

# Trabajo Fin de Grado

## Ingeniería de Telecomunicación

### Desarrollo de una plataforma de bajo coste para medir convertidores ADC Sigma-Delta

Autor: Francisco José Barrero Serrano

Tutores: José María Hinojo Montero

Fernando Muñoz Chavero

**Dpto. Teoría de la Señal y Comunicaciones**  
**Escuela Técnica Superior de Ingeniería**  
**Universidad de Sevilla**

Sevilla, 2023



**DINEL**  
DPTO. INGENIERÍA ELECTRÓNICA



Proyecto Fin de Carrera  
Ingeniería de Telecomunicación

# **Desarrollo de una plataforma de bajo coste para medir convertidores ADC Sigma-Delta**

Autor:

Francisco José Barrero Serrano

Tutores:

José María Hinojo Montero

Investigador postdoctoral Talento

Fernando Muñoz Chavero

Catedrático de Universidad

Dpto. Ingeniería Electrónica  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2023



Trabajo Fin de Grado: Desarrollo de una plataforma de bajo coste para medir convertidores ADC Sigma-Delta

Autor: Francisco José Barrero Serrano

Tutor: José María Hinojo Montero  
Fernando Muñoz Chavero

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2013

El Secretario del Tribunal



*A mi familia*

*A mis amistades*





# Agradecimientos

---

Con este Trabajo de Fin de Grado, se cierra un ciclo bonito y a la vez complicado del cual han formado parte numerosas personas cercanas a mí. Quiero agradecer, por tanto, a todas esas personas, empezando por mis amigos, por el apoyo incondicional y por todo lo vivido, y en especial a mi familia, por esa paciencia y aguante, haciendo una mención especial a mi padre, que tiene el cielo ganado.

Finalmente, agradecer también a mi tutor, José María Hinojo, por todo lo que me ha ayudado y al cual espero no haber desquiciado mucho a lo largo de estos meses.

*Francisco José Barrero Serrano*

*Grado en Ingeniería de las Telecomunicaciones*

*Sevilla, 2023*



Este trabajo fin de grado se ha centrado en el diseño, desarrollo e implementación de un sistema automatizado de caracterización de convertidores analógico-digital (ADC) de tipo Sigma-Delta. El sistema emplea la instrumentación electrónica necesaria (fuente de alimentación, generador de onda y dispositivo generador de patrones con funcionamiento como analizador lógico) con capacidad de ser controlada en modo remoto. Se ha desarrollado un software de manejo de la instrumentación empleando el lenguaje Python, para lo cual se ha utilizado una API específica de manejo de instrumentación virtual en modo remoto (VISA) denominado PyVISA, que es un paquete de software libre asociado a Python que permite controlar todo tipo de dispositivos de medida, independientemente de la interfaz utilizada para su control remoto.

Para desarrollar el sistema, se han estudiado en detalle los ADC Sigma-Delta, un tipo de convertidor ampliamente utilizado en el mundo de las telecomunicaciones, con idea de conocer los componentes principales que lo forman (Filtro Anti-aliasing, Decimador y Modulador Sigma-Delta) y explorar las métricas fundamentales que nos permiten caracterizarlo (Relación Señal-Ruido o SNR, Distorsión Armónica Total o THD, Relación Señal a Ruido y Distorsión o SINAD, Distorsión de Espurias de Frecuencia o SFDR y el Número de Bits Efectivos o ENOB).

Destacar que la herramienta realizada me ha permitido profundizar en el conocimiento de tecnologías estudiadas en diferentes asignaturas del grado (programación en Python, manejo de instrumentación electrónica, ADC de tipo Sigma-Delta) más allá de lo transmitido en las clases, desarrollando un proyecto realista y práctico; un banco de pruebas que será útil en el futuro desarrollo de convertidores que se diseñen e implementen en el Grupo de investigación de Ingeniería Electrónica, del Departamento de Ingeniería Electrónica de la Escuela Técnica Superior de Ingenieros de la Universidad de Sevilla.



# Abstract

---

This final degree project has focused on the design, development, and implementation of an automated system for the characterization of Sigma-Delta analog-to-digital converters (ADCs). The system utilizes the necessary electronic instrumentation (power supply, waveform generator, and pattern generator device with logic analyzer functionality) with the capability of being controlled remotely. A management software for the instrumentation has been developed using the Python language, for which a specific remote virtual instrumentation API (VISA) called PyVISA has been used. PyVISA is an open-source software package associated with Python that allows the control of various types of measurement devices, regardless of the interface used for remote control.

To develop the system, Sigma-Delta ADCs have been studied in detail. These are a type of converter widely used in the telecommunications field. The aim was to understand the main components that constitute it (Anti-aliasing Filter, Decimator, and Sigma-Delta Modulator) and explore the fundamental metrics that allow us to characterize it (Signal-to-Noise Ratio or SNR, Total Harmonic Distortion or THD, Signal to Noise and Distortion Ratio or SINAD, Spurious Free Dynamic Range or SFDR, and Effective Number of Bits or ENOB).

It's worth noting that the developed tool has allowed me to delve deeper into the knowledge of technologies studied in various subjects of the degree (Python programming, electronic instrumentation management, Sigma-Delta ADCs) beyond what was taught in classes. This involved creating a realistic and practical project—a test bench that will be useful for future development of converters designed and implemented within the Electronic Engineering Research Group of the Department of Electronic Engineering at the Higher Technical School of Engineers of the University of Seville.

# Índice

---

<b>Agradecimientos</b>	<b>ix</b>
<b>Resumen</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Índice</b>	<b>xiv</b>
<b>Índice de Tablas</b>	<b>xvi</b>
<b>Índice de Figuras</b>	<b>xviii</b>
<b>Notación</b>	<b>xxii</b>
<b>1 Introducción</b>	<b>1</b>
<b>2 Arquitectura de un modulador Sigma-Delta y sus principales características</b>	<b>7</b>
2.1. <i>El convertidor Sigma-Delta</i>	11
2.1.1 Filtro Anti-aliasing	14
2.1.2 Decimador	15
2.1.3 Modulador Sigma-Delta	15
2.2. <i>Caracterización de un ADC</i>	15
2.1.4 SNR	17
2.1.5 THD	18
2.1.6 SINAD	18
2.1.7 SFDR	18
2.1.8 ENOB	18
2.1.9 Relaciones entre las magnitudes anteriores	19
2.2 <i>Protocolos de Instrumentación</i>	23
2.2.1 DIN-MESSBUS	24
2.2.2 GPIB	24
2.2.3 LXI	24
2.2.4 HART	25
2.2.5 M-BUS o EN13757	25
2.2.6 Otros	25
<b>3 Descripción del Banco de Pruebas</b>	<b>27</b>
3.1 <i>Instrumentación electrónica del Banco de Pruebas</i>	28
3.1.1 Keithley 2281-S	28
3.1.2 Tektronix AFG1022	29
3.1.3 Digital Discovery	33
3.2 <i>Sigma-Delta a evaluar</i>	37
3.3 <i>Herramienta para la gestión del banco de pruebas</i>	40
<b>4 Implementación de la Herramienta de Control y Procesamiento</b>	<b>41</b>
4.1 <i>Configuración de la Herramienta de control y procesado</i>	42
4.2 <i>Módulo de configuración de los instrumentos</i>	45
4.2.1 Generador de los puntos del espacio de prueba	45
4.2.2 Configuración del generador de patrones: temporización	46
4.2.3 Configuración del generador de funciones	48

4.2.4	Configuración de la fuente de alimentación	48
4.3	<i>Módulo de adquisición</i>	49
4.4	<i>Módulo de procesado</i>	52
4.4.1	Herramienta de procesado	52
4.5	<i>Ejecución de la herramienta</i>	54
<b>5</b>	<b>Resultados</b>	<b>55</b>
5.1	<i>Validación de las señales de control</i>	55
5.2	<i>Validación del procesado</i>	60
5.3	<i>Validación del sistema de adquisición</i>	62
<b>6</b>	<b>Conclusiones</b>	<b>65</b>
6.1	<i>Líneas futuras</i>	65
	<b>Referencias</b>	<b>67</b>

# ÍNDICE DE TABLAS

---

Tabla 2-1. LSB en ADC con diferentes resoluciones y rango de conversión de 1V	17
Tabla 3-1. Características principales del equipo Keithley 2281-S	29
Tabla 3-2. Características principales del equipo Tektronix AFG1022	31
Tabla 4-1. Descripción de los parámetros definidos	43





# ÍNDICE DE FIGURAS

---

Figura 1-1. (a) Osciloscopio analógico. (b) Diagrama de bloques de un osciloscopio analógico.	1
Figura 1-2. (a) Osciloscopio digital. (b) Diagrama de bloques de un osciloscopio digital.	2
Figura 2-1. ADC de tipo flash y resolución de 3 bits.	7
Figura 2-2. ADC de tipo rampa simple.	8
Figura 2-3. ADC de tipo rampa doble.	8
Figura 2-4. ADC de aproximaciones sucesivas de 4 bits.	9
Figura 2-5. Ejemplo de funcionamiento del CAD de aproximaciones sucesivas.	9
Figura 2-6. ADC pipeline de 12 bits y 4 etapas.	10
Figura 2-7. Error de cuantización en un ADC.	10
Figura 2-8. Ejemplo de error de cuantización (EQ) en una conversión.	11
Figura 2-9. Comparativa entre diferentes tecnologías de ADCs.	12
Figura 2-10. Bloques funcionales de un ADC de tipo Sigma-Delta ( $\Sigma$ - $\Delta$ ).	12
Figura 2-11. Modulador Sigma-Delta de un bit.	13
Figura 2-12. ADC de tipo Sigma-Delta.	14
Figura 2-13. Efecto aliasing y cómo se mitiga con filtros analógicos antes del muestreo.	14
Figura 2-14. Comparativa del efecto aliasing entre ADC sin sobremuestreo (figura superior) y con sobremuestreo (figura inferior), como es el caso de un ADC Sigma Delta	15
Figura 2-15. Ejemplos para diferentes tiempos de conversión	16
Figura 2-16. Valores típicos de la velocidad de conversión (inversa de tiempo de conversión) y resolución en diferentes tipos de ADCs: comparativa	16
Figura 2-17. Parámetros que determinan el funcionamiento dinámico de un ADCs	19
Figura 2-18. FFT aplicada a los datos muestreados de un ADC de 12 bits. En este gráfico FFT, hay 4096 muestras que el convertidor adquiere a una velocidad de datos de 200 kmuestras/seg. La SNR (punto C en este gráfico) es un valor calculado.	20
Figura 2-19. Análisis FFT para cuantificación de las características dinámicas de un ADC	20
Figura 2-20. Errores de desplazamiento y ganancia en un ADC	21
Figura 2-21. Error DNL en un ADC	22
Figura 2-22. Error INL en un ADC	22
Figura 2-23. Ejemplo de red de instrumentación para la caracterización remota de dispositivos o sistemas (DUT)	24
Figura 2-24. Arquitecturas de aplicaciones software que utilicen el estándar VISA	26
Figura 2-25. Características generales de la aplicación software para la caracterización de un ADC	26
Figura 3-1. Banco de pruebas del Sigma-Delta	27
Figura 3-2. Fuente de alimentación Keithley 2281-S	28
Figura 3-3. Tektronix AFG1022	30
Figura 3-4. Digital Discovery	33

Figura 3-5. Instrumentación del Digital Discovery	34
Figura 3-6. Programa Waveforms	34
Figura 3-7. Diagrama de bloques DigitalIn	35
Figura 3-8. Diagrama de bloques DigitalOut	35
Figura 3-9. Registros	36
Figura 3-10. Contador	37
Figura 3-11. Arquitectura interna del ADC propuesto en [1], denominado SeP-D	38
Figura 3-12. Temporización asociada al ADC propuesto en [1]	38
Figura 3-13. Fases de control	39
Figura 3-14. Implementación del ADC propuesto en [1] y denominado SeP-D (a), y de uno de tipo P-D (b). La implementación se realizó en tecnología estándar CMOS de 180 nm	39
Figura 3-15. Prototipo de placa de circuito impreso en la que se aprecia el ADC desarrollado en la tesis doctoral de Doña Elena Cabrera	40
Figura 4-1. Diagrama de flujo que sigue la herramienta de control y procesado desarrollada.	42
Figura 4-2. Diagramas de flujo de las funciones que determinan la amplitud (izquierda) y la frecuencia (derecha) del barrido de la onda generada en función del tipo (lineal o logarítmica), los valores inicial y final, así como el paso entre señales generadas.	45
Figura 4-3. Esquema de conexión del ADC (DUT) con el Digital Discovery	46
Figura 4-4. Diagrama de flujo del generador de patrones que integra el Digital Discovery	48
Figura 4-5. Diagrama de flujo de la fuente de alimentación	49
Figura 4-6. Diagrama de flujo asociado a la configuración del generador de funciones	51
Figura 4-7. Diagrama de flujo asociado al módulo de procesamiento	53
Figura 4-8. Ejemplo de ejecución de logging	54
Figura 5-1. Esquema de medida para validar las señales de control	56
Figura 5-2. Temporización de las señales de control del ADC	56
Figura 5-3. Detalle del desfase entre Ph3 (azul) y Ph4 (rojo)	57
Figura 5-4. Detalle del retraso entre Ph5 (azul) y Ph6 (rojo)	57
Figura 5-5. Captura de osciloscopio que muestra la temporización de las fases 1, 2, 3 y 4 generadas	57
Figura 5-6. Captura del osciloscopio aplicando zoom en el desfase entre Ph3 y Ph4	58
Figura 5-7. Captura de osciloscopio que muestra la temporización de las fases 1, 2, 5 y 6 generadas	58
Figura 5-8. Captura de osciloscopio aplicando zoom en la zona de subida de las fases 5 y 6	59
Figura 5-9. Captura de osciloscopio aplicando zoom en la zona de bajada de las fases 5 y 6	59
Figura 5-10. Banco de pruebas durante la obtención de las señales de control del ADC	60
Figura 5-11. SNR y SNDR obtenida por (a) la herramienta desarrollada y (b) presentada en [8]	61
Figura 5-12. Espectro obtenido por (a) la herramienta desarrollada y (b) presentada en el trabajo [8] para una señal de entrada de 0 dBFS	61
Figura 5-13. Comparativa de los armónicos obtenidos en la herramienta desarrollada (a), frente a los obtenidos en el trabajo [8]	61
Figura 5-14. Banco de pruebas general	62
Figura 5-15. Banco de pruebas con los diferentes equipos controlados remotamente	63





# Notación

---

ADC	Convertidor analógico digital (Analog to Digital Converter)
SNR	Signal-to-noise ratio
SNDR	Signal-to-noise and distortion ratio
THD	Total harmonic distortion
SINAD	Signal-to-noise and distortion ratio
SFDR	Spurious-free dynamic range
ENOB	Effective number of bits
MSE	Minimum square error
:	Tal que
<	Menor o igual
>	Mayor o igual
\	Backslash
↔	Si y sólo si

# 1 INTRODUCCIÓN

Se denomina electrónica a la disciplina técnica y científica que se dedica al estudio y la producción de sistemas físicos basados en la conducción y el control de un flujo de electrones o partículas cargadas eléctricamente. La aparición del transistor, y gracias a él, el desarrollo de la electrónica desde mediados del siglo XX, ha impulsado una revolución tecnológica que ha transformado significativamente la manera en que los seres humanos interactúan entre sí. Este cambio ha dado origen a una variedad de dispositivos de "soporte" o "apoyo" que nos acompañan cotidianamente, y que se engloban en lo que conocemos como "electrónica de consumo".

En un primer momento de este desarrollo tecnológico, las aplicaciones del procesado analógico eran las predominantes en el mercado de la electrónica de consumo. Las señales, eminentemente de tipo analógico, se procesaban, principalmente, con técnicas analógicas, lo que implicaba que dichas señales se manipulaban mediante circuitos electrónicos compuestos por componentes pasivos (tales como resistencias, condensadores o bobinas) y activos (transistores y diodos, así como circuitos integrados formados por una combinación de éstos y con una función específica -hoy en día, a este tipo de circuitos se les denomina circuito integrado específico de aplicación o ASICs, por sus siglas en inglés-). Un ejemplo de tales dispositivos, los cuales se empleaban para el procesamiento de señales analógicas, podría ser el uso de circuitos para la estimación de trayectorias o la resolución de ecuaciones diferenciales. Estos ejemplos ilustran cómo la electrónica permitía abordar desafíos complejos. Además, entre los dispositivos más conocidos de esta época se encuentran los osciloscopios, que, si bien no son representativos de la complejidad de la electrónica, eran instrumentos esenciales para visualizar y medir señales de voltaje en tiempo real. Los osciloscopios, utilizaban un circuito de amplificación para aumentar la amplitud de la señal para posteriormente mostrarla en una pantalla gráfica, dónde entra en juego el sistema de visualización. En la figura posterior se muestra, a modo de ejemplo, un osciloscopio analógico comercializado por el fabricante Hameg.

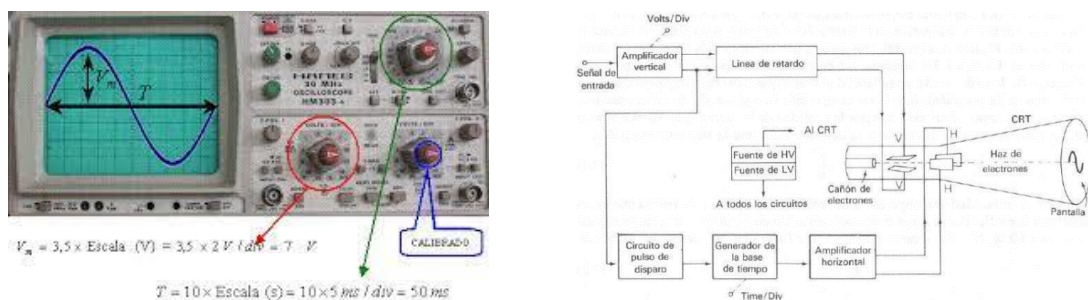


Figura 1-1. (a) Osciloscopio analógico. (b) Diagrama de bloques de un osciloscopio analógico.

Sin embargo, estos métodos presentaban limitaciones significativas, ya que la manipulación analógica de señales resultaba altamente sensible a las variaciones de componentes y a los efectos externos provocados por la interferencia electromagnética. Esta sensibilidad tenía como consecuencia un impacto considerable en la precisión de los sistemas. Además, estas técnicas enfrentaban dificultades al procesar grandes volúmenes de datos, una característica ausente en este tipo de sistemas electrónicos, lo que limitaba su aplicabilidad en contextos más complejos.

La electrónica digital (también podríamos denominarla electrónica basada en dispositivos que gestionan ceros y

unos) es una evolución de la propia tecnología electrónica que ofrece su primer gran hito con la aparición del Intel 4004, el primer microprocesador de la historia, y que se desarrolla desde la década de 1970. Aparece como una forma de mitigar los principales inconvenientes, anteriormente descritos, que ofrecen los sistemas electrónicos analógicos. Para ello, plantean el procesamiento basado en una codificación binaria de los datos obtenidos a partir de sensores o transductores analógicos, lo que hace necesaria la introducción y desarrollo de dispositivos que permitan obtener dicha codificación binaria a partir de la monitorización u observación electrónica analógica de variables y magnitudes físicas. Este tipo de dispositivos se denominan convertidores analógico-digital, ADC de sus siglas en inglés. En la figura inferior se muestra, a modo de ejemplo, un osciloscopio digital comercializado por el Hameg.

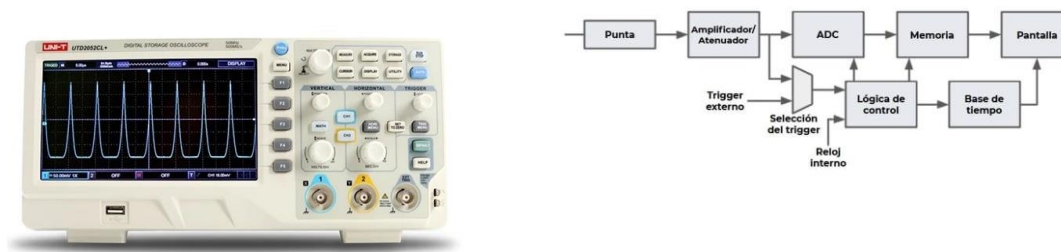


Figura 1-2. (a) Osciloscopio digital. (b) Diagrama de bloques de un osciloscopio digital.

La introducción de los ADCs (Convertidores Analógico-Digitales) marcó un hito en el ámbito del procesamiento de señales y la automatización de sistemas. Este avance permitió la conversión precisa y fiable de señales analógicas a formato digital, superando las limitaciones del procesamiento analógico que era costoso y susceptible a variaciones tecnológicas y tolerancias. La manipulación de señales analógicas conllevaba complejidades al procesar un gran número de señales, lo que resultaba costoso tanto en términos económicos como técnicos.

La electrónica digital, en contraste, posibilitó el desarrollo de aplicaciones más complejas de manera más eficiente. Sin embargo, la conexión entre el mundo analógico y el digital era un desafío fundamental. Aquí es donde entra en juego el ADC. Estos dispositivos, en esencia, permiten transformar las fluctuaciones de señales analógicas, como voltajes y corrientes cambiantes, en representaciones numéricas en formato binario. Esta conversión habilita el posterior procesamiento y almacenamiento de las señales en sistemas electrónicos digitales basados en microprocesadores, así como la interacción con el entorno real y cambiante donde las magnitudes analógicas evolucionan constantemente, impulsadas por sensores y transductores.

Mientras las señales analógicas son continuas y ofrecen un número infinito de diferentes valores de voltaje y/o corriente; los circuitos digitales trabajan únicamente con datos binarios, cuya unidad elemental es el bit y puede adoptar únicamente dos valores, “1” lógico o HIGH y “0” lógico o LOW. El ADC conforma un elemento fundamental en los sistemas electrónicos modernos, al ser el único dispositivo electrónico con capacidad de convertir del dominio analógico, adscrito a la magnitud física, al dominio digital, centrado en el procesamiento y almacenamiento de la información.

El proceso de conversión de una señal analógica a digital involucra múltiples etapas. En primer lugar, se realiza el muestreo y mantenimiento (S&H por sus siglas en inglés), donde se captura la señal analógica en un instante determinado. Luego, la señal pasa por una etapa de filtrado antialiasing para prevenir el propio aliasing y posteriormente, se lleva a cabo la cuantización, que asigna un valor digital equivalente a la muestra tomada. Finalmente, este valor digital se almacena en un formato apropiado para su procesamiento por el sistema electrónico. Este procedimiento de asignación del dato digital equivalente a la muestra analógica obtenida se implementa de maneras muy diversas, dando lugar a la aparición de diferentes tipos de ADCs, con diferentes características y especificaciones entre las que



destacan la tasa de muestreo o velocidad de la conversión y la resolución o número de dígitos binarios (bits) usados para representar el valor analógico de la señal convertida. Estas características dependerán, esencialmente, de la aplicación final para la que se emplee el sistema electrónico con ADC, en nuestro caso de estudio, centrado en la transmisión de señales digitales en sistemas de comunicaciones.

A la hora de fabricar un ADC, es necesario, por tanto, caracterizar su rendimiento, lo cual se refiere a las características y habilidades que tiene el dispositivo en la conversión de señales analógicas a valores digitales. Hay varios factores a tener en cuenta que afectan al rendimiento de dicho convertidor:

1. **Resolución:** La resolución de un ADC se refiere a la cantidad de bits en la salida digital, es decir, a la precisión del convertidor. A mayor resolución se tendrá una mayor precisión en la representación digital de la señal analógica.
2. **Exactitud:** La exactitud se refiere a cómo de cerca está la salida digital de la señal analógica real. Influyen factores como la naturaleza de la señal de entrada, la precisión de los componentes utilizados en el convertidor o la temperatura ambiente.
3. **Linealidad:** La linealidad se refiere a la capacidad de ADC para proporcionar una representación lineal de la señal analógica. Un convertidor no lineal proporcionará una representación distorsionada de la señal analógica.
4. **Velocidad de conversión:** La velocidad de conversión se refiere a la cantidad de muestras que el convertidor puede procesar por segundo. Esta característica es importante para aplicaciones que requieren una tasa de muestreo alta.
5. **Ruido:** El ruido hace referencia a una interferencia no deseada con la señal de entrada, que se suma a ella y afecta en la precisión de la conversión.

En definitiva, es esencial considerar estos factores distintivos del ADC al elegir el dispositivo apropiado para una aplicación específica. Esta caracterización puede ser llevada a cabo de forma manual, lo que requeriría la intervención de un técnico cada vez que se deba realizar el proceso en un ADC, o de manera automática, lo que permitiría sistematizar tanto el procedimiento de caracterización como el proceso en sí sin depender de técnicos, excepto quizás para la colocación y preparación inicial del ADC a caracterizar.

El objetivo central de este TFG se enfoca en el diseño de un sistema de medidas automatizado específicamente orientado a la caracterización del funcionamiento de ADCs (convertidores analógico-digital) de tipo Sigma-Delta. Es fundamental destacar que el sistema propuesto está adaptado para este tipo particular de ADCs y podría requerir ajustes en la instrumentación y el código para aplicaciones con otros tipos de convertidores.

Para ello, desarrollaremos un sistema automatizado que permita generar los estímulos requeridos por el convertidor Sigma-Delta, a la vez que monitorizar su salida.

Utilizando los datos recopilados, la aplicación calculará una serie de parámetros que describirán su funcionamiento. La automatización de las mediciones del ADC requerirá la utilización de equipos y software especializados, posibilitando mediciones automáticas y repetibles. Esta automatización mejorará la fiabilidad y consistencia de las mediciones, que podrán llevarse a cabo en tiempo real y en condiciones de prueba controladas.

Los pasos a seguir para la caracterización del ADC serán:

1. En primer lugar, se seleccionará el equipo de medida adecuado para la aplicación, lo que en nuestro caso incluye generadores y medidores de voltaje, principalmente.
2. Posteriormente se determinarán los sensores que conectaremos al ADC con idea de permitir el funcionamiento del DUT en un rango amplio de operación para que genere una señal analógica representativa.
3. Será necesario determinar la forma en la que se realizará la recogida y almacenamiento de los datos convertidos, lo que deberá realizarse en un dispositivo específico como puede ser un ordenador personal.
4. Los datos almacenados deberán analizarse mediante un software especializado que nos permita obtener resultados y conclusiones, así como la visualización de ambos en formato gráfico o numérico, para su

posterior análisis y presentación.

Una vez diseñada la plataforma automática de caracterización, se probará empleando un ADC de tipo Sigma-Delta para realizar su validación y evaluar su desempeño; lo cual incluye la medición de parámetros críticos como la resolución, la tasa de muestreo, la precisión o la dinámica entre otros.





