



UNIVERSIDAD  
**AUTÓNOMA**  
DE QUERÉTARO



**FACULTAD**  
DE INGENIERÍA

# PRÁCTICA 4: ADC MICROCONTROLADORES

*RAMÍREZ ÁLVAREZ CARLO IVÁN - 280847*  
*ONTIVEROS MARTÍNEZ BEATRIZ - 244784*  
*TREJO DOMÍNGUEZ NELLY BIBIANA - 242494*

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO  
INGENIERÍA BIOMÉDICA

MICROCONTROLADORES  
ING. JOSÉ DE JESÚS SANTANA RAMÍREZ  
ENERO, 2023

## I. OBJETIVOS

Configurar la tarjeta a una frecuencia asignada, para adquirir señales analógicas a una velocidad de 1 millón de muestras por segundo y guardar los valores en un arreglo para ser enviados con botones externos asociados a un gpio a través del protocolo de comunicación asíncrona, todo esto usando interrupciones.

## II. MARCO TEÓRICO

### a. Conversión básica de señales

Un transductor permite relacionar las señales del mundo real y sus análogas eléctricas. Para compatibilizar la información con un sistema digital, se requiere de convertidores de datos del tipo ADC o DAC, según corresponda.

El diagrama de bloques de la Fig.1 muestra la secuencia desde que la variable física entra al sistema hasta que es transformada a señal digital (código binario). Para que dicha señal ingrese al convertidor análogo - digital, ésta debe ser muestreada, es decir, se toman valores discretos en instantes de tiempo de la señal análoga, lo que recibe el nombre de sampling. Matemáticamente es el equivalente a multiplicar la señal análoga por una secuencia de impulsos de periodo constante. Como resultado se obtiene un tren de impulsos con amplitudes limitadas por la envolvente de la señal analógica.

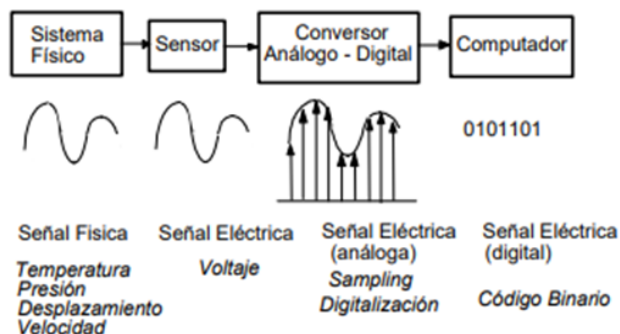


Figura 1. Conversión análogo – digital.

### b. Convertidor Analógico/Digital - ADC

El objetivo básico de un ADC es transformar una señal eléctrica análoga en un número digital equivalente. De la misma forma, un DAC transforma un número digital en una señal eléctrica análoga. Cada conversor ADC ó DAC, está determinado por una función de transferencia ideal de entrada – salida, que muestra la equivalencia entre el mundo digital y el análogo.

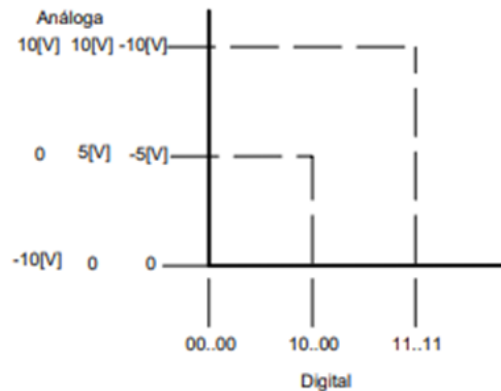


Figura 2. Curva de entrada - salida de un conversor ADC ó DAC.

En el caso de una señal unipolar entre 0 a 10 [V], su equivalente digital para n bits sería entre 00..00 para 0[V] (zero - scale), 10..00 para 5[V] y 11..11 a 10[V] (full - scale).

### c. ADC en TM4C123GH6PM

Un convertidor de analógico a digital (ADC) es un periférico que convierte un voltaje analógico continuo en un número digital discreto. Se incluyen dos módulos convertidores idénticos, que comparten 12 canales de entrada.

