



**UNIVERSITE HASSIBA BENBOUALI DE CHLEF**

**Faculté de Technologie**

**Département d'Electronique**

## **MEMOIRE DE MASTER**

Domaine : SCIENCES ET TECHNOLOGIES

Filière : : Electronique

Spécialité : Electronique des Systèmes Embarqués

Design and Implementation of a Mobile Application for Real-Time  
Control and Monitoring of Smart Home Systems

Par

**Abdelheq MEKKI**

Encadreur :

M. BENICHOU Boucif

Maître de Conférences A à l'UHBC

Co-Encadreur :

Melle. DJEGHLOUF Asma

Maitre de Conférences B à l'UHBC

Chlef, Juin 2025

## Acknowledgements

"الشكر و الحمد لله"

*First and foremost, I would like to thank Allah, the Almighty, who gave me the strength, patience, and determination to complete this work, Alhamdulillah.*

*I sincerely thank my supervisors, Mr. BENICHOU Boucif and Ms. DJEGHLOUF Asmaa, Professors at Hassiba Benbouali University of Chlef, for their expert advice, availability, support, and patience.*

*I'm deeply grateful for everything they have done. The valuable time they dedicated to me.*

*I would also like to acknowledge the members of the jury who kindly reviewed and evaluated my work.*

*I warmly thank my parents from the bottom of my hearts for their unwavering support morally, emotionally, and financially throughout this journey.*

*Lastly, my gratitude extends to everyone who contributed, directly or indirectly, to the completion of this modest work.*

## Dedications

*I have the great pleasure to dedicate this modest work:*

*To my dearest mother: I always find her by my side, she who is the serenity of my heart, my comfort, and my joy in life. Her presence is my inspiration and strength to face various obstacles.*

*To my dear father: You have always been my support and security in life, my courage, my trust, and my strength.*

*You always seek my comfort and ensure nothing hinders my completion of studies. You have always been by my side to support and encourage me. May this work reflect my gratitude and affection.*

*To my dear brother (AYOUB) and sister: May God grant them health, happiness, courage, and above all, success.*

*To all my best friends. And to all those who have helped and contributed to this modest work*

*ABDELHEQ*

## **Abstract**

This study presents the design and implementation of a smart home system based on the Arduino Mega 2560 microcontroller. The system enhances security and comfort through features including voice-controlled doors and lighting, along with real-time smoke, gas, temperature, and humidity detection (using flame, gas, temperature, and humidity sensors). All functions are remotely controlled via an Android application, ensuring resident safety and convenience. The project combines electronics, automation, and IoT technologies to optimize energy efficiency while providing real-time monitoring.

**Keywords:** Smart Home, Sensors, Home Automation, Arduino Mega 2560, IoT, Android Application, Voice Recognition, Security.

## ملخص

تقدم هذه الدراسة تصميم وتنفيذ نظام منزل ذكي يعتمد على متحكم Arduino Mega 2560. يعمل النظام على تعزيز الأمان والراحة من خلال ميزات تشمل التحكم الصوتي في الأبواب والإضاءة، بالإضافة إلى كشف الدخان والغاز ودرجة الحرارة والرطوبة (باستخدام مستشعرات اللهب والغاز والحرارة والرطوبة) بشكل لحظي. يتم التحكم بجميع هذه الوظائف عن بعد عبر تطبيق أندرويد، مما يضمن الأمان والراحة للسكان. يدمج المشروع بين التقنيات الإلكترونية وأنظمة الأتمتة وإنترنت الأشياء لتحسين كفاءة الطاقة مع توفير مراقبة فورية.

إنترنت، Arduino Mega 2560 المنزل الذكي، أجهزة الاستشعار، الأتمتة المنزلية، **الكلمات المفتاحية** الأشياء، تطبيق أندرويد، التعرف الصوتي، الأمان.

**Résumé**

Cette étude présente la conception et la réalisation d'un système domotique basé sur le microcontrôleur Arduino Mega 2560. Le système améliore la sécurité et le confort grâce à des fonctionnalités incluant le contrôle vocal des portes et de l'éclairage, ainsi que la détection en temps réel de fumée, de gaz, de température et d'humidité (à l'aide de capteurs de flamme, de gaz, de température et d'humidité). Toutes ces fonctions sont contrôlables à distance via une application Android, garantissant sécurité et commodité aux résidents. Le projet combine l'électronique, l'automatisation et les technologies IoT pour optimiser l'efficacité énergétique tout en permettant une surveillance en temps réel.

**Mots-clés:** Maison Intelligente, Capteurs, Domotique, Arduino Mega 2560, IoT, Application Android, Reconnaissance Vocale, Sécurité.

## Table of contents

Acknowledgements .....	i
Dedications .....	ii
Abstract.....	iii
ملخص.....	iv
Résumé .....	v
Table of Contents.....	vi
Table of figures .....	x
List of tables .....	xii
List of abbreviations.....	xiii
<b>General Introduction.....</b>	<b>1</b>
References .....	3

### Chapter I : Generalities on Smart Homes

<b>I.1. Introduction.....</b>	<b>5</b>
<b>I.2. problematic .....</b>	<b>5</b>
<b>I.3. History of Smart Home .....</b>	<b>5</b>
I.3.1. Here Comes Wireless .....	6
I.3.2. Smart Homes Now .....	7
<b>I.4. Definition of Smart Home .....</b>	<b>8</b>
<b>I.5. Benefits of a Smart Home.....</b>	<b>9</b>
<b>I.6. Smart Home Technology Overview .....</b>	<b>9</b>
<b>I.7. Internet of thing (IOT) in Smart Home .....</b>	<b>10</b>
<b>I.8. IOT Smart Home Automation Future.....</b>	<b>12</b>
<b>I.9. Advantages of Smart Home.....</b>	<b>13</b>
<b>I.10. Disadvantages of Smart Home.....</b>	<b>15</b>
<b>I.11. Conclusion.....</b>	<b>15</b>
References.....	17

### Chapter II : Hardware used in Smart Homes

<b>II.1. Introduction .....</b>	<b>19</b>
<b>II.2. Materials used.....</b>	<b>19</b>
<b>II.3. Choice of the programmable card .....</b>	<b>20</b>
<b>II.4. What is an Arduino? .....</b>	<b>20</b>
<b>II.5. What is an Arduino board? .....</b>	<b>21</b>
<b>II.6. Arduino board family.....</b>	<b>22</b>

II.6.1. Arduino UNO board .....	22
II.6.2. Arduino Mega 2560 board .....	23
II.6.3. Arduino LEONARDO board.....	23
II.6.4. Arduino Micro board.....	23
II.6.5. Arduino NANO board .....	23
II.6.6. Arduino Due board.....	23
II.6.7. Arduino Esplora board .....	23
II.6.8. Arduino Yun Mini board.....	24
<b>II.7. Arduino Mega 2560 board .....</b>	<b>24</b>
II.7.1. Why the Arduino Mega 2560 board? .....	24
II.7.2. Technical specifications of the Arduino Mega 2560 board.....	25
<b>II.8. What is a sensor? .....</b>	<b>27</b>
<b>II.9. Sensor used.....</b>	<b>27</b>
II.9.1. Gas Sensor Module MQ-05 .....	28
II.9.2. DHT11 Temperature and Humidity Sensor.....	29
II.9.3. Rain Sensor .....	30
II.9.4. Flame Sensor.....	31
<b>II.10. Module used .....</b>	<b>32</b>
II.10.1. HC-05 Bluetooth Module.....	32
<b>II.11. Actuators used.....</b>	<b>34</b>
II.11.1. Servomotor .....	34
II.11.2. Fan.....	35
II.11.3. Buzzer.....	36
II.11.4. LEDs.....	37
II.11.5. LCD (Liquid crystal display) .....	38
<b>II.12. Conclusion.....</b>	<b>39</b>
<b>References.....</b>	<b>40</b>

### **Chapter III : Software used in Smart Homes**

<b>III.1. Introduction.....</b>	<b>42</b>
<b>III.2. Arduino programming platform .....</b>	<b>42</b>
<b>III.3. Presentation of the Software .....</b>	<b>42</b>
<b>III.4. Buttons .....</b>	<b>44</b>
<b>III.5. Simulation platform.....</b>	<b>45</b>
III.5.1. Proteus Software .....	45
III.5.2. ISIS .....	46
III.5.3. ARES .....	47

<b>III.6. Development of the Arduino application .....</b>	<b>49</b>
III.6.1. Android .....	49
III.6.2. MIT App Inventor .....	50
III.6.2.1. What we can program with App Inventor .....	50
III.6.2.2. Why MIT App Inventor.....	50
III.6.2.3. Structure of an MIT App Inventor.....	51
<b>III.7. Conclusion .....</b>	<b>56</b>
<b>References.....</b>	<b>57</b>

### **Chapter IV : Smart Home Prototype Development: From System Design to Mobile Application Integration**

<b>IV.1. Introduction.....</b>	<b>59</b>
<b>IV.2.General synoptic diagram.....</b>	<b>59</b>
<b>IV.3. Specification and analysis of requirements .....</b>	<b>60</b>
IV.3.1. Functional Requirements and No-Functional Requirements .....	60
IV.3.1.1. Functional Requirements.....	60
IV.3.1.2. No-Functional Requirements.....	61
<b>IV.4. Overall system architecture .....</b>	<b>61</b>
IV.4.1. User Part.....	61
IV.4.2. Arduino and software Part.....	62
IV.4.3. Hardware Part.....	62
<b>IV.5. Simulation.....</b>	<b>62</b>
IV.5.1. Simulation Result.....	63
IV.5.1.1. Gas leak Detection.....	63
IV.5.1.1.1. Flowchart for the Gas sensor .....	64
IV.5.1.2. Flame detection .....	64
IV.5.1.2.1. Flowchart for the Flame sensor .....	65
IV.5.1.3.Rain detection .....	65
IV.5.1.3.1. Flowchart for the Rain sensor.....	66
IV.5.1.4.DHT11 Sensor (Temperature/Humidity) .....	67
IV.5.1.4.1. Flowchart for the DHT11 sensor(Temperature/Humidity).....	68
IV.5.1.5.Opening /closing the door (servomotor).....	68
IV.5.1.5.1. Flowchart for the Opening /closing the door (servomotor).....	69
IV.5.1.6. An indoor lighting system (LEDs) .....	70
IV.5.1.5.1. Flowchart for the indoor lighting system(LEDs) .....	71
<b>IV.6. Design and implementation part .....</b>	<b>71</b>
IV.6.1. Mobile application Design .....	71

IV.6.2. application interface .....	72
IV.6.2.1. App Control by voice.....	73
IV.6.2.2. Application Results.....	73
IV.6.3. Smart Home Design .....	76
IV.6.3.1. Prototype.....	76
IV.6.3.2. Final Result.....	77
IV.7. Conclusion .....	78
<b>General Conclusion and Perspectives.....</b>	<b>80</b>

## Table of figures

<b>Figure (I.1):</b> X10 Modules and Remote. ....	6
<b>Figure (I.2):</b> Wireless communication. ....	7
<b>Figure (I.3):</b> Wireless Technology of the Smart Home. ....	8
<b>Figure (I.4):</b> Smart Home System. ....	8
<b>Figure (I.5):</b> Smart Home Technology. ....	10
<b>Figure (I.6):</b> Remote Home Control and Monitoring. ....	11
<b>Figure (I.7):</b> Internet of things (IOT). ....	12
<b>Figure (I.8):</b> Home Security System. ....	14
<b>Figure (I.9):</b> Accessibility in Smart Home. ....	14
<b>Figure (II.1):</b> A variety of electronic components. ....	19
<b>Figure (II.2):</b> Arduino Logo. ....	21
<b>Figure (II.3):</b> Arduino UNO board. ....	21
<b>Figure (II.4):</b> Different types of Arduino boards. ....	22
<b>Figure (II.5):</b> Arduino Mega 2560 board. ....	25
<b>Figure (II.6):</b> MQ-05 Gas Sensor Module. ....	28
<b>Figure (II.7):</b> DHT11 Temperature and Humidity Sensor. ....	29
<b>Figure (II.8):</b> Rain Sensor. ....	31
<b>Figure (II.9):</b> Flame Sensor. ....	32
<b>Figure (II.10):</b> HC-05 Bluetooth Module. ....	33
<b>Figure (II.11):</b> Relay Module 5V. ....	34
<b>Figure (II.12):</b> Fan 5V. ....	35
<b>Figure (II.13):</b> Buzzer 5V. ....	36
<b>Figure (II.14):</b> LEDs. ....	37
<b>Figure (II.15):</b> LCD. ....	38
<b>Figure (III.1):</b> Logo of Software Arduino IDE. ....	42
<b>Figure (III.2):</b> The Arduino IDE interface in detail. ....	43
<b>Figure (III.3):</b> The toolbar. ....	44
<b>Figure (III.4):</b> General Structure of an Arduino sketch. ....	44
<b>Figure (III.5):</b> Logo proteus software. ....	45
<b>Figure (III.6):</b> Logo isis. ....	46
<b>Figure (III.7):</b> Principal window of ISIS. ....	47
<b>Figure (III.8):</b> Logo ARES. ....	47
<b>Figure (III.9):</b> Principal window of ARES. ....	48
<b>Figure (III.10):</b> Android Logo. ....	49
<b>Figure (III.11):</b> MIT APP INVENTOR Logo. ....	50

---

<b>Figure (III.12):</b> MIT APP INVENTOR Interface. ....	51
<b>Figure (III.13):</b> Designer tab.....	52
<b>Figure (III.14):</b> Design Interface. ....	53
<b>Figure (III.15):</b> Blocks editor tab.....	53
<b>Figure (III.16):</b> Control of Blocks editor. ....	54
<b>Figure (III.17):</b> Blocks Editor. ....	55
<b>Figure (IV.1):</b> General synoptic diagram. ....	60
<b>Figure (IV.2):</b> Overall system architecture.....	62
<b>Figure (IV.3):</b> Gas Sensor detection. ....	63
<b>Figure (IV.4):</b> Flowchart for the Gas Sensor management.....	64
<b>Figure (IV.5):</b> Flame Sensordetection. ....	64
<b>Figure (IV.6):</b> Flowchart For the Flame Sensor management.....	65
<b>Figure (IV.7):</b> Rain sensor detection. ....	66
<b>Figure (IV.8):</b> Flowchart For the Rain sensormanagement.....	67
<b>Figure (IV.9):</b> DHT11 Sensor.....	67
<b>Figure (IV.10):</b> Flowchart For the DHT11 Sensor. ....	68
<b>Figure (IV.11):</b> Servomotor. ....	69
<b>Figure (IV.12):</b> Flowchart For the Servomotor. ....	70
<b>Figure (IV.13):</b> Rooms lighting control.....	70
<b>Figure (IV.14):</b> Flowchart for the rooms lighting control. ....	71
<b>Figure (IV.15):</b> Voice recognition. ....	72
<b>Figure (IV.16):</b> Application interface.....	72
<b>Figure (IV.17):</b> voice control and monitoring the value of sensors in real-time. ....	73
<b>Figure (IV.18):</b> Control in the rooms lighting and opening the door by voice.....	74
<b>Figure (IV.19):</b> Automatically controlling the rain sensor to protect clothes from getting wet. ....	74
<b>Figure (IV.20):</b> The final results of a mobile application for real-time control and monitoring. ....	75
<b>Figure (IV.21):</b> Prototype of Smart Home.....	76
<b>Figure (IV.22):</b> Implemented tasks of a Mobile Application for Real-Time Control and Monitoring of Smart Home Systems. ....	77

## List of tables

<b>Table (II.1):</b> Technical specifications of the Arduino Mega 2560 board.....	27
<b>Table (II.2):</b> FeaturesMQ-05 Gas Sensor .....	28
<b>Table (II.3):</b> FeaturesDHT11 Temperature and Humidity Sensor .....	30
<b>Table (II.4):</b> Features Rain Sensor . .....	31
<b>Table (II.5):</b> Features Flame Sensor .....	32
<b>Table (II.6):</b> FeaturesHC-05 Bluetooth Module .....	33
<b>Table (II.7):</b> Features servomotor.....	35
<b>Table (II.8):</b> Features Fan.....	36
<b>Table (II.9):</b> Features Buzzer .....	37
<b>Table (II.10):</b> Features LEDs .....	38
<b>Table (II.11):</b> Features LCD .....	39

## **List of abbreviations**

**IOT:** internet of thing

**AI:** artificial intelligence

**AC:** alternating current

**DC:** direct current

**PCB:** printed circuit board

**UART:** universal asynchronous receiver transmitter

**USB:** Universal Serial Bus

**RS232:** recommended standard 232 (serial communication)

**RX:** receive (data line)

**TX:** transmit (data line)

**I2C:** inter-integrated circuit (2-wire communication)

**SPI:** serial peripheral interface

**SDA:** serial data line (I2C)

**SCL:** serial clock line (I2C)

**MISO:** master In slave Out (SPI)

**MOSI:** master Out slave In ( SPI)

**SCK:** serial clock (SPI)

**NPN:** negative-positive-negative (transistor 2N2222)

**CMOS:** complementary metal-oxide-semiconductor

**WIFI:** wireless fidelity

**DHT11:** digital Humidity & Temperature sensor

**LED:** light Emitting Diode

**LCD:** liquid crystal display

**GND:** ground

**VCC:** voltage common collector

**Arduino IDE:** integrated development environment for Arduino

**MIT App Inventor:** Massachusetts Institute of Technology- application inventor

**PWN:** pulse width modulation

**I/O:** In-put / Out-put

**RH%:** rated humidity

**LPG:** liquefied petroleum Gas

**SRAM:** static random-Access Memory

**EEPROM:** Electrically Erasable programming Read-Only memory

**DIY:** do it yourself

**SPP:** serial part protocol

**EDR:** Enhanced Data Rate

**AFH:** adaptive frequency hopping function

**CAD:** computer-Aided Design

*GENERAL*

*INTRODUCTION*

## **General Introduction**

The home is an extremely important place for each of us, serving as a sanctuary where we live and return. For many, especially the elderly who spend most of their time at home, the quality of housing significantly impacts their quality of life. This emphasizes the need to improve safety and comfort in homes as a matter of great social importance. In recent years, technology has been applied to create the "smart home," a residence equipped with electronics, automation, IT, and communication technologies, aimed at assisting residents in various aspects of daily life [1]. The smart home is also a specialization in the construction field, combining technologies that enable home control, automation, and function programming. This field is known as "Domotics," where its application helps users enhance comfort by creating automation scenarios tailored to their lifestyles, while also achieving significant savings in heating and electricity consumption, though accurately determining actual savings remains challenging [2]. Connected devices can also help prevent household malfunctions and avoid unnecessary expenses.

Our project aims to contribute to this field by designing and implementing a mobile application for Android that enables remote control of several home functions, such as gas leak detection, indoor temperature and humidity monitoring, lighting control, door opening and closing management, and rain detection.

