CME466 (2022-23) **Deliverable # 3: IoT based smart parking system** **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, we will design a simple IoT based smart parking system. In deliverable 1, we designed an edge node. In deliverable 2, we established a full duplex communication between the edge node and the lab PC using MQTT protocol. Now, we will design a user interface (UI) that will be combined with the previous two designs, and put the entire framework into a practical application, like smart parking.

The parking system is shown in Fig 1.

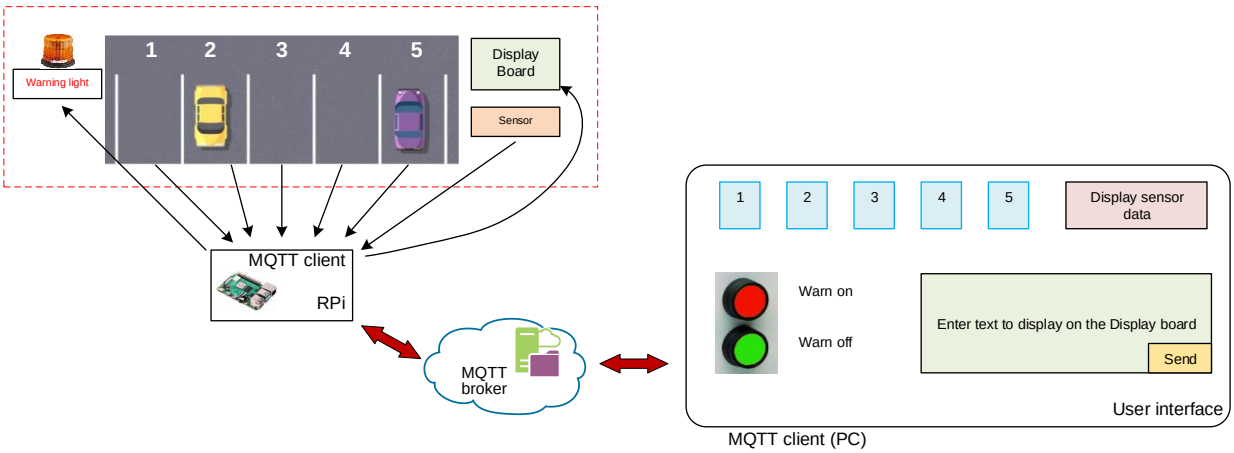

Fig. 1: An IoT based smart parking system

The components inside the red dotted box are meant to be ground sensors and devices. There are five parking spots available that are marked by numbers. Assume that these spots are equipped with sensors that can detect the presence of a vehicle. If the spot is empty, a "0" is generated. If the spot is occupied, a "1" is generated. Your system will notify the parking attendant (PA) which spot is empty and which one is occupied.

In addition, the parking lot is equipped with:

1. a warning light: PA will activate it when there is an emergency. When activated, the light will flash continuously.
2. a sensor (use any of your choice): reads some data, e.g., temp, motion, air, etc.
3. a display board: displays any message to the public/visitor as sent by the PA

All these sensors/devices are connected to a Raspberry Pi (RPi) that forms our edge node. The RPi collects the real time data and sends it to the PA over the internet using MQTT protocol. The PA sees the information and controls the sensors/devices in the parking lot using a user interface (shown in Fig 1).

The user interface shown in Fig 1 is for illustration purpose only. Use your imagination and style while designing your own UI. You may add additional features if you wish. However, the design requirements are given below. Therefore, the UI must have at least the following four components:

a) Parking spot indicator: It could be five lights (when a specific spot is empty, it should show green, otherwise red), or five checkboxes (when a specific spot is empty, it should be checked,

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).

otherwise unchecked), or some other way to indicate which spot is empty or which one is occupied.

b) Two buttons to control the warning light: When the "ON" button is pressed, the warning light on the ground must flash; when the "OFF" button is pressed, it is turned off.

c) A test box or some type of display box to show the outside sensor data (as you receive it continuously from the sensor)

d) A text box or window where the PA can put a message to be displayed on the display board. When the "Send" button is pressed, the message will be displayed on the board (for demo purpose, just print the message on RPi's terminal)

While you are designing your system, you may simulate the state of the parking slots (i.e., empty or occupied) using a switch or simply connecting the corresponding GPIO pin to a 3.3V/Ground pin of the RPi. You may use the RPi terminal to display any message (in place of the Display Board).

Deliverables:

- A report containing:
  - your python codes,
  - 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.