



universidad  
de león



# Escuela de Ingenierías Industrial, Informática y Aeroespacial

## MÁSTER EN INGENIERÍA INFORMÁTICA

Trabajo de Fin de Máster

SOFTWARE PARA MUESTRO CON TIMESTAMP

SOFTWARE FOR SAMPLING WITH TIMESTAMP

Autor: Álvaro Vila Flecha  
Tutor: Natalia Prieto Fernández

(Julio, 2022)

**UNIVERSIDAD DE LEÓN**  
**Escuela de Ingenierías Industrial, Informática y**  
**Aeroespacial**

**MÁSTER EN INGENIERÍA INFORMÁTICA**  
**Trabajo de Fin de Máster**

**ALUMNO:** Álvaro Vila Flecha

**TUTOR:** Natalia Prieto Fernández

**TÍTULO:** Software para muestreo con timestamp

**TITLE:** Software for sampling with timestamp

**CONVOCATORIA:** Julio, 2022

**RESUMEN:**

El presente proyecto tiene como objetivo el uso e implementación de un sistema de adquisición de datos para el muestreo de señales digitales, las cuales son captadas mediante un módulo de adquisición y son procesadas por una aplicación diseñada y desarrollada por el alumno que se ejecuta en un ordenador, con el propósito final de realizar un marcado temporal o timestamping de la señal muestreada.

Para ello se utiliza el kit de desarrollo que proporciona el fabricante del módulo de adquisición de datos, se trata de una API compatible con distintos lenguajes de programación. Como reloj se utiliza el interno del ordenador donde se ejecuta el programa. El método de procesado de los datos es asíncrono, mediante interrupciones, lo cual aporta ventajas para abordar este desafío en cual se debe tratar de realizar el muestreo y timestamping, de la manera más precisa posible, evitando introducir incertidumbre en el sistema. La aplicación desarrollada tiene en cuenta estas condiciones para que el usuario final pueda muestrear, guardar y representar los datos recogidos y procesados con timestamping.

Este proyecto es de carácter experimental, donde existe por una parte investigación y estudio del equipo proporcionado, y por otra parte existe un diseño y desarrollo de la aplicación para conseguir lograr los objetivos propuestos.

**ABSTRACT:**

The objective of this project is the use and implementation of a data acquisition system for the sampling of digital signals, which are captured by means of an acquisition module and are processed by an application designed and developed by the student that is executed in a computer, with the ultimate purpose of performing a timestamping of the sampled signal.

For this, the development kit provided by the manufacturer of the data acquisition module is used, it is an API compatible with different programming

languages. The internal clock of the computer where the program is executed is used as the clock. The data processing method is asynchronous, through interruptions, which provides advantages to address this challenge in which sampling and timestamping must be performed in the most precise way possible, avoiding introducing uncertainty into the system. The developed application takes these conditions into account so that the end user can sample, save and represent the data collected and processed with timestamping.

This project is of an experimental nature, where there is on the one hand research and study of the equipment provided, and on the other hand there is a design and development of the application to achieve the proposed objectives.

**Palabras clave:** adquisición, datos, timestamping, aplicación, módulo, señal, interrupción

**Firma del alumno:**

**VºBº Tutor/es:**

# Resumen

El presente proyecto tiene como objetivo el uso e implementación de un sistema de adquisición de datos para el muestreo de señales digitales, las cuales son captadas mediante un módulo de adquisición y son procesadas por una aplicación diseñada y desarrollada por el alumno que se ejecuta en un ordenador, con el propósito final de realizar un marcado temporal o timestamping de la señal muestreada.

Para el desarrollo de este proyecto se utiliza el kit de desarrollo que proporciona el fabricante del módulo de adquisición de datos, la empresa Advantech. Se trata de una API compatible con distintos lenguajes de programación que se puede integrar en cualquier programa para el control, la administración y lectura/escritura de datos de los dispositivos. Como reloj para el marcado temporal único se utiliza el interno del ordenador donde se ejecute el programa. El método de procesamiento de los datos es asíncrono, mediante interrupciones desde el módulo de adquisición de datos, lo cual aporta ventajas para abordar este desafío en cual se debe tratar de realizar el muestreo y timestamping, de la manera más precisa y fiable posible, evitando introducir incertidumbre en el sistema. La aplicación desarrollada en C/C++ tanto con interfaz gráfica como ejecutable por terminal, tiene en cuenta estas condiciones para que el usuario final pueda muestrear, guardar y representar los datos recogidos y procesados con timestamping.

Este proyecto es de carácter experimental, y ha estado dividido por etapas que abarcan desde la investigación y estudio del equipo y dispositivos proporcionados por la tutora, pasando por el diseño y desarrollo de la aplicación con el propósito de conseguir lograr los objetivos propuestos y, por último, la etapa de ensayos donde se ha tenido la ocasión de probar y testar la aplicación junto al dispositivos en tres entornos distintos: ideal y simulación, en el laboratorio con generadores de señales, y real, muestreando señales de un sistema ABS de un vehículo.

# Abstract

The objective of this project is the use and implementation of a data acquisition system for the tester of digital signals, which are captured by means of an acquisition module and are processed by an application designed and developed by the student that is executed in a computer, with the final purpose of performing a timestamping of the sampled signal.

For the development of this project, the development kit provided by the manufacturer of the data acquisition module, the company Advantech, is used. It is an API compatible with different programming languages that can be integrated into any program for the control, administration and reading/writing of device data. The internal clock of the computer where the program is executed is used as the clock for the unique timestamp. The data processing method is asynchronous, through interruptions from the data acquisition module, which provides advantages to address this challenge in which it is necessary to try to perform the sample and timestamping, in the most precise and reliable way possible, preventing introduce uncertainty into the system. The application developed in C/C++, both with a graphical interface and executable by terminal, takes these conditions into account so that the end user can sample, save and represent the data collected and processed with timestamping.

This project is of an experimental nature, and has been divided into stages that range from research and study of the equipment and devices provided by the tutor, through the design and development of the application with the purpose of achieving the proposed objectives and, Lastly, the testing stage where we had the opportunity to try and test the application together with the devices in three different environments: ideal and simulation, in the laboratory with signal generators, and real, sampling signals from a vehicle's ABS system. .

# Índice de contenidos

1.	Introducción .....	1
2.	Objetivos.....	3
3.	Estado del Arte .....	4
4.	Desarrollo .....	11
4.1.	Metodología y herramientas de trabajo.....	11
4.1.1.	SCRUM .....	12
4.1.2.	KANBAN .....	14
4.1.3.	Conclusión .....	15
4.1.4.	Aplicaciones y herramientas .....	16
4.1.4.1.	Jira .....	16
4.1.4.2.	GitLab .....	17
4.1.4.3.	ClickUp .....	18
4.2.	Planteamiento del problema y modelado de la solución.....	19
4.2.1.	Antecedentes.....	19
4.2.2.	Nuevo planteamiento.....	21
4.2.3.	El paradigma de las interrupciones y de los eventos síncronos y asíncronos	23
4.2.4.	El uso de la API del dispositivo .....	24
4.2.5.	Diseño de la aplicación .....	26
4.2.6.	Diseño de la interfaz gráfica .....	26
4.2.7.	Arquitectura interna de la aplicación .....	28
4.3.	Tecnologías empleadas: Descripción de hardware y software utilizado .....	31
4.3.1.	Hardware .....	31
4.3.2.	Software .....	40

4.4.	Problemas y soluciones.....	46
4.4.1.	Error en la instalación de una característica en el adaptador de red .....	46
4.4.2.	Actualización driver modulo USB-5830 en Windows 10.....	47
4.4.3.	Frecuencias y voltajes de muestreo .....	47
4.4.4.	La frecuencia de resonancia interna .....	48
4.5.	Ensayos y resultados.....	50
4.5.1.	Ensayo ideal .....	50
4.5.2.	Ensayo de simulación .....	52
4.5.3.	Ensayo real.....	54
5.	Conclusiones y líneas futuras .....	55
	Lista de Referencias Bibliográficas .....	57
	Anexo A. Planificación SCRUM .....	1
	Anexo B. Capturas de pantalla de las ventanas del programa.....	2
	Anexo C. Código del programa: Simple_DIInterrupt.c .....	4
	Anexo D. Código del programa: Clase C++ DIInterrupt .....	7

# Índice de Figuras

Figura 3.1 - Registrador de tiras de papel (Fuente: [1]) .....	5
Figura 3.2 - Sistema de Adquisición de Datos IBM 7700 (Fuente: [3]).....	6
Figura 3.3 - Computadora Macintosh que ejecuta el entorno de programación LabView de National Instrument y tarjetas de adquisición de datos (Fuente: [4]) .....	7
Figura 4.1 - Esquema de la metodología SCRUM (Fuente: [12]) .....	13
Figura 4.2 - Modelo de ejemplo de tablero KANBAN (Fuente: [16]) .....	15
Figura 4.3 - Logo de Jira (Fuente: [20]).....	17
Figura 4.4 - A la izq. logo de git, a la drcha. logo de GitLab (Fuente: [24]) .....	18
Figura 4.5 - Logo de ClickUp (Fuente: [27]) .....	18
Figura 4.6 - Módulo Advantech USB-4751L (Fuente: [28]) .....	19
Figura 4.7 - Módulo Advantech USB-5830 (Fuente: [30]) .....	21
Figura 4.8 - Diagrama conceptual de uso de Signals & Slots de Qt (Fuente: [34]).....	30
Figura 4.9 - Módulo Advantech USB-5830 con desglose de sus conexiones (Fuente: [36])	31
Figura 4.10 - Generador de señales JT-JDS6600 (Fuente: [41]) .....	35
Figura 4.11 - Captura del programa para controlar el generador de señales desde el ordenador (Fuente: propia).....	36
Figura 4.12 - Generador de funciones RIGOL DG1022Z (Fuente: [44]).....	38
Figura 4.13 - Arquitectura del kit de desarrollo proporcionado por Advantech para el uso de sus dispositivos (Fuente: DAQNavi Documentation) .....	41
Figura 4.14 - Programa Advantech Navigator, en la pantalla de testeo de dispositivo (Fuente: propia).....	41
Figura 4.15 - Qt Creator IDE del proyecto actual (Fuente: propia) .....	44
Figura 4.16 - Configuración de equipos para prueba ideal (Fuente: propia) .....	51
Figura 4.17 - Captura de pantalla durante el muestreo a frecuencia fija (Fuente: propia)	52
Figura 4.18 - Configuración de equipos en el laboratorio para ensayo de simulación (Fuente: propia) .....	53
Figura 4.19 - Captura de pantalla durante el muestreo con barridos de frecuencia (Fuente: propia) .....	53



Figura B.0.1 - Icono de la aplicación (Fuente: propia).....	2
Figura B.0.2 - Captura del dialogo de configuración (Fuente: propia) .....	2
Figura B.0.3 - Captura de la ventana principal del programa (Fuente: propia) .....	3

# Índice de Cuadros y Tablas

Tabla 4.1 - Datasheet del modulo Advantech USB-5830 .....	32
Tabla 4.2 - Datasheet del generador de señales JT-JDS6600 .....	36
Tabla 4.3 - Datasheet del generador de funciones RIGOL DG1022Z.....	39

# Glosario

**API:** Application Programming Interface | Interfaz de Programación de Aplicaciones

**BPS:** Bits Per Second | Bits Por Segundo

**CAN:** Controller Area Network | Red de Area de Control

**COM:** COMmunication

**CSV:** Comma-Separated Values | Valores Separados por Coma

**DAS:** Data Acquisition System | Sistema de Adquisición de Datos

**DAQ:** Data AcQquisition | Adquisición de Datos

**DLL:** Dynamic-Link Library | Biblioteca de Enlace Dinámico

**GUI:** Graphical User Interface | Interfaz Gráfica de Usuario

**IDE:** Integrated Development Environment | Entorno de Desarrollo Integrado

**IOT:** Internet of Things | Intenet de las cosas

**POO:** Programación Orientada a Objetos

**PLC:** Programmable Logic Controller | Controlador Lógico Programable

**RAM:** Random Access Memory | Memoria de Acceso Aleatorio

**SDK:** Software Development Kit | Kit de Desarrollo de Software

**USB:** Universal Serial Bus | Bus Serial Universal

# 1. Introducción

La humanidad está viviendo en la era de la información en la que a diario se recolectan y generan enormes cantidades de información. El papel de la informática nunca ha sido mayor para el desarrollo global, sobretodo en el ámbito de los algoritmos y el procesamiento de datos. Pero como bien se sabe, la informática está llegando a todos los campos pensables, como es el caso de la industria 4.0, donde hoy en día se implementan sistemas que unen la innovación tecnológica con la informática.

Este proyecto se centra en los sistemas de adquisición de datos, es decir, en el muestreo y captación de eventos físicos, en este caso en máquinas, para su digitalización con el objetivo principal de realizar un marcado de tiempo o timestamping de la señal muestreada.

De esta fuente de datos se obtiene la información necesaria del proceso con una marca temporal única. Estos datos son guardados y procesados por un software diseñado y desarrollado por el alumno para este propósito en concreto con herramientas y equipo proporcionada por la tutora del proyecto.

El reto de este proyecto es realizar el timestamping de manera precisa, sin que otros aspectos del sistema lo afecten o introduzcan un exceso de incertidumbre. Para ello se utiliza el reloj interno del ordenador donde se ejecuta la aplicación, que por norma general tiene una desviación temporal relativamente baja.

El método en el que la aplicación procesa la información es de manera asíncrona, mediante interrupciones que se generan en el módulo de adquisición de datos. Este método tiene ventajas en la manera de abordar este problema y esto se ha tenido en cuenta en el desarrollo de la solución.

Este proyecto es de carácter experimental, donde existe por una parte investigación y estudio del equipo proporcionado, y por otra parte existe un diseño y desarrollo de la aplicación para conseguir lograr los objetivos propuestos.

El proyecto ha dado sus frutos consiguiendo un sistema de adquisición de datos, en este caso de señales digitales para su marcado mediante timestamping a través de un módulo

que se puede ejecutar en cualquier ordenador y situación de una manera fácil, sencilla e intuitiva para un usuario final.

Los datos experimentales procesados confirman la idea inicial, los cuales también han permitido la optimización del sistema para su uso en un entorno real. A su vez esto ha permitido que este proyecto pueda seguir evolucionando en un futuro.

Finalmente, se ha podido realizar una prueba en un sistema ABS de un vehículo donde se ha podido probar el rendimiento de la aplicación y el módulo de adquisición en un entorno real.

## 2. Objetivos

Los objetivos que abarca este proyecto de adquisición de datos se centran tanto en la parte hardware como software del mismo. La primera haría referencia a la investigación y puesta a punto del dispositivo de adquisición de datos. Mientras la parte software comprendería el desarrollo de la aplicación que gestionaría y controlaría dicho dispositivo.

Más específicamente, estos objetivos se podrían desglosar:

Sobre el dispositivo:

- Investigación de las capacidades del dispositivo de adquisición de datos Advantech USB-5830 proporcionado por la tutora.
- Estudio y prueba de los límites del dispositivo para su integración en un sistema de adquisición de datos de señales digitales con timestamping.

Sobre la aplicación:

- Desarrollo de una aplicación de escritorio con el objetivo principal de procesar las interrupciones enviadas por el dispositivo al momento de muestrear una señal digital y marcarlas con timestamping, para su posterior guardado.
- Asimismo, la aplicación mostrara información procesada de la señal en tiempo real por pantalla a través de distintos widgets.

Tanto el dispositivo como la aplicación deberán ser capaces de muestrear y procesar sin pérdidas de datos a velocidades de muestreo elevadas. A su vez, se desea que la aplicación final sea portable a cualquier dispositivo Windows compatible.

## 3. Estado del Arte

Desde que hay procesos, hay señales a medir.

La mayoría de persona conocen el concepto de grabación de datos. En los inicios de la adquisición de datos y su conservación, las lecturas se hacían de manera manual, se escribían a mano en papel, y posteriormente se procesaban, o bien de manera manual, o se preparaban los datos para introducirlos en los computadores disponibles en esa época. Poco a poco, la tecnología se fue desarrollando permitiendo que las mediciones se registrasen en una cinta magnética y posteriormente de manera continua en un gráfico. Estos dos métodos de registro y conservación de datos científicos o recolectados en tiempo real fueron dominantes hasta la década de 1980.

La definición actual de un sistema de adquisición de datos es un dispositivo (dispositivo DAS o DAQ, comúnmente conocidos) que convierte señales eléctricas analógicas de fenómenos físicos a formato digital, es decir, en datos digitales tratables por ordenadores. Una vez esos datos digitales son recibidos por el ordenador, este puede visualizarlos, almacenarlos y/o analizarlos a través de un software o programa.

Algunos de los términos que se utilizan para hacer referencia a estos sistemas de adquisición de datos son:

- DAS o DAQ: la abreviatura de “Data AcQuisition” o “Data Acquisition System” en ingles
- Instrumento de medición científica: ya que se trata de un dispositivo que mide y registra fenómenos físicos como tensión, presión, vibración, temperatura y más.
- Señales analógicas: señales continuas no discretas de fenómenos físicos del mundo real.
- Dispositivo de entrada y salida: Los dispositivos modernos suelen tener tanto entradas como salidas. El dispositivo utiliza la entrada para leer cierta señal o datos. A través de la salida, es capaz de transmitir señales o datos a otros dispositivos o sistemas.

- Acondicionadores de señal: Hace unos años, se utilizaban acondicionadores de señales que estaban intercalados entre los generadores de señales como son los sensores y el convertidos A/D de adquisición de datos. Hoy en día, los dispositivos actuales realizan esa etapa previa antes de digitalizar.

