

Virtual Laboratory for Pulse Width Modulation using Python Based Desktop Application

Akash Dutta,
Dept. of ECE, JISCE

Goutam Sutradhar,
Dept. of ECE, JISCE

Arindam Banerjee,
Dept. of ECE,
JISCE

Debabrata Debnath,
Dept. of ECE, JISCE

Dibyendu Paul,
Dept. of ECE, JISCE

Abstract— In this paper, a desktop application has been developed to perform virtual laboratory experiment for the creation of Pulse Width Modulation (PWM). This experiment can be performed at home in the PC or laptop. This application can be downloaded to perform the virtual experiment. In the application, there are provisions for the parameter selection to change the pulse width of a square pulse. When the program is run then the waveform is generated in a separate window. The desktop application has been developed by Python language. Moreover the application is interfaced with a Liquid Crystal Display (LCD) screen to show the amplitude and the duty cycle of the PWM signal.

Keywords— Desktop Application, Numpy, Matplotlib, Scipy, Python, Microcontroller, Liquid Crystal Display

I. INTRODUCTION

In the Pandemic situation caused by COVID-19, the students as well as the teachers are confined at home. Classes are going on virtually but it is quite difficult to perform the laboratory experiments virtually. Therefore an initiative has been taken to perform laboratory at home such that the students can have the technical knowledge about the experiments. Many institutions have created their own virtual labs [1-3] and from their website it can be accessed. These are the web applications.

Getting inspired by their initiative, we have also started to design our own virtual labs. But the major difference of our virtual laboratory experiment with respect to the earlier reported virtual laboratories is that it is in the offline mode which means without the internet connection the experiment can be performed. The application and the datasheet is downloaded by the students and the experiment can be performed by running the application. If one time the application is downloaded then it can be performed multiple times as required. The students need not worry about the internet issues. The novelty of the work is the creation of desktop application for performing the laboratory experiments virtually at home and also the parametric results can be displayed in the LCD screen to participants which was not done earlier. In this respect, the work is totally new to the best of the knowledge of the authors and therefore can not be compared with other similar works.

The major limitation of the work is the database maintenance. To store the result data, local database system (like Microsoft Excel or Access) is used. Central database can not be maintained such that the observer or the laboratory demonstrator can observe the results obtained by the participants.

The paper has been organized as follows. Section II describes the desktop application. Section III describes the hardware application that is the interfacing of the application with the hardware. Section IV is the conclusion.

II. DESKTOP APPLICATION

The application has been developed in Python. Here we have used a well known library “tkinter” to create the user application. To create user interface at first we have to import the class “`from tkinter import *`” to include all existing methods. In this experiment we perform PWM using the well known “555 Timer” integrated circuit. The circuit diagram has been drawn in the paint app and it has been included in the main application as a file dialog. The code “`from tkinter import filedialog as fd`” indicates the inclusion of the image as file dialog in the main application. The image is included either in a canvas or in a label. Here we have included the image in the label. The image file is at first taken from the file dialog box using the code “`fg=fd.askopenfilename()`”. This image file is included in the label as a photo image with the following code “`fig=PhotoImage(file=fg)`”. Next to create the application, a window is created with the following codes “`tk=Tk()`”, “`tk.title("Pulse Width Modulation using 555 Timer")`” and “`tk.geometry("1200x700")`”. The first code is to create the class, second code is to create window title and the third code is to assign the size. Then a label is created with the following code “`Label(master object,figure=fig).grid(row=value,column=value)`” where the figure is displayed. “`tk.mainloop()`” maps the application interface in the main loop. In this application we aim to design a pulse width modulator circuit using 555 timer IC. In 555 timer circuit, a square wave can be generated with the time period given by the following equation.

$$T = 0.69 \times (R_1 + 2 \times R_2) \times C \quad (1)$$

Here the high and the low time intervals can be expressed with the following equations as follows.

$$T_{HIGH} = 0.69 \times (R_1 + R_2) \times C \quad (2)$$

$$T_{LOW} = 0.69 \times R_2 \times C \quad (3)$$

There is another term in PWM which is very important and that is the duty cycle. Duty cycle can be defined as follows.

$$Duty\ Cycle = \frac{T_H}{T} \times 100\% = \frac{0.69 \times (R_1 + R_2) \times C}{0.69 \times (R_1 + 2 \times R_2) \times C} \times 100\% = \frac{(R_1 + R_2)}{(R_1 + 2 \times R_2)} \times 100\% \quad (4)$$

To achieve a perfect square wave, R_2 is taken to be far greater than R_1 i.e. $R_2 \gg R_1$. But in PWM, we need to vary the pulse width. Therefore R_1 needs to be varied. Therefore in the application, There is a provision for varying R_1 .

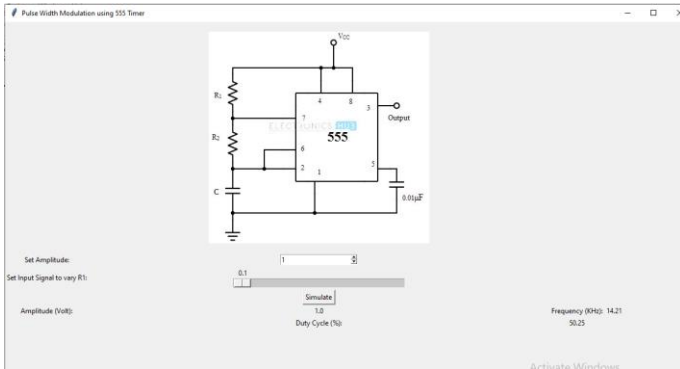


Figure 1 Application window for the virtual experiment

In the application window, there is a slider by which R_1 can be varied. To create a slider, the following code can be written

`sld=Scale(tk,from_=0.1,to=10,resolution=0.1,variable=x,length=300,orient=HORIZONTAL)` and `sld.grid(row=2,column=1)`. The first code creates a scale (slider) with the starting value (“from_=0.1”) 0.1 and the ending value (“to=10”) 10 with an increment of 0.1 (“resolution=0.1”). A variable “x” has been taken which is of double type (“x=DoubleVar()”). It takes the value after each sliding. The option “length” gives the size of the slider. There is another option “orient” which sets the alignment of the slider (horizontal or vertical). The second code is to set the position of the slider onto the application window. There are different layouts for positioning: (i) Grid layout, (ii) Box layout. Here grid layout has been used.

To set the amplitude, a spinbox has been used. The spinbox can be created with the following codes `spb1=Spinbox(tk,from_=1,to=10)` and `spb1.grid(row=1,column=1)`. The first code creates a scale (slider) with the starting value (“from_=1”) 1 and the ending value (“to=10”) 10. The second code is to set the position of the slider onto the application window. Here also grid layout has been used.

To simulate for the waveform, there is a button. The following code can create the button `Button(tk,text="Simulate",command=simulate).grid(row=3,column=1)`. The button caption can be set by the option “text”. When the button is triggered then command option is triggered which calls the method “simulate()”. Inside the “simulate()” method, the calculation for the pulse width is performed and the waveform is generated. To generate the waveform, three libraries are needed “numpy”, “matplotlib” and “scipy”. “numpy” is used to create the timing array for the waveform plotting. “scipy” is used to generate the square wave based on the timing array created by the “numpy” library. Finally “matplotlib” is used to plot and show the waveform. Figure 1 shows the application window for the virtual lab. Figures 2 and 3 show the application window before and after simulation. At first the amplitude and the resistor value (R_1) is set. Then the “Simulate” button is pressed. Inside the button, the code for interfacing the hardware and the application is given. Figure 4 shows the waveform window after the simulation. It is created using the “matplotlib” library.

When the button is pressed then the corresponding reading of the experiment is displayed in the LCD screen. To interface the application with the hardware “pyserial” library has been used. In the “serial” class, port number and the baud rate is given. The following code is to identify the hardware `board=serial.Serial(“COMX”,baudrate)”`. `board.write(stringdata.encode())` converts the Unicode string (UTF-8 format) to byte and transfers to the microcontroller serial port.

