Abstract:

1. **Problem Statement:**
   In order to meet the increasing need for computing power in personal computers, we use high-end processors. The CPUs frequently generate a lot of heat, and our external cooling pads are used to help them cool down.
   However, the cooling pads have two main issues:
   - Manual turn on and off as the CPU's usage changes.
   - Static fans that are fixed in place, but vents in different laptop models are located in different places, resulting into power loss because most of the air from the cooling pad does not enter the vents on the laptop.

2. **Solution proposed:**
   The current hardware projects employ a driver to transmit the CPU utilization to the cooling pad, which processes it and activates the cooling pad as needed. Additionally, we used a goose-neck cable to mount exhaust fans so that it may be positioned wherever is appropriate depending on laptop models.
3. **Methodology (what is deployed):**
   We have built an automatic pc cooling pad using python coding language to run the driver and Arduino Uno and gooseneck cables.
4. **How the problem is resolved:**
   - The cooling pad automatically understands when to switch on and off thanks to a Python driver thus saving power.
   - Gooseneck cables allow us to position fans virtually anywhere on the vent, which makes cooling efficiency independent of laptop model.
5. **Conclusion:**
   We will be able to meet the continually rising cooling requirements and save power of personal computers if the suggested solution is put into practice. Additionally, by employing goose neck clamps, we can build a single cooling system that properly operates on all machines.
6. **Future scope:**
   The future scope of this project entails the exploration of strategies to enhance the efficiency and sustainability of Crypto mining operations. Addressing the substantial processing power utilized in Crypto mining, which leads to heightened heat generation within mining rigs and necessitates the continuous operation of cooling mechanisms, becomes a pivotal area of focus. Consequently, the project can delve into innovative cooling solutions that mitigate electricity consumption and subsequently augment overall profitability. Additionally, a promising avenue for further investigation involves the implementation of a targeted cooling approach. By identifying specific mining units exhibiting elevated temperatures, the proposed cooling system could be selectively activated, optimizing the cooling process and potentially refining the operational dynamics of the mining rig. This avenue of inquiry offers the prospect of refining the energy-intensive nature of Crypto mining and holds potential for advancing both its economic viability and environmental sustainability.

Apparatus list:

1. Arduino Uno
2. LCD display (16*2)
3. Relay
4. 2 battery(5watt)
5. Bread Board
6. Jumper wire
7. Gooseneck cables.
8. Cosmic byte cooling pad
9. Laptop stands
10. Resistor (1K Ω)

Required apparatus:

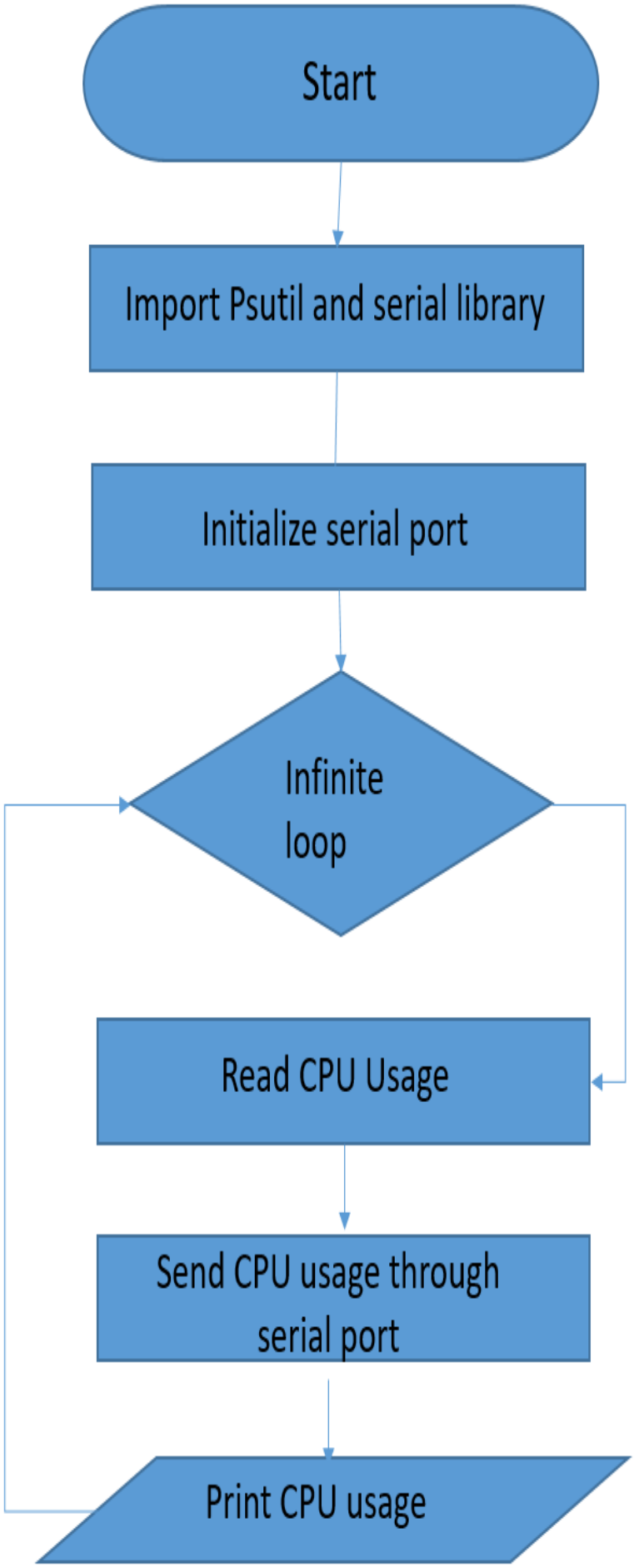
1. Exhaust fans
2. 3D printer filament

Coding Languages used:

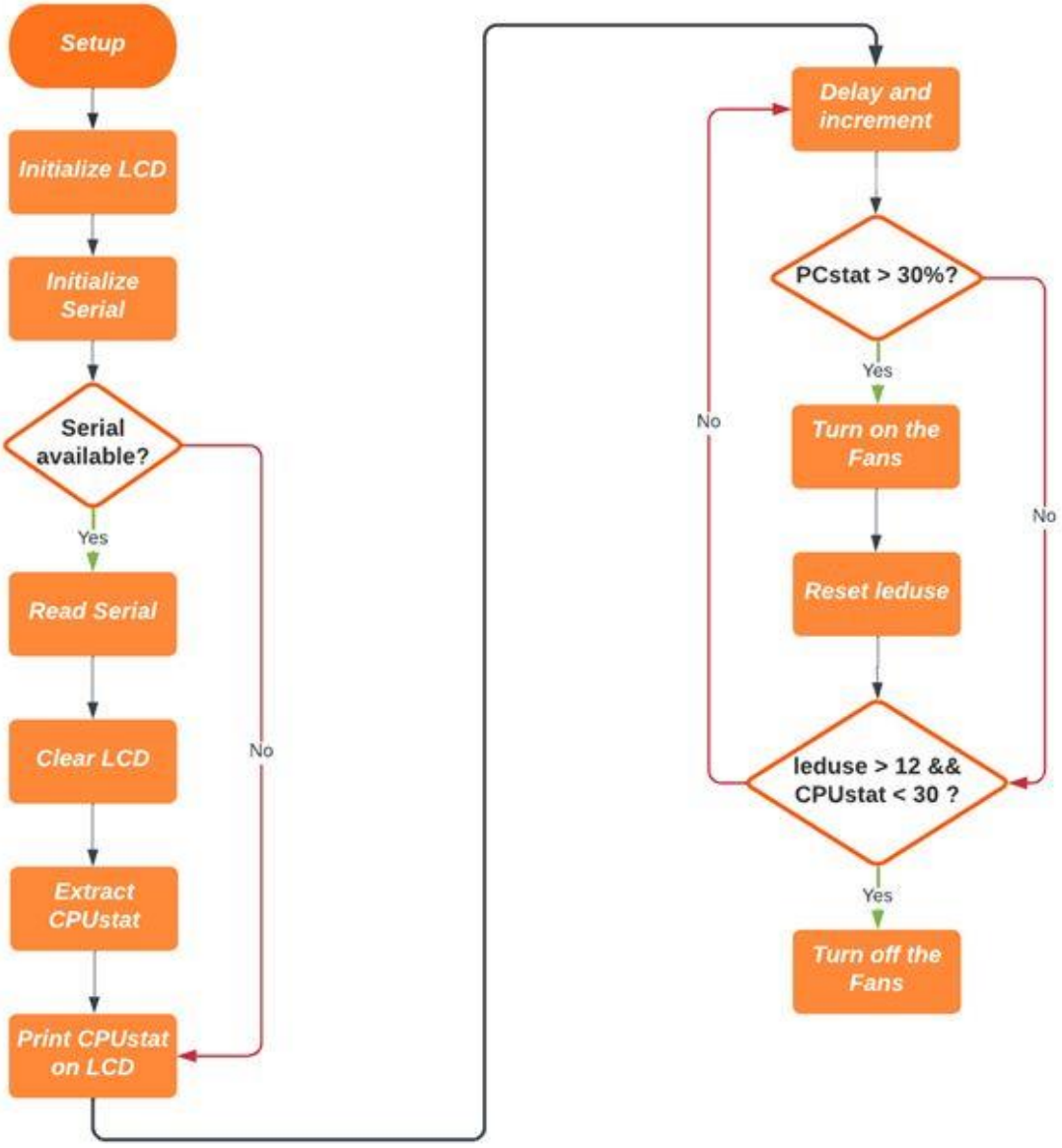
1. Python
2. C

Methodology:

In our project, we employed Python as the programming language for developing the driver. Within the Python program, we utilized the Psutil library to retrieve comprehensive information about CPU usage. Furthermore, the Serial library was instrumental in facilitating the transmission of CPU usage data to the Serial port. This data is subsequently transmitted to an Arduino device for further analysis and subsequent processing.


Start
Import Psutil and serial library
Initialize serial port
Infinite loop
Read CPU Usage
Send CPU usage through serial port
Print CPU usage

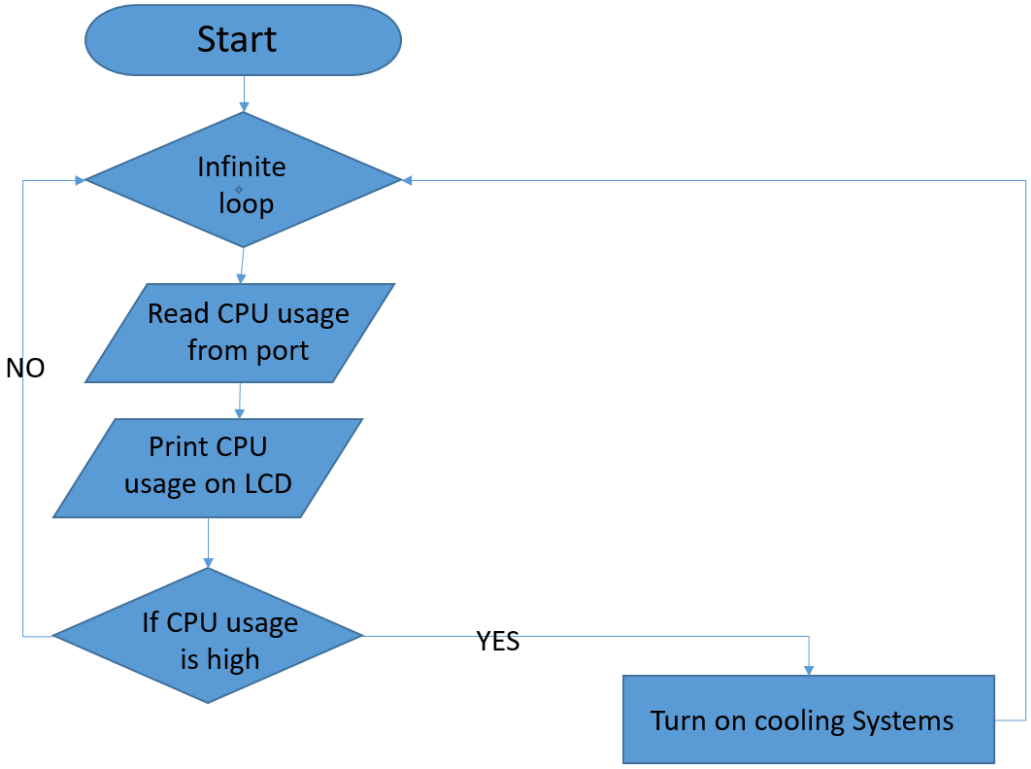
Arduino Working:



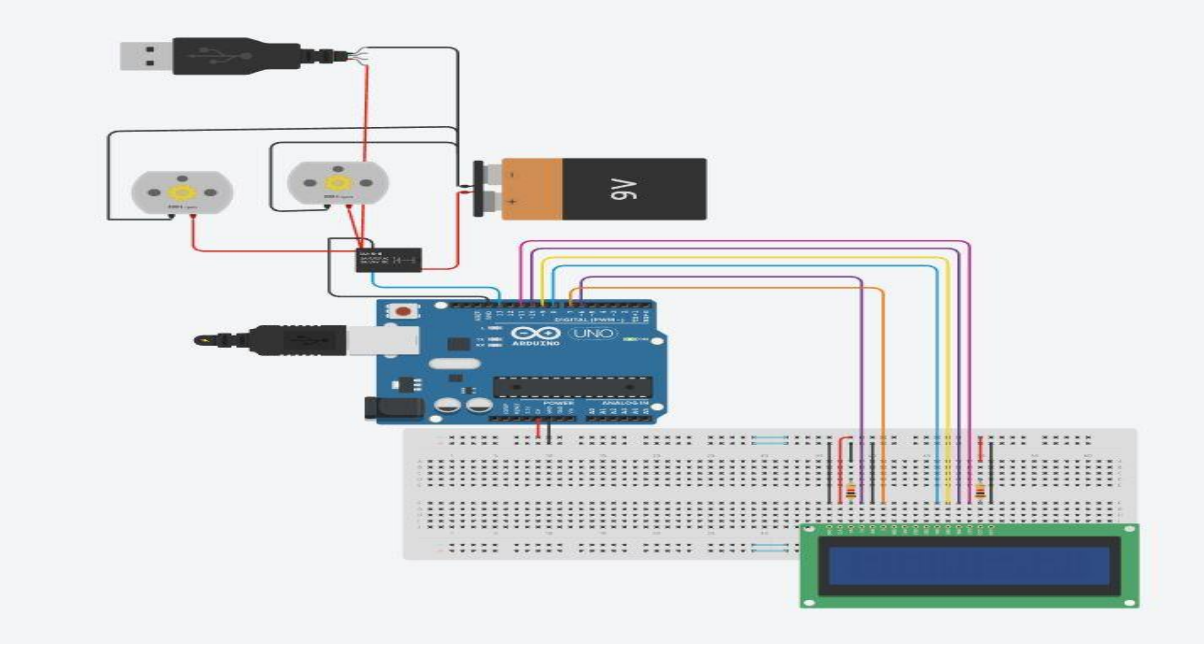
Functionality of the Hardware Setup: The core functioning of our hardware setup involves the transmission of CPU usage data to an Arduino microcontroller. Upon reception, the Arduino processes this data and displays it on an LCD screen. This real-time data display serves as a basis for comparison against a predefined threshold value. If the measured CPU usage surpasses this specified threshold, a cooling system is promptly activated. This cooling mechanism remains operational until the CPU usage falls below the designated threshold, ensuring efficient temperature management.

In this arrangement, the Arduino acts as a pivotal control unit, constantly monitoring and responding to the CPU usage levels. Its integration with the LCD screen facilitates immediate visualization, enabling users to observe and assess the current CPU performance. Should the CPU usage exceed the predefined threshold, the Arduino's decision-making process triggers

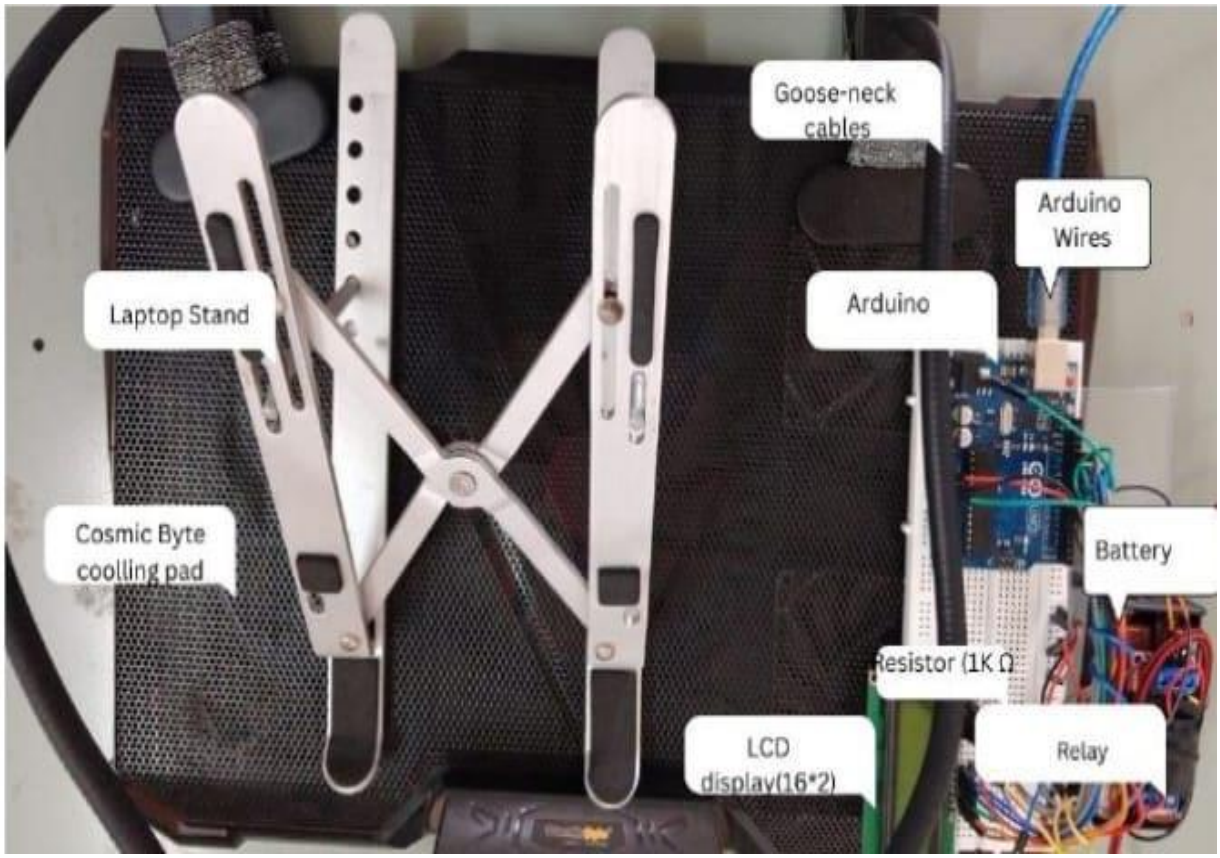
the engagement of the cooling system, preventing potential overheating issues and maintaining optimal performance. This closed-loop system guarantees the effective regulation of CPU temperature, contributing to enhanced hardware longevity and stability.



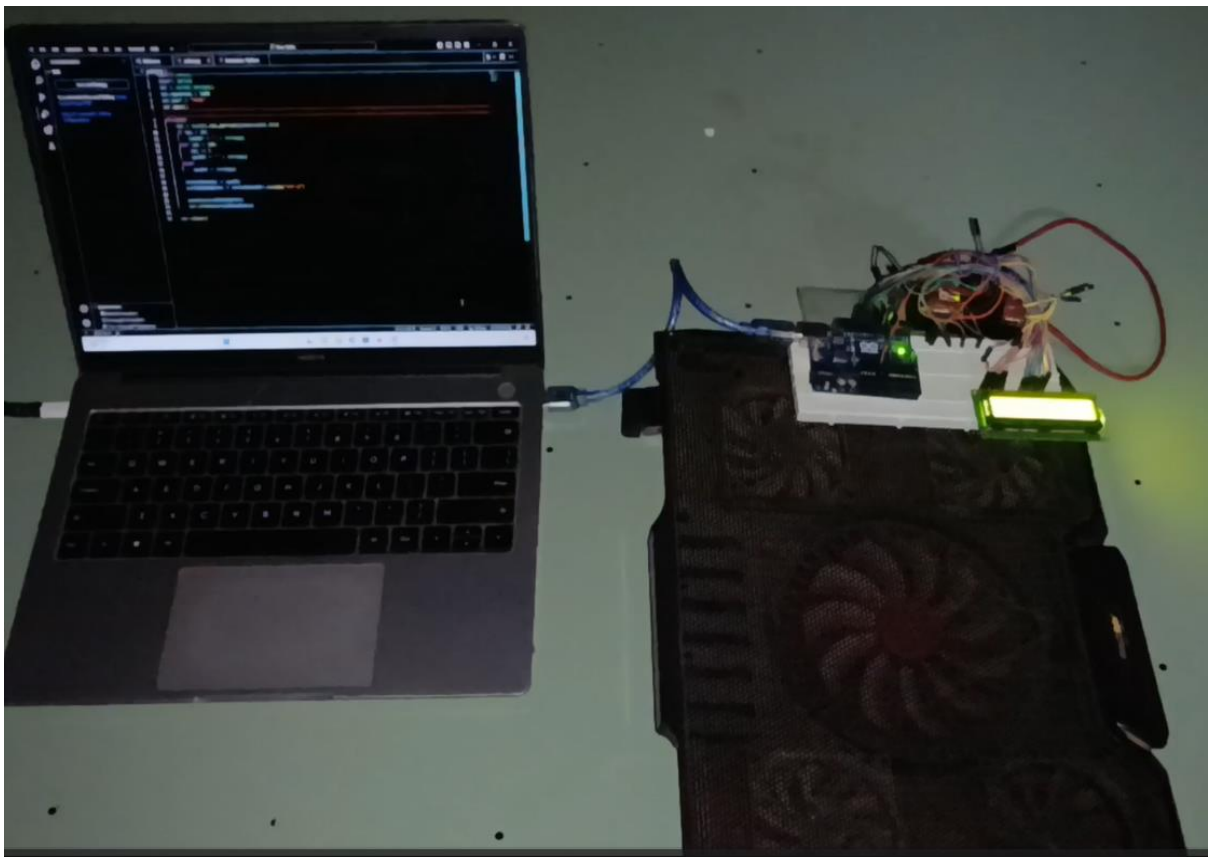
Labelled Circuit diagram:

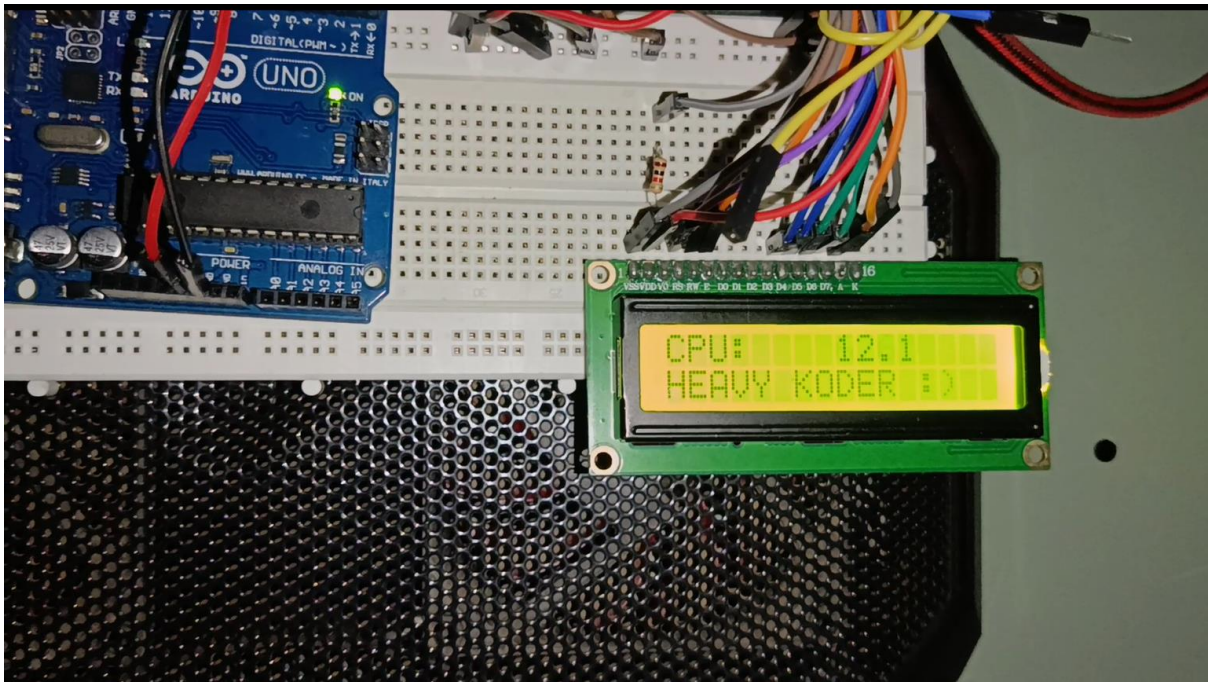


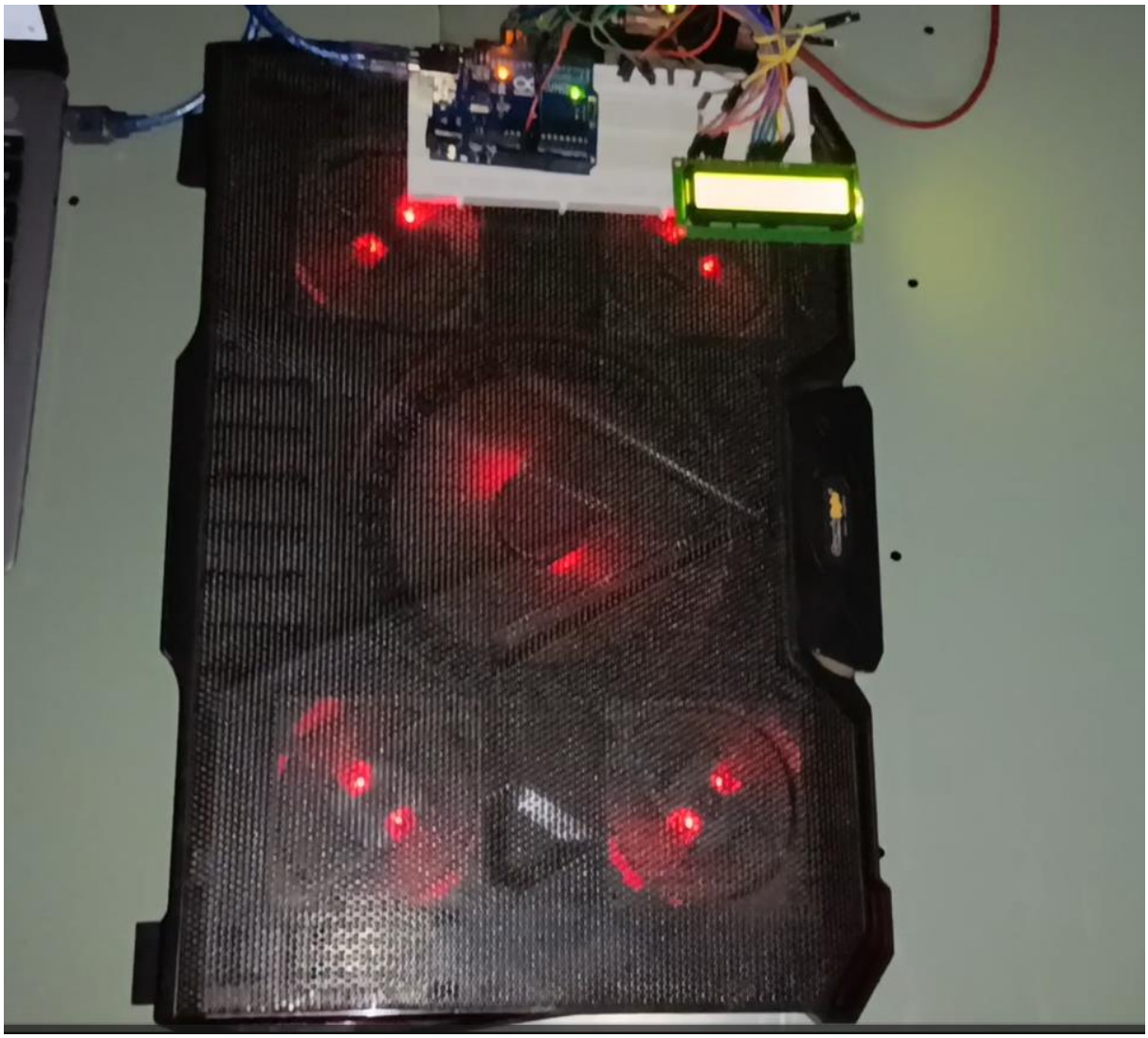
Labelled hardware diagram:

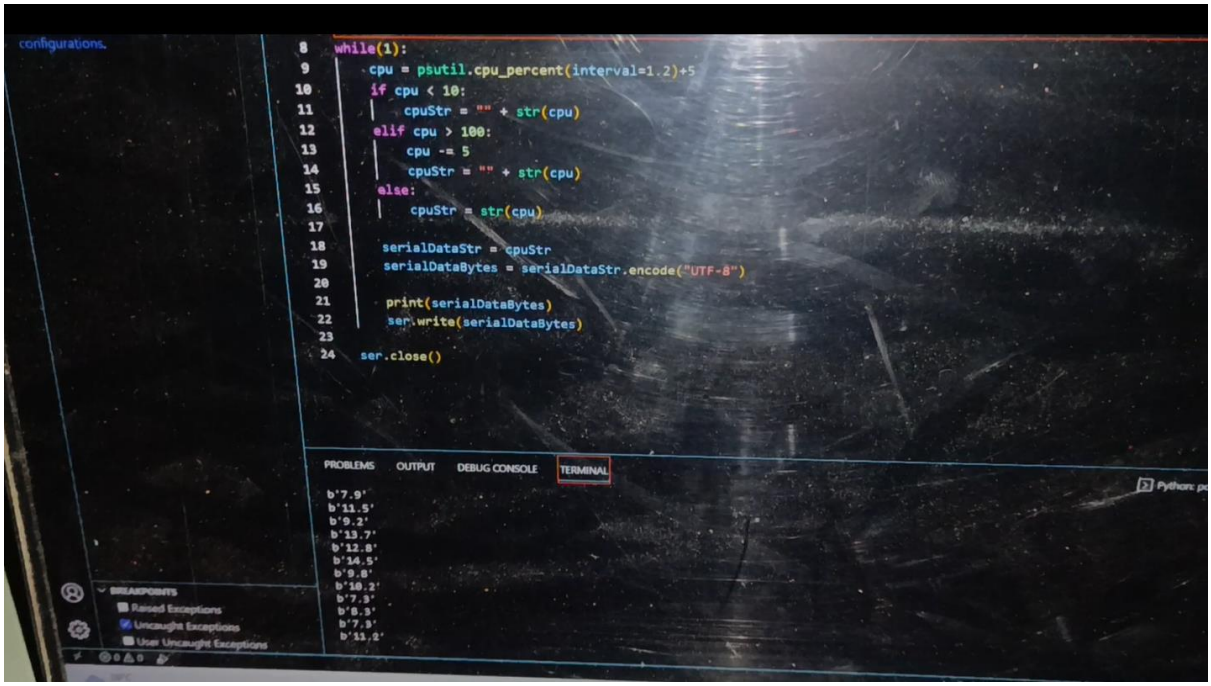


Pics of the hardware working:




UNO
CPU: 12.1
HEAVY KODER :)




configurations.
while(1):
cpu = psutil.cpu_percent(interval=1.2)+5
if cpu < 10:
cpuStr = "" + str(cpu)
elif cpu > 100:
cpu -= 5
cpuStr = "" + str(cpu)
else:
cpuStr = str(cpu)
serialDataStr = cpuStr
serialDataBytes = serialDataStr.encode("UTF-8")
print(serialDataBytes)
ser.write(serialDataBytes)
ser.close()
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
b'7.9'
b'11.5'
b'9.2'
b'13.7'
b'12.8'
b'14.5'
b'9.8'
b'10.2'
b'7.5'
b'8.3'
b'7.3'
b'11.2'
BREAKPOINTS
Raised Exceptions
Uncaught Exceptions
User Uncaught Exceptions