



# Sistemas Electrónicos Digitales

## Tema #3

### 9. Conversores A/D y D/A



1. Introducción
2. GPIO: General Purpose Input/Output
3. Arquitectura Arm Cortex-M4
4. Interrupciones
5. C en ensamblador
6. Temporizadores (Timers)
7. Direct Memory Access
8. Comunicaciones Serie
9. Conversores A/D y D/A

# INDICE ESPECÍFICO

- **Introducción**
- **Conceptos generales de convertidores D/A**
  - Tipos de convertidores D/A
  - Características de los convertidores D/A
  - Especificaciones de los convertidores D/A
- **Conceptos generales de convertidores A/D**
  - Tipos de convertidores A/D
  - Características de los convertidores A/D
  - Especificaciones de los convertidores A/D

# INDICE ESPECÍFICO

- **Introducción**
- **Conceptos generales de convertidores D/A**
  - Tipos de convertidores D/A
  - Características de los convertidores D/A
  - Especificaciones de los convertidores D/A
- **Conceptos generales de convertidores A/D**
  - Tipos de convertidores A/D
  - Características de los convertidores A/D
  - Especificaciones de los convertidores A/D

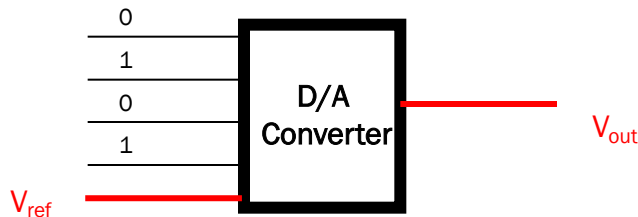


# ¿Por qué se necesitan los convertidores?

- Los sistemas embebidos precisan a menudo:
    - Medir valores de magnitudes físicas
    - Controlar actuadores analógicos
  - Estos parámetros son normalmente continuos (analógicos).
  - No están en un formato digital que un microprocesador pueda procesar.
  - Es necesario convertirlos de analógico a digital o viceversa.
- 
- Temperatura
    - ✓ Termostatos
    - ✓ Controlador de motor de coche
    - ✓ Monitor de reacción química
    - ✓ Seguridad (Ej. gestión térmica del microprocesador)
  - Intensidad de la luz (infrarroja o ultravioleta)
    - ✓ Cámara digital
    - ✓ Receptor de control remoto por infrarrojos
    - ✓ Monitor UV
  - Posición rotatoria
    - ✓ Medidor de viento
  - Presión
    - Monitor de presión arterial
    - Altímetro
    - Ordenador de buceo
  - Aceleración
    - Controlador *airbag*
    - Estabilidad del vehículo
    - Videojuego remoto
  - Tensión mecánica
  - Otro
    - EKG, EEG
    - Alcoholímetro

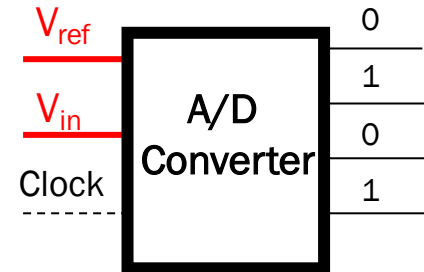
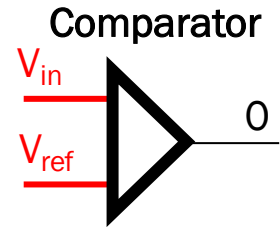
# Convertidor Digital a Analógico

- Objetivo: **generar una tensión o corriente analógica como señal de salida**
  - Ej. Señal de audio, brillo de señal de video.
- DAC: "Generar el voltaje analógico que es esta fracción de  $V_{ref}$ "
- Ecuación del convertidor digital a analógico
  - $n$  = código de entrada
  - $N$  = número de bits de resolución del convertidor
  - $V_{ref}$  = voltaje de referencia
  - $V_{out}$  = voltaje de salida. Ya sea:
    - $V_{out} = V_{ref} * n / (2N)$  o
    - $V_{out} = V_{ref} * (n + 1) / (2N)$
    - El término offset +1 depende de la configuración interna del DAC - verifique la hoja de datos para asegurarse



# Convertidor Analógico a Digital

- Un **comparador** nos dice "**¿Es  $V_{in} > V_{ref}$ ?**"
  - Compara una tensión de entrada analógica con una tensión de referencia analógica y determina cuál es mayor, devolviendo un número de 1 bit
- Un **convertidor analógico a digital [AD o ADC]** nos dice cómo de grande es  $V_{in}$  referido a una fracción de  $V_{ref}$ .
  - Lee una señal de entrada analógica (generalmente un voltaje) y produce un número de múltiples bits correspondiente en la salida.

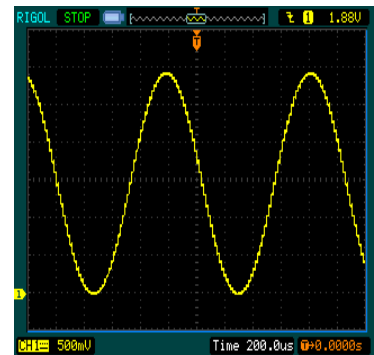
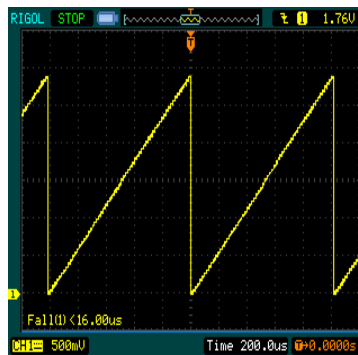
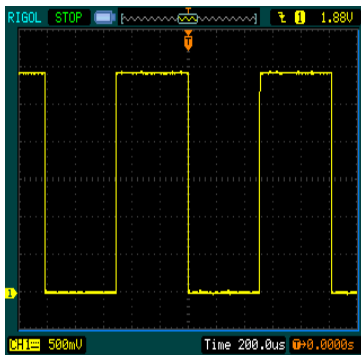


# INDICE ESPECÍFICO

- **Introducción**
- **Conceptos generales de convertidores D/A**
  - Tipos de convertidores D/A
  - Características de los convertidores D/A
  - Especificaciones de los convertidores D/A
- **Conceptos generales de convertidores A/D**
  - Tipos de convertidores A/D
  - Características de los convertidores A/D
  - Especificaciones de los convertidores A/D

