

Acquisition of pulse oximetry signals and their remote processing with Matlab

Carlos Sánchez-Márquez¹, Jorge Contreras-Cossío² Johnny Romero-Milián³
Universidad Privada del Norte, Facultad de Ingeniería Electrónica y Mecatrónica, Los Olivos, Lima-Perú
¹augusto.sanchez@upn.pe, ²jorge.contreras@upn.edu.pe, ³johnny.romero@upn.edu.pe

Abstract– The study and treatment of the pulse oximetry signal, in these times of the pandemic due to COVID 19, becomes more important and necessary every day. Considering this problem, this research work is summarized in the remote processing of pulse oximetry signals. This is intended to support the health sector in the medical diagnosis of cardiorespiratory diseases in patients who live in the interior of our country or specifically in rural areas, where it is not possible to count on the professional services of clinical specialists. In this sense, in this work the technologies associated with the IoT are applied, through which the processes of transmission, reception and processing of the pulse oximetry signal are facilitated. It is necessary to take into account the influence of internal and external disturbances that can affect the pulse oximetry signal, and therefore the processing of the signal is essential to obtain a signal without any level of noise, and that helps the doctor to make a good clinical diagnosis of patients. For the processing of the pulse oximetry signal, the Matlab platform has been used, through the development of digital filter programs. There are many other platforms on the market to achieve this purpose. We hope that this contribution will help reduce the incidence of cardiorespiratory diseases.

Keywords: *IoT, oxygen saturation, signal processing, pulse oximetry.*

Digital Object Identifier (DOI):

<http://dx.doi.org/10.18687/LACCEI2022.1.1.644>

ISBN: 978-628-95207-0-5 **ISSN:** 2414-6390

Adquisición de señales de pulso oximetría y su procesamiento remoto con Matlab

Carlos Sánchez-Márquez¹, Jorge Contreras-Cossío², Johnny Romero-Milián³
Universidad Privada del Norte, Facultad de Ingeniería Electrónica y Mecatrónica, Los Olivos, Lima-Perú

¹augusto.sanchez@upn.pe, ²jorge.contreras@upn.edu.pe, ³johnny.romero@upn.edu.pe

Resumen– El estudio y tratamiento de la señal de pulso oximetría, en estos tiempos de la pandemia a causa del COVID 19, se vuelve cada día más importante y necesario. Considerando este problema, este trabajo de investigación se resume en el procesamiento de señales de saturación de oxígeno de manera remota. Con ello se pretende apoyar al sector salud en el diagnóstico o médico de enfermedades cardiorrespiratorias en pacientes que habitan el interior de nuestro país o específicamente en las zonas rurales, en donde no es posible contar con los servicios profesionales de los especialistas clínicos. En ese sentido, en este trabajo se aplica las tecnologías asociadas al IoT, mediante el cual se facilitan los procesos de transmisión, recepción y procesamiento de la señal de pulso oximetría. Es preciso tener en consideración la influencia de las perturbaciones internas y externas que pueden afectar la señal de saturación de oxígeno, y por ello el procesamiento de la señal resulta fundamental, para obtener una señal sin ningún nivel de ruido, y que ayude al médico a realizar un buen diagnóstico clínico de los pacientes. Para el procesamiento de la señal de pulso oximetría se ha utilizado la plataforma Matlab, mediante el desarrollo de programas de filtros digitales. En el mercado existen muchas otras plataformas para lograr este propósito. Esperamos que este aporte ayude en la disminución de la incidencia de enfermedades cardiorrespiratorias.

Palabras Clave: IoT, saturación de oxígeno, procesamiento de señales, pulso oximetría.

