

1. ADQUISICIÓN DE DATOS

En el mundo hay diversidad de señales físicas que encontramos naturalmente (temperatura, presión, humedad, etc.). La dificultad se encuentra en que se necesita una forma de capturar electrónicamente esa señal y procesarla digitalmente para poder analizarla en una computadora o cualquier otro dispositivo. Para eso, se requiere una serie de dispositivos interconectados que se ven de esta manera:

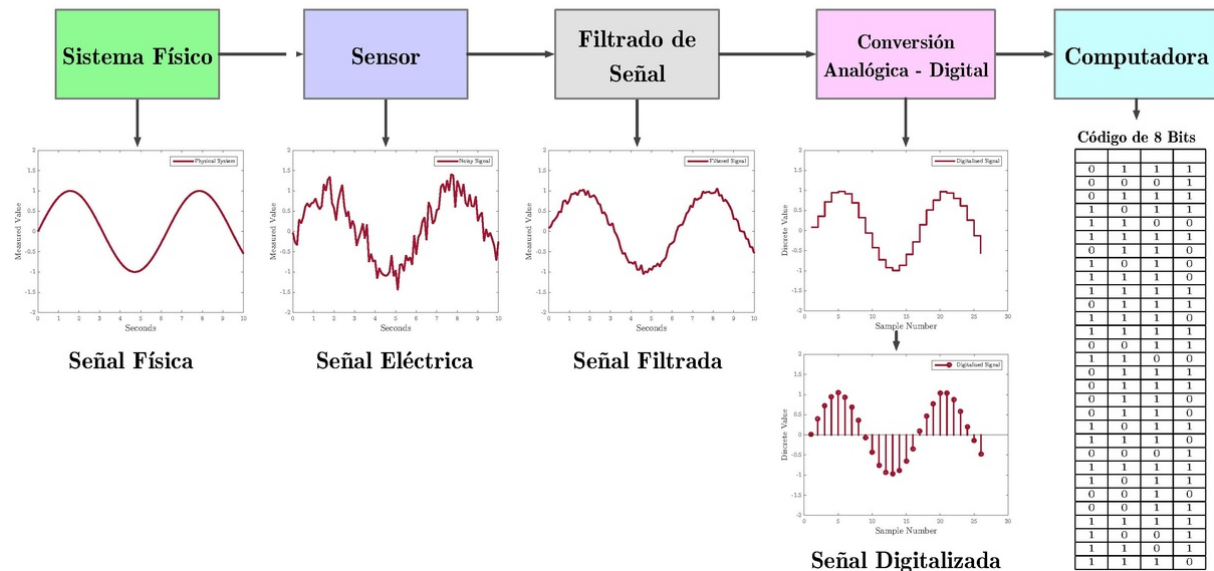


Figura 1: Sistema de adquisición de datos

En este diagrama encontramos:

1. La *señal física* que consiste en cualquier magnitud que se encuentre en la naturaleza y que se desee medir.
2. Un *sensor* que no es más que un dispositivo sensible a la magnitud que interesa medir y es capaz de transformarla a una señal eléctrica que es función de la física.
3. Un *filtro* que elimine cualquier ruido eléctrico. Dependerá de la característica del sensor usado.
4. Un *ADC* o *Conversor Analógico a Digital* que se encarga de capturar periódicamente la señal y obtener valores discretos con una resolución acorde a la cantidad de bits del ADC.
5. Una computadora o cualquier dispositivo para mostrar o graficar los valores obtenidos.

Algunos sensores incorporan el filtro y el ADC, por lo que la adquisición simplemente consiste en solicitarle al sensor los datos digitalizados que obtuvo a través de algún protocolo de comunicación serial que suele ser SPI, I2C o CAN.

1.1 ANALOGICO VS DIGITAL

Una señal analógica se define como una señal continua que posee infinitos valores. En cambio, una señal digital consiste en una señal discontinua con saltos discretos o definidos.

