

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Adquisición de datos

Artículo Científico

M.C. Maestro: Antonio Cayetano Lozano García

Semestre: Ene – Jun 2022

Hora: LMV M3

Grupo: 001

Matrícula	Nombre	Carrera
1902373	Samuel Ernesto Torres Cosío	IMTC
1901410	Julio de Jesús Moreno Sánchez	IMTC
1894698	Christopher Russ Acuña Rodríguez	IMTC
1895481	Alfonso Emiliano Sandoval Juárez	IMTC
1820718	Cesar Alonso Cantú Espinosa	IMTC
1827011	Emilio González Rojas	IMTC

Lugar: Pedro de Alba SN, Niños Héroes, Ciudad Universitaria, San Nicolás de los Garza, N.L.

Fecha: 27/05/2022

Índice

Índice	2
Tabla de figuras	2
Resumen	4
Introducción	4
Antecedentes	4
¿En qué consisten los sistemas de adquisición de datos?	5
Los Primeros Sistemas de Adquisición de Datos.	5
Funcionamiento de sistemas de adquisición de datos.	6
Ventajas de un sistema abierto de adquisición de datos.	7
Identificación de sistemas.	7
Tipos de modelos.	8
¿Qué es Python?	8
Propuesta	10
Resultados	10
Señal Escalón:	10
Señal Rampa:	11
Señal Ruido:	13
Señal Senoidal:	14
Tiempo de muestreo	18
Conclusiones	19
Bibliografía	21

Tabla de figuras

Ilustración 1: Señales medidas de entrada y salida de la planta. Señal escalón de 2 V (azul) y respuesta transitoria (amarillo), 6 s de medición a .1 s de muestreo	10
Ilustración 2: Señales medidas de entrada y salida de la planta. Señal escalón de 2.4 V (azul) y respuesta transitoria (amarillo), 6 s de medición a 0.01 s de muestreo (fuente propia)	11
Ilustración 3: Señales medidas de entrada y salida de la planta. Señal rampa de 5 a 0 V (azul) y respuesta transitoria (amarillo), 5 s de medición a 0.01 s de muestreo (fuente propia)	11
Ilustración 4: Señales medidas de entrada y salida de la planta. Señal rampa de 0 a 3 V (azul) y respuesta transitoria (amarillo), 10 s de medición a 0.1 s de muestreo	12

Ilustración 5: Señales medidas de entrada y salida de la planta. Señal rampa de 4 a 0 V (azul) y respuesta transitoria (amarillo), 10 s de medición a 0.01 s de muestreo	12
Ilustración 6: Señales medidas de entrada y salida de la planta. Señal ruido intervalo de 0 a 1 V (azul) y respuesta transitoria (amarillo), 5 s de medición a 0.1 s de muestreo	13
Ilustración 7: Señales medidas de entrada y salida de la planta. Señal ruido de 0 a 3.83 V (azul) y respuesta transitoria (amarillo), 10 s de medición a 0.01 s de muestreo.....	13
Ilustración 8: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 2V (azul), periodo de 1 s y voltaje de equilibrio de 2.5 V y respuesta transitoria (amarillo), 5 s de medición a 0.01 s de muestreo	14
Ilustración 9: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 5V (azul), periodo de 0.6 s y voltaje de equilibrio de 5 V y respuesta transitoria (amarillo), 2 s de medición a 0.01 s de muestreo	14
Ilustración 10: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 5 V (azul), periodo de 0.6 s y voltaje de equilibrio de 0 V y respuesta transitoria (amarillo), 2 s de medición a 0.01 s de muestreo	15
Ilustración 11: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 5 V (azul), periodo de 0.01 s y voltaje de equilibrio de 0 V y respuesta transitoria (amarillo), 2 s de medición a 0.002 s de muestreo	15
Ilustración 12: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 5 V (azul), periodo de 0.005 s y voltaje de equilibrio de 0 V y respuesta transitoria (amarillo), 2 s de medición a 0.002 s de muestreo	16
Ilustración 13: Señales medidas de entrada y salida de la planta con zoom. Señal senoidal amplitud de 5 V (azul), periodo de 0.01 s y voltaje de equilibrio de 0 V y respuesta transitoria (amarillo), 2 s de medición a 0.002 s de muestreo	16
Ilustración 14: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 5 V (azul), periodo de 0.002 s y voltaje de equilibrio de 0 V y respuesta transitoria (amarillo), 2 s de medición a 0.0018 s de muestreo	17
Ilustración 15: Circuito con el que se obtuvieron las señales anteriores.....	17

Resumen

El objetivo primordial de esta investigación fue el desarrollar una fuente generadora de señales junto con su respectivo sistema de adquisición de datos incorporado, esto con la intención de implementarlo en la identificación experimental de sistemas o plantas a partir de la relación de entradas y salidas. Esto fue posible mediante la programación en Python e IDE de arduino, aunado con el uso de la placa arduino UNO. El resultado obtenido es un sistema completo, capaz de generar y medir diferentes señales con parámetros y características ajustables.

Introducción

El contexto como tal de la línea de investigación que se realizó se centra en que se necesitan el poder identificar sistemas a partir de relación de entradas y salidas, esto ya se había logrado realizar en trabajos previos desarrollado por estudiantes de semestres anteriores, pero se presentaba un problema, el cual está relacionado con las herramientas con las cuales se contaba, específicamente hablando del software de LabView, este no es una herramienta fácil de aprender ni de utilizar, más aún en el contexto estudiantil, entonces surge el incentivo de continuar con los avances de nuestros compañeros pero resolviendo el problema anterior.

La solución a esto se presenta con el desarrollo de una interfaz gráfica capaz de ser utilizada por personas del ámbito de adquisición como del ámbito de control y también logrando cumplir con el objetivo de utilizar un microcontrolador de acceso facil y economico como lo es arduino para poderlo utilizar como un generador de funciones logrando así medir diferentes tipos de señales y como un plus se propuso el objetivo de poder colocar toda la información adquirida por el proceso en archivos ASCII.

Antecedentes

Anteriormente, se realizó un primer esfuerzo por comenzar este proyecto por parte de un alumno en el año 2015. La idea base era la misma, desarrollar un sistema de bajo costo para la adquisición de datos, a diferencia de que, debido a las herramientas comúnmente utilizadas por parte del alumnado en aquellos años eran distintas y más ambiguas comparadas con las de ahora, ellos desarrollaron el proyecto con el uso directo de la placa de desarrollo Arduino como instrumento físico, y utilizando el software de LabVIEW para la programación e interpretación de datos del mismo.

A pesar de ser un tema primordial en el ámbito de la ingeniería de control, no existe ningún otro proyecto basado en el uso de componentes electrónicos para demostrar académicamente a un grupo de alumnos la relación entrada-salida de diferentes tipos de señales con entradas diferentes, sin embargo, existen proyectos interesados en la aplicación de la ingeniería de control para mantener la estabilidad en sistemas físicos, los cuales no serán anexados debido a la diferencia casi completa de objetivos en común.

¿En qué consisten los sistemas de adquisición de datos?

Estos consisten en el proceso de muestreo de señales que miden las condiciones físicas del mundo real y convierten las muestras resultantes en valores numéricos que pueden ser manipulados mediante una computadora. Los sistemas de adquisición de datos, abreviados comúnmente por las siglas “DAS” o “DAQ”, normalmente convierten las formas de onda analógicas en valores digitales para su procesamiento.

Entre los componentes principales de los sistemas de adquisición de datos destacan: 1) sensores, para convertir los parámetros físicos en señales eléctricas; 2) circuito de acondicionamiento de señal, para convertir las señales del sensor en una forma capaz de convertirse en valores digitales; y 3) convertidores analógicos-digitales, encargados de convertir las señales de los sensores acondicionados en valores digitales.