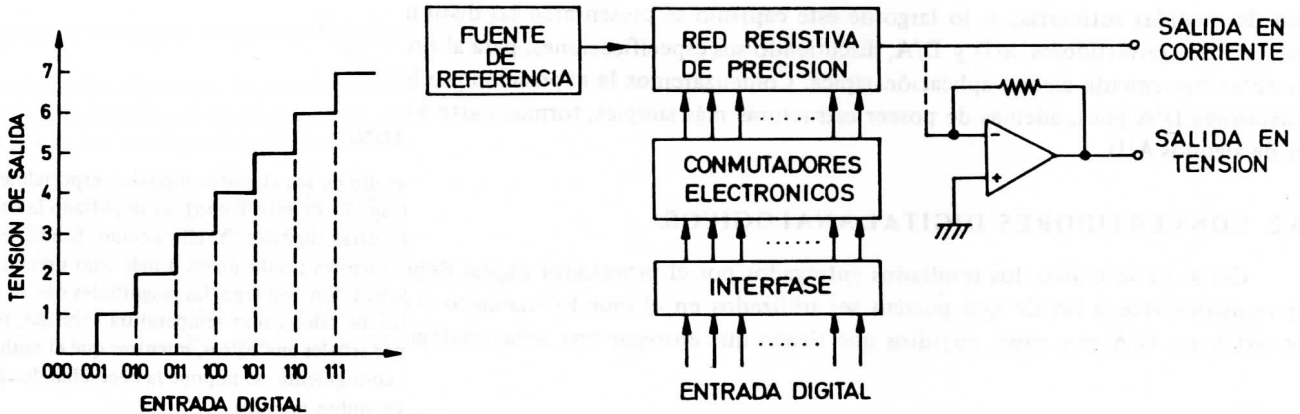
# Ejemplo: Generador de forma de onda.

- El convertor digital-analógico (DAC) puede usarse para generar formas de onda arbitrarias.
- Pasos:
  - Generar una tabla de valores
  - Actualizar el valor de salida del DAC
  - Retardo adecuado
  - Repetir desde el paso segundo



# Convertidores D/A

- Transforman un código binario en un valor de tensión o corriente relacionado
- Habitualmente realizados con redes resistivas

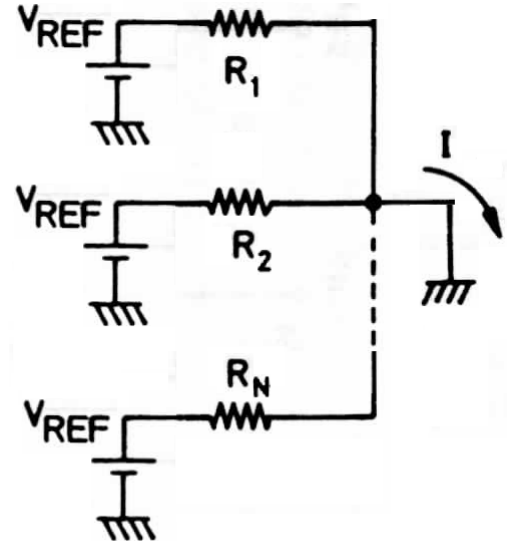


# Convertidores D/A: principio de funcionamiento

- En la figura:

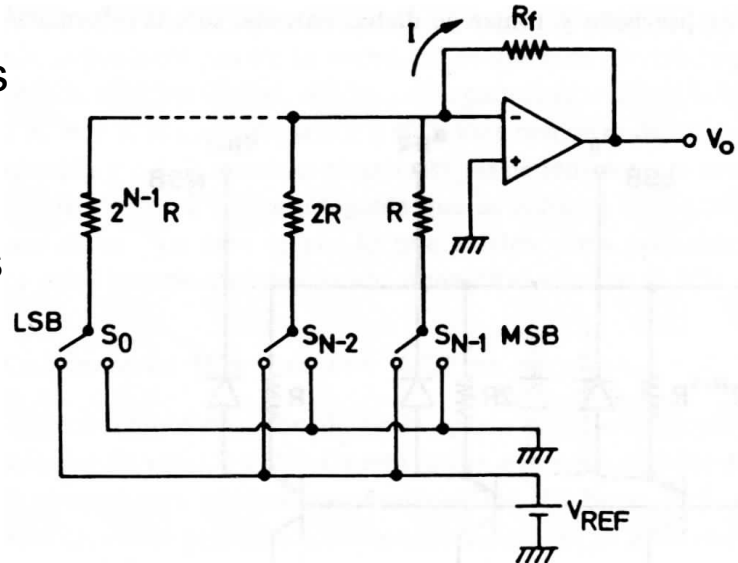
$$I = I_1 + I_2 + \dots + I_N = V_{REF} \left( \frac{1}{R_1} + \dots + \frac{1}{R_N} \right)$$

- Si  $R_1, R_2, \dots, R_N$  son proporcionales a los pesos en un código binario, entonces tendremos una corriente proporcional a  $V_{ref}$  y dicho código.



## Convertidor con resistencias ponderadas

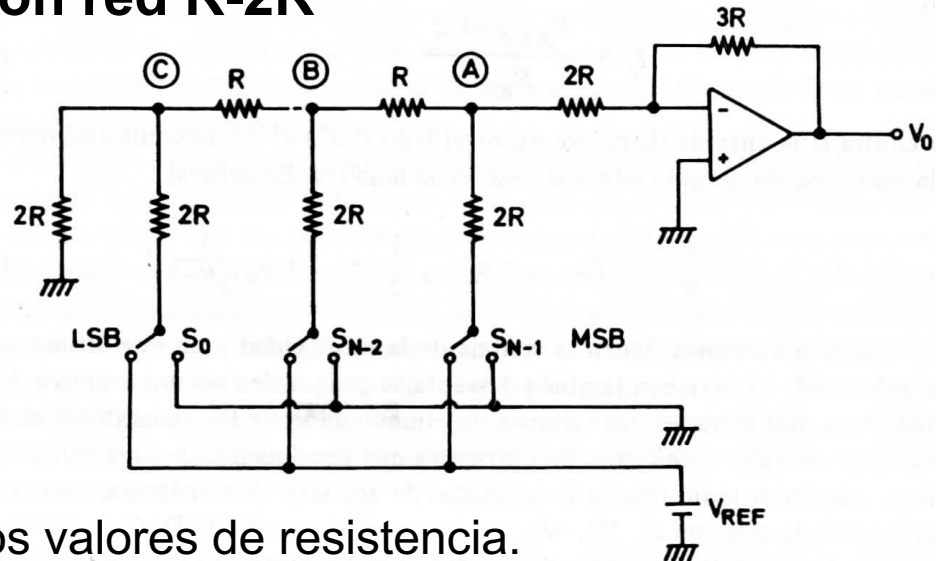
- El valor de las resistencias es proporcional al peso que representan.
- Esto hace que sus valores sean muy dispares y por tanto sea difícil de conseguir con la precisión deseada.



$$V_0 = \frac{V_{REF} R_f}{R} \left( a_{N-1} + a_{N-2} \frac{1}{2} + \dots + a_0 \frac{1}{2^{N-1}} \right)$$



