



MINI PROJECT

การสร้าง GUI Application โดยใช้ MATLAB App Designer

จัดทำโดย

รวีสรา ชลวิจารณ์ 6101012630120

วัชรวิทย์ ประสาทไทย 6101012630151

ธราดล บ้านเนิน 6201012620104

เสนอ

อาจารย์ เรวัตติ ศิริโกคาภิรมย์

รายงานนี้เป็นส่วนหนึ่งของรายวิชา Intro to Signals & Systems

รหัสวิชา 010123106

ภาคเรียนที่ 1 ปีการศึกษา 2564

สาขาวิชา คอมพิวเตอร์ ภาควิชา ไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

หัวข้อที่ 1 สร้างสัญญาณรายคาบและแสดงรูปของสัญญาณ

วัตถุประสงค์

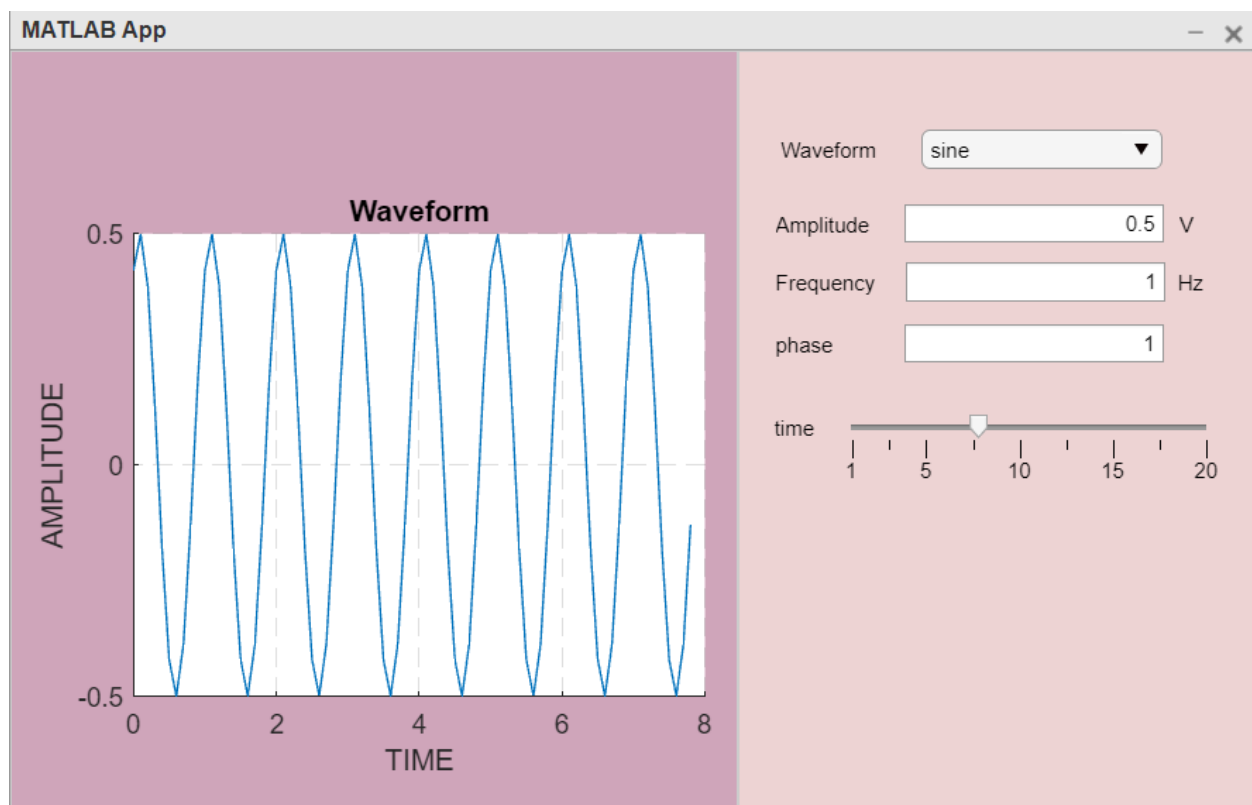
- ทดลองการใช้ MATLAB GUI
- สร้างสัญญาณมีคาบตามที่ต้องการ
- แสดงรูปของสัญญาณรายคาบที่สร้างออกมาได้