I. INTRODUCCIÓN

El presente trabajo de investigación se enfoca en el procesamiento de la señal de pulso oximetría de manera remota [1], buscando solucionar la falta de los servicios médicos especializados en las zonas del interior del Perú, como es el caso de las especialidades de otorrinolaringología, neumología o cardiología. Según las estadísticas del MINSA y el INEI [2], existen escasos centros médicos en las zonas rurales del Perú, y a causa de la pandemia, es necesario contar con médicos especialistas en enfermedades cardiorrespiratorias. Por ese motivo, es necesario desarrollar un sistema de diagnóstico a distancia, utilizando la tecnología asociada al IoT, en donde de manera remota [3], se pueda enviar los resultados de las pruebas de un paciente, hasta un médico especialista que se encuentra en un consultorio de un hospital o clínica en una gran ciudad. Con este trabajo, se brindaran mejores oportunidades para solucionar los problemas de salud que presentan los pacientes rurales [4], quienes en muchos casos deben realizar largos

viajes a las grandes ciudades, buscando solucionar sus problemas de salud.

En resumen, las etapas fundamentales del proyecto son la etapa de adquisición y acondicionamiento de la señal mediante módulo WI-FI [5], la etapa de configuración del canal de transmisión de información [6], el programa modular en Arduino IDE adquirir y transmitir la señal al módem y colgar en la nube de Internet la información de las señales. Por otro lado, la etapa de exportación y procesamiento de la señal de pulso oximetría.

A. Objetivo general

Adquirir señales de pulso oximetría de manera remota y efectuar un procesamiento remoto mediante Matlab.

B. Objetivos específicos

Implementar un circuito para la adquisición y transmisión de señales de pulso oximetría.

Crear programas en Matlab, para el procesamiento de la señal de pulso oximetría.

II. MARCO TEORICO

A. Señal de pulso oximetría

La señal de pulso oximetría SPO₂, no permite conocer el porcentaje de oxígeno en la sangre y puede ser adquirida a partir de cualquier dedo de la mano del paciente. En condiciones normales, las variaciones de esta señal se sincronizan con el ritmo cardiaco y su valor máximo está en fase con la onda R de la señal cardiaca. Esto permite a la señal de pulso oximetría ser la referencia para la estimación de la frecuencia cardiaca instantánea [1].

Un dispositivo llamado oxímetro de pulso, mide los niveles de oxígeno en la sangre. Este nivel de oxígeno en la sangre se denomina “nivel de saturación de oxígeno” (SPO₂).

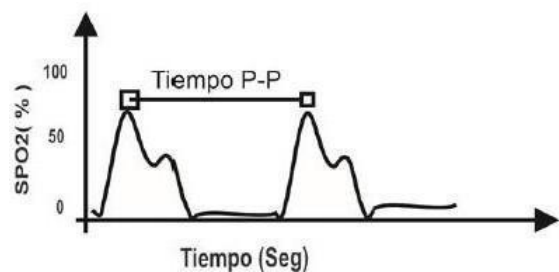


Figura N° 1. Señal de pulso oximetría, SPO₂
Fuente: González-Velandia-Lyma-Ospina [1]

Digital Object Identifier (DOI):
<http://dx.doi.org/10.18687/LACCEI2022.1.1.644>
ISBN: 978-628-95207-0-5 ISSN: 2414-6390

Este porcentaje indica la cantidad de oxígeno que transporta el torrente sanguíneo en relación a un nivel máximo que podría transportar [6]. La SPO2 promedio a nivel del mar es entre 97 a 99%, con límites inferiores de 94%. La saturación de oxígeno tiene relación con la tensión arterial de oxígeno (PAO2), es decir que a pequeños cambios en la SPO2 se producen grandes cambios en la PAO2, por esa razón es recomendable que al nivel del mar se use oxígeno en pacientes con SPO2 menor a 90% y no menor a 94% [7].

B. El integrado MCU ESP8266

El integrado ESP8266 MCU está orientado a Internet de las cosas (IoT), y tiene como núcleo el SoM ESP-12E que a su vez se basa en el SoC Wi-Fi ESP8266, e integra el conversor CP2102 USB-Serial TTL y el conector micro-USB necesario para la programación y comunicación con la computadora. Se graba igual que Arduino a través de su cable de datos. Los pines de entrada/salida (GPIO) se alimentan con 3,3 V y la entrada dañada se conecta al pin A0 [8]. Este chip resulta en una excelente solución de red Wi-Fi autónoma [9].

