```
function varargout = project(varargin)
% PROJECT MATLAB code for project.fig
%      PROJECT, by itself, creates a new PROJECT or
raises the existing
%      singleton*.
%
%      H = PROJECT returns the handle to a new
PROJECT or the handle to
%      the existing singleton*.
%
%
PROJECT('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in PROJECT.M with
the given input arguments.
%
%      PROJECT('Property','Value',...) creates a
new PROJECT or raises the
%      existing singleton*.  Starting from the
left, property value pairs are
%      applied to the GUI before project_OpeningFcn
gets called.  An
%      unrecognized property name or invalid value
makes property application
%      stop.  All inputs are passed to
project_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.
Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to
help project

% Last Modified by GUIDE v2.5 22-May-2023 11:42:12
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton,
...
                   'gui_OpeningFcn',
@project_OpeningFcn, ...
                   'gui_OutputFcn',
@project_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
% End initialization code - DO NOT EDIT


% --- Executes just before project is made visible.
function project_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
% varargin   command line arguments to project (see
VARARGIN)

% Choose default command line output for project
handles.output = hObject;
handles.fullpathname='';
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes project wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to
the command line.
function varargout = project_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args
(see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% Get default command line output from handles
structure
varargout{1} = handles.output;


% --- Executes on button press in search.
function search_Callback(hObject, eventdata,
handles)
% hObject    handle to search (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
[file path] = uigetfile({'*.wav'},'File Selector');
handles.fullpathname = strcat(path, file);
if(isempty(handles.fullpathname))
    f = msgbox('Enter File!!!');
else
```

```
    [handles.y,handles.Fs] =
audioread(handles.fullpathname);
    global rate;
    rate = 1;
end
set(handles.file, 'String',handles.fullpathname)
%showing fullpathname
guidata(hObject,handles)


% --- Executes on slider movement.
function volume_Callback(hObject, eventdata,
handles)
% hObject    handle to volume (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% --- Executes during object creation, after
setting all properties.
function volume_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to volume (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until
after all CreateFcns called

% Hint: slider controls usually have a light gray
background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in play.
function play_Callback(hObject, eventdata, handles)
% hObject    handle to play (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
global player;
if(strcmp(handles.fullpathname,'')==1)
    search_Callback(hObject, eventdata, handles);
else
play_equalizer(hObject, handles);
play(player);
end
guidata(hObject,handles)

% --- Executes on button press in stop.
function stop_Callback(hObject, eventdata, handles)
% hObject    handle to stop (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
global player;
if(strcmp(handles.fullpathname,'')==1)
    search_Callback(hObject, eventdata, handles);
else
play_equalizer(hObject, handles);
stop(player);
end
guidata(hObject,handles)

% --- Executes on button press in resume.
function resume_Callback(hObject, eventdata,
handles)
% hObject    handle to resume (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
global player;
```

```
if(strcmp(handles.fullpathname,'')==1)
    search_Callback(hObject, eventdata, handles);
else
play_equalizer(hObject, handles);
resume(player);
end
guidata(hObject,handles)

% --- Executes on button press in pause.
