Informe técnico sobre el proyecto “Generador de señales”

Presentado por:

Sofia Casas Centeno y Santiago Gallón Restrepo

Asignatura: Informática II

Profesor: Augusto Salazar

Universidad de Antioquia

**Introducción**

Arduino es una plataforma electrónica de código abierto con software y hardware accesibles. Estas placas pueden interpretar entradas como la luz de un sensor o la presión de un botón y luego convertirlas en salidas, como el arranque de un motor o el encendido de un LED. Además, por medio de este entorno se pueden simular diferentes tipos de programas. El presente proyecto ofrecerá un análisis detallado del problema planteado, describiendo las tareas definidas para el desarrollo e implementación de los algoritmos propuestos. Se abordarán los desafíos enfrentados durante la creación del código fuente y se presentará la solución final implementada, destacando su eficiencia y validación a través de pruebas exhaustivas.

**Desarrollo técnico**

Una empresa requiere que se identifique a partir de una señal analógica sus características principales y posteriormente visualizar los resultados de acuerdo con las siguientes indicaciones a tener en cuenta:

* Por medio de la activación de un pulsador se hará comienzo a la adquisición de datos.
* En cualquier instante se le solicitará al programa la información capturada de la señal.
* A través de una pantalla LCD se podrán observar las características que contendrá la señal.

Ahora, previamente descrito el problema se dará pie a una idea de solución la cual, garantizará y cumplirá con las directrices propuestas. Para ello se usará como medio de trabajo la plataforma ¨Tinkercad¨, la cual ofrece todas las herramientas requeridas para la creación del programa.

Los implementos que serán utilizados son: Arduino Uno, pulsadores, generador de ondas y pantalla LCD.

A continuación, se ensamblará el sistema de circuitos de Arduino, para ello se necesitarán un Arduino Uno, pantalla LCD de 4 pines con conectores: VCC, GNL, SCL, SDA, 2 pulsadores (Botones 1 y 2), resistencias (ambas de 10kΩ), cableado y fuente de alimentación.

Luego, se dispondrán a hacer las conexiones necesarias.

|  |  |
| --- | --- |
| **Partes** | **Conexiones** |
| **Pantalla LCD:** | VCC --> 5V de Arduino  GND --> GND de Arduino  SCL --> Pin A5 de Arduino (SCL)  SDA --> Pin A4 de Arduino (SDA) |
| **Pulsadores:** | Botón 1:  Un lado --> Pin 2 de Arduino (INT0)  Otro lado --> GND  Botón 2:  Un lado --> Pin 4 de Arduino  Otro lado --> GND |
| **Resistencias:** | - 10kΩ entre el pin de cada pulsador y VCC (pull-up)  - 10kΩ entre el pin de cada pulsador y VCC (pull-up) |
| **Cableado:** | - Se conectarán los cables según las conexiones descritas anteriormente.  - Se deberá tener precaución de que no haya cortocircuitos. |
| **Configuración en el código:** | - En el código, se configurarán los pines correspondientes para los pulsadores y la pantalla LCD.  - Se utilizará la librería *Adafruit\_LiquidCrystal.h.* para controlar la pantalla LCD. |
| **Montaje del circuito:** | - Se colocará la pantalla LCD en una base estable.  - Se conectarán los pulsadores en una posición accesible.  - Se pondrá el Arduino en una posición segura. |
| **Prueba (Verificación):** | - Se conectará la fuente de alimentación.  - Se verificará que la pantalla LCD se encienda y muestre la información correctamente.  - Se probarán los pulsadores para asegurarse de que funcionen correctamente. |

**Esquema del código:**

Previamente descrito el ensamblaje del sistema de circuitos de Arduino, se procederá a dar una breve explicación del esquema del código fuente para el desarrollo de los algoritmos requeridos, teniendo en cuenta que el principal objetivo de este código es lograr leer datos a partir de un sensor analógico, imprimirlos en la pantalla LCD y posteriormente identificar el tipo de onda captada (senoidal, cuadrada o dientes de sierra) gracias a la utilización de pulsadores.

El orden del código es:

1. Inicialización (Función setup() ):

* Se definirán los pines correspondientes para los componentes, se utilizará la librería *Adafruit\_LiquidCrystal.h,* y se inicializará la pantalla LCD.

1. Lectura de datos (Función loop() ):

* Se leen los datos del sensor analógico y se almacenarán en un arreglo.

1. Mostrar datos en pantalla LCD:

- Se imprimirán los datos obtenidos en la pantalla LCD.

4. Detección de tipo de onda (Función detectarTipoOnda() ):

- Se decidirá utilizar el método de reconocimiento de patrones y luego se identificará el tipo de onda

5. Función mostrarDatos():

- S mostrarán los datos obtenidos en la pantalla LCD

6. Cálculo de la frecuencia y la amplitud:

- Se crearán dos funciones calcularFrecuencia() para calcular la frecuencia de la onda y calcularAmplitud() para la amplitud.

7. Control de pulsadores:

- Se leerá el estado de los pulsadores y según sea el estado (inicio/parar lectura de datos) se realizarán acciones.

8. Variables:

- datos: Arreglo para almacenar datos del sensor

- tipoOnda: Variable para almacenar tipo de onda detectado

- frecuencia: Variable para almacenar frecuencia calculada

- amplitud: Variable para almacenar amplitud calculada

- pulsador1 y pulsador2: Variables para almacenar estado de los pulsadores.

**Problemas afrontados a partir del desarrollo de los algoritmos:**

1. Error de compilación: Incomprensión de cómo se refería a la librería *Adafruit\_LiquidCrystal.h*
2. Configuración de Tinkercad: Se presentaron dificultades para configurar correctamente la librería, a causa de utilizar la incorrecta.
3. Conexiones incorrectas: Se presentaron dificultades sobre como se conectaba la pantalla LCD y los pulsadores.
4. Lectura incorrecta de datos: Se notó en complejidad la manera en la se estaban guardando los datos para que no sucedieran inconsistencias.
5. Detección de tipo de onda: Se previeron dificultades para implementar un método para la detección de onda ya sea senoidal, cuadrada o dientes de sierra, puesto que cada una presenta diferentes formas de evaluar sus frecuencias y amplitudes.
6. Control de pulsadores: Se tuvieron problemas con el pull-up y pull-down e implementar el control de los pulsadores para iniciar/detener la lectura de datos.
7. Visualización de datos: Problemas con la visualización de los datos en la pantalla LCD.
8. Ajuste de parámetros: Necesidad de ajustar parámetros como el umbral para la detección de tipo de onda.
9. Compatibilidad con Tinkercad: Problemas con la compatibilidad del código con la versión de Tinkercad.
10. Tamaño del arreglo: Problemas de con el arreglo dinámico.

**Soluciones resultantes:**

- Tener en cuenta la revisión de la documentación de Tinkercad y la biblioteca LiquidCrystal.

- Se debe hacer ajuste de las conexiones y la configuración de los componentes.

- Implementación del método de patrones para la detección de tipo de onda.

- Revisión de las fórmulas para calcular la frecuencia y amplitud.

- Implementación de un sistema de control para los pulsadores.

**Conclusión**

A partir de los conocimientos adquiridos en clase y del esfuerzo realizado para la entrega de este proyecto se puede llegar a la conclusión de que los punteros son una herramienta útil y versátil en los procesos de programación, también el uso de arreglos dinámicos para guardar grandes cantidades de datos en la memoria.