

Wireless Home Automation System using IoT

Alexandra MIHALACHE

Faculty of Cybernetics, Statistics and Economic Informatics

Bucharest University of Economic Studies, Romania

alexa.mihalache@yahoo.com

Nowadays, the chance of having an automated home is no longer a fancy luxury, but a reality accessible to a wide range of consumers, because smart home systems have replaced those that only automated the home in the past. More and more solutions based on IoT are being developed to transform homes into smart ones, but the problem is that the benefits of home automation are still not clear to everyone as they are not promoted enough, so we cannot talk about a broad mass of consumers already using integrated or DIY solutions to improve their lives. In this paper, I will present a home automation system using Arduino Uno integrated with relevant modules which are used to allow remote control of lights or fans, changes being made on the basis of different sensors data. The system is designed to be low cost and expandable, bringing accessibility, convenience and energy efficiency.

Keywords: Home automation, Arduino Uno, Appliances remote control, IoT

1 Introduction

Despite the popularity gained in the '80s, home automation has never reached the market, especially because of economic reasons, whereas installing applications, purchasing devices, and maintenance were very expensive. Apart from costs, another problem was that applications integrated into an automated system are considered to be open information systems, so they could be attacked at any time by malicious persons or controlled by unauthorized individuals. It took more than two decades before technology win the battle against security doubts and the concept "home automation" became reality.

In the current context, home automation system has less to do with its predecessor, the domotic system, because it is based on a revolutionary concept: IoT ("Internet of Things"), which represented the beginning of a new era for technology. It represented the moment when the technology had the capacity to change fiction into reality through simple developments of automated systems, wirelessly interconnected and managed using only one mobile device, a better idea than domotics, which required a multitude of interconnected computers to manage related applications of domestic appliances.

Before describing the system and its functionalities I want to clarify the difference between

"home automation" and "smart home", because I noticed they are usually used as synonyms by general public and even by experts. The first concept was designed to integrate household appliances, relying solely on the use of subroutines in order to ease and improve people's lifestyles. The system I have implemented and I will present in the next sections is based on this concept, because it has no intelligence, the devices being automated according to my preferences using programming languages. "Smart home" was a term used for the first time in the mid-1980s during the boom of the domotic system and has its origins in "home automation", but it is a much wider concept that includes a large range of features, technologies and industries, all interconnected through IoT. [1] Therefore, we can say that "home automation" lies at the heart of the intelligent houses known today, a fully interconnected concept that has learning abilities. That being said, through artificial intelligence the system learns the behavior of the inhabitants and can take independent decisions. Nowadays systems have enhanced capabilities, allowing monitoring and control of home from a single mobile/web application, managing security and providing remote healthcare, or even reducing energy consumption as a result of optimization algorithms. Fiction has

been transposed as much as possible into reality, so today we are facing with talking houses that originally existed only in books or films and integrated technologies where everything is interconnected and the possibilities are endless.

2 Literature Review

The aim of this section is to establish a base of the existing literature, and to present different HAS with their features, benefits or limitations.

Shortcomings of existing technologies were mentioned in 2013 in a paper which introduced a convenient and flexible smart home system based on IoT. There was presented a demo version of the system which used a phone to query and control the system locally and remotely. The experimental results shown that the system could provide a real-time management for the home and it is considered that with the introduction of IoT the research and implementation of home automation are getting more popular. [2]

In 2014, another paper presented a flexible, low-cost automated home system, which is based on a mobile application for Android which communicates with the micro-web server from the Arduino Ethernet. The proposed design is based on the management control of energy systems such as lightings, security, heating, air conditioning, fire or intrusion detection using a siren and email notifications. [3]

In the same year, another relevant paper presented the necessary background regarding the IoT paradigm. They evaluated a subset of 50 important projects based on research and commercial solutions appeared over the last decade (2001-2011). A major objective of this paper was to help elderly/ handicapped people by controlling various home appliances and provide security using Android phone/tablet. So, the solution is designed for an Android phone upon a home automation system which uses Arduino Mega ADK and embedded devices/sensors. [4]

In April 2016, according to a study made by Comcast on how the notion of house automation is understood, with an emphasis on how

people want to use this solution, the result obtained has clearly demonstrated that the most of them want to benefit from the functionalities of a smart home without too much effort, leaving everything in the providers of such services' jobs. [5]

However, the existing works were mainly focused on switching and controlling home appliances or connected devices rather than remotely monitoring of home environment which is the focus of nowadays systems.

3 System Analysis

In my opinion home automation systems (HAS) face three main challenges: high costs, vulnerability, and difficulty in achieving security. The main purposes of this paper is to design and implement a HAS using IoT, capable of controlling and automating the next systems: lightening, cooling, gate locking and seismic warning, through an easy and convenient Android application. The proposed system has a great flexibility based on Wi-Fi technology used to interconnect and control a part of the modules (the relay, LEDs) and its distributed sensors (motion sensor) to HAS server, all being developed around Arduino Uno. The Android application is named "HomeControl", and it represents the key to an automated home. Besides remote control of devices and a lot of discussion when we have guests, this application was designed to ease the lives of those who use it, helping them to manage efficiently their time or costs, because it was developed from the idea of energy efficiency, aspects that are not neglected today at all.

When time becomes too short, the solution is to try avoid the routine or other things that does not need us physically, becoming others' duties. By automating various actions previously undertaken by humans, the possibility of time decrease becomes a real benefit. For example, if the living-room temperature exceeds a preset value, the user can choose to turn the air conditioning automatically on, and also set the operating time. Activating or deactivating sensors is also an important requirement as there may be times when we want a different behavior of the HAS. In case of an

evening party at the pool, users would certainly not want the lighting in the yard to be based on motion detection, but permanent. Gate control is another great advantage of the DIY home automation system I implemented. If one resident is upstairs and another one comes with luggage or has no keys, he/she can open him/her with just a click from the mobile device used. Another advantage of SI achieved is to be effective and prompt in getting a response on vibration monitoring that is designed to simulate a possible earthquake. Depending on the sensitivity chosen in the application and assuming the sensor is active, its role is to start the buzzer, which will notify the residents of the house that an earthquake is happening. Does it seem complicated? No, rather ideal.

4 System Design and Implementation

In this section I will detail the functionalities of the software component through which the control of the hardware components is achieved, together with the objectives defined in the previous paragraph and I will present the design methodology on the basis of which the information system was developed. I will also give an overview of the system by representing the activities undertaken by a user through a mobile device, using the CASE tool Visual Paradigm. The functional requirements underlying the development of the “HomeControl” application will be determined and modeled using the use case diagram (Figure 1). This illustrates the interaction between the system and the users, or other external components named actors, and the actions they want to accomplish.

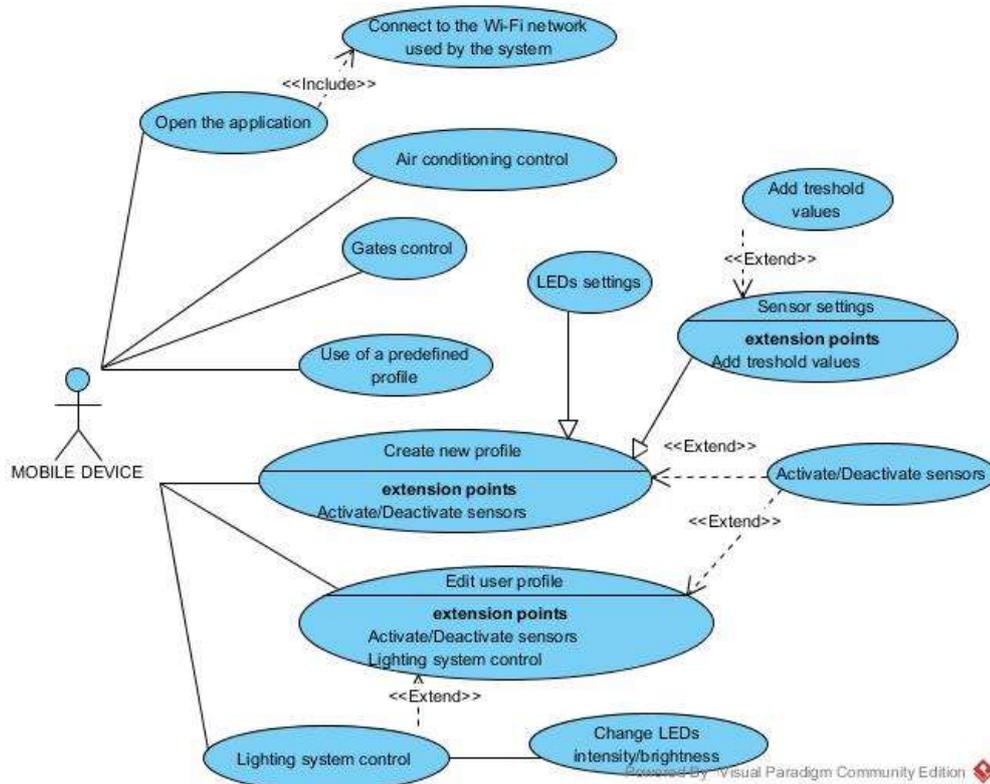


Fig. 1. Use Case Diagram – HomeControl Application

4.1 Proposed Home Automation System

The proposed model of the HAS is shown in

the Figure 3 and it consists of different modules.

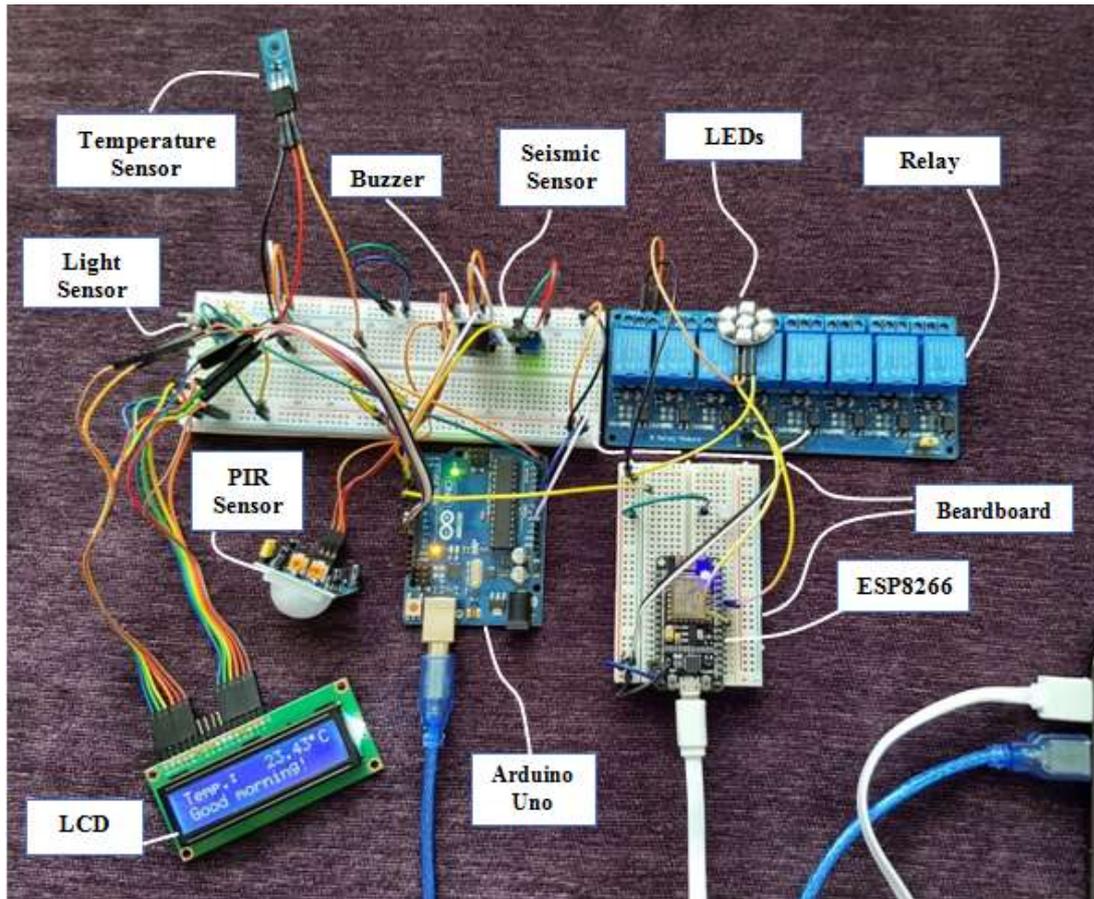


Fig. 3. Proposed HAS Setup

Some of them are based on sensors like: temperature, motion, light or seismic, through which is retrieved the data from the environment. The rest: Wi-Fi, LEDs, buzzer, relay and the stepper motor have independent functions that control and modify the system behavior. The picture provides an overview of the circuit, and the highlighted components will be described in the next section.

As a simple mention, because I will detail further how the system was created, two breadboards were used to multiply the number of pins belonging to Arduino Uno/ ESP8266 board, integrating the elements of the circuit as I wished. The main purpose of the system was to automate the following elements: lighting system (using LEDs), air conditioning (using a mini-propeller) or the gate (using a stepper motor) and to manage the environmental information received from the following sensors: motion, light, seismic or temperature. For controlling the system, I mentioned before that I developed a mobile application named

“HomeControl”, in order to connect exhaustively various devices belonging hypothetically to subsystems of a house, improving the life quality of the inhabitants.

4.2 Hardware Design

4.2.1 Arduino Uno

Arduino Uno is an open-source development board that can be resembled with a mini computer. It is built around the ATmega328P microprocessor which allows it to capture and interpret environmental data from a series of sensors and perform various actions on connected devices, such as displaying information obtained on an LCD screen, changing the LEDs colors and so on. The microprocessor is capable of running code very similar to C++, having some domain-specific libraries which add alternative names for some types and customizes functions. [6] The technical specifications of the board will be presented later in Table 1, along with the ESP8266 characteristics and the components are shown in Figure 4.

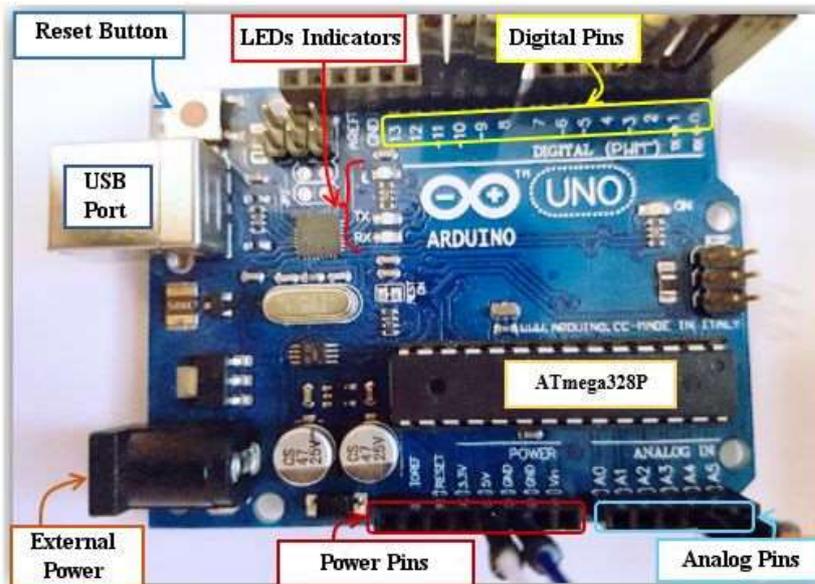


Fig. 4. Arduino Uno board

In the implemented system, the role of Arduino Uno is to control and integrate some of the modules used, hosting the code that manages their behavior. So I am talking about the integration of the following modules: motion sensor, temperature sensor, light sensor, seismic sensor, LCD screen and buzzer.

4.2.2 ESP8266 Wi-Fi Module

The ESP8266 Wi-Fi Module is a System on a

Chip (SOC) with integrated TCP/IP protocol that allows any other microcontroller to access the Wi-Fi network. This integrated circuit (Figure 5) is built on a microcontroller (not a microprocessor, because it does not have RAM, flash or I/O ports) and has advanced peripherals such as the ESP-12 Wi-Fi module. It can operate independently and perform actions better than Arduino Uno.

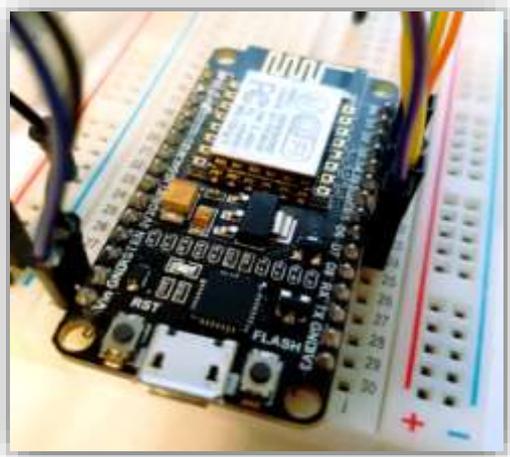


Fig. 5. WiFi Nodemcu module with ESP8266

Via the GPIO pins, the board can be integrated with specific sensors or devices, having enough processing and storage capacity. ESP8266 is capable of hosting and taking over

all wireless network functions from another application processor. [6] Its specifications are detailed in Table 1.

Table 1. Technical specifications for Arduino Uno and ESP8266

Source: [7] and [8]

Technical Specs/ Board	Arduino Uno R3	ESP8266-12E
Operating Voltage	5V	3.3V
Microcontroller	ATmega328P	ESP8266
Digital I/O pins	14 (6 PWM)	13 (9 PWM)
Analog Input Pins	6	1
Flash Memory	32 KB	1MB
SRAM	2 KB	64KB
EEPROM	1 KB	-
Clock Speed	16 MHz	80 Mhz
External power	Yes	No

The ESP8266 module has three modes of operation:

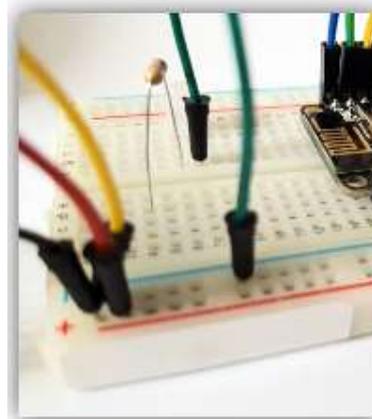
1. Access Point (AP) - The Wi-Fi module acts as a wireless network or access point (hence the name), allowing other devices to connect to it. A bidirectional communication is established between the module and the devices connected to it via Wi-Fi.
2. Station (STA) - The ESP8266 module can connect to an access point (AP), such as the Wi-Fi router used in the home. This allows any device connected at the same Wi-Fi network to communicate with the module remotely, building a local network.
3. Access point and station - in this mode, it acts both in AP and STA mode.