function pause_Callback(hObject, eventdata,
handles)
% hObject    handle to pause (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
global player;
if(strcmp(handles.fullpathname,'')==1)
    search_Callback(hObject, eventdata, handles);
else
play_equalizer(hObject, handles);
pause(player);
end
guidata(hObject,handles)

% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata,
handles)
% hObject    handle to reset (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
set(handles.slider1,'value',0);
set(handles.slider2,'value',0);
set(handles.slider3,'value',0);
set(handles.slider4,'value',0);
set(handles.slider5,'value',0);
set(handles.slider6,'value',0);
set(handles.slider7,'value',0);
set(handles.slider8,'value',0);
set(handles.slider9,'value',0);
guidata(hObject,handles)


function [handles]= values(hObject, handles)
global player;
%[handles.y,handles.Fs] =
audioread(handles.fullpathname);
handles.Volume=get(handles.volume,'value');
handles.g1=get(handles.slider1,'value');
handles.g2=get(handles.slider2,'value');
handles.g3=get(handles.slider3,'value');
handles.g4=get(handles.slider4,'value');
handles.g5=get(handles.slider5,'value');
handles.g6=get(handles.slider6,'value');
handles.g7=get(handles.slider7,'value');
handles.g8=get(handles.slider8,'value');
handles.g9=get(handles.slider9,'value');
guidata(hObject,handles)

function axesPlot(hObject, handles)
plot(handles.Otime ,handles.y); %Original otime
Domain Signal
plot(handles.Ftime,handles.yT); %Filtered Time
Domain Signal
plot(handles.Ofreq,abs(fftshift(fft(handles.y))));
%Original Freq Domain Signal
plot(handles.Ffreq,abs(fftshift(fft(handles.yT))));
%Filtered Freq Domain Signal
guidata(hObject,handles)

function [handles]= freqFilters(hObject,
handles,cut_off,order)

freq=
[170,300,610,1005,3000,6000,12000,14000,20000];
```

```
global a0 a1 a2 a3 a4 a5 a6 a7 a8;
global b0 b1 b2 b3 b4 b5 b6 b7 b8;
a0=1;a1=1;a2=1;a3=1;a4=1;a5=1;a6=1;a7=1;a8=1;

b0=fir1(order,cut_off/(handles.Fs/2),'low');
y1=handles.g1*filter(b0,1,handles.y);

% 171-->300
b1=fir1(order,[(freq(1)+1)/(handles.Fs/2)
freq(2)/(handles.Fs/2)],'bandpass');
y2=handles.g2*filter(b1,1,handles.y);

% 301-->610
b2=fir1(order,[(freq(2)+1)/(handles.Fs/2)
freq(3)/(handles.Fs/2)],'bandpass');
y3=handles.g3*filter(b2,1,handles.y);

% 611-->1005
b3=fir1(order,[(freq(3)+1)/(handles.Fs/2)
freq(4)/(handles.Fs/2)],'bandpass');
y4=handles.g4*filter(b3,1,handles.y);

% 1006-->3000
b4=fir1(order,[(freq(4)+1)/(handles.Fs/2)
freq(5)/(handles.Fs/2)],'bandpass');
y5=handles.g5*filter(b4,1,handles.y);

% 3001-->6000
b5=fir1(order,[(freq(5)+1)/(handles.Fs/2)
freq(6)/(handles.Fs/2)],'bandpass');
y6=handles.g6*filter(b5,1,handles.y);

% 6001-->12000
b6=fir1(order,[(freq(6)+1)/(handles.Fs/2)
freq(7)/(handles.Fs/2)],'bandpass');
y7=handles.g7*filter(b6,1,handles.y);

% 12001-->14000
```

```
b7=fir1(order,[(freq(7)+1)/(handles.Fs/2)
freq(8)/(handles.Fs/2)],'bandpass');
y8=handles.g8*filter(b7,1,handles.y);

% 14001-->20000
b8=fir1(order,[(freq(8)+1)/(handles.Fs/2)
freq(9)/(handles.Fs/2)],'bandpass');
y9=handles.g9*filter(b8,1,handles.y);

%highpass
global filtered;
handles.yT=y1+y2+y3+y4+y5+y6+y7+y8+y9;
filtered =handles.yT;
guidata(hObject,handles)

function play_equalizer(hObject, handles)
global player;
global rate;
[handles] = values(hObject, handles); %reads input
from slider

[handles]= freqFilters(hObject, handles,170,16);
%gets the filtered signal
handles.Fs = handles.Fs*rate; %tzbat el frequency
rate
global freqRate;
freqRate =handles.Fs;
player =
audioplayer(handles.Volume*handles.yT,freqRate);
axesPlot(hObject,handles);
guidata(hObject,handles)


% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata,
handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
```

```
% handles    structure with handles and user data
(see GUIDATA)

% --- Executes during object creation, after
setting all properties.
