

Informe Desafio

Informatica II

Juan Pablo Mesa Marin

Departamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Septiembre de 2024

Índice

1. Analisis del problema	2
2. Esquema de tareas	3
3. Problemas de desarrollo que presentó	4
4. Evolucion del algoritmo	5

1. Analisis del problema

Se solicita un algoritmo capaz de detectar que tipo de señal se vería, la frecuencia y la amplitud, partiendo de los valores recibidos por medio de la entrada analógica, todo esto proveniente de un generador de señales. Aparte de hacer las conexiones necesarias del Arduino el LCD, generador de señales y dos pulsadores, debemos implementar el código de c++ con el que venimos experimentando para que funcione en Arduino.

La solución propuesta primero montar todo lo necesario en tinkercad, que son conectar el LCD, generador de señales y comprobar que ambos pulsadores cumplen su respectiva función de apagar/prender el generador de señales y mostrar datos en el LCD. Una vez listo el sistema en tinkercad empezar a experimentar con el generador de señales para poder observar que patrón de comportamiento tiene frente a diferentes valores dados previamente. Frente al análisis de los patrones de comportamiento empezar a mirar como se relacionan matemáticamente los máximos y mínimos con la amplitud, cuántos valores se reciben en cierto lapso de tiempo y saber si siempre es un valor constante, en base a si el de valores recibidos en cierto lapso de tiempo es constante podemos calcular la frecuencia viendo cuanto tarda en llegar a su pico máximo, luego de hallar frecuencia y magnitud experimentar con que valores la señal es irreconocible (patrón fuera de los propuestos en el problema) para plantear los límites de reconocimiento, por último buscar una relación con los máximos, mínimos y valores capturados en el arreglo con la forma de la señal

2. Esquema de tareas

1 - Montar el circuito con el LCD, generador de funciones y los dos pulsadores con todo debidamente conectado.

2 - Empezar el informe del parcial e inicializar el repositorio.

3 - Una función que reciba una variable tipo puntero que apunta hacia un arreglo en memoria dinámica, otra variable tipo entero que contiene la longitud a tomar(cuantos valores tomados en cierto lapso de tiempo a través del pin analógico del Arduino) y nos llene este arreglo dinámico con los valores tomados a través del pin analógico.

4 - Una función para hallar el punto medio por así decir, que vendría siendo, según el desfase de cc de la señal saber en que valor analógico se encuentra el punto medio. Esto nos ayudara para hallar la amplitud.

5 - Una función para calcular la amplitud dado un puntero hacia un arreglo en la memoria dinámica que contiene los valores capturados en cierto lapso de tiempo y la cantidad de valores capturados. Recorremos este arreglo buscando el máximo y el mínimo, una vez hallados, la amplitud seria igual a la diferencia de estos dos dividido dos $((\text{Valor máximo} - \text{valor mínimo})/2)$

6 - Una función para hallar el intervalo de muestra, es decir el tiempo que toma Arduino en almacenar un valor en nuestro arreglo, esta nos ayudara para calcular la frecuencia.

7 - Una función para hallar la frecuencia.

8 - Una función para saber que tipo de onda es.

9 - Terminar de organizar el informe.

9 - Grabar el video explicativo.

3. Problemas de desarrollo que presentó

El primer inconveniente al comenzar con la elaboración del proyecto fue el como configurar los pulsadores y que el generador de funciones enviara valores continuamente a través del pin analógico, ya que estos al ser presionados toman un valor de 1 solo por un pequeño instante, luego vuelven a 0, por lo cual tuve que implementar un booleano y una variable donde guardar el último estado del botón para que funcionará

El segundo inconveniente fue que el programa no terminaba su ejecución al invocar la función donde iba a almacenar todos los valores leídos desde el pin analógico en un arreglo ubicado en la memoria dinámica pero se soluciono disminuyendo el tamaño del arreglo, aunque al estar trabajando directamente sobre tinkercad no sabia como depurar por lo cual me tarde un poco en saber que era por falta de memoria que no podía utilizar un arreglo de tamaño 1000.

El tercer inconveniente fue que a la hora de calcular la amplitud, cuando tenia que cambiarla a voltios, si era mayor a 6V ya no era tan efectiva la formula utilizada y entre mayor el valor a 6V mas se aleja la aproximación, para valores menores a 6V funciona muy bien.