## Convertidor con red R-2R

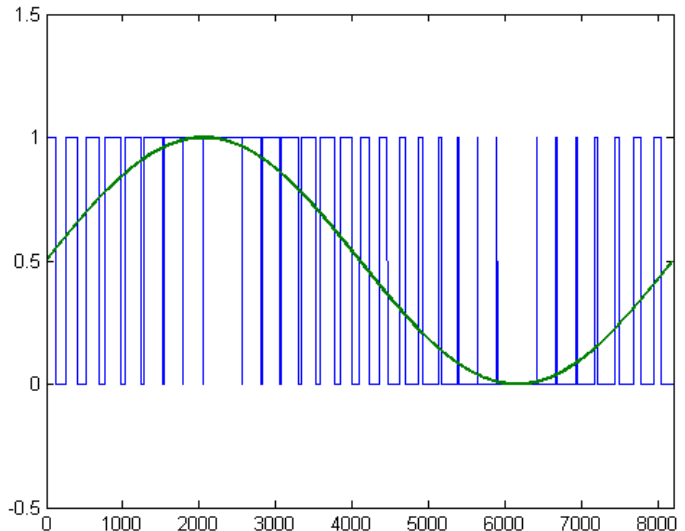
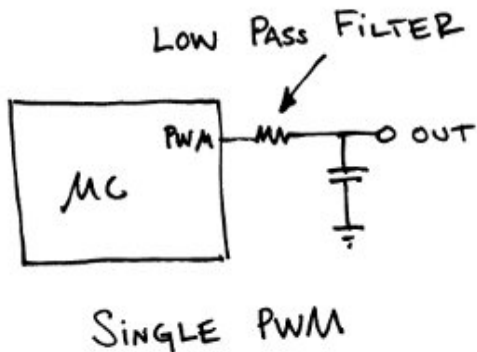


- Sólo presenta dos valores de resistencia.
- Esto hace que sea el más usado.

$$V_0 = -V_{REF} \left( a_{N-1} \frac{1}{2} + a_{N-2} \frac{1}{2^2} + \dots + a_0 \frac{1}{2^N} \right)$$

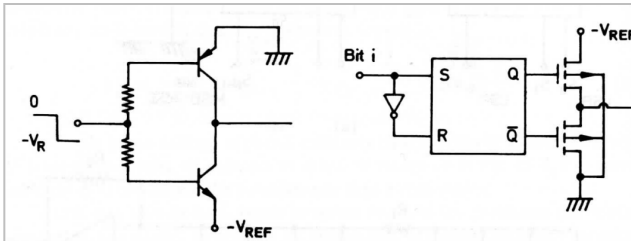
## Convertidor PWM

- Genera una señal PWM, que si filtramos nos dará un valor medio dependiente del ancho de pulso.



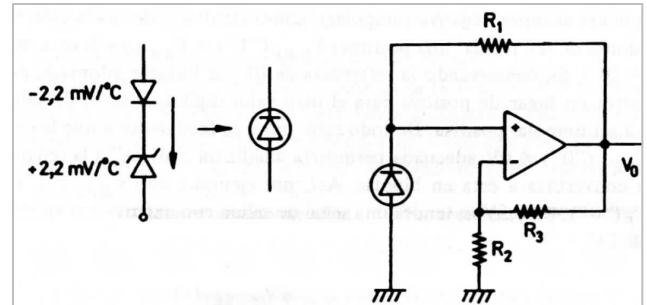
# Características de los DAC: Elementos

## Conmutadores



- Transistores bipolares
- MOS + biestable D

## Fuente de referencia



- Diodo Zener compensado en temperatura

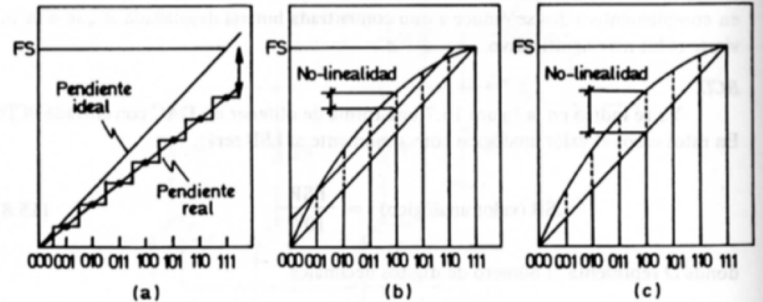
$$V_0 = V_Z \left(1 + \frac{R_3}{R_2}\right)$$

# Características de los DAC: **Formatos**

- Formatos de entrada:
  - Binario natural y desplazado
  - Complementos (a uno y a dos)
  - BCD
- Formatos de salida:
  - Tensión o corriente
  - Unipolar / Bipolar

# Especificaciones de los DAC

- Resolución.
- Error de “offset”.
- Error de escala.
- Linealidad.
- Precisión.
- Tiempo de asentamiento.
- Sensibilidad con la temperatura.





# Características del DAC en STM32.

- Formato de entrada
  - Binario natural
- Formato de salida
  - Unipolar
  - Bipolar
- Resolución
  - 12 Bits



# Ejemplo DAC: Código de inicialización

```
__HAL_DAC_ENABLE(&hdac, DAC_CHANNEL_1);
HAL_DAC_Start(&hdac, DAC_CHANNEL_1);
HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1, DAC_ALIGN_12B_R, 2048);
uint16_t s = 0;
float f = 0;
while (1)
{
    f = f+ 0.01f;
    if (f > 6.28f) f = 0;
    s = (uint16_t) (1000.0f*sin(f)+1000.0f);
    HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1, DAC_ALIGN_12B_R, s);
    HAL_DAC_Start(&hdac, DAC_CHANNEL_1);
}
```



# Ejemplo DAC: Función generadora senoide

```
void tone_init(void) {
    dac_init();
    sinewave_init();
}

void sinewave_init(void) {
    int n;
    for (n = 0; n < NUM_STEPS; n++) {
        sine_table[n] = MAX_DAC_CODE * (1 + sin(n*2*PI/NUM_STEPS)) / 2;
    }
}
```



# Ejemplo DAC: Función salida senoide o rampa.

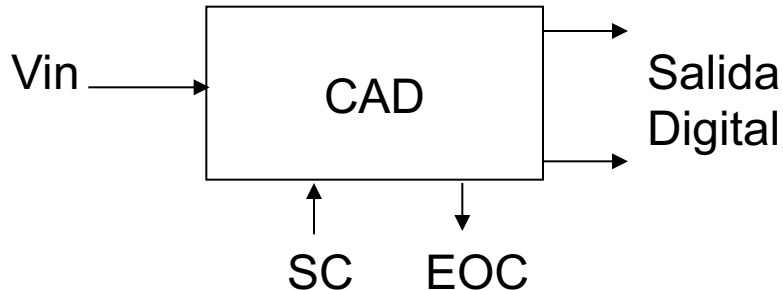
```
void tone_play(int period_us, int num_cycles, wavetype wave) {
    int sample, step;
    while(num_cycles-- > 0) {
        for (step = 0; step < NUM_STEPS; step++) {
            switch(wave) {
                case SINE: sample = sine_table[step];
                    break;
                case SQUARE: sample = step < NUM_STEPS / 2 ? 0 : MAX_DAC_CODE;
                    break;
                case RAMP: sample = (step * MAX_DAC_CODE) / NUM_STEPS;
                    break;
            }
            dac_set(sample);
            delay_us(period_us);
        }
    }
}
```

# INDICE ESPECÍFICO

- **Introducción**
- **Conceptos generales de convertidores D/A**
  - Tipos de convertidores D/A
  - Características de los convertidores D/A
  - Especificaciones de los convertidores D/A
- **Conceptos generales de convertidores A/D**
  - Tipos de convertidores A/D
  - Características de los convertidores A/D
  - Especificaciones de los convertidores A/D
- **Convertidores en ARM CORTEX M4**