Las aplicaciones de adquisición de datos suelen ser controladas por programas de software desarrollados y utilizando varios lenguajes de programación de propósito general como BASIC, Assembly, C, C ++, Fortran, Java, Lisp, LabVIEW, etcétera. Asimismo, hay paquetes de software de código abierto que proporcionan todas las herramientas necesarias para adquirir datos desde diferentes equipos de hardware. Estas herramientas provienen de las comunidades científicas donde las experimentaciones complejas requieren del respaldo de un software rápido, flexible y adaptable. Dichos paquetes suelen ser de ajuste personalizado y se usan en varios experimentos de física de todo el mundo. (JM Industrial Technology SA de CV, 2022).

Los Primeros Sistemas de Adquisición de Datos.

La compañía estadounidense de computadoras IBM desarrolló computadoras a principios de la década de 1960 que estaban destinadas específicamente a registrar

datos científicos. Esto comenzó con el Sistema de Adquisición de Datos IBM 7700, y su sucesor, el Sistema de Adquisición y Control de Datos IBM 1800. Esto es mucho antes de la llegada de la computadora personal (PC), por lo que estos sistemas eran computadoras a gran escala y muy costosas que requerían una programación y configuración significativas para hacer su trabajo. Pero representaron un gran paso adelante en el registro de datos y condujeron directamente a los sistemas de adquisición de datos basados en PC de hoy en día.

A mediados de la década de 1980, la empresa estadounidense National Instruments Corporation estaba vendiendo componentes elementales como tarjetas de adquisición de datos GPIB y placas convertidoras de analógico a digital (placa DAQ) que se podían instalar en computadoras personales de bajo costo.

Además del avance clave del uso de una PC como plataforma de adquisición de datos, el componente más importante de este sistema fue un programa de software llamado LabVIEW, lanzado en 1986 para la plataforma de computadora personal Macintosh.

Al usar LabVIEW, los ingenieros podrían usar entornos de programación gráfica para desarrollar su propio sistema de adquisición de datos (sistema DAQ), con una amplia variedad de funciones integradas para el procesamiento de datos, análisis y visualización en tiempo real en el monitor de la computadora.

En 1989 se lanzó una versión para PC IBM basada en DOS del software LabVIEW DAQ, llamada LabWindows / CVI, para abordar la mayor parte del mercado de PC. Cuando Microsoft desarrolló su sistema operativo gráfico Windows para la plataforma de PC IBM, la PC de repente tenía capacidades gráficas más comparables a la plataforma Macintosh. En 1992, National Instruments lanzó una versión de LabVIEW para PC con Windows y ha mantenido esa compatibilidad desde entonces. (Smith Grant Maloy, 2020).

Funcionamiento de sistemas de adquisición de datos.

Las señales eléctricas procedentes de los sensores implementados en el sistema se introducen en instrumentos de adquisición de datos que digitalizan las señales y las envían a un disco duro o a otro medio de almacenamiento desde el que puedan manipularse y analizarse haciendo uso de herramientas de software especializadas. Tanto si se dan cuenta como si no, la adquisición de datos desempeña un papel fundamental en campos como la investigación en las ciencias de la vida, la ingeniería

civil y el mantenimiento industrial. Dicho proceso es adaptable a cualquier espacio, utilidad pública o laboratorio de investigación, entre otros ámbitos en los que se requiera el uso de un dispositivo de adquisición de datos que supervise en silencio un parámetro u otro. Una vez que los datos son recopilados pueden utilizarse para mejorar la eficiencia de una empresa, garantizar la fiabilidad o para asegurar que algunas maquinarias (como las relacionadas con la temperatura) funcionen de forma segura.

Ventajas de un sistema abierto de adquisición de datos.

- El componente de mejor rendimiento se puede seleccionar para cada función.
- Los costos se reducen debido a que muchos proveedores están compitiendo en cada área de componente.
- Si un proveedor deja de producir un componente, el usuario final puede cambiar a uno que esté activo.

Los sistemas DAQ cerrados fueron favorecidos por muchos proveedores, debido a que sus usuarios estaban bloqueados, lo que permitió subir los precios, especialmente para repuestos y piezas personalizadas. Sin embargo, los usuarios finales han forzado estos sistemas cerrados y ahora están cosechando los beneficios. (JM Industrial Technology SA de CV, 2022).

Identificación de sistemas.

La identificación de sistemas es un paso previo al diseño del controlador los cuales pueden ser continuos o discretos ya sea mediante técnicas clásicas o por variables de estado.

El modelo le permite al diseñador realizar y validar mediante simulación, el ajuste de los parámetros del controlador. La identificación de sistemas es útil cuando se tienen plantas difíciles de modelar a partir de leyes físicas, como, por ejemplo: el estado del tiempo en un espacio controlado.

Cuando se tienen sistemas no lineales o con parámetros variables en el tiempo (identificación de sistemas on-line).

Cuando se requiere de forma rápida y exacta un modelo matemático. Ejemplo: TOOLBOX de identificación de sistemas de MATLAB

Cuando nos enfrentamos a dinámicas no modeladas, plantas nuevas, por ejemplo: esto es aquellas a las que no se les conoce sus características físicas y químicas.

SISTEMA: es toda realidad en la que interactúan variables de diferentes tipos para producir señales observables.

Las señales observables se denominan: salida del sistema y las señales que pueden ser manipuladas libremente se llaman: entradas del sistema y aquellas que ingresan al sistema de forma aleatoria y que no se pueden controlar se llaman perturbaciones.

Tipos de modelos.

MODELO CAJA NEGRA: no se conoce la dinámica del sistema. Pero a futuro se determina la misma a partir de datos experimentales

MODELO CAJA BLANCA: se conoce la dinámica del sistema y se determina a partir de las leyes físico-químicas.

MODELO CAJA GRIS: Es una combinación de los dos modelos anteriores. Por lo general se obtiene un modelo promedio.

MODELOS MENTALES: es conocido como mentales, intuitivos o verbales. Son aquellos que se describen por la experiencia y carecen de formalismo matemático.

MODELOS NO PARAMÉTRICOS: son aquellos que se caracterizan porque se obtienen a partir de un gráfico o una tabla.

Ejemplo: curva de reacción, diagramas de bode, análisis de Fourier.

MODELOS PARAMÉTRICOS: describen el comportamiento del sistema a partir de ecuaciones diferenciales en tiempo continuo. (SENSORICX, 2022).

¿Qué es Python?

Python es en sí un lenguaje de programación, esto quiere decir que, al ser un lenguaje, requiere del uso de símbolos o caracteres que sigan unas reglas específicas de escritura, similares al lenguaje natural que usan los humanos para comunicarse. Mientras que es dicho de ser “*de programación*” ya que permite organizar esos símbolos a modo de instrucciones que siguen un determinado orden de ejecución.

De este modo “el lenguaje de programación viene a ser una interfaz de comunicación entre la computadora y el usuario” a modo de establecer un programa que ejecute una serie de instrucciones de forma lógico-secuencial utilizando recursos del hardware que compone a la computadora.

Una de las características más notables es el ser considerado como un lenguaje de alto nivel. Esto quiere decir que determinadas operaciones repetitivas son llevadas a

cabo de modo automático, además de permitir un alto grado de abstracción para plasmar y desarrollar código de forma rápida. A diferencia de otros tipos de clasificación de lenguajes de programación a como lo es el lenguaje máquina (cuya principal vía de comunicación es binaria: 1 y 0), el lenguaje ensamblador y lenguajes de bajo nivel que requieren una mayor explicitación a la hora de programar; y que por tanto los vuelven menos intuitivos, más propensos de error y complejos de ser manipulados por usuarios con nulo conocimiento de programación (curva de aprendizaje lenta).

