Voice Operated Lift Control System with Safety

by AVCOE Central Library

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The first known elevator was invented by Archimedes in 236 BC. It was a simple hoist powered by human or animal labor. In 1852, Elisha Otis demonstrated the first safety-based elevator. This elevator had a brake that would engage if the cable broke, preventing the elevator from falling. In 1857, Otis installed the first passenger elevator in a New York City store. This elevator was powered by steam. The first residential elevator was created by Clearance Conrad in 1929. This elevator was powered by electricity. In the 1950s, Otis introduced the Autotropic system, which was the first predictive elevator control system. This system could anticipate traffic patterns within a building and deploy elevators in the most efficient manner [1].

Blind people face many problems every day. One of these challenges is the use of elevators in many buildings. visually impaired should be able to enjoy using the elevator easily. Chapter To overcome this challenge for the blind, we must focus on the following issues: Make sure the blind person is at the elevator door Chapter Accept the idea of getting down for the blind person Chapter Attention coming into the elevator from the seat of the blind person [2].

Speech is the superior personality of the human beings gifted by the nature. Speech helps to deliver the thoughts and messages between human. Speech recognition is the process of the computer recognizing human speech to generate a string of words or commands. Sometimes it is known as automatic speech recognition. Speech recognition is becoming more perplexing and difficult task. The speech recognition research is focuses mainly on large vocabulary, continuous speech capabilities and speaker independence. The design of speech recognition requires cautious attention to some issues like speech representation, depiction of various types of speech Classes, techniques, and database and performance evaluation [3]

A voice-operated elevator system is proposed where the user's input commands to control the movement of the elevator system are kept convenient for the users. The commands include voice input for the floor operations, directions, elevator car's door operation, and a special option to place a call of speaker's choice in case of any unexpected event that requires immediate action [4]

1.2 NEED OF PROJECT

The visually challenged people cannot use the elevator easily. It is difficult to use the keypad if they cannot see it. Sometimes the keypad has Braille technique, but they will have hard time for locating its place. Even though they found the keypad, how can they know the number if they do not know Braille?

They always need help in elevators from someone to press the button for them and to tell them when the elevator cabin arrives. In case of emergency how they will act if there is no one with them to help. So voice-controlled elevator can be a very good option for this people. One more drawback of the current lift is that it cannot tell on which floor the lift is, nor the parameters like temperature of motor, fire detection inside the lift. But by using this voice operated lift we can solve all these problems.

Also traditional lifts do not give an auditory feedback to its users, in case of physically and visually challenged people it will be very useful to give the auditory feedback. In traditional lifts the user standing in lobby has to press a button in order to call lift on the corresponding floor, it is challenging for visually and physically disabled people. So we can use IR sensors to detect users presence on particular floor and user will not need to press button to call lift.

1.3 AIM OF THE PROJECT

To design and develop a Voice Operated Lift Control System for efficient elevator control using voice commands, enhance safety through integrated sensors, provide auditory feedback to the user.

1.4 OBJECTIVES OF THE PROJECT

- To develop a system that allows elevator operation through voice commands.
- To implement load, flame, and temperature sensors to monitor elevator safety parameters.
- To create error handling mechanisms that display error messages and stop the elevator motor in case of emergency.
- To incorporate an audio player module to audibly announce the arrived floor to passengers.

1.5 LIST OF PUBLICATIONS

SR.NO	NAME OF PAPER	PUBLICATION	AUTHOR NAME	PUBLICATION YEAR
1	Voice Operated Lift Control System With Safety	International Research Journal of Modernization in Engineering Technology and Science	Prof. V. R. Aware Bhoknal Gayatri Varpe Aditya Waman Onkar Zanjare Shaileshkumar	April-2024

Table 1 : List of Publication

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 PROJECT BACKGROUND

The project Voice Operated Lift Control System with Safety aims to enhance the functionality and safety of traditional elevator systems through the integration of voice recognition technology and advanced safety mechanisms. Elevator systems are an integral part of modern infrastructure, facilitating efficient vertical transportation in buildings of all sizes. However, conventional elevator control interfaces, typically consisting of buttons and switches, may present challenges for individuals with disabilities or impairments, as well as for users seeking a more intuitive and convenient interaction method.

To address these limitations, the project proposes the implementation of a voice-operated control system for elevators. By leveraging voice recognition technology, users will be able to command the elevator simply by speaking their desired floor number. This hands-free interface not only enhances accessibility but also offers a more user-friendly and convenient experience for all passengers.

Moreover, the project emphasizes the integration of safety features to ensure secure operation of the elevator system. Safety mechanisms such as lift weight, detection of fire, motor temperature systems will be incorporated to mitigate potential hazards and minimize the risk of accidents. These safety features are essential to instil confidence among users and regulatory authorities regarding the reliability of the voice-operated lift control system.

Additionally, the project seeks to provide auditory feedback to users to enhance the overall user experience and convey important information during operation. Auditory cues, such as confirmation messages upon receiving voice commands, status updates regarding elevator position, and alerts for emergency situations, will be implemented to ensure effective communication with passengers.

By combining voice recognition technology with robust safety features and auditory feedback mechanisms, the proposed project aims to modernize elevator control systems, making them more accessible, intuitive, and secure for users of all abilities.