To implement this project, an Arduino Mega2560 board was used as the main controller, along with a set of integrated sensors, including a gas sensor to detect hazardous leaks, a flame sensor for early fire detection, a DHT11 sensor to measure temperature and humidity, and a rain sensor to monitor weather conditions [3]. Additionally, an HC-05 Bluetooth module was incorporated to enable wireless communication between the system and the mobile app, which was designed using MIT App Inventor, providing a user-friendly interface for controlling home devices and displaying real-time data [4]. Furthermore, the system includes a voice recognition

feature that allows users to open or close doors and control lighting using voice commands, enhancing usability and accessibility [5].

Through this project, we aim to provide a comprehensive and cost-effective solution to enhance home automation, increase safety, improve energy efficiency, and enable remote control and real-time monitoring, all while delivering a simple and modern user experience that meets the needs of contemporary living [6].

This Master's thesis is organized into four chapters, preceded by a general introduction. The first Chapter provides an overview of smart homes, including their history, key advantages, and associated technologies, particularly the Internet of Things (IoT). The second chapter focuses on the various hardware components required for the implementation of the home automation project. Chapter III presents the software tools necessary for the development of the system. Finally, chapter IV covers the design and development of the mobile interface, the implementation of sensor communication, and the integration of voice control features to enhance user convenience. The thesis concludes with a general summary of the main results achieved and outlines possible future perspectives.

## **References**

- [1] Alaa, M., et al. (2017). A Review of Smart Home Applications Based on Internet of Things. *Journal of Network and Computer Applications*.
- [2] Balta-Ozkan, N., et al. (2021). Energy Efficiency in Smart Homes: A Cross-Country Comparative Analysis. *Energy Policy*.
- [3] Arduino (2018). Arduino Mega 2560 Rev3 datasheet. *Arduino Documentation*.
- [4] MIT App Inventor. (2023). Official Documentation for Bluetooth-Enabled Apps.
- [5] Pradhan, A., et al. (2021). Voice-Controlled Smart Home Systems: A Usability Study. *International Journal of Human-Computer Interaction*.
- [6] Cook, D. J., & Das, S. K. (2020). *Smart Environments: Technology, Protocols, and Applications*. Wiley.

# *CHAPTER I:*

## *Generalities on Smart Homes*

## **I.1. Introduction**

This chapter covers some generalities on the smart homes, its history, its different advantages, its technology, and the internet of things (IOT).

Smart home products are changing the way we interact with the spaces we live in. Smart homes are gaining in popularity, giving homeowners greater control over their living environment and the convenience and comfort of connected living. Now that there's new technology available and more smart home integration possible, it is easy to understand why so many people are making the switch.

## **I.2. problematic**

Conventional homes encounter multiple problems. These problems lead consumers to seek out smart homes.

Among these problems, security is of utmost concern. Consumers worry about burglaries and fires, as well as accidents occurring at a home. Smart home devices can also deliver an additional convenience through remote monitoring through the use of health monitoring systems for convenience and reliability, for those who are elderly or disabled. The high cost of energy has also prompted the search for a smart home. Smart home devices also save time [1].

## **I.3. History of Smart Home**

The history of smart home products dates all the way back to 1975 at least, to the creation of X10. Compared to traditional home automation, X10 communication technology, which utilizes AC wiring within a home, requires communication between a control module and a compatible device integrated into the house. While innovative for the time, such systems could be expensive and would need to be professionally installed.

Although limited to begin with, X10 was a precursor to controlling many things from a distance. However, it wasn't without issues such as sluggish speed, no security and the loss of signal and interference concerns [2].



**Figure (I.1):** X10 Modules and Remote.

### **I.3.1. Here Comes Wireless**

Moving ahead to the early 2000s, wireless technologies were introduced. Smart home features that were once not possible are now a reality thanks to Zigbee and Z-Wave. DIY enthusiasts were now able to create smart homes since there was no need for a cable to be routed through walls. During this time, there were many competing standards, and a lot of proprietary and incompatible systems. But it was the first step in improving security regarding the smart home. Since their inception, Zigbee and Z-Wave have used encrypted communications [2].

1. **Wi-Fi (Wireless Fidelity):** is a widely adopted wireless technology that allows devices to connect to the internet and communicate within a local network.
2. **Bluetooth:** is a short-range wireless technology that allows devices to communicate wirelessly over short distances.
3. **Zigbee:** is a low-power, wireless communication standard designed for home automation and Internet of Things (IoT) applications.

4. **Z-Wave:** is another wireless technology specifically designed for home automation. It operates on a low-power mesh network and provides reliable communication between devices.



**Figure (I.2):** Wireless communication.

### **I.3.2. Smart Homes Now**

Modern smart home systems allow end users more transparency than ever before. Gone are the days of expensive and inflexible systems. Modern smart homes are all tailored to individual preferences.

Smart devices like Google Home and Amazon Alexa are no longer restricted to turning your lights and appliances. They are integrated with many services. With the help of these systems, you can now stream audio, set the thermostat, get alerts from your home alarm system, and order groceries [2].

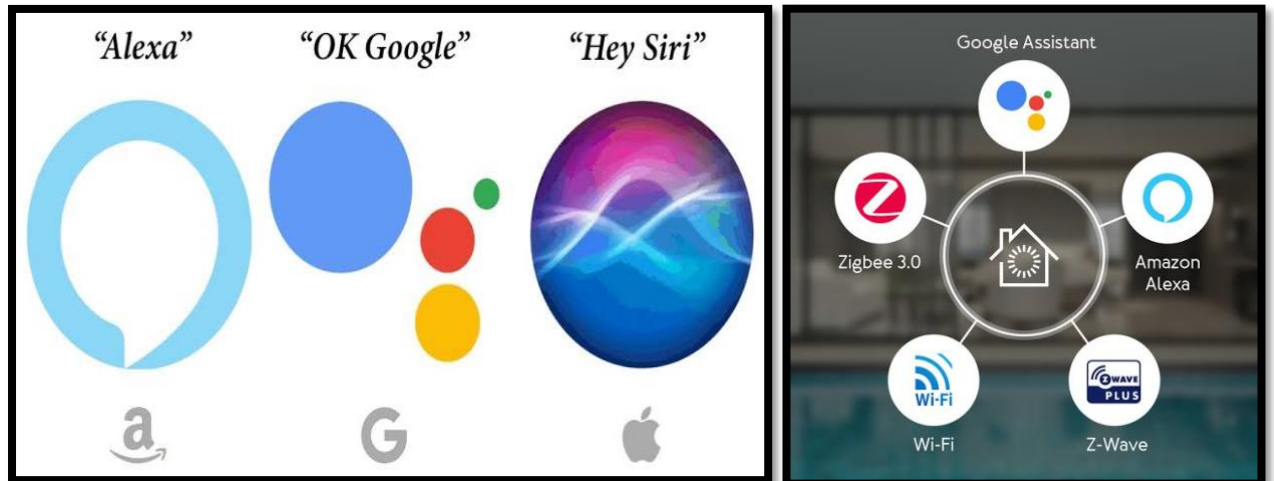


Figure (I.3): Wireless Technology of the Smart Home.

#### I.4. Definition of Smart Home

A smart home is a technology created for the service of residents. It includes residential buildings with smart technologies in order to enhance the comfort level of users (residents) through increased security, health and power saving. The home energy management system offers remote monitoring capabilities through telecommunication technologies, which can control and monitor your smart home appliances with ease [3].

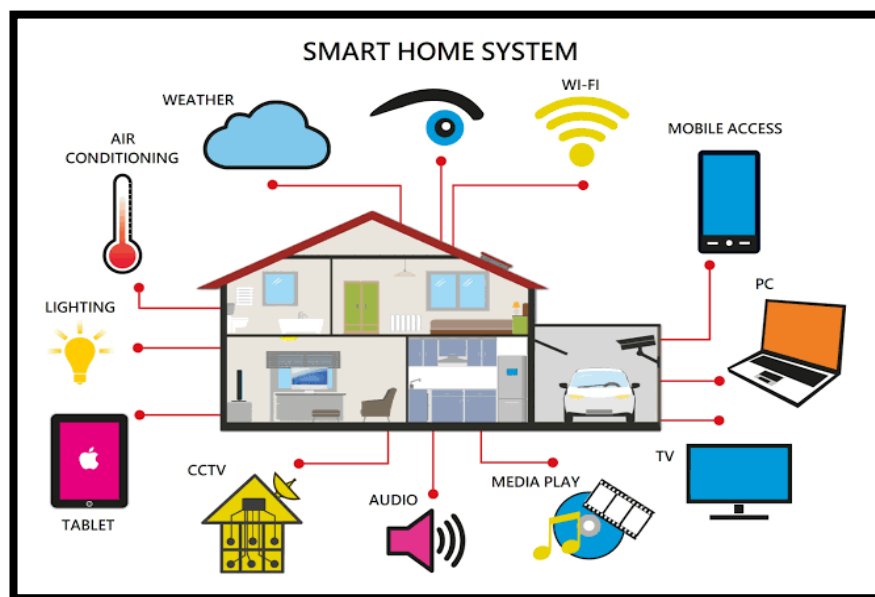


Figure (I.4): Smart Home System.

## **I.5. Benefits of a Smart Home**

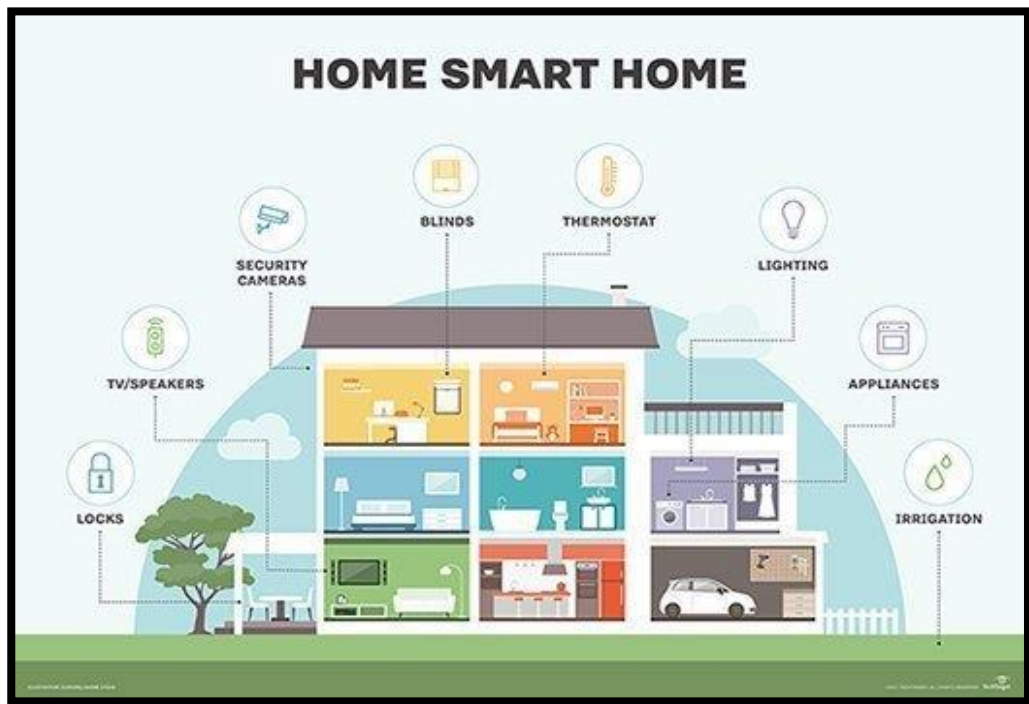
Smart homes provide three main advantages which include convenience features and energy efficiency capabilities and improved security measures. The automated execution of tasks enables residents to save time which simplifies their daily activities. The devices learn about the habits of inhabitants to deliver customized service. A smart thermostat learns the preferred temperature settings of household members to adjust the temperature automatically. Smart home technology helps people save energy. Smart thermostats and smart lights operate based on presence detection to minimize energy waste. The devices offer energy usage data which enables residents to make better decisions about their power consumption [4].

## **I.6. Smart Home Technology Overview**

Smart home technology encompasses a wide range of devices and systems, including:

- **Smart Lighting:** Control lights remotely and automate lighting schedules.
- **Smart Thermostats:** Optimize energy usage and control temperature remotely.
- **Smart Security Systems:** Monitor your home remotely and receive alerts.
- **Smart Locks:** Control access to your home remotely.
- **Smart Appliances:** Control appliances remotely and automate tasks.
- **Smart Entertainment Systems:** Control your home theater and music systems.

- **Voice Assistants:** Control devices and systems using voice commands [5].



**Figure (I.5):** Smart Home Technology.

### I.7. Internet of thing (IOT) in Smart Home

The Internet of Things (IoT) plays an essential role in smart homes because it enables the creation of fully connected intelligent living spaces. Through IoT technology devices and appliances can exchange information with each other and with residents which turns regular houses into automated smart spaces. Here are some key roles that IoT plays in smart homes:

- ❖ **Automation and Control:** The Internet of Things (IoT) enables users to automate and control different home components including lighting systems and heating systems and security systems and entertainment devices.

- ❖ **Remote Access and Monitoring:** The main advantage of IoT in smart homes is the capability to monitor and control devices from any location.



❖ **Figure (I.6):** Remote Home Control and Monitoring.

- ❖ **Energy Efficiency:** IoT technology is useful for dealing with energy usage for a smart home. Sensors, analysis, and automation are great ways to use IOT.
- ❖ **Enhanced Security and Safety:** IoT technology improves the security and safety aspects of a smart home.
- ❖ **Personalized Experiences:** IOT technology allows for a tailored, curated view of the smart home experience. Through observing and learning patterns of behaviour a smart home can be ready to apply to your pattern, upon, day, week, need, etc [6].



- ❖ **Expansion of IoT Ecosystem:** This IoT ecosystem will continue to expand, with more and more devices becoming IoT-enabled in the future. This expansion will cover numerous domains of home living, including health monitoring, kitchen appliances, home entertainment, etc., with the aim of increasing convenience and efficiency.
- ❖ **Data Security and Privacy:** Data security and privacy will be even more important as more devices become connected. Future advancements will consist of stronger encryption mechanisms and authentication methods to protect sensitive user data [6].

## **I.9. Advantages of Smart Home**

### ❖ **Convenience**

The best thing about a smart home appliance is the convenience you get for day-to-day tasks. The ability to control your smart appliances from your phone is an excellent advantage to daily simplicity based on the control you will have.

### ❖ **Energy Efficiency**

Smart appliances have features that aim to conserve our energy. Smart thermostats learn your preferences and will change accordingly while smart lights will turn off when it recognizes there is no one in the room.

❖ Security

A smart home security system typically provides you with more advanced protection than you would typically find in a typical home security system would. A smart security system may offer options for video surveillance, smart locks, and motion sensors.



Figure (I.8): Home Security System.

❖ Accessibility

Smart homes offer a way for people with disabilities to regain freedom in their own homes [7].



Figure (I.9): Accessibility in Smart Home.

## **I.10. Disadvantages of Smart Home**

### **❖ Cost**

When setting up your smart home for the first time, it can be quite costly.

### **❖ Complexity**

Setting up and using smart home tech can be quite complex and intimidating for users who have little experience with smart home technology.

### **❖ Cybersecurity Risk**

As is the case with any connected device there is always a level of cybersecurity vulnerability when using smart home appliances.

### **❖ Privacy Concerns**

Similar to concerns about cybersecurity, some people have issues regarding the data collected from their smart home appliances.

### **❖ Dependency on Technology**

Frustration can arise from dependence on the technology when you experience an internet outage or some other malfunctions [8].

## **I.11. Conclusion**

In this chapter, we addressed the Smart Home, how it started, the technology and systems that it comprises.

A smart home encompasses a combination of technologies and devices which can be controlled remotely or automated to create convenience, energy savings, security and enhanced quality of life.

In addition, the growth in smart homes has been driven by wireless communications advancements promoting wireless communication between devices (Bluetooth, Wi-Fi, Zigbee, Z-Wave) by developing seamless connectivity between the devices.

The Internet of Things (IoT) serves as ubiquitous underlying technology, allowing devices to collect, share and analyze data interacting intelligently with the intention of providing smart home automation and experiences.

Finally, we identified both pros and cons of smart home systems.

## References

- [1] Yahi.Y et Kouri.L, «Contrôle et suivi d'une maison intelligente via internet»,Mémoire de Master, Université Akli Mohand Oulhadj-Bouira, 2018.
- [2] <https://homegenius.sg/blog/brief-history-and-evolution-of-smart-homes-where-it-all-began/>
- [3] Sharif Naser Makhadmeh, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, Syibrah Naim, Ammar Kamal Abasi, Zaid Abdi Alkareem Alyasseri, Optimization methods for power scheduling problems in smart home: Survey. Renewable and Sustainable Energy Reviews, 115 (2019), 109362
- [4] [www.vationventures.com/glossary/smart-home-definition-explanation-and-use-cases](http://www.vationventures.com/glossary/smart-home-definition-explanation-and-use-cases)
- [5] Li, X. et al. (2023). Interoperability in Smart Home Ecosystems: Challenges and Solution. IEEE Internet of Things Journal.
- [6] <https://scand.com/company/blog/internet-of-things-in-smart-home/>
- [7] Zennouche .K et Hadj Ali .Z, «Etude et réalisation d'un système de contrôle pour la sécurité, le confort et l'environnement d'une maison intelligente-Smart house», Mémoire de Master, Université Akli Mohand Oulhadj-Bouira , 2019.
- [8] Djaber .M, «Modélisation et Simulation d'un Système IoT Pour Smart Home», Mémoire de Master, Université 20 Août 1955 de Skikda, 2023.

## *CHAPTER II :*

### *Hardware used in Smart Homes*

## II.1. Introduction

Nowadays, electronic systems are increasingly present in everyday life, but they are now embedded in common objects. These systems are of strategic importance in modern economies. Embedded technology represents the fastest-growing sector of information and communication technologies, as its major advantage lies in simplifying electronic designs by reducing the use of electronic components, thereby lowering the manufacturing cost of these products. The result is more complex and efficient systems in a smaller space.

In this chapter, I will explore the concept of embedded electronics used in my project, which combines the power of programming with the capabilities of electronics. To do this, I will present the various hardware components necessary for the implementation of my home automation project.

