Published by *APT Kerala*
Article ID 220102, **2022**, (pp. 1-7)

Special Theme : *Approaches to Online Teaching*

Computer Interfaced Science Experiments

Ajith Kumar B P *

Abstract. Science experiments generally involve measurement and control of physical parameters. Interfacing appropriate sensor/control elements to a computer enables it to assist in performing science experiments with a finer degree of control. A cost effective implementation of such a system is explained taking ExpEYES as an example. The hardware and software features of the setup are described along with an example experiment. This article is meant to be a pointer to the resources available on the Internet about this device.

1. Introduction

Science is the study of the physical world by systematic observation and experiments. Even though both theory and experiment are equally important in science, the latter is often less emphasised in schools and colleges, partly due to lack of equipment. A science experiment generally involves measurement and control of several physical parameters such as temperature, velocity, voltage etc. For each experiment, a dedicated equipment is required for this purpose, resulting in high associated costs. Every equipment that is designed to measure some physical parameter will have a sensor element that generates a proportional voltage or time signal, some signal processing circuits, a digitizing circuit, and a user interface consisting of displays and controls. The advent of cost effective personal computers offers an alternative solution to this stack of elements. Interfacing the sensor and control elements to computer allows shifting of the user interface to the software. In this method the configuration of the measurements becomes very flexible and the same equipment can be used for a wide range of experiments just by changing the sensors and some parts of the software. This flexibility also resulted in developing a lot of demonstrations that can support classroom teaching.

Inter-University Accelerator Centre, New Delhi, in 2005 developed an interface that could perform a set of physics experiments. It was done under a project called “Physics with Home-made equipment ans Innovative Experiments (PHOENIX).[1] The design provided a hardware and software framework that enabled accessing sensor/control elements to perform science experiments without getting into the details of electronics or computer programming. The first version utilized the parallel port of the PC and later versions used the USB interface. The software is distributed under the GNU General Public License and the hardware under the CERN Open Hardware License. There have been several versions from IUAC and other sources. The latest version is called ExpEYES-17, open sourced by Jithin B.P.[3] The following paragraphs offer a brief overview of this equipment. For further

*Former Scientist, Inter-University Accelerator Centre, New Delhi, Email: bpajith@gmail.com

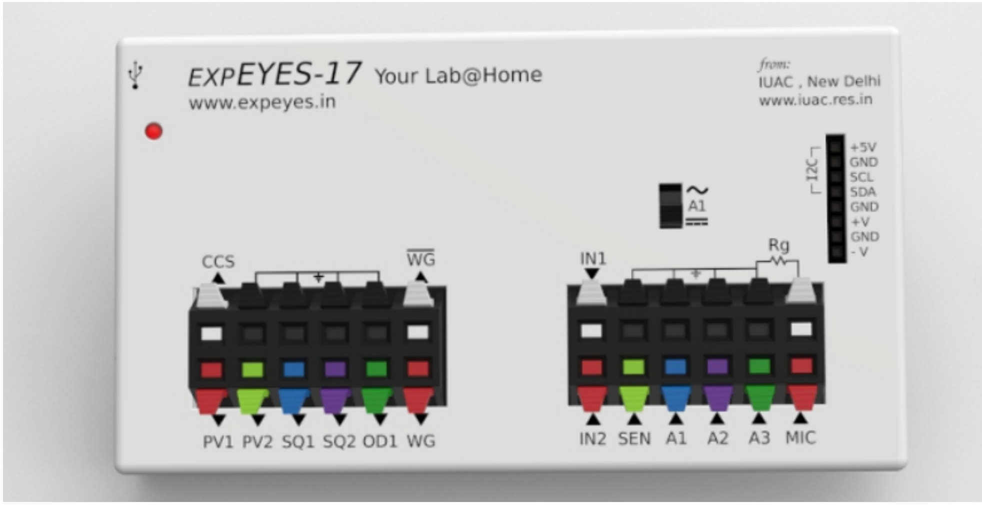

Fig. 1. ExpEYES Hardware.

details, visit the website.[2]

2. Hardware

A photograph of the instrument is shown in Fig. 1. A set of Input/Output terminals are available on the top panel to facilitate the measurement and control functions. Programmable Voltage Sources, Waveform Generators, Voltmeters, Oscilloscope inputs, Capacitance meter etc. are some of the main features of this device. The hardware consists of a micro-controller which handles all data acquisition and control tasks, and is programmed to communicate with a connected computer via USB. It is augmented with various analog circuits based on parts such as Op-amps and Hex buffers in order handle wide voltage ranges and supply additional currents than what the micro-controller is capable of. The various power supply voltages required by these analog circuits are also derived from the 5 volt power supply of the USB interface by means of charge pumps and voltage regulators. In order to measure physical parameters such as temperature, pressure etc., appropriate sensor elements are used to generate proportional electrical signals. ExpEYES also has a dedicated connector for plugging in commercially available I2C sensors, greatly enhancing the functionality of the device. All protocols for interfacing with I2C supported sensors are implemented in the software, and no additional changes to the hardware are necessary. At present, ExpEYES supports sensors which measure magnetic fields, angular velocity, acceleration, luminous intensity, pressure, temperature, humidity, distance, LIDAR and more .

