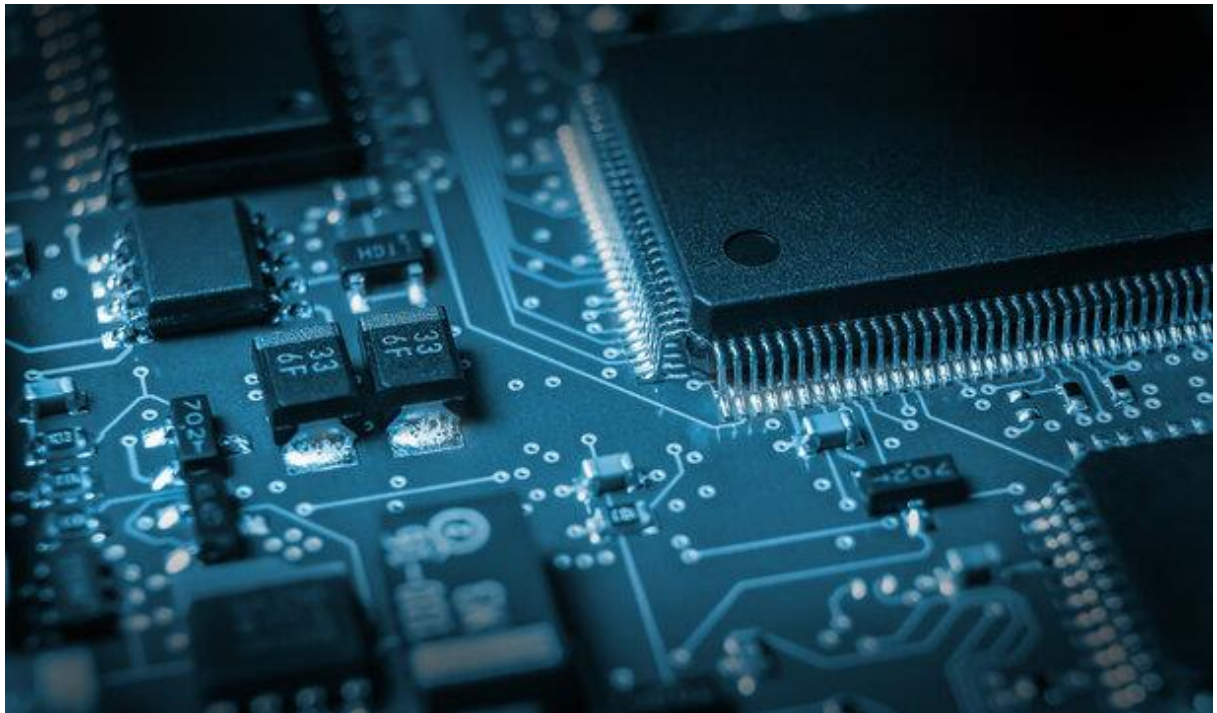




UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA

29 DE ENERO DE 2023



MICROCONTROLADORES

PRÁCTICA 4

➤ ING. JOSÉ DE JESÚS SANTANA RAMÍREZ.
+ ANGEL FLORES MORENO.

INGENIERÍA BIOMÉDICA.

GRUPO: 41



Introducción.

ADC

El ADC es un circuito integrado electrónico o dispositivo modular de uso común que convierte una señal analógica, generalmente un voltaje, en una serie de representaciones digitales discretas muestreadas, o en números. Los ADC realizan tres operaciones distintas: muestreo, cuantificación y codificación. Forman el corazón de muchos instrumentos digitales comunes, como voltímetros, osciloscopios y analizadores de espectro. También se incorporan en el front end de los circuitos digitales que procesan señales analógicas provenientes de dispositivos como micrófonos, acelerómetros, y otros transductores que necesitan convertir su salida al dominio digital para que un microprocesador pueda trabajar con los datos.

Hay muchas arquitecturas de ADC, o topologías, que se han desarrollado para muestrear y digitalizar señales analógicas. Cada forma del ADC tiene sus propias características, beneficios y debilidades. La elección de un tipo específico de ADC para una aplicación dada generalmente se define por los requisitos de medición de velocidad, resolución, precisión, consumo de energía y tamaño físico.