## 2 ARQUITECTURA DE UN MODULADOR SIGMA-DELTA Y SUS PRINCIPALES CARACTERÍSTICAS

Desde la introducción de los sistemas electrónicos digitales como tecnología que podríamos denominar estándar en la automatización de todo tipo de procesos industriales, el diseño y desarrollo de dispositivos ADC ha constituido una parte importante en el progreso de la ciencia y tecnologías modernas. Se han creado numerosos métodos capaces de convertir una magnitud eléctrica analógica en una forma digital equivalente, que a su vez representa una magnitud física observada.

Inicialmente, este tipo de dispositivos generaban la salida digital equivalente a la entrada analógica sin realimentación que ofreciese posibilidad alguna de corrección de la salida generada, funcionando en un modo habitualmente conocido como ADC de bucle abierto. Un ejemplo de este tipo de dispositivos es el denominado convertidor de tipo “flash” o paralelo, que se muestra en la figura inferior (esquema del dispositivo a la izquierda y curva característica a la derecha) para una resolución de 3 bits o  $n=3$ , y que precisa de  $2^n-1$  comparadores independientes para generar otras tantas entradas digitales a un codificador binario que genera en  $n$  bits el dato digital equivalente a la entrada analógica. La precisión del dispositivo está influenciada por las tolerancias y discrepancias entre componentes (conocido como "mismatch"), que incluyen elementos como las resistencias, así como por las condiciones ambientales en las que opera. Así, si todas las resistencias empleadas en la red de comparación de entrada a los comparadores son iguales, la salida genera un error intrínseco que varía entre 0 y el valor analógico correspondiente a la conversión del bit menos significativo (LSB por sus siglas en inglés). La conversión generaría la escalera que aparece en negro en la figura inferior derecha. Una modificación que sustituyese las resistencias superior e inferior de la red de comparación por  $R/2$ , genera la escalera de conversión indicada en rojo, que ofrece un error intrínseco de la conversión que varía entre  $-LSB/2$  y  $LSB/2$ .

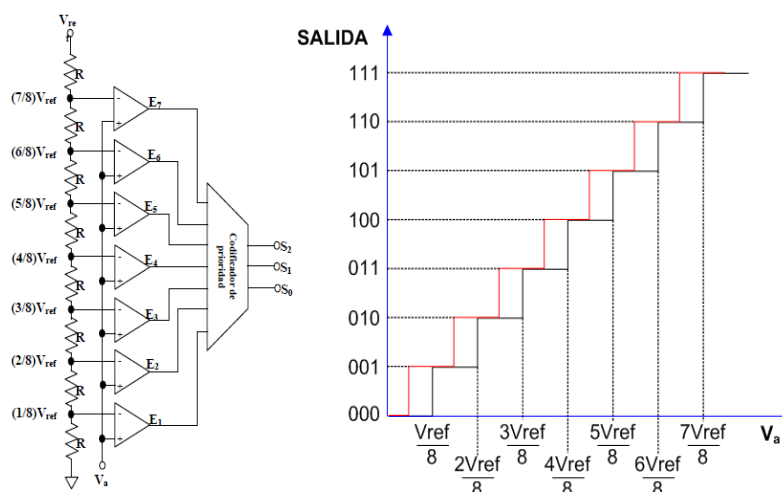


Figura 2-1. ADC de tipo flash y resolución de 3 bits.

La principal ventaja que ofrecen estos ADC proviene de su velocidad de conversión, a costa de una complejidad notable de diseño, derivada de la necesidad de integrar multitud de dispositivos electrónicos de comparación (amplificadores operacionales) y resistencias, así como de su ecualización para que el funcionamiento sea el

descrito en la curva característica, sin que aparezcan no linealidades derivadas de la fabricación.

La mitigación de estos inconvenientes da paso a otras tecnologías de ADC, como los ADC de rampa simple y doble, en los que se minimiza, e incluso anula, el efecto que tienen en el resultado de la conversión los componentes analógicos empleados, mitigando sus efectos a costa de aumentar de manera notable el tiempo que tarda el dispositivo en generar la salida digital (disminución importante de la velocidad de la conversión).

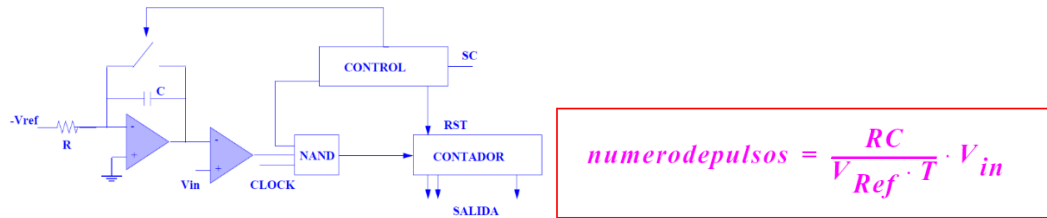


Figura 2-2. ADC de tipo rampa simple.

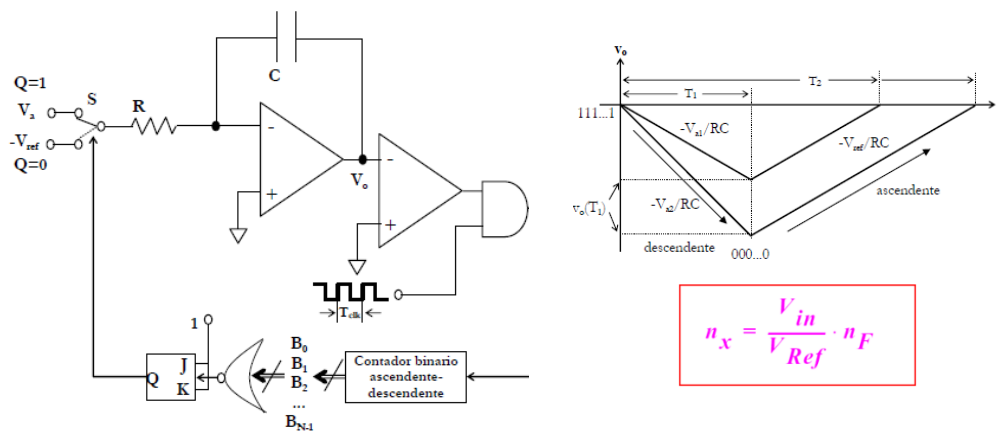


Figura 2-3. ADC de tipo rampa doble.

La necesidad de integrar este tipo de componentes electrónicos en dispositivos microprocesadores para desarrollar soluciones electrónicas integradas con capacidad de interactuar con sensores y actuadores de sistemas industriales, ha favorecido la aparición de otras tecnologías de conversión más habituales hoy en día. Una de ellas es el denominado ADC de aproximaciones sucesivas. En este caso, la salida digital se obtiene de forma algorítmica, evaluando mediante sucesivas aproximaciones la salida digital que mejor representa el valor analógico que se desea convertir. Por este motivo, a este tipo de dispositivos ADC, que son realimentados, se les denomina de esta forma. La figura inferior muestra el esquema genérico de un ADC de aproximaciones sucesivas de 3 bits, indicando el algoritmo seguido para la obtención del valor digital resultado de la conversión. Es interesante destacar que el tiempo de conversión es constante, dependiendo únicamente del número n de bits del convertidor (resolución del ADC).

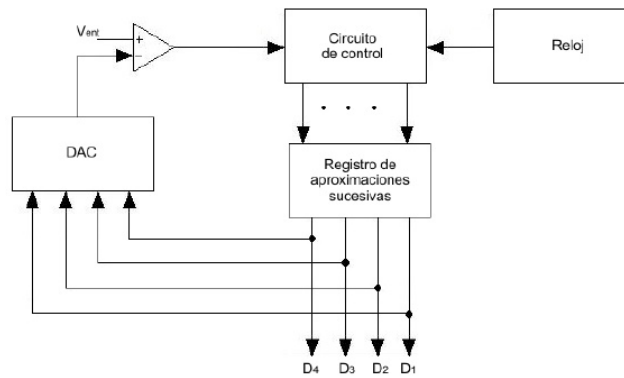


Figura 2-4. ADC de aproximaciones sucesivas de 4 bits.

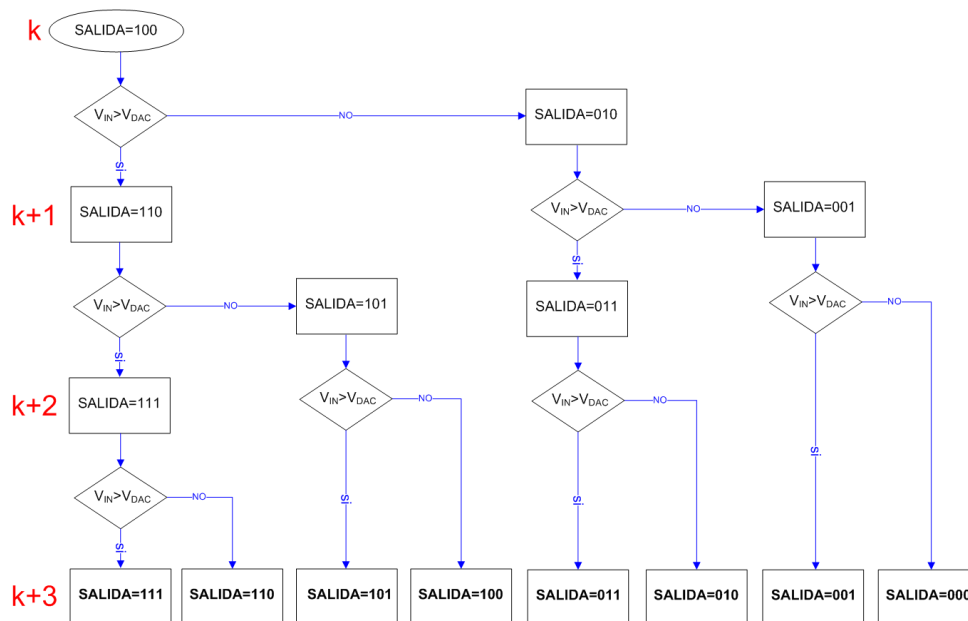


Figura 2-5. Ejemplo de funcionamiento del CAD de aproximaciones sucesivas.

Otro de los ADC empleados habitualmente en microprocesadores es el denominado pipeline, que realiza el proceso de conversión por etapas (stages en la figura). Utilizan una arquitectura de conversión rápidas, de bucle abierto y de pocos bits que se ha visto anteriormente (ADC de tipo flash, de 3 bits en la figura), para determinar la salida digital en varios pasos. En un primer paso, la conversión de un número reducido de bits determina los bits más significativos del dato, valores digitales que se transforman nuevamente en un valor analógico mediante un DAC. Este valor analógico es restado de la entrada original, pasando el resto a una segunda etapa de conversión. El proceso se convierte en iterativo, conformando el número de etapas del convertidor y dando lugar al tiempo de conversión.

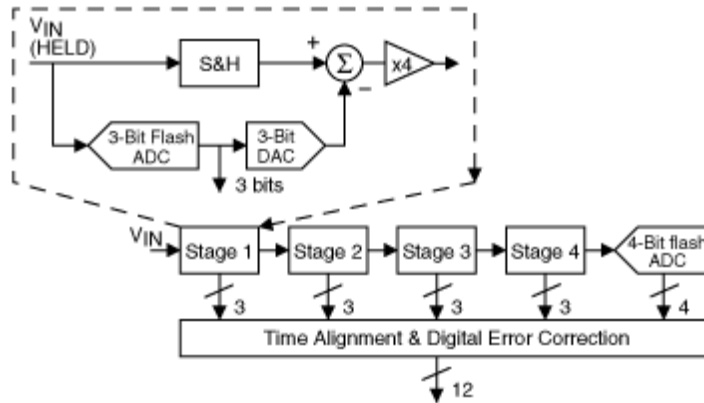


Figura 2-6. ADC pipeline de 12 bits y 4 etapas.

En todo proceso de conversión, la señal analógica de entrada es muestreada y convertida en un valor discreto, denominándose cuantización o cuantificación al proceso que asigna un valor digital a cada muestra analógica tomada. El ADC utiliza código binario en la cuantización, para representar los diferentes estados del cuantificador, código que será procesado en un sistema digital:

$$b_{n_1}b_{n_2} \dots b_1b_0 = b_{n_1} \cdot 2^{n-1} + b_{n_2} \cdot 2^{n-2} + \dots + b_1 \cdot 2^1 + b_0 \cdot 2^0, \text{ con } b \in [0,1]$$

En la ecuación superior se puede observar un ejemplo de paso a código binario de n bits, donde el bit del extremo izquierdo se considera bit más significativo, de mayor peso o MSB (Most Significant Bit) y al bit del extremo derecho se define como bit menos significativo, de menor peso o LSB (Least Significant Bit). Por otra parte, respecto a los códigos binarios bipolares, entre los más comunes se tiene el binario con signo, el binario en complemento a uno, el binario en complemento a dos y el binario desplazado o binario con offset.

Es interesante en este punto destacar que la discretización puede significar una considerable e importante pérdida de información debido a que la señal analógica es continua en el tiempo y puede tener infinitas variaciones en cada momento de tiempo, mientras que la señal digital solo puede tomar un número de valores limitados. El error de cuantización se produce debido al proceso de discretización, es decir, cuando la señal analógica continua es muestreada y convertida en una señal discreta. El error de cuantización se debe a que la señal analógica puede tener un valor entre dos niveles discretos posibles, lo que da lugar a una inexactitud en la representación digital.

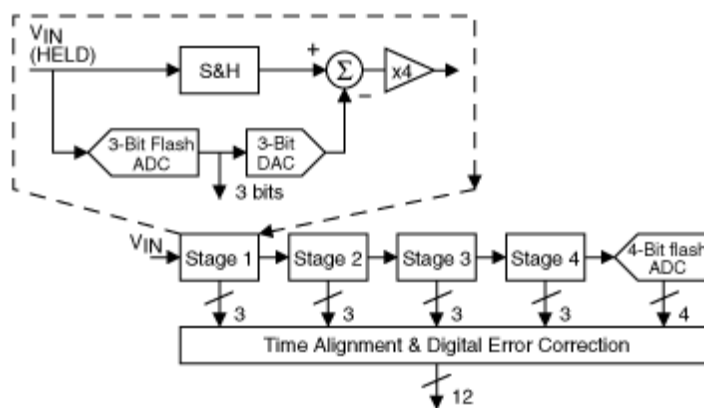


Figura 2-7. Error de cuantización en un ADC.



El error de cuantificación, EQ, se puede calcular empleando

$$EQ = \frac{V_{fs}}{2^N}$$

donde  $V_{fs}$  es el rango completo de tensión analógica a la entrada del ADC y  $N$  es el número de bits del convertidor. A medida que el número de bits ( $N$ ) va aumentando, el error de cuantización disminuye y se mejora por tanto la calidad de la señal digital. Un ejemplo representativo del error de cuantificación se aprecia en la Figura 2-8.

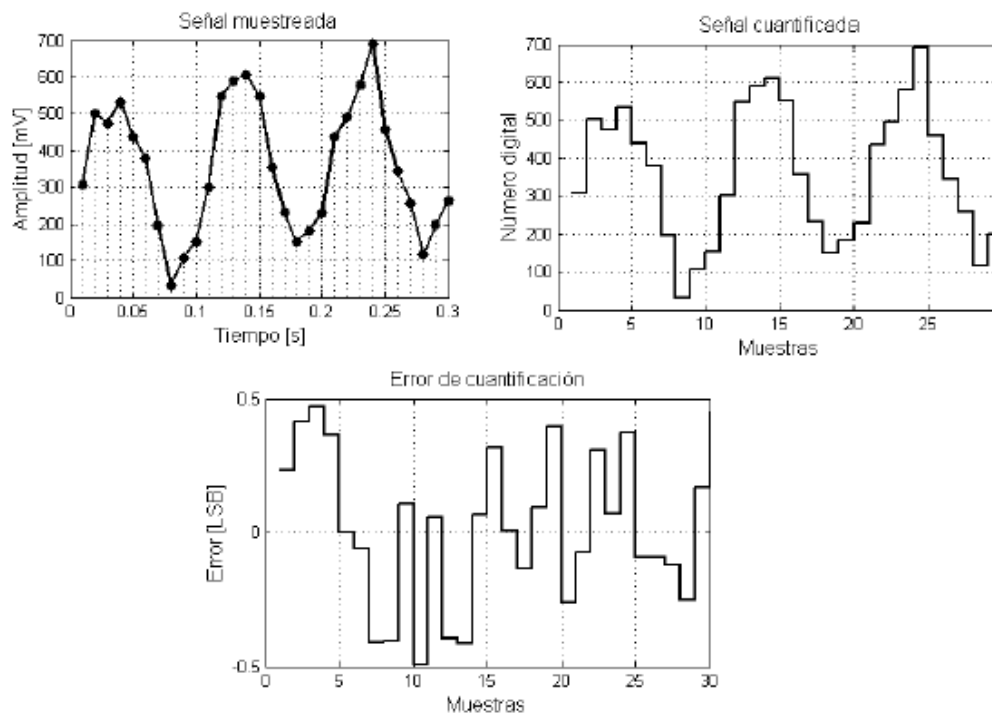


Figura 2-8. Ejemplo de error de cuantización (EQ) en una conversión.

Como posibles soluciones para mitigar en la medida de lo posible el error de cuantización están aumentar el número de bits, lo que disminuye el tamaño de cada intervalo de cuantización, o mejorar el filtrado y la precisión de muestreo en el diseño del convertidor. Además, otra posible solución es usar técnicas de “dithering” que consisten en añadir, de manera controlada, pequeñas cantidades de ruido aleatorio al proceso de cuantización para así poder reducir la correlación entre las diferentes muestras, reduciendo por tanto el error de cuantización. Otra técnica comúnmente utilizada es la del sobremuestreo, la cual es el principio en los convertidores Sigma-Delta que veremos en el siguiente apartado, que permite que se realice una interpolación de los valores de la señal entre los niveles discretos, reduciendo por tanto el error de cuantización.

## 2.1. El convertidor Sigma-Delta

Los convertidores anteriormente descritos tienen como característica común el muestreo de la señal analógica a una tasa superior al doble de la frecuencia fundamental de la señal analógica observada (tasa de Nyquist), para generar un dato digital equivalente en la resolución del dispositivo electrónico ( $n$  bits). En la figura inferior se comparan diferentes tecnologías de ADC, en función de sus prestaciones más importantes (resolución y velocidad de conversión).

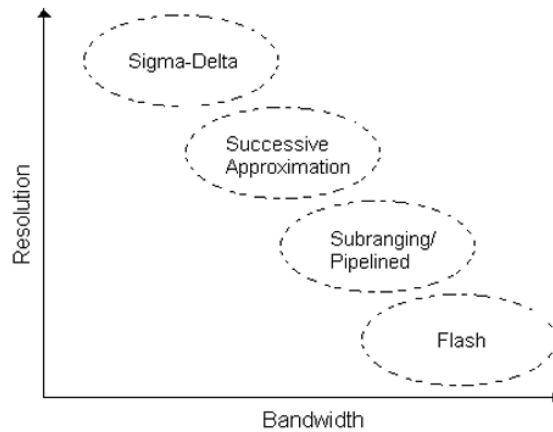


Figura 2-9. Comparativa entre diferentes tecnologías de ADCs.

Una de las tecnologías más avanzadas en el ámbito de los convertidores analógico-digitales, que aparece en la comparativa de ADC como el dispositivo que ofrece una mayor resolución, es el llamado convertidor Sigma-Delta o de sobremuestreo, que son aquellos en los que se utiliza una frecuencia de muestreo de la señal bastante mayor que su frecuencia de Nyquist. En el contexto de los convertidores Sigma-Delta, la codificación es un paso crucial en el proceso de conversión analógico a digital. La señal de entrada se convierte en una señal digital de alta resolución a través del proceso de sobremuestreo y modulación, por lo que mediante la codificación conseguimos que esa señal digital de alta resolución pase a ser una señal digital de baja resolución que pueda ser procesada y transmitida con mayor facilidad. La codificación en los convertidores Sigma-Delta se realiza mediante la técnica de codificación de retroalimentación, en la que la salida digital de la etapa de cuantización se realimenta a la entrada para poder controlar la ganancia del amplificador de entrada.

Este ADC Sigma Delta consta de dos grandes bloques funcionales, que se muestran en la siguiente figura.

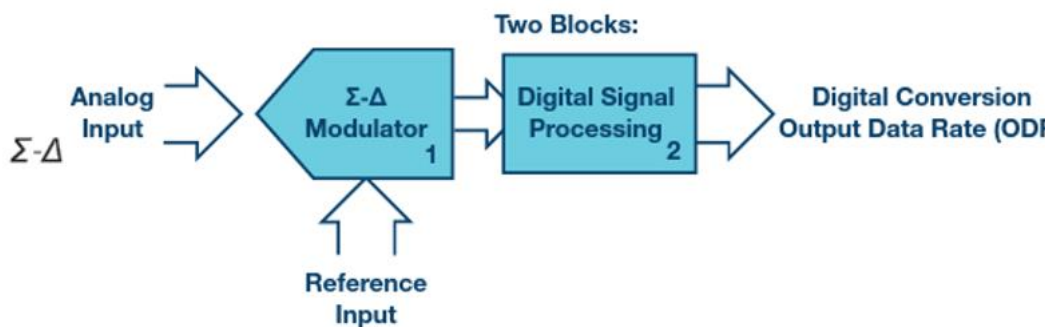


Figura 2-10. Bloques funcionales de un ADC de tipo Sigma-Delta ( $\Sigma$ - $\Delta$ ).

Por un lado, un modulador Sigma-Delta se encarga del muestreo y la cuantización de la entrada analógica a la frecuencia de muestreo, habitualmente en un solo bit. Posteriormente, un segundo bloque puramente digital, y denominado decimador, realiza un procesamiento de señal para determinar el dato digital equivalente a la entrada analógica. Este segundo bloque suele implementar un filtro digital que elimina todas las componentes fuera de la banda de la señal, incluido gran parte del error de cuantización que se genera en el primer bloque. La principal ventaja de este dispositivo de conversión es que relaja los requerimientos de la circuitería analógica del dispositivo (particularmente en lo referente a la definición de filtros antialiasing) a expensas de hacer uso de una circuitería digital más compleja, aunque más robusta e inmune a las imperfecciones de la electrónica.

En el bloque de modulación de estos ADC, la señal analógica de entrada  $V_{in}$  que se desea convertir, se compara con una señal retroalimentada que proviene de un sumador y del modulador Delta (que ha generado una señal digital de 1 bit, llamado bit de error, como diferencia entre la entrada analógica y la señal retroalimentada). La salida de este comparador se conecta a la entrada de un integrador, cuya salida se cuantifica en un bit que actúa como clasificador, generando un 1 o un 0 dependiendo de si la salida del integrador es positiva o negativa. La

cadena de valores digitales de salida ofrece una tasa de variación de bits equivalente a la magnitud  $V_{in}$ . El esquemático de un modulador Sigma-Delta de 1 bit es el siguiente:

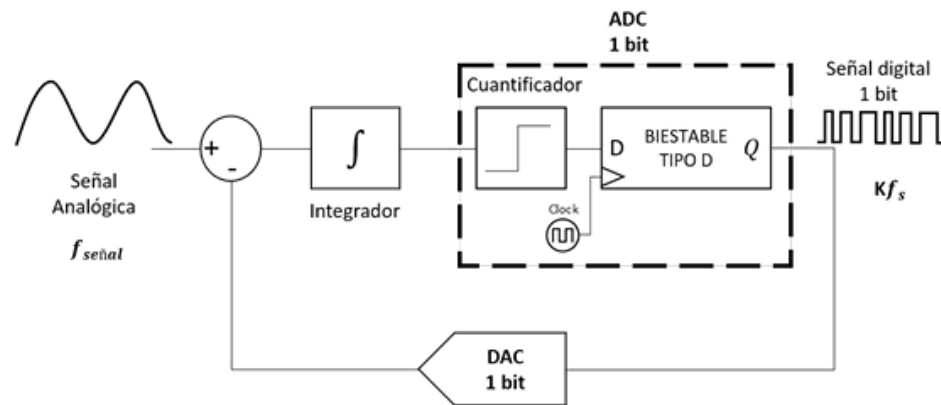


Figura 2-11. Modulador Sigma-Delta de un bit.

A continuación, se aplica el sobremuestreo. El bit de error se filtra a través de un filtro digital paso bajo para eliminar la alta frecuencia del ruido de cuantificación y se sobremuestra a una frecuencia mucho mayor a la frecuencia de muestreo de la entrada analógica, para así poder reducir el ruido y aumentar la resolución. Finalmente, los bits de error sobremuestreados se promedian en un contador acumulativo y el resultado se convierte en un número binario que representa la salida digital del ADC.

El ADC Sigma Delta es una arquitectura que intercambia la resolución en amplitud por tiempo, evitando así el desarrollo e implementación de complejos circuitos analógicos de precisión, por lo que representan un enfoque atractivo para implementar circuitos de conversión de precisión en tecnologías VLSI digitales escaladas. De hecho, estos convertidores se utilizan en una amplia gama de aplicaciones como la medición de sensores de alta precisión, la digitalización de señales de control o en aplicaciones relacionadas con la electrónica de consumo, como pueden ser los reproductores de audio digital, televisores de alta definición y dispositivos móviles debido a su alta resolución y bajo ruido y coste, por lo que se espera que sigan siendo muy importantes en el futuro.

La base de este tipo de ADC es el concepto de sobremuestreo, o proceso en el cual una señal es muestreada a una frecuencia mayor a la requerida, definido por el parámetro OSR (Over Sampling Ratio), que es lo que permite, como ventaja, que la cuantificación del ruido se esparza sobre un mayor rango de frecuencia, reduciendo la densidad espectral de potencia del ruido respecto a la señal de interés.

$$OSR = \frac{f_{osr}}{2 \cdot f_N}$$

donde  $f_{osr}$  es la frecuencia de sobremuestreo y  $f_N$  la frecuencia de Nyquist (frecuencia máxima de la señal de entrada).

