classdef GUI < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

ELECTRONICSTETHOSCOPELabel matlab.ui.control.Label

Label2 matlab.ui.control.Label

RPMLabel matlab.ui.control.Label

BPMLabel matlab.ui.control.Label

LableLabel matlab.ui.control.Label

ORDEROFFILTEREditField matlab.ui.control.NumericEditField

ORDEROFFILTEREditFieldLabel matlab.ui.control.Label

LPCUTOFFFREQEditField matlab.ui.control.NumericEditField

LPCUTOFFFREQEditFieldLabel matlab.ui.control.Label

HPCUTOFFFREQEditField matlab.ui.control.NumericEditField

HPCUTOFFFREQEditFieldLabel matlab.ui.control.Label

STOPButton matlab.ui.control.Button

STARTButton matlab.ui.control.Button

UIAxes\_4 matlab.ui.control.UIAxes

UIAxes\_3 matlab.ui.control.UIAxes

UIAxes\_2 matlab.ui.control.UIAxes

UIAxes matlab.ui.control.UIAxes

end

properties (Access = private)

value

f

f1

cf1

cf2

X1

X2

X3

X4

serialport

Fs = 8000;

fs = 5000;

fc1 = 30;

fc2 = 200;

T = 1/5000;

m = 1;

plotWindow = 2000;

IsRunning = false;

Timer

Data

ffData

fdata

coefficient\_1

coefficient\_2

fc3=35

bpm

coefficient\_3

coefficient\_4

orderOfFilter=4;

end

methods (Access = private)

function timerCallback = func(app)

if ~app.IsRunning

stop(app.Timer);

end

end

end

% Callbacks that handle component events

methods (Access = private)

% Code that executes after component creation

function startupFcn(app)

app.IsRunning=true;

app.Timer=timer;

app.Timer.Period=0.1;

app.Timer.ExecutionMode='fixedRate';

app.Timer.TimerFcn=@(~,~)app.timerCallback;

start(app.Timer);

end

% Button pushed function: STARTButton

function STARTButtonPushed(app, event)

app.IsRunning = true;

app.value = serialport("COM8", 9600);

configureTerminator(app.value, "CR/LF");

flush(app.value);

app.value.UserData = struct("Data", [], "Order", 1, "fData", [],"ffData",[]);

app.IsRunning = true;

while app.value.UserData.Order < 100000

app.Data = readline(app.value);

app.value.UserData.Data(end+1) = str2double(app.Data);

app.value.UserData.Order = app.value.UserData.Order + 1;

if mod(app.value.UserData.Order, 10) == 0

configureCallback(app.value, "off");

plot(app.UIAxes, app.value.UserData.Data(max(1,end-app.plotWindow+1):end),'Color','y')

[app.coefficient\_1, app.coefficient\_2] = butter(4, app.fc3/(app.fs/2), 'high');

app.value.UserData.ffData = filter(app.coefficient\_1, app.coefficient\_2, app.value.UserData.Data(1:end));

[app.coefficient\_3, app.coefficient\_4] = butter(app.orderOfFilter, [app.fc1/(app.fs/2), app.fc2/(app.fs/2)], 'bandpass');

app.value.UserData.fData = filter(app.coefficient\_3, app.coefficient\_4, app.value.UserData.Data);

plot(app.UIAxes\_2, app.value.UserData.fData(max(1, end-app.plotWindow+1):end) \* 3,'Color','y');

drawnow;

configureTerminator(app.value, "CR/LF");

end

if mod(app.value.UserData.Order,500)==0

configureCallback(app.value, "off");

app.cf1 = fft(app.value.UserData.ffData(end-499+1:end));

app.X2 = abs(app.cf1 / 500); % Use 400 for normalization

app.f = app.fs \* (0:(length(app.cf1)/2))/length(app.cf1); % Adjust frequency axis

app.X1 = app.X2(1:length(app.cf1)/2+1);

