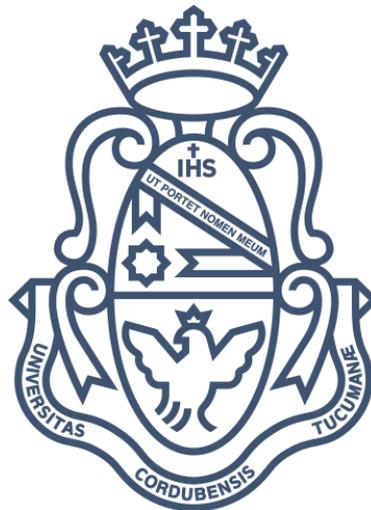


Universidad Nacional de Córdoba – Facultad de Ciencias Exactas Físicas y  
Naturales  
Ingeniería Biomédica

## Proyecto Integrador

# Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua



Alumno:

Sisko, Manuel Augusto

Matrícula:

37.763.611

Director:

Del Águila Heidenreich, Ronald



Córdoba, diciembre de 2018

## Agradecimientos

*A mi familia, por preocuparse siempre y esforzarse por ayudarme, aunque los kilómetros nos hayan dejado lejos.*

*A mi director, Ronald Del Águila, por brindarme su ayuda permanente y buscar los recursos para que este proyecto pudiera realizarse.*

*A mis amigos y colegas, en especial Franco y Agustín, por ayudarme cuando no lo pedí, pero si lo necesité.*

*Y a mis hermanos del Karate, porque me enseñaron a no rendirme nunca, por más que cueste y duela.*

## Resumen

*La medición de la presión arterial es uno de los procedimientos mas importantes de la medicina y se utiliza tanto en diagnóstico como en monitoreo. Los métodos de medición utilizados actualmente presentan desventajas que son difíciles de resolver.*

*En este Proyecto Integrador se diseñó y construyó un prototipo para medición de presión arterial que utiliza el método de compensación volumétrica, en el que se obtiene la presión arterial instantánea de forma continua y no invasiva. Se utilizó el diseño propuesto por Ken-Ichi Yamakoshi, con un fotopletismógrafo integrado en una almohadilla oclusora presurizada, para la ejecución de este método, que trata de colaborar en resolver algunas de las desventajas antes mencionadas.*

*El prototipo consiste de tres componentes principales. Los circuitos electrónicos, tanto de señales y amplificación como de potencia y alimentación, la programación de las herramientas digitales (software y firmware del microcontrolador) y el sistema neumático presurizado, con su bomba de compresión, sus electroválvulas, mangueras y conectores.*

*Se desarrollaron tres circuitos electrónicos diferentes, uno de amplificación y filtrado de señales, otro de potencia y alimentación de electroválvulas y un último circuito, de comunicación y control.*

*Para el componente neumático del prototipo se adaptó una rampa de inyección de GNC, para implementar un sistema de control de presión de aire comprimido en un sistema cerrado. Se diseñó una almohadilla oclusora con una membrana de Látex y se controló la presión del circuito neumático mediante una regulación por PWM de las electroválvulas y la bomba.*

*El Software se desarrolló en C++ y Qt 5, con la intención de que sea una herramienta de utilización simple. El microcontrolador utilizado fue un PIC18F2550 y su firmware se desarrolló con el compilador CCS. Las variables de control se establecieron mediante sistemas de lazo cerrado y controladores PID a nivel del software de PC.*

*Las mediciones obtenidas con el prototipo se compararon con las de un Monitor Multiparamétrico Phillips MP20, en donde se concluyó que la exactitud de las presiones Sistólica, Media y Diastólica obtenidas es aceptable, de acuerdo con los objetivos establecidos. Se determinó la calidad de la forma de onda obtenida por el prototipo mediante un análisis cualitativo, y se concluyó que esta presenta una forma general aceptable, pero con varios defectos, ocasionados por una lenta respuesta temporal del sistema de control de presión. Se propusieron modificaciones a futuro con el fin de mejorar la calidad de la forma de onda obtenida por el prototipo.*

## Contenido

Agradecimientos.....	- 1 -
Resumen.....	- 2 -
Contenido.....	3
Índice de Ilustraciones.....	6
Introducción.....	9
Objetivos .....	10
Objetivos Generales .....	10
Objetivos Específicos:.....	10
Capítulo 1: Marco Teórico. ....	11
El Sistema Circulatorio.....	11
El Corazón.....	11
El Árbol Vascular. ....	12
La Presión Arterial Normal: Forma de Onda.....	15
Medición de la Presión Arterial. ....	18
Métodos iniciales con Brazalete Constrictor. ....	18
El Método Auscultatorio.....	19
El Método Oscilométrico:.....	20
El Método Pletismo-Oscilométrico.....	21
Métodos Continuos No Invasivos.....	23
Métodos con la Velocidad de Onda de Pulso ( <i>PWV</i> ).....	23
Tonometría Aplanatoria. ....	24
Métodos de Descarga o Compensación Volumétrica. ....	25
Dispositivos que utilizan la Compensación Volumétrica. ....	26
Dispositivos Finapres.....	26
Dispositivos con Almohadilla Oclusora. ....	28
Importancia de la medición de la PA instantánea.....	31
Capítulo 2: Proceso de Diseño.....	33
Materiales Utilizados.....	34
Capítulo 3: Circuitos de Adquisición de Señal.....	36
El Amplificador de Transimpedancia. ....	38
Acondicionamiento de la Señal Fotopletismográfica. ....	41
Amplificación y Acondicionamiento de Señal de Presión. ....	45

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

Diseño y Construcción del Circuito: .....	49
Capítulo 4: El Sistema Neumático.....	51
La Almohadilla Oclusora y el Segmento Neumático Funcional. ....	52
La Bomba Neumática. ....	54
El Controlador de Presión.....	55
Integración del Sistema Neumático. ....	58
Circuito de Potencia: Alimentación de Bomba y Válvulas.....	59
Capítulo 5: El Circuito de Control y Comunicación.....	63
Selección del Microcontrolador.....	63
El Proceso de Comunicación.....	65
El Proceso de Muestreo y Empaquetamiento de Señales.....	67
El Proceso de Control del Sistema Neumático. ....	71
Capítulo 6: El Sistema de Control. ....	73
Modelo del Sistema Estable: El controlador de Presión. ....	76
Modelo del Sistema Dinámico: El Controlador de Volumen. ....	78
Capítulo 7: El Software de PC.....	81
Post Procesamiento de señales. ....	82
Proceso de Detección de Pulsos.....	83
Proceso de Calibración.....	84
Proceso de Recalibración.....	85
Capítulo 8: Normativa.....	87
Norma 93/42/CE.....	87
Requisitos de la Norma IEC 60601 y sus colaterales.....	88
Ensayos de Tipo:.....	89
Clasificaciones del Dispositivo:.....	90
Protección contra choque eléctrico:.....	90
Norma IEC 60601-2-30.....	92
Capítulo 9: Mediciones y Resultados. ....	93
Respuesta en Frecuencia del Filtro Pasa Banda.....	94
Calibración del Sensor de Presión.....	94
Comparación con Monitor Multiparamétrico. ....	96
Formas de Onda: Respuesta Temporal del Sistema. ....	100
Conclusiones y Observaciones. ....	105

Sobre la Exactitud de las Mediciones. ....	105
Sobre la Respuesta Temporal del Sistema.....	107
Sobre los Materiales. ....	109
Resumen de las Mejoras Propuestas. ....	111
Bibliografía y Referencias .....	112
Anexos .....	118
Anexo N° 1: Esquemáticos y PCBs.....	119
Circuito de Control y Comunicación.....	119
Circuito de Señales: Amplificación y Filtros. ....	121
Circuito de Potencia y Control de Válvulas.....	123
Anexo N° 2: Firmware del Microcontrolador. ....	125
Anexo N° 3: Código del Software de PC. ....	129

## Índice de Ilustraciones

Ilustración 1: Estructura del corazón y trayecto del flujo sanguíneo .....	11
Ilustración 2: Diagrama Volumen-Presión durante el ciclo cardíaco .....	12
Ilustración 3: Corte Histológico de un pedículo vascular .....	14
Ilustración 4: La onda de presión arterial, principales características ....	15
Ilustración 5: Forma de Onda Normal de la Presión Arterial .....	16
Ilustración 6: Comparación de las formas de onda de la presión arterial entre Cayado Aórtico y Arteria Femoral .....	17
Ilustración 7: Variación de la presión arterial aórtica – femoral .....	17
Ilustración 8: El método Oscilométrico .....	20
Ilustración 9: El método Pletismo-Oscilométrico .....	22
Ilustración 10: Esquema de Tonometría Aplanatoria .....	24
Ilustración 11: Esquema del método de compensación volumétrica. ....	25
Ilustración 12: Secuencia de Start-up y de re-calibración del Finapres .	27
Ilustración 13: Almohadilla neumática desarrollada por Yamakoshi .....	29
Ilustración 14: Posicionamiento y fijación de almohadilla .....	29
Ilustración 15: Comparación de las mediciones obtenidas durante un estudio de ejercitación por ciclismo .....	31
Ilustración 16: Formas de onda de presión arterial patológicas .....	32
Ilustración 17: Diagrama de bloques del prototipo .....	33
Ilustración 18: Procedimiento de diseño .....	34
Ilustración 19: Composición de una señal fotopletismográfica .....	37
Ilustración 20: Coef. de absorción lumínica de la sangre .....	38
Ilustración 21: El Amplificador de Transimpedancia .....	39
Ilustración 22; Capacidades parásitas del Amp. de Transimpedancia...	39
Ilustración 23: Respuesta temp. y en frec. de modulación simple AM...	42
Ilustración 24: Circuito y respuesta en frecuencia del filtro pasa banda	44
Ilustración 25: Topología de filtro Pasa Banda MFB fuente única .....	44
Ilustración 26: Circuito de Buffer para tensión de referencia .....	45
Ilustración 27: Sensor FPM-07PG .....	46
Ilustración 28: Amplificador Operacional en config. Diferenciadora .....	48
Ilustración 29: Circuito del Filtro Pasa Bajo y Resp. en frecuencia .....	49
Ilustración 30: Inestabilidad del amplificador de transimpedancia .....	50
Ilustración 31: Disco de PLA para el transductor y la almohadilla .....	53
Ilustración 32: Prototipo de la estructura retén de la almohadilla .....	54
Ilustración 33: Motor de corriente continua de la bomba neumática .....	55
Ilustración 34: Rampa de inyección de GNC Valtek de 5ta Gen .....	56
Ilustración 35: Modificaciones realizadas a la Rampa de Inyección. ....	57
Ilustración 36: Conector Automático Cruz de 6 mm. marca Intor .....	58
Ilustración 37: Diagrama del circuito neumático del prototipo .....	59
Ilustración 38: Configuración del Transistor MOSFET como Switch .....	60
Ilustración 39: Configuración del transistor PNP BJT como Switch .....	61

Ilustración 40: Circuito de control de Válvula SRS y bomba neumática	62
Ilustración 41: Circuito de control de Electroválvulas de 12 Vdc.....	62
Ilustración 42: Pinout del PIC 18F2550. <sup>12</sup> .....	64
Ilustración 43: Diagrama del recorrido de la señal fotopleismográfica. .	67
Ilustración 44: Representación de los puntos de muestreo .....	68
Ilustración 45: Ruido provocado por los desplazamientos de la señal...	69
Ilustración 46: Tren de pulsos para control de electroválvulas. ....	72
Ilustración 47: Diagrama de Bloques: Sistema de Presión Estable. ....	77
Ilustración 48: Oscilaciones utilizadas para el Método Ziegler Nichols. .	77
Ilustración 49: Respuesta temporal del sistema de control estable .....	78
Ilustración 50: Sistema de control con bucle anidado. ....	78
Ilustración 51: Diagrama de bloques: Sistema de control Dinámico .....	79
Ilustración 52: Resultados de medición sin predicción .....	80
Ilustración 53: Interfaz de usuario del software de PC.....	81
Ilustración 54: Respuesta en frecuencia del filtro IIR pasa bajo. ....	82
Ilustración 55: Ruido remanente previo al Post Procesamiento Digital. .	83
Ilustración 56: Calibración de medición inicial, sin recalibración.....	85
Ilustración 57: Fotografías del Prototipo finalizado .....	93
Ilustración 58: Respuesta en frec. del filtro pasa banda de 1.6 kHz. ....	94
Ilustración 59: Calibración del sensor de presión con columna de Hg... .	95
Ilustración 60: Calibración con Simulador Fluke. ....	95
Ilustración 61: Medición con método foto-oscilométrico .....	96
Ilustración 62: Medición 1, falange basal media .....	97
Ilustración 63: Medición 2, falange basal anular. ....	98
Ilustración 64: Medición 3, extremidad del dedo pulgar .....	99
Ilustración 65: Medición de presión, variación de la referencia.....	100
Ilustración 66: Inestabilidad de la medición por exceso de ganancia ..	101
Ilustración 67: Respuesta temporal del sistema de compensación .....	101
Ilustración 68: Predicción de pulso arterial inexacta. ....	102
Ilustración 69: Efecto de las fugas en la almohadilla. ....	103
Ilustración 70: Medición de presión arterial calibrada. ....	104
Ilustración 71: Formas de onda obtenidas con el prototipo .....	105
Ilustración 72: Variación de presión arterial detectada con VCM.....	107
Ilustración 73: Defectos de medición debidos a la respuesta lenta .....	108
Ilustración 74: Controlador de Presión <i>FESTO</i> de la serie <i>VEAB</i> .....	110
Ilustración 75: Esquemático del circuito de comunicación.....	119
Ilustración 76: PCB de comunicación. Top Layer. ....	120
Ilustración 77: PCB de comunicación. Bottom Layer. ....	120
Ilustración 78: Esquemático del circuito de amplificación y filtros.....	121
Ilustración 79: PCB del circuito de señales. Bottom Layer.....	122
Ilustración 80: PCB del circuito de señales. Bottom Layer.....	122
Ilustración 81: Esquemático del circuito de potencia .....	123
Ilustración 82: PCB del circuito de potencia. Bottom Layer .....	124
Ilustración 83: PCB del circuito de potencia. Top Layer.....	124



## Introducción

El cuerpo humano funciona como una máquina de gran precisión y complejidad. Su funcionamiento depende de una enorme cantidad de variables y factores, tanto externos como internos. Podríamos interpretar al organismo como un gran circuito electrónico, con múltiples componentes de carácter alineal y con una única fuente de alimentación, el corazón, que impulsa la energía a través de un circuito de conductores, el árbol vascular.

Así como en un circuito electrónico normal, en donde la fuente y las tensiones en cada punto del mismo son de primordial importancia, también lo son el funcionamiento del corazón y la presión arterial, en el organismo. Las características de la presión arterial son causa y efecto de incontables patologías, y existe una gran cantidad de investigación dedicada a caracterizar estos parámetros [1, 2].

La tecnología médica ha avanzado a pasos agigantados en el transcurso de los últimos cien años, y aún lo hace hoy en día. Pero estos avances no se han logrado de la misma forma en el campo de la medición de presión arterial. Los métodos de medición de la presión arterial para diagnóstico y seguimiento utilizados actualmente fueron desarrollados hace más de un siglo [3].

La medición de presión arterial para diagnóstico y seguimiento debe ser de un riesgo aceptable, es decir, debe ser de carácter no invasivo, y la medición no invasiva presenta grandes dificultades que atentan contra la exactitud y la cantidad de información obtenida por estos métodos. Es posible que esto haya ocasionado una investigación aminorada de los beneficios diagnósticos de la medición de presión arterial real, resultando en una baja inversión en el desarrollo de equipamiento diseñado para esta función.

Uno de nuestros objetivos como profesionales de la Bioingeniería, es la aplicación, implementación y desarrollo de nuevas tecnologías en el campo de la Salud y la Sanidad. Debemos encontrar formas de circunventar las dificultades que se presentan en el estudio de la fisiología, normal y patológica, del cuerpo humano, con el fin de garantizar herramientas efectivas en la práctica de la Medicina.

Este Proyecto Integrador intenta ser un aporte hacia una de las ramas de la tecnología electromédica que se encuentra menos desarrollada, pero cuya importancia podría ser mucho mayor de lo que se cree.

## Objetivos

### Objetivos Generales

Diseñar y construir un prototipo para medición de presión arterial en tiempo real y no invasiva, basado en el método de compensación volumétrica, capaz de obtener mediciones con una exactitud comparable a los instrumentos utilizados para esta función en la actualidad.

### Objetivos Específicos:

- Diseño y construcción de todos los circuitos electrónicos necesarios para el proceso de medición.
- Construcción de un circuito neumático capaz de ejecutar el proceso de medición.
- Desarrollo de un Software simple y práctico para el control del dispositivo, con protocolos fáciles de utilizar que requieran una capacitación mínima.
- Evaluar el correcto funcionamiento del prototipo, mediante la comparación de su medición con un equipo calibrado.

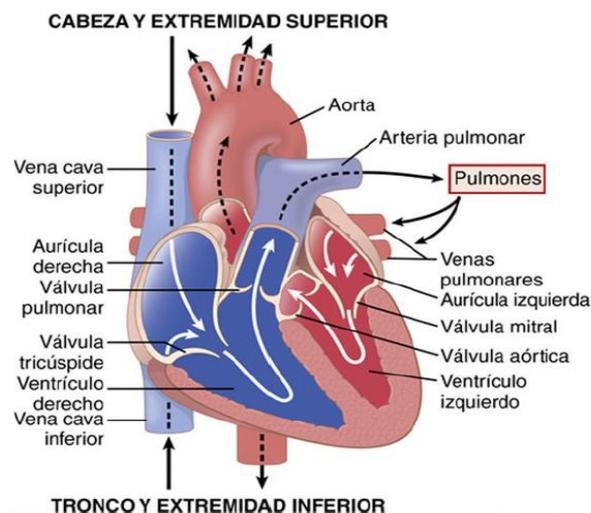
## Capítulo 1: Marco Teórico.

### El Sistema Circulatorio.

El sistema circulatorio en el adulto se compone de dos elementos principales, el corazón y el árbol vascular. El primero funciona como una bomba secuencial y eyecta sangre mediante contracciones coordinadas, mientras que el segundo actúa como un sistema de conducción dinámico y variable. Las contracciones cardíacas generan una presión que se transmite como un frente de onda, a través de la Aorta y hacia el resto del árbol vascular. Las características contráctiles del corazón, las condiciones morfológicas y fisiológicas del árbol vascular y de la composición de la sangre, constituyen los parámetros determinantes de la función circulatoria en el organismo humano [4].

#### El Corazón.

El corazón se encuentra situado en la parte media de la cavidad torácica y sus dimensiones y peso dependen de la edad y el sexo, en el hombre adulto, pesa aproximadamente 275 gramos, mide 98 milímetros de altura y 105 milímetros de amplitud. Este órgano está formado por cuatro cámaras, dos aurículas y dos ventrículos. De estas cuatro, aquella cuya función es más crítica para el organismo, y que presenta la mayor capacidad contráctil, es el ventrículo izquierdo, situado verticalmente sobre el espacio precordial, entre el 3er y 5to espacio intercostal izquierdo. Esta cavidad, separada de la Aurícula Izquierda por la válvula Mitral y del nacimiento de la Aorta por la válvula Aórtica, se contrae y relaja de forma rítmica, expulsando sangre en cada contracción (denominada sístole) y recibiendo sangre de la aurícula en cada relajación (diástole). La sangre atraviesa el árbol vascular impulsada por el gradiente de presiones generado por las contracciones cardíacas [4].



**Ilustración 1: Estructura del corazón y trayecto del flujo sanguíneo a través de las cavidades cardíacas y de las válvulas cardíacas [6]**

El ciclo cardíaco está formado por un período de relajación, que se denomina diástole, seguido de un período de contracción, denominado sístole. La duración del ciclo cardíaco total, incluida la sístole y la diástole, es el valor inverso de la frecuencia cardíaca. Por ejemplo, si la frecuencia cardíaca es de 72 latidos por minuto, la duración del ciclo cardíaco es  $1/72$  minutos, aproximadamente 0.833 segundos. Este ciclo consiste de cuatro períodos y se estudia centralizando en el ventrículo izquierdo. Inicialmente, el ventrículo en relajación, inicia la sístole con una contracción iso-volumétrica, en la que la presión intra-ventricular es inferior a la presión aórtica, por lo que la válvula aórtica permanece cerrada. Cuando la presión ventricular alcanza a la presión aórtica, la válvula se abre y comienza el período de eyección. Cuando la eyección finaliza, el volumen de sangre ventricular alcanza su valor mínimo (denominado volumen telesistólico), la válvula aórtica se cierra y comienza el período de relajación iso-volumétrica. Finalmente, la presión ventricular cae, a causa de la relajación del músculo cardíaco ventricular, por debajo de la presión auricular, ocasionando la apertura de la válvula mitral y el inicio del período de llenado ventricular [5].

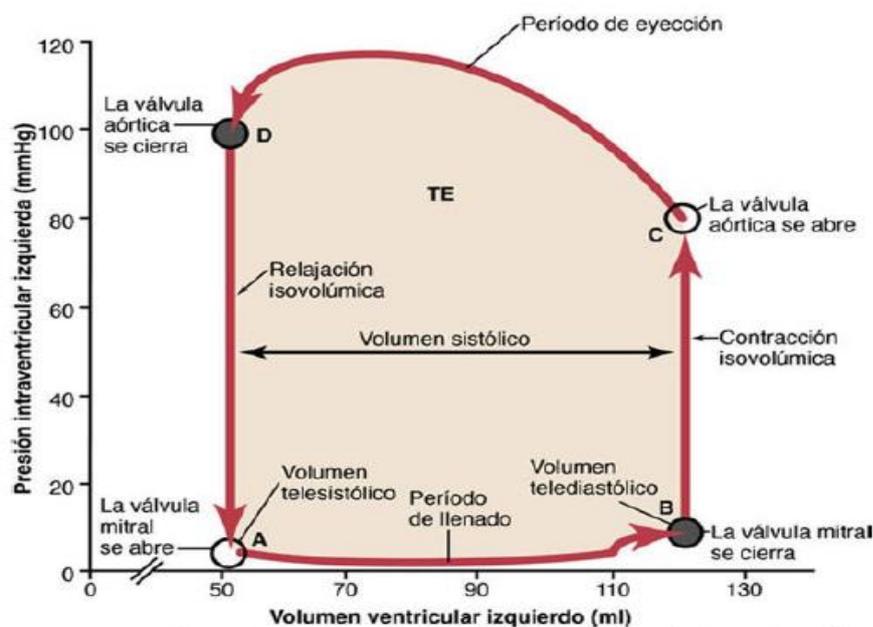


Ilustración 2: Diagrama Volumen-Presión durante el ciclo cardíaco, en el ventrículo izquierdo. Se destaca: (B) Inicio de la sístole ventricular; (D) Final de la sístole, e inicio de la diástole ventricular [5]

## El Árbol Vascolar.

El árbol vascular está constituido por las arterias, arteriolas, capilares, vénulas y venas, formando un circuito cerrado de contención y circulación sanguínea. Cada una de estas estructuras presentan diferencias morfológicas y funcionales, como también superficies transversales totales diferentes. Debido a estas últimas, el volumen total de contención de cada uno de estos segmentos

vasculares, es diferente. En consecuencia, la sangre se distribuye asimétricamente en los mismos, de modo tal que, del volumen de sangre total, el 64% se encuentra en venas, vénulas y senos venosos, mientras que solo el 20% se distribuye en arterias, arteriolas y capilares (de la circulación sistémica) [6]. Esto resulta de gran importancia, ya que como el flujo sanguíneo promedio es idéntico a través de todo el árbol vascular, se deduce que la proporción volumétrica de sangre transportada a través de las arterias es mucho mayor que en las venas, y por ende las presiones que se transmiten son muy superiores (desde 80-120 mmHg, en grandes arterias hasta 5-10 mmHg. en vénulas) [6].

Como se mencionó, no solo existen diferencias volumétricas entre los diferentes segmentos del árbol vascular, sino que también se observan diferencias morfológicas y funcionales. El factor determinante de estas diferencias, es la distensibilidad vascular. La distensibilidad vascular se expresa como el incremento fraccionado del volumen por cada milímetro de mercurio que aumenta la presión [7].

$$\text{Distensibilidad Vascular} = \frac{\text{Aumento del Volumen}}{\text{Aumento de Presión} \times \text{Volumen Original}}$$

**Ecuación 1: Definición de la distensibilidad vascular [7].**

La pared vascular de las arterias es mucho mas gruesa que en las venas, en consecuencia, estas últimas son hasta ocho veces mas distensibles [7]. Como consecuencia, el cambio de volumen que un determinado aumento de presión genera en la circulación venosa, es mucho menor, que el cambio que este mismo aumento de presión genera en la circulación arterial. Y además, como las variaciones de presión de la circulación venosa son muy inferiores y los volúmenes totales son mayores, que en la circulación arterial, variaciones volumétricas del árbol venoso ocasionadas por la contracción cardíaca, son despreciables con respecto a las generadas en el árbol arterial.

La pared vascular de los vasos sanguíneos está compuesta por tres capas, llamadas tunicas. La túnica íntima, que es la capa mas interna, constituida por el endotelio, la membrana basal y el subendotelio. La túnica media, formada principalmente por estratos circunferenciales de fibras musculares lisas, separadas entre sí por elastina, fibras reticulares y proteoglicanos. Y la túnica adventicia, que es la mas externa, formada por tejido conjuntivo, principalmente colágeno de distribución longitudinal. La diferencia morfológica mas importante entre venas y arterias, es el espesor y capacidad contráctil de la túnica media. En las arterias, esta capa es mucho mas gruesa, mientras que en las venas, la capa predominante es la adventicia, la que les brinda gran elasticidad [8].

El grado de contracción del músculo liso en la túnica media, se regula de forma sistémica, y modifica el diámetro del vaso, por lo que regula la resistencia periférica y el flujo de sangre que atraviesa este segmento. Esta pared muscular es muy gruesa en las arteriolas, provocando que la gran mayoría de la resistencia periférica del árbol vascular esté determinada por las condiciones contráctiles de las mismas. Debido a esto, se suelen llamar a las arteriolas, los “vasos de resistencia” del organismo. Como el grado de contracción de este músculo liso se regula de forma sistémica, el organismo es capaz de incrementar o disminuir la irrigación sanguínea, de acuerdo a las necesidades de cada tejido [8].

Como la resistencia periférica se constituye principalmente de su componente arteriolar, este segmento del árbol vascular constituye una “pared” funcional, sobre la que las ondas mecánicas de presión arterial se reflejan, generando variaciones características de la onda de presión [9].

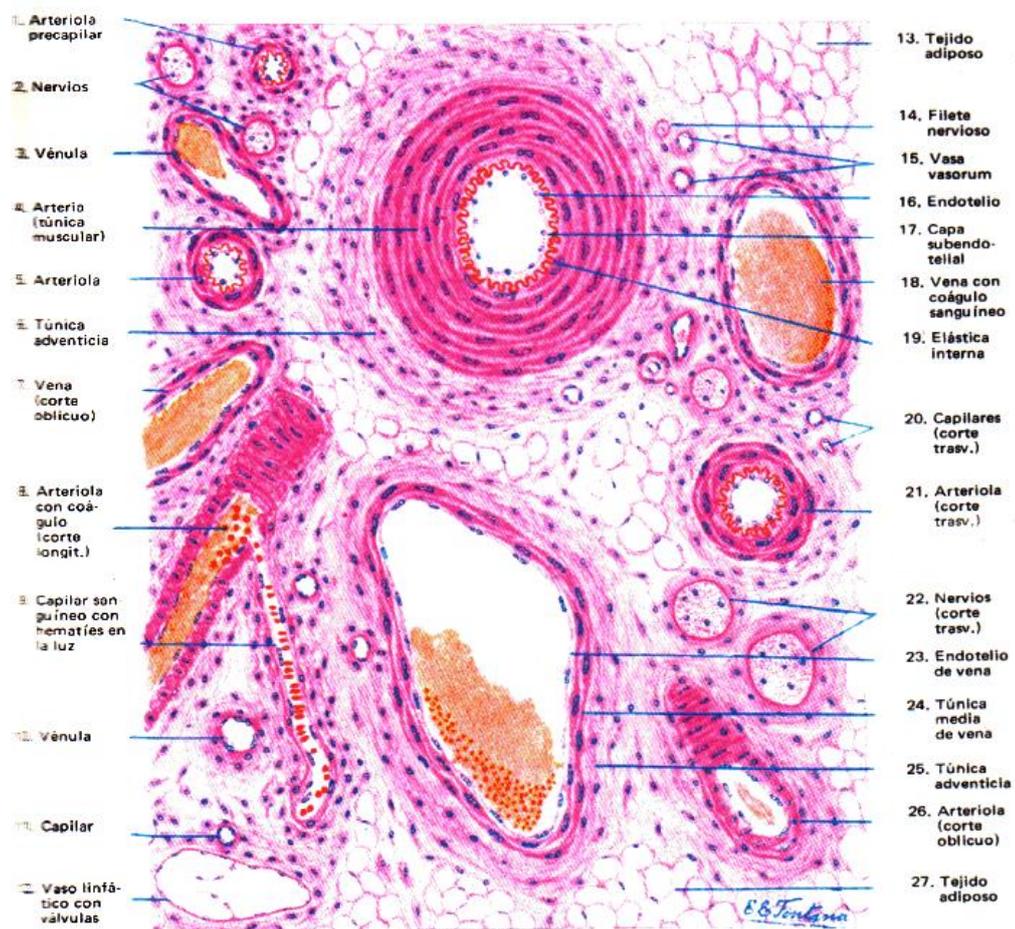


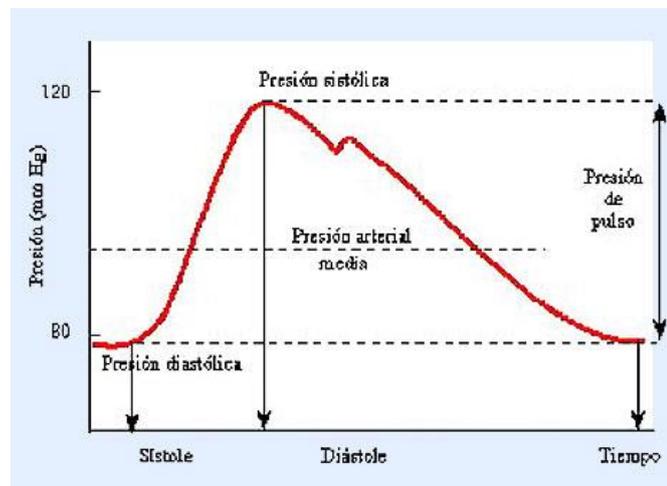
Ilustración 3: Corte Histológico de un pedículo vascular, teñido con Hematoxilina-Eosina. Aumento de 160x [8]

### La Presión Arterial Normal: Forma de Onda.

La presión arterial es la fuerza por unidad de área que la sangre ejerce sobre las paredes de las arterias en cada instante. Podemos destacar tres puntos principales de la misma:

- La presión sistólica, es la presión máxima que ejerce la sangre, coincidente con el punto de mayor contracción ventricular durante la sístole.
- La presión diastólica, es la presión mínima remanente durante la relajación ventricular.
- La presión arterial media, equivalente al valor promedio de la presión arterial pulsátil, a lo largo del tiempo.

Es importante destacar que, estos valores varían en diferentes partes del árbol arterial, si bien en el ventrículo izquierdo estas presiones van desde los 5 mmHg hasta los 150 mmHg., en las grandes arterias, la presión diastólica no cae (en condiciones normales) por debajo de los 60 mmHg [6].<sup>1</sup>



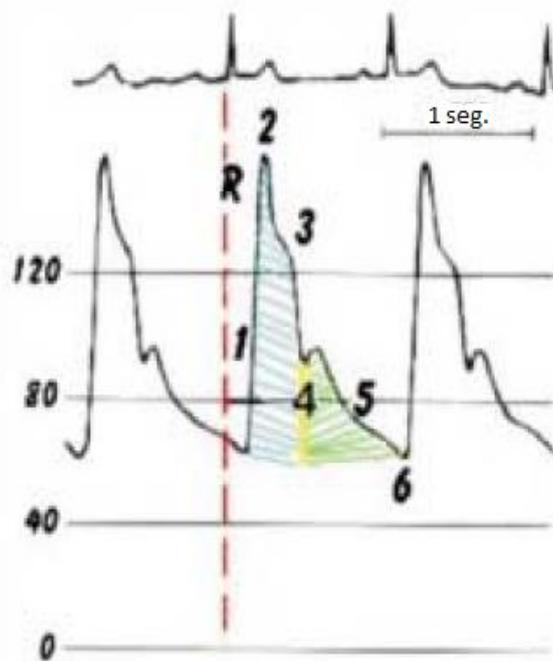
**Ilustración 4: La onda de presión arterial y sus principales características**

La onda de presión arterial es de naturaleza pulsátil, y es originada por la contracción ventricular izquierda. Esta onda se inicia varios milisegundos luego de la Onda R de la señal electrocardiográfica, esta última, producto de la despolarización de las fibras musculares del músculo cardíaco ventricular izquierdo. Este retraso, entre la Onda R y el inicio de la onda de presión arterial, es debido al tiempo que demoran: la despolarización de las fibras musculares, la

<sup>1</sup> Imagen extraída de <https://ocw.unican.es/mod/page/view.php?id=538>

contracción de la cámara ventricular, la apertura de la válvula aórtica y la eyección del volumen sistólico [1].

La onda de presión tiene una componente rápida y una lenta (de 10 m/seg. y 0,5 m/seg. respectivamente) [10], por lo que la onda de presión no se transmite de forma instantánea a todo el árbol vascular, sino que presenta una demora, dependiente de la distancia total desde el punto observado hasta el ventrículo izquierdo. Estas componentes de presión se suman junto con las componentes de presión que se reflejan en las paredes vasculares (mayormente, en el segmento arteriolar), y componen la forma de onda de presión arterial en cada segmento arterial.

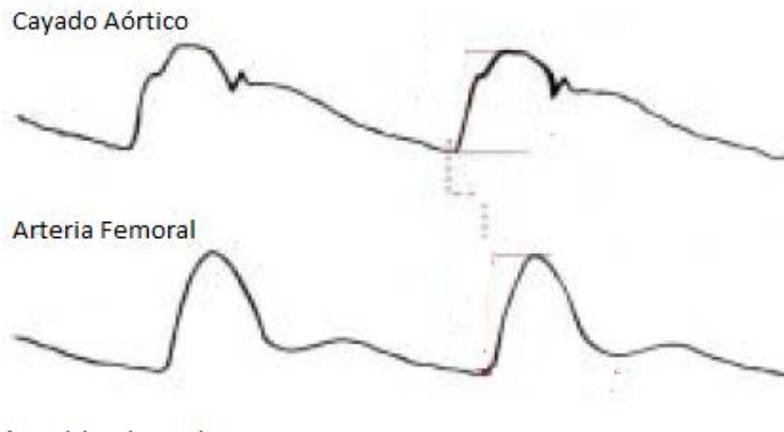


**Ilustración 5: Forma de Onda Normal de la Presión Arterial. 1- Frente Sistólico. 2- Presión Sistólica Máxima. 3- Caída Sistólica. 4- Muesca Dicrótica 5- Pendiente Diastólica 6- Presión Diastólica Final. R- Onda R del electrocardiograma [1]**

En consecuencia, la forma de la onda de presión arterial es diferente para cada sección del árbol arterial, no solo se modifica la amplitud de la onda, sino que también su morfología. En la Ilustración 5 se puede observar una forma de onda normal a nivel cardíaco. El pico abrupto ascendente es el resultado de la presurización súbita previa a la apertura de la válvula aórtica, y la eyección de sangre tiende a mantener dicha presión cuando esta se abre. La incisura o Muesca Dicrótica, a nivel aórtico, es provocado por el cierre de la misma válvula [1].

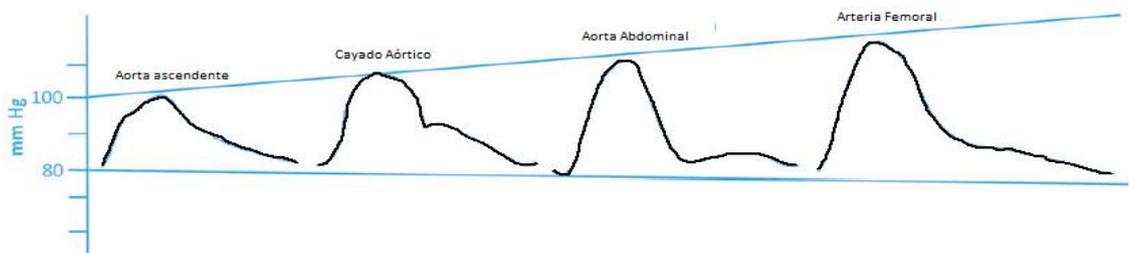
La forma de la onda se suaviza a medida que se aleja del cayado aórtico, la Ilustración 6 grafica dos curvas típicas de forma de onda a nivel aórtico y femoral. Se puede apreciar la similitud entre las formas, ambas presentan los

mismos puntos característicos. Sin embargo, existen discrepancias claras. A nivel periférico, la incisura dicrótica no es generada por el cierre de la válvula aórtica, sino que por la suma de la onda reflejada a nivel arteriolar [9]. Además, la onda se suaviza en vasos pequeños, debido a la resistencia opuesta por las arteriolas [9]. Por otro lado, la presión sistólica máxima es superior a la central, pero la presión media y la diastólica son inferiores, en consecuencia, la forma de onda presenta mayor amplitud [1].



**Ilustración 6: Comparación de las formas de onda de la presión arterial entre Cayado Aórtico y Arteria Femoral [1]**

No solo cambia la forma de onda a medida que se desplaza el punto de medición a lo largo del árbol arterial, sino que cambian también los valores de presión sistólica, media y diastólica. Estos cambios también se producen latido a latido, y son afectados también por la respiración del individuo [3]. Por ende, para interpretar la forma de onda de la señal, es importante identificar sus principales características. Identificar la calidad de la forma de onda es de crucial importancia, y se debe adaptar o corregir el método de medición de forma correspondiente en base a la misma [2].



**Ilustración 7: Variación de la presión arterial desde la válvula aórtica hasta la arteria femoral [2]**

## Medición de la Presión Arterial.

La medición de la presión arterial es de gran utilidad en Diagnóstico, Revisión y Seguimiento de innumerables patologías, procedimientos quirúrgicos e intervenciones médicas. Es por ello que se han desarrollado múltiples sistemas de medición de presión arterial. Estos sistemas aprovechan las propiedades básicas de la física de la circulación para determinar la presión arterial en varios puntos del árbol vascular.

Podemos distinguir dos grandes tipos de sistemas de medición:

1. **Medición Invasiva o Directa:** Utilizan un catéter introducido en un vaso sanguíneo de gran calibre. Dicho catéter cuenta con un transductor de presión, que convierte las variaciones de presión de la sangre en el punto de medición, en señales eléctricas que luego son amplificadas y acondicionadas. Estos métodos presentan riesgos elevados para los pacientes, pero cuentan con la ventaja de ser los métodos más exactos de medición de la presión en todos los puntos donde se puede insertar un catéter de medición [3].
2. **Medición No Invasiva o Indirecta:** Utilizan diferentes principios de la física circulatoria para obtener aproximaciones o estimaciones de la presión arterial, sin la necesidad de un procedimiento quirúrgico. Presentan un costo reducido, riesgos bajos o nulos para el paciente y una gran utilidad para diagnóstico y seguimiento. Existen varios métodos diferentes, la mayoría de ellos permiten obtener estimaciones de la presión sistólica, diastólica y/o media, pero algunos permiten además, una medición de la presión arterial instantánea.

Otra clasificación importante de los procesos de medición, es con respecto a los resultados que se obtienen. Hablamos de sistemas de medición **continuos o de presión instantánea**, cuando el sistema es capaz de estimar la presión arterial en cada instante a lo largo de un período determinado de tiempo. Algunos autores también utilizan el término “continuo” para caracterizar a los equipos que pueden realizar múltiples mediciones a lo largo de períodos prolongados, sin la intervención del operador, pero nos referiremos a dichos equipos como “automáticos” de aquí en más.

En el desarrollo de este proyecto integrador nos enfocamos únicamente en los sistemas de medición indirecta.

### Métodos iniciales con Brazalete Constrictor.

Cuando el brazalete constrictor para miembros fue introducido a principios del siglo XX, se comenzó a medir la presión sistólica mediante métodos poco prácticos y altamente variables. Por ej., en el método palpatorio, el médico estimaba la presión sistólica luego de que la arteria radial hubiera sido ocluida por

el brazalete a nivel braquial, palpando el pulso radial en la muñeca a medida que se reducía lentamente la presión constrictora del brazalete. En otro método, se ocluía la arteria radial a nivel braquial con una venda, una vez el miembro se tornaba blanco (debido al cese de la irrigación), se colocaba el brazalete presurizado y se soltaba la venda para luego despresurizar gradualmente el brazalete. La presión sistólica se estimaba a partir de la presión en la que el paciente comenzaba a sentir la circulación sanguínea y a la que la extremidad recuperaba el color rosado original [3].

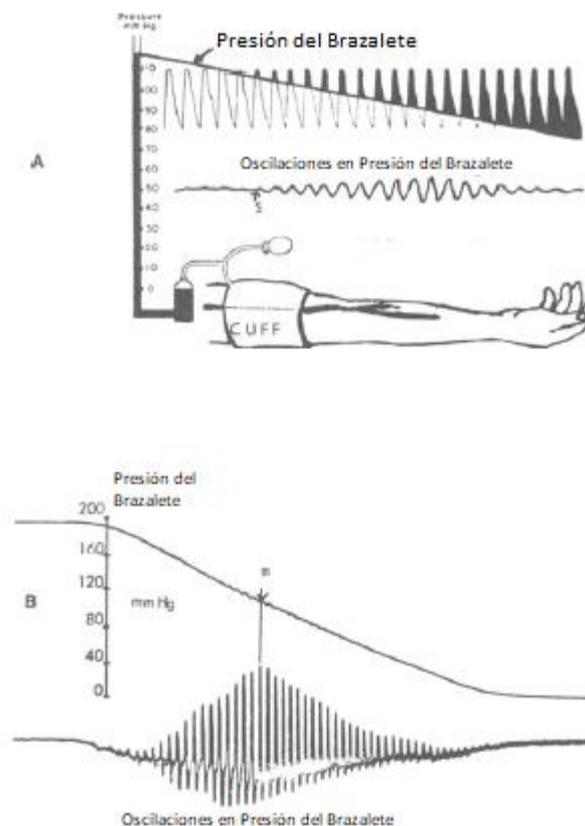
### El Método Auscultatorio.

Ampliamente difundido y aceptado por la técnica clínica, este método fue detallado por Korotkoff en 1905. Se dispone un brazalete constrictor a nivel braquial, ligeramente por encima de la articulación humero-cubital. Se posiciona el transductor de un estetoscopio por debajo del brazalete, sobre la fosa antero-cubital, por donde pasa la arteria radial. El brazalete se presuriza por encima de la presión sistólica, y se descomprime gradualmente. Cuando la presión sistólica arterial comienza a superar la del brazalete, pequeños volúmenes de sangre atraviesan la arteria radial ocluida, con un flujo turbulento. La turbulencia produce un sonido característico en el instante del latido cardíaco, este sonido (cuyas características se modifican a medida que disminuye la presión del brazalete), es escuchado por el usuario a través del estetoscopio. El conjunto de sonidos producidos por la sangre al atravesar la arteria parcialmente ocluida se conoce como “Sonidos de Korotkoff”, comienzan a producirse cuando el brazalete alcanza la presión sistólica, y desaparecen cuando este se despresuriza por debajo de la presión diastólica [3].

Este método es simple, repetible y de bajo costo, sin embargo, presenta múltiples desventajas, entre las que se destacan:

- Necesidad de un usuario entrenado con buena audición.
- Requiere un entorno silencioso.
- Demora entre 15 segundos y un minuto para realizarse.
- Presenta dificultades para estimar la presión diastólica [3].

### El Método Oscilométrico:



**Ilustración 8: El método Oscilométrico (A) y la presión del brazalete. Oscilaciones del brazalete amplificadas (B) indicando el punto de máxima amplitud [11]**

Originalmente introducido por Maney en 1876, este método es utilizado en la actualidad para monitorear la presión arterial braquial a lo largo de un período prolongado de tiempo, ya sea en un centro de salud o de forma domiciliaria, o portátil. Consiste en la utilización de un brazalete presurizador similar al utilizado en el método Auscultatorio. Este se infla con una bomba por encima de la presión sistólica y se despresuriza gradualmente. A medida que la presión arterial supera la del brazalete, el flujo sanguíneo que genera los sonidos de Korotkoff transmite una vibración pulsátil hacia el brazalete, ya que este se encuentra solidario a la arteria, a través de los tejidos superficiales comprimidos. Estas vibraciones producidas por la transmisión del pulso de presión a través de los tejidos hacia el brazalete, son detectadas por el sensor de presión, comienzan a acrecentarse ligeramente por encima de la presión sistólica y desaparecen ligeramente por debajo de la presión diastólica. Se ha comprobado que el punto de mayor amplitud de las oscilaciones, es cuando la presión del brazalete es igual a la Presión Arterial Media [11].

Debido a que no existen puntos definidos que permitan identificar las presiones sistólica o diastólica, estas deben ser estimadas o aproximadas. Existen múltiples formas de obtener resultados con exactitud aceptable, aún bajo condiciones de hipotensión [12].

Debido a la simplicidad del método y su posibilidad de automatización, en la actualidad se producen dispositivos capaces de realizar mediciones automáticamente y de forma periódica. Su mayor desventaja es la imposibilidad de realizar una medición precisa si el paciente se encuentra temblando, agitado o en vibración.

Es importante destacar que este método es gravemente afectado por las dimensiones del brazalete. El punto de máxima oscilación asociado con la presión arterial media varía ampliamente con la capacidad volumétrica del brazalete ocluidor utilizado [13].

### El Método Pletismo-Oscilométrico.

A partir del principio de funcionamiento del método oscilométrico surgen múltiples técnicas que aprovechan la relación entre la oclusión de una arteria mediante un brazalete neumático constrictor y el flujo de sangre que atraviesa dicha arteria. Si en lugar de medir las vibraciones que se transmiten desde el tejido hacia el brazalete, medimos o estimamos el volumen relativo o el flujo relativo del tejido, aguas debajo de la oclusión, podremos realizar mediciones de presión sistólica, diastólica y media con gran precisión.

Una de las formas para medir este flujo de sangre es mediante la aplicación del efecto Doppler. Varios investigadores han realizado mediciones utilizando este método, con un transductor ultrasónico aguas abajo del brazalete ocluidor dispuesto a nivel braquial [14]. Esta técnica solo permite medir la presión sistólica, sin embargo, si se dispone el transductor de forma perpendicular a la pared arterial y se mide la velocidad de desplazamiento de la misma, si se puede obtener una medición adecuada de presiones media y diastólica [15].

Otra técnica derivada del método oscilométrico consiste en la utilización de un foto-pletismógrafo. Dicho dispositivo consiste de un emisor luminoso y un transductor receptor (típicamente un led IR y un fotodiodo apropiado), dispuestos de forma tal que permitan detectar la variación adimensional del volumen del tejido ocasionada por el flujo de sangre a través de los vasos sanguíneos. La disposición del fotoplestismógrafo puede ser para detección por transmitancia (emisor y receptor en oposición con el tejido entre ellos) o reflectancia (ambos orientados hacia el tejido, posicionados de forma contigua).

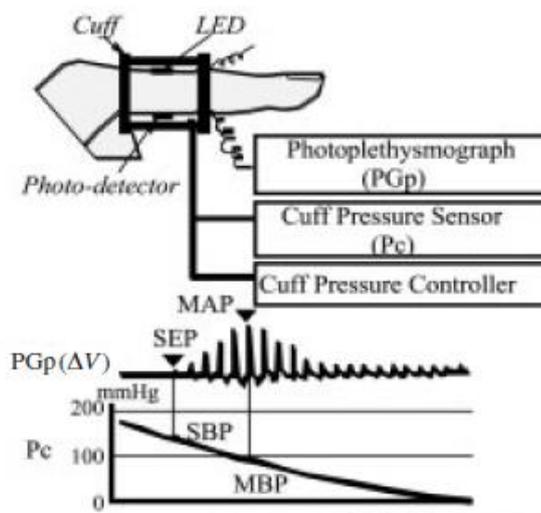


Ilustración 9: Principio de funcionamiento del método Pletismo-Oscilométrico presentado por Yamakoshi [17]

Este método fue presentado por Yamakoshi *et al.* (1982) [16] tomando la señal fotoplethysmográfica de las arterias de la falange. En dicha publicación se obtuvieron los siguientes resultados.

Como podemos ver en la Ilustración 9, el flujo sanguíneo comenzó a detectarse a partir de que la presión del brazalete oclusor descendió por debajo de la presión sistólica, pero las oscilaciones detectadas permanecieron aun cuando esta presión alcanzaba los 0 mmHg. Por lo tanto, esta técnica no permite medir la presión diastólica, esta debe ser calculada a partir de la presión sistólica (SBP) y la media (MBP). Podemos obtener la presión arterial diastólica utilizando la Ecuación 2 **Error! Reference source not found.** [17].

Las presiones obtenidas con este método correlacionaron correctamente con las obtenidas mediante medición directa de la presión braquial (invasiva), aunque sus valores resultaron 5 – 10 mmHg. inferiores. Este método de medición es mas preciso y menos variable que el método oscilométrico convencional [16].

$$DBP = MBP - \frac{\alpha_v}{1 - \alpha_v} (SBP - MBP)$$

**Ecuación 2: Estimación de la presión diastólica en el método foto-oscilométrico [17]. En donde:** DBP es la presión arterial diastólica, SBP es la presión sistólica, MBP es la presión arterial media y  $\alpha_v$  es un factor volumétrico obtenido a partir de la forma de las pulsaciones volumétricas.

Otros métodos han sido desarrollados para la medición de la presión arterial utilizando el principio pletismo-oscilométrico, como, por ejemplo, mediante la utilización de un pletismógrafo por impedanciometría. La impedancia entre dos puntos opuestos de una extremidad varía al modificarse su volumen, consecuencia del paso de sangre. Este principio ha sido utilizado para medir la presión arterial periférica [18],

## Métodos Continuos No Invasivos.

Es importante destacar que existen dos tipos de técnicas de medición denominadas como continuas. Por un lado, aquellas que miden las presiones sistólicas, diastólicas y/o medias, en cada latido. Este tipo de técnicas obtiene mediciones de presión arterial latido-a-latido, pero nada informan respecto a la forma de la curva de presión arterial. Los métodos de medición continua reales o instantáneos, obtienen la presión instantánea a lo largo de un periodo de tiempo, de modo tal que es posible graficar la curva de presión, además de su envolvente latido-a-latido.

