**Desafío I**

**Laura Nataly Oquendo Vera cc 1036684701**

**Brayan David Sabogal Ramírez**

**Facultad de Ingeniería Universidad de Antioquia**

**Informática II**

**Septiembre 2024**

**PROYECTO INFORMA2**

**Descripción del Problema**

La empresa Informa2 necesita un sistema que permita identificar las características principales de una señal analógica y visualizar los resultados. El sistema debe iniciar la adquisición de datos mediante un pulsador y permitir la solicitud de información de la señal capturada en cualquier momento. Durante el procesamiento de la información, la adquisición de datos puede ser suspendida y debe reanudarse una vez finalizado el procesamiento.

**Requisitos Específicos**

* Iniciar la adquisición de datos con un pulsador.
* Solicitar información de la señal capturada con otro pulsador.
* Visualizar las características de la señal en una pantalla LCD.
* Medir la frecuencia y la amplitud de la señal.
* Identificar la forma de onda de la señal.

**Consideraciones para la Solución Propuesta**

**Alternativa de Solución**

La solución propuesta implica el uso de una plataforma de simulación como Tinkercad para diseñar y probar el sistema. Se utilizará un Arduino para la adquisición y procesamiento de la señal, junto con una pantalla LCD para la visualización de los resultados.

Componentes Utilizados:

* Arduino Uno: Para la adquisición y procesamiento de la señal.
* Generador de Señales: Para generar la señal analógica de entrada.
* Pulsadores: Para iniciar la adquisición de datos y solicitar la información de la señal.
* Pantalla LCD: Para visualizar las características de la señal.
* Librería Adafruit\_LiquidCrystal.h: Para facilitar la interacción con la pantalla LCD.

**Pasos para la Implementación:**

1. **Montaje del Circuito:**
   * Conectar el generador de señales al pin de entrada analógica del Arduino.
   * Conectar los pulsadores a los pines digitales del Arduino.
   * Conectar la pantalla LCD al Arduino utilizando la librería Adafruit\_LiquidCrystal.h.
2. **Desarrollo del Código:**
   * Escribir un código en Arduino para iniciar la adquisición de datos con un pulsador.
   * Implementar funciones para medir la frecuencia y la amplitud de la señal.
   * Desarrollar un algoritmo para identificar la forma de onda de la señal.
   * Mostrar los resultados en la pantalla LCD.
3. **Pruebas y Validación:**
   * Realizar pruebas con diferentes formas de onda generadas por el generador de señales.
   * Validar que el sistema mide correctamente la frecuencia y la amplitud.
   * Verificar que el sistema identifica correctamente la forma de onda y muestra los resultados en la pantalla LCD.

**Esquema de Tareas para el Desarrollo**

**1. Configuración Inicial**

**Tarea**: Configurar los componentes del hardware y las variables iniciales.

* **Subtareas**:
  + Inicializar la pantalla LCD.
  + Configurar pines de botones y pin de entrada analógica.
  + Configurar el buffer de datos.

**Objetivo**: Preparar el sistema para la adquisición y procesamiento de datos.

**2. Iniciar Adquisición**

**Tarea**: Iniciar el proceso de adquisición de datos desde el pin analógico.

* **Subtareas**:
  + Verificar si la adquisición está activa.
  + Limpiar la pantalla LCD y mostrar mensaje de inicio.
  + Leer datos del pin analógico y almacenarlos en el buffer.
  + Controlar el tamaño del buffer y detener la adquisición si es necesario.

**Objetivo**: Capturar datos en tiempo real y almacenarlos para su posterior análisis.

**3. Solicitar Información**

**Tarea**: Procesar y mostrar la información de la señal capturada.

* **Subtareas**:
  + Mostrar mensaje de procesamiento en la pantalla LCD.
  + Pausar temporalmente la adquisición de datos si es necesario.
  + Llamar a la función de análisis de señal.
  + Reanudar la adquisición después del procesamiento.

**Objetivo**: Permitir al usuario solicitar información y realizar el procesamiento necesario para mostrar los resultados.

**4. Analizar Señal**

**Tarea**: Procesar los datos del buffer para extraer características de la señal.

* **Subtareas**:
  + Encontrar los valores máximos y mínimos en el buffer.
  + Calcular la amplitud de la señal.
  + Implementar lógica para determinar la frecuencia y forma de onda.

