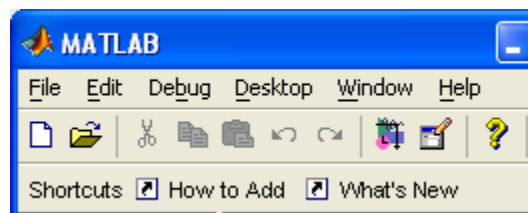


How to write a graphic user interface in Matlab

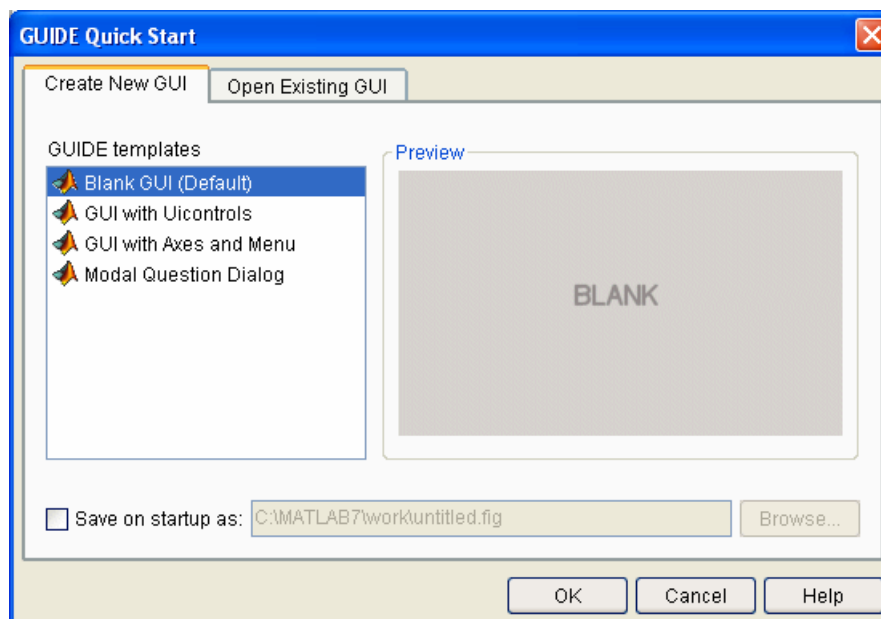
This document will describe how to create a simple Matlab graphic user interface (GUI). Please refer to the topic “Creating Graphical User Interfaces” in Matlab Help for a detail instruction. In this example, we will learn how to create a slider bar and a callback function to execute codes when the slider bar is moved. We will use this slider bar to control position of a servo motor on the Servo Robot. We also need to create a button (of course, with its callback function) to close the serial port, before we close the window.

Here is step-by-step instructions:

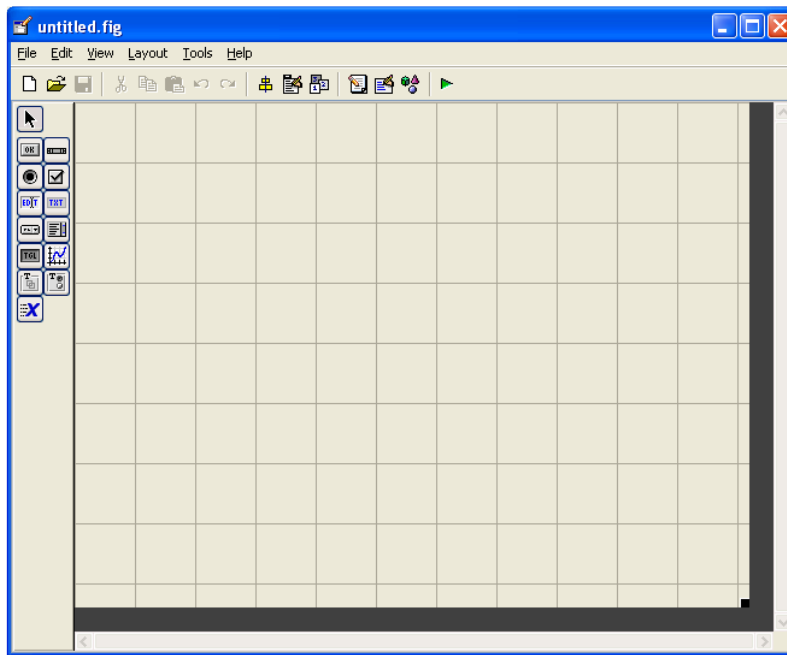
1. Type command “guide” in the command line, or click on the toolbar icon.



