**Informe De Desarrollo Parcial 1**

**Informática II**

*Cristian Murillo*

Informatica II  
2024-2

A. análisis del problema

Para el problema planteado implica la creación de un algoritmo para identificar una señal dada por un generador de señales en un Arduino uno, para este desafío las principales cosas a tener en cuenta para una correcta solución del ejercicio es la limitación en la memoria, almacenar los datos, el correcto muestreo y criterio para evaluar los datos obtenidos e identificar de manera adecuada la señal enviada.

Para la solución se propuso el almacenar los datos leídos en un arreglo dinámico, y por medio de analizar los cambios de un elemento al siguiente del arreglo encontrar la señal enviada, para la amplitud se usa el calcular la diferencia entre el pico y el valle y para la frecuencia se cuentan los cruces por el valor medio y se calcula con la diferencia entre el inicio del programa y el tiempo hasta que paso el valor medio.

B. Esquema de tareas realizadas.

1. Se inicializa el Arduino.

2. Se comienza la captura de datos.

3. Se almacenan los datos en un arreglo dinámico.

4. Se detiene la captura de datos.

5. Se calculan la amplitud y la frecuencia.

6. Se identifica el tipo de señal.

7. Se muestran los resultados en la pantalla LCD.C. Algoritmos implementados.

Se implementaron los siguientes algoritmos para la solución del problema.

HandleButtons: que se encarga del manejo de botones y del estado del botón de inicio y final de toma de datos.

StartDataCollection: se encarga del inicio de la toma de datos cuando se pulsa el botón correspondiente.

StopDataCollection: detiene la toma de datos cuando se pula el botón correspondiente

CollectData: captura los datos con el delay correspondiente entre muestra y muestra.

DetectZeroCrossing: detecta los cruces de la señal cuando pasa por el valor medio de la amplitud.

StoreData: Almacena los datos recolectados en un arreglo usando memoria dinámica.

CalculateAmplitude: calcula y retorna la amplitud de la onda.

CalculateFrequency: Calcula y retorna la frecuencia de una onda.

CountPikesAndValleys: cuenta la cantidad de picos y valles del arreglo de la señal

IdentifySignalType: Identifica y cuenta los cambios en los valores de la señal.

IsSineWave: función booleana que retorna e identifica si una señal es sinusoidal

IsSquareWave: función booleana que retorna e identifica si una señal es cuadrada

IsTriangularWave: función booleana que retorna e identifica si una señal es triangula.

DisplayResults: muestra en la pantalla lcd los datos obtenidos.

ResetState: reinicia los datos para poder tomar unos nuevos.

D. Problemas de desarrollo que afrontó.

Durante el desarrollo identifique 3 principales problemas en la realización y desarrollo del desafío 1 que son:

1. **Problema con la memoria**: Debido a las limitaciones de memoria en la arquitectura del Arduino, la capacidad para reservar suficiente memoria para el arreglo puede verse limitada, lo que podría impedir la identificación correcta de la señal si se reciben demasiados datos.

2. **Problema con la identificación de la señal sinusoidal**: La lógica actual para identificar señales sinusoidales puede confundirlas con otras señales. Además, el rango de frecuencias debe tener en cuenta las limitaciones del muestreo, no superando los 50 Hz para obtener resultados precisos.

3. **Errores por rebote en la pulsación del botón**: Puede haber problemas al registrar las pulsaciones del botón debido a rebotes, lo que puede resultar en la detección de múltiples pulsaciones y en la detención prematura de la captura de datos.