2.2 LITERATURE SURVEY:

Sr no	Title of the Paper	Year of Publication	Authors	Methodology
1	Voice Operated Elevator	2021	Aishwarya Pokharkar, Niriksha Poojari, Harish Pawar , Amey Patil	This device acts as a human-machine communication system Speech recognition is that the method of recognizing the spoken words to require the mandatory actions in line with the commands.
2	voice Operated Intelligent Lift With Emergency Indicator	2017	Anu K G, Anupriya K S, Lekshmi M S	This project makes use of a DC motor for moving the lift/elevator based on the voice/speech commands given by the user. Voice recognition system is used for recognition of the voice commands.
3	Voice Control Elevator for Prevention of Physical Touch	2020	Archana L. Rane, Nikhil Patil	The proposed system demonstrate the use of smart elevators using Smartphone. This system makes use of a DC motor for moving the elevator based on the voice/speech commands given by the user on their Smartphone.
4	Implementation of Voice based Touchless Lift System	2021	B. Swathi, Akshay S Prathap, Aiswarya V Kumar, Ranjitha R, Raviteja Kaki	The proposed system uses an offline voice recognition module for voice commands. It can store 80 commands of 1500 mS length.

5	FPGA Implementation of Biometric based Elevator Controller	2018	Dilip Mathuria, Aditya Gaur	This proposed system uses biometric sensor for verification and system is developed using FPGA. This allows some features as restriction to other users, etc.
6	Elevator for blind people using voice recognition	2018	Farouk Salah, Mohamed Saod, Dr. Maher M. Abdel-Aziz	Proposed system makes easy way to use the elevator especially for the blind people. Provided by the remote in which will give the blind person a fully control over the elevator.
7	Elevator Control Using Speech Recognition for People with Physical Disabilities	2021	Komal Mahajan, Riddhi Nahar, Dhanali Khairnar, Shrutika Kinge, Sujata Suryawanshi	The proposed system uses offline raspberry pi-based voice recognition by using Sphinx4 platform locally running on pi. It also includes commands for emergency.
8	Controlling of Electric Elevator by using Voice Announcement, Speed Control and Mini Lift Model System	2021	Omkar Jadhav, Shubhanshu Bishwash, Manisha Ganguly, Omkar Nayak	The usual problem with voice input-based elevator system is that it takes only one input at a time. However, this proposed elevator system takes care of that problem. With the features like PWM based speed control and voice input for multiple destination at once, this project greatly mimics the traditional elevator system.

Table 2 : Literature Survey

CHAPTER 3

SYSTEM DESIGN

3.1 BLOCK DIAGRAM

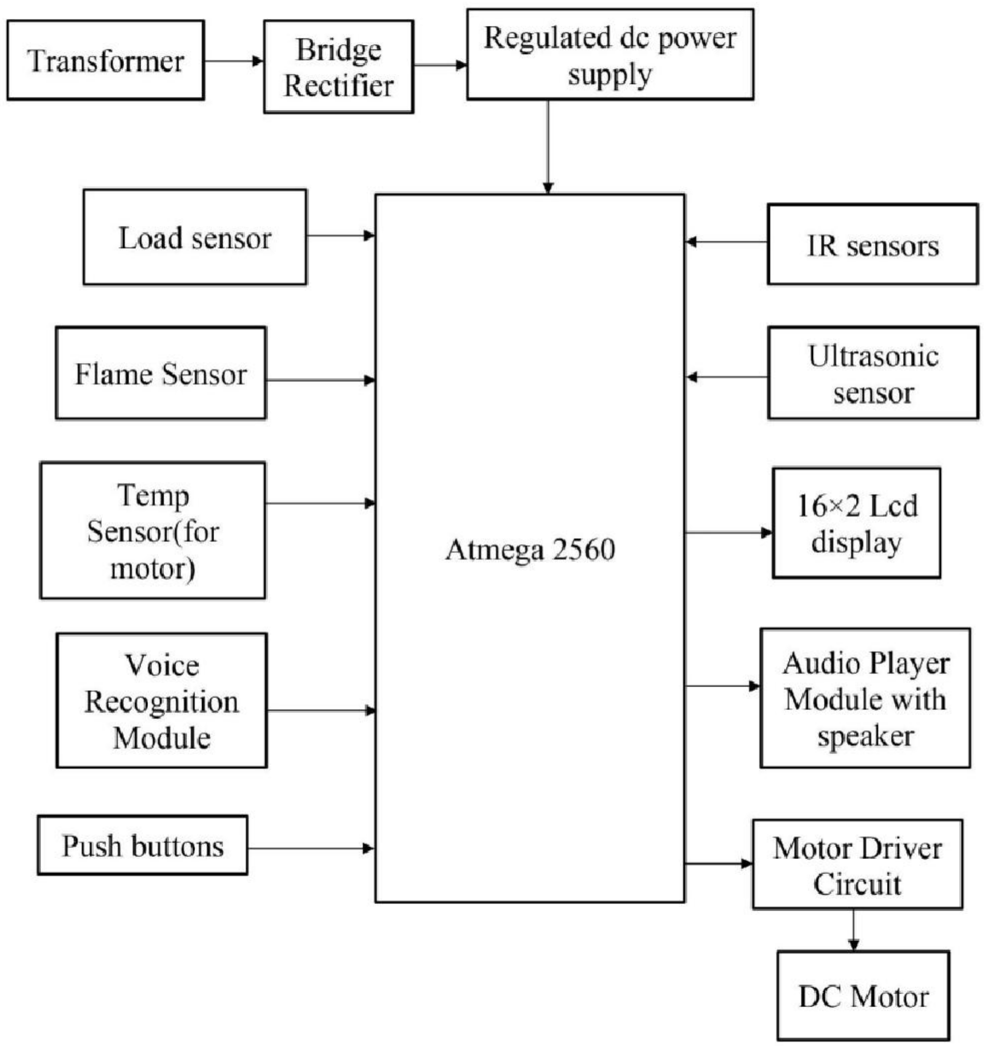

Fig.3.1: Block Diagram of Lift System

3.1.1 BLOCK DIAGRAM DESCRIPTION:

- .Here the rectifier is used as a component of power supply. The 230V AC is step down to 12V AC using transformer and is rectified to 12V DC by the rectifier. LM7805 is used as the voltage regulator, which regulates the voltage to 5V [6].
- The IR sensor is used in the lobby in front of lift doors to recognize whether anyone is near the door of lift. If yes then lift automatically goes on that floor, this feature is very useful for visually and physically disabled people who otherwise will have found it difficult to press the lift button in the lobby.
- The voice recognition system is the main part of this project. Voice recognition module is communication mechanism between the user and microcontroller. The project will make the use of DC motor for the moving of lift. Microcontroller is programmed, with the help of C++ programming.
- The microcontroller processes the received voice commands using a voice recognition module. Based on the recognized voice commands, the microcontroller's control logic will determine the appropriate actions to control the elevator, interfacing with the elevator's control circuitry and motor drivers to execute the commands accurately [7].
- Load sensors placed within the elevator continuously monitor the weight of the carriage and passengers, ensuring that the elevator does not exceed its maximum weight capacity for safe operation. Also the temperature sensor monitors temperature of motor and flame sensor finds if there is any fire in lift.
- Overall, this voice-controlled elevator system provides an innovative, user-friendly, and safe elevator experience for passengers.

3.2 COMPONENTS REQUIRED:

3.2.1 ARDUINO MEGA

The Arduino Mega, a member of the Arduino microcontroller platform, is a remarkable development board that offers an extensive range of features and capabilities. It is based on the ATmega2560 microcontroller, which provides ample processing power, a variety of input and output pins, and extensive memory for program storage. The Mega is an ideal choice for complex projects, allowing for the creation of sophisticated electronic systems and automation applications.

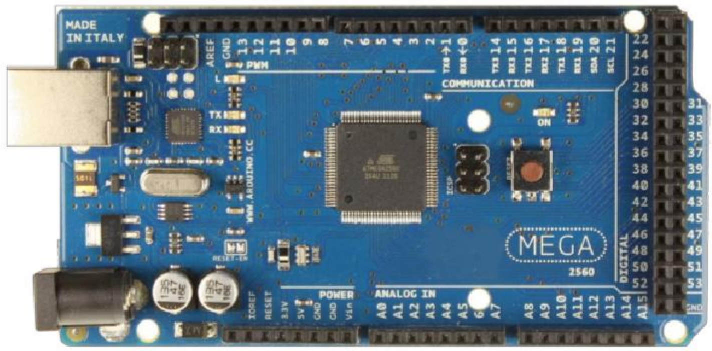

Fig.3.2: Arduino Mega

Working Principle:

The Arduino Mega operates on a microcontroller unit (MCU), Atmel ATmega2560, which integrates processing power and input/output capabilities. It communicates with external devices through digital and analog pins, executing programmed instructions stored in its flash memory. Its versatile design enables it to interface with various sensors, actuators, and communication modules, making it suitable for a wide range of projects.

Specifications:

1. Microcontroller: ATmega2560
2. Operating Voltage: 5V
3. Input Voltage (recommended): 7-12V
4. Input Voltage (limits): 6-20V
5. Digital I/O Pins: 54 (of which 15 provide PWM output)
6. Analog Input Pins: 16
7. DC Current per I/O Pin: 20 mA
8. DC Current for 3.3V Pin: 50 mA
9. Flash Memory: 256 KB of which 8 KB used by bootloader
10. SRAM: 8 KB
11. EEPROM: 4 KB
12. Clock Speed: 16 MHz

3.2.2 INFRARED SENSOR (IR-08H)

Infrared (IR) sensor, also known as IR detectors or IR receivers, are devices that are designed to detect and respond to infrared radiation. They are commonly used in a variety of applications, including remote controls, motion sensors, temperature sensors, and more. Here is some detailed information about IR sensors:

Working Principle:

IR sensors work based on the principle that certain materials or semiconductors are sensitive to infrared radiation. When infrared radiation falls on the sensor, it can either generate a voltage, change its resistance, or produce a digital signal, depending on the specific type.

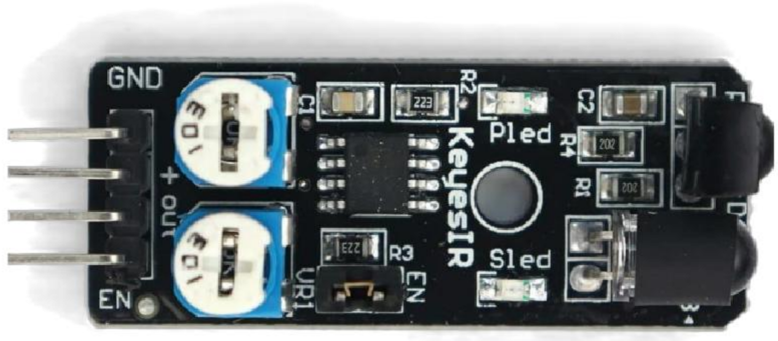

Fig.3.3: Infrared Sensor

Specifications:

1. Operating Voltage: 3.6 - 5 V DC
2. Supply current: 20 mA
3. Detection Angle: 35°
4. Distance Measuring Range: 2 – 30 cm
5. I/O pins are 5V and 3.3V compliant
6. Built-in Ambient Light Sensor
7. Adjustable Range
8. Adjustable Frequency
9. LED Indicators

3.2.3: VOICE RECOGNITION MODULE (Elechouse V3)

Converts the user's spoken commands into a command number which is already stored in module. Voice Recognition Module product is a speaker-dependent voice recognition module. Users need to train the module first before let it recognizing any voice command.

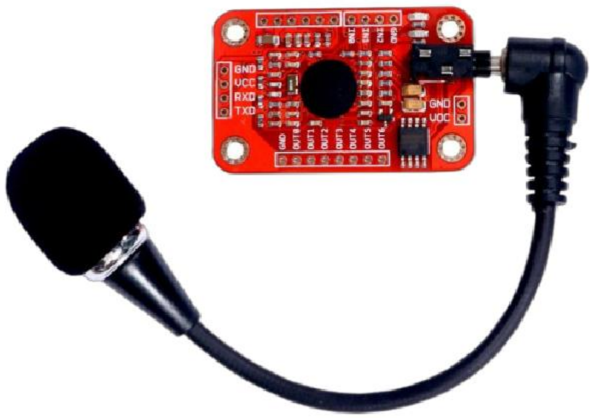

Fig 3.4: Voice Recognition Module

Working Principle:

The Elechouse Voice Recognition Module operates on the principle of analysing input sound signals through its built-in microphone. It uses a pattern recognition algorithm to match these signals against pre-recorded voice samples stored in its memory. When a match is found, it triggers a corresponding output signal, such as activating a relay or sending a command to a microcontroller. This module essentially enables voice-controlled functionality by recognizing specific spoken words or phrases within its database.

Specifications:

1. Support a maximum of 80 voice commands, with each voice 1500ms (one or two words speaking).
2. Maximum 7 voice commands effectively at the same time.
3. User-control General Pin Output.
4. Analog Interface: 3.5 mm mono-channel microphone connector + microphone pin interface.
5. Input supply voltage: 4.5 - 5.5 V DC
6. Current: 40 mA
7. Recognition accuracy: 99%
8. Protocol: UART/GPIO

3.2.4 : 16X2 LCD

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs.

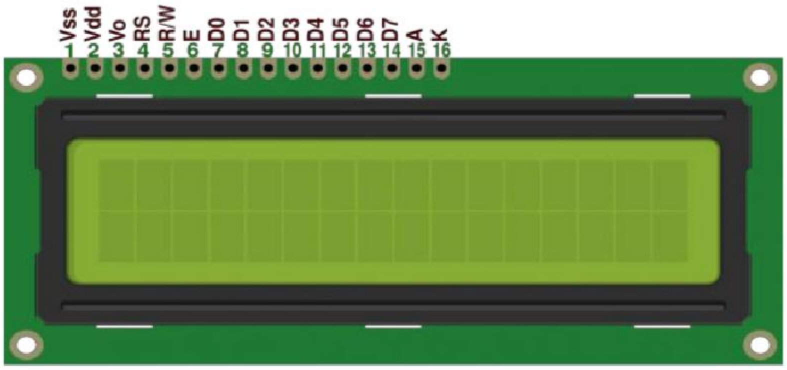

Fig.3.5: 16X2 LCD

Pin no.	Symbol	External connection	Function
1	Vss	Power supply	Signal ground for LCM
2	V_{DD}		Power supply for logic for LCM
3	V_0		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

Table 3: Pin description of 16x2 LCD

Working Principle:

The 16x2 LCD operates by applying electrical signals to liquid crystal cells, manipulating their alignment to control the passage of light. By sending specific commands and data to the LCD module, alphanumeric characters and symbols can be displayed on the screen, making it a fundamental component for visual feedback in various electronic projects.

Specifications:

1. Display capacity: 16 characters x 2 rows
2. Character Size: Typically 5x8 dots for each character.
3. Operating Temperature: Typically operates in the 0°C to 50°C range.
4. Character size: 2.95 mm wide x 4.35 mm high
5. Current requirements: 2 mA @ 5 V DC
6. Voltage requirements: 5 VDC +/- 0.5V
7. Character pixels: 5 W x 7 H
8. Total pins: 16
9. Data pins: 8 (D0 to D7)
10. Display color: Blue backlit
11. Contrast Adjustment

3.2.5. I2C serial interface adapter module

This LCD I2C adapter module is designed to fit directly below the standard 16x2 LCD display. Then through this module, the LCD can communicate through I2C protocol which requires only 2 pins from the controller side and then uses the PCF8574 IC to receive data from I2C and display them on the LCD screen.

