**Desafío 1**

**José Vicente Zabaleta Montiel**

**Juan Pablo Arrubla Arenas**

**Informática II**

**Universidad de Antioquia**

**Facultad de Ingeniería Electrónica**

Nos enfrentamos a la necesidad de desarrollar un sistema que permita la adquisición y visualización de las características de una señal analógica. Este proyecto tiene como principal reto identificar y procesar señales analógicas de manera eficiente, midiendo sus características principales como frecuencia, amplitud y forma de onda, haciendo la implementación usando punteros, arreglos y memoria dinámica. A partir de estas mediciones, se debe mostrar la información en una pantalla LCD, lo cual nos exige no solo implementar algoritmos de medición precisos, sino también asegurar una interacción adecuada entre los componentes físicos y el código.

**Análisis del problema:** El problema se plantea en un contexto donde el desafío requiere un sistema de monitoreo en tiempo real de señales analógicas. El principal desafío aquí es la necesidad de adquirir la señal de entrada de manera continua, procesarla y luego visualizar sus características en una pantalla, permitiendo además la posibilidad de suspender la adquisición mientras se solicita la información procesada. En términos de complejidad, esto implica manejar de forma adecuada tanto la parte física (Arduino, pulsadores, LCD, y generador de señales) como la lógica (algoritmos para detectar las características de la señal).

Un punto clave de este proyecto es la identificación de la forma de la onda de la señal. El generador de señales en Tinkercad nos permite generar señales de diversas formas (senoidal, triangular, cuadrada, etc.). Esto significa que, además de medir parámetros simples como frecuencia y amplitud, nuestro algoritmo debe tener la capacidad de reconocer patrones más complejos en las señales, lo que aumenta significativamente la dificultad del problema.

Consideraciones para la alternativa de solución: En cuanto a las alternativas de solución, optamos por una arquitectura que permita la adquisición en tiempo real de la señal y su procesamiento en intervalos controlados por un pulsador. Esto nos permite cumplir con el requisito de poder pausar la adquisición mientras se procesa la información y luego reanudarla una vez finalizado el proceso. Además, decidimos utilizar una pantalla LCD para mostrar los valores medidos, ya que esta ofrece una interfaz simple y eficiente para visualizar los resultados sin necesidad de una conexión a un dispositivo externo.

Finalmente, al diseñar el algoritmo para medir la forma de onda, consideramos un enfoque de reconocimiento basado en comparación de patrones. Esto nos permite identificar si la señal corresponde a una onda senoidal, cuadrada, triangular o si es desconocida, lo cual cubre todas las posibilidades que ofrece el generador de señales de Tinkercad.

Esta solución, aunque viable, implica desafíos en cuanto al manejo eficiente de los tiempos de adquisición y procesamiento, así como en la precisión de los algoritmos de medición y reconocimiento.

Los algoritmos implementados están diseñados para captar y procesar la señal de entrada que proporciona el generador de señales de la plataforma Tinkercad. El enfoque principal se centró en la identificación de las características de la señal, como su amplitud, frecuencia y forma de onda, lo que permite identificar si la señal es senoidal, triangular, cuadrada o desconocida.

Adquisición de la señal: La señal es adquirida a través de un generador de señales conectado al Arduino, utilizando la función analogRead() para obtener los valores de la señal analógica. Esta parte del código se encarga de tomar lecturas periódicas de la señal, almacenarlas en un arreglo dinámico para análisis posterior.

Detección de la pendiente: Este algoritmo realiza el cálculo de la pendiente entre dos puntos consecutivos de la señal. La pendiente nos permite determinar si la forma de la señal tiene patrones repetitivos o bruscos. El valor de la pendiente se compara con un valor de referencia inicial, lo que nos permite identificar si estamos ante una señal triangular (pendientes constantes) o senoidal.

