

# PYTHON BASED SIGNALS AND FUNCTION ANALYZER USING GRAPHICAL USER INTERFACE

**BY:** Gedeon MUHAWENAYO, Hirwa CHRISTIAN, Sylvia MUTESI

**EMAIL:** [gedeonmuhawenayo@gmail.com](mailto:gedeonmuhawenayo@gmail.com)

**TEL:** +250 788 365 398

**SUPERVISOR:** Emmanuel HITIYISE

## ABSTRACT

Python is a powerful programming language for handling engineering and scientific computational tasks efficiently [1]. It offers an extremely wide selection of tools for tasks such as scientific computing, signal processing, Web site construction, database programming, and graphical user interface (GUI) design. The language is platform independent with most programs running on Linux, Microsoft Windows, or MAC OS virtually unchanged [2]. In signal and functions analysis, python can be used for real time analysis and plot of signals. These days' people mainly students in the engineering field have a problem of poor understanding of signals and functions behaviors because they use methods that do not provide away to interact with them. As a solution, exploiting the power of python programming language, we have designed a graphical user interface to be used for real time plotting of signals and functions as well, this would be a better way of providing good understanding of signals behavior so the special things in our project it is possible to acquire input signal from any circuit to see all the parameters related to it. We have made an executable file that could be shared and installed in windows computer, since every student can afford to have his/her own computer. Furthermore, the UI is user friendly, and it has a menu bar containing user guide, and all output sketches are real time plotted either in 2D or 3D to make it more understandable.

# TABLE OF CONTENTS

ABSTRACT .....	i
TABLE OF CONTENTS .....	ii
LIST OF ABBREVIATIONS AND FIGURES .....	iv
LIST OF TABLES.....	v
LIST OF ABBREVIATIONS.....	vi
1 GENERAL INTRODUCTION .....	7
1.1 BACKGROUND .....	7
1.2 PROBLEM STATEMENT.....	7
1.3 SCOPE OF THE PROJECT.....	8
1.4 PROJECT GOAL AND OBJECTIVES.....	8
1.4.1 MAIN OBJECTIVE.....	8
1.4.2 SPECIFIC OBJECTIVE .....	8
2 LITERATURE REVIEW .....	9
2.1 INTRODUCTION TO PYTHON PROGRAMMING LANGUAGES .....	9
2.1.1 COMPANY WHICH USE PYTHON.....	10
2.2 FEATURES OF PYTHON .....	10
2.3 ADVANTAGES OF PYTHON .....	10
2.4 PYTHON APPLICATION .....	11
2.5 PYTHON BASIC SYNTAX.....	11
2.5.1 PYTHON KEYWORDS.....	11
2.5.2 COMMENTS IN PYTHON.....	11
2.5.3 VARIABLE TYPES .....	12
2.5.4 PYTHON LISTS .....	12
2.5.5 PYTHON BASIC OPERATORS.....	12
2.5.6 DECISION MAKING.....	13
2.5.7 FUNCTION DEFINITIONS AND FUNCTION CALLS .....	13
2.6 GRAPHICAL USER INTERFACE.....	15
2.6.1 INTRODUCTION.....	15
2.6.2 DESIGNING USER INTERFACES.....	15
2.6.3 PYTHON GUI PROGRAMMING .....	16
2.6.4 TKINTER PROGRAMMING .....	16
2.6.5 TKINTER WIDGETS.....	17
2.6.6 TYPES OF WIDGETS IN TKINTER .....	17
3 RESEARCH METHODOLOGY .....	18
3.1 FLOWCHART OF THE SYSTEM .....	18

4 EXPERIMENTS, RESULTS AND DISCUSSION ..... 19

4.1 INTRODUCTION ..... 19

4.2 UI DETAILS ..... 19

4.2.1 LOGIC BEHIND BUILT IN FUNCTIONS ..... 20

4.2.2 LOGIC BEHIND USER DEFINED FUNCTIONS ..... 21

4.2.3 MENU OF THE USER INTERFACE ..... 22

4.3 BUILDING OF THE GUI ..... 22

4.4 EXPERIMENT AND RESULTS ..... 22

4.4.1 INSERTING EQUATION ..... 22

4.4.2 BUILT IN FUNCTION ..... 23

4.4.3 ACQUIRING SERIAL SIGNAL ..... 25

5 CONCLUSIONS, RECOMMENDATIONS AND FUTURE SCOPE ..... 1

5.1 CONCLUSION ..... 1

5.2 FUTURE SCOPE ..... 2

6 REFERENCES ..... 3

7 APPENDIX ..... 4

# LIST OF FIGURES

figure 1. number of company uses python[16] .....	15
figure 2. application of python[9] .....	16
figure 3. Function call and function definition .....	19
figure 4. creating a class in python .....	19
figure 5. TKinter widgets .....	21
figure 6. block diagram of the system .....	23
figure 7. GUI meu .....	24
figure 8. logic behind GUI .....	25
figure 9. idea behind user defined function .....	26
figure 10. usable libraries .....	27
figure 11. Output results of quadratic equation .....	28
figure 12. output of cosine function .....	29
figure 13. dumping oscillation .....	29
figure 14. three phase current .....	30
figure 15. Sinc function output .....	30
figure 16. Acquiring output signal .....	31
figure 17. Acquiring serial Data input .....	31
figure 18. hardware of reading digital signal.....	32
figure 19. result of reading hardware signal .....	32
figure 20. reading analog signal .....	33
figure 21. analog signal plot .....	33

## LIST OF TABLES

Table 1. python keywords [4] .....	16
Table 2. decision making statement [8] .....	18
Table 3. Types of widget[15] .....	22

## **LIST OF ABBREVIATIONS**

**GUI:** Graphical User Interface

**I/O:** Input output

**COM:** communication

**PC:** Personal computer

**UR:** University of Rwanda

**EEE:** Electrical and Electronics Engineering

**UI:** User Interface

**IDE:** Integrated development environment

**UR-CST:** University of Rwanda College of Science and Technology.

# **1 GENERAL INTRODUCTION**

## **1.1 BACKGROUND**

Rwanda and other African countries are advancing in science and technology, due to this reason studying and understanding well the science theories and principles is very critical. Science and technology are very essential in our daily life using them we can solve most of the social issues. Therefore, better understanding of science and engineering principles is the key. The analysis of different signals and functions is also based on different principles and theories which were set by scientists, in this world science evolves every day and other scientists discover more and more day by day [3]. because of this reason many things like signal and functions analysis becomes more complex and difficult to learners and students in general. Many students in the field of engineering or science in general learn about mathematical function based on plotting graphs or characteristics of different devices or systems but it becomes hard to obtain the results or output of the system. We as the student in the field of electronics and electrical engineering have faced the same problem. Most of the times people learn theory without getting the knowledge about the system behavior and characteristics of the signal and functions. So our project called python based function analysis will be able to plot most functions (functions with not more than three variables) which will help academic students to know the systems' behavior and also the analysis will be real time so that it becomes more understandable compared to the past years. Additionally, our project will help people as researchers or the people who want to learn more about python, how it works, because will be provided with a graphical user interface(GIU) where a user is able to interact with the system. This project requires programming skills, function and signal analysis skills. It also requires us to know about the mathematical expressions and equations so that we understand well what we are doing before providing this Graphical User Interface (GUI). As engineering students we got inspired from our daily's courses where we need to plot more function characteristics like amplitude, phase and frequency that is why our project will be mostly used in electrical and electronics engineering field. Hopefully, this project will help students to get better understanding of signals and functions.

## **1.2 PROBLEM STATEMENT**

In our current world of engineering and technology we need to look system behavior, understand signal behavior and to know real time changes of signal. It is quite necessary to find a way of making it easier for anyone who wish to understand signals and functions. This is what our project will do. It will help engineering students and researchers to plot signals in real time manner. We have seen programming languages that help to plot and analyze different functions and signals like matlab but our project will help the user to analyze them in real time manner and also it will also be able to acquire signal from input; this

means that any put like voltage or current connected to the computer, it will analyze and plot it in real time manner.

### **1.3 SCOPE OF THE PROJECT**

Due to our academic background and the level of contribution that we wish our project to give to all students we implemented our project, the specific goal is that we will develop a graphical user interface (GUI) which will help a user to interact with the system and be able to insert or remove the library and also provide a setup (executable file) which can be installed in windows computers which will help students to share it and help each other. Specifically, this GUI will be dedicated to every student who wish to use this interface in his everyday education life to get a better understanding about most of phenomenon in electrical, electronics and telecommunication engineering fields. This project because it is python based and we have not studied that, we have a task of learning it and also implement the project, so it will take us much time and other learning costs so that we come to the end of the project, in implementation of our project we will study functions of not more than three variables due to the fact of no having enough python programming skills.

### **1.4 PROJECT GOAL AND OBJECTIVES**

#### **1.4.1 MAIN OBJECTIVE**

The main objective of our project is to build a functional and signal analyzer user interface that can enable peoples specifically students in Electrical and Electronics Engineering field, to get better understanding and analyze the signal or function behaviors by plotting signals with real time behaviors or changes.

#### **1.4.2 SPECIFIC OBJECTIVE**

The specific objective of our project is to Build an interactive graphical user interface (GUI) for users for getting the system behaviors and analyses signal trends to get graphical presentation of the results.

This project won't be implemented everywhere, there are some specific places this will have to be put, like in universities, high school and others place related to sciences and engineering students.



## 2 LITERATURE REVIEW

### 2.1 INTRODUCTION TO PYTHON PROGRAMMING LANGUAGES

Python is a high-level, interpreted and general-purpose dynamic programming language that focuses on code readability. The syntax in Python helps the programmers to do coding in fewer steps as compared to Java or C++. The language founded in the year 1991 by the developer Guido Van Rossum has made the programming easy and fun to do [4]. Python is widely used in bigger organizations because of its multiple programming paradigms. They usually involve imperative and object-oriented functional programming. It has a comprehensive and large standard library that has automatic memory management and dynamic features.

An ever-increasing number of scientific studies are generating larger, more complex, and multi-modal datasets. This results in data analysis tasks becoming more demanding. To help tackle these new challenges, more disciplines now need to incorporate advanced visualization techniques into their standard data processing and analysis methods. While many systems have been developed to allow scientists to explore, analyze, and visualize their data, many of these solutions are domain specific, limiting their scope as general processing tools. One way to enhance their flexibility is to build on top of an interpreted language [6].

Python programming language provides a development environment suitable to both computational and visualization tasks. One of the key advantages of Python is that packages can be used to extend the language to provide advanced capabilities such as array and matrix manipulation, image processing, digital signal processing, and visualization. Several popular data exploration and visualization tools have been built in Python. Python-enabled application would be capable of producing similar results [7].

Scope of Python in today's world is vast. All the major companies today are using Python. It is the most taught language in the world currently and there's a good reason for it. It is one of the fastest growing languages and has undergone a successful span of more than 25 years as far as its adoption is concerned. This success also reveals a promising future scope of python programming language.

In fact, it has been continuously serving as the best programming language for application development, web development, game development, system administration, scientific and numeric computing, GIS and Mapping etc.

### 2.1.1 COMPANY WHICH USE PYTHON

For any new piece of software coming out today, there's a Python API, this is because Python is getting a lot of traction for very good reasons. This is how other companies work on python in our daily life. see below the number of companies using Python:



figure 1. number of company uses python [16]

## 2.2 FEATURES OF PYTHON

The programmers of big companies use Python as it has created a mark for itself in the software development with characteristic features like [7]:

- Interactive
- Interpreted
- Modular
- Dynamic
- Object-oriented
- Portable
- High level
- Extensible in C++ & C

## 2.3 ADVANTAGES OF PYTHON

The Python language has diversified application in the software development companies such as in gaming, web frameworks and applications, language development, prototyping, graphic design applications, and others [5].

**Some of its advantages are:**

- Extensive Support Libraries
- Integration Feature

- Improved Programmer's Productivity
- Productivity

## 2.4 PYTHON APPLICATION

Python is used in many application domains in our daily life as Python is interpreted object-oriented language. Those applications include the following:

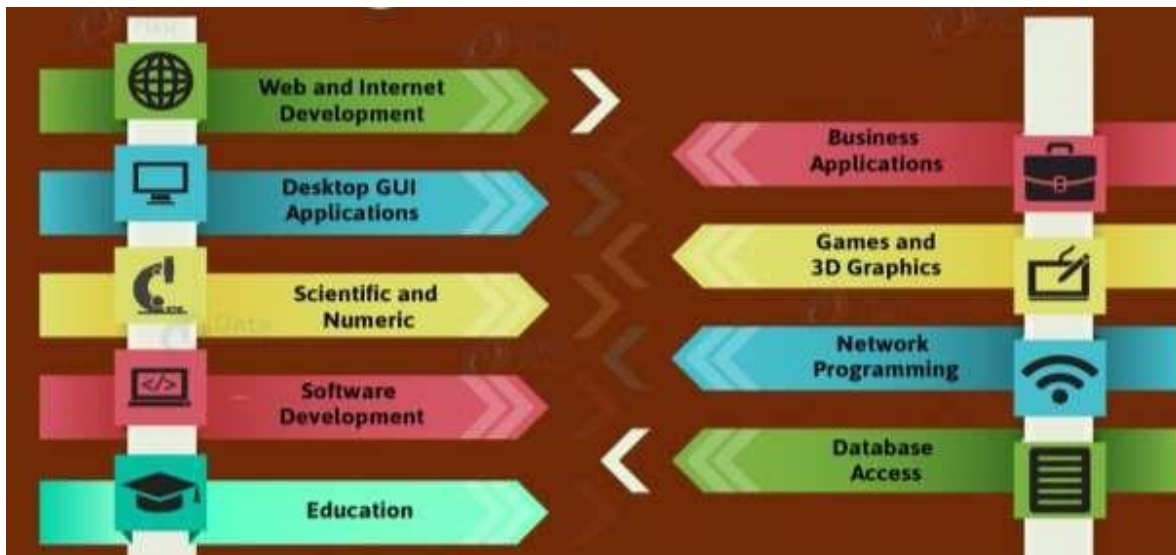


figure 2. application of python [9]

## 2.5 PYTHON BASIC SYNTAX

### 2.5.1 PYTHON KEYWORDS

The following list shows the Python keywords. These are reserved words and you cannot use them as constant or variable or any other identifier names. All the Python keywords contain lowercase letters only.

<b>And</b>	<b>exec</b>	<b>Not</b>
<b>Assert</b>	<b>finally</b>	<b>or</b>
<b>Break</b>	<b>for</b>	<b>pass</b>
<b>Class</b>	<b>from</b>	<b>print</b>
<b>Continue</b>	<b>global</b>	<b>raise</b>
<b>def</b>	<b>if</b>	<b>return</b>
<b>del</b>	<b>import</b>	<b>try</b>
<b>elif</b>	<b>in</b>	<b>while</b>
<b>else</b>	<b>is</b>	<b>with</b>
<b>except</b>	<b>lambda</b>	<b>yield</b>

Table 1. Python keywords [4]

### 2.5.2 COMMENTS IN PYTHON

In Python, there are two ways to annotate your code:

- The first is to include comments that detail or indicate what a section of code (single line).
- The second makes use of multi-line comments or paragraphs that serve as documentation for others reading your code.

Single-line comments are created simply by beginning a line with the hash (#) character, and they are automatically terminated by the end of line. [11]

### 2.5.3 VARIABLE TYPES

Python has five standard data types:

- **Numeric:** Python numeric data type is used to hold numeric values like integer, long, float and complex number
- **String:** The string is a sequence of characters. Python supports Unicode characters. Generally, strings are represented by either single or double quotes.
- **List:** The list is a versatile data type exclusive in Python. In a sense, it is the same as the array in C/C++. But the interesting thing about the list in Python is it can simultaneously hold different types of data.
- **Tuple:** Tuple is another data type which is a sequence of data similar to list. But it is immutable.
- That means data in a tuple is write protected. Data in a tuple is written using parenthesis and commas.
- **Dictionary:** Python Dictionary is an unordered sequence of data of key-value pair form. It is similar to the hash table type. [8]

### 2.5.4 PYTHON LISTS

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]).

Note that all the items belonging to a list can be of different data type.

The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1.[8]

The important characteristics of Python lists are as follows:

- Lists are ordered.
- Lists can be nested to arbitrary depth.
- Lists can contain any arbitrary objects.
- Lists are mutable.
- List elements can be accessed by index.
- Lists are dynamic.

### 2.5.5 PYTHON BASIC OPERATORS

Python language supports the following types of operators:

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

## 2.5.6 DECISION MAKING

Decision making is anticipation of conditions occurring while execution of the program and specifying actions taken according to the conditions. Decision structures evaluate multiple expressions which produce TRUE or FALSE as outcome.

Statement	Description
if statements	<b>if statement</b> consists of a Boolean expression followed by one or more statements.
if...else statements	<b>if statement</b> can be followed by an optional <b>else statement</b> , which executes when the Boolean expression is FALSE.
nested if statements	You can use one <b>if</b> or <b>else if</b> statement inside another <b>if</b> or <b>else if</b> statement(s).

Table 2. decision making statement [8]

## 2.5.7 FUNCTION DEFINITIONS AND FUNCTION CALLS

Function in Python are declared using the def. keyword.

These are the basic steps in writing user defined function in python:

**Step 1:** declare the function with the keyword def. followed by the function name

**Step 2:** write the arguments inside the opening and closing parenthesis of the function and end the declaration with a colon

**Step 3:** add program statement to be executed.

**Step 4:** end the function with or without return statement

The basic syntax is as follows:

**def** func name(arg1, arg2, . . . , argN):

statements

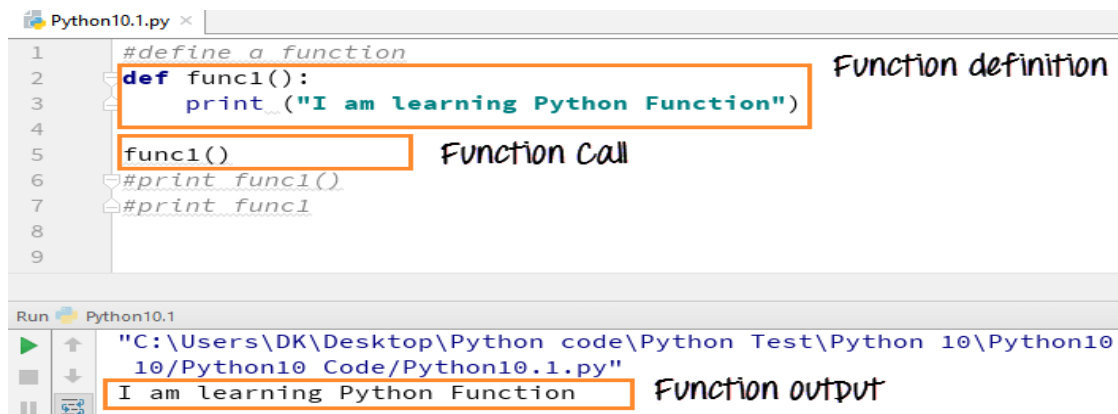


figure 3. Function call and function definition

Since the language is untyped, no function definition has a return type, and parameters are simply listed. If a function is to return a value, the keyword `return` can be used to return it. The same keyword used without a value will stop the execution of the function.

