

Automation of Spectrometer

A. Robson Benjamin,

Department of physics, The American College, Madurai, South India.

-----ABSTRACT-----

In this paper the spectrometer normal incidence mode experiment is automated using the recent techniques. The data from the spectrometer is taken with the help of a microcontroller and sent to the personal computer using serial communication for further analysis. Photo transistor is used as detector. Host software is written using Python which controls the stepper rotation and data analysis.

KEYWORDS;- Spectrometer, Photo transistor, Python, LPC1768

Date of Submission: 25-02-2020

Date of Acceptance: 05-03-2020

I. INTRODUCTION

Automation provides the advantage of improving quality by reducing errors. Automation not only reduces the time for completing an experiment but also gives flexibility to process the acquired data. Using the recent techniques the spectrometer normal incidence mode experiment is automated. The data from the spectrometer is collected with the help of a microcontroller and sent to the personal computer using serial communication for further analysis.

II. DESCRIPTION OF THE EXPERIMENTAL SETUP

The apparatus consists of a spectrometer, stepper motor, microcontroller, photo-transistor and computer. The experiment is done under the normal incidence mode. The photo-transistor is used as a sensor to view the lines in the diffraction pattern instead of the human eye. A PVC pipe is attached to the circular mount with M-Seal as shown in figure 1.1. Two lens are kept inside the pipe (in contact $1/f = 1/f_1 + 1/f_2$), along with the photo-transistor which can be moved to and fro for focusing. The photo-transistor is the detector. The friction between the moveable parts in the spectrometer is reduced by placing a couple of transparent sheets cut in the form of a ring to allow them to pass over the central pillar on which the prism table rotates.



FIGURE 1.1 Photograph of detector placed in a PVC pipe on a spectrometer

The belt with grooves (removed from a printer) is pasted on the spectrometer rotator table to make it a toothed wheel. The resolution of this apparatus is 6 ". To receive the signal from the photo transistor, a shielded cable is used as shown in figure 1.2. The detector (photo transistor) is connected to the data logger through biasing circuit. The readings are stored in the computer from the data-logger through port. The biasing circuit is shown in figure 1.3.

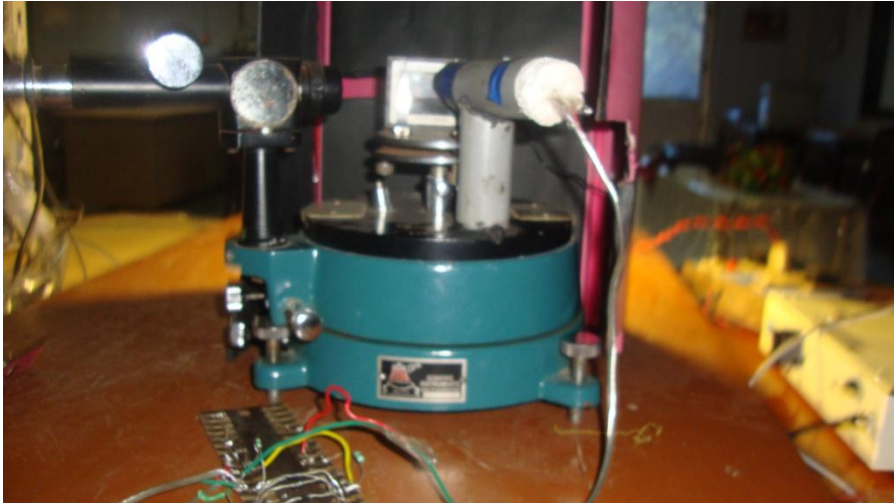


FIGURE 1.2 Photograph of detector placed in a PVC pipe on a spectrometer connected to circuitry with a shielded cable

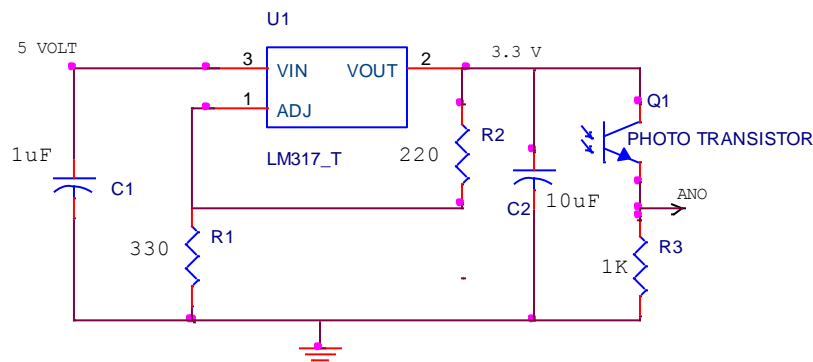


Fig 1.3 Detector Circuit Diagram

III. INTERFACING STEPPER MOTOR AND DATA LOGGER

Stepper motor rotation is controlled by LPC1768 ARM microcontroller. The output voltage is 3.3 volt which is not sufficient to run the stepper motor. Hence a suitable driver circuit is constructed. The driver circuit is shown in figure 1.4. The host software is written in Python program. Serial communication is used to communicate with the host computer. The stepper motor is connected to Port0 of the microcontroller. The serial port of the microcontroller acting as data logger is hooked to the computer with baud rate 9600 (data Logger = serial.Serial ('/dev/ttyS0', 9600, timeout=0.5)). The output data is stored in a text file. To get better data reliability, nsample data is collected by running a loop. Then it is averaged. The experiment is repeated for a reverse sweep. The resolution is .1 degree. When the stepper motor moves, the photo-transistor which is inserted in the pipe also moves in steps. The mercury lamp is used as a source for this experiment.

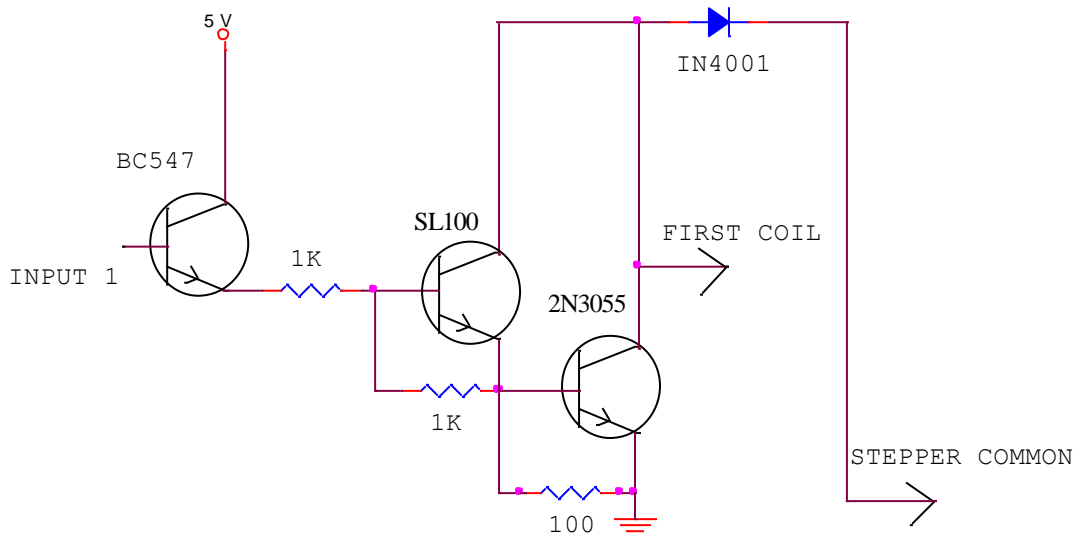


FIGURE 1.4: One segment of the Stepper motor driver circuit.