ขอบเขตของงาน

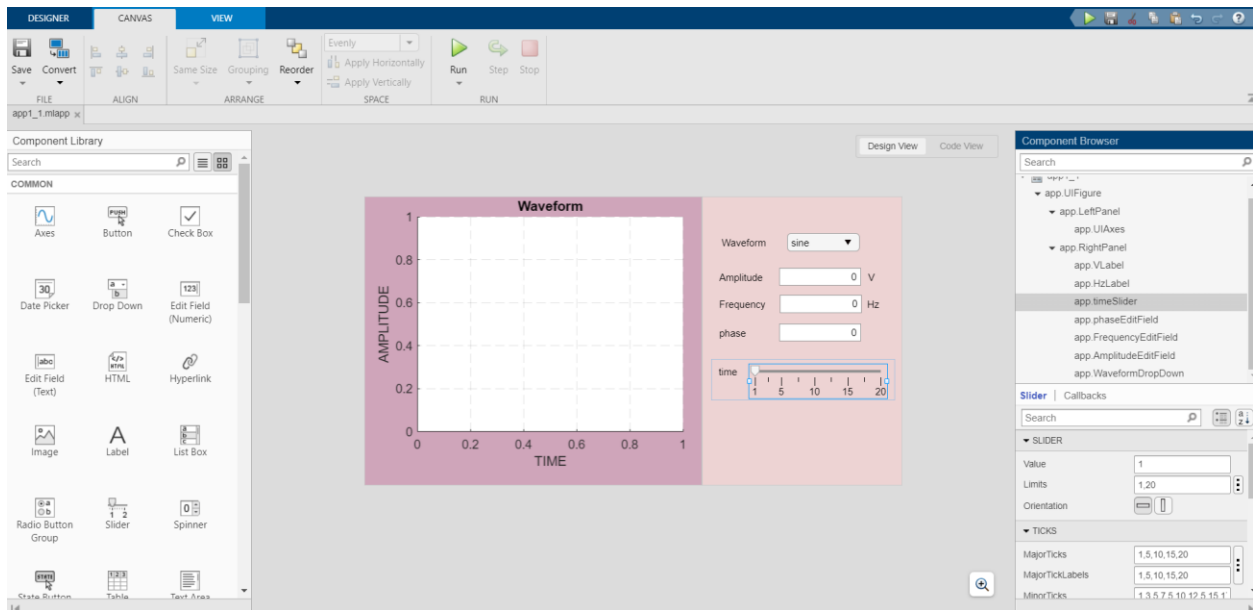
- ผู้ใช้สามารถเลือกชนิดของสัญญาณมีคาบ จาก dropdown menu เช่น square, triangular, PWM
- ตั้งค่าตัวเลขสำหรับพารามิเตอร์สำหรับสัญญาณที่เลือก เช่น แอมพลิจูด, ความถี่,
- แสดงรูปคลื่นสัญญาณตามพารามิเตอร์ของสัญญาณที่ผู้ใช้ปรับเปลี่ยน

ผลของการดำเนินการ

ทำการสร้างโปรแกรมที่สามารถใส่ค่า Amplitude, Frequency, phase, time เพื่อให้เห็นค่ากราฟที่ได้จากค่าเหล่านั้น โดยสามารถเลือกได้ว่าให้แสดงเป็นกราฟรูปแบบใด



สามารถสร้างหน้า GUI ง่ายๆเพียงแค่เลือก component ที่ต้องการใช้แล้วนำมาลากลงในหน้าจอ จึงทำให้ง่ายต่อการสร้างเป็นอย่างมาก



โค้ดจะประกอบไปด้วยสองส่วนคือ

1. ส่วนของ GUI ที่ MATLAB APP DESIGNER สร้างขึ้นมาโดยอัตโนมัติจากการที่เราลาก component และการตั้งค่า component ต่างๆในหน้า Design View
2. ส่วนของ Logic ต่างๆที่เราสร้างขึ้นเองเพื่อให้โปรแกรมทำงานตามที่เราต้องการ

GUI CODE

ในส่วนของ GUI CODE จะเป็นโค้ดที่จะกำหนดค่าต่างๆเพื่อสร้างหน้า GUI โดยจะถูกสร้างขึ้นเองโดยอัตโนมัติตามหน้า GUI ที่ผู้ใช้สร้างขึ้น โดยจะมี comment ให้อยู่ด้วย หากต้องการรู้ว่าสร้างยังไงก็สามารถหาส่วนที่ต้องการรู้ใน comment เพื่อหาโค้ดที่ต้องการได้

```
1 classdef app1_1 < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         UIFigure                matlab.ui.Figure
6         GridLayout              matlab.ui.container.GridLayout
7         LeftPanel               matlab.ui.container.Panel
8         UIAxes                  matlab.ui.control.UIAxes
9         RightPanel              matlab.ui.container.Panel
10        VLabel                   matlab.ui.control.Label
11        HzLabel                  matlab.ui.control.Label
12        timeSlider               matlab.ui.control.Slider
13        timeSliderLabel          matlab.ui.control.Label
14        phaseEditField           matlab.ui.control.NumericEditField
15        phaseEditFieldLabel      matlab.ui.control.Label
16        FrequencyEditField       matlab.ui.control.NumericEditField
17        FrequencyEditFieldLabel  matlab.ui.control.Label
18        AmplitudeEditField       matlab.ui.control.NumericEditField
19        AmplitudeEditFieldLabel  matlab.ui.control.Label
20        WaveformDropDown         matlab.ui.control.DropDown
21        WaveformDropDownLabel    matlab.ui.control.Label
22    end
23
24    % Properties that correspond to apps with auto-reflow
25    properties (Access = private)
26        onePanelWidth = 576;
27    end
28
29    % Callbacks that handle component events
30    methods (Access = private)
31
32        % Callback function: AmplitudeEditField, FrequencyEditField,
33        % WaveformDropDown, WaveformDropDown, phaseEditField, timeSlider,
34        % timeSlider
35        function runButtonPushed(app, event)
```