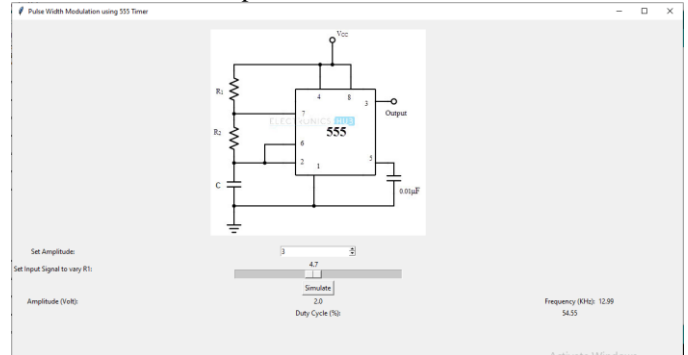


Figure 2 Parameters (amplitude, resistor value etc.) setting inside the application before simulation

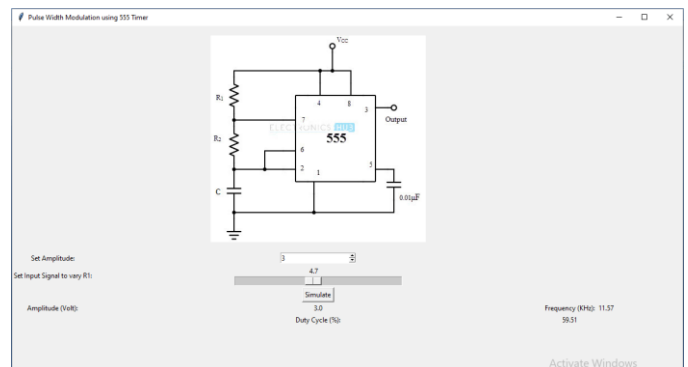


Figure 3 Results (amplitude, frequency and duty cycle) display inside the application after simulation (after the button is pressed)

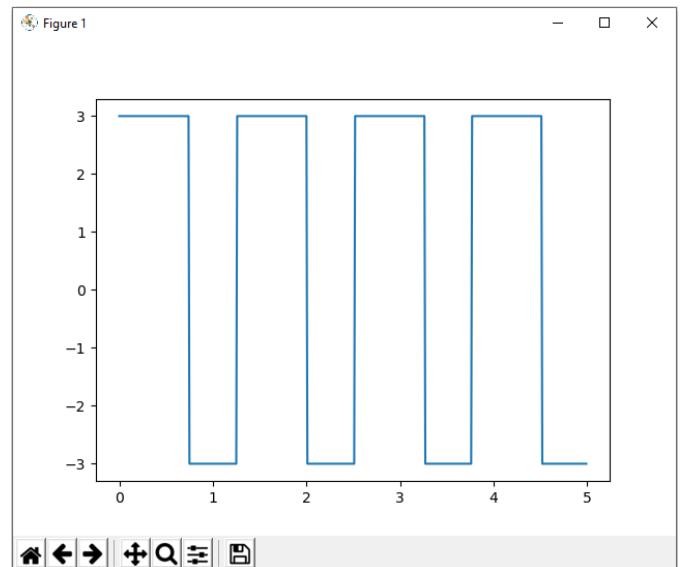


Figure 4 PWM waveform

III. HARDWARE APPLICATION USING MICROCONTROLLER

A. Microcontroller Programming Fundamentals

In this work, as a microcontroller Arduino Uno AtMega328P has been used though Arduino Mega 2560 can also be used. Figure 5 shows the diagram for the microcontroller board. It is a 16MHz microcontroller and run by 5V volt DC power supply. The programming language used here is embedded C. For serial communication, the code “`Serial.begin(9600)`” is used. Here 9600 is the baud rate or the data transfer rate. “`Serial.readString()`” converts the byte code to Unicode string (UTF-8). The simulation reading from the application is taken as a string value. Here at a time duty cycle percentage and the amplitude are converted into integer and then into string inside application. The two string values are then concatenated (duty cycle + amplitude) and converted to byte form with the “`encode()`” method inside the application. In embedded C code, “`readString()`” method again converts byte code to string. This string is then converted to integer with the help of “`toInt()`” method which is inbuilt in the “String” class. The percentage of duty cycle and the amplitude is then separated from the integer reading and it is fed to the LCD module.

