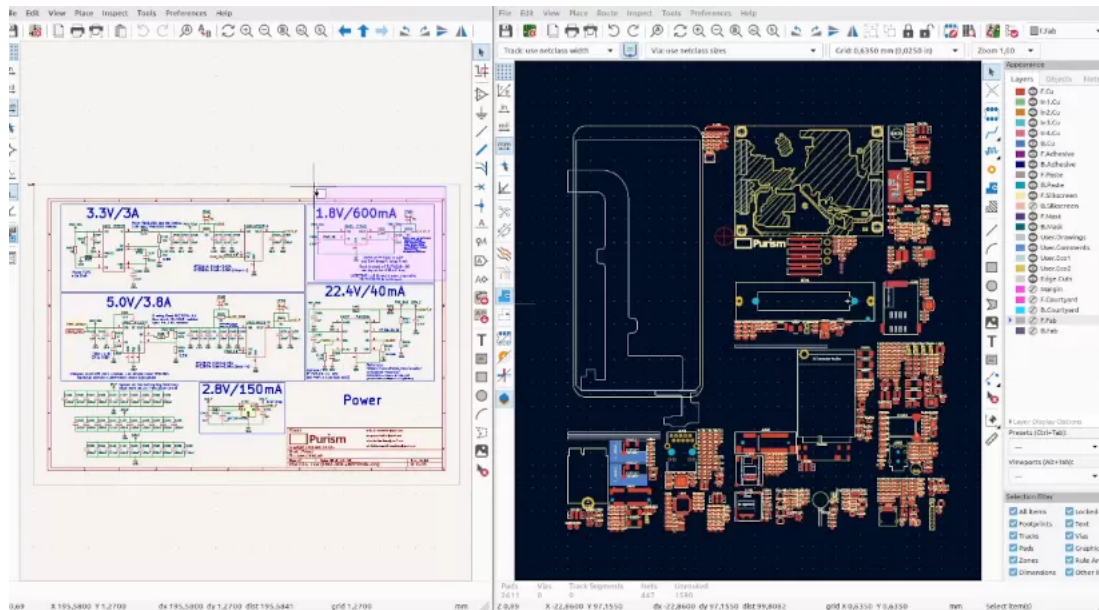


Figura 4.1. Interfaz gráfica de KiCad [54]

4.1.2 Lenguaje de programación C++

El lenguaje C++ es un tipo de lenguaje de programación compilado, multiparadigma, principalmente de tipo imperativo y orientado a objetos [31]. Se trata de una extensión del lenguaje C buscando poder manipular objetos. Sus principales ventajas son el alto rendimiento que ofrece ya que puede hacer llamadas directas al sistema operativo y posee una gran variedad de parámetros de optimización. Además, se trata de un lenguaje que se ha ido actualizando con los años, permitiendo operar con datos complejos e implementando múltiples patrones de diseño [32].

Debido a que nuestra PCB integra el microprocesador ATMEGA2560 presente en el producto comercial Arduino MEGA [33], se ha decidido trabajar con el IDE de Arduino

que emplea el lenguaje C++ como lenguaje nativo. Esto nos permite desarrollar y cargar el código en nuestra placa de manera sencilla dentro del propio entorno de trabajo de Arduino.

4.2 Hardware NIRS

Los elementos necesarios para poder llevar a cabo este procedimiento serían los siguientes: LEDs infrarrojos de diferentes longitudes de onda, un fotorreceptor con la sensibilidad suficiente como para detectar la reflexión de los rayos y un pequeño hardware capaz de ejecutar un programa C que permita realizar la emisión y la captura de haces de luz y datos.

4.2.1 LEDs infrarrojos

Los LEDs infrarrojos, como ya se ha comentado, son diodos emisores de luz que emiten radiación en la región del infrarrojo del espectro electromagnético, generalmente en el rango de 700-1000 nanómetros. Además, tienen la característica que son muy eficientes en términos de energía ya que convierten una alta proporción de la energía eléctrica en radiación infrarroja y suelen ser muy direccionales, esto quiere decir que emiten luz en un haz estrecho permitiendo así poder llegar a capas más profundas de la piel.

En la siguiente Tabla 4.1 se recogen los 3 LEDs que se van a utilizar durante este TFM para el montaje de las sondas de medida. En apartados posteriores se abordará en detalle el diseño necesario para que funcionen correctamente.

LONGITUD DE ONDA	MODELO
735 NM	LXML-PF01 [34]
810 NM	MTE8120CP [35]
850 NM	MTE8600MC [36]

Tabla 4.1. Listado de LEDs utilizados para las sondas.

4.2.2 Fotorreceptor

Un fotorreceptor es un dispositivo capaz de responder con una señal eléctrica a estímulos de señales luminosas [37]. Teniendo esto en cuenta, el fotorreceptor desempeña un papel fundamental dentro de nuestro sistema NIRS ya que es el componente encargado de detectar la luz infrarroja emitida por los LEDs, proporcionando así la información crítica sobre la fisiología del tejido biológico que se está estudiando.

En nuestro caso se va a trabajar con el fotorreceptor S6775 [38] de *Hamamatsu*, debido a su gran área fotosensible, alta sensibilidad y rápida respuesta. Además, este diodo de silicio está diseñado para aplicaciones de detección de luz en un amplio rango de longitudes de onda, desde la ultravioleta hasta el infrarrojo cercano. Estas características son cruciales ya que se mejora la eficiencia de recepción, permitiendo captar más luz y obteniendo mediciones precisas de señales que pueden llegar a ser un tanto débiles.

En la Figura 4.2 se puede ver una imagen del fotorreceptor comentado, donde se puede ver su diseño compacto y robusto que facilita su integración en diversos sistemas, mientras que su bajo consumo de energía lo hace especialmente adecuado para aplicaciones donde las restricciones energéticas son un problema.

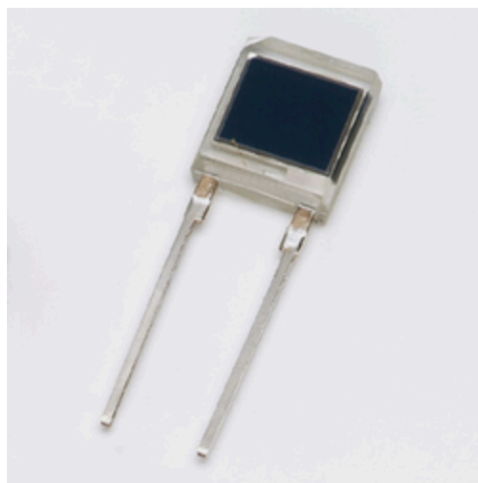


Figura 4.2. Fotorreceptor S6775 [38]

4.2.3 Arduino Mega 2560

Para llevar a cabo todo el control del sistema NIRS implementado en este TFM se ha decidido utilizar el Arduino Mega 2560 [33] debido a sus capacidades avanzadas y su flexibilidad. Este microcontrolador ofrece una plataforma robusta y accesible para la implementación de sistemas complejos, siendo ideal para proyectos que requieren múltiples entradas y salidas.

El Arduino Mega 2560 destaca por varias características esenciales que lo hace adecuado para este proyecto. Posee un microcontrolador ATmega2560, el mismo que se emplea en la PCB v0.3, que proporciona 54 pines digitales de entrada/salida, de los cuales 15 pueden ser utilizados como salidas PWM (*Pulse Width Modulation*), y 16 pines de entrada analógica. Esta amplia disponibilidad de pines permite conectar diversos sensores y actuadores, lo cual es crucial para el futuro crecimiento de la parte NIRS desarrollada.

Además, este microprocesador destaca por su facilidad de programación y su compatibilidad con una amplia gama de bibliotecas y módulos. Utiliza un entorno de desarrollo integrado de Arduino, que simplifica el proceso de escritura y carga de código en la placa. Su capacidad de manejar comunicaciones serie, SPI (*Serial Peripheral Interface*) e I2C (*Inter-Integrated Circuit*) facilita la integración con otros dispositivos y sensores, ampliando las posibilidades de desarrollo. Estas características, junto a su extensa documentación, convierten al Arduino Mega 2560 en una herramienta potente y versátil para llevar a cabo proyectos como el de este TFM. En la Figura 4.3 se puede ver el diseño y los componentes que implementa dicha placa.

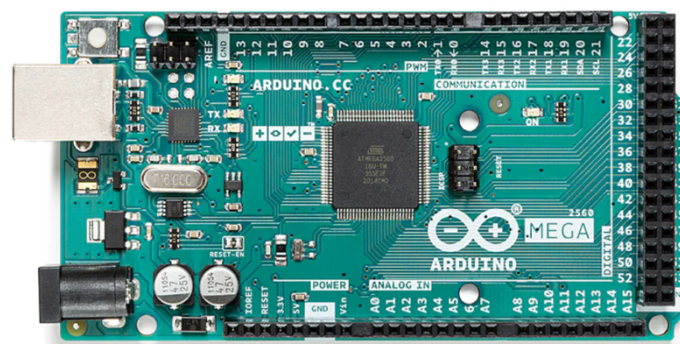


Figura 4.3. Arduino Mega 2560 [33]

4.3 Interfaz Gráfica de Usuario

Para el desarrollo de la interfaz gráfica de usuario se ha optado por utilizar Python junto con la biblioteca *CustomTkinter*, una extensión de *Tkinter* que permite crear interfaces más modernas y personalizadas. Se decidió emplear estas herramientas ya que proporcionan la flexibilidad y funcionalidad necesarias para diseñar una GUI intuitiva y eficiente. A continuación, se profundizará en las características y ventajas de ellas, destacando su aplicación específica en el desarrollo de la interfaz.

4.3.1 Python

Python es un lenguaje de programación interpretado, lo que significa que se ejecuta línea por línea sin necesidad de compilación previa. Es multiplataforma, por lo que puede funcionar en diversos sistemas operativos, lo que lo hace adecuado para una amplia gama de aplicaciones. Además, permite la creación de programas utilizando diferentes estilos de programación, ya que admite varios paradigmas, como la programación orientada a objetos, estructurada, imperativa y funcional [39].

En nuestro caso se trabajó con la versión de Python 3.9.12 [40] combinándolo con la biblioteca *CustomTkinter* [41]. Para ser exactos, dicha biblioteca es una extensión de la biblioteca tradicional *Tkinter* de Python. Se decidió trabajar con esta extensión ya que ofrece una apariencia más moderna y atractiva facilitando así la creación de GUI con una estética más actual y con características adicionales que no están disponibles en *Tkinter* nativo. Por ejemplo, cuenta con una amplia gama de componentes como botones, etiquetas o incluso cuadros de texto a los que se le puede modificar colores, fuente, bordes y otros aspectos visuales consiguiendo así un entorno mucho más moderno y personalizable. En la Figura 4.4 se puede ver la gran diferencia que existe al crear una simple calculadora utilizando la *Tkinter* o la extensión comentada, *CustomTkinter*.

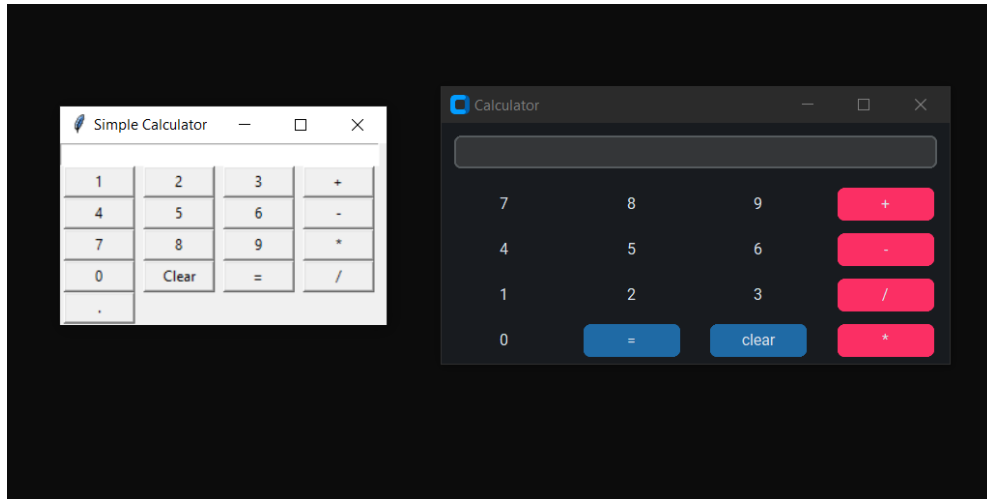


Figura 4.4. Comparación entre Tkinter y CustomTkinter [55]

5. Diseño

A continuación, se profundiza en cada una de las soluciones alcanzadas para llevar a cabo la implementación de los requisitos vistos anteriormente. Para ello, se va a explicar el diseño de las tres partes principales del proyecto, es decir, el hardware de control de la parte neumática, el hardware de la parte NIRS y el diseño de la GUI. En cada uno de los apartados se va a diferenciar el diseño de la parte *hardware* del diseño de la parte *software*, según sea necesario.

5.1 Diseño del hardware de control

En este caso se va a comenzar explicando el diseño implementado para el desarrollo de la PCB encargada de controlar todo el sistema de llenado y vaciado de mangas, así como de la recopilación de datos durante la prueba. Una vez hecho eso, se profundizará en el diseño del programa *software* que se ejecutará en el microprocesador de la placa fabricada.

5.1.1 Diseño hardware

En la Figura 5.1 se puede ver la PCB v0.2 que había sido fabricada hasta el momento de mi incorporación al grupo de trabajo y la cual se va a tomar como base para el desarrollo de la PCB v0.3 que forma parte del objetivo de este TFM.

Dicha placa se puede dividir en varias partes para así facilitar su comprensión:

- A. Red de alimentación. Es la parte de la placa encargada de convertir los 24 V de la fuente de alimentación en 5 V que necesita la placa para poder trabajar. Esta conversión se realiza a través de un *Buck converter* o convertidor reductor que permite reducir el voltaje de entrada a un nivel más bajo en la salida.
- B. Red de comunicación. Formada por un circuito integrado cuyo objetivo es la conversión de USB (*Universal Serial Bus*) a interfaz serie. En este caso se implementó para permitir la comunicación entre un conector micro-USB y la interfaz serie UART (*Universal Asynchronous Receiver/Transmitter*) del microprocesador.
- C. Red NIRS. Esta parte está preparada para que se pueda conectar esta placa al sistema NIRS desarrollado y que puedan cooperar y trabajar juntos. Sin embargo, para el desarrollo de este TFM se implementó el sistema NIRS como un sistema independiente, pero conservando las conexiones en la PCB v0.3 para futuras etapas del proyecto.
- D. Salidas digitales. Se necesitaba disponer de una serie de salidas digitales que permitiesen actuar sobre determinados elementos de la parte neumática del

dispositivo, por ejemplo, encender el ventilador del equipo cuando se cumplan una serie de condiciones establecidas.

- E. Interfaz manual. La placa v0.2 únicamente cuenta con dos botones, un *display LCD (Liquid Crystal Display)* y un *encoder*, es decir, un codificador que permite convertir la posición de un objeto en una señal eléctrica que puede ser leída y procesada por un sistema electrónico. De esta forma, el usuario pueda elegir la presión objetivo a aplicar en las mangas haciendo girar el *encoder*.
- F. Salidas analógicas. Se requiere de una serie de salidas analógicas debido a que el llenado y vaciado de las mangas está regulado por una serie de válvulas que trabajan con señales analógicas. Sin embargo, en esta placa está implementado el microprocesador Atmega2560, el cual no cuenta con señales analógicas puras, sino que emplea señales analógicas PWM. Estas son una forma de señales digitales que simulan una señal analógica mediante la variación del ancho de los pulsos en una secuencia de ciclos de trabajo. Además, a estas señales se le añadió una etapa de filtrado tratando así de estabilizar la señal.
- G. Entradas analógicas. Son una serie de entradas analógicas para la recopilación de los datos necesarios, por ejemplo, la presión alcanzada en cada una de las mangas o la presión del tanque de aire comprimido.

Sin embargo, como ya se comentó anteriormente, esta placa presenta una serie de limitaciones que deben ser subsanadas en la placa v0.3. A continuación, se detallan las soluciones que se han diseñado para cumplir con los requisitos de diseño establecidos.

Atendiendo al primer requisito, relativo a la limitación en el nivel de presión que alcanzaba el dispositivo, se concluyó que era debido a la limitación de tensión que presentaban las señales de las salidas digitales del microprocesador Atmega-2560 [42]. Este está diseñado para operar a un voltaje de suministro de 4,5 V a 5,5 V, por lo tanto, todas las señales trabajan con niveles de señal 0 V (bajo) y 5 V (alto).

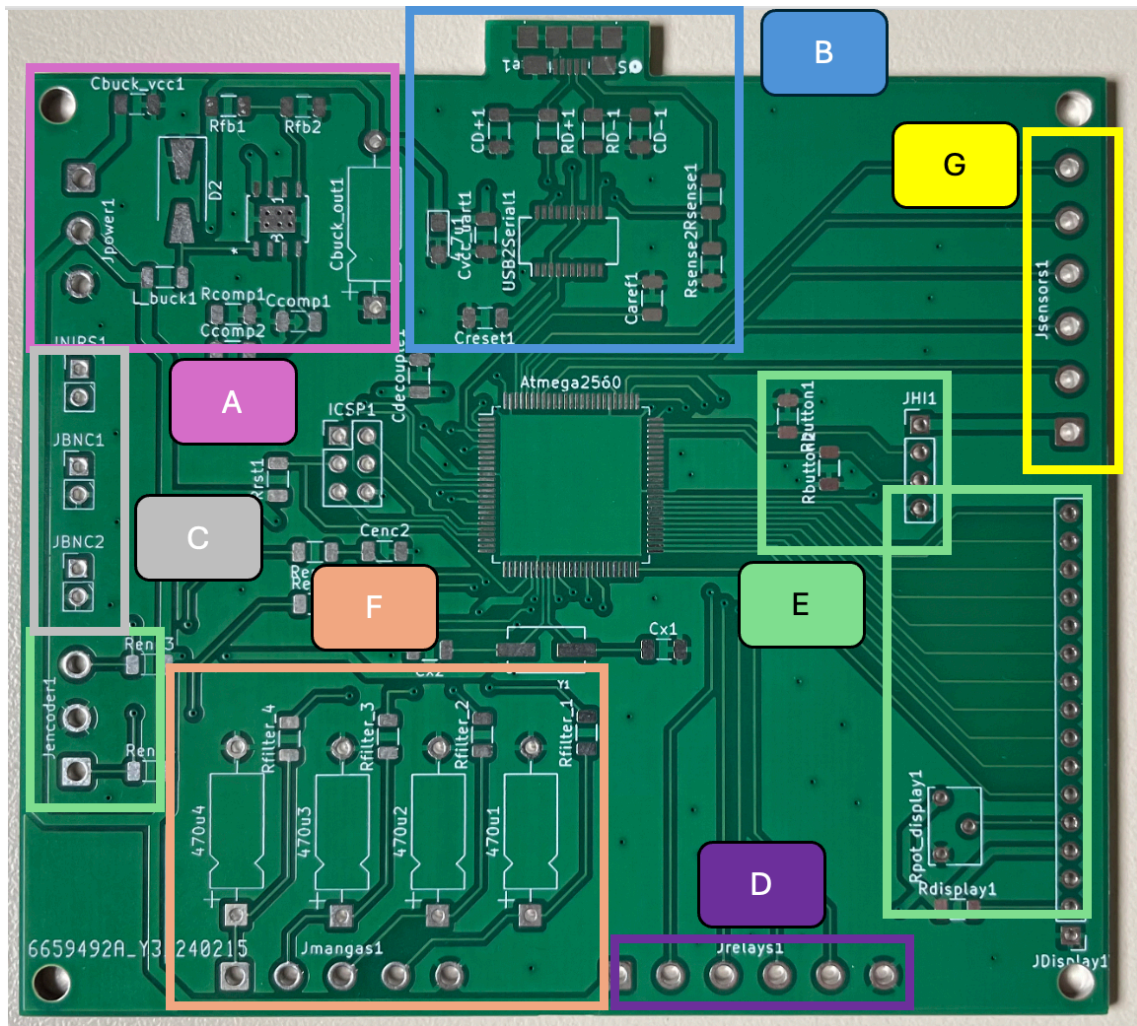


Figura 5.1. PCB v0.2 fabricada

Además, sabiendo las limitaciones del microprocesador para generar señales analógicas puras, se llegó a la conclusión que para implementar el primer requisito de diseño la mejor opción era incluir un DAC (*Digital-to-Analog Converter*) en nuestra PCB. Este conversor nos va a permitir convertir las señales digitales del microprocesador, que son discretas y basadas en valores binarios, en señales analógicas, que son continuas y pueden representar diferentes valores dentro de un rango determinado.

Otra razón que nos hizo decantarnos por el uso de este tipo de componentes es que aplicándole una tensión de referencia determinada nos iba a permitir aumentar el rango de la señal analógica de salida, con lo cual se podría alcanzar una tensión mayor a 5 V y solventar el problema que se tenía en la versión anterior.

Para llevar a cabo esta solución, se decidió eliminar la etapa de filtrado de la región F que se muestra en la Figura 5.1 e implementar el DAC AD5724AREZ [43]. Este componente, fabricado por *Analog Devices* [44], es un conversor de 12 bits, 4 canales, permitiéndonos así el manejo de hasta 4 mangas de presión, y voltaje de salida programable, pudiendo así superar la limitación de 5 V que se tenía anteriormente.

Para poder aplicarle al DAC una tensión de referencia superior a 5 V, concretamente se fijó la tensión objetivo en 7 V, se decidió reconfigurar toda la red de alimentación, es decir, la región A de la Figura 5.1. En la versión anterior se había diseñado para convertir de 24 V a 5 V. Sin embargo, para esta nueva versión se decidió incorporar una segunda fuente de alimentación externa de 5 V. De esta forma, se tenía la fuente de alimentación de 24 V para toda la parte neumática del dispositivo y una fuente de 5 V para alimentar nuestra nueva PCB v0.3. Al realizar esta modificación se podía reutilizar toda la red de alimentación para convertir de 24 V a 7 V, y así poder fijar la tensión de referencia que necesita el DAC para operar. En la Figura 5.2 se puede ver el conexionado en KiCad realizado para llevar a cabo la implementación de la nueva red de alimentación. El uso de una resistencia variable, *R_Pot_Buck_1*, nos va a permitir regular la tensión de salida del reductor de tensión hasta alcanzar los 7 V con los que se quiere trabajar.

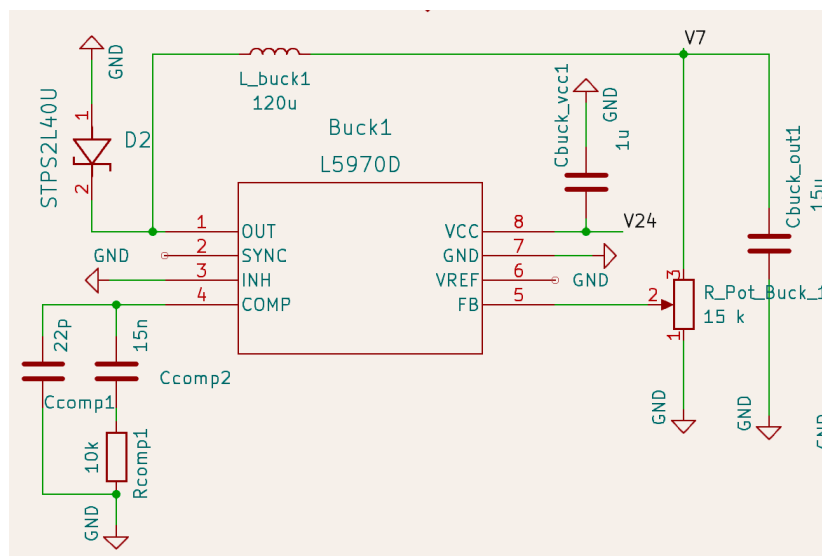


Figura 5.2. Diseño y conexionado de la red de alimentación en KiCad

Por otro lado, en la Figura 5.3 se puede ver como se realizó el conexionado del componente DAC dentro del entorno de KiCad, y en la Tabla 5.1 se detallan los pines y las conexiones necesarias para la correcta comunicación entre el DAC y el resto de los componentes de la placa.

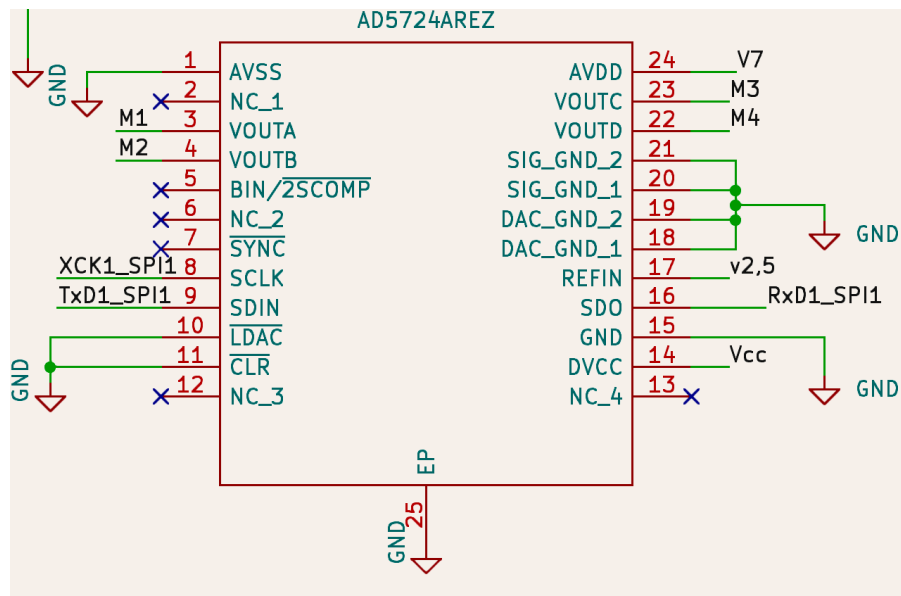


Figura 5.3. Conexionado del DAC en el entorno de KiCad

Una vez estudiado el conexionado necesario para la implementación del DAC es necesario comentar que dicho componente se va a comunicar con el microprocesador a través de la interfaz SPI del mismo. Por lo tanto, se van a configurar las conexiones MOSI (*Master Out Slave In*) que corresponde con el pin TX_SPI, el MISO (*Master In Slave Out*) que corresponde con el RX_SPI y, por último, SCLK (*Serial Clock*) con el XCL_SPI.

Además, uno de los pines del componente necesitaba ser alimentado con 2,5 V que corresponde la tensión interna del DAC. Esto no supuso mayor problema ya que a través de un simple divisor de tensión con dos resistencias 10 de kilo ohmios se puede convertir la tensión de alimentación de la placa de 5 V a la tensión requerida. En la Figura 5.4 se muestra el conexionado que fue necesario implementar en KiCad para terminar de completar el correcto conexionado del DAC.