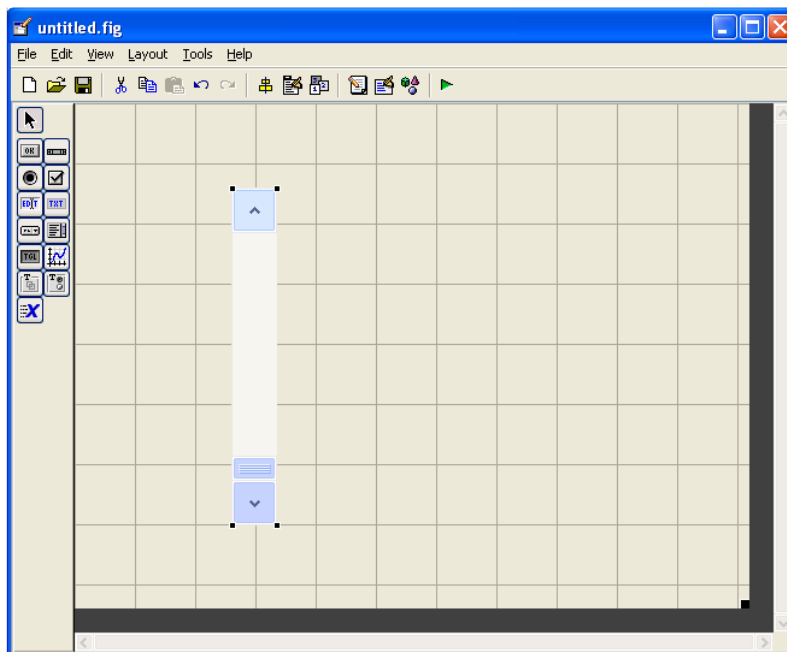
2. Matlab will open GUIDE window, In GUIDE window, Select Blank GUI and click OK.



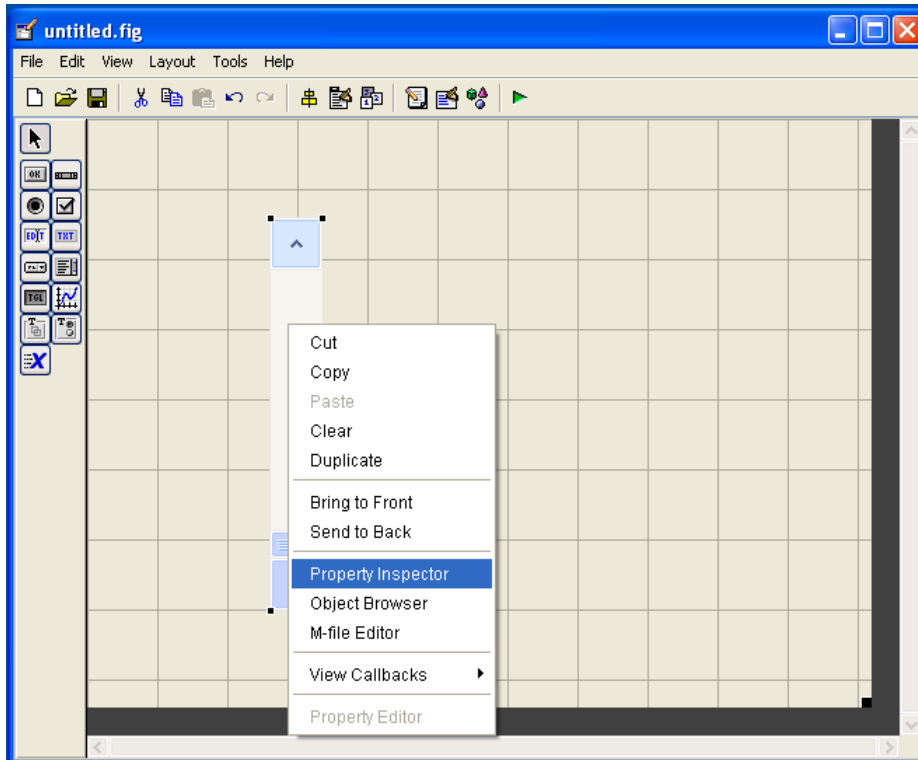
3. Matlab will open a design workspace.