function slider1_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until
after all CreateFcns called

% Hint: slider controls usually have a light gray
background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider2_Callback(hObject, eventdata,
handles)
% hObject    handle to slider2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% --- Executes during object creation, after
setting all properties.
function slider2_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to slider2 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until
after all CreateFcns called
```

```
% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider3_Callback(hObject, eventdata, handles)
% hObject    handle to slider3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function slider3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider4_Callback(hObject, eventdata, handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% --- Executes during object creation, after
setting all properties.
function slider4_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until
after all CreateFcns called

% Hint: slider controls usually have a light gray
background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider5_Callback(hObject, eventdata,
handles)
% hObject    handle to slider5 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% --- Executes during object creation, after
setting all properties.
function slider5_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to slider5 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
```

```
% handles    empty - handles not created until
after all CreateFcns called

% Hint: slider controls usually have a light gray
background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider6_Callback(hObject, eventdata,
handles)
% hObject    handle to slider6 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% --- Executes during object creation, after
setting all properties.
function slider6_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to slider6 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until
after all CreateFcns called

% Hint: slider controls usually have a light gray
background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
```

```
function slider7_Callback(hObject, eventdata,
handles)
% hObject    handle to slider7 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% --- Executes during object creation, after
setting all properties.
function slider7_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to slider7 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until
after all CreateFcns called

% Hint: slider controls usually have a light gray
background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider8_Callback(hObject, eventdata,
handles)
% hObject    handle to slider8 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% --- Executes during object creation, after
setting all properties.
function slider8_CreateFcn(hObject, eventdata,
handles)
```

```
% hObject    handle to slider8 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until
after all CreateFcns called

% Hint: slider controls usually have a light gray
background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider9_Callback(hObject, eventdata,
handles)
% hObject    handle to slider9 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)

% --- Executes during object creation, after
setting all properties.
function slider9_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to slider9 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until
after all CreateFcns called

% Hint: slider controls usually have a light gray
background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```

```
function [handles]= freqIIR(hObject,
handles,cut_off,order)
freq=
[170,300,610,1005,3000,6000,12000,14000,20000];

global a0 a1 a2 a3 a4 a5 a6 a7 a8;
global b0 b1 b2 b3 b4 b5 b6 b7 b8;

% 0-->170
[b0,a0]=butter(order,cut_off/(handles.Fs/2),'low');
y1=handles.g1*filter(b0,a0,handles.y);
% 171-->300
[b1,a1]=butter(order,[(freq(1)+1)/(handles.Fs/2)
freq(2)/(handles.Fs/2)],'bandpass');
y2=handles.g2*filter(b1,a1,handles.y);
