Demonstration of Online Practical Experiment during COVID-19 Pandemic: Determination of Boltzmann constant & Reverse Saturation Current using ExpEYES-17 kit

Durjoy Roy[1] and Subhrajyoti Biswas[2]

[1]Department of Electronic Science, Rishi Bankim Chandra College, Naihati 743165, India.
roy.durjoy@gmail.com

[2]Department of Physics, Rishi Bankim Chandra College, Naihati 743165, India.
anjansubhra@gmail.com

Submitted on 13-01-2022

Abstract

The Boltzmann constant can be easily obtained in teaching laboratories using a traditional method. The method devised here uses the ExpEYES-17, a low-cost, microcontroller-based digital data acquisition system, where the actual experiment is performed with real diodes, and Python is used to complement the learning.

1 Introduction

The determination of fundamental physical constants plays a crucial role in understanding the underlying principles of nature. One such constant, the **Boltzmann constant** (k_B), establishes a fundamental relationship between temperature and energy, serving as a cornerstone in statistical mechanics and thermodynamics. Traditional methods for determining k_B often involve complex experimental setups and extensive manual data collection. However, modern technological advancements have enabled the development of cost-effective and efficient methodologies that enhance accuracy while simplifying the process.

The COVID-19 pandemic and subsequent lockdowns disrupted traditional hands-on laboratory education, highlighting the need for a new paradigm of **online and remote practical experiments**.

The transition to virtual learning necessitated the adoption of digital tools that could facilitate real-time data acquisition, remote control of experiments, and computational analysis. In response, digital laboratory platforms such as **ExpEYES-17** have emerged as powerful alternatives, allowing students and researchers to conduct physics experiments from their homes or in hybrid learning environments. The availability of low-cost, open-source hardware and software has further expanded access to experimental learning, overcoming geographical and logistical barriers.

In this study, we present an alternative approach to determine the Boltzmann constant using a PN junction diode and the **ExpEYES-17** data acquisition system. Previously, a study [7] was published to find Boltzmann Constant. In this article, we have applied curve fitting method, and determined the reverse saturation current as well.

ExpEYES-17 is a low-cost, microcontroller-based digital data acquisition system specifically designed for educational and research applications. It offers precise control over experimental parameters, real-time data visualization, and seamless integration with Python-based computational tools. The versatility of ExpEYES-17 has been demonstrated in various experimental setups, including measurements of electronic circuit parameters, sensor-based studies, and thermodynamic experiments [1, 10].

The experiment is based on the **current-voltage (I-V) characteristics of a PN junction diode**, which follows the well-known diode equation. By analyzing the exponential relationship between current and voltage in the forward bias region, we extract the Boltzmann constant using a curve-fitting technique. The experiment also incorporates a **Platinum Resistance Thermometer (PT100)** to measure the temperature of the diode during operation, ensuring accuracy in calculations.

Previous studies have successfully employed ExpEYES-17 for undergraduate physics experiments, as documented in works published in *IOP Physics Education* and *IAPT Physics Education* [8, 9, 11, 14, 15]. These works highlight the effectiveness of the system in hands-on learning and experimental physics education, demonstrating its applicability in teaching fundamental physics principles. Inspired by these studies, our work further extends its utility by providing a refined approach for determining the Boltzmann constant.

This approach provides students and researchers with an accessible and efficient method for determining the Boltzmann constant while reinforcing fundamental concepts in semiconductor physics, thermodynamics, and experimental data analysis. The integration of Python-based computation further enhances the learning experience, making it a valuable addition to undergraduate and postgraduate physics laboratories.

2 The ExpEYES-17

The ExpEYES-17 is basically a data acquisition system / kit along with a four channel digital storage oscilloscope (DSO) which was developed by a group of scientists and researchers at the Inter-University Accelerator Center (IUAC), New Delhi, India. The name of the kit is short form of *expEriments for Young Engineers & Scientists*. The main architecture has been designed using the Micro-controller $PIC24EP64GP204$ and runs by Python. The hardware design and necessary software are freely available [17] to share knowledge and foster interest in experiments worldwide.

The top view of the kit has been shown in Fig. 1. There are separate connector blocks - output block, input block and I^2C modules. The output block contains programmable sources for various signals and voltages. $PV1$ and $PV2$ are two programable direct voltage sources with 12 bit resolution. It has two square waves generators $SQ1$ and $SQ2$. The frequency of these sources can be varied from $4Hz$ to $100kHz$ as well as duty cycle. However, the frequency range of sine or triangular waves is lower ($5Hz$ to $5kHz$) compared to that of square waves. The waveforms other than square wave is obtained from WG. The $\overline{WG}$ indicates the complements of the signals of WG. Apart from these outputs, there are One digital Output $OD1$, and one constant current source CCS which can supply $1mA$ current. Any other waveforms which are not specified can also be generated by writing a simple python code as discussed in the user manual.

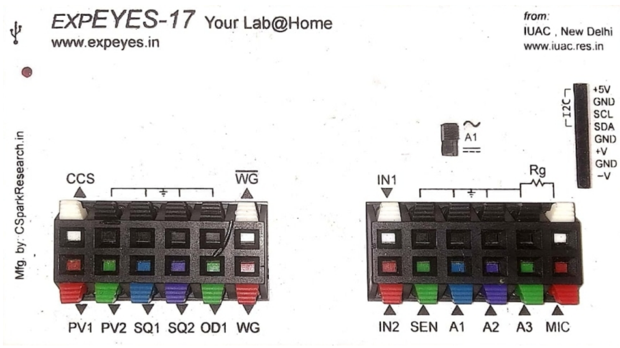

Figure 1: The inputs & outputs of the ExpEYES-17 kit.

On the other hand, inputs for signal capturing have been integrated into the input block, which contains six input terminals. The $IN1$ is used to measure capacitance upto $10nF$ and the frequency counter $IN2$ is able to measure frequencies upto several MHz. The input terminals $A1$ and $A2$ are capable of measuring analog voltages within $\pm 16V$ range having maximum $1Msps$ sampling rate. Other inputs, $A3$ can measure $\pm 3.3V$ and MIC is capable of capturing audio signals.A detailed discussion about the device can be found in article [10].

The kit is connected to laptop / computer through usb port and no external power is needed to run the device. The Fig. 2 displays the graphical user interface GUI which is written on C and python. The GUI consists of oscilloscope display and the control sliders for $PV1$, $PV2$, $SQ1$, WG and four oscilloscope channels $A1$, $A2$, $A3$ and MIC including the time base and trigger control. There is a list of about 50 experiments from school to graduate levels that can be performed with this kit. But one may design other experiments too.

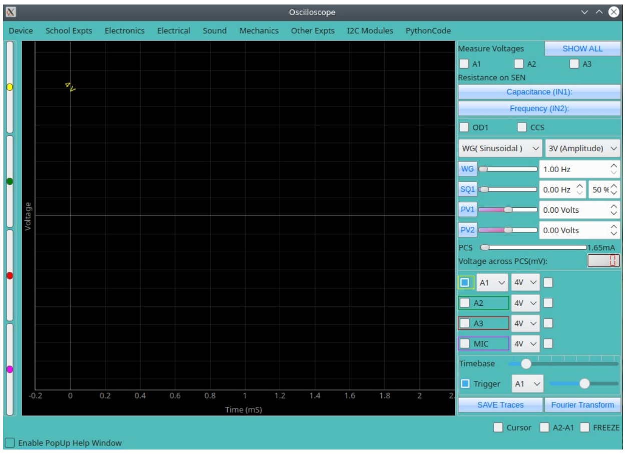

Figure 2: A view of the User Interface (Software) of the ExpEYES-17 kit.

Boltzmann constant.

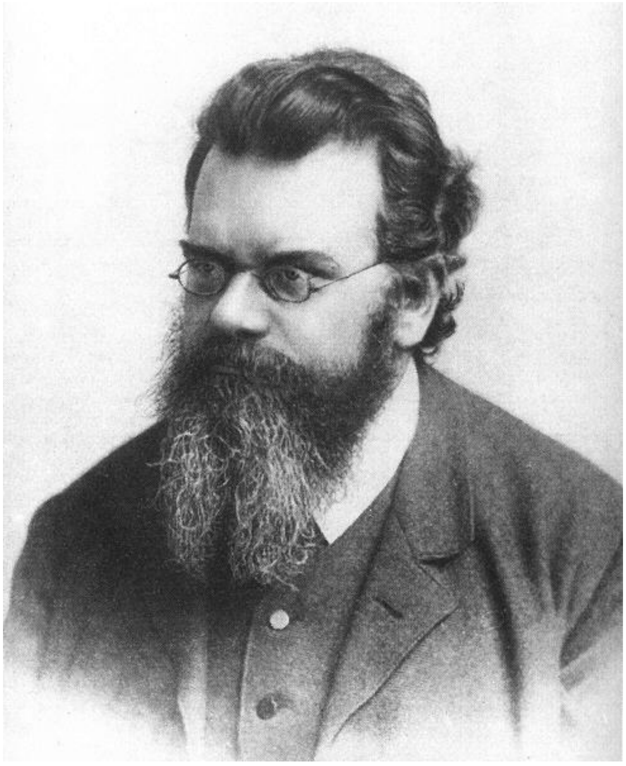

Figure 3: Ludwig Eduard Boltzmann (1844-1906)

3 Theory:

3.1 Boltzmann's Constant:

Ludwig Eduard Boltzmann (1844-1906) was an Austrian theoretical physicist and philosopher. His greatest achievements were the development of statistical mechanics, and the statistical explanation of the second law of thermodynamics. In 1877 he provided the current definition of entropy, $S = K_B ln(\omega)$, interpreted as a measure of the statistical disorder of a system.[19]. Max Planck named the constant k_B the Boltzmann constant.

3.2 PN Junction diode, and finding the Boltzmann Constant from its equation:

The discovery of semiconductors ushered in a new era of electronic technology. The pure semiconductors behave like ordinary resistance and follow the Ohm's current-voltage relation and they have no practical applications in electronic devices. Some impurities are doped to produce impure semiconductors. Depending upon the doping materials, the impure semiconductors are either p-type or n-type. The former contains a large number of positively charged holes and a small number of electrons. The n-type semiconductor contains large number of electrons and few numbers of holes. In Fig. (4) p-type and n-type semiconductors are shown.

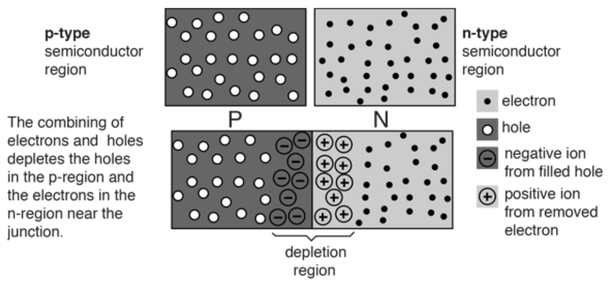

Figure 4: p type and n type semiconductors, and p-n junction diode, formed on a single crystal [18]

When different materials are doped onto a single crystal in such a way that one side forms a p-type region and the other side forms an n-type region, it creates a PN junction, as shown in Fig (4). Almost all the electronic devices contains one or more such pn junctions. Since the p-side has a high concentration of holes and the n-side has a high concentration of electrons, electrons naturally diffuse toward the p-side and holes toward the n-side. Electrons and holes recombine as they diffuse across the p- and n-sides. However, some electrons on the p-side near the junction cannot recombine. Similarly, holes on the n-side near the junction line can not recombine. The electrons near the junction line on the p-side and holes on the n-side set up an electric field and this electric field prevents further diffusion of holes from p-side and electrons from the n-side. A charge depletion region is created across the junction, as shown in Fig (4) forming a barrier potential V_0. This barrier potential varies from $0.1V$ to $0.3V$.

Such pn junction is called a junction diode or simply diode, which may be connected to a power supply in two different ways. When p-side is connected with positive terminal to the source and n-side to its negative terminal, is called forward biasing. In this connection, holes on p-side and electrons on n-side get repulsion by the positive and negative terminals, respectively. If the potential difference V across the junction is greater than the barrier potential V_0, holes from p-side and electrons from n-side cross the barrier and produces the forward current I. This forward current and applied potential difference are related by

$$I = I_s \left(e^{\frac{qV}{nK_BT}} - 1 \right) , \qquad (1)$$

where, q is the charge of electron, K_B is Boltzmann's constant, T is absolute temperature of the junction and $n = 1$ for Germanium (Ge) cryatal and 2 for Silicon (Si) crystal. The revers saturation current is denoted by I_s. Another type of connection, called reverse bias connection when p-side is connected with the negative terminal and n-side to the positive terminal of the source. In this connection, holes from p-side and electrons from n-sides are attracted by the negative and positive terminals, respectively. Only, few electrons from p-side and holes from n-side get repulsion and diffuse across the junction and produces very small amount of reverse saturation current I_s.

In this experiment we use **both** Ge and Si diode in forward bias. At the room temperature, $(T \approx 300K)$, and $V = 0.5V$ to $1V$, $e^{\frac{qV}{nK_BT}}$ varies from of the order 10^4 to 10^8 REF and thus $e^{\frac{qV}{nK_BT}} >> 1$. This enables us to neglect 1 compared with the exponantial term and approximate the Eq. (1) as

$$I = I_s \, e^{\frac{qV}{nK_BT}} \, , \tag{2}$$

The above equation can be written after taking the logarithim of both side

$$ln(I) = \frac{qV}{nK_BT} + ln(I_s) \, , \tag{3}$$

which is the equation of a straight line

$$y = m\,x + c \, . \tag{4}$$

In the above equation, $m = \frac{\triangle y}{\triangle x}$ is the slope of the straight line and c denotes the length where the straight line cuts the y-axis. Now comparing, the Eqs. (3) and (4), one can write

$$\begin{aligned} y &= ln(I) \, , \\ x &= V \, , \\ m &= \frac{q}{nK_BT} \, , \\ c &= ln(I_s) \, . \end{aligned}$$

In the experiment, we plot graph of $ln(I) - V$ and determine the slope $m = \frac{\triangle ln(I)}{\triangle V}$. This value of the slope m is equal to $\frac{q}{nK_BT}$. Therefore,

$$K_B = \frac{\frac{q}{nT}}{m} \, . \tag{5}$$

Eq. (5) is our working formula for Determination of the value of the Boltzmann's constant.

3.3 Temperature sensing using RTD (PT100) Sensor:

Resistor Temperature Detectors (RTD) are temerature sensors that sense change of temperature by measuring the change in the value of a resistor. Many RTD elements consist of a length of fine wire wrapped around a ceramic or glass core but other types constructions are alsocommon. The RTD wire is a pure material, typically platinum, nickel, or copper. The material has an accurate resistance/temperature relationship which is used to provide an indication of temperature. As RTD elements are fragile, they are often housed in protective probes, mostly covered in a steel tube that is hermatically sealed.

Platinum, a noble metal and having the most stable resistance–temperature relationship over the largest temperature range, was proposed by Sir William Siemens as an element for a resistance temperature detector back in 1871. Figure (5) shows a schematic of a common wire-wound Platinum Resistor Thermal sensor.

One important property of metals that is used to construct the resistive elements of RTDs is the linear approximation of the resistance versus temperature relationship between 0° C and 100° C. This temperature coefficient of resistance is denoted by α and having units of $\Omega/(\Omega°C)$:

$$\alpha = \frac{R_{100} - R_0}{100 \times R_0} \tag{6}$$

where R_{100} is the resistance of the metal at 100° C, and R_0 is the resistance at 0° C.

Typically, industrial PRTs have a nominal alpha value of $\alpha = 3.85 \times 10^{-3}$ per ° C.

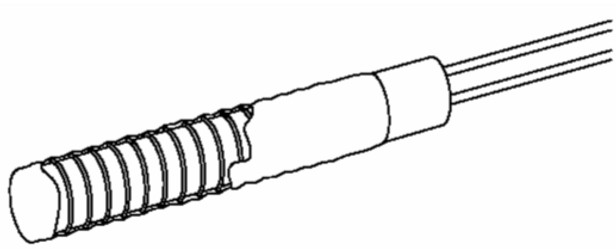

Figure 5: Schematic of a PRT.

Eq. (5) shows that the Boltzmann Constant K_B is a function of temperature in absolute scale. We have used a PT100 Platinum Ressistance Thermometer to measure the temperature of the P-N Junction diode during the experiment, and then convert the masured temperature to absolute scale. PT100 has a typical resistance value of 100Ω at the 0° C. The Resistance Temperature relation of a PT100 sensor for temperatures greater than 0° C is given by the Callendar–Van Dusen equation

$$R_t = R_0(1 + At + Bt^2) \tag{7}$$

where, t=temperature, R_t= resistance at temperature t, and R_0= resistance at 0° C.

For industrial grade of PRT, standard EN 60751:1995 provides values for the coefficients from the value of α of Eq (6) as

$$A = 3.9083 \times 10^{-3} {}^\circ C^{-1},$$
$$B = -5.775 \times 10^{-7} C^{-2}$$

Since the coefficient B is very small, the resistance changes almost linearly with the temperature.

For positive changes in temperature, solution of the quadratic equation using the famous Sreedhar Achraya formula yields the following relationship between temperature and resistance:

$$t = \frac{-A + \sqrt{A^2 - 4B(1 - \frac{R_t - R_{offset}}{R_0 - R_{offset}})}}{2B} \tag{8}$$

PT100 sensors comes with a length of wire, and thus, the resistance of the wire is added as a offset value with the measured temperature, which must me subtracted from the values of the measured resistances as shown in the equation (8). This equation is used in the python program to determine the temperature of the diode during the experiment.

3.4 Curve fitting to find the Boltzmann Constant:

Equation (3) shows that a straight line can be found by plotting the $Log_{10}I_d$ vs V_d data. We must fit a starigt line that is best fit with the experimental data, find its slope(m) and intercept(c) to find the values of K_B and I_s.

Linear regressions of x and y, a Least Squares Regressions Method for fitting curve is used here. In this method, a straight line is fitted by the means of minimizing the vertical distances between the actual data points and the straight line. The coefficients of an equation analogous to equation (4), $y = a_1x + a_0$ are calculated from the following relations:

$$a_0 = \frac{\sum y_i \sum x_i^2 - \sum x_i \sum x_i y_i}{n \sum x_i^2 - (\sum x_i)^2} \tag{9}$$

$$a_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \tag{10}$$

where m and c are analogous to a_1 and a_0, respectively. x_i and y_i are the deviations between the experimental data and fitted line points. The details of the expressions are out of the scope of discussions of this article, and can be found in [4].

4 Experimental Set-up:

The whole experimental set-up consists of two parts, namely,

- Hardware, i.e., the ExpEYES-17 kit and the Circuit for the experiment, and
- Python Program, i.e., the Software for the experiment

The parts are elaborated below in the following sections.

4.1 Circuit for the experiment:

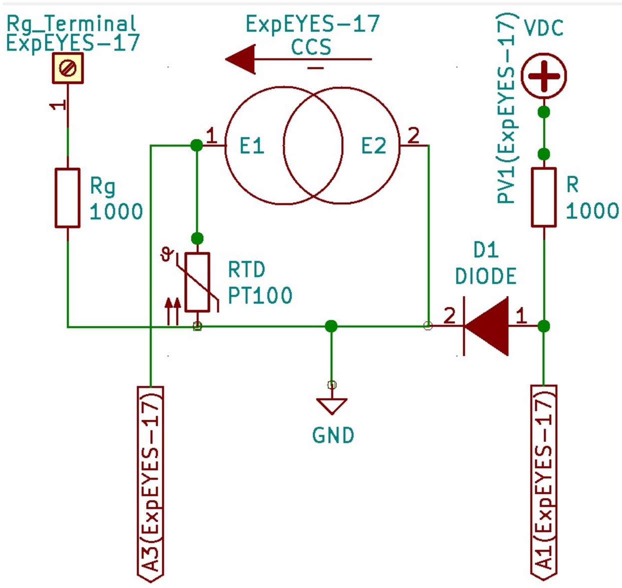

Figure 6: Circuit for finding Diode Characteristics for K_B.

The ExpEYES-17 is the main equipment used in the experiment to generate and measure the required signals. A simple circuit, as shown in the left hand side with respect to ground in the figure (6), is constructed with the help of wires and crocodile clips. We have used two diodes in this experiment. Diode 1N4148 is a Silicon diode, and 1N60 is a Germanium diode. The diode under test is shown as D1. The diode is biased through a resistance R1. The purpose of using the resistor is to determine the diode current using an equation, as ExpEYES-17 cannot measure current directly. The voltage accross the diode is sensed by the Channel A1 of the ExpEYES-17 while the diode gets dc bias from the variable voltage source PV1. PV1 is controlled by the python program listed in Listing (1).

