

1.2.- Sistemas de adquisición y Procesamiento de datos

1.2.1.- Proceso de adquisición de datos

Como paso previo, antes de realizar un tratamiento digital de la información está, el proceso de adquisición de los datos.

Ejemplo 1: Consideramos el caso de que queremos procesar a través de un computador una melodía procedente de una banda de música, antes de poder pasarla por el computador, necesitamos convertirla a una información compresible por él, es decir, a una secuencia de '0s' y '1s'.

Ejemplo 2: Supongamos que queremos tratar informáticamente la información recibida en una estación de radar, se reciben señales analógicas que es preciso convertir a informaciones digitales para su procesamiento.

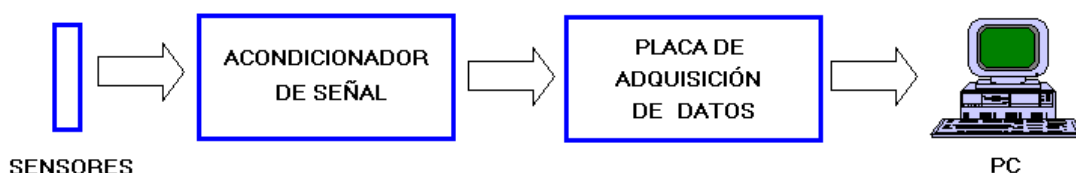


En definitiva, en este proceso se realiza una medición, a través de un computador, de una magnitud física.

El proceso a seguir con las señales desde la adquisición de estas, por medio del sensor adecuado, hasta la digitalización consta de tres etapas:

1. *Conversión* de la magnitud a una señal eléctrica
2. *Adaptación* de la señal eléctrica para su lectura digital
3. Sistema hardware de *adquisición* de datos, generalmente a través de un computador.

Proceso de adquisición de datos:



Esquema de Sistema de Adquisición de Datos

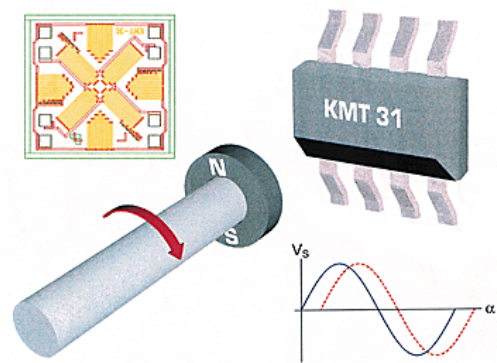
1.2.1.A.-Sensores

Una parte fundamental en todo sistema de adquisición de datos es el elemento encargado de percibir la magnitud a medir:

Definición: Los **sensores** son dispositivos capaces de convertir una magnitud física, como puede ser la temperatura, la presión, el valor de pH, etc., ... en una diferencia de potencial o una variación de intensidad. Es decir, realizan una *conversión de energías y suministran información sobre el estado y tamaño de la magnitud.*

Los sensores informan de su entorno y además esa información es cuantificable, es decir, medible por algún instrumento.

Ejemplo:



Sensor de campo magnético

La posibilidad de fabricar un sensor que perciba una determinada magnitud depende de:

1) Que **exista una propiedad en algún material que cambie en función de esa magnitud.**

Preferiblemente esa función debe ser lineal para el rango en el que estemos interesados.

Ejemplo: La relación que se da en los conductores entre su resistencia al paso de la corriente eléctrica y la temperatura.

Todos sabemos que a mayor resistencia, mayor temperatura:

$$R=f(T)$$

2) En otras ocasiones **existe una relación entre una magnitud y un fenómeno físico.**

Si la relación es predecible, suave y estable, el fenómeno se puede usar como base para la determinación de la magnitud.

Ejemplo: En el *efecto Hall* se genera una tensión cuando un material es atravesado por una corriente en presencia de un campo magnético. La tensión generada es proporcional a la corriente y al campo, de modo que si mantenemos la corriente constante podemos *sentir* y medir el campo magnético.

$$V=f(B)_{I=cte}$$

En muchas ocasiones la dificultad está en conseguir que la propiedad o fenómeno sea función únicamente de la magnitud que queremos evaluar. Suele ocurrir que la medida es función de varios factores, de tal manera que cualquier variación en alguno de ellos altera el resultado final.

En la actualidad, la mayor parte de los sensores:

- Generan una salida en tensión o corriente, o bien,
- Modifican una propiedad que puede ser evaluada de forma eléctrica.

De esta manera, y con el debido acondicionamiento, la señal de salida puede ser tratada por una equipo automático de adquisición de datos.

Las señales del mundo real son, en general, **analógicas** y varían de manera continua en el tiempo, para que un computador sea capaz de procesarla se debe convertir a **datos digitales**.

Cada uno de estos sensores tiene unas características propias y genera una tensión o intensidad determinada, por lo que estas señales tienen que ser adaptadas para ser tratadas en una tarjeta de adquisición de datos.

En el tratamiento de imagen y sonido, los sensores más utilizados son:

- **Micrófono:** Capta la información sonora que se propaga por el aire.



- **Cámara:** Capta la información visual.



1.2.1.B. Acondicionador de señal

El **objetivo** del acondicionador de señal es *generar, a partir de lo obtenido por los sensores, una señal que sea aceptable por las tarjetas de adquisición de datos*. Las tarjetas de adquisición de datos suelen admitir niveles de tensión que van entre unos márgenes determinados: -10V a 10V, 0 a 10V, 0 a 5V, etc., ...