Entre otros atributos del lenguaje Python está la posibilidad de declarar variables cuyo tipo de datos son interpretados automáticamente sin necesariamente declararse formalmente. De este modo se pueden operar con datos del tipo entero, flotante, booleano, complejo, listas, diccionarios, tuplas, arreglos o “arrays”, cadenas de texto, sets, números de distintos sistemas numéricos y efectuar operaciones aritméticas y/o booleanas. Aunado a esto se complementa con el uso de condicionales y bucles, funciones, comentarios y clases (implementa la programación orientada a objetos) (González, Romo, Barrientos, Francisco González, 2012).

También cuenta con módulos y librerías que amplían las capacidades del lenguaje en sí. Algunos módulos están predefinidos o incorporados al momento de instalar el intérprete de python, pero se pueden descargar módulos externos con el fin de realizar tareas de distinta índole; cosa que se detalla más adelante (Pine, 2019). Es de hecho a partir de estos módulos que Python gana prestigio por su implementación en aplicaciones de estilo científico en áreas como ciencia de datos, inteligencia artificial, diseño web entre otros.

La creación y lanzamiento del lenguaje es atribuida a Guido van Rossum en 1991. Python cuenta con “una gestión automática de memoria, sistema del tipo dinámico, al igual que soporta múltiples paradigmas de programación, como el imperativo y orientada a objetos”(Stack Overflow Contributors). Por otro lado este lenguaje es también denominado “*open source*” debido que está disponible para uso público y de forma gratuita; Sin mencionar que también es portable por lo que puede ser ejecutado en distintos sistemas operativos como Windows, Linux y MacOS. Siendo estas las principales razones por la cual se busca la implementación de este lenguaje en el proyecto.

Propuesta

La propuesta que se tiene implica construir una herramienta de adquisición de datos, la cual incluye un programa accesible que cuenta con una interfaz visual, amigable y de uso intuitivo para enviar señales eléctricas concretas y ajustables; al igual que medirlas y guardarlas. Los ajustes incluyen diferentes tipos de señales eléctricas como escalón, rampa, ruido y seno, y sus respectivas características como amplitud, voltaje, tiempo de aplicación y tiempo de muestreo.

Esta herramienta consta de dos componentes, uno vía software, que sería el programa escrito en lenguaje Python-Arduino y el otro vía hardware, que implica el uso del microcontrolador arduino UNO y acondicionadores de señal.

Resultados

A continuación se muestran los resultados para los distintos tipos de señales:

Señal Escalón:

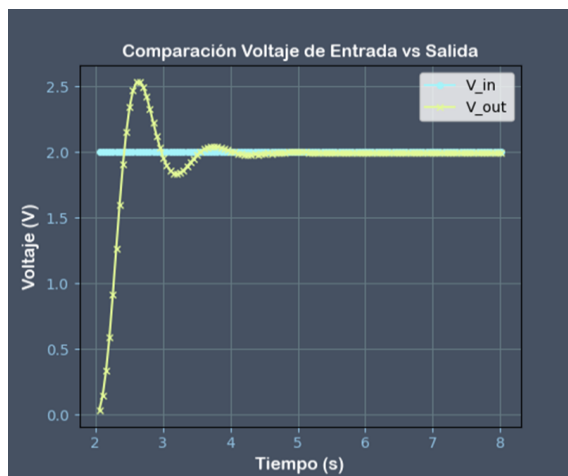


Ilustración 1: Señales medidas de entrada y salida de la planta. Señal escalón de 2 V (azul) y respuesta transitoria (amarillo), 6 s de medición a .1 s de muestreo

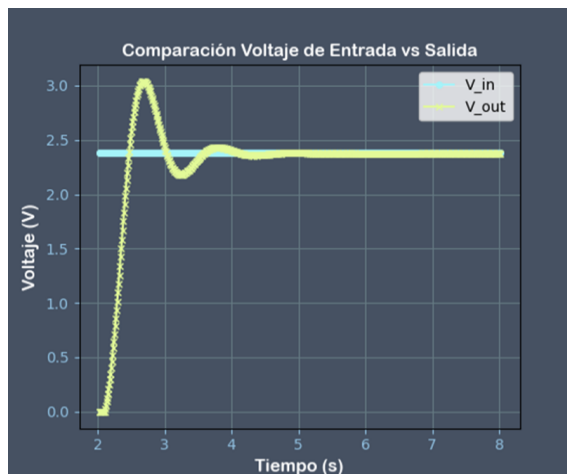


Ilustración 2: Señales medidas de entrada y salida de la planta. Señal escalón de 2.4 V (azul) y respuesta transitoria (amarillo), 6 s de medición a 0.01 s de muestreo (fuente propia)

Señal Rampa:

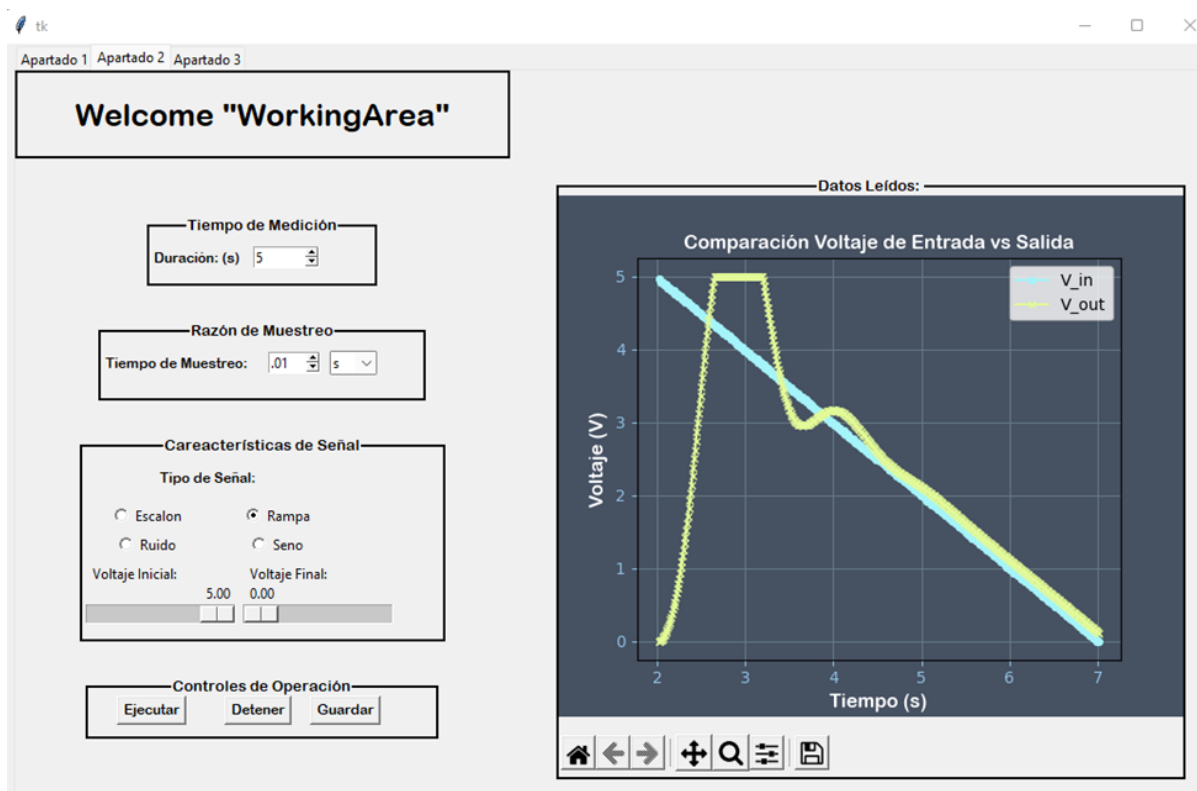


Ilustración 3: Señales medidas de entrada y salida de la planta. Señal rampa de 5 a 0 V (azul) y respuesta transitoria (amarillo), 5 s de medición a 0.01 s de muestreo (fuente propia)



Ilustración 4: Señales medidas de entrada y salida de la planta. Señal rampa de 0 a 3 V (azul) y respuesta transitoria (amarillo), 10 s de medición a 0.1 s de muestreo

