

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculté des Sciences et de la technologie

Département d'Electronique

# Mémoire

Présenté pour obtenir

## LE DIPLOME DE MASTER

FILIERE : ELECTRONIQUE

Spécialité : Electronique Industrielles

Par

**BRAHIMI RAMZI SEYF EDDINE**

**LAHGUI FATAH**

*Intitulé*

*Conception et réalisation d'un contrôleur LED avancée.*

*Soutenu le : .....*

*Devant le Jury composé de :*

<i>Nom &amp; Prénom</i>	<i>Grade</i>	<i>Qualité</i>	<i>Etablissement</i>
<i>Dr. SIDAHMED S.</i>	<i>MAB</i>	<i>Président</i>	<i>Univ-BBA</i>
<i>Dr. BELHADAD Y.</i>	<i>MCB</i>	<i>Encadreur</i>	<i>Univ-BBA</i>
<i>Dr. BOUDECHICHE D.</i>	<i>MCB</i>	<i>Examineur</i>	<i>Univ-BBA</i>

*Année Universitaire 2021/2022*

### **Dedication:**

We thank ALLAH the almighty for giving us the patience and anything that helped us to carry out this modest work. We would like also to extend our deepest thanks to our fathers, mothers our families because they were our source of support throughout the university career.

A special thanks to our teacher before he was our supervisor **Dr. BELHADAD Yehya** with all his patience, his support and his encouragement, to carry out this work for this project.

A big thank to all of our classmates from the MCIL5 class.

## Summary:

In this work, a controller based on (NodeMCU) is designed for lighting devices (led and RGB led) control. A smart phone application is developed to send the controller commands via the local Wi-Fi network, the application written in Java is designed specifically to work with the controller. The application will send commands based on input from the user, the controller reacts upon receiving commands in the form of an HTTP request. A mini home model is built to showcase the project feasibility.

## Résumé :

Dans ce travail, un contrôleur basé sur (NodeMCU) est conçu pour le contrôle des dispositifs d'éclairage (led et led RVB). Une application pour téléphone intelligent est développée pour envoyer les commandes du contrôleur via le réseau Wi-Fi local, l'application écrite en Java est conçue spécifiquement pour fonctionner avec le contrôleur. L'application enverra des commandes basées sur l'entrée de l'utilisateur, le contrôleur réagit à la réception de commandes sous la forme d'une requête HTTP. Un mini modèle de maison est construit pour montrer la faisabilité du projet.

## ملخص :

في هذا العمل ، تم تصميم وحدة تحكم تعتمد على (NodeMCU) للتحكم في أجهزة الإضاءة (led و RGB led). تم تطوير تطبيق هاتف ذكي لإرسال أوامر وحدة التحكم عبر شبكة Wi-Fi المحلية ، وقد تم تصميم التطبيق المكتوب بلغة Java خصيصًا للعمل مع وحدة التحكم. سيرسل التطبيق أوامر بناءً على المدخلات من المستخدم ، وتتفاعل وحدة التحكم عند تلقي الأوامر في شكل طلب HTTP. تم بناء نموذج منزل صغير لعرض جدوى المشروع.

# Table of Contents:

Dedication: .....	2
Summary: .....	3
Table of Contents: .....	4
List of figures: .....	7
List of Tables:.....	8
List of Abbreviations:.....	8
General Introduction .....	9
Chapter 01: Introduction to smart homes .....	11
1. Introduction:.....	12
2. History: .....	12
3. Advantage /Disadvantage: .....	13
3.1. Advantages of Smart Homes: .....	13
3.1.1. Wireless Control: .....	13
3.1.2. energy and resource efficiency: .....	13
3.1.3. Safety and home security:.....	14
3.2. Disadvantages of Smart Homes: .....	14
3.2.1. Cost:.....	14
3.2.2. Internet use: .....	14
3.2.3. Privacy concerns:.....	14
3.2.4. Security issues: .....	15
3.2.5. Compatibility issues:.....	15
4. Common modules: .....	15
4.1. Lighting module:.....	15
4.2. Heating module:.....	15
4.3. Security & Safety module:.....	15
4.4. Weather module: .....	16
5. Smart home components:.....	16
5.1. The control network:.....	16
5.1.1. Wi-Fi: .....	16
5.1.2. ZigBee: .....	17
5.1.3. Bluetooth:.....	17
5.1.4. Infrared:.....	17
5.2. The controller: .....	18
5.3. Remote-control devices: .....	18
5.4. Sensors and actuators: .....	19
6. Conclusion: .....	19

Chapter 02: Smart wireless android led controller .....	20
1. Introduction :.....	21
2. Background idea:.....	21
3. System architecture: .....	21
4. System design:.....	22
4.1. Hardware:.....	22
4.1.1. Electronic design: .....	22
4.1.2. Controller Core: .....	23
4.1.2.1. Specification: .....	23
4.1.2.2. Pin Definition: .....	24
4.1.3. Controller Parts: .....	24
4.1.4. Controlled devices: .....	25
4.1.4.1. Led or Normal led:.....	25
4.1.4.2. RGB Led: .....	25
4.2. Software: .....	26
4.2.1. Application Interface design:.....	26
4.2.2. Application code organization:.....	27
4.2.2.1. Classes: .....	27
4.2.2.2. Resources: .....	27
4.2.2.3. Core code:.....	27
4.2.2.4. Core actions.....	28
4.2.3. The Controller Code: .....	29
4.2.3.1. HTTP request: .....	31
4.2.4. Library used: .....	32
4.2.4.1. Volley:.....	32
4.2.4.2. Holocolor: .....	32
4.2.4.3. SQLite Database: .....	32
4.2.4.4. Material design:.....	33
4.2.4.5. Esp8622.h: .....	33
4.2.5. Programs used in development: .....	33
5. Conclusion: .....	33
Chapter 03: Tests and results .....	34
1. Introduction:.....	35
2. Electric circuit: .....	35
2.1. Circuit of Normal led: .....	35
2.1.1. Component of circuit:.....	35
2.1.2. Function explanation:.....	35

2.2.	Circuit of RGB led: .....	36
2.2.1.	Component of circuit:.....	36
2.2.2.	Function explanation:.....	36
3.	Making Mini Home Modal:.....	36
4.	Implementation of the Project:.....	38
4.1.	Configuration of NodeMCU:.....	38
4.2.	Pin selection: .....	38
4.3.	Configuration of smart light application: .....	39
4.3.1.	Add new device: .....	39
4.3.2.	Edit or delete device:.....	41
5.	Using a relay: .....	42
6.	Testing & the result: .....	42
6.1.	Analyzing the result:.....	45
7.	Conclusion: .....	47
	Final Conclusion.....	48

## List of figures:

Figure 01: Smart home Features. ....	12
Figure 02: Wi-Fi module.....	16
Figure 03: ZigBee module.....	17
Figure 04: Bluetooth module.....	17
Figure 05: Infrared module.....	17
Figure 06: Schema of connection between NodeMCU, the application, the user.....	22
Figure 07: schema of the controller.....	23
Figure 08: NodeMCU modal.....	23
Figure 09: Pin Definition of NodeMCU.....	24
Figure 10: Normal LED. ....	25
Figure 11: RGB led with 4 pins.....	25
Figure 12: Interface Diagram of the application .....	26
Figure 13: Schema of the application work.....	28
Figure 14: Volley java code. ....	29
Figure 15: Normal Light Java class.....	29
Figure 16: RGB Light Java class.....	29
Figure 17: Schema of NodeMCU Processing request.....	31
Figure 18: Arduino IDE code Example.....	32
Figure 19: Android studio Icon. ....	33
Figure 20: Circuit of NodeMCU wired with Normal led. ....	35
Figure 21: Circuit of NodeMCU wired with RGB led.....	36
Figure 22: Mini home modal design. ....	37
Figure 23: Real photo of mini home modal Top View angle.....	37
Figure 24: Real photo of mini home modal Front View angle.....	37
Figure 25: NodeMCU connected to pc.....	38
Figure 26: Example of pin selecting of NodeMCU\.....	39
Figure 27: Application screen shot of add normal device activity. ....	39
Figure 28: Application screen shot of add RGB device activity. ....	39
Figure 29: Application screen shot of Rooms section.....	40
Figure 30: Application screenshot of normal light control and RGB light control.....	41
Figure 31: A- example of edit device screen, B-Device section. ....	42
Figure 32: Device 01 in Bedroom turn On.....	44
Figure 33: Device 01 in Office room turn On. ....	44
Figure 34: Device 01 in Bedroom change color from red to blue green. ....	44
Figure 35: All mini home devices is turn On. ....	45
Figure 36: All mini home devices is turn Off.....	45
Figure 37: All devices turns OFF.....	45
Figure 38: Two RGB led Turns on.....	46
Figure 39: Two RGB led Turns on with different color.....	46
Figure 40: All devices is on.....	46
Figure 41: Office room off and Bedroom on.....	46

## List of Tables:

Table 01: Electronic component used with controller.....	25
Table 02:Table of Pin configuration.....	30

## List of Abbreviations:

<b>MCU</b>	Microcontroller unit.
<b>GND</b>	Ground pin.
<b>RGB</b>	Red, Green, Blue.
<b>LED</b>	Light-emitting Diode.
<b>HTTP</b>	Hypertext Transfer Protocol.
<b>GPIO</b>	General-purpose input/output.
<b>SSID</b>	Service Set Identifier.



## **General Introduction**

Smart home technologies are on the rise despite the cost associated with smart home equipment, smart homes allow electronic devices to communicate with each other through networks (whether it is Wi-Fi or any other network), in order to facilitate people's daily tasks while reducing resource usage and providing them with valuable information while saving valuable time. The ultimate goal of smart homes is to not only aid the user to reach a sophisticated level of comfort but also reduce resource consumption and raises the standard of living by facilitating Speeding up routines and preserving the environment are some of the areas that help us in our daily lives.

Smart home systems allow the user to use easily accessible devices (such as smartphones, tablets laptop or other smart devices and interfaces) to control the house features. All smart home systems on the market are multifunctional systems and therefore their prices are high and they are complex and some are not even easily adjustable, especially by ordinary people.

Smart homes can be divided into sub-modules for the control network, the focus of this work is on the lighting system. In this thesis, a controller is designed to control the lighting of a home using a smartphone to send remote commands to a NodeMCU controller connected to local WI-FI network. This thesis is organized as follows:

The first chapter presents the functionalities, equipment and components used in smart homes. Advantages and disadvantages are presented.

The second chapter presents the design of the controller (hardware) and the application (Android application). Details about the application code are provided.

The third chapter presents the design of a mini home model to showcase the use of the controller along with the application to control the lighting inside the home model.

**Chapter 01: Introduction  
to smart homes**

## 1. Introduction:

A Smart home allows us to connect all different home electronic devices with each other giving them harmonic interaction with the user [1], in this regards, everyday house duties related to these electronic devices could be automated without user intervention (the ultimate goal of smart homes). Sometimes, such duties could get controlled through remote control to allow the user to have an easier, convenient, efficient, safer, and less expensive way to get daily tasks done. To improve users' lives by reducing tedious and repetitive actions, accelerating complex processes, helping them enjoying a new experience using technology.

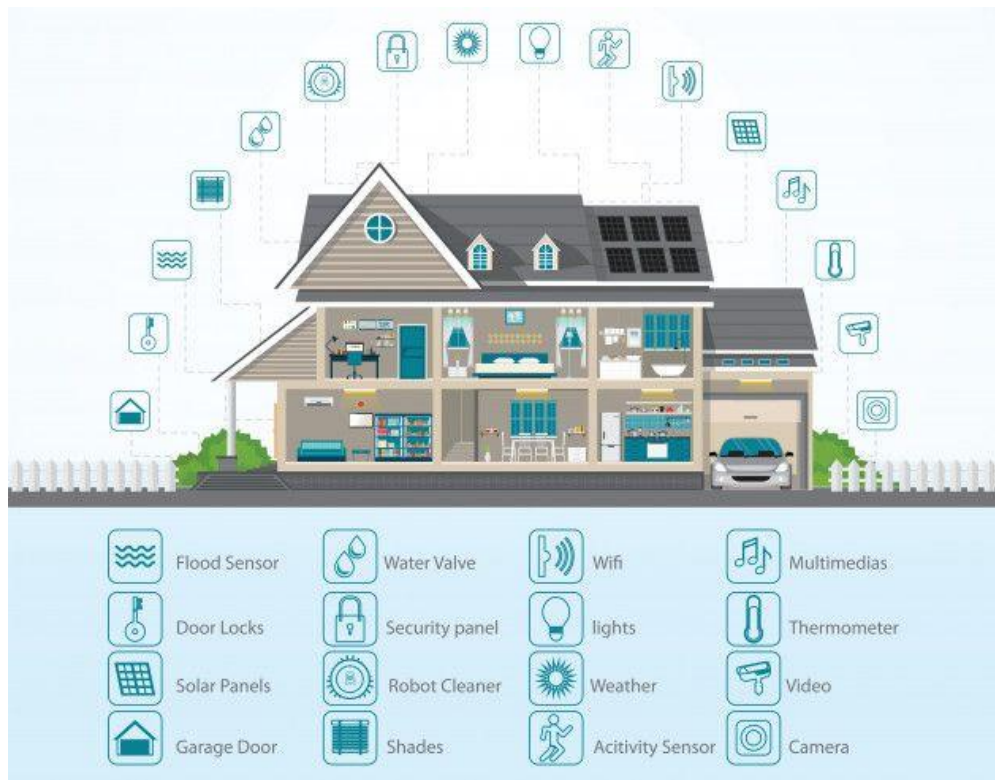


Figure 01: Smart home Features.

## 2. History:

Smart home is a modern term that came up through imaginary Sci-fi ideas and later became a real aspect in our life, and possibly the only thing viable in the future, in the following we show the progress of the term smart home through time.

**In the past**, the term “smart home” began to appear officially in 1984 through the Home Builders Association [2], and this despite the existence of previous projects of ideas similar the smart homes. However, due to the lack of appropriate technology including power consumption feasibility, computing capacity and also how smart devices can be at that time. The concept of smart homes has always stayed

almost the same: devices that are controllable, easy to use, and eliminate unnecessary and repetitive house chores.

**In the present**, the advances in technology allowed the spread of the idea of smart home systems with the provision of an unlimited number of smart devices, controllers and remote-control devices of various prices, shapes and features. However, it is still spreading at a slow rate for what was expected of it due to the lack of a general use devices and also the cost associated with the smart home.

**In the future**, replacing ordinary homes with smart home systems, improving the level of control, increasing the advantages and reducing the negatives and obstacles that have not allowed their spread so far. It may create the possibility of creating smart cities and even smart countries by exploiting technology and progress in it.

### **3. Advantage /Disadvantage:**

Smart homes can have useful features that a home user need. And it depends on several considerations:

#### **3.1. Advantages of Smart Homes:**

##### *3.1.1. Wireless Control:*

One of the best advantages of a smart home is to control several devices from a distance, this was only possible after the invention of wireless networks. User now can use a central hub to send several commands through the use of smartphones or even smart speakers that now can even allow voice control over the network.

Some commands may include turning some smart lights on, change what's playing on the television, or even in some advanced scenarios opening the garage door [3]. All this, without the need to stop what the user from what he is doing or even moving to accomplish those tasks reducing the time consumed by these tasks even further.

##### *3.1.2. energy and resource efficiency:*

Smart home devices can be used to help improve the efficiency of your home, saving both costs and helping the environment. One long dreamed about feature is the economy in resources using artificial intelligence to use only the optimum amount of each resource to the lowest possible reducing both the costs of home usage and the effects of waste resources on the planet and environment.

### ***3.1.3. Safety and home security:***

One major advantage of Smart homes is to improve safety for its users, some features may include connected alarms that can alert users via phone applications even remotely through the use of internet connected devices. A smart home should allow its users to watch live footage of doorbells feed for example to add an extra layer of security to their front door for their kids and even from stranger peoples [3].

## **3.2. Disadvantages of Smart Homes:**

On the other hand, smart homes also get a list of disadvantages, such as privacy concerns, compatibility issues of devices which limit the choices over the devices that can be connected, different network protocols and expensive smart homes equipment.

### ***3.2.1. Cost:***

The high cost of smart home systems is due to several reasons, including both the installation and the maintenance. Adding to that, smart equipment that are ready to be connected with a smart home system can cost as much as twice a regular device or in some case even more. Another feature that tends to raise the cost of maintenance is security systems that some use services such as cloud storage which may require ongoing subscriptions to access all of the features.

### ***3.2.2. Internet use:***

The dependency on an active internet connection is another issue with smart homes, this is due to the fact that remote commands are usually sent to the cloud, where they are processed before being sent to the appropriate devices or applications. On another hand, the use of internet in some cases may put the privacy of users on compromise hence the next disadvantage.

### ***3.2.3. Privacy concerns:***

One of the major downsides of spreading technology is privacy which became a major consideration lately. This does not spare smart devices nor smart homes. The way user entrusts details about daily habits, location details, buying habits and even more to current technologies does not only open the way to malicious code to send unwanted private data to third parties. Adding to this, some user's preferences about their private details that they are willing to share with both current technology and future ones closes the door in front of smart homes to a huge market share. Hence the slow spreading of this technology.

### ***3.2.4. Security issues:***

Smart home devices can be hacked, this can provide malicious persons access to people's devices. These unwanted accesses can be used in different ways to harm the home users, this can include both spying, theft, kidnapping and all kind of stuff that can happen to a compromised home security system.

### ***3.2.5. Compatibility issues:***

Smart homes rely on equipment/gadgets communicating with one another to provide users with valuable features and functionalities. While device interoperability is improving, there is still a major drawback to smart homes right now that is compatibility both in term between equipment and systems and sometimes between equipment and equipment.

The lack of standards led The ZigBee Alliance for example to put together a number of manufacturers, including Samsung, Philips Hue, Huawei, Tuya, Somfy, and others, to produce open international standards. However, other brands, such as Apple's Home Kit, make it extremely difficult to incorporate gadgets from other manufacturers [3].

## **4. Common modules:**

### **4.1. Lighting module:**

Lighting module enables the user to make change to the light setting in his home, settings like color change, powering on/off even remotely or change light direction using smart phones or speakers.

### **4.2. Heating module:**

The heating module allows the user to change and adjust both manually and automatically the temperature of a room. The change can be adjusted according to sensors readings (single or multiple), The heating module can also turn on fans or air conditioner if the temperature readings are high or turn on central heating (or single heating equipment) if the temperature readings are low.

### **4.3. Security & Safety module:**

The security & safety system is a basic and necessary part of a smart home to protect users from theft or others breaking into your home. Anti-theft functions through the use of just presence sensors and sirens that can alert users. Other components include gas, fire sensors ...etc.

#### **4.4. Weather module:**

Within the weather module, sensors that are able to measure data such as rain intensity, humidity and wind speed, the data from such sensors help the system execute commands related to weather. Such as in the case a user not at home [4], and suddenly the weather change and it starts to rain, the system can close the windows with prestored commands.

### **5. Smart home components:**

A smart home system is one that makes use of modern technology in order to work such as internet and modern electronic, smart home consists mainly of five building blocks [5]:

#### **5.1. The control network:**

In order to have a smart home a control network is needed in order to connect the controlled devices, sensors and actuators to the controller device. There are three main technology options for home control networks today [5]:

- Power line Communication.
- Basic Wireless Transmission.
- Advanced Wireline Transmission.

The type of network varies with the price of components and the cost of installation. Wire lines are found in the distinguished residential sector and industrial buildings, while the rest are found in residential homes and ordinary neighborhoods.

The following are the options in wireless communication that can be used in smart homes systems:

##### *5.1.1. Wi-Fi:*



*Figure 02: Wi-Fi module*



## Chapter 01: Introduction to smart homes

the most used wireless communication in modern life is the WI-FI technology, which requires a minimum of two peripherals or two WIFI devices to connect.

### 5.1.2. ZigBee:



Figure 03: ZigBee module

It is short distance and simple protocol inspired by Bluetooth technology in the 90s. It is low cost and the transmission power is adjustable.

### 5.1.3. Bluetooth:



Figure 04: Bluetooth module.

The Bluetooth technology was invented in 1994 by Ericsson. is economical with low price and it has a wider range of data transmission compared to infrared but it still less advanced than Wi-Fi.

### 5.1.4. Infrared:

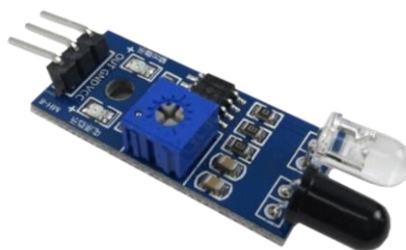


Figure 05: Infrared module.

Infrared radiation is electromagnetic radiation with longer wavelengths than visible light. Infrared rays only connect two devices with no obstacles in the path of light that is not visible to humans,

this renders the infrared not very useful for remote control over the internet and only used by physical in place classical remotes.

## **5.2. The controller:**

A smart home controller is a computer system and the core for the advanced features of a smart home [6]. Its role ranges from collecting information using sensors, reacting to commands from remote controls or direct commands from the user through applications or other devices to the automation of such commands based on input conditions. It can also get advanced artificial intelligence pre-programmed commands based on sensor reading or even by studying the user behavior.

The most prominent remote-control devices in today are smartphones or tablets, this makes it possible to even pass voice commands without the need to fancy equipment.

There are two types of controllers that are used:

### **- Microcontroller Based:**

Smart home microcontroller-based controller can use one of the following controller devices: Arduino, raspberry pi. esp. ...etc. and various other types of programmable devices. This type of controllers does not allow for the implementation of an advanced functionalities that requires more computation such as machine learning based functionalities.

### **- Smart hubs:**

Smart home hubs are typically designed specifically for home control, the cost for such hubs in on the high end however, multiple advanced functionalities can be embedded into these whether by using advanced programming or by using third party software plugins. The most used hubs today include: Siri, HomeKit, Alexa .... etc.

## **5.3. Remote-control devices:**

The role of remote-control devices is to connect the user to the smart home controller, the connection makes use of the control network such as described previously. In some advanced scenarios where the remotes are in a form of a smart phone application [7], the functionalities can (if available in the system) be activated remotely through internet.

### **- Custom Remote:**

They are remote control devices that are specially adapted for a specific type of control device and that use their own technology and components.

- **Smart phones & tablet:**

Smartphones and tablet can be used to control other devices using a custom application that allows the user to send commands to the controller.

**5.4. Sensors and actuators:**

A key element for monitoring and acting in a smart home are the sensors and actuators. Sensors on one hand allows the controller to read the state of some variables (such as temperature, light intensity) and react upon reading. On another hand, actuators allow a controller to accomplish mechanical action that usually need human intervention such as opening and closing doors, garage, movable ceilings. An example of such actuators includes pumps, electric motors.

**6. Conclusion:**

In this chapter we talked about smart homes and its features, functionalities and components we mentioned advantages and disadvantages, in the following chapter we focus more on the scope of our thesis which is the lighting module in a smart home and provide advanced design details of our contribution.

**Chapter 02: Smart wireless  
android led controller**

## **1. Introduction :**

Smart homes are an expected development, or rather, an evolution that must occur, for a variety of reasons, such as the usage of smart gadgets from televisions and phones to refrigeration and air conditioning systems, and so on. In homes, the smart home is typically used to make life easier for humans, but it is a massive and complex project that takes a lot of resources, especially because most homes now include a variety of devices.

We picked the lighting system for this project to transform it from traditional to smart. This is because the system is widely used and straightforward, but there may be times when we want to switch off the lights from a distance or install a light with multiple colors that we can change depending on our mood. This is exactly what we'll achieve in this project: we'll create a lighting system which can be remotely controlled using a smartphone with cheapest price [8].

## **2. Background idea:**

The purpose of this work is to provide a home lighting system with a smart phone wireless controller that allows the change of the lighting configuration remotely. Our system comprises of the following components:

- **The control network:** Wi-Fi network.
- **The controller:** NodeMCU and other electrical components.
- **Controlled devices:** Normal LED & RGB LED.
- **The remote-control:** Smartphones.

## **3. System architecture:**

An android application allows us to control the core of our controller which is a NodeMCU using the Wi-Fi network by sending special HTTP commands, these commands are read by controller and information from the command is extracted that allows the controller to act upon.

Before the user can use the system, he has to install the lighting system in the house and connect the light devices to the proper pins on the control board. The user then opens the android application from a smartphone. In the first time the user uses the application, he needs to add devices and rooms according to his setup.

Based on the device type, whenever a user clicks on a device in the application, he can then change the parameters and configure the lighting device accordingly. The following diagram illustrates the system architecture further:

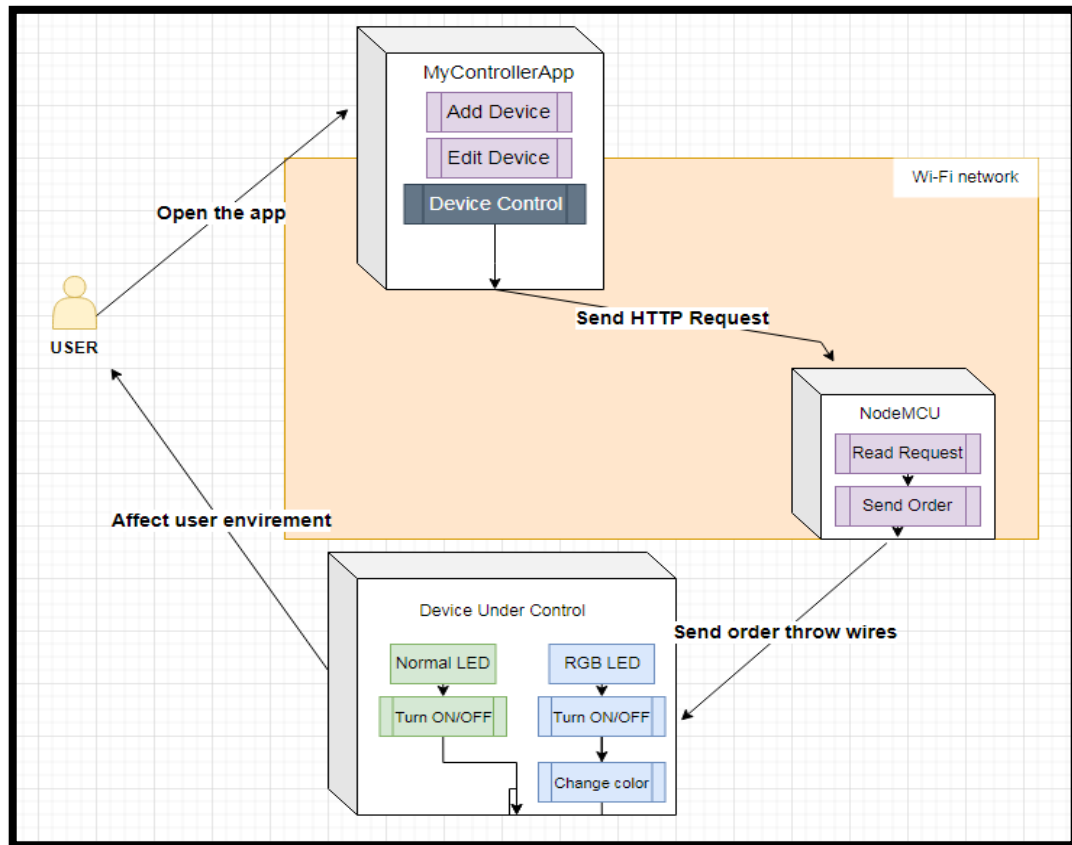


Figure 06: Schema of connection between NodeMCU, the application, the user.

The application then sends the control command to the controller to base on the user input.

## 4. System design:

### 4.1. Hardware:

This part consists of many electronic components wired together to form the electronic circuitry needed by our system. For the sake of this work, the lighting devices used are simple 5 volts led lights. In the real-life scenario higher voltage lighting devices needs to be connected through the use of relays based on the device type.

#### 4.1.1. Electronic design:

The controller (NodeMCU) connects with a resistor of (220 Ohms) through the pins D0 to D8 these pins will connect to led devices, J1 to J9 are female pins that connect to the male wire of the led, J10 to J18 will connect to the GND pin.

J19 to J21 is connected to 3V pin are needed in the case of RGB Led common Anode (common cathode has GND pins). The controller requires a 5V power source to be connected to the pin Vin.

## Chapter 02: Smart wireless android led controller

The following schema shows the pins architecture of the controller and the linked components.

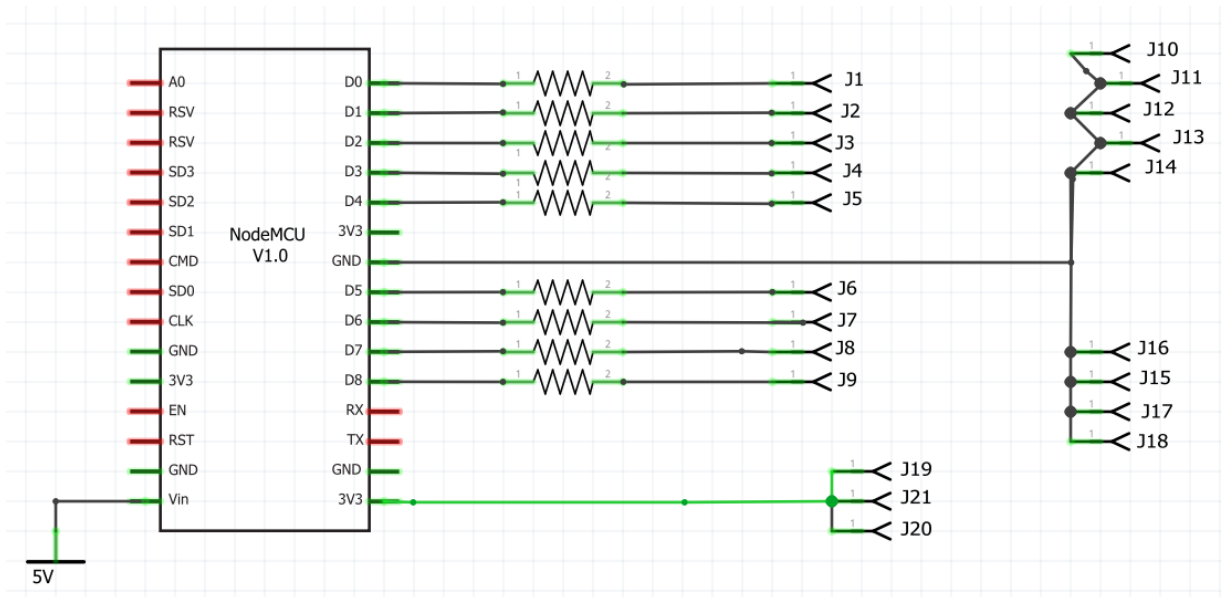


Figure 07: schema of the controller.

### 4.1.2. Controller Core:

The **ESP8266** is the name of a micro controller designed by Espressif Systems [9]. It can run applications based on itself or associate it with other projects and circuits that need a Wi-Fi network as a control factor.



Figure 08: NodeMCU modal

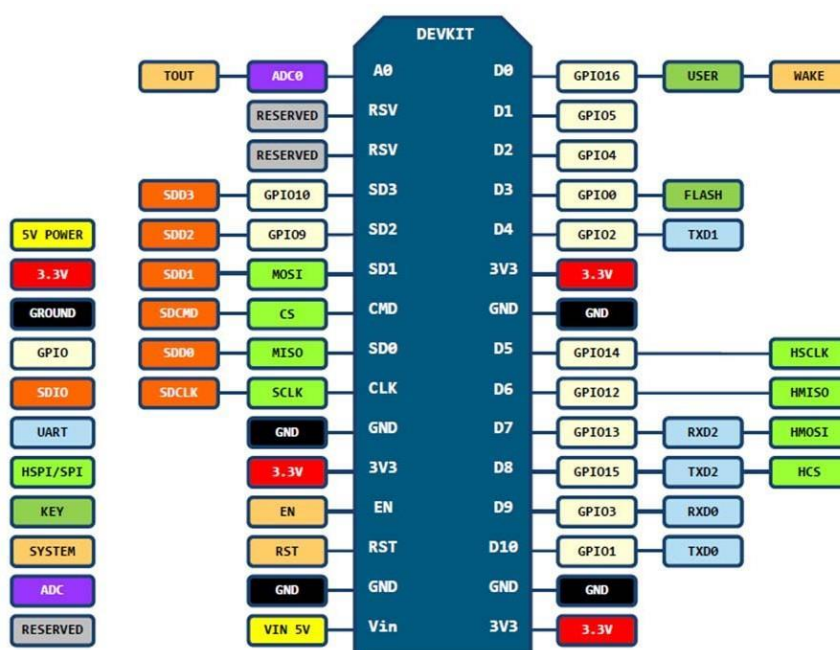
#### 4.1.2.1. Specification:

- Voltage:3.3V.
- Wi-Fi Direct (P2P), soft-AP.

## Chapter 02: Smart wireless android led controller

- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5.

### 4.1.2.2. Pin Definition:



D0(GPI016) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.




Figure 09: Pin Definition of NodeMCU

### 4.1.3. Controller Parts:

The following table shows the electrical components that are used in this project:



## Chapter 02: Smart wireless android led controller

Component	Resistance	Relay	Jump wire
Value	220 ohms	(1,2,4,8) channel (5v,12v,24v)	/
Photo			

*Table 01: Electronic component used with controller.*

### 4.1.4. Controlled devices:

#### 4.1.4.1. Led or Normal led:

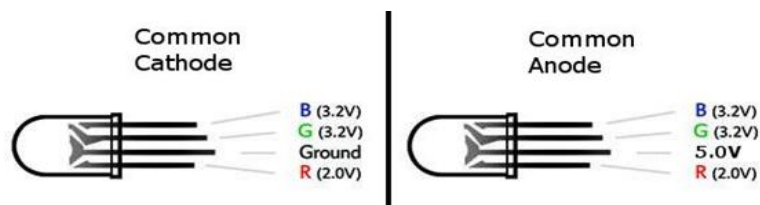
LED is a semiconductor device that emits light when an electric current flow through it. The intensity of the lighting can be controlled by changing the flow of electric current and it is available in many different colors and shapes.



*Figure 10: Normal LED.*

#### 4.1.4.2. RGB Led:

RGB LEDs is mix of 3 colors led inside a single LED package and have 4 pins 3 for the red, green, blue and last one for GND if it is common cathode or 5V through a resistor for common anode shown in below.



*Figure 11: RGB led with 4 pins.*

### 5mm RGB LED Specifications:

- 5mm diameter.
- Red: 630 nm wavelength, Green: 525 nm, Blue: 430 nm.
- Red: 2.1-2.5V Forward Voltage, at 20mA current, Green: 3.8-4.5V, Blue: 3.8-4.5V.

## Chapter 02: Smart wireless android led controller

- Red: 500 mcd typical brightness, Green: 600 mcd, Blue: 300 mcd.
- 5mm RGB LED pin connections.

There are other types of RGB led like: **SMD RGB 5050 LED** or **RGB LED Strip**.

### 4.2. Software:

In this section, the design of the software part of our system is presented.

#### 4.2.1. Application Interface design:

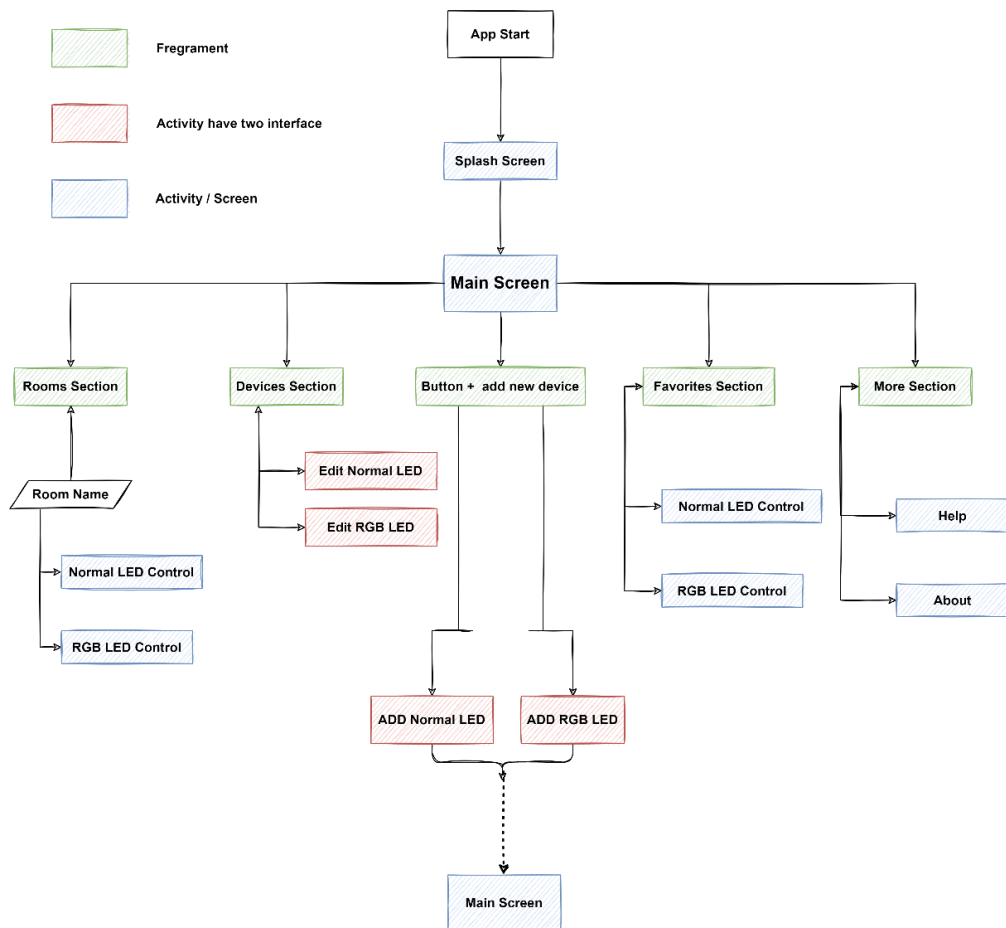


Figure 12: Interface Diagram of the application

In the above figure, the screen interfaces that user will interact with are shown. The main screen is a user interface that users will see right after opening the application and allows him to select any other section for example:

- In Rooms section user can found all rooms he adds and it is device correspondent.
- In Device section user can found all devices list and edit device info or remove it.
- In Favorites section user can found all devices that he adds to favorite list in device information.

## Chapter 02: Smart wireless android led controller

### 4.2.2. Application code organization:

The android application code written in Java consists of two main parts: a part for Java Classes and the other part for resources.

#### 4.2.2.1. Classes:

In the classes part we can find the background code that allows the application to perform the tasks needed.

#### 4.2.2.2. Resources:

In the resources:

- **Drawable:** all icons and images used in the application.
- **Layout:** all screen layout (interface) code of the application.
- **Mipmap:** icon and background for the application in the smart phone.
- **Values:** all useful string & color & theme code needed for the application interface

and there two more important files **AndroidManifest.xml** & **build.gradle**.

#### 4.2.2.3. Core code:

The core java code used in our application is organized as follows:

- **Activities:** in this folder we find the code that allows the user to interact with the interface and is organized as follows.
  1. *Main activity:* this activity contains the code for the main screen interface.
  2. *Add/Edit Normal Light Activity:* this activity contains the code that allows the user to interact to add normal light device information (device name, device IP, device pin, device room) or edit/delete devices.
  3. *Add/Edit RGB Light Activity:* this activity contains the code that allows the user to interact to add RGB light device information (device name, device IP, device red pin, device green pin, device blue pin, device room) or edit/delete devices.
  4. *Help & about activity:* this activity user can see some tips how to use the application and the application information.
- **Adapters:** in this folder we have two files needed to show the devices as a list in the interface.

## Chapter 02: Smart wireless android led controller

- **Controls:** two files for the control screen **Simple Light Control & RGB Light Control** that analyses the user input and send the proper command to the controller.
- **Database:** in this folder we have two files necessary for the application to save device information in an SQLite Database and load it when needed.
- **Fragments:** here we have four files, one for the Rooms section where users can see the rooms list and its devices. Enter the control screen for the device, another for Devices section where user can change device information according to its type.
- **Utilis:** Java classes that the application needs to work, Normal Light class, RGB light class these classes allow the application to form a command based on the type of light selected.

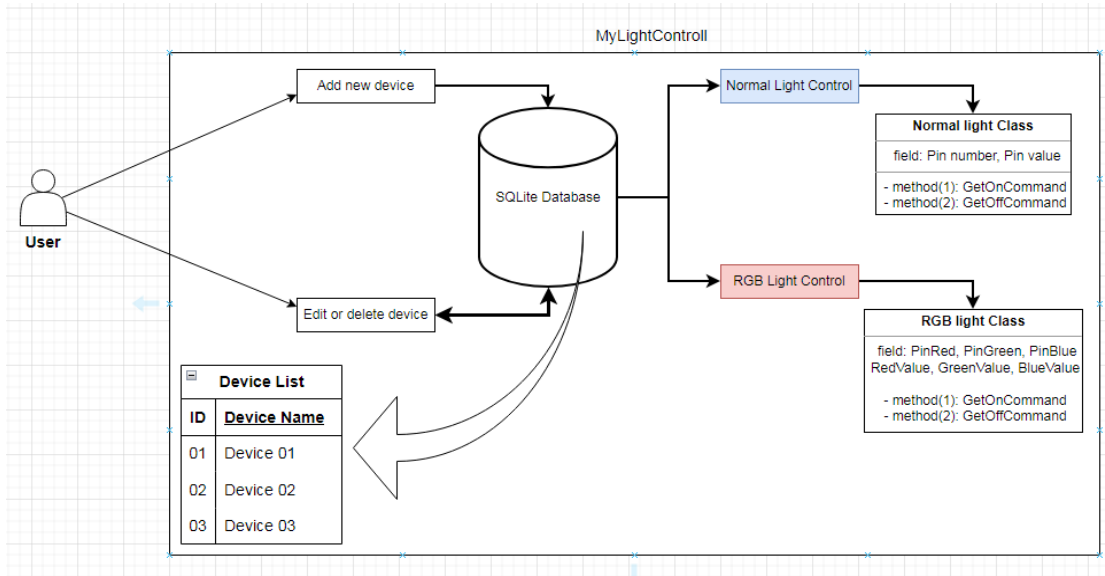


Figure 13: Schema of the application work.

The application allows the user to add a new device to be controlled via Wi-Fi after choosing one of the available device types (**Normal** or **RGB Light**). After selecting the type of device to be controlled, the application asks the user to enter the required device information:

- **Device Name:** user can input any name of his choice.
- **Device IP address:** the IP address of the NodeMCU controller after it connects to the same Wi-Fi network as the smart phone.
- **Device Pin(s) number(s):** For Normal light a single pin is used to connect to the controller, For RGB light three pins (red pin, green pin, blue pin) are used to connect to the light to the controller.
- **Device Room:** it is indicative of the location of the device, the application can handle as much of controller devices as the user has.

### 4.2.2.4. Core actions

To send HTTP request to the controller from the phone we use Volley library.

## Chapter 02: Smart wireless android led controller

The command is send using the following method:

```
protected void sendRequest(String url) {
    RequestQueue queue = Volley.newRequestQueue( context: this);
    StringRequest stringRequest = new StringRequest(Request.Method.GET, url, System.out::println, System.out::println);
    // Add the request to the RequestQueue.
    queue.add(stringRequest);
}
```

Figure 14: Volley java code.

For the **Normal Light** the application forms the HTTP request with a command that contain the pin number of controller and the following values (HIGH or 255 to power ON / LOW or 0 to power OFF the light). The following methods helps to get the string of the command from the Normal Light class.

```
public String GetOnCommand(String URL) { // send On command to NodeMCU (HIGH=255)
    return URL+"command="+this.Pin+"_255_";
}
public String GetOffCommand(String URL){ // send On command to NodeMCU (LOW=000)
    return URL+"command="+this.Pin+"_000_";
}
```

Figure 15: Normal Light Java class.

For the **RGB Light** the application sends an HTTP request with a command that contains the information about three pins (red, green and blue pins) and the values for each pin that ranges from 0 to 255. The following methods helps to get the string of the command from the RGB Light class.

```
public String GetOnCommand(String URL,String RValue, String GValue, String BValue)
{ // send On command and color Value to NodeMCU (Value from 000 to 255) RGB Value
    return URL+"command="+this.PinR+"_"+RValue+"_"+this.PinG+"_"+GValue+"_"+this.PinB+"_"+BValue+"_";
}
public String GetOffCommand(String URL){ // send OFF command to NodeMCU (LOW=00)
    return URL+"command="+this.PinR+"_000_"+this.PinG+"_000_"+this.PinB+"_000_";
}
```

Figure 16: RGB Light Java class

### 4.2.3. The Controller Code:

In the previous section we presented the application code architecture, organization and actions, in this section we present the code that allows the controller to read and analyses the commands sent from the controlling device (smartphone) and perform the actual action the user intends.

Since the controller is a NodeMCU, the programming is done using the C++ language in the Arduino IDE. The controller has a maximum of 16 GPIO to use, so we can have 16 Simple lights or 5 RGB lights to control in one controller and that is why all GPIO will be output.

*Note:* for this project will only use 9 GPIO (D0 To D8) to enter pin number to the application to work see the following table:

## Chapter 02: Smart wireless android led controller

NodeMCU Pin	The application pin
D0	16
D1	05 or 5
D2	04 or 4
D3	00 or 0
D4	02 or 2
D5	14
D6	12
D7	13
D8	15

*Table 02:Table of Pin configuration.*

Example:

If the user connects pin D3 to a normal led he need to add 00 or 0 in the Device pin in the application so that the led work.

The following two steps allows the controller to work as a light module controller:

- The controller will connect to the Wi-Fi network available after user add the information needed (SSID, PASSWORD). After that, the NodeMCU will connect to the network and show the IP Address in the serial monitor (the IP can be affected in the home router using Mac address).
- The controller receives an HTTP request from android application that contains the command to perform by the controller.

## Chapter 02: Smart wireless android led controller

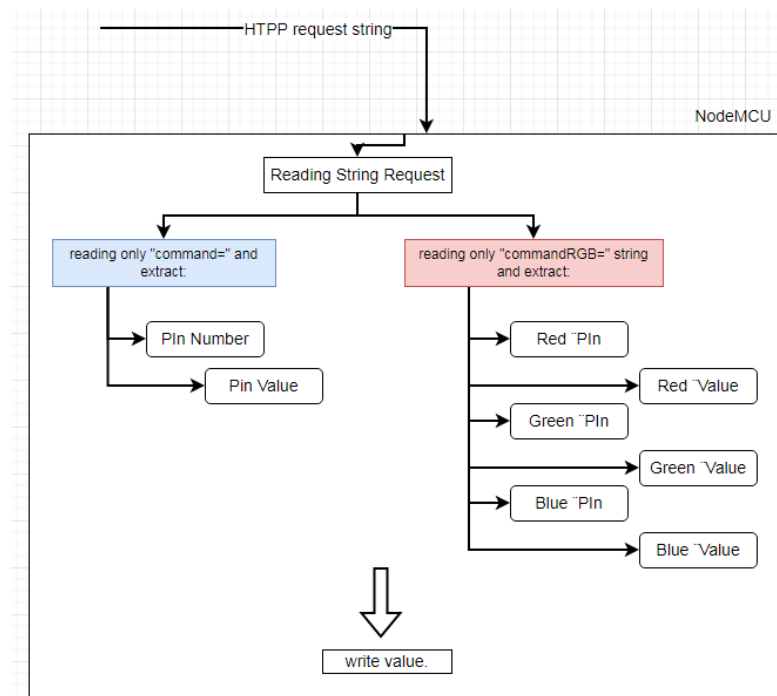


Figure 17: Schema of NodeMCU Processing request

### 4.2.3.1. HTTP request:

The HTTP request that has the command from the user input has a specific form that starts with, for Simple light "command=":

Example of the full request:

```
<< http://ip-adress-of-nodemcu/command=Pin_000_ >>
```

From this request the NodeMCU pin number related to the device and its value to make the device work we use:

```
digitalWrite(PinNumber, PinValue);
```

For RGB light it starts with "commandRGB=":

Example of the full request:

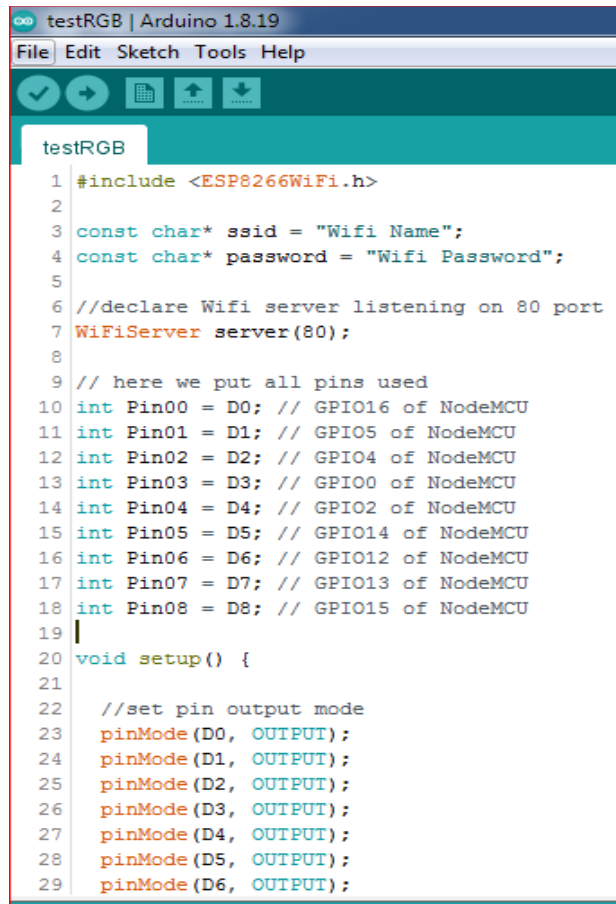
```
<< http://ip-adress-of-nodemcu/commandRGB=PinRed_000_PinGreen_000_PinBlue_000_ >>
```

From this request the NodeMCU pin number related to the device and its value to make the device work we use:

```
digitalWrite(RedPinNumber, RedPinValue);
```

```
digitalWrite(GreenPinNumber, GreenPinValue);
```

```
digitalWrite(BluePinNumber, BluePinValue);
```

The image shows a screenshot of the Arduino IDE interface. The window title is 'testRGB | Arduino 1.8.19'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for a checkmark, a right arrow, a document, an upload arrow, and a download arrow. The main editor area shows the code for a sketch named 'testRGB'. The code includes a preprocessor directive for the ESP8266 WiFi library, defines SSID and password constants, declares a WiFiServer on port 80, and lists pin definitions for GPIO pins D0 through D8. The setup function configures these pins as outputs.

```
testRGB
1 #include <ESP8266WiFi.h>
2
3 const char* ssid = "Wifi Name";
4 const char* password = "Wifi Password";
5
6 //declare Wifi server listening on 80 port
7 WiFiServer server(80);
8
9 // here we put all pins used
10 int Pin00 = D0; // GPIO16 of NodeMCU
11 int Pin01 = D1; // GPIO5 of NodeMCU
12 int Pin02 = D2; // GPIO4 of NodeMCU
13 int Pin03 = D3; // GPIO0 of NodeMCU
14 int Pin04 = D4; // GPIO2 of NodeMCU
15 int Pin05 = D5; // GPIO14 of NodeMCU
16 int Pin06 = D6; // GPIO12 of NodeMCU
17 int Pin07 = D7; // GPIO13 of NodeMCU
18 int Pin08 = D8; // GPIO15 of NodeMCU
19
20 void setup() {
21
22     //set pin output mode
23     pinMode(D0, OUTPUT);
24     pinMode(D1, OUTPUT);
25     pinMode(D2, OUTPUT);
26     pinMode(D3, OUTPUT);
27     pinMode(D4, OUTPUT);
28     pinMode(D5, OUTPUT);
29     pinMode(D6, OUTPUT);
```

Figure 18: Arduino IDE code Example.

### 4.2.4. Library used:

This is our 3d part library we used in our project, they are free to use and open source.

#### 4.2.4.1. Volley:

Volley is a library has the features like automatic scheduling of network request, multiple concurrent connections, request prioritization, faster. Volley Library was announced by Ficus Kirkpatrick at Google I/O '13. And now it is a standalone library.

#### 4.2.4.2. Holocolor:

Holo Color Picker is a library that displays a Color Picker for choosing colors. It can be used with all views, from Text View, Buttons to Layout. It's an Android Holo themed color picker designed by Marie Schweiz.

#### 4.2.4.3. SQLite Database:

SQLite is an open source SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation [10].



## *Chapter 02: Smart wireless android led controller*

### 4.2.4.4. Material design:

Material is a design system created by Google to help teams build high-quality digital experiences for Android, iOS, Flutter, and the web.

### 4.2.4.5. Esp8622.h:

ESP8266 library for Arduino that facilitates the implementation of Wi-Fi communication via user sketches.

### 4.2.5. *Programs used in development:*

Android Studio is program needed to make android application developed by Google company and JetBrains. It works based on IntelliJ IDEA.



*Figure 19: Android studio Icon.*

Arduino IDE is the software use to program Arduino codes and upload it to different type of Arduino microcontroller you can also use other different boards.

## **5. Conclusion:**

In this chapter we talk about our objective to create smart light system remotely controlled throw android application we custom made it, we explain what electronic part we need and our code how it works.

## **Chapter 03: Tests and results**

## 1. Introduction:

To present any project in a good and clear way, we need to make the project close and touch the reality in order to make suitable examples for it, and this is what we have done by designing a mini home suitable for displaying the project

## 2. Electric circuit:

Show the method of connecting all electrical wires used between the controller and devices under control in this project.

### 2.1. Circuit of Normal led:

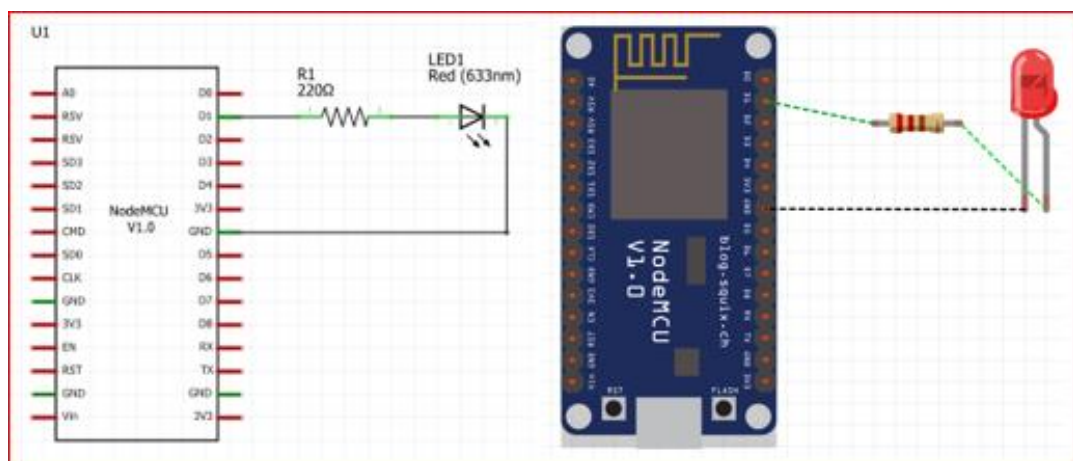


Figure 20: Circuit of NodeMCU wired with Normal led.

#### 2.1.1. Component of circuit:

- **UI: NodeMCU v1.**
- **R1: Resistance 220 ohms.**
- **LED 1: Normal single color led.**

#### 2.1.2. Function explanation:

We are connecting NodeMCU with Normal led in pin D1, which is correspondent to GPIO4, and the other for GND. After NodeMCU finishes reading the string request command coming from the phone application and extracting the necessary information (Pin number, Pin value), the LED status will change according to the command.

## 2.2. Circuit of RGB led:

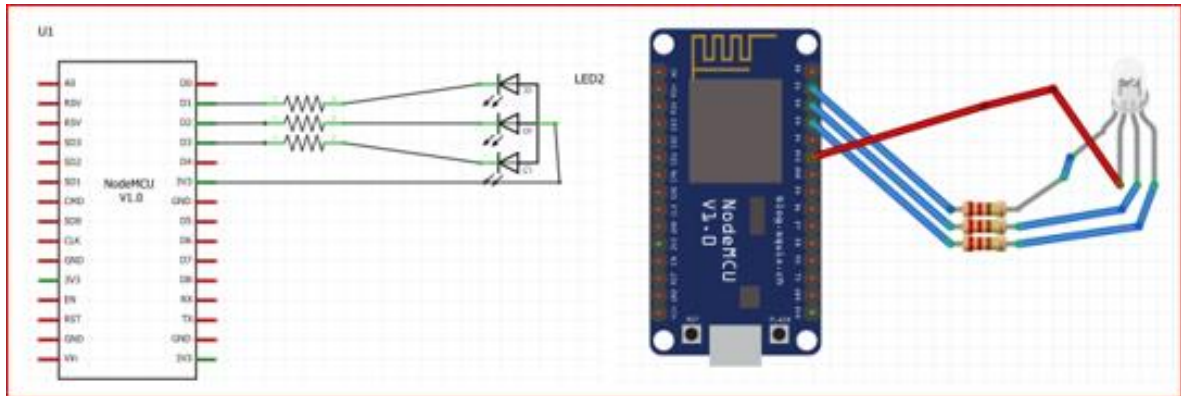


Figure 21: Circuit of NodeMCU wired with RGB led

### 2.2.1. Component of circuit:

- **U1: NodeMCU v1.**
- **3 Resistance 220 ohms.**
- **RGB led (common anode).**

### 2.2.2. Function explanation:

NodeMCU pins D1, D2, D3 connecting to RGB Pins red, green, blue, and the other pin connecting to 3v pin of NodeMCU. After NodeMCU finishes reading the string request command coming from the phone application and extracting the necessary information (Pin number, Pin value for every color), the LED status will change according to the command.

## 3. Making Mini Home Modal:

As essential part of this project was making and designing mini home modals (to be more specific two rooms, not the hole home) for simulation, testing and analyzing results. We created the modal design using word office shapes. The material we used to create the modal was wood parts. We cut the house pieces at the carpenters, prepared them in appropriate sizes. After that, we assembled all the pieces to form the final form, as the below Fig shows.:

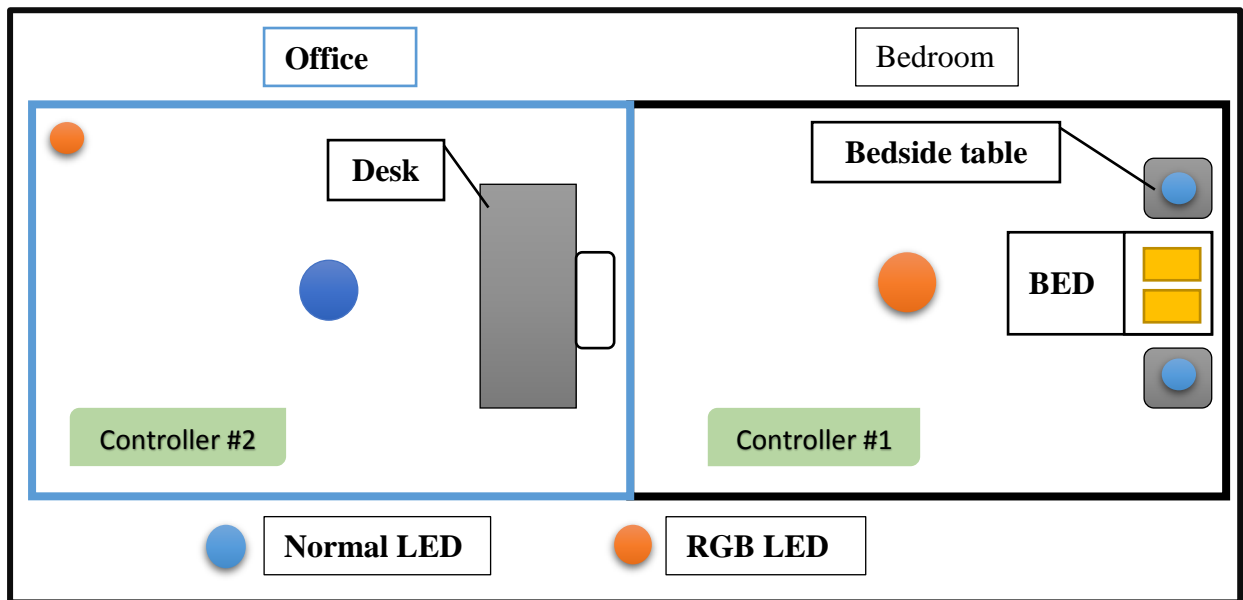


Figure 22: Mini home modal design.



Figure 23: Real photo of mini home modal Top View angle.



Figure 24: Real photo of mini home modal Front View angle.

## 4. Implementation of the Project:

Implementation of the controller in mini home modal is easy and simple. All we need are a few steps to follow:

### 4.1. Configuration of NodeMCU:

Connect the NodeMCU to our computer and open Arduino IDE select the board ESP8266 and select port the NodeMCU connects to it, open controller code and modify the Wi-Fi network SSID and PASSWORD, and upload the code to our controller.

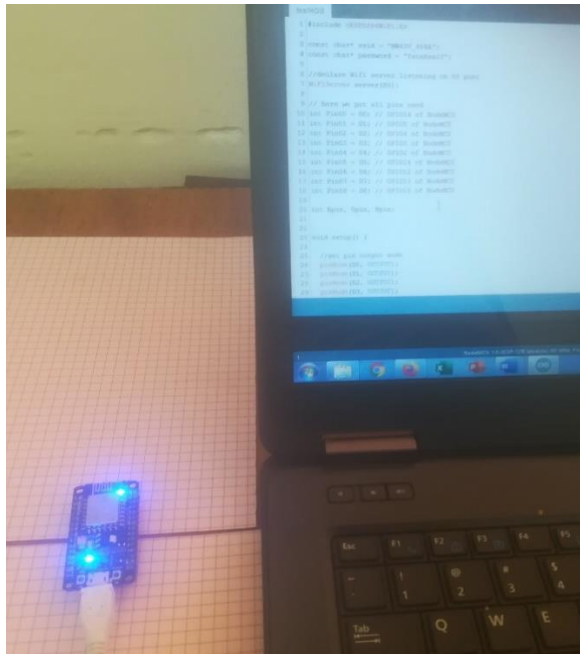


Figure 25: NodeMCU connected to pc.

We need a router to get the Wi-Fi network that will be the bridge between controller and remote control (the application), which will allow us to remotely control the lighting devices of our mini-house.

### 4.2. Pin selection:

Connect NodeMCU to led devices via jumper wire to any pin from D0 to D8. So, we can have either 3 RGB Only or 9 Normal led or combination of them together.

We will add a controller under each room (Bedroom and Office) to connect the lighting devices with their rooms.

Connect wire to selected controller pins. For Normal light devices we need 1 pin to control and the other for GND, and for RGB light devices we need 3 pins (1 for red, 1 for green, 1 for blue) and 1 pin for GND.

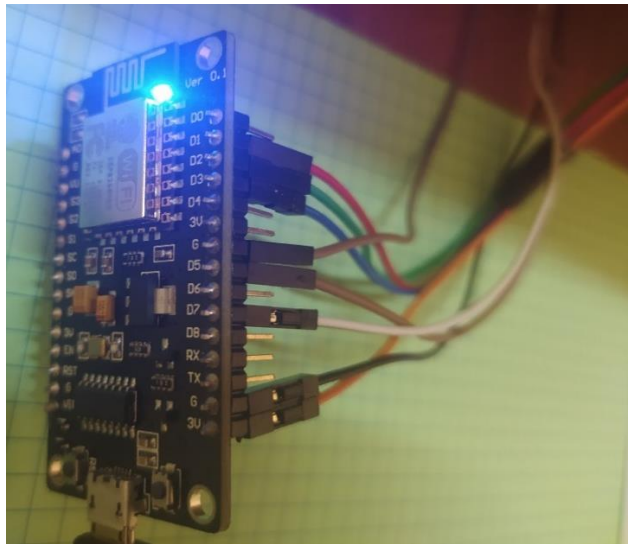
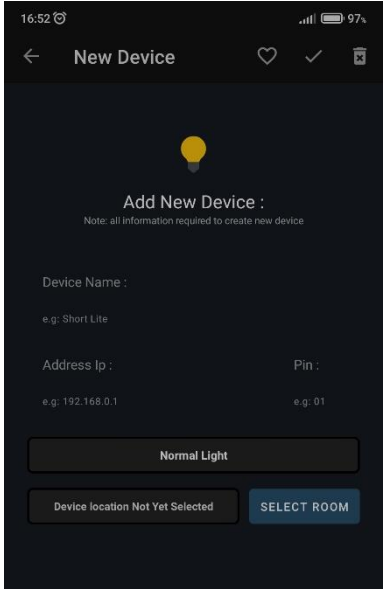
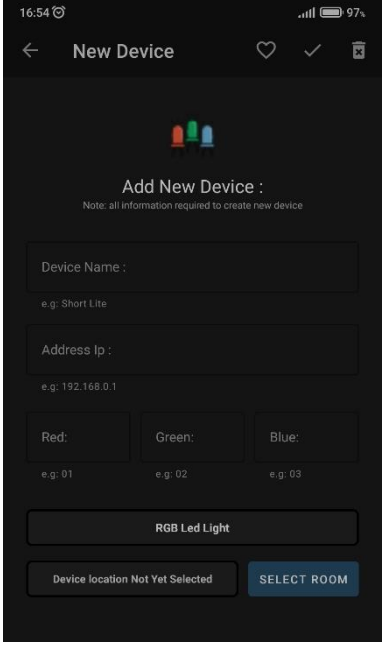


Figure 26: Example of pin selecting of NodeMCU\

### 4.3. Configuration of smart light application:

#### 4.3.1. Add new device:

To add new device, we click on the Plus + button a dialog screen appears, we select device type form two option:

Normal LED	RGB LED
 <p>Figure 27: Application screen shot of add normal device activity.</p>	 <p>Figure 28: Application screen shot of add RGB device activity.</p>
<p>Normal led needs this information: Device name, NodeMCU IP that is connected to the device, pin number, and finally room name and click save..</p>	<p>RGB led needs this information: Device name, NodeMCU IP that is connected to the device, Red pin number, Green pin number, Blue pin number, and finally room name and click save.</p>

## Chapter 03: Tests and results

Next, we go to **Rooms Section**, we will notice the names of the rooms, and when you click on any room name, the devices list of that room will appear as showing below:

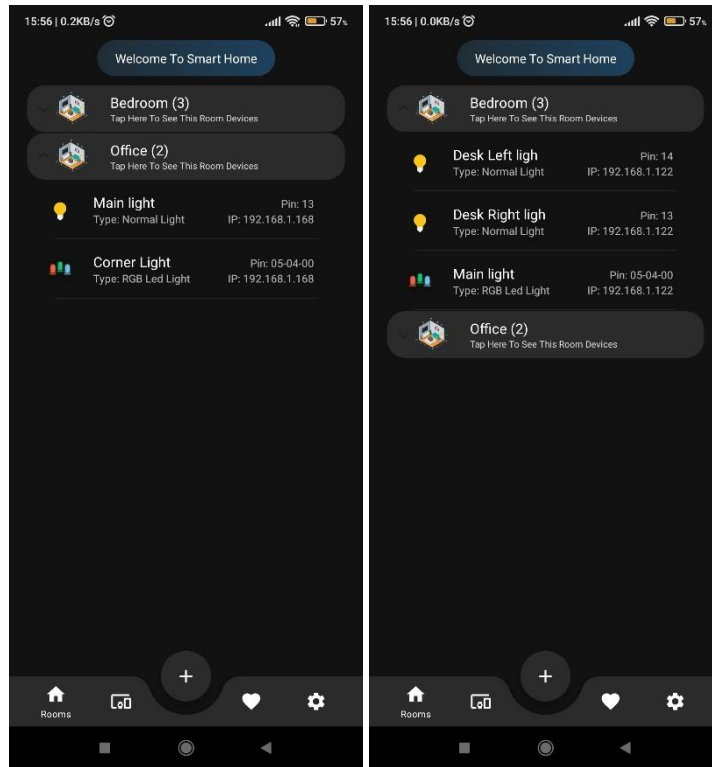


Figure 29: Application screen shot of Rooms section.

Click on the name of any device to open its control interface, which contains the device's information as well as the control buttons (such as turning on and off, changing the intensity of lighting) for Normal light and for RGB light we have the device's information as well as the control buttons (such as turning on and off, color picker and 2 lines of color to pick from them).



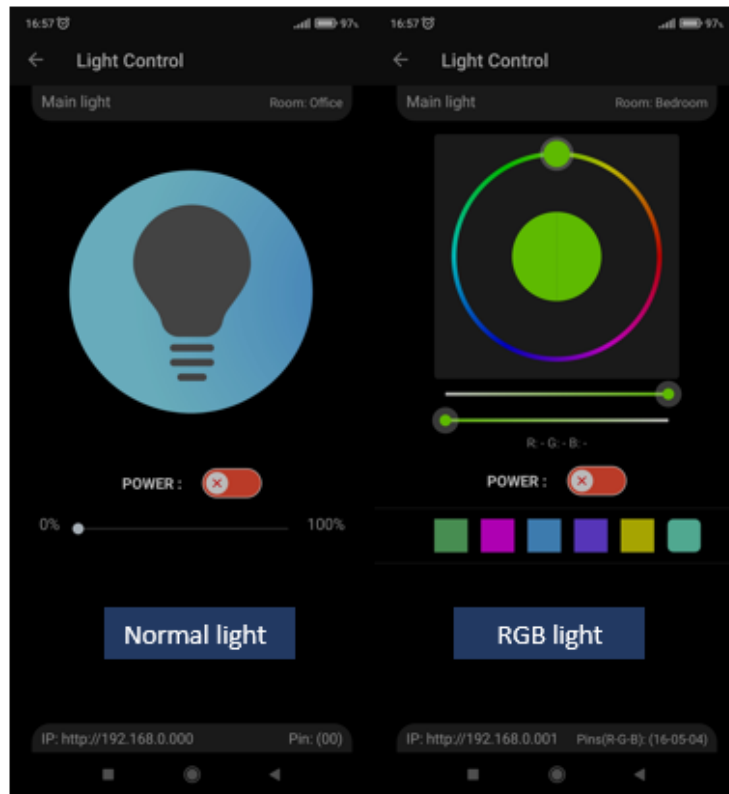


Figure 30: Application screenshot of normal light control and RGB light control.

#### 4.3.2. Edit or delete device:

To change the device information or delete it, we go to **Devices Section** and click name of any device we want. After that, a new screen appears (Edit device screen). We modify the information and press Save to modify the device information, or we can delete the device by clicking the trash icon in top bar and click Ok.

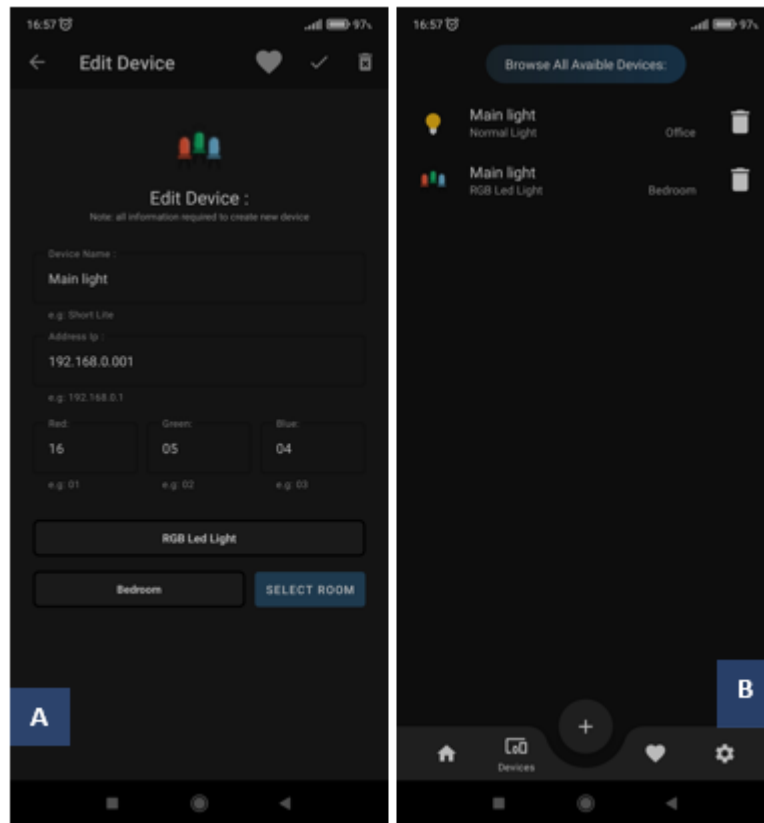


Figure 31: A- example of edit device screen, B-Device section.

Other way to delete device is by clicking in the trash icon in **Devices section**, and the device will be deleted immediately after clicking ok.

## 5. Using a relay:

The implementation of the controller with relay is not that different from Implementation of the controller in mini home modal all, we need to do is:

- Connect the NodeMCU with relay (what you need to use either 1, 2, 4 channel 5v, 12, 24v) after that connect the light device with relay.

Then continue with the previous steps.

## 6. Testing & the result:

The mini house model consists of two rooms, a Bedroom and an Office room the rooms have the following facilities:

In the **bedroom** we have:

- Two Normal led lighting devices.
- RGB led light device.

In the **office** room we have:

- Normal led lighting devices.
- RGB led light device

### *Chapter 03: Tests and results*

Devices information for the Bedroom is:

- **Device 01:** RGB led
  - Device Name: Main light.
  - Device IP: 192.168.1.122.
  - Device Pins(R-G-B): (05-04-00).
  - Device Room: Bedroom.
- **Device 02:** Normal led
  - Device Name: Right bed light.
  - Device IP: 192.168.1.122.
  - Device Pin: 13.
  - Device Room: Bedroom.
- **Device 03:** RGB led
  - Device Name: Left bed light.
  - Device IP: 192.168.1.122.
  - Device Pin: 14.
  - Device Room: Bedroom.

Devices information for the Office room is:

- **Device 01:** Normal led
  - Device Name: Main light.
  - Device IP: 192.168.1.168.
  - Device Pin: 13.
  - Device Room: Office.
- **Device 02:** RGB led
  - Device Name: Corner light.
  - Device IP: 192.168.1.168.
  - Device Pins(R-G-B): (05-04-00).
  - Device Room: Office.

Now we turn on or off same device for example turn on Device 1 in Bedroom and Device 2 in Office and see the result:

### Chapter 03: Tests and results



Figure 32: Device 01 in Bedroom turn On.

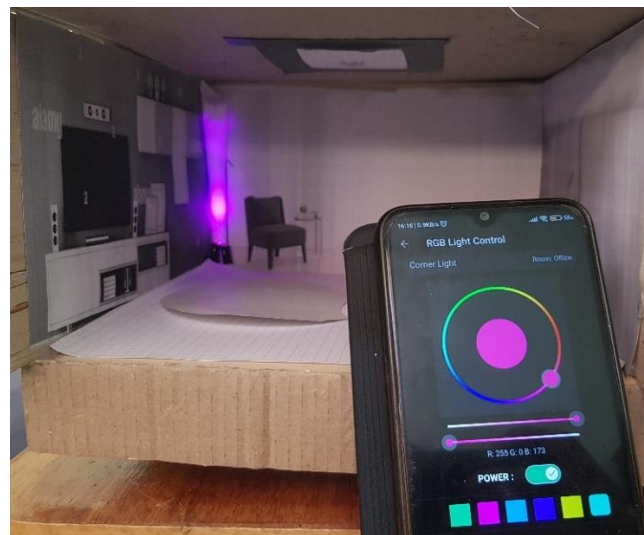


Figure 33: Device 01 in Office room turn On.

Next, we change the color of Device 01 of Bedroom:



Figure 34: Device 01 in Bedroom change color from red to blue green.

Now turn on all lights in rooms:

## Chapter 03: Tests and results



Figure 35: All mini home devices is turn On.

Turn of all lights in rooms:

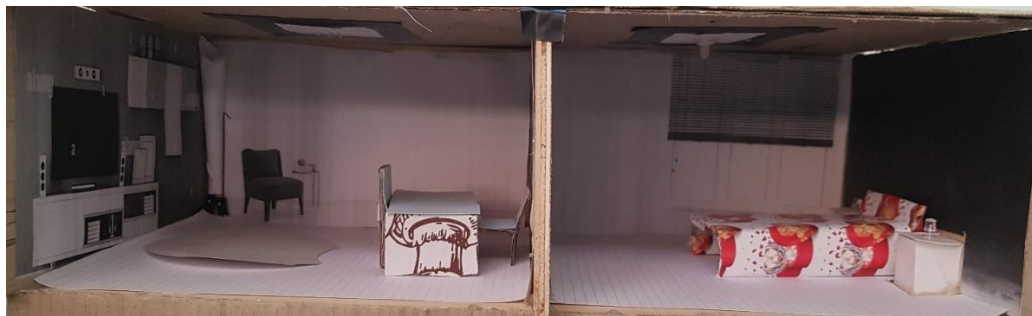


Figure 36: All mini home devices is turn Off.

### 6.1. Analyzing the result:

All we can see is that the project is as expected. User can add light devices and control them. The android application works fine device information saved and the control status and color get saved too for the user if he wants to exit from the application and come back again..

Our best result was when we were at night, those photos will tell more:



Figure 37: All devices turns OFF.

When the two RGB led turn on:

## Chapter 03: Tests and results



Figure 38: Two RGB led Turns on.

When change the RGB colors:



Figure 39: Two RGB led Turns on with different color.

When all devices are on:



Figure 40: All devices is on.

When all office room lights are off and all bedroom lights are on:



Figure 41: Office room off and Bedroom on.

### *Chapter 03: Tests and results*

The app save device status like device is on or off and save its color (RGB only) when the user exit from the app and came back again he will find the same info before he exits.

#### **7. Conclusion:**

In this chapter we talk about implementation of the controller in mini home modal who we create to display the objective of this Project. We mention all needed step to implement, photo to show the result and preparation.

## **Final Conclusion**



The smart home is the next step for societies, especially after the spread of networks and smart devices. It is also necessary for ordinary homes to develop because they have different types of controllable electronic devices, in order to reduce repetitive actions and facilitate people's lives. One of the most important things is the availability of many options at different prices in the market, and this will speed up and improve the spread of the idea of smart homes, or at least sub-home systems such as lighting and protection .... etc.

The goal of this project is to design a controller to for the lighting devices remotely via the Wi-Fi network and through an Android application, and we achieved this despite the problems and obstacles we faced and many limits that did not allow us to provide more. These limits are the maximum number for the number of devices that we can connect to it with one controller. In a better scenario, multiple controllers can be used to showcase the abilities of the android application that can control not only one device but multiple devices at once rendering the application more useful.

For future developments, our Android application is ready and capable of adding other types of devices, and it is adjustable for any similar purpose, such as a protection system or a heat system... etc. Adding a machine learning module or artificial intelligence module to the application to allow the application to select the actions to perform is a necessity in the future.

## ***Bibliography:***

[1] International Journal of Application or Innovation in Engineering & Management (IJAIEM), 2013. Smart home design using wireless sensor network and biometric technologies. [online], pp.413-416. Accessed 17 June 2022 Available at: <[https://www.researchgate.net/publication/241508323\\_Smart\\_Home\\_Design\\_using\\_Wireless\\_Sensor\\_Network\\_and\\_Biometric\\_Technologies](https://www.researchgate.net/publication/241508323_Smart_Home_Design_using_Wireless_Sensor_Network_and_Biometric_Technologies)>.

[2] R. Harper, Inside the smart home. London: Springer, 2006, pp. 1-4. Accessed: Jun 17, 2022. [Online]. Available: <<https://books.google.dz/books?id=3J0MBwAAQBAJ&lpg=PA1&ots=zohCgMjfkF&dq=history%20of%20smart%20homes&lr&pg=PP1#v=onepage&q&f=false>>.

[3] "Advantages and Disadvantages of Smart Homes", Smart Home Insider, 2021. Accessed: Jun 17, 2022. [Online]. Available: <<https://smarthomeinsider.co.uk/advantages-and-disadvantages-of-smart-homes/>>.

[4] B. Dvoršak and J. Havelka and Other, SMART HOME SYSTEMS. European Union, pp. 7-12. Accessed: Jun 16, 2022. [Online]. Available: <[https://ec.europa.eu/programmes/erasmus-plus/project-result-content/4df4e928-8958-4552-80da-146977e666b9/Smart\\_Home\\_systems\\_FINAL.pdf](https://ec.europa.eu/programmes/erasmus-plus/project-result-content/4df4e928-8958-4552-80da-146977e666b9/Smart_Home_systems_FINAL.pdf)>.

[5] K. Omar, How to Smart Home: *A Step by Step Guide Using Internet, Z-Wave, KNX & OpenRemote*, 1th ed. e.K., Wyk, Germany: Key Concept Press, 2013, pp. 23–24. Accessed: Jun 16, 2022. [Online]. Available: <<http://www.openremote.com/wp-content/uploads/2013/12/How-To-Smart-Home-PDF-OR.pdf>>.

[6] H. V. Bhatnagar, P. Kumar, S. Rawat and T. Choudhury, "Implementation model of Wi-Fi based Smart Home System," *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 2018, pp. 23-28, Accessed: Jun 16, 2022. [Online]. doi: <10.1109/ICACCE.2018.8441703>.

[7] R. A. Ramlee, M. A. Othman, M. H. Leong, M. M. Ismail and S. S. S. Ranjit, "Smart home system using android application," *2013 International Conference of Information and Communication Technology (ICoICT)*, 2013, pp. 277-280, doi: <10.1109/ICoICT.2013.6574587>.

[8] G. M. Madhu and C. Vyjayanthi, "Implementation of Cost-Effective Smart Home Controller with Android Application Using Node MCU and Internet of Things (IOT)," *2018 2nd International*

### ***Bibliography:***

*Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE)*, 2018, pp. 1-5, Accessed: Jun 16, 2022. [Online]. doi: <10.1109/EPETSG.2018.8659128>.

[9] <<https://handsontec.com/dataspecs/module/esp8266-V13.pdf>>.

[10] Aditya, S.K. and Karn, V.K., 2014. *Android SQLite essentials* (pp. 88-90). Packt Publishing. [Online]. Available : <[http://www.bibliotecasyarchivos.net/sitio\\_bdusam/bibliotecadigital/ebook/facultad\\_de\\_ciencias\\_empresariales/Licenciatura\\_en\\_computacion/Aplicaciones\\_Moviles/Android\\_SQLite\\_Essentials\\_nodrm.pdf](http://www.bibliotecasyarchivos.net/sitio_bdusam/bibliotecadigital/ebook/facultad_de_ciencias_empresariales/Licenciatura_en_computacion/Aplicaciones_Moviles/Android_SQLite_Essentials_nodrm.pdf)>

