

Angie Marisela García Ríos
Estefany Cuervo Suárez
Juan Felipe Arroyave Zapata
Electrónica digital II

Índice

1. Identificación del proyecto	2
2. Resumen	2
2.1. Objetivos Específicos	2
3. Descripción del hardware	2
4. Descripción del software	3
5. Estructura del código	4
6. Explicación de funciones principales	4
7. Funcion Principal de Ejecucion	5
8. Funciones de retroalimentación	5
9. Retroalimentacion por consola	6
10.Filtros Digitales Implementados	6
10.1. Filtro de Promedio Móvil	6
10.2. Filtro de Mediana	6
10.3. Filtro Exponencial (IIR)	7
11.Comparativa de Filtros	7
12.Recomendaciones de Uso	7
13.Parámetros de Implementación	7

1. Identificación del proyecto

- **Nombre del proyecto:** Informe de Proyecto - Arreglo para Lectura de Señal Cardíaca con AD8232
- **Integrantes del equipo:** Angie Marisela García Ríos, Estefany Cuervo Suárez, Juan Felipe Arroyave Zapata

2. Resumen

Este proyecto consiste en el diseño e implementación de un sistema de adquisición y procesamiento de señales cardíacas. Utilizando el sensor AD8232, se captura la señal analógica, la cual es digitalizada mediante el conversor integrado en el microcontrolador ESP32. Posteriormente, se aplican diversos métodos de filtrado digital para eliminar ruido y obtener una señal limpia. La señal procesada se transmite en tiempo real a través del puerto serial hacia un dispositivo externo, donde puede ser visualizada y analizada con mayor detalle.

2.1. Objetivos Específicos

- Diseñar un sistema de adquisición de señales cardíacas con resolución de 12 bits
- Implementar 4 algoritmos de filtrado digital para mejorar la relación señal-ruido
- Desarrollar una interfaz de usuario intuitiva para control en tiempo real
- Garantizar una frecuencia de muestreo estable entre 10 a 500 Hz

3. Descripción del hardware

- **ESP32** (modelo con ADC y pines GPIO)
- **Sensor AD8232**
- **Led** conectado al pin 2
- **Resistencia** conectadas al led

Diagrama de conexión:

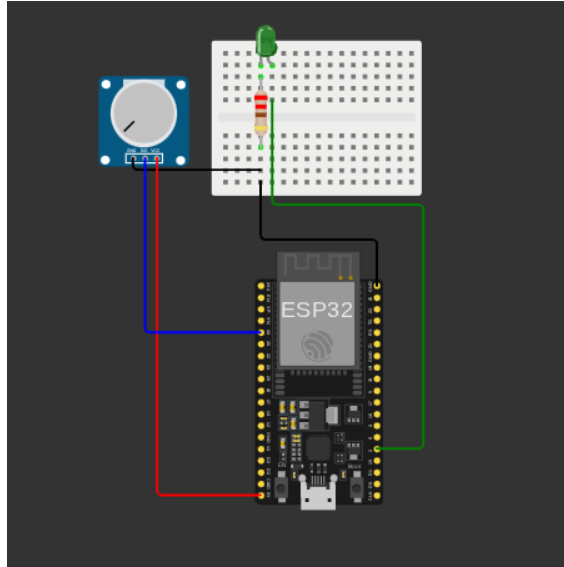


Diagrama de Conexión

4. Descripción del software

- **Lenguaje usado:** MicroPython
- **Librerías:** machine, time, timer, ADC
- **IDE:** Thonny y Wokwi

Flujo general del programa:

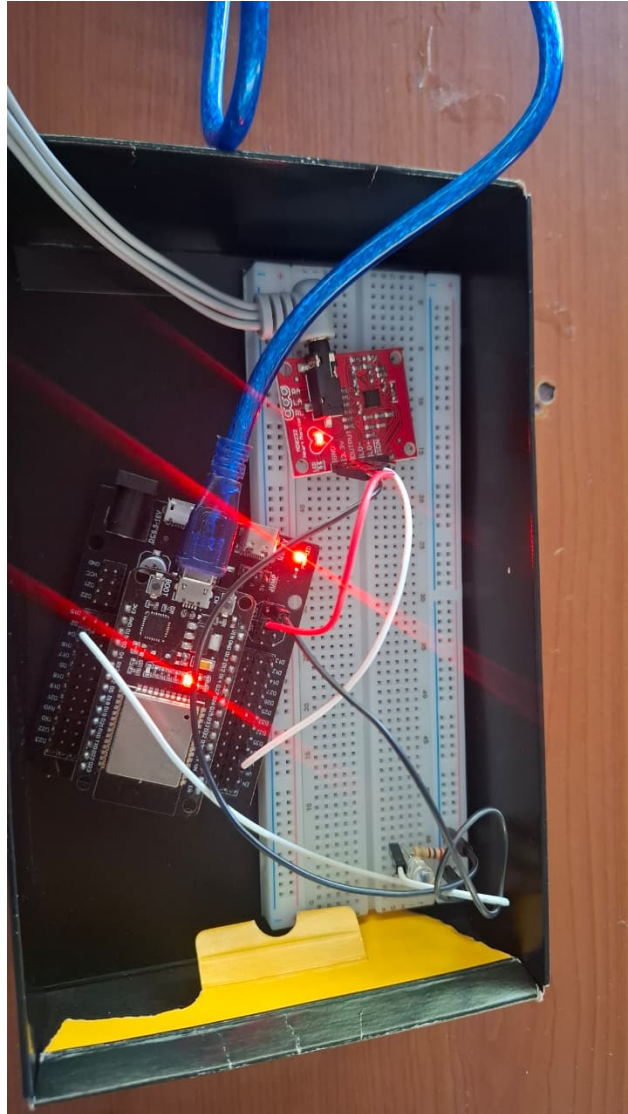
1. Configuración de pines y variables
2. Configuración primaria para filtros y timers
3. Código para prueba de diagnóstico del sensor y evaluación de resultado del diagnóstico
4. Establecimiento de las funciones para cada filtro digital
5. Configuración de frecuencia de muestreo y de la función para el guardado de datos
6. Función para la visualización de datos en el Serial Plotter y el uso de los filtros
7. Creación del menú principal y configuración del mismo para su uso

