UNIVERSITY OF MONTENEGRO

FACULTY OF ELECTRICAL ENGINEERING

COURSE:MEDICAL ELECTRONICS

THEME: *The first laboratory exercise*-ECG registration, display and recognition

The second laboratory exercise-EMG registration, display and recognition

Mentor :

Prof. dr Radovan Stojanović

Students :

Petar Planinić
Luka Vulanović
Jovana Ćipranić
Amar Bralić
Ajka Ćatović

Date and place: **18/04/2022 Podgorica**

Contents

Introduction

This is a laboratory project, made in MEDEL electronics laboratory at the University of Montenegro, with mentoring of prof. dr Radovan Stojanović. The main theme is to registrate, display and recognite ECG and EMG signal.
We used Arduino Ide and Matlab to solve the tasks.
Components and supplies:

- Arduino Uno
- Breadboard
- Jumper wires x6
- Analog Vernier connector x2
- Electrodes x3
- ECG Vernier Sensor
- Dynamometer (Vernier)

The first laboratory exercise

Hardware solution

We implemented the SENSOR-ARDUINO-MATLAB system by adding a connector to which we connect the sensor. The signal sensor is connected to A0. To the connector are connected both + 5V and ground.And finally we connected the ECG sensor to the Arduino(Figure 1.1). The electrodes are attached to the student's body(Figure 1.2).

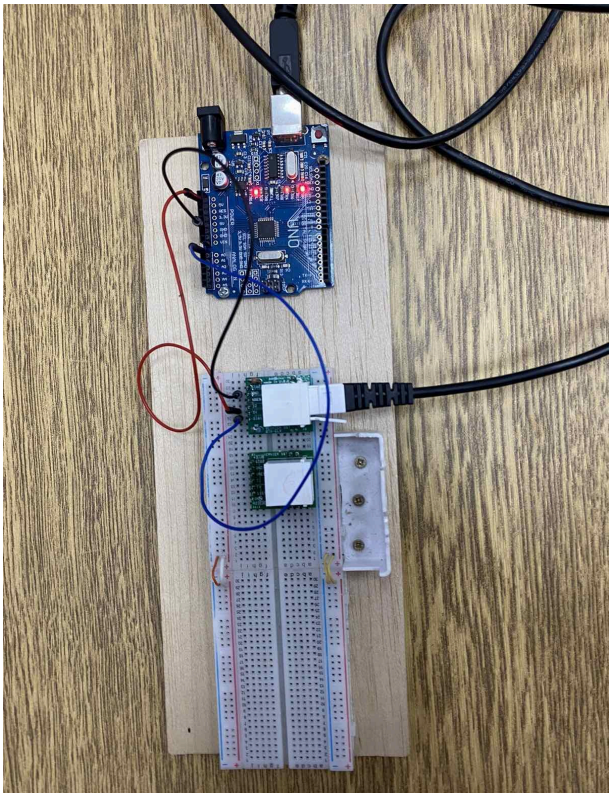

Figure 1.1

Figure 1.2

Software solution

As a software solution we used Arduino Uno, an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. Source code is written in Arduino IDE.
We created a program in Arduino that selects an ECG signal with a frequency of 200Hz (5ms) and sends it according to its terminal emulator (plotter), on a PC, with a speed of 57600 bps.Buffer signal width is 512.This signal is sent to MATLAB with a speed of 57600 bps in a format suitable for Matlab.Next.We created a program that accepts the frame and displays it in graphical form.

The next paragraph represents the Arduino code:

```
//ARDUINO ECG 1 sends samples from A0 A/D (in range 0-1023 to serial port by speed 5760 bps

int sensorValue;

void setup() // run once, when the sketch starts

{

Serial.begin(57600); // set up Serial library at 19200 bps

}

void loop() // loop

{

sensorValue = analogRead(A0);

Serial.println(sensorValue);

delay(5); // 200Hz sampling freq, delay 5ms , 1/0.005 = 200hz

}
```

Result from Arduino plotter(Figure 1.3)

