

# Proyecto Señales

Vargas Borda Juan Sebastian

Profesor: Milton Jiménez

**Resumen**—En este documento se busca realizar la toma de datos de un paciente (saturación de oxígeno, ritmo cardíaco) para su posterior análisis; para dicho fin se emplea el sensor MAX 30102, y el sensor AD8232, los cuales serán manejados mediante la tarjeta stm32767zi, también se va a manejar una interfaz gráfica por la visualización de los datos por medio del software libre Python.

**Palabras clave** — Python, filtros, frecuencia cardíaca, saturación de oxígeno.

**Abstract** - The following document wants to collect data from a patient (oxygen saturation, heart rate) for subsequent analysis; For this purpose, the MAX30102 and the AD8232 sensor are used, which will be managed through the STM767ZI card, also a graphical interface will be used for data visualization through python which is open source.

**keywords-** Python, filters, Heart rate, SpO2, MAX30102, STM767ZI card, AD8232

## I. Introducción

En este informe encontraremos distintos apartados en los que se hablará del proceso de construcción de un sistema que nos permita medir la frecuencia cardíaca de una persona mediante el sensor AD8232 conectando unos electrodos al cuerpo del paciente, también podremos tomar los datos de oxigenación en la sangre del paciente mediante el sensor Max 30102 en cual funciona mediante i2c, los datos que recopilamos de los sensores haciendo uso del microcontrolador, serán enviados a la interfaz gráfica diseñada en Python en el escritor de código sublime, en el cual se programaron 4 botones para el inicio de la toma de datos del sensor, otro botón para terminar el proceso de la toma de datos, otro botón que nos permita limpiar la gráfica de toma de datos del sensor y un botón que nos permita cerrar o salir de la interfaz gráfica, mediante la librería PySerial realizamos la comunicación del microcontrolador con nuestra interfaz gráfica y mediante la librería.

## A. Marco teórico

Algunos de los conceptos más importantes para la realización del proyecto:

La serie MAX3010x es un sensor óptico, que basa su funcionamiento en el distinto comportamiento que la sangre tiene ante la luz, en función de su grado de saturación de oxígeno.

Para ello, el MAX30102 incorpora dos LED, uno de espectro rojo y otro de infrarrojo. El MAX30102 se pone sobre la piel, por ejemplo, en el dedo o la muñeca. El sensor detecta la luz reflejada, y determina el grado de saturación.

La sangre oxigenada absorbe mayor cantidad de luz infrarroja, mientras que la sangre poco oxigenada absorbe mayor luz roja. En partes del cuerpo donde la piel es suficientemente fina y bajo la que pasan vasos sanguíneos, es posible emplear esta diferencia para determinar el grado de saturación.

El MAX 30102 incorpora dos LED, uno de espectro rojo (660nm) y otro de infrarrojo (880nm), así como fotodiodos para medir la luz reflejada y un ADC de 18 bits y frecuencia de muestreo de 50sps (samples per second) a 3200sps.

El módulo AD8232 permite registrar la actividad eléctrica del corazón, es decir, obtener un electrocardiograma o ECG. El electrocardiograma (EKG o ECG) es una prueba diagnóstica no invasiva que evalúa el ritmo y la función cardíaca a través de un registro de la actividad eléctrica del corazón

El corazón late porque unas señales eléctricas se transmiten por unas vías específicas dentro del corazón, dando lugar al latido cardíaco. Esta actividad eléctrica puede ser detectada por electrodos colocados en la piel, específicamente en la parte anterior del pecho, en los brazos y piernas.

El AD8232 es un integrado con amplificadores de señal y filtros de ruido especialmente calibrados para las señales ECG. El módulo suprime el ruido de 60Hz generado por la electricidad doméstica. La salida del módulo es de tipo analógico, sólo es necesario soldar los

pinos y conectar el módulo a un microcontrolador como Arduino. Dentro del programa debemos realizar la conversión analógica a digital y así podremos observar el ECG en el plotter de Arduino.