It is also possible to combine the two methods in the same function call, provided that all arguments following the first one that is passed by name are also passed by name also the arguments can receive a default value, in which case no value needs to be explicitly specified for the argument. If a value is indeed specified, it replaces the default value for the argument. Default values are simply assigned to the arguments in the function declaration. [15]

## 2.5.8 CREATING CLASSES

The class statement creates a new class definition. The name of the class immediately follows the keyword `class` followed by a colon as follows:

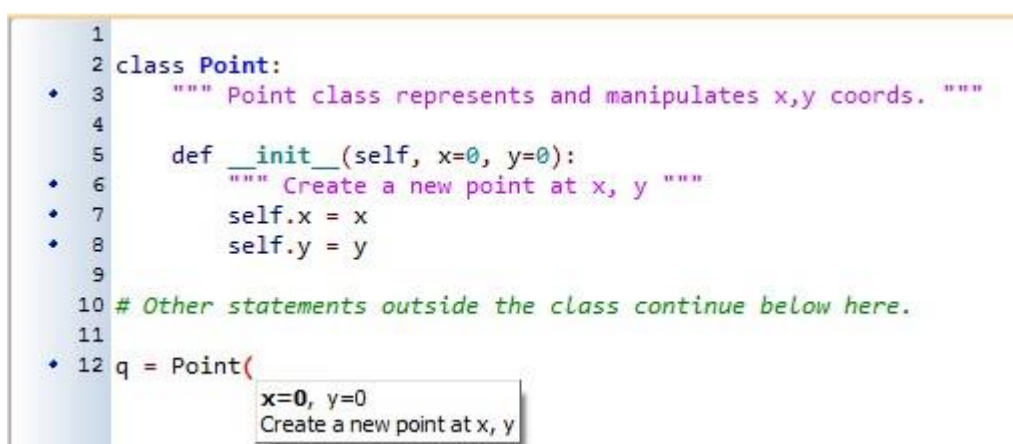


figure 4. creating a class in python

Python has no access specifier. By default, all methods declared within a class are public. Simulating private methods is still possible by using the same name mangling feature as it was done for variables. This is done

by preceding the method name by two or more underscores. Inside a class, the self-keyword refers to the current instance of this class.

## **2.6 GRAPHICAL USER INTERFACE**

### **2.6.1 INTRODUCTION**

A **graphical user interface (GUI)** is an interface through which a user interacts with electronic devices such as computers, hand-held devices and other appliances. This interface uses icons, menus and other visual indicator (graphics) representations to display information and related user controls. GUI representations are manipulated by a pointing device such as a mouse, trackball, stylus, or a finger on a touch screen. In this regards, we are developing a GUI in python language with the help of GUI module called Tkinter. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. [14]

### **2.6.2 DESIGNING USER INTERFACES**

User interfaces are what allows end users to interact with an application. An application can be excellent, but without a good user interface, it becomes more difficult to use, and less enjoyable. It is thus very important to design good user interfaces.

Designing user interface takes place at two different levels:

- the graphical level
- the event level.

Graphical elements of a user interface are called widgets. Widgets are basic components like buttons, scrollbars, etc. But user interfaces involve more than a collection of widgets placed in a window. The application must be able to respond to mouse clicks, keyboard actions or system events such as minimizing the window. For this to happen, events must be associated to some pieces of code. This process is called binding. [15]

## 2.6.3 PYTHON GUI PROGRAMMING

Python provides various options for developing graphical user interfaces (GUIs). Most important are listed below:

- **Tkinter:** Tkinter is the Python interface to the Tk GUI toolkit shipped with Python.
- **wxPython:** This is an open-source Python interface for wx Windows.
- **JPython:** JPython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine.

Nearly every GUI program we will write will contain the following three lines:

```
from tkinter import *
```

```
root = Tk()
```

```
mainloop()
```

- The first line imports all of the GUI stuff from the tkinter module.
- The second line creates a window on the screen, which we call root.
- The third line puts the program into what is essentially a long-running while loop called the event loop. This loop runs, waiting for keypresses, button clicks, etc., and it exits when the user closes the window [13].

## 2.6.4 TKINTER PROGRAMMING

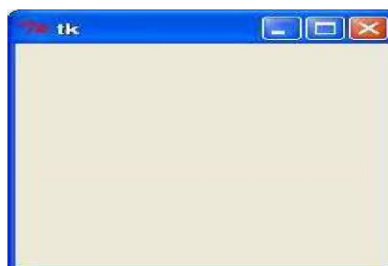
Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps:

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user. [14]

### Example

```
import Tkinter
top = Tkinter.Tk()
# Code to add widgets will go here...
top.mainloop()
```

**results:**



**figure 5.** TKinter widgets



## 2.6.5 TKINTER WIDGETS

A widget is an element of a graphical user interface (GUI) that displays information or provides a specific way for a user to interact with the operating system or an application.

A widget is therefore a graphical object that is available from the Tkinter library. It is a kind of graphical building block. Intuitively, widgets are implemented as classes in Tkinter. Each widget therefore has a constructor, a destructor, its own set of properties and methods, and so on. While most other GUI toolkits have a very complex widget hierarchy, Tkinter's hierarchy is extremely simple.

All widgets (like Button, Check button, etc.) are derived from the Widget class. All widget subclasses occupy the same level in the hierarchy tree.

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application.

These controls are commonly called widgets. [14]

## 2.6.6 TYPES OF WIDGETS IN TKINTER

We present these widgets as well as a brief description in the following table:

OPERATOR	DESCRIPTION
<b>Button</b>	The Button widget is used to display buttons in your application.
<b>Canvas</b>	The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
<b>Check button</b>	The Check button widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
<b>Entry</b>	The Entry widget is used to display a single-line text field for accepting values from a user.
<b>Frame</b>	The Frame widget is used as a container widget to organize other widgets.
<b>Listbox</b>	The List box widget is used to provide a list of options to a user.
<b>Menubutton</b>	The Menu button widget is used to display menus in your application.
<b>Message</b>	The Message widget is used to display multiline text fields for accepting values from a user.
<b>Scrollbar</b>	The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
<b>Text</b>	The Text widget is used to display text in multiple lines.
<b>Toplevel</b>	Top level The Toplevel widget is used to provide a separate window container.
<b>Labelframe</b>	A label frame is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.
<b>tkmessage</b>	This module is used to display message boxes in your applications.

Table 3. Types of widget[15]

### 3 RESEARCH METHODOLOGY

The research methodology is the system being used to solve a research problem. This involves methods and technics that are being used to gather information to carry out the analysis as well as implementation of our project. In this research we have searched information from books, journals and exchanging idea with supervisor and colleagues. This has been helping us in this project to get desired ideas about the current signals and functions analyzer systems being used in Rwanda and worldwide like spectrum analyzer, and MATLAB software.

For analyzing our project system, we use documentation technique, which is done by consulting, and analyzing different types of documents related to python based on graphical user interface. In addition, we applied web research for enquiring all related information to the proposed system. Python programming software is used for analyzing, concluding and code writing for the system performance, which gives all required characteristics of system behaviors and analysis.

#### 3.1 FLOWCHART OF THE SYSTEM

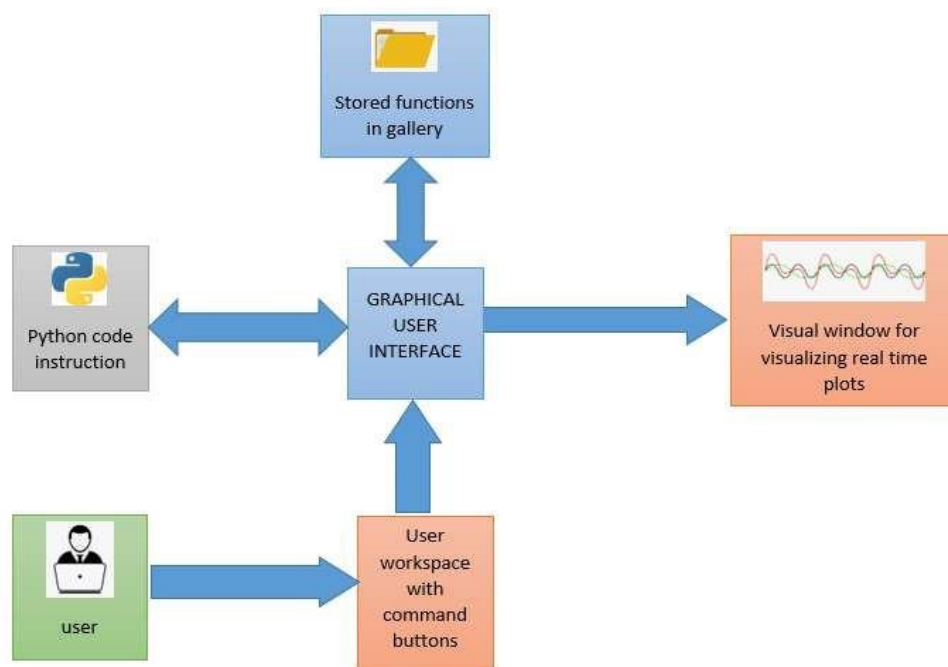


figure 6. block diagram of the system

In our project, we have used python languages to implement algorithms, user can access our application using the workspace where they choose the types of equation that he/she wants to understand its behavior like quadratic equation, sinusoidal wave equation and others so when the equation is entered, you can place PLOT button. This time our python code executes your function and provides a visual output plot. Furthermore, user can save results for future use. Apart from that there is a gallery of some popular functions which can be accessed too.

## 4 EXPERIMENTS, RESULTS AND DISCUSSION

### 4.1 INTRODUCTION

This part will help to understand well our project called **python based signal and function analyzer using graphical user interface**". It will explain more about the methods that we used during the simulations and implementations, it will also provide a block diagram as well as flowcharts that will show well how our project will work in good manner and provide the expectable results and some of the results will be shown as photos(screenshots) in this chapter.

### 4.2 UI DETAILS

Our project is a graphical user interface which works in windows computer, its main purpose is to help users to analyze signals and functions. The GUI has three main parts; Built in functions, User defined functions and Menu.