### Métodos con la Velocidad de Onda de Pulso (PWV).

Todos los métodos que hemos analizado hasta ahora cuentan con una gran desventaja, todos utilizan un brazalete ocluser-constrictor que restringe la circulación sanguínea en el segmento vascular utilizado para la medición. Esto presenta problemas de diseño, ya que se debe calibrar e incorporar un sistema neumático adecuado, y de comodidad. Los pacientes pueden experimentar desde incomodidad leve hasta dolor y entumecimiento, debido a la utilización del brazalete, además de que algunos de estos métodos requieren un posicionamiento adecuado del mismo, en consecuencia, la aplicación práctica de estas técnicas resulta compleja.

Se han desarrollado modelos matemáticos para representar las arterias como vasos cilíndricos de características conocidas (Módulo de Elasticidad, diámetro, longitud). Si se estima la viscosidad de la sangre y medidos la Velocidad de la Onda de Presión (PWV: Pulse Wave Velocity), como el largo del segmento dividido por la diferencia de tiempo entre el arribo de la onda de presión a cada extremo del segmento, podemos calcular la diferencia de presión sistólica diastólica con la ecuación de Bramwell-Hills y Moens-Kortweg [19].

$$PWV = \frac{L}{\text{Time Delay}} = \sqrt{\frac{hE_0 e^{\alpha P}}{\rho d}}$$

**Ecuación 3: Cálculo de la Velocidad de la onda de presión arterial [19]. En donde:  $E_0$  es el módulo de Young de la arteria para presión arterial nula,  $\alpha$  es el número de Euler,  $h$  es el espesor de la pared arterial,  $d$  es el diámetro de la arteria y  $\rho$  es la viscosidad de la sangre.**

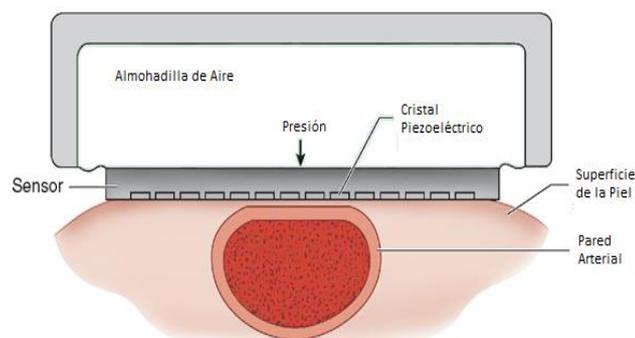
Si se parametriza adecuadamente las variables involucradas y se mide con precisión el lapso de tiempo y la longitud de la arteria, se puede medir de forma continua la presión central instantánea. Si bien existen múltiples publicaciones con este método [19, 20, 21, 22], aún requiere de mucha investigación para poder ser incorporado como un método práctico. Algunas variantes de este método utilizan la variable PAT (pulse arrival time), como la diferencia de tiempo entre la onda R del ECG y la pendiente sistólica del pulso de presión. Estos se basan en múltiples modelos matemáticos para estimar la PWV a partir del PAT, pero estos modelos también requieren de parametrizaciones adecuadas para su

correcta implementación. Se han desarrollado dispositivos que utilizan esta técnica de medición, y hasta algunos de los más recientes Relojes Inteligentes (SmartWatch) incorporan esta técnica. Sin embargo, es importante destacar que los ensayos demuestran que los dispositivos deben ser “entrenados” para cada usuario y para cada postura o posición [20].

### Tonometría Aplanatoria.

Presentado originalmente por Pressman & Newgard en 1963 es un método que permite la medición continua de presión arterial sobre cualquier segmento arterial superficial con suficiente soporte posterior óseo, como la arteria radial. Utiliza un transductor miniatura que se dispone sobre la arteria, a nivel de la piel. Los tejidos interpuestos transmiten la presión entre la arteria y el transductor.

El transductor se posiciona correctamente cuando la amplitud de las pulsaciones medidas es máxima. Seguidamente, el transductor se presiona contra la arteria (por ej., con una almohadilla de aire), con una presión tal que la arteria se aplane en su sección superficial. Cuando la arteria se aplane contra el transductor, y si este es rígido y de dimensiones adecuadas, la presión arterial se transmite hacia el mismo de forma continua y unitaria. Los transductores se componen con arreglos de cristales piezoeléctricos o piezo-resistivos. Se han comercializado dispositivos que utilizan esta técnica de medición desde 1992 (por Colin Electronics Co. Ltd., Japón) [23].<sup>2</sup>



**Ilustración 10: Esquema de la técnica de medición de presión arterial por tonometría aplanatoria. Se puede observar que solo algunos de los cristales del arreglo son utilizados para la medición.**<sup>2</sup>

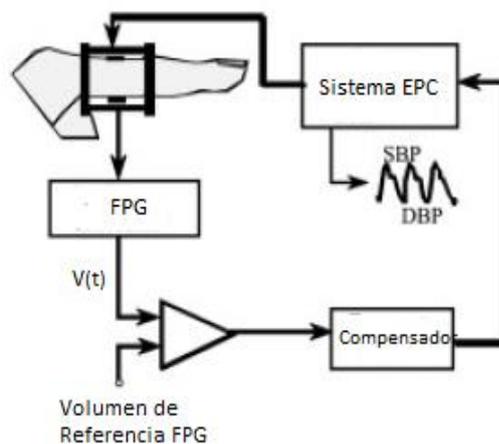
Equipos de medición que utilizan esta técnica han sido aprobados y validados múltiples veces, y constituyen el “gold standard” de medición continua de presión arterial no invasiva, ya que se han demostrado resultados aceptables aún en situaciones complejas [24]. Sin embargo, este método presenta desventajas prácticas que impiden su implementación de forma masiva, como son:

<sup>2</sup> Imagen extraída de [https://openi.nlm.nih.gov/detailedresult.php?img=PMC3409361\\_CHR-8-80\\_F1&req=4](https://openi.nlm.nih.gov/detailedresult.php?img=PMC3409361_CHR-8-80_F1&req=4)

- El posicionamiento del transductor es de importancia crítica. El usuario requiere una capacitación adecuada para determinar apropiadamente su correcto posicionamiento.
- Se debe evitar o restringir el movimiento del paciente, ya que este puede provocar el desplazamiento del transductor y/o la oclusión de su arteria.

### Métodos de Descarga o Compensación Volumétrica.

Este método fue concebido a mediados del siglo pasado y fue implementado por primera vez por J. Peñáz en 1973 [25, 26]. Está basado en la compensación volumétrica de la pared vascular. Se dispone un brazalete ocluidor alrededor de una extremidad adecuada (típicamente, la falange anular), y se controla la presión del mismo. Si la presión media dentro del brazalete  $\bar{P}_c$ , es igual a la presión intra-arterial media  $\bar{P}_a$ , de modo tal que la presión transmural ( $\bar{P}_{tr} = \bar{P}_a - \bar{P}_c$ ) es igual a cero, la arteria se encuentra en un estado “descargado” (‘unloaded state’). Cuando se ejerce una variación de presión  $\Delta P$  sobre la presión arterial  $\bar{P}_a$ , el volumen vascular cambiará de forma correspondiente  $\Delta V$ . Esta variación de volumen puede ser compensada aplicando la misma variación de presión  $\Delta P$  sobre  $\bar{P}_c$ , para mantener un volumen constante en el estado descargado. En este estado, la presión del brazalete ( $P_c = \bar{P}_c \pm \Delta P$ ), es igual a la presión instantánea intra-arterial  $\bar{P}_a$  [17].



**Ilustración 11: Esquema del sistema de control del método de compensación vascular [35].**

Para la implementación de este método se requiere de un sistema de control servo-volumétrico (EPC: Electro Pneumatic Control). Se establece una señal volumétrica de referencia, la cual es obtenida mediante un fotopletismógrafo implementado junto con el brazalete ocluidor. El volumen de referencia se determina

en el punto medio de la curva fotopletismográfica (de ahora en más, FPG), cuando el brazalete se presuriza a la presión arterial media.

Seguidamente, la variación de volumen genera una variación de la señal FPG que es utilizada como señal Error. La señal error controla la apertura y cierre del sistema de control neumático, de modo tal que la compensación volumétrica se realiza de forma automática. Para la selección del volumen de referencia se puede utilizar el método de medición pletismo-oscilométrico [27].

Es importante destacar algunas desventajas del método implementado por Peñaz en 1973. Como se utilizó un brazalete ocluser alrededor de la falange, y como la presión en el mismo es siempre igual a la presión intra-arterial, no existe flujo sanguíneo a través de la extremidad, aguas debajo de la oclusión. Esto genera dos grandes problemas, por un lado, la sensación de incomodidad y hasta dolor que el paciente experimenta, y por otro, la acumulación venosa de sangre. No se conoce el efecto que la acumulación venosa produce en la medición, a través de períodos prolongados de tiempo.

### Dispositivos que utilizan la Compensación Volumétrica.

#### Dispositivos Finapres.

El dispositivo para medición de presión arterial no invasivo más utilizado en la actualidad que utiliza el principio de descarga vascular de Peñaz. Este equipo fue desarrollado por Wesseling *et al* [28], y es comercializado actualmente por FMC (*Finapres Medical Systems, Netherlands*). Este utiliza un diseño muy similar al originalmente utilizado por Peñaz, pero con algunas mejoras:

- Facilita el proceso de selección del Volumen de Referencia.
- Mejora el sistema de Control Neumático con una frecuencia de trabajo de al menos 30 Hz [29].
- Incorpora un sistema de auto calibración del set-point de referencia, periódico y regular.

De estas, la mayor ventaja presentada es la calibración automática del punto de referencia de volumen. La Ilustración 12 muestra la secuencia de inicio calibración empleada por el Finapres. Se utiliza el método Oscilométrico para determinar el volumen de referencia inicial. Una vez que se ha iniciado la técnica de compensación, y a intervalos regulares, la presión del brazalete ocluser se estabiliza temporalmente, primero en la presión media, luego ligeramente por

arriba de la misma y finalmente, ligeramente por debajo. Las oscilaciones del FPG para estos tres valores se detectan y se determina su amplitud, si la oscilación correspondiente a la presión media es inferior a alguna de las otras dos, se ajusta el volumen de referencia de forma correspondiente, de forma tal que el reajuste corrija los cambios en el volumen del tejido que se pudieran haber introducido por artefactos externos [30, 29].

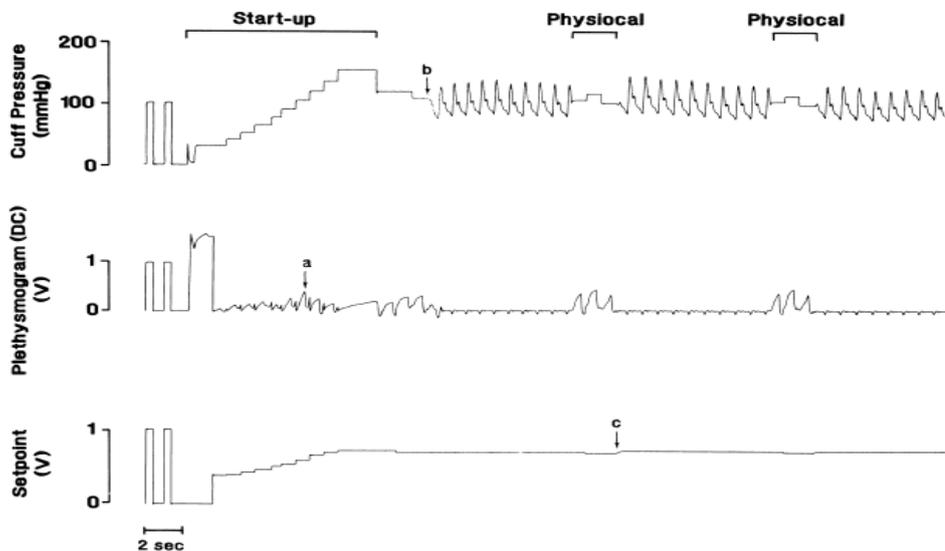


Ilustración 12: Secuencia de Start-up y de re-calibración del Finapre [29],

Es importante apreciar que, si bien el sistema Finapres presenta grandes ventajas con respecto al método original y se ha comercializado mucho más que sus competidores, aún cuenta con la gran desventaja de utilizar un brazalete ocluidor completo. Por lo que el paciente experimenta gran incomodidad luego de su utilización por períodos prolongados de tiempo.

El sistema Finapres ha sido evaluado en numerosos ensayos y ha sido comparado con múltiples otros sistemas [29, 30, 31]. De todos estos ensayos se concluye que permite medir la presión arterial en casi todos los pacientes, inclusive en pacientes con vaso-espasticidad o con aterosclerosis severa [32]. Pero se destacan algunas recomendaciones que deben ser tenidas en cuenta.

Como la vasoconstricción periférica afecta la usabilidad del dispositivo, se debe evitar utilizar en pacientes que tengan sensación de frío, y se debe mantener la temperatura ambiente por encima de los 22 °C. Se recomienda mantener caliente la mano y la utilización de los dedos medio o anular. Se debe prestar especial atención al posicionamiento del brazalete y este no debe ser movido para mantener mediciones precisas [29].

Otros estudios han evaluado el Finapres contra Tonometría Radial [30] y han sugerido que pueden haber ciertos problemas con los principios de funcionamiento del primero. Entre los que se destacan que:

- Es posible que la presión del brazalete no sea transmitida uniformemente hacia el tejido, de modo que la presión efectiva ejercida sobre la arteria puede diferir con respecto a la presión del brazalete.
- Es posible que la presión a la que se obtiene la máxima amplitud de la señal FPG no sea igual a la presión arterial media.
- El dispositivo asume que el único volumen de sangre debajo del brazalete se encuentra en las arterias, debido a que las venas y capilares se encuentran ocluidos. Sin embargo, existe un pequeño flujo sanguíneo por las arterias que eleva gradualmente la presión venosa. Eventualmente esta presión podría comenzar a afectar las mediciones de formas desconocidas.

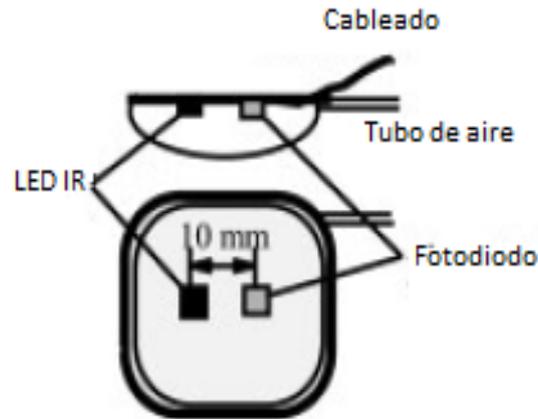
En otros estudios, se comparó el Finapres contra métodos de medición invasivos en pacientes con hipotensión profunda (durante un procedimiento quirúrgico para melanóma) [31]. En este estudio se concluyó que el dispositivo estimaba con precisión aceptable la presión sistólica, pero que sobreestimaba las presiones diastólica y media, con aproximaciones inaceptables. Aunque es importante destacar que aún la presión sistólica presentó errores porcentuales de entre 5 y 20% en 50% de las mediciones.

Es importante notar que, en la mayoría de los estudios se analizó únicamente la correlación entre SBP, MBP y DBP de las técnicas comparadas. Pero en ninguno de los ensayos clínicos que hemos encontrado se observa la concordancia entre las curvas de presión.

### Dispositivos con Almohadilla Oclusora.

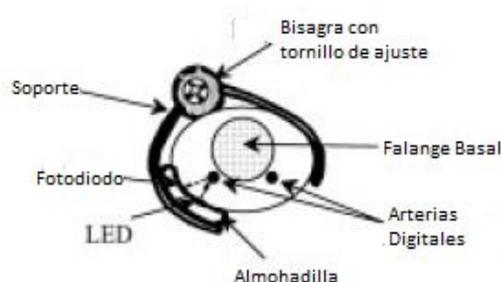
Otro grupo de investigadores, liderados por Ken-Ichi Yamakoshi, desarrolló múltiples dispositivos y ensayos clínicos investigando los diferentes métodos de medición de presión arterial continua no invasiva (NIBP). Desde 1980 se han realizado un gran número de estudios con equipos basados en el diseño de J. Peñaz [27, 33, 34].

Este mismo grupo de investigación japonés desarrolló un nuevo dispositivo para la técnica de compensación volumétrica que presentó una gran ventaja con respecto al diseño de Peñaz y el Finapres. En lugar de ocluir todos los vasos de una extremidad por la utilización de un brazalete constrictor, se diseñó una almohadilla neumática localizada, cuya función consiste en ocluir únicamente la arteria de interés. Este diseño fue implementado originalmente para la arteria radial, ya que esta cuenta con un soporte óseo posterior ideal para que la almohadilla sea capaz de provocar su oclusión, pero luego se adaptó el diseño para su utilización en las falanges del adulto [17].



**Ilustración 13: Almohadilla neumática desarrollada por Yamakoshi, et al, para medición en arteria radial con método de compensación volumétrica [17]**

La Ilustración 13: Almohadilla neumática desarrollada por Yamakoshi, et al, para medición en arteria radial con método de compensación volumétrica. La Ilustración 13 muestra el diseño original implementado por el grupo de investigación de Yamakoshi, *et al*, para la medición en la arteria radial. El fotopleetismógrafo de reflectancia se encuentra incorporado dentro de la almohadilla oclusora, la cual se sella herméticamente. La superficie de la almohadilla que entra en contacto con la piel está formada por una lámina delgada de goma elástica y transparente, que permite el traspaso de la iluminación infrarroja del fotopleetismógrafo. Para esta lámina, se han utilizado diferentes materiales, como membranas de silicona elástica y uretano transparentes [17, 35].

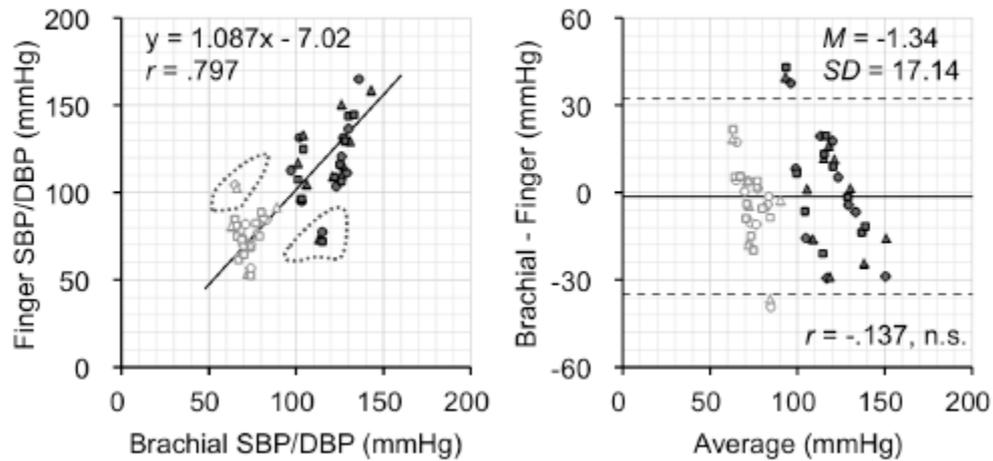


**Ilustración 14: Posicionamiento y fijación de almohadilla digital, por Yamakoshi et al [17]**

La almohadilla se adaptó para su utilización en la falange basal de la mano, diseñándola como un disco de 20 milímetros de diámetro, posicionado lateralmente a la palma del dedo, de modo tal que su centro coincida con la posición de la arteria digital colateral interna de la falange basal, como demuestra la Ilustración 14. La principal ventaja de este diseño radica en la capacidad vascular remanente del tejido. Como la arteria digital externa permanece libre, así como las venas del lado externo, el tejido permanece irrigado aguas abajo. En consecuencia, no se produce acumulación o elevación de la presión venosa y el paciente no experimenta mayor incomodidad, dolor o entumecimiento. [17] Esta aseveración fue demostrada realizando mediciones en la falange con el método de compensación vascular utilizando el brazalete ocluser y la almohadilla. En ambos casos se utilizó un fotopleletismógrafo posicionado en la sección distal del dedo correspondiente, y se midió el volumen durante el procedimiento. Este volumen se normalizó para ambas mediciones de 0 a 100%, expresando a 0% como el nivel lumínico detectado sin oclusión alguna, y a 100% como el nivel lumínico cuando un brazalete ocluser restringía la circulación de forma total en la sección distal del mismo dedo. La prueba demostró que la congestión sanguínea excedía el 100% para el caso del brazalete ocluser de Peñaz, pero que la almohadilla disco ocasionaba una congestión que no superó un 35% a lo largo de una hora de medición continua [17, 36].

El sistema de medición con almohadilla neumática ha sido evaluado en más de una publicación, inclusive se ha implementado un método de auto-calibración igual al empleado por el Finapres, para el reajuste del volumen de referencia en el método de compensación volumétrica (bautizado "AVCM: Advanced Volume Compensation Method"). Se ha demostrado su precisión comparándolo contra medición braquial por oscilometría convencional, durante un período de ejercicio. La Ilustración 15 muestra los resultados obtenidos por este estudio, se puede observar la enorme mejora que el proceso de auto-calibración implica para el resultado de las mediciones subsecuentes [35].

## VCM



## AVCM

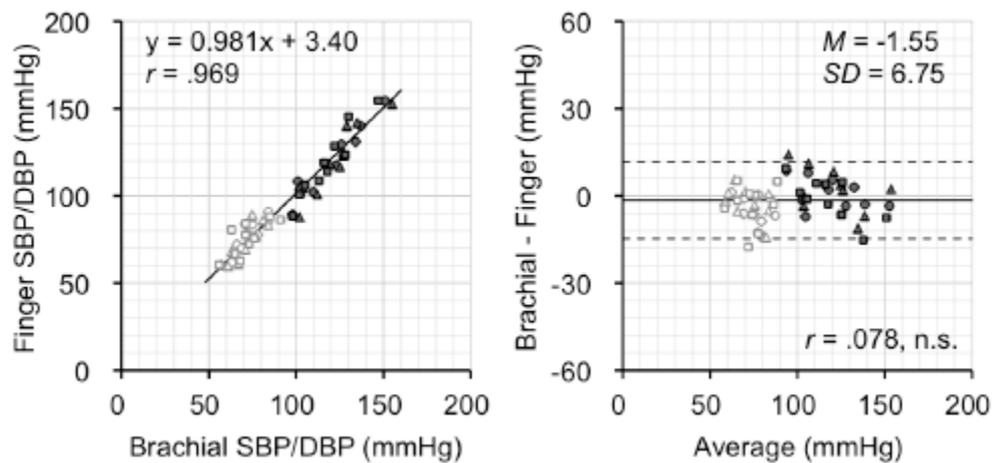
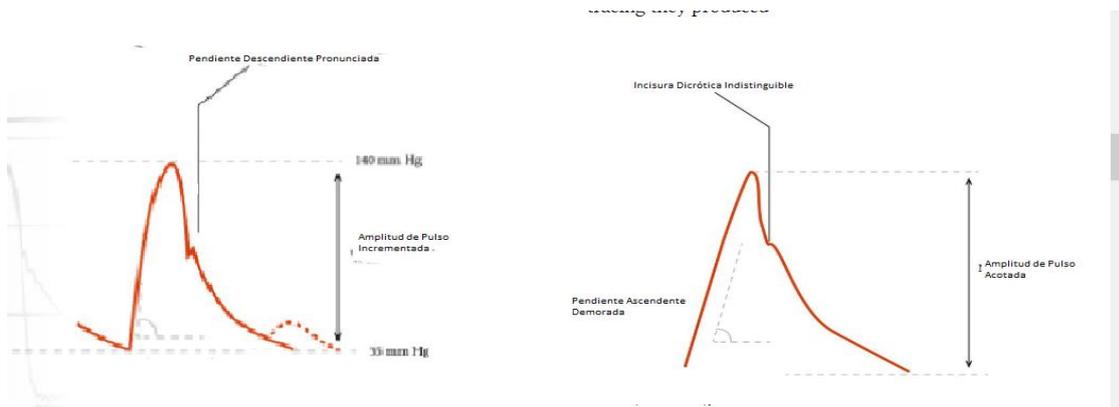


Ilustración 15: Comparación de las mediciones obtenidas durante un estudio de ejercitación por ciclismo. Se muestran las correlaciones entre mediciones de presión Sistólica y Diastólica entre las obtenidas mediante oscilometría convencional y los dispositivos de compensación volumétrica propuestos por Yamakoshi et al. Arriba: Método convencional sin calibración automática de volumen de referencia. Abajo: Método mejorado con calibración regular de referencia [35]

### Importancia de la medición de la PA instantánea.

En la actualidad, la medición continua no invasiva de la presión arterial es utilizada escasamente en el ámbito de la Salud, particularmente en nuestro país. Durante el desarrollo de este proyecto integrador, no se encontraron equipos de medición no invasiva continua de presión arterial, en los principales centros de salud de la ciudad de Córdoba. Se pueden suponer múltiples causas para dicha ausencia, desde la reducida oferta de equipos con la capacidad de realizar estas mediciones, la dificultad de su utilización, entre otros. Existen múltiples estudios demostrando características distintivas de la forma de onda de presión en individuos con patologías cardiovasculares. En la Ilustración 16 se observan formas de onda características en pacientes con Soplo Aórtico y Estenosis Valvular

Aórtica. En el primer caso se observa una pendiente descendente post sistólica pronunciada, provocada por el retorno de sangre hacia el ventrículo que provoca la caída de presión acelerada [37]. En el segundo caso, la estenosis valvular provoca un incremento en la resistencia periférica para la bomba cardíaca, en el primer segmento del circuito circulatorio, lo que provoca una pendiente ascendente inclinada y una amplitud inferior [38]. Por lo que podemos encontrar grandes beneficios diagnósticos para la utilización de tecnologías de medición de la presión arterial instantánea de forma no invasiva.



**Ilustración 16: Formas de onda de presión arterial características de Soplo Aórtico (izquierda) y Estenosis Valvular Aórtica (derecha) [2, 37]**

## Capítulo 2: Proceso de Diseño.

Para establecer un proceso de diseño adecuado, lo primero que se debió hacer fue diagramar claramente los elementos del sistema de medición y los diferentes componentes necesarios para el prototipo. En la Ilustración 17 se observa un diagrama de bloques funcional del sistema de medición. A diferencia de los prototipos desarrollados por *Yamakoshi* [17, 27] y *Wesseling* [28], que controlaban la presión de oclusión mediante la apertura de una única válvula presurizadora (de tipo *nozzle flapper*, lamina oscilante), en este proyecto se decidió utilizar dos electroválvulas de conmutación a alta frecuencia.

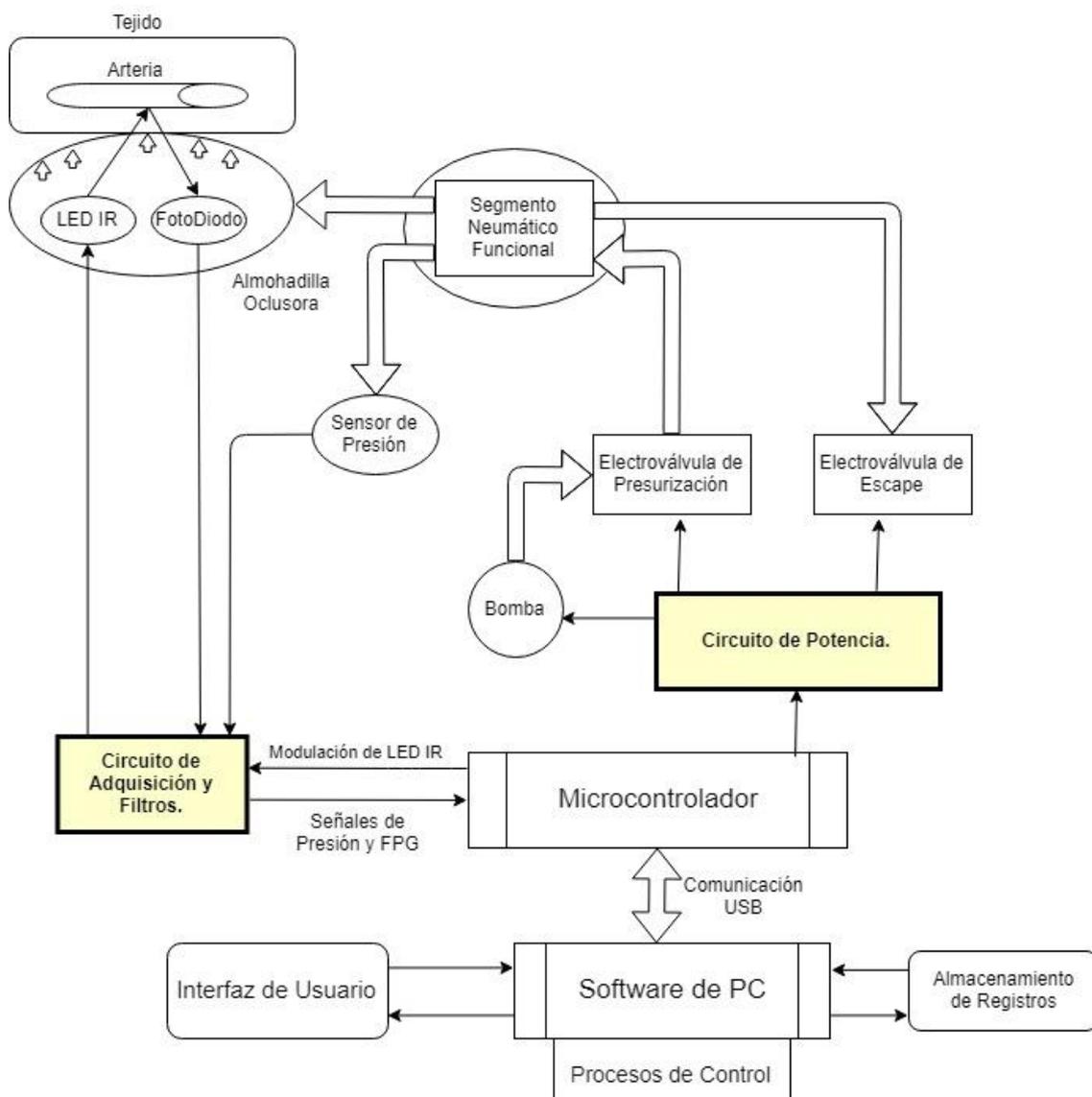
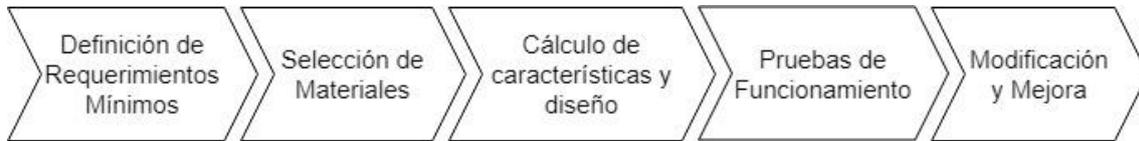


Ilustración 17: Diagrama de bloques del prototipo.

Una vez habiendo establecido los componentes del sistema de medición, se procedió con el proceso de diseño de cada uno de estos componentes, de forma individual, y siguiendo la siguiente secuencia de pasos para cada uno:



**Ilustración 18: Procedimiento de diseño**

Cada uno de estos pasos constituyó subprocesos diferentes, para cada componente del sistema de medición. Por lo que serán descritos en los capítulos correspondientes.

Una vez construido el prototipo se procedió con la sintonización de controladores y pruebas de funcionamiento. En esta etapa del proceso de diseño se buscó mejorar la calidad de las mediciones, mediante funciones que buscan mejorar la respuesta temporal del sistema de medición y recalibrar los parámetros de funcionamiento de forma automática.

Finalmente, se realizaron pruebas de medición comparándolas con un equipo de Oscilometría.

### Materiales Utilizados.

Para la realización de este prototipo se requirieron los siguientes componentes:

- Sensor de Presión Fujikura FPM-07PG
- Rampa de inyección para GNC Valtek de 5ta Generación.
- Bomba neumática de 5 Vdc.
- Electroválvula Pneutronics SRS.
- Fuente de alimentación de 6 Vdc y 1 A.
- Fuente de alimentación de 12 Vdc y 6 A.
- Mangueras y conectores neumáticos de 4 y 6 mm. de diámetro.
- Fotodiodo PT908 y LED IR928 de Everlight.
- Microcontrolador PIC 18F2550 de Microchip.
- Transistores: IRF3205, BC327.
- Circuitos integrados: MCP6004, OP07, LM7805.
- Resistencias y Capacitores de varios valores.
- Fichas y Cables USB 2.0.
- Látex Líquido.
- Sellador de Silicona.
- Gabinetes de plástico y de Madera.
- Papel Aluminio.

Además, se utilizaron las siguientes herramientas de trabajo:

- Taladro, Minitorno, Pinzas, Destornilladores, Soldador y estaño.
- Programador de PIC clon de PICKit 2.
- Multímetro.
- Osciloscopio.
- Generador de Funciones.
- Impresora 3-D Trideo Printbox Kit.
- Computadora portátil.
- Pistola de silicona.
- Simulador de Signos Vitales Fluke ProSim 8.
- Monitor Multiparamétrico Phillips MP20.

Las herramientas virtuales que se emplearon son:

- SolidWorks 2016.
- MATLAB 2016a y SimuLink.
- Visual Studio 2015 Enterprise.
- Compilador PIC C CCS.
- AutoDesk's EAGLE.
- FilterPRO Desktop.
- Simplify 3D.

## Capítulo 3: Circuitos de Adquisición de Señal.

Las señales necesarias para el correcto funcionamiento de este prototipo son la señal fotopletismográfica y la señal de presión. Estas señales deben ser generadas, amplificadas, acondicionadas y muestreadas, de modo tal que garanticen el funcionamiento deseado. Por ende, el primer paso debió ser la definición apropiada de todos los requerimientos que las señales deberían cumplir.

En primer lugar, se definieron las características de la señal pletismográfica. El transductor generador de la misma es un fotodiodo, estimulado por iluminación de un Diodo LED. El primer requerimiento que surge del principio de funcionamiento del prototipo de *Yamakoshi et al* [17], es que el fotopletismógrafo debía ser incorporado dentro de la almohadilla oclusora y sellado herméticamente. Como se inició el proceso de diseño del prototipo a partir de la implementación de los circuitos de señales, fue necesario tener en consideración las posibles dificultades que pudieran surgir en el futuro, en particular, las dimensiones de los componentes del fotopletismógrafo: el fotodiodo y el LED infrarrojo (más adelante se explica el motivo de la elección de un espectro de emisión infrarrojo para el diodo LED).

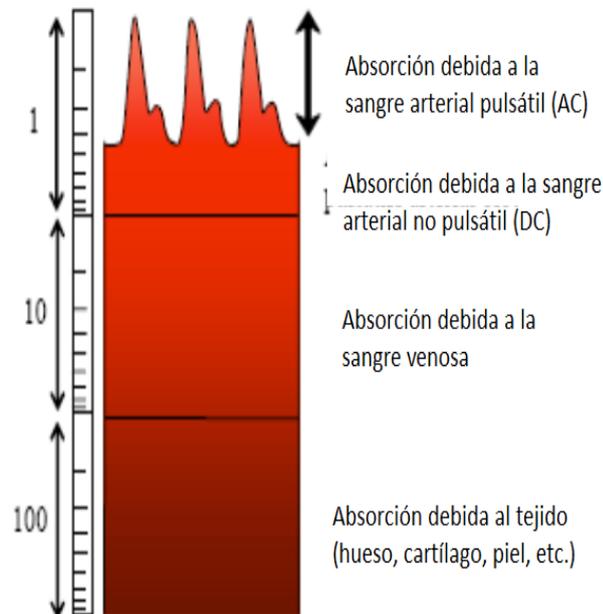
Como estos componentes deben estar incluidos en una almohadilla relativamente pequeña, debimos de diseñar nuestro circuito a partir de los componentes con dimensiones más reducidas posibles. En base a esto, se eligieron los transductores más pequeños disponibles en el mercado:

- Led Infrarrojo Everlight IR928-6C
- Fototransistor Everlight PT908-7C

La señal fotopletismográfica recibida por el fotodiodo, al estar estimulada por la iluminación del LED IR está formada por múltiples componentes. Por un lado, la iluminación ambiente que introduce una componente de corriente continua en el receptor y por otro la corriente generada por la luz del emisor, que atraviesa el tejido y es atenuada por el mismo. Si modulamos la iluminación del LED IR, podemos distinguir estas dos componentes muy claramente, y filtrar una de la otra, ya que la luz ambiente siempre tendrá un espectro constante (corriente continua o de una frecuencia específica, como la de la línea de 50 Hz).

Sin embargo, la señal producida por la iluminación del emisor también está constituida por varias componentes. Está formada por la sumatoria de las atenuaciones provocadas por los diferentes tejidos y la sangre, que atraviesa la luz. De esta atenuación total, aquella producida por la circulación pulsátil de sangre que atraviesa el tejido es la de mayor interés. Ya que es en base a esta componente, que se regula el control de presión del sistema de compensación vascular. Sin embargo, también las demás componentes de la señal presentan importancia, ya que nos brindan información con respecto al desplazamiento de tejido, a la variación de congestión vascular y a la variación del nivel de presión

debida a la respiración del individuo. En consecuencia, es importante que la componente pulsátil de la señal sea lo mas grande posible, pero que permanezcan las componentes estáticas de la señal.



**Ilustración 19: Esquema de la composición de una señal fotopleletismográfica en la punta del dedo humano [39]**

La proporción de señal ocupada por la componente pulsátil depende mayormente, del posicionamiento del emisor y el receptor, con respecto al tejido. En condiciones típicas, esta componente ocupa un 1-5% de la señal fotopleletismográfica total [39]. Otra forma de incrementar la proporción de señal pulsátil, radica en percatarse de que esta componente es generada, principalmente, por sangre oxigenada. En contraste, la sangre venosa es, en su gran mayoría, carboxi-oxigenada. Como la Hemoglobina Oxigenada presenta una absorbancia mayor a longitudes de onda infrarroja (mayores a 800 nm), que la Hemoglobina Desoxigenada [40], si utilizamos un espectro de emisión en este rango, la señal pulsátil se incrementará con respecto a la señal venosa. Si en cambio, se utilizaran espectros de emisión en los que la carboxi-hemoglobina presenta mayor absorbancia que la oxi-hemoglobina, el efecto sería el inverso.

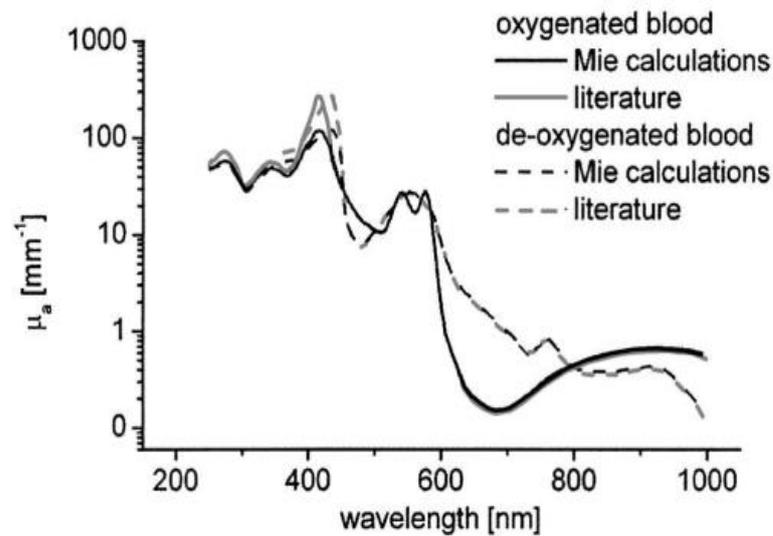


Ilustración 20: Coeficiente de absorción vs Longitud de Onda de la sangre oxigenada y desoxigenada [40]

Debido a todo lo antes mencionado, la integridad de la señal (ausencia de ruido), la modulación y el correcto posicionamiento del transductor, así como la longitud de onda utilizada, son los principales requerimientos de la señal fotopleletismográfica.

Los requerimientos de la señal de presión resultan mucho mas simples de caracterizar. Como se definió como objetivo específico, que el sistema de medición sea de una exactitud superior o comparable a los métodos de medición de presión utilizados convencionalmente, se estableció una tolerancia de error para la medición de presión de  $\pm 5$  mmHg. Consecuentemente, se trató de diseñar el circuito de amplificación y acondicionamiento de la señal de presión con la menor cantidad posible de etapas, a fin de garantizar la integridad de la señal y la reducción del error.

### El Amplificador de Transimpedancia.

La señal óptica recibida por el fotodiodo es una señal de corriente, por lo que debe ser convertida a tensión para poder ser interpretada por el conversor ADC del circuito lógico. Se decidió utilizar la configuración propuesta por Yamakoshi, *et al* [17, 35], el amplificador de transimpedancia. Esta configuración utiliza la corriente establecida por el fotodiodo, alimentada a la entrada inversora de un amplificador operacional y con una resistencia de realimentación de valor elevado, para generar una señal de tensión.

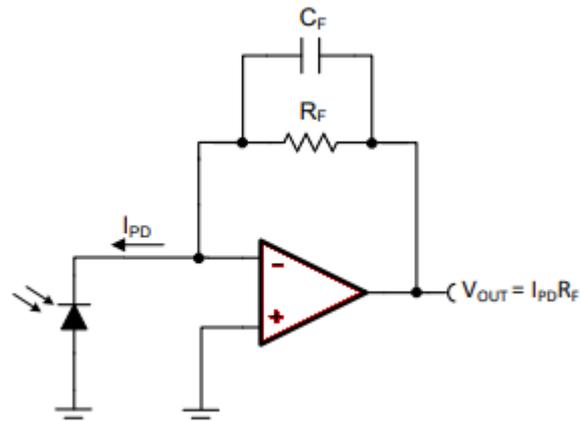


Ilustración 21: El Amplificador de Transimpedancia [41]

La salida de este amplificador responde a la siguiente ecuación:

$$V_{out} = I_{pd} \cdot R_f$$

**Ecuación 4: Ganancia del amplificador de transimpedancia.** En donde:  $I_{pd}$  es la corriente inversa del fotodiodo y  $R_f$  es la resistencia de realimentación.

Esta configuración de amplificador requiere de un capacitor de retroalimentación  $C_f$  para conseguir la estabilidad de la salida. Este capacitor se implementa para compensar la capacidad parásita presente en la entrada inversora del amplificador operacional. Esta capacidad parásita es el producto de la combinación entre la capacidad de junta del fotodiodo, la capacidad parásita entre los conductores de los cables del transductor y las capacidades de modo diferencial y modo común del amplificador operacional.

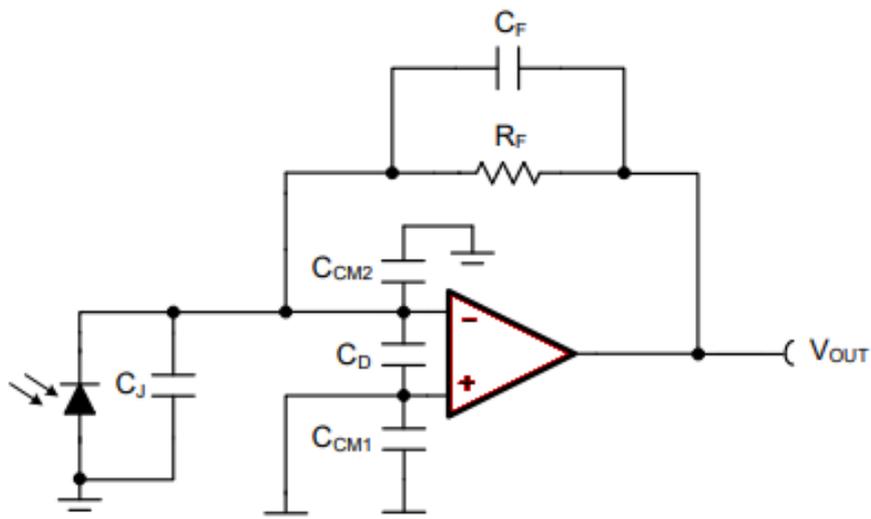


Ilustración 22; Capacidades parásitas en el amplificador de transimpedancia [41]

De la Ilustración 22 podemos observar que la capacidad parásita presente a la entrada inversora del amplificador es:

$$C_{in} = C_j + C_D + C_{CM2}$$

Ecuación 5: Capacitancia de entrada inversora del TIA.

Estos parámetros dependen tanto de los componentes utilizados, como del diseño del circuito impreso y de las dimensiones de los cables del transductor.

Para determinar el valor de nuestro capacitor de realimentación debemos considerar que la malla de realimentación forma un Polo en la respuesta en frecuencia del amplificador. Este polo establece una atenuación en la ganancia para frecuencias mayores a su frecuencia de corte. Por lo que, si conocemos la frecuencia máxima a la que nuestro circuito debe trabajar y la resistencia de ganancia necesaria, podemos determinar el máximo valor que nuestro capacitor puede tener [41].

$$f_p = \frac{1}{2\pi C_f R_f} \quad C_f \leq \frac{1}{2\pi R_f f_p}$$

Ecuación 6: Capacidad máxima de retroalimentación para el amplificador de transimpedancia [41].

Si bien nuestro circuito funciona a una frecuencia de modulación de 1600 Hz, esta modulación se realiza mediante una forma de onda cuadrada, por lo que el espectro presentará componentes de frecuencia en 4800 Hz, 8000 Hz, etc., ya que los armónicos pares de una onda cuadrada ideal son nulos. Las amplitudes de las  $n$  componentes impares de la onda cuadrada serán [42]:

$$A_n = \frac{2}{\pi \cdot n}$$

Ecuación 7: Amplitud de armónicos impares de una onda cuadrada ideal.

Habiendo establecido esto, y considerando que la onda generada por el microcontrolador no es una onda cuadrada ideal, seleccionamos una frecuencia máxima de 24 kHz. Utilizando la Ecuación 6 y considerando una resistencia de realimentación  $R_f$  de 1 M $\Omega$ , se obtuvo una capacidad  $C_f$  máxima de 6.63 pF. Basándonos en publicaciones oficiales de *Texas Instruments* [41, 43], seleccionamos un valor inicial de 2.7 pF.

A partir de este resultado, podemos utilizar la capacidad parásita de la entrada inversora del amplificador, para determinar el ancho de banda de ganancia (*GBW: Gain Band-Width*) necesario del Amplificador Operacional. Supondremos un valor de Capacidad de entrada del circuito integrado igual a 20 pF y una capacidad parásita del fotodiodo igual a 150 pF. A partir de estas suposiciones, obtenemos la GBW necesaria como [41]:

$$f_{GBW} \geq \frac{C_{in} + C_f}{2\pi R_f C_f^2}$$

Ecuación 8: Mínimo ancho de banda de ganancia necesario para el AO.

Con estos valores, se obtuvo una *GBW* necesaria de 3.77 MHz. Como la mayoría de los amplificadores operacionales disponibles cuentan con una *GBW* de 1 MHz, modificamos la resistencia de realimentación  $R_f$  y le asignamos un valor de 220 KΩ.

Otros requerimientos que se establecieron para el amplificador operacional fueron:

- Tensión de alimentación de 6 Vdc.
- Rechazo de Modo Común mayor a 90 dB.
- Capacitancia de entrada menor a 20 pF.
- Circuito integrado tipo Quad (con cuatro amplificadores operacionales por chip), con el fin de reducir la cantidad de componentes del circuito.

Con estos parámetros, y sumando además el requisito de que el amplificador operacional sea de tipo *Rail-to-Rail*, a fin de evitar saturación para la salida en proximidad a la alimentación (ya que nos será de utilidad para su utilización en el circuito de filtración), seleccionamos el circuito integrado **MCP6004**.

### Acondicionamiento de la Señal Fotopletismográfica.

Una de las mayores ventajas que presenta la modulación de la señal fotopletismográfica, es que nuestra señal óptica, de baja frecuencia, se traslada en el espectro de frecuencias. Es decir, si la señal original presentaba una frecuencia fundamental  $f_o$  y la modulamos a una frecuencia  $f_m$ , la frecuencia fundamental resultante de la nueva señal tendrá tres componentes [44]:

- La frecuencia de modulación (o *Carrier*):  $f_m$
- La porción superior de la traslación:  $f_m + f_o$
- La porción inferior de la traslación:  $f_m - f_o$

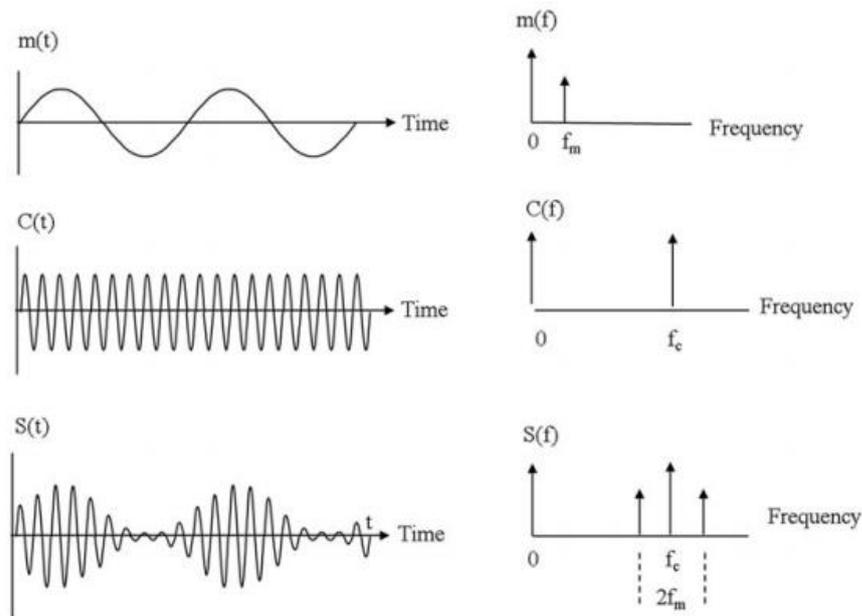


Ilustración 23: Respuesta temporal y en frecuencia de una modulación simple de AM [44]

Esta nueva frecuencia es mucho más simple de aislar que, por ejemplo, una señal con una elevada componente de muy baja frecuencia o de cualquier interferencia electromagnética cuyo espectro sea cercano al espectro original de la señal. En nuestro caso de aplicación, la iluminación artificial del entorno, de frecuencia igual a la frecuencia de línea (50 Hz), ingresa con facilidad al circuito a través del fotodiodo. Al modular nuestra señal, mediante la modulación del led infrarrojo, podemos filtrar las interferencias inferiores a nuestra nueva frecuencia fundamental sin afectar apreciablemente nuestra señal. Se eligió una frecuencia de modulación de 1600 Hz.