3. Software

The Software is written in a layered manner. The most fundamental layer is the code running on the micro-controller that performs all the real-time measurements, with sub-microsecond timing resolution. This is also known as the *firmware*. It also listens on the USB interface for commands from the connected host computer. Operations requested by the host are carried out, and the resultant data or error code is sent back to the computer.

The next layer is the communication library that runs on the host computer, and it is written in the Python language. It implements function calls handling every feature of the hardware, such as voltage measurement from any of the available analog inputs, or setting the voltage on any of the analog outputs. There are several more complex function calls such as the *capture* routines which read a set of voltage values at precise time intervals, thereby enabling digitization of waveforms. The topmost layer consists of application programs for a wide range of experiments. Their function is to acquire data from the hardware using the communication library, and provide the user interface for each experiment. The data analysis and graphical features of Python are combined with the real-time measurement capabilities of the ‘C’ code running on the micro-controller to implement a host of experiments. Each control and measurement feature implemented in the hardware will have a corresponding function call in the Python library to enable the user to invoke it.

4. Example Experiment

The functioning and advantages of this hardware and software combination may be demonstrated better by using a simple experiment. Measuring the transient response of an LCR circuit is chosen for this purpose. The purpose of the experiment is to apply a voltage step across the circuit and study the resulting voltage variation across the capacitor. A coil, a resistor, and a capacitor are connected in series as shown in the Fig. 2.

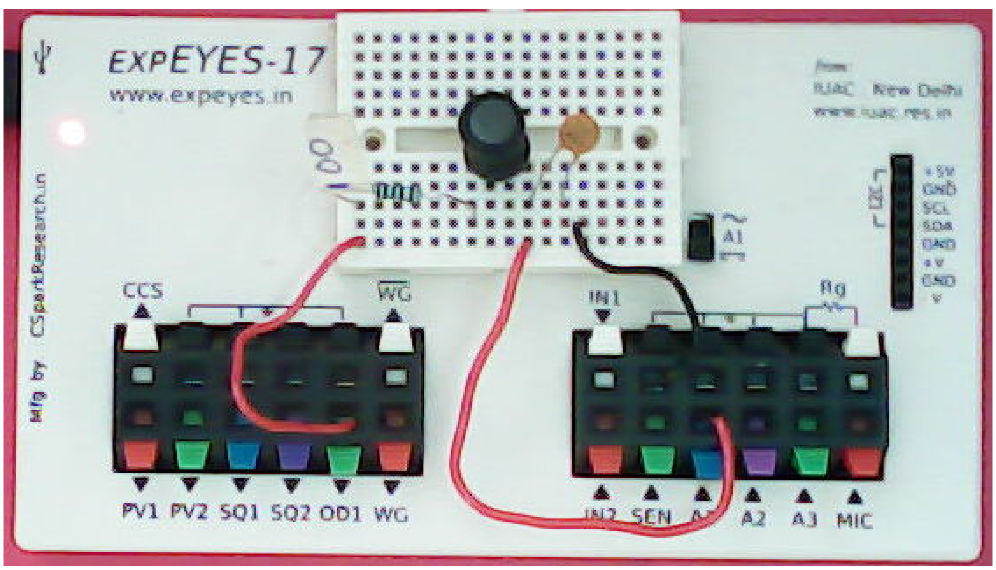

Fig. 2. The LCR circuit under study.

4.1. Simulating the expected results—Before proceeding with the actual experiment, we will try to simulate the results using a Python program by implementing the equations of transient response.[4] The voltage across the capacitor is given by the equation $v(t) = e^{\beta t} K \sin \gamma t + \theta$. The Python code given below evaluates this equation for a given set of L, C and R values. The result of the program is shown in Fig. 3.

```
from pylab import *

R = 100 + 20
L = 10e-3
C = 100e-9

B=-R/(2*L)
G=sqrt(1/L/C - (R/2/L)**2)
damping = (R/L)/(2/sqrt(L*C))
print ('Freq = %5.1f  Damping
 Factor = %5.2f'%(G/(2*pi), damping))

t = linspace(0, .001, 200) # 0 to 1 mS
v = 5* exp(B*t) * sin(G*t+pi/2)
plot(t,v)
show()
```