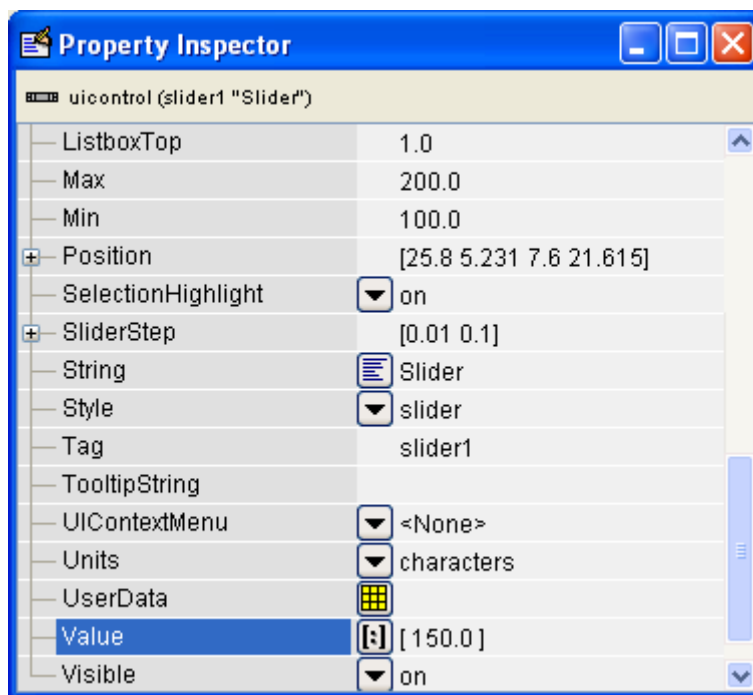
4. We will need a slider bar to control the position of the servo motor. So let's click on the slider bar icon and draw a slider in the empty workspace.