% 301-->610
[b2,a2]=butter(order,[(freq(2)+1)/(handles.Fs/2)
freq(3)/(handles.Fs/2)],'bandpass');
y3=handles.g3*filter(b2,a2,handles.y);
% 611-->1005
[b3,a3]=butter(order,[(freq(3)+1)/(handles.Fs/2)
freq(4)/(handles.Fs/2)],'bandpass');
y4=handles.g4*filter(b3,a3,handles.y);
% 1006-->3000
[b4,a4]=butter(order,[(freq(4)+1)/(handles.Fs/2)
freq(5)/(handles.Fs/2)],'bandpass');
y5=handles.g5*filter(b4,a4,handles.y);
% 3001-->6000
[b5,a5]=butter(order,[(freq(5)+1)/(handles.Fs/2)
freq(6)/(handles.Fs/2)],'bandpass');
y6=handles.g6*filter(b5,a5,handles.y);
% 6001-->12000
[b6,a6]=butter(order,[(freq(6)+1)/(handles.Fs/2)
freq(7)/(handles.Fs/2)],'bandpass');
y7=handles.g7*filter(b6,a6,handles.y);
% 12001-->14000
[b7,a7]=butter(order,[(freq(7)+1)/(handles.Fs/2)
freq(8)/(handles.Fs/2)],'bandpass');
```

```
y8=handles.g8*filter(b7,a7,handles.y);
% 14001-->20000
[b8,a8]=butter(order,[(freq(8)+1)/(handles.Fs/2)
freq(9)/(handles.Fs/2)],'bandpass');
y9=handles.g9*filter(b8,a8,handles.y);
global filtered;
handles.yT=y1+y2+y3+y4+y5+y6+y7+y8+y9;
filtered =handles.yT;
guidata(hObject,handles)


function IIR(hObject, handles)
global player;
global rate;
[handles] = values(hObject, handles);
[handles]= freqIIR(hObject, handles,170,2);
handles.Fs =handles.Fs*rate;
global freqRate;
freqRate =handles.Fs;
player = audioplayer(handles.Volume*handles.yT,
handles.Fs);
axesPlot(hObject,handles);
guidata(hObject,handles)

function FIR(hObject, handles)
global player;
global rate;
[handles] = values(hObject, handles);
[handles]= freqFilters(hObject, handles,170,1);
handles.Fs =handles.Fs*rate;
global freqRate;
freqRate =handles.Fs;
player = audioplayer(handles.Volume*handles.yT,
handles.Fs);
axesPlot(hObject,handles);
guidata(hObject,handles)

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

```
function iir_Callback(hObject, eventdata, handles)
% hObject    handle to iir (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
global player;
if(strcmp(handles.fullpathname,'')==1)
    search_Callback(hObject, eventdata, handles);
else
IIR(hObject, handles);
play(player);
end
guidata(hObject,handles)

% --- Executes on button press in fir.
function fir_Callback(hObject, eventdata, handles)
% hObject    handle to fir (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
global player;
if(strcmp(handles.fullpathname,'')==1)
    search_Callback(hObject, eventdata, handles);
else
FIR(hObject, handles);
play(player);
end
guidata(hObject,handles)



% --- Executes on button press in sr0.
function sr0_Callback(hObject, eventdata, handles)
global b0 a0;
handles.w= stepz(b0,a0);
plot(handles.response,handles.w);
```

```
% --- Executes on button press in ir0.
function ir0_Callback(hObject, eventdata, handles)
global b0 a0;
handles.w= impz(b0,a0);
plot(handles.response,handles.w);

% --- Executes on button press in z0.
function z0_Callback(hObject, eventdata, handles)
global b0 a0;
H= tf(b0,a0);
axes(handles.response);
pzmap(H);
zplane(b0,a0);
plot(pzmap(H));

% --- Executes on button press in x0.
function x0_Callback(hObject, eventdata, handles)
global b0 a0;
handles.w= freqz(b0,a0);
axes(handles.response);
plot(abs(handles.w));
title('Gain: Blue      Phase: Orange');
hold on
plot(angle(handles.w));
hold off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
% --- Executes on button press in sr1.
function sr1_Callback(hObject, eventdata, handles)
global b1 a1;
handles.w= stepz(b1,a1);
plot(handles.response,handles.w);

% --- Executes on button press in ir1.
function ir1_Callback(hObject, eventdata, handles)
global b1 a1;
handles.w= impz(b1,a1);
plot(handles.response,handles.w);
```

```
% --- Executes on button press in z1.
function z1_Callback(hObject, eventdata, handles)
global b1 a1;
H= tf(b1,a1);
axes(handles.response);
pzmap(H);
zplane(b1,a1);
plot(pzmap(H));

% --- Executes on button press in x1.
function x1_Callback(hObject, eventdata, handles)
global b1 a1;
handles.w= freqz(b1,a1);
axes(handles.response);
plot(abs(handles.w));
title('Gain: Blue      Phase: Orange');
hold on
plot(angle(handles.w));
hold off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%

% --- Executes on button press in sr2.
