

Detalle Para La Solución Del Desafío De Informática.

Pablo Emilio Valencia Morales.

Estudiante de Ingeniería electrónica de la Universidad de Antioquia.

Introducción

El presente informe detalla la solución propuesta para el desafío planteado por la empresa Informa2, el cual consiste en el desarrollo de un sistema de adquisición y visualización de características de una señal analógica. El objetivo principal es implementar un sistema que permita medir la frecuencia y amplitud de una señal de entrada, identificar su forma de onda y visualizar los resultados en una pantalla LCD. A continuación, se presenta un análisis del problema, el esquema de tareas, los algoritmos implementados, los problemas de desarrollo encontrados y la evolución de la solución.

Análisis del Problema

La adquisición de una señal analógica, cuya frecuencia y amplitud pueden variar, exige un procesamiento eficiente para garantizar una identificación precisa de sus características. En este sistema, el proceso de captura y análisis de la señal se lleva a cabo a través de varios componentes conectados al microcontrolador Arduino, como se observa en el diagrama. El proyecto integra botones, un generador de señales y una pantalla LCD para la interacción con el usuario y la visualización de los resultados.

- **Proceso de adquisición y análisis de la señal:** El proceso comienza con la activación de un pulsador que inicia la adquisición de la señal analógica. Esta señal es capturada en tiempo real mediante el puerto analógico del Arduino (A0), que es capaz de leer variaciones de voltaje en el rango de 0V a 5V, representadas por valores entre 0 y 1023. Los valores obtenidos se almacenan dinámicamente en un array que se redimensiona según el volumen de datos, aprovechando la memoria dinámica del sistema.

Para el procesamiento eficiente de los datos, se emplea un algoritmo que se encarga de extraer los parámetros fundamentales de la señal, tales como:

Frecuencia: Expresada en Hertz (Hz), la frecuencia es calculada midiendo los ciclos completos de la señal en un intervalo de tiempo determinado.

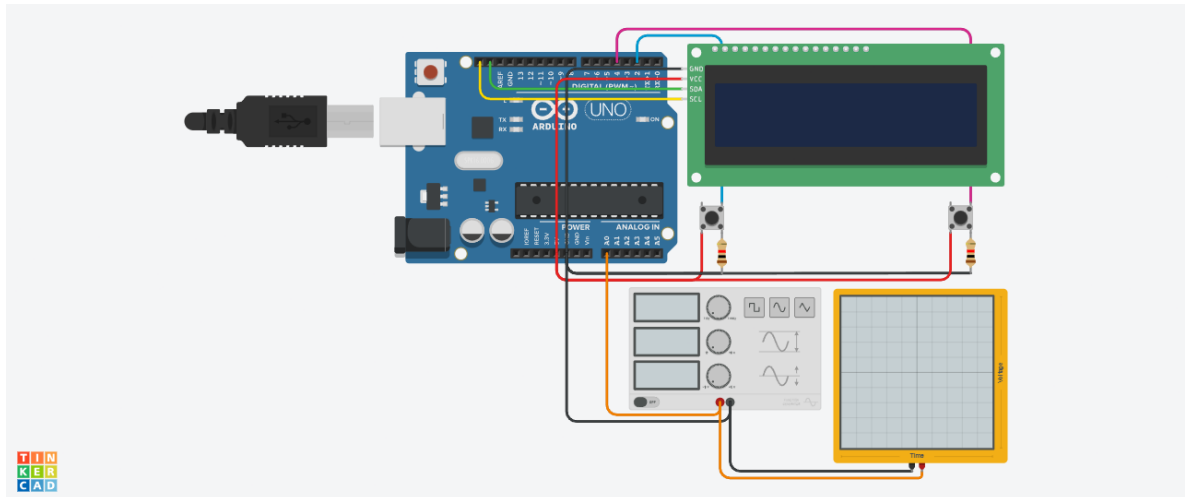
Amplitud: En voltios (V), que representa la diferencia entre el valor máximo y mínimo de la señal.

Forma de onda: El sistema identifica si la señal corresponde a una forma senoidal, triangular o cuadrada mediante el análisis de las transiciones y el patrón de los valores.

- **Almacenamiento y visualización:** Dado que el espacio de memoria RAM disponible en el microcontrolador es limitado (alrededor de 2 KB en el Arduino Uno), es esencial gestionar eficientemente el almacenamiento de los datos. Para ello, se emplea memoria dinámica, lo que permite ajustar el tamaño del array en función de la cantidad de datos recibidos. Esta estrategia asegura que se pueda procesar una mayor cantidad de muestras sin desperdiciar memoria ni comprometer el rendimiento del sistema.

Una vez procesada la señal, los resultados se presentan en una pantalla LCD conectada al sistema. La pantalla muestra en tiempo real los parámetros identificados de la señal: frecuencia, amplitud y el tipo de forma de onda detectada. Esta información se actualiza periódicamente, lo que permite al usuario tener un seguimiento continuo del comportamiento de la señal en análisis.