Las funciones principales que va a tener que realizar el acondicionador de señal son las siguientes:

- Transformación
 - Amplificación
 - Conversión por medio de optoacopladores
 - Filtrado
 - Excitación
 - Linealización
-
- **Transformación:** Los sensores pueden proporcionar una diferencia de potencial, o una variable de intensidad. Normalmente las tarjetas de adquisición de datos admiten diferencias de potencial, por lo que si el sensor proporciona una variación de intensidad, esta debe ser convertida en una diferencia de potencial proporcional.
 - **Amplificación:** La señal proporcionada por los sensores suele ser de un valor muy pequeño, por lo que debe ser amplificada con el fin de que pueda ser detectada correctamente por la tarjeta de adquisición de datos. La amplificación debe ser tal que las variaciones de la señal recorran todo el margen de la tarjeta de adquisición de datos. La amplificación de las señales, en su origen, reduce el ruido que les puede afectar en su transmisión hasta el computador.
 - **Conversión por medio de optoacopladores:** Consiste en la conversión de una señal eléctrica en una señal óptica, de luz. El principal objetivo de esta conversión consiste en aislar los sistemas eléctricos de los sensores de los sistemas eléctricos de la tarjeta de adquisición para que de esta forma, se evite tener que usar masas comunes, que en algunos casos producen problemas de derivación de corrientes. Conviene que los sensores de calidad realicen esta conversión por medio de optoacopladores.
 - **Filtrado:** Con el filtrado se pretende eliminar ruidos de alta frecuencia que pueden hacer perder exactitud al sistema de adquisición de datos. Lo ideal es transportar la señal del sensor lo más limpia posible a la tarjeta de adquisición.
 - **Excitación:** Hay muchos sensores que necesitan de una excitación, bien en corriente, bien en tensión, para producir la variación proporcional a la magnitud a medir.
 - **Linealización:** No todos los sensores tienen una variación lineal con respecto a las variaciones de la magnitud que se miden; a veces es necesario realizar unos cálculos para convertir la respuesta del sensor en lineal.

1.2.1.C.- Tarjetas de adquisición de datos

El último paso en un sistema de adquisición de datos son las **tarjetas de adquisición de datos**, estas tarjetas se encargan de:

- Las *conversiones de señales desde analógica a digital. ADC.*
- *La comunicación con el ordenador.*

Características:

Como características más relevantes de una tarjeta de adquisición de datos están:

- Número de canales analógicos
- Velocidad de muestreo
- Resolución
- Rango de entrada.
- Capacidad de temporización
- Forma de comunicarse con el computador



- **Número de canales analógicos:** Nos indica la cantidad de magnitudes distintas que podemos adquirir con la misma tarjeta. Generalmente las tarjetas disponen de un único ADC y los diferentes canales se generan por medio de un multiplexor analógico.

Velocidad de muestreo: Cuanto mayor sea la velocidad de muestreo mejor representación obtendremos de la señal analógica, en cualquier caso la velocidad de muestreo debe ser siempre mayor que el doble de la frecuencia de la señal que queremos muestrear. /*Según el Teorema de Nyquist*/

Ejemplo: Si queremos digitalizar una señal de audio cuya frecuencia está comprendida entre 20 Hz y 20KHz, la mínima velocidad de muestreo que necesitamos es de 40.000 muestras por segundo. La reproducción de la señal obtenida, será de mayor fidelidad con velocidades de muestreo superiores.

La velocidad de muestreo depende de los canales que queramos tener activos. Las especificaciones de las tarjetas suelen indicar la velocidad de muestreo del ADC, a medida que aumenta el número de canales que este debe atender disminuirá el número de muestras por segundo que podemos obtener en cada canal.

$$V_{\text{muestreo por canal}} = \frac{V_{\text{muestreo ADC}}}{n^{\circ} \text{ canales}}$$

Resolución: Viene dada por el *número de bits del ADC que se utilizan para representar cada muestra*, a mayor número de bits del ADC la tarjeta será capaz de detectar variaciones menores en la señal. El número de distintos niveles en que se divide la señal a convertir viene dada por 2^n , siendo n la longitud de palabra del conversor.

Ejemplo: Un conversor de 8 bits tendrá 256 niveles distintos, suponiendo un rango de entrada de 10V proporcionaría una resolución de:

$$\frac{10 \text{ V}}{256} = 39\text{mV}$$

Es decir, será capaz de detectar variaciones de tensión de hasta **39mV** como mínimo.

Con las mismas condiciones en un conversor de 16 bits obtendríamos 65536 niveles y una resolución de:

$$\frac{10\text{V}}{65536} = 152\mu\text{V}$$

- **Rango de entrada**: Indica los márgenes entre los que debe estar la señal de entrada para que pueda ser convertida. Las tarjetas de adquisición de datos suelen dar varias posibilidades que se pueden seleccionar por hardware o por software.

Estas 4 características vienen a determinar la **capacidad** y la **precisión** de la tarjeta de adquisición:

- A mayor número de canales → Mayor capacidad.
- A mayor velocidad de muestreo → Mayor capacidad
- A mayor resolución → Mayor precisión
- A menor rango de entrada → Mayor precisión, ya que se con los mismos bits de resolución se tendrá que representar un menor rango.

- **Capacidad de temporización**: La capacidad de temporización interna en la propia tarjeta de adquisición de datos es una característica interesante en estos sistemas, ya que permite unas funcionalidades adicionales:

- Puede controlar los momentos en los que se debe leer una señal
- Identificar cuantas veces se ha producido un evento
- Generar formas de onda de acuerdo al reloj
- Etc., ...

Y de esta forma descarga de estas misiones al computador que podrá usar ese tiempo para otras ocupaciones. También proporciona una forma de trabajo en tiempo real en aquellos casos en los que el computador no puede atenderla debido a sobrecargas o a limitaciones en su sistema operativo.

- **Forma de comunicarse con el computador**: Su funcionamiento, como dispositivo periférico se puede realizar de dos formas:

- Mediante *entrada-salida por interrupción*, lo normal.
- Mediante *acceso directo a memoria (DMA)*. En aquellos casos en los que el flujo de datos puede ser elevado.

1.2.1.D.- Computador de Procesamiento

Dentro del ámbito industrial o científico, resulta común encontrar procesos que requieren un control automático e inteligente:

- El **control automático** será aquel que se realizará sin presencia humana
- El **control inteligente** será capaz de tomar decisiones en función del estado del proceso en cada momento.

Hasta hace pocos años, un sistema de control inteligente habría necesitado de un equipo especializado y caro. El desarrollo que han sufrido los ordenadores personales en los últimos años ha posibilitado soluciones de este tipo con un coste relativamente bajo.



