Ministry of Higher Education and Scientific

Research University of Hamma Lakhdar

El Oued

Faculty of sciences and technology

Department of Electrical Engineering

Academic Master's Thesis

To obtain the Master's degree awarded by **Hamma Lakhdar El Oued**

Specialty: **Telecommunication Systems**

Control of an Elevator by a Microcontroller

Presented by:

Fares Okba

June 2024

Jury members:

Mr. AJGOU Riadh	PROF	Chairman	UHLElOeud
Mr. Khalil abdelatif	PROF	Examinator	UHLElOeud
Mrs. Boukaous Chahra	MCA	Supervisor	UBS Constantine 3
Mr. HimaAbdelkader	MCA	Co-supervisor	UHL ElOeud

Academic year: 2023/2024

Dedications

I dedicate this work to:

To my mom: if anyone deserves this dedication the most, then it should be my mom

To My dad thank you for all of the sacrifice, you did it for us i am really grateful without you

i won't be where i am now

To all of my brothers and sisters, Thank you for always standing by my side and for your unwavering belief in my abilities.

An last but definitely not least, to my self, to the hardworking, passionate, determined guy that loves challenges, ME.

Fares

Acknowledgments

First and foremost, I would like to thank God for providing me with the strength, wisdom, and perseverance to complete this work. Your guidance and blessings have been my source of inspiration and hope throughout this journey.

A big thanx to ***Dr Boukaoues Chahra*** *for supervising this work.*

I would like to thank my supervisor, ***Dr Hima Abdelkader****, professor at the University of ElOeud, for the time he devoted, providing the necessary advice and ideas for the conduct of this thesis.*

My gratitude also goes to the teachers of our Department of Electrical Engineering for their contributions to our education, and their patience with us throughout the years.

Abstract:

The elevator is a mobile device allowing the movement of people or objects in a car on a predefined vertical axis within a multi-storey construction. The objective of this master thesis is to study and produce a prototype for controlling an elevator by Arduino uno, especially since the elevator is an interesting automated system and that its realization calls for several technological fields (automatic, computer, mechanical, etc), moreover, it is a means of transport very used and more and more widespread.

Keywords: Elevator Automation, Vertical Transport, Elevator Prototype, Arduino Uno ,Microcontroller, Automated System, Circuit Boards.

Résumé

L'ascenseur est un dispositif mobile permettant le déplacement des personnes ou des objets dans une cabine sur un axe vertical prédéfini au sein d'une construction à plusieurs étages. Ce mémoire a pour objectif d'étudier et de réaliser un prototype de commande d'un ascenseur par Arduino uno, l'ascenseur est un système automatisé intéressant et sa réalisation fait appel à plusieurs domaines technologiques (automatique, informatique, mécanique, etc), de plus, c'est un moyen de déplacement très utilisé et de plus en plus répandu.

Mots-clés : Automatisation d'ascenseur, prototype, transport vertical, Arduin Uno, microcontrôleur, système automatisé, cartes de circuits imprimés.

الملخص:

المصعد عبارة عن جهاز متحرك يسمح بحركة الأشخاص أو الأشياء في حجرة على محور عمودي محدد مسبقًا داخل مبنىمتعدد الطوابق. الهدف من هذه الأطروحة هو دراسة وإنتاج نموذج أولي للتحكم في المصعد بواسطة أردوينو اونو ،خاصة وأن المصعد هو نظام آلي مثير للاهتمام وأن تحقيقه يتطلب العديد من المجالات التكنولوجية (آلي، كمبيوتر، ميكانيكي، إلخ،) علاوة على ذلك، فهي وسيلة نقل مستخدمة للغاية وأكثر انتشارا.

الكلمات المفتاحية :اتمتت المصاعد, نموذج تجريبي , النقل العمودي ,اردوينو اونو , ميكروكونترولر , متحكم ذكي, مصعد .

Table of contents

Table of the figure

Table of charts

General Introduction

General Introduction:

The elevator is a significant invention that has grown with the development of multi-story buildings, including governmental and residential structures. The elevator is both an incredible invention and an everyday way of mobility.

Elevators are now used to connect floors, replacing traditional stairwells. Elevators satisfy current criteria for autonomy, mobility, accessibility, and speed. It reduces tiredness, saves time, and makes travel and shopping more convenient. It enables independent living at home for older people and those with limited mobility.

Elevators was not always the fast and advance lifting devices that we know today. Originally, this technique was used to reach remote locations, including Greek monasteries and mines. As civilization increased, the elevator was continually improved to better fulfill its demands. Elevators are now programmable, making them easier to use, safe, energy-efficient, roomy, and accessible to the elderly and disabled. They are also more intelligent and can handle unexpected traffic, encouraging optimal and smoother traffic flow.

The use of microprocessors has revolutionized elevator operating systems.

Elevator control using electronic components, such as microprocessors and microcontrollers, has been developed to provide a modern, simple, affordable solution.

The objective of our work is to create a control system that manages the elevator's resources while meeting quality and operational safety standards.

This memory has three key chapters:

The first chapter contains an introduction of elevators, including a brief history and a discussion of its many components. This chapter will cover the many types of elevators, their benefits and drawbacks, as well as their features and safety requirements based on their application.

The second chapter will be focused on introducing the microcontroller, including its characteristics and features, a historical background, a suggestion, and the method employed in this project.

In the third chapter, we discussed the circuit installation, programming and explained the algorithm used in this work.

Finally, in conclusion, we summarized the entire work and also mentioned tips for future improvements.

Chapter I:
General information about the elevators

I.1.Introduction

The simple yet incredibly useful elevator has completely changed how people move across the towering spaces in our built world. Elevators have advanced in complexity and sophistication from their humble origins as simple lifting systems to today's breathtaking skyscrapers.

This chapter provides an advanced overview of elevator technology. We start with a quick historical refresher, then define the elevator and its many components. Then we describe the varieties of elevators and reveal their advantages, disadvantages and regions of application.

We complete this chapter by providing their features according to their usage, as well as its components and parts.

I.2. Evolution of elevators

The idea of a lifting device that will allow heavy loads to be moved vertically has always been one of man's worries, in fact, the Romans and ancient Egyptians invented the first models of freight elevators, in order to to facilitate the transport of gladiators and wild beasts for the former, and the materials necessary for the construction of the pyramids, for the latter. The earliest steam elevators were built in mines but were shortly abandoned due to their unreliability.

In 1854, the American Elisha Graves Otisa invented an elevator with protection against any fall: the parachute brake, the elevator finally became safe, and could be trusted be put in public spaces [1].

In 1864, the Frenchman Léon Edoux identified his hydraulic device, which employs water under pressure as energy, by the term “Elevator”.

The year 1889 was noted by the inauguration of the Eiffel Tower, with an elevator exceeding all performances attained till then. The latter was then recognized for its height and speed: 0.8 m/s, Léon Edoux and the Otis brothers are the inventors of this technological achievement [1].

In 1924, the automated elevator made its debut in England. It is a device that does not require machinists, is totally electronic, and has more safety features. It is usually found in most buildings [1].

The elevator became industrialized around 1970 to better satisfy the demands of increasing population.

In the 1980s, the introduction of computers and microprocessors revolutionized elevator operating. Elevators are now programmable, making them more pleasant, quicker, safer, energy-efficient, roomy, and accessible to older people and disabled [1].

The elevators that were made and installed in the 1990s went through a new development; they grew more intelligent to encourage optimal traffic and learnt to handle the unexpected thanks to the participation of microcomputing [2].

The cabin has been modified on the interior, with new control panels, improved lighting, more diversified and appropriate ornamental materials, and better signage that improves user information and makes it easier to use.

In the 2000s, the prospect of more unique design, along with technical developments that allowed for greater equipment compactness, supported the introduction of concepts such as elevators without machine rooms. In this novel idea, the elevator is reduced to a volume: the shaft and it has various advantages, such as:

- During construction: no machinery room and the comparable surface space is recovered for marketing purposes (cellar, parking, or livable square meter).
- For installation in existing buildings: no structural modification or surface recovery required for the machinery, therefore simplicity of implementation.

In addition to these advances in architectural integration, advances enable more "intelligent" elevators that anticipate destination management, leading to dramatically improvement in traffic in buildings [1] .

I.3. Definition of elevator:

An elevator is a vertical lifting mechanism used for carrying people or cargo between floors of a building. It features a cabin powered by an electric motor via a metal wire; there are really three sorts of devices:

- Elevators are designed for human use only;
- Elevator that designed for products and humans;
- Industrial freight elevators intended for single goods [2].

I.4. Basics of Elevator Classification:

There are two types of elevators in common use today.

I.4.1. Hydraulic type elevators:

The hydraulic elevators are designed to lift a car using a hydraulic ram using a fluid-driven piston mounted inside a cylinder. The cylinder is connected to a fluid-pumping system, as shown in the figure I.1. [2].

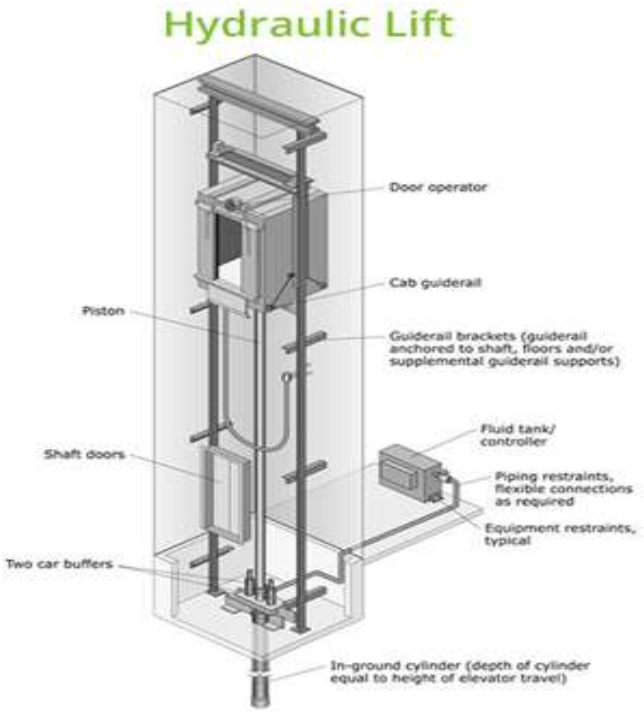

Fig I.1. Hydraulic elevator [3].