PIN	CONEXIÓN
1	GND
2	NO CONECTADO
3	MANGA 1
4	MANGA 2
5	NO CONECTADO
6	NO CONECTADO
7	NO CONECTADO
8	XCK_SPI
9	TX_SPI
10	GND
11	GND
12	NO CONECTADO
13	NO CONECTADO
14	5V
15	GND
16	RX_SPI
17	2,5 V
18	GND
19	GND
20	GND
21	GND
22	MANGA 4
23	MANGA 3
24	7 V

Tabla 5.1. Conexión de los pines del DAC

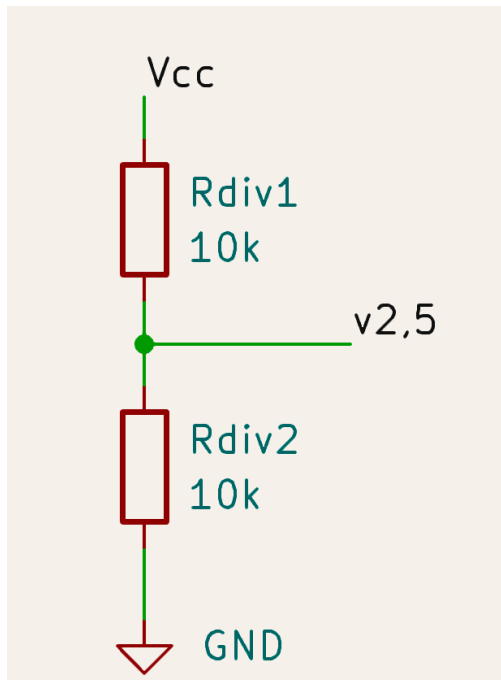


Figura 5.4. Divisor de tensión diseñado en KiCad

Cuando se terminó de realizar el conexionado de cada uno de los componentes, fue necesario llevar a cabo el enrutamiento de los mismos. Aunque posteriormente se mostrará el enrutado general de la placa, ahora se muestra con mayor precisión las dos partes de la placa que están siendo estudiadas. En la Figura 5.5 se puede ver el enrutamiento de la red de alimentación comentada. En el punto A está llegando la alimentación de 24 V y Vcc (5 V) proveniente de las dos fuentes de alimentación externas. Sin embargo, se puede ver como la línea de 24 V entra dentro de la red de reducción y a la salida, en el punto B, se obtienen los 7 V con los que opera el DAC implementado.

También es necesario llevar a cabo el enrutado del conversor DAC, el cual se puede ver en la Figura 5.6. Para este caso simplemente es necesario conectar los pines según corresponda, ya sea a otros pines del microprocesador, a los pines de salidas analógicas o a las líneas de alimentación según la Tabla 5.1.

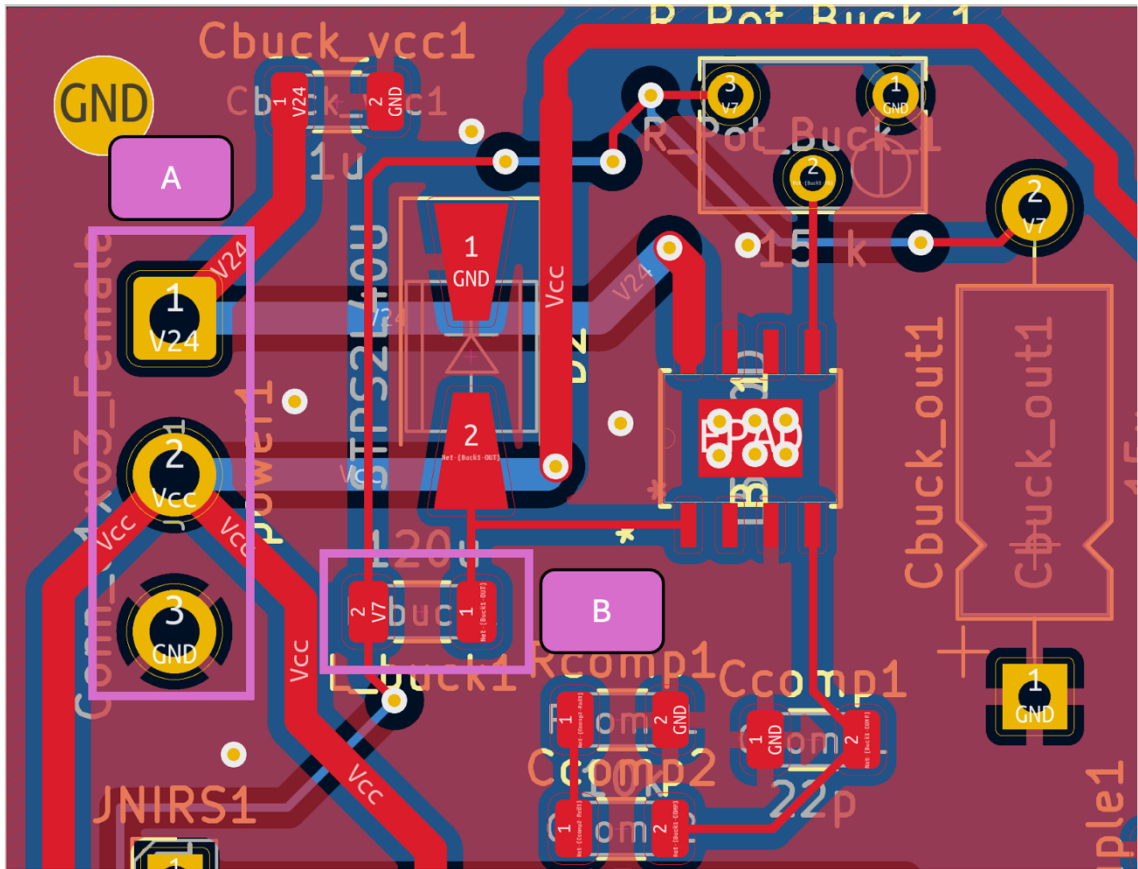


Figura 5.5. Enrutado de la red de alimentación

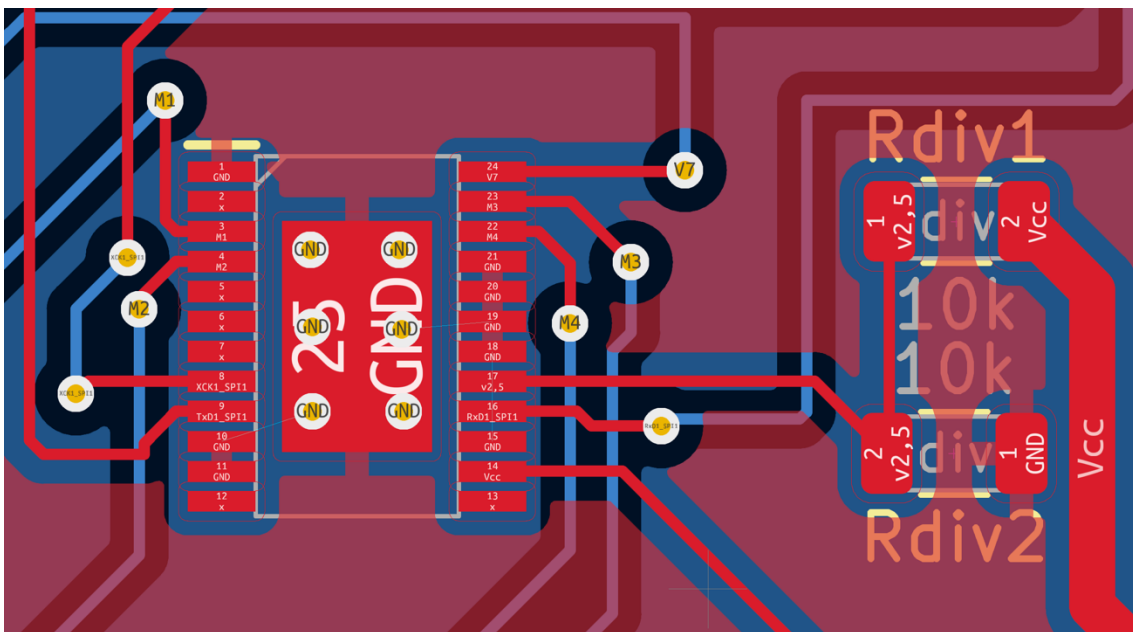


Figura 5.6. Enrutado del DAC

Llegados a este punto, ya se tiene implementada la solución para solventar el primer requisito de la nueva placa v0.3. Sin embargo, aún quedan otros dos que necesitan ser estudiados. El primero de ellos recogía la necesidad de contar con una interfaz manual para el manejo del dispositivo, y el otro, que se implementara un elemento que permita al sujeto que realiza la prueba, interactuar con el equipo en un determinado momento. Atendiendo a estas necesidades, se decidió implementar la solución que se muestra en el esquema de la Figura 5.7

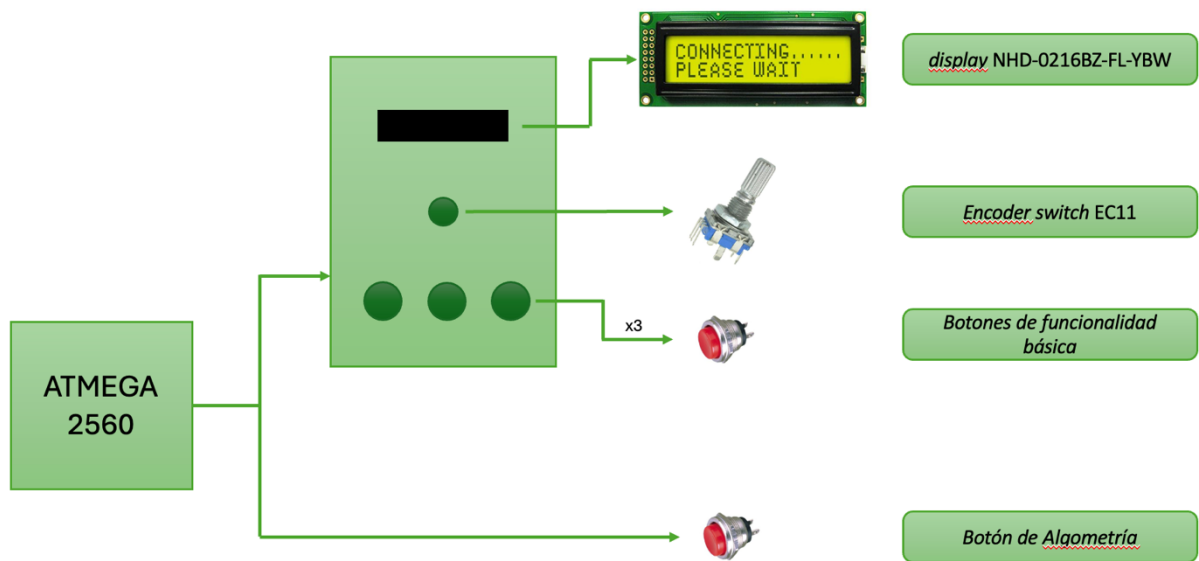


Figura 5.7. Esquema de diseño de la interfaz manual

Como se puede ver en la Figura 5.7, se ha decidido implementar una interfaz manual que cuente con un *display*, que permite leer datos e interactuar con el dispositivo, un *encoder*, utilizado para seleccionar la presión que se quiere alcanzar y la manga con la que se quiere trabajar, ya sea una de ellas o ambas, y, por último, tres botones, uno para el llenado, otro para el vaciado y otro para futuras implementaciones del diseño.

En cuanto al *display*, su conexión con el microprocesador ya se encontraba implementado en la placa v0.2 con lo cual no fue necesario realizar ninguna modificación. Por otro lado, para el caso del *encoder* sí que fue necesario modificar la versión anterior. El *encoder* rotatorio EC11 [45], utilizado en este proyecto, es un

dispositivo electromecánico que convierte el movimiento de un eje en señales de pulsos. Estos pulsos pueden ser interpretados por un microcontrolador para determinar la dirección y velocidad de rotación. Además, incluye un botón pulsador integrado que permite al usuario tanto rotar como presionar el eje. Se ha elegido este componente ya que, debido a sus características, permite usar su rotación para seleccionar el valor de presión que se quiere aplicar y su botón para indicar que manga se quiere utilizar, la derecha, la izquierda o ambas.

Teniendo en cuenta el conexionado de pines que requiere este componente, Tabla 5.2, es necesario modificar la PCB ya que antes únicamente se utilizaban tres pines que permitían detectar la rotación del *encoder*. Ahora estos pines se van a ampliar para así poder emplear todas las funcionalidades del componente.

En primer lugar, se empieza modificando el esquema de conexión en el entorno de KiCad. Se necesita añadir dos pines adicionales, uno (ENC 3) conectado al micro a través de una resistencia de 10 kilo ohmios para reducir el efecto rebote clásico presente en los botones al pulsar sobre ellos y otro pin conectado a tierra. En la Figura 5.8 se muestra el conexionado final del componente y en la Figura 5.9 se encuentra, resaltado en blanco, todo el enrutamiento correspondiente a este.

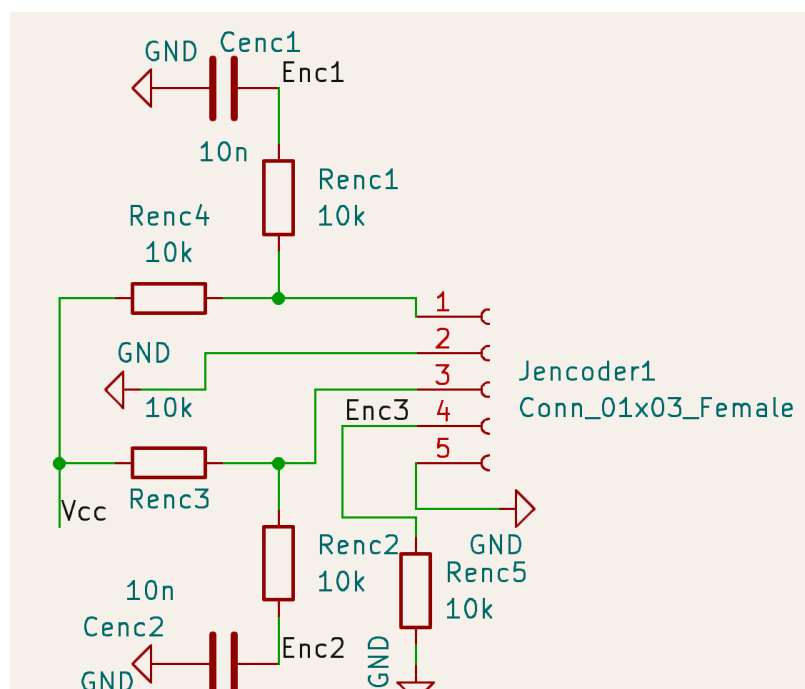


Figura 5.8. Esquema de conexión del encoder en KiCad

PIN	CONEXIÓN
1	ENC 1
2	GND
3	ENC 2
4	ENC 3
5	GND

Tabla 5.2. Conexión de los pines del encoder

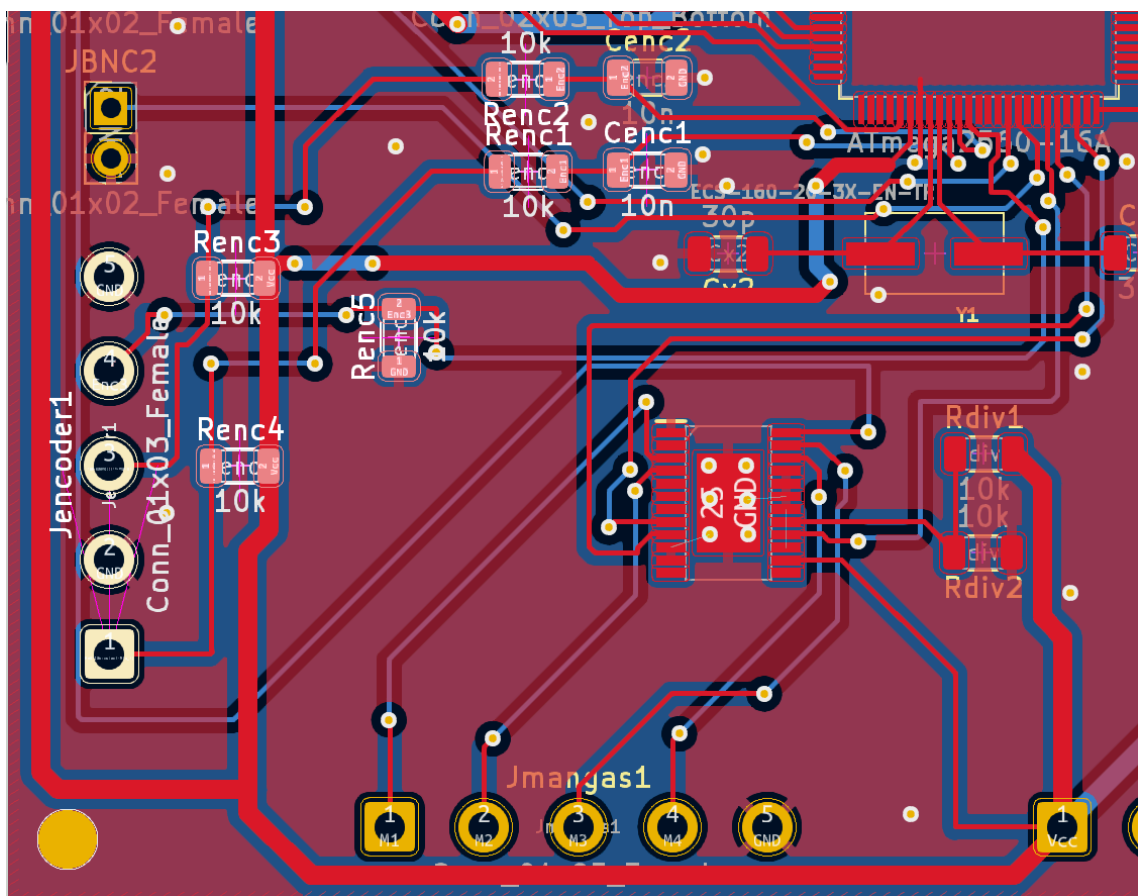


Figura 5.9. Enrutamiento correspondiente al encoder

Por último, se llegó a la conclusión de que la mejor solución posible para implementar la necesidad de que el usuario interactuase con el equipo durante la prueba era el uso de un botón, denominado botón de algometría. Sin embargo, este botón a diferencia de los otros tres no puede encontrarse ubicado en la carcasa del dispositivo, por lo tanto, se va a instalar mediante un cable. Así se consigue que el dispositivo esté en un lugar seguro y que el sujeto realice la prueba física de manera

cómoda sujetando un pequeño botón que le permite interactuar con nuestro equipo. Teniendo esto en cuenta, el número total de botones que se deben implementar son cuatro, tres para la interfaz manual y uno adicional para las pruebas de algometría.

En la versión anterior de la placa ya se había implementado dos botones con lo cual para esta versión actual se debían añadir dos más. Los botones son dispositivos de entrada muy simples, que se utilizan para cerrar un circuito cuando se presionan. Por lo tanto, cada botón requiere de dos pines y una resistencia configurada como *pull-down*. Un pin del botón se encuentra conectado a la fuente de voltaje, en este caso, 5 V, y el otro pin se conecta a una entrada digital del microprocesador y a una resistencia *pull-down* que va a tierra, siguiendo el esquema de la Figura 5.10. Esta configuración permite que cuando el botón no está presionado, el pin de entrada digital se mantiene en estado bajo (0 V) gracias a la resistencia comentada. Cuando el botón se presiona, el pin de entrada digital se conecta a 5 V, registrando un estado alto. En la Figura 5.11 se puede ver el conexionado final realizado en el entorno de KiCad y su enrutado en el diseño de la PCB.

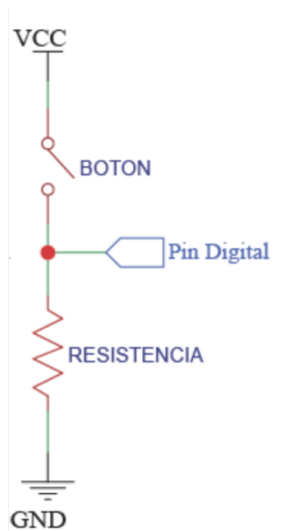


Figura 5.10. Conexión del botón para configuración *pull-down* [56]

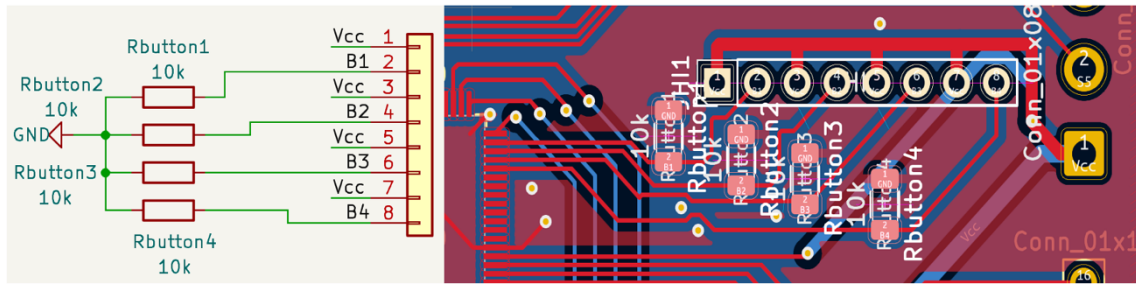


Figura 5.11. Conexionado y enrutamiento final de los botones.

Una vez hecho esto, la nueva placa v0.3 está completamente diseñada y cumpliendo con todos los requisitos previamente establecidos. El resultado final obtenido, tanto en el conexionado como en el enrutamiento, se puede ver en la Figura 5.12 y la Figura 5.13. A continuación, se procederá al desarrollo del software necesario para programarla. Este programa será esencial para completar la parte del sistema responsable de manejar la neumática del dispositivo.

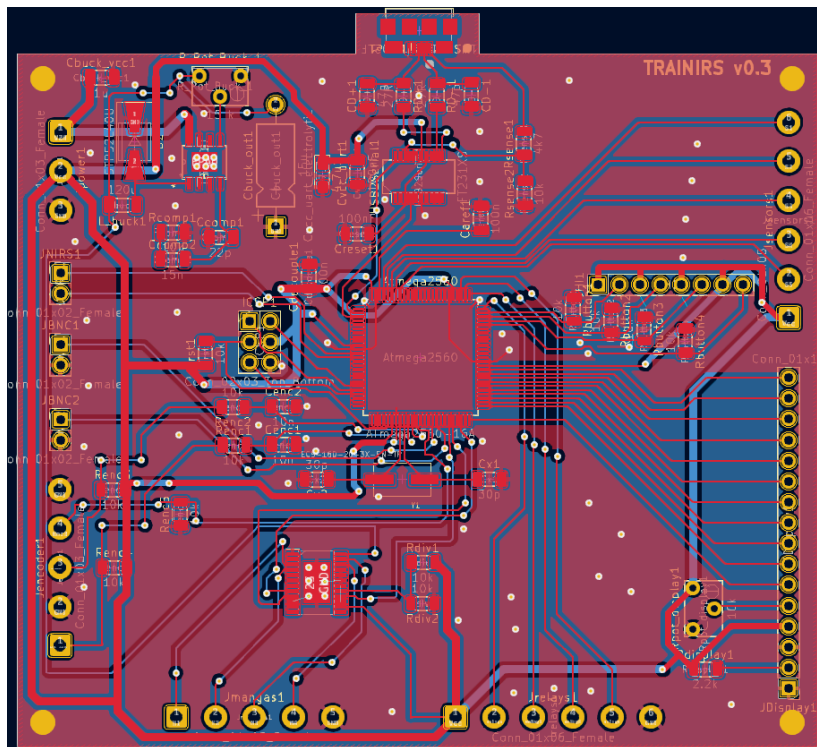


Figura 5.12. Enrutado final de la placa v0.3

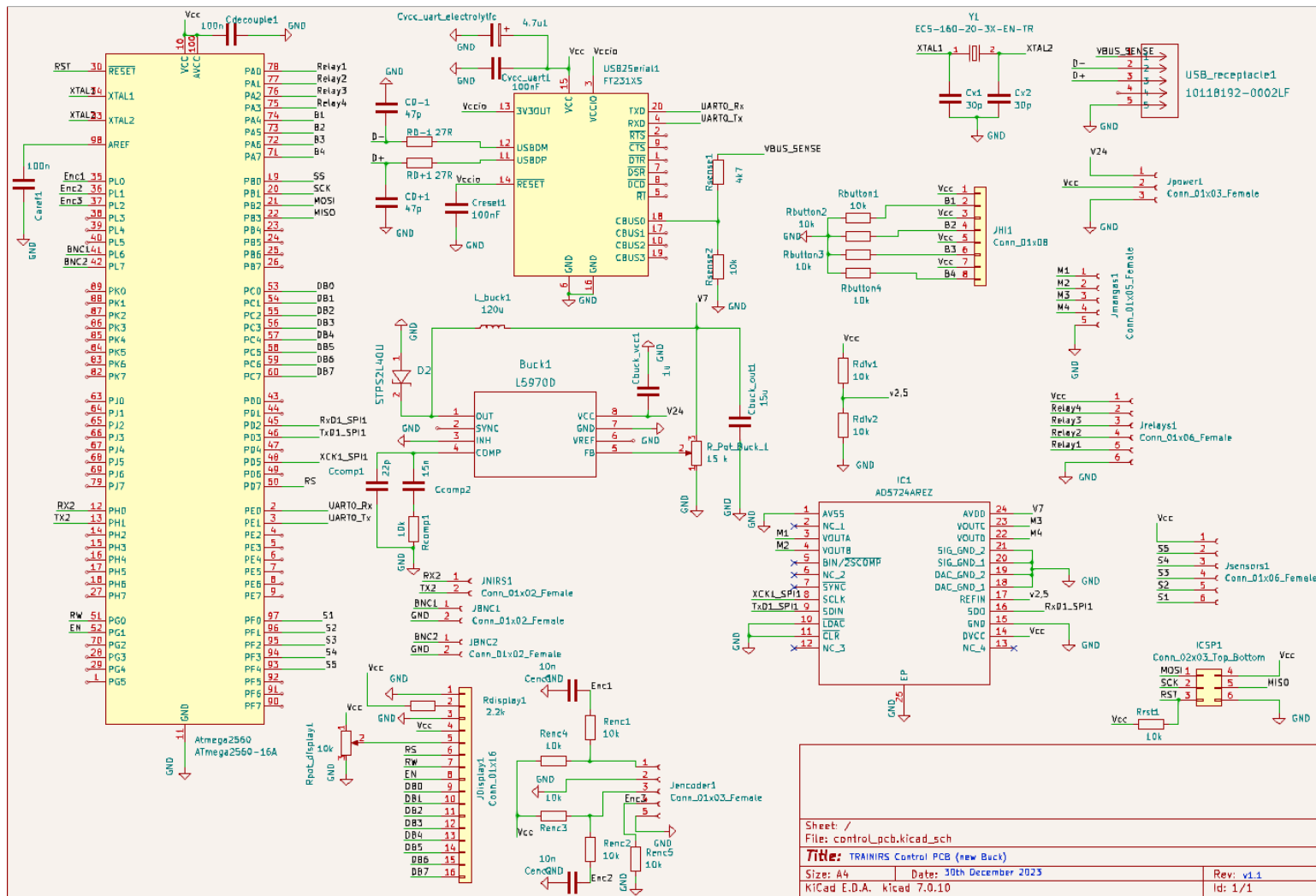


Figura 5.13. Conexionado final de la PCB v0.3

5.1.2 Diseño software

Para el desarrollo de esta parte *software* se lleva a cabo un enfoque de programación en el que múltiples tareas o procesos se ejecutan de manera cooperativa en un solo núcleo de CPU (*Central Processing Unit*), llamado *Collaborative Multitasking* o multitarea colaborativa. Con esta técnica, múltiples tareas cooperan entre sí para compartir el tiempo de ejecución de forma eficiente a diferencia de la multitarea con interrupciones, donde un sistema operativo o un núcleo se encarga de interrumpir y cambiar de tarea en función de prioridades o tiempos asignados [46]. La multitarea colaborativa facilita el diseño y disminuye el uso de recursos, algo especialmente útil en sistemas con recursos limitados, aunque es esencial que cada tarea esté bien planificada para ceder el control del procesador en el momento adecuado y así evitar el monopolio de los recursos, garantizando así un sistema más eficiente y receptivo.