**Objetivo**: Extraer características importantes de la señal, como amplitud, frecuencia y forma de onda.

**5. Mostrar Resultados**

**Tarea**: Presentar los resultados del análisis en la pantalla LCD.

* **Subtareas**:
  + Identificar la forma de onda basada en los datos analizados.
  + Mostrar la amplitud y frecuencia en la pantalla LCD.
  + Actualizar la pantalla con la forma de onda identificada.

**Objetivo**: Visualizar los resultados del análisis de señal de forma clara y comprensible para el usuario.

**6. Identificar Forma de Onda**

**Tarea**: Determinar el tipo de forma de onda de la señal capturada.

* **Subtareas**:
  + Analizar los datos del buffer para detectar patrones de ondas.
  + Comparar los patrones con las formas de onda conocidas (pulso, triangular, cuadrada).
  + Identificar si la forma de onda es desconocida.

**Objetivo**: Clasificar la forma de onda de acuerdo con los patrones predefinidos.

**Descripción de Diagrama de Flujo**

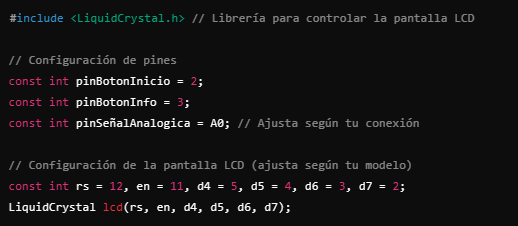
1. **Inicio**: El programa comienza su ejecución.
2. **Configuración Inicial**:
   * Inicializa la pantalla LCD.
   * Configura los pines de los botones y del generador de señales.
3. **Esperar Acción del Usuario**:
   * **Botón de Inicio Presionado**:
     + Comienza la adquisición de datos.
     + Lee datos del pin analógico y almacena en el buffer.
     + Si el buffer está lleno o se detiene la adquisición, pasa al siguiente paso.
   * **Botón de Información Presionado**:
     + Procesa la información (frecuencia, amplitud, forma de onda).
     + Muestra la información en la pantalla LCD.
4. **Identificar Forma de Onda**:
   * Analiza los datos almacenados en el buffer.
   * Identifica la forma de onda (pulso, triangular, cuadrada, desconocida).
5. **Mostrar Resultados**:
   * Muestra la frecuencia, amplitud, y forma de onda en la pantalla LCD.
6. **Reanudar Adquisición**:
   * Continúa la adquisición de datos si fue pausada.
7. **Fin**: El proceso se repite continuamente.

**Imagen que contiene Forma

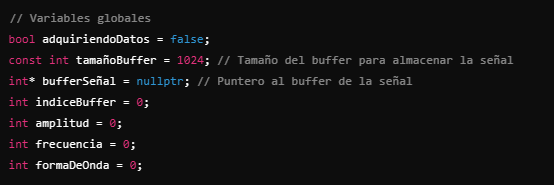
Descripción generada automáticamente**

**Algoritmos implementados**

* + **Inclusión de Librerías y Configuración de Pines**



* **Librería LiquidCrystal.h:** Se incluye para controlar la pantalla LCD. Esta librería proporciona funciones para interactuar con la pantalla.
* **Configuración de pines:** Se definen los pines a los que están conectados los botones y el pin de señal analógica.
* **Configuración LCD:** Se definen los pines utilizados para conectar la pantalla LCD a la placa Arduino.
  + **Declaración de Variables Globales**



* **adquiriendoDatos**: Indica si se están capturando datos en este momento.
* **tamañoBuffer**: Define el tamaño del buffer que se usará para almacenar los datos de la señal.
* **bufferSeñal**: Puntero a un arreglo dinámico que almacenará los datos de la señal.
* **indiceBuffer**: Indica la posición actual en el buffer donde se almacenará el próximo dato.
* **amplitud, frecuencia, formaDeOnda**: Variables para almacenar la amplitud, la frecuencia y la forma de onda de la señal analizada.
  + **Prototipos de Funciones**