- **Interacción y control:** La interacción con el sistema se realiza mediante los pulsadores. Un botón se utiliza para iniciar el proceso de adquisición de datos, mientras que otro permite mostrar los resultados almacenados en la pantalla LCD. Este flujo de trabajo asegura que el usuario tenga un control sencillo pero efectivo sobre el sistema,



Algoritmos Implementados

A continuación, se describen los algoritmos clave implementados en la solución:

- **Adquisición de Señal y Almacenamiento Dinámico:**

Se implementó un algoritmo basado en el uso de un arreglo dinámico para gestionar el almacenamiento de los valores de la señal. A diferencia de los arreglos estáticos, que tienen un tamaño fijo desde su declaración, el arreglo dinámico se ajusta de forma automática conforme crece el número de muestras capturadas. Para lograr esto, se utiliza la función ***new*** para la asignación de memoria en tiempo de ejecución, lo que permite expandir el arreglo según sea necesario. A medida que se alcanzan los límites de la capacidad actual del arreglo, se redimensiona duplicando su tamaño, lo que minimiza la cantidad de reasignaciones y mejora la eficiencia.

Además, para evitar fugas de memoria y garantizar un uso eficiente de los recursos del sistema, se emplea ***delete[]*** para liberar la memoria asignada cuando ya no es necesaria. Este proceso de gestión dinámica del almacenamiento permite manejar grandes cantidades de datos sin desperdiciar memoria, lo que es crucial en entornos con recursos limitados como los sistemas embebidos. El algoritmo también tiene en cuenta las limitaciones de memoria del sistema y se adapta de forma eficiente, garantizando que el procesamiento de los datos sea escalable y controlado. Este enfoque dinámico optimiza el uso de la memoria y asegura que el sistema funcione dentro de sus capacidades sin comprometer el rendimiento.

- **Estructura del Array Dinámico:**

- Se inicia con un tamaño base que se expande de manera exponencial hasta alcanzar el límite máximo permitido por la memoria disponible, permitiendo almacenar hasta 300 valores de la señal. Este límite se determinó tomando en cuenta que la memoria RAM disponible es de 2 KB. Al calcular el espacio utilizado por las variables globales (aproximadamente 50 bytes) y reservar un 20% de la memoria total para evitar sobrecargas (unos 410 bytes), el resto de la memoria se asignó a dos arreglos dinámicos: uno de uso constante y uno auxiliar. Ambos arreglos almacenan valores enteros, que ocupan 2 bytes por elemento. Con estas consideraciones, se concluyó que la capacidad máxima del arreglo dinámico sería de 300 valores. Los valores capturados se almacenan en el array y se procesan en cada ciclo de adquisición.
- Los valores capturados se almacenan en el array y se procesan en cada ciclo de adquisición.

Identificación de Parámetros Clave de la Señal

- **Frecuencia:** El cálculo de la frecuencia se realiza midiendo los cruces por cero, tomando como referencia el valor medio de la amplitud de la señal. Cada cruce de la señal por este umbral cuenta como medio ciclo, y la suma de estos cruces durante un segundo se utiliza para determinar la frecuencia.
- **Amplitud:** La amplitud se obtiene calculando la diferencia entre los valores máximo y mínimo de la señal. Estos valores se actualizan continuamente a medida que se capturan nuevas muestras.
- **Forma de Onda:** Para la identificación de la forma de onda (cuadrada, senoidal o triangular), se analizan las transiciones entre los valores de la señal. Se utilizan comparaciones de los cambios bruscos y suaves en la señal para determinar la clasificación:
 - **Onda Cuadrada:** Se identifica por tener transiciones abruptas (cambios del 90% de la amplitud).
 - **Onda Triangular:** Se identifica por transiciones lineales y más suaves.
 - **Onda Senoidal:** Se asume cuando las transiciones no corresponden a ninguna de las otras dos formas de onda.

Visualización de los Resultados

Los resultados de la medición, como el valor mínimo, máximo, frecuencia, amplitud y tipo de señal, se muestran en una pantalla LCD. La pantalla se actualiza cada segundo con los valores más recientes de la señal.

Problemas de Desarrollo

Durante el desarrollo, se identificaron varios desafíos:

1. **Limitaciones de Memoria:** A medida que se recopilan más datos, el array puede necesitar redimensionarse, lo que puede provocar problemas si la memoria disponible es limitada o si el proceso de redimensionamiento no es eficiente.

Se optó por la optimización del Array Dinámico, el cual llega a un límite antes descrito y posteriormente se reescribe sobre sí, corriéndose un dato a la izquierda, dando espacio a un nuevo dato.

2. **Precisión en el Cálculo de Frecuencia:** La precisión en el cálculo de la frecuencia puede verse afectada por la resolución temporal y la manera en que se detectan los cruces por cero. El método utilizado en el código puede no ser lo suficientemente preciso si los datos de la señal no están bien muestreados.