C. ThingSpeak

Es una herramienta de software que se puede utilizar de forma gratuita y para compartir datos en canales públicos de la nube. Es una plataforma de Internet de las cosas (IoT) que permite recopilar y almacenar datos de sensores en la nube y desarrollar aplicaciones de IoT. A partir de los datos almacenados, se pueden analizar y efectuar procesamiento de señales con Matlab u otra plataforma, además se puede generar alarmas, programar eventos- Funciona con Arduino, Raspberry, ESP8266 entre otros. La aplicación genera dos claves API y un ID de identificación, que se especifica en el código para lograr la comunicación [10].

G. Acerca de IoT

Cuando hablamos de Internet de las Cosas, IoT, nos referimos a la interacción entre la conectividad de la red de Internet y la capacidad de cómputo que se extiende a objetos, sensores y artículos de uso diario, permitiendo que se generen e intercambien datos, sin la intervención humana [11]. Como los objetos y las máquinas se relacionan entre sí en su entorno todos los días, las poblaciones están cada vez más conectadas o entrelazadas [12]. Así mismo, IoT utiliza protocolos para la comunicación de datos. Uno de los protocolos más importantes es el Protocolo de aplicación restringida (MQTT), y nos permiten conocer el estado de salud de una red IoT [13]. Algunos la definen como la infraestructura de red global dinámica con capacidades de autoconfiguración basadas en estándares y protocolos de comunicación [14], [15]. MQTT es un protocolo de mensajería de publicación/suscripción extremadamente simple y liviano diseñado para dispositivos restringidos y redes de bajo ancho de banda, alta latencia o poco confiables [16].

III. MATERIALES Y MÉTODOS

En esta etapa de la investigación, se realizó una revisión

sistemática de la bibliografía existente a nivel mundial en base de datos extraídos de ResearchGate, Redalyc, IEEEExplore, Scencedirect y de bases de datos de tesis de Universidades cuyos repositorios cuentan con artículos originales y manuales tecnológicos relacionados con la adquisición y procesamiento de señales de pulso oximetría. De cada revista o tesis se seleccionaron manualmente los trabajos relacionados con procesamiento de señales de pulso oximetría, publicados entre 2016 y 2022. De todos los originales, 8 fueron seleccionados para el estudio. En cuanto a las variables consideradas, estas fueron: el tema de la investigación, la tecnología a aplicar en la investigación, el procesamiento de la señal de manera remota, el procedimiento de transmisión y recepción de la señal y la respuesta de la señal de pulso oximetría. En ese sentido, surgió la pregunta de investigación: ¿La adquisición y transmisión de señales de pulso oximetría mediante tecnología IoT garantizan el procesamiento de la señal exportada a Matlab? Asimismo, el diseño del prototipo se realizó en un periodo de dos meses, previo estudio de las variables. Algunas etapas del estudio, diseño e implementación se realizaron de manera transversal. Se aplicaron la plataforma ThingSpeak [17] y de Mathworks [18] y los modelos y ejemplos de algunos programas sirvieron como guía. El trabajo pudo ser terminado, gracias al aporte de profesionales de diversas áreas de la ingeniería.

IV. RESULTADOS

A. Adquisición de la señal de pulso oximetría

La señal de pulso oximetría puede ser adquirida por diversos métodos y tecnologías, como por ejemplo utilizando un oxímetro de pulso o también puede ser adquirida por un simple dispositivo conectado a Arduino [9]. Además, se puede adquirir una señal de pulso oximetría a través amplificadores operacionales o un circuito de instrumentación, con otras etapas como filtros de atenuación de ruido, entre otros. En este trabajo de investigación se ha utilizado para la adquisición de datos el sensor MAX30102 de saturación de oxígeno para Arduino y el módulo MCU EPS8266 [8] para la transmisión y almacenamiento de datos en la nube. Asimismo, se ha utilizado la plataforma Matlab ThingSpeak [17] para canalizar la señal de pulso oximetría, a la nube, a través de un canal que se configura previamente; una computadora compatible, un módem Technicolor de 60 Mb y la plataforma Matlab para procesar la señal de saturación exportada.