En resumen, un ADC de tipo Sigma Delta dispone de los bloques que se muestran en la figura interior:

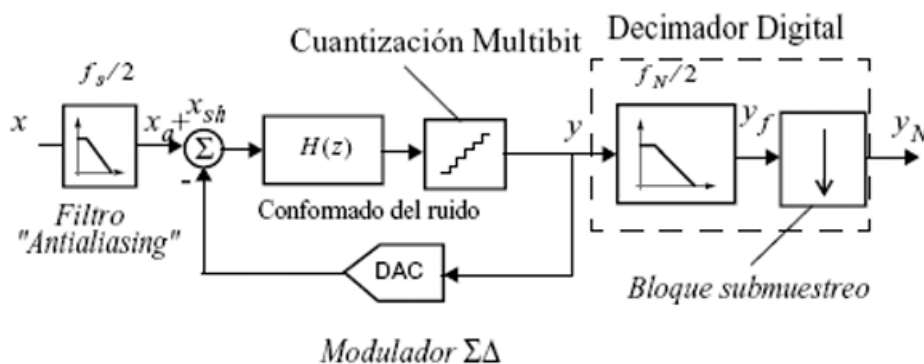


Figura 2-12. ADC de tipo Sigma-Delta.

### 2.1.1 Filtro Anti-aliasing

El aliasing o solapamiento se produce cuando una señal analógica continua no es muestreada a una frecuencia adecuada (frecuencia más baja de lo que debería ser). El efecto que genera el aliasing en la discretización de la señal es equivalente a una distorsión del valor obtenido por superposición de ruido en el espectro de frecuencia de cola de la propia señal (ver figura inferior). La introducción de este filtro anti-aliasing es equivalente a disminuir la frecuencia de la señal para evitar el aliasing mediante la eliminación de las componentes de la señal de entrada que están por encima de la frecuencia de Nyquist, que es la frecuencia  $f_s$  mínima a la que puede ser muestreada la señal sin introducir aliasing. En el caso de los ADC Sigma Delta, y debido al sobremuestreo, las especificaciones sobre el filtro anti-aliasing se pueden relajar, sustituyéndolo por un simple filtro pasivo de primer orden de la forma

$$H(s) = \frac{K}{(1 + sT)}$$

siendo  $H(s)$  la función de transferencia de dicho filtro,  $K$  la ganancia,  $s$  la frecuencia compleja y  $T$  la constante de tiempo.

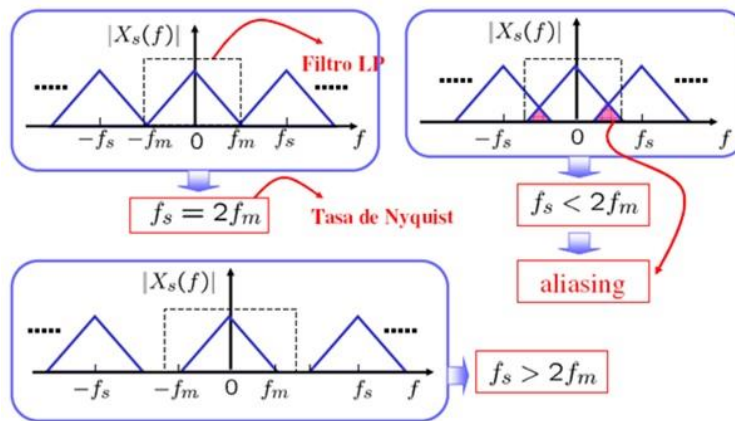


Figura 2-13. Efecto aliasing y cómo se mitiga con filtros analógicos antes del muestreo.

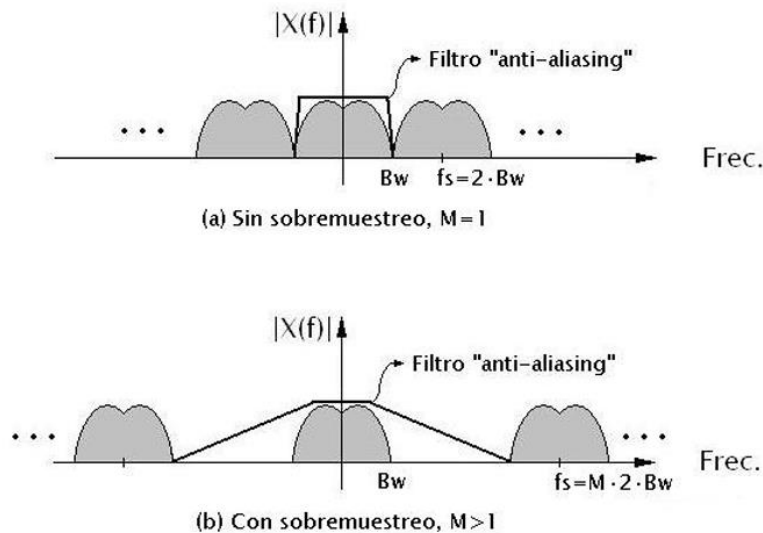


Figura 2-14. Comparativa del efecto aliasing entre ADC sin sobremuestreo (figura superior) y con sobremuestreo (figura inferior), como es el caso de un ADC Sigma Delta

### 2.1.2 Decimador

Bloque digital que se encarga de realizar dos operaciones y que está compuesto por dos bloques. El primero de ellos, un filtro digital, se encarga de eliminar el ruido de cuantización que ha quedado excluido de la banda tras su paso por el modulador sigma-delta. El segundo bloque, o bloque de sobremuestreo, reduce la frecuencia de muestreo a la frecuencia de Nyquist de la señal original.

### 2.1.3 Modulador Sigma-Delta

Dispositivo o circuito que se utiliza para modificar la señal de entrada, denominada señal moduladora, para convertirla en una señal de salida, llamada señal modulada, con características diferentes a la señal de entrada. El modulador consta de otros dos bloques. El primero de ellos es un filtro. Su salida se conecta a un segundo bloque que es un convertidor analógico-digital, que muestrea y cuantiza la señal que le llega del filtro. Normalmente, ese convertidor analógico-digital es de un solo bit y actúa como comparador, introduciendo un error de cuantización.

## 2.2. Caracterización de un ADC

Los ADC se encuentran en el límite de los dominios digital y analógico, y hemos visto anteriormente que existen numerosas arquitecturas que desarrollan este tipo de dispositivos. Se trata de componentes que son ampliamente utilizados en los sistemas electrónicos actuales, disponibles en todo tipo de productos electrónicos de consumo como teléfonos celulares, cámaras y tarjetas de sonido, etc. Los requisitos de la aplicación final asociada al sistema electrónico en el que se encuentra embebido un ADC fijan el tipo de ADC necesario para garantizar el rendimiento y la funcionalidad óptimos del propio sistema. Es necesario, por tanto, conocer en profundidad las características de los ADCs, y en particular de los ADC Sigma-Delta que son el núcleo central de este trabajo, que serán las que nos permitan conocer el tipo de aplicaciones compatibles con el dispositivo y su utilidad. Las características deben analizarse y evaluarse antes de incorporarlos en el sistema electrónico, y algunas de las más básicas las hemos comentado anteriormente: resolución, velocidad, rango de conversión y rango dinámico.

La **resolución** en un ADC puede definirse como el cambio más pequeño en el voltaje analógico de entrada que puede detectarse a la salida y viene determinado por el número de bits que utiliza el ADC para digitalizar la entrada. Un ADC de  $n$ -bits es capaz de generar  $2^n$  valores diferentes a la salida (por ejemplo, un ADC de 12 bits generaría  $2^{12}$  o 4096 salidas diferentes). Por ello, la resolución suele expresarse como  $n$  o número de bits

asociados al ADC.

La **velocidad de conversión**, o **frecuencia de conversión** ( $F_s$ ), hace referencia a las muestras por segundo que somos capaces de obtener a partir del ADC, de forma que mide qué tan rápido puede convertir con precisión el dispositivo una señal analógica. Esta característica es la inversa del denominado **tiempo de conversión** ( $T_c$ ) que, como indica su nombre, es el tiempo que tarda un convertidor en realizar una conversión, es decir, el tiempo transcurrido desde que se da la orden de conversión hasta que se acaba y se expresa en ciclos de reloj del oscilador base que controla el ADC o en segundos, dependiendo del ámbito del que estemos hablando. Esta característica puede ser constante o variable dependiendo del tipo de ADC. La velocidad de conversión es la inversa del tiempo de conversión, y determina la frecuencia máxima a la cual el convertidor puede ofrecer datos a la salida de manera estable, controlada y precisa. Se expresa en Hz o en muestras por segundo.

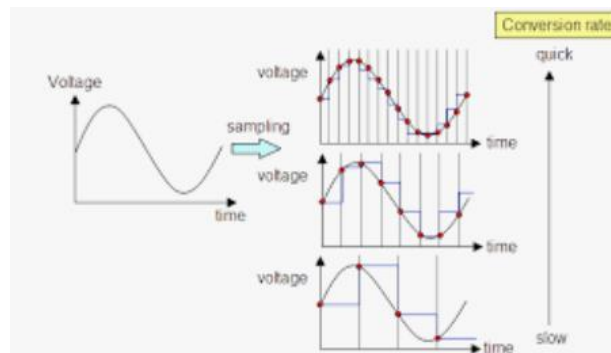


Figura 2-15. Ejemplos para diferentes tiempos de conversión

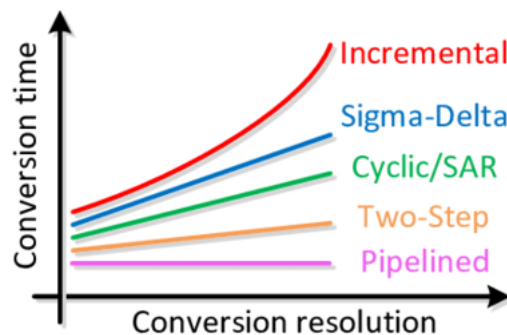


Figura 2-16. Valores típicos de la velocidad de conversión (inversa de tiempo de conversión) y resolución en diferentes tipos de ADCs: comparativa

Mientras que el **rango de conversión** define el rango de valores analógicos a la entrada del ADC que éste es capaz de digitalizar, el bit menos significativo (LSB) del dato obtenido a la salida de un ADC identifica el intervalo de entrada más pequeño que se puede detectar. Para convertir LSB en tensión, se divide el rango de conversión del dispositivo por dos elevado a su resolución. La Tabla 2-1 muestra el LSB en un ADC con un rango de entrada de un voltio y diferentes resoluciones entre 8 y 16 bits.

Tabla 2-1. LSB en ADC con diferentes resoluciones y rango de conversión de 1V

Resolución	LSB
8 bits	3.906 mV
10 bits	0.977 mV
12 bits	0.244 mV
14 bits	61 $\mu$ V
16 bits	15 $\mu$ V

El rango dinámico (DR) hace referencia a la relación entre el valor analógico máximo y mínimo que se puede convertir a la entrada del ADC. Se define, por tanto, como la relación entre los valores más grandes y pequeños que el ADC mide de manera confiable. Se expresa habitualmente en dB de la siguiente forma:

$$DR (dB) = 20 \log_{10} \left( \frac{(2^N - 1)LSB}{LSB} \right) \approx 6.02 \cdot N$$

Donde N indica la resolución en bits del ADC. Un ADC de 10 bits tiene, por tanto, un DR de 60.2 dB, lo que significa que el ADC podría resolver amplitudes de señal desde x hasta aproximadamente 1000x, donde x es el mínimo que se puede detectar.

Ciertas características de los ADC determinan su comportamiento y rendimiento dinámicos. Entre éstas se incluyen los términos relación señal a ruido (SNR), distorsión armónica total (THD), relación señal a ruido y distorsión (SINAD) y rango dinámico libre de espúreos (SNDR), obtenidos a unas determinadas frecuencias de la señal de entrada y tasa de muestreo o de datos de salida, que se definen de la siguiente manera:

#### 2.1.4 SNR

Es la relación entre el valor rms de la señal de entrada y el valor rms de todos los demás espectros de componentes por debajo de la frecuencia de Nyquist, excluyendo armónicos y corriente continua. Luego, se multiplica el logaritmo en base 10 de esa relación por 20 para obtener SNR en decibelios. El valor SNR ideal de un ADC que no sea sobremuestreado es igual a  $6.02N + 1.76$  dB, siendo N el número de bits del ADC. Este valor se obtiene de suponer, en primer lugar, que la señal de entrada es sinusoidal, de manera que su valor rms se obtiene de dividir por  $\sqrt{2}$  el rango de escala completa del convertidor:  $2(N-1)LSB$ . Por otro lado, se asume que el ruido que se crea en el ADC ideal proviene sólo de la cuantización, es triangular y varía entre  $\pm LSB/2$ , lo que genera una componente rms de ruido igual a  $\pm(LSB/2)/\sqrt{3} = LSB/\sqrt{12}$ :

$$SNR (dB) = 20 \log_{10} \left( \frac{rms \text{ señal entrada}}{rms \text{ ruido cuantización}} \right)$$

$$SNR(dB)_{No \text{ Sobremuestreado}} = 20 \log_{10} \left( \frac{2^{N-1} \frac{LSB}{\sqrt{2}}}{\frac{LSB}{\sqrt{12}}} \right) = 6.02N + 1.76$$

Sin embargo, los convertidores Sigma-Delta distribuyen la potencia del ruido en un rango de frecuencia más amplio debido al sobremuestreo, además de tener un modulador que empuja el ruido de baja frecuencia a frecuencias más altas. La inclusión, además, de un filtro digital que promedia el ruido y lo elimina en las frecuencias más altas, hace que el valor de SNR cambie respecto de otros convertidores. Habitualmente se asume que SNR puede obtenerse en un ADC Sigma-Delta como  $6.02(N + NINC) + 1.76$ , donde N es el número de bits del modulador y NINC es el aumento de resolución:

$$N_{INC} = \frac{1}{6.02} \left[ (20M + 10) \log_{10} OSR - 20 \log_{10} \left( \frac{\pi^M}{\sqrt{2M + 1}} \right) \right]$$

En esta última fórmula, M es el orden del modulador y OSR es la relación de sobremuestreo durante la conversión, como ya se definió anteriormente. De esta manera, el valor SNR del ADC Sigma-Delta con un modulador de primer orden (M=1) sería:  $6.02N + 1.76 - 5.17 + 30 \log_{10} OSR$ .

Esta fórmula asume que la linealidad, el ruido y los errores de compensación de los convertidores en el dispositivo (por lo general, de 1 bit) son ideales y que el filtro digital también tiene una respuesta ideal, eliminando toda frecuencia por encima de la frecuencia de corte del filtro. Los ADC Sigma-Delta reales no se comportarán de forma tan ideal, por lo que será necesario caracterizar el ADC experimentalmente para determinar su verdadero rendimiento. En realidad, esta caracterización nos permitirá obtener una visión realista de las capacidades de cualquier ADC y la forma de determinar SNR en cualquier ADC es mediante la siguiente fórmula:

$$SNR = 20 \log_{10} \left( \frac{V_{RMS-FS}}{V_{RMS-NOISE}} \right)$$

### 2.1.5 THD

es la relación entre la distorsión que genera el ADC (suma de los valores rms de todas las componentes de los diferentes armónicos, obtenidos de la aplicación de un análisis FFT que excluya la componente de DC), y el valor rms de una señal de entrada senoidal pura a escala completa. Se suelen coger sólo los primeros armónicos que es donde se representa la mayoría de la distorsión. Se expresa en dB.

$$THD (dB) = 20 \log_{10} \left( \sqrt{\frac{V_{f2}^2 + V_{f3}^2 + \dots + V_{fi}^2}{V_{f1}^2}} \right)$$

### 2.1.6 SINAD

es la relación del valor rms de la señal de entrada real con todos los otros componentes espectrales por debajo de la frecuencia de Nyquist, incluyendo armónicos, pero excluyendo la componente de continua (suma de ruido y distorsión). La SINAD se expresa en dB, y se puede calcular aplicando una onda senoidal pura (de frecuencia  $f_1$ ) a la entrada del ADC, con valores extremos cercanos al límite del ADC. Es una medida útil del rendimiento del ADC porque incluye el efecto de todo el ruido, la distorsión y los armónicos introducidos por el propio ADC. Asociado a este parámetro se asocian el denominado nivel de sobrecarga, que se produce cuando la potencia de la señal de entrada sinusoidal produce una caída de 3 dB en el valor SINAD respecto de su valor máximo, y el fondo de ruido, el cual se define como el nivel de ruido presente en la señal en ausencia de señal de interés (es decir, se trata de un ruido ambiental y produce un SINAD de 0 dB).

$$\begin{aligned} SINAD(dB) &= 20 \log_{10} \left( \sqrt{\frac{V_{f1}^2}{V_{f2}^2 + V_{f3}^2 + \dots + V_{fi}^2 + V_{rms \text{ ruido cuantización}}^2}} \right) \\ &= -10 \cdot \log_{10}(10^{-SNR/10} + 10^{THD/10}) \end{aligned}$$

### 2.1.7 SFDR

proporciona información sobre la diferencia entre el tono de amplitud máxima en el espectro de frecuencia, excluyendo la componente fundamental, y el tono de entrada fundamental. El parámetro suele definirse en dB, y se obtiene de restar la energía de la componente fundamental de entrada y la del máximo de las energías de las diferentes componentes de frecuencia, exceptuando la componente fundamental.

### 2.1.8 ENOB

Otro parámetro que se pueden obtener a partir de los anteriores es el número efectivo de bits (ENOB), cuya fórmula representa la relación señal a ruido más distorsión en términos de bits:

$$ENOB = \frac{SINAD - 1.76}{6.02}$$



### 2.1.9 Relaciones entre las magnitudes anteriores

La mayoría de los fabricantes de ADC han adoptado estas especificaciones para caracterizar los dispositivos, y es posible obtener relaciones matemáticas entre ellos, asumiendo que todos los valores se miden con la misma frecuencia y amplitud de la señal de entrada. Algunos de estos parámetros de comportamiento dinámico del ADC aparecen representados en la siguiente gráfica:

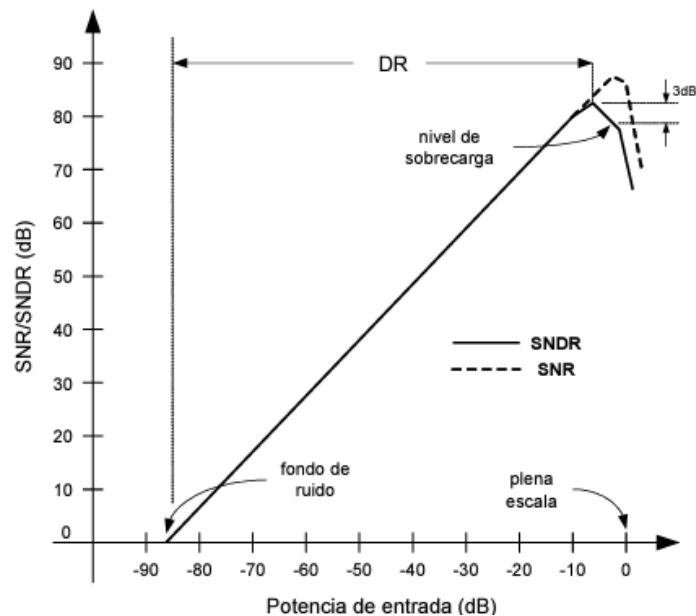


Figura 2-17. Parámetros que determinan el funcionamiento dinámico de un ADCs

Lo que resulta evidente de lo descrito es que todo mecanismo que cuantifique las características dinámicas de un ADC requiere del empleo de un generador de señal, preferiblemente de bajo ruido, que imponga una señal senoidal a la entrada del ADC, así como de la aplicación a la salida del ADC de un algoritmo FFT. La Figura inferior muestra el gráfico FFT de la salida de un ADC de 12 bits con una señal de entrada de 9,9 kHz. Con la prueba real, el ruido en la señal de entrada sinusoidal debería ser más de tres veces menor que el ruido ideal teórico del ADC. Si esta situación es cierta, la señal rms es igual a la magnitud de la señal dividida por  $\sqrt{2}$ . El ruido generado por el ADC se puede obtener utilizando la magnitud medida de los armónicos generados, excluyendo para el cálculo la componente fundamental de la entrada y sus múltiples armónicos, en nuestro caso 19,8 kHz, 29,7 kHz, etc. El ruido rms del ADC se obtiene tomando la magnitud del resto de componentes en frecuencia, elevándolas al cuadrado, sumándolas y calculando luego la raíz cuadrada de esa suma. El punto C de la Figura muestra la SNR calculada (72 dB) de los datos de este convertidor de 12 bits, que está cerca del valor ideal de 74 dB.

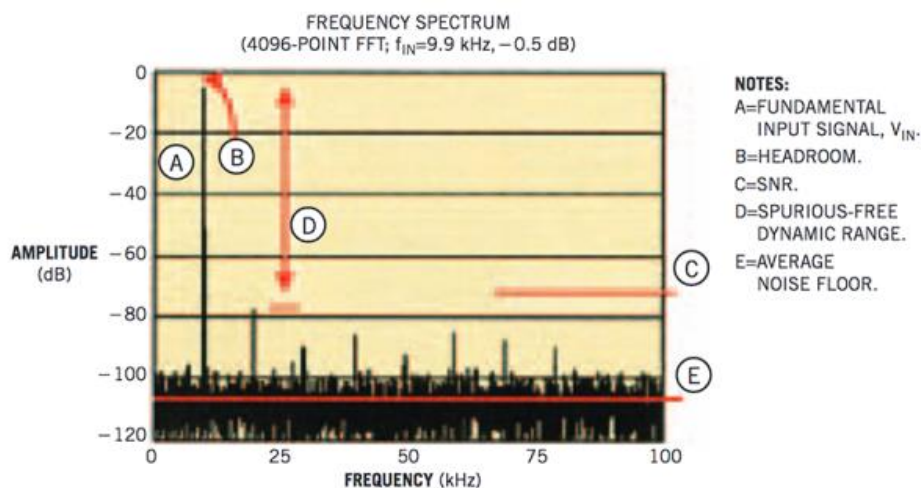


Figura 2-18. FFT aplicada a los datos muestreados de un ADC de 12 bits. En este gráfico FFT, hay 4096 muestras que el convertidor adquiere a una velocidad de datos de 200 kmuestras/seg. La SNR (punto C en este gráfico) es un valor calculado.

La salida de la FFT se puede utilizar como un analizador de espectro analógico para poder medir la amplitud de las distintas componentes armónicas de la señal de entrada, de manera que el análisis FFT deberá utilizar una configuración como la que se muestra en la figura inferior:

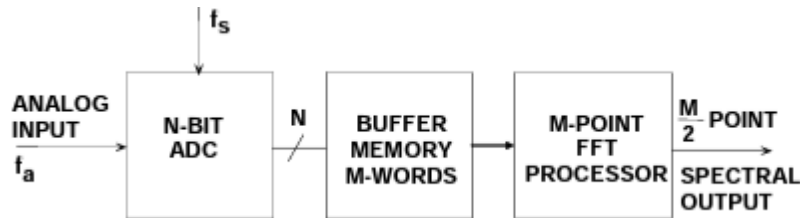


Figura 2-19. Análisis FFT para cuantificación de las características dinámicas de un ADC

En nuestro caso, la FFT es válida al tratarse de un análisis de una señal periódica conectada al ADC, si bien es importante tener en cuenta que la sección de la señal muestreada debe contener un número entero de períodos, evitándose así la obtención de un espectro FFT “manchado” (efecto de fuga).

En el primer paso, se tomará una muestra de una sección de la señal que se va a medir y se almacenará en memoria para su posterior procesamiento. Dos parámetros son relevantes aquí:

- La tasa de muestreo del sistema de medición ( $f_s$ ).
- El número de muestras seleccionado o longitud del bloque M, que en la FFT debe ser siempre una potencia de 2.

Estos parámetros determinan otros parámetros a destacar de la medición:

- El ancho de banda o frecuencia de Nyquist,  $f_n$ , que indica la frecuencia máxima teórica que puede ser determinada por la FFT y que teóricamente es  $f_n = f_s/2$  (en la práctica, será algo menor de esta cantidad).
- Duración de la medición D, que se calcula a partir de la tasa de muestreo  $f_s$  y la longitud del bloque M como  $D = M/f_s$ .
- Resolución de frecuencia  $df$ , que la diferencia de frecuencia entre dos resultados de medición:  $df = f_s/M$ .

En la práctica, la frecuencia de muestreo vendrá establecida por el sistema empleado en el análisis, mientras que la duración de la medición y la resolución de la frecuencia se definen seleccionando la longitud del bloque M de manera que un valor pequeño de M da como resultado rápidas repeticiones de medición con una resolución de frecuencia gruesa y un valor elevado de M resulta en repeticiones de medición más lentas con una resolución de frecuencia fina.

En definitiva, la FFT actúa como un analizador de espectro de banda estrecha con un ancho de banda igual a  $f_s/M$  que barre todo el ancho de banda de Nyquist de la señal: la salida espectral de la FFT será una serie de M/2 puntos en el dominio de la frecuencia, con un espacio entre puntos y con un ancho de cada “contenedor” de frecuencia, también denominado resolución de la FFT,  $df$ , y un rango de frecuencia total cubierto de  $f_s/2$ .

Otras características de los ADC que no se han comentado y que determinan su comportamiento y rendimiento en DC son los errores de desplazamiento (offset o error de escala cero) y ganancia (error de plena escala), la no linealidad diferencial e integral (DNL e INL) y el error total no ajustado (TUE).

El error de desplazamiento se define como la diferencia (desplazamiento) entre la curva ideal y la real, y puede ser un valor positivo o negativo. El error de ganancia es la diferencia entre la tensión ideal que proporciona la salida de escala completa frente a la tensión real para la cual el ADC genera la salida de escala completa.

