

## Exercises 4: MAE Unit 4

**Name:** Josep Famadas Alsamora

**Collaboration Info:** I have used some information from mathworks.com.

---

### Exercise 1. DTMF Signal Generator and GUI.

#### Explanation:

##### (1) DTMF\_Keypad:

First of all, using the GUIDE, I designed the graphic interface of the keypad, with one static text box and 16 + 4 push buttons separated in a 4x4 grid and in a 2x2 grid.

When the interface was ready I started to configure the .m file. First, I created a variable which was going to contain the text (handles.str) and its maximum length (handles.max\_length). Then, I set the function of the different push buttons.

- All the 4x4 grid buttons do a similar function when pressed, they concatenate their own value to the handles.str, (0-9, A-D, \* or #) and copy this value to the String feature of the Display. If the handles.str was empty before pressing the button it enables the DEL and Send buttons. If after pressing the button handles.str reaches the max length, all the 4x4 buttons are disabled.
- The DEL button deletes the last character from handles.str. If now handles.str is empty it disables DEL and Send. If handles.str was full before pressing DEL it enables the 4x4 buttons.
- The Send button calls the function DTMF\_Signal with handles.str as a parameter and erases all the content of the variable disabling the buttons DEL and Send. If before pressing it, handles.str was full it enables the 4x4 buttons.
- The Close button closes all the opened figures.
- The Help button calls the function About\_DTMF\_Keypad.

##### (2) DTMF\_Signal:

First of all, using the GUIDE, I designed the graphic interface of the figure, with one axis, 2 push buttons and a slide bar.

When the interface was ready I started to configure the .m file. First, I modified the opening function so when called, it computes a signal consisting in series of N pairs of added sines (length = 300ms and frequencies the associated with each of the input string characters) with a separation between deferens pairs of 100ms, being N the number of input string characters.

Once the signal has been computed, it plots the whole signal but showing only its first 125ms, with the possibility of showing the res by moving the slide bar.

After that, I configured the 2 buttons and the sidebar:

- The Close button closes the current figure.
- The Play button plays the signal using the function 'soundsc'.
- The sidebar is configured in order to get its current position and modify the xlims of the axis to this value and this value + 125ms.

### (3) About\_DTMF\_Keypad

First of all, using the GUIDE, I designed the graphic interface of the figure, with one static text with the text “A toy example of Matlab GUI programming J. Famadas 2016” and a push button.

When the interface was ready I started to configure the .m file. I just configure the Close button in order to close the current figure when pushed.

#### MATLAB Code:

(1) DTMF\_Keypad: (There is only the code for 1 of the 4x4 buttons because the other are very similar and would just occupied space).

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

% Last Modified by GUIDE v2.5 23-Nov-2016 18:35:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @DTMF_Keypad_OpeningFcn, ...
                  'gui_OutputFcn',  @DTMF_Keypad_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 DTMF_Keypad is made visible.
function DTMF_Keypad_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 DTMF_Keypad (see VARARGIN)

% Choose default command line output for DTMF_Keypad
handles.output = hObject;
handles.str = '';
handles.max_length = 20;
% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout = DTMF_Keypad_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 button_1.
function button_1_Callback(hObject, eventdata, handles)
% hObject    handle to button_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
if(length(handles.str)==0)
    set(handles.button_DEL,'Enable','on');
    set(handles.button_Send,'Enable','on');
end
handles.str = [handles.str '1'];
set(handles.Display_str,'String',handles.str);
if(length(handles.str)==handles.max_length)
    set(handles.button_1,'Enable','off');
    set(handles.button_2,'Enable','off');
    set(handles.button_3,'Enable','off');

```

```

        set(handles.button_4,'Enable','off');
        set(handles.button_5,'Enable','off');
        set(handles.button_6,'Enable','off');
        set(handles.button_7,'Enable','off');
        set(handles.button_8,'Enable','off');
        set(handles.button_9,'Enable','off');
        set(handles.button_0,'Enable','off');
        set(handles.button_A,'Enable','off');
        set(handles.button_B,'Enable','off');
        set(handles.button_C,'Enable','off');
        set(handles.button_D,'Enable','off');
        set(handles.button_hashtag,'Enable','off');
        set(handles.button_asterisc,'Enable','off');
    end
    guidata(hObject,handles);