En principio, un computador es una máquina de uso general cuya falta de especialización y baja velocidad lo hacían, antiguamente inadecuada para aplicaciones de tratamiento de imagen y sonido. Actualmente, con las nuevas familias de procesadores se alcanzan las velocidades de ejecución que se precisan para cubrir la mayor parte de las necesidades.

Por otra parte, un PC que pretende *relacionarse con el exterior* necesita de una vía de comunicación, un canal de transmisión y recepción de datos.

Las interfaces habituales entre el ordenador y el usuario son, como sabemos:

- El **teclado** para la entrada de datos
- La **pantalla** o impresora para la salida de datos.
- Otra posibilidad es la comunicación a través de un **módem** que utiliza la línea telefónica y que permite la transmisión de datos de un ordenador a otro.

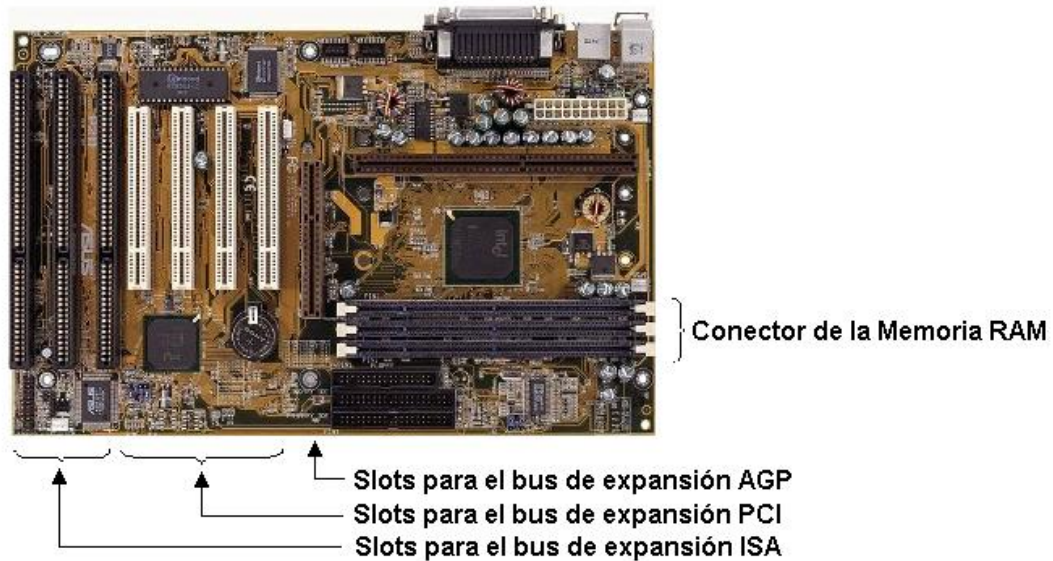
Esta escasa variedad de canales obliga a que se establezca, en la arquitectura de un computador, un acceso físico directo a su bus mediante algún mecanismo, sobre el cual pueden disponerse los sistemas de adquisición de datos: este mecanismo lo constituyen los **slots de expansión**, que no son más que ranuras sobre el bus de expansión del sistema, en las cuales se “*pinchan*” a modo de tarjetas los controladores de los periféricos y los sistemas de adquisición de datos. Este acceso resulta tener carácter general y se particulariza a cada caso concreto mediante el adecuado periférico.

Hay **slots** de distinto tipo dependiendo a qué bus de expansión accedan:

- Slots al bus ISA
- Slots al bus PCI
- Slots al bus AGP

El sistema así formado presenta dos importantes características:

- **Versatilidad:** Debido a la propia versatilidad del computador en tanto en cuanto es programable y de propósito general, evita el tener que adquirir equipos específico. En cuanto a la tarjeta, que al no ser un instrumento orientado a una aplicación específica, soporta también su propia programación.
- **Bajo precio:** Ya que la tarjeta, al no ser un dispositivo autónomo, no tiene que ir dotada de fuente de alimentación ni de controles externos.



Placa Base de un computador estándar

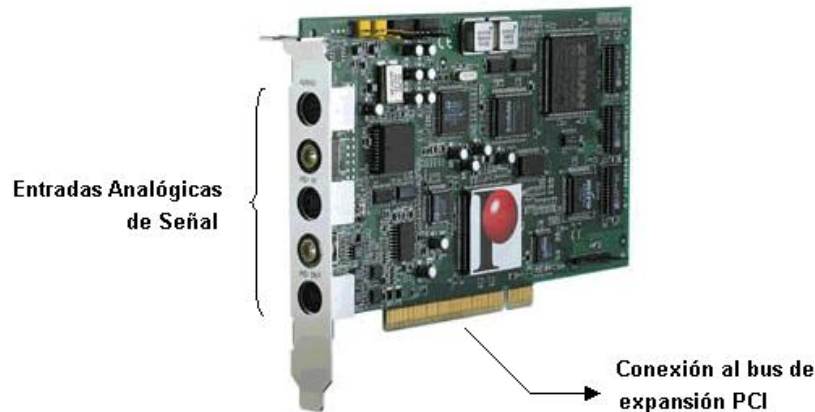
Como vemos, con la incorporación de tarjetas de adquisición de datos adecuadas, un computador de propósito general puede convertirse en una auténtica unidad de control adecuada para la mayor parte de las necesidades industriales y de procesamiento de imagen y sonido.



Además, la oferta actual de tarjetas de adquisición de datos presente en el mercado es muy amplia. Esto está motivado por el deseo de los distintos fabricantes de diseñar productos adecuados para usos o ámbitos de aplicación específicos. Las prestaciones y estructura de cada tarjeta pueden diferir significativamente de unas a otras.

Ejemplo: Para un control de temperatura no se requerirá una gran velocidad de muestreo, mientras que una tarjeta de sonido si, ya que al menos necesitará de 40.000 muestras por segundo (40 KHz).

La elección del modelo de tarjeta adecuado no ha de basarse en la búsqueda de las mejores prestaciones (que podría disparar su coste) sino en las prestaciones suficientes para cada aplicación. Lo mejor es adquirir exactamente lo que se necesite, ya que, como hemos visto, en caso de necesitar en un futuro mayores prestaciones, nos será fácil sustituir una tarjeta por otra y adaptarla al computador.