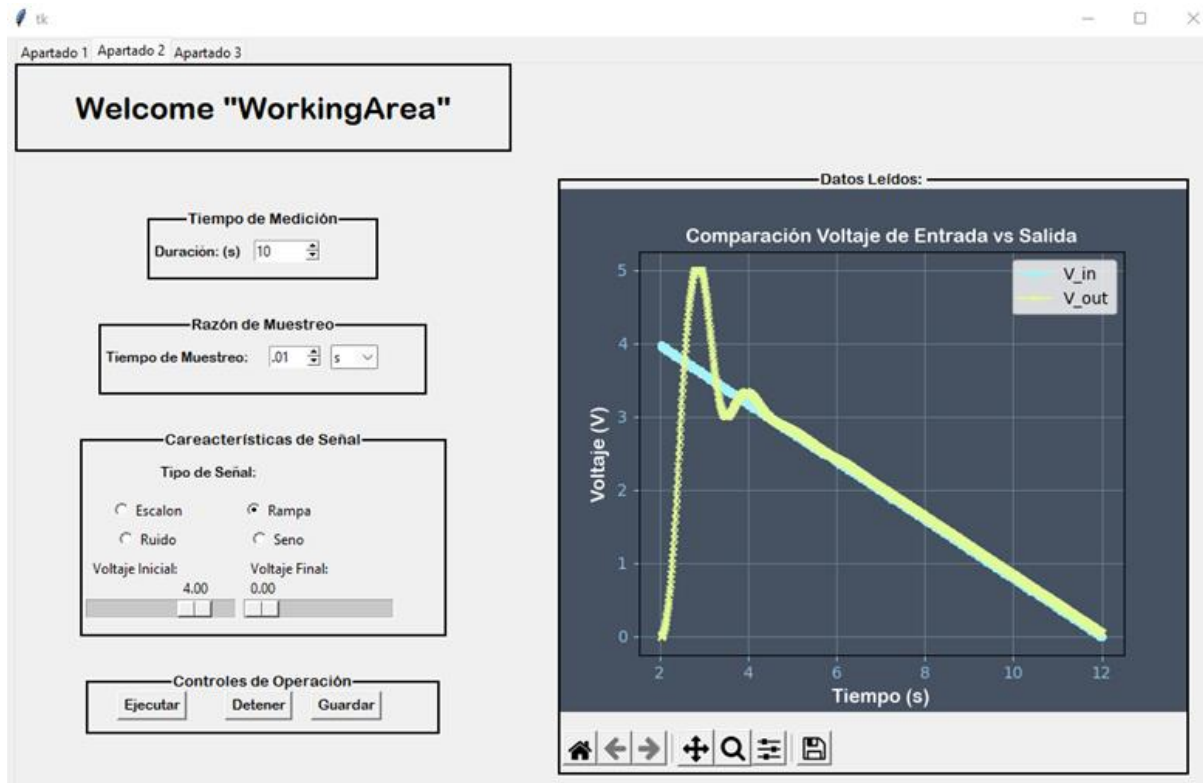


Ilustración 5: Señales medidas de entrada y salida de la planta. Señal rampa de 4 a 0 V (azul) y respuesta transitoria (amarillo), 10 s de medición a 0.01 s de muestreo

Señal Ruido:

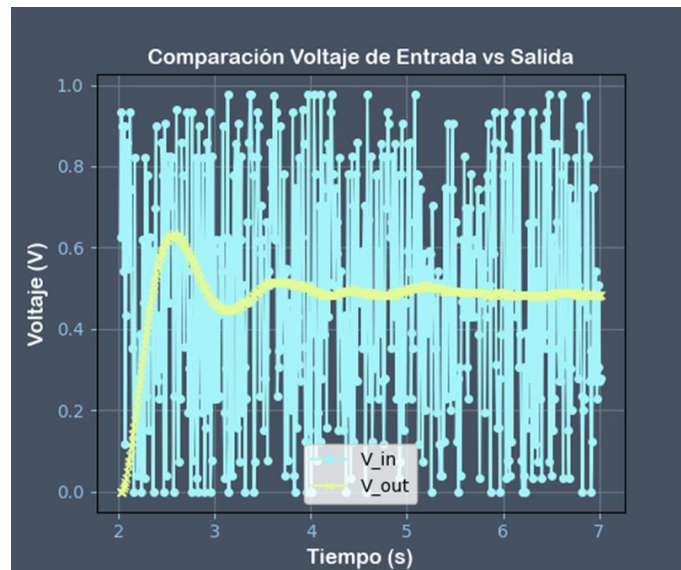


Ilustración 6: Señales medidas de entrada y salida de la planta. Señal ruido intervalo de 0 a 1 V (azul) y respuesta transitoria (amarillo), 5 s de medición a 0.1 s de muestreo

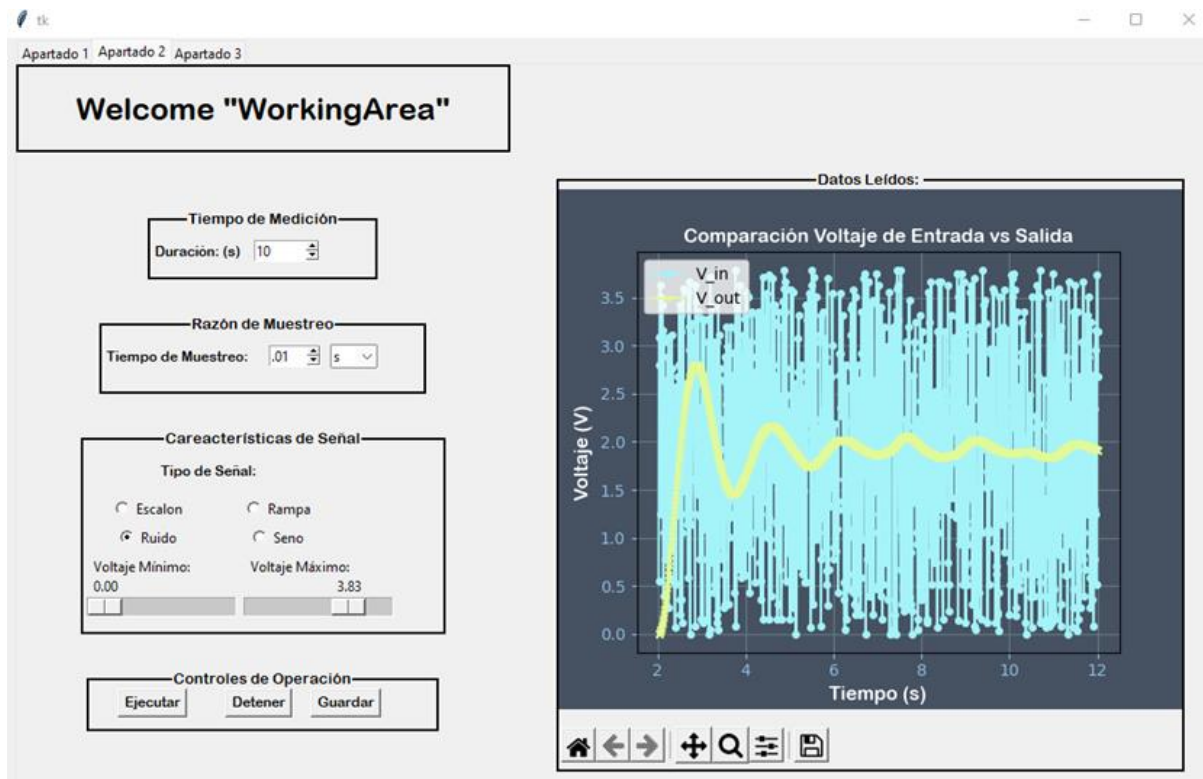


Ilustración 7: Señales medidas de entrada y salida de la planta. Señal ruido de 0 a 3.83 V (azul) y respuesta transitoria (amarillo), 10 s de medición a 0.01 s de muestreo