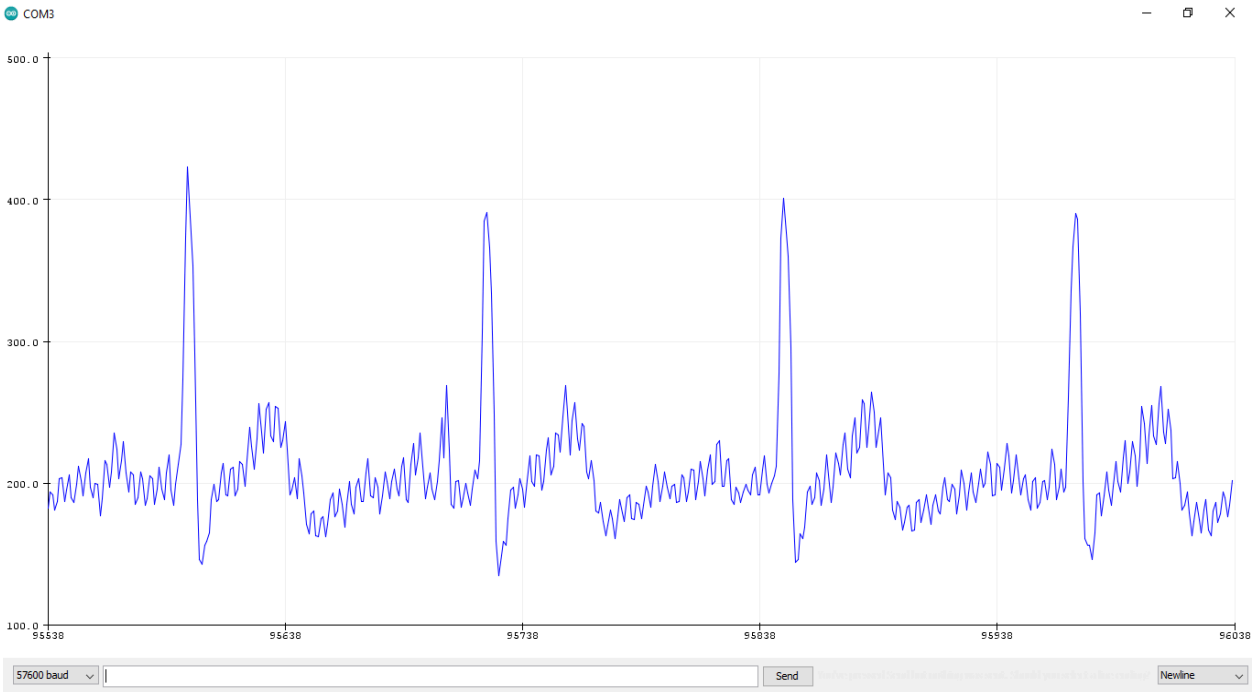

Figure 1.3

Here is the code for the Arduino when the signal is sent to MATLAB with a speed of 57600 bps in a format suitable for Matlab:

```
//ARDUINO ECG, send 512 samples in the frame, which ends with CR/LF
#define LEN 512 //set number of samples in frame
#define FREQ 200 //set sampling frequency
int T; //sampling period
int sensorValue; //value od A/D read
void setup() // run once, when the sketch starts
{
T=1000/((int)(FREQ)); //set sampling period
Serial.begin(57600); // set up Serial library at 57600 bps


}
void loop() // loop
{
int i=0;


for(i=0; i<LEN; i++){ //in loop of the length LEN
sensorValue = analogRead(A0); //read A/D
Serial.print(String(sensorValue)+" "); //print value
delay(T); // delay T
}
Serial.println(); // put at the end of frame CR/LF sign
}
```