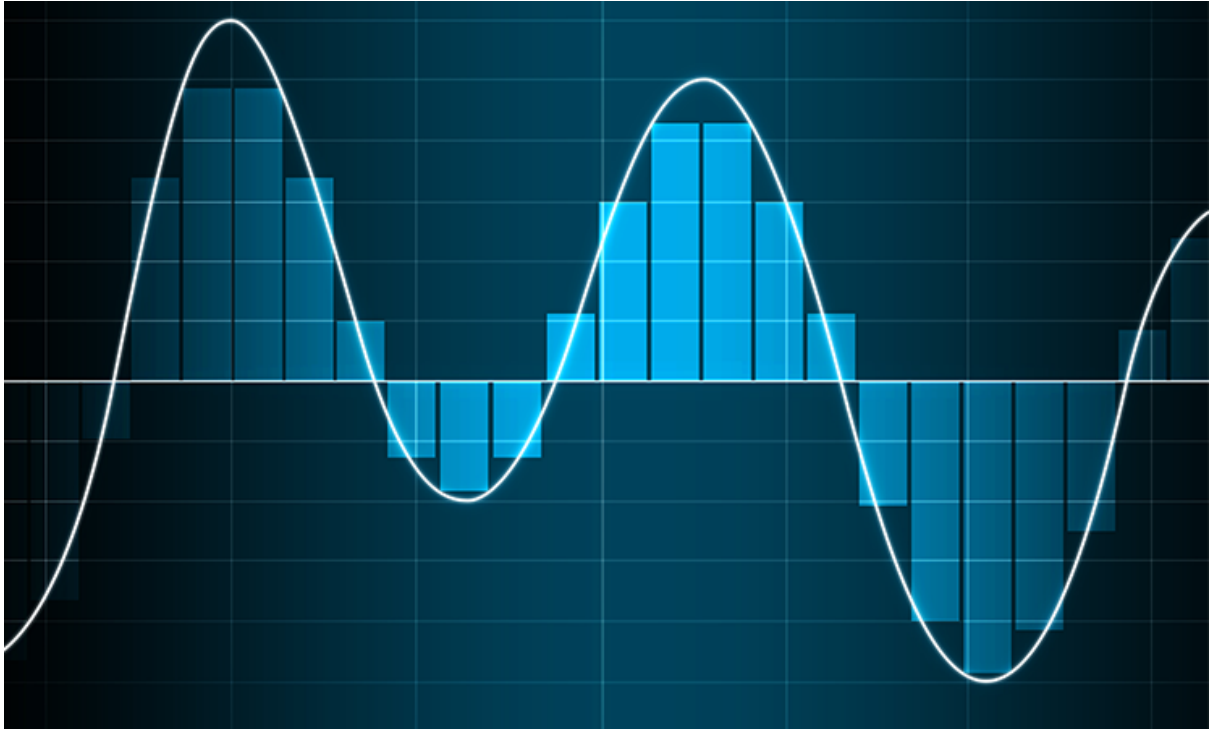


Figura 2: Señal analógica vs señal digital

Se puede ver que, si bien la señal analógica es continua y tiene curvatura suave, la señal digitalizada tiene valores cuantificados, es decir, valores específicos a los que saltar, no tiene valores intermedios. Esto puede mejorarse si se aumenta la resolución del conversor, dando más valores posibles.

Debido a que una señal digital tiene valores discretos, nunca es capaz de reproducir exactamente la señal original, sino que hay un error que se está cometiendo cada vez que se digitaliza.

1.2 ANALOG TO DIGITAL CONVERTER (ADC)

Una tecnología que permite hacer adquisición de datos es un ADC. Ya sea que se consiga de forma integrada o como periférico de un MCU, son los responsables de digitalizar cualquier señal analógica. Hay distintos tipos de ADC que tienen filosofías de funcionamiento diferentes, cada uno de ellos con sus particularidades. Si bien no es el objetivo estudiarlos todos, si vale la pena mencionarlos y resaltar algunas particularidades.

