

Smart Citofono

1. Concept and Requirements

The project consists in the realization of a smart interphone, which can be managed both using radio devices connected to the same network of the interphone node and via a NodeRed dashboard.

The idea is to connect an existing interphone to a radio device, which can forward calls received by the interphone on a local network to the motes, is able to open door or gate when a corresponding message is received.

2. Design

The general design deals with two different types of node: an interphone node (citofono) and possibly many mote nodes. They communicate using MQTT. An RPL border router is needed for the communication between nodes, and also an MQTT broker that can run on the same physical node of the router.

Empty Diagram

For this purpose, it has been chosen to implement these nodes using Contiki OS, which provides a framework to work with constrained resources on embedded devices using an event-based design. Contiki gives the possibility to simulate the entire network on Cooja running the real firmware, which is good for a proof-of-concept development.

The downside of Contiki is that not many embedded devices are supported. For this reason it is likely not a solution that can be easily adopted in the implementation phase.

Contiki also provides an MQTT engine, which has been used to build the MQTT layer of the software, shared by both the mote and the interphone software.

Finally, the application layer is different between nodes and interphone. Each of these two types of nodes define their own internal state machine, as well as event handler when an event is received. Also, each publishes in the same channel where the other subscribes.

2.1 MQTT Layer

MQTT engine provided by Contiki in the `apps/` folder.

FSM

Communication protocol

2.2 Interphone node

... FSM

2.3 Motes

... FSM

3. Simulation

The simulation is done using Cooja. For simulating purposes, the call is represented by a button and the actuators by LEDs.

In particular, the simulations have been carried out using the RPL-border-router provided by Contiki, 1 citofono and 2 motes. Tunslip has been used to communicate between the router serial port and the MQTT server (mosquitto). Finally,

- connection phase
- idle phase
- call received - nobody responds
- call received - someone responds

Disegni

4. Implementation

While Contiki has been useful for simulation purposes, for the implementation part some esp32 and a raspberry pi have been chosen for the motes and interphone node respectively. The implementation is still work in progress.

- security
- audio & video streaming
- raspi for interphone+border router+MQTT
- esp32 with vibration and button for motes