The hydraulic system has three key parts:

- A tank (which is the fluid reservoir);
- A pump (powered by an electric motor);
- A valve.

The pump moves fluid from the tank into a tube that leads to the cylinder. When the valve is opened, the pressured fluid follows the path of least resistance and returns to the fluid

reservoir. However, when the valve is closed, the pressured fluid has nowhere to go but inside the cylinder. As the fluid accumulates in the cylinder, it pulls the piston upward, elevating the elevator car [4].

When the car approaches the correct floor, the control system sends a signal to the electric motor, which gradually turns off the pump. With the pump turned off, no additional fluid enters the cylinder, but the fluid that is currently inside cannot exit. The piston sits on the fluid, and the car remains still .

To lower the car, the control system for the elevator sends a signal to the valve. A simple solenoid switch operates the valve electrically. When the solenoid opens the valve, the fluid collected in the cylinder can flow into the fluid reservoir. The weight of the car and its load presses down on the piston, pushing the fluid into the reservoir. The car gradually descended. To stop the car at a lower floor, the control system closes the valve once again.

I.4.2. Roped elevators:

It is the most common elevator type (figure I.2); the car is lifted and lowered by traction steel cables. The cables are attached to the elevator car and looped around a pulley. The pulley grabs the ropes, causing the ropes to move when the electric motor turns the pulley. When the engine spins one way, the pulley raises the elevator; when the engine spins the opposite direction, the pulley lowers the elevatorand the following figureis a Cable elevator type.[2]

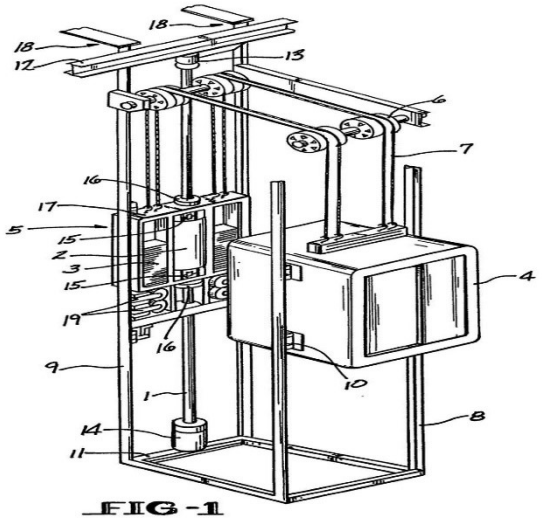

Figure 2. Cable elevator [5].

There are two types of cable:

- Geared, the motor rotates the sheaves directly;
- Gearless, the motor turns a gear train that rotates the sheave [2] .

Roped elevators are highly more efficient than hydraulic elevators, with better safety features.

I.5. Use of elevator:

Elevators are currently created based on their expected usage, therefore elevators:

- Apartment buildings with a capacity suitable for 8 people prioritize functionality and appearance.
- Those of offices and hotels are more refined, simple and quick with rotations of entrances and exits.
- Factory elevators are designed more for the transport of heavy loads (in tons) more than people.
- When elevators are installed in public places and panoramic cabins (airports, educational establishments, Eiffel-type panoramic towers, etc.), they are transparent, in particular because of a security aim for the occupants. But apart from that, it also aims to give the opportunity to admire the exterior panorama
- Elevators at hospitals and hospices are designed to fulfill the capacity and comfort requirements of transportation as well as the patient's condition.

I.6.Construction of an elevator:

Regardless of the type and intended use, the elevator is supposed to include the following elements:

I.6.1. Car of the elevator:

The cabin is the elevator's main piece of equipment; it welcomes and carries passengers as they go vertically. It is designed with precise arrangements that are well-suited to its intended usage; its layout must be functional and pleasant. Depending on the demands, the cabin may take on different elements; it can be fitted with a control panel, lighting system, a ventilation device, and a safety braking system, as well as other accessories that contribute to the comfort and safety of users.

I.6.2. Hopper of the elevator:

The space, place or path in which the elevator car moves.

I.6.3. Machinery room:

The room containing the motor and elevator control panel [2].

I.6.4. Movement transmission system:

It is a system which includes a reduction box, a pulley, a wired mat with a counterweight or a drum winch or a Revin and finally a piston for hydraulic type elevators.

The movement of the cabin is controlled by a button activated by people inside the cabin or outside, in other words on the landing, for the call, these buttons are located in an electrical circuit connected to a cabinet control.

For moving the elevator car, there are three types of technologies. In the case of the electric elevator, we use an electric motor, a pulley system, a counterweight, and cables.

For the hydraulic elevator, we also have an electric motor, a hydraulic pump, an oil tank and cables.

For the elevator without machinery we have a compact electric motor without gearbox integrated into the shaft, and thanks to which the cabin goes up and down without a mechanical winch, it is controlled via a frequency variator which injects current by modulating the voltage and frequency so as to move the cabin at variable speeds [2].

I.6.5. Landing doors:

These are secure doors that only open when the elevator is present, employing a latch system to trigger their opening and shutting, preventing them from opening into an empty shaft and causing tragic falls.

I.6.6.Parachute:

It is a system of clamps which slows down the speed of the cabin and can even stop it in the event of an alert for too high speed.

I.6.7. Control system:

The control system plays the role of the brain. It is a system for closing and opening landing doors as well as moving and stopping the cabin with an emergency stop device. However, a remote monitoring system can be installed in the cabin and can serve as an in-cab

control and communication system, which allows the central unit to monitor and carry out remote diagnostics on the main component of the installation

I.7. Sizing an Elevator:

Sizing an elevator requires determining the necessary size, capacity, and requirements to satisfy the needs of a building and its occupants. The size process takes into account a variety of elements, including the kind of building, estimated traffic, and particular requirements like accessibility.

I.7.1.Building Type and Usage:

- **Residential Buildings**: Elevators in residential buildings are generally lower in capacity and built for modest use.
- **Commercial Buildings**: To handle more traffic, these buildings require elevators with larger capacity and faster speeds.
- **Industrial Buildings**: Elevators may be required to lift huge loads and built to transfer commodities rather than passengers.

I.7.2. Capacity:

- **Passenger Capacity**: This is commonly assessed in terms of the elevator's capacity. Common capacities range from four to twenty persons.
- **Weight Capacity**: Measured in kilograms or pounds, typical capacities range from 320 kg (700 lbs) to 2500 kg (5500 lbs) [2].

I.7.3. Car Dimensions:

- The dimensions of the elevator car should be sufficient to comfortably fit the maximum number of passengers or the expected cargo.
- Standard dimensions can vary, but for a typical passenger elevator, a car might be around 1.1 to 2.1 meters wide and 1.4 to 2.7 meters deep [2].

I.7.4. Speed:

The speed of the elevator is an important factor, especially in taller buildings. Speeds range from 0.5 meters per second (m/s) in low-rise buildings to up to 10 m/s in high-rise buildings.

I.7.5. Travel Distance and Stops:

The total travel distance of the elevator shaft and the number of stops it will make must be considered. High-rise buildings require more powerful elevators with the ability to travel long distances quickly and efficiently.

I.7.6. Door Dimensions and Opening Type:

Standard door widths range from 800 mm to 1100 mm.Door types include center-opening, side-opening, and telescopic opening doors.

I.7.7. Code and Regulatory Compliance:

Elevators must comply with local and international safety standards and accessibility requirements. Standards such as the Americans with Disabilities Act (ADA) in the United States ensure that elevators are accessible to all individuals.

I.8. Explanation of How an Elevator Works:

I.8.1.Calling the Elevator:

- **Press Call Button**: Passengers press the call button on their floor.
- **Elevator Assignment**: The control system assigns the nearest available elevator car to respond.

I.8.2. Elevator Arrives:

- **Movement**: The motor and drive system move the elevator car to the requested floor.
- **Stopping**: The car slows down and stops precisely at the floor, guided by sensors.
- **Doors Open**: The doors open automatically, allowing passengers to enter or exit.

I.8.3. Entering and Selecting Destination:

- **Passenger Entry**: Passengers enter the elevator car.
- **Select Floor**: Passengers select their destination floor using the control panel.
- **Doors Close**: The doors close automatically after a brief delay or when the door-

closebutton is pressed.

I.8.4.Transporting Passengers:

- **Initiate Movement**: The control system directs the car to the selected floor.
- **Guidance**: The car moves along guide rails, and in traction elevators, a counterweight aids in efficient movement.
- **Speed Control**: The elevator accelerates to a smooth ride speed and maintains it during travel.

I.8.5. Arriving at the Destination

- **Approaching Floor**: The car slows down as it nears the selected floor.
- **Stopping**: The car stops accurately at the floor, with the floor level aligned.
- **Doors Open**: The doors open automatically for passengers to exit.

I.8.6. Safety Mechanisms :

- **Emergency Brakes**: Activate to stop the car safely in case of a malfunction or over speed.
- **Sensors**: Door sensors prevent closing if an obstruction is detected, and interlocks keep doors closed during movement.
- **Maintenance**: Regular checks ensure all components are functioning correctly, and monitoring systems detect issues for prompt repair.

I.9. Conclusion:

Through this chapter, a general overview of elevators and their components has been provided. Presented, with an exhibition of the different types of elevators of which we have cited the advantages and disadvantages of each. In the next chapter, our study will be based on the design approaches of the elevators control system based on a microcontroller ATmega328Pthat exist in the Arduino uno.

ChapterII : The controller

II.1. Introduction:

With advancements in electronics and computing, command and control systems have significantly improved, leading to the development of microcontrollers and microprocessors that can seamlessly integrate into industrial processes, such as elevator control. These controllers can perform increasingly sophisticated tasks, including control, data processing, data movement, and simulation, under optimal conditions and with high safety standards. As technology progressed, these controllers became smaller, culminating in the creation of the microcontroller, which has simplified tasks and made previously unattainable objectives achievable.

II.2. Definition of Microcontroller:

A microcontroller is a compact integrated circuit designed to manage specific functions within an embedded system. It typically incorporates a central processing unit (CPU), memory (both RAM and ROM), and input/output (I/O) peripherals all on a single chip. This integration enables microcontrollers to efficiently execute dedicated control tasks, such as reading sensor inputs, processing data, and controlling actuators. They are essential components in various applications, ranging from household appliances and automotive systems to industrial automation and robotics. Microcontrollers are prized for their ability to perform real-time operations with minimal power consumption, making them ideal for battery-powered devices. Their programmability allows for flexibility and customization, enabling precise control over the targeted operation. As a result, microcontrollers play a crucial role in modern technology, simplifying complex tasks and enhancing the functionality and efficiency of electronic devices and the following figure (Figure II.1) shows a real Micro controller.

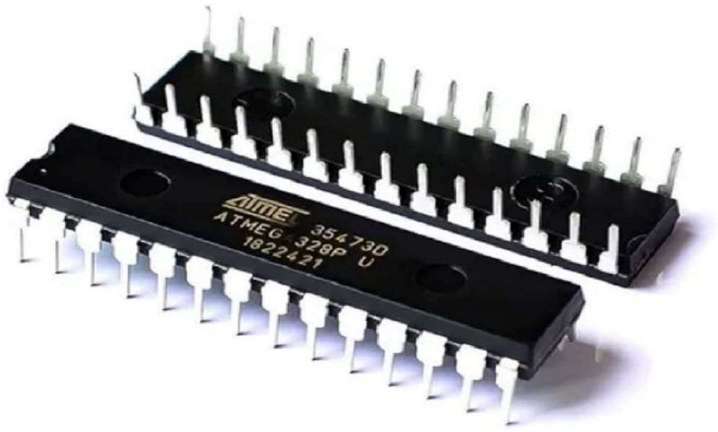

Figure II.1. Micro controller [6].

II.3. Different Models of Microcontrollers:

II.3.1. Atmel (now Microchip Technology) :

The ATmega328, a widely used microcontroller in the Arduino Uno, features 32KB of flash memory, 1KB of EEPROM, and 2KB of SRAM. Its versatility and ease of use make it popular for various DIY and educational projects. In comparison, the ATmega2560, found in the Arduino Mega, offers 256KB of flash memory, 4KB of EEPROM, and 8KB of SRAM. This significant expansion in capabilities makes it ideal for more complex applications requiring additional resources. On the smaller end of the spectrum, the ATtiny85 is a compact microcontroller with 8KB of flash memory, 512 bytes of EEPROM, and 512 bytes of SRAM, perfect for small projects where space and power efficiency are crucial.

When looking at Microchip Technology's offerings, the PIC16F877A stands out in both industrial and academic projects. It provides 14KB of flash memory, 368 bytes of SRAM, and multiple I/O ports, making it a robust choice for a wide range of applications. Meanwhile, the PIC18F4550 features a USB interface, 32KB of flash memory, 2KB of SRAM, and advanced peripherals for communication and control. This makes it particularly well-suited for projects that require sophisticated connectivity and control capabilities.

Comparing these microcontrollers across brands, the ATmega328 and PIC16F877A both cater to similar segments with a focus on versatility and ease of use. The ATmega2560 and PIC18F4550, with their higher memory capacities and advanced features, target more demanding applications. The compact ATtiny85 offers a niche solution for space-constrained projects, whereas the PIC18F4550’s USB interface sets it apart for connectivity-focused applications. This range of options ensures that developers can find the right microcontroller for their specific needs, whether they require simplicity, advanced functionality, or compact design.

II.3.2. NXP Semiconductors

The LPC1768, an ARM Cortex-M3-based microcontroller, is renowned for its low power consumption and extensive I/O capabilities, making it a popular choice for applications where energy efficiency and peripheral connectivity are paramount. On the other hand, the Kinetis K20, an ARM Cortex-M4-based microcontroller, caters to applications requiring

higher processing power and advanced peripherals. While the LPC1768 excels in power-sensitive and I/O-intensive tasks, the Kinetis K20 stands out in scenarios demanding robust computational performance and sophisticated peripheral integration. Together, these models offer a spectrum of options for developers, from energy-efficient designs to high-performance applications, ensuring that the right microcontroller can be chosen based on specific project requirements.

II.4.Arduino:

Arduino is an open-source electronics platform with simple hardware and software. It comprises of a microcontroller board and an integrated development environment (IDE) that allows you to write and upload code to the board. There are different models of Arduino and the following figureII.2 shows an Arduino uno:

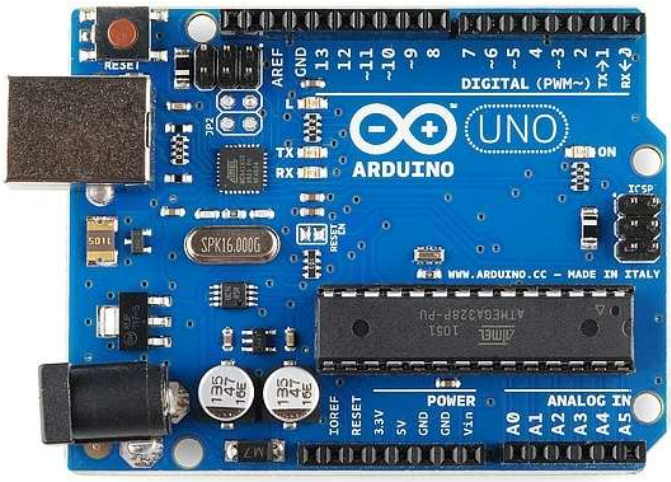

Figure II.2. Arduino uno.

II.4.1Arduino Uno

The ATmega328P, central to the Arduino Uno platform, features 14 digital I/O pins (six of which support PWM), and six analog input pins, making it versatile for tasks ranging from motor control to sensor interfacing. It offers 32KB of flash memory for program storage, 2KB of SRAM for dynamic data handling, and 1KB of EEPROM for persistent data storage. Operating at a standard 5V, it ensures compatibility with a wide range of components. This combination of digital and analog capabilities, ample memory, and reliable operating voltage makes the ATmega328P a powerful choice for diverse embedded systems and electronic projects.

II.4.2. Arduino Mega 2560

The ATmega2560, integral to the Arduino Mega, boasts 54 digital I/O pins, including 15 PWM outputs, making it highly suitable for complex projects requiring numerous connections and precise control, such as robotics and automation systems. With 16 analog input pins, it can interface with a wide array of sensors, providing detailed environmental feedback. It features a substantial 256KB of flash memory for extensive program storage, 8KB of SRAM for efficient data handling during program execution, and 4KB of EEPROM for reliable storage of configuration settings and calibration data. Operating at 5V, it ensures compatibility with standard electronic components and modules. This combination of extensive I/O capabilities, large memory resources, and standard operating voltage makes the ATmega2560 a robust and versatile microcontroller for advanced embedded systems and electronic projects.

II.4.3. Arduino Nano

The ATmega328, a key component of the Arduino Uno, offers 14 digital I/O pins, including 6 PWM outputs, making it ideal for projects involving motor control, LED dimming, and other tasks requiring precise signal modulation. It also features 8 analog input pins, allowing it to interface with various sensors for detailed data collection. With 32KB of flash memory, it provides ample space for complex programs, while its 2KB of SRAM ensures efficient handling of dynamic data during execution. Additionally, the 1KB of EEPROM allows for the storage of critical data such as configuration settings and calibration parameters. Operating at a standard 5V, it is compatible with a wide range of components and modules, making the ATmega328 a versatile and reliable choice for numerous embedded systems and electronic projects.

II.4.4. Arduino Leonardo

The ATmega32u4, commonly used in the Arduino Leonardo, features 20 digital I/O pins with 7 PWM outputs, making it suitable for tasks requiring precise control such as motor management and LED dimming. It includes 12 analog input pins, allowing it to interface with a wide range of sensors for comprehensive data acquisition. The microcontroller provides 32KB of flash memory for storing extensive programs, 2.5KB of SRAM for efficient data handling during execution, and 1KB of EEPROM for persistent storage of important configuration settings and calibration data. Operating at a standard 5V, it ensures

compatibility with numerous components and modules. Additionally, the ATmega32u4 includes built-in USB communication, enabling direct interface with computers for projects that require USB connectivity, such as custom HID devices. This blend of digital and analog capabilities, ample memory, and integrated USB support makes the ATmega32u4 a powerful and versatile microcontroller for a wide range of embedded applications.

II.4.5. Arduino Due

The Atmel SAM3X8E ARM Cortex-M3, used in the Arduino Due, features 54 digital I/O pins, including 12 PWM outputs, making it ideal for complex control tasks such as robotics and automation. It offers 12 analog input pins for interfacing with various sensors and 2 analog output pins with DAC (Digital-to-Analog Converter) capability, allowing for high-precision analog signal generation. With a substantial 512KB of flash memory, it supports large and sophisticated programs, while its 96KB of SRAM ensures efficient data processing and storage during execution. Operating at 3.3V, it is compatible with modern low-voltage components, offering enhanced power efficiency. This combination of extensive I/O capabilities, large memory resources, and advanced features like DAC output makes the SAM3X8E a powerful and versatile choice for advanced embedded systems and electronic projects.

II.4.6. Arduino Pro Mini

The ATmega328, a staple in the Arduino Uno, offers 14 digital I/O pins with 6 PWM outputs, making it well-suited for applications like motor control and LED dimming. It also features 8 analog input pins, providing the capability to interface with a variety of sensors for detailed data acquisition. With 32KB of flash memory, it accommodates extensive programs, while its 2KB of SRAM ensures efficient handling of dynamic data during execution. Additionally, the 1KB of EEPROM allows for persistent storage of critical data, such as configuration settings and calibration parameters. The ATmega328 operates at either 3.3V or 5V, enhancing its versatility and compatibility with a wide range of components and modules. This flexibility in operating voltage, combined with its ample memory and I/O capabilities, makes the ATmega328 a robust choice for diverse embedded systems and electronic projects.

II.4.7. Arduino MKR Series

Microcontrollers in the MKR series, such as the SAMD21 Cortex-M0+ used in the MKR Zero and MKR1000, offer advanced connectivity and substantial memory resources

tailored for IoT applications. These microcontrollers often feature built-in communication modules, including Wi-Fi, LoRa, and GSM, enabling seamless integration with wireless networks and remote data transmission. For instance, the MKR1000 includes 256KB of flash memory for extensive program storage and 32KB of SRAM for efficient data handling and processing. This robust combination of versatile connectivity options, significant memory capacity, and the low-power, high-performance ARM Cortex-M0+ architecture makes the MKR series ideal for developing sophisticated, connected embedded systems and IoT projects.

II.5. ATmega328P in general and Arduino in particular:

We chose the ATmega328P (FigureII.3) in general and the Arduino in particular for several reasons:

II.5.1. Ease of Use

- **User-Friendly**: The ATmega328P, combined with the Arduino platform, provides an accessible and straightforward environment for both beginners and experienced users.
- **Simplified Programming**: The Arduino IDE simplifies the process of writing and uploading code, making microcontroller programming less daunting.

II.5.2. Community Support

- **Extensive Resources**: The ATmega328P has a large, active community. This means extensive libraries, tutorials, forums, and examples are available, which greatly aid in development and troubleshooting.

II.5.3. Cost-Effectiveness

- **Affordable**: The ATmega328P is cost-effective, making it a popular choice for a wide range of applications, especially in educational and hobbyist projects.

II.5.4 . Versatility

- **Rich Peripheral Set**: The ATmega328P includes a variety of peripherals (e.g., digital I/O, analog inputs, PWM outputs, timers, and communication interfaces), making it suitable for diverse applications and the Figure 3 shows theatmega328 pins.

- **Memory and Speed**: It offers sufficient memory (32KB flash, 2KB SRAM, 1KB EEPROM) and speed (16MHz clock) for many embedded applications.

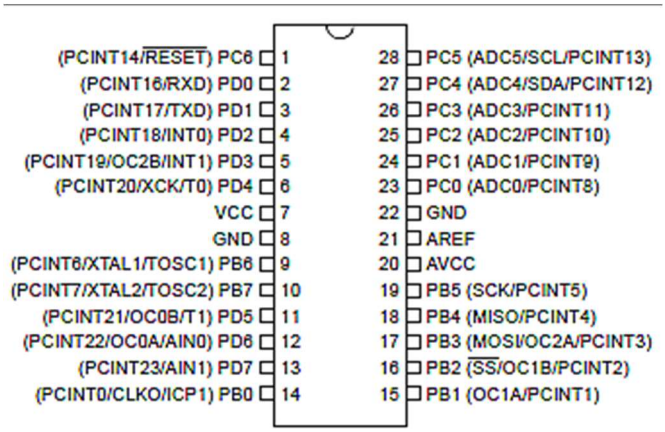

Figure II.3. Atmega328 pins [7].

II.6. Use Arduino to Make Elevator

The Arduino Uno, which uses the ATmega328P microcontroller, is a popular choice for creating a variety of control systems, including a basic elevator. Its popularity stems from a number of major characteristics, yet it also has certain drawbacks.

II.6.1. Strengths:

- **Ease of Use**: The Arduino platform is designed to be user-friendly, enabling beginners to start with microcontroller programming without a steep learning curve.
- **Community Support**: Extensive resources, tutorials, and libraries are available, significantly aiding development and troubleshooting.
- **Cost-Effectiveness**: The Arduino Uno is affordable, making it accessible for educational purposes, hobbyists, and small-scale projects.
- **Versatility**: It offers a rich set of peripherals, including digital I/O, analog inputs, PWM outputs, and various communication interfaces, suitable for a wide range of applications.
- **Simplified Programming**: The Arduino IDE simplifies writing and uploading code, allowing for rapid prototyping and quick iterations.

II.6.2. Weaknesses:

- **Limited Memory**: With 32KB of flash memory, 2KB of SRAM, and 1KB of EEPROM, it can be a constraint for more complex or resource-intensive applications.
- **Processing Speed**: The 16MHz clock speed might not be sufficient for tasks requiring higher performance or faster execution times.

II.7. Arduino Uno specifications :

II.7.1. Alimentation de la carte ARDUINO :

The ARDUINO UNO board may be powered via USB or an external power supply. The source is picked automatically. The external supply voltage (excluding USB) can be from an AC-DC adaptor or batteries.

The adapter is attached by a 2.1 mm positive "jack" connection in the middle. To connect to a battery pack, utilize the power connector's GND and VIN terminals.

The card can operate using an external voltage of 7 to 12 Volts. The power pins are as follows :

- VIN (to be distinguished from the 5V of the USB connection of the USB connection or 5V regulated). This is the positive input voltage when the ARDUINO board is used with an external voltage source (as distinguished from 5V from the USB connection or other regulated 5V source).

You can power the board using this pin, or, if power is supplied by the power jack, access the power voltages on this pin).

- 5V: this is the regulated voltage used to operate the microcontroller and the other components of the card (the regulated voltage is obtained using a regulator integrated into the ARDUIONO card).

The 5V provided by this pin can thus originate from the supply voltage VIN via the card regulator, the USB connection (which offers regulated 5V), or any other regulated power source.

- 3V3: a 3.3V power supply is provided by the FTDI integrated circuit (integrated circuit adapting the signal between the USB port of your computer and the serial port

of the ATmega) of the card; this is relevant for some external devices that require this voltage or instead of 5V.

The maximum current available on this pin is 50 mA.

- GND: ground pin (or 0V).

II.7.2. Digital inputs/outputs:

Each of the UNO board's 14 digital pins (numbered 0 to 13) can be used as either a digital input or a digital output, using the PinMode() instructions; DigitalWrite() and DigitalRead() of ARDUINO language. These pins operate at 5V. Each pin can supply or receive a maximum of 40mA of current and has an internal “pull-up” resistor (disconnected by default) of 20-25 KOhms[9][8].

This internal resistor activates on an input pin using the DigitalWite instruction (pin, HIGH).

Between these pins there are those which have additional functionalities:

- **Serial communication (**Pin 1 and 2**):**

These pins used to receive (RX) and transmit (TX) TTL level serial data.

Its relatives are connected to the corresponding pins of the ATmega8U2 integrated circuit programmed as a USB-to-serial converter of the card (component that ensures the interface between the TTL levels and the USB port of the computer). We recall serial transmission through these pins with the Serial.Print() instruction, provided that the USB cable is disconnected, otherwise there will be an overlap [8-9].

- **External interrupts (**Pin 2 and 3**):**

These pins can be configured to trigger an interrupt on a low value, on a rising or falling edge, or on a change of values. See the attachInterrupt() statement for details..[9].

- **Pulse Width Modulated (PWM pulse) (Pins 3, 5, 6, 9,10 and 11) :**

Provide an 8bit PWM pulse using the analogwrite() instruction..[9].

- **Serial Peripheral Interface (SPI),Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) :**

These pins support the SPI communication available with the SPI communication library. The SPI pins are also connected on the ICSP connector.

- 12C
- Pins 4 (SDA) and 5 (SCL), support 12C protocol communications, available using the Wire /12C library
- Or TWO Wire.
- LEDs.

There is an LED included in the board connected to pin 13. When the pin is high, the LED is on, when the pin is low, the LED is off.

There are two other pins available on the board:

- **AREF**: Reference voltage for analog inputs (if different from 5V), used with the analog Reference() instruction.
- **Reset**: Setting this pin low causes the microcontroller to be reset. Typically this pin is used to add a reset button on the circuit which blocks the one present on the card and the figure II.4 shows the Various sections of Arduino.

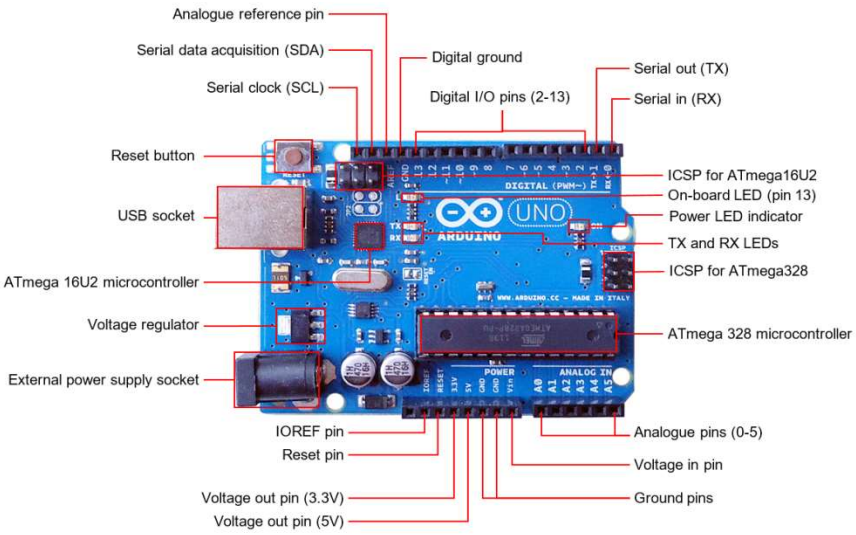

Figure II.4 Various sections of Arduino [10].

II.8. The other Hardware and Components Needed:

II.8.1DC Motor:

Small DC motors are popular due to their low voltage (1.5V to 12V), fast speed (1000 to 3000 RPM), and moderate torque, making them suitable for light loads. The figure.5 shows a DC motor shape in real life, Their tiny, lightweight form enables for simple incorporation into small places. They provide easy speed and direction control via PWM and polarity reversal, are typically efficient, and have fast reaction times. Maintenance is minimal, however brushed motors may require occasional repair. While they do cause some noise and vibration, they are usually minimal.

- Operating Voltage: 1.5V to 12V.
- Speed : 1000 to 3000 RPM.
- Torque : acceptable.
- Size: compact and lightweight.
- Control: Simple speed and direction control.
- Efficiency: Generally efficient, having rapid reaction times.
- Maintenance: Noise and vibration levels are low.

Figure II.5. DC motor [11].

II.8.2. Motor Driver (H-bridge):

The H-bridge is an electrical structure used to manage the polarity of a dipole's terminals. It is made up of four communication parts that are schematically organized in a H shape; The figure II.6 shows a H bridge shape in real life. This structure is used in many power electronics applications, including motor control, converters, and choppers. The H-bridge allows you to route the current based on the condition of the four switches (which may be changed by transistors) [12].

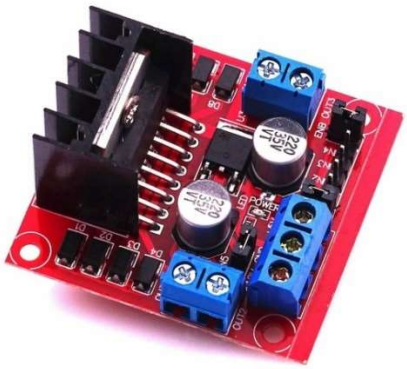

Figure II.6. H bridge [13].

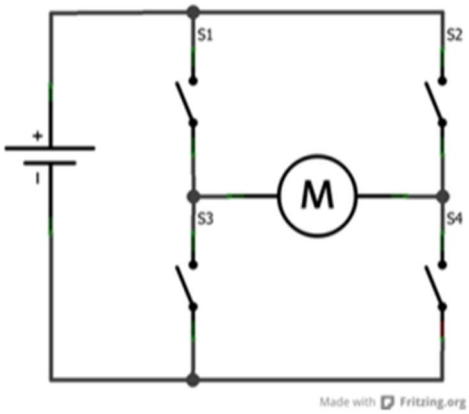

Figure II.7. Diagram of the H-bridge [12].

II.8.3. IR Sensors:

IR sensors, or infrared sensors, are devices that detect infrared radiation, commonly used for object detection, distance measurement, and motion sensing. The Figure II.8showsaIR sensor shape in In reality. They consist of an IR emitter and an IR detector. The emitter sends out infrared light, which reflects off objects and is detected by the receiver, enabling the sensor to determine the presence and distance of objects based on the reflected light. There are many sensitivity factors such:

- **Ambient Light**: Can cause interference; filters and modulation techniques are used to improve accuracy.
- **Reflective Surfaces**: Highly reflective or very dark surfaces can impact performance.
- **Angle of Incidence**: The detection angle affects sensitivity and accuracy.

Figure II.8. IR Sensor [14].

II.8.4. Segment display:

As the name implies, the 7-segment display contains seven segments; it is a component that displays numbers ranging from 0 to 9. This form of display is commonly seen on calculators and watches with digital displays: characters (numbers, however certain letters are utilized for hexadecimal display) are created by turning on or off seven segments. When all seven parts are lighted, we get the number 8 and The Figure II.9shows a segment display shape in reality [15].

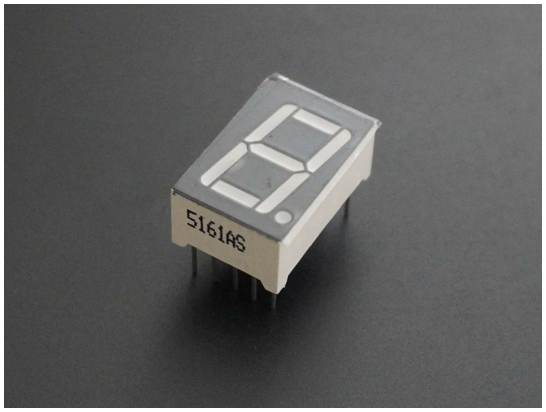

Figure II.9. Segment display [15].

One segment is actually a flat LED, and the 7 are arranged in the shape of an eight. To arrange the numbers you have to turn on certain segments, and leave others off. It is segment to be easily identified and associated with letter.

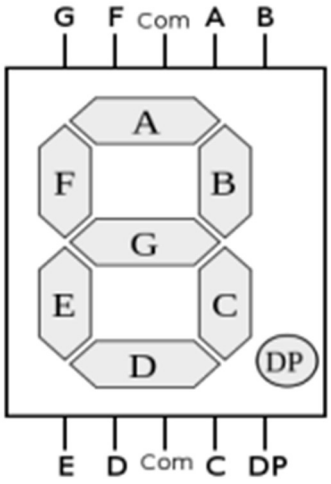

Figure II.10. Segment identified with letters [15].

II.5.5.Pushbutton:

The push button or push switch that shown in the Figure II.11is the basis of interactivity between man and machine. It allows you to control an object, a machine or simply a light.

Figure II.11 Pushbutton [16].

II.5.6. Power source (battery):

A 9-volt battery(Figure II.13), works as an ideal solution for this project as it is able to supply the circuit with sufficient energy to operate it and manage the engine, in addition to its low price.

Figure II.14. 9. Volt battery [17].

II.8.5. Connecting wires:

Jumper wires are commonly used to connect components. They come in male-to-male, male-to-female, and female-to-female types, and various lengths and colors for easy identification (Figure II.15). Made from flexible, durable materials, they ensure reliable connections and fit snugly into Arduino headers and breadboards, making prototyping straightforward and secure.

Figure II.16. Connecting wires [18].

II.9. Conclusion

In this chapter we introduced the features of the Arduino UNO card. To do this, we describe the hardware part. On the material side we are highlighted the analog and digital inputs/outputs of the card and its internal components. The elements used in the project were also presented.

Chapter III: Project execution

III.1. Introduction:

This last chapter presents the creation of the elevator controller, the different program steps for this model; as well as the different organization charts of the different programs.

A user present in front of the elevator calls the latter if the cabin is not at their level,The Arduino board receives these call commands. The processor processes all its variables, and consequently generates the control signals for the DC motor.

The processor determines the current position of the cabin using the signals generated by the presence sensors (IR sensors) and displays them using a 7-segment display. Whether or not the single-level car stops depends on the priority calculation developed by the algorithm.

III.2.Controller card design

This part is based on the design of 3 (three) circuits; a power circuit for controlling the MCC motor, a display circuit with the 7 segment display and a circuit for the IR sensors and push buttons, and for the power supply used a 9V battery.

III.3.Circuit :

III.3.1 Circuit diagram :

The setup includes an Arduino board to control the elevator's operations, a seven-segment display to show the current floor, and three push-buttons for selecting floors. The elevator movement is powered by a motor, which is controlled by a motor controller. Three IR sensors detect the elevator's position on each floor. The components are connected using a breadboard and wires, forming a functional and interactive model of an elevator system. This project demonstrates basic principles of automation and control systems and the installation of the circuit is as shown in the following figure (Figure III.1).

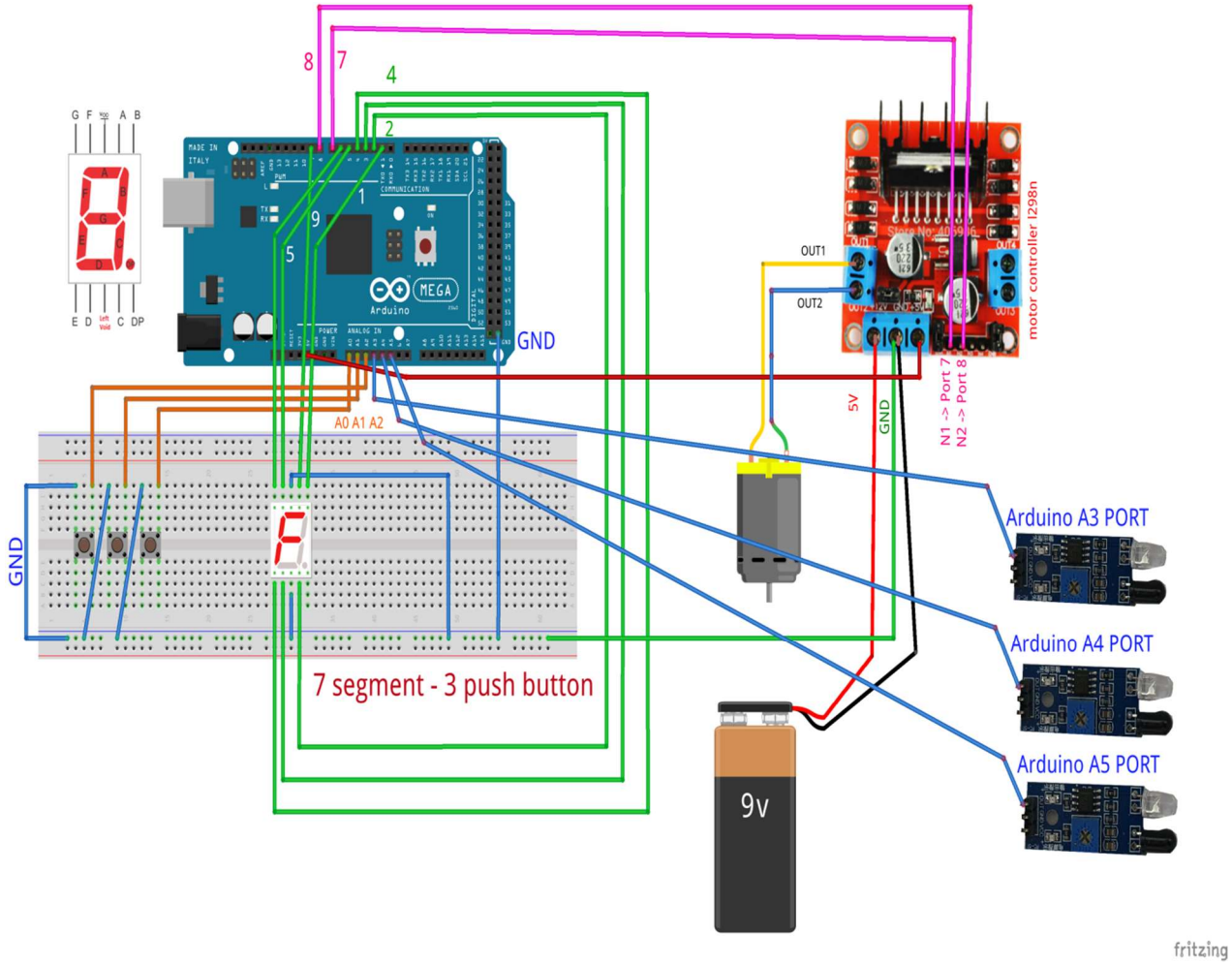

Figure III.1. Circuit diagram.

III.3.2. Circuit connection explanation:

Pin Declarations:

1. **7-Segment Display Pins:** These pins control the segments of the 7-segment display to show numbers:
    - pin_A = 9;
    - pin_b = 1;
    - pin_c = 2;
    - pin_d = 3;
    - pin_e = 4;
    - pin_f = 5;
    - pin_g = 6;

The 7-segment display is used to show the current floor of the elevator.

Motor Control Pins:

These pins control the motor's direction and speed:

- OUT_Motor_1 = 7;
- OUT_Motor_2 = 8;

OUT_Motor_1controls the motor to move the elevator up.OUT_Motor_2controls the motor to move the elevator down

Floor Buttons:

These pins are connected to buttons for requesting floors:

- floor_0 = A0
- floor_1 = A1
- floor_2 = A2

Each pin corresponds to a button press for a specific floor (0, 1, 2)

Floor Sensors:

These pins are connected to sensors that detect when the elevator reaches a floor:

- sensor_0 = A3
- sensor_1 = A4
- sensor_2 = A5

Each pin corresponds to a sensor that detects if the elevator has reached the specific floor.

III.4. programming software (Arduino IDE):

III.4.1. Definition of the software:

The Arduino Integrated Development Environment (IDE) is a Java-based cross-platform application (for Windows, MacOS, and Linux). It allows you to develop and download programs to Arduino-compatible boards. The software and Structure of a program are shown in the figure III.2, figure III.3 and figure III.4.

The IDE source code is released under the GNU General Public License, version 2. The Arduino IDE supports C and C++ languages using special code structuring rules. The Arduino IDE provides a Wiring Project software library, which provides many common input and output procedures. The user-written code requires only two basic functions, starting the sketch

and the main program loop, which are compiled and linked to a main() program stub into a cyclic executable program with the string useful GNU, also included in the IDE distribution

The Arduino IDE uses the Avrdude program to convert executable code into a hexadecimal-encoded text file loaded into the Arduino board by a loading program in the board's firmware.

III.4.2. Software Presentation :

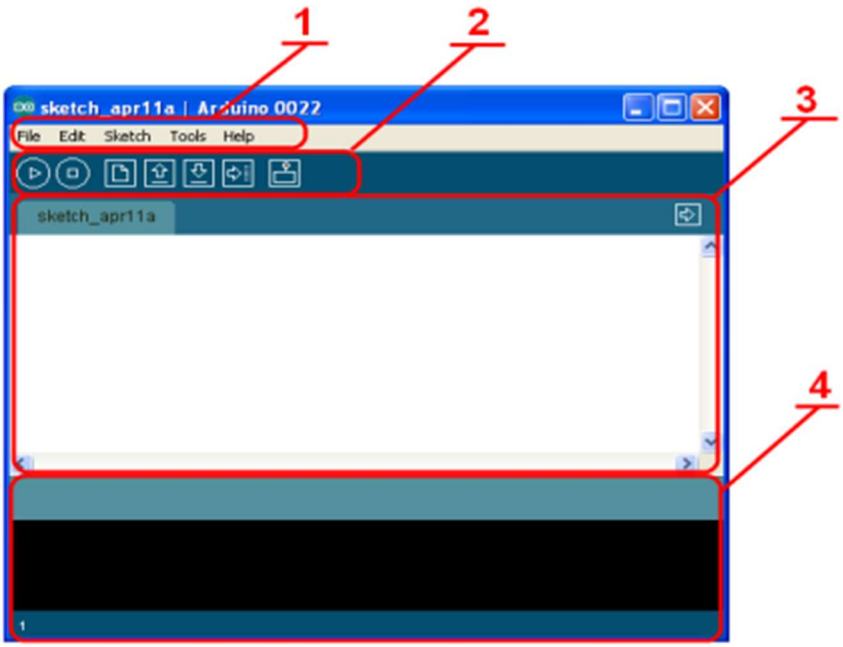

Figure III.2. The software interface.

1. software configuration options
2. buttons for card programming the chip
3. program to created
4. debugger (display of programming errors)

III.4.3. Buttons :

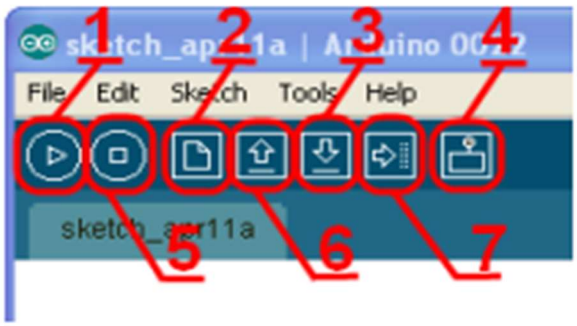

Figure III.3. The toolbar.

- **Button 1**: allows you to check the program, it activates a module which looks for errors in the program.
- **Button 2**: Create a new file.
- **Button 3**: Save the current program.
- **Button 4** : Serial link.
- **Button 5** : Stops checking.
- **Button 6**: Load an existing program.
- **Button 7**: Compile and send the program to the card.

III.4.4. Structure of a program:

An Arduino program is a series of elementary and sequential instructions, in all Arduino programs there are three steps. To highlight these phases we will present the structure of a simple programming example which consists of turning on an LED for 1 second then turning it off for 3 seconds on pin No. 13 and so on until a new program is introduced. or stop the power supply.

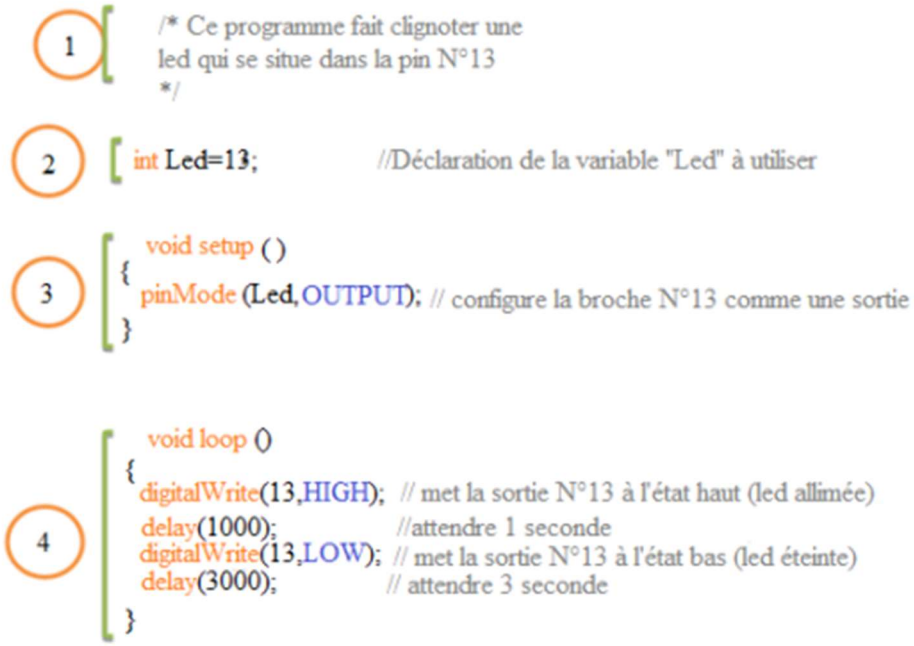

Figure III.4. Example of an Arduino program.

1. **Comments:**

To write comments on the program (this helps with the rereading of the program and its understanding by a person other than the one who wrote it) we have 2 possibilities: Either in multiple lines, by putting them between the signs /** * */. Either on a line of code by separating them from the code with the // signs [19].

2. **Definition and declaration of variables:**

For our example, we will use a digital output from the card which is for example the 13th digital output. This variable must be defined and named in this part; w

e give it an arbitrary name Led [8, 19, 20].

3. **Input-output configuration void setup:**

The digital pins of the Arduino can be configured as digital inputs or digital outputs. Here we will configure LED pin output PinMode (name, state) is one of the functions relating to digital inputs-outputs [8, 19].

4. **Programming voidloop interactions:**

In this loop, we define the operations to be performed, in order:

- digitalWrite (name, state): One of the functions relating to digital input-output.

- delay (time in milliseconds): The wait command between two other instructions.
- each line of instruction is ended with a semicolon.
- the braces surround the loop [8, 19, 20].

III.5. Programming:

First, we feed the system. A call from a particular floor is represented as follows:

- If the call comes from the floor where the elevator is located, 7-segment display indicates the presence of the car on the requested floor
- If it comes from another floor, then the elevator goes up or down to the requested floor, the display indicates the floor level at the same time.

III.5.1. Program implementation

To simplify the program, we divided it into a group of sub-stages that make up the main function.

III.5.1.1. Program flow chart:

The following chart (chart 1)shows the basic algorithm for running the program, where both numbers 1 and 2 are two secondary algorithms that form the primary algorithm.

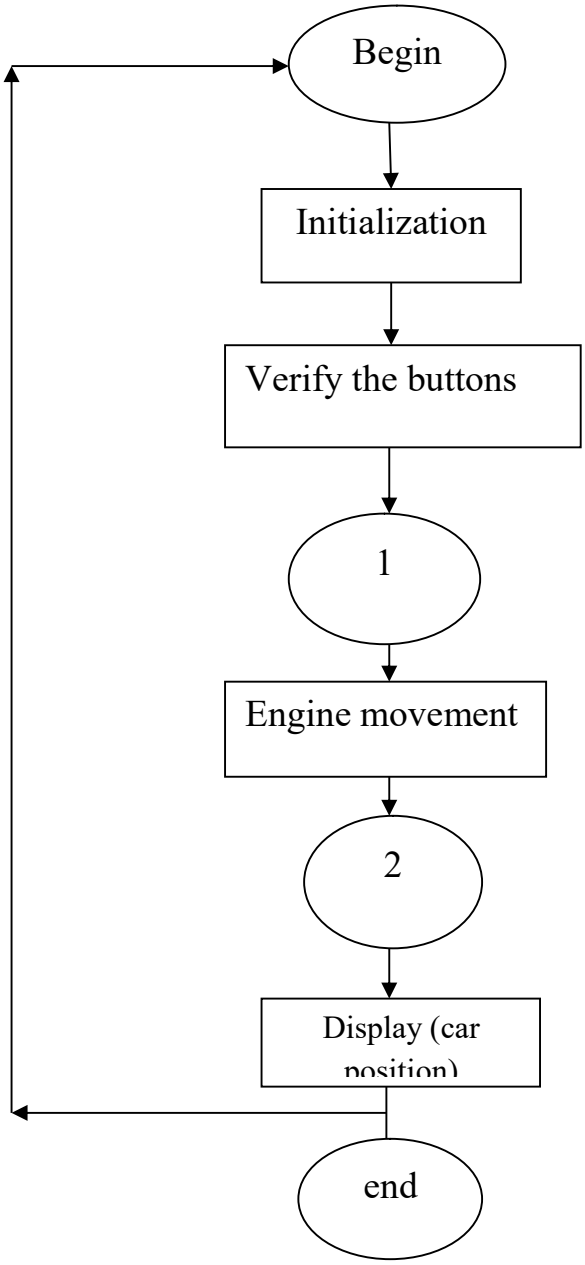

Chart 1: Main program chart

The following Chart (Chart 2) shows the basic algorithm for thebuttons, where the two numbers 1 represent the basic algorithm

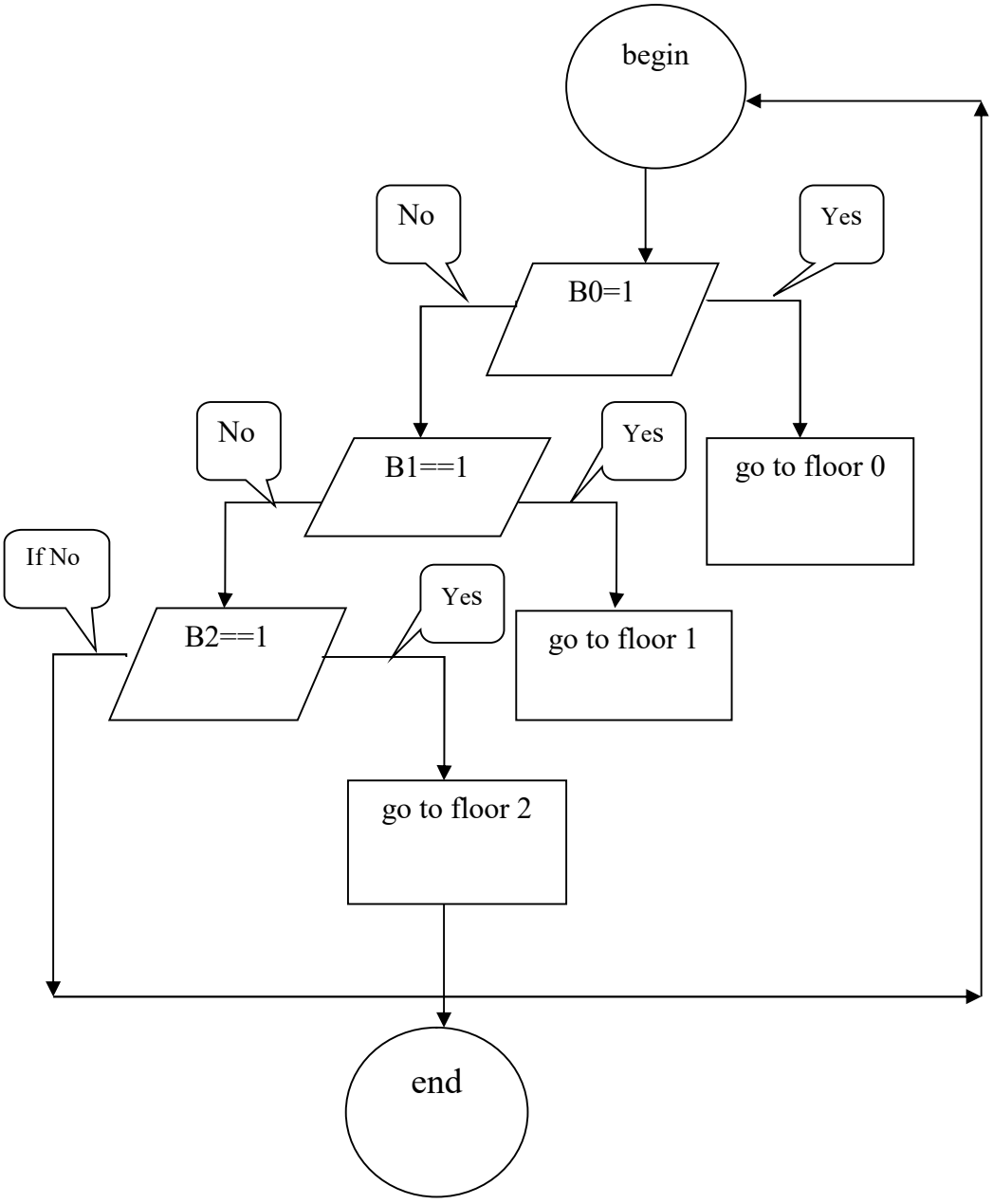

Chart 2: Main program (buttons verification (1).

The following Chart (Chart 3) shows the basic algorithm for the engine's operation, as it represents how its movement is determined.

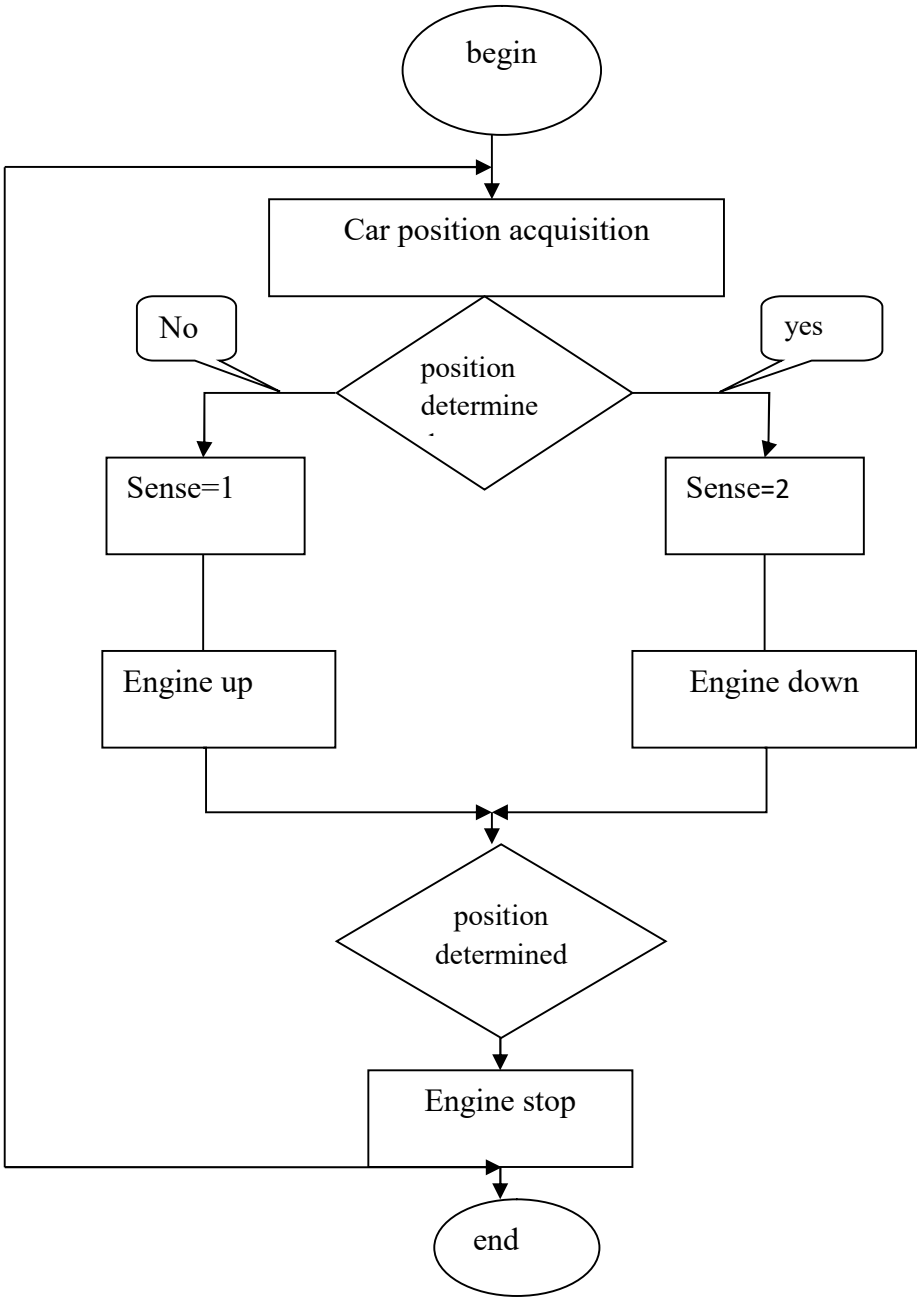

Chart 3: Engine displacement (2)

The following chart (Chart 4) shows the basic algorithm for making the sensors, where digital represents 2 in the basic algorithm.

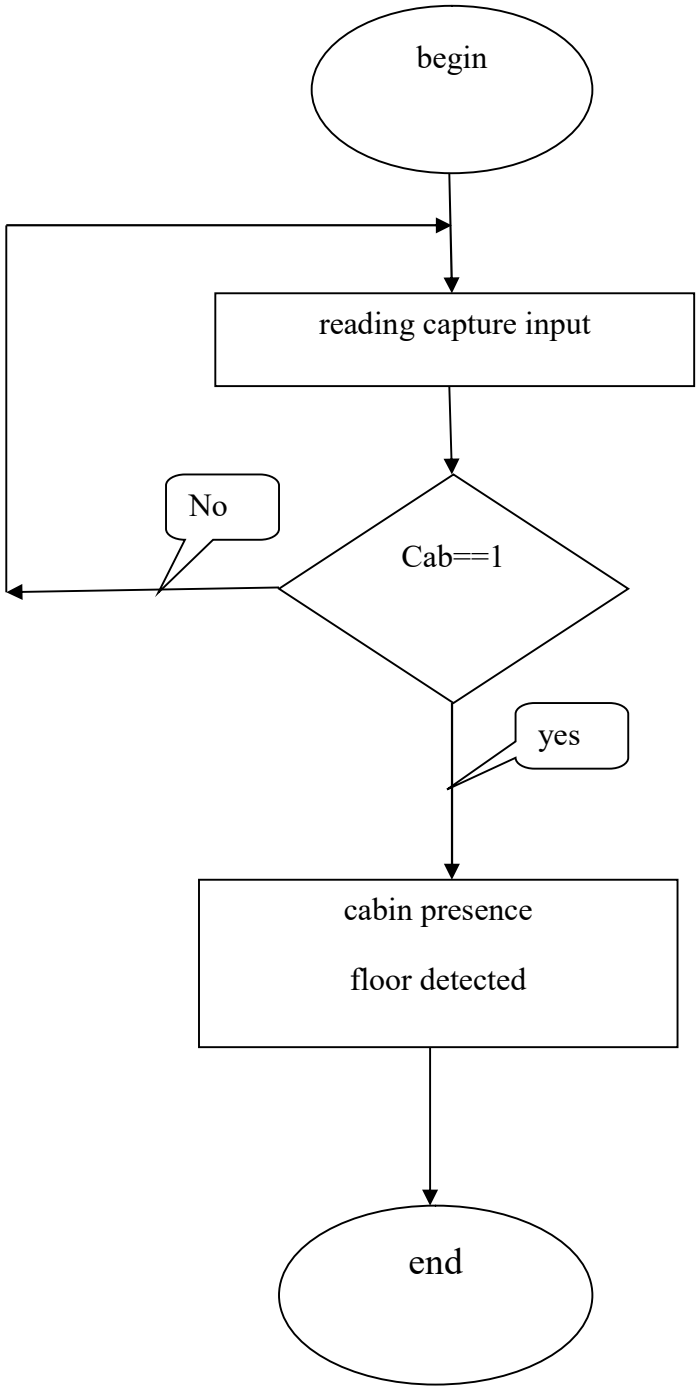

Chart 4: position detection.

III.5.2. Code Discussion and explanation:

After setting up the circuit showed in the diagram, we will go over the Arduino code that controls it, as the given code controls a three-story elevator using an Arduino. It entails interpreting inputs from floor buttons and sensors and directing a motor to direct the elevator

to the specified floor. It also refreshes the seven-segment display to represent the current floor as it appeared in the charts 1,2 and 3.

III.5.2.1. Setup Function:

The `setup` function initializes the pins and sets up the initial conditions.

```
void setup() {
  // put your setup code here, to run once:
pinMode(sensor_0,INPUT_PULLUP);
pinMode(sensor_1,INPUT_PULLUP);
pinMode(sensor_2,INPUT_PULLUP);
pinMode(floor_0,INPUT_PULLUP);
pinMode(floor_1,INPUT_PULLUP);
pinMode(floor_2,INPUT_PULLUP);
pinMode(OUT_Motor_1,OUTPUT);
pinMode(OUT_Motor_2,OUTPUT);

analogWrite(OUT_Motor_2,HIGH);
delay(500);
}
```

Initialization:

Configures sensor and floor button pins as INPUT_PULLUP. This uses internal pull-up resistors to ensure the pins read HIGH when not pressed.

Configures motor control pins as OUTPUT to control the motor direction and speed.

Starts the motor with a brief initialization by setting OUT_Motor_2 to HIGH and delaying for 500 milliseconds.

III.5.2.2.Main Loop:

The main loop checks the state of the floor buttons and moves the elevator accordingly.

```
void loop() {
  // put your main code here, to run repeatedly:
if ( digitalRead(floor_0) ==0 ){
  if ( digitalRead(sensor_0) ==1){
    analogWrite(OUT_Motor_1 ,speed_);
    while(digitalRead(sensor_0)){}
    digitalWrite(OUT_Motor_1 ,0);
  }
}
  ////////////////////
if ( digitalRead(floor_2) ==0 ){
  if ( digitalRead(sensor_2) ==1){
    analogWrite(OUT_Motor_2 ,speed_);
    while(digitalRead(sensor_2)){}
    delay(400);
    digitalWrite(OUT_Motor_2 ,0);
  }
}
  ////////////////////
 if ( digitalRead(floor_1)==0 ){
   if ( digitalRead(sensor_1) ==1){
    ////////////
     if ( digitalRead(sensor_2) ==0){
     analogWrite(OUT_Motor_1 ,speed_);
     while(digitalRead(sensor_1)){}
     digitalWrite(OUT_Motor_1 ,0);
     }
```

```
      .
      //////////
      else if (digitalRead(sensor_0)==0 ){
      analogWrite(OUT_Motor_2 ,speed_);
       while(digitalRead(sensor_1)){}
       delay(400);
      digitalWrite(OUT_Motor_2 ,0);
      }
      else {
       digitalWrite(OUT_Motor_1 ,0);
       digitalWrite(OUT_Motor_2 ,0);
      }
      }
    }

if ( digitalRead(sensor_0) ==0){
  digitalWrite(pin_A ,1);
  digitalWrite(pin_b ,1);
  digitalWrite(pin_c ,1);
  digitalWrite(pin_d ,1);
  digitalWrite(pin_e ,1);
  digitalWrite(pin_f ,1);
  digitalWrite(pin_g ,0);

}
   if ( digitalRead(sensor_1) ==0){
  digitalWrite(pin_A ,0);
  digitalWrite(pin_b ,1);
  digitalWrite(pin_c ,1);

   }
      if ( digitalRead(sensor_1) ==0){
    digitalWrite(pin_A ,0);
    digitalWrite(pin_b ,1);
    digitalWrite(pin_c ,1);
    digitalWrite(pin_d ,0);
    digitalWrite(pin_e ,0);
    digitalWrite(pin_f ,0);
    digitalWrite(pin_g ,0);

   }
   if ( digitalRead(sensor_2) ==0){
    digitalWrite(pin_A ,1);
    digitalWrite(pin_b ,1);
    digitalWrite(pin_c ,0);
    digitalWrite(pin_d ,1);
    digitalWrite(pin_e ,1);
    digitalWrite(pin_f ,0);
    digitalWrite(pin_g ,1);

   }
    }
```

Floor 0 Request:

- Checks if the button for floor 0 is pressed (`digitalRead (floor_0) == 0`).
- If the elevator is not at floor 0 (sensor_0 is HIGH), it moves to floor 0 by activating `OUT_Motor_1`.
- The motor stops when the sensor_0 reads LOW, indicating the elevator has reached floor 0.

Floor 2 Request:

- Checks if the button for floor 2 is pressed (`digitalRead(floor_2) == 0`).
- If the elevator is not at floor 2 (sensor_2 is HIGH), it moves to floor 2 by activating `OUT_Motor_2`.
- The motor stops when the sensor_2 reads LOW, indicating the elevator has reached floor 2.

Floor 1 Request:

- Checks if the button for floor 1 is pressed (`digitalRead(floor_1) == 0`).
- If the elevator is not at floor 1 (sensor_1 is HIGH), it decides which direction to move based on the current position.
- If the elevator is above floor 1 (at floor 2), it moves down by activating `OUT_Motor_1`.
- If the elevator is below floor 1 (at floor 0), it moves up by activating `OUT_Motor_2`.
- The motor stops when the sensor_1 reads LOW, indicating the elevator has reached floor 1.

7-Segment Display Update:

Updates the 7-segment display based on the current floor by turning on/off the appropriate segments.

III.5.2.testing

After installing the circuit and sending the code to the Arduino, we test the elevator and verify the efficiency of the elevator's work by:

- Press each floor button and observe the elevator motor moving to the correct floor
- Ensure the sensors correctly detect the elevator's position and stop the motor at each floor.

- Check that the 7-segment display correctly shows the current floor number.

III.6. Conclusion

This chapter was the core of our final study project. Finally, we explained Running the prototype, installing the basic circuit, clarifying the programming algorithm, and explaining the code.

General Conclusion

General Conclusion

The initial objective of this work was to create a control card for an educational elevator based on an Arduino uno card and to adopt a methodology allowing the study of the different stages of operation of these devices.

This project allowed us to develop our knowledge and apply our know-how to realize an end-of-study project. The bibliographic research was rich in discoveries about elevator studies and the elevator industry.

Unfortunately, with the complexity of our programs (in size and number of functions), it has become very difficult to add other functions to the program, such as the interrupt system, the development of call priority management , or even door opening control. In fact, we will consider expanding this project in terms of application while adding some options to the model. The following suggested improvements are:

- Include safety sensors in case the typical speed is crossed.
- Automate door opening and closing.
- Prevent cabin movement while doors are open.
- Create floor selecting buttons (send buttons)

References

[1] GRAY, Lee Edward. *From Ascending rooms to express elevators: A history of the passenger elevator in the 19th century*. Elevator World Inc, 2002.

[2] Sandor Markon, Hajime Kita, Hiroshi Kise and Thomas Bartz-Beielstein “Control of Traffic Systems in Buildings” Springer-Verlag London Limited 2006, 2007.

[3] https://www.pioneerelevatorpk.com/hydraulic-elevator/as at/03/13 2024.

[4] M.Y.H. Bangash, T. Bangash “Lifts, Elevators, Escalators and Moving Walkways, Travelators” a.a. Balkema Publishers Leiden, London, New york, Philadelphia Singapore, Taylor & Francis e-Library.

[5] https://www.pinterest.com/pin/89509111339638637/as at2024/03/12.

[6] https://m.media-amazon.com/images/I/51mxIljPNML._AC_UF1000,1000_QL80_.jpgas at2024/03/18.

[7]https://www.researchgate.net/figure/Datasheet-ATMega328-Les-principales-caracteristiques-dATMega328-sont-FLASH_fig13_339999966as at14/4/2024.

[8] SIMON Landranlt, HIPPOLYTE Weisslinger, «Premier pas en informatique embarquée»,Edition CC BY-NC-SA, 01 Juin 2014.

[9] BANZI, Massimo; SHILOH, Michael. *Getting started with Arduino*. Maker Media, Inc., 2022.

[10] https://bdavison.napier.ac.uk/iot/Notes/microprocessors/img/ArduinoUnoR3.pngas at16/4/2024

[11] https://www.betterequipped.co.uk/pub/media/catalog/product/cache/a7b20ef70c34db9edb64d1963c018af1/4/1/4179_1.jpgas at20/5/2024.

[12] https://passionelectronique.fr/tutoriel-l298n/.

[13] https://probots.co.in/l298n-2a-dual-motor-driver-module-with-pwm-control.htmlas at18/5/2024.

[14] https://store.nerokas.co.ke/sku-1803.

[15] https://www.electronicwings.com/sensors-modules/7-segment-led-displayas at18/4/2024

[16] https://stemextreme.com/2022/09/21/parts-of-a-circuit-push-button/

[17] https://calcuttaelectronics.com/wp-content/uploads/2019/11/9V-battery.jpgas at11/5/2024.

[18] https://4.imimg.com/data4/SX/NI/MY-4535820/connecting-wire-1000x1000.jpgas at19/4/2024.

[19] Christian Tavenir, «Arduino, maîtrisez sa programmation et ces cartes d'interface (Schields)»,CollectionDunod, Mars 2014

[20] EVANS, Brian. *Beginning arduino programming*. A press, 2011.