



Universidad Veracruzana

**PROTOTIPO DE
SOFTWARE EDUCATIVO**



“Simulador de Señales Determinísticas y Fisiológicas”

AUTORES:

**M. EN I. B. LUIS JULIÁN VARELA LARA
DR. PABLO SAMUEL LUNA LOZANO**

**INGENIERÍA EN INSTRUMENTACIÓN ELECTRÓNICA
FACULTAD DE INSTRUMENTACIÓN ELECTRÓNICA
Septiembre 2021**

1. Descripción General.

Este Prototipo de Software Educativo que se ha denominado: “Simulador de Señales Determinísticas y Fisiológicas” es un módulo de software, cuya funcionalidad permite:

- a) Transferir información de señales digitalizadas como lo haría un hardware de adquisición de señales conectado a un puerto de comunicación serial RS-232. Con este software se elimina la necesidad de contar con hardware como tarjetas de adquisición de datos, generadores de funciones, fuentes de alimentación, circuitos electrónicos para obtención de señales fisiológicas (amplificadores, sensores, entre otros).
- b) El alumno aprenderá a desarrollar software que reciba/grafique/almacene/procese las señales que reciba desde el simulador, lo cual lo haría de manera idéntica que si tuviera todo el hardware que genere las mismas señales.
- c) Simular señales determinísticas (sinusoidales, triangulares y cuadradas), así como fisiológicas (ECG, EMG). Las primeras sirven como referencia al estudiante, al saber su forma, frecuencia y amplitud, el alumno puede comprobar con relativa facilidad que la señal que está observando en el software que esté desarrollando, corresponde a la que se está enviando desde el simulador. Las segundas, las fisiológicas, permitirán llevar a cabo el desarrollo de algoritmos de procesamiento para extracción de información.
- d) Contar con secciones que presentan ejercicios sugeridos en torno a su uso, y con los cuales el estudiante podría mejorar sus habilidades en el desarrollo de aplicaciones de software para computadora personal, sin requerir hardware adicional a la computadora para su prueba.

Como profesor de las experiencias educativas de Instrumentación Biomédica (Tópicos Avanzados de Ingeniería en Instrumentación Electrónica II) del programa educativo de Ingeniería en Instrumentación Electrónica, así como de Programación Computacional Avanzada del programa educativo de Ingeniería Biomédica, he identificado la necesidad de que los alumnos puedan contar con una herramienta a la medida que pueda servir como medio de experimentación y prueba de algoritmos computacionales para: adquirir/graficar/almacenar/procesar señales, tanto señales de prueba con las cuales comprobar sus resultados con relativa facilidad (señales determinísticas), como señales reales (fisiológicas) de las que requeriría de hardware especializado para poder observarlas, pero en esta etapa donde lo importante dentro de la experiencia educativa no es el desarrollo de hardware, el contar con este simulador facilita la experimentación con este tipo de señales tal como si se contara con el hardware especializado.

2. Datos técnicos.

El software fue desarrollado bajo el entorno de programación de Visual C# de Microsoft Visual Studio Community 2019, consta de un solo archivo ejecutable: “**Simulador Señales Determinísticas y Fisiológicas.exe**”; este entorno de programación es gratuito para uso académico, los archivos ejecutables son compatible con sistemas operativos Microsoft Windows

Vista, 7 y 10, sin requerimientos especiales de espacio de disco duro ni memoria. Se requiere adicionalmente de una aplicación de creación de puertos seriales virtuales, de las que existen varias compañías que ofrecen versiones gratuitas o de prueba, más adelante se detalla el uso de una en específico.

El archivo ejecutable **“Simulador Señales Determinísticas y Fisiológicas.exe”**, está diseñado para operar en conjunto con aplicaciones que se estén desarrollando como parte de los procesos de aprendizaje en experiencias educativas en temas cuya finalidad sea el desarrollo de aplicaciones de software en torno a la adquisición, almacenamiento, graficado y y/o procesamiento de señales. La comunicación entre esta herramienta de software y la aplicación que se esté desarrollándose realiza por medio de puertos seriales virtuales, mismos que deben ser activados en una computadora personal con ayuda de herramientas de software libre como la que se describe más adelante.

3. Introducción.

El uso del prototipo de software educativo **“Simulador Señales Determinísticas y Fisiológicas.exe”** sustituye la necesidad de contar con hardware externo (tarjetas de adquisición de datos, generadores de funciones, circuitos electrónicos con amplificadores y sensores para obtener señales fisiológicas), y poder recibir la misma información, pero a través de un software de simulación.

Un ejemplo en el cual se muestra un sistema completo requerido para el desarrollo de aplicaciones para una computadora personal relacionadas con diversos procedimientos que ha de requerir de una señal, se muestra en la figura 1.

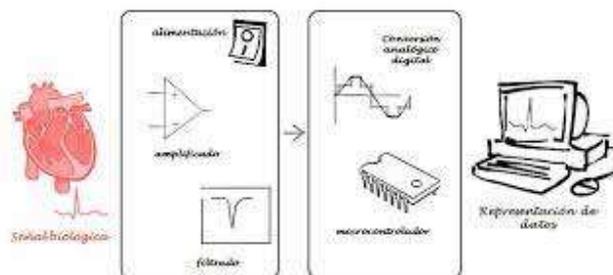


Figura 1. Ejemplo de un sistema de adquisición y graficado de señales eléctricas del corazón.

Este ejemplo se trata de la señal eléctrica proveniente del corazón, la cual típicamente se grafica (electrocardiografía o ECG) y es una herramienta de diagnóstico útil para los profesionales de la salud. Esta señal puede requerir ser almacenada, o procesada para obtener información de manera automática. Si se observa la figura 1 se podrá identificar que se requieren varios elementos antes de llegar hasta una computadora personal y llevar a cabo los procesos de graficado, almacenamiento y/o procesamiento, elementos tales como:

- a) Un sujeto de pruebas desde el cual obtener la señal eléctrica del corazón
- b) Un sensor con el cual obtener la señal (electrodos y cables adecuados para tomar la señal desde el cuerpo).
- c) Un amplificador de señales con características adecuadas para obtener una señal de amplitud y calidad adecuada.
- d) Un dispositivo que lleve a cabo una conversión analógica a digital.
- e) Un puerto de comunicación entre el hardware de conversión analógico a digital y la computadora personal, para hacer la entrega de las muestras digitales de la señal.

Sin embargo, si dentro de los objetivos de aprendizaje de algunas experiencias educativas, las habilidades que se quieren mejorar en el alumno están más bien enfocadas en ejercitar su habilidad en desarrollo de aplicaciones de software para la adquisición, almacenamiento, graficado y/o procesamiento de señales, la necesidad de contar con todos estos elementos adicionales dificulta el hecho de que los estudiantes puedan experimentar en su desarrollo.

Así entonces el software de simulación **“Simulador Señales Determinísticas y Fisiológicas”** que aquí se describe permite reemplazar la necesidad de contar con todos estos elementos y enfocar los esfuerzos del alumno en el desarrollo de la aplicación que ha de ejecutarse en la computadora personal, con la seguridad de que su funcionalidad será idéntica a si se contara con los elementos externos adicionales.

4. Guía de uso.

En esta sección se presenta la forma de utilizar el módulo de software desarrollado y la forma de comunicarlo con una aplicación que se este desarrollando para la adquisición, graficado, almacenamiento, y/o procesamiento de señales.

3.1 Comunicación serial entre dos módulos de software

En primer lugar, debemos explicar que dos módulos de software pueden intercambiar información entre ellos por ejemplo como en este caso a través de “puertos seriales virtuales”, en un esquema como el mostrado en la figura 2.

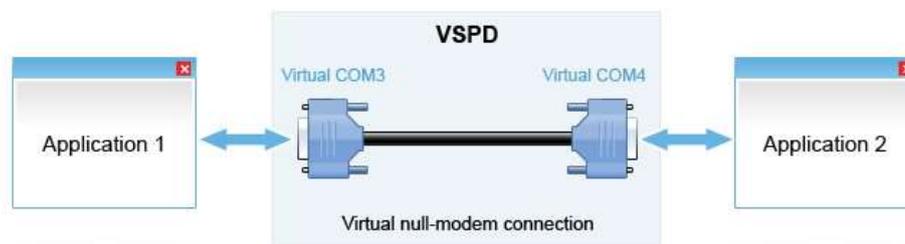


Figura 2. Configuración de puertos virtuales necesaria

De la figura 2 debemos entender lo siguiente: la aplicación 1, será el módulo “Simulador de Señales Determinísticas y Fisiológicas”, y la aplicación 2, será el software que el alumno esté desarrollando para procesos de adquisición, graficado, almacenado y/o procesamiento de señales. En el medio, vemos un par de puertos seriales identificados como COMN, que en este ejemplo están marcados como COM3 y COM4, y que están virtualmente creados e interconectados, de tal forma que se pueda transferir información de COM3 a COM4, o de COM4 a COM3, es decir desde la aplicación 1 a la aplicación 2, y de la aplicación 2 a la aplicación 1, lo que se conoce como comunicación bidireccional, al unir los puertos COM3 y COM4 en una conexión llamada NULL-MODEM.

Creados este par de puertos virtuales, y su interconexión, la aplicación 1 deberá “abrir” para sí el puerto COM3, y la aplicación 2 deberá “abrir” para sí el puerto COM4, y así lo que se envíe o reciba la aplicación 1, será lo recibido o lo enviado por la aplicación 2 respectivamente.

Para crear estos puertos virtuales existen diversas aplicaciones comerciales y gratuitas, como la llamada “HHD Virtual Serial Ports Tools”, que, tras ser instalada y ejecutada se mostrará una ventana como la mostrada en la figura 3, en la que permite llevar a cabo la creación de los puertos virtuales que requerimos

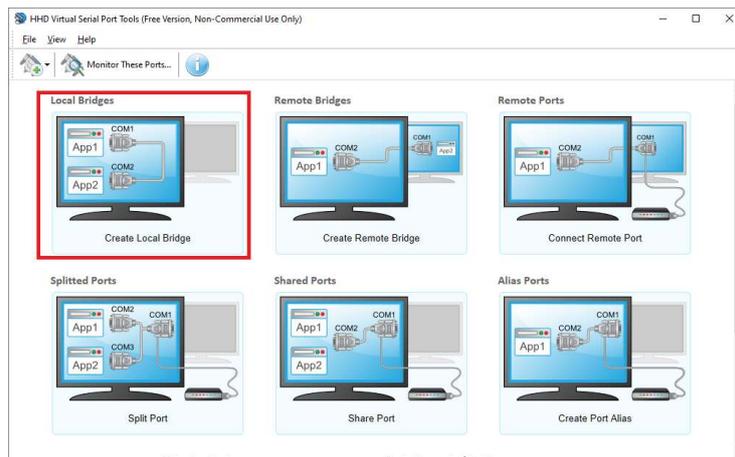


Figura 3. Ventana de configuraciones para la creación de puertos virtuales

Al dar clic sobre la opción remarcada en rojo en la figura 3, aparecerá una ventana (figura 4) donde nos muestra el par de puertos virtuales a crear en conexión Null-Modem.

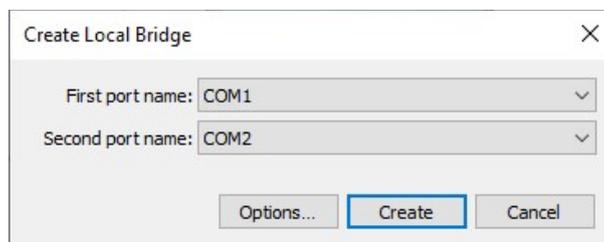


Figura 4. Selección de puertos a crear

Tras dar clic en el botón crear en la ventana mostrada en la figura 4, se abrirá una ventana como la mostrada en la figura 5. Aquí cabe destacar que esta aplicación puede ser utilizada de forma gratuita, pero con restricciones, una de ellas es que uno no puede elegir los nombres de los puertos virtuales a crear, y por lo tanto habrá que utilizar los que la aplicación nos ofrezca por default. Considere además que en las ventanas que la aplicación nos mostrará, nos estará indicando cuales opciones están disponibles en la versión gratuita, como en la figura 5.



Figura 5. Advirtiendo que en uso libre están restringidas algunas opciones.

Una vez hecha la creación en la ventana principal de esta aplicación se mostrará que puertos virtuales están creados, como se observa en la parte remarcada en rojo de la figura 6.

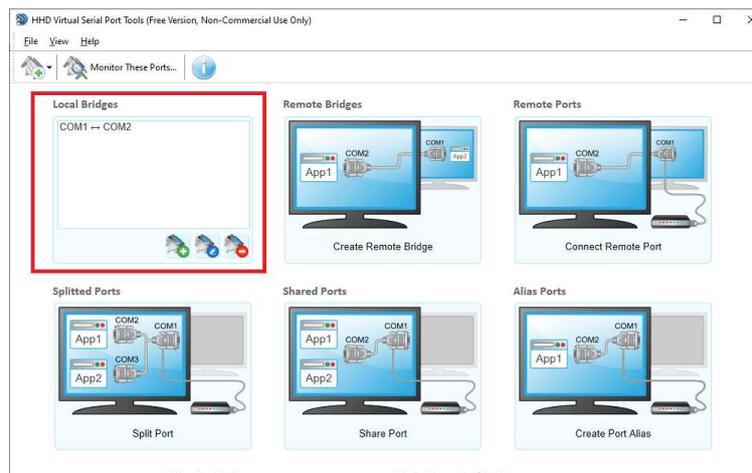


Figura 6. Aplicación mostrando el par de puertos virtuales generados.

Una vez configurado el par de puertos seriales virtuales, podemos cerrar la aplicación y esta configuración se conservará mientras no reiniciemos nuestra PC, que es otra de las restricciones de esta aplicación en su versión gratuita, por lo que cada vez que se reinicie la computadora, habrá que rehacer la creación de los puertos virtuales.

La correcta creación de los puertos virtuales además puede ser verificada desde el administrador de dispositivos de Windows, donde tras haber realizado el proceso de creación de los puertos virtuales observaremos que en la parte de puertos COM nos muestra la información como en la figura 7.

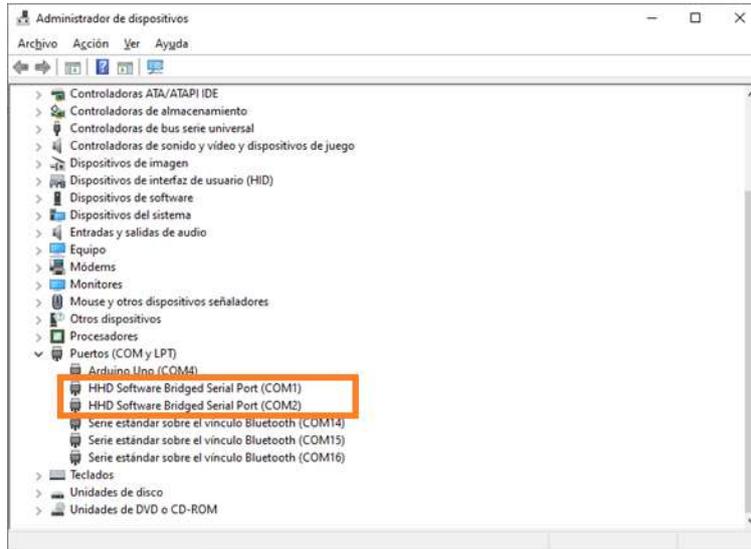


Figura 7. Visualizando los puertos virtuales generados

Una vez creado el puente de comunicación virtual entre las dos aplicaciones, podemos entonces comenzar a utilizar el “Simulador de Señales Determinísticas y Fisiológicas”.

3.2 Ejemplos de uso

Una vez que se cuenta con la conexión NULL-MODEM entre dos puertos seriales, se ejecuta la aplicación “Simulador de Señales Determinísticas y Fisiológicas” y se mostrará una ventana como la que se muestra en la figura 8.

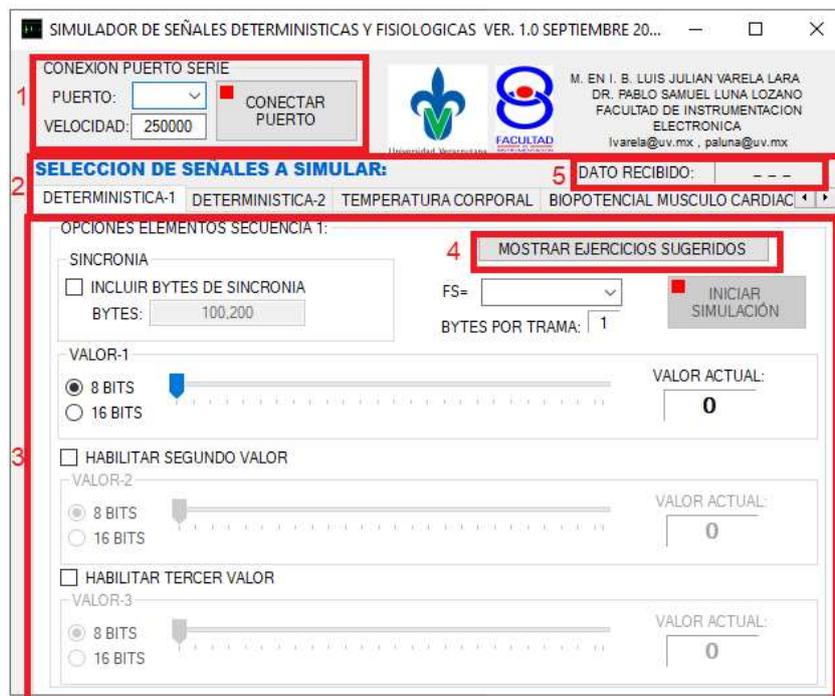


Figura 8. Pantalla inicial del software “Simulador de Señales Determinísticas y Fisiológicas”