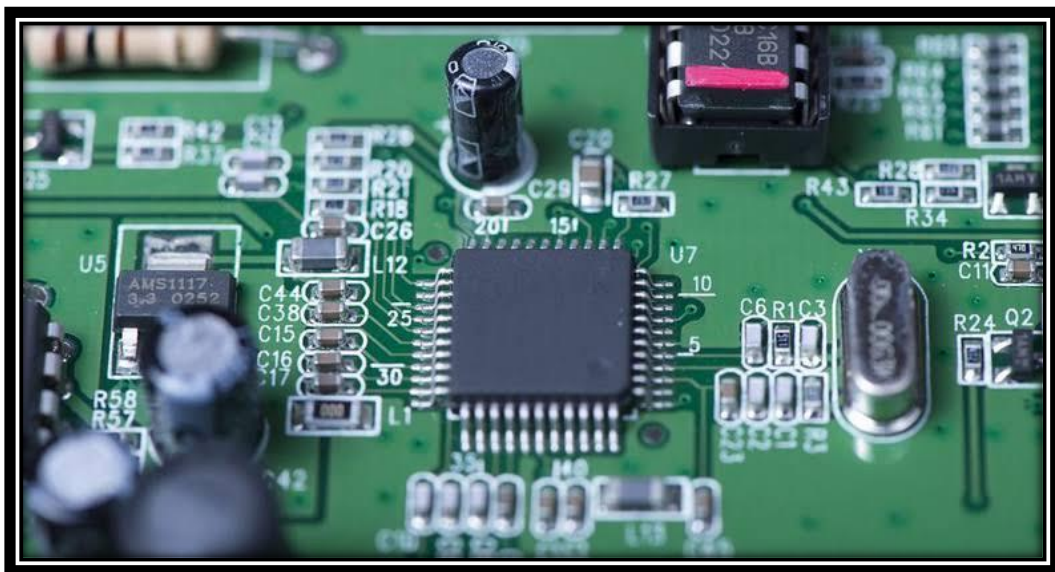


Figure (II.1): A variety of electronic components.

## II.2. Materials used

In this section, I discuss the choice of the programmable board, as well as the different components used to develop my project.

The components are:

- ✚ Arduino Mega 2560 board
- ✚ Gas Sensor Module MQ-05
- ✚ DHT11 Sensor
- ✚ Rain Sensor
- ✚ Flame Sensor
- ✚ HC-05 Bluetooth Module
- ✚ Relay Module 5V
- ✚ Servomotor
- ✚ Fan
- ✚ Buzzer
- ✚ LEDs
- ✚ LCD

### **II.3. Choice of the programmable card**

During my research on programmable boards, I observed that over the past few years, many modular development boards have emerged. Each board has its advantages and disadvantages and differs in terms of functionality, complexity, and price. For my project, Arduino boards provide the embedded computing for my smart home system, as they are sufficient and better suited to my needs. Additionally, their cost-performance ratio is very affordable. These boards have become very popular among the general public and benefit from a very large community.

### **II.4. What is an Arduino?**

Arduino, whose logo is shown in Fig. (II.2) is an open-source electronics prototyping platform, based on both hardware and a set of easy-to-use software tools [1]



**Figure (II.2):** Arduino Logo.

### **II.5. What is an Arduino board?**

An Arduino board is a small electronic board with a microcontroller. The microcontroller allows programming and controlling of actuators based on events perceived by sensors. The 'Arduino' board is a printed circuit which relies on two main pillars: a programmable electronic board (hardware) with a number of components, integrated circuits and peripherals of semiconductor parts; and a programming interface (software) written in a niche languages of programming based upon C/C++, chosen or developed for its capabilities [2].



**Figure (II.3):** Arduino UNO board.

## II.6. Arduino board family

Arduino is not itself a specific type of board or microcontroller. Instead, it refers to an entire family of boards.

Choosing the right one for your project is not a simple task, and it is important to carefully consider which one you will use. We list some of them below, as shown in Figure (II.4).



Figure (II.4): Different types of Arduino boards.

### II.6.1. Arduino UNO board

The Arduino UNO board is based on an ATmega328 microcontroller clocked at 16MHz. Connectors located on the outer edges of the printed circuit board allow for the attachment of a series of complementary modules [3].

### **II.6.2. Arduino Mega 2560 board**

The Arduino Mega 2560 board is based on an ATmega2560 microcontroller clocked at 16 MHz. It features 54 I/O pins, including 14 PWM, 16 analog inputs, and 4 UARTs [3].

### **II.6.3. Arduino LEONARDO board**

The Arduino LEONARDO board is based on an ATmega32u4 microcontroller clocked at 16 MHz, enabling USB port management by a single processor [3].

### **II.6.4. Arduino Micro board**

The Arduino Micro board is based on an ATmega32u4 microcontroller clocked at 16 MHz, allowing USB port control via a single processor [3].

### **II.6.5. Arduino NANO board**

The Arduino Nano board is based on an ATmega328 microcontroller. With 32kB of memory and multiple I/O pins, it is an ideal component for embedded systems or applications requiring multitasking [3].

### **II.6.6. Arduino Due board**

The Arduino DUE board features 54 I/O pins, including 12 PWM, 12 analog inputs, and 4 UARTs. It is ideal for applications requiring more advanced features than the UNO [3].

### **II.6.7. Arduino Esplora board**

The Esplora board is designed like a game controller and comes equipped with sensors and actuators, allowing it to be used directly for testing purposes [3].

### II.6.8. Arduino Yun Mini board

The Arduino Yun Mini board combines an ATmega32u4 (like the Leonardo) and an AtherosAR9331, running a Linino Linux distribution based on Open WRT [3].

### II.7. Arduino Mega 2560 board

Among the types of Arduino mentioned above, I have chosen an Arduino Mega 2560 board, which will serve as the core of my project.

#### II.7.1. Why the Arduino Mega 2560 board?

My choice is justified by the ease of designing and implementing such a board. It is also characterized by simplifying the way of working with microcontrollers while offering several advantages to interested parties, such as:

- Far superior to its little sister, the Arduino Uno, the Mega is a smart choice for projects requiring numerous input/output (I/O) ports and large memory capacity.
- **Reduced Costs:** It is a relatively in expensive card compared to other platforms.
- **Cross-Platform Compatibility:** The Arduino software, written in Java, runs on Windows, Macintosh, and Linux operating systems. Most microcontroller-based systems are limited to Windows.
- **Clear and Simple Programming Environment:** The Arduino programming environment (Arduino IDE) is beginner-friendly while remaining flexible enough for advanced users to benefit from it. Numerous tips, tutorials, and examples are available online (forums, personal websites, etc.), and many libraries with various implemented functions are readily accessible.

- **Open-Source and Extensible Software:** Released under an open-source license, the Arduino software and its language can be expanded by experienced programmers. The programming software for Arduino modules is a cross-platform Java application (running on any operating system), functioning as a code editor and compiler, and capable of uploading programs via serial communication (RS232, Bluetooth, or USB, depending on the module) [1].

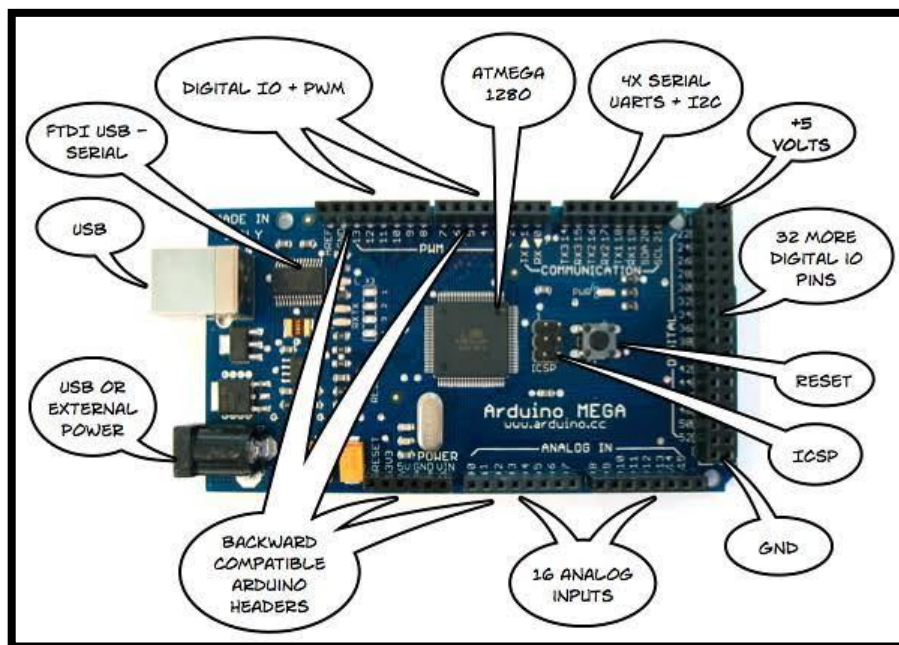


Figure (II.5): Arduino Mega 2560 board.

## II.7.2. Technical specifications of the Arduino Mega 2560 board

The Arduino Mega board features include in the table:

<b>Microcontroller</b>	<b>ATMEGA 2560</b>
<b>Operating voltage</b>	5V
<b>Input voltage (Recommender)</b>	7-12V
<b>Digital I/O Pins</b>	54(15 provide PWM Output)
<b>Analog Input Pins</b>	16
<b>Flash Momery</b>	256 KB
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Clock Speed</b>	16 MHZ
<b>Vin</b>	Positive input voltage
<b>5V</b>	The regulated voltage used to Power the microcontroller and other board components
<b>3.3V</b>	A 3.3V Supply Provided by the board's FTDI integrated circuit
<b>GND</b>	Ground Pin(0V)
<b>Serial Communication</b>	Four serial Ports (0,1,2,3) used for receiving (RX) and transmitting (TX) TTL-Level serial data
<b>SPI (Serial Peripheral Interface)</b>	<ul style="list-style-type: none"> <li>• Pin 50 (MISO)</li> <li>• Pin 51 (MOSI)</li> <li>• Pin 52 (SCK)</li> <li>• Pin 53 (SS)</li> </ul>

	These pins support SPI communication, available via the SPI Library
<b>I2C</b>	<ul style="list-style-type: none"> <li>• Pin 20 (SDA)</li> <li>• Pin 21 (SCL)</li> </ul> These Pins support I2C protocol communication using the Wire/I2C Library

**Table (II.1):** Technical specifications of the Arduino Mega 2560 board [3].

## II.8. What is a sensor?

A sensor is a device, module, machine, or subsystem that monitors changes or events and transmits information to other electronics, which usually include a computer processor. Typically, a sensor detects, measures or recognizes a physical phenomenon and produces a signal that a computer can read, which can then be shown, read or further processed. The figure illustrates how a sensor operates. Many experts and researchers classify sensors in many different ways. In the first classification the sensors can be categorized as Active and Passive. Active sensors require an external excitation or power signal to operate [4].

## II.9. Sensor used

In this project, I used the sensors detailed below, presenting their characteristics as well as their wiring with the Arduino Mega 2560 board.

### II.9.1. Gas Sensor Module MQ-05

- **Presentation**

The MQ5 gas sensor module is useful for detecting gas leaks (in homes and industry). It can detect LPG, natural gas, town gas, etc. Measurements are almost instantaneous because it has a quick response time after preheating. The sensitivity of this module can be adjusted by a potentiometer [5].



Figure (II.6): MQ-05 Gas Sensor Module.

- **Features MQ-05 Gas Sensor**

<b>Operating Voltage</b>	5V DC
<b>Working current</b>	150 mA
<b>Output Signal</b>	Analog and Digital
<b>Operating Humidity</b>	95% RH (rated humidity 65%)
<b>Dimension</b>	32 x 21 x 20 mm
<b>Weight</b>	9 g

Table (II.2): FeaturesMQ-05 Gas Sensor [6].

## II.9.2. DHT11 Temperature and Humidity Sensor

- **Presentation**

The DHT11 module allows for a digital output equivalent to temperature and humidity measured by the sensor. The technology used to produce the DHT11 sensor facilitates high reliability, excellent long-term stability, and a very fast response time.

The DHT11 sensor can measure temperatures from 0 to +50°C with an accuracy of +/- 2°C, and it can measure rates of relative humidity from 20 to 80% with an accuracy of +/- 5%. A measurement can be taken every second. This is a 3-pin module. It requires a very simple communication with the Arduino through one of its inputs (digital input). The other two pins are for its 5V and ground (GND) power supply [7].

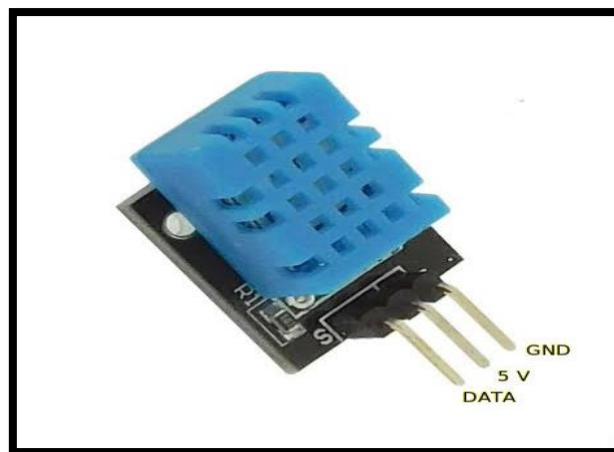


Figure (II.7): DHT11 Temperature and Humidity Sensor.

- **Features DHT11 Sensor**

<b>Operating Voltage</b>	3 to 5V DC
<b>Max consumption</b>	2.5 mA

<b>Measuring Range</b> <ul style="list-style-type: none"><li>• <b>Temperature</b></li><li>• <b>Humidity</b></li></ul>	0 to +50 °C 20 to 90% RH
<b>Accuracy</b> <ul style="list-style-type: none"><li>• <b>Temperature</b></li><li>• <b>Humidity</b></li></ul>	±2 °C ±5% RH
<b>Dimensions</b>	16 × 12 × 7 mm

**Table (II.3):** Features DHT11 Temperature and Humidity Sensor [6].

### II.9.3. Rain Sensor

- **Presentation**

The snow and rain drop detection sensor was designed to detect the presence of both snow and rain drops in the ambient environment. A snow and rain drop detection sensor uses electrical conductivity - or resistance based methods to detect drops of water. It is capable of detecting snow and measuring properties of snow such as intensity of the snow fall, depth of the snow, and the size of the snow flakes as to describe snow-related weather conditions. It is capable of detecting rain drops as well as measuring rainfall intensity - it assists in quantifying rainfall as a function of the amount of precipitation occurring at a given point in time. It is capable of detecting humidity - this could mean measuring absolute moisture content or relative humidity of the surrounding atmosphere [7].

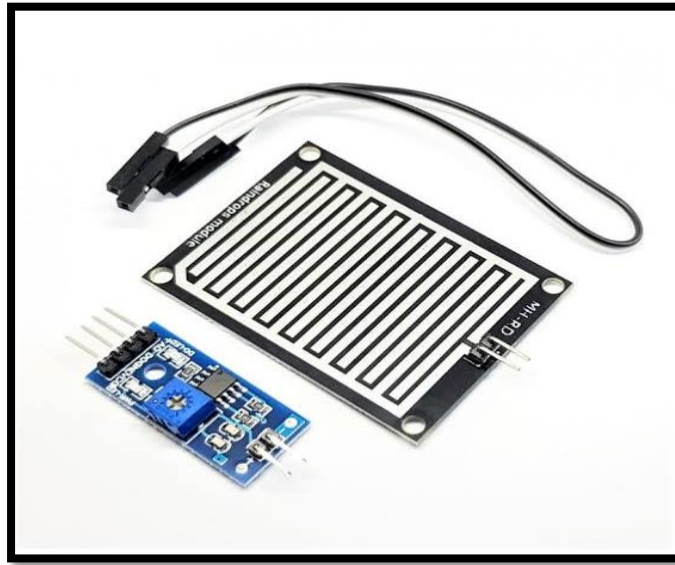


Figure (II.8): Rain Sensor.

- **Features Rain Sensor**

<b>Voltage</b>	3.3V to 5V
<b>Sensor Tracks</b>	Dual-Sided (Sensor is active on bothsides)
<b>Sensor Dimensions</b>	40 x 54 mm

Table (II.4): Features Rain Sensor [6].

#### II.9.4. Flame Sensor

- **Presentation**

The flame sensor can be used to detect a fire source, or other light sources, with wavelengths of 760 nm to 1100 nm. The flame sensor is based on the YG1006, a high speed and very sensitive NPN silicon phototransistor. The flame sensor is sensitive to infrared radiation because of its black epoxy material. The flame sensor in the firefighting robot kit plays a very important role; it can be used as the robot's eyes to find the fire source [6].

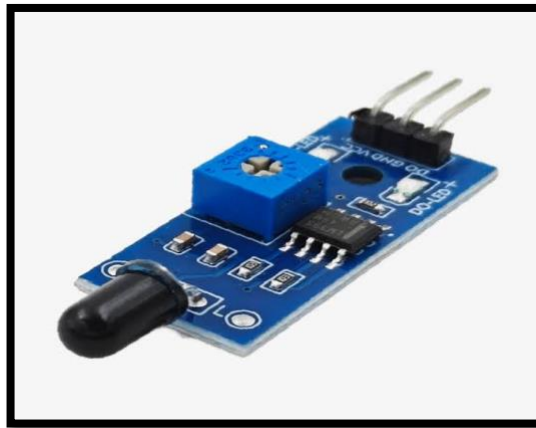


Figure (II.9): Flame Sensor

- **Features Flame Sensor**

<b>Operating Voltage</b>	3.3V to 5V
<b>Detection angle</b>	0 – 60 degree
<b>Operating temperature</b>	-40°C to +85°C
<b>Dimensions</b>	42 x 16 x 15 mm

Table (II.5): Features Flame Sensor [6].

## II.10. Module used

### II.10.1. HC-05 Bluetooth Module

- **Presentation**

The HC-05 module is a Bluetooth SPP (Serial Port Protocol) module, that lets you easily set up a wireless serial connection. The serial port Bluetooth module is a complete Bluetooth V2.0 + EDR (Enhanced Data Rate) 3Mbps modulation with full 2.4GHz radio transceiver and baseband. The HC-05 module uses a CSR Bluecore 04-External single-chip Bluetooth system (in CMOS), with AFH (Adaptive Frequency Hopping function) [7].

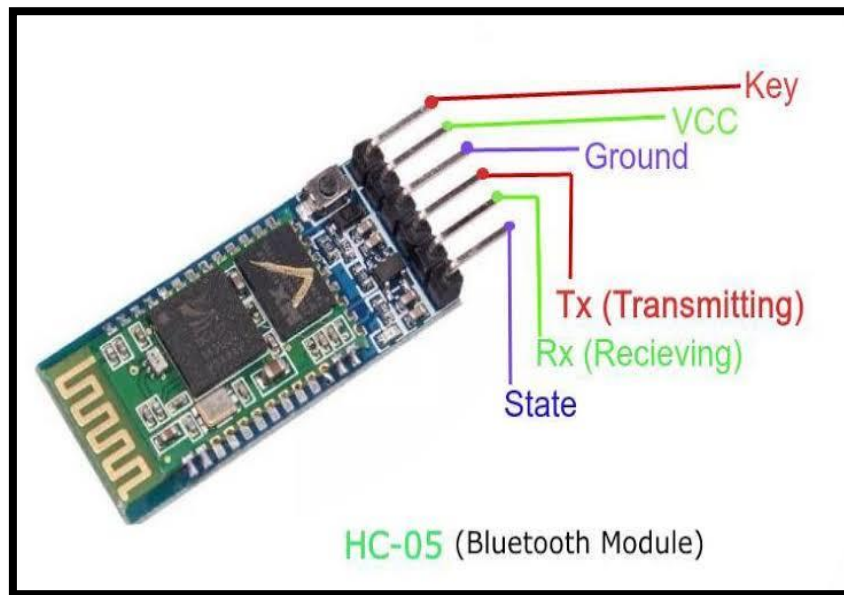


Figure (II.10): HC-05 Bluetooth Module.

