

ESCUELA POLITÉCNICA NACIONAL



PROCESAMIENTO DIGITAL DE SEÑALES

PROYECTO #1

PAMELA CONDOY
PATRICIO YEPEZ

JUNIO, 12 del 2017

QUITO - ECUADOR

PROYECTO #1

1. OBJETIVOS

Adquirir señales de electromiograma por medio de un puerto de audio, con ellos observar la señal en el momento en que se abre y se cierra la mano.

2. MATERIALES

- Sensor de presión.

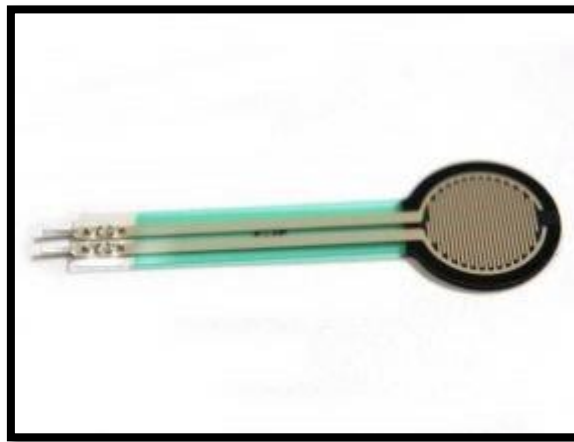


Figura Nº 1: Sensor de presión

- Arduino UNO

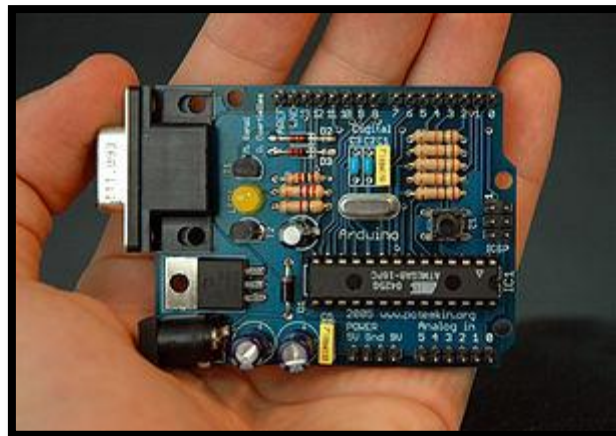


Figura Nº 2: Arduino UNO

- Resistencia de 1Mohmios, realizamos una conexión en serie con el arduino.
- Cable UTP

3. SOFTWARE

- Vamos a la pestaña home y seleccionamos Add-Ons e instalamos la librería de Arduino UNO en Matlab R2015a o R2015b,

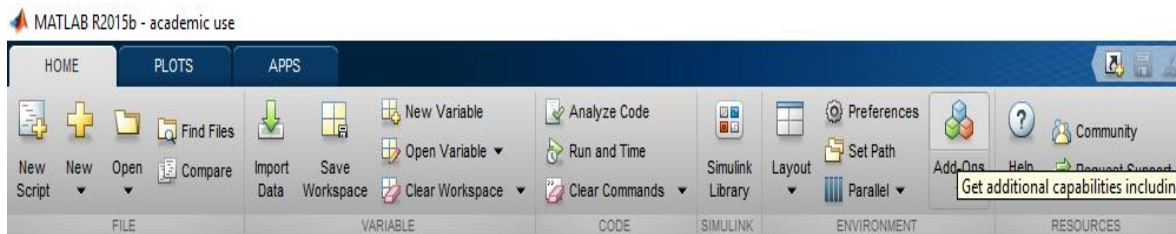


Figura N° 3: Home de MATLAB

- Aquí seleccionamos Get Hardware Support Packages.