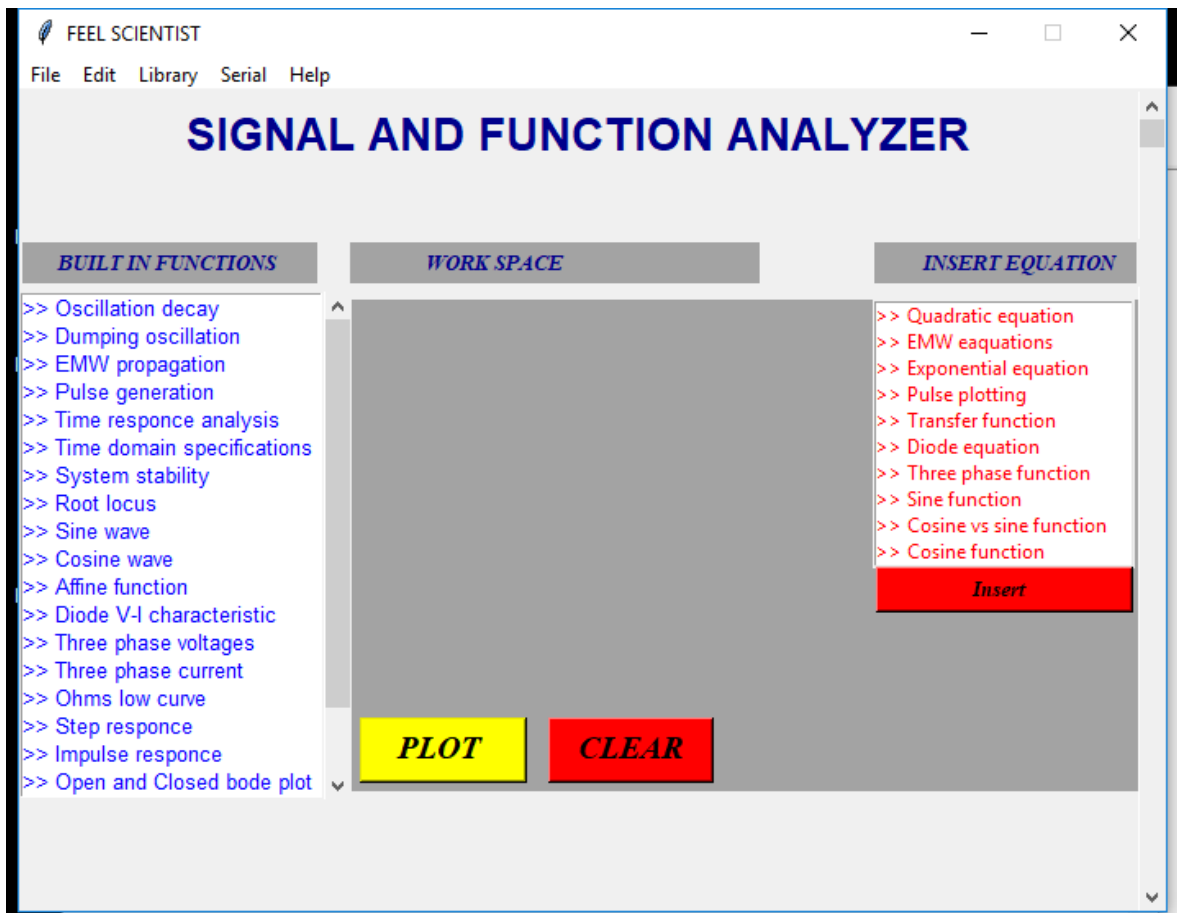


figure 7. GUI menu

#### 4.2.1 LOGIC BEHIND BUILT IN FUNCTIONS

This is a list of some of most common signals and functions in the field of engineering specifically in EEE department. These functions can be accessed just by selecting one preferred and press the plot button.

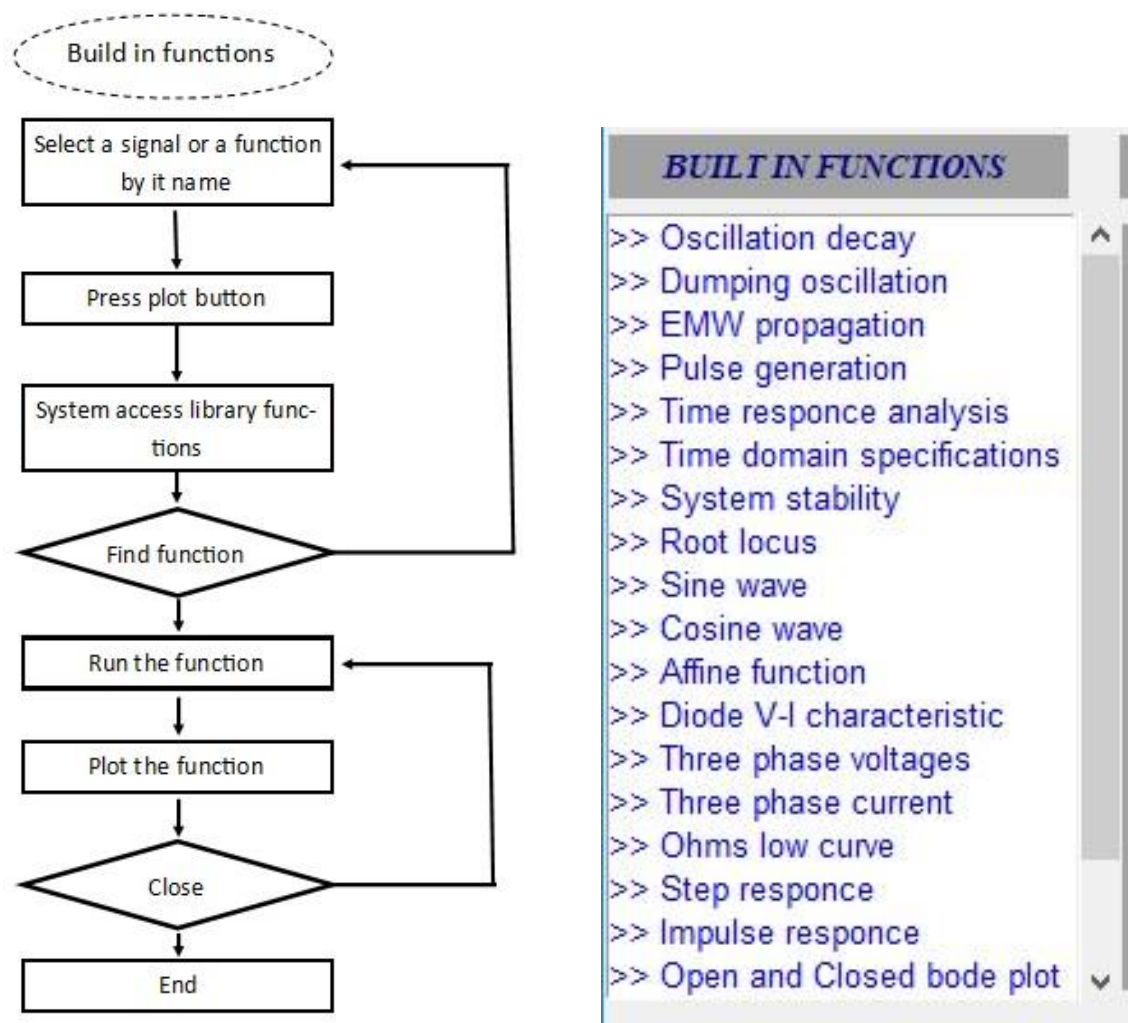


figure 8. logic behind GUI

### 4.2.2 LOGIC BEHIND USER DEFINED FUNCTIONS

This enables user to enter a function that he wishes to plot, it contains a list of some of most common signals and functions in the field of engineering specifically in EEE department. After selecting a preferred function or signal type, user press the insert. This will open a new widget to enter parameters or function coefficients. Afterwards you just press PLOT button then your function will be plotted, for further analysis you can enter different parameters and again plot.

This time you will be able to compare all your functions and any conclusion can be drawn.