Tarjeta de expansión PCI

Incorporación de microcontroladores y DSP

La tendencia actual en los sistemas de adquisición y tratamiento de datos está en el diseño de sistemas capaces de hacer esa *adquisición y tratamiento de manera autónoma*, comunicándose con el computador central únicamente para la transmisión de bloques de datos elaborados, recibir órdenes de control o informar de situaciones particulares.

Es un paso más es el diseño de **sistemas distribuidos de adquisición y tratamiento**, dejando la mayor parte del trabajo a módulos independientes. En esta situación el computador actuará como un **coordinador** del sistema completo.

Estas tendencias exigen la inteligencia en los dispositivos y aumentar en cierto grado su capacidad de proceso, y para ello nada mejor que el uso de **microcontroladores y DSP**, que dotan al sistema de la potencia de los microprocesadores añadiendo, además, puertos de entrada-salida y el propio conversor analógico-digital incluido dentro del mismo chip del microcontrolador.

Por lo general todas estas tarjetas disponen de comunicación serie con el computador y la posibilidad de ser programadas en lenguajes de alto nivel.

1.3.- Fundamentos de procesamiento de señal

1.3.1.- Muestreo de señales Analógicas

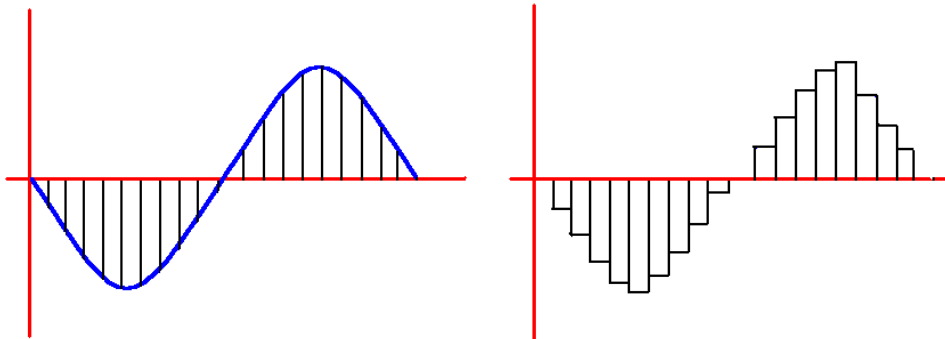
Para obtener datos digitales a partir de señales analógicas, la señal debe ser **muestreada**: Esto significa *tomar el valor instantáneo de la señal en un momento determinado*. Para una señal continua, las muestras se toman *a intervalos regulares*, generalmente con un periodo de muestreo fijo entre medidas.

Para recoger información útil, un factor clave es el **ritmo** o **frecuencia** con la que se toman las medidas.

En una aplicación de procesamiento de señal en la que tenemos que muestrear una señal continua **¿cómo sabemos qué frecuencia de muestreo debemos utilizar?** El teorema que define la mínima frecuencia requerida para representar de una manera precisa una señal analógica se denomina **Teorema de Nyquist**.

Teorema de Nyquist

El *Teorema de Nyquist* indica que la frecuencia de muestreo mínima que tenemos que utilizar debe ser mayor que $2 \cdot f_{\max}$, donde f_{\max} es la frecuencia máxima de la señal. Si utilizamos esa frecuencia de muestreo, podremos reproducir posteriormente la señal a partir de las muestras tomadas. La demostración matemática de este teorema se puede buscar en muchos libros de física.



Ejemplo:

Consideremos una onda senoidal con una frecuencia de **100 Hz**. ¿Con qué frecuencia necesitamos muestrear esta señal para poder representar exactamente la onda inicial? El Teorema de Nyquist indica que debemos utilizar una frecuencia de muestreo mayor que 200 Hz.

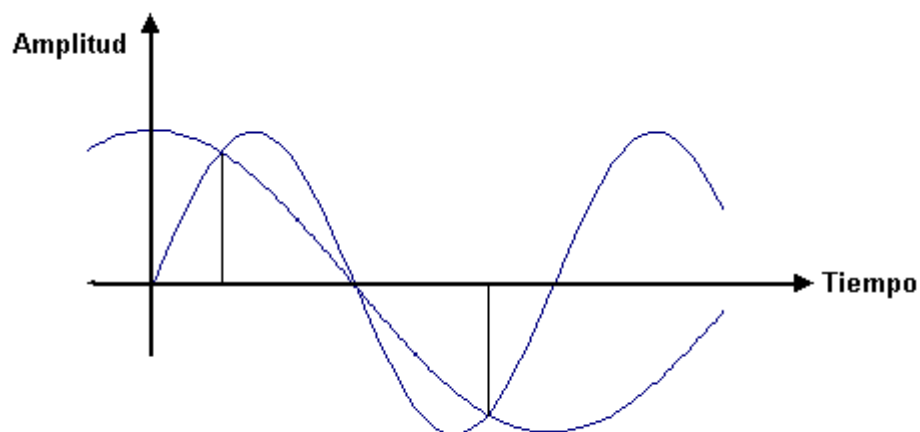
- **¿Qué sucede si utilizáramos una frecuencia de muestreo más alta?**

Si utilizáramos una frecuencia más alta que la que nos dice Nyquist obtendríamos una representación **más exacta** de la señal de entrada. Sin embargo, debemos buscar un compromiso: cuanto más alta es la frecuencia de

muestreo más alto es el número de muestras por segundo que se deberán procesar, por lo que los procesadores de señal deberán ser más potentes o realizar menos procesamiento en cada muestra.

- **¿Qué sucede si utilizamos una frecuencia por debajo de la especificada por el Teorema de Nyquist?**

Si utilizamos una frecuencia inferior aparece un fenómeno llamado **Aliasing**. Esto significa que cuando intentamos reconstruir la señal aparece una onda senoidal de frecuencia más baja. Este fenómeno se presenta por el hecho de que las muestras se pueden unir para crear una forma de onda de frecuencia más baja que es totalmente diferente de la onda original.