De modo que nuestro filtro principal consistirá de un filtro pasa-banda cuya banda de paso debe ser desde 1500 a 1700 Hz, y que debe anular las frecuencias inferiores a 1200 Hz y mayores a 2000 Hz. Para esto, debemos implementar un filtro activo de alto orden, cuyas características principales son [45]:

- Respuesta en frecuencia: Se conoce como la relación entre las amplitudes de las señales de salida y de entrada del filtro, para las distintas frecuencias de la señal. Se busca que la respuesta en frecuencia tenga ganancia unitaria en la banda de paso y atenuaciones elevadas en la banda de corte.
- Respuesta temporal: Como todo circuito, la respuesta temporal del filtro, se describe a partir de los parámetros resultantes de su salida ante una entrada escalón. Estos parámetros son: Tiempo de retraso, tiempo de subida, tiempo de establecimiento y sobre amortiguamiento.

Existen dos características importantes para determinar la configuración de filtro que utilizamos. En primer lugar, se debe elegir la respuesta del filtro y, por otro lado, la configuración del circuito. Para determinar la respuesta del filtro, debemos elegir y definir dos de entre tres variables [46]:

- Ripple de la banda de paso: Diferencia máxima entre la ganancia del sistema entre dos frecuencias cualesquiera de su banda de paso.
- Pendiente de transición de la banda de paso a la banda de corte: Rango de la respuesta en frecuencia del sistema en la que la ganancia se encuentra entre la banda de paso (hasta -3 dB) y la banda de corte (a la que definimos -45 dB).
- Orden del filtro: Se denomina orden del filtro al número de polos de la respuesta en frecuencia del mismo. Los polos son los puntos del plano  $S$  (frecuencia - tiempo), en los que la salida de la función de transferencia del filtro  $H(s)$ , es infinita, es decir, cuando se anula el denominador de la función. Los polos actúan como filtros en los que la señal se atenúa -20 dB/década por encima de la frecuencia de corte. A mayor orden, mayor será la pendiente de la banda de paso a la banda de corte, y más etapas serán necesarias para construir el circuito.

Una vez definidos dos de estos parámetros, podemos elegir la respuesta del filtro que vamos a utilizar. Habiendo establecido los parámetros de Orden de Filtro en 6, y un Ripple de 1 dB, utilizamos el software *FilterPRO Desktop*, para obtener las posibles respuestas de filtro que cumplen con nuestros requisitos. Establecimos además ganancia unitaria y frecuencia central del filtro en 1.6 kHz.

Seguidamente, elegimos la respuesta de filtro tipo Butterworth, ya que presentaba la mayor estabilidad en la banda de paso [46] y, por ende, nos permitió modificar la frecuencia de modulación dentro de la banda de paso, sin necesitar reajustar la ganancia de la etapa de amplificación de señal.

Una vez obtenida la respuesta del filtro y sus principales características, debimos elegir la configuración del circuito. Existen muchos tipos de filtro, pero las configuraciones más utilizadas son los filtros *Sallen Key* y los filtros *Multiple FeedBack (MFB)*. Ambos filtros son de segundo orden por etapa y cuentan con la ventaja de ser utilizados masivamente. El filtro Sallen-Key es menos dependiente que el MFB, ya que este último configura al amplificador operacional como un integrador. Sin embargo, decidimos utilizar la configuración MFB ya que presenta ciertas ventajas [47]:

1. Mayor independencia de las variaciones en los valores de los componentes utilizados.
2. No se encontraron evidencias de filtros Sallen Key Pasa Banda, en configuración de fuente única, a diferencia del filtro MFB que si cuenta con evidencias de uso.

# Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

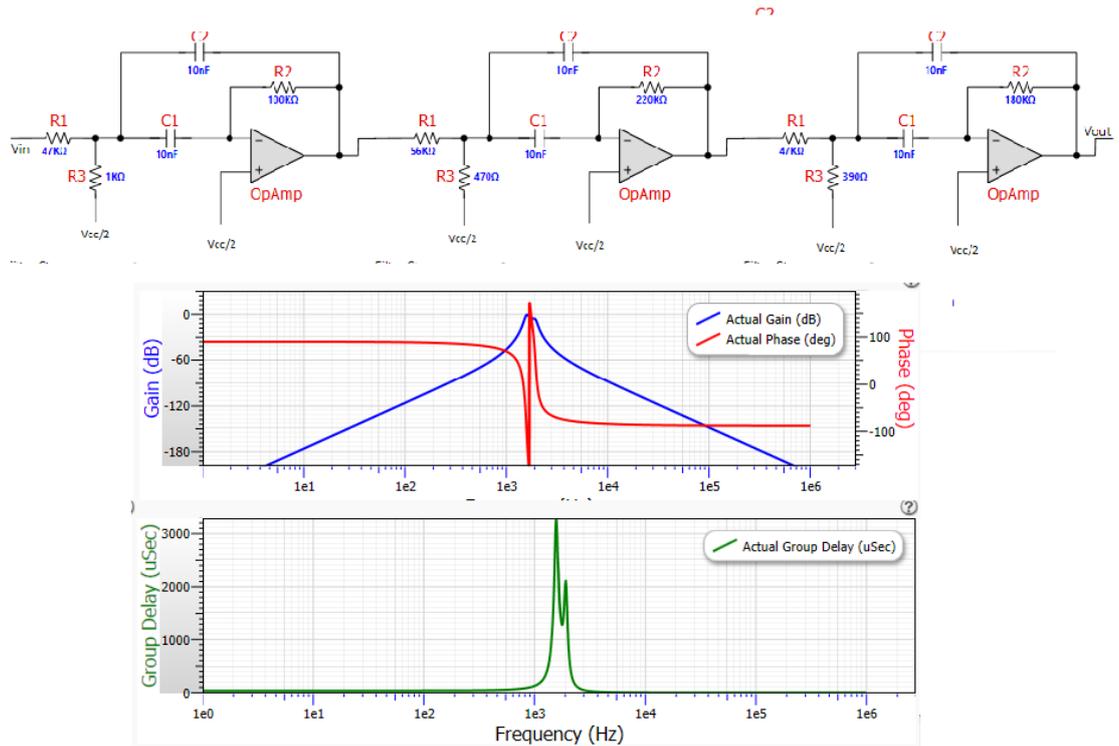


Ilustración 24: Circuito y respuesta en frecuencia del filtro pasa banda de 6to orden. <sup>3</sup>

El software *FilterPRO Desktop* determinó los valores de los componentes, a partir de los requerimientos preestablecidos del filtro y de la Tolerancia de las resistencias y capacitores (utilizamos componentes con tolerancia E12: +10%). El circuito generado por *FilterPRO* se encuentra en configuración de fuente doble, por lo que lo adaptamos según la configuración propuesta por *Texas Instruments* [48] para su utilización con fuente única. <sup>3</sup>

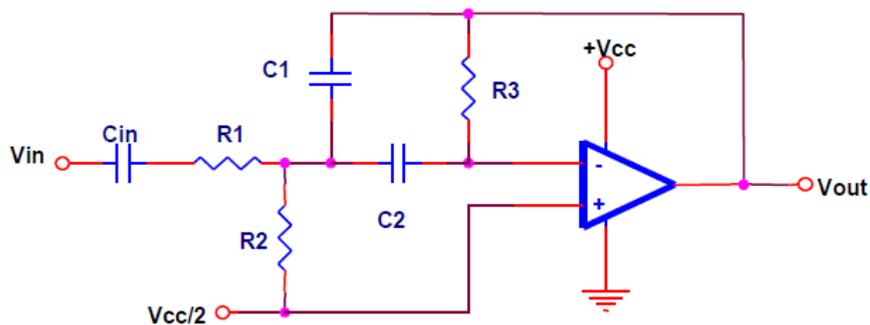
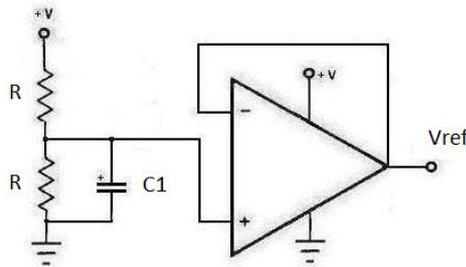


Ilustración 25: Topología de filtro Pasa Banda MFB en configuración de fuente única [48]

<sup>3</sup> Circuito y gráficas generados con *FilterPRO Desktop*.

En este caso, se estableció la referencia del circuito  $V_{ref} = V_{cc}/2$ , con un divisor resistivo de la alimentación del circuito y un amplificador operacional en configuración de *Buffer* [45].



**Ilustración 26: Circuito de Buffer para tensión de referencia de los filtros [45].**

### Amplificación y Acondicionamiento de Señal de Presión.

El factor determinante las características principales del circuito para la señal de presión, es el transductor mismo. Este sensor debe presentar un error bajo las condiciones normales de trabajo, inferior a 5 mmHg, su rango debe incluir el rango de presiones arteriales fisiológicas (30 a 220 mmHg.) y debe medir presión manométrica. Además, debe presentar un tiempo de respuesta pequeño (despreciable para el espectro de la señal de presión) y su salida debe poder ser amplificada en reducidas etapas.

Se nos donó un sensor de presión manométrica FPM-07PG de Fujikura (reemplazable por Panasonic ADP1121 y Amphenol Advanced Sensors SP20C-G501U), el cual cuenta con las siguiente características:

- Principio de funcionamiento: Strain-Gauge.
- Corriente de Polarización (DC): 1.5 mA.
- Resistencia del puente de Wheatstone: 4 a 6 k $\Omega$ .
- Rango de Presión: +- 48.26 kPa / +- 362 mmHg.
- Precisión: Linealidad +-0.3 %FS e Histéresis +- 0.2 %FS.
- Rango de Tensión de Salida: 60 a 140 mV.
- Tensión de Offset de Salida: +- 20 mV (a 0 kPa).

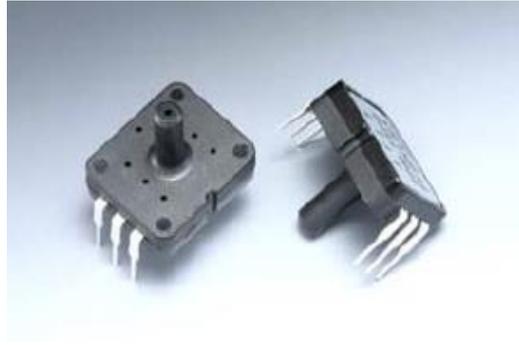


Ilustración 27: Sensor FPM-07PG.<sup>4</sup>

La sensibilidad térmica de la tensión de offset de salida y la del rango de salida son de  $\pm 5$  y  $2.5 \%FS/0-50$  °C respectivamente. Si consideramos estos cambios, la desviación del sensor puede exceder los requerimientos de tolerancia de error, por lo que consideraremos que la temperatura se debe mantener dentro de un rango de  $\pm 5$  °C de la temperatura de calibración durante los procesos de medición.<sup>4</sup>

Con estos parámetros, podemos calcular la exactitud del sensor como:

$$\begin{aligned} \text{error} &= \pm \%FS \times \sqrt{e_{lin}^2 + e_{his}^2 + e_{temp}^2} \\ \text{error} &= \pm \frac{362 \text{ mmHg}}{100} \times \sqrt{0.3^2 + 0.2^2 + 0.25^2} \\ \text{error} &= \pm 1.59 \text{ mmHg} \end{aligned}$$

Ecuación 9: Cálculo de la exactitud del sensor de presión.

Por lo que la exactitud de este sensor se encuentra dentro de los requerimientos preestablecidos, siempre y cuando se controle la temperatura ambiente dentro de ciertos parámetros, lo cual no debería ser un problema adicional dentro de un centro de salud, pero si puede llegar a ser una limitación importante en uso domiciliario.

Como la corriente de Polarización es de 1.5 mA y la resistencia es de un máximo de 6000  $\Omega$ , se obtiene que el puente de Wheatstone del sensor de presión requiere una tensión de alimentación de 6 Vdc.

La hoja de datos del sensor no especifica una sensibilidad de medición, sino que nos da un rango posible de tensión de salida de 60 a 140 mV, pero tampoco nos aclara los factores que influyen en esta enorme variación del rango de salida. En consecuencia, debimos suponer que esta variación se debe a los posibles valores de resistencia del puente de Wheatstone, y a las variaciones

<sup>4</sup> Imagen extraída de la hoja de datos: [http://www.fujikura.co.jp/eng/re-source/pdf/fpm\\_e.pdf](http://www.fujikura.co.jp/eng/re-source/pdf/fpm_e.pdf)

consideradas por el fabricante para la corriente de polarización. Teniendo en consideración estos parámetros, determinamos el rango de ganancia que el circuito de acondicionamiento debe poder generar para obtener la señal de tensión necesaria a la entrada del conversor analógico-digital. Utilizamos el ADC integrado en nuestro microcontrolador PIC, por lo que la tensión de referencia superior debe ser de 5 Vdc. Además, se estableció una referencia para la ganancia del circuito de señal de presión  $V_{ref} = V_{cc}/2$ , con el fin de utilizar la misma tensión de referencia para el circuito de acondicionamiento de señal de presión y fotepletismográfica.

En consecuencia, calculamos la ganancia del circuito de presión como:

$$Gain = \frac{V_{out}}{V_{in}} = \frac{5V - \frac{6V}{2}}{V_{in}}$$

$$V_{in} = V_{pr} \pm V_{offset}$$

$$V_{pr-min} = 60 \text{ mV} \left( \frac{250 \text{ mmHg}}{362 \text{ mmHg}} \right) = 41.43 \text{ mV}$$

$$V_{pr-max} = 140 \text{ mV} \left( \frac{250 \text{ mmHg}}{362 \text{ mmHg}} \right) = 96.68 \text{ mV}$$

$$V_{in-min} = V_{pr-min} - V_{offset} = 21.43 \text{ mV}$$

$$V_{in-max} = V_{pr-max} + V_{offset} = 116.68 \text{ mV}$$

$$Gain_{min} = \frac{2V}{116.68 \text{ mV}} = 17.14 \quad Gain_{max} = \frac{2V}{21.43 \text{ mV}} = 93.41$$

**Ecuación 10: Cálculo del rango de ganancia necesario para las etapas de amplificación de presión.**

Por lo que, el circuito de acondicionamiento debe permitir un rango de ganancia total de 17 a 94. Para implementar esta variable, se utilizó un circuito de 3 etapas:

- 1) Un amplificador para el puente de Wheatstone del sensor, constituido por un AO, en configuración diferencial, con una ganancia fija  $A_v = 9$ .
- 2) Un Filtro Pasa-Bajo Sallen-Key de fuente única, con frecuencia de corte en 500 Hz y ganancia unitaria, para eliminar ruido de alta frecuencia.
- 3) Un amplificador final, con un AO en configuración no inversora, y con una ganancia regulable de 1 a 5.

Para la primera etapa de amplificación se utilizó un AO en configuración diferenciadora de fuente única, tal como se muestra en la Ilustración 28. En la que se estableció:

- $R1 \parallel R2 = R3 \parallel R4$
- $V_o = G \times (V_{in2} - V_{in1})$
- $G = R2 / R1$
- En consecuencia, se eligieron:  $R2 = R4 = 33 \text{ k}\Omega$  y  $R1 = R3 = 1.8 \text{ k}\Omega$ .

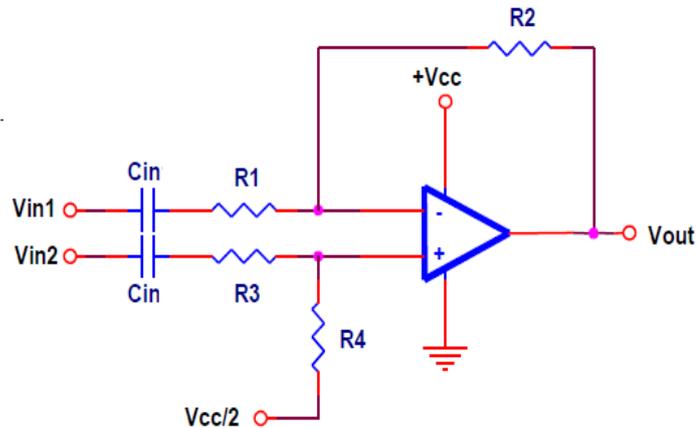


Ilustración 28: Amplificador Operacional en configuración Diferenciadora [48]

Es importante destacar que, para garantizar un correcto funcionamiento de esta etapa de amplificación sin que se observe saturación de la señal final de entrada, se debió considerar que los amplificadores operacionales introducen un offset de tensión a sus entradas. Como no consideramos este offset en los cálculos de ganancia, y para que dicho offset no reduzca el rango útil de amplificación, se decidió utilizar un amplificador operacional de Bajo Nivel de Offset (*low offset voltage Op. Amp.*), para esta primera etapa de amplificación. Se seleccionó un Amplificador Operacional **OP07**, cuyo voltaje máximo de offset es de  $75 \mu\text{V}$ .

El circuito del filtro se obtuvo con el software *FilterPRO Desktop*, se eligió una respuesta de tipo *Butterworth* y una configuración Sallen-Key. Para esta etapa de filtración. Para el último amplificador en configuración no inversora, se utilizaron amplificadores remanentes de uno de los chips MCP6004 utilizados para la amplificación de la señal fotopletoisomográfica.

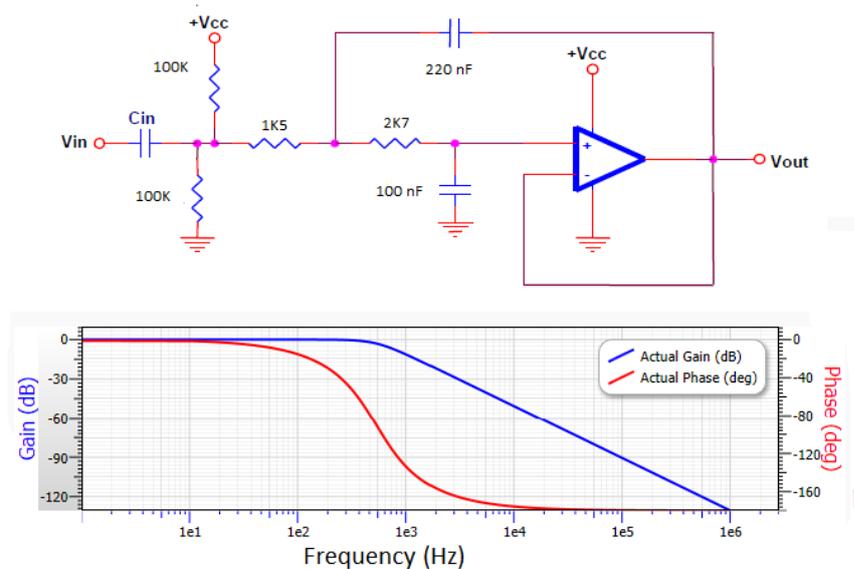


Ilustración 29: Circuito del Filtro Pasa Bajo y Respuesta en frecuencia.<sup>5</sup>

5

### Diseño y Construcción del Circuito:

Los circuitos de señales de presión y fotopleletismográfica se probaron en *Proto Board* lo máximo posible y se implementaron en una única PCB (*printed circuit board*) simple faz, diseñada con el software *AutoDesk's EAGLE*. La placa se alimenta con una fuente de alimentación única de 6 Vdc. Debido a los requisitos de integridad de las señales que se debían cumplir, esta placa de circuito impresa fue producida por una empresa profesional de circuitos impresos (CI-TEM SRL), en placa de fibra de vidrio con perforaciones de soldadura platinadas y Mascara Anti-Soldante de protección.

El sensor de presión se incorporó al circuito impreso, pero el fotodiodo y el diodo LED debieron ser dispuestos de forma remota para su correcta utilización. Para esto se utilizó cable USB tipo A de cuatro filamentos, en un extremo se soldaron los componentes ópticos y en el otro se dejó la ficha USB macho para facilitar su conexión con el circuito mediante una ficha hembra fija al circuito impreso. El blindaje del cable se dejó desconectado en el extremo distal.

Mucho después de su construcción, a lo largo de múltiples pruebas de calibración y mediciones, se llegó a concluir que el circuito del amplificador de transimpedancia era inestable. Se observaban, de forma aleatoria, oscilaciones y acoplamientos modulados de la señal fotopleletismográfica, como muestra la Ilustración 30. Para solucionar este problema, fue necesario incrementar el valor del capacitor de realimentación  $C_f$  del amplificador de transimpedancia, con el

<sup>5</sup> Circuito y Gráfica generados con *FilterPRO Desktop*.

fin de mejorar la estabilidad del amplificador. Agregando un capacitor de 22 pF, dicha inestabilidad se volvió despreciable.

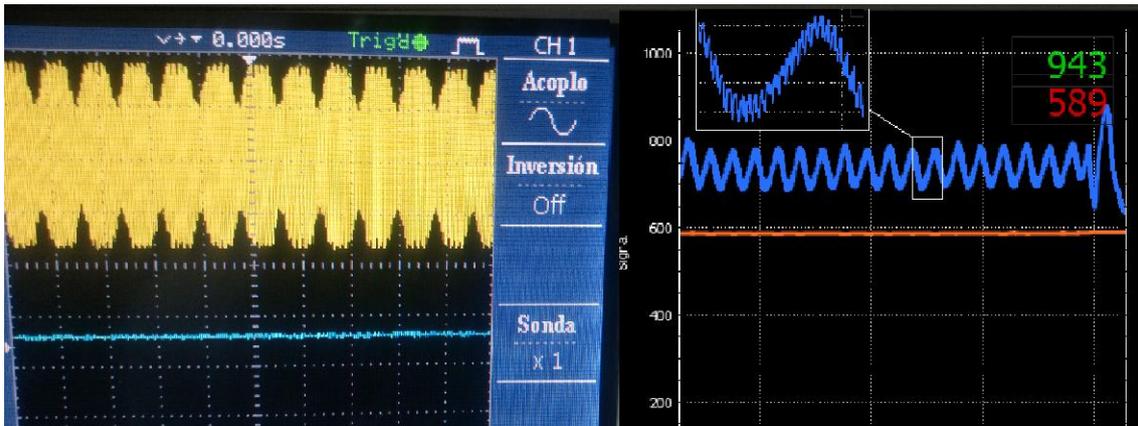


Ilustración 30: Ejemplo de señal oscilante causada por la inestabilidad del amplificador de transimpedancia. Señal observada después del filtro con osciloscopio digital (izquierda) y señal observada en el software de PC luego del muestreo (derecha).

## Capítulo 4: El Sistema Neumático.

Si bien la adquisición y el procesamiento de las señales necesarias es una función de primordial importancia para el correcto funcionamiento de nuestro prototipo, tal vez más aún lo son, la estabilidad y armonía de funcionamiento del sistema neumático. Los principales componentes constituyentes de este sistema son:

- 1) La almohadilla oclusora, que contiene los componentes del fotopleletismógrafo y realiza la oclusión de las arterias utilizadas, junto con las mangueras y conectores utilizados.
- 2) La bomba neumática, responsable de generar la diferencia de presión y el caudal necesario para generar la presurización instantánea necesaria, en cada proceso de control involucrado en la medición de la presión arterial.
- 3) El controlador de presión, las electroválvulas que manejan la conmutación del segmento neumático de la almohadilla con la presurización de la bomba y/o con el escape atmosférico.
- 4) El circuito electrónico de potencia responsable de energizar y regular el ciclo de trabajo de la bomba y las electroválvulas.

Estos componentes del sistema deben funcionar en perfecta armonía. Si fuera a existir una fuga de aire en alguno de estos segmentos, un componente cuyo sobrecalentamiento afecte su *performance* o un *delay* elevado o desconsiderado en su respuesta temporal, todo el sistema podría ser afectado de modo tal, que esta causa podría llegar a ser muy difícil de detectar. En consecuencia, se debió revisar y catalogar todas las variables y posibles factores que afecten a cada elemento, individualmente, antes de iniciar a trabajar el sistema neumático en conjunto.

La cavidad del sistema neumático formada por el interior de la almohadilla oclusora, las mangueras que la comunican con las terminales de las electroválvulas, los conectores y con el sensor de presión y cualquier otra cavidad que este en permanente comunicación con esta misma, constituyen un volumen constante y de gran importancia. Si consideramos que no existe una resistencia considerable a la circulación de aire entre dos puntos cualesquiera de esta cavidad, podemos suponer que la presión dentro de todos los puntos de este volumen es, en cualquier momento, la misma. Como esta presión es la responsable del proceso de Compensación Vascolar, denominamos a esta cavidad o volumen, como el **segmento o circuito neumático funcional**.

El volumen de contención de este circuito es de vital importancia, ya que la presión en dicho volumen depende de sus dimensiones y del caudal neto que experimenta esta cavidad. A partir de la Ley de los Gases Ideales, y suponiendo que el volumen del circuito funcional  $V_0$  permanece constante, para generar una

variación  $\Delta P$  de presión en el mismo, en un período de tiempo  $\Delta t$ , se requiere de un caudal neto medio  $Q$  determinado por:

$$P_o \cdot V_o = n_o RT \quad P_f \cdot V_o = n_f RT$$
$$(P_f - P_o) \cdot V_o = (n_f - n_o) RT$$
$$\Delta P \cdot V_o = Q \Delta t RT \quad \frac{\Delta P}{\Delta t} = \frac{Q \cdot RT}{V_o}$$

**Ecuación 11: Relación entre la variación de presión, el caudal y el volumen de una cavidad.**

En donde  $R$  es la constante universal de los gases ideales,  $T$  es la temperatura en  $^{\circ}K$ ,  $n_o$  y  $n_f$  son los Moles de Gas inicial y final,  $\Delta t$  es el tiempo en segundos y  $Q$  es el caudal en *Moles/seg*.

Podemos observar que el caudal necesario para generar una pendiente de presión determinada  $dP/dt$ , es directamente proporcional al volumen  $V_o$  de la cavidad. En consecuencia, para que el sistema tenga una respuesta temporal rápida y se necesite de un caudal menor (lo cual implica menores dimensiones de la bomba y de las mangueras necesarias), es importante reducir al mínimo el volumen del circuito funcional.

### La Almohadilla Oclusora y el Segmento Neumático Funcional.

Como las únicas publicaciones que utilizan el método de Compensación Volumétrica con una almohadilla oclusora diferente al brazalete propuesto por Peñaz, son las presentadas por *Yamakoshi, et al* [17, 27, 35], los principales requerimientos estructurales y funcionales para este componente se originan de estas mismas publicaciones.

La superficie de la almohadilla que se posiciona sobre la piel debe ser translúcida para la frecuencia de onda infrarroja utilizada y debe ser muy elástica. En base a estos requerimientos, y a partir de la disponibilidad de diferentes materiales del mercado, se determinó que esta superficie debía ser de Látex, para la cual se utilizó Látex líquido.

El diseño ideal para esta almohadilla es el utilizado por *Yamakoshi, et al* [17], en la que se utilizan fotodiodo y fototransistor planos, de sensibilidad lateral, y en la que los cables de dichos componentes salen de la almohadilla de forma lateral, junto con la línea de aire. Inicialmente, se ensamblaron varias almohadillas con este formato, de dimensiones y con algunas características diferentes. Sin embargo, estas almohadillas presentaban ciertos problemas poco intuitivos para la señal fotopletoislográfica.

La goma de Látex presenta un coeficiente dieléctrico de 24.0 [49], por lo que este material consiste un dieléctrico de capacidades considerables. Como los cables del fotopleletismógrafo deben atravesar este material por un espesor considerable (de modo tal que la almohadilla sea hermética), la capacitancia parásita del transductor se incrementa considerablemente. Bajo estas condiciones, la capacidad de respuesta del amplificador de transimpedancia y la integridad de la señal fotopleletismográfica resultan degradadas considerablemente. En consecuencia, no fue posible construir una almohadilla con esta configuración que pueda realizar las mediciones necesarias. Para esto, se necesitaría una goma de poliuretano o silicona, que presente una constante dieléctrica menor.

De modo tal que se implementó una modificación de diseño. En lugar de retirar los cables de forma lateral, se diseñó un disco pequeño de 24 mm. de diámetro, con perforaciones de 5 mm. de diámetro. Este diseño se dibujó en *SolidWorks* y se imprimió con una impresora 3-D de FDM, en ácido poliláctico (PLA). Se utilizaron un fotodiodo y un LED IR genéricos, de encapsulado circular de 5 mm. y se los introdujeron en estas perforaciones. <sup>6</sup>



**Ilustración 31: Disco de PLA de soporte para el fotopleletismógrafo y la almohadilla.<sup>6</sup>**

Alrededor de este disco, por encima del terminal emisor-receptor del par fotodiodo-LED, se dispuso una capa elástica de Látex de 1 mm. de espesor y se la selló sobre la periferia del disco con cemento de contacto y sellador de silicona, de modo tal que no quedara Látex remanente entre los cables del fotopleletismógrafo. Una tercera perforación de 3 mm. de diámetro se agregó al disco para insertar un conector macho-macho donde se conecta la manguera neumática que presuriza la almohadilla.

Una vez se desarrolló y se probó la almohadilla oclusora, se diseñó una estructura destinada a mantener dicha almohadilla en la posición adecuada durante la medición. Para esto, se diseñó en *SolidWorks* el modelo de una

---

<sup>6</sup> Modelo generado con *SolidWorks* 2016.

abrazadera articulada donde se posiciona la almohadilla, por un lado, y por el otro el brazo articulado, que presiona contra el extremo opuesto de la falange utilizada para la medición. La estructura de posicionamiento se articula por una bisagra, que se acomoda de acuerdo con el paciente, y se ajusta con un tornillo y una tuerca mariposa. Este diseño de abrazadera se construyó a partir de impresión 3-D con tecnología FDM, con filamento de PLA de 0.5 mm. de diámetro. El modelo diseñado se procesó con el software *Simplify 3D* y se imprimió con una Impresora 3-D FDM *Trideo Printbox Kit*.



Ilustración 32: Prototipo de la estructura re-tén de la almohadilla.<sup>6</sup>

### La Bomba Neumática.

Se utilizó una bomba neumática de doble membrana, accionada por un motor de corriente continua de 5 Vdc. Este tipo de bomba funciona con desplazamiento alternativo positivo, en la que el aumento de presión se realiza por el empuje de unas membranas elásticas (o diafragmas) que permiten variar el volumen de la cámara de bombeo. Utilizan válvulas anti retorno para controlar la dirección del flujo de aire.

La integridad del encapsulado de este dispositivo resulta de enorme importancia, así como su velocidad de respuesta. Como la respuesta temporal del motor de corriente continua, acoplada a la respuesta temporal de la generación de presión, debida a la inercia del volumen de aire y de las membranas, es relativamente lenta, este motor debe permanecer en funcionamiento, aun cuando la presión del circuito neumático funcional se encuentre por encima de la presión deseada. Ya que, al mantener la presurización y mantener a la bomba en funcionamiento parcial, la respuesta temporal del sistema neumático se agiliza

considerablemente. Para evitar un sobreesfuerzo de la bomba, se regula su tensión de alimentación mediante un control por PWM (*Pulse Width Modulation*).<sup>7</sup>

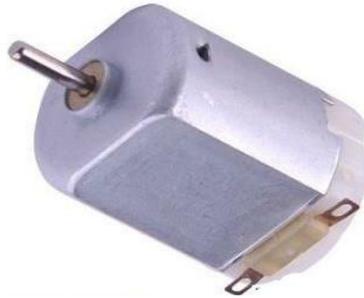


Ilustración 33: Motor de corriente continua de la bomba neumática.<sup>7</sup>

### El Controlador de Presión.

Las publicaciones precedentes de sistemas de medición que utilizan VCM utilizan un controlador electro-neumático (EPC: *Electro Pneumatic Controller*) [17, 26, 27, 29, 34]. Este tipo de controladores funcionan con diferentes principios, desde actuadores piezoeléctricos hasta una aleta metálica oscilante sobre una boquilla presurizada (regulada por un campo magnético variable). Sin embargo, todos estos funcionan con el mismo sistema de control, en el que el proceso de presurización utiliza una señal error analógica y emite una salida en caudal de aire, que luego de generar un cambio de presión se realimenta para establecer el nuevo error del sistema.

Este tipo de controladores es extremadamente simple de implementar, pero presentan enormes dificultades de diseño y manufactura, ya que la hermeticidad o el nivel de fuga del sistema afectan enormemente a la respuesta del sistema. Por ejemplo, el modelo de *Yamakoshi, et al* original [17], mantenía una fuga constante en el circuito neumático funcional cuyo caudal generaba una caída de presión máxima igual o levemente superior, a la pendiente máxima de una curva de presión arterial, y lo único que se controlaba era el flujo de entrada al sistema, mediante un EPC de lámina oscilante. Este sistema de pérdida constante, además de presentar una considerable dificultad de calibración, conlleva una problemática implícita: Como la variación de presión en un determinado sistema depende de la presión original  $P_o$ , de la presión del sistema de escape  $P_e$ , del volumen total del sistema funcional  $V_f$  y de la resistencia neumática de la línea de escape, la respuesta del sistema puede verse afectada por múltiples factores externos y su calibración y mantenimiento pueden llegar a presentar un desafío permanente.

<sup>7</sup> Imagen extraída de <https://www.electronicoscaldas.com/motores-y-servos/270-motor-dc-iman-permanente-escobillas-5v-ppn7pa12c1.html>

Inicialmente, debimos definir los requerimientos mínimos que nuestras electroválvulas debían cumplir para garantizar la utilidad del sistema. Las publicaciones precedentes mencionan que los controladores EPC deben funcionar a una frecuencia de trabajo mínima de 20 a 30 Hz [17, 35]. Teniendo en cuenta que estos tipos de controladores son de apertura proporcional, para obtener un funcionamiento similar con electroválvulas de conmutación ON/OFF, la frecuencia de trabajo debe ser muy superior. Por lo que definimos una frecuencia de trabajo mínima de 300 Hz y un tiempo de conmutación máximo de 0.5 milisegundos. También debimos de considerar que utilizar tensiones de alimentación de corriente continua de valor bajo (hasta 24 Vdc), facilitaría enormemente la implementación del circuito de control.

De los productos disponibles comercialmente en nuestro país, los únicos que se encontraron que funcionan a frecuencias elevadas y con tensiones de alimentación reducidas, son los sistemas de inyección de GNC para el automotor. Por lo que se eligió adaptar y utilizar una Rampa de Inyección de GNC Valtek de 5ta Generación.<sup>8</sup>



**Ilustración 34: Rampa de inyección de GNC Valtek de 5ta Generación.<sup>9</sup>**

Dicha rampa cuenta con las siguientes características:

- Cuatro electroválvulas 2/2 unidireccionales con bobinados de 3.5  $\Omega$  y tensión de alimentación de 12 Vdc.
- Una conexión de alimentación de gas comprimido con un Manifold común.
- Cuatro conexiones de salida.

<sup>8</sup> Imagen extraída de [https://listado.mercadolibre.com.ar/valtek-5ta-gen-gnc#D\[A:valtek%205ta%20gen%20gnc\]](https://listado.mercadolibre.com.ar/valtek-5ta-gen-gnc#D[A:valtek%205ta%20gen%20gnc])

- Ciclo de trabajo máximo del 50% debido al sobrecalentamiento de los devanados.

En consecuencia, se debieron realizar algunas adaptaciones al conjunto de inyección de modo tal que sea útil para nuestra aplicación. Por un lado, la unidireccionalidad de las válvulas presentó un problema importante, ya que la línea de alimentación es común para todas las válvulas. En consecuencia, no se podría utilizar la línea común para la conexión de la bomba, ya que no habría una configuración posible para utilizar una de estas electroválvulas como control de escape. Para resolver este problema se realizaron las siguientes modificaciones:

- 1) Se ocluyó la línea de alimentación de la rampa, ya que su terminal de conexión era de un diámetro muy elevado.
- 2) Se desbloqueó una de las válvulas de modo que quedara permanentemente en apertura.
- 3) Se decidió utilizar la línea común del Manifold como parte del circuito neumático funcional.
- 4) Una de las salidas del Manifold se utilizó como vía de escape, con su electroválvula correspondiente como regulador.
- 5) Otra de las salidas del Manifold, se utilizó en inversión, como conexión de alimentación de aire comprimido de la línea común, regulado por la electroválvula correspondiente.

Para que este último punto funcionara correctamente, se incrementó la fuerza de oclusión de la válvula correspondiente, incrementando las dimensiones del resorte del mecanismo ocluidor. Con esta modificación, la presión inversa, desde la bomba hasta el circuito neumático funcional, no es lo suficientemente grande como para provocar la apertura no deseada de la electroválvula presurización.

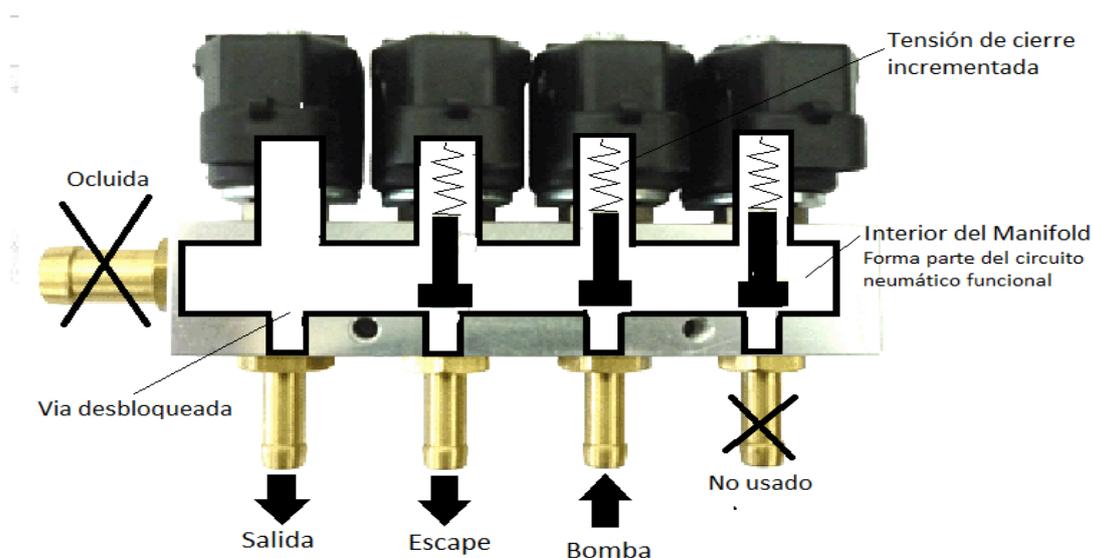


Ilustración 35: Diagrama ejemplificando las modificaciones realizadas a la Rampa de Inyección.

Con estas modificaciones, y regulando la frecuencia de conmutación por PWM de las válvulas, se pudo adaptar la rampa de inyección para que cumpla con todos los requerimientos del sistema de control de presión.

#### Integración del Sistema Neumático.

Se utilizaron mangueras de Polietileno de 4 mm. y 6 mm. de diámetro para las conexiones neumáticas y se utilizó un conector automático tipo Cruz de 6 mm. con sello NBR, de la marca *Intor*, para unir las conexiones neumáticas necesarias.<sup>9</sup>



Ilustración 36: Conector Automático Cruz de 6 mm. con sello NBR, marca *Intor*.<sup>10</sup>

Además de los elementos mencionados hasta ahora, también se utilizó una Electroválvula de Escape 2/2 de la línea *SRS Valve* de la marca *Pneutronics*. Esta válvula se conectó al circuito neumático funcional con el fin de funcionar como válvula de seguridad, que libere la presión contenida en el circuito en caso de cualquier eventualidad y para evitar la utilización del sistema de control para el proceso de calibración de la medición de presión, en el que el circuito funcional debe establecerse a presión atmosférica estable para determinar el offset de la señal emitida por el sensor.

También se dispuso un conector de 2 vías de 4 mm. de la misma marca, con uno de sus terminales hacia afuera del gabinete, con el fin de facilitar la conexión y desconexión del extremo paciente del circuito.

<sup>9</sup> Imagen extraída de [https://listado.mercadolibre.com.ar/intor-cruz-conector#D\[A:intor%20cruz%20conector\]](https://listado.mercadolibre.com.ar/intor-cruz-conector#D[A:intor%20cruz%20conector])

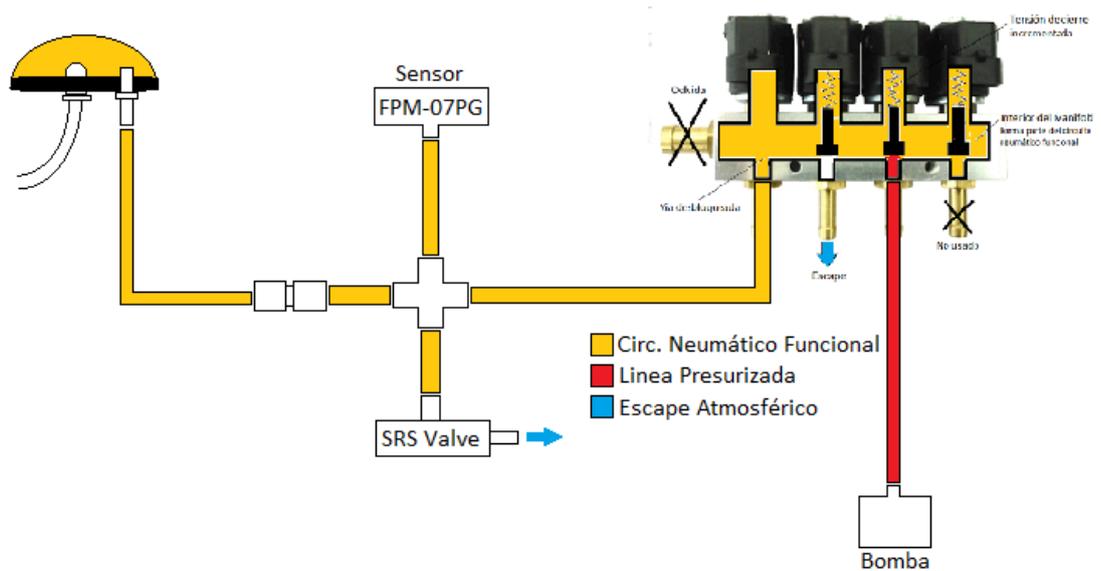


Ilustración 37: Diagrama del circuito neumático del prototipo.

### Circuito de Potencia: Alimentación de Bomba y Válvulas.

Como ya hemos mencionado, el circuito neumático cuenta con cuatro componentes activos, 3 electroválvulas y un motor. La electroválvula SRS de *Pneutronics* consume una potencia de 0.5 Watts y requiere una tensión de alimentación de 5 Vdc. En consecuencia, el consumo de corriente de esta válvula es igual a:

$$I_{SRS} = \frac{0.5 \text{ W}}{5 \text{ V}} = 100 \text{ mA}$$

Ecuación 12: Corriente consumida por electroválvula *Pneutronics* SRS.

Mientras que la bomba utiliza un motor genérico de corriente continua de 5 Volts con un consumo de corriente estable de 150 mA. Por lo que ambos componentes requieren de la misma tensión de alimentación y un consumo relativamente bajo. Sin embargo, estos consumos pueden generar desviaciones del punto de masa en los instantes de conmutación y, por ende, deben ser alimentados por un circuito diferente al de amplificación y muestreo de señales.

Por otro lado, las electroválvulas de 5ta Gen. *Valtek* requieren de una alimentación de 12 Vdc y presentan una resistencia de devanado de 3 a 3.5  $\Omega$ . Por lo que requieren de un consumo de corriente estable de 4 Amperes, si suponemos que en ninguna circunstancia se conmutarán ambas válvulas de forma simultánea. Este consumo elevado también requiere de separación de las masas, y además implica un requerimiento adicional.

Estas corrientes, al ser moduladas, generan una emisión electromagnética considerable. Esta emisión podría llegar a generar interferencias en nuestros otros circuitos, ya que se encuentran próximos entre sí. En consecuencia, para garantizar la integridad de las señales, se debió implementar un blindaje electromagnético entre los circuitos de señales y el resto del sistema. Esto se consiguió con un gabinete pequeño de plástico, cuyo interior se forró con varias capas de papel aluminio, conectadas y que se conectaron a la tierra de protección.

Una vez habiendo establecido los requerimientos, se adquirió una fuente de alimentación de 12 Vdc y 6 Amper, para suministrar la corriente requerida por los componentes activos del sistema neumático.

Como las electroválvulas de GNC requieren de una corriente elevada, su conmutación debe ser controlada por un componente adecuado, un Transistor de potencia MOSFET es el ideal para esta función [50]. Considerando que se necesita un MOSFET para cada válvula, y que estos deben soportar hasta 12 Vdc en corte y hasta 4 A. en saturación, sin provocar una caída de tensión considerable que limite la potencia entregada a las electroválvulas, se eligió el transistor IRF3205 (reemplazable por 55N06). Este transistor presenta una  $R_{DS(ON)}$  de  $8\text{ m}\Omega$ , por lo que la caída de tensión en el transistor y la disipación de potencia en el mismo son iguales a:

$$V_{ON} = 8\text{ m}\Omega \times 4\text{ A} = 16\text{ mV} \quad P_{ON} = 16\text{ mV} \times 4\text{ A} = 64\text{ mW}$$

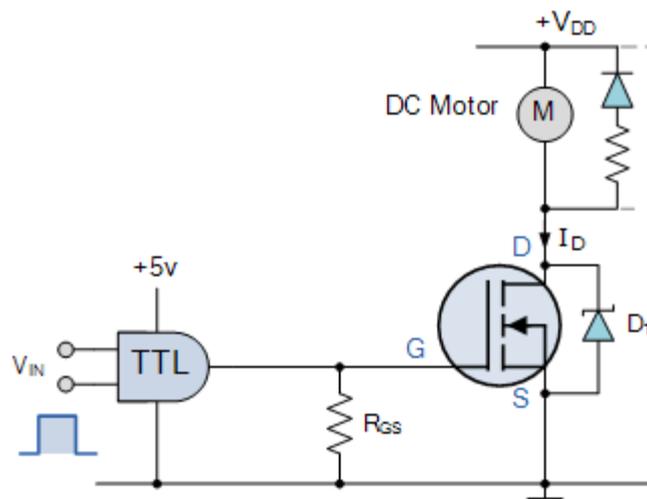


Ilustración 38: Configuración del Transistor MOSFET como Switch [50].

Además, debemos agregar un diodo *flyback* en polarización inversa y en paralelo con cada salida de las válvulas y el motor. Como estas cargas son inductivas, presentan un riesgo elevado para los componentes de conmutación. Debido a la propiedad física de las inductancias, de almacenar energía en forma de campo magnético, cuando el transistor pasa de conducción a corte, esa

energía remanente se debe fugar del bobinado por algún medio. Al implementar un diodo de *flyback*, evitamos que dicha energía dañe el transistor o provoque pequeños arcos de corriente.

Para controlar el transistor MOSFET con el microcontrolador PIC, y aún mantener las masas separadas, se debió utilizar optoacopladores ópticos. Estos componentes transmiten la señal hacia un fototransistor que debe controlar el MOSFET. Para poder generar este control y para generar una lógica mediante la cual, si no existe tensión en el circuito del microcontrolador, las electroválvulas no se energizan, se incorporó además una etapa adicional de conmutación, con un transistor BJT PNP BC327 [51].

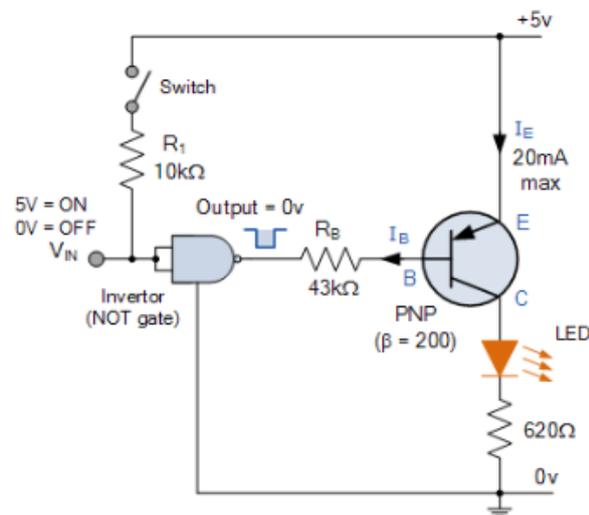


Ilustración 39: Configuración del transistor PNP BJT como Switch [51]

Para la alimentación del motor de DC y de la electroválvula SRS se utilizaron dos circuitos integrados LM7805 (uno por elemento). Estos reguladores de tensión utilizan la fuente de 12 Vdc y generan una tensión estable de salida de 5 Vdc, con una capacidad de corriente de 1 A. Es importante considerar la potencia consumida por el estos integrados cuando los elementos se encuentran en conducción. Esta potencia es:

$$P_{reg} = (V_s - 5 V) \times I = (12 V - 5 V) \times 150 mA$$

$$P_{reg} = 1.05 W$$

Ecuación 13: Potencia disipada por el regulador de tensión 7805.