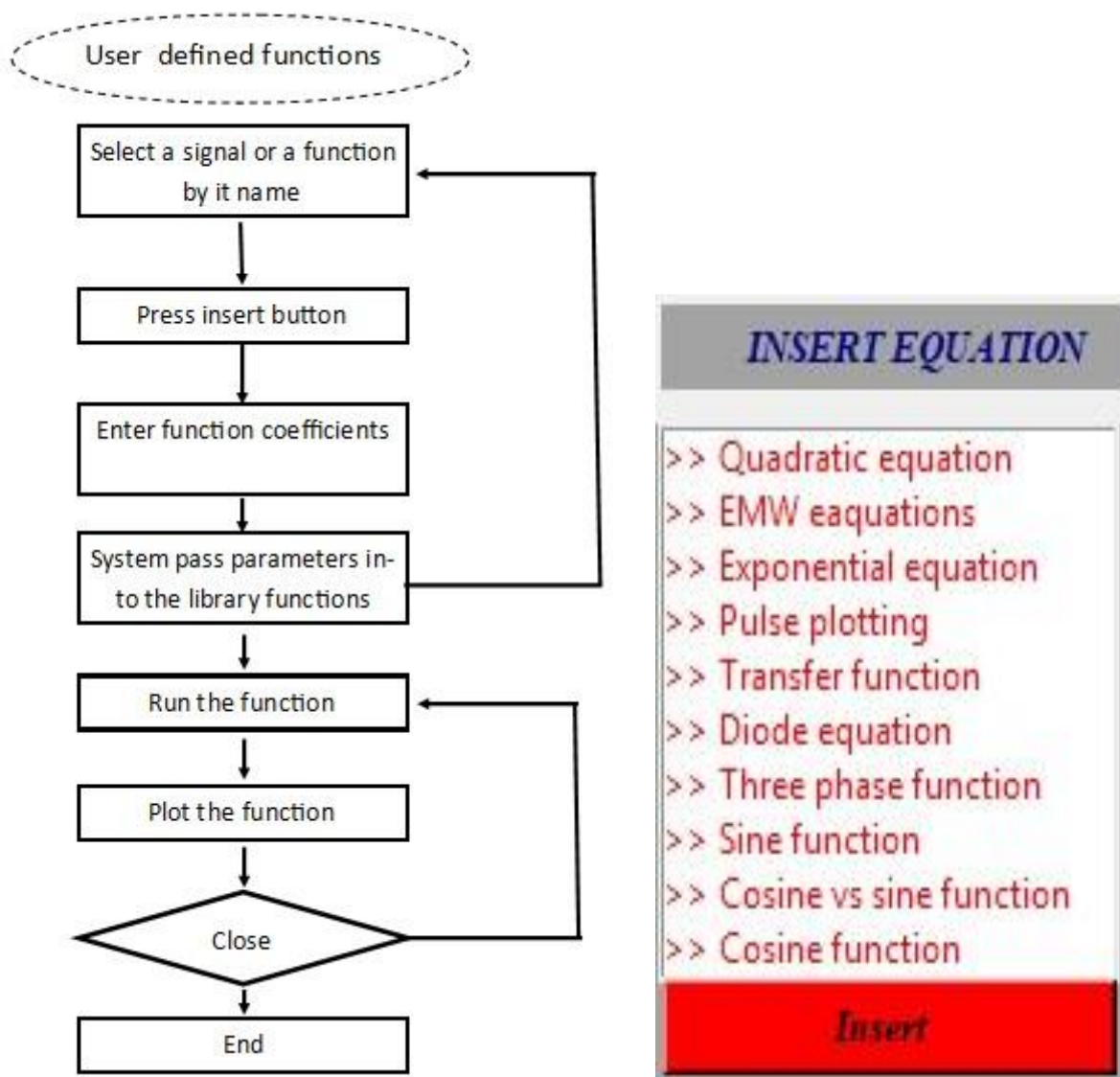


figure 9. idea behind user defined function

### 4.2.3 MENU OF THE USER INTERFACE

## 4.3 BUILDING OF THE GUI

As we are using python programming to develop our system, first thing we have done was to download and install python software from <https://www.python.org/downloads/> and learn some basics as well as most popular libraries used in signal and functions analysis.

In our case we are using **matplotlib**, **numpy**, **tkinter**, **pyautogui** modules and others according to the task. Furthermore, we have developed our own modules in another to achieve our goals, here is the list of modules we have developed; gallery function, control function and user defined function.

```
#used modules
import tkinter #gui module
import pyautogui #screenshot
import time
import uuid #random string generator

# _____ MATH AND PLOT LIBRAY _____
import numpy as np
from matplotlib.lines import Line2D
import matplotlib.pyplot as plt
import matplotlib.animation as animation

import numpy
from matplotlib.pylab import *
from mpl_toolkits.axes_grid1 import host_subplot
import matplotlib.animation as animation

from tkinter import *
from tkinter import messagebox as tkMessageBox

# _____ OUR DEFINED LIBRARY _____

import garelly_functions
import control_functions
import user_defined_functions
```

figure 10. usable libraries

## 4.4 EXPERIMENT AND RESULTS

To explain clearly our work lets, go through some of experiments.

### 4.4.1 INSERTING EQUATION

For inserting function, you click on the type of equation you want to enter and the insert button. This will open a new widget where you will enter equation coefficients.

- **Polynomial equation**

It should be noted that our UI is limited to the polynomial equation of third order.

Once you fill all the coefficient, there is plot button under the coefficients you can press on it to see the results and you can change the coefficient according to your equation.

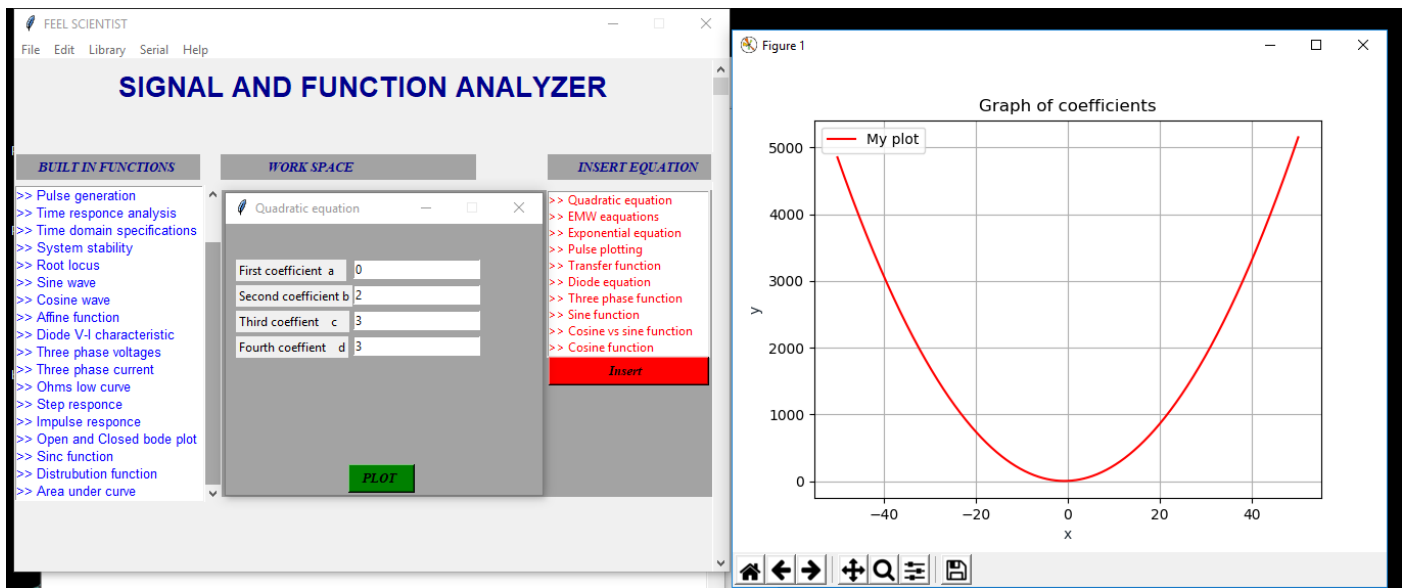


figure 11. Output results of Polynomial equation

- **Cosine signal**

Now we want to plot cosine function which have amplitude, phase and frequency and the result is shown below and the step of inserting parameters is the same as what we have done on the above results.

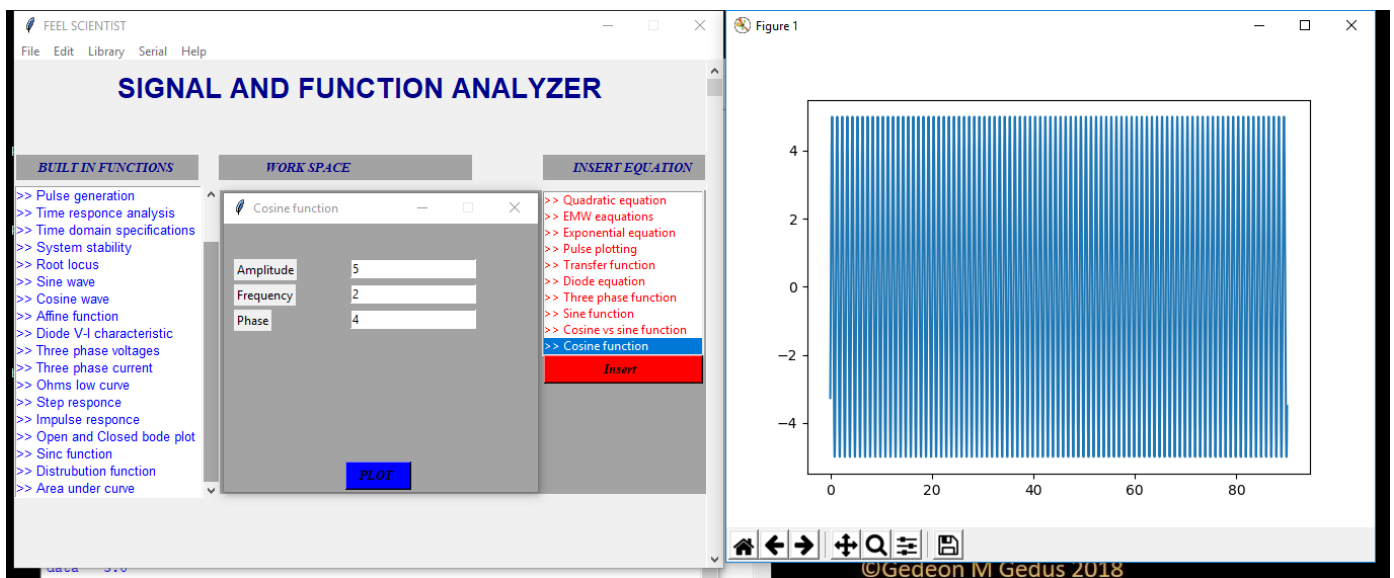


figure 12. output of cosine function

#### 4.4.2 BUILT IN FUNCTION

Built in function contain many function we need in our daily life, so it is stored in the gallery. If you want to see the behavior of any signal stored in it is very easy to access by clicking on it and PLOT. in our gallery we have many functions like dumping function, oscillation decay, sine wave, cosine wave and others