% --- Executes on button press in button_DEL.
function button_DEL_Callback(hObject, eventdata, handles)
% hObject      handle to button_DEL (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
if(length(handles.str)==handles.max_length)
    set(handles.button_1,'Enable','on');
    set(handles.button_2,'Enable','on');
    set(handles.button_3,'Enable','on');
    set(handles.button_4,'Enable','on');
    set(handles.button_5,'Enable','on');
    set(handles.button_6,'Enable','on');
    set(handles.button_7,'Enable','on');
    set(handles.button_8,'Enable','on');
    set(handles.button_9,'Enable','on');
    set(handles.button_0,'Enable','on');
    set(handles.button_A,'Enable','on');
    set(handles.button_B,'Enable','on');
    set(handles.button_C,'Enable','on');
    set(handles.button_D,'Enable','on');
    set(handles.button_hashtag,'Enable','on');
    set(handles.button_asterisc,'Enable','on');
end
handles.str = handles.str(1:end-1);
set(handles.Display_str,'String',handles.str);
if(length(handles.str)==0)
    set(handles.button_DEL,'Enable','off');
    set(handles.button_Send,'Enable','off');
end
guidata(hObject,handles);

% --- Executes on button press in button_Send.
function button_Send_Callback(hObject, eventdata, handles)
% hObject      handle to button_Send (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
DTMF_Signal(handles.str);
if(length(handles.str)==handles.max_length)
    set(handles.button_1,'Enable','on');
    set(handles.button_2,'Enable','on');
    set(handles.button_3,'Enable','on');
    set(handles.button_4,'Enable','on');
    set(handles.button_5,'Enable','on');
    set(handles.button_6,'Enable','on');

```

```

        set(handles.button_7,'Enable','on');
        set(handles.button_8,'Enable','on');
        set(handles.button_9,'Enable','on');
        set(handles.button_0,'Enable','on');
        set(handles.button_A,'Enable','on');
        set(handles.button_B,'Enable','on');
        set(handles.button_C,'Enable','on');
        set(handles.button_D,'Enable','on');
        set(handles.button_hashtag,'Enable','on');
        set(handles.button_asterisc,'Enable','on');
    end
    handles.str = '';
    set(handles.Display_str,'String',handles.str);
    set(handles.button_DEL,'Enable','off');
    set(handles.button_Send,'Enable','off');
    guidata(hObject,handles);

% --- Executes on button press in button_Close.
function button_Close_Callback(hObject, eventdata, handles)
% hObject    handle to button_Close (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;

% --- Executes on button press in button_Help.
function button_Help_Callback(hObject, eventdata, handles)
% hObject    handle to button_Help (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
About_DTMF_Keypad()

```

## (2) DTMF\_Signal:

```

function varargout = DTMF_Signal(varargin)
% DTMF_SIGNAL MATLAB code for DTMF_Signal.fig
%     DTMF_SIGNAL, by itself, creates a new DTMF_SIGNAL or raises the
%     existing
%     singleton*.
%
%     H = DTMF_SIGNAL returns the handle to a new DTMF_SIGNAL or the
%     handle to
%     the existing singleton*.
%
%     DTMF_SIGNAL('CALLBACK',hObject,eventData,handles,...) calls the
%     local
%     function named CALLBACK in DTMF_SIGNAL.M with the given input
%     arguments.
%
%     DTMF_SIGNAL('Property','Value',...) creates a new DTMF_SIGNAL
%     or raises the

```

```

%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before DTMF_Signal_OpeningFcn gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to DTMF_Signal_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 DTMF_Signal