Este consumo de potencia genera una disipación térmica considerable en los reguladores de tensión, por ende, debemos utilizar disipadores de aluminio para estos componentes. Teniendo en cuenta estos factores y calculando apropiadamente las resistencias de modo tal que los transistores pasen fácilmente

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

de saturación a corte, se diseñaron los circuitos que se observan en la Ilustración 40, y en la Ilustración 41.<sup>10</sup>

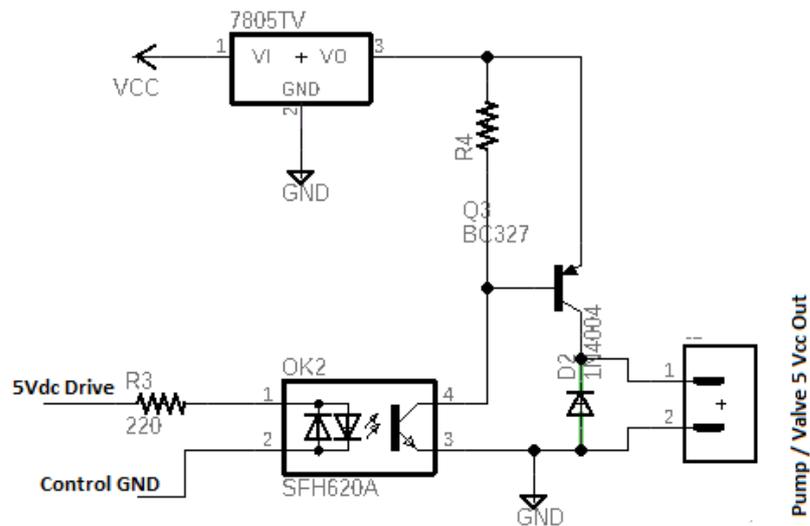


Ilustración 40: Circuito de control de Válvula SRS y bomba neumática de 5 Vdc.<sup>11</sup>

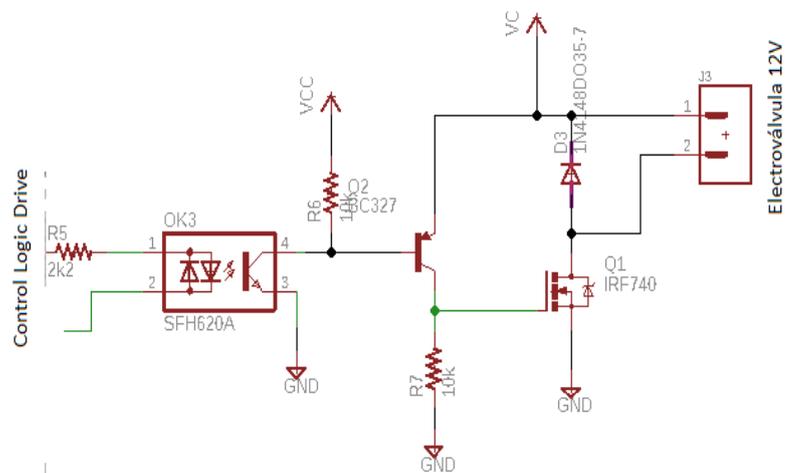


Ilustración 41: Circuito de control de Electroválvulas de 12 Vdc.<sup>11</sup>

<sup>10</sup> Esquemáticos generados con *AutoDesk's EAGLE*.

## Capítulo 5: El Circuito de Control y Comunicación.

Otro componente de gran importancia para el funcionamiento de nuestro dispositivo, es la placa de control y comunicación, así como sus procesos y programación. Si bien resulta muy simple definir los requerimientos que este elemento del sistema debe cumplir, no es tan fácil ni intuitivo implementar dichos requerimientos.

Iniciamos nuestro proceso de diseño describiendo las funciones que el circuito debe poder realizar:

- 1) Comunicación con la PC: Mediante USB para mayor velocidad y masividad de uso del dispositivo. Debe transmitir toda la información pertinente, sin pérdida de datos, y debe procesar todos los comandos generados y enviados por el usuario. Debe resultar en una comunicación simple y eficiente.
- 2) Adquisición de Señales: Se deben muestrear las señales de presión y fotopleletismográfica de forma tal que no exista pérdida de información. Se debe pre-procesar esta información para su transmisión mediante la comunicación USB. Se deben minimizar los artefactos y el ruido mediante procedimientos de muestreo y conversión AD apropiados.
- 3) Modulación de señal lumínica IR: Se debe modular la conducción del LED infrarrojo de modo tal que se garantice un correcto filtrado de la señal fotopleletismográfica.
- 4) Control del Sistema Neumático: Se debe manejar la conmutación de todas las electroválvulas, y la frecuencia y ciclo de trabajo de la bomba neumática. Se debe garantizar un tiempo de respuesta despreciable y una secuencia adecuada de operación independiente de la PC, en caso que la comunicación falle o el cable se desconecte.

Si bien estas funciones ya establecen un desafío considerable, como hemos decidido que la lógica de funcionamiento y los procesos complejos de control se realicen en la PC, es imperativo que el programa de control a nivel circuital sea lo más simple, eficiente y rápido posible. Si llegaran a existir demoras elevadas entre la transmisión de un comando y su ejecución, el sistema podría llegar a presentar un funcionamiento inestable.

### Selección del Microcontrolador.

El primer paso en nuestro proceso de diseño fue determinar los requerimientos específicos del microcontrolador, con el objetivo de elegir el componente adecuado. Dichos requerimientos son los siguientes:

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

- Capacidad de conversión ADC de hasta 2 señales de forma simultánea y estable, con una resolución mínima de 10 bits.
- Módulo de comunicación USB de alta velocidad.
- Capacidad para manejar dos procesos separados, uno de muestreo de alta frecuencia y uno de comunicación y procesamiento de comandos, de baja frecuencia.
- Suministro corriente capaz de polarizar los optoacopladores de las electroválvulas, la bomba y el LED infrarrojo de forma simultánea y con comodidad.

En base a estos requerimientos específicos, se seleccionó el microcontrolador **PIC18F2550**, sobre otros dispositivos PIC o cualquier controlador AtMega, debido a su simplicidad y disponibilidad. Las características de interés de este microcontrolador son:

- Procesador de 8 bits con Memoria de programa de 32 kB, memoria SRAM de 2 kB y EEPROM de 256 Bytes, 24 puertos de entrada/salida.
- Tensión de alimentación de 2 a 5.5 Vdc.
- Módulo USB con capacidad para LOW-SPEED (1.5 Mbps) o FULL-SPEED (12 Mbps).
- Cuatro configuraciones de cristales de oscilación, incluyendo módulo PLL de alta precisión.
- Suministro de corriente máximo de 200 mA (total) y de 25 mA por compuerta lógica.
- Conversor ADC de 10 bits y hasta 13 canales, con tiempo de adquisición programable.
- Cuatro módulos de temporización *TIMER0* a *TIMER3*, tres de ellos de hasta 16 bits.<sup>11</sup>

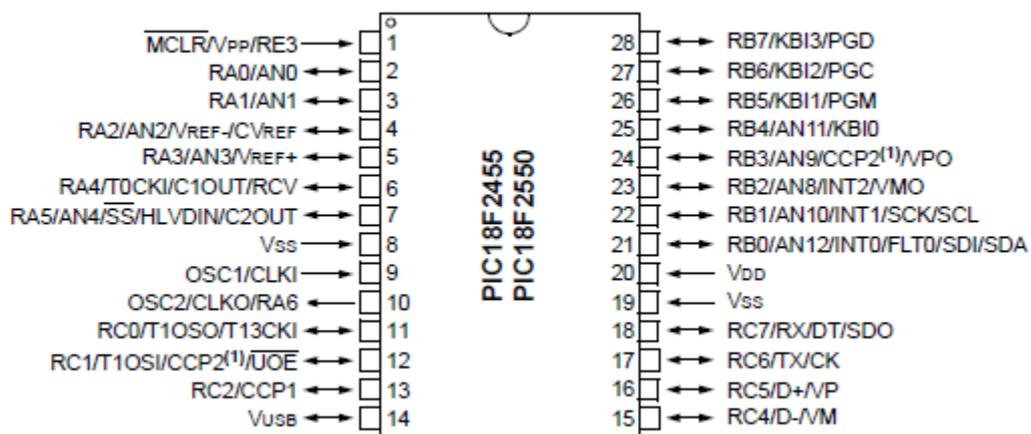


Ilustración 42: Pinout del PIC 18F2550. <sup>12</sup>

<sup>11</sup> Imagen extraída de la hoja de datos: <https://ww1.microchip.com/downloads/en/device-doc/39632c.pdf>

Una vez seleccionado el microcontrolador, se desarrolló un circuito impreso para pruebas iniciales. Este circuito consiste simplemente del PIC, terminales de conexión para *ICSP (In-Circuit Serial Programming)*, los puertos de entrada y salida, el conector de USB hembra, los capacitores de polarización del módulo USB y de desacoplamiento y dos LEDs de verificación. Se decidió utilizar la tensión de alimentación de la PC mediante USB para alimentar al microcontrolador, con el fin de evitar la necesidad de incorporar una fuente de alimentación externa adicional. Esto nos obligó a unir las masas del circuito de control y de señales, ya que existen diferencias de potencial estáticas entre la alimentación del circuito de señales y la conexión USB de cada PC, y sus valores presentar riesgos elevados para los circuitos de los mismos.

Se utilizó lenguaje **C** para la programación del PIC, con el compilador **CCS PIC C**, debido a su simplicidad y masividad de uso.

### El Proceso de Comunicación.

Tal vez la secuencia más simple de la programación del microcontrolador, sea el proceso de comunicación. Para esto, se decidió utilizar la configuración *FS (full speed)* que permite hasta 12 millones de bits de información por segundo, en cada dirección. La comunicación USB funciona mediante un proceso de paquetes de datos de dimensión constante. Esta dimensión es de 64 Bytes. Por lo que, para obtener la mayor velocidad de transmisión posible, estos 64 Bytes deben contener información útil.

Si consideramos que esos datos son muestras de dos señales diferentes adquiridas a más de 1600 Hz, entonces esos 64 Bytes se completan con facilidad en pocos ciclos. Sin embargo, en el otro extremo de la comunicación, solo se envían comandos y actualizaciones de variables de control, que se emiten a frecuencias mucho menores. Por lo tanto, la dimensión de los paquetes transmitidos desde la PC hacia el microcontrolador, serán mucho menores, pero se envían inmediatamente después de que se genera un comando o información pertinente.

Para establecer esta comunicación asincrónica y asimétrica, se estableció un bloque de comando muy compacto en el cuerpo principal del programa del microcontrolador. Este bloque realiza dos tareas principales:

- 1) Revisar el BUS USB para determinar si se recibió un paquete de comandos. En caso afirmativo, se establece una bandera y se habilita la interrupción del *TIMER1*, encargada de procesar la información recibida. El BUS de recepción se copia y se libera.
- 2) Enviar el paquete de 64 Bytes con la información de las señales, una vez el BUS de transmisión se ha llenado con información útil. Esta decisión de envío es determinada por el valor de una bandera, establecida por la función de empaquetamiento de datos.

La información recibida por el microcontrolador consiste de dos tipos de datos. Por un lado, se encuentra el comando específico emitido por el programa de la PC, y por el otro, las variables de control y las señales procesadas por el bucle de control.

El procesamiento de los comandos es muy simple y de naturaleza digital. Su ejecución se realiza mediante una operación *switch* en la función de interrupción del *TIMER1*. Esta interrupción debe generarse con una frecuencia superior a la frecuencia de generación de comandos y señales de control. Cuando describamos el bloque de procesamiento del software de PC veremos que este envía paquetes de datos a una frecuencia de aproximadamente 100 Hz. En consecuencia, se establece la interrupción del *TIMER1* al doble, 200 Hz.

Los comandos recibidos y procesados por el microcontrolador son los siguientes:

- Comandos de inicio y finalización de proceso de muestreo, así como el reseteo de las variables temporales de sincronización.
- Comandos de encendido y apagado de la bomba y/o la electroválvula de seguridad de 5 Vdc.
- Comandos de alteración del formato de las variables transmitidas. Estos comandos fueron de utilidad en el proceso de prototipado únicamente y no presentan utilidad funcional.
- Comandos de habilitación de los procesos de control de presión. Habilitan banderas que permiten la ejecución de instrucciones de PWM del sistema neumático.

Todos estos comandos son realizados por la función de interrupción de forma inmediata. Seguidamente, se continúa con el proceso de desempaqueamiento de la información recibida. Con el fin de obtener una comunicación más rápida, con menor cantidad de Bytes, los 2 bits superiores, en todos los datos de 10 bits de la comunicación, son empaquetados dentro de un mismo Byte común, que nos permite enviar 4 datos de 10 bits en 5 Bytes, en lugar de utilizar 8.

Los datos recibidos por el microcontrolador son operados por la función *unPackSignals()*. Estos datos son utilizados por funciones iteradas específicas para ejecutar los comandos resultantes del proceso de control, y son las variables determinantes de la regulación de presión en el sistema neumático. En el Capítulo 6 se detalla la utilidad de estas variables y su importancia para el proceso de control. Estos datos constituyen 6 Bytes de información y son:

- 1) Las señales de presión y fotopletiemográfica, post-procesamiento digital (filtros, offsets, etc.).
- 2) Los valores de referencia de presión del proceso de control.
- 3) El Ciclo de trabajo (*Duty Cycle*) del PWM de la bomba presurizadora.
- 4) La longitud del tren de pulsos que regula la apertura de las electroválvulas de control de presión.

## El Proceso de Muestreo y Empaquetamiento de Señales.

La función de muestreo y conversión analógico digital es la más importante para el funcionamiento de controlador. Sin información adecuada, la lógica de comando no puede operar. Existen múltiples variables que se debieron tener en cuenta para construir un proceso estable de adquisición. Es importante destacar que, si bien la señal de presión es de suma importancia para nuestro prototipo, esta señal es muy estable y de muy baja frecuencia. Sus niveles y características no presentan mayor dificultad para nuestro proceso de adquisición. En consecuencia, de aquí en más nos enfocaremos únicamente en la señal fotople-tismográfica.

Antes de comenzar a programar la adquisición de información, fue necesario resolver un inconveniente menor. Como el microcontrolador utiliza la alimentación del puerto USB de la PC, y el valor de esta alimentación no es constante, sino que difiere de una computadora a otra y de un instante a otro, no se puede utilizar esta tensión como referencia superior de las conversiones AD. Para solucionar esto, se agregó al circuito final de la placa de control, una red de polarización de Diodo Zener con un valor de 5.1 Vdc que se utilizó como referencia superior del módulo ADC (conectándola a la terminal +Vref del PIC18F2550 y configurando la referencia por software).

Inicialmente, se debió caracterizar el proceso mediante el cual se genera esta señal y las variables que la condicionan. En la Ilustración 43 se observa un diagrama representativo de la generación y procesamiento de la señal fotople-tismográfica en el circuito de señales. Es de suma importancia considerar que cada uno de estos bloques introduce un *delay* en la señal fotople-tismográfica. Por ejemplo, en la Ilustración 24 se observa la estimación del *Group Delay* introducido por el Filtro Pasa Banda de 6to orden.

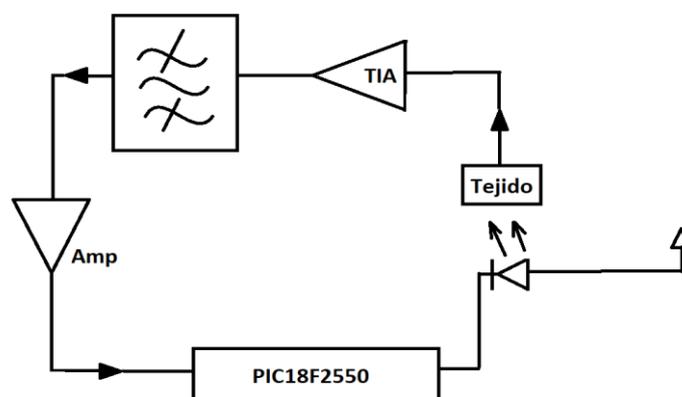


Ilustración 43: Diagrama del recorrido de la señal fotople-tismográfica.

Resulta evidente que existe un beneficio en utilizar la misma función de interrupción para generar la modulación del LED IR y para muestrear la señal, ya que ambas operaciones se deben establecer a la misma frecuencia. En consecuencia, se utiliza la función de interrupción del *TIMERO* para esta tarea.

De acuerdo con el Teorema de Nyquist, para obtener una caracterización adecuada de la señal, es necesario muestrear a una frecuencia mínima de 3200 Hz. Originalmente, se consideró que, si bien la señal se presenta a la entrada del ADC con un *delay* considerable, como la señal se genera y se muestrea por la misma función, si se tomaban las precauciones adecuadas para la programación de la misma, se podría obtener una señal limpia y estable mediante muestreo sincrónico a 3200 Hz. Se consideró sincronizar la modulación del LED y los instantes de muestreo, de modo tal que estos últimos coincidan con los extremos de la señal fotopletoisométrica, como muestra la Ilustración 44.

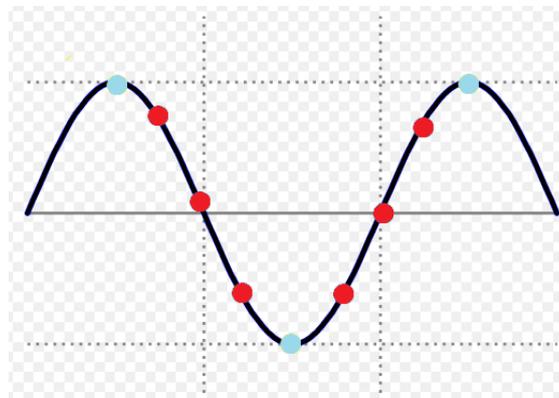


Ilustración 44: Representación de los puntos de muestreo deseados (en verde) y los no deseados (en rojo).

Sin embargo, por más precauciones que se tuvieron, el *delay* instantáneo de la señal fotopletoisométrica varía de forma constante. Es decir, en lugar de que las muestras se tomaran a intervalos idénticos y sincrónicos con la señal, tanto los intervalos como la señal se desfasaban entre sí de forma constante en el tiempo, ya sea por imprecisiones en el sincronismo de operaciones del microcontrolador o por variaciones del *Group Delay* del circuito de amplificación y filtrado. En consecuencia, la señal observada por el usuario presentaba un ruido de modulación de naturaleza periódica cuya amplitud era tan grande como la señal misma y cuya frecuencia dependía de las variaciones en la programación de las rutinas de interrupción, tal como muestra la Ilustración 45.



previamente obtenidos en este último ciclo (ciclos de 1.6 kHz). Este proceso se realiza para extraer solo los extremos de la señal fotopleletismográfica en cada ciclo, indistintamente del desfase de la misma con el inicio del ciclo. Luego de dicha operación se inicia una nueva conversión. Si consideramos el tiempo adicional de realizar estas instrucciones, podemos estimar que se realiza una conversión cada  $35 \mu\text{seg}$ . En consecuencia, la frecuencia real de muestreo es:

$$F_m = \frac{1}{35 \mu\text{seg.}} = 28571 \text{ Hz} = 28.5 \text{ kHz.}$$

**Ecuación 15: Frecuencia de muestreo del módulo ADC.**

Con esta nueva frecuencia de muestreo, la señal fotopleletismográfica se observa estable y, si bien existe una interferencia electromagnética, su amplitud máxima en condiciones normales no supera los 50 mV a la entrada del ADC, y puede ser filtrada de forma digital sin mayores inconvenientes.

Además del muestreo de señales y la modulación del LED infrarrojo, el *TIMERO* también realiza el empaquetamiento de los datos. Por cada ciclo de operación a 1600 Hz, la interrupción del *TIMERO* obtiene dos datos de señal fotopleletismográfica y uno de presión. La componente superior de la conversión (con tensiones de 3 Vdc a +Vref), es una señal reflejada de la componente inferior. Para sumarlas, se invierte el valor de la componente superior y luego se la suma a la inferior. De modo tal que existe un único dato de señal fotopleletismográfica, con una amplitud total de 10 bits.

De los 64 Bytes disponibles en el BUS de transferencia USB, los primeros 4 son reservados para comunicación de confirmación de comandos y variables de realimentación del PIC a la PC, y los 60 restantes se distribuyen en grupos de 4 datos por cada iteración. Cada grupo de 4 Bytes está formado por:

- 1) Byte de sincronismo temporal *timeL*, que varía secuencialmente de 0 a 255 con cada muestra. Es utilizado para verificar que no exista pérdida de información en la comunicación.
- 2) Byte combinado de los 2 bits superiores de las señales de presión y fotopleletismográfica. Este Byte contiene la información como **0b000AAABB**, donde **AAA** son los bits superiores de la señal fotopleletismográfica y **BB** son los bits superiores de la señal de presión.
- 3) Byte inferior (LSB) de la señal óptica.
- 4) Byte inferior (LSB) de la señal de presión.

Estos datos son organizados secuencialmente por las funciones *packReading()* y *packSamples()*. Una vez el BUS de transmisión se llena con muestras de las señales, se establece la bandera de envío para que el programa principal realice la transmisión mediante USB. Bajo este proceso de

empaquetamiento, el microcontrolador envía 15 muestras de cada señal por transmisión (60 Bytes / 4 Bytes por ciclo), a una frecuencia de 106 Hz (1600 Hz / 15).

El software de PC solo puede tomar decisiones cuando recibe información, en consecuencia, la frecuencia del sistema de control también será de 106 Hz. El *software* de PC presenta una complejidad mucho mayor que el *firmware* del microcontrolador y su estructura y funcionalidad son descriptos en los Capítulos 6 y 7.

### El Proceso de Control del Sistema Neumático.

Como ya hemos mencionado, las principales variables del control del sistema neumático son calculadas en el programa de PC, debido a que requieren de un proceso computacional un poco complejo, y realizar estos cálculos en el microcontrolador provocaría graves demoras.

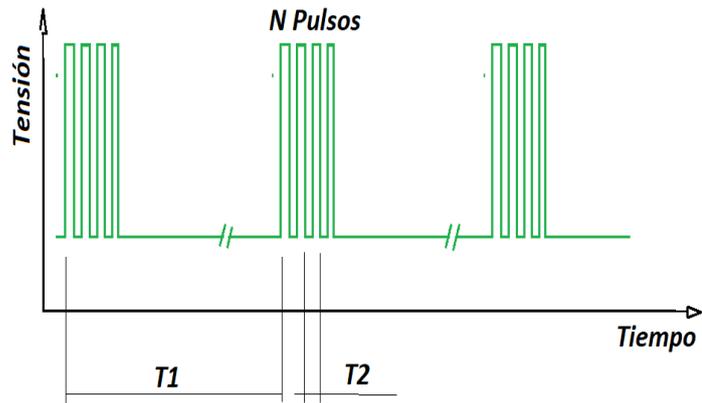
En consecuencia, las variables de control son recibidas mediante USB y utilizadas por el microcontrolador. Dichas variables son:

- Ciclo de trabajo de la bomba neumática: Valor entero positivo de 0 a 30, donde 30 es 100% de *Duty Cycle*. Resolución de 3.33%
- Número de pulsos de ancho y frecuencia constante, cada período de tiempo  $T$  también constante, para la conmutación de las válvulas de control.

El control de la bomba se realiza mediante un simple PWM. Su función de trabajo se incluye en la interrupción del *TIMER0*, por lo que, como esta interrupción se dispara a una frecuencia de 25.6 kHz y el ciclo de trabajo establecido por la función *pwmPump()* requiere de 30 iteraciones, la bomba se regula con un PWM a 853 Hz.

Sin embargo, la funcionalidad de las electroválvulas es un poco más compleja. Como las electroválvulas deben trabajar contra fuerzas estáticas y dinámicas de diferente valor, una regulación de su conmutación por porcentaje de ciclo de trabajo no es muy práctica. Ya que un 10% de *Duty Cycle* no es suficiente energía como para generar ni siquiera un 1% de apertura valvular a la frecuencia antes mencionada de 853 Hz.

En consecuencia, el PWM de las electroválvulas debió modularse con una frecuencia elevada, es decir, el ciclo de trabajo resulta de la envolvente de un tren de pulsos de duración  $T_2$ , emitidos de forma constante a un período  $T_1$ . El ciclo de período  $T_1$  está regulado por la función *dynamicControl()* ejecutada en la interrupción *TIMER1* y la modulación de período  $T_2$  se ejecuta en la interrupción del *TIMER0*. En consecuencia, esta envolvente tiene una frecuencia de 183 Hz y la frecuencia del tren de pulsos es de 25.6 kHz.



**Ilustración 46:** Tren de pulsos para control de electroválvulas.

Esta forma de control de frecuencia complementaria fue la que obtuvo mayor linealidad en la respuesta del sistema neumático al realizar las pruebas del sistema de control.

## Capítulo 6: El Sistema de Control.

Un sistema de control automático es una interconexión de elementos que forman una configuración denominada *sistema*, de tal manera que el arreglo resultante es capaz de controlarse a si mismo. Cada sistema se constituye por bloques o elementos a los que se les aplica una (o varias) señal(es) de entrada, y del cual surge una señal de salida [52].

Para establecer la señal de presión o la señal fotopletoislográfica al nivel deseado, fue necesario establecer un lazo cerrado de control. El sistema descrito para el método VCM es de lazo cerrado naturalmente, sin embargo, fue necesario incorporar un proceso de control más complejo, para manipular la apertura de las válvulas, ya que el control meramente proporcional presenta múltiples limitaciones, como error en estado estable y tendencias a oscilar.

El controlador PID es la forma más común de realimentación de procesos. Este tipo de controlador establece el valor de su salida, a partir de una señal *error* y según una ecuación de la forma [53]:

$$u(t) = K_p \left( e(t) + \frac{1}{T_I} \int_0^t e(\tau) \cdot d\tau + T_d \frac{de(t)}{dt} \right)$$

**Ecuación 16: Proceso de control PID.** En donde,  $u(t)$  es la variable de control,  $e(t)$  es el error, es decir la señal a controlar menos la referencia,  $K_p$  es la ganancia proporcional.  $T_i$  es la ganancia integradora y  $T_d$  es la ganancia derivadora. [53]

Con este tipo de controladores, mediante la ganancia integral, se pueden resolver los problemas de error constante en estado estable. Además, con la ganancia derivadora, se pueden reducir las oscilaciones y el sobreamortiguamiento de los sistemas [53]. Para sistemas discretos, es posible implementar este controlador de acuerdo con la Ecuación 17.

$$u[n] = K_p e[n] + K_I \sum_{n=0}^N e[n] + K_D (e[n] - e[n - 1])$$

$$e[n] = x[n] - x_{ref} \quad K_I = \frac{K_p}{T_i} \quad K_D = K_p \cdot T_d$$

**Ecuación 17: Fórmula Discreta de Control PID [54].**

El sistema de control utilizado en este prototipo es no lineal, ya que no cumple ni con el principio de superposición, ni con la propiedad de homogeneidad. Además, este sistema presenta una gran histéresis y saturación. La histéresis es el retraso que experimenta un fenómeno con respecto a la causa que lo genera, lo cual provoca que la magnitud resultante dependa de los estados anteriores del sistema [52].

Cabe destacar, que las principales alinealidades que encontramos en nuestro sistema son debido a la saturación de los componentes utilizados. Tanto la bomba, como la capacidad de apertura de las válvulas, presentan un límite considerable. Sería relativamente simple linealizar el sistema bajo estas condiciones, si no fuera porque, además, ciertas propiedades de las electroválvulas cambian con la temperatura. Estas válvulas experimentan un calentamiento considerable, debido a la elevada corriente que manejan. A medida que el Manifold se calienta, la componente integradora del mecanismo de apertura se exacerba.

Habiendo dicho esto, es posible linealizar parcialmente el sistema si establecemos ciertos límites de operación:

- 1) Región de trabajo entre 50 y 150 mmHg. Como el caudal de aire que atraviesa un conducto depende de la diferencia de presión entre sus extremos, y como las presiones en cuestión se modifican a medida que este caudal se transmite de una cavidad hacia la otra, la respuesta del sistema en lazo abierto es una exponencial inversa. Para simplificar la dificultad de caracterizar el sistema neumático, se redujo el rango de trabajo de modo que no sea mayor al estrictamente necesario.
- 2) Linealización del trabajo de la bomba. Como se mencionó en el Capítulo 4, para generar diferentes cambios de presión, se requieren diferentes caudales, dependiendo de la presión inicial y del volumen. En consecuencia, para suponer que una apertura unitaria de la válvula presurizadora genera un cambio unitario de presión, independientemente de la presión original, se debió escalar el caudal suministrado por la bomba proporcionalmente con la presión actual de trabajo.
- 3) Propiedad integral de la bomba. Como la bomba presenta una componente integradora importante, su acción de comando debe tenerla en cuenta para obtener una respuesta temporal con un *delay* mínimo. Es decir, si esperamos a que la bomba se encienda cuando se genera un cambio del error de la señal de presión, el tiempo de respuesta podría ser de hasta 1 segundo. En consecuencia, la acción de control de la bomba debe ser proporcional a la integral del caudal necesario. El caudal necesario depende del proceso de funcionamiento (estable o dinámico) y, por ende, debimos de adaptarlo de acuerdo a las necesidades.

Otra alinealidad importante es que, si bien las electroválvulas de presurización y escape son similares, sus respuestas temporales serán completamente diferentes. La apertura de la electroválvula de escape provocará un caudal de fuga igual a:

$$Q_e = \frac{P_{fun} - P_{amb}}{R_{valvula}}$$

. Ecuación 18: Caudal de fuga de la válvula de escape.  $P_{fun}$  es la presión del circuito funcional,  $P_{amb}$  es la presión atmosférica y  $R_{valvula}$  es la resistencia al paso del volumen de aire.

Esto implica que la respuesta de esta válvula puede ser linealizada con bastante exactitud. Sin embargo, el caudal originado por la electroválvula de presurización será:

$$Q_c = Q_b * \left( \frac{P_{pr} - P_{fun}}{R_{válvula}} \right)$$

**Ecuación 19: Caudal de presurización. En donde  $P_{pr}$  es la presión suministrada por la bomba y  $Q_b$  es el caudal impulsado por la bomba.**

En donde  $P_{pr}$  es la presión suministrada por la bomba y  $Q_b$  es el caudal impulsado por la bomba. Teniendo en cuenta que la presión de la bomba depende del caudal que esta impulsa, de modo tal que, si la bomba se encontraba en funcionamiento previo a la conmutación, el caudal será máximo al inicio y caerá hasta un valor estable, suponiendo que la presión del circuito funcional permanezca constante.

En consecuencia, aun teniendo en consideración que se necesita un controlador diferente para cada válvula, debemos tener en cuenta que la respuesta de presurización depende del producto temporal entre la respuesta de la bomba neumática y de la válvula de presurización. Como no es posible caracterizar la respuesta de cada uno de estos elementos por separado, con el fin de generar un modelo teórico de nuestro sistema, ya que deberíamos medir las variables de Caudal y Resistencia de Conducción, para múltiples puntos determinados de cada una de las 3 variables de control, debimos recurrir a herramientas diferentes para sintonizar y calibrar los controladores.

Con el objetivo de simplificar el proceso de control del sistema, el controlador del PWM de la bomba de presión se estableció mediante una ecuación Proporcional Ponderada, cuyas constantes fueron calibradas para la presión media de funcionamiento en diferentes operaciones, mediante aproximación recursiva. Esta ecuación de control es:

$$\%Duty = \frac{prPoint}{calPr} * Error * K_n + K_o$$

**Ecuación 20: Fórmula de control del ciclo de trabajo de la bomba.**

En donde:

- $\%Duty$  se expresa como un valor entero que se satura en 0 y 30 (0 a 100%).
- $prPoint$ : es el valor actual de presión mas un escalón de 25 mmHg (para evitar que el  $Duty$  se anule a 0 mmHg).
- $calPr$ : Es la presión media a la cual se calibraron los demás parámetros. En nuestro código este valor es 90 (65 mmHg mas el escalón de 25 mmHg).

- *Error*: es el error de señal, ya sea de presión o de señal fotopletimográfica.
- $K_p$ : es un factor de ponderación proporcional. En nuestra calibración resultó de 0.7
- $R_o$ : es el valor mínimo de ciclo de trabajo a partir del cual la bomba genera un caudal mensurable. Como la bomba se satura para ciclos de trabajo cercanos a 0, esta es necesaria para evitar el uso de un controlador integrador.

Este tipo de control garantiza que el caudal entregado por la bomba será proporcional a la presión actual del circuito funcional, de modo tal que el incremento de presión frente a un error determinado no sea lineal para todo el rango de funcionamiento.

Habiendo establecido esto, para el proceso de compensación volumétrica, el caudal promedio de presurización requerido por el circuito neumático es mucho mayor, ya que las pendientes de la curva de presión son pronunciadas y se repiten constantemente. En consecuencia, el ciclo de trabajo en estas condiciones se estableció permanentemente al 100%.

#### Modelo del Sistema Estable: El controlador de Presión.

Si bien resulta extremadamente complejo caracterizar las funciones de los procesos o sus respuestas temporales, o linealizar el funcionamiento del sistema, diseñar el modelo representativo resultó de gran utilidad. Esto nos permitió conceptualizar las variables determinantes del proceso de control y determinar la incidencia que estas tienen sobre el resultado final.

En primer lugar, se diagramó un sistema simple de control de presión. La presión de aire depende directamente del caudal neto en la cavidad de análisis. Este caudal es la sumatoria de los efectos producidos por la apertura de válvulas y de pérdidas de aire del circuito. Las pérdidas de aire son proporcionales a la presión, pero pueden ser despreciadas si el circuito funcional es construido correctamente. El caudal de salida depende del ciclo de trabajo (corriente neta) suministrado a la válvula correspondiente.

Por otro lado, el caudal de entrada depende de la corriente neta suministrada a la válvula y del caudal suministrado por la bomba de presión, que a su vez depende de la corriente neta suministrada a la bomba. Por lo que el caudal de entrada es el resultado de un producto de procesos. En la Ilustración 47 se observa el diagrama de bloques diseñado en *SimuLink*, del sistema de control de presión estable.

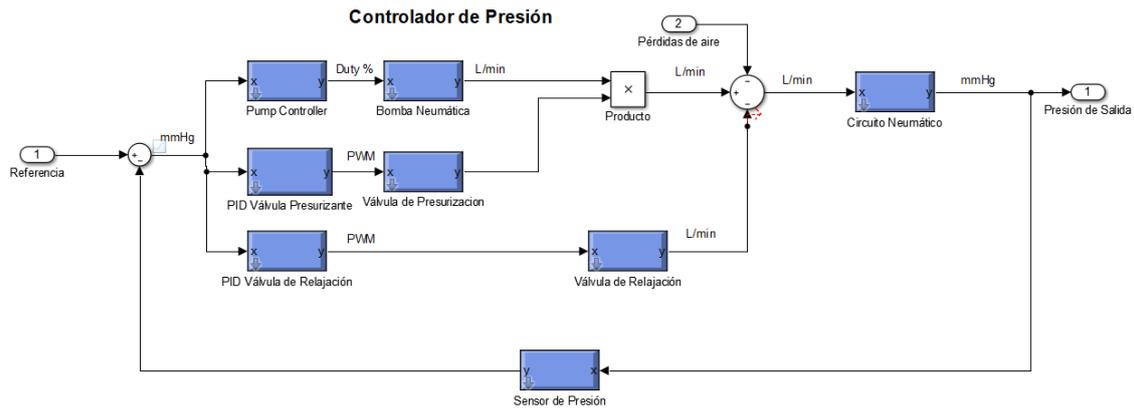


Ilustración 47: Diagrama de Bloques del Sistema de Control de Presión Estable.

Si bien los controladores pueden resultar difíciles de parametrizar, se realizó una simplificación al proceso de funcionamiento, a fin de facilitar la sintonización. Para esto, se estableció el ciclo de trabajo de la bomba al 100 %, de modo tal que el flujo de ingreso al circuito sea igual a una constante, por la salida del proceso de la válvula de presurización. En consecuencia, fue posible utilizar el Método de Ziegler Nichols para sintonizar los controladores.

Este método de sintonización en lazo cerrado consiste en anular las componentes integradoras y derivadoras de los controladores e incrementar gradualmente la ganancia proporcional, hasta alcanzar un estado de oscilación constante. Ese estado de oscilación es equivalente a una estabilidad marginal. De la oscilación resultante, se mide el período  $T_o$  y junto con la componente proporcional  $K_p$ , se obtienen los demás parámetros [52, 54].

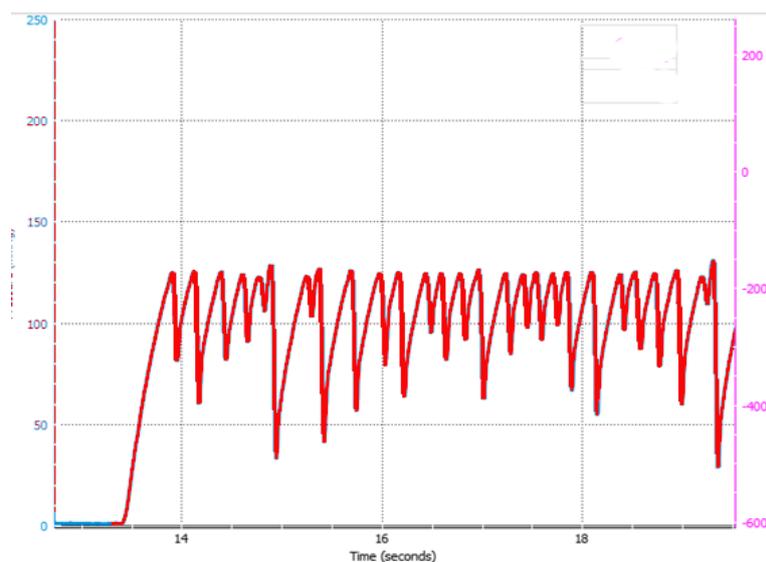


Ilustración 48: Oscilaciones utilizadas para el Método Ziegler Nichols.

Con este método, y mediante una ponderación adecuada correspondiente a la frecuencia de muestreo del sistema, fue posible determinar los parámetros de los controladores PID de las válvulas en estado de control de presión estable. Se obtuvo la respuesta temporal del sistema de control estable a la entrada escalón de 130 mmHg, tal como muestra la Ilustración 49.

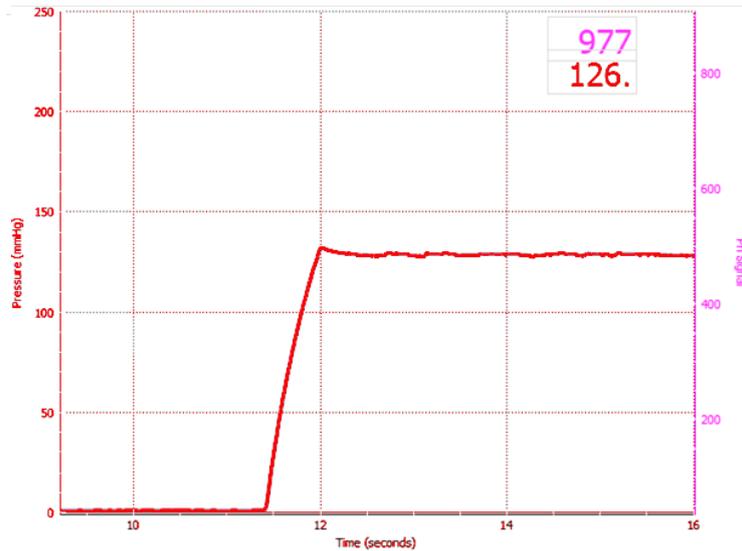


Ilustración 49: Respuesta temporal del sistema de control estable al escalón de 130 mmHg.

### Modelo del Sistema Dinámico: El Controlador de Volumen.

Una vez habiendo calibrado el sistema de control estable, y habiendo obtenido la respuesta temporal del sistema, se consideró incorporar un nuevo bucle de control con un nuevo controlador PID, tal como muestra la Ilustración 50. Ya que, conociendo la respuesta temporal del bucle interno, es posible obtener su función de transferencia y con ella estimar los parámetros del bucle externo.

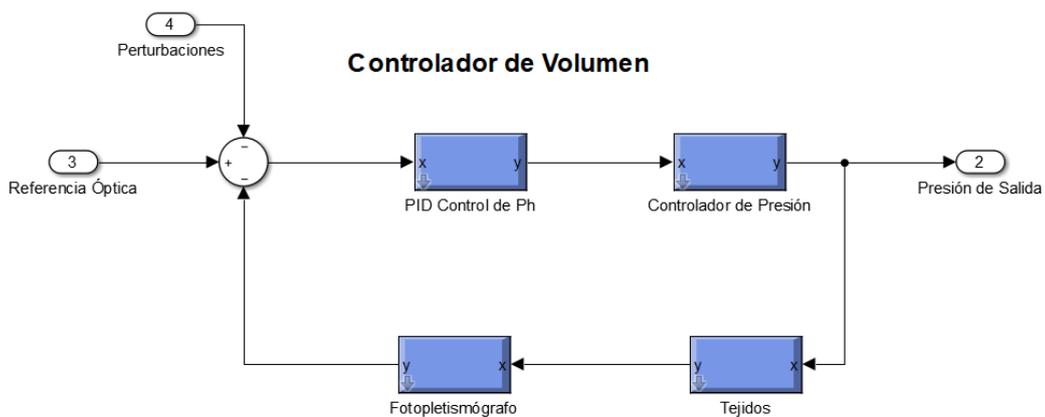


Ilustración 50: Sistema de control con bucle anidado.

Sin embargo, existen dos grandes problemas. Por un lado, las funciones de transferencia fotopleletismográficas del tejido y del transductor, dependen en gran medida de cada individuo y del posicionamiento de la almohadilla. Por lo que la calibración para una persona puede requerir grandes modificaciones para ser utilizada en otra.

Y por el otro, el tiempo de respuesta del bucle interno es muy elevada, como ya vimos, hasta 0.5 segundos para una variación de 100 mmHg. Como el pulso de presión tiene un período total de entre 0.4 y 1.2 segundos en condiciones normales, la sintonización del bucle externo solo incorporaría mayor demora en la respuesta del sistema anidado.

En consecuencia, se decidió evitar la implementación de un nuevo controlador, y se utilizó un arreglo muy similar al modelo dinámico que para el controlador estable, con la diferencia que se utilizó la referencia óptica en lugar de la referencia de presión. Por lo que la amplitud de la señal *error* es mucho menor. Pero resultó mucho más simple de sintonizar los controladores de las electroválvulas para este sistema, a partir de las ganancias obtenidas para el sistema de control estable.

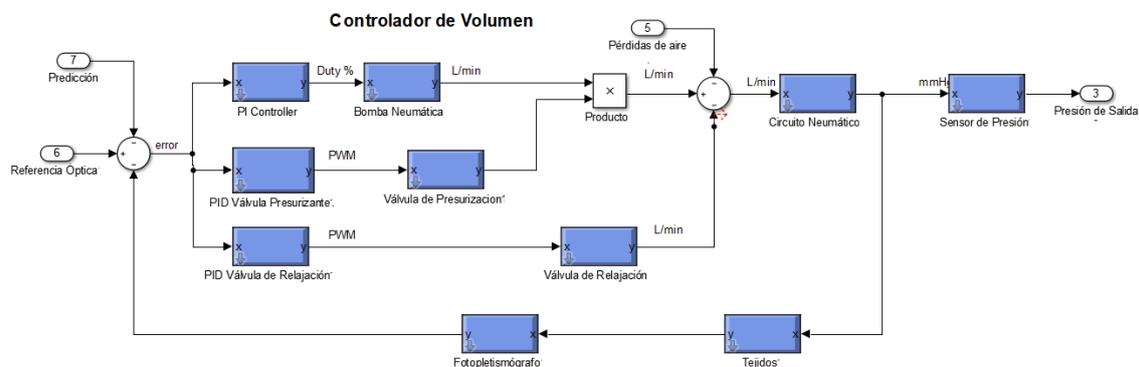
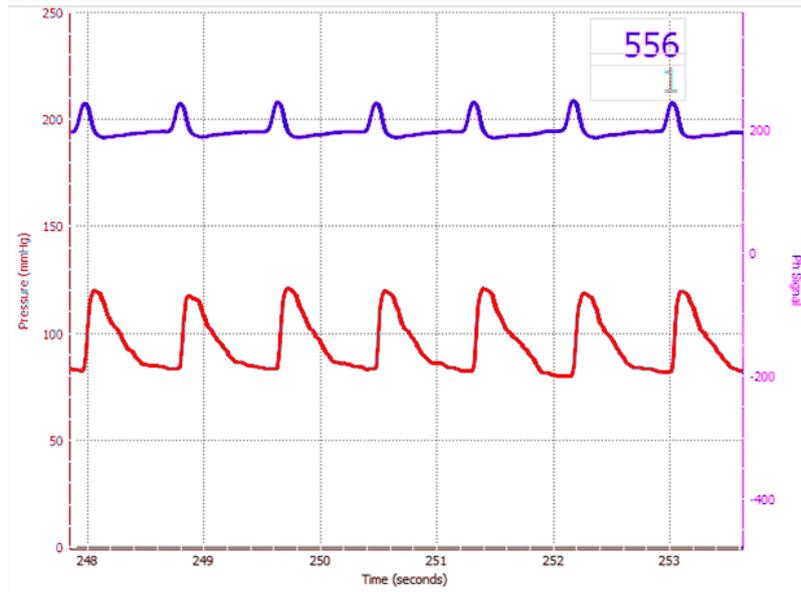


Ilustración 51: Diagrama de bloques del sistema de control Dinámico de Volumen.

De esta forma la sintonización se adecuó perfectamente a las necesidades. En la Ilustración 52 se observa una de las primeras mediciones de la presión arterial utilizando el método de compensación volumétrica y con el sistema de control diagramado en la Ilustración 51. La referencia óptica para esta medición se estableció manualmente por aproximación. Se puede observar un retraso importante en la respuesta del sistema de presurización, de hasta 300 milisegundos. En el Capítulo 7 se detalla el procedimiento que se utilizó para reducir este *delay* mediante un mecanismo de *predicción*.



**Ilustración 52: Resultados de medición sin predicción. Primera sintonización de controladores.**

## Capítulo 7: El Software de PC.

La última porción del sistema integral de control está constituida por la interfaz de usuario y su *backend* correspondiente. Si bien no merece la pena describir en detalle todas las funciones y la estructura general del programa, si es importante describir las funciones y operaciones que interceden en el proceso de control. Como ya hemos mencionado, algunas de las funciones más importantes realizadas por este software son:

- 1) Comunicación bilateral, recepción de información, aplicación de filtros digitales y acondicionamiento de señales, ejecución y transmisión de comandos de control.
- 2) Exposición de los datos, gráficas y tablas con información pertinente para la comprensión del proceso.
- 3) Detección de picos, periodicidad de señales, frecuencia cardíaca, frecuencia de muestreo y la integridad de la señal.
- 4) Coordinación y ejecución de todos los procesos de control, calibración y resolución de acciones.

El programa principal se programó en **C++** y la interfaz de usuario con **Qt 5.10**. Se utilizó **Visual Studio 2015 Enterprise** para el entorno de desarrollo. Se utilizaron librerías *open source* de módulos USB HID, controladores PID y, principalmente, la herramienta *QCustomPlot* de Qt.

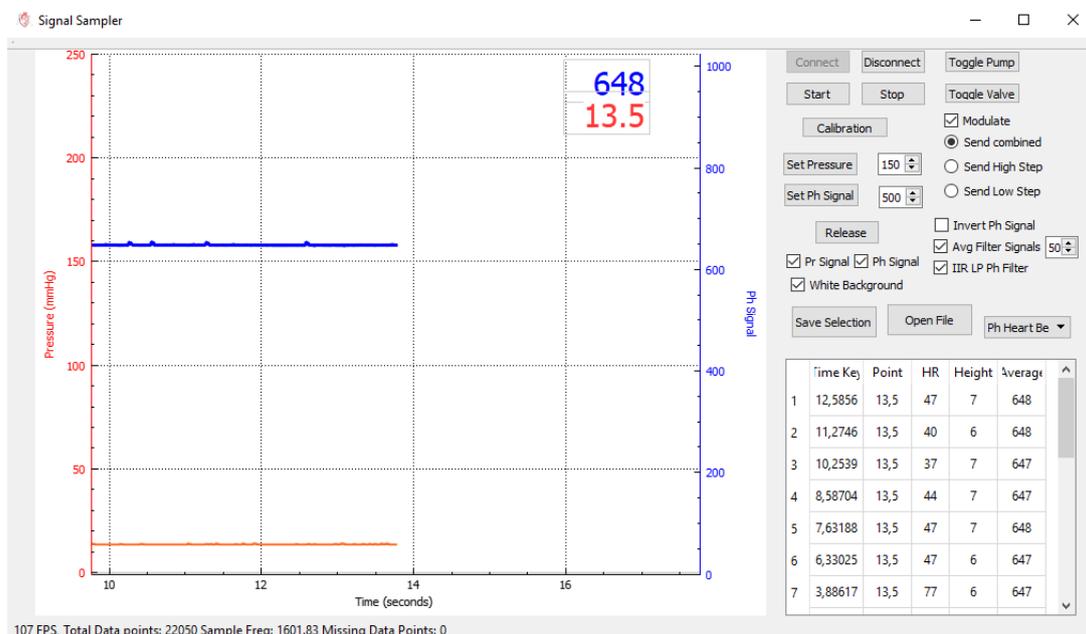


Ilustración 53: Interfaz de usuario del software de PC.

### Post Procesamiento de señales.

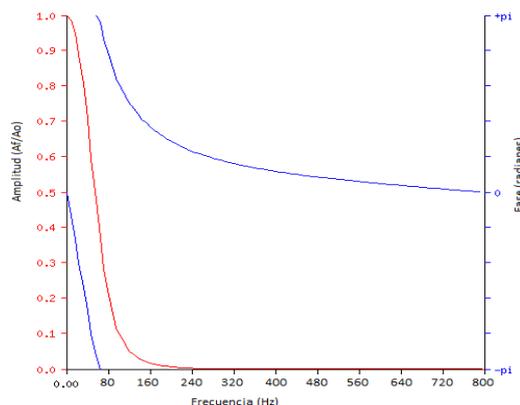
Pese a las precauciones tomadas en el diseño de circuitos electrónicos y de procesos de comunicación, las señales pueden presentar un cierto nivel de ruido debido a múltiples causas, como son: Inestabilidad del amplificador de transimpedancia, perturbaciones en el proceso de conversión AD, interferencia electromagnética, etc. Este ruido presenta dos componentes, una de alta frecuencia, constante y de baja amplitud, y otra de naturaleza esporádica de frecuencia aún mayor y de gran amplitud. En normales, el ruido constante no supera los 150 mV y se presenta en ambas señales, y los picos de ruido esporádico, presentes únicamente en la señal fotopletoislográfica, alcanzan los 400 mV. Para eliminar este ruido se aplicaron tres filtros diferentes.

Sobre la señal de presión se aplica un filtro promediador móvil (*Moving Average Filter*), de  $N$  muestras, donde  $N$  es seleccionado por el usuario. Para filtrar la señal fotopletoislográfica, se implementó un filtro pasa bajo digital de tipo IIR (*Infinite Impulse Response*) de orden 4, cuya ecuación es [55]:

$$Y[4] = (x[0] + x[4]) + \alpha_0(x[1] + x[3]) + \alpha_1x[2] + \sum_{n=0}^3 \beta_n Y[n]$$

**Ecuación 21: Filtro discreto de 4to Orden Tipo IIR [55].**

Los coeficientes  $\alpha$  y  $\beta$  se obtuvieron con la herramienta web *Interactive Digital Filter Design mkFilter* (publicada por la Universidad de York, Reino Unido, 1999). Se seleccionó una frecuencia de corte de 40 Hz y una respuesta Bessel. En la Ilustración 54, se observa la respuesta en frecuencia del filtro obtenido con esta herramienta.<sup>12</sup>



**Ilustración 54: Respuesta en frecuencia del filtro IIR pasa bajo. En Rojo, la magnitud de la respuesta (eje Y izquierdo) y en Azul, la fase (eje Y derecho).<sup>13</sup>**

<sup>12</sup> Herramienta MkFilter: <https://www-users.cs.york.ac.uk/~fisher/mkfilter/>

Finalmente, para eliminar el ruido esporádico, se debió primero caracterizar estas perturbaciones. Se observaron como picos abruptos de distribución aleatoria, de 0.005 a 0.012 segundos de duración y con una amplitud de 200 a 400 mV (40 a 80 bits de información digital). Por lo que, para eliminar estos picos, sin introducir un filtro pasa bajo de mayor orden que pueda disminuir la amplitud de la señal pulsátil, se implementó un código de detección mediante la derivada instantánea de la señal y se los eliminó, haciendo a todos los puntos hasta 12 milisegundos después de la detección, iguales al último punto medido previo a la perturbación (una atenuación total de 19 datos).