Figura. N° 2. Conexión ESP8266, y MAX30102
Elaboración propia

B. Adquisición de la señal

Para adquirir la señal de pulso oximetría, utilizamos los siguientes dispositivos y plataformas: el integrado MCU

ESP8266 [9], [18], el sensor MAX30102, la plataforma ThingSpeak [17], una computadora compatible, un módem Technicolor de 60 Mb y la plataforma Matlab. En primer lugar se configuró y programó la adquisición y transmisión de la señal mediante el integrado ESP8266, sobre Arduino IDE, y canalizando mediante ThingSpeak [17], luego instalamos el sensor MAX30102 al módulo EPS8266. Luego procedemos a encender el circuito y colocamos el dedo índice sobre el sensor MAX30102 y esperamos que aparezcan los primeros pulsos y las señales de SPO2, con niveles de porcentaje.

En el plotter de Arduino, se observa inicialmente la señal de pulso oximetría, SPO2, saturada de ruido. El programa grabado en el módulo ESP8266, permite enviar los datos a través del módem y almacenarlos cada 15 segundos en la nube mediante ThingSpeak- Estos datos, posteriormente se pueden exportar para ser procesados a través de programas desarrollados en Matlab u otra plataforma de programación. Así, podemos visualizar el siguiente diagrama de bloques del sistema de adquisición y transmisión de señales de SPO2, para su procesamiento.

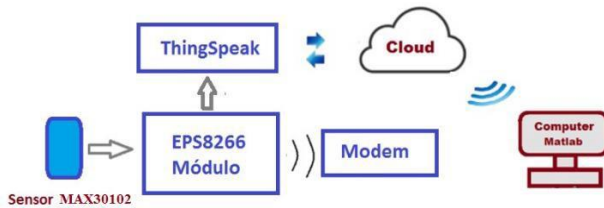


Figura N° 3. Adquisición y almacenamiento de señales de SPO2 vía IoT
Fuente: Elaboración propia

C. Configuración de canal en ThingSpeak

A continuación, vemos la configuración del canal de transmisión de los datos a la nube, a través de ThingSpeak.



Figura N° 4. Configuración del canal en ThingSpeak
Fuente: elaboración propia



Figura N° 5. Señal de SPO2 canalizada por ThingSpeak
Fuente: Elaboración propia

D. Configuración y programación de EPS8266

Configuración y programación de MCU EPS8266, para la adquisición y transmisión de la señal de pulso oximetría, SPO2.

```

SPO2FINAL
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <ThingSpeak.h>
#include "MAX30105.h"
float BPM;
int Valor_Pulso = 520;
long Pulso_Anterior = 0;
unsigned long Canal = 1749499;
const char * API_Key = "011ZUL8MR5YUHNWO";
const char *ssid = "*****";
const char *pass = "*****";
  
```

Figura N° 6. Programación de ESP8266, modem y MAX30102
Fuente: Elaboración propia

E. Procesamiento de la señal de pulso oximetría exportada

En seguida se exporta la señal SPO2, almacena en la nube mediante ThingSpeak. Este dato exportado, puede reducirse a un archivo de texto o a un archivo de Excel, en donde se adjunta un nombre al archivo, como por ejemplo 'senalspo2.txt'. Este archivo, almacenado en la computadora, se carga en Matlab, para su procesamiento.