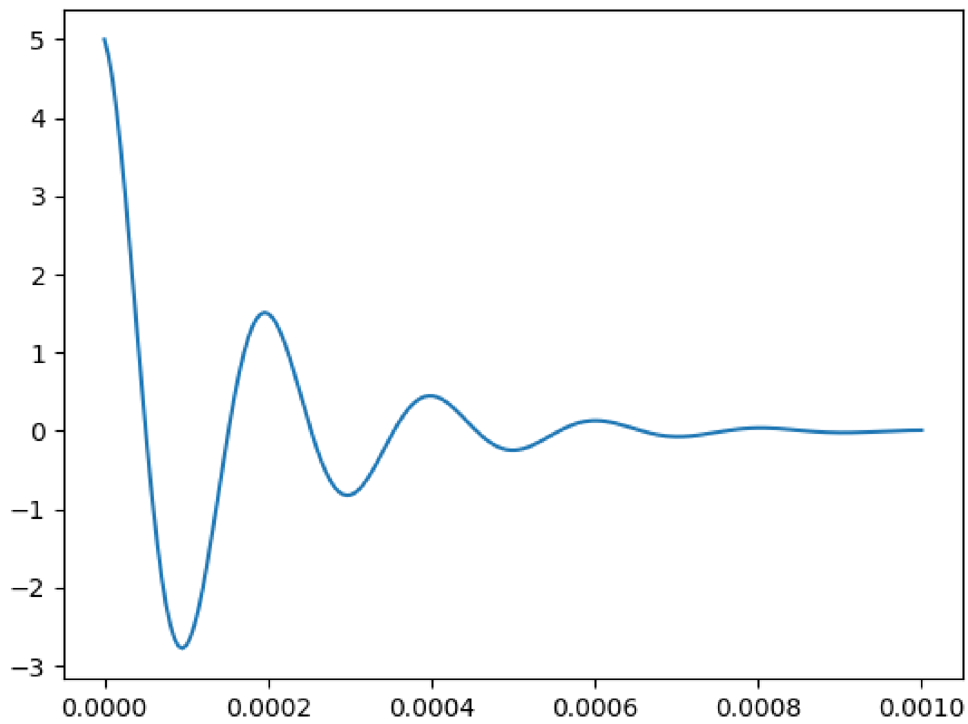

Fig. 3. Simulating LCR Transient response with Python.

4.2. Measurement of real values from the LCR circuit—For studying a functioning LCR circuit, connections are made according to Fig. 4. The LCR elements are connected in series with one end of the capacitor to ground. The other end of the capacitor is connected to the voltage measurement/oscilloscope terminal 'A1'. One end of the resistor is connected to the Digital Output terminal 'OD1'. The procedure is to apply a voltage step on OD1, and

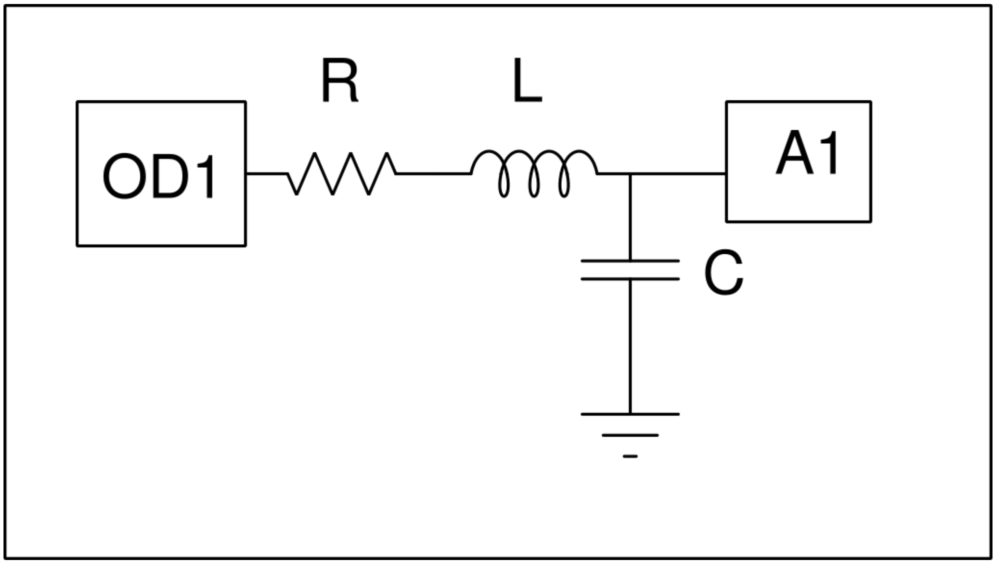

Fig. 4. LCR Transient response, schematic.

capture the resulting voltage variations at A1 caused by the inductor and the capacitor. The simulation shows that the major voltage fluctuations that follow the change in voltage level at the input (OD1 connected to R), are completed within a millisecond, and this means that we need to measure the voltage variations only for that duration or perhaps a little more to account for component tolerances.