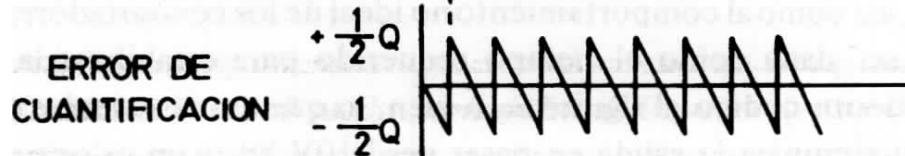
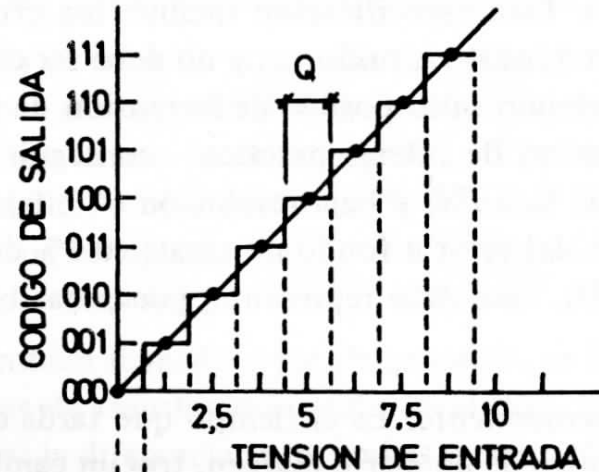
# Convertidores A/D

- Proporcionan un código digital en función del valor de tensión o corriente a su entrada.
- Los códigos digitales suelen tener un número de bits comprendido entre 8 y 24, siendo los más comunes 8, 12 y 16.
- Es la parte principal de un Sistema de Adquisición de Datos (SAD).



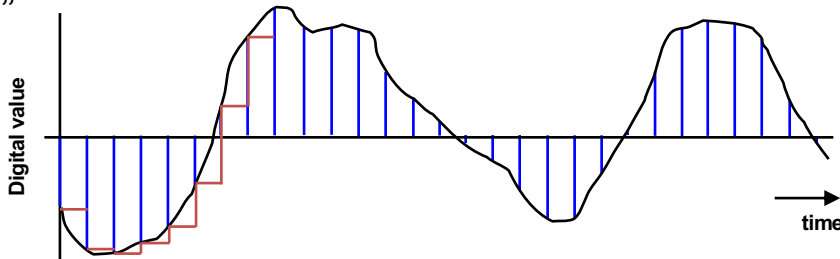
# Convertidores A/D: fases de la conversión

- **Cuantificación:** la señal de entrada se clasifica en un conjunto discreto de estados.
- **Codificación:** a cada estado se le asigna un código digital.
- La **escalera** es la mejor aproximación posible para la clasificación en estados, por lo que introducen un **error de cuantificación**.



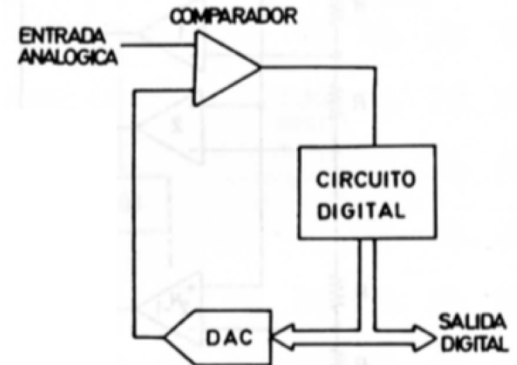
# Convertidores A/D: muestreo, cuantificación y codificación

- La onda de entrada es muestreada a una cadencia constante  $\Delta t$ .
  - Cada muestra representa la amplitud instantánea en el instante de muestreo.
  - Por ejemplo: “en 37 ms, el valor era 1.91341.. V”
- La señal se discretiza.
  - El valor se clasifica en un grupo. “El valor 1.91341.. V entra dentro del grupo “X”, que abarca los valores comprendidos entre 1.901 y 1.998 V”
- Y se codifica.
  - “Al grupo “X” (1.901 - 1.998 V) le corresponde el código 0x018”



## Tipos de convertidores A/D

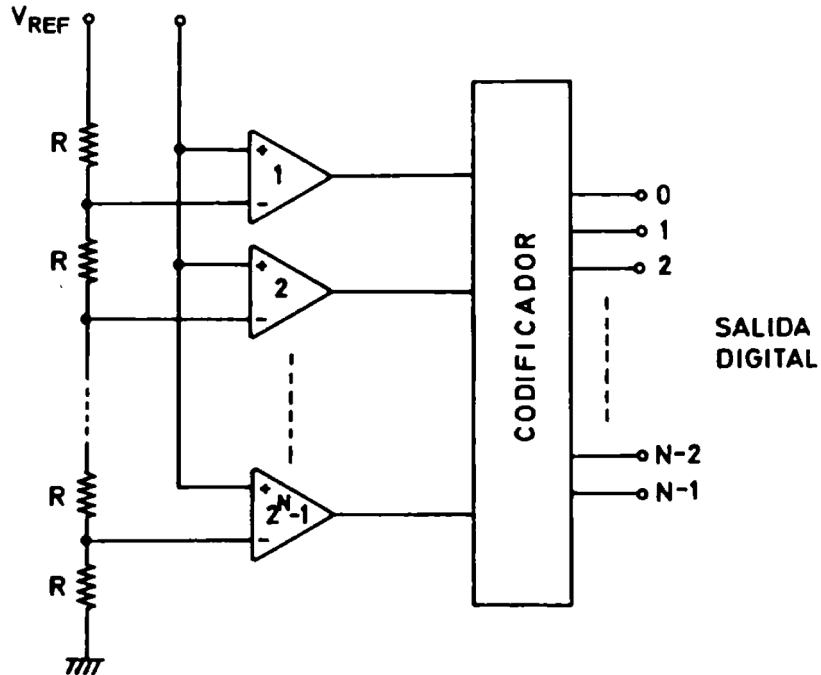
- Bucle abierto.
  - **Flash.**
  - **A ancho de impulso.**
  - **De doble rampa.**
- Bucle cerrado.
  - **Con contadores.**
  - **De aproximaciones sucesivas.**
  - **Sigma-Delta.**
  - **Pipeline.**