Matlab Code:

```
function ex1(op)
global COMM_P % The serial port object
global COMM_P_ON % Serial port on or off
global h % A structure containing handles to the GUI controls, axis, etc
global fs
global port
global baud
global buffer
global buffer_y
global PAUSE
global file_counter
fs=200;
port='COM3';
baud=57600;
buffer=32000;
if nargin == 0 % if no input argument, draw the GUI
 op = 0;
end
switch op
case 0
 COMM_P_ON=0;
ss=strcat('OSC1.2','-',num2str(fs),':',port,':',num2str(baud),':',num2str(buffer));
set(0,'DefaultFigurePosition',[6 6 728 412]); %possition of GUI
%GUI-Figure
h.fig = figure ('NumberTitle','off','Name',...
ss,...
'Resize','on','Pointer','watch','Color',[0.5 0.5 0.5],...
'InvertHardcopy','off','CloseRequestFcn', 'ex1(3)');
%GUI-INIT button
h.ctrl(8)=uicontrol('Style','Pushbutton','BackgroundColor',[0.6 0.6 0.6],'Position',...
[10 280 60 25],'String','>>','Enable','on','Callback','ex1(7)');
h.ctrl(7)=uicontrol('Style','Pushbutton','BackgroundColor',[0.6 0.6 0.6],'Position',...
[10 250 60 25],'String','[]','Enable','on','Callback','ex1(6)');
h.ctrl(6)=uicontrol('Style','Pushbutton','BackgroundColor',[0.6 0.6 0.6],'Position',...
[10 50 60 25],'String','CLEAR','Enable','on','Callback','ex1(5)');
h.ctrl(1)=uicontrol('Style','Pushbutton','BackgroundColor',[0.6 0.6 0.6],'Position',...
[10 80 60 25],'String','INIT','Enable','on','Callback','ex1(1)');
h.ctrl(2)=uicontrol('Style','Pushbutton','BackgroundColor',[0.6 0.6 0.6],'Position',...
[10 110 60 25],'String','SAVE','Enable','on','Callback','ex1(4)');
set(gcf,'Pointer','arrow')
%GUI-Status text box
h.ctrl(3) = uicontrol('Style','text','Position',[10 160 60 25],'String','Ready...',...
```

```
'HorizontalAlignment','left','FontSize',8);
h.ctrl(4) = uicontrol('Style','text','Position',[10 5 320 25],'BackgroundColor',[0.8 0.8
0.8],'String','Buffer...',...
'HorizontalAlignment','left','FontSize',10);
h.ctrl(5) = uicontrol('Style','text','Position',[10 220 60 25],'String','File...',...
'HorizontalAlignment','left','FontSize',8);
 case 1 %if INIT button is pressed
 if(COMM_P_ON==0) %if serial port is closed
 COMM_P=serial(port,'BaudRate',baud,'InputBufferSize',buffer); %define
 fopen(COMM_P); %open
 COMM_P.BytesAvailableFcnMode = 'terminator'; %on terminator
 COMM_P.BytesAvailableFcn = 'ex1(2)'; %go to ex1(2), call back function
 COMM_P_ON=1; %set flag
 fprintf(COMM_P,'a'); %request from MC to send frame
 end
 answer=COMM_P.status; %display port status
 if strcmp(answer,'open')
 set(h.ctrl(3),'String','CONECTED')
 file_counter=0;
 else
 set(h.ctrl(3),'String','COMM ERR')
 end
 case 2 %Callback function on "terminator"
 pp=fscanf(COMM_P); %read port's FIFO characters
 y=str2num(pp); %convert in numbers
 len_all=length(y); %find length
 x=[1:1:len_all]/fs; %intialise x
 if len_all>0 % if length > of some character
 if(PAUSE==0)
 %PUT PROCESSING ALGORITHMS
 subplot(311) %ploting
 plot(x,y,'red');
 xlabel('[s]');
 ylabel('Ch1[V]');
 grid minor

 %PROCESSING
 [B,A] = butter(3,0.1,'low');
 y1=filter(B,A,y);
 subplot(312) %ploting
 plot(x,y1,'green');
 xlabel('[s]');
 ylabel('Ch1-processed[V]');
 grid minor
```

```
buffer_y=[buffer_y,y1]; %

subplot(313) %ploting
plot(buffer_y,'blue');
xlabel('[s]');
ylabel('All signal');
grid minor
end
end
case 3 %dialog box for closing GUI
answer = questdlg('Really close PPG','','Yes','No','No');
if strcmp(answer,'Yes')
if COMM_P_ON==1
fprintf(COMM_P,'s'); %stop sending
fclose(COMM_P);
PAUSE=0;
end
clear all;
delete(gcf);
end

case 4
file_name=strcat('MEDLog',num2str(file_counter),'.txt');
fid=fopen(file_name,'wt');
fprintf(fid, '%f\r\n', fs);
fprintf(fid, '%f\r\n', buffer_y');
fclose(fid);
file_counter=file_counter+1;
set(h.ctrl(5),'String',file_name)
case 5
buffer_y=[];
case 6
PAUSE=1;
case 7
PAUSE=0;

end
```