function sr2_Callback(hObject, eventdata, handles)
global b2 a2;
handles.w= stepz(b2,a2);
plot(handles.response,handles.w);

% --- Executes on button press in ir2.
function ir2_Callback(hObject, eventdata, handles)
global b2 a2;
handles.w= impz(b2,a2);
plot(handles.response,handles.w);

% --- Executes on button press in z2.
function z2_Callback(hObject, eventdata, handles)
global b2 a2;
H= tf(b2,a2);
```

```
axes(handles.response);
pzmap(H);
zplane(b2,a2);
plot(pzmap(H));


% --- Executes on button press in x2.
function x2_Callback(hObject, eventdata, handles)
global b2 a2;
handles.w= freqz(b2,a2);
axes(handles.response);
plot(abs(handles.w));
title('Gain: Blue      Phase: Orange');
hold on
plot(angle(handles.w));
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%
% --- Executes on button press in sr3.
function sr3_Callback(hObject, eventdata, handles)
global b3 a3;
handles.w= stepz(b3,a3);
plot(handles.response,handles.w);

% --- Executes on button press in ir3.
function ir3_Callback(hObject, eventdata, handles)
global b3 a3;
handles.w= impz(b3,a3);
plot(handles.response,handles.w);

% --- Executes on button press in z3.
function z3_Callback(hObject, eventdata, handles)
global b3 a3;
H= tf(b3,a3);
axes(handles.response);
pzmap(H);
zplane(b3,a3);
plot(pzmap(H));
```

```
% --- Executes on button press in x3.
function x3_Callback(hObject, eventdata, handles)
global b3 a3;
handles.w= freqz(b3,a3);
axes(handles.response);
plot(abs(handles.w));
title('Gain: Blue      Phase: Orange');
hold on
plot(angle(handles.w));
hold off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% --- Executes on button press in sr4.
function sr4_Callback(hObject, eventdata, handles)
global b4 a4;
handles.w= stepz(b4,a4);
plot(handles.response,handles.w);

% --- Executes on button press in ir4.
function ir4_Callback(hObject, eventdata, handles)
global b4 a4;
handles.w= impz(b4,a4);
plot(handles.response,handles.w);

% --- Executes on button press in z4.
function z4_Callback(hObject, eventdata, handles)
global b4 a4;
H= tf(b4,a4);
axes(handles.response);
pzmap(H);
zplane(b4,a4);
plot(pzmap(H));


% --- Executes on button press in x4.
function x4_Callback(hObject, eventdata, handles)
global b4 a4;
handles.w= freqz(b4,a4);
```

```
axes(handles.response);
plot(abs(handles.w));
title('Gain: Blue      Phase: Orange');
hold on
plot(angle(handles.w));
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% --- Executes on button press in sr5.
function sr5_Callback(hObject, eventdata, handles)
global b5 a5;
handles.w= stepz(b5,a5);
plot(handles.response,handles.w);

% --- Executes on button press in ir5.
function ir5_Callback(hObject, eventdata, handles)
global b5 a5;
handles.w= impz(b5,a5);
plot(handles.response,handles.w);

% --- Executes on button press in z5.
function z5_Callback(hObject, eventdata, handles)
global b5 a5;
H= tf(b5,a5);
axes(handles.response);
pzmap(H);
zplane(b5,a5);
plot(pzmap(H));


% --- Executes on button press in x5.
function x5_Callback(hObject, eventdata, handles)
global b5 a5;
handles.w= freqz(b5,a5);
axes(handles.response);
plot(abs(handles.w));
title('Gain: Blue      Phase: Orange');
hold on
```

```
plot(angle(handles.w));
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%5
% --- Executes on button press in sr6.
function sr6_Callback(hObject, eventdata, handles)
global b6 a6;
handles.w= stepz(b6,a6);
plot(handles.response,handles.w);

% --- Executes on button press in ir6.