The left hand side of the figure (6), with respect to the ground, shows the schematic of the connection of PT100 temperature sensor. In ExpEYES-17, a constant current source is available for use via the terminal CCS. It is designed to deliver 1 mA of constant current, however, due to tolerances of the components used, this current varies a little bit from device to device. The PT100 is connected between the CCS and the ground terminal, and the voltage caused by the flow of the costant current generated by CCS accross the PT100 is sensed via the A3 terminal of the ExpEYES-17 kit. The A3 terminal can sense $\pm 3.3V$and has a high input resistance of $10M\Omega$. A non-inverting amplifier built around a TL082 OP-AMP in the ExpEYES allows the gain of the A3 A3 input to be set by connecting a resistor from terminal R_g to ground, to ground, as given by equation (11)

$$Gain = 1 + \frac{10000}{R_g} \qquad (11)$$

The temperature obtained by this arrangement is used to compute the value of the Boltzmann constant using the python programs.

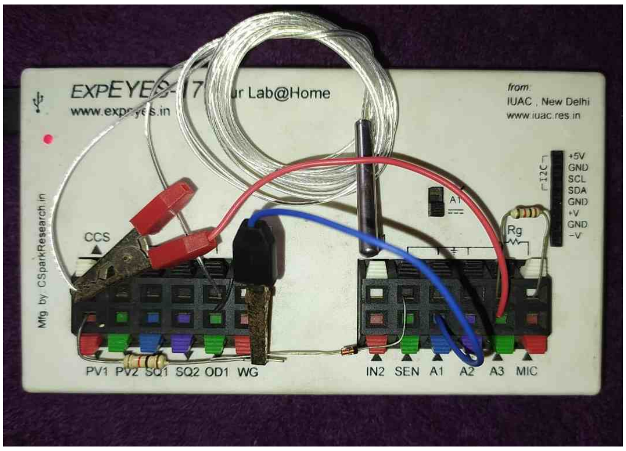

Figure 7: Actual circuit with the ExpEYES-17 for the experiment. The n side of the 1N60 Diode is connected to ground, and p side to PV1 via a $1K\Omega$ resistor. The black clip takes the V_D to A1. The PT100 is connected between CCS and ground, the red clip takes the V_t to A3. Another $1K\Omega$ resistor is connected between R_g and ground.

Figure (7) shows the actual connection of the circuit on the ExpEYES-17 kit.

4.2 Software for the experiment:

The software for this program is nothing but a couple of programs written in python. Libraries specific to ExpEYES-17 are included in the program, and the libraries required to plot the diode characteristics, transfer the data from one program to another, and to curve fit the plots.

4.2.1 Program 1:

The first Python program (1) generates a list of diode voltages and currents by varying PV1. The range of the non-linear section is selected using `vdliml` and `vdlimh` since the logarithm of the non-linear region results in a linear plot. Each voltage-current reading is accompanied by temperature data from the PT100 sensor, converted to absolute values and stored in `latemp[]` for averaging in program (2).

This program produces two sets of lists: `vda[]` and `ida[]` for plotting in figures (9) and (10), and `lvda[]` and `lida[]` for calculating K_B in program (2). The Callendar–Van Dusen equation (7) is used to compute temperature from PT100 resistance changes, accounting for device offsets. The wire resistance at 0°C, denoted as r_{offset}, is used for calibration. Data transfer between the two programs is handled via text files using the `pickle` library.

4.2.2 Program 2:

The second Python program (2) processes the voltage-current lists from program (1) to plot $\ln(I)$ vs. V and determine the slope $m = \frac{\Delta \ln(I)}{\Delta V}$ through curve fitting (3.4). The resulting straight-line fit, located in the fourth quadrant with a positive slope, is used to compute the Boltzmann constant separately for Si and Ge diodes.

Unlike assuming room temperature, this program utilizes temperature readings recorded in program (1) to compute an accurate average. The calculated K_B values are compared with standard values, and the percentage error is determined. Additionally, the logarithm of the reverse saturation current, $\log_{10} I_s$, is derived from the y-axis intercept.

Figure (8) shows typical output for a Silicon Diode.

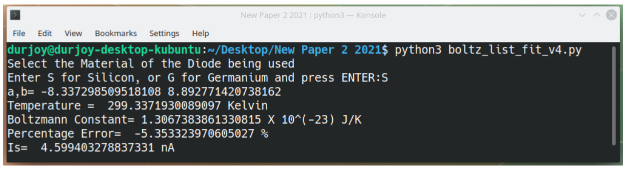

Figure 8: Typical Program output, as given out by program (2) for a Si diode 1N4148

5 Results:

Plots in figures (9) and (10), show the VI characteristics of a Ge and a Si diode. The data generated to plot the curves are used to compute the Boltzmann constant.

Figures (11) and (12) show the plots of the experimental data and the linear fitted curve for the Ge and Si Diodes, respectively.

Figure (12) shows the screenshot of the program output as well with the plot. The calculated values of K_B and I_s along with the average temperature in Kelvin, are displayed as output.

