function varargout = SoftwareForsoeg(varargin)

gui\_Singleton = 1;

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

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

'gui\_OpeningFcn', @SoftwareForsoeg\_OpeningFcn, ...

'gui\_OutputFcn', @SoftwareForsoeg\_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

% --- Køres lige før Patient\_Oevelse gøres synlig.

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

% Choose default command line output for SoftwareForsoeg

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% --- Outputs fra denne funktion returneres til kommando linjen.

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

xlabel('Tid [Sek]'); % Koordinatsystemets x-label.

xlim auto; % Koordinatsystemets x-akse.

ylabel('Hældning [Grader]'); % Koordinatsystemet y-label

ylim([-90 90]); % Koordinatsystemets y-akse

refline(0,13) % Indstiller en refline i GUI'ens koordinatsystem ved 13 grader

refline(0,8) % Indstiller en refline i GUI'ens koordinatsystem ved 8 grader

refline(0,-8) % Indstiller en refline i GUI'ens koordinatsystem ved -8 grader

refline(0,-13) % Indstiller en refline i GUI'ens koordinatsystem ved -13 grader

hold on % Gør at der bliver hold fast i det indstillede koordinatsystem

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Ved tryk på StartKnap køres denne funktion.

function StartKnap\_Callback(hObject, eventdata, handles)

global s % Definerer vores s til at være global så den kan kaldes fra andre funktioner.

device = daq.getDevices;% Finder vores nidaq device.

s = daq.createSession('ni');% Starter en session s

aich = addAnalogInputChannel(s, 'Dev1', 'ai0', 'Voltage');% Tilføjer vores kanal på

% nidaqen til matlab, så der vides hvorfra der skal hentes data.

% aich.InputType = 'SingleEnded'

% s.Channels.TerminalConfig = 'SingleEnded'; %Definerer vores måling til at være singleended

s.IsContinuous = true; % Specificerer en kontinuert måling

s.Rate = 500 % Sampleraten af målingen

s.NotifyWhenDataAvailableExceeds = 25; %Indstiller hvor meget signal der skal optages før listener kaldes

lh = addlistener(s, 'DataAvailable',@plotData); %Listener der aktiveres når

%NotifyWhenDataAvailableExceeds overstiger den specificerede længde. Kalder funktionen plotdata

s.startBackground;%Starter optagelsen af signaler i baggrunden

function plotData(src,event) %Funktion der kaldes fra vores listener

ax = gca; %Finder det koordinatsystemet der er i GUI'en og gemmer det i ax

refline(ax, 0,13) % Indstiller en refline i GUI'ens koordinatsystem ved 13 grader

refline(ax, 0,8) % Indstiller en refline i GUI'ens koordinatsystem ved 8 grader

refline(ax, 0,-8) % Indstiller en refline i GUI'ens koordinatsystem ved -8 grader

refline(ax, 0,-13) % Indstiller en refline i GUI'ens koordinatsystem ved -13 grader

hold on % Gør at vores reflines ikke forsvinder

data = event.Data+1.3988; %Plussser med offsettet fra nidaqen

if 0<=data % Hvis dataen er over eller lig med 0 ganges dataen, således det er i grader

data = data\*(90/3.0147)

elseif data<=0 % Hvis dataen er under eller lig med 0 ganges dataen, således det er i grader

data = data\*(90/2.9417)

else % Hvis dataen hverken er over eller under 0 ganges dataen, således det er i grader

data = data\*(90/2.9417)

end

plot(ax, event.TimeStamps, data, 'r'); %Plotter vores data som kommer

%fra listeneren, i vores koordinatsystem

% --- Ved tryk af StopKnap køres denne funktion.

function StopKnap\_Callback(hObject, eventdata, handles)

global s % Finder vores globale variabel s

stop(s); % Stopper vores session s

release(s); % Gør at nidaqen kan bruges af andre programmer

% --- Ved tryk på SletKnap køres denne funktion.

function SletKnap\_Callback(hObject, eventdata, handles)

h = findobj(gca, 'color', 'red'); %Finder et objekt der er rødt i vores koordinatsystem og gemmer det i h

delete(h); %Sletter h, som er det røde i vores koordinatsystem

% --- Ved tryk på GemKnap køres denne funktion.

function GemKnap\_Callback(hObject, eventdata, handles)

[filename, pathname] = uiputfile({'\*.jpg;\*.tif;\*.png;\*.gif','All Image Files';...

'\*.\*','All Files' },'Save Image',...

'C:\Work\newfile.jpg') %Gør at der kommer en popup, der beder om navn,

%typen som filen kan gemmes som er også defineret her

name=fullfile(pathname, filename); % Gør at vi gemmer et helt filnavn sammensat af pathname og filename,

%hvilket gør at brugeren selv kan bestemme hvor hver fil skal gemmes.

fig = gcf; %Henter vores figur som den ser ud i GUI'en

fig.PaperPositionMode = 'auto'; %Auto skalerer vores fig så den ser ud som den gør i vores GUI

print(name,'-dpng','-r0') %Gemmer figuren.