Características principales de un ADC

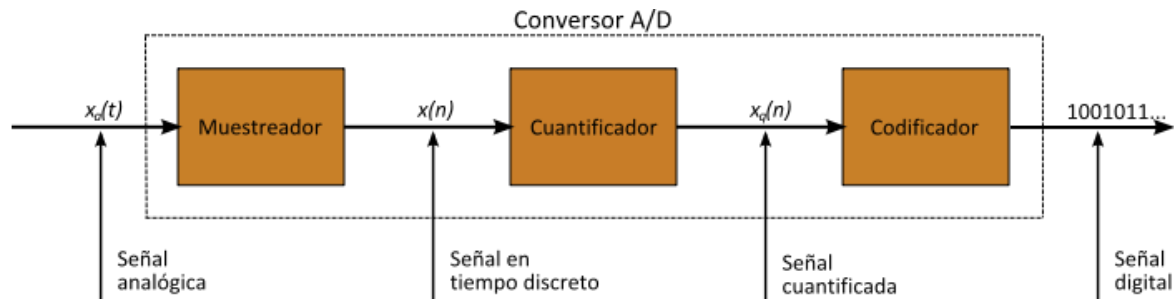
La primera operación que un ADC debe realizar es muestrear la señal analógica. El muestreo se realiza mediante un circuito de muestreo y retención o de seguimiento y retención. El muestreo o el teorema de Nyquist requiere que la velocidad de muestreo sea mayor que el doble del ancho de banda de la señal para poder reconstruir la señal analógica de las muestras digitalizadas. Por lo tanto, la primera característica significativa del ADC es la velocidad de muestreo, que determina el componente de frecuencia de señal máxima que se puede digitalizar.

El ADC debe cuantificar cada muestra y dividir el voltaje muestreado en un número finito de niveles de amplitud discretos. Esta característica generalmente se describe como el número de bits de resolución. Por ejemplo, si una señal se divide en 8 bits, significa que hay 28 o 256 niveles discretos. Un ADC de 16 bits el rango de voltaje en 65,536 niveles de cuantificación.

Tanto la resolución como la velocidad máxima de muestreo están determinadas por el hardware del ADC. En general, cuanto mayor es la resolución del ADC, más limitada es la velocidad máxima de muestreo.

La precisión de un ADC depende tanto de la resolución como de la velocidad de muestreo. La resolución afecta la precisión y la exactitud de la amplitud. Otros factores que afectan la precisión de la amplitud son la linealidad del proceso de cuantificación y los efectos del ruido vertical. La frecuencia de muestreo determina la precisión y la exactitud de la temporización.

El tamaño físico y el consumo de energía son otros atributos que dependen de la topología del circuito. Son las principales preocupaciones para las aplicaciones que tienen un tamaño o potencia disponible limitados, como dispositivos IoT o instrumentos portátiles que funcionan con baterías. El tamaño físico y el consumo de energía dependen en gran medida de la topología del ADC.



Objetivo.

Usando el módulo 0 y 1 , configurar la tarjeta a la frecuencia asignada, para adquirir 6 señales analógicas a una velocidad de 1 millón de muestras por segundo, por los canales asignados y guardar los valores en un arreglo para ser enviados con un botones externos asociado al gpio D a través del protocolo de comunicación asíncrona a una velocidad de 115200 todo esto usando interrupciones.

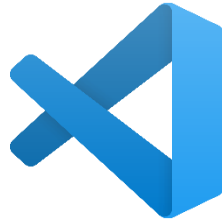
- A. 1,3,4,5,11,9- 55Mhz- 115200 -sec 2 ,sec 1
- B. 2,4,6,7,10,1- 30MHZ -57600 -sec3, sec1
- C. 3,5,7,4,2,8 70 MHZ 9600, -sec2, sec3, sec
- D. 11,9,6,7,4,2 33MHZ 19200 -sec3,sec0,sec2
- E. 1,5,7,11,6,8 80MHZ 4800 -sec0 ,sec3

la captura será en Matlab en 6 graficas.

Materiales y descripción.

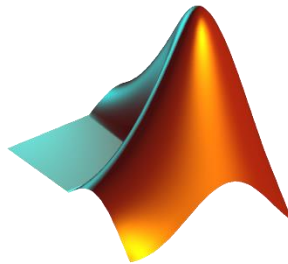
Visual studio code.

Visual Studio Code es un editor de código optimizado con soporte para operaciones de desarrollo como depuración, ejecución de tareas y control de versiones. Su objetivo es proporcionar las herramientas que un desarrollador necesita para un ciclo rápido de creación y depuración de código y deja los flujos de trabajo más complejos para los IDE con funciones más completas, como el IDE de Visual Studio.



Matlab.

MATLAB es una plataforma de programación y cálculo numérico utilizada por millones de ingenieros y científicos para analizar datos, desarrollar algoritmos y crear modelos.

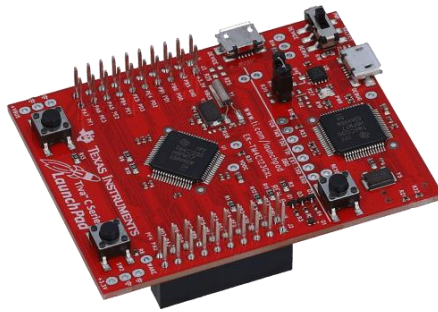


Tiva EK-TM4C123GXL.

El kit de evaluación LaunchPad TM4C123G es una plataforma de evaluación de bajo costo para microcontroladores basados en ARM Cortex-M4F de Texas Instruments. El diseño del TM4C123G LaunchPad destaca el microcontrolador TM4C123GH6PM con una interfaz de dispositivo USB 2.0 y un módulo de hibernación.

Características

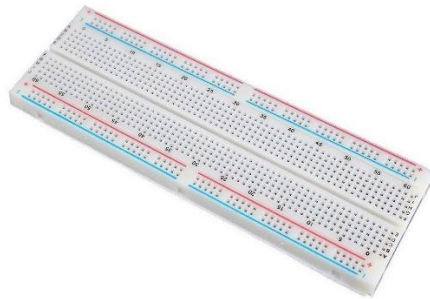
- MCU TM4C123GH6PM de alto rendimiento:
- CPU de 80 MHz 32 microcontroladores basados en ARM Cortex-M4
- 256KB Flash, 32KB SRAM, 2KB EEPROM
- Dos módulos de red de área del controlador (CAN)
- USB 2.0 Host / Dispositivo / OTG + PHY
- ADC 2MSPS de 12 bits dobles, PWM de control de movimiento
- 8 UART, 6 I2C, 4 SPI
- Interfaz de depuración en circuito (ICDI) a bordo
- Cable USB Micro-B a cable USB-A
- Aplicación de inicio rápido RGB precargada
- Guía de inicio rápido de ReadMe First



Protoboard

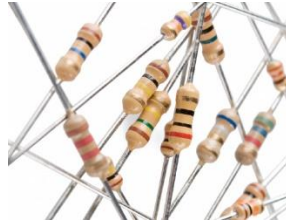
Una protoboard, o breadboard, es prácticamente una PCB temporal con una forma y tamaño generalizados. Utilizada comúnmente para pruebas y prototipos temporales de circuitos. Se usa insertando las terminales de los dispositivos electrónicos en los orificios de la protoboard de la forma en que tengan continuidad.

Una protoboard debe usarse meramente para hacer pruebas y prototipos temporales. Puesto que, aunque se pueden diseñar una infinidad de circuitos en ellas, estos circuitos no pueden ser muy grandes debido su espacio limitado. Sin embargo, varias protoboard se pueden unir si es que sus puntos de ensamblaje coinciden.



Resistencias.

La resistencia es un componente electrónico diseñado para causar una caída de tensión al flujo de electricidad en un punto dado, es decir. En otras palabras se opone al paso de la corriente en un circuito electrónico, su magnitud de resistencia depende de su cantidad de ohmio $[\Omega]$ (Unidad de medida de la resistencia).



Cable para proto.

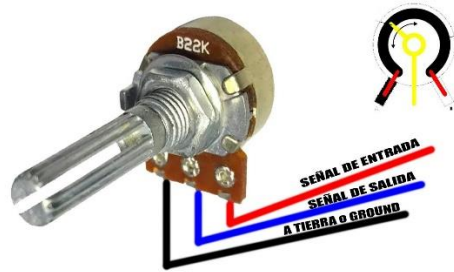
Un cable puente para prototipos (o simplemente puente para prototipos), es un cable con un conector en cada punta (o a veces sin ellos), que se usa normalmente para interconectar entre sí los componentes en una placa de pruebas.



Potenciómetros.

Un potenciómetro es un componente electrónico similar a los resistores pero cuyo valor de resistencia en vez de ser fijo es variable, permitiendo controlar la intensidad de corriente a lo largo de un circuito conectándolo en paralelo o la caída de tensión al conectarlo en serie. Un potenciómetro es un elemento muy similar a un reóstato, la diferencia es que este último disipa más potencia y es utilizado para circuitos de mayor corriente, debido a esta característica, por lo general los potenciómetros son generalmente usados para variar el voltaje en un circuito colocados en paralelo, mientras que los reóstatos se utilizan en serie para variar la corriente.

Un potenciómetro está compuesto por una resistencia de valor total constante a lo largo de la cual se mueve un cursor, que es un contacto móvil que divide la resistencia total en dos resistencias de valor variable y cuya suma es la resistencia total, por lo que al mover el cursor una aumenta y la otra disminuye. A la hora de conectar un potenciómetro, se puede utilizar el valor de su resistencia total o el de una de las resistencias variables ya que los potenciómetros tienen tres terminales, dos de ellos en los extremos de la resistencia total y otro unido al cursor.



Botón.

Un pulsador permite abrir o cerrar el circuito solo mientras estemos actuando sobre él. Cuando dejamos de presionar vuelve a su posición inicial. Pulsador normalmente abierto (NA): En el estado de reposo el circuito está abierto, y se cierra cuándo se presiona.



Descripción general del código.

Usando el módulo 0 y 1, se configuro la tarjeta a la frecuencia asignada de 55_MHz, para adquirir 6 señales analógicas a una velocidad de 1 millón de muestras por segundo, por los canales asignados y guardar los valores en un arreglo para ser enviados con un botón externo asociado al gpio D, utilizando el pin número 7 a través del protocolo de comunicación asíncrona a una velocidad de 115200 todo esto usando interrupciones. Utilizando el secuenciador 1 y 2 con los canales 1, 3, 4, 5, 11 y 9, los cuales en nuestra tiva fueron utilizados con los pines E2, B5, E0, B4, D2, D3. La gráfica utilizada que muestra los resultados se encuentra en Matlab utilizando comunicación por UART.



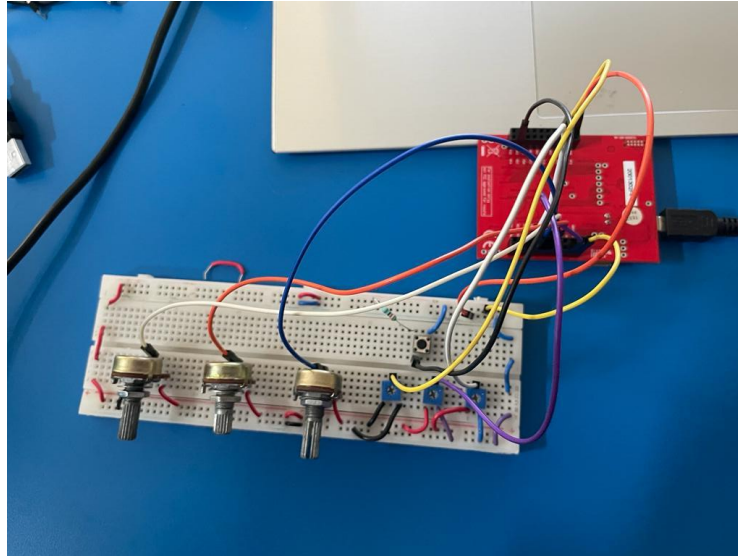
Evidencias y resultados.

Código elaborando en Matlab.

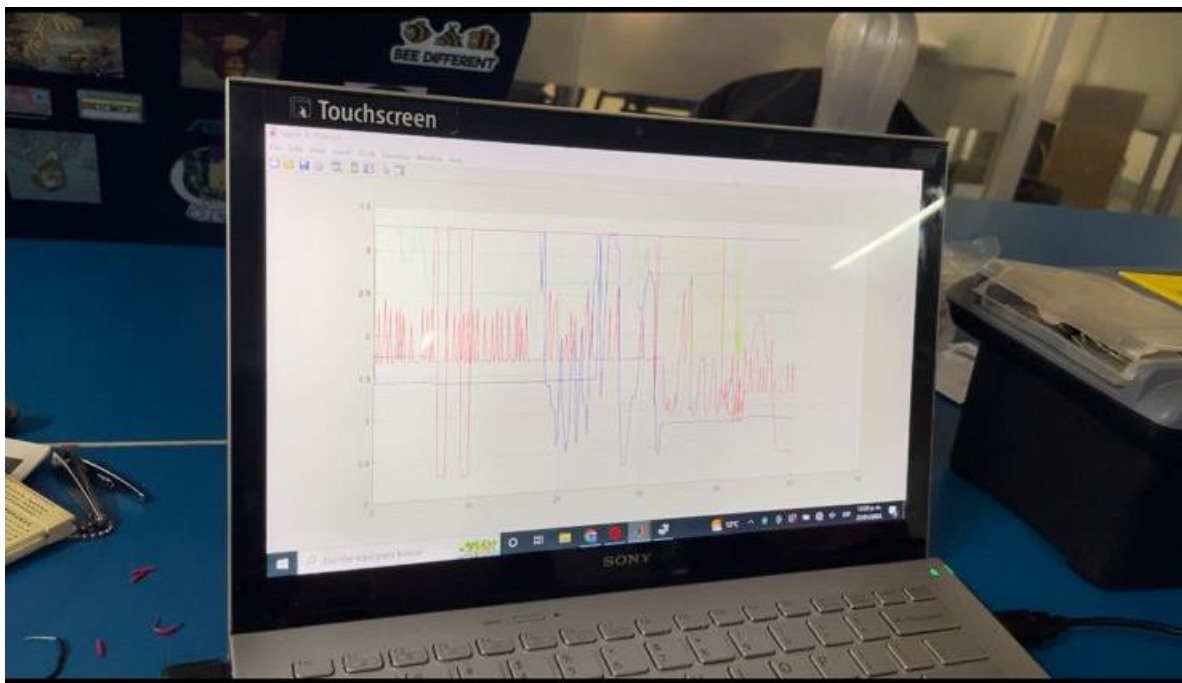
```
Editor - C:\Users\Angel\Desktop\micro\adcmat.m
adcmat.m x +
1 %% MICROCONTROLADORES
2 clear all
3 clc
4 warning('off','all')
5 UART0 = serialport("COM4",115200,"Timeout",60)
6 flush(UART0)
7 i = 1;
8 time = 0;
9 tmax = 249;
10
11 figure('Name','TIVA123')
12 l1=line(nan,nan,'Color','red');
13 l2=line(nan,nan,'Color','magenta');
14 l3=line(nan,nan,'Color','black');
15 l4=line(nan,nan,'Color','green');
16 l5=line(nan,nan,'Color','cyan');
17 l6=line(nan,nan,'Color','blue');
18 %l7=line(nan,nan,'Color','cyan');
19 %l8=line(nan,nan,'Color','magenta');
20 %l9=line(nan,nan,'Color','yellow');
21 % axes('XLim',[0 100],'YLim',[0 3.3]);
22 grid on; hold on
23 tic
24
```

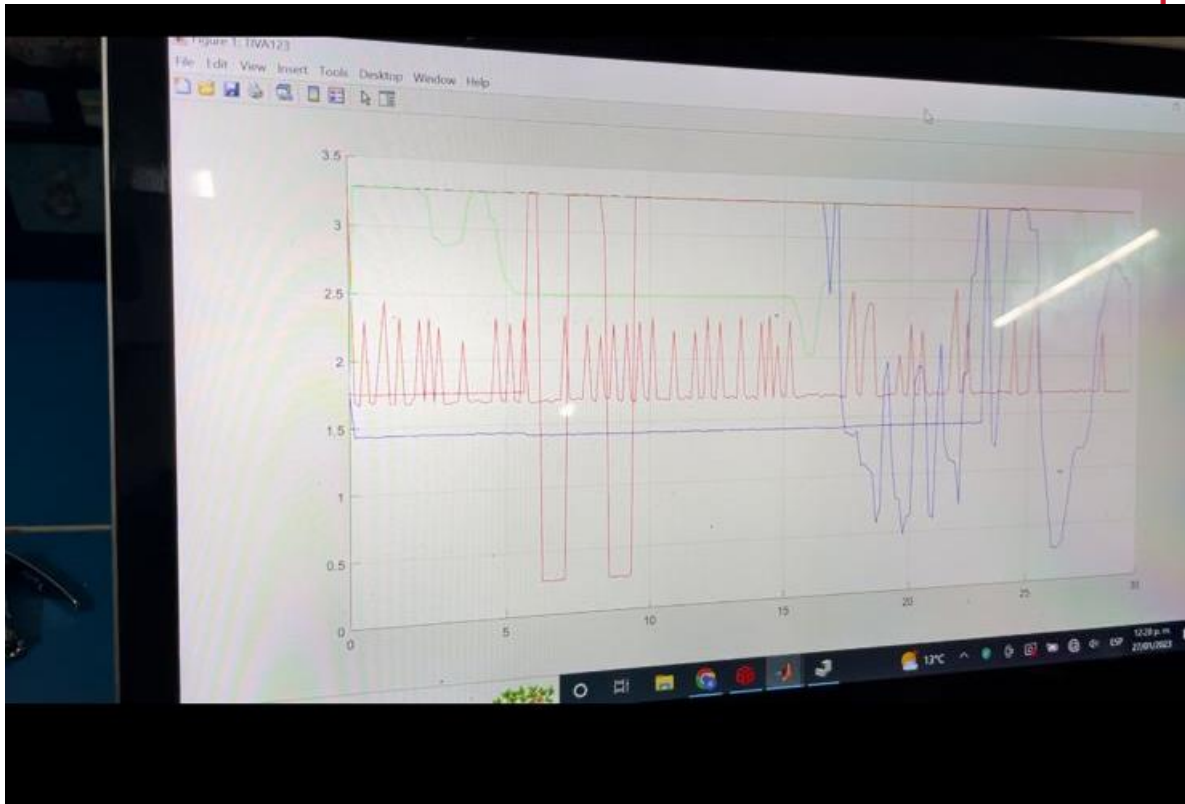
```
Editor - C:\Users\Angel\Desktop\micro\adcmat.m
adcmat.m x +
22 grid on; hold on
23 tic
24
25 while(time<tmax)
26     BT = readline(UART0);
27     a=split(BT,',');
28     b = (str2double(a).*3.3)./4095.0;
29     v1(i)=b(1); v2(i)=b(2); v3(i)=b(3); v4(i)=b(4); v5(i)=b(5); v6(i)=b(6);
30     %v7(i)=b(7); v8(i)=b(8); v9(i)=b(9);
31     time=toc
32     x=linspace(0,time,i);
33     set(l1,'YData',v1(1:i),'XData',x);
34     set(l2,'YData',v2(1:i),'XData',x);
35     set(l3,'YData',v3(1:i),'XData',x);
36     set(l4,'YData',v4(1:i),'XData',x);
37     set(l5,'YData',v5(1:i),'XData',x);
38     set(l6,'YData',v6(1:i),'XData',x);
39     %set(l7,'YData',v7(1:i),'XData',x);
40     %set(l8,'YData',v8(1:i),'XData',x);
41     %set(l9,'YData',v9(1:i),'XData',x);
42     hold on;
43     drawnow;
44     i = i+1;
45 end
```


Circuito elaborado para el funcionamiento de nuestra práctica, utilizamos 6 potenciómetros para nuestras 6 señales y un botón que nos fue dando los cambios generados en nuestra grafica.



Gráficas proyectadas en matlab las cuales son las señales de nuestro programa y se ven reflejadas dependiendo la posición de de la manipulación de nuestros potenciómetros.





Conclusión.

En esta práctica se aprendió a utilizar el adc, anteriormente ya se había trabajado con adc en materias pasadas, pero nunca habíamos trabajado en un microcontrolador como la tiva, se aprendió a utilizar los registros del adc y gracias al programa de Matlab y la comunicación con uart pudimos ver reflejado el adc a través de la manipulación de unos potenciómetros en el protoboard, fue una practica compleja pero muy interesante.

Bibliografía.

- <https://www.digikey.com.mx/es/articles/match-the-right-adc-to-the-application>
- <https://code.visualstudio.com/docs/supporting/FAQ>
- <https://es.mathworks.com/products/matlab.html>