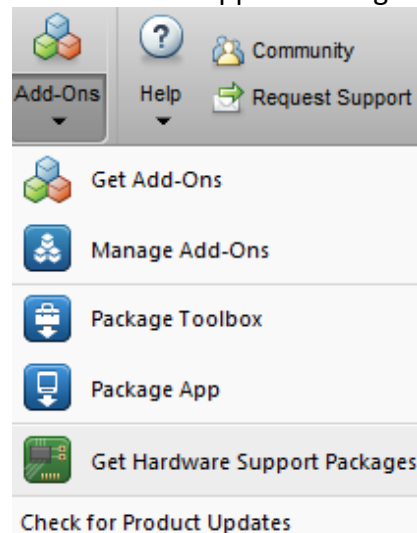


Figura N° 4: Add-Ons

- A continuación se observa la siguiente ventana y ponemos next:

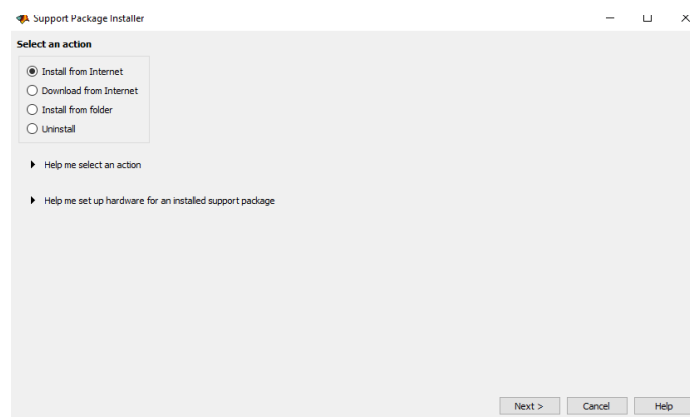


Figura N° 5: Support Package Installer

- En esta ventana escogemos la opción Arduino.

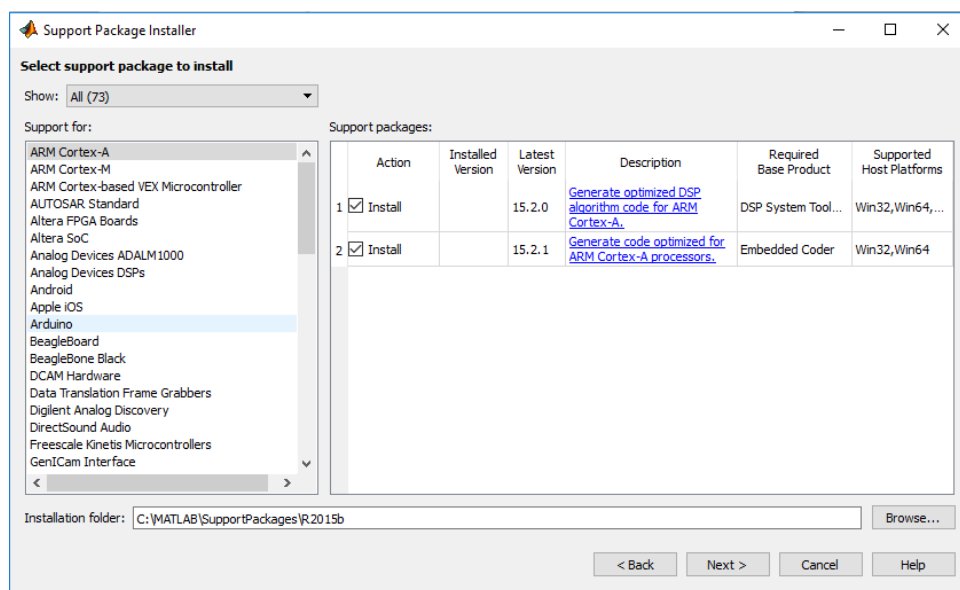


Figura N° 6: Seleccionar el paquete e instalar

- Presionamos next y se nos muestra la siguiente pantalla:

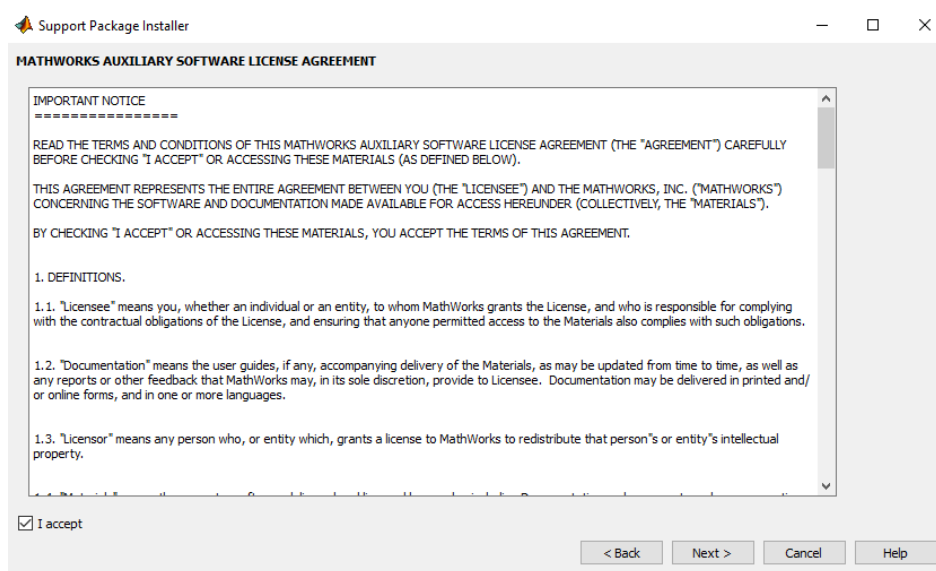


Figura N°7: Agreement

- Una vez realizada la instalación se podrá visualizar lo siguiente:

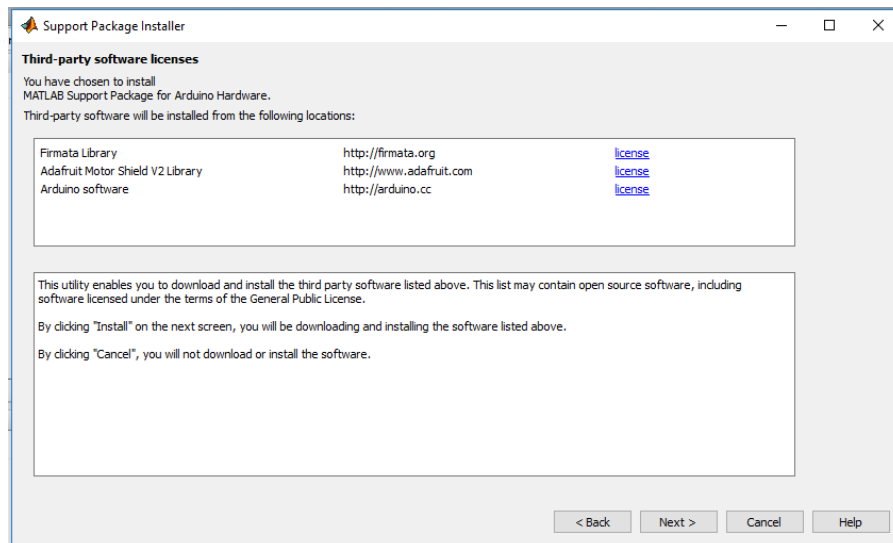


Figura N°8: Paquetes de Arduino a Instalar

- Presionamos next e inicia la instalación.

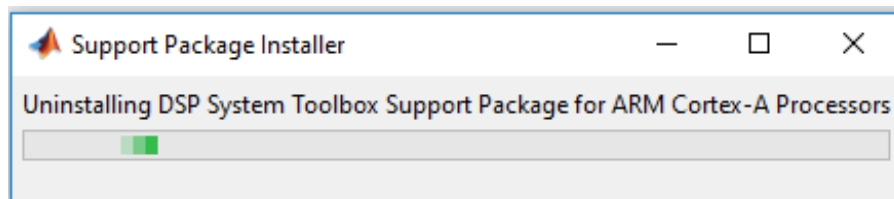


Figura N° 9: Instalación

- Continuamos con la instalación y se muestra la siguiente pantalla:

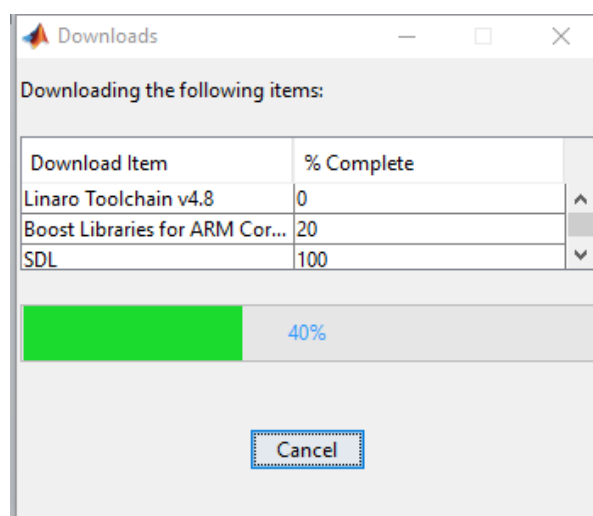


Figura N°10: Descarga

- Y finalmente una vez que se haya instalado to ponemos finish.