Siguiendo este paradigma de programación, se propone hacer uso de un scheduler, mediante la librería *TaskScheduler* de Arduino [47], que gestione las tareas a llevar a cabo con cada restricción temporal y el uso de tareas no bloqueantes. En la Figura 5.14 se puede ver un ejemplo de cómo se llevaría a cabo la gestión de las tareas siguiendo el esquema comentado.

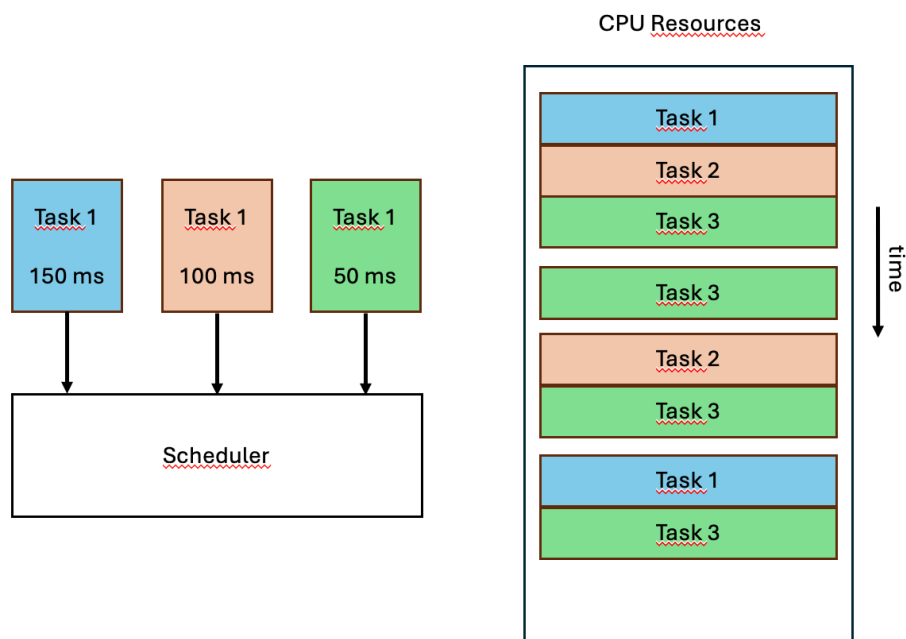


Figura 5.14. Ejemplo de funcionamiento de scheduler con tareas de diferentes periodos

A continuación, se va a desglosar el código implementado para tratar de explicarlo de la forma más clara posible.

5.1.2.1. Gestión de la configuración de los pines

Como ya se comentó anteriormente se está empleando Arduino, con lo cual el código cuenta con una rutina *setup()* que llevará a cabo la inicialización de los pines de entrada/salida del sistema. Para ello, se definió una función genérica, llamada *initialize_pins*, que recibe como parámetro de entrada un array con todos aquellos pines a iniciar y el tipo de configuración, en este caso, *input* u *output*. En total existen 4 tipos de arrays, uno por cada tipo de pin que se tiene: *digital/analog* o *input/output* y sus diferentes combinaciones. Por ejemplo, el array para los pines digitales de entrada se declararía de la siguiente forma:

```
int input_digital_pins[] = {PIN_A, PIN_B, PIN_B};
```

Buscando simplificar el proceso, los pines se definieron mediante directivas de preprocesador tipo *#define*, transformándolos en constantes. De acuerdo con el ejemplo presentado anteriormente, cada tipo de pin se inicializará dentro de una rutina *setup()* de la siguiente manera:

```
Initialize_pins( input_digital_pins, INPUT);
```

5.1.2.2. Gestión de la asignación de las tareas

Ya que se hace uso de la librería *TaskScheduler*, las tareas deben ser primero añadidas a una instancia *scheduler* y posteriormente activadas. Para ello, se usó *arrays* de punteros a *Task*, la cual es una clase empleada por el *scheduler*, por ejemplo:

```
Task* tareas[] = {&tarea1, &tarea2};
```

En este caso, cada tarea será de la clase *Task*. Esta clase de la librería *TaskScheduler* precisa que se le pase por parámetros el periodo de ejecución de cada tarea, el tipo de ejecución y un puntero a la función que tiene que ejecutar. Por ejemplo,

la tarea1, ejecutada cada PERIODO segundos y de forma constante (TASK_FOREVER) se declara de la siguiente manera:

Task tarea1(PERIODO, TASK_FOREVER, &function1);

5.1.2.3. Implementación de funciones

A continuación, en la Tabla 5.3 se recogen todas las funciones implementadas en el código usando la estructura de tareas comentada anteriormente, su funcionamiento principal y el periodo de ejecución de cada una de ellas.

Tarea	Descripción	Periodo
<i>Samplig_task</i>	Adquisición de las medidas de presión de las mangas y envío al ordenador a través del puerto USB.	100 ms
<i>Human_interfacing_task</i>	Manejo de los botones y del <i>encoder</i> de presión objetivo de la interfaz manual, cuyo diseño y funcionalidad se pueden ver en la Figura 5.15. Únicamente maneja y actualiza <i>flags</i> asociados a los botones	1 ms
<i>Pressure_setting_task</i>	Establecimiento de la presión seleccionada en las mangas, en una de ellas o en ambas, según los <i>flags</i> activados por el resto de las tareas.	100 ms
<i>Remote_control_task</i>	Recepción de los comandos provenientes del ordenador de control y activación de <i>flags</i> en consecuencia para ejecutar la tarea correspondiente, llenado o vaciado de mangas.	100 ms
<i>Compresor_activation_task</i>	Activación y desactivación del compresor cuando los niveles de presión superen o desciendan	1000 ms

	por debajo de los valores establecidos	
<i>Fan_function_task</i>	Activación del ventilador del equipo en el momento que el compresor se activa	1000 ms
<i>Purge_function_task</i>	Activación de la electroválvula de purga, proceso necesario para la refrigeración de la parte mecánica, tras la activación del compresor un total de 10 veces	1000 ms
<i>Evaporator_function_task</i>	Activación de la válvula de evaporación durante un tiempo determinado en el momento que se activa la electroválvula de purga	1000 ms
<i>LCD_function_task</i>	Presentación en la pantalla de la información del proceso actual. Como se puede ver en la Figura 5.15 muestra información sobre la presión objetivo, la presión real obtenida en cada manga, sobre la manga que se está trabajando y el tiempo de duración de la isquemia	100 ms
<i>Counter_ischemia_task</i>	Cálculo del tiempo que dura cada uno de los procesos de isquemia realizados	1000 ms
<i>Algometer_task</i>	Activación del <i>flag</i> de algometría en el momento que el usuario pulsa el botón implementado para dicha función	100 ms
<i>Switch_task</i>	Selección de la manga con la que trabajar al pulsar el botón del eje del <i>encoder</i> .	100 ms

Tabla 5.3. Resumen de tareas del TaskScheduler

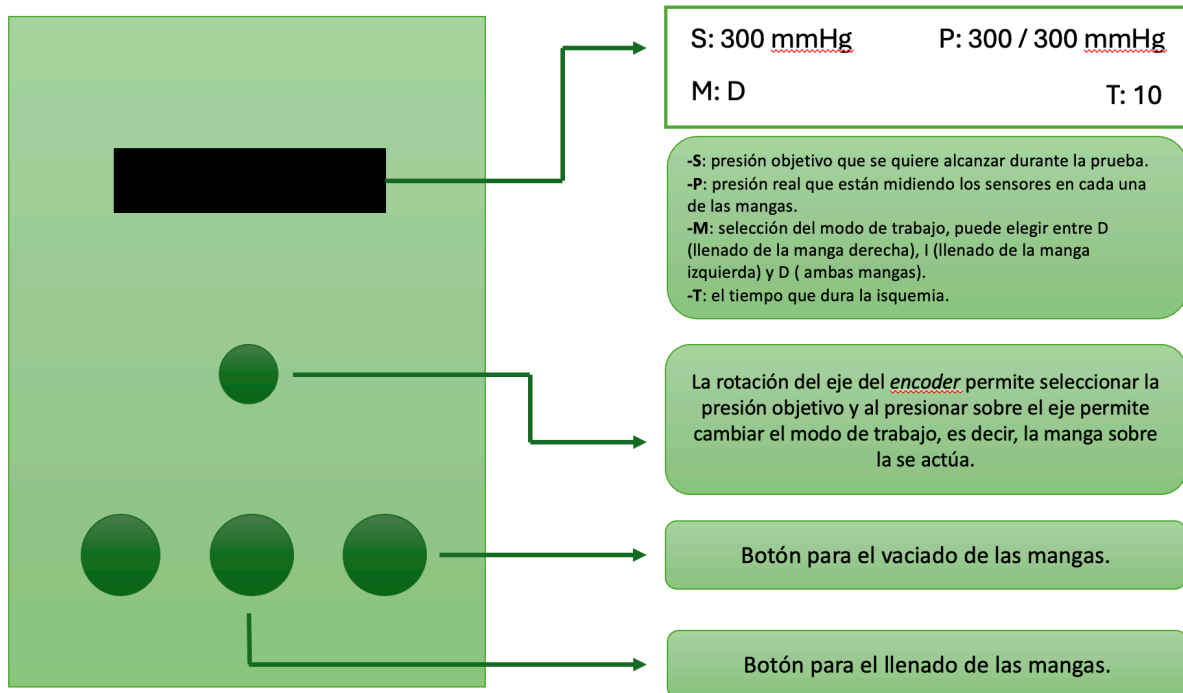


Figura 5.15. Funcionalidad de la interfaz manual diseñada

Por último, es necesario comentar que la tarea *sampling_task*, una vez adquirida todas las medidas necesarias, prepara una trama con el formato que se muestra a continuación para su envío al ordenador de control a través del puerto USB. Una vez se reciben los datos en el ordenador, es la interfaz gráfica la encargada de tratar esos datos y operar con ellos. La implementación de esta interfaz gráfica se mostrará en capítulos posteriores.

[*tiempo, presión_medida_A, presión_objetivo_A, presión_medida_B, presión_objetivo_B, presión_tanque, estado_compresor, contador_compresor, flag_algoemtría, timer_mangaA, timer_mangaB*]

5.2 Diseño del hardware NIRS

En este apartado se va a profundizar en el diseño implementado para el desarrollo de las sondas de medida empleando tecnología NIRS. Además, también se mostrará el diseño del *software* necesario para el control del *hardware* NIRS y adquisición de los datos de medida necesario.

5.2.1 Diseño hardware

Anteriormente ya se explicaron los elementos *hardware* básicos que permiten cumplir con los dos primeros requisitos de diseño y que se van a utilizar para el desarrollo de las sondas de medida. Estos dos requisitos de diseño fijaban la necesidad de trabajar con varios LEDs de diferentes longitudes de onda que permitiesen una mayor precisión y capacidad de análisis de la oxigenación del tejido y un fotorreceptor con la sensibilidad necesaria que permitiese la detección de los rayos reflejados en las células musculares.

Una vez solventada esa primera parte, se tiene que garantizar que dichos componentes cumplen con las normativas de seguridad vigente. Para ello se determinó que la potencia óptica emitida por los LEDs debe ser menor a 10 mW para asegurarnos así que en ningún caso se puedan producir quemaduras en la piel.

Fue necesario diseñar una alternativa al conexionado directo de los diodos con el pin digital del Arduino Mega porque la corriente de trabajo de los diodos superaba el límite máximo permitido por el dispositivo. Tras valorar diferentes opciones, se determinó que la mejor solución posible para llevar a cabo el diseño de las sondas era la implementación de una serie de espejos de corriente que permitan limitar la corriente que circula por el LED y así hacerlo trabajar en la región deseada. La implementación de un espejo de corriente, cuyo esquema básico se muestra en la Figura 5.16, es bastante sencilla. Su funcionamiento se basa en el uso de transistores, en este caso bipolares BJT (*Bipolar Junction Transistor*), configurados de tal manera que una corriente de referencia en un transistor se refleja en otro, creando así una corriente de salida igual a la corriente de referencia. Por lo tanto, calculando el valor de la resistencia de referencia se puede fijar el valor de la corriente. A continuación, se detalla el cálculo matemático realizado para determinar el valor de la resistencia de cada uno de los LEDs.

Para la resolución de este tipo de esquema se tiene la siguiente ecuación matemática:

$$I_{ref} = \frac{V_{cc} - V_{BE1}}{R_{ref}} \times \frac{1}{1 + \frac{2}{\beta_F}}$$

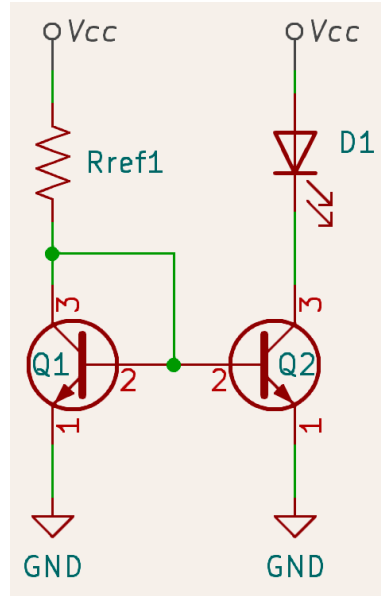


Figura 5.16. Conexión del espejo de corriente implementado

En el caso de que $\beta_F \gg 2$, en los BJT 337-16 [48] utilizado suele tener un valor entre 100-250. Por lo tanto, el valor obtenido de la segunda expresión de la ecuación es muy cercano a 1 siendo prácticamente despreciable. En consecuencia, la expresión matemática anterior quedaría de la siguiente manera:

$$I_{ref} = \frac{V_{cc} - V_{BE1}}{R_{ref}}$$

Si se despeja el valor de R_{ref} de la expresión, que es el valor que se quiere calcular, y se tiene en cuenta que en los transistores BJT de silicio la tensión base-emisor es aproximadamente 0,7 V se obtiene la siguiente expresión:

$$R_{ref} = \frac{V_{cc} - 0,7}{I_{ref}}$$

El último paso para poder obtener el valor final de cada una de las resistencias es conocer la corriente que debe circular por cada LED, la cual coincide con la I_{ref} como se explicó anteriormente. Para determinar el valor de esta se va a hacer uso de las hojas

de datos de cada uno de los LEDs ya que suelen incluir una gráfica que permite relacionar la corriente con la potencia óptica emitida.

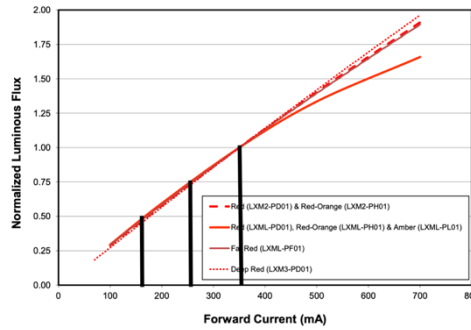


Figura 5.17. Gráfico corriente-potencia óptica del LED de 735 nm [34]

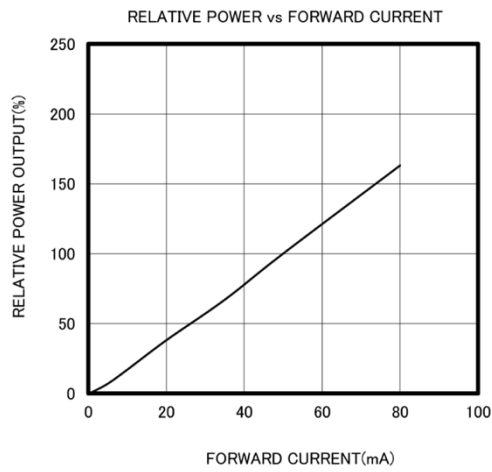


Figura 5.18. Gráfico corriente-potencia relativa del LED de 810 nm [35]

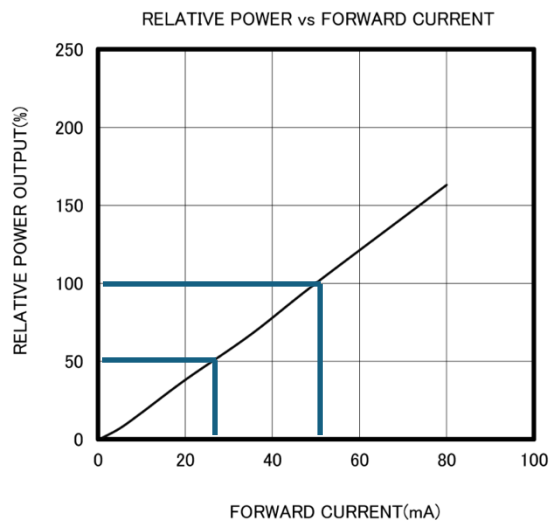


Figura 5.19. Gráfico corriente-potencia relativa del LED de 850 nm [36]

El componente elegido para operar a 735 nm es un LED de alta potencia y como se puede ver en la Figura 5.17 a 350 mA se obtiene el 100% de potencia óptica, que mirando en su hoja de datos [34] corresponde con 260 mW. Se fijó su potencia óptica de trabajo en 7 mW, sin embargo, al ser utilizado normalmente para altas potencias, los rangos buscados de operación no se encuentran representados. Ante esta situación, se decidió obtener el valor de corriente a través de una regresión lineal del tipo $P = \alpha \times I$.

La regresión lineal es una técnica estadística que nos permite encontrar la función lineal, es decir, la recta que mejor se ajuste a un conjunto de datos. Este conjunto de datos se obtiene de los puntos conocidos que sí se representan en la gráfica de la Figura 5.17. En la Tabla 5.4 se recogen los puntos necesarios para realizar el cálculo correspondiente.

	<i>I</i>	<i>P</i>	<i>I</i> × <i>P</i>	<i>I</i> ²
	150	130	19500	22500
	250	195	48750	62500
	350	260	91000	122500
Σ	750	585	159250	207500

Tabla 5.4. Puntos de trabajo conocidos del LED-735

Con el grupo de puntos anterior ya se pueden calcular el valor de α , a través de la siguiente expresión matemática:

$$\alpha = \frac{\sum_{i=1}^n I_i P_i}{\sum_{i=1}^n I_i^2}$$

Obteniendo que $\alpha = 0,7675$. Por la tanto, despejando de la ecuación $P = \alpha \times I$ se obtiene que para una potencia óptica de 7 mW se debe operar con una corriente de 9,12 mA. Por último, a través de la expresión que se obtuvo anteriormente, se calcula el valor de la R_{ref} para el LED de 730 nm:

$$R_{ref750} = \frac{V_{cc} - 0,7}{I_{ref}} = \frac{5 V - 0,7 V}{9,12 mA} = 471,49 \Omega$$

A continuación, se procede a calcular el valor de R_{ref} para el segundo de los LEDs, es decir, el de 810 nm. Sin embargo, para este caso, debido a las características del LED el cálculo es mucho más sencillo. Tras analizar la hoja de datos del componente [35] y comprobar la gráfica de la Figura 5.18 se pudo determinar que el 100% de la potencia óptica se obtiene al trabajar con una corriente de 50 mA, que corresponde con una potencia óptica de 4,5 mW, que se encuentra dentro del rango de seguridad establecido de 10 mW. Por la tanto, conociendo la corriente de referencia con la que se quiere trabajar ya se puede aplicar la fórmula como se hizo para el caso anterior y obtener el valor de R_{ref} como se muestra a continuación:

$$R_{ref810} = \frac{V_{cc} - 0,7}{I_{ref}} = \frac{5 V - 0,7 V}{50 mA} = 86 \Omega$$

Por último, quedaría calcular la resistencia de referencia para el LED de 850 nm fijando su potencia óptica en 4,5 mW, como en el caso anterior. Al observar la Figura 5.19 se aprecia que se trata de una recta perfecta, con lo cual se decide obtener la ecuación de la recta y así obtener el valor de corriente más preciso posible. Para ello se obtiene dos puntos conocidos de la gráfica y se calcula la pendiente de la recta como se muestra a continuación:

$$y = m \times x$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{100 - 50}{50 - 25} = 2$$

De la hoja de datos del componente [36] se obtiene que el 100% de su potencia óptica corresponde con 15 mW, por la tanto, 4,5 mW, que son lo que se quieren obtener corresponde con el 30% de su potencia máxima. Este 30% corresponde con el valor y de la ecuación anterior, teniendo así ya todos los valores necesarios para obtener x :

$$y = m \times x$$

$$x = \frac{y}{m} = \frac{30}{2} = 15 mA$$

Como sucedió en el resto de los casos, una vez calculada la corriente de referencia, ya se puede obtener el valor de la resistencia:

$$R_{ref850} = \frac{V_{cc} - 0,7}{I_{ref}} = \frac{5\text{ V} - 0,7\text{ V}}{15\text{ mA}} = 286,67\ \Omega$$

LED	Potencia	Corriente	Resistencia
735	7 mW	9,12 mA	471,49 Ω
810	4,5 mW	50 mA	86 Ω
850	4,5 mW	15 mA	286,67 Ω

Tabla 5.5. Resumen de los valores obtenidos para el diseño de los espejos de corriente

En la Tabla 5.5 se muestra un resumen de las potencias, corrientes y resistencias obtenidas para cada uno de los LEDs utilizados. Una vez diseñada toda la parte correspondiente a los emisores, es necesario realizar el diseño de la parte receptora.

El fotorreceptor es un componente que es capaz de reaccionar a la luz generando una corriente proporcional a la luz recibida. Teniendo en cuenta que la entrada analógica de nuestro microcontrolador solo es capaz de leer niveles de tensión, se decidió implementar un divisor de tensión entre el fotorreceptor y una resistencia variable, como se muestra en la Figura 5.20.

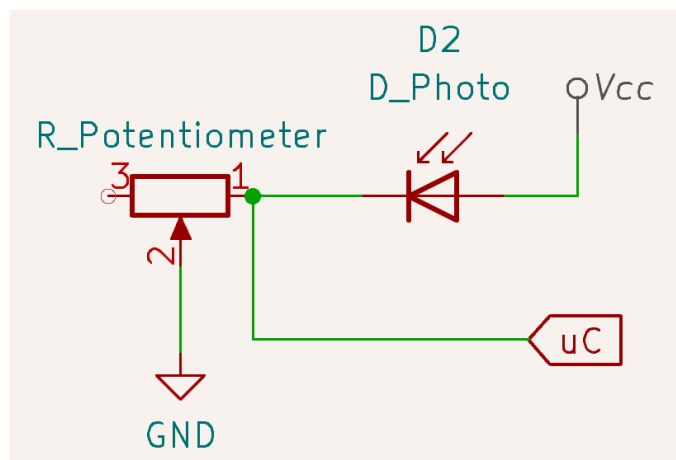


Figura 5.20. Esquema diseñado para el fotorreceptor

La resistencia del fotorreceptor varía en función de la luz recibida, bajo condiciones de poca iluminación, la resistencia aumenta y bajo condiciones de mucha iluminación la resistencia disminuye. Por otro lado, la otra resistencia variable que forma parte del divisor de tensión se varía de forma manual para así ajustar la sensibilidad del receptor, obteniendo los mejores resultados posibles.

Con esta configuración se consigue que en el punto de unión entre el fotorreceptor y la resistencia variable caiga un nivel de tensión proporcional a la iluminación detectada por el fotorreceptor y que pueda ser detectada por el microcontrolador.

Llegados a este punto, ya se tienen todos los componentes que forman parte de la sonda diseñados. Sin embargo, es necesario atender al resto de requisitos *hardware* del diseño para poder dar por concluido el diseño de las sondas. En consecuencia, es necesario estudiar que soluciones se pueden llevar a cabo para garantizar la robustez del sistema montado siendo capaz de soportar las condiciones típicas de una prueba física.

Se debe tener en cuenta que los LEDs y el fotorreceptor irán soldados en una PCB, cuya implementación se mostrará en capítulos posteriores, con lo cual se decidió que la mejor opción para tratar de garantizar la seguridad de los componentes era fabricar una sencilla caja mediante impresión en 3 dimensiones que permita resguardar los componentes.

Por otro lado, se pensó también que era posible la existencia de tensiones en los cables ya que la persona lleva las sondas pegadas al cuerpo y está en constante movimiento. La solución alcanzada para tratar de solventar este problema fue la incorporación de conectores tipo DIN (*Deutsches Institut für Normung*) de 8 hilos que absorbiera las posibles tensiones que se pudieran originar durante la prueba. En el siguiente capítulo 5 se mostrará como se llevó a cabo la implementación de estas soluciones y el resultado final obtenido.

5.2.2 Diseño software

Es necesario diseñar un pequeño programa *software* que permita dar solución a los dos últimos requisitos de diseño que faltan por implementar. Este programa se va a ejecutar en un Arduino Mega de forma independiente a la PCB de control vista anteriormente. Este software tiene que ser capaz de controlar tanto el encendido y apagado de los LEDs como la recepción de los haces de luz en el fotorreceptor. Todos los datos recogidos se enviarán a través de la UART al ordenador personal y se almacenarán en un fichero CSV (valores separados por coma).

Según los requisitos de diseño, todo el proceso de captura de datos de cada una de las longitudes de onda y envío de los datos debe tener un periodo de 10 ms. Además, se debe incorporar algún mecanismo que permita garantizar el sincronismo con el resto de los equipos del laboratorio. Teniendo esto en cuenta, se ha decidido introducir una señal de sincronismo en el programa de tal forma que al inicio de cada proceso de captura de datos se va a enviar un pulso activo al resto de los equipos conectados. Este pulso actúa como una marca temporal ya que le permite saber a los dispositivos a que instante de tiempo corresponde cada medida.

En la Figura 5.21 se muestra gráficamente el funcionamiento completo del programa, como se puede ver, se diseñó para manejar dos sondas, denominadas sonda A y sonda B. Ambas sondas son idénticas, sin embargo, se decidió montar dos para poder ajustar la distancia entre emisor y receptor y así poder trabajar con cualquier tipo de músculo. Como se explicó anteriormente, esta distancia va a determinar la profundidad que se pueda alcanzar dentro del músculo. Además, para tener una recopilación de datos más completa cuando se encienda el primer LED de la sonda A se harán lecturas con el fotorreceptor de ambas sondas, de esta forma se consigue tener una medida de la oxigenación de la superficie (sonda A) y del músculo interno (sonda B) y viceversa cuando se enciendan los LEDs de la sonda B.