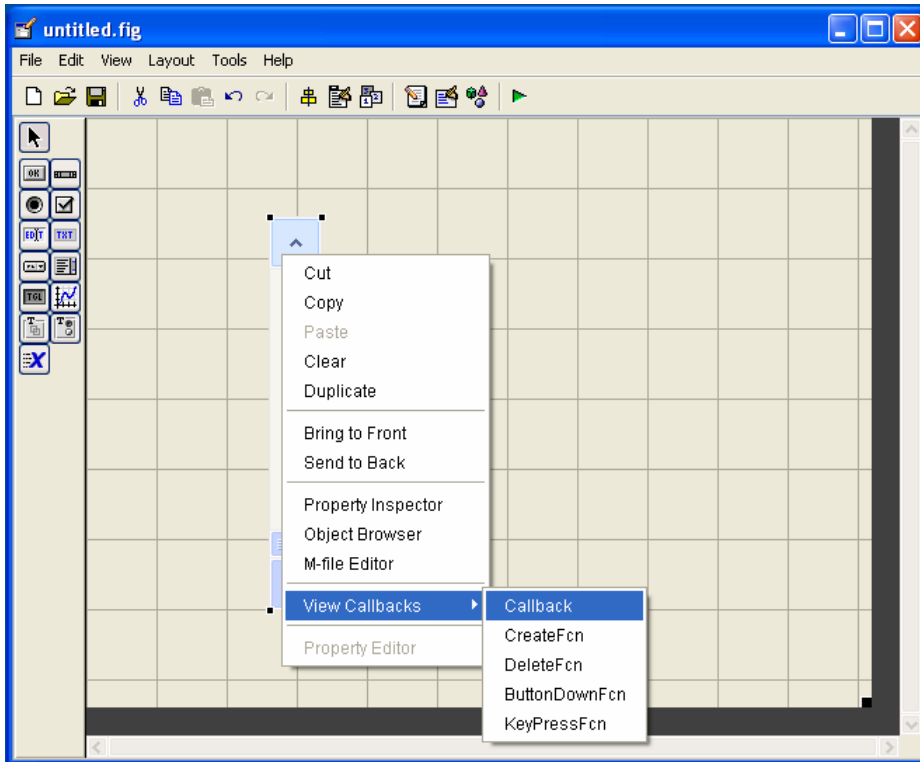
5. Right click on the slider bar and select Property Inspector.



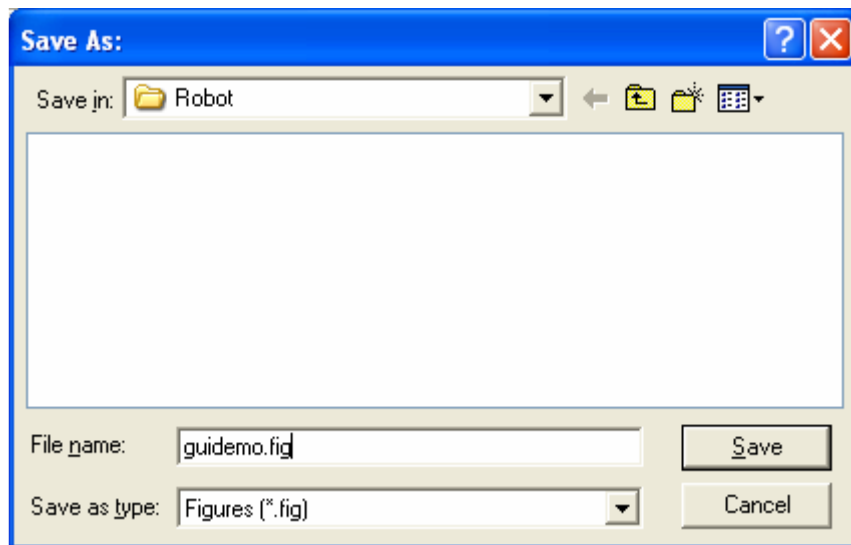
6. Change value in the Property Inspector window. Max = 200 , Min = 100, and Value = 150. After finish, close the Property Inspector window. Note that Max/Min is the maximum/minimum positions of the slider bar. Value is the initial position of the slider bar.



7. Right click on the slider bar, and select View Callbacks > Callback.



8. A Save As dialog will appear. Save it as the file: guidemo.fig



9. Matlab will automatically generate file guidemo.m.

```

Editor - C:\Uttichote\At Work\Mahidol Research\Servo Robot\Matlab\Robot\guidemo.m
File Edit Text Cell Tools Debug Desktop Window Help
74 % Get default command line output from handles structure
75 varargout{1} = handles.output;
76
77
78 % --- Executes on slider movement.
79 function slider1_Callback(hObject, eventdata, handles)
80 % hObject    handle to slider1 (see GCBO)
81 % eventdata  reserved - to be defined in a future version of MA
82 % handles    structure with handles and user data (see GUIDATA)
83
84 % Hints: get(hObject,'Value') returns position of slider
85 %         get(hObject,'Min') and get(hObject,'Max') to determine
86
87
88 % --- Executes during object creation, after setting all proper