Posible Solución:

- **Mejora en la Detección de Cruces por Cero:** Implementar un método más preciso para detectar cruces por cero, como usar un umbral más definido y considerar el uso de técnicas de filtrado para mejorar la precisión de la detección.
- **Incremento de Resolución Temporal:** Asegurarse de que el tiempo de muestreo sea lo suficientemente rápido para capturar con precisión los cruces por cero y calcular la frecuencia. Esto puede requerir una revisión del valor de delayMicroseconds y el método de muestreo.

3. **Interrupción de los Botones:** El problema con la interrupción de los botones se debió a la falta de solución para manejar la entrada de botones sin interferir con el proceso de adquisición de datos.

Posible Solución:

- **Uso de Interrupciones:** Implementar interrupciones para manejar los botones si es necesario. Esto permite que el microcontrolador responda a las entradas de botones sin interrumpir el procesamiento de datos.
 - **Debounce de Botones:** Existen técnicas de "debouncing" para evitar múltiples lecturas de un solo evento de botón.
4. **Precisión en la Identificación de la Forma de la Señal:** La precisión en la identificación de la forma de la señal puede verse afectada por la cantidad de muestras disponibles y la forma en que se analiza la señal.

Posible Solución:

- **Aumento de Muestras:** Asegurarse de que se están recopilando suficientes muestras para una identificación precisa de la forma de la señal.
- **Validación y Pruebas:** Realizar pruebas exhaustivas con diferentes tipos de señales para validar el algoritmo de identificación de forma y ajustarlo según sea necesario.

Evolución de la Solución

La solución comenzó con una implementación básica centrada en el almacenamiento de los valores de la señal. En esta etapa, el enfoque principal fue recopilar y guardar los datos de la señal sin considerar aspectos avanzados de procesamiento o análisis.

A medida que se identificaron limitaciones en el almacenamiento de datos, se incorporó la capacidad de ajustar dinámicamente el tamaño del array. Este paso permitió gestionar una mayor cantidad de datos y adaptarse a las variaciones en el número de muestras.

Se implementaron algoritmos para procesar los datos obtenidos y hallar los picos máximos, y mínimos, la amplitud la frecuencia.

Se desarrolló un algoritmo teórico para identificar la forma de la señal, basándose en el análisis de las transiciones y patrones de los datos. Este enfoque pretendía clasificar la señal como cuadrada, triangular o senoidal. A pesar de los esfuerzos realizados, la solución no logró satisfacer completamente los requisitos del proyecto. Los problemas principales incluyeron:

- **Falta de Análisis de Muestras Adecuado:** La insuficiencia en la cantidad y calidad de las muestras recopiladas impidió una identificación precisa de la forma de la señal.
- **Clasificación de la Forma de la Señal:** El algoritmo para determinar la forma de la señal no pudo diferenciar correctamente entre las diferentes formas de onda debido a las limitaciones en el análisis de datos. Como resultado, la señal siempre se identificó como senoidal, lo que no reflejaba con precisión la realidad de las señales analizadas.
- **Implementación de interrupción de botones:** La interrupción de los botones para la recolección de datos fue un desafío adicional. Aunque se intentó abordar el problema, no se logró implementar una solución efectiva que permitiera la activación y desactivación de los botones sin interrumpir el procesamiento de datos. Como resultado, se optó por un enfoque basado en la comprobación de la información en intervalos de tiempo definidos.
- **Frecuencias Extremadamente Altas:** El manejo de frecuencias extremadamente altas presentó un desafío significativo. Ajustar el tiempo de adquisición y procesamiento resultó crucial para mejorar la precisión de los resultados. Sin embargo, debido a las limitaciones del sistema, la precisión en la detección de frecuencias altas fue afectada, indicando la necesidad de optimizar el tiempo de muestreo y procesamiento para manejar estas frecuencias con mayor eficacia.

Conclusión

El desarrollo del sistema de adquisición y visualización de señales analógicas para la empresa Informa2 ha sido un proceso complejo y desafiante, caracterizado por una serie de iteraciones y mejoras en la solución propuesta. A lo largo del proyecto, se ha abordado la implementación de un sistema capaz de medir la frecuencia, amplitud y forma de onda de una señal analógica, además de proporcionar una visualización en tiempo real a través de una pantalla LCD.

En resumen, aunque se lograron avances significativos en el desarrollo del sistema, varios problemas clave afectaron el cumplimiento total de los requisitos del proyecto. La falta de

un análisis adecuado de las muestras, la precisión en la identificación de la forma de la señal, la interrupción de botones y el manejo de frecuencias altas son áreas que requieren atención y mejoras adicionales. Para futuros desarrollos, se recomienda realizar una revisión exhaustiva de las técnicas de muestreo, optimizar el manejo de interrupciones y mejorar el análisis de datos para alcanzar una solución más precisa y efectiva.