En la figura 8 anterior podemos observar 54 regiones marcadas en donde cada una tiene las siguientes funcionalidades:

1. En esta sección se puede seleccionar el número de puerto COM a elegir para comunicarse con la aplicación que se esté desarrollando, recordando que será uno de los dos puertos virtuales en conexión NULL-MODEM que se hayan creado.
2. En esta región se puede seleccionar la región a presentar en pantalla para mostrar las opciones a configurar entre los distintos tipos de señales determinísticas o fisiológicas que se pueden simular.
3. La zona marcada con el número 3 en la figura 8, muestra las opciones de configuración que el usuario puede manejar cuando de entre las diferentes tipos de señales que se pueden elegir se ha tomado la de “DETERMINISTICA-1” sin embargo, en el caso de que se elija otra de las opciones, en esta misma zona aparecerán las opciones de configuración asociadas a cada tipo de señal, como en la figura 9, se muestra como se ve la aplicación al seleccionar “SEÑAL ELECTRICA CORAZON”.



Figura 9. Pantalla de la aplicación al seleccionar la opción “BIOPOTENCIAL MUSCULO CARDIACO”

4. La región marcada en la figura 8 la región con el número 4, es un botón que, al dar clic sobre él, muestra una ventana como la de la figura 10, donde se describen algunos ejercicios sugeridos para practicar con la adquisición, almacenamiento, graficado y/o procesamiento de señales relacionados con el tipo de señal que esté seleccionado.

5. Finalmente la región 5 de la figura 8, es una zona donde se visualizará el valor numérico de algún byte que sea recibido por la aplicación, que puede servir para depurar código en ejercicios que requieran hacer envíos de datos por medio del puerto serial hacia el simulador.

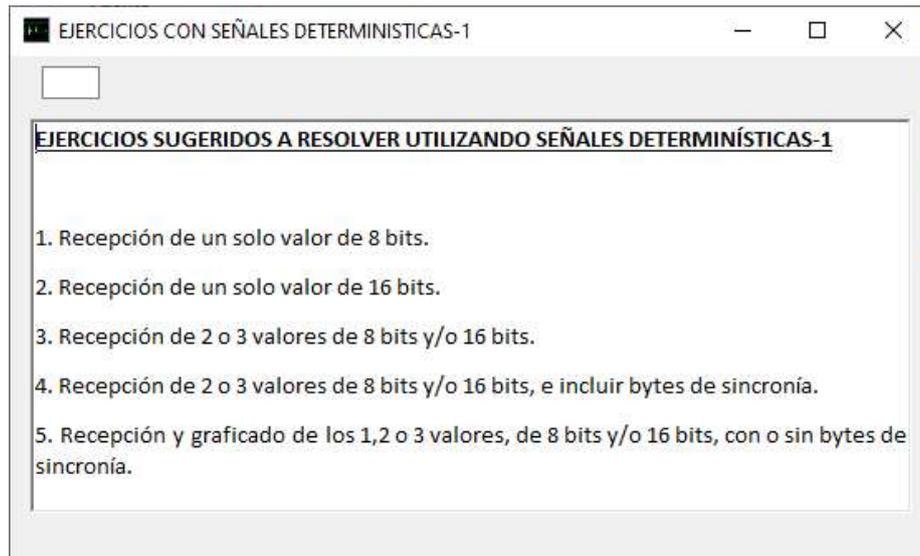


Figura 10. Ventana de ejercicios sugeridos para la sección actual de "DETERMINISTICA-1".

A continuación, se dará una serie de ejemplos de como debieran llevarse a cabo algunas configuraciones en el Simulador de acuerdo con lo que en cada caso se pide.

Ejemplo1: Simular la generación de dos señales determinísticas, una onda sinusoidal de 2.5Hz digitalizada a 16 bits, con una amplitud que abarca el 50% del rango de conversión, y una triangular de 4Hz digitalizada a 8 bits, con una amplitud que abarque el 90% de la escala de conversión, ambas muestreadas a una frecuencia de muestreo de 50Hz.

Para resolver este ejercicio, se puede hacer uso de una sola de las secciones de la opción "DETERMINÍSTICA-2" del software, ya que ambas señales se muestrearán a la misma frecuencia y son solo dos señales de hasta tres que se pueden agregar en cada sección, y se deberán hacer las siguientes configuraciones:

a) Como se muestra en la figura 11, se selecciona la opción de señal sinusoidal, se especifica su frecuencia (2.5Hz), se marca la opción "16 BITS" y se ajusta el porcentaje de amplitud respecto a la escala máxima de los 16 bits al 50% tal como lo pide el ejercicio y se clic al botón "AGREGAR"

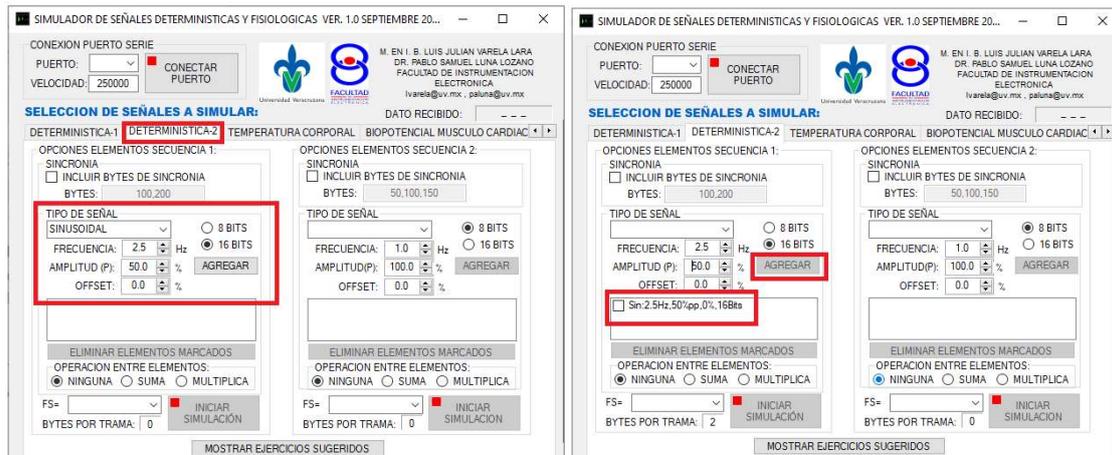


Figura 11. Configurando la generación de una señal sinusoidal digitalizada a 16 bits, de frecuencia .5Hz y abarcando el 50% de la escala de conversión. A la izquierda pantalla donde se configuran los parámetros de la señal a generar, a la derecha una vez habiendo dado clic al botón “AGREGAR”.

b) Se hace algo semejante para agregar la segunda señal, ahora seleccionando tipo “TRIANGULAR” y estableciendo los parámetros de amplitud y bits que se indican, y dando clic en el botón agregar, como se muestra en la figura 12.



Figura 12. Configurando la generación de una señal triangular de digitalizada a 8 bits, de frecuencia 4Hz y abarcando el 90% de la escala de conversión. A la izquierda pantalla donde se configuran los parámetros de la señal a generar, a la derecha una vez habiendo dado clic al botón “AGREGAR”.

c) Ahora deberemos seleccionar la frecuencia a la que se muestrearán ambas señales, desde la parte inferior en “FS”, como se muestra en la figura 13. Aquí es importante mencionar que el software está limitado a solo poder generar muestreos de las señales determinísticas a velocidades de 1,5,10,25,50 y 100Hz.

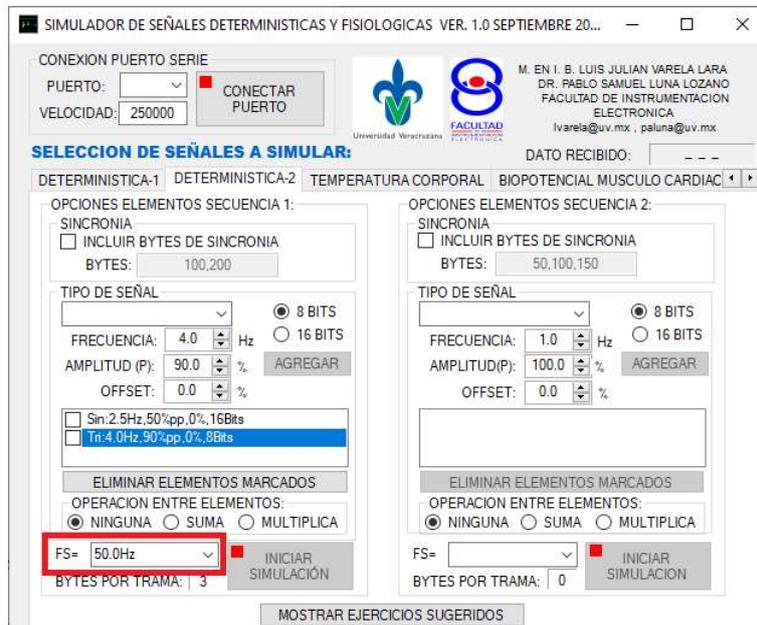


Figura 13. Selección de la frecuencia de muestreo de las señales determinísticas agregadas.

d) Ahora se deberá llevar a cabo la selección del puerto de comunicación, suponiendo que los puertos COM1 y COM2 son los puertos virtuales creados en conexión NULL-MODEM, entonces se deberá seleccionar uno que se encuentre libre, y el otro lo deberá usar la aplicación que se esté desarrollando para recibir la información. Se escribe el valor de la velocidad de comunicación y se dará clic en el botón “CONECTAR PUERTO”, y si todo es correcto, si el puerto no está ocupado por otra aplicación se mostrará la pantalla como se observa en la figura 14.

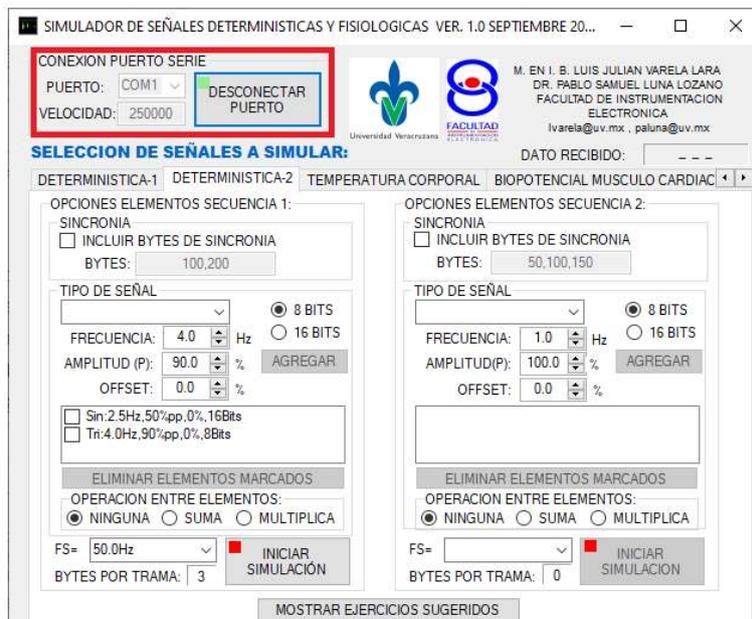


Figura 14. Aplicación cuando se ha configurado el puerto COM de comunicación de manera correcta

En la figura 14, se observa como además de mostrarse un pequeño recuadro que era rojo en el botón “CONECTAR PUERTO” ahora es rojo, simulando un LED que cambia indicando que ahora la conexión está disponible y el texto del mismo botón ahora cambió de función a “DESCONECTAR PUERTO”, puesto que ahora el software se encuentra conectado. Por otro parte adicionalmente en la parte inferior el botón “INICIAR SIMULACION” ya se encontrará habilitado, pero son su correspondiente indicador en rojo, pues las muestras de las señales configuradas ya pueden ser generadas y enviadas por el puerto serial, sin embargo, la simulación será iniciada hasta dar clic en ese botón.

e) Cuando se de clic sobre el botón “INICIAR SIMULACION” ahora ese botón cambiará de función a “DETENER SIMULACION” y su indicador será verde, como se muestra en la figura 15.



Figura 15. Pantalla que se muestra cuando ya se ha iniciado la simulación de las señales determinísticas configuradas.

¿Qué es lo que envía el simulador en este ejemplo?

Dado que se seleccionó una señal de 16 bits y otra de 8 bits, se requieren un total de 3 Bytes para enviar la información de las muestras que generaría un sistema de conversión analógico a digital que estuviera muestreando las dos señales, por lo que el sistema estará enviando una secuencia que encada transmisión estará dada por tres bytes de la forma: H1, L1, L2, H1, L1, L2, H1, L1, L2, H1, L1, L2, H1, L1, L2..., donde H1 y L1 son los bytes “alto” (H) y “bajo” (L) de la sinusoidal de 16 bits, y L2 es el byte de la señal triangular de 8 bits. Esta secuencia estará siendo transmitida una y otra vez con los datos de como cambien los valores de las señales determinísticas con el tiempo, la aplicación que las reciba deberá encargarse entonces de recoger grupos de 3 bytes y rearmar la información, sabiendo que los dos primeros corresponden al valor de la primera señal, y el tercero al valor de la segunda señal.

Note que precisamente en las figuras 12 a la 15, existe un recuadro que indica los “BYTES POR TRAMA”, que para este caso en particular son los 3 bytes que ya se describieron.

Este ejemplo ya sería funcional para que los estudiantes pudieran avocarse a construir una aplicación que reciba los bytes de información y pueda por ejemplo graficar cada una de las señales por separado y/o almacenar la información en un archivo.

Por otro lado, cuando el alumno intente desarrollar la aplicación que recupere la información enviada por el simulador en este punto, podrá tener la oportunidad de detectar un problema que puede surgir ocasionalmente con esta forma de transmitir los datos. Este problema se dará ocasionalmente cuando la aplicación de recepción de información que se esté desarrollando arranque posterior a la puesta en marcha del simulador, y en donde este arranque se haga justo en una parte intermedia de la transmisión de la trama, recibándose por ejemplo los bytes: L1, L2, H1, L1, L2, H1, L1, L2, H1, L1, L2, H1, L1, L2..., note como la secuencia debería iniciar con el byte H1, pero como la aplicación que recibe la información arrancó por razones aleatorias justo después de que este se enviara, la aplicación ya no lo recibió, y por tanto la recepción de información comenzó con la recepción del byte L1, con lo cual la aplicación de recepción de información que hasta el momento solo está programada para considerar que el primer y segundo byte recibido forman el valor de la primera señal, y el tercer byte sea el valor de la segunda señal, y nuevamente de la misma manera los siguientes tres bytes recibidos, vea que entonces se genera un error al armar el valor de la primera señal con los bytes L1 y L2 (en lugar de con H1 y L1), y al tomar el valor de la segunda señal con el byte H1 (en lugar de L2), lo cual es incorrecto, y note que no hay forma de identificar que esto es un error ya que no es posible identificar si la aplicación inicia recibiendo primero el byte H1.

Este tipo de errores pueden evitarse a través de lo que se conoce como bytes de sincronía, que son algunos bytes de valor fijo que se agregan a la trama, y que sirven para marcar donde empezará realmente la secuencia de datos correctamente, y en el peor de los casos se perderá la primera trama, pero las subsecuentes serán recibidas en el orden correcto. Estos bytes de sincronía entonces pueden ser configurados en la aplicación activando la sección “INCLUIR BYTES DE SINCRONÍA”, con ello se tendrá acceso a escribir el valor decimal del byte o bytes fijos que se desean utilizar para marcar en donde inicia la información realmente. En la figura 16 se muestra un ejemplo de usar dos bytes de sincronía, con valores fijos de 100 y 200 (se pueden especificar valores entre 0 y 255 separados por comas para especificar cada byte de sincronía a utilizar).

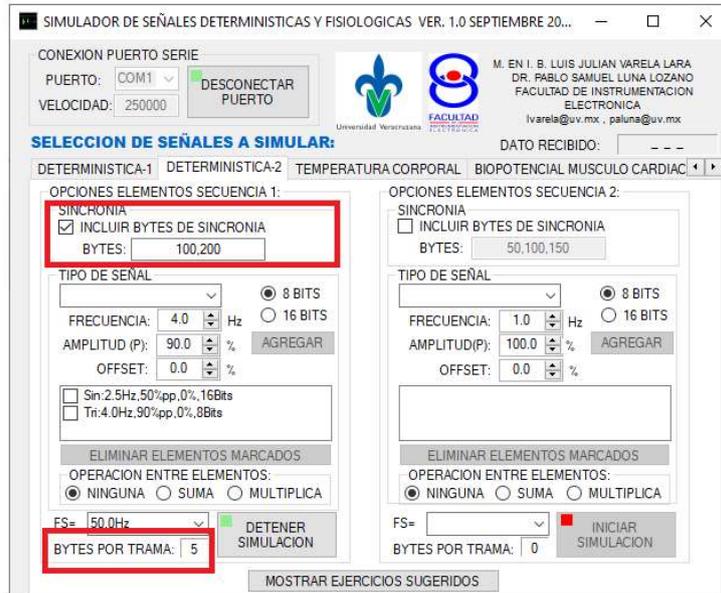


Figura 16. Incluyendo bytes de sincronía en la trama de envío

Así por ejemplo si una aplicación que se esté desarrollando para recibir información ahora recibiera además de los bytes correspondientes a las señales muestreadas, dos bytes adicionales al inicio, uno de valor 100 y otro de 200, y si la aplicación de recepción arrancara de forma posterior y al medio de una trama, recibiendo por ejemplo la secuencia: L1, L2, 100, 200, H1, L1, L2, 100, 200, H1, L1, L2, 100, 200, H1, L1, L2, 100, 200, H1, L1, L2, 100, 200, H1, L1, L2..., note que ahora es posible que la aplicación desarrollada que recibe los datos del simulador de señales, sea programada para esperar a que se reciba la secuencia fija "100, 200" para saber con certeza que los siguientes tres bytes recibidos corresponden al orden correcto a los correspondientes a cada señal muestreada, que aunque en esta secuencia de ejemplo de los bytes que fueron recibidos los primeros dos bytes recibidos son L1 y L2 serán desechados, pero posteriormente las señales podrán ser recuperadas correctamente. Note que si estos dos primeros bytes L1 y L2 por casualidad tomaran valores de 100 y 200 respectivamente la aplicación los tomará como los bytes de sincronía y cometerá un error tomando los siguientes tres bytes como los valores de las señales, pero note también que la probabilidad de que los primeros bytes de la trama entrecortada correspondan al orden esperado de los bytes de sincronía es baja, y en la medida que se aumenten la cantidad de bytes de sincronía esta probabilidad es cada vez menor, y que si la aplicación comete este error inicial la aplicación puede recuperarse posteriormente. En la parte inferior de la figura 16, se observa que el software ahora indica que la trama generada en este ejemplo constará de 5 bytes, debido a los 2 de sincronía (100 y 200), los 2 bytes de la sinusoidal digitalizada a 16 bits (H1, L1) y 1 byte más de la señal triangular digitalizada a 8 bits (L2).

Ejemplo 2: simular la generación de una señal triangular de 5Hz a 16 bits con una amplitud del 2% de la escala máxima, y una señal cuadrada de 1.5Hz de igual amplitud y bits que la anterior. Ambas señales deberán ser muestreadas a 100Hz y agregar los bytes de sincronía 100 y 200 al inicio.

De manera semejante que en el ejemplo 1, este segundo ejemplo puede ser resuelto con una sola sección de “DETERMINISTICA-2”, quedando como se muestra en la figura 17, en la cual además se muestra una aplicación de recepción que grafica los datos recibidos de cada señal.

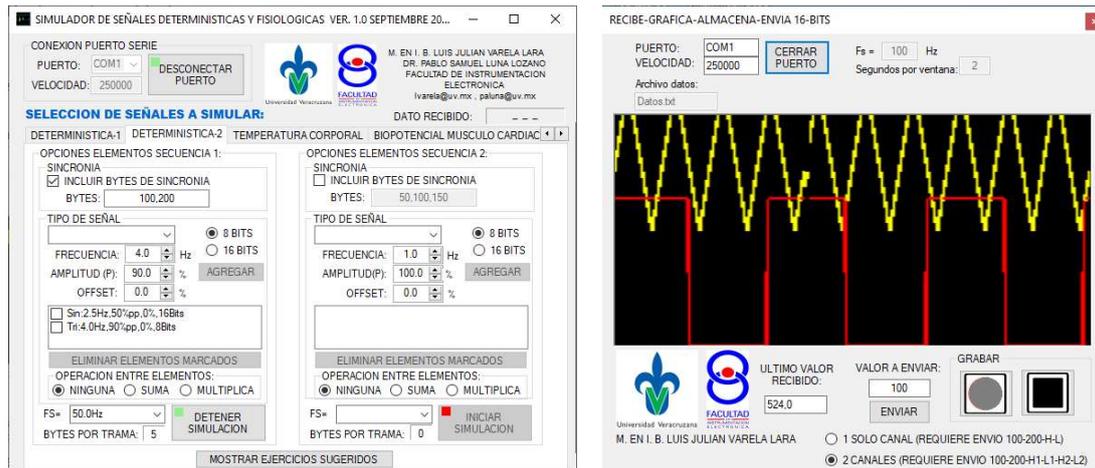


Figura 17. A la izquierda configuración al software “Simulador de Señales Determinísticas y Fisiológicas”, a la derecha ejemplo de una aplicación que recibe, grafica y almacena la información de las señales.

Ejemplo 3: simular la generación de una señal eléctrica del corazón, en la que pueda ser agregando de ruido de 60Hz y movimiento de la línea base.

El software “Simulación de Señales Determinísticas y Fisiológicas”, cuenta con la sección “BIOPOTENCIAL MUSCULO CARDIACO”, como la mostrada previamente en la figura 9. Pero si ahora para esta señal se activan gradualmente las opciones de generación de ruido y movimiento de línea base, en la figura 18 muestra diferentes momentos en que el simulador envía y un ejemplo de software que recibe dicha señal.

En la figura 18 se muestran tres momentos en que se prueba la simulación de la señal eléctrica del corazón, en la parte superior con la opción de simular una señal ECG de 10 Bits, incluir dos bytes de sincronía (100 y 200), por lo que se requieren de 4 bytes en la trama. En la parte central se muestra el momento en que se activa la opción de simular ruido de 60Hz, y en la parte de abajo se muestra el momento en que adicionalmente se activa la generación de movimiento de la línea base, la cual se genera de manera automática cada 15 segundos en el software, con la amplitud que fija el usuario. A la derecha en cada caso se muestra lo que adquiere una aplicación de ejemplo que se encarga de graficar la señal en el formato de la trama establecida.

Con este tipo de ejemplos de señales, el alumno puede practicar algoritmos computacionales para graficar señales, almacenarlas, y de procesamiento como por ejemplo el determinar la frecuencia cardiaca, filtrar la señal para eliminación del ruido añadido, entre otros.

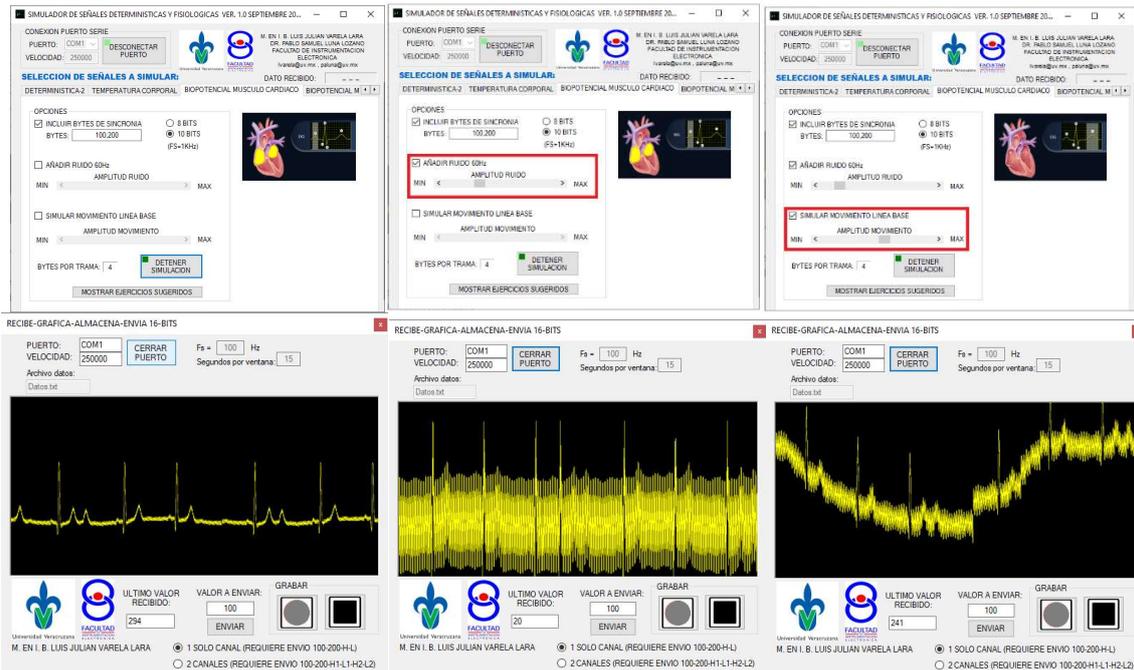


Figura 18. Software “Simulador de Señales Determinísticas y Fisiológicas” configurado para generar una señal de ECG, izquierda simple, en medio con ruido y a la derecha con ruido y movimiento de la línea base

Los otros tipos de señales que puede generar este software “Simulador de Señales Determinísticas y Fisiológicas” son:

-La simulación de temperatura corporal, la cual tiene la apariencia que se muestra en la figura 19, en este tipo de señal pueden establecerse el rango de variación en que se desea simular se mueve el valor de la temperatura corporal, de entre los cuales la simulación elegirá aleatoriamente un valor cada 15 segundos por default (aunque puede ser ajustado por el usuario), pero de lo que adicionalmente, si este valor elegido cae en un rango que represente un problema de hipotermia o hipertermia, la simulación no permitirá que la temperatura vuelva a rondar en valores normales, y se quedará en valores dentro del rango del problema que se está presentando (hipertermia o hipotermia), y solo hasta cuando el simulador reciba por el puerto serial un byte con valor de 10 o 20, para hipotermia e hipertermia respectivamente, el simulador reconocerá que el problema ha sido detectado y atendido, y podrá regresar a simular valores de temperatura corporal normales.



Figura 19. Aplicación al seleccionar la simulación de señal de temperatura corporal.

- La simulación de señales musculares esqueléticas, específicamente de los músculos bíceps y deltoides del brazo, al realizar movimientos de aducción, aducción, flexión y extensión del brazo, con las cuales además de poder apreciar las diferencias entre ellas, el alumno puede generar algoritmos para determinar a partir de estas señales cual es el movimiento que se está realizando, esta opción se muestra en la figura 20.



Figura 20. Opción de simulación de “BIOPOTENCIAL MUSCULO ESQUELETICO”

- Adicionalmente podemos mencionar que existen funciones adicionales sobre las señales determinísticas con las que se puede generar su suma o su producto, y con esto es posible llevar a cabo prácticas para separación de señales por medio de filtros, o bien de demodulación.

En versiones futuras del software se tiene pensado agregar otro tipo de señales, así como funcionalidades añadidas a las señales que ya existen en esta primera versión.

3.5 Datos finales.

La primera versión de este desarrollo fue concluida en septiembre de 2021.

Comentarios y sugerencias a: lvarela@uv.mx o paluna@uv.mx