Result from the Matlab(Figure 1.4)

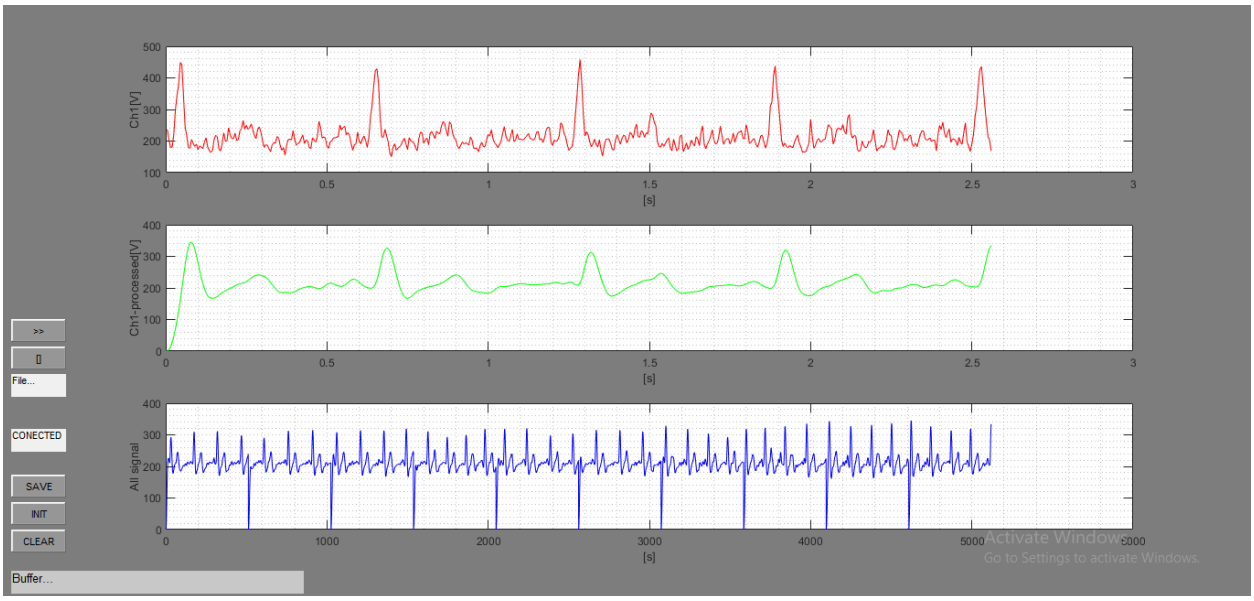

Figure 1.4

When bioelectricity signals such as ECG (Electrocardiograph)are abstracted, they are mixed with plenty of other signals. We call the other signals interfering signals. The main interfering signal is 50Hz electromagnetic signal, because it exists in all kinds of bioelectricity signals and its intensity is greatly more than bioelectricity signals. So it is necessary for us to filter 50Hz interfering signal. To solve a problem, we designed a **notch filter** with a frequency range of about 50Hz in Matlab, width 5Hz or 10Hz.

The second laboratory exercise

Hardware solution

We implemented the SENSOR-ARDUINO-MATLAB system by adding a connectors to which we connect the sensor. The signal sensor of ECG sensor is connected to A0, the signal sensor of DYN is connected to A1. To the connectors are connected both + 5V and ground. And finally we connected the ECG sensor and Dynamometer to the Arduino (Figure 2.1). The electrodes are attached to the student's body(Figure 2.2).

