

## TERM PROJECT REPORT

Nebile Şevval Karahan - 2443299

Ahmet Alp Güneş – 2443190

## INTRODUCTION

By transmitting data across network devices, signaling is essential to the telecommunications industry as it helps to facilitate communication sessions. The development of signaling techniques over time has been crucial in raising the effectiveness and potential of phone communications. In the 1960s, Dual-Tone Multi-Frequency (DTMF) signaling replaced rotary dialing systems, and this study explores this important technological development. In contrast to its predecessor, DTMF signaling greatly increases the speed and adaptability of phone conversations by encoding each key using a distinct combination of high- and low-frequency tones. This study provides a thorough examination of the fundamental ideas and real-world uses of this crucial component of telecommunication technology. It describes the design and implementation of a MATLAB-based system for producing, transmitting, receiving, and decoding audio DTMF signals. Users can manipulate parameters, generate DTMF signals, and decode received audio through a Graphical User Interface (GUI), providing a hands-on investigation of the complexities of DTMF signaling in contemporary communication systems.

## I) TRANSMITTER

In the first part of the project, we are asked to create the DTMF encoded sequence of any given input signal written by the keypad. To achieve this, we first create a GUI using MATLAB App Designer, as can be seen in Figure 1 and Figure 2. In our GUI, the user must decide on a transmitter or receiver panel.

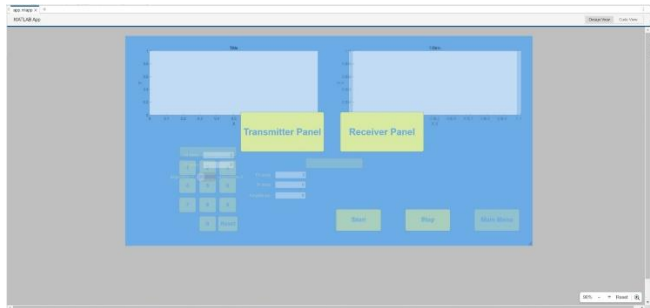


Figure-1: GUI created by using MATLAB App Designer



Figure-2: Main Menu

A keypad, two axes, and additional input parameters are displayed on the screen in Figure 3 following the pressing of the transmitter panel button. Users can generate a DTMF encode for any sequence they enter using the keypad on this page. By using the save and play buttons, users can also create and listen to a time domain signal and spectrogram of this encoded signal.

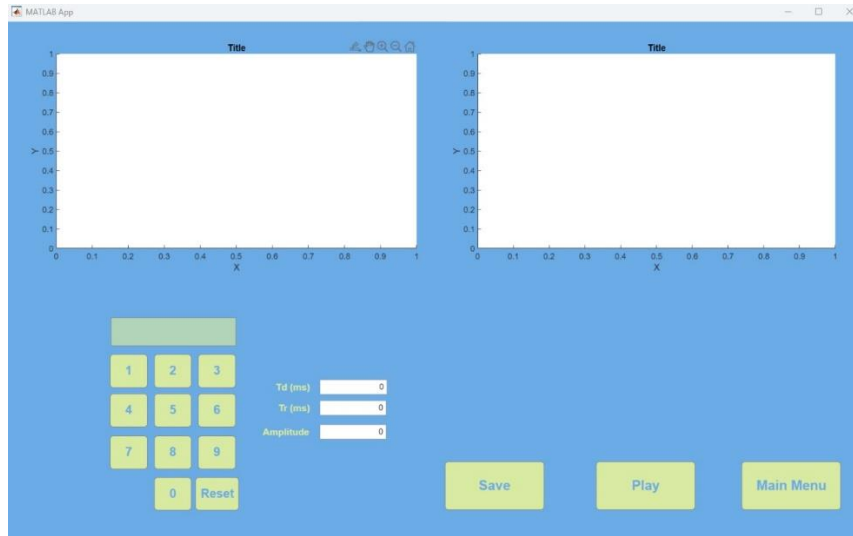


Figure-3: Transmitter Page

The user must first enter the sequence using the GUI keypad, as shown in Figure 4. It is then necessary to enter the  $T_d$ ,  $T_r$ , and amplitude parameters before clicking the Save button. The user will see a spectrogram and time-domain encoded signal on the axis at the moment they click the Save button. The user simply needs to click the Play button in order to listen to this time domain signal.

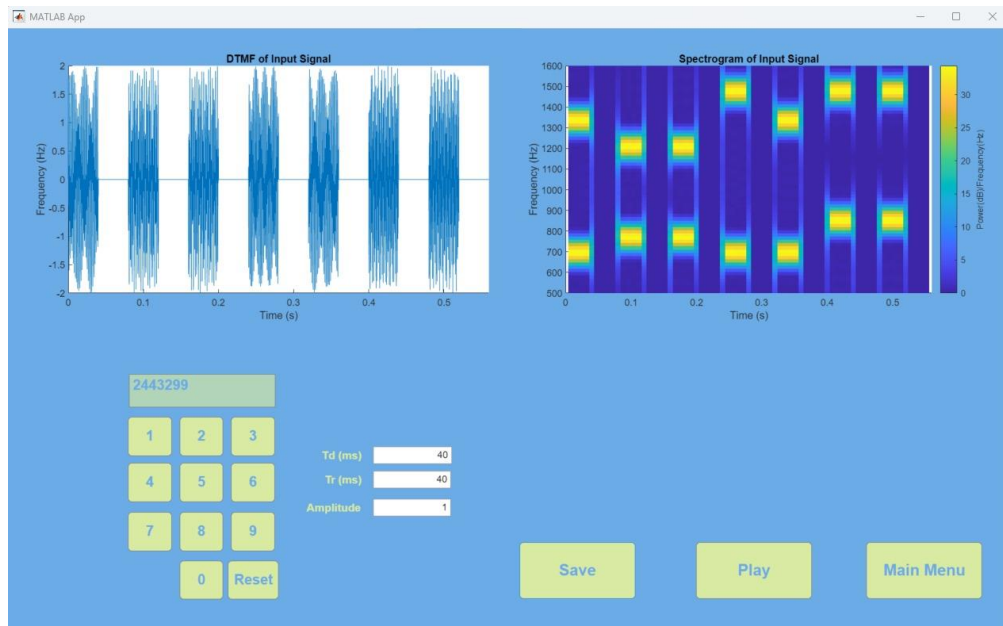


Figure-4: Generated DTMF encoded Time-Domain signal and Spectrogram of the sequence 2443299 for  $T_d$  and  $T_r$  are 40 msec

Background of the GUI, we first recorded the given sequence in number buttons as in Figure 5. After all the numbers are given, in the save button we mapped the numbers to corresponding high and low frequencies in lines between 416 and 428 in Figure 6. When we map all the frequencies with the numbers given, we calculated the encoded DTMF sequence according to below two formulas.

$$s^{(k)}(t; T_d) = (\sin(f_L^{(k)} t) + \sin(f_H^{(k)} t)) \cdot (u(t) - u(t - T_d))$$

$$m(t; T_d, T_r) = \sum_{k=0}^{N-1} s^{(x[k])}(t - k(T_d + T_r); T_d)$$

where  $T_r$  is the resting duration between two consecutively pressed keys and  $T_d$  is the signaling duration per key. In the last part of the code section, we used built-in spectrogram method to plot graph on the axes.

```

211 function Button_5Pushed(app, event)
212     app.PhoneNumber = strcat(app.PhoneNumber, '5');
213     app.TextArea.Value = app.PhoneNumber;

```

Figure-5: Recording Input Number to a String

```

405 % Button pushed function: SaveButton
406 function SaveButtonPushed(app, event)
407     app.NumberArray = str2num(app.PhoneNumber.');
408     app.InputLength = strlen(app.PhoneNumber);
409     app.m = 0;
410     app.LowFrequencyMatrix = [697 770 852 941];
411     app.HighFrequencyMatrix = [1209 1336 1477];
412     max = app.InputLength*(app.Td + app.Tr);
413
414     Fs = 8000;
415     for i=1:app.InputLength
416         k = app.NumberArray(i);
417         if mod(k,3)~=0 && k~=0
418             fL = app.LowFrequencyMatrix(floor(k/3.1)+1);
419             fH = app.HighFrequencyMatrix(mod(k,3));
420         elseif k==0
421             fL = app.LowFrequencyMatrix(4);
422             fH = app.HighFrequencyMatrix(2);
423         else
424             fL = app.LowFrequencyMatrix(floor(k/3.1)+1);
425             fH = app.HighFrequencyMatrix(3);
426         end
427         t = 0:(1/8000):max;
428         phase_rad_low = fL*(i-1)*(app.Td+app.Tr)*pi;
429         phase_rad_high = fH*(i-1)*(app.Td+app.Tr)*pi;
430         angle_low = 2*pi*fL*t;
431         angle_high = 2*pi*fH*t;
432         sin_low = app.Amplitude* sin(angle_low-phase_rad_low);
433         sin_high = app.Amplitude* sin(angle_high-phase_rad_high);
434         sin_sum = sin_low + sin_high;
435         shifted_unit_step = (t>=(i-1)*(app.Td+app.Tr))&(t<=((i-1)*(app.Td+app.Tr)+app.Td));
436         s = sin_sum .* shifted_unit_step;
437         app.m = app.m + s;
438     end
439     axes(app.UIAxes_2);
440     if app.Td < 100
441         [sp,f,ts]=spectrogram(app.m, hamming(128), 64,512,Fs);
442     else
443         [sp,f,ts]=spectrogram(app.m, hamming(256), 128,512,Fs);
444     end
445 end

```

Figure-6: Encoding the Input Sequence

In the play button we simply used sound method to listen the time domain signal as can be seen in Figure 7.

```

65 % Button pushed function: PlayButton
66 function PlayButtonPushed(app, event)
67     Fs = 8000;
68     sound(app.m, Fs);

```

Figure-7: Playing the Encoded Signal

Based on our results we did not observe any disturbances in the transmitter side of the app. We are able to encode any signal in the test scenarios ( $(T_d, T_r) \in \{(250ms, 250ms), (100ms, 100ms), (40ms, 40ms)\}$ ) as can be seen in Figures 4, 8 and 9.

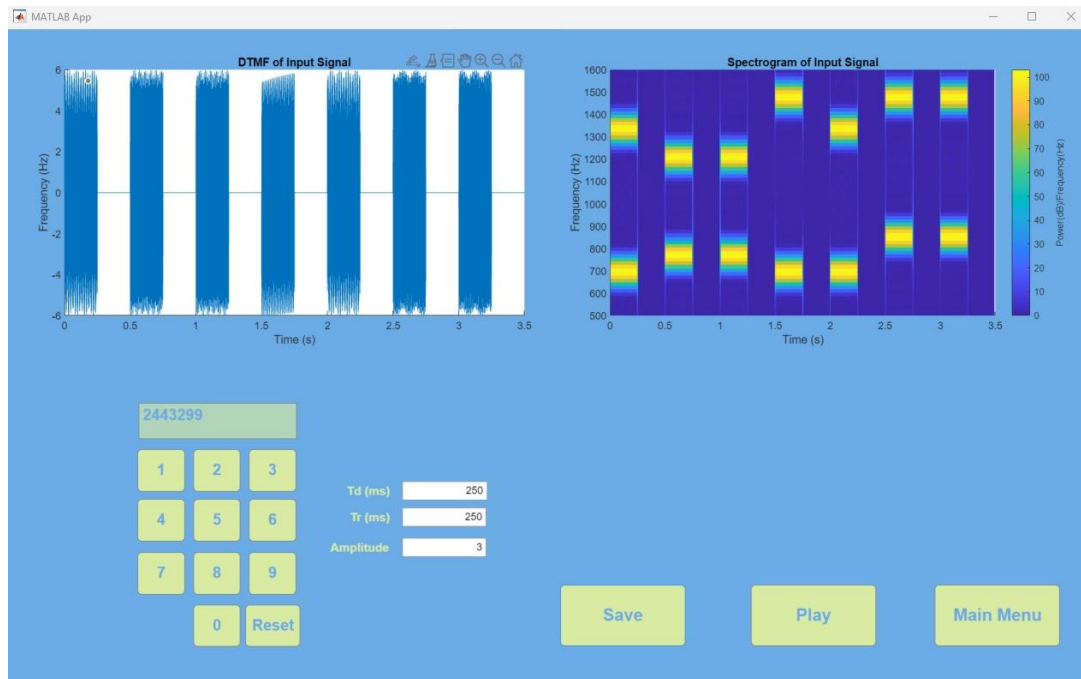


Figure-8: Encoded Signal  $T_d, T_r$  are 250 msec

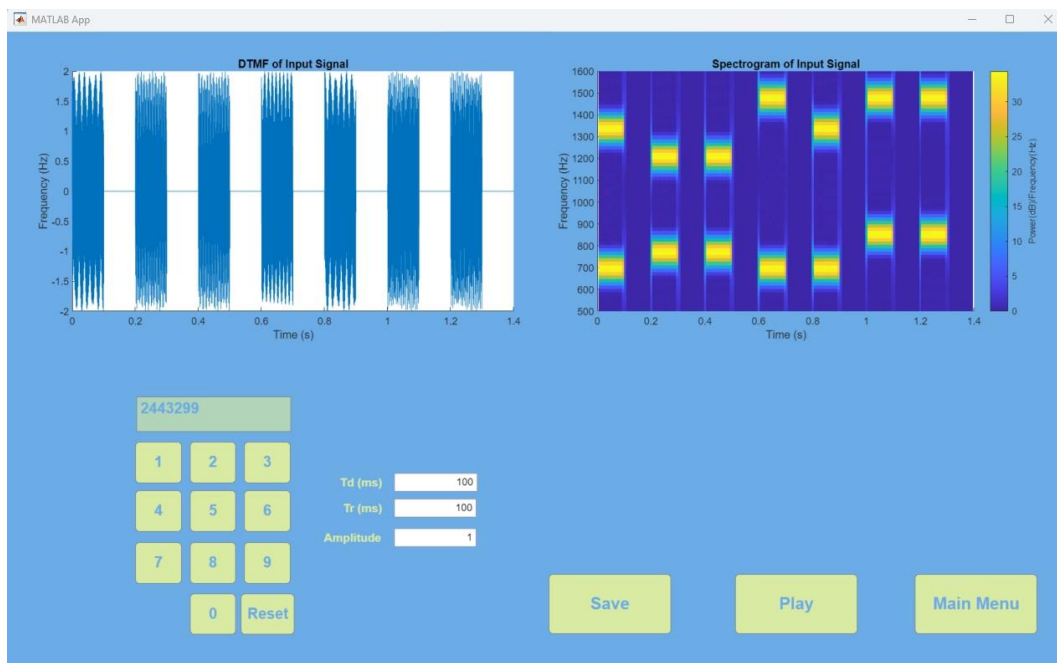


Figure-9: Encoded Signal  $T_d, T_r$  are 100 msec

## II) RECEIVER

In the second part of the project, we are asked to listen the audio signal via microphone and decode this signal. In receiver panel of GUI there are two axes, start-stop buttons and input parameters as can be seen in Figure 10. In order to encode the listened signal, user must enter the values  $T_d$  and  $T_r$  according to the audio, otherwise app will give an error message.

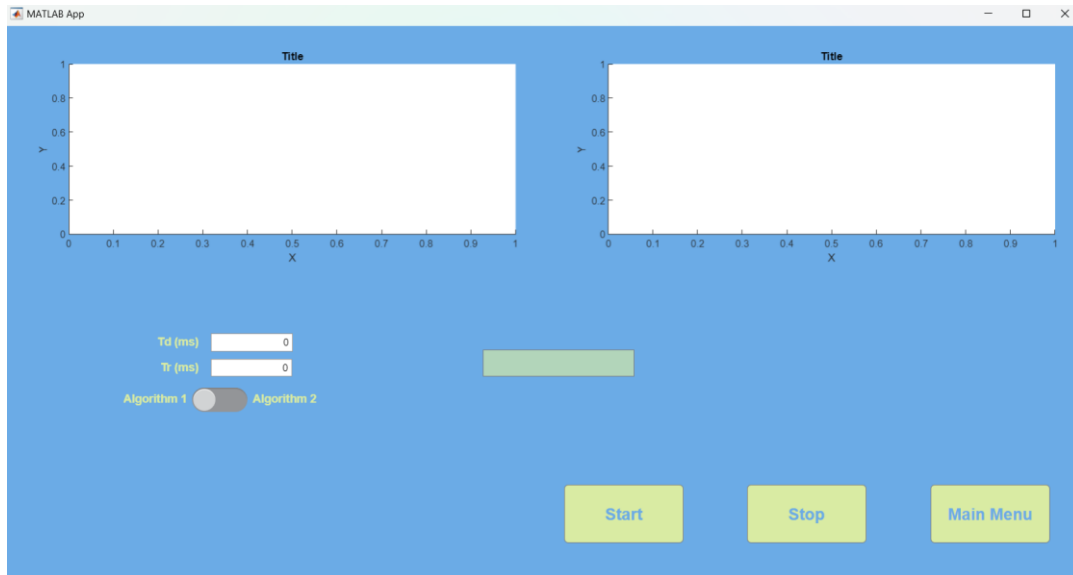


Figure-10: Receiver Panel of GUI

When we play the file recorded by the transmitter,  $T_d$  and  $T_r$  are 250 msec, we get the results in Figure 11. In our algorithm to decode audio we used peak selection from spectrogram method. In the spectrogram the points we want to select have higher power level compared to others. Hence, we set a threshold, and we captured all the points above this value, can be seen in the Figure 12.

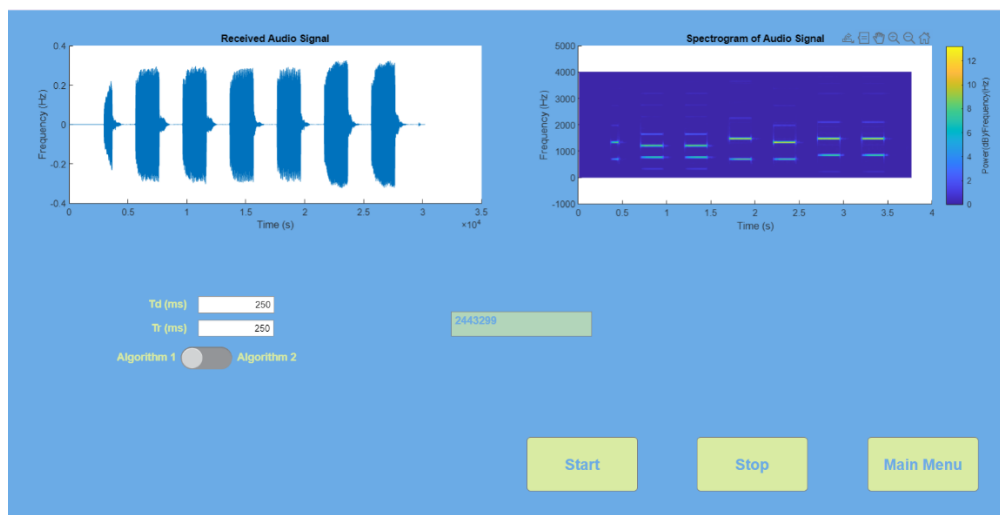


Figure-11: Decoding Results for  $T_d$  and  $T_r$  250 msec

```

266         Td = app.Td2;
267         Tr = app.Tr2;
268         Ts = Td+Tr;
269
270         points = cell(0,2);
271         for i=1:length(ts)-1
272             for j=1:length(f)-1
273                 power = abs(sp(j,i));
274                 if power>5
275                     point=[ts(i) f(j)];
276                     points{end+1}=point;
277                 end
278             end
279         end

```

Figure-12: Selecting the Points above Threshold

After this process we divide the chunks according to their time values and classify the frequency values in the same chunk as high and low frequencies. Following, we calculated the average high and low frequency values for each chunk. Based on our results this calculation gives very close ( $\pm 5$ ) values to real ones. In the last, we compared the calculated frequencies with the given ones and decide the number, this algorithm can be seen in Figure 13 and 14. In line 353 in Figure 14 we print the obtained sequence to the screen.

```

286         start = points{1}(1);
287         for i=2:length(points)
288             t_value = points{i}(1);
289             chunk_num = ceil((t_value-start)/Ts);
290
291             if points{i-1}(1)+Td/2<points{i}(1)
292                 high_frequencies(chunk_num) = (highs/num_highs);
293                 highs = 0;
294                 num_highs = 0;
295
296                 low_frequencies(chunk_num) = (lows/num_lows);
297                 lows = 0;
298                 num_lows = 0;
299             elseif (i==length(points))
300                 high_frequencies(chunk_num) = (highs/num_highs);
301                 highs = 0;
302                 num_highs = 0;
303
304                 low_frequencies(chunk_num) = (lows/num_lows);
305                 lows = 0;
306                 num_lows = 0;
307
308             else
309                 if points{i-1}(2) > 1071
310                     highs = highs+points{i-1}(2);
311                     num_highs = num_highs+1;
312                 else
313                     lows = lows+points{i-1}(2);
314                     num_lows = num_lows+1;
315                 end
316             end
317         end
318         disp('highs');
319         disp(high_frequencies);
320         disp('lows');
321         disp(low_frequencies);
322         app.Digits = '';

```

Figure-13: Calculating Average Frequencies

```

323         for i=1:length(low_frequencies)
324             lowfreq = low_frequencies(i);
325             highfreq = high_frequencies(i);
326
327             if lowfreq>687 && lowfreq<707 && highfreq<1219 && highfreq>1199
328                 app.Digits = strcat(app.Digits,'1');
329             elseif lowfreq>760 && lowfreq<780 && highfreq<1219 && highfreq>1199
330                 app.Digits = strcat(app.Digits,'4');
331             elseif lowfreq>842 && lowfreq<862 && highfreq<1219 && highfreq>1199
332                 app.Digits = strcat(app.Digits,'7');
333             elseif lowfreq>687 && lowfreq<707 && highfreq<1346 && highfreq>1326
334                 app.Digits = strcat(app.Digits,'2');
335             elseif lowfreq>760 && lowfreq<780 && highfreq<1346 && highfreq>1326
336                 app.Digits = strcat(app.Digits,'5');
337             elseif lowfreq>842 && lowfreq<862 && highfreq<1346 && highfreq>1326
338                 app.Digits = strcat(app.Digits,'8');
339             elseif lowfreq>687 && lowfreq<707 && highfreq<1487 && highfreq>1467
340                 app.Digits = strcat(app.Digits,'3');
341             elseif lowfreq>760 && lowfreq<780 && highfreq<1487 && highfreq>1467
342                 app.Digits = strcat(app.Digits,'6');
343             elseif lowfreq>842 && lowfreq<862 && highfreq<1487 && highfreq>1467
344                 app.Digits = strcat(app.Digits,'9');
345             elseif lowfreq>931 && lowfreq<951 && highfreq<1346 && highfreq>1326
346                 app.Digits = strcat(app.Digits,'0');
347             else
348                 app.Digits = strcat(app.Digits,'-');
349             end
350         end
351
352
353         app.TextArea_2.Value = app.Digits;

```

Figure-14: Comparing Calculated and Given Frequencies

In our tests trials we saw that performance of the app is better when  $T_d$  and  $T_r$  values are higher. In other words, our test results show that for shorter duration of resting and signaling per key, this algorithm cannot be used. For  $T_d$  and  $T_r$  values (250, 100, 40) our app performances are 87.5%, 72.5%, 34% respectively. Screenshots for test results can be seen in Figure 11, 15 and 16.

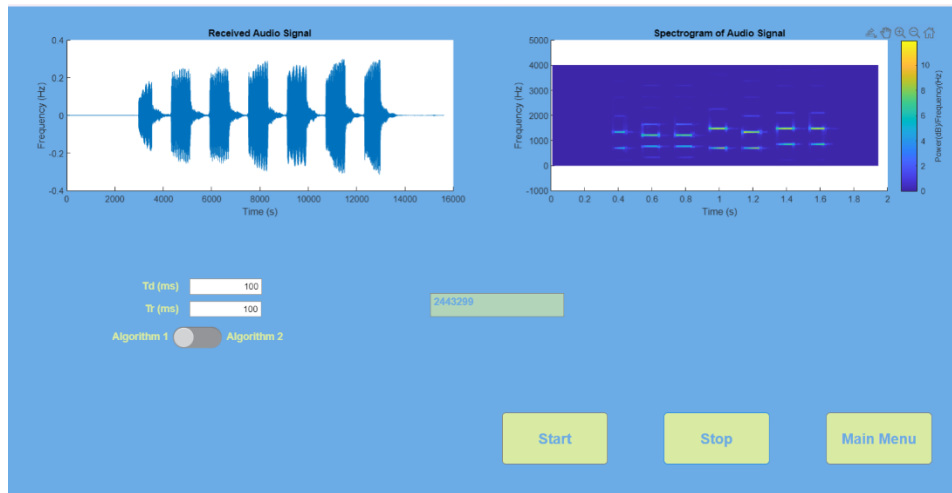


Figure-15: Decoding Results for  $T_d$ ,  $T_r$  are 100 msec

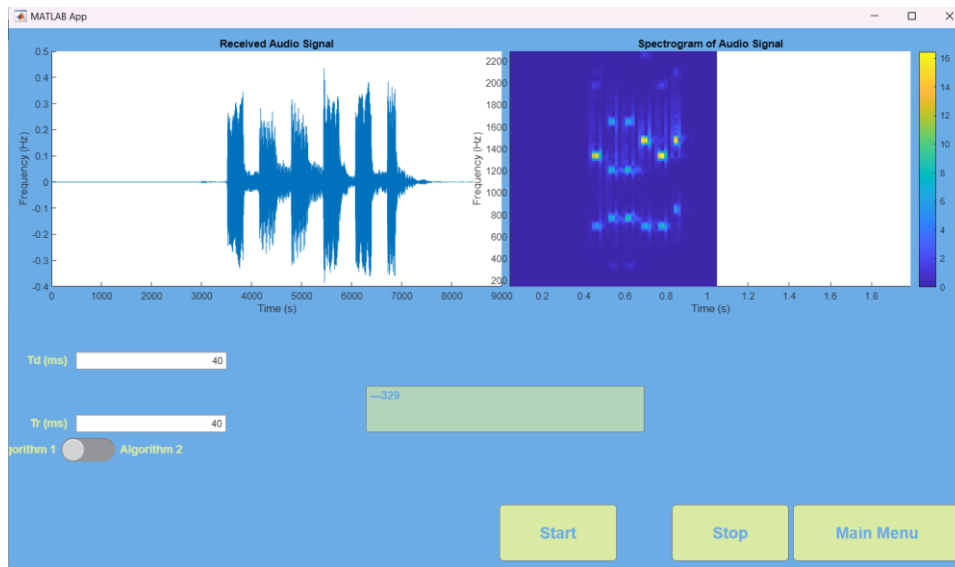


Figure-16: Decoding Results for  $T_d$ ,  $T_r$  are 40 msec

## CONCLUSION

To sum up, this study has effectively explored the complexities of Dual-Tone Multi-Frequency (DTMF) signaling. Accomplished with an intuitive Graphical User Interface (GUI), the MATLAB-based system has proven to be able to precisely generate, send, receive, and decode audio DTMF signals. We have learned a great deal about the capabilities and difficulties of DTMF signaling systems through the investigation of encoding techniques, the creation of an extensive graphical user interface (GUI), and the use of decoding algorithms. Through the implementation of diverse parameter configurations, such as signaling and resting lengths, this research has facilitated an in-depth comprehension of the system's functionality in numerous scenarios. As this project comes to an end, it is clear that the information and abilities gained here advance the field of digital signal processing and telecommunications technologies, demonstrating the continued relevance and significance of DTMF signaling in contemporary communication systems.

## Appendix

```
classdef app < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        TextArea_2              matlab.ui.control.TextArea
        TextArea                matlab.ui.control.TextArea
        Switch                  matlab.ui.control.Switch
        StartButton             matlab.ui.control.Button
        StopButton              matlab.ui.control.Button
        TrmsEditField_2         matlab.ui.control.NumericEditField
        TrmsEditField_2Label    matlab.ui.control.Label
        TdmsEditField_2         matlab.ui.control.NumericEditField
        TdmsEditField_2Label    matlab.ui.control.Label
        PlayButton              matlab.ui.control.Button
        SaveButton              matlab.ui.control.Button
        MainMenuButton          matlab.ui.control.Button
        AmplitudeEditField      matlab.ui.control.NumericEditField
        AmplitudeEditFieldLabel matlab.ui.control.Label
        TrmsEditField           matlab.ui.control.NumericEditField
        TrmsEditFieldLabel      matlab.ui.control.Label
        TdmsEditField           matlab.ui.control.NumericEditField
        TdmsEditFieldLabel      matlab.ui.control.Label
        ResetButton             matlab.ui.control.Button
        Button_5                matlab.ui.control.Button
        Button_0                matlab.ui.control.Button
        Button_9                matlab.ui.control.Button
        Button_8                matlab.ui.control.Button
        Button_7                matlab.ui.control.Button
        Button_6                matlab.ui.control.Button
        Button_4                matlab.ui.control.Button
        Button_3                matlab.ui.control.Button
        Button_2                matlab.ui.control.Button
        Button_1                matlab.ui.control.Button
        ReceiverPanelButton     matlab.ui.control.Button
        TransmitterPanelButton  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)
        PhoneNumber % Phone number to be displayed
        Recorder % To store audio data
        Td % Duration
        Tr % Rest
        Amplitude % Amplitude
        NumberArray % Array of phone digits
        InputLength % Length of input number
        m % DTMF encoded sequence
        LowFrequencyMatrix % Matrix of low frequencies
        HighFrequencyMatrix % Matrix of high frequencies
        Digits % Decoded sequence
        Td2
    end
end
```



```

        Tr2 % Description
    end

% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: PlayButton
    function PlayButtonPushed(app, event)
        Fs = 8000;
        sound(app.m,Fs);
    end

    % Button pushed function: TransmitterPanelButton
    function TransmitterPanelButtonPushed(app, event)
        app.PlayButton.Visible = "on";
        app.SaveButton.Visible = "on";
        app.MainMenuButton.Visible = "on";
        app.AmplitudeEditField.Visible = "on";
        app.TrmsEditField.Visible = "on";
        app.TdmsEditField.Visible = "on";
        app.ResetButton.Visible = "on";
        app.TextArea.Visible = "on";
        app.Button_1.Visible = "on";
        app.Button_8.Visible = "on";
        app.Button_9.Visible = "on";
        app.Button_0.Visible = "on";
        app.Button_2.Visible = "on";
        app.Button_3.Visible = "on";
        app.Button_4.Visible = "on";
        app.Button_5.Visible = "on";
        app.Button_6.Visible = "on";
        app.Button_7.Visible = "on";
        app.ReceiverPanelButton.Visible = "off";
        app.TransmitterPanelButton.Visible = "off";
        app.UIAxes_2.Visible = "on";
        app.UIAxes.Visible = "on";
        app.AmplitudeEditFieldLabel.Visible = "on";
        app.TrmsEditFieldLabel.Visible = "on";
        app.TdmsEditFieldLabel.Visible = "on";
        app.PhoneNumber = '';

        app.LowFrequencyMatrix = [697 770 852 941];
        app.HighFrequencyMatrix = [1209 1336 1477];
    end

    % Button pushed function: MainMenuButton
    function MainMenuButtonPushed(app, event)
        app.PlayButton.Visible = "off";
        app.SaveButton.Visible = "off";
        app.MainMenuButton.Visible = "off";
        app.AmplitudeEditField.Visible = "off";
        app.TrmsEditField.Visible = "off";
        app.TdmsEditField.Visible = "off";
        app.ResetButton.Visible = "off";
        app.TextArea.Visible = "off";
        app.Button_1.Visible = "off";
    end

```

```

app.Button_8.Visible = "off";
app.Button_9.Visible = "off";
app.Button_0.Visible = "off";
app.Button_2.Visible = "off";
app.Button_3.Visible = "off";
app.Button_4.Visible = "off";
app.Button_5.Visible = "off";
app.Button_6.Visible = "off";
app.Button_7.Visible = "off";
app.ReceiverPanelButton.Visible = "on";
app.TransmitterPanelButton.Visible = "on";
app.UIAxes_2.Visible = "off";
app.UIAxes.Visible = "off";
app.AmplitudeEditFieldLabel.Visible = "off";
app.TrmsEditFieldLabel.Visible = "off";
app.TdmsEditFieldLabel.Visible = "off";
app.TextArea_2.Visible = "off";
app.Switch.Visible = "off";
app.StartButton.Visible = "off";
app.StopButton.Visible = "off";
app.TrmsEditField_2.Visible = "off";
app.TrmsEditField_2Label.Visible = "off";
app.TdmsEditField_2.Visible = "off";
app.TdmsEditField_2Label.Visible = "off";
app.UIAxes_3.Visible = "off";
app.UIAxes_4.Visible = "off";
cla(app.UIAxes);
cla(app.UIAxes_2);
cla(app.UIAxes_3);
cla(app.UIAxes_4);
colorbarHandle = findobj(app.UIAxes_2.Parent, 'Type', 'ColorBar');
delete(colorbarHandle);

```

end

% Button pushed function: ReceiverPanelButton

```

function ReceiverPanelButtonPushed(app, event)
    app.TextArea_2.Visible = "on";
    app.Switch.Visible = "on";
    app.StartButton.Visible = "on";
    app.StopButton.Visible = "on";
    app.TrmsEditField_2.Visible = "on";
    app.TrmsEditField_2Label.Visible = "on";
    app.TdmsEditField_2.Visible = "on";
    app.TdmsEditField_2Label.Visible = "on";
    app.MainMenuButton.Visible = "on";
    app.ReceiverPanelButton.Visible = "off";
    app.TransmitterPanelButton.Visible = "off";
    app.UIAxes_3.Visible = "on";
    app.UIAxes_4.Visible = "on";

    app.LowFrequencyMatrix = [697 770 852 941];
    app.HighFrequencyMatrix = [1209 1336 1477];

```

end

% Button pushed function: Button\_1

```

function Button_1Pushed(app, event)
    app.PhoneNumber = strcat(app.PhoneNumber, '1');

```

```

        app.TextArea.Value = app.PhoneNumber;

end

% Callback function
function EditFieldValueChanged(app, event)

end

% Value changed function: TextArea
function TextAreaValueChanged(app, event)

end

% Button pushed function: Button_2
function Button_2Pushed(app, event)
    app.PhoneNumber = strcat(app.PhoneNumber, '2');
    app.TextArea.Value = app.PhoneNumber;
end

% Button pushed function: Button_3
function Button_3Pushed(app, event)
    app.PhoneNumber = strcat(app.PhoneNumber, '3');
    app.TextArea.Value = app.PhoneNumber;
end

% Button pushed function: Button_4
function Button_4Pushed(app, event)
    app.PhoneNumber = strcat(app.PhoneNumber, '4');
    app.TextArea.Value = app.PhoneNumber;
end

% Button pushed function: Button_5
function Button_5Pushed(app, event)
    app.PhoneNumber = strcat(app.PhoneNumber, '5');
    app.TextArea.Value = app.PhoneNumber;
end

% Button pushed function: Button_6
function Button_6Pushed(app, event)
    app.PhoneNumber = strcat(app.PhoneNumber, '6');
    app.TextArea.Value = app.PhoneNumber;
end

% Button pushed function: Button_7
function Button_7Pushed(app, event)
    app.PhoneNumber = strcat(app.PhoneNumber, '7');
    app.TextArea.Value = app.PhoneNumber;
end

% Button pushed function: Button_8
function Button_8Pushed(app, event)
    app.PhoneNumber = strcat(app.PhoneNumber, '8');
    app.TextArea.Value = app.PhoneNumber;
end

% Button pushed function: Button_9
function Button_9Pushed(app, event)
    app.PhoneNumber = strcat(app.PhoneNumber, '9');

```

```

        app.TextArea.Value = app.PhoneNumber;
    end

% Button pushed function: Button_0
function Button_0Pushed(app, event)
    app.PhoneNumber = strcat(app.PhoneNumber, '0');
    app.TextArea.Value = app.PhoneNumber;
end

% Button pushed function: ResetButton
function ResetButtonPushed(app, event)
    app.PhoneNumber = '';
    app.TextArea.Value = app.PhoneNumber;
end

% Button pushed function: StartButton
function StartButtonPushed(app, event)
    app.Recorder = audiorecorder(8000,24,1);
    record(app.Recorder);
    app.TextArea_2.Value = '';
end

% Button pushed function: StopButton
function StopButtonPushed(app, event)
    stop(app.Recorder);
    audioData = getaudiodata(app.Recorder);
    plot(app.UIAxes_3,audioData);
    [sp,f,ts]=spectrogram(audioData, hamming(256), 128,512,8000);
    axes(app.UIAxes_4);
    imagesc(app.UIAxes_4,ts,f,abs(sp));
    colorbar(app.UIAxes_4);
    set(app.UIAxes_4, 'YDir', 'normal');
    Td = app.Td2;
    Tr = app.Tr2;
    Ts = Td+Tr;

    points = cell(0,2);
    for i=1:length(ts)-1
        for j=1:length(f)-1
            power = abs(sp(j,i));
            if power>5
                point=[ts(i) f(j)];
                points{end+1}=point;
            end
        end
    end
    low_frequencies=[];
    high_frequencies=[];
    highs=0;
    num_highs=0;
    lows=0;
    num_lows=0;
    start = points{1}(1);
    for i=2:length(points)
        t_value = points{i}(1);
        chunk_num = ceil((t_value-start)/Ts);

        if points{i-1}(1)+Tr/2<points{i}(1)
            high_frequencies(chunk_num) = (highs/num_highs);

```

```

        highs = 0;
        num_highs = 0;

        low_frequencies(chunk_num) = (lows/num_lows);
        lows = 0;
        num_lows = 0;
elseif (i==length(points))
    high_frequencies(chunk_num) = (highs/num_highs);
    highs = 0;
    num_highs = 0;

    low_frequencies(chunk_num) = (lows/num_lows);
    lows = 0;
    num_lows = 0;

else
    if points{i-1}(2) >1071
        highs = highs+points{i-1}(2);
        num_highs = num_highs+1;
    else
        lows = lows+points{i-1}(2);
        num_lows = num_lows+1;
    end
end
end
disp('highs');
disp(high_frequencies);
disp('lows');
disp(low_frequencies);
app.Digits = '';
for i=1:length(low_frequencies)
    lowfreq = low_frequencies(i);
    highfreq = high_frequencies(i);

    if lowfreq>687 && lowfreq<707 && highfreq<1219 && highfreq>1199
        app.Digits = strcat(app.Digits,'1');
    elseif lowfreq>760 && lowfreq<780 && highfreq<1219 &&
highfreq>1199
        app.Digits = strcat(app.Digits,'4');
    elseif lowfreq>842 && lowfreq<862 && highfreq<1219 &&
highfreq>1199
        app.Digits = strcat(app.Digits,'7');
    elseif lowfreq>687 && lowfreq<707 && highfreq<1346 &&
highfreq>1326
        app.Digits = strcat(app.Digits,'2');
    elseif lowfreq>760 && lowfreq<780 && highfreq<1346 &&
highfreq>1326
        app.Digits = strcat(app.Digits,'5');
    elseif lowfreq>842 && lowfreq<862 && highfreq<1346 &&
highfreq>1326
        app.Digits = strcat(app.Digits,'8');
    elseif lowfreq>687 && lowfreq<707 && highfreq<1487 &&
highfreq>1467
        app.Digits = strcat(app.Digits,'3');
    elseif lowfreq>760 && lowfreq<780 && highfreq<1487 &&
highfreq>1467
        app.Digits = strcat(app.Digits,'6');
    elseif lowfreq>842 && lowfreq<862 && highfreq<1487 &&
highfreq>1467

```

```

        app.Digits = strcat(app.Digits, '9');
    elseif lowfreq>931 && lowfreq<951 && highfreq<1346 &&
highfreq>1326
        app.Digits = strcat(app.Digits, '0');
    else
        app.Digits = strcat(app.Digits, '-');
    end

end

app.TextArea_2.Value = app.Digits;
xlabel(app.UIAxes_3, 'Time (s)');
ylabel(app.UIAxes_3, 'Frequency (Hz)');
xlabel(app.UIAxes_4, 'Time (s)');
ylabel(app.UIAxes_4, 'Frequency (Hz)');
ylim(app.UIAxes_4, [500 1600]);
colorbarLabel = 'Power(dB)/Frequency(Hz)';
c = colorbar(app.UIAxes_4);
c.Label.String = colorbarLabel;
title(app.UIAxes_3, 'Received Audio Signal');
title(app.UIAxes_4, 'Spectrogram of Audio Signal');

end

% Value changed function: TdmsEditField
function TdmsEditFieldValueChanged(app, event)
    app.Td = app.TdmsEditField.Value/1000;

end

% Value changed function: TrmsEditField
function TrmsEditFieldValueChanged(app, event)
    app.Tr = app.TrmsEditField.Value/1000;

end

% Value changed function: AmplitudeEditField
function AmplitudeEditFieldValueChanged(app, event)
    app.Amplitude = app.AmplitudeEditField.Value;

end

```

```

% Button pushed function: SaveButton
function SaveButtonPushed(app, event)
    app.NumberArray = str2num(app.PhoneNumber.');
    app.InputLength = strlen(app.PhoneNumber);
    app.m = 0;
    app.LowFrequencyMatrix = [697 770 852 941];
    app.HighFrequencyMatrix = [1209 1336 1477];
    max = app.InputLength*(app.Td + app.Tr);

    Fs = 8000;
    for i=1:app.InputLength
        k = app.NumberArray(i);
        if mod(k,3)~=0 && k~=0
            fL = app.LowFrequencyMatrix(floor(k/3.1)+1);
            fH = app.HighFrequencyMatrix(mod(k,3));
        elseif k==0
            fL = app.LowFrequencyMatrix(4);
            fH = app.HighFrequencyMatrix(2);
        else
            fL = app.LowFrequencyMatrix(floor(k/3.1)+1);
            fH = app.HighFrequencyMatrix(3);
        end
        t = 0:(1/8000):max;
        phase_rad_low = fL*(i-1)*(app.Td+app.Tr)*pi;
        phase_rad_high = fH*(i-1)*(app.Td+app.Tr)*pi;
        angle_low = 2*pi*fL*t;
        angle_high = 2*pi*fH*t;
        sin_low = app.Amplitude* sin(angle_low-phase_rad_low);
        sin_high = app.Amplitude * sin(angle_high-phase_rad_high);
        sin_sum = sin_low + sin_high;
        shifted_unit_step = (t>=(i-1)*(app.Td+app.Tr))&(t<=((i-
1)*(app.Td+app.Tr)+app.Td));
        s = sin_sum .* shifted_unit_step;
        app.m = app.m + s;
    end
    axes(app.UIAxes_2);
    if app.Td <100
        [sp,f,ts]=spectrogram(app.m, hamming(128), 64,512,Fs);
    else
        [sp,f,ts]=spectrogram(app.m, hamming(256), 128,512,Fs);
    end

    imagesc(app.UIAxes_2,ts,f,abs(sp));
    colorbar(app.UIAxes_2);
    set(app.UIAxes_2, 'YDir', 'normal');
    plot(app.UIAxes, t,app.m);
    ylim(app.UIAxes_2, [500 1600]);
    xlim(app.UIAxes_2, [0 max]);
    xlim(app.UIAxes, [0 max])
    xlabel(app.UIAxes, 'Time (s)');
    ylabel(app.UIAxes, 'Frequency (Hz)');
    xlabel(app.UIAxes_2, 'Time (s)');
    ylabel(app.UIAxes_2, 'Frequency (Hz)');
    colorbarLabel = 'Power(dB)/Frequency(Hz)';
    c = colorbar(app.UIAxes_2);

```

```

        c.Label.String = colorbarLabel;
        title(app.UIAxes, 'DTMF of Input Signal');
        title(app.UIAxes_2, 'Spectrogram of Input Signal');
        file_name = 'DTMF_encoded.wav';
        audiowrite(file_name, app.m, Fs)

    end

    % Value changed function: TextArea_2
    function TextArea_2ValueChanged(app, event)

    end

    % Value changed function: TrmsEditField_2
    function TrmsEditField_2ValueChanged(app, event)
        app.Tr2 = app.TrmsEditField_2.Value/1000;
    end

    % Value changed function: TdmsEditField_2
    function TdmsEditField_2ValueChanged(app, event)
        app.Td2 = app.TdmsEditField_2.Value/1000;
    end

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 = [0.4196 0.6706 0.902];
        app UIFigure.Position = [100 100 1318 678];
        app UIFigure.Name = 'MATLAB App';

        % Create UIAxes
        app.UIAxes = uiaxes(app UIFigure);
        title(app.UIAxes, 'Title')
        xlabel(app.UIAxes, 'X')
        ylabel(app.UIAxes, 'Y')
        zlabel(app.UIAxes, 'Z')
        app.UIAxes.Visible = 'off';
        app.UIAxes.Position = [40 389 593 262];

        % Create UIAxes_2
        app.UIAxes_2 = uiaxes(app UIFigure);
        title(app.UIAxes_2, 'Title')
        xlabel(app.UIAxes_2, 'X')
        ylabel(app.UIAxes_2, 'Y')
        zlabel(app.UIAxes_2, 'Z')
        app.UIAxes_2.Visible = 'off';
        app.UIAxes_2.Position = [684 389 593 262];
    end
end

```



```

% Create UIAxes_3
app.UIAxes_3 = uiaxes(app.UIFigure);
title(app.UIAxes_3, 'Title')
xlabel(app.UIAxes_3, 'X')
ylabel(app.UIAxes_3, 'Y')
zlabel(app.UIAxes_3, 'Z')
app.UIAxes_3.Visible = 'off';
app.UIAxes_3.Position = [40 389 593 262];

% Create UIAxes_4
app.UIAxes_4 = uiaxes(app.UIFigure);
title(app.UIAxes_4, 'Title')
xlabel(app.UIAxes_4, 'X')
ylabel(app.UIAxes_4, 'Y')
zlabel(app.UIAxes_4, 'Z')
app.UIAxes_4.Visible = 'off';
app.UIAxes_4.Position = [697 389 593 262];

% Create TransmitterPanelButton
app.TransmitterPanelButton = uibutton(app.UIFigure, 'push');
app.TransmitterPanelButton.ButtonPushedFcn = createCallbackFcn(app,
@TransmitterPanelButtonPushed, true);
app.TransmitterPanelButton.BackgroundColor = [0.851 0.9216 0.6392];
app.TransmitterPanelButton.FontSize = 30;
app.TransmitterPanelButton.FontWeight = 'bold';
app.TransmitterPanelButton.FontColor = [0.4196 0.6706 0.902];
app.TransmitterPanelButton.Position = [374 305 270 129];
app.TransmitterPanelButton.Text = 'Transmitter Panel';

% Create ReceiverPanelButton
app.ReceiverPanelButton = uibutton(app.UIFigure, 'push');
app.ReceiverPanelButton.ButtonPushedFcn = createCallbackFcn(app,
@ReceiverPanelButtonPushed, true);
app.ReceiverPanelButton.BackgroundColor = [0.851 0.9216 0.6392];
app.ReceiverPanelButton.FontSize = 30;
app.ReceiverPanelButton.FontWeight = 'bold';
app.ReceiverPanelButton.FontColor = [0.4196 0.6706 0.902];
app.ReceiverPanelButton.Position = [697 305 270 129];
app.ReceiverPanelButton.Text = 'Receiver Panel';

% Create Button_1
app.Button_1 = uibutton(app.UIFigure, 'push');
app.Button_1.ButtonPushedFcn = createCallbackFcn(app, @Button_1Pushed,
true);
app.Button_1.BackgroundColor = [0.851 0.9216 0.6392];
app.Button_1.FontSize = 18;
app.Button_1.FontWeight = 'bold';
app.Button_1.FontColor = [0.4196 0.6706 0.902];
app.Button_1.Visible = 'off';
app.Button_1.Position = [175 229 55 49];
app.Button_1.Text = '1';

% Create Button_2
app.Button_2 = uibutton(app.UIFigure, 'push');
app.Button_2.ButtonPushedFcn = createCallbackFcn(app, @Button_2Pushed,
true);
app.Button_2.BackgroundColor = [0.851 0.9216 0.6392];
app.Button_2.FontSize = 18;

```

```

app.Button_2.FontWeight = 'bold';
app.Button_2.FontColor = [0.4196 0.6706 0.902];
app.Button_2.Visible = 'off';
app.Button_2.Position = [240 229 55 49];
app.Button_2.Text = '2';

% Create Button_3
app.Button_3 = uibutton(app.UIFigure, 'push');
app.Button_3.ButtonPushedFcn = createCallbackFcn(app, @Button_3Pushed,
true);

app.Button_3.BackgroundColor = [0.851 0.9216 0.6392];
app.Button_3.FontSize = 18;
app.Button_3.FontWeight = 'bold';
app.Button_3.FontColor = [0.4196 0.6706 0.902];
app.Button_3.Visible = 'off';
app.Button_3.Position = [305 229 55 49];
app.Button_3.Text = '3';

% Create Button_4
app.Button_4 = uibutton(app.UIFigure, 'push');
app.Button_4.ButtonPushedFcn = createCallbackFcn(app, @Button_4Pushed,
true);

app.Button_4.BackgroundColor = [0.851 0.9216 0.6392];
app.Button_4.FontSize = 18;
app.Button_4.FontWeight = 'bold';
app.Button_4.FontColor = [0.4196 0.6706 0.902];
app.Button_4.Visible = 'off';
app.Button_4.Position = [175 171 55 49];
app.Button_4.Text = '4';

% Create Button_6
app.Button_6 = uibutton(app.UIFigure, 'push');
app.Button_6.ButtonPushedFcn = createCallbackFcn(app, @Button_6Pushed,
true);

app.Button_6.BackgroundColor = [0.851 0.9216 0.6392];
app.Button_6.FontSize = 18;
app.Button_6.FontWeight = 'bold';
app.Button_6.FontColor = [0.4196 0.6706 0.902];
app.Button_6.Visible = 'off';
app.Button_6.Position = [305 171 55 49];
app.Button_6.Text = '6';

% Create Button_7
app.Button_7 = uibutton(app.UIFigure, 'push');
app.Button_7.ButtonPushedFcn = createCallbackFcn(app, @Button_7Pushed,
true);

app.Button_7.BackgroundColor = [0.851 0.9216 0.6392];
app.Button_7.FontSize = 18;
app.Button_7.FontWeight = 'bold';
app.Button_7.FontColor = [0.4196 0.6706 0.902];
app.Button_7.Visible = 'off';
app.Button_7.Position = [175 109 55 49];
app.Button_7.Text = '7';

% Create Button_8
app.Button_8 = uibutton(app.UIFigure, 'push');
app.Button_8.ButtonPushedFcn = createCallbackFcn(app, @Button_8Pushed,
true);

app.Button_8.BackgroundColor = [0.851 0.9216 0.6392];

```

```

app.Button_8.FontSize = 18;
app.Button_8.FontWeight = 'bold';
app.Button_8.FontColor = [0.4196 0.6706 0.902];
app.Button_8.Visible = 'off';
app.Button_8.Position = [240 109 55 49];
app.Button_8.Text = '8';

% Create Button_9
app.Button_9 = uibutton(app.UIFigure, 'push');
app.Button_9.ButtonPushedFcn = createCallbackFcn(app, @Button_9Pushed,
true);

app.Button_9.BackgroundColor = [0.851 0.9216 0.6392];
app.Button_9.FontSize = 18;
app.Button_9.FontWeight = 'bold';
app.Button_9.FontColor = [0.4196 0.6706 0.902];
app.Button_9.Visible = 'off';
app.Button_9.Position = [305 109 55 49];
app.Button_9.Text = '9';

% Create Button_0
app.Button_0 = uibutton(app.UIFigure, 'push');
app.Button_0.ButtonPushedFcn = createCallbackFcn(app, @Button_0Pushed,
true);

app.Button_0.BackgroundColor = [0.851 0.9216 0.6392];
app.Button_0.FontSize = 18;
app.Button_0.FontWeight = 'bold';
app.Button_0.FontColor = [0.4196 0.6706 0.902];
app.Button_0.Visible = 'off';
app.Button_0.Position = [240 48 55 49];
app.Button_0.Text = '0';

% Create Button_5
app.Button_5 = uibutton(app.UIFigure, 'push');
app.Button_5.ButtonPushedFcn = createCallbackFcn(app, @Button_5Pushed,
true);

app.Button_5.BackgroundColor = [0.851 0.9216 0.6392];
app.Button_5.FontSize = 18;
app.Button_5.FontWeight = 'bold';
app.Button_5.FontColor = [0.4196 0.6706 0.902];
app.Button_5.Visible = 'off';
app.Button_5.Position = [240 171 55 49];
app.Button_5.Text = '5';

% Create ResetButton
app.ResetButton = uibutton(app.UIFigure, 'push');
app.ResetButton.ButtonPushedFcn = createCallbackFcn(app,
@ResetButtonPushed, true);

app.ResetButton.BackgroundColor = [0.851 0.9216 0.6392];
app.ResetButton.FontSize = 18;
app.ResetButton.FontWeight = 'bold';
app.ResetButton.FontColor = [0.4196 0.6706 0.902];
app.ResetButton.Visible = 'off';
app.ResetButton.Position = [301 48 64 49];
app.ResetButton.Text = 'Reset';

% Create TdmsEditFieldLabel
app.TdmsEditFieldLabel = uilabel(app.UIFigure);
app.TdmsEditFieldLabel.HorizontalAlignment = 'right';
app.TdmsEditFieldLabel.FontSize = 14;

```

```

app.TdmsEditFieldLabel.FontWeight = 'bold';
app.TdmsEditFieldLabel.FontColor = [0.851 0.9216 0.6392];
app.TdmsEditFieldLabel.Visible = 'off';
app.TdmsEditFieldLabel.Position = [415 219 56 22];
app.TdmsEditFieldLabel.Text = 'Td (ms)';

% Create TdmsEditField
app.TdmsEditField = uieditfield(app.UIFigure, 'numeric');
app.TdmsEditField.ValueChangedFcn = createCallbackFcn(app,
@TdmsEditFieldValueChanged, true);
app.TdmsEditField.Visible = 'off';
app.TdmsEditField.Position = [486 219 100 22];

% Create TrmsEditFieldLabel
app.TrmsEditFieldLabel = uilabel(app.UIFigure);
app.TrmsEditFieldLabel.HorizontalAlignment = 'right';
app.TrmsEditFieldLabel.FontSize = 14;
app.TrmsEditFieldLabel.FontWeight = 'bold';
app.TrmsEditFieldLabel.FontColor = [0.851 0.9216 0.6392];
app.TrmsEditFieldLabel.Visible = 'off';
app.TrmsEditFieldLabel.Position = [419 188 52 22];
app.TrmsEditFieldLabel.Text = 'Tr (ms)';

% Create TrmsEditField
app.TrmsEditField = uieditfield(app.UIFigure, 'numeric');
app.TrmsEditField.ValueChangedFcn = createCallbackFcn(app,
@TrmsEditFieldValueChanged, true);
app.TrmsEditField.Visible = 'off';
app.TrmsEditField.Position = [486 188 99 22];

% Create AmplitudeEditFieldLabel
app.AmplitudeEditFieldLabel = uilabel(app.UIFigure);
app.AmplitudeEditFieldLabel.HorizontalAlignment = 'right';
app.AmplitudeEditFieldLabel.FontSize = 14;
app.AmplitudeEditFieldLabel.FontWeight = 'bold';
app.AmplitudeEditFieldLabel.FontColor = [0.851 0.9216 0.6392];
app.AmplitudeEditFieldLabel.Visible = 'off';
app.AmplitudeEditFieldLabel.Position = [397 153 73 22];
app.AmplitudeEditFieldLabel.Text = 'Amplitude';

% Create AmplitudeEditField
app.AmplitudeEditField = uieditfield(app.UIFigure, 'numeric');
app.AmplitudeEditField.ValueChangedFcn = createCallbackFcn(app,
@AmplitudeEditFieldValueChanged, true);
app.AmplitudeEditField.Visible = 'off';
app.AmplitudeEditField.Position = [486 153 99 22];

% Create MainMenuButton
app.MainMenuButton = uibutton(app.UIFigure, 'push');
app.MainMenuButton.ButtonPushedFcn = createCallbackFcn(app,
@MainMenuButtonPushed, true);
app.MainMenuButton.BackgroundColor = [0.851 0.9216 0.6392];
app.MainMenuButton.FontSize = 20;
app.MainMenuButton.FontWeight = 'bold';
app.MainMenuButton.FontColor = [0.4196 0.6706 0.902];
app.MainMenuButton.Visible = 'off';
app.MainMenuButton.Position = [1131 49 146 71];
app.MainMenuButton.Text = 'Main Menu';

```

```

    % Create SaveButton
    app.SaveButton = uibutton(app.UIFigure, 'push');
    app.SaveButton.ButtonPushedFcn = createCallbackFcn(app,
@SaveButtonPushed, true);
    app.SaveButton.BackgroundColor = [0.851 0.9216 0.6392];
    app.SaveButton.FontSize = 20;
    app.SaveButton.FontWeight = 'bold';
    app.SaveButton.FontColor = [0.4196 0.6706 0.902];
    app.SaveButton.Visible = 'off';
    app.SaveButton.Position = [681 49 146 71];
    app.SaveButton.Text = 'Save';

    % Create PlayButton
    app.PlayButton = uibutton(app.UIFigure, 'push');
    app.PlayButton.ButtonPushedFcn = createCallbackFcn(app,
@PlayButtonPushed, true);
    app.PlayButton.BackgroundColor = [0.851 0.9216 0.6392];
    app.PlayButton.FontSize = 20;
    app.PlayButton.FontWeight = 'bold';
    app.PlayButton.FontColor = [0.4196 0.6706 0.902];
    app.PlayButton.Visible = 'off';
    app.PlayButton.Position = [907 49 146 71];
    app.PlayButton.Text = 'Play';

    % Create TdmsEditField_2Label
    app.TdmsEditField_2Label = uilabel(app.UIFigure);
    app.TdmsEditField_2Label.HorizontalAlignment = 'right';
    app.TdmsEditField_2Label.FontSize = 14;
    app.TdmsEditField_2Label.FontWeight = 'bold';
    app.TdmsEditField_2Label.FontColor = [0.851 0.9216 0.6392];
    app.TdmsEditField_2Label.Visible = 'off';
    app.TdmsEditField_2Label.Position = [182 283 56 22];
    app.TdmsEditField_2Label.Text = 'Td (ms)';

    % Create TdmsEditField_2
    app.TdmsEditField_2 = uieditfield(app.UIFigure, 'numeric');
    app.TdmsEditField_2.ValueChangedFcn = createCallbackFcn(app,
@TdmsEditField_2ValueChanged, true);
    app.TdmsEditField_2.Visible = 'off';
    app.TdmsEditField_2.Position = [253 283 100 22];

    % Create TrmsEditField_2Label
    app.TrmsEditField_2Label = uilabel(app.UIFigure);
    app.TrmsEditField_2Label.HorizontalAlignment = 'right';
    app.TrmsEditField_2Label.FontSize = 14;
    app.TrmsEditField_2Label.FontWeight = 'bold';
    app.TrmsEditField_2Label.FontColor = [0.851 0.9216 0.6392];
    app.TrmsEditField_2Label.Visible = 'off';
    app.TrmsEditField_2Label.Position = [186 252 52 22];
    app.TrmsEditField_2Label.Text = 'Tr (ms)';

    % Create TrmsEditField_2
    app.TrmsEditField_2 = uieditfield(app.UIFigure, 'numeric');
    app.TrmsEditField_2.ValueChangedFcn = createCallbackFcn(app,
@TrmsEditField_2ValueChanged, true);
    app.TrmsEditField_2.Visible = 'off';
    app.TrmsEditField_2.Position = [253 252 99 22];

    % Create StopButton

```

```

        app.StopButton = uibutton(app.UIFigure, 'push');
        app.StopButton.ButtonPushedFcn = createCallbackFcn(app,
@StopButtonPushed, true);
        app.StopButton.BackgroundColor = [0.851 0.9216 0.6392];
        app.StopButton.FontSize = 20;
        app.StopButton.FontWeight = 'bold';
        app.StopButton.FontColor = [0.4196 0.6706 0.902];
        app.StopButton.Visible = 'off';
        app.StopButton.Position = [907 49 146 71];
        app.StopButton.Text = 'Stop';

% Create StartButton
        app.StartButton = uibutton(app.UIFigure, 'push');
        app.StartButton.ButtonPushedFcn = createCallbackFcn(app,
@StartButtonPushed, true);
        app.StartButton.BackgroundColor = [0.851 0.9216 0.6392];
        app.StartButton.FontSize = 20;
        app.StartButton.FontWeight = 'bold';
        app.StartButton.FontColor = [0.4196 0.6706 0.902];
        app.StartButton.Visible = 'off';
        app.StartButton.Position = [684 49 146 71];
        app.StartButton.Text = 'Start';

% Create Switch
        app.Switch = uiswitch(app.UIFigure, 'slider');
        app.Switch.Items = {'Algorithm 1', 'Algorithm 2'};
        app.Switch.Visible = 'off';
        app.Switch.FontSize = 14;
        app.Switch.FontWeight = 'bold';
        app.Switch.FontColor = [0.851 0.9216 0.6392];
        app.Switch.Position = [230 209 68 30];
        app.Switch.Value = 'Algorithm 1';

% Create TextArea
        app.TextArea = uitextarea(app.UIFigure);
        app.TextArea.ValueChangedFcn = createCallbackFcn(app,
@TextAreaValueChanged, true);
        app.TextArea.Editable = 'off';
        app.TextArea.FontSize = 18;
        app.TextArea.FontWeight = 'bold';
        app.TextArea.FontColor = [0.4196 0.6706 0.902];
        app.TextArea.BackgroundColor = [0.851 0.9216 0.6392];
        app.TextArea.Visible = 'off';
        app.TextArea.Position = [176 290 185 32];

% Create TextArea_2
        app.TextArea_2 = uitextarea(app.UIFigure);
        app.TextArea_2.ValueChangedFcn = createCallbackFcn(app,
@TextArea_2ValueChanged, true);
        app.TextArea_2.Editable = 'off';
        app.TextArea_2.FontSize = 14;
        app.TextArea_2.FontWeight = 'bold';
        app.TextArea_2.FontColor = [0.4196 0.6706 0.902];
        app.TextArea_2.BackgroundColor = [0.851 0.9216 0.6392];
        app.TextArea_2.Visible = 'off';
        app.TextArea_2.Position = [585 252 185 33];

% 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 = app

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        if nargin == 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
end

```