**Software Design Document**

User Interface sections are presented. The Graphical User Interface (GUI)

* The OptionMenu
* The Label
* The Event List
* Entry
* Buttons

4.1 The Menus

This section of the GUI is simply the menu system our program will be using. There are a number of

menu items, as detailed below, which the user will have access to. For simplicity, we have grouped

similar items under the one menu in a similar way to many other Windows applications. This will

increase the intuitiveness of the GUI and allow the the user to ﬁnd the desired item quickly and

easily. A detailed description of the menu items and their functions follows:

4.2 The Toolbar

This section of the GUI contains various buttons, combo boxes and text displays which enable

control of the program and give the user information.

4.3 The Event List and Display

This section of the GUI displays the Events on a scrolling graph and allows the user to restrict which

Events are to be displayed.

4.3.1 The Event List

All possible events for display (as deﬁned in the appropriate conﬁg ﬁle) are listed here in a CheckBox

list. The list is horizontally opposite the Graph Display, as shown in the diagram below. The vertical

position of a speciﬁc Event corresponds to the vertical position of the Event on the Graph Display.

The user can select which events to display by checking the appropriate CheckBoxes.

4.3.2 The Graph Display

This is where the simulation is graphically represented after recording or loading. The x-axis is a time

scale (as shown across the top of the display). The y-axis is the Events, as described in the previous

section. To select a speciﬁc area of the graph for zooming purposes, the user will follow these steps:

• Left Click on the desired start time, which will be displayed in the From display on the Toolbar.

• Right Click on the desired end time, which will be displayed in the To display on the Toolbar.

• Click the Zoom Region button on the toolbar, which will cause the Graph Display to zoom as

appropriate.

• The user can then either select a new region to zoom, or can restore the default view by clicking

the Zoom Full button on the Toolbar.

Class diagram

This diagram uses UML to represent the classes.

 Function template

Reset()

Purpose: Reset widgets related to the User Interface.

Prototype: def reset()

Inputs: None.

Outputs: None.

Restrictions: None.

Called by: Reset button click and events

Algorithm: –

• Reset the Type of measurement option menu to “Voltage – AC”.

• Reset the Range option menu to 0.

• Reset the signal frequency text box to 0.

• Reset the Amplitude text box to 0

• Reset the Sampling frequency text box to 0

2.13.2 setGUI

Purpose: To set the User Interface that the Error will display messages to.

Prototype: public static void setGUI( Form main )

Inputs: A Windows form to display error dialogs to.

Outputs: None.

Restrictions: None.

Called by: –

• MainEntry.Main()

• Error.Init()

Calls: None

SOFTWARE REQUIREMENTS: -

Anaconda



Anaconda is a open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.

Visual Studio Code



 There are majorly two main files namely:

 Master

 GUI

 Master handles the data processing part and delivers the required data(signals)

from the GUI to the DSP (Digital Signal Processing) and vice versa.

 GUI handles the user interface part where it takes the input from the user and

provides the required output.

 The communication between the Master and the GUI is through the IPC (Inter

Process Communication).

 Any communication to the GUI happens through the Master

only.

 The Master communicates with DSP using the SPI (Serial peripheral Interface).

 Application runs as a Linux process with multiple threads under the Linux

Scheduler. There are 9 threads with some priority to execute various functionalities.