The firmware includes modules performing the following tasks

i) Initialisation UART communication:

```
import serial
import time
import string
data Logger = serial.Serial ('/dev/ttyS0', 9600, timeout=0.5)
f = open ("test.txt", "w")
```

ii) Receiving the data and averaging the data:

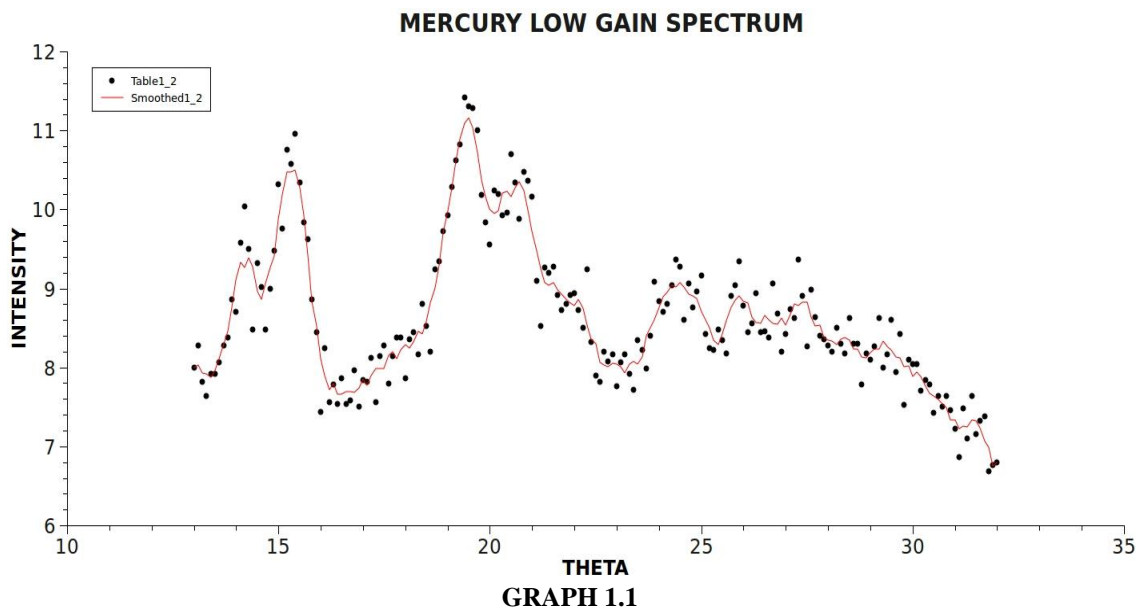
```
sum = 0
for j in range(nsamples):
    time.sleep(delay)
    line = dataLogger.readline()
    if len(line) < 18: line = linep
    linep = line
    sum = sum + string.atof(string.split(line)[0])
sum = sum/nsamples
ctr = ctr+1
line = str(ctr)+" "+str(sum)+"\n"
print line
print >> f.write(line)
```

