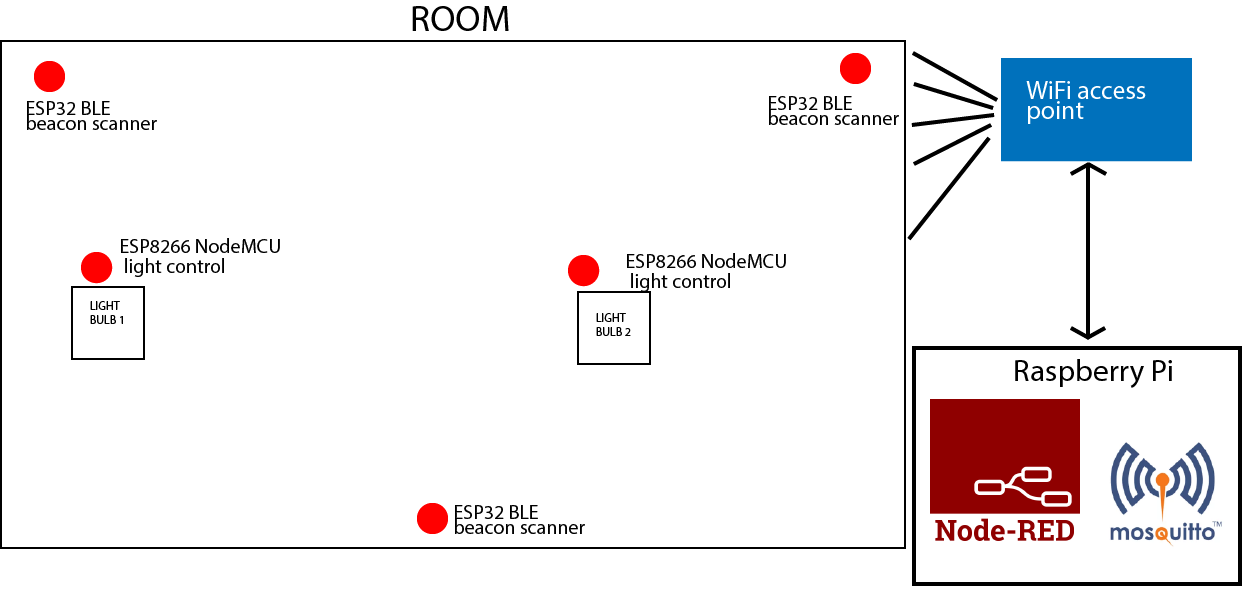
Position dependent room automation system using Bluetooth LE beacons

**Problem intended to be solved**

In presence-based automation systems, there is little intelligence of the actual position of the user by the system. The proximity based BLE beacons fails to accommodate scenarios where the user is not very close to the system. We intend to solve this problem by using 3 BLE beacon scanners. With the proposed system it is desired to achieve positioning even at areas where it is impossible to plant a BLE beacon or a scanner.

**System overview**

Graphical user interface, application

Description automatically generated**The designed system consists of 3 ESP32 modules working in BT + WiFi dual mode. The user must have a beaconing device when he enters the room. This is achieved by using a generic app in Playstore. We have used iBeacons in the implemented system.**

A picture containing diagram

Description automatically generated

**The UUIDs of the beacons are used to identify a known user. When a known beacon is received by any of the beacon scanners, they publish the UUID and the Received Signal Strength Indication value along with the scanner ID to a specific topic on MQTT broker running on the RaspberryPi. The published data from all three beacon scanners is processed by a Node Red flow which is also hosted on the RaspberryPi. The relevant decisions are taken according to the data and the two lights are turned on accordingly.**

**Furthermore, the system integrates a weather API to support the decision making. One of the lights will only turn on after a previously set time of the day. The Node-Red dashboard is used for this purpose and other important data are also displayed on the Node Red dashboard.**

**The designed system is intended to identify three positions inside the room distinctively.**

**System operation**

**User with a beacon enabled smart device enters the room. The RSSI values corresponding to the UUID of the device is sent to the Node Red flow. It identifies whether the user is inside the room or not using the RSSI values of the received beacons and operates the lights accordingly.**

**The RSSI values depends heavily on the device being used. Hence initially a calibration process is carried out. User must use the “Calibration” tab of the MQTT dashboard and select his device and the position he intends to calibrate, and then start the calibration by clicking the button. He and the device should be stationary during this process. The Node-Red flow stores the corresponding RSSI values in text files on RaspberryPi local storage. Hence this calibration is only needed to be carried out once, the data is preserved even with system restarts.**

**After calibration, the flow reads in the data from the files. The current RSSI values from a specific device is continuously checked with the previously calibrated data to identify the current position of the user. The detected position is displayed in the Node Red dashboard as well.**

**If the user is inside the room & the set time has passed or user is in a specified position the lights will turn on. This is achieved by using two ESP8266 Node MCUs which are also subscribed to the same MQTT broker on different topics. A relay module is used to handle high currents.**

**Implementation**

***Hardware***

***ESP32 modules:***