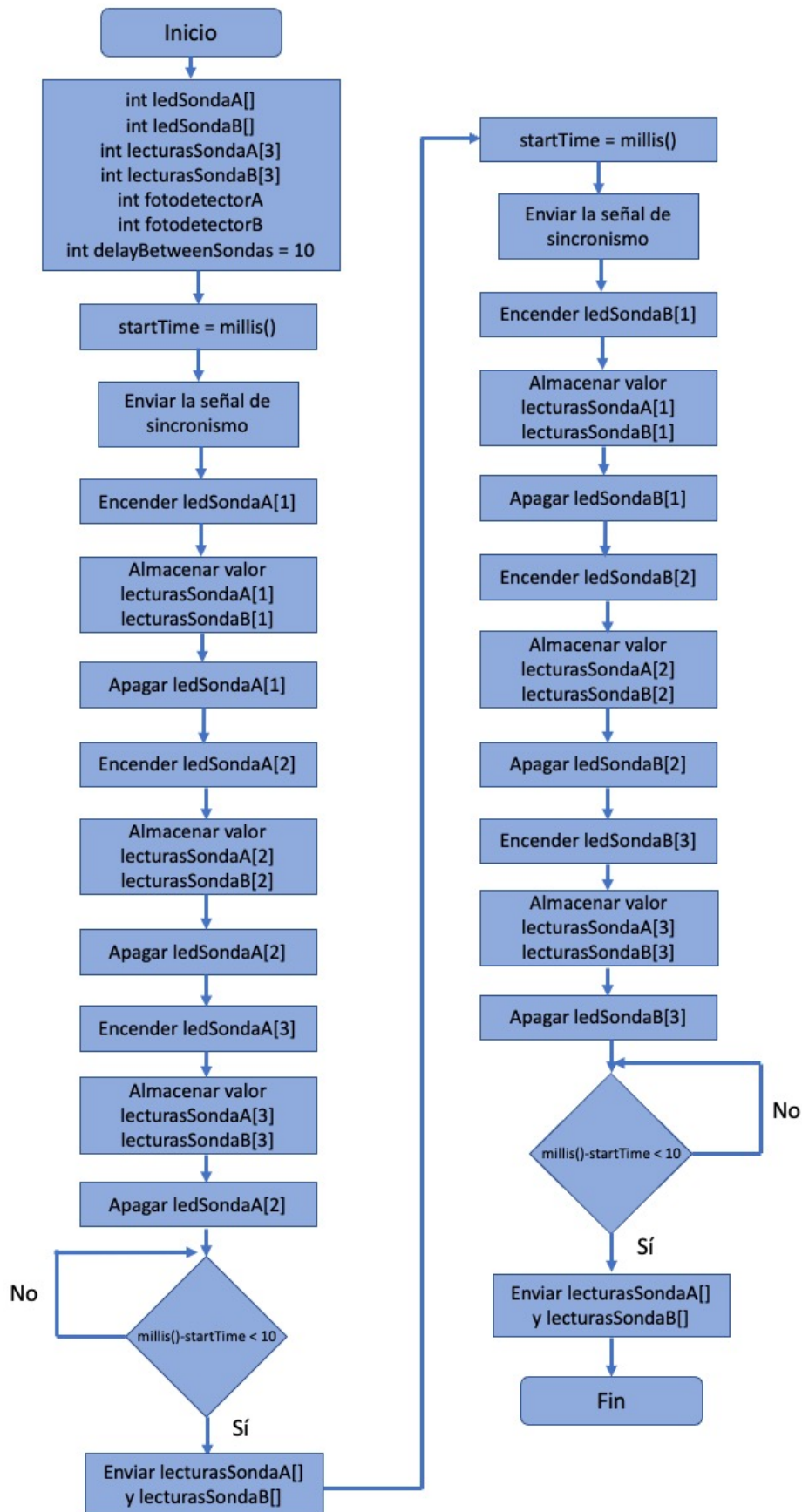


Figura 5.21. Diagrama de funcionamiento del software de control NIRS

5.3 Diseño de la GUI

Por último, es necesario desarrollar una interfaz gráfica que permita al usuario tanto poder manejar el dispositivo como recibir y almacenar los datos de la sesión de trabajo. El uso de esta interfaz nos va a permitir aumentar las funcionalidades del equipo, permitiendo implementar una serie de protocolos de trabajo que facilitarán el uso del dispositivo. Se van a programar tres protocolos de trabajo, que como se explicó anteriormente, están orientados a estudiar el nivel de dolor experimentado por el sujeto.

El diseño de la interfaz se puede dividir en tres secciones bien diferenciadas. La primera sección corresponde con el control básico del equipo, es decir, botones para el inflado y desinflado de las mangas, la segunda, donde se recogen los protocolos de algometría, y, por último, la lectura de datos en tiempo real.

La implementación de los controles básicos es muy sencilla. Se habilitó un campo de texto para introducir la presión y dos botones: uno para inflar y otro para desinflar las mangas, según el modo de trabajo. Consiguiendo así que cada manga o ambas mangas se puedan usar de forma independiente. Al pulsar uno de estos botones, el programa Python va a enviar un comando a través del puerto serie para hacérselo llegar a la PCB de control. Anteriormente se explicó, que una de las tareas del *software* de la placa de control se encargaba de estar siempre leyendo la llegada de posibles comandos para activar los *flags* correspondientes que permitiesen realizar la acción necesaria. En la Tabla 5.6 se recogen todos los comandos que se emplean para manejar el equipo.

El envío de estos comandos es la base del funcionamiento de la GUI. En consecuencia, los protocolos que se van a explicar a continuación emplean la misma metodología cada vez que actúan sobre las mangas del dispositivo. Para la explicación de estos protocolos se va a hacer uso de diferentes diagramas de flujo que permiten plasmar mejor como operan y las acciones que realizan.

Comando	Acción
'a presión'	Llenado de la manga A al nivel <i>presión</i> indicado
'a 0'	Vaciado de la manga A
'b presión'	Llenado de la manga B al nivel <i>presión</i> indicado
'b 0'	Vaciado de la manga B
'p presión'	Llenado de ambas mangas al nivel <i>presión</i> indicado
'p 0'	Vaciado de ambas mangas

Tabla 5.6. Comandos de operación de la GUI

El primero de los protocolos que se va a explicar se denomina “*Algometer Gradual*”. Básicamente este protocolo va aumentando la presión de la manga de forma progresiva, en saltos de 5 mmHg, con el objetivo de que en el momento que el sujeto sienta dolor, pulse el botón de algometría que se implementó en la parte hardware y así quede registrado en el sistema. Una vez el usuario pulsa el botón, la manga de desinfla. La primera vez se realiza únicamente con la manga A, luego se repite el proceso con la manga B y finalmente se realiza con ambas mangas. En la Figura 5.22 se muestra de forma gráfica el funcionamiento del protocolo implementado.

Para el segundo de los protocolos, el “*Algometer Square*”, se cambia la metodología de trabajo. En este caso va a comenzar el proceso de isquemia a una presión previamente establecida y cuando se alcance un tiempo determinado, llamado “tiempo de pitido” se va a hacer sonar un *buzzer* para avisarle al sujeto que debe indicar al equipo médico su estado actual. Una vez finalice el tiempo que dura la isquemia se desinfla la manga. Este proceso se va a repetir tantas veces como número de isquemias se hayan establecido al inicio de la sesión. Los parámetros que se van a poder configurar son: los niveles de presión de cada una de las isquemias, el tiempo de duración de estas, el tiempo de descanso entre isquemias y el instante de tiempo en el que se quiere que

suena el *buzzer* una vez comenzado el proceso de isquemia. Nuevamente, en la Figura 5.23 se muestra de forma gráfica la implementación del código de este protocolo.

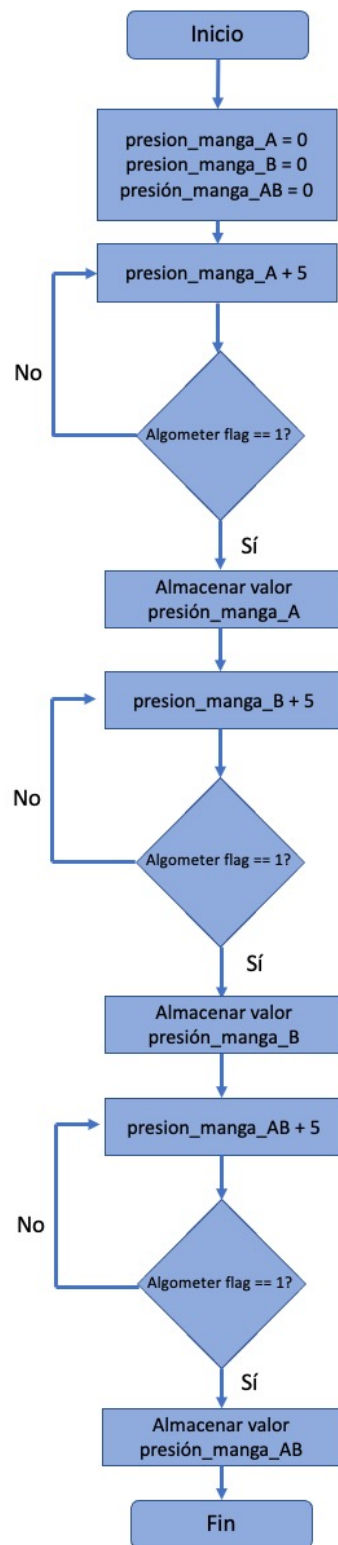


Figura 5.22. Diagrama de flujo del procotolo "Algometer Gradual"

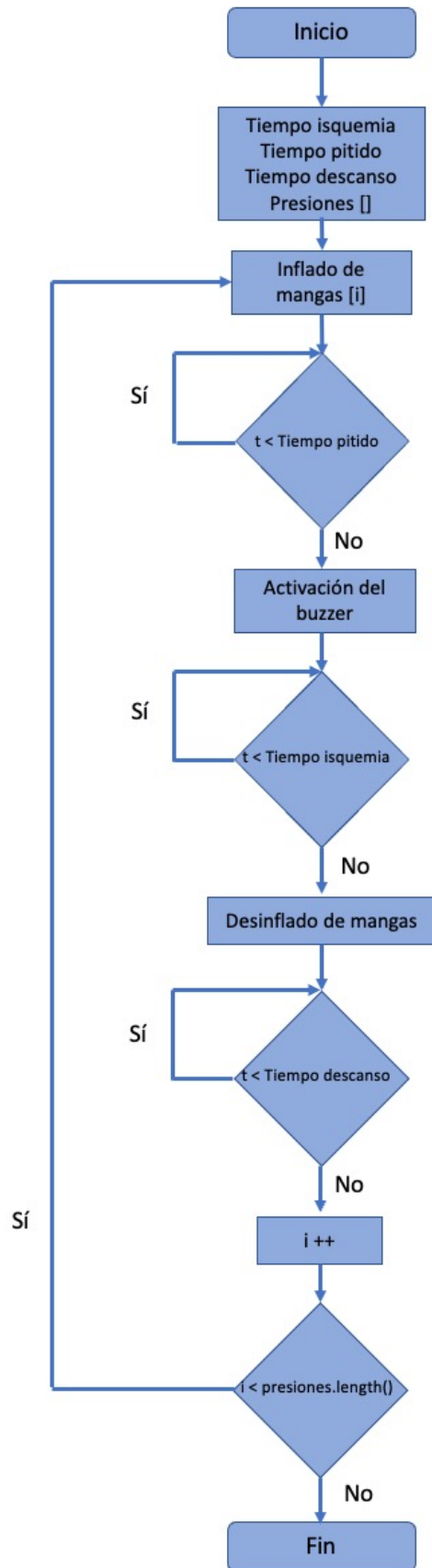


Figura 5.23. Diagrama de flujo del protocolo "Algometer Square"

Finalmente, el "Basic Protocol" permite automatizar la sesión de trabajo, eliminando la necesidad de que el usuario infle y desinfe manualmente las mangas de manera constante. Esta funcionalidad resulta particularmente interesante, ya que ofrece la posibilidad de preconfigurar la sesión, permitiendo al usuario concentrarse en otros aspectos del estudio una vez que este ha comenzado. A continuación, en la Tabla 5.7, se detallan los parámetros que pueden configurarse y en la Figura 5.24 se muestra el diagrama de flujo del protocolo implementado.

Parámetro	Funcionalidad
Tiempo isquemia	Tiempo que va a durar la primera isquemia que se realice
Número isquemia	Establece el número de isquemias consecutivas que se van a realizar
Tiempo de descanso	Establece el tiempo de descanso que se realizará entre la primera y la segunda isquemia
Presión	Nivel de presión que se va a aplicar. La misma presión para todos los procesos de isquemias
Tiempo de isquemia consecutivo	Tiempo que va a durar el resto de las isquemias consecutivas que se realicen
Tiempo de descanso consecutivo	Establece el tiempo de descanso que se realizará entre el resto de las isquemias consecutivas que se realicen

Tabla 5.7. Parámetros a configurar en el protocolo "Basic Protocol"

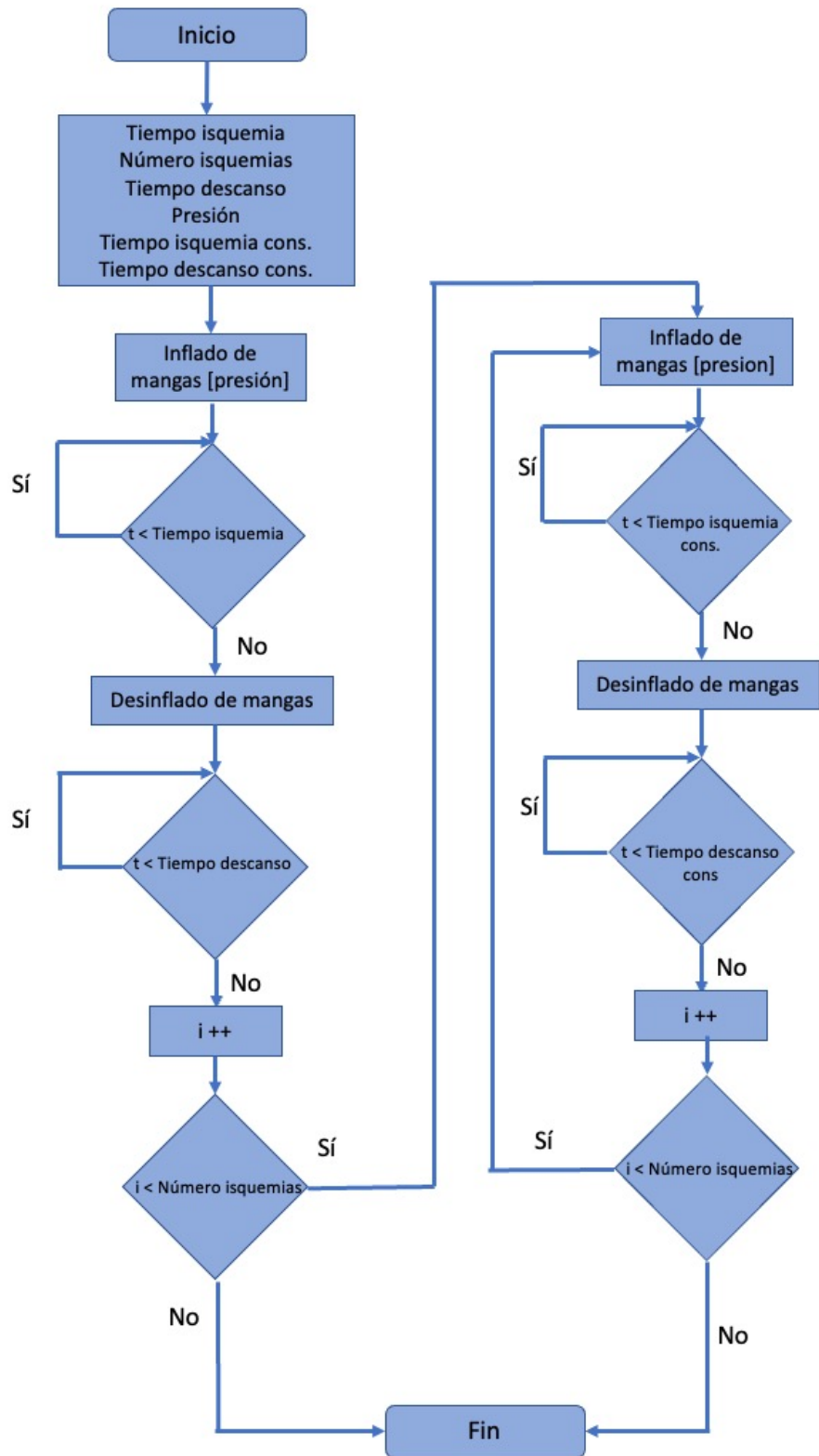


Figura 5.24. Diagrama de flujo del protocolo "Basic Protocol"

Finalmente, en la sección de representación de datos en tiempo real se van a estar presentando, tanto de forma gráfica como numérica, los distintos valores que la PCB de control está constantemente enviando al ordenador. Concretamente se presenta los niveles de presión que se están leyendo en cada una de las mangas, el nivel del tanque de presión y el tiempo que ha durado el proceso de isquemia. Además, al usuario se le da la opción de activar el guardado automático de los datos haciendo que todos los datos recibidos se almacenen en un archivo CSV generado para la sesión. Toda esta recepción de datos se está realizando mediante lecturas no bloqueantes que realiza el programa Python al puerto serie de comunicación con la placa. En el siguiente capítulo se mostrará el resultado de la implementación realizada.

6. Implementación

En esta sección se presenta la implementación realizada del sistema, abarcando desde el diseño de la PCB hasta la fabricación de las sondas NIRS y la interfaz gráfica. Se detallan los procesos y métodos utilizados para desarrollar cada componente, así como los desafíos encontrados y las soluciones adoptadas.

6.1 Implementación del hardware de control

Una vez realizado el diseño de la PCB v0.3 se tiene que llevar a cabo su implementación. Para ello se consideró que la mejor opción era fabricar la placa a través de una empresa que facilitara y agilizase todo el proceso. Tras analizar las diferentes opciones del mercado se optó por trabajar con JLCPCB [49] debido a su bajo coste y al reducido tiempo de fabricación y envío.

Para poder realizar el pedido era necesario subir los archivos *Gerber* de nuestra placa a su página web. Estos son un conjunto de archivos estándar que contiene la información específica de cada una de las capas de la PCB necesaria para llevar a cabo su fabricación exactamente como se diseñó. KiCad permite generar estos archivos fácilmente mediante los siguientes pasos: *File>Fabrications Outputs>Gerbers*.

La ventana que se abre tras realizar los pasos anteriores permite seleccionar todas aquellas capas que se quieren incluir en el archivo y que así puedan ser fabricadas, como se muestra en la Figura 6.1.

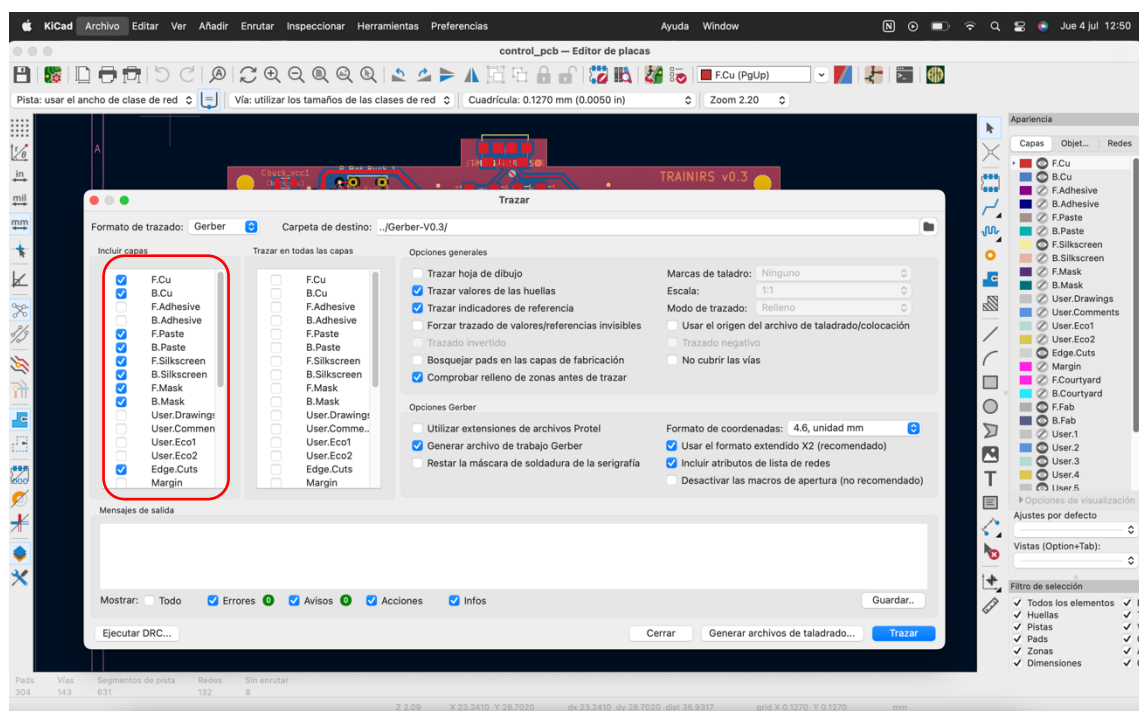


Figura 6.1. Ventana de selección de capas para el archivo gerber

Para concluir la implementación de la placa, fue necesario soldar los componentes restantes. Utilizando una estación de soldadura de punta fina y estaño, se completó el proceso de soldadura de los elementos. De este modo, se dio por finalizada la implementación de la PCB v0.3 que se muestra en la Figura 6.3, encargada de gestionar la parte mecánica del sistema.

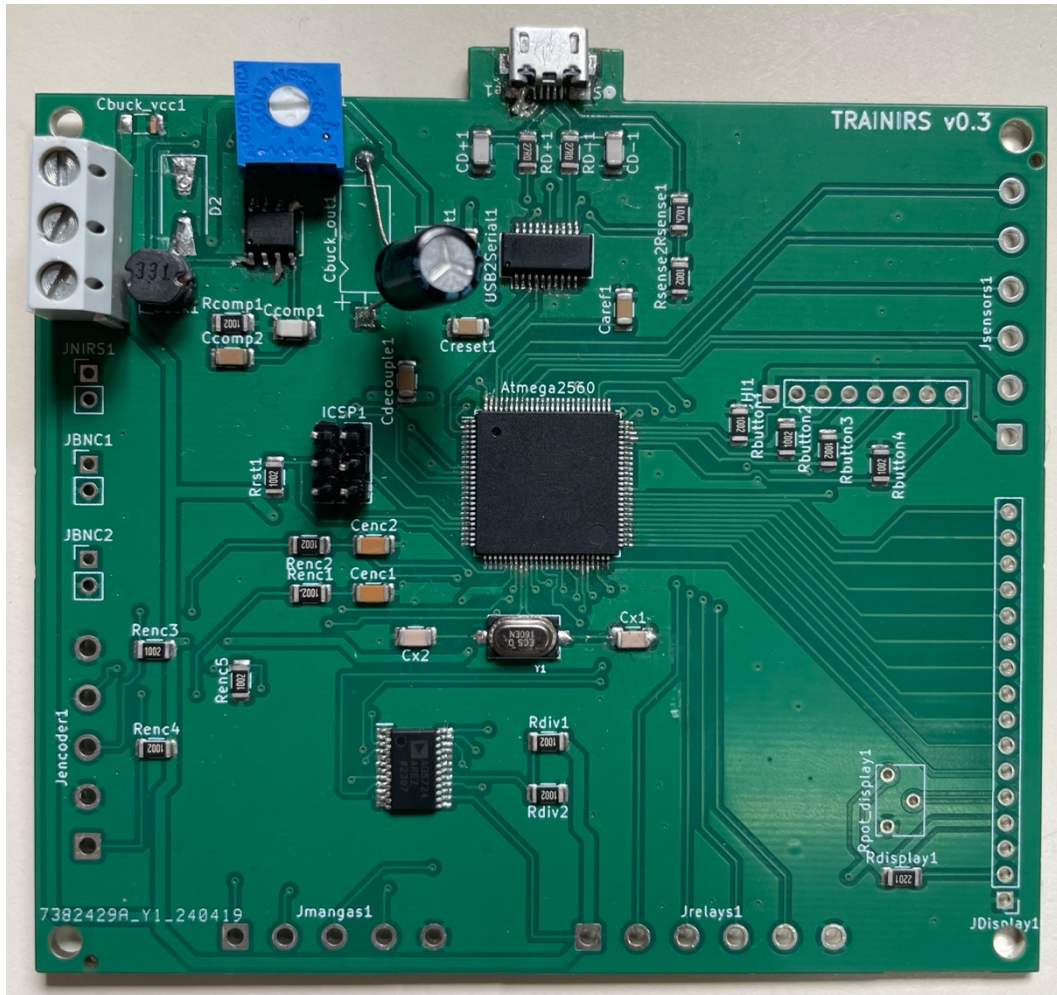


Figura 6.3. PCB v0.3 totalmente implementada

6.2 Implementación del hardware NIRS

Para llevar a cabo la implementación de las sondas de medida primero se tuvo que analizar cuál era la mejor estructura para seguir teniendo en cuenta que debía soportar el movimiento típico de una actividad deportiva. Tras analizar diferentes

opciones se decidió implementar el esquema que se muestra en la Figura 6.4, ya que permite proteger los componentes más sensibles y dotar de cierta robustez al sistema.

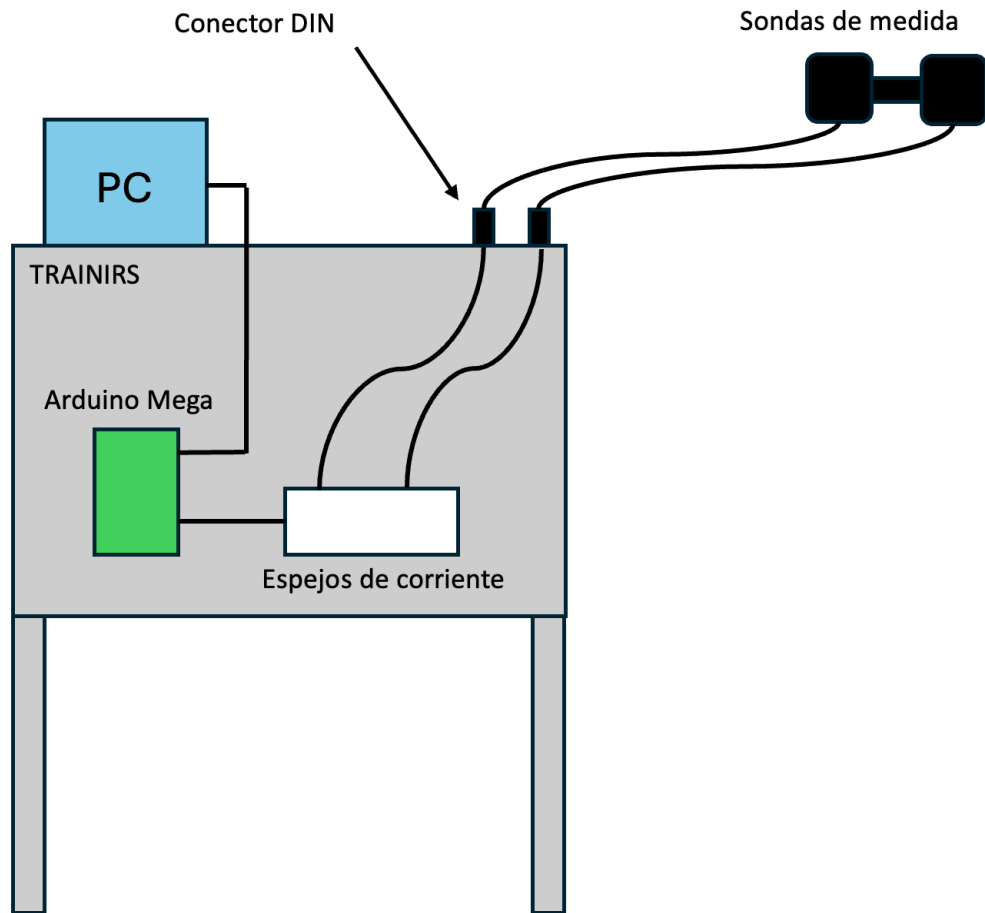


Figura 6.4. Esquema para montaje del sistema NIRS

Se decide comenzar montando las sondas de medida. Para ellos, tras estudiar diferentes propuestas se decidió implementar la que se muestra en el esquema de la Figura 6.5. Como se puede ver, la distancia entre componentes es la mínima disponible, tratando que la sonda resultante sea lo más pequeña posible y no moleste al sujeto a la hora de moverse. Se debe tener en cuenta que, debido a las distancias utilizadas, cada sonda de forma independiente solo puede ser capaz de medir la oxigenación de la piel del sujeto ya que el transmisor y el receptor están muy próximos. Debido a ello, se fabricaron dos sondas y se combinan ambos receptores en la medida de un único LED, como se explicó anteriormente. Así se consigue obtener información tanto de la

oxigenación de la piel como del músculo. Otros autores implementan el transmisor en una sonda y el receptor en otra, pero ese no es nuestro caso.

