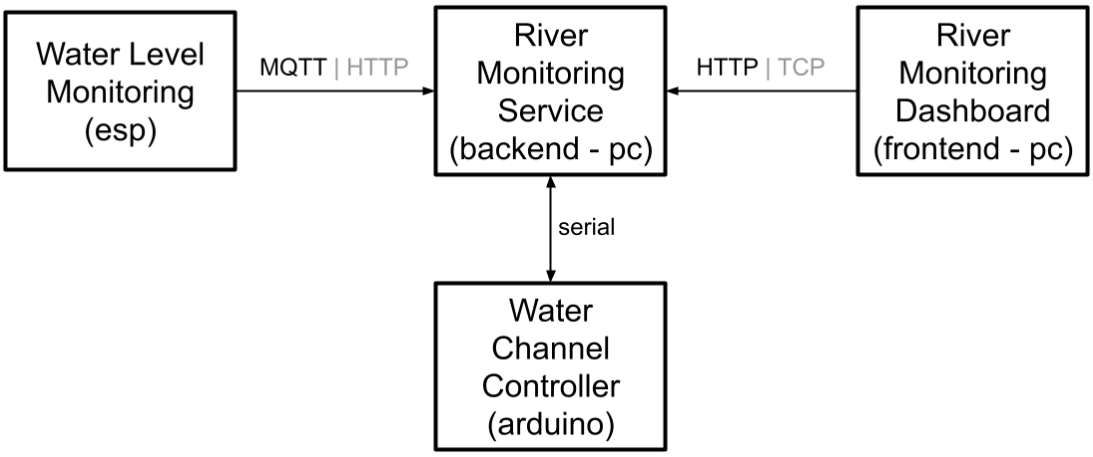
Embedded Systems and IoT - a.y. 2023-2024

Assignment #03 - *Smart River Monitoring*

v1.1-20240127

We want to realise an IoT system implementing a simplified version of a *smart river monitoring system,* as a system monitoring the water level of rivers and controlling related water channels. The system is composed of 4 subsystems:



- **Water Level Monitoring subsystem (based on ESP)**
  - embedded system to monitor the water level of a river
  - It interacts with the River Monitoring Service subsystem (preferably via MQTT[1] - HTTP can be used as a second option)
- **River Monitoring Service subsystem (backend - running on a PC server)**
  - service functioning as the main unit governing the management of the smart river system
  - it interacts through the serial line with the Controller
  - it interacts via MQTT[1] with the Water Level Monitoring
  - it interacts via HTTP[2] with the Dashboard
- **Water Channel Controller subsystem (Arduino)**
  - embedded system controlling the gate/valve of a water channel
  - it interacts via serial line with the River Monitoring System
- **River Monitoring Dashboard subsystem (Frontend/web app on the PC)**
  - front-end to visualise and track the state of the river monitoring
  - it interacts with the River Monitoring Service

[1] HTTP can be used instead of MQTT as a secondary choice, if have problems using MQTT
[2] TCP can be used instead of HTTP as a secondary choice, if you have problems using HTTP

Hardware components

- Water Level Monitoring subsystem
    - SoC ESP32 board (or ESP8266) including
        - 1 green led
        - 1 red led
        - 1 sonar
- Water Channel Controller subsystem
    - Microcontroller Arduino UNO board including:
        - 1 servo motor
        - 1 button
        - 1 potentiometer
        - 1 LCD display

General Behaviour of the system

The Smart River Monitoring system is meant to monitor the water level of a river and - depending on the level - controlling a valve to distribute the water to some channels. Details:

- About the Water Level Monitoring subsystem
    - The water level monitoring subsystem is responsible for continuously monitoring the level of the water, by means of the sonar.
    - The water level is sampled and sent to the River Monitor Service with some frequency F
        - This frequency depends on the state of the system and is established by the River Monitoring Service (see later).
    - When the system is working correctly (network ok, sending data ok) the green led is on and the red is off; otherwise – in the case of network problems – the red led should be on and the green led off.
- About the Water Channel Controller subsystem
    - The Water Channel Controller subsystem is responsible for controlling the valve establishing how much water should flow to the channels, and this is determined by the valve opening level – from 0% = full closed (no water flows from river to channels), up to 100% = full open
        - The valve is implemented by the servo motor – angle 0° corresponds to valve opening level 0%, angle 180° corresponds to valve opening level 100%
    - The valve opening level depends on the state of the system, established by the River Monitoring Service (see later)
    - The Water Channel Controller provides also a button to enable a manual control modality
        - When the button is pressed, the controller enters in manual mode, so that the valve opening level can be manually controlled by operators using a

potentiometer. To exit from the manual model, the button should be pressed again.
    - The Water Channel Controller subsystem is equipped also with an LCD display reporting the current valve opening level and current modality (AUTOMATIC or MANUAL)
- About the River Monitoring Service
    - This is the subsystem deciding the overall policy and behaviour of the river monitoring system, depending on the water level as measured by the Water Level Monitoring subsystem
    - Policy:
        - When the water level is in the range [WL1, WL2], then the system is considered in a NORMAL state. In the NORMAL state:
            - the frequency to be used for monitoring the water level is F1
            - the valve opening level should be 25%
        - When the water level is < WL1, the system is in an ALARM-TOO-LOW state.
            - In this state, the valve opening level should be 0%
        - When the water level is > WL2, there are three further cases
            - WL2 < water-level <= WL3 → the system is in a PRE-ALARM-TOO-HIGH state.
                - In this state, the frequency to be used for monitoring the water level should be increased to F2 (where F2 > F1)
            - WL3 < water-level <= WL4 → ALARM-TOO-HIGH state
                - In this state the frequency is still F2, but the valve opening level must be 50%
            - water-level > WL4 → ALARM-TOO-HIGH-CRITIC state
                - In this state, the frequency is still F2, but the valve opening level should be 100%
- About the River Monitoring Dashboard
    - The dashboard has two main responsibilities:
        - To visualise the state of the River Monitoring system. In particular:
            - The graph of water level trend, considering a certain temporal window (the last N minutes)
            - The state of the system
                - NORMAL, ALARM-TOO-LOW, PRE-ALARM-TOO-HIGH, ALARM-TOO-HIGH, ALARM-TOO-HIGH-CRITIC
            - The valve opening level
        - To allow a user for controlling manually, from remote, the valve opening level

---

The assignment

Design and develop a prototype of the Smart River Monitoring system, considering as further requirements:

- **Water Level Monitoring subsystem**
  - Run on ESP32 or ESP8266
  - Must use preferably MQTT to communicate with the River Monitoring Service
- **Water Channel Controller**
  - Run on Arduino
  - The control logic must be designed and implemented using finite state machines (synchronous or asynchronous)
  - Communicate with the River Monitoring Service via serial line
- **River Monitoring Service**
  - Run on a PC
    - No specific constraints about the programming/sw technology to be used
  - Use preferably MQTT to communicate with the Water Level Monitoring subsystem
- **River Monitoring Dashboard**
  - Run on a PC
  - No specific constraints on the technologies to be used
    - Can be implemented as a web app running in a browser interacting via HTTP with the service or a PC app based on sockets or HTTP

The Deliverable

The deliverable consists in a zipped folder **assignment-03.zip** including:

- 4 subfolders (one for each subsystem)
  - water-level-monitoring-subsystem
  - river-monitoring-service
  - water-channel-controller
  - river-monitoring-dashboard

- **doc** folder
  - including a brief report **(report.pdf)** describing the system, including also a description of FSMs, a representation of the schema/breadboard and the link to a short video demonstrating the system.