Tipo	Ventajas	Desventajas	Máxima resolución	Máxima frecuencia de muestreo	Aplicaciones

Aproximaciones sucesivas (SAR)	Buena relación velocidad / resolución	No tiene protección anti-aliasing	18 bits	10 MHz	Adquisición de datos
Sigma delta ($\Delta\Sigma$)	Buena performance dinámica, protección anti-aliasing	Histéresis en señales no naturales	32 bits	1 MHz	Adquisición de datos, audio
Doble pendiente	Precisos, baratos	Baja velocidad	20 bits	100 Hz	Voltímetros
Pipelined	Muy rápidos	Resolución limitada	16 bits	1 GHz	Osciloscopios
Flash	El más rápido	Baja resolución	12 bits	10 GHz	Osciloscopios

Tabla 1: Tipos de ADC

Como mayormente nos enfocaremos en usar ADC del tipo SAR, en adelante siempre se hablara de ese tipo, ya que es el más común de encontrar en un MCU.

1.2.1 CONVERSIÓN POR APROXIMACIONES SUCESIVAS (SAR)

Al trabajar con este tipo de conversores, dos parámetros son importantes para usar el resultado obtenido: la resolución y la tensión de referencia.

La resolución es la cantidad de bits de resultado que tiene el ADC. Esto permite obtener como resultado de la conversión un número entre 0 y $2^n - 1$ donde n es la cantidad de bits (ejemplo, para 12 bits, los valores resultantes posibles son 0 a 4095). A mayor cantidad de bits, menor es el error cometido en la medición. Valores de resolución típicos en ADC de un MCU son 8, 10, 12 y 16 bits.

En cuanto a la tensión de referencia, esta es la tensión máxima que puede medir el ADC. Esta tensión luego es dividida por 2^n que representa el valor más chico de tensión que es capaz de medir.

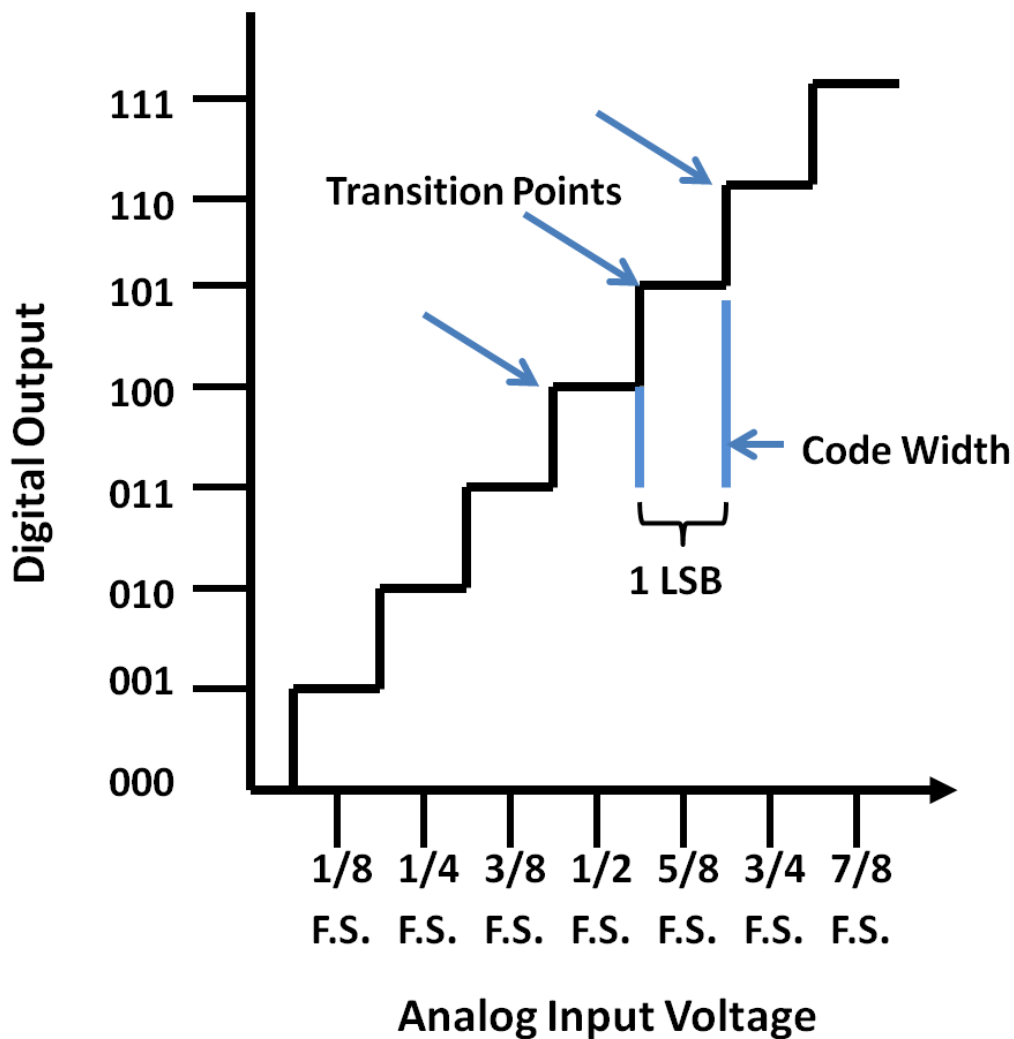


Figura 3: Resultado ADC vs tensión de entrada

En la figura anterior se muestra un ADC de 3 bits de resolución, por lo tanto la tensión de referencia queda dividida en ocho partes. Solo cuando el valor de tensión supere un umbral de tensión es que el resultado del ADC cambiará. Por ejemplo, para 3 bits y 8 V, los escalones de tensión de de 1 V, por lo que el resultado del ADC solo cambiará cuando la tensión medida supere el 1 V y luego los 2 V, 3 V y así respectivamente.

Desde el punto de vista del MCU, para obtener el valor de tensión que indica el conversor, se puede calcular como:

$$V_{in} = \frac{V_{REF}}{2^n} \times R \quad (1)$$

Donde V_{in} es la tensión medida que se desea calcular, V_{REF} es la tensión de referencia, n es la cantidad de bits de resolución y R es el resultado del ADC. Ejemplo, para una tensión de referencia de 3.3 V, resolución de 12 bits y un resultado de 3095 el valor de tensión que se está midiendo es de 2.49 V.

1.3 TEOREMA DE NYQUIST

El teorema de Nyquist de muestreo afirma que para poder capturar una señal apropiadamente, se necesita una velocidad de muestreo igual o mayor a dos veces la frecuencia de interés.

$$f_s \geq 2 * f_N \quad (1)$$

Esto asegura que no se produzca el efecto de aliasing, que es una ilusión de que en la señal hay una frecuencia que no existe.

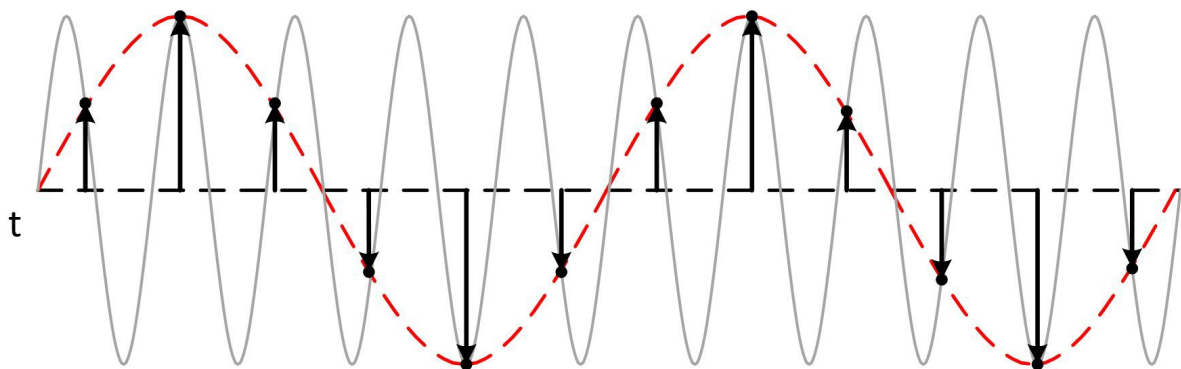


Figura 4: Efecto aliasing

Arriba se ve una señal que es muestreada a una frecuencia inferior a la necesaria, y se ve que si se recompusiera la señal original a partir de las muestras tomadas, el resultado (en línea roja punteada) es una señal de frecuencia inferior que no es real.

1.4 GLOSARIO

A continuación hay una lista de términos usados con sus respectivas definiciones para aclarar cualquier posible confusión.