LOGIC CODE

```
36
37 Phase = app.phaseEditField.Value;
38 freq = app.FrequencyEditField.Value;
39 Ampl =app.AmplitudeEditField.Value;
40 T = app.timeSlider.Value;
41 time = 0:0.10:T;
42
43 switch app.WaveformDropDown.Value
44     case 'sine'
45         Volt = Ampl*sin(2*pi*freq*time+Phase);
46         plot(app.UIAxes, time, Volt);
47     case 'square'
48         offset=0;
49         duty=50;
50         sq_wav=offset+Ampl*square(2*pi*freq*time,duty);
51         plot(app.UIAxes, time, sq_wav);
52     case 'triangle'
53         plot(app.UIAxes, time, Ampl*sawtooth(2*pi*freq*time,1/2));
54     case 'sawtooth'
55         plot(app.UIAxes, time, Ampl*sawtooth(2*pi*freq*time));
56 end
```

ในส่วนนี้เราจะทำการเขียนโค้ดโดยสร้างตัวแปรที่จะรับค่าต่างๆที่ต้องใส่จากหน้า GUI ที่เราต้องการนำมาเข้าสู่ตรรกศาสตร์เพื่อสร้างกราฟรูปแบบต่างๆที่เราต้องการ โดยเราได้ทำการนำสูตรการวาดกราฟมา 4 ชนิดคือ กราฟ sine, square, triangle, sawtooth

GUI CODE

```
57         end
58
59         % Changes arrangement of the app based on UIFigure width
60         function updateAppLayout(app, event)
61             currentFigureWidth = app.UIFigure.Position(3);
62             if(currentFigureWidth <= app.onePanelWidth)
63                 % Change to a 2x1 grid
64                 app.GridLayout.RowHeight = {353, 353};
65                 app.GridLayout.ColumnWidth = {'1x'};
66                 app.RightPanel.Layout.Row = 2;
67                 app.RightPanel.Layout.Column = 1;
68             else
69                 % Change to a 1x2 grid
70                 app.GridLayout.RowHeight = {'1x'};
71                 app.GridLayout.ColumnWidth = {413, '1x'};
72                 app.RightPanel.Layout.Row = 1;
73                 app.RightPanel.Layout.Column = 2;
74             end
75         end
76     end
77
78     % Component initialization
79     methods (Access = private)
80
81         % Create UIFigure and components
82         function createComponents(app)
83
84             % Create UIFigure and hide until all components are created
85             app.UIFigure = uifigure('Visible', 'off');
86             app.UIFigure.AutoResizeChildren = 'off';
87             app.UIFigure.Position = [100 100 658 353];
88             app.UIFigure.Name = 'MATLAB App';
89             app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);
90
91             % Create GridLayout
92             app.GridLayout = uigridlayout(app.UIFigure);
93             app.GridLayout.ColumnWidth = {413, '1x'};
```

```

94     app.GridLayout.RowHeight = {'1x'};
95     app.GridLayout.ColumnSpacing = 0;
96     app.GridLayout.RowSpacing = 0;
97     app.GridLayout.Padding = [0 0 0 0];
98     app.GridLayout.Scrollable = 'on';
99
100    % Create LeftPanel
101    app.LeftPanel = uipanel(app.GridLayout);
102    app.LeftPanel.BackgroundColor = [0.8118 0.6471 0.7294];
103    app.LeftPanel.Layout.Row = 1;
104    app.LeftPanel.Layout.Column = 1;
105
106    % Create UIAxes
107    app.UIAxes = uiaxes(app.LeftPanel);
108    title(app.UIAxes, 'Waveform')
109    xlabel(app.UIAxes, 'TIME')
110    ylabel(app.UIAxes, 'AMPLITUDE')
111    zlabel(app.UIAxes, 'Z')
112    app.UIAxes.GridLineStyle = '--';
113    app.UIAxes.XGrid = 'on';
114    app.UIAxes.YGrid = 'on';
115    app.UIAxes.FontSize = 15;
116    app.UIAxes.MinorGridColor = [0 0 0];
117    app.UIAxes.Position = [14 20 385 332];
118
119    % Create RightPanel
120    app.RightPanel = uipanel(app.GridLayout);
121    app.RightPanel.BackgroundColor = [0.9294 0.8275 0.8275];
122    app.RightPanel.Layout.Row = 1;
123    app.RightPanel.Layout.Column = 2;
124
125    % Create WaveformDropDownLabel
126    app.WaveformDropDownLabel = uilabel(app.RightPanel);
127    app.WaveformDropDownLabel.Position = [24 286 59 22];
128    app.WaveformDropDownLabel.Text = 'Waveform';
129