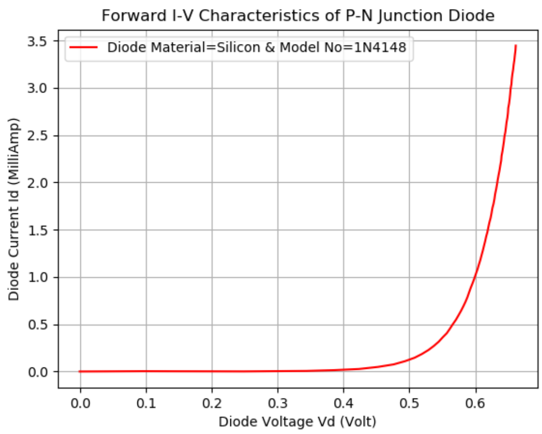

Figure 9: Ge (1N60) Diode Characteristics for K_B generated running Program (1).

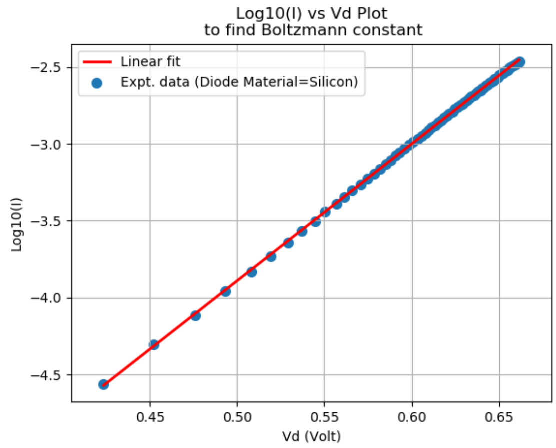

Figure 11: $Log_{10}I_s$ plot (Scatter), and fitted curve (Line) plot for Si (1N4148) Diode to find K_B, generated running Program (2).

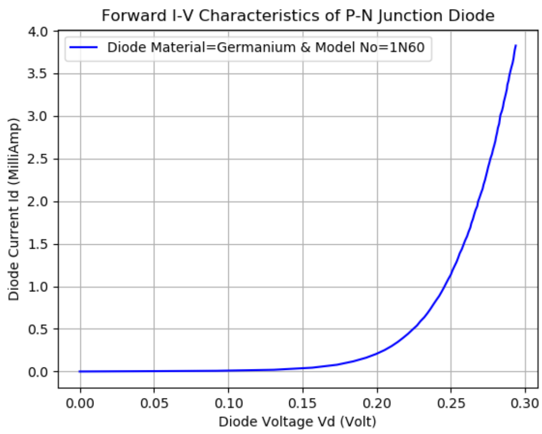

Figure 10: Si (1N4148) Diode Characteristics for K_B, generated running Program (1).

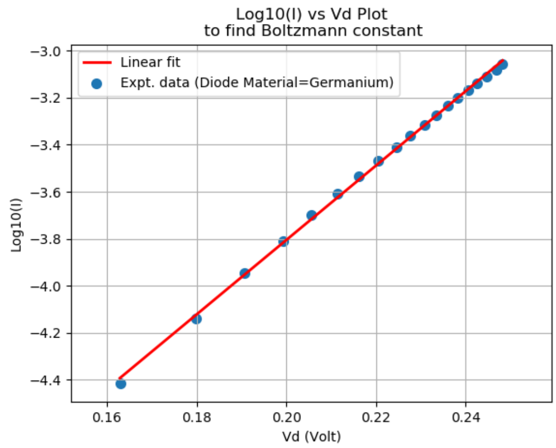

Figure 12: $Log_{10}I_s$ plot (Scatter), and fitted curve (Line) plot for Ge (1N60) Diode to find K_B, generated running Program (2).

The percentage error of measurement δ is also calculated by the program using the given relation:

$$\delta = \frac{v_A - v_E}{v_E} \times 100\% \qquad (12)$$

Where v_A and v_E are the actual and exact value of any parameter being measured. Table (1) tabulates the value v_A obtained by the experiment, and the value of the percentage error δ in each case, given the exact value v_E of the Boltzmann Constant is $K_B = 1.38 \times 10^{-23} JK^{-1}$.

Table (2) tabulates the values of the reverse saturation current I_S for each case. It appears from the data that in case of

Table 1: Data recorded for K_B

Model	Mat.	v_A of K_B in JK^{-1}	δ (%)
1N60	Ge	$1.47 \times 10^{-}23$	6.72
1N4148	Si	$1.30 \times 10^{-}23$	-5.35

Table 2: Data recorded for I_S

Model	Mat.	I_S in nano Amperes
1N60	Ge	114.11
1N4148	Si	4.59

Germanium, the amount of I_S is higher than that of a Si diode.

6 Discussions:

The experiment designed here is not new to undergraduate physics students, but the approach is. This method may provide an additional way to solve the problem, complementing the traditional hands-on experiment method, where each data point is collected manually. The use of *python* is another interesting approach to design such age-old experiments in a new way. The authors are at opinion that the use of ExpEYES-17 in the undergraduate laboratories along with the traditional methods can definitely widen the horizon of learning.

Acknowledgments

The authors thank all those who have developed and contributed to the design of the kit, Especially, Dr. Ajith Kumar B. P., Satyanarayana V. V. V. of IUAC, and Jithin B. P. of Himachal Pradesh Open University. The book[3] on python has helped the authors a lot in coding in python.