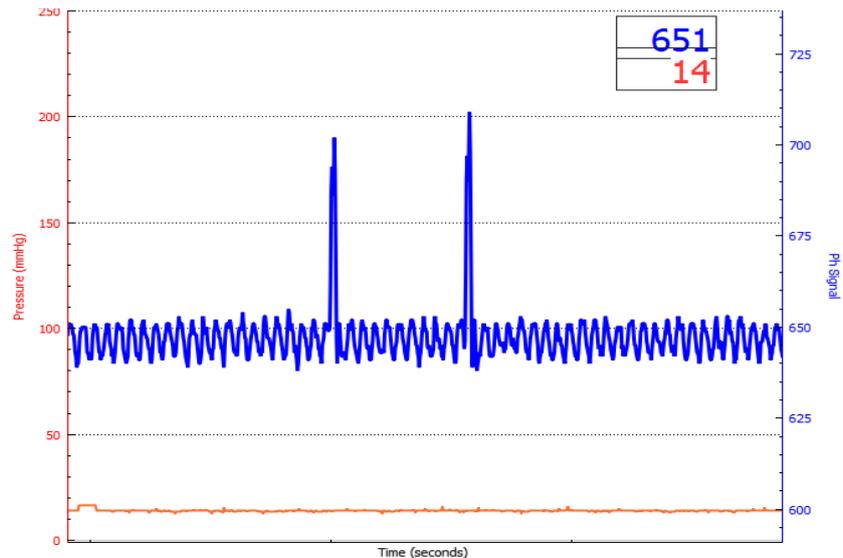


Ilustración 55: Ruido remanente previo al Post Procesamiento Digital.

### Proceso de Detección de Pulsos.

La detección de pulsos es necesaria para determinar múltiples variables necesarias para el funcionamiento del sistema de control, como medición de la presión media, de la frecuencia cardíaca, etc. Los pulsos pueden ser detectados por medio de ambas señales indistintamente, pero mediante procesos ligeramente diferentes. La señal fotopleletismográfica presenta un rango mucho mayor, puede establecerse a niveles diferentes y la amplitud de las pulsaciones es de un máximo de 250 mV. Por otro lado, la señal de presión es mucho más estable y de variaciones fácilmente predecibles.

Por lo que la detección de pulsos de fotopleletismografía requiere de detección de derivadas de valores inferiores, de detección de picos, de valores iniciales y de estabilidad de la señal. La función *phPeakDetect()* realiza esta tarea. La función *prPeakDetect()* que detecta los pulsos en la señal de presión, simplemente determina un pulso por la superación de un umbral en la derivada de la señal. La derivada de las señales se obtiene a partir de la diferencia entre el último valor de la señal y el promedio de las últimas 30 muestras.

Para cada pulso se calculan las siguientes variables:

- *timeKey*: Instante temporal del inicio del pulso.
- *pxAverage*: Valor promedio de la señal, calculado como el promedio de la señal durante el pulso anterior.
- *pxPoint*: Valor instantáneo de la otra señal (FPG o Presión), en el momento inicial del pulso.
- *pxHeight*: Amplitud máxima del pulso.
- *HRFreq*: Frecuencia cardíaca del pulso, calculada como la inversa del período promedio de los últimos 5 pulsos.

Estos parámetros se calculan para cada pulso y se los agrega al principio de la tabla dinámica que figura en la sección inferior derecha de la interfaz de usuario.

### Proceso de Calibración.

El control de presión se realiza mediante los bucles de control PID y se establece con un valor de referencia *prRef*. Para establecer el funcionamiento automatizado, es necesario determinar los valores de presión media del individuo y de la referencia fotopleletismográfica adecuada.

El proceso de calibración se inicia con el circuito funcional abierto a la presión atmosférica y determinando el offset de la señal de presión. Este *offset* establece la señal a 0 mmHg, y seguidamente cierra el circuito neumático. A continuación, y de forma iterativa, se incrementa la referencia de presión para generar una recta ascendente de presión. Este ascenso es utilizado para realizar una medición de presión arterial media con el método foto-oscilométrico.

Una vez determinada la presión arterial media, se establece la presión de referencia a este valor. Luego de un leve intervalo, se detectan los pulsos de la señal fotopleletismográfica y se obtiene el valor promedio de los primeros 5 pulsos. Este valor promedio constituye la nueva referencia de señal fotopleletismográfica necesaria para el proceso de compensación volumétrica. Se emite una ventana con una solicitud de confirmación para el usuario, con los valores obtenidos de presión media y de referencia de FPG. Si el usuario determina que estos valores son adecuados, se inicia el proceso de compensación volumétrica.

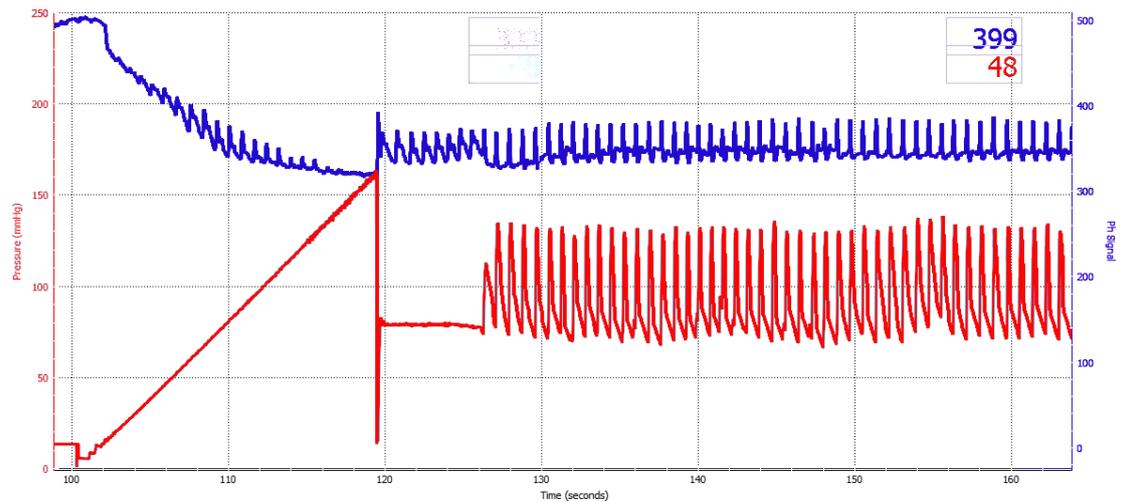


Ilustración 56: Calibración de medición inicial, sin recalibración de presión.

### Proceso de Recalibración.

La principal desventaja de método propuesto por Peñaz [26], radica en el desplazamiento del tejido subyacente al brazalete ocluser. Como se aplica una presión variable sobre el tejido, este se ve desplazado con el paso del tiempo, por lo que la referencia volumétrica cambia y, en consecuencia, también cambia la medición de presión. Como la referencia es de gran importancia para este proceso de medición, esta debe ser re adaptada.

Tanto los dispositivos *Finapres*, como los prototipos propuestos por *Yamakoshi et al*, realizan una recalibración similar, en la que cada un intervalo determinado de tiempo, el punto de referencia se varia mediante escalones abruptos. Estos cambios de la referencia establecida producen cambios en la medición que alteran la presión media. El objetivo de la calibración es igualar la presión media de las curvas obtenidas con el método de compensación volumétrica a la presión media obtenida en el proceso de calibración inicial. Por lo que se calcula una nueva referencia, que obtiene esta igualdad, mediante interpolación de los escalones establecidos en la referencia y las presiones medias de las mediciones con dichos escalones [34, 31, 29, 28, 17].

Para realizar el proceso de recalibración en este PI se propuso un proceso diferente. La presión media se mide latido a latido y cada 5 segundos se determina si esta presión se encuentra dentro de un rango aceptable, de la presión media de calibración. Si la presión media difiere con la calibrada, se obtiene la diferencia entre estas dos presiones, se la multiplica por un factor constante, y se la adiciona a la referencia fotopletoyomográfica. Bajo estas condiciones, si la presión media medida está por encima de la buscada, se incrementa la

referencia óptica y la señal de presión se desplaza hacia abajo. En caso contrario, se reduce la referencia y la señal de presión se desplaza hacia arriba. El usuario tiene la posibilidad de modificar las referencias durante el proceso de medición si así lo deseara.

## Capítulo 8: Normativa.

El marco regulatorio es el responsable de garantizar la seguridad y el correcto funcionamiento de los equipos. Los principales entes reguladores son aquellos de la Comunidad Europea. En nuestro país las normas para equipos electromédicos son creadas a partir de las regulaciones europeas. Por ende, nos enfocaremos en estas últimas.

Este prototipo no fue diseñado teniendo en cuenta los requerimientos normativos. Si bien se trató de construir un dispositivo seguro y confiable, las características de seguridad fueron determinadas en base a la experiencia y el sentido común. Sin embargo, es importante revisar el marco regulatorio, ya que es el mayor factor determinante de la viabilidad comercial de un equipo.

La comercialización de equipos electromédicos en la Unión Europea se regula mediante el Mercado CE. Este mercado se dispone sobre todos los equipos que cumplen con los estándares y ensayos necesarios para garantizar su seguridad y performance. La norma principal que determina los requerimientos para el mercado CE, es la norma 93/42/CE.

### Norma 93/42/CE.

Para poder navegar el marco regulatorio establecido por esta norma, primero debemos establecer las características del equipo que son pertinentes a su clasificación, ya que dicha clasificación afecta directamente a los requisitos que el dispositivo diseñado debe cumplir. La clasificación se determinó a partir de las definiciones de la norma, y consiste de varias categorías. Para la clasificación general nos guiamos por el Anexo IX de la norma.

Con respecto a la duración del uso, el dispositivo de medición se diseñó considerando un uso máximo de 24 horas, por lo que se dispone como equipo de **uso a corto plazo** (60´ a 31 días). Además clasifica como **producto activo para diagnóstico** (ya que requiere de una fuente de alimentación). Es importante destacar que, debido al punto 2.3, el programa informático y el firmware del producto también resultan incluidos en esta clasificación.

Con respecto a la clasificación específica, por ser producto activo de diagnóstico, aplica la *Regla 10* del punto 3.2, la cual indica que todos los productos activos de diagnóstico pertenecen a la clase IIa.

Una vez establecida la clasificación general, se utilizaron las tablas armonizadas de la norma 93/42/CE, para determinar las regulaciones pertinentes a nuestro equipo. Como nuestro método de medición utiliza una almohadilla constrictora, debemos cumplir con todas las normas pertinentes a requisitos para esfingomanómetros y para dispositivos de medición de la presión arterial. Las normas que aplican a estos equipos son las siguientes:

- **IEC 60601-1** Equipos electromédicos. Parte 1: Requisitos generales para la seguridad básica y funcionamiento esencial.
- **IEC 60601-1-1** Requisitos para la seguridad en sistemas Electromédicos.
- **IEC 60601-1-2** Norma colateral: Compatibilidad electromagnética. Requisitos y ensayos.
- **IEC 60601-1-4** Norma colateral: Sistemas electromédicos programables.
- **IEC 60601-1-6** Norma colateral: Aptitud de uso.
- **IEC 60601-2-30** Requisitos particulares para la seguridad, incluyendo la actuación esencial, de los equipos de vigilancia de la presión sanguínea, de funcionamiento automático, cíclico y no invasivo.
- **ISO 14971** Productos sanitarios. Aplicación de la gestión de riesgos a los productos sanitarios.
- **ISO 13485** Productos sanitarios. Sistemas de gestión de la calidad. Requisitos para fines reglamentarios.
- **ISO 1060-3** Esfigmomanómetros no invasivos. Parte 3: Requisitos suplementarios aplicables a los sistemas electromecánicos de medición de la presión sanguínea.
- **ISO 1060-4** Esfigmomanómetros no invasivos. Parte 4: Procedimientos de ensayo para determinar la exactitud del sistema total de esfigmomanómetros no invasivos automatizados.

De estos estándares, los que presentan mayor especificidad para nuestro producto, son la norma colateral IEC 60601-2-30 y las normas ISO 1060-3 y 1060-4. De éstas, la norma ISO 1060-3 aplica únicamente a equipos que realizan medición de la presión arterial mediante aplicación de un *cuff* a nivel del brazo, la muñeca o el muslo. Y la norma ISO 1060-4 se encuentra oficialmente retirada, sin embargo, sigue figurando entre los requerimientos de la norma 93/42/CE. El sitio oficial IEC indica que la conformidad con esta norma puede demostrarse utilizando la norma retirada<sup>13</sup>.

### Requisitos de la Norma IEC 60601 y sus colaterales.

Si bien la lista de requerimientos es extensa, muchos de estos son de aplicación general y no afectan severamente al diseño del dispositivo o sus características (si bien son de vital importancia, su aplicación no implica gastos o modificaciones que requieran de muchos recursos). En consecuencia, nos avocaremos a detallar los requisitos de performance esencial y de seguridad que implican características de diseño importantes.

Es importante caracterizar las diferencias entre seguridad básica y performance esencial. La norma general define:

- **Seguridad Básica:** Libre de RIESGOS inaceptables ocasionados por PELIGROS físicos, cuando los equipos electromédicos son utilizados bajo condiciones NORMALES y de PRIMERA FALLA.

---

<sup>13</sup> Información disponible en: <https://www.evs.ee/products/evs-en-1060-4-2004>

- **Performance Esencial:** Características de funcionamiento necesarias para asegurar el uso del equipo libre de RIESGOS inaceptables. Implican que estos equipos deben mantener ciertas características que, de ser degradadas o perdidas, conllevarían a un riesgo para el paciente, el usuario o el público general.

Para entender estas definiciones se debe considerar también, las definiciones siguientes:

- **Peligro:** Fuente potencial de DAÑO.
- **Daño:** Lesión física o daño a la salud de las personas o animales, o daño a la propiedad o al medio ambiente.
- **Situación Peligrosa:** Circunstancias en las cuales, las personas, la propiedad o el medio ambiente, se encuentran expuestas a uno o más PELIGROS.
- **Riesgo:** Combinación entre la probabilidad de ocurrencia de un DAÑO y la SEVERIDAD de dicho daño.
- **Severidad:** Medida de las posibles consecuencias de un PELIGRO.

Las primeras características requeridas para el equipo son aquellas pertinentes a la posibilidad de ocurrencia de una situación de primera falla. Se debe garantizar que las probabilidades de ocurrencia de una primera falla son despreciables, o bien que una primera falla puede ser detectada de forma inmediata durante la vida útil del equipo. Se deben utilizar componentes o materiales de alta integridad para aquellos cuya falla pudiera ocasionar un riesgo inaceptable.

Alimentación del Dispositivo: El equipo se debe alimentar con la tensión de línea, ya que debe mantener un funcionamiento continuo durante muchas horas. Se debe preparar el equipo para mantener un funcionamiento adecuado suponiendo una variación en la alimentación de 90-110% de la tensión nominal. Considerando alimentación de línea de 220 VAC, estos límites son 198 y 242 VAC, respectivamente. La potencia nominal no debe variar más que un 10% bajo estas condiciones. Se debe realizar una medición de corriente con un amperímetro de tipo R.M.S. y obtener la potencia en V.A. como el producto de esta corriente estable y la tensión de alimentación.

### Ensayos de Tipo:

Los ensayos de Tipo son aquellos de naturaleza destructiva, destinados a evaluar las características de seguridad y performance esencial establecidos por la norma. Estos ensayos deben ser realizados bajo condiciones controladas, en un número suficientemente representativo de muestras y bajo las situaciones mas desfavorables para cada ensayo, de las indicadas en las instrucciones de uso del dispositivo. Las características mas desfavorables pertinentes serán todas aquellas funcionales y variables que el fabricante ha considerado en su análisis de riesgo, desde condiciones ambientales (humedad, temperatura,

iluminación, etc.) hasta características eléctricas (variaciones de la alimentación de línea, condiciones de falla, etc.).

Previo a la ejecución de los ensayos de tipo, el equipo debe ser sometido al pre-acondicionamiento de humedad. Se lo debe mantener a una temperatura  $T$  entre 24 y 32 °C durante cuatro horas, para luego introducirlo a un gabinete cerrado con 93% de humedad relativa durante 48 horas. Una vez terminado el pre-acondicionamiento, se continúa con los ensayos de tipo.

Los ensayos de tipo incluyen pruebas de rigidez dieléctrica de aislaciones, de capacidad de corriente e impedancia de las conexiones de tierra, de las corrientes de fuga, de accesibilidad a las partes activas, de integridad de las envolturas frente a la acción del agua y elementos físicos.

### Clasificaciones del Dispositivo:

Como ya mencionamos, la clasificación del equipo implica diferentes requisitos de performance. Entre las clasificaciones inherentes a los componentes del mismo, debemos destacar:

- Clasificación de la protección contra choque eléctrico: Como el equipo requiere de un blindaje adecuado, debido a los requisitos de integridad de las señales utilizadas, se dispone que el equipo sea de **CLASE I**.
- Clasificación de las partes aplicadas: Tipo B, BF o CF, siendo o no aptas para descarga de desfibriladores.
- Clasificación de la protección de las envolturas: Número IP de dos dígitos, donde el primero es la protección contra elementos físicos y el segundo indica la protección contra el acción del agua.
- Clasificación según método de esterilización, modo de operación (continuo) y aptitud para funcionamiento en atmósferas ricas en oxígeno.

### Protección contra choque eléctrico:

De acuerdo con la norma general, los ensayos de tipo de seguridad eléctrica deben realizarse bajo condiciones normales. Que incluyen todos estos condicionantes (simultáneamente):

- Presencia de señales de entrada o salida de cualquier otro equipo de funcionamiento eléctrico, descrito o permitido en las condiciones de operación del equipo. En nuestro caso, sería con la PC conectada y en funcionamiento normal.
- Inversión de las conexiones de alimentación, Fase y Neutro.
- Cortocircuito de cualquier aislación que no cumple con los requerimientos de la sección 8.8.
- Cortocircuito de todas las distancias de aire o de separación eléctrica que no cumplen con los requerimientos de la sección 8.9.

- Apertura de todas las conexiones de tierra de protección que no cumplan con los requerimientos de la sección 8.8 y de todas las tierras funcionales.

Además, los ensayos deben realizarse y cumplir sus requerimientos bajo todas las situaciones posibles de PRIMERA FALLA. Si se dieran dos de estas situaciones de forma simultánea, el equipo estaría en condición de SEGUNDA FALLA, por lo que se deben revisar separadamente. Dichas situaciones incluyen:

- Cortocircuito de cualquier aislación que cumpla con los requerimientos de la sección 8.8.
- Cortocircuito y apertura de cualquier componente (que no sean de Alta Integridad), que esté conectado en paralelo con una aislación, una distancia regulada de aire o distancia de separación (a no ser que se pueda demostrar que su cortocircuito o apertura no constituya una falla para dicho componente).
- Apertura de cualquier tierra de protección que cumpla con los requerimientos de la sección 8.8.
- Apertura de cualquier cable de alimentación.
- Movimiento de cualquier componente cuyo desplazamiento pueda ocurrirse durante su vida útil normal y desconexión de cualquier conductor o cable cuya separación pueda generar una situación peligrosa (de acuerdo con el análisis de riesgo del equipo).

Un factor muy importante que debemos considerar, es que hemos diseñado el dispositivo para ser utilizado con una computadora para la adquisición, procesamiento e ilustración de señales, así como para el control del funcionamiento. En consecuencia, el punto 8.2.1 requiere que la computadora sea considerada como parte del equipo electromédico, de modo que se le aplicarían todos los requerimientos pertinentes. Esto impediría un cumplimiento de la norma, debido a que no utilizamos una computadora que cumpla estos requisitos. Para su correcta comercialización, nuestro dispositivo debería ser integrado con un equipamiento de monitoreo multiparamétrico, o bien, contar con las funciones desarrolladas por la PC en un componente integral del equipo.

El equipo debe contar con dos medios de protección, que eviten la conexión o circulación de corrientes peligrosas a través del paciente o el usuario del equipo. Como este equipo se diseñó de Clase I, estos medios de protección son, una aislación dieléctrica y una conexión de tierra de protección.

Los requerimientos para las conexiones a de tierra de protección son genéricos para todos los equipos. En particular destacamos que, la impedancia entre el terminal de tierra del cable de alimentación y cualquier punto del equipo indicado como “protegido por conexión de tierra de protección”, no debe exceder los 200 m $\Omega$ . Esta propiedad se verifica con el ensayo tipo correspondiente, circulando una corriente de 25 A. y 50 Hz entre los puntos pertinentes, durante 5

segundos. Nuestro equipo no requiere de conexión de tierra funcional ni de conexión para barras de equipotencialización.

#### Norma IEC 60601-2-30.

La norma colateral -2-30 de las IEC 60601 establece los requerimientos de seguridad y performance esencial específicos para equipos de medición de la presión sanguínea de forma cíclica, automática y no invasiva. Esta norma es utilizada principalmente para determinar los requerimientos de equipos de oscilometría, pero aplica de igual forma a nuestro dispositivo.

Esta norma define requerimientos similares a los dispuestos por la norma general, como seguridad contra choque eléctrico, rigidez dieléctrica, seguridad contra incendios, etc. Pero también establece requerimientos específicos de exactitud mínima y consideraciones especiales para los vasos sanguíneos a los que se les aplican presiones externas.

Además de los requerimientos de exactitud y seguridad, también se establecen características necesarias para los brazaletes/almohadillas, los conectores y mangueras neumáticas, tiempos máximos de operación y ciclos de operación automáticos (estos últimos, destinados especialmente a equipos de oscilometría).

## Capítulo 9: Mediciones y Resultados.

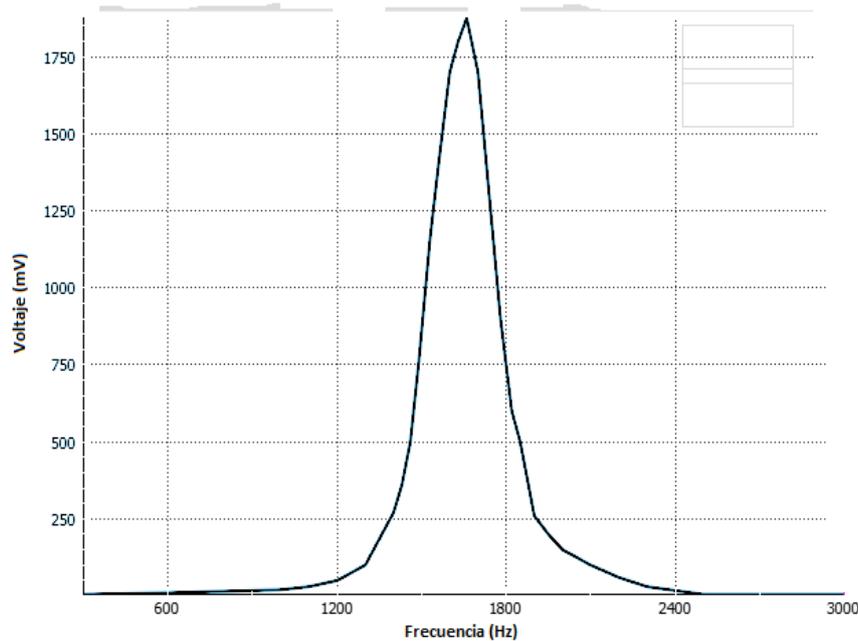
Una vez finalizada la construcción del prototipo, se procedió con las pruebas de funcionamiento, de los distintos componentes y del sistema integrado.



Ilustración 57: Fotografías del Prototipo finalizado. Arriba: Vista superior del interior del gabinete. Abajo Izquierda: Panel frontal de conectores. Abajo Derecha: Gabinete cerrado con dispositivo de medición conectado.

### Respuesta en Frecuencia del Filtro Pasa Banda.

Se midió la respuesta en frecuencia del filtro pasa banda MFB de 1600 Hz. Se utilizó un generador de funciones para generar una señal senoidal de frecuencia variable y amplitud de 1 Vpp. Se midió la salida del filtro con un osciloscopio digital, con la PCB de señales alimentada por una fuente de 6 Vdc. Se obtuvo la siguiente salida:

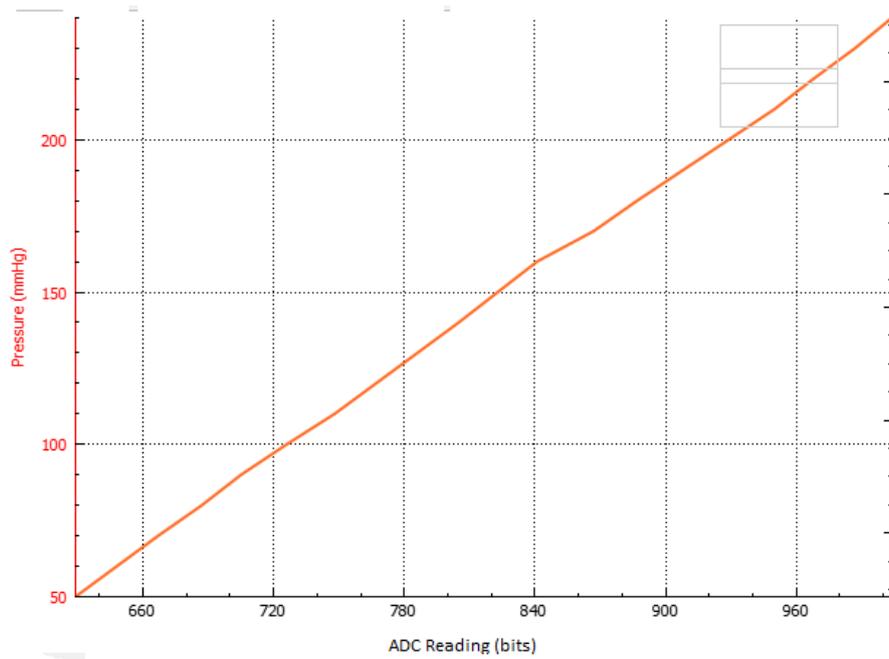


**Ilustración 58:** Respuesta en frecuencia del filtro pasa banda de 1600 Hz.

Por lo que el filtro presenta una ganancia mayor que la prevista en la banda de paso (ganancia de 5 dB a 1600 Hz), pero la salida resultó estable, manteniendo la forma senoidal y con una respuesta en frecuencia aceptable.

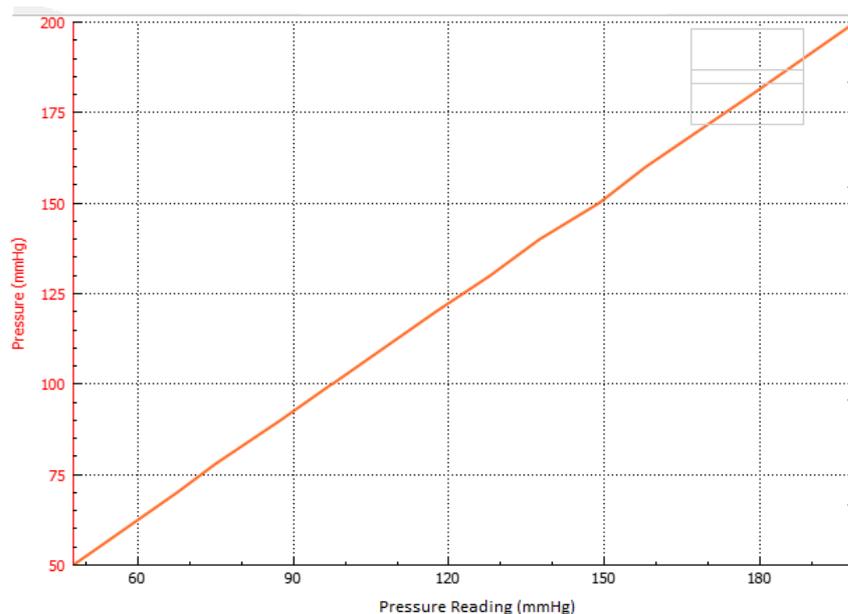
### Calibración del Sensor de Presión.

Se realizaron dos calibraciones del sensor de presión. En primer lugar se realizó una calibración presurizando el sistema neumático funcional con una manguito manual y se comparó la medición con una columna de mercurio. Se obtuvo la siguiente recta de calibración:



**Ilustración 59: Calibración del sensor de presión con columna de mercurio.**

A partir de la recta se estableció la escala de la señal digital como 0.5 mmHg/bit. El *offset* de la señal depende de la temperatura ambiente, de la presión atmosférica, entre otros. Por lo que, se estableció una rutina de calibración en el programa de PC que elimina este *offset*. Luego se realizó una comparación de la medición obtenida en varios puntos estables de presión, por el sistema calibrado, con la medición de un Simulador de Signos Vitales *Fluke ProSim 8*. Se obtuvo la siguiente correlación:



**Ilustración 60: Comparación de mediciones de presión con Simulador Fluke.**

La presión determinada por el prototipo resultó 2.5 mmHg inferior a la del Simulador Fluke, en todos los puntos y de forma constante. Una parte de este error se debió a un factor de -2 bits que se le aplicaba a la calibración del *offset*, a fin de evitar que se obtuviera una presión de -0.5 mmHg en la medición. Una vez eliminado este factor, el error fue constante de -1.5 mmHg, el cual se encuentra dentro de la exactitud calculada para el sensor en el Capítulo 3.

### Comparación con Monitor Multiparamétrico.

Para evaluar la exactitud y precisión del proceso de medición foto-oscilométrico, y la exactitud de las mediciones subsecuentes, se realizaron múltiples mediciones simultáneas con el prototipo desarrollado y un Monitor Multiparamétrico Phillips MP20 que realiza medición de presión no invasiva Braquial, mediante el método oscilométrico.

Se obtuvieron repetidas mediciones con ambos equipos en un paciente sano de sexo masculino y 25 años de edad, durante un período de 15 minutos. Se registraron los valores resultantes de presión sistólica, media y diastólica.

Como se utilizó el mismo miembro para las mediciones de ambos equipos, se realizaron de forma secuencial. Primero se obtuvo la medición del Monitor Multiparamétrico, y luego la del prototipo. Esta última se realizó en forma completa, con el método foto-oscilométrico seguido de 30 segundos de medición con el método de compensación volumétrica. Sin embargo, solo se utilizaron las muestras obtenidas por este segundo método en los últimos 20 segundos de medición, ya que el método oscilométrico fallaba en el correcto establecimiento de la referencia. Como se puede observar en la Ilustración 61, debido a un error de programación, se subestimaba el nivel de referencia y la función de recalibración tardaba cerca de 10 segundos en corregir este *setpoint*.

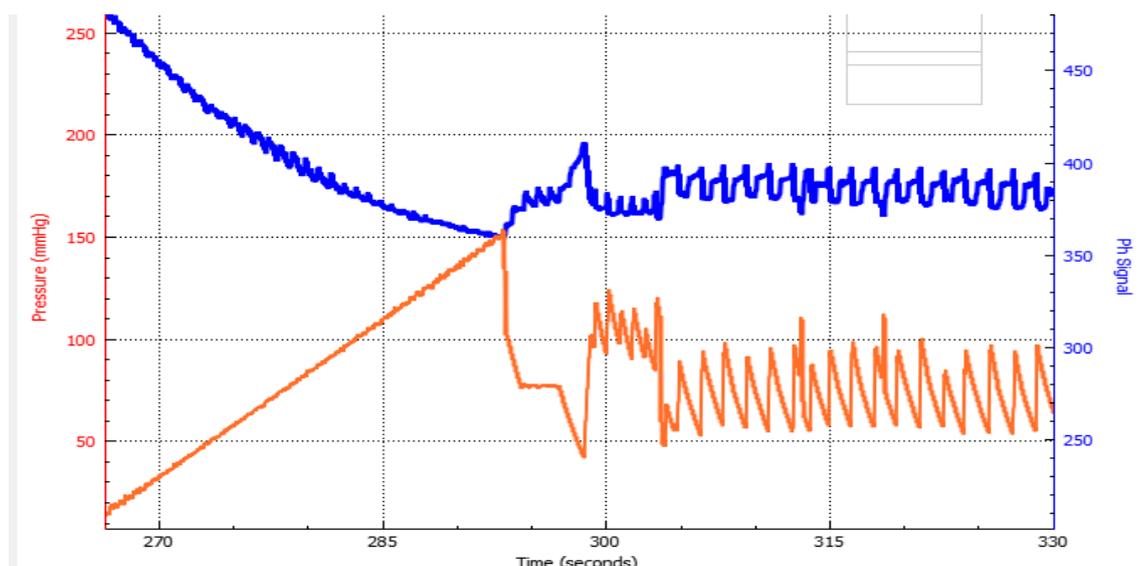


Ilustración 61: Medición con método foto-oscilométrico, seguida de recalibración de la referencia.

*Medición 1: Falange Basal Media.*

La primera medición se realizó con el brazalete de oscilometría y la mano a nivel del ventrículo izquierdo, en sedestación. Se utilizó el dedo medio. La medición del monitor determinó:

- Presión Sistólica: 97 mmHg.
- Presión Media: 78 mmHg.
- Presión Diastólica: 69 mmHg.

Los resultados obtenidos por el dispositivo se muestran a continuación.

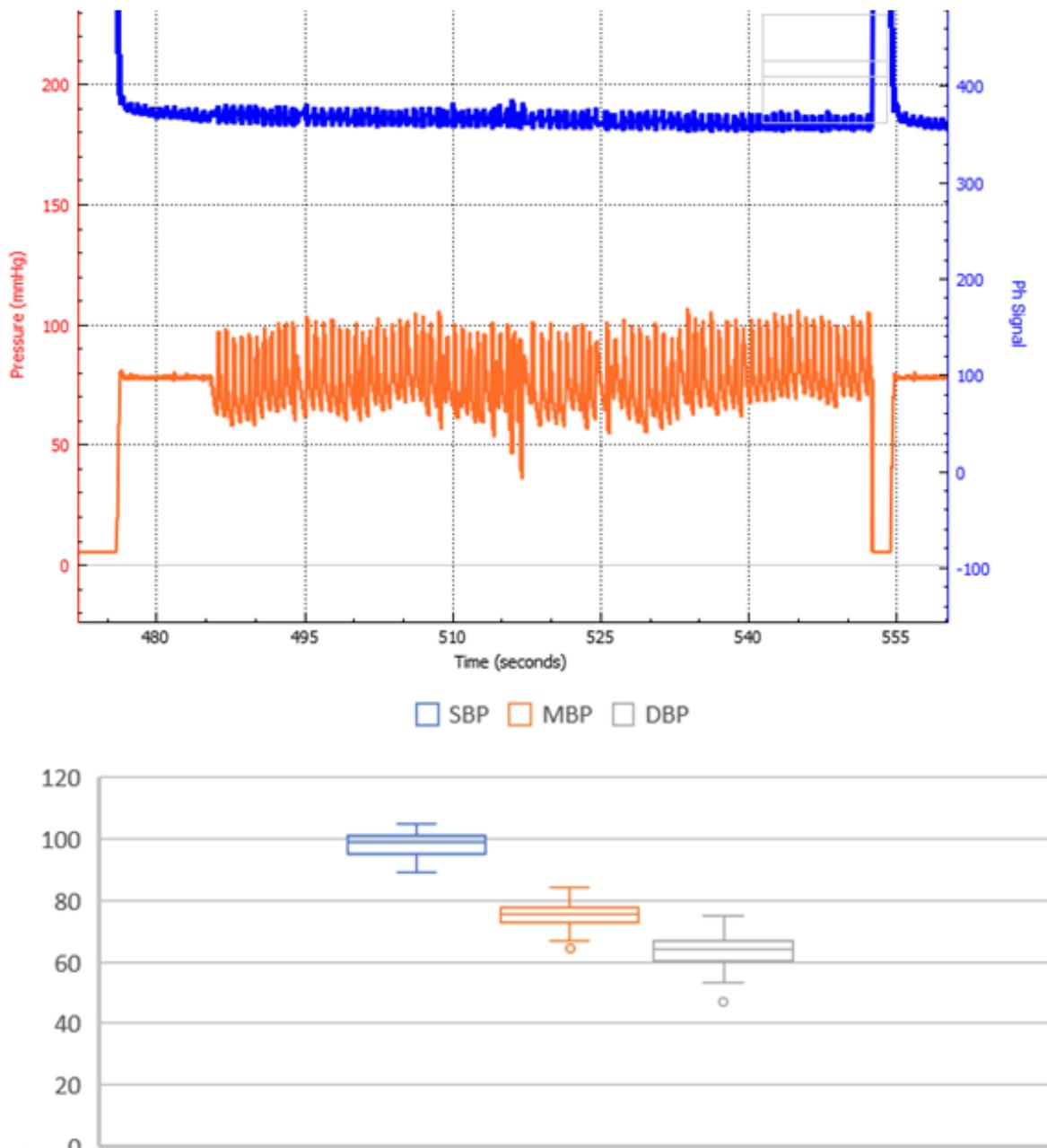


Ilustración 62: Medición 1, falange basal media. Arriba: Período de medición utilizado. Abajo: Distribución de puntos de presión sistólica, media y diastólica durante el período de medición.

*Medición 2: Falange Basal Anular.*

La segunda medición se realizó con el brazalete de oscilometría y la mano a nivel del ventrículo izquierdo, en sedestación. Se utilizó el dedo anular. La medición del monitor determinó:

- Presión Sistólica: 95 mmHg.
- Presión Media: 72 mmHg.
- Presión Diastólica: 63 mmHg.

Los resultados obtenidos por el dispositivo se muestran a continuación.

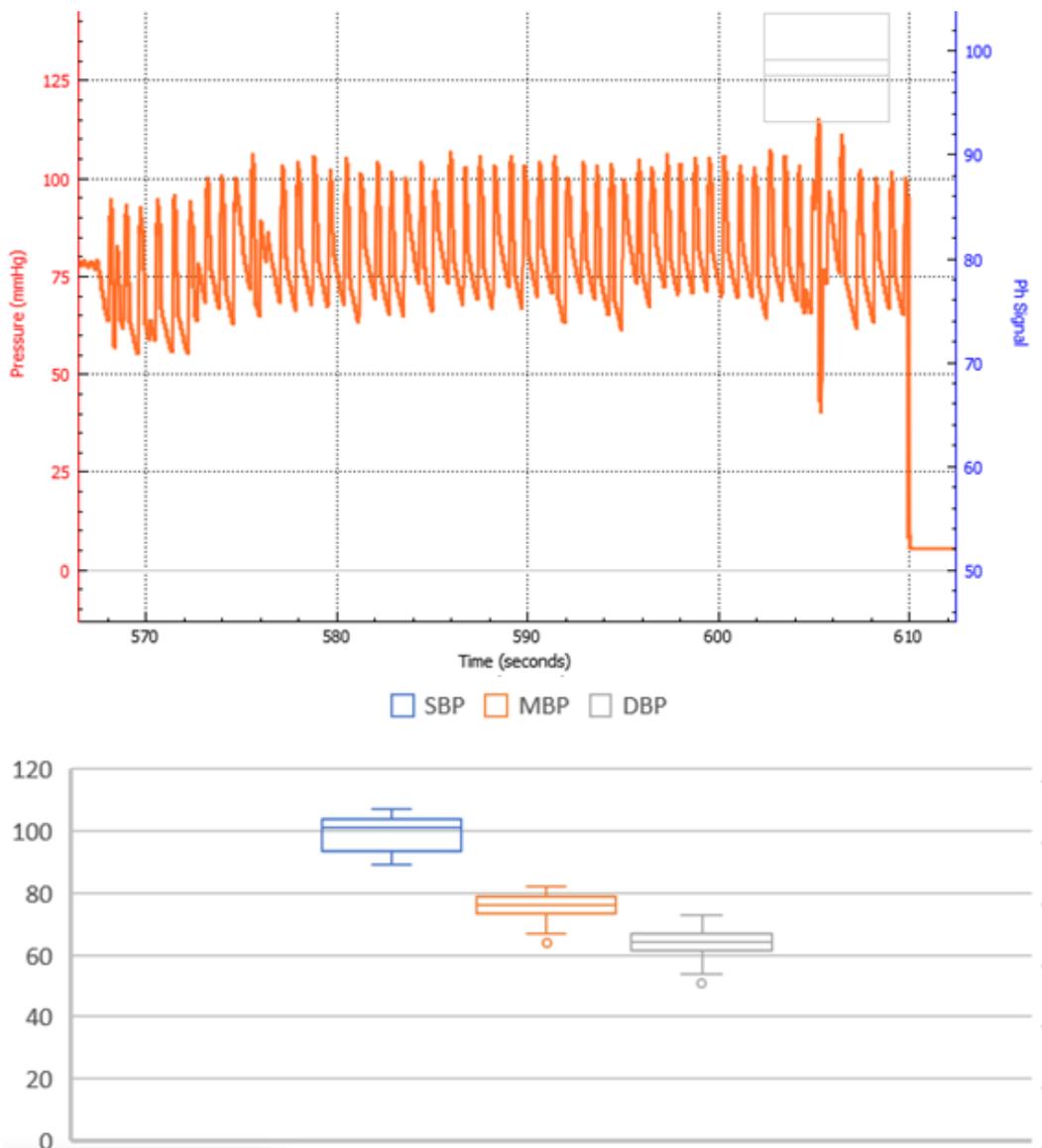


Ilustración 63: Medición 2, falange basal anular. Arriba: Período de medición utilizado. Abajo: Distribución de puntos de presión sistólica, media y diastólica durante el período de medición.

### Medición 3: Falange Distal del Pulgar.

Se realizó una tercera medición, con dos diferencias. Por un lado, aún en sedestación, se colocaron el brazo y la mano izquierda a igual altura, pero 7 centímetros por debajo del nacimiento aórtico, con el fin de presentar una presión superior en el segmento de medición. Y por el otro, se posicionó el prototipo sobre el extremo del pulgar, en lugar de sobre la falange basal, ya que en este punto, la amplitud de la señal pulsátil de FPG era superior. Los resultados obtenidos por el monitor fueron:

- Presión Sistólica: 121 mmHg.
- Presión Media: 94 mmHg.
- Presión Diastólica: 81 mmHg.

Los resultados obtenidos con el prototipo se muestran a continuación.

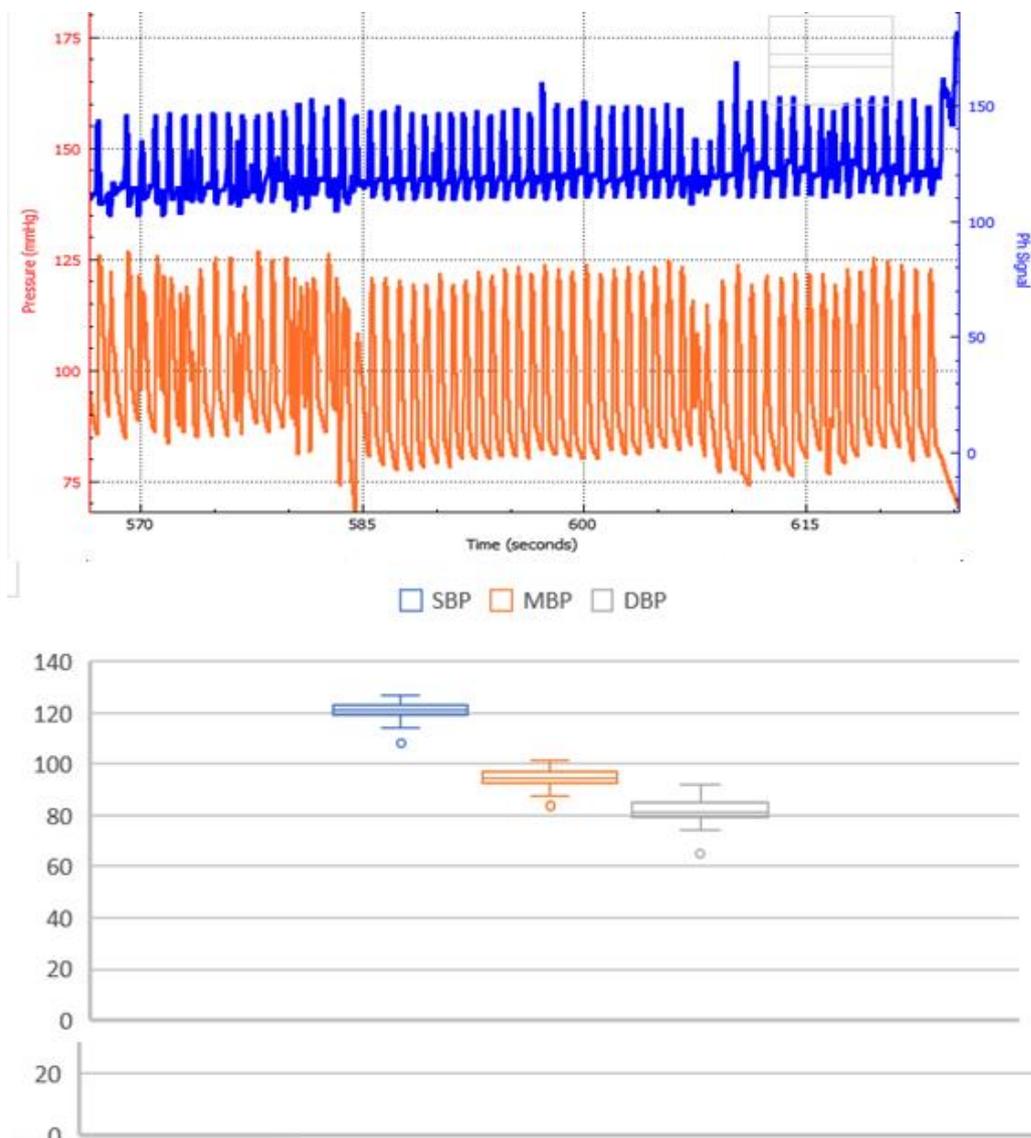
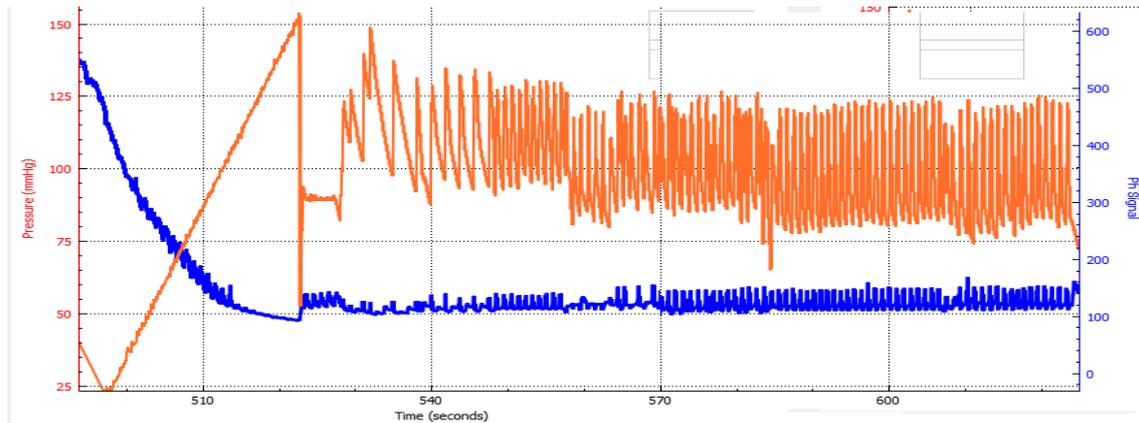


Ilustración 64: Medición 3, extremidad del dedo pulgar. Arriba: Período de medición utilizado. Abajo: Distribución de puntos de presión sistólica, media y diastólica durante el período de medición.

Al evaluar las mediciones obtenidas con el prototipo de compensación volumétrica se observa una exactitud aceptable. Es importante apreciar que, cuando la señal pulsátil fotopleletismográfica es de mayor amplitud, la precisión de las mediciones del prototipo es mayor. Esto se debe a que la resolución de la información de la que dispone el sistema de control es mayor, y por ende, el proceso de compensación volumétrica resulta mas estable. En la Ilustración 65 se observa claramente esta mejoría.



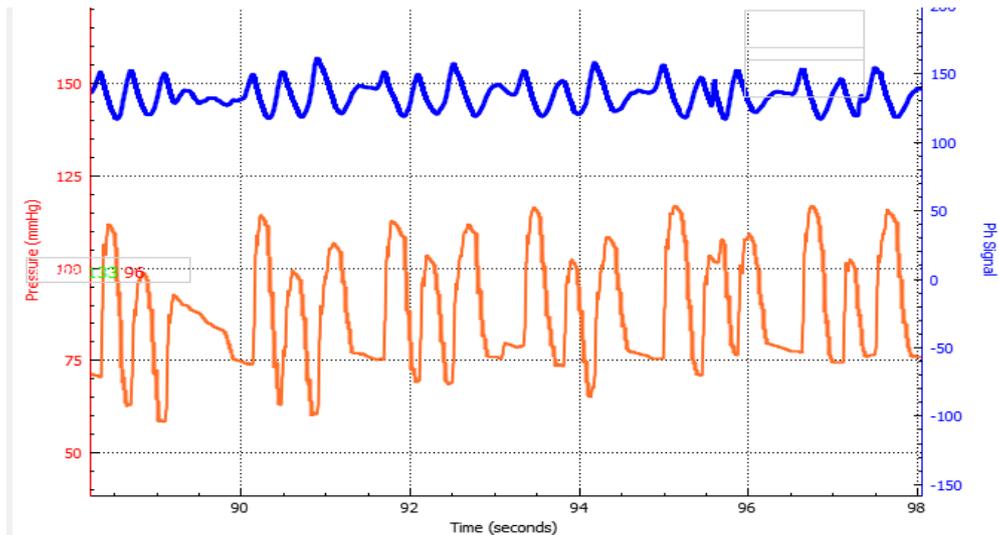
**Ilustración 65: Medición de presión. Se destaca la variación en la amplitud de la señal fotopleletismográfica a medida que se calibra el punto de referencia.**

Se puede apreciar como, a medida que se recalibra la referencia fotopleletismográfica de VCM, se incrementa la amplitud de esta misma señal, y consecuentemente, la medición de presión resultante es mucho mas estable.