# Tipos de convertidores A/D (II)

## Convertidor flash

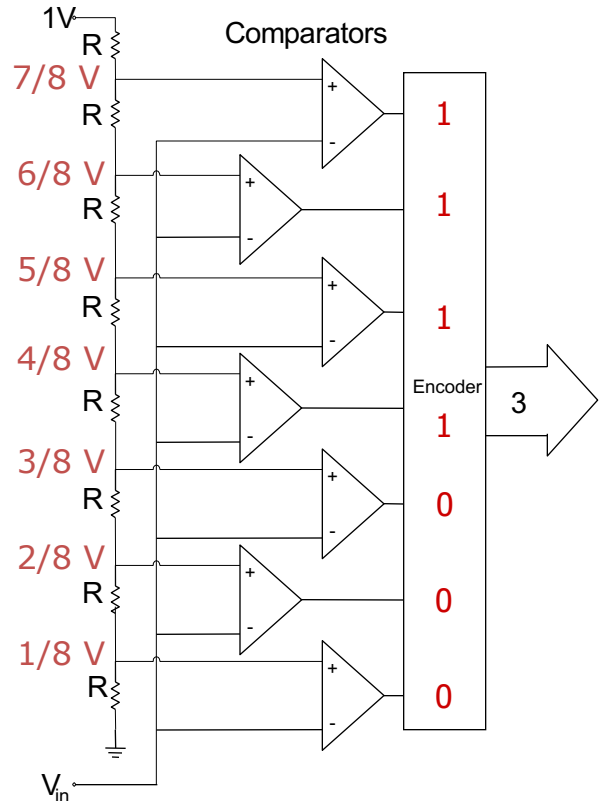
- Procesos de cuantificación y codificación separados:
  - La cuantificación es mediante comparadores.
  - El codificador genera la salida digital.
- De alta velocidad.
- Para un número reducido de bits.



# Tipos de convertidores A/D (III)

## Convertidor flash

- El divisor de tensión establece los niveles de voltaje de cada estado.
- Un comparador determina si la entrada es mayor o menor que su estado.
- La salida de los comparadores se codifica con un valor binario (codificador de prioridad)
- Componentes necesarios
  - $2^N$  resistores
  - $2^N - 1$  comparadores

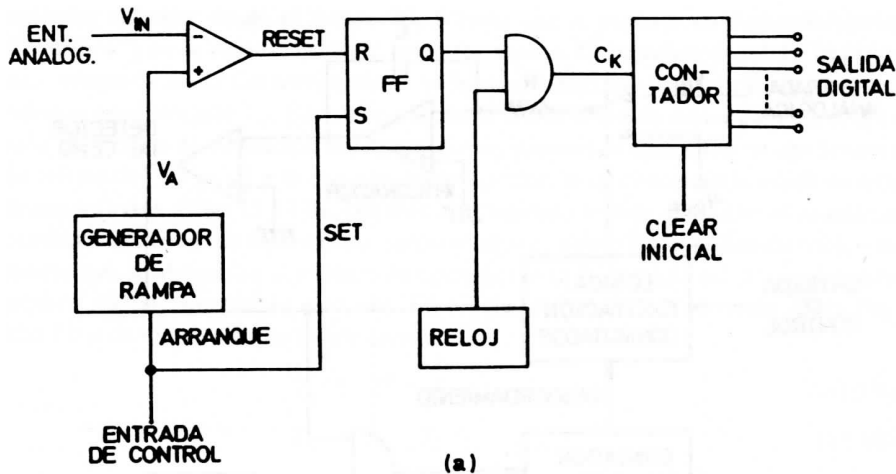




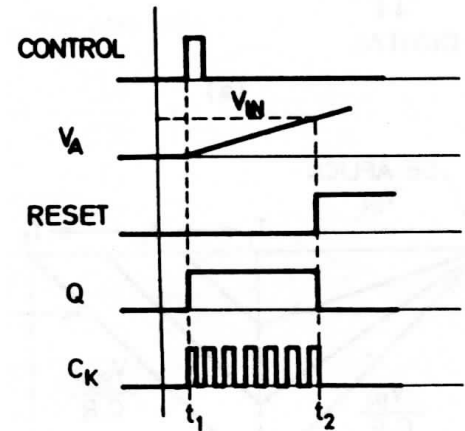
# Tipos de convertidores A/D (IV)

## Convertidor a ancho de impulso

- Integrador de simple pendiente: transforma la entrada en un impulso temporal que se mide mediante un contador.



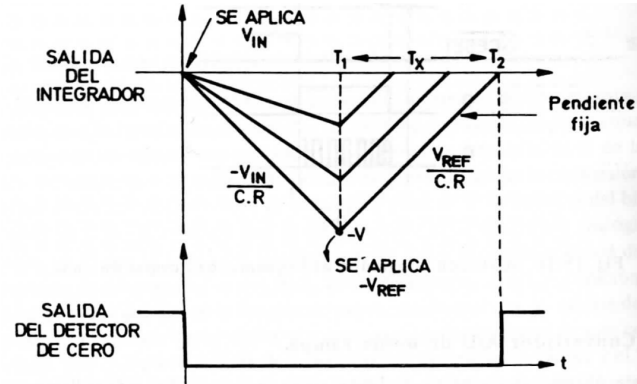
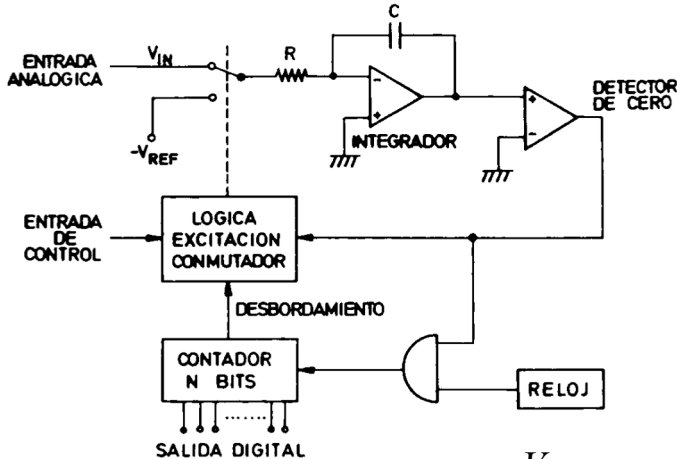
(a)



# Tipos de convertidores A/D (V)

## Convertidor de doble rampa

- Se generan dos cuentas: una durante un tiempo fijo  $T_1$  y otra a pendiente constante un tiempo  $T_x$ .
- Convertidor de buena precisión, pero lento.



$$T_1 = 2^N T$$

$$T_x = \lambda T$$



$$-V = -\frac{V_{IN}}{CR} T_1 = -\frac{V_{REF}}{CR} T_x$$

$$T_x = \lambda T = \frac{V_{IN}}{V_{REF}} T_1 = \frac{V_{IN}}{V_{REF}} 2^N T$$

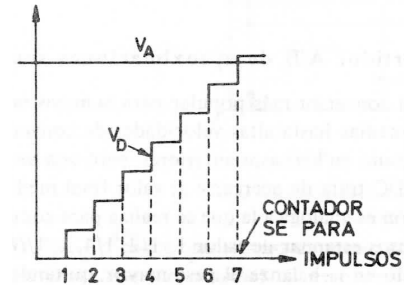
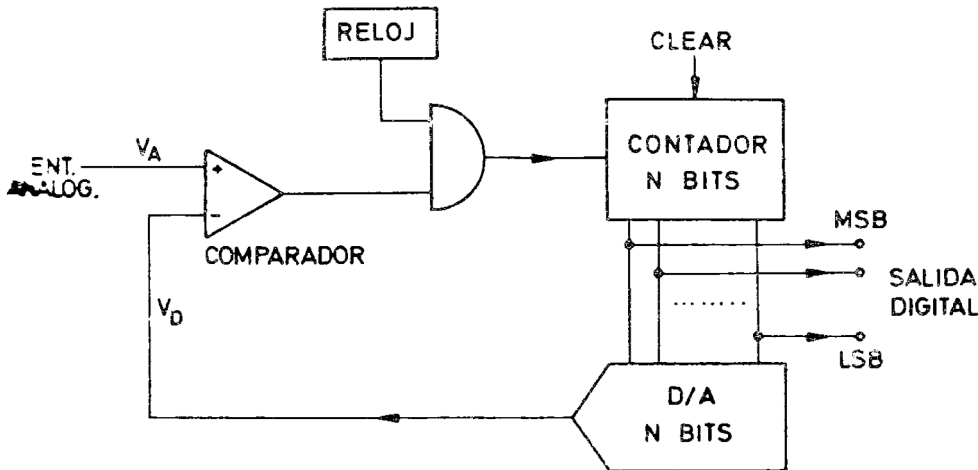


$$\lambda = V_{IN}$$

# Tipos de convertidores A/D (VI)

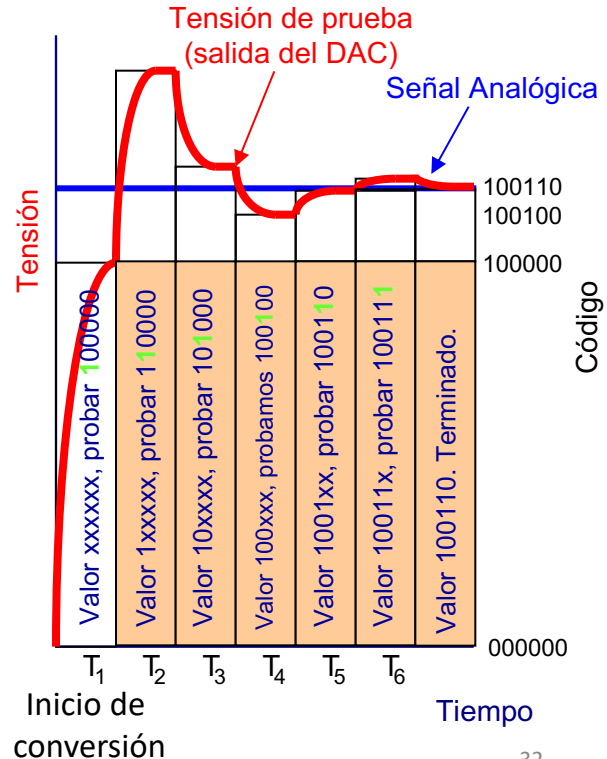
## Convertidor con contadores

- Existen dos tipos:
  - Con rampa en escalera
  - De cuenta continua

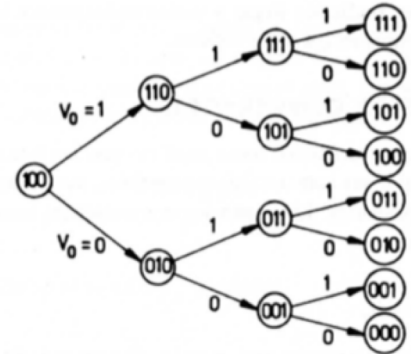
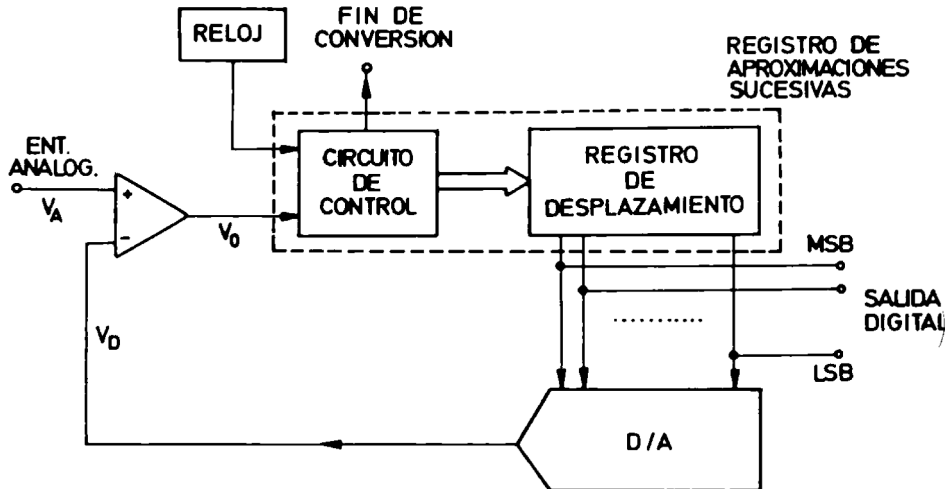


## Convertidor de aproximaciones sucesivas: principio de funcionamiento.

- Similar a la pesada con pesos 1, 1/2, 1/4 ,.. 1/N Kg.
- Pasos:
  1. Se pone el bit más significativo a "1".
  2. Si la entrada es menor, se cambia a "0", sino se queda a "1".
  3. Se pone el siguiente bit significativo a "1".
  4. Repetimos el proceso.
  5. Terminamos cuando llegamos al bit menos significativo.



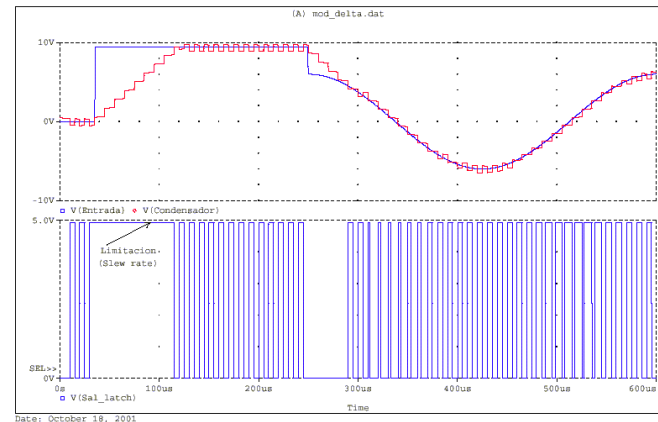
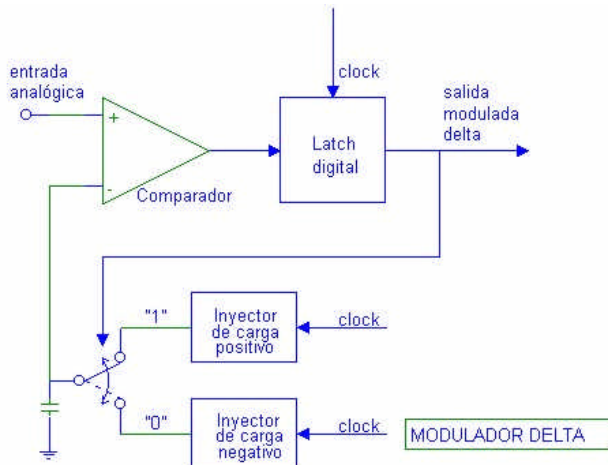
## Convertidor de aproximaciones sucesivas: esquema de bloques



# Tipos de convertidores A/D (IX)

## Convertidor sigma-delta

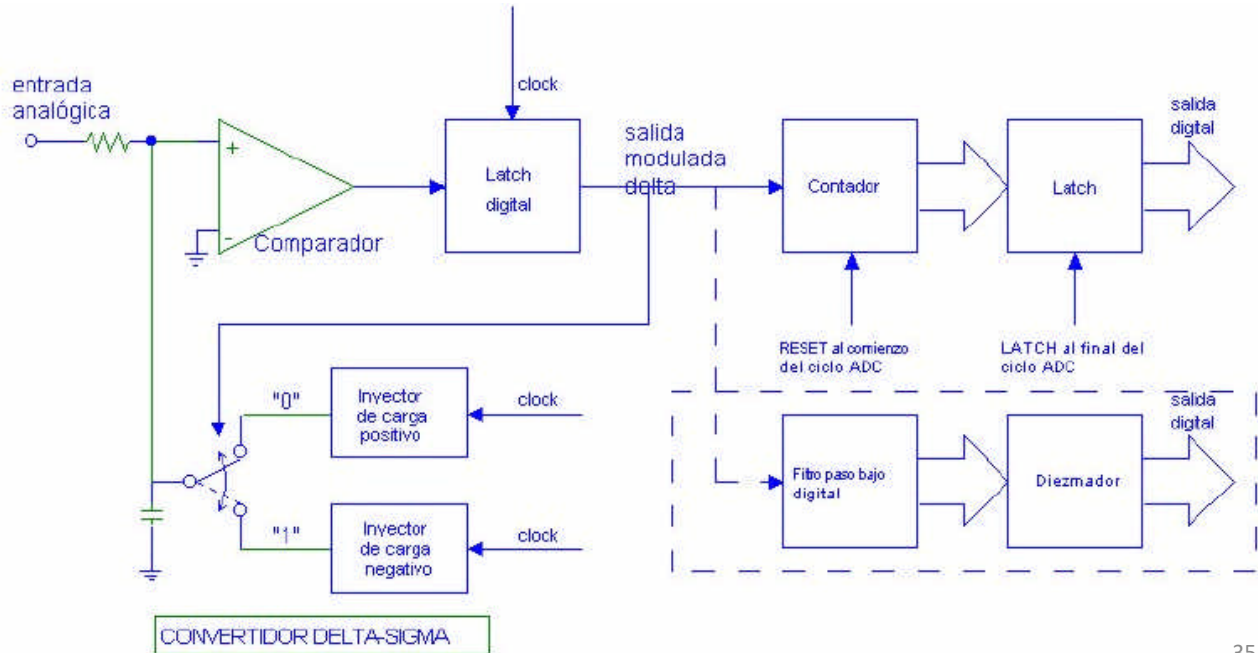
- Se trata de convertidores A/D de un solo bit.
- Ejemplo de modulador delta, que genera un número relativo de “1” y “0” proporcional a la pendiente de la señal de entrada.



# Tipos de convertidores A/D (X)

## Convertidor sigma-delta (II)

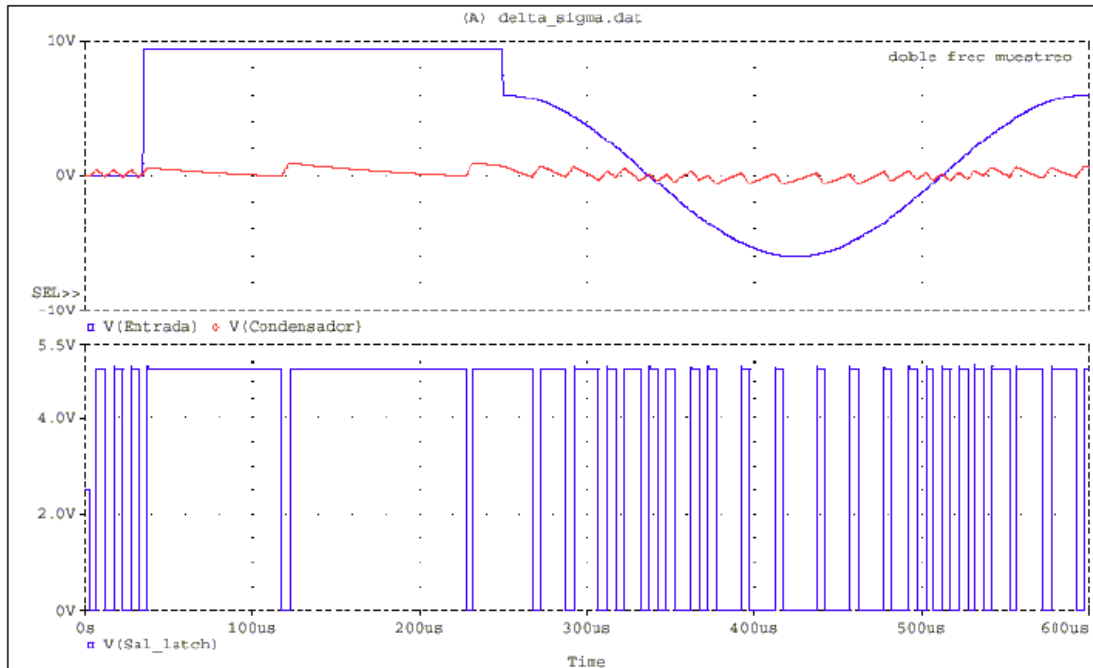
- El convertidor combina un modulador **delta** y un contador sumador **sigma**.



# Tipos de convertidores A/D (XI)

## Convertidor sigma-delta (III)

- Ejemplo de conversión.

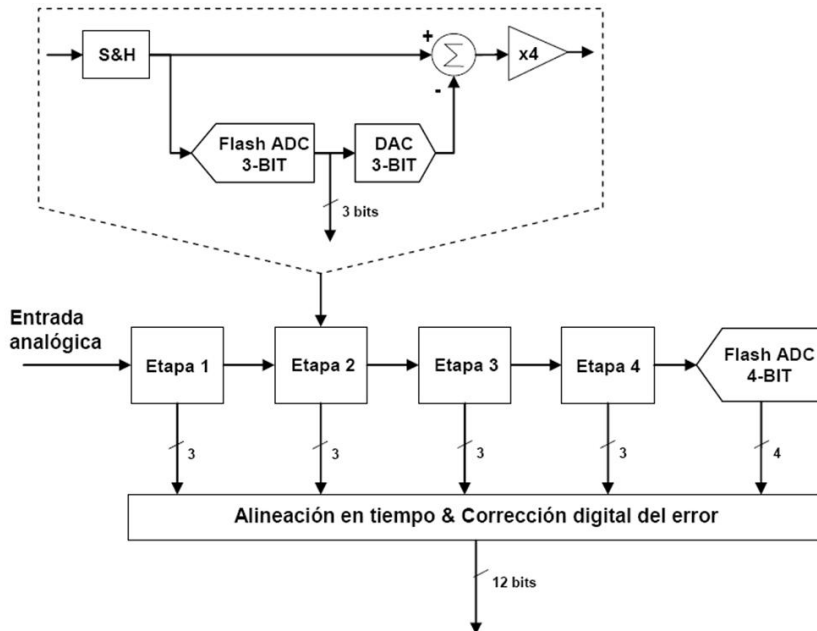




# Tipos de convertidores A/D (XII)

## Convertidor pipeline

- Son convertidores tipo flash trabajando en paralelo, cada uno con un número reducido de bits. En el ejemplo, son 3 bits.