5. Estructura del código

- Archivo principal: `main.py`
- No se usan módulos adicionales.
- Código basado en menú interactivo con adquisición en segundo plano y múltiples modos de operación para procesamiento de señales ECG.

6. Explicación de funciones principales

- `diagnostico_sensor()`: Realiza 10 lecturas de prueba del sensor y analiza los valores para detectar problemas y proporcionar recomendaciones.
- `configurar_frecuencia(frecuencia)`: Cambia la frecuencia de muestreo desde 10 a 200Hz, reconfigura el timer y ajusta el periodo de muestreo.
- `configurar_filtros()`: Menu para activar o desactivar los filtros o activar funciones predefinidas de los filtros.
- `muestrear_ecg(timer)`: Lee el valor actual del sensor, mantiene el buffer de las ultimas 30 muestras y aplica los filtros configurados.
- `aplicar_filtros(valor_crudo)`: Esta función es fundamental en el flujo de procesamiento de señales y se llama desde `muestrear-ecg()`, pero no estaba incluida en tu lista original.
- `mostrar_serial_plotter()`: Exporta los datos en formato CSV para plotting y muestra la señal segun los filtros.
- `guardar_datos_archivo()`: Guarda los archivos en un archivo txt con linea de tiempo y formato CSV con todas las señales de filtros
- `ver_estadisticas()`: Muestra los conteos de muestras por cada filtro y da informacion de frecuencia de muestreo actual, el rango y la variabilidad de los valores.
- `mostrar_menu()`: Nos muestra la interfaz de usuario y presenta las opciones disponibles.



Evidencia visual del montaje en protoboard

7. Funcion Principal de Ejecucion

- Bucle Principal con manejo robusto de excepciones
- Control de Flujo entre diferentes modos de operacion
- Gestion Segura de Interrupciones
- Limpieza de recursos al salir

8. Funciones de retroalimentación

- Led Integrado que parpadea con al variacion de señal ECG.

- Serial Plotter que muestra graficos en tiempo real de todas las señales
- Estadísticas que muestra los conteos y rangos de valores actuales.

9. Retroalimentación por consola

- Diagnostico Detallado del estado del sensor.
- Confirmación de configuraciones aplicadas
- Mensajes de Error descriptivos
- Menu Interactivo que ofrece navegación entre funciones.

10. Filtros Digitales Implementados

10.1. Filtro de Promedio Móvil

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k] \quad (1)$$

Características:

- Tipo: FIR (Respuesta Finita al Impulso)
- Ventana: 5 puntos
- Retardo: 25 ms a 100 Hz
- Efectividad: Excelente para ruido de alta frecuencia

10.2. Filtro de Mediana

$$y[n] = \text{median}(x[n], x[n-1], x[n-2], x[n-3], x[n-4]) \quad (2)$$

Características:

- Tipo: No lineal
- Ventana: 5 puntos
- Retardo: 20 ms a 100 Hz
- Efectividad: Elimina picos y ruido impulsivo

10.3. Filtro Exponencial (IIR)

$$y[n] = \alpha \cdot x[n] + (1 - \alpha) \cdot y[n - 1] \quad (3)$$

Características:

- Tipo: IIR (Respuesta Infinita al Impulso)
- Parámetro: $\alpha = 0,3$
- Retardo: ≈ 15 ms
- Efectividad: Balance entre suavizado y respuesta rápida

11. Comparativa de Filtros

Cuadro 1: Comparativa de Características de Filtros

Característica	Crudo	Promedio	Mediana	Exponencial
Tipo	–	FIR	No lineal	IIR
Retardo (ms)	0	25	20	15
Uso de CPU	Muy bajo	Medio	Alto	Muy bajo
Memoria	1 punto	5 puntos	5 puntos	1 punto

12. Recomendaciones de Uso

- **Señal cruda:** Ideal para diagnóstico inicial y análisis de ruido
- **Promedio móvil:** Recomendado para ruido de alta frecuencia (60 Hz)
- **Mediana:** Óptimo para artefactos por movimiento y picos abruptos
- **Exponencial:** Balance ideal para monitoreo continuo en tiempo real

13. Parámetros de Implementación

- Frecuencia de muestreo: 250 Hz (configurable 10-500 Hz)
- Tamaño de buffer: 500 muestras por filtro
- α exponencial: 0.3 (balanceado para señales cardíacas)
- Ventana de filtros: 5 muestras (50 ms a 100 Hz)