```

10. The function slider1_Callback() is the function that is called when the slider is moved. The position of the sliderbar can be obtained by using the command get(hObject,'Value'). Add the command to move the servo motor #1 when the slider is moved.

```
sendcmd(handles.s,1,get(hObject,'Value'));
```

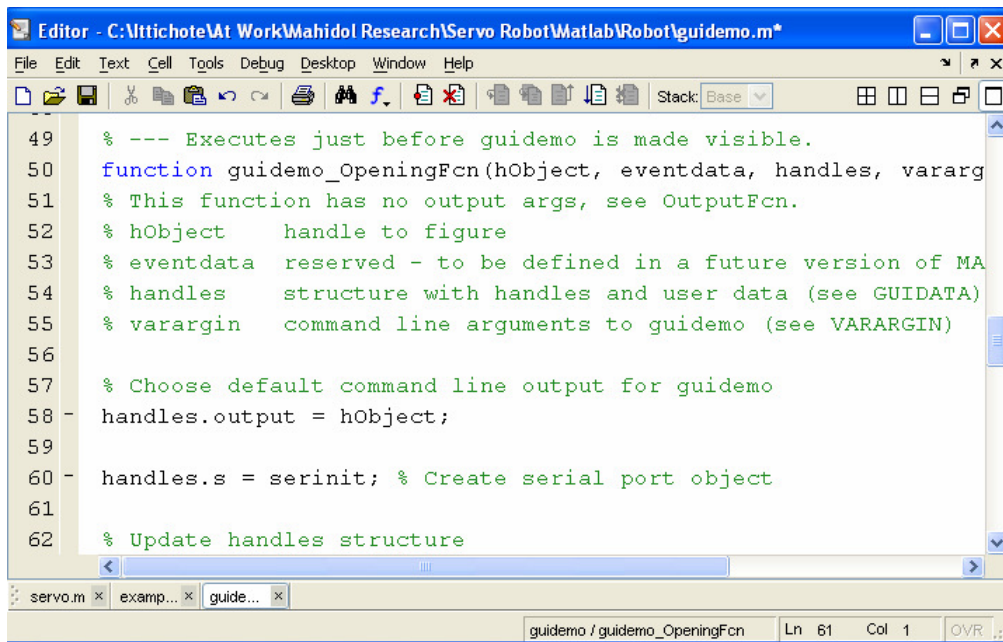
```

Editor - C:\Uttichote\At Work\Mahidol Research\Servo Robot\Matlab\Robot\guidemo.m
File Edit Text Cell Tools Debug Desktop Window Help
79
80 % --- Executes on slider movement.
81 function slider1_Callback(hObject, eventdata, handles)
82 % hObject    handle to slider1 (see GCBO)
83 % eventdata  reserved - to be defined in a future version of MA
84 % handles    structure with handles and user data (see GUIDATA)
85
86 % Hints: get(hObject,'Value') returns position of slider
87 %         get(hObject,'Min') and get(hObject,'Max') to determine
88 sendcmd(handles.s,1,get(hObject,'Value'));
89
90 % --- Executes during object creation, after setting all proper
91 function slider1_CreateFcn(hObject, eventdata, handles)
92 % hObject    handle to slider1 (see GCBO)
93 % eventdata  reserved - to be defined in a future version of MA