El cuarto inconveniente es que para calcular la frecuencia necesitábamos tener ciertas condiciones de control del tiempo, al menos conocer cuanto tiempo tarda el arreglo en llenarse y en base a eso hallar la frecuencia con una formula de tiempo

El quinto inconveniente fue que tuve un problema con la solución que tenia pensada para la identificación ya que mi resultado partía de comparar y en este caso no se puede comparar, solo se nos dan los datos de un tipo de onda en el momento, por lo cual mirando en el arreglo de la primeraDerivada con diferentes frecuencias y voltios vi que era mas fácil identificar aquí los patrones de la señal, por lo cual en base a varios ensayos pude encontrar un método no muy preciso pero que nos da el resultado en algunos casos de que tipo de onda es.

El sexto inconveniente fue que cuando me disponía a eliminar los prints que utilizaba para ver la señal como se comportaba me di cuenta que cometí un gran problema y era que al tener este print el arreglo no se llenaba con la misma velocidad con la cual se llena cuando tengo el print en el lugar que lo tenia, esto influyo un poco en el desarrollo del algoritmo, ya que si hubiera dado con ese detalle antes creo que el desarrollo hubiera sido mas fácil y preciso pero ahora no dispongo del tiempo suficiente para hacerlo de ese modo, tendría que cambiar casi todo el código y no puedo quitar los "serial.print" de los valores recibidos analógicamente ya que como las funciones están hechas en base a los valores recibidos cuando estaba el print al quitarlo los valores cambian bruscamente a la hora de compararlos, por lo cual los criterios utilizados ya no funcionarían.

4. Evolucion del algoritmo

1er Avance: Se implemento el modelo de Arduino con sus respectivas conexiones a un generador de señales, pantalla lcd y dos pulsadores. Falta configurar los pulsadores.

2do Avance: Se configuraron correctamente todos los componentes, ahora solo faltan las funciones respectivas para hallar los valores correspondientes.

3er Avance: Se hizo la función que inicializa un puntero en memoria dinámica con un arreglo que contiene 500 valores tomados a través del pin analógico luego de ser pulsado el segundo pulsador.

4to Avance: Se hizo la función que hallaba el valor medio, es decir el punto medio entre el máximo y el mínimo, para así poder hallar la amplitud en base al valor medio, ya que al restar el valor máximo menos el valor medio se obtiene un aproximado de la amplitud pico, esta al multiplicarse por dos nos da un aproximado de la amplitud pico a pico y ya con una formula pasamos a voltaje.

5to Avance: Se hizo la función que hallaba la frecuencia. En este punto me di cuenta que tenia un problema y era que no tenia contabilizado el tiempo en ningún momento y para la formula de la frecuencia que había encontrado necesitaba saber cuanto tiempo tardaba en tomar los 500 valores de muestra que estaba tomando luego de que el usuario presionara el segundo pulsador, para esto me toco devolverme a la función de toma de muestras (valores recibidos a través del pin analógico) y utilizar la función micros() la cual funciona como un contabilizador en microsegundos, utilizando esta llegue al tiempo total que tomaba en llenar el arreglo, este tiempo total se divide por el numero de muestras para hallar el promedio que toma Arduino en almacenar cada valor en el arreglo. En base a estos valores hayamos la frecuencia con una formula encontrada.

6to Avance: Se empezó la función de ver que tipo de onda es, la mas fácil era la cuadrada ya que solo tomaba 2 valores, la cosa se complica para hallar la senoidal y la triangular ya que ambas tienen valores muy similares pero investigando di que podía calcular la primera derivada y la segunda para ver que cambios habían, y así hallar un patrón, la derivada seria el cambio de un valor a otro, es decir crear 1 arreglo que me calcule $\text{primeraDerivada}[0] = \text{arreglo}[1] - \text{arreglo}[0]$ así sucesivamente, tambien aplica lo mismo para la segunda derivada solo que ya se aplica sobre la primera. Imprimiendo este patron en pantalla pude notar que en la segunda derivada cuando la señal es triangular el maximo del arreglo de la segundaDerivada es mayor a cuando la señal es senoidal (Esto con mismos atributos de frecuencia, amplitud y desfase), aun me falta implementar los condicionales para eso y otra funcion para saber si la señal es desconocida.

7mo Avance: Se hizo la funcion para identificar la onda en base a los ensayos con diferentes valores y que se veian reflejados en el arreglo de la primeraDerivada, pese a que el metodo no es muy preciso nos da un resultado correcto en muchos casos.