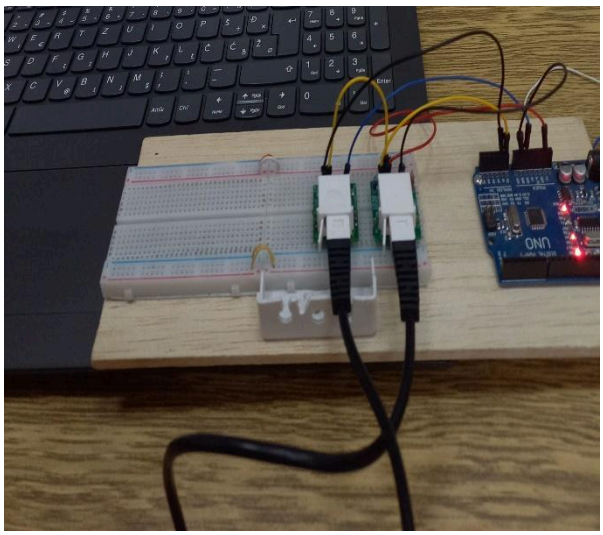

Figure 2.1

Figure 2.2

Software solution

We created a program in Arduino that selects EMG and Dynamometer (DYN) signals with a frequency of 200Hz (5ms) and sends them to Matlab at 57600bps. The signal buffer width is 512. This signal is sent to MATLAB with a speed of 57600 bps in a format suitable for Matlab.Next, we created a program that accepts the frame and displays it in graphical form.

Arduino code:

```
//ARDUINO ECG, send 512 samples in the frame from 2 chanels ,sensorValue1, sensorValue2, which
ends with CR/LF
#define LEN 512 //set number of samples in frame
#define FREQ 200 //set sampling frequency
int T; //sampling period
int sensorValue1; //value od A/D read
int sensorValue2;
void setup() // run once, when the sketch starts
{
T=1000/((int)(FREQ)); //set sampling period
Serial.begin(57600); // set up Serial library at 57600 bps
}
void loop() // loop
{
int i=0;


for(i=0; i<LEN; i++){ //in loop of the length LEN
sensorValue1 = analogRead(A0); //read A/D
sensorValue2 = analogRead(A1); //read A/D
Serial.print(String(sensorValue1)+" "+String(sensorValue2)); //print value
delay(T); // delay T
}
Serial.println(); } // put at the end of frame CR/LF sign
```

Matlab Code:

```
function ex1(op)
global COMM_P % The serial port object


global COMM_P_ON % Serial port on or off
global h % A structure containing handles to the GUI controls, axis, etc
global fs
global port
global baud
global buffer
global buffer_y1
global buffer_y2
global PAUSE
global file_counter
fs=200;
port='COM3';
baud=57600;
buffer=32000;
if nargin == 0 % if no input argument, draw the GUI
op = 0;
end
switch op
case 0
COMM_P_ON=0;
ss=strcat('Jovana Cipranic','-',num2str(fs),':',port,':',num2str(baud),':',num2str(buffer));
set(0,'DefaultFigurePosition',[6 6 728 412]); %possition of GUI
%GUI-Figure
h.fig = figure ('NumberTitle','off','Name',...
ss,...
```

```
'Resize','on','Pointer','watch','Color',[0.5 0.5 0.5],...
'InvertHardcopy','off','CloseRequestFcn', 'ex1(3)');


%GUI-INIT button
h.ctrl(8)=uicontrol('Style','Pushbutton','BackgroundColor',[0.6 0.6 0.6],'Position',...
[10 280 60 25],'String','>>','Enable','on','Callback','ex1(7)');
h.ctrl(7)=uicontrol('Style','Pushbutton','BackgroundColor',[0.6 0.6 0.6],'Position',...
[10 250 60 25],'String','[]','Enable','on','Callback','ex1(6)');
h.ctrl(6)=uicontrol('Style','Pushbutton','BackgroundColor',[0.6 0.6 0.6],'Position',...
[10 50 60 25],'String','CLEAR','Enable','on','Callback','ex1(5)');
h.ctrl(1)=uicontrol('Style','Pushbutton','BackgroundColor',[0.6 0.6 0.6],'Position',...
[10 80 60 25],'String','INIT','Enable','on','Callback','ex1(1)');
h.ctrl(2)=uicontrol('Style','Pushbutton','BackgroundColor',[0.6 0.6 0.6],'Position',...
[10 110 60 25],'String','SAVE','Enable','on','Callback','ex1(4)');
set(gcf,'Pointer','arrow')
%GUI-Status text box
h.ctrl(3) = uicontrol('Style','text','Position',[10 160 60 25],'String','Ready...',...
'HorizontalAlignment','left','FontSize',8);
h.ctrl(4) = uicontrol('Style','text','Position',[10 5 320 25],'BackgroundColor',[0.8 0.8
0.8],'String','Buffer...',...
'HorizontalAlignment','left','FontSize',10);
h.ctrl(5) = uicontrol('Style','text','Position',[10 220 60 25],'String','File...',...
'HorizontalAlignment','left','FontSize',8);
case 1 %if INIT button is pressed
if(COMM_P_ON==0) %if serial port is closed
COMM_P=serial(port,'BaudRate',baud,'InputBufferSize',buffer); %define
fopen(COMM_P); %open
COMM_P.BytesAvailableFcnMode = 'terminator'; %on terminator
COMM_P.BytesAvailableFcn = 'ex1(2)'; %go to ex1(2), call back function
```

```
COMM_P_ON=1; %set flag


fprintf(COMM_P,'a'); %request from MC to send frame
end
answer=COMM_P.status; %display port status
if strcmp(answer,'open')
set(h.ctrl(3),'String','CONECTED')
file_counter=0;
else
set(h.ctrl(3),'String','COMM ERR')
end
case 2 %Callback function on "terminator"
pp=fscanf(COMM_P); %read port's FIFO characters
y=str2num(pp); %convert in numbers
len_all=length(y);
y1=y(1:2:length(y));
y2=y(2:2:length(y));
len_all1=length(y1);%find length
len_all2=length(y2);
x1=[1:1:len_all1]/fs;%intialise x
x2=[1:1:len_all2]/fs;
if len_all>0 % if length > of some character
if(PAUSE==0)
%PUT PROCESSING ALGORITHMS
y1=y1*0.8564;
subplot(311)
plot(x1,y1,'red');
xlabel('[s]');
ylabel('DIN[N]');
```

```
grid minor

%PROCESSING
%ploting
subplot(312)
plot(x2,y2,'green');
xlabel('[s]');
ylabel('EMG[V]');
grid minor

buffer_y1=[buffer_y1,y1];
buffer_y2=[buffer_y2,y2];

subplot(313) %ploting
plot(buffer_y1+400,'blue');

hold on;
plot(buffer_y2,'red');
hold off;
xlabel('[s]');
ylabel('All signal');
grid minor
end
end
case 3 %dialog box for closing GUI
answer = questdlg('Really close PPG','','Yes','No','No');
if strcmp(answer,'Yes')
```

```
if COMM_P_ON==1
fprintf(COMM_P,'s'); %stop sending
fclose(COMM_P);
PAUSE=0;
end
clear all;
delete(gcf);
end

case 4
file_name=strcat('MEDLog',num2str(file_counter),'.txt');
fid=fopen(file_name,'wt');
fprintf(fid, '%f\r\n', fs);
fprintf(fid, '%f\r\n', buffer_y');
fclose(fid);
file_counter=file_counter+1;
set(h.ctrl(5),'String',file_name)
case 5
buffer_y=[];
case 6
PAUSE=1;
case 7
PAUSE=0;
end
```

Results in Matlab(Figure 2.3, Figure 2.4):

