

Vježba1_MEDEL_kod

```
//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
```

```
}
```

```
//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 function for displaying and processing frame received by serial port

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='COM5';

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]); %position 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','CONNECTED')

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