- Features HC-05 Bluetooth Module

<b>Operating Voltage</b>	3.3V
<b>Input Voltage</b>	3.3 to 6V
<b>Protocol</b>	V2.0
<b>Dimensions</b>	26.9 mm * 13 mm * 2.2 mm
<b>Current</b>	30 mA during operation, 8 mA to communication
<b>Serial connection</b>	4800 to 1382400 baud
<b>Antenna</b>	Integrated 2.4 GHZ
<b>Regulator</b>	150 mA 3.3V
<b>6-Pin pinout</b>	Vcc , GND, TxD, RxD

Table (II.6): FeaturesHC-05 Bluetooth Module [6].

## II.11. Actuators used

### II.11.1. Servomotor

- **Presentation**

The SG90 9G 180° micro servo motor is a compact servo motor that provides precise motion control in numerous applications. It is able to rotate to a 180-degree angle, which makes it possible to accurately revolve in an arc from 0° to 180°. It is able to correctly position mechanisms in this range of motion. The SG90 180° 9G micro servo motor is a popular component to use in applications which require precise motion control in a small space. It is small, has a limited 180 degree range of rotation, and utilizes PWM control which makes it a useful part for a variety of electronics and robotics project [8].

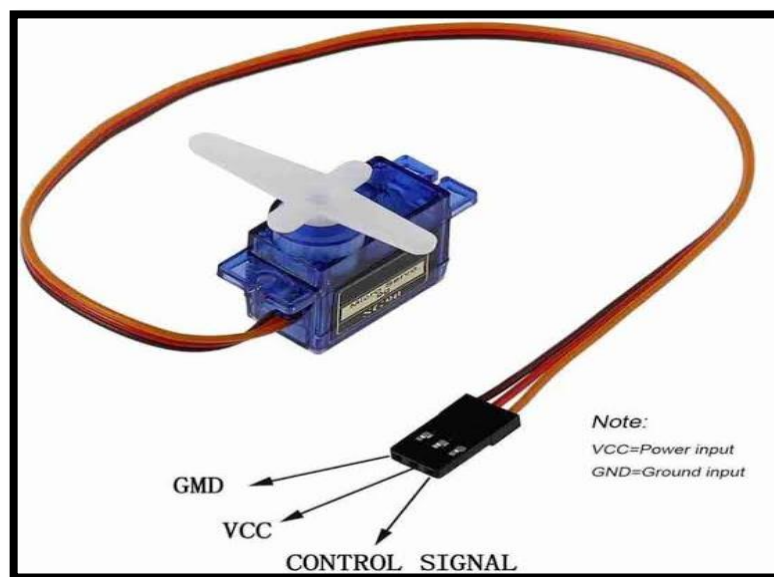


Figure (II.11): Relay Module 5V.

- **Features Servomotor**

<b>Operating Voltage</b>	4.8V to 16V
<b>NoLoad speed</b>	0,12 seconde/ 60° (4,8 V)
<b>Stall torque</b>	1,6 kg/cm (4,8 V)

<b>Operating temperature</b>	- 30 C ° à + 60 C °
<b>End point adjustment</b>	7 microseconds
<b>Operating current</b>	Less than 500 mA
<b>Cable Length</b>	180 mm
<b>Dimensions</b>	23mm x 12.2mm x 29mm

Table (II.7): Features servomotor [6].

### II.11.2. Fan

- **Presentation**

A fan 5V is a fan that operates on 5 volts of direct current (DC). These fans are typically used in electronics projects, PC cases, and small or compact devices to provide cooling in a quiet and efficient manner. Fan 5v fans are usually powered by energy sources including USB ports or 5v battery sources [7].



Figure (II.12): Fan 5V.

- **Features Fan**

<b>Operating Voltage</b>	5V DC
<b>Rotation speed</b>	4200 rpm
<b>Current consumption</b>	0.20 A
<b>Dimensions</b>	60 x 4060 x 20 mm

**Table (II.8):** Features Fan [6].

### II.11.3. Buzzer

- **Presentation**

The 5V, 8mm active continuous-beep buzzer is an electronic sound device that produces continuous sounds when voltage is supplied between 2 and 5 volts. With a diameter of 8mm, the device is small and compact for applications with size constraints. The buzzer has a very special sound quality due to its advanced design and can be useful on a number of projects when an audible warning is needed [8].



**Figure (II.13):** Buzzer 5V.

- **Features Buzzer**

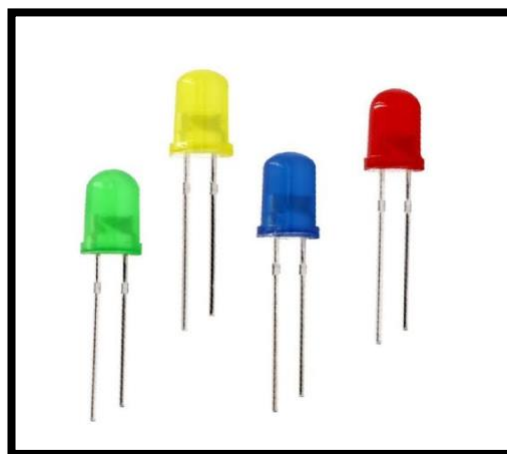
<b>Operating voltage</b>	2V to 5V DC
<b>Maximum current consumption</b>	30 mA
<b>Resonance Frequency</b>	$3.1 \pm 0.5$ kHz
<b>Resonant frequency</b>	$\sim 2300$ Hz

**Table (II.9):** Features Buzzer [6].

#### II.11.4. LEDs

- **Presentation**

A light-emitting diode (LED) is a device made of semiconductor material that emits light when electric current runs through it. Inside the semiconductor, electrons recombine with electron holes releasing energy as photons, which emitted light/emitted electromagnetic radiation. The color of light emitted is associated with the energy of the emitted photons determined by how much energy the electron lost to traverse the band gap of the semiconductor [9].



**Figure (II.14):** LEDs.

- **Features LEDs**

<b>Standard Round Directivityof</b>	5 mm
<b>Proxy-resistant epoxy</b>	UV
<b>Forward current (IF)</b>	30 mA
<b>Forward Voltage (VF)</b>	1.8V to 2.4V
<b>Reverse Voltage</b>	5V
<b>Operating temperature</b>	-30 ° à + 85 °
<b>Storage temperature</b>	-40 ° à + 100 °
<b>Luminous intensity</b>	20 mcd

**Table (II.10):** Features LEDs [6]

### II.11.5. LCD (Liquid crystal display)

- **Presentation**

An LCD (liquid crystal display) consists of two layers of polarizing material with a liquid crystal solution sandwiched in between. When electrical current flows through the solution, it generates an image on the screen [8].



**Figure (II.15):** LCD.

- **Features LCD**

<b>Operating Voltage</b>	5V
<b>Interface</b>	I2C
<b>PCB SIZE</b>	60mm * 99mm

**Table (II.11):** Features LCD [6].

## II.12. Conclusion

In this chapter, we explored the embedded electronics element of my smart home project. Determining the hardware components was in the plan for getting the system to operate correctly, which led to choosing the Arduino Mega 2560 as the main control board due to its more capable processing, as well as all of the I/O pins associated with it.

To enhance safety and automation at home, we integrated multiple sensors into the system, including a gas sensor (MQ-5), flame sensor, rain sensor, and DHT11 for temperature and humidity.

Wireless communication was possible using the HC-05 Bluetooth module to communicate between the Arduino and whatever else we decide to integrate for remote monitoring and controlling.

We also included several actuators, including LEDs, a buzzer, cooling fan, and a servo motor that could respond to the sensor data, which would certainly enhance the safety and convenience associated with home automation.

Overall, with the integration of the various sensors, wireless communication, and actuators, the results show the potential of embedded electronics as a means for enhancing home automation, safety, and convenience.

## References

- [1] <https://www.arduino.cc/en/Guide/Introduction/>
- [2] Krama Abdelbasset et Gougui Abdelmoumen, « Etude et réalisation d'une carte de contrôle par Arduino via le système Android », Mémoire de Master, Université Ouargla, Algérie. 2015.
- [3] [https://en.wikipedia.org/wiki/List\\_of\\_Arduino\\_boards\\_and\\_compatible\\_systems](https://en.wikipedia.org/wiki/List_of_Arduino_boards_and_compatible_systems)
- [4] Mohd Javaid, Abid Haleem, Shanay Rab, Ravi Pratap, Singh, Rajiv Suman, Significance of machine learning in healthcare: Features, pillars and applications, International Journal of Intelligent Networks, 3 (2022).
- [5] Boudjedir Imen, « Un système embarqué pour la détection des gaz dangereux », Mémoire de Master, Université LARBI Ben M'Hidi d'Oum El Bouagui, Algérie. 2017.
- [6] <https://youpilab.com/components/product-category/capteurs>
- [7] Reha, A., Ounayn, H., Kellili, M., El Abdi, N., Ismaili, O., Satar, M., ... & Goucheq, A. (2019, June). « Conception et réalisation d'une maison intelligente ».
- [8] Geddes, M. (2016). Arduino Project handbook: 25 practical projects to get you started (Vol. 2). No Starch Press.
- [9] Edwards, Kimberly D. Light Emitting Diodes (PDF). University of California, Irvine. p.2. Archived from the original (PDF) on February 14, (2019).

## *CHAPTER III :*

### *Software used in Smart Homes*

### III.1. Introduction

In this chapter, we will explore the concept of embedded electronics used in my project, which combines the power of programming with the capabilities of electronics. To do this, we will present the various software necessary for the implementation of my home automation project.

### III.2. Arduino programming platform

The Arduino Integrated Development Environment - or Arduino Software (IDE) - has a text editor to write code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with it. Figure (III.1) included the software Arduino IDE logo [1].

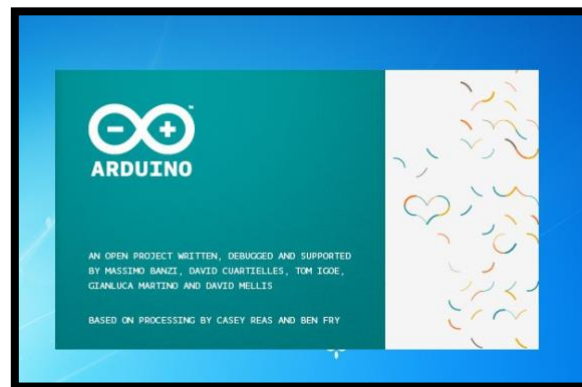


Figure (III.1): Logo of Software Arduino IDE.

### III.3. Presentation of the Software

The Arduino IDE interface is quite simple; it offers 4 areas, each of which has a very specific functionality.

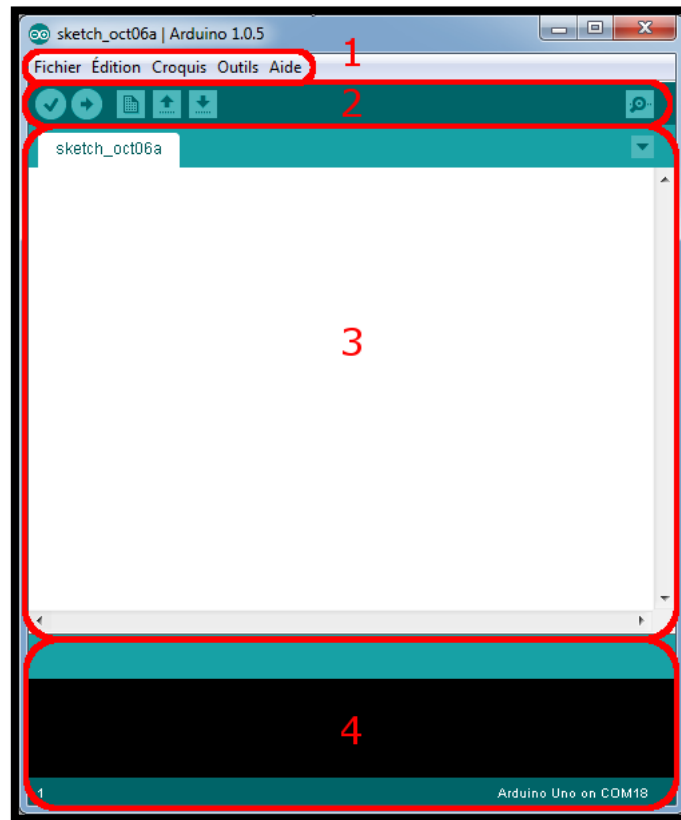
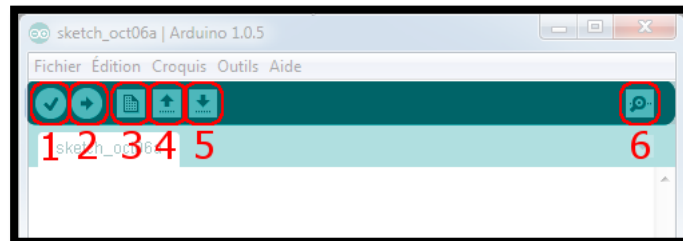


Figure (III.2): The Arduino IDE interface in detail.

- **A menu bar :** these are the Software configuration options
- **A toolbar (icon bar):** it contains the buttons that will serve us when we are going to program our cards
- **An editing area:** this block will contain the program we are going to create
- **An information and status bar:** this important, because it will help us correct mistakes in our program. It is the debugger [2].

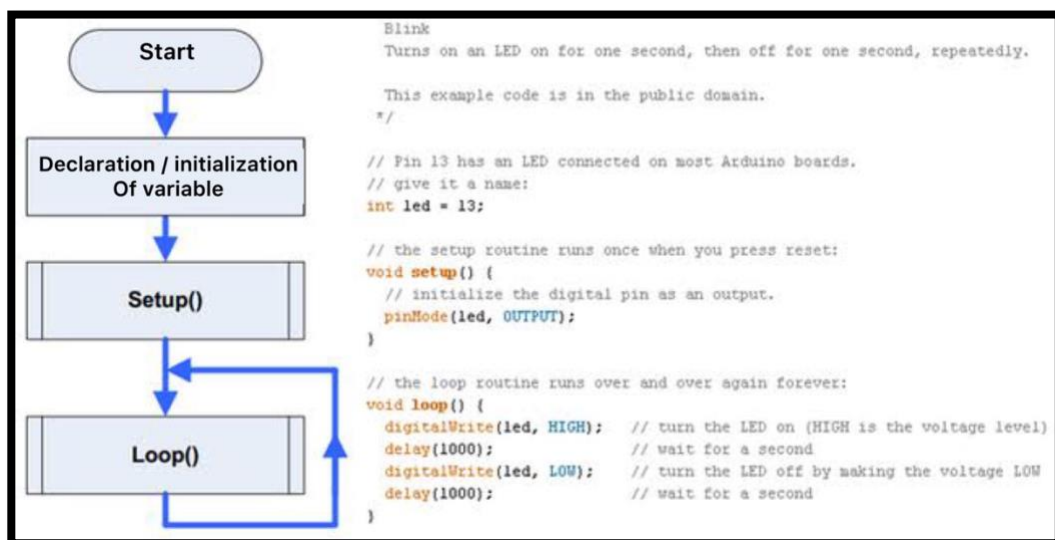
### III.4. Buttons

Let's now see what the buttons are for, framed in red and numbered



**Figure (III.3):** The toolbar.

- **Button 1:** This button allows you to check the program, it activates a module that looks for errors in your program
- **Button 2:** Load (download) the program into the Arduino board.
- **Button 3:** Create a new file.
- **Button 4:** Open a file.
- **Button 5:** Save the file.
- **Button 6:** Search [2].



**Figure (III.4):** General Structure of an Arduino sketch.

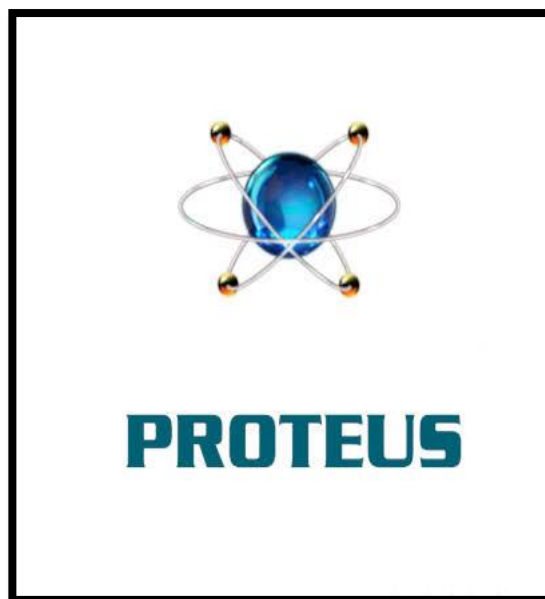
## III.5. Simulation platform

### III.5.1. Proteus Software

- **Definition**

Proteus Professional is an electronics software suite developed by Labcenter Electronics. The software that is used in Proteus Professional allows CAD (Computer-Aided Design) for electronics. It also consists of two pieces of software (ISIS, ARES). This software suite is widely recognized and acknowledged in the electronics industry and is used by many companies, training institutions (high schools and universities). Now, just because it is a well-known piece of software does not mean that it is the right piece of software for you, but it is worth exploring, Proteus Professional offers several advantages:

- A package that is simple and fast to learn and use
- Efficient technical support
- A virtual prototyping tool that contributes to lower hardware and software costs when developing project [3].



**Figure (III.5):** Logo proteus software.

### III.5.2. ISIS

- **Definition**

The ISIS program, in Proteus Professional, is mostly used for editing electrical schematics. However, the software also provides a means of simulation of the electrical schematic, which can identify some specific errors during the design phase. The documentation used for the electrical circuits that are ultimately created with this software can also be used indirectly, as the software provides the user control over almost all graphical aspects of the electrical circuits [4].



**Figure (III.6):** Logo isis.

- **Presentation**

The main area of the screen is called the Editing Window, which functions like a drawing window. This is where you want to place and wire your components. In this window you will find toolbars that include: tools menu, design tools, and command tools [4]. Figure (III.7) is principal window of ISIS:

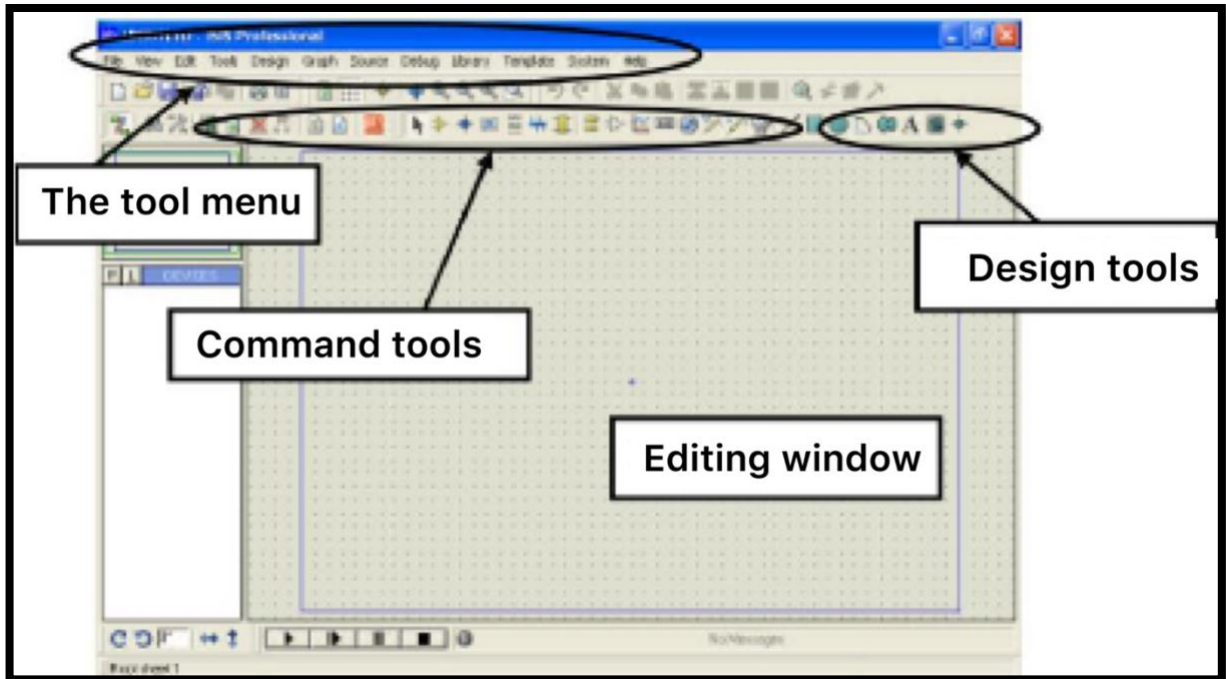


Figure (III.7): Principal window of ISIS.

### III.5.3. ARES

- **Definition**

ARES software is a design and routing tool which works well with ISIS. An electrical schematic made with ISIS can be imported into ARES to design the PCB (Printed Circuit Board) for the electronic board easily. Many engineers still prefer to work out the layout of a printed circuit by hand, but the ARES software provides automated component placement and routing [4].



Figure (III.8): Logo ARES.

- **Presentation**

The general appearance of ARES is similar to that of ISIS, as shown in Fig. (III.9). It includes:

- A menu bar.
- A main window where you can create your routing and also design new components.
- A preview window in the top-right corner.
- A palette below, along with a package selector.
- An active layer selector at the bottom left [4].

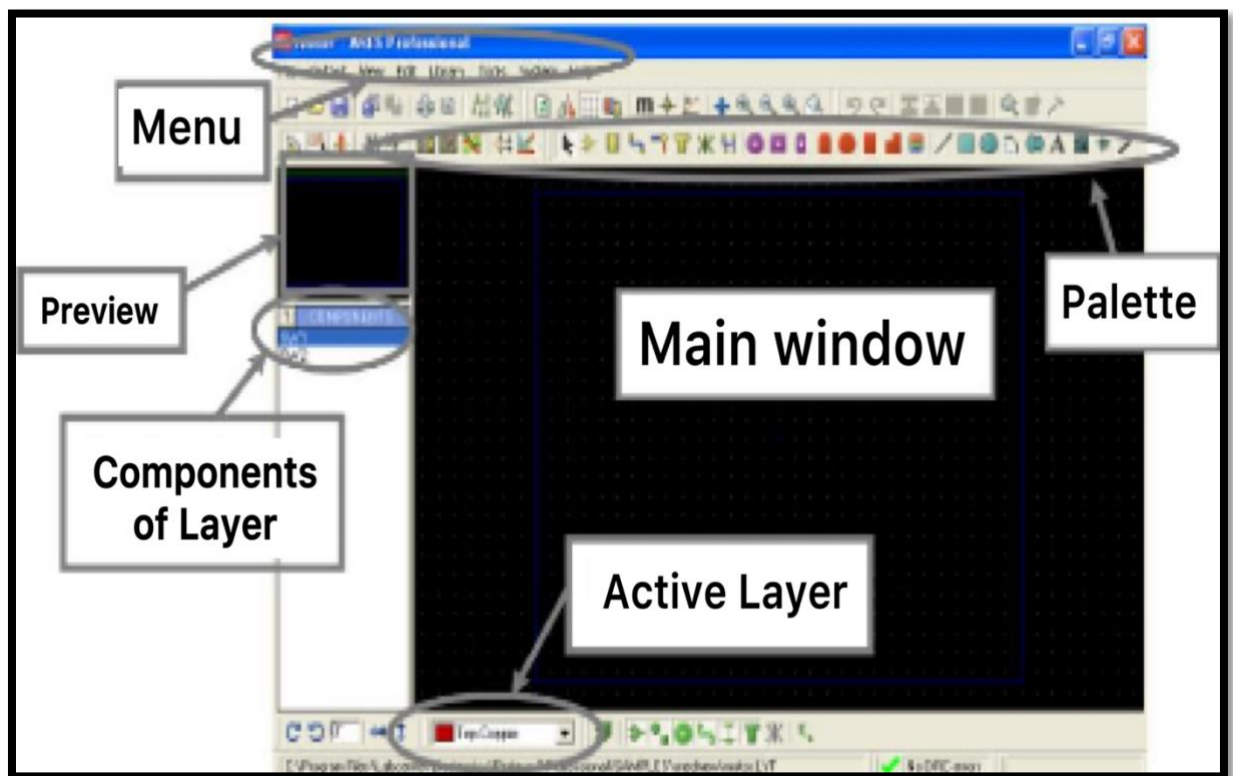


Figure (III.9): Principal window of ARES.

## III.6. Development of the Arduino application

### III.6.1. Android

- **Definition**

Android is a mobile equipment operating system that is largely a modified version of Linux. It was originally prepared by a startup called Android, Inc. In 2005 Google purchased Android and assumed responsibility for all Android development (and all developers also) as part of their strategy to penetrate the mobile space [5].



**Figure (III.10):** Android Logo.

- **Key Features of Android Include:**
  1. Open Source
  2. Google Play Store
  3. User Interface
  4. Customization
  5. Multitasking & Google Integration
  6. Updates [5].

### III.6.2. MIT App Inventor

- **Definition**

It is an online development tool, they can create applications for Android devices.

This tool was created by Google in 2009 and is currently maintained by the Massachusetts Institute of Technology (MIT) [6].



**Figure (III.11):** MIT APP INVENTOR Logo.

#### III.6.2.1. What we can program with App Inventor

1. Game
2. Questionnaires (Quiz)
3. Application using the components of a smartphone [7].

#### III.6.2.2. Why MIT App Inventor

I chose to work with App Inventor for the following reasons:

- It is a well-documented software.
- Quick to learn for non-specialists.
- A simple and efficient environment.
- No programming language to learn, so no risk of syntax errors.
- Free, open-source, and cross-platform software.

### III.6.2.3. Structure of an MIT App Inventor

The App Inventor IDE consists of two interfaces:

1. Design interface.
2. Blocks editor [7].

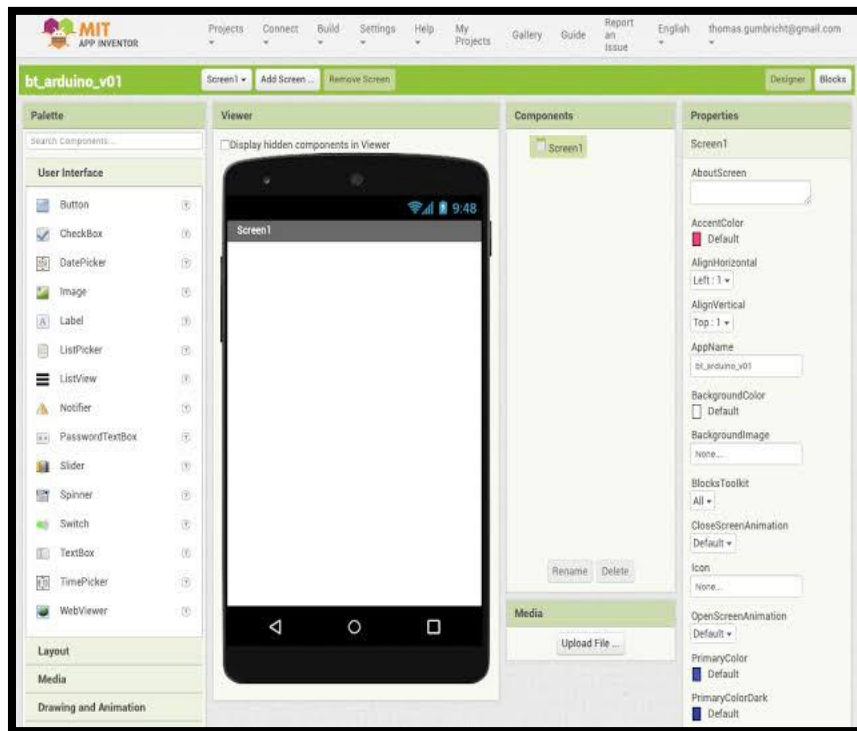


Figure (III.12): MIT APP INVENTOR Interface.

#### ➤ Design interface

You'll be presented with the Designer tab as shown in the following figure (III.13).

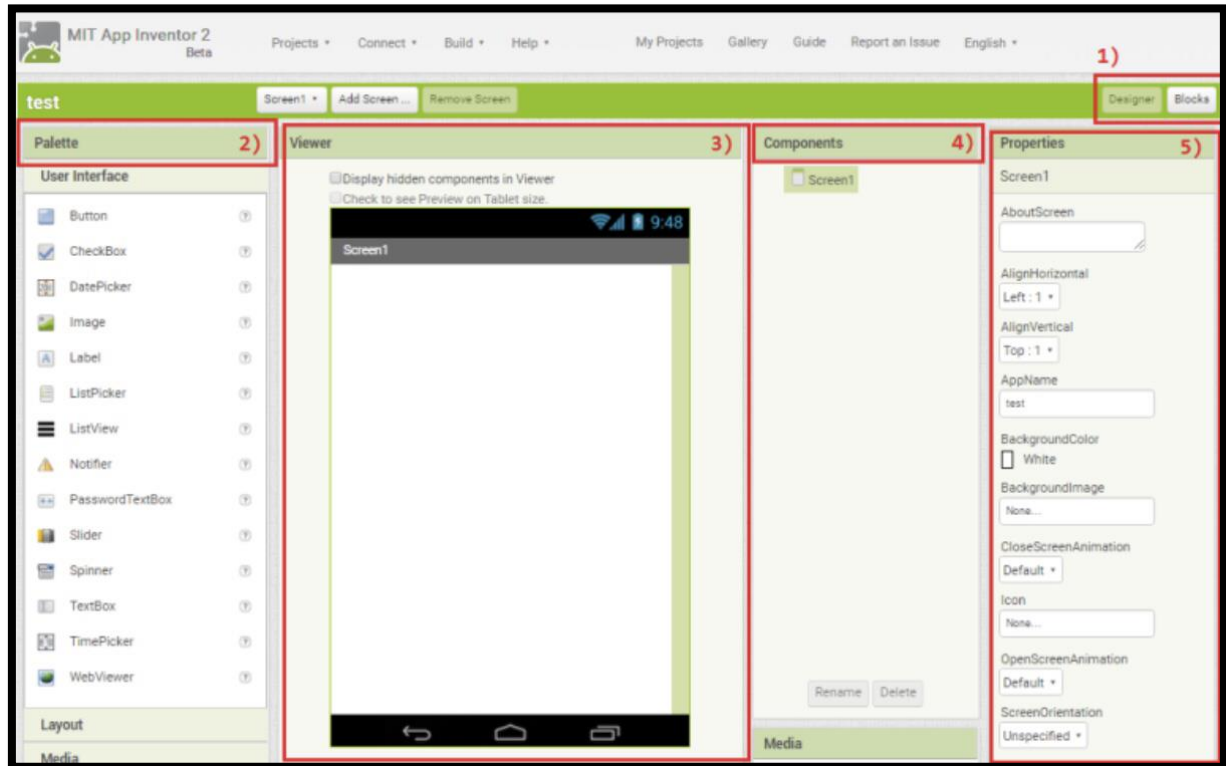


Figure (III.13): Designer tab.

- 1) You select whether you are on **the Designer** or in **the Blocks Editor** tab. With MIT App Inventor you have 2 main sections: Designer and Blocks. **The designer** gives you the ability to add buttons, add text, add screens and edit the overall app look. **The Blocks** section allows you to create custom functionality for your app, so when you press the buttons it actually does something with that event.
- 2) **The Palette:** contains the components to build the app design like buttons, sliders, images, labels, etc...
- 3) **The Viewer:** This is where you drag the components to build the app look.
- 4) **Components:** You can see all the components added to your app and how they are organized hierarchically.

- 5) **Properties:** This is where you select your components' properties like color, size and orientation [7].

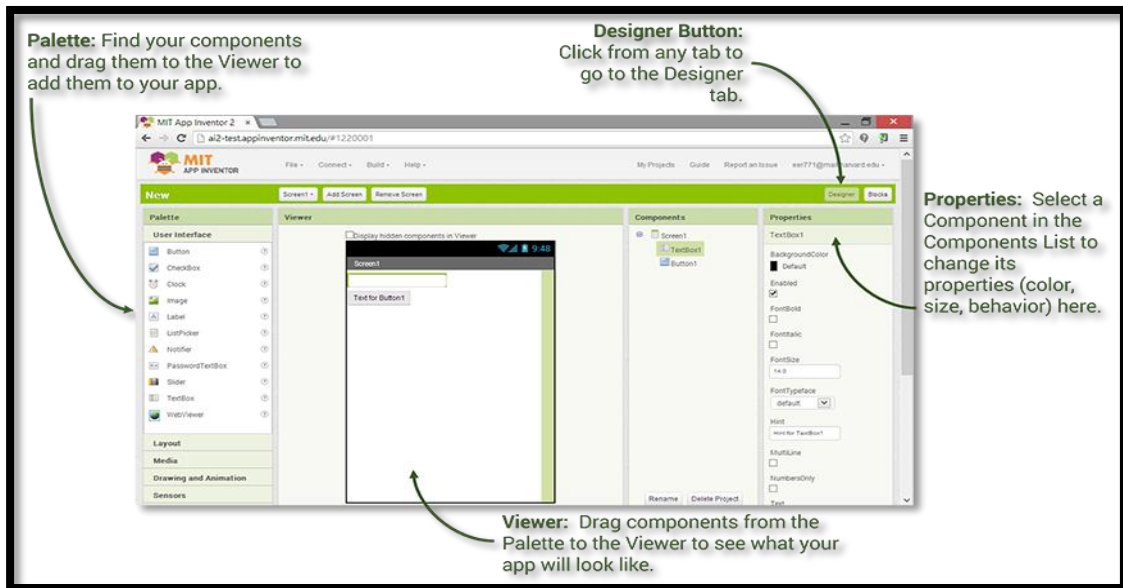


Figure (III.14): Design Interface.

➤ **Blocks Editor**

Open the **Blocks** editor tab. In the **Blocks** editor tab, you have several sections:

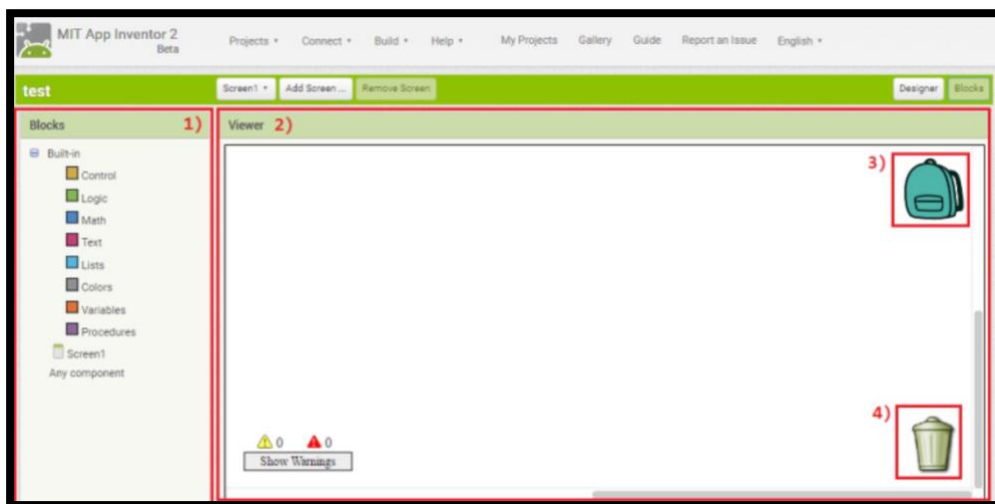
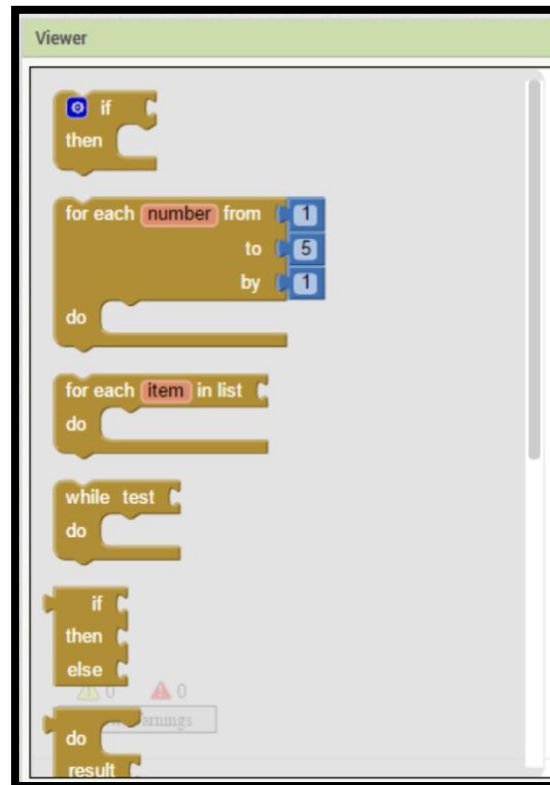


Figure (III.15): Blocks editor tab.

1) Contains the built-in blocks for making the app logic. This is what makes the app define the buttons actions, communicate with Arduino, connect to Bluetooth module, etc. You have multiple blocks grouped into categories [7]:

➤ **Control:** *if/else* statements, *while loops*, etc...



**Figure (III.16):** Control of Blocks editor.

- **Logic:** *True, False, equal, not equal*, etc...
- **Math:** math operators.
- **Text:** blocks that deal with text.
- **Lists:** blocks for handling lists.
- **Colors:** blocks to handle colors, like choosing a color, make color and split colors.
- **Variables:** initialize variables, setting variables values, get variables values, etc...

- **Procedures:** procedures are like functions. A procedure is a sequence of code blocks with a given name. Later, you can call that sequence of blocks instead of creating the same long sequence.

2) **Viewer:** you drag the blocks and join them in a specific way to make something happen.

We recommend that you navigate inside the blocks section and explore what's inside. The blocks look like puzzle pieces that fit into each other or not. If you can't do something with certain blocks, they won't fit.

3) **Backpack:** you save code blocks to use later.

4) **Dustbin:** to delete them [7].

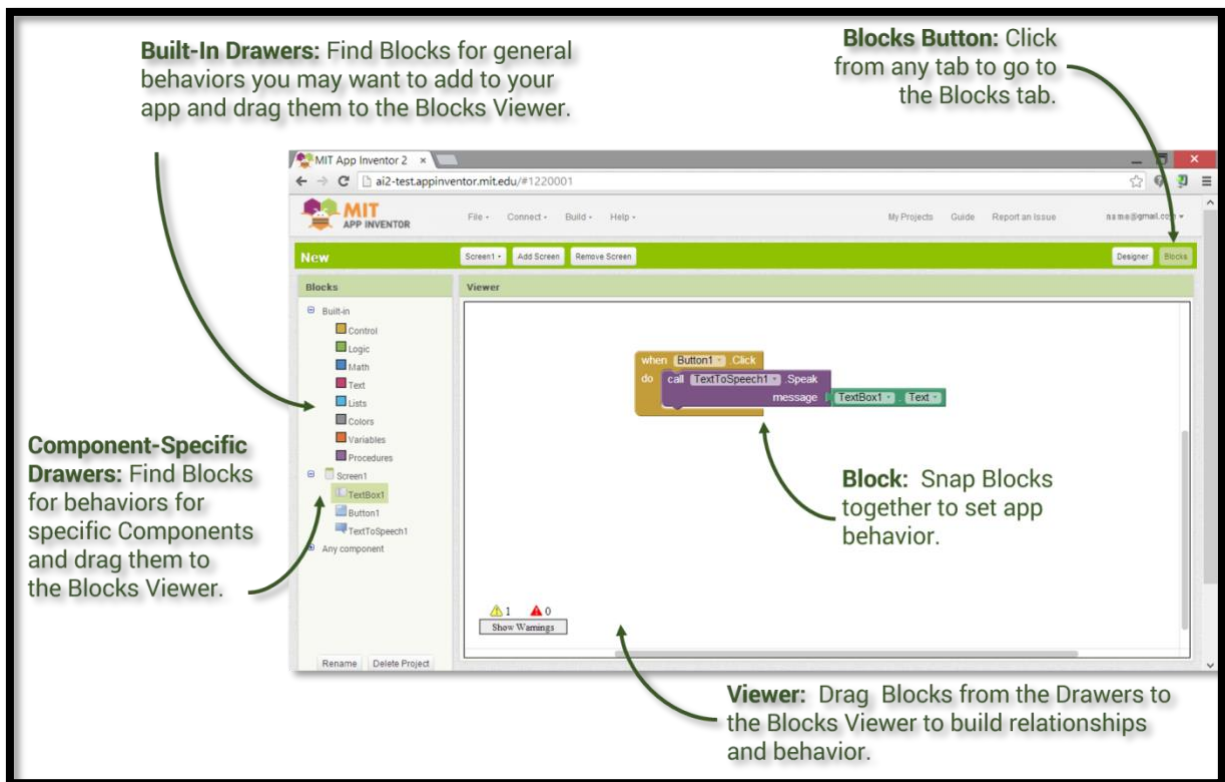


Figure (III.17): Blocks Editor.

### **III.7. Conclusion**

In this chapter, we examined the software components of a smart home system, with a focus on essential tools such as the Arduino IDE for microcontroller programming, Proteus Simulation (ISIS and ARES) for designing and testing circuits, and an Android application developed using MIT App Inventor for remote control. Together, these tools facilitate efficient system development, simulation, and user interaction within the smart home environment.

Their integration results in a functional, user-friendly, and scalable automation system, highlighting the importance of combining hardware programming, simulation tools, and mobile interfaces in modern IoT solutions.

Looking ahead, future improvements may include AI-driven automation, cloud-based services, and expanded device compatibility to create smarter, more responsive home management systems.

## References

- [1] Monk, S. (2021). Programming Arduino: Getting Started with Sketches (2<sup>e</sup> éd.). McGraw Hill.
- [2] Belmeliani .A et Rahiel .N, « Conception et réalisation d'un prototype d'une maison intelligente », Mémoire de Master, Université d'Oran 2 Mohamed Ben Ahmed, 2020.
- [3] Sadi .A, « Conception et simulation d'une carte d'acquisition de données météorologique en temps réels à base d'un microcontrôleur 18F45K22 », Mémoire de Master, Université Mouloud Mammeri Tizi-Ouzou, 2018.
- [4] Cours de proteusprofessional (isis et ares). University Badji Mokhtar d'Annaba.
- [5] Wei-Meng Lee, "BEGINNING ANDROID™ 4 APPLICATION DEVELOPMENT ", Ch1 , John Wiley & Sons , 2012.
- [6] Mile Inguel .M, « Conception et réalisation d'un système domotique par GSM», Mémoire de Master, Université Mouloud Mammeri Tizi-Ouzou, 2017.
- [7] <https://info789.jimdofree.com/cours/9%C3%A8me/app-inventor/>

# *CHAPTER IV:*

## *Smart Home Prototype*

*Development: From System Design  
to Mobile Application Integration*

## **IV.1. Introduction**

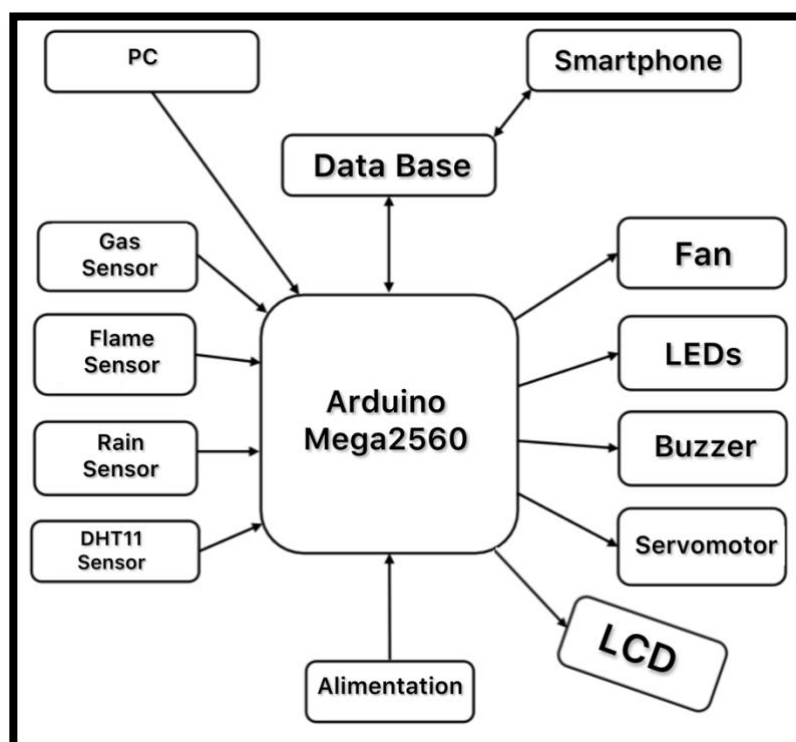
In this chapter, we will focus on the development of a mobile application dedicated to the real-time control and monitoring of a smart home system. This application will serve as a central interface, allowing users to interact with various components of the smart home, such as lights, temperature sensors, door locks, and other connected devices, from their smartphone.

To bring this project to life, we will integrate both hardware and software elements, creating a functional prototype that reflects the core features of an intelligent home. The system will rely on real-time sensors to collect data and respond dynamically to user commands or environmental changes. These sensors will transmit information to a microcontroller, which will communicate with the mobile app through wireless protocols such as Bluetooth or Wi-Fi.

This chapter will cover the design and development of the mobile interface, the implementation of sensor communication, and the integration of voice control features to enhance user convenience. The ultimate goal is to demonstrate a seamless interaction between the physical system and the digital application, showcasing the potential of IoT in modern home automation.

## **IV.2. General synoptic diagram**

Figure IV.1 shows a block diagram of the prototype.



**Figure (IV.1):** General synoptic diagram.

### IV.3. Specification and analysis of requirements

This phase involves understanding the system's context. It aims to determine the functionalities and the actors, as well as to identify the use cases.

#### IV.3.1. Functional Requirements and No-Functional Requirements

The system to be developed must meet and fulfill the user's needs. In this regard, the functional and no-functional requirements are presented as follows:

##### IV.3.1.1. Functional Requirements

A functional requirement relates to functions, meaning what the system will do. It thus concerns the system's functionalities, which translate into requirements specifying a clear input/output behavior of the system.

In our case, the functional services are as follows:

- Gas leak detection.
- An indoor lighting system.
- A ventilation system for temperature regulation.
- Regularly monitoring and checking internal temperature and humidity.
- Controlling home access.
- Flame detection.
- Rain detection.

#### **IV.3.1.2. No-Functional Requirements**

No-functional requirements define a system's characteristics in terms of performance, robustness; hardware used, or design approach. The system must meet the following qualities:

- Processing speed.
- Performance.
- Ease of use.
- Real-Time response.

#### **IV.4. Overall system architecture**

In our project, we used a simple system architecture that meets both the functional and non-functional requirements mentioned earlier. The diagram in Fig (IV.2) illustrates the overall architecture of our system, which consists of three main parts:

##### **IV.4.1. User Part**

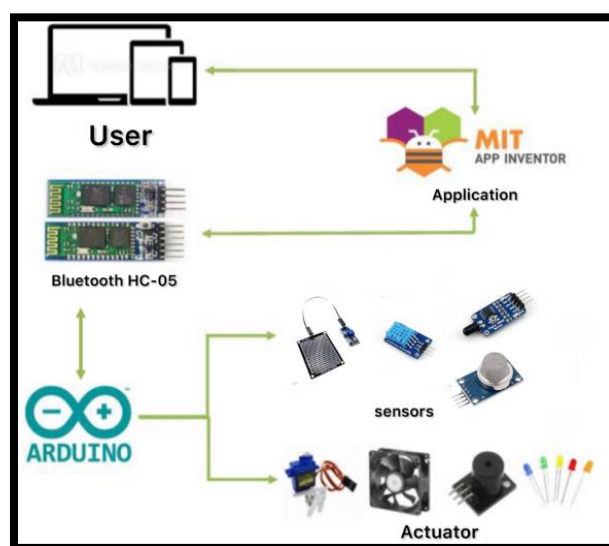
The user is the one who can control the system via multiple devices such as a smartphone, tablet, or PC using a Bluetooth module connected to the Arduino. We can control and monitor the system's status via an Android application (App Inventor).

### IV.4.2. Arduino and software Part

This is the most important part of the system, where the Arduino serves as the main unit responsible for processing data and commands received from the sensors and the user via an application. In this case, the Arduino transmits the commands to the actuators for real-time execution.

### IV.4.3. Hardware Part

This part consists of all the equipment used to control our system (sensors, actuators, and communication means such as cables and Bluetooth, which are connected to the Arduino board).



**Figure (IV.2):** Overall system architecture.

### IV.5. Simulation

In a smart home simulation, all aspects of daily life are managed to ensure comfort, efficiency, and safety. Environmental conditions in the house are tracked by sensors located around the home that adjust the lighting, temperature, and air quality automatically. Household appliances are linked to the network and all can be remote control via a mobile app.

## IV.5.1. Simulation Result

### IV.5.1.1. Gas leak Detection

To simulate gas detection, use an Arduino board along with a gas sensor, a buzzer for alarms, an LEDs, and an LCD to display the result ('Gas' or 'No Gas').

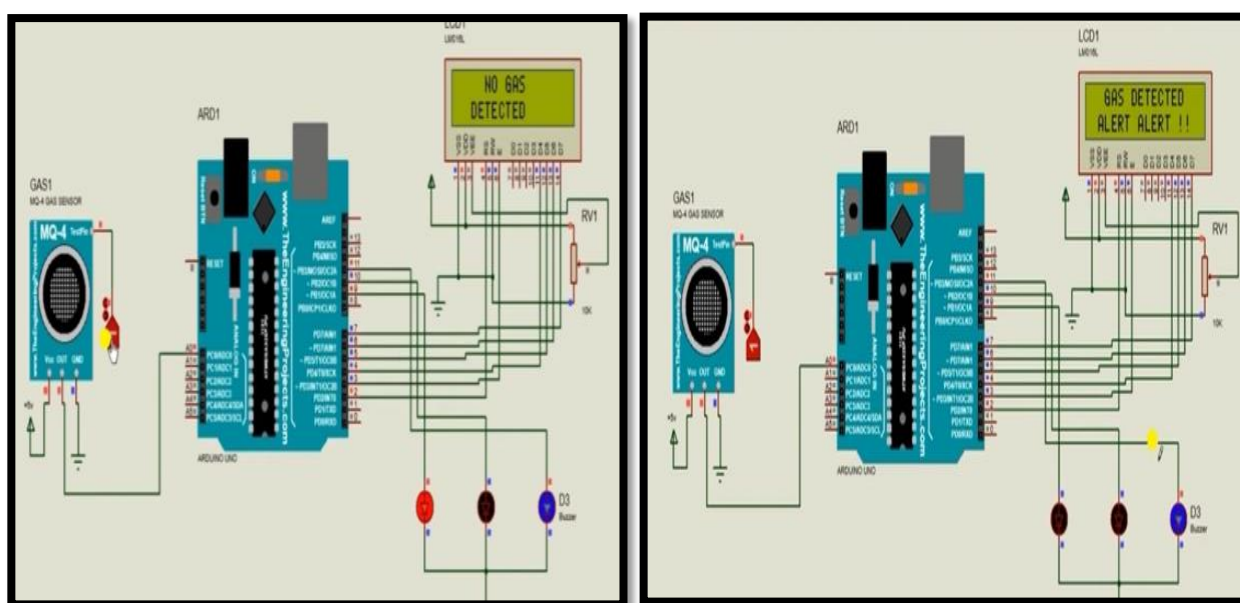


Figure (IV.3): Gas Sensor detection.

- **Comments**

This simulation demonstrates a basic gas detection system where a sensor sends data to an Arduino, which in turn triggers a visual alert (LCD and LEDs) and an audible alarm (buzzer) when gas levels exceed a certain threshold.

### IV.5.1.1.1. Flowchart For the Gas Sensor

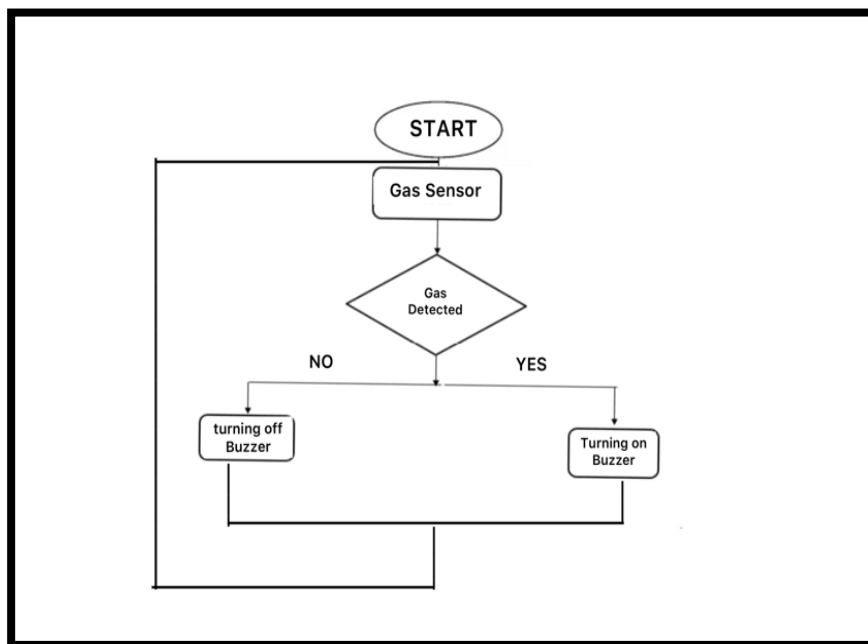


Figure (IV.4): Flowchart for the Gas Sensor management.

### IV.5.1.2. Flame detection

To simulate Flame detection, use an Arduino board along with a Flame sensor, a buzzer for alarm and an LCD to display the result ('Flame Detected').