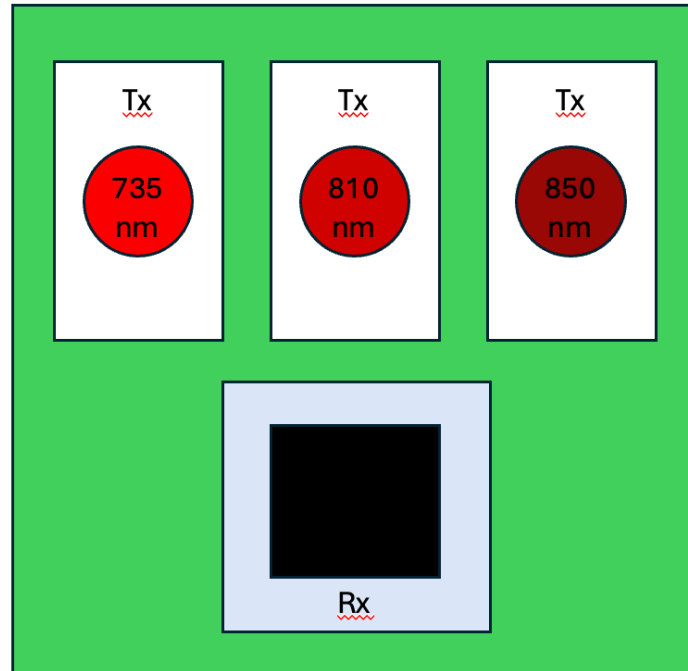


Figura 6.5 Esquema de diseño para la sonda de medida

Con el esquema de montaje claro, se pasó a soldar todos los componentes en una placa perforada obteniendo el resultado que se muestra en la Figura 6.6. Como se puede ver, se soldaron 8 hilos, para posteriormente preparar el conector DIN. En la Tabla 6.1 se recoge la correspondencia entre cada elemento y los hilos.

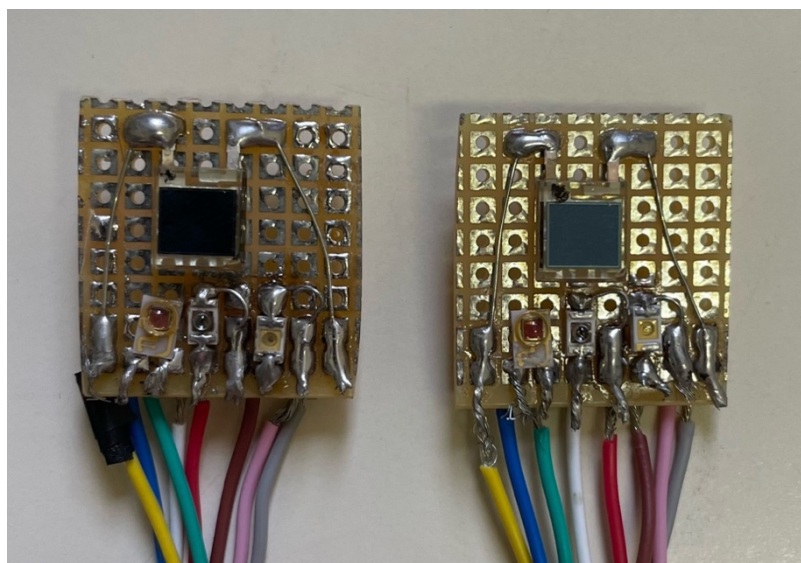


Figura 6.6. Resultado de la implementación de los componentes de las sondas

Color	Componente
ROSA	Cátodo 850
ROJO	Cátodo 810
BLANCO	Ánodo 810
AZUL	Cátodo 730
AMARILLO	Fotorreceptor +
VERDE	Ánodo 730
GRIS	Fotorreceptor -
MARRÓN	Ánodo 850

Tabla 6.1. Correspondencia entre color del cable y componente

Para tratar de dar una mayor protección a la sonda ya que va a estar en constante movimiento se decidió que la mejor opción era fabricar una caja mediante impresión 3D. Los requisitos de la caja eran muy básicos, simplemente que cubriese toda la PCB para protegerla de golpes. El diseño final de la caja implementada se puede ver en la Figura 6.7. Como se puede apreciar, la caja permite cubrir totalmente la sonda y, además, se decidió añadir una separación entre los transmisores y el receptor tratando de impedir que el fotodetector pueda captar radiación directa de los transmisores. De esta forma lo que se consigue es asegurarnos que toda la radiación captada por el receptor es proveniente de la reflexión de los haces de luz con el cuerpo humano. También se le añadió el detalle que se muestra en la Figura 6.8 para asegurar la separación de los cables entre sí evitando cualquier posible contacto indeseado entre ellos que pueda afectar en el funcionamiento de las sondas.

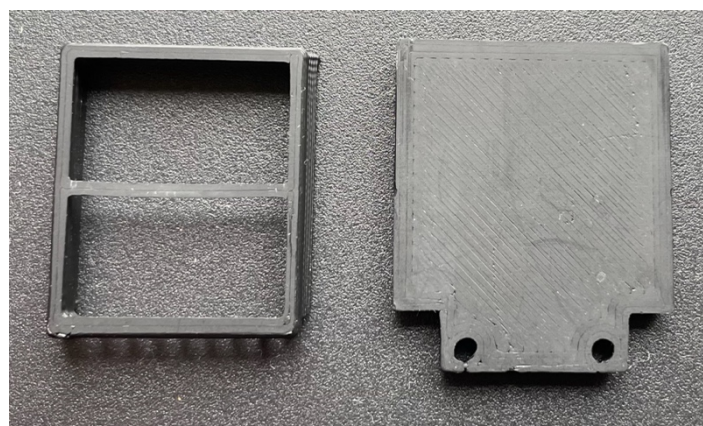


Figura 6.7. Caja de protección para la sonda

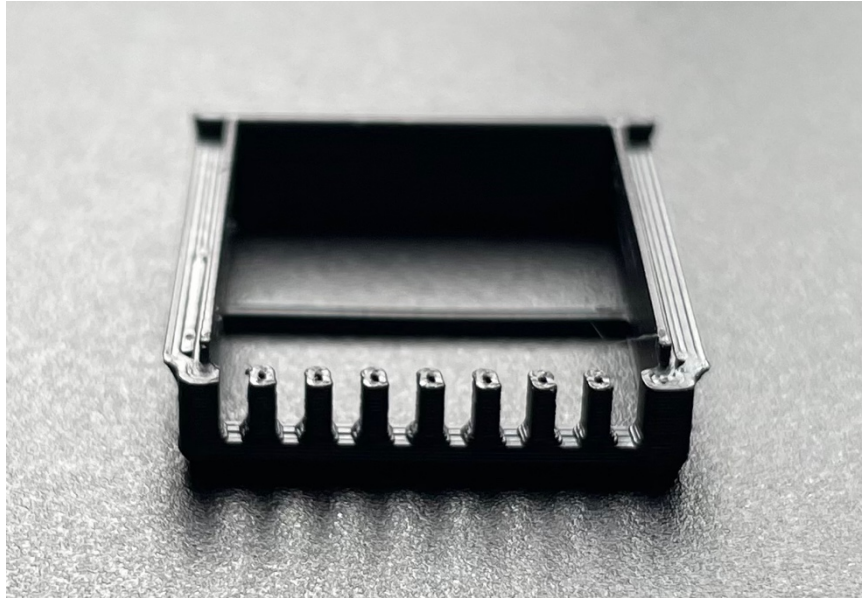


Figura 6.8. Separador de cables

Finalmente, se optó por fabricar una pieza adicional que permitiera la unión de ambas sondas, manteniendo cierta flexibilidad para adaptarse a la curvatura de un brazo o una pierna. El resultado final de este diseño se muestra en la Figura 6.9. La pieza fue impresa de manera que la distancia entre el transmisor y el receptor fuera de 2,5 cm, ya que esta distancia es ideal para medir la oxigenación en la pierna. Se eligió este diseño porque permite, en futuras pruebas, modificar fácilmente la distancia mediante la fabricación de una nueva pieza, posibilitando así la adaptación a diferentes partes del cuerpo. En la Figura 6.10 se puede ver ambas cajas ya montadas y unidas por la pieza comentada.

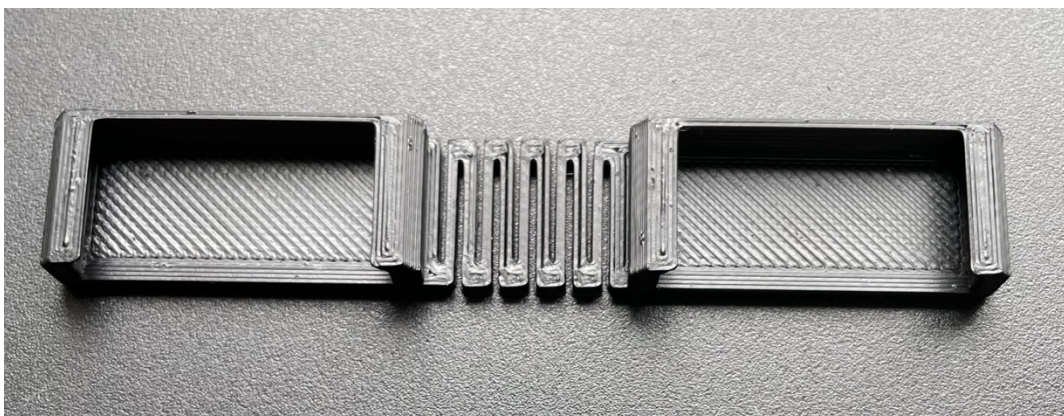


Figura 6.9. Pieza de unión para ambas cajas

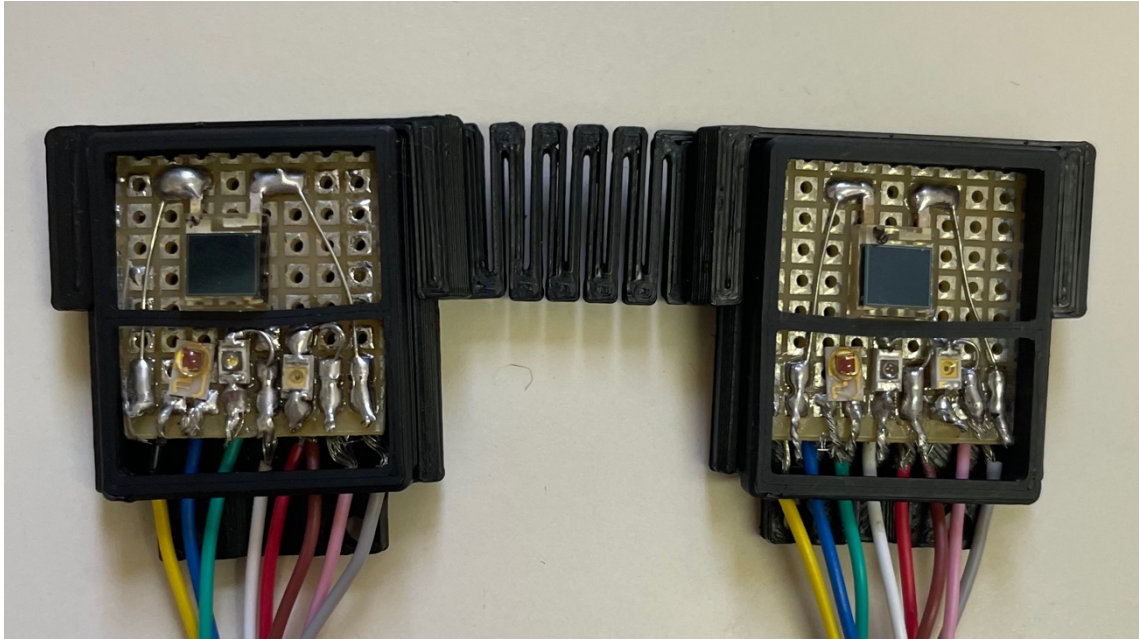


Figura 6.10. Montaje final del transmisor y el receptor

Una vez terminado el montaje de la sonda, se pasó a preparar el conector DIN que permitiese conectarlas con el equipo principal. Para ello se utilizó el conector que se muestra en la Figura 6.11. Se tuvo que desmontar la carcasa del conector y estañar la punta de todos los cables para así mejorar la soldadura y prevenir el deshilachado de los cables. Luego, se fue soldando cada hilo con el pin del conector y verificando que no se producían cortocircuitos. Finalmente, se montó nuevamente la carcasa del conector obteniendo el resultado que se muestra en la Figura 6.12. Por otro lado, en la Figura 6.13 se muestra el resultado final del montaje de la sonda.

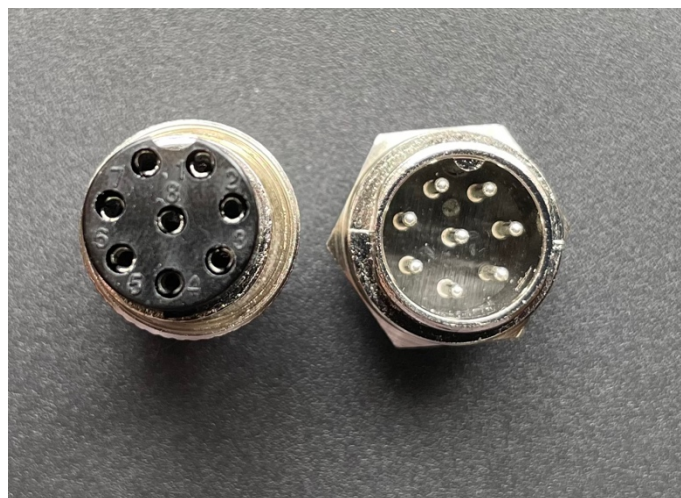


Figura 6.11. Conector DIN de 8 hilos utilizado



Figura 6.12. Resultado final del montaje del conector hembra



Figura 6.13. Resultado final del montaje de la sonda

A continuación, se pasa a montar el otro extremo del conector, es decir, el conector DIN macho. Esta parte es mucho más sencilla ya que no es necesario desmontar el conector. Únicamente se tuvo que soldar cada cable al pin del conector garantizando que no se produzcan cortocircuitos como se hizo anteriormente. En resultado final obtenido se muestra en la Figura 6.14.

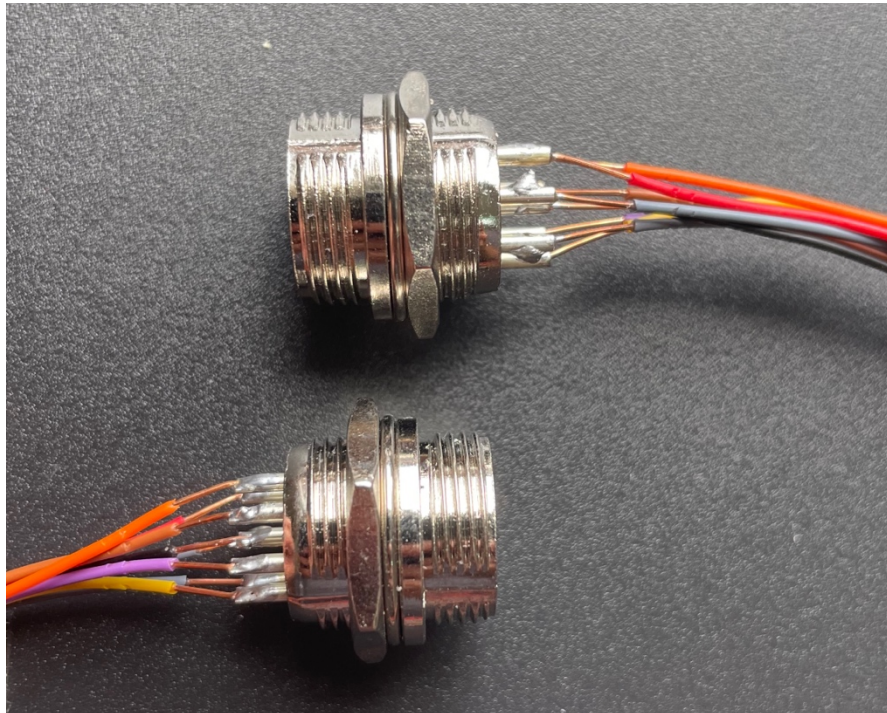


Figura 6.14. Conector DIN macho implementado

Finalmente, para completar el montaje del hardware NIRS, es necesario implementar los espejos de corriente previamente estudiados. Se decidió utilizar una *protoboard* debido a que permite la realización del conexionado requerido de manera sencilla y rápida usando cables y componentes tipos THT (*Thru-Hole Technology*) que son muy fáciles de conseguir. En la Figura 6.15 se muestra el esquema de conexionado que se debe implementar para cada una de las sondas. Este esquemático se tendría que replicar para cada sonda que se quiera implementar. Por último, en la Figura 6.16 se presenta el resultado final obtenido del montaje de los espejos.

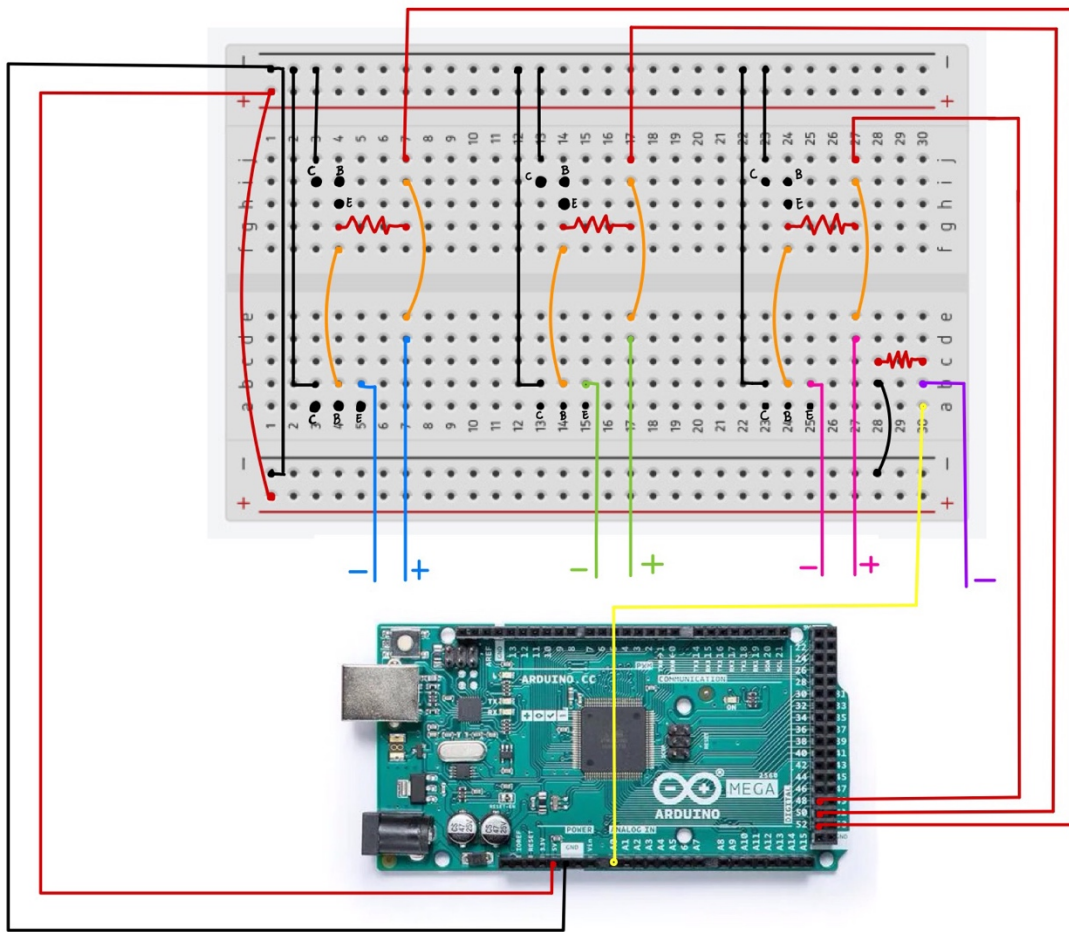


Figura 6.15. Esquema de conexionado

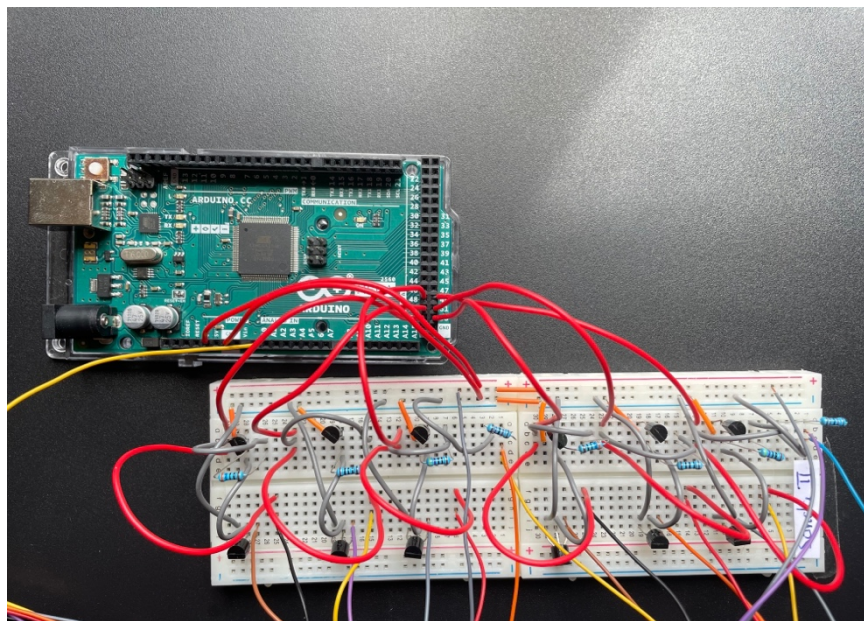


Figura 6.16. Montaje final de los espejos de corriente

En la Figura 6.17 se muestra el sistema completo implementado. Se puede ver como se conectan perfectamente las sondas, que saldrían de la carcasa del dispositivo, con los espejos, que se colocarán dentro del dispositivo. Con este montaje se asegura que cualquier tensión que pudiera generarse no afecte al sistema, evitando así la desconexión de algún cable de la *protoboard* que pueda comprometer el desarrollo de la prueba.

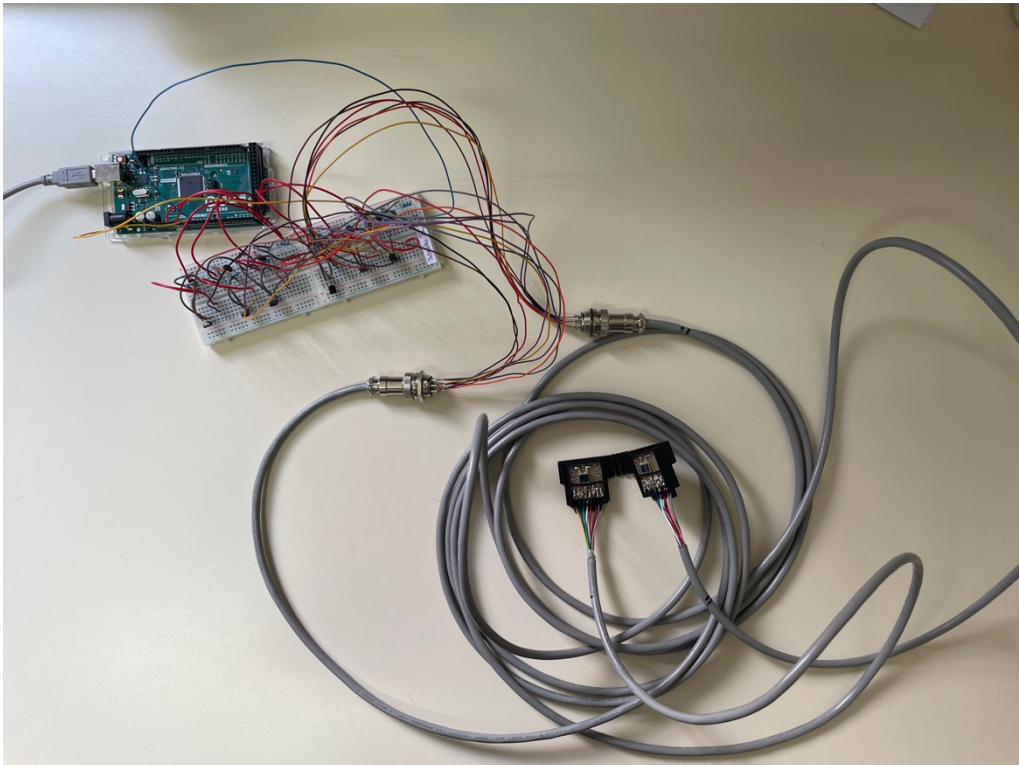


Figura 6.17. Sistema NIRS totalmente implementado

6.3 Implementación de la GUI

En este último apartado se va a mostrar la implementación de la interfaz gráfica. Para ello se van a añadir capturas de pantalla de esta en funcionamiento, así se pueden apreciar las diferentes partes desarrolladas y posteriormente, en el próximo capítulo, se realizará la validación.

En primer lugar, en la Figura 6.18, se puede ver el diseño básico de la interfaz. La sección A corresponde con los controles básicos del equipo, es decir, inflado y desinflado de cada manga. La sección B corresponde con la sección de protocolos desarrollados

que se muestra con más detalle a continuación, y, por último, la sección C que muestra en tiempo real los datos de la prueba y permite activar al guardado automático de los mismos.



Figura 6.18. Secciones de la GUI básica

Como se comentó anteriormente, la sección B se corresponde con la sección que permite el seleccionado del protocolo a implementar. Para ello se diseñó un menú desplegable, el cual, una vez elegido el protocolo correspondiente, muestra todos los parámetros configurables como se puede ver en la Figura 6.19 y Figura 6.20. Además, cuenta con un botón “ON” para iniciar el protocolo y un botón “OFF” para finalizarlo antes de tiempo ante cualquier emergencia o contratiempo que pueda surgir. En el caso del protocolo “*Algometer Gradual*”, no es necesario introducir ningún parámetro. Solo se muestran los botones de inicio y de *stop*, como se observa en la Figura 6.21.

En el caso de que el usuario no configure ningún parámetro e inicie el protocolo, se han establecido parámetros por defecto, que corresponden a los habituales de trabajo. Esto permite al usuario agilizar el manejo de la interfaz siempre que se disponga a realizar los protocolos clásicos diseñados.

Select Protocol: Algoter Square ▾

Tiempo de isquemia Tiempo de pitido

Tiempo de descanso d Presiones

ON OFF

Figura 6.19. Implementación del protocolo Algoter Square

Select Protocol: Basic Protocol ▾

Tiempo de isquemia Tiempo de descanso

Numero de isquemias T isquemia cons.

T descanso cons. Presión

ON OFF

Figura 6.20. Implementación del protocolo básico de funcionamiento

Select Protocol: Algoter Gradual ▾

ON OFF

Figura 6.21. Implementación del protocolo Algoter Gradual

7. Validación

En este capítulo, se procederá a la validación de los componentes desarrollados en los capítulos anteriores, garantizando su correcto funcionamiento y cohesión dentro del sistema. La validación incluye una serie de pruebas diseñadas para evaluar la eficacia y precisión de cada parte diseñada por separado. Los resultados obtenidos se presentarán y analizarán para asegurar que cumplen con los requisitos especificados y los objetivos planteados al inicio del proyecto.

7.1 Validación del hardware de control

Para la validación de la PCB v0.3 que ha sido fabricada en este TFM se van a llevar cabo una serie de pruebas que permitan comprobar cada una de las soluciones diseñadas. Se ha establecido que para considerar como correcta la implementación de la placa se deben cumplir los 3 requisitos de diseño especificados anteriormente, es decir, que las mangas de presión puedan alcanzar una tensión mayor a 5 V al llegar al nivel de presión máximo, que se pueda implementar una interfaz manual totalmente operativa y, por último, que el botón de algometría funcione correctamente.

Lo primero que se debe realizar es tratar de programar nuestro microprocesador y poder leer lo que nos devuelve, de esta forma se puede saber que toda la placa alimentada por 5 V funciona correctamente. Para poder subir el código diseñado es necesario alimentar nuestra placa con una fuente de alimentación externa de 5 V y programar el dispositivo a través de la entrada ICSP (*In Chip Serial Programmer*) siguiendo el esquema de la Figura 7.1. Es necesario hacer uso de una placa Arduino Mega adicional que permita ejecutar el fichero programador “*ArduinoISP*” [50] que permite subir nuestro código a la PCB v0.3. El montaje que se tuvo que llevar a cabo se puede ver en la Figura 7.2.



Figura 7.1. Esquema de conexión de la entrada ICSP

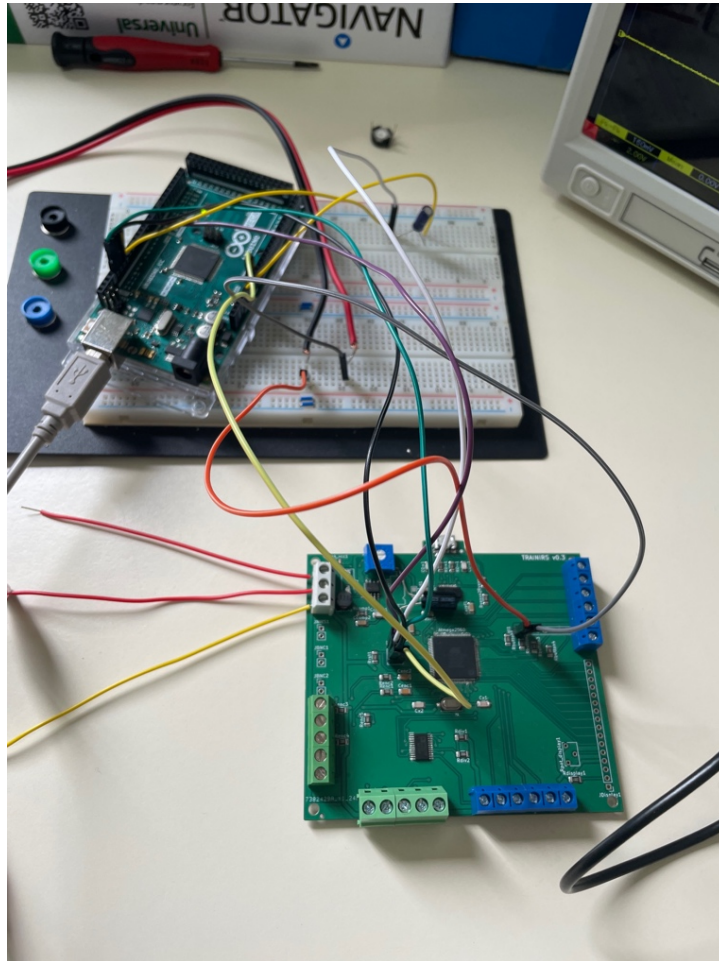


Figura 7.2. Montaje para la programación del microprocesador

Hecho esto se pudo comprobar que todo funcionaba correctamente ya que se podían leer todos los valores que devolvía la placa a través del puerto serie. Llegados a este punto, ya se puede comenzar con las pruebas para validar la placa implementada.

Se va a comenzar estudiando la tensión de salida que se aplica a cada una de las mangas. Para ello, en primer lugar, se debe comprobar que el diseño de la red de alimentación este convirtiendo correctamente de 24 V a 7 V. Por lo tanto, se conectó la placa a una fuente de 24 V como se puede ver en la Figura 7.3. Una vez la placa estaba alimentada, con ayuda de un osciloscopio se midió la tensión que se obtiene a la salida del conversor. Como se puede ver en Figura 7.4 donde se muestra la medida hecha con el osciloscopio, se obtienen los 7 V necesarios. De esta forma se puede concluir que esta primera comprobación es satisfactoria.

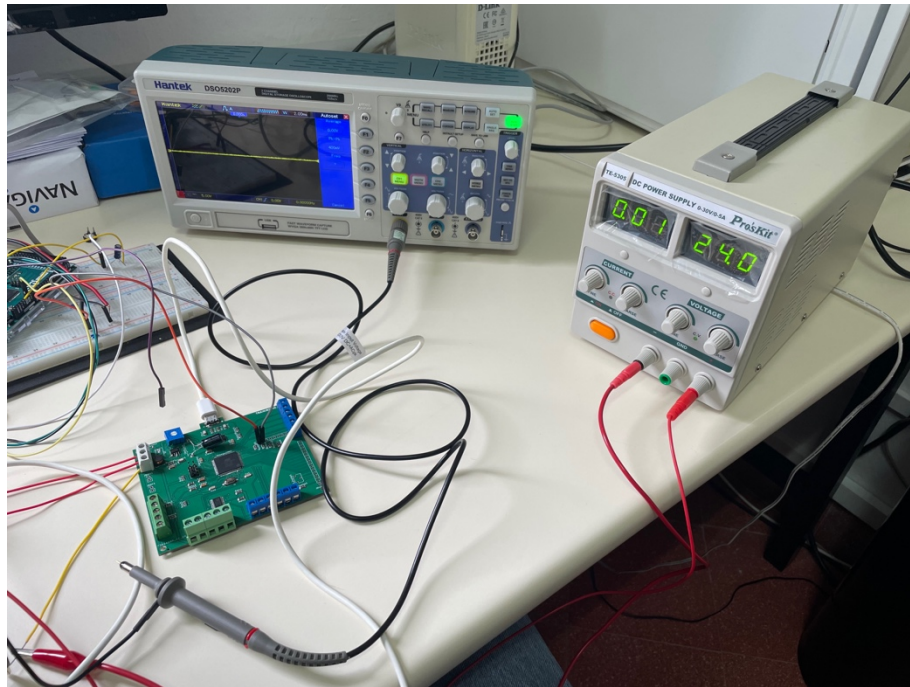


Figura 7.3. Conexión de la fuente de alimentación de 24 V



Figura 7.4. Medida de 7 V a la salida de la red de conversión

Una vez hecho, se va a enviar la orden de inflado de mangas con el ordenador de control por el puerto serie para ver que tensión se alcanza en la salida de las mangas. Para ello se va a enviar el siguiente comando “a 375”, indicando que se infle la manga A

al nivel máximo de presión 375 mmHg y con la ayuda de un voltímetro se mide la tensión en el pin de salida del DAC correspondiente a la manga A. Tras realizar esta prueba se detectó un error en el enrutado de las conexiones del DAC, debido a que dos pines fueron conectados de manera incorrecta. Para el diseño de la siguiente versión de la placa, PCB v0.4, se debe tener en cuenta que el pin *SYNC*, que corresponde con el *Chip Select*, debe ser conectado a un pin digital del microprocesador. Por otro lado, el pin *CLR*, que fue conectado directamente a tierra debe ser conectado a otro pin digital ya que se encargará de restablecer las salidas del DAC, es decir, volver a un estado conocido tras un evento de reinicio.

Siguiendo la metodología explicada anteriormente, este proceso de validación ha permitido detectar un fallo en el enrutado de la placa que será arreglado para la siguiente versión. A continuación, el siguiente paso será comprobar que el botón de algometría implementado permite cumplir la función que se planteó en un primer momento. Para llevar a cabo esta prueba, se fabricó un pequeño botón que se muestra en la Figura 7.5. Este botón se conectó a los pines correspondientes de la placa y se comprobó que al pulsarlo el microprocesador era capaz de detectarlo y modificar el *flag* necesario. Para mostrar su funcionamiento se capturó el puerto serie con toda la información que devuelve nuestra placa. Teniendo en cuenta la trama que devuelve la placa, analizada en el apartado 5.1.2.3. Implementación de funciones, al pulsar el botón se tiene que modificar el *flag* denominado “*algometer_flag*” que corresponde con el noveno valor.

En la Figura 7.6 se puede ver perfectamente como al pulsar el botón el *flag* pasa de 1 a 0, debido a la forma en la que se diseñó el programa, y cuando se deja de pulsar el botón se vuelve a cambiar el valor a 1. Esto lo que nos permite es poder detectar cuando la persona ha pulsado el botón y actuar en consecuencia.

Por último, se va a comprobar el funcionamiento de la interfaz manual. Para ello, se debe conectar a la bandeja del dispositivo que fue fabricada por el equipo encargado del diseño de la parte mecánica. La función principal de esta bandeja es sujetar los botones de inflado y desinflado, el *encoder* y la pantalla LCD. Una vez conectada la placa, se puede apreciar a simple vista como la pantalla LCD está bien implementada ya que

su diseño se ajusta al resultado obtenido como se muestra en la Figura 7.7. Además, tras manipular el *encoder* se puede ver como permite ajustar la presión objetivo que se quiere alcanzar como se muestra en la Figura 7.8.

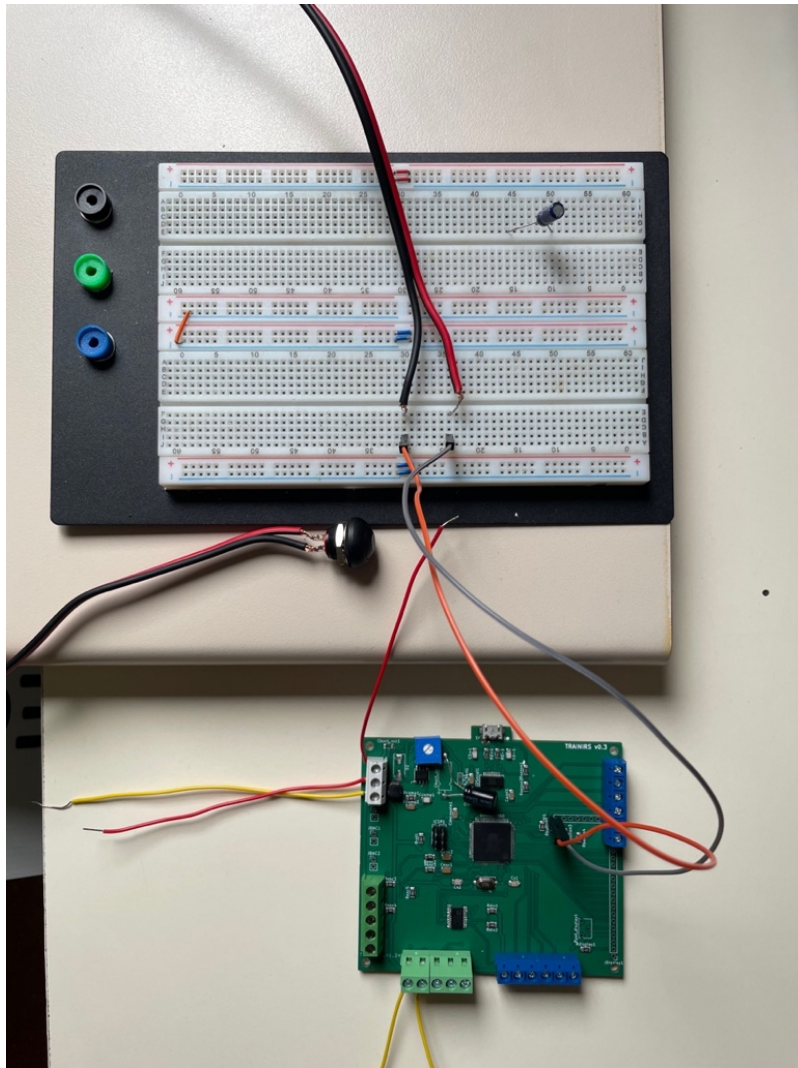


Figura 7.5. Implementación del botón de algometría

```

17:44:44.834 -> 53223,317,0,319,0,318,1,0,1,0,0
17:44:44.927 -> 53323,317,0,319,0,318,1,0,1,0,0
17:44:45.052 -> 53424,317,0,319,0,318,1,0,1,0,0
17:44:45.132 -> 53521,314,0,316,0,315,1,0,0,0,0
17:44:45.213 -> 53623,315,0,318,0,316,1,0,0,0,0
17:44:45.349 -> 53722,313,0,316,0,314,1,0,0,0,0
17:44:45.442 -> 53823,314,0,317,0,315,1,0,0,0,0
17:44:45.535 -> 53921,311,0,315,0,313,1,0,0,0,0
17:44:45.613 -> 54024,313,0,317,0,314,1,0,0,0,0
17:44:45.751 -> 54122,311,0,315,0,313,1,0,0,0,0
17:44:45.829 -> 54223,313,0,316,0,314,1,0,0,0,0
17:44:45.923 -> 54322,310,0,314,0,312,1,0,0,0,0
17:44:46.017 -> 54421,307,0,312,0,309,1,0,0,0,0
17:44:46.157 -> 54522,309,0,314,0,311,1,0,0,0,0
17:44:46.235 -> 54624,310,0,315,0,312,1,0,0,0,0
17:44:46.328 -> 54722,308,0,313,0,310,1,0,0,0,0
17:44:46.413 -> 54823,310,0,315,0,312,1,0,0,0,0
17:44:46.547 -> 54922,307,0,313,0,310,1,0,0,0,0
17:44:46.640 -> 55020,305,0,310,0,307,1,0,0,0,0
17:44:46.734 -> 55121,306,0,312,0,309,1,0,0,0,0
17:44:46.814 -> 55224,309,0,313,0,311,1,0,1,0,0
17:44:46.952 -> 55322,307,0,312,0,309,1,0,1,0,0
17:44:47.044 -> 55424,310,0,314,0,312,1,0,1,0,0
17:44:47.138 -> 55523,309,0,313,0,311,1,0,1,0,0
17:44:47.232 -> 55621,307,0,311,0,309,1,0,1,0,0
17:44:47.356 -> 55723,310,0,314,0,312,1,0,1,0,0
17:44:47.450 -> 55821,308,0,312,0,310,1,0,1,0,0
17:44:47.546 -> 55923,311,0,315,0,313,1,0,1,0,0

```

Figura 7.6. Modificación del flag al pulsar el botón

Una vez seleccionada la presión, se pulsó el botón de inflado y desinflado para comprobar que también funcionaban correctamente. Para mostrar el resultado se hizo uso de la lectura en tiempo real que permite la interfaz gráfica, donde se puede ver como se llevó a cabo el inflado y posteriormente el desinflado. Se puede ver en la Figura 7.9 la línea azul correspondiente a la manga derecha y la línea roja a la manga izquierda.

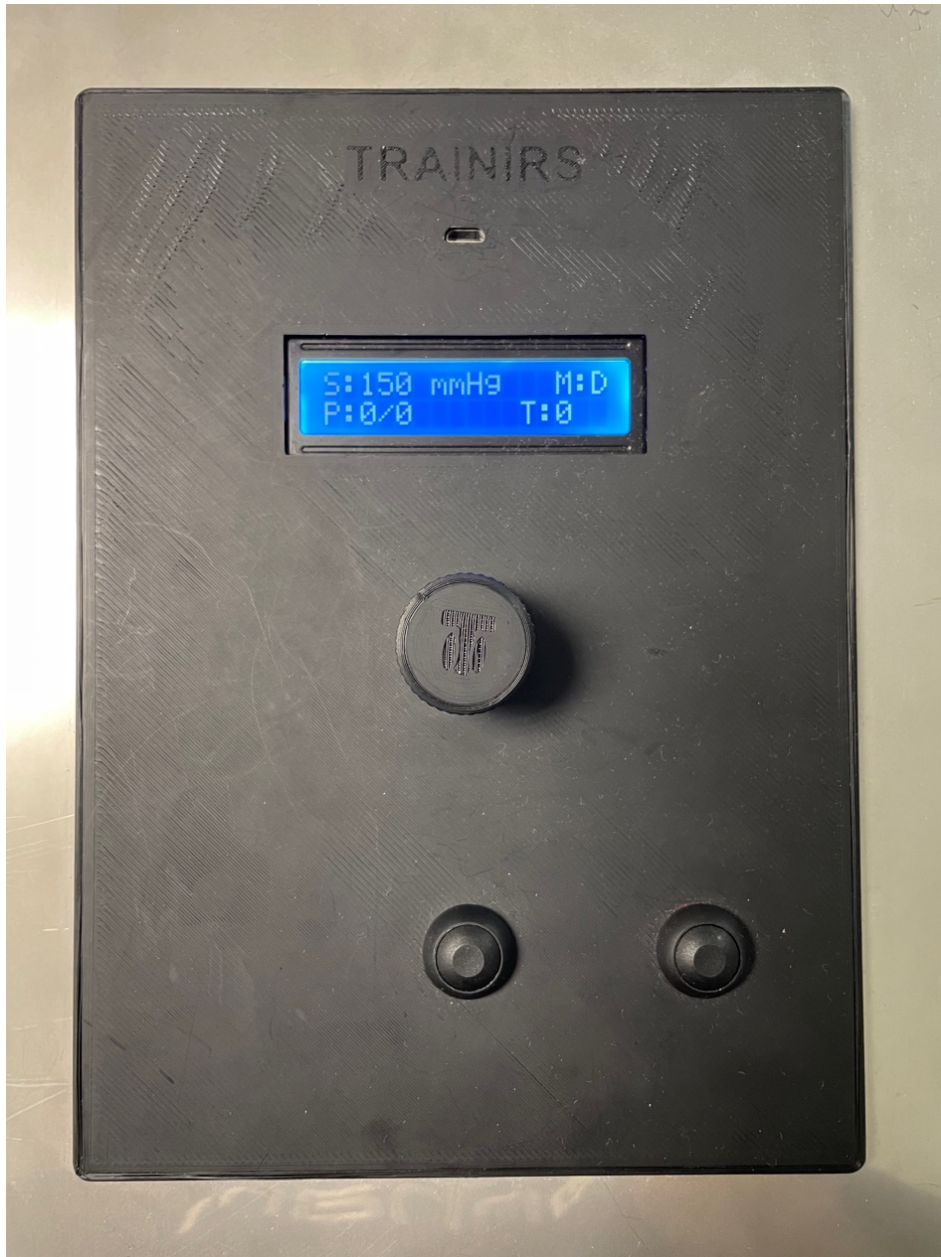


Figura 7.7. Interfaz manual completamente implementada



Figura 7.8. LCD correctamente programada

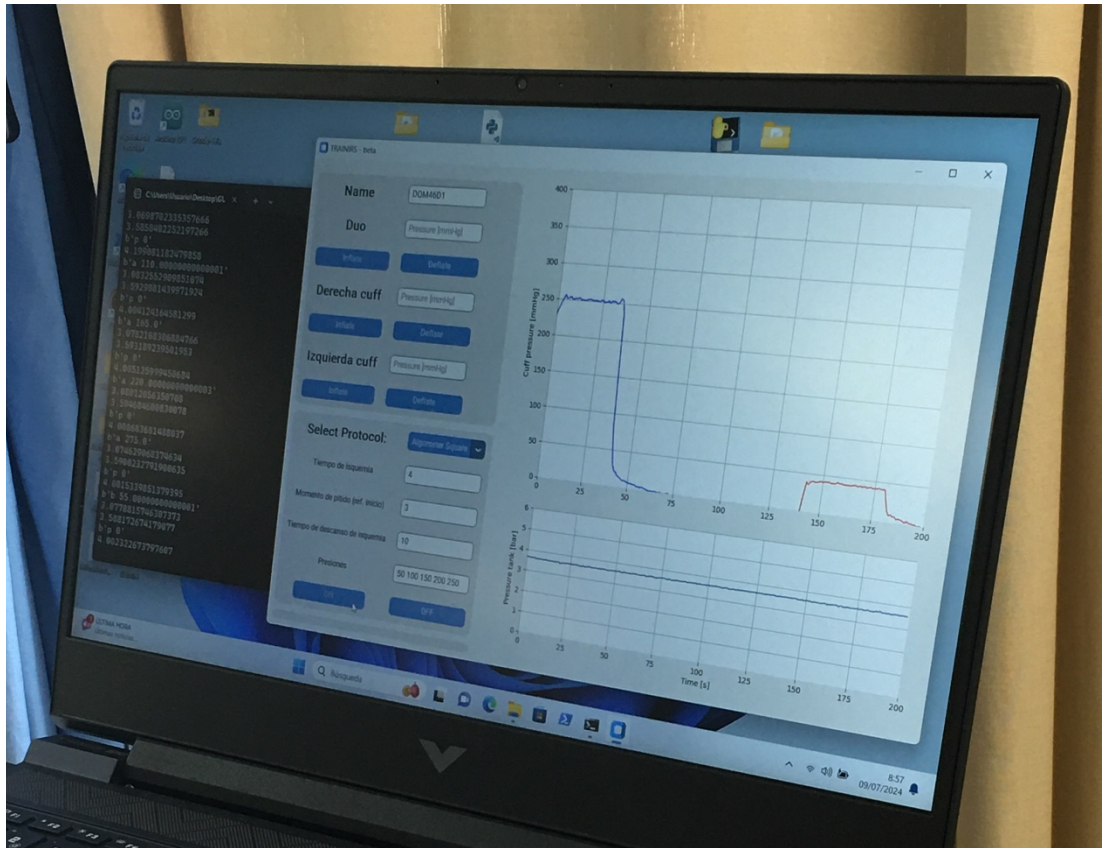


Figura 7.9. Visualización del inflado de cada manga

7.2 Validación del hardware NIRS

Para la validación del hardware NIRS desarrollado se va a simular una situación real de uso, permitiéndonos así comprobar que el sistema es capaz de obtener datos soportando las condiciones de actividad física a los que será sometido. Para ello, se le colocará el dispositivo a un sujeto que tendrá que realizar una actividad física, es este caso, bicicleta estática.

Si el dispositivo es capaz de capturar datos correctamente y funcionar sin ningún tipo de problema durante la realización de una prueba deportiva, se considerará validado su funcionamiento. En futuras etapas, será necesario realizar un procesado de los datos obtenidos para extraer información detallada sobre la oxigenación del músculo. Sin embargo, este es un proyecto que se encuentra aún en desarrollo y el objetivo de este TFM se centró únicamente en la capacidad del dispositivo para capturar datos.

Para llevar a cabo la prueba, en primer lugar, se debe realizar el montaje entero del sistema, el cual se muestra en la Figura 7.10.

Sin embargo, antes de poder utilizar las sondas con un sujeto es necesario comprobar la región de trabajo de cada LED para así garantizar que no se producirán quemaduras en la piel. Para ello, con el equipo en funcionamiento, se hizo uso de un osciloscopio para medir la tensión en cada resistencia de referencia de cada espejo. Conociendo la tensión y el valor de la resistencia, mediante la ley Ohm, se puede conocer la corriente que circula por una de las ramas del espejo de corriente que debe ser la misma que en la otra. Si la corriente obtenida es menor o igual a la calculada de forma teórica se considerará que el sistema cumple con la normativa de seguridad. A continuación, en la Tabla 7.1 y Tabla 7.2 se recoge la medida obtenida para cada una de las sondas.

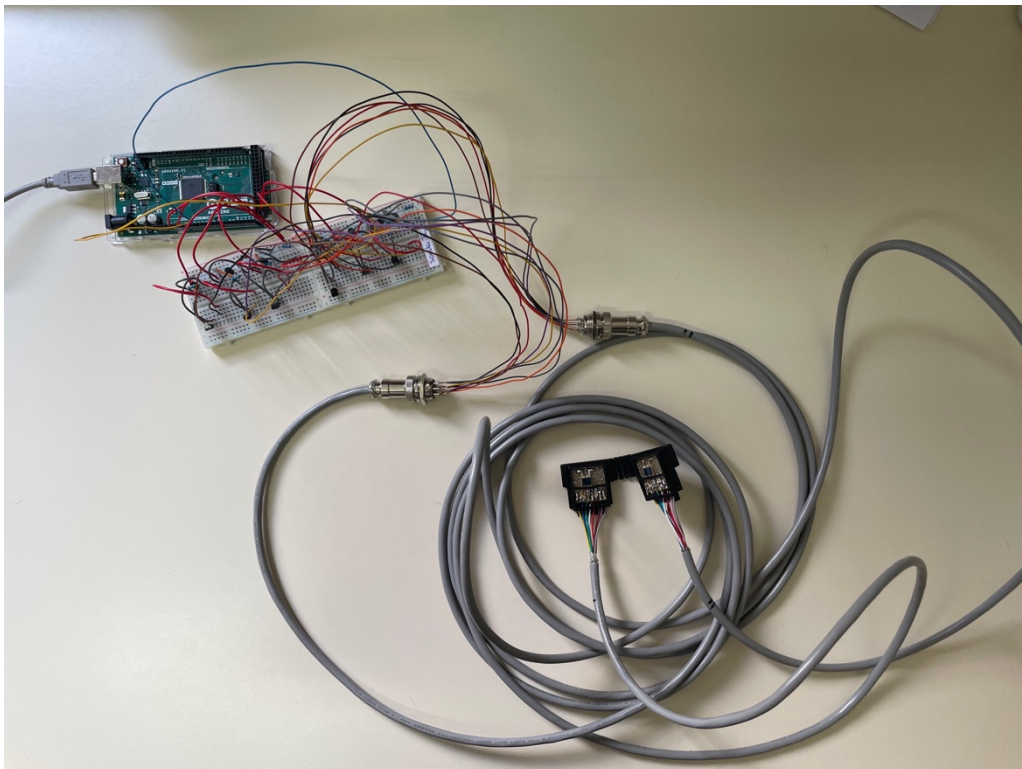


Figura 7.10. Montaje del sistema NIRS completo

Sonda A	735 nm	810 nm	850 nm
Resistencia (Ω)	471,9	100	361
Tensión (V)	3,96	2,88	3,92
Corriente (mA)	8,39	28,8	10,85
Teórica (mA)	9,12	50	15

Tabla 7.1. Valores medidos para la sonda A

Sonda B	735 nm	810 nm	850 nm
Resistencia (Ω)	471,3	99,6	361,1
Tensión (V)	3,96	2,88	3,84
Corriente (mA)	8,40	28,92	10,63
Teórica (mA)	9,12	50	15

Tabla 7.2. Valores medidos para la sonda B

Se puede comprobar que en ningún caso se está superando la corriente teórica que se fijó, con lo cual se está trabajando dentro de la región esperada y se puede usar con sujetos de forma segura. Se debe tener en cuenta que se usaron valores de resistencia comerciales que en algunos casos eran mucho mayores que los calculados de forma teórica, pudiendo eso influenciar la diferencia entre corriente obtenida y la calculada.