Working Principle:

The I2C serial interface adapter module facilitates communication between microcontrollers and peripherals using the I2C (Inter-Integrated Circuit) protocol. It functions by enabling bidirectional serial data transfer between devices over two wires: SDA (data) and SCL (clock). This module acts as a bridge, managing data exchange, addressing, and timing synchronization between connected devices, enabling seamless communication in complex electronic systems.

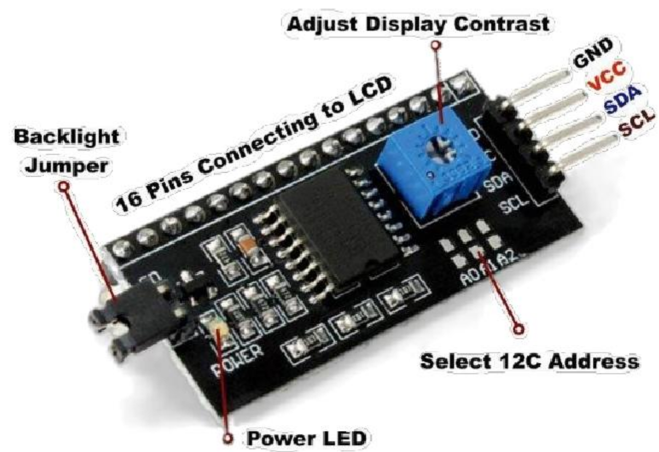

Fig.3.6: I2C module

Specification:

1. Interface: I2C (Inter-Integrated Circuit)
2. Operating Voltage: 3.3V to 5V
3. Communication Speed: Standard mode (up to 100 kHz), Fast mode (up to 400 kHz), High-speed mode (up to 3.4 MHz)
4. Supported Devices: Compatible with various I2C devices such as sensors, EEPROMs, LCD displays, etc.
5. Number of Channels: Single-channel or multi-channel (if it supports multiple I2C buses)
6. Connector Type: Often comes with pin headers or terminal blocks for easy connection
7. Voltage Level Shifting: Some modules include voltage level shifting circuitry to support different logic levels between devices
8. Pull-Up Resistors: May include built-in pull-up resistors for the SDA and SCL lines
9. Compatibility: Compatible with popular microcontroller platforms such as Arduino, Raspberry Pi, etc.
10. Dimensions: Physical dimensions of the module, which can vary depending on the design

3.2.6. Audio player module (APR33A3)

The APR33A3 Voice Recorder and Playback Module is an incredible device that allows users to particularly record and playback audio files with ease. It features a built-in microphone, support for multiple audio formats, and significantly a 3.5mm headphone jack. It is a great choice in any case for anyone looking to capture voice recordings and playback quickly and easily them back with clear audio quality.

Working Principle:

The APR33A3 audio player module operates by storing audio files in an onboard flash memory. When triggered, it retrieves and plays back audio files based on control signals received through its interface pins. The module typically includes features for volume control, track selection, and playback modes.

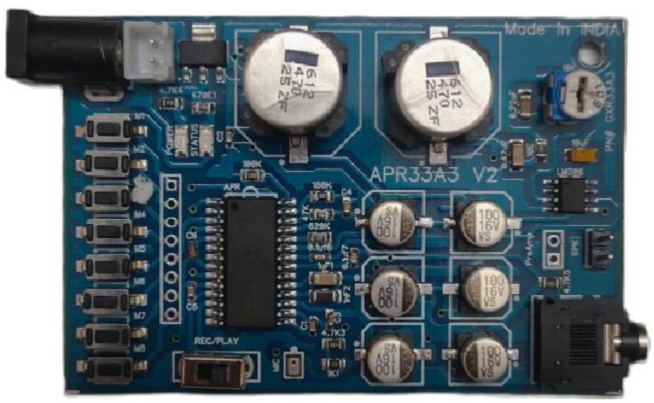

Fig.3.7: Audio player module

Specification:

1. Working voltage: 5 V DC
2. Current: 500 mA
3. Number of Channels: 8 Channel
4. Recording time on all the 8 Channels: 10 min
5. Mic: On Board
6. Speaker: 1/2 W Speaker
7. Playback and Recording Switch
8. Microcontroller TTL Interface Pins with respect to Ground (not soldered on Board).

3.2.7. Ultrasonic Sensor (HC-SR04)

This HC-SR04-Ultrasonic Range Finder is a very popular sensor that is found in many applications where it requires measuring distance and detecting objects. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. It offers excellent range accuracy and stable readings in an easy-to-use package. Its operation is not affected by sunlight or black material.

Working Principle:

The HC-SR04 ultrasonic sensor emits high-frequency sound waves. It measures the time it takes for the sound waves to bounce back after hitting an object. By calculating the distance based on the time taken for the echo to return, it determines the proximity of the object in front of it.

Fig.3.8: Ultrasonic sensor

Specification:

1. Power Supply: 5 V DC
2. Working Current: 15 mA
3. Working Frequency: 40 kHz
4. Ranging Distance : 2 cm – 400 cm
5. Resolution : 0.3 cm
6. Measuring Angle: 15°
7. Trigger Input Pulse width: 10 μS

3.2.8. Temperature sensor (ds18b20)

The DS18B20 is a programmable digital temperature sensor that communicates using the 1-Wire method. It does not require any external components to operate. We have used it to measure the temperature of DC motor to avoid overheating of motor.