- **Dumping oscillation**

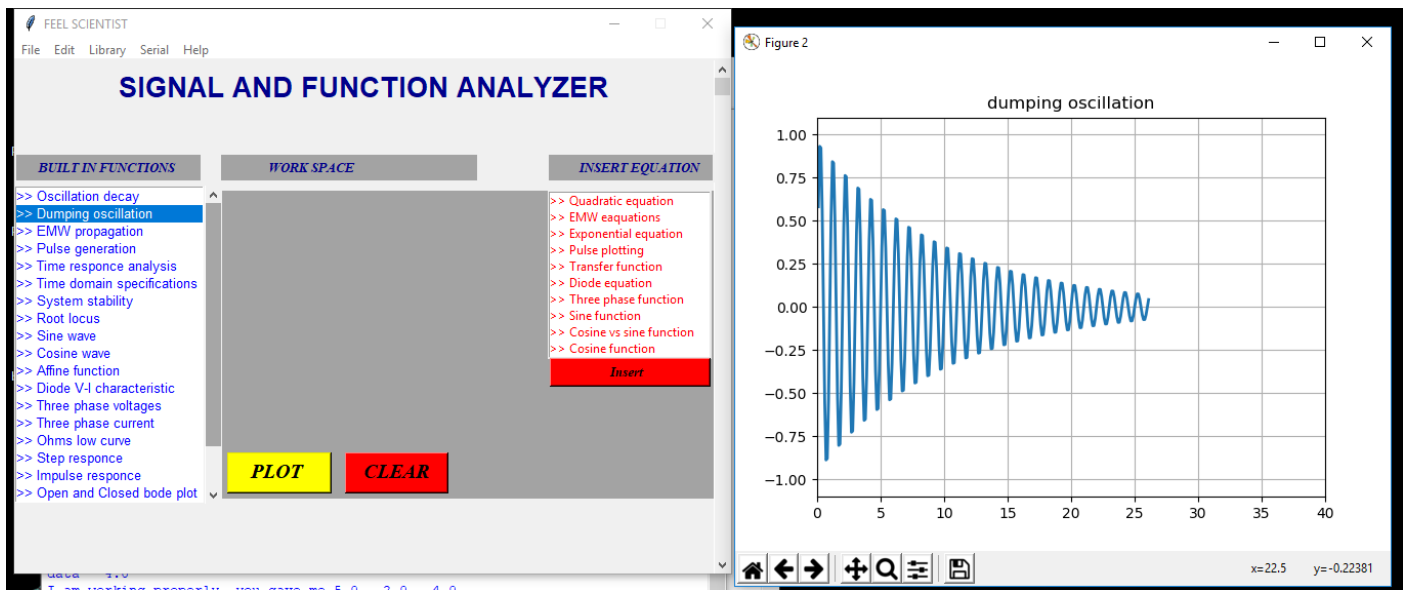


figure 13. dumping oscillation

- **Three phase current**

With our UI it is easy to study behaviors of three phase signals representing either voltage or current.

Here is the image elaborating it.

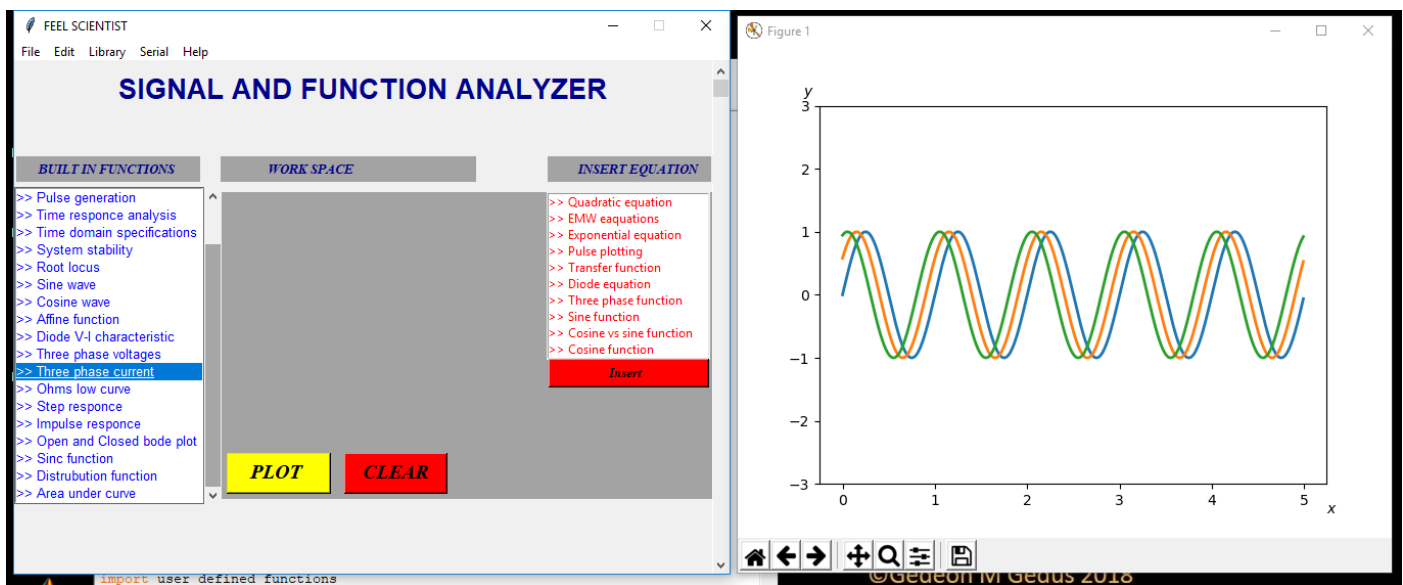


figure 14. three phase current



- **Sinc function**

It is used in the concept of reconstructing a continuous bandlimited signal from uniformly spaced samples of that signal.

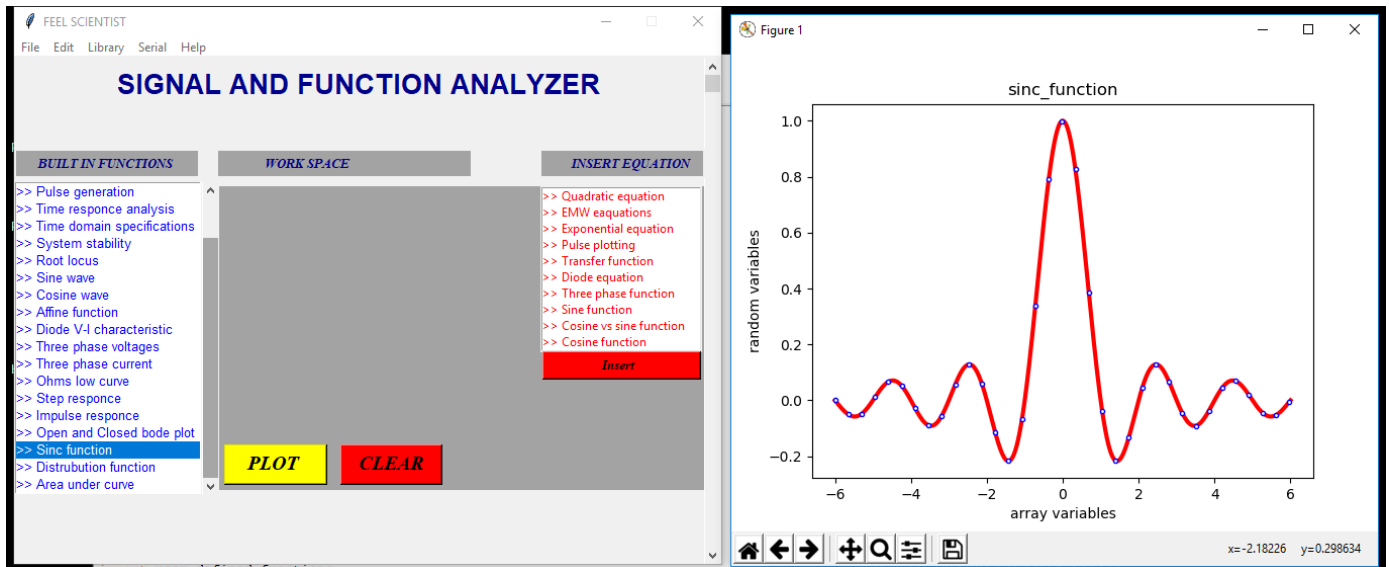
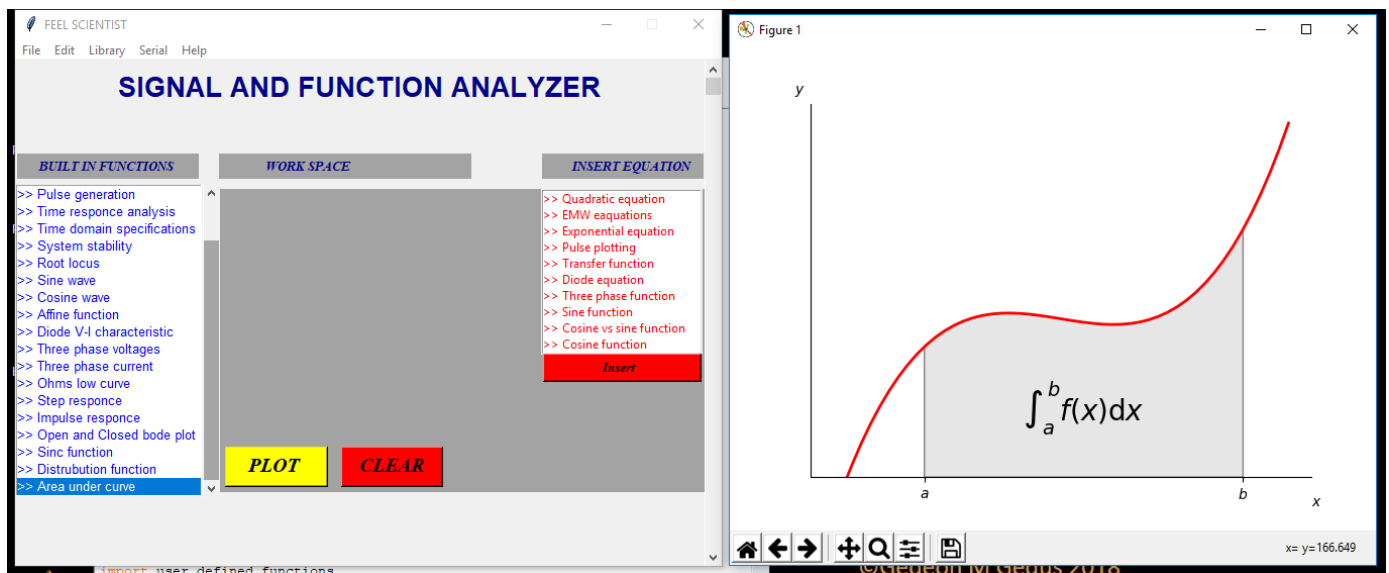


figure 15. Sinc function output

- **Area under the curve**



#### 4.4.3 ACQUIRING SERIAL SIGNAL

Other special thing in our project, it is possible to acquire an external signal from an input device(s).