Una vez comprobada la seguridad del equipo, el siguiente paso sería la colocación de las sondas en la parte del músculo que se quiere medir, en nuestro caso la parte exterior del cuádriceps. Es muy recomendable rasurar la zona donde se va a colocar el dispositivo ya que la presencia de bello puede afectar a la captura de los datos.

Por otro lado, tras analizar diferentes métodos se concluyó que la manera más práctica de colocación de las sondas era mediante una cinta de doble cara transparente ya que permite la fijación del dispositivo y no afecta a la toma de medida del equipo. Para esta prueba resultó ser más que eficiente, sin embargo, para otras pruebas que requieran más movimiento como la carrera continua puede que sea necesario añadir algún método más de fijación, por ejemplo, una venda que permita agarrar mejor la sonda al músculo.

Una vez fijada la posición de las sondas, como se muestra en la Figura 7.11 y Figura 7.12, se inició la prueba de medición. El sujeto comenzó a pedalear y se inició el equipo para la lectura de datos. Para facilitar la lectura de los datos se implementó un sencillo programa en Python que únicamente se encargaba de leer el puerto serie y almacenar todos los datos obtenidos en un archivo CSV. En la Figura 7.13 se muestra un extracto del archivo generado para poder analizar los datos obtenidos.

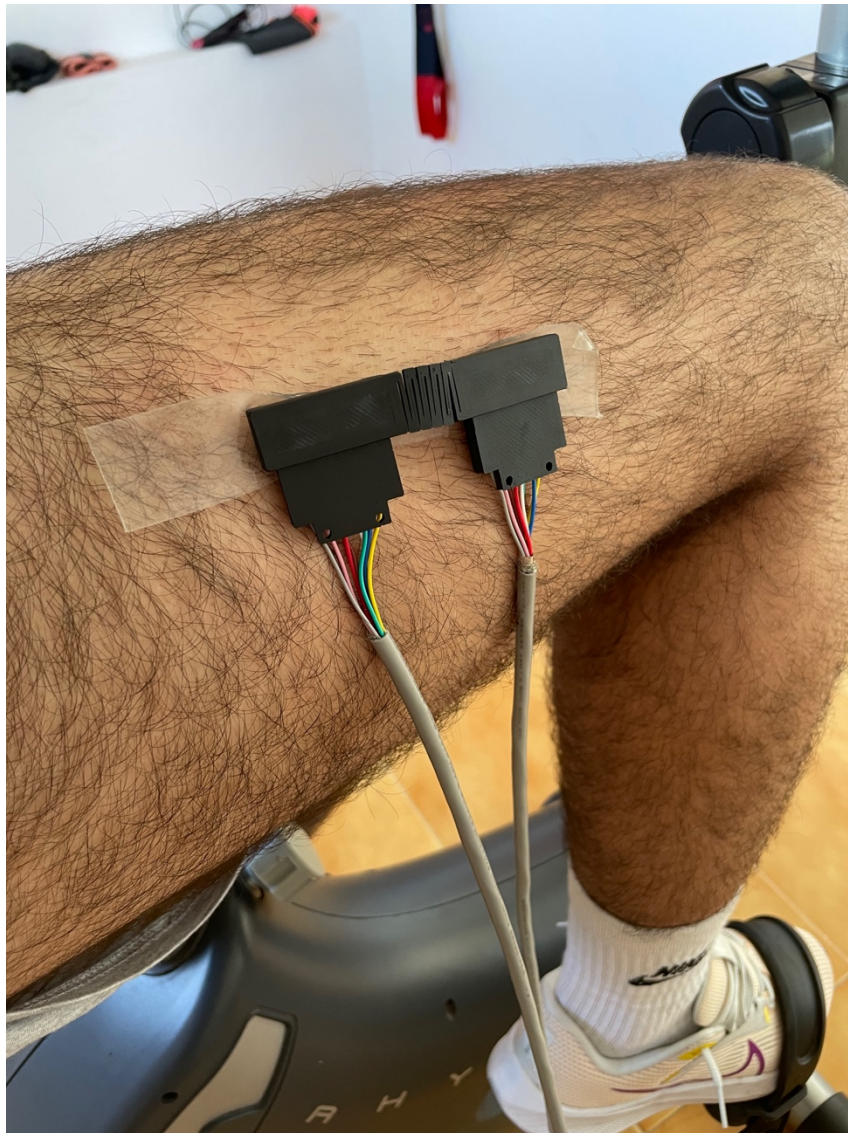


Figura 7.11. Colocación de las sondas NIRS 1



Figura 7.12. Colocación de las sondas NIRS 2

1	0,0,0,0,0,0	23	0,0,0,0,0,0
2	192,85,65,3,1,1	24	1,0,26,204,106,117
3	0,0,0,0,0,0	25	0,0,0,0,0,0
4	1,1,26,204,104,117	26	189,83,62,3,2,3
5	0,0,0,0,0,0	27	0,0,0,0,0,0
6	189,85,65,3,1,0	28	2,2,23,200,103,117
7	0,0,0,0,0,0	29	0,0,0,0,0,0
8	2,0,24,201,103,118	30	192,81,63,6,3,2
9	0,0,0,0,0,0	31	0,0,0,0,0,0
10	191,82,62,5,4,1	32	0,1,26,204,104,116
11	0,0,0,0,0,0	33	0,0,0,0,0,0
12	3,2,26,201,103,116	34	190,84,64,3,2,2
13	0,0,0,0,0,0	35	0,0,0,0,0,0
14	189,83,62,3,0,3	36	3,0,27,201,105,116
15	0,0,0,0,0,0	37	0,0,0,0,0,0
16	2,0,23,201,104,118	38	192,84,63,6,1,2
17	0,0,0,0,0,0	39	0,0,0,0,0,0
18	190,81,64,3,4,1	40	0,0,26,204,106,120
19	0,0,0,0,0,0	41	0,0,0,0,0,0
20	3,3,25,200,103,116	42	188,84,64,3,3,0
21	0,0,0,0,0,0	43	0,0,0,0,0,0
22	193,83,65,5,2,1	44	2,3,30,201,103,117

Figura 7.13. Fichero de los datos NIRS capturados

Como se puede observar, en el fichero de la Figura 7.13 se alternan filas de valores medidos con filas de todo cero. Esto se debe a que el programa Python, comentado anteriormente, se diseñó de forma que cuando se reciba la señal activa de sincronismo se registre una fila entera de ceros en el documento. De esta forma se puede sincronizar nuestras medidas con el de resto de equipos ya que se tiene marcado el instante de tiempo en el que comenzó la medida correspondiente.

La estructura de cada trama es la siguiente:

LED735_A, LED810_A, LED850_A, LED735_B, LED810_B, LED850_B

Y al analizar exclusivamente las tramas que contienen valores, se puede apreciar una marcada diferencia entre los tres primeros valores de una trama (correspondiente a la sonda A) y los siguientes, observándose un patrón similar con el siguiente grupo de tres valores (correspondiente a la sonda B). Esto ocurre debido a que, como se explicó anteriormente, en la trama 1 únicamente se encienden los LEDs de la sonda A. En consecuencia, la luz captada por el receptor de dicha sonda es significativamente mayor que la captada por el receptor de la sonda B. Este mismo fenómeno se repite en la siguiente trama, donde se encienden los LEDs de la sonda B, y el proceso se repite de manera constante.

Considerando todos los aspectos anteriormente mencionados, se puede concluir que la parte hardware diseñada para la captura de datos mediante el uso de tecnología NIRS ha sido validada exitosamente. Se comprobó que el dispositivo es capaz de capturar datos de forma precisa y fiable durante la realización de pruebas físicas, cumpliendo así con los requisitos establecidos. Adicionalmente, se verificó que la señal de sincronismo opera de la manera esperada, garantizando su compatibilidad y funcionamiento con el resto de los equipos existentes en el laboratorio.

Es importante resaltar que todas las pruebas y validaciones se llevaron a cabo respetando la normativa vigente en términos de seguridad, lo cual se evidenció mediante la medición de corriente y tensión de cada componente. Esta validación no solo confirma la funcionalidad del dispositivo, sino también su seguridad y fiabilidad para su uso en entornos de investigación. En resumen, el *hardware* desarrollado

demuestra ser una herramienta efectiva y segura para la captura de datos mediante tecnología NIRS, proporcionando una base sólida para futuras pruebas de investigaciones que se quieran realizar.

7.3. Validación de la GUI

La validación de la GUI se decidió realizar en un entorno de uso real, aprovechando que el grupo de investigación tenía preparado un conjunto de sujetos para llevar a cabo pruebas de algometría. Estas pruebas se realizaron utilizando el equipo TRAINIRS v0.2, lo que implica que se empleó la PCB anterior para el control del equipo, en lugar de la desarrollada en este TFM. Esta decisión se basó en la necesidad de utilizar un equipo fiable y sin posibles problemas operativos, ya que los datos obtenidos de estas pruebas oficiales serían utilizados en las investigaciones del grupo de rendimiento humano. Por lo tanto, se optó por no emplear la PCB v0.3, dado que aún se encontraba en fase de desarrollo en el momento de la realización de dichas pruebas.

La prueba que se va a realizar tiene una duración de tres días, es decir, se van a tomar medidas del dolor experimentado por el sujeto en tres días diferentes bajo condiciones diferentes. El primer día se aplica al sujeto tanto el protocolo “*Gradual*” como el “*Square*” en estado de reposo al inicio del día. Luego, durante ese mismo día deberá practicar algún tipo de actividad física moderada tratando de desgastar el músculo. Al día siguiente se vuelve a someter a las mismas pruebas para comprobar cómo ha podido influir en su percepción del dolor el desgaste del músculo y la posible presencia de agujetas. Ese día volverá a realizar actividad física pero esta vez de intensidad elevada buscando así el mayor desgaste posible en sus músculos. Al tercer día se repetirá por última vez la prueba para ver cómo evoluciona el dolor en función del cambio en el tejido del músculo.

Este tipo de prueba permite recopilar datos para el estudio de una posible relación del aumento de la percepción del dolor con el daño en el tejido muscular ocasionado por la actividad física. Los datos obtenidos en las pruebas no se mostrarán en este TFM debido a la privacidad de los pacientes y que esta parte no es objeto de estudio en este TFM. Sin embargo, si se analizará el correcto funcionamiento o no de la interfaz gráfica diseñada.

Tras comprobar que los tres protocolos estaban bien implementados se le explicó al personal encargado de realizar la prueba como debía utilizarse el equipo para así familiarizarse con su funcionamiento. Una vez hecho esto, ellos se encargaron de dirigir las pruebas a un total de 20 sujetos, a los cuales se le realizaban las pruebas 3 veces, por lo tanto, se realizaron un total de 60 pruebas reales las cuales permitían comprobar el funcionamiento de la interfaz de forma exhaustiva. En la Figura 7.14 se puede ver como el sujeto ya tiene colocadas las mangas de presión y tiene el botón de algometría en el mano preparado para realizar el protocolo de aumento gradual de presión. Por otro lado, en la Figura 7.15 se muestra una foto de la GUI durante la prueba donde se puede ver la lectura de presiones en tiempo real.



Figura 7.14. Sujeto listo para la realización del protocolo gradual

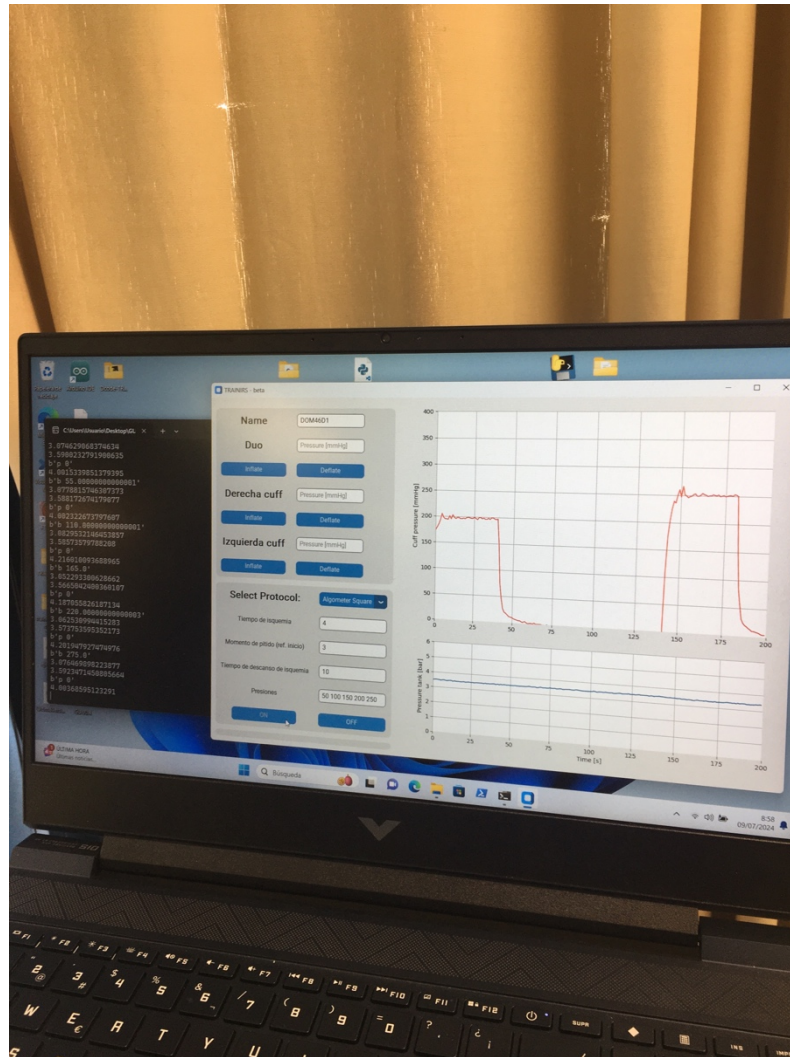


Figura 7.15. Lectura en tiempo real de la prueba que se está realizando

Atendiendo a los resultados obtenidos, la interfaz gráfica respondió de manera óptima en todas las pruebas realizadas, mostrando una gran estabilidad y fiabilidad. No se presentaron problemas técnicos ni fallos operativos, lo que confirma su robustez y su validez para su uso en estudios clínicos y de investigación. Estos resultados demuestran que la interfaz gráfica es una herramienta eficaz para la gestión y control tanto de los protocolos de algometría como del equipo en sí, facilitando la recopilación y análisis de los datos de manera eficiente. En la Figura 7.16 se puede ver la estructura del fichero obtenido para uno de los sujetos que se sometió a las pruebas. Si se mira con detalle en la Figura 7.17 se puede ver como quedó perfectamente registrado el momento exacto y la presión en el que la persona pulsó el botón de algometría implementado.

Time	pressure_ITV_A	target_ITV_A	pressure_ITV_B	target_ITV_B	pressure_tank	compressor_status	compressor_counter	bot_status	timer_ITV_A	timer_ITV_B
51.32	-8	52	-7	52	3.81	0	65535	0	0	0
51.421	-8	52	-8	52	3.79	0	65535	0	0	0
51.52	-9	52	-9	52	3.79	0	65535	0	0	0
51.619	-8	52	-8	52	3.77	0	65535	0	0	0
51.719	-8	52	-8	52	3.75	0	65535	0	0	0
51.819	-8	52	-8	52	3.77	0	65535	0	0	0
51.921	-9	52	-9	52	3.81	0	65535	0	0	0
52.021	-9	52	-9	52	3.81	0	65535	0	0	0
52.12	-9	52	-9	52	3.8	0	65535	0	0	0
52.22	-7	52	-8	52	3.78	0	65535	0	0	0
52.32	-7	52	-8	52	3.77	0	65535	0	0	0
52.419	-8	52	-8	52	3.78	0	65535	0	0	0
52.508	-8	52	-8	52	3.86	0	65535	0	0	0
52.618	-9	52	-9	52	3.78	0	65535	0	0	0
52.721	-8	52	-8	52	3.82	0	65535	0	0	0
52.82	-9	52	-8	52	3.79	0	65535	0	0	0
52.92	-9	52	-8	52	3.79	0	65535	0	0	0
53.02	-8	52	-8	52	3.78	0	65535	0	0	0
53.12	-9	52	-9	52	3.77	0	65535	0	0	0
53.219	-9	52	-8	52	3.79	0	65535	0	0	0
53.321	-8	52	-7	52	3.8	0	65535	0	0	0
53.422	-9	52	-9	52	3.78	0	65535	0	0	0
53.521	-8	52	-8	52	3.81	0	65535	0	0	0
53.62	-9	52	-8	52	3.77	0	65535	0	0	0
53.72	-9	52	-9	52	3.77	0	65535	0	0	0
53.819	-7	52	-7	52	3.77	0	65535	0	0	0
53.919	-9	52	-9	52	3.77	0	65535	0	0	0
54.022	-9	52	-9	52	3.82	0	65535	0	0	0
54.121	-8	52	-9	52	3.84	0	65535	0	0	0
54.221	-9	52	-8	52	3.81	0	65535	0	0	0
54.321	-9	52	-9	52	3.8	0	65535	0	0	0
54.42	-8	52	-9	52	3.78	0	65535	0	0	0
54.519	-10	52	-7	52	3.77	0	65535	0	0	0
54.619	-9	52	-8	52	3.79	0	65535	0	0	0
54.719	-9	52	-9	52	3.78	0	65535	0	0	0

Figura 7.16. Fichero de datos generado para uno de los sujetos

299.42	190	211	-10	0	3.28	0	65535	0
299.521	192	211	-8	0	3.29	0	65535	0
299.621	192	211	-8	0	3.24	0	65535	1
299.721	193	211	-9	0	3.27	0	65535	1

Figura 7.17. Momento en el cual se registra cuando el usuario pulsa el botón durante la prueba

En conclusión, la interfaz gráfica ha superado con éxito todas las pruebas de validación, cumpliendo con los objetivos establecidos y demostrando su potencial para aplicaciones futuras en el ámbito de la investigación del dolor.

8. Conclusión

En este capítulo se muestran las conclusiones conseguidas tras la configuración y el montaje de cada una de las partes del sistema, así como las de los experimentos realizados. Finalmente, se presentan una serie de posibles ampliaciones que se podrían llevar a cabo.

8.1 Conclusiones y posibles ampliaciones

En conclusión, nuestro dispositivo, aunque aún se encuentra en sus primeras etapas de desarrollo ya que este es un proyecto que comenzó recientemente, ha mostrado resultados prometedores que sientan las bases para posibles avances significativos en los campos de los deportes de alto rendimiento y la salud. El objetivo principal de este trabajo fue desarrollar un dispositivo innovador con aplicaciones en estos ámbitos, abordando las limitaciones de los dispositivos existentes mediante la fabricación e implementación de una PCB de control para la parte neumática del dispositivo y el diseño y fabricación de un subsistema NIRS capaz de medir la oxigenación del músculo. Además, se propuso la implementación de una interfaz tanto gráfica como manual aumentado así las posibilidades de uso del equipo.

A través de la fabricación e implementación que se hizo de la nueva PCB de control para la parte neumática se superó con éxito las limitaciones de los dispositivos existentes en el mercado. Se integró una interfaz manual que permite la interacción con el equipo y retroalimentación en tiempo real a través de la pantalla LCD. Además, se mejoró la red de alimentación permitiendo el uso de un DAC, que requiere de una modificación de su enrutado, y se incorporó un botón de algometría para la posible interacción con el paciente. Estas mejoras dotaron al equipo de características y funcionalidades únicas que incrementan considerablemente sus posibilidades de uso. Las pruebas de validación realizadas, mediante la medición de las diferentes señales obtenidas de la placa, permitieron comprobar que la PCB funciona correctamente utilizando ambas interfaces, y que el botón de algometría opera eficazmente.

El diseño del subsistema NIRS llevado a cabo sienta las bases para futuras etapas del proyecto donde se podrá obtener una gran cantidad de información valiosa sobre el comportamiento celular de los músculos permitiendo así el avance en la búsqueda de nuevos tratamientos médicos y deportivos. A este subsistema NIRS se le suma la fabricación de sus propias sondas de medidas, que gracias al diseño implementado se pueden obtener mediciones tanto de la oxigenación del músculo como de la piel. Gracias a las pruebas de validación realizadas, en este caso se optó por realizar pruebas en situaciones reales de uso como una sesión de bicicleta estática, se comprobó que se

pueden capturar y almacenar datos de forma exitosa de la actividad celular del músculo durante el entrenamiento.

El nuevo diseño que se realizó de interfaz gráfica permitió incrementar en gran medida las funcionalidades del equipo existente. Tras las pruebas realizadas se comprobó que la nueva GUI ha facilitado significativamente el uso del dispositivo, permitiendo a los usuarios interactuar de manera intuitiva y eficiente con el equipo. Además, el desarrollo de protocolos de algometría ha sido fundamental para la realización de investigaciones completamente nuevas. De hecho, la validación de la GUI se ha realizado a través de pruebas reales que el grupo de rendimiento llevó a cabo para una nueva línea de investigación iniciada gracias al nuevo dispositivo.

La combinación de todas las implementaciones realizadas en este TFM posiciona este proyecto como un posible referente en el campo, proporcionando herramientas valiosas para futuras investigaciones y aplicaciones prácticas. El desarrollo de este dispositivo podría conducir a soluciones innovadoras en la ciencia del deporte y en las aplicaciones médicas, contribuyendo a una sociedad mucho más saludable. Aunque el dispositivo está en sus primeras etapas, los resultados obtenidos son prometedores y sugieren un impacto significativo en su ámbito de aplicación.

Sin embargo, es importante reconocer las limitaciones del trabajo y que deberán ser subsanadas en futuras etapas de diseño. En primer lugar, será necesario realizar un procesamiento de las señales capturadas mediante las sondas NIRS para la correcta interpretación de los datos capturados. Para ello se hará uso de otros equipos disponibles en el laboratorio, de ahí que la implementación de la señal de sincronismo que se realizó fuera tan importante para garantizar la cooperación de nuestro sistema con el entorno. Además, tras las pruebas implementadas se recomienda que en futuras etapas se fabrique una nueva PCB para el subsistema NIRS que incorpore los espejos de corriente en la propia placa pudiendo así evitar el uso de una *protoboard*. Este cambio permitirá seguir aumentando la robustez del sistema y así poder ser usado en pruebas físicas todavía más exigentes.

En resumen, el dispositivo desarrollado en este TFM ha demostrado ser un avance prometedor a pesar de sus limitaciones, ofreciendo mejoras significativas sobre

los dispositivos actuales del mercado. Teniendo todo esto en cuenta, las perspectivas futuras son optimistas y sugieren que este dispositivo podría convertirse en una herramienta de valor en los campos de los deportes de alto rendimiento y la salud, contribuyendo a la innovación y al bienestar de la sociedad.

