**CME466 Winter 2024** **Deliverable # 1: Design an IoT edge node** **Total marks: 10**

NOTE: For more information on the deliverables, please follow the lecture materials and in-class discussions. If you have further questions, please consult with the instructor(s).

In the module, you will design an edge or device or ground node of an internet of things (IoT) system. An edge node in an IoT system "provides the intelligence to sense, measure, interpret, and connect to an internet gateway to the cloud. The data can be preprocessed with some form of analytics before it is transmitted for deeper data mining intelligence." [1]

Edge computing was developed due to the exponential growth of IoT devices, which connect to the internet for either receiving information from the cloud or delivering data back to the cloud. And many IoT devices generate enormous amounts of data during the course of their operations [2]. Therefore, it is important to consider power consumption and performance tradeoffs while choosing the right device/platform in an IoT system.

In your smart edge node, you will use a Raspberry pi (RPi) as the central processing unit as well as the gateway. Python3 will be used to program the RPi.

**Part 1: Basic edge node with RPi, LED and switch**

1. Get a RPi board and make yourself familiar with it. Learn about the GPIO header pins. Follow the lecture materials to learn how to configure a RPi.

2. Now, download the required package into the RPi to work with its GPIO pins. Connect one LED and one switch with two suitable GPIO pins. Write a program in Python to turn on the LED.

3. Next, modify the program to include the switch so that the LED turns on when the switch is pressed.

   At this point, you are familiar with the basic configuration and workings of an RPi board. Let us move to the next part to connect sensor modules.

**Part 2: Smart edge node with RPi and sensor modules**

4. In this part, you are free to choose any sensor module to integrate with the RPi. However, note that the edge node will become a part of a smart home surveillance system in later deliverables. Therefore, you should choose sensors that may be useful in designing such a system. In addition to that, you must choose a sensor module that uses I2C, UART or SPI bus interface connection protocol. Do not use the sensor module(s) shown in the tutorial.

   Note that, the RPi does not have an integrated hardware analog to digital converter (ADC). Therefore, you cannot connect an analog sensor directly to it (like the way you connected analog sensors to a microcontroller in CME331). If you must use an analog sensor, there are external ADC modules (in the SunFounder pack) that you have to use with the RPi.

5. Once you chose one sensor module, download the RPi package for the particular sensor/module. You will also need to download the package or library to enable I2C/UART/SPI module on the RPi. The package and source will vary depending on the type of sensor you choose. Follow the instructions given in the class.

6. Once setup, write your code to read the sensor data and display it on the Python terminal continuously in a format that is understandable (e.g., Temperature is 25c).

NOTE: For more information on the deliverables, please follow the lecture materials and in-class discussions. If you have further questions, please consult with the instructor(s).

7. In this last section, we will integrate parts 1 and 2, i.e., RPi with the LED, switch, and the sensor module. You can choose any way of your liking. One example would to modify your code to include the switch, so that the sensor reads data and displays the reading once on the terminal, only when the switch is pressed. After displaying the value, you may flash the LED twice.

**Deliverables:**

- A report containing:
  - (a) your python code,
  - (b) block diagram of the circuitry (hand drawn or printed - both fine),
  - (c) a list of python packages you downloaded to perform the tasks
- Demo the system to your instructor. The instructor may ask questions to assess your design and request to see certain measurements made in real time.