Working Principle:

The DS18B20 temperature sensor operates on the principle of measuring the voltage drop across a semiconductor junction with temperature. It utilizes the change in resistance of its internal sensor element with temperature, which is then converted into a digital signal by an analog-to-digital converter (ADC) within the sensor. This digital signal is then transmitted serially using the one-wire protocol, allowing for easy integration with microcontrollers.

Fig.3.9: Temperature sensor

Specification:

1. Temperature range: -55°C to +125°C
2. Accuracy: ±0.5°C
3. Output resolution: 9-bit to 12-bit (programmable)
4. Operating voltage: 3V to 5V
5. Current Consumption: 1 mA
6. Conversion time: 750ms at 12-bit
7. Standby current: Typically 3μA at +125°C

3.2.9. Flame sensor (YL-38)

This tiny Flame sensor infrared receiver module can use to detect flame or wavelength of the light source within 760nm~1100nm also useful for Lighter flame detect at the distance 80cm. We are going to use it to detect fire inside the lift.

Working Principle:

The flame sensor YL-38 detects infrared light emitted by flames. When a flame is present, the sensor generates an electrical signal. This signal is processed by a connected microcontroller or circuit to trigger an action, such as activating an alarm or shutting off a gas valve. It is commonly used in fire detection systems to sense the presence of flames and initiate appropriate responses.

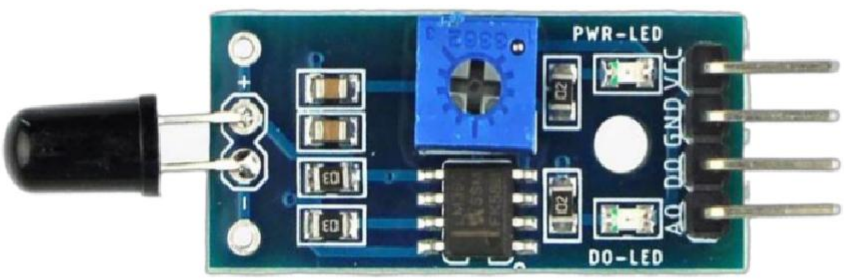

Fig.3.10: Flame sensor

Specification:

1. Operating voltage 3.3V - 5V
2. Wavelength: 760 nm - 1100 nm
3. Detection range: up to 100 cm
4. Detection angle: 60°
5. Comparator chip LM393 makes module readings stable.
6. Fast Response Time
7. Sensitivity adjustable
8. Adjustable detection range.

3.2.10. Load cell with hx711 sensor

This weighing Load Cell Sensor (sometimes called a strain gauge) can translate up to 1 kg of pressure (force) into an electrical signal. Each load cell is able to measure the electrical resistance that changes in response to, and proportional of, the strain (e.g. pressure or force) applied to the bar.

The HX711 Dual-Channel 24 Bit Precision A/D weight Pressure Sensor is a Load Cell Amplifier breakout board, for the HX711 IC that allows us to easily read load cells to measure weight. We will be able to read the changes in the resistance of the load cell, and with some calibration. We are using it to measure whether lift has overweight or not.

Working Principle:

A load cell measures force or weight by converting mechanical deformation into electrical signals. The HX711 sensor amplifies and digitizes these signals, providing precise measurements through a digital interface like I2C or SPI, making it suitable for applications such as scales and force sensors.

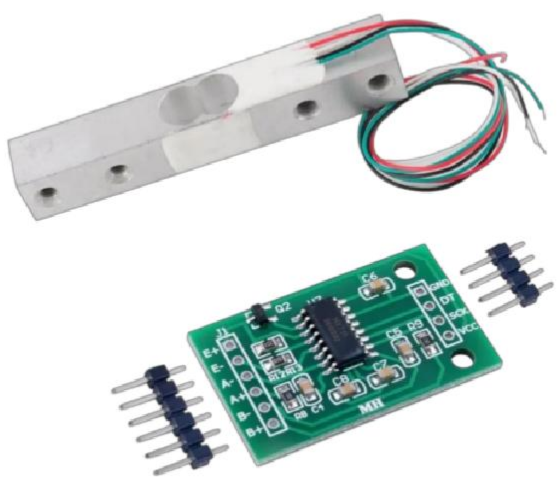

Fig.3.11: Load cell with hx711 sensor

Specification:

1. Operating voltage: 5 - 10 VDC
2. Operating current: 10 mA
3. Differential input voltage: ± 40 mV

4. Data accuracy: 24 bits
5. Refresh frequency: 10/80 Hz
6. Operating Temperature: -40 to +85 °C

3.2.11. Push button

They are used to manually use the elevator. This push buttons can be used as floor number burtons, lift arriving button, etc.

Working Principle:

A push button works by completing an electrical circuit when pressed. Inside, it has two metal contacts separated by a non-conductive material. When pushed, the contacts connect, allowing current to flow. Upon release, the contacts disconnect, breaking the circuit. This change in electrical state can be detected by a microcontroller or other circuitry, triggering desired actions

Fig.3.12: Push button

Specification:

1. Voltage rating: 24 V DC
2. Current rating: 50 mA
3. Insulation resistance: 100 MΩ
4. Operating force: 2.55±0.69 N
5. Contact resistance: 100 mΩ
6. Operating temperature range: -20 to +70 °C

3.2.12. Motor Driver (L293)

L293D Motor Driver Module is a medium power motor driver perfect for driving DC Motors and Stepper Motors. It uses the popular L293 motor driver IC. It can drive 4 DC motors on and off, or drive 2 DC motors with directional and speed control.

Working Principle:

The L293 motor driver circuit facilitates bidirectional control of DC motors. It operates by controlling the flow of current through its output channels using external logic signals. By varying these signals, the circuit can control the direction and speed of connected motors making it suitable for applications like robotics and motorized projects.

Fig.3.13: Motor driver

Specification:

1. Supply Voltage: 4.5V to 36V
2. Output Current: 600mA per channel (1.2A peak)
3. Number of Channels: 4 (can drive two DC motors bidirectionally)
4. Input Logic Voltage: TTL-compatible
5. Protection: Thermal shutdown and internal clamp diodes for overcurrent protection
6. Package: Available in DIP and surface-mount packages
7. Operating Temperature: -40°C to 125°C
8. Control Interface: Simple logic control for direction and speed control of motors

3.2.13. DC Motor

These motor is simple DC Motor featuring gears for the shaft for obtaining the optimal performance characteristics. They are known as Center Shaft DC Geared Motors because their shaft extends through the center of their gearbox assembly. We are using it for moving lift up-down.

Working Principle:

Center shaft DC geared motors typically consist of a DC motor with a gearbox attached to its shaft. The gearbox reduces the speed of the motor while increasing its torque. The center shaft design means that the output shaft is located at the center of the motor body, providing balanced rotational movement. These motors are commonly used in applications requiring precise control of speed and direction, such as robotics, conveyor systems, and automation equipment.

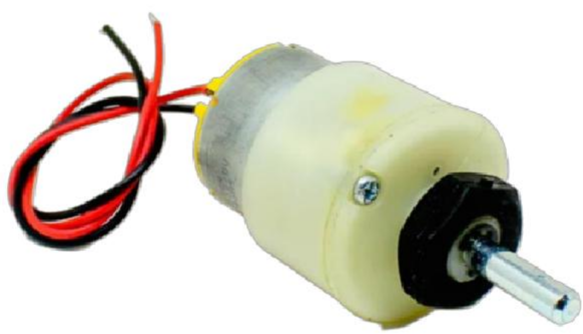

Fig.3.14: DC Motor

Specification:

1. Rated Speed: 200 RPM
2. Operating Voltage: 12 V DC
3. Load Current Max: 300 mA
4. No-Load Current: 60 mA
5. Rated Torque: 1.5 kg-cm
6. Stall Torque: 5.4 kg-cm

3.2.14. Transformer

Steps down the AC mains voltage to a suitable level for the rest of the system.

Working Principle:

It achieves voltage step down through a process of electromagnetic induction, where alternating current in the primary coil induces a changing magnetic field, which in turn induces a voltage in the secondary coil, scaled down to 12 volts. The transformer's construction ensures that the output voltage remains stable at 12 volts, while its current rating of 2A indicates the maximum continuous current it can safely supply to the connected load.

Fig.3.15: Transformer

Specification:

1. Input Voltage: 230 V AC
2. Output Voltage: 12 V AC
3. Output Current: 2 A
4. Mounting: Vertical mount type
5. Winding: Copper

3.2.15. Diode (1N4007)

A diode is a two terminal electronic component that conducts primarily in one direction. It has low resistance to the current in one direction and high resistance in other. The most common function of a diode is to allow an electric current to pass in one direction while blocking current in the opposite direction. This unidirectional behavior is called rectification and it is used to convert ac to dc. Here we use 1N4007 diodes.

Fig.3.16: Diode

3.2.16. [illegible]

C[illegible] a[illegible] c[illegible] i[illegible] e[illegible] a[illegible]

Fig.3.17: Capacitor

3.2.17. Resistor

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. Resistors act to reduce current flow, and, at the same time, act to lower voltage levels within circuits. Here, in our project we are using 10Ω, 1kΩ, 2.2KΩ and 10KΩ.

Fig.3.18: Resistor

3.2.18. Voltage regulator (LM-7812)

The LM7812 is a voltage regulator integrated circuit (IC) designed to provide a stable output voltage of +12 volts. It operates by taking in a higher input voltage, typically from a DC power source, and regulating it down to +12 volts DC

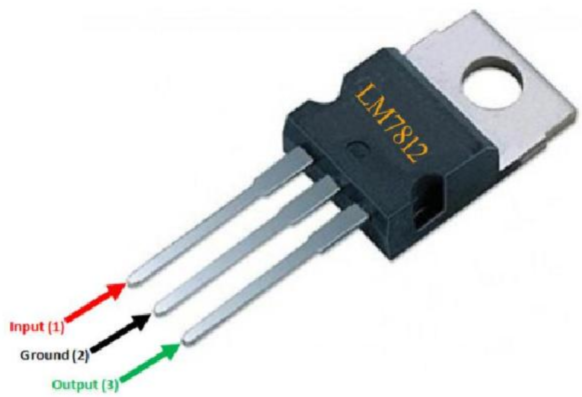

Fig.3.19: Voltage regulator IC

Specification:

1. Output Voltage: +12 volts
2. Input Voltage Range: Typically up to 35 volts
3. Output Current: Up to 1.5 amps (1.0 amp for some variants)
4. Dropout Voltage: Typically around 2 volts
5. Output Voltage Tolerance: ±4%
6. Quiescent Current: Typically around 5-10 mA
7. Operating Temperature Range: 0°C to 125°C

3.3 CIRCUIT DIAGRAM:

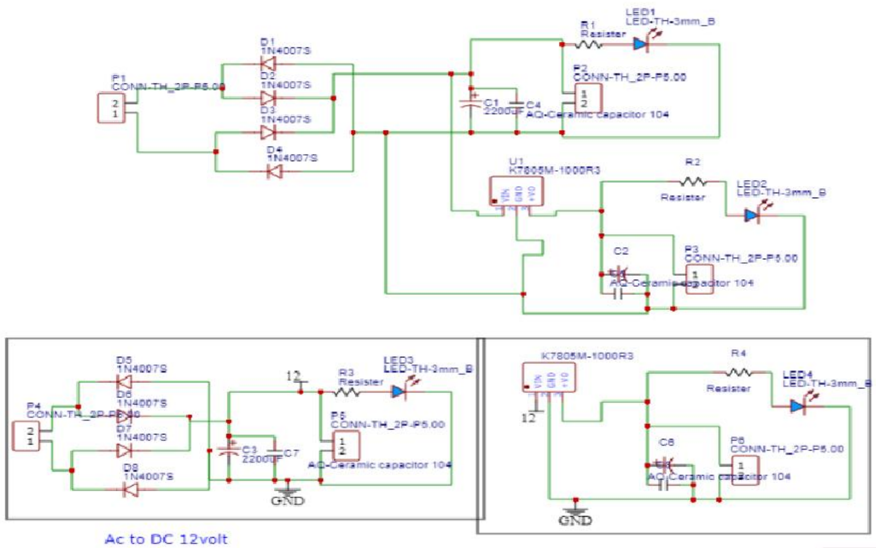

Fig.3.20: Circuit Diagram of Power Supply

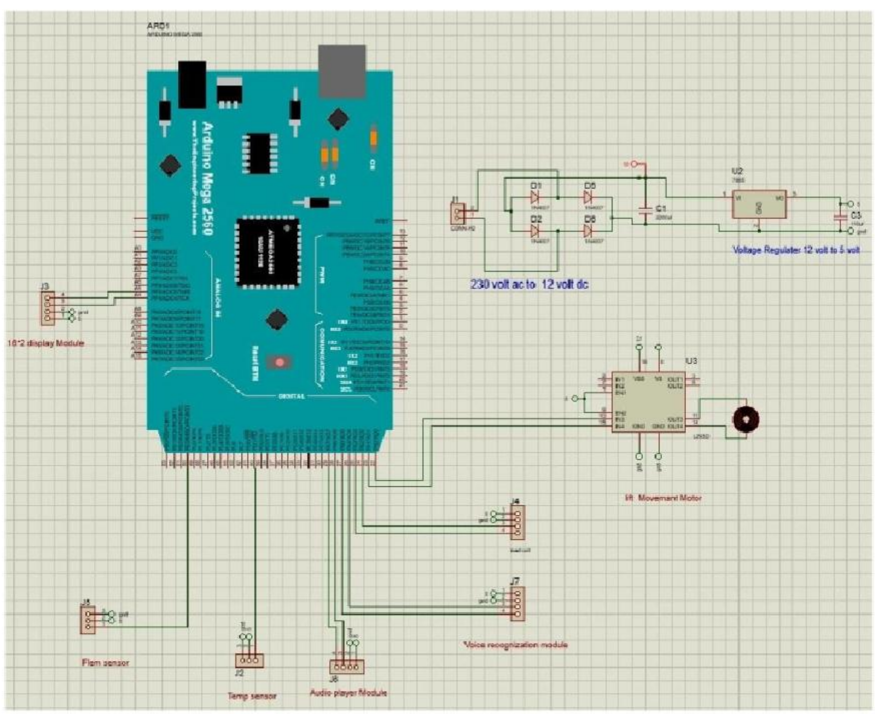

Fig.3.21 Circuit Diagram of System

3.4 ALGORITHM

1. Initialization:
   - Initialize the system, including the Arduino Mega and peripheral devices.
   - Set up the LCD display for real-time information.
2. Sensor Monitoring:

   Continuously monitor the Load sensor, Flame sensor, Temperature sensor
3. Sensor Fault Check:
   - Check for sensor faults:
   - If any sensor detects an issue, proceed to error handling.
4. Voice Command Detection:
   - Utilize the voice recognition module to listen for voice commands from users.
5. Voice Command Processing:
   - Process the recognized voice command:
   - Identify the floor to which the user wants to go.
   - Translate the voice command into a numerical floor selection.
6. Display Selection:
   - Display the selected floor on the LCD for user confirmation.
7. Elevator Motor Operation:
   - Control the DC motor to move the elevator to the selected floor:
   - Adjust the motor's direction and movement.
   - Announce the selected floor using the audio player module.
8. Manual Input Check:
   - Continuously monitor for manual inputs, such as physical switches and IR sensors.
9. Manual Input Handling:
   - If manual input is detected, adjust elevator operation based on the input.
10. Status Update:
    - Continuously update the LCD display with the current system status.
11. Error Handling:
    - If a sensor fault occurs:
    - Display an error message on the LCD.
    - Stop the elevator motor for safety.
12. End:
    - The software's main loop continues, ensuring uninterrupted operation.

3.5 FLOWCHART:

