

**INFORME DESAFÍO I INFORMATICA II**

Jaider Bedoya Carmona

Wilbergt Alexander Osorio Trespalacios

Informe sobre proceso del desafío uno

PROFESORES

Augusto Salazar

Aníbal Guerra

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería Electrónica

Medellín

2024

**Proceso de análisis**

Como primer análisis tenemos que cada tipo de onda cuando cruza el 0 se comporta de forma distinta, esto nos puede servir para determinar con que naturaleza asciende o desciende la onda. El plan a seguir que desarrollaremos mañana será tomar valores cercanos al cero de cada onda y ver como se comporta el aumento de disminución del dato de entrada, ya que determinamos una forma característica de cambiar para cada onda, por ejemplo, la onda cuadrada, tiene un cambió abrupto, pero, por otro lado, en la onda triangular, su cambio es lineal, y la onda senoidal tiene un cambio más suave.

La razón de cambio de la onda senoidal es mayor a la de la onda triangular, por lo que en la onda senoidal su cambió es exponencial, mientras que en la triangular hay un cambio más lineal, con respecto a la onda cuadrada sabemos que sus valores son binarios, solo tiene un valor máximo y uno mínimo. Encontramos diferencias en las ondas en que entran al cero de forma parecida pero los valores cercanos a sus puntas o valles son diferentes.

Encontramos que en tinkercad la gráfica simulada de las funciones es algo irregular por lo que es difícil encontrar patrones que siempre se repitan. Sabemos que su amplitud, también como sus picos deben ser iguales en al menos un segundo para encontrar su periodo y por lo tanto su frecuencia.

Descubrimos que los datos pueden ser tomados en un intervalo de tiempo bastante bajo por lo que al tomar datos por el puerto análogo a una velocidad mas alta podemos encontrar cifras más precias, tal vez tomándolos en un arreglo.

Miércoles (11 sep) en la noche tenemos algunas conclusiones:

1. Empezamos midiendo la amplitud de onda
2. El tiempo entre punta y punta es el periodo
3. 1/periodo es la frecuencia
4. Crear arreglo de tamaño cuatro veces amplitud
5. Guardar datos desde punta a punta
6. Hallar razón de cambio
7. Decidir que señal es

Tenemos que el tiempo transcurrido entre un pico y otro es el periodo, viendo cuantos periodos ocurren en un segundo podemos calcular la frecuencia. Pensamos en crear un ciclo que mide cuantas veces llega al valor máximo la función en un segundo, el valor máximo ya fue tomado anteriormente como el punto medio y la amplitud.

Amplitud será igual a el valor máximo tomado menos el valor mínimo tomado todo divido entre dos.

Viernes (13 sep), decidimos que nuestra estrategia principal será la siguiente:

Tomar los datos de entrada, durante un segundo estos datos se guardarán en un arreglo, de aquí ya con otras estrategias podemos inferir los datos que necesitamos, ya que en este arreglo tendremos, valores máximos, valores mínimos, punto medio, y ya que se midió en un segundo, podemos saber cuál será la frecuencia.

Cabe resaltar que en algunos casos el tipo de dato nos afecta la información ya que, si en algún momento el dato que estemos tratando es menor que uno y mayor a cero, no se puede tratar como un entero porque sería igual a cero, debe ser tratado como un float para poder tener su valor, también como lo son los negativos, deben tener su signo y los positivos deben ser sin signo.

Hemos aprendido a que el Arduino tiene una memoria limitada, intentamos crear un arreglo de 1000 datos, pero era justo el tope de memoria del Arduino, por lo que fue más fácil tomar un dato cada 2 milisegundos por un segundo, para tener 500 datos para poder inferir la información que necesitamos

Lunes (16 sep) ya que mañana es el día de entrega, tenemos una solución pero también algunos problemas que nos complican nuestra implementación aunque somos conscientes de ellos, hoy en nuestro desarrollo de la implementación conseguimos diferenciar cada tipo de onda pero con muchos errores, por el flujo del programa que puede afectar el tiempo y las operaciones de los datos, tenemos un serio problema con la ejecución en el tiempo ya que a veces si y a veces no, toma bien el tiempo durante la ejecución.

El uso del arreglo dinámico es un problema ya que no conseguimos implementarlo como pensábamos ya que con el problema del tiempo se toman datos diferentes cada vez que se ejecuta por lo que resulta muchas veces en espacios basura en el arreglo o un desbordamiento, por lo que hace fallar el programa a menudo ya que esto afecta directamente cuando recorremos el arreglo y tomamos decisiones.

Usar ciclos como el while afectan el flujo del sistema ya que está dentro de una función loop, cuando entra al while debe frenar la ejecución del loop mientras realiza las tareas dentro del while, lo que significa que el tiempo de ejecución del loop pausa hasta que el while sea roto y continue el resto de tareas en el loop.

Para agregar sabemos que falla ya que el colapso de memoria a veces se presenta más que todo cuando toca medir el segundo y los datos que ingresan son variables, es decir, que pued que recolecte mas datos o menos cada vez que se cuenta el segundo

Nuestro programa tiene un rango de funcionamiento donde alcanza a distinguir entre cada onda, primero, entre las tipo senoidal y la triangular, las diferencia en frecuencias bajas entre 1 y 2Hz, aún detectando la frecuencia y amplitud, hasta los 100 Hz sigue diferenciando los otros tipos de onda de la cuadrada, con su frecuencia y amplitud, después no alcanza a distinguir el tipo de onda y no siempre la amplitud y frecuencia.

**Problemas**

La onda senoidal se comporta de manera parecida a la onda triangular en cuanto a su paso por el 0.

Cuando la frecuencia es alta, es difícil capturar los valores ya que la entrada de estos no es regular y es complicado encontrar un patrón que nos ayude a encontrar los datos necesarios.

Cuando se toman los datos de entrada por el puerto análogo, se toman como números enteros, haciendo pruebas encontramos que al imprimir otros datos que no sean estos valores, esto interrumpe la generación de la onda y la daña.

En el ciclo que creamos para contar el tiempo, se cuenta las iteraciones dentro de ese bucle y nos afecta el tiempo que se cuenta en el loop principal.  
tenemos un serio problema con la ejecución en el tiempo ya que a veces si y a veces no, toma bien el tiempo durante la ejecución

Tomar los datos en un lapso de tiempo muy amplio o muy corto nos generará problemas, ya que en uno muy amplio es complicado encontrar diferencias, y en uno muy tenemos muchos datos y se repiten varios.

Por falta de tiempo y de análisis en el proceso hemos cometido muchos errores en la implementación de nuestras ideas y consideraciones para el funcionamiento del programa, ya que hay porciones de código y de lógica que no marcharon como lo pensamos en un inicio, sin embargo, estamos conscientes de los errores y de los puntos que tenemos que mejorar y tener en cuenta.

Usar ciclos como el while afectan la ejecución en tiempo del sistema ya que está dentro de un void loop, cuando se entra al while, para la ejecución de este debe frenar la ejecución del loop, y no con

**Algoritmos implementados**

El primer algoritmo desarrollado fue en el condicional encendido, donde se busca hallar el valor máximo y mínimo que toma el valor leído por el pin análogo del generador de señales, con el fin de que con el valor máximo y mínimo hallar la diferencia y dividirlo entre dos que seria la distancia de pico a valor promedio.

Luego el siguiente algoritmo tiene cuerpo en el condicional ciclo, en el cual se inicia un contador por un segundo, en el cual durante ese tiempo se guardan datos leidos de la señal en un arreglo previamente reservado para luego que se recolecten hallar características de la onda.

Luego dentro de un bucle for que recorra todo el arreglo vemos en que momento los valores cambian entre positivo y negativo con un par de booleanos que marcan el estado, así para aumentar un contador de cruces por ceros que con el análisis hecho previamente notamos que cada dos cruces por ceros es un ciclo, con este contador luego lo dividimos entre dos para hallar la frecuencia.

También dentro del mismo bucle for nombramos una variable llamada determinante que se encarga de notar cuando los valores leídos se repiten, esto fue parte del análisis notar que en una escala desde el mayor al mínimo la onda cuadrada seria la primera que mas repite valores, ya que oscila entre únicamente dos, seguido de esto seria la sinusoidal que se repite en sus picos para lograr el efecto de curva y finalmente la que menos se repite es la triangular ya que esta simula un efecto de vértice que en un caso ideal seria únicamente un valor.

Finalmente, con la variable media que fue sumando dentro del bucle for cada dato recolectado de la señal, la dividimos por el numero de datos para hallar el valor promedio y ver si cuenta como cíclica ya que las ondas cuadradas, sinusoidales y triangulares tienen la característica en común que la misma distancia que sube es la misma que baja, por lo cual su media es cero (en un caso ideal).

Con ayuda de la variable determinante y la media podemos diferenciar entre tipos de ondas.

Para finalizar, todas las variables se reinician y se vuelve a iniciar el programa para seguir con la lógica de recolección, análisis e impresión de datos.