app.X1(2:end-1) = 2 \* app.X1(2:end-1);

plot(app.UIAxes\_3,app.f,app.X1,'Color','c');

set(gca,'ylim',[0,10]);

app.cf2 = fft(app.value.UserData.fData(end-499+1:end));

app.X3 = abs(app.cf2 / 500); % Use 400 for normalization

app.f1 = app.fs \* (0:(length(app.cf2)/2))/length(app.cf2); % Adjust frequency axis

app.X4 = app.X3(1:length(app.cf2)/2+1);

app.X4(2:end-1) = 2 \* app.X4(2:end-1);

plot(app.UIAxes\_4, app.f1,app.X4,'Color','c');

set(gca,'ylim',[0,15]);

grid on;

app.m=app.m+500;

configureTerminator(app.value,"CR/LF");

end

if mod(app.value.UserData.Order, 1500) == 0

% Measure elapsed time

timeAtPointBPM = toc;

% Reset the timer immediately for the next interval

% Detect peaks in the data

threshold = 490; % Adjust this threshold as needed

[peaks, ~] = findpeaks(app.value.UserData.Data(end-1499+1:end), 'MinPeakHeight', threshold);

numPeaks = numel(peaks); % Count the number of peaks

% Calculate BPM

app.bpm = (60 \* numPeaks) / timeAtPointBPM; % BPM calculation

disp(int16(app.bpm)); % Display BPM

tic;

app.LableLabel.Text = num2str(app.bpm);

end

end

end

% Button pushed function: STOPButton

function STOPButtonPushed(app, event)

app.IsRunning=false;

delete(app.value);

end

% Value changed function: LPCUTOFFFREQEditField

function LPCUTOFFFREQEditFieldValueChanged(app, event)

b=app.LPCUTOFFFREQEditField.Value;

app.fc2=b;

end

% Value changed function: HPCUTOFFFREQEditField

function HPCUTOFFFREQEditFieldValueChanged(app, event)

c= app.HPCUTOFFFREQEditField.Value;

app.fc1=c;

end

% Value changed function: ORDEROFFILTEREditField

function ORDEROFFILTEREditFieldValueChanged(app, event)

j = app.ORDEROFFILTEREditField.Value;

app.orderOfFilter=j;

end

end

% Component initialization

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure and hide until all components are created

app.UIFigure = uifigure('Visible', 'off');

app.UIFigure.Color = [0.502 0.502 0.502];

app.UIFigure.Position = [100 100 1127 792];

app.UIFigure.Name = 'MATLAB App';

% Create UIAxes

app.UIAxes = uiaxes(app.UIFigure);

title(app.UIAxes, 'ORIGINAL SIGNAL')

xlabel(app.UIAxes, 'TIME')

ylabel(app.UIAxes, 'AMPLITUDE')

zlabel(app.UIAxes, 'Z')

app.UIAxes.Color = [0.149 0.149 0.149];

app.UIAxes.GridColor = [1 1 1];

app.UIAxes.MinorGridColor = [0 0 0];

app.UIAxes.XGrid = 'on';

app.UIAxes.YGrid = 'on';

app.UIAxes.ZGrid = 'on';

app.UIAxes.Position = [24 350 478 311];

% Create UIAxes\_2

app.UIAxes\_2 = uiaxes(app.UIFigure);

title(app.UIAxes\_2, 'FILTERED SIGNAL')

xlabel(app.UIAxes\_2, 'TIME')

ylabel(app.UIAxes\_2, 'AMPLUTED')

zlabel(app.UIAxes\_2, 'Z')

app.UIAxes\_2.Color = [0.149 0.149 0.149];

app.UIAxes\_2.GridColor = [1 1 1];

app.UIAxes\_2.XGrid = 'on';

app.UIAxes\_2.YGrid = 'on';

app.UIAxes\_2.ZGrid = 'on';

app.UIAxes\_2.Position = [26 23 478 311];