```

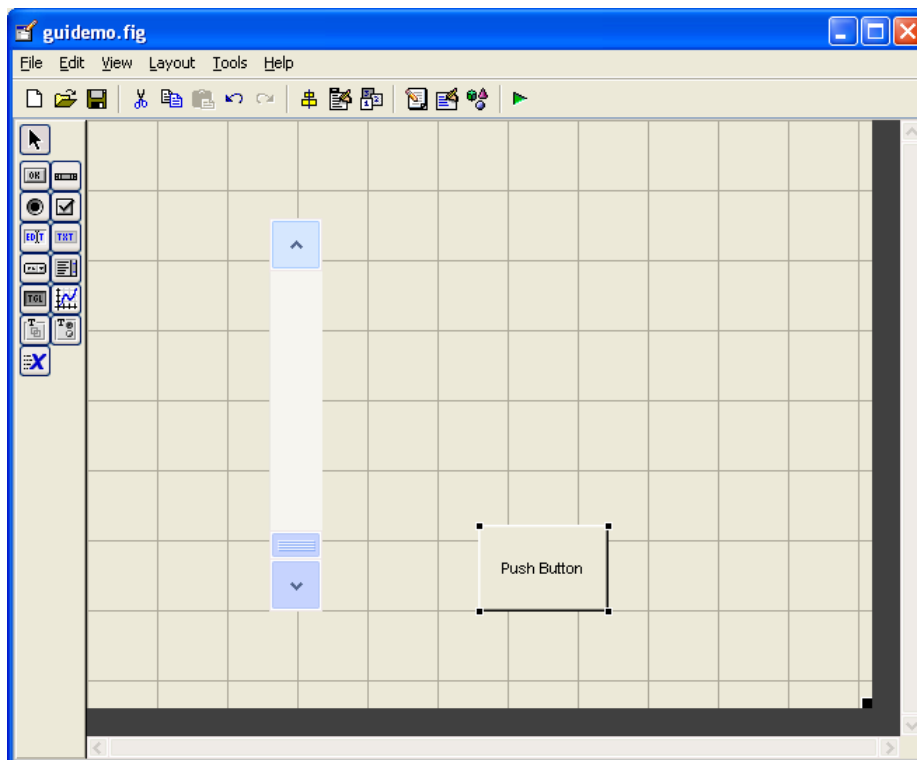
11. In function guidemo_OpeningFcn(), Add the command to initialize the serial port. This function will be called when the GUI window is initialized. So the serial port will be initialed automatically when the window is started.

```
handles.s = serinit; % Create serial port object
```

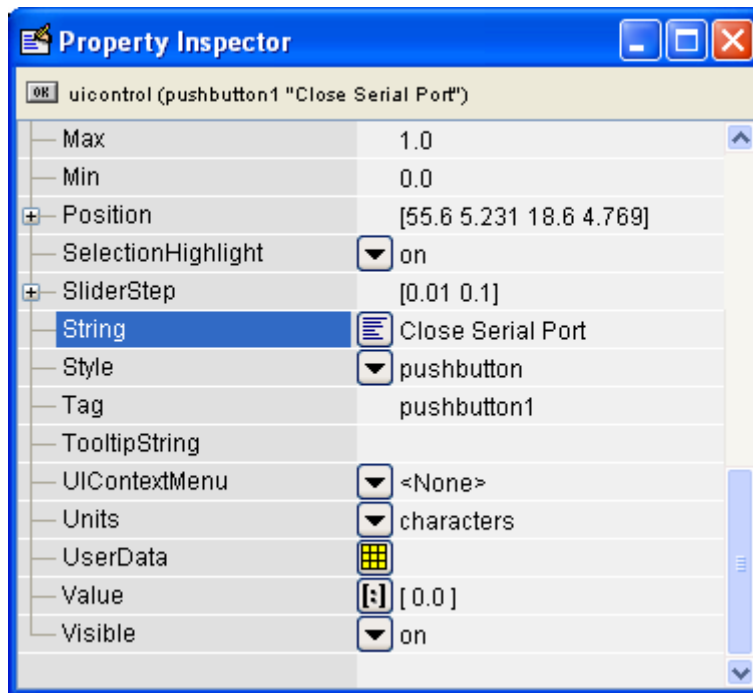


```
49 % --- Executes just before guidemo is made visible.
50 function guidemo_OpeningFcn(hObject, eventdata, handles, varargin)
51 % This function has no output args, see OutputFcn.
52 % hObject    handle to figure
53 % eventdata  reserved - to be defined in a future version of MA
54 % handles    structure with handles and user data (see GUIDATA)
55 % varargin   command line arguments to guidemo (see VARARGIN)
56
57 % Choose default command line output for guidemo
58 handles.output = hObject;
59
60 handles.s = serinit; % Create serial port object
61
62 % Update handles structure
```

