King Fahd University of Petroleum and Minerals

Computer Engineering Department

Homework Assignment #1

Due date: Monday February 24, 2025 @ 11:59 PM (before midnight)

Instructions:

1. Your solution should be uploaded through Blackboard as a single compressed zip file named COE550_HW1_FNAME_LNAME_ID.zip where FNAME and LNAME are your first and last names, respectively, and ID is your student ID. Name the individual parts as HW1-1-cloud.py, HW1-1-edge.pkt. Also, Include a screenshot of the output dashboard in Problem 1.
2. Use meaningful names for variables and identifiers. Add comment to your code. Good programming practices will be accounted for when grading.
3. Students are encouraged to discuss the homework in groups, but each student must write his/her own answers in his/her own words.
4. If cheating or copying is detected, both parties will get zero in the assignment.
5. No late submissions will be accepted.

Objectives: After finishing this assignment, you should be able to:

1. Write a fully working client and server programs using Python Sockets.
2. Write a simulator of an application with Cloud and Edge layers.
3. Use simple dashboard to visualize data.

Preparations:

1. Lecture slides on Python 3 Tutorial.
2. Basics of Python 3 socket programming. (https://www.tutorialspoint.com/python3/python_networking.htm).
3. Read about matplotlib, a rich python package for plots and charts. (https://matplotlib.org)
4. Study the lecture/video on Cisco Packet Tracer and socket programming.

Homework Problems:

Problem 1: (100 points): A Smart Factory Ventilation System

A smart factory ventilation system includes two MCUs. MCU1 reads the *CO* and the *ambient temperature* levels and reports them to a cloud application. On the other hand, MCU2 receives commands form the cloud application to open/close a *window* and a *fan*.

You are required to build a simulation of the above application consisting of a *cloud* and an *edge layers*. The cloud layer consists of a single process which controls the system and displays a dashboard showing a chart of the reported readings and the status of the window and fan. The edge layer contains the two MCUs. Figure 1 shows an architectural view of the system.

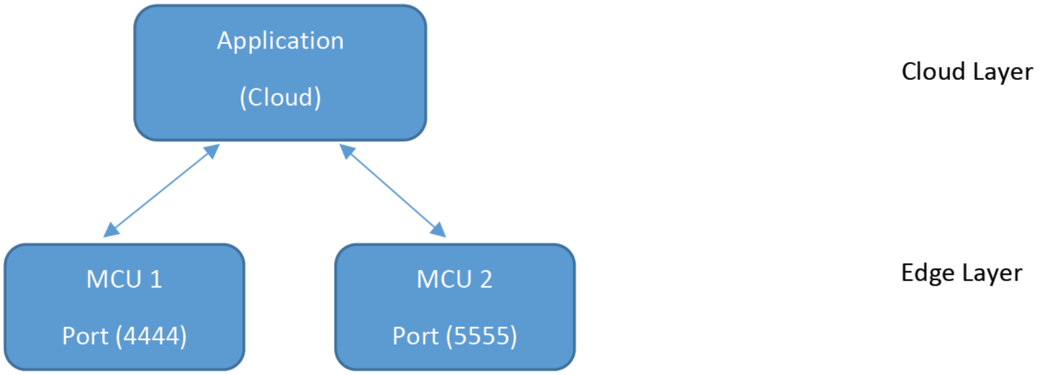

Figure 1: Architectural view of a Smart Factory Ventilation System.

The cloud application periodically polls the sensor nodes to retrieve the sensor data and plots the data using an *animated* graph. If the sensor reading exceeds certain thresholds, **a command is sent by the cloud** to turn on/off the devices. You need to implement the MCUs inside PT and the cloud service outside PT (using Python Script).

Use the TCP/UDP port number as shown in the Figure above. Your system must implement the following requirements:

1. The cloud service polls each sensor node once every 100 *ms* to request the sensors' readings.
2. The cloud service must display a live *line chart* showing the readings of each sensor and the status of the devices.
3. The reading should be continuous, and plots should keep updating continuously.
4. You need to define a simple protocol (messages) to request turning on/off the warning lights.
5. You need to define the threshold values for the sensors.

Connect your sensors to an SBC or MCU to enable connectivity. For programming in PT, use the RealTCPServer(), RealUDPServer() , RealTCPClient() and/or RealUDPClient() template.

Report: attach a report with your solution that includes:

1. Screenshot of the program output.
2. Types of sensors used and threshold you define for each sensor.
3. Any problems or limitations in your solution.
4. Protocol messages you specify between the cloud and sensor nodes.

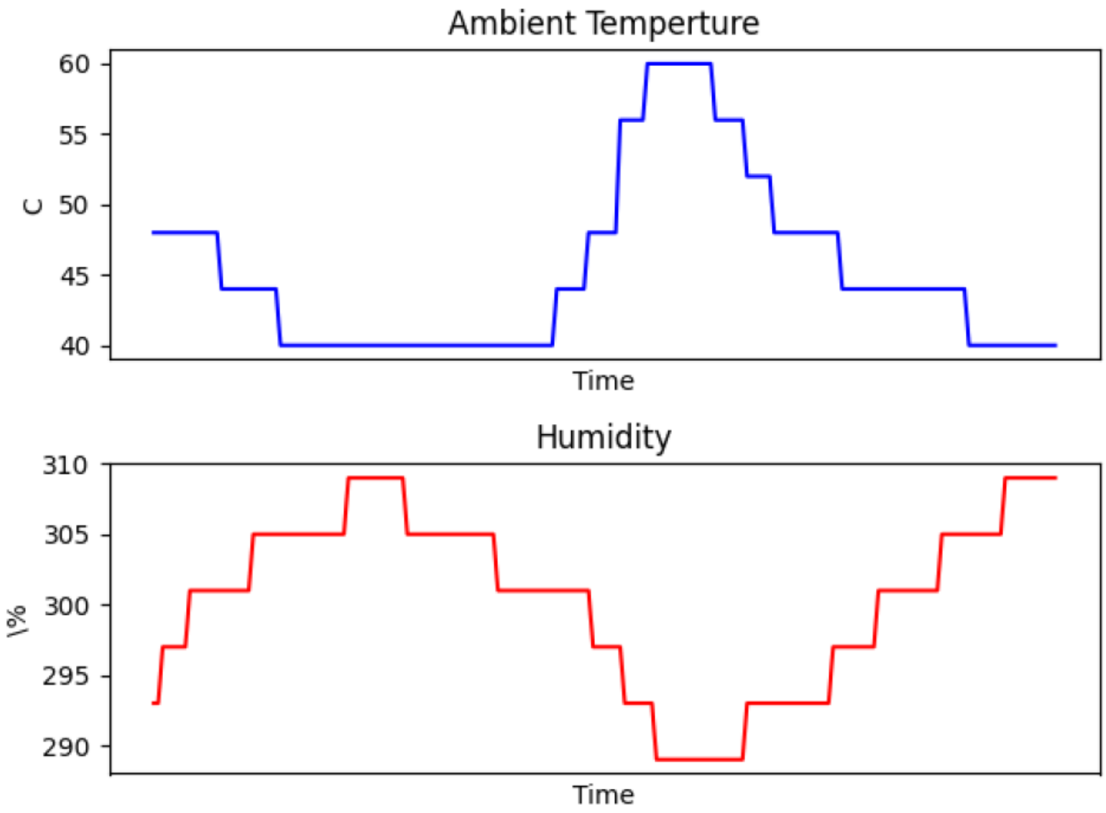

Figure 2: Example Output for Problem 1.