El módulo ADC TM4C123GH6PM presenta una resolución de conversión de 12 bits y admite 12 canales de entrada, además de un sensor de temperatura interno. Cada módulo ADC contiene cuatro secuenciadores programables que permiten el muestreo de múltiples fuentes de entrada analógicas sin la intervención del controlador. Cada secuenciador de muestra proporciona una programación flexible

con una fuente de entrada totalmente configurable, eventos de activación, generación de interrupciones y prioridad del secuenciador. Además, el valor de conversión se puede desviar opcionalmente a un módulo comparador digital. Cada módulo ADC proporciona ocho comparadores digitales. Cada comparador digital evalúa el valor de conversión ADC contra sus dos valores definidos por el usuario para determinar el rango operativo de la señal. La fuente de disparo para ADC0 y ADC1 puede ser independiente o los dos módulos ADC pueden operar desde la misma fuente de disparo y operar con las mismas o diferentes entradas. Un cambiador de fase puede retrasar el inicio del muestreo en un ángulo de fase específico. Cuando se utilizan ambos módulos ADC, es posible configurar los convertidores para iniciar las conversiones de forma coincidente o dentro de una fase relativa entre sí.

d. Diagrama de bloque

El microcontrolador TM4C123GH6PM contiene dos módulos convertidores de analógico a digital idénticos.

Estos dos módulos, ADC0 y ADC1, comparten los mismos 12 canales de entrada analógica. Cada módulo ADC funciona de forma independiente y, por lo tanto, puede ejecutar diferentes secuencias de muestreo así, muestreando cualquiera de los canales de entrada analógica en cualquier momento y generar diferentes interrupciones y disparadores.

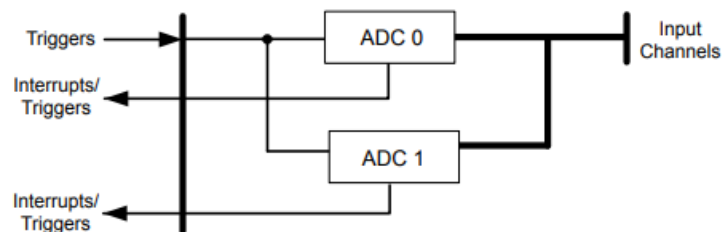


Figura 3. Conexión de dos módulos a las entradas analógicas y al bus del sistema.

### III. DESARROLLO DE LA PRÁCTICA Y RESULTADOS

#### i. Lectura y envío del ADC

Se habilitaron los puertos E y D ya que, son puertos donde están la mayoría de los ADC:

| Pin Name | Pin Number | Pin Mux / Pin Assignment | Pin Type | Buffer Type <sup>a</sup> | Description                          |
|----------|------------|--------------------------|----------|--------------------------|--------------------------------------|
| AIN0     | 6          | PE3                      | I        | Analog                   | Analog-to-digital converter input 0. |
| AIN1     | 7          | PE2                      | I        | Analog                   | Analog-to-digital converter input 1. |
| AIN2     | 8          | PE1                      | I        | Analog                   | Analog-to-digital converter input 2. |
| AIN3     | 9          | PE0                      | I        | Analog                   | Analog-to-digital converter input 3. |
| AIN4     | 64         | PD3                      | I        | Analog                   | Analog-to-digital converter input 4. |
| AIN5     | 63         | PD2                      | I        | Analog                   | Analog-to-digital converter input 5. |
| AIN6     | 62         | PD1                      | I        | Analog                   | Analog-to-digital converter input 6. |
| AIN7     | 61         | PD0                      | I        | Analog                   | Analog-to-digital converter input 7. |
| AIN8     | 60         | PE5                      | I        | Analog                   | Analog-to-digital converter input 8. |

Figura 4. Señales del ADC.

Como las especificaciones indican leer 6 canales, se puso una constante para saber cuantos canales se leerían y así, si en un futuro (como en la práctica 5) se necesitan leer menos o más canales sería de una manera sencilla pues sólo se le movería el valor de esa constante y se leerían como en la figura 5:

```
7
8 #define canales 6 ///defino cuantos canales he de leer
9 uint16_t adc_data[canales] = {0}; // arreglos de enteros con 4 casillas
```

Figura 5. Definición de canales.

El arreglo `adc_data` que va ligado a los canales, indica que será un arreglo de 6 casillas y podrá guardar enteros en cada una.

Posteriormente, se habilitan las funciones a usar como la configuración del ADC y el UART.

```

ADC_config_modulo_0();

while(1){

    ADC_leer_canal(adc_data);
    // Delay();
    duty0 = adc_data[0] ; /// E3
    duty1 = adc_data[1] ; ///E2
    duty2 = adc_data[2] ; ///E1
    duty3 = adc_data[3] ; ///D3
    duty4 = adc_data[4] ; /// D2
    duty5 = adc_data[5] ; ///D1

```

Figura 6. Habilitación de funciones.

Se aprecia que cada casilla del arreglo (posterior a la lectura del ADC) se mete en una variable entera para poder hacer más fácil su manejo.

```

/////////comentar o descomentar para saber cuál graficar.
/* sprintf(data_str, "%04u", duty0);
   Tx_string(data_str); */
/* sprintf(data_str, "%04u", duty1);
   Tx_string(data_str); */
   sprintf(data_str, "%04u", duty2);
   Tx_string(data_str);
/* sprintf(data_str, "%04u", duty3);
   Tx_string(data_str); */
/* sprintf(data_str, "%04u", duty4);
   Tx_string(data_str); */
/* sprintf(data_str, "%04u", duty5);
   Tx_string(data_str); */

```

Figura 7. Arreglo de datos.

La computadora utilizada se trababa al intentar graficar todos los ADC a la vez así que, para obtener las 6 gráficas de distintos ADC se comentan y descomentan según se requiera. Para la gráfica se usó un script en matlab.

```

while i<=muestras

    if mod(i,100)==0;
        xlim([0 i+100]);
    end

    str_1=readline(puerto);

    ADC1= str2double(str_1);

    voltaje1(i)=(ADC1*3.3)/4095;

    plot(voltaje1,'r');
    drawnow

```

Figura 8. Script de habilitación.

Dicho script manejaba el habilitar y leer el puerto donde se encontraba nuestro UART para posterior graficación de datos gracias a la conversión de:  $(\text{valor\_ADC} \times \text{volataje\_máximo}) / \text{Valor\_máximo\_ADC}$  logrando dar gráficas parecidas a la siguiente:

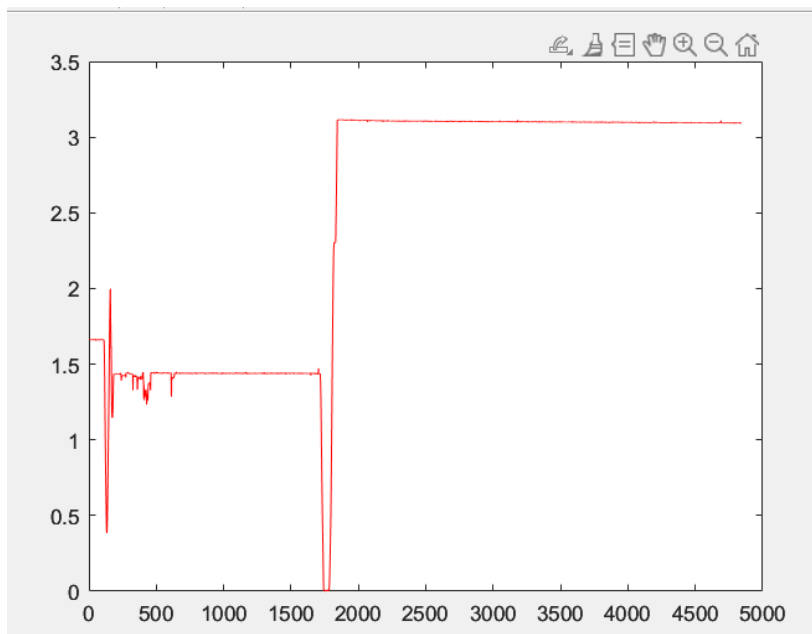


Figura 9. Gráfica obtenida.

#### IV. BIBLIOGRAFÍA

Texas Instruments Incorporated. (2007). *Tiva™ TM4C123GH6PM Microcontroller - Data Sheet*. -. Recuperado 20 de enero de 2023, de [https://www.ti.com/lit/ds/spms376e/spms376e.pdf?ts=1646079818207&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/spms376e/spms376e.pdf?ts=1646079818207&ref_url=https%253A%252F%252Fwww.google.com%252F)