

ESCUELA POLITÉCNICA NACIONAL

INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

PROCESAMIENTO DIGITAL DE SEÑALES

ALUMNOS: Marialejandra Contreras

Melanny Dávila

SEMESTRE: 2020B

PARALELO: GR1

FECHA: 23/12/2020

PROFESOR: DR. ROBIN ÁLVAREZ

TEMA: Generación y adquisición de onda mediante hardware – Arduino PARTE 3

**PARTE 3:** **GENERADOR** **SECUENCIAL** CON ARDUINO, **MEDICIÓN** CON ARDUINO+MATLAB Y **ALMACENAMIENTO** EN ARCHIVO MEDIANTE MATLAB.

Repita la parte 2 pero ahora la verificación de las ondas debe hacerlo desde Matlab (en un GUI) a través del Arduino y no mediante el puerto de audio.

**NOTA**: Los informes deben estar en documentos de Word separados y desarrollados en formato paso a paso con lujo de detalles. Los códigos deben estar comentados y realizados también en partes. Al final debe adjuntar el código completo.

La programación de GUI de MatLab es el siguiente

function varargout = ArduinoGUIv2(varargin)

% ARDUINOGUIV2 MATLAB code for ArduinoGUIv2.fig

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help ArduinoGUIv2

% Last Modified by GUIDE v2.5 23-Dec-2020 15:14:15

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @ArduinoGUIv2\_OpeningFcn, ...

'gui\_OutputFcn', @ArduinoGUIv2\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before ArduinoGUIv2 is made visible.

function ArduinoGUIv2\_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for ArduinoGUIv2

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

clear all; clc; % borrar todas las variables

delete(instrfind({'Port'},{'COM6'}));% cierra las conexiones en el puerto de comunicación a utilizar

clear a;

global a;

a= arduino('COM6','Uno'); % creación del objeto Arduino, uso una placa Uno para la lectura

% UIWAIT makes ArduinoGUIv2 wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = ArduinoGUIv2\_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in btn\_leer.

function btn\_leer\_Callback(hObject, eventdata, handles)

global a b% declaración de variables

b=1;

lecturas=100; % lecturas a realizar

almacenamiento=zeros(1, lecturas); % almacenamiento de datos

x=0;

while b==1 % lazo que permite que la grafica se mueva a medida que se van leyendo los valores

x=x+1;

AllValues=0;

axes(handles.axes1);

for i=1: lecturas

almacenamiento(i)=readVoltage(a,'A0');% lectura analogica en el pin A0 del arduino Uno

AllValues=almacenamiento(i)+AllValues;

plot(handles.axes1,almacenamiento)

ylim(handles.axes1,[0 3.5])

grid on

end

fileID = fopen('senial.txt','w+'); % creación del archivo donde se guardan los valores de la grafica

fprintf(fileID,'%f\n',almacenamiento);

end

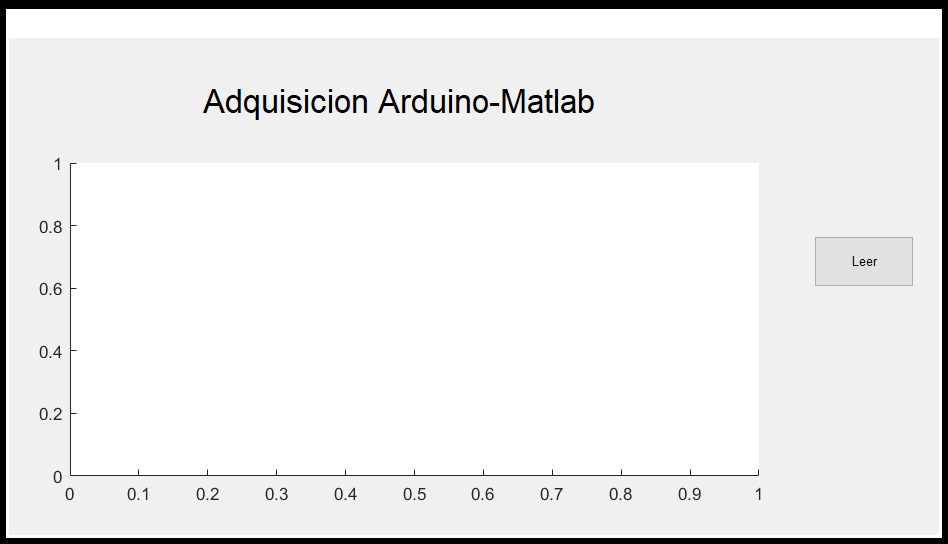
****

Figura 1. Guide-Matlab.

