EE430 Project Part – 2

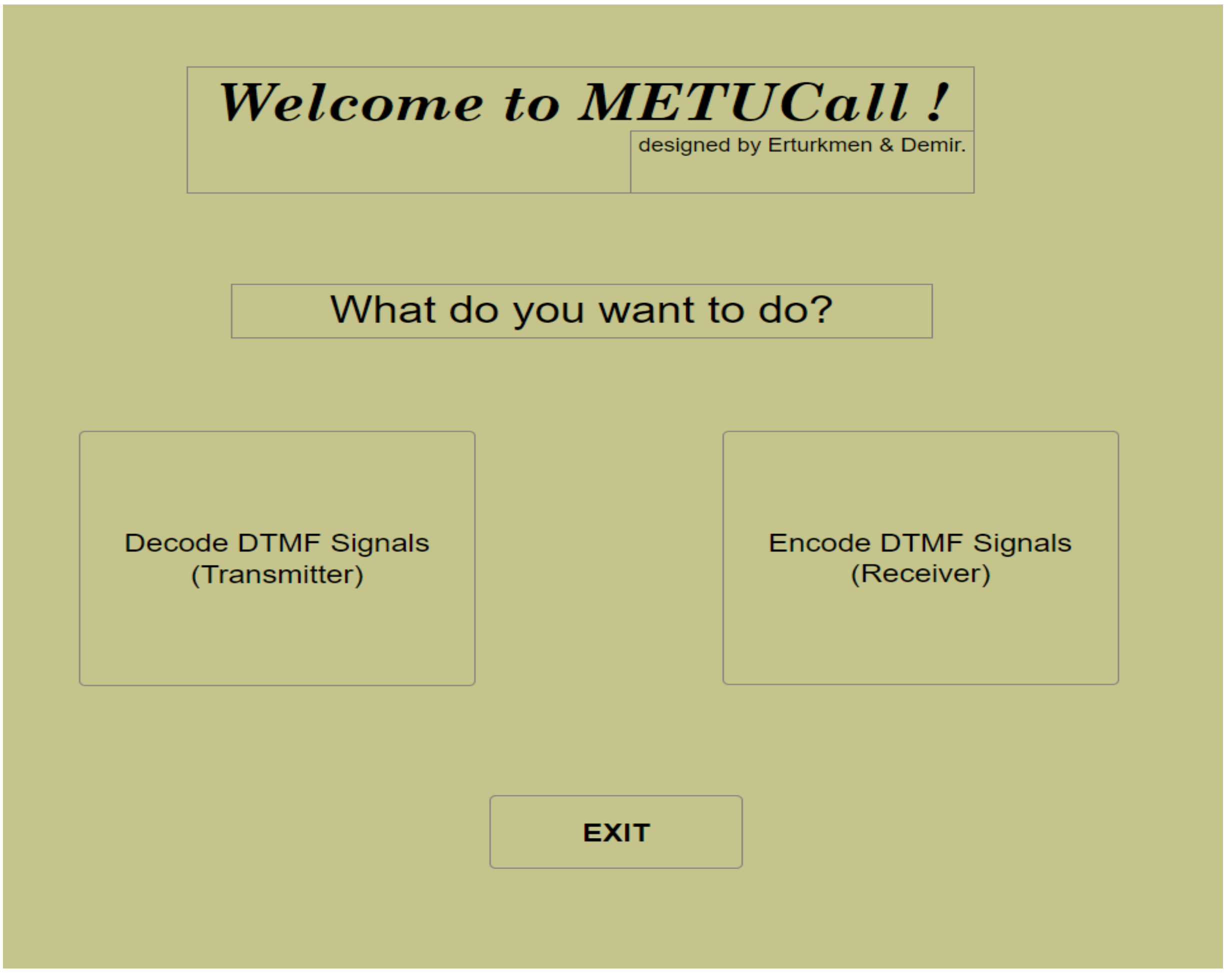
Eray DEMİR
2442853

Ahmet ERTÜRKMEN
2443000

Introduction

This report examines the role of Dual-Tone Multi-Frequency (DTMF) signaling in telecommunications and contrasts two main algorithms for encoding these signals: the Goertzel and Spectrogram-Based Algorithms. DTMF signaling is key for transmitting information over telephone lines, especially for dialing and automated systems. The Goertzel Algorithm is known for its precise and efficient detection of DTMF tones, while the Spectrogram-Based Algorithm provides a wider analysis of frequencies, useful for more extensive applications. The aim is to highlight the functions of DTMF signaling and compare these algorithms in terms of their effectiveness and suitability in different scenarios.

Main Menu



Welcome to METUCall !

designed by Erturkmen & Demir.

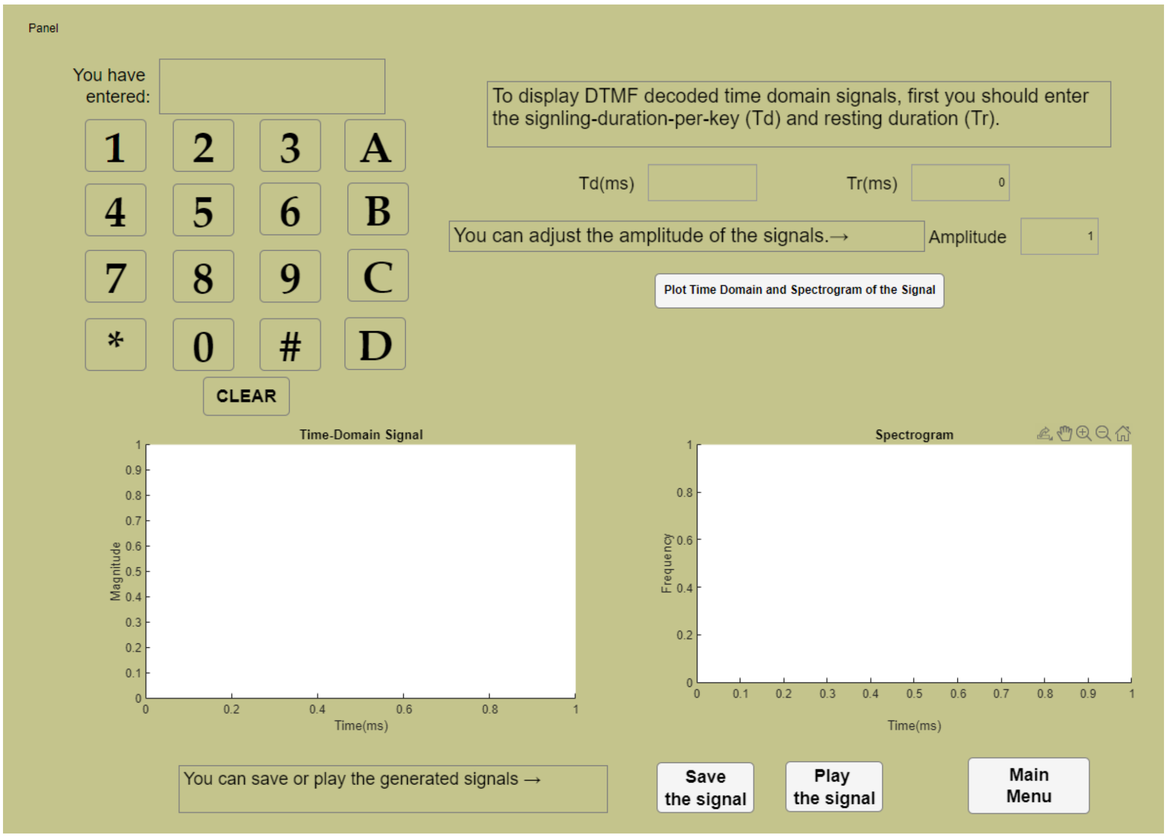
What do you want to do?

Decode DTMF Signals (Transmitter)
Encode DTMF Signals (Receiver)
EXIT

Figure 1 The screenshot of the Main Menu.

- There are three buttons in the main page.
    1. The “Decode DTMF Signals (Transmitter)” button functions as to open transmitter panel where user can create DTMF signals.
    2. The “Encode DTMF Signals (Receiver)” button functions as to open receiver panel where user can encode the DTMF signals.
    3. The “Exit” button is to exit the program.

Transmitter Panel



You have entered:

To display DTMF decoded time domain signals, first you should enter the signling-duration-per-key (Td) and resting duration (Tr).

Td(ms)

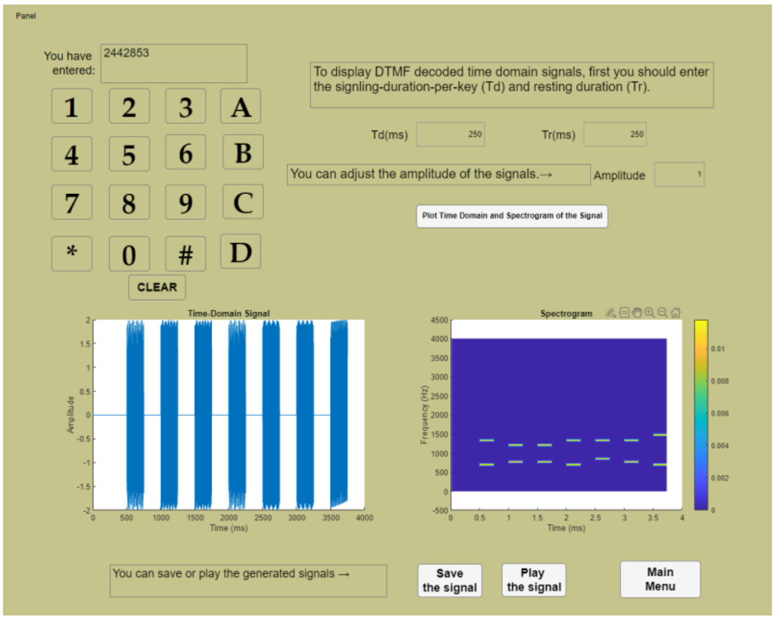
Tr(ms) 0

You can adjust the amplitude of the signals.→ Amplitude 1

You can save or play the generated signals →

Figure 2 The screenshot of the Transmitter Panel.

- There is a keypad in this panel which enables the user to enter a key. Also, the text area above the keypad shows the numbers entered.
- Clear button is to delete all number that have been entered.
- First user should enter the inputs to the areas Td(ms), Tr(ms) and Amplitude. Then, by pushing the “Plot Time Domain and Spectrogram of the Signal” button, Time-Domain, and Spectrogram of the DTMF signal will be plotted on the axes.
- By “Save the signal” button user can save the created input signal as a .wav file, additionally this button allows user to choose the path that the file will be saved in.
- By “Play the signal” button user can listen the created signal.
- By “Main Menu” button user can go back to main menu.


Panel

You have entered: 2442853

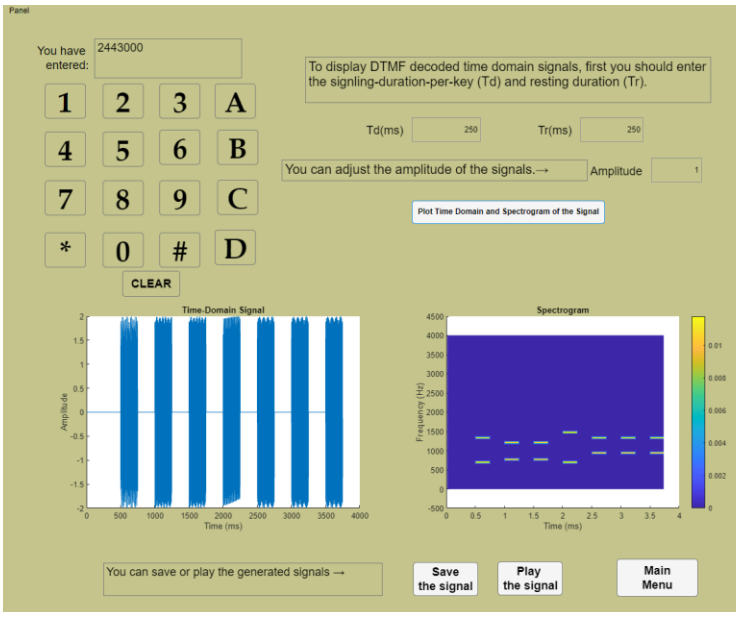
To display DTMF decoded time domain signals, first you should enter the signling-duration-per-key (Td) and resting duration (Tr).

Td(ms) 250 Tr(ms) 250

You can adjust the amplitude of the signals.→ Amplitude 1

You can save or play the generated signals →

Figure 3 Signaling [2,4,4,2,8,5,3] with DTMF (*Td, Tr*) = (250ms/250ms).


Panel

You have entered: 2443000

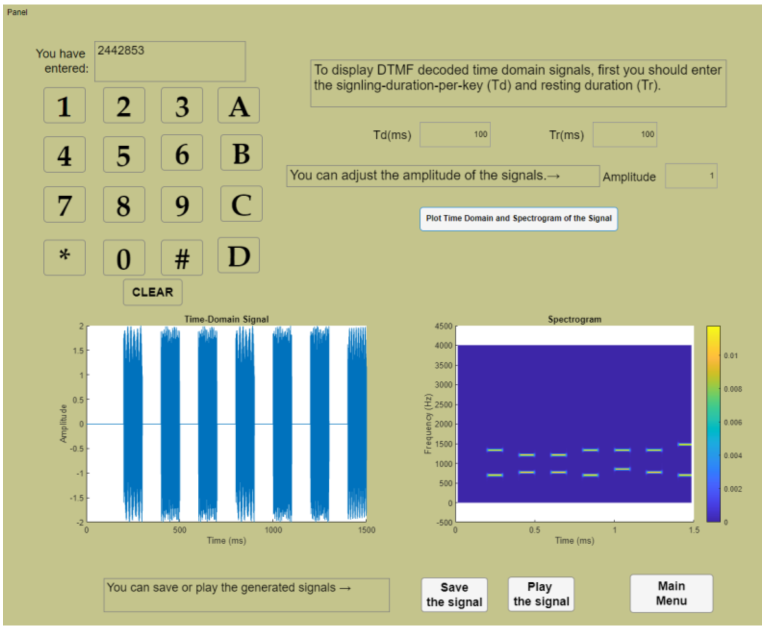
To display DTMF decoded time domain signals, first you should enter the signling-duration-per-key (Td) and resting duration (Tr).

Td(ms) 250 Tr(ms) 250

You can adjust the amplitude of the signals.→ Amplitude 1

You can save or play the generated signals →

Figure 4 Signaling [2,4,4,3,0,0,0] with DTMF (*Td, Tr*) = (250ms/250ms).


Panel

You have entered: 2442853

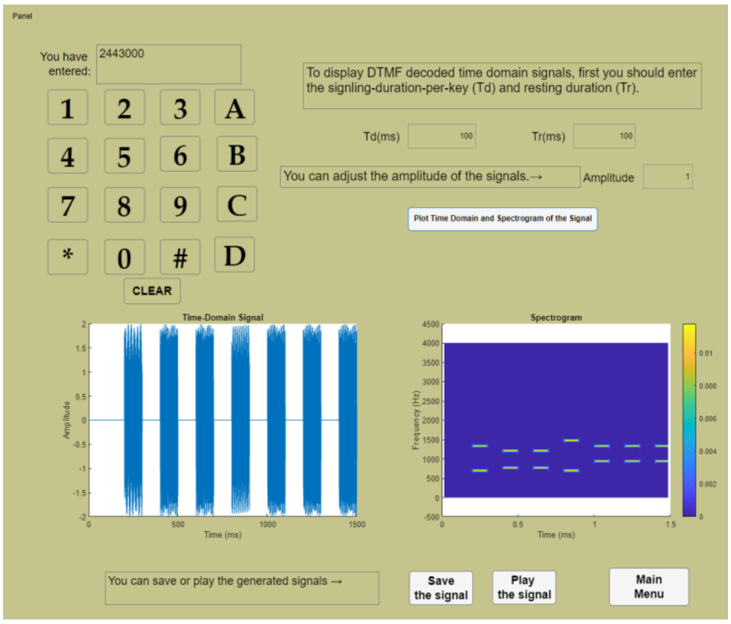
To display DTMF decoded time domain signals, first you should enter the signling-duration-per-key (Td) and resting duration (Tr).

Td(ms) 100 Tr(ms) 100

You can adjust the amplitude of the signals.→ Amplitude 1

You can save or play the generated signals →

Figure 5 Signaling [2,4,4,2,8,5,3] with DTMF (*Td, Tr*) = (100ms/100ms).


Panel

You have entered: 2443000

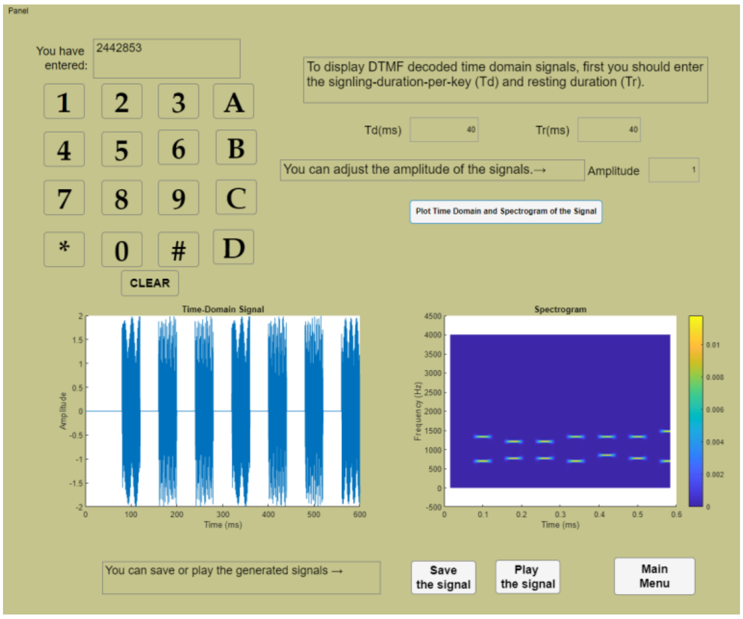
To display DTMF decoded time domain signals, first you should enter the signling-duration-per-key (Td) and resting duration (Tr).

Td(ms) 100 Tr(ms) 100

You can adjust the amplitude of the signals.→ Amplitude 1

You can save or play the generated signals →

Figure 6 Signaling [2,4,4,3,0,0,0] with DTMF (*Td, Tr*) = (100ms/100ms).



You have entered: 2442853

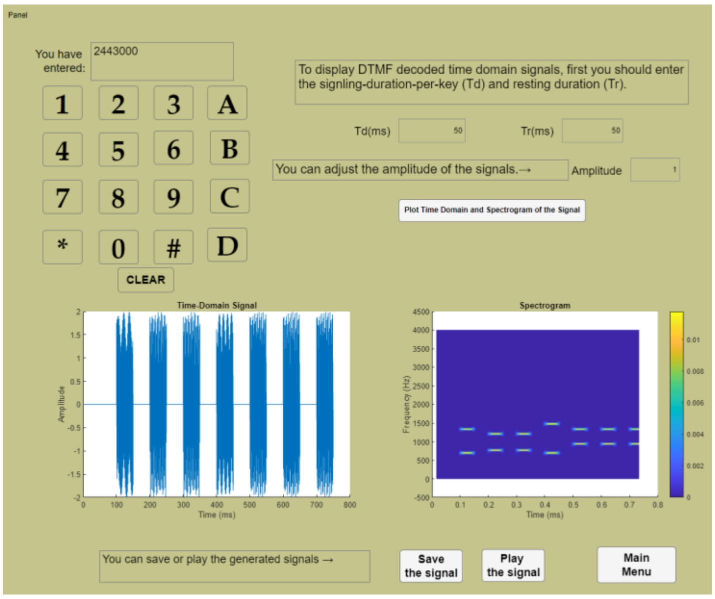
To display DTMF decoded time domain signals, first you should enter the signling-duration-per-key (Td) and resting duration (Tr).

Td(ms) 40 Tr(ms) 40

You can adjust the amplitude of the signals.→ Amplitude 1

You can save or play the generated signals →

Figure 7 Signaling [2,4,4,2,8,5,3] with DTMF (𝑇𝑑, 𝑇𝑟) = (50ms/50ms).



You have entered: 2443000

To display DTMF decoded time domain signals, first you should enter the signling-duration-per-key (Td) and resting duration (Tr).

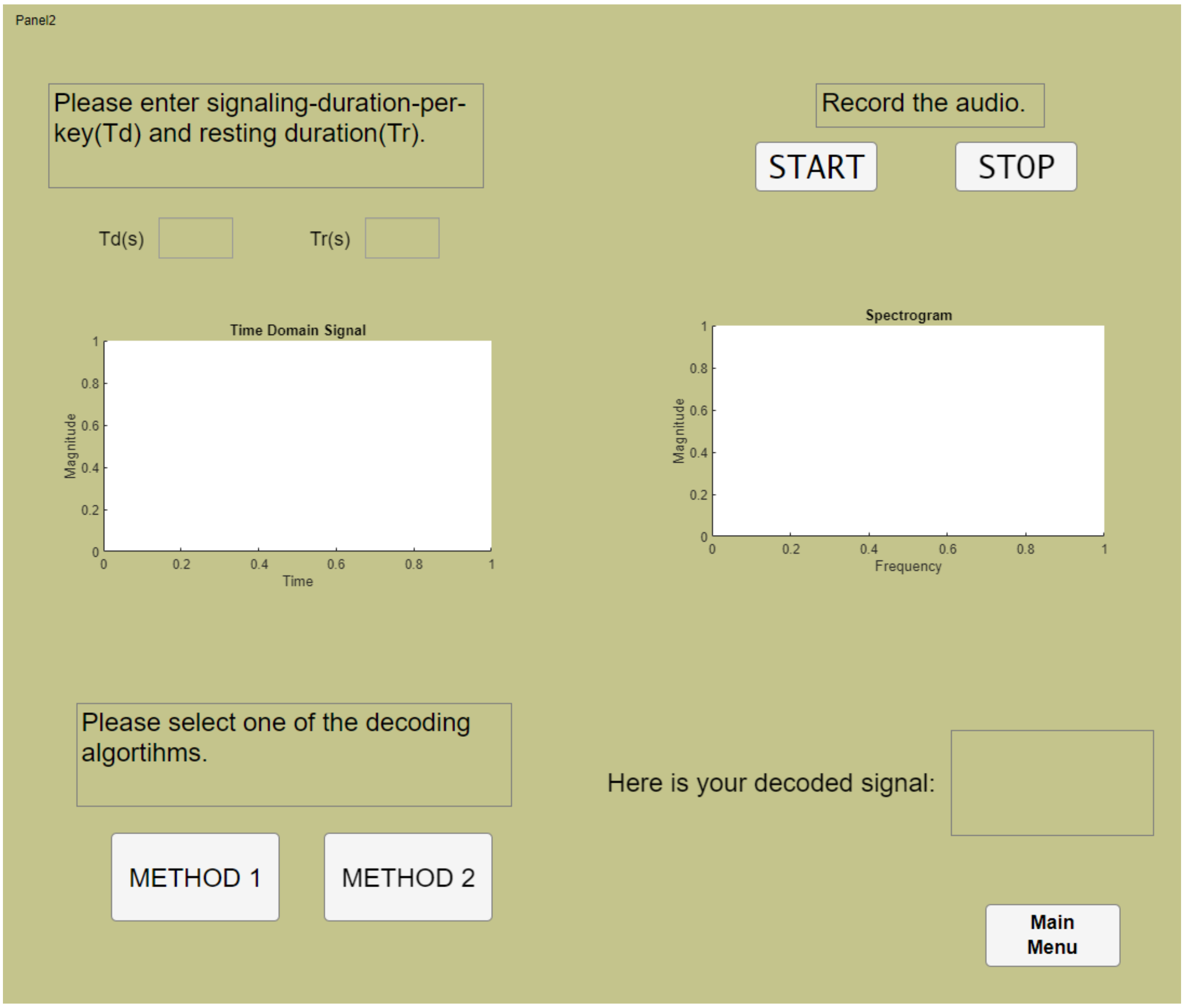
Td(ms) 50 Tr(ms) 50

You can adjust the amplitude of the signals.→ Amplitude 1

You can save or play the generated signals →

Figure 8 Signaling [2,4,4,3,0,0,0] with DTMF (𝑇𝑑, 𝑇𝑟) = (50ms/50ms).

Receiver Panel



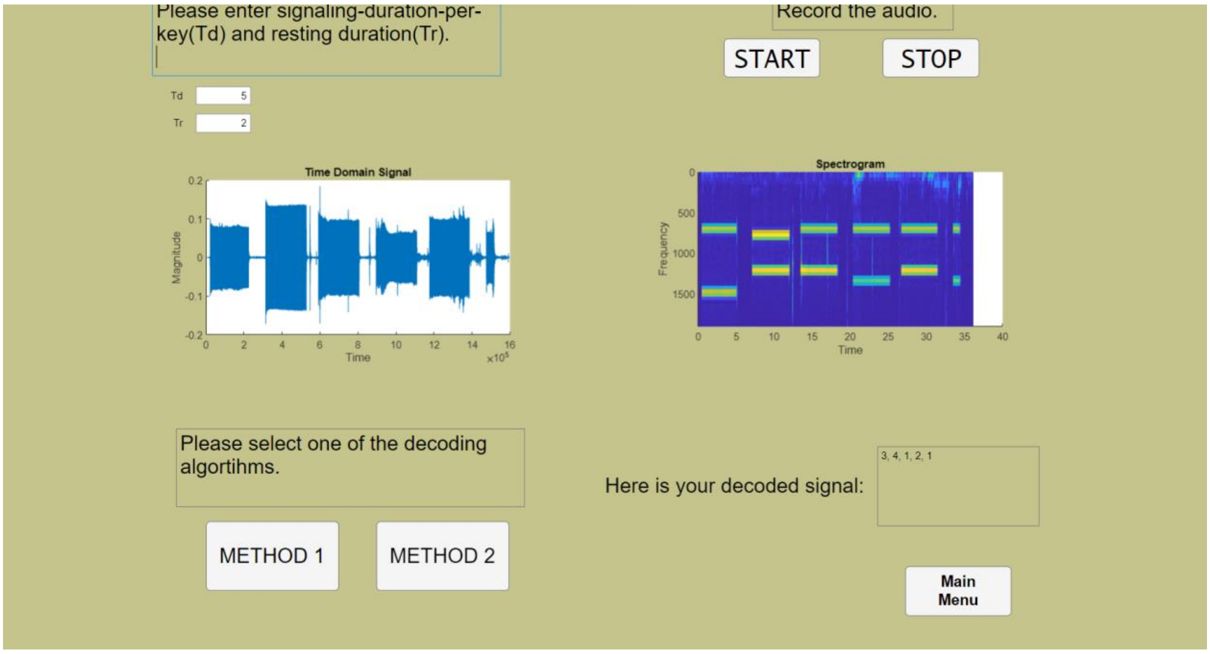
Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Please select one of the decoding algortihms.

Here is your decoded signal:

Figure 9 The screenshot of the Receiver Panel.

- After entering input values Td(s) and Tr(s), user can start to record audio by pushing “START” button and to end the recording process “STOP” button should be pushed.
- When the “STOP” button is pushed, Time Domain and Spectrogram graphs of the recorded signal will be plotted on the axes.
- User can choose any of two algorithms which are “Goertzel Algorithm” and “Spectrogram Based Algorithm” to decode the signal. After pushing the corresponding button, the decoded signal will be shown in the text area.
- By “Main Menu” button user can go back to main menu.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 5

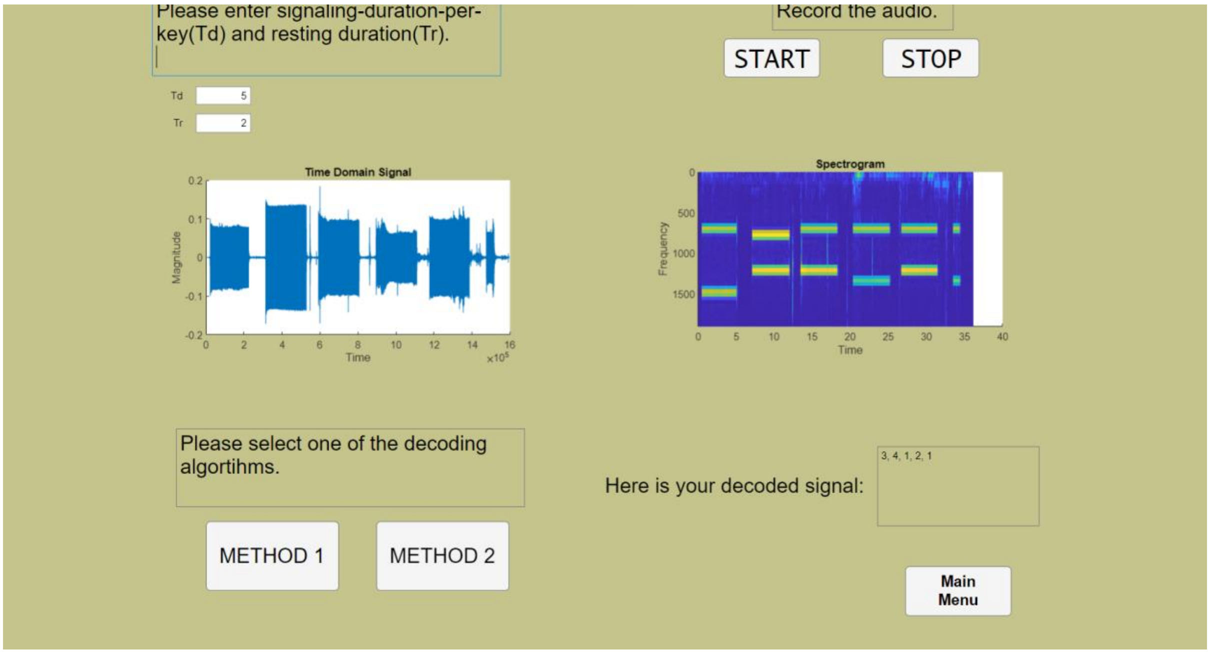
Tr 2

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 3, 4, 1, 2, 1

Figure 10 Example Decoding for [3,4,1,2,1] with Goertzel Algorithm



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 5

Tr 2

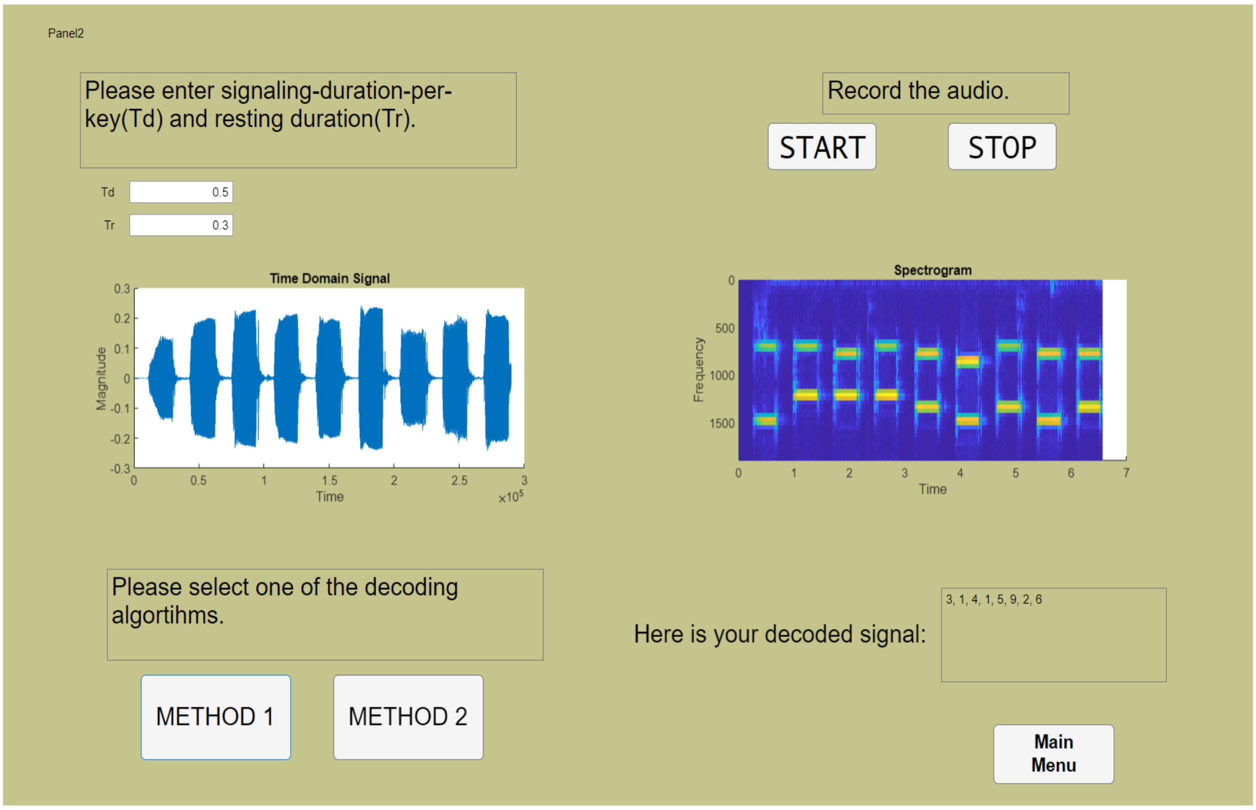
Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 3, 4, 1, 2, 1

Figure 11:Example Decoding for [3,4,1,2,1] with Spectrogram Based Algorithm

The implementation difference between Goertzel and spectrogram based algorithm is not clear here. However, in demonstration lab they will both be implemented.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 0.5

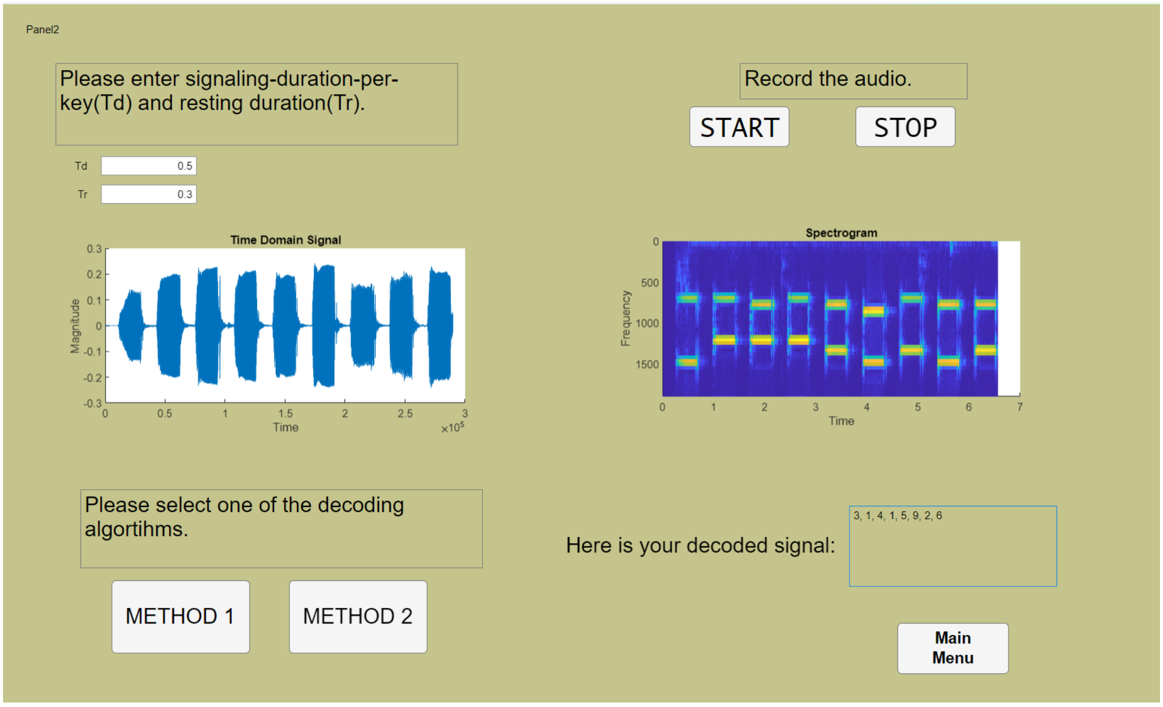
Tr 0.3

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 3, 1, 4, 1, 5, 9, 2, 6

Figure 12 Decoding of [3,1,4,1,5,9,2,6] with Spectrogram based method.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 0.5

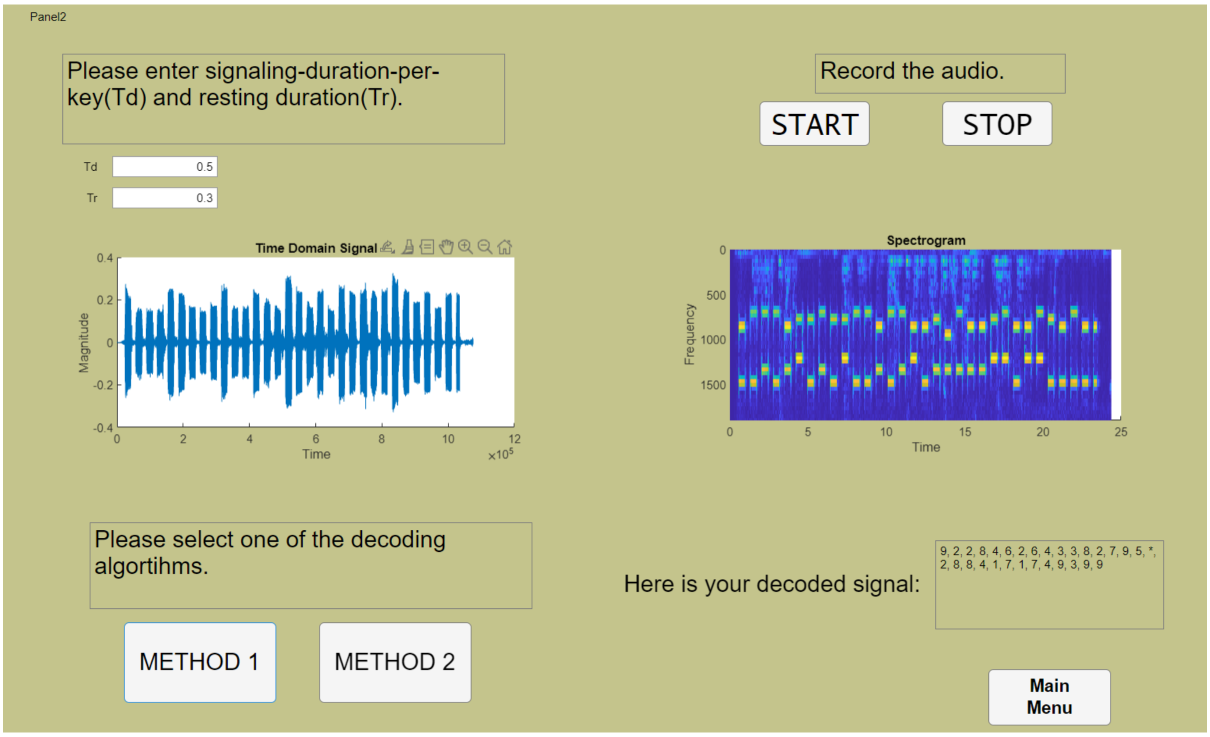
Tr 0.3

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 3, 1, 4, 1, 5, 9, 2, 6

Figure 13 Decoding of [3,1,4,5,9,2,6] with Goertzel



Panel2

Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 0.5

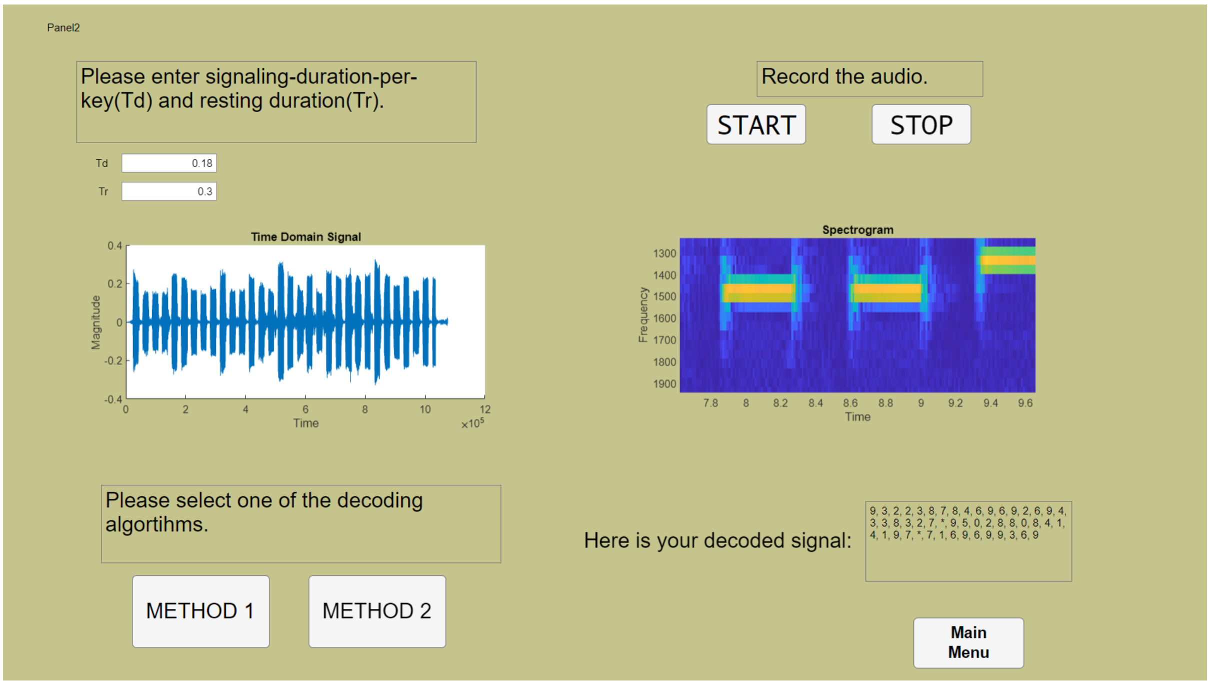
Tr 0.3

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 9, 2, 2, 8, 4, 6, 2, 6, 4, 3, 3, 8, 2, 7, 9, 5, *, 2, 8, 8, 4, 1, 7, 1, 7, 4, 9, 3, 9, 9

Figure 14 Decoding of [9, 2, 2, 8, 4, 6, 2, 6, 4, 3, 3, 8, 2, 7, 9, 5, 2, 8, 8, 4, 1, 7, 1, 7, 4, 9, 3, 9, 9] with Spectrogram algorithm



Panel2

Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 0.18

Tr 0.3

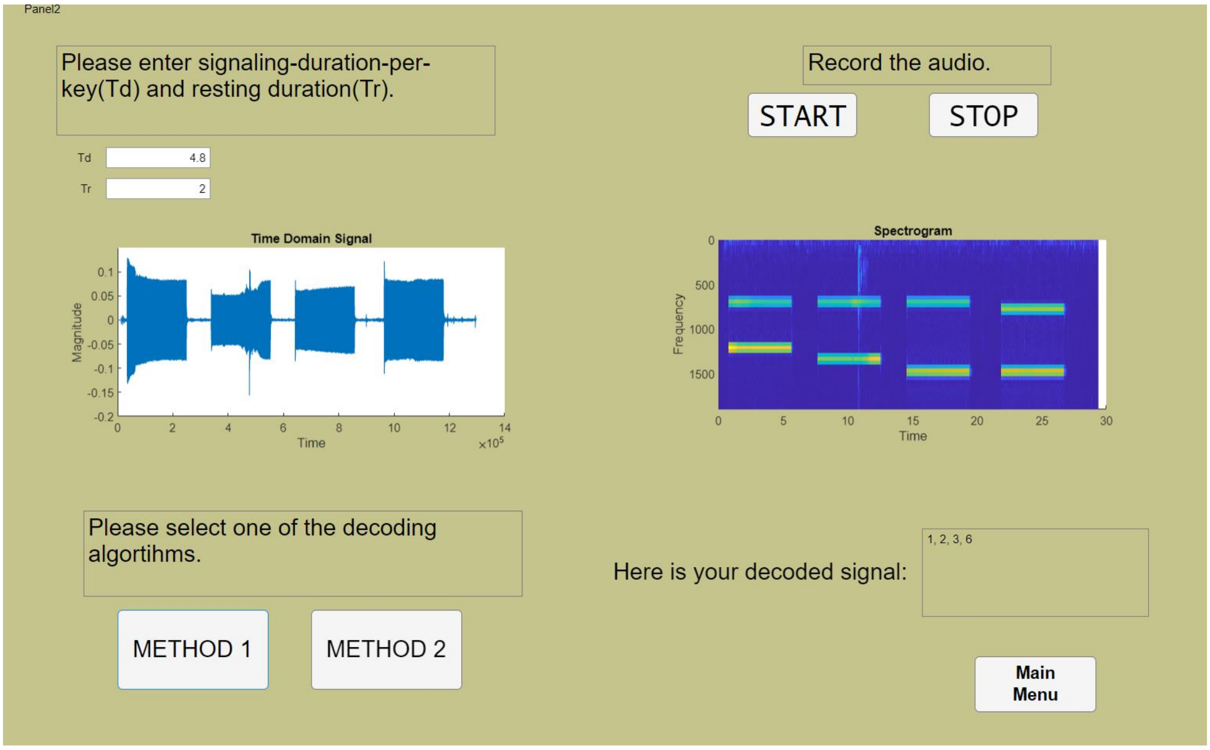
Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 9, 3, 2, 2, 3, 8, 7, 8, 4, 6, 9, 6, 9, 2, 6, 9, 4, 3, 3, 8, 3, 2, 7, *, 9, 5, 0, 2, 8, 8, 0, 8, 4, 1, 4, 1, 9, 7, *, 7, 1, 6, 9, 6, 9, 9, 3, 6, 9

Figure 15 Decoding of [9, 2, 2, 8, 4, 6, 2, 6, 4, 3, 3, 8, 2, 7, 9, 5, 2, 8, 8, 4, 1, 7, 1, 7, 4, 9, 3, 9, 9] with Goertzel algorithm

Divergence in Goertzel can be seen here due to short time duration.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 4.8

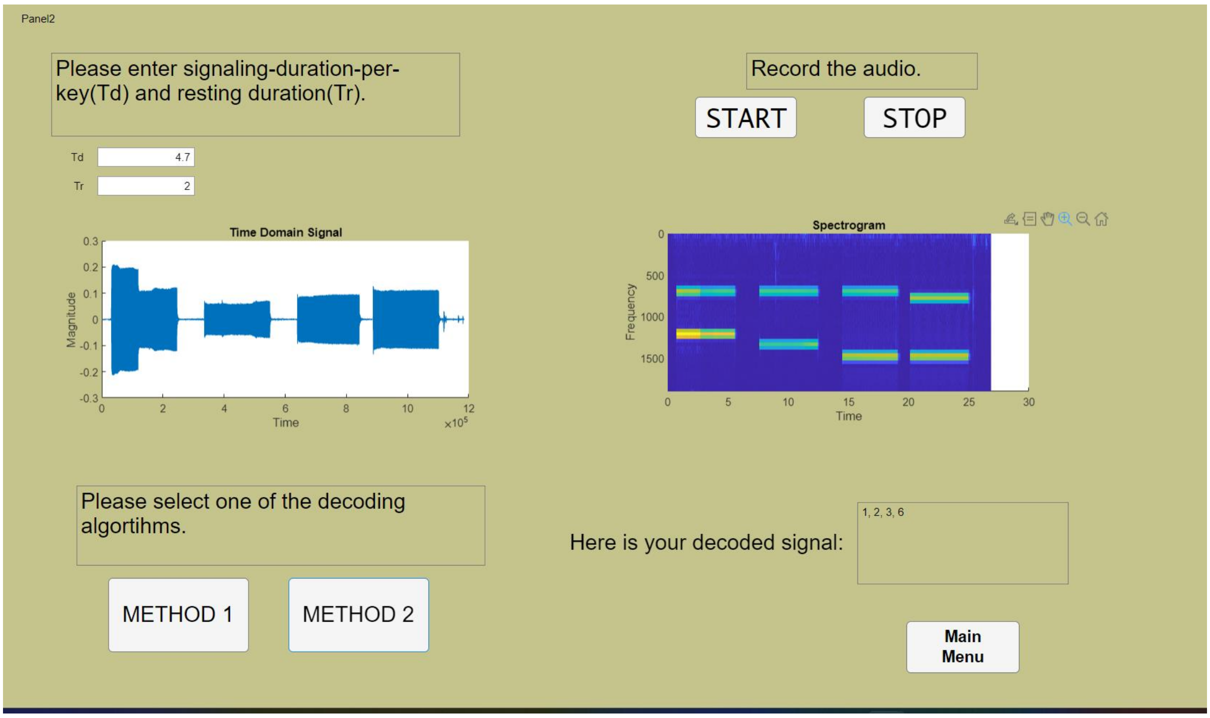
Tr 2

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 1, 2, 3, 6

Figure 16 Decoding [1,2,3,6] with Spectrogram Based Algorithm



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 4.7

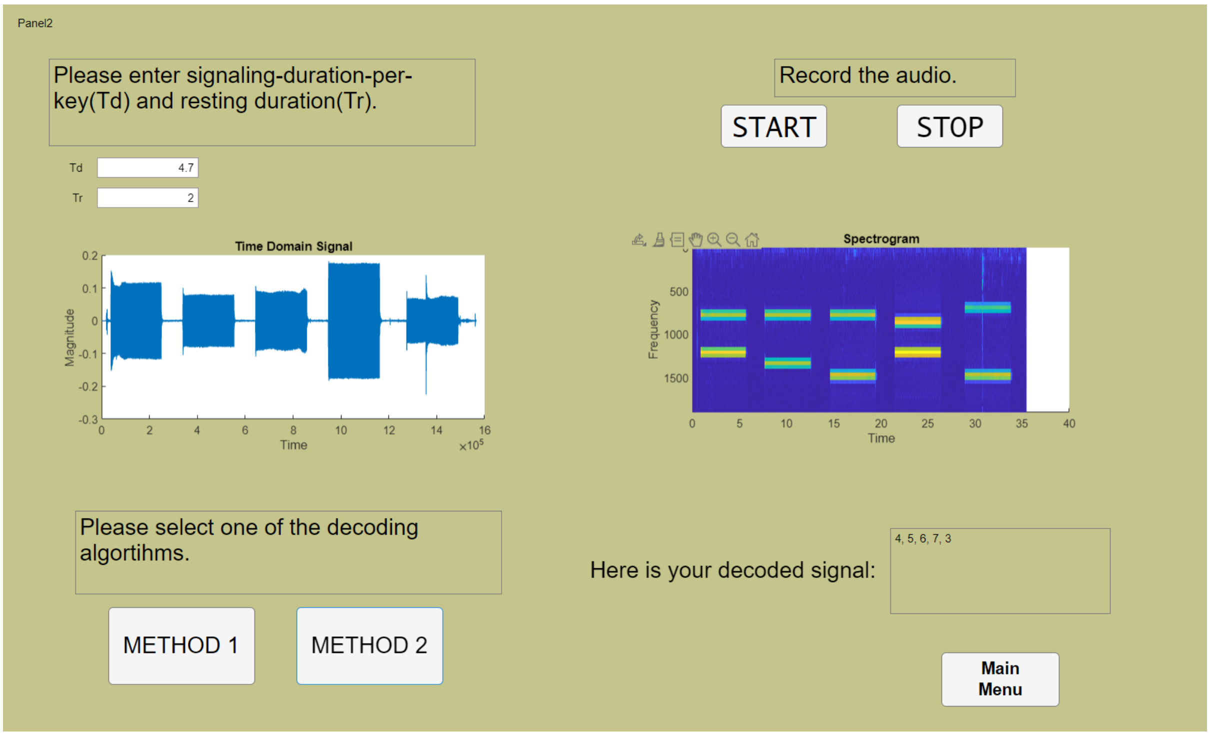
Tr 2

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 1, 2, 3, 6

Figure 1715 Decoding [1,2,3,6] with Goertzel.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 4.7

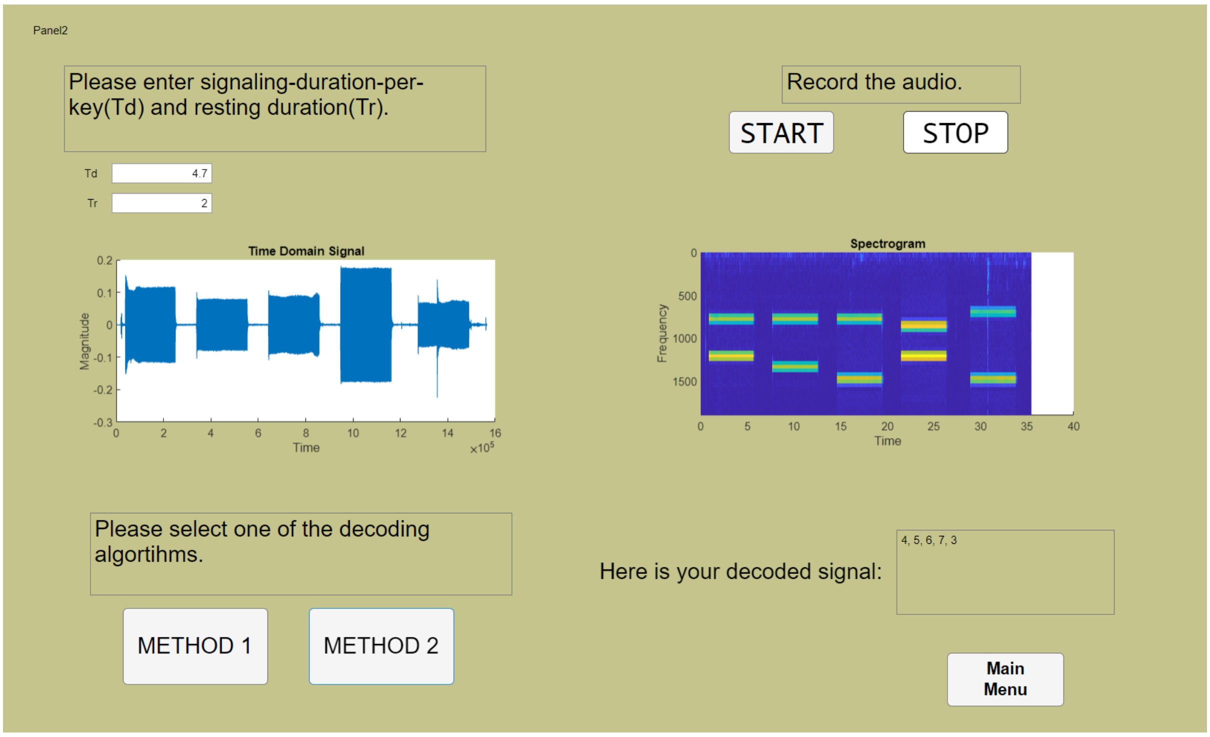
Tr 2

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 4, 5, 6, 7, 3

Figure 18 Encoding [4,5,6,7,3] with Spectrogram based algorithm.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 4.7

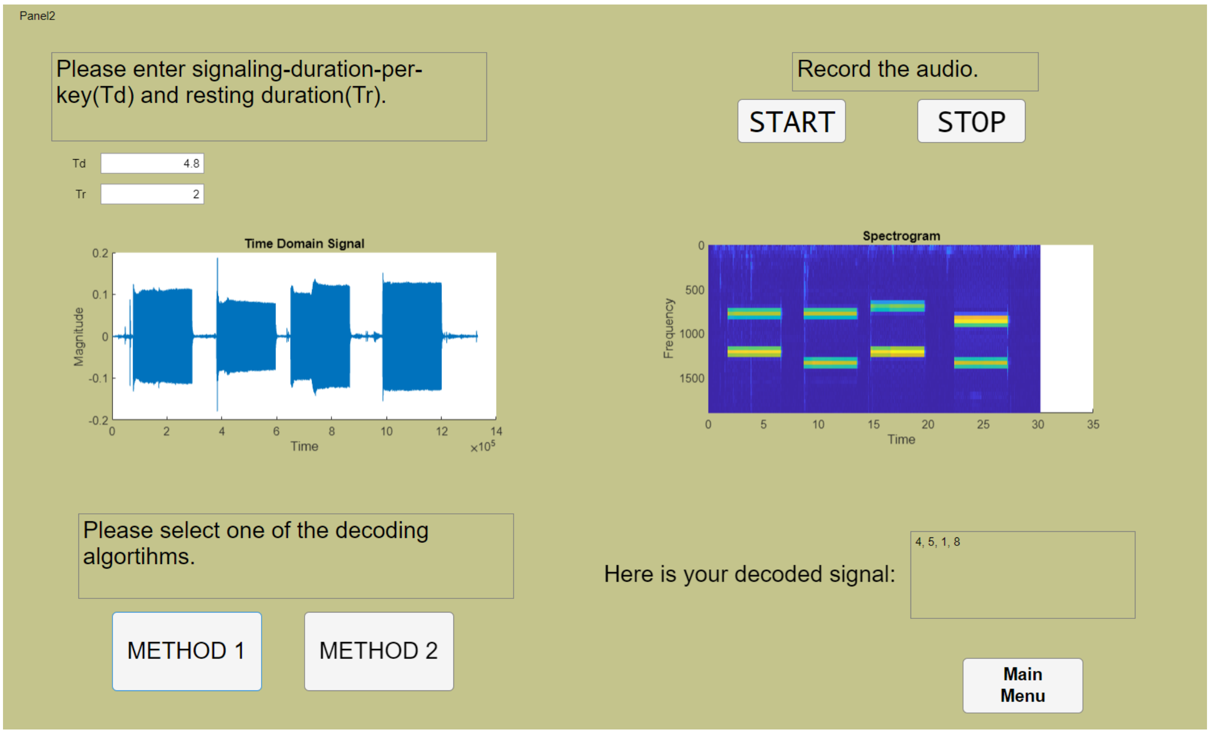
Tr 2

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 4, 5, 6, 7, 3

Figure 19 Decoding [4,5,6,7,3] with Goertzel.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 4.8

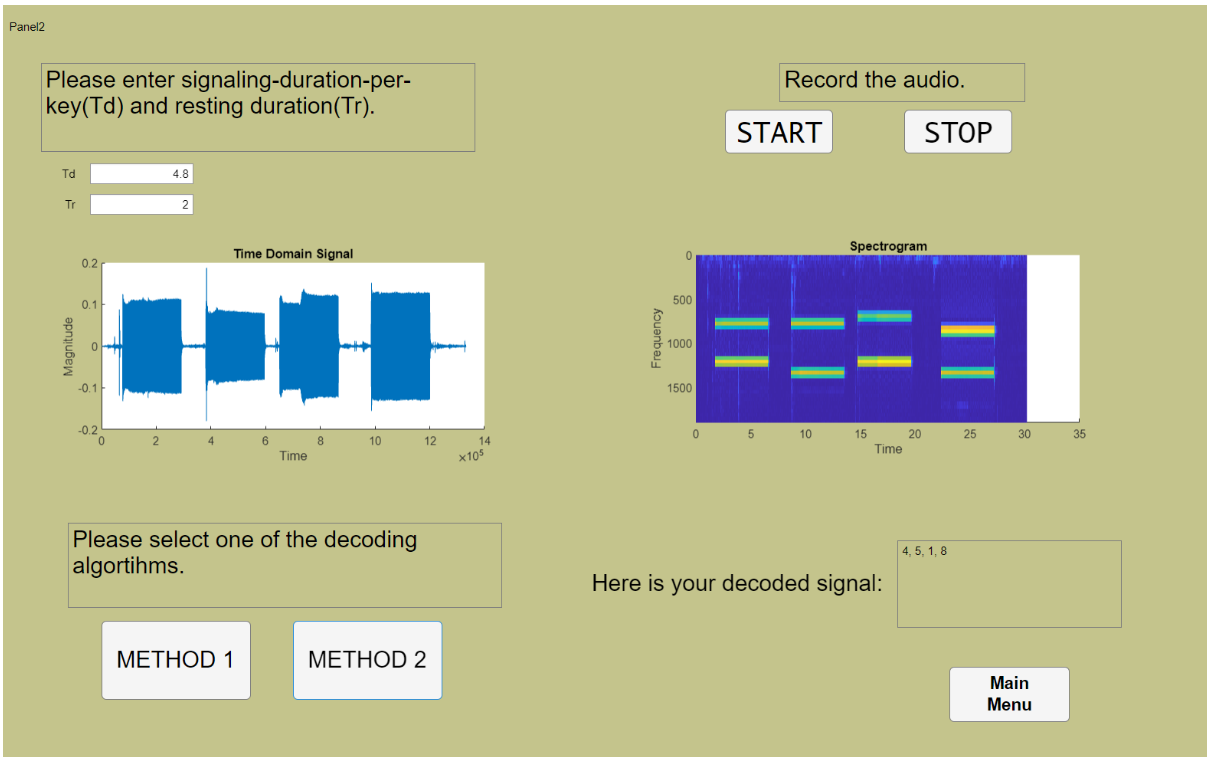
Tr 2

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 4, 5, 1, 8

Figure 20 Decoding [4,5,1,8] with Spectrogram Based Algorithm.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 4.8

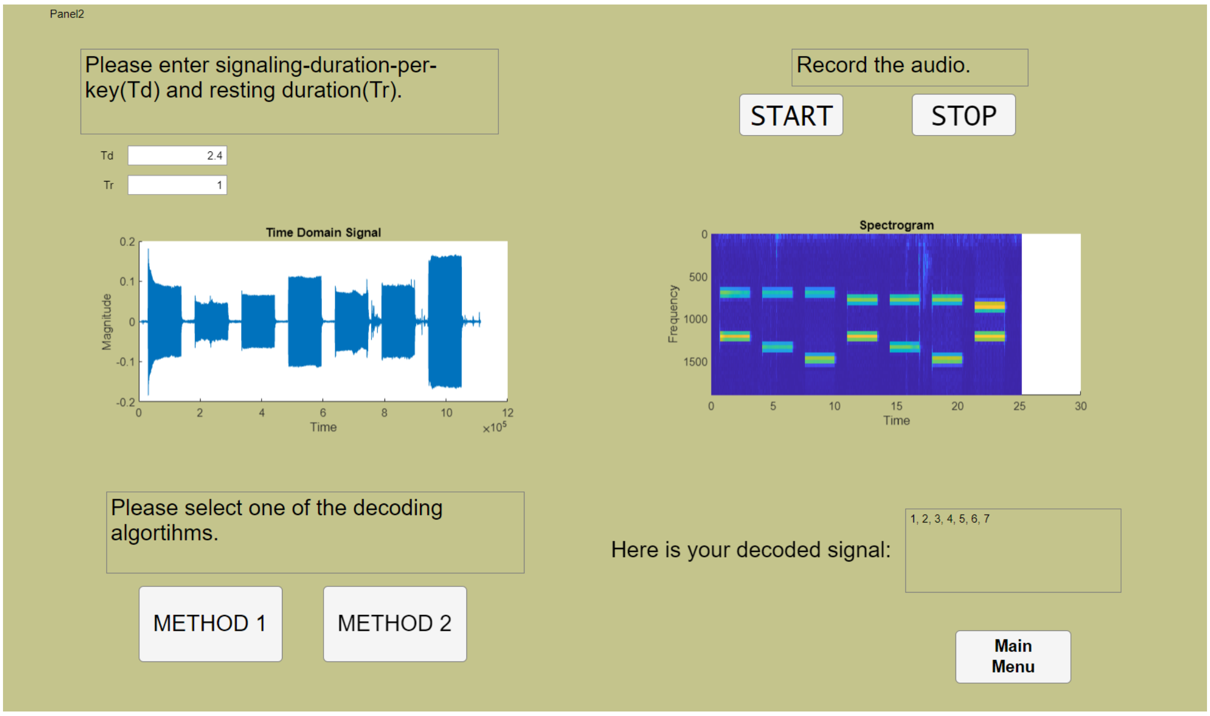
Tr 2

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 4, 5, 1, 8

Figure 21 Decoding [4,5,1,8] with Goertzel Based Algorithm.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 2.4

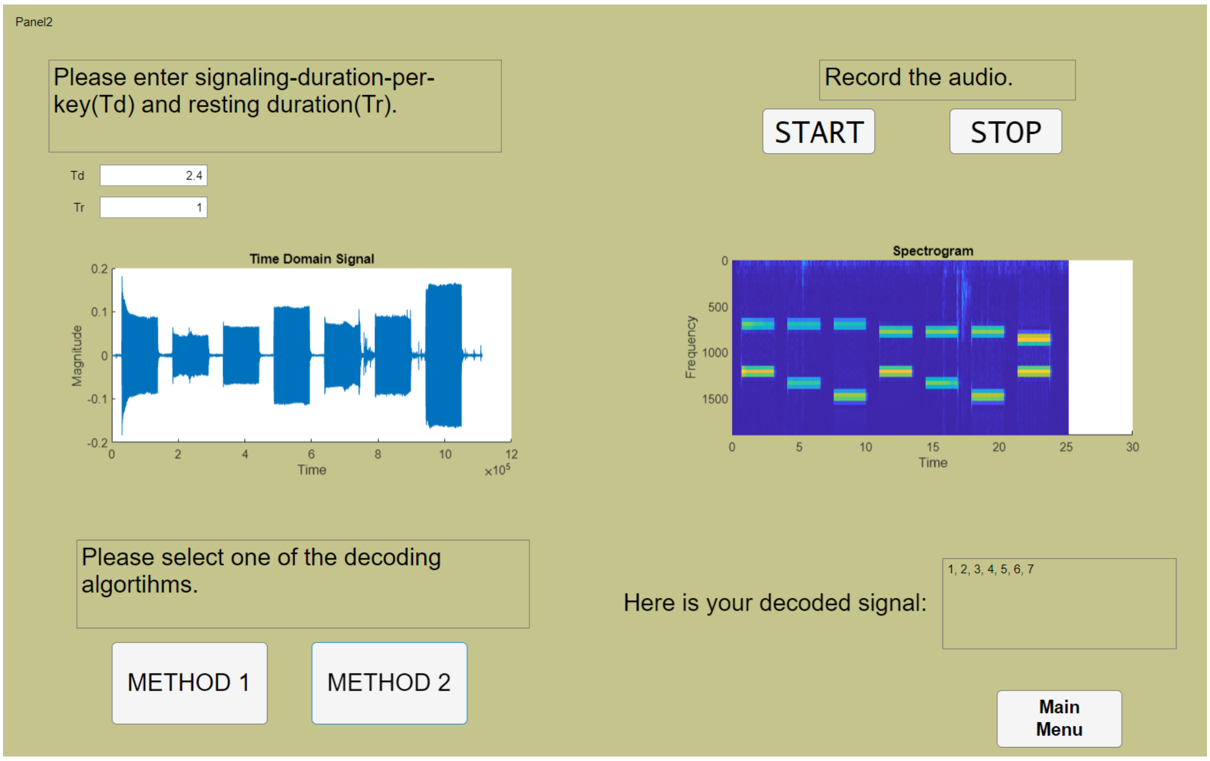
Tr 1

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 1, 2, 3, 4, 5, 6, 7

Figure 22 Decoding [1,2,3,4,5,6,7] with Goertzel Based Algorithm.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 2.4

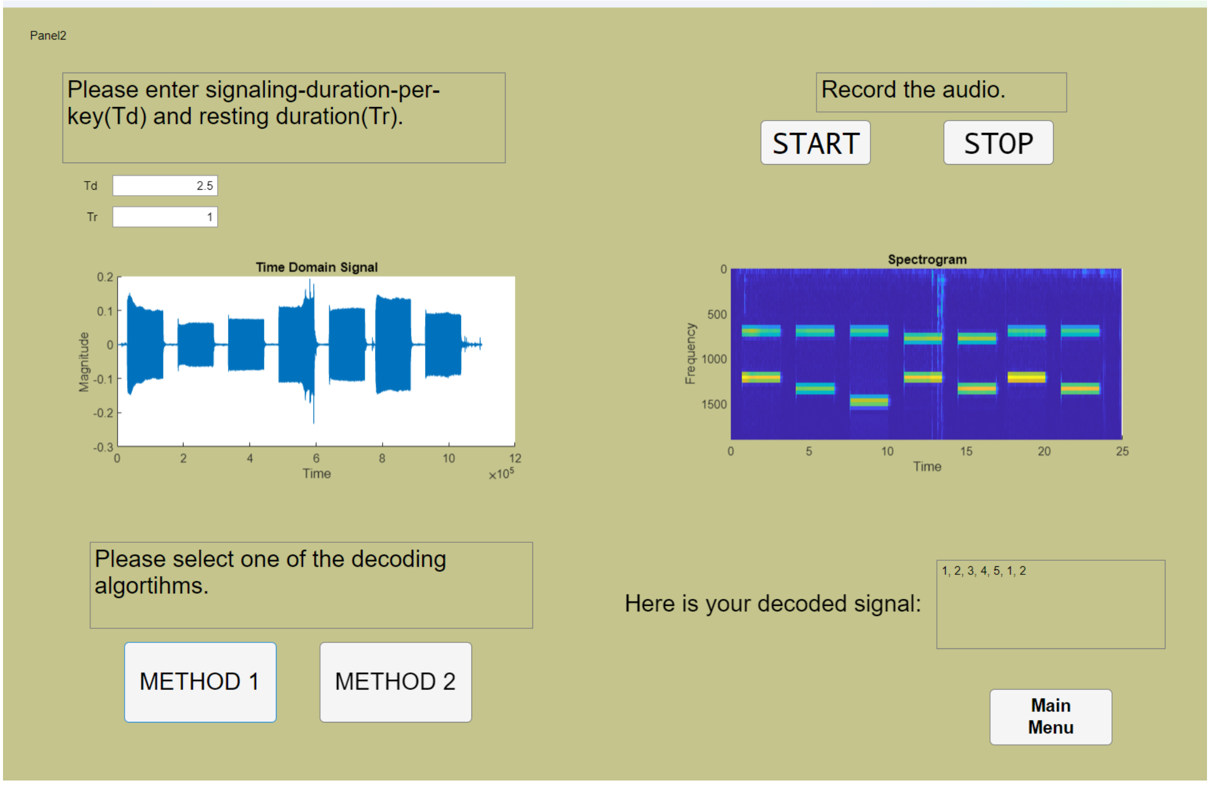
Tr 1

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 1, 2, 3, 4, 5, 6, 7

Figure 23 Decoding [1,2,3,4,5,6,7] with Goertzel Based Algorithm.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 2.5

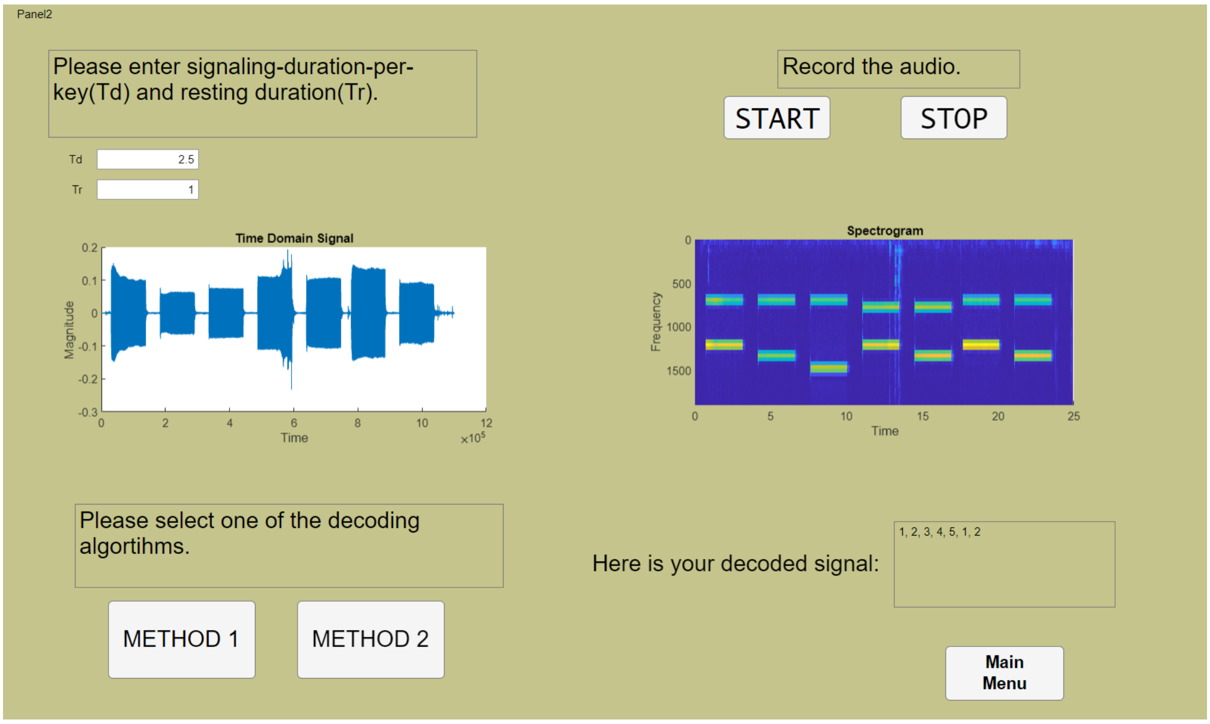
Tr 1

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 1, 2, 3, 4, 5, 1, 2

Figure 24 Decoding [1,2,3,4,5,1,2] with Spectrogram Based Algorithm.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 2.5

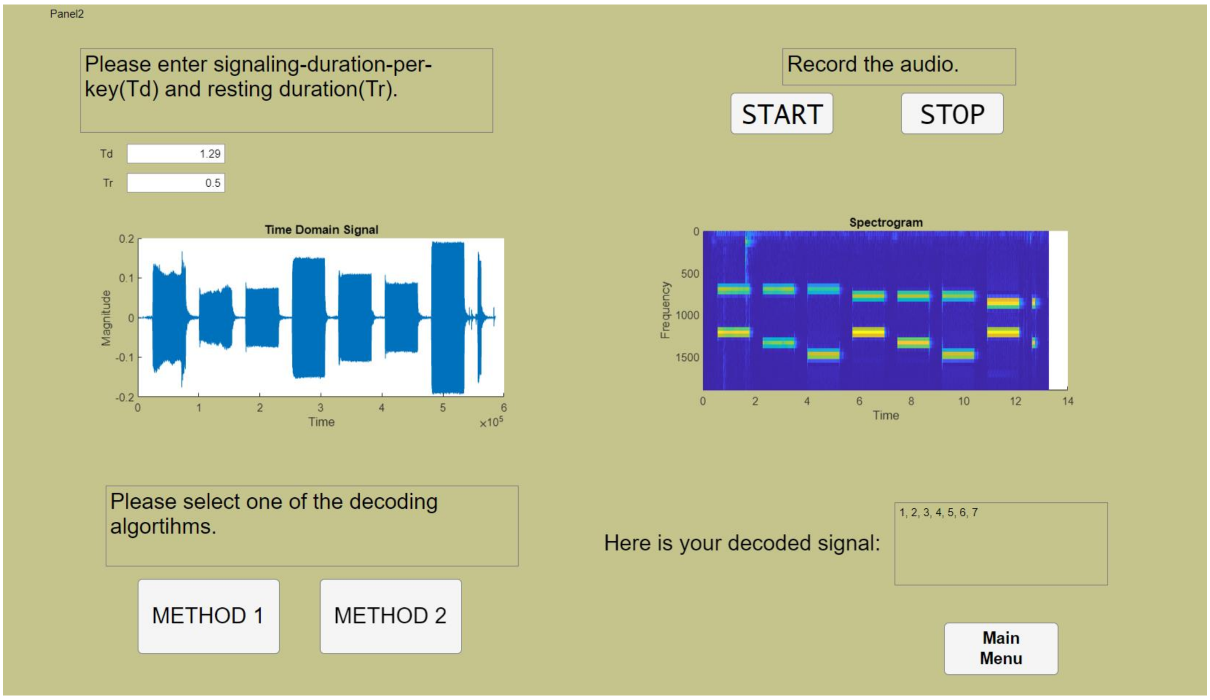
Tr 1

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 1, 2, 3, 4, 5, 1, 2

Figure 25 Decoding [1,2,3,4,5,1,2] with Goertzel Based Algorithm.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 1.29

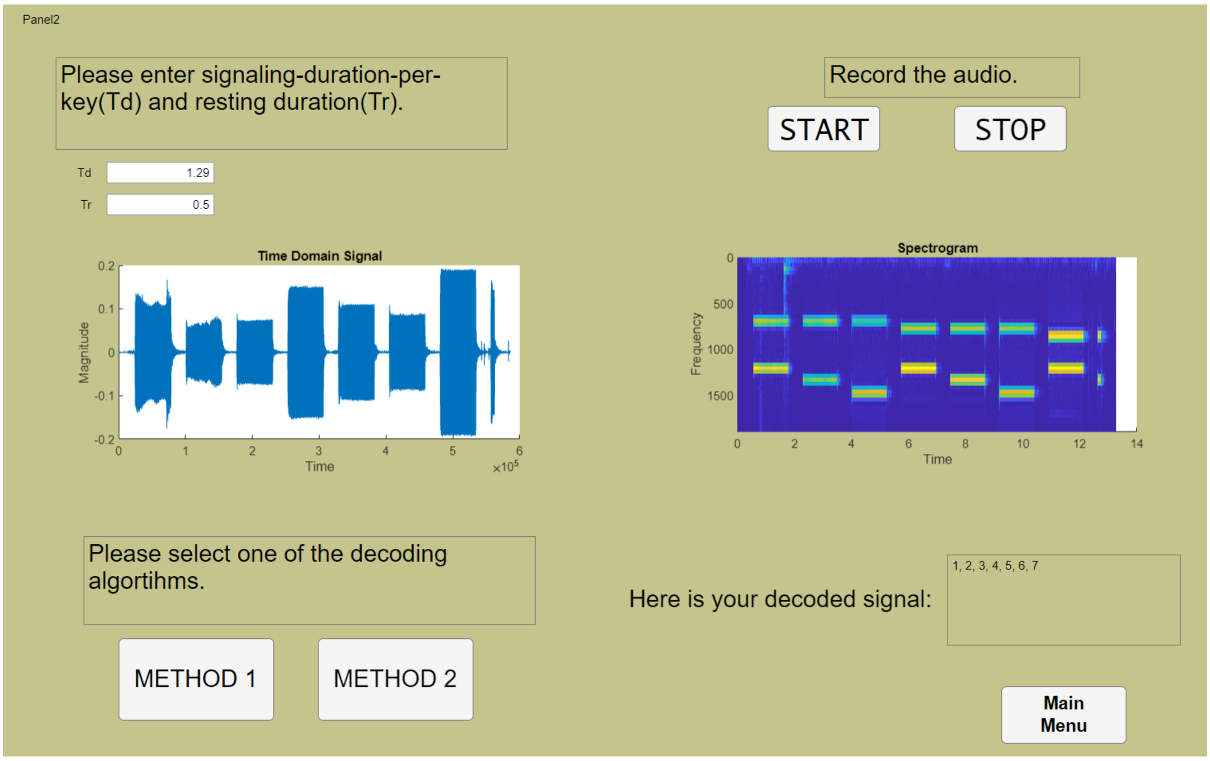
Tr 0.5

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 1, 2, 3, 4, 5, 6, 7

Figure 26 Decoding [1,2,3,4,5,1,2] with Spectrogram Based Algorithm.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 1.29

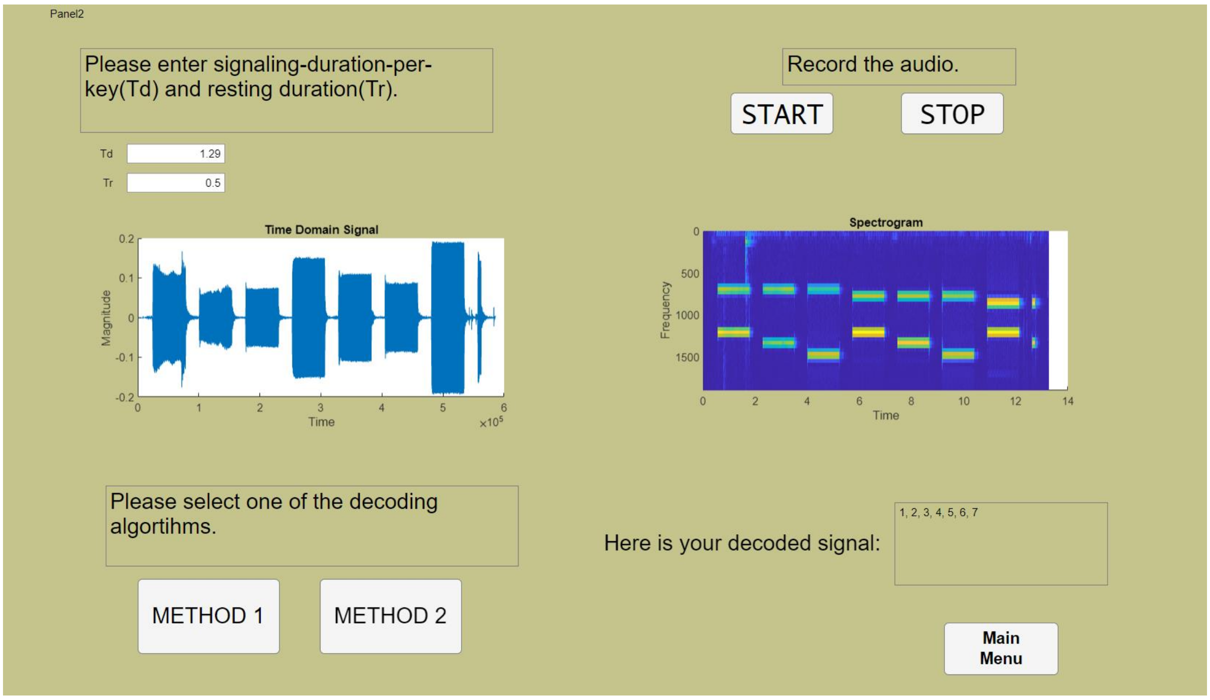
Tr 0.5

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 1, 2, 3, 4, 5, 6, 7

Figure 27 Decoding [1,2,3,4,5,1,2] with Goertzel Based Algorithm.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 1.29

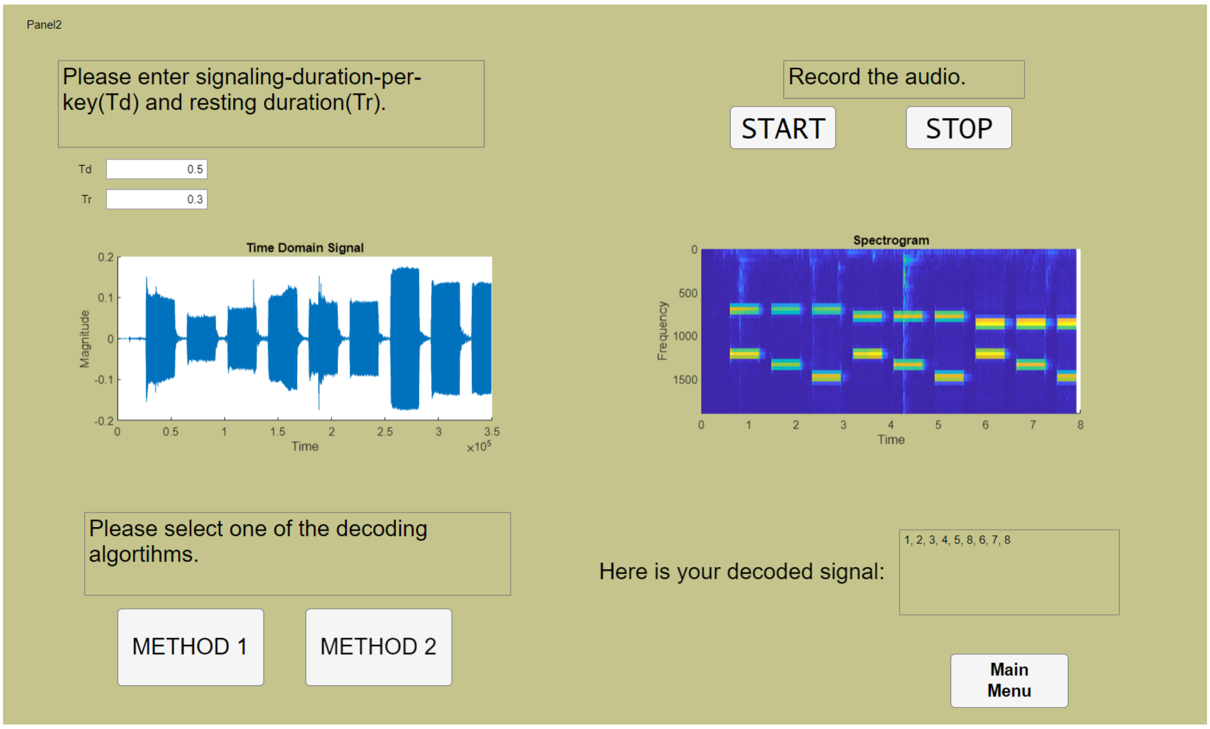
Tr 0.5

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 1, 2, 3, 4, 5, 6, 7

Figure 28 Decoding [1,2,3,4,5,6,7] with Spectrogram Based Algorithm.



Please enter signaling-duration-per-key(Td) and resting duration(Tr).

Td 0.5

Tr 0.3

Record the audio.

Please select one of the decoding algortihms.

Here is your decoded signal: 1, 2, 3, 4, 5, 8, 6, 7, 8

Figure 29 Decoding [1,2,3,4,5,6,7] with Goertzel Based Algorithm.

The Goertzel Algorithm excels in efficiently and accurately detecting specific frequencies, such as those used in DTMF signaling. It's also relatively easy to implement in both hardware and software. However, its focus on narrow frequency ranges limits its adaptability for broader spectral analysis.

In contrast, the Spectrogram-Based Algorithm offers versatility in analyzing a wide range of frequencies, providing a visual representation of frequency distribution over time through a spectrogram. This makes it suitable for diverse applications beyond DTMF. However, it demands more computational power and can be more complex to implement.

Ultimately, the optimal choice hinges on the specific requirements of your application. Prioritize the Goertzel Algorithm for efficient and precise detection of specific frequencies, while the Spectrogram-Based Algorithm shines in its flexibility and ability to visualize a broader spectrum.

Conclusion

In the field of telecommunications, DTMF signaling is vital, and choosing the right encoding tool is important. The Goertzel Algorithm is excellent for specific tone detection, making it ideal for direct DTMF applications. On the other hand, the Spectrogram-Based Algorithm allows for a broader frequency analysis, although it requires more computational effort and is complex to implement. Deciding between these algorithms depends on the needs of the project, including factors like the range of frequencies to be analyzed and the level of detail needed in the analysis. This comparison not only focuses on the algorithms themselves but also underscores their significance in efficiently handling DTMF signals, underlining the need for appropriate solutions in this area.

Appendix – MATLAB Codes

```
classdef dspprojectpart2 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                     matlab.ui.Figure
        transmitterpanel             matlab.ui.container.Panel
        DButton                      matlab.ui.control.Button
        CButton                      matlab.ui.control.Button
        BButton                      matlab.ui.control.Button
        AButton                      matlab.ui.control.Button
        MainMenuButton               matlab.ui.control.Button
        PlotTimeDomainandSpectrogramoftheSignalButton  matlab.ui.control.Button
        PlaythesignalButton          matlab.ui.control.Button
        SavethesignalButton          matlab.ui.control.Button
        TextArea_7                   matlab.ui.control.TextArea
        AmplitudeEditField           matlab.ui.control.NumericEditField
        AmplitudeEditFieldLabel      matlab.ui.control.Label
        TextArea_6                   matlab.ui.control.TextArea
        TrmsEditField                matlab.ui.control.NumericEditField
        TrmsEditFieldLabel           matlab.ui.control.Label
        TdmsEditField                matlab.ui.control.NumericEditField
        TdmsEditFieldLabel           matlab.ui.control.Label
        TextArea_5                   matlab.ui.control.TextArea
        CLEARButton                  matlab.ui.control.Button
        YouhaveenteredTextArea       matlab.ui.control.TextArea
        YouhaveenteredTextAreaLabel  matlab.ui.control.Label
        Button_12                    matlab.ui.control.Button
        Button_11                    matlab.ui.control.Button
        Button_10                    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_5                     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
        UIAxes_2                     matlab.ui.control.UIAxes
        UIAxes                       matlab.ui.control.UIAxes
        receiverpanel                matlab.ui.container.Panel
        MainMenuButton_2             matlab.ui.control.Button
        HereisyourdecodedsignalTextArea  matlab.ui.control.TextArea
        HereisyourdecodedsignalTextAreaLabel  matlab.ui.control.Label
        METHOD2Button                matlab.ui.control.Button
        METHOD1Button                matlab.ui.control.Button
        TextArea_10                  matlab.ui.control.TextArea
        STOPButton                   matlab.ui.control.Button
        STARTButton                  matlab.ui.control.Button
        TextArea_9                   matlab.ui.control.TextArea
        TrsEditField                 matlab.ui.control.NumericEditField
        TrsEditFieldLabel            matlab.ui.control.Label
        TdsEditField                 matlab.ui.control.NumericEditField
        TdsEditFieldLabel            matlab.ui.control.Label
        TextArea_8                   matlab.ui.control.TextArea
        UIAxes2_2                    matlab.ui.control.UIAxes
        UIAxes2                      matlab.ui.control.UIAxes
        EncodeDTMFSignalsReceiverButton  matlab.ui.control.Button
        DecodeDTMFSignalsTransmitterButton  matlab.ui.control.Button
        TextArea_4                   matlab.ui.control.TextArea
        TextArea_3                   matlab.ui.control.TextArea
        EXITButton                   matlab.ui.control.Button
        TextArea_2                   matlab.ui.control.TextArea
    end


    properties (Access = public)
        dtmfSignal
        enteredkey
        Recorder=[] % Description
        audioData=[]
    end

    methods (Access = public)

        function dtmf_signal = generate_key_signal(~, indexLow, indexHigh, Td)
            % DTMF frequencies for the 16 keys
            fL = [697 770 852 941];
            fH = [1209 1336 1477 1633];
            fs = 44100; % Sampling frequency
            t = 0:1/fs:Td; % Time vector in seconds
            % Generate DTMF signal
            dtmf_signal = (sin(2 * pi * fL(indexLow) * t) + sin(2 * pi *
fH(indexHigh) * t)) ;
        end

        function superpositioned_signal = generate_dtmf_signal(app, key, Td, Tr)
            keypad = ['1', '2', '3', 'A', '4', '5', '6', 'B', '7', '8', '9', 'C',
'*', '0', '#', 'D'];
            fs = 44100; % Sampling frequency

            % Calculate the total number of samples needed
            total_samples = length(key) * (Td * fs) + (length(key) - 1) * (Tr *
fs);
            superpositioned_signal = zeros(1, total_samples);

            for k = 1:length(key)
                if isempty(key{k})
                    continue; % Skip empty entries
                end

                key_index = find(keypad == key{k}, 1); % Ensure only one index is
returned

                lF_index = idivide(int8(key_index - 1), int8(4)) + 1;
                hF_index = mod(int8(key_index - 1), int8(4)) + 1;

                one_tone = app.generate_key_signal(lF_index, hF_index, Td);

                % Ensure one_tone has the correct length
                one_tone = one_tone(1:Td * fs);

                % Determine the start and end indices for the current tone
                tone_start = (k-1) * ((Td + Tr) * fs) + 1;
                tone_end = tone_start + length(one_tone) - 1;

                % Assign the tone to the appropriate place in the signal array
                superpositioned_signal(tone_start:tone_end) = one_tone;

            end
            app.dtmfSignal = superpositioned_signal;
        end
    end



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

        % Button pushed function: DecodeDTMFSignalsTransmitterButton
        function DecodeDTMFSignalsTransmitterButtonPushed(app, event)
            app.TextArea_4.Visible ="off";
            app.TextArea_3.Visible ="off";
            app.TextArea_2.Visible ="off";
            app.EncodeDTMFSignalsReceiverButton.Visible="off";
            app.DecodeDTMFSignalsTransmitterButton.Visible="off";
            app.transmitterpanel.Visible="on";
        end

        % Button pushed function: MainMenuButton
        function MainMenuButtonPushed(app, event)
            app.TextArea_4.Visible ="on";
            app.TextArea_3.Visible ="on";
            app.TextArea_2.Visible ="on";
            app.EncodeDTMFSignalsReceiverButton.Visible="on";
            app.DecodeDTMFSignalsTransmitterButton.Visible="on";
            app.transmitterpanel.Visible="off";
        end

        % Button pushed function: EncodeDTMFSignalsReceiverButton
        function EncodeDTMFSignalsReceiverButtonPushed(app, event)
            app.TextArea_4.Visible ="off";
            app.TextArea_3.Visible ="off";
            app.TextArea_2.Visible ="off";
            app.EncodeDTMFSignalsReceiverButton.Visible="off";
            app.DecodeDTMFSignalsTransmitterButton.Visible="off";
            app.receiverpanel.Visible="on";
        end

        % Button pushed function: MainMenuButton_2
        function MainMenuButton_2Pushed(app, event)
            app.TextArea_4.Visible ="on";
            app.TextArea_3.Visible ="on";
            app.TextArea_2.Visible ="on";
            app.EncodeDTMFSignalsReceiverButton.Visible="on";
            app.DecodeDTMFSignalsTransmitterButton.Visible="on";
            app.receiverpanel.Visible="off";
        end

        % Button pushed function: CLEARButton
        function CLEARButtonPushed(app, event)
            app.YouhaveenteredTextArea.Value = '';
            cla(app.UIAxes_2);
```

```
            cla(app.UIAxes);
            app.enteredkey = '';
        end

        % Button pushed function: Button_1
        function Button_1Pushed(app, event)
            app.YouhaveenteredTextArea.Value =
strcat(app.YouhaveenteredTextArea.Value,'1');
            app.enteredkey = strcat(app.enteredkey, ' 1');
        end

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

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

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

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

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

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

        % Button pushed function: Button_8
        function Button_8Pushed(app, event)
            app.YouhaveenteredTextArea.Value =
strcat(app.YouhaveenteredTextArea.Value,'8');
            app.enteredkey = strcat(app.enteredkey,' 8');
        end
        % Button pushed function: Button_9
        function Button_9Pushed(app, event)
            app.YouhaveenteredTextArea.Value =
strcat(app.YouhaveenteredTextArea.Value,'9');
            app.enteredkey = strcat(app.enteredkey, ' 9');
        end

        % Button pushed function: Button_10
        function Button_10Pushed(app, event)
            app.YouhaveenteredTextArea.Value =
strcat(app.YouhaveenteredTextArea.Value,'*');
            app.enteredkey = strcat(app.enteredkey, ' *');
        end

        % Button pushed function: Button_11
        function Button_11Pushed(app, event)
            app.YouhaveenteredTextArea.Value =
strcat(app.YouhaveenteredTextArea.Value,'0');
            app.enteredkey = strcat(app.enteredkey, ' 0');
        end

        % Button pushed function: Button_12
        function Button_12Pushed(app, event)
            app.YouhaveenteredTextArea.Value =
strcat(app.YouhaveenteredTextArea.Value,'#');
            app.enteredkey = strcat(app.enteredkey, ' #');
        end

        % Button pushed function:
        % PlotTimeDomainandSpectrogramoftheSignalButton
        function PlotTimeDomainandSpectrogramoftheSignalButtonPushed(app, event)
            key_cell = strsplit(app.enteredkey);
            keysignal = app.generate_dtmf_signal(key_cell, app.TdmsEditField.Value
/ 1000 ,app.TrmsEditField.Value / 1000) .* app.AmplitudeEditField.Value;
            % Time-domain plot
            fs = 44100;
            cla(app.UIAxes);
            plot(app.UIAxes, (0:length(keysignal)-1)*(1000/fs), keysignal); %
Scale time to ms
            xlabel(app.UIAxes, 'Time (ms)');
            ylabel(app.UIAxes, 'Amplitude');

            % Spectrogram adjustments
            noverlap =  512; % Adjust as necessary
            window = 1024;
            nfft = window;
            % Plot the spectrogram
            cla(app.UIAxes_2); % Clear the previous plot
            [S, F, T] = spectrogram(keysignal, window, noverlap, nfft, fs);
            logSpectrogram_1 = 10 * log10(abs(S) + 1);
            imagesc(app.UIAxes_2, T, F, logSpectrogram_1);
            axis(app.UIAxes_2, 'xy');
            xlabel(app.UIAxes_2, 'Time (s)'); % Change to seconds if T is in
seconds
            ylabel(app.UIAxes_2, 'Frequency (Hz)');
            colorbar(app.UIAxes_2);
            % Optionally, add a label to the colorbar
            ylabel(app.UIAxes_2.Colorbar, 'Power/Frequency (dB)');
        end

        % Button pushed function: AButton
        function AButtonPushed(app, event)
            app.YouhaveenteredTextArea.Value =
strcat(app.YouhaveenteredTextArea.Value,'A');
            app.enteredkey = strcat(app.enteredkey, ' A');
        end

        % Button pushed function: BButton
        function BButtonPushed(app, event)
            app.YouhaveenteredTextArea.Value =
strcat(app.YouhaveenteredTextArea.Value,'B');
            app.enteredkey = strcat(app.enteredkey, ' B');
        end

        % Button pushed function: CButton
        function CButtonPushed(app, event)
            app.YouhaveenteredTextArea.Value =
strcat(app.YouhaveenteredTextArea.Value,'C');
            app.enteredkey = strcat(app.enteredkey, ' C');
        end

        % Button pushed function: DButton
        function DButtonPushed(app, event)
            app.YouhaveenteredTextArea.Value =
strcat(app.YouhaveenteredTextArea.Value,'D');
            app.enteredkey = strcat(app.enteredkey, ' D');
        end

        % Button pushed function: SavethesignalButton
        function SavethesignalButtonPushed(app, event)
            fs = 44100; % Replace with your actual sampling frequency
            [file, path] = uiputfile('*.wav', 'Save as WAV file');
            if isequal(file, 0) || isequal(path, 0)
                disp('User pressed cancel.');
            else
                fullSavePath = fullfile(path, file);
                audiowrite(fullSavePath, app.dtmfSignal, fs);
                disp(['File saved to: ', fullSavePath]);
            end
        end

        % Button pushed function: PlaythesignalButton
        function PlaythesignalButtonPushed(app, event)
            fs = 44100;
            sound(app.dtmfSignal, fs);
        end

        % Button pushed function: EXITButton
        function EXITButtonPushed(app, event)
            delete(app.UIFigure);
        end

        % Button pushed function: STARTButton
        function STARTButtonPushed(app, event)
            % Set the audio parameters
            fs = 44100;          % Sampling rate (in Hz)
            % Create an audiorecorder object
            app.Recorder = audiorecorder(fs, 16, 1);
            % Start recording
            disp('Recording...');
            record(app.Recorder);
        end

        % Button pushed function: STOPButton
        function STOPButtonPushed(app, event)
            stop(app.Recorder);
            % Get the recorded audio data
            app.audioData = getaudiodata(app.Recorder);
            plot(app.UIAxes2,app.audioData );
            % Set spectrogram parameters
            windowSize = 1024;  % Size of the window for each segmen
            overlap = 512;      % Overlap between consecutive segments
            % Create spectrogram
            [S, F, T] = spectrogram(app.audioData , windowSize, overlap,
windowSize, 44100, 'yaxis');
            % Use logarithmic scale for better visualization
            logSpectrogram = 10 * log10(abs(S) + 1);  % Adding 1 to avoid log(0)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % Plot the logarithmic scale spectrogram
            imagesc(app.UIAxes2_2,T, F, logSpectrogram);

        end

        % Button pushed function: METHOD1Button
        function METHOD1ButtonPushed(app, event)
            Tred=app.TrsEditField.Value;
            Tded=app.TdsEditField.Value; % Duration of extracted segment from each
chunk
            Ts = Tred+Tded;   % Total duration of each chunk
            Td=Tded;
            Fs=44100;
            numChunks = floor(length(app.audioData) / (Ts * 44100));



            currentList = {};
            for i = 1:numChunks
                % Extract the current chunk
                chunkStart = round((i - 1) * Ts * Fs) + 1;
                chunkEnd = round(i * Ts * Fs);
                currentChunk = app.audioData(chunkStart:chunkEnd);

                % Extract the desired duration from the chunk
                extractedSegment = currentChunk(1:Td * Fs);
                plot(extractedSegment)

                N=4000;
                sample_rate=44100;
                targetFrequencies = [697,770,852,941,1209,1336,1477,1633];
                flag0=0;
                flag1=0;
                flag2=0;
                flag3=0;
                flag4=0;
                flag5=0;
```

```
flag6=0;
flag7=0;
flag8=0;
flag9=0;
flag10=0;
flag11=0;
flag12=0;
flag13=0;
flag14=0;
flag15=0;
flag16=0;
flag17=0;
flag18=0;
for m=1:length(targetFrequencies)

    target_freq=targetFrequencies(m);
    k=0.5+(N*target_freq)/sample_rate;
    w=((2*pi)/N)*k;
    cosine=cos(w);
    sine=sin(w);
    coeff=2*cosine;
    for i = 1:N:length(extractedSegment)
        endIndex = min(i + N - 1, length(extractedSegment));
        segment = extractedSegment(i:endIndex);
        q1=0;
        q2=0;
        k=endIndex-i;
        for j=1:1:k
            % Process or analyze the current segment here
            q0=coeff*q1-q2+segment(j);
            q2=q1;
            q1=q0;
            h_mag=sqrt(q1^2+q2^2-q1*q2*coeff);

            if h_mag>60

                if(target_freq==697)&&flag0==0
                    flag0=1;
                    disp(target_freq)
                end
                if (target_freq==770)&&flag1==0
                    flag1=1;
                    disp(target_freq)
                end
                if (target_freq==852)&&flag2==0
                    flag2=1;
                    disp(target_freq)
                end
                if (target_freq==941)&&flag3==0
                    flag3=1;
                    disp(target_freq)
                end
                if (target_freq==1336)&&flag4==0
                    flag4=1;
                    disp(target_freq)
                end
                if (target_freq==1477)&&flag5==0
                    flag5=1;
                    disp(target_freq)
```

```
end
if (target_freq==1209)&&flag6==0
    flag6=1;
    disp(target_freq)
end
if(flag0==1&&flag6==1)&&flag7==0
    disp("1")
    flag7=1;
    flag0=0;
    flag6=0;
    currentList = [currentList, '1'];
elseif(flag0==1&&flag4==1)&&flag8==0
    disp("2")
    flag8=1;
    flag0=0;
    flag4=0;
    currentList = [currentList, '2'];
elseif(flag0==1&&flag5==1)&&flag9==0
    disp("3")
    flag9=1;
    flag0=0;
    flag5=0;
    currentList = [currentList, '3'];
elseif(flag1==1&&flag6==1)&&flag10==0
    disp("4")
    flag10=1;
    flag1=0;
    flag6=0;
    currentList = [currentList, '4'];
elseif(flag1==1&&flag4==1)&&flag11==0
    disp("5")
    flag11=1;
    flag1=0;
    flag4=0;
    currentList = [currentList, '5'];
elseif(flag1==1&&flag5==1)&&flag12==0
    disp("6")
    flag12=1;
    flag1=0;
    flag5=0;
    currentList = [currentList, '6'];
elseif(flag2==1&&flag6==1)&&flag13==0
    disp("7")
    flag13=1;
    flag2=0;
    flag6=0;
    currentList = [currentList, '7'];
elseif(flag2==1&&flag4==1)&&flag14==0
    disp("8")
    flag14=1;
    flag2=0;
    flag4=0;
    currentList = [currentList, '8'];
elseif(flag2==1&&flag5==1)&&flag15==0
    disp("9")
    flag15=1;
    flag2=0;
    flag5=0;
    currentList = [currentList, '9'];
```

```
                                elseif(flag3==1&&flag6==1)&&flag16==0
                                    disp("*")
                                    flag16=1;
                                    flag3=0;
                                    flag6=0;
                                    currentList = [currentList, '*'];
                                elseif(flag3==1&&flag4==1)&&flag17==0
                                    disp("0")
                                    flag17=1;
                                    flag3=0;
                                    flag4=0;
                                    currentList = [currentList, '0'];
                                elseif(flag3==1&&flag5==1)&&flag18==0
                                    disp("#")
                                    flag18=1;
                                    flag3=0;
                                    flag5=0;
                                    currentList = [currentList, '#'];
                                end
                            end
                            % powerSignal = mean(h_mag.^2);
                            % if powerSignal>20
                            % % Calculate power
                            %     plot(h_mag);
                            %     fprintf('The power of the signal is: %f\n',
powerSignal);
                            % end
                        end
                    end
                end
                myValuesString = strjoin(currentList, ', ');
                disp(myValuesString)
                % Update the TextArea with the updated text
                app.HereisyourdecodedsignalTextArea.Value = myValuesString;


            end

        end

        % Button pushed function: METHOD2Button
        function METHOD2ButtonPushed(app, event)
            Tred=app.TrsEditField.Value;
            Tded=app.TdsEditField.Value; % Duration of extracted segment from each
chunk
            Ts = Tred+Tded;   % Total duration of each chunk
            Td=Tded;
            Fs=44100;
            numChunks = floor(length(app.audioData) / (Ts * 44100));


            currentList2 = {};
            for i = 1:numChunks
                chunkStart = round((i - 1) * Ts * Fs) + 1;
                chunkEnd = round(i * Ts * Fs);
                currentChunk = app.audioData(chunkStart:chunkEnd);

                extractedSegment = currentChunk(1:Td * Fs);
                windowSize = 1024;  % Size of the window for each segmen
                overlap = 512;      % Overlap between consecutive segments
                [S, F, T] = spectrogram(extractedSegment , windowSize, overlap,
windowSize, 44100, 'yaxis');
                %logSpectrogram = 10 * log10(abs(S) + 1);  % Adding 1 to avoid
log(0)
                %imagesc(app.UIAxes2_2, F, T, logSpectrogram);
                ylim([450,1900]);
                f0 = 0;
                f1 = 0;
                f2 = 0;
                f3 = 0;
                f4 = 0;
                f5 = 0;
                f6 = 0;
                f7 = 0;
                f8 = 0;
                f9 = 0;
                f10 = 0;
                f11 = 0;
                f12 = 0;
                f13 = 0;
                f14 = 0;
                f15 = 0;
                f16 = 0;
                f17 = 0;
                f18 = 0;
                for i=1:length(T)-1

                    for k=1:length(F)-1

                        S_p=abs(S(k,i));
                        if(S_p>5)
                            detected=F(k);

                            if (677 < detected) && (detected < 717) && (f0 == 0)
                                f0 = 1;
                            end

                            if (750 < detected) && (detected < 790) && (f1 == 0)
                                f1 = 1;
                            end

                            if (832 < detected) && (detected < 872) && (f2 == 0)
                                f2 = 1;
                            end

                            if (921 < detected) && (detected < 961) && (f3 == 0)
                                f3 = 1;
                            end

                            if (1189 < detected) && (detected < 1229) && (f4 == 0)
                                f4 = 1;
                            end

                            if (1316 < detected) && (detected < 1356) && (f5 == 0)
                                f5 = 1;
                            end

                            if (1457 < detected) && (detected < 1497) && (f6 == 0)
```

```
    f6 = 1;
end



if (f0 && f4) && f7 == 0
    disp("1");
    f7 = 1;
    currentList2 = [currentList2, '1'];
end

if (f0 && f5) && f8 == 0
    disp("2");
    f8 = 1;
    currentList2 = [currentList2, '2'];
end

if (f0 && f6) && f9 == 0
    disp("3");
    f9 = 1;
    currentList2 = [currentList2, '3'];
end

if (f1 && f4) && f10 == 0
    disp("4");
    f10 = 1;
    currentList2 = [currentList2, '4'];
end

if (f1 && f5) && f11 == 0
    disp("5");
    f11 = 1;
    currentList2 = [currentList2, '5'];
end

if (f1 && f6) && f12 == 0
    disp("6");
    f12 = 1;
    currentList2 = [currentList2, '6'];
end

if (f2 && f4) && f13 == 0
    disp("7");
    f13 = 1;
    currentList2 = [currentList2, '7'];
end

if (f2 && f5) && f14 == 0
    disp("8");
    f14 = 1;
    currentList2 = [currentList2, '8'];
end

if (f2 && f6) && f15 == 0
    disp("9");
    f15 = 1;
    currentList2 = [currentList2, '9'];
end
```

```
                        if (f3 && f4) && f16 == 0
                            disp("*");
                            f16 = 1;
                            currentList2 = [currentList2, '*'];
                        end

                        if (f3 && f5) && f17 == 0
                            disp("0");
                            f17 = 1;
                            currentList2 = [currentList2, '0'];
                        end

                        if (f3 && f6) && f18 == 0
                            disp("#");
                            f18 = 1;
                            currentList2 = [currentList2, '#'];
                        end



                    end
                end

            end

            myValuesString2 = strjoin(currentList2, ', ');
            disp(myValuesString2)
            % Update the TextArea with the updated text
            app.HereisyourdecodedsignalTextArea.Value = myValuesString2;
        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 = [0.7686 0.7686 0.549];
        app.UIFigure.Position = [100 100 1316 1024];
        app.UIFigure.Name = 'MATLAB App';

        % Create TextArea_2
        app.TextArea_2 = uitextarea(app.UIFigure);
        app.TextArea_2.Editable = 'off';
        app.TextArea_2.HorizontalAlignment = 'center';
        app.TextArea_2.FontName = 'Century Schoolbook';
        app.TextArea_2.FontSize = 48;
        app.TextArea_2.FontWeight = 'bold';
        app.TextArea_2.FontAngle = 'italic';
        app.TextArea_2.BackgroundColor = [0.7686 0.7686 0.549];
        app.TextArea_2.Position = [363 783 678 158];
        app.TextArea_2.Value = {'Welcome to METUCall !'; ''};

        % Create EXITButton
        app.EXITButton = uibutton(app.UIFigure, 'push');
```

```
            app.EXITButton.ButtonPushedFcn = createCallbackFcn(app, 
@EXITButtonPushed, true);
            app.EXITButton.BackgroundColor = [0.7686 0.7686 0.549];
            app.EXITButton.FontSize = 24;
            app.EXITButton.FontWeight = 'bold';
            app.EXITButton.Position = [645 157 200 70];
            app.EXITButton.Text = 'EXIT';

            % Create TextArea_3
            app.TextArea_3 = uitextarea(app.UIFigure);
            app.TextArea_3.Editable = 'off';
            app.TextArea_3.HorizontalAlignment = 'center';
            app.TextArea_3.FontSize = 18;
            app.TextArea_3.BackgroundColor = [0.7686 0.7686 0.549];
            app.TextArea_3.Position = [758 814 276 60];
            app.TextArea_3.Value = {'designed by Erturkmen & Demir.'};

            % Create TextArea_4
            app.TextArea_4 = uitextarea(app.UIFigure);
            app.TextArea_4.HorizontalAlignment = 'center';
            app.TextArea_4.FontSize = 36;
            app.TextArea_4.BackgroundColor = [0.7686 0.7686 0.549];
            app.TextArea_4.Position = [436 671 563 52];
            app.TextArea_4.Value = {'What do you want to do?'};

            % Create DecodeDTMFSignalsTransmitterButton
            app.DecodeDTMFSignalsTransmitterButton = uibutton(app.UIFigure, 
'push');
            app.DecodeDTMFSignalsTransmitterButton.ButtonPushedFcn = 
createCallbackFcn(app, @DecodeDTMFSignalsTransmitterButtonPushed, true);
            app.DecodeDTMFSignalsTransmitterButton.BackgroundColor = [0.7686 
0.7686 0.549];
            app.DecodeDTMFSignalsTransmitterButton.FontSize = 24;
            app.DecodeDTMFSignalsTransmitterButton.Position = [311 336 313 243];
            app.DecodeDTMFSignalsTransmitterButton.Text = {'Decode DTMF Signals'; 
'(Transmitter)'};

            % Create EncodeDTMFSignalsReceiverButton
            app.EncodeDTMFSignalsReceiverButton = uibutton(app.UIFigure, 'push');
            app.EncodeDTMFSignalsReceiverButton.ButtonPushedFcn = 
createCallbackFcn(app, @EncodeDTMFSignalsReceiverButtonPushed, true);
            app.EncodeDTMFSignalsReceiverButton.BackgroundColor = [0.7686 0.7686 
0.549];
            app.EncodeDTMFSignalsReceiverButton.FontSize = 24;
            app.EncodeDTMFSignalsReceiverButton.Position = [843 337 313 242];
            app.EncodeDTMFSignalsReceiverButton.Text = {'Encode DTMF Signals'; 
'(Receiver)'};

            % Create receiverpanel
            app.receiverpanel = uipanel(app.UIFigure);
            app.receiverpanel.BorderType = 'none';
            app.receiverpanel.BorderWidth = 0;
            app.receiverpanel.Title = 'Panel2';
            app.receiverpanel.Visible = 'off';
            app.receiverpanel.BackgroundColor = [0.7686 0.7686 0.549];
            app.receiverpanel.Position = [135 45 1134 952];

            % Create UIAxes2
            app.UIAxes2 = uiaxes(app.receiverpanel);
            title(app.UIAxes2, 'Time Domain Signal')
            xlabel(app.UIAxes2, 'Time')
            ylabel(app.UIAxes2, 'Magnitude')
            zlabel(app.UIAxes2, 'Z')
            app.UIAxes2.Position = [50 403 414 253];

            % Create UIAxes2_2
            app.UIAxes2_2 = uiaxes(app.receiverpanel);
            title(app.UIAxes2_2, 'Spectrogram')
            xlabel(app.UIAxes2_2, 'Time (s)')
            ylabel(app.UIAxes2_2, 'Frequency')
            zlabel(app.UIAxes2_2, 'Z')
            app.UIAxes2_2.YLim = [450 1900];
            app.UIAxes2_2.Position = [628 418 414 253];

            % Create TextArea_8
            app.TextArea_8 = uitextarea(app.receiverpanel);
            app.TextArea_8.FontSize = 24;
            app.TextArea_8.BackgroundColor = [0.7686 0.7686 0.549];
            app.TextArea_8.Position = [37 784 410 98];
            app.TextArea_8.Value = {'Please enter signaling-duration-per-key(Td)
and resting duration(Tr).'; ''};

            % Create TdsEditFieldLabel
            app.TdsEditFieldLabel = uilabel(app.receiverpanel);
            app.TdsEditFieldLabel.HorizontalAlignment = 'right';
            app.TdsEditFieldLabel.FontSize = 18;
            app.TdsEditFieldLabel.Position = [80 724 47 23];
            app.TdsEditFieldLabel.Text = 'Td(s)';

            % Create TdsEditField
            app.TdsEditField = uieditfield(app.receiverpanel, 'numeric');
            app.TdsEditField.AllowEmpty = 'on';
            app.TdsEditField.BackgroundColor = [0.7686 0.7686 0.549];
            app.TdsEditField.Position = [141 717 70 38];
            app.TdsEditField.Value = [];

            % Create TrsEditFieldLabel
            app.TrsEditFieldLabel = uilabel(app.receiverpanel);
            app.TrsEditFieldLabel.HorizontalAlignment = 'right';
            app.TrsEditFieldLabel.FontSize = 18;
            app.TrsEditFieldLabel.Position = [280 724 42 23];
            app.TrsEditFieldLabel.Text = 'Tr(s)';

            % Create TrsEditField
            app.TrsEditField = uieditfield(app.receiverpanel, 'numeric');
            app.TrsEditField.AllowEmpty = 'on';
            app.TrsEditField.BackgroundColor = [0.7686 0.7686 0.549];
            app.TrsEditField.Position = [336 717 70 38];
            app.TrsEditField.Value = [];

            % Create TextArea_9
            app.TextArea_9 = uitextarea(app.receiverpanel);
            app.TextArea_9.FontSize = 24;
            app.TextArea_9.BackgroundColor = [0.7686 0.7686 0.549];
            app.TextArea_9.Position = [765 841 213 41];
            app.TextArea_9.Value = {'Record the audio.'};

            % Create STARTButton
            app.STARTButton = uibutton(app.receiverpanel, 'push');
            app.STARTButton.ButtonPushedFcn = createCallbackFcn(app,
@STARTButtonPushed, true);
            app.STARTButton.FontName = 'Lucida Sans Typewriter';
            app.STARTButton.FontSize = 30;
            app.STARTButton.Position = [708 781 113 46];
            app.STARTButton.Text = 'START';

            % Create STOPButton
            app.STOPButton = uibutton(app.receiverpanel, 'push');
            app.STOPButton.ButtonPushedFcn = createCallbackFcn(app,
@STOPButtonPushed, true);
            app.STOPButton.FontName = 'Lucida Sans Typewriter';
            app.STOPButton.FontSize = 30;
            app.STOPButton.Position = [895 781 113 46];
            app.STOPButton.Text = 'STOP';

            % Create TextArea_10
            app.TextArea_10 = uitextarea(app.receiverpanel);
            app.TextArea_10.FontSize = 24;
            app.TextArea_10.BackgroundColor = [0.7686 0.7686 0.549];
            app.TextArea_10.Position = [65 193 410 98];
            app.TextArea_10.Value = {'Please select one of the decoding
algortihms.'};

            % Create METHOD1Button
            app.METHOD1Button = uibutton(app.receiverpanel, 'push');
            app.METHOD1Button.ButtonPushedFcn = createCallbackFcn(app,
@METHOD1ButtonPushed, true);
            app.METHOD1Button.FontSize = 24;
            app.METHOD1Button.Position = [100 86 156 82];
            app.METHOD1Button.Text = 'METHOD 1';

            % Create METHOD2Button
            app.METHOD2Button = uibutton(app.receiverpanel, 'push');
            app.METHOD2Button.ButtonPushedFcn = createCallbackFcn(app,
@METHOD2ButtonPushed, true);
            app.METHOD2Button.FontSize = 24;
            app.METHOD2Button.Position = [300 86 156 82];
            app.METHOD2Button.Text = 'METHOD 2';

            % Create HereisyourdecodedsignalTextAreaLabel
            app.HereisyourdecodedsignalTextAreaLabel = uilabel(app.receiverpanel);
            app.HereisyourdecodedsignalTextAreaLabel.HorizontalAlignment =
'right';
            app.HereisyourdecodedsignalTextAreaLabel.FontSize = 24;
            app.HereisyourdecodedsignalTextAreaLabel.Position = [560 186 313 61];
            app.HereisyourdecodedsignalTextAreaLabel.Text = 'Here is your decoded
signal:';

            % Create HereisyourdecodedsignalTextArea
            app.HereisyourdecodedsignalTextArea = uitextarea(app.receiverpanel);
            app.HereisyourdecodedsignalTextArea.BackgroundColor = [0.7686 0.7686
0.549];
            app.HereisyourdecodedsignalTextArea.Position = [888 166 191 99];

            % Create MainMenuButton_2
            app.MainMenuButton_2 = uibutton(app.receiverpanel, 'push');
            app.MainMenuButton_2.ButtonPushedFcn = createCallbackFcn(app, 
@MainMenuButton_2Pushed, true);
            app.MainMenuButton_2.FontSize = 18;
            app.MainMenuButton_2.FontWeight = 'bold';
            app.MainMenuButton_2.Position = [921 41 125 58];
            app.MainMenuButton_2.Text = {'Main'; 'Menu'};

            % Create transmitterpanel
            app.transmitterpanel = uipanel(app.UIFigure);
            app.transmitterpanel.BorderType = 'none';
            app.transmitterpanel.BorderWidth = 0;
            app.transmitterpanel.Title = 'Panel';
            app.transmitterpanel.Visible = 'off';
            app.transmitterpanel.BackgroundColor = [0.7686 0.7686 0.549];
            app.transmitterpanel.Position = [147 45 1078 936];

            % Create UIAxes
            app.UIAxes = uiaxes(app.transmitterpanel);
            title(app.UIAxes, 'Time-Domain Signal')
            xlabel(app.UIAxes, 'Time(ms)')
            ylabel(app.UIAxes, 'Magnitude')
            zlabel(app.UIAxes, 'Z')
            app.UIAxes.HandleVisibility = 'off';
            app.UIAxes.PickableParts = 'none';
            colormap(app.UIAxes, 'jet')
            app.UIAxes.Position = [85 122 452 345];

            % Create UIAxes_2
            app.UIAxes_2 = uiaxes(app.transmitterpanel);
            title(app.UIAxes_2, 'Spectrogram')
            xlabel(app.UIAxes_2, 'Time(ms)')
            ylabel(app.UIAxes_2, 'Frequency')
            zlabel(app.UIAxes_2, 'Z')
            app.UIAxes_2.YLim = [450 1900];
            app.UIAxes_2.Position = [613 122 457 345];

            % Create Button_1
            app.Button_1 = uibutton(app.transmitterpanel, 'push');
            app.Button_1.ButtonPushedFcn = createCallbackFcn(app, @Button_1Pushed, 
true);
            app.Button_1.IconAlignment = 'center';
            app.Button_1.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_1.FontName = 'Book Antiqua';
            app.Button_1.FontSize = 44;
            app.Button_1.FontWeight = 'bold';
            app.Button_1.Position = [62 760 63 54];
            app.Button_1.Text = '1';

            % Create Button_2
            app.Button_2 = uibutton(app.transmitterpanel, 'push');
            app.Button_2.ButtonPushedFcn = createCallbackFcn(app, @Button_2Pushed, 
true);
            app.Button_2.IconAlignment = 'center';
            app.Button_2.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_2.FontName = 'Book Antiqua';
            app.Button_2.FontSize = 44;
            app.Button_2.FontWeight = 'bold';
            app.Button_2.Position = [152 760 63 54];
            app.Button_2.Text = '2';
            % Create Button_3
            app.Button_3 = uibutton(app.transmitterpanel, 'push');
            app.Button_3.ButtonPushedFcn = createCallbackFcn(app, @Button_3Pushed,
true);
            app.Button_3.IconAlignment = 'center';
            app.Button_3.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_3.FontName = 'Book Antiqua';
            app.Button_3.FontSize = 44;
            app.Button_3.FontWeight = 'bold';
            app.Button_3.Position = [241 760 63 54];
            app.Button_3.Text = '3';

            % Create Button_4
            app.Button_4 = uibutton(app.transmitterpanel, 'push');
            app.Button_4.ButtonPushedFcn = createCallbackFcn(app, @Button_4Pushed,
true);
            app.Button_4.IconAlignment = 'center';
            app.Button_4.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_4.FontName = 'Book Antiqua';
            app.Button_4.FontSize = 44;
            app.Button_4.FontWeight = 'bold';
            app.Button_4.Position = [62 684 63 54];
            app.Button_4.Text = '4';

            % Create Button_5
            app.Button_5 = uibutton(app.transmitterpanel, 'push');
            app.Button_5.ButtonPushedFcn = createCallbackFcn(app, @Button_5Pushed,
true);
            app.Button_5.IconAlignment = 'center';
            app.Button_5.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_5.FontName = 'Book Antiqua';
            app.Button_5.FontSize = 44;
            app.Button_5.FontWeight = 'bold';
            app.Button_5.Position = [152 684 63 54];
            app.Button_5.Text = '5';

            % Create Button_6
            app.Button_6 = uibutton(app.transmitterpanel, 'push');
            app.Button_6.ButtonPushedFcn = createCallbackFcn(app, @Button_6Pushed,
true);
            app.Button_6.IconAlignment = 'center';
            app.Button_6.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_6.FontName = 'Book Antiqua';
            app.Button_6.FontSize = 44;
            app.Button_6.FontWeight = 'bold';
            app.Button_6.Position = [241 687 63 54];
            app.Button_6.Text = '6';

            % Create Button_7
            app.Button_7 = uibutton(app.transmitterpanel, 'push');
            app.Button_7.ButtonPushedFcn = createCallbackFcn(app, @Button_7Pushed,
true);
            app.Button_7.IconAlignment = 'center';
            app.Button_7.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_7.FontName = 'Book Antiqua';
            app.Button_7.FontSize = 44;
            app.Button_7.FontWeight = 'bold';
            app.Button_7.Position = [62 606 63 54];
            app.Button_7.Text = '7';

            % Create Button_8
            app.Button_8 = uibutton(app.transmitterpanel, 'push');
            app.Button_8.ButtonPushedFcn = createCallbackFcn(app, @Button_8Pushed,
true);
            app.Button_8.IconAlignment = 'center';
            app.Button_8.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_8.FontName = 'Book Antiqua';
            app.Button_8.FontSize = 44;
            app.Button_8.FontWeight = 'bold';
            app.Button_8.Position = [152 606 63 54];
            app.Button_8.Text = '8';

            % Create Button_9
            app.Button_9 = uibutton(app.transmitterpanel, 'push');
            app.Button_9.ButtonPushedFcn = createCallbackFcn(app, @Button_9Pushed,
true);
            app.Button_9.IconAlignment = 'center';
            app.Button_9.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_9.FontName = 'Book Antiqua';
            app.Button_9.FontSize = 44;
            app.Button_9.FontWeight = 'bold';
            app.Button_9.Position = [241 606 63 54];
            app.Button_9.Text = '9';

            % Create Button_10
            app.Button_10 = uibutton(app.transmitterpanel, 'push');
            app.Button_10.ButtonPushedFcn = createCallbackFcn(app,
@Button_10Pushed, true);
            app.Button_10.IconAlignment = 'center';
            app.Button_10.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_10.FontName = 'Book Antiqua';
            app.Button_10.FontSize = 44;
            app.Button_10.FontWeight = 'bold';
            app.Button_10.Position = [62 524 63 54];
            app.Button_10.Text = '*';

            % Create Button_11
            app.Button_11 = uibutton(app.transmitterpanel, 'push');
            app.Button_11.ButtonPushedFcn = createCallbackFcn(app,
@Button_11Pushed, true);
            app.Button_11.IconAlignment = 'center';
            app.Button_11.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_11.FontName = 'Book Antiqua';
            app.Button_11.FontSize = 44;
            app.Button_11.FontWeight = 'bold';
            app.Button_11.Position = [152 524 63 54];
            app.Button_11.Text = '0';

            % Create Button_12
            app.Button_12 = uibutton(app.transmitterpanel, 'push');
            app.Button_12.ButtonPushedFcn = createCallbackFcn(app,
@Button_12Pushed, true);
            app.Button_12.IconAlignment = 'center';
            app.Button_12.BackgroundColor = [0.7686 0.7686 0.549];
            app.Button_12.FontName = 'Book Antiqua';
            app.Button_12.FontSize = 44;
            app.Button_12.FontWeight = 'bold';
            app.Button_12.Position = [241 524 63 54];
            app.Button_12.Text = '#';

            % Create YouhaveenteredTextAreaLabel
            app.YouhaveenteredTextAreaLabel = uilabel(app.transmitterpanel);
            app.YouhaveenteredTextAreaLabel.BackgroundColor = [0.7686 0.7686
0.549];
            app.YouhaveenteredTextAreaLabel.HorizontalAlignment = 'right';
            app.YouhaveenteredTextAreaLabel.FontSize = 18;
            app.YouhaveenteredTextAreaLabel.Position = [45 831 84 44];
            app.YouhaveenteredTextAreaLabel.Text = {'You have '; 'entered:'};

            % Create YouhaveenteredTextArea
            app.YouhaveenteredTextArea = uitextarea(app.transmitterpanel);
            app.YouhaveenteredTextArea.Editable = 'off';
            app.YouhaveenteredTextArea.FontSize = 18;
            app.YouhaveenteredTextArea.BackgroundColor = [0.7686 0.7686 0.549];
            app.YouhaveenteredTextArea.Position = [138 824 213 57];

            % Create CLEARButton
            app.CLEARButton = uibutton(app.transmitterpanel, 'push');
            app.CLEARButton.ButtonPushedFcn = createCallbackFcn(app,
@CLEARButtonPushed, true);
            app.CLEARButton.BackgroundColor = [0.7686 0.7686 0.549];
            app.CLEARButton.FontSize = 18;
            app.CLEARButton.FontWeight = 'bold';
            app.CLEARButton.Position = [183 478 89 40];
            app.CLEARButton.Text = 'CLEAR';

            % Create TextArea_5
            app.TextArea_5 = uitextarea(app.transmitterpanel);
            app.TextArea_5.Editable = 'off';
            app.TextArea_5.FontName = 'Bookman';
            app.TextArea_5.FontSize = 20;
            app.TextArea_5.BackgroundColor = [0.7686 0.7686 0.549];
            app.TextArea_5.Position = [455 783 586 68];
            app.TextArea_5.Value = {'To display DTMF decoded time domain signals,
first you should enter the signling-duration-per-key (Td) and resting duration
(Tr).'};

            % Create TdmsEditFieldLabel
            app.TdmsEditFieldLabel = uilabel(app.transmitterpanel);
            app.TdmsEditFieldLabel.HorizontalAlignment = 'right';
            app.TdmsEditFieldLabel.FontSize = 18;
            app.TdmsEditFieldLabel.Position = [563 728 62 23];
            app.TdmsEditFieldLabel.Text = 'Td(ms)';

            % Create TdmsEditField
            app.TdmsEditField = uieditfield(app.transmitterpanel, 'numeric');
            app.TdmsEditField.AllowEmpty = 'on';
            app.TdmsEditField.BackgroundColor = [0.7686 0.7686 0.549];
            app.TdmsEditField.Position = [639 721 70 38];
            app.TdmsEditField.Value = [];

            % Create TrmsEditFieldLabel
            app.TrmsEditFieldLabel = uilabel(app.transmitterpanel);
            app.TrmsEditFieldLabel.HorizontalAlignment = 'right';
            app.TrmsEditFieldLabel.FontSize = 18;
            app.TrmsEditFieldLabel.Position = [796 728 57 23];
            app.TrmsEditFieldLabel.Text = 'Tr(ms)';

            % Create TrmsEditField
            app.TrmsEditField = uieditfield(app.transmitterpanel, 'numeric');
            app.TrmsEditField.AllowEmpty = 'on';
            app.TrmsEditField.BackgroundColor = [0.7686 0.7686 0.549];
            app.TrmsEditField.Position = [867 721 70 38];

            % Create TextArea_6
            app.TextArea_6 = uitextarea(app.transmitterpanel);
            app.TextArea_6.Editable = 'off';
            app.TextArea_6.FontName = 'Bookman';
            app.TextArea_6.FontSize = 20;
            app.TextArea_6.BackgroundColor = [0.7686 0.7686 0.549];
            app.TextArea_6.Position = [435 660 425 32];
            app.TextArea_6.Value = {'You can adjust the amplitude of the
signals.→'};

            % Create AmplitudeEditFieldLabel
            app.AmplitudeEditFieldLabel = uilabel(app.transmitterpanel);
            app.AmplitudeEditFieldLabel.HorizontalAlignment = 'right';
            app.AmplitudeEditFieldLabel.FontSize = 18;
            app.AmplitudeEditFieldLabel.Position = [859 664 85 23];
            app.AmplitudeEditFieldLabel.Text = 'Amplitude';

            % Create AmplitudeEditField
            app.AmplitudeEditField = uieditfield(app.transmitterpanel, 'numeric');
            app.AmplitudeEditField.AllowEmpty = 'on';
            app.AmplitudeEditField.BackgroundColor = [0.7686 0.7686 0.549];
            app.AmplitudeEditField.Position = [958 657 70 38];
            app.AmplitudeEditField.Value = 1;

            % Create TextArea_7
            app.TextArea_7 = uitextarea(app.transmitterpanel);
            app.TextArea_7.Editable = 'off';
            app.TextArea_7.FontSize = 18;
            app.TextArea_7.BackgroundColor = [0.7686 0.7686 0.549];
            app.TextArea_7.Position = [158 19 394 49];
            app.TextArea_7.Value = {'You can save or play the generated signals
→'};

            % Create SavethesignalButton
            app.SavethesignalButton = uibutton(app.transmitterpanel, 'push');
            app.SavethesignalButton.ButtonPushedFcn = createCallbackFcn(app,
@SavethesignalButtonPushed, true);
            app.SavethesignalButton.FontSize = 18;
            app.SavethesignalButton.FontWeight = 'bold';
            app.SavethesignalButton.Position = [602 19 100 52];
            app.SavethesignalButton.Text = {'Save'; 'the signal'};

            % Create PlaythesignalButton
            app.PlaythesignalButton = uibutton(app.transmitterpanel, 'push');
            app.PlaythesignalButton.ButtonPushedFcn = createCallbackFcn(app,
@PlaythesignalButtonPushed, true);
            app.PlaythesignalButton.FontSize = 18;
            app.PlaythesignalButton.FontWeight = 'bold';
            app.PlaythesignalButton.Position = [734 20 100 52];
            app.PlaythesignalButton.Text = {'Play '; 'the signal'; ''};
            % Create PlotTimeDomainandSpectrogramoftheSignalButton
            app.PlotTimeDomainandSpectrogramoftheSignalButton =
uibutton(app.transmitterpanel, 'push');
            app.PlotTimeDomainandSpectrogramoftheSignalButton.ButtonPushedFcn =
createCallbackFcn(app, @PlotTimeDomainandSpectrogramoftheSignalButtonPushed,
true);
            app.PlotTimeDomainandSpectrogramoftheSignalButton.FontWeight = 'bold';
            app.PlotTimeDomainandSpectrogramoftheSignalButton.Position = [600 592
297 36];
            app.PlotTimeDomainandSpectrogramoftheSignalButton.Text = 'Plot Time
Domain and Spectrogram of the Signal';

            % Create MainMenuButton
            app.MainMenuButton = uibutton(app.transmitterpanel, 'push');
            app.MainMenuButton.ButtonPushedFcn = createCallbackFcn(app,
@MainMenuButtonPushed, true);
            app.MainMenuButton.FontSize = 18;
            app.MainMenuButton.FontWeight = 'bold';
            app.MainMenuButton.Position = [921 18 125 58];
            app.MainMenuButton.Text = {'Main'; 'Menu'};

            % Create AButton
            app.AButton = uibutton(app.transmitterpanel, 'push');
            app.AButton.ButtonPushedFcn = createCallbackFcn(app, @AButtonPushed,
true);
            app.AButton.IconAlignment = 'center';
            app.AButton.BackgroundColor = [0.7686 0.7686 0.549];
            app.AButton.FontName = 'Book Antiqua';
            app.AButton.FontSize = 44;
            app.AButton.FontWeight = 'bold';
            app.AButton.Position = [328 760 63 54];
            app.AButton.Text = 'A';

            % Create BButton
            app.BButton = uibutton(app.transmitterpanel, 'push');
            app.BButton.ButtonPushedFcn = createCallbackFcn(app, @BButtonPushed,
true);
            app.BButton.IconAlignment = 'center';
            app.BButton.BackgroundColor = [0.7686 0.7686 0.549];
            app.BButton.FontName = 'Book Antiqua';
            app.BButton.FontSize = 44;
            app.BButton.FontWeight = 'bold';
            app.BButton.Position = [331 687 63 54];
            app.BButton.Text = 'B';

            % Create CButton
            app.CButton = uibutton(app.transmitterpanel, 'push');
            app.CButton.ButtonPushedFcn = createCallbackFcn(app, @CButtonPushed,
true);
            app.CButton.IconAlignment = 'center';
            app.CButton.BackgroundColor = [0.7686 0.7686 0.549];
            app.CButton.FontName = 'Book Antiqua';
            app.CButton.FontSize = 44;
            app.CButton.FontWeight = 'bold';
            app.CButton.Position = [331 607 63 54];
            app.CButton.Text = 'C';

            % Create DButton
            app.DButton = uibutton(app.transmitterpanel, 'push');
            app.DButton.ButtonPushedFcn = createCallbackFcn(app, @DButtonPushed, 
true);
            app.DButton.IconAlignment = 'center';
            app.DButton.BackgroundColor = [0.7686 0.7686 0.549];
            app.DButton.FontName = 'Book Antiqua';
            app.DButton.FontSize = 44;
            app.DButton.FontWeight = 'bold';
            app.DButton.Position = [328 528 63 54];
            app.DButton.Text = 'D';

            % 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 = dspprojectpart2

            runningApp = getRunningApp(app);

            % Check for running singleton app
            if isempty(runningApp)

                % Create UIFigure and components
                createComponents(app)

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

                % Focus the running singleton app
                figure(runningApp.UIFigure)

                app = runningApp;
            end

            if nargout == 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
```