```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% User Interface
% Author: LCCI Team
% Version One
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%TIMESTAMP IS NOT WORKING

% Description: Intended for user to select preexisting audio file, record
% and save audio, specify carrier frequency and display relevant audio
% information
% Purpose: Intended to streamine process of testing LCCI device

function varargout = LCCIUserInterface(varargin)
% LCCIUSERINTERFACE MATLAB code for LCCIUserInterface.fig
%      LCCIUSERINTERFACE, by itself, creates a new LCCIUSERINTERFACE or raises the existing
%      singleton*.
%
%      H = LCCIUSERINTERFACE returns the handle to a new LCCIUSERINTERFACE or the handle to
%      the existing singleton*.
%
%      LCCIUSERINTERFACE('CALLBACK',hObject,eventdata,handles,...) calls the local
%      function named CALLBACK in LCCIUSERINTERFACE.M with the given input arguments.
%
%      LCCIUSERINTERFACE('Property','Value',...) creates a new LCCIUSERINTERFACE or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before LCCIUserInterface_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to LCCIUserInterface_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 LCCIUserInterface

% Last Modified by GUIDE v2.5 11-Feb-2020 20:30:34

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @LCCIUserInterface_OpeningFcn, ...
                   'gui_OutputFcn',  @LCCIUserInterface_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 LCCIUserInterface is made visible.
function LCCIUserInterface_OpeningFcn(hObject, ~, 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 LCCIUserInterface (see VARARGIN)

% Choose default command line output for LCCIUserInterface
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

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

set(handles.pushbutton1,'String','Select');
set(handles.pushbutton2,'String','Start')
set(handles.pushbutton3,'String','Stop')
set(handles.pushbutton4,'String','Play')
set(handles.pushbutton5,'String','Save')
set(handles.pushbutton6,'String','Preview')
set(handles.pushbutton7,'String','Exit')
set(handles.pushbutton8,'String','Submit')
set(handles.pushbutton9,'String','Reset')
set(handles.edit1,'String','Select Audio File');
set(handles.edit2,'String','Record Audio');
set(handles.edit4,'String','');
set(handles.edit5,'String','');
set(handles.edit5, 'Min', 0, 'Max', 2);
set(handles.text2,'String','Carrier Frequency in kHz [30 kHz to 100 kHz]');
set(handles.text3,'String','Timestamp');
set(handles.checkbox1,'String','');
set(handles.checkbox1,'value',0)
global timestamp
timestamp = 0;
global no_recording
no_recording = 0;
global carrier
```

```
carrier = 0;
% --- Outputs from this function are returned to the command line.
function varargout = LCCIUserInterface_OutputFcn(~, ~, 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 during object creation, after setting all properties.
function edit1_CreateFcn(hObject, ~, ~)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit 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

% --- Executes on button press for Select Button.
function pushbutton1_Callback(~, ~, handles) % Select Button To Search For Audio File
[file,~] = uigetfile('*.*'); % Open File Explorer
    if file ~= 0 % File was selected
    set(handles.edit1,'String',file); % File Name Entered Into Text Box
    global selected_file
    selected_file = file; % File Name Stored into Global Variable To Be Used In Other↙
Functions
    global no_recording
    no_recording = 1; % No Audio Recording
    global select_record
    select_record = 0;
end

% --- Executes on button press for Start Button.
function pushbutton2_Callback(~, ~, handles) % Begin Recording
global recObj
recObj = audiorecorder(44100,16,2) ; % Record Audio
record(recObj); % Start Recording
set(handles.edit2,'String','Recording'); % Text Indicator That Recording Has Started


% --- Executes on button press for Select Button.
function pushbutton3_Callback(~, ~, handles) % Stop Recording
global recObj
global recording
stop(recObj) % Stop
```

```
recording = getaudiodata(recObj); % Store Recording
set(handles.edit2,'String','Recording Stopped'); % Text Indicator That Recording Has↙
Stopped


% --- Executes on button press for Play Button.
function pushbutton4_Callback(~, ~, handles) % Play Recording
global recording
audiowrite('recording.wav',recording,44100) % Temporary Saves Audio Recording into audio↙
file 'recording.wav'
[y,fs]=audioread('recording.wav');
sound(y,fs)
set(handles.edit2,'String','Playing Recording'); % Text Indicator That Recording Is↙
Playing

function edit2_Callback(~, ~, ~)
% hObject    handle to edit2 (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 edit2_CreateFcn(hObject, ~, ~)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit 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

% --- Executes on button press for Save Button.
