Senior Design
ENG EC 464

To: Professor Pisano

From: Aidan McCall, Jonas Escobar

Team: Smart Grid (30)

Date: 04/29/22

Subject: Software Readme

1.0 Introduction

For more specific information on the feedback loop MSP430 code, refer to the code and documentation in the previous group's folder.

2.0 Software Setup

The following programs/packages are required for operation of the Smart Grid:

- MATLAB
- MATLAB Add-Ons:
  - MATLAB Signal Processing Toolbox
  - MATLAB Data Acquisition Toolbox
  - MATLAB Data Acquisition Toolbox Support Package for NI-DAQmx Devices
  - Phase Difference Measurement function
- DAQ driver software: NI DAQmx Elvis
- Final MATLAB GUI
- PicoScope 6 App

3.0 PicoScope

To set up and run the PicoScope, plug the device into the computer and open the PicoScope 6 app. When the PicoScope is connected, the program will automatically begin recording and displaying live data. The user may have to adjust the settings to make sure both channels are set to "Auto," and the start/stop buttons in the bottom left can be used to start or pause the live feed (Figure 1). Refer to the Hardware Readme for details on how to attach the leads to the grid.

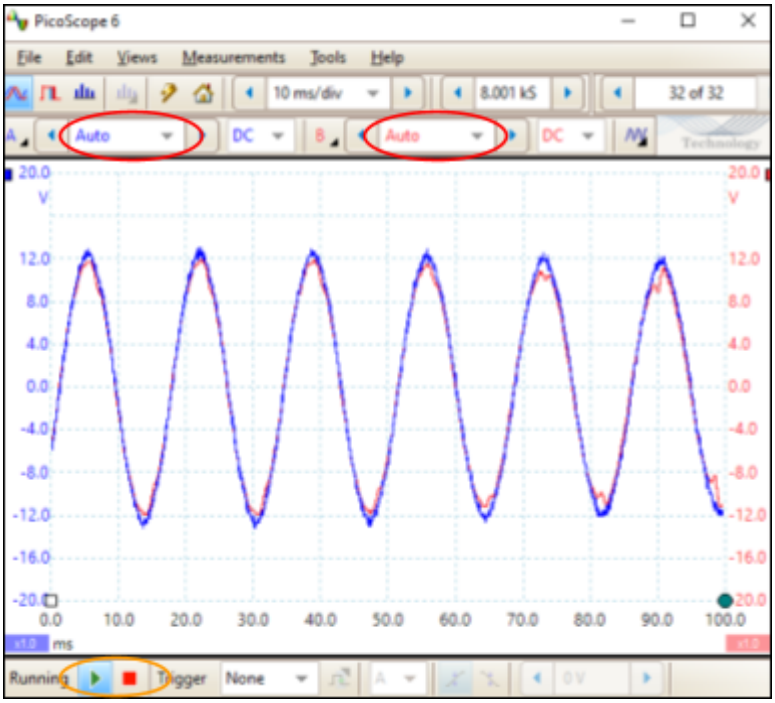

Figure 1: PicoScope 6 with both channels running live. The Channel A waveform is in blue, while B is in red, and their respective axes are color-coded accordingly. The circled dropdown menus should be set to Auto, and the play/pause buttons in the bottom left toggle live acquisition.

4.0 Data Acquisition & App Control

There are several steps required to access and run the data collection GUI code:

1. Open MATLAB
2. Type “appdesigner” in the command window; this will open the App Designer edit space
3. Click Open and select the SmartGrid.mlapp file
4. Once open, “Design View” can change default values for text boxes, and “Code View” allows one to see and edit the code for the app
5. Press the green Run arrow to run the application

<u>Setup Tab</u>

The first tab of the GUI is used to connect to the data acquisition board (DAQ) and test the acquisition. First, plug the DAQ into the laptop and ensure that the blue light next to the cable port is illuminated and solid. The “Connect DAQ” button initiates a connection, and if successful the button will be disabled and the bulb next to it will turn from gray to green, as shown in Figure 2. Once a load configuration is attached to the DAQ inputs, the “Record DAQ” button will collect data for the amount of time specified in the text box. This data is automatically displayed in the graphs below, with a frequency analysis of each channel. The frequency analysis serves as confirmation that the operating frequency of the grid is correct at 60 Hz.

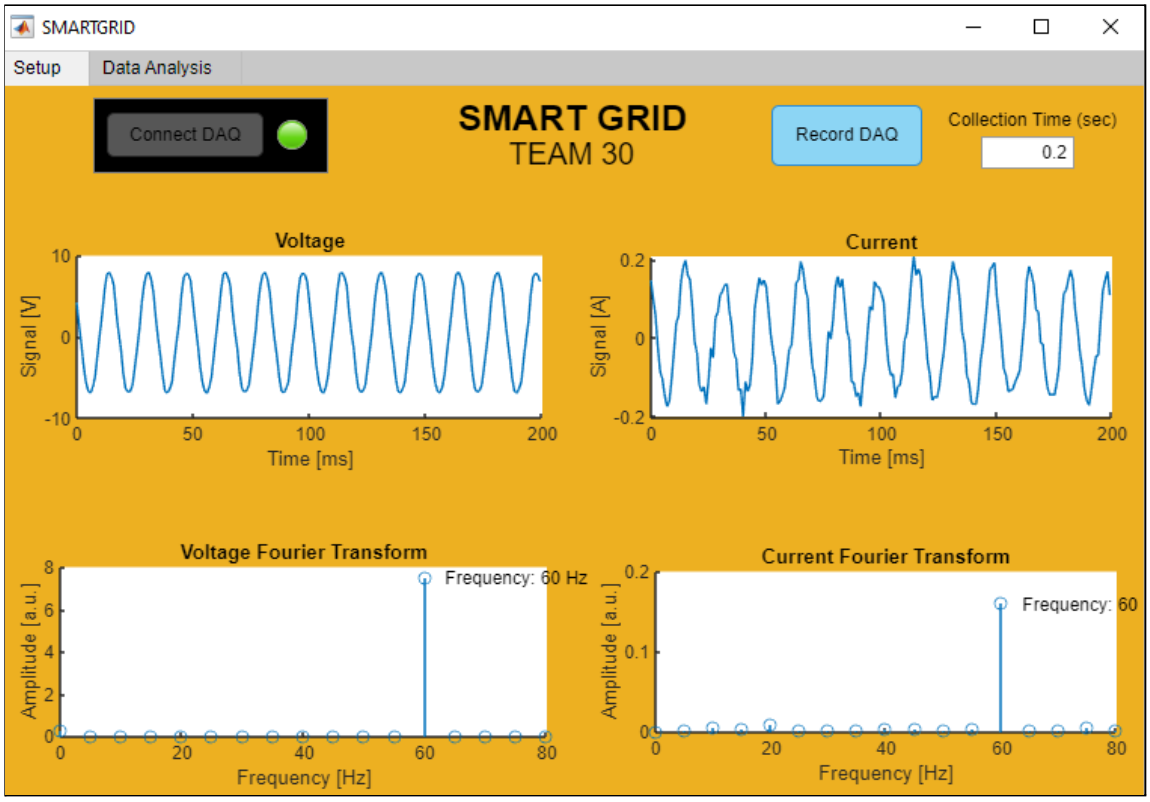

Figure 2: Setup tab of DAQ interface

Data Analysis Tab

The next tab of the GUI is where data is analyzed for phase, power factor, and RMS magnitude. The "Record DAQ" button works the same as in the Setup tab, with the time that data will be collected over specified in the text box below. After data has been collected, it will automatically plot in the Voltage and Current axes. If desired, the waveforms shown in these axes can be saved by pressing the "Save Data" button. When the button is pressed, it will bring the user to a file explorer, where a filename can be input; the data is saved to this file as a '.mat' file.

After data has been collected and plotted in the axes, the "Phaser Calculation" button calculates and displays the polar coordinates of both voltage and current phasers. This is just converting the time signal to an amplitude and phase, with voltage as the reference; therefore, the angle of the voltage signal is always zero, and the plotted angle of the current signal is the same as the value displayed in the "Phase (degrees)" box. Thus, the angle between the two phasers, the resulting power factor, and the respective RMS magnitudes are displayed in the boxes below the phase diagram.

After switching the load configuration, the user can collect the new data by again pressing "Record DAQ" and subsequently "Phaser Calculation" buttons in order. This order must be repeated for each new set of data taken; collect data, then perform analysis.

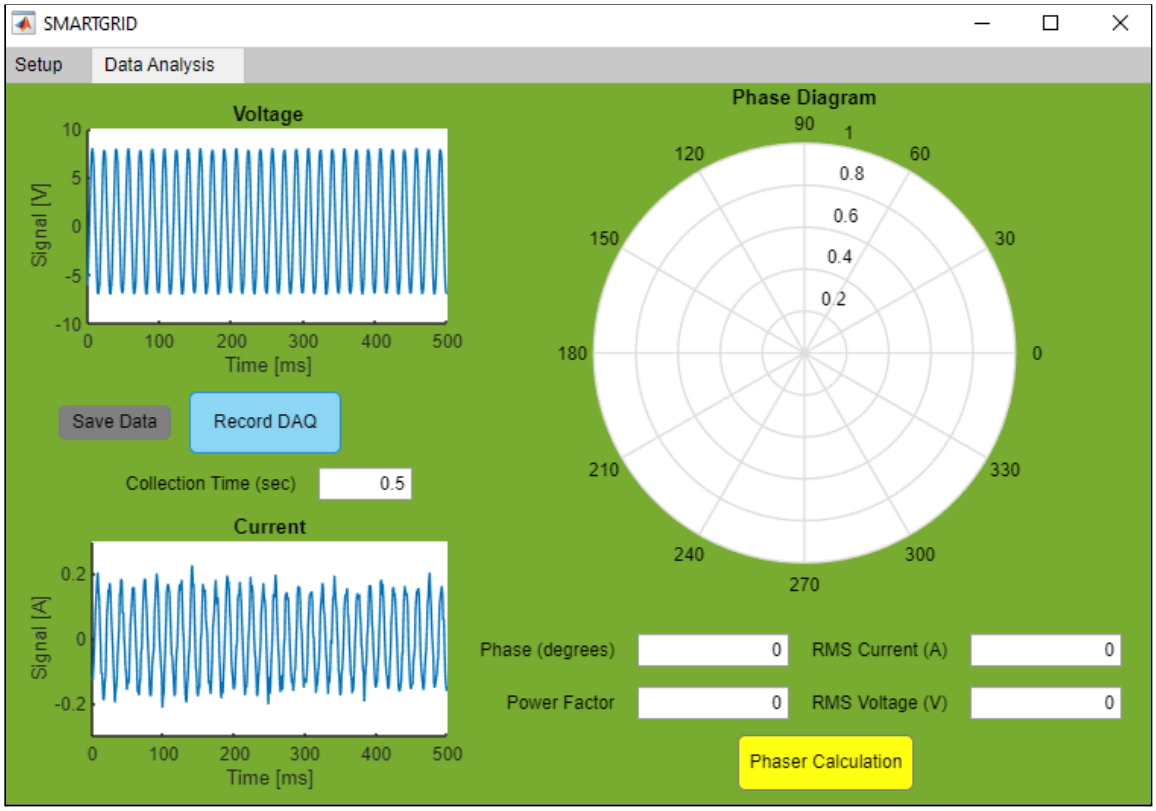

Figure 3: Data Analysis tab of DAQ interface

4.0 Troubleshooting Common Issues

With all proper software packages installed, the software components may run into various issues in operation; the following sections are common errors and how to avoid or resolve them.

DAQ Fails to Connect

If the "Connect DAQ" button returns an error and connection to the device fails, there are several possible solutions. In some cases, the main cause may be that the DAQ is not being supplied enough power from the computer, and this is indicated by the blue LED on the DAQ flickering while connected to the computer. If the power is stable, it will be solid.

If the LED is lit but there is still a MATLAB error from connecting to the DAQ object, here are some other basic things to try as well:

- Unplug the USB cord, plug back into the computer, and try running the app again.
- Close and restart MATLAB to reset the instrumentation.
- Completely restart the computer.

DAQ Changes Device Name

Currently, the DAQ object that the code creates labels the device as ‘myDAQ2’. We have experienced the DAQ change this device name on its own, for unknown reasons. If this occurs, MATLAB should tell you the name of the available devices in the error popup. If not provided, the user can type “daq(‘ni’)” in the command window to view the available daq device properties. The correct name should be updated in the Code View “ConnectDAQPushed” callback function.

No Buttons Work or GUI Won’t Close

A common mistake with App Designer applications is that the app itself will lose access to the functions if the MATLAB working folder is changed. If MATLAB is unable to see a function or there is an access issue, make sure the .mlapp file is in the current working folder.