Efecto Aliasing

Ejemplo de la aplicación del Teorema de Nyquist:

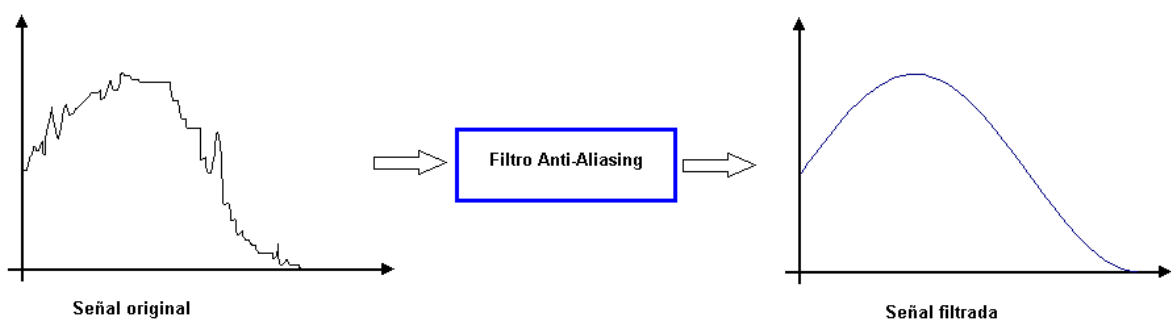
- En los **CD de música**, se almacena el sonido muestreado a 22,05 KHz. Esto es suficiente para cubrir todas las frecuencias perceptibles por el oído humano, y por tanto permite la reconstrucción completa del margen de audio útil.
- La **telefonía**, por otra parte, utiliza un ancho de banda más bajo alrededor de 4 KHz ya que el rango de frecuencias que se pueden transmitir por el cable telefónico es menor, y por tanto la frecuencia de muestreo también es menor, alrededor de 8 KHz.

1.3.2.- Proceso de obtención de la Señal Digital

El proceso de muestreo convierte una señal digital continua en una serie de valores digitales discretos.

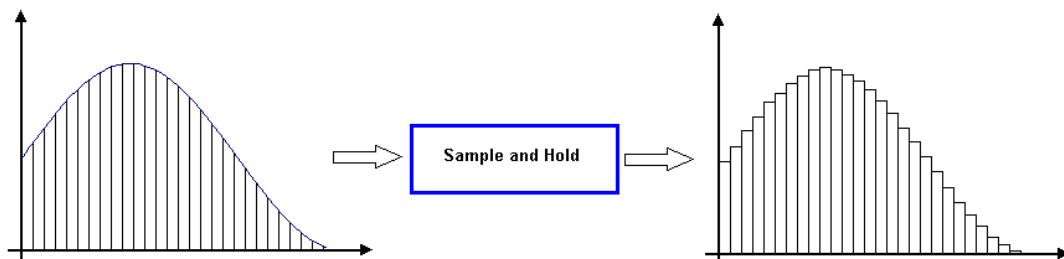
En primer lugar debemos eliminar primero las componentes de la señal por encima de la frecuencia de Nyquist ya que estas, no deben ser muestreadas. Esto se consigue usando un usando un **filtro anti-aliasing**. El filtro anti-aliasing no es más que un filtro paso bajo analógico.

1 Filtro Anti-Aliasing



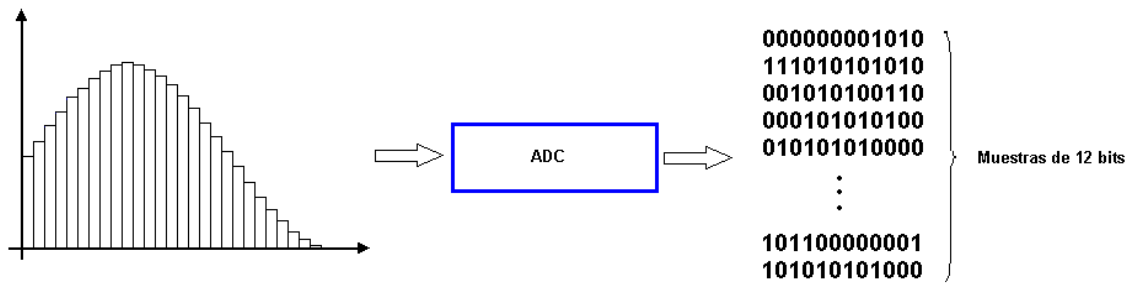
La señal de entrada continua se pasa entonces a un circuito *sample and hold*. Este circuito toma muestras a un ritmo fijo y mantiene el valor hasta que se toma la muestra siguiente, dando como resultado una forma de onda escalonada.