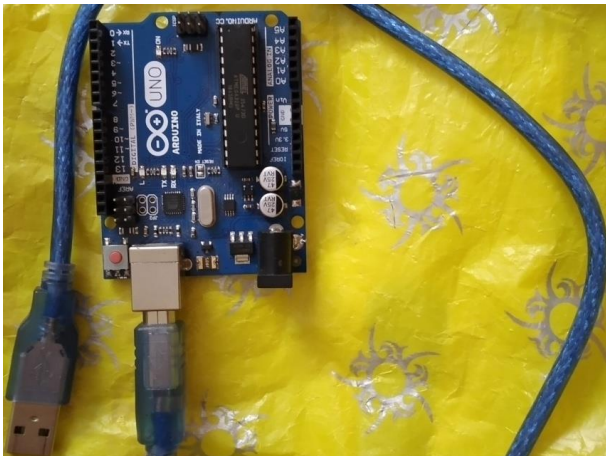


Figure 5 Arduino Uno AtMega328P microcontroller board

B. Liquid Crystal Display (LCD)

Liquid Crystal Display is used to display the reading following the command given by the application. Figures 7, 8 and 9 show the reading display of the LCD module. Basically LCD has 8 pins but here another I2C (Inter Integrated Circuit) module has been interfaced with the LCD such that the data from the 6 pins (excluding supply and ground pin) can be easily transferred by serial communication. In I2C module there are SDA (Serial Data Adapter) and SCL (Serial Clock) pins for serial communication. The library used for LCD display is “`LiquidCrystal_I2C.h`”. Here three parameters are required (i) device address, (ii) column number and (iii)

row number. For our case, the device address is **0x27 (27Hex)**. The LCD module has 16 columns and 2 rows. At first we need to initialize the device with the method “`.init()`”. “`.backlight()`” method turns on the backlight for the display. “`.setCursor(column index, row index)`” sets the printing position of the LCD. “`.print(string or integer data)`” prints the data onto the screen. Figure 6 shows the duty cycle and the amplitude reading after the simulation when the “Simulate” button in the application is pressed.

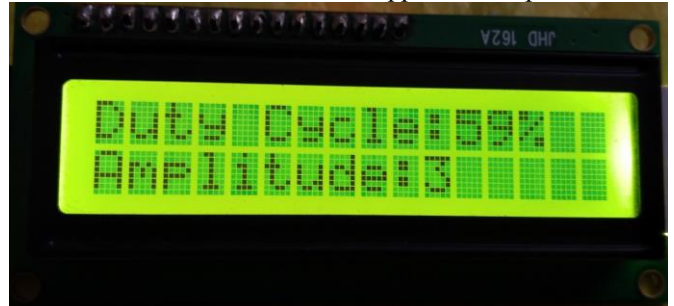


Figure 6 Amplitude and Duty Cycle Reading in LCD

The following equation converts Celsius reading in Fahrenheit scale.

$$F = 32 + \frac{C \times 9}{5} = 32 + 1.8 \times C \quad (1)$$

where C and F are the values in Celsius scale and Fahrenheit scale respectively.

IV. CONCLUSION

In this paper, we designed a virtual platform for PWM waveform generation. A desktop application was developed to create the virtual laboratory experiment. Moreover the reading of the experimental data was displayed onto a LCD screen through the microcontroller. The application was developed in python and the hardware interfacing with the microcontroller was achieved with the application was achieved with the help of a predefined library. The work was totally new to the best of the knowledge of the authors and so could not be compared with any other existing works.

ACKNOWLEDGMENT

The authors express their heartiest gratitude to the authorities of JIS College of Engineering for giving the opportunity to perform the research.

REFERENCES

- [1] IIT Kharagpur, "Virtual Lab," [Online]. Available: <https://www.vlab.co.in/participating-institute-iit-kharagpur>.
- [2] IIT Guwahati, "Virtual Lab," [Online]. Available: <https://www.vlab.co.in/participating-institute-iit-guwahati>.
- [3] IIT Bombay, "Virtual Lab," [Online]. Available: <http://vlabs.iitb.ac.in/vlab/>.