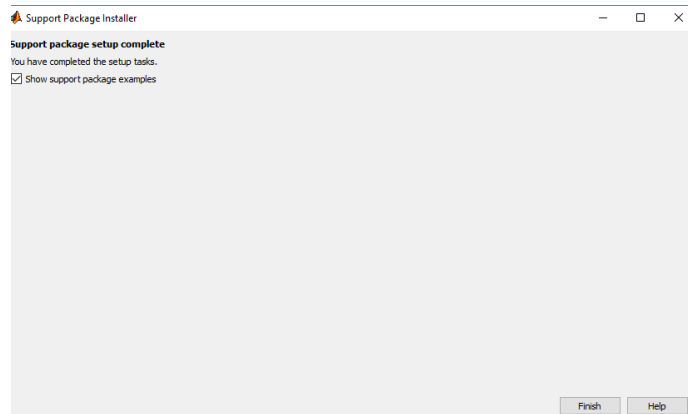


Figura N°11:Finaliza

- 3.1.** Una vez realizado todo el proceso anterior, procedemos a poner en el antebrazo el sensor como se observa en la siguiente gráfica:

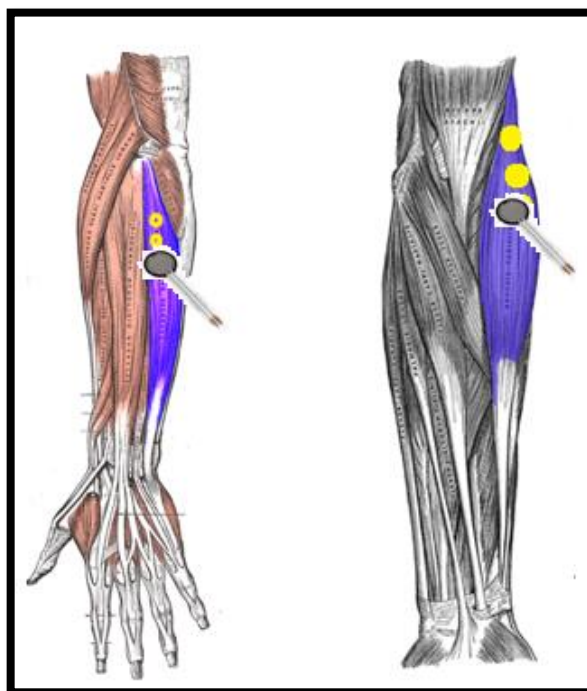


Figura N°12:Grafico Brazo y Sensor

Con ellos deberá mostrar en GUI tanto las gráficas de la señal abierta y cerrada la mano.

4. CÓDIGO

- MENU

```
function varargout = menu(varargin)

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @menu_OpeningFcn, ...
                  'gui_OutputFcn',  @menu_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

% --- Executes just before menu is made visible.
function menu_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = menu_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;
%imagen epn
axes(handles.logo1);
imagen=imread('LOGO_EPN.jpg');
imshow(imagen);

%imagen antena
axes(handles.logo2);
imagen1=imread('mano.png');
imshow(imagen1);

%imagen aeie
axes(handles.logo3);
imagen2=imread('AEIE.jpg');
imshow(imagen2);

% --- Executes on button press in valores.
function valores_Callback(hObject, eventdata, handles)
close
Electromiograma;

% --- Executes on button press in pushbutton3.
```

```
function pushbutton3_Callback(hObject, eventdata, handles)

opc=questdlg('Desea Salir?', 'SALIR', 'SI', 'NO', 'NO');
    if strcmp(opc, 'NO')
        return
    end
clear, clc, close all;
```

• ELECTROMIOGRAMA

```
function varargout = Electromiograma(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Electromiograma_OpeningFcn, ...
                  'gui_OutputFcn',  @Electromiograma_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function Electromiograma_OpeningFcn(hObject, eventdata,
handles, varargin)
global abierta
global punio
global mano_abierta
global mano_cerrada

abierta= imread ('abierta.jpg');
punio= imread ('punio.jpg');
mano_abierta=imread ('mano_abierta.jpg');
mano_cerrada=imread ('mano_cerrada.jpg');

axes(handles.grafico);
imshow(abierta);

axes(handles.caratula);
imshow(mano_abierta);

% Choose default command line output for Electromiograma
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
```



```

function varargout = Electromiograma_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in inicio.
function inicio_Callback(hObject, eventdata, handles)

global abierta
global punio
global mano_cerrada
global mano_abierta

senal=zeros(1,1000);
A=arduino('COM11','Uno')
inicio=true
while inicio
    senal=zeros(1,200);
    for numero=1:200
        acele=readVoltage(A,'A0');
        if acele>=0.45

            axes(handles.caratula);
            imshow(mano_cerrada);
            axes(handles.grafico);
            imshow(punio);

        else
            axes(handles.caratula);
            imshow(mano_abierta);
            axes(handles.grafico);
            imshow(abierta);
        end
        senal(numero)=acele;
        axes(handles.senal)
        plot(senal)
        drawnow
        if ~inicio
            break
        end
    end
end
end

clear A

function salir_Callback(hObject, eventdata, handles)

opc=questdlg('Desea Salir?', 'SALIR', 'SI', 'NO', 'NO');
if strcmp(opc, 'NO')
    return
end
clear, clc, close all;

```

- **GRAFICA_SEÑAL**

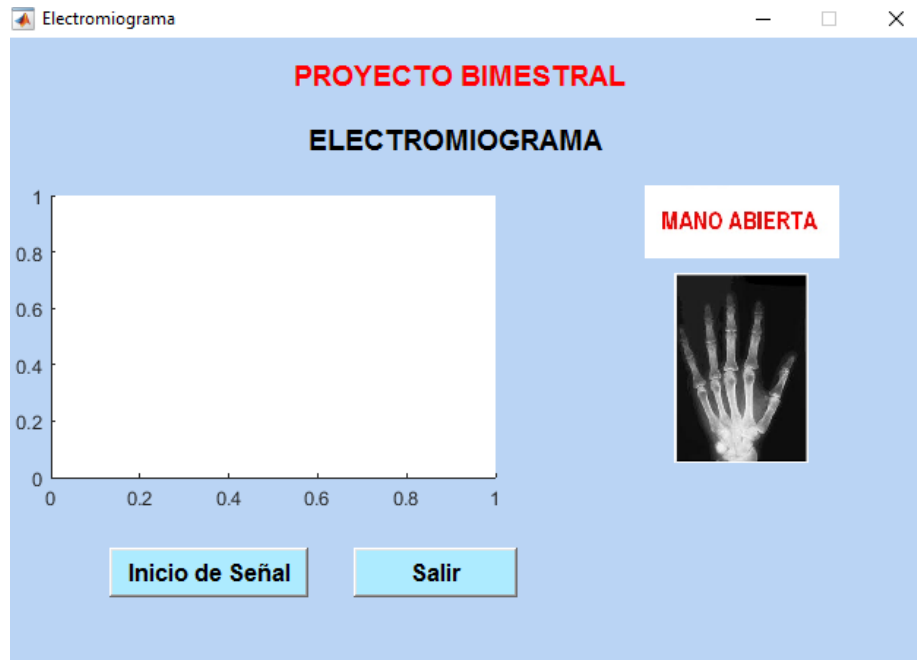
```
clc
close all
clear all
senal=zeros(1,1000);
a=arduino('COM10','Uno');
figure
for i=1:5
    senal=zeros(1,100);
    for numero=1:500
        emg=readVoltage(a,'A0')
        senal(numero)=emg;
        plot(senal)
        title('Electromiograma')
        xlabel('Tiempo')
        ylabel('señal')
        drawnow
    end
end
clear a
```

4.1. VENTANAS

- MENÚ



- ELECTROMIOGRAMA**



En ejecución:

