**Informe del Proyecto de Procesamiento de Señales con Arduino**

Análisis de las características principales de una señal (Frecuencia, amplitud y forma de onda)

Autores: Jesús David Guevara y Martín Santiago Rodríguez Forero

**Informática II - Universidad de Antioquia**

**Introducción**

Este proyecto consiste en la adquisición y procesamiento de señales analógicas utilizando un Arduino UNO, con el objetivo de calcular la **frecuencia** y el **ancho de onda** de la señal, representando los resultados en una pantalla **LCD**. La implementación se orienta al uso de **memoria dinámica**, para almacenar de forma flexible los valores de la señal mientras se realizan las lecturas. A través de dos pulsadores, el sistema puede iniciar y detener la adquisición de la señal, y finalmente, tras el procesamiento, se muestran los valores de frecuencia y amplitud.

Este informe estará compuesto por:

* Descripción del problema.
* Análisis del problema.
* Esquema de la solución.
* Estructura del código.
* Problemas abordados durante el desarrollo.
* Evolución de la solución.
* Resultados finales y conclusión.

**Descripción del Problema**

Se requiere un sistema capaz de:

1. **Leer una señal** generada y capturada a través del pin A0 del Arduino.
2. **Iniciar y detener** la adquisición de datos mediante pulsadores (uno para iniciar y otro para detener la lectura).
3. **Almacenar dinámicamente** los valores de la señal durante el intervalo de lectura.
4. **Procesar la señal** para calcular la frecuencia y la amplitud, y mostrar los resultados en la pantalla LCD y el monitor serial.
5. **Mostrar "Procesando datos"** mientras la adquisición está en curso.

**Recursos utilizados**

* Tinkercad y sus materiales incorporados (Como el generador de señales, los pulsadores y la pantalla LCD, entre otros) para las simulaciones de señales.
* Plataforma que permita programar en lenguaje C++ para la codificación del programa.
* Aplicaciones Git y Github para manejo de archivos con control de versiones y trabajo colaborativo remoto o en la nube.
* Microsoft Word para la creación del informe.

**Análisis del Problema**

La lectura de la señal debe ser controlada por pulsadores, y la cantidad de datos leída puede variar, lo que hace necesario el uso de **memoria dinámica** para no limitar el tamaño del almacenamiento. Los datos de la señal se almacenarán en un arreglo dinámico que crecerá conforme se vayan obteniendo las lecturas. Después de detener la lectura, se procederá a calcular la **frecuencia**, a partir del conteo de picos de la señal, y la **amplitud**, calculada como la diferencia entre el valor máximo y mínimo de la señal.

**Esquema de la Solución**

1. **Inicio de la Lectura**:
   * Al presionar el pulsador de inicio, se activa el modo de lectura, se inicializa la memoria dinámica para almacenar los valores, y se muestra "Iniciando lectura" en la pantalla LCD.
2. **Lectura de la Señal**:
   * Mientras el pulsador de inicio está activado, se leen continuamente los valores del pin analógico A0. Los datos se almacenan en un arreglo dinámico que crece con cada nueva lectura. Durante la lectura, se muestra "Procesando datos" en la pantalla LCD.
3. **Finalización de la Lectura**:
   * Al presionar el pulsador de parada, se detiene la lectura y se procede a procesar los datos almacenados. Se calcula la frecuencia y el ancho de onda (amplitud) de la señal. Los resultados se muestran en la pantalla LCD y el monitor serial.

**Estructura del Código**

**Variables Principales**

* int pinAnalogico: Pin analógico utilizado para lectura.
* int pinSenal: Pin de la señal del generador.
* bool estaLeyendo: Variable que controla el estado de la lectura.
* int pinBotonInicio, pinBotonParada: Pines de los pulsadores.

**Flujo del Programa**

1. **Setup**:
   * Se inicializa la pantalla LCD y la comunicación serie.
   * Se configuran los pulsadores como entradas.
2. **Loop**:
   * Se monitorean los pulsadores. Si se presiona el pulsador de inicio, se comienza la lectura de la señal y se almacenan los valores en el arreglo dinámico. Si se presiona el pulsador de parada, se detiene la lectura y se procesan los datos.
3. **Procesamiento de la señal**:
   * Una vez detenida la lectura, se calcula la frecuencia de la señal contando los picos (cada vez que la señal cruza un valor umbral) y el ancho de onda mediante la diferencia entre los valores máximo y mínimo.

**Funciones Principales**

* processSignal(int \*signalValues, int numReadings): Procesa los valores almacenados, calculando la frecuencia y la amplitud de la señal.

**Algoritmos Implementados**

1. **Lectura de la señal**: Se usa analogRead() para capturar los valores de la señal. Estos se almacenan en un arreglo dinámico que crece con cada nueva lectura usando new y delete[] para gestionar la memoria.
2. **Cálculo de la frecuencia**: Se cuentan los picos de la señal comparando los valores de la señal en cada punto del tiempo.
3. **Cálculo de la amplitud**: Se busca el valor máximo y mínimo de la señal durante el intervalo de lectura, y la diferencia entre ambos se convierte a voltios.

**Problemas Abordados**

1. **Manejo de memoria dinámica**: Para evitar el uso de arreglos de tamaño fijo, se implementó el uso de punteros y memoria dinámica, permitiendo el almacenamiento de un número indeterminado de lecturas.
2. **Detección de picos en la señal**: La frecuencia se calculó contando los cruces de la señal por un valor umbral (en este caso, 512), lo que permitió estimar el número de ciclos de la señal.

**Evolución de la Solución**

Durante el desarrollo, se añadieron los siguientes componentes:

1. **Control por pulsadores**: Dos pulsadores permiten controlar cuándo empieza y cuándo termina la lectura de la señal, lo que hace que el sistema sea interactivo.
2. **Uso de memoria dinámica**: Inicialmente, se utilizaban variables estáticas, pero se reemplazaron por punteros y manejo dinámico de la memoria para hacer el sistema más flexible.
3. **Procesamiento de la señal**: Se integró el procesamiento de la señal tras la lectura, lo que permite calcular y visualizar los resultados.

**Resultados Finales**

Al presionar el pulsador de inicio, el sistema comienza a capturar y almacenar los valores de la señal, mostrando "Procesando datos" en la pantalla LCD. Cuando se presiona el pulsador de parada, el sistema detiene la captura, procesa los datos almacenados y muestra en la pantalla LCD los resultados de la frecuencia y la amplitud de la señal, calculados en base a las lecturas realizadas.

**Conclusión**

Este proyecto ha logrado implementar con éxito un sistema de adquisición y procesamiento de señales utilizando Arduino y memoria dinámica. La integración de dos pulsadores para controlar el inicio y la parada de la lectura, junto con el uso de memoria dinámica para almacenar los datos y calcular la frecuencia y amplitud de la señal, ha permitido cumplir con los requisitos del proyecto. El siguiente paso podría ser optimizar el tiempo de lectura y mejorar la detección de picos para hacer el sistema más preciso.