```

```

129
130 % Create WaveformDropDown
131 app.WaveformDropDown = uidropdown(app.RightPanel);
132 app.WaveformDropDown.Items = {'sine', 'square', 'triangle', 'sawtooth', 'Option 4'};
133 app.WaveformDropDown.DropDownOpeningFcn = createCallbackFcn(app, @runButtonPushed, true);
134 app.WaveformDropDown.ValueChangedFcn = createCallbackFcn(app, @runButtonPushed, true);
135 app.WaveformDropDown.Position = [104 286 89 22];
136 app.WaveformDropDown.Value = 'sine';
137
138 % Create AmplitudeEditFieldLabel
139 app.AmplitudeEditFieldLabel = uilabel(app.RightPanel);
140 app.AmplitudeEditFieldLabel.Position = [21 244 58 22];
141 app.AmplitudeEditFieldLabel.Text = 'Amplitude';
142
143 % Create AmplitudeEditField
144 app.AmplitudeEditField = uieditfield(app.RightPanel, 'numeric');
145 app.AmplitudeEditField.ValueChangedFcn = createCallbackFcn(app, @runButtonPushed, true);
146 app.AmplitudeEditField.Position = [94 244 100 22];
147
148 % Create FrequencyEditFieldLabel
149 app.FrequencyEditFieldLabel = uilabel(app.RightPanel);
150 app.FrequencyEditFieldLabel.Position = [21 211 62 22];
151 app.FrequencyEditFieldLabel.Text = 'Frequency';
152
153 % Create FrequencyEditField
154 app.FrequencyEditField = uieditfield(app.RightPanel, 'numeric');
155 app.FrequencyEditField.ValueChangedFcn = createCallbackFcn(app, @runButtonPushed, true);
156 app.FrequencyEditField.Position = [95 211 100 22];
157
158 % Create phaseEditFieldLabel
159 app.phaseEditFieldLabel = uilabel(app.RightPanel);
160 app.phaseEditFieldLabel.Position = [21 176 38 22];
161 app.phaseEditFieldLabel.Text = 'phase';

```



```

163         % Create phaseEditField
164         app.phaseEditField = uieditfield(app.RightPanel, 'numeric');
165         app.phaseEditField.ValueChangedFcn = createCallbackFcn(app, @runButtonPushed, true);
166         app.phaseEditField.Position = [94 176 100 22];
167
168         % Create timeSliderLabel
169         app.timeSliderLabel = uilabel(app.RightPanel);
170         app.timeSliderLabel.HorizontalAlignment = 'right';
171         app.timeSliderLabel.Position = [15 129 28 22];
172         app.timeSliderLabel.Text = 'time';
173
174         % Create timeSlider
175         app.timeSlider = uislider(app.RightPanel);
176         app.timeSlider.Limits = [1 20];
177         app.timeSlider.MajorTicks = [1 5 10 15 20];
178         app.timeSlider.ValueChangedFcn = createCallbackFcn(app, @runButtonPushed, true);
179         app.timeSlider.ValueChangingFcn = createCallbackFcn(app, @runButtonPushed, true);
180         app.timeSlider.MinorTicks = [1 3 5 7.5 10 12.5 15 17.5 20];
181         app.timeSlider.Position = [64 138 154 3];
182         app.timeSlider.Value = 1;
183
184         % Create HzLabel
185         app.HzLabel = uilabel(app.RightPanel);
186         app.HzLabel.Position = [202 211 25 22];
187         app.HzLabel.Text = 'Hz';
188
189         % Create VLabel
190         app.VLabel = uilabel(app.RightPanel);
191         app.VLabel.Position = [203 244 25 22];
192         app.VLabel.Text = 'V';
193
194         % Show the figure after all components are created
195         app.UIFigure.Visible = 'on';
196     end
197 end

```

```
198
199     % App creation and deletion
200     methods (Access = public)
201
202         % Construct app
203         function app = app1_1
204
205             % Create UIFigure and components
206             createComponents(app)
207
208             % Register the app with App Designer
209             registerApp(app, app.UIFigure)
210
211             if nargin == 0
212                 clear app
213             end
214         end
215
216         % Code that executes before app deletion
217         function delete(app)
218
219             % Delete UIFigure when app is deleted
220             delete(app.UIFigure)
221         end
222     end
223 end
```