Deployment of a Wireless Sensor Network for Odor Distribution Mapping

Juraj Korcek

Professor : Alcherio Martinoli
Assistant(s) : Faezeh Rahbar

The goal of this project was to design and implement software components for the stationary wireless sensor network that is part of a distributed system for Odor Distribution Mapping.

The role of the system is to measure acetone gas levels at various places in a wind tunnel and store it for later analysis.

The system is built to be robust and as close to real-time as possible in order to minimize probability of data loss.

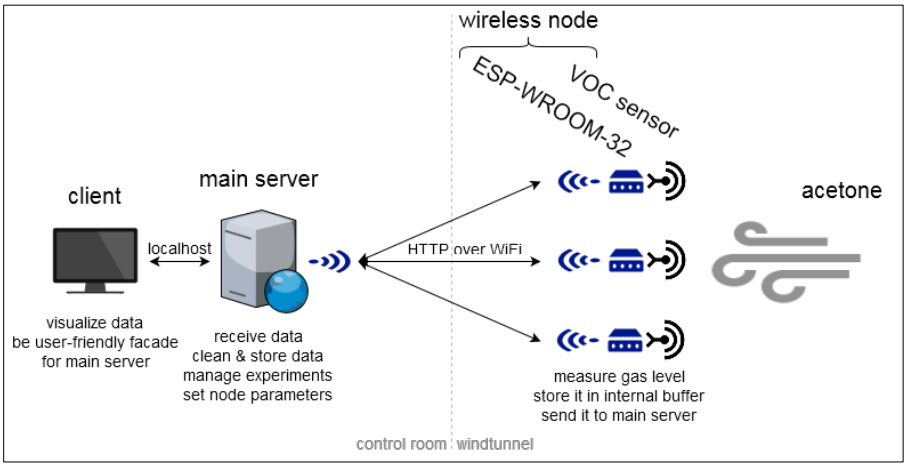
The system consists of several wireless sensor nodes, main server, client application and WiFi network. The wireless node comprises main board, sensor board and a VOC sensor.

At the core of battery-powered wireless node lies ESP-WROOM-32, a generic Wi-Fi+BT+BLE MCU module, which is programmed to serve as a middleman for the webserver and VOC sensor. It is responsible for reading values from the sensor, storing them in internal buffers and sending them to the main server. Arduino IDE (with Arduino core for ESP32 add-on) is used to program the module. The fact that the module has 2 cores allows me to run asynchronous HTTP webserver as well as client on the module, i.e. it allows to answer and send requests from/to main server while doing other computation on the background. Network Time Protocol is used to synchronize time among several nodes.

The remote main sever is used to set settings of as well as to receive data from the wireless nodes. Additionally, it manages experiments and persists the received data. It is programmed using python with Flask web framework. Redis, a key-value store, is the choice for the data store.

On the top of the main server sits a client that provides web user interface allowing to visualize data and control the system easily. HTML, CSS, JavaScript and jQuery, i.e. the common web development stack, is used to build the client. Bootstrap toolkit is used to make the application frontend responsive while flot library facilitates easy data visualization. The web application approach is chosen because, compared to local applications, it allows easier multiplatform and multiple-form-factor compatibility as well as easy remote access.

In order to minimize likelihood of data loss in case of technical problems, the system is robust to WiFi network crash and congestion, main server application crash or stalling, main server computer shutdown - both graceful and abrupt. Additionally, the system automatically recovers from wireless node restart. Recovery in this last case means that the wireless node reconnects and the currently running experiment continues. Naturally, in this case a few datapoints spanning a couple of seconds are lost between node restart and reconnect.



One of the goals has also been to write the code in clean, understandable, well-commented and modular fashion so that it can serve as a template or a reference for future projects of this kind in DISAL laboratory.

System components including tasks that they are responsible for