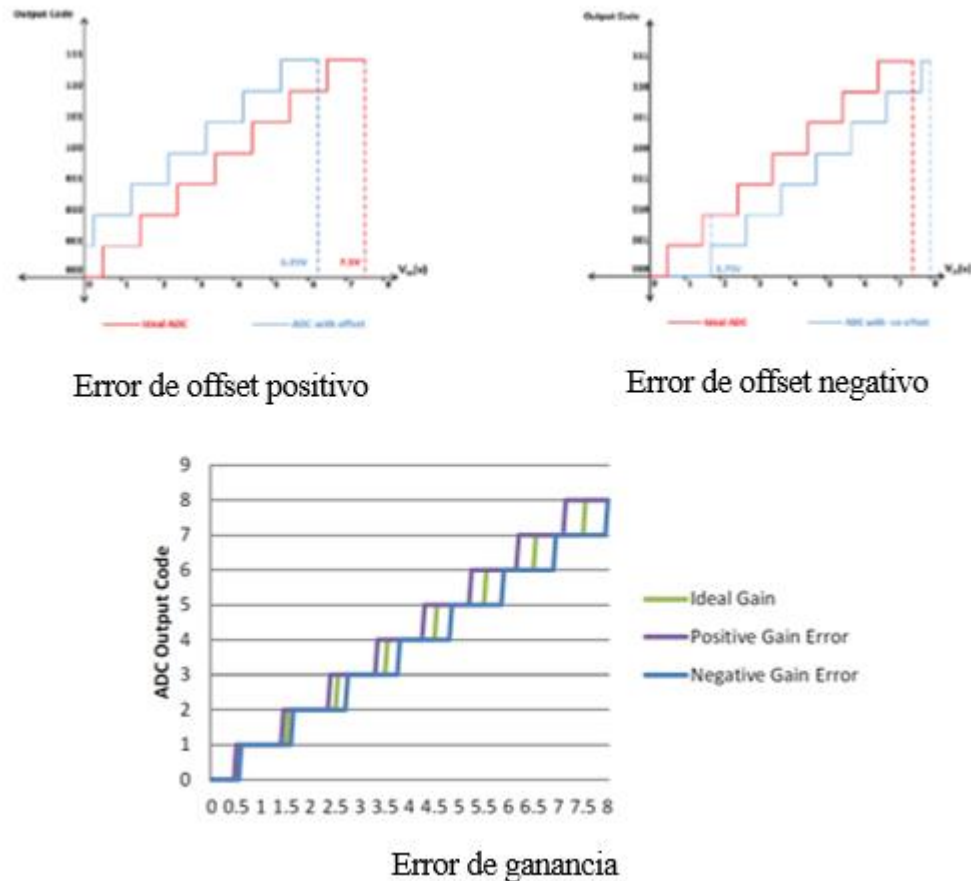


Figura 2-20. Errores de desplazamiento y ganancia en un ADC

El error DNL, del inglés Differential Non-Linearity, mide la variación a la salida del convertidor cuando se aplica un cambio a la entrada del bit menos significativo. Es decir, mide la diferencia máxima o mínima entre dos niveles de salida adyacentes y se expresa en términos de LSB. En un ADC ideal, en el que  $DNL=0\text{LSB}$ , cada paso analógico es igual a  $1\text{LSB}$ , donde  $1\text{LSB}=V_{\text{FSR}}/2^N$ ,  $V_{\text{FSR}}$  es el rango de escala completa y  $N$  es la resolución del ADC. Los valores de transición asociados al ADC se separan exactamente  $1\text{LSB}$ . Una especificación de error DNL menor o igual a  $1\text{LSB}$  garantiza una función de transferencia monótona en el ADC: la salida digital aumenta (o permanece constante) con una señal analógica de entrada creciente, evitando cambios de signo en la pendiente de la curva de transferencia. Si  $V_D$  es el valor físico correspondiente al código de salida digital  $D$ ,  $N$  es la resolución del ADC y  $V_{\text{LSB-ideal}}$  es el salto ideal para dos códigos digitales adyacentes, DNL se expresa como:

$$DNL = \left| \frac{(V_{D+1} - V_D)}{V_{\text{LSB-ideal}}} - 1 \right|, \quad 0 < D < 2^{N-2}$$

Al agregar ruido y componentes espurios más allá de los efectos de la cuantificación, los valores más altos de DNL generalmente limitan el rendimiento del ADC en términos del SNR y DR.

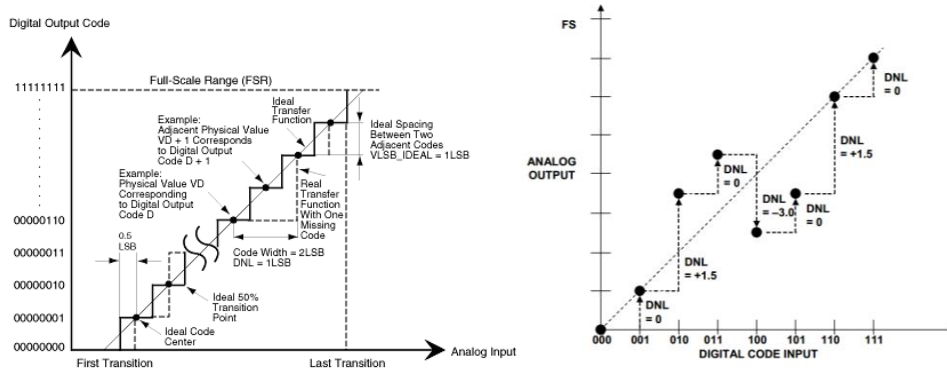


Figura 2-21. Error DNL en un ADC

Por otro lado, el error INL, del inglés Integral Non-Linearity, mide la desviación máxima a la salida del convertidor en comparación con una línea recta que se ajusta a la escala de salida, es decir, describe la desviación de la función de transferencia real del ADC respecto de esa línea recta, la cual representa la función de transferencia ideal del dispositivo. Esta desviación se expresa en términos de LSB o porcentaje del rango de escala completa (FSR). Este parámetro debe mantenerse lo más bajo posible para garantizar la mayor precisión posible en la conversión. La magnitud del error INL depende de la posición elegida para esta línea recta, y habitualmente se definen dos líneas rectas, el mejor INL de línea recta y el INL de punto final.

El mejor INL de línea recta utiliza como línea recta de comparación la aproximación más cercana a la función de transferencia real del ADC. La posición exacta de la línea no está claramente definida pero este enfoque sirve como representación de la linealidad, proporcionando información sobre el error de offset y ganancia.

El INL de punto final define como línea recta de comparación la que pasa por los puntos finales de la función de transferencia del convertidor, definiendo así una línea recta precisa que, en un ADC de N bits, se define por sus salidas cero (los N bits todos a cero) y de escala completa (los N bits todos a uno).

Generalmente se suele escoger para determinar INL el mejor enfoque de línea recta de comparación, es decir, aquella que produzca mejores resultados y el valor de INL se obtiene después de anular los errores de compensación estática y de ganancia, de manera que puede describirse como:

$$INL = \left| \frac{(V_D - V_{ZERO})}{V_{LSB-ideal}} - D \right|, \quad 0 < D < 2^{N-1}$$

Donde  $V_D$  es el valor analógico representado por el código de salida digital  $D$ ,  $N$  es la resolución del ADC,  $V_{ZERO}$  es la entrada analógica mínima correspondiente a un código de salida todo cero y  $V_{LSB-ideal}$  es el espacio ideal para dos códigos de salida adyacentes.

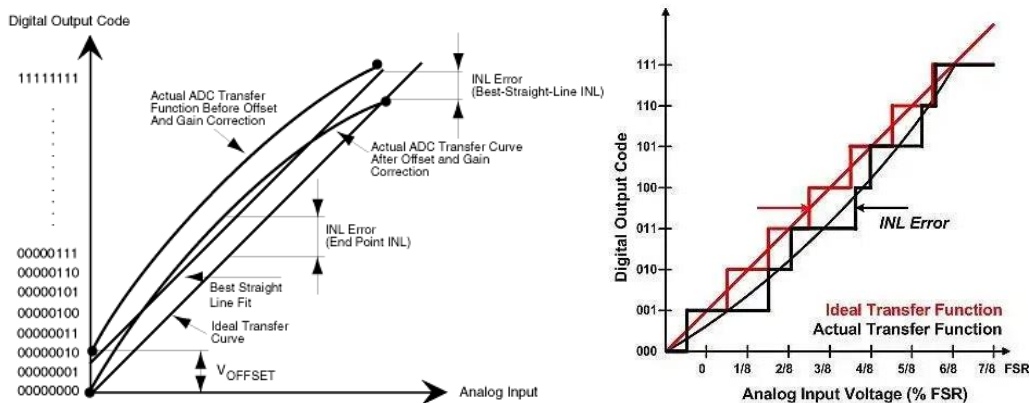


Figura 2-22. Error INL en un ADC

INL y DNL se pueden estimar empleando una forma de onda de tensión a la entrada del ADC de baja frecuencia que se irá cambiando. La prueba requiere:

- Una fuente de señal analógica de precisión, que cree las tensiones de prueba a la entrada del ADC mediante un barrido lento de su rango de entrada (desde cero hasta la escala completa):  $V_{in}$ .
- Un analizador lógico conectado a la salida del ADC para determinar los errores de offset y ganancia.
- Un convertidor digital-analógico (DAC) de alta precisión (mucho más alto que el del ADC). Este dispositivo generará un valor de tensión  $V_{out}$  por cada valor  $V_{in}$  que se fuerce a la entrada del ADC.
- Un dispositivo con función de traza X-Y.

Para cada valor  $V_{in}$  impuesto, se deberá calcular  $VDIFF=V_{out}-V_{in}$ , valor que se puede mostrar en el trazador X-Y y vincular a los errores INL y DNL. Un cambio en el nivel de cuantificación indica no linealidad diferencial y una desviación de  $VDIFF$  de cero indica la presencia de no linealidad integral.

Finalmente, el error total no ajustado o TUE es una indicación del peor error rms del ADC sin aplicar cualquier corrección de compensación o error de ganancia, con lo que se calcula como una suma de los diferentes errores de DC (offset, ganancia, DNL e INL):

$$TUE = \sqrt{(error\ de\ offset)^2 + (error\ de\ ganancia)^2 + (DNL)^2 + (INL)^2}$$

## 2.2 Protocolos de Instrumentación

Según se ha visto en el apartado anterior, la caracterización del comportamiento real de cualquier ADC requiere manejar cierta instrumentación electrónica y de la implementación de un algoritmo FFT.

Inicialmente, el término “Instrumentación Electrónica” se relaciona con dispositivos basados en componentes electrónicos que permiten la medición de magnitudes eléctricas. Hoy en día, el término instrumentación electrónica se puede decir que ha extendido su definición, existiendo una amplísima gama de instrumentos en el mercado que, de forma genérica, se pueden clasificar en:

1. Instrumentos de medida, entre los que se incluyen indicadores locales, transmisores e interruptores, cuya función consiste en dar una señal visual, neumática, electrónica o de otro tipo proporcional a la variable medida.
2. Instrumentos de regulación o control, que son los encargados de mantener la variable controlada en un valor deseado o bien dentro de ciertos límites.
3. Instrumentos actuadores, que actúan directa o indirectamente sobre la variable controlada en función de las señales recibidas de los instrumentos del grupo anterior.

El progresivo ajuste a nivel industrial de costes, así como de los tiempos de desarrollo de los sistemas y automatismos industriales, ha condicionado el reciente desarrollo y la integración de la instrumentación electrónica, derivando el desarrollo moderno hacia la creación de herramientas de ingeniería más eficientes e integradas a lo largo de las diferentes etapas del diseño, lo que incluye la reusabilidad del Software y del Hardware, entre otras. El continuo cambio de normativas o la necesidad de mejorar y aumentar la fiabilidad de los sistemas de control e instrumentación son hechos que afectan directamente al diseño de los proyectos y que fomentan aún más la necesidad de la creación de herramientas eficientes de ingeniería que incorporen reusabilidad de Software y Hardware.

En la actualidad, una de las herramientas de ingeniería disponible para la medición y testeo industrial y de dispositivos se basa en las denominadas redes de instrumentación, que permiten el intercambio automatizado de datos entre instrumentos, dispositivos y otros sistemas. Pueden ser transmisores de medidas, de comandos o de aplicaciones de software utilizadas para ajustar, calibrar y diagnosticar sistemas y dispositivos. Todas estas redes de instrumentación se utilizan para comunicar entre sí sensores de proceso y actuadores que generalmente envían y reciben los datos actuales del sistema a su unidad de control, ofreciendo mecanismos automáticos para el

mantenimiento y control de sensores y actuadores.

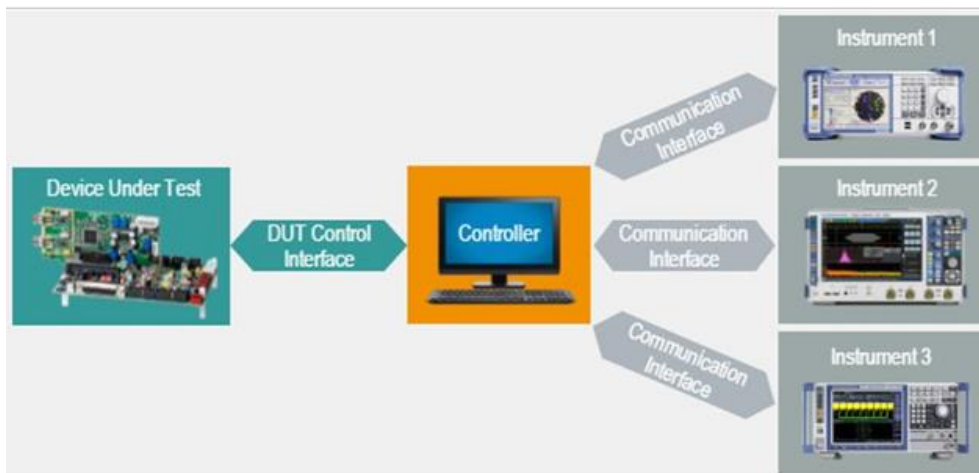


Figura 2-23. Ejemplo de red de instrumentación para la caracterización remota de dispositivos o sistemas (DUT)

Desde un punto de vista hardware (lo que podríamos circunscribir al nivel físico y de enlace en un sistema de telecomunicación), existen multitud de mecanismos que permiten interaccionar los diferentes instrumentos. Algunos de los estándares más importantes en este sentido serán revisados a continuación.

### 2.2.1 DIN-MESSBUS

Conocido también como bus de medición, interfaz de red de bajo coste para medidas y pruebas creada por la industria alemana. Se trata de una interfaz electrónica de bajo coste, que emplea una arquitectura maestro-esclavo y que ha sido expresamente diseñada para comunicar una medición industrial. Una de las particularidades de esta interfaz es que, una vez iniciado el sondeo, un nodo esclavo puede ser desconectado y conectado nuevamente, de manera que una eventual transferencia de datos que se haya interrumpido se reiniciará y continuará sin pérdida de datos. Sus principales características son la sencillez, robustez, arquitectura libre y la sencillez del medio utilizado para la transferencia de la información, el estándar RS 485. La estructura de mensajes permite transmitir valores de medición, configuración y datos de diagnóstico, si bien la flexibilidad y simplicidad de la

### 2.2.2 GPIB

de sus siglas en inglés, General Purpose Interface Bus), HPIB y/o estándar IEEE488. Se trata de una red de instrumentación estándar creada para usos generales. La empresa Hewlett-Packard (HP) inició el desarrollo de esta interfaz versátil y la denominó HPIB, o bus de instrumentación de Hewlett Packard. Dada la popularidad que tomó HPIB, la comisión IEEE 488 lo rebautizó como GPIB, que evolucionó posteriormente a la versión IEEE 488.2, con la integración de la interfaz SCPI (interfaz de comandos estándar para instrumentos programables), con el fin de evitar extensiones de los fabricantes de dispositivos, y proporcionar un GPIB universal. El bus IEEE-488 se desarrolla con la intención de emplearse para gestionar la comunicación con instrumentos inteligentes y programables, proporcionando una interfaz estandarizada para intercambiar datos entre dispositivos de diferentes fabricantes. El medio físico consiste en 24 pares trenzados (cableado paralelo), de los que 8 pares se utilizan para transmitir los datos y los mensajes de comando (para una longitud de datos de 7 u 8 bits), 5 cables se utilizan para gestionar el bus y 3 cables son para gestionar el intercambio de información.

### 2.2.3 LXI

del inglés Network-based Extensions For Instrumentation. Se trata de una red rápida y abierta para instrumentación y pruebas basada en LAN (Local Area Network), muy utilizada en laboratorios de prueba o en departamentos de investigación y desarrollo y que suele considerarse como el sucesor del bus GPIB en muchos sentidos. Es resultado de la migración de los sistemas VXI (red de instrumentación compacto que integra las

mejores características de los buses VME, Versa Module Eurocard, y GPIB) y PXI (bus de instrumentación que incorpora las características principales de PCI, interfaz de componentes periféricos, y GPIB), que integra de forma nativa las capacidades de medida GPIB en Ethernet. Se accede a las funciones proporcionadas por GPIB a través de una puerta de enlace dedicada, y las funciones de sincronización se operan a través de la integración del estándar de comunicación en tiempo real IEEE 1588. La herencia de las capacidades VXI/PXI, de las funciones de Ethernet e Internet, de los rendimientos de medición de GPIB y de los servicios IEEE 1588 hace que LXI sea un sistema de red de equipos de prueba muy potente, rápido, flexible, confiable y compacto. Dado que utiliza Ethernet, LXI también se beneficia de las capacidades de Internet al incorporar un servidor web en los instrumentos, interconectarlos y controlarlos con una aplicación web que se ejecuta, por ejemplo, en un ordenador personal. En consecuencia, los dispositivos ya no necesitan ningún panel de control frontal, y el uso de tecnologías populares y extendidas contribuye a minimizar los costos.

#### 2.2.4 HART

Highway Addressable Remote Transducer), protocolo de comunicación digital muy popular para instrumentación industrial, basado en el estándar de bucle de corriente de 4-20 mA. Sistema de comunicación que ha sido especialmente diseñado para intercambiar datos de configuración, calibración, ajuste y diagnóstico y que se ha mostrado muy útil en trabajos de instalación y mantenimiento.

#### 2.2.5 M-BUS o EN13757

bus de medida para redes de distribución de energía. Este estándar de comunicación digital se utiliza especialmente para leer medidas de consumo y medidas de temperatura a bajo coste. Se trata de un sistema bastante versátil que utiliza topología lineal, topología en estrella o topología en árbol y que puede incorporarse a una estructura de red inalámbrica, haciendo todo eso usando una única estructura maestro/esclavo vinculada a un proceso de comunicación cliente/servidor. Una de sus características más interesantes es que no se especifican los telegramas de datos, dejando libertad a los fabricantes para definirlos y especificarlos. En las versiones cableadas, otra de las características del sistema es que tanto el maestro como los esclavos se comunican modulando las señales a través de la línea de alimentación. Las consultas del maestro se transmiten modulando la tensión de alimentación (un nivel de señal de 36 Vdc significa un nivel lógico de «1», mientras un nivel de señal de 24 Vdc significa un nivel lógico de «0»). Las respuestas del esclavo se transmiten modulando su consumo de corriente (un consumo de corriente de 1,5 mA significa un nivel lógico de «1», mientras que un consumo de corriente entre 11 y 20 mA significa un nivel lógico de «0»).

#### 2.2.6 Otros

Existen otros muchos sistemas que se pueden emplear para establecer una comunicación digital entre diferentes instrumentos electrónicos, como PROFIBUS o CAN. En cualquier caso, tanta versatilidad o, mejor dicho, la falta de un estándar único de comunicación a nivel de enlace dificulta la creación de herramientas de ingeniería que permitan reutilizar el hardware diseñado y la programación de instrumentos de medición puede ser un verdadero quebradero de cabeza.

Para solucionar este problema, se definió a mediados de los 90 el estándar VISA (Virtual Instrument Software Architecture), la cual es una interfaz de programación de aplicaciones usada para el ámbito de pruebas y mediciones de dispositivos, y diseñada para comunicarse con cualquier instrumento desde una aplicación.

VISA proporciona una interfaz de programación entre el hardware y diferentes entornos de programación como pueden ser LabVIEW, LabWindows/CVI, Virtual Studio, Matlab o Python, permitiendo la programación independientemente del bus que se esté utilizando, ya que con una misma librería se puede interactuar con diferentes buses como se muestra a continuación en la siguiente imagen:



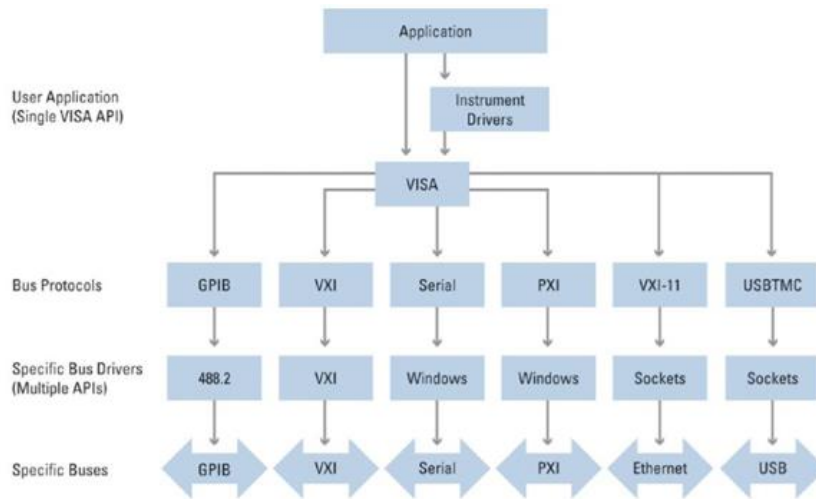


Figura 2-24. Arquitecturas de aplicaciones software que utilicen el estándar VISA

Actualmente es una de las tecnologías más importantes dentro de los sistemas operativos más populares del sector empresarial, ya que compañías destacadas dentro del área de medición y pruebas, como pueden ser Rohde&Schwarz, Keysight Technologies o National Instruments, entre otras, se encargan de su comercialización y distribución.

Dado que Python se ha convertido en uno de los lenguajes de programación más populares en la actualidad debido a su sintaxis, eficiencia y aplicabilidad amigables para principiantes en una amplia gama de casos de uso, en este trabajo utilizaremos este lenguaje de programación para controlar en modo remoto una plataforma de ensayos (test-bench) que nos permita determinar las características reales de un ADC. Es en este punto donde aparece PyVISA, que es un paquete de Python que se apoya en el estándar VISA y que ofrece una solución para el desarrollo de herramientas eficientes de ingeniería que incorporan reusabilidad del software y hardware asociados al control remoto de cualquier red de instrumentación. Se trata de una librería que permite controlar todo tipo de dispositivos de medición independientemente de la interfaz que utilice (GPIB, RS232, USB, Ethernet, etc.). Con PyVISA será posible automatizar los equipos de instrumentación necesarios para la caracterización del ADC, empleando alguno de los enlaces de comunicación que tenga incorporados el instrumento (GPIB, USB, serie o LAN), Python y un ordenador personal en el que se cree un puerto serie virtual. En el ordenador personal se podrá programar la FFT para obtener la caracterización de los diferentes parámetros del ADC.

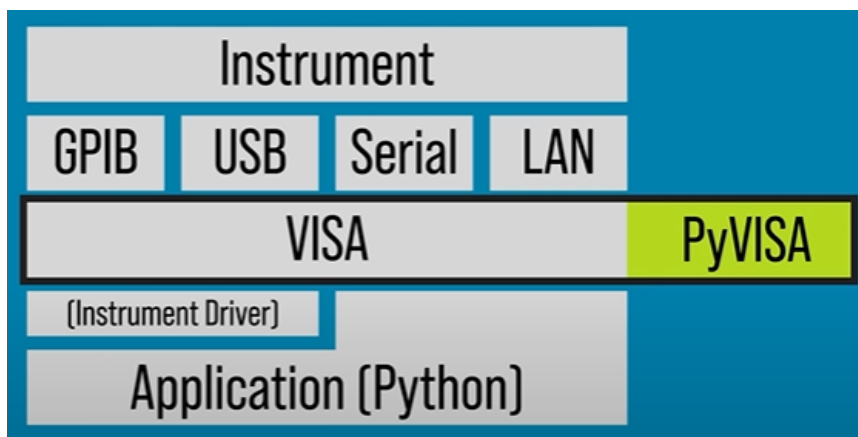


Figura 2-25. Características generales de la aplicación software para la caracterización de un ADC



### 3 DESCRIPCIÓN DEL BANCO DE PRUEBAS

A la hora de caracterizar el funcionamiento de un dispositivo de conversión analógico-digital, en nuestro caso un ADC de tipo sigma-delta, es necesario desarrollar un banco de ensayos que permita determinar tanto su correcto funcionamiento como sus prestaciones. Para ello, el banco de pruebas que se diseñe debe poder evaluar el convertidor analógico-digital bajo prueba (en adelante, DUT – Device Under Test) en todo su rango de operación, permitiendo a los diseñadores obtener de forma experimental sus prestaciones.

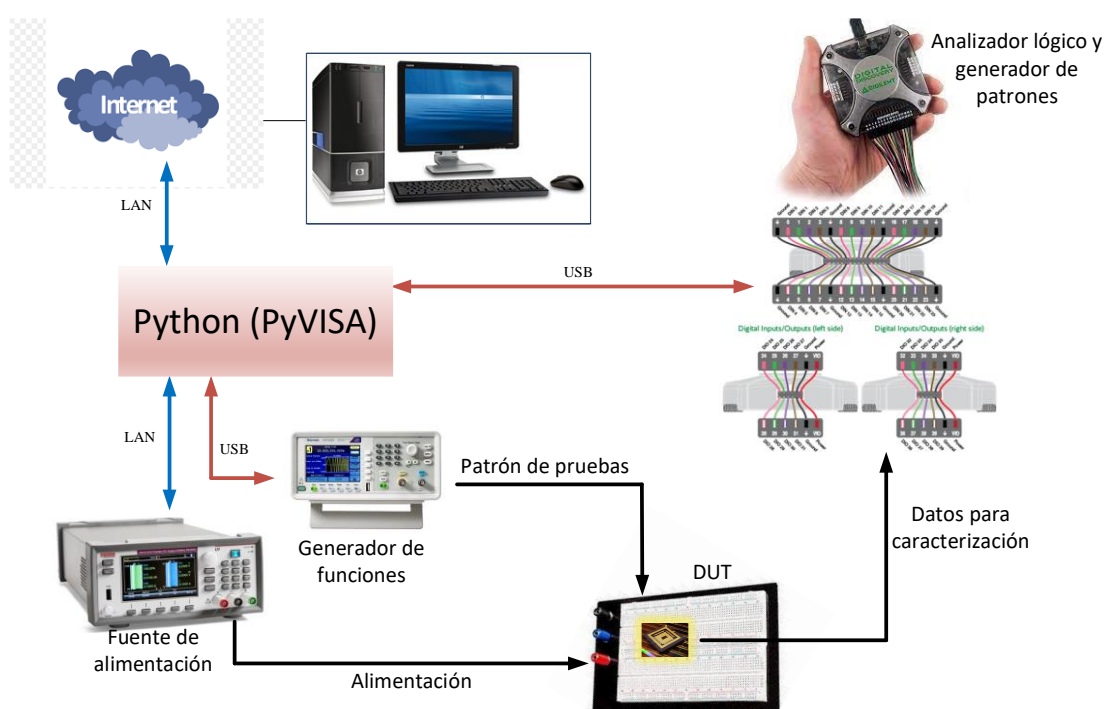


Figura 3-1. Banco de pruebas del Sigma-Delta

La Figura 3-1 representa un diagrama de bloques del banco de pruebas implementado. Como se puede observar, el sistema está compuesto por los siguientes bloques:

- Bloque de excitación: está compuesto por un generador de funciones Tektronix AFG1022 para la inyección de las señales de entradas del DUT y el módulo de generación de patrones digitales del dispositivo Digital Discovery, del fabricante Digilent.
- Bloque de alimentación: está compuesta por una fuente de alimentación de banco de la marca Keithley. En concreto, se hará uso del modelo 2281-S.
- Bloque adquisición: se hará uso del módulo analizador lógico que integra el instrumento Digital Discovery.
- Bloque de control: está compuesto por una herramienta software desarrollada en Python. Esta se encarga de generar las diferentes fases de reloj requeridas por el DUT, configurar los diferentes instrumentos de medida, adquirir la salida del DUT y, posteriormente, procesar esta información para generar la información necesaria que permita evaluar las prestaciones del DUT.