### Formas de Onda: Respuesta Temporal del Sistema.

Se realizaron múltiples mediciones posicionando la almohadilla de diferentes formas, con pacientes, ganancias de controladores, funciones de predicción y referencias diferentes. A medida que se realizaban diferentes mediciones, se trataba de conseguir la mayor estabilidad y la mejor respuesta temporal, de modo tal que la forma de onda obtenida se asemejara lo mas posible a la forma de onda teórica de la presión arterial. Se destacarán las principales variables que se modificaron y su efecto en la forma de onda de la medición.

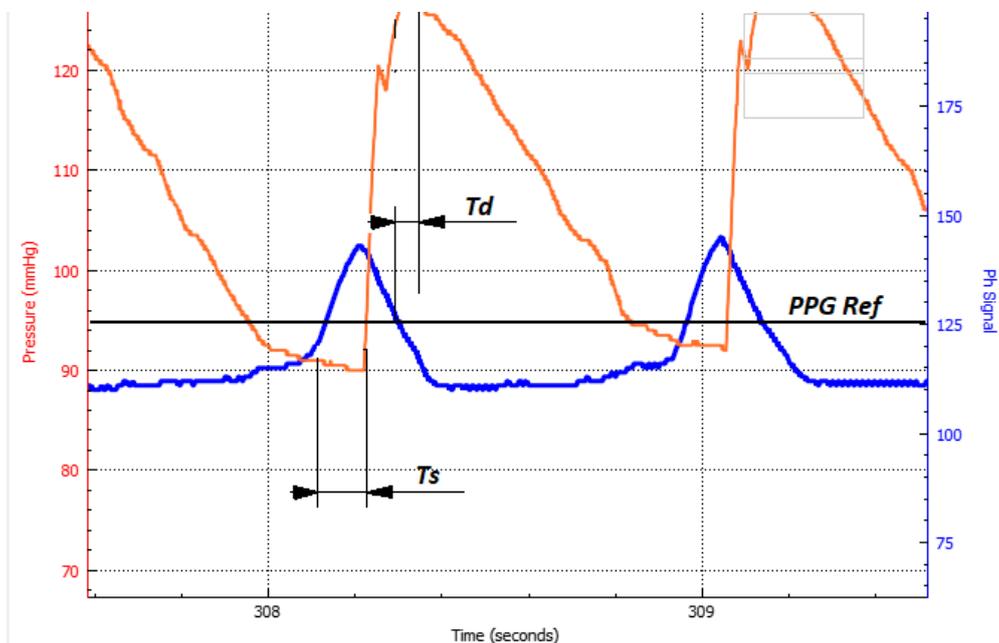
En primer lugar, se trabajó sobre la sintonización de los controladores PID. La ganancia del controlador de presurización presentó mucha flexibilidad. Pero el controlador y la válvula de escape demostraron una gran sensibilidad. Pequeñas modificaciones en la ganancia proporcional del controlador resultaron en grandes desviaciones de performance de la electroválvula, pudiendo variar desde oclusión total a inestabilidad en saltos reducidos. Además, esta ganancia debió reajustarse cada vez que se encendía el equipo y durante los primeros minutos de funcionamiento, ya que la electroválvula modifica su funcionamiento a medida que se calienta. La Ilustración 66 muestra el resultado de una medición en el que la ganancia proporcional del controlador de escape era muy excesiva.



**Ilustración 66:** Inestabilidad de la medición provocada por un exceso de ganancia en el controlador de la válvula de escape.

Una vez sintonizados los controladores, se observó la respuesta temporal del sistema. Para esto, definimos dos parámetros para caracterizar esta misma.

- Inercia Estática: Tiempo entre el inicio del pulso en la señal fotopletismográfica y la respuesta de presurización compensatoria.
- Inercia Dinámica: Tiempo entre la anulación de la señal error y la anulación de la pendiente de la señal de presión.



**Ilustración 67:** Respuesta temporal del sistema de compensación. Donde,  $T_s$  es el tiempo de la inercia estática,  $T_d$  es el tiempo de la respuesta dinámica y PPG Ref es el nivel de referencia fotopletismográfico.

La inercia dinámica fue casi constantemente de 100 milisegundos, mientras que la inercia dinámica osciló entre 40 y 75 milisegundos. Para reducir la inercia estática, y en consecuencia mejorar la forma de onda, se implementó una función de predicción, en la que, bajo frecuencia cardíaca estable, se sumaba un impulso a la señal error, 100 milisegundos antes del próximo pulso de la señal fotopletomográfica. Sin embargo, la variación de la frecuencia cardíaca instantánea resultó ser mucho mayor a la esperada. Por lo que los pulsos provocados por la función de predicción se presentaban a intervalos muy variados, por lo que la señal se volvía inestable. En la Ilustración 68 se observa un pulso de presión arterial en el que la predicción se adelantó considerablemente al latido cardíaco.

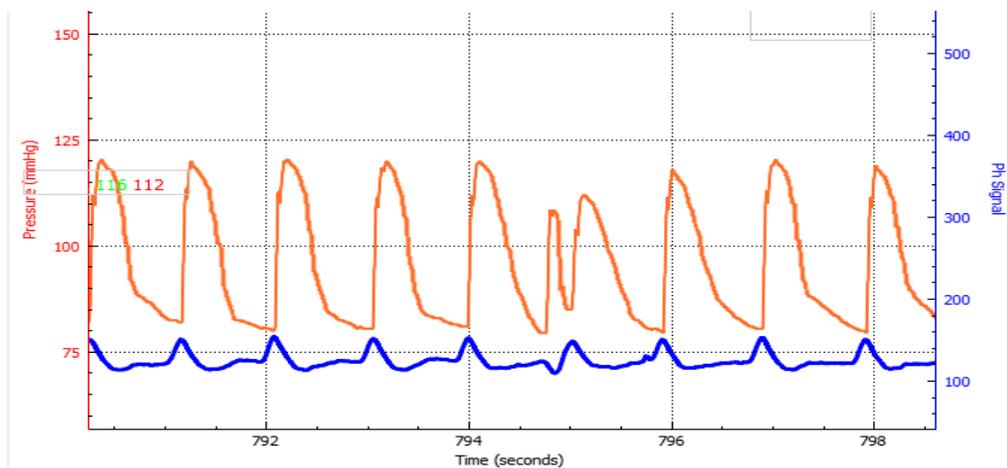
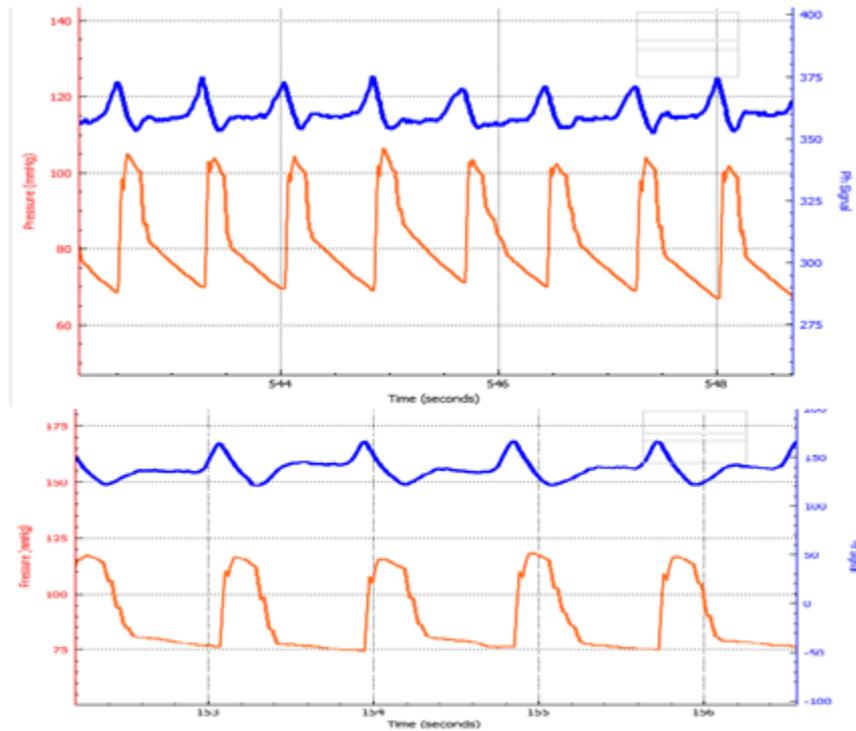


Ilustración 68: Predicción de pulso arterial inexacta.

Por último, se observó la diferencia en la forma de onda cuando el circuito neumático funcional se encuentra hermético y cuando este presenta una fuga muy pequeña.



**Ilustración 69: Efecto de la hermeticidad de las fugas en la almohadilla.**  
**Arriba: Medición con fuga de aire. Abajo: Medición con almohadilla herméticamente cerrada.**

En la Ilustración 69 se observan dos mediciones. En la primera, la almohadilla oclusora presentaba una pequeña fuga de aire constante. Como resultado, la forma de onda de presión arterial obtenida se correlaciona con la pendiente de presión resultante de esta fuga, cuando el error de la señal no es suficiente para generar una apertura considerable de la válvula de escape. En la segunda, la almohadilla se selló herméticamente, y por ende, cuando este error es demasiado bajo, la presión se mantiene casi constante, en consecuencia la forma de onda se ve saturada.

Para evitar el problema de la saturación de la forma de onda y para no depender de la pendiente constante debida a una fuga de aire, se modificaron los parámetros de los controladores y se obtuvieron mediciones como se muestran a continuación.

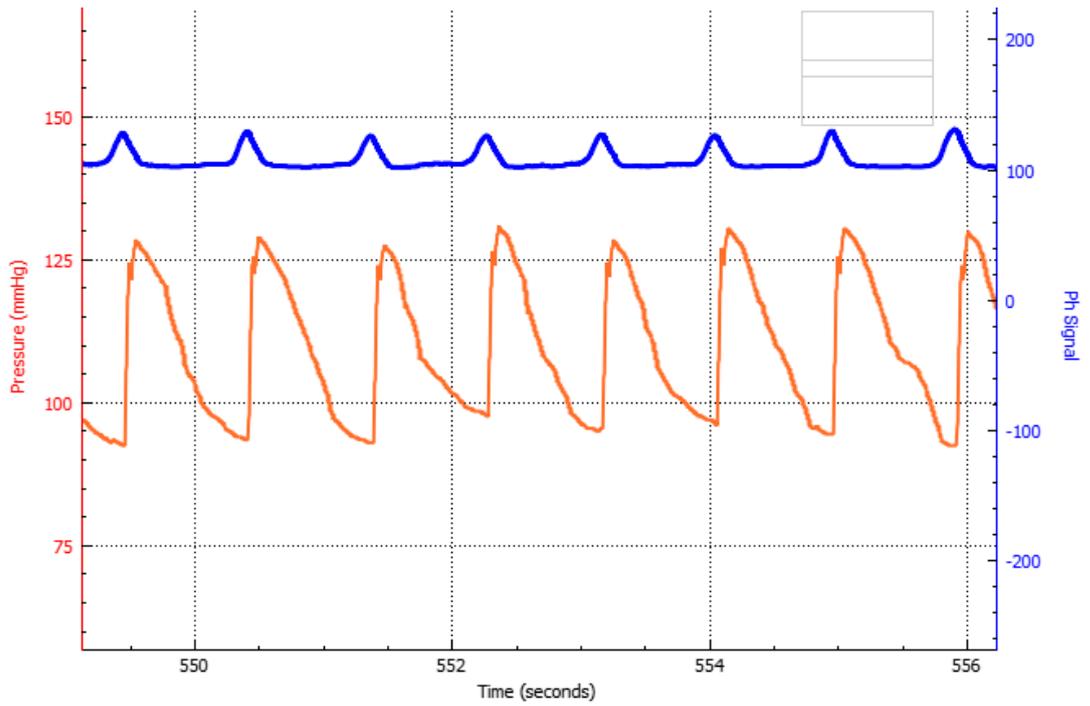


Ilustración 70: Medición de presión arterial calibrada.

## Conclusiones y Observaciones.

Durante el desarrollo de este Proyecto Integrador se enfrentaron dificultades de diferente naturaleza, desde errores de diseño hasta limitaciones severas de materiales y componentes. Se buscó solventar estas dificultades con técnicas diferentes y se pudo completar el desarrollo del prototipo exitosamente.

Si bien las mediciones obtenidas por el dispositivo no pudieron ser correctamente evaluadas (debido a la indisponibilidad de un equipo adecuado para comparar), se pudieron obtener mediciones con una forma de onda muy similar a la descrita en la bibliografía. En la Ilustración 71 se observa una de estas mediciones en las que las condiciones de medición fueron establecidas correctamente. Estas curvas presentan defectos, pero se consideró que estos defectos podrían ser corregidos en proyectos futuros.

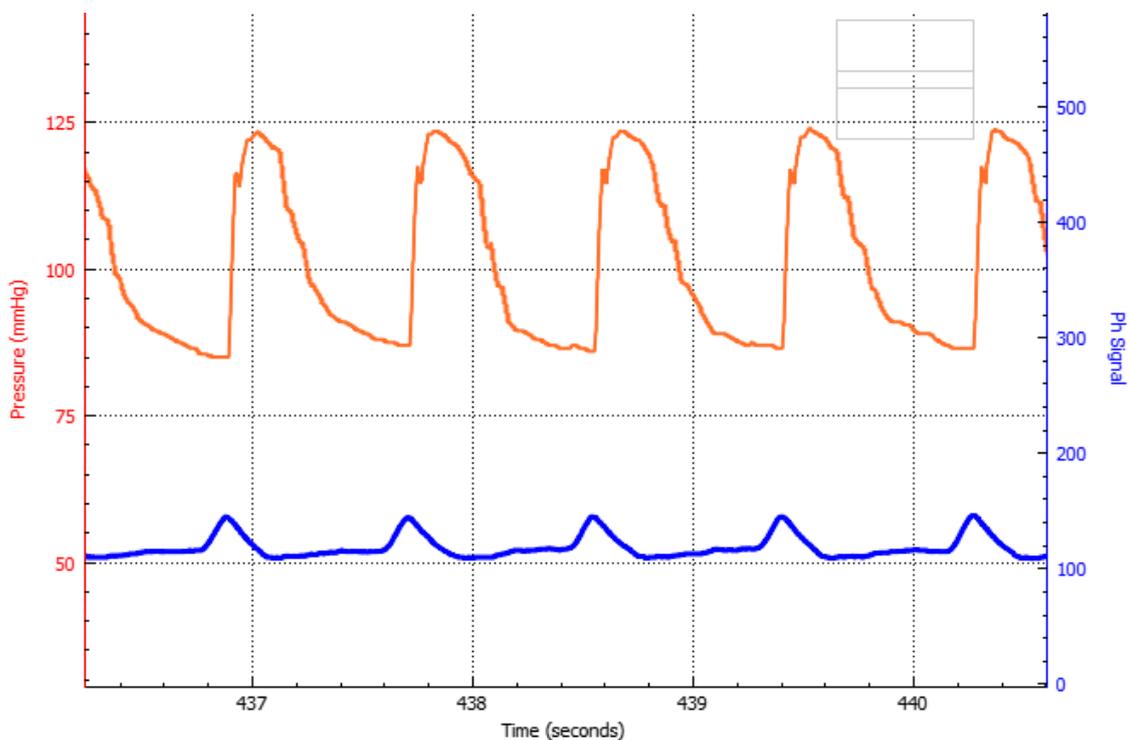


Ilustración 71: Formas de onda obtenidas con el prototipo bajo condiciones adecuadas.

### Sobre la Exactitud de las Mediciones.

De las comparaciones realizadas con la medición del monitor multiparamétrico podemos concluir que el prototipo puede determinar correctamente los niveles de presión sistólica, media y diastólica, con exactitud y precisión adecuada a los objetivos de este Proyecto Integrador. Es importante destacar que

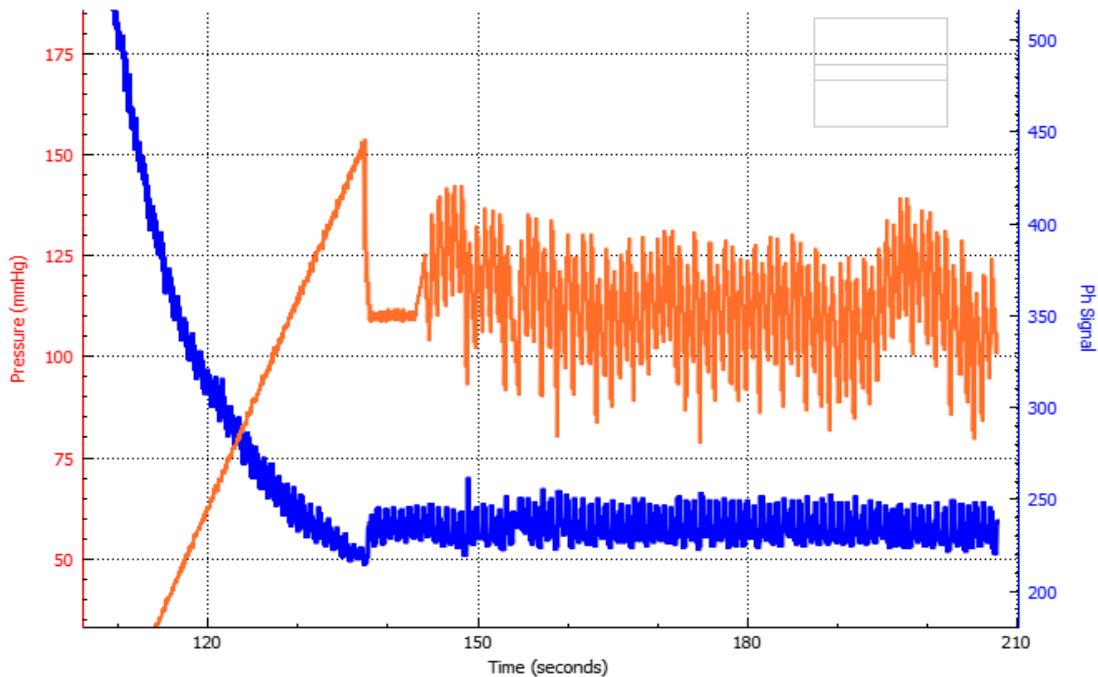
estos niveles dependen casi totalmente, de la correcta determinación del punto de referencia. Para establecer correctamente esta referencia es necesario que la señal fotopleletismográfica presente una amplitud suficiente, ya que de otro modo, tanto la calibración por el método foto-oscilométrico, como la estabilidad del método de compensación volumétrica, se ven comprometidos. Para una medición adecuada y estable se debe garantizar una amplitud de señal pulsátil mínima de 20 bits de resolución, a presión arterial media.

Otra consecuencia de una amplitud adecuada de la señal fotopleletismográfica, es la estabilidad del proceso de medición por compensación volumétrica. De los resultados de medición podemos observar que la medición realizada con el dedo pulgar, en la que la señal fotopleletismográfica tenía mayor amplitud, presenta una mayor precisión que las mediciones con la falange basal de los dedos medio o anular.

Para garantizar esta amplitud mínima es importante destacar dos variables, el posicionamiento adecuado de la almohadilla (lateralmente sobre la falange basal del dedo anular o medio) y las condiciones de la misma.

El Látex no solo presenta las dificultades que mencionamos anteriormente, sino que también resulta ser un material sucio. Es decir, que se degrada enormemente con el uso. De modo tal que su opacidad y su fragilidad se incrementan gradualmente. En consecuencia, a no ser que se renueve la almohadilla, la señal fotopleletismográfica será cada vez menor y el Látex será más propenso a dañarse por sobrepresión o fatiga. En consecuencia, se obtendría una mejora considerable si se utilizara un material diferente al Látex, como una goma de Uretano transparente. Para solucionar el problema de la señal degradada debido a la opacidad del Látex, se estableció la ganancia de amplificación en exceso y se incorporó un potenciómetro para regular fácilmente la intensidad luminosa del Led IR del fotopleletismógrafo.

Por otro lado, es importante destacar que el mecanismo de recalibración mejora la exactitud de la medición, pero presenta una gran desventaja. Para procesos prolongados de medición, en los que la presión arterial puede variar con el tiempo si se aplica constantemente este proceso no será posible detectar estos cambios, ya que la referencia óptica se establece contra la presión media de calibración. En la Ilustración 72 se observa un proceso de medición en el que se indujo una variación en la presión arterial (al bajar y subir la mano utilizada para la medición), que no hubiera sido detectada si se hubiera recalibrado el punto de referencia fotopleletismográfico. Nótese la constancia de la señal fotopleletismográfica durante la variación de presión. En consecuencia, se debe bloquear la activación de este mecanismo luego de 1 o 2 minutos de iniciado el proceso de medición, o bien, se debe recalibrar regularmente la medición con el método fotooscilométrico.



**Ilustración 72: Variación de presión arterial detectada con el método de compensación volumétrica.**

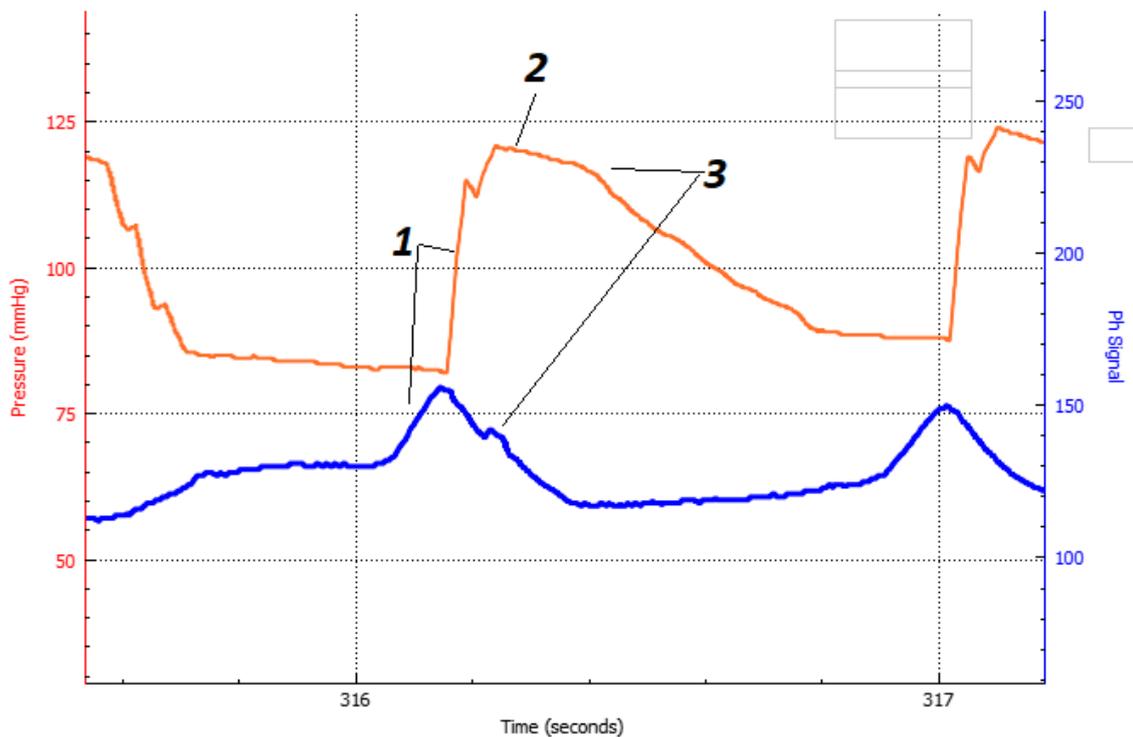
En consecuencia, se debe bloquear la activación de este mecanismo luego de 1 o 2 minutos de iniciado el proceso de medición, o bien, se debe recalibrar regularmente la medición con el método fotooscilométrico.

### **Sobre la Respuesta Temporal del Sistema.**

Como se mencionó en el capítulo anterior, el tiempo de respuesta del sistema de control, entre el inicio del pulso arterial y del pulso de presión compensatoria, es de aproximadamente 100 milisegundos.

Esta velocidad de respuesta es demasiado lenta como para que el equipo pueda compensar instantáneamente las diferencias de presión. Si todas las publicaciones precedentes [29, 31, 35] presentan un cierto delay en la compensación, de modo tal que la señal fotopleletismográfica nunca se vuelve constante, la respuesta de estos dispositivos les permite una reacción mas acelerada del sistema.

Una de las principales consecuencias de un sistema lento es la deformación de la onda de presión arterial obtenida. Como los puntos característicos de la curva de presión arterial son de una amplitud considerablemente inferior a la amplitud de la curva misma, es normal que estos puntos sean integrados por el sistema de control y desaparezcan de la curva.



**Ilustración 73:** Defectos de medición debidos a la respuesta lenta del sistema. 1: *Pendiente sistólica abrupta debido a la inercia estática.* 2: *Pico de la curva agudo debido al cierre abrupto de las válvulas.* 3: *Incisura dicrótica atenuada por la inercia dinámica del sistema.*

En la Ilustración 73 se observan los principales problemas causados por esta respuesta del sistema. Por una lado, como la inercia estática provoca que la apertura valvular se efectúa cuando la señal error ya es muy elevada, la pendiente ascendente será siempre abrupta. En consecuencia, cualquier información patológica contenida en esta característica de la curva (como una pendiente aminorada por una estenosis valvular aórtica) se perderá.

El segundo problema radica en la naturaleza de conmutación de las válvulas. Como el cierre no es proporcional y como el sistema es lento, el pico de presión es muy agudo y la pendiente de presión cambia de forma instantánea.

Por último, como la incisura dicrótica es de amplitud reducida y como el sistema demora en establecer la referencia, cuando la incisura dicrótica intraarterial se presenta, pueden darse dos situaciones. Que la presión de la almohadilla es muy superior a la del pico dicrótico, en consecuencia, la incisura dicrótica desaparece totalmente de la gráfica obtenida. O, que la inercia dinámica del sistema le impida compensar correctamente esta variación de la señal fotopletomográfica y por ende, la incisura dicrótica obtenida por el prototipo es una versión atenuada de la real.

Si bien la presión instantánea obtenida por el prototipo resulta ser una versión integrada de la presión arterial real, la forma de onda general se conserva correctamente si se sintonizan correctamente los controladores PID. Para esto,

es importante realizar una prueba de funcionamiento prematura a cada medición, de modo tal que podamos estar seguros de que los parámetros de control son los adecuados.

Para mejorar la forma de onda obtenida primero se debe determinar el error instantáneo de la curva obtenido en cada segmento de la misma. Para esto, existen dos posibles soluciones, ambas radican en la comparación de la medición con otros equipos. Por un lado, se podría obtener la presión invasiva mediante cateterización de un paciente y obtener la presión no invasiva con este prototipo de forma simultánea. Y, por otro lado, se podría comparar la medición con un equipo de tonometría aplanatoria correctamente calibrado y utilizado por un profesional capacitado. Cabe destacar que no fue posible conseguir un equipo de Tonometría Aplanatoria adecuado para realizar esta comparación durante el desarrollo de este Proyecto Integrador.

Una vez caracterizado el error de medición, se deben implementar las mejoras adecuadas a las características de este error. En particular, se recomienda la implementación de un proceso de predicción adecuado, posiblemente mediante la implementación de una señal de ECG para sincronizar la predicción. Otra mejora posible es la implementación de un controlador neumático de mayor calidad. Estimamos que, para obtener una respuesta adecuada, se debería de contar con un sistema de control de presión con una inercia máxima de 10 milisegundos.

### **Sobre los Materiales.**

Se observaron dos grandes problemas durante el desarrollo de este proyecto integrador con respecto a los materiales y componentes utilizados. Por un lado, las dificultades presentadas por el Látex, tanto su capacidad dieléctrica como sus características mecánicas. Como se mencionó anteriormente, se podría obtener un diseño más confiable del prototipo de medición si se utilizara una goma de uretano elástica y transparente.

Por otro lado, las dificultades del sistema de control de presión, formado principalmente, por las electroválvulas. Un sistema de regulación ideal para nuestro problema, consiste de un sistema neumático funcional constituido por un contenedor hermético, cuya presión se regula por dos electroválvulas proporcionales 2/2 (o una única electroválvula proporcional 3/3), una de escape atmosférico y la otra, de presurización conectada a una línea de presión constante. Si las electroválvulas presentan una respuesta temporal adecuada, con tiempos de respuesta e inercias despreciables, y la presión de alimentación es constante, conocida y con una capacidad de caudal muy por encima del caudal necesario, entonces este sistema de control sería muy simple de modelar y calibrar.

Sin embargo, se encontraron dos grandes problemas para la implementación de dicho sistema ideal de control. Por un lado, la disponibilidad comercial

de electroválvulas proporcionales adecuadas, y por el otro, la utilidad objetivo del prototipo.

Como este tipo de electroválvulas proporcionales de caudal bajo y tensiones de control reducidas son utilizadas, casi exclusivamente, por la industria biomédica, en nuestro país, la demanda por estos instrumentos es prácticamente nula. En consecuencia, las empresas que se dedican a la manufactura de dichas válvulas no las ofrecen al mercado local, o las atienden a tiempos de entrega y costos de envío excesivos (hasta 200 euros por unidad). En consecuencia, no fue posible utilizar alguna de estas electroválvulas para el desarrollo de nuestro prototipo.

En consecuencia, las electroválvulas ideales para nuestra aplicación son ofrecidas por *Festo (Alemania)* y por *Camozzi (Italia)*, con las dificultades comentadas en el párrafo anterior. En particular, se destaca la línea de controladores de presión proporcional FESTO VEAB, ya que, si bien las hojas de datos no indican la respuesta temporal de estos controladores, dicha respuesta puede ser medida y caracterizada con facilidad.<sup>14</sup>



**Ilustración 74: Controlador de Presión FESTO de la serie VEAB.**<sup>14</sup>

Por último, se encontraron grandes dificultades para obtener una señal buena señal fotopletoisomográfica debido a la inestabilidad del amplificador de transimpedancia. Ya sea que los componentes utilizados no hayan sido los ideales, o que el diseño y la construcción de los circuitos incorporasen grandes capacitancias parásitas, este problema fue muy difícil de corregir. Si bien todas las publicaciones que se encontraron de implementación de fotopletoisomógrafos utilizaban esta configuración de amplificación para obtener la conversión de

---

<sup>14</sup> Imagen extraída de [https://www.festo.com/cat/es\\_es/products\\_VEAB](https://www.festo.com/cat/es_es/products_VEAB)

corriente a voltaje, es posible que sea conveniente explorar otras posibilidades. En particular, se recomienda como mejora el reemplazo del amplificador de transimpedancia por un amplificador de instrumentación con una red de polarización de baja impedancia para el fotodiodo.

### **Resumen de las Mejoras Propuestas.**

- Evaluación del error de medición mediante comparación con un equipo de Tonometría Aplanatoria o con Medición de Presión Invasiva.
- Implementación de un controlador de presión proporcional con tiempos de respuesta reducido.
- Implementación un proceso de predicción de los pulsos, con el fin de mejorar la respuesta del sistema de control. Posiblemente, mediante la utilización de una señal de ECG.
- Reemplazo del amplificador de transimpedancia por un amplificador de instrumentación con polarización del fotodiodo.
- Utilización de un material diferente al Látex para la almohadilla oclusora. Rediseño de la misma para mejorar su ergonomía y simplificar su utilización.

## Bibliografía y Referencias

- [1] I. M. Moxham, «Understanding Arterial Pressure Waveforms,» *Southern African Journal of Anaesthesia and Analgesia.*, vol. 9:1, pp. 40-42, 2003.
- [2] R. R. Townsend, H. R. Black, J. A. Chirinos, P. U. Feig y K. C. Ferdinand, «Clinical Use of Pulse Wave Analysis: Proceedings from a symposium sponsored by North American Artery,» *Journal of Clinical Hypertension.*, vol. 17, nº 7, pp. 503 - 514, 2015.
- [3] L. A. Geddes, *Handbook of Blood Pressure Measurement*, New York: The Humana Press Inc., 1991.
- [4] L. Testut y A. Latarjet, *Compendio de Anatomía Descriptiva*, vol. Tomo II, México: Nacional, 1949.
- [5] A. Guyton y J. E. Hall, «Unidad III. Capítulo 9.,» de *Tratado de Fisiología Médica.*, pp. 283-317.
- [6] A. Guyton y J. E. Hall, *Tratado de Fisiología Médica.*, México, 2016.
- [7] A. Guyton y J. E. Hall, «Unidad IV. Capítulo 15.,» de *Tratado de Fisiología Humana*, p. 462.
- [8] Pawlina y Ross, *Histología: Texto y Atlas color con Biología Celular y Molecular*. 5ta Ed., Editorial Médica Panamericana., 2007.
- [9] M. F. O'Rourke y T. Yaginuma, «Waveform Reflections and the Arterial Pulse,» *Arch Intern Med*, nº 144, pp. 366-371, 1984.
- [10] J. R. M. Bruner, «Comparison of direct and indirect methods of measuring arterial blood pressures.,» *Med Instrum*, vol. 15:1, pp. 11-21, 1981.
- [11] J. A. Posey y L. A. Geddes, «The meaning of the point of maximum oscillations in cuff pressure in the indirect measurement of blood pressure.,» *Cardiovascular Res. Cent. Bulletin*, vol. 8, nº 1, pp. 15-25, 1969.
- [12] M. Gourdeau, R. Martin y Y. Lamarche, «Oscillometry and Direct Blood Pressure: a comparative clinical study during deliberate

hypotension.,» *Can Anaesth Soc Journal*, vol. 33, nº 3, pp. 300-307, 1986.

- [13] G. Mauck, C. R. Smith, L. A. Geddes y J. D. Bourland, «The meaning of the point of maximum oscillations in cuff pressure in the indirect measurement of blood pressure. Part II.,» *Journal Biomech Eng.*, nº 102, pp. 28-33, 1980.
- [14] R. W. Ware y C. J. Laenger, «Indirect recording of the entire arterial pressure wave.,» *Proc. Ann. Conf. Eng. Biol. Med.*, vol. 19th, p. 51, 1966.
- [15] M. B. Kardon, B. W. Ware, H. P. Stegall y W. L. Kemmerer, «Indirect measurement of blood pressure using Doppler-shifted ultrasound.,» de *Digest of the 7th Inter. Conf. on Med. and Biol. Eng.*, Stockholm, Sweden., 1967.
- [16] K. Yamakoshi, H. Shimazu, M. Shibata y K. A., «New oscillometric method for indirect measurement of systolic and mean arterial pressure in the human finger.,» *Med. & BioL Eng. & Comput.*, vol. 20, nº 1, pp. 314 - 318, 1982.
- [17] K. Yamakoshi, «Chapter 5: Non-Invasive Cardiovascular Hemodynamic Measurements,» de *Sensors in Medicine and Health Care.*, Wiley-VCH Verlag GmbH & Co., 2004, pp. 107-161.
- [18] H. Shimazu, A. Kawarade y H. Ito, «Electric impedance cuff for the indirect measurement of blood pressure and volume elastic modulus in human limb and finger arteries.,» de *Medical & Biological Engineering & Computing*, 1989.
- [19] M. Sharma, K. Barbosa, V. Ho, D. Griggs, T. Ghirmai, S. Krishnan, T. Hsiai y J. C. Chiao, «Cuff-less and continuous blood pressure monitoring: A Methodological review.,» *MDPI Technologies*, 2017.
- [20] S. S. Thomas, V. Nathan, C. Zong, K. Soudarapandian, X. Shi y R. Jafari, «BioWatch: A NI wrist based blood pressure monitor that incorporates Training techniques for posture and subject variability.,» *JBHI*, vol. 94, pp. 1-10, 2015.
- [21] S. Puke, T. Suzuki, K. Nakayama, H. Tanaka y S. Minami, «Blood pressure estimation from pulse wave velocity measured on the chest.,» *Conf Proc IEEE Eng Med Biol Soc*, 2013.

- [22] J. M. Padilla, E. J. Berjano, J. Saiz, L. Facilla, P. Diaz y S. Mercé, «Assessment of Relationships between Blood Pressure, Pulse Wave Velocity and Digital Volume Pulse,» *Computers in Cardiology*, vol. 33, pp. 893-896, 2006.
- [23] H. Sorvoja, *Noninvasive Blood Pressure Pulse Detection and Blood Pressure Determination.*, Oulu, Finland: Faculty of Technology, University of Oulu., 2006.
- [24] G. Greiwe, P. A. Tariparast, C. Behem, M. Petzoldt, L. Herich, C. J. Trepte, D. A. Reuter y S. A. Haas, «Is applanation tonometry a reliable method for monitoring BP in morbidly obese patients undergoing bariatric surgery?,» *British Journal of Anaesthesia.*, vol. 116, nº 6, pp. 790-796, 2016.
- [25] H. Shirer, «Blood Pressure Measuring Methods.,» *IRE Transactions on Bio-Medical Electronics*, pp. 116-126, 1962.
- [26] J. Peñaz, «Photoelectric measurement of blood pressure, volume and flow in the finger,» de *Digest 10th Int. Conf. Med. Biol. Eng.*, Dresden, Germany., 1973.
- [27] K. Yamakoshi, H. Shimazu y T. Togawa, «Indirect measurement of instantaneous arterial blood pressure in the human finger by the vascular unloading technique.,» *IEEE Trans. Biomed. Eng.*, Vols. %1 de %2BME-27, pp. 150-155, 1980.
- [28] K. Wesseling, «Finapres, continuous noninvasive finger arterial pressure based on the method of Peñaz.,» de *Blood Pressure Measurements: New Techniques in Automatic and 24-hour Indirect Monitoring.*, 1990, pp. 161-172.
- [29] Imholz, Wieling, v. Montfrans y Wesseling, «Fifteen years experience with finger arterial pressure monitoring: assessment of the technology.,» *Cardiovascular Research Journal.*, vol. 38, pp. 605-616, 1998.
- [30] T. Birch, «Continuous Non-Invasive Blood Pressure Measurements.,» de *7th Mathematics in Medicine Study Group. University of Southampton.*, Southampton., 2007.
- [31] H. A. Aitken, J. G. Todd y G. N. C. Kenny, «Comparison of the Finapres and Direct Arterial Pressure monitoring during profound

- hypotensive anaesthesia.,» *British Journal of Anaesthesia.*, vol. 67, pp. 36-40, 1991.
- [32] G. A. Rongen, W. J. W. Bos, J. Lenders y G. A. van Montfrans, «Comparison of intra-brachial and finger blood pressure in healthy elderly volunteers.,» *American Journal of Hypertension.*, vol. 8, pp. 237-248, 1995.
- [33] M. Nakagawara y K. Yamakoshi, «Development of a finger cuff unit for a non-invasive blood pressure monitoring system based on the volume compensation method.,» *Jap. J. Med. Electr. & Biol. Eng.*, vol. 38, pp. 283-290, 2000.
- [34] A. Kwarada, H. Shimazu, H. Ito y Y. K., «Ambulatory monitoring of indirect beat-to-beat arterial pressure in human fingers by volume-compensation method.,» *Med. & Biol. Eng. & Comp.*, vol. 29, pp. 56-62, 1992.
- [35] K. Matsumura, T. Yamakoshi, P. Rolfe y K. Yamakoshi, «Advanced Volume-Compensation Method for Indirect Finger Arterial Pressure Determination: Comparison with Brachial Sphygmomanometry.,» *Trans. on Biomedical Engineering.*, vol. 64, nº 5, pp. 1131-1137, 2017.
- [36] Y. Song, S. Gao, A. Ikarashi y K. Yamakoshi, «A new cuff unit for measuring instantaneous blood pressure at the finger artery by local pressurization.,» de *3rd International Conf. on Bioinformatics & Biomed. Eng.*, Beijing, China., 2009.
- [37] M. Petzoldt, «Stroke volume determination using transcadiopulmonary thermodilution and arterial pulse contour analysis in severe aortic valve disease.,» *Intensive Care Medicine*, vol. 39, nº 4, pp. 601-611, 2013.
- [38] H. N. Sabbah y P. D. Stein, «Valve origin of the aortic incisura.,» *American Journal of Cardiology.*, vol. 41, nº 1, pp. 32-38, 1978.
- [39] S. Swarup, K. B. Akash y S. Gyanesh, «Fingertip Pulse Wave (PPG signal) Analysis and Heart Rate Detection.,» *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, nº 9, pp. 404-409, 2012.

- [40] D. J. Faber, «Oxygen Saturation-Dependent Absorption and Scattering of Blood.,» The American Physical Society, Amsterdam, The Netherlands., 2004.
- [41] J. Caldwell, «Transimpedance Amplifiers: What Op Amp Bandwidth do I Need?,» Texas Instruments., 2015.
- [42] E. Bogatin, «Chapter 2: Time and Frequency Domains.,» de *Signal and Power Integrity - Simplified.*, Prentice Hall, 2018.
- [43] J. Caldwell, «1 MHz, Single Supply, Photodiode Amplifier Reference Design.,» Texas Instruments., 2014.
- [44] S. Faruque, «Chapter 2: Amplitude Modulation.,» de *Radio Frequency Modulation Made Easy.*, The Author(s), 2017, pp. 17-34.
- [45] P. Horowitz y W. Hill, «Chapter 6: Filters.,» de *The Art of Electronics 3rd Ed.*, Cambridge University Press, 2015, pp. 391-425.
- [46] S. Winder, *Analog and Digital Filter Design. 2nd Ed.*, Newnes, Elsevier Science., 2002.
- [47] H. Zumbahlen, «Chapter 8: Analog Filters.,» de *Basic Linear Design.*, Analog Devices, Inc., 2007.
- [48] B. Carter, «A Single-Supply Op-Amp Circuit Collection.,» Texas Instruments., 2000.
- [49] K. Kaida Bin y W. Mohamad Daud, «Dielectric Properties of Natural Rubber Latex at frequencies from 200 MHz to 2500 MHz.,» *J. not. Rubb. Res.*, vol. 7, nº 4, pp. 281-289, 1992.
- [50] W. Storr, «MOSFET as a Switch.,» Electronics Tutorials, 2015. [En línea]. Available: [https://www.electronics-tutorials.ws/transistor/tran\\_7.html](https://www.electronics-tutorials.ws/transistor/tran_7.html). [Último acceso: 2018].
- [51] W. Storr, «The BJT Transistor as a Switch.,» 2015. [En línea]. Available: [https://www.electronics-tutorials.ws/transistor/tran\\_4.html](https://www.electronics-tutorials.ws/transistor/tran_4.html). [Último acceso: 2018].
- [52] R. Gaviño Hernandez, *Introducción a los sistemas de control: Conceptos, aplicaciones y simulación con MATLAB.*, Naucalpán de Juárez, México.: Prentice Hall, 2010.

- [53] H. Johan Astrom, «PID Control,» de *Control System Design.*, 2002, pp. 216-254.
- [54] V. Bobál, G. Macháček y R. Prokop, «Tuning of Digital PID Controllers based on Ziegler-Nichols Method.,» *IFAC Proceedings Volumes*, vol. 30, nº 21, pp. 145-150, 1997.
- [55] E. S. Gopi, «Chapter 2: Infinite Impulse Response (IIR) Filter.,» de *Multi-Disciplinary Digital Signal Processing.*, Springer International Publishing., 2018, pp. 43-78.
- [56] *Directiva 93/42/CEE*, Europeas, Consejo de las Comunidades, 1993.

# Anexos

## Anexo N° 1: Esquemáticos y PCBs.

Circuito de Control y Comunicación.

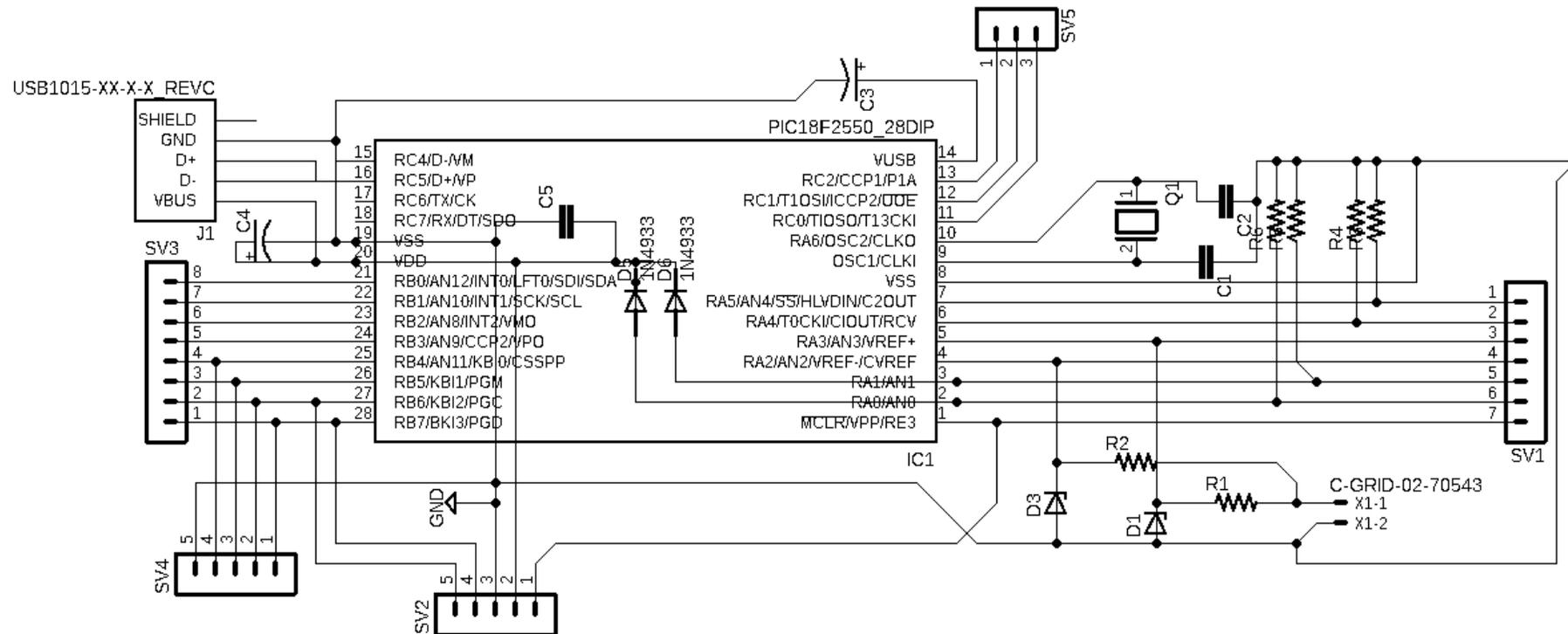


Ilustración 75: Esquemático del circuito de comunicación y control. Diseñado con Autodesk's EAGLE.

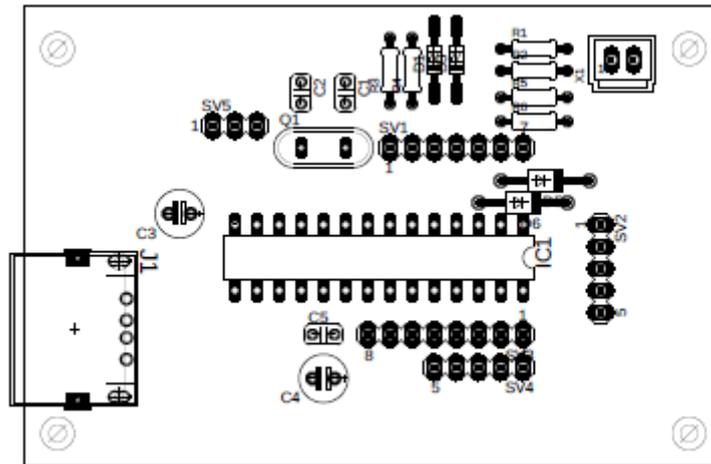


Ilustración 76: PCB de comunicación. Top Layer. Diseñado con Autodesk's EAGLE.

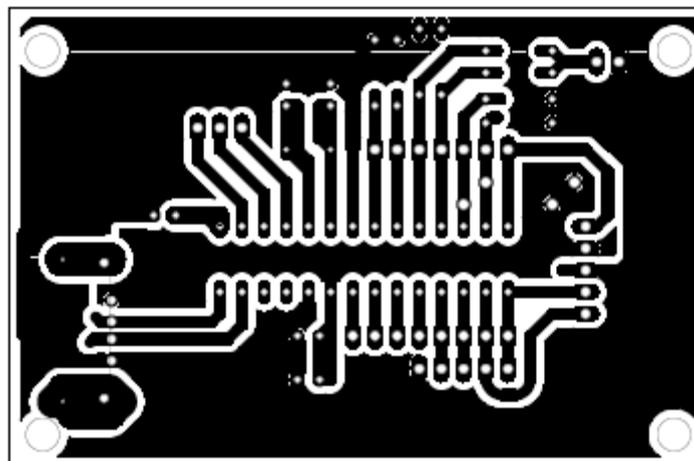


Ilustración 77: PCB de comunicación. Bottom Layer. Diseñado con Autodesk's EAGLE.

Circuito de Señales: Amplificación y Filtros.

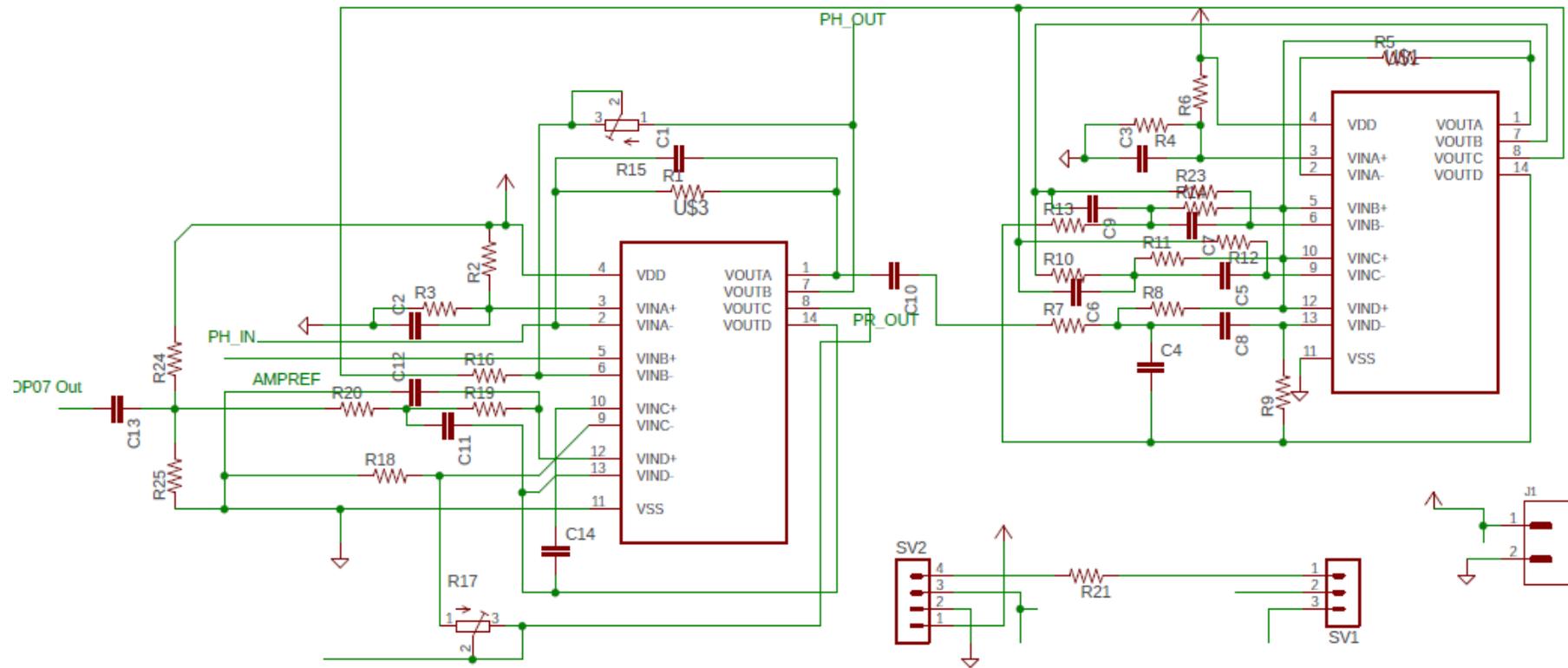


Ilustración 78: Esquemático del circuito de señales, amplificación y filtros. Diseñado con Autodesk's EAGLE.



