

**DEPARTMENT OF  
ELECTRONICS AND COMMUNICATION ENGINEERING  
College of Engineering and Technology  
SRM Institute of Science and Technology**

**MINI PROJECT REPORT**

**ODD Semester, 2022-2023**

**Lab code & Sub Name** : 18ECO109J-LAB-E  
Embedded System Design using Raspberry Pi

**Year & Semester** : 2022 / 5<sup>th</sup> Semester

**Project Title** : Temperature Logger

**Lab Supervisor** : **Ms.G.Suganthi Brindha**

**Team Members** :  
1. Prakash Y (RA2011003010044)  
2. Kevin Thomas Koshy (RA2011003010018)  
3. Soubarno Gupta (RA2011003010050)

Particulars	Max. Marks	Marks Obtained
		Name: Kevin Thomas Koshy
		Register No : RA2011003010018
Program and Execution	20	
Demo verification & viva	15	
Project Report	05	
<b>Total</b>	<b>40</b>	

## **Temperature Logger**

### **Objectives:**

To create a temperature logger which records the temperature at a given interval and write them into a log file.

### **Abstract:**

The system on a chip (SoC) of the Raspberry Pi has a temperature sensor that can be used to measure its temperature from the command line. This project's aim is to create a simple Python script that can run automatically as you boot up your Raspberry Pi, take measurements from the temperature sensor at given intervals, and write them into log files that can be viewed later.

This project can be approached in different ways.

We can connect an external temperature sensor to the Raspberry Pi and fetch the data through it or we can use the in built temperature sensor present on the the SoC of Raspberry Pi.

Using the former method will get us more accurate results as the sensor is a separate entity as won't be affected by heating caused by the processor.

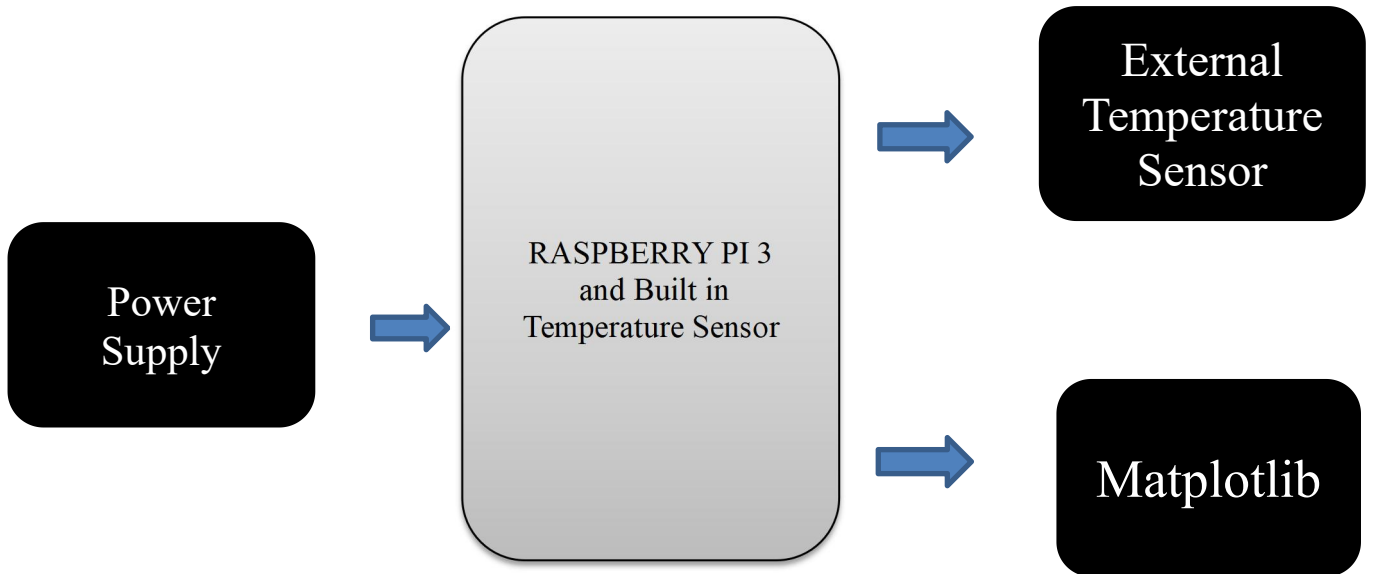
**Hardware/Software Requirements:** You will need this hardware:

- Raspberry Pi Computer

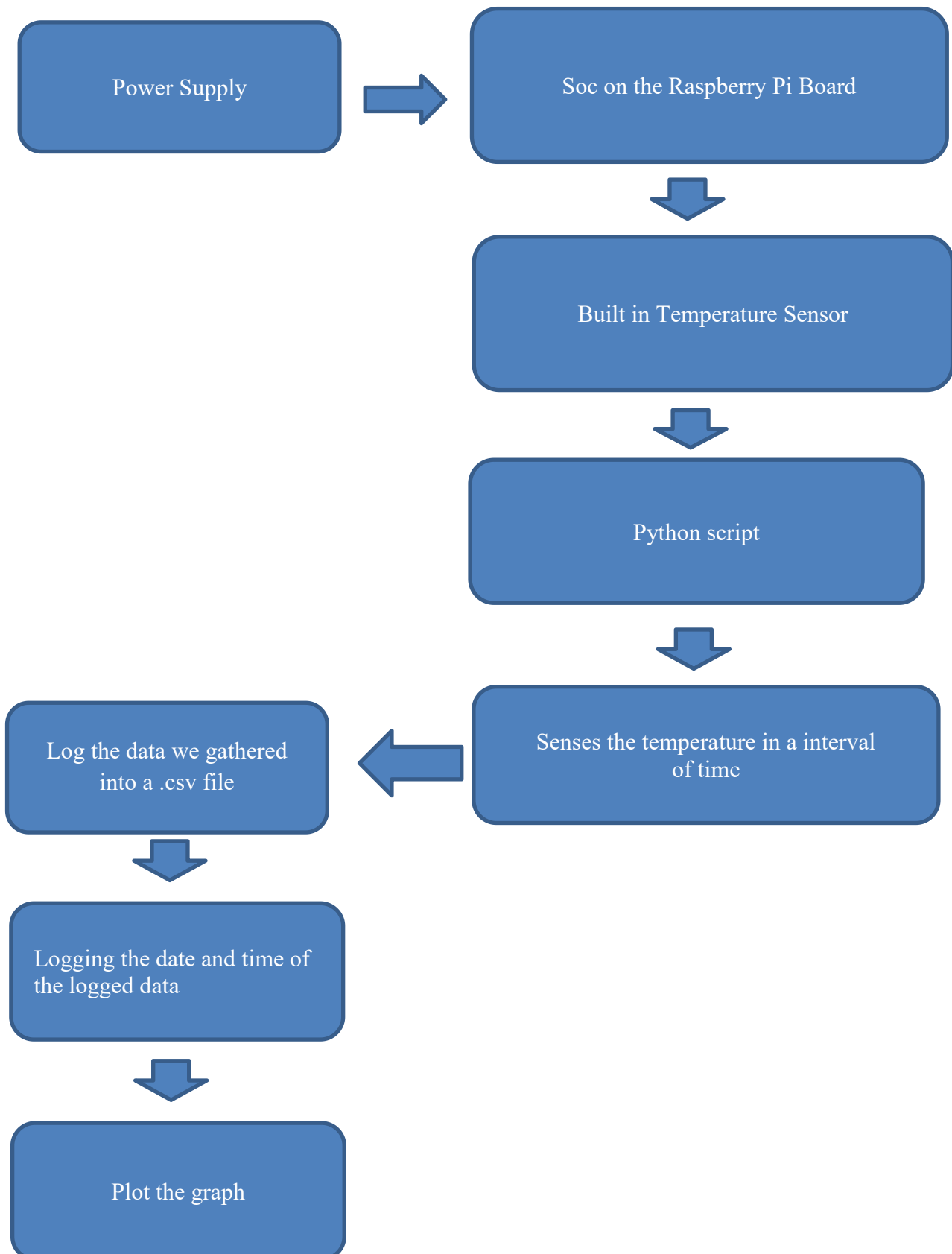
**Software Requirements:**

- Matplotlib

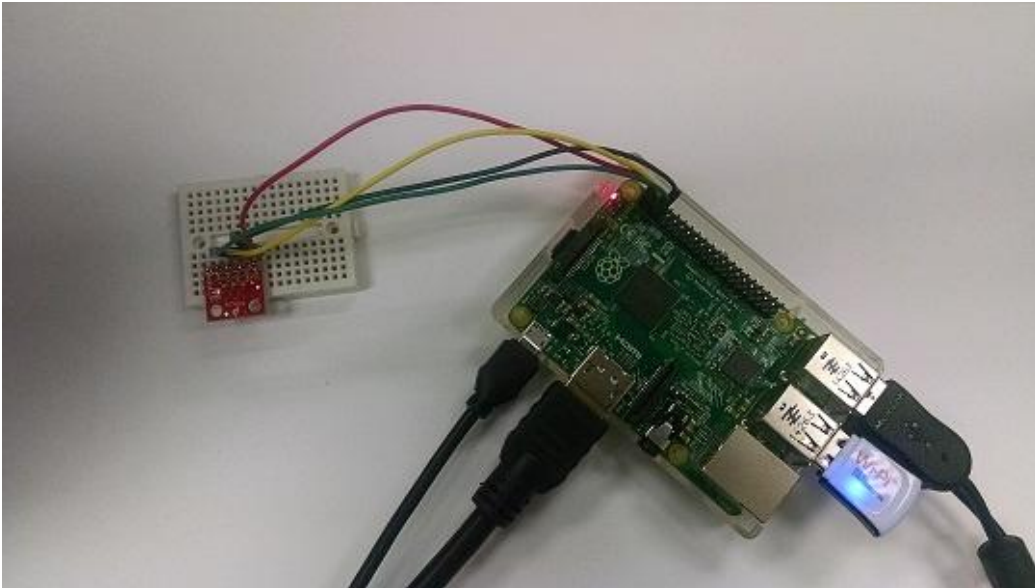
**SIMPLE BLOCK DIAGRAM:**



## DETAILED BLOCK DIAGRAM:



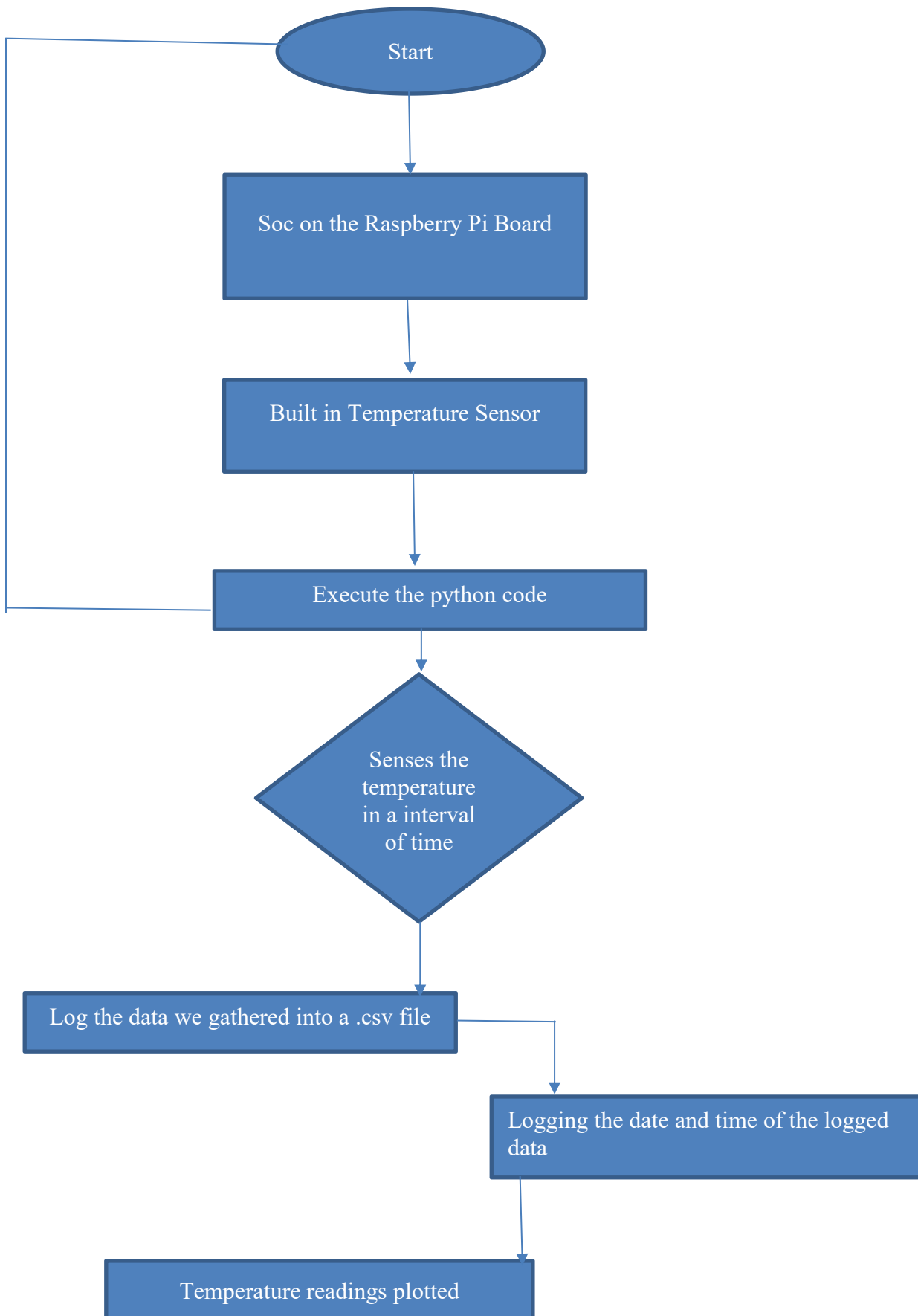
## CONNECTION DIAGRAM:



## Algorithm:

- First, we make all the necessary connections and connect the Raspberry Pi to a monitor and use the Raspbian OS.
- Once we have setup the connection, we have to create a python script for gathering the data.
- We will use the GPIOZero module to import CPUtemperature and use it to log the temperature of the processor.
- We don't have an external sensor so we will be proceeding this way.
- Next we have to log the data we gathered into a .csv file for easy representation.
- We will be importing the time module to log certain characteristics such as strftime.
- We will be logging the date and time of the logged data and using that to chart a graph.
- For plotting the graph, we will be using the matplotlib library.

## Flow Chart:



## Program:

```
from gpiozero import CPUTemperature
from time import sleep, strftime, time
import matplotlib.pyplot as plt

cpu = CPUTemperature()

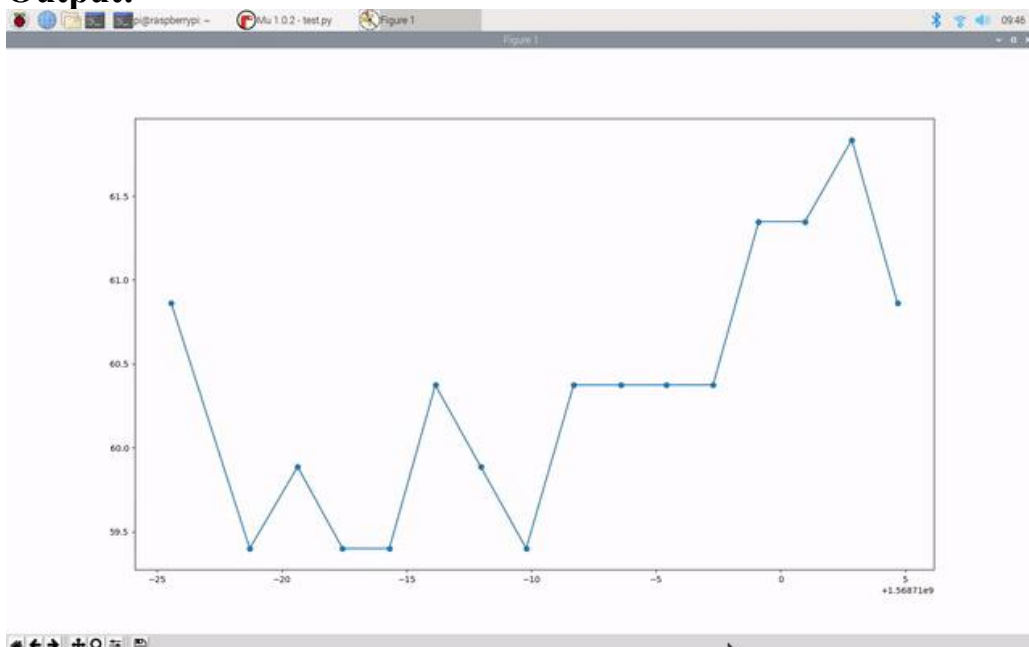
plt.ion()
x = []
y = []

def write_temp(temp):
    with open("/home/pi/cpu_temp.csv", "a") as log:
        log.write("{0},{1}\n".format(strftime("%Y-%m-%d %H:%M:%S"),str(temp)))

def graph(temp):
    y.append(temp)
    x.append(time())
    plt.clf()
    plt.scatter(x,y)
    plt.plot(x,y)
    plt.draw()

while True:
    temp = cpu.temperature
    write_temp(temp)
    graph(temp)
    plt.pause(1)
```

## Output:



## **Real Time Constraints:**

1. Disparate Data and Data Storage Limitations
2. Missed Alarms During Network Failure
3. Risk of Human Error in Setup and Configuration
4. Hidden Costs
5. Limited Parameters, Multiple Systems
6. Battery Power

## **Conclusion:**

Due to time constraints, we were not able to implement this project in the best possible manner such as using dedicated temperature sensors which would have given us more accurate readings.

However the core methodology essentially remains the same. Just that the data will be fetched through another sensor and there would be some extra routing.

The extended versions of this project play an extensive role in today's world by keeping check on ever-changing temperatures and plotting various graphs for easy representation and data tracking.

## **REFERENCES:**

***Published by [Raspberry Pi Foundation](#) under a [Creative Commons license](#).  
[View project & license on GitHub](#)***