Detección de cambio brusco: En el caso de las señales cuadradas, se emplea un mecanismo de detección de cambios bruscos, donde se observa un salto significativo en los valores de la señal(El salto es detectado cuando la diferencia de 2 valores consecutivos es mayor a 100). Si se detectan varios de estos saltos de manera consecutiva, el algoritmo determina que la señal es cuadrada.

Medición de amplitud y frecuencia: La amplitud se calcula tomando el valor máximo de la señal dentro del conjunto de lecturas. La frecuencia se obtiene mediante el cálculo del número de cruces por cero, para identificar cuántas veces la señal cambia de signo dentro de un intervalo de tiempo dado. Estos valores se presentan al usuario en la pantalla LCD para una visualización rápida

También se utilizan funciones para asignar y liberar la memoria dinámica usada en el ciclo loop de nuestro Arduino, primero en el setup se inicia la memoria asignada 1 vez, permitiendo almacenar los valores del generador en la lista dinámica y también permitiendo el uso de apuntadores, que luego serán liberados y re apuntados nullptr para no ocasionar fugas en memoria.

Problemas del desarrollo

El gran problema principal y el causante de que algunas funcionalidades del código hayan quedado afectadas es la gran impresicion del programa para capturar los datos que se daban por el generador de señales, el que los datos no se guardaran contiguamente en la lista fue un gran dolor de cabeza para nosotros, ya que tuvimos que pensar en ideas que debían tener no debían tener una solución precisa sino que se tuvo que asignar márgenes de error, pero en algún punto terminaban fallando ya que al aumentar la frecuencia del generador estos datos eran muchísimo mas discontinuos, esto debido a que el Arduino no tenía la capacidad de procesar el volumen de datos que llegaba en tan corto tiempo.

Debido al problema ya mencionado , la diferenciación entre señales analógicas y señales triangulares han resultado en problemas muy difíciles de corregir, ya que son calculadas mediante la adquisición de la pendiente que es tomada por el dato evaluado menos el dato anterior, y dependía mucho de la frecuencia a la cual se tomaban los datos, provocando así que en frecuencias muy bajas las 2 ondas sean tomadas como analógicas y a frecuencias altas las ondas sean tomadas como triangulares.

El no haber programado nunca en Arduino también fue un desafio que aunque no tan grande, nos llevó a un importante componente investigativo que rápidamente fue analizado y solucionado.

Evolución del código

El desarrollo del código fue una evolución constante en la que se fueron ajustando bastantes aspectos tanto del manejo de la señal como de la optimización de la memoria y la precisión en la detección. Inicialmente, nos enfocamos en adquirir y procesar las lecturas de la señal en intervalos de tiempo definidos, lo cual resultó ser insuficiente para detectar correctamente la forma de las ondas, debido a la impresicion de los datos.

Un reto importante fue diferenciar entre señales cuadradas, triangulares y analógicas , lo cual implicó el uso de una combinación de punteros, estructuras de control, y manejo dinámico de memoria. Este enfoque nos permitió identificar patrones en las pendientes de las señales, una técnica clave para clasificar las señales con mayor precisión. Se introdujeron nuevas variables, como el conteo de pendientes (C) e inicialmente estaba planteado un arreglo dinámico para guardar estas pendientes en el, pero aumentó mucho el tiempo de procesamiento de datos, dando así datos mas imprecisos, por lo que esta implementación quedó descartada.

El ajuste de los umbrales de detección de cambios bruscos en las señales fue otro factor crítico en la evolución del código. Tuvimos que calibrar estos umbrales en función de la frecuencia detectada para evitar errores en los cálculos de las señales, aun así se nos hizo inevitable que ocurrieran algunas falencias a la hora de poner en practica el código.

En palabras de lecciones aprendidas, es muy tener en cuenta el impacto de la capacidad de procesamiento de un circuito y su efecto en el rendimiento general del sistema. En futuras implementaciones, sería útil integrar más técnicas de filtrado y suavización de las señales para mejorar la estabilidad en la detección de formas de onda,y desarrollar métodos para que los datos sean capturados de una mejor forma.