Análisis y desarrollo de la implementación del desafio 1.

Informática ll

Jose Andrés Henao Alzate.
Departamento de ingeniería electrónica
Universidad de Antioquia
Medellín, Colombia
andres.henao9@udea.edu.co

Juan Esteban Franco Aranzalez.
Departamento de ingeniería electrónica
Universidad de Antioquia
Medellín, Colombia
juan.francoa@udea.edu.co

***Resumen*—El proceso de resolución de problemas en programación comienza con la identificación y comprensión detallada del problema, seguida por la fase de implementación y concluyendo con las pruebas del código desarrollado. Este documento presenta el análisis y desarrollo de la solución correspondiente al Desafío 1 del curso de Informática II. Se explican los códigos implementados, el razonamiento detrás de su formulación y los desafíos encontrados durante la etapa de implementación, ofreciendo una visión general de las soluciones propuestas y los problemas enfrentados durante la etapa de ejecución.**

***Index Terms*—Lenguaje C++, Razonamiento, Eficiencia, Analisis, Voltaje, Frecuencia, Señales, coeficiente de correlación.**

I. Introducción

Partiendo del análisis y segemtación del problema, se procedió con la busquedad de algoritmos que solucionaran cada uno de esto subproblemas. Los subproblemas identificados para la solucion general del problema son:

* Obtener la frecuencia de la señal analógica, la cual esta ingresando por el puerto A0. *Obterner el voltaje de la señal analógica que ingresa por dicho puerto. * Generar una función, la cual a partir del muestreo de la señal analoga, identifique por medio de la correlación con señales bases generadas de forma dinámica, el tipo de señal.

II. Análisis del problema y consideraciones para la alternativa de solución propuesta..

Lo primero que se observó del problema, fue el hecho de que debido a su complejidad, lo mas comveniente era generar una división del problema global, en problemas mas simples los cuales llevaran a la solución global. Una de las principales consideraciones que se tuvieron para la implementación de el sistema fue las limitaciones de procesamiento del arduino y que se estaba trabajando en una plataforma de simulacion, también las limitancias que podrían presentarse en alta frecuencia debido a cuestiones físicas como lo es la frecuencia de muestreo frente a la frecuencia de respuesta del arduino.

III. Esquema que describe las tarea sy funcionalidad del programa.

El siguiente esquema muestra la funcionalidad el código y las tareas y técnicas empleadas para su operación.

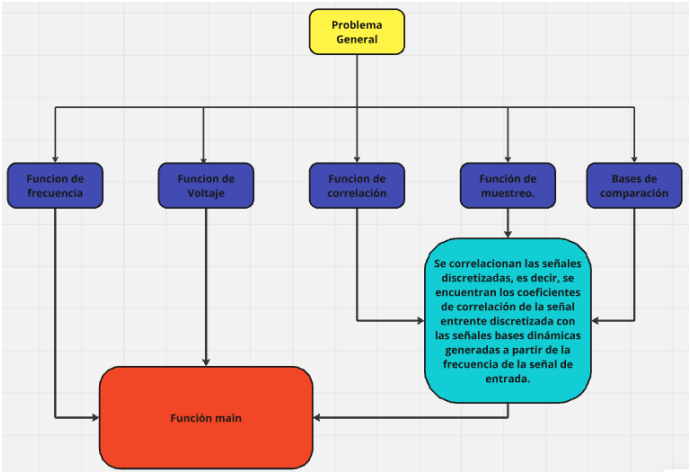

Figura 1. Esquema gráfico de la implementación del desafio 1.

IV. Algoritmos usados en la implementación.

IV-A. Función `Amplitud()`

La función `Amplitud()` mide la **amplitud de la señal analógica** en el pin de entrada, tomando 100 muestras de la señal.

