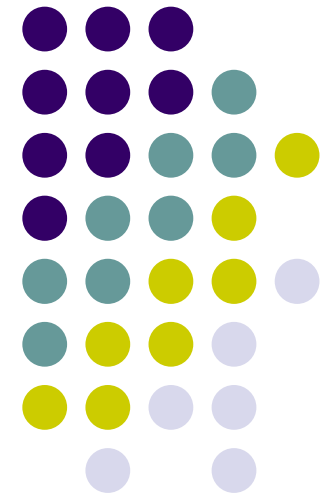


SCM

Sistemas de Comunicação Móvel

4th Assignment



© Paulo Simões – DEI/FCTUC

Delivery Dates



- December 4th
(submission, using InforEstudante)

SmartBuilding Controller

Exchanging configurations and sensor data



ACME wants to develop an advanced controller for its smart building platform, based on the following components:

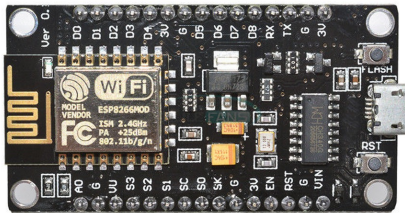
- an **application for controlling a building** (a PC/smartphone application built using any language and OS you prefer)
- a **multiroom AVAC unit** that heats/cool/ventilates the building rooms (represented by a PC/smartphone application)
- a **multiroom light controller** that turns on/off the lights of multiple rooms (represented by a PC/smartphone application)
- An **environmental sensor** (human presence, temperature, crepuscular, CO2) (hosted in an Arduino)
- a **debugging application** that shows all messages exchanged between the three system components.
- an **MQTT broker** supporting communications between all components (based on any opensource broker you choose).

Assignment description

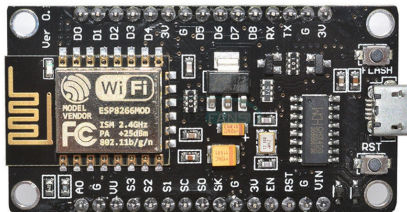
Reference Scenario



Environmental sensor
(Room A)



Environmental sensor
(Room B)



Debugging
Application

MQTT Broker



Management
Application



Multiroom
AVAC unit



Multiroom light
controller

Automatic Control



Please consider at least the following scenarios for each room (consider at least three rooms):

- Room temperature must be kept between MIN1 and MAX1 in case of human presence
- Room temperature must be kept between MIN2 and MAX2 in case of *no* human presence
- Room ventilation must be turned on when CO2 levels exceed MAX_CO2
- Room ventilation must be turned off when CO2 levels go below SAFE_CO2
- Lights must be turned off when no humans are present
- Lights must be turned on when the crepuscular sensor reports a value below LIGHT_THRESHOLD and there is human presence
- The management application may check the current “software version” of the sensors (an integer), and trigger “software updates” on specific sensors.
- When a specific sensor receives a “software update” command, its LED blinks for 2 seconds and “, and it will increase its “version number”.
- Sensor reports all temperature changes and all occupancy changes
 - Temperature, CO2, light and presence changes are emulated using the serial console
- AVAC unit reports activation/deactivation of heating or cooling
 - Moreover, AC unit also reports when it boots and when there is a shutdown
 - AVAC shows in its interface (window or text console) when it is heating, cooling and/or ventilating
 - able to work autonomously without the management application, using just the sensor’s data
- Light controller has no internal intelligence, being commanded by the management application alone

Management Application



- Management application, AVAC unit and multiroom light controller can be developed using any language, any OS and any terminal of your choice (e.g., computer, smartphone).
- This management application must allow to:
 - Monitor the conditions in each room (at least three rooms).
 - define for each room the parameters: MIN1, MAX1, MIN2, MAX2, SAFE_CO2, MAX_CO2, LIGHT_THRESHOLD
 - Monitor the status of the AVAC
 - Monitor the status of the light's controller
- Ideally, your application should provide a decent user interface

AVAC and lights controller



- Simple applications, with no local inputs (except at setup, if necessary)
- Local interface just provides local status of the unit, so we understand what it is doing.
- AVAC is able to work autonomously without the management application, using directly just the sensor's data and the parameters provided by the management application
- The lights controller is a dummy application. Every action is remotely triggered by the management application.

Broker and debugger



- For installing the broker use whatever publicly available broker you prefer. In case you don't succeed in the installation, use the publicly available DEI broker mentioned in the T class slides.
- The debugging application just needs to show in the screen all messages exchanged by the broker
- **Carefully define MQTT communications**
(naming of data, subscription strategies, special MQTT parameters, security features, support for multiple houses and multiple rooms, etc.)

Assignment description

Delivery format:



- PDF Report, including:
 - *Description of the developed system, explaining the general architecture, expected behavior, used libraries and opensource software, (optionally) snapshots of the management applications and the debugging application, etc.*
 - *Description of the solutions devised for MQTT communications, including naming of data, data persistence options, security features, etc.*
 - *Any further remarks and comments you may want to provide (e.g., implementation options, found issues, devised solutions, etc.)*
- Zipped source code
(please properly comment your source code so I can understand it)