2 Sample and Hold

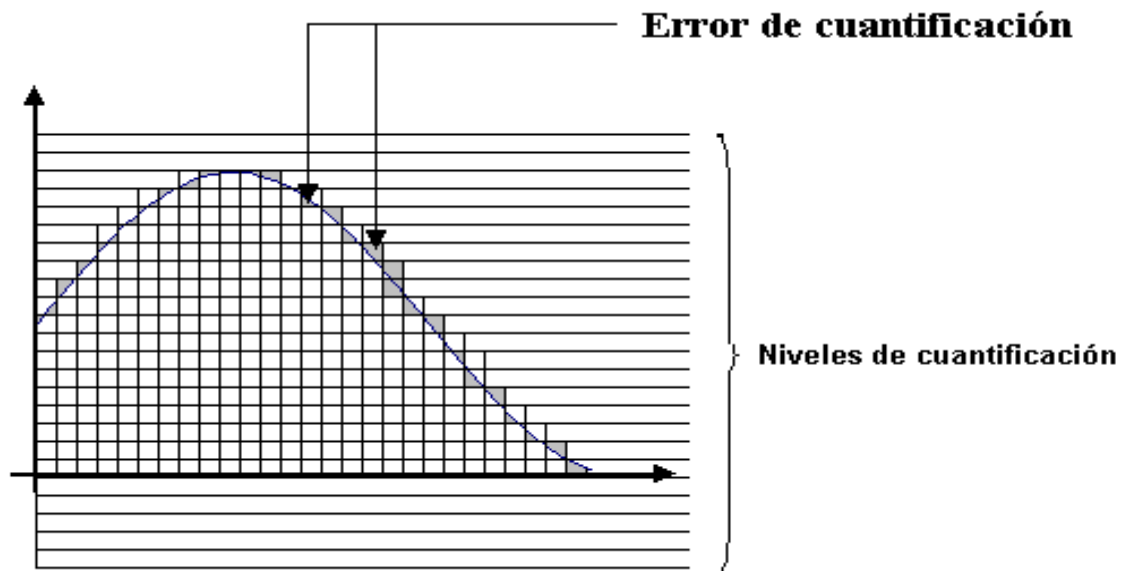


Los valores muestreados se deben convertir ahora en números que ya están listos para ser procesados. Este proceso se denomina **cuantificación**, y se realiza mediante un **convertor analógico/digital (ADC)**.

3 Cuantificación



Para cuantificar la señal, utilizamos un número de niveles que representan escalones de amplitud de entrada; para cada muestras ADC devuelve un valor que representa el nivel más cercano al valor de la muestra. Esto implica inevitablemente una aproximación llamada **Error de Cuantificación**.



Cuanto más niveles de cuantificación utilizemos, podremos representar más exactamente la señal analógica. Si en un sistema DSP, el convertidor analógico/digital (ADC) tiene una resolución de 12 dígitos binarios, en total nos encontraríamos con $2^{12}=4096$ niveles.

El número de bits usados depende generalmente de los requisitos de la aplicación. Por ejemplo, un CD de música utiliza 16 bits, mientras que el teléfono utiliza menos, típicamente entre 8 y 13 bits.

Si se asume que la señal de entrada es senoidal, podemos calcular la máxima **relación señal a ruido (SNR)** teóricamente posible con un número dado de bits, N, como sigue:

$$SNR \text{ (dB)} = 10 \log_{10} (1,5 \cdot 22N) = 1,76 + 6,02N$$

Ejemplo: Con un ADC de 12 bits, la máxima SNR posible de manera teórica, teniendo en cuenta el ruido de cuantificación, es de 74 dB. En un sistema real, habrá siempre un ruido adicional introducido por el propio sistema, por ejemplo errores de redondeo en los cálculos realizados por el procesador.

También se utiliza un **filtro analógico de reconstrucción** sobre la señal producida por el convertor digital/analógico (DAC) para eliminar las altas frecuencias de la señal, las cuales se deben a los niveles discretos en la señal reconstruida que son constantes en un periodo de muestra completo, y después cambian rápidamente entre los niveles discretos.

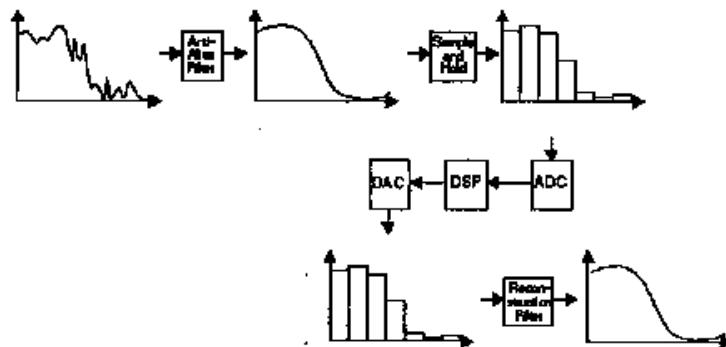


Diagrama de Bloques de un sistema DSP con filtros anti-aliasing y de reconstrucción

2.2.4.- Cuantificación no uniforme

Para mejorar la calidad de la señal digitalizada para un número fijo de bits por muestra, una técnica usada a menudo es la cuantificación no uniforme. Esto es muy común en aplicaciones de voz y audio. Un esquema uniforme de cuantificación, es un sistema según lo descrito arriba, donde el escalón entre cada valor digitalizado es el mismo. Sin embargo, para cuantificar los sonidos más suaves de manera precisa, se requiere más exactitud que los sonidos ruidosos. Para solucionar este problema podemos utilizar un esquema diferente de cuantificación, tal como intervalos logarítmicos, en donde el intervalo del escalón varía con la amplitud de la señal.

Para audio utilizamos niveles más espaciados en amplitudes más bajas para asegurarnos de que los sonidos suaves se representan más exactamente.

2.3.- ADCs y DACs

Para convertir una señal analógica en digital y viceversa, el sistema requiere convertidores analógico/digital y digital/analógico.

El ADC y el DAC cada vez con mayor frecuencia se pueden integrar en un solo dispositivo junto con los filtros necesarios antialiasing y de reconstrucción.

Hay dos tipos principales:

- Los chips de Interfaz Analógico (AIC).

Los chips Codificador/DECodificador (CODEC).

La diferencia principal entre estos dos tipos es que los AIC están disponibles para un rango más amplio de aplicaciones, mientras que los CODEC se orientan generalmente a las telecomunicaciones utilizando cuantificación logarítmica y un ancho de banda fijo.

2.3.1.- Conversores Analógico/Digital (ADC):

Hay diversos tipos de ADC disponibles actualmente, la mayoría de los cuales también contienen el circuito *sample and hold* requerido para digitalizar señales analógicas. Los cuatro tipos más populares son:

- **ADC de aproximación sucesiva:** Este tipo de ADC compara la forma de onda de la entrada de información con la salida de un DAC dentro del dispositivo. Repetidamente dividen el voltaje de entrada por la mitad produciendo una salida de n bits en n ciclos de su reloj. Los ADC de aproximación sucesiva son generalmente precisos y rápidos, y relativamente baratos de producir. Sin embargo, pueden ser lentos al responder a cambios repentinos en la señal de entrada y son sensibles a picos en la propia señal de entrada.
- **ADC de Doble Pendiente:** Los ADC de Doble pendiente utilizan un condensador conectado a un voltaje de referencia. Este se carga durante un tiempo determinado a través del voltaje de entrada y después se descarga a través del voltaje de referencia. El tiempo para descargarse se almacena utilizando un contador digital, y es proporcional al voltaje de entrada. Esto produce resultados muy precisos, mientras que la técnica de pendiente doble cancela la mayoría de las variaciones de la señal, pero es muy lento y generalmente costoso.
- **ADC Flash:** Este tipo de ADC compara el voltaje de entrada con un conjunto de voltajes de referencia creados por una escala de resistencias de igual valor. Se utilizan comparadores entre cada nivel de voltaje y la salida digital decodificada. Este proceso es muy rápido pero requiere 2^n resistencias exactas para un ADC de n bits. Esto hace el ADC flash caro, y comercialmente no práctico para utilizar una exactitud de 8 bits.
- **ADC Sigma Delta:** ADC Sigma Delta hace uso de técnicas de sobremuestreo y de reducción de ruido. A diferencia de los ADC convencionales, utilizan principalmente técnicas y filtros digitales. Esto significa que pueden ser implementados VLSI, y por tanto relativamente económicos, o incluso integrar sobre el mismo chip procesador de la señal. Estos ADC proporcionan una buena protección frente al ruido para resoluciones de hasta 20 bits, y son excelentes para las aplicaciones de audio.

Sobremuestreo: Significa muestrear la señal de entrada a un ritmo mayor dos veces la frecuencia de Nyquist. Esto permite que la especificación del filtro analógico anti-aliasing sea bastante más sencilla, reduciendo los requerimientos del circuito analógico. La señal sobremuestreada es entonces filtrada digitalmente paso bajo y después *diezmada* para obtener una serie de

muestras a dos veces la frecuencia de Nyquist. *Diezmar* quiere decir eliminar muestras y conservar solamente las muestras requeridas para la frecuencia mas baja. Por ejemplo si una frecuencia de sobremuestreo de 8 veces la frecuencia de Nyquist fuera utilizada, cada cuatro muestras sería una conservada cuando la señal fuese diezmada.

Como la señal sobremuestreada, el ruido de cuantificación se separa uniformemente a través del espectro hasta la frecuencia de muestreo, mientras que la señal se mantiene en el rango de frecuencias hasta la frecuencia de Nyquist. Esto significa que el nivel de ruido de cuantificación de la banda de frecuencia de interés es reducido, permitiendo más bits de resolución de los que el cuantificador proporciona realmente.

Más que cuantificar el valor absoluto de la muestra, el ADC Sigma Delta devuelve un solo bit de valor +1 o -1, dependiendo de si una muestra determinada es mayor o menor que la muestra anterior. El uso de un ADC de un solo bit evita tener que utilizar componentes tan precisos como los requeridos por otros tipos de ADC.

2.3.2.- Conversores Digital/Analógico (DAC)

Para reconstruir una señal analógica después de procesar la señal digital, se utiliza un conversor digital/analógico (DAC). Los DAC son generalmente mucho más baratos que el ADC usado en un sistema. Hay dos tipos principales de DAC, que son:

- **DAC Multiplicador:** El valor digital de la entrada se utiliza para activar la selección de fuentes de corrientes con los valores apropiados para cada dígito binario, por ejemplo **0,25 mA**, **0,5 mA**, **1 mA**, etc. Las fuentes de corrientes se suman y la corriente resultante se extrae directamente o se convierte aun voltaje de salida. Alternativamente, algunos DAC utilizan un conjunto de resistores escalados y una fuente de voltaje d referencia. Los resistores se conectan a tierra o a una Vcc dependiendo del valor del bit correspondiente de la señal, y se genera una salida proporcional a la suma de los voltajes usados.

Los DAC multiplicadores son el tipo usado más común, y tienen la ventaja de ser rápidos. La salida será generalmente una forma de onda escalonada que se suaviza con un filtro de reconstrucción para eliminar las altas frecuencias.

La desventaja de los DAC multiplicador es que el voltaje o la corriente generada para el bit más significativo (MSB) debe ser muy exacta, lo cual es extremadamente complicado en la práctica.

- **DAC Bitstream:** Un DAC bitstream utiliza el mismo concepto descrito anteriormente para el ADC sigma delta, con un número más pequeño de bits de salida para una frecuencia de muestreo mucho más alta. Esto proporciona ventajas similares al ADC sigma delta, con lo que el DAC bitstream puede ser implementado principalmente en el dominio digital.

3.- Diseño de sistemas de adquisición y distribución de señales

Para el diseño de un SADS hay que seguir los principios generales de diseño de todo sistema electrónico.

El diseño metódico de un sistema consta de cinco etapas:

Definición del problema
Diseño sistemático
Realización del diseño
Verificación y corrección
Documentación, mantenimiento y actualización.

3.1.- Definición del problema

La definición del problema implica:

Describirlo
Dar las características de un sistema que lo resuelva
Establecer las condiciones de contorno o ambientales en que deben mantenerse dichas características
Justificar el tipo de sistema propuesto para la solución, frente a otras posibles alternativas.

La **descripción del problema** debe incluir el análisis del contexto técnico, funcional y económico en que debe integrarse el sistema. Es decir, qué es lo que hay, cómo funciona, y cuanto cuesta. El diseñador del sistema no tiene porqué realizar dicho análisis, pero debe conocer sus resultados.

La **especificación de las características del SADS** debe seguir las normas técnicas habituales para el tipo de sistema. Debe incluir lo siguiente:

La descripción de qué debe hacer el sistema.
Las prestaciones técnicas de su función.
Los detalles de la interacción usuario-sistema.
La interfaz del sistema con el entorno.
La conducta en caso de errores y el diagnóstico de averías.

Las **condiciones de contorno o ambientales** pueden ser tanto climáticas (temperatura, humedad), como electromagnéticas (interferencias, fluctuaciones de la alimentación), químicas (atmósferas corrosivas, polvo) o mecánicas (vibraciones).

Ejemplo: Nos piden que desarrollemos un sistema de grabación digital en una unidad móvil (un coche) que está sujeto a movimiento y vibraciones, y en donde las características climatológicas dependen de donde esté situada esa unidad móvil: grabando un rally en el desierto, en la nieve, con viento, ...

