```
% Name: Aditya Gole
% MIS: 111909037
% Microcontroller project

function varargout = LDR_GUI(varargin)
%LDR_GUI MATLAB code file for LDR_GUI.fig
%      LDR_GUI, by itself, creates a new LDR_GUI or raises the existing
%      singleton*.
%
%      H = LDR_GUI returns the handle to a new LDR_GUI or the handle to
%      the existing singleton*.
%
%      LDR_GUI('Property','Value',...) creates a new LDR_GUI using the
%      given property value pairs. Unrecognized properties are passed via
%      varargin to LDR_GUI_OpeningFcn.  This calling syntax produces a
%      warning when there is an existing singleton*.
%
%      LDR_GUI('CALLBACK') and LDR_GUI('CALLBACK',hObject,...) call the
%      local function named CALLBACK in LDR_GUI.M with the given input
%      arguments.
%
%      *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 LDR_GUI

% Last Modified by GUIDE v2.5 03-Nov-2021 12:34:53

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @LDR_GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @LDR_GUI_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 LDR_GUI is made visible.
function LDR_GUI_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   unrecognized PropertyName/PropertyValue pairs from the
%            command line (see VARARGIN)

handles.red = 'D9';
handles.green = 'D8';
handles.trigger = 'D6';
handles.echo = 'D5';
handles.buzzPin = 'D3';
handles.button = 'D11';
handles.led = 'D12';
% Choose default command line output for LDR_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

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


% --- Outputs from this function are returned to the command line.
function varargout = LDR_GUI_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 Plot_data_CreateFcn(~, ~, ~)
% hObject    handle to Plot_data (see GCBO)
% handles    empty - handles not created until after all CreateFcns called

% --- Executes on button press in start.
function start_Callback(hObject, ~, handles)
% hObject    handle to start (see GCBO)
% handles    structure with handles and user data (see GUIDATA)
delete(instrfind({'PORT'}, {'COM6'}));
clear a;
clc;
global a;

handles.a = arduino('COM6', 'UNO', 'Libraries','Ultrasonic');
if ~isempty(handles.a) == 1
    set(handles.start, 'BackgroundColor', [0 1 0]); % when board is on button turns↙
green
end

if(handles.button == 1)
```

```matlab
    handles.a.writeDigitalPin(handles.led, 0);
end

if(handles.button == 0)
    handles.a.writeDigitalPin(handles.led, 1);
end
guidata(hObject, handles);

% --- Executes on button press in exit.
function exit_Callback(~, ~, handles)
% hObject    handle to exit (see GCBO)
% handles    structure with handles and user data (see GUIDATA)

set(handles.start, 'BackgroundColor', [0.94 0.94 0.94]);
set(handles.exit, 'BackgroundColor', [1 0 0]); % when board is off button turns red

pause(2);
clc;
clear handles.a; % disconnect the board from matlab
closereq();


% --- Executes on button press in simulate.
function simulate_Callback(~, ~, handles)
x = 0;
global i
for i=1:1:handles.time
   h =  handles.a.readVoltage('A0');
   x = [x, h];
   plot(x, 'LineWidth', 2); grid on;
   xlabel('Time(s)----->');ylabel('Voltage(V)----->');
   axis([i-100 i+100 0 5]);
   pause(0.0001)

   if (h>1.5) %when voltage>1.5 green led is on and light on box turns green
       handles.a.writeDigitalPin(handles.green, 1);
       handles.a.writeDigitalPin(handles.red, 0);
       set(handles.light_on, 'BackgroundColor', [0 1 0]); % green color
       set(handles.light_off, 'BackgroundColor', [1 1 1]);
   end

   if (h<1.5) %when voltage<1.5 red led on and light off box turns red
       handles.a.writeDigitalPin(handles.green, 0);
       handles.a.writeDigitalPin(handles.red, 1);
       set(handles.light_off, 'BackgroundColor', [1 0 0]); % red color
       set(handles.light_on, 'BackgroundColor', [1 1 1]);
   end
end


function time_value_Callback(hObject, ~, handles)

% Hints: get(hObject,'String') returns contents of time_value as text
%        str2double(get(hObject,'String')) returns contents of time_value as a↙
```

```
double
handles.data1 = get(hObject, 'string');
handles.time = str2double(handles.data1);
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function time_value_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'), get↙
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in measure_distance.
function measure_distance_Callback(~, ~, handles)

%arduinoObj = arduino('COM6','Uno','Libraries','Ultrasonic');
handles.ultra = ultrasonic(handles.a ,handles.trigger,handles.↙
echo,'OutputFormat','double');

while(1)

    handles.time = readEchoTime(handles.ultra);
    handles.TravelDistance = (340*handles.time/2)*100; % Distance in cm
    set(handles.distance, 'String', handles.TravelDistance);

    if (handles.TravelDistance < 5.000) %if distance < 5cm play buzzer
       set(handles.distance, 'String', handles.TravelDistance);
       handles.a.writeDigitalPin(handles.buzzPin, 1);
       pause(0.1)
       set(handles.warning,'BackgroundColor', [1 0 0], 'String', "WARNING");
       %handles.TravelDistance = (340*handles.time/2)*100; % Distance in cm
    else
        handles.a.writeDigitalPin(handles.buzzPin, 0);
        pause(0.1)
        set(handles.warning,'BackgroundColor', [0.07 0.62 1], 'String', " ");
    end

end

%
```