function ir6_Callback(hObject, eventdata, handles)
global b6 a6;
handles.w= impz(b6,a6);
plot(handles.response,handles.w);

% --- Executes on button press in z6.
function z6_Callback(hObject, eventdata, handles)
global b6 a6;
H= tf(b6,a6);
axes(handles.response);
pzmap(H);
zplane(b6,a6);
plot(pzmap(H));

% --- Executes on button press in x6.
function x6_Callback(hObject, eventdata, handles)
global b6 a6;
handles.w= freqz(b6,a6);
axes(handles.response);
plot(abs(handles.w));
title('Gain: Blue      Phase: Orange');
hold on
plot(angle(handles.w));
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% --- Executes on button press in sr7.
```

```
function sr7_Callback(hObject, eventdata, handles)
global b7 a7;
handles.w= stepz(b7,a7);
plot(handles.response,handles.w);

% --- Executes on button press in ir7.
function ir7_Callback(hObject, eventdata, handles)
global b7 a7;
handles.w= impz(b7,a7);
plot(handles.response,handles.w);

% --- Executes on button press in z7.
function z7_Callback(hObject, eventdata, handles)
global b7 a7;
H= tf(b7,a7);
axes(handles.response);
pzmap(H);
zplane(b7,a7);
plot(pzmap(H));


% --- Executes on button press in x7.
function x7_Callback(hObject, eventdata, handles)
global b7 a7;
handles.w= freqz(b7,a7);
axes(handles.response);
plot(abs(handles.w));
title('Gain: Blue      Phase: Orange');
hold on
plot(angle(handles.w));
hold off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
% --- Executes on button press in sr8.
function sr8_Callback(hObject, eventdata, handles)
global b8 a8;
handles.w= stepz(b8,a8);
plot(handles.response,handles.w);

% --- Executes on button press in ir8.
```

```
function ir8_Callback(hObject, eventdata, handles)
global b8 a8;
handles.w= impz(b8,a8);
plot(handles.response,handles.w);

% --- Executes on button press in z8.
function z8_Callback(hObject, eventdata, handles)
global b8 a8;
H= tf(b8,a8);
axes(handles.response);
pzmap(H);
zplane(b8,a8);
plot(pzmap(H));


% --- Executes on button press in x8.
function x8_Callback(hObject, eventdata, handles)
global b8 a8;
handles.w= freqz(b8,a8);
axes(handles.response);
plot(abs(handles.w));
title('Gain: Blue      Phase: Orange');
hold on
plot(angle(handles.w));
hold off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5


% --- Executes on selection change in fs.
function fs_Callback(hObject, eventdata, handles)
% hObject    handle to fs (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
s= char('1','0.25','0.5','2','1.25');
set(handles.fs,'string',s);
contents = cellstr(get(hObject,'String'));
global rate;
```

```
rate = str2num(contents{get(hObject,'Value')});

% Hints: contents = cellstr(get(hObject,'String'))
returns fs contents as cell array
%        contents{get(hObject,'Value')} returns
selected item from fs


% --- Executes during object creation, after
setting all properties.
function fs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fs (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until
after all CreateFcns called

% Hint: popupmenu controls usually have a white
background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- If Enable == 'on', executes on mouse press in
5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel
border or over fs.
function fs_ButtonDownFcn(hObject, eventdata,
handles)
% hObject    handle to fs (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
```

```
% --- Executes on button press in save.
function save_Callback(hObject, eventdata, handles)
% hObject    handle to save (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
prompt = {'Enter Sound Name:'};
dlgtitle = 'Input';
dims = [1 35];
definput = {'filename','hsv'};
name = inputdlg(prompt,dlgtitle,dims,definput);
newfile = strcat(name,'.wav')
global filtered;
global freqRate;
audiowrite(newfile{1,1},filtered,freqRate);
```