3.2.- Diseño sistemático

Un diseño sistemático consiste en una serie de subsistemas modulares interconectados. Un método para definirlos es ir **subdividiendo la función a realizar en otras funciones**, cada una de las cuales se subdivide hasta llegar a funciones elementales de fácil realización.

La decisión de dividir o no una función viene condicionada por la disponibilidad de elementos simples para realizarla (sean hardware o software). Ello implica que, por una parte, hay que tener un buen conocimiento de la oferta, y por otra, saber contrapesar las ventajas y limitaciones respectivas al hardware y del software. Para ello hay que desarrollar el hábito de informarse, normalmente a través de la lectura.

En términos generales, las soluciones hardware son más caras pero también más rápidas de ejecución que las soluciones software. Algunos de los factores a tener en cuenta son:

Coste por unidad producida.

Tiempo y coste del desarrollo de la solución.

Fiabilidad o capacidad de funcionamiento dentro de las especificaciones durante largo tiempo tras la puesta en marcha.

Compatibilidad con otras aplicaciones dentro del entorno del sistema.

Potencia disipada.

Espacio ocupado.

Para grandes series de producción, cuanto menos hardware mejor, aunque sea a costa de requerir un software más complejo, porque mientras en una solución Hardware el coste de los componentes incorporados en cada producto se mantiene fijo, en cambio, en una solución Software el coste de la programación se reparte entre todas las unidades producidas y sólo el coste de la memoria necesaria en cada unidad repercute directamente en su precio.

Al estimar el coste hay que tener en cuenta la disponibilidad de los componentes necesarios; tareas como la codificación de teclas o interruptores, la decodificación para elementos de presentación alfanuméricos (displays, pantallas) y las interfaces serie/paralelo, son tan comunes que hay una amplia oferta hardware a bajo precio. Para los problemas en los que intervienen señales analógicas, las soluciones integradas disponibles son muy limitadas.

En las soluciones **software**, muchas veces se puede elegir entre aquellas que requieren tiempo y aquellas que requieren memoria o potencia. Por ejemplo, un resultado calculado mediante la aplicación de una fórmula requiere mucho tiempo y poca memoria, el mismo resultado obtenido a partir de una tabla requiere poco tiempo pero mucha memoria. Incluso dentro de una solución software es posible, a veces, elegir procesadores con distintas velocidades, seleccionar pues aquel que mejor se ajuste a nuestras necesidades.

En términos generales, una solución hardware es eléctricamente menos fiable que una solución software debido al mayor número de conexiones. Además, consume mayor potencia y ocupa más espacio. Además, por software se pueden realizar varias tareas, mientras que el hardware es muy rígido.

3.3.- Realización del diseño

La realización del sistema incluye en general dos partes: hardware y software. Las tareas a realizar en cada caso dependen de la complejidad del sistema y del nivel de diseño considerado.

En el hardware digital lo más práctico es emplear como componentes periféricos al microprocesador (Tarjetas). Si el diseño consiste en aplicar tarjetas de circuito impreso, deben ser todas compatibles con el bus central del sistema. Conviene que el número de tensiones de alimentación distintas sea el mínimo. La disposición de los elementos del sistema, sean los componentes de un circuito impreso, sean las tarjetas dentro de un chasis modular, hay que hacerla teniendo en cuenta la compatibilidad electromagnética entre los elementos.

En el diseño del software la decisión central es qué tipo de lenguaje utilizar. La programación de lenguaje de alto nivel (BASIC, FORTRAN, PASCAL) o intermedio (Lenguaje C) es más rápida y fácil de corregir y mantener pero exige que se disponga de un compilador, que es el traductor capaz de convertir las instrucciones en órdenes inteligibles por el microprocesador utilizado. Los programas en lenguajes de alto nivel son además más fiables por cuanto incluyen muchas menos instrucciones que los programas en ensamblador.

Un inconveniente de los lenguajes de alto nivel es que son menos eficientes por lo que el código máquina generado a partir de ellos suele ser mucho más largo que el generado a partir de un programa escrito en ensamblador, de modo que necesitan menos memoria. Peor es, sin embargo, el hecho de que en muchas aplicaciones el objetivo es sacar provecho de detalles específicos de la arquitectura de un determinado microprocesador, o controlador, y ello no es posible con lenguajes de alto nivel. Por ello la mejor solución suele ser utilizar una combinación de lenguajes de alto nivel con lenguaje ensamblador. Los sistemas de desarrollo de software actuales permiten combinar ambos tipos de lenguaje.

3.4. Verificación y corrección del diseño

La verificación y la corrección del sistema consiste en comprobar si el sistema cumple con las especificaciones propuestas y tomar las acciones necesarias para que las cumpla si no lo hace. Normalmente se recurre a analizar la salida ante unas condiciones predeterminadas; caso de que no cumpla con lo previsto se va retrocediendo desde el punto en el que se detecta el error hasta encontrar su causa. Si el sistema está concebido de forma modular conviene verificar cada módulo por separado una vez está completo, antes de verificar el sistema en su conjunto.

3.5.- Documentación, mantenimiento y actualización

Esta etapa se olvida con frecuencia. La ausencia de componentes mecánicos induce a pensar que si el sistema funciona bien al principio, lo seguirá haciendo de forma indefinida. La experiencia demuestra que no es así. Una razón que explica este hecho es que el entorno de funcionamiento del sistema puede ser distinto al entorno en el que se verifica, o puede cambiar con el tiempo. Puede suceder también que las

pruebas del sistema no incluyeran una combinación de entradas que luego se produce en la práctica, o simplemente que se exige al sistema unas funciones que difieren de las previstas inicialmente, por lo que su diseño debe ser actualizado.

Estas y otras circunstancias aconsejan documentar muy bien tanto el hardware como el software de la solución adoptada. Esta etapa debe correr a cargo de quien diseña el sistema. La documentación debe estar dirigida y ser inteligible a un ingeniero que no haya tomado parte en el diseño, pues normalmente esta es la situación real.