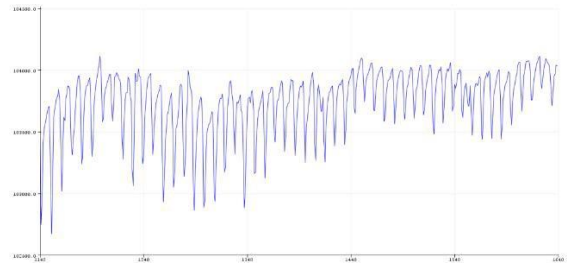


Figure N° 7. Señal de pulso oximetría en plotter de Arduino IDE
Fuente: Elaboración propia

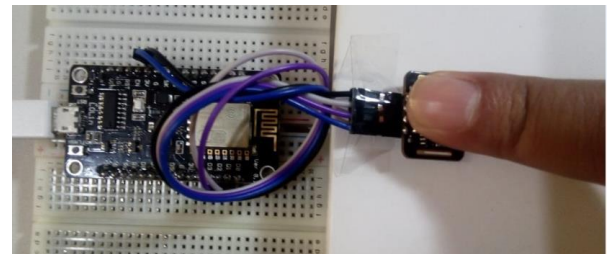


Figure N° 8. Conexión puerto serial ESP8266, MAX30102 y paciente
Elaboración propia

Programa N° 1: Carga de la señal SPO2

```

>>s1 = load('SPO2.txt') % Carga de archivo 'SPO2.txt'
>>Fs1 =402 % Frecuencia de muestreo
>>t =0:Fs1-1 % Vector de tiempo
>>N1 = length(t) % Vector de longitud en función de t
>>deltaF1 = Fs1/N1 % Escalamiento
>>plot(s1) % Graficando
>>grid on %Rejilla
  
```

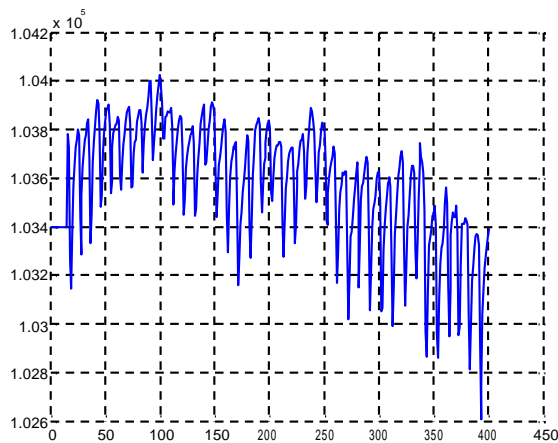



Figura N° 9. Señal SPO2 exportada y cargada por Matlab
Fuente: Elaboración propia

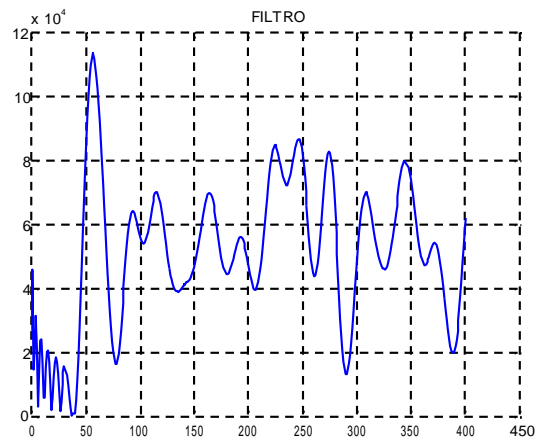


Figura N° 11. Señal SPO2 exportada y filtrada por Matlab
Fuente: Elaboración propia

Programa N° 2: Muestreo de la señal de SPO2

```
>>Fs = 80; %Frecuencia de muestreo
>>F = 3; %Frecuencia fundamental
>>n = 0:Fs-1; %número de muestras
>>Fase = 90; %Desfasaje
>>s1 = xlsread('SPO243.xlsx')+Fase %datos adquiridos
>>stem(n,s1,'b') %Grafico discreto
>>hold on %sobreponer los gráficos
>>grid on
>>plot(n,s1,'r') %Grafico continuo
```

