



Internet of Things - 089073

# Home Automation and Monitoring

*Angelo Claudio Re*  
*Giovanni Scotti*

***Professor: Matteo Cesana***

Academic Year 2017/2018

Document version: 1.0

# Contents

<b>Contents</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Purpose . . . . .	2
1.2 Scope . . . . .	2
1.3 Definitions, Acronyms, Abbreviations . . . . .	3
<b>2 System Implementation</b>	<b>4</b>
2.1 Architecture Overview . . . . .	4
2.2 Configuring the RPi . . . . .	5
2.2.1 Mosquitto . . . . .	5
2.2.2 Node-RED . . . . .	5
2.3 Getting started with ESP boards . . . . .	5
2.4 Connecting to ThingSpeak . . . . .	5
<b>3 System Deployment</b>	<b>6</b>

# Section 1

## Introduction

### 1.1 Purpose

This document is intended to describe a *Home Automation and Monitoring* system, its electronic components as well as its constraints and the interaction with the real world and the end user. A useful work environment example will be provided to clarify the scope of the system itself and how it can be deployed.

The documentation is full of diagrams, pictures and schematics to let the final user easily implement the infrastructure and take advantage of its countless features.

This document is mainly addressed to computer scientists, IoT enthusiasts, makers and anyone, with some electronics and programming skills, who wants to keep an eye on his/her home and to control devices remotely.

### 1.2 Scope

The *Home Automation and Monitoring* system aims to offer a smart solution to home automation and monitoring needs. It is intended for those kind of users who want to visualize information about their home, such as temperature or humidity, and control devices remotely.

The system consists mainly of:

- a *back-end*, which is in charge of managing MQTT messaging protocol and running Node-RED. Data coming from boards provided with sensors and actuators are gathered and sent to a remote service provider to store information into the cloud and to open up further analysis.

- a *front end*, which offered by a user-friendly and web-accessible Node-RED dashboard that can be easily reached from any device connected to the same network or to the Internet by means of port forwarding.

The system must be secure. This is why authentication is required in order to access the dashboard. Moreover, MQTT must be secured too and clients provide credentials to the broker to join the network. Last but not least, power consumption has been taken into account: battery-powered boards can enter a deep sleep state after publishing data.

## 1.3 Definitions, Acronyms, Abbreviations

- **AP:** wireless Access Point, it allows Wi-Fi devices to connect a to wired network. Usually it is an integral component of routers for home or office use.
- **Back-end:** any device or computer program that remains in the background and offers application logic and communication interfaces to work with the front-end counterpart. It can provide a data access layer.
- **Front-end:** any part of a system the users directly interact with. It provides the so called presentation layer.
- **IoT:** Internet Of Things.
- **LAN:** Local Area Network.
- **MQTT:** Message Queuing Telemetry Transport is a publish-subscribe messaging protocol.
- **System:** the entire infrastructure and all the devices involved in the Home Automation and Monitoring project.

## Section 2

# System Implementation

### 2.1 Architecture Overview

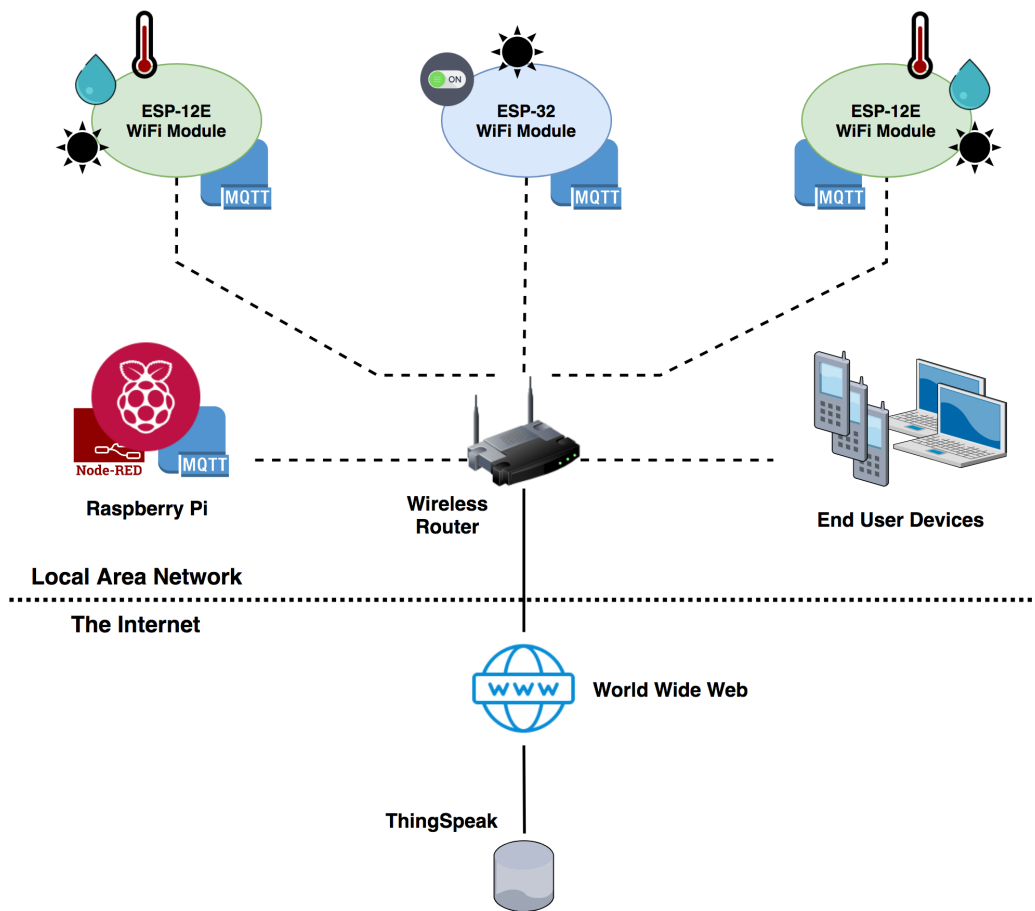
The overall system architecture is depicted by Figure 2.1.

To begin with, the whole system mainly operates within the user Local Area Network. A wireless router is the central point of the entire infrastructure because all the involved devices are connected to it via wireless communication.

The Raspberry Pi board is the backbone of the back-end processing activities. It is in charge of running Node-RED and acting as MQTT broker. It collects information that are sent out by several WiFi-capable modules taking advantage of the publish/subscribe MQTT connectivity protocol. Each module is provided with sensors that gather information about the surrounding environment.

ESP-12E and ESP-32 are the Wi-Fi boards that have been chosen to design the infrastructure. There exists a great variety of sensors that can be hooked up to these ESP modules and ease of programming via Arduino IDE makes them an appropriate choice too. Moreover, they offer full TCP/IP stack support. Each board does not only publish data, but can also subscribe to a specific topic and receive remote commands to control actuators, such as relays.

The end user is able to visualize a user-friendly, web-accessible dashboard directly from his/her own devices, which must be in the same network of the other system components, in case port forwarding is disabled. The interface is provided by Node-RED development tool. Furthermore, data are forwarded to ThingSpeak by the Raspberry board in order to log them in a remote database and to enable further analysis: ThingSpeak has integrated support from the numerical computing software MATLAB.



**Figure 2.1:** System architecture overview. Dashed lines represent wireless communication, solid lines stand for wired connections.

## 2.2 Configuring the RPi

### 2.2.1 Mosquitto

### 2.2.2 Node-RED

## 2.3 Getting started with ESP boards

## 2.4 Connecting to ThingSpeak

## Section 3

# System Deployment