## II. COMPETENCIAS POR DESARROLLAR

- Competencia 1. Realizar la medición de ritmo cardíaco mediante el sensor **AD8232** y la medición de oxigenación en sangre mediante el sensor **MAX30102**
- Competencia 2. Realizar una interfaz gráfica en la que podamos observar las magnitudes de las señales de los sensores
- Competencia 3. Realización de un filtro FIR que nos permita realizar el filtrado de la señal del son sensores de ritmo cardíaco y oxigenación en la sangre

## III. Desarrollo de la práctica

Comenzamos utilizando el sensor **MAX30102** En el cual la pulsioximetría óptica es un método no invasivo para determinar el porcentaje de saturación de oxígeno en sangre. Su funcionamiento se basa en que la hemoglobina (Hb) y la hemoglobina saturada (oxihemoglobina, HbO<sub>2</sub>) tienen distintos coeficientes de absorción de luz para distintas longitudes de onda.

### Desarrollo de la interfaz gráfica

Para el desarrollo de la interfaz gráfica de usuario que nos permitiera visualizar los datos de los sensores de frecuencia cardíaca y oximetría Max 30110 y el AD8232, El código del programa se realizó en sublime mediante el lenguaje de programación Python, se utilizaron varias librerías como..., se programaron 4 botones para que el usuario interactúe con el sistema, contamos con un botón de inicio que nos permite dar cabida a la toma de datos, utilizamos un botón de terminar para dar fin a la toma de datos, un botón limpiar para borrar la gráfica y finalmente un botón salir para cerrar la interfaz gráfica.

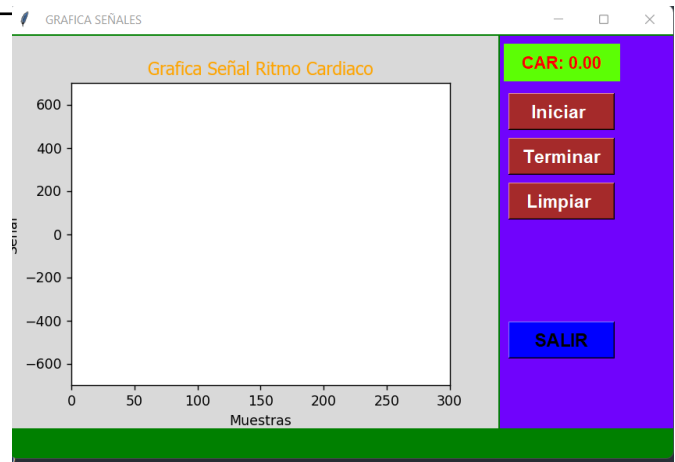


Imagen 1. Diseño interfaz gráfica Python.

### Filtro

En este caso se usó un filtro Fir en donde el kernel usado es de forma cuadrada

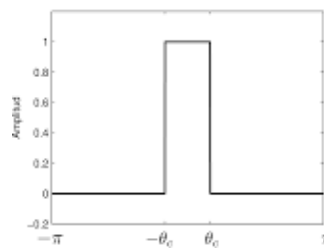


Imagen 2. kernel cuadrado

donde en ancho o ventana que va a tener este filtro es de 100; vale la pena tener en cuenta que este filtro funciona como un promedio entre los datos que se tienen a la entrada de la comunicación, por lo que es útil para eliminar el ruido

### Código:

Para el desarrollo de la interfaz gráfica en Python se utilizaron las siguientes librerías importamos serial, time, collections, matplotlib.pyplot as plt, matplotlib.animation, FigureCanvasTkAgg, Thread, Tk. Posteriormente se declaran las variables que vamos a utilizar en el programa tales como el tamaño de la amplitud de la gráfica, el número de muestras que vamos a recopilar.