IV. SPECTROMETER CALIBRATION

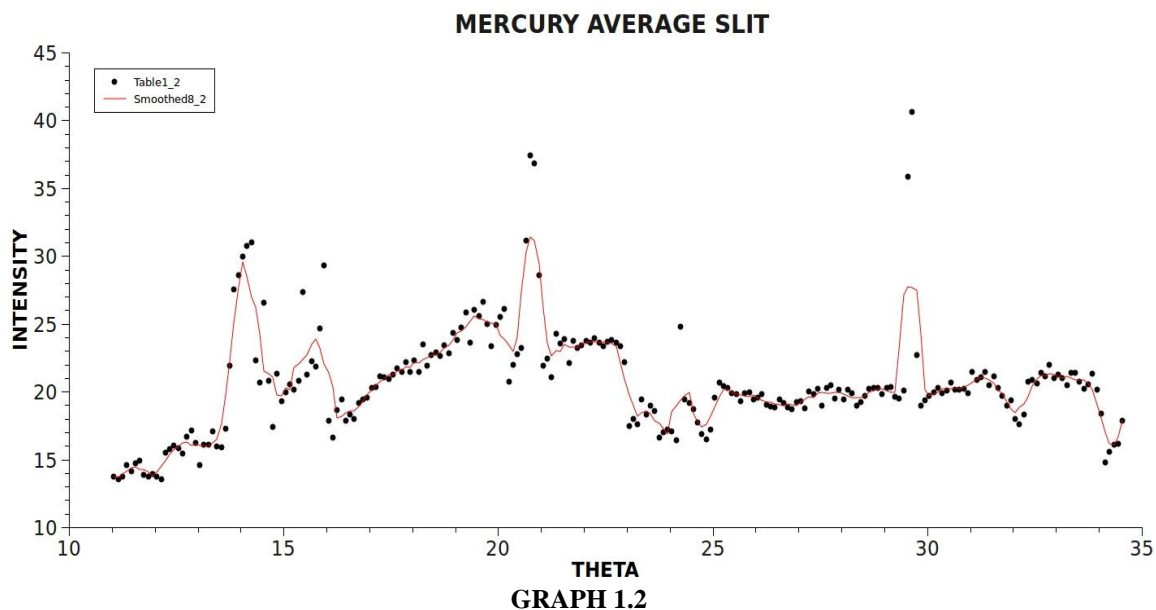
The real telescope of the spectrometer is adjusted for distant objects. The collimator is adjusted to receive parallel rays with the telescope that has been focused for distant objects. The telescope is moved out of the way. The resolution of the spectrometer in this configuration is .1 degree. This has been experimentally verified. The PVC pipe telescope is moved with the stepper motor. When it comes to the direct beam, it reaches a maximum. That is taken as the reference point. If on the other hand the ADC overflows, the mean of the two extreme positions in steps, where overflow just sets in/out, is taken as the zero position or direct beam. Let the direct beam position in steps be db_s . Let any random pos in steps be rps . The random position in angles is $Pos_in_angles = (rps - db_s) * .1$. The spectrometer does 400 steps and then retraces and comes back to start.

V. RESULTS AND DISCUSSION

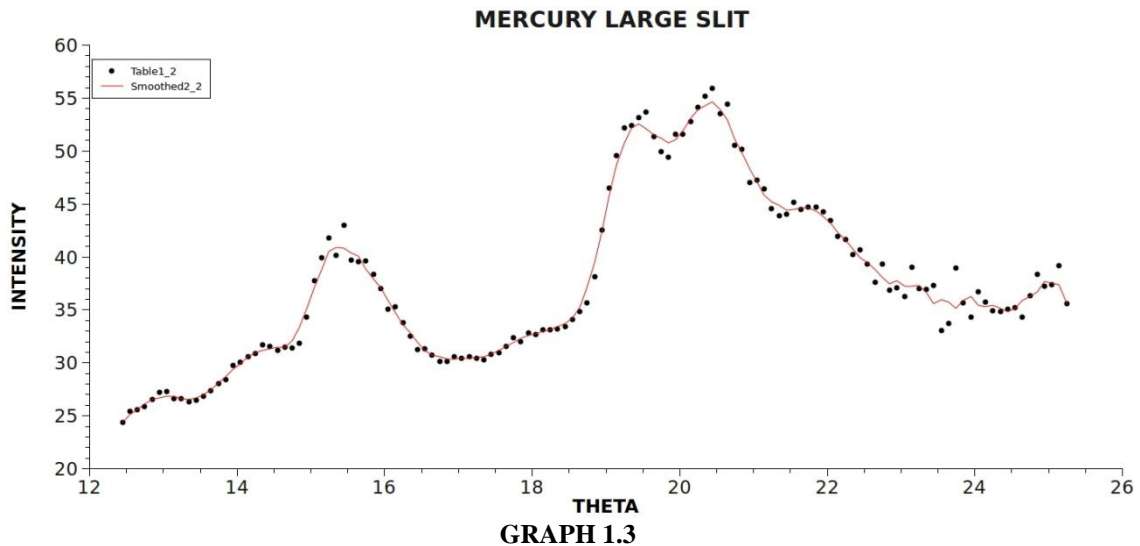
Data from the data logger is acquired in the computer. This data is plotted using QTI plot (available in LINUX) QTI plot is compatible with ORIGIN software in WINDOWS. When the data is non averaged and the detection system has low gain the plot is just a profile which shows up as a scatter [1.1].