- **MCU**: microcontrolador.
- **ADC**: conversor analógico a digital.
- **SAR**: ADC del tipo aproximaciones sucesivas.
- **Aliasing**: efecto luego de digitalizar que hace que aparezca una señal que nunca existió. Producido por no muestrear a la velocidad adecuada.
- **Resolución**: cantidad de bits de resultado de la conversión del ADC.
- **Tensión de referencia**: máxima tensión que mide el ADC.
- **LSB**: bit menos significativo, asociado al error mínimo que se comete en una conversión A/D.

1.5 REFERENCIAS ADICIONALES

A continuación se deja material adicional para el que desee profundizar más en el tema que se describió.

- [Types of A/D Converters](#)

ANEXO I: ADC EN LA RASPBERRY PI PICO

En este MCU tenemos un ADC de 12 bits por aproximaciones sucesivas. Para hacer uso de este, vamos a aprovechar el SDK en lenguaje C. La documentación completa se puede encontrar [aquí](#).

Tiene cuatro entradas analógicas conectadas a los GPIO 26 a 29 (canales 0 a 3 respectivamente). Cada entrada debe habilitarse previamente a usarse para desconectarla de cualquier otra función que pueda cumplir. Un canal adicional, el 4, está conectado a un sensor de temperatura interno que mide la temperatura del procesador.

Solo existe un conversor en el MCU, por lo que las entradas están multiplexadas, es decir, solo una a la vez se toma para convertir. Por eso, si se desea medir más de un canal, debe cambiarse la entrada elegida antes de realizar la conversión.

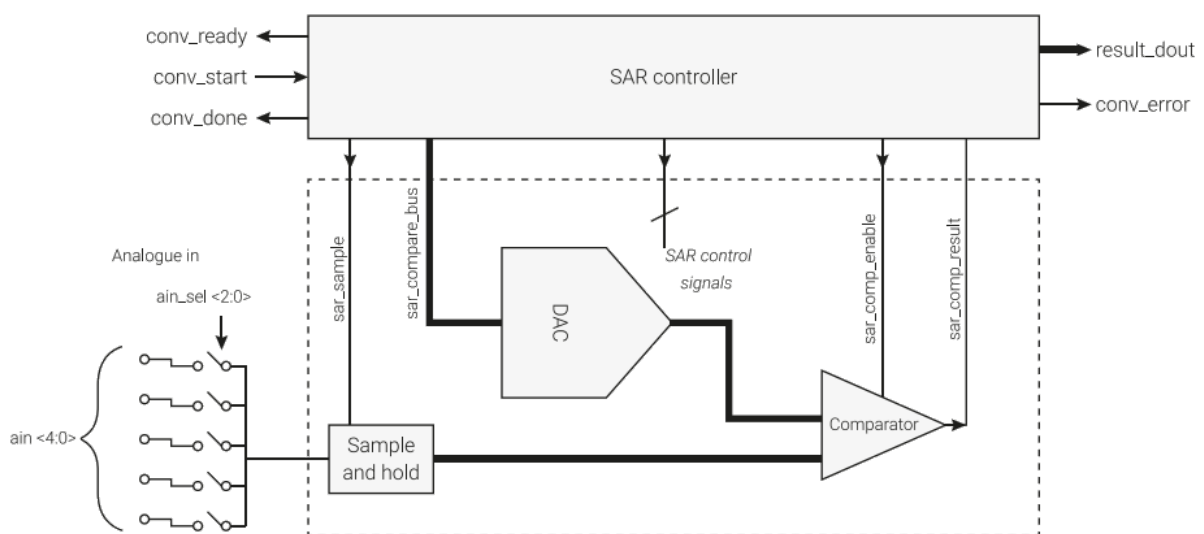


Figura 5: Diagrama en bloques del ADC de la Raspberry Pi Pico

Para hacer uso del ADC, hacen falta algunas pocas funciones:

```
// Habilitar el ADC
adc_init();
// Asegurar que el GPIO sea habilitado como entrada analógica y no digital
adc_gpio_init(26);
// Elegir la entrada analógica a usar (canal 0 para GPIO26)
adc_select_input(0);
// Leer el canal elegido
uint16_t res = adc_read();
```