% Last Modified by GUIDE v2.5 23-Nov-2016 18:46:36

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @DTMF_Signal_OpeningFcn, ...
                  'gui_OutputFcn',  @DTMF_Signal_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 DTMF_Signal is made visible.
function DTMF_Signal_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 DTMF_Signal (see VARARGIN)

% Choose default command line output for DTMF_Signal
handles.output = hObject;
A = strjoin(varargin{1});
X = [];
handles.fs = 8000;
%Signal computing
for i=1:length(A)

    t = (1/(0.3*handles.fs):1/(0.3*handles.fs):1)*0.3;

    switch A(i)

```

```

        case '1'
            f1 = 697;
            f2 = 1209;
        case '2'
            f1 = 697;
            f2 = 1336;
        case '3'
            f1 = 697;
            f2 = 1477;
        case '4'
            f1 = 770;
            f2 = 1209;
        case '5'
            f1 = 770;
            f2 = 1336;
        case '6'
            f1 = 770;
            f2 = 1477;
        case '7'
            f1 = 852;
            f2 = 1209;
        case '8'
            f1 = 852;
            f2 = 1336;
        case '9'
            f1 = 852;
            f2 = 1477;
        case '0'
            f1 = 941;
            f2 = 1336;
        case '*'
            f1 = 941;
            f2 = 1209;
        case '#'
            f1 = 941;
            f2 = 1477;
        case 'A'
            f1 = 697;
            f2 = 1633;
        case 'B'
            f1 = 770;
            f2 = 1633;
        case 'C'
            f1 = 852;
            f2 = 1633;
        case 'D'
            f1 = 941;
            f2 = 1633;
    end

    signal = sin(2*pi*f1*t) + sin(2*pi*f2*t);
    X = [X signal zeros(1,0.1*8000)];

end

X = X(1:end-800);
handles.vectorX = X;
x = 1/handles.fs:1/handles.fs:length(handles.vectorX)/handles.fs;
plot(x,handles.vectorX,'Parent',handles.axes1)
xlim(handles.axes1,[1/handles.fs 1000/handles.fs])

%slide bar initialization

```

```

set(handles.slider1,'Max',(length(handles.vectorX)-
1000)/handles.fs,'Min',0,'Value',0);

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout = DTMF_Signal_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 button_Close.
function button_Close_Callback(hObject, eventdata, handles)
% hObject      handle to button_Close (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close;

% --- Executes on button press in button_Play.
function button_Play_Callback(hObject, eventdata, handles)
% hObject      handle to button_Play (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
soundsc(handles.vectorX);
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)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
xlim(handles.axes1,[get(hObject,'Value')
get(hObject,'Value')+1000/handles.fs]);
guidata(hObject, handles);

% --- 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
```

### (3) About\_DTMF\_Keypad

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

% Last Modified by GUIDE v2.5 23-Nov-2016 18:36:53

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @About_DTMF_Keypad_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @About_DTMF_Keypad_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
```

```

        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before About_DTMF_Keypad is made visible.
function About_DTMF_Keypad_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 About_DTMF_Keypad (see
VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout = About_DTMF_Keypad_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)

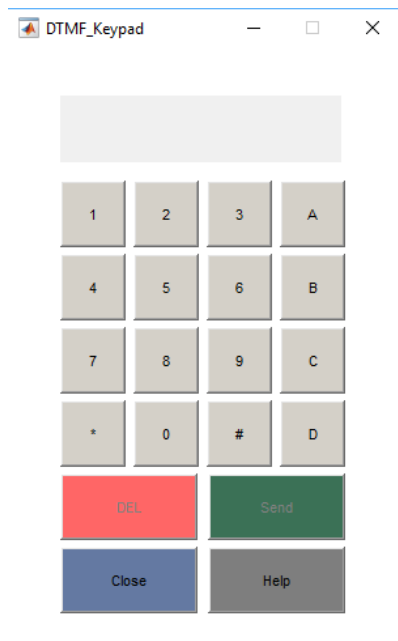
% Get default command line output from handles structure
varargout{1} = handles.output;

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

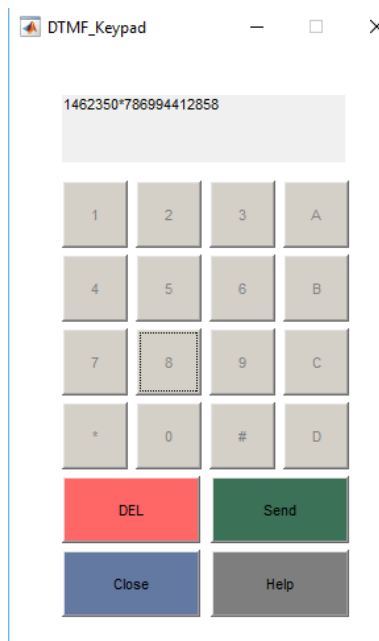
```

