**Informe del Desafío: Entrega final - (Simón Valencia soto).**

**Análisis del Problema y Consideraciones para la Alternativa de Solución Propuesta**

El desafío me puso a consideración de varios retos, los cuales analíticamente se cumplieron, pero que mediante el desarrollo, la codificación y manipulación de los datos se dificultaron.

Digo codificación pero en realidad solo fue la manipulación de la aleatoriedad de los datos que se recibía desde el generador de señales lo que dificultaba la codificación.

El primer problema del reto a considerar era el proceso de guardar los datos que se recibían, y la alternativa de solución fue un arreglo dinámico en el que se definió un tamaño de 100 espacios en la memoria del heap, definí un tamaño específico dado que cuando se alcanzaba el límite del arreglo, los valores nuevos se sobrescribían sobre los valores antiguos los cuales no necesitaba más, y donde al final estos valores sobrescritos se podían liberar con la impresión del botón que para la adquisición de datos. Esta estrategia permite que el algoritmo y la simulación nunca paren de recibir datos hasta que sea oprimido el botón que para la adquisición.

Después de esto realizado, procedo a definir cuál es la amplitud y cuál es la frecuencia de las ondas en general. Realizar esto se me dificultó por el manejo de los datos aleatorios (incluso sin delay) que se guardaban, era casi imposible que se guardara un cero en el arreglo cada vez que la función x pasara por el mismo. Cuando digo aleatorios me refiero a que no son continuos: (1, 2, 3, 4, 5, 6) Sino que se guardan por intervalos: (2, 5, 7, 20, 26, 43…A) hasta llegar a la amplitud o valor máximo; supongo que esto pasa por la cantidad de datos que la plataforma debe procesar al mismo tiempo que imprimir como onda y números en el monitor serial, y también guardar en el arreglo. La alternativa de solución fue la misma para los 3 tipos de señal y es que se cada señal va a tener un máximo referente a la amplitud el cual se va a guardar después de que ciertas veces la señal llegue a ese mismo máximo, donde al final ese máximo se dividirá por 100 y tener una amplitud en Volts, aunque también debo tener en cuenta que la amplitud máxima que puede recibir el microprocesador de Arduino es de 5V, y el offset estando en 2.5V, lo que significa que el valor máximo entregado por el generador es de 1023. La frecuencia fue más costosa pero no muy diferente y es que voy a detectar cada vez que la señal cambie de un valor negativo a uno positivo que es la alternativa a evaluar cruces por cero, haciendo esto puedo definir la frecuencia como el número de veces que pasa esto dividido entre el tiempo, el tiempo es tomado por la función “ millis( ) ” y dado que es un intervalo, el tiempo será la diferencia entre el tiempo final y el inicial, este tiempo lo divido entre mil dado que son milisegundos, y posteriormente procedo a dividir el número de cruces entre el tiempo, obteniendo la frecuencia de todas las señales.

Después de haber tenido esto realizado, codificado y muestreado se van a definir los tres tipos de señales. Para la señal cuadrada implemento una condición de cambios bruscos que dice que si el valor absoluto de la diferencia entre un valor actual y un valor anterior es mayor que cien (cien debido a la amplitud mínima) entonces habrá un cambio brusco y si lo hay y se presentan valores constantes o repetidos en la señal, entonces se va a tener una señal cuadrada. Para la señal analógica y triangular se implementa un proceso algo un poco más robusto y es que primero tomo en cuenta que la señal analógica nunca tiene cambios realmente bruscos, sino que es una onda suave con valores distintos a medida que transcurre el tiempo, por eso quise evaluar el cambio en la pendiente de la señal analógica, donde analíticamente se deduce que debe ser un cambio suave. Definí la pendiente generalmente como la diferencia de un valor mayor y otro valor menor, en este caso el que me interesa es el valor del máximo como el valor mayor, y un valor continuo al máximo como menor, una vez tenida esta diferencia, se debe tener la certeza de que el cambio en la pendiente en la cresta de la señal va a ser menos brusco en la analógica que en la triangular, pero dado a que los valores son tan aleatorios tuve que hacer un proceso de muestreo de los datos y de su pendiente cuando llegan al máximo, y por estadística noté que en todas las ocasiones o sea 10 de 10 veces, en la señal analógica la pendiente en la cresta siempre va a ser menor que dos veces el valor de la frecuencia, y que en la señal triangular la pendiente en la cresta siempre va a ser mayor que dos veces la frecuencia; Luego, si nada de esto pasa se tendrá como argumento que será una señal desconocida. He implementado apuntadores en los parámetros de las señales, variables booleanas y variables estáticas, Y una vez que se oprima el pulsador de detención de adquisición entonces se va a liberar la memoria, y luego, se asignará de nuevo memoria a las variables y arreglo dinámicos por si posteriormente se oprime de nuevo el pulsador de adquisición, pero antes de eso se imprimirá por pantalla el tipo de onda, su frecuencia y amplitud, esto usando la librería sugerida para imprimir por la pantalla del LCD.

De esta manera desarrollé el desafío número uno, un gran reto dado que el tema de tratar con el arduino me cogió frio pero pese a ello y de pronto a las fallas, pude aprender bastante.

Gracias.

Imagen de las conexiones en Tinkercad.