## Especificaciones de los ADC

- Resolución
  - Linealidad
  - Precisión
  - Tiempo de conversión
  - Error de cuantificación
- } Similar a los DAC

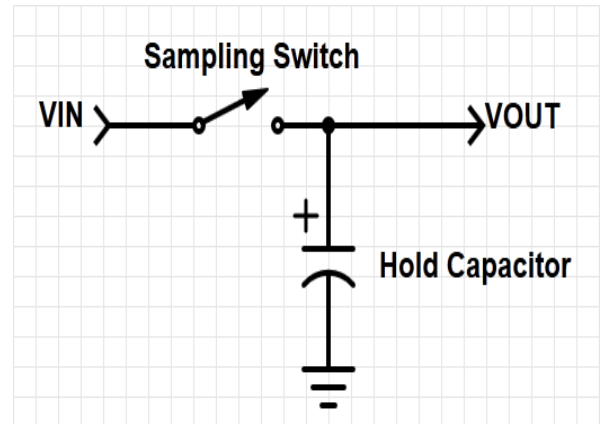
# Cuestiones acerca del muestreo de las señales

- Criterio de Nyquist
  - $F_{\text{muestreo}} \geq 2 * F_{\text{max}}$  de la señal
  - Componentes con frecuencia superior a  $\frac{1}{2} F_{\text{muestreo}}$  son reflejados (*aliased*) distorsionando la señal
  - Necesitamos un filtro paso bajo para asegurar que no haya componentes superiores a  $\frac{1}{2} F_{\text{muestreo}}$
- Aplicación práctica del criterio de Nyquist
  - El teorema asume que tenemos un filtro perfecto (caída abrupta)
  - Los filtros reales no son así; los económicos tienen caídas leves (un filtro de primer orden: 20 dB/década)
  - Así que tenemos que elegir una frecuencia de muestreo suficientemente alta para que el filtro atenúe adecuadamente las componentes de alta frecuencia

- Diferencial
  - Usa dos canales, y calcula la diferencia entre ambos
  - Alta inmunidad al ruido
  - Algunos circuitos de medida son de naturaleza diferencial (Puente de Wheatstone)
- Multiplexado
  - Normalmente se comparte un ADC entre múltiples entradas
  - Se necesita seleccionar la entrada, y dejar un tiempo de establecimiento antes de muestrear
- Acondicionado de señal
  - Amplificar y filtrar la señal de entrada
  - Proteger las entradas del CAD con “clamping diodes”<sup>40</sup>

# Muestreo y retención

- Algunos conversores A/D requieren que la señal de entrada se mantenga constante durante la conversión, (ej. los de aproximaciones sucesivas)
- En otros casos, capturar el valor pico o muestrear un punto específico requiere un dispositivo de muestreo.
- Esta función la realiza el circuito de Muestreo y Retención (sample and hold device, S&H)



# Características del ADC en STM32.

- Tipo de convertidor
  - De aproximaciones sucesivas (tres por chip)
- Formato de entrada
  - Unipolar (Tensiones comprendidas entre 0 V y  $V_{ref}=3,6$  V)
- Formato de salida
  - Binario natural
  - Bipolar
- Resolución
  - 12 Bits
- Velocidad de muestreo
  - 2 Msps

# Forward Transfer Function Equation

¿Qué código “n” usará el ADC para representar  $V_{in}$ ?

## General Equation

n = converted code

$V_{in}$  = sampled input voltage

$V_{+ref}$  = upper voltage reference

$V_{-ref}$  = lower voltage reference

N = number of bits of resolution in ADC

$$n = \left\lfloor \frac{V_{in} - V_{-ref}}{V_{+ref} - V_{-ref}} 2^N + 1/2 \right\rfloor$$

## Simplification with $V_{-ref} = 0 V$

$$n = \left\lfloor \frac{V_{in}}{V_{+ref}} 2^N + 1/2 \right\rfloor$$

$$n = \left\lfloor \frac{3.3V}{5V} 2^{10} + 1/2 \right\rfloor = 388$$

$\lfloor X \rfloor = \text{floor}(X)$     *floor(X) nearest integer I such that  $I \leq X$*   
*floor(x+0.5) rounds x to the nearest integer*

# Inverse Transfer Function

¿Qué rango de voltajes  $V_{in\_min}$  a  $V_{in\_max}$  representa el código “n”?

## General Equation

$n$  = converted code

$V_{in\_min}$  = minimum input voltage for code  $n$

$V_{in\_max}$  = maximum input voltage for code  $n$

$V_{+ref}$  = upper voltage reference

$V_{-ref}$  = lower voltage reference

$N$  = number of bits of resolution in ADC

$$V_{in\_min} = \frac{n - \frac{1}{2}}{2^N} (V_{+ref} - V_{-ref}) + V_{-ref}$$

$$V_{in\_max} = \frac{n + \frac{1}{2}}{2^N} (V_{+ref} - V_{-ref}) + V_{-ref}$$

## Simplification with $V_{-ref} = 0$ V

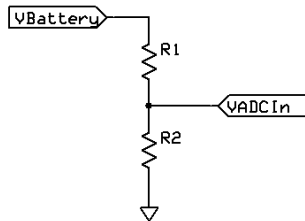
$$V_{in\_min} = \frac{n - \frac{1}{2}}{2^N} (V_{+ref})$$

$$V_{in\_max} = \frac{n + \frac{1}{2}}{2^N} (V_{+ref})$$



# Ejemplo: Lectura de la tensión en una pila botón

- El conversor analógico-digital-(ADC) puede usarse para leer el valor de tensión de una pila de 3 V.
- OJO: poner en la entrada del ADC una tensión mayor que  $V_{DD}+0.3$  V puede dañarlo.
- Un divisor resistivo convenientemente calculado permite eliminar el riesgo .



- Por ejemplo, si  $R1 = R2 = 1$  Mohm, la tensión se reduce a la mitad. Una pila de 3V proporcionaría como máximo 1,5 V.



# Ejemplo ADC: Código.

```
#include <platform.h>
#include <adc.h>

#define R1 (1e6)
#define R2 (1e6)
#define SCALE_FACTOR ((R1+R2)/(R2))
#define VREF (3.3)

int main(void) {
    adc_init();

    while(1) {
        volatile float vbat;
        volatile int res = adc_read();

        // Scale the adc result to a voltage.
        vbat = (float)res * SCALE_FACTOR * VREF / ADC_MASK;
    }
}
```



UNIVERSIDAD  
POLITÉCNICA  
DE MADRID

POLITÉCNICA

# Fin