The measurement process can be implemented using the following Python code.

```
import eyes17.eyes
p = eyes17.eyes.open()

p.set_state(OD1=1)          # OD1 to HIGH
time.sleep(0.1)
t,v = p.capture_action('A1', 500, 2, 'SET_LOW')

from pylab import *
plot(t,v)
show()
```

The first two lines of code establishes the connection with the hardware. The next line sets the Digital Output OD1 to five volts, and then the program pauses for a short while to allow the voltage to settle. After this, the $capture\_action$ function is invoked with a $'SET\_LOW'$ argument. This function first sets the voltage of OD1 to 0 volts, and immediately starts recording 500 voltage measurements on A1 with the time interval between two readings set to 2 μS. The total duration of the measurement cycle thus total 1 mS. The final three lines of the code plots the resultant waveform using Python's Matplotlib module.

We can write similar programs for all the possible experiments that can be done with the available hardware. However, graphical programs for a range of common experiments

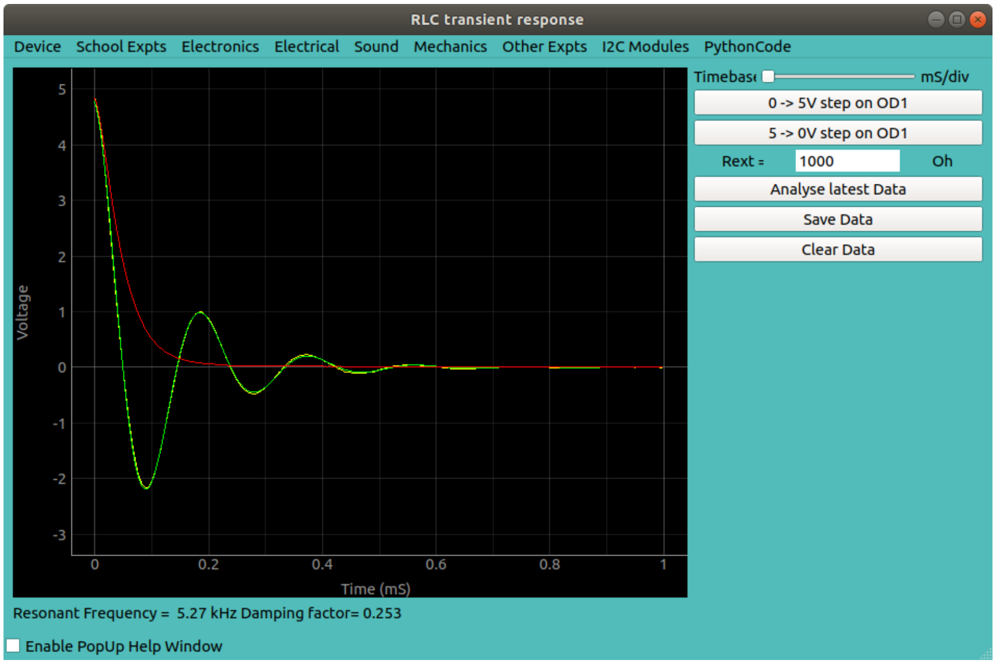

Fig. 5. Graphical User Interface for the LCR Experiment showing two plots for underdamped and fully damped LCR circuits.

are provided to make the process easier. It is up to the user to choose between the two approaches. The screen capture of the GUI screen for the RLC transient response experiment is shown in Fig. 5, and one can see that it includes buttons for toggling the voltage, and a slider for indicating the expected measurement duration.

5. Conclusion

The method of developing a computer interfaced laboratory equipment has been discussed. Only a single experiment is described, even though a wide range of them have been performed and documented. The usage of the Python language makes the comparison of the experimental results with the theoretical calculations much easier. The automated measurement process enables taking a large number of readings so that the statistical error can be estimated. Courses for Experiments with ExpEYES are available freely,[5] and further details can be found in the references given.

Acknowledgements

This work was initiated by a team led by the author. The contributions of Mr V V V Satyanarayana and others are highly acknowledged. The contribution from Jithin B P has enhanced the scope of the project significantly. This work has progressed and reached the present level thanks to the encouragement and feedback from the academic community.

Notes and References

[1] The phoenix Project, IUAC

[2] Home page for the ExpEYES Project

[3] Development history of the device showing the various hardware prototypes

[4] Course material from NPTEL on DC transients in RLC circuits

[5] A MOODLE course on ExpEYES