Drexel University

To: Dr. Peters

From: Tyler Ostinato

Date: October 25, 2021

Re: ECE 303-Lab 5-Graphical Interfaces

Purpose

Create a working graphical user interface in MATLAB to interact with the Arduino. In this lab we will create a GUI to display Voltage, Resistance, and Current in our photosensor from lab 4 in real time.

Discussion

App Designer | 1.1

Creating GUIs in MATLAB is made very simple using the app designer built into MATLAB. This application allows users to drag and drop buttons, text boxes, graphs, etc. to create a GUI to work with MATLAB functions.

Using this app designer, I created a simple GUI to track voltage, current, and resistance in our photoresistor circuit at increasing duty cycles. I was able to due this using the Axes option for the app designer, which allows MATLAB to input data into a graph in real time. I also added a text box to display the current duty cycle of the LED circuit (Figure 1).

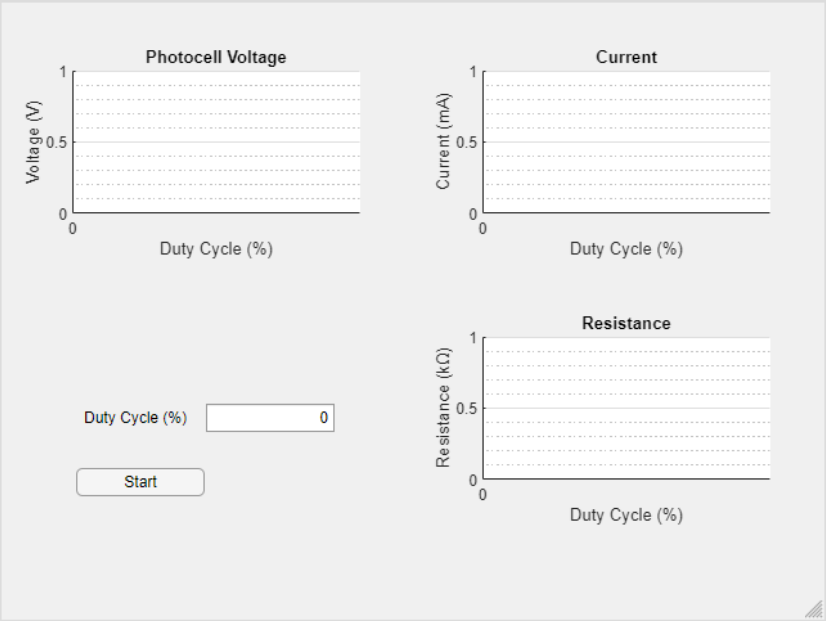


Figure 1:GUI created using MATLAB's App Designer

Callbacks | 1.2

Now that we have added all our buttons, we need to make them do something. To do this we can right click on a button a select “callbacks”. This will bring you to the code view of the GUI and allow you to add logic to a button. For this button I created a call back for the start button. This button contains the code to talk with the Arduino to get the data of our photosensor circuit, so when we push the button, it will run our code from lab 4.

Plotting | 1.3

Lastly, we need to get our data from the Arduino onto the GUIs plots. This can be done by calling the respective graph (ex: app.UIAxes_1) and input our x and y data as we would normally do in MATLAB (Figure 2).

```
plot(app.UIAxes,DC(1:(K+1)),V_pc(1:(K+1)),'bo','MarkerFaceColor','b')
plot(app.UIAxes_2,DC(1:(K+1)),I(1:(K+1)),'bo','MarkerFaceColor','b')
plot(app.UIAxes_3,DC(1:(K+1)),R(1:(K+1)),'bo','MarkerFaceColor','b')
```

Figure 2: Setup GUI graph output.

Results | 1.3

After all buttons have been setup, we can now run the GUI using the start button. MATLAB will then start plotting the data in real time (Figure 3).