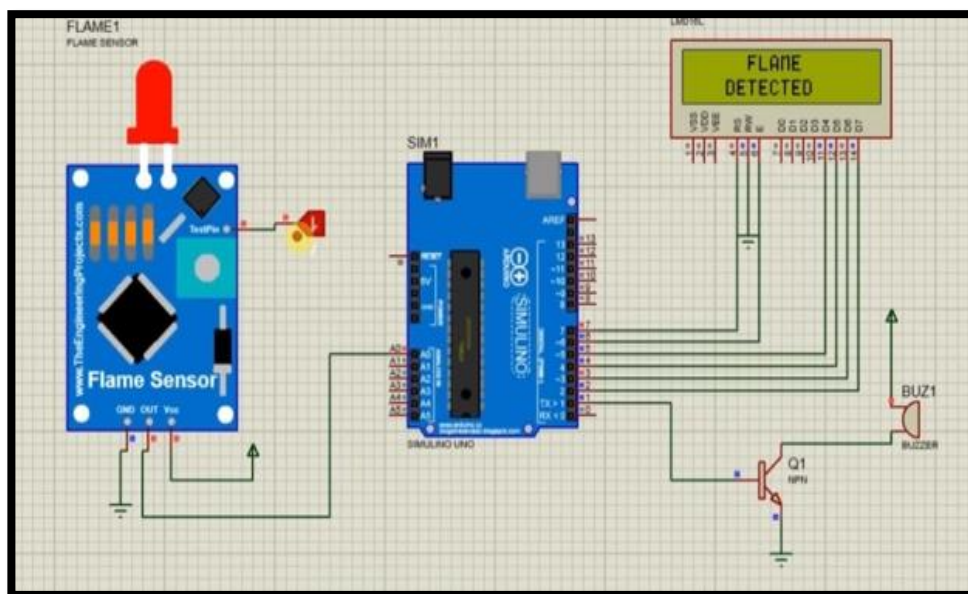
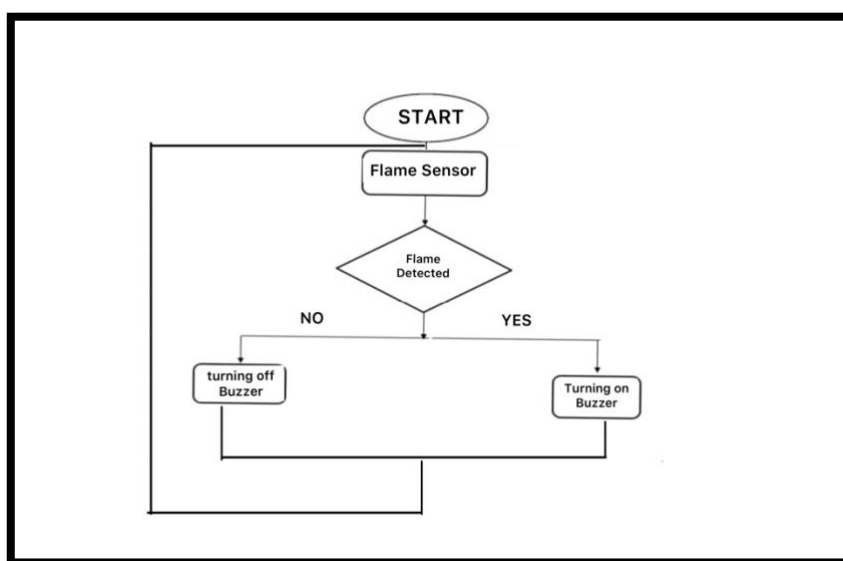


Figure (IV.5): Flame Sensor detection.

- **Comments**

This simulation demonstrates a simple flame detection system where a sensor detects the presence of a flame and signals an Arduino, which then reacts by displaying a message on an LCD screen and activating a buzzer to alert the user.

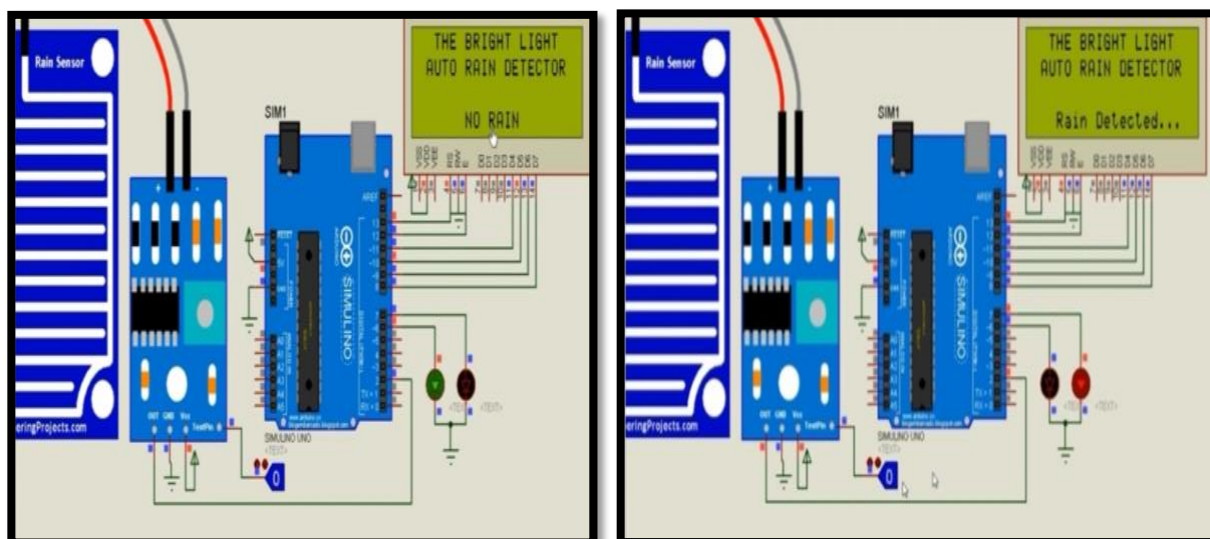
#### IV.5.1.2.1. Flowchart For the Flame Sensor



**Figure (IV.6):** Flowchart For the Flame Sensor management.

#### IV.5.1.3. Rain detection

To simulate rain detection, use an Arduino board along with a rain sensor, 2 LEDs for on/off, an LCD to display the result ('Rain detected' or 'No Rain').

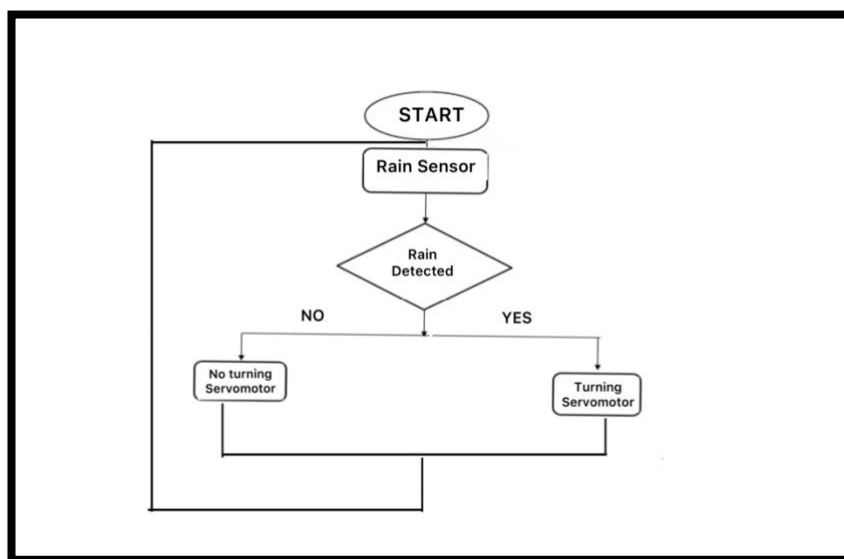


**Figure (IV.7):** Rain sensor detection.

- **Comments**

This simulation illustrates the operation of a basic rain detector where a sensor sends data to an Arduino, which then uses two different-colored LEDs and an LCD screen to visually indicate whether rain is detected or not. The green LED signals no rain, while the red LED and the message on the LCD screen indicate rain detection.

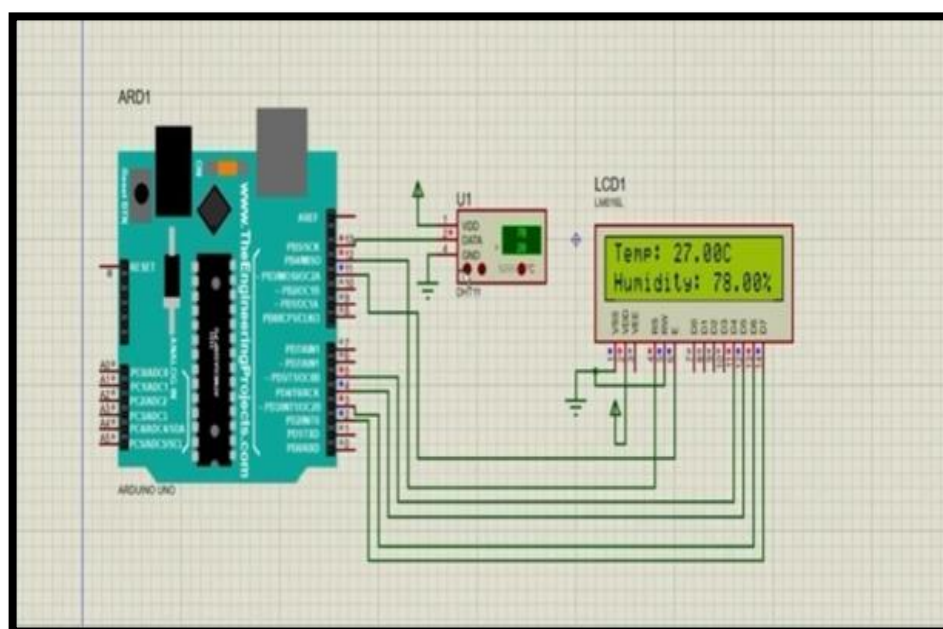
#### IV.5.1.3.1. Flowchart For the Rain Sensor



**Figure (IV.8):** Flowchart For the Rain sensor management.

#### IV.5.1.4. DHT11 Sensor (Temperature/Humidity)

To simulate Temperature and Humidity, use an Arduino board along with a DHT11 sensor, an LCD to display the result ('Temperature /Humidity').

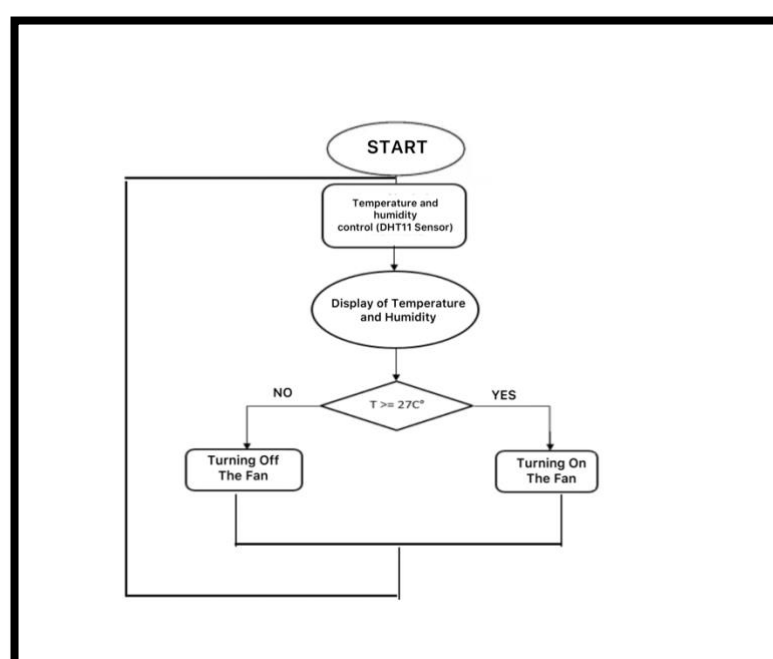


**Figure (IV.9):** DHT11 Sensor.

- **Comments**

This simulation demonstrates how an Arduino can read temperature and humidity data from a DHT11 sensor and display it in a readable format on an LCD screen. It is a fundamental example of environmental data acquisition and presentation using a microcontroller.

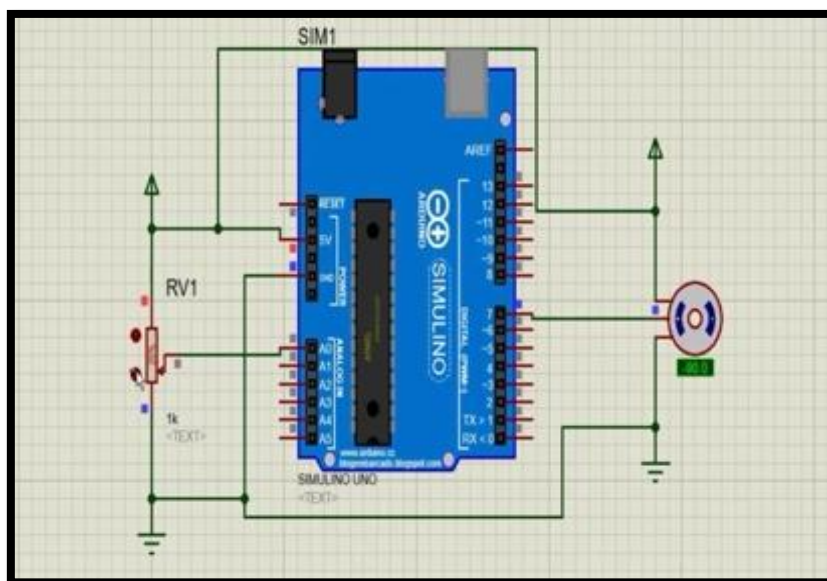
#### IV.5.1.4.1. Flowchart for the DHT11 Sensor (Temperature/Humidity)



**Figure (IV.10):** Flowchart For the DHT11 Sensor.

#### IV.5.1.5. Opening /closing the door (servomotor)

To simulate opening or closing the door, use an Arduino board along with a Servomotor.



**Figure (IV.11):** Servomotor.

- **Comments**

This simulation demonstrates how an Arduino can be used to control the angle of a servo motor by reading input from a potentiometer. The Arduino reads the potentiometer's position and generates an appropriate PWM signal to adjust the servo motor's angle.

#### **IV.5.1.5.1. Flowchart for the Opening /closing the door (servomotor)**

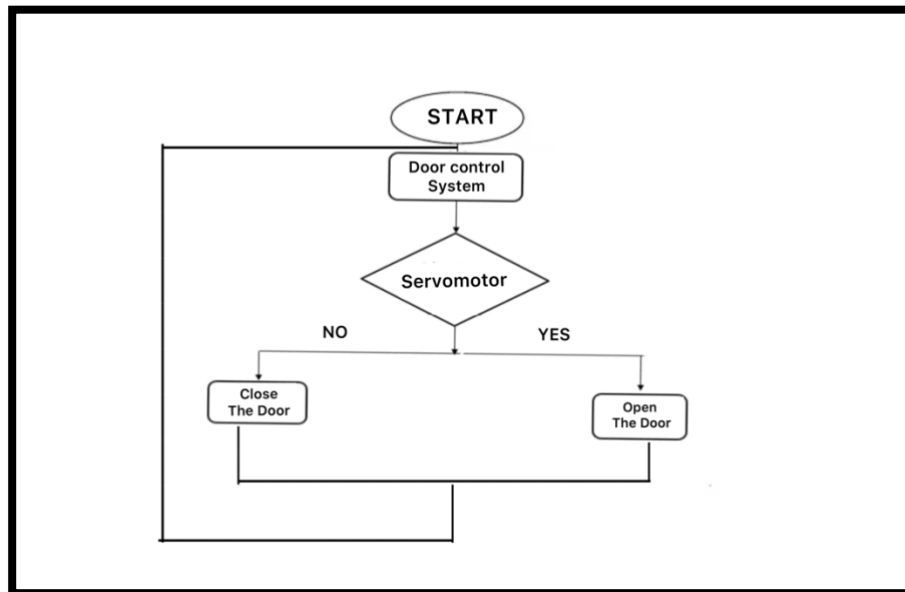


Figure (IV.12): Flowchart For the Servomotor.

#### IV.5.1.6. An indoor lighting system (LEDs)

To simulate an indoor lighting system, use an Arduino board along with 4 LEDs.

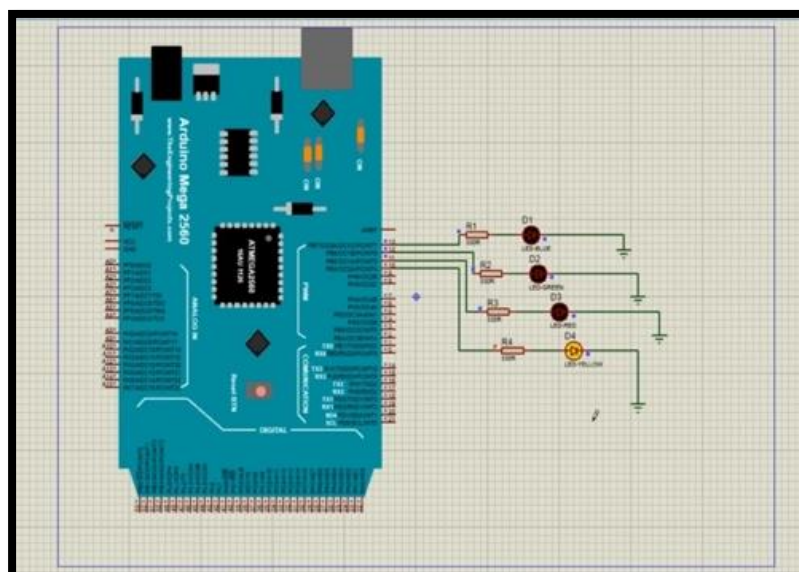
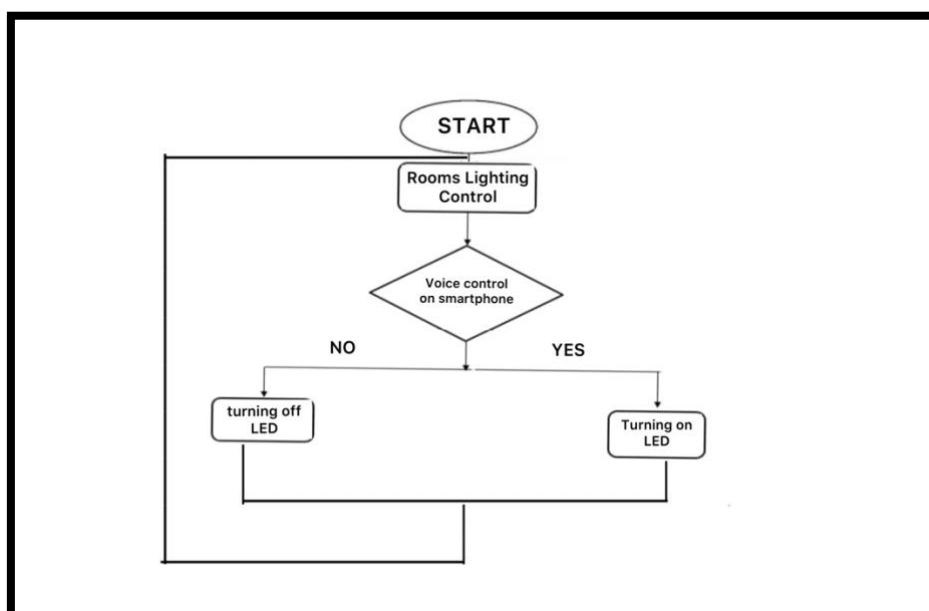


Figure (IV.13): Rooms lighting control.

- **Comments**

This simulation represents a basic indoor lighting system where an Arduino Mega 2560 board is used to control the activation of four LEDs. The program on the Arduino determines the behavior of the LEDs, and resistors are used to limit the current and protect the components.

#### IV.5.1.6.1. Flowchart For the indoor lighting system (LEDs)



**Figure (IV.14):** Flowchart for the rooms lighting control.

## IV.6. Design and implementation part

### IV.6.1. Mobile application Design

I will create an application that will allow us to give voice commands to control lighting and door opening. I will use the MIT App Inventor platform to control these devices. Once the application is created, I will download it from the Play Store onto my phone

and then launch it. Next, I will scan the QR code displayed on the platform to link it to application.



Figure (IV.15): Voice recognition.

#### IV.6.2. application interface

This app uses the built-in voice recognition feature of Android mobile devices to transmit voice commands to smart home.

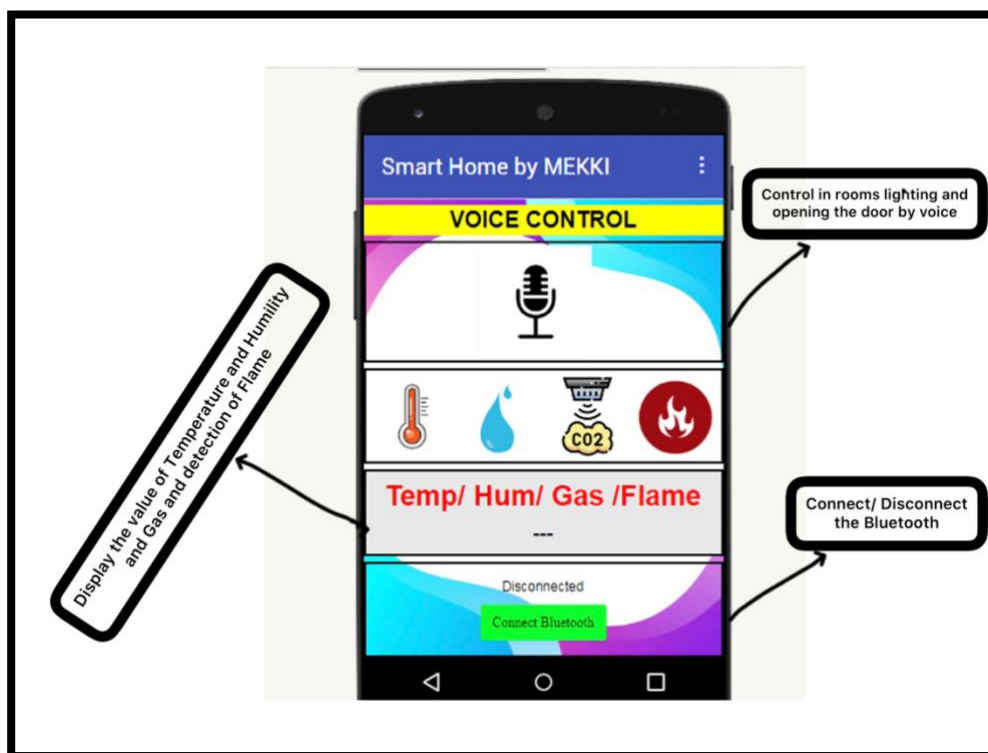


Figure (IV.16): Application interface.

### IV.6.2.1. App Control by voice

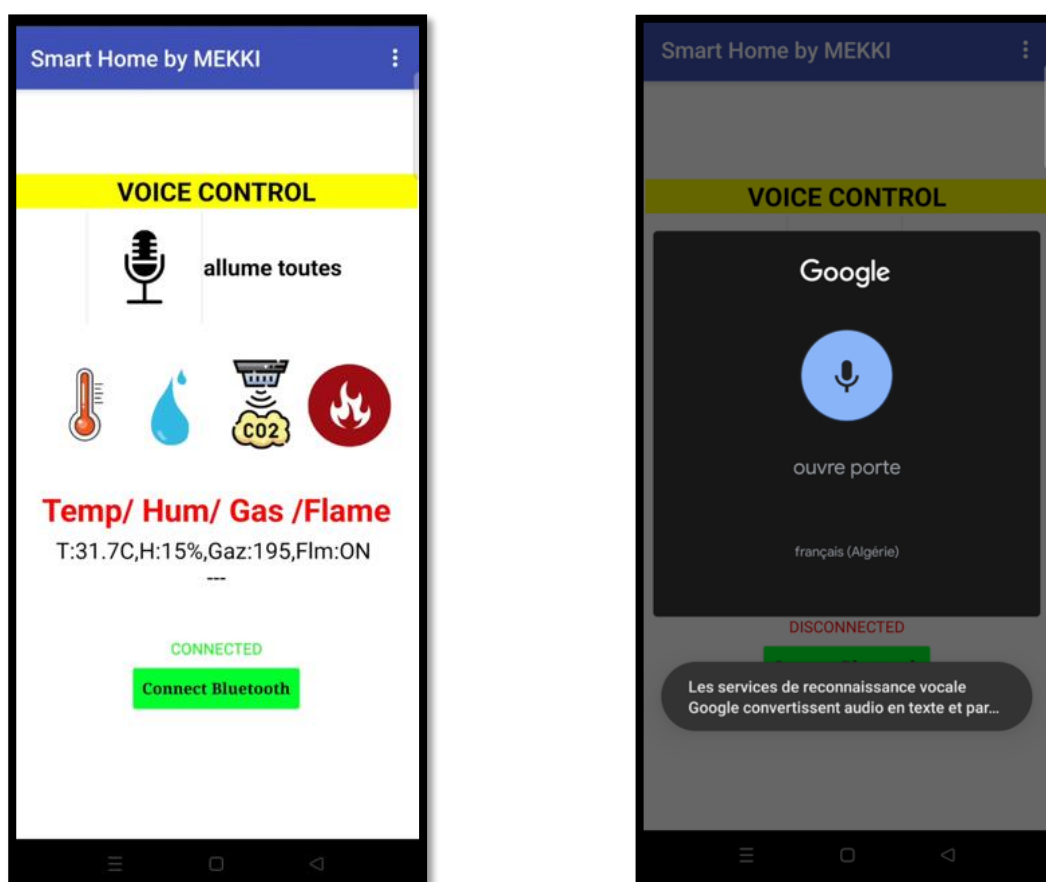
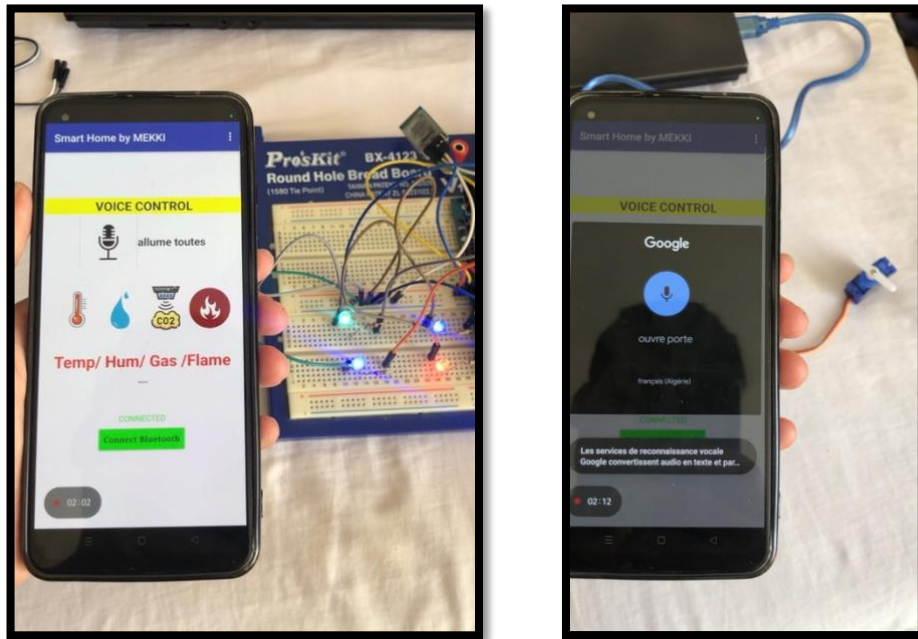


Figure (IV.17): voice control and monitoring the value of sensors in real-time.

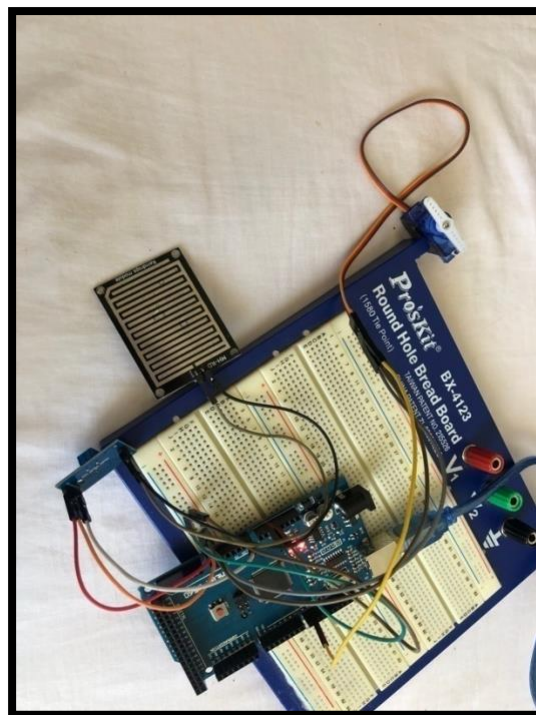
### IV.6.2.2. Application Results

To control the ignition and extinguishing by voice, I say:

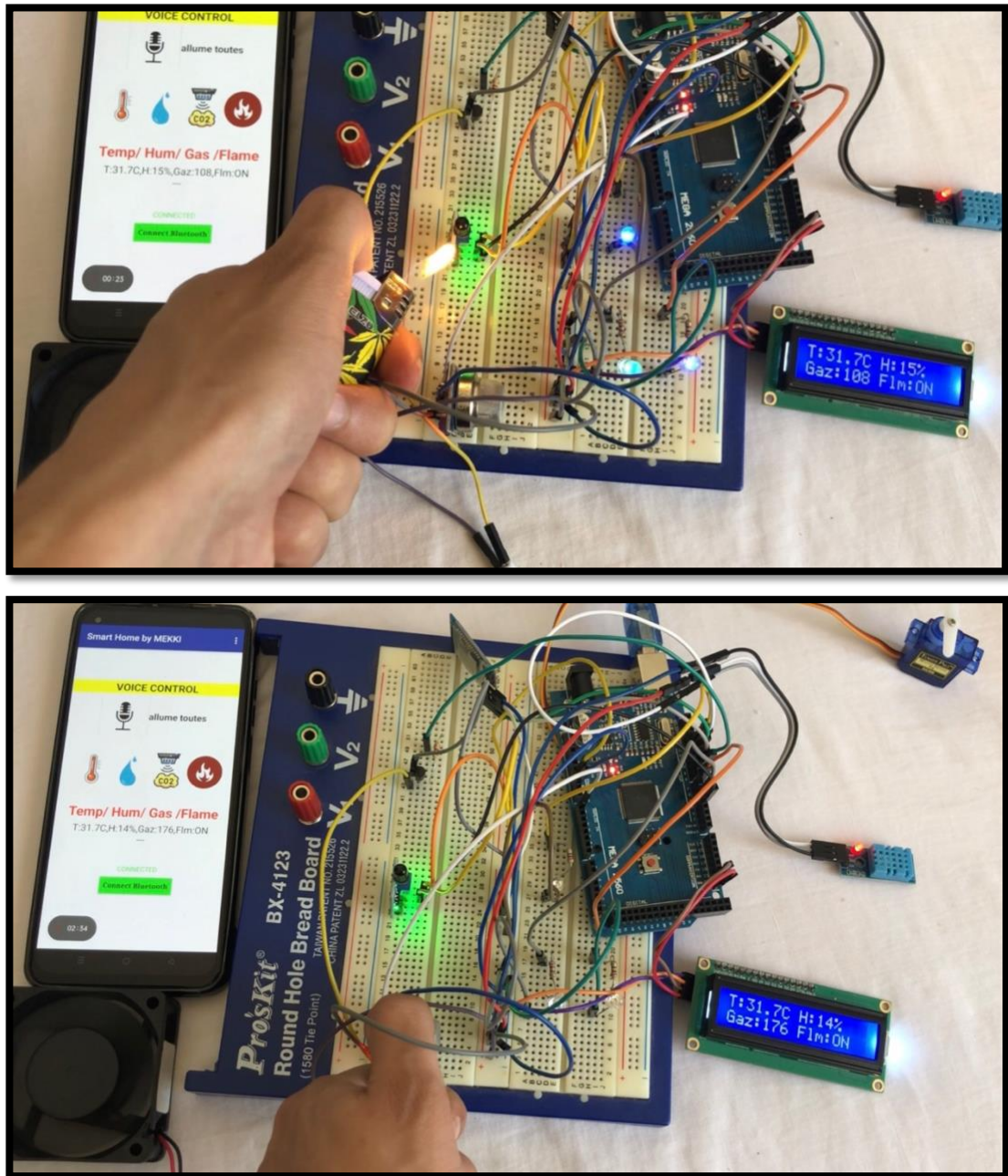
- The Rooms : “allume toutes “ / “ arrête toutes “.
- Bedroom: “allume chambre “ / “arrête chambre “.
- Kitchen: “allume cuisine “ / “arrête cuisine “.
- Living Room: “allume salon “ / arrête salon “.
- Bathroom: “allume douche “ / “arrête douche “.
- The Door: “ ouvre la porte “ / “ferme la porte “.



**Figure (IV.18):** Control in the rooms lighting and opening the door by voice.



**Figure (IV.19):** Automatically controlling the rain sensor to protect clothes from getting wet.



**Figure (IV.20):** The final results of a mobile application for real-time control and monitoring.

### **IV.6.3.Smart Home Design**

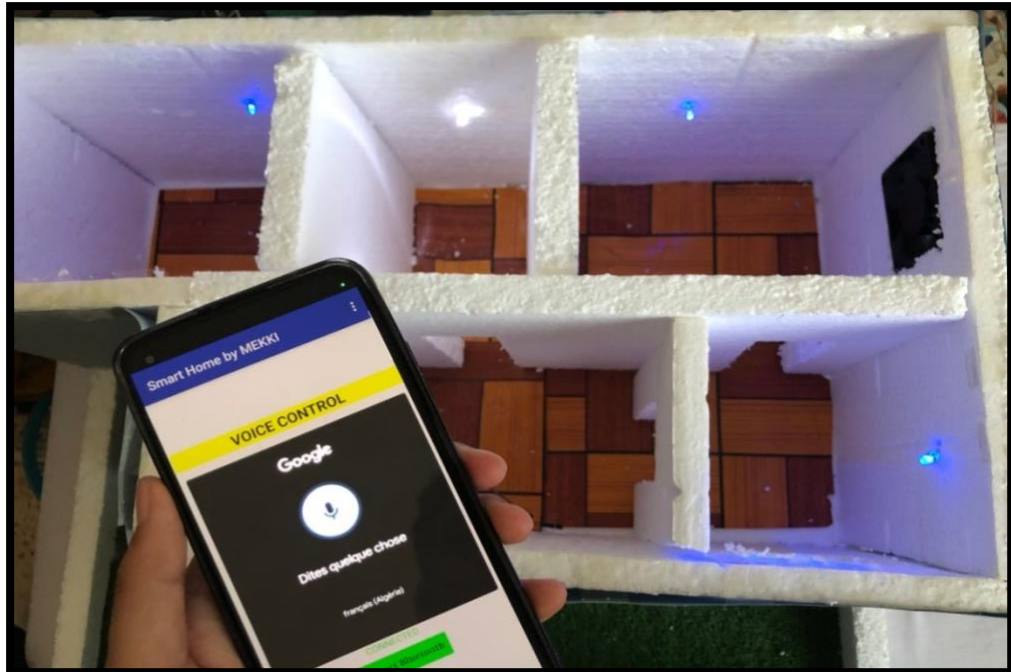
#### **IV.6.3.1.Prototype**

The prototype house includes a living room, a bathroom, a kitchen, a bedroom, and a garden. To ensure optimal operation, the motors and communication units are integrated and coordinated by an Arduino module, along with sensors.



**Figure (IV.21):** Prototype of Smart Home.

### IV.6.3.2. Final Result



**Figure (IV.22):** Implemented tasks of a Mobile Application for Real-Time Control and Monitoring of Smart Home Systems.

## **IV.7. Conclusion**

In this chapter, we completed the practical implementation of my project. To achieve this, we first designed the overall system architecture. Then, we presented the simulation of each sensor and actuator, along with the corresponding flowcharts. We also developed an Android application that allows real-time remote control and monitoring of the house using voice recognition.

Finally, we wired several circuits in the house model and included actual photographs. The tests conducted on the prototype produced satisfactory results.

*GENERAL  
CONCLUSION*

## **General Conclusion and Perspectives**

The smart home represents a significant advancement in our modern lives, combining cutting-edge technologies, automation, and connectivity to achieve the highest levels of safety, comfort, and energy efficiency. Through this project, we successfully developed an integrated smart home system by designing a mobile application that enables remote control and real-time monitoring of various home functions. The system relies on an Arduino board and a set of advanced sensors for gas leak detection, temperature and humidity measurement, as well as lighting control and door management through voice commands. The Proteus simulation software was used to ensure performance accuracy before practical implementation.

The results demonstrated the system's success in achieving its primary objectives: enhancing home safety, optimizing energy consumption, and providing greater user comfort. The project also contributed to the development of our technical skills in programming, electronics, and automation.

Looking to the future, this system holds great potential for development through IoT integration and artificial intelligence. Prospects for this work could include:

- Adding new features to the Android app for greater comfort and convenience, such as an automatic watering system or a pool maintenance system.
- Using a 'Wi-Fi' module to control our system via an online platform, allowing it to be managed from anywhere with an internet connection."
- Enhanced AI Integration, implementing machine learning algorithms to predict user behavior and automate home settings (lighting, heating, etc.) for greater efficiency.
- Expanded IoT Connectivity, Integrating more smart appliances (refrigerators, washing machines, etc.) into the system for a fully interconnected home ecosystem.

- Advanced cybersecurity measures ensure robust protection against potential hacking threats.
- Incorporation of health monitoring systems, such as wearable device synchronization for elderly care or emergency medical alerts.
- Advanced Security Features, incorporating facial recognition, motion detection, and cloud-based surveillance to improve home security.

This project represents a step toward realizing the vision of smart homes that offer safer, more sustainable, and easier living, confirming that these technologies have become modern necessities rather than mere luxuries.

## ***Abstract***

*This study presents the design and implementation of a smart home system based on the Arduino Mega 2560 microcontroller. The system enhances security and comfort through features including voice-controlled doors and lighting, along with real-time smoke, gas, temperature, and humidity detection (using flame, gas, temperature, and humidity sensors). All functions are remotely controlled via an Android application, ensuring resident safety and convenience. The project combines electronics, automation, and IoT technologies to optimize energy efficiency while providing real-time monitoring.*

***Keywords:*** *Smart Home, Sensors, Home Automation, Arduino Mega 2560, IoT, Android Application, Voice Recognition, Security.*