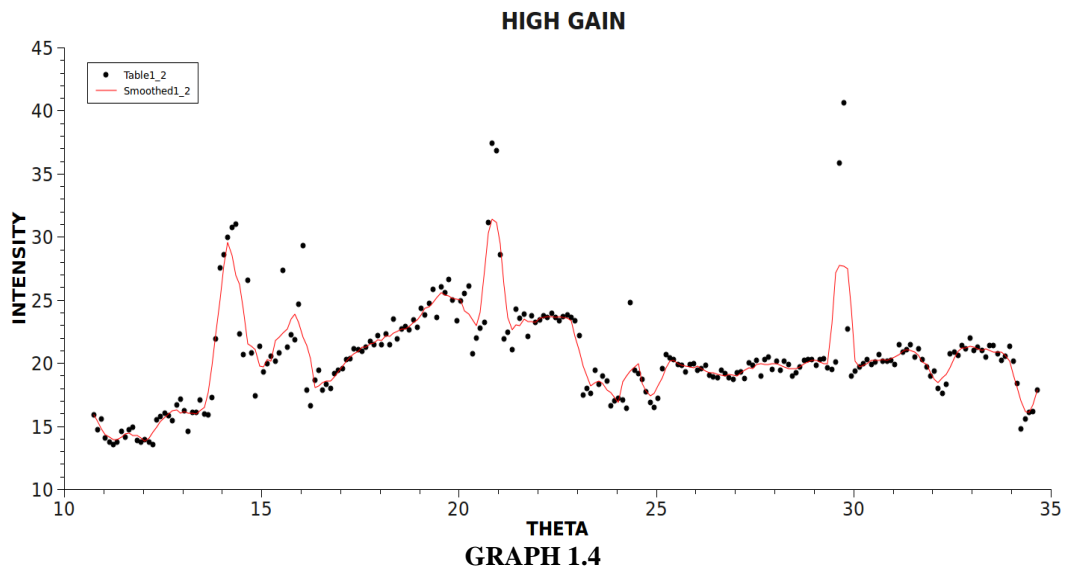
This is improved on averaging. On increasing the gain the line shapes improve and narrow down [1.2].



Increasing the slit width increases the signal strength. However the lines broaden [1.3].



The points are averaged over 5 points each to get a smooth curve. As is seen, a high gain detection system with large averaging and small slit width is ideal for data.



From the above graphs, different theta values are received for different colors. This value is close to the standard value. (6000 lines per cm has been assumed).

COLOUR	THETA Expected	THETA obtained
Violet	14.28 (14.3+/- .1)	14.3+/- .1
Blue	15.29 (15.8 +/- .1)	15.8 +/- .1
Green	19.29 (19. +/- .1)	19. +/- .1
Yellow	20.51 (21. +/- .1)	21. +/- .1
Red	23.53 (24.1 +/- .1)	24.1 +/- .1

Table1.1 Experimental Results

Table1.1 shows that the data obtained is close to the expected value. Hence it is concluded that the calibration is good.

BIBLIOGRAPHY

- [1]. LPC1768 Data Sheet. [Online] [Cited: May 4, 2018.]
- [2]. Design and Realisation of a Portable Data Logger for Physiological Sensing. **R.Luharuka et.al.** June 2003, IEEE Transaction on Instrumentation and Measurement.
- [3]. LPC1768 Data Sheet. [Online] [Cited: May 4, 2018.]
- [4]. <http://www.diffgrating.html>