Fast-prototyping Approach to Design and Validate Architectures for Smart Home

Teodoro Montanaro, Ilaria Sergi, Giovanni Marco Bisanti, Alessandro Cambò, and Luigi Patrono

Original scientific article

***Abstract*—The Internet of Things has contributed to make smarter houses and buildings in the last decades. Different existing works already integrate IoT technologies in homes, but end-user needs continuously change and researchers must face this challenge in identifying platforms to fast prototype solutions satisfying these new needs. This paper presents a solution that demonstrates how well-known fast-prototyping technologies like Node-RED, IBM Watson, Telegram, Raspberry Pi 4, and secured MQTT can contribute to develop complex systems facing the challenge. The selected tools are used within a smart home context to support features inspired by people needs and allow users to: a) consult real time conditions (i.e., temperature, humidity, gas), b) remotely manage lights, c) save energy through a light management system based on user movements, d) remotely monitor the house through dedicated webcams, e) generate warning notifications in case of danger. Users can interact with the systems through a web Node-RED dashboard and a Telegram bot. Differently from existing works, the feasibility of the implemented system and the efficacy of the exploited platforms are demonstrated through a running scenario extracted from a consolidated study on user needs in smart homes. The performed experiment can facilitate the fast prototyping of new solutions.**

Index Terms*—Fast-prototyping, IBM Watson, IoT, MQTT, Node-RED, Raspberry Pi, Smart Home, Telegram.

I. Introduction

Within the last decades, Internet of Things (IoT) has been bringing innovation in almost every sector by facilitating and enhancing almost every part of people lives [1], [2]. The home domain is one of the sectors that is mainly benefitting from these enhancements [3] and the Smart Home concept is the result of such integration. As declared by Waleed et al. [4], it represents "living environments supplied with advanced intelligent technologies that manipulate and respond according to the requirements of the home residents".

Different works and commercial solutions have already been proposed to simplify daily activities or provide new services to enhance user comfort in the home context. In fact, literature is full of artifacts that enhance security, user experience in common activities (e.g., make a cup of coffee at the right moment), or cleaning duties [5]-[8] and huge number of commercial products are created every day to introduce or increase entertainment in people lives (e.g., Google Nest Mini device can play a song while the user is having a shower). However, people's wishes continuously change and researchers need fast prototyping tools and platforms to quickly develop new solutions, investigate the effects of their proposals and consequently contribute to the enhancement of our society.

As declared by [9], prototypes are mainly designed and developed to evaluate new ideas to reveal design weaknesses, clarify technical requirements and try out innovations to be sure they actually fit the designed purpose. Specifically, in rapid prototyping a trial version of the designed solution is created to perform a sort of "test drive" of an idea that could be completely changed in the next phases. Furthermore, it is also exploited for teaching and learning activities.

Different artifacts have been already proposed in literature on the combination of the fast-prototyping domain and the smart home environment and, they are mainly focused on two main objectives: a) demonstrate the effectiveness of new programming languages, framework or hardware for the fast-prototyping of smart home solutions, and b) demonstrate that a small group of existing fast-prototyping tools can be effectively applied to the smart home domain to develop new solutions. Based on the experience described in these existing related works, this paper proposes a solution that can be grouped in the second category. In fact, as highlighted by some cited works, the fastest and easiest way of facilitating fast-prototyping is through the exploitation of existing solutions instead of new programming languages, frameworks or hardware.

This paper exploits various fast-prototyping technologies and platforms most frequently applied to IoT [14] - [22], [25], [26] to design a system architecture to support the development of fast-prototyping IoT projects in smart home environment. Node-RED [31], IBM Watson [32], Telegram [33], Raspberry Pi 4 [34], Android [35] platforms and secured MQTT protocol [36] are exploited to demonstrate that it is actually possible to support most of the features needed by people in smart homes. Differently from existing works, the selection of such features was supported by the work proposed by Prange et al. [27] that represent a solid and consolidated study of user needs in homes. As a testbed, a running scenario is designed and implemented focusing on the most desirable features for a smart home: a) consult real time house related condition (i.e., temperature, humidity and gas), b) remotely manage lights, c) save energy thanks to an autonomous light management system based on user movements monitoring, d) remotely monitor the house through dedicated webcams, e) generate warning notifications

Manuscript received January 7, 2021; revised May 31, 2021. Date of publication June 17, 2021. Date of current version June 17, 2021.

Authors are with the Department of Engineering for Innovation, Università del Salento, Lecce, Italy.

E-mails: {teodoro.montanaro, luigi.patrono, ilaria.sergi}@unisalento.it and {giovannimarco.bisanti, alessandro.cambo}@studenti.unisalento.it).

Digital Object Identifier (DOI): 10.24138/jcomss-2021-0005

in case of danger.

The rest of the paper is organized as follows: Section II analyzes motivations and existing related works, Section III reports a scenario used to identify requirements and needs, Section IV describes the modular proposed architecture and Section V describes the details of the technical implementation of the prototype realized to demonstrate how fast-prototyping solutions could simplify the development of smart-home solutions. Then, Section V presents the validation scenario used to test the system and discusses, together with Section VII the outcome of the work. Finally, Section VI concludes the paper with some considerations and future works.

II. Related Works

Several works can be found in literature regarding the smart home context. In this section we will analyze only the ones that particularly exploit fast-prototyping solutions.

Two main approaches have been identified in existing works. The works grouped in the first one mainly propose new frameworks, programming languages, or hardware to simplify the design and development of rapid prototypes. On the contrary, the second one mainly exposes solutions that exploit existing systems to understand if such tools applied to a specific sector actually facilitate the fast prototyping of innovative solutions. Artifacts related to both groups are presented in this section, to motivate the choice of focusing this paper on the second category.

Various artifacts belong to the first category. Invidia et al.[10] present a solution based on the STM32 Open Development Environment (ODE) [28], a framework that provides a complete suite of hardware and software tools to create BLE-based star topology networks for a wide range of applications. This work shows, through a real use case, how all the provided tools can be used to fast prototype applications addressing all user requirements in a smart home context.

Peterson et al. [11] present a comprehensive study on how the usage of the JavaScript programming language can foster the development of rapid prototyping in educational activities and specifically in laboratory classes. Even though the objective of the work is a bit far from the one presented in this paper, the research is dropped in the smart home context and, considering that the educational sector has almost the same requirements as the research one in terms of fast prototyping, the results presented by the study are relevant for the present work.

A similar work is proposed by Lee et al. [12]. They propose a new framework for fast prototyping: Esperanto. It is composed by a new programming language based on the well-known C++ language, a compiler and a runtime environment. Authors demonstrate the efficacy of the proposed framework in developing IoT prototypes with respect to other existing solutions.

Rizzo et al. [13] proposes UAPPI, an open source extension of the prototyping environment "MIT App Inventor" [29] to support the fast prototyping of Internet of Things solutions that uses Machine Learning (ML) techniques for their purposes. The proposed solution is tested in a smart home environment in which the usage of the UDOO X86 board [30] was integrated with the UAPPI interface to exploit existing ML models to map predefined inputs to desired outputs.

On the other hand, a similar number of artifacts belongs to the second group of works that exploit existing solutions to establish if they actually facilitate the fast prototyping of innovative solutions.

Tanganelli et al. [14] present a comprehensive analysis of the solutions for developers that want to realize a fast prototype in the IoT domain. The paper is mainly concentrated on the usage of the Python programming language on top of the presented platforms and shows the benefits of adopting Python as a language for IoT development, specifically when rapid prototyping is a concern. It is actually an article that is at the border of the two groups, in fact it neither presents a new programming language nor exploits existing tools, but it is cited because it highlights the benefits that can be brought by the usage of existing programming languages (or tools) instead of new ones, and, consequently, highlight the reason why the present paper is focused on the category of papers that exploit existing solutions.

Furthermore, Fandrich et al. [15] propose a fast-prototyping kit based on three main platforms: the 32-bit microcontroller ESP8266 used to connect sensors, the Raspberry Pi 4 board used as central unit and the Node-RED framework used for linking all the components. In addition, all the communications have been run through the MQTT protocol. The reported experiments demonstrate that Node-RED with some adaptations is a good didactic tool for smart home projects and, consequently, a good instrument for fast prototyping purposes. Differently from the present paper, the selection of the features to be implemented is not based on a solid up-to-date consolidated study that strengthen the demonstrated feasibility and the possibility of reusing similar solutions in future smart home projects.

Lekić et al. [16], present the implementation of an IoT application that exploits the DHT11 temperature and humidity sensor [37], the Raspberry Pi, the Node-RED platform and the IBM Bluemix. The platforms are used to develop a simple use case in which the temperature and humidity are collected and shared on the IBM Bluemix platform. Authors do not aim at demonstrating how the considered tools can boost the development of fast prototyping solutions and, in addition, not even in this case, the selection of the features to be implemented is not based on a solid study.

Chauhan et al. [17] present a solution in which fast prototyping technologies like Google Firebase, Raspberry Pi, Node-RED are exploited to support a smart home use. Also in this case the purpose of the paper is not concentrated on the advantages taken by the usage of rapid prototyping tools and the selection of the features to be implemented is not based on a solid study.

Different other works have been proposed without the specific aim of demonstrating the effectiveness of fast-prototyping solutions in smart homes, and consequently, they are different from the present paper because of the complete absence of a superfluous analysis of the advantages that the used tools could bring in narrowly designed use cases.

One of these works is the one proposed by Zare et al. [18]. Authors exploit analog sensors like the ESP32, Node-RED, and the MQTT protocol to support a Supervisory Control and Data Acquisition (SCADA) system for home monitoring and control.

Kishore Kodali et al. [19] present the development of a home automation system that is based on MQTT, Node-RED, IFTTT

[38], Mongoose OS [39], Android app and Google Assistant platforms.

Ibrahim et al. [20] propose a solution that leverages on an Android-based mobile application, a Raspberry Pi, the Node-Red and the ThingSpeak platform [40] to support a web-based surveillance system.

Prabaharan et al. [21] present a comprehensive solution that exploit the NodeMCUs [41] and the Adafruit IO platforms [42] to manage an automation system in home environment. The system remotely authorizes people to control all the appliances and the doors in the house, to consult the total power consumption, and so reduce the wastage of power, and to detect water leakage and motion.

De Luca et al. [22] developed and validated an architecture, both hardware and software, able to monitor and manage a KNX-based home automation system through an Android mobile device in an efficient and safe way.

Tseng et al. [22] proposes an IoT-based home automation system that leverages on Chatbot UI, Node-Red and the MQTT protocol to feed an autonomous system that enhance home environment monitoring and control function.

Caione et al. [24] propose an experience in rapid design and prototyping of an ambient assisted living system that detects the movements of elderly people in their home, acquiring data through sensors in an unobtrusive way.

Finally, with the aim of demonstrating that the MQTT protocol can be useful in various situations, Cornel et al. [25] also use fast-prototyping platforms like the Raspberry Pi, OpenHAB [43] and Mosquitto [44] to realize a smart home automation system.

In addition to their contribution in highlighting the gaps filled by the present paper, the analyzed works allowed the identification of the most used fast-prototyping tools and environment to be exploited within the present paper. Furthermore, they point out the benefits that can be brought by the usage of existing programming languages (or tools) instead of new ones, and, consequently, highlight the reason why the present paper is focused on the category of papers that exploit existing solutions.

III. Scenario

With the aim of identifying the most significant features that people wish to benefit from in their home, the study conducted by Prange et al. [27] has been used as inspiration.

In fact, authors conducted an online survey in which they assessed situations of users wishing for features or devices in their home to be smart(er). Participants described various scenarios for their desired smart home appliances that can mainly be divided into three phases: (1) (before) coming home, (2) being at home, and (3) leaving home. Before coming home, users wished to pre-control devices to save time upon arrival or increase comfort. Within their home, users imagined remote control of arbitrary devices with the aim of easing certain tasks or increase comfort. Finally, when leaving or being away from home, users would appreciate a possibility to check the state of their or access information about their home with the aim of reach the users' "peace-of-mind".

Starting from such overview of user needs, this paper identified the following situations that fast-prototyping frameworks should help to support:

- Monitor the temperature, humidity and gas/smoke level in the house either from within the house or from outside
- Receive warnings about abnormal values about monitored parameters
- Manually switch on or off the lights through a remote application
- Automatically switch on or off the lights when a user come in/out from a room
- Recognize user presence in the house
- Visualize the house rooms through a camera when needed

Consequently, starting from the description of the just shown situations, we describe a simple scenario that will guide the presentation of the following sections.

We focus our attention on a generic participant named Bob.

Bob is at work, but it is the end of the working day, so he closes his notebook and starts to think at the next hour. He has to buy groceries for the dinner and then he would like to come back home to have a shower, but he does not know if the temperature in the house is the ideal one for doing it. Unfortunately, he has never installed any smart system to consult such an information, so he goes shopping and then he comes back home but he finds out that the inside temperature is 15 degrees Celsius. He turns on the heating system and waits an hour to have the right temperature for having the shower.

Then, while he is having the shower he hears a noise at the second floor of his house. He is worried, so he exited the shower and he runs to the place in which he had listened the noise to understand that it was the cat that had broken a jar.

After the shower he goes to bed, but he notices that he had forgot to turn off the light in the kitchen so, he has to go there to turn it off.

In the morning he wakes up late so he has the fastest breakfast he has ever had and he runs to the office. At ten, the neighbor calls him because he smells a bad smell of gas. So he runs at home to discover that he was only a false alarm. He comes back office, but he decides to exploit his knowledge in IoT domain to create an easy smart home system to help him in his daily activities.

IV. System Architecture

Starting from the presented scenario, a modular architecture has been designed with the aim of providing an improved experience to Bob.

As depicted in Fig. 1, the system is composed of five main components described in the following paragraphs. Every IoT system leverages on sensors to collect data from the field and actuators to act on the environment (*Sensors & Actuators*). In this case Bob would like to monitor the temperature of his home, the humidity, the gas level and the presence of people. In addition, he would be able to turn on a camera when he is warned about an abnormal situation (e.g., a noise in another room while he is having a shower). Furthermore, he would be able to turn on/off the light through a dedicated actuator.

The *Local Gateway* is a central device that enables the communication between all the sensors and the central Cloud Gateway. It usually standardizes data before sending it to the other components and exposes a standard interface (e.g., REST APIs or publish/subscribe services like the MQTT one) to let other components send commands to the actuators without caring about the specific supported protocol or technology.

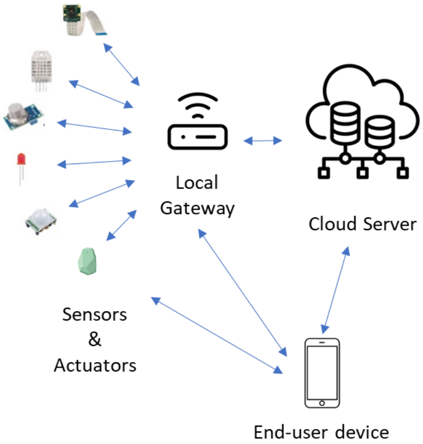


Fig. 1. System Architecture design.

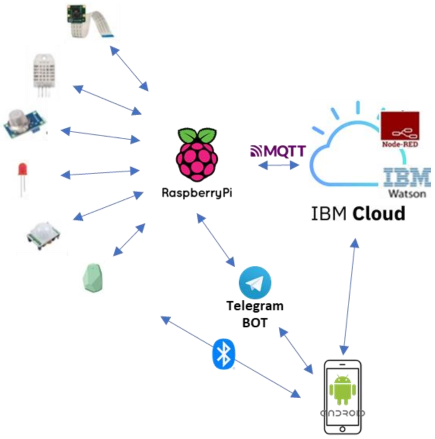


Fig. 2. Implementation details of the System Architecture.

The *Cloud Gateway* is a remote server that collects the standardized data and provides it to both users (through a Dashboard) and other systems through dedicated services. In addition, it usually collects data into a dedicated database to let also access historical data and provides a dashboard and some APIs to access and consult data.

The *End-user device* is a device that the user exploits to access the platform and, in some cases, to also send direct commands in the local network to specific devices. In addition, the end-user devices can be used to monitor user behavior and status (e.g., Bob could be interested in using his smartphone to trace his presence in the house).

In the next Section a prototype of the described architecture will be implemented through some selected existing fast-prototyping platforms to demonstrate the feasibility and ease of creation of such type of systems.

V. Implementation

In order to demonstrate how fast-prototyping solutions could simplify the development of smart-home solutions, in this paragraph a prototype of the proposed architecture will be implemented and tested on the designed scenario.

Fig. 2 reports a specialized version of the architecture reported in the previous section, in which it is possible to depict which are the existing platforms used for each component.

Some of the most frequently used platforms and technologies for fast-prototyping have been selected from literature for each component. The following section will present each of them.
As shown in Fig. 3, in this first prototypal experiment a testbed version of a house has been developed. At first, different sensors and actuators has been used within the current work. In this first prototypal experiment, the real devices usually used in a home, like smart lights or thermometers, have been substituted through fast-prototyping sensors connected to a "Raspberry Pi 4" board. Due to its popularity, in fact, it supports developers with a huge number of documented tutorials and operational developer guides to help the rapid development of prototypes. It is designed to work with a Linux based operating system and it is equipped with dedicated ports and hardware to be easily used to get data from real sensors. The configuration used to implement the prototype is shown in Fig. 4.

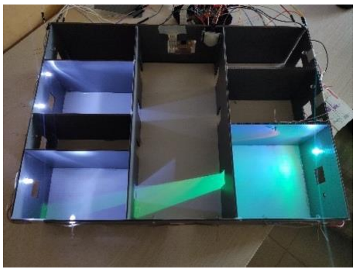


Fig. 3. Testbed version of a house used within the prototypal experiment.

At first, several LEDs has been used to simulate the home lights. In fact, imaging the house made by 7 rooms (living room, kitchen, 2 bathrooms and 3 bedrooms), a LED has been connected to the *Raspberry pi 4* board for each room of the house. In addition, for the kitchen and the living room an RGB LED has been also selected to simulate the possibility of changing intensity and color of the room light. Then, as temperature and humidity sensor, the *DHT22 sensor* has been selected. It allows to reveal the temperature and humidity in the house either from within the house or from outside. In addition, the MQ-5 and Buzzer analog sensors have been used to detect the presence of smoke or rather gases (in ppm), mainly combustible ones in the house.
Furthermore, a Passive InfraRed (PIR) has been used to detect user movements in a room and a PiCamera has been exploited to collect pictures or videos of the house when it is required. In addition to this configuration, a further technology was used to support the user movements detection: 4 BlueUp Beacons [45] virtually installed in the 4 different rooms were, in fact, used to detect registered people (with a registered smartphone) entrance. These sensors are the only ones that are not connected to the Raspberry Pi 4 to acquire data, in fact the smartphone app is responsible for their identification and the communication of the information to the Cloud Server, but the details will be discussed later.

Then, the Raspberry Pi 4 board already used for getting data from the sensors has been also selected as the best candidate for the implementation of the Local Gateway of the system due to its flexibility. For the purpose of this paper, the communication between the Local Gateway and the Cloud Gateway has been based on the publish/subscribe paradigm and specifically on the MQTT protocol, consequently, the Local Gateway was set to

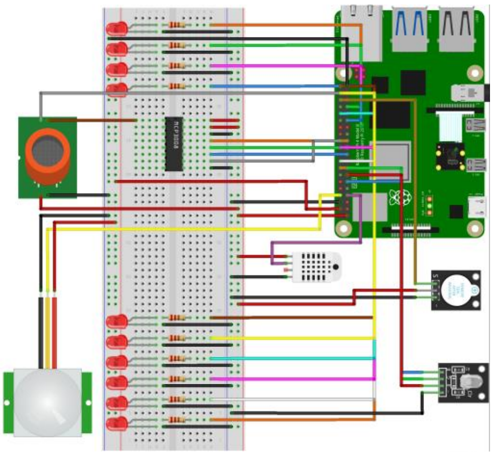


Fig. 4. Raspberry Pi and Sensors configuration used for the implementation of the prototype.

publish data and receive commands to/from an MQTT Broker exposed by the Cloud Server. In fact, on some dedicated topics both the Cloud Server and the Telegram BOT are allowed to send commands to the Gateway (then forwarded to the actuators) and receive new data or notifications revealed or generated by the sensors.

Thus, as a summary, within the present paper, the Raspberry Pi 4 has been used to *a)* collect data from all the devices apart from the BlueUp beacon devices, *b)* interact with the IBM Cloud to store all the collected data, *c)* receive commands from the Telegram BOT to turn on/off the camera and the lights, *d)* send images and videos to the Telegram BOT if the functionality is activated, and *e)* send notifications generated by the Cloud Server.

The "IBM Cloud" has been selected as fast-prototyping Cloud Server of the system. It is a platform that combines platform as a service (PaaS) with infrastructure as a service (IaaS) by providing a set of cloud computing services. Within the present work it has been used to reach three main aims: *a)* collect data from the sensors and the Android application, *b)* provide a web-based Graphical User Interface (GUI) for configuring automatic actions, and *c)* save received data into a dedicated internal database. The first aim has been achieved through the adoption of the "IBM Watson IoT" platform provided by the "IBM Cloud" environment. Specifically, it hosts an MQTT Broker to which sensors can send all the collected data and from which the GUI can get needed information to, then, take decisions and perform automatic actions. In addition, the GUI has been provided by the "Node-RED" platform that is hosted on the "IBM Cloud" and is connected to the "IBM Watson IoT" interface. Node-RED is a web-based Node.js tool for graphically wiring hardware devices, APIs and online services. The dashboard allows to monitor the temperature, the humidity and the gases revealed in the house, but it also exposes some buttons to turn on/off lights and activate the camera.

Fig. 5 show the NodeRed configuration set to obtain gas level data and entrance monitoring (the one revealed through the Beacon) and propagate them to the dashboard and generate needed notifications. Other similar configurations were also set for all the other features exploited within the realized prototype.

Furthermore, considering that the Node-RED does not provide its own authentication system, a simple but robust PIN verification system has been implemented as suggested in various Node-RED tutorials. To this aim, a specific PIN interface has been instantiated on the Raspberry Pi 4 board and, consequently, the Node-RED dashboard has been connected to it. Consequently, when a new data arrives from the connected sensors, it is published on the MQTT Broker and shared with the Node-RED platform. Then, if the user wants to see such data he can unlock the Node-RED dashboard through the PIN interface and, through the shown GUI, he can both see the data and set the rules to be applied to take decision in case of necessity (e.g., send a notification to the user).

Moreover, an Android App has been developed to manage two main different situations: a) interact with the BlueUp Beacon modules to support the needs of identifying people inside a room through the triangulation of the signal power received by the Android app from each near sensor (every time Bob will enter a room the Beacon technology will recognize him and register his entrance), b) send all the revealed data to the IBM Cloud to let it make decisions (e.g., turn on the light when Bob enters a room) and update the Node-RED dashboard.

Fig. 6(a) shows a screenshot of the app section responsible for the detection of the entrances..

Finally, a Telegram BOT has been also implemented by using the Python programming language. As shown in Fig. 6(b) that presents a screenshot of an interaction with the BOT, it is responsible for different purposes: a) when a notification is generated by the Cloud server, a notification is sent also to the Telegram BOT that will redirect it to registered users or groups, b) receive commands from the users to enable or disable camera in case of necessity (e.g., Bob hear a noise in another room, so he turns on the camera to see if anyone is present in the room) to take a picture or a video. In the last case, the BOT allows also to select the number of seconds of a video to be seen.

All the communications among all the blocks are established over a secured MQTT connection to guarantee reliability, speed in communications and security.

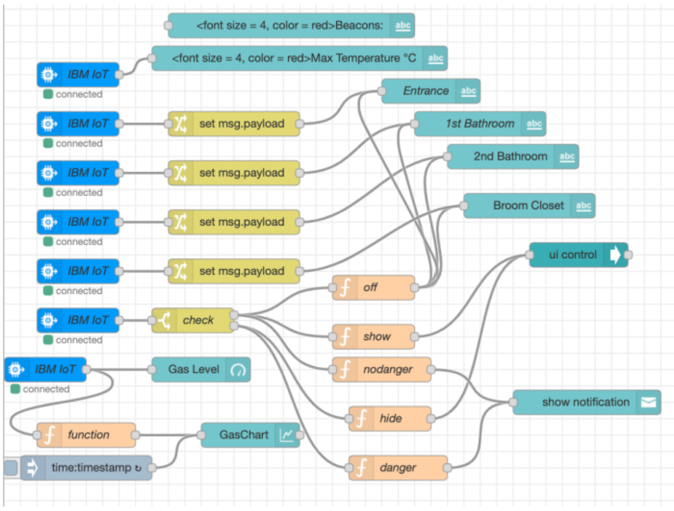
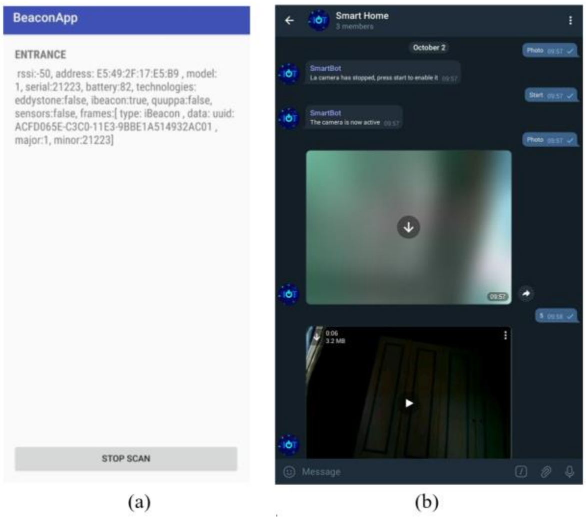


Fig. 5. Detailed overview of the NodeRed configuration related to gas and beacon monitoring.



BeaconApp

ENTRANCE

rssi:-50, address: E5:49:2F:17:E5:B9 , model: 1, serial:21223, battery:82, technologies: eddystone:false, ibeacon:true, quuppa:false, sensors:false, frames:[type: iBeacon , data: uuid: ACFD065E-C3C0-11E3-9BBE1A514932AC01 , major:1, minor:21223]

STOP SCAN

Smart Home
3 members

October 2

Photo 09:57

SmartBot
La camera has stopped, press start to enable it 09:57

Start 09:57

SmartBot
The camera is now active 09:57

Photo 09:57

09:57

5 09:58

0:06
3.2 MB

Message

Fig. 6. Screenshots of the Android mobile apps. Entrance detection (a) and Telegram Bot interaction (b).

VI. Validation

To demonstrate the actual efficacy of the implemented fast-prototyped solution the running scenario presented in Section III has been emulated within the developed testbed actually taking care at the possible improvements to the "bad" effects revealed in the scenario due to the absence of a smart home system.

Consequently, the following "Validation Scenario" inspired by the one presented in Section III was designed and tested.

Bob is at work, but it is the end of the working day, so he closes his notebook and starts to think at the next hour. He has to buy groceries for the dinner and then he would like to come back home to have a shower, consequently he opens the Node-RED web interface through the browser of his smartphone and he actually see that the temperature in the house is set at 15 degrees Celsius so he decides to come back home, turn on the heating system before going shopping.

Then, while he is having the shower, he hears a noise at the second floor of his house. He is worried, but he can count on the Telegram BOT to get pictures of the house and the Node-RED dashboard that does not show any notification if no one was revealed in coming in the house. In fact, he realizes that the noise was caused by the cat, so he continues to have the shower and, only then, go to clear the floor from the broken jar.

After the shower he goes to bed, but he notices that he had forgot to turn off the light in the kitchen so, he uses the Node-RED dashboard to turn it off.

In the morning he wakes up late, so he has the fastest breakfast he has ever had, and he runs to the office. At ten, the neighbor calls him because he smells a bad smell of gas. So, he opens the Node-RED dashboard to see that no gas leak is present in the house, so he continues to work without any concern.

During the experiments also some performance evaluations were performed on the interactions with the actuators, the one that from our point of view are the most important because, actually, can make the difference in warning or dangerous situations (e.g., if an actuator is present and can turn the gas off). Specifically, the time needed to perform the requested action was monitored. Fig. 7 shows two charts reporting the time needed to perform 2 actions: a) turn on/off a light through the Node-RED dashboard, b) taking a picture through the camera managed through the Telegram BOT. The experiments were performed through a FTTC 100 Mbps connection (actual network performance are the following. Down: 40Mbps - Up: 7Mbps). The charts reveal that within 6 trials, the command is executed in less then 40ms even though different steps are followed (Node-RED + MQTT + Raspberry). It is a good result for a fast prototypal solution that users can exploit to understand if a solution can be really useful in specific contexts or for solving specific problems.

VII. Discussion

As already discussed in the previous section, to finally demonstrate the actual efficacy of the implemented fast-prototyped solution the Validation Scenario presented in Section VI has been emulated within the developed testbed.

The first situation described in the scenario was related to Bob's needs about consulting the current status how his home from his office. By using the implemented Node-RED based GUI Bob is now actually able to look at such an information from his office. Thus, if he notices that the temperature is not ideal for having a shower he can for sure anticipate his coming back home to adjust the temperature as soon as possible, however he could also use other services exposed by external third party smart heating systems to start to heat the house up or to close a window if it is the cause of the cold.

Then, the second situation that Bob would have liked to solve is the one in which he is having the shower and he hears a noise at the second floor. Now, through the service provided by the BlueUp Beacon and the PIR devices he can immediately know that nobody entered the room and through the camera and Telegram, he can ask the system to record a video of 20 seconds to see what is happening at the second floor. He can now see his cat and the broken jar. In addition, some additional services could be added to these sensors and systems, for instance, to directly recognize the cat through specific deep learning techniques and directly notify Bob with a message broadcasted through an Hi-Fi saying "Bob, don't worry, it was the cat".

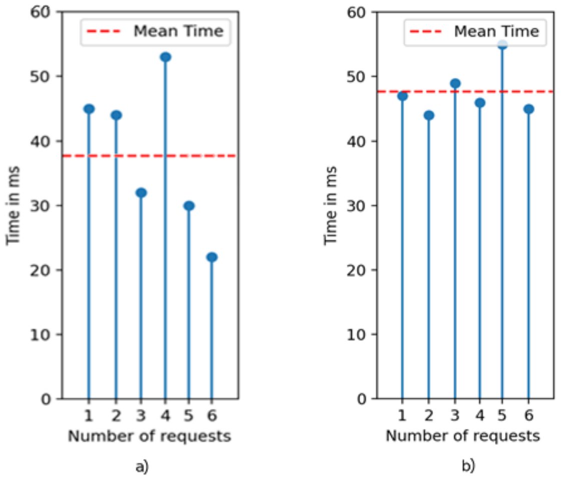


Fig. 7. Mean time needed to a) turn on/off a light and b) take a picture through the supported camera.

These scenarios will be further evaluated in future works that will also evaluate Machine Learning tools for fast prototyping.

In addition, Bob is now able to turn the kitchen light off from his bed and he can verify if any gas leak is actually present in the house.

Consequently, the developed solution can actually work as expected even though it has been developed through fast-prototyping platforms and tools and this actually demonstrates that such fast-prototyping environments can help developments in smart home context.

Obviously, the developed setting is not a complete solution that can serve every situation that can occur in an house, but the exploited services are only a little part of the whole functionalities exposed by the used platforms, so, future works can, for sure, expand the prototype with additional support to also new situations and needs and introduce new technologies and platforms to support new and/or more complex scenarios.

In addition, Artificial Intelligence modules could be integrated in the proposed solution to further enhance the already provided services, like the identification of people present in a room, the automatic identification of the cause of a noise, or the automatic activation of the heating system when the user is expected to come back home and have a shower.

VIII. Conclusion

Due to the continuous development and commercialization of new IoT devices for the smart home context, user needs continuously change and the community needs fast-prototyping solutions to be prepared for such rapid necessities. This paper demonstrated how it is possible to develop a complex smart home system implementing features extracted by a solid and up-to-date study on user needs in smart homes, through some well know fast-prototyping platforms and technologies like Node-RED, IBM Watson, Telegram, Raspberry Pi 4, and the secured MQTT protocol. A running scenario has been used in both the design and the development of a reference architecture and, finally, it has been exploited to demonstrate the efficacy of the proposed solutions. Through the developed infrastructure, in fact users can perform the actions that, for the considered literature, are the most important for people, such as: a) consult real time house related condition (i.e., temperature, humidity and gas), b) remotely manage lights, c) save energy thanks to an autonomous light management system based on user movements monitoring, d) remotely monitor the house through dedicated webcams, e) generate warning notifications in case of danger. The performed experiment can be exploited as inspiration for future works that would propose innovative fast prototyping solutions. Due to their similarities in terms of necessities and tendency of exploiting hands-on for their purposes, both research and education domains can benefit from such work.

References

[1] L. Mainetti, L. Manco, L. Patrono, I. Sergi and R. Vergallo, "Web of Topics: An IoT-aware model-driven designing approach" 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, 2015, pp. 46-51 , doi: 10.1109/WF-IoT.2015.7389025.

[2] E. Fadda, L. Gobbato, G. Perboli, M. Rosano, R. Tadei, "Waste Collection in Urban Areas: A Case Study", 2018, INFORMS Journal on Applied Analytics, Vol. 48, No. 4, doi: 10.1287/inte.2018.0943.

[3] J. Y. Kim, H. Lee, J. Son and J. Park, "Smart home web of objects-based IoT management model and methods for home data mining", 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), Busan, 2015, pp. 327-331, doi: 10.1109/APNOMS.2015.7275448.

[4] J. Waleed, A. M. Abduldaim, T. M. Hasan and Q. S. Mohaisin, "Smart home as a new trend, a simplicity led to revolution", 2018 1st International Scientific Conference of Engineering Sciences - 3rd Scientific Conference of Engineering Science (ISCES), Diyala, 2018, pp. 30-33, doi: 10.1109/ISCES.2018.8340523.

[5] K. Agarwal, A. Agarwal and G. Misra, "Review and Performance Analysis on Wireless Smart Home and Home Automation using IoT", 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 629-633 , doi: 10.1109/I-SMAC47947.2019.9032629.

[6] V. Mighali, G. Del Fiore, L. Patrono, L. Mainetti, S. Alletto, G. Serra, R. Cucchiara, "Innovative IoT-aware Services for a Smart Museum", 2015, AW4City ACM Workshop, International World Wide Web Conference 2015, doi: 10.13140/RG.2.1.3800.1762.

[7] R. Mulero et al., "An IoT-Aware Approach for Elderly-Friendly Cities", in IEEE Access, vol. 6, pp. 7941-7957, 2018, doi: 10.1109/ACCESS.2018.2800161.

[8] A. Almeida, R. Mulero, P. Rametta, V. Urošević, M. Andrić, L. Patrono, "A critical analysis of an IoT—aware AAL system for elderly monitoring", Future Generation Computer Systems, Volume 97, 2019, Pages 598-619, doi: 10.1016/j.future.2019.03.019.

[9] K. Karvinen and T. Karvinen, "IoT rapid prototyping laboratory setup", 2018, International Journal of Engineering Education. 34. 263-272

[10] L. Invidia, S.L. Oliva, A. Palmieri, L. Patrono, P. Rametta, "An IoT-oriented fast prototyping platform for BLE-based star topology networks", (2019) Journal of Communications Software and Systems, 15 (2), pp. 138-149, doi: 10.24138/jcomss.v15i2.682.

[11] B. Peterson and B. Vogel, "Prototyping the Internet of Things with Web Technologies: Is It Easy?", 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Athens, 2018, pp. 518-522, doi: 10.1109/PERCOMW.2018.8480268.

[12] G. Lee, S. Heo, B. Kim, J. Kim, and H. Kim. "Rapid prototyping of IoT applications with Esperanto compiler", 2017, In Proceedings of the 28th International Symposium on Rapid System Prototyping: Shortening the Path from Specification to Prototype (RSP '17). Association for Computing Machinery, New York, NY, USA, 85–91.

[13] A. Rizzo, F. Montefoschi, M. Caporali, A. Gisondi, G. Burresi, and R. Giorgi. "Rapid prototyping IoT solutions based on Machine Learning", In Proceedings of the European Conference on Cognitive Ergonomics 2017 (ECCE 2017). Association for Computing Machinery, New York, NY, USA, 184–187, doi: 10.1145/3121283.3121291.

[14] G. Tanganelli, C. Vallati and E. Mingozzi, "Rapid Prototyping of IoT Solutions: A Developer's Perspective", in IEEE Internet Computing, vol. 23, no. 4, pp. 43-52, 1 July-Aug. 2019, doi: 10.1109/MIC.2019.2927202.

[15] A. Fandrich, T. Stuckenberg, and I. Diethelm, "DIY Smart Home: The Development of an Exemplary Internet of Things Infrastructure for Computer Science Education", In Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '20). Association for Computing Machinery, New York, NY, USA, 523–524, doi: 10.1145/3341525.3393987.

[16] M. Lekić and G. Gardašević, "IoT sensor integration to Node-RED platform", 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, 2018, pp. 1-5 , doi: 10.1109/INFOTEH.2018.8345544.

[17] N. Chauhan, M. Malik, "Home Automation With NoSQL and Node-RED Through Message Queuing Telemetry Transport", (2021) Advances in Intelligent Systems and Computing, 1172, pp. 427-434, doi: 10.1007/978-981-15-5566-4_37.

[18] A. Zare and M. T. Iqbal, "Low-Cost ESP32, Raspberry Pi, Node-Red, and MQTT Protocol Based SCADA System", 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, 2020, pp. 1-5, doi: 10.1109/IEMTRONICS51293.2020.9216412.

[19] R. Kishore Kodali, S. C. Rajanarayanan, L. Boppana, S. Sharma and A. Kumar, "Low Cost Smart Home Automation System using Smart Phone", 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129), Depok, West Java, Indonesia, 2019, pp. 120-125, doi: 10.1109/IEMTRONICS51293.2020.9216412.

[20] S.N. Ibrahim, A.H.H. Basri, A.L. Asnawi, "Development of web-based surveillance system for internet of things (IoT) application", (2019) Bulletin of Electrical Engineering and Informatics, 8 (3), pp. 1108-1116, doi: 10.11591/eei.v8i3.1520.

[21] J. Prabaharan, A. Swamy, A. Sharma, K. N. Bharath, P. R. Mundra and K. J. Mohammed, "Wireless home automation and security system using MQTT protocol", 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), doi: 10.1109/RTEICT.2017.8256958.

[22] G. De Luca, P. Lillo, L. Mainetti, V. Mighali, L. Patrono and I. Sergi, "The use of NFC and Android technologies to enable a KNX-based smart home", 2013 21st International Conference on Software, Telecommunications and Computer Networks - (SoftCOM 2013), Primosten, 2013, pp. 1-7, doi: 10.1109/SoftCOM.2013.6671904.

[23] C. Tseng, C. Cheng, Y. Hsu and B. Yang, "An IoT-Based Home Automation System Using Wi-Fi Wireless Sensor Networks", 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) , doi: 10.1109/SMC.2018.00417.

[24] A. Caione, A. Fiore, L. Mainetti, L. Manco and R. Vergallo, "Rapid prototyping Internet of Things solutions through a model-driven approach: A case study in AAL", 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, 2017, pp. 1-6

[25] A. Cornel - Cristian, T. Gabriel, M. Arhip-Calin and A. Zamfirescu, "Smart home automation with MQTT", 2019 54th International Universities Power Engineering Conference (UPEC), doi: 10.1109/UPEC.2019.8893617.

[26] M. Lekić and G. Gardašević, "IoT sensor integration to Node-RED platform", 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, 2018, pp. 1-5, doi: 10.1109/INFOTEH.2018. 8345544.

[27] S. Prange and F. Alt., "I Wish You Were Smart(er): Investigating Users' Desires and Needs Towards Home Appliances", Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20). Association for Computing Machinery, New York, NY, USA, 1–8, doi: 10.1145/3334480.3382910.

[28] STM32 Open Development Environment (ODE). Available online: https://www.st.com/en/ecosystems/stm32-open-development-environment.html (accessed on 23 December 2020).

[29] MIT App Inventor. Available online: https://appinventor.mit.edu/ (accessed on 23 December 2020).

[30] UDOO X86 board. Available online: https://www.udoo.org/ (accessed on 23 December 2020).

[31] Node-RED. Available online: https://nodered.org/ (accessed on 23 December 2020).

[32] IBM Watson IoT Platform. Available online: https://www.ibm.com/ cloud/watson-iot-platform (accessed on 23 December 2020).

[33] Telegram Messaging App. Available online: https://telegram.org/ (accessed on 23 December 2020).

[34] Raspberry Pi – Fast-prototyping board. Available online: https://www. raspberrypi.org/ (accessed on 23 December 2020).

[35] Android OS. Available online: https://www.android.com/intl/it_it/ (accessed on 23 December 2020).

[36] MQTT Protocol. Available online: https://mqtt.org/ (accessed on 23 December 2020).

[37] DHT Sensors. Available online: https://learn.adafruit.com/dht (accessed on 23 December 2020).

[38] IFTTT platform. Available online: https://ifttt.com/ (accessed on 23 December 2020).

[39] Mongoose OS. Available online: https://mongoose-os.com/ (accessed on 23 December 2020).

[40] Thingspeak. Available online: https://thingspeak.com/ (accessed on 23 December 2020).

[41] NodeMcu, An open-source firmware and development kit that helps to prototype IOT products. Available online: https://www.nodemcu.com/ index_en.html (accessed on 23 December 2020).

[42] Adafruit IO. Available online: https://io.adafruit.com/ (accessed on 23 December 2020).

[43] OpenHAB, Available online: https://www.openhab.org/ (accessed on 23 December 2020).

[44] Eclipse Mosquitto, Available online: https://mosquitto.org/ (accessed on 23 December 2020).

[45] BlueUp Beacons. Available online: https://www.blueupbeacons.com/ index.php?lang=it (accessed on 11 June 2021).

Teodoro Montanaro (Member, IEEE) received the M.S. degree in Computer Engineering in 2014 and the Ph.D. degree in Computer and Control Engineering in 2018 from the Politecnico di Torino, Turin, Italy. From 2017 to 2020 he collaborated with the Istituto Superiore Mario Boella (ISMB) and LINKS Foundation within different European projects like IoF2020, S4G, MONICA, MAESTRI that brought innovations in different fields (e.g., food and farm, smart grid, city, health). Since 2020, he collaborates with the IDentification Automation Laboratory (IDA Lab), Department of Innovation Engineering, Università del Salento. His current research interests include IoT applications focused on the exploitation of fog computing, DLT, blockchain, and AI in different domains like smart grids, smart homes, smart cities, industrial processes, food traceability, and smart health. He has authored different papers on international journals and conferences.

Ilaria Sergi received the M.S. degree in Automation Engineering from the Università del Salento, Lecce, Italy in 2012 and the Ph.D. in Engineering of complex systems from the Università del Salento, Lecce, Italy in 2019. Since 2012, she collaborates with the Identification Automation Laboratory (IDA Lab) of the Department of Engineering for Innovation, Università del Salento. Her research interests include RFID, IoT, smart environments, and homecare solutions. She has authored several papers on international journals and conferences.

Giovanni Marco Bisanti received the B.S. degree in Information Technology Engineering in 2019 from the University of Salento, Lecce, Italy. His thesis concerned the study of an Early-late gate symbol synchronizer in digital communications. He is currently attending a M.S. degree program in Computer Engineering at the University of Salento.

Alessandro Cambò received his B.S. in Information Technology Engineering from University of Salento, Lecce, Italy, in 2019. In his thesis work he studied the difference between analogical and digital communications, focusing on how to migrate from one to the other. He is currently attending a master's degree program in Computer Engineering at University of Salento.

Luigi Patrono (Member, IEEE) received the M.S. degree in computer engineering from the University of Lecce, Lecce, Italy, in 1999, and the Ph.D. degree in innovative materials and technologies for satellite networks from the ISUFI-University of Lecce, Italy, in 2003. He is an Associate Professor of Computer Networks and Internet of Things at the University of Salento, Lecce, where he is also the Pro-Vice Chancellor for Digital Technologies. His research interests include RFID, IoT, wireless sensor networks, and embedded systems. He has authored more than 130 scientific papers published in international journals and conferences. He has been the Organizing Chair of some international symposia and workshops, technically co-sponsored by the IEEE Communication Society, focused on the Internet of Things.