**Diseño de un Sistema de Adquisición de Señales Analógicas con Arduino**

Juan David Naranjo Jara y José Alejandro Pabón Leal  
  
  
  
Informa2  
  
  
  
12/09/24

**Introducción**

En este proyecto se busca desarrollar una solución que permita medir y analizar características de una onda electromagnética como su amplitud, frecuencia y forma de onda, utilizando un Arduino Uno. Estas señales pueden tener diferentes formas, como ondas senoidales, cuadradas, triangulares, entre otras. La medición precisa de estas propiedades es fundamental en aplicaciones como la sincronización de sistemas.

El objetivo principal es diseñar un algoritmo que, mediante la adquisición de datos de una señal, determine su amplitud, frecuencia y otros parámetros importantes, proporcionando una solución robusta y eficiente para el análisis de señales.

**Análisis del Problema**

Las señales analógicas son representaciones físicas de magnitudes que varían en el tiempo. Para este proyecto, se utilizarán señales generadas artificialmente, que serán analizadas a través de una de las entradas analógicas del Arduino Uno. Los parámetros clave a medir incluyen:

Forma de la onda: Determinada a partir de los datos adquiridos en función del tiempo.

Precisión de las lecturas: Las señales serán leídas a través del ADC de 10 bits del Arduino, con un rango de 0 a 1023. Para señales rápidas, la precisión puede verse afectada por el ruido y la tasa de muestreo.

Amplitud y Frecuencia: Las lecturas continuas deben registrar los valores máximos y mínimos de la señal para calcular la amplitud y medir el tiempo entre picos para calcular la frecuencia.

Uso de la memoria: Dado que el Arduino tiene recursos limitados, es importante controlar el uso de la memoria dinámica (heap), especialmente cuando se almacena un gran número de datos.

Consideraciones para la Solución Propuesta

Uso de la memoria dinámica: La memoria dinámica es limitada en el Arduino Uno (2 KB de SRAM), lo que restringe la cantidad de datos que pueden almacenarse temporalmente. Para evitar el agotamiento del heap, se debe imponer un límite en el almacenamiento de datos, utilizando estructuras eficientes, como buffers circulares o mecanismos de redimensionamiento controlado.

Optimización del uso de memoria: Se propone el uso de la memoria flash para almacenar datos o constantes que no cambian durante la ejecución. Esto ayuda a reducir el consumo de la SRAM.

Control del heap: Se debe monitorear el consumo de memoria dinámica, implementando un algoritmo que evite el almacenamiento indefinido de datos. En este sentido, se limitará el número de datos adquiridos, estableciendo un umbral máximo de uso de memoria de 500 bytes o1000 bytes para evitar problemas de rendimiento.

**Solución Propuesta**

El siguiente algoritmo será implementado para medir las características de la señal analógica:

Adquisición de la señal: Usar la función analogRead() para leer el valor de la señal analógica en tiempo real.

Determinación de picos: Detectar los picos máximos y mínimos de la señal para calcular la amplitud.

Cálculo de la frecuencia: Medir el tiempo entre dos picos consecutivos para calcular el periodo de la señal, que luego se usará para estimar la frecuencia.

Control del heap: Limitar el almacenamiento de datos a un máximo de 1000 bytes para evitar el consumo excesivo de la memoria dinámica.

Uso de buffers circulares: Implementar un buffer circular para evitar el desbordamiento del heap, donde los datos más antiguos se sobrescriben con los más nuevos una vez que se ha alcanzado la capacidad máxima.

Además del enfoque básico, se han considerado los siguientes puntos para mejorar la precisión y eficiencia del algoritmo:

Amplitud máxima y mínima: El algoritmo debe adquirir los valores máximos y mínimos en un intervalo de tiempo para determinar la frecuencia y el comportamiento general de la señal.

Puntos en la señal: La adquisición de datos debe realizarse en varios puntos a lo largo del tiempo para detectar posibles cambios en la forma de la señal, permitiendo así identificar si la señal es senoidal, cuadrada, triangular, entre otras.

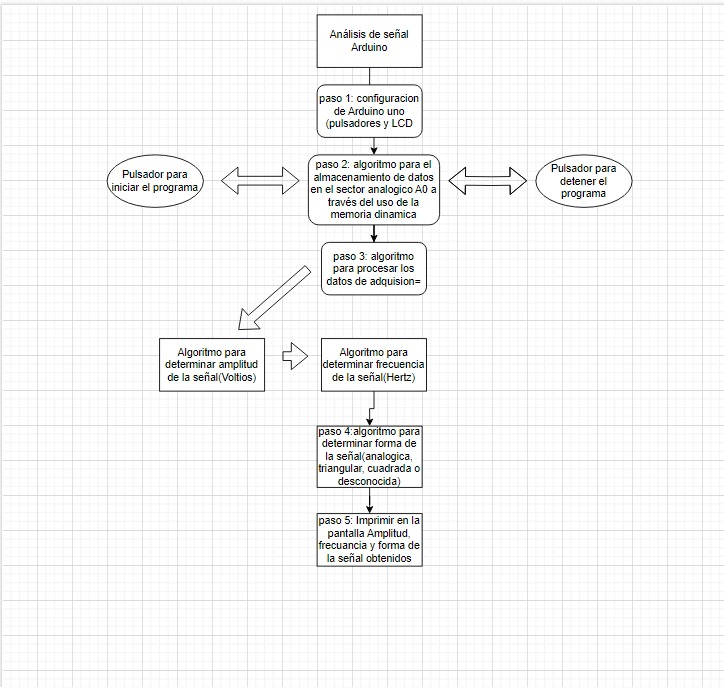
Tiempo en segundos: Se implementará un contador en tiempo real para medir el periodo de la onda con precisión, utilizando la función millis() para registrar el tiempo entre dos picos máximos consecutivos.

Condición para la amplitud: Se implementará una condición que evitará cambiar la amplitud durante la adquisición de datos, asegurando que los valores máximos y mínimos de la señal sean consistentes.

Cálculo del periodo: Se calculará el tiempo entre dos picos máximos consecutivos para obtener el periodo de la señal, y a partir de este, se calculará la frecuencia (Hz).

Control del heap: Se limitará el número de datos almacenados para evitar el agotamiento de la memoria dinámica, utilizando un mecanismo de redimensionamiento controlado o un buffer circular con un límite máximo de datos de 500 muestras.

Análisis de intervalos: Se analizarán diferentes intervalos de voltaje para identificar variaciones en la frecuencia y forma de la señal. Por ejemplo, analizar señales que varíen entre 2 y 4 voltios puede ofrecer información adicional sobre su comportamiento

Diagrama del análisis propuesto