Señal Senoidal:

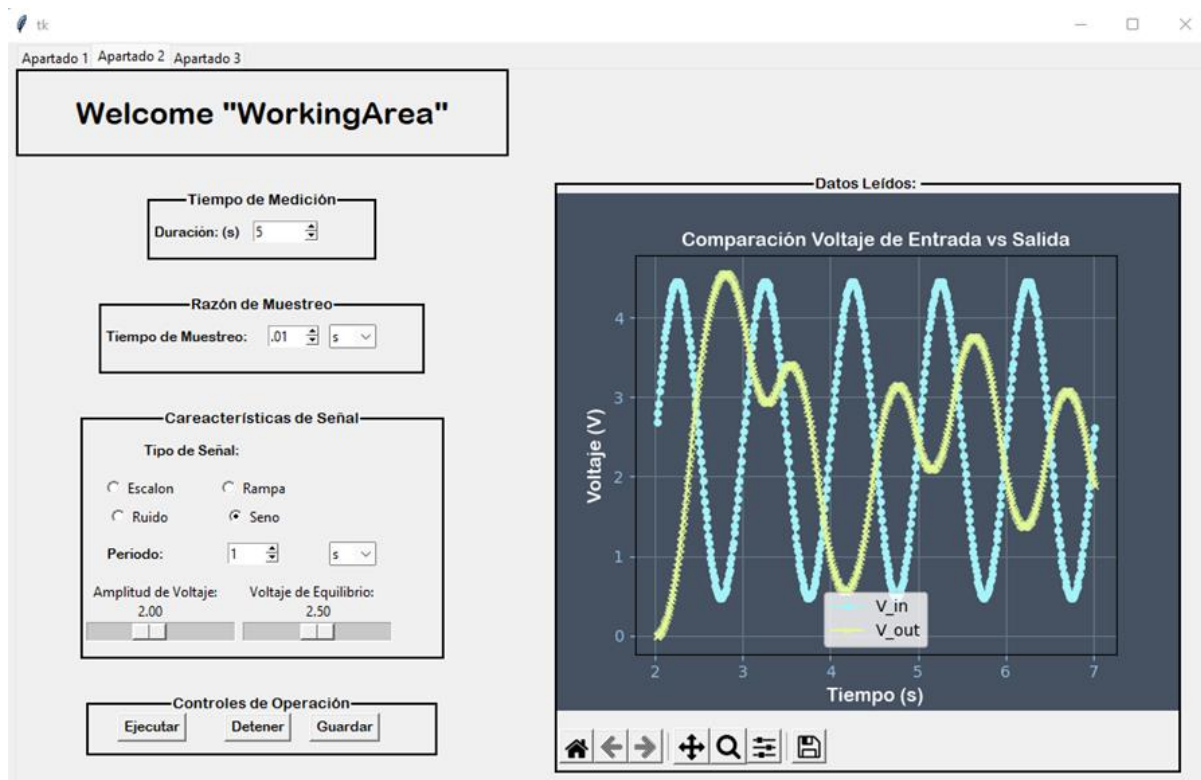


Ilustración 8: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 2V (azul), periodo de 1 s y voltaje de equilibrio de 2.5 V y respuesta transitoria (amarillo), 5 s de medición a 0.01 s de muestreo

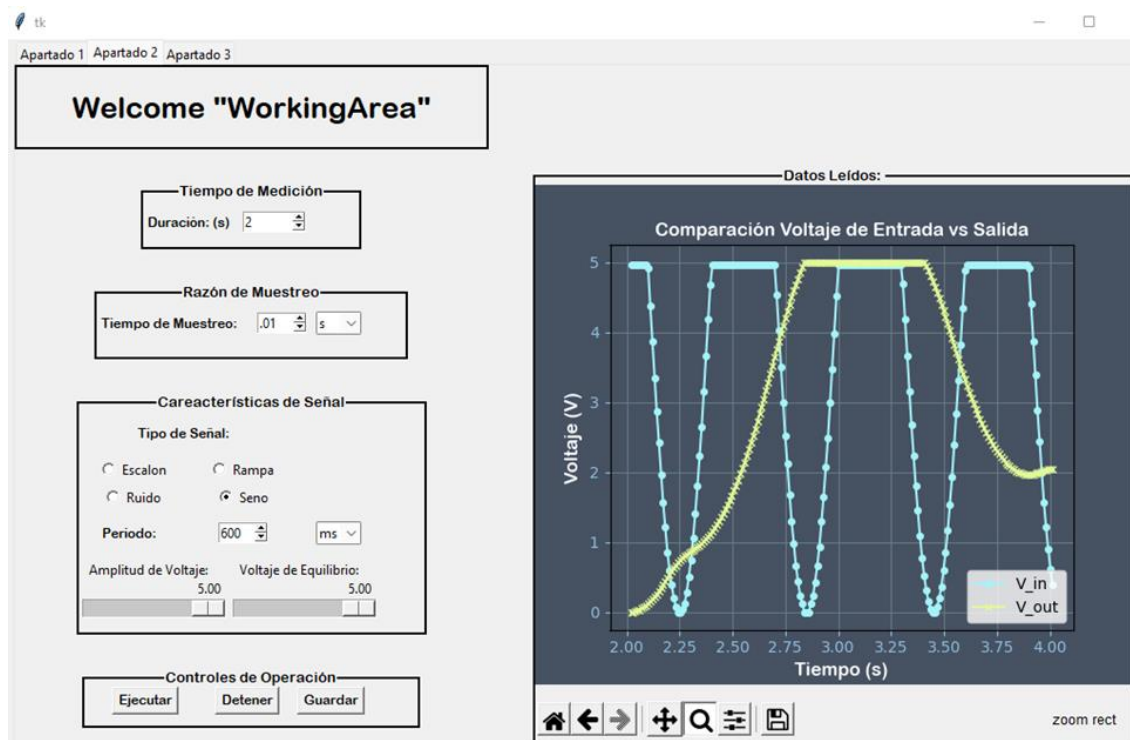


Ilustración 9: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 5V (azul), periodo de 0.6 s y voltaje de equilibrio de 5 V y respuesta transitoria (amarillo), 2 s de medición a 0.01 s de muestreo

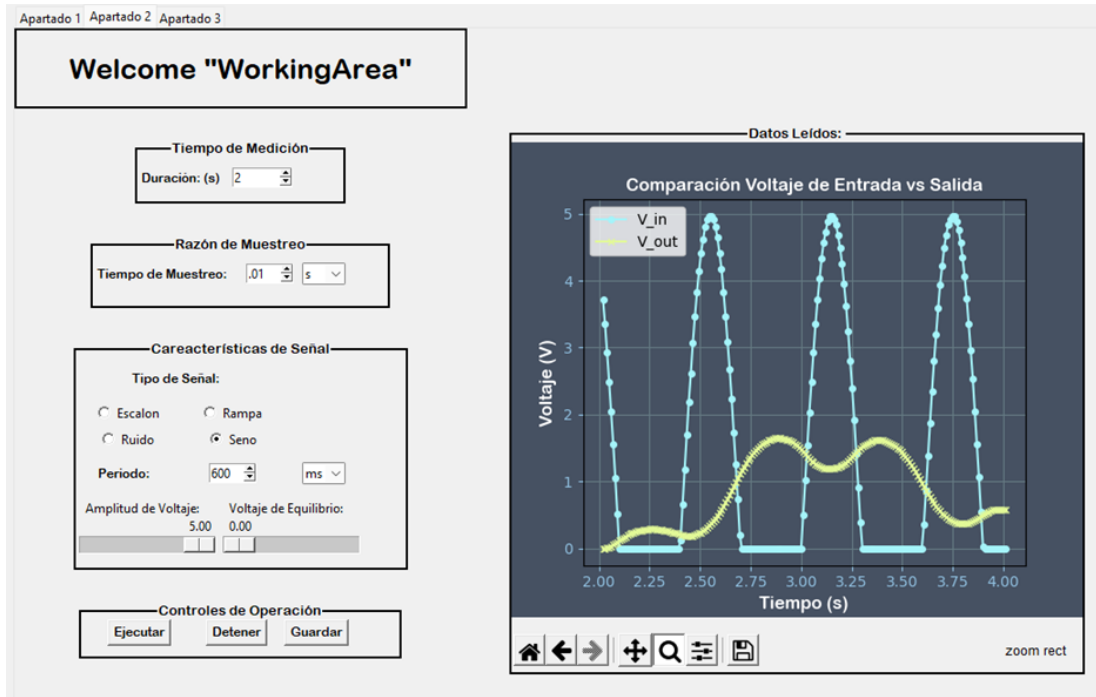


Ilustración 10: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 5 V (azul), periodo de 0.6 s y voltaje de equilibrio de 0 V y respuesta transitoria (amarillo), 2 s de medición a 0.01 s de muestreo