% Create UIAxes\_3

app.UIAxes\_3 = uiaxes(app.UIFigure);

title(app.UIAxes\_3, 'ORIGINAL SIGNAL FFT')

xlabel(app.UIAxes\_3, 'FREQUENCY')

ylabel(app.UIAxes\_3, 'AMPLUTED')

zlabel(app.UIAxes\_3, 'Z')

app.UIAxes\_3.Color = [0.149 0.149 0.149];

app.UIAxes\_3.GridColor = [1 1 1];

app.UIAxes\_3.XGrid = 'on';

app.UIAxes\_3.YGrid = 'on';

app.UIAxes\_3.ZGrid = 'on';

app.UIAxes\_3.Position = [612 350 478 311];

% Create UIAxes\_4

app.UIAxes\_4 = uiaxes(app.UIFigure);

title(app.UIAxes\_4, 'FILTERED SIGNAL FFT')

xlabel(app.UIAxes\_4, 'FREQUENCY')

ylabel(app.UIAxes\_4, 'AMPLUTED')

zlabel(app.UIAxes\_4, 'Z')

app.UIAxes\_4.Color = [0.149 0.149 0.149];

app.UIAxes\_4.GridColor = [1 1 1];

app.UIAxes\_4.XGrid = 'on';

app.UIAxes\_4.YGrid = 'on';

app.UIAxes\_4.ZGrid = 'on';

app.UIAxes\_4.Position = [611 23 478 311];

% Create STARTButton

app.STARTButton = uibutton(app.UIFigure, 'push');

app.STARTButton.ButtonPushedFcn = createCallbackFcn(app, @STARTButtonPushed, true);

app.STARTButton.BackgroundColor = [0 1 1];

app.STARTButton.FontWeight = 'bold';

app.STARTButton.Position = [296 719 100 23];

app.STARTButton.Text = 'START';

% Create STOPButton

app.STOPButton = uibutton(app.UIFigure, 'push');

app.STOPButton.ButtonPushedFcn = createCallbackFcn(app, @STOPButtonPushed, true);

app.STOPButton.BackgroundColor = [1 0 0];

app.STOPButton.FontWeight = 'bold';

app.STOPButton.Position = [730 719 100 23];

app.STOPButton.Text = 'STOP';

% Create HPCUTOFFFREQEditFieldLabel

app.HPCUTOFFFREQEditFieldLabel = uilabel(app.UIFigure);

app.HPCUTOFFFREQEditFieldLabel.HorizontalAlignment = 'right';

app.HPCUTOFFFREQEditFieldLabel.FontWeight = 'bold';

app.HPCUTOFFFREQEditFieldLabel.Position = [720 674 113 22];

app.HPCUTOFFFREQEditFieldLabel.Text = 'HP CUT-OFF FREQ';

% Create HPCUTOFFFREQEditField

app.HPCUTOFFFREQEditField = uieditfield(app.UIFigure, 'numeric');

app.HPCUTOFFFREQEditField.ValueChangedFcn = createCallbackFcn(app, @HPCUTOFFFREQEditFieldValueChanged, true);

app.HPCUTOFFFREQEditField.FontWeight = 'bold';

app.HPCUTOFFFREQEditField.BackgroundColor = [0.8 0.8 0.8];

app.HPCUTOFFFREQEditField.Position = [848 674 100 22];

app.HPCUTOFFFREQEditField.Value = 30;

% Create LPCUTOFFFREQEditFieldLabel

app.LPCUTOFFFREQEditFieldLabel = uilabel(app.UIFigure);

app.LPCUTOFFFREQEditFieldLabel.BackgroundColor = [0.502 0.502 0.502];

app.LPCUTOFFFREQEditFieldLabel.HorizontalAlignment = 'right';

app.LPCUTOFFFREQEditFieldLabel.FontWeight = 'bold';

app.LPCUTOFFFREQEditFieldLabel.Position = [134 674 112 22];

app.LPCUTOFFFREQEditFieldLabel.Text = 'LP CUT-OFF FREQ';

% Create LPCUTOFFFREQEditField

app.LPCUTOFFFREQEditField = uieditfield(app.UIFigure, 'numeric');

app.LPCUTOFFFREQEditField.ValueChangedFcn = createCallbackFcn(app, @LPCUTOFFFREQEditFieldValueChanged, true);

app.LPCUTOFFFREQEditField.FontWeight = 'bold';

app.LPCUTOFFFREQEditField.BackgroundColor = [0.8 0.8 0.8];

app.LPCUTOFFFREQEditField.Position = [260 674 100 22];

app.LPCUTOFFFREQEditField.Value = 200;

% Create ORDEROFFILTEREditFieldLabel

app.ORDEROFFILTEREditFieldLabel = uilabel(app.UIFigure);

app.ORDEROFFILTEREditFieldLabel.HorizontalAlignment = 'right';

app.ORDEROFFILTEREditFieldLabel.FontName = 'AvantGarde';

app.ORDEROFFILTEREditFieldLabel.FontWeight = 'bold';

app.ORDEROFFILTEREditFieldLabel.Position = [424 674 113 22];

app.ORDEROFFILTEREditFieldLabel.Text = 'ORDER OF FILTER';

% Create ORDEROFFILTEREditField

app.ORDEROFFILTEREditField = uieditfield(app.UIFigure, 'numeric');

app.ORDEROFFILTEREditField.ValueChangedFcn = createCallbackFcn(app, @ORDEROFFILTEREditFieldValueChanged, true);

app.ORDEROFFILTEREditField.BackgroundColor = [0.8 0.8 0.8];

app.ORDEROFFILTEREditField.Position = [553 674 100 22];

app.ORDEROFFILTEREditField.Value = 4;

% Create LableLabel

app.LableLabel = uilabel(app.UIFigure);

app.LableLabel.FontName = 'AvantGarde';

app.LableLabel.FontWeight = 'bold';

app.LableLabel.Position = [544 550 37 22];

app.LableLabel.Text = 'Lable';

% Create BPMLabel

app.BPMLabel = uilabel(app.UIFigure);

app.BPMLabel.FontSize = 14;

app.BPMLabel.FontWeight = 'bold';

app.BPMLabel.Position = [540 588 45 22];

app.BPMLabel.Text = 'BPM: ';

% Create RPMLabel

app.RPMLabel = uilabel(app.UIFigure);

app.RPMLabel.FontSize = 14;

app.RPMLabel.FontWeight = 'bold';

app.RPMLabel.Position = [549 232 36 22];

app.RPMLabel.Text = 'RPM';

% Create Label2

app.Label2 = uilabel(app.UIFigure);

app.Label2.FontName = 'AvantGarde';

app.Label2.FontWeight = 'bold';

app.Label2.Position = [545 195 43 22];

app.Label2.Text = 'Label2';

% Create ELECTRONICSTETHOSCOPELabel

app.ELECTRONICSTETHOSCOPELabel = uilabel(app.UIFigure);

app.ELECTRONICSTETHOSCOPELabel.HorizontalAlignment = 'center';

app.ELECTRONICSTETHOSCOPELabel.FontName = 'Bell MT';

app.ELECTRONICSTETHOSCOPELabel.FontSize = 36;

app.ELECTRONICSTETHOSCOPELabel.FontWeight = 'bold';

app.ELECTRONICSTETHOSCOPELabel.Position = [295 741 534 47];

app.ELECTRONICSTETHOSCOPELabel.Text = 'ELECTRONIC STETHOSCOPE';

% Show the figure after all components are created

app.UIFigure.Visible = 'on';

end

end

% App creation and deletion

methods (Access = public)

% Construct app

function app = GUI

% Create UIFigure and components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @startupFcn)

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end