- Inicializa las variables `maxVal` y `minVal` para registrar los valores máximos y mínimos.
- Dentro de un ciclo `for`, realiza 100 lecturas de la señal usando `analogRead(pinEntrada)`.
- Comparativamente, actualiza `maxVal` si la lectura es mayor y `minVal` si es menor.
- Después de las 100 muestras, calcula la amplitud en voltios mediante la fórmula:

$$\text{Amplitud} = (\text{maxVal} - \text{minVal}) \times \frac{5{,}0}{1023{,}0}$$

En el `loop()`, la función `Amplitud()` es llamada cuando la variable `recopilacion` está activada. Esto se controla mediante los botones conectados a los pines 6 y 7. El valor de la amplitud se muestra en el LCD y se guarda para más análisis.

IV-B. Función `base_seno(float frecuencia, int muestras)`

Esta función genera una **onda senoidal teórica** con la frecuencia y número de muestras especificados:

- Calcula el intervalo de tiempo entre las muestras basado en la frecuencia.
- Usa la fórmula del seno, ajustada para que la onda comience en su valor máximo (`sin(angulo)`).
- Almacena los valores generados en un arreglo dinámico que será comparado con las muestras de la señal.

En el `loop()`, la función `base_seno()` es llamada después de capturar las muestras de la señal real mediante la función `Muestras()`. El objetivo es generar una onda senoidal de referencia que será utilizada para calcular la correlación con la señal medida y determinar si la señal es de tipo senoidal.

IV-C. Función `Frecuencia()`

La función `Frecuencia()` calcula la **frecuencia de la señal** detectando cuántas veces cruza el umbral (promedio entre el valor máximo y mínimo) en 2 segundos.

- Inicializa `valorMax` y `valorMin` para registrar los extremos de la señal.
- Monitorea el número de veces que la señal cruza el umbral (`umbral = (valorMax + valorMin) / 2`).
- El número de cruces se utiliza para calcular la frecuencia:

$$\text{Frecuencia} = \frac{\text{Cruces}}{2}$$

- La función se repite en un ciclo mientras se tomen datos durante 2 segundos.

En el `loop()`, `Frecuencia()` es llamada cuando se inicia la recopilación de datos, y su resultado es utilizado para ajustar el número de muestras a tomar y para mostrar la frecuencia en el LCD.

V. Proeblmeas encontrados y evolución de la implementación.

Al inicio, la implementación del circuito y las funciones encargadas de encontrar la frecuencia y amplitud se realizó con relativa facilidad, gracias a los conocimientos previos en el área de procesamiento de señales y circuitos eléctricos. Sin embargo, el principal desafío surgió durante la identificación de la señal. A pesar de calcular los coeficientes de correlación para distinguir entre las señales senoidal, triangular y cuadrada, se encontraron resultados similares en varias ocasiones entre las señales senoidales y triangulares, así como también con la señal cuadrada.

Un problema adicional fue que el Arduino no pudo seguir la frecuencia de muestreo de las señales usadas como bases (seno, triangular y cuadrada) a frecuencias altas. La distorsión de las señales resultante afectó la precisión de los coeficientes de correlación, lo que complicó la identificación precisa de las señales. Esta distorsión podría haber sido exacerbada por posibles retrasos en el simulador de Tinkercad o problemas en el compilador, aunque esta última es solo una especulación.

En términos generales, todo cuadró y dió buenos resultados inclusive en altas frecuencias (alrededor de 160HZ) a excepción del coeficiente de correlación y la identificación de las señales.

VI. Conclusiones.

- Desafíos en la Identificación de la Señal: El principal desafío fue la identificación precisa de la señal. Los coeficientes de correlación, utilizados para distinguir entre señales senoidales, triangulares y cuadradas, no siempre produjeron resultados claros. Las similitudes entre las señales senoidales y triangulares, y también con la señal cuadrada, indicaron que el método de correlación no fue completamente efectivo en esta aplicación o no se implmentó de la mejor manera.
- Mirando en términos de cálculos de amplitud y frecuencia, los resultados fueron bastante idóneos según las expectativas puestas sobre los resultados. Por ende se considera un logro el resultado experimental de estos items.