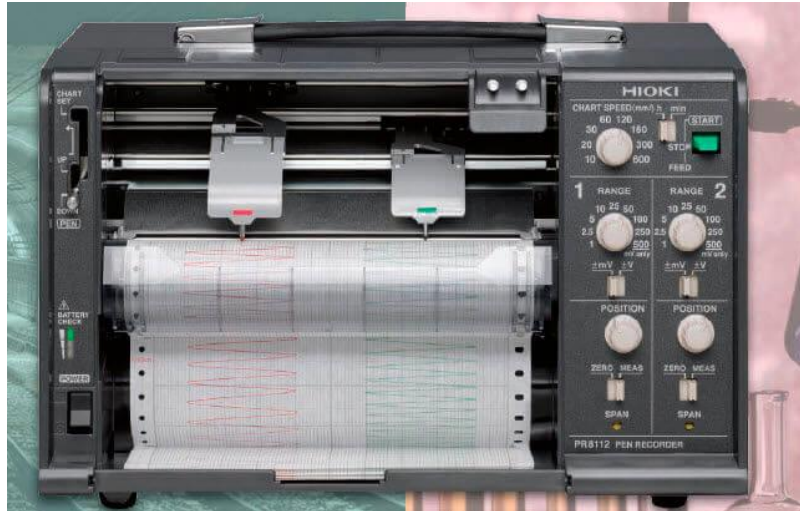


Figura 3.1 - Registrador de tiras de papel (Fuente: [1])

Desde la grabación de datos hasta los sistemas actuales de adquisición ha habido una serie de avances tecnológicos a lo largo de las últimas siete décadas. A continuación, un poco de historia[2].

Antes de la llegada de los sistemas de adquisición de datos como tal, se desarrollaron desde la década de 1950 distintos tipos de registradores que se van a repasar rápidamente a continuación:

- Data Loggers: estos dispositivos fueron diseñados para muestrear a velocidades más lentas de un sistema DAQ, ya que su principal función era registrar datos que cambian lentamente a lo largo de un periodo de tiempo.
- Grabador de cintas de papel: también conocidos como registradores oscilográficos, era una máquina que tenía un bolígrafo móvil que registraba las señales entrantes en un gráfico de papel móvil, a una velocidad de entre 40 y 70 muestras por segundo. El principal inconveniente era el ancho de banda delimitado por la física de mover el bolígrafo.



- Grabador de cinta magnética: ofrecían un ancho de banda superior a las cintas de papel y también mayores tiempo de grabación, pero no tenían la posibilidad de visualizar los datos en tiempo real, excepto en algún caso. Este método se sigue utilizando a día de hoy debido a su robustez.

Los primeros sistemas de adquisición de datos fueron desarrollados por la compañía estadounidense IBM, que diseñaba y construía computadores en la década de 1960 destinadas específicamente a registrar datos científicos. Por ejemplo, su sistema de adquisición de datos IBM 7700 o el sistema de adquisición y control de datos IBM 1800.



*Figura 3.2 - Sistema de Adquisición de Datos IBM 7700 (Fuente: [3])*

Esto fue mucho antes de la llegada del ordenador personal (PC), por lo que eran sistemas grandes, costosos y muy especializados, que fueron utilizados por gobiernos y empresas con capacidad. Aun así, fue un paso importante hacia los dispositivos de hoy en día.

Dado que los sistemas construidos por IBM eran extremadamente caros y voluminosos, entre las décadas 1960 y 1990, fue la era de los grabadores de cinta de papel y cinta magnética. Normalmente se utilizaban en paralelo para proporcionar las mejores cualidades de cada sistema.

Su evolución estas décadas, fue el aumento de ancho de banda para una mayor recolección de datos, en ambas tecnologías.

---

La primera adquisición de datos basada en PC

A mediados de la década de 1980, la empresa estadounidense National Instruments Corporation empezó a vender tarjetas de adquisición de datos y placas convertidoras de analógico a digital que se podían instalar en computadoras personales de bajo costo.



*Figura 3.3 - Computadora Macintosh que ejecuta el entorno de programación LabView de National Instrument y tarjetas de adquisición de datos (Fuente: [4])*

Además de poder usar un ordenador personal como plataforma para adquirir datos, aparece el componente más importante de los sistemas DAQ de hoy en día, un programa software que permitía a los ingenieros desarrollar su propio sistema de adquisición de datos.

Este programa se llama LabVIEW y fue lanzada en 1986 para plataformas Macintosh. Su principal característica es la posibilidad de crear sistemas DAQ con entornos de programación gráfica, en la que se utilizaban multitud de funciones integradas para el procesamiento de datos, análisis y visualización en tiempo real en el monitor de la computadora.

En 1989 se lanzó la versión LabVIEW DAQ para los PC IBM, para abordar gran parte del mercado de ordenadores personales. En 1992, se lanzó la versión para el sistema operativo Windows.

Tras el éxito de LabVIEW en la década de 1990, el cual permitía a los ingenieros crear cualquier sistema de adquisición, lo único que tenían que hacerlo ellos mismo o subcontratar la programación.

Después de este impulso de “Hacerlo uno mismo” tanto en hardware como en software de los sistemas de adquisición, el mercado siguió creciendo y desarrollándose, pero con un problema principal: los elementos eran instrumentos de “llave en mano” con características limitadas, aunque era compatibles con muchos ordenadores gracias al plug-and-play y programables.

Distintas compañías vieron esta brecha entre el hardware y el software y se diseñaron PCs industriales totalmente integrados con tarjetas de adquisición de datos de todo tipo. En el panel frontal contaban con una pantalla, teclado y ratón. Contaban con Windows como sistema operativo y a través de programas ya instalados se podía configurar y visualizar los datos en tiempo real.

Desde entonces la adquisición de datos basada en PC fue en forma de un instrumento, con especificaciones rastreables y además portable.

Desde principios de siglo, las distintas compañías[5] de adquisición de datos han ido innovando y desarrollando dispositivos cada vez más pequeños, más precisos y más baratos, gracias a la evolución tecnológica.

En la última década, los datos se han convertido en información extremadamente valiosa llegando a ser las estrategias comerciales de algunas compañías. La transformación digital en curso está propiciando que los sistemas tradicionales de recopilación de datos se están fusionando con soluciones basadas en modelos digitales. Mismo propósito, técnica moderna.

Ahora los sistemas DAQ modernos deben combinar los requisitos tradiciones de control en tiempo real, distribución de señal remota, facilidad de implementación, escalabilidad, disponibilidad, costos de implementación y mantenimiento reducidos, junto a característica de la denominada industria 4.0, como son la virtualización en la nube, acceso multidispositivo, gestión de grandes volúmenes históricos y su adaptación al Business Intelligence, al IoT y el BigData.

El hardware ha tenido que evolucionar para albergar estas nuevas características, y ahora más allá de las limitaciones, se encuentra el software que se encarga de realizar nuevas tareas con los datos.

Pero para ello hay que comprender lo que está suponiendo en estos momentos la transformación hacia la industria 4.0, lo que se considera como la Cuarta Revolución Industrial[6].

La base de una fábrica, maquina o proceso inteligente es la existencia de sensores y lógica, en otras palabras, hardware y software trabajando juntos. Las tecnologías digitales conducen a una mayor automatización, un mejor mantenimiento y una optimización de los procesos mejorando la eficacia, capacidad y rentabilidad.

Al final, estos sensores que se están instalando a lo largo y ancho del proceso, se encargan de leer e interpretar un evento o señal física y transformarla a datos digitales que pasan por algoritmos para conseguir los objetivos previamente descritos.

Este es el nuevo desafío de la adquisición de datos, la capacidad de integrarse como un servicio en esta revolución tecnológica e industrial, siendo parte del Internet de las cosas (IoT), la computación en la nube, la IA y el machine learning, o bien, en la computación de borde.

Hoy en día existen empresas que venden dispositivos de adquisición de datos para cualquier área donde sea necesario, con características muy avanzadas, fácilmente instalables y configurables, pero por norma general de carácter profesional y genérico lo que implica precios elevados para usos específicos.

Con estos dispositivos se pueden realizar diversas operaciones y tienen integradas multitud de funciones, como el timestamping.

El timestamp o marcado de tiempo se define como un registro impreso o digital que muestra el momento de tiempo en el que un evento ha ocurrido o ha sido realizado. Básicamente es grabar un momento de manera irreplicable.

Este marcado es utilizado en determinados procesos donde se desea obtener información y datos de manera secuencial en el espectro del tiempo, en la que se puede representar la ocurrencia de un evento único

Este método es utilizado por sistemas donde el tiempo es un factor crítico. En aeronáutica, se utiliza para que todos los sistemas de una aeronave estén sincronizados[7] y den su

información en el momento correcto. En sistemas distribuidos, también existe el timestamp para que las comunicaciones entre procesos de distintas maquinas estén sincronizados correctamente[8].

Es un método que permite que el proceso que se esté llevando acabo se realice o se cumpla de manera consecutiva aportando fiabilidad.

## 4. Desarrollo

### 4.1. Metodología y herramientas de trabajo

En este capítulo se detallará las metodologías investigadas y finalmente, se desarrollará la metodología utilizada para la realización de este TFM.

La RAE define metodología como el “Conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.”[9] Por lo que se puede deducir que las metodologías de trabajo son conjuntos de métodos, técnicas, reglas y herramientas para desarrollar un trabajo y conseguir un objetivo determinado.

Desde el nacimiento de las metodologías de trabajo, su enfoque ha sido de carácter estático, realizando una planificación y diseño extensivo antes de su ejecución.

En el mundo del desarrollo de software, el uso de una metodología tan estática va en contra de los intereses del proyecto, por lo que, desde principios de siglo, tras la publicación del manifiesto ágil[10], en los proyectos de software se usan las denominadas metodología ágiles.

Se denominan ágiles ya permiten cambios aun no estando en una fase temprana del proyecto, y es esta flexibilidad lo que permite que el proyecto evolucione, cambie y se adapte a las nuevas necesidades o requerimientos.

Estas metodologías ágiles se enfocan en el desarrollo de software de manera iterativa e incremental, donde los requisitos y soluciones evolucionan a lo largo del tiempo según las circunstancias del proyecto. Su composición está formada por equipos pequeños auto-organizados y multidisciplinarios.

Estos equipos se encargan de gestionarse a sí mismos, tanto en tiempo como en recursos, pero con el objetivo de asegurar un flujo de trabajo constante donde se controlen de manera muy eficiente las posibles eventualidades. Para ello se enfatiza la reuniones cortas, comunicaciones directas y herramientas donde todo el equipo tenga acceso a la información de lo que se está haciendo.

De esta manera, dado que el ciclo de vida del proyecto a realizar está dividido en iteraciones, en los que es obligatorio obtener un entregable funcional, el equipo se prepara antes de empezar el trabajo para esa iteración, se van gestionando durante la iteración para completar los objetivos y finalmente concluyen obteniendo un entregables que incorpora un incremento en el total del proyecto.

Existen un abanico de metodologías ágiles disponibles, pero para este proyecto se explicarán y se compararan dos seleccionadas que se ajustan de manera adecuada para el desarrollo del software requerido.

#### 4.1.1. SCRUM

El nombre se SCRUM[11] proviene de una jugada de rugby que consiste en la unión de todo el equipo para conseguir un objetivo. Aunque normalmente se incluye como una metodología ágil, en realidad es un framework o marco de trabajo para la gestión de proyectos de software de una forma iterativa e incremental.

La diferencia radica en que una metodología está formada por reglas y métodos que hay que cumplir de manera obligatoria, mientras que un marco de trabajo es un conjunto de herramientas que se pueden personalizar según el equipo y/o el proyecto donde se apliquen.

SCRUM define tres roles principales que permite gestionar al equipo de manera eficiente y eficaz. Los roles son los siguientes:

- Product Owner: aunque su identificación puede hacer referencia al cliente, en realidad es la persona encargada de definir los requisitos que se espera obtener del producto a desarrollar. Estos pueden ir cambiando según la necesidad o las circunstancias, y esta persona se encarga de que están bien definidos y actualizados.
- Scrum Master: Esta persona es la encargada de guiar al equipo de desarrollo, ya que es el que más conoce sobre la metodología SCRUM. Su trabajo consiste en dirigir las reuniones, comunicarse con los integrantes del equipo de desarrollo y actuar

entre puente entre el Product Owner y el equipo de desarrollo cuando surgen cambios o complicaciones. El Scrum Master, en ningún momento manda sobre el equipo de desarrollo, sino más bien lo asiste y apoya, permitiendo su auto-gestión.

- Team o equipo: compuesto por personas de distintos ámbitos, son los encargados de desarrollar el proyecto.

Este método se ejecuta en bloques temporales de un determinado tiempo, que se denominan Sprint o iteración. Cada uno de ellos debe dar un resultado completo para ser entregado al cliente. Los Sprints permiten el proceso iterativo e incremental en el proyecto. Normalmente un Sprint dura entre una y cuatro semanas, un periodo de tiempo suficiente para conseguir un entregable funcional.

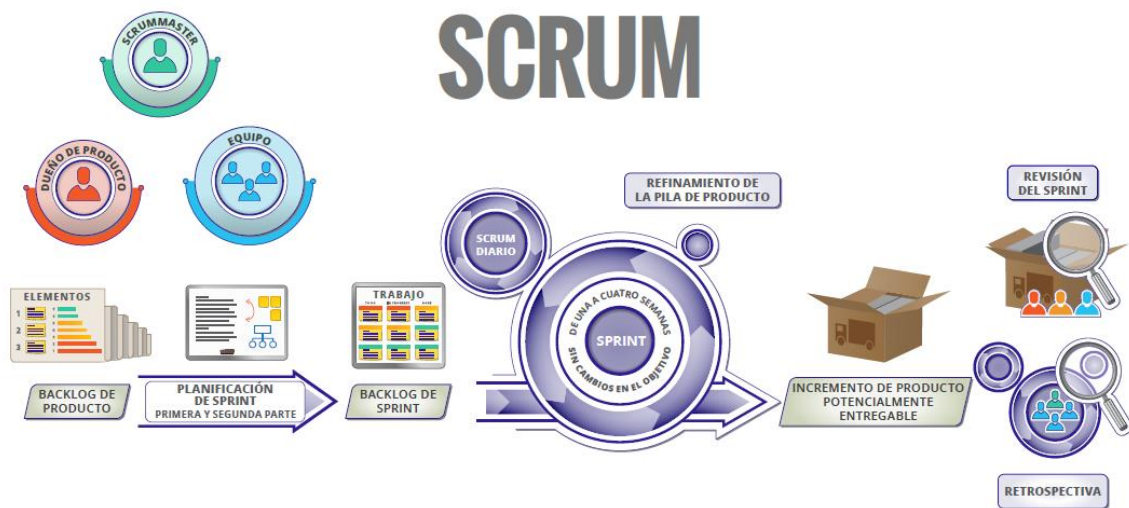


Figura 4.1 - Esquema de la metodología SCRUM (Fuente: [12])

Hay que detallar los elementos y eventos que giran alrededor de los ciclos de vida en SCRUM[13]:

- Backlog de Producto: Es la lista de requisitos y funcionalidades, normalmente denominadas historias de usuario, hecha desde la perspectiva del cliente. Esta lista está ordenada de manera prioritaria, y define el trabajo a realizar en el desarrollo del proyecto. Esta lista evoluciona durante la vida del producto y refleja en una visión única y definitiva todo lo que se podría realizar en algún momento por el equipo de desarrollo. El Product Owner es el encargado de administrar esta lista y prioriza las tareas según los intereses de todos los stakeholders implicados.



- Backlog del Sprint: Es la lista de trabajo que debe realizar el equipo en un determinado Sprint. Los requisitos se dividen en tareas bien definidas y detalladas.
- Planificación del Sprint: Es una reunión para preparar el próximo Sprint a ejecutar.
- Scrum Diario: Es una reunión corta, de quince minutos de máximo, donde el ScrumMaster y el equipo comparten el estado actual del Sprint.
- Refinamiento del Backlog del Producto: Consiste en cambiar el estado del Backlog, ya sea por redefinición de requisitos o bien reestimación o repriorización de tareas.
- Incremento de producto potencialmente entregable: es el resultado de cada Sprint.
- Revisión del Sprint: Es la inspección o demostración del entregable, normalmente al Product Owner. Supone el final de un Sprint.
- Retrospectiva del Sprint: Reunión entre el Scrum Master y el equipo para analizar la manera de trabajar y los problemas que hayan podido surgir, para adaptarse y mejorar el entorno de trabajo.

#### 4.1.2. KANBAN

Kanban[14] es una metodología ágil que nació en Japón, a finales de la década de 1940, con el objetivo de gestionar tareas. La palabra Kanban en japonés significa tarjetas visuales, donde Kan es visual y Ban significa tarjeta.

A principios del siglo XXI, la industria del software adoptó la filosofía Kanban para la producción y entrega de productos y servicios. Esta nueva metodología se basa en cuatro principios base:

- 1) Kanban no requiere configuración y puede ser aplicado sobre flujos reales de trabajo o procesos activos.
- 2) Se persigue el cambio incremental y evolutivo a través de pequeños y continuos cambios incrementales en el proceso donde se aplica.
- 3) Respeta los procesos, las responsabilidades y los cargos actuales
- 4) Animar el liderazgo en todos los niveles de la organización, es decir, fomentar una mentalidad de mejora continua para alcanzar el rendimiento óptimo del equipo.

