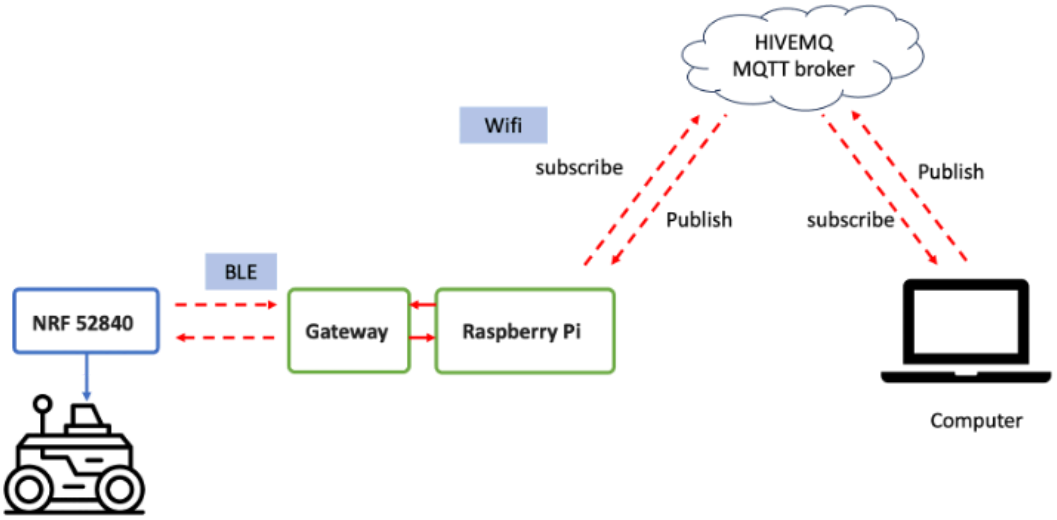
Industrial Internet Infrastructure (III), Group Assignment, 2024

1. Introduction

You will be in a group of 4 to 5 students, designing an end-to-end IoT system from the sensor/actuator to the cloud. The system should be well designed, secure, reliable and performant according to design principles that you have learned about in the III lectures and prior classes.

2. Requirements

The system consists of a sensor/actuator, two gateways and a server. The sensor/actuator sends sensing information to the server, while the server sends commands to the sensor/actuator. A user can use the server to control the sensor/actuator and also see all the sensed information via a web front end offered by the server. We provide a wheeled robot as the sensor/actuator, an nRF52840[1] as the Bluetooth Low Energy (BLE) gateway and a raspberry pi zero as the Internet gateway.



2.1. Sensor/Actuator

The **sensor/actuator** is a wheeled robot. An nRF52840 is the controller of the robot. The APIs to move the wheels are provided by us. You need to write a program on the robot which contains the following three major functional blocks:

a. Securely establish a BLE connection with the **BLE gateway**

[1] https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.0.pdf

b. Receive commands from the **BLE gateway**, and call APIs to move the robot according to the commands
c. Read the voltage of the super capacitor and the RPM of the four motors every 200ms, and then send the average voltage and RPM of the last 5 readings (1s) to the **BLE gateway** in a single message.

You are free to define the format of messages between the robot and the BLE gateway yourself. Each message should contain your group number, relevant security data (if any) and the sensed data described above.

2.2. BLE Gateway

The **BLE gateway** is a nRF52840 DK board. You should connect the gateway to the **sensor/actuator** by BLE, and to the **Internet gateway** (raspberry pi zero) by UART. You should write a synchronized multi-thread program to:

a. Establish a secured BLE connection with the **sensor/actuator**
b. Establish a UART connection with the **Internet Gateway**
c. Receive messages from the **sensor/actuator** and send them to the **Internet Gateway**
d. Receive messages from the **Internet gateway** and send them to the **sensor/actuator**

2.3. Internet Gateway

The **Internet gateway** is a Raspberry pi zero in this system. The gateway is connected to the **BLE Gateway** (nRF52840) by UART, and connected to the Internet by WiFi. You need to write a synchronized multi-thread program on the raspberry pi zero to:

a. Run an MQTT client to publish/subscribe messages
b. Publish the voltage and the RPM information to the MQTT broker
c. Subscribe and receive the moving command MQTT messages and send the commands to the robot

The MQTT topics should be “III2024/your_group_number/sense” and “III2024/your_group_number/control”.

2.4. Server

The **server** runs on a laptop provided by your group. The server runs:

a. A user-friendly tool to control the robot
b. A correctly configured instance of the TICK stack, wherein:
    i. Telegraf is used to collect sensing data from the robot over MQTT.
    ii. InfluxDB is used to store that data in an appropriate format.
    iii. Chronograph or Grafana is used to effectively visualize the data.

iv. Kapacitor is used to generate alerts on appropriate channels whenever the charge available on the super-capacitor drops by an additional 10% (i.e. 90%, 80%, 70%, etc.).

You should apply security methods to ensure end-to-end security across your system. You can either set up a MQTT broker (e.g. Mosquitto) on your computer or use a public MQTT broker on the internet. The super capacitor voltage range is 0.9V-2.8V, so we consider 2.8V as 100% and 0.9V as 0%.

3. Required Deliverables

- By 23:59 on Friday May 12th (submission procedure to be announced):
  - A report of up to 3 pages that describes the design of your system.
  - A well-organized zip file containing all of your source code.
- By appointment in the week following submission (scheduling to follow):
  - A presentation of 5 minutes introducing your solution.
  - A demonstration and walk-through of your system in operation.

For questions regarding the project, you can send an email to any of the instructors:

- yinze.li@kuleuven.be (**coordinator** and cloud/middleware help)
- bingwu.fang@kuleuven.be (cloud/middleware help)
- brendan.mackenzie@kuleuven.be (embedded help)
- lowie.deferme@kuleuven.be (embedded help)