9. Bibliografía

9.1 Bibliografía

- [1] S. D. Patterson *et al.*, “Blood flow restriction exercise position stand: Considerations of methodology, application, and safety,” *Frontiers in Physiology*, vol. 10, no. MAY. Frontiers Media S.A., 2019. doi: 10.3389/fphys.2019.00533.
- [2] A. Megias and A. Saborido, “Adaptación muscular al ejercicio.”
- [3] C. Reina-Ramos and R. Domínguez, “Entrenamiento con restricción del flujo sanguíneo e hipertrofia muscular,” *RICYDE: Revista Internacional de Ciencias del Deporte*, vol. 10, no. 38, pp. 366–382, Oct. 2014, doi: 10.5232/ricyde2014.03806.
- [4] C. Pignanelli, D. Christiansen, and J. F. Burr, “Blood flow restriction training and the high-performance athlete: science to application,” *Journal of Applied Physiology*, vol. 130, no. 4. American Physiological Society, pp. 1163–1170, Apr. 01, 2021. doi: 10.1152/jappphysiol.00982.2020.
- [5] “Entrenamiento deportivo: qué es, principios y tipos | Ciencias Deportivas.” Accessed: Jun. 26, 2024. [Online]. Available: <https://cienciasdeportivas.com/entrenamiento-deportivo-puntos-principales/>
- [6] “ENTRENAMIENTO con RESTRICCIÓN del FLUJO SANGUÍNEO para escalada: BFR o RFS.” Accessed: Apr. 01, 2024. [Online]. Available: <https://www.pasoclave.com/entrenamiento-restriccion-flujo-sanguineo-bfr-rfs/>
- [7] C. Reina-Ramos and R. Domínguez, “Entrenamiento con restricción del flujo sanguíneo e hipertrofia muscular,” *RICYDE: Revista Internacional de Ciencias del Deporte*, vol. 10, no. 38, pp. 366–382, Oct. 2014, doi: 10.5232/ricyde2014.03806.
- [8] Y. Sato, “The history and future of KAATSU Training,” vol. 1, pp. 1–5, 2005.
- [9] Tomohiro Yasuda, Miwa Meguro, Yoshiaki Sato, and Toshiaki Nakajima, “Use and safety of KAATSU training: Results of a national survey in 2016,” 2017.
- [10] Y. Takarada, H. Takazawa, and N. Ishii, “Applications of vascular occlusion diminish disuse atrophy of knee extensor muscles,” 2000. [Online]. Available: <http://journals.lww.com/acsm-msse>
- [11] H. Takano *et al.*, “Hemodynamic and hormonal responses to a short-term low-intensity resistance exercise with the reduction of muscle blood flow,” *Eur J Appl Physiol*, vol. 95, no. 1, pp. 65–73, Sep. 2005, doi: 10.1007/s00421-005-1389-1.
- [12] J. M. Wilson, R. P. Lowery, J. M. Joy, J. P. Loenneke, and M. A. Naimo, “PRACTICAL BLOOD FLOW RESTRICTION TRAINING INCREASES ACUTE DETERMINANTS OF HYPERTROPHY WITHOUT INCREASING INDICES OF MUSCLE DAMAGE.” [Online]. Available: www.nsc.com
- [13] J. P. Loenneke and T. J. Pujol, “The Use of Occlusion Training to Produce Muscle Hypertrophy.” [Online]. Available: <http://journals.lww.com/nsca-scj>
- [14] J. P. Loenneke, K. C. Young, J. M. Wilson, and J. C. Andersen, “Rehabilitation of an osteochondral fracture using blood flow restricted exercise: A case review,” *J Bodyw Mov Ther*, vol. 17, no. 1, pp. 42–45, Jan. 2013, doi: 10.1016/j.jbmt.2012.04.006.
- [15] E. Oragui, A. Parsons, T. White, U. G. Longo, and W. S. Khan, “Tourniquet use in upper limb surgery,” *Hand*, vol. 6, no. 2. pp. 165–173, Jun. 2011. doi: 10.1007/s11552-010-9312-6.
- [16] “isquemia | Definición | Diccionario de la lengua española | RAE - ASALE.” Accessed: Jun. 25, 2024. [Online]. Available: <https://dle.rae.es/isquemia>
- [17] D. Wyser, O. Lamercy, F. Scholkmann, M. Wolf, and R. Gassert, “Wearable and modular functional near-infrared spectroscopy instrument with multidistance

- measurements at four wavelengths,” *Neurophotonics*, vol. 4, no. 04, p. 1, Aug. 2017, doi: 10.1117/1.nph.4.4.041413.
- [18] “KiCad EDA - Schematic Capture & PCB Design Software.” Accessed: Jun. 27, 2024. [Online]. Available: <https://www.kicad.org/>
- [19] “Software | Arduino.” Accessed: Jun. 27, 2024. [Online]. Available: <https://www.arduino.cc/en/software>
- [20] P. Andersen and A. Lundberg, “Non-invasive optical spectroscopy and imaging of human brain function,” Oxford University Press & Young, 1997.
- [21] M. Ferrari and V. Quaresima, “A brief review on the history of human functional near-infrared spectroscopy (fNIRS) development and fields of application,” *NeuroImage*, vol. 63, no. 2, pp. 921–935, Nov. 01, 2012. doi: 10.1016/j.neuroimage.2012.03.049.
- [22] F. Scholkmann *et al.*, “A review on continuous wave functional near-infrared spectroscopy and imaging instrumentation and methodology,” *NeuroImage*, vol. 85, pp. 6–27, Jan. 15, 2014. doi: 10.1016/j.neuroimage.2013.05.004.
- [23] S. Lloyd-Fox, A. Blasi, and C. E. Elwell, “Illuminating the developing brain: The past, present and future of functional near infrared spectroscopy,” *Neuroscience and Biobehavioral Reviews*, vol. 34, no. 3, pp. 269–284, Feb. 2010. doi: 10.1016/j.neubiorev.2009.07.008.
- [24] F. F. Jö Bsis-Vandervliet, “DISCOVERY OF THE NEAR-INFRARED WINDOW INTO THE BODY AND THE EARLY DEVELOPMENT OF NEAR-INFRARED SPECTROSCOPY,” 1999.
- [25] P. A. Jackson and D. O. Kennedy, “The application of near infrared spectroscopy in nutritional intervention studies,” *Frontiers in Human Neuroscience*, no. AUG. Frontiers Media S. A., Aug. 13, 2013. doi: 10.3389/fnhum.2013.00473.
- [26] A. C. Ehlis, M. J. Herrmann, A. Wagener, and A. J. Fallgatter, “Multi-channel near-infrared spectroscopy detects specific inferior-frontal activation during incongruent Stroop trials,” *Biol Psychol*, vol. 69, no. 3, pp. 315–331, Jul. 2005, doi: 10.1016/j.biopsycho.2004.09.003.
- [27] I. M. Kopton and P. Kenning, “Near-infrared spectroscopy (NIRS) as a new tool for neuroeconomic research,” *Frontiers in Human Neuroscience*, vol. 8, no. AUG. Frontiers Media S. A., Aug. 07, 2014. doi: 10.3389/fnhum.2014.00549.
- [28] A. A. Sanni and K. K. McCully, “Interpretation of near-infrared spectroscopy (NIRS) signals in skeletal muscle,” *J Funct Morphol Kinesiol*, vol. 4, no. 2, Jun. 2019, doi: 10.3390/jfmk4020028.
- [29] S. Ayala Pastorino, G. Varaldi Azcoytia, L. Illescas Calegari, and P. J. Castroman Espasandín Espasandín, “Test cuantitativo sensorial con algometría de presión: experiencia preliminar en un servicio de anestesiología,” *Revista de la Sociedad Española del Dolor*, 2022, doi: 10.20986/resed.2022.3885/2021.
- [30] C. KiCad, “KiCad, Herramienta de Software Libre de Modelado de Circuitos Impresos para el Desarrollo de Hardware,” vol. 38, no. 2, 2017.
- [31] “El lenguaje C++ — Fundamentos de Programación en C++.” Accessed: Jun. 27, 2024. [Online]. Available: https://www2.eii.uva.es/fund_inf/cpp/temas/1_introduccion/introduccion.html#:~:text=lenguaje%20C%2B%2B-,El%20lenguaje%20C%2B%2B,m%C3%A1s%20adelante%20en%20el%20curso.

- [32] "Qué es C++: Características y aplicaciones | OpenWebinars." Accessed: Jun. 27, 2024. [Online]. Available: <https://openwebinars.net/blog/que-es-cpp/>
- [33] "Arduino Mega 2560 Rev3 — Arduino Official Store." Accessed: Jun. 27, 2024. [Online]. Available: <https://store.arduino.cc/products/arduino-mega-2560-rev3>
- [34] "Illumination DS68 LUXEON Rebel Color Line Product Datasheet." Accessed: Jul. 01, 2024. [Online]. Available: <https://lumileds.com/wp-content/uploads/files/DS068-luxeon-rebel-color-line-datasheet.pdf>
- [35] "Marktech-MTE8120CP." Accessed: Jul. 01, 2024. [Online]. Available: <https://www.mouser.es/datasheet/2/1094/MTE8120CP-2999543.pdf>
- [36] "Marktech-MTE8600MC." Accessed: Jul. 01, 2024. [Online]. Available: <https://www.mouser.es/datasheet/2/1094/MTE8600MC-2999672.pdf>
- [37] "LECCIÓN 12 FOTODETECTORES 1.-INTRODUCCIÓN".
- [38] "Si PIN photodiode S6775 | Hamamatsu Photonics." Accessed: Jul. 01, 2024. [Online]. Available: <https://www.hamamatsu.com/jp/en/product/optical-sensors/photodiodes/si-photodiodes/S6775.html>
- [39] "Python: El Lenguaje de Programación que Impulsa la Innovación y la Eficiencia." Accessed: Jun. 28, 2024. [Online]. Available: <https://eiposgrados.com/blog-python/python-programacion-eficiencia/>
- [40] "3.9.12 Documentation." Accessed: Jun. 28, 2024. [Online]. Available: <https://docs.python.org/release/3.9.12/>
- [41] "Official Documentation And Tutorial | CustomTkinter." Accessed: Jun. 28, 2024. [Online]. Available: <https://customtkinter.tomschimansky.com/>
- [42] "ATMEL-2560", Accessed: Jul. 01, 2024. [Online]. Available: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf
- [43] A. Devices, "DAC-AD5724AREZ." Accessed: Jul. 01, 2024. [Online]. Available: https://www.mouser.es/datasheet/2/609/AD5724_5734_5754-1501775.pdf
- [44] "Mixed-signal and digital signal processing ICs | Analog Devices." Accessed: Jul. 01, 2024. [Online]. Available: <https://www.analog.com/en/index.html>
- [45] "THIDO - EC11 Potenciometro Encoder Rotativo Con Interruptor Push." Accessed: Jul. 06, 2024. [Online]. Available: <https://electronicathido.com/detallesProducto.php?id=aThlaEc1bENuVjZsZzFM RW1xb2ZKQT09>
- [46] "blog de avelino herrera morales - Multitarea apropiativa en microcontroladores: prueba de concepto sobre Arduino Leonardo." Accessed: Jul. 02, 2024. [Online]. Available: <http://avelinoherrera.com/blog/index.php?m=12&y=17&entry=entry171214-152214>
- [47] "TaskScheduler - Arduino Reference." Accessed: Jul. 02, 2024. [Online]. Available: <https://www.arduino.cc/reference/en/libraries/taskscheduler/>
- [48] "BC337-16 Series_B14-521022".
- [49] "PCB Prototype & PCB Fabrication Manufacturer - JLCPCB." Accessed: Jul. 04, 2024. [Online]. Available: <https://jlcpcb.com/>
- [50] "Arduino as ISP and Arduino Bootloaders | Arduino Documentation." Accessed: Jul. 10, 2024. [Online]. Available: <https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoISP/>

- [51] "Manguito para ecografía doppler vascular - MD6 - D. E. Hokanson." Accessed: Jul. 10, 2024. [Online]. Available: <https://www.medicalexpo.es/prod/d-e-hokanson/product-68146-421972.html>
- [52] "Rapid Cuff Inflation System | Hokanson." Accessed: Jun. 26, 2024. [Online]. Available: <https://hokansonvascular.com/products/133448>
- [53] "NIRS - Near InfraRed Spectroscopy - TRAINALYZED." Accessed: Jun. 28, 2024. [Online]. Available: <https://trainalyzed.com/en/2022/02/07/nirs-nahinfrarotspektroskopie/>
- [54] "KiCad 7.0 Released For This Open-Source Electronics Design Automation Suite - Phoronix." Accessed: Jun. 27, 2024. [Online]. Available: <https://www.phoronix.com/news/KiCad-7.0-Released>
- [55] "Discover the World of CustomTkinter: Your First GUI App with Hello, World! - python-hub." Accessed: Jun. 28, 2024. [Online]. Available: <https://python-hub.com/customtkinter-python-basics/>
- [56] "Push Button con Arduino - Automatización para Todos." Accessed: Jul. 02, 2024. [Online]. Available: <https://www.automatizacionparatodos.com/push-button-con-arduino/>

10. Pliego de Condiciones

En este episodio se detallan los aspectos relacionados con la información de licencias, los derechos de autoría y las responsabilidades. Así como las condiciones bajo las que se ha desarrollado este TFM y las que se establecen al usuario para poder usar el equipo que se ofrece.

PL1. Condiciones Hardware

Durante el desarrollo de este TFG se han usado los dispositivos hardware recogidos en la Tabla 10.1

Equipo	Modelo	Fabricante
Osciloscopio	500 Series	Hantek
Soldador	SS 207	Pro's Kit
Ordenador portátil	MacBook Pro 2021	Apple

Arduino Mega	Mega 2560	Arduino
Fuente de alimentación	TE-5305	Pro's Kit

Tabla 10.1. Condiciones del Hardware

PL2. Condiciones Software

En la Tabla 10.2 se exponen las herramientas software utilizadas, especificando su versión.

Aplicación	Versión
MacOS Monterrey	12.5.1
Microsoft Office	16.86
Python	3.9.12
KiCad	7.0.10
Visual Studio Code	1.87.2
Arduino IDE	2.3.2

Tabla 10.2. Condiciones del Software

PL3. Condiciones de licencia

El código desarrollado en este trabajo es propiedad de la Universidad de Las Palmas de Gran Canaria y todos los que pretendan hacer uso de él deben aceptar todas las cláusulas establecidas en esta licencia. El uso de las plataformas se podría hacer bajo autorización del autor, tutores del TFM, y la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria.

PL4. Derechos de autor

Tanto el código como la plataforma e información que se adjunta están protegidos por las leyes de propiedad intelectual que les sean aplicables, así como las disposiciones de los tratados internacionales. Por tanto, el código se considera un producto protegido por derechos de autor. Esto no es óbice para que una persona pueda copiar o utilizar el código bajo la autorización del autor, tutores del TFM, y la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria.

PL5. Restricciones

No se permite el uso de ingeniería inversa. Ni tampoco se permite la transferencia del código a un tercero.

PL6. Garantías

El autor del TFG lo presenta "AS IS" (tal cual), sin garantía implícita de ningún tipo. No se responsabiliza de los daños que pudieran causar a equipos o personas por el uso del código o la documentación. El autor no asegura, garantiza, o realiza ninguna declaración sobre el uso y resultados derivados de la utilización del código y/o del resto de la información proporcionada.

El código y la plataforma elaborada no están exentos de errores y no está diseñado para entornos de riesgo que requieran de un funcionamiento a prueba de fallos. El autor rechaza expresamente cualquier garantía explícita e implícita de adecuación del código para actividades de riesgo.

PL7. Limitación de responsabilidad

En ningún caso el autor ni los tutores, ni la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de las Palmas de Gran Canaria serían responsables de los perjuicios directos, indirectos incidentales o consiguientes, gastos, lucro cesante, pérdida de ahorros, interrupción de negocios, pérdida de información comercial o de negocio, o cualquier otra pérdida que resulte del uso o de la incapacidad de usar el código o la documentación. El usuario conoce y acepta este riesgo, así como el resto de las cláusulas y restricciones. El autor no reconoce otra garantía que no haya sido indicada anteriormente.

PL8. Otras consideraciones

En el supuesto de que cualquier disposición de esta licencia sea declarada total o parcialmente inválida, las cláusulas afectadas serían modificadas convenientemente

de manera que sean ejecutables una vez modificadas, permaneciendo el resto de este contrato en vigencia. Este contrato se rige por las leyes de España. El usuario acepta la jurisdicción exclusiva de los tribunales españoles con relación a las disputas derivadas de la presente licencia.

11. Presupuesto

Este capítulo contiene el presupuesto que recoge los gastos generados en la realización del presente TFM.

P.1 Componentes del presupuesto

El presupuesto calculado se divide en las siguientes partes:

- Recursos materiales.
- Trabajo tarifado por tiempo empleado.
- Aplicación de impuestos y coste total.

P.2 Recursos Materiales

Para la correcta ejecución y desarrollo del TFG han sido necesarios diversos recursos hardware y herramientas software, las cuales pueden llevar asociadas un coste en sus licencias.

Así cabe destacar que el software utilizado no conlleva un pago de licencia, a excepción del Microsoft Office.

Con el fin de establecer un cálculo de la amortización, se presupone el sistema de amortización como lineal, de tal forma que se asume que el inmovilizado material se desprecia de forma constante a lo largo de su vida útil y se estima un valor residual del 10% del valor de la adquisición.

Así, para llevar a cabo el cálculo de la cuota de amortización anual, se calcula usando la siguiente ecuación:

$$Cuota\ Anual = \frac{Valor\ de\ adquisición - Valor\ residual}{Vida\ útil}$$

De esta manera, la amortización total aparece reflejada en la Tabla 11.1, así como los diferentes valores y cuotas de los diferentes recursos empleados en este TFM.

Recurso	Unidades	Valor de adquisición (€)	Valor residual (€)	Vida útil (años)	Cuota anual (€)	Uso (meses)	Cuota aplicable (€)
MacBook Pro	1	1300	130	4	292,5	5	121,875
Arduino Mega	2	40	8	4	8	5	3,33
Capacitor_Tantalum_SMD:CP_EIA-3216-12	1	0,21	0,021	3	0,063	5	0,02625
Microprocesador	1	6,50	0,65	3	1,95	5	0,8125
Buck:HTSOP-J8_ROM	1	2,22	0,222	3	0,666	5	0,2775
Capacitor_SMD:C_1206_3216Metric	4	0,10	0,04	3	0,02	5	0,00833
Capacitor_THT	1	3,25	0,325	3	0,975	5	0,40625
Capacitor_SMD:C_1206_3216Metric_1u	1	0,14	0,014	3	0,042	5	0,0175
Capacitor_SMD:C_1206_3216Metric_22p	1	0,21	0,021	3	0,063	5	0,02625
Capacitor_SMD:C_1206_3216Metric_15n	1	0,70	0,07	3	0,21	5	0,0875
Capacitor_SMD:C_1206_3216Metric_47p	2	0,01	0,002	3	0,0026	5	0,0011
Capacitor_SMD:C_1206_3216Metric_10n	2	0,44	0,088	3	0,11733	5	0,0488
Capacitor_SMD:C_1206_3216Metric_30p	2	0,01	0,002	3	0,0026	5	0,0011
Diode_SMD:D_SMA-SMB_Universal_Handsoldering	1	0,36	0,036	3	0,108	5	0,045
DAC Converter	1	13,75	1,375	3	4,125	5	1,7187

Inductor_SMD:L_1206_3216Metric_Pad1.22x1.90m m_HandSolder	1	0,42	0,042	3	0,126	5	0,0525
Resistor_SMD:R_1206_3216Metric_10k	2	0,06	0,0112	3	0,0149	5	0,0062
Resistor_SMD:R_1206_3216Metric_27	1	1,96	0,196	3	0,588	5	0,245
Resistor_SMD:R_1206_3216Metric_2.2k	1	1,96	0,196	3	0,588	5	0,245
Resistor_SMD:R_1206_3216Metric_4.7k	1	1,51	0,151	3	0,453	5	0,18875
Potentiometer_THT:Potentiometer_Bourns_15k	1	3,52	0,352	3	1,056	5	0,44
Potentiometer_THT:Potentiometer_Bourns_10k	1	2,18	0,218	3	0,654	5	0,2725
USB2Serial1_Package_SO:SSOP	1	0,40	0,04	3	0,12	5	0,05
USB receptacle:CONN_10118192-0002LF_AMP	1	0,35	0,035	3	0,105	5	0,04375
Crystal Osc:XTAL_ECS-160-20-3X-EN-TR	1	33,83	3,383	3	10,149	5	4,22875
Fotorreceptor	1	3,58	0,358	3	1,074	5	0,4475
LED 735	1	36,62	3,662	3	10,986	5	4,5775
LED 810	1	11,46	1,146	3	3,438	5	1,4325
LED 850	2	4,83	0,966	3	1,288	5	0,5366
Cables 8 hilos 3 m	1	1300	130	4	292,5	5	121,875
Microsoft Office	1	0	0	0	0	0	0
KiCad	1	0	0	0	0	0	0
Arduino IDE	1	0	0	0	0	0	0
TOTAL							141,45

Tabla 11.1. Recursos Materiales

El coste de los materiales amortizables es de un total de: ciento cuarenta y un euros con cuarenta y cinco céntimos (141,45€).

P.3 Trabajo tarificado por tiempo empleado

Este concepto contabiliza los gastos que corresponden a la mano de obra, según el salario correspondiente a la hora de trabajo de un Ingeniero Técnico de Telecomunicaciones con un máster o equivalente.

Según la tabla retributiva de personal contratado en proyectos de investigación elaborada por la ULPGC en febrero de 2023, este salario, con una dedicación de 20 horas semanales, asciende a 833,13€ mensuales. Lo que se aproxima a una retribución de 10,41 €/h. Este TFM tal como comprende el Proyecto Docente ha conllevado 450 horas, por lo que se calcula el coste total por tiempo empleado en:

$$12,03 * 450 = 4684,50 \text{ €}$$

Por lo tanto, el trabajo tarificado por tiempo empleado asciende a la cantidad de cuatro mil seiscientos ochenta y cuatro euros con cincuenta céntimos (4.684,50€).

P.3 Aplicación de impuestos y coste total

El presupuesto total del presente TFG está gravado por el Impuesto General Indirecto Canario (IGIC), que está establecido en la actualidad en un siete por ciento (7 %). El coste total de este TFG se encuentra desglosado en la TABLA.

Descripción	Subtotal (€)
Recursos Materiales	141,45
Trabajo tarificado por tiempo empleado	5.413,50
Suma (€)	5.554, 95
IGIC (7%)	388,85
TOTAL	5.943,80

El presupuesto total del proyecto Sistema de control y monitorización de oclusión de flujo sanguíneo para la mejora del rendimiento físico y rehabilitación asciende a: cinco mil novecientos cuarenta y tres euros con ochenta céntimos (5.943,80€).

A handwritten signature in black ink, appearing to read 'Francisco Carlos Felipe Rodríguez', written in a cursive style.

Fdo.: D. Francisco Carlos Felipe Rodríguez (Manuscrita)

Fdo.: D. Francisco Carlos Felipe Rodríguez (Electrónica)

En Las Palmas de Gran Canaria a 17 de julio de 2024.

ANEXO

A continuación, se muestra el código simplificado para cada una de las partes diseñadas de proyecto. Principalmente se muestra la declaración de las variables básicas de funcionamiento y las principales funciones de cada programa, sin entrar en el detalle del mismo.

Software de la PCB de control

```
18 #define PRESSURE_DIAL A4
19 #define SET_BUTTON 26
20 #define FLUSH_BUTTON 27
21 #define encoder_CLK 48
22 #define encoder_DT 49
23 #define encoder_SWITCH 47
24 #define Tercer_BUTTON 28
25 #define ALGOMETER_BUTTON 29
26
27 #define GET_TANK_PRESSURE A1
28 #define GET_PRESSURE_ITV_A A2
29 #define GET_PRESSURE_ITV_B A3
30
31 // Actuators
32 #define EV_PURGUE 22
33 #define EVAPORATOR 23
34 #define FAN 24
35 #define COMPRESSOR 25
36 #define SET_PRESSURE_ITV_A 2
37 #define SET_PRESSURE_ITV_B 7
38
39 // Structures for easy and scalable pin definition
40 extern int input_digital_pins[];
41 extern int output_digital_pins[];
42 extern int input_analog_pins[];
43 extern int output_analog_pins[];
44
45 extern int input_digital_pins_size;
46 extern int output_digital_pins_size;
47 extern int input_analog_pins_size;
48 extern int output_analog_pins_size;
```

Figura A.1. Constantes del software de la PCB de control

```

57 // -----Task periodicity ----- //
58 class Task;
59 extern Task sampling_task;           // --> For acquiring pressure data
60 extern Task human_intefacing_task;   // --> For handling buttons
61 extern Task nirs_transfer_task;      // --> For transferring data to the PC
62 extern Task pressure_setting_task;   // --> For setting pressure
63 extern Task human_triggered_task;    // --> For performing an action when a
button is pressed
64 extern Task remote_control_task;     // --> For receiving remote control actions
65 extern Task compressor_activation_task; // --> For activating with hystheresis the
compressor
66 extern Task fan_function_task;       // --> For fan activation/desactivation
67 extern Task evaporator_function_task; // --> For convert liquid H2O into gas
68 extern Task purgue_function_task;    // --> For purgue EV
activation/desactivation
69 extern Task counter_ischemia_task;    // --> For purgue EV
activation/desactivation
70 extern Task algometer_task;          // --> For select the pain threshold
71 extern Task switch_task;            // --> For select the working mode
72
73 #define SAMPLING_PERIOD 100 // 100ms
74 #define HUMAN_INTERFACING_PERIOD 1
75 #define NIRS_TRANSFER_PERIOD 75
76 #define PRESSURE_SETTING_PERIOD 100 // 50ms
77 #define HUMAN_TRIGGERED_PERIOD 100
78 #define REMOTE_CONTROL_PERIOD 100
79 #define COMPRESSOR_PERIOD 1000
80 #define PURGUE_PERIOD 1000
81 #define EVAPORATOR_PERIOD 1000
82 #define LCD_PERIOD 100
83 #define COUNTER_ISCHEMIA_PERIOD 1000
84 #define SWITCH_PERIOD 100

```

Figura A.2. Tareas del TaskScheduler y sus períodos

```

// Function definitions for the above-declared tasks
void sampling_function();
void human_interfacing_function();
void nirs_transfer_function(); //
void pressure_setting_function();
void human_triggered_function(); //
void remote_control_function();
void compressor_activation_function();
void fan_function();
void evaporator_function();
void purgue_function();
void LCD_function();
void counter_ischemia_function();
void algometer_function();
void switch_function();

```

Figura A.3. Funciones del programa

```
void setup() {  
  // Pin initialization  
  initialize_pins(input_digital_pins, INPUT, input_digital_pins_size);  
  initialize_pins(output_digital_pins, OUTPUT, output_digital_pins_size);  
  initialize_pins(input_analog_pins, INPUT, input_analog_pins_size);  
  initialize_pins(output_analog_pins, OUTPUT, output_analog_pins_size);  
  pinMode(encoder_CLK, INPUT_PULLUP);  
  pinMode(encoder_DT, INPUT_PULLUP);  
  pinMode(encoder_SWITCH, INPUT_PULLUP);  
  aLastState = digitalRead(encoder_CLK);  
  
  // Communication initialization  
  Serial.begin(115200);  
  Serial.setTimeout(SERIAL_TIMEOUT);  
}
```

Figura A.4. Setup del software de la PCB de control

Software de control NIRS

```
1  const int ledsSondaA[] = {48, 50, 52}; //LEDs de la sonda A
2  const int ledsSondaB[] = {49, 51, 53}; //LEDs de la sonda B
3  const int numLeds = 3;
4  const int fotodetectorA = A0;          //Fotorreceptor sonda A
5  const int fotodetectorB = A1;          //Fotorreceptor sonda B
6  const int delayBetweenSondas = 10; // 10 ms
7
8  //Estructura para almacenar los datos capturados por cada sonda
9  int lecturasSondaA[numLeds];
10 int lecturasSondaB[numLeds];
11
12 void setup() {
13
14     for (int i = 0; i < numLeds; i++) {
15         pinMode(ledsSondaA[i], OUTPUT);
16         pinMode(ledsSondaB[i], OUTPUT);
17     }
18     pinMode(fotodetectorA, INPUT);
19     pinMode(fotodetectorB, INPUT);
20     Serial.begin(115200);
21 }
22
```

Figura A.5. Constantes, variables e inicialización del software de control NIRS

```

1 import serial
2 import csv
3 import time
4 import datetime
5 from datetime import date
6 from datetime import datetime
7 import os
8
9 # Configuración del puerto serie
10 SERIAL_PORT = '/dev/tty.usbmodem1101'
11 BAUD_RATE = 115200
12 TIMEOUT = 1 # Tiempo de espera en segundos para leer desde el puerto serie
13
14 def obtener_directorio_actual():
15     return os.path.dirname(os.path.realpath(__file__))
16
17 def leer_puerto_serie():
18     try:
19         # Abre el puerto serie
20         with serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=TIMEOUT) as ser:
21             print(f"Conectado al puerto: {SERIAL_PORT} a {BAUD_RATE} baudios")
22             while True:
23                 # Lee una línea de datos del puerto serie
24                 linea = ser.readline().decode('utf-8').strip()
25                 if linea:
26                     procesar_linea(linea)
27     except serial.SerialException as e:
28         print(f"Error al abrir el puerto serie: {e}")
29
30 def procesar_linea(linea):
31     # Abre el archivo CSV
32     directorio_actual = obtener_directorio_actual()
33     csv_file = os.path.join(directorio_actual, 'NIRS_' + f"{datetime.now():%d-%m-%Y_%H%M}" + '.csv')
34     with open(csv_file, 'a', newline='') as archivo_csv:
35         escritor_csv = csv.writer(archivo_csv)
36         # Si la línea es '1', escribe '0,0,0,0,0,0'
37         if linea == '1':
38             datos = [0, 0, 0, 0, 0, 0]
39         else:
40             # Convierte la línea en una lista de enteros
41             datos = [int(valor) for valor in linea.split(',')]
42
43         # Escribe los datos en el archivo CSV
44         escritor_csv.writerow(datos)
45         print(f"Datos escritos: {datos}")
46
47 if __name__ == "__main__":
48     leer_puerto_serie()

```

Figura A.6. Programa Python para la captura de datos NIRS