La herramienta principal de esta metodología es un tablero dividido en columnas verticales[15] donde de manera muy visual se visualiza el estado actual del proceso o proyecto. Este tablero es visible por todos los integrantes de equipo. Cada tarea se representa como una tarjeta que define parte o complemente una funcionalidad a desarrollar

Normalmente, el tablero consta de cuatro columnas básicas:

- Backlog: Columna destinada al conjunto de todas las tareas del proyecto, que están a la espera para ponerse en la cola de trabajo.
- To Do: Columna destinada a las tareas que se pueden ir realizando, si flujo de trabajo lo permite.
- Doing: Columna destinada a las tareas que se están ejecutando actualmente
- Done: Columna destinada a las tareas finalizadas.



Figura 4.2 - Modelo de ejemplo de tablero KANBAN (Fuente: [16])

Al igual que SCRUM, la metodología Kanban busca una visualización del flujo de trabajo, estableciendo límites a las tareas que están en procesos, para asegurar un evolución continua, equilibrada y estable del producto.

### 4.1.3. Conclusión

Como se ha podido apreciar, tanto Scrum como Kanban, son metodologías adecuadas a un proyecto de software e informática como es este Trabajo Fin de Máster.

Ambos siguen una filosofía ágil, flexible, continua e incremental que permite tanto al promotor del proyecto como al autor tener una alta capacidad y rendimiento en el proyecto. Además, son fáciles de implementar y usar, y son muy visuales.

Por otra parte, Scrum demanda que existan roles y reuniones fijas a lo largo del proyecto. A su vez, el tablero se borra completamente en cada iteración. En cambio, Kanban no requiere de división de roles o de reuniones, y el tablero es el mismo durante todo el ciclo de vida.

Estas diferencias[17], permiten ser usadas en distintas aplicaciones, siendo para Kanban mejor los procesos operativos o de mantenimiento. Por el contrario, Scrum se basa en el compromiso del equipo y es mejor para proyecto de desarrollo de corto tiempo.

Por lo que se decide usar para este proyecto el marco de trabajo SCRUM, donde el rol de Product Owner esta desempeñado por el Director del departamento ISA, Carlos Fernández López y la tutora de este proyecto, Natalia Prieto Fernández. Por otra parte, el rol de Scrum Master y del equipo de desarrollo para este proyecto esta desempeñado por el autor de este TFM, Álvaro Vila Flecha.

Para este proyecto se ha determinado que cada Sprint durara dos semanas naturales. En el siguiente apartado se desglosa las aplicaciones y herramientas empleadas durante el desarrollo de este proyecto

#### 4.1.4. Aplicaciones y herramientas

Para el desarrollo de este proyecto y su organización se han utilizado una serie de herramientas que implementan la metodología SCRUM para cada una de las etapas del mismo. A continuación, se presentan:

##### 4.1.4.1. Jira

Para la aplicación de SCRUM en este proyecto se va a hacer uso de la herramienta en línea Jira[18] para el seguimiento del Backlog y de los Sprints.

Jira es una herramienta online para la gestión de tareas de un proyecto, el seguimiento de errores e incidencias. La herramienta fue desarrollada por la empresa australiana Atlassian. Inicialmente Jira se utilizó para el desarrollo de software, sirviendo de apoyo para la gestión de requisitos, seguimiento del estado de desarrollo y más tarde para la gestión de errores. Jira puede ser utilizado para la gestión y mejora de los procesos, gracias a sus funciones para la organización de flujos de trabajo.[19]



*Figura 4.3 - Logo de Jira (Fuente: [20])*

#### 4.1.4.2. GitLab

GitLab[21] es un gestor de repositorios Git basado en web que ofrece repositorios tanto abiertos y públicos como privados, servicios de seguimiento de bugs y wikis, convirtiéndose en una plataforma completa de DevOps para proyectos de software.

Según la propia empresa[22], ayuda a los equipos de trabajo desde la planificación del proyecto y la administración del código hasta el monitorio, la seguridad y el despliegue. Y muchos otros servicios en todas las etapas para reducir los tiempos e incrementar la productividad, todo ello en un sistema seguro.

Un repositorio Git[23] es un sistema de control de versiones utilizado para rastrear cambios en los archivos, en este caso, los archivos de código del proyecto. El objetivo principal es administrar los cambios realizados a lo largo del tiempo y mantenerlos como copia de seguridad o por si es necesario volver a una versión diferente o anterior.

El uso de este servicio se debe a tener una copia del repositorio Git utilizado para el desarrollo de la aplicación para este proyecto, ya que hubo algún problema con las versiones de la API y del sistema operativo usado y se tuvo que trasladar el proyecto a otro equipo de desarrollo.

El proyecto se halla en un repositorio privado de la plataforma GitLab:

<https://gitlab.com/AlvaroVila/tfm-sampling-app>



Figura 4.4 - A la izq. logo de git, a la drcha. logo de GitLab (Fuente: [24])

#### 4.1.4.3. ClickUp

ClickUp[25] es una herramienta de productividad tanto para empresas como freelance, que ofrece una manera fácil y personalizable de gestionar proyectos y realizar colaboraciones a través de su plataforma basada en la nube.

Las funciones incluyen herramientas de comunicación y de colaboración, asignación de tareas, estados y alertas a las propias tareas. Los usuarios se pueden organizar en grupos y equipos. Los proyectos se pueden organizar o visualizar de distintas maneras según la naturaleza del mismo.

Esto permite la organización del proyecto de manera eficiente, donde el flujo de actividad muestra las tareas a medida que se crean y completan en tiempo real.

Entre otras características permite la integración con aplicaciones de terceros[26], como ha sido el caso de Google Calendar, repositorios Git y la plataforma Jira. Por lo que ha sido usado por el alumno como plataforma de organización centralizada.



Figura 4.5 - Logo de ClickUp (Fuente: [27])

## 4.2. Planteamiento del problema y modelado de la solución

El sistema que se presenta en este proyecto sigue la siguiente iteración lógica de un proyecto anterior donde también se pretendía la creación de un sistema de adquisición de datos mediante un dispositivo destinado a ello.

Antes de adentrarse en el sistema creado en este proyecto, se repasará desde que punto de partida se ha empezado, comentando el proyecto del cual se hereda.

### 4.2.1. Antecedentes

El proyecto anterior tenía como objetivo la adquisición de una señal digital mediante un dispositivo que es un módulo de adquisición de datos portables de la empresa Advantech. Su referencia es USB-4751L.



Figura 4.6 - Módulo Advantech USB-4751L (Fuente: [28])

Este dispositivo cuenta con veinticuatro canales que podían ser configurados como entradas o como salidas. Y incorporaba dos contadores que podían ser configurados como entradas para contar pulsos o frecuencia, o como salidas para control de dispositivos digitales. La interfaz de conexión es de tipo USB 2.0.[29]

Dicha señal digital era un tren de pulsos originado en un sistema ABS de un coche que varía en frecuencia según la velocidad del mismo. Por lo que, a mayor velocidad del vehículo, más pulsos, en otras palabras, más frecuencia en la señal. Pudiendo llegar a más de mil muestras por segundos a partir de los 30 km/h.

Por lo que era necesario que el dispositivo pudiera muestrear más allá de 1kHz, para tener lecturas fiables, pero no era posible debido a que el modulo no es capaz de muestrear más allá de 1kHz. Si se superaba este límite, aumentando la frecuencia de la señal, el dispositivo se saturaba y ya no se podía recoger muestras, o bien, daba muestras falsas durante un determinado tiempo. Después se restablecía, pero si se volvía a superar ese límite, el módulo volvía a bloquearse.

Para solventar este problema, se incorporó un reloj externo al dispositivo para que cada pulso tuviera una marca de tiempo con dos objetivos: conocer el tiempo de cada pulso y conocer cuántos pulsos se perdían cuando el modulo se saturaba.

De esta manera se podía descubrir cuantas y cuando faltaban lecturas del muestreo y se realizaba una operación de interpolación posterior para solventar el problema. Pero este método introducía más incertidumbre a las muestras recogidas a mayores del propio sistema por lo que no era muy fiable y preciso.

Además, el método de adquisición era mediante llamada síncrona desde el ordenador al módulo, lo que obligaba al módulo atender la petición de datos del programa lo que a veces junto al problema de saturación durante el muestreo provocaba el bloqueo total del dispositivo.

Este cuello de botella se producía en gran medida debido a la necesidad de solicitar datos en una frecuencia muy elevada durante el muestreo y que la interfaz de conexión con el ordenador es de tipo USB 2.0, con una velocidad máxima teórica de 480 Mbps, lo cual se demostraba que era insuficiente banda para la transferencia de datos entre el ordenador y el dispositivo.

Este sistema implementado funcionaba correctamente, siempre y cuando no se superase la frecuencia máxima de muestreo, lo cual implicaba que la velocidad del vehículo no superase los 30 km/h. A partir de ese momento, el modulo podía fallar, corrompiendo las lecturas.

### 4.2.2. Nuevo planteamiento

Para superar las limitaciones presentes en el módulo de adquisición, por parte del departamento de Ingeniería de Sistemas y de Automática se compró el módulo de datos industrial de la empresa Advantech, cuyo número de referencia es USB-5830.

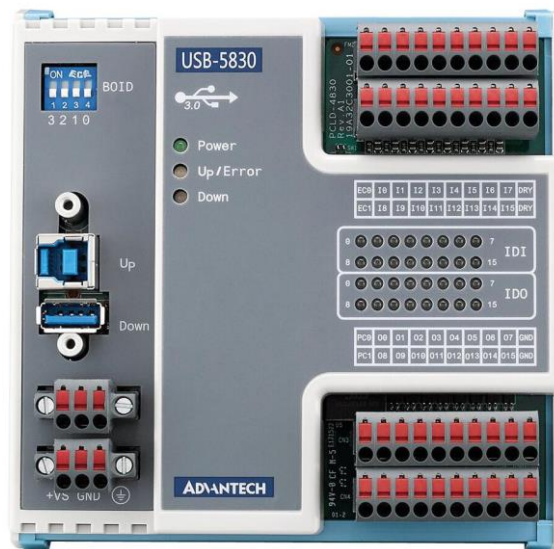


Figura 4.7 - Módulo Advantech USB-5830 (Fuente: [30])

Como se puede apreciar en su número de referencia, es una versión más moderna y avanzada del módulo utilizado anteriormente. Este nuevo módulo tiene unas características tecnológicas y de diseño más cercanas a la industria 4.0 que actualmente se está desarrollando.

Este dispositivo cuenta con dos canales de ocho bits cada de entradas digitales, y otros dos canales de ocho bits cada uno de salidas digitales. La interfaz de comunicaciones es de tipo USB 3.0 y con la posibilidad de conectar más módulos en cadena.

Las operaciones que se pueden realizar con este módulo son las tres siguientes: StaticDI, StaticDO y DIInterrupt.

Las dos primeras permiten la lectura y la escritura de los correspondientes pines de las bornes, mientras que la función DIInterrupt detecta cambio en los canales de entrada. Más específicamente, se pueden configurar para detectar flancos de subida, de bajada o ambos en el primer bit de cada canal de entrada del módulo.



Esta función es el método principal que se utiliza en este proyecto para satisfacer los objetivos. Su porqué se explica a continuación.

El proyecto actual consiste en el timestamping o marcado de tiempo de una señal, específicamente de una señal digital discreta a frecuencia variable. Para realizar el marcado de una señal, sería necesario un contador y un reloj que emparejase cada valor con el momento de su muestra. Dado que este dispositivo no cuenta con reloj interno, se requeriría de una señal de reloj externa, como se realizaba en con el anterior dispositivo de adquisición.

La función DIInterrupt soluciona este problema proporcionando la señal digital como un evento de interrupción, según la configuración del dispositivo. Por lo tanto, cada flanco detectado por el modulo es enviado como un evento al ordenador.

Dicho ordenador sí posee de un reloj, que gobierna las acciones del procesador, en un rango de hasta nanosegundos. Cada evento se maneja con la interrupción que es, y se marca con el tiempo del reloj del ordenador. De esta manera se consigue el timestamping de la señal.

Es importante resaltar el cambio de paradigma que supone el uso de interrupciones con el dispositivo actual al uso de llamadas con el anterior, ya que este es más eficaz a la hora de muestrear una señal de frecuencia variable.

Cada vez que se detecte un flanco de la señal, se generara una interrupción única e independiente que se manejara en el ordenador. Además, hay que destacar que el controlador encargado de gestionar los eventos, garantiza el tratamiento de los eventos uno a uno.

Por lo tanto, se traslada el peso de la acción del dispositivo al ordenador, el cual pasa de realizar llamadas síncronas que provocaban la saturación del antiguo dispositivo a recibir eventos de interrupciones, también síncronos. La ventaja es que el ordenador es una máquina de proceso mucho más potente que el propio dispositivo o que otros sistemas embebidos. En el caso que fuera necesario se podría convertir esta interrupción en un evento asíncrono para el ordenador con la oportuna configuración y programación.

El marcado de tiempo se realiza durante la función manejadora del evento en el ordenador, donde se asigna el tiempo conocido proporcionado por la llamada a la API del sistema operativo que devuelve el tiempo del reloj en ese instante.

De esta manera no se depende de más sistemas externos para el marcado de tiempo y se pueden tratar eventos de manera indefinida, sin problemas de desbordamientos de buffers o falsas lecturas de la señal.

A esto hay que sumarle que la interfaz de comunicaciones es de tipo USB 3.0, con una velocidad máxima teórica de 5 Gbps y capacidad bidireccional por lo que no se producirán cuellos de botella en las transmisiones de datos.

#### 4.2.3. El paradigma de las interrupciones y de los eventos síncronos y asíncronos

Por norma general, con los dispositivos de adquisición de datos, la acción que se realiza es leer o escribir sobre ellos desde el equipo de control, normalmente un ordenador o un PLC, realizando una llamada, que nuevamente suele ser síncrona, lo que puede derivar a lo largo del tiempo en errores, concretamente en los de lecturas.

Si el sistema fuera con un propósito de recolección de datos de una tarea no crítica, estos errores no serían considerados como problemas y podrían ser fácilmente desechados o interpolados. Un sencillo ejemplo sería la toma de datos de una sonda de temperatura de un proceso frigorífico.

En el caso de que fuera un sistema crítico o en tiempo real, como es el caso, usar eventos que manifiesten el cambio de la señal física a muestrear, es mucho más óptimo e ideal para la gestión y el procesamiento del mismo. Cada cambio supone un evento único e independiente en la señal, por lo que, al comunicarlo como una interrupción, esta también es una interrupción única e independiente.

Siguiendo con la cadena de sucesos, a la hora de ser procesada también se asienta en el principio de unicidad al integrarle el timestamping, una marca irrepetible.

Con todos estos razonamientos anteriores, se llega a la conclusión que la mejor manera de realizar timestamping a una señal es mediante su muestreo por eventos o interrupciones, en vez de muestreo por llamadas, el cual puede derivar en errores.

#### 4.2.4. El uso de la API del dispositivo

Una vez determinada la forma de obtener los datos, se muestra a continuación la manera en que ha sido diseñada la aplicación de adquisición de datos en la que se ha tenido que usar el kit de desarrollo de la empresa Advantech para el desarrollo de la aplicación adecuada al problema planteado del muestreo de señales digitales.

En primer lugar, se comentará el contexto de la comunicación entre el modulo y el ordenador donde esté conectado. En la instalación del kit de desarrollo, se instalan distintos componentes, que se enumeran a continuación:

- Drivers
- Manuales
- Guías y ejemplos
- *Device Manager* (Administrador de dispositivos)
- *Runtime Library* (Librería de tiempo de ejecución)
- *Development Kit* (Kit de desarrollo, que contiene las librerías)
- El programa: Navigator Framework

El Device Manager y la Runtime Library son los componentes que hacen posible la comunicación y el intercambio de ordenes e información entre el ordenador y el dispositivo.

El Device Manager se instala como un programa que puede ser invocado por otros programas o bien directamente por el usuario a través de terminal, y cuya función es permitir al usuario encontrar rápidamente el dispositivo apropiado en un buscador de dispositivos, la descripción de sus características, sus configuraciones e incluso probarlo con funcionalidades de prueba.

Por su parte, la Runtime Library proporciona soporte de tiempo de ejecución para los drivers y los programas que invoquen recursos de los dispositivos. Además, provee de un entorno esencial y necesario para el funcionamiento de los dispositivos, como el manejo de errores. Es decir, es la encargada de gestionar

Como se ha descrito, el nivel más bajo es la comunicación entre el modulo y el ordenador, que está gestionado por estos dos componentes obligatorios de instalar. Son la base y para que cualquier programa desarrollado con la API del kit de desarrollo funcione, es necesario que estén instalados en el ordenador.

Existe la posibilidad de instalar el kit en el ordenador de despliegue final, pero existe una segunda opción más sencilla. Una vez desarrollado el programa y probado, se puede crear un paquete que incorpora únicamente: el Device Manager, la Runtime Library y el driver del dispositivo elegido. Con estos tres componentes base ya se puede ejecutar la aplicación desarrollada sin problemas. Esta acción se realiza desde el programa Navigator Framework.

En segundo lugar, se comentará las librerías necesarias que conforma la API dentro de la aplicación para poder conectarse y realizar operaciones con el dispositivo.

En la instalación, se proporcionan una API en forma de librerías. Dependiendo de que lenguaje de programación se use, se debe enlazar una u otra.

Para este caso particular, se va a usar el lenguaje C++ dada que mezcla las características de rapidez y fidelidad de C junto a las características de los lenguajes orientados a objetos que permiten entre otras cosas, incorporar interfaz gráfica.