A continuación se presentan los resultados obtenidos a distintas frecuencias. El cambio de frecuencias se hace a traves del código programado al Arduino Due.

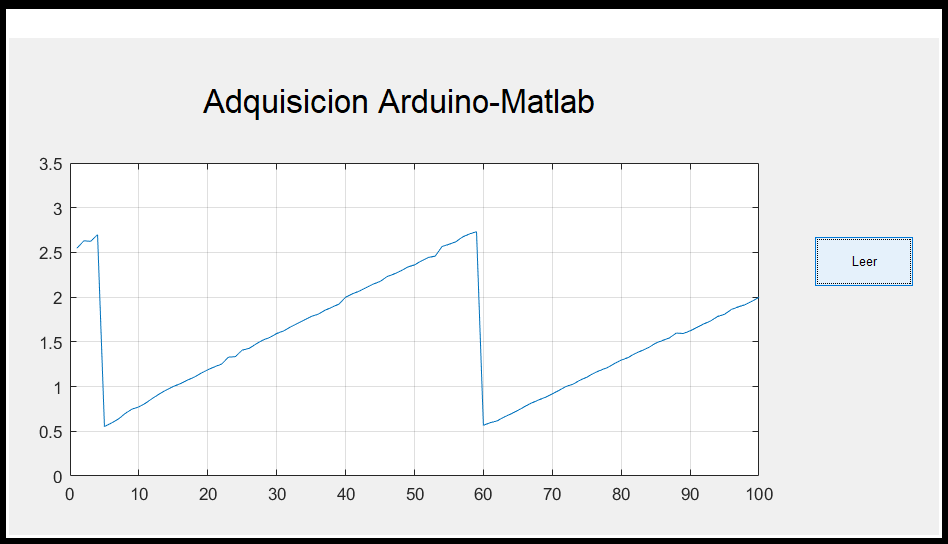


Figura 2. Onda diente de sierra a 1 Hz.

Esta primera frecuencia es aceptaba sin problemas puesto que se grafica correctamente.

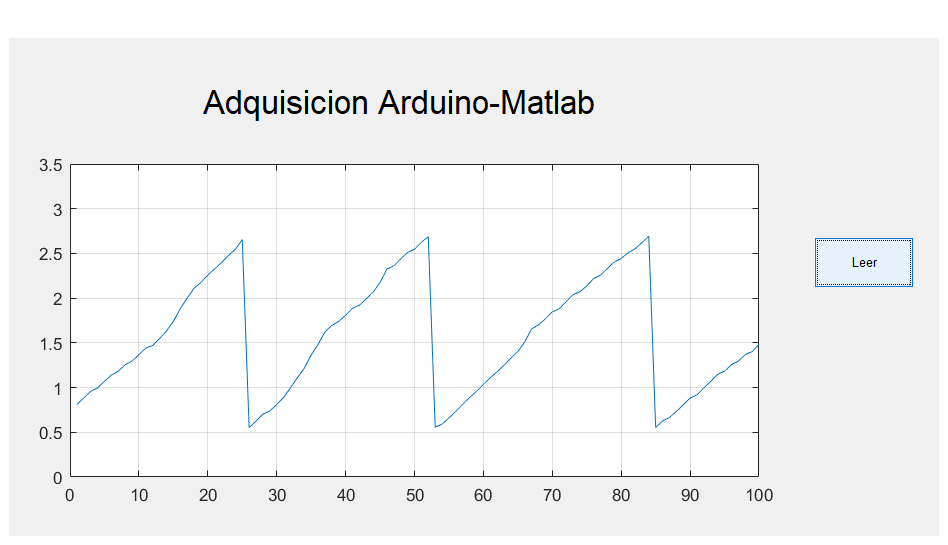


Figura 3. Onda diente de sierra a 2 Hz.

También es una frecuencia valida porque su grafica es bastante aceptable.

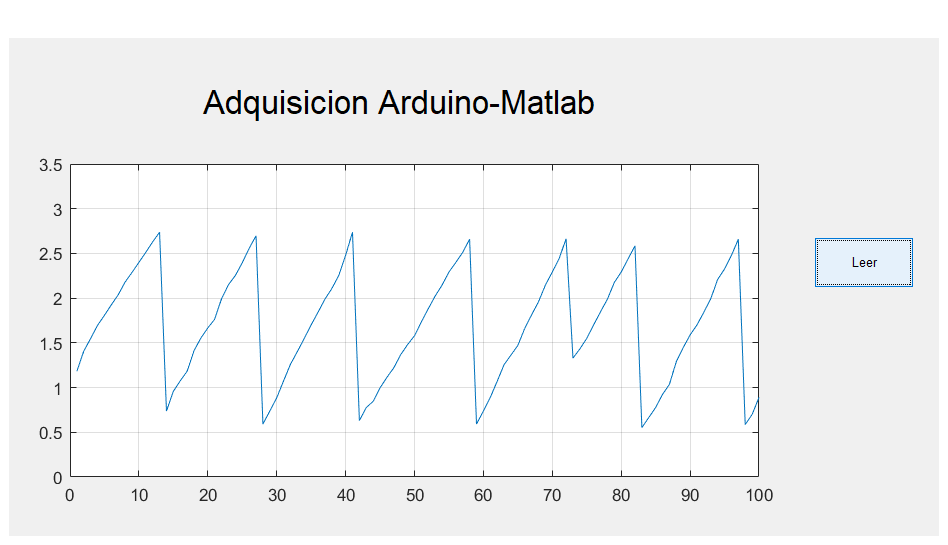


Figura 4. Onda diente de sierra a 4 Hz.

En 4 Hz, se puede empezar a apreciar ciertas deformaciones en la onda diente de sierra, esto podría generar amónicos distintos a los esperados. Se concluye que no es una frecuencia útil.

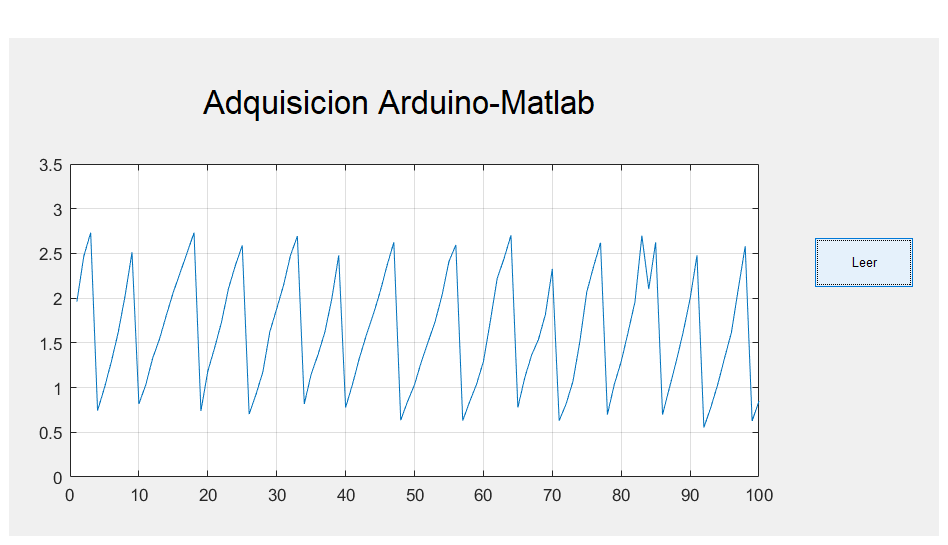


Figura 5. Onda diente de sierra a 8 Hz.

Como ultimo resultado, presentamos la onda a una frecuencia de 8 Hz, se puede observar que esta completamente deformada. Por lo tanto, podemos concluir que a frecuencias mayores a estas, el fenómeno de aliasing será dominante por lo que la señal seria solo ruido.

¿Cuál es la frecuencia máxima que puede ser adquirida correctamente desde Matlab?

El rango en el que la señal se grafica adecuadamente es de 0,9 hasta unos 2,5 Hz, aproximadamente.

Realice una discusión de las razones por las que se tiene esta limitación y cómo podría ser mejorado este resultado.

Arduino tiene sus limitaciones en cuanto a su respuesta en tiempo real y su rendimiento de cálculo. Por ello, lo que se puede hacer es establecer una comunicación por el puerto USB serial que recoja los datos, sean procesados y devueltos a la placa para ejecutar otra serie de acciones.

Link del video del programa:

<https://epnecuador-my.sharepoint.com/:v:/g/personal/marialejandra_contreras_epn_edu_ec/EeFCa_YafgJHmKpD-jLJKp8BednIa8Ct9rjKvNwhM2cVKQ?e=73EDWO>