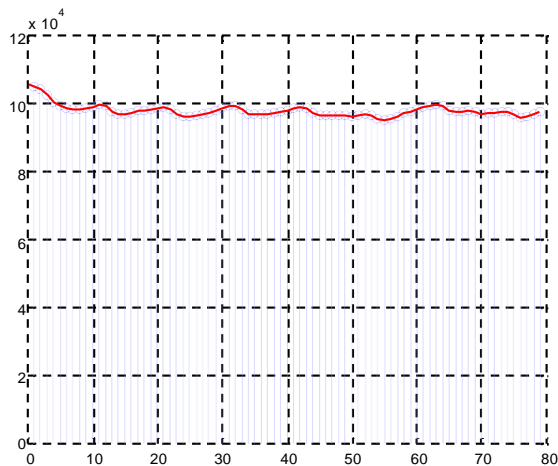


Figura N° 10. Señal SPO2 exportada y muestreada
Fuente: Elaboración propia

Programa N° 3: Filtrado de la señal SPO2

```
>>Fc = 16 %Frecuencia de corte
>>F = Fc/(Fs/2); %Normalización
>>[A,B]= butter(18,F,'high') %Máscara filtro pasa bajos IIR
>>FT=filter(A,B,s1); %Ejecución del filtro
>>plot(t,abs(FT)) %Gráfico
>>title('FILTRO')
>>grid on
```

V. DISCUSIÓN

La última figura de la etapa de procesamiento de la señal, resulta de aplicar un filtro IIR, pasa alto de orden 18. Este filtrado se ha efectuado a la señal de pulso oximetría original enviada remotamente y almacenada en la nube a través de ThingSpeak. Esta señal, posteriormente ha sido exportada a una computadora en formato de texto (pudo haber sido también a una hoja de Excel), en donde esté instalado Matlab para realizar el procesamiento de la señal. La frecuencia de corte aplicada es de $F_c = 14$ Hz, con la finalidad de atenuar armónicos inferiores a esa frecuencia, y las posteriores no sufren ninguna atenuación. Hay que tener en cuenta que la señal de pulso oximetría es de baja frecuencia. Así mismo, es posible también aplicar un filtro FIR1, para reducir los armónicos indeseables que afectan la señal de pulso oximetría.

Asimismo, para el muestreo de la señal, se han tomado entre 80 y 402 muestras de la señal de SOP2 (Figura N° 10). El objetivo del muestreo es convertir la señal analógica a una señal discreta, en el camino que lleva a la digitalización de la señal de SOP2, con el fin de obtener una mejor presentación de la señal y con ello facilitar el diagnóstico clínico de los pacientes.

Es importante, tener en cuenta la influencia de la señal de ruido producida por la corriente alterna de 220 Vac de 60 Hz, que se acopla a la señal de pulso oximetría. Se observa esta influencia cuando la computadora portátil está conectada a la fuente de alimentación, en ese sentido, para reducir el acoplo del ruido es preciso desconectarlo del tomacorriente. En cuanto a la atenuación del ruido que produce esta señal de voltaje alterno, es necesario programar un filtro rechaza banda, con una frecuencia central de aproximadamente 60 Hz, de orden = 50, con el cual se podría mejorar aún más la señal saturada de ruido. En cuanto a la transformada de Fourier, esta herramienta es adecuada para el análisis de frecuencia de la señal, donde podemos observar la presencia de ruido y otros armónicos de la señal, podría haberse aplicado también la Transformada de Wavelet para este propósito.

Es necesario señalar que inicialmente se ensayó con una tarjeta Arduino UNO, un módulo genérico 8266, un sensor genérico EPS8232 y un MAX30102, sin embargo la programación se hizo muy compleja, además la señal presentó mayores niveles de ruido, por lo que se decidió implementar el prototipo con un módulo MCU EPS8266 de mejor desempeño que el genérico cuyo programa se pudo compilar en la plataforma de Arduino IDE. Es importante indicar también, que para el procesamiento de la señal, existen muchos procedimientos alternativos y muchas plataformas de software parte de Matlab para poder implementarlos.

En cuanto a la puesta a tierra también es muy importante, por ello la conexión de los cables de los sensores, se debe de reducir al máximo, o todas las que no participen o estén cerca de nuestro prototipo, con esto conseguimos una mejor calidad de la señal de pulso oximetría.

Otros aspectos a considerar, es la adquisición de la señal de pulso oximetría a través del sensor MAX30102, en donde se observa que el sensor es muy sensible a cambio o movimientos de posición del dedo que le oprime, es decir, ante ligeros movimientos, la señal cambia abruptamente, por el mismo hecho de que no tiene forma de dedal como sí lo son los pulsioxímetros, en donde es posible introducir el dedo índice y este queda muy asegurado y ahí es cuando no se reducen las influencias del ruido que perturba a la señal. Quizá en versiones mejoradas, los sensores de Arduino, evolucionen considerando estos aspectos.

Por otro lado, en cuanto a la plataforma de ThingSpeak, se puede crear una cuenta a través del registro en Mathworks. Sin embargo, para la cuenta gratuita, la canalización de la señal no se da en tiempo real, sino que se refresca cada diez segundos, dándole una forma angular a la señal que progresivamente se va graficando en la plantilla del canal generado. Nos indica, ThingSpeak, que realizando un contrato mediante pago, este tiempo de refresco de la señal se puede reducir a casi un tiempo real, que sería muy práctico para futuras investigaciones de otros investigadores. Por esa razón se observa que la señal que se grafica parece no ajustarse a la señal que se genera en el plotter de Arduino IDE.

En futuras investigaciones será posible aplicar otros tipos de tecnologías asociadas siempre al IoT, así mismo otros posibles programas y transformaciones, con el fin de mejorar el procesamiento de las señales adquiridas.

A. Conclusiones

- Es posible estudiar y analizar el comportamiento de la señal de pulso oximetría de manera remota, mediante el procesamiento de señales con Matlab, la tecnología de IoT y los filtros digitales con FIR, La Transformada de Fourier, entre otros.
- En cuanto al filtro de Respuesta al Impulso infinito (IIR), modelo Butterworth, pasa altos, aplicado en el procesamiento de la señal de pulso oximetría ha resultado muy efectivo, considerando la baja frecuencia de la señal y el elevado orden del mismo.

- De la conclusión anterior se desprende que es posible aplicar también los filtros FIR1, dados su respuesta lineal en fase y su alto grado de estabilidad. Sin embargo para futuras investigaciones, se puede aplicar por ejemplo el filtro de Wavelet.
- Tanto la plataforma de ThingSpeak, el módem del consultorio médico y el módulo de Wi-Fi MCU EPS8266, han permitido transmisión y el almacenamiento de las señales adquiridas por sensores biomédicos en la nube de Internet, para su posterior procesamiento de señales y retransmisión remota para el diagnóstico clínico.
- La plataforma de Matlab, es robusta y amigable. En ese sentido ha permitido cargar la señal de pulso oximetría, almacenada en la nube, mediante ThingSpeak, para luego poder visualizarla, filtrarla, muestrearla y en donde es posible también aplicar por ejemplo la Transformada de Fourier, durante el procesamiento de la señal.
- Mediante Matlab es posible también estructurar y aplicar un filtro rechaza banda, para atenuar la señal alterna de alimentación con una frecuencia de 60 Hz que se acopla a la señal de pulso oximetría.
- Es importante indicar que la plataforma de Arduino IDE al ser compatible con otras plataformas como la MCU MPS8266 y el sensor MAX30102, robustece todos los sistemas de adquisición de datos mediante sensores biomédicos, y resulta sencillo el desarrollo de programas para la transmisión de datos mediante Wi-Fi.
- La señal de pulso oximetría adquirida y procesada, es similar a las señales adquiridas y procesadas por los monitores multiparámetros convencionales, y útil para el diagnóstico clínico. La diferencia radica, en el bajo costo de su implementación y en la posibilidad de su uso en Telemedicina.
- Como conclusión final, podemos agregar que a partir de los datos adquiridos, se cuenta con una base de datos, y a partir de la misma podemos realizar la analítica de datos procurando la toma de decisiones, muy importante en el campo de la ingeniería biomédica en especial.