Luego se realiza la lectura del puerto serial, definiendo el puerto COM y la velocidad de comunicación, para esta parte utilizamos la librería PySerial que nos permite utilizar el puerto serial del computador para transmitir datos de los sensores a la computadora y de esta forma ser visualizados en la interfaz gráfica de Python diseñada.

```
C:\Users\varga\OneDrive\Escritorio\python\grafica_GUI.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

comunicacion_serial.py x main.py x Filtro_FIR.py

4
5
6 import serial,time,collections
7 import matplotlib.pyplot as plt
8 import matplotlib.animation as animacion
9 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
10 from threading import Thread
11 from tkinter import Tk, Frame, StringVar, Label,Button,Entry
12
13
14 isReceiving= False
15 isRun = True
16 datos = 0.0 # lavel para indicar el voltaje
17 muestraD = 300
18 data = collections.deque([0]*muestraD, maxlen=muestraD)
19 xmin = 0
20 xmax = muestraD
21 ymin = -700 # amplitud en el eje negativo
22 ymax = 700 # amplitud en el eje positivo
23
24 #"/dev/ttyUSB0"
25
26 try:
27     arduino = serial.Serial("COM3", 9600 , timeout=1)
28
29 except:
30     print("Error de coneccion con el puerto")
31
32
33 def Iniciar(): #Parametros de boton de Inicio
34     global datos
35     global isReceiving
36     global isRun
37     isReceiving = True
38     isRun = True
```

Imagen 3. Código Python

Pasamos a ver el código base para dividir las dos señales de los sensores desde el microcontrolador, generamos la comunicación serial a 9600 baudios y en el while realizamos la lectura de los datos enviados por los sensores mediante el ADC del microcontrolador, posteriormente se envían estos datos separados por dos puntos (:) en simultáneo y cuando este llega a Python dividimos los datos y los almacenamos en una nueva variable por separado. (Ver Figura 4)

```
datos = arduino.readline().decode('utf-8')
pos=datos.index(":")
elec=float(datos[:pos])#Leemos el valor ante
oxi=float(datos[pos+1:])#Leemos el valor des
isReceive = True
```

Imagen 4. Código Python

```
FrecuenciaAD8232$
void setup(){
    // Inicializar la comunicacion en serie:
    Serial.begin(9600);
    pinMode(10, INPUT); // Configuración para la detección LO +
    pinMode(11, INPUT); // Configuración para la detección LO -
}
void loop() {
    int elec = analogRead(A0);
    int oxi = analogRead(A5);
    if((digitalRead(10) == 1) || (digitalRead(11) == 1)){
        Serial.println("!");
    }
    else{
        // Imprimir la lectura del puerto A0
        Serial.println(String(elec)+"."+String(oxi));
        //Serial.println(analogRead(A0));
    }
}
```

Imagen 5. código comunicación de prueba.

```
#include <Wire.h>
#include "MAX30105.h"
#include "spo2_algorithm.h"

MAX30105 particleSensor;

#define MAX_BRIGHTNESS 255

#if defined(__AVR_ATmega328P__) || defined(__AVR_ATmega168__)
//Arduino Uno doesn't have enough SRAM to store 100 samples of IR led data
//To solve this problem, 16-bit MSB of the sampled data will be truncated.
uint16_t irBuffer[100]; //infrared LED sensor data
uint16_t redBuffer[100]; //red LED sensor data
#else
uint32_t irBuffer[100]; //infrared LED sensor data
uint32_t redBuffer[100]; //red LED sensor data
#endif

int32_t bufferLength; //data length
```

Imagen 6. Código comunicación I2C



Imagen 7: Conexiones físicas de los sensores, Max30102 y AD8232.

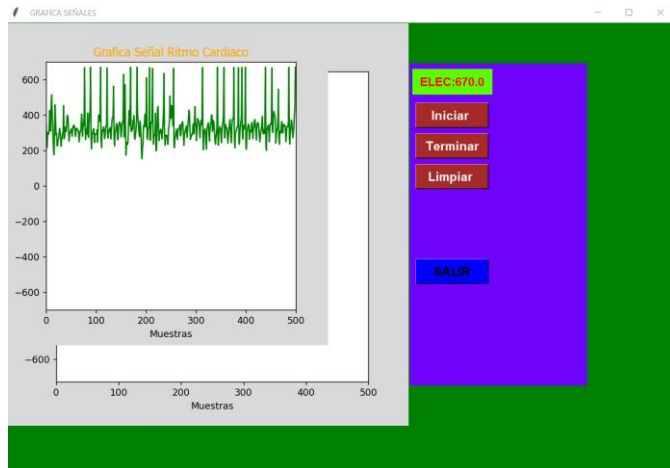


Imagen 8: Lectura del electrocardiograma en la interfaz gráfica en Python.

Imagen 10: Detección de movimiento en la persona al estar ansioso

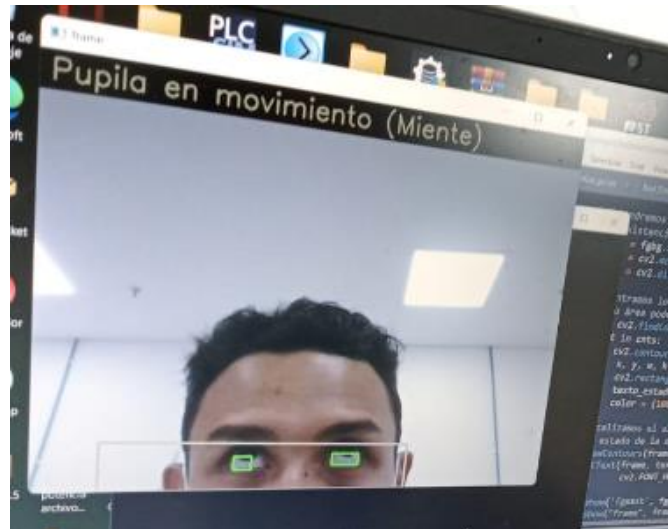


Imagen 11: Detección de movimiento en las pupilas, otro parámetro que se tiene en cuenta a la hora de mentir.



Imagen 9: Detección inmovilidad en la persona al estar relajado

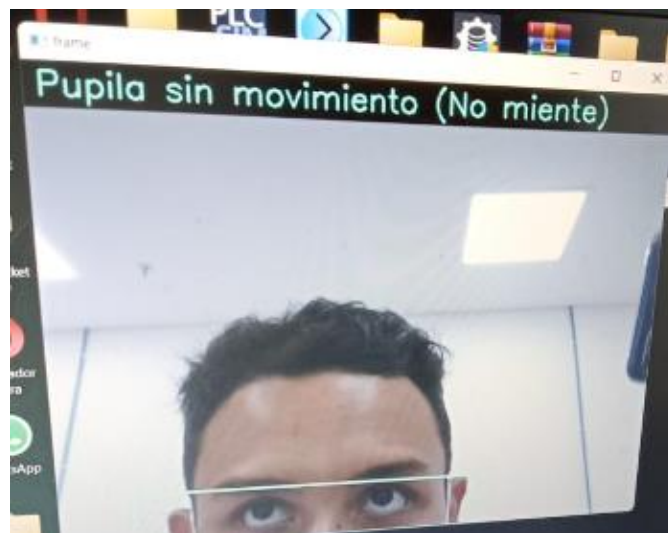


Imagen 12: Sujeto de prueba que no genera movimiento en los ojos, por lo tanto, no estaría mintiendo.



### Actividad electrodérmica

La piel tiene un comportamiento eléctrico denominado EDA que corresponde a el cambio de la conductancia de la piel entre 2 puntos, este comportamiento va ligado a las emociones presentes en el individuo por lo cual resulta útil al momento de querer identificar algún comportamiento en específico como el de mentir por ejemplo.



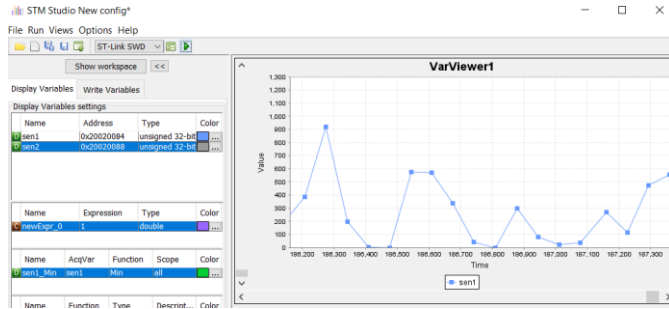


Imagen 13: visualización actividad electrotérmica a través de STM32F767zi

Para el análisis de esta variable se implementaron unos electrodos reutilizables los cuales se diseñaron en SolidWorks y fueron impresos en tpu para que tomara la forma del dedo.



Imagen 14: Sensor actividad electrodérmica.

#### IV. Conclusiones

- Las personas tienen ciertos patrones de comportamiento que las hacen predecibles y que permiten concluir ciertas acciones como la de mentir; además se tienen ciertas variables fisiológicas que no son visibles a simple vista y se deben usar sensores especiales para estirarlas estas son la SpO2, BPM, y la actividad electromiográfica EDA.
- Mediante el uso de la librería PySerial () podemos acceder al puerto serial de la computadora para que nos transmita los datos a Python desde el microcontrolador
- Se diseñó la interfaz gráfica en Python a punta de código y haciendo uso de las librerías que nos permiten plotear una señal como lo es la librería matplotlib.pyplot

- Los filtros son importantes para el procesamiento de señales ya que permiten tener una aproximación mayor a la señal buscada; esto con el fin de poder procesar de mejor manera ya que una señal muy ruidosa o con muchas fluctuaciones resulta deficiente al momento de querer analizarla.
- La programación en Python es bastante entendible y se puede utilizar una gran cantidad de librerías que facilitan la programación de la interfaz gráfica además del uso de hilos para la visualización de los datos enviados desde el microcontrolador.
- Para la lectura del sensor max30102, al hacer uso de la STM para su funcionamiento encontramos que se podía utilizar las librerías de Arduino para la lectura de datos del sensor desde Keil, pero en vez de definir un puerto analógico del Arduino se define el pin del microcontrolador STM.
- El AD8232 es un integrado con amplificadores de señal y filtros de ruido especialmente calibrados para las señales ECG. El módulo suprime el ruido de 60Hz generado por la electricidad doméstica.

#### REFERENCIAS

- [1] Hasan, D. e Ismaeel, A. (2020). Diseño de un sistema de atención médica de monitoreo de ECG basado en la aplicación Blynk de Internet de las cosas. *Revista de tendencias en ciencia aplicada y tecnología*, 1 (3), 106-111.
- [2] Prasad, AS y Kavanashree, N. (julio de 2019). Sistema de monitorización de ECG mediante sensor AD8232. En *2019 Conferencia Internacional sobre Sistemas de Comunicación y Electrónica (ICCES)* (págs. 976-980). IEEE.
- [3] Kanani, P. y Padole, M. (abril de 2018). Reconocimiento de anomalías de ECG en tiempo real utilizando Arduino, AD8232 y Java. En *Conferencia internacional sobre avances en informática y ciencias de datos* (págs. 54-64). Springer, Singapur.
- [4] Flores, J. L. E. (2021). *Sistema de monitoreo cardiaco portátil de una terminación para atención primaria o control médico remoto de pacientes con anomalías cardíacas* (Doctoral dissertation, UNIVERSIDAD DE CUENCA).