PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY YAHIA FARES OF MEDEA

Faculty of Technology

Department of Mechanical Engineering



Master's Thesis

Commande vocale de dispositifs électriques

Field: Electromechanics

Specialty: Mechatronics

Presented by:

OUTOUAT MOHAMED DHIA EDDINE

ZIANE WALID

Proposed and supervised by:

Dr. KHOUKHA Berrahal

2024/2025

Acknowledgments

First, we thank Allah for giving us the strength and courage through this wonderful journey. We would like to express our sincere gratitude to Ms. KHOUKHA BERRAHAL for her valuable advice and guidance in the process of making our project. We would like to present our sincere thanks to all the teachers that helped us to build the right mentality of an engineer and always encouraged us to keep moving forward under all conditions.

We sincerely thank Mr. Mounir ZIRARI, Professor at the University of Medea, for having giving us the honor of chairing the jury for my dissertation. We extend my respectful thanks to Mr. REBEHI Redha, Associate Professor at the University of Medea, for agreeing to review this work.

Dedication

I would like to dedicate this humble work to:

To my beloved mother and my role model in life, my father

To my precious brother walid & my princess aya

To my brother from another mother safouane

friends: redha , zaki , nori , walid , mohammed , ramzi , reyadh , ziad(ract)

Specially: Ibrahim faci who made my life easier and Abdelfattah, with pleasure and dedication

Dhia eddine

Dedication

To the queen of my world, my mother habiba taleb

To my beloved father

To my support in life, my brothers and sisters

To All my uncles and aunts

To my friends zarok, rikos,zaki ,

my college friends nori,riadh,ridha,rect,Youssef,marmouz,mazoiri,chebli zaki,moh dz

And to everyone who helped us in the study

To my dear friend faci Ibrahim & dhia eddine

Walid

Abstract

This project aims to control electrical and household devices using voice commands, facilitating interaction with the surrounding environment without the need for direct manual intervention. It relies on voice recognition technology to execute commands quickly and accurately, making it especially suitable for people with special needs and the elderly. This contributes to improved efficiency and provides a higher level of comfort and convenience in daily life.

Keywords : voice command, Smart devices, Home automated system, Bluetooth module.

Résumé :

Ce projet vise à contrôler les appareils électriques et domestiques à l'aide de commandes vocales, ce qui facilite l'interaction avec l'environnement sans nécessiter d'intervention manuelle directe. Il repose sur la technologie de reconnaissance vocale pour exécuter les commandes de manière rapide et précise, ce qui le rend particulièrement adapté aux personnes ayant des besoins spécifiques et aux personnes âgées. Cela contribue à améliorer l'efficacité et à offrir un niveau supérieur de confort et de commodité dans la vie quotidienne.

Mots-clés : Commande vocale, Appareils intelligents, Système domotique, Module Bluetooth.

ملخص :

يهدف هذا المشروع إلى التحكم في الأجهزة الكهربائية والمنزلية باستخدام الأوامر الصوتية، مما يُسهم في تسهيل التفاعل مع البيئة المحيطة دون الحاجة إلى تدخل يدوي مباشر. يعتمد النظام على تقنية التعرف على الصوت، حيث تُنفذ الأوامر بسرعة ودقة عالية، مما يجعله مناسبًا بشكل خاص للأشخاص ذوي الاحتياجات الخاصة وكبار السن. ويساهم ذلك في تحسين كفاءة الاستخدام وتحقيق مستوى أعلى من الراحة والرفاهية في الحياة اليومية.

الكلمات المفتاحية: الأوامر الصوتية، الأجهزة الذكية، أنظمة التشغيل الآلي للمنزل، وحدة بلوتوث.

Index

General Introduction 1
Chapter I: Voice control technology 1
I.1. Introduction 2
I.2. Background 2
I.3. Voice Control Technologies 2
I.4. Internet of Things 4
I.4.1. IOT applications 5
I.4.2. IOT and voice control 6
I.5. Voice recognition 7
I.6. Speech recognition 8
I.7. History of voice control 9
I.8. Available voice assistants 10
I.9. Voice command and main areas of use 12
I.10. Conclusion 13
Chapter II: Capturing and conditioning human speech signals 14
II.1. Introduction 14
II.2. Voice Dynamics 14
II.3. Sound wave acquisition 14
II.4. Analog signal and Pre-amplification 15
II.5. Analog-to-digital conversion (ADC) 15
II.6. Digital signal processing (DSP) 16
II.6.1. Noise reduction and filtering 16
II.6.2. Feature extraction 17
II.7. Speech recognition pipeline (Acoustic model, language model, decoding) 17
II.7.1 Acoustic Model 17
II.7.2 Language Model 18
II.8. Text output generation and interpretation 18
II.9. Android audio capture and speech recognition 18
II.10. App Logic 19
II.11. System data flow diagram 19
II.12. Conclusion 20
Chapter III: Introducing the Components of the system 21
III.1. Introduction 22
III.2. Components of the system 22

III.2.1. Microcontroller (Arduino) 22
III.2.1.1. Hardware Components of Arduino 23
III.2.1.1.2. Arduino Architecture 25
III.2.1.2. Arduino types 27
III.2.2. Ultrasonic sensor 28
III.2.2.1. Ultrasonic sensor working principle 28
III.2.2.2. Module specification 29
III.2.3. Module Bluetooth HC-05 30
III.2.3.1. Definition 30
III.2.3.2. Working principle 30
III.2.4. Cables and breadboard 31
III.2.5. Relay module 31
III.2.5.1. Relay Module Function 32
III.2.5.2. Relay module working 32
III.2.5.3. Types of relay modules 33
III.2.6. System outputs (lamp, heater, fan) 33
III.2.6.1. Lamp 33
III.2.6.2. Electrical heater 33
III.2.6.3. Electrical fan 34
III.3. Conclusion 34
Chapter IV: System Realization 23
IV.1. Introduction 35
IV.2. System diagram 35
IV.3. System schematic 36
IV.4. Project work 38
IV.5. Application work 39
IV.6. System code 41
IV.6.1. Ultrasonic sensor 41
IV.6.2. Bluetooth Module (HC-05) Interface 41
IV.6.3. Controlling devices with Relay 43
IV.6.4. Final code 44
IV.7. Conclusion 47
General conclusion 49

Table of Figures

Figure I.1: Voice Control System Components.....03

Figure I. 2 Smart system IoT architecture 04

Figure I.3: Core Components of the Internet of Things (IoT) 05

Figure I.4: A subset of IoT use cases.....06

Figure I.4: A subset of IoT use cases.....08

Figure I.6: Voice and Text-Based Command Processing..... 12

Architecture for Smart Home Control

Figure I.7: global voice Technologies users 13

Figure.II.1: human sound waveform and spectrogram 14

Figure.II.2: voice signal acquisition and processing.....15

Figure.II.3: analog to digital signal sand conversion.....16

Figure.II.4: digital conditioning of sound signal 16

Figure.II.5: filtring and noise reduction digital sound signal 17

Figure.II.6: voice controlled device management system architecture..... 20

Figure III .1: Arduino Uno.....23

Figure III.2 digital signal… 24

Figure III.3 analog signal….....24

Figure III.4 Arduino Technology Architecture.....25

Figure III.5: architecture of Arduino 26

figure III.6 sensor ultrasonic HC-SR04 28

Figure.III.7: ultrasonic sensor working principle 29

Figure III.8 Ultrasonic Sensor Dimensions and Performance 29

Figure III.9 Module Bluetooth HC-05….....30

Figure III.10 Bluetooth model basics 30

Figure III.11 Breadboard 31

Figure III.12: four channels relay module 31

Figure III.13: The principle of operation of the relay 32

Figure.IV.1: Flowchart of the global system 36

Figure.IV.2: electrical system diagram of the project 37

Figure IV. 3: Circuit diagram 38

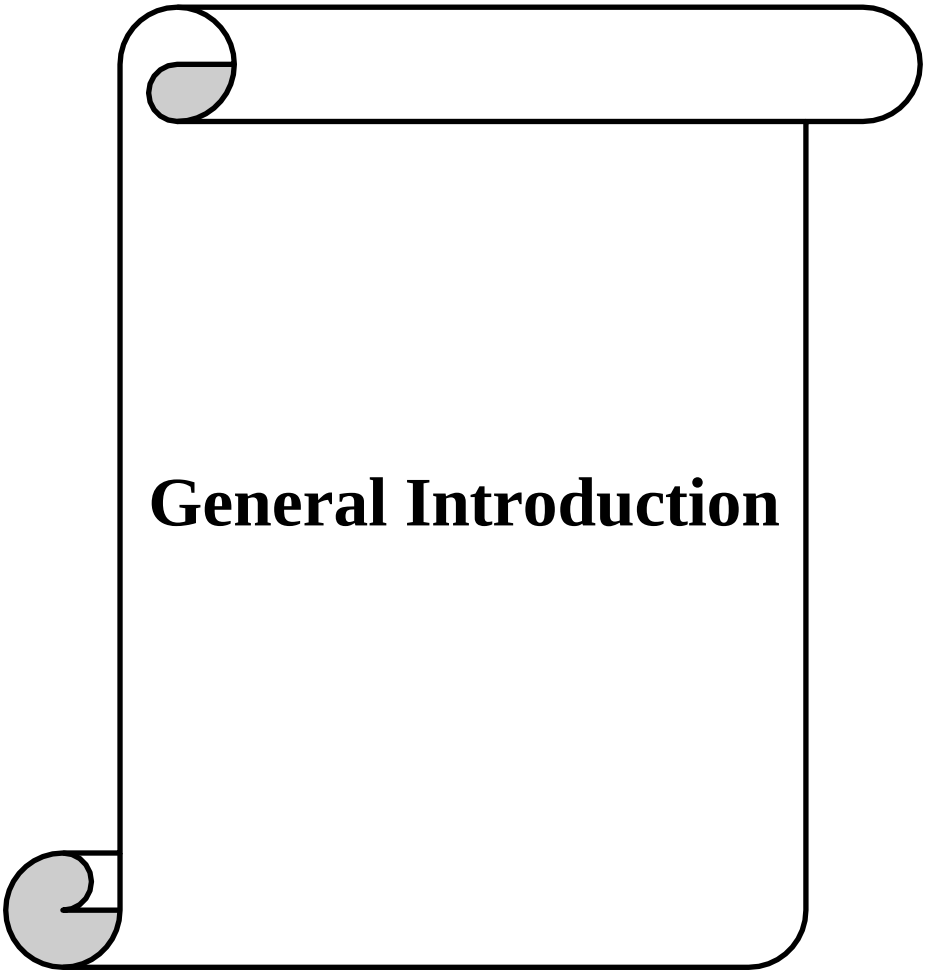
Figure IV. 4: final prototype 39

Figure IV. 5: voice command application 40

Table list

Tableau III.1: Board Arduino type 28

Table III.2: Sensor characterization 29


General Introduction

General Introduction

In this project, we've designed a simple and practical system that allows people to control electrical devices using just their voice. The goal is to make everyday life easier especially for those who may have difficulty using traditional switches, like the elderly or people with disabilities. By using an Android app, the user can say something like "light on," and the command is instantly recognized and sent via Bluetooth to an Arduino. The Arduino then controls a relay that turns the device on or off. This whole process happens quickly and without the need to touch anything. It's a great way to bring smart home technology into real life using affordable and accessible tools. The system is also useful in busy environments where hands-free operation saves time and effort.

The project consists of four chapters structured as follows:

- Chapter I : Voice control technology

In this chapter, we will explore voice control technology and provide an overview of the Internet of Things (IoT).

- Chapter II : Capturing and Conditioning Human Speech Signals

In this chapter, we will discuss voice waveforms and how they are captured and processed using a microphone.

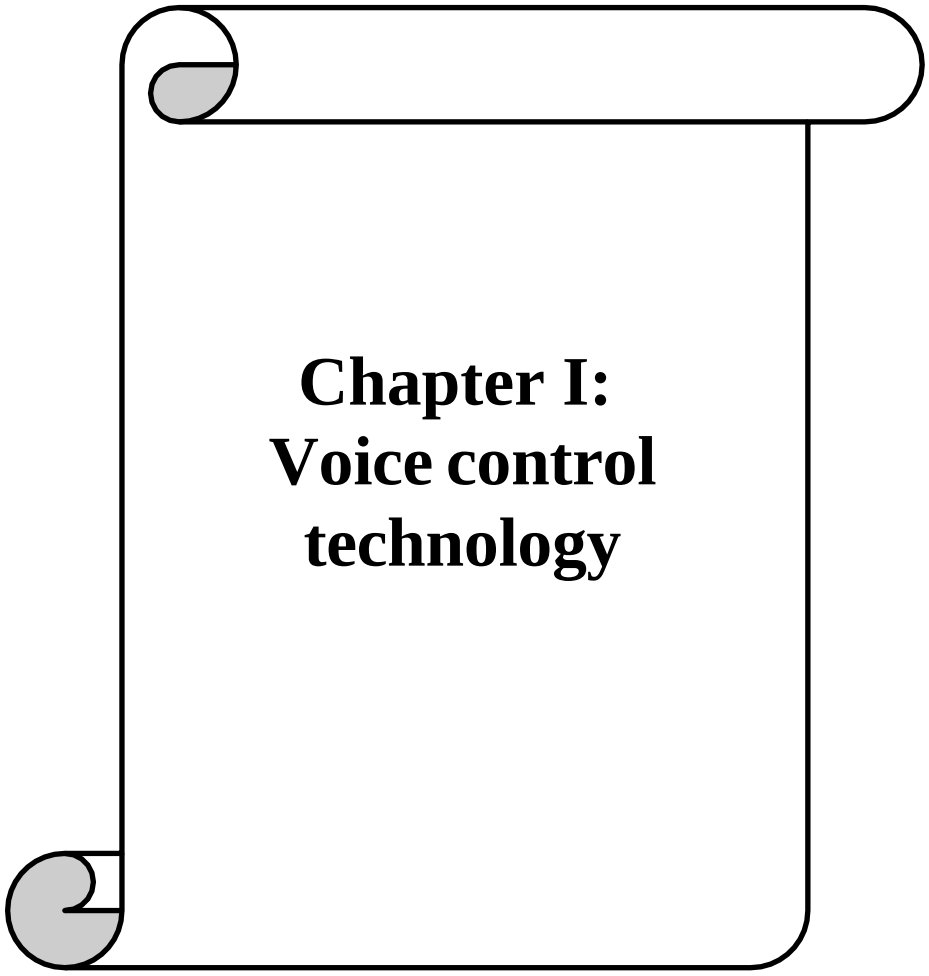
- Chapter III: Introducing the Components of the system.

In this chapter, we will present and explain the main components used in our system.

- Chapter IV : System Realization

In this final chapter, we will present the implementation of our system and discuss the results of our work.

It shows how modern speech recognition and electronics can come together to solve real-world problems and improve comfort, safety, and independence in the home.



Chapter I: Voice control technology

I.1. Introduction

From its first appearance back in 1952, voice control evolved from a simple digit recognizer to a very important tool in multiple domains such as smart home applications, electronic devices, and even some large industries thanks to it integration in internet of things technics. Voice control systems work on the principle of multi-stage signal processing and intelligent interpretation by using beamforming algorithms with microphone arrays to capture a user's speech while isolating the ambient noise Digital Signal Processing (DSP) Algorithms.

The raw data is then prepared by background noise elimination and speech frame segmentation with methods such as spectral subtraction and voice activity detection Informa [1].

Downstream recognition involves feature extraction algorithms calculating Mel-Frequency spectral coefficients which serve as compact descriptors of the speech spectrum.

Territorial recognition is the process of transcribing spoken words into text using acoustic mode which include HMMs (Hidden Markov Models), deep neural nets, and some probabilistic language models as system components.

In natural language understanding modules, control messages are issued via IoT protocols like Message Queue Telemetry Transport (MQTT) to capture the voice commands, empowering them to control the connected devices directly [2].

I.2. Background

Voice controlled automation systems could offer people more comfortable lifestyle and simplify ordinary tasks. Voice control within sustainable are especially beneficial for people with disabilities, enabling a lifestyle, which were previously impossible. An implementation of voice command systems could have great benefits, also offering help with assistance at the workplace. Although the concept has been brought forward multiple times in the past, it was not until recent years the development targeting the general public took place.in this chapter will see the voice control technologies and categorizations in our system.

I.3. Voice Control Technologies

Voice control entails computational transcription of the spoken word, and the interpretation of user intentions for device control or information seeking. Voice controlled systems are described in terms of three modules in [3]: Automatic Speech Recognition (ASR), Natural Language Understanding (NLU) and Decision Making. Alternatively [4], define voice-

controlled dialogue systems to be composed of five stages: Voice Activity Detection (VAD), ASR, NLU, a decision stage, and a communication stage.

Is defined voice control in the home to entail keyword spotting (KWS) for system entry, ASR for transcribing user utterances, and NLU for interpreting the action specified in the utterance. Following [5], It was defined voice assistants (VAs) to build on these core functionalities, and make use of a dialogue manager, natural language generator (NLG) and speech synthesis to enable the two-way interaction seen in dialogue systems. We present the relationship between is defined modules comprising voice assistants in Figure I.2

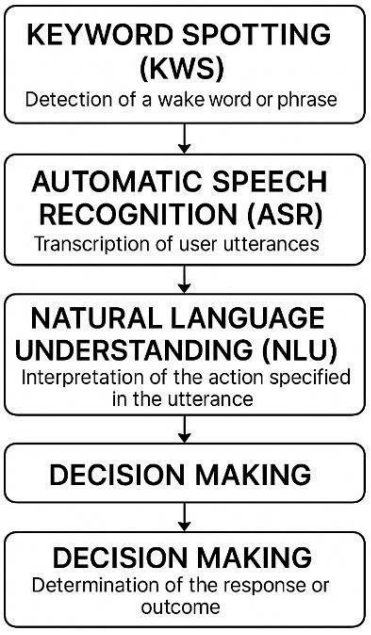


Figure I.1: Voice Control System Components.

Spoken Language Understanding (SLU) refers specifically to the task of inferring the meaning or intent of a spoken utterance [6]. ASR and NLU modules comprise a conventional SLU system. More recently end-to-end SLU systems have gained popularity, where a single model is used to map speech input directly to user intent, without the intermediary step of producing a transcript.

By jointly optimizing the ASR and NLU components, cascading errors are reduced which helps training and gives the technique an advantage [7]. There exist a wide range of additional speech processing tasks that voice assistants have been seen to use. For example, smart homes that have multiple voice-enabled devices use device arbitration, speech enhancement, and speech localization models to improve the performance and user experience [8].

All voice assistants use ASR for recognizing what the user has said, and NLU is necessary for finding meaning in natural language commands [9].

I.4. Internet of things

The Internet of Things (IoT) defines an idea where objects become part of the Internet, and expand it in such a way that our digital and physical worlds are fused together [10]. At a technical level this vision is actualised using networked microcontrollers, sensors and actuators [11]. These devices are consistently subject to downward pressure on cost and energy consumption, similarly to embedded electronics and distinct from general purpose computers (e.g. desktops, smartphones).

The IoT refers to the networked interconnection of those everyday objects, which are often equipped with some kind of intelligence. In this context, Internet can be also a platform for devices to communicate electronically and share information and specific data with the world around them.

The IoT introduces a step change in individuals' quality of life by offering a lot of new opportunities to data access, specific services in education, security, health care or transportation among others. On the other hand, it will be a key to increase enterprises' productivity by offering a widely distributed, locally intelligent network of smart devices and new services that can be personalized to customer needs. The IoT brings benefits from improved management and tracking of assets and products, it increases the amount of information data and allows the optimization of equipment and use of resources that can be translated into costs saving. Moreover, it offers the opportunity to create new smart interconnected devices and explore new business models.

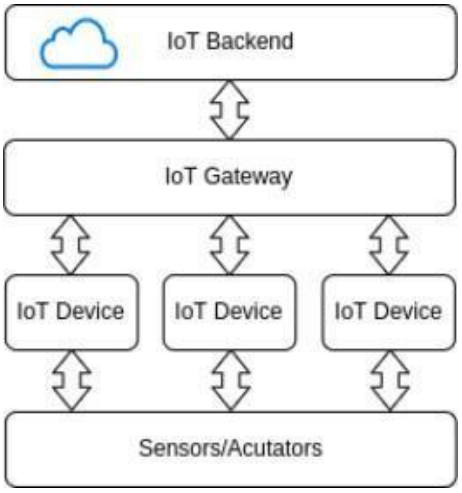


Figure I. 2 Smart system IoT architecture.

I.4.1. IOT applications

The IoT can be seen as a combination of sensors and actuators providing and receiving information that is digitalized and placed into bidirectional networks able to transmit all data to be used by a lot of different services and final users [12].

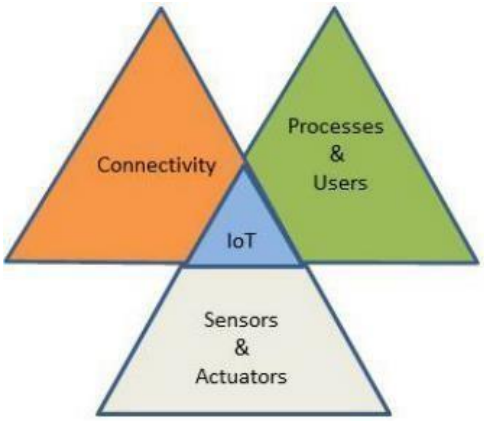


Figure I.3: Core Components of the Internet of Things (IoT).

The number of applications and services that can provide IoT is practically unlimited and can be adapted to many fields of human activity by facilitating and enhancing their quality of life in multiple ways. This chapter gives a short list of applications and services based on IoT. However, it is just a limited description in order to understand all possible new applications and services that IoT could provide. IoT Applications and services:

• **Connected intelligent buildings:** Improvements in efficiency (energy management and saving) and security (sensors and alarms).

• **Domotic applications** including smart sensors and actuators to control home appliances.

• **Health and education services at home:** Remote control of treatments for patients. Cable/satellite services. Energy storage/generation systems. Automatic shutdown of electronics when not in use. Smart thermostats. Smoke detectors and alarms. Access control applications. Smart door locks. Sensors built into building infrastructure to guide first responders and assistances. Safety for all family members.

• **Education:** Linking virtual and physical classrooms to make learning more efficient and accessible, e-learning. Access services to virtual libraries and educational portals. Interchange of reports and results in real time. Lifelong learning. Foreign languages learning. Attendance management.

• **Consumer electronics:** Smart phones, Smart TV, Laptops, computers and tablets. Smart refrigerators, washers and dryers. Smart home theatre systems.

• **Health:** Monitoring of chronic diseases. Improvement of the quality of care and quality of life for patients. Activity Trackers. Remote diagnostic. Connected bracelets. Interactive belts. Sport and fitness monitoring. Intelligent tags for drugs. Drug usage tracking. Biochips. Brain-computer interfaces. Monitoring eating habits.

• **Automotive:** Smart Cars. Traffic control. Advance information about what is broken. Wireless monitoring of tire pressure of car. Smart energy management and control. Self-diagnosis. Accelerometers. Position, presence and proximity sensors. Analysis of the best way to go in real time. GPS tracking. Vehicle speed control. Autonomous vehicles using IoT services.

• Shopping: Intelligent shopping. RFID and other electronic tags and readers. Barcodes in retail. Inventory control. Control of geographical origin of food and products. Control food quality and safety.

• **Manufacturing:** Gas and flow sensors. Smart sensors of humidity, temperature, motion, force, load, leaks/levels. Machine vision. Acoustic and vibration sensing. Compound applications. Smart control of robots. Control and optimization of fabrication processes. Pattern recognition. Machine Learning. Predictive Analytics. Mobile logistics. Warehouse management. Prevent overproduction. Efficient logistics [13].

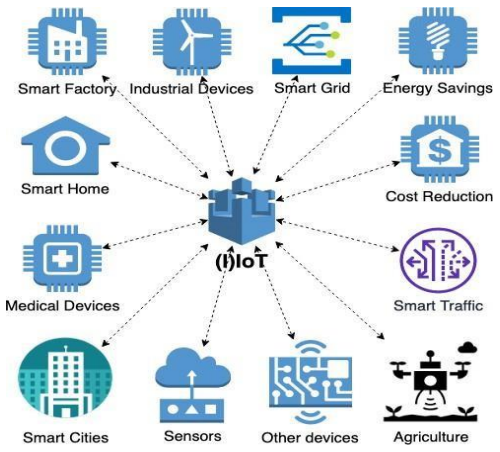


Figure I.4: A subset of IoT use cases [12].

I.4.2. IOT and voice control

IoT (Internet of Things) technology has most certainly had a rather important impact on the very development of voice command systems and also has made them much smarter and more

responsive to their environment. A wide range of connected devices, such as smart lights, air conditioners, electronic locks, as well as industrial systems, can now be controlled with this very integration via speaking voice commands. IoT technology does allow for voice command systems to access real-time sensor data, and that enables much faster, and more accurate command execution. It also improves what the system can do for learning when users behave in a way. The system delivers personalized and efficient responses owing to this improved ability. Voice command has evolved as a result of IoT from one control method into such interface powered by dynamic interaction and real-time data processing [14].

I.5. Voice recognition

The ability of a machine or program to receive and comprehend spoken commands or to interpret dictation is known as voice or speaker recognition.

Voice recognition systems enable hands-free requests, reminders, and other basic tasks by allowing users to communicate with technology by simply speaking to it.

Voice recognition software uses Automatic Speech Recognition (ASR) to recognize and differentiate voices. For a more accurate speech-to-text conversion, some ASR programs require users to first train the program to recognize their voice. The frequency, accent, and speech pattern of a voice are assessed by voice recognition software.

Despite their frequent interchangeability, voice recognition and speech recognition are not the same, and it is important to distinguish between the two. While speech recognition assesses what is said, voice recognition identifies the speaker.

Voice recognition software on computers requires analog audio to be converted into digital signals, known as analog-to-digital (A/D) conversion. For a computer to decipher a signal, it must have a digital database of words or syllables as well as a quick process for comparing this data to signals. The speech patterns are stored on the hard drive and loaded into memory when the program is run. A comparator checks these stored patterns against the output of the A/D converter an action called pattern recognition.

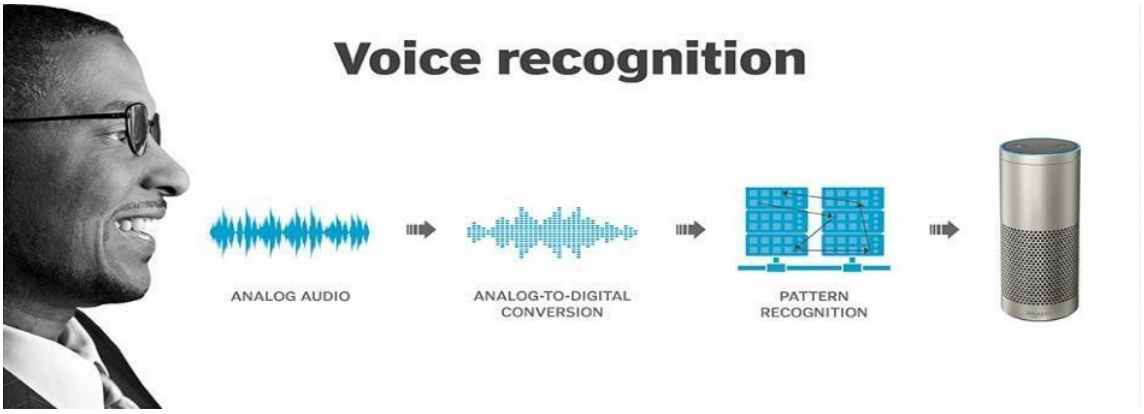


Figure I.5: Voice recognition converts analog audio to a digital signal [1].

In practice, the size of a voice recognition program's effective vocabulary is directly related to the RAM capacity of the computer in which it's installed. A voice recognition program runs many times faster if the entire vocabulary can be loaded into RAM compared to searching the hard drive for some of the matches. Processing speed is critical, as it affects how fast the computer can search the RAM for matches.

Audio also must be processed for clarity, so some devices may filter out background noise. In some voice recognition systems, certain frequencies in the audio are emphasized so the device can recognize a voice better.

Voice recognition systems analyze speech through one of two models: the hidden Markov model and neural networks. The hidden Markov model breaks down spoken words into their phonemes, while recurrent neural networks use the output from previous steps to influence the input to the current step.

As uses for voice recognition technology grow and more users interact with it, the organizations implementing voice recognition software will have more data and information to feed into neural networks for voice recognition systems. This improves the capabilities and accuracy of voice recognition products [1].

I.6. Speech recognition

Speech recognition enables hands-free control of various devices and equipment (a particular boon to many disabled persons), provides input to automatic translation, and creates print-ready dictation. Among the earliest applications for speech recognition were automated telephone systems and medical dictation software. It is frequently used for dictation, for querying databases, and for giving commands to computer-based systems, especially in

professions that rely on specialized vocabularies. It also enables personal assistants in vehicles and smartphones, such as Apple's Siri.

Before any machine can interpret speech, a microphone must translate the vibrations of a person's voice into a wavelike electrical signal. This signal in turn is converted by the system's hardware for instance, a computer's sound card into a digital signal. It is the digital signal that a speech recognition program analyzes in order to recognize separate phonemes, the basic building blocks of speech. The phonemes are then recombined into words. However, many words sound alike, and, in order to select the appropriate word, the program must rely on the context. Many programs establish context through trigram analysis, a method based on a database of frequent three-word clusters in which probabilities are assigned that any two words will be followed by a given third word. For example, if a speaker says "who am," the next word will be recognized as the pronoun "I" rather than the similar-sounding but less likely "eye." Nevertheless, human intervention is sometimes needed to correct errors.

Programs for recognizing a few isolated words, such as telephone voice navigation systems, work for almost every user. On the other hand, continuous speech programs, such as dictation programs, must be trained to recognize an individual's speech patterns; training involves the user reading aloud samples of text. Today, with the growing power of personal computers and mobile devices, the accuracy of speech recognition has improved markedly. Error rates have been reduced to about 5 percent in vocabularies containing tens of thousands of words. Even greater accuracy is reached in limited vocabularies for specialized applications such as dictation of voice recognition [15].

I.7. History of voice control

Voice command technology has evolved significantly over the years, from basic speech recognition systems to the AI-powered virtual assistants we use today. Here's a look at its key milestones:

1. Early Beginnings (1950s – 1970s)

- **1952 – Audrey (Bell Labs)**: One of the first speech recognition systems, developed by Bell Labs, could recognize spoken digits (0-9).
- **1962 – IBM Shoebox**: IBM introduced a device capable of recognizing 16 spoken words and digits, marking a step toward practical voice recognition.
- **1971 – DARPA Speech Understanding Research (SUR) Project**: Funded by the U.S. government, this project aimed to improve speech recognition accuracy, leading to systems that could understand hundreds of words.

2. Growth and Development (1980s – 1990s)

- **1984 – IBM Tangora**: A voice recognition system capable of understanding up to 20,000 words.
- **1990s – Dragon Dictate (Dragon Systems)**: One of the first commercial speech recognition software for consumers, later evolving into Dragon NaturallySpeaking.
- **1997 – Microsoft Speech API**: Microsoft released an API for developers to integrate speech recognition into applications, expanding its use.

3. AI and Machine Learning Advancements (2000s – 2010s)

- **2001 – IBM's Watson Speech-to-Text**: IBM improved speech-to-text accuracy, making voice commands more practical.
- **2011 – Apple Siri**: The launch of Siri in the iPhone 4S marked the first mainstream AI-powered voice assistant.
- **2014 – Amazon Alexa**: Introduced in Amazon Echo, Alexa set a new standard for smart home voice control.
- **2016 – Google Assistant**: Google's AI-based assistant further advanced natural language processing and integration with smart devices [16].

4. Modern Era (2020s)

- **Improved AI and Machine Learning**: Voice assistants like Siri, Alexa, and Google Assistant now use deep learning to improve accuracy and contextual understanding.
- **Expansion into Smart Homes & IoT**: Voice commands are now used for controlling smart appliances, cars, and even healthcare applications.
- **Multilingual and Emotion Detection**: Modern systems can understand multiple languages and detect user emotions for better interactions [16].

5. Present

With advancements in **AI, 5G, and edge computing**, voice technology is more efficient, secure, and personalized. improvements may include **greater contextual understanding, emotion-based responses, and enhanced security features** to make interactions even more seamless [16].

I.8. Available voice assistants

A voice assistant is a software agent that can perform tasks or services for an individual, by using the technology of voice recognition.

1. Apple’s Siri

Siri is the virtual assistant with which Apple has developed and introduced on iOS and made part of several other services in the Apple ecosystem (for iPhones, iPads, MacBooks, etc.). It listens to voice commands and can assist with a wide range of tasks, from finding out information to controlling smart home devices [17].

2. Amazon's Alexa

Alexa is one of the most popular voice-assistants on today's market, developed by Amazon.com INC and initially released in 2014 (Amazon, 2018). Alexa's main purpose is to reach out to a wider platform by being non-restricted and used in various products within various areas

Alexa is Amazon's cloud-based voice service available on more than 100 million devices from Amazon and third-party device manufacturers. With Alexa, allow customers to build natural voice experiences offering them a more intuitive way to interact with the technology they use every day. It offers a collection of tools, APIs, reference solutions, and documentation to make it easier to build for Alexa [18].

3. Microsoft's Cortana

Cortana is a virtual assistant developed by Microsoft, designed to provide voice-controlled assistance on various devices, including Windows computers, smartphones, and smart speakers. It utilizes natural language processing and machine learning to understand and respond to user commands and queries, making it an essential tool for technology, computing, programming, and communications tasks.

Cortana uses advanced algorithms to interpret user voice commands, analyze context, and generate appropriate responses. It relies on Microsoft's cloud-based services to process information, learn from user interactions, and provide personalized recommendations. Its integration with various applications and services allows seamless control over tasks like setting reminders, sending emails, and searching the internet [19].

4. Google Assistant

The Google Assistant was developed by Google Limited Liability Company (LLC) and initially released in 2016. It is primarily available on mobile and smart home device and it can engage in two-way conversations. The assistant not only answers your questions correctly, but also gives some additional context and cites the source website for given information, as expected since it's backed by Google's powerful search technology [19] [20].

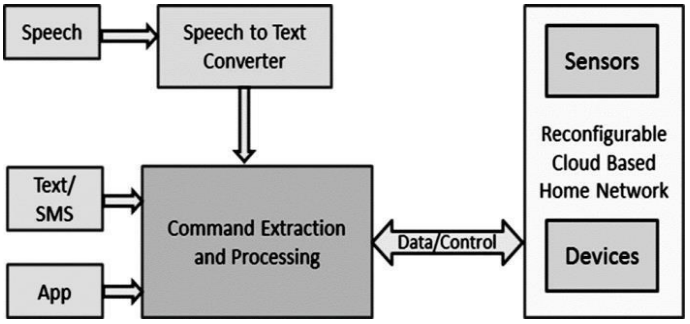


Figure I.6: Voice and text-based command processing architecture for smart home control [20].

Voice assistants can be distinguished as being manually activated, speech activated, or always on. in our work is much being manually to start the system [20].

I.9. Voice command and main areas of use

Today's voice command devices stretch over several different market fields. It can be found in devices such as phones or tablets, as well as in the car or TV-remote.

Almost 60 % today's smartphone users interact with their voice assistant to do online searches. While merely 16% use it to conduct home management tasks. voice command can be divided into two different main types of questions; Tasks that are completed exclusively through voice and tasks that are initiated by voice and completed on screen.

The main reason for usage of smart speakers and voice command is due to situations where the user's hands or eyes are busy. For example, when driving or walking. Other reasons include it producing faster results as well as difficulties to use written commands on some devices. A further advanced AI will provide more extensive usage of voice command, the interaction between the voice assistant and a screen is however most likely to remain even with more advanced machine learning. An approximation of the global regular voice tech users and the usage areas is presented in figure I.7 [21].

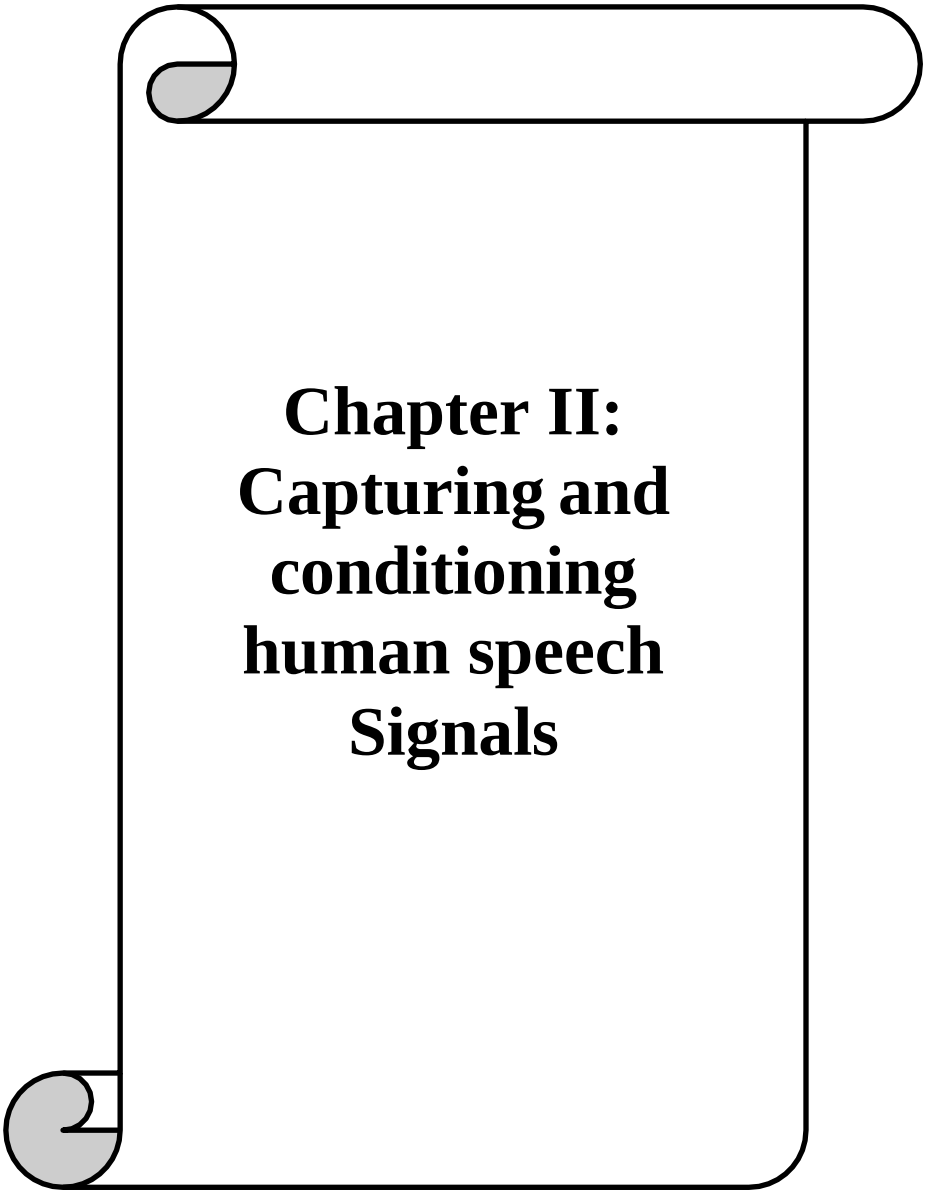


Figure I.7: global voice Technologies users [21].

I.10. Conclusion

In this chapter, we discussed voice command technology, which evolved from a basic idea into an influential tool to change human interaction with machines. The voice command adventure reflects on major milestones within computing, linguistics, and within human-computer interaction, from early speech recognition systems back in the mid-20th century to the advanced AI-powered assistants of today. Internet of Things (IoT) integration has further improved voice control capabilities via connecting it to a wide network of smart devices. This allows smooth automation and real-time responsiveness besides clever decision-making. Such widely available voice assistants demonstrate practical applications of this technology in everyday life today.

In any case of voice command powered through IoT serving particular homes, certain industries, or even some public services, it keeps on growing to be a vital portion of modern smart systems because it offers much convenience, much efficiency, and an even more natural form for communication in between users with technology, In the next chapter we will learn how sound is transmitted, starting from pronouncing the sound until it is transformed into speech.



Chapter II: Capturing and conditioning human speech Signals

II.1. Introduction

This chapter explores the fundamental processes that allow electronic devices to capture, transform, and interpret voice commands. It begins by examining the dynamics of human voice production and how sound waves are generated. It then delves into the role of microphones in receiving these waves and converting them into electrical signals. These analog signals must be pre-amplified and passed through an analog-to-digital converter (ADC) to be converted into a format readable by microcontrollers. Finally, the chapter demonstrates how digital signal processing (DSP) can be applied to extract and analyze relevant information from digital audio data.

II.2. Voice dynamics

When a person says *"Light on"*, the lungs push air through the vocal tract. The respiratory muscles (diaphragm, chest) generate subglottal pressure that forces the vocal folds (vocal cords) in the larynx to vibrate. This vocal-fold vibration produces a "buzzy" voiced sound (air pressure pulses at roughly 100–300 Hz). The raw sound wave then resonates in the throat, mouth, and nasal cavities: the shape of these resonators (tongue, palate, lips, etc.) filters and amplifies frequencies to form the distinct speech sounds. In effect, the vocal folds convert air pressure into a pressure-wave sound, and the articulators mold this into the spoken words *"Light on"* [4].

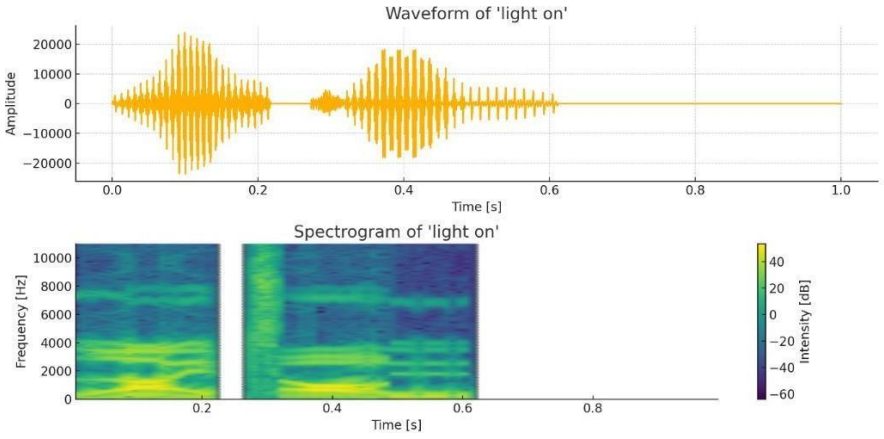


Figure.II.1: Human sound waveform and spectrogram [4].

II.3. Sound wave acquisition

The acoustic sound wave propagates through the air until it reaches the microphone. The microphone element (diaphragm) moves in response to the pressure variations of the *"Light on"* sound. This mechanical motion is then transduced into an electrical signal. In a dynamic microphone, the diaphragm is attached to a coil of wire suspended in a magnetic field. As sound

waves strike the diaphragm, the coil moves back and forth in the magnet's field, inducing a small voltage (via electromagnetic induction) that mirrors the sound waveform. In a condenser (capacitor) microphone, the diaphragm and a fixed back-plate form a capacitor.

The diaphragm is electrically charged (via phantom power), and when it vibrates with the sound wave, the capacitance changes, causing a corresponding voltage change. In both cases the result is an analog voltage waveform proportional to the pressure wave of *"Light on"* [4].

II.4. Analog signal and Pre-amplification

The raw microphone output is typically a very low-level analog signal (millivolts). This signal is routed to a preamplifier to boost its strength. A microphone preamplifier provides gain, raising the mic-level signal up to line level (around 1 V) without introducing distortion. (For example, a 16-bit 44.1 kHz ADC typically expects ~1 Vpp for full-scale input.) In the case of a condenser microphone, the system also applies phantom power (~48 V) to maintain the diaphragm charge; this 48 V also powers any internal impedance converter inside the mic. After preamplification the analog signal is a faithful, scaled-up electric copy of the *"Light on"* sound waveform, ready for digitization. [4].

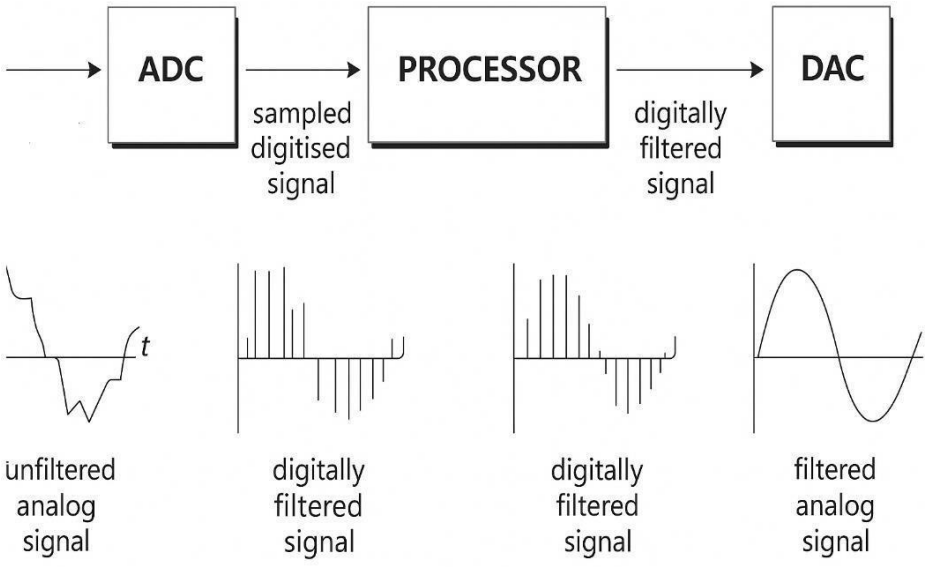


Figure.II.2: Voice signal acquisition and processing.

II.5. Analog-to-digital conversion (ADC)

The amplified analog signal is then digitized. An **ADC (analog-to-digital converter)** periodically samples the incoming voltage at a fixed rate, converting it into discrete numerical values. Sampling must follow the Nyquist criterion (at least twice the highest speech frequency) to avoid aliasing.

At each sample instant, the ADC quantizes the amplitude to a finite number of bits. For example, a 16-bit ADC maps the analog voltage into one of 2^16 levels, introducing a small quantization error. In practice the ADC outputs a digital a time series of binary numbers (often two's-complement) proportional to the sound pressure at each sample. The end result is a digital waveform in memory (or in a data file) that represents the spoken phrase *"Light on"* with all its time-varying amplitude and frequency content [2].

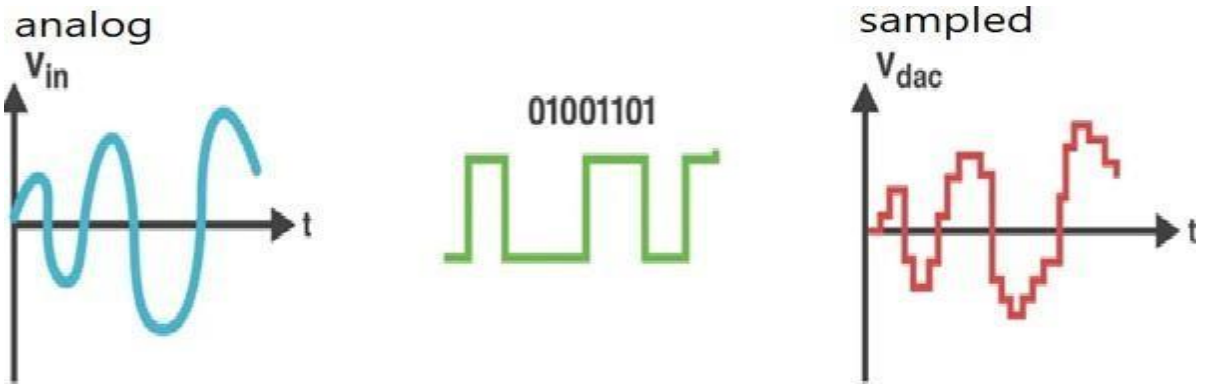


Figure.II.3: analog to digital signal sand conversion

II.6. Digital signal processing (DSP)

Once in the digital domain, the speech signal can be pre-processed to improve recognition. Common DSP steps include noise reduction, filtering, and feature extraction:

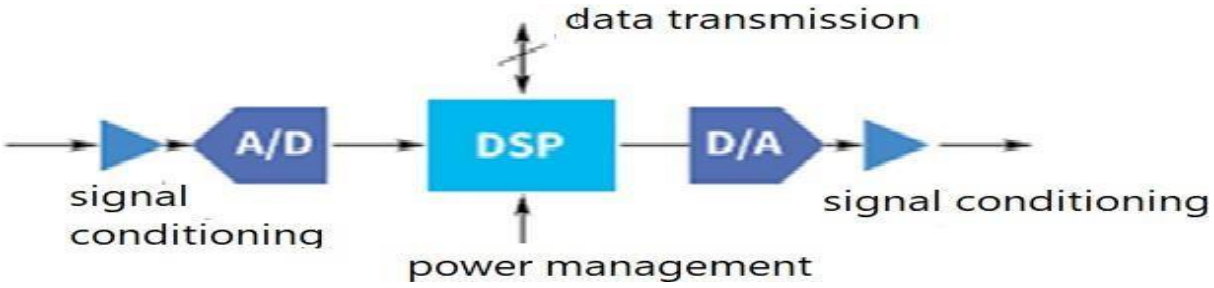


Figure.II.4: Digital conditioning of sound signal.

II.6.1. Noise reduction and filtering:

The signal may be passed through digital filters (bandpass, high-pass, etc.) to focus on the frequency band of human speech (roughly 100 Hz–8 kHz) and to remove low-frequency hum or high frequency hiss. Spectral subtraction or Wiener filtering can suppress background noise. These operations increase the signal-to-noise ratio so that the recognizer sees a cleaner version of *"Light on"*.

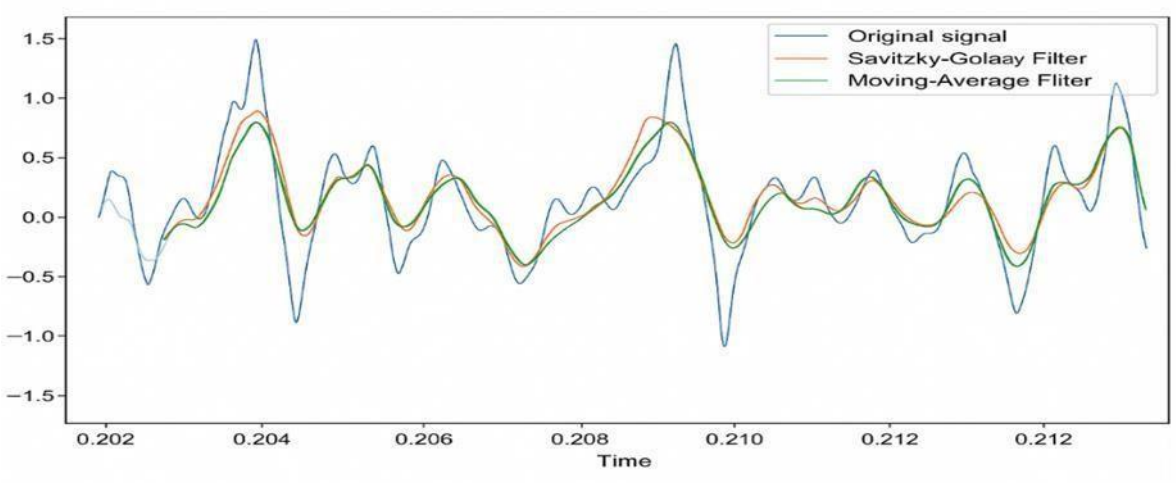


Figure.II.5: Filtering and noise reduction digital sound signal.

II.6.2. Feature extraction:

The filtered speech is then analyzed frame-by-frame. Typically, the audio is divided into short overlapping frames, each multiplied by a window function (Hamming, etc.) to minimize edge effects. For each frame, the DSP performs an FFT to compute the power spectrum. This spectrum is then passed through a Mel-frequency filter bank (triangular filters spaced on the Mel scale) to mimic human auditory perception (more resolution at low frequencies). The log of each filter's energy is taken, and then a discrete cosine transform (DCT) produces a set of Mel-frequency cepstral coefficients (MFCCs).

Typically, 12–13 MFCCs (plus energy or delta features) are used as a compact summary of each frame's spectral envelope [2]. These MFCCs capture the resonant characteristics of the speaker's vocal tract and are robust features for distinguishing phonetic content.

II.7. Speech recognition pipeline (Acoustic model, language model, decoding)

The sequence of MFCC feature vectors is then fed into the speech recognition system. Modern ASR systems use two main statistical models:

II.7.1 Acoustic model:

A trained model that estimates the probability of phonetic units (phones or senones) given the input features. It was built by training on many hours of speech aligned to transcripts, learning which feature patterns correspond to which sounds. When processing *"Light on"*, the acoustic model computes likelihoods over phonemes or characters for each frame of MFCCs.

II.7.2 Language model:

A model that encodes the statistical likelihood of word sequences. For example, the language model captures that "light on" is a plausible word sequence in context, whereas a random sequence might be less likely. It provides prior probabilities for word (or subword) sequences.

A decoder (search algorithm) then combines these models. Essentially, it searches over possible word strings and alignments to maximize the joint probability under the acoustic and language models. Commonly this is done by a Viterbi or beam-search algorithm that finds the single best (or top-ranked) transcription. Figure below illustrates a typical pipeline: the raw audio is converted to features (spectrogram/MFCC), fed into the acoustic model (often a neural network) which outputs time-aligned phoneme probabilities, and then the decoder (using a language model) generates the final word sequence.

The audio is transformed into features (e.g. mel-spectrogram), fed through a neural acoustic model to produce probabilities, and then decoded (with a language model) into text.

Through this process, the system finally outputs the recognized phrase. In our case, the decoder would yield the text string **"Light on"** as the most likely transcription of the audio [2].

II.8. Text output generation and interpretation

The recognizer's output is the text *"Light on"*. Often some post-processing (auto-capitalization, punctuation) is applied, but for a simple command the text may remain as two words. This text string is the digital interpretation of the spoken command. At this stage, any downstream logic can interpret it. For example, a voice assistant might parse "light on" as a command with intent "turn on" applied to the device "light".

II.9. Android audio capture and speech recognition

On the Android app side, the app must request the RECORD_AUDIO permission to use the microphone. When the user triggers the voice input (often by tapping a mic icon), the app starts listening. Android provides a Speech Recognizer API (Application Programming Interface) or an Intent (Recognizer Intent) to capture live audio and recognize speech. Internally, the API usually streams short audio buffers to Google's servers for transcription. (On-device recognition is also possible with newer Android versions or libraries, but it requires downloaded language models.)

Once invoked, the speech recognizer processes the incoming digital audio (computing features like a spectrogram) and produces candidate text hypotheses. These come back in a callback such as on Results (Bundle), which includes an ordered list of possible transcriptions. The first string in results. get String Array List (Speech Recognizer).

RESULTS_RECOGNITION) is the most likely phrase. For example, if the user said “light on”, the recognizer will likely return a list with "light on" as the first result. (The app may also consider alternatives for noise robustness.)

Android’s standard recognizer is connected (via internet) by default, so it works with rich language models but consumes bandwidth and power [21]. For offline use, one can integrate a local ASR engine (e.g. Vosk), which supports Android and works without internet. In either case, the output at the end of this stage is a recognized text string (like "light on").

II.10. App logic

After obtaining the speech text (e.g. "light on"), the app’s logic must decide what command to send. Typically, the app normalizes the text (lowercase, trim) and matches it against expected commands. For example, it may check if the text contains the word “light” and “on”. Upon matching “light on”, the app maps this to a short command token, such as "ON", "L1", or simply "1", which is easier to send over Bluetooth

The app then prepares to send this command via Bluetooth. It first obtains a Bluetooth Adapter and ensures Bluetooth is enabled.

Next, it either scans or (more commonly) uses a paired device.

The Bluetooth device module on the microcontroller side must have been paired with the phone ahead of time (default PIN is often “1234”).

The app can fetch bonded devices. Once the user taps “Connect”, the app opens a Bluetooth Socket using the standard Serial Port Profile and calls socket. Connect (). This establishes an RFCOMM link to the Bluetooth device. With the socket connected, the app obtains an Output Stream and writes the command bytes.

For example: java socket gets OutputStream (). write ("1”. getBytes());

This transmits the character '1' over Bluetooth to Bluetooth’s serial interface. The app can then wait for any response or simply close if done. In practice, one may also listen on a Buffered reader attached to the socket’s InputStream if two-way feedback is needed (for example, the Arduino echoing status) [3].

After pairing, the app’s Bluetooth socket. Connect () call will associate with the Bluetooth device without needing the PIN again. On microcontroller’s side (Arduino in our case), the Bluetooth device simply sends and receives serial bytes over its TX/RX pins as if wired serial [3].

II.11. System data flow diagram

In summary, the data/control flow passes through several stages:

- User's voice ("Light on") → Phone mic captures analog sound → *digitized audio* → Android Speech-to-Text → *recognized text "light on"* → App parses command → *command token ("1" or "ON")* → Bluetooth wireless link → *serial data "1"* → Arduino UART receive → *interpreted command* → Arduino digitalWrite to relay → *relay energizes* → Light turns on [18][21].

Each stage involves a transformation: acoustic →electrical →digital audio → text →encoded signal →wireless packet →electrical signal→ mechanical swift.

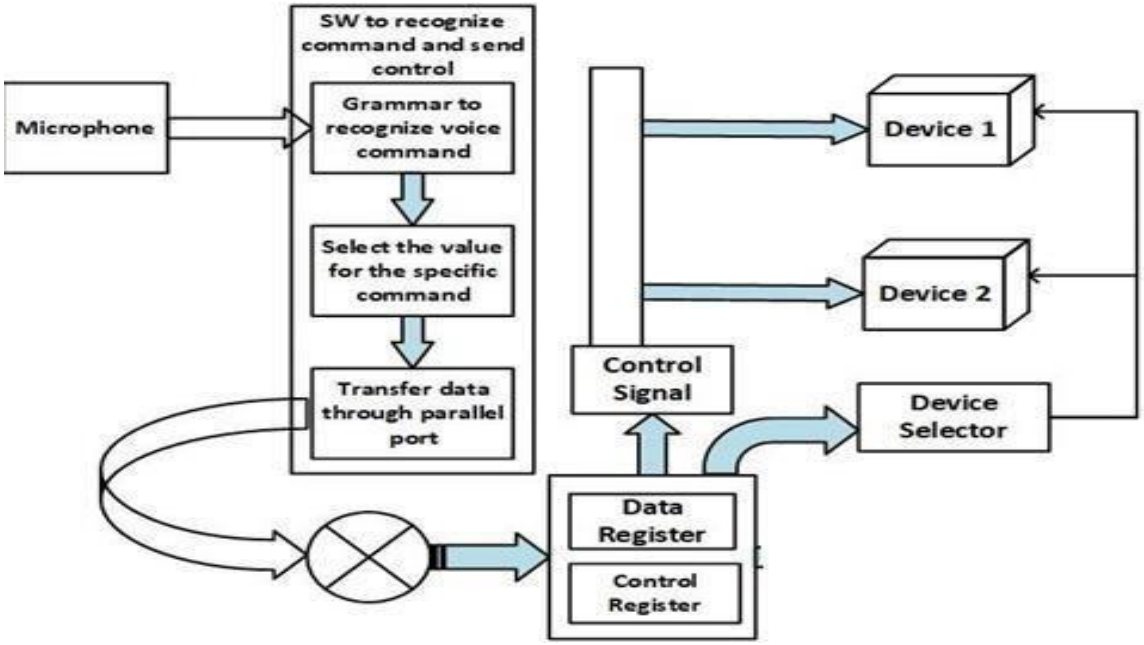


Figure.II.6: Voice controlled device management system architecture [21].

II.12. Conclusion

In conclusion, this chapter covers the entire journey of an audio signal, from natural sound to a digital command understood by electronic systems. Starting with the basics of audio dynamics, we explored how sound waves are produced and captured by microphones. These waves are first converted into analog electrical signals. Through analog-to-digital conversion (ADC), these analog signals are then converted into binary data, enabling further processing.

Finally, digital signal processing (DSP) refines and interprets this data, allowing intelligent systems—such as microcontrollers or computers—to recognize and respond to voice commands. Together, these stages form the foundation of modern voice-controlled applications and play a pivotal role in the development of intelligent automation technologies.

In next chapter we'll present the components of our system**.**

Chapter III: Introducing the components of the system.

III.1. Introduction

With the rapid development of embedded systems, it has become possible to design smart control systems that combine programming and electronics to control various devices easily and efficiently. The project for controlling electrical devices using voice commands is a practical example of this progress. It employs an Arduino board as a central control unit, along with an ultrasonic sensor to detect distances, a Bluetooth module to transmit commands from a smartphone, and a relay module that allows for controlling the on and off of electrical loads.

This chapter of the project aims to study the components of this system, including inputs, outputs, and control, making these devices suitable for smart home applications and assistance for people with special needs. It also serves as a model for integrating software with hardware within an effective automated system.

III.2. Components of the system

Voice control is a modern automation application that facilitates human-machine interaction, enabling the user to control electrical devices using simple voice commands. This system relies on a set of integrated electronic components that work in harmony to execute the required commands. In this project, the system consists of an Arduino module, which serves as the main control brain, and an HC-05 Bluetooth module, which enables receiving voice commands from a smartphone application. An ultrasonic sensor is also used to detect the presence of a person. To control the system, three different electrical devices are connected via a 4-Channel Relay Module. Here are the basic components of this system:

III.2.1. Microcontroller (Arduino)

The Arduino board known as an open-source electronic platform that relies on a set of easy-to-use hardware and software components. It consists of a microcontroller that can be used to read inputs from various sensors and to simplify the control of outputs such as lights, motors, and other devices. the Arduino board is widely used in prototyping and small-scale projects; it is considered one of the simplest electronic boards. The cost of an Arduino board depends on the type of board and the specifications it offers, such as the number of pins on each board.

The electronic board Arduino is programmed in IDE environment built on C language; the code is then sent to the board’s microcontroller to be executed via o USB wire [22].

III.2.1.1. Hardware components of Arduino

We aim to explain how to set up the Arduino board in order to understand how to choose the right board that will simplify the development of our system and projects, and allow for their future expansion and improvement over time [22].



Figure III .1: Arduino Uno.

III.2.1.1.1. Microcontroller

Every micro programmable circuit contain the basic elements which are:

- CPU: is the main element in the -controller, it governs everything that occurs inside the device. It retrieves the program instructions from flash memory and executes them with the assistance of RAM.
- RAM: This is a runtime memory, used to store the temporary data.
- Flash memory: The program uploaded from the Arduino IDE is saved here.
- EEPROM: This is akin of nonvolatile memory, and it retains the data even after device reboot and reset [22].

III.2.1.1.2. USB Port

Every Arduino board needs electricity to power it. We power the board using an external power supply or a USB cable connected to a computer (5V). We can also use it to transfer code to the board [16].

III.2.1.1.3. DC Jack connecter

This is a circular port for connecting an external power supply to the Arduino board. is used This port to power the Arduino when it is not connected to a computer via USB. A voltage

lower than 20 volts can only power the Arduino board. Any power supply higher than this value could destroy the board. The recommended voltage is between 6 and 12 volts [22].

III.2.1.1.4. Brooches

These are the points or connectors used to connect the Arduino pins to electronic components. Electrical signals can be received and data can be read from them.

III.2.1.1.4.A. Digital Pins

It can be programmed Input or output to output a signal of 5 V or 0 V, there are at most 54 pins.

It has PWM pins that enable us to control the output signals, for example 1.2 Volts, 1.7 Volts, etc,

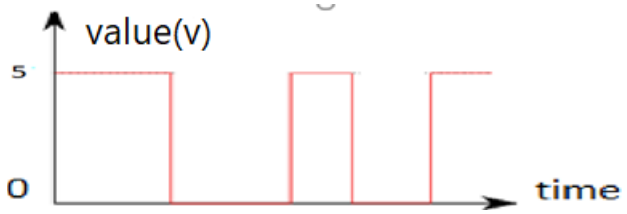


Figure III.2 digital signal.

III.2.1.1.4.B. Analog input

Used to read variable signals such as sensors. It gives signals between 0 to 5 volts, there are at most 16 pins [22].

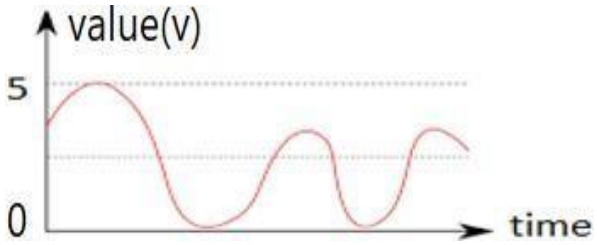


Figure III.3 Analog signal.

III.2.1.1.4.C. Power pins

- 5V or 3.3V: Provides 5V or 3.3V power to drive components.
- GND: Ground, the pin that closes the electrical circuit.
- VIN: The input voltage when the Arduino is powered by an external power supply, it should be between 7 and 12 volts. If it reaches 20 volts, the regulator or the board itself may be damaged.

III.2.1.1.4.D. Special pins

- AREF: Used to provide voltage to analog inputs to improve reading accuracy and quality.

- Reset: Used to manually reset the Arduino microcontroller board, the code in the Arduino is not deleted.
- These are the Serial communication pins that are used to exchange data between the Arduino and other devices, TX (Transmit) and RX (Receive)

  For example, Serial LCD Display and Bluetooth module (HC-05 or HC-06).

III.2.1.1.4.E. Power on LED indicator

It's a small LED on the Arduino board. It lights up when the board is powered up or when the code is uploaded.

III.2.1.1.4.F. Voltage regulator

Regulate the input voltage between 7 or 12 volts (it should not reach 20 volts) to a stable and safe level of 5 or 3.3 volts. To protect the Arduino from damage [22].

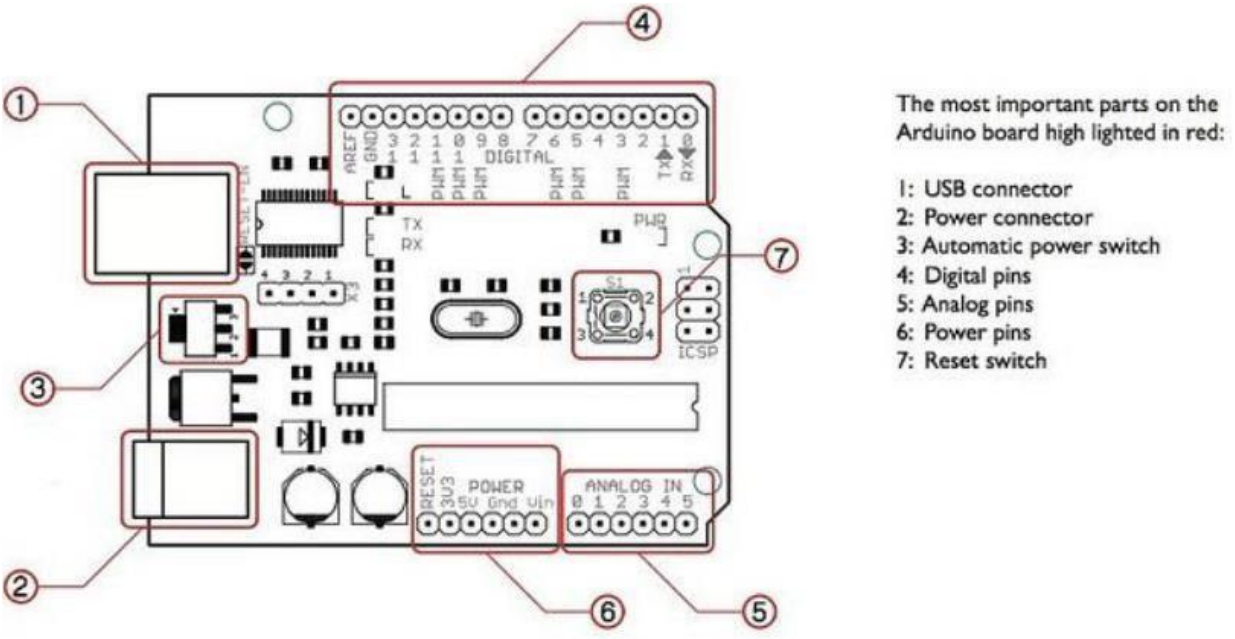


Figure III.4 Arduino Technology Architecture [22].

III.2.1.1.2. Arduino architecture

The Arduino architecture relies on powerful microcontrollers, which enable us to process data in electronic devices. The Arduino architecture also features multiple digital and analog outputs, along with support for various communication features such as, enabling seamless control of various inputs and outputs.

• The processor

• Communications

• User INPUT/OUTPUT pins

• Power supply control

• Peripherals

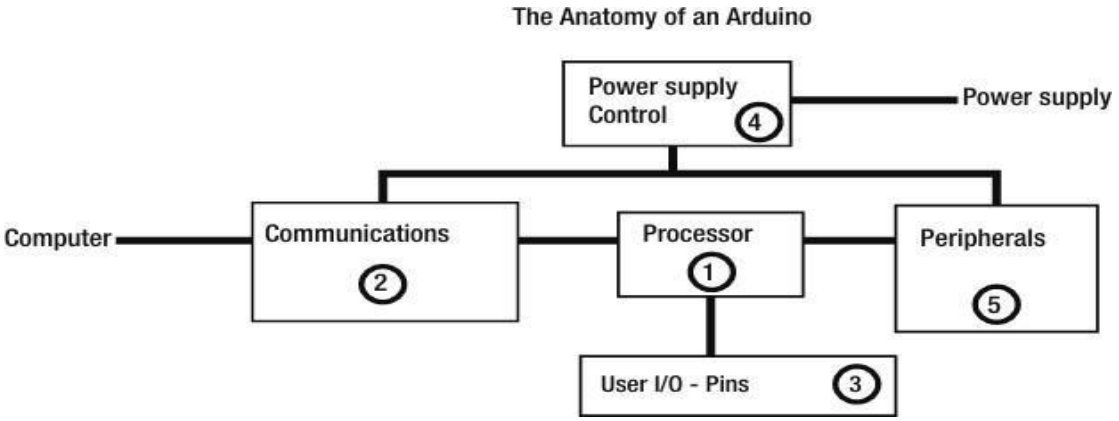


Figure III.5: architecture of Arduino [22].

III.2.1.1.2.A. The processor block

The heart of the Arduino board is the processor, which defines the board's capabilities and features.

It reads signals, processes data, sends commands, and manages communication with external components: the most famous processors are ATmega328P, ATmega2560, ATmega32u4, etc [22].

III.2.1.1.2.B. The communications block

The Arduino's communication module converts signals between the computer and the Arduino board, often via USB-to-serial converter. Initially, a simple serial interface was used, but as computers evolved, a USB converter became necessary. The Future Technology Devices International (FTDI) chip was used in early versions, but later versions were replaced with the ATmega8U2. Some Arduino boards do not have a built-in communication module and are dependent on the user to provide one [22].

III.2.1.1.2.C. The user (INPUT/OUTPUT) pins

The input and output pins on Arduino boards are very important, and the number of pins on any Arduino board can be increased. It's important to note that the use of the word "pin" in Arduino to refer to a single input and output line is different from the actual pin on the chip. Traditionally, input and output lines on processors are connected to pins, each containing eight bits. Arduino converts these bits into pins, and these pins are controlled by software. This allows programs to run on different Arduino models, giving the system an organized appearance [22].

III.2.1.1.2.D. The power supply control block

The power supply control on Arduino boards is responsible for determining the power source for the board. Previously, the user had to choose whether the power source would be from the USB port or an external power supply.

However, in recent versions, this function has been improved so that the board includes an electronic circuit that detects the power source. If there is power from the external input (such as a power adapter), the system switches to use it. If there is no power from the external input, the Arduino relies on the USB port.

III.2.1.1.2.E. The on-board peripherals block

There are two kinds of onboard peripherals (those within the processor and those mounted on the Arduino board itself).

- Peripherals inside the processor

Processor-based peripherals include internal hardware such as counters/timers, serial ports, pin interrupts, as well as A/D converters, which are capable of affecting pin functions and are thoroughly detailed inside the processor's datasheet. People manage a couple outputs while counters/timers have the capacity to yield fixed-frequency PWM signals and function separately. The ATmega328 processor (e.g., Arduino Uno) possesses three timers that support half a dozen PWM signals. The ATmega2560 (Arduino Mega) furnishes a greater number of timers, in addition to a maximum of 15 PWM outputs [22].

- Periphrases mounted on the board

Arduino boards can be differentiated, as opposed to external shields, via integral peripherals, such as Ethernet controllers (Arduino Ethernet), WiFi modules (Arduino WiFi), and Bluetooth capability (Arduino Bluetooth). Arduino Esplora boards incorporate diverse constituents within a gamepad configuration. These components include LEDs, sensors, speaker, buttons, slider, joystick, as well as microphone. Nonetheless, the microphone on the Esplora operates merely as an envelope detector, seizing sole peak sound amplitude without exhaustive audio information [22].

III.2.1.2. Arduino types

Many Arduino boards vary from one another in terms of specifications. Here, we mention four common boards:

Tableau III.1: Board Arduino type [21] :

Arduino types	Microcontroller	Digital pins	Analog pins	Memory (kb)
Arduino Uno	ATmega328P	14 (6 PWM)	6	32
Arduino Mega 2560	ATmega2560	54 (15 PWM)	16	256
Arduino Nano	ATmega328P	14 (6 PWM)	8	32
Arduino Leonardo	ATmega32u4	20 (7 PWM)	12	32

III.2.2. Ultrasonic sensor

It is a device used to measure the distance between it and a specific object by sending high-frequency sound waves (Duration 10 microseconds) and then receiving the echo resulting when the sound waves bounce off the object. Its most prominent uses include measuring distances, avoiding collisions in robots, parking cars, and security systems...etc [23].



Figure III.6 sensor ultrasonic HC-SR04.

III.2.2.1. Ultrasonic sensor working principle

An ultrasonic sensor works by transmitting high-frequency sound waves (usually 40 kHz) through a terminal called a trigger, which transmits the sound pulse. If an object is present, the emitted wave It is reflected when it hits an object and is received via echo. The sensor calculates the time elapsed between transmission and reception. The time measured is sent to the microcontroller. The distance between the sensor and the object can be calculated by the formula:

$$\text{Distance} = {}^{\text{Time Speed of Sound}}/_{2}$$

REMARK:

The module is not suggested to be connect directly to electric power. The sensor does not send waves straight, but in a cone shape. The maximum distance the sensor can measure is 4 meters.

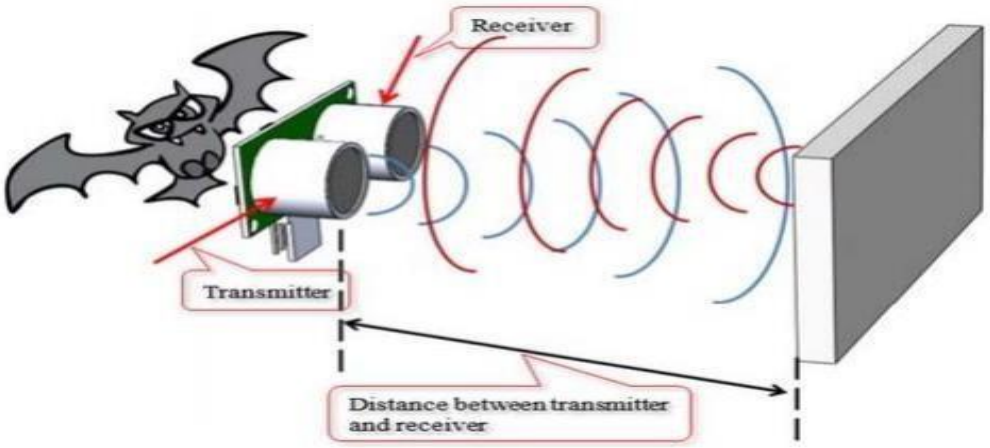


Figure.III.7: ultrasonic sensor working principle [23].

III.2.2.2. Module specification

Table III.2: Sensor characterization [23]:

Information	**Value**
Operating Voltage	3.3Vdc ~ 5Vdc
Quiescent Current	<2mA
Operating Current	15mA
Operating Frequency	40KHz
Operating Range & Accuracy	2cm ~ 400cm (1in ~ 13ft) ± 3mm
Sensitivity	-65dB min
Sound Pressure	112dB
Effective Angle	15°
Connector	4-pins header with 2.54mm pitch
Dimension	45mm x 20mm x 15mm
Weight	9g

The following image represents the best angle for measuring the distance relative to the ultrasonic sensor, which is 30 degrees.

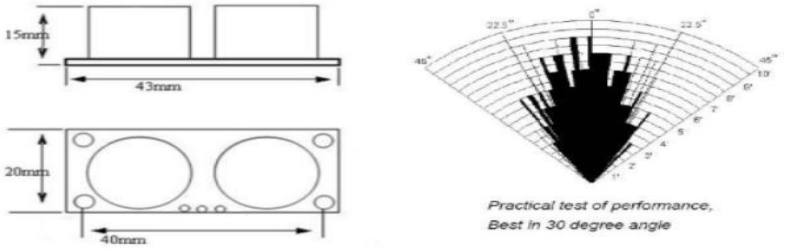


Figure III.8 Ultrasonic sensor dimensions and performance [22].

III.2.3. Module Bluetooth HC-05

III.2.3.1. Definition

The HC-05 is a Bluetooth module used in embedded systems projects such as Arduino, enabling wireless communication between the microcontroller and other devices (smartphones, laptops, etc.). It is used in remote control of appliances, smart home systems, mobile robots, voice recognition projects, etc [23].

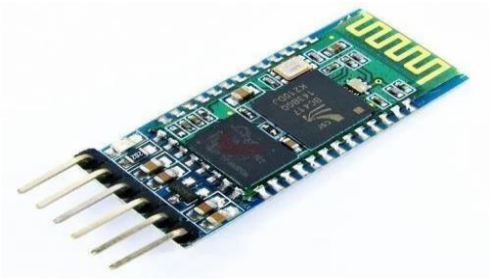


Figure III.9 Module Bluetooth HC-05.

III.2.3.2. Working principle

The HC-05 Bluetooth module is a popular wireless communication module designed to provide transparent serial communication between microcontrollers and smart devices such as phones or computers. This module comes pre-configured to operate as a slave, transferring data directly without the need for programming. The module supports two operating modes:

- Data Mode, which is used to exchange data after pairing.
- Command Mode, which allows the user to modify settings (name, icon, etc.) by sending Attention (AT) commands via a terminal program. Changes are stored even after a power failure. The unit can also be reconfigured as a master unit to automatically start pairing with other Bluetooth units [23].
- It operates with voltage between 3.6V and 6V, but the RXD input operates at a 3.3V logic level only, requiring the use of a logic level converter when connecting it to 5V microcontrollers such as the Arduino Uno or Mega [23].

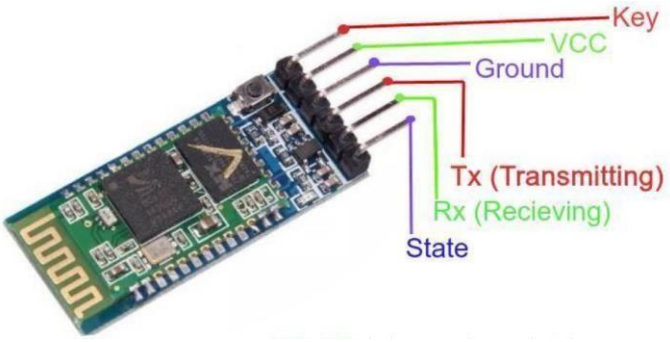


Figure III.10 Bluetooth model basics.

III.2.4. Cables and breadboard

Jumper wires are used to connect electronic components to each other on a breadboard or to boards such as the Arduino. They are available in various types, such as male-to-male, female to male, and female-to-female, depending on the desired connection type a breadboard is a tool used to build and test electronic circuits without the need for welding. It includes a network of internally connected holes in an organized manner that allows components to be easily connected and quickly rearranged during experimentation and development. They are used together to connect external devices to the Arduino, such as sensors, motors, lights, etc... [22].

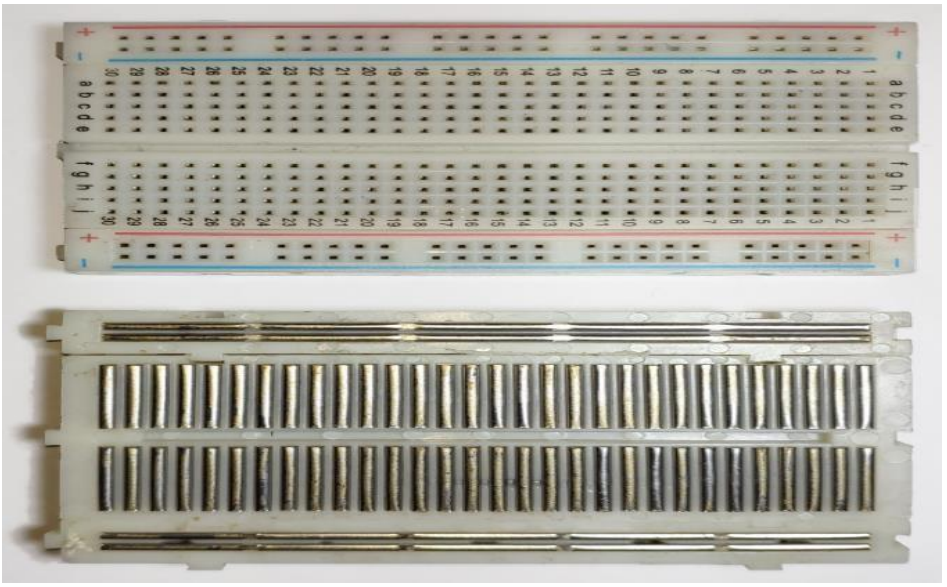


Figure III.11 Breadboard.

III.2.5. Relay module

A relay module is a switch in gadget, the control circuit that functions with low-power signals. It allows low-power supply circuit to switch on or govern a high-power supply circuit without merging it with the same circuit or electrical device [23].



Figure III.12 Four channels relay module.

III.2.5.1. Relay Module Function

The relay module is broadly used in electrical applications, because it enables low-voltage circuits to safely control high-voltage or high-current loads. Its primary function is to electrically isolate the control circuit from the load circuit, safeguarding delicate components. It can also amplify signals by employing a small input current to generate a larger output current, which is vital in many industrial operations. Moreover, in safety applications, the relay module can trigger alarms or shut down systems in the event of failure, aiding to prevent accidents [23].

III.2.5.2. Relay module working

To comprehend how a relay module functions, it's crucial to initially explore its internal structure. At its heart, a relay module comprises a coil, an armature, a spring, and a set of contact spots. When electrical current passes through the coil, it produces a magnetic field that pulls the armature towards it.

The contact spots in a relay are commonly categorized as either "normally open" (NO) or "normally closed" (NC), signifying their standard state when no current is present. When the relay is energized, the magnetic field causes the armature to move, altering the placement of the contacts to either complete or interrupt the electrical link to the load. Once the current is disconnected, the spring mechanism restores the contacts to their original condition [23].

The operating concept of a relay module is based on electromagnetic induction, allowing it to perform switching operations swiftly and dependably. This makes it appropriate for a broad spectrum of uses from basic duties like switching lights to complex industrial automation setups.

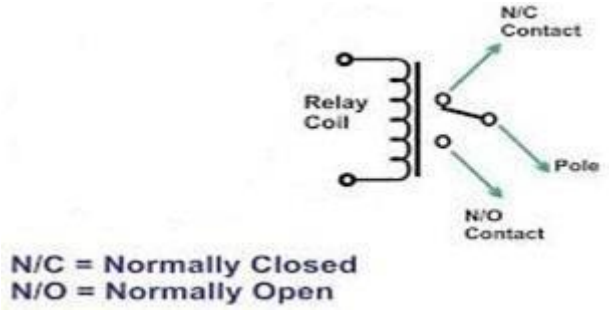


Figure III.13 The principle of operation of the relay [23].

III.2.5.3. Types of relay modules

- **Electromagnetic Relay Modules:** These are the most common types that operate on electromagnetic induction. They are used in a wide range of applications, including automotive and industrial applications, as well as residential and commercial uses.
- **Solid-State Relay Modules:** Unlike electromagnetic relays, SSRs do not include a contact mechanism that moves. They utilize semiconductor devices to switch circuits, operate quicker than relays, and possess a longer service life. It is suitable for applications where high-speed switching is desired.
- **Reed Relay Modules:** These relays operate on the principle of Reed Switch, which is regulated by an electromagnetic coil. A trait of the Reed relay is its swift response time, and it's chiefly selected for use in applications where the switching action has to be very rapid.
- **Time-Delay Relay Modules:** These relays provide a time lag between the actuating signal and the circuit switching actions. Several of the most common applications for these devices encompass circuits where a reaction is needed following a certain delay, such as motor start/stop circuits.
- **Latching Relay Modules:** Latching relays are those capable of remaining in a specific state (either the ON or OFF position) even after the control signal is removed. These are essential in applications where the relay must stay in one state and is not furnished with a continuous power source, like in-memory utilized in computers.
- **Polarized Relay Modules:** These relays are susceptible to the polarity of the control signal; this, in essence, signifies that they will only function as required when the polarity of the controlling signal is correct. They are used in circuits where polarity is very sensitive and impacts the efficiency of the circuit [23].

III.2.6. System outputs (lamp, heater, fan)

III.2.6.1. Lamp

A lamp is a device that produces light, typically powered by electricity or batteries. It can be used for illumination, decoration, or creating a cozy atmosphere. Lamps come in many styles, such as table lamps, floor lamps, and wall-mounted designs.

III.2.6.2. Electrical heater

Electric heaters are one of the most prevalent heating appliances utilized in homes. They transform electrical energy into warmth using an internal heating element, frequently an electrical resistance like a chrome wire. They function at 220 volts. They are available in diverse

sizes and designs to fit a variety of applications and are extensively used in homes, offices, and small areas to furnish effective and secure heating.

III.2.6.3. Electrical fan

It is a device used for air conditioning and cooling applications; it has an electric motor that turns electrical energy into rotational movement to help the system work efficiently.

III.3. Conclusion

Concluding this chapter, we discussed the components of audio-controlled system, which uses an Arduino board as the main control unit. It combines multiple technologies, such as an ultrasonic sensor for detecting distances, a Bluetooth module for receiving voice commands from a phone or computer, and a relay module for controlling electrical devices. This project highlights the great potential of integrating these technics to build smart and efficient control systems used in automation and smart home applications. In the next chapter, we'll explore the nature of this system's operation, its implementation, and the integration of devices.

Chapter IV: System Realization

IV.1. Introduction

This chapter focuses on the system testing phase, one of the most important stages in developing smart systems. This phase verifies the efficiency of the electronic components, the integrity of the connections, and the accuracy of the software used to execute commands. This section tests the main system functions and analyzes their performance under various conditions to ensure their stability and effectiveness in a real-world operating environment.

The chapter focuses on explaining the field experiments conducted on a system designed to control electrical appliances using voice commands via a smartphone via the HC-05 Bluetooth module. It also utilizes an ultrasonic sensor to automatically detect the presence of people and activate or deactivate the system.

A 4-channel relay module is also used to control the actual on/off of devices such as lights, heaters, and fans. To facilitate reading and understanding of this system, we have created a schematic diagram and electrical circuit.

IV.2. System diagram

Our system can turn on and off any electrical device like a fan, light, or heater via Bluetooth by using the Android application to capture and process the voice via smartphones.

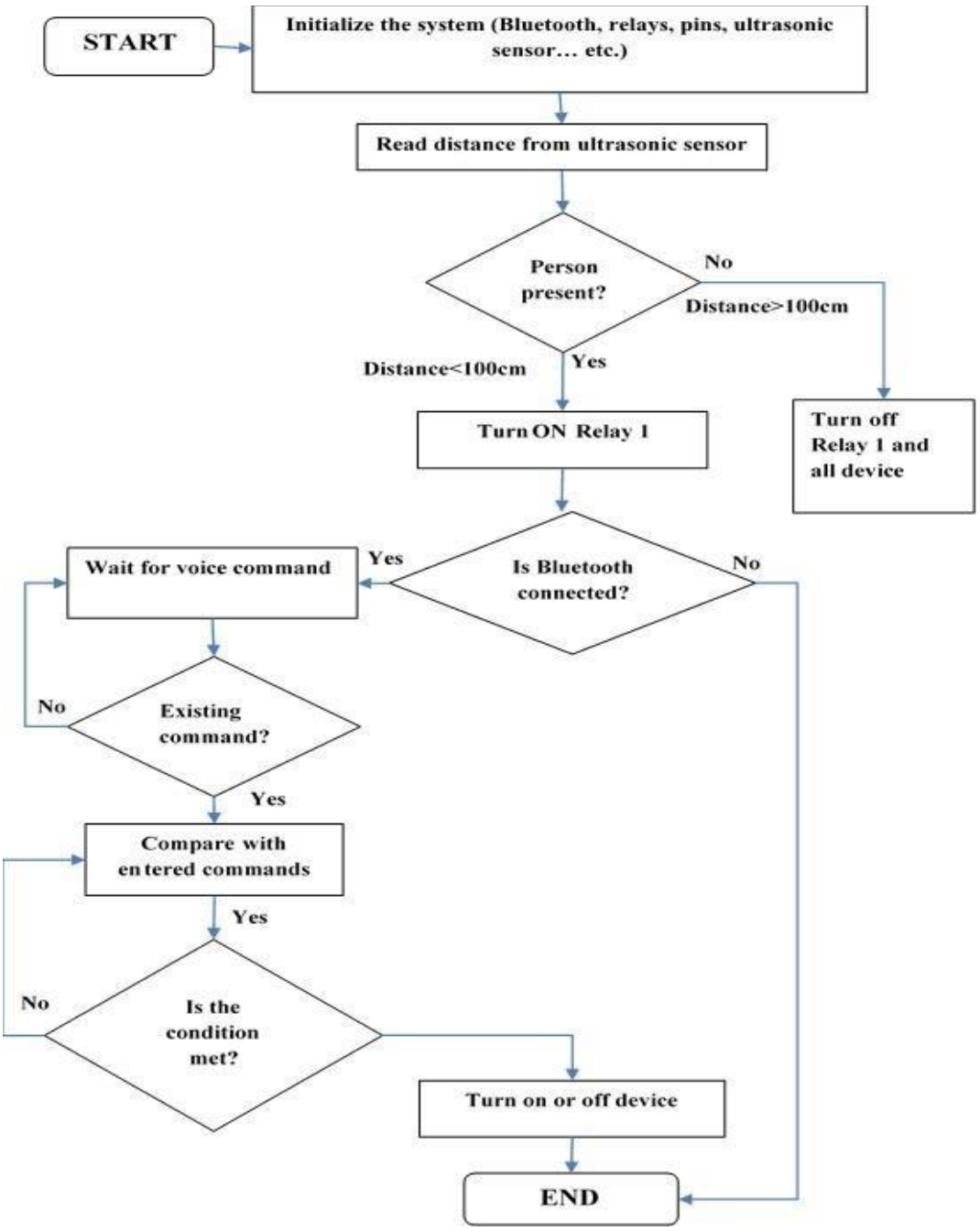


Figure.IV.1: Flowchart of the global system.

IV.3. System schematic

The following circuit, designed in Proteus, represents the global system configuration. It shows how the components are connected and work together to make the system function.

Proteus is a popular software suite used primarily for:

- Electronic circuit simulation
- Microcontroller simulation
- PCB (Printed Circuit Board) design and layout

It allows engineers and students to design circuits, simulate their behavior (including microcontroller firmware), and test the logic before physically building the hardware.

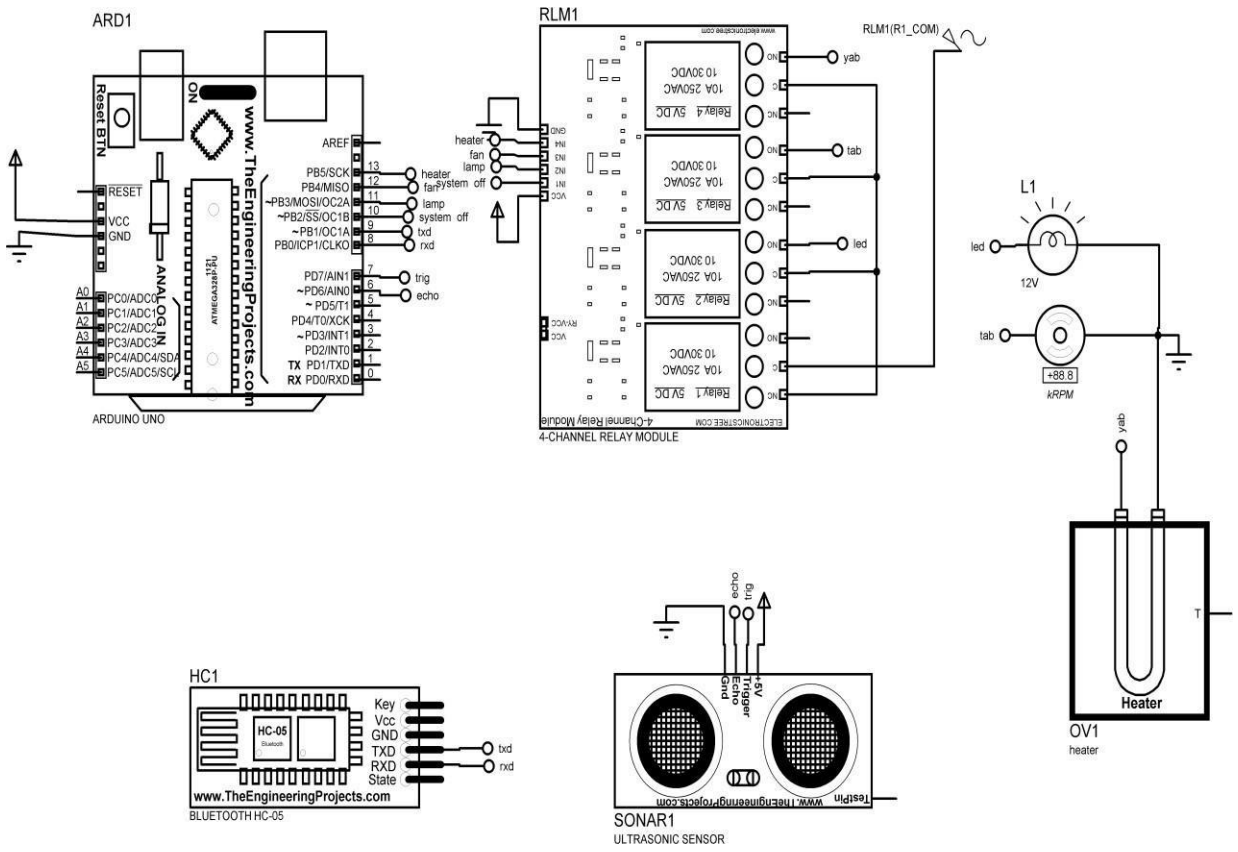


Figure.IV.2: Electrical system diagram of the project.

This diagram represents an automated system for controlling electrical devices (such as a lamp, fan, and a heater) using voice commands via the HC-05 Bluetooth module, as well as an ultrasonic sensor for individual detection.

The system is centered around an Arduino UNO (ARD1), which receives voice commands from a mobile application through the Bluetooth module. These commands are processed and used to control the four-channel relay module (RLM1), which turns electrical appliances on or off accordingly.

The command portion (IN1, IN2, IN3, IN4(and the power portion of the relay are connected to the first relay (COM) and the 220V external power supply, respectively. The (NO) ports of the three relays are connected to the (COM) ports of the remaining three relays (2, 3, 4), and each relay is connected to its own device.

The Bluetooth module (HC-05) is connected to the Arduino via digital pins D9 and D8, utilizing the Software Serial library to handle serial communication without interfering with the default hardware serial pins. This setup allows for simultaneous Bluetooth communication and USB

debugging, ultrasonic sensor (SONAR1) is used to measure distance and can trigger or deactivate certain devices depending on object proximity. It is connected to pins D7 (Trig) and D6 (Echo) on the Arduino. System controls a 220V lamp (L1), a fan (tab) with a speed monitor, and a heater (OV1), all through relay channels. The relays act as switches to isolate and manage the high-power circuits separately from the Arduino.

IV.4. Project work

After completing the electronic components, the work begins by uploading the programming code to the Arduino board via the Arduino IDE using a USB cable. This project instructs the Arduino to control four relays using both the HC-SR04, ultrasonic distance sensor and the HC-05 Bluetooth module. The sensor is connected to the Arduino to measure the distance between it and an object in front of it. This measurement is used to automatically activate the first relay "IN1" when the person approaches within 100 centimeters. In parallel, the Bluetooth module is used to receive voice commands from a smartphone via an app such as AMR_Voice. The user Gives order to manipulate devices for example “on light" to activate the second relay"IN2" in which a lampe is connected, "on fan" to activate the third"IN3", and "on heater" to activate the fourth “IN4, in addition to shutdown commands such as "off light," and so on. The Arduino translates these commands into digital signals that activate or deactivate electrical devices connected to the relays, combining voice control and automation into one smart system.

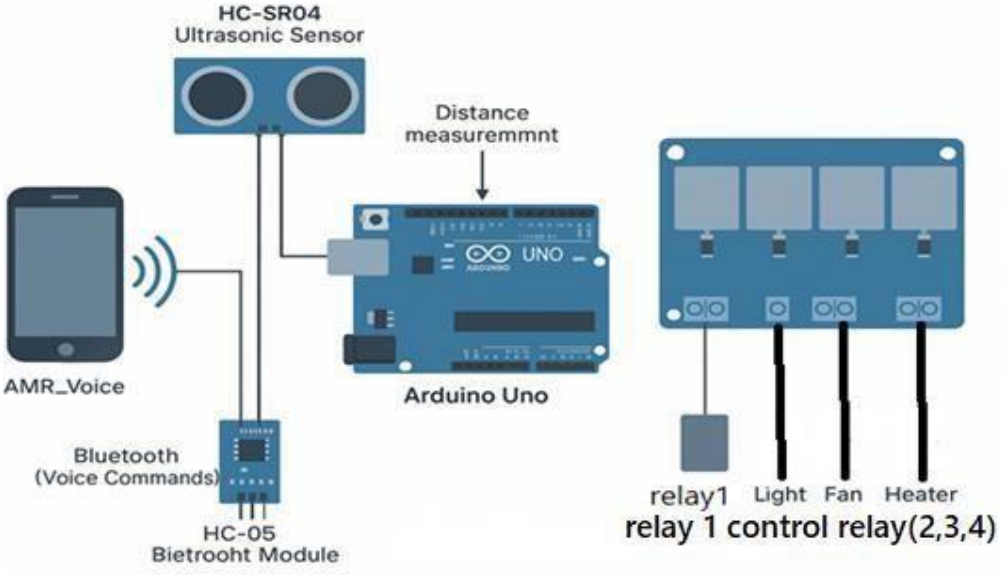


Figure IV. 3: Circuit diagram.

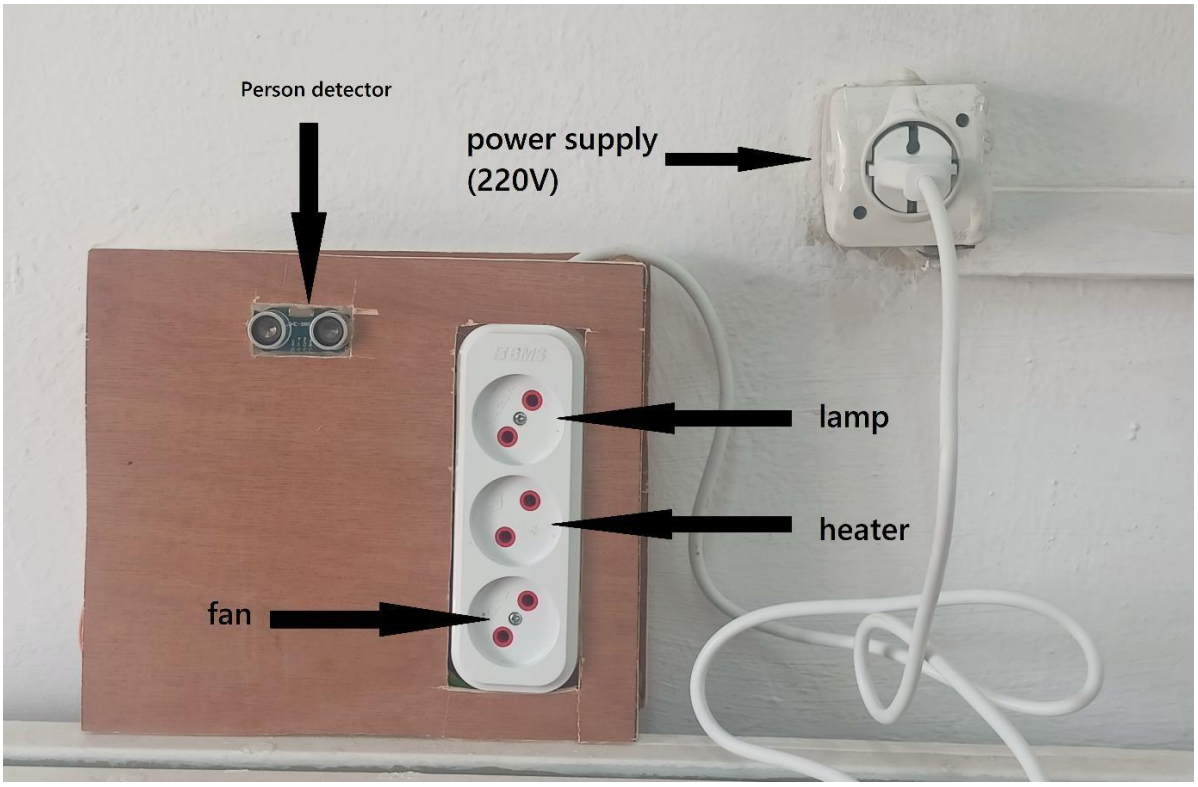


Figure IV. 4 : final prototype.

IV.5. Application work

AMR_Voice is a free Android application used to translate voice commands into words and sentences and transmit them from the phone to an electronic board (Arduino) using the hc-05 Bluetooth module to control electronic devices or any electronic component. This application acts as an intermediary between the user's voice and the control unit (such as the Arduino board).

To connect, open the app and press the "Connect" button to select a Bluetooth device (such as the HC-05) and connect to it. Bluetooth must be enabled on your phone and the HC-05 must have been previously paired.

When connected, we press the microphone button in the app and speak.

Google's voice recognition service is used to convert the voice to text.

The text is sent to the Arduino via the HC-05, where it is read and parsed. The words in it contain these two symbols (*#).

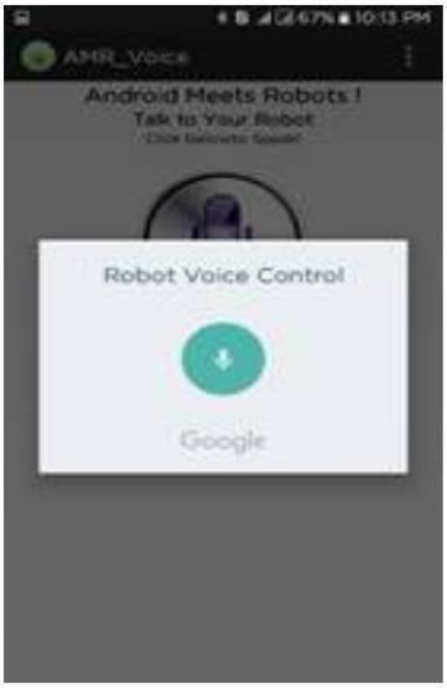
The application sends strings in this configuration *command#, the reference bullet (*) demonstrates the beginning of another charge, and the hash-tag (#) shows the finish of an order.



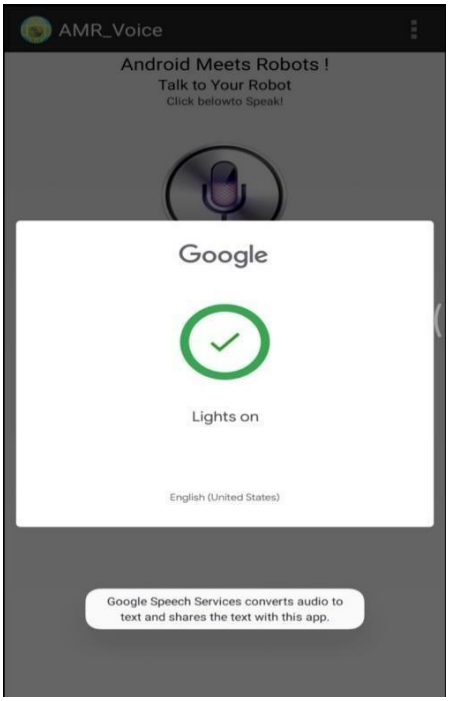
a- Connecting to Bluetooth module



b- Connection successful



c- Waiting for voice command



d- Command passed successful

Figure IV. 5: Voice command application.

IV.6. System code

IV.6.1. Ultrasonic sensor

The HC-SR04 measure distance by sending a short pulse (HIGH for 10 microseconds) through the Trig pin and listens for the reflected signal on the Echo pin. The duration of the pulse is measured using the pulseIn() function. To achieve and read the sensor response, the sensor is connected to a microcontroller such as (Arduino) and we program it via the (Arduino IDE) application with the following code:

```
// Define pin numbers for the ultrasonic sensor
#define TRIG_PIN 9
#define ECHO_PIN 12
// declaration of distance with float type
float distance;
void setup() {
  // Set the TRIG_PIN &&ECHO_PIN pin as OUPUT and INPUT
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  // Start serial communication at 9600 baud rate
  Serial.begin(9600);
}
void loop() {
  // Make sure the TRIG_PIN pin is LOW initially
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  // Send a 10 microsecond HIGH pulse to TRIG_PIN the sensor
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  // Read the ECHO_PIN  pin and measure the duration of the pulse
  long duration = pulseIn(ECHO_PIN, HIGH);
  distance = duration * 0.034 / 2;
  // Print the distance to the Serial Monitor
  Serial.println(distance);
  Serial.println(" cm");
  // Delay before the next reading
  delay(500);
}
```

IV.6.2. Bluetooth module (HC-05) interface

The HC-05 (or HC-06) is a common Bluetooth SPP (Serial Port Profile) module that acts as a wireless UART bridge. Its key pins are VCC, GND, TXD, RXD, and optionally KEY/EN and STATE.

According to the module's specifications, VCC can accept 4–6V (it has an onboard regulator), so it's usually powered from the Arduino's 5V pin.

The TXD pin (transmit data from the module) operates at 3.3V logic, and the RXD pin (receive data into the module) also expects

In wiring, you connect Arduino TX → HC-05 RXD and Arduino RX → HC-05 TXD. For example, one guide states: "Pin 3 (Arduino TX) goes to the RxD of the Bluetooth module, and pin 2 (Arduino RX) to the TXD". The HC-05's VCC goes to Arduino 5V, and GND to Arduino GND. Because the HC-05's RXD is 3.3V, a simple resistor voltage divider (or level shifter) should be used on the Arduino TX line to prevent 5V from feeding the module. The HC-05's TXD can be connected directly to the Arduino RX (the 3.3V it outputs is recognized as a high by the Arduino). In summary, power VCC=5V, GND=GND, then cross TX↔RX with proper level shifting [14].

Once wired and powered, the HC-05's LED will blink until a connection. In the Android app, pairing is done through the phone's Bluetooth settings (entering the PIN 1234 if prompted).

The HC-05 is connected to a microcontroller such as (Arduino) and we program it via the (Arduino IDE) application with the following code:

```
#include <SoftwareSerial.h>
SoftwareSerial bluetooth(10, 11);
void setup() {
  Serial.begin(9600); // Start serial communication with HC-05 and
Serial Monitor
}

void loop() {
  // Check if data is available from Bluetooth
  if (Serial.available()) {
    // Read the incoming character
    char c = Serial.read();

    // Print the character to the Serial Monitor
    Serial.print(c);
  }
}
```

IV.6.3. Controlling devices with Relay

The 4-channel relay module is a board containing four electronic switches (relays), used to control the on and off of high-voltage electrical devices using low-voltage control signals from microcontrollers such as the Arduino. They are connected in the order IN1, IN2, IN3, IN4, and 5 to pins 2, 3, 4, and 5.

The power component is connected to the COM port of the first relay, which connects to the power supply. The NO port remains connected to the COM port of the three relays, 2, 3, and 4, while the NO port of the three relays connects to their respective devices.

```
#define RELAY1 2 // Relay 1: main switch (master relay)
#define lamp 3    // Relay 2: controls the lamp
#define heater 4   // Relay 3: controls the heater
#define fan 5     // Relay 4: controls the fan

void setup() {

  for (int i = 2; i <= 5; i++) {
    // Configure pin as output
    pinMode(i, OUTPUT);
    // Set pin HIGH = turn OFF relay (because relay is active LOW)
    digitalWrite(i, HIGH);

  }

}

void loop() {

  // If a person is  detected
  if (personPresent) {
    digitalWrite(RELAY1, LOW);
    //Variable to determine the relay status ON or OFF
    mainPowerOn = true;
  }
  // If a person is not detected
  else if (!personPresent) {
    for (int i = 2; i < 5; i++) {
      // Turn OFF all devices (relays 1, 2, 3, 4)
      digitalWrite(i, HIGH);
    }
    mainPowerOn = false;
  }
  // Turn ON the light
```

```
if (command == "*light on#") {
  digitalWrite(lamp, LOW);
}
// Turn OFF the light
else if (command == "*off light#") {
  digitalWrite(lamp, HIGH);
}
// Turn ON the heater
else if (command == "*on heater#") {
  digitalWrite(heater, LOW);
}
// Turn OFF the heater
else if (command == "*off heater#") {
  digitalWrite(heater, HIGH);
}
// Turn ON the FAN
else if (command == "*on fan#") {
  digitalWrite(fan, LOW);
}
// Turn OFF the FAN
else if (command == "*of fan#") {
  digitalWrite(fan, HIGH);
}
// Turn OFF all devices
else if (command == "*system off#" || command == "*system of#") {
  for (int i = 2; i <= 5; i++) {
    digitalWrite(i, HIGH);
  }
}
```

IV.6.4. Final code

This Arduino sketch implements a voice-controlled relay system enhanced with presence detection: when a person enters the sensing zone (0–100 cm) as measured by an HC-SR04 ultrasonic sensor, the master relay (RELAY1) is energized to power downstream circuits. At that point, the HC-05 Bluetooth module listens for spoken commands—such as "*lights on#," "*heater off#," or "*fan on#"—and toggles the corresponding relays (LAMP, HEATER, FAN) accordingly. All relays default to OFF at startup to prevent unintended activation, and a continuous loop triggers ultrasonic pulses, measures echo time, computes distance, logs it over USB serial for debugging, and updates the system state. Voice commands are processed only while the master relay is active, ensuring that devices respond only when someone is present, and an emergency "*system off#" command instantly deactivates every relay and suspends further command handling.

```
//  Voice-Controlled Relay System with Ultrasonic Presence Detection
  ------------------------------------------------------------------------------------------------
  #include <SoftwareSerial.h>

//  Pin Definitions:

#define RELAY1   2
#define LAMP     3
#define HEATER   4
#define FAN      5
#define TRIG_PIN 9
#define ECHO_PIN 12
SoftwareSerial bluetooth(10, 11);
String command;
float distance;
bool mainPowerOn = false;

void setup() {

  Serial.begin(9600);
  bluetooth.begin(9600);

  //    Relay Pin Configuration:

  for (int pin = RELAY1; pin <= FAN; pin++) {
    pinMode(pin, OUTPUT);
    digitalWrite(pin, HIGH);
  }

  //    Ultrasonic Sensor Setup:

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  //    System Ready Notification:

  Serial.println("System Ready...");
}

void loop() {
  //    Ultrasonic Distance Measurement Sequence:

  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH);
```

```
distance = duration * 0.034 / 2;

//    Debug Logging:

Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
delay(500);

//    Presence Detection Logic:

bool personPresent = (distance > 0 && distance < 100);

//    Master Relay Control:
if (!personPresent && mainPowerOn) {
  for (int pin = FAN; pin >= RELAY1; pin--) {
    digitalWrite(pin, HIGH);
  }
  mainPowerOn = false;
}
if (personPresent) {
  digitalWrite(RELAY1, LOW);
  mainPowerOn = true;
}

//    Bluetooth Voice Command Handling:
  -
if (bluetooth.available() && mainPowerOn) {
  command = bluetooth.readStringUntil('\n');
  command.trim();
  command.toLowerCase();

  /* Lighting Control Commands */
  if (command == "*lights on#") {
    digitalWrite(LAMP, LOW);  // Activates lighting relay
  } else if (command == "*lights off#") {
    digitalWrite(LAMP, HIGH); // Deactivates lighting relay
  }

  /* Heating Control Commands */
  else if (command == "*heater on#") {
    digitalWrite(HEATER, LOW);  // Turns heater ON
  } else if (command == "*heater off#") {
    digitalWrite(HEATER, HIGH); // Turns heater OFF
  }

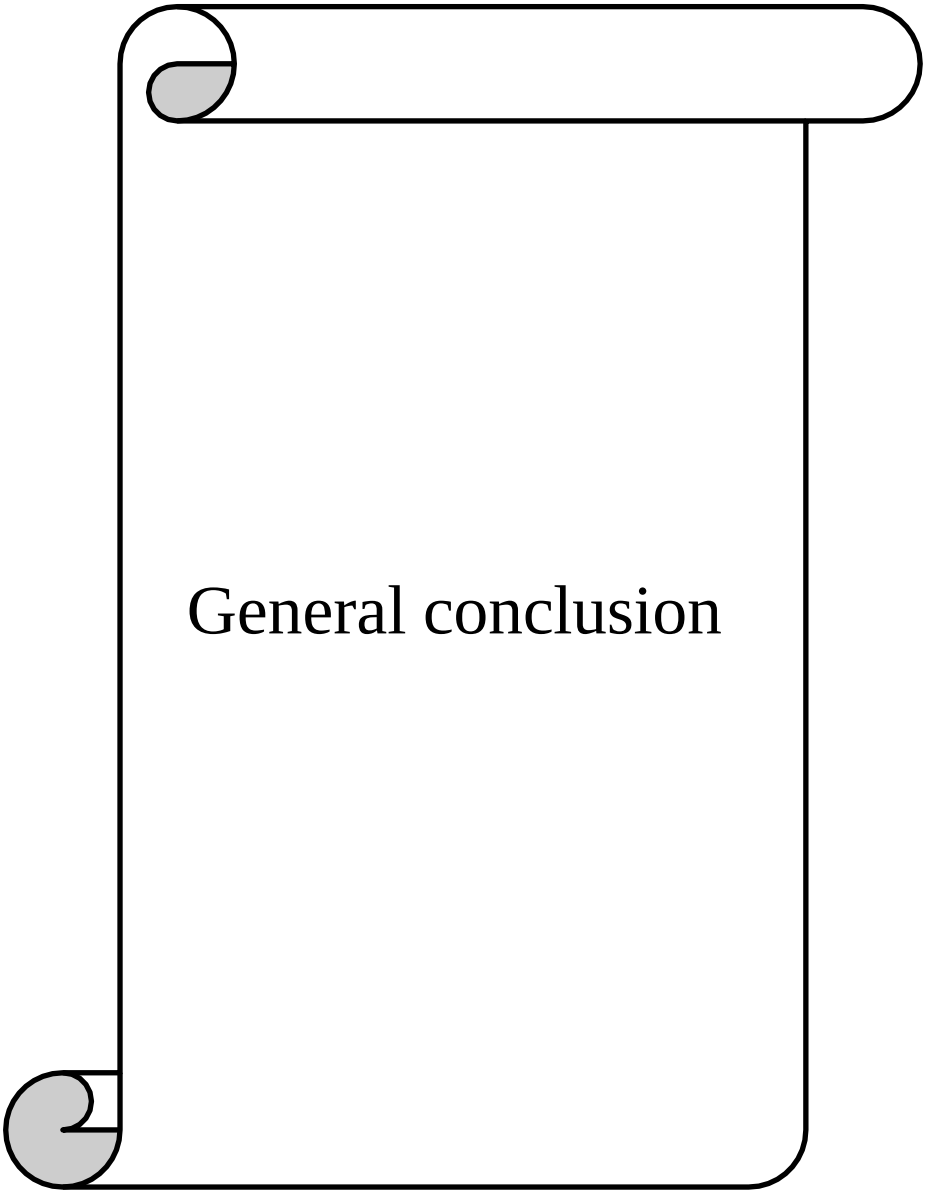
  /* Ventilation Control Commands */
  else if (command == "*fan on#") {
    digitalWrite(FAN, LOW);  // Turns fan ON
```

```
    } else if (command == "*fan off#") {
      digitalWrite(FAN, HIGH); // Turns fan OFF
    }

    /* Emergency System Shutdown */
    else if (command == "*system off#") {
      for (int pin = RELAY1; pin <= FAN; pin++) {
        digitalWrite(pin, HIGH); // Ensure all relays OFF
      }
      mainPowerOn = false;       // Disable further command processing
    }
  }
}
```

IV.7. Conclusion

This project succeeded through coordination between the project's organizational structure, system diagram, and programming code. The organizational structure facilitated the project's overall organization. The project diagram also provided a precise understanding of the relationships between system components and data flow, which helped design an integrated and efficient system. The project's code translated theoretical ideas into a practical program that operates accurately and smoothly, achieving the desired objectives.


General conclusion

General conclusion

During this study, we traced the development and application of voice command technology, from its inception to its practical applications. Starting with the fundamentals of sound production and audio signal processing, we explored how sound is captured, amplified, digitized, and processed by digital systems to enable accurate voice recognition. We then examined how these digital commands can be used in practical applications using components such as microphones, Bluetooth modules, ultrasonic sensors, and relays with an Arduino to create a smart voice-controlled system.

This project demonstrates the tremendous potential of integrating voice recognition technology with IoT and automation technologies. Integrating voice commands with smart devices enables instant interaction, increased efficiency, and intuitive control in various fields such as home automation, industrial applications, and public services. The success of this system lies in the careful coordination between hardware design, software implementation, and a clear system architecture.

Ultimately, the project represents more than just a technical achievement—it reflects the growing trend toward natural human-machine interaction. As voice control continues to evolve, it is expected to become an integral part of smart systems. In the following chapters, we will delve into the technical implementation and performance of the developed system, highlighting its practical impact and future potential.

Future outlook

- Support more voice commands, languages , and accents
- Offline voice control can improve reliability and privacy
- AI can make it smarter by learning user speech patterns
- Enable remote access and future cloud integration
- Voice commands to change system behavior or personality

Resource:

[1] Alexander S. Gillis, « What is voice recognition and how does it work? », Search Customer Experience, TechTarget, 28 août 2024.

[2] Ramil,T., Dabimel,N. N., Parimon, N, and Porle, R. R., *Simple speech controlled home automation system using android devices, Journal of Scientifc Research and Development, Vol.3, Issue 1, pp-33-38* (2016)

[3] Mishakova, E., Ivanov, D., et al., «Voice-Controlled Smart Home Automation Systems», *Journal of Physics: Conference Series,* IOP Publishing, 2019.

[4] Michel Vacher, Sybille Caffiau, François Portet, Brigitte Meillon, Camille Roux, Emmanuel Elias, Benjamin Lecouteux, Pedro Chahuara, « Evaluation of a Context-Aware Voice Interface for Ambient Assisted Living: Qualitative User Study vs. Quantitative System Evaluation », *ACM Transactions on Accessible Computing*, ACM, 2015.

[5] Huang, Z., Zweig, G., & Dumoulin, B., « Cache Based Recurrent Neural Network Language Model Inference for First Pass Speech Recognition », *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, Florence, Italy.

[6] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, Yoshua Bengio, «Speech Model Pre-training for End-to-End Spoken Language Understanding», 7 april 2019Electrical Engineering and Systems Science.

[7] Thierry Desot, François Portet, Michel Vacher, « End-to-End Spoken Language Understanding: Performance Analyses of a Voice Command Task in a Low Resource Setting», *Computer Speech & Language*, Elsevier, 2022.

[8] Gregory Ciccarelli, Jarred Barber, Arun Nair, Israel Cohen, Tao Zhang, « Challenges and Opportunities in Multi-device Speech Processing», 27 jun 2022, Electrical Engineering and Systems Science**.**

[9] Mary Hewitt, Hamish Cunningham, «Taxonomic Classification of IoT Smart Home», 24 OCT 2022 Electrical Engineering and Systems Science.

[10] L. Coetzee & J. Eksteen, « The Internet of Things – Promise for the Future? An Introduction», *2011 IST-Africa Conference Proceedings*, IEEE, 2011, Afrique du Sud.

[11] K. S, S. J, H. S and N. K. G, "An IoT-Based Power Monitoring Solution with Real-Time Data Display," *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2024, pp. 551-554, doi: 10.1109/ICCES63552.2024.10860009.

[12] Charith Perera, Arkady Zaslavsky, Peter Christen, Dimitrios Georgakopoulos, « Sensing as a Service Model for Smart Cities Supported by Internet of Things », Transactions on Emerging Telecommunications Technologies, John Wiley & Sons, janvier 2014.

[13] Jordi Salazar et Santiago Silvestre, Internet de las cosas, Published by: České vysoké učení technické v Praze, first edition, 2017.

[14] Nisar, K., Ibrahim, A.A.A., Wu, L., Adamov, A., and Deen. M.J. (2016). Smart home for elderly living using wireless sensor networks and an android application. *Proceedings of the IEEE 10th International Conference on Application of Information and Communication Technologies (AICT).*

[15] Dallmer-Zerbe, S. andHaase, J. (2021). Adapting smart home voice assistants to users' privacy needs using a Raspberry-Pi based and Self-Adapting system. In 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE), pages 1–6.

[16] Mestouri, Abdellatif Benlamkadem, Development of a Smart Home System for Comfort, Security and Energy Savings Based on Arduino Microcontroller, Engineering, Computer Science, 2017.

[17] Minder, B., Wolf, P., Baldauf, M. *et al.* Voice assistants in private households: a conceptual framework for future research in an interdisciplinary field. *Humanit Soc Sci Commun* **10**, 173 (2023). https://doi.org/10.1057/s41599-023-01615-z

[18] Rabelo, L.P., Sodré, D., dos Santos, M.S. *et al. ForAlexa*, an online tool for the rapid development of artificial intelligence skills for the teaching of evolutionary biology using Amazon's Alexa. *Evo Edu Outreach* 15, 10 (2022). https://doi.org/10.1186/s12052-022-00169-z

[19] V. Këpuska and G. Bohouta, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2018, pp. 99-103, doi: 10.1109/CCWC.2018.8301638.

[20] S. Imran Hussain, S. Deepalakshmi, R. J. Benilla and V. Charu Nivetha, "Automation of Smart Home using Smart Phone via Google Assistant," *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2023, pp. 499-505, doi: 10.1109/ICSSIT55814.2023.10060979.

[21] Fredrik Norberg, Zelal Yldirim, «The implementation of Voice Command in Smart Homes», mémoire, KTH University, Industrial Engineering and Management, Suède, 2022.

[22] Mike Cook, « Arduino Music and Audio Projects », *Foreword by Massimo Banzi and David Cuartielles*, Edition, Maker Media, 2012, USA.

[23] Vijay Verma , «+50 Sensors To Use With Arduino: All Sensors». Self-published, 2021.