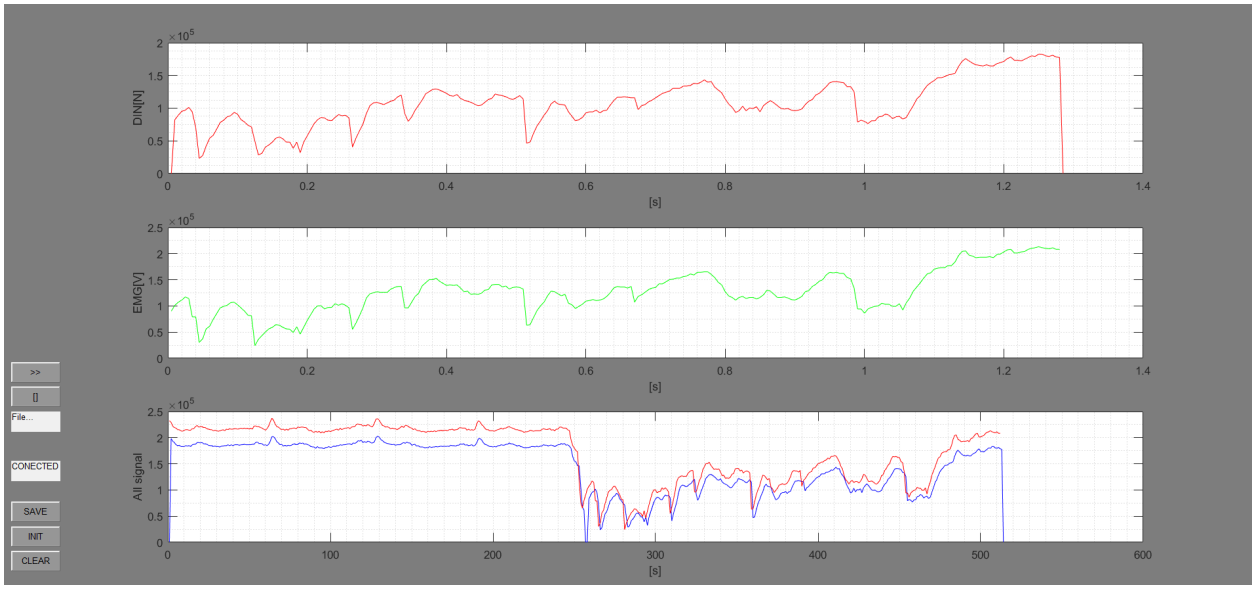

Figure 2.3

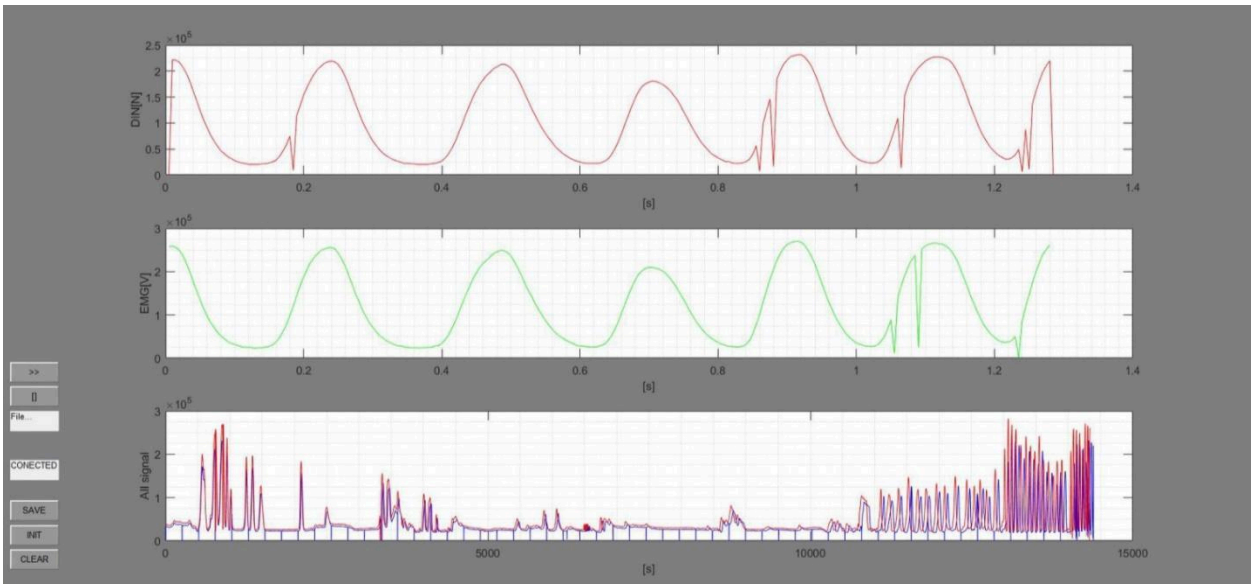

Figure 2.4

Link for the video

- *https://www.youtube.com/watch?v=hOPXecMyneI&t=6s*

Literature

1. Radovan D. Stojanovic-Bioelektrični procesi u organizmu, elektrografija, ECG, EEG, EMG
2. https://www.vernier.com/manuals/ekg-bta/
3. https://www.vernier.com/product/hand-dynamometer/
4. https://en.wikipedia.org/wiki/Arduino_Uno
5. https://en.wikipedia.org/wiki/MATLAB