Para ello, en el proyecto de desarrollo de la aplicación ha sido necesario incluir las siguientes dos librerías:

- bdaqctrl.h: que contiene la API completa con todas las funciones para todos los dispositivos compatibles. Se puede utilizar tanto para C como C++ o C#.
- compatibility.h: una pequeña librería que tiene en cuenta el sistema operativo donde se está ejecutando, ya sea Windows o Linux.

### 4.2.5. Diseño de la aplicación

Ahora que se tiene todo el contexto, se presenta el diseño de la aplicación. Se realiza con las siguientes características:

- Escrito en C++ mediante el framework Qt: Se ha decidido programar en C++ dada su robustez y rapidez, y se va a utilizar las características ofrecidas por el framework Qt para este lenguaje para la interfaz gráfica y el proceso de los datos.
- Uso del paradigma de Programación Orientada a Objetos (POO): La aplicación se puede desarrollar en programas de lenguajes con paradigma imperativo[31], pero las características de la programación orientada a objetos que proporciona C++ permitirán implementar métodos adecuados para el tratamiento de los eventos.
- Procesado asíncrono: Como se ha dicho, se usará Qt que añade meta-características al lenguaje C++ entre las que destaca la capacidad de generar eventos internos en el programa, que serán utilizados para implementar un procesado asíncrono de los datos, separando la recepción de la interrupción de su procesado.
- Interfaz gráfica para visualización en tiempo real: Se incorpora una interfaz básica donde se mostrará información procesada en tiempo real con los datos recibidos por el modulo, con el fin de comprobar el correcto funcionamiento.

### 4.2.6. Diseño de la interfaz gráfica

Para el desarrollo de la interfaz gráfica de usuario (GUI, en inglés), se ha utilizado el diseñador de formularios que incluye el IDE de Qt, con el que se ha creado de manera sencilla a través de los *widgets* que proporciona dos ventanas para el usuario:

- Dialogo de configuración (configdialog)
- Ventana de muestreo (mainwindow)

En el dialogo de configuración, el usuario podrá introducir los parámetros para el muestreo entre los que se encuentran:

**El dispositivo:** el cual es un selector que se completa con los dispositivos disponibles que son escaneados por el programa al arranque. Si no hubiese ningún dispositivo disponible o

en el ordenador no se estuviese ejecutando el administrador de dispositivos, previamente comentado, saltaría un mensaje de advertencia informando del problema y se cerraría el programa.

**La ruta del archivo de perfil (opcional):** en este apartado, se permite al usuario indicar la ruta al archivo de perfil del dispositivo mediante el dialogo de archivos del sistema accesible con el botón “explorar”, en el caso de que se quisiera integrar en la configuración. Dado que es de carácter opcional, si no se selecciona ningún archivo, no saltará ninguna notificación ni se cargará ningún archivo en el proceso de configuración.

**La ruta del archivo de guardado de muestras:** en este apartado, se debe indicar la ruta del archivo de salida de datos, donde se guardarán las muestras recibidas desde el dispositivo. Este apartado, al igual que el anterior, permite al usuario indicar la ruta del archivo a través de un dialogo de archivos del sistema accesible desde el botón “explorar”. Como es un campo obligatorio de completar, no se permitirá proseguir sin seleccionar un destino.

**Elección de canales:** el ultimo parámetro a indicar por el usuario de manera obligatoria es el número de canales que se van a procesar durante el muestreo. Para este caso en particular, el modulo USB-5830 tiene hasta dos canales disponibles, que además coincide con el número de canales que normalmente tiene la familia de este dispositivo. Por lo que el usuario puede configurar el muestreo para un canal, el primero, o bien, ambos, los canales 1 y 2. Como es un campo obligatorio de completar, no se permitirá proseguir sin seleccionar al menos una de las dos opciones disponibles.

Finalmente, hay disponibles dos botones, uno de confirmación y una de cancelación. Si se han completado los campos obligatorios, y se presiona el botón OK, se cerrará este dialogo y se ejecutará la ventana de muestreo. Si se ejecuta la cancelación, el programa finalizara.

La ventana de muestreo es la ventana principal donde se ejecuta y se visualizar la información del muestreo actual.

La ventana está dividida en cuatro cuadrantes. Según el número de canales en activo: Los dos cuadrantes a la izquierda corresponden a información relativa al primer canal, que siempre estará activa, ya es un requisito mínimo. Por el otro lado, los dos cuadrantes a la

derecha contienen la información del canal 2 en caso de que se configure su activación. En caso contrario, se encontraría visible pero deshabilitado.

Según la naturaleza de la información que presentan, los dos cuadrantes superiores contienen las gráficas en tiempo real de los dos canales. Los cuadrantes inferiores contienen los elementos que muestran información relativa al muestreo para cada uno de los canales.

A mayores, la ventana incorpora una barra de menús con dos accesos en la parte superior, una barra de estado de color gris en la parte inferior, y una botonera de tres botones para ejecutar las operaciones de manera rápida e intuitiva.

Los botones son los siguientes:

- **Start:** que arranca el muestreo con la configuración actual.
- **Stop:** detiene el muestreo actual.
- **Configure:** permite cambiar las configuraciones, haciendo reaparecer el dialogo de configuración.

Estas operaciones también se pueden ejecutar desde el menú Archivo de la barra de menús, que añade también la operación de salir.

A su derecha, se encuentra el menú Ayuda, que contiene dos operaciones: ver la configuración actual y un Acerca de. Al ejecutar ambas, muestra un mensaje informativo con su correspondiente información.

Por último, en la barra de estado se van mostrando mensaje de los eventos que van ocurriendo en esta ventana.

#### 4.2.7. Arquitectura interna de la aplicación

Se ha desarrollado una clase encargada de las operaciones de recepción y marcado de tiempo, la cual incluirá las librerías necesarias para conectarse al módulo de adquisición de datos, configurarlo y recibir las interrupciones.

La limitación viene impuesta por la API del módulo, ya que obliga que la función manejadora donde se recibe la interrupción del evento de muestreo sea de carácter estática (static).

Lo cual si se procesase todo en esa función podría llegar otra interrupción que bien, entrase en la cola, o que generase una segunda interrupción dentro de la primera. En cualquier caso, el timestamping quedaría comprometido y dejaría de ser preciso.

Por lo que se ha decidido que esta función manejadora, solo y únicamente, se encargara de realizar el marcado de tiempo de la interrupción y enviara esta información a otra función que se encargara de realizar todas las tareas de procesamiento generando un evento interno. De esta manera, lo que se pretende es que la función manejadora este siempre “desocupada” para que el tratamiento de los eventos y su marcado se realice de manera correcta.

Ese evento transmitirá los datos a la función de procesamiento, cuyas tareas son las siguientes:

- Guardado en un buffer temporal en la memoria
- Guardado del timestamping en el archivo de salida
- Procesado de los datos en el buffer para su representación en la interfaz grafica
- Representación y refresco de los datos procesados en la interfaz gráfica (gráficas y labels)

Para realizar este traspaso de información entre los dos objetos implicados, se utilizará una característica de evento asíncronos que incorpora el framework de Qt sobre C++, en la que en las clases donde se declara el concepto Q\_OBJECT[32], el cual define las características incorporadas por Qt, se puedan realizar tareas de asociación entre funciones del mismo objeto o de distintos objetos. De esta manera se manejan los eventos dentro de la aplicación, como, por ejemplo, los eventos generados en la interfaz gráfica por el usuario, pueden ser fácilmente trasladados a la función controladora.

La característica usada tiene como nombre Signals & Slots[33], según la documentación de Qt son usados para la comunicación entre objetos. Es considerado un mecanismo o



característica importante que aporta el sistema de meta-objetos que proporciona Qt para el desarrollo de aplicaciones.

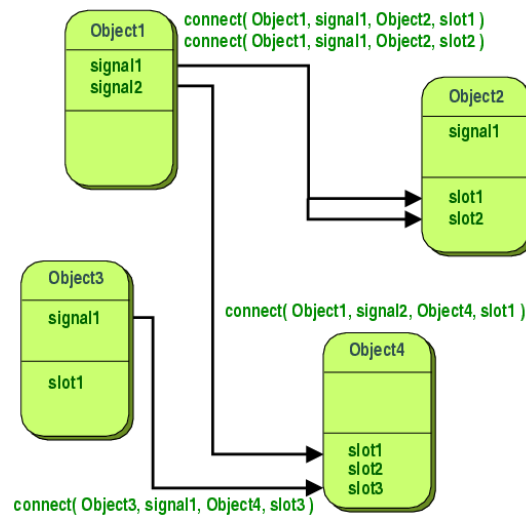


Figura 4.8 - Diagrama conceptual de uso de Signals & Slots de Qt (Fuente: [34])

Para que esto fuera posible, se tuvo que implementar este mecanismo entre la clase controladora que se encarga de las tareas de procesar los datos y la clase encargada del módulo. Además, se implementó un objeto propio y personalizado para transferir esa información, el cual también tenía que ser un objeto como meta-características.

Esa estructura se creaba en el momento de recibir la interrupción y se guardaba en ella los siguientes datos en sus campos:

- Identificador (int)
- Canal (int)
- Segundos (long)
- Microsegundos (long)

### 4.3. Tecnologías empleadas: Descripción de hardware y software utilizado

En este apartado se expone todas las herramientas, tecnologías y elementos usados en el proyecto, clasificados por su naturaleza: Hardware y Software.

#### 4.3.1. Hardware

##### 4.3.1.1. Advantech USB-5830

El núcleo del proyecto es el dispositivo de muestreo Advantech USB-5830[35]. Este módulo industrial de entradas y salidas, ambas digitales, aisladas y con conectividad por USB 3.0 (SuperSpeed), pertenece a la familia USB-5800.

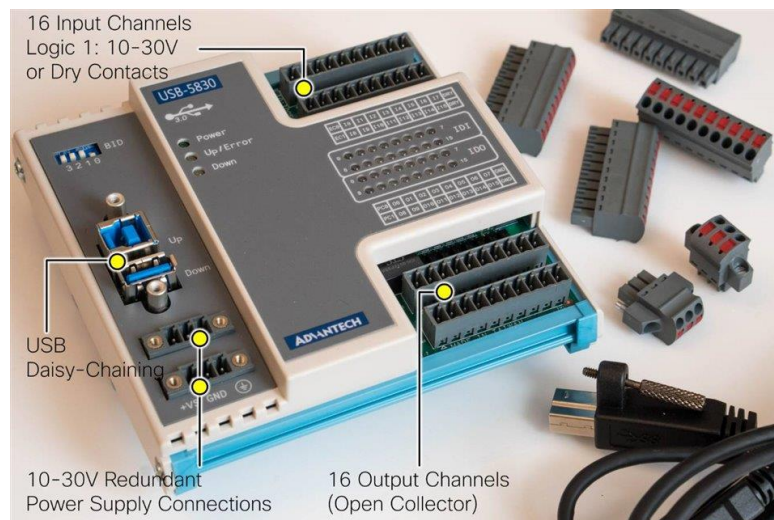


Figura 4.9 - Módulo Advantech USB-5830 con desglose de sus conexiones (Fuente: [36])

Su tamaño compacto y su riel DIN permiten su fácil montaje. La conexión USB incorporada permite conectar varios dispositivos Advantech en una topología de cadena. Los indicadores LED del panel muestran al usuario el estado del dispositivo en todo momento. Los bloques de los terminales cuentan con conectores de tipo europeo. Además, para garantizar una fácil implementación e integración con los sistemas existentes, los módulos USB-5800 admiten numerosas herramientas de desarrollo, incluidas C#, C++, LabVIEW, VB.Net, MFC, BCB, VB6, Java, Delphi y Qt, y son compatibles con Sistemas operativos Windows XP/7/8/10 y Linux.

La familia de módulos USB-5800 ofrece las siguientes características principales[37]:

Conectividad USB 3.0 (SuperSpeed): permitiendo conexiones de hasta 5 Gbps.

Fácil mantenimiento y manipulación: Los indicadores LED, los interruptores giratorios y los bloques de los terminales con enchufes de tipo europeo permiten un fácil acceso y cableado y simplifican el mantenimiento y la manipulación.

Tamaño compacto: Un diseño compacto junto al rail DIN incorporado, permite una fácil instalación.

Concentrador USB incorporado con soporte para conexión en cadena: Los módulos están equipados con un concentrador USB que permite conectar varios dispositivos Advantech en una topología de cadena, permitiendo hasta cinco dispositivos integrados como un solo sistema.

Suministro redundante: Los módulos de esta familia incluyen dos terminales de entrada de alimentación, de rango +10 a +30 VCC, con soporte de redundancia.

Selector de identificador de placa: Los dispositivos incluyen un selector integrado para indicar el identificador (ID) dentro del sistema

Datasheet[38]:

*Tabla 4.1 - Datasheet del modulo Advantech USB-5830*

<b>Entradas Digitales</b>	
<b>Canales</b>	16
<b>Voltaje de entrada</b>	Lógica 0: 3 V máx. Lógica 1: 10 V min. (30 V máx.)
<b>Protección de aislamiento</b>	2500 Vcc
<b>Salidas Digitales</b>	
<b>Canales</b>	16
<b>Voltaje de salida</b>	5 – 40 Vcc
<b>Corriente de salida</b>	350 mA/canal (disipador) @ 25 °C 250 mA/canal (disipador) @ 60 °C
<b>Protección de aislamiento</b>	2500 Vcc

<b>Tiempo de respuesta del Opto-acoplador</b>	100 us
<b>General</b>	
<b>Interfaz</b>	USB 3.0
<b>Velocidad de transferencia</b>	5 Gbps
<b>Conexiones</b>	Bloque de terminales de 10 pines (3.81 mm) x 4 (E/S) Bloque de terminales atornillables (alimentación) x 2 USB 3.0 tipo A (ordenador) USB 3.0 tipo B (hub)
<b>Dimensiones</b>	120 x 120 x 40 mm
<b>Temperatura de trabajo</b>	0 a 60 °C
<b>Alimentación</b>	10 – 30 Vcc

De este módulo adquirido por el departamento ISA y cedido al alumno para este Trabajo Final, junto con un generador de señales, es el principal dispositivo del proyecto.

Como se ha detallado en sus especificaciones y tal como se muestra en la imagen, el dispositivo cuenta con dos canales de entradas, cada uno de 8 bits y dos puertos: Neutro y Común.

Dado que son el eje del proyecto, a continuación, se hablará más en detalle de sus funciones y capacidades.

Como funcionalidad general en la familia de los módulos, pero en específico para este modelo USB-5830, las funciones que soporta son las siguientes:

- **DI\_StaticDI:** Lee una entrada digital repetidamente a través del método "Instant" y muestra el resultado.
- **DI\_DIInterrupt:** Utiliza la función de interrupción y empieza a capturar los cambios que ocurren en los puertos de un canal a través de eventos.
- **DO\_StaticDO:** Escribe el estado en los puertos de salida digital de acuerdo al valor escrito por el usuario.

Para el interés de este Proyecto se han investigado y experimentado con las funcionalidades de los canales de entrada. Según el fabricante tiene una velocidad de muestreo de 800 kilo-muestras/s.

Lo que permite el muestreo tanto de señales digitales como analógicas (ADC) a través de los dos canales y un total de 16 bits de información.

En el caso particular de la función de interrupción por eventos, hay que destacar que el dispositivo actual mantiene los dos canales independientes, por lo que es capaz de capturar dos eventos simultáneamente.

#### *4.3.1.2. Ordenador Windows*

Para el uso y configuración de del módulo Advantech USB-5830, se requiere de un ordenador en el cual corra un Sistema Operativo Windows en una de las siguientes versiones: XP, 7, 8, 8.1 o 10. También opera en Sistemas Operativos Linux[39], siendo los siguientes donde se pueden instalar los drivers: Ubuntu 20.4, Redhat Enterprise Linux Server 8.1, OpenSUSE 15.2, Debian 10.5, CentOS 8.2.

Los drivers están disponibles solo para procesadores de tipo x86, pudiendo operar en sistemas tanto de 32 bits como de 64 bits. Siendo el unico requisito físico o de hardware que debe integrar el ordenador es tener conectividad USB 3.0 de alta velocidad (SuperSpeed) para aprovechar el máximo de banda que permite el dispositivo.

Teóricamente, la versión USB 3.0 permite hasta un máximo de 5 Gbps (600 MB/s), si el dispositivo se conectara a una versión anterior, no se aprovecharía su capacidad.

El ordenador de investigación es un HP 500-353ns, el cual tiene instalado el Sistema Operativo Windows 8.1 de 64 bits. Posee un procesador AMD A10-6700 APU with Radeon(tm) HD Graphics de 3.7 GHz, 16 GB de memoria RAM y dos puertos USB 3.0 incorporados en la placa base.

Para el uso del dispositivo y de las aplicaciones desarrolladas para dicho modulo, solo se requiere un ordenador que cumpla con los requisitos anteriormente descritos y que las aplicaciones estén compiladas para la arquitectura final de uso.

#### 4.3.1.3. Generador de señales

Para la investigación y experimentación con el módulo Advantech USB-5830, se ha utilizado un generador de señales. En este caso, se le ha prestado al alumno, durante la realización de este proyecto el generador de señales JT-JDS6600[40] propiedad del departamento ISA de la Universidad de León.



Figura 4.10 - Generador de señales JT-JDS6600 (Fuente: [41])

El generador de señales JT-JDS6600 es de doble canal y es capaz de generar señales hasta los 60 MHz. A mayores, incorpora una entrada que se puede utilizar como contador.