Texto

Descripción generada automáticamente

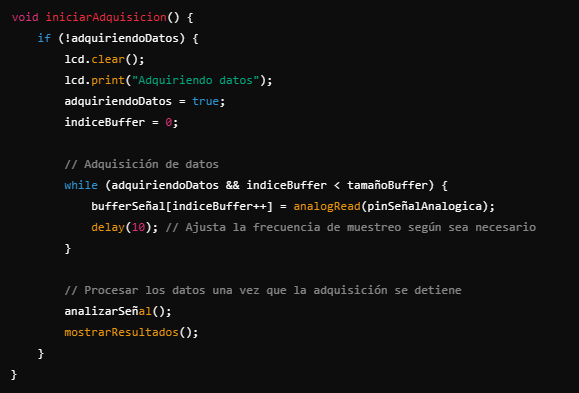
* **Prototipos:** Declaraciones anticipadas de funciones que se implementarán más adelante. Esto permite que las funciones sean llamadas en cualquier parte del código antes de que se definan.
  + **Configuración Inicial (setup)**

Texto

Descripción generada automáticamente

* **Inicialización de LCD:** Se inicializa la pantalla LCD con 16 columnas y 2 filas.
* **Configuración de Pines:** Los pines de los botones se configuran como entradas con resistencia pull-up interna, lo que significa que estarán en estado HIGH cuando no se presionen.
* **Inicialización del Buffer:** Se asigna memoria dinámica para el buffer usando new, que se usará para almacenar los datos de la señal.
* **Bucle Principal (loop)** Texto

  Descripción generada automáticamente
* **Esperar Acción del Usuario**: Se lee el estado de los botones. Si el botón de inicio se presiona (LOW), se llama a iniciarAdquisicion(). Si el botón de información se presiona, se llama a solicitarInformacion().
* **Retraso**: Se agrega un retraso de 500 ms para evitar que se detecten múltiples presiones de botón debido a rebotes.
* **Iniciar Adquisición (iniciarAdquisicion)**



* **Verificar Adquisición**: Solo comienza la adquisición si no se están capturando datos actualmente (adquiriendoDatos es false).
* **Mensaje en LCD**: Limpia la pantalla y muestra un mensaje indicando que se está capturando datos.
* **Captura de Datos**: Lee datos del pin analógico y los almacena en el buffer. Se realiza un retraso de 10 ms entre lecturas para controlar la frecuencia de muestreo.
* **Procesamiento y Visualización**: Una vez que se llena el buffer, se llama a analizarSeñal() para procesar los datos y a mostrarResultados() para mostrar los resultados en la pantalla LCD.
* **Solicitar Información (solicitarInformacion)**

Texto

Descripción generada automáticamente

* **Mensaje en LCD**: Muestra un mensaje indicando que se está procesando la información.
* **Pausar Adquisición**: Se detiene la captura de datos para procesar la información actual.
* **Retraso y Reanudación**: Se realiza un retraso de 2000 ms para simular el procesamiento de datos, luego se reanuda la adquisición de datos.
* **Analizar Señal (analizarSeñal)**

Texto

Descripción generada automáticamente

* **Verificación de Buffer**: Asegura que el buffer no sea nullptr antes de procesar los datos.
* **Cálculo de Amplitud**: Encuentra el valor máximo y mínimo en el buffer para calcular la amplitud de la señal.
* **Calcular Frecuencia y Identificar Forma de Onda**: Llama a las funciones calcularFrecuencia() y identificarFormaDeOnda() para obtener la frecuencia y la forma de onda de la señal.
* **Calcular Frecuencia (calcularFrecuenciaEnBuffer(int valor)**

Texto

Descripción generada automáticamente

* Cuenta cuántas veces aparece un valor específico en el buffer.
* **Identificar Forma de Onda (identificarFormaDeOnda)**

Texto

Descripción generada automáticamente

* Cuenta las transiciones positivas y negativas.
* Clasifica la forma de onda en sinusoidal, cuadrada o triangular basándose en la relación entre las transiciones.
* **Mostrar Resultados (mostrarResultados)**



* **Mostrar Datos:** Limpia la pantalla LCD y muestra los resultados de la señal, incluyendo la forma de onda, la amplitud y la frecuencia.
* **Switch para Forma de Onda**: Utiliza un switch para mostrar el tipo de forma de onda basado en el valor de formaDeOnda.