12. Add a button on the design workspace. This button will be used to close the serial port.



13. Right click on the button and select Property Inspector. In Property Inspector window, change the value String =“Close Serial Port”. Close the Property Inspector window, and the string on the button should be changed to what we just put in.



14. Right click on the slider bar, and select View Callbacks > Callback. Matlab will automatically add callback function for the button into the file guidemo.m. In the file guidemo.m, add the command to close the serial port.

```

serclose(handles.s);

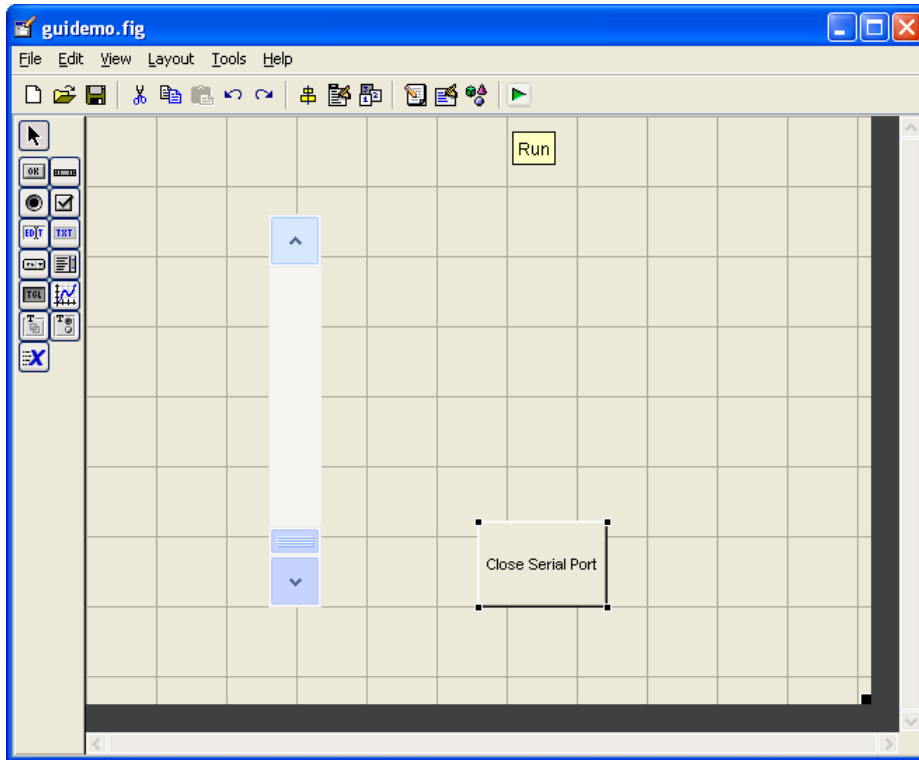
```

```

102 -     set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgr
103 - end
104
105
106
107
108 % --- Executes on button press in pushbutton1.
109 function pushbutton1_Callback(hObject, eventdata, handles)
110 % hObject     handle to pushbutton1 (see GCBO)
111 % eventdata   reserved - to be defined in a future version of MA
112 % handles     structure with handles and user data (see GUIDATA)
113 - serclose(handles.s);
114
115

```

15. Run the GUI window by clicking the run icon (green arrow sign).



16. The window is running. If the robot is connected, you should be able to move joint 1 of the robot by using the slider bar. Before exit, you must click Close Serial Port once, otherwise, you will not be able to reinitialize the serial port next time the program is started. In this case, you can close and restart Matlab program to fix the problem.

