**1. Análisis del Problema y Consideraciones**

**Descripción del problema:**

El objetivo del proyecto es crear un sistema que adquiera señales analógicas y las procese utilizando un Arduino UNO. Los datos adquiridos, como la frecuencia y amplitud de la señal, deben ser mostrados en una pantalla LCD. El control del inicio y pausa de la adquisición de datos se maneja mediante pulsadores.

**Consideraciones:**

* **Uso de dos pulsadores**: Uno para iniciar la adquisición y otro para pausarla.
* **Uso de la pantalla LCD**: La pantalla LCD de 16x2 muestra los mensajes de estado del sistema.
* **Librería Adafruit\_LiquidCrystal.h**: Se utiliza para controlar la pantalla LCD.
* **Monitor serial**: Los valores de la señal analógica se imprimen en el monitor serial para seguimiento.

**2. Esquema de Tareas**

El desarrollo del proyecto se organiza en las siguientes etapas:

**1. Montaje del Hardware**

* **Conectar los pulsadores** a los pines digitales:
  + **Pin 3**: Para iniciar la adquisición de datos.
  + **Pin 5**: Para pausar la adquisición de datos.
* **Conectar la pantalla LCD**:
  + Usar la librería Adafruit\_LiquidCrystal.h para controlar una pantalla LCD de 16x2.
* **Conectar la señal analógica** al **pin A0** del Arduino, que será la entrada de la señal a procesar.

**2. Implementación de la Lógica del Pulsador (Completado)**

* **Pulsador de inicio** (pin 3): Inicia la adquisición de datos y muestra "Inicio" en la pantalla LCD.
* **Pulsador de pausa** (pin 5): Pausa la adquisición de datos y muestra "Pausa" en la pantalla LCD.
* Mostrar los valores de la señal adquirida en el **monitor serial**.

**3. Implementación de la Medición de la Señal**

* **Frecuencia de la señal**:
  + Implementar un algoritmo que detecte los picos de la señal analógica.
  + Calcular el tiempo entre picos consecutivos para determinar la frecuencia en Hertz (Hz).
* **Amplitud de la señal**:
  + Calcular el valor máximo y el valor mínimo de la señal leída.
  + Usar estos valores para calcular la amplitud en Voltios.

**4. Identificación de la Forma de la Onda**

* Implementar un algoritmo que clasifique la señal adquirida según su forma.
* Formas de onda a identificar:
  + Señal senoidal.
  + Señal triangular.
  + Señal cuadrada.
  + Desconocida
* Si la señal no coincide con ninguna de las formas predefinidas, mostrar como "señal desconocida".

**5. Visualización de los Datos en la Pantalla LCD**

* Mostrar en la pantalla LCD los valores calculados:
  + **Frecuencia** en Hertz.
  + **Amplitud** en Voltios.
  + **Forma de la señal** identificada.

**3. Algoritmos Implementados**

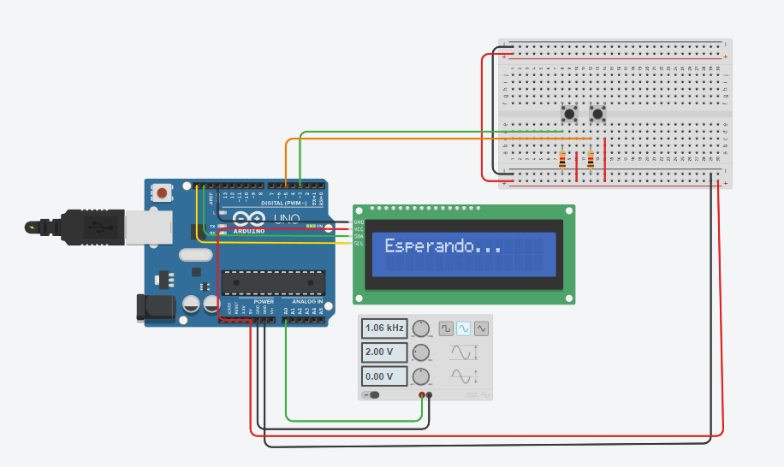
**1) Primer commit - Control de Pulsadores:**

Se implementaron dos pulsadores, conectados a los pines 3 y 5 del Arduino:

* **Pulsador 1 (pin 3)**: Inicia la adquisición de datos cuando se presiona.
* **Pulsador 2 (pin 5)**: Pausa la adquisición de datos cuando se presiona.

La variable booleana confirm controla el estado del sistema:

* Si confirm es true, el sistema está en estado de adquisición.
* Si confirm es false, el sistema está en pausa o esperando.