Por último, para gestionar los diferentes elementos se hará uso de los siguientes protocolos:

- VISA: control del AWG y fuente (indicar que uno es por USB y otro por LAN)
- Digital Discovery SDK: conjunto de funciones que suministra Digilent para gestionar el instrumento Digital Discovery.

El sistema consta de una serie de elementos hardware, que incluyen equipos de instrumentación electrónica para alimentar el DUT, controlar su funcionamiento e imponer diferentes patrones de entradas analógicas que se deseen convertir. Constará, además, de instrumentación electrónica para monitorizar la salida digital generada por el propio DUT. Todos los dispositivos serán controlados desde un PC, empleando un software específico diseñado en lenguaje Python, que permitirá la adquisición de las salidas que genere el DUT sincronizadas con los diferentes patrones de señales analógicas de entrada que se impongan. El software que se diseñe incluirá herramientas de procesamiento de señal que nos permitan extraer las principales características del ADC.

### 3.1 Instrumentación electrónica del Banco de Pruebas

A continuación, se van a describir las principales características de los instrumentos seleccionados.

#### 3.1.1 Keithley 2281-S

Para poder realizar un banco de pruebas que pueda controlarse de forma remota, uno de los requisitos que debe presentar la fuente de alimentación empleada es que sea compatible con algunos de los protocolos de instrumentación descritos en el Capítulo 2. Además, será necesario que la regulación de carga y de línea sean suficientemente altas como para no alterar las medidas obtenidas. Asimismo, se seleccionará una fuente de alimentación cuya tensión de salida presente bajo ruido. Debido a estos requisitos, emplearemos la fuente de alimentación Keithley 2281-S, que permite generar una salida de tensión en DC programable hasta 20 V y que ofrece una corriente máxima a dicha tensión de 6 A (máximo 120 W generados).

Es interesante destacar que la fuente de alimentación seleccionada tiene la capacidad de ofrecer medidas de corriente de carga desde los 100 nA y tensiones de alimentación desde 100  $\mu$ V, con una resolución en la medida de hasta 6½ dígitos.



Figura 3-2. Fuente de alimentación Keithley 2281-S

Ofrece E/S digitales para comunicación directa con otros dispositivos mediante las Interfaces GPIB (General Purpose Interface Bus o estándar IEEE488), USB y LAN. Dado que queremos controlar el instrumento en modo remoto empleando Python y VISA, es importante destacar en este punto la existencia de drivers y librerías en el repositorio GitHub para este dispositivo: <https://github.com/tektronix/keithley/tree/main/Drivers/PS-228xS>.

Las librerías localizadas emplean el enlace de comunicación LAN, que será el empleado en nuestro trabajo.

A continuación, se muestran unas tablas resumen con las principales características del dispositivo.

Tabla 3-1. Características principales del equipo Keithley 2281-S

Parámetro	Rango
Voltaje	0-12 Voltios
Corriente	0-6 Amperios
Potencia Máxima	120 Watios
Precisión de la tensión de salida	$\pm(0.02\% + 3 \text{ mV})$
Resolución de la Tensión de Salida	1 mV
Regulación de Carga	$\pm(0.01\% + 2 \text{ mV})$
Regulación de Línea	$\pm(0.01\% + 1 \text{ mV})$
Amperios a Escala Completa	6.1 A
Precisión de la Corriente de Salida	$\pm(0.05\% + 5 \text{ mA})$ .
Resolución de la Corriente de Salida	0.1 mA
Modos de Visualización de Medición	de Voltaje y corriente, solo voltaje, de solo corriente
Control de Adquisición de Medición	Continuo, manual, entrada digital externa, bus de PC
Corriente de Modo Común	$<6 \mu\text{A}$ pico a pico (típico)

### 3.1.2 Tektronix AFG1022

En relación con la selección del generador de funciones a emplear, se han explorado dos opciones. En primera instancia, se consideró la posibilidad de utilizar el AM300 del fabricante Rhode-Schwarz (R&S). El R&S AM300 es un generador de funciones arbitrarias de doble canal que ofrece una funcionalidad y pureza espectral notables. Puede programar la generación digital de señales analógicas con una distorsión mínima, incluso en niveles de salida y frecuencias elevadas. Este generador permite implementar una amplia variedad de tipos de onda debido a su alta tasa de muestreo (hasta 100 Msample/s) y una memoria de forma de onda de 256k-puntos por canal. Ambos canales pueden configurarse individualmente en términos de frecuencia, forma de onda y amplitud, y también es posible acoplarlos en fase, permitiendo la generación de señales I/Q analógicas que pueden utilizarse, por ejemplo, para modular el generador de señales de RF. En cuanto a la frecuencia, el límite superior es de 35 MHz para señales sinusoidales y 50 MHz para señales cuadradas.

El software de control en modo remoto proporcionado por el fabricante requiere una conexión a un dispositivo con sistema operativo Windows (2000 y XP) y la Interfaz USB para el enlace físico. Para la programación en modo remoto, el fabricante brinda opciones como Visual C/C++, Visual Basic, LabView o LabWindows/CVI de National Instruments, así como VEE de Agilent. Por otro lado, el driver VXIplug&play también habilita su uso desde el entorno Matlab. Existe una alternativa a PyVISA que ofrece R&S, denominada Pycharm, que permitiría manejar en modo remoto instrumentación de este fabricante, pero luego de analizarse en detenimiento se observó que no se puede usar para este instrumento.

Es importante destacar que, para las pruebas en las que sea necesario, se contempla la utilización de un generador de señales de baja distorsión. Sin embargo, debido a que el R&S AM300 no es compatible con un protocolo de instrumentación estándar, su integración en el banco de pruebas actual se encuentra fuera del alcance de este trabajo.

El segundo instrumento considerado, y que finalmente ha sido el elegido para formar parte del banco de pruebas diseñado, es el AFG1022 de Tektronix, generador de funciones arbitrarias que incluye dos canales, dispone de

un ancho de banda de 25 MHz y genera señales de una amplitud de salida de hasta 10 Vpp. Los cuatro modos de ejecución, las 50 formas de onda integradas de uso frecuente y el contador de frecuencia integrado de 200 MHz de que dispone el dispositivo cubren la mayoría de las necesidades de generación de formas de onda para trabajos de experimentación y prueba. Respecto al generador de ondas R&S, este dispositivo ofrece librerías en Python gratuitas y de libre acceso para su control a través de PyVISA y la interfaz USB.



Figura 3-3. Tektronix AFG1022

Tabla 3-2. Características principales del equipo Tektronix AFG1022

Parámetro	Rango
Número de canales	2
Formas de Onda Incorporadas	Seno, Cuadrada, Pulso, Rampa, Ruido y 45 formas de onda arbitrarias frecuentemente utilizadas
Rango de Onda Senoidal	1 $\mu$ Hz a 25 MHz
Frecuencia Efectiva	Máxima 25 MHz
Distorsión Armónica	< -50 dBc, 1 Vp-p, 1 $\mu$ Hz a 25 MHz
Distorsión Armónica Total	< 0.2% (10 Hz a 20 kHz, 1 Vp-p)
Espurias	< -45 dBc, 1 Vp-p, 1 $\mu$ Hz a 25 MHz
Ruido de Fase a 1 MHz	< -110 dBc/Hz a 10 kHz de desplazamiento, 1 Vp-p (típico)
Ruido Residual de Reloj	-57 dBm (típico)
Rango de Onda de Pulso	1 mHz a 12.5 MHz
Resolución de Ancho de Pulso	10 ps o 5 dígitos
Ciclo de Trabajo	<1 MHz: 0.001% a 99.999% (aplican limitaciones de ancho del ciclo del pulso) 1 MHz a 12.5 MHz: 50% fijo
Ruido	Ancho de Banda de Ruido (-3 dB): 25 MHz Tipo de Ruido: Gaussiano Blanco Rango de CC: -5 V a +5 V, carga de 50 $\Omega$ 10 V a +10 V, circuito abierto o carga de alta Z
Rango de Onda Arbitraria	1 $\mu$ Hz a 10 MHz
Ancho de Banda Analógica Efectiva (-3 dB)	30 MHz
Memoria No Volátil	64 MByte
Tasa de Muestreo	125 MS/s
Resolución Vertical	14 bits
Tiempo de Subida/Descenso	< 10 ns
Resolución de Frecuencia	de 1 $\mu$ Hz o 12 dígitos
Precisión de Amplitud	1 mVp-p, 1 mVrms o 4 dígitos
Impedancia de Salida	50 $\Omega$
Protección de Salida de	Tolerancia a cortocircuitos. La

---

Señal	salida principal se desactiva automáticamente en caso de exceso de corriente
-------	--

---

### 3.1.3 Digital Discovery

Además de los dos instrumentos comentados anteriormente, necesitaremos un instrumento que nos permita generar y visualizar señales, de las cuales hablaremos en apartados posteriores. Este instrumento, como bien se ha comentado antes, permite visualizar y generar señales digitales. Se trata de un dispositivo de medida muy versátil ya que integra las funcionalidades de diferentes instrumentos de medida en un dispositivo de tamaño reducido y de bajo coste.



Figura 3-4. Digital Discovery

De forma específica, se puede configurar para que actúe como algunos de los siguientes instrumentos:

1. Analizador lógico digital de 24 canales (CMOS de 1.2 hasta 3.3V), con capacidad de realizar hasta 800 MS/s utilizando un adaptador de alta velocidad.
2. Generador de patrones de 16 canales (CMOS de 1.2 hasta 3.3 V), con capacidad de generar 100 MS/s.
3. Puertos de E/S digital virtual de 16 canales que incluye botones, interruptores y LEDs.
4. Una fuente de alimentación programable desde 1.2 hasta 3.3 V y 100 mA como máximo. El mismo voltaje alimenta los buffers de entrada del analizador lógico y los buffers de entrada/salida del generador de patrones, para mantener así la compatibilidad del nivel lógico con el circuito de prueba.
5. Analizadores de bus digital (SCPI, I2C, UART, I2S, CAN, Paralelo).

Este dispositivo irá conectado a un ordenador mediante un cable USB. El fabricante facilita junto con el dispositivo hardware un conjunto de herramientas de desarrollo que permiten crear soluciones de software personalizadas para manejarlo remotamente desde un ordenador personal. Este conjunto de herramientas (API SDK de WaveForms) está disponible para diferentes lenguajes de programación, lo que facilita notablemente su uso.

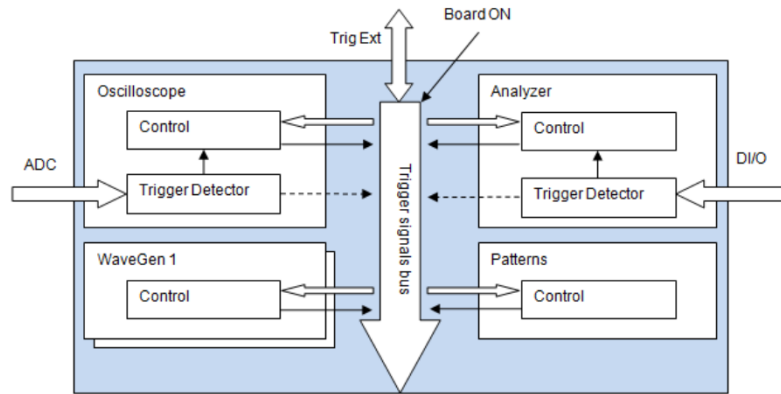


Figura 3-5. Instrumentación del Digital Discovery

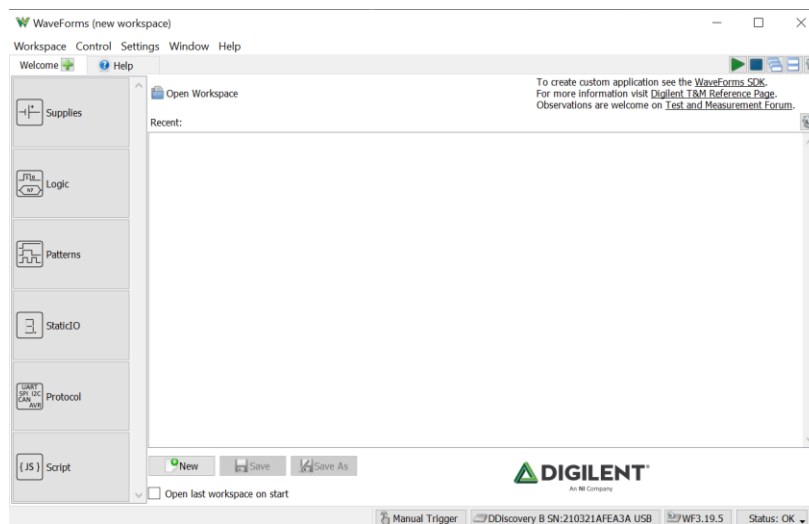


Figura 3-6. Programa WaveForms

En la imagen superior se incluye una captura de la pantalla de inicio del programa WaveForms, suite que convierte Digital Discovery en un instrumento virtual, permitiendo la configuración, calibración y el control del dispositivo de prueba y medición al que se conecte. WaveForms proporciona una interfaz gráfica que permite a los usuarios interactuar con el hardware del dispositivo. Asimismo, WaveForms SDK ofrece acceso a los usuarios de este sistema a una interfaz de programación de aplicaciones públicas (API), lo que permite crear aplicaciones de PC personalizadas.

El paquete software que ofrece WaveForms a los usuarios del Digital Discovery está compuesto de multitud de componentes, siendo el más visible la propia aplicación WaveForms. Existen, sin embargo, un conjunto de paneles de instrumentos gráficos que brindan acceso completo a los diferentes instrumentos analógicos y digitales que nos ofrece el dispositivo a través de la aplicación WaveForms Runtime, compuesta de la biblioteca dinámica DWF (que utiliza Digilent Adept Runtime para proporcionar comunicación básica con los instrumentos hardware específicos) y de varios archivos de configuración. Todo lo necesario para escribir aplicaciones personalizadas que utilicen el Digital Discovery se incluye en el SDK de WaveForms, que proporciona los archivos de encabezado/biblioteca y la documentación para acceder a la API de la biblioteca DWF. La aplicación personalizada que se desarrolle deberá vincularse correctamente a estos archivos para realizar las llamadas de función API adecuadas, de forma que cada función en la API pública de WaveForms se deberá declarar en el archivo de encabezado dwf.h. El uso básico de la API de WaveForms se puede dividir en



los siguientes pasos:

- Llamada a las funciones de enumeración para descubrir los dispositivos de hardware que estén conectados.
- Llamada a la función `FDwfDeviceOpen` para establecer una conexión con un dispositivo de hardware específico.
- Llamada a la función para habilitar la conexión.
- Llamada a la función para configurar el instrumento y adquirir y generar señales.
- Llamada a la función para deshabilitar el dispositivo.
- Llamada a la función `FDwfDeviceClose` para desconectarse del dispositivo.

Hay nueve grupos principales de funciones API, cada uno nombrado con prefijos diferentes, entre los cuales destacamos dos de ellos, que nos harán falta en nuestro caso: las funcionalidades `DigitalIn` (Analizador Lógico) y `DigitalOut` (Generador de Muestras).

El instrumento de medidas de entrada (`DigitalIn`) funciona respetando el diagrama de bloques que se muestra en la Figura 3-7. Los diferentes estados se definen en `dwf.h` como tipo `DwfState`:

- **Ready:** Estado inicial, al que se llega después de `FDwfDigitalInConfigure` (la reconfiguración pasa al estado “Configure”) o de cualquier llamada de función `FDwfDigitalIn*Set`.
- **Configure,** que reestablece el trigger.
- **Prefill,** que precarga el buffer con las muestras necesarias antes de cada disparo (trigger).
- **Armed,** que espera la ejecución de un trigger.
- **Running:** Estado en ejecución, en el que la pantalla de escaneo permanece hasta que la configuración o cualquier función establecida de este instrumento concluya.
- **Done:** Estado final.

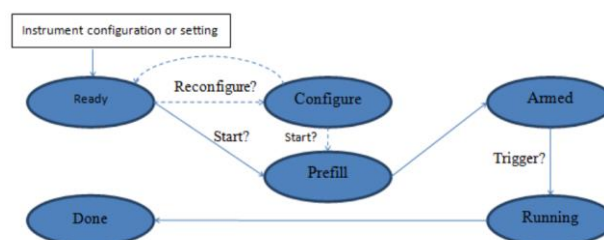


Figura 3-7. Diagrama de bloques `DigitalIn`

Por otro lado, el instrumento que genera las diferentes muestras, `DigitalOut` o Generador de patrones, funciona siguiendo el siguiente diagrama de bloques:

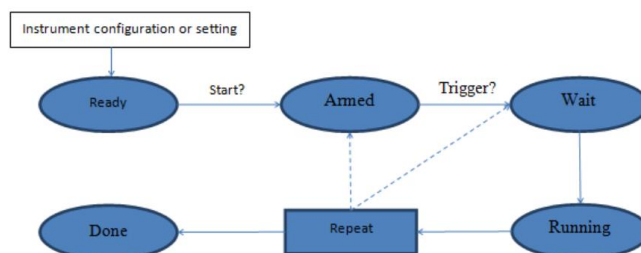


Figura 3-8. Diagrama de bloques `DigitalOut`

Los diferentes estados, al igual que en Analizador Lógico, se definen en dwf.h como tipo DwfState:

- Ready: Estado inicial, al que se llega después de FDwfDigitalOutConfigure o de cualquier llamada a la función FDwfDigitalOut\*Set.
- Armed, estado en el que se espera un disparo o trigger.
- Wait, estado en el que permanece durante el periodo de tiempo especificado por la función FDwfDigitalOutWaitSet.
- Running o ejecución. Estado en el que permanece el tiempo que se haya especificado con la función FDwfDigitalOutRunSet.
- Repeat. Pasa al estado Armed o Wait, en función de la configuración de FDwfDigitalOutRepeatTriggerSet, la cantidad de veces especificada por FDwfDigitalOutRepeatSet.
- Done: Estado final.

Las máquinas de estados controlan todos los canales digitales del dispositivo, que ofrecen una gran variedad de configuraciones. Los valores iniciales para los registros divisor y contador especifican los valores que se cargan en ellos inicialmente, al entrar en estado Running, y hacen referencia al retardo inicial programado para el disparo. El divisor especifica cómo se realiza la división del reloj, estableciendo la frecuencia personalizada de muestra del dispositivo, así como el paso para el contador. Al ingresar al estado en ejecución, se carga un valor inicial con FDwfDigitalOutDividerInitSet, de manera que cuando este valor expire (el valor inicial se irá decrementando con el transcurso del tiempo), el valor definido por FDwfDigitalOutDividerSet se cargará en cada vencimiento. El valor inicial del contador se establece mediante la función FDwfDigitalOutCounterInitSet, que también establece el nivel inicial de la señal y cuyos valores se recargan con la función FDwfDigitalOutCounterSet. Al expirar el contador, el nivel de salida conmuta automáticamente, salvo que algún valor inicial sea cero lo que provoca que se mantenga el nivel. El Modo de salida (FDwfDigitalOutModeSet) selecciona entre PP, OS, OD y TS. La salida inactiva (FDwfDigitalOutIdleSet) selecciona la salida mientras no está en estado de ejecución. El nivel y valor iniciales se especifican para la señal de pulso con la función FDwfDigitalOutCounterInitSet, y se cargan al entrar en el estado de ejecución.

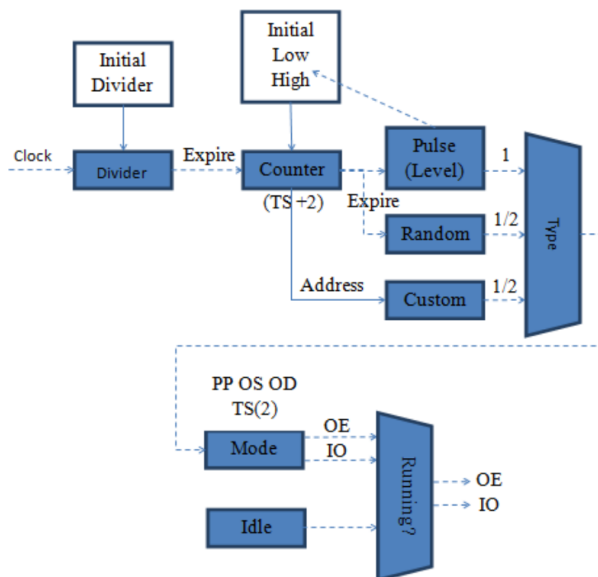


Figura 3-9. Registros

El contador se usa para:

- Pulse: establecer los estados de nivel alto y bajo.
- Random: establecer la tasa de actualización.
- Custom: direccionar al buffer. Las muestras se configuran mediante la función

FDwfDigitalOutDataSet. Esto también configura el contador bajo/alto según el parámetro countOfBits. En el modo TS, el paso del contador es doble y proporciona dos bits de muestras para la salida: valor y habilitación.

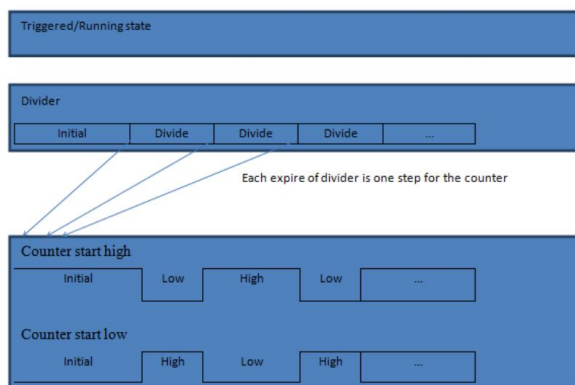


Figura 3-10. Contador

## 3.2 Sigma-Delta a evaluar

Se trata de un ADC Sigma-delta basado sobre la técnica pseudo-pseudo diferencial desarrollado en [1], que incorpora una implementación totalmente diferencial. En este apartado se va a proceder a describir brevemente el sistema, puesto que sirve de base para la realización de este trabajo fin de grado. No se va a entrar en detalle en su estructura interna, puesto que sólo nos centraremos en la generación de las señales de control y en la adquisición de los datos obtenidos del circuito para su caracterización. Sí se intentará describir la necesidad de las señales de control, así como su temporización, que son la base que permiten hacer funcionar al ADC.

El ADC desarrollado se basa en capacidades conmutadas, e intenta mitigar dos de las principales limitaciones para su uso en aplicaciones biomédicas: el ruido Flicker, que se incorpora a la banda de señal, y la distorsión armónica. Para ello, y para aumentar la linealidad en el funcionamiento del ADC, éste se implementa empleando la técnica Pseudo-Pseudo Diferencial (P2D), en la que una señal de un solo extremo se procesa de forma diferencial. En [1] se presenta la primera implementación en silicio del ADC a analizar en este trabajo, que se basa en un modulador delta-sigma de condensadores conmutados basado en comparador (CBSC) de segundo orden sobre una variante de la técnica P2D. El dispositivo presentado se denomina SeP-D, y su arquitectura interna y temporización de señales se muestran en las siguientes figuras:

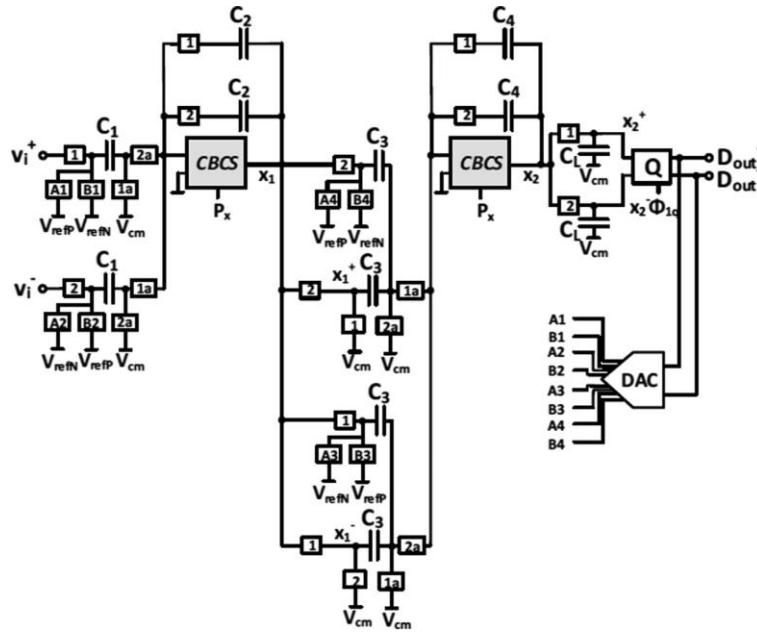


Figura 3-11. Arquitectura interna del ADC propuesto en [1], denominado SeP-D

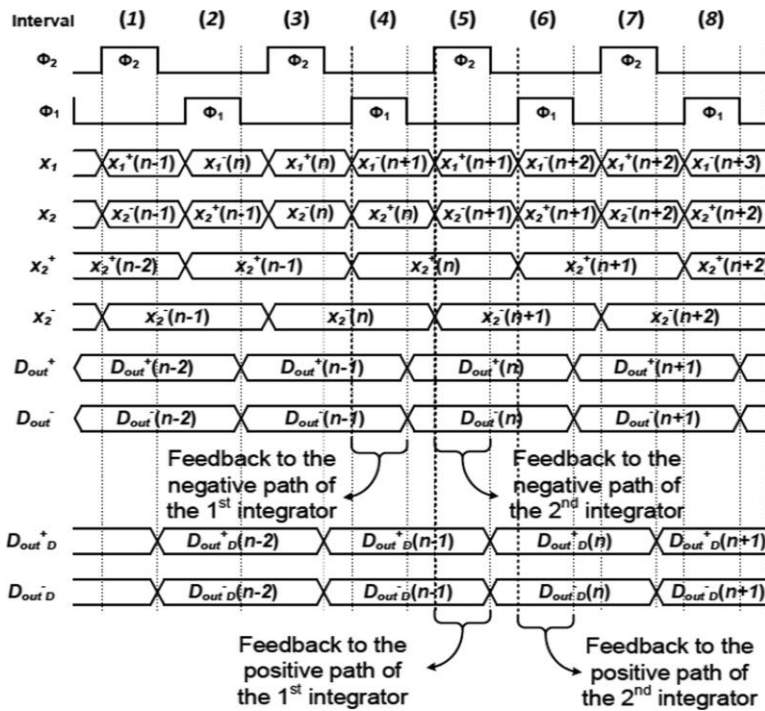


Figura 3-12. Temporización asociada al ADC propuesto en [1]

Podemos observar en Figura 3-11 y Figura 3-12 que esta estructura logra una operación verdaderamente diferencial mediante el procesamiento de señales de modo único. La operación es la siguiente: durante  $\phi_1$ , la rama positiva se encuentra en la fase de transferencia de carga, mientras que la negativa está en la fase de muestreo. Lo contrario ocurre durante  $\phi_2$ .

