

División de Estudios de Posgrado e Investigación

**Dispositivo portátil para la medición de ritmo cardiaco
saturación de oxígeno y gasto calórico**

TESIS

Presentada como requisito parcial para obtener grado de:

Maestría en Ingeniería Electrónica

Ing. Horacio Valencia Córdova

Director:

M.C. José Manuel Chávez

Hermosillo, Sonora.

Noviembre 2017

Agradecimientos

El presente trabajo va dirigido con una expresión de gratitud para mis profesores y amigos de la maestría que fueron de gran apoyo para poder terminar mis estudios.

A mis revisores de tesis por sus comentarios y sugerencias: M.C. Tarín Fontes Jesús y Dr. Pereyda Pierre Carlos.

Especial agradecimiento a mi director de tesis M.C. Chávez José Manuel el cual además de ser un gran maestro y guía durante toda la maestría lo puedo considerar como un amigo. Gracias por su tiempo y dedicación.

A mis compañeros de generación Fernando Joaquín Ramírez Coronel, Leonor García Gámez, Jesús Naím León Ortega, Adrián Francisco Gallardo Tullez, Juan Antonio Borboa Griego y Jesús Alejandro Medellín que nos apoyamos mutuamente en nuestra formación profesional y personal. No se puede pedir por mejores compañeros.

A mis padres, hermano y gran amiga y novia, Argelia Zalazar, que con el apoyo incondicional de ellos he sido capaz de alcanzar nuevas metas tanto profesionales como personales. Muchas gracias.

Al personal del Instituto Tecnológico de Hermosillo y a CONACyT, por proporcionar los medios para realizar mis estudios de maestría.

Contenido

1. Introducción	8
1.1 Antecedentes	8
1.2 Planteamiento del Problema	9
1.3 Hipótesis	10
1.4 Objetivo	10
1.4.1 Objetivo general.....	10
1.4.2 Objetivos específicos.....	10
1.5 Justificación	11
1.6 Metas y alcances	11
1.7 Organización del Documento	11
2. Conceptos	12
2.1 Dispositivos Portables Inteligentes para el Cuidado de la Salud.....	12
2.2 Oxímetro de Pulsos.....	13
2.2.1 Calculo de la saturación de Oxígeno	14
2.2.2 Métodos para obtener los pulsos por minuto	14
2.2.3 Parámetros aplicables al pulso cardiaco	16
2.3 Gasto Calórico.....	17
2.3.1 Método mediante acelerómetros como podómetro	18
2.3.2 Método mediante frecuencia cardiaca	18
2.4 Memoria Externa.....	19
2.5 Margen de error permitido según CENETEC y FDA	20
2.5.1 Comparación con oxímetro de pulsos Measupro OX250.....	20
2.6 Conclusiones	20
3. Desarrollo	21
3.1 Requerimientos necesarios.....	21
3.1.1 Oxímetro de pulsos MAX 30100	21
3.1.2 Memoria EEPROM 25LC512	22
3.1.3 Bluetooth HC-05	22
3.1.4 ATmega328P y MicroVIEW.....	23
3.1.5 Processing.....	24
3.2 Prototipo inalámbrico	24

3.2.1 Programación y configuración MAX30100	25
3.2.2 Pruebas de calibración del MAX30100.....	28
3.2.3 Configuración de la memoria EEPROM	29
3.3 Diseño de filtro digital.....	30
3.3.1 Filtro pasabanda.....	30
3.3.2 Diseño de filtro Transformada Discreta de Fourier.....	34
3.3.3 Diseño de filtro Cruce por Cero	38
3.4 Calculo de la saturación de oxígeno	39
3.5 Calculo del gasto Calórico	42
3.6 Modos de funcionamiento del dispositivo.....	43
3.6.1 Procedimiento de funcionamiento del programa del microcontrolador	45
3.7 Envió de alertas por servidor local	45
3.8 Conclusiones	46
4. Resultados	47
4.1 Metodología general	47
4.2 Resultados Pulsos por minuto (ppm) del método cruce por cero en función de ppm de Measupro.....	47
4.2.1 Metodología.....	47
4.2.2 Pruebas.....	48
4.3 Resultados Pulsos por minuto de la transformada discreta de Fourier en función de ppm de Measupro.....	51
4.3.1 Metodología.....	51
4.3.2 Pruebas.....	51
4.3.1 Comparativa	54
4.4 Saturación de oxígeno capilar periférica SpO ₂	54
4.4.1 Metodología.....	54
4.4.2 Resultados SpO ₂ con respiración normal	55
4.4.3 Resultados SpO ₂ al sostener la respiración.....	58
4.5 Conclusiones	61
5. Conclusiones.....	63
5.1 Conclusiones finales.....	63
5.2 Trabajos futuros y recomendaciones	63

Figuras

Figura 2.1 Función de método cruce por cero.....	15
Figura 3.1 Diagrama MAX30100.....	22
Figura 3.2 Diagrama de conexiones del Módulo HC-05.....	23
Figura 3.3 SparkFun MicroView.....	24
Figura 3.4 Diagrama de dispositivo portátil.....	24
Figura 3.5 Diagrama de conexiones del dispositivo.....	25
Figura 3.6 Seudocódigo de Inicialización mediante I2C MAX30100.....	27
Figura 3.7 LED infrarrojo 24mA y LED rojo 17.4mA (Izquierda). LED infrarrojo 50mA y LED rojo 27mA (Derecha).	28
Figura 3.8 LED infrarrojo 24mA y LED rojo 14.2mA.....	28
Figura 3.9 Memoria EEPROM 24LC256.....	29
Figura 3.10 Comparativa Filtros pasaalta respuesta en la frecuencia.....	31
Figura 3.11 Señal infrarroja con filtro pasaalta. Butterworth (a), Chebyshev (b) y Cauer (c).....	32
Figura 3.12 Comparativa Filtros pasabaja respuesta en la frecuencia.....	33
Figura 3.13 Filtro pasabaja en la señal infrarroja. Butterworth (a), Chebyshev (b) y Cauer (c).....	33
Figura 3.14 Filtro pasabanda resultante.....	34
Figura 3.15 FFT en Matlab.....	35
Figura 3.16 Resultados FFT en Matlab.....	35
Figura 3.17 Diagrama Transformada Discreta de Fourier.....	36
Figura 3.18 Resultados FFT Arduino.....	36
Figura 3.19 Filtro TDF en Processing.....	37
Figura 3.20 Diagrama de flujo de la función cálculo de PPM.....	39
Figura 3.21 Promedio señal IR en Arduino IDE.....	40
Figura 3.22 Diagrama máximos y mínimos en base al promedio de la señal.....	40
Figura 3.23 Gráfica de valores mínimos IR (a) y RED (b).....	41
Figura 3.24 Valores RMS (rojo) aplicando filtro pasabanda.....	41
Figura 3.25 Curva empírica R vs SpO ₂ Measupro.....	42
Figura 3.26 Gasto Calórico en Processing.....	43
Figura 3.27 Terminal serial del programador Arduino IDE.....	44
Figura 3.28 Modo Tiempo Real en Processing.....	44
Figura 3.29 Processing como servidor.....	45
Figura 3.30 Envío de datos por Conexión IP Local.....	46
Figura 3.31 Alerta por conexión IP Local.....	46
Figura 4.1 Gráfica comparativa de valores de Measupro con los medidos.....	48
Figura 4.2 Error PPM del método Cruce por cero sin filtro en función de Ppm Measupro.....	48
Figura 4.3 Gráfica comparativa de valores de Measupro con los medidos aplicando filtro de media 10 muestras.....	49

Figura 4.4 Error PPM del método Cruce por cero con filtro de media 10 muestras en función de Ppm Measupro.....	49
Figura 4.5 Gráfica comparativa de valores de Measupro con los medidos aplicando filtro de media 20 muestras.....	50
Figura 4.6 Error PPM del método Cruce por cero con filtro de media 20 muestras en función de Ppm Measupro.....	50
Figura 4.7 Regresión lineal PPM Measupro y Ppm TDF	52
Figura 4.8 Error PPM del TDF en función de Ppm Measupro.....	52
Figura 4.9 Regresión lineal PPM Measupro y Ppm TDF promedio	53
Figura 4.10 Error PPM del método TDF promedio en función de Ppm Measupro	53
Figura 4.11 Comparativa sin filtro de la SpO ₂ de la Ec. 2.2, 5.5 y Measupro respiración normal.....	55
Figura 4.12 Gráfica de Error SpO ₂ Ec. 2.2 respiración normal.....	55
Figura 4.13 Gráfica de Error SpO ₂ Ec. 3.5 respiración normal.....	56
Figura 4.14 Comparativa con filtro de media de la SpO ₂ de la Ec. 2.2, 5.5 y Measupro respiración normal.....	56
Figura 4.15 Gráfica de Error SpO ₂ Ec. 2.2 con filtro de media y respiración normal	57
Figura 4.16 Gráfica de Error SpO ₂ Ec. 3.5 con filtro de media y respiración normal	57
Figura 4.17 Comparativa sin filtro de la SpO ₂ de la Ec. 2.2, 5.5 y Measupro.....	58
Figura 4.18 Gráfica de Error SpO ₂ Ec. 2.2.....	59
Figura 4.19 Gráfica de Error SpO ₂ Ec. 3.5.....	59
Figura 4.20 Comparativa con filtro de la SpO ₂ de la Ec. 2.2, 5.5 y Measupro	60
Figura 4.21 Gráfica de Error de Ec. 2.2 con filtro de media de ventana móvil de 1000 muestras.....	60
Figura 4.22 Gráfica de Error de Ec. 3.5 con filtro de media de ventana móvil de 1000 muestras.....	61

Tablas

Tabla 1.1 Dispositivos portables comerciales	8
Tabla 2.1 Bioseñales y sus sensores.....	12
Tabla 2.2 Índice de Recuperación del Ritmo Cardíaco.....	17
Tabla 2.3 Parámetros Measupro OX250	20
Tabla 3.1 Parámetros 24LC512.....	22
Tabla 3.2 Cumplimiento de requerimientos Atmega328.....	23
Tabla 3.3 Modo frecuencia cardíaca.....	26
Tabla 3.4 Modos de configuración	26
Tabla 3.5 Corriente LED (mA) Modo Saturación de oxígeno y Ritmo Cardíaco.....	26
Tabla 3.6 Corriente LED (mA) Modo Ritmo cardíaco únicamente.....	27
Tabla 3.7 Comparativa de Orden de filtros pasaalta.....	31
Tabla 3.8 Comparativa de Orden de filtros pasaalta.....	33
Tabla 4.1 Comparativa de aplicación de filtros de media en base a la desviación estándar del error.....	51
Tabla 4.2 Comparativa Error cuadrático medio	54
Tabla 4.3 Comparativa del error de la SpO ₂ al respirar normalmente	58
Tabla 4.4 Comparativa del error de la SpO ₂ al aguantarse la respiración.....	61

Capítulo 1

1. Introducción

1.1 Antecedentes

Desde hace varias décadas los filtros digitales son la base del procesamiento de señales digitales, los cuales son utilizados en un gran número de aplicaciones como comunicaciones, radares, sistemas de control, biomedicina, entre otros [1].

Los avances en las tecnologías de información y comunicaciones junto con los progresos en las micro/nano tecnologías ofrecen nuevas capacidades a las comunicaciones móviles e inalámbricas, procesamiento de señales, interfaces humano-máquina y cuidados de la salud personalizadas [2], lo que ha aumentado el interés por dispositivos portables capaces de medir distintas variables del cuerpo como se muestra en la Tabla 1.1, teniendo todas en común el monitoreo de actividad física mediante acelerómetros u ritmo cardiaco y una aplicación para mostrar las variables.

Tabla 1.1 Dispositivos portables comerciales

Aplicación	Acelerómetro	Ritmo Cardíaco	GPS	Luz Ambiental	Temperatura ambiental	Temperatura Piel	UV	Respuesta Galvánica
Adidas Fit	X	X	X					
Apple iWatch	X	X	X					
Jawbone	X	X	X		X	X		X
Fitbit	X	X						
Microsoft Band	X	X	X	X		X	X	X
Moto 360	X	X	X	X				
Samsung Gear Fit	X	X						

Desde hace más de 10 años Europa ha contribuido sustancialmente en las aplicaciones de Dispositivos Portables Inteligentes para el Cuidado de la Salud (DPICS) promoviendo y apoyando diversos prototipos de sistemas y aplicaciones de DPICS. Dichas aplicaciones están enfocadas a un grupo específico de usuarios y aplicaciones, como el monitoreo de pacientes con enfermedades crónicas, administración de drogas, manejo y monitoreo de enfermedades cardiovasculares y pulmonares, monitoreo en desordenes del sueño y asma, y monitoreo en el desempeño deportivo [3].

En la India se presentó un DPICS el cual consiste en una camisa capaz de monitorear los parámetros de temperatura del cuerpo humano, resistencia galvánica de la piel, pulso cardíaco y el porcentaje de saturación de oxígeno capilar periférica (**SpO₂**) implementado en un microcontrolador ATmega328 y visualizando los resultados en la consola de *Arduino Software IDE* (Arduino), graficando la SpO₂ en la consola de *Processing* [4].

1. 2 Planteamiento del Problema

Los avances tecnológicos han permitido la creación de nuevos dispositivos capaces de medir, procesar y monitorear de manera continua, no invasiva e inalámbrica los signos vitales del cuerpo humano. Sin embargo las tecnologías biomédicas son importadas aumentando considerablemente los precios en el país.

La población global con sobrepeso es alarmante y México no queda exento, debido a lo anterior y como consecuencia la esperanza de vida es la más baja de los países que forman parte de la OCDE. Una manera de ayudar al usuario a mantener un peso adecuado es observar cuantitativamente las calorías consumidas debido a la actividad física.

La implementación de dispositivos capaces de medir el ritmo cardíaco, saturación de oxígeno, gasto calórico además de desplegar los datos inalámbricamente en una computadora manteniendo un error en el cálculo del ritmo cardíaco inferior al 10% o ± 5 pulsos por minuto y un error de la saturación de oxígeno inferior al 2%, según la Federación de Drogas y Alimentos (FDA), es de vital importancia impulsar el desarrollo de tecnologías biomédicas que a su vez impacten positivamente en la salud del usuario al ser capaz de monitorear sus signos vitales y cuantificar su consumo energético a lo largo del día.

1.3 Hipótesis

Es posible realizar un prototipo inalámbrico que monitoree el ritmo cardiaco con un error máximo de ± 5 pulsos por minuto (**ppm**) mediante el método de cruce por cero y transformada discreta de Fourier (**TDF**) así como mostrar la SpO₂ con un error menor al 2%, de manera que se pueda calcular el gasto calórico y mostrar los resultados en tiempo real en la computadora y el prototipo.

1.4 Objetivo

Este trabajo tiene los siguientes objetivos generales y específicos.

1.4.1 Objetivo general

Diseñar un prototipo inalámbrico que monitoree el ritmo cardiaco con un error máximo de ± 5 ppm mediante el método de cruce por cero y TFD, así como la SpO₂, gasto calórico y mostrar los datos en tiempo real en la computadora y dispositivo.

1.4.2 Objetivos específicos

1. Diseñar un filtro digital para acondicionamiento de señal de los sensores ópticos utilizados para la medición de frecuencia cardiaca.
2. Calcular el consumo de calorías mediante los datos obtenidos a partir del sensor oxímetro de pulsos (**OP**) e información personal del usuario.
3. Almacenar datos del ritmo cardiaco en una memoria dentro del dispositivo al no estar conectado a la computadora.
4. Visualizar en tiempo real el ritmo cardiaco, SpO₂ y gasto calórico en un computador.
5. Sistema de alertas en caso de superar límites de ritmo cardiaco.
6. Validar con respecto al OP comercial Measupro OX250, que los métodos de cruce por cero y TDF calculen el ritmo cardiaco con un error menor a ± 5 .
7. Validar con respecto al OP comercial Measupro OX250, que las mediciones de SpO₂ tiene un error menor al 2%.

1.5 Justificación

Entre las razones para abordar el desarrollo de un prototipo inalámbrico para el monitoreo de ritmo cardiaco, SpO₂ y gasto calórico mediante los métodos cruce por cero y TDF son:

- A pesar de que el avance en la tecnología ha permitido la miniaturización, disminución en costos, consumo de energía y peso en sensores y circuitos integrados permitiendo la creación de nuevos dispositivos capaces de medir, procesar y monitorear de manera continua, no invasiva e inalámbrica los signos vitales del cuerpo humano [3], el 90% de las tecnologías biomédicas son importadas a México aumentando considerablemente los precios en el país [5].
- La población global con sobrepeso es alarmante. Hay que mencionar, además que en México de manera semejante el sobrepeso es un problema, ya que el 32% de adolescentes y adultos padecen de sobrepeso, siendo el segundo lugar después de Estados Unidos con mayor obesidad. Más aún la esperanza de vida de México es la más baja de los países que forman parte de la OCDE [6].
- Una manera de ayudar al usuario a mantener un peso adecuado es observar cuantitativamente las calorías consumidas debido a la actividad física [7]. En la tesis [8] se evalúa un DPICS, específicamente el Niketm FuelBand, y se confirma que existe efecto positivo en el aumento de actividad física en el usuario al utilizar DPICS que monitoreen dicha actividad (como pasos y gasto calórico) y actualice objetivos.

1.6 Metas y alcances

El prototipo debe medir el ritmo cardiaco y la SpO₂ con un error menor a ± 5 ppm y $\pm 2\%$ respectivamente en comparación con el OP Measupro OX250 y mostrar la información en tiempo real en la computadora de manera inalámbrica.

1.7 Organización del Documento

En el Capítulo 2 se abordan las temáticas fundamentales referentes al funcionamiento de un OP, ecuaciones para cálculos necesarios para obtener la SpO₂ y gasto calórico y métodos necesarios para obtener la frecuencia cardiaca desde la señal del OP.

En el Capítulo 3 se desarrolla la configuración del OP MAX30100, la programación del microcontrolador ATmega328 y el diseño de los filtros digitales, así como la TDF.

En el Capítulo 4 se validan los resultados del ritmo cardiaco y SpO₂ calculando el error en comparación con Measupro OX250.

En el Capítulo 5 se incluyen las conclusiones, trabajos futuros y recomendaciones.

Capítulo 2

2. Conceptos

En este capítulo se muestran los diferentes conceptos, métodos y ecuaciones matemáticas necesarias para realizar la programación del microcontrolador, filtrar la señal y calcular el ritmo cardíaco y gasto calórico. En la Sección 2.1 se muestran los diferentes tipos de dispositivos portátiles para el monitoreo de variables del cuerpo. En la Sección 2.2 se explica el funcionamiento de un OP y los métodos para el cálculo de la SpO₂ y ritmo cardíaco. En la Sección 2.3 se explica que es y cómo se calcula el gasto calórico. Por último, en la Sección 2.4 se explica el margen de error permitido y las características del OP Measupro OX250.

2.1 Dispositivos Portables Inteligentes para el Cuidado de la Salud

Los *Dispositivos Portables Inteligentes para el Cuidado de la Salud (DPICS)* tienen el propósito de brindar información al usuario o al respectivo supervisor o centro de salud referente a la salud del usuario. Los biosensores de los DPICS miden varios parámetros fisiológicos entre los cuales se encuentran el ritmo cardíaco, la presión arterial, temperatura de la piel y cuerpo, SpO₂, ritmo respiratorio, electrocardiograma, sudoración, glucosa, electromiografía, electroencefalograma, movimientos del cuerpo, entre otros [9] como se muestra en la Tabla 2.1.

Tabla 2.1 Bioseñales y sus sensores.

Bioseñal	Sensor
Electrocardiograma	Electrodos de pecho/piel
Presión arterial	Banda de presión
Ritmo respiratorio	Piezoeléctrico
Saturación de oxígeno	Oxímetro de pulso
Ritmo cardíaco	Oxímetro de pulsos/ electrodos
Sudoración	Respuesta galvánica
Glucosa	Glucómetro
Movimientos del cuerpo	Acelerómetro

Los DPICS cuentan con una variedad de categorías dependiendo de su funcionalidad y características [4], entre esas categorías se encuentran

- 1) Sistemas basados en telas inteligentes: en esta categoría los sistemas incluyen los sensores en la vestimenta. Un ejemplo se propone en [10], donde mediante un cinturón colocado a una altura justo debajo del pecho monitorea frecuencia, amplitud y ppm del corazón, ritmo respiratorio, temperatura corporal, ángulos en tres dimensiones y aceleraciones principalmente en usuarios mayores de edad y los que se encuentran bajo alguna terapia en casa, las cuales constantemente corren el riesgo de cambios abruptos en su salud así como también de accidentes que puedan ocasionar alguna lesión. Dicho dispositivo cuenta con dos subsistemas, el primero adquiere los datos (el cinturón) y los transmite inalámbricamente al siguiente subsistema el cual se encarga de retransmitir los parámetros por internet para su diagnóstico.
- 2) Sistemas basados en redes en áreas del cuerpo (BAN por sus siglas en inglés): para formar un sistema BAN se requiere de pequeños nodos comunicados inalámbricamente, donde cada nodo es responsable de la medición de una o más variables del cuerpo y transmitirla a un nodo central. Además, estos dispositivos pueden ser configurados para medir diferentes señales y parámetros como electrocardiograma, electromiografía, aceleraciones y velocidades angulares 3D y también tener la capacidad de desplegar su información en diferentes plataformas de procesamiento de datos como Labview, Matlab y teléfonos celulares inteligentes [11].
- 3) Plataformas basadas en microcontroladores o plataformas personalizadas: los dispositivos que entran en esta categoría usualmente son los sensores que transmiten los datos a un microcontrolador de manera alámbrica para que procese la información.
- 4) Sistemas portátiles para el monitoreo de la salud basadas en sensores *bluetooth* comerciales y teléfonos celulares.

El dispositivo propuesto será una variante de la categoría número 4, de DPICS basadas en sensores *bluetooth* comerciales y teléfonos celulares, con la diferencia de que en vez de utilizar teléfono celular se conectaría a un computador. Sin embargo, no se descarta trabajo futuro con dichos dispositivos.

2.2 Oxímetro de Pulsos

El OP se ha convertido en el procedimiento estándar para la medición de SpO₂ en los hospitales [12], así como también utilizados entre los DPICS para la medición del

SpO₂ y los ppm [9]. Los OP más comunes utilizan un LED rojo (660nm), un LED infrarrojo (940nm) y un receptor óptico. En el artículo [12] menciona que los OP aprovechan que la hemoglobina reducida (que contiene menos de 4 moléculas de oxígeno) tiene la propiedad de absorber 10 veces más luz que la hemoglobina oxigenada, por lo que el receptor óptico detecta una señal DC que representa la sangre arterial residual, sangre de las venas y tejido sin sangre. Una pequeña porción (alrededor del 1%) de la señal es AC, el cual es obtenido con un filtro pasaalta que representa el pulso cardiaco.

2.2.1 Calculo de la saturación de Oxígeno

La saturación de oxígeno (SaO₂) se refiere al porcentaje de hemoglobina que carga oxígeno en la sangre la cual puede ser calculada mediante la extracción de sangre arterial o la medición de la presión parcial del oxígeno y usando curvas de disociación de oxígeno y hemoglobina. Sin embargo, es posible calcular la SaO₂ mediante métodos no invasivos utilizando el OP aprovechando el hecho de que existe una diferencia en la absorción de la luz en la desoxiemoglobina (Hb) y oxihemoglobina (HbO₂) lo que se conoce como SpO₂ [12] [13].

Primero se calcula la proporción normalizada (R) mediante las mediciones de absorción del LED infrarrojo (940nm) y LED rojo (660nm) utilizando la ecuación (2.1) [12] [13]:

$$R = \frac{R_{LED}}{IR_{LED}} = \frac{\left(\frac{AC_{660}}{DC_{660}}\right)}{\left(\frac{AC_{940}}{DC_{940}}\right)} \quad (2.1)$$

Posteriormente para calibración se obtiene la curva empírica de SpO₂ mediante el uso de R comparando con SaO₂. Por último se obtiene una ecuación lineal (2.2) que se aproxima al comportamiento de la curva empírica [12]:

$$SpO_2 = 110 - (25 * R) \quad (2.2)$$

Con estas ecuaciones no se obtienen lecturas precisas por debajo del 70% de SpO₂.

2.2.2 Métodos para obtener los pulsos por minuto

Para obtener los ppm a partir de la señal AC se utilizan algoritmos como cruce por cero o aplicar la TDF. Previamente se utiliza un filtro digital pasabanda para poder trabajar mejor con la señal y eliminar ruido, verlo en el Capítulo 3 Sección 3.3.3.

2.2.2.1 Cruce por cero

En la Figura 2.1 se muestra el algoritmo del método cruce por cero utilizado para calcular el tiempo entre valles mediante la comparación de la señal a la salida del filtro con una variable llamada “valorMin” que debe ser menor a -100.

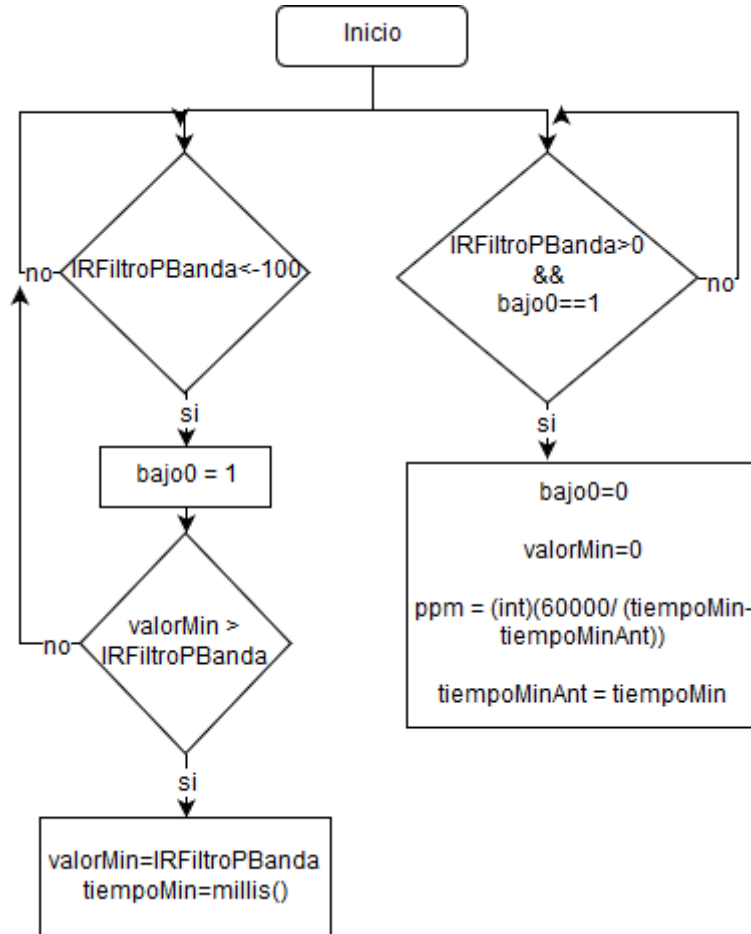


Figura 2.1 Función de método cruce por cero

Posteriormente para obtener los ppm se sustituye el tiempo entre valles en la ecuación (2.3):

$$ppm = (60 * 1000) / (tiempoMin - tiempoMinAnt) \quad (2.3)$$

2.2.2.2 Transformada discreta de Fourier

Para calcular la TDF se utilizó la expresión dada por la ecuación (2.4) que también puede simplificarse como se muestra en la ecuación (2.5) las cuales permite obtener la magnitud de las frecuencias del pulso cardiaco:

$$X_{[k]} = \sum_{n=0}^{N-1} x_{(n)} e^{-j \frac{2\pi n k}{N}} \quad k = 0, 1, 2, \dots, N - 1$$

(2.5)

(2.4)

$$X_{[k]} = \sum_{n=0}^{N-1} x_{(n)} W_N^{nk} \quad k = 0, 1, 2, \dots, N - 1$$

Donde $W_N^{nk} = e^{-j\frac{2\pi}{N}nk}$

Para calcular el número de muestras necesarias a utilizar en la TDF en base a la resolución se obtiene mediante la ecuación (2.6):

$$\mathbf{muestras} = Fm/\text{resolución} \quad (2.6)$$

Siendo Fm la frecuencia de muestreo, el cual son 50 muestras por segundo y la resolución es de 0.0166667Hz o 1ppm. Esto da como resultado 3000 muestras por lo que se realizan las pruebas en la computadora usando el software de *Processing* y no en el microcontrolador debido a que se supera el límite de memoria del mismo.

2.2.3 Parámetros aplicables al pulso cardiaco

Existen ciertos parámetros que se aplican al *Ritmo Cardíaco* del usuario para evaluar si se encuentra en rangos adecuados. La ecuación (2.7) del *Ritmo Cardíaco Máximo (RCM)*, obtenido en [14] mediante estudios en laboratorio, se utiliza para determinar la máxima capacidad aeróbica:

$$RCM = 208 - 0.7 * \text{edad} \quad (2.7)$$

Otro parámetro del ritmo cardiaco es el *Ritmo Cardíaco Objetivo (RCO)* el cual es el rango en el cual la mayoría de los usuarios pueden mantenerse por un mayor periodo de tiempo y no se produce ácido láctico mientras se realizan actividades físicas de tipo aeróbico [15]. El RCO recomendado es entre el 70 y 80% el RCM, para ello se emplea la ecuación (2.8):

$$RCO = 0.75 * RCM \quad (2.8)$$

Además del RCM y el RCO también existe el *Índice de Recuperación del Ritmo Cardíaco (IRrc)* el cual es utilizado para evaluar el estado de la condición física. Para calcularlo se toma el valor (ppm) del ritmo cardíaco a los 180 segundos y 270 segundos posteriores de haber finalizado la prueba física mediante la siguiente ecuación (2.9):

$$IRrc = \frac{ppm(t = 180) - ppm(t = 270)}{10} \quad (2.9)$$

Una vez obtenido el IRrc se compara con la Tabla 2.2 adquirida de [15]:

Tabla 2.2 Índice de Recuperación del Ritmo Cardíaco

Categoría	IRrc
Pobre	< 2
Aceptable	2 - 3
Bueno	3 - 4
Excelente	4 - 6
Excepcional	> 6

2.3 Gasto Calórico

En la actualidad el monitoreo del gasto energético (GE) es indispensable en los sistemas de monitoreo de la salud debido a que es necesario tener un balance entre la energía consumida y los alimentos ingeridos. El GE es la energía que el cuerpo humano libera a través de calor, el movimiento mecánico internos, necesario para mantener el cuerpo con vida, y los externos para mantener el estilo de vida [16]. Para calcular la energía consumida se utiliza el Equivalente Metabólico (**MET**), el cual es definido como la relación que existe entre el ritmo metabólico de una persona mientras realiza actividades comparadas con el ritmo metabólico en reposo. MET se correlaciona con el requerimiento de oxígeno de un cuerpo humano ante diferentes actividades, por lo que la unidad de medición es definida como en la ecuación (2.10) [7] [16]:

$$1MET = \frac{3.5mL}{(min/Kg)} = 1Kcal/h/kg \quad (2.10)$$

2.3.1 Método mediante acelerómetros como podómetro

Un método para obtener el gasto calórico (GC) mediante el monitoreo de las aceleraciones, es mediante el uso de acelerómetros que tengan la funcionalidad de podómetros. El podómetro se basa en el conteo de los pasos. La forma de calcular los MET mediante los pasos varía dependiendo si se encuentra caminando o corriendo respectivamente, como se muestra en la ecuación (2.11) y (2.12) [7]:

Al caminar,

$$MET = 0.0272 \times velocidad(m/min) + 1.2, \quad (2.11)$$

Al correr,

$$MET = 0.093 \times velocidad(m/min) - 4.7. \quad (2.12)$$

Para calcular la velocidad es en base al número de pasos (2.13) y el tamaño de la zancada (2.14):

$$Velocidad = \frac{tz(m)}{t(min)} \times pasos \quad (2.13)$$

$$tz = altura(m) \times (0.4236 \text{ en hombres o } 0.4356 \text{ en mujeres}) \quad (2.14)$$

Siendo tz el tamaño de zancada.

Otras posibles aplicaciones para los acelerómetros es la habilidad de reconocer movimientos o posiciones como por ejemplo caídas, estar sentado o parado [17].

2.3.2 Método mediante frecuencia cardiaca

Como se menciona en [18] y [19] la estimación de gasto energético mediante el uso de la frecuencia cardiaca es mejor que los métodos de podómetros y acelerómetros por si solos en diversos tipos de actividades cotidianas. Sin embargo, también indican que este tipo de método sobre estima el gasto energético un 9.6% en hombres y 11.1% en mujeres. A continuación, se muestra en las ecuaciones (2.16) y (2.17) el algoritmo utilizado en [19] para el cálculo del gasto energético por actividad (**GEA**):

$$GEA = \frac{kcal}{Kg \cdot min} \quad (2.16)$$

$$\begin{aligned}
 & \hspace{15em} (2.17) \\
 GEA = & \left\{ \begin{array}{ll} \frac{(5.95 * FCsR) + (0.23 * Edad) + (84 * gen) - 134}{23} \times FCsR, & FCsR < 23 \\ \frac{(5.95 * FCsR) + (0.23 * Edad) + (84 * gen) - 134}{4186.8}, & FCsR \geq 23 \end{array} \right.
 \end{aligned}$$

Para calcular GEA se requiere de la frecuencia cardiaca sobre el reposo (**FCsR**) el cual es calculado como la frecuencia cardiaca (**Fc**) menos la frecuencia cardiaca al dormir (**FcD**) como se muestra en las ecuaciones (2.18) y (2.19):

$$FCsR = Fc - FcD \quad (2.18)$$

$$FcD = 0.83 \times FcA \quad (2.19)$$

La ecuación (2.19) requiere de la frecuencia cardiaca al estar acostado (**FcA**). Sin embargo, para simplificar, en [19] el promedio **FcA** de 48 participantes con una edad de 35 ± 11.4 años es de 66, por lo que **FcD** sería una constante de 54.78 como lo indica la ecuación (2.20):

$$FCsR = Fc - 54.78 \quad (2.20)$$

2.4 Memoria Externa

El dispositivo requiere de un sistema de almacenamiento que permita guardar datos del ritmo cardiaco mientras no exista conexión con un computador que le permita descargar y visualizar dichos datos. La memoria no debe de perderse en caso de falta de energía en las baterías.

Para esto es necesario diferenciar los principales tipos de memorias que se utilizan en computadoras:

- RAM (Memoria de acceso aleatorio) puede almacenar y recuperar datos. La memoria RAM es volátil, lo que significa que al remover la conexión eléctrica sus datos son perdidos.
- ROM (Memoria de solo lectura) sólo puede recuperar datos almacenados permanentemente.
- EEPROM el cual es una memoria eléctricamente borrable y programable, por lo que es posible escribir y recuperar datos y mantenerlos incluso al perder conexión eléctrica.

El tipo de memoria EEPROM es la que se utilizará en este proyecto.

2.5 Margen de error permitido según CENETEC y FDA

En el documento de la guía tecnológica N°38 Oxímetro GMDN45607 (Nomenclatura Global de Dispositivos Médicos por sus siglas en inglés) y la FDA, proponen que para que la SpO₂ y los ppm sean adecuados, su desviación estándar del error no debe ser superior al 3% y al 10% (± 5 ppm) respectivamente.

2.5.1 Comparación con oxímetro de pulsos Measupro OX250

Este OP no está diseñado para su uso durante actividad física y sólo es aplicable en pacientes de 15 a 60 años. Los parámetros se muestran en la Tabla 2.3:

Tabla 2.3 Parámetros Measupro OX250

Rango	SpO ₂	35% a 100%
	Frecuencia cardiaca	25ppm a 250ppm
Resolución	SpO ₂	1%
	Frecuencia cardiaca	1ppm
Precisión	SpO ₂	$\pm 2\%$ (70% a 100%)
	Frecuencia cardiaca	± 2 ppm

2.6 Conclusiones

En este capítulo se presentan los métodos y ecuaciones para el cálculo de la SpO₂, ritmo cardiaco y gasto calórico. Para el gasto calórico se analizan 3 métodos entre los cuales se seleccionó el método del ritmo cardiaco por las investigaciones presentadas, que demuestran que el uso de podómetros y acelerómetros por si solos tienen un mayor margen de error que el método del ritmo cardiaco para el cálculo de GE. Por último, se presenta el OP Measupro OX250 con el cual se utilizara para validar el prototipo para la medición de ritmo cardiaco y SpO₂.

Capítulo 3

3. Desarrollo

En este capítulo se presenta el desarrollo de la programación y configuración del OP MAX30100, el diseño del filtro digital para eliminación de ruido, programación de la TDF, selección y programación de la memoria y modos de funcionamiento del prototipo. En la Sección 3.1 se muestra la funcionalidad general del dispositivo portátil, configuración y programación del OP MAX30100 y la memoria EEPROM 24LC512. En la Sección 3.2 se explica el diseño de los filtros digitales los cuales son el filtro pasabanda y la TDF. En la Sección 3.3 se explica la programación y cómo se calcula el gasto calórico. En la Sección 3.4 se exponen los modos de operación del dispositivo. Por último, en la Sección 3.5 se explica el sistema de alertas por servidor local.

3.1 Requerimientos necesarios

En esta sección se presentan los requerimientos que debe cumplir el dispositivo propuesto. En los capítulos anteriores se mostró que:

- Para medir el gasto calórico se necesita el ritmo cardiaco, peso, edad y género del que utiliza el dispositivo.
- Utilizar el OP como método no invasivo para la medición de SpO₂ y el ritmo cardiaco.
- Se requiere una memoria EEPROM externa para guardar información mientras el dispositivo no esté comunicado con la computadora.

3.1.1 Oxímetro de pulsos MAX 30100

El OP MAX30100 es utilizado en aplicaciones de monitoreo médico, como asistente de entrenamiento físico y en dispositivos portátiles tipo “*wearables*” principalmente. Algunos de los beneficios son las dimensiones reducidas, bajo consumo energético, filtrado automático de luz exterior, velocidad de muestreo y la corriente de los LED rojo e infrarrojo programable, lo que permite ahorro de energía y facilita las pruebas.

En la Figura 3.1 se observa un diagrama de funcionamiento interno del MAX30100 en el cual se aprecia los pines de alimentación del LED rojo y LED infrarrojo, SCL, SDA e INT para el protocolo de comunicación I²C.