**Explicación del Código:**

* **Inicio**: El sistema comienza en modo de espera, mostrando "Esperando..." en la pantalla LCD.
* **Iniciar adquisición**: Cuando el pulsador conectado al **pin 3** se presiona, la pantalla LCD muestra "Inicio" y el sistema comienza a adquirir datos de la señal analógica a través del pin A0.
* **Pausa de adquisición**: Si el pulsador conectado al **pin 5** se presiona, el sistema muestra "Pausa" en la pantalla LCD y se detiene la adquisición de datos.
* Los valores de la señal analógica se imprimen en el monitor serial para su monitoreo.

Hasta el momento, el sistema permite controlar la adquisición de datos de una señal analógica utilizando dos pulsadores. Se ha implementado la visualización de los estados del sistema en una pantalla LCD. El siguiente paso será agregar la funcionalidad para calcular la frecuencia y la amplitud de la señal adquirida, así como implementar la detección de la forma de la onda.

**2) Segundo commit - Control de Pulsadores:**

Se agregó el arreglo dinámico donde se van a guardar los valores para procesarlos.

La segunda parte del proyecto consistió en diseñar e implementar una estructura de datos que permitiera almacenar los valores de la señal de entrada de forma eficiente. Dado que la cantidad de datos podía variar, era necesario usar **memoria dinámica** para gestionar el almacenamiento de los datos.

**Objetivo:**

* Implementar un sistema que pudiera manejar la adquisición de datos sin saber de antemano el tamaño de la señal, evitando desbordamientos de memoria y haciendo un uso eficiente de los recursos disponibles.

**Estrategia:**

* Empezamos con un arreglo de tamaño fijo que se duplicaba dinámicamente a medida que se llenaba.
* Se estableció un **tamaño máximo del arreglo** para evitar que se consumiera toda la memoria disponible en el microcontrolador.

**Problemas Iniciales:**

* **Fugas de memoria**: Al no liberar correctamente la memoria tras el uso, había riesgo de **desbordamiento de memoria**.
* **Sobrecarga en la duplicación**: Aumentar el tamaño del arreglo de manera indefinida podría ocasionar que la memoria del microcontrolador se agotara rápidamente.

**Soluciones:**

* Se agregó un límite máximo al tamaño del arreglo.
* Se aseguró que la memoria anterior fuera liberada correctamente con delete[] antes de redimensionar el arreglo.

**3) Implementación de la Medición de Datos**

Una vez que el sistema de almacenamiento dinámico estuvo funcionando correctamente, pasamos a la **medición de los datos de la señal**

**3.1 Lectura y Almacenamiento de Datos**

La lectura de datos se implementó usando el pin **A0**, donde se capturaba la señal analógica del generador de funciones. Estos datos se almacenaban en el arreglo dinámico implementado previamente. Para controlar el inicio y la pausa de la adquisición de datos, se usaron dos pulsadores conectados a los pines **3** y **5** (para iniciar y pausar, respectivamente).

**3.2 Cálculo de Frecuencia**

El cálculo inicial de la frecuencia se realizó detectando **picos** en la señal. Sin embargo, se encontró que este método no era eficiente para ondas cuadradas, ya que las transiciones abruptas no generaban suficientes picos. Por ello, se implementó un método basado en **contar las transiciones** entre valores altos y bajos en la señal.

Este enfoque se intentó resolver los problemas con las señales cuadradas al detectar correctamente las transiciones.

**3.3 Cálculo de Amplitud**

El cálculo de la amplitud fue sencillo y consistió en encontrar la diferencia entre los valores máximo y mínimo de la señal. Luego se ajustaron estos valores al rango de voltaje correspondiente (0-5V).

**4. Mejoras y Retos Finales: Detección de la Forma de la Onda**

**4.1 Problemas Iniciales**

Para identificar la **forma de la onda**, inicialmente se propuso contar la cantidad de **valores únicos** en los datos. Sin embargo, este método resultó **ineficiente** para señales con grandes cantidades de datos, ya que no diferenciaba claramente las formas de onda cuando las señales tenían pequeñas variaciones.

**4.2 Nueva Estrategia Implementada**

Finalmente, se implementó una estrategia que analiza los **patrones** en los datos:

* **Onda Cuadrada**: Se detecta cuando la señal pasa la mayor parte del tiempo en dos niveles claramente definidos (alto y bajo), con transiciones bruscas entre ellos.
* **Onda Triangular**: Se detecta cuando la señal sube y baja de manera continua y gradual.
* **Onda Senoidal**: Si la señal no cumple con los criterios anteriores, se clasifica como senoidal.

Esta implementación permitió una clasificación precisa de las formas de onda, incluso en señales con pequeñas variaciones.

aunque el sistema intenta cumplir con su propósito principal y ofrece una solución funcional p, **no es perfecto** y tiene sus limitaciones técnicas. Sin embargo, estas limitaciones proporcionan una base sólida para futuras mejoras y optimizaciones en proyectos de análisis de señales más complejos.