REFERENCIAS

- [1] González-Barajas, Velandia, Lyma, Ospina (2016). Análisis espectral a través de bancos de filtros aplicado al pre-procesamiento para la umbralización de señales de pulso oximetría. División de Ingenierías, Universidad Santo Tomás, Bogotá-Colombia. <http://www.scielo.org.co/pdf/teclo/v19n37/v19n37a03.pdf>
- [2] Ministerio de Salud, «Situación de salud del Perú – Indicadores Básicos 2014,» 2014. [En línea]. Available: <http://bvs.minsa.gob.pe/local/MINSA/3441.pdf>
- [3] Rodríguez, López, Vega, Flórez (2017). Sistema de monitoreo y control remoto usando IOT para un regulador de presión. Departamento Electrónica y Automatización, Universidad Autónoma de Manizales, Manizales, Colombia <https://www.redalyc.org/pdf/849/84955649011.pdf>
- [4] Bendezu, Q. G., Mari, H., Taype, R. A., Mejía C. R. Effect of rural and marginal urban health service on the physicians' perception of primary health care in Perú. Universidad San Ignacio de Loyola, Universidad Ricardo Palma, Lima, Perú.
- [5] Pallás A. R. Adquisición de compatibilidad de señales entre señales y amplificadores, Universidad Politécnica de Cataluña, ResearchGate.

- [6] American Thoracic Society (2013). Oximetría de pulso. Serie de información al paciente de la ATS ©2011 American Thoracic Society <https://www.thoracic.org/patients/patient-resources/resources/spanish/pulse-oximetry.pdf>
- [7] Naylamp Electronics (2022). NODEMCU V2 ESP8266 WIFI. <https://naylampmechatronics.com/espressif-esp/153-nodemcu-v2-esp8266-wifi.html>.
- [8] Ceja, J., Renteria, R., Ruelas, R., Ochoa, G. (2017). ESP8266 module and its applications in the internet of things. *Electrical Engineering Magazine*.
- [9] Joshi, A., Ravindran, S., and Miller, A. (2011). Heart-rate monitor implementation on the launchpad using msp430g2xx.
- [10] S. S. Winder, *Digital Signal Processing a Practical Guide for Engineers and Scientists*, 2003.
- [11] Benito Condori, Gerson Teodor. (2019). *The Internet of Things IoT*. Universidad Nacional de Educación Enrique Guzmán y Valle.
- [12] Sánchez Gómez, David. 2021. *Design and Implementation of a System for Monitoring the OiT Protocol*. UAM. Department of Electronic Technology and Communications.
- [13] Zhihong Yang, Yingzhao Yue, Yu Yang, Yufeng Peng, Xiaobo Wang and Wenji Liu, (2011.)"Study and application on the architecture and key technologies for IOT," 2011 International Conference on Multimedia Technology.
- [14] Venerdini, A, Sánchez, G., Alvarado, P., Bilbao, I., Ammirati, J.B. (2016). New determinations of P-wave and S-wave velocities for the seismic crust of the Cuyania terrane in the Andean retroarc. *Mexican magazine of geological sciences*. Retrieved in 24, January, 2022, from: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1026-87742016000100059&lng=es&tlng=es.
- [15] Venerdini, A, Sánchez, G., Alvarado, P., Bilbao, I., Ammirati, J.B. (2016). New determinations of P-wave and S-wave velocities for the seismic crust of the Cuyania terrane in the Andean retroarc. *Mexican magazine of geological sciences*. Retrieved in 24, January, 2022, from http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1026-87742016000100059&lng=es&tlng=es.
- [16] Polianytsia, A., Starkova, O. (2017) Survey of the IoT data transmission protocols. 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology.
- [17] <https://thingspeak.com/>
- [18] <http://www.Mathworks.com/>.