Circuito de Potencia y Control de Válvulas.

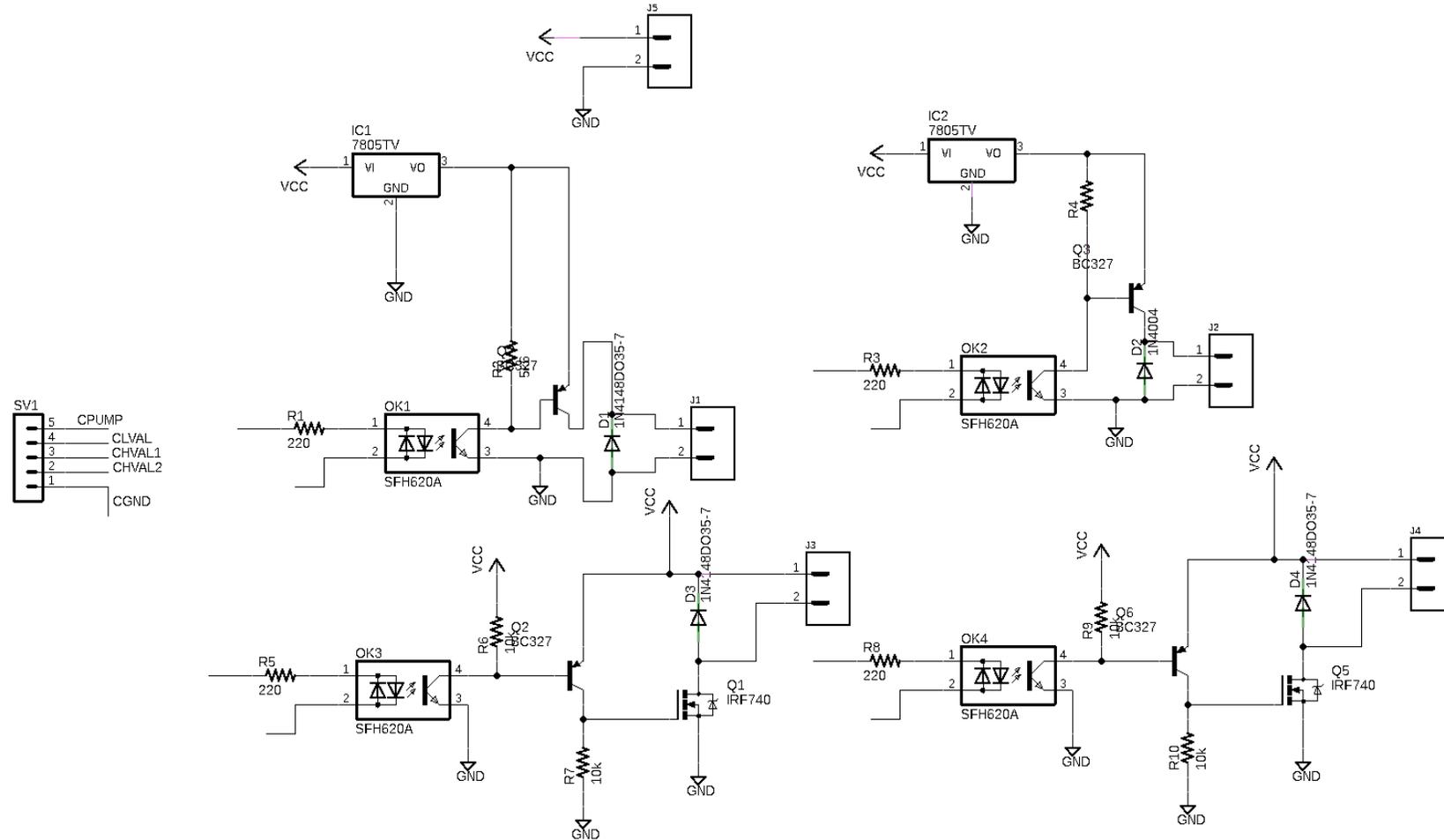


Ilustración 81: Esquemático del circuito de potencia y control de válvulas. Diseñado con Autodesk's EAGLE.

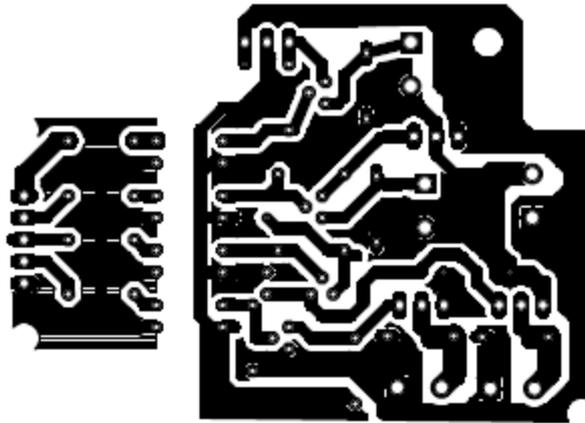


Ilustración 82: PCB del circuito de potencia. Bottom Layer. Diseñado con Autodesk's EAGLE.

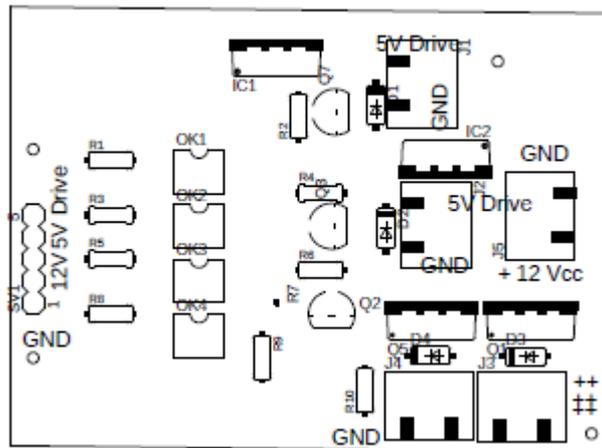


Ilustración 83: PCB del circuito de potencia. Top Layer. Diseñado con Autodesk's EAGLE.

## Anexo N° 2: Firmware del Microcontrolador.

### USB MAIN.H

```
#include <18F2550.h>
#device ADC=10

#FUSES NOWDT           //No Watch Dog Timer
#FUSES WDT128          //Watch Dog Timer uses
1:128 Postscale
#FUSES NOBROWNOUT     //No brownout reset
#FUSES NOMCLR         //Master Clear pin used
for I/O
#FUSES NOLVP           //No low voltage prgming,
B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOXINST        //Extended set extension
and Indexed Addressing mode disabled (Legacy mode)
#FUSES VREGEN
#FUSES HSPLL
#FUSES CPUDIV1
#FUSES PLL5
#FUSES USBDIV

#use delay(clock=48000000)
//#use FIXED_IO( C_outputs=PIN_C1,PIN_C2 )

#bit SPEN = 0xFAB.7
#byte TRISC = 0xF94
#byte TMR0 = 0xFD6
#byte TMR1L = 0xFCE
#byte TMR1H = 0xFCF
#bit TC7 = 0xF94.7
#bit TC6 = 0xF94.6
#bit IRDrive = 0xF81.4
#bit IRTTris = 0xF93.4
#bit ADFlag = 0xF9E.6

#define T0SC 2

#define CHECK_BIT(var,pos) ((var) & (1<<(pos)))
#define USB_USE_FULL_SPEED TRUE
#define USB_HID_DEVICE TRUE
#define USB_CONFIG_HID_TX_SIZE 64
#define USB_CONFIG_HID_RX_SIZE 8
#define TX_SIZE USB_CONFIG_HID_TX_SIZE
#define RX_SIZE USB_CONFIG_HID_RX_SIZE
#define USB_CONFIG_PID USB_CONFIG_PID 0x01FF
//Chnage Product Id
#define USB_CONFIG_VID 0x04D8 //Chnage
Vendor Id
#ifndef __EX_USB_COMMON_H__
#define __EX_USB_COMMON_H__
#define MAX_PACKET_SIZE 61

#define COM_NO_STATE 0x11
#define COM_EXE_DONE 0x07

#define CMD_T0_ON 0x37
#define CMD_T0_OFF 0x2A
#define CMD_CLR_TIME 0x17
#define CMD_TOG_PUMP 0x1A
#define CMD_TOG_VALV 0x1B
#define CMD_NO_IR 0x3D
#define CMD_DO_IR 0x4D
```

```
#define CMD_SEND_HIGH 0x51
#define CMD_SEND_LOW 0x55
#define CMD_SEND_COMB 0x58
#define CMD_CON_PR 0x72
#define CMD_CON_PH 0x77
#define CMD_NO_CON 0x74
#define CMD_OPEN_VALVE 0x79
#define CMD_CLOSE_VALVE 0x7B
#define PROC_CALIBR 0x71

#define PUMP_PIN_B6
#define VALVE_PIN_B7
#define valHi PIN_B4 // Pressurizing Valve
#define valLo PIN_B5 //Depressurizing valve
#define LED_PIN_C1
#define IRLED_PIN_A4
#define loadDelay 0x5A
#define valPrsN 0x01 //0x05
#define pumpPWMN 30
#define DELAY 350
#define sampleSize 10
#define PhPin 0
#define PrPin 1
```

```
volatile unsigned char sendData[TX_SIZE],
recData[RX_SIZE], cmdBuff[sampleSize];
volatile unsigned int8 timeL = 0x00,timeH = 0x00;
volatile int8 currentPacket = 4, totalCMDCount = 0;
volatile int1 sendPacket = 0, dolRMod = TRUE, send-
dHigh = TRUE, valLoDrive = FALSE, valHiDrive =
FALSE, conPrRef = FALSE, cmdRec = FALSE, send-
Comb = TRUE;
volatile int1 onPrChannel = FALSE, hasReadings =
FALSE;
volatile int8 sampState = 0, procState = 0x00, doProc
= 0x00, valPrs = valPrsN, pumpPWMReg = pump-
PWMN, pumpPWMDuty = 0x00, pumpDutyLock =
0x00;
volatile int8 valHiPuls = 0x00, valHiLock = 0x00, val-
LoPuls = 0x00, valLoLock = 0x00;
volatile unsigned int16 reading[3], prRef = 0x3FF, phRef
= 0x3FF, procSignal[2];
volatile int8 phLSB, prLSB, combMSB;
volatile int8 procSLSB,refLSB, combMSB2,sampMSB2;
```

### USB MAIN.C

```
#include <USB_main.h>
```

```
#include <math.h>
#include <usb_desc_hid.h>
#include <pic18_usb.h>
#include <usb.c>
```

```
void calibrProc(){
doProc = 0x00;
}
```

```
void packInfo(){
procSLSB = make8(procSignal[0],0);
refLSB = make8(prRef, 0);
combMSB2 = make8(procSignal[0],1);
sampMSB2 = make8(prRef, 1);
```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```
    combMSB2 = sampMSB2 | (combMSB2 << 2);
    sendData[3] = reflSB;
    sendData[2] = combMSB2;
}

void packReadings(){
    int16 samplePH;
    if (!sendHigh) samplePH = reading[2];
    if (sendHigh) samplePH = reading[0];
    if (sendComb) samplePH = reading[2] + reading[0];
    phLSB = make8(samplePH,0);
    prLSB = make8(reading[1],0);
    combMSB = make8(samplePH,1);
    int8 sampMSB = make8(reading[1],1);
    combMSB = (combMSB << 2) | sampMSB;
}

void packSamples(int1 offset){
    /*if(timeL== 0xFF){
        timeH++;
    }*/
    timeL++;
    if(currentPacket == 0){
        currentPacket = 4;
    }
    delay_us(1);
    packReadings();
    packInfo();
    if(currentPacket < MAX_PACKET_SIZE /*&&
lastPhSignal != phSignal[0]*){
        sendData[currentPacket] = timeL;
        sendData[currentPacket+1] = combMSB;
        sendData[currentPacket+2] = phLSB;
        sendData[currentPacket+3] = prLSB;
        currentPacket += 4;
    }
    if(currentPacket >= MAX_PACKET_SIZE){
        sendPacket = 1;
    }
}

void pwmPump(){
    pumpPWMReg--;
    if(pumpPWMDuty == 0x00){
        OUTPUT_HIGH(PUMP);
    }else {
        pumpPWMDuty--;
        OUTPUT_LOW(PUMP);
    }
    if(pumpPWMReg == 0x00){
        pumpPWMReg = pumpPWMN;
        pumpPWMDuty = pumpPWMN - pumpDutyLock;
    }
}

void dynamicControl(){
    valLoPuls = valLoLock;
    valHiPuls = valHiLock;
    if(pumpDutyLock > 30) pumpDutyLock = 30;
}

void controlPressure(){
    if(valLoDrive){
        OUTPUT_HIGH(valLo);
    } else {
        OUTPUT_LOW(valLo);
    }
    if(valHiDrive){
        OUTPUT_HIGH(valHi);
    }else{
        OUTPUT_LOW(valHi);
    }
    if(valPrs-- == 0x00){
        if(valLoPuls != 0x00){
            valLoPuls--;
            valLoDrive = !valLoDrive;
        }else{
            valLoDrive = FALSE;
        }
    }
    if(valHiPuls != 0x00){
        valHiPuls--;
        valHiDrive = !valHiDrive;
    }else{
        valHiDrive = FALSE;
    }
    valPrs = valPrsN;
}

}

#INT_RTCC
void t0_int(void){
    TMR0 += 0x8D;
    sampState++;
    switch (sampState){
        case 3:
            OUTPUT_HIGH(IRLED);
            break;
        case 7:
            if(doIRMod) OUTPUT_LOW(IRLED);
            break;
        case 8:
            reading[2] = 0x3FF - reading[2];
            packSamples(FALSE);
            reading[0] = reading[1] = reading[2] = 0;
            hasReadings = FALSE;
            clear_interrupt(INT_AD);
            set_adc_channel(PrPin);
            delay_us(5);
            read_adc(ADC_START_ONLY);
            onPrChannel = TRUE;
            sampState = 0x00;
    }

    controlPressure();
    if(conPrRef) pwmPump();
    if (reading[1] > 950){
        OUTPUT_LOW(PUMP);
    }
}

#INT_AD
void AD_ISR(){
    int16 readResult = read_adc(ADC_READ_ONLY);
    if(onPrChannel){
        reading[1] = readResult;
        set_adc_channel(PhPin);
        onPrChannel = FALSE;
    }
}
```

```

    delay_us(5);
    read_adc(ADC_START_ONLY);
} else {
    if(!hasReadings){
        reading[0] = reading[2] = readResult;
        hasReadings = TRUE;
    }else{
        if(readResult > reading[2]) reading[2] = readResult;
        if(readResult < reading[0]) reading[0] = readResult;
    }
    read_adc(ADC_START_ONLY);
}

void unpackSignals(int1 hasRef){
    procSignal[0] = make16((cmdBuff[1] & 0b00011100
    >> 2), cmdBuff[2]);
    procSignal[1] = make16(((cmdBuff[1] & 0x03)),
    cmdBuff[3]);
    if(hasRef){
        prRef = make16(((cmdBuff[1] & 0b11100000) >> 5),
    cmdBuff[4]);
    }
    pumpDutyLock = cmdBuff[5];
    valHiLock = cmdBuff[6];
    valLoLock = cmdBuff[7];
    if(valLoLock == 0x00) OUTPUT_LOW(valLo);
    if(valHiLock == 0x00) OUTPUT_LOW(valHi);

    //if(pumpDutyLock == 0x00) OUTPUT_LOW(PUMP);
}

#INT_TIMER1
void t1_int(void){
    TMR1H += 22;
    if(cmdRec){
        int1 hasRef = FALSE;
        switch(cmdBuff[0]){
            case CMD_CLR_TIME:
                timeH = 0;
                timeL = 0;
                OUTPUT_LOW(valLo);
                OUTPUT_LOW(valHi);
                conPrRef = FALSE;
                valLoLock = valHiLock = 0;
                pumpDutyLock = FALSE;
                totalCMDCount = CMD_CLR_TIME;
                break;
            case CMD_T0_ON:
                enable_interrupts(INT_TIMER0);
                enable_interrupts(INT_AD);
                totalCMDCount = CMD_T0_ON;
                break;
            case CMD_T0_OFF:
                disable_interrupts(INT_TIMER0);
                disable_interrupts(INT_AD);
                totalCMDCount = CMD_T0_OFF;
                OUTPUT_LOW(valLo);
                OUTPUT_LOW(valHi);
                OUTPUT_LOW(PUMP);
                conPrRef = FALSE;
                valLoLock = valHiLock = 0;

                pumpDutyLock = FALSE;
                break;
            case CMD_TOG_PUMP:
                output_toggle(PUMP);
                totalCMDCount = CMD_TOG_PUMP;
                break;
            case CMD_OPEN_VALVE:
                OUTPUT_LOW(VALUE);
                totalCMDCount = CMD_OPEN_VALVE;
                break;
            case CMD_CLOSE_VALVE:
                OUTPUT_HIGH(VALUE);
                totalCMDCount = CMD_CLOSE_VALVE;
                break;
            case CMD_TOG_VALV:
                output_toggle(VALUE);
                //output_toggle(valLo);
                //output_toggle(valHi);
                totalCMDCount = CMD_TOG_VALV;
                break;
            case CMD_DO_IR:
                doIRMod = TRUE;
                totalCMDCount = CMD_DO_IR;
                break;
            case CMD_NO_IR:
                doIRMod = FALSE;
                totalCMDCount = CMD_NO_IR;
                break;
            case CMD_SEND_HIGH:
                sendHigh = TRUE;
                sendComb = FALSE;
                totalCMDCount = CMD_SEND_HIGH;
                break;
            case CMD_SEND_LOW:
                sendHigh = FALSE;
                sendComb = FALSE;
                totalCMDCount = CMD_SEND_LOW;
                break;
            case CMD_SEND_COMB:
                sendComb = TRUE;
                totalCMDCount = CMD_SEND_COMB;
                break;
            case CMD_CON_PR:
                conPrRef = TRUE;
                hasRef = TRUE;
                totalCMDCount = CMD_CON_PR;
                break;
            case CMD_NO_CON:
                conPrRef = FALSE;
                totalCMDCount = CMD_NO_CON;
                valLoLock = valHiLock = pumpDutyLock = 0x00;
                OUTPUT_LOW(PUMP);
                break;
            case PROC_CALIBR:
                doProc = PROC_CALIBR;
                procState = 0x00;
                totalCMDCount = PROC_CALIBR;
                break;
            case 0x00:
                totalCMDCount = 0x00;
                break;
        }
        cmdRec = FALSE;
        if(cmdBuff[0] != 0x00) sendData[0] =
        COM_EXE_DONE;
    }
}

```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```
    if(cmdBuff[0] == CMD_CLR_TIME || cmdBuff[0] ==
CMD_T0_OFF){
        sendPacket = 1;
        memset(&sendData[currentPacket], 0x00, 64 -
currentPacket);

    }
    cmdBuff[0] = 0;
    unpackSignals(hasRef);
    if(conPrRef) dynamicControl();
}

void main()
{

setup_adc(ADC_CLOCK_DIV_64|ADC_TAD_MUL_8);
setup_adc_ports(AN0_TO_AN4|VSS_VREF);
set_adc_channel(PrPin);
setup_timer_0(RTCC_INTER-
NAL|RTCC_DIV_8|RTCC_8_BIT); //Overflow evert
21.333*RTCC_PRESCALER usec //51,2 us overflow
setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
enable_interrupts(GLOBAL);
enable_interrupts(PERIPH);
disable_interrupts(INT_TIMER1);
disable_interrupts(INT_TIMER0);
disable_interrupts(INT_AD);
disable_interrupts(INT_CCP2);
setup_ccp1(CCP_PWM);
doProc = 0;

SPEN = 0;
//OUTPUT_HIGH(LED);
OUTPUT_LOW(IRLED);
//OUTPUT_LOW(PIN_C2);
OUTPUT_LOW(PUMP);
OUTPUT_LOW(VALVE);
OUTPUT_LOW(valHi);
OUTPUT_LOW(valLo);

//setup_ccp2(CCP_PWM);
usb_init_cs();

//Example blinking LED program
while(true)
{
    usb_task();

    if(usb_enumerated()){
        if(usb_kbhit(1)){
            usb_get_packet(1, recData, USB_CON-
FIG_HID_RX_SIZE);
            int i;
            cmdRec = TRUE; //REMEMBER TO
DELETE THIS ON TIME FOR COMMANDS LINE
WORKING
            for(i=0; i < RX_SIZE; i++){
                cmdBuff[i] = recData[i];
                recData[i] = 0;
            }
            enable_interrupts(INT_TIMER1);
            delay_us(3);
            //usb_put_packet(1, recData, USB_CON-
FIG_HID_TX_SIZE, USB_DTS_TOGGLE);
        }
        if(sendPacket){
            disable_interrupts(GLOBAL);
            if(sendData[0] == 0x00){
                sendData[0] = COM_NO_STATE;
            }else if(sendData[0] == COM_EXE_DONE){
                sendData[1] = totalCMDCount;
                totalCMDCount = 0;
            }
            usb_put_packet(1, sendData, USB_CON-
FIG_HID_TX_SIZE, USB_DTS_TOGGLE);
            currentPacket = 4;
            memset(sendData, 0x00, sizeof(sendData));
            sendData[0] = sendData[1] = 0x00;
            sendPacket = 0;
            enable_interrupts(GLOBAL);
        }
    }
}
```

## Anexo N° 3: Código del Software de PC.

### QtUSBGraph.h

```

#pragma once

#include <QtWidgets/QMainWindow>
#include "ui_QtUSBGraph.h"
#include "hidapi.h"
#include "pid.h"
#include <fftw3.h>

#define VID_NUM 0x04D8
#define PID_NUM 0x01FF
#define TX_SIZE 8
//// ----- Timer0 Adjustment variables for time calibration ----- ////
#define T0_PRESCALER 8*8 //Timer0 Prescaler
* Number of States before Counting Time Step
#define T0_MIN_OFF 0.08441999 / 1000 //Calibrated Timer0 Period = (1/(Clock (48 MHz) / 4))
#define T0_PL_Adj (0xFF - 0x8D) //Timer0 Preload Adjustment

#define prScaleMMHG 0.5
#define xAxisRange 8
#define maxAvgSize 200
#define nOfPeaks 42
#define DFT_SIZE 2048

// Constant communication instructions for PIC HID service
#define COM_NO_STATE 0x11
#define COM_EXE_DONE 0x07

#define CMD_T0_ON 0x37
#define CMD_T0_OFF 0x2A
#define CMD_CLR_TIME 0x17
#define CMD_TOG_PUMP 0x1A
#define CMD_TOG_VALV 0x1B
#define CMD_NO_IR 0x3D
#define CMD_DO_IR 0x4D
#define CMD_SEND_HIGH 0x51
#define CMD_SEND_LOW 0x55
#define CMD_SEND_COMB 0x58
#define CMD_CON_PR 0x72
#define CMD_CON_PH 0x77
#define CMD_NO_CON 0x74

#define PROC_CALIBR 0x81
#define PROC_CAL_RISE 0x82
#define PROC_CAL_FALL 0x84
#define PROC_CAL_MEDPR 0x85
#define PROC_PRE_CAL_MEDPH 0x80
#define PROC_CAL_MEDPH 0x86
#define PROC_CON_PH 0x87

struct pulseDataSet {
    float timeKey = 0;
    int pxAverage;
    int pxPoint;
    int pxHeight;
    float HRFreq;

    bool operator==(const pulseDataSet& Y) {
        bool result = TRUE;
        result = (timeKey == Y.timeKey) &&
                (pxAverage == Y.pxAverage) &&
                (pxPoint == Y.pxPoint) &&
                (pxHeight == Y.pxHeight) &&
                (HRFreq == Y.HRFreq);
    }

    bool operator!=(const pulseDataSet& Y) {
        bool result = FALSE;
        if (timeKey != Y.timeKey) result = TRUE;
        if (pxAverage != Y.pxAverage) result = TRUE;
        if (pxPoint != Y.pxPoint) result = TRUE;
        if (pxHeight != Y.pxHeight) result = TRUE;
        if (HRFreq != Y.HRFreq) result = TRUE;
        return result;
    }
};

class QtUSBGraph : public QMainWindow
{
    Q_OBJECT

public:
    QtUSBGraph(QWidget *parent = Q_NULLPTR);

private:
    Ui::QtUSBGraphClass ui;
    QTimer *dataTimer;
    QTimer *cmdTimer;
    pid_ctrl_t pumpPIDCon;
    pid_ctrl_t valHiPIDCon;
    pid_ctrl_t valLoPIDCon;
    pid_ctrl_t phPIDCon;
    QStandardItemModel *tvModel = new QStandardItemModel(9, 4, this); //2 Rows and 3 Columns
    QStandardItemModel *tvModel2 = new QStandardItemModel(9, 4, this); //2 Rows and 3 Columns
    fftw_plan fftPlan;
    double fftInArray[DFT_SIZE];
    double *fftIn = fftInArray;
    fftw_complex fftOutArray[DFT_SIZE];
    fftw_complex *fftOut = fftOutArray;
    double timeCorrection = 0.98661402;
    int lastSampleSize = 0;
    double lastFpsKey = 0.00;
    int frameCount = 0;
    int lastPrSignal = 0;
    int lastPhSignal = 0;
    double lastTimeKey = 0;
    int lastTimeStamp = 0;
    int phSignal[maxAvgSize];
    int prSignal[maxAvgSize];
    int phSignalTrail[30];
    int prSignalTrail[30];
    float phSigDeriv = 1.0;

```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```
float prSigDeriv = 1.0;
pulseDataSet phPulseData[nOfPeaks];
pulseDataSet prPulseData[nOfPeaks];
int prAverageSumm = 0, phAverageSumm
= 0;
int prRef = 0, phRef = 0, nValPuls =
0, pumpDutyLock = 0, valHiLock = 0, val-
LoLock = 0;
bool conPrRef = FALSE, conPhRef =
FALSE, riseCalOK = FALSE;
int sampleCountPh = 0, sampleCountPr
= 0;
float sampleFreq = 0, heartFreq = 0;
unsigned char cmdState = 0x00;
bool onPeak = FALSE;
int maxPhPeak = 0;
float lastPhPeak = 0, initPhPeak =
0;
int HRPlotIndex = 0;
int prSigOffset = 500;
int medPresSetpoint = 0, medPhSet-
point = 0;
hid_device *handle;
unsigned char *cmdBuff[10];
unsigned char *comBuff[4];
bool sendCMD = FALSE, checkCMD =
FALSE, incrStamp = FALSE;
unsigned char *verifCMD = 0x00;
unsigned int timeStampMultiplier =
0;
int deprPoints = 0;
int nOfSamples = 0;
int dataSent[3];
double predPeriod;
//double timeStamp = 0.0;
//clock_t initTime = clock();

//
private slots:
void realtimeDataSlot();
void handleCMD();
void startTimer();
void stopTimer();
void togglePump();
void toggleValve();
void changeModulate();
void changeSentData();
void connectDevice();
void disconnectDevice();
void addNewSigSample(int phSample,
int prSample);
void showGraphs();
void initCalibr();
void packSignals();
void conPress();
void releasePr();
void controlProcess();
void valueOnMouseOver(QMouseEvent
*evt);
void phPeakDetect();
void prPIDControl();
void prPIDControl2();
void plotPhHeartFreq();
void plotPrHeartFreq();
void conPhSignal();
void reTunePIDs(bool onDynamic);
void keyPressEvent(QKeyEvent
*event);
void keyReleaseEvent(QKeyEvent
*event);
void saveSelectedPlot();
void openPlotFile();
```

```
void plotFFTSig();
void swapVisiblePlots();
int predictiveLoop();
void prPeakDetect();
int findPhSetpoint(int medPr);
void setPlotBg();
void reCalPhSetpoint(bool reset);
};
```

### QtUSBGraph.cpp

```
#include "QtUSBGraph.h"

#include <QDebug>
#include <QDesktopWidget>
#include <QScreen>
#include <QMessageBox>
#include <QMetaEnum>
#include <stdio.h>
#include <ctype.h>
#include <stdarg.h>
#include <wchar.h>
#include <string.h>
#include <stdlib.h>
#include <iostream>
#include <iomanip>
#include <sstream>

void __cdecl odprintf(const char *format,
...)
{
    char buf[4096], *p = buf;
    va_list args;
    int n;

    va_start(args, format);
    n = _vsnprintf(p, sizeof buf - 3,
format, args); // buf-3 is room for
CR/LF/NUL
    va_end(args);

    p += (n < 0) ? sizeof buf - 3 : n;

    while (p > buf && isspace(p[-1]))
        *--p = '\0';

    *p++ = '\n';
    *p++ = '\n';
    *p = '\0';
    wchar_t* wString = new
wchar_t[4096];
    MultiByteToWideChar(CP_ACP, 0, buf,
-1, wString, 4096);
    OutputDebugString(wString);
}

int filterLowPass(int values[maxAvgSize],
int sizeAvg) {
    int result = 0;
    for (int i = 0; i < sizeAvg; i++) {
        result += values[i];
    }
    result = result / sizeAvg;
    return result;
}

int highOrderLPFilter(int newSample) {
#define NZEROS 4
#define NPOLES 4
#define GAIN 7.169593472e+03

    static float xv[NZEROS + 1],
yv[NPOLES + 1];
```

```

        xv[0] = xv[1]; xv[1] = xv[2]; xv[2]
= xv[3]; xv[3] = xv[4];
        xv[4] = newSample / GAIN;
        yv[0] = yv[1]; yv[1] = yv[2]; yv[2]
= yv[3]; yv[3] = yv[4];
        yv[4] = (xv[0] + xv[4]) + 4 * (xv[1]
+ xv[3]) + 6 * xv[2]
            + (-0.4758998940 * yv[0]) +
(2.2686871057 * yv[1])
            + (-4.0819008839 * yv[2]) +
(3.2868820255 * yv[3]);
        return yv[4];
}

```

```

QtUSBGraph::QtUSBGraph(QWidget *parent)
: QMainWindow(parent)
{
    ui.setupUi(this);

#pragma region USB_Connection

    // Open the device using the VID,
    PID,
    // and optionally the Serial number.
    ////handle = hid_open(0x4d8, 0x3f,
    L"12345");

#pragma endregion

    //test_interrupt_transfer(deviceHan-
    dle);

    dataTimer = new QTimer(this);
    cmdTimer = new QTimer(this);
    ui.QCPlot1->axisRect()->insetLayout()->setInsetAlignment(0, Qt::AlignBottom
| Qt::AlignRight);
    ui.QCPlot1->addGraph(ui.QCPlot1-
>xAxis, ui.QCPlot1->yAxis2); // blue line
    ui.QCPlot1->graph(0)-
>setPen(QPen(QColor(Qt::green), 2));
    ui.QCPlot1->addGraph(); // red line
    ui.QCPlot1->graph(1)-
>setPen(QPen(QColor(255, 110, 40), 2));
    ui.QCPlot2->addGraph(); // blue line
    ui.QCPlot2->graph(0)-
>setPen(QPen(QColor(Qt::green), 3));

#pragma region graphSetting
    ui.QCPlot1->setBack-
ground(Qt::black);
    ui.QCPlot2->setBack-
ground(Qt::black);

    QSharedPointer<QCPAxisTickerTime>
timeTicker(new QCPAxisTickerTime);
    timeTicker->setTimeFormat("%s");
    ui.QCPlot1->xAxis->set-
Ticker(timeTicker);
    ui.QCPlot1->xAxis->setTick-
LabelColor(Qt::white);
    ui.QCPlot1->yAxis->setTick-
LabelColor(QColor(255, 110, 40));
    ui.QCPlot1->yAxis2->setTick-
LabelColor(QColor(Qt::green));

```

```

        ui.QCPlot2->xAxis->set-
Ticker(timeTicker);
        ui.QCPlot2->yAxis->setTick-
LabelColor(Qt::white);
        ui.QCPlot2->yAxis->setTick-
LabelColor(Qt::white);

        ui.QCPlot1->xAxis->set-
BasePen(QPen(Qt::white));
        ui.QCPlot1->yAxis2->set-
BasePen(QPen(QColor(Qt::green)));
        ui.QCPlot1->yAxis->set-
BasePen(QPen(QColor(255, 110, 40)));

        ui.QCPlot2->xAxis->set-
BasePen(QPen(Qt::white));
        ui.QCPlot2->yAxis->set-
BasePen(QPen(Qt::white));

        ui.QCPlot1->yAxis2->setVisi-
ble(true);
        ui.QCPlot1->xAxis-
>setLabel(QString("Time (seconds)"));
        ui.QCPlot1->yAxis-
>setLabel(QString("Pressure (mmHg)"));
        ui.QCPlot1->yAxis2-
>setLabel(QString("Ph Signal"));
        ui.QCPlot1->graph(0)->set-
Name(QString("Ph Signal"));
        ui.QCPlot2->xAxis-
>setLabel(QString("Heart Beats"));
        ui.QCPlot2->yAxis-
>setLabel(QString("Frequency"));

        //ui.QCPlot1->yAxis2-
>setLabel(QString("Signal"));
        ui.QCPlot1->xAxis-
>setLabelColor(QColor(255, 255, 255));
        ui.QCPlot1->yAxis-
>setLabelColor(QColor(255, 110, 40));
        ui.QCPlot1->yAxis2-
>setLabelColor(QColor(Qt::green));
        ui.QCPlot2->xAxis-
>setLabelColor(QColor(255, 255, 255));
        ui.QCPlot2->yAxis-
>setLabelColor(QColor(255, 255, 255));

        //ui.QCPlot1->axisRect()->set-
upFullAxesBox();
        ui.QCPlot1->yAxis->setRange(-1,
250);
        ui.QCPlot1->yAxis2->setRange(-1,
1024);

        ui.QCPlot2->axisRect()->setUpFullAx-
esBox();
        ui.QCPlot2->yAxis->setRange(-1,
200);
        ui.QCPlot2->xAxis->setRange(-10,
10);

        ui.QCPlot1->setInterac-
tion(QCP::iRangeDrag, true);

        ui.QCPlot2->setInterac-
tion(QCP::iRangeDrag, true);
        ui.QCPlot2->setInterac-
tion(QCP::iRangeZoom, true);

        ui.QCPlot1->setInterac-
tion(QCP::iRangeZoom, true);

```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```

        ui.QCPlot1->axisRect()-
>setRangeDragAxes(ui.QCPlot1->xAxis,
ui.QCPlot1->yAxis2);
        ui.QCPlot1->axisRect()->setRangeZoo-
mAxes(ui.QCPlot1->xAxis, ui.QCPlot1-
>yAxis2);
        ui.QCPlot1->graph(0)->set-
Selectable(QCP::stMultipleDataRanges);
        ui.QCPlot1->graph(1)->set-
Selectable(QCP::stMultipleDataRanges);
        ui.QCPlot1->setInteraction(QCP::iSe-
lectPlottables, true);
        ui.QCPlot1->setInterac-
tion(QCP::iMultiSelect);
        ui.QCPlot1->setMultiSelectModi-
fier(Qt::ControlModifier);
        //ui.QCPlot1->axisRect()->setRange-
Zoom(Qt::Vertical);
        //ui.QCPlot2->axisRect()-
>setRangeDrag(Qt::Horizontal);
        ui.QCPlot2->yAxis->setScale-
Type(QCPAxis::stLogarithmic);
        //ui.QCPlot2->xAxis->setScale-
Type(QCPAxis::stLogarithmic);

        // -> ui.QCPlot1->graph(0)->se-
lected();
        ui.sigText1->setTextColor(QColor(0,
255, 0, 200));
        ui.sigText2-
>setTextColor(QColor(255, 0, 0, 200));
        ui.sigText2->setTextBackground-
Color(QColor(0, 0, 0, 0));
        ui.sigText1->setTextBackground-
Color(QColor(0, 0, 0, 0));
        //ui.hrFreqText->setTextBackground-
Color(QColor(0, 0, 0, 0));
        QFont font = ui.sigText1->font();
        //font.setBold(true);
        font.setPixelSize(30);
        QFont font1 = ui.tooltipTxt->font();
        font1.setPixelSize(12);
        ui.tooltipTxt->setFont(font1);
        ui.sigText1->setFont(font);
        ui.sigText2->setFont(font);
        font.setPixelSize(20);
        //ui.hrFreqText->setFont(font);
        ui.sigText1->setEnabled(false);
        ui.sigText2->setEnabled(false);
        //ui.hrFreqText->setEnabled(false);
        ui.tooltipTxt->setEnabled(false);
        ui.tooltipTxt->setMouseTrack-
ing(true);
        //ui.tooltipTxt->setHidden(true);
        ui.sigText1->viewport()->setAu-
toFillBackground(false);
        ui.sigText2->viewport()->setAu-
toFillBackground(false);
        //ui.hrFreqText->viewport()->setAu-
toFillBackground(false);
        ui.tooltipTxt->viewport()->setAu-
toFillBackground(false);
        ui.sigText1->setText("");
        ui.sigText2->setText("");
        //ui.hrFreqText->setText("0");
        ui.sigText1->setAlignment(Qt::Align-
Right);
        ui.sigText2->setAlignment(Qt::Align-
Right);
        //ui.hrFreqText->setAlign-
ment(Qt::AlignRight);
        ui.QCPlot2->setVisible(FALSE);

        tvModel->setHorizontalHeaderItem(0,
new QStandardItem(QString("Time Key")));
        tvModel->setHorizontalHeaderItem(1,
new QStandardItem(QString("Point")));
        tvModel->setHorizontalHeaderItem(2,
new QStandardItem(QString("HR")));
        tvModel->setHorizontalHeaderItem(3,
new QStandardItem(QString("Height")));
        tvModel->setHorizontalHeaderItem(4,
new QStandardItem(QString("Average")));

        tvModel2->setHorizontalHeaderItem(0,
new QStandardItem(QString("Time Key")));
        tvModel2->setHorizontalHeaderItem(1,
new QStandardItem(QString("Point")));
        tvModel2->setHorizontalHeaderItem(2,
new QStandardItem(QString("HR")));
        tvModel2->setHorizontalHeaderItem(3,
new QStandardItem(QString("Height")));
        tvModel2->setHorizontalHeaderItem(4,
new QStandardItem(QString("Average")));

        ui.hbTableView->setModel(tvModel);
        ui.hbTableView2->setModel(tvModel2);
        ui.hbTableView->setColumnWidth(0,
50);
        ui.hbTableView->setColumnWidth(1,
45);
        ui.hbTableView->setColumnWidth(2,
35);
        ui.hbTableView->setColumnWidth(3,
45);
        ui.hbTableView->setColumnWidth(4,
45);
        ui.hbTableView2->setColumnWidth(0,
50);
        ui.hbTableView2->setColumnWidth(1,
45);
        ui.hbTableView2->setColumnWidth(2,
35);
        ui.hbTableView2->setColumnWidth(3,
45);
        ui.hbTableView2->setColumnWidth(4,
45);

        ui.hbTableView->show();
        ui.hbTableView2->show();
        ui.hbTableView2->setVisible(FALSE);
#pragma endregion

        // make left and bottom axes transfer
their ranges to right and top axes:
        connect(ui.QCPlot1->xAxis, SIG-
NAL(rangeChanged(QCPRange)), ui.QCPlot1-
>xAxis2, SLOT(setRange(QCPRange)));
        //connect(ui.QCPlot1->yAxis, SIG-
NAL(rangeChanged(QCPRange)), ui.QCPlot1-
>yAxis2, SLOT(setRange(QCPRange)));
        // setup a timer that repeatedly
calls MainWindow::realtimeDataSlot:
        qDebug() << connect(ui.stopBtn, SIG-
NAL(clicked()), this, SLOT(stopTimer()));
        qDebug() << connect(ui.startBtn,
SIGNAL(clicked()), this, SLOT(start-
Timer()));
        connect(ui.connectBtn, SIG-
NAL(clicked()), this, SLOT(connectDe-
vice()));
        connect(ui.dcBtn, SIGNAL(clicked()),
this, SLOT(disconnectDevice()));
        connect(ui.pumpToggleBtn, SIG-
NAL(clicked()), this, SLOT(togglePump()));

```

```

        connect(ui.valveToggleBtn, SIG-
NAL(clicked()), this, SLOT(toggleValve()));
        connect(dataTimer, SIG-
NAL(timeout()), this,
SLOT(realtimeDataSlot()));
        connect(ui.modChk, SIG-
NAL(clicked()), this, SLOT(changeModu-
late()));
        connect(ui.stepRBtn1, SIG-
NAL(clicked()), this, SLOT(changeSent-
Data()));
        connect(ui.stepRBtn2, SIG-
NAL(clicked()), this, SLOT(changeSent-
Data()));
        connect(ui.stepRBtn3, SIG-
NAL(clicked()), this, SLOT(changeSent-
Data()));
        connect(ui.sig1VisChk, SIG-
NAL(clicked()), this, SLOT(showGraphs()));
        connect(ui.sig2VisChk, SIG-
NAL(clicked()), this, SLOT(showGraphs()));
        connect(cmdTimer, SIGNAL(timeout()),
this, SLOT(controlProcess()));
        connect(ui.calibrBtn, SIG-
NAL(clicked()), this, SLOT(initCalibr()));
        connect(ui.noPrConBtn, SIG-
NAL(clicked()), this, SLOT(releasePr()));
        connect(ui.conPrBtn, SIG-
NAL(clicked()), this, SLOT(conPress()));
        connect(ui.conPhBtn, SIG-
NAL(clicked()), this, SLOT(conPhSignal()));
        connect(ui.QCPlot1, SIGNAL(mouse-
Move(QMouseEvent*)), this, SLOT(valueOn-
MouseOver(QMouseEvent*)));
        connect(ui.saveBtn, SIG-
NAL(clicked()), this, SLOT(saveSelect-
edPlot()));
        connect(ui.plotSelectBox, SIG-
NAL(currentIndexChanged(int)), this,
SLOT(swapVisiblePlots()));
        connect(ui.openFileBtn, SIG-
NAL(clicked()), this, SLOT(openPlotFile()));
        connect(ui.whiteBgChk, SIG-
NAL(clicked()), this, SLOT(setPlotBg()));
        //connect(ui.tooltipTxt, SIG-
NAL(mouseMove(mouseMoveEvent*)), this,
SLOT(valueOnMouseOver(QMouseEvent*)));
        //dataTimer->start(0); // Interval 0
means to refresh as fast as possible
    }

void QtUSBGraph::keyPressEvent(QKeyEvent
*event) {
    if (event->key() == Qt::Key_Control)
    {
        ui.QCPlot1->setSelection-
RectMode(QCP::srmSelect);
    }
}

void QtUSBGraph::keyReleaseEvent(QKeyEvent
*event) {
    if (event->key() == Qt::Key_Control)
    {
        ui.QCPlot1->setSelection-
RectMode(QCP::srmNone);
    }
}

void QtUSBGraph::valueOnMouseO-
ver(QMouseEvent *evt) {
    double mouseX = ui.QCPlot1-
>xAxis->pixelToCoord(evt->pos().x());
    int mouseY = ui.QCPlot1-
>yAxis->pixelToCoord(evt->pos().y());
    int mouseY2 = ui.QCPlot1-
>yAxis2->pixelToCoord(evt->pos().y());
    int indexPH = ui.QCPlot1-
>graph(0)->findBegin(mouseX);
    int indexPR = ui.QCPlot1-
>graph(1)->findBegin(mouseX);
    double phX = ui.QCPlot1-
>graph(0)->dataMainKey(indexPH);
    double prX = ui.QCPlot1-
>graph(1)->dataMainKey(indexPR);
    int phY = ui.QCPlot1-
>graph(0)->dataMainValue(indexPH);
    int prY = ui.QCPlot1-
>graph(1)->dataMainValue(indexPR);

    if (std::abs(mouseX - phX) <
0.3 || std::abs(mouseX - prX) < 0.3) {
        if (std::abs(mouseY
- prY) < 15) {
            ui.tooltipTxt->set-
PlainText(QString(""));

            ui.tooltipTxt-
>setTextColor(QColor(Qt::white));

            ui.tooltipTxt->in-
sertPlainText(QString("%1 ").arg(phX, 0,
'f', 2));

            ui.tooltipTxt-
>setTextColor(QColor(Qt::green));

            ui.tooltipTxt->in-
sertPlainText(QString("%1 ").arg(phY));

            ui.tooltipTxt-
>setTextColor(QColor(Qt::red));

            ui.tooltipTxt->in-
sertPlainText(QString("%1").arg(prY));

            ui.tooltipTxt->setVisible(TRUE);
            int posX =
ui.QCPlot1->xAxis->coordToPixel(prX);
            int posY =
ui.QCPlot1->yAxis->coordToPixel(prY) - 20;

            ui.tooltipTxt->move(posX, posY);
        }
    }
    else if
(std::abs(mouseY2 - phY) < 15) {
        ui.tooltipTxt->set-
PlainText(QString(""));

        ui.tooltipTxt-
>setTextColor(QColor(Qt::white));

        ui.tooltipTxt->in-
sertPlainText(QString("%1 ").arg(phX, 0,
'f', 2));

        ui.tooltipTxt-
>setTextColor(QColor(Qt::green));

        ui.tooltipTxt->in-
sertPlainText(QString("%1 ").arg(phY));

        ui.tooltipTxt-
>setTextColor(QColor(Qt::red));
}
}

```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```

        ui.tooltipTxt->insertPlainText(QString("%1").arg(prY));

        ui.tooltipTxt->setVisible(TRUE);
        int posX = ui.QCPlot1->xAxis->coordToPixel(phX);
        int posY = ui.QCPlot1->yAxis2->coordToPixel(phY) - 20;

        ui.tooltipTxt->move(posX, posY);
    }
    else {
        ui.tooltipTxt->setVisible(FALSE);
    }
    else {
        ui.tooltipTxt->setVisible(FALSE);
    }
    //odprintf("%d", xx);
}

void QtUSBGraph::handleCMD() {
    //odprintf("Entering handleCMD");
    if (checkCMD && comBuff[0] == (unsigned char*)COM_EXE_DONE){
        //odprintf("verifCMD: %02hhx - Received COM: %02hhx ", verifCMD, comBuff[1]);
        if (verifCMD == comBuff[1])
        {
            //odprintf("Command was executed properly!\n");
            checkCMD = FALSE;
            verifCMD = 0x00;
        }
        else {
            //odprintf("Command failed to execute! \n");
            checkCMD = FALSE;
            //verifCMD = 0x00;
        }
    }
    if (sendCMD) {
        if (conPhRef) prPIDControl2();
        if (conPrRef && !conPhRef) prPIDControl();
        //dynamicControl();
        packSignals();
        //odprintf("Ready to Send Command");
        unsigned char buff[6];
        int t;

        for (t = 0; t < TX_SIZE; t++) {
            buff[t + 1] = (unsigned char)cmdBuff[t + 1];
        }
        int res;
        res = hid_write(handle, buff, 17);

        if (res >= 0) {
            //if (buff[1] != 0x00) odprintf("Writing succesfull! Command sent: %02hhx - %02hhx \n",buff[1],buff[2]);
            sendCMD = FALSE;
            if (buff[1] != 0x00)
                checkCMD = TRUE;
            int i;

            verifCMD = 0;
            for (i = 0; i < TX_SIZE + 1; i++) {
                cmdBuff[i] = 0;
            }
            //odprintf("Leaving handleCMD");
        }
    }

void QtUSBGraph::realtimeDataSlot() {
    handleCMD();
    //ui.statusBar->showMessage(QString("Func Running OK"));
    //odprintf("Entering realtimeDataSlot");
    unsigned char recData[256];
    static double fixedTime = 0.0;
    static double fixedTime2 = 0.0;
    static double refixedTime = 0.0;

    memset(recData, 0x00, sizeof(recData));
    int res;
    res = hid_read(handle, recData,81);
    if (res < 0) {
        //odprintf("Reading Failed\n");
        return;
    }
    if (recData[0] == 0x00) {
        //odprintf("No packet found %02hhx \n", recData[0]);
        //odprintf("%02hhx\n", recData[1]);
        return;
    }
    if (recData[0] == (unsigned char)COM_EXE_DONE) {
        memset(comBuff, 0x00, sizeof(comBuff));
        comBuff[0] = (unsigned char*)recData[0];
        comBuff[1] = (unsigned char*)recData[1];
        comBuff[2] = (unsigned char*)recData[2];
        comBuff[3] = (unsigned char*)recData[3];
        //odprintf("%02hhx \n", comBuff[0]);
        //odprintf("%02hhx \n", comBuff[1]);
        int signalFB = ((recData[2] & 0b00000011) << 8) + recData[3];
        //if (signalFB != dataSent[2]) odprintf("Mismatch Sent: %d - Received: %d", dataSent[0], signalFB);
        //odprintf("%d - %d %d \n", pressReference, recData[2], recData[3]);
        //checkCMD = TRUE;
    }
    else {
        comBuff[0] = 0x00;
        comBuff[1] = 0x00;
    }

    double timeKey;
    int packInit = 4;
    //OutputDebugString(L"New Packet Received:\n");
}

```