If someone design any circuit this project can help him to find the output signal and analyses its behavior, trends or other parameters by clicking on serial and chose acquire serial signal.

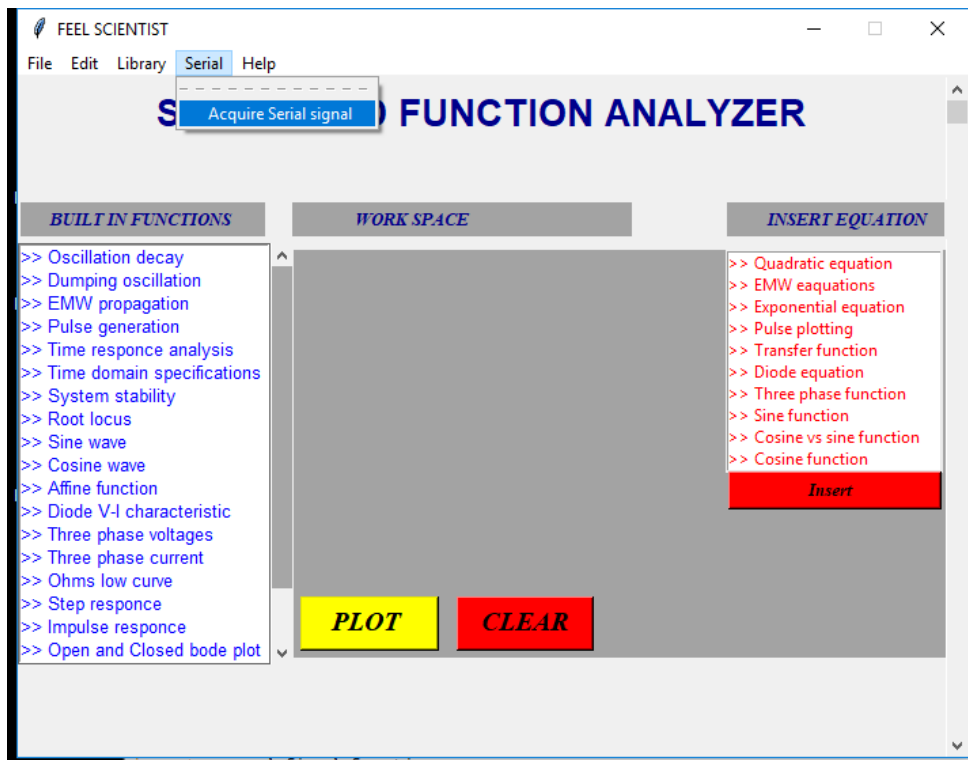


figure 16. Acquiring output signal

Once you click on acquire serial signal, you got the table on the workspace which contain the port and frequency. So you have to name the port and fill the frequency. In our experiment we use PORT 7 and 9600 hz as frequency used.

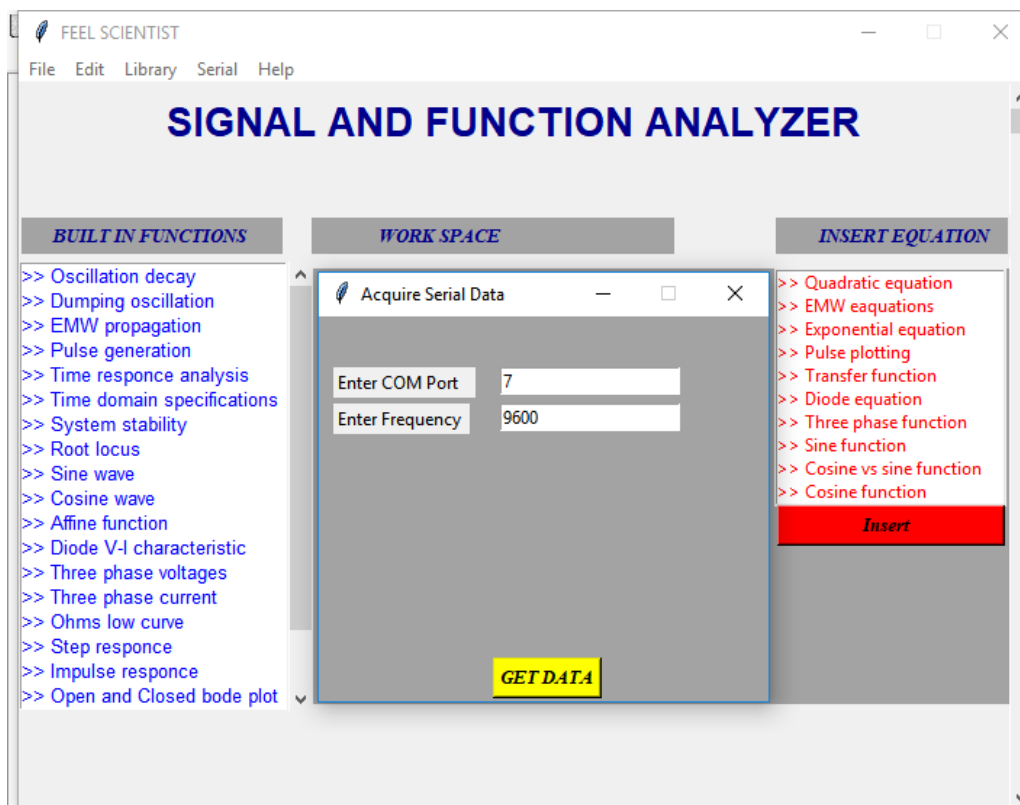


figure 17. Acquiring serial Data input

- **Reading digital signal**

This example shows you how to read button states, the button is connected to an Arduino, and once pressed it sends serial signal to the UI. For anyone who is learning this is quite good to show him signal transition and also to show that the button does not change state in zero time, though it is faster but the delay is there.

### Hardware required

- Arduino Board
- 1k ohm potentiometer

### Circuit

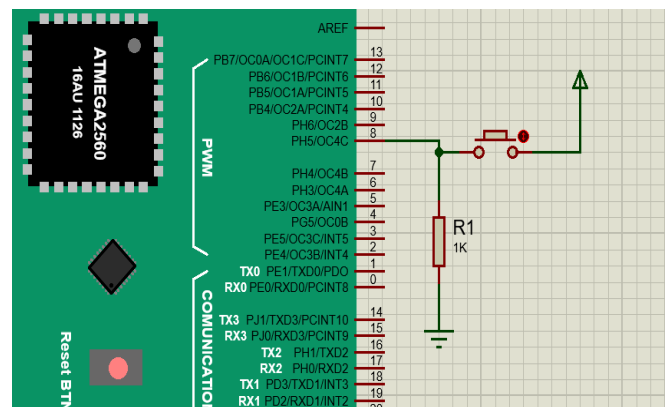


figure 18. hardware of reading digital signal

When the **button** is pressed, the input pin is pulled low. The value of **resistor** R1=1k controls how much current you want to flow from VCC, through the **button**, and then to ground. When the **button** is not pressed, the input pin is pulled high. The value of the **pull-up resistor** controls the voltage on the input pin.

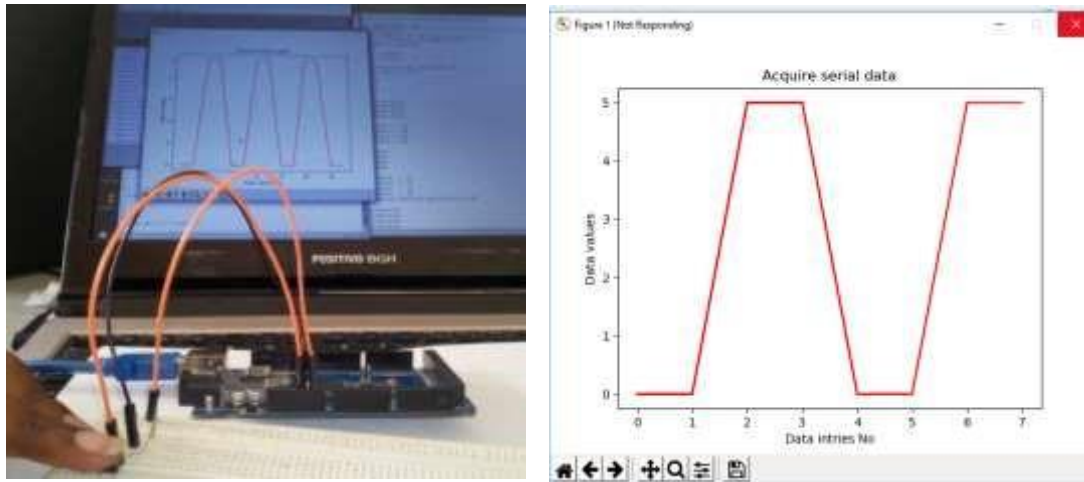


figure 19. result of reading hardware signal

- **Reading analog signal**

We often need to measure signals that vary; these are called **analog** signals. A 5V **analog** sensor may output is between 0.01V and 4.99V. Luckily, nearly all microcontrollers have a device built into them that allows us to convert these voltages into **values** these devices are Analog to digital converter (ADC).

Here is the circuit schematic of our experiment

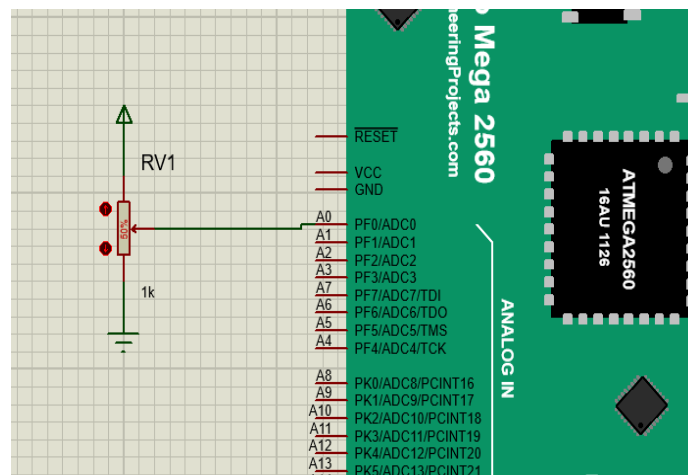


figure 20. reading analog signal

Connect the three wires from the potentiometer to your board. The first goes to ground from one of the outer pins of the potentiometer. The second goes to 5 volts from the other outer pin of the potentiometer. The third goes from the middle pin of the potentiometer to analog input 0.

By turning the shaft of the potentiometer, you change the amount of resistance on either side of the wiper which is connected to the center pin of the potentiometer. This changes the voltage at the center pin. When the resistance between the center and the side connected to 5 volts is close to, the voltage at the center pin nears 5 volts.

Com port and baud rate have been entered, for getting the output signal and all behavior of the

acquired signal click on **GET DATA** button.

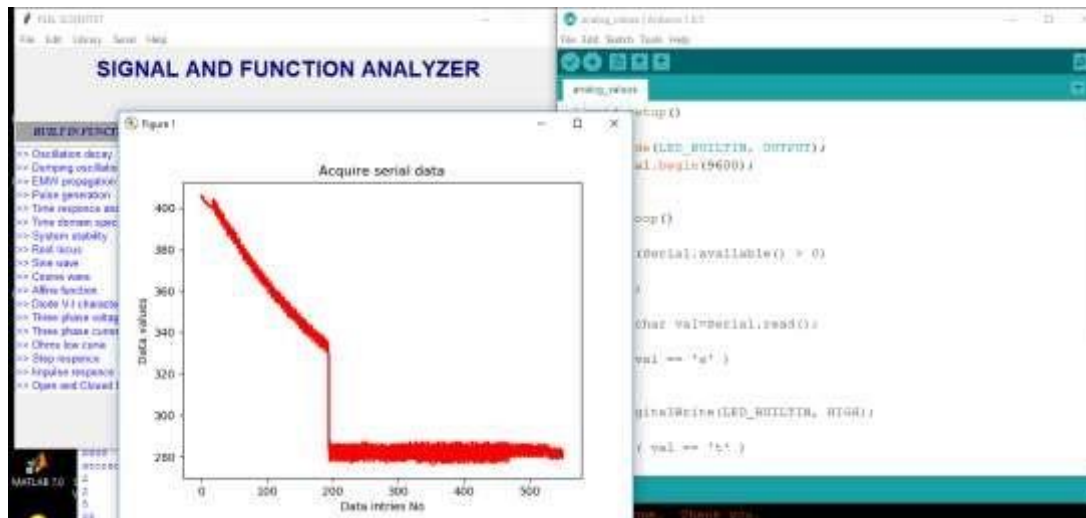


figure 21. analog signal plot

#### 4.4.4 DISCUSSION

We tried many things during project implementation as this programming language was new to us, we put in our efforts in order to learn and perform well our task. We tried and failed but all in all we have managed to come up with good results.

In our project we were dealing with python programming language which has many elements or features which helped us to write the codes to develop a graphical user interface to analyses signal and functions which is our very big task. We also experienced the python syntax which includes keywords, comment, basic operators, decision making statements, classes and objects.

Our signal and function analyzer have more factor which make it better for use to someone who want to analyzer signal behaviors and trends:

- It is possible to acquire signal from the output
- It is possible to take screenshot for future use
- It can be installed on computer and sharing it with other users

## **5 CONCLUSIONS, RECOMMENDATIONS AND FUTURE SCOPE**

### **5.1 CONCLUSION**

In this project of Python based signal and function analyzer using graphical user interface, we have seen that python is one of the major languages are used for the development of both desktop and web applications. Python has features that take care of common programming tasks. Python is simple to learn and easy to use. Sometimes, python marks as slower than other widely used programming languages like Java. Python applications can speed up by simply maintaining the code and using custom runtime. Python does support the modules and packages, which encourages program modularity and code reuse. Python provides an increase in productivity, which makes the first choice of developers. It has a great learning curve as it supports functional and procedural programming language. It is open source and can be freely distributed. The programming language mainly selected based on the requirement and compatibility with platforms and database. We have presented a small case-study of the use of Python as a foundation for the exploration and analysis of multi-modal data. Python's large user community and array of libraries enhance the language by providing new functionality useful in every aspect of data processing and management. The availability, flexibility, and ease of use of this language facilitates scientific endeavors from computationally intense applications to collaborative analysis.

Python is not alone in its use of compiled libraries. Other interpreted languages also take advantage of external libraries to overcome execution speed barriers. Python is open-source, making it quite an attractive solution to many applications. Applications developed in Python can be widely distributable, making it easier to enable collaboration between scientists at various locations.

## **5.2 FUTURE SCOPE**

According to our project python helps use to analyses the signal, function and acquiring the signal from the serial port. This UI can be used in our daily life in education or someone who want to know all about the signal behaviors specifically in EEE field

In the future we plan to make our executable file cross platform so that is will no longer be operating system dependent, and we plan to make way such that user can customize library (built in functions) this will all the UI to feet in all fields.

## **5.3 RECOMMENDATIONS.**

We commend that the system can be implemented everywhere mainly in engineering students to visualize the signals behavior and trends for more understanding we also made the system of acquiring the signal from the output of any circuit.

Also we recommend University of Rwanda especially College of Science and Technology department of Electrical and Electronics Engineering to take care of final year projects by providing all necessary means to students including A regular practical lab which has an internet connection, all needed software and hardware components that can help students to do their projects and be able to implement them and other necessary components needed by students should be available from the time a student comes in Year 1 till the end of his/her studies. This has to be more considerably for year 4 students so that they can produce the best memoires for their researches. Also It will be good if the students are motivated in reading books both hard and online/soft copies, for them to improve their knowledge. We recommend also to arrange students for more field trips because it helps them to know what really is being done in the world (outside the classroom). Also, we would like to suggest that UR-CST administration has to try to find more time for students to do their projects, this includes to reschedule final year class activities to give enough time for students to do their research. For better implementation we recommend our department to deepen embedded courses and to encourage students to pursue more hands-on practices.

## 6 REFERENCES

- [1] Noyan, Kinayman, "python for microwave and RF engineers", IEEE microwave magazine IEEE, vol 12, no. 7, pp. 1-6, 2011.
- [2] Noyan Kinayman, "Parallel Programming with GPUs: Parallel Programming Using Graphics Processing Units with Numerical Examples for Microwave Engineering", Microwave Magazine IEEE, vol. 14, no. 4, pp. 102-115, 2013.
- [3] L. Bavoil, S. P. Callahan, P. J. Crossno, J. Freire, C. E. Scheidegger, C. T. Silva, and H. T. Vo. VisTrails: Enabling interactive multiple-view visualizations. Proceedings of IEEE Visualization, 2005.
- [4] Martelli, A., Ravenscroft, A. M., and Ascher, D., editors (2005). Python Cookbook. O'Reilly, Sebastopol, California, 2nd edition.
- [5] [online] <https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-ofpython-programming-language-fd0b394f2121>
- [6] Langtangen, H. P. (2005). Python Scripting for Computational Science, volume 3 of Texts in Computational Science and Engineering. Springer. ISBN 3540294155.
- [7] B. Uengtrakul, D. Bunnjaweht, "A Block Diagram Programming System for Interactive DSP Simulation", Proceeding of the 11th International Conference on Electrical Engineering and Electronics Computer Telecommunications and Information Technology Proc. 11th IEEE ECTI, pp. 1-6, 14–17, may 2014.
- [8] T. E. Oliphant. Python for scientific computing. Computing in Science and Engineering, 9:10–20, 2007.
- [9] T. Zachariah et al. "The Internet of Things Has a Gateway Problem Categories and Subject Descriptors," *University of Michigan*, pp. 3–4, 2015.
- [10] J. C. Alexa Huth, "The Basics of Cloud Computing," *United States Computer Emergency Readiness Team*, pp. 123–149, 2014.
- [11] Roushan, T. and Chaki, D. et al. (2013), University course advising: Overcoming the challenges using decision support system, in Computer and Information Technology (ICCIT), 16th International Conference on, 2014, p. 13-18.
- [12] De Silva, Clarence W. Vibration and Shock Handbook, CRC Press, 2005, p. 16-63 Lecklider, Tom; Evaluation Engineering, October 2011, accessed October 10, 2011
- [13] K. P. Moran, C. Bernal-Cárdenas, M. Curcio, R. Bonett and D. Poshyvanyk, "Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps," in *IEEE Transactions on Software Engineering*.
- [14] K. P. Moran, C. Bernal-Cárdenas, M. Curcio, R. Bonett and D. Poshyvanyk, "Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps," in *IEEE Transactions on Software Engineering*.
- [15] Piotrowski Przemyslaw Build a Rapid Web Development Environment for Python Server Pages and Oracle / Przemyslaw Piotrowski // Oracle Technology Network. – Oracle. – 2012.
- [16] Roushan, T. and Chaki, D. et al. (2013), University course advising: Overcoming the challenges using decision support system, in Computer and Information Technology (ICCIT), 16th International Conference on, 2014, p. 13-18.



## 7 APPENDIX

This graphical user interface will help you understand the behaviors of different functions and signals in a real time manner and it is designed specifically for equations with three variables. Our file uses simple English when you want to insert an equation you use insert; you use plot to plot the inserted equation when you want to clear you use clear. This interface has built in functions and user inserted equations depending on what is in insert equation.

The toolbar contains four icons: - file, edit, library, help.

File: this will help you to create new file, open the file and it will show you the recent files that have been opened and also give you some examples. This icon will also help you take screenshot (capture screen) in case you want to keep an image of something you have plotted. It will also help to quit the program if you want to exit.

Edit: here you will be able to edit what you have written like undo and also to change the font or the appearance of the file.

Library: there are built in functions, with this icon you will be able to add or remove a library.

Serial: this will help to acquire signal from the input, it will ask you the port in which the circuit is connected to and even the speed or the frequency it will operate on.