Es capaz de generar una gran variedad de ondas de manera DDS[42], entre las que se encuentran: sinodal, cuadrada, triangular, pulso y ruido aleatorio.

Entre sus características más relevantes se encuentra la capacidad de poder sincronizar ambas salidas, o bien, dejarlas independientes. Si se emparejan, se pueden sincronizar por determinadas propiedades: frecuencia, forma de la onda, amplitud, 'offset' y 'duty'.

Toda la información y configuración se puede visualizar y cambiar desde su pantalla LCD a color y los distintos botones que incorpora. Además, se puede conectar a un ordenador a través de USB, por el cual se puede controlar y visualizar su estado desde la aplicación que ofrece el fabricante.

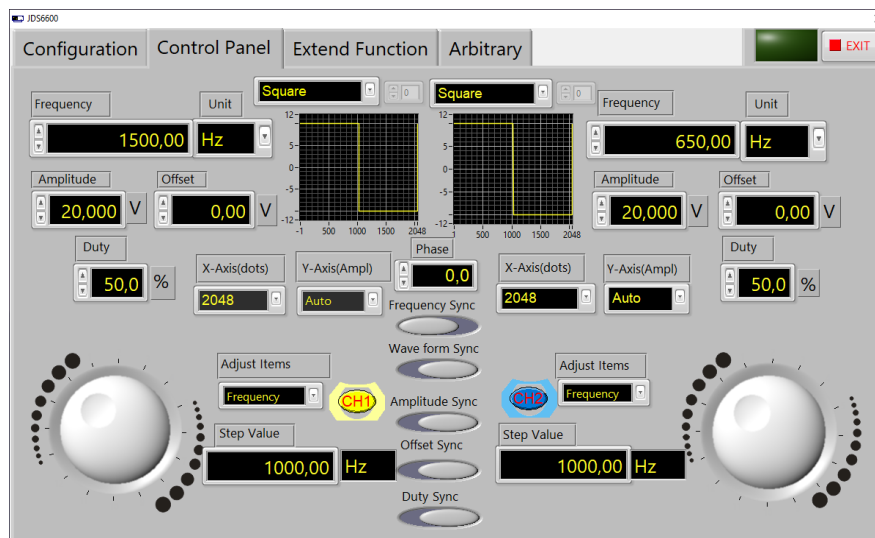


Figura 4.11 - Captura del programa para controlar el generador de señales desde el ordenador (Fuente: propia)

A continuación, se muestra una tabla resumen de las características[43] del dispositivo:

Tabla 4.2 - Datasheet del generador de señales JT-JDS6600

<b>Características de frecuencia</b>	
<b>Rango de frecuencia en señal cuadrada</b>	0 – 15 MHz
<b>Tiempo de subida en señal cuadrada</b>	$\leq 15$ ns
<b>Resolución mínima de la frecuencia</b>	0.01 $\mu$ Hz
<b>Características de la forma de onda</b>	
<b>Formas disponibles</b>	Sine, Square, Triangle, Pulse ,Partia Sine, CMOS, DC level, Half-wave, Full-Wave, Pos-Ladder, Neg-Ladder, Noise, Exponential Rise, Exponential Fall, Tone, Sinc Pulse, Lorentz Pulse, and 60 kinds user defined waveform.
<b>Tamaño de la forma de onda</b>	2048 bits
<b>Resolución vertical de la forma de onda</b>	14 bits
<b>Sobresaliente (overshoot) (onda cuadrada)</b>	$\leq 5\%$
<b>Características de las salidas</b>	
<b>Rango de amplitud</b>	Para frecuencias menores de 10 MHz:

	2 mVpp – 20 Vpp
<b>Resolución de la amplitud</b>	1 mV
<b>Estabilidad de la amplitud</b>	+/- 0.5% / 5h
<b>Offset DC</b>	
<b>Rango de ajuste de offset</b>	Para amplitudes >4V: -9.99V – 9.99V
<b>Resolución del offset</b>	0.01 V
<b>Parámetros técnicos generales</b>	
<b>Pantalla</b>	TFT color LCD de 2.4 pulgadas
<b>Interfaz</b>	Modo: USB a serial
	Velocidad de comm: 115200 bps
	Protocolo comm: Command-line mode, open protocol
<b>Alimentación</b>	5 VDC +/- 0.5V
<b>Características de operación</b>	Control total por botonera, regulador continuo
<b>Condiciones de trabajo</b>	Temperatura: 0-40 °C, Humedad: <80%

Este dispositivo generador ha sido utilizado para probar las capacidades del módulo Advantech USB-5830 y llevarlas hasta el límite real de aplicación, ya que las medidas ofrecidas por el fabricante son poco concretas y generalistas. Ha servido para crear un modelo experimental del dispositivo en funcionamiento.

#### 4.3.1.4. *Generador de funciones*

Para las pruebas y ensayos de simulación con el módulo Advantech USB-5830, se ha utilizado un generador de funciones RIGOL DG1022Z, disponible en el laboratorio B2 del edificio adjunto de la Escuela de Ingenierías.





Figura 4.12 - Generador de funciones RIGOL DG1022Z (Fuente: [44])

El RIGOL DG1022Z es un potente generador de señales y funciones. Posee dos canales de salida que pueden generar frecuencia de hasta 25 MHz. También incluye una entrada con funciones de conteo o lectura de señales, según se programe.

Este generador puede realizar hasta doce funciones distintas, que van desde el control de estándar de los canales, pasando por la generación personalizada de formas de onda, barridos de señales, modulación y generación de armónicos. Todas estas características se pueden realizar en variedad de tipos de ondas entre las que se encuentran: sinodal, cuadrada, triangular, pulso y ruido aleatorio.

Toda la información y configuración se puede visualizar y cambiar desde su pantalla LCD de 3.5 pulgadas a color y su amplio panel de botones. A mayores, posee un puerto USB y un puerto RJ45 LAN para poder conectarse desde un ordenador, ya sea directamente o por red, y desde el cual se puede controlar, programar y visualizar su estado y configuración.

Para ello es necesario la instalación del driver NI-VISA[45]. Para su control se ha utilizado la librería de Python pyvisa[46], que permite la interacción y programación de este tipo de dispositivos de manera sencilla

A continuación, se muestra una tabla resumen de las características[47] del dispositivo:

Tabla 4.3 - Datasheet del generador de funciones RIGOL DG1022Z

<b>Modelo</b>	
<b>Nº canales</b>	2
<b>Frecuencia Máxima</b>	25 MHz
<b>Ratio de muestreo</b>	200 MSa/s
<b>Formas de onda</b>	
<b>Formas de onda básicas</b>	Sinodal, Cuadrada, Rampa, Pulso, Ruido
<b>Formas de onda incluidas</b>	160 tipos (Exponencial, Gauss, Lorentz...)
<b>Características de la señal: Cuadrada</b>	
<b>Tiempo de subida y bajada</b>	< 10 ns
<b>Sobresaliente (overshoot)</b>	< 5%
<b>Ciclo</b>	0.01% a 99.99%
<b>Características de salida</b>	
<b>Rango</b>	< 10 MHz: 1.0 mVpp to 10 Vpp
<b>Precisión</b>	± 1 mV
<b>Resolución</b>	0.1 mVpp o 4 dígitos
<b>Características del barrido de frecuencia</b>	
<b>Formas de onda soportadas</b>	Senoidal, Cuadrada, Rampa
<b>Tipo de barrido</b>	Lineal, Logaritmico, Pasos
<b>Dirección</b>	Ascendente y descendente
<b>Frecuencia de inicio y final</b>	La misma que los límites de la forma de onda
<b>Tiempo de barrido y retorno</b>	1 ms a 500 s
<b>Especificaciones generales</b>	
<b>Voltaje de alimentación</b>	100 V a 240 V
<b>Consumo</b>	< 40 W
<b>Pantalla</b>	3.5 pulgadas TFT LCD. 16 M colores
<b>Condiciones de trabajo</b>	Temperatura : 0 - 50 °C. Humedad: < 95%
<b>Dimensiones</b>	261.5 mm × 112 mm × 318.4 mm
<b>Interfaces</b>	USB Host, USB Device, LAN

Este dispositivo generador se ha utilizado para simular cambios en la frecuencia de la señal digital muestreada simulada por el módulo de adquisición de datos Advantech USB-5830, a través de la función de barrido. De esta manera se simula el comportamiento de aceleración y frenado de un coche. Con esta herramienta se ha podido verificar de manera experimental el funcionamiento de la aplicación desarrollada

### 4.3.2. Software

#### 4.3.2.1. *Kit de desarrollo DAQNav*

El fabricante del módulo UBS-5830, la empresa Advantech, ofrece un software de instalación de nombre XNavi[48], que permite instalar en un ordenador con sistema operativo Windows compatible, las siguientes características:

- Los drivers de sus dispositivos DAQ, COM y CAN
- Un kit de desarrollo de software (SDK)
- Herramientas para la configuración y uso de los dispositivos

XNavi realiza una instalación de los paquetes desde la web oficial, con las versiones más recientes disponibles. Permite crear un paquete de instalación 'offline' para equipos que no tengan conexión a Internet y la posibilidad de elegir que paquetes se desean instalar, de esta manera solo es necesario instalar los paquetes del dispositivo que se va a usar junto a los paquetes necesarios para su correcto funcionamiento.

La idea del fabricante con su herramienta DAQNav[49] es simplificar la tarea de programación de los dispositivos de adquisición de datos (DAQ) a sus usuarios que, por normal general, es una tarea compleja debido a la variedad de aplicaciones y versiones existentes en el mercado. De esta manera se le proporciona una interfaz (API) multilenguaje y multiplataforma que es compatible con los dispositivos adquiridos y que puede ser implementado en cualquier sistema compatible que ya posea el usuario, o bien, crear un adaptado a sus necesidades de manera fácil.

Para ello el kit de desarrollo hace uso del concepto de los componentes de software: Propiedad, Funciones y Eventos como interfaces para comunicarse con las aplicaciones.

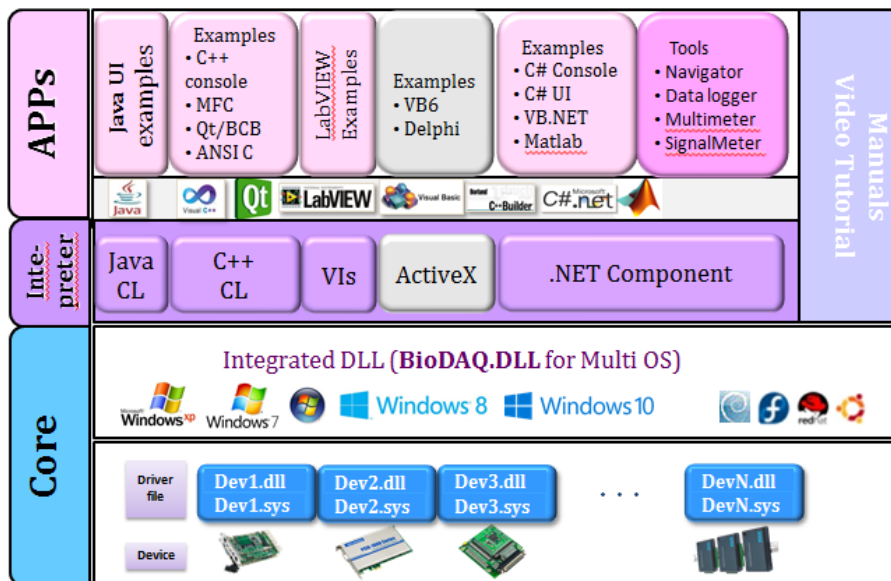


Figura 4.13 - Arquitectura del kit de desarrollo proporcionado por Advantech para el uso de sus dispositivos (Fuente: DAQNAVI Documentation)

A mayores ofrece una aplicación, Advantech Navigator que permite la visualización, configuración y prueba de manera sencilla e intuitiva de los dispositivos.

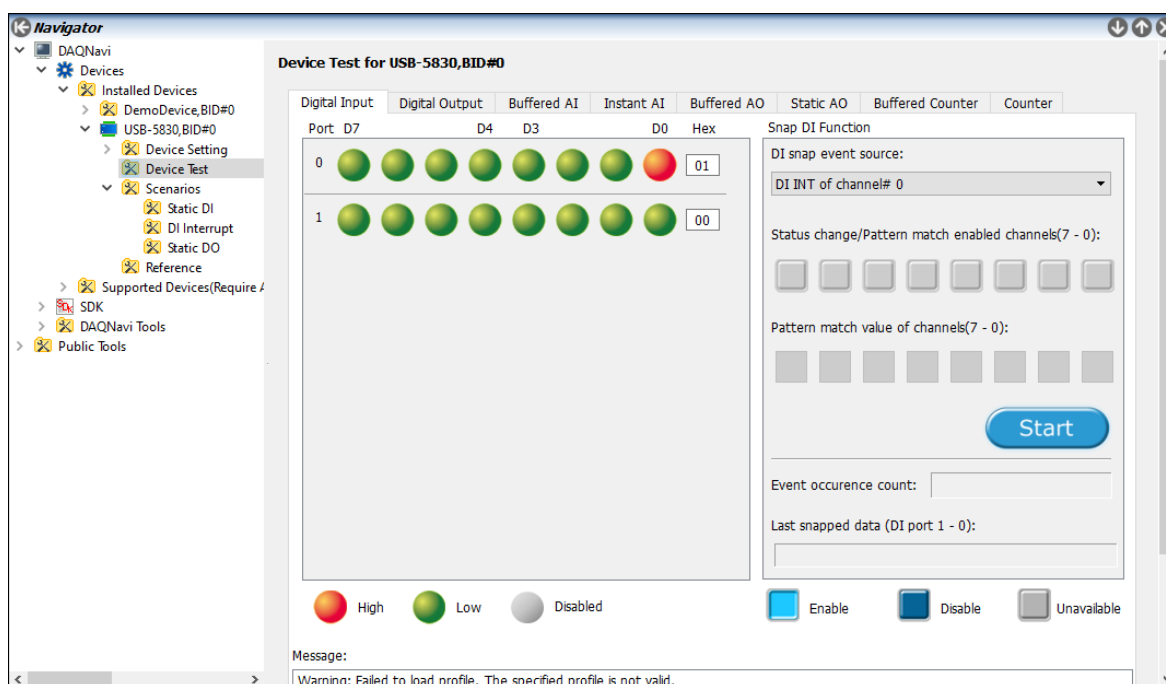


Figura 4.14 - Programa Advantech Navigator, en la pantalla de testeo de dispositivo (Fuente: propia)

DAQNAVI incluye los siguientes componentes:

- Drivers para los dispositivos DAQ
- Librerías integradas DLL

- Ensamblador DAQ
- Advantech Navigator
- Ejemplos
- Manuales

La ventaja de usar librerías DLL es que integra las diversas características de los diferentes dispositivos en una única API. De esta manera es fácil desarrollar una aplicación independientemente del hardware utilizado.

La intención es experimentar y desarrollar la aplicación para el modulo USB-5830, objeto de este proyecto, en un ordenador, pero que se pueda utilizar en otro cualquiera que cumpla con los requisitos. De esta manera la aplicación será totalmente portable en sistemas Windows.

Para un correcto desarrollo, Advantech ofrece ejemplos y manuales para facilitar el entorno de desarrollo, en distintos escenarios y herramientas.

Para el caso de usar Linux como sistema operativo, no se cuenta ni con la herramienta de instalación personalizada XNavi ni con ciertas herramientas de administración y configuración. El fabricante proporciona los drivers, las librerías de las clases y ejemplos para las distribuciones de Linux más populares. Además, proporciona el código fuentes y los paquetes binarios

#### 4.3.2.2. *Drivers*

Los drivers son las características más importantes del software proporcionados ya que permiten la comunicación entre el dispositivo y el ordenador, y junto a las librerías permiten su interacción con las aplicaciones.

Por ello deben cumplir altos estándares de seguridad y sobretodo fiabilidad. Por el ello, el fabricante asegura que sus drivers cuentan con las siguientes características: seguridad de subprocesos, respuesta en tiempo real y protección de múltiples recursos, permitiendo que el dispositivo rinda al máximo en entornos multitarea y multinucleo.

También es necesario destacar que la familia de dispositivos UBS-5800 cuenta con la tecnología AutoRecovery[50] para tratar errores inesperados y minimizar la tasa general de fallos.

#### 4.3.2.3. *Qt Creator IDE*

Para el desarrollo de la aplicación se utiliza Qt[51], que es un framework de desarrollo de aplicaciones multiplataforma tanto para escritorio como para entornos embebidos y móviles.

Como tal, Qt no es un lenguaje de programación, sino un framework escrito en C++, que incorpora un preprocesador para extender el lenguaje C++ con funciones extras que ofrece este software como, por ejemplo, la interfaz gráfica. Dicho preprocesador genera el código C++ fuente estándar, compatible con cualquier compilador, ya que el propio framework, las aplicaciones y las librerías están escritas en C++.

Los compiladores más populares como Clang, Gcc, MinGW y MSVC pueden compilar el código generado con Qt.

Qt está disponible en varios tipos de licencias. Como licencia comercial y como software libre bajo licencia GPL y LGPL. Bajo este último, se desarrolla la aplicación para este proyecto.

Qt proporciona su propio entorno de desarrollo integrado (IDE), llamada QT Creator. Se puede instalar y usar en sistemas operativos Windows, macOS y Linux. Ofrece finalización de código inteligente, resaltado de sintaxis, un sistema de ayuda integrado, depuración y también integración de sistemas de control de versiones como, por ejemplo, Git.

Una de las mayores ventajas de Qt es la posibilidad de incorporar interfaz a las aplicaciones que se desarrollan en él, de una manera sencilla, con ayuda de formularios y de herramientas de diseño como son los Widgets.

Las interfaces de usuarios se pueden escribir directamente en C++ usando módulos Widgets y se pueden diseñar mediante la herramienta interactiva Qt Designer que funciona como un generador de código para interfaces gráficas de usuario.

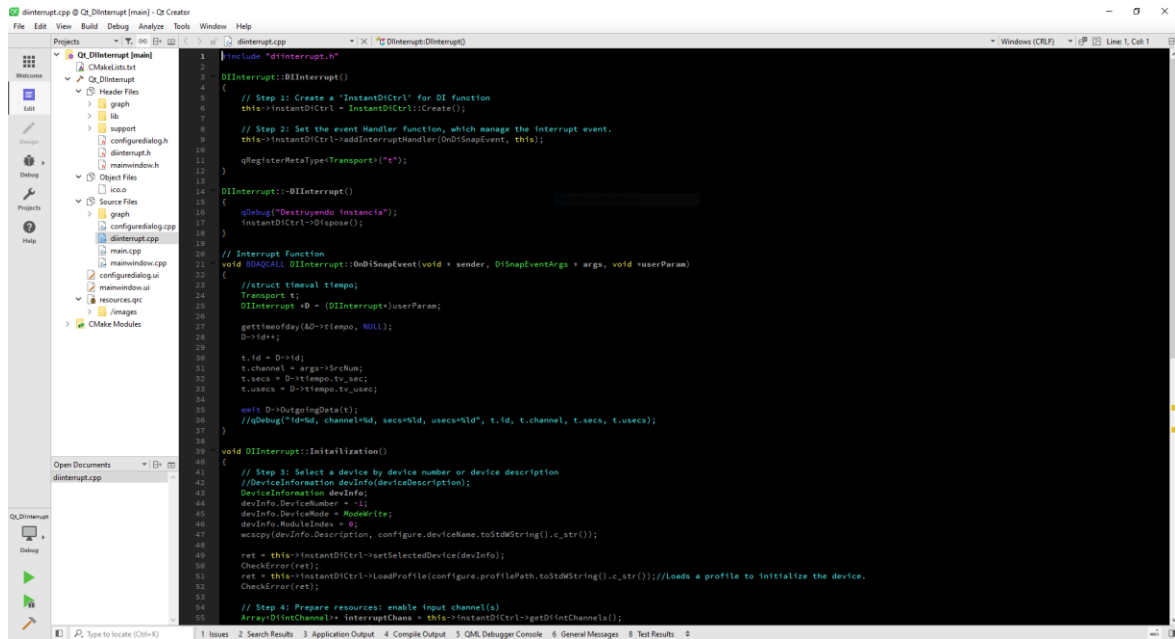


Figura 4.15 - Qt Creator IDE del proyecto actual (Fuente: propia)

Además, Qt incorpora diversos módulos que se están desarrollando por la comunidad constantemente y que permiten programar sobre áreas como las redes, bases de datos, tecnologías web, gráficos como OpenGL, protocolos de comunicación, PDF y una larga lista de módulos oficiales disponibles.

Aunque principalmente, este framework está diseñado para trabajar con el lenguaje C++, se puede desarrollar en otros lenguajes como Python gracias a vinculadores proporcionados por terceros.

La elección de Qt viene motivada por la necesidad de incorporar una interfaz gráfica a la aplicación de adquisición de datos que finalmente será escrita en el lenguaje C++, y el framework de Qt permite su desarrollo, tanto funcional como grafico de forma ágil y práctica.

#### 4.3.2.4. Lenguaje de programación C++

El lenguaje de programación C++ fue creado por Bjarne Stroustrup en 1979, con el objetivo principal de dotar al lenguaje C de características de la programación orientada a objetos.

Fue diseñado con el objetivo de facilitar la programación de proyectos reales, aunque todos los añadidos son de carácter opcional, dejando la elección de usar los subconjuntos que se deseen en el usuario.

Según el creador, estas son las nuevas características incorporadas en C++:

- Un mejor C: Se parte de un lenguaje de programación extremadamente eficiente, cercano a la máquina, pero con mejoras importantes como la comprobación de tipos, argumentos por defecto y funciones en línea.
- Abstracción de datos: Se permiten definir nuevos tipos de datos, con soporte de clases y objetos, ocultación de información y definición de espacios de nombres.
- Programación orientada a objetos: objetivo principal, disponer de clases, herencia, polimorfismo, clases abstractas y demás características propias de los lenguajes orientados a objetos.
- Programación genérica: permite definir clases y funciones genéricas mediante plantillas. Además, incorpora un potente mecanismo de sobrecarga de operadores.

Al ser su base el lenguaje C, los programas compilados en C++ son muy rápidos, compactos y utilizan pocos recursos. C++, como lenguaje de programación demostró que se podía programar a alto nivel, y si es necesario, descender al bajo nivel para implementar una rutina específica propia de C.



## 4.4. Problemas y soluciones

En este apartado se recogen los distintos problemas e incidentes ocurridos a lo largo del desarrollo del proyecto. A su vez se comentarán las soluciones descubiertas e implementadas:

### 4.4.1. Error en la instalación de una característica en el adaptador de red

Durante la instalación del kit de desarrollo proporcionado por la empresa Advantech, se pide al usuario si se desea realizar una instalación completa con todos los drivers, manuales y herramientas, o si bien, personalizar la instalación y que el usuario elija que elementos instalar.

La primera vez que se probó el dispositivo, se realiza una instalación completa que ocasiono la instalación de un driver para dispositivos Advantech que se comunican por Ethernet, y por lo tanto se debía instalar un driver específico junto a una característica en la tarjeta de red de Ethernet del equipo donde se esté instalando.

Una vez finalizada la instalación y de haber realizado el oportuno reinicio para completar la correcta instalación, la tarjeta de red del equipo no funcionaba correctamente, no tenía conectividad.

Se realizaron varios diagnósticos y se probaron distintas soluciones: ejecutar el solucionador de problemas de Windows, deshabilitar y habilitar el adaptador de red, probar configuraciones estáticas y deshabilitar el componente recién instalado. En ningún caso la conectividad se restableció.

La solución que funciono y restauro la conexión a través de cable Ethernet fue la desinstalación del adaptador de red y su reinstalación de manera genérica por el sistema operativo.

No es la primera vez que se notifica este error a la hora de instalar los drivers de Advantech, el cual ocasiona la perdida de conexión por el adaptador de cable Ethernet a los equipos.

Se recomienda encarecidamente que en el momento de instalar el kit de desarrollo de Advantech, se instale solo los drivers del módulo que se vaya a usar y los componentes

suficientes y necesarios, ya que de esta manera no se originará este fallo y la instalación será más rápida y menos pesada, en términos de cantidad de drivers instalados, ya que la empresa Advantech posee alrededor de cien módulos y tarjetas disponibles.

#### 4.4.2. Actualización driver modulo USB-5830 en Windows 10

El pasado [día y mes] se lanzó una actualización del firmware del dispositivo que se está utilizando en este proyecto, el modulo USB-5830. Y a su vez, una nueva versión del driver para su comunicación con el ordenador.

A pesar de que la empresa Advantech asegura que sus dispositivos son compatibles con versiones de Windows como XP, 7 u 8, esta nueva actualización a restringido el uso del módulo exclusivamente a la versión de Windows 10.

Se entiende este cambio de política por parte del fabricante como una manera de permitir su uso también de este dispositivo relativamente moderno a la última versión del sistema operativo Windows, Windows 11, que ha sido lanzada este año 2022.

Esta actualización supone un problema para la portabilidad del programa que se está desarrollando para este dispositivo y requiere de una comprobación y actualización de las librerías de apoyo.

#### 4.4.3. Frecuencias y voltajes de muestreo

Según especifica el fabricante, la familia de módulos de adquisición de datos USB-5000 puede muestrear hasta 800 KS/s, una cifra muy genérica y sin ningún contexto. Da a entender que los dispositivos son capaces de muestrear hasta 800 kHz de frecuencia. Lo cual resulta falso.

Según la investigación realizada con el dispositivo de este proyecto, la máxima frecuencia a la cual se puede muestrear es alrededor de 20 kHz. A partir de esa frecuencia y hasta alrededor de los 50 kHz las muestras son marcadas con un patrón de tiempo como si fuera

a frecuencia de 20 kHz. Lo cual indica que el controlador interno sí que detecta las interrupciones, pero no es capaz de procesarlas más allá de 20 kHz.

A partir de 50 kHz, la señal se detecta como una señal continua y no se genera ningún evento al respecto.

Esto indica que la máxima frecuencia teórica de muestreo es 20 kHz.

Otro inconveniente es que, según las especificaciones del dispositivo, los canales de entrada determinan si están en estado lógico 0 si en la borna la diferencia de tensión esta entre 0 y 3 V; y se marca como estado lógico 1 en el rango de 10 a 30 V.

Pero normalmente las señales digitales tienen su rango entre 0 y 5 V para representar los estados lógicos 0 y 1, respectivamente. Lo cual, a partir de la frecuencia de 10 kHz, no se detecta correctamente una señal digital común y es requerido aumentar su voltaje al menos a un valor dentro del rango que establece el fabricante.

Esto es un inconveniente, ya que es necesario la interposición de un sistema electrónico que haga esta función entre la salida de la señal y la entrada del módulo de adquisición.

#### 4.4.4. La frecuencia de resonancia interna

Durante las múltiples pruebas realizadas al dispositivo en ninguna hubo fallo o error alguno, excepto cuando se muestreaba de manera continua una señal cuadrada de exactamente 5 kHz de frecuencia. En este caso particular, pasado un tiempo arbitrario, el dispositivo fallaba internamente y quedaba totalmente bloqueado, dando la apariencia que todo seguía en correcto funcionamiento, pero en realidad el dispositivo dejaba de transmitir y se producían errores de comunicación.

Es curioso destacar que solo ocurría el bloqueo exactamente a 5 kHz, si la señal se configuraba a unos pocos hercios hacia abajo o hacia arriba, no se bloqueaba. Por lo que se cree que es la frecuencia interna de resonancia de algún componente interno que provocaría un fallo o una corrupción en el controlador interno.

La única forma de solucionar esta situación cuando surgía es el reinicio forzado del dispositivo, desconectándolo de la alimentación y volviéndolo a conectar.

A pesar de lo inoportuno de este problema, se puede pasar por esta frecuencia sin que se origine el problema, surge solo cuando se mantiene durante un tiempo en dicha frecuencia. Por lo que se recomienda no muestrear a una frecuencia continua de 5 kHz.

## 4.5. Ensayos y resultados

A lo largo del avance del proyecto, se fueron realizando pruebas de rendimiento para conocer las capacidades del módulo de adquisición de datos y también las capacidades del ordenador. Este último debido a que el método asíncrono por interrupciones requiere que los recursos y la arquitectura del programa estén preparados para ello, de lo contrario podría introducirse error en el muestreo.

Se han planteado tres tipos de pruebas que se han ejecutado en distintas etapas del proyecto:

- Ideal: con el generador de señales
- Simulación: con el generador de funciones del laboratorio
- Real: conectado al sistema ABS de un coche

### 4.5.1. Ensayo ideal

Gracias al generador de señales proporcionado por la tutora, el alumno ha podido realizar pruebas en las primeras etapas del proyecto para conocer el rendimiento del dispositivo y su comportamiento con distintos tipos de señales digitales en distintos parámetros de frecuencia y voltaje.

La configuración de estas pruebas era la siguiente: el generador estaba conectado al dispositivo por sondas, una por cada canal disponible. Y el dispositivo conectado al ordenador mediante su interfaz USB 3.0.

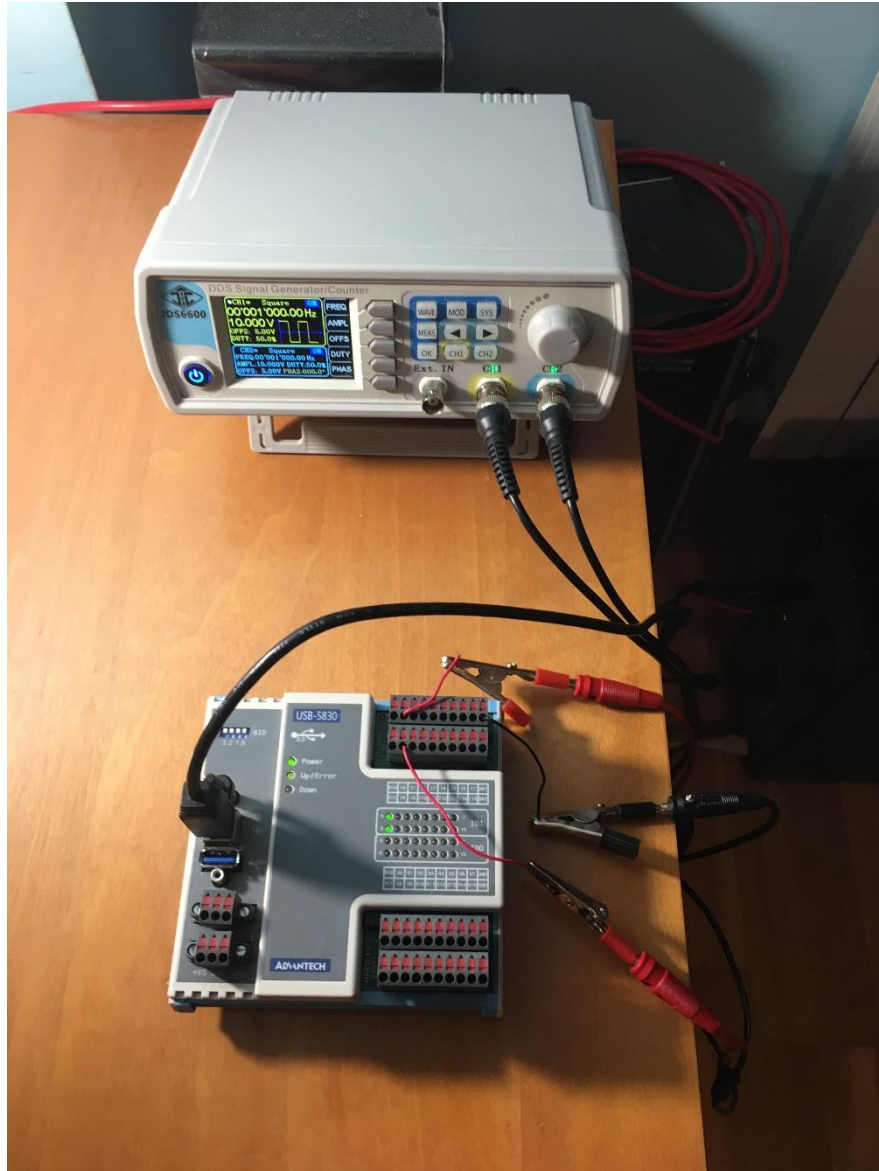


Figura 4.16 - Configuración de equipos para prueba ideal (Fuente: propia)

Estas pruebas sirven para conocer la frecuencia a la que se está muestreando ya que este generador de señales no permite realizar barridos de frecuencia y los cambios se producen girando manualmente el dial del generador, o bien, introduciendo un nuevo valor desde la aplicación de control del ordenador.

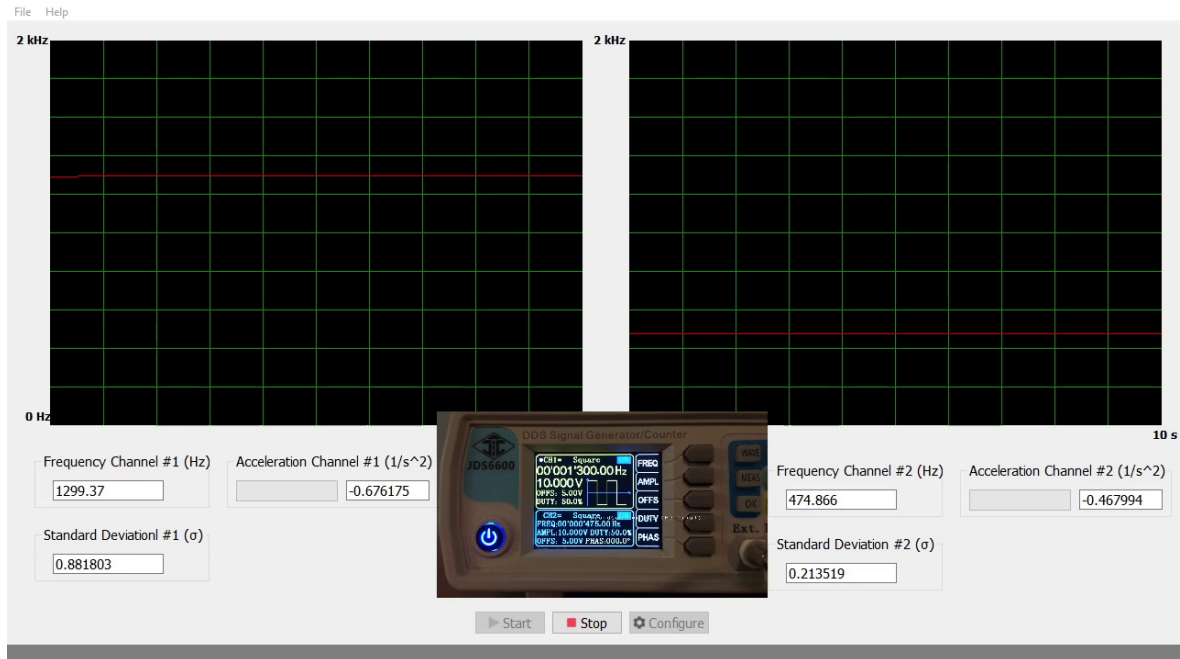


Figura 4.17 - Captura de pantalla durante el muestreo a frecuencia fija (Fuente: propia)

#### 4.5.2. Ensayo de simulación

Con el ensayo anterior se pudieron determinar tanto el comportamiento del módulo como el de la aplicación y su estabilidad a lo largo del tiempo. Pero no se pudo simular un comportamiento normal de una señal digital que cambia su frecuencia a lo largo del tiempo.