```

        if (recData[4] == 0x00 && recData[5]
== 0x00 && recData[9] == 0x00) {
            //odprintf("Empty packet,
just command source.");
            return;
        }
        for (packInit = 4; packInit < 61;
packInit += 4) {
            int packs[3] = {
recData[packInit + 1], recData[packInit +
2], recData[packInit + 3] };
            int signal2 = ((packs[0] &
0x03) << 8) + packs[2];
            int signal1 = ((packs[0] &
0b00111100) << 6) + packs[1];
            /*if (std::abs(signal1 -
lastPhSignal) > 8 && (lastTimeKey - fixed-
Time) > 0.015) {
                signal1 = lastPhSig-
nal;
                fixedTime =
lastTimeKey;
            }
            else if ((lastTimeKey -
fixedTime) < 0.012 && (lastTimeKey - fixed-
Time) > 0.00) {
                signal1 = lastPhSig-
nal;
            }*/
            /*if (std::abs(signal1 -
lastPhSignal) > 8 && (lastTimeKey - refixed-
Time) > 0.04) {
                signal1 = lastPhSig-
nal;
                refixedTime =
lastTimeKey;
            }*/
            if (std::abs(signal2 -
lastPrSignal) > 5 && (lastTimeKey - fixed-
Time2) > 0.05) {
                signal2 = lastPrSig-
nal;
                fixedTime2 =
lastTimeKey;
            }
            else if (std::abs(signal2 -
lastPrSignal) > 4 && (lastTimeKey - fixed-
Time2) < 0.03 && (lastTimeKey - fixedTime2)
> 0.00) {
                signal2 = lastPrSig-
nal;
            }
            //odprintf("packs: %d %d
%d", packs[0], packs[1], packs[2]);
            if (ui.invPhChk->is-
Checked()) signal1 = 1025 - signal1;
            if (ui.filterChk->is-
Checked()) {
                QtUSBGraph::addNewSigSample(signal1,
signal2);
                signal1 = filterLow-
Pass(phSignal, ui.nAvgBox->value());
                signal2 = filterLow-
Pass(prSignal, ui.nAvgBox->value());
            }
            if (ui.filterLP2Chk->is-
Checked()) {
                signal1 =
highOrderLPFilter(signal1);
            }
            //odprintf("%d", signal1);
            //int signal2 =
recData[packInit + 3];

            int timeStamp;
            timeStamp =
(recData[packInit]);
            if (timeStamp > 240) {
                incrStamp = TRUE;
            }
            if (incrStamp && timeStamp <
15) {
                incrStamp = FALSE;
                timeStampMultiplier
+= 1;
            }
            timeStamp = timeStamp +
timeStampMultiplier * 256;
            if (recData[packInit] == 0
&& recData[packInit + 1] == 0 &&
recData[packInit + 2] == 0 &&
recData[packInit + 3] == 0) timeStamp = 0;
            //odprintf("%d", timeStamp);
            double const timeScale =
(T0_MIN_OFFP * T0_PRESCALER*T0_PL_Adj);
            //Time Step in msec
            //Change this for Real In-
formation to 1000000
            if (timeStamp != 0){
                timeKey = ((time-
Scale * timeStamp) / 1000) / timeCorrection;
                //odprintf("Data 0:
%d - PhData: %d - tStamp: %d - Time: %4.2f -
nOfSamples: %d",signal1,sig-
nal2,timeStamp,timeKey,
nOfSamples);
                // add data to
lines:
                ui.QCPlot1-
>graph(0)->addData(timeKey, signal1);
                ui.QCPlot1-
>graph(1)->addData(timeKey, (signal2 -
prSigOffset)*prScaleMMHG);
                ui.sigText1-
>setText(QString::number(signal1));
                ui.sigText2-
>setText(QString::number((signal2 -
prSigOffset)*prScaleMMHG));
                ui.sigText1->se-
tAlignment(Qt::AlignRight);
                ui.sigText2->se-
tAlignment(Qt::AlignRight);
                // rescale value
(vertical) axis to fit the current data:
                //ui->customPlot-
>graph(0)->rescaleValueAxis();
                //ui->customPlot-
>graph(1)->rescaleValueAxis(true);
                phAverageSumm +=
signal1;
                prAverageSumm +=
signal2;
                if (onPeak && sig-
nal1 > maxPhPeak) maxPhPeak = signal1;
                lastTimeKey = time-
Key;
                lastPrSignal = sig-
nal2;
                lastPhSignal = sig-
nal1;
                for (int i = 29; i >
0; i--) {
                    phSignal-
Trail[i] = phSignalTrail[i - 1];
                    prSignal-
Trail[i] = prSignalTrail[i - 1];
                }
                phSignalTrail[0] =
signal1;

```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```

        prSignalTrail[0] =
signal2;

        sampleCountPh++;
        sampleCountPr++;
        nOfSamples++;
    }
    if(timestamp !=
lastTimeStamp + 1){
        deprPoints +=
(timestamp - lastTimeStamp);
        //odprintf("Missing
data point: %d / %d - Package: %d",
timestamp, lastTimeStamp, packInit);
    }
    lastTimeStamp = timestamp;
}

//odprintf("N of Iters per PACK:
%d", recData[3]);
//odprintf("Sample adding com-
pleted!");
// make key axis range scroll with
the data (at a constant range size of 8):

// calculate frames per second:

++frameCount;
static float lastCorrTime = 0.0;
static clock_t initTime = clock();
static double timestamp = (dou-
ble)initTime / (double)CLOCKS_PER_SEC;

if (timeKey - lastFpsKey > 0.05) //
average fps over 2 seconds
{
    //timeCorrection =
timeElapsed / (timeKey - lastFpsKey);
    //lastClock = loopClock;
    //odprintf("Signal Point De-
rivative = %f - %f", phSigDeriv, timeKey);
    ui.QCPlot1->xAxis-
>setRange(timeKey, xAxisRange, Qt::Align-
Center);

    ui.QCPlot1->replot();
    sampleFreq = (nOfSamples -
lastSampleSize) / (timeKey - lastFpsKey);
    ui.statusBar->showMessage(
        QString("%1 FPS, To-
tal Data points: %2 Sample Freq: %3 Missing
Data Points: %4")
        .arg(frameCount /
(timeKey - lastFpsKey), 0, 'f', 0)
        .arg(nOfSamples)
        .arg(sampleFreq)
        .arg(deprPoints)
        , 0);
    lastFpsKey = timeKey;
    lastSampleSize = nOfSam-
ples;
    frameCount = 0;
    if (timeKey - lastCorrTime >
2.0) {
        clock_t timer =
clock();
        double nowStamp =
(double)timer / (double)CLOCKS_PER_SEC;
        double diffTime =
nowStamp - timestamp;
        double timeCorr =
(nOfSamples / diffTime) / sampleFreq;
        //odprintf("time
corr: %f", timeCorr);
    }

    //timestamp =
nowStamp;
    lastCorrTime = time-
Key;
}

}
sendCMD = TRUE;
//odprintf("Timer Read Slot Com-
pleted");
}

void QtUSBGraph::prPeakDetect() {
    static bool onPrPeak = FALSE;
    static int minPrPoint = 500;
    float prSigAvg = 0;
    for (int k = 1; k < 30; k++) prSig-
Avg += prSignalTrail[k];
    prSigAvg = prSigAvg / 29;
    prSigDeriv = lastPrSignal - prSig-
Avg;
    if (!onPrPeak && prSigDeriv > 5.0) {
        onPrPeak = TRUE;
        minPrPoint = lastPrSignal;
        for (int i = nOfPeaks - 1; i
> 0; i--) {
            prPulseData[i] =
prPulseData[i - 1];
        }
        prPulseData[0].timeKey =
lastTimeKey;
        prPulseData[0].pxPoint =
lastPhSignal;
        prPulseData[0].pxAverage =
(prAverageSumm / sampleCountPr);
        sampleCountPr = prAver-
ageSumm = 0;
        prPulseData[0].HRFreq = 60 /
(prPulseData[1].timeKey - lastTimeKey);
        plotPrHeartFreq();
    }
    else if (onPrPeak && lastPrSignal <=
minPrPoint) {
        onPrPeak = FALSE;
    }
    else if ((lastTimeKey -
prPulseData[0].timeKey) > 2.0){
        onPrPeak = FALSE;
        minPrPoint = 500;
    }
}

int QtUSBGraph::findPhSetpoint(int medPr){
    int avgPress = 0;
    for (int i = 0; i < 5; i++) {
        avgPress +=
prPulseData[i].pxAverage / 5;
    }
    if (avgPress != medPr) {
        int varPh = (avgPress -
medPr)*0.05;
        return varPh;
    }
    else {
        return 0;
    }
}

void QtUSBGraph::phPeakDetect() {
    static float lastPhPeak = 0, init-
PhPeak = 0;
    float phSigAvg = 0;
}

```

```

        for (int k = 1; k < 30; k++) phSig-
Avg += phSignalTrail[k];
        phSigAvg = phSigAvg / 29;
        phSigDeriv = lastPhSignal - phSig-
Avg;
        if (!onPeak && phSigDeriv > 2.0 &&
(lastTimeKey - phPulseData[0].timeKey) >
0.40) {
            float preampHR = 60 /
(lastTimeKey - phPulseData[0].timeKey);
            if (preampHR > 200) return;
            onPeak = TRUE;
            for (int i = nOfPeaks - 1; i
> 0; i--) {
                phPulseData[i] =
phPulseData[i - 1];
            }
            phPulseData[0].pxAverage =
phAverageSumm / sampleCountPh;
            phAverageSumm = sam-
pleCountPh = 0;
            initPhPeak = phSignal-
Trail[0];
            for (int j = 1; j < 30; j++)
            {
                if (initPhPeak >
phSignalTrail[j]) initPhPeak = phSignal-
Trail[j];
            }
            phPulseData[0].timeKey =
lastTimeKey;
            heartFreq = 0;
            for (int i = 0; i < 5; i++)
            {
                heartFreq +=
(phPulseData[i].timeKey - phPulseData[i +
1].timeKey) / 5;
            }
            heartFreq = (60 / heart-
Freq);
        }
        else if (onPeak && phSigDeriv < 0) {
            onPeak = FALSE;
            lastPhPeak = maxPhPeak;
            phPulseData[0].pxHeight =
maxPhPeak - initPhPeak;
            phPulseData[0].pxPoint =
lastPrSignal;
            phPulseData[0].HRFreq =
heartFreq;
            maxPhPeak = 0;
            plotFFTSig();
            plotPhHeartFreq();
        }
        if (phPulseData[0].timeKey -
lastTimeKey > 2.0) {
            onPeak = FALSE;
            lastPhPeak = 0;
            initPhPeak = 0;
            phAverageSumm = sam-
pleCountPh = 0;
        }
    }
}

void QtUSBGraph::reTunePIDs(bool onDynamic)
{
    float kp, kd, ki, iLimit;
    if (!onDynamic) {
        kp = 1.9; ki = 0.005; kd =
0.1; iLimit = 10000;
    }
    else {
        kp = 2.5; ki = 0.00; kd =
0.0; iLimit = 10000;
    }
    pid_set_gains(&valHiPIDCon, kp, ki,
kd);
    pid_set_integral_limit(&valHiPIDCon,
iLimit);
    pid_set_frequency(&valHiPIDCon, sam-
pleFreq / 15);
    if (!onDynamic) {
        kp = 0.30; ki = 0.000; kd =
0.005; iLimit = 10000;
    }
    else {
        kp = 0.70; ki = 0.040; kd =
0.01; iLimit = 10000;
    }
    pid_set_gains(&valLoPIDCon, kp, ki,
kd);
    pid_set_integral_limit(&valLoPIDCon,
iLimit);
    pid_set_frequency(&valLoPIDCon, sam-
pleFreq / 15);
    if (!onDynamic) {
        kp = 30.0; ki = 1000.0; kd =
0.0; iLimit = 50000;
    }
    else {
        kp = 15.0; ki = 5.0; kd =
0.0; iLimit = 10000;
    }
    pid_set_gains(&pumpPIDCon, kp, ki,
kd);
    pid_set_integral_limit(&pumpPIDCon,
iLimit);
    pid_set_frequency(&pumpPIDCon, sam-
pleFreq / 15);
    pid_reset_integral(&valLoPIDCon);
    pid_reset_integral(&valHiPIDCon);
}

/*
void QtUSBGraph::reTunePIDs(bool onDynamic)
{
    float kp, kd, ki, iLimit;
    if (!onDynamic) {
        kp = 0.40; ki = 0.25; kd = 0.25; iLimit =
10000;
    }
    else {
        kp = 5.0; ki = 0.05; kd = 0.0; iLimit =
10000;
    }
    pid_set_gains(&valHiPIDCon, kp, ki, kd);
    pid_set_integral_limit(&valHiPIDCon,
iLimit);
    pid_set_frequency(&valHiPIDCon, sampleFreq /
15);
    if (!onDynamic) {
        kp = 1.0; ki = 0.5; kd = 0.015; iLimit =
10000;
    }
    else {
        kp = 0.45; ki = 0.0; kd = 0.00; iLimit =
10000;
    }
    pid_set_gains(&valLoPIDCon, kp, ki, kd);
}

```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```

pid_set_integral_limit(&valLoPIDCon,
iLimit);
pid_set_frequency(&valLoPIDCon, sampleFreq /
15);

if (!onDynamic) {
kp = 3.0; ki = 1.0; kd = 0.0; iLimit = 5000;
}
else {
kp = 1.0; ki = 0.0; kd = 0.0; iLimit = 1000;
}

pid_set_gains(&pumpPIDCon, kp, ki, kd);
pid_set_integral_limit(&pumpPIDCon, iLimit);
pid_set_frequency(&pumpPIDCon, sampleFreq /
15);

pid_reset_integral(&valLoPIDCon);
pid_reset_integral(&valHiPIDCon);
}

*/

void QtUSBGraph::prPIDControl() {
signed int refError;
float calcVar;
float valHiPulsOutput, valLoPulsOut-
put, pumpDutyOutput;
int prOffset = lastPrSignal -
prSigOffset + 50;

refError = lastPrSignal - prRef;
//if (conPhRef) refError = phRef -
lastPhSignal;
valHiPulsOutput = pid_pro-
cess(&valHiPIDCon, refError);
valLoPulsOutput = pid_process(&val-
LoPIDCon, -refError);
if (refError < -10) valLoPulsOutput
*= 0.05;
//pumpDutyOutput = pid_pro-
cess(&pumpPIDCon, refError);
if (!conPhRef) calcVar = (prOffset /
180.00) * (5 - refError) * 0.70 + 9.50;
if (conPhRef) calcVar = (prOffset /
180.00) * (5 - refError) * 0.70 + 15.50;

pumpDutyOutput = calcVar;

if (valHiPulsOutput < 0)
valHiPulsOutput = 0;
if (valLoPulsOutput < 0) valLoPul-
sOutput = 0;
valHiLock = (int)valHiPulsOutput;
valLoLock = (int)valLoPulsOutput;
pumpDutyLock = (int)(pumpDutyOut-
put);
if (valHiLock > 255) valHiLock =
255;
if (valLoLock > 255) valLoLock =
255;
if (pumpDutyLock > 30) pumpDutyLock
= 30;
if (pumpDutyLock < 0) pumpDutyLock =
0;
//if (std::abs(refError) > 5) od-
printf("Delay counter++ %d", refError);
}

void QtUSBGraph::prPIDControl2() {
prPeakDetect();
signed int refError;

float calcVar;
float valHiPulsOutput, valLoPulsOut-
put, pumpDutyOutput;
int prOffset = lastPrSignal -
prSigOffset + 50;

refError = (phRef - lastPhSignal) +
predictiveLoop();
//if (conPhRef) refError = phRef -
lastPhSignal;
valHiPulsOutput = pid_pro-
cess(&valHiPIDCon, refError);
valLoPulsOutput = pid_process(&val-
LoPIDCon, -refError);
//if (refError < -10) valLoPulsOut-
put *= 0.05;
//pumpDutyOutput = pid_pro-
cess(&pumpPIDCon, refError);
//calcVar = (prOffset / 180.00) * (5
- refError) * 0.70 + 45.50;

pumpDutyOutput = 25.0;

if (valHiPulsOutput < 0)
valHiPulsOutput = 0;
if (valLoPulsOutput < 0) valLoPul-
sOutput = 0;
valHiLock = (int)valHiPulsOutput;
valLoLock = (int)valLoPulsOutput;
pumpDutyLock = (int)(pumpDutyOut-
put);
if (valHiLock > 255) valHiLock =
255;
if (valLoLock > 255) valLoLock =
255;
if (pumpDutyLock > 30) pumpDutyLock
= 30;
if (pumpDutyLock < 0) pumpDutyLock =
0;
//if (std::abs(refError) > 5) od-
printf("Delay counter++ %d", refError);
}

int QtUSBGraph::predictiveLoop() {
static bool predict = FALSE;
if ((lastTimeKey -
prPulseData[2].timeKey) > 3.5) return 0;
double pulsePeriod[3];
for (int i = 0; i < 3; i++) {
pulsePeriod[i] =
prPulseData[i].timeKey - prPulseData[i +
1].timeKey;
}
double periodError = pulsePeriod[0]
* 2 - pulsePeriod[1] - pulsePeriod[2];
if (std::abs(periodError) < 0.15) {
predPeriod = pulsePeriod[0];
predict = TRUE;
}
else {
predict = FALSE;
}
int addError;
double nextPulseTime;
if (std::abs(prPulseData[0].timeKey
- lastTimeKey) < 1.2) {
nextPulseTime =
prPulseData[0].timeKey + predPeriod;
}
else {
predict = FALSE;
}
}

```

```

        double pulseTimeLeft = nextPulseTime
- lastTimeKey;
        if (predict && pulseTimeLeft < 0.2
&& pulseTimeLeft > 0.1) {
            addError = 0 * (10 *
std::abs(pulseTimeLeft - 0.1));
            //odprintf("prediction error
summed: %d", addError);
            return -addError;
        }
        else {
            return 0;
        }
    }
}

void QtUSBGraph::reCalPhSetpoint(bool reset)
{
    static double prevCalTime = 0;
    if (reset) {
        prevCalTime = lastTimeKey;
        return;
    }
    if ((lastTimeKey - prevCalTime) >
5.0) {
        float avgPr =
(prPulseData[0].pxAverage +
prPulseData[1].pxAverage) / 2;
        if (std::abs(avgPr -
medPresSetpoint) > 8) {
            phRef +=
(int)((avgPr - medPresSetpoint)*0.15);
            odprintf("New Ph
Setpoint: %d - Time: %f", phRef, lastTime-
Key);
        }
        prevCalTime = lastTimeKey;
    }
}

void QtUSBGraph::plotFFTSig() {
    //Computes the FFT for the last
Heart Beat Detected.
    double initTime =
phPulseData[1].timeKey;
    double endTime =
phPulseData[0].timeKey;
    int indexInit = ui.QCPlot1-
>graph(0)->findBegin(initTime);
    int indexEnd = ui.QCPlot1->graph(0)-
>findEnd(endTime);
    int nPoints = 0;
    QCPGraphDataContainer::const_itera-
tor begin = ui.QCPlot1->graph(0)->data()-
>at(indexInit);
    QCPGraphDataContainer::const_itera-
tor end = ui.QCPlot1->graph(0)->data()-
>at(indexEnd);
    double minVal = 1500;
    for (QCPGraphDataCon-
tainer::const_iterator it = begin; it !=
end; ++it) {
        fftInArray[nPoints] = it-
>value;
        if (it->value < minVal) min-
Val = it->value;
        if (nPoints++ == DFT_SIZE)
it = end-1;
    }
    for (int k = 0; k < nPoints; k++)
fftInArray[k] = fftInArray[k] - minVal;
    for (int j = nPoints; j < DFT_SIZE;
j++) fftInArray[j] = 0;
    memset(fftOutArray, 0,
sizeof(fftOutArray));
}

//now the in array contains the pad-
ded data points for the heart beat signal;
fftw_execute(fftwPlan);
ui.QCPlot2->graph(0)->data()-
>clear();
double freqRes = (sampleFreq /
DFT_SIZE);
for (int i = 0; i < DFT_SIZE; i++) {
    double freq = i * freqRes;
    double powSpectrum =
sqrt(fftOutArray[i][0] * fftOutArray[i][0] +
fftOutArray[i][1] * fftOutArray[i][1]);
    //if (freq < 0.5) powSpec-
trum *= freq*freq;
    ui.QCPlot2->graph(0)-
>addData(freq, powSpectrum);
}
ui.QCPlot2->rescaleAxes();
ui.QCPlot2->xAxis->setRangeUp-
per(30.0);
ui.QCPlot2->yAxis->setRange-
Lower(500.0);
ui.QCPlot2->replot();
}

void QtUSBGraph::plotPhHeartFreq() {
    QModelIndex index = tvModel->in-
dex(0, 0, QModelIndex());
    int row = ui.hbTableView->currentIn-
dex().row();
    int newRow = row + 1;
    QList<QStandardItem *> list =
tvModel->takeRow(row);
    tvModel->insertRow(newRow, list);

    // 0 for all data
    tvModel->setData(index, Qt::Align-
Center, Qt::TextAlignmentRole);
    tvModel->setData(index,
phPulseData[0].timeKey);
    index = tvModel->index(0, 1, QModelIndex());
    tvModel->setData(index, Qt::Align-
Center, Qt::TextAlignmentRole);
    tvModel->setData(index,
((phPulseData[0].pxPoint - prSigOffset) *
prScaleMMHG));
    index = tvModel->index(0, 2, QModelIndex());
    tvModel->setData(index, Qt::Align-
Center, Qt::TextAlignmentRole);
    tvModel->setData(index,
(int)phPulseData[0].HRFreq);
    index = tvModel->index(0, 3, QModelIndex());
    tvModel->setData(index, Qt::Align-
Center, Qt::TextAlignmentRole);
    tvModel->setData(index,
phPulseData[0].pxHeight);
    index = tvModel->index(0, 4, QModelIndex());
    tvModel->setData(index, Qt::Align-
Center, Qt::TextAlignmentRole);
    tvModel->setData(index,
phPulseData[0].pxAverage);
}

void QtUSBGraph::plotPrHeartFreq() {
    QModelIndex index = tvModel2->in-
dex(0, 0, QModelIndex());
    int row = ui.hbTableView2->cur-
rentIndex().row();
}

```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```

        int newRow = row + 1;
        QList<QStandardItem *> list =
tvModel1->takeRow(row);
        tvModel2->insertRow(newRow, list);

        // 0 for all data
        tvModel2->setData(index, Qt::Align-
Center, Qt::TextAlignmentRole);
        tvModel2->setData(index,
prPulseData[0].timeKey);
        index = tvModel2->index(0, 1, QMod-
elIndex());
        tvModel2->setData(index, Qt::Align-
Center, Qt::TextAlignmentRole);
        tvModel2->setData(index,
prPulseData[0].pxPoint);
        index = tvModel2->index(0, 2, QMod-
elIndex());
        tvModel2->setData(index, Qt::Align-
Center, Qt::TextAlignmentRole);
        tvModel2->setData(index,
(int)prPulseData[0].HRFreq);
        index = tvModel2->index(0, 3, QMod-
elIndex());
        tvModel2->setData(index, Qt::Align-
Center, Qt::TextAlignmentRole);
        tvModel2->setData(index,
prPulseData[0].pxHeight);
        index = tvModel2->index(0, 4, QMod-
elIndex());
        tvModel2->setData(index, Qt::Align-
Center, Qt::TextAlignmentRole);
        tvModel2->setData(index,
((prPulseData[0].pxAverage - prSigOffset) *
prScaleMMHG));
    }

void QtUSBGraph::openPlotFile() {
    if (dataTimer->isActive()) return;
    QString fileName = QFileDialog::get-
OpenFileName(this,
        tr("Open Plot File"), "",
        tr("Text Files (*.txt);All
Files (*)"));
    QFile openFile(fileName);
    if (openFile.open(QFile::ReadOnly))
    {
        ui.QCPlot1->graph(0)-
>data()->clear();
        ui.QCPlot1->graph(1)-
>data()->clear();
        QTextStream stream(&open-
File);
        while (!stream.atEnd()) {
            QString line =
stream.readLine();
            if (line ==
QString(" -----")) break;
        }
        while (!stream.atEnd()) {
            QString line =
stream.readLine();
            if (line ==
QString(" ")) break;
            QStringList pair =
line.split(" ");
            double time =
pair[0].toDouble();
            int dato =
pair[1].toInt();

            ui.QCPlot1-
>graph(0)->addData(time, dato);
        }

        QString line = stream.read-
Line();
        line = stream.readLine();
        line = stream.readLine();
        while (!stream.atEnd()) {
            QString line =
stream.readLine();
            QStringList pair =
line.split(" ");
            double time =
pair[0].toDouble();
            double dato =
pair[1].toDouble();
            ui.QCPlot1-
>graph(1)->addData(time, dato);
        }
        ui.QCPlot1->rescaleAxes();
        ui.QCPlot1->replot();
    }
}

void QtUSBGraph::saveSelectedPlot() {
    QCPDataSelection phSelection =
ui.QCPlot1->graph(0)->selection();
    QCPDataSelection prSelection =
ui.QCPlot1->graph(1)->selection();
    if (phSelection.isEmpty() && prSe-
lection.isEmpty()) {
        QMessageBox::StandardButton
warning;
        const QString noDeviceEN =
"Warning: No selected data points!";
        const QString noDeviceET =
"You have not se-
lected any plot points to save!\n"
        "You can select en-
tire graphs by clicking on them, or select
sections by holding down the Ctrl Key and
dragging a rectangular selection box over
the plot.";
        warning = QMessageBox::warn-
ing(this, noDeviceEN, noDeviceET, QMessage-
Box::Ok);
        return;
    }

    QString fileName = QFileDialog-
::getSaveFileName(this,
        tr("Save selected Plot
Points"), "",
        tr("Text Files (*.txt);All
Files (*)"));
    QFile saveFile(fileName);
    if (saveFile.open(QFile::WriteOnly |
QFile::Text)) {
        QTextStream stream(&save-
File);
        stream << QString("Signal
Sampler Record Plot Points:\n");
        stream << QString(" \n \n
Photopleysmograph Signal: \n Time Key - Ph
Value \n -----");
        foreach(QCPDataRange data-
Range, phSelection.dataRanges()) {
            QCPGraphDataCon-
tainer::const_iterator begin = ui.QCPlot1-
>graph(0)->data()->at(dataRange.begin());
            QCPGraphDataCon-
tainer::const_iterator end = ui.QCPlot1-
>graph(0)->data()->at(dataRange.end());

```

```

        for
(QCPGraphDataContainer::const_iterator it =
begin; it != end; ++it) {
            stream <<
QString("%1 %2\n").arg(it->key).arg((int)it-
>value);
        }
        stream << QString(" \n \n
Pressure Signal: \n Time Key - Pressure
(mmHg) \n ----- \n");
        foreach(QCPDataRange data-
Range, prSelection.dataRanges()) {
            QCPGraphDataCon-
tainer::const_iterator begin = ui.QCPlot1-
>graph(1)->data()->at(dataRange.begin());
            QCPGraphDataCon-
tainer::const_iterator end = ui.QCPlot1-
>graph(1)->data()->at(dataRange.end());
            for
(QCPGraphDataContainer::const_iterator it =
begin; it != end; ++it) {
                stream <<
QString("%1 %2\n").arg(it->key).arg(it-
>value);
            }
            saveFile.close();
        }
}

void QtUSBGraph::packSignals() {
    static int samplePH;
    static int samplePR;
    static float sampleKey;
    samplePH = lastPhSignal;
    samplePR = lastPrSignal;
    sampleKey = lastTimeKey;
    if(lastTimeKey > 1.0) phPeak-
Detect();
    //odprintf("Signal Der = %f , %f",
phSigDeriv, sampleKey);
    //odprintf(" Proc Signals: %d - %d
", samplePH, samplePR);
    unsigned char phLSB = (unsigned
char)samplePH;
    unsigned char prLSB = (unsigned
char)samplePR;
    unsigned char combMSB = ((unsigned
char)(samplePR >> 8));
    unsigned char sampMSB = ((unsigned
char)(samplePH >> 8));
    unsigned char nVPByte = (unsigned
char)nValPuls;
    unsigned char valHiLByte = (unsigned
char)valHiLock;
    unsigned char valLoLByte = (unsigned
char)valLoLock;
    unsigned char pDLByte = (unsigned
char)pumpDutyLock;
    //odprintf("PrRef: %d - prSig: %d -
pumpDL: %d - nVPulse: %d", prRef, lastPrSig-
nal, pDLByte, nVPByte);
    combMSB = combMSB | (sampMSB << 2);

    cmdBuff[2] = (unsigned
char*)combMSB;
    cmdBuff[3] = (unsigned char*)phLSB;
    cmdBuff[4] = (unsigned char*)prLSB;
    int refVal;
    if (conPrRef) refVal = prRef;

    if (cmdBuff[1] == (unsigned
char*)CMD_CON_PR || cmdBuff[1] == (unsigned
char*)CMD_CON_PH) {
        unsigned char prRefLSB =
(unsigned char)refVal;
        unsigned char sampRefMSB =
(unsigned char)(refVal >> 8);
        combMSB = combMSB | (sam-
pRefMSB << 5);
        cmdBuff[2] = (unsigned
char*)combMSB;
        cmdBuff[5] = (unsigned
char*)prRefLSB;
    }
    cmdBuff[6] = (unsigned char*)pDL-
Byte;
    cmdBuff[7] = (unsigned
char*)nVPByte;
    cmdBuff[7] = (unsigned char*)valHiL-
Byte;
    cmdBuff[8] = (unsigned char*)valLoL-
Byte;
    dataSent[0] = lastPhSignal;
    dataSent[1] = lastPrSignal;
    if (conPrRef || conPhRef) da-
taSent[2] = refVal;
}

void QtUSBGraph::initCalibr() {
    cmdTimer->start(50);
    cmdState = PROC_CALIBR;
}

void QtUSBGraph::controlProcess() {
    static int offsetSamples = 0;
    static int preOffsetCal = 0;
    static pulseDataSet indexedPulse;
    static int pulseIndexInit;
    static int riseIndex = 0, fallIndex
= 0;
    static float startRiseKey, start-
FallKey;
    static QMessageBox::StandardButton
confirm;

    switch (cmdState) {
    case PROC_CALIBR:
        confirm = QMessageBox::No;
        if (!conPrRef) {
            conPrRef = TRUE;
            reTunePIDs(false);
            prRef = 450;
            cmdBuff[1] = (un-
signed char*)CMD_CON_PR;
            sendCMD = TRUE;
            break;
        }
        if (offsetSamples < 10) {
            preOffsetCal +=
lastPrSignal;
            offsetSamples++;
        }
        else {
            prSigOffset =
(preOffsetCal / 10) - 2;
            prRef = lastPrSig-
nal;
            conPrRef = TRUE;
            reTunePIDs(false);
            conPhRef = FALSE;
            cmdState =
PROC_CAL_RISE;
            riseCalOK = TRUE;
        }
    }
}

```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```

        startRiseKey =
lastTimeKey;
    }
    break;
    case PROC_CAL_RISE:
        if (lastPrSignal >= 850) {
            cmdState =
PROC_CAL_MEDPR;
            startFallKey =
lastTimeKey;
        }
        else {
            prRef++;
            cmdBuff[1] = (un-
signed char*)CMD_CON_PR;
            sendCMD = TRUE;
            conPrRef = TRUE;
            reTunePIDs(false);
            conPhRef = FALSE;
        }
        break;
    /*case PROC_CAL_FALL:
        if (lastPrSignal <= 550) {
            cmdBuff[1] = (un-
signed char*)CMD_NO_CON;
            sendCMD = TRUE;
            conPrRef = FALSE;
            cmdState =
PROC_CAL_MEDPR;
        }
        else {
            prRef--;
            cmdBuff[1] = (un-
signed char*)CMD_CON_PR;
            sendCMD = TRUE;
            conPrRef = TRUE;
            reTunePIDs(false);
            conPhRef = FALSE;
        }
        break;
    */case PROC_CAL_MEDPR:
        for (int i = 0; i <
nOfPeaks; i++) {
            if ((startRiseKey <
phPulseData[i].timeKey) &&
(phPulseData[i].timeKey < lastTimeKey)) {
                if
(phPulseData[i].pxHeight >
phPulseData[riseIndex].pxHeight) riseIndex =
i;
            }
        }
        odprintf("Detected highest
peaks: %f , %d and %f , %d ",
phPulseData[riseIndex].timeKey,
phPulseData[riseIndex].pxHeight,
phPulseData[fallIndex].timeKey,
phPulseData[fallIndex].pxHeight);

        medPresSetpoint =
phPulseData[riseIndex].pxPoint;

        indexedPulse =
phPulseData[0];
        prRef = medPresSetpoint;
        odprintf("Calculated PR Set:
%d", prRef);
        cmdBuff[1] = (unsigned
char*)CMD_CON_PR;
        sendCMD = TRUE;
        conPrRef = TRUE;
        reTunePIDs(false);
        conPhRef = FALSE;

        cmdState =
PROC_PRE_CAL_MEDPH;
        break;
        case PROC_PRE_CAL_MEDPH:
            if (lastTimeKey - in-
dexedPulse.timeKey < 2.5) return;
            for (int i = 0; i <
nOfPeaks; i++) {
                if (phPulseData[i]
== indexedPulse) {
                    od-
printf("Matching! Index point = %d", i);
                    pulseIndexI-
nit = i;
                    i =
nOfPeaks;
                }
            }
            cmdState = PROC_CAL_MEDPH;
            break;
            case PROC_CAL_MEDPH:
                if (phPulseData[pulseIndexI-
nit + 5] == indexedPulse) {
                    medPhSetpoint = 0;
                    odprintf("Using the
following pulses:");
                    for (int j = 0; j <
5; j++) {
                        /*od-
printf("T: %f - Height: %d - HR: %f - Aver-
age: %d",
                            phPulseData[j].timeKey,
                            phPulseData[j].pxHeight,
                            phPulseData[j].HRFreq,
                            phPulseData[j].pxAverage);*/
                        medPhSet-
point += (phPulseData[j].pxAverage);
                    }
                    medPhSetpoint =
medPhSetpoint / 5;
                    conPrRef = FALSE;
                    conPhRef = FALSE;
                    odprintf("Calibra-
tion complete:");
                    odprintf("PrSetpoint
= %d, PhSetpoint = %d", medPresSetpoint,
medPhSetpoint);
                    const QString
startVCMT = "Calibration complete! Begin
Volume Compensation?";
                    QString startVCMM =
QString("The
calibration process has finished. The fol-
lowing results were obtained.\n"
                        "Medium
Blood Pressure: %1 mmHg. Ph Medium Setpoint:
%2\n"
                        "Start Vol-
ume Compensation Method on these parame-
ters?").arg((medPresSetpoint - prSigOff-
set)*prScaleMMHG).arg(medPhSetpoint);
                    confirm = QMessageBox-
Box::question(this, startVCMT, startVCMM,
QMessageBox::Yes | QMessageBox::No);
                    if (confirm == QMes-
sageBox::Yes) {
                        cmdState =
PROC_CON_PH;
                        cmdBuff[1] =
(unsigned char*)CMD_CON_PH;

```

```

TRUE;
medPhSetpoint;
TRUE;
PIDs(true);
point(true);
TRUE;
    }
    else {
(unsigned char*)CMD_NO_CON;
TRUE;
0x00;
>stop();
FALSE;
    }
    }
    else {
        return;
    }
    break;
case PROC_CON_PH:
    reCalPhSetpoint(false);
    break;
}
}

void QtUSBGraph::startTimer() {
    cmdBuff[1] = (unsigned
char*)CMD_T0_ON;
    sendCMD = TRUE;
}
void QtUSBGraph::stopTimer() {
    cmdBuff[1] = (unsigned
char*)CMD_T0_OFF;
    sendCMD = TRUE;
}

void QtUSBGraph::changeModulate() {
    bool isChk = ui.modChk->isChecked();
    if (isChk) {
char*)CMD_DO_IR;
    }
    else {
char*)CMD_NO_IR;
    }
    sendCMD = TRUE;
}
void QtUSBGraph::changeSentData() {
    if (ui.stepRBtn1->isChecked()) {
char*)CMD_SEND_COMB;
    }
    else if (ui.stepRBtn2->isChecked())
{
        cmdBuff[1] = (unsigned
char*)CMD_SEND_HIGH;
    }
    else if (ui.stepRBtn3->isChecked())
{
        cmdBuff[1] = (unsigned
char*)CMD_SEND_LOW;
    }
}

sendCMD =
phRef =
conPhRef =
reTune-
reCalPhSet-
conPrRef =
cmdBuff[1] =
sendCMD =
cmdState =
cmdTimer-
conPrRef =
    }
}

sendCMD = TRUE;
}

void QtUSBGraph::togglePump() {
    cmdBuff[1] = (unsigned
char*)CMD_TOG_PUMP;
    odprintf("Toggling Pump...");
    sendCMD = TRUE;
}

void QtUSBGraph::swapVisiblePlots() {
    if (ui.plotSelectBox->currentText()
== QString("Ph Heart Beats")) {
        ui.QCPlot2->setVisi-
ble(FALSE);
        ui.hbTableView->setVisi-
ble(TRUE);
        ui.hbTableView2->setVisi-
ble(FALSE);
    }
    else if (ui.plotSelectBox->cur-
rentText() == QString("Pr Heart Beats")) {
        ui.QCPlot2->setVisi-
ble(FALSE);
        ui.hbTableView->setVisi-
ble(FALSE);
        ui.hbTableView2->setVisi-
ble(TRUE);
    }
    else {
        ui.QCPlot2->setVisi-
ble(TRUE);
        ui.hbTableView->setVisi-
ble(FALSE);
        ui.hbTableView2->setVisi-
ble(FALSE);
    }
}

void QtUSBGraph::toggleValve() {
    cmdBuff[1] = (unsigned
char*)CMD_TOG_VALV;
    odprintf("Toggling Valve...");
    sendCMD = TRUE;
}

}

void QtUSBGraph::conPress(){
    cmdBuff[1] = (unsigned
char*)CMD_CON_PR;
    odprintf("Setting pressure refer-
ence... ");
    prRef = ((int)ui.prRefBox->value())/
prScaleMMHG + prSigOffset;
    cmdState = 0x00;
    conPrRef = TRUE;
    reTunePIDs(false);
    conPhRef = FALSE;
    sendCMD = TRUE;
}

void QtUSBGraph::conPhSignal() {
    cmdBuff[1] = (unsigned
char*)CMD_CON_PH;
    odprintf("Setting ph signal refer-
ence... ");
    phRef = ((int)ui.phRefBox->value());
    cmdState = 0x00;
    conPrRef = TRUE;
    reTunePIDs(true);
    conPhRef = TRUE;
    sendCMD = TRUE;
}
}

```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```

void QtUSBGraph::releasePr() {
    cmdBuff[1] = (unsigned
char*)CMD_NO_CON;
    odprintf("Releasing Control...");
    sendCMD = TRUE;
    conPrRef = FALSE;
    conPhRef = FALSE;
    cmdState = 0x00;
}

void QtUSBGraph::connectDevice() {
    memset(phSignal, 0x00, maxAvgSize);
    int res;
    unsigned char buf[256];
#define MAX_STR 255
    wchar_t wstr[MAX_STR];
    //hid_device *handle;
    int i;

    struct hid_device_info *devs,
*cur_dev;

    devs = hid_enumerate(VID_NUM,
PID_NUM);
    cur_dev = devs;
    /*while (cur_dev) {
        odprintf("Device Found\n type:
%04hx %04hx\n path: %s\n serial_number:
%s", cur_dev->vendor_id, cur_dev->prod-
uct_id, cur_dev->path, cur_dev->serial_num-
ber);
        OutputDebugString(L"\n");
        odprintf(" Manufacturer: %ls\n",
cur_dev->manufacturer_string);
        odprintf(" Product: %ls\n",
cur_dev->product_string);
        odprintf(" Release: %hx\n",
cur_dev->release_number);
        odprintf(" Interface: %d\n",
cur_dev->interface_number);
        OutputDebugString(L"\n");
        cur_dev = cur_dev->next;
    }*/
    if (!cur_dev) {
        odprintf("No devices found
with the correct PID and VID \n");
        hid_free_enumeration(devs);
        QMessageBox::StandardButton
warning;
        const QString noDeviceEN =
"Error: No device found!";
        const QString noDeviceET =
"There are no con-
nected devices with the correct PID and VID
Identifications.\n"
        "Please try to re-
connect device.";
        warning = QMessageBox::warn-
ing(this, noDeviceEN, noDeviceET, QMessage-
Box::Ok);
        return;
    }
    hid_free_enumeration(devs);

    // Set up the command buffer.
    memset(buf, 0x00, sizeof(buf));
    buf[0] = 0x01;
    buf[1] = 0x81;

    handle = hid_open(VID_NUM, PID_NUM,
NULL);
    if (!handle) {
        odprintf("unable to open de-
vice\n");
        //return 1;
    }

    // Read the Manufacturer String
    wstr[0] = 0x0000;
    res = hid_get_manufac-
turer_string(handle, wstr, MAX_STR);
    if (res < 0)
        odprintf("Unable to read
manufacturer string\n");
    odprintf("Manufacturer String:
%ls\n", wstr);

    // Read the Product String
    wstr[0] = 0x0000;
    res = hid_get_product_string(handle,
wstr, MAX_STR);
    if (res < 0)
        odprintf("Unable to read
product string\n");
    odprintf("Product String: %ls\n",
wstr);

    // Read the Serial Number String
    wstr[0] = 0x0000;
    res = hid_get_serial_num-
ber_string(handle, wstr, MAX_STR);
    if (res < 0)
        odprintf("Unable to read se-
rial number string\n");
    odprintf("Serial Number String: (%d)
%ls", wstr[0], wstr);
    odprintf("\n");

    // Read Indexed String 1
    wstr[0] = 0x0000;
    res = hid_get_indexed_string(handle,
1, wstr, MAX_STR);
    if (res < 0)
        odprintf("Unable to read in-
dexed string 1\n");
    odprintf("Indexed String 1: %ls\n",
wstr);

    hid_set_nonblocking(handle, 1); //if
0 waits for a read data to arrive, 1 returns
00 if there's no data

    dataTimer->start();
    pid_init(&valHiPIDCon);
    pid_init(&valLoPIDCon);
    pid_init(&pumpPIDCon);
    pid_init(&phPIDCon);
    cmdBuff[1] = (unsigned
char*)CMD_CLR_TIME;
    sendCMD = TRUE;
    ui.startBtn->setEnabled(TRUE);
    ui.stopBtn->setEnabled(TRUE);
    ui.dcBtn->setEnabled(TRUE);
    ui.pumpToggleBtn->setEnabled(TRUE);
    ui.valveToggleBtn->setEnabled(TRUE);
    ui.connectBtn->setEnabled(FALSE);
    ui.calibrBtn->setEnabled(TRUE);
    ui.noPrConBtn->setEnabled(TRUE);
    ui.conPrBtn->setEnabled(TRUE);
    ui.prRefBox->setEnabled(TRUE);
    ui.conPhBtn->setEnabled(TRUE);
    ui.phRefBox->setEnabled(TRUE);
    ui.stepRBtn1->setEnabled(TRUE);
    ui.stepRBtn2->setEnabled(TRUE);
    ui.stepRBtn3->setEnabled(TRUE);
    ui.modChk->setEnabled(TRUE);
}

```

```

        ui.filterChk->setEnabled(TRUE);
        ui.filterLP2Chk->setEnabled(TRUE);
        ui.invPhChk->setEnabled(TRUE);
        ui.QCPlot1->graph(0)->data()-
>clear();
        ui.QCPlot1->graph(1)->data()-
>clear();
        fftPlan =
fftw_plan_dft_r2c_1d(DFT_SIZE, fftIn,
fftOut, FFTW_ESTIMATE);
}

void QtUSBGraph::disconnectDevice() {
    hid_close(handle);
    //cmdTimer->stop();
    dataTimer->stop();
    ui.startBtn->setEnabled(FALSE);
    ui.stopBtn->setEnabled(FALSE);
    ui.dcBtn->setEnabled(FALSE);
    ui.pumpToggleBtn->setEnabled(FALSE);
    ui.valveToggleBtn->setEn-
abled(FALSE);
    ui.noPrConBtn->setEnabled(FALSE);
    ui.conPrBtn->setEnabled(FALSE);
    ui.prRefBox->setEnabled(FALSE);
    ui.conPhBtn->setEnabled(FALSE);
    ui.phRefBox->setEnabled(FALSE);
    ui.connectBtn->setEnabled(TRUE);
    ui.calibrBtn->setEnabled(FALSE);
    ui.stepRBtn1->setEnabled(FALSE);
    ui.stepRBtn2->setEnabled(FALSE);
    ui.stepRBtn3->setEnabled(FALSE);
    ui.modChk->setEnabled(FALSE);
    ui.filterChk->setEnabled(FALSE);
    ui.filterLP2Chk->setEnabled(FALSE);
    ui.invPhChk->setEnabled(FALSE);
    fftw_destroy_plan(fftPlan);
}

void QtUSBGraph::showGraphs() {
    ui.QCPlot1->graph(1)->setVisi-
ble(ui.sig1VisChk->isChecked());
    ui.QCPlot1->graph(0)->setVisi-
ble(ui.sig2VisChk->isChecked());
    ui.QCPlot1->replot();
}

void QtUSBGraph::addNewSigSample(int phSam-
ple, int prSample) {
    for (int i = maxAvgSize - 1; i > 0;
i--) {
        phSignal[i] = phSignal[i -
1];
        prSignal[i] = prSignal[i -
1];
    }
    phSignal[0] = phSample;
    prSignal[0] = prSample;
}

void QtUSBGraph::setPlotBg() {
    if (ui.whiteBgChk->isChecked()) {

        ui.QCPlot1->setBack-
ground(Qt::white);
        ui.QCPlot2->setBack-
ground(Qt::white);
        ui.QCPlot1->xAxis->setTick-
LabelColor(Qt::black);
        ui.QCPlot1->yAxis->setTick-
LabelColor(QColor(Qt::red));
        ui.QCPlot1->yAxis2->setTick-
LabelColor(QColor(Qt::blue));

        ui.QCPlot2->xAxis->setTick-
LabelColor(Qt::black);
        ui.QCPlot2->yAxis->setTick-
LabelColor(Qt::black);

        ui.QCPlot1->xAxis->set-
BasePen(QPen(Qt::black));
        ui.QCPlot1->yAxis2->set-
BasePen(QPen(QColor(Qt::blue)));
        ui.QCPlot1->yAxis->set-
BasePen(QPen(QColor(Qt::red)));

        ui.QCPlot2->xAxis->set-
BasePen(QPen(Qt::black));
        ui.QCPlot2->yAxis->set-
BasePen(QPen(Qt::black));

        //ui.QCPlot1->yAxis2-
>setLabel(QString("Signal"));
        ui.QCPlot1->xAxis-
>setLabelColor(QColor(Qt::black));
        ui.QCPlot1->yAxis->setLabel-
Color(QColor(Qt::red));
        ui.QCPlot1->yAxis2-
>setLabelColor(QColor(Qt::blue));
        ui.QCPlot2->xAxis-
>setLabelColor(QColor(Qt::black));
        ui.QCPlot2->yAxis-
>setLabelColor(QColor(Qt::black));
        ui.QCPlot1->graph(0)-
>setPen(QPen(QColor(Qt::blue), 3));

        // -> ui.QCPlot1->graph(0)-
>selected();
        ui.sigText1-
>setTextColor(QColor(Qt::blue));
        ui.sigText2-
>setTextColor(QColor(255, 0, 0, 200));
        ui.sigText2->setTextBack-
groundColor(QColor(Qt::white));
        ui.sigText1->setTextBack-
groundColor(QColor(Qt::white));

        QPen gridPen =
QPen(QColor(Qt::black));
        gridPen.setStyle(Qt::Dot-
Line);
        gridPen.setWidth(1);

        //ui.QCPlot1->yAxis->grid()-
>setSubGridVisible(true);
        //ui.QCPlot1->xAxis->grid()-
>setSubGridVisible(true);
        ui.QCPlot1->yAxis->grid()-
>setPen(gridPen);
        ui.QCPlot1->xAxis->grid()-
>setPen(gridPen);

        ui.QCPlot2->graph(0)-
>setPen(QPen(QColor(0, 128, 0), 3));
    }
    else {
        ui.QCPlot1->setBack-
ground(Qt::black);
        ui.QCPlot2->setBack-
ground(Qt::black);
        ui.QCPlot1->graph(0)-
>setPen(QPen(QColor(Qt::green), 3));

        ui.QCPlot1->xAxis->setTick-
LabelColor(Qt::white);
        ui.QCPlot1->yAxis->setTick-
LabelColor(QColor(255, 110, 40));
    }
}

```

## Desarrollo de Dispositivo para Medición de Presión Arterial No Invasiva Continua

```
        ui.QCPlot1->yAxis2->setTick-
LabelColor(QColor(Qt::green));
        ui.QCPlot2->xAxis->setTick-
LabelColor(Qt::white);
        ui.QCPlot2->yAxis->setTick-
LabelColor(Qt::white);

        ui.QCPlot1->xAxis->set-
BasePen(QPen(Qt::white));
        ui.QCPlot1->yAxis2->set-
BasePen(QPen(QColor(Qt::green)));
        ui.QCPlot1->yAxis->set-
BasePen(QPen(QColor(255, 110, 40)));

        ui.QCPlot2->xAxis->set-
BasePen(QPen(Qt::white));
        ui.QCPlot2->yAxis->set-
BasePen(QPen(Qt::white));
        //ui.QCPlot1->yAxis2-
>setLabel(QString("Signal"));
        ui.QCPlot1->xAxis-
>setLabelColor(QColor(255, 255, 255));
        ui.QCPlot1->yAxis-
>setLabelColor(QColor(255, 110, 40));
        ui.QCPlot1->yAxis2-
>setLabelColor(QColor(Qt::green));
        ui.QCPlot2->xAxis-
>setLabelColor(QColor(255, 255, 255));
        ui.QCPlot2->yAxis-
>setLabelColor(QColor(255, 255, 255));
        ui.sigText1-
>setTextColor(QColor(0, 255, 0, 200));
        ui.sigText2-
>setTextColor(QColor(255, 0, 0, 200));
        ui.sigText2->setTextBack-
groundColor(QColor(0, 0, 0, 0));
```

```
        ui.sigText1->setTextBack-
groundColor(QColor(0, 0, 0, 0));

        QPen gridPen =
QPen(QColor(Qt::white));
        gridPen.setStyle(Qt::Dot-
Line);

        ui.QCPlot1->yAxis->grid()-
>setPen(gridPen);
        ui.QCPlot1->xAxis->grid()-
>setPen(gridPen);

        ui.QCPlot2->graph(0)-
>setPen(QPen(QColor(Qt::green), 3));

    }
    ui.QCPlot1->replot();
    ui.QCPlot2->replot();
}
```