## Results:

### (1) DTMF\_Keypad:

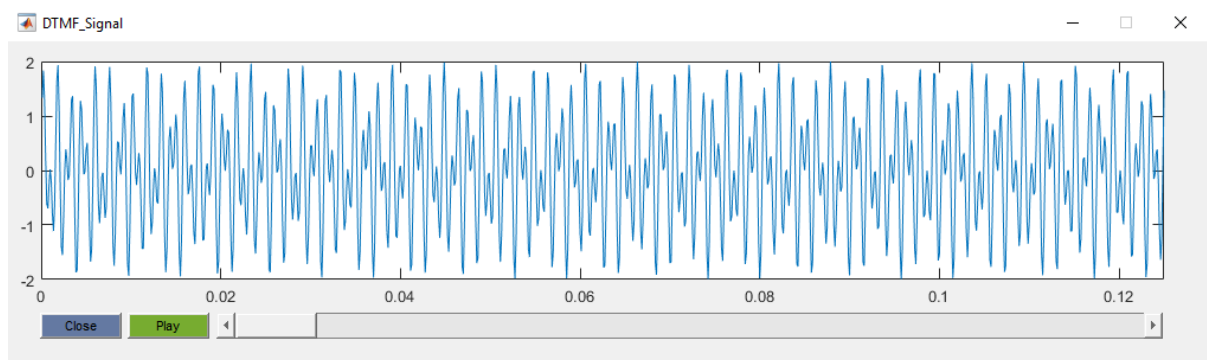


Empty

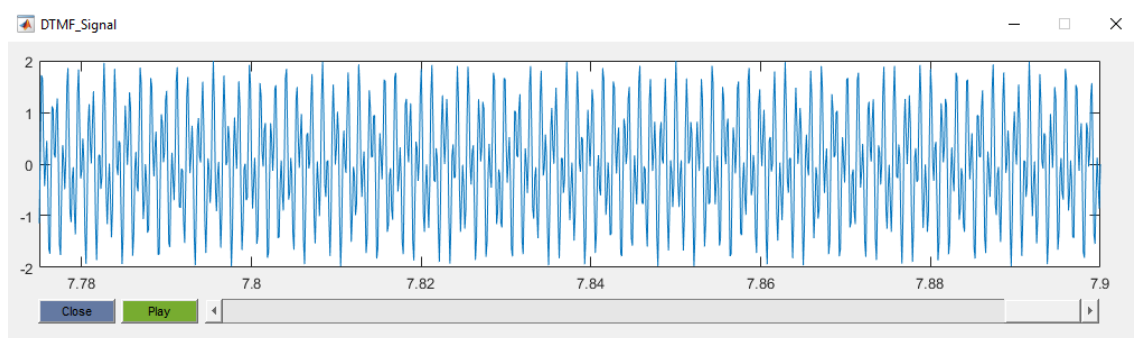


Full

### (2) DTMF\_Signal: (Example of Send with 20 Characters)



Here we can see that the  $t_{min} = 0s$ .



Here we can see that the  $t_{max} = 20 \cdot 0.3 + 19 \cdot 0.1 = 7.9s$ .

(3) About\_DTMF\_Keypad



Coments: