OPEN ENDED LAB REPORT

Course Code : EEE 4416

Course Name : Simulation Lab

App Name : Analog Modulation Scheme

Members:

NAME	ID
Khandaker Adeba Tabassum	200021102
Sadat Al Rashad	200021106
Md. Nazmul Aman	200021132

Description:

In this project we made a MATLAB App named 'Analog Modulation Scheme' using gui which takes input for message and carrier waveform amplitude, frequency, modulation index and sampling frequency. It shows message waveform, carrier waveform and modulated waveform as output. It can modulate the message signal using AM, FM, PM techniques and PAM, PPM, PWM techniques when carrier is pulse.

MATLAB Code:

```
classdef ModOEL_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                    matlab.ui.Figure
        ANALOGMODULATIONSCHEMESLabel  matlab.ui.control.Label
        Image                       matlab.ui.control.Image
        WarningsPanel               matlab.ui.container.Panel
        SawtoothAmplitudeErrorLabel   matlab.ui.control.Label
        Lamp_3                      matlab.ui.control.Lamp
        LowCarrierFreqLabel         matlab.ui.control.Label
        OvermodulationLabel         matlab.ui.control.Label
        Lamp_2                      matlab.ui.control.Lamp
        Lamp                        matlab.ui.control.Lamp
        ModulationtypeKnob          matlab.ui.control.DiscreteKnob
        ModulationtypeKnobLabel     matlab.ui.control.Label
        CarrierPanel                matlab.ui.container.Panel
        ModulateButton              matlab.ui.control.Button
        TextArea_6                  matlab.ui.control.TextArea
        TextArea_3                  matlab.ui.control.TextArea
        AmplitudeSlider_carrier     matlab.ui.control.Slider
        AmplitudeSlider_2Label      matlab.ui.control.Label
        CarrierFrequencyKnob        matlab.ui.control.Knob
        CarrierFrequencyKnobLabel   matlab.ui.control.Label
        CarrierWaveformDropDown     matlab.ui.control.DropDown
        CarrierWaveformDropDownLabel  matlab.ui.control.Label
        MessagePanel                matlab.ui.container.Panel
        TextArea_5                  matlab.ui.control.TextArea
        TextArea_4                  matlab.ui.control.TextArea
        TextArea_2                  matlab.ui.control.TextArea
        TextArea                    matlab.ui.control.TextArea
        ModulationIndexSlider       matlab.ui.control.Slider
        ModulationIndexSliderLabel  matlab.ui.control.Label
        MessagewaveformDropDown     matlab.ui.control.DropDown
        MessagewaveformDropDownLabel  matlab.ui.control.Label
        SamplingFrequencyKnob       matlab.ui.control.Knob
        SamplingFrequencyKnobLabel  matlab.ui.control.Label
        MessageFrequencyKnob        matlab.ui.control.Knob
        MessageFrequencyKnobLabel   matlab.ui.control.Label
        AmplitudeSlider_msg         matlab.ui.control.Slider
        AmplitudeSliderLabel        matlab.ui.control.Label
        UIAxes_mod                  matlab.ui.control.UIAxes
        UIAxes_carrier              matlab.ui.control.UIAxes
        UIAxes_msg                  matlab.ui.control.UIAxes
    end


    properties (Access = private)
        Am % Description
        fm
        fs
        fc
        Ac
        k
    end


    % Callbacks that handle component events
```

```
methods (Access = private)

    % Value changed function: CarrierWaveformDropDown
    function CarrierWaveformDropDownValueChanged(app, event)
        if strcmp(app.CarrierWaveformDropDown.Value,{'Pulse'})
            app.ModulationtypeKnob.Items = {'PAM', 'PPM', 'PWM'};
        else
            app.ModulationtypeKnob.Items = {'AM', 'FM', 'PM'};
        end
    end

    % Value changed function: AmplitudeSlider_msg
    function AmplitudeSlider_msgValueChanged(app, event)
        app.Am = app.AmplitudeSlider_msg.Value;

    end

    % Value changed function: MessageFrequencyKnob
    function MessageFrequencyKnobValueChanged(app, event)
        app.fm = app.MessageFrequencyKnob.Value;
        app.TextArea.Value=num2str(app.fm);
    end

    % Value changed function: SamplingFrequencyKnob
    function SamplingFrequencyKnobValueChanged(app, event)
        app.fs = app.SamplingFrequencyKnob.Value;
        app.TextArea_2.Value=num2str(app.fs);
    end

    % Value changed function: AmplitudeSlider_carrier
    function AmplitudeSlider_carrierValueChanged(app, event)
        app.Ac = app.AmplitudeSlider_carrier.Value;

    end

    % Value changed function: CarrierFrequencyKnob
    function CarrierFrequencyKnobValueChanged(app, event)
        app.fc = app.CarrierFrequencyKnob.Value;
        app.TextArea_3.Value=num2str(app.fc);
    end

    % Value changed function: ModulationIndexSlider
    function ModulationIndexSliderValueChanged(app, event)
        app.k = app.ModulationIndexSlider.Value;

    end

    % Button pushed function: ModulateButton
    function ModulateButtonPushed(app, event)
        t=0:1/app.fs:1;
        str=app.MessagewaveformDropDown.Value;
        switch str
            case {'Sinusoidal'}
                m= app.Am*sin(2*pi*app.fm*t);
            case {'Square'}
                m= app.Am*square(2*pi*app.fm*t,50);
            case {'Triangular'}
                m= app.Am*sawtooth(2*pi*app.fm*t,0.50);
        end
        plot(app.UIAxes_msg,t,m,'r');
        str2=app.CarrierWaveformDropDown.Value;
        switch str2
            case {'Sinusoidal'}
                c= app.Ac*sin(2*pi*app.fc*t);
            case {'Square'}
                c= app.Ac*square(2*pi*app.fc*t,50);
            case {'Pulse'}
                c=square(2*pi*app.fc*t,50);
                c(c<0)=0;

        end
        plot(app.UIAxes_carrier,t,c,'g');
        str3=app.ModulationtypeKnob.Value;
        switch str3
            case {'AM'}
                x = (1 + app.k * m) .* c;
                if app.k*(app.Am)>1
                    app.Lamp.Color='r';
                else
                    app.Lamp.Color='g';
                end
            case {'FM'}
                m_int = (1/app.fs)*cumsum(m);
```

```
            x = app.Ac*cos(2*pi*app.fc*t + 2*pi*app.k*m_int);
        case {'PM'}
            x=app.Ac*cos(2*pi*app.fc*t+ app.k*m);
        case {'PAM'}
            x=m.*c;
        case {'PWM'}
            s=app.Ac*sawtooth(2*pi*10*app.fm*t+pi);
            n=length(s);
            x=zeros(1,n);
            for i=1:n
                if (m(i)>=s(i))
                    x(i)=1;
                end
            end
            if app.Ac<app.Am
                app.Lamp_3.Color='r';
            else
                app.Lamp_3.Color='g';
            end
        case {'PPM'}
            s=app.Ac*sawtooth(2*pi*10*app.fm*t+pi);
            n=length(s);
            pwm=zeros(1,n);
            for i=1:n
                if (m(i)>=s(i))
                    pwm(i)=1;
                end
            end
            b=diff(pwm);
            x=zeros(size(pwm));
            z=find(b==-1);
            a=z(1:end-1);
            b=z(2:end);
            p=floor(0.5*min(b-a));
            for i=1:size(z,2)
                x(z(i):(z(i)+p))=ones(1,p+1);
            end
            x=x(1:size(t,2));
            if app.Ac<app.Am
                app.Lamp_3.Color='r';
            else
                app.Lamp_3.Color='g';
            end
    end
    plot(app.UIAxes_mod,t,x,'b');
    if app.fc<app.fm
        app.Lamp_2.Color='r';
    else
        app.Lamp_2.Color='g';
    end
end

% Value changed function: ModulationtypeKnob
function ModulationtypeKnobValueChanged(app, event)
    value = app.ModulationtypeKnob.Value;
    if strcmp(value, {'FM'})
        app.ModulationIndexSlider.Limits=[0 500];
        app.CarrierFrequencyKnob.Limits=[0 500];
    elseif strcmp(value, {'PM'})
        app.ModulationIndexSlider.Limits=[0 500];
        app.CarrierFrequencyKnob.Limits=[0 500];
    else
        app.ModulationIndexSlider.Limits=[0 2];
        app.CarrierFrequencyKnob.Limits=[0 50];
    end
end

% Value changing function: ModulationIndexSlider
function ModulationIndexSliderValueChanging(app, event)
    app.k = event.Value;
    app.TextArea_4.Value=num2str(app.k);

end

% Value changing function: AmplitudeSlider_msg
function AmplitudeSlider_msgValueChanging(app, event)
    app.Am = event.Value;
    app.TextArea_5.Value=num2str(app.Am);
end

% Value changing function: AmplitudeSlider_carrier
function AmplitudeSlider_carrierValueChanging(app, event)
```

```
            app.Ac = event.Value;
            app.TextArea_6.Value=num2str(app.Ac);
        end

        % Value changing function: MessageFrequencyKnob
        function MessageFrequencyKnobValueChanging(app, event)
            app.fm = event.Value;
            app.TextArea.Value=num2str(app.fm);
        end

        % Value changing function: SamplingFrequencyKnob
        function SamplingFrequencyKnobValueChanging(app, event)
            app.fs= event.Value;
            app.TextArea_2.Value=num2str(app.fs);
        end

        % Value changing function: CarrierFrequencyKnob
        function CarrierFrequencyKnobValueChanging(app, event)
            app.fc = event.Value;
            app.TextArea_3.Value=num2str(app.fc);
        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 = [0.8392 0.8471 0.9569];
            app.UIFigure.Position = [100 100 939 848];
            app.UIFigure.Name = 'MATLAB App';

            % Create UIAxes_msg
            app.UIAxes_msg = uiaxes(app.UIFigure);
            title(app.UIAxes_msg, {'Message waveform'; ''})
            xlabel(app.UIAxes_msg, 'X')
            ylabel(app.UIAxes_msg, 'Y')
            app.UIAxes_msg.FontName = 'Comic Sans MS';
            app.UIAxes_msg.FontWeight = 'bold';
            app.UIAxes_msg.XGrid = 'on';
            app.UIAxes_msg.YGrid = 'on';
            app.UIAxes_msg.FontSize = 12;
            app.UIAxes_msg.Position = [323 377 298 183];

            % Create UIAxes_carrier
            app.UIAxes_carrier = uiaxes(app.UIFigure);
            title(app.UIAxes_carrier, {'Carrier waveform'; ''})
            xlabel(app.UIAxes_carrier, 'X')
            ylabel(app.UIAxes_carrier, 'Y')
            app.UIAxes_carrier.FontName = 'Comic Sans MS';
            app.UIAxes_carrier.FontWeight = 'bold';
            app.UIAxes_carrier.XGrid = 'on';
            app.UIAxes_carrier.YGrid = 'on';
            app.UIAxes_carrier.Position = [323 195 298 183];

            % Create UIAxes_mod
            app.UIAxes_mod = uiaxes(app.UIFigure);
            title(app.UIAxes_mod, {'Modulated signal waveform'; ''})
            xlabel(app.UIAxes_mod, 'X')
            ylabel(app.UIAxes_mod, 'Y')
            app.UIAxes_mod.FontName = 'Comic Sans MS';
            app.UIAxes_mod.FontWeight = 'bold';
            app.UIAxes_mod.XGrid = 'on';
            app.UIAxes_mod.YGrid = 'on';
            app.UIAxes_mod.Position = [323 13 298 183];

            % Create MessagePanel
            app.MessagePanel = uipanel(app.UIFigure);
            app.MessagePanel.TitlePosition = 'centertop';
            app.MessagePanel.Title = 'Message';
            app.MessagePanel.BackgroundColor = [0.9216 0.9294 1];
            app.MessagePanel.Position = [27 243 287 548];

            % Create AmplitudeSliderLabel
            app.AmplitudeSliderLabel = uilabel(app.MessagePanel);
            app.AmplitudeSliderLabel.HorizontalAlignment = 'right';
            app.AmplitudeSliderLabel.Position = [23 443 59 22];
            app.AmplitudeSliderLabel.Text = {'Amplitude'; ''};

            % Create AmplitudeSlider_msg
```

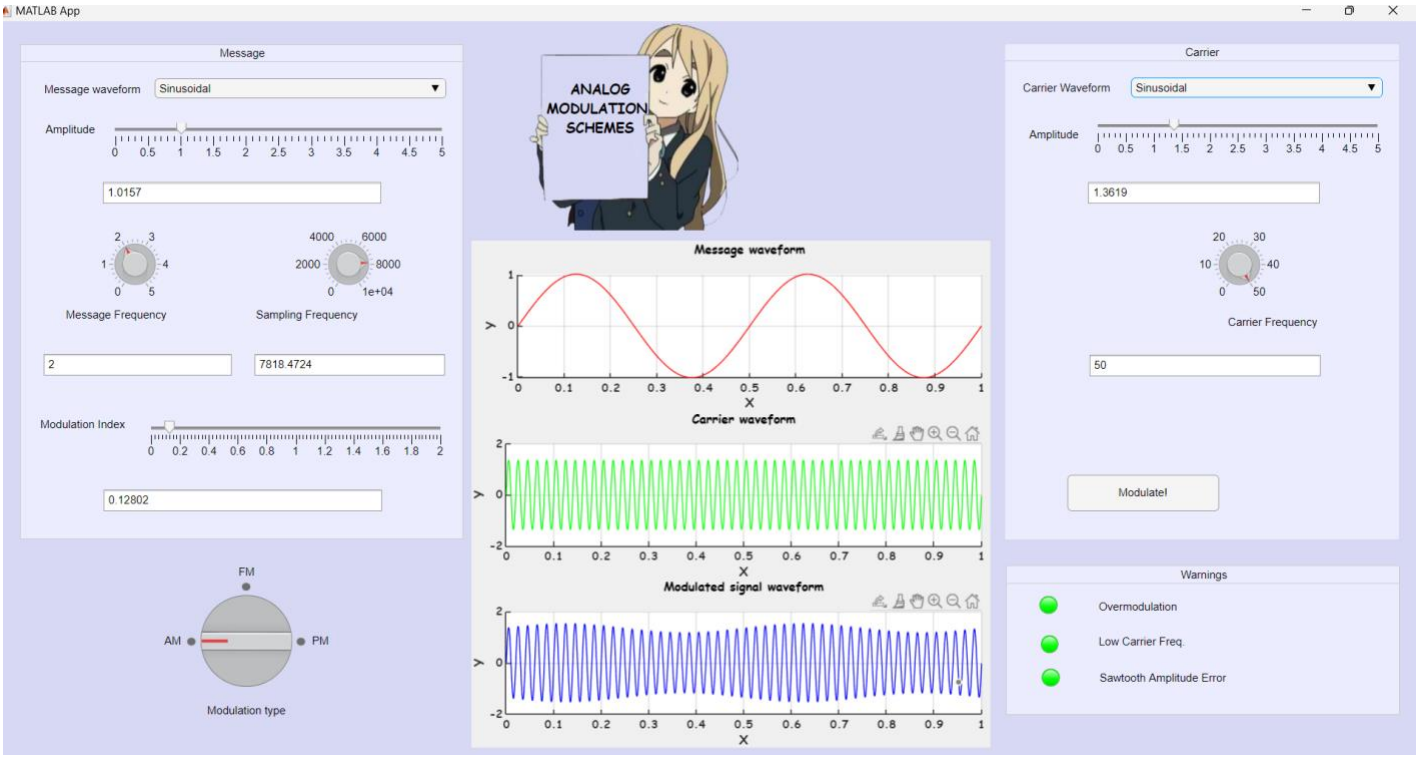
```
            app.AmplitudeSlider_msg = uislider(app.MessagePanel);
            app.AmplitudeSlider_msg.Limits = [0 5];
            app.AmplitudeSlider_msg.ValueChangedFcn = createCallbackFcn(app, @AmplitudeSlider_msgValueChanged,
true);
            app.AmplitudeSlider_msg.ValueChangingFcn = createCallbackFcn(app,
@AmplitudeSlider_msgValueChanging, true);
            app.AmplitudeSlider_msg.Position = [103 452 161 3];

            % Create MessageFrequencyKnobLabel
            app.MessageFrequencyKnobLabel = uilabel(app.MessagePanel);
            app.MessageFrequencyKnobLabel.HorizontalAlignment = 'center';
            app.MessageFrequencyKnobLabel.Position = [24 240 114 22];
            app.MessageFrequencyKnobLabel.Text = {'Message Frequency'; ''};

            % Create MessageFrequencyKnob
            app.MessageFrequencyKnob = uiknob(app.MessagePanel, 'continuous');
            app.MessageFrequencyKnob.Limits = [0 5];
            app.MessageFrequencyKnob.ValueChangedFcn = createCallbackFcn(app,
@MessageFrequencyKnobValueChanged, true);
            app.MessageFrequencyKnob.ValueChangingFcn = createCallbackFcn(app,
@MessageFrequencyKnobValueChanging, true);
            app.MessageFrequencyKnob.Position = [58 283 44 44];

            % Create SamplingFrequencyKnobLabel
            app.SamplingFrequencyKnobLabel = uilabel(app.MessagePanel);
            app.SamplingFrequencyKnobLabel.HorizontalAlignment = 'center';
            app.SamplingFrequencyKnobLabel.Position = [158 240 115 22];
            app.SamplingFrequencyKnobLabel.Text = {'Sampling Frequency'; ''};

            % Create SamplingFrequencyKnob
            app.SamplingFrequencyKnob = uiknob(app.MessagePanel, 'continuous');
            app.SamplingFrequencyKnob.Limits = [0 10000];
            app.SamplingFrequencyKnob.ValueChangedFcn = createCallbackFcn(app,
@SamplingFrequencyKnobValueChanged, true);
            app.SamplingFrequencyKnob.ValueChangingFcn = createCallbackFcn(app,
@SamplingFrequencyKnobValueChanging, true);
            app.SamplingFrequencyKnob.Position = [192 283 44 44];

            % Create MessagewaveformDropDownLabel
            app.MessagewaveformDropDownLabel = uilabel(app.MessagePanel);
            app.MessagewaveformDropDownLabel.HorizontalAlignment = 'right';
            app.MessagewaveformDropDownLabel.Position = [21 488 110 22];
            app.MessagewaveformDropDownLabel.Text = {'Message waveform'; ''};

            % Create MessagewaveformDropDown
            app.MessagewaveformDropDown = uidropdown(app.MessagePanel);
            app.MessagewaveformDropDown.Items = {'Sinusoidal', 'Square', 'Triangular', ''};
            app.MessagewaveformDropDown.Position = [146 488 123 22];
            app.MessagewaveformDropDown.Value = 'Sinusoidal';

            % Create ModulationIndexSliderLabel
            app.ModulationIndexSliderLabel = uilabel(app.MessagePanel);
            app.ModulationIndexSliderLabel.HorizontalAlignment = 'right';
            app.ModulationIndexSliderLabel.Position = [17 108 97 27];
            app.ModulationIndexSliderLabel.Text = {'Modulation Index'; ''; ''};

            % Create ModulationIndexSlider
            app.ModulationIndexSlider = uislider(app.MessagePanel);
            app.ModulationIndexSlider.Limits = [0 2];
            app.ModulationIndexSlider.ValueChangedFcn = createCallbackFcn(app,
@ModulationIndexSliderValueChanged, true);
            app.ModulationIndexSlider.ValueChangingFcn = createCallbackFcn(app,
@ModulationIndexSliderValueChanging, true);
            app.ModulationIndexSlider.Position = [142 122 120 3];

            % Create TextArea
            app.TextArea = uitextarea(app.MessagePanel);
            app.TextArea.Position = [26 183 108 23];

            % Create TextArea_2
            app.TextArea_2 = uitextarea(app.MessagePanel);
            app.TextArea_2.Position = [159 183 108 23];

            % Create TextArea_4
            app.TextArea_4 = uitextarea(app.MessagePanel);
            app.TextArea_4.Position = [91 32 108 23];

            % Create TextArea_5
            app.TextArea_5 = uitextarea(app.MessagePanel);
            app.TextArea_5.Position = [90 372 108 23];

            % Create CarrierPanel
            app.CarrierPanel = uipanel(app.UIFigure);
```

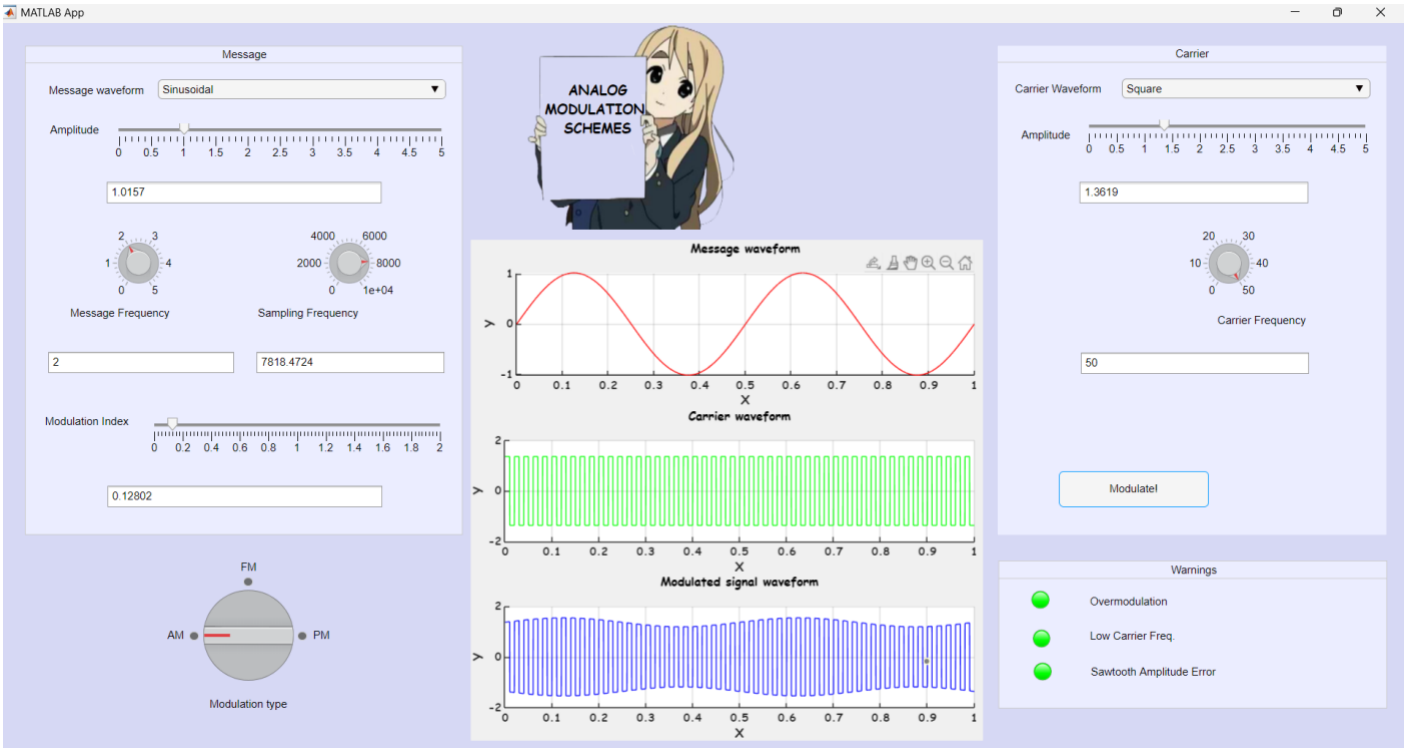
```
            app.CarrierPanel.BorderType = 'none';
            app.CarrierPanel.TitlePosition = 'centertop';
            app.CarrierPanel.Title = 'Carrier';
            app.CarrierPanel.BackgroundColor = [0.9216 0.9294 1];
            app.CarrierPanel.Position = [637 243 284 548];

            % Create CarrierWaveformDropDownLabel
            app.CarrierWaveformDropDownLabel = uilabel(app.CarrierPanel);
            app.CarrierWaveformDropDownLabel.HorizontalAlignment = 'right';
            app.CarrierWaveformDropDownLabel.Position = [12 490 103 22];
            app.CarrierWaveformDropDownLabel.Text = 'Carrier Waveform';

            % Create CarrierWaveformDropDown
            app.CarrierWaveformDropDown = uidropdown(app.CarrierPanel);
            app.CarrierWaveformDropDown.Items = {'Sinusoidal', 'Square', 'Pulse'};
            app.CarrierWaveformDropDown.ValueChangedFcn = createCallbackFcn(app,
@CarrierWaveformDropDownValueChanged, true);
            app.CarrierWaveformDropDown.Position = [137 490 130 22];
            app.CarrierWaveformDropDown.Value = 'Sinusoidal';

            % Create CarrierFrequencyKnobLabel
            app.CarrierFrequencyKnobLabel = uilabel(app.CarrierPanel);
            app.CarrierFrequencyKnobLabel.HorizontalAlignment = 'center';
            app.CarrierFrequencyKnobLabel.Position = [97 223 103 27];
            app.CarrierFrequencyKnobLabel.Text = {'Carrier Frequency'; ''; ''};

            % Create CarrierFrequencyKnob
            app.CarrierFrequencyKnob = uiknob(app.CarrierPanel, 'continuous');
            app.CarrierFrequencyKnob.Limits = [0 50];
            app.CarrierFrequencyKnob.ValueChangedFcn = createCallbackFcn(app,
@CarrierFrequencyKnobValueChanged, true);
            app.CarrierFrequencyKnob.ValueChangingFcn = createCallbackFcn(app,
@CarrierFrequencyKnobValueChanging, true);
            app.CarrierFrequencyKnob.Position = [126 284 44 44];

            % Create AmplitudeSlider_2Label
            app.AmplitudeSlider_2Label = uilabel(app.CarrierPanel);
            app.AmplitudeSlider_2Label.HorizontalAlignment = 'right';
            app.AmplitudeSlider_2Label.Position = [22 439 59 22];
            app.AmplitudeSlider_2Label.Text = {'Amplitude'; ''};

            % Create AmplitudeSlider_carrier
            app.AmplitudeSlider_carrier = uislider(app.CarrierPanel);
            app.AmplitudeSlider_carrier.Limits = [0 5];
            app.AmplitudeSlider_carrier.ValueChangedFcn = createCallbackFcn(app,
@AmplitudeSlider_carrierValueChanged, true);
            app.AmplitudeSlider_carrier.ValueChangingFcn = createCallbackFcn(app,
@AmplitudeSlider_carrierValueChanging, true);
            app.AmplitudeSlider_carrier.Position = [101 457 161 3];

            % Create TextArea_3
            app.TextArea_3 = uitextarea(app.CarrierPanel);
            app.TextArea_3.Position = [92 184 108 23];

            % Create TextArea_6
            app.TextArea_6 = uitextarea(app.CarrierPanel);
            app.TextArea_6.Position = [91 373 108 23];

            % Create ModulateButton
            app.ModulateButton = uibutton(app.CarrierPanel, 'push');
            app.ModulateButton.ButtonPushedFcn = createCallbackFcn(app, @ModulateButtonPushed, true);
            app.ModulateButton.Position = [68 33 164 39];
            app.ModulateButton.Text = 'Modulate!';

            % Create ModulationtypeKnobLabel
            app.ModulationtypeKnobLabel = uilabel(app.UIFigure);
            app.ModulationtypeKnobLabel.HorizontalAlignment = 'center';
            app.ModulationtypeKnobLabel.Position = [131 43 90 22];
            app.ModulationtypeKnobLabel.Text = {'Modulation type'; ''};

            % Create ModulationtypeKnob
            app.ModulationtypeKnob = uiknob(app.UIFigure, 'discrete');
            app.ModulationtypeKnob.Items = {'AM', 'FM', 'PM'};
            app.ModulationtypeKnob.ValueChangedFcn = createCallbackFcn(app, @ModulationtypeKnobValueChanged,
true);
            app.ModulationtypeKnob.Position = [126 80 100 100];
            app.ModulationtypeKnob.Value = 'AM';

            % Create WarningsPanel
            app.WarningsPanel = uipanel(app.UIFigure);
            app.WarningsPanel.TitlePosition = 'centertop';
            app.WarningsPanel.Title = 'Warnings';
            app.WarningsPanel.BackgroundColor = [0.9216 0.9294 1];
```

```
            app.WarningsPanel.Position = [638 48 284 165];

            % Create Lamp
            app.Lamp = uilamp(app.WarningsPanel);
            app.Lamp.Position = [36 112 20 20];

            % Create Lamp_2
            app.Lamp_2 = uilamp(app.WarningsPanel);
            app.Lamp_2.Position = [37 69 20 20];

            % Create OvermodulationLabel
            app.OvermodulationLabel = uilabel(app.WarningsPanel);
            app.OvermodulationLabel.Position = [100 110 139 22];
            app.OvermodulationLabel.Text = 'Overmodulation';

            % Create LowCarrierFreqLabel
            app.LowCarrierFreqLabel = uilabel(app.WarningsPanel);
            app.LowCarrierFreqLabel.Position = [100 71 139 22];
            app.LowCarrierFreqLabel.Text = 'Low Carrier Freq.';

            % Create Lamp_3
            app.Lamp_3 = uilamp(app.WarningsPanel);
            app.Lamp_3.Position = [38 32 20 20];

            % Create SawtoothAmplitudeErrorLabel
            app.SawtoothAmplitudeErrorLabel = uilabel(app.WarningsPanel);
            app.SawtoothAmplitudeErrorLabel.Position = [101 31 142 22];
            app.SawtoothAmplitudeErrorLabel.Text = 'Sawtooth Amplitude Error';

            % Create Image
            app.Image = uiimage(app.UIFigure);
            app.Image.Position = [379 570 232 236];
            app.Image.ImageSource = 'PikPng.com_anime-transparent-png_1850423.png';

            % Create ANALOGMODULATIONSCHEMESLabel
            app.ANALOGMODULATIONSCHEMESLabel = uilabel(app.UIFigure);
            app.ANALOGMODULATIONSCHEMESLabel.HorizontalAlignment = 'center';
            app.ANALOGMODULATIONSCHEMESLabel.FontName = 'Comic Sans MS';
            app.ANALOGMODULATIONSCHEMESLabel.FontSize = 15;
            app.ANALOGMODULATIONSCHEMESLabel.FontWeight = 'bold';
            app.ANALOGMODULATIONSCHEMESLabel.Position = [369 669 186 96];
            app.ANALOGMODULATIONSCHEMESLabel.Text = {'ANALOG'; 'MODULATION '; 'SCHEMES'; ''};

            % 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 = ModOEL_exported

            % Create UIFigure and components
            createComponents(app)

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

            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
```

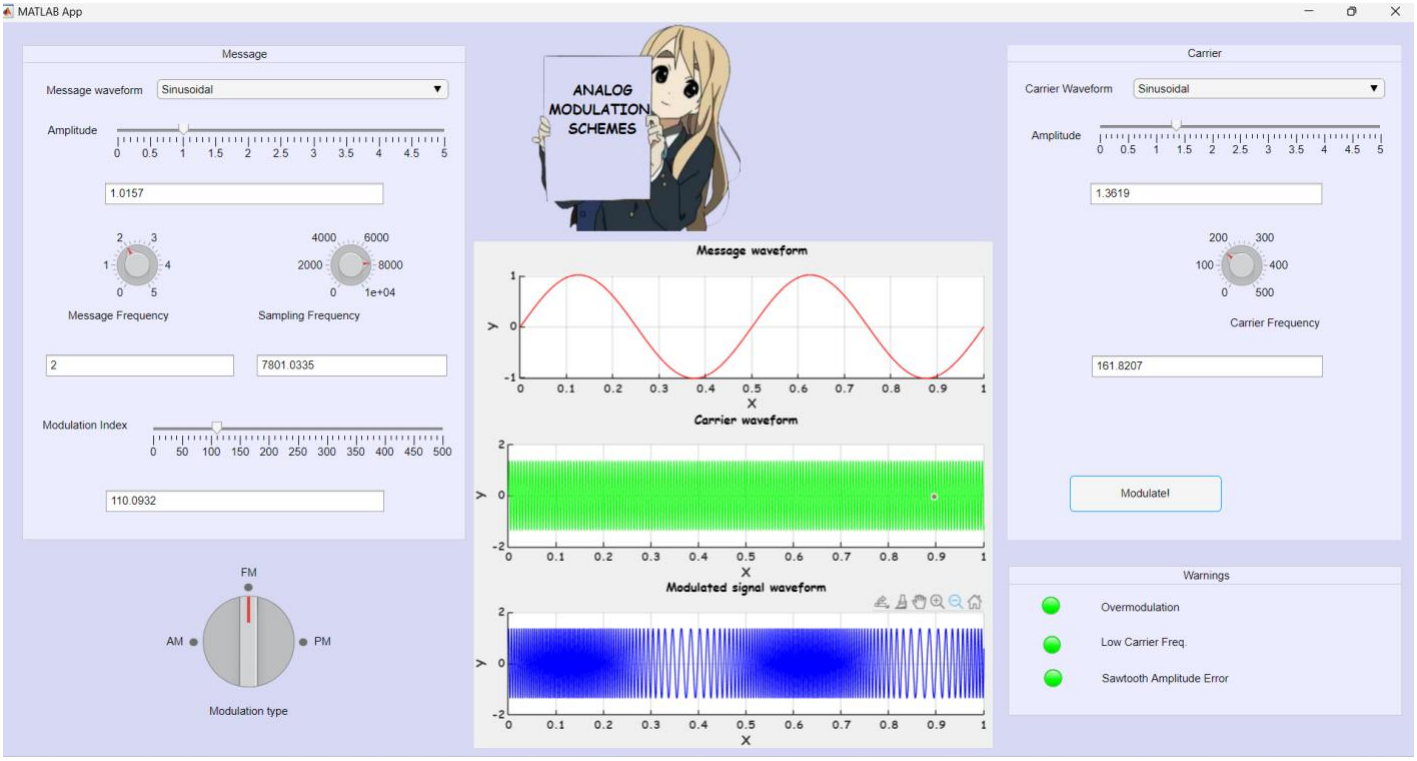
Output:

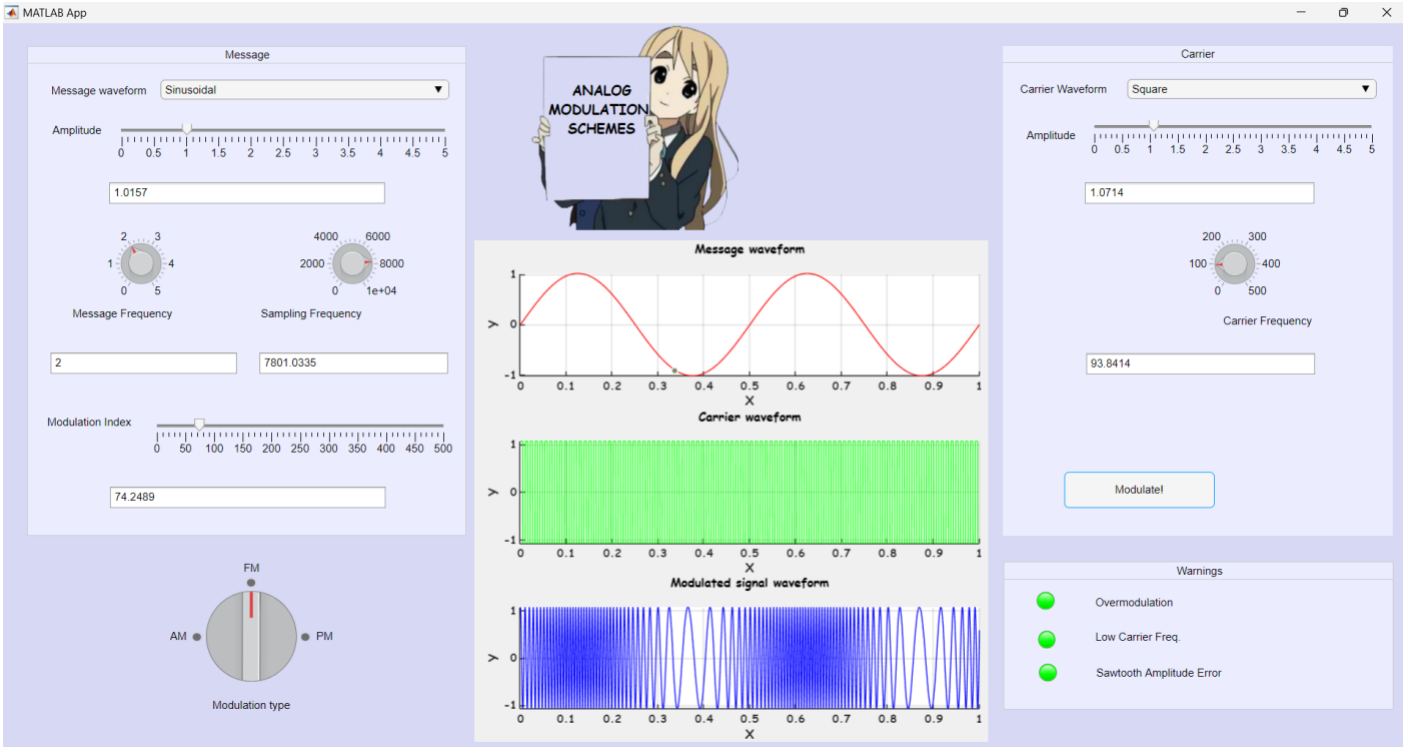
AM:


MATLAB App
Message
Message waveform
Sinusoidal
Amplitude
1.0157
Message Frequency
Sampling Frequency
2
7818.4724
Modulation Index
0.12802
FM
AM
PM
Modulation type
ANALOG MODULATION SCHEMES
Message waveform
Carrier waveform
Modulated signal waveform
Carrier
Carrier Waveform
Sinusoidal
Amplitude
1.3619
Carrier Frequency
50
Modulate!
Warnings
Overmodulation
Low Carrier Freq.
Sawtooth Amplitude Error

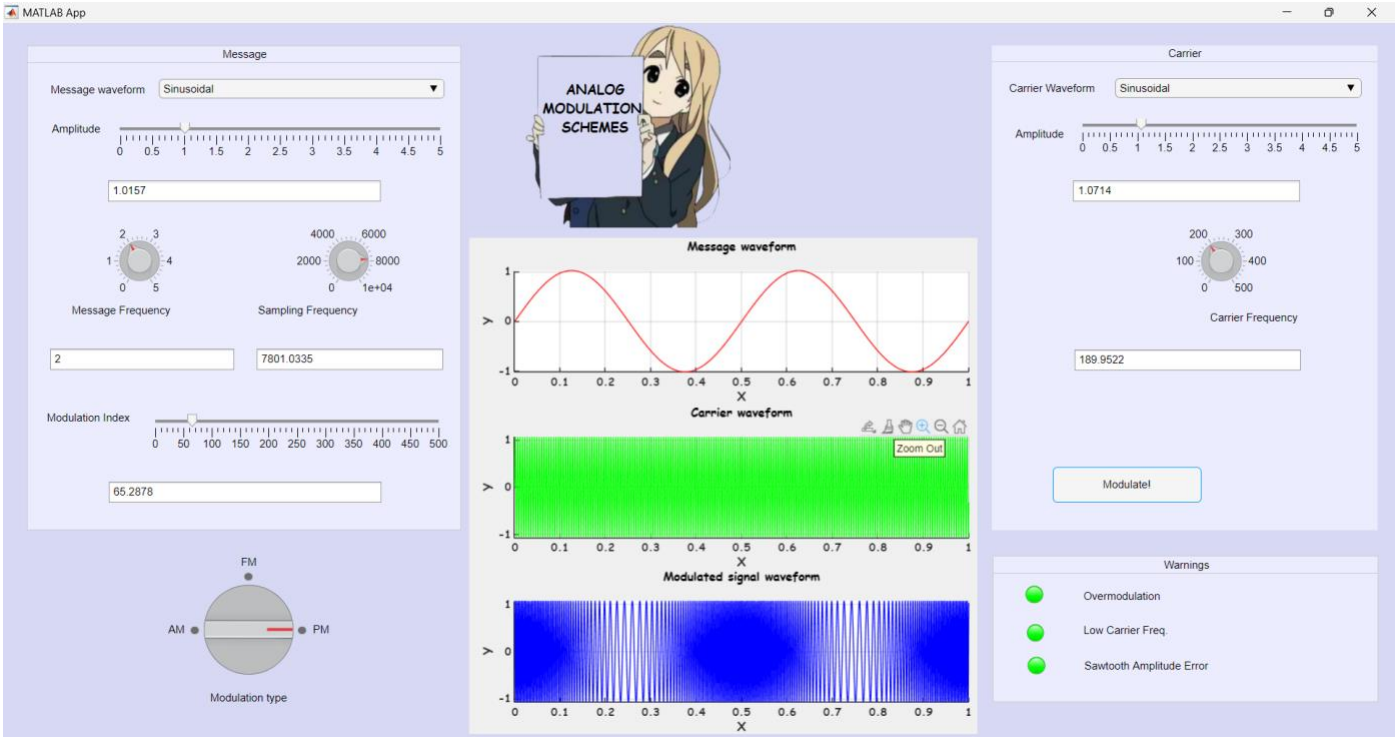

MATLAB App
Message
Message waveform
Sinusoidal
Amplitude
1.0157
Message Frequency
Sampling Frequency
2
7818.4724
Modulation Index
0.12802
FM
AM
PM
Modulation type
ANALOG MODULATION SCHEMES
Message waveform
Carrier waveform
Modulated signal waveform
Carrier
Carrier Waveform
Square
Amplitude
1.3619
Carrier Frequency
50
Modulate!
Warnings
Overmodulation
Low Carrier Freq.
Sawtooth Amplitude Error

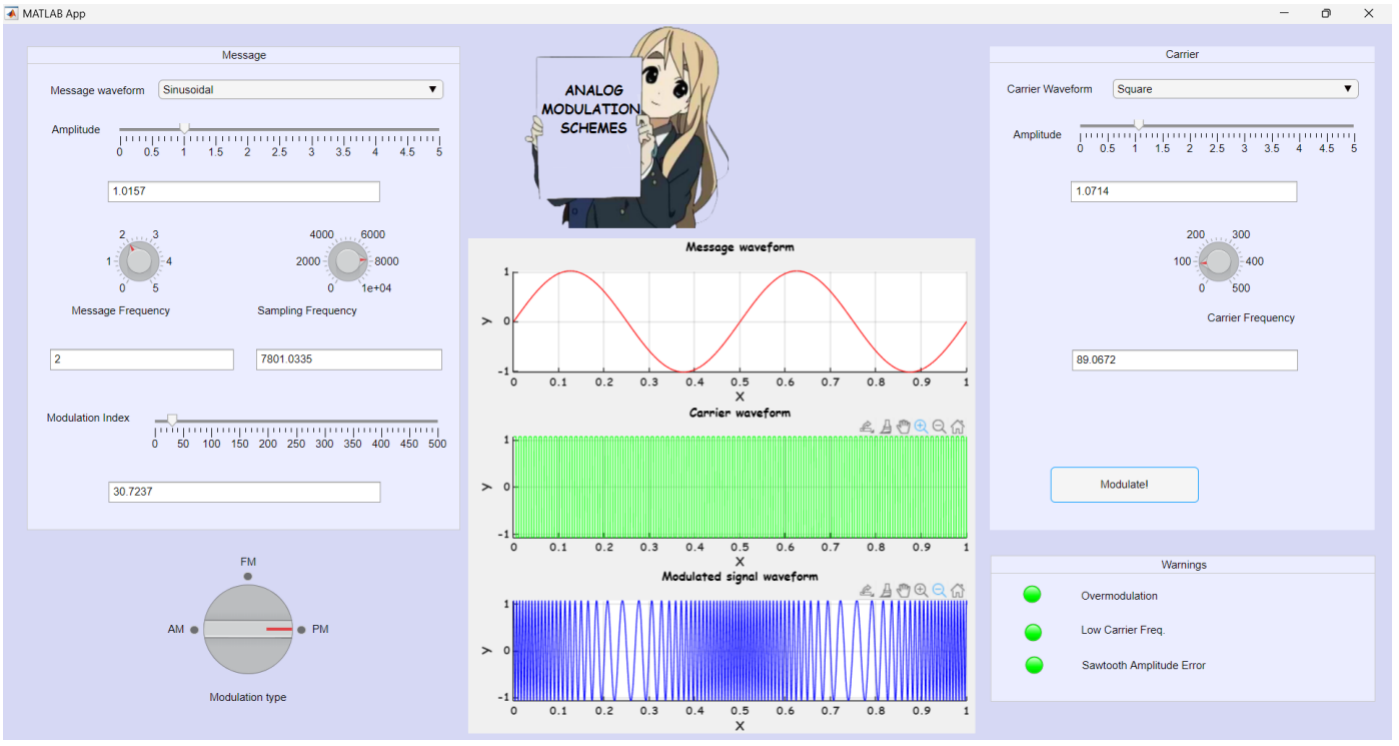
FM:


MATLAB App
Message
Message waveform
Sinusoidal
Amplitude
1.0157
Message Frequency
Sampling Frequency
2
7801.0335
Modulation Index
110.0932
FM
AM
PM
Modulation type
ANALOG MODULATION SCHEMES
Message waveform
Carrier waveform
Modulated signal waveform
Carrier
Carrier Waveform
Sinusoidal
Amplitude
1.3619
Carrier Frequency
161.8207
Modulate!
Warnings
Overmodulation
Low Carrier Freq.
Sawtooth Amplitude Error

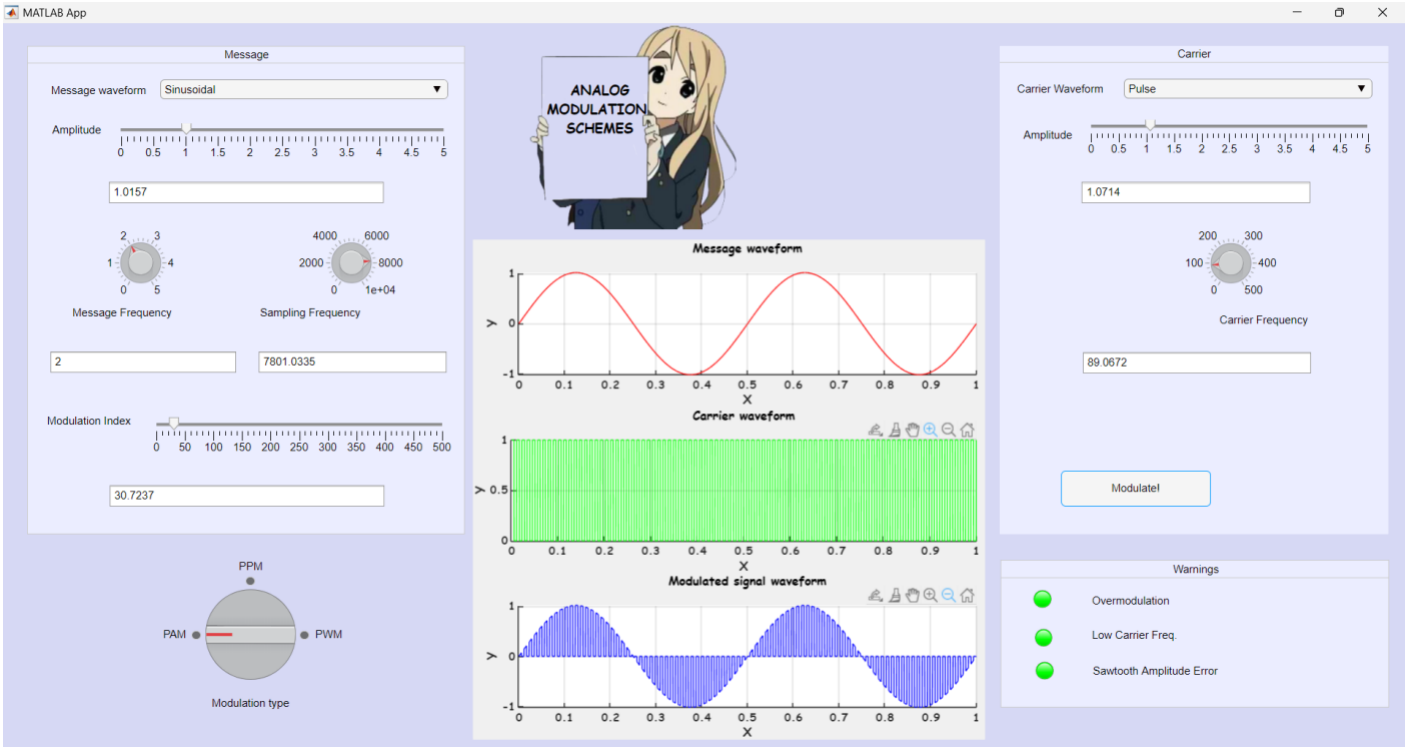

MATLAB App
Message
Message waveform
Sinusoidal
Amplitude
1.0157
Message Frequency
Sampling Frequency
2
7801.0335
Modulation Index
74.2489
FM
AM
PM
Modulation type
ANALOG MODULATION SCHEMES
Message waveform
Carrier waveform
Modulated signal waveform
Carrier
Carrier Waveform
Square
Amplitude
1.0714
Carrier Frequency
93.8414
Modulate!
Warnings
Overmodulation
Low Carrier Freq.
Sawtooth Amplitude Error

PM:

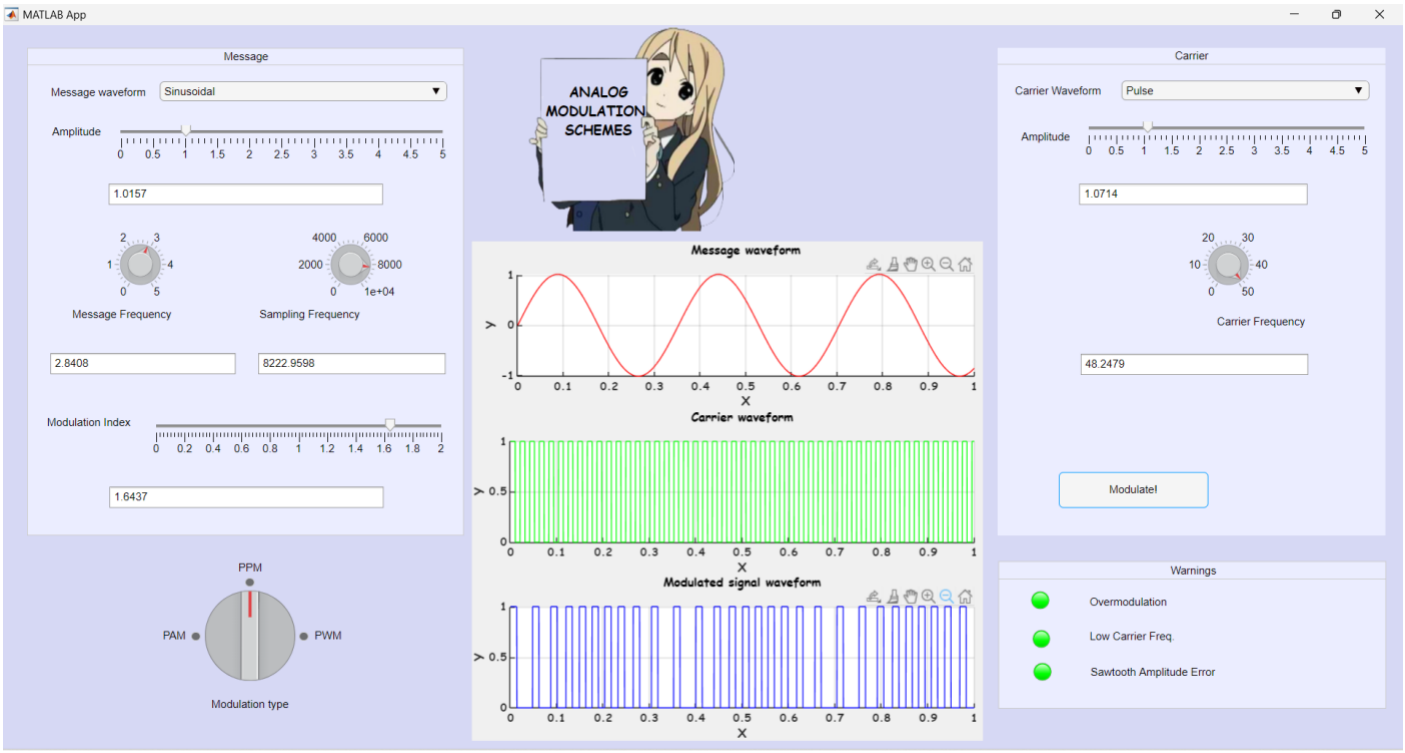

MATLAB App
Message
Message waveform
Sinusoidal
Amplitude
1.0157
Message Frequency
Sampling Frequency
2
7801.0335
Modulation Index
65.2878
FM
AM
PM
Modulation type
ANALOG MODULATION SCHEMES
Message waveform
Carrier waveform
Zoom Out
Modulated signal waveform
Carrier
Carrier Waveform
Sinusoidal
Amplitude
1.0714
Carrier Frequency
189.9522
Modulate!
Warnings
Overmodulation
Low Carrier Freq.
Sawtooth Amplitude Error


MATLAB App
Message
Message waveform
Sinusoidal
Amplitude
1.0157
Message Frequency
Sampling Frequency
2
7801.0335
Modulation Index
30.7237
FM
AM
PM
Modulation type
ANALOG MODULATION SCHEMES
Message waveform
Carrier waveform
Modulated signal waveform
Carrier
Carrier Waveform
Square
Amplitude
1.0714
Carrier Frequency
89.0672
Modulate!
Warnings
Overmodulation
Low Carrier Freq.
Sawtooth Amplitude Error

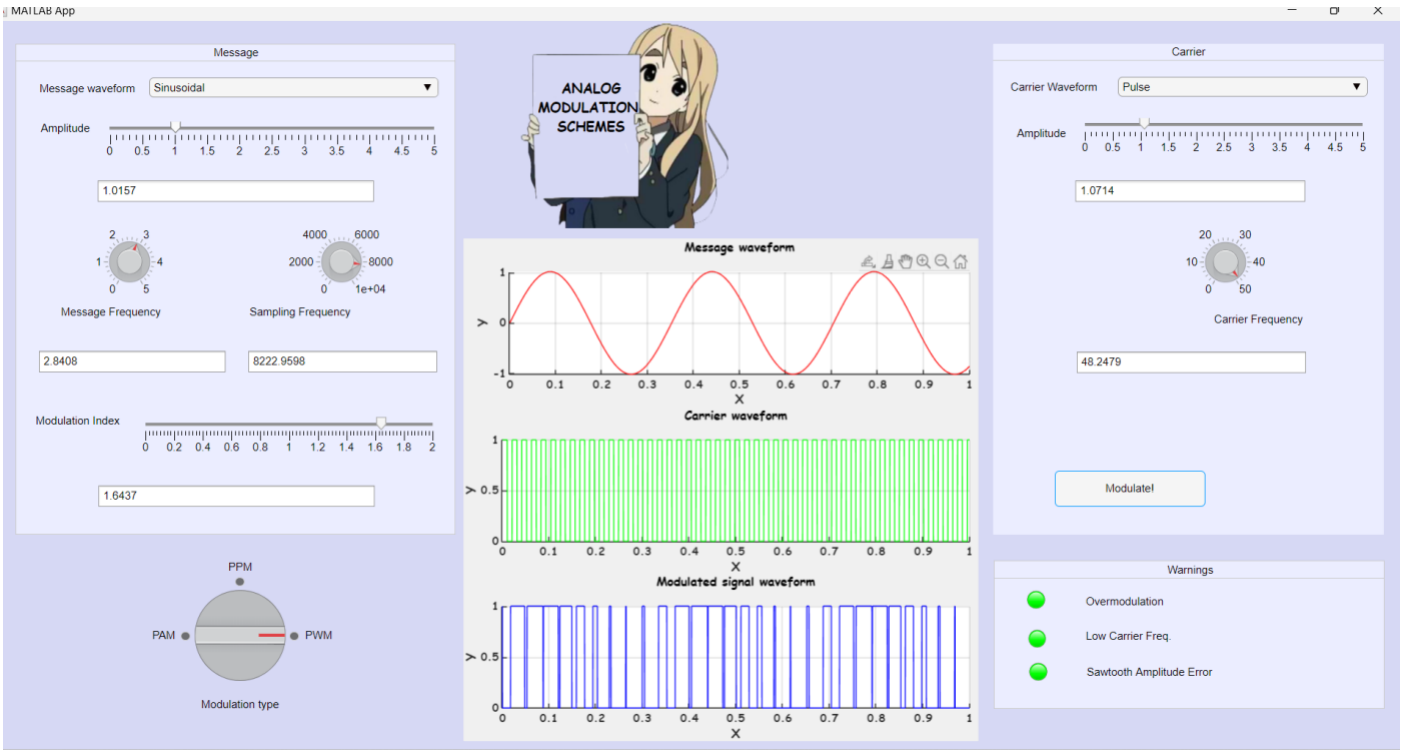
PAM:


MATLAB App
Message
Message waveform
Sinusoidal
Amplitude
1.0157
Message Frequency
Sampling Frequency
2
7801.0335
Modulation Index
30.7237
PPM
PAM
PWM
Modulation type
ANALOG MODULATION SCHEMES
Message waveform
Carrier waveform
Modulated signal waveform
Carrier
Carrier Waveform
Pulse
Amplitude
1.0714
Carrier Frequency
89.0672
Modulate!
Warnings
Overmodulation
Low Carrier Freq.
Sawtooth Amplitude Error

PPM:

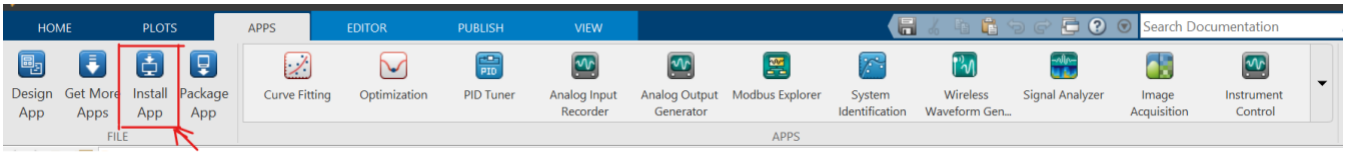

MATLAB App
Message
Message waveform
Sinusoidal
Amplitude
1.0157
Message Frequency
Sampling Frequency
2.8408
8222.9598
Modulation Index
1.6437
PPM
PAM
PWM
Modulation type
ANALOG MODULATION SCHEMES
Message waveform
Carrier waveform
Modulated signal waveform
Carrier
Carrier Waveform
Pulse
Amplitude
1.0714
Carrier Frequency
48.2479
Modulate!
Warnings
Overmodulation
Low Carrier Freq.
Sawtooth Amplitude Error

PWM:



Opening the GUI:

As we have designed this UI in Appdesigner, .fig file could not be created which is for Guide. We have provided the .mlappinstall file as a replacement of .fig file. To open it at first install the app from “App” tab in MATLAB:



Then choose the file and install it. Then open it from the app list.

Additional features:

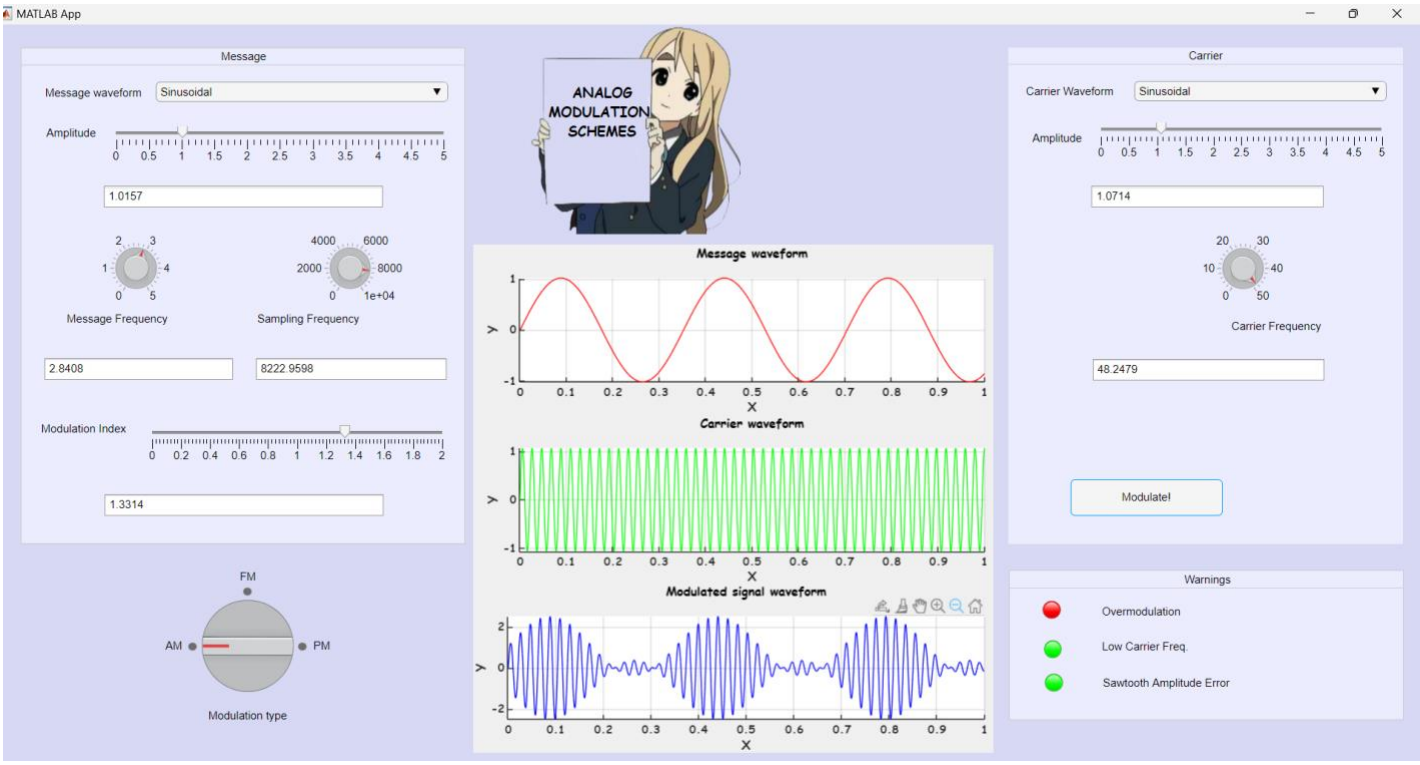
This app shows 3 warnings:

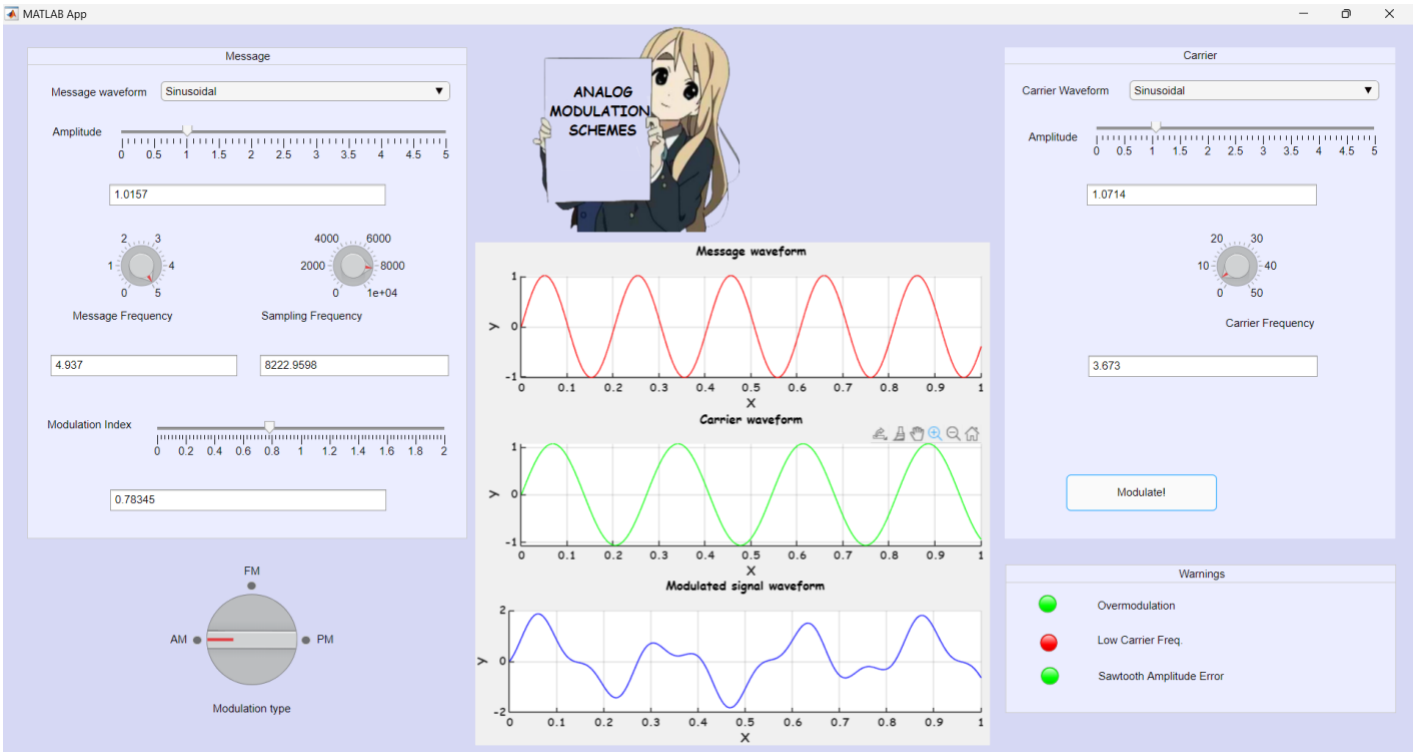
1. Over modulation warning when modulation index is greater than 1 in AM.

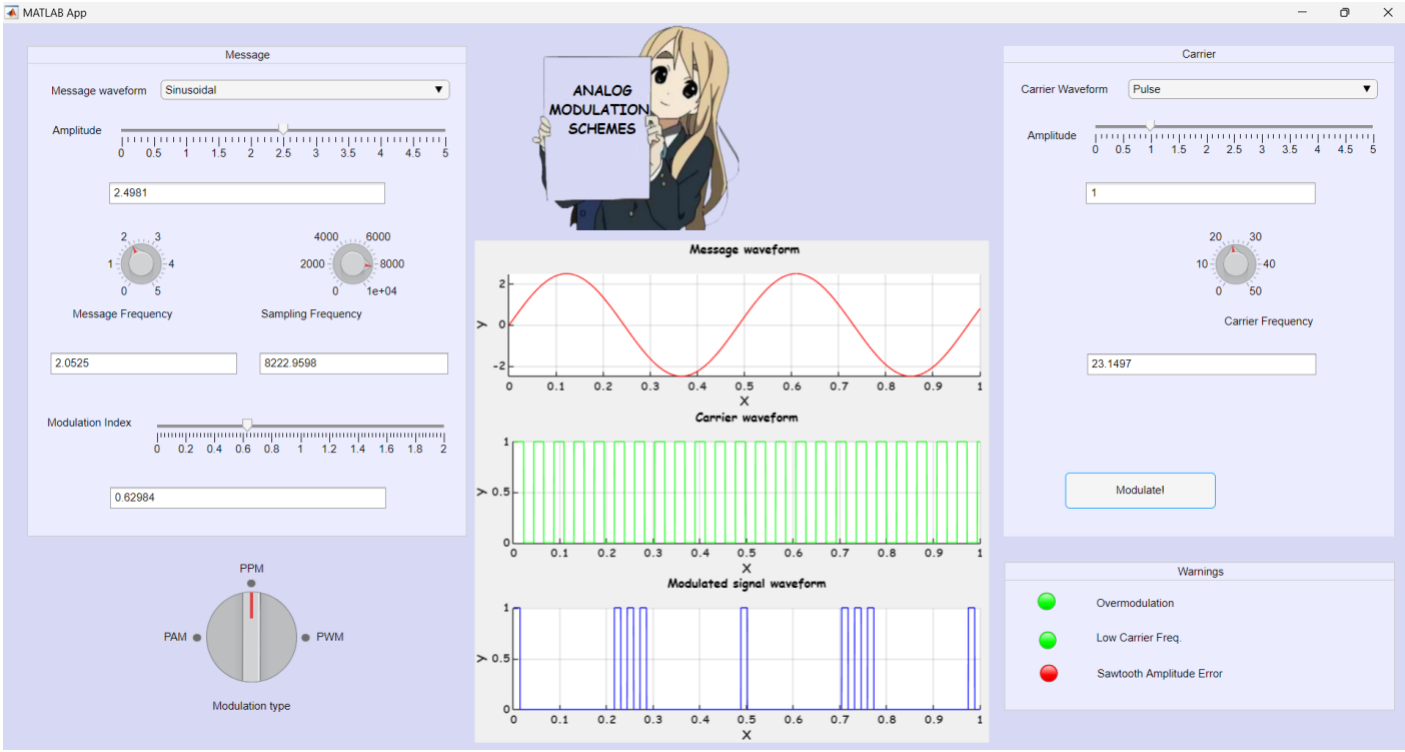
2. Low carrier frequency warning when $f_m > f_c$

3. Sawtooth amplitude error when $A_m > A_c$ in PWM and PPM.

Outputs for errors are given below:




MATLAB App
Message
Message waveform
Sinusoidal
Amplitude
1.0157
Message Frequency
4.937
Sampling Frequency
8222.9598
Modulation Index
0.78345
FM
AM
PM
Modulation type
ANALOG MODULATION SCHEMES
Message waveform
Carrier waveform
Modulated signal waveform
Carrier
Carrier Waveform
Sinusoidal
Amplitude
1.0714
Carrier Frequency
3.673
Modulate!
Warnings
Overmodulation
Low Carrier Freq.
Sawtooth Amplitude Error


MATLAB App
Message
Message waveform
Sinusoidal
Amplitude
2.4981
Message Frequency
2.0525
Sampling Frequency
8222.9598
Modulation Index
0.62984
PPM
PAM
PWM
Modulation type
ANALOG MODULATION SCHEMES
Message waveform
Carrier waveform
Modulated signal waveform
Carrier
Carrier Waveform
Pulse
Amplitude
1
Carrier Frequency
23.1497
Modulate!
Warnings
Overmodulation
Low Carrier Freq.
Sawtooth Amplitude Error

Limitations:

Sometimes it shows carrier waveform instead of showing modulated signal waveform. This is because of technical error so we couldn't fix it. But in that case if 'modulate' button is pressed again, it shows the correct output.

In case of showing warnings/error, if the bulb is on, then it must be turned off by giving proper input before going to other modulation techniques. Otherwise the bulb remains on.

We are planning to work on its limitations and add some other features to improve its performance in future.