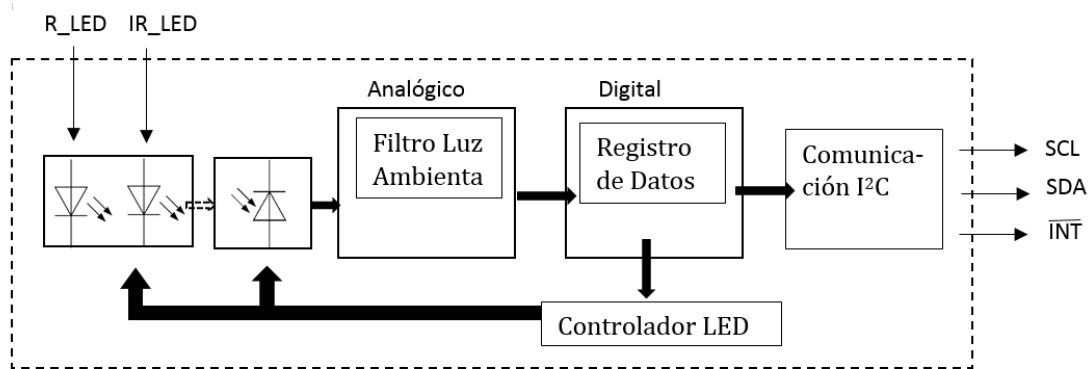


Figura 3.1 Diagrama MAX30100

3.1.2 Memoria EEPROM 25LC512

La memoria EEPROM 24LC512 es una memoria de 512Kbit o 64kilobyte lo que permitiría guardar 64,000 variables enteras sin signo tipo `uint8_t` (carácter sin signo) de 1byte. Las variables tipo `uint8_t` varían entre 0 y 255 lo que es adecuado para guardar el ritmo cardiaco que varía entre aproximadamente 50 y 220 ppm. Si se guarda el pulso promedio cada 30 segundos es posible guardar durante poco más de 22 días seguidos.

Tabla 3.1 Parámetros 24LC512

Memoria EEPROM 24LC512	
Rango Vcc	1.7-5.5 V
Protocolo comunicación	I ² C
Ciclos de Escritura/Borrado	>1,000,000
Memoria	64kB

Como se observa en la Tabla 3.1 el protocolo de comunicación es I²C al igual que en el OP MAX30100 lo que facilita su programación. También se observa en la Tabla 3.1 que los ciclos de escritura y borrado son mayores a un millón de veces lo que permite guardar el pulso cardiaco cada 30 segundos durante aproximadamente 347 días.

3.1.3 Bluetooth HC-05

No es motivo de este trabajo de investigación presentar dispositivos *bluetooth* o comparativas para elegir los más capaces, ahorradores o de mayor alcance, la finalidad es cumplir con la necesidad de comunicación inalámbrica del dispositivo.

El módulo *bluetooth* HC-05 con adaptador fue seleccionado debido al bajo costo y porque es un *bluetooth* SPP (protocolo de puerto serial) que permite el servicio de trabajar con el *bluetooth* remoto como un puerto serie virtual en la computadora. Esto facilita la comunicación inalámbrica por *bluetooth* a la computadora.

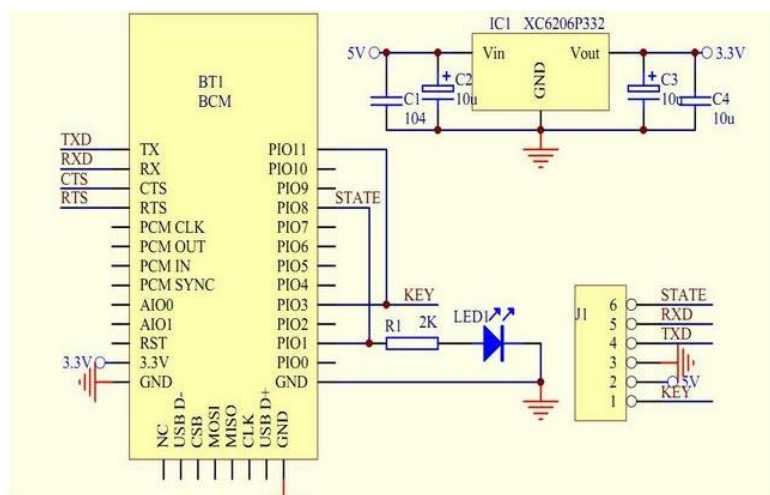


Figura 3.2 Diagrama de conexiones del Módulo HC-05

3.1.4 ATmega328P y MicroVIEW

Se seleccionó el microcontrolador ATmega328P debido a la facilidad de su uso, la gran cantidad de información existente y porque sus especificaciones cumplen con los requerimientos del dispositivo (Tabla 3.2). El microcontrolador Atmega328 se conoce por sus plataformas Arduino las cuales se destacan por ser de fácil aprendizaje y aplicación debido a la documentación y gran número de librerías disponibles para aplicaciones específicas. El software con el que se programa es Arduino Software IDE el cual es de código abierto.

Tabla 3.2 Cumplimiento de requerimientos Atmega328

Requerimientos	Cumple
Comunicación I ² C	Si
Comunicación Serial	Si
Memoria SRAM	Suficiente (2KBytes)

Entre las plataformas que utilizan Arduino se encuentra MicroView (Figura 3.3) el cual es un módulo que utiliza el microcontrolador ATmega328P junto con una pantalla OLED de 64x48 pixeles. Además, existe documentación y librerías para su programación en el software Arduino Software IDE lo que facilita su aplicación.

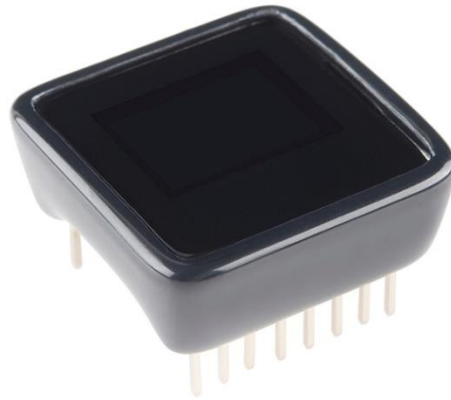


Figura 3.3 SparkFun MicroView

3.1.5 Processing

Processing es un software gratuito y de código abierto que tiene gran cantidad de documentación y librerías sobre manejo de datos, geometría, visión, visualización artística entre otros que facilita el trabajar en conjunto con el microcontrolador ATmega328P para desplegar los datos en diferentes tipos de gráficas y aplicar operaciones matemáticas como la TDF.

3.2 Prototipo inalámbrico

El dispositivo portátil está conformado por un OP MAX30100, el microcontrolador ATmega328, la memoria EEPROM 24LC512, una pantalla OLED de 64x48 pixeles y el *bluetooth* HC-05 como se muestra en el diagrama de la Figura 3.4.

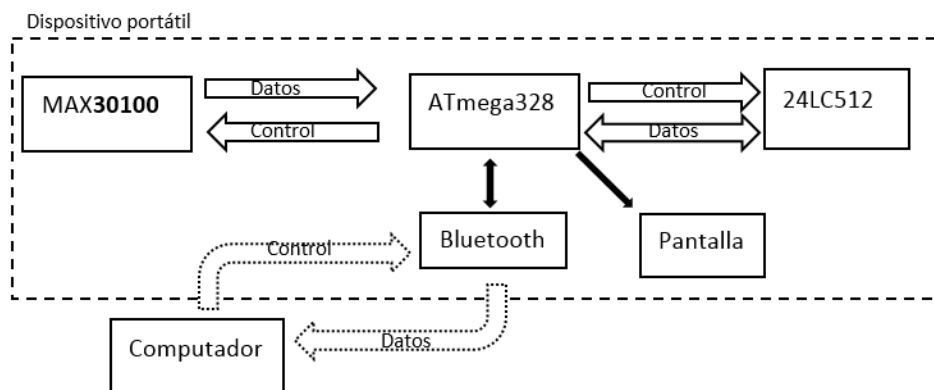


Figura 3.4 Diagrama de dispositivo portátil

El microcontrolador ATmega328 se encarga de configurar y obtener datos del OP MAX30100 para calcular la SpO₂ y obtener el pulso mediante el método de cruce por cero. Posteriormente muestra información en la pantalla OLED dependiendo del modo de operación en el que se encuentra, guarda el pulso cada 30 segundos en la memoria 24LC512 cuando se encuentra en modo portátil y por último enviar los datos por *bluetooth* para mostrar y procesar los datos en tiempo real desde una computadora mediante el software de Processing.

En el diagrama de la Figura 3.5 se pueden observar las conexiones realizadas para el dispositivo. Además de incluir el módulo MicroView, la memoria EEPROM 24LC512, el *bluetooth* HC-05 y el OP MAX30100 se agrega un cargador para baterías de 3.3V LiPO con elevador de corriente directa a corriente directa de 5V. Esto debido a que varios dispositivos requieren de una alimentación de 5V.

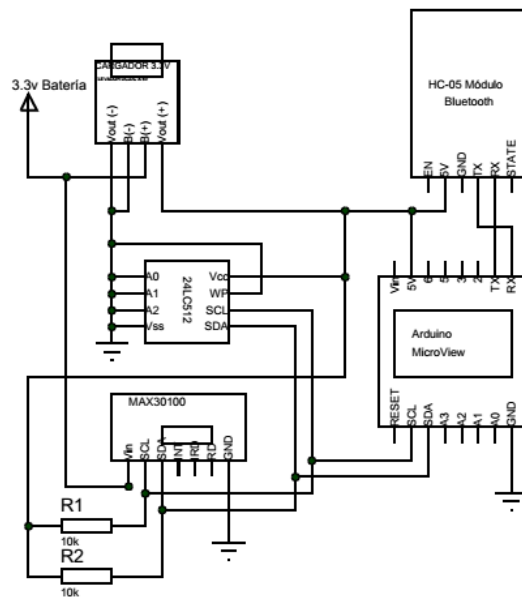


Figura 3.5 Diagrama de conexiones del dispositivo

3.2.1 Programación y configuración MAX30100

El dispositivo cuenta con un convertidor analógico-digital (ADC) de 16-bit y su tiempo de muestreo puede ser configurado entre las 50 muestras por segundo (mps) a 1000 mps. El máximo tiempo de muestreo del ADC depende del ancho de pulso seleccionado, como se muestra en la Tabla 3.3.

Tabla 3.3 Modo frecuencia cardiaca

Muestras por segundo	Ancho de pulso (us)			
	200	400	600	1600
50	X	X	X	X
100	X	X	X	X
167	X	X	X	
200	X	X	X	
400	X	X		
600	X	X		
800	X	X		
1000	X	X		
Resolución (bits)	13	14	15	16

El OP MAX30100 puede ser configurado para seleccionar su modo de operación como se indica en la Tabla 3.4. Entre sus modos de operación se encuentra el modo “Ritmo cardiaco únicamente” en el cual se mide el ritmo cardiaco mediante el uso del LED infrarrojo y el modo “Saturación de Oxígeno y Ritmo Cardiaco” medir tanto la SpO₂ como el ritmo cardiaco activando ambos LEDs, LED infrarrojo y LED rojo.

Tabla 3.4 Modos de configuración

Bits	MODO
010	Ritmo cardiaco únicamente
011	Saturación de Oxígeno y Ritmo Cardiaco

Tabla 3.5 Corriente LED (mA) Modo Saturación de oxígeno y Ritmo Cardiaco

Muestras por segundo	Ancho de pulso (us)			
	200	400	800	1600
50	0.66	1.33	2.62	5.172
100	1.26	2.51	4.96	9.76
167	2.07	4.14	8.173	
200	2.49	4.93	9.68	
400	4.89	9.76		
600	7.31			
800	9.75			
1000	12.17			

Tabla 3.6 Corriente LED (mA) Modo Ritmo cardiaco únicamente

Muestras por segundo	Ancho de pulso (us)			
	200	400	800	1600
50	0.25	0.51	1.02	2.04
100	0.51	1.02	2.04	4.07
167	0.85	1.70	3.40	6.79
200	1.02	2.04	4.07	8.13
400	2.04	4.07	8.12	
600	3.05	6.08		
800	4.07			
1000	5.07			

En la Figura 3.6 se observa el pseudocódigo para inicializar el MAX30100 con las configuraciones de los modos de operación como se muestran en la Tabla 3.4, muestras por segundo y resolución en bits como se muestra en Tabla 3.3 y uso de corriente en los LED rojo e infrarrojo como se indica en la Tabla 3.5 y Tabla 3.6. Mediante las configuraciones mostradas en Figura 3.6 es posible comenzar la operación del OP MAX30100 mediante el uso de ATmega328.

```

Función Inicializar
Inicio
    Dirección 0x06;
    Datos 0x0B;
    I2C_Escribir(Dirección,Datos);

    Dirección 0x07;
    Datos 0x43;
    I2C_Escribir(Dirección,Datos);

    Dirección 0x09;
    Datos 0x37;
    I2C_Escribir(Dirección,Datos);

    Dirección 0x01;
    Datos 0x70;
    I2C_Escribir(Dirección,Datos);
Fin
    
```

Figura 3.6 Seudocódigo de Inicialización mediante I2C MAX30100

Al activar la modalidad del sensor MAX30100 de “Ritmo-Cardiaco Únicamente” el LED rojo se encuentra inactivo y solo el led infrarrojo es utilizado para la captura de datos ópticos los cuales son utilizados para dicho cálculo. Este modo permite el ahorro de energía debido a que únicamente se utiliza el LED infrarrojo.

3.2.2 Pruebas de calibración del MAX30100

Para obtener una medición adecuada en la SpO₂ mediante la ecuación (2.1), la calibración del OP MAX30100 requiere que las mediciones de LED rojo y LED infrarrojo obtengan mediciones de corriente directa lo más similar posible. En la Tabla 3.3, Tabla 3.5 y Tabla 3.6 se observan las posibles configuraciones del ancho de pulso en la frecuencia del parpadeo del LED infrarrojo y rojo así como su consumo de corriente para la detección del ritmo cardiaco tomando en cuenta las muestras por segundo y la resolución en bits.

En la Figura 3.7 se presentan dos ejemplos donde las señales de corriente directa no se encuentran suficientemente cerca. La señal en rojo representa el LED rojo y la señal en azul representa el LED infrarrojo.

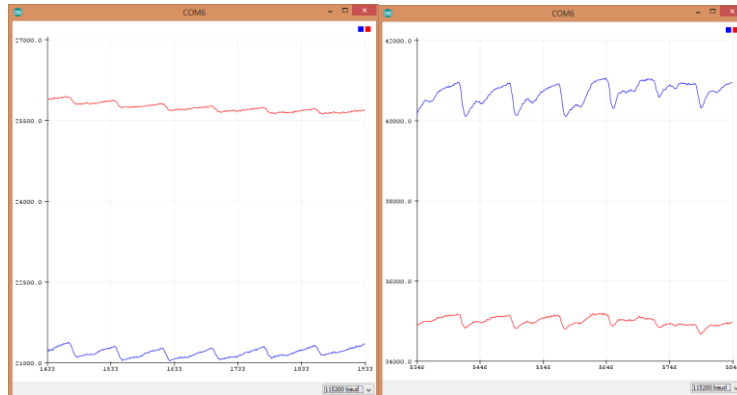


Figura 3.7 LED infrarrojo 24mA y LED rojo 17.4mA (Izquierda). LED infrarrojo 50mA y LED rojo 27mA (Derecha).

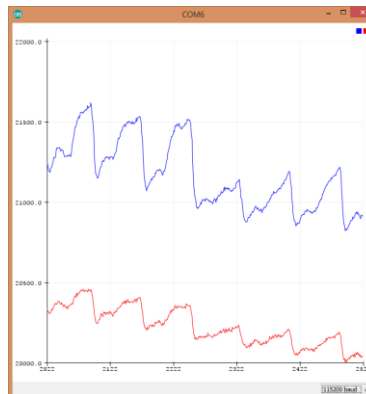


Figura 3.8 LED infrarrojo 24mA y LED rojo 14.2mA

La configuración de corrientes que mostró el mejor resultado es, LED infrarrojo a 24.0 mA y LED rojo a 14.2mA como se muestra en la Figura 3.8.

En base a esta configuración se realizó el trabajo para obtención la SpO₂ y el ritmo cardiaco.

3.2.3 Configuración de la memoria EEPROM

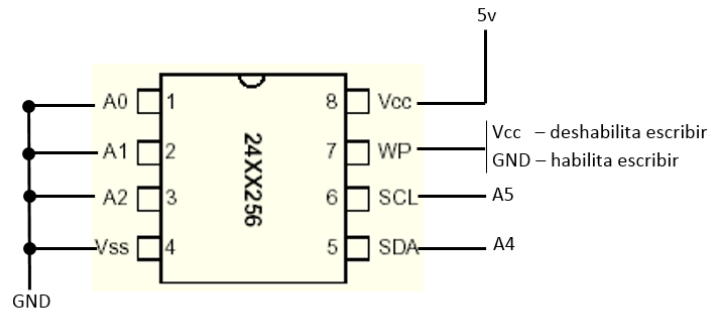


Figura 3.9 Memoria EEPROM 24LC256

La memoria 24LC512, Figura 3.9, tiene un voltaje de operación entre 2.5 a 5.5V lo que nos permite alimentarlo con 3.3V o 5V. Los pines A0, A1, A2 se utilizan para configurar la dirección I²C de 0x50 en adelante en este caso para utilizar la dirección 0x50 se conectan todos a tierra. Para la comunicación I²C se utilizan los pines SCL y SDA los cuales se interconectan al microcontrolador ATmega328 en los pines A5 y A4 respectivamente. El pin número 7 del 24LC512 sirve para habilitar o deshabilitar escritura conectándolo a tierra o Vcc respectivamente. En este caso se mantiene conectado a tierra para permitir la escritura en la memoria. La memoria EEPROM 24LC512 tiene 64kbytes por lo que cuenta con 65536 posiciones donde guardar los datos de un byte. El tiempo de escritura en este dispositivo demora máximo 5 milisegundos por lo que se requiere agregar ese tiempo de espera antes de volver a escribir o leer.

Para guardar datos en la memoria EEPROM se utilizó el siguiente pseudocódigo:

Función Void EscribirEEPROM (Dirección, eeAddress, Datos)

InicioEscrituraI²C (Dirección)//0x50 en adelante

EnviarI²C (eeAddress>>8)//eeAddress --> 0 a 65535

EnviarI²C (eeAddress & 0xFF)

EnviarI²C (Datos)//byte a guardar

Cerrar²C

Esperar 5ms

Cerrar Función;

Para leer datos en la memoria EEPROM se utilizó el siguiente pseudocódigo:

Función byte LeerEEPROM (Dirección, eeAddress, Datos)

InicioEscritura²C (Dirección)

Enviar²C (eeAddress>>8)

Enviar²C (eeAddress & 0xFF)

Cerrar²C

Leer²C (Dirección, 1byte)

Lectura = I²C_Recibido ()

Regresar (Lectura)

Cerrar Función;

3.3 Diseño de filtro digital

3.3.1 Filtro pasabanda

En la investigación [10] y [14] se menciona que existen mediciones entre los 40ppm y 220ppm por lo que se decidió iniciar con un filtro de Respuesta Infinita al Impulso (IIR) pasabanda cuyas frecuencias de corte sean 0.6667Hz a 3.666Hz respectivamente.

Para el diseño del filtro pasabanda se utilizó Matlab con las siguientes funciones:

- *fdesign()*, permite el diseño del filtro en base a especificaciones de frecuencia de borde pasabanda, parabanda, rizo pasabanda, orden del filtro, entre otros.
- *designmethods()*, muestra los métodos (butterworth, chebychev tipo1, chebychev tipo2 y elíptico) disponibles en base a las especificaciones descritas.
- *design()*, diseña el filtro en base a un método seleccionado y especificaciones descritas.
- *fvtool()*, interfaz gráfica que permite analizar filtros digitales.
- *sos2tf()*, pasa del formato tipo df2sos a ecuación de diferencias.

Primero se diseñó el filtro IIR pasabanda con un rizo pasabanda de 1dB, una frecuencia de borde de pasabanda de 0.6667Hz, una frecuencia de borde parabanda de 0.1Hz y el rizo parabanda de 60dB. Mediante estas especificaciones se realizó un filtro

Butterworth, Chebyshev y Cauer como se muestra en la Figura 3.10 y el orden de los filtros obtenidos se muestran en la Tabla 3.7.

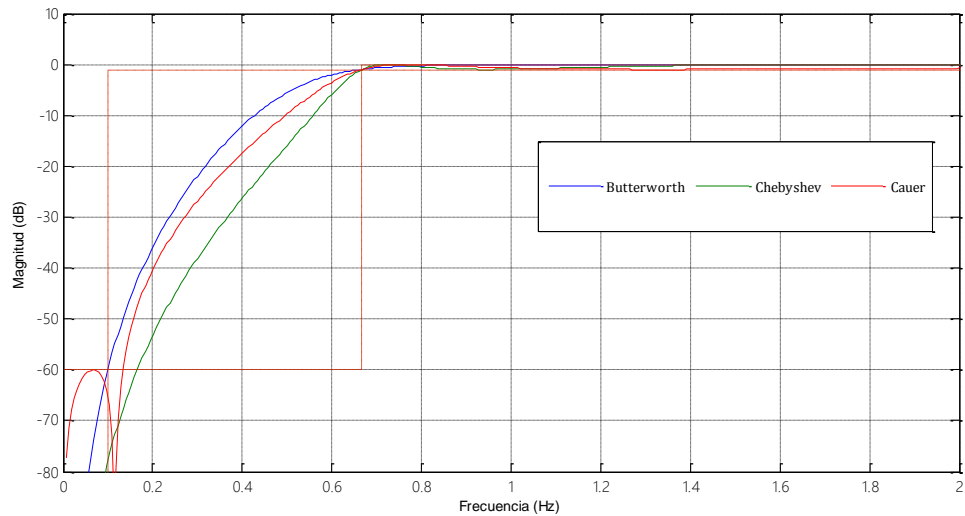
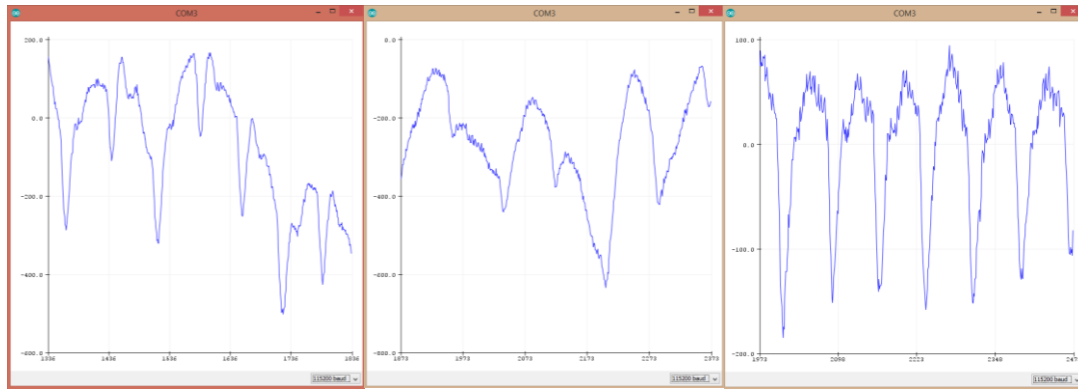


Figura 3.10 Comparativa Filtros pasaalta respuesta en la frecuencia

Tabla 3.7 Comparativa de Orden de filtros pasaalta

Filtro	Orden
Butterworth	4
Chebyshev	4
Cauer	3

Al realizar la programación del código digital en Arduino Software IDE se obtienen los resultados mostrados en la Figura 3.11.



(a)

(b)

(c)

Figura 3.11 Señal infrarroja con filtro pasaalta. Butterworth (a), Chebyshev (b) y Cauer (c).

Se puede observar que tanto la señal de Butterworth como Chebyshev muestran un comportamiento no deseado Figura 3.11 (a y b).

Se seleccionó el filtro pasaalta de Cauer de 3er orden debido a que logra eliminar la señal de corriente directa y permite analizar las frecuencias donde se presenta la señal del pulso, Figura 3.11 (c).

La ecuación en diferencias del filtro Cauer 3er orden obtenida a partir de Matlab en base al diseño de los filtros especificados previamente se muestra en (3.1):

$$\frac{0.93U[n] - 2.81U[n - 1] + 2.81U[n - 2] - 0.93U[n - 3]}{Y[n] - 2.87Y[n - 1] + 2.75Y[n - 2] - 0.88Y[n - 3]} \quad (3.1)$$

Posteriormente para eliminar el ruido de alta frecuencia se diseñó el filtro IIR pasabaja de Butterworth, Chebyshev y Cauer con un rizo pasabanda de 1dB, una frecuencia de borde de pasabanda de 5Hz, una frecuencia de borde parabanda de 45Hz y un rizo parabanda de 100dB como se muestra en la Figura 3.12.

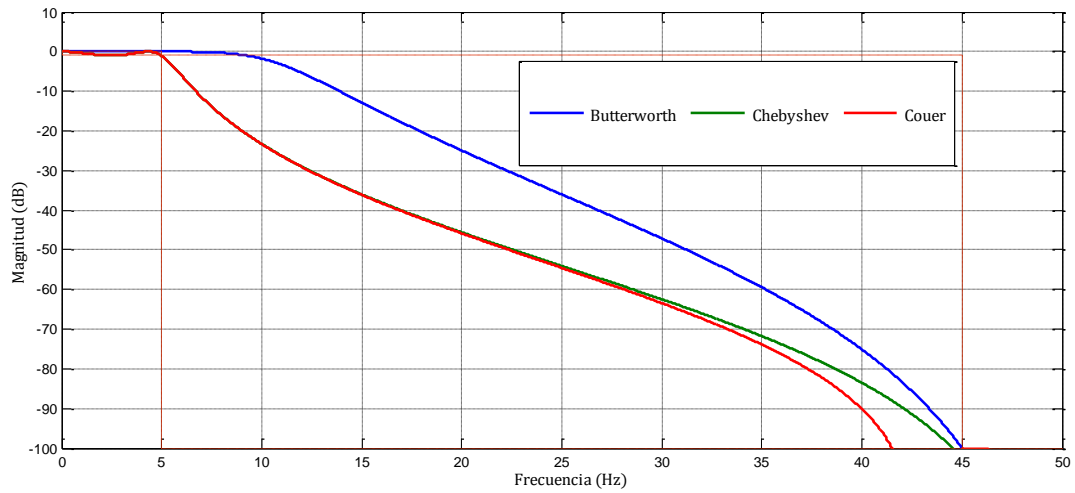


Figura 3.12 Comparativa Filtros pasabaja respuesta en la frecuencia

El orden de los filtros resultantes se observa en la Tabla 3.8.

Tabla 3.8 Comparativa de Orden de filtros pasaalta

Filtro	Orden
Butterworth	4
Chebyshev	3
Cauer	3

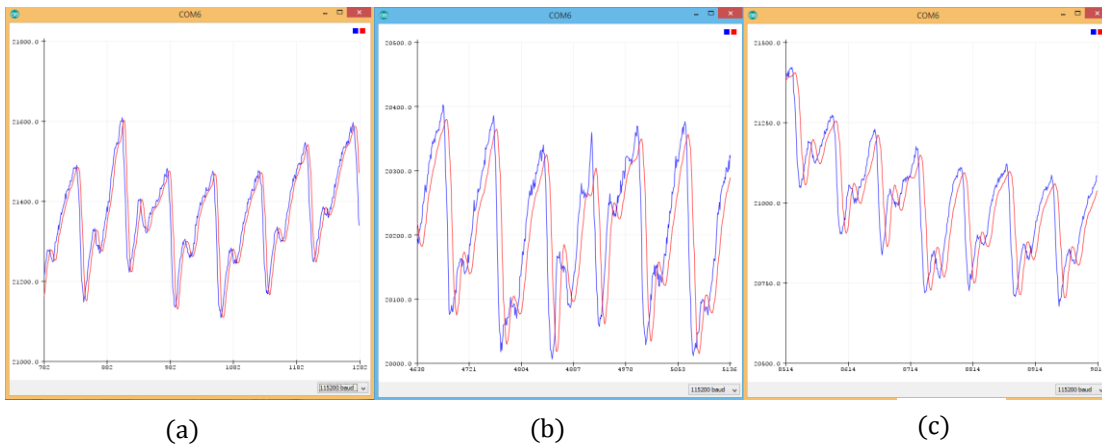


Figura 3.13 Filtro pasabaja en la señal infrarroja. Butterworth (a), Chebyshev (b) y Cauer (c).

El filtro pasabaja seleccionado fue Butterworth debido a que se observan menores atenuaciones en la señal como se observa en la Figura 3.13 (a).

La ecuación de diferencias del filtro pasabaja Butterworth obtenida a partir de Matlab en base al diseño de los filtros especificados previamente se muestra en (3.2):

$$\frac{0.0063U[n] + 0.0255U[n - 1] + 0.0382U[n - 2] + 0.0255U[n - 3] + 0.0063U[n - 4]}{Y[n] - 2.23Y[n - 1] + 2.10Y[n - 2] - 0.92Y[n - 3] + 0.16Y[n - 4]} \quad (3.2)$$

Al utilizar el filtro pasaalta de Cauer 3er orden y filtro pasabaja de Butterworth 4to orden en la señal infrarroja se obtiene lo que se muestra en la Figura 3.14. La señal filtrada facilita analizar la señal de corriente alterna para calcular la SpO₂, ver Sección 3.4, y obtener el pulso cardiaco mediante los métodos de la TDF Sección 3.3.2 y cruce por cero Sección 3.3.3.

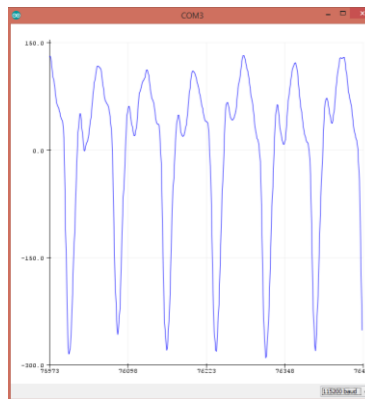


Figura 3.14 Filtro pasabanda resultante

3.3.2 Diseño de filtro Transformada Discreta de Fourier

Se implementó una simulación en Matlab con una señal de entrada sinusoidal de 20hz y se guardaron 11 muestras a una velocidad de 400muestras por segundo para demostrar el comportamiento de la implementación de los filtros, en la Figura 3.15 se muestran los resultados obtenidos.

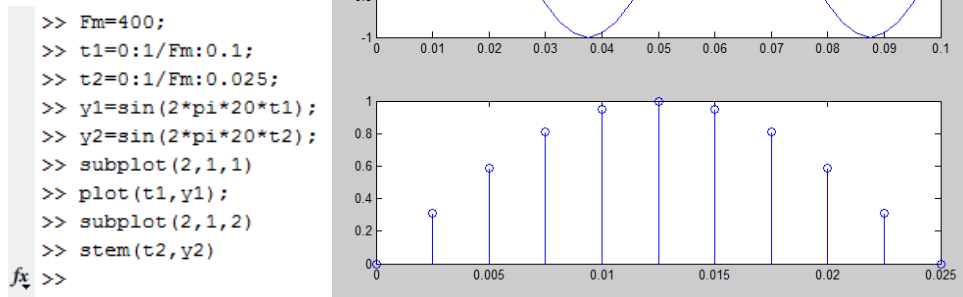


Figura 3.15 FFT en Matlab

Las lecturas obtenidas son las siguientes:

$$Y = \{0, 0.3090, 0.5878, 0.8090, 0.9511, 1, 0.9511, 0.8090, 0.5878, 0.3090, 0\}$$

Mediante las lecturas obtenidas se realizó una comparativa entre los cálculos del filtro FFT en MATLAB y el microcontrolador ATmega-328 para verificar que la implementación fuera correcta.

Para obtener el módulo de los valores entregados por la función de la FFT en MATLAB se obtiene los valores observados en la Figura 3.16:

```

>> Yfft=abs(fft(Y))

Yfft =

Columns 1 through 9
    6.3138    2.7003    0.4853    0.1851    0.0799    0.0230    0.0230    0.0799    0.1851

Columns 10 through 11
    0.4853    2.7003

```

Figura 3.16 Resultados FFT en Matlab

En el caso del microcontrolador se realizó la función TDF explicada en el diagrama de flujo de la Figura 3.17:

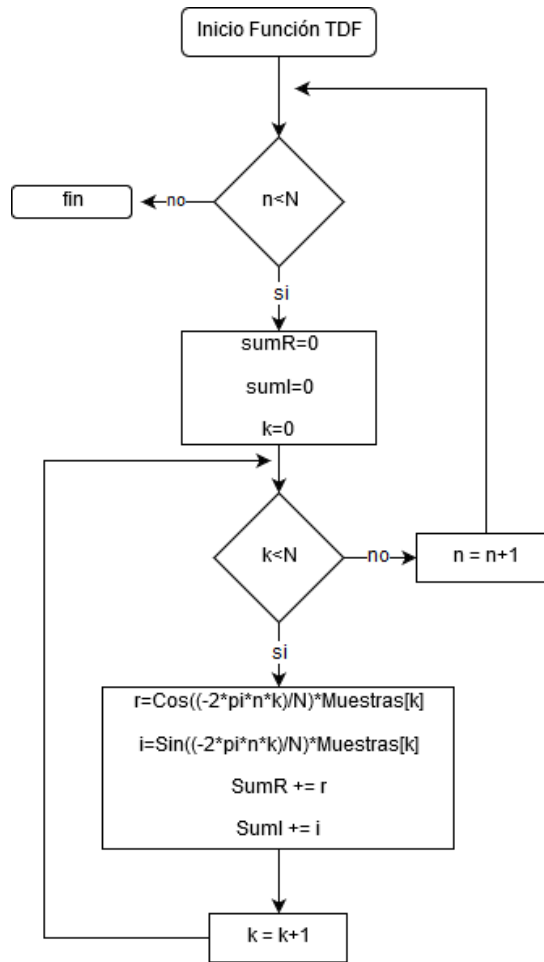


Figura 3.17 Diagrama Transformada Discreta de Fourier

Los resultados de la función se envían por el puerto serial del microcontrolador ATmega328 a la computadora. El software utilizado para mostrar los resultados se llama *Arduino Software IDE*, programa de código abierto que permite programar fácilmente las tarjetas de *Arduino* y microcontroladores Atmel, como se muestra en la Figura 3.18:

```

salidaFFT[0]: 6.3138
salidaFFT[1]: 2.7004
salidaFFT[2]: 0.4853
salidaFFT[3]: 0.1851
salidaFFT[4]: 0.0799
salidaFFT[5]: 0.0229
salidaFFT[6]: 0.0229
salidaFFT[7]: 0.0799
salidaFFT[8]: 0.1851
salidaFFT[9]: 0.4853
salidaFFT[10]: 2.7003
  
```

Figura 3.18 Resultados FFT *Arduino*

Al comparar los datos obtenidos de Matlab con los arrojados por la función del microcontrolador podemos decir que los resultados son adecuados pues estos presentan únicamente una discrepancia de 1 diezmilésima.

3.3.2.1 Resolución de frecuencias TDF

Para conocer la resolución de frecuencias de la función TDF se dividen las muestras por segundo (f_m) entre el tamaño del arreglo de muestras a analizar, indicado por la ecuación (3.3).

$$\text{Resolución} = f_m / \text{muestras} \quad (3.3)$$

Al configurar el OP MAX30100 a 50 muestras por segundo y a la resolución deseada es de 1ppm o 0.016667Hz esto da como resultado que el tamaño del arreglo a utilizar debe ser de 3000 muestras. Esto hace que no sea posible realizar el filtro de la TDF en el microcontrolador ATmega328 debido a que la resolución exige un arreglo de 3000 muestras sobrepasando el límite de memoria del microcontrolador de 2KBytes [20], por lo que se optó por realizar el filtrado desde el programa de la computadora utilizando el software *Processing*.

Además, se optó por utilizar una ventana que muestre el índice de 30 a 230 ppm de la TDF así como realizar el cálculo cada vez que se obtengan las 75 muestras como se muestra en la Figura 3.19. Esto permite aumentar considerablemente la velocidad de procesamiento pues tardaría un minuto en obtener las 3000 muestras.

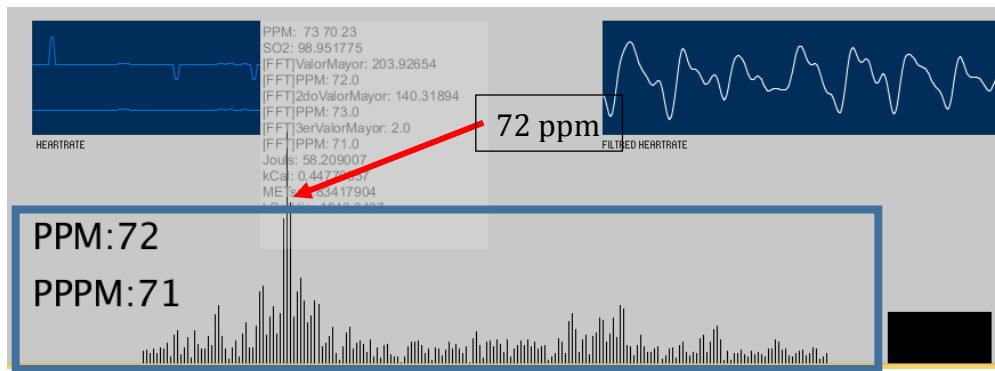


Figura 3.19 Filtro TDF en *Processing*

3.3.3 Diseño de filtro Cruce por Cero

Para el cálculo de los ppm se utiliza la señal del LED IR filtrado mediante el filtro pasabanda (IRfiltrada) visto en la Sección 3.3.1. Posteriormente a IRfiltrada se toma el tiempo entre valles y se calcula usando la ecuación (3.4):

$$ppm = \frac{60000}{tiempoMin - t_{anterior}} \quad (3.4)$$

Siendo tiempoMin el tiempo en milisegundos del valle actual y t_anterior el tiempo en milisegundos del valle anterior. La manera en que el valle es detectado se basa en detectar el valor mínimo (valorMin) de cada ciclo y guardar el tiempo al encontrar el valor mínimo como se observa en el diagrama de flujo de la Figura 3.20.

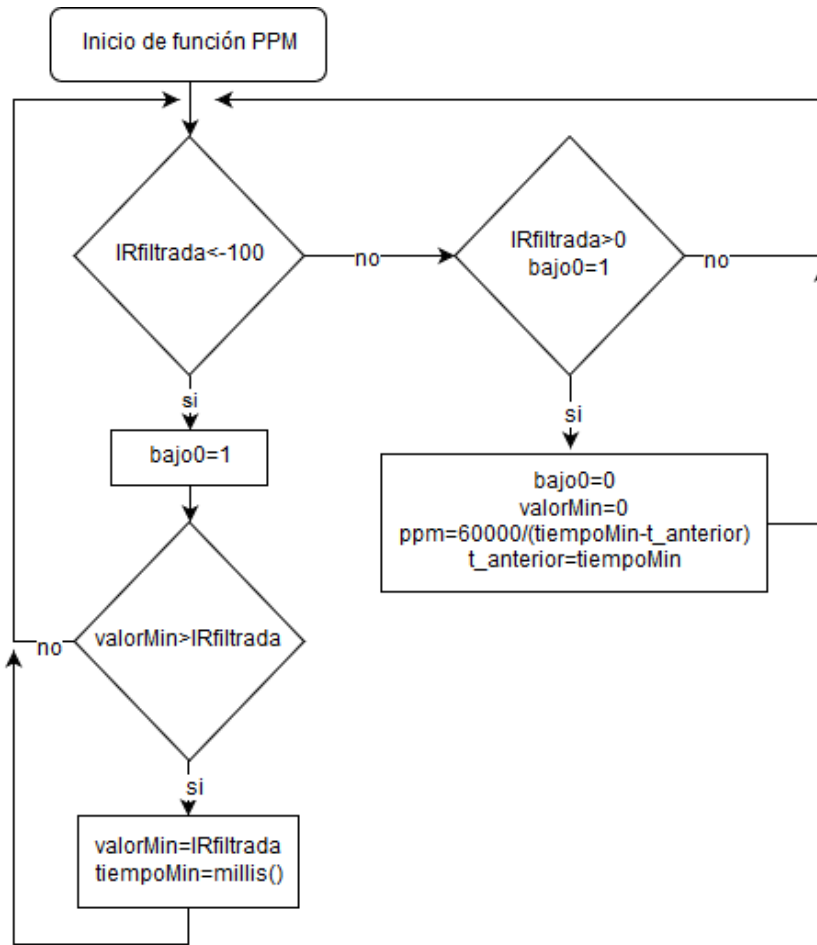


Figura 3.20 Diagrama de flujo de la función cálculo de PPM

3.4 Cálculo de la saturación de oxígeno

Como se menciona en el Capítulo 2 Sección 2.2.1 para calcular la SpO₂ se necesita obtener la proporción R descrita por la ecuación (2.1). Para obtener R se necesita encontrar la raíz de la media cuadrática (RMS) de la señal expresada en (3.4) tanto de la señal del LED rojo e infrarrojo:

$$RMS = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}} \quad (3.4)$$

Además del RMS, para calcular R es necesario obtener el valor de la señal directa sobre la que se encuentra montada la señal alterna como se muestra en la Figura 3.13 para ello es necesario conseguir el valor mínimo de la señal.

Para conseguir los valores máximos y mínimos de la señal se realiza un promedio de la señal de las últimas 100 muestras como se muestra en la Figura 3.21.

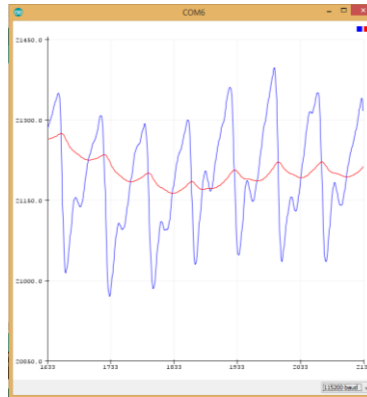


Figura 3.21 Promedio señal IR en Arduino IDE

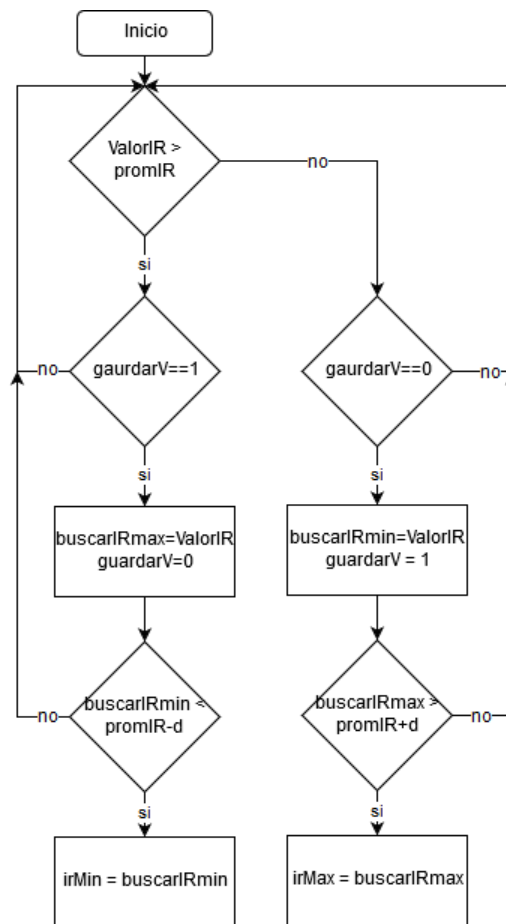


Figura 3.22 Diagrama máximos y mínimos en base al promedio de la señal

Posteriormente se compara si la señal se encuentra sobre el promedio para calcular máximos o bajo el promedio para calcular mínimos. Se obtiene los valores máximos y mínimos como se muestra en el diagrama de la Figura 3.22.

En la Figura 3.23 se pueden apreciar de manera gráfica los valores mínimos calculados mediante el método del diagrama de la Figura 3.22.

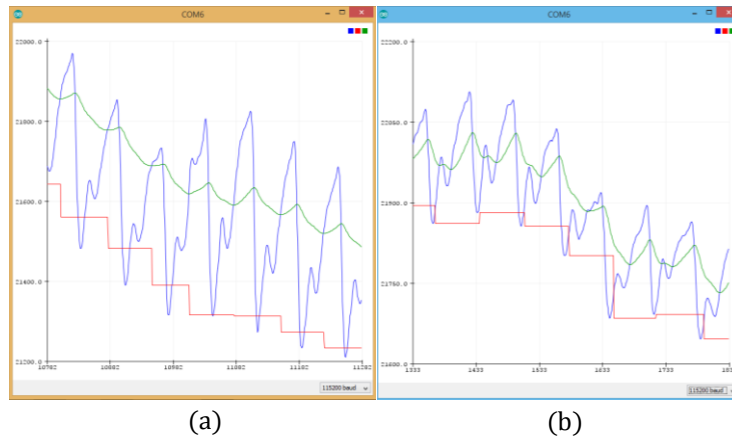


Figura 3.23 Gráfica de valores mínimos IR (a) y RED (b).

Para la señal RMS se aplicó el filtro pasabanda descrito en la Sección 3.3.1 lo que elimina la señal de corriente directa y permite obtener la señal alterna. Al aplicar la ecuación (3.4) se obtiene el valor RMS de la señal como se observa como en la Figura 3.24.

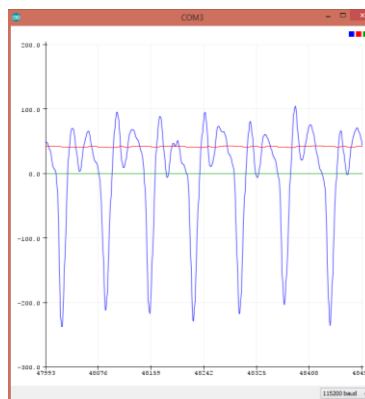


Figura 3.24 Valores RMS (rojo) aplicando filtro pasabanda.

Al obtener R mediante la ecuación (2.1) se realizó una curva empírica, Figura 3.25, comparando R y la SpO₂ del dispositivo Measupro.

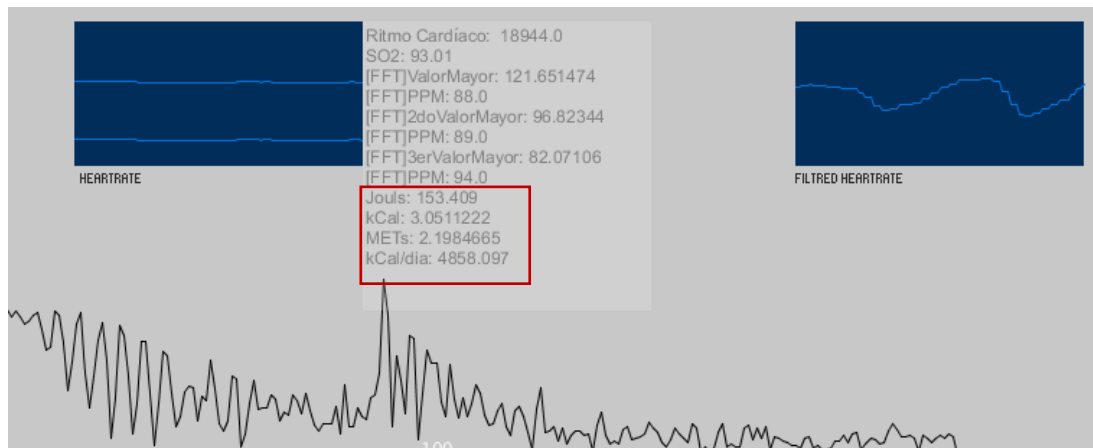


Figura 3.26 Gasto Calórico en *Processing*

3.6 Modos de funcionamiento del dispositivo

Al encender por primera vez el dispositivo inalámbrico se inicializa en modo “Apagado” en el cual el sensor MAX30100 se encuentra apagado y únicamente obtiene el valor de datos guardados en la memoria EEPROM. El modo “Apagado” espera a que se conecte al software *Processing* de la computadora para iniciar su operación.

Sin contar el modo “Apagado”, el sistema tiene tres modos de operación:

1. **Modo Sin Conexión:** Para ingresar a este modo es necesario hacer clic en el botón de software *Processing*. En este modo el microcontrolador inicializa el OP MAX30100. Posteriormente se aplica el filtro pasabanda a la señal enviada por el OP, como se muestra en la Figura 3.14, la salida del filtro se utiliza como entrada al método de cruce por cero para obtener el pulso cardíaco. El promedio de los ppm resultante se guarda en la memoria EEPROM 24LC512 y se guarda en un contador el número de datos guardados hasta el momento en la memoria. El proceso se repite cada 210 segundos (3.5 min). En la Figura 3.27 se observan los valores obtenidos del pulso durante 30 segundos. Posteriormente se muestra el número de valores que se encuentren dentro de rangos aceptables (mayores a 40ppm y menores a 220ppm) indicados. Además se muestra el promedio de los valores guardados en el arreglo dando como resultado, en esta prueba, 83.63ppm. Se muestra en la pantalla el mensaje “Guardando en MEMORIA...” como indicador de funcionamiento correcto al realizar las pruebas y por último el valor de 75ppm que es el valor guardado en memoria como resultado del procesamiento.

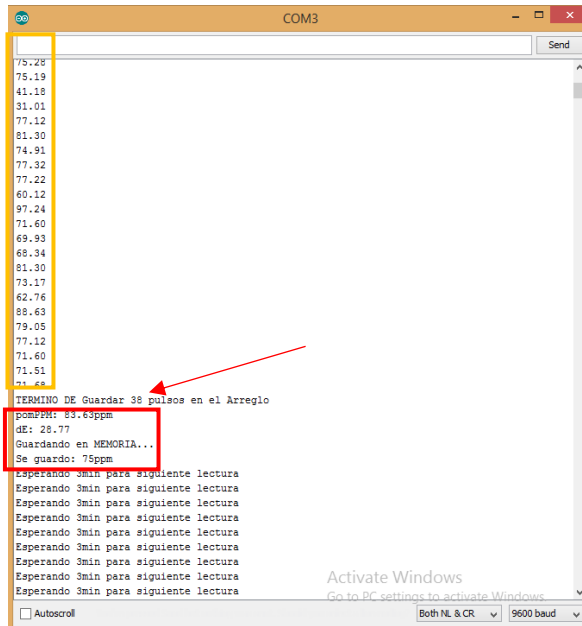


Figura 3.27 Terminal serial del programador Arduino IDE

2. **Modo Datos:** Para ingresar a este modo el dispositivo debe estar conectado al software implementado en *Processing*, el cual debe detectar que los datos guardados en la memoria EEPROM son mayor o igual a 1. Este modo se encarga de enviar todos los datos en memoria EEPROM y actualizar la variable “posMemoria” a 0, que contiene la cantidad de datos que se encuentran guardados en memoria.

3. **Modo Tiempo Real:** Para ingresar a este modo al conectarse al software de *Processing* debe de detectar que los datos guardados en la memoria EEPROM son 0. El Modo tiempo real es el que se observa en la Figura 3.28 el cual muestra los ppm y la SpO₂.

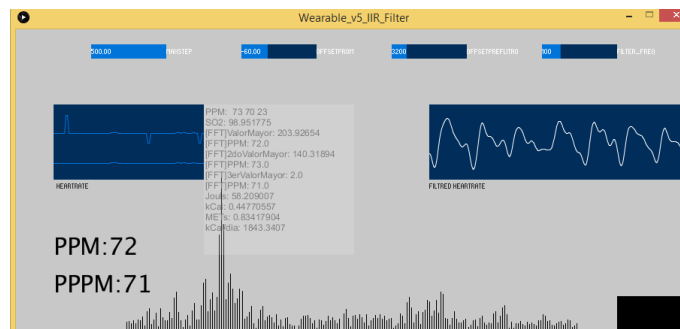


Figura 3.28 Modo Tiempo Real en *Processing*

3.6.1 Procedimiento de funcionamiento del programa del microcontrolador

1. Inicializa variables y actualiza “posMemoria” desde la memoria EEPROM.
2. Inicia en modo de operación apagado, el cual espera la conexión a la computadora.
 - a. Al iniciar comunicación con la computadora envía número de datos guardados en memoria
3. Selección automática de modo de operación dependiendo del número de datos en memoria (esto lo realiza el sistema automáticamente en base al resultado obtenido en paso 2):
 - a. Zero datos en memoria, ingresa a Modo “Tiempo Real”.
 - b. Igual o mayor a 1 ingresa a Modo “Datos” y posteriormente modo “Tiempo Real”.
4. Al presionar el botón Desconectar ingresa a modo “Sin Conexión”.

3.7 Envió de alertas por servidor local

Al estar en modo “Tiempo Real” el software de *Processing*, Figura 3.29, actúa como servidor y actualiza constantemente la frecuencia cardiaca como se muestra en la Figura 3.30. Para crear el servidor se utiliza la librería “http” de *Processing*.

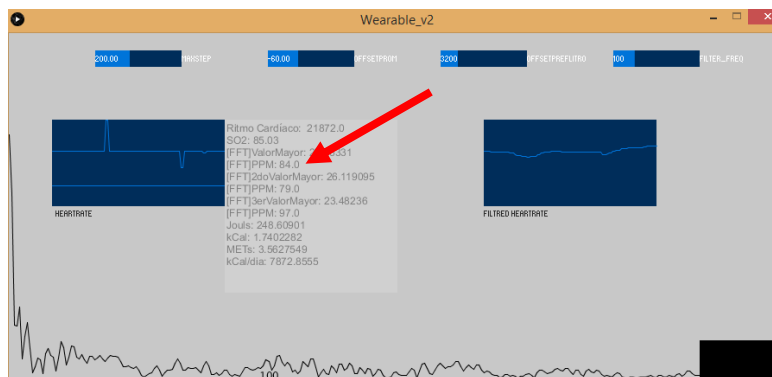


Figura 3.29 *Processing* como servidor

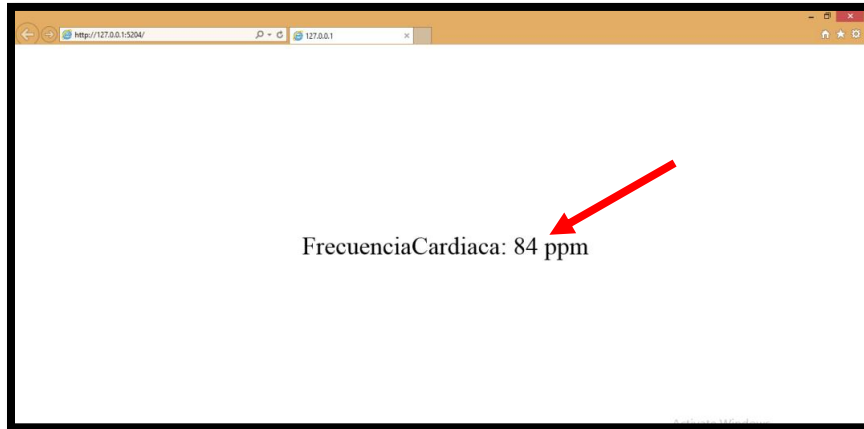


Figura 3.30 Envió de datos por Conexión IP Local

En caso de superarse límite del ritmo cardiaco se muestra un mensaje de alerta a través de una conexión de IP local, que le permite al usuario realizar una configuración de dicho límite. En este caso se utilizó para la prueba un límite superior de 100 ppm como se puede observar en la Figura 3.31.

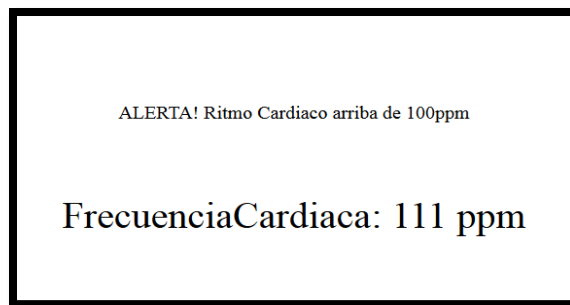


Figura 3.31 Alerta por conexión IP Local

3.8 Conclusiones

En este capítulo se presenta la selección de los componentes del prototipo inalámbrico, programación y configuración del OP MAX30100 y la memoria EEPROM 24LC512, diseño y programación de los filtros digitales y los modos de operación del dispositivo.

La aplicación del filtro pasabanda fue esencial para facilitar la detección de la frecuencia cardiaca. Además, es de vital importancia seleccionar el filtro de menor orden posible para poder utilizarlo en el microcontrolador ATmega328.

Capítulo 4

4. Resultados

En este capítulo se presenta la validación experimental del cálculo del ritmo cardiaco mediante el uso de cruce por cero y la TDF utilizando como referencia el OP Measupro. En la Sección 4.1 se presenta la metodología utilizada para encontrar el error de los métodos de cruce por cero y TDF. En la Sección 4.2 se exponen los resultados del método cruce por cero. En la Sección 4.3 se muestra los resultados de método de la TDF para la obtención del ritmo cardiaco en tiempo real. En la Sección 4.4 se indican los resultados de las ecuaciones (2.2) y (3.5) para el cálculo de la SpO₂. Por último, se presentan las conclusiones del capítulo.

4.1 Metodología general

En todos los casos para el análisis del error se utiliza la desviación estándar del error (e_{rms}), el cual es calculado mediante la ecuación (4.1):

$$e_{rms} = \sqrt{\frac{\sum_{n=1}^N (\hat{x}_k - x_k)^2}{N}} \quad (4.1)$$

donde \hat{x}_k es el valor estimado y x_k es el valor obtenido por Measupro. Las validaciones se presentan mediante las gráficas donde se comparan los valores obtenidos por ambos dispositivos. Para el caso del cálculo del ritmo cardiaco se validan los métodos de cruce por cero y TDF y en el caso de la SpO₂ las ecuaciones (2.2) y (3.5).

4.2 Resultados Pulsos por minuto (ppm) del método cruce por cero en función de ppm de Measupro

4.2.1 Metodología

Se tomaron 300 muestras del pulso cardiaco simultáneamente con el dispositivo implementado y el OP Measupro. A las muestras obtenidas por el dispositivo desarrollado se les aplicó un filtro de media móvil variando la cantidad de muestras de

la ventana. Esto con el fin de identificar el número de muestras de la ventana con menor error.

4.2.2 Pruebas

En la Figura 4.1 se muestran las lecturas realizadas por ambos dispositivos y se calculan los errores obtenidos, Figura 4.2.

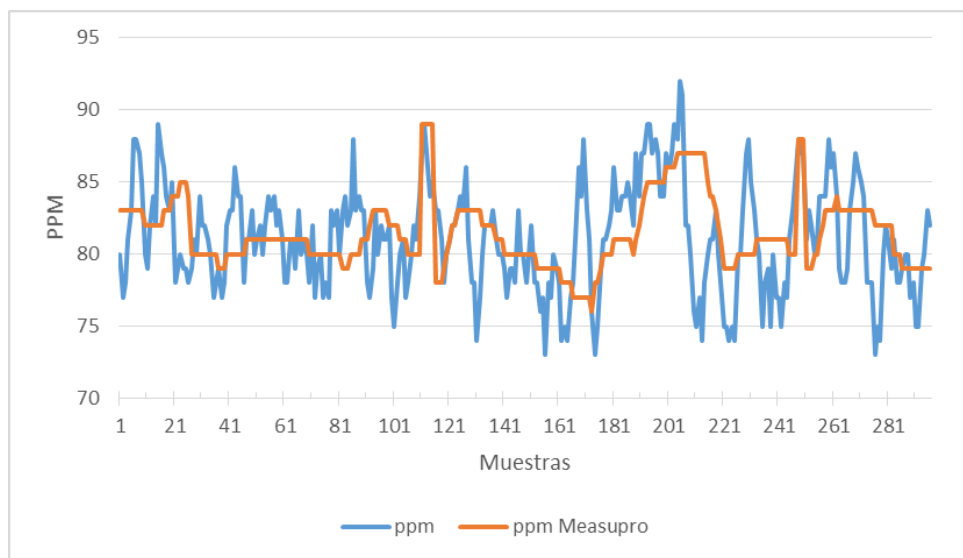


Figura 4.1 Gráfica comparativa de valores de Measupro con los medidos

El pulso calculado por el dispositivo desarrollado tiene una desviación estándar del error de ± 3.64 . El error por muestra puede ser observado en la Figura 4.2.

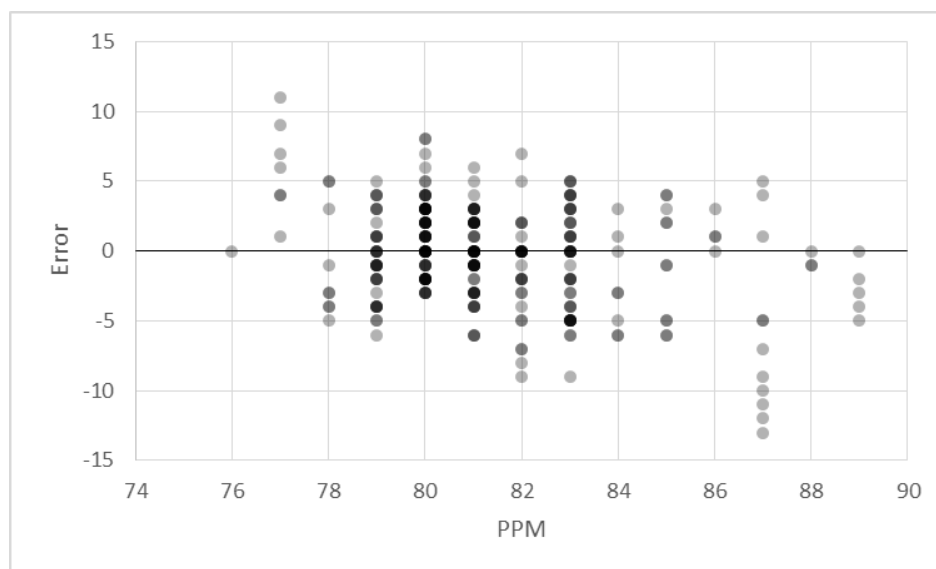


Figura 4.2 Error PPM del método Cruce por cero sin filtro en función de Ppm Measupro

En las Figuras 4.3 y 4.4 se muestran los resultados aplicando un filtro de media de ventana móvil de 10 muestras.

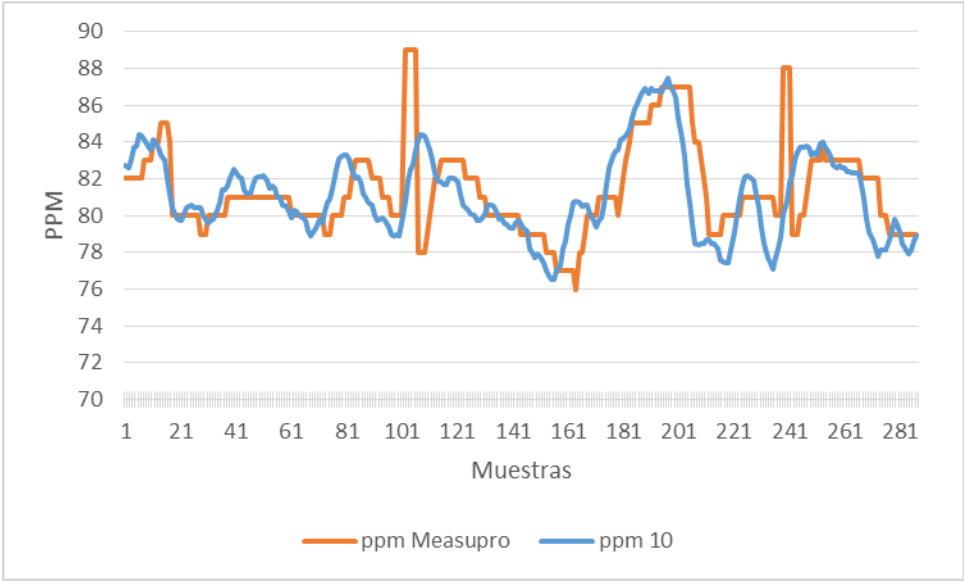


Figura 4.3 Gráfica comparativa de valores de Measupro con los medidos aplicando filtro de media 10 muestras

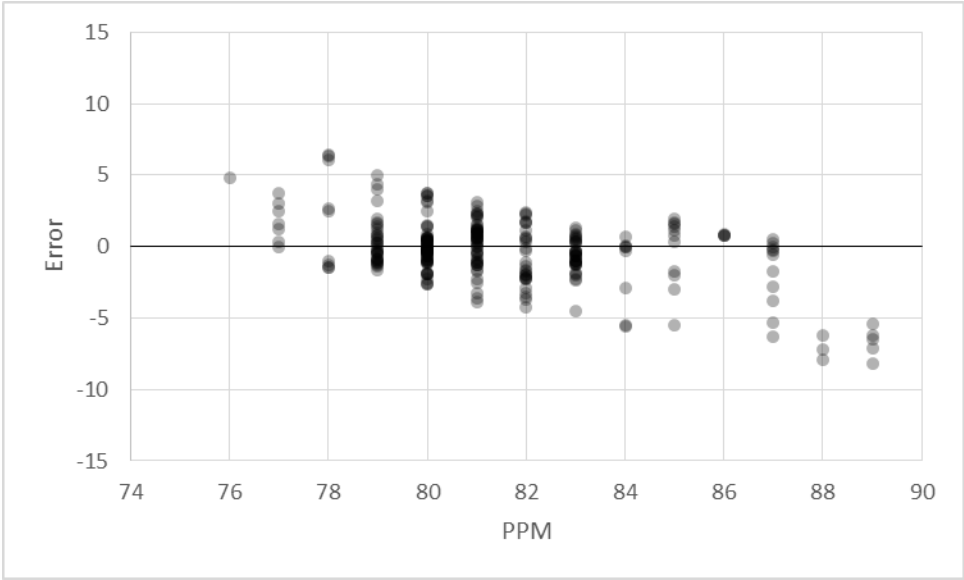


Figura 4.4 Error PPM del método Cruce por cero con filtro de media 10 muestras en función de Ppm Measupro

La desviación estándar del error al aplicar un promedio en las 10 muestras anteriores es de ± 2.19 .

En las Figuras 4.5 y 4.6 se muestran los resultados aplicando un filtro de media de ventana móvil de 20 muestras.

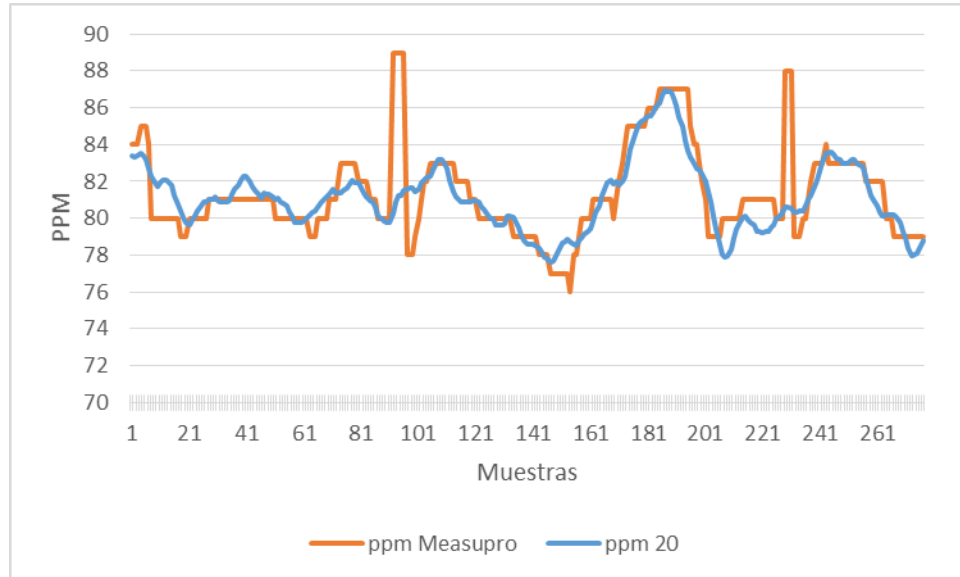


Figura 4.5 Gráfica comparativa de valores de Measupro con los medidos aplicando filtro de media 20 muestras

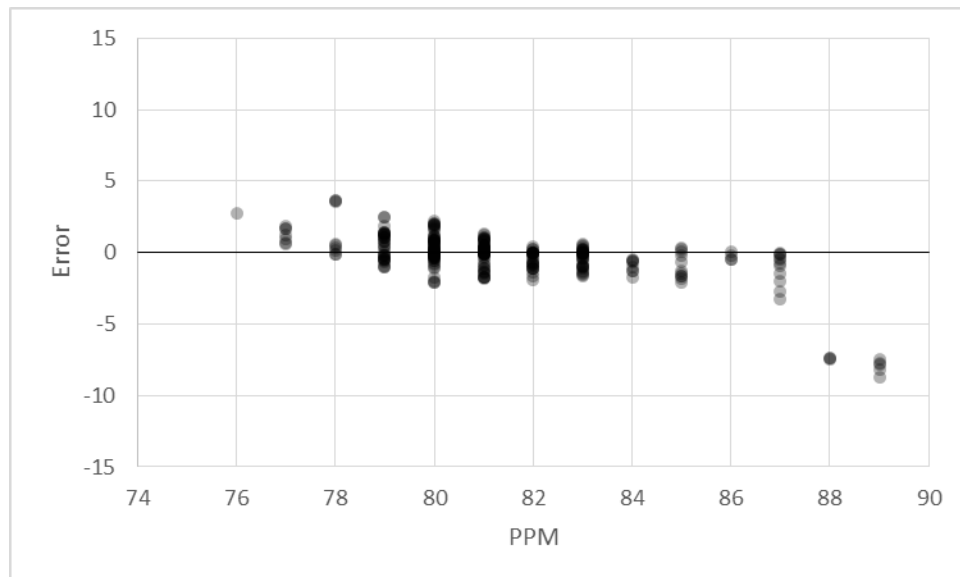


Figura 4.6 Error PPM del método Cruce por cero con filtro de media 20 muestras en función de Ppm Measupro

La desviación estándar del error al aplicar un filtro de media móvil con una ventana de 20 muestras es de ± 1.690 .

Además de las pruebas de filtros con ventanas de 10 y 20 muestras, se probó con diferentes números de muestras. Los resultados de todas las pruebas se muestran en la Tabla 4.1.

Tabla 4.1 Comparativa de aplicación de filtros de media en base a la desviación estándar del error

Filtro	Desviación Estándar del Error
Sin filtro	± 3.640
Media 10 muestras	± 2.195
Media 18 muestras	± 1.717
Media 19 muestras	± 1.697
Media 20 muestras	± 1.690
Media 21 muestras	± 1.699
Media 22 muestras	± 1.719
Media 25 muestras	± 1.800
Media 30 muestras	± 1.939

4.3 Resultados Pulsos por minuto de la transformada discreta de Fourier en función de ppm de Measupro

4.3.1 Metodología

Se configuró el MAX30100 a 50 muestras por segundo y se tomaron un total de 120 muestras a 10 usuarios diferentes. Las 120 muestras se tomaron simultáneamente de la TDF y del OP Measupro. La TDF tarda 1.5 segundos en obtener un resultado debido a que requiere 75 muestras del OP MAX30100 por lo que el tiempo total de cada prueba es de 3min.

4.3.2 Pruebas

En las Figuras 4.7 y 4.8 se muestran la comparativa de los ppm calculados mediante la TDF sin aplicar promedio a las frecuencias de mayor amplitud.

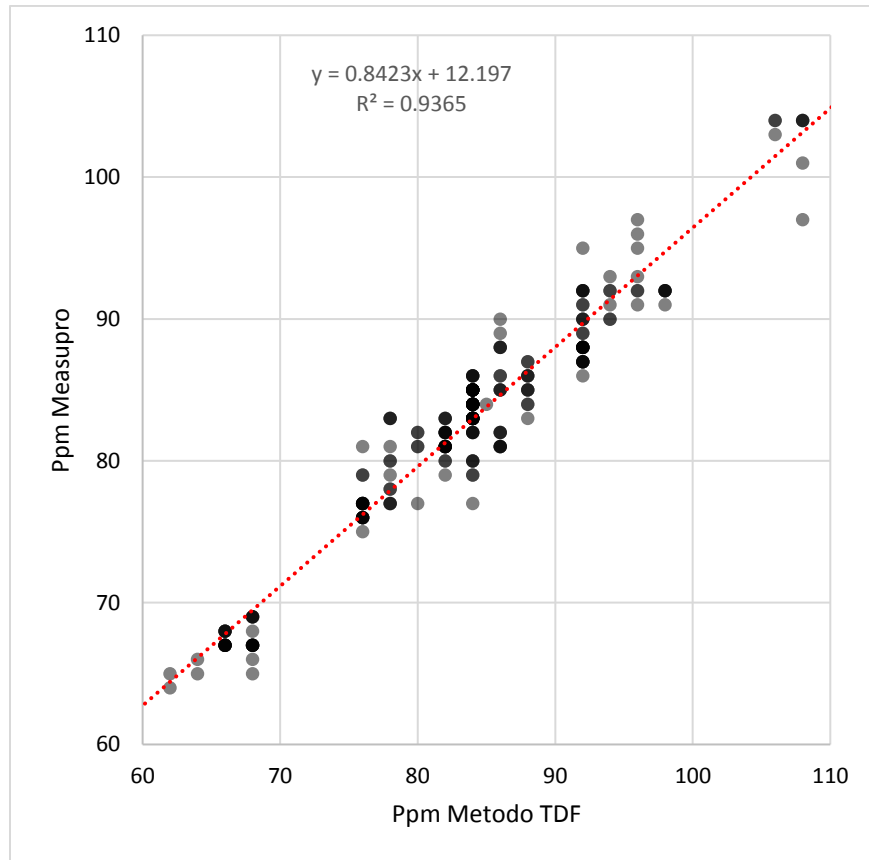


Figura 4.7 Regresión lineal PPM Measupro y Ppm TDF

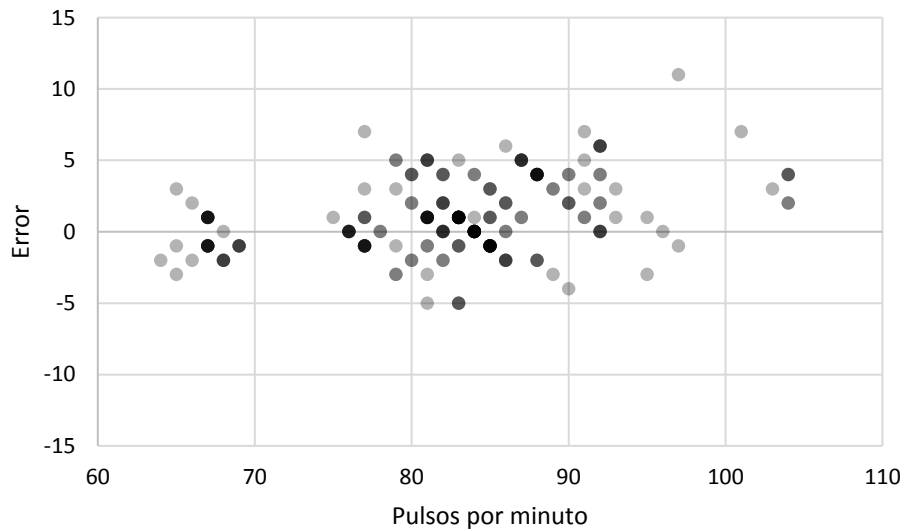


Figura 4.8 Error PPM del TDF en función de Ppm Measupro

La ecuación resultante del ppm TDF (pt) es $\text{ppm} = (0.8423 \times \text{pt}) + 12.197$; con R^2 de 0.93 (Figura 4.7). El e_{rms} es de ± 2.72 .

En las Figuras 4.9 y 4.10 se muestran la comparativa de los ppm calculados mediante la TDF al aplicar el promedio a las frecuencias de mayor amplitud.

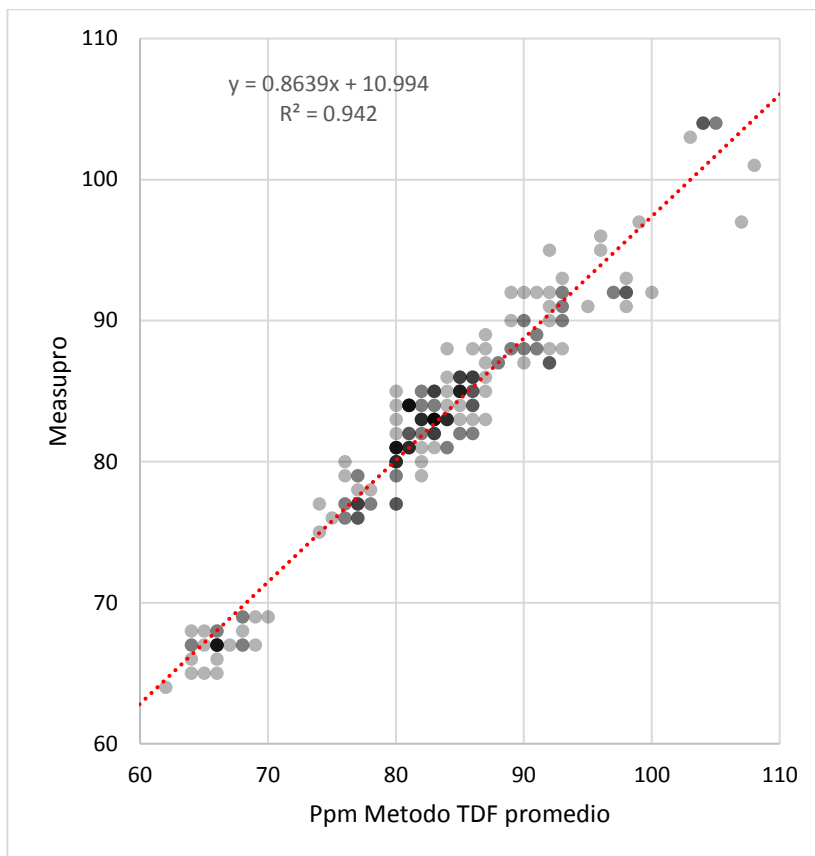


Figura 4.9 Regresión lineal PPM Measupro y Ppm TDF promedio

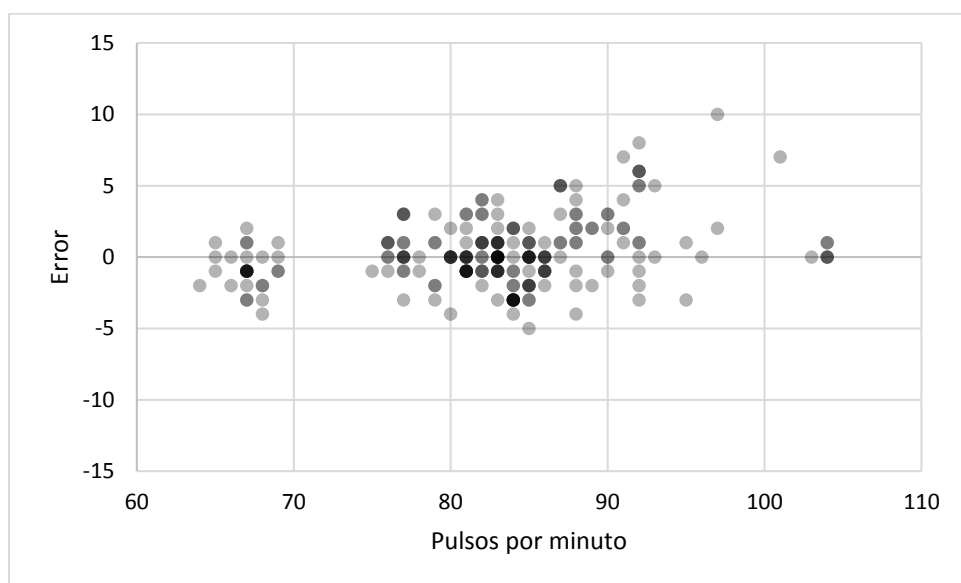


Figura 4.10 Error PPM del método TDF promedio en función de Ppm Measupro

La ecuación resultante del ppm TDF promedio (ptp) es $\text{ppm} = (0.8639 \times \text{ptp}) + 10.994$; con R^2 de 0.94 (Figura 4.9). El e_{rms} es de ± 2.34 .

4.3.1 Comparativa

Con los datos obtenidos es posible comparar los métodos de cruce por cero, TDF y TDF promedio los cuales son mostrados en la Tabla 4.2.

Tabla 4.2 Comparativa Error cuadrático medio

Método	Error cuadrático medio(e_{rms})
Cruce por cero	± 3.64
Cruce por cero filtro de media	± 1.69
TDF	± 2.72
TDF promedio	± 2.34

Se puede observar que el método de cruce por cero con filtro de media de ventana móvil de 20 muestras tiene mejores resultados para estimar el ritmo cardiaco en comparación a los métodos de la TDF. Los métodos de TDF mostraron ser precisos cuando la frecuencia cardiaca tenía pocas variaciones, sin embargo, si la variación del ritmo cardiaco tiene alta variabilidad en una medición dentro de un rango de 30 segundos presentaban errores altos.

4.4 Saturación de oxígeno capilar periférica SpO₂

4.4.1 Metodología

La validación del cálculo de SpO₂ y la prueba del sistema se lleva a cabo mediante el uso del dispositivo desarrollado y el OP Measupro simultáneamente. Para la prueba se realizan 100 muestras por segundo durante 5 minutos, dando un total de 30,000 muestras de la ecuación (2.2) y 30,000 muestras de la ecuación (3.5). A las muestras obtenidas se les aplicó un filtro de media móvil con una ventana de 1000 muestras. Se realizó una prueba con respiración normal y otra sosteniendo la respiración. En la prueba de sostener la respiración el usuario sostiene la respiración durante los primeros 30 segundos aproximadamente, posteriormente procede a respirar en forma normal hasta la mitad la mitad de la prueba donde nuevamente se abstiene de respirar durante otros 30 segundos volviendo a su respiración normal hasta finalizar el tiempo

de 5 minutos, el cual es la duración de la prueba. En ambas pruebas se utiliza las ecuaciones de SpO₂ con y sin filtro de media.

4.4.2 Resultados SpO₂ con respiración normal

Las validaciones de los resultados obtenidos de las pruebas de respiración normal se presentan mediante gráficas de las Figuras 4.11, 4.12 y 4.13 donde se comparan la SpO₂ obtenidas por las ecuaciones (3.5) y (2.2) y la SpO₂ medida mediante Measupro.

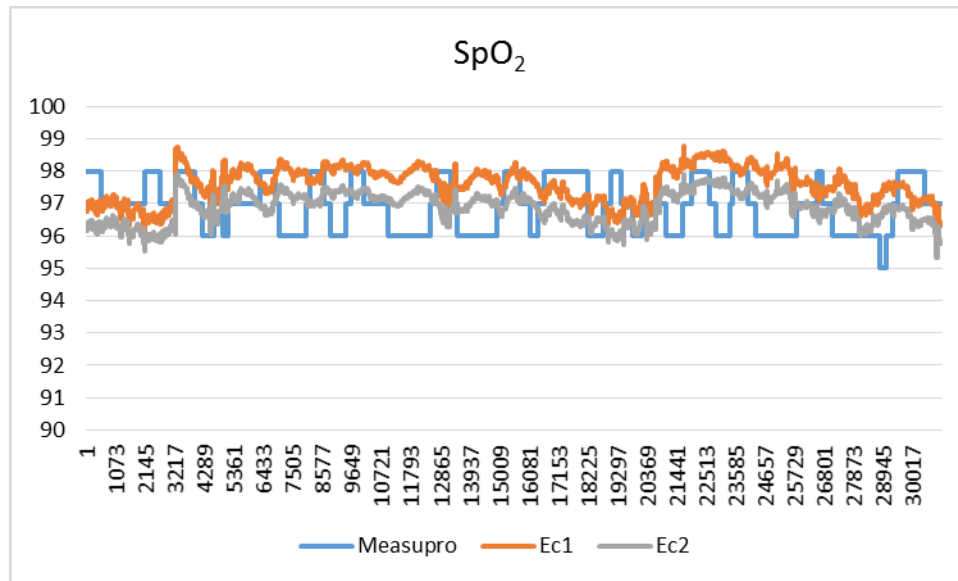


Figura 4.11 Comparativa sin filtro de la SpO₂ de la Ec. 2.2, 5.5 y Measupro respiración normal

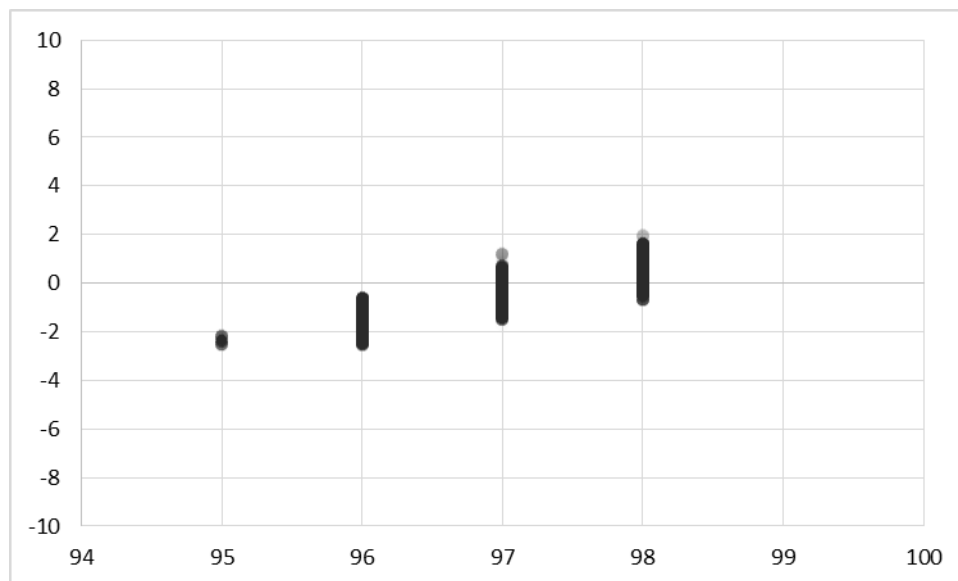


Figura 4.12 Gráfica de Error SpO₂ Ec. 2.2 respiración normal

En la Figura 4.12 se puede observar el error de 30,000 muestras utilizando la ecuación (2.2), obteniendo una desviación estándar del error de ± 1.238 .

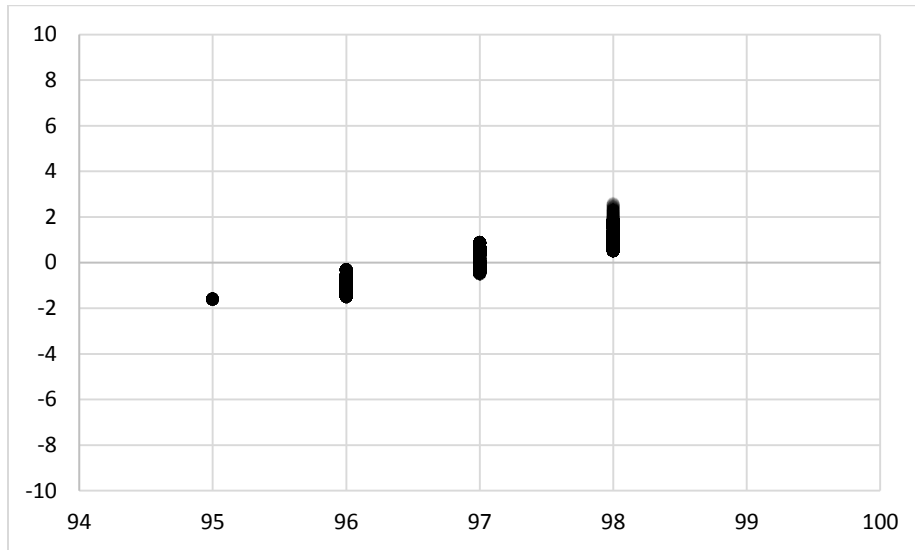


Figura 4.13 Gráfica de Error SpO₂ Ec. 3.5 respiración normal

En la Figura 4.13 se puede observar el error de 30,000 muestras utilizando la ecuación (3.5), obteniendo una desviación estándar del error de ± 0.999 .

La misma prueba se realiza utilizando el filtro de media ventana móvil de 1000 muestras, esto se puede observar en la gráfica de la Figura 4.14 así como los resultados de los errores al utilizar las ecuaciones (2.2) y (3.5) en las gráficas de las Figuras 4.15 y 4.16.

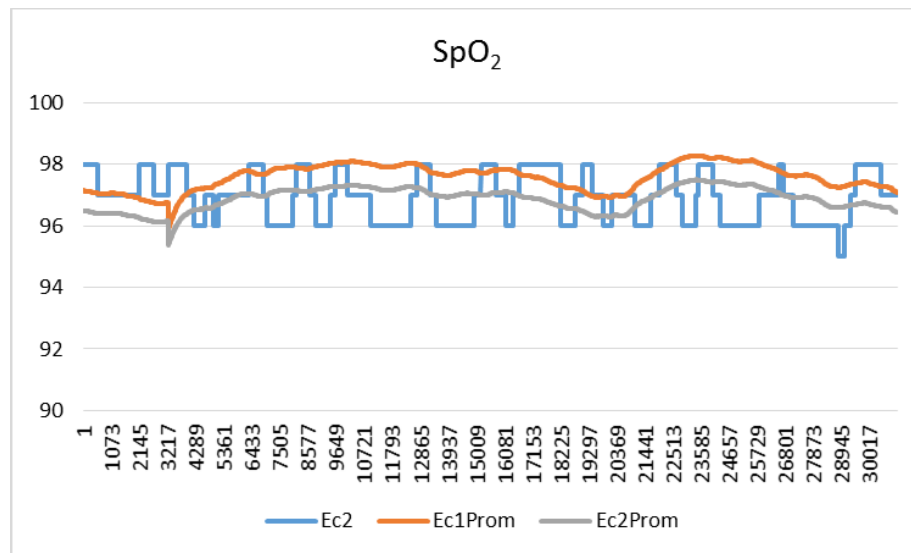


Figura 4.14 Comparativa con filtro de media de la SpO₂ de la Ec. 2.2, 5.5 y Measupro respiración normal

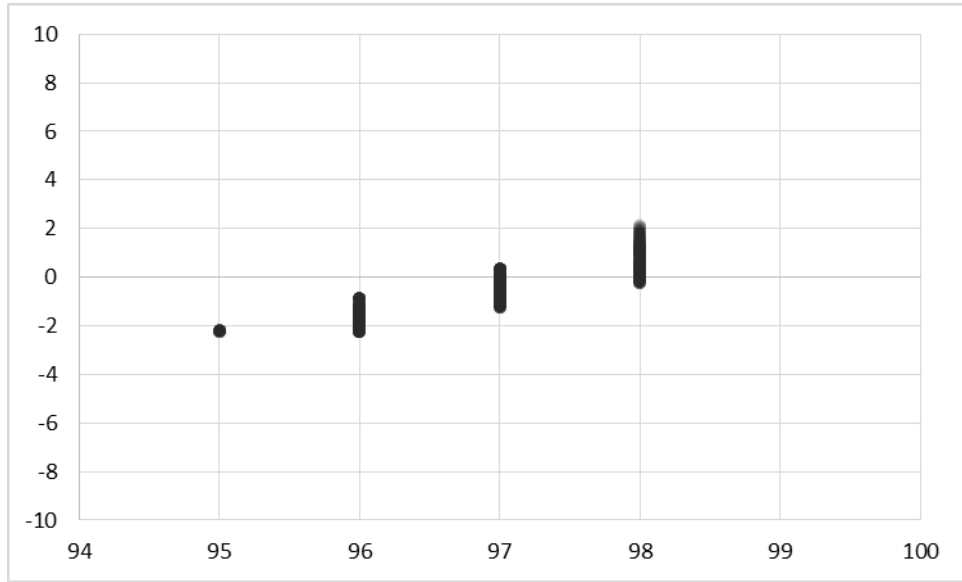


Figura 4.15 Gráfica de Error SpO₂ Ec. 2.2 con filtro de media y respiración normal

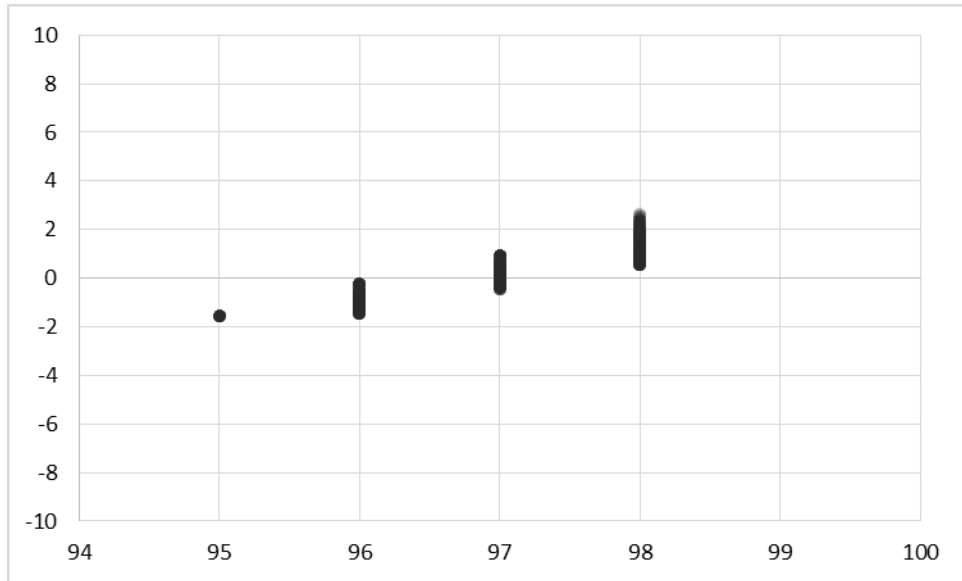


Figura 4.16 Gráfica de Error SpO₂ Ec. 3.5 con filtro de media y respiración normal

Como se puede observar en la Tabla 4.3 la SpO₂ utilizando la ecuación (3.5), muestra mejores resultados que la ecuación (2.2) tanto al usar filtro de media de 1000 muestras como sin usarlo. Sin embargo, ambas ecuaciones cumplen con la meta de un error inferior al 2%.

Tabla 4.3 Comparativa del error de la SpO₂ al respirar normalmente

	Desviación Estándar del Error	
	Ec. 2.2	Ec. 3.5
Sin Filtro	±1.238%	±0.999%
Filtro de Media 1000 muestras	±1.178%	±0.955%

4.4.3 Resultados SpO₂ al sostener la respiración

Las validaciones de los resultados obtenidos de las pruebas de respiración sostenida se presentan mediante gráficas de las Figuras 4.17, 4.18 y 4.19 donde se comparan la SpO₂ obtenidas por las ecuaciones (3.5) y (2.2) sin el uso del filtro de media y la SpO₂ medida mediante Measupro.

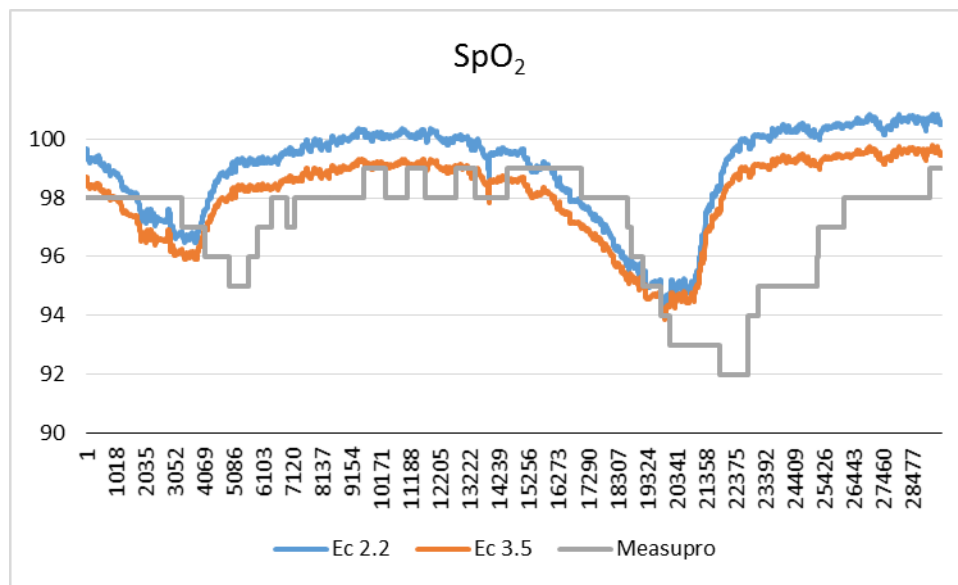


Figura 4.17 Comparativa sin filtro de la SpO₂ de la Ec. 2.2, 5.5 y Measupro

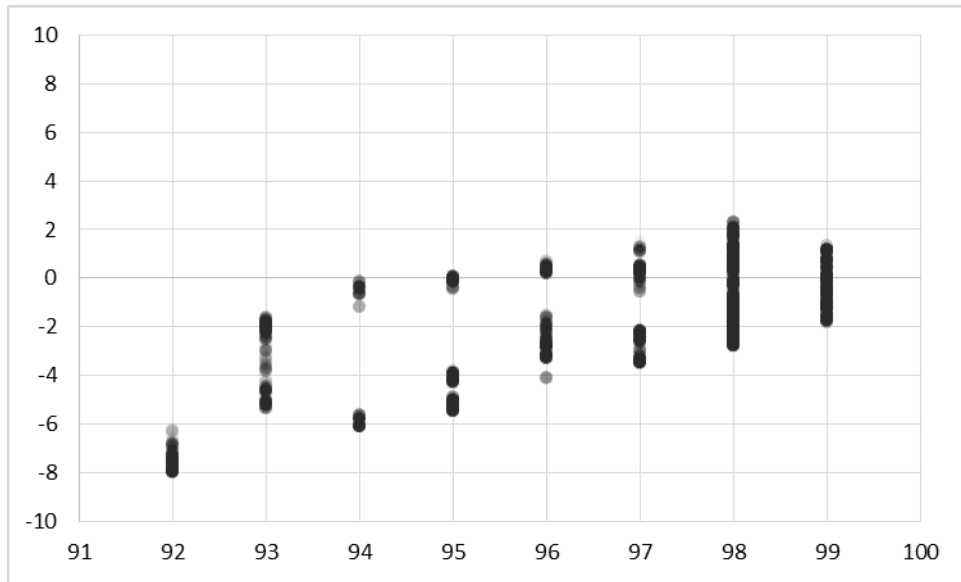


Figura 4.18 Gráfica de Error SpO₂ Ec. 2.2

La desviación estándar del error de la Ec. 2.2 sin filtro es de ± 2.777 .

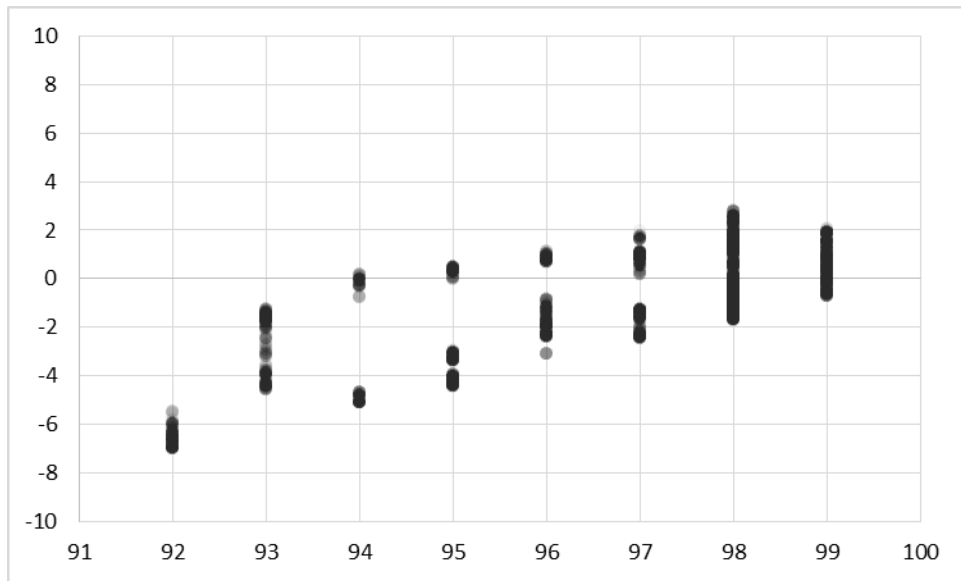


Figura 4.19 Gráfica de Error SpO₂ Ec. 3.5

La desviación estándar del error de la Ec. 3.5 sin filtro es de ± 2.209 .

La misma prueba se realiza utilizando el filtro de media ventana móvil de 1000 muestras, esto se puede observar en la gráfica de la Figura 4.20 así como los resultados de los errores al utilizar las ecuaciones (2.2) y (3.5) en las gráficas de las Figuras 4.21 y 4.22.

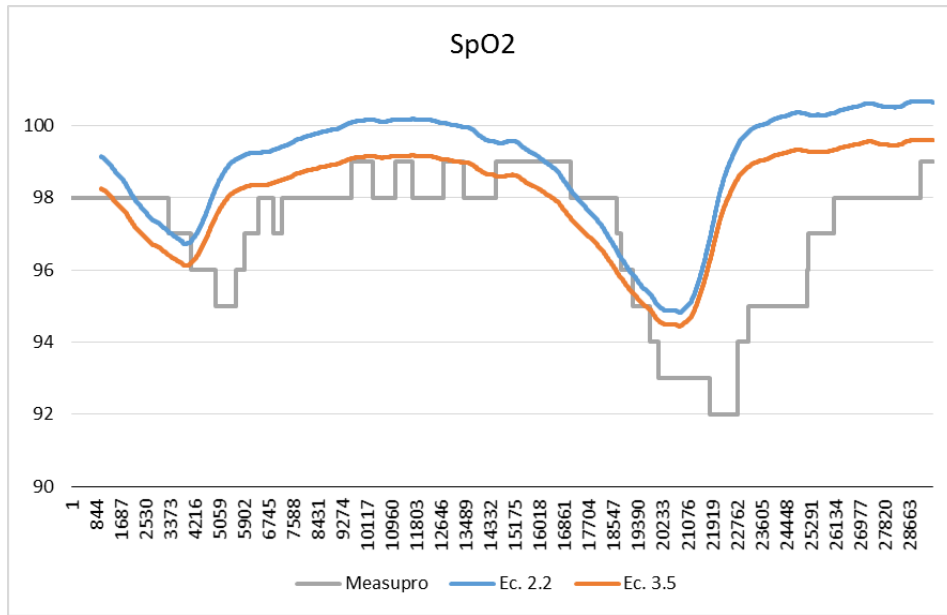


Figura 4.20 Comparativa con filtro de la SpO₂ de la Ec. 2.2, 5.5 y Measupro

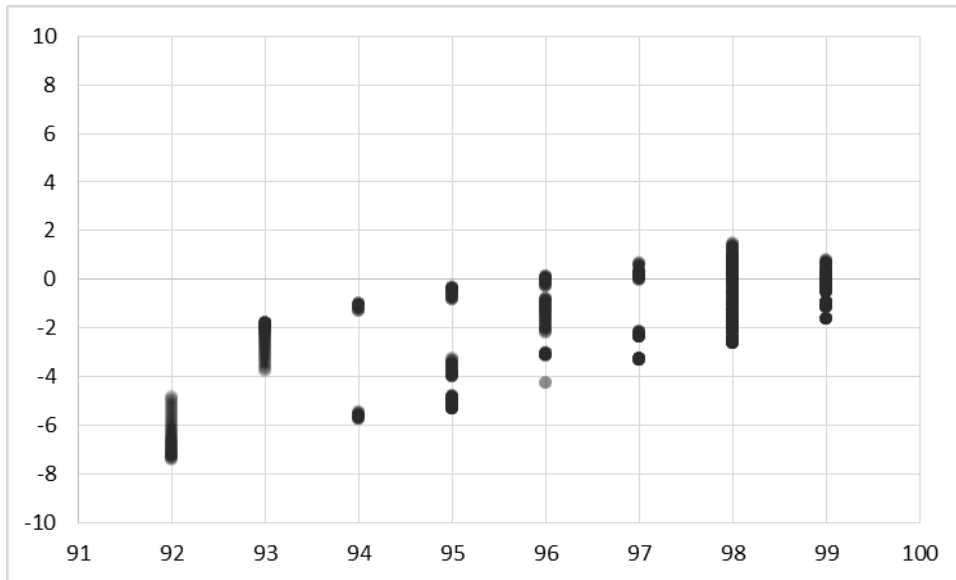


Figura 4.21 Gráfica de Error de Ec. 2.2 con filtro de media de ventana móvil de 1000 muestras

La desviación estándar del error de la Ec. 2.2 con filtro de media de ventana móvil de 1000 muestras es de ± 2.574 .

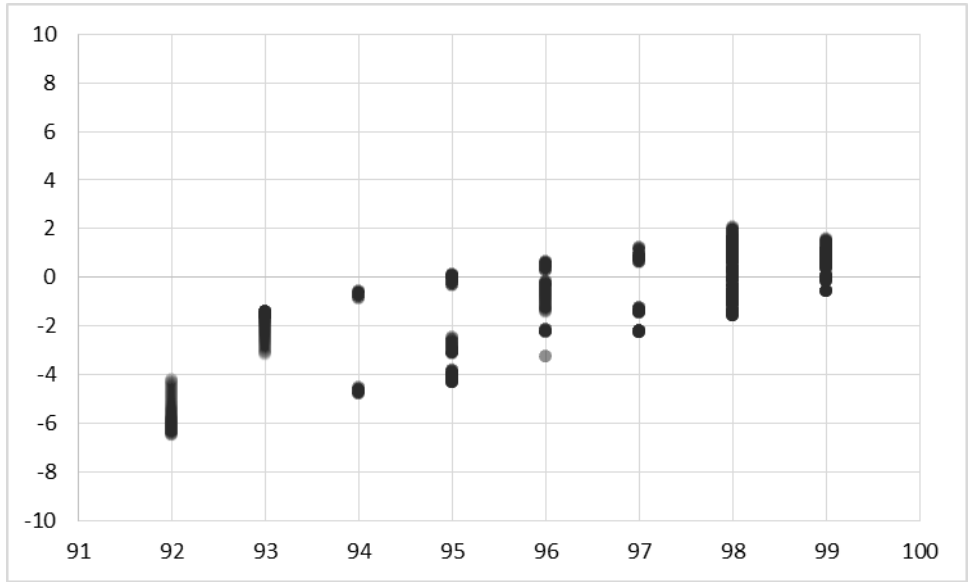


Figura 4.22 Gráfica de Error de Ec. 3.5 con filtro de media de ventana móvil de 1000 muestras

La desviación estándar del error de la Ec. 3.5 con filtro de media de ventana móvil de 1000 muestras es de ± 1.987 .

Tabla 4.4 Comparativa del error de la SpO₂ al aguantarse la respiración

	Desviación estándar del error	
	Ec. 2.2	Ec. 3.5
Sin filtro	$\pm 2.777\%$	$\pm 2.209\%$
Filtro de media 1000 muestras	$\pm 2.574\%$	$\pm 1.987\%$

Como se puede observar en la Tabla 4.4 la SpO₂ utilizando la ecuación (3.5), muestra mejores resultados que la ecuación (2.2) tanto al usar filtro de media de 1000 muestras como sin usarla. Sin embargo, únicamente la ecuación (3.5) al utilizar filtro de media de 1000 muestras cumple con la meta de un error inferior al 2%.

4.5 Conclusiones

En este capítulo se presentó la validación de los tres métodos para el cálculo del ritmo cardiaco y las dos ecuaciones de la SpO₂ mediante el uso del prototipo desarrollado. Se utilizó el error cuadrático medio como herramienta de validación y

comparación de los métodos. En el caso del ritmo cardiaco, todos los métodos obtuvieron resultados satisfactorios con errores menores a ± 5 ppm. El método de cruce por cero obtuvo los mejores resultados con un error cuadrático medio de ± 1.82 . Se puede observar en la Tabla 4.2 que el método de cruce por cero obtuvo mejores resultados para estimar el ritmo cardiaco en comparación a los métodos de la TDF. Los métodos de TDF muestran ser precisos cuando la frecuencia cardiaca tenía pocas variaciones. Por último, los resultados en las mediciones de la SpO₂ muestran una mejor respuesta para la ecuación (3.5) aplicando el filtro de media de ventana móvil de 1000 muestras con una desviación estándar del error de $\pm 0.955\%$ cuando se respiraba normalmente.

Capítulo 5

5. Conclusiones

5.1 Conclusiones finales

En este proyecto de tesis se desarrolló un sistema para prototipo inalámbrico para medición de ritmo cardiaco, SpO₂ y gasto calórico con sistema de alertas. Para esto se investigó y se implementaron métodos para el cálculo del ritmo cardiaco, SpO₂ y gasto calórico, así como filtros digitales IIR y la TDF para el desarrollo del software. Posteriormente se seleccionó el ritmo cardiaco para calcular el gasto calórico y el filtro pasabanda para eliminar el ruido en la señal del OP para obtener en ritmo cardiaco y la SpO₂. Por último, se validaron los tres métodos para el cálculo del ritmo cardiaco y 2 ecuaciones para el cálculo de la SpO₂ mediante pruebas físicas del prototipo y el uso del error cuadrático medio.

El resultado más importante de este proyecto es que se demuestra que es posible obtener un error menor a ± 5 ppm mediante los métodos cruce por cero y TDF así como obtener error menor a $\pm 2\%$ en la medición de la SpO₂ al respirar normalmente. Además, fue posible mostrar el gasto calórico en tiempo real y mostrar alertas en caso de superar límites del ritmo cardiaco utilizando IP local.

5.2 Trabajos futuros y recomendaciones

Queda para futuros trabajos de investigación el aplicar un sistema de alertas que sean enviadas por GSM mediante un correo electrónico, mensaje o llamada, uso de GPS para conocer ubicación y miniaturizar prototipo para uso práctico.

Se recomienda realizar pruebas con un dispositivo de mayor precisión tanto para el cálculo del ritmo cardiaco como para la medición de la SpO₂ y realizar pruebas con la TDF configurando la velocidad de muestreo del OP MAX30100 a diferentes velocidades de muestreo.

ANEXOS

Arduino Software IDE: software de código abierto escrito en Java y basado en Processing y otros software de código abierto. Utilizado principalmente para la programación de los microcontroladores con plataforma Arduino.

Bluetooth: estandarizado por la IEEE como IEEE 802.15.1 el cual es una tecnología inalámbrica para el intercambio de datos en el ancho de banda de 2.4 a 2.485 GHz.

Processing Software: software gratuito y de código abierto utilizado principalmente en la enseñanza de programación de software en el área de diseño y artes visuales. Aunque también es herramienta de desarrollo de profesionales tanto en las áreas antes mencionadas como en el prototipado de software y visualización de datos.

Wearable: dispositivo electrónico con microcontroladores que puede ser utilizado en el cuerpo mediante implantes o accesorios.

Referencias

- [1] F. Xiao, «Fast Design of IIR Digital Filters With a General Chebyshev Characteristic,» *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS*, vol. 61, nº 12, p. 962, 2014.
- [2] J. Wilkinson, «Medical Market for Microsystem,» *International Newsletter on Microsystems and MEMS*, p. 37, 2002.
- [3] A. Lymberis, «Wearable health systems and applications: The contribution of information and communication technologies,» de *Engineering in medicine and biology*, Shanghai, China, 2005.
- [4] G. G. R. G. K. J. T. Akshay Sugathan, «Application of Arduino Based Platform for Wearable Health Monitoring System,» de *IEEE 1st International Conference on Condition Assessment Techniques in Electrical Systems*, CHENNAI, INDIA, 2013.
- [5] J. Dyro, «Technology Problems in Mexico,» de *Clinical Engineering Handbook*, Academic Press, 2004, pp. 80-83.
- [6] OCDE, «oecd.org,» 2013. [En línea]. Available: http://www.oecd.org/mexico/Health-at-a-Glance-2013-Press-Release-Mexico_in-Spanish.pdf. [Último acceso: 20 Diciembre 2015].
- [7] C. Choi, B.-h. Ko, T. Lee, G. Park y K. Shin, «Exercise Amount Calculation Using a Wearable Half-Cell Potential,» de *35th Annual International Conference of the IEEE EMBS*, Japon, 2013.
- [8] C. M. Nieves, «Evaluating the effectiveness of a wearable thecnology for increasing physical activity,» Graduate Theses and Dissertations., 2014. [En línea]. Available: <http://scholarcommons.usf.edu/etd/5542>.
- [9] A. Pantelopoulos y N. G. Bourbakis, «A Survey on Wearable Sensor-Based Systems,» *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETIC*, vol. 40, nº 1, 2010.
- [10] E. Sardini y M. Serpelloni, «Instrumented Wearable Belt for Wireless Health Monitoring,» *Procedia Engineering* 5, pp. 580-583, 2010.

- [11] M. Tavana, «A reconfigurable supporting connected health environment,» de *Healthcare Informatics and Analytics: Emerging Issues and Trends*, IGI Global, 2014, p. 236.
- [12] A.S.Ekhare, «Design and Development of Low Unit Cost and Longer Battery Life Wireless Pulse Oxymetry System,» de *Fourth International Conference on Advances in Computing and Communications*, Maharashtra, India, 2014.
- [13] D. K. Radovan Stojanovic, «Design of an Oximeter Based on LED-LED Configuration and FPGA Technology,» *Sensors*, pp. 574-579, 2013.
- [14] K. D. M. D. R. S. Hirofumi Tanaka, «Age-Predicted Maximal Heart Rate Revisited,» *Journal of the American College of Cardiology*, vol. 37, n^o 1, pp. 153-155, 2001.
- [15] E. ElSamahy y A. Genedy, «A Computer-based System for Safe Physical Fitness Evaluation,» de *4th International Conference on Biomedical Engineering and Informatics*, Cairo, Egipto, 2011.
- [16] G. B. d. Pozo, «D4.1 State of the Art - Wearable Sensors,» Octubre 2014. [En línea]. Available: <http://www.daphne-fp7.eu/sites/default/files/D4.1%20State%20of%20the%20Art%20Wearable%20Sensors%20.pdf>. [Último acceso: Junio 2016].
- [17] D. M. Karantonis, «Implementation of a Real - Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring,» *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, vol. 10, n^o 1, pp. 156-167, 2006.
- [18] S. J. D. R. B. J. A. M. S. a. D. L. T. STRATH, «Simultaneous heart rate-motion sensor technique to estimate energy expenditure,» *Med. Sci. Sports Exerc*, vol. 33, n^o 12, p. 2118–2123, 2001.
- [19] J. C. a. D. B. J. SE Crouter, «Accuracy of the Actiheart for the assessment of energy expenditure in adults,» *European Journal of Clinical Nutrition*, vol. 62, p. 704–711, 2007.
- [20] Atmel, «ATmega328/P hoja de datos,» [En línea]. Available: http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf. [Último acceso: 2017].
- [21] C. O. A. Milenković y E. Jovanov, «Wireless sensor networks for personal health monitoring: Issues and an implementation,» de *Computer Communications*, Huntsville, Elsevier, 2006, p. 2521–2533.