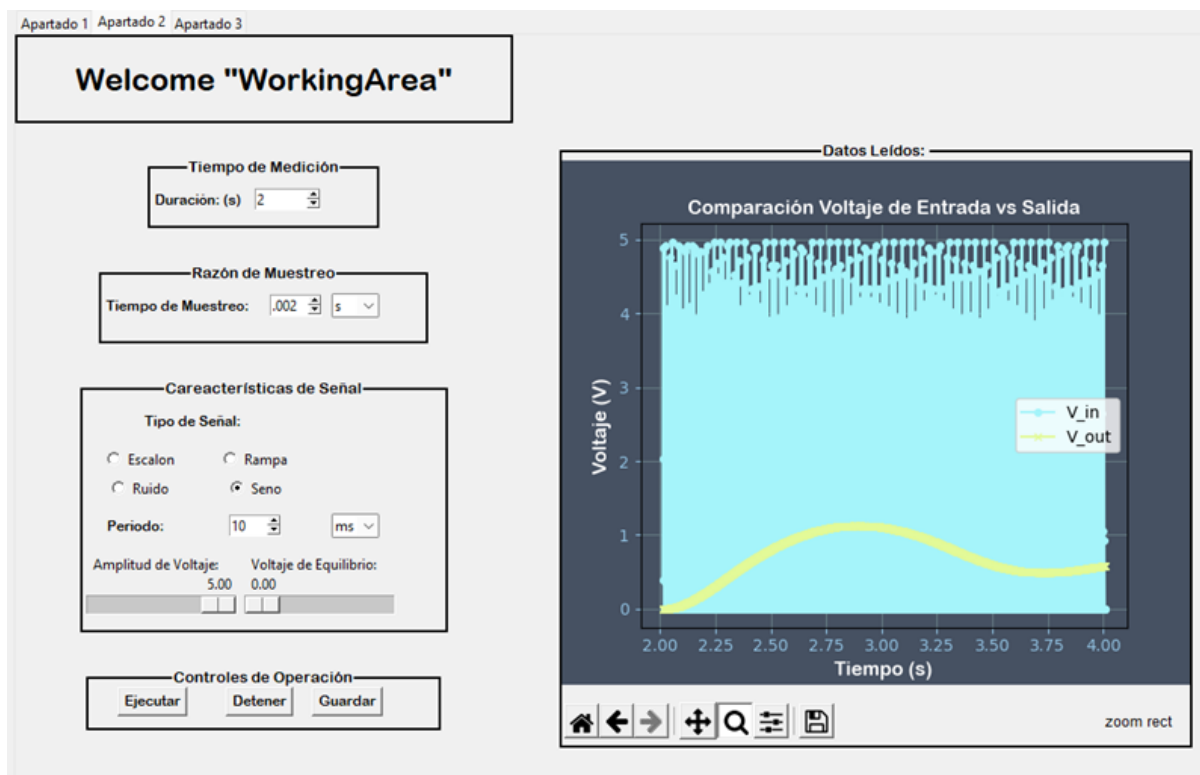


Ilustración 11: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 5 V (azul), periodo de 0.01 s y voltaje de equilibrio de 0 V y respuesta transitoria (amarillo), 2 s de medición a 0.002 s de muestreo

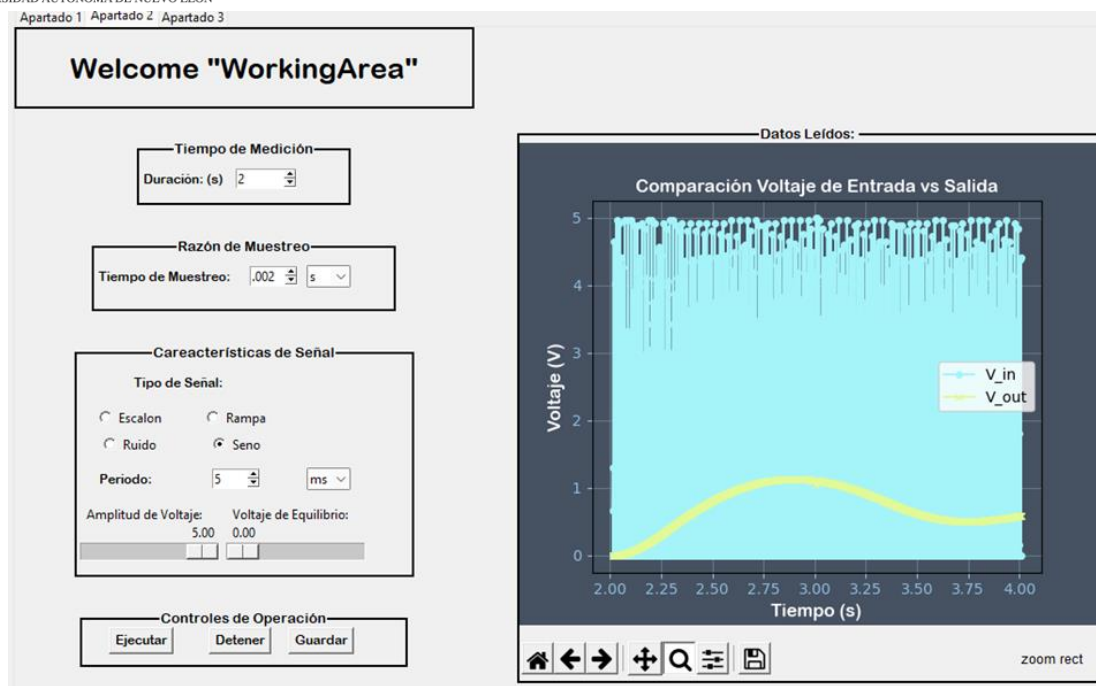


Ilustración 12: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 5 V (azul), periodo de 0.005 s y voltaje de equilibrio de 0 V y respuesta transitoria (amarillo), 2 s de medición a 0.002 s de muestreo

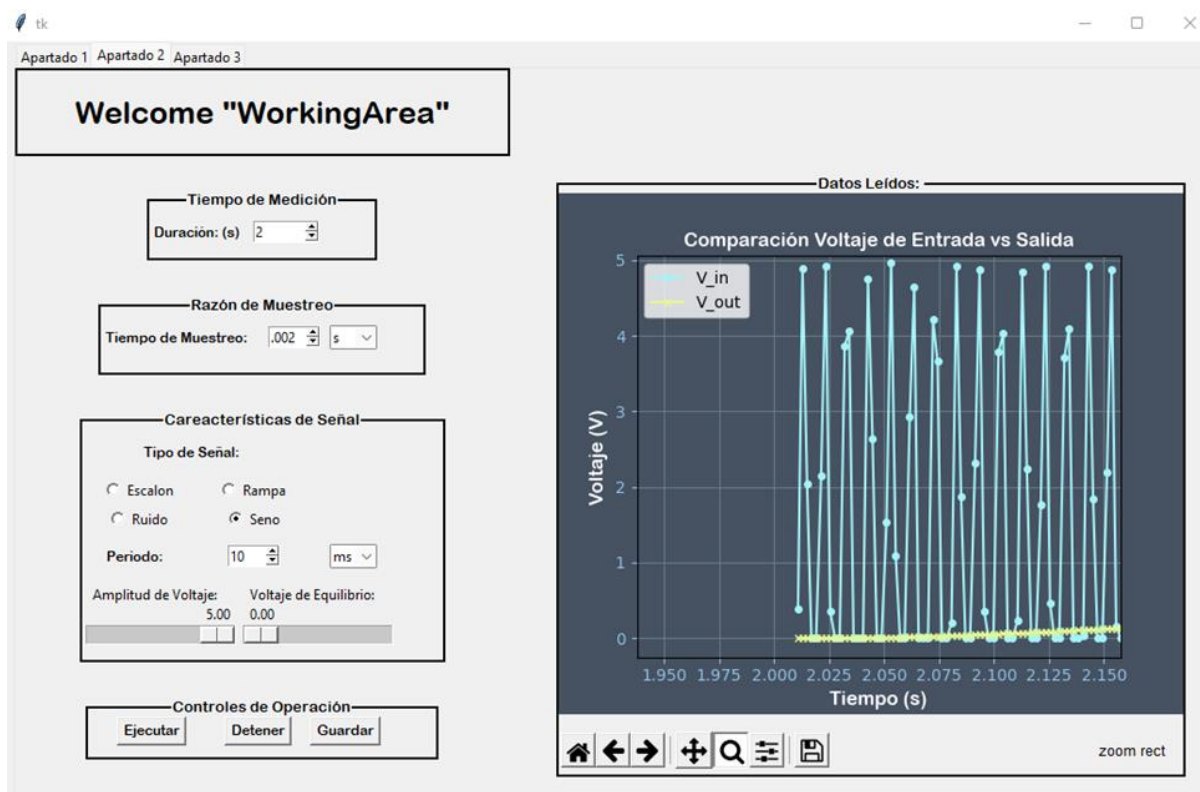


Ilustración 13: Señales medidas de entrada y salida de la planta con zoom. Señal senoidal amplitud de 5 V (azul), periodo de 0.01 s y voltaje de equilibrio de 0 V y respuesta transitoria (amarillo), 2 s de medición a 0.002 s de muestreo

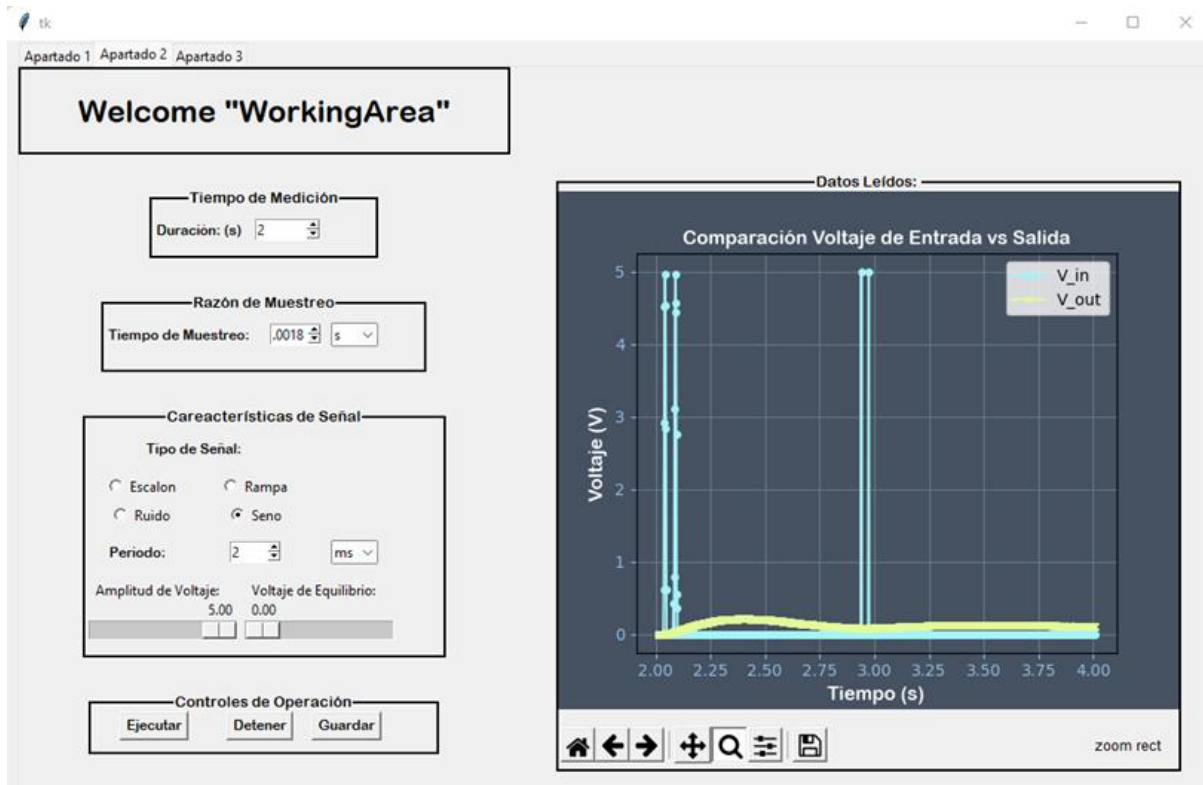


Ilustración 14: Señales medidas de entrada y salida de la planta. Señal senoidal amplitud de 5 V (azul), periodo de 0.002 s y voltaje de equilibrio de 0 V y respuesta transitoria (amarillo), 2 s de medición a 0.0018 s de muestreo

Como puede observarse a partir de las imágenes anteriores, las señales establecidas en la GUI son las esperadas en las gráficas, al menos esto aplica para la señal de entrada (en azul), ya que la señal de salida no se estabiliza con señales diferentes de la escalón, pues están en constante cambio; además que el circuito fue diseñado principalmente a modo que se pueda apreciar la respuesta transitoria tras una entrada escalón y finalmente alcanzar el asentamiento.

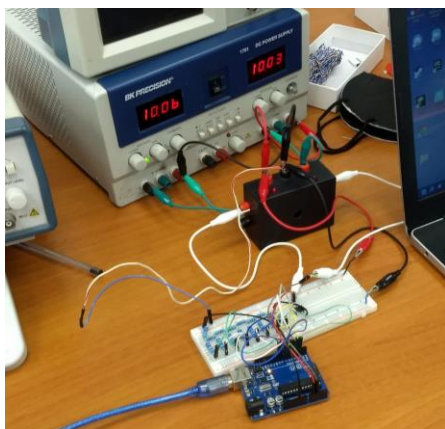


Ilustración 15: Circuito con el que se obtuvieron las señales anteriores

Además en el caso de las señales cambiantes, dicho cambio se produce lo suficientemente rápido como para no afectar el hecho de realizar una medición casi

instantáneamente. Se podría afirmar que estas señales están en “ejecución libre”, pues a pesar de cambiar en base a ecuaciones que dictan su comportamiento, la señal seguirá siendo emitida a pesar de que se mida o no.

Tiempo de muestreo

Otro aspecto técnico muy importante y que es necesario resaltar, es que el tiempo de muestreo mínimo al que funciona el Arduino según datos del fabricante, es de 100 microsegundos. y el alcanzado en la práctica es de 1800 microsegundos. Es decir un 1800 % superior al ideal.

La razón del resultado es tal debido al modo de programación, pues ya que la comunicación es vía puerto serial, la información requiere un determinado tiempo para enviarse, y ese tiempo además está en función del tamaño de la palabra enviada (cantidad de caracteres y los bits de cada carácter) y del baudaje utilizado. Así que si se intenta aproximar este tiempo se tiene que:

$$T.I. = Palabra * [caracter] * \left[\frac{bits}{caracter} \right] * baudaje \left[\frac{bits}{s} \right]$$

En donde T.I. se refiere al tiempo de impresión (envío por puerto serial), la palabra indica la cantidad de caracteres enviados por puerto serial, y de ahí las unidades que la describen (caracteres). Se debe contemplar la cantidad de bits por carácter al igual que la velocidad de envío de bits por segundo, es decir el baudaje. Con esto se puede plantear una buena aproximación.

Atendiendo al tamaño de la “palabra”, se debe considerar que los datos enviados por puerto son 3 por muestra: voltaje de entrada, voltaje de salida y tiempo de medición. En el caso de los voltajes los valores enviados van del 0 al 1023 ya que dependen de los bits del ADC, por lo que aquí ya se cuenta una extensión máxima de 4 caracteres.

Por otro lado el tiempo se mide en microsegundos, y los tiempos más comunes usados en todas las pruebas van de 1 a 30 segundos, por lo que su equivalencia en microsegundos es de 30,000,000 microsegundos; que en caso máximo se tiene un tamaño de palabra de 8 caracteres.

Estos valores se envían por el puerto serial a modo de cadena de texto en donde además se agregan saltos de línea; por lo que a cada palabra se le deben sumar 2 bytes, ya que automáticamente se agrega “\r\n”, en donde r es el retorno y \n el salto de línea (1 byte = 8 bits). Ahora bien también se agrega un último término nulo que

ocupa un byte más. Este término indica el fin de la cadena de texto. Finalmente se considera que cada carácter está constituido por 8 bits; de estos datos se puede determinar aproximadamente el tiempo que toma enviar cada uno por puerto serial e incluso un valor total de la muestra que involucra a los 3 datos. Así que se procede:

Tiempos de impresión:

Voltaje de entrada:

$$T.I._{min} = \frac{(4 + 2 + 1) * [caracter] * \left[\frac{8 \text{ bits}}{caracter} \right]}{115200 \left[\frac{\text{bits}}{s} \right]} \approx 4.86 * 10^{-4} s \approx 486 \mu s$$

Voltaje de salida:

$$T.I._{min} = \frac{(4 + 2 + 1) * [caracter] * \left[\frac{8 \text{ bits}}{caracter} \right]}{115200 \left[\frac{\text{bits}}{s} \right]} \approx 4.86 * 10^{-4} s \approx 486 \mu s$$

Tiempo de medición:

$$T.I._{min} = \frac{(8 + 2 + 1) * [caracter] * \left[\frac{8 \text{ bits}}{1 \text{ caracter}} \right]}{115200 \left[\frac{\text{bits}}{s} \right]} \approx 7.63 * 10^{-4} s \approx 763 \mu s$$

La suma de estos tiempos indica el tiempo mínimo de muestreo el cual es el siguiente:

$$T.M._{min} = (486 \mu s) * 2 + 763 \mu s = 1736 \mu s \cong 1800 \mu s$$

Y como se aprecia, es muy próximo al obtenido experimentalmente.

Por otro lado el microcontrolador funciona a base de tecnología TTL por lo que los voltajes de operación están condicionados al rango de los 0 y 5 voltios. Lo que implica una limitante para la medición de voltajes superiores en cuyo caso se deberían agregar más acondicionamientos a modo que las señales coinciden con la resolución y rango del DAQ.

Conclusiones

Se puede destacar que el proyecto logró cumplir con los alcances que se plantearon al comienzo, logrando tener una interfaz sencilla de usar con las herramientas necesarias para ser utilizada, también queremos dejar en claro que toda la

información y los procesos realizados dentro del desarrollo de esto se dejarán como acceso libre para así en un futuro las siguientes personas que tomen la batuta de mejorar lo desarrollado durante este semestre, puedan contar con las bases que realizamos y logren tener una visión más amplia de lo que se pueda mejorar.

Prestando especial atención al tiempo de muestreo alcanzado, el análisis sugiere que a tiempos mayores de medición probablemente el tiempo de muestreo tienda a ser mayor, ya que el tamaño de la palabra/valor aumenta; y de forma inversa si se trabajan tiempos menores, el tiempo de muestreo será menor. A fin de cuentas se vería afectado el desempeño de las lecturas.

Cabe mencionar que fue la lógica de programación la que produjo estos resultados, es decir, enviar los datos a cómo se leían. Otra alternativa implica usar arreglos (vectores), en los que se van guardando las lecturas, a modo que la impresión o envío de datos ocurra una vez acabe el tiempo de medición.

Esta opción reduce el tiempo de muestreo significativamente ya que las lecturas están en ejecución libre, pero la restricción reside en la memoria del microcontrolador, que al ser limitada, impide tener tiempos de medición prolongados. De esta forma el método usado garantiza tener un tiempo de muestreo algo superior pero el tiempo de medición se puede prolongar considerablemente ya que ahora esos datos pasan a guardarse en el ordenador.

Otra opción sería utilizar la función `millis()` en lugar de `micros()` para reducir el tamaño de la palabra. Pero en consecuencia se presume que hay pérdida de precisión en la lectura de tiempo por lo que si se llegase a requerir la escala en microsegundos, no estaría disponible. También existe la posibilidad de realizar la programación directa del microcontrolador para intentar eficientar la operación del ADC.

En cuanto a la precisión en la señal analógica obtenida, la precisión que esta alcanza se puede incrementar, pues depende del “largo” de la escalera de resistencias del convertidor. La modificación va sujeta a leves cambios al programa, pues se requiere una conversión al sistema decimal al igual que más pines digitales que deben ser programados.

Finalmente se plantea que incluso si se quiere acceder a un microcontrolador con las características como las del Arduino UNO, en el mercado se siguen desarrollando placas cada vez más potentes en cuanto a capacidades pero en contraposición

algunos de esos sistemas embebidos también tienden a miniaturizarse con el fin de tener placas más pequeñas pero lo suficientemente funcionales como ser consideradas de buena calidad y al mismo tiempo baratas. Este sería el caso del arduino nano y abre la posibilidad de abaratar aún más el desarrollo del proyecto.

Bibliografía

Referencia de los pasos:

<https://www.tec.ac.cr/debe-contener-articulo-cientifico>

Páginas de referencia extras:

<http://revistas.untref.edu.ar/index.php/innova/pautas-articulos>

<https://spie.org/news/photonics-focus/janfeb-2020/how-to-write-a-scientific-paper?SSO=1>

¿Qué es la adquisición de datos y por qué es importante?, JMIndustrial Technology,
<https://www.jmi.com.mx/literatura/blog/item/32-que-es-la-adquisicion-de-datos-y-por-que-es-importante.html>

Smith Grant Maloy, Historia de la Adquisición de Datos: desde Registradores Gráficos de Tira hasta DAQ's digitales, 13 Jan 2020, <https://dewesoft.com/es/daq/historia-adquisicion-de-datos#the-first-daq>

INTRODUCCIÓN A LA IDENTIFICACIÓN DE SISTEMAS, Sensoricx,
<https://sensoricx.com/identificacion-de-sistemas/introduccion-a-la-identificacion-de-sistemas/>

González, C., Romo, A., Barrientos, J., & Francisco González. (2012). *Paper Lenguajes de Programación*. Recuperado el 17 de febrero de 2022, de www.academia.edu:

https://www.academia.edu/3887936/90710967_Paper_Lenguajes_de_Programacion_2012

Pine, D. J. (2019). *Introduction to Python for Science and Engineering*. Boca Raton, Florida: CRC Press. Recuperado el 17 de febrero de 2022

Stack Overflow Contributors. (s.f.). *Aprendizaje Python Language*. Recuperado el 17 de febrero de 2022, de [riptutorial.com: https://riptutorial.com/Download/python-language-es.pdf](https://riptutorial.com/Download/python-language-es.pdf)