

# RelazioneElaborato3

Luca Rapolla

March 2023

## 1 Introduction

The prototype is meant to emulate a smart room as a smart system monitoring and controlling a room environment. The system is divided in 5 subsystems:

- Room Sensor Board (Based on Esp8266 board)
- Room Service (Backend Server based on Python's *Flask* library)
- Room Dashboard (Web Frontend)
- Room Controller (Based on Arduino board)
- Room Mobile (Based on Android)

The system detects two main characteristics of the room: if there are any people inside the room and if there's enough light. This detection is powered by the **Room Sensor Board** based on **ESP-8266** chip. The chip mounts two main sensors:

- Photoresistor.
- Pir sensor.

The *Room Sensor Board* communicates to the server via **MQTT** protocol which takes the decisions to control the room environment. It is possible to keep track of the room status in the *Room Dashboard* which request the data from the server via **HTML** protocol. Whenever the server states that a decision must be taken, he act by communicating to the *Room Controller* via serial line what needs to be changed (e.g. switch on/off the lights, opening/closing the roller blinds). It is possible to control the *Room Controller* via the android app too.

## 2 Room Sensor Board

I've took the decision to swap the PIR Sensor with a Button because my PIR module was not working properly. By the way, in the report i will continue to refer to the PIR sensor.

The *Room Sensor Board* module is based on *ESP-8266* board. The component mounts three parts:

- One PIR sensor (in this case switched with a Button).
- One Photoresistor.
- One Led.

The board detects periodically the room's situation and, if there are people inside the room it switches ON the led, otherwise it keeps it OFF. Then the board will send the data to the server via **MQTT** protocol. The sent data is formatted as a *JSON* file in this way:

```
{  
  Light: Light Value  
  Pir: 0 or 1 depending if it detects someone  
}
```

The *Room Sensor Board* project is available at: <https://github.com/Cippopo/IoTAssignment3/tree/main/Room-Sensor-Board>

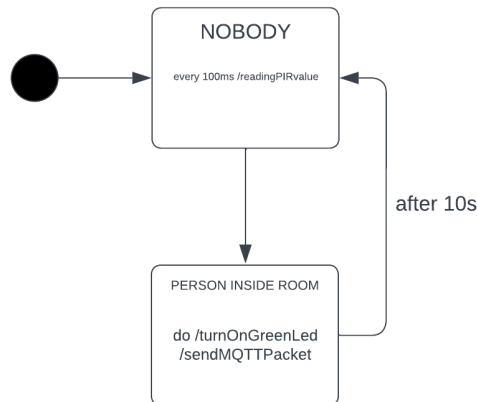


Figure 1: Room-Sensor-Board State Machine

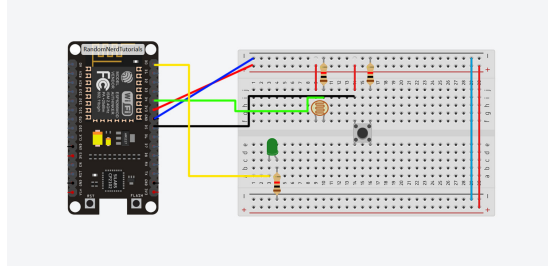


Figure 2: Room-Sensor-Board with Button Schematic

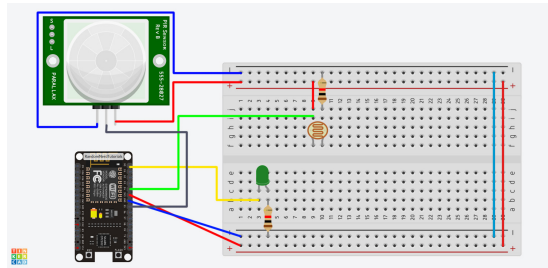


Figure 3: Room-Sensor-Board with PIR Schematic

### 3 Room Service

The *Room Service* is the brain of the entire system. It accepts data from the *Room Sensor Board* and controls the *Room Controller*. It uses 3 principal technologies:

- Flask library for the HTTP server.
- PAHO.MQTT library for the MQTT communication.
- Serial library for the serial communication.

The server implements three main threads, one for each component, in order to not waste any data. The server uses JSON format in order to communicate with the *Arduino* board through the serial line. The JSON data is formatted in the following way:

```
{
  Light: 1 if it needs to be turned on, 0 otherwise
  Servo: The value of the servo angle
}
```

The time is managed by checking every command if it's in the right time. (e.g. The server will not ask to the *Arduino* board to raise the roller blinds after 7 o' clock). The MQTT broker is subscribed to the same broker as the *ESP* module, in that way if the *ESP* sends a message, the server receive it.

The *Room Service* project is available at: <https://github.com/Cippopo/IoTAssignment3/tree/main/Room-Service>

## 4 Room Controller

The *Room Controller* module is based on an *Arduino* board. The board is connected to two main parts:

- One Servo motor, which emulates the Roller Blinds.
- One Led, which stands for the room's light.

The *Room Controller* communicates with the *Room Service* via Serial Line. The board checks periodically if there any new messages coming from the Serial Line or from the Android App. When the *Room Controller* reads a message, it perform the operation that need to be performed.

The *Room Controller* project is available at: <https://github.com/Cippopo/IoTAssignment3/tree/main/Room-Controller>

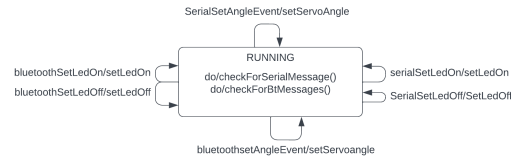


Figure 4: Room-Controller State Machine

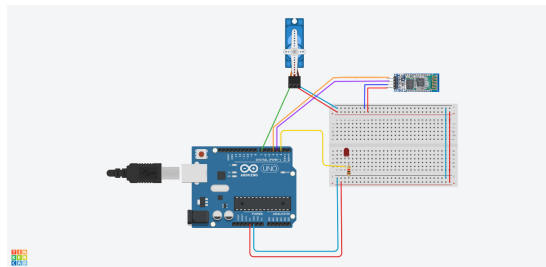


Figure 5: Room-Controller Wirings

## 5 Room Mobile App

The *Room-Mobile-App* is a simple Android Application which first makes you to connect with the Bluetooth transmitter mounted on the Arduino board, then gives you the possibility to Switch ON and OFF the lights in the room. The app gives to possibility to control the rolling curtains too using a slider. The *Room Mobile App* project is available at: <https://github.com/Cipppo/IoTAssignment3/tree/main/Room-Mobile-App>

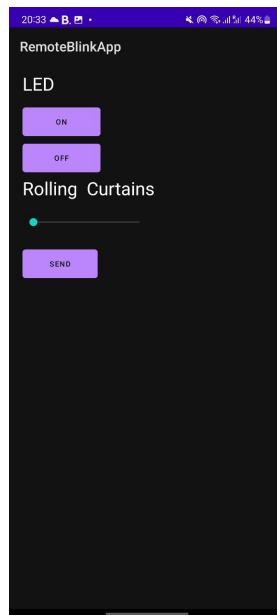


Figure 6: Room-Mobile-App