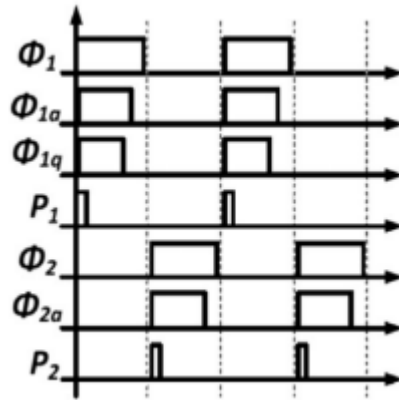


Figura 3-13. Fases de control

Las fases de reloj también están representadas en la Figura 3-13, donde  $\phi_{1a}$  y  $\phi_{2a}$  son las versiones avanzadas de  $\phi_{11}$  y  $\phi_{22}$ , respectivamente.  $P_1$  y  $P_2$  son las fases preestablecidas requeridas en una implementación CBSC y  $\phi_{1q}$  es la señal de control del cuantificador.

He de comentar que en [1] se compara experimentalmente la arquitectura propuesta con otra de tipo P-D, implementando ambas arquitecturas en la misma pastilla de silicio:

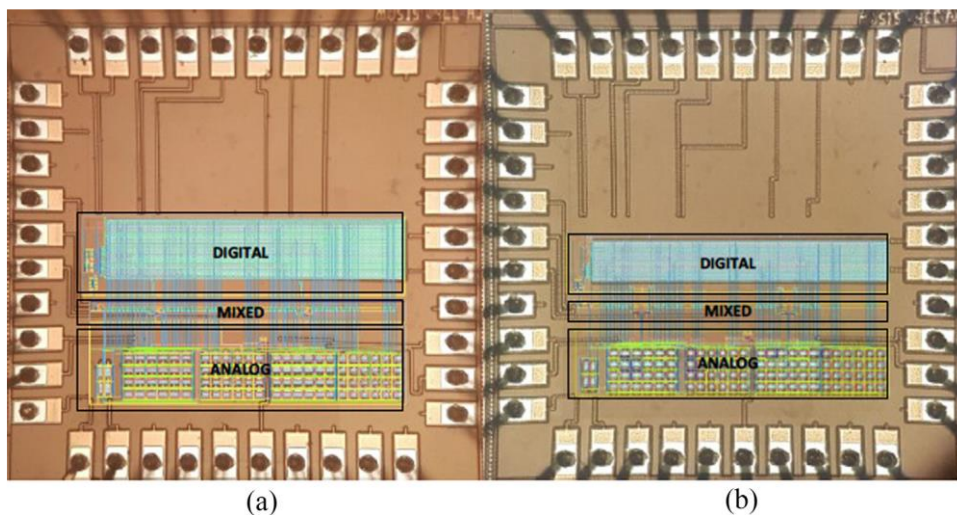


Figura 3-14. Implementación del ADC propuesto en [1] y denominado SeP-D (a), y de uno de tipo P-D (b). La implementación se realizó en tecnología estándar CMOS de 180 nm

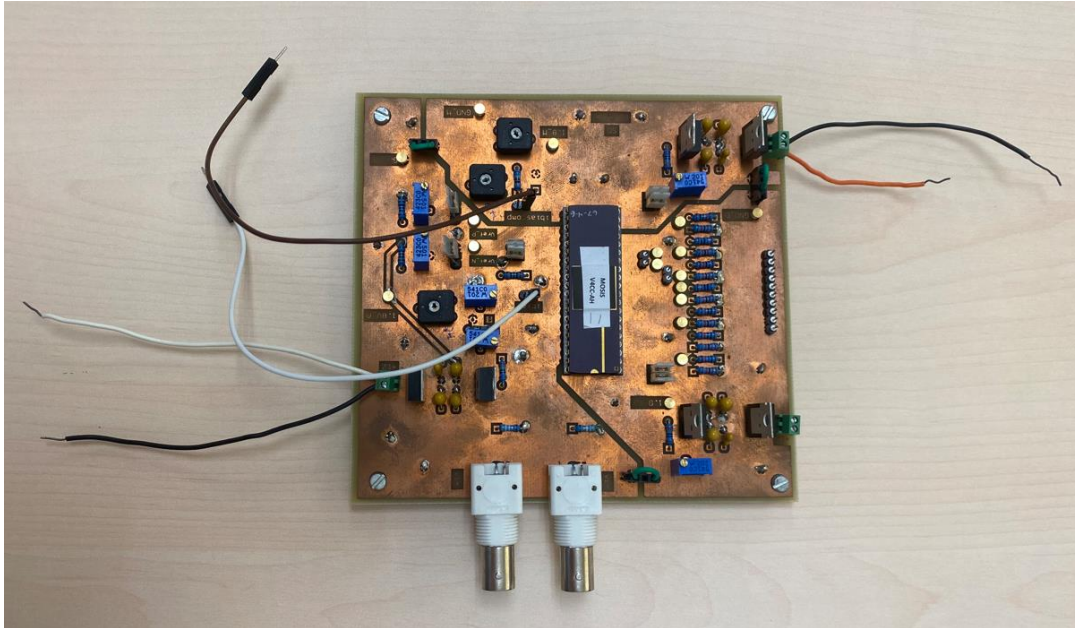


Figura 3-15. Prototipo de placa de circuito impreso en la que se aprecia el ADC desarrollado en la tesis doctoral de Doña Elena Cabrera

### 3.3 Herramienta para la gestión del banco de pruebas

Es relevante hacer una distinción entre la aplicación de control y la instrumentación, ya que cumplen roles diferentes dentro del banco de pruebas diseñado para caracterizar el convertidor.

La aplicación de control consiste en el software o la aplicación utilizada para regular las condiciones operativas del convertidor durante la prueba. Esto permite al usuario ajustar parámetros como la frecuencia, la tensión o la corriente para evaluar su rendimiento. La aplicación puede ser personalizada o provenir de una solución comercial, y puede ejecutarse en una computadora o en un teléfono móvil.

Por otro lado, la instrumentación de medición abarca el conjunto de dispositivos de medición empleados para analizar y medir las señales eléctricas en varios puntos del sistema. Estos dispositivos han sido previamente detallados.

Es importante mencionar que la implementación detallada de cada uno de los bloques mencionados se abordará en el siguiente capítulo.

En cuanto a la aplicación de control, es relevante destacar que se ha desarrollado en Python y está diseñada para ser ejecutada en un PC con sistema operativo Linux. La versión específica de Python utilizada y los detalles de la configuración del equipo serán proporcionados también en el siguiente capítulo. Cabe destacar que el dispositivo que ejecute la aplicación puede ser cualquier ordenador conectado a la misma red que la fuente de alimentación, y que tenga conectado el AWG y el dispositivo Digital Discovery de Digilent a través de USB. Por lo tanto, se requerirá la configuración adecuada de los controladores y otros componentes. En caso de optar por un entorno Windows, el PC que ejecute la herramienta de control deberá cumplir con requisitos específicos

# 4 IMPLEMENTACIÓN DE LA HERRAMIENTA DE CONTROL Y PROCESAMIENTO

---

El funcionamiento del programa desarrollado en Python debe realizar las tareas que se resumen a continuación y que se indican en el diagrama general de flujo del programa, Figura 4-1.

La aplicación se organiza en tres grandes bloques:

- Configuración de los instrumentos del banco de medida: funciones de tipo CONTROL SETUP.
- Adquisición de datos del DUT (ADC de tipo Sigma-delta): funciones ADQ.
- Almacenamiento y procesado de los datos obtenidos: funciones de tipo PROCESADO.

Estas funciones se realizan según el siguiente diagrama de flujos:

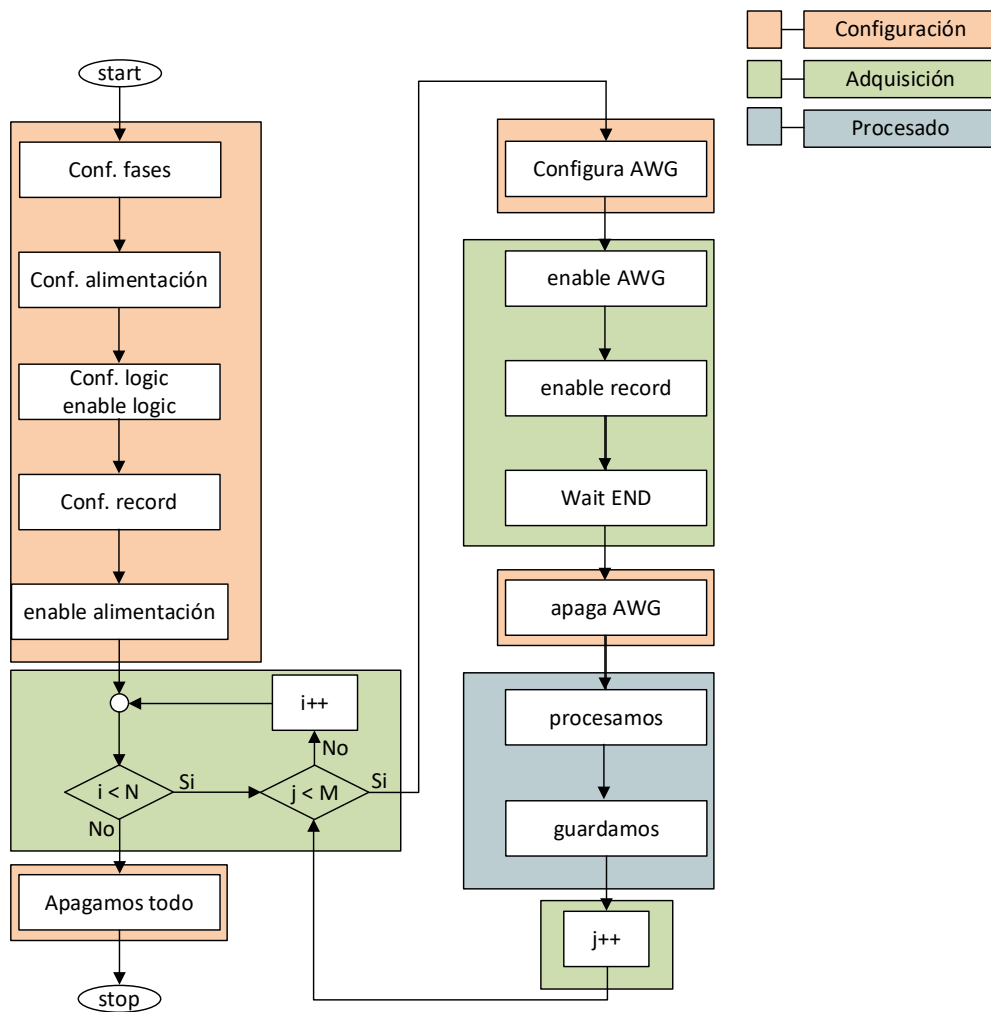


Figura 4-1. Diagrama de flujo que sigue la herramienta de control y procesado desarrollada.

## 4.1 Configuración de la Herramienta de control y procesado

Con el objetivo de hacer la aplicación lo más general posible, se han definido un conjunto de parámetros que definan el funcionamiento de la aplicación. De esta forma, se podrá adaptar la herramienta a nuevos convertidores sin necesidad de modificar las funciones desarrolladas para la configuración, gestión y procesado de la herramienta.

La Tabla 4-1 recoge una descripción del significado de cada parámetro, así como el valor asignado para el banco de pruebas implementado.



Tabla 4-1. Descripción de los parámetros definidos

Categoría	Parámetro	Descripción	Valor asignado
Control de la prueba	REP	Número de repeticiones del experimento que se desea realizar	1
Instrumentos de medida	VDD_DUT	Establece la tensión de alimentación de la tarjeta de prueba que contiene el DUT a evaluar	3.3 [V]
	I_LIMIT_DUT	Fija el límite de corriente máximo de la fuente de alimentación empleada para evitar daños en caso de producirse un mal funcionamiento del DUT	0.5 Amps
	GEN_PERIOD	Establece el período de muestreo del generador de patrones integrado en el Digital Discovery	10
	PS_ADDRESS	Dirección IP asignada a la fuente de alimentación.	192.168.1.21
	AWG_ADDRESS	Dirección VISA del generador de funciones empleado.	USB0::1689::851::1650092::0::INSTR
Características del Sigma Delta a evaluar	ADC_SAMPLING_PERIOD	Periodo de adquisición del convertidor	78125 ns
	ADC_SAMPLING_FREQ	Frecuencia de adquisición del convertidor	12800 Hz
	ADC_OSR	Factor de sobremuestreo del convertidor a evaluar	64
	ADC_NUM_PHASES	Establece el número total de fases a generar para controlar el convertidor bajo prueba	6

Categoría	Parámetro	Descripción	Valor asignado
Temporización	PH1_PULSE_WIDTH	Duración del tiempo que la fase de control 1 está activa (uno lógico)	3904.25 ns
	PH1_DELAY	Retraso introducido a la fase 1	0
	PH2_PULSE_WIDTH	Duración del tiempo que la fase de control 2 está activa (uno lógico)	3907.15 ns
	PH2_DELAY	Retraso introducido a la fase 2	39061.5 ns
	PH3_PULSE_WIDTH	Duración del tiempo que la fase de control 3 está activa (uno lógico)	3886.51 ns
	PH3_DELAY	Retraso introducido a la fase 3	0
	PH4_PULSE_WIDTH	Duración del tiempo que la fase de control 4 está activa (uno lógico)	3866.51 ns
	PH4_DELAY	Retraso introducido a la fase 4	0
	PH5_PULSE_WIDTH	Duración del tiempo que la fase de control 5 está activa (uno lógico)	3907.35 ns
	PH5_DELAY	Retraso introducido a la fase 5	39061.5 ns
	PH6_PULSE_WIDTH	Duración del tiempo que la fase de control 6 está activa (uno lógico)	3871.15 ns
	PH6_DELAY	Retraso introducido a la fase 6	30 ns
Procesado	N_CYCLES	Número de ciclos completos de señal adquiridos	3
	N_TOTAL_SAMPLES	Número de muestras totales	$TW/ADC\_SAMPLING\_FREQ$
	N_SAMPLES_BW	Número de muestras del ancho de banda	$N\_TOTAL\_SAMPLES/(2 * ADC\_OSR)$
	F_SIGNAL	Frecuencia de la señal	Se corresponde con los puntos del espacio de test
	TW	Tamaño de la ventana de adquisición	$N\_CYCLES/F\_SIGNAL$

## 4.2 Módulo de configuración de los instrumentos

En esta sección, se van a describir el procedimiento realizado para configurar los instrumentos de medida y definir los puntos del espacio de prueba.

### 4.2.1 Generador de los puntos del espacio de prueba

A la hora de planificar la herramienta de prueba, se ha considerado que se van a realizar mediciones de múltiples frecuencias, así como diferentes valores de amplitud. Por tanto, el espacio de pruebas a realizar quedará establecido por el número de frecuencias y amplitudes a evaluar. Con el objetivo de aportar flexibilidad al usuario, se han desarrollado dos funciones que permiten generar un número arbitrario de puntos usando diferentes aproximaciones.. Esta metodología se ha implementado mediante la utilización las funciones: *generar\_vectores\_amp* y *generar\_vectores\_freq*.

La función *generar\_vectores\_amp* se ha diseñado con el propósito de generar vectores con diferentes valores de amplitud. Para ello, la función admite como argumentos el rango de amplitudes que se desea barrer (en el caso del DUT, este varía entre 10mV y un valor máximo de 1.8Vpp), el número de puntos a generar y el tipo de interpolación que se desea implementar. Para ello, se ha implementado dos tipos:

- **Interpolación lineal (“lin”).** La función devuelve un vector de longitud el número de puntos especificados cuya separación entre sus componentes es constante.
- **Interpolación logarítmica (“log”).** La función devuelve un vector de longitud el número de puntos especificados cuya separación entre sus componentes es constante.

Finalmente, cabe indicar que el usuario puede definir el vector de amplitudes de forma manual. En ese caso, la función *generar\_vectores\_amp* no altera su valor. En la Figura 4-2 se muestra el diagrama de flujo asociado.

De manera análoga, la función *generar\_vectores\_freq* se ha implementado con el fin de generar vectores de frecuencia. Para el DUT analizado, el vector de frecuencias se ha seleccionado entre el rango de frecuencias desde 10 Hz hasta 30 Hz utilizando una interpolación lineal. El número de puntos elegidos ha sido de 10. La Figura 4-2, figura de la derecha, muestra una representación del funcionamiento de la función.

En ambos casos, el tercer parámetro, establecido en "3", denota el número de puntos o valores a generar en cada conjunto. Esta selección permite controlar la densidad de los datos generados, ofreciendo flexibilidad en la adaptación de la información a las necesidades específicas del análisis o experimentación.

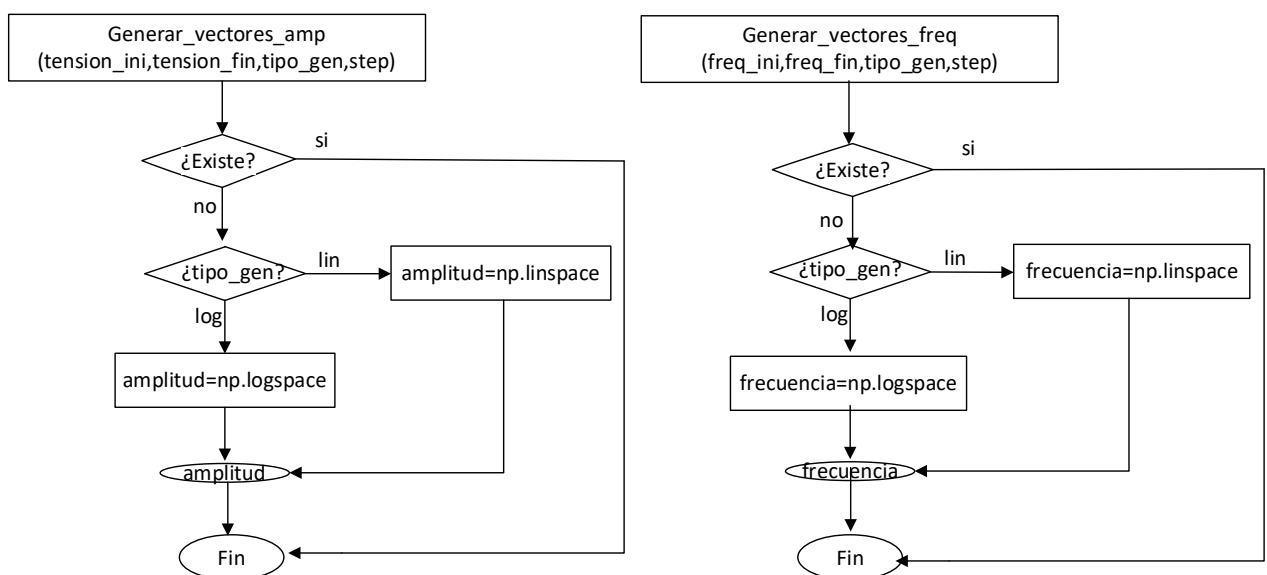


Figura 4-2. Diagramas de flujo de las funciones que determinan la amplitud (izquierda) y la frecuencia

(derecha) del barrido de la onda generada en función del tipo (lineal o logarítmica), los valores inicial y final, así como el paso entre señales generadas.

#### 4.2.2 Configuración del generador de patrones: temporización

Tal y como se ha mostrado en la sección 3.2 del capítulo 3, es necesario generar 6 señales de control con el módulo lógico del instrumento Digital Discovery que estén sincronizadas entre sí. Estas señales son necesarias para controlar el correcto funcionamiento del ADC bajo prueba. Estas señales gobiernan el proceso de generación de la conversión que realiza el DUT (1 bit). El dato convertido se genera en los flancos de subida de la señal de control Phi1, permaneciendo estable todo el periodo. El instrumento deberá almacenar y leer los dos bits convertidos, para lo que se utilizará la señal Phi2 (flanco de subida o bajada, dado que en ambos la señal de salida del ADC es estable). El periodo de funcionamiento de este instrumento de adquisición será de 78.125  $\mu$ s. Las señales de control se generan sincronizadas y los datos leídos se recogen aproximadamente a mitad del periodo, cuando éstos están estables.

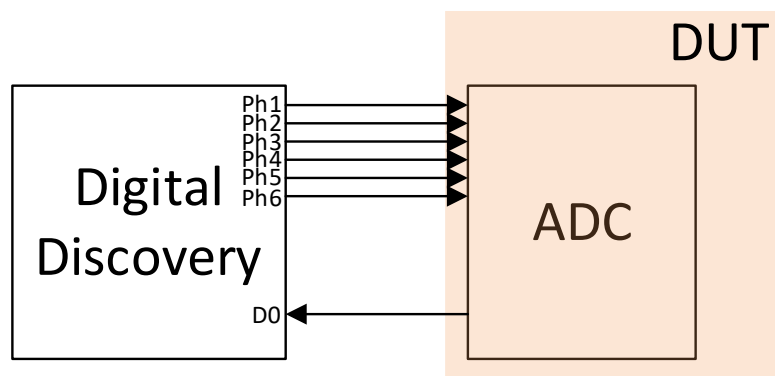


Figura 4-3. Esquema de conexión del ADC (DUT) con el Digital Discovery

El funcionamiento de este dispositivo es el siguiente en el programa desarrollado:

- Primero se inicializa el Digital Discovery.
- Se generan las 6 señales de control, mediante la función *pattern.generate*, pasándole como parámetros el canal por el que se va a transmitir esa señal, el tipo de función -en nuestro caso va a ser una función personalizada- y, por último, la tasa de generación de muestras. Es importante seleccionar el tipo de función personalizada ya que de esta forma el valor de las muestras se lee desde una posición de memoria, permitiendo al usuario usar formas de ondas propias. Asimismo, cabe indicar que se ha seleccionado como valor para la tasa de generación de muestras el valor máximo de 100 MHz.
- Las señales de control se almacenan en un fichero CSV para poder ser analizadas posteriormente y detectar cualquier posible problema o para facilitar la representación de las señales de control junto con los datos obtenidos.

Para la generación de las formas de onda, se ha creado una clase, llamada “phases”, en cuyo constructor se inicializan varios atributos, con valores iniciales asignados:

- El tiempo de muestreo.
- El periodo de la señal.
- La frecuencia de la señal, la cual se calcula a partir de la inversa del periodo de la señal.
- El delay.
- El ancho del pulso.
- El número de repeticiones.

- Una lista para almacenar la señal generada.
- Otra lista para almacenar la inversa de la señal generada.
- El número de muestras por periodo, el cual se calcularía a partir de la división entre el periodo de la señal y el tiempo de muestreo.

Posteriormente, dentro de la clase “phases” generada, se definen varias funciones o métodos, las cuales nos interesan para la hora de crear señales, y en cuya llamada se le irán pasando los atributos con las características de cada una de las fases a generar:

- Métodos para establecer y/o actualizar los parámetros de configuración:
  - Set\_generation\_period: este método establece el periodo de muestreo y recibe como argumentos:
    - El objeto
    - El periodo nuevo de configuración
  - Set\_signal\_period: este método muestra el periodo de la señal y recibe como argumentos:
    - El objeto
    - El periodo nuevo de la señal
  - Set\_delay: este método fija el retardo, haciendo también una comprobación para que este no sea mayor que el periodo de la señal. Recibe como argumentos:
    - El objeto
    - El nuevo delay
  - Set\_pulse\_width: este método determina el ancho del pulso y recibe como argumentos:
    - El objeto
    - El nuevo ancho del pulso
  - Set\_repetitions: este método establece el número de repeticiones y recibe como argumentos:
    - El objeto
    - El número de repeticiones
  - Set\_n\_sample: este método fija el número de muestras por periodo y recibe como argumentos:
    - El objeto
    - El número de muestras que se desea
  - Update\_n\_samp: este método establece el número de muestras actualizadas. Se actualiza el número de muestras por período basado en el período de muestreo y la señal. Recibe como argumentos únicamente el objeto.
- Generate\_pulse: método para la generación del pulso, donde se indicará la muestra inicial y la muestra final. Genera una señal de pulso y almacena los resultados en las listas signal y signal\_n. Recibe como argumentos el objeto y se puede incluir un gráfico si se establece el argumento plot en True.
- Save\_to\_csv: método para guardar las señales generadas en ficheros csv. Recibe como argumentos:
  - El objeto
  - El nombre del fichero csv donde se desea guardar la señal generada.

En la llamada a la función, vamos cargando los diferentes valores para cada una de las fases y los guardamos en ficheros CSV, para posteriormente cargar esos datos almacenados en la llamada a la función “pattern.generate”, lo cual será nuestra señal customizada, comentada anteriormente.

La función para generar el pulso en función de los valores que después le pasemos en el código principal, lo que hace es establecer un punto de partida y un punto de final, calculado a partir de la división entre el delay y el

tiempo de muestreo en el caso del punto de partida (será cero siempre) y a partir de la división de la suma del delay con el ancho del pulso entre el tiempo de muestreo para el punto de terminación de la generación. Después estableceremos los valores (1 o 0), en función de cada una de las fases y las almacenaremos en la lista definida anteriormente o en su negada.

A continuación, en la Figura 4-4, se muestra el diagrama de flujos del Digital Discovery:

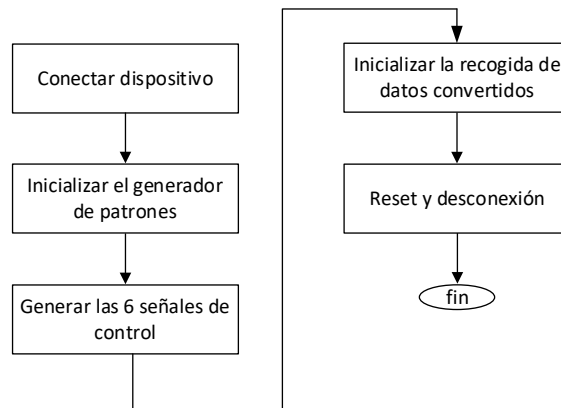


Figura 4-4. Diagrama de flujo del generador de patrones que integra el Digital Discovery

### 4.2.3 Configuración del generador de funciones

La gestión de este dispositivo en el programa desarrollado se muestra a continuación

- Apertura del dispositivo Visa y comprobación del ID para saber que el equipo es el correcto.
- Se definen las funciones que se generan a la entrada del ADC, así como las principales características de estas formas de onda: frecuencia, amplitud, offset y paso.
- Se configuran los dos canales con las funciones (se utilizan los dos canales del generador de función con un desfase de  $180^\circ$  entre ambos canales).
- Se generan dos tablas con los valores de frecuencia y amplitud a imponer en cada canal de salida del generador de función (la separación entre puntos en las tablas se corresponde con el paso definido o step).
- Se enciende el dispositivo y para cada frecuencia y amplitud incluidas en las tablas se imponen valores a la salida de los dos canales.
- Una vez completada la iteración, se apaga el dispositivo.

Dado que para la ejecución de cada prueba es necesario modificar los valores de amplitud y de frecuencia, el diagrama de flujos asociado a su configuración se recoge dentro del proceso de adquisición. Este se muestra en la Figura 4-6.

### 4.2.4 Configuración de la fuente de alimentación

La gestión de este instrumento en el programa desarrollado se describe en las siguientes líneas:

- Primero de todo, se realiza la configuración de la fuente de alimentación comentada en apartados anteriores mediante la apertura de una conexión TCP/IP. Para ello, se debe abrir un socket, que es una forma de establecer una comunicación bidireccional entre dos dispositivos o aplicaciones diferentes, con la siguiente información:
  - Dirección IP del equipo

- Puerto de comunicación
- Tiempo máximo de espera
- Otros valores relacionados con la operación de la fuente, como son la habilitación de operaciones de eco, de operación de reinicio y consultas a la identificación del dispositivo, por lo que esos valores los ponemos a 1 en la llamada a la función.
- A continuación, se configuran la tensión y la corriente de salida, en principio, 3.3 V y 0.5 A, así como las protecciones contra sobre tensión y sobre corriente.
- Tras ello, se leen los valores configurados para confirmar que los cambios se han efectuado correctamente y se establece el formato de los datos.
- Finalmente, se deja abierto la comunicación para poder

Este código es fundamental para asegurar un funcionamiento adecuado y seguro de la fuente de alimentación en una configuración experimental.

El diagrama de flujo que muestra el manejo de la fuente de alimentación se detalla en la Figura 4-5.

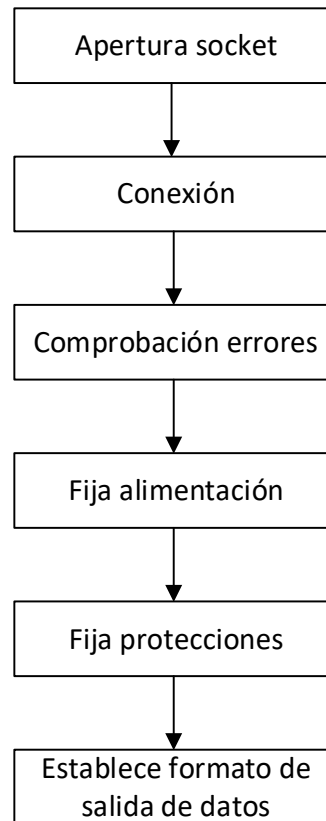


Figura 4-5. Diagrama de flujo de la fuente de alimentación

### 4.3 Módulo de adquisición

En esta sección, se van a describir los pasos necesarios a ejecutar para poder llevar a cabo la adquisición de la salida del DUT. Para ello, el procedimiento definido se describe a continuación:

- Primero, se habilita la fuente de alimentación.
- A continuación, se hará la habilitación del Generador de Formas de Onda Arbitrarias o AWG.

- Tras ello, se procede a configurar y habilitar el generador de formas de onda arbitrarias. En este caso, haciendo uso de la librería PyVISA, se procede a buscar el dispositivo AWG a través de su dirección VISA y, si se encuentra, establece una conexión para configurar y preparar el AWG. En caso contrario, se aborta la ejecución del programa devolviendo un error.
- Tras establecer la conexión, se realiza una consulta de identificación al AWG para obtener información sobre el dispositivo y se realiza un restablecimiento de su estado (comando "\*CLS") y un restablecimiento completo (comando "\*\*RST") para asegurarse de que el AWG esté en una configuración inicial conocida antes de su uso.
- Después, se pasará a la inicialización del analizador lógico utilizando la función `open`.

Tras la ejecución de estos pasos, se llega a la parte principal del módulo de adquisición. Esta se corresponde con un doble bucle *for* anidado donde se recorren las listas de la frecuencia y de amplitud generadas previamente. El objetivo de este bloque es configurar el AWG para ejecutar la prueba asociada a cada combinación de frecuencia y amplitud posible.

Dentro del bucle anidado, se llama a una función llamada *gen\_afg* con los siguientes argumentos:

1. Tipo de forma de onda ("SINE" en este caso). Además de la forma senoidal, se admiten los siguientes tipos:
  - SQUARE
  - RAMP
  - ARBITRARY
2. Valor de la amplitud
3. Valor de la frecuencia
4. Offset.

Esta función se encarga de enviar los parámetros al AWG para configurarlo con la forma de onda sinusoidal, la amplitud y la frecuencia especificadas.

Finalmente, se habilitarán dos canales en el analizador lógico para la lectura de resultados obtenidos por el DUT, del cual seleccionaremos uno u otro dependiendo de nuestras preferencias en cada momento, obteniendo así una lista con los diferentes resultados obtenidos.



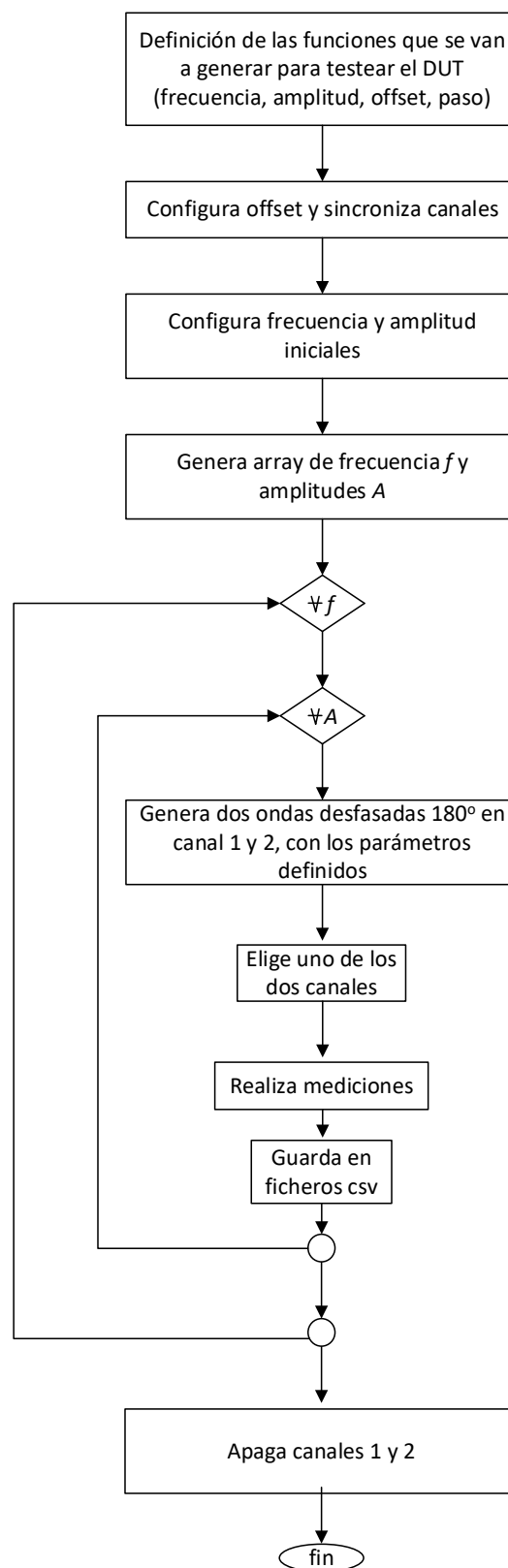


Figura 4-6. Diagrama de flujo asociado a la configuración del generador de funciones

## 4.4 Módulo de procesado

Por último, la parte de procesado consiste en el análisis de los datos obtenidos en las pruebas realizadas sobre el DUT.

En nuestro caso, se obtienen los armónicos, así como el cálculo de la potencia y el espectro de las diferentes señales obtenidas en cada una de las pruebas.

El código implementado para la parte de procesamiento de esos datos obtenidos consiste en lo siguiente:

- El código utiliza dos bucles anidados para iterar a través de diferentes frecuencias y amplitudes. Para cada combinación de frecuencia y amplitud, se genera una señal sinusoidal en el Generador de Formas de Onda (AWG). Esto se logra llamando a la función `gen_afg()` con los parámetros adecuados, como bien se ha comentado anteriormente.
- A partir de la obtención de los diferentes datos para amplitud y frecuencia, se calculan valores enteros `Nf_in`, `Nhd2`, `Nhd3` y `Nhd4`. Estos valores representan puntos en el dominio del tiempo que corresponden a la frecuencia fundamental y sus armónicos respectivos en la señal adquirida.
- Se inicializan listas y variables que se utilizarán para los cálculos posteriores, como potencia, relación señal-ruido (SNR) y armónicos. También se calculan índices y rangos para los ciclos y puntos en la señal.
- Calculamos la densidad espectral de potencia (PSD) de la señal `sd_out` utilizando la función `plt.psd()` de la librería `matplotlib`. La PSD y su versión en decibelios son agregadas a las listas correspondientes.
- Se preparan los datos para el cálculo de la relación señal-ruido (SNR) eliminando el ruido. Se toma una ventana alrededor de la frecuencia fundamental y se guarda la potencia de la señal en `Psignal`, mientras que el ruido se almacena en `P_noise`.
- Se calcula la relación señal-ruido-distorsión y se agrega a la lista `sndr`.
- Se realizan cálculos relacionados con la estimación de armónicos en la señal. Esto involucra calcular y guardar los valores de los armónicos fundamentales y sus amplitudes relativas.
- Se calcula la relación señal-ruido (SNR) para cada punto en la ventana de frecuencia fundamental y se agrega a la lista `snr`.

La Figura 4-7 muestra el diagrama de flujo asociado al módulo de procesamiento.

### 4.4.1 Herramienta de procesado

Todo lo anterior se ha hecho independiente del resto de módulos para sacar las prestaciones a partir de ficheros CSV que detallan el comportamiento del ADC diseñado en la Tesis Doctoral de Elena Cabrera. Dichos ficheros constan de 4 columnas de datos, separados por comas, que detallan el número de muestra (primera columna), la salida del ADC (segunda columna), el valor lógico del reloj (tercera columna) y el tiempo de muestreo (cuarta columna). La llamada al módulo de procesado se hace de la siguiente manera `sort_csv_file_list(csv_files)`, donde `csv_files` lee todos los ficheros CSV necesarios para el procesado. En el contexto del procesamiento de datos, se ha optado por utilizar como punto de partida los archivos CSV para realizar esta tarea de forma completamente independiente con respecto a los demás módulos, simplificando y clarificando el proceso en su conjunto.

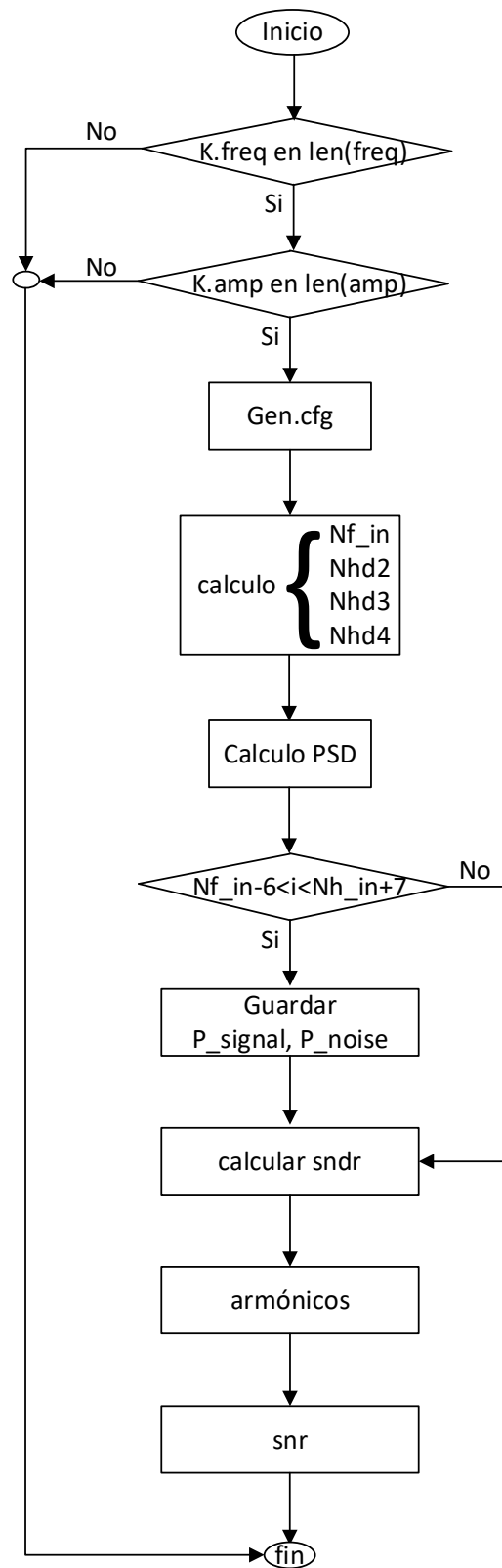


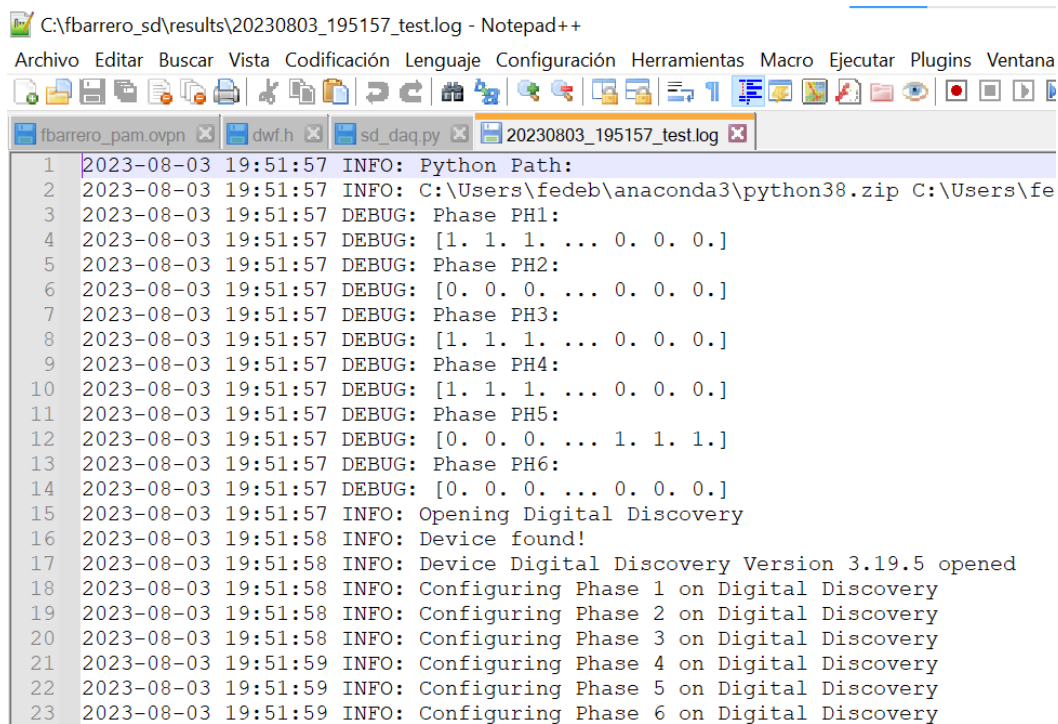
Figura 4-7. Diagrama de flujo asociado al módulo de procesamiento

## 4.5 Ejecución de la herramienta

Para configurar el registro de información de un programa (logging), hacemos lo siguiente en el código de Python desarrollado:

1. Primero se comprueba la cantidad de argumentos que se pasan por la línea de comandos utilizando `len(sys.argv)`. Si no se proporciona la ruta, el nombre del fichero log y el nivel de verbosidad (“verbose level”), sale del script.
2. Se obtiene la fecha y hora actual utilizando `datetime.now()`, y se crea un nombre de archivo de registro basado en la fecha.
3. Por último, se establece el nivel de verbosidad del registro. Los posibles niveles son:
  - 0: Sin registro (NOTSET)
  - 1: Depuración (DEBUG)
  - 2: Información (INFO)
  - 3: Advertencia (WARNING)
  - 4: Error (ERROR)
  - 5: Crítico (CRITICAL)

A continuación, se muestra un ejemplo de ejecución de dicho código:



```

C:\fbarrero_sd\results\20230803_195157_test.log - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana
fbarrero_pam.ovpn dwf.h sd_daq.py 20230803_195157_test.log
1 2023-08-03 19:51:57 INFO: Python Path:
2 2023-08-03 19:51:57 INFO: C:\Users\fedeb\anaconda3\python38.zip C:\Users\fe
3 2023-08-03 19:51:57 DEBUG: Phase PH1:
4 2023-08-03 19:51:57 DEBUG: [1. 1. 1. ... 0. 0. 0.]
5 2023-08-03 19:51:57 DEBUG: Phase PH2:
6 2023-08-03 19:51:57 DEBUG: [0. 0. 0. ... 0. 0. 0.]
7 2023-08-03 19:51:57 DEBUG: Phase PH3:
8 2023-08-03 19:51:57 DEBUG: [1. 1. 1. ... 0. 0. 0.]
9 2023-08-03 19:51:57 DEBUG: Phase PH4:
10 2023-08-03 19:51:57 DEBUG: [1. 1. 1. ... 0. 0. 0.]
11 2023-08-03 19:51:57 DEBUG: Phase PH5:
12 2023-08-03 19:51:57 DEBUG: [0. 0. 0. ... 1. 1. 1.]
13 2023-08-03 19:51:57 DEBUG: Phase PH6:
14 2023-08-03 19:51:57 DEBUG: [0. 0. 0. ... 0. 0. 0.]
15 2023-08-03 19:51:57 INFO: Opening Digital Discovery
16 2023-08-03 19:51:58 INFO: Device found!
17 2023-08-03 19:51:58 INFO: Device Digital Discovery Version 3.19.5 opened
18 2023-08-03 19:51:58 INFO: Configuring Phase 1 on Digital Discovery
19 2023-08-03 19:51:58 INFO: Configuring Phase 2 on Digital Discovery
20 2023-08-03 19:51:58 INFO: Configuring Phase 3 on Digital Discovery
21 2023-08-03 19:51:59 INFO: Configuring Phase 4 on Digital Discovery
22 2023-08-03 19:51:59 INFO: Configuring Phase 5 on Digital Discovery
23 2023-08-03 19:51:59 INFO: Configuring Phase 6 on Digital Discovery
  
```

Figura 4-8. Ejemplo de ejecución de logging

# 5 RESULTADOS

---

**E**n este capítulo se va a proceder a describir los diferentes resultados obtenidos durante el desarrollo del banco de pruebas automatizado para la caracterización de convertidores analógicos-digital.

En este apartado, es importante resaltar el hecho de que el sistema no se ha podido validar de forma real debido a un problema con la tarjeta impresa de medida, responsable de polarizar el circuito integrado bajo test. Tras diferentes pruebas en el laboratorio, no se consiguió obtener una respuesta a la salida del circuito integrado. Por tanto, debido a la imposibilidad de ejecutar el sistema completo de control, se ha realizado una prueba parcial de cada uno de los elementos para demostrar la validez del sistema.

Con este objetivo, las pruebas realizadas han sido:

1. Validación de la generación de las señales de temporización del ADC bajo prueba.
2. Validación del bloque de adquisición haciendo uso de un tren de pulsos
3. Validación del módulo de procesado haciendo uso de los ficheros CSV de las medidas originales del DUT bajo prueba.

## 5.1 Validación de las señales de control

Para llevar a cabo esta validación, se ha usado el esquema mostrado en la Figura 5-1. Los canales usados para la generación de las señales de control se corresponden con los números 24-29 mientras que la lectura de los canales se han usado las entradas D0-D5 debido a que estas se corresponden con señales de alta velocidad cuya frecuencia de muestreo es configurable hasta los 800MHz en modo 8 bits, velocidad suficiente para adquirir con precisión las señales de control.

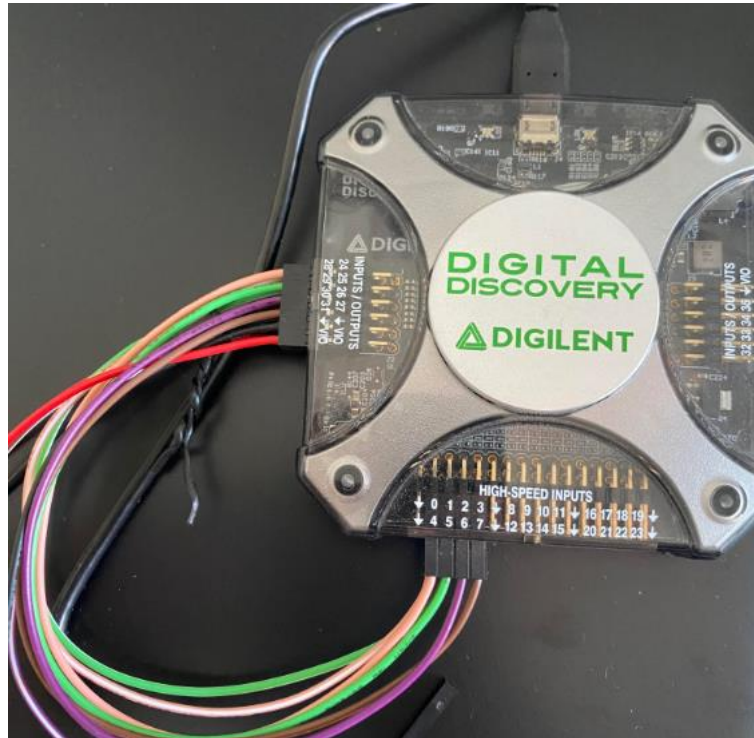


Figura 5-1. Esquema de medida para validar las señales de control

El resultado de la adquisición se muestra en la Figura 5-2, la Figura 5-3 y la Figura 5-4. Asimismo, se muestra una captura de pantalla de las señales para demostrar su correcto funcionamiento.

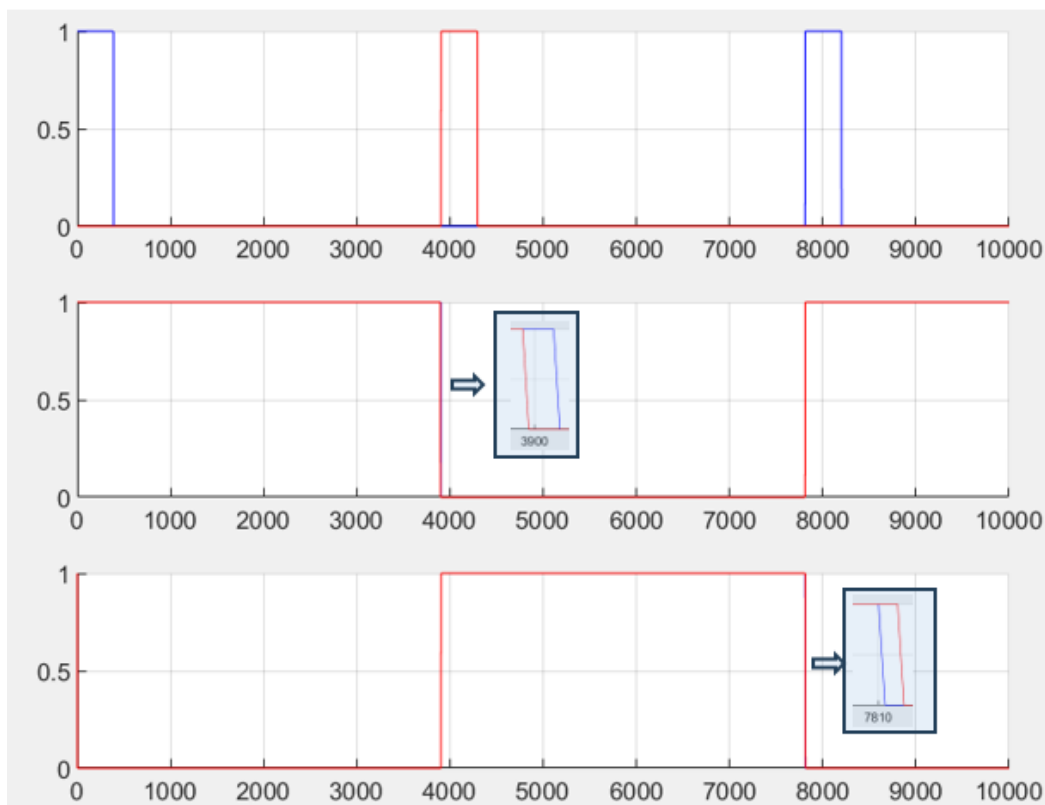


Figura 5-2. Temporización de las señales de control del ADC

Cada gráfica mostrada arriba, representa dos señales de control relacionadas, en azul y rojo. Ph1 (azul) y Ph2 (rojo) en la figura superior, Ph3 (azul) y Ph4 (rojo) en la figura intermedia y Ph5 (azul) y Ph6 (rojo) en la figura inferior.