Appendix

The Program listings are provided below.

```
# Data Collection for finding Boltzmann Constant

#Initialization of device
import eyes17.eyes
p = eyes17.eyes.open()
from pylab import *
import numpy as np
import matplotlib.pyplot as plt

#import libraries
import time
import math
```

```python
import pickle

#Get the Material of the Diode being used
mat=input ("Type S for Silicon, or G for Germanium and press ENTER:")
if (mat=='S'):
    eta=2
    mat1='Silicon'
    colname='red'
    vdliml= 0.40        #the lower limit of the exponential part of the V-I plot.
    vdlimh= 0.75        #the upper limit of the exponential part of the V-I plot.
elif (mat=='G'):
    eta=1
    mat1='Germanium'
    colname='blue'
    vdliml= 0.15        #the lower limit of the exponential part of the V-I plot.
    vdlimh= 0.25        #the upper limit of the exponential part of the V-I plot.
else:
    print('Diode Material not defined')

#Get the Number of the Diode being used
mod=input ("Type the diode model number and press ENTER:")

#define constants for Temperature recording
A = 3.9083e-3                       #value of coefficient A
B = -5.7750e-7                      #value of coefficient B

#measured values of offset and constant current source
a3offset = 0.0005908659726263022    #offset voltage at a3
iccs = 0.0011132813656059932        #ccs current in A

#other parameters for temperature recording
Rg = 977                            #value of 1k resisor
gain = 1 + 10000/Rg                 #gain at a3 using Rg
r0 = 101.8                          #pt100 resistance at 0 deg c

```

```python
#Finding the data for plot and calculation
vda = []          # Initiate Diode voltage list
ida = []          # Initiate Diode current list
lida = []         # Initiate log I list
lvda = []         # Initiate list for next plot
latemp = []       # Initialize list for Absolute Temperature recording

pv1=0.0
while (pv1 <= 4.00):
    p.set_pv1(pv1)                  #set bias voltage
    p.set_state(CCS=1)              #enable ccs
    time.sleep(0.025)               #give the device time to set the voltage/current
    vs = pv1                        #check bias voltage
    vd = p.get_voltage('A1')        #get Diode voltage
    va3 = p.get_voltage('A3')       #Read A3 for temperature measuremnent
    vt = va3 - (gain * a3offset)    #correct offset in measuring A3
    vpt100 = vt/gain                # voltage across PT100
    rt = vpt100 / iccs              #voltage for curent teperature
    C = 1 - (rt - 1.8)/(r0 - 1.8)   #find c and exlude offset resistor of pt100 wire
    ctemp = (-A + math.sqrt( A*A - 4 * B * C) )/ (2.0 * B)    #find current temperature
    atemp = 273.15 + ctemp          #find temperature in absolute scale
    idx = (vs-vd)/969               #find Diode current for 1K Ohm in Amp
    idma = idx*1000                 #current in mA for I-V plot
    if vdliml <= vd <= vdlimh:      #non-linear section of the plot
        if idx > 0 :                #discard negative value, if any
            idlog = math.log10(idx) #convert I to log
            lida.append(idlog)      #append in the list of id
            lvda.append(vd)         #append in the list of vd
            latemp.append(atemp)    #append in the list of absolute temperature
    vda.append(vd)                  #append in list for vd vs id plot
    ida.append(idma)                #append in list for vd vs id plot
    p.set_state(ccs=0)              #disable ccs
```

```python
    pv1 += 0.05                         #next value of source voltage


#save the lists in a file to calculate the Boltzmann Constant
with open("ivdata.txt", "wb") as f:
    pickle.dump(lvda, f)                #dump the vda values in a text file
    pickle.dump(lida, f)                #dump the log(10)ida values in a text file

#save the temeratures list in a file to calculate the Boltzmann Constant
with open("tempdata.txt", "wb") as f:
    pickle.dump(latemp, f)              #dump the temperature values in Kelvina in a text file

#Plot
plt.title('Forward I-V Characteristics of P-N Junction Diode')  #Plot Title
plt.xlabel('Diode Voltage Vd (Volt)')        #x axis label
plt.ylabel('Diode Current Id (MilliAmp)')    #y axis label
plt.plot(vda, ida, color= colname, label='Diode Material='+str(mat1)+' & Model No='+str(mod)) #plot graph
plt.grid(True)                               #show gridlines
plt.legend(loc='upper left')                 #show legend
plt.show()                                   #show graph
```