function pushbutton5_Callback(~, ~, handles) % Save Recording
global recording
global record_file_name
global select_record
set(handles.edit2,'String','Save Recorded Audio');
audiowrite('recording.wav',recording,44100) % Redudancy In Case User Did Not Choose To↙
Play Recording
[y,fs]=audioread('recording.wav');
delete 'recording.wav' % Deleted So Only User Saved Version Remains
[nfname,npath]=uiputfile('.wav','Save sound','recording.wav'); % Name of Saved .wav File
if isequal(nfname,0) || isequal(npath,0)
    return
    else
nwavfile=fullfile(npath, nfname);
nwavfile = cast(nwavfile,'char'); % Needed To Convert File Name To Character Array
audiowrite(nwavfile,y,fs);
record_file_name = nwavfile; % Global Variable To Store Recorded File Name
```

```
select_record = 1;
set(handles.edit2,'String','Saved'); % Text Indicator That Recording Is Saved
end

% --- Executes on button press for Preview Button.
function pushbutton6_Callback(~, ~, ~)
global selected_file
[y,Fs] = audioread(selected_file);
sound(y,Fs);
pause(10); % 15 Second Preview of Selected Audio File
clear sound; % Stop Audio

% --- Executes on entry of Carrier Frequency.
function edit4_Callback(hObject, ~, handles)
global error
global carrier
global carrier_frequency
carrier_frequency = str2double(get(hObject,'String')); % Global Variable to store User↙
Specifed Carrier Frequency
if carrier_frequency >= 30 && carrier_frequency <= 100 % Specified Carrier Frequency Not↙
Supported By Device
    set(handles.edit5,'String','');
    error = 0; % Global Variable To Determine if Error Exists
    carrier = 1;
else
    set(handles.edit5,'String','Specified Carrier Frequency is not within the parameters↙
of 30 kHz to 100 kHz');
    error = 1; % Error Exits -> Not Able to Submit
end

function edit4_CreateFcn(hObject, ~, ~)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit 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

% --- Executes during object creation, after setting all properties.
function text2_CreateFcn(~, ~, ~)
% hObject    handle to text2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

function edit5_Callback(~, ~, ~) %Display Error, Audio Information, Timestampy
% hObject    handle to edit5 (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 edit5_CreateFcn(hObject, ~, ~)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit 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

% --- Executes during object creation, after setting all properties.
function text3_CreateFcn(~, ~, ~)
% hObject    handle to text3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes on button press for Timestamp Checkmark.
function checkbox1_Callback(~, ~, ~)
global timestamp
if timestamp == 0
    timestamp = 1; % Timestamp Requested
elseif timestamp == 1
    timestamp = 0;
end

% --- Executes on button press of Exit Button.
function pushbutton7_Callback(~, ~, ~)
closereq();

% --- Executes on button press on Submit Button.
function pushbutton8_Callback(~, ~, handles)
% Main Function that uses all Global Variables
global selected_file
global record_file_name
global select_record
global error
global carrier
if error == 1 && carrier == 0 % Error Occured OR Carrier Frequency Not Specified
    return % No Submission
end
if select_record == 1 % Selected Recorded File
    info = audioinfo(record_file_name);
    submit_file = record_file_name;
else % Selected Preexisting Audio File
    info = audioinfo(selected_file);
    submit_file = selected_file;
```

```
end
% Code Founded Online to Display Relevant Audio Information within Text Box
field = fieldnames(info);
data  = struct2cell(info);
audio_info_str  = cell(1, numel(field));
for k = 1:numel(field)
  if ischar(data{k})
    audio_info_str{k} = sprintf('%s: %s', field{k}, data{k});
  elseif isnumeric(data{k})
    if isempty(data{k})
      audio_info_str{k} = sprintf('%s: []', field{k});
    else
      audio_info_str{k} = sprintf('%s: %s', field{k}, num2str(data{k}));
    end
  else
    audio_info_str{k} = sprintf('%s: [%s]', field{k}, class(data{k}));
  end
end
global timestamp
if timestamp == 1  % Timestamp Requested
    t = datestr(datetime('now')); % Timestamp For Research Purposes
    audio_info_str{k+1} = sprintf('%s', t);
    timestamp = 0;
    set(handles.checkbox1,'value',0)
end
set(handles.edit5, 'String', audio_info_str);
% Save .wav File onto SD Card
[y,fs]=audioread(submit_file); % Name of selected .wav file
[nfname,npath]=uiputfile('.wav','Save onto SD Card',submit_file); % Name of Saved .wav↙
File In New Location
if isequal(nfname,0) || isequal(npath,0)
    return   % or whatever other action if 'CANCEL'
    else
nwavfile=fullfile(npath, nfname);
nwavfile=strcat(nwavfile(1:end-3),'wav'); % Changes File To .wav Audio File
nwavfile = cast(nwavfile,'char'); % Needed To Convert File Name To Character Array
audiowrite(nwavfile,y,fs);
timestamp = 0; % Reset Timestamp Parameter
end

% --- Executes on button press on Reset Button.
function pushbutton9_Callback(~, ~, handles)
% Resets Entire User Interface
global timestamp
set(handles.checkbox1,'value',0)
set(handles.edit1,'String','Select Audio File');
set(handles.edit2,'String','Record Audio');
set(handles.edit4, 'String', '');
set(handles.edit5, 'String', '');
clearvars -global
clearvars
```

```
timestamp = 0;
```