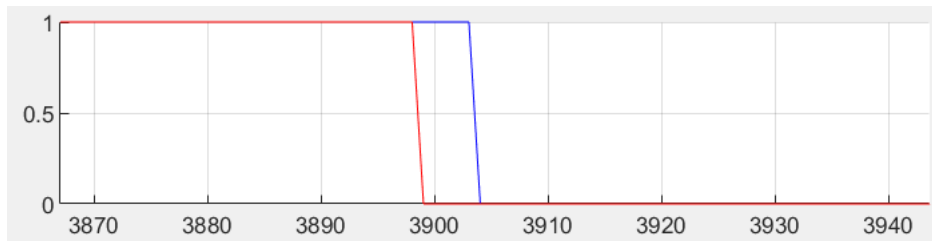


Figura 5-3. Detalle del desfase entre Ph3 (azul) y Ph4 (rojo)

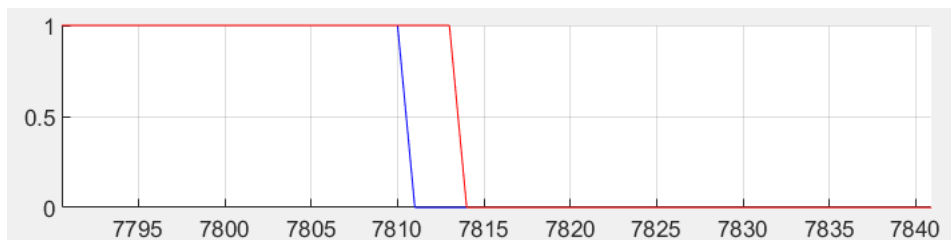


Figura 5-4. Detalle del retraso entre Ph5 (azul) y Ph6 (rojo)

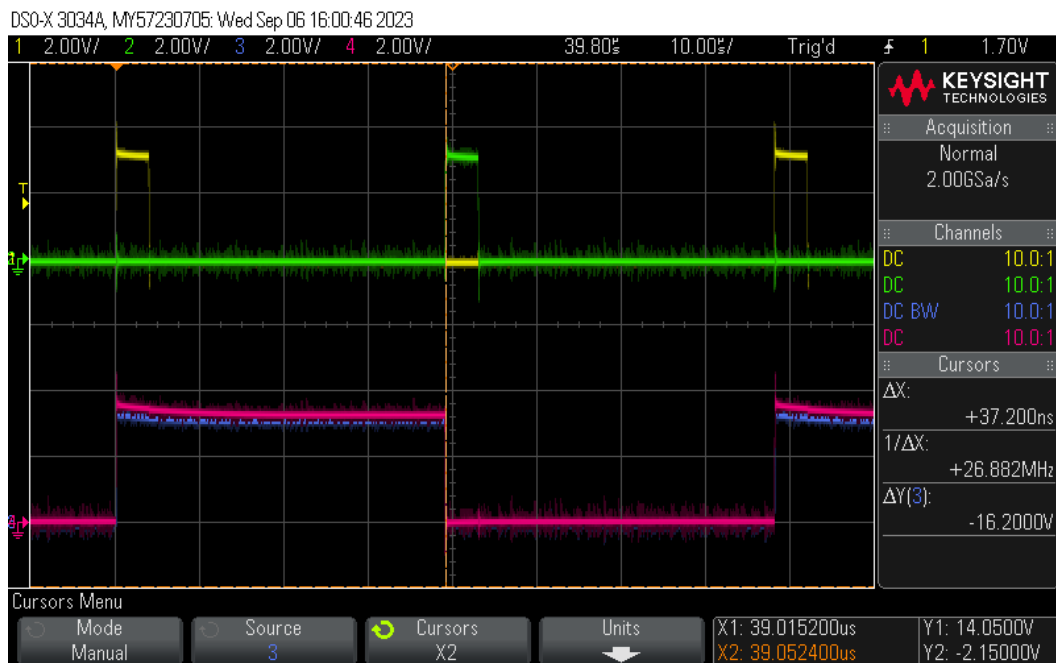


Figura 5-5. Captura de osciloscopio que muestra la temporización de las fases 1, 2, 3 y 4 generadas



Figura 5-6. Captura del osciloscopio aplicando zoom en el desfase entre Ph3 y Ph4

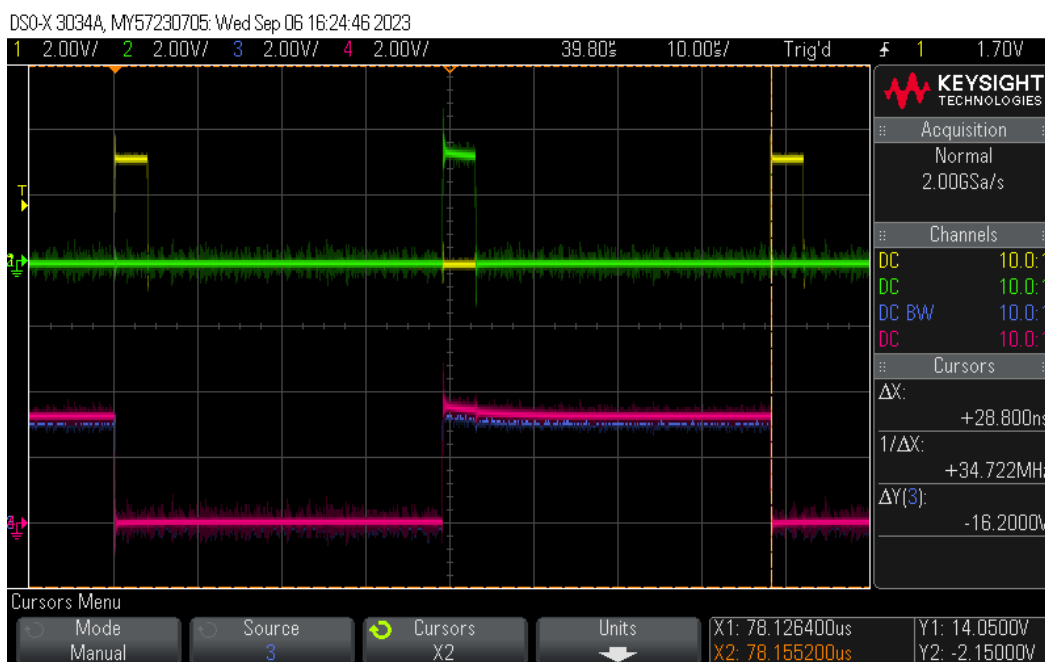


Figura 5-7. Captura de osciloscopio que muestra la temporización de las fases 1, 2, 5 y 6 generadas





Figura 5-8. Captura de osciloscopio aplicando zoom en la zona de subida de las fases 5 y 6



Figura 5-9. Captura de osciloscopio aplicando zoom en la zona de bajada de las fases 5 y 6

En la siguiente figura se muestra el banco de pruebas durante la obtención de las señales de control del ADC.

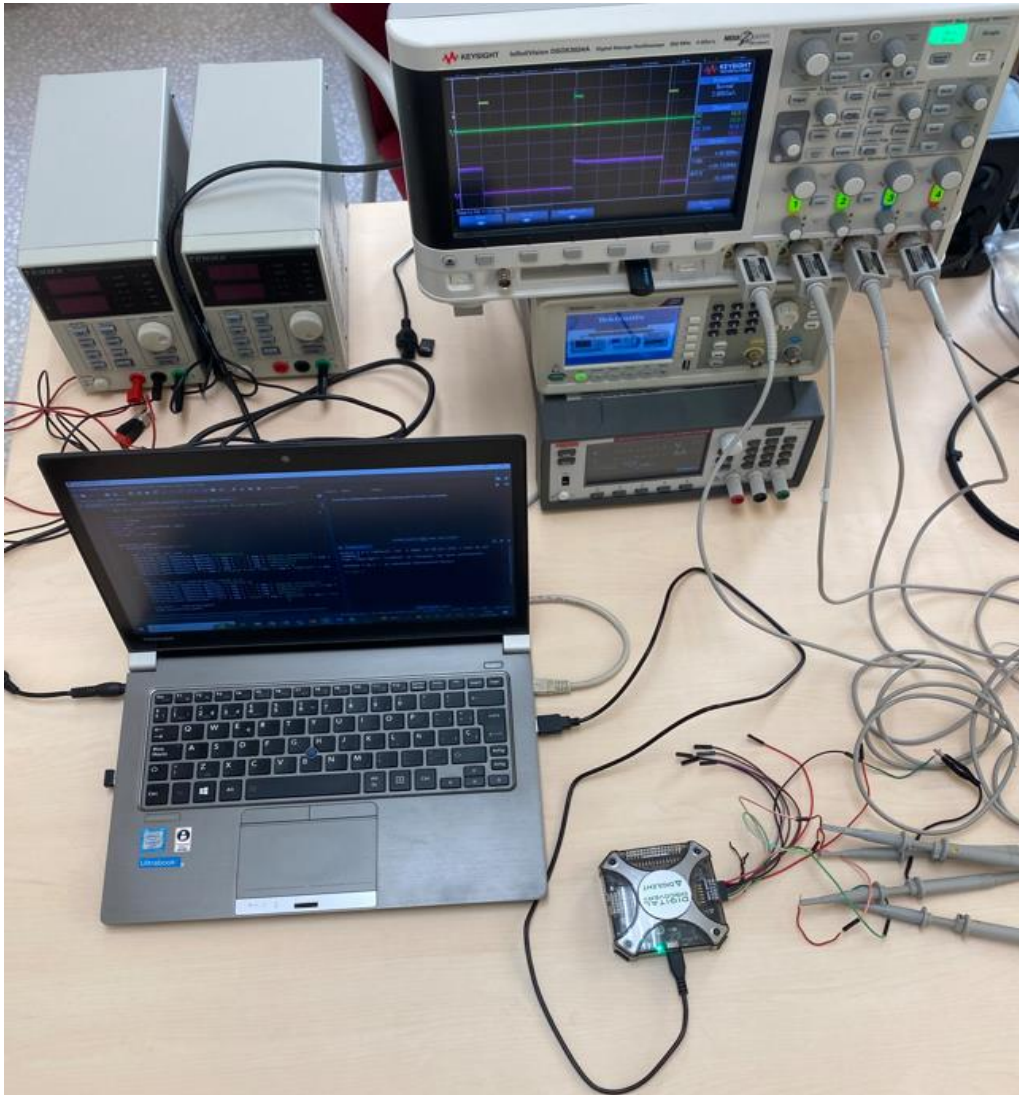


Figura 5-10. Banco de pruebas durante la obtención de las señales de control del ADC

## 5.2 Validación del procesado

La validación del sistema de procesado se realizará a partir de los archivos CSV generados durante la realización de la tesis doctoral [8] y presentados a la comunidad científica en el artículo de investigación [1]. El motivo de utilizar estos archivos se debe a que permitirá establecer una comparativa clara entre los dos esquemas de procesamiento, ya que los datos de entrada serán los mismos.

Por tanto, se introducirán en la herramienta designada para este propósito. Posteriormente, se procederá a comparar los resultados generados por la herramienta con los datos y resultados presentados en el documento de referencia. Esta comparativa se muestra en las siguientes figuras:

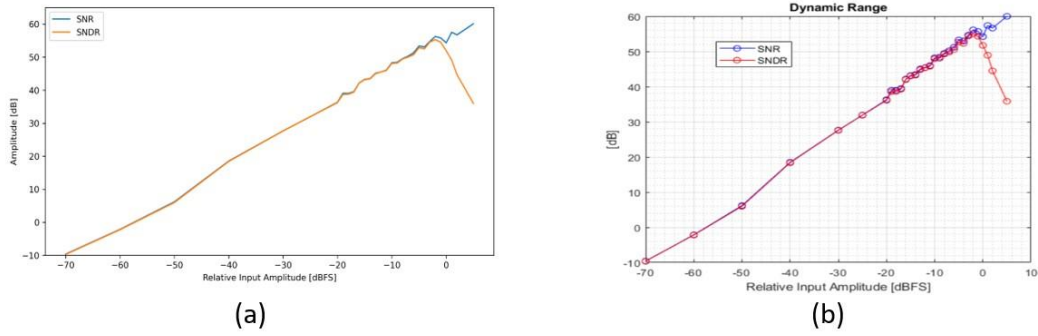


Figura 5-11. SNR y SNDR obtenida por (a) la herramienta desarrollada y (b) presentada en [8]

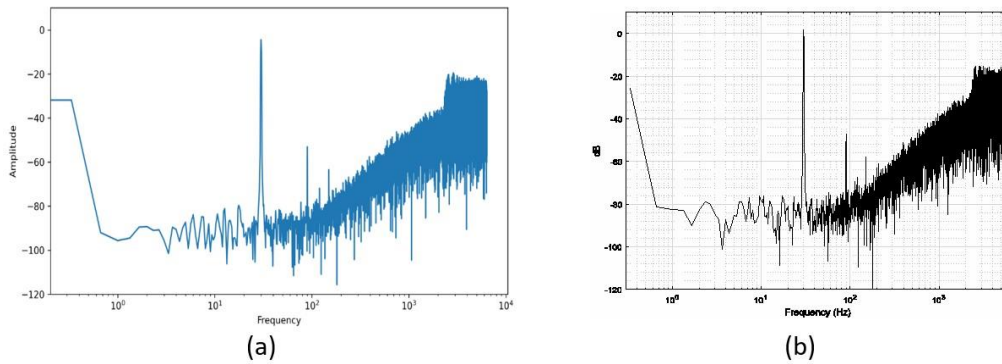


Figura 5-12. Espectro obtenido por (a) la herramienta desarrollada y (b) presentada en el trabajo [8] para una señal de entrada de 0 dBFS

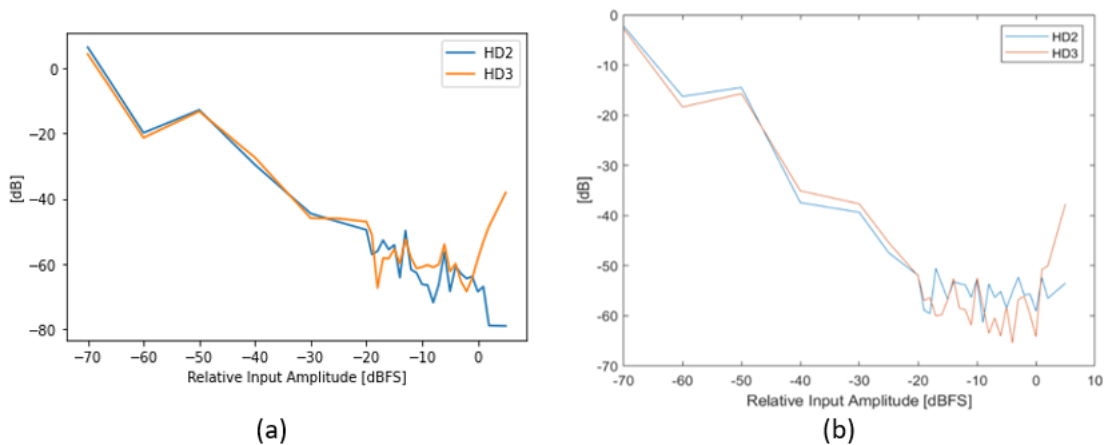


Figura 5-13. Comparativa de los armónicos obtenidos en la herramienta desarrollada (a), frente a los obtenidos en el trabajo [8]

Como se desprende de las figuras anteriores, los resultados obtenidos a través de este proceso de comparación coinciden de manera sustancial con los valores y hallazgos previamente reportados. En consecuencia, se puede afirmar que no existen discrepancias significativas y que los resultados obtenidos se encuentran en consonancia con las conclusiones previamente establecidas en el documento de referencia.

### 5.3 Validación del sistema de adquisición

Se procede a poner en funcionamiento el sistema, a través de la activación de sus componentes. Se realizan las mediciones correspondientes de acuerdo con los valores indicados en el documento de referencia [1]. Estas mediciones son sometidas a un proceso de comparación con los valores establecidos en dicho documento. Una vez completado este proceso comparativo, se confirma que los resultados son satisfactorios y no presentan inconvenientes significativos.

A continuación de muestra en banco de pruebas que se ha montado para la validación del trabajo desarrollado en este TFG.

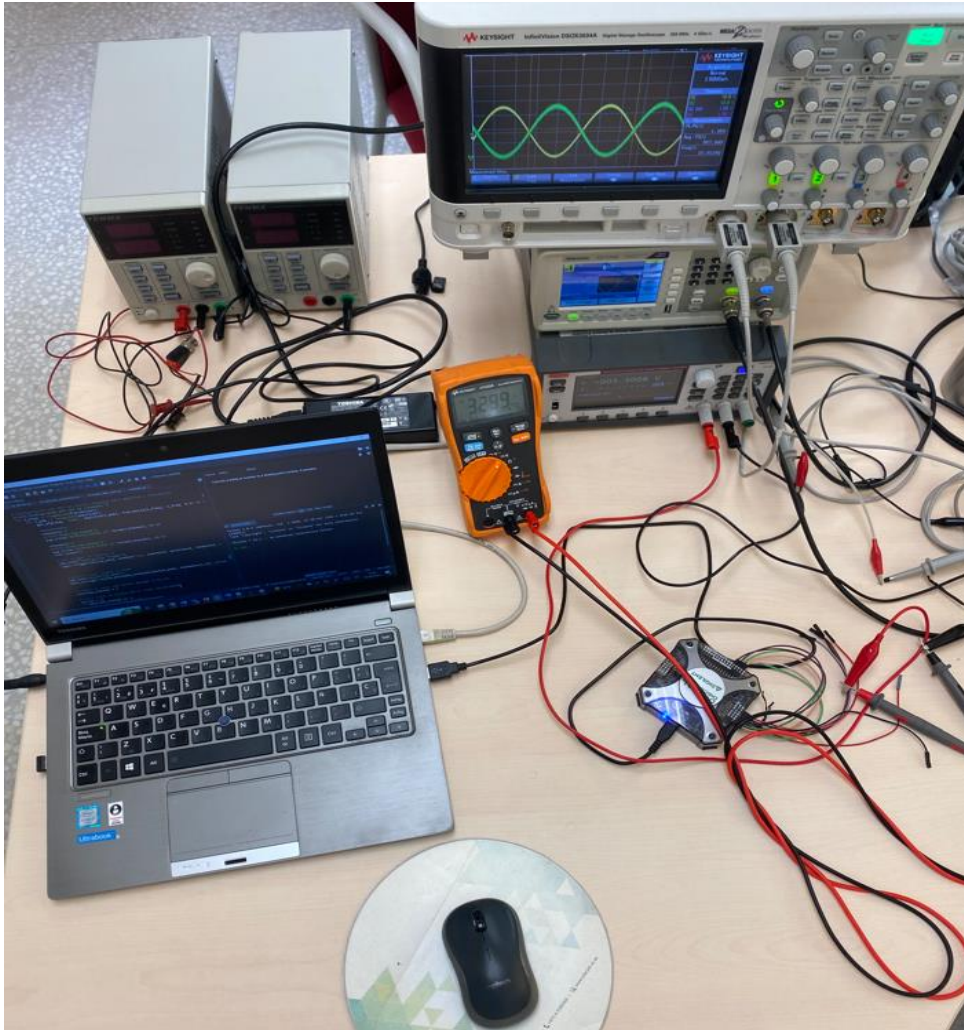


Figura 5-14. Banco de pruebas general

Dichos componentes se han controlado de manera remota como se puede observar en la siguiente figura, dónde la fuente de alimentación tiene activo el modo REMOTE.



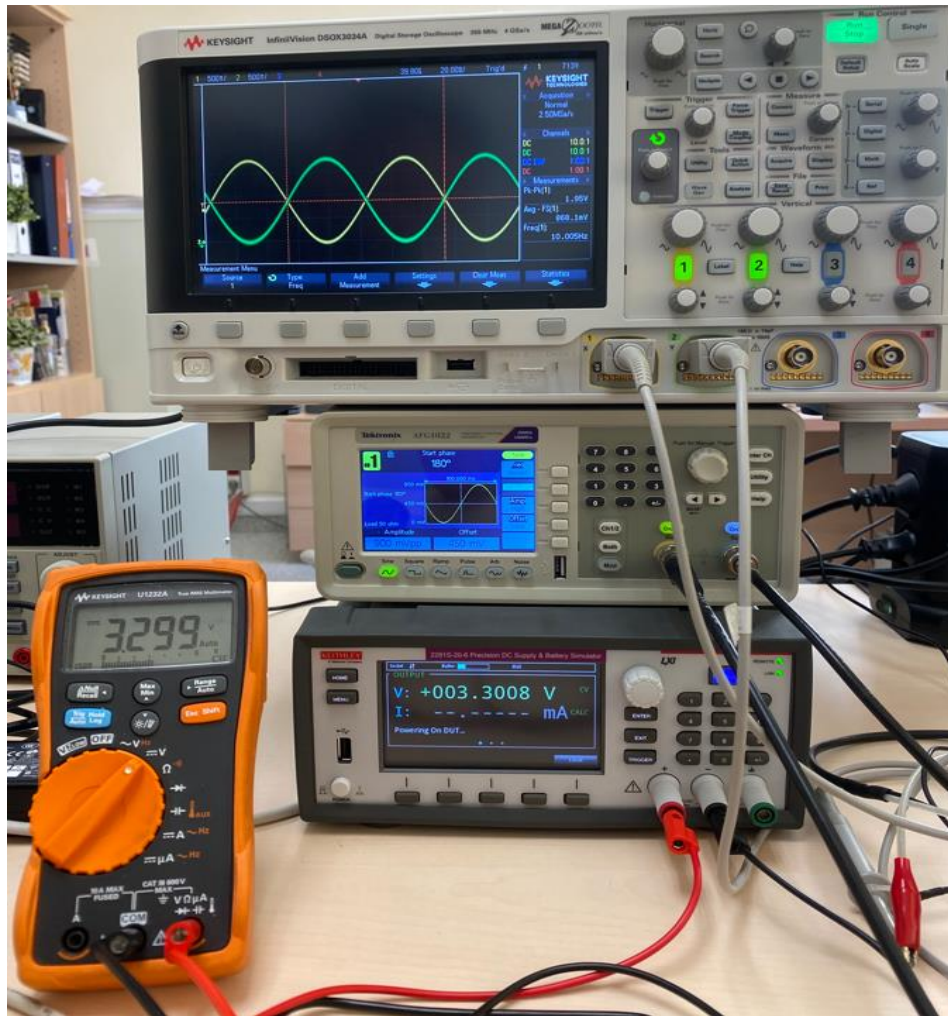


Figura 5-15. Banco de pruebas con los diferentes equipos controlados remotamente



# 6 CONCLUSIONES

---

Este trabajo fin de grado se ha centrado en el desarrollo de un sistema de caracterización automática de convertidores sigma-delta. El trabajo realizado ha cumplido con el objetivo de crear un sistema de test del dispositivo analizado, con la capacidad de obtención de sus parámetros característicos. Para ello, se ha desarrollado un banco de pruebas para la realización de los ensayos formada por tres instrumentos de medida o dispositivos: Digital Discovery (encargado de generar las señales de control, así como de registrar y almacenar las conversiones que genera), la fuente de alimentación Keithley-Tektronix serie 2281S (que alimenta al ADC con 3.3 V) y el generador de funciones Tektronix AFG1022 (que genera el patrón de señales analógicas a convertir por el DUT). Todos los dispositivos mencionados son controlados remotamente mediante un programa desarrollado en Python que utiliza el estándar PyVISA para su gestión. El programa desarrollado impone el patrón de señales de control del ADC mediante seis salidas digitales del Digital Discovery, procediendo a la lectura sincronizada del dato convertido, para lo que genera una señal adicional de sincronización o trigger. Los resultados obtenidos, que son almacenados en un fichero CSV, se procesan dentro del programa para obtener las características del ADC, almacenándose en el fichero CVS para su posterior revisión.

Con este Trabajo Fin de Grado se pretende complementar una línea de investigación desarrollada en el Departamento de Ingeniería Electrónica de la universidad de Sevilla centrada en el diseño y desarrollo de ADC de tipo Sigma Delta. Se espera que el desarrollo de este sistema de prueba automático contribuya a la caracterización de futuros diseños y contribuya a la internacionalización de los resultados obtenidos.

## 6.1 Líneas futuras

Se identifican diversas direcciones prometedoras para continuar la investigación en el marco de este Trabajo de Fin de Grado (TFG). Algunas de las líneas de trabajo principales son las siguientes:

- Ampliación de la Plataforma de Programación para el Dispositivo Digital Discovery de Digilent: Un paso adelante en este proyecto podría ser el desarrollo de un sistema de programación más versátil y adaptable para el dispositivo Digital Discovery de Digilent. Esto permitiría la integración y análisis de una gama más amplia de dispositivos ADC (Convertidores Analógico-Digital), en lugar de limitarse al dispositivo específico utilizado en este estudio.
- Diseño de una Interfaz Gráfica Intuitiva: Otra área de mejora sería la creación de una interfaz gráfica más sencilla de utilizar. Esta interfaz podría ser diseñada para configurar los equipos y dispositivos empleados en el análisis. Además, podría facilitar la integración de los resultados obtenidos con un software como Matlab, lo que simplificaría el procesamiento posterior de los datos y permitiría la verificación de los resultados generados en los archivos CSV.
- Automatización de Procesos: Una línea de trabajo interesante sería explorar formas de automatizar ciertos procesos en el análisis y evaluación de los datos. Esto podría incluir la implementación de algoritmos para la detección automática de características relevantes en las mediciones, lo que agilizaría el proceso de análisis y reduciría la carga manual.
- Exploración de Nuevas Aplicaciones: Se podría investigar la posibilidad de aplicar la metodología y herramientas desarrolladas en este proyecto a diferentes áreas o aplicaciones. Explorar cómo se podría adaptar este enfoque a otros tipos de análisis o mediciones podría abrir nuevas oportunidades de

investigación y desarrollo.

Cada una de estas líneas de trabajo podría aportar contribuciones significativas y enriquecer la comprensión en el área de estudio.



# REFERENCIAS

---

- [1] Elena Cabrera-Bernal, Fernando Muñoz, Antonio Torralba  $\uparrow$ , Clara Luján-Martínez, “Fully differential implementation of a delta-sigma modulator based on the pseudo-pseudo differential technique”, *Int. J. Electron. Commun. (AEÜ)*, Vol. 99, pp. 331–340, 2019.
  
- [2] <https://digilent.com/reference/test-and-measurement/digital-discovery/reference-manual>
  
- [3] <https://digilent.com/reference/test-and-measurement/digital-discovery/demos/start>
  
- [4] [https://download.tek.com/manual/071343001\\_2281\\_QSG\\_Feb\\_2016\\_EN\\_web.pdf](https://download.tek.com/manual/071343001_2281_QSG_Feb_2016_EN_web.pdf)
  
- [5] <https://github.com/tektronix/keithley/tree/main/Drivers/PS-228xS>
  
- [6] <https://www.tek.com/en/arbitrary-function-generator/afg1000-manual>
  
- [7] <https://github.com/topics/afg1022>
  
- [8] Elena Cabrera, “Performance enhancement in the desing of amplifier and amplifier-less circuits in modern CMOS technologies”, Tesis Doctoral, Julio de 2018. Disponible en abierto en el enlace <https://idus.us.es/handle/11441/77587> (repositorio de Tesis Doctorales de la Universidad de Sevilla).