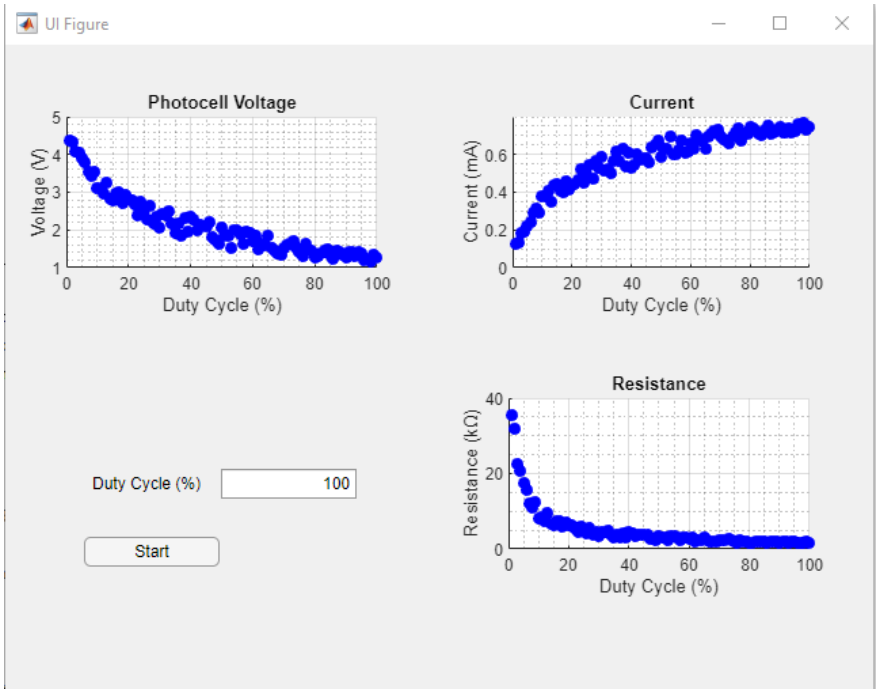
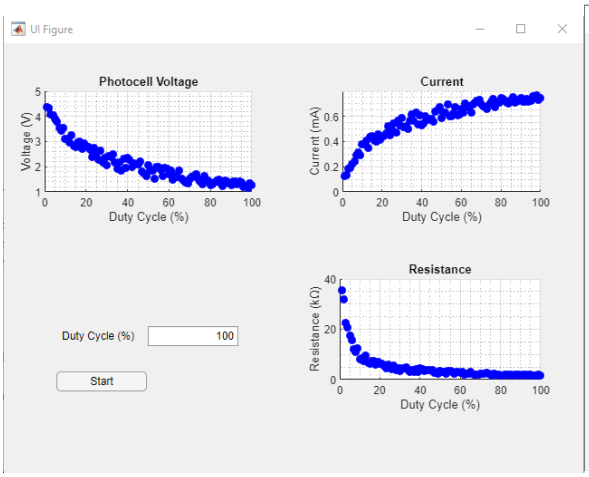


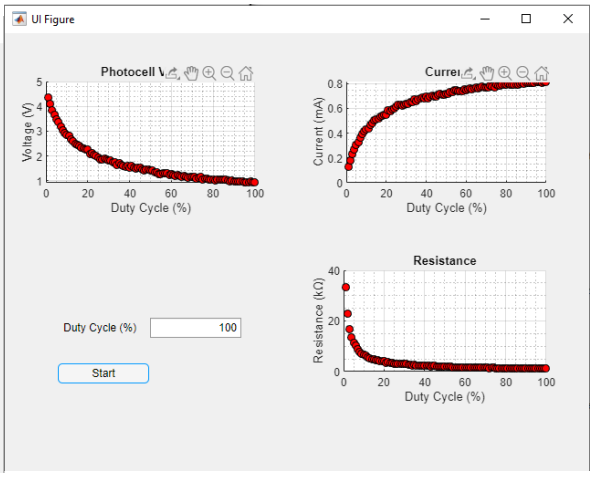
Figure 3:Working GUI using App Designer.

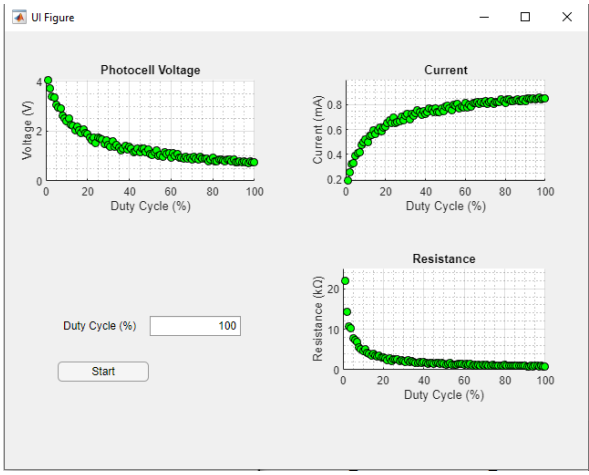
Conclusion

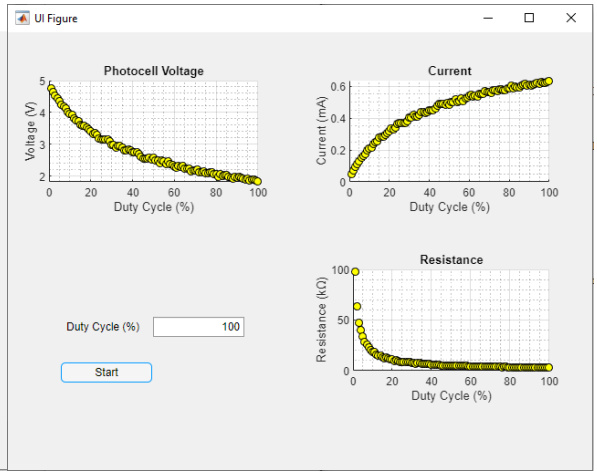
In this lab we successfully created a GUI in MATLAB using the app designer. This GUI allowed us to create a more convenient way to run our Arduino code and display the results. Before the GUI MATLAB would display the graphs separately in a non-user-friendly way. GUIs allow users to create a centralized place for information to go that is convenient for users. Now that I have experience with the MATLAB app designer, I can create more GUIs in the future to better display my data.

Appendix


UI Figure
Photocell Voltage
Voltage (V)
Duty Cycle (%)
Current
Current (mA)
Duty Cycle (%)
Resistance
Resistance (kΩ)
Duty Cycle (%)
Duty Cycle (%)
100
Start


UI Figure
Voltage (V)
Duty Cycle (%)
Current (mA)
Duty Cycle (%)
Resistance
Resistance (kΩ)
Duty Cycle (%)
Duty Cycle (%)
100
Start


UI Figure
Photocell Voltage
Voltage (V)
Duty Cycle (%)
Current
Current (mA)
Duty Cycle (%)
Resistance
Resistance (kΩ)
Duty Cycle (%)
Duty Cycle (%)
100
Start


UI Figure
Photocell Voltage
Voltage (V)
Duty Cycle (%)
Current
Current (mA)
Duty Cycle (%)
Resistance
Resistance (kΩ)
Duty Cycle (%)
Duty Cycle (%)
100
Start

App Designer Code

```
classdef lab5_gui2 < matlab.apps.AppBase


    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        UIAxes                  matlab.ui.control.UIAxes
        UIAxes_2                matlab.ui.control.UIAxes
        UIAxes_3                matlab.ui.control.UIAxes
        StartButton             matlab.ui.control.Button
        DutyCycleEditFieldLabel  matlab.ui.control.Label
        DutyCycleEditField      matlab.ui.control.NumericEditField
    end


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


        % Button pushed function: StartButton
        function StartButtonPushed(app, event)
            % Set up communications
            arduino=serialport("COM8",9600,"Timeout",15);
            pause(1)
            num_points=100;
            DC=zeros(1,num_points);
            V_res=zeros(1,num_points);
            V_pc=zeros(1,num_points);
            I=zeros(1,num_points);
            R=zeros(1,num_points);

            for K=0:(num_points)
                DC(K+1)=K;
                flush(arduino)
                write(arduino,2,'string');
                pause(0.5)
                a=read(arduino,4,'string');
                flush(arduino)
                V_res(K+1)=str2double(a)/1023*5;
                V_pc(K+1)=5-V_res(K+1);
                I(K+1)=V_res(K+1)/5000*1000;
                R(K+1)=V_pc(K+1)/I(K+1);
                app.DutyCycleEditField.Value=DC(K+1);

plot(app.UIAxes,DC(1:(K+1)),V_pc(1:(K+1)),'bo','MarkerFaceColor','b')

plot(app.UIAxes_2,DC(1:(K+1)),I(1:(K+1)),'bo','MarkerFaceColor','b')
```

```
plot(app.UIAxes_3,DC(1:(K+1)),R(1:(K+1)),'bo','MarkerFaceColor','b')
            end
            delete(arduino);
            clear arduino;
        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.Position = [100 100 640 480];
            app.UIFigure.Name = 'UI Figure';


            % Create UIAxes
            app.UIAxes = uiaxes(app.UIFigure);
            title(app.UIAxes, 'Photocell Voltage')
            xlabel(app.UIAxes, 'Duty Cycle (%)')
            ylabel(app.UIAxes, 'Voltage (V)')
            app.UIAxes.XTick = [0 20 40 60 80 100];
            app.UIAxes.XTickLabel = {'0'; '20'; '40'; '60'; '80'; '100'};
            app.UIAxes.XGrid = 'on';
            app.UIAxes.XMinorGrid = 'on';
            app.UIAxes.YGrid = 'on';
            app.UIAxes.YMinorGrid = 'on';
            app.UIAxes.Position = [17 282 268 166];


            % Create UIAxes_2
            app.UIAxes_2 = uiaxes(app.UIFigure);
            title(app.UIAxes_2, 'Current')
            xlabel(app.UIAxes_2, 'Duty Cycle (%)')
            ylabel(app.UIAxes_2, 'Current (mA)')
            app.UIAxes_2.XTick = [0 20 40 60 80 100];
            app.UIAxes_2.XTickLabel = {'0'; '20'; '40'; '60'; '80'; '100'};
            app.UIAxes_2.XGrid = 'on';
            app.UIAxes_2.XMinorGrid = 'on';
            app.UIAxes_2.YGrid = 'on';
            app.UIAxes_2.YMinorGrid = 'on';
            app.UIAxes_2.Position = [336 282 268 166];
```

```
            % Create UIAxes_3
            app.UIAxes_3 = uiaxes(app.UIFigure);
            title(app.UIAxes_3, 'Resistance')
            xlabel(app.UIAxes_3, 'Duty Cycle (%)')
            ylabel(app.UIAxes_3, 'Resistance (kΩ)')
            app.UIAxes_3.XTick = [0 20 40 60 80 100];
            app.UIAxes_3.XTickLabel = {'0'; '20'; '40'; '60'; '80'; '100'};
            app.UIAxes_3.XGrid = 'on';
            app.UIAxes_3.XMinorGrid = 'on';
            app.UIAxes_3.YGrid = 'on';
            app.UIAxes_3.YMinorGrid = 'on';
            app.UIAxes_3.Position = [336 75 268 166];


            % Create StartButton
            app.StartButton = uibutton(app.UIFigure, 'push');
            app.StartButton.ButtonPushedFcn = createCallbackFcn(app,
@StartButtonPushed, true);
            app.StartButton.Position = [59 97 100 22];
            app.StartButton.Text = 'Start';


            % Create DutyCycleEditFieldLabel
            app.DutyCycleEditFieldLabel = uilabel(app.UIFigure);
            app.DutyCycleEditFieldLabel.HorizontalAlignment = 'right';
            app.DutyCycleEditFieldLabel.Position = [59 147 86 22];
            app.DutyCycleEditFieldLabel.Text = 'Duty Cycle (%)';


            % Create DutyCycleEditField
            app.DutyCycleEditField = uieditfield(app.UIFigure, 'numeric');
            app.DutyCycleEditField.Position = [160 147 100 22];


            % 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 = lab5_gui2


            % 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
```

Arduino Sketch

```
int photo_pin(A0);

int val1=0;

unsigned int val=0;

unsigned int counter=0;

int integer_arr[10];


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(5,OUTPUT);
  Serial.setTimeout(100);
}
```

```
void loop() {

  int count_samples=0;
  int mean=0;
  // Get 10 samples into an array
  while(count_samples < 10){
    val=Serial.parseInt();
    analogWrite(5,counter); // Change duty cycle
    val1 = analogRead(photo_pin);
    integer_arr[count_samples] = val1;
    count_samples++;
    mean+= val1;
  }

  // Calculate standard deviation and remove values that fall outside of it
  mean = mean/10;
  // Calculate the variance
  float variance_total=0;
  int variance_count=0;
  while(variance_count < 10){
    int power_val = (integer_arr[variance_count] - mean) * (integer_arr[variance_count] - mean);
    variance_total += power_val;
    variance_count++;
  }

  // Get the standard deviation
  variance_total = variance_total/10;
```

```
 int sd = sqrt(variance_total);

 // Remove values over 1 SD away from the mean
 // Add remaining values and take the avg; this will be the avg of the valid sample data
 int avg_of_sample=0;
 int num_of_valid_samples=0;
 for(int i=0; i<10; i++){
  if(abs(integer_arr[i] - mean) < variance_total){
    avg_of_sample += integer_arr[i];
    num_of_valid_samples++;
  }
 }

 Serial.println(abs(avg_of_sample/num_of_valid_samples));

 counter+=2.55; // increment the LED duty cycle after getting samples
}
```