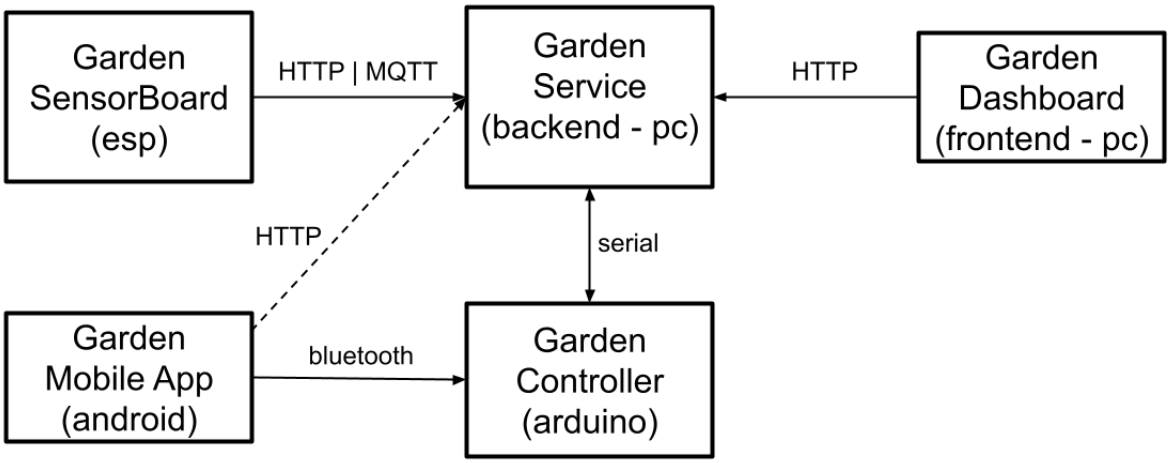
Embedded Systems and IoT - a.y. 2021-2022

Assignment #03 - *Smart Garden*

v1.0-20220531

We want to realise an IoT system implementing a simplified version of a smart garden, i.e. a smart system monitoring and controlling the state of a garden.

The system is composed of 5 subsystems:



- **Garden SensorBoard (esp)**
  - embedded system to monitor the state of the garden by using a set of sensors
  - It sends the data to the Garden Service (either via HTTP or MQTT)
- **Garden Service (backend - pc)**
  - service functioning as the main control unit, governing the management of the garden.
  - it interacts through the serial line with the Controller (arduino) and via HTTP or MQTT with the Garden SensorBoard (esp), and via HTTP with the Dashboard (frontend/PC)
- **Garden Controller (Arduino)**
  - embedded system controlling the irrigation system and lighting
  - it interacts via serial line with the Garden Service and via Bluetooth with the Mobile App, used by garden "operators"
- **Garden App (Android - smartphone)**
  - mobile app that makes it possible to manually control the irrigation system and lighting.
  - it interacts with the Garden Controller via Bluetooth

- **Garden Dashboard (Frontend/web app on the PC)**
    - front end to visualise and track the state of the garden
    - it interacts with the Garden Service to get the data

Hardware components

- Garden SensorBoard
    - SoC ESP32 board including
        - a green led
        - 1 temperature analog sensor
        - 1 photoresistor analog sensor
- Garden Controller
    - Microcontroller Arduino UNO board including:
        - 4 green leds, simulating a light subsystems
        - 1 servo motor simulating an actuator for the irrigation system
        - 1 Bluetooth module HC-06 o HC-05

General Behaviour of the system

The Smart Garden functions in three main modes:

- AUTO
- MANUAL
- ALARM

In AUTO mode, the system monitors the state of the garden by periodically reading the value of a set of garden parameters (temperature, luminosity). Depending on the value of the parameters and of the state of the garden, the service decides and executes a control strategy, controlling the irrigation system and the light system. The irrigation system control concerns opening, closing and setting the speed. The light system control concerns setting on/off for two lamps and setting the intensity for other two lamps.

In MANUAL mode, the control is taken by the human operator, through the mobile app. In MANUAL mode, the operator controls the irrigation (close/open, speed) and the lighting systems (2 lamps on/off, 2 lamps with intensity).

By default the system starts in AUTO mode and it can be switched to MANUAL mode by the human operator through the Garden App.

The system enters automatically in the ALARM mode when some conditions over the garden state are verified (e.g. temperature in some range) and the operator intervention is requested both in the dashboard and in the app. The ALARM mode can be deactivated only by the operator using the app.

Detailed Behaviour of the system

The data detected by the photoresistor must be mapped in a range of 8 values

- if the value < 5, the brightness system must be activated.
    - the first two "green" LEDs must light up;
    - while the brightness of the other two LEDs must be adjusted to an interval of 5 values (0-4).
- if the value < 2 the irrigation system must be activated (we presumably irrigate at night).

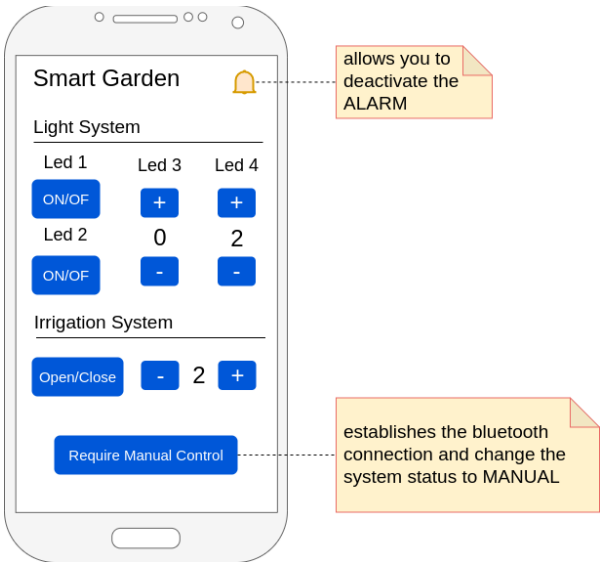
The value detected by the temperature sensor must also be mapped over a range of 5 values.

- the calculated value will determine the speed of the irrigation system;
- the irrigation system stops after Y seconds and stays asleep for X minutes (during this time it cannot operate).

If the temperature value is equal to 5 and the irrigation system is in pause, the system status must be changed to ALARM.

- the "green" LED of the ESP must turn off.

A possible layout of the application is presented below in order to summarise the main functions of the systems.



The control buttons can only be enabled after connecting, through the specific button, the application to the Arduino via Bluetooth (the connection is not a sufficient requirement, it must be ascertained that the system is really in MANUAL mode).

The assignment

Design and develop a prototype of the Smart Garden system, considering the following requirements

- **Garden SensorBoard - based on ESP32**
  - must use either the HTTP or MQTT to communicate with the Garden Service
- **Garden Controller - based on Arduino**
  - the control logic must be designed and implemented using finite state machines (synchronous or asynchronous)
  - must communicate with the Garden Service via serial line
- **Garden Service - in execution on a PC**
  - no specific constraints about the programming/sw technology to be used
  - must use either HTTP or MQTT to communicate with the Garden SensorBoard
- **Garden App - based on Android (either real device or emulated)**
  - for real device, the communication with the Garden Control must be base on the BT wireless technology
  - for emulated devices, the communication can be done using the serial line communicating with the Android Emulator through a software bridge, as presented in lab
- **Garden Dashboard - to be run on a PC**
  - no specific constraints on the technologies to be used
  - it can be implemented as a web app running in a browser or a PC app based on sockets

The Deliverable

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

- 5 subfolders (one for each subsystem)
  - garden-service
  - garden-sensorboard
  - garden-controller
  - garden-dashboard
  - garden-app

- **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.