In the system I implemented, I used the Wi-Fi module in STA mode, to represent a control point between the objects connected to the network and the mobile device, also connected to the same network as ESP8266. This module allows the users to create an autonomous web server using various libraries by which the client (the mobile application) can efficiently control and manage the LEDs, light sensor, motion sensor and the stepper motor attached to the gate, receiving power through the 8-relay module. Thus, through the mobile application that connects to the HTTP server and sends requests, various responses are returned depending on the needs of each user.

4.2.3 The Breadboard

The breadboard (Figure 6) is a device that

connects the electronic components very easily and without soldering, using male to male wires (with pins at both ends).

**Fig. 6.** The breadboard

Its role in making the circuit is to allow the supply and connection of as many modules as possible, representing a bridge between components and a major power supplier.

4.2.4 The Relay Module

The 8 electromechanical relay module from the next picture (Figure 7) allows 8 different electrical devices to be controlled via 8 Arduino digital ports. [6]

Its role within the circuit is to turn on/off the power of the devices connected to it: the LEDs, the light and motions sensors or the stepper motor, using the information received from the Wi-Fi module.

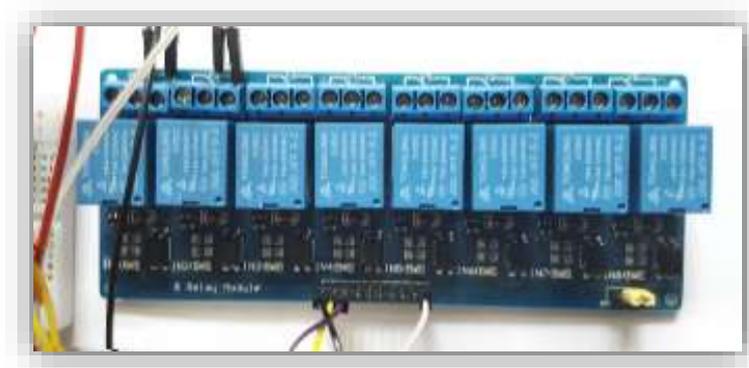


Fig. 7. The Relay Module

It plays a key role when it is desired to activate the “sleep” mode, because it stops power for some devices and reduces energy consumption.

4.2.5 The Stepper Motor Module

The stepper motor module shown in Figure 8 is used in the circuit to automate the gates from the front yard.



Fig. 8. Stepper Motor Module

Their control is done by a simple click, representing a great benefit for the homeowners, since it will be much easier for them to come into with their cars or if they have luggage and it is not at their fingertips to open the gates by themselves.

4.2.6 LCD Module

The LCD from Figure 9 has a 2-rows and 16-columns character matrix, which are used to display various information.

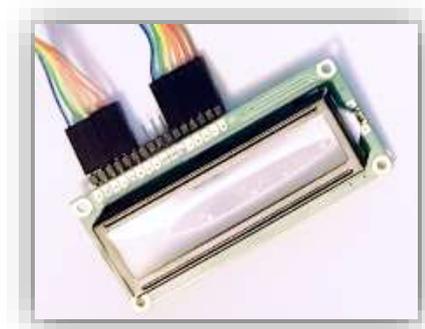


Fig. 9. The LCD

Its function within the circuit is to display temperature in real time, being a copy of a digital thermometer, and also to display air conditioning information.

4.2.7 Temperature Sensor

The temperature sensor (Figure 10) is a component that perceives environmental temperature changes and sends signals to the Arduino board.

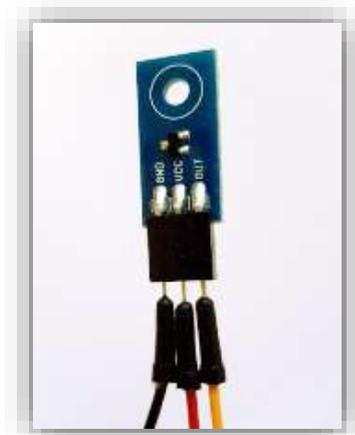


Fig. 10. Temperature Sensor

Within the system, its functionality is to start

the fan/air conditioning if the temperature exceeds the preset threshold measured with a potentiometer.

The temperature is displayed in real-time using an LCD and these three components: the temperature sensor, the potentiometer and the liquid crystal display, were being designed to simulate an automated thermostat.

4.2.8 Motion Sensor

The motion sensor from Figure 11 is based on IR (Infrared) technology and is used to detect the presence of a human/object at a distance up to 7m. [6]

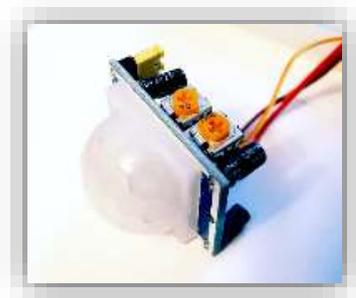


Fig. 11. PIR(Motion) Sensor

I chose to use it with the light sensor to control the automatic lighting in the garden. When it gets dark and a person arrive at home or leaves, the LEDs will light up, allowing him/her to lock/unlock quickly and easily.

4.2.9 Light Sensor

The light sensor shown in Figure 12 is a component which retrieves data from the environment and I used it to control the lighting system around the house. It detects intensity changes of light, and helps the users to achieve an optimal energy control along with the motion sensor.

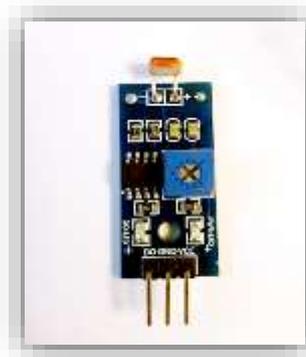


Fig. 12. Light Sensor

As long as its functionality is activated, the LEDs in the yard will be on only at night when motion is detected.

4.2.10 Seismic Sensor

The seismic sensor (Figure 13, left) is a component that senses mechanical vibration. In order to accomplish the project I decided to complete its functionality using a buzzer.

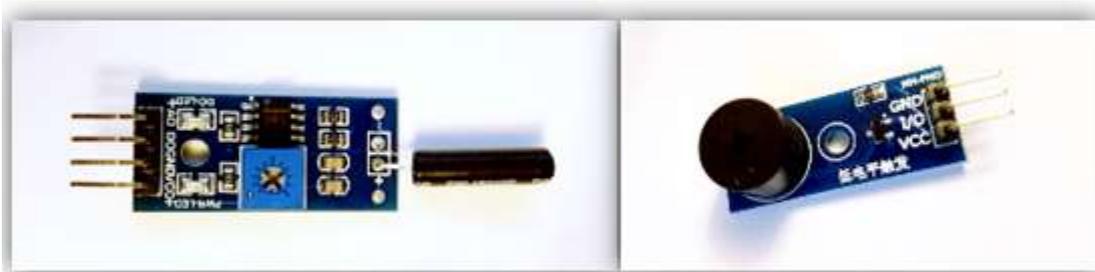


Fig. 13. Seismic Sensor (left) and buzzer (right)

When vibrations greater than a preset value are detected, illustrating the case of an earthquake, the buzzer will turn on an alarm to alert the users.

4.2.11 LEDs Module

For the lighting system control I used both RGB LEDs and NeoPixel LEDs (Figure 14) that allow creating different ambient light profiles.

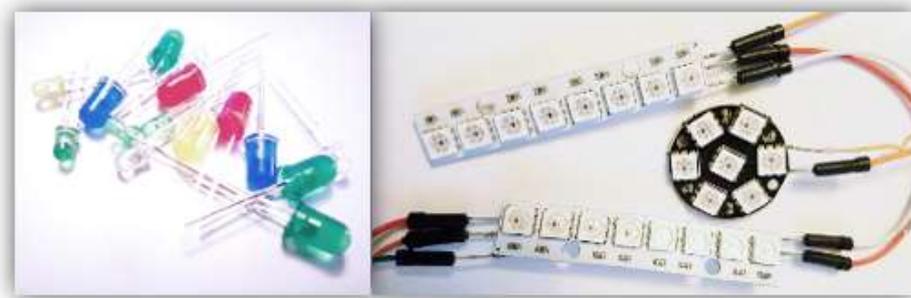


Fig. 14. RGB LEDs (a) and NeoPixel (b)

In the mobile application, the user has the ability to change the intensity of the lights, but also to choose a predefined environmental program such as “reading”, “disco” or “relaxing” which imitate the reality obtained for example with Philips Hue.

4.3 Software Design

The information system presented in this paper has been implemented by highlighting the

facilities offered by the Arduino and Android platforms, along with the utility of the SQLite SGBD. The interface between the user and the database is represented by the mobile application and was created using Android Studio IDE. Regarding the back-end part, the database control and management is made from the mobile application, while the elements that make up the circuit are controlled through the mobile application, Arduino and Wi-Fi board.

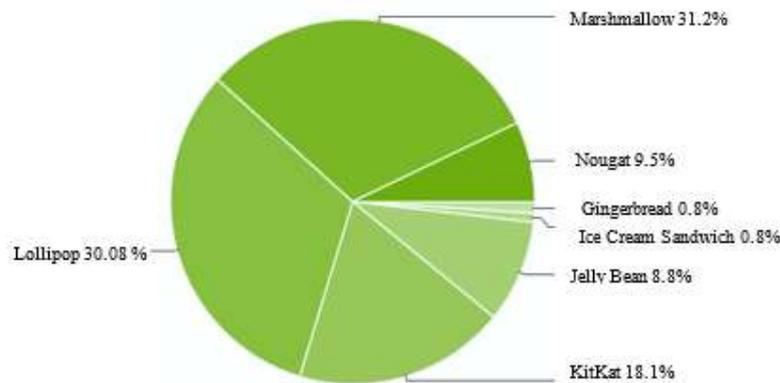


Fig. 15. Android Platform Versions, June 2017

Source:[9]

The application, “HomeControl” was developed on the Android OS using version 6.0 “Marshmallow”, which has associated level 23 of the programming interface (API). According to the next chart (Figure 15), this version is the most used at the moment, and because the device I own supports this version I chose it.

In the circumstances that it is possible to make an application compatible with one or more versions of the Android platform I chose as minimum SDK the API level 21, which corre-

sponds to version 5.0 “Lollipop”, thus covering around 63% of the devices which are currently using the Android OS. The development for controlling and managing the elements that make up the circuit meant programming the Arduino microcontroller and the Wi-Fi module through special rules for organizing written code in C ++. Thus, I created two programs (called sketches) one for each component mentioned before, using the official development environment Arduino IDE.

4.4 Setup Implementation

In this section I will describe how I connected a part of the hardware components to obtain the system, starting from Arduino Uno and ESP8266 boards, to some specific devices connected and controlled by them.

I will start with the Arduino board, which according to the technical specifications from Table 1, has 14 digital pins used to send only signals, 6 analog pins to receive/send data from/to the connected components and 5 power supply pins, all shown in Figure 4 from section 4. In the following presentations, the power supply pin (plus) is called Vcc, the minus is called Gnd, and the one used to receive/send signals or data Out or D₀. This board can be powered externally or directly from the computer via USB and the sensors are used to receive environmental information, on the basis of which I have decided what actions should happen.

Under these circumstances, to Arduino Uno I decided to connect modules that do not require changing parameters just thresholds, such as the light sensor, motion sensor, seismic sensor, LCD screen or the buzzer, and I am going to describe how they were connected to the board in the following paragraphs. Since Arduino has a limited number of power pins, I needed to use a breadboard to supply power to all the components and create an explicit circuit.

So, the generally connection used to power all the modules is noted in the paper as relation X and involved the next two steps:

1 I linked the Vcc (+) to the breadboard power line (+)

2 I linked the Gnd (-) to the breadboard ground line (-)

The concept of these two links mentioned above and represented in Figure 16 is also used to link all the modules to the breadboard, so I will start to detail how the light sensor connects.

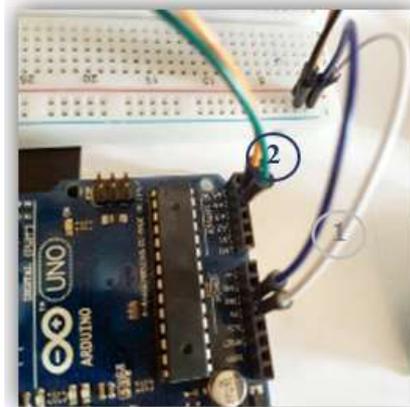


Fig. 16. Standard links between Arduino and breadboard

To get the information from the light sensor I connected the Out pin (D₀) to the third analog pin of Arduino **1**, and to supply power to the module I applied the relation X (links no **2** and **3**), using the breadboard. All these connections are shown in Figure 17.

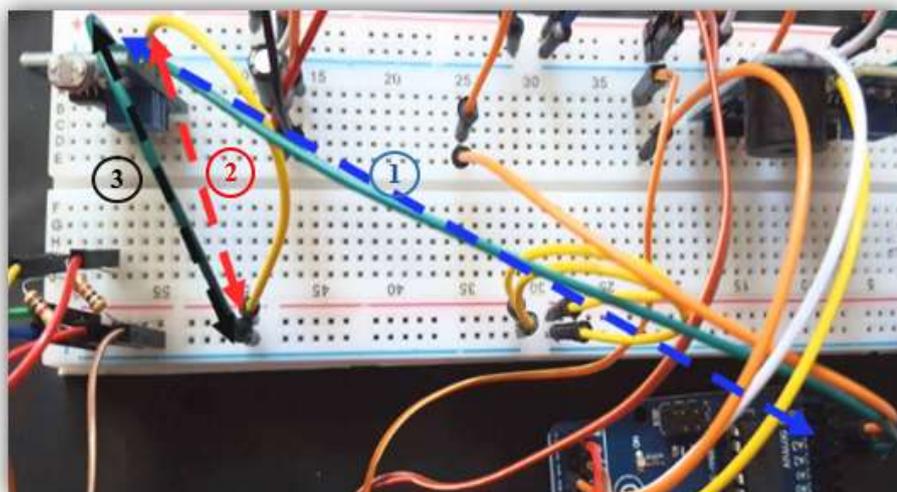


Fig. 17. Light Sensor Connection

The module is linked to an analog pin because values are read by the sensor and they need to be transferred to Arduino for further interpretation.

The next connected sensor is the temperature one. The way it receives power is identical to what I have previously presented for the breadboard and the light sensor (relation X: links no 2 and 3). To receive information from

the temperature sensor, I coupled its signal pin D_0 to the 4th analog pin from Arduino 1. Figure 18 also shows the links between the breadboard and the Arduino Uno.

As the light sensor, this one is connected to an analog pin too, because environmental values are read and transmitted to Arduino for interpretation.

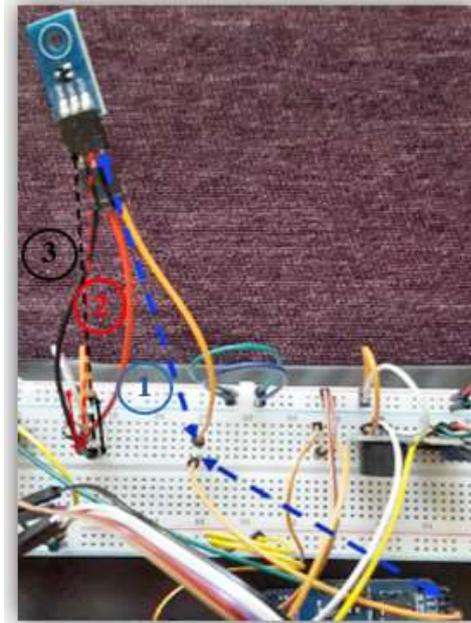


Fig. 18. Temperature Sensor Connection

The following two components: the seismic sensor and the buzzer will be presented in parallel. The way they receive power is the same as I have previously presented (relation X: links no 2 and 3). To get information from the seismic sensor I

linked the output pin D_0 to the 4th Arduino digital pin 1, and regarding the buzzer I linked the Out pin, D_0 , to the 3rd digital pin from Arduino 1[?]. The connections described can be seen in the figures shown below 19 (front) and 20 (back).

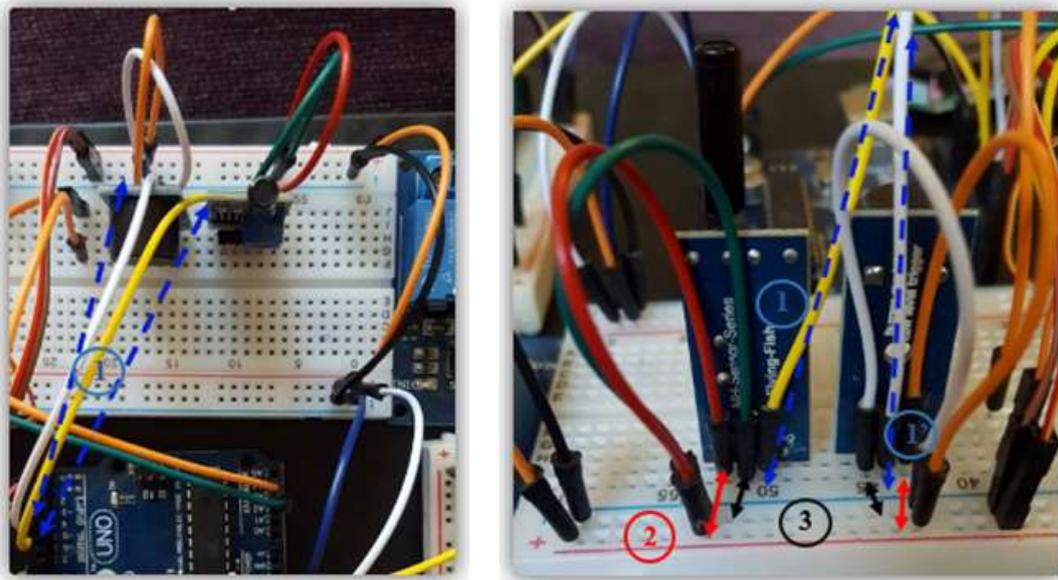


Fig. 19. Seismic Sensor and Buzzer Connections (back – left, front - right)

When the specialized sensor detects a vibration, signals will be sent to the board, and if this vibration exceeds the threshold preset in application, signals will be sent to the buzzer, which will produce a series of sounds such as an earthquake warning.

The last two components for which I described the way they are connected in the system are the 8-relay module and the LEDs module.

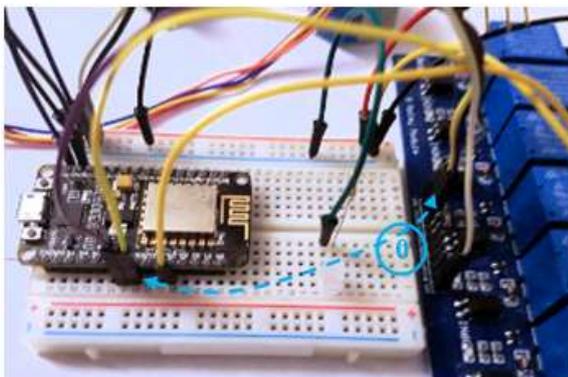


Fig. 20. Relay Module connected to ESP8266

They are connected and controlled by the

ESP8266. In order to benefit from the module's functionality, I linked its Vcc (+) pin to the breadboard plus the Gnd (-) pin to minus (relation X). In order to manage the power supply through the first relay of the module, I coupled the IN1 pin to the GPIO_14 pin of the ESP board 0, as shown in Figures 20 and 21.

Running one of the relays involves writing the LOW value on the GPIO pin of the Wi-Fi board with which it is paired. Thus, by pressing the existing button in the mobile application, used to control the lights, a request to the server will start the relay. Regarding the lights supply power, there will be written LOW on GPIO_14 to start the relay so that the LEDs are on. Otherwise, HIGH will be written on GPIO_14 to stop the relay and the power of the lights.

The last connection explained is for the NeoPixel LEDs. To connect this module in order to perform various light changes, I linked the elements as follows: the input of the first relay was connected to the breadboard power line (plus) 1 and its output to the Vcc (+) 1' pin of the LEDs, so setting on/off the relay will cause the lights to go on/off too.

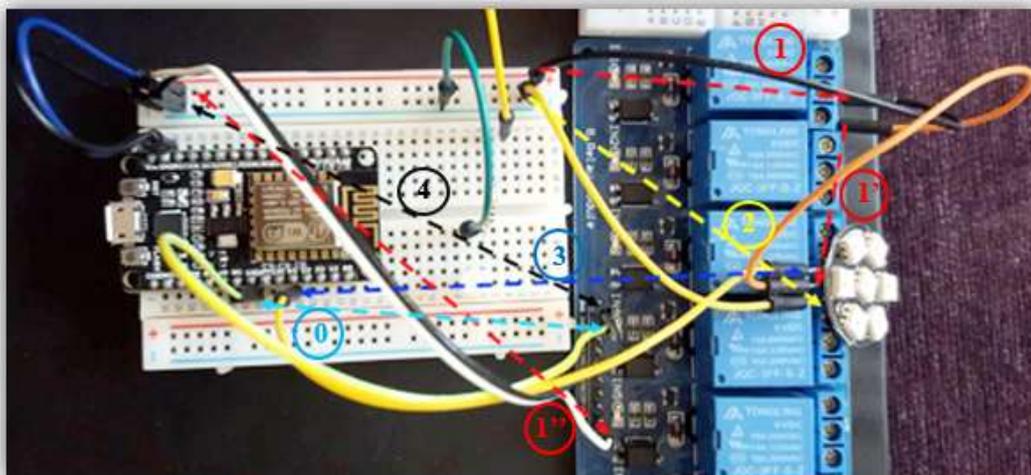


Fig. 21. NeoPixel LEDs connection

Gnd (-) of the LEDs is linked to breadboard minus (ground line) 2, and the IN pin to the GPIO_2 pin of the ESP board 3. Using this digital pin, information such as: light intensity or shades obtained by combining RGB base colors is transmitted.

The breadboard to which the 2 modules above are linked has the power line powered by the Vin pin of the ESP 1'' and the ground line from the Gnd Pin of the board. 4

5 Results and Findings

As I mentioned before, the mobile application I have developed has the purpose of controlling various components belonging hypothetically to subsystems from a dwelling, in order to improve the life quality of the users.

When the HomeControl application opens, the menu (Figure 22) appears and it allows the users to manage and control different devices in their house. The application runs in background a method by which every mobile device is identified and saved in the database according to its MAC address, and it is not necessary to create an account.

In the previous section I mentioned the automated components and their behavior and now I am going to present their functionalities through some images.



Fig. 22. Application menu – HomeControl

1) Temperature control - according to a preset value in the application the air conditioning will start automatically, but it can also be started manually pressing the 4th menu button.

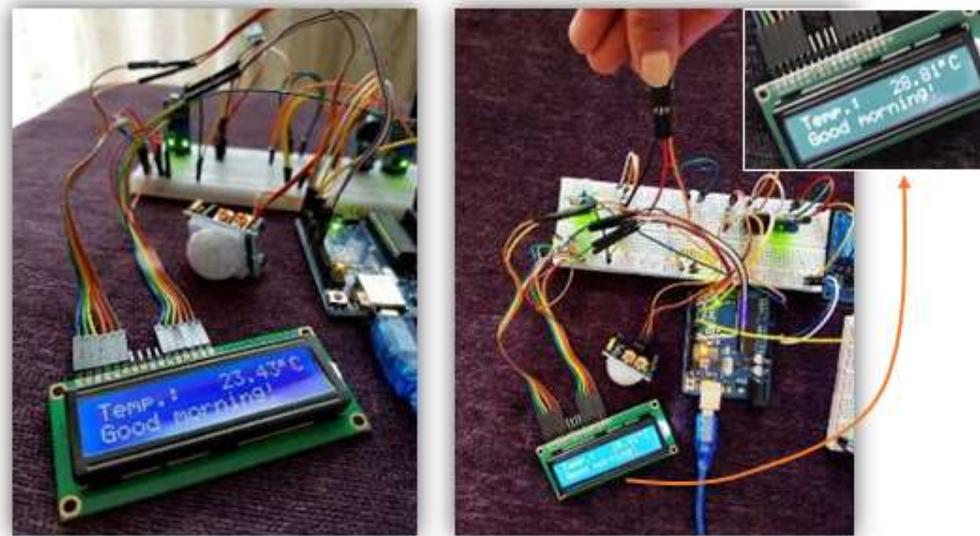


Fig. 23. Temperature Module – measurement display

In Figure 23 are presented the temperature changes happened at the intervention of external factors, increasing from 23.43 ° C to 28.81 ° C.

2) Control of the lighting system assures the management of the courtyard lamps based on the light and motion sensors, for which intensities will be set as threshold trigger and also

for the house lighting system, giving the user different possibilities regarding LEDs colors, such as Philips Hue bulbs. The courtyard LEDs and the others used only for monochrome light can be controlled manually using the first button from the menu as shown in Figure 24, left.

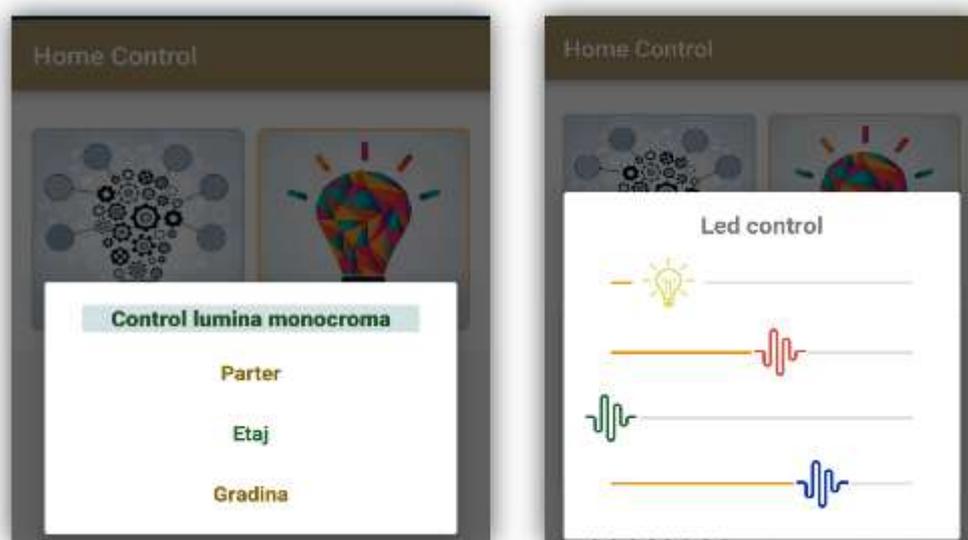


Fig. 24. Monochrome LEDs Control (left) and NeoPixel LEDs Control (right)

To create various light games from the application menu, the user can select the top right button that renders initially a monochrome light with a brightness of 10% (Figure 24, right), having the possibility to change the LEDs' colors in real time or to create a new lighting profile from Settings, the button

placed in the center of the menu.

In Figure 25 is presented the result of combining red, green and blue colors at different intensities.

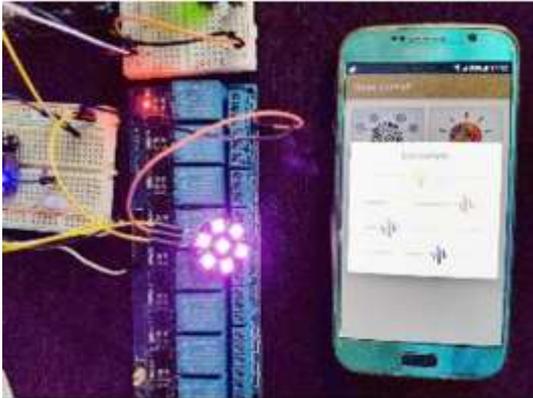


Fig. 25. NeoPixel LEDs Control

3) Earthquake warning is based on a seismic sensor that detects vibrations. When is exceeded a preset value, the alarm (buzzer) will be turned on to notify the inhabitants about the incident.

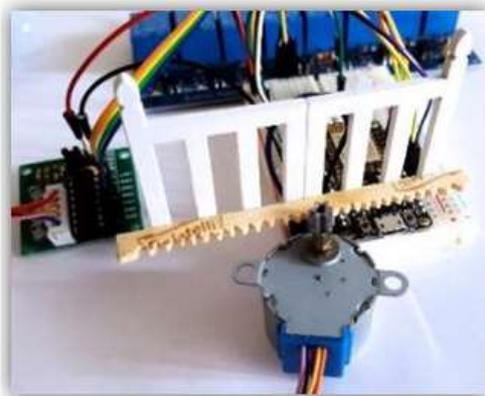


Fig. 26. Stepper Motor Control

4) Gate automation was realized using a stepper motor (Figure 26) and I think it represents a big advantage for those who own an automated house. For example, when they are in their cars or they have big luggage, only one click can represent the rescue.

6 Conclusion

From my point of view, the information system is in its start-up phase, which I would call it a first version. In this sense, I already have ideas on future developments and I started to document about the upgrade implementation. Therefore, in the future I'm going to move the local database on the server, to implement the voice control option and also to integrate multiple sensors into the circuit to increase the

system complexity and provide as many functionalities as possible.

I started from a great wish and a small scale implementation, but I chose not to stop just to that. Through the research for this paper I find out a lot of interesting things and I started to take small steps to automate my home and benefit from different functionalities like the ones of the system described. If everything spins around IoT, the choice to walk into an automated house is absolutely normal for me, and I think I would like my home much more than I do now because I am very excited when it comes about technology. The fact that I can control various household appliances through the latest technologies was an objective that I have proposed to achieve from half a year ago and now I almost reach it. Although such technology is quite complex at its core, I think that a flexible and user-friendly interface makes the products (automated systems) friendly but especially useful.

I am impressed by the evolution of technology and at the same time I feel lucky to have the opportunity to benefit from things that many decades ago were just fantasies written or illustrated in animations or movies.

References

- [1] M. Miller, *The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities Are Changing The World.* (pp 81-83). Indianapolis, IN: Que.
- [2] Y. Liu Y., *Study on Smart Home System Based on Internet of Things Technology.* In *Informatics and Management Science IV.* vol. 207, (pp. 73-81). W. Du, Ed: Springer London.
- [3] S. Kumar, *Ubiquitous Smart Home System Using Android Application.* In *International Journal of Computer Networks & Communications (IJCNC)* Vol.6, No.1. Retrieved April 04, 2017, from https://www.researchgate.net/publication/260127438_Ubiquitous_Smart_Home_System_Using_Android_Application
- [4] C. Perera, A. Zaslavsky, Christen P. & Georgakopoulos D. (2013), *Context Aware Computing for The Internet of*

- Things: A Survey. In IEEE Communications Surveys & Tutorial. Retrieved May 06, 2017 from <http://ieeexplore.ieee.org/document/6512846/>
- [5] The Safe & Smart Home: Security in the Smart Home Era. (n.d.). Retrieved April 12, 2017 from http://corporate.com-cast.com/images/August_Xfinity-Safe-and-Secure-Study-Report.pdf
- [6] Hardware components details. Retrieved March 22, 2017 from www.robofun.ro
- [7] Arduino Board Technical Specs. Retrieved April, 2017 from <https://www.arduino.cc/en/main/arduinoBoardUno>
- [8] ESP8266 Board Technical Specs. Retrieved April, 2017 from <https://en.wikipedia.org/wiki/NodeMCU>
- [9] Android Platform Versions. Retrieved June, 2017 from <https://developer.android.com/about/dashboards/index.html>



Alexandra MIHALACHE graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2015. She is currently pursuing his studies in Economic Informatics and this year she will obtain her master's degree at Bucharest University of Economic Studies. Her interest focuses on IoT and its applications in different fields.