Por ello se plantea este ensayo de simulación en la que se recrea mediante un generador de funciones disponible en el laboratorio B2 del edificio adjunto de la Escuela de Ingenierías, un barrido de frecuencias de una señal digital cuadrada como parámetros muestreables.

La idea es simular el comportamiento que tendría al muestrear el sistema ABS de un coche, donde al acelerar el vehículo, la frecuencia aumenta, y al frenar, la frecuencia disminuye. De esta manera el muestreo por timestamping representa valores más reales.

En el laboratorio se montó la configuración necesaria para realizar este ensayo, el cual incluye un osciloscopio para confirmar y verificar que la señal generada cumple y varía a lo largo del tiempo.

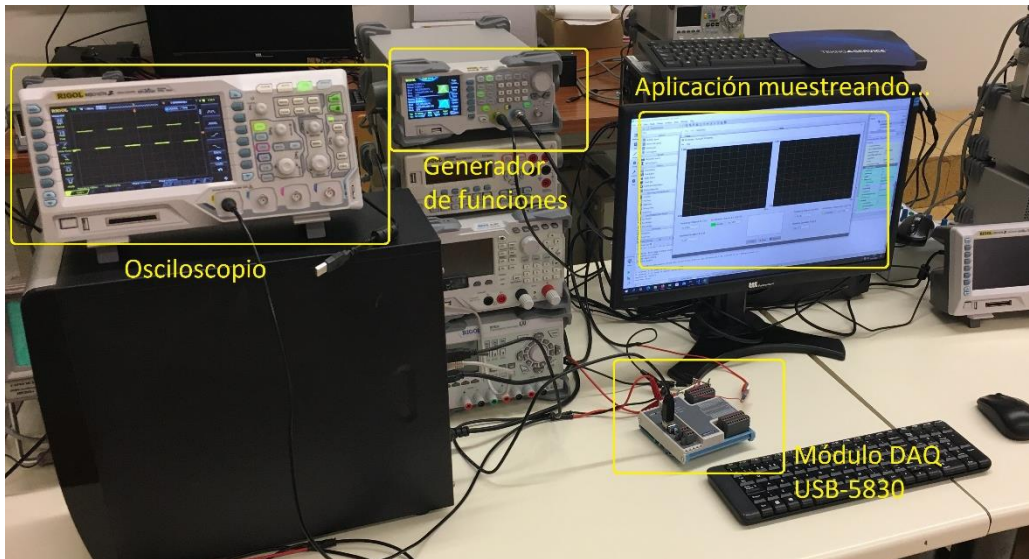


Figura 4.18 - Configuración de equipos en el laboratorio para ensayo de simulación (Fuente: propia)

Para esta prueba, se programa al generador de funciones, un barrido de frecuencia sobre una señal cuadrada. Pero en cada canal, se programa un barrido distinto en sus parámetros de duración y rango de frecuencias. De esta manera las muestras no son iguales en ambos canales y simulación una posible aceleración y frenada, de maneras más bruscas o más leves.

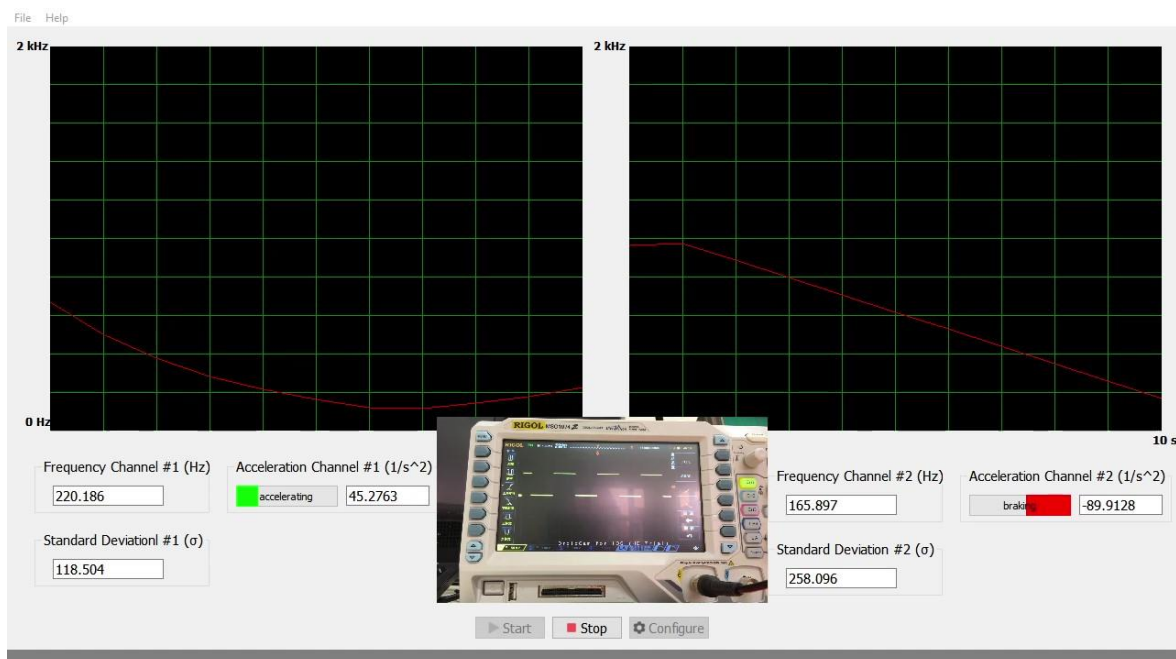


Figura 4.19 - Captura de pantalla durante el muestreo con barridos de frecuencia (Fuente: propia)



### 4.5.3. Ensayo real

Con el ensayo real sobre un sistema ABS de un vehículo, se pretende cerrar este proyecto poniendo a prueba todos los componentes desarrollados hasta el momento. Esta prueba se realiza una vez se ha acabado con toda la documentación.

Para este ensayo, se ha requerido de un componente extra, una electrónica que se conecta a la electrónica de ABS y que realiza una eliminación de ruido para que no haya falsos pulsos.

Desde ese punto ya se muestrea con el módulo de adquisición de datos a través de la aplicación.

## 5. Conclusiones y líneas futuras

En la última década, la sociedad moderna está siendo testigo de una transformación tecnológica sin precedentes gracias a la ingeniería, que es el corazón de todos los procesos y en este proyecto también ha sido el caso. El trabajo que se ha realizado para este proyecto ha sido de carácter multidisciplinar, debido a que no solo ha tocado una rama de ingeniería en concreto, si no que se han necesitado habilidades y conocimientos de más campos de la ingeniería para la investigación de los sistemas involucrados, ampliar conocimientos de campos específicos y coordinar procesos independientes con el fin de conseguir los objetivos propuesto desde el principio.

El resultado obtenido es satisfactorio y positivo, ya que se han conseguido alcanzar los objetivos propuestos desde el principio de realizar timestamping durante el muestreo de una señal digital discreta a través del módulo de adquisición de datos proporcionado.

El desarrollo de este proyecto se ha cursado sin gran problemática, con continuos pasos de avance hacia los objetivos, en donde se han ido tomado decisiones para llegar al punto donde se encuentra actualmente el proyecto.

Los datos que se obtiene durante la ejecución del muestreo son fiables, y han sido comprobado y verificado por otras herramientas. Lo cual, el resultado es un sistema de timestamping preciso, fiable y adaptable.

Como alumno titulado en Ingeniería Electrónica, Industrial y Automática e Ingeniería Informática me siento en la disposición de poder decir que ha sido un camino de autoaprendizaje en cada una de las etapas del proyecto donde he tenido la oportunidad de juntar estas dos ramas que me apasionan.

Las líneas futuras de este proyecto son numerosas debido principalmente a la flexibilidad que ofrece la empresa Advantech con sus diferentes dispositivos, ya que todos ellos pueden ser controlados y administrados con la misma API, por lo que este proyecto sienta las bases para crear nuevos sistemas de adquisición de datos con timestamping, como por ejemplo el muestreo de señales analógicas con módulos compatibles.

Por otra parte, la API proporcionada ofrece muchas más funcionalidades que las implementadas en este proyecto, aunque la mayoría depende del dispositivo que se use, muchas características permiten implementar este sistema en cualquier aplicación que se desee desarrollar, abriendo un sinfín de soluciones informáticas para adquisición y procesado de datos obtenidos de este método presentado.

Otra línea sería el procesado de los datos recogidos, el paso siguiente, que se puede realizar en tiempo real, si el sistema lo requiriese, o bien, con los datos ya guardados o almacenados para realizar estudios o análisis de los mismos.

## Lista de Referencias Bibliográficas

- [1] «Registrador de tiras de papel». [En línea]. Disponible en: <https://dewesoft.com/upload/news/daq/history/hioki-strip-chart-recorder.jpg>.
- [2] «Historia de los Sistemas de Adquisición de Datos [Actualizado 2020] | Dewesoft». [En línea]. Disponible en: <https://dewesoft.com/es/daq/historia-adquisicion-de-datos>. [Accedido: 19-may-2022].
- [3] «Sistema de Adquisición de Datos IBM 7700». [En línea]. Disponible en: <https://dewesoft.com/upload/news/daq/history/ibm-7700-data-acquisition-system.jpg>.
- [4] «Computadora Macintosh que ejecuta el entorno de programación LabView de National Instrument y tarjetas de adquisición de datos». [En línea]. Disponible en: <https://dewesoft.com/upload/news/daq/history/apple-macintosh-ni-labview.jpg>.
- [5] «Industrial Data Acquisition Market Size, Growth | Industry Outlook 2021 to 2026 With COVID Impact - Mordor Intelligence». [En línea]. Disponible en: <https://www.mordorintelligence.com/industry-reports/global-data-acquisition-daq-market-industry>. [Accedido: 19-may-2022].
- [6] «¿Qué es la Industria 4.0 y cómo funciona? | IBM». [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/industry-4-0>. [Accedido: 04-jul-2022].
- [7] K. Schalk, K. Muller, K. Peter, y J. Sebald, «Microsecond-precision time stamping in a deterministic distributed sensor network utilizing openPOWERLINK», *2017 IEEE Int. Conf. Wirel. Sp. Extrem. Environ. WiSEE 2017*, pp. 52-56, 2017.
- [8] T. Qiu, X. Liu, M. Han, H. Ning, y D. O. Wu, «A Secure Time Synchronization Protocol Against Fake Timestamps for Large-Scale Internet of Things», *IEEE Internet Things J.*, vol. 4, n.º 6, pp. 1879-1889, 2017.
- [9] «metodología | Definición | Diccionario de la lengua española | RAE - ASALE». [En línea]. Disponible en: <https://dle.rae.es/metodología>. [Accedido: 28-abr-2022].

- [10] J. S. y D. T. Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, «Principios del Manifiesto Ágil», *Agil. Manif. Page*, pp. 1-2, 2001.
- [11] K. Schwaber y J. Sutherland, «The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game», 2020.
- [12] «Esquema de la metodología SCRUM». [En línea]. Disponible en: <https://www.vbote.com/dam/jcr:ad906d21-6cf2-45c1-8b5e-f8d19e13b1c1/scrum.2018-05-04-08-05-02.png>.
- [13] EDteam, «¿Qué es SCRUM? - YouTube», 2020. [En línea]. Disponible en: <https://www.youtube.com/watch?v=sLexw-z13Fo>. [Accedido: 05-jul-2022].
- [14] «Qué es Kanban: Definición, Características y Ventajas». [En línea]. Disponible en: <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>. [Accedido: 29-abr-2022].
- [15] M. Falcón, «Gestión de Proyectos con KANBAN ? - YouTube», 2019. [En línea]. Disponible en: <https://www.youtube.com/watch?v=WP6Nt5XV980>. [Accedido: 05-jul-2022].
- [16] «Modelo de ejemplo de tablero KANBAN». .
- [17] I. Y. E. T. E. I. UPNA, «SCRUM vs KANBAN - Conceptos, Desarrollo, Similitudes, Diferencias y Aplicaciones». 2018.
- [18] «Jira | Software de seguimiento de proyectos e incidencias». [En línea]. Disponible en: <https://www.atlassian.com/es/software/jira>. [Accedido: 29-abr-2022].
- [19] «Jira - Wikipedia, la enciclopedia libre». [En línea]. Disponible en: <https://es.wikipedia.org/wiki/Jira>. [Accedido: 29-abr-2022].
- [20] «Logo de Jira». [En línea]. Disponible en: <https://logos-marcas.com/wp-content/uploads/2021/03/Jira-Logo.png>.
- [21] K. Kelley, «What is GitLab and How To use It?», 2022. [En línea]. Disponible en:

- <https://www.simplilearn.com/tutorials/git-tutorial/what-is-gitlab>. [Accedido: 29-jun-2022].
- [22] «The One DevOps Platform | GitLab». [En línea]. Disponible en: <https://about.gitlab.com/>. [Accedido: 29-jun-2022].
- [23] «Git». [En línea]. Disponible en: <https://git-scm.com/>. [Accedido: 29-jun-2022].
- [24] «Git + GitLab». [En línea]. Disponible en: [https://www.irontec.com/image/trainingsFull/50-git\\_gitlab.png](https://www.irontec.com/image/trainingsFull/50-git_gitlab.png).
- [25] «ClickUp™ | One app to replace them all». [En línea]. Disponible en: <https://clickup.com/>. [Accedido: 30-jun-2022].
- [26] «ClickUp Reviews, Demo & Pricing - 2022». [En línea]. Disponible en: <https://www.softwareadvice.com/project-management/clickup-profile/>. [Accedido: 30-jun-2022].
- [27] «Logo de ClickUp». [En línea]. Disponible en: [https://mma.prnewswire.com/media/1195692/ClickUP\\_Logo.jpg](https://mma.prnewswire.com/media/1195692/ClickUP_Logo.jpg).
- [28] «Módulo Advantech USB-4751L». [En línea]. Disponible en: [https://advanbuy.com/wp-content/uploads/2020/09/USB-4751-L\\_S-02.jpg](https://advanbuy.com/wp-content/uploads/2020/09/USB-4751-L_S-02.jpg).
- [29] Advantech, «Datasheet USB-4751L», p. 4751, 2018.
- [30] «Módulo Advantech USB-5830». [En línea]. Disponible en: [https://th.element14.com/productimages/large/en\\_US/49AC8829-40.jpg](https://th.element14.com/productimages/large/en_US/49AC8829-40.jpg).
- [31] «Programación imperativa: ventajas y desventajas del paradigma - IONOS». [En línea]. Disponible en: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/programacion-imperativa/>. [Accedido: 11-jul-2022].
- [32] Qt, «The Meta-Object System | Qt Core 5.15.10», 2022. [En línea]. Disponible en: <https://doc.qt.io/qt-5/metaobjects.html>. [Accedido: 05-jul-2022].
- [33] Qt, «Signals & Slots | Qt Core 5.15.10», 2022. [En línea]. Disponible en: <https://doc.qt.io/qt-5/signalsandslots.html>. [Accedido: 05-jul-2022].

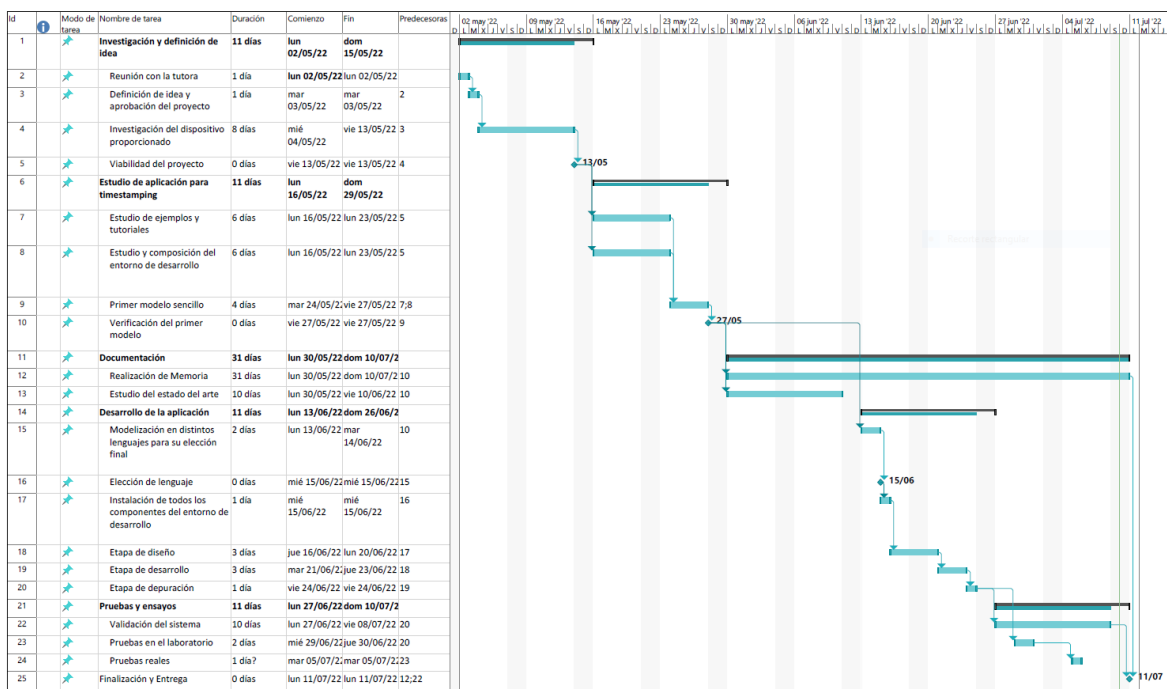
- [34] «Diagrama conceptual de uso de Signals & Slots de Qt». [En línea]. Disponible en: <https://doc.qt.io/qt-5/images/abstract-connections.png>.
- [35] Advantech, «USB-5830 - 16-ch Isolated DI and 16-ch Isolated DO USB 3.0 I/O Module - Advantech». [En línea]. Disponible en: [https://www.advantech.eu/products/1-2mlkno/usb-5830/mod\\_272f4bb0-0071-4b43-8ac9-5cc5faf62809](https://www.advantech.eu/products/1-2mlkno/usb-5830/mod_272f4bb0-0071-4b43-8ac9-5cc5faf62809). [Accedido: 03-may-2022].
- [36] «Módulo Advantech USB-5830 con desglose de sus conexiones». [En línea]. Disponible en: [https://community.element14.com/resized-image/\\_\\_size/955x639/\\_\\_key/roadtestreviewfiles/25618cb8f988429dafd820a3bb2068e7/6355.contentimage\\_5F00\\_162301.jpg](https://community.element14.com/resized-image/__size/955x639/__key/roadtestreviewfiles/25618cb8f988429dafd820a3bb2068e7/6355.contentimage_5F00_162301.jpg).
- [37] Advantech, «User Manual ADAM-6200 Series Intelligent Ethernet I/O Module ADAM-6200 User Manual ii», p. 220, 2018.
- [38] Advantech, «Datasheet USB-5830», p. 5860, 2018.
- [39] Advantech, «DAQNavi Driver for Linux - Advantech Support». [En línea]. Disponible en: <https://www.advantech.eu/support/details/driver?id=1-LXHFQJ>. [Accedido: 04-may-2022].
- [40] Joy-IT, «JDS6600 | Joy-IT». [En línea]. Disponible en: <https://joy-it.net/en/products/JT-JDS6600>. [Accedido: 09-may-2022].
- [41] «Generador de señales JT-JDS6600». [En línea]. Disponible en: <http://img.pccomponentes.com/articles/23/233065/jt-jds6600-joyit-logo1.jpg>.
- [42] E. Murphy, «Ask The Application Engineer—33 All About Direct Digital Synthesis», p. 1, 2004.
- [43] «OPERATING MANUAL JDS6600 Series Digital Control Dual-channel DDS Signal Generator», 2017.
- [44] «Rigol DG1022Z». [En línea]. Disponible en: <https://static.eleshop.nl/mage/media/catalog/product/cache/10/thumbnail/800x/040ec09b1e35df139433887a97daa66f/1/0/1022z.jpg>.

- [45] «Descargar NI-VISA - NI». [En línea]. Disponible en: <https://www.ni.com/es-es/support/downloads/drivers/download.ni-visa.html#442805>. [Accedido: 11-jul-2022].
- [46] «PyVISA: Control your instruments with Python — PyVISA 1.12.1.dev1+g93329a3 documentation». [En línea]. Disponible en: <https://pyvisa.readthedocs.io/en/latest/>. [Accedido: 11-jul-2022].
- [47] I. RIGOL TECHNOLOGIES, «Datasheet DG1000ZSeries», 2016.
- [48] Advantech, «XNavi, DAQ/COM/CAN series drivers/SDK/utilities installation and repack tool - Advantech Support - Advantech». [En línea]. Disponible en: <https://www.advantech.eu/support/details/driver?id=1-1YPCECD>. [Accedido: 12-may-2022].
- [49] Advantech, «DAQNavi Component Based DAQ Device Programming Model».
- [50] Advantech, «Advantech Launches Industry's First Robust SuperSpeed USB 3.0 Digital I/O Modules - Advantech». [En línea]. Disponible en: <https://www.advantech.eu/resources/news/2f3fbd09-86b5-4ae4-8539-11b69f3856c5>. [Accedido: 12-may-2022].
- [51] Qt, «About Qt/es - Qt Wiki», 2022. [En línea]. Disponible en: [https://wiki.qt.io/About\\_Qt/es](https://wiki.qt.io/About_Qt/es). [Accedido: 13-may-2022].



# Anexo A. Planificación SCRUM

<b>Sprint 1</b> 2 mayo a 15 mayo	<b>Investigación y definición de idea:</b> Aprobación del proyecto, investigación del dispositivo proporcionado y viabilidad del proyecto.
<b>Sprint 2</b> 16 mayo a 29 mayo	<b>Estudio de aplicación para timestamping:</b> Estudio de ejemplos, tutoriales y entorno de desarrollo. Primer modelo sencillo.
<b>Sprint 3</b> 30 mayo a 12 junio	<b>Documentación:</b> Realización de memoria y estudio del estado de arte
<b>Sprint 4</b> 13 junio a 26 junio	<b>Desarrollo de la aplicación:</b> Elección de lenguaje. Instalación de dependencias. Diseño, desarrollo y depuración.
<b>Sprint 5</b> 27 junio a 10 julio	<b>Pruebas y ensayos:</b> Validación del sistema, pruebas de laboratorio y reales.



## Anexo B. Capturas de pantalla de las ventanas del programa



Figura B.0.1 - Icono de la aplicación (Fuente: propia)

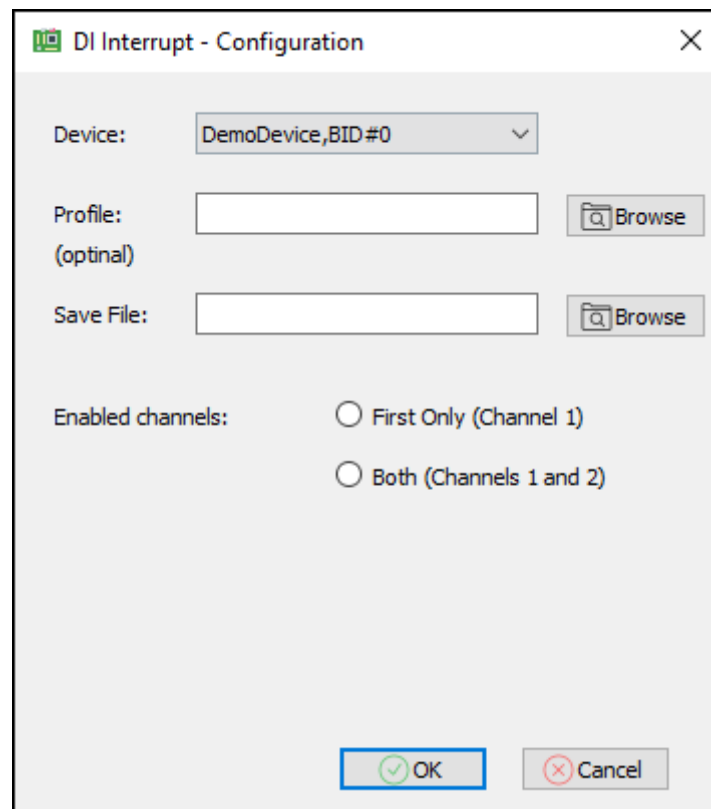


Figura B.0.2 - Captura del dialogo de configuración (Fuente: propia)

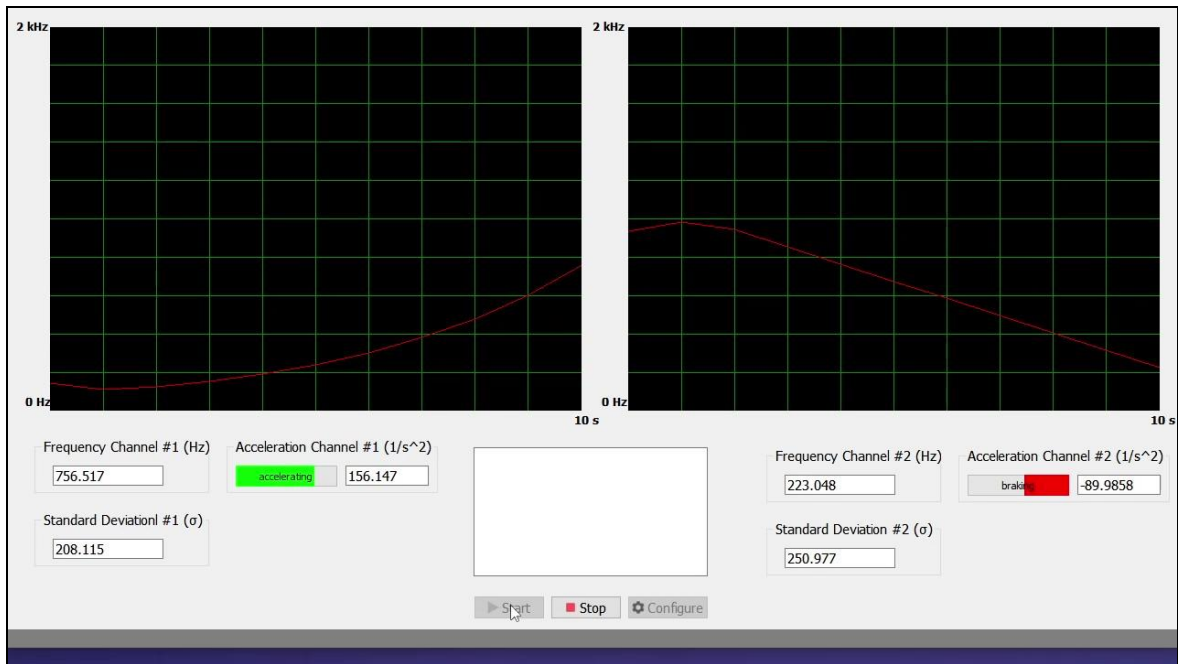


Figura B.0.3 - Captura de la ventana principal del programa (Fuente: propia)

# Anexo C. Código del programa: Simple\_DIInterrupt.c

```

/*****
Simple_DIInterrupt.c

Author: Álvaro Vila Flecha
Date: 20th May 2022
*****/

#include <stdio.h>
#include <stdlib.h>
#include <wchar.h>
#include <time.h>
#include "compatibility.h"
#include "inc/bdaqctrl.h"

// Parameters of device
#define deviceDescription L"USB-5830,BID#0"
const wchar_t *profilePath = L"./AllRising.xml";

// Global variables
FILE *outputFile;
int iter = 0;
struct timeval tv;

// This is function declaration. the function is used to deal with
'Interrupt' Event.
void BDAQCALL OnDiSnapEvent(void *sender, DiSnapEventArgs *args, void
*userParam);

int main(int argc, char *argv[])
{
    ErrorCode ret = Success;
    DiintChannel *diIntChannel = NULL;
    DiintChannel *diIntChannel2 = NULL;
    IArray *interruptChans = NULL;
    wchar_t enumString[256];

    if (argc == 2) {
        outputFile = fopen(argv[1], "w");
    } else {
        printf("File file argument necessary: Use %s sample.txt\n",argv[0]);
        exit(1);
    }
}

```

```

}

if (outputFile == NULL) {
    printf("Error opening file!");
    exit(1);
}

// Step 1: Create a 'InstantDiCtrl' for DI function.
InstantDiCtrl *instantDiCtrl = InstantDiCtrl_Create();

// Step 2: Set the notification event Handler by which we can know the
state of operation effectively.
InstantDiCtrl_addInterruptHandler(instantDiCtrl, OnDiSnapEvent, NULL);

do {
    // Step 3: Select a device by device number or device description
and specify the access mode.
    DeviceInformation devInfo;
    devInfo.DeviceNumber = -1;
    devInfo.DeviceMode = ModeWrite;
    devInfo.ModuleIndex = 0;
    wcsncpy(devInfo.Description, deviceDescription);
    ret = InstantDiCtrl_setSelectedDevice(instantDiCtrl, &devInfo);
    CHK_RESULT(ret);
    ret = InstantDiCtrl_LoadProfile(instantDiCtrl, profilePath); //
Loads a profile to initialize the device.
    CHK_RESULT(ret);

    // Step 4: Set necessary parameters
    interruptChans = InstantDiCtrl_getDiintChannels(instantDiCtrl);
    if (interruptChans == NULL) {
        printf(" The device doesn't support DI interrupt!\n");
        waitAnyKey();
        return 0;
    }
    // First channel
    diIntChannel = (DiintChannel *)Array_getItem(interruptChans, 0);
    ret = DiintChannel_setEnabled(diIntChannel, TRUE);
    CHK_RESULT(ret);

    //Second channel
    diIntChannel2 = (DiintChannel *)Array_getItem(interruptChans, 1);
    ret = DiintChannel_setEnabled(diIntChannel2, TRUE);
    CHK_RESULT(ret);

    printf(" DI channel %d is used to detect interrupt!\n",
DiintChannel_getChannel(diIntChannel));

```

```

    printf(" DI channel %d is used to detect interrupt!\n",
DiintChannel_getChannel(diIntChannel2));

    // Step 5: Start DIInterrupt
    ret = InstantDiCtrl_SnapStart(instantDiCtrl);
    CHK_RESULT(ret);

    // Step 6: The device is working.
    printf(" Snap has start, any key to quit !\n");
    do { SLEEP(1); } while (!kbhit());

    // Step 7: Stop DIInterrupt
    ret = InstantDiCtrl_SnapStop(instantDiCtrl);
    CHK_RESULT(ret);
    fclose(outputFile);

} while (FALSE);

// Step 8: Close device, release any allocated resource.
InstantDiCtrl_Dispose(instantDiCtrl);

// If something wrong in this execution, print the error code on screen
for tracking.
if (BioFailed(ret)) {
    AdxEnumToString(L"ErrorCode", (int32)ret, 256, enumString);
    printf("Some error occurred. And the last error code is 0x%X.
[%!s]\n", ret, enumString);
    waitAnyKey();
}
return 0;
}

void BDAQCALL OnDiSnapEvent(void *sender, DiSnapEventArgs *args, void
*userParam) {
    // double t = wallclock();
    gettimeofday(&tv, NULL);
    iter++;

    fprintf(outputFile, "%d,%d,%ld,%ld\n", iter, args->SrcNum,
(long)tv.tv_sec, (long)tv.tv_usec);
}

```

# Anexo D. Código del programa: Clase C++ DIInterrupt

```

#include "diinterrupt.h"

DIInterrupt::DIInterrupt() {

    this->instantDiCtrl = InstantDiCtrl::Create();
    this->instantDiCtrl->addInterruptHandler(OnDiSnapEvent, this);
}

DIInterrupt::~DIInterrupt() {
    if (fclose(salida) != EOF) printf("\nArchivo cerrando correctamente\n");
    instantDiCtrl->Dispose();
    waitAnyKey();
}

void BDAQCALL DIInterrupt::OnDiSnapEvent(void * sender, DiSnapEventArgs *
args, void *userParam)
{
    DIInterrupt *D = (DIInterrupt*)userParam;

    gettimeofday(&D->tiempo, NULL);
    D->id++;

    fprintf(D->outputFile, "%d,%d,%ld,%ld\n", D->id, args->SrcNum, (long)D-
>tiempo.tv_sec, (long)D->tiempo.tv_usec);
}

void DIInterrupt::Inicializar() {
    //DeviceInformation devInfo(deviceDescription);
    DeviceInformation devInfo;
    devInfo.DeviceNumber = -1;
    devInfo.DeviceMode = ModeWrite;
    devInfo.ModuleIndex = 0;
    wcsncpy(devInfo.Description, deviceDescription);

    ret = this->instantDiCtrl->setSelectedDevice(devInfo);
    CheckError(ret);
    ret = this->instantDiCtrl->LoadProfile(profilePath);
    CheckError(ret);
}

```

```

    Array<DiintChannel>* interruptChans = this->instantDiCtrl-
>getDiintChannels();
    if (interruptChans == NULL) {
        printf("The device doesn't support DI interrupt!\n");
        waitAnyKey();
        return;
    }

    interruptChans->getItem(0).setEnabled(true);
    printf("DI channel %d is used to detect interrupt!\n", interruptChans-
>getItem(0).getChannel());

    if (n_canales == 2){
        interruptChans->getItem(1).setEnabled(true);
        printf("DI channel %d is used to detect interrupt!\n",
interruptChans->getItem(1).getChannel());
    }
}

void DIInterrupt::IniciarCaptura() {
    ret = this->instantDiCtrl->SnapStart();
    CheckError(ret);
    printf("Snap has start, any key to quit !\n");
}

void DIInterrupt::DetenerCaptura() {
    ret = this->instantDiCtrl->SnapStop();
    CheckError(ret);
    printf("Snap stopped !\n");
}

void DIInterrupt::CheckError(ErrorCode errorCode) {
    if (BioFailed(errorCode)) {
        AdxEnumToString(L"ErrorCode", (int32)ret, 256, enumString);
        printf("Some error occurred. And the last error code is 0x%X.
[%1s]\n", ret, enumString);
        waitAnyKey();
        exit(0);
    }
}
}

```