Listing 1: Python Program to generate data to find Boltzmann Constant using p-n junction diode, and to plot the diode characteristics.

```python
# Plotting of V - Log10I plot and calculation of Boltzmann Constant

#import libraries
import matplotlib.pyplot as plt
import scipy.optimize as opt
```

```
import numpy as np
import pickle
from math import *


#Get the Material of the Diode being used
print('Select the Material of the Diode being used')
mat=input ("Enter S for Silicon, or G for Germanium and press ENTER:")
if (mat=='S'):
    eta=2
    mat1='Silicon'
    colname='red'

elif (mat=='G'):
    eta=1
    mat1='Germanium'
    colname='blue'

else:
    print('Diode Material not defined')

q = 1.602176634e-19                         #standard value of unit charge
ks = 1.380649e-23                           #standard value of k

with open("ivdata.txt", "rb") as f:         # Unpickling V-log10I data
    Vd = pickle.load(f)
    Id = pickle.load(f)

with open("tempdata.txt", "rb") as f:       # Unpickling Temperature data
    atemp = pickle.load(f)
    at=sum(atemp) / len(atemp)                 # Find avg value of Temperature in Kelvin

#intialization for curve fitting
x,y=[],[]
for i in range(len(Vd) ):
    x.append(Vd[i])
    y.append(Id[i])
n=len(x)
```

```

## Linear equation : y=a+bx fit
def func(x,a,b):
    return a+b*x
sx=0.0
sy=0.0
sxy=0.0
sx2=0.0
for i in range(n):
    sx=sx+x[i]
    sy=sy+y[i]
    sxy=sxy+x[i]*y[i]
    sx2=sx2+x[i]*x[i]
D=n*sx2-sx*sx
A=sy*sx2-sx*sxy
B=n*sxy-sx*sy
a=A/D
b=B/D
print ("a,b=",a,b)

fx=[]
for i in range(n):
    fx.append(func(x[i],a,b))


k = q/(2.303 * eta * (at) * b)        #find k from fitted curve
pe = ((k - ks)/ks)*100
Is = 10 ** a
print("Temperature = ", at, "Kelvin")#print temperature during experiment
print("Boltzmann Constant=", k/1.0e-23, "X 10^(-23) J/K") #print the value of k
print("Percentage Error= ", pe, "%")
print("Is= ", Is*1.0e9, "nA") # Print Is

## Plot
plt.title('Log10(I) vs Vd Plot\n to find Boltzmann constant')
plt.xlabel('Vd (Volt)')
plt.ylabel('Log10(I)')
```

```python

plt.scatter(x, y, label="Expt. data"+" (Diode Material="+str(mat1)+")",linewidth=2)
plt.plot(x, fx, "red", label="Linear fit", linewidth=2)
plt.grid(True)
plt.legend(loc="upper left")
plt.show()
```

Listing 2: Python Program to find Boltzmann Constant using curve fitting of a straight line based on the data obtained from program (1)

References

[1] Kumar, A. B. P., and Satyanarayana, V. V. V. (2017). *The ExpEYES-17 User's Manual*. Inter-University Accelerator Center, New Delhi.

[2] Kumar, A. B. P. *Python for Education*. The Phoenix Project, Inter-University Accelerator Center, New Delhi.

[3] Brown, M. C. (2018). *Python: The Complete Reference* (4th ed.). McGraw Hill.

[4] Rajaraman, V. (1999). *Computer Oriented Numerical Methods* (3rd ed.). Prentice-Hall India, pp. 119-121.

[5] Singh, J. (2009). *Semiconductor Devices: Basic Principles*. Wiley India, pp. 476-477, ISBN 9788126511020.

[6] Kumar, A. B. P., et al. (2009). "Innovative Science Experiments Using Phoenix." *Physics Education*, **44**, 469.

[7] Luthra, V., Kaur, A., Saini, S., Jaglan, S., Dhasmana, S., Apoorva, and Kumar, A. B. P., 2016, "Evaluation of Boltzmann's constant: Revisit using interfaced data analysis," *Physics Education*, **32**(3), Article 2, Jul–Sep 2016.

[8] Kumar, A. B. P. (2018). "Innovative Science Experiments Using ExpEYES-17." *IAPT Physics Education*, **35**(2), 32-40.

[9] Satyanarayana, V. V. V. (2019). "Using ExpEYES-17 for Teaching Basic Physics Concepts." *IOP Physics Education*, **54**(6), 065015.

[10] Roy, D. (2020). "Junction Field Effect Transistor Characteristics: A New Approach Using ExpEYES-17." *Physics Education*, **36**(3), April-June, ISSN 0970-5953.

[11] Biswas, S., 2022, "Determination of the band gap of germanium and silicon using ExpEYES-17 kit," *Physics Education*, **57**(2), 025026. https://doi.org/10.1088/1361-6552/ac3b95

[12] Biswas, S., 2022, "Study of Fourier series of user-defined waveforms us-

ing ExpEYES-17 kit," *Physics Education*, **57**(3), 035008. https://doi.org/10.1088/1361-6552/ac420a

[13] Biswas, S., and Roy, D., 2022, "Microcontroller-based study of diode thermometers for online demonstration of undergraduate laboratory classes in COVID-19 lockdown," *Physics Education*, **57**(4), 045011. https://doi.org/10.1088/1361-6552/ac563f

[14] Roy, A., Roy, D., and Mallick, A., 2023, "Construction and remote demonstration of an inexpensive but efficient linear differential variable transformer (LVDT) for physics or electronics teaching during COVID-19 pandemic," *Physics Education*, **58**(1), 015007. https://doi.org/10.1088/1361-6552/ac93de

[15] Roy, A., Ghosal, M., Mallick, A., Roy, D., and Biswas, B., 2024, "Construction and remote demonstration of an inexpensive but efficient experimental setup for studying self-inductance and mutual inductance between two coils," *Physics Education*, **59**(2), 025022. https://doi.org/10.1088/1361-6552/ad22f3

[16] CERN OHL. Available at: https://ohwr.org/cernohl.

[17] ExpEYES-17 Project Website. Available at: https://www.expeyes.in.

[18] HyperPhysics, Georgia State University. Available at: http://hyperphysics.phy-astr.gsu.edu/hbase/Solids/pnjun.html.

[19] Inter-University Accelerator Center Phoenix Project. Available at: https://www.iuac.res.in/phoenix/.

[20] Nobel Prize in Physics 1918. Available at: https://www.nobelprize.org/prizes/physics/1918/planck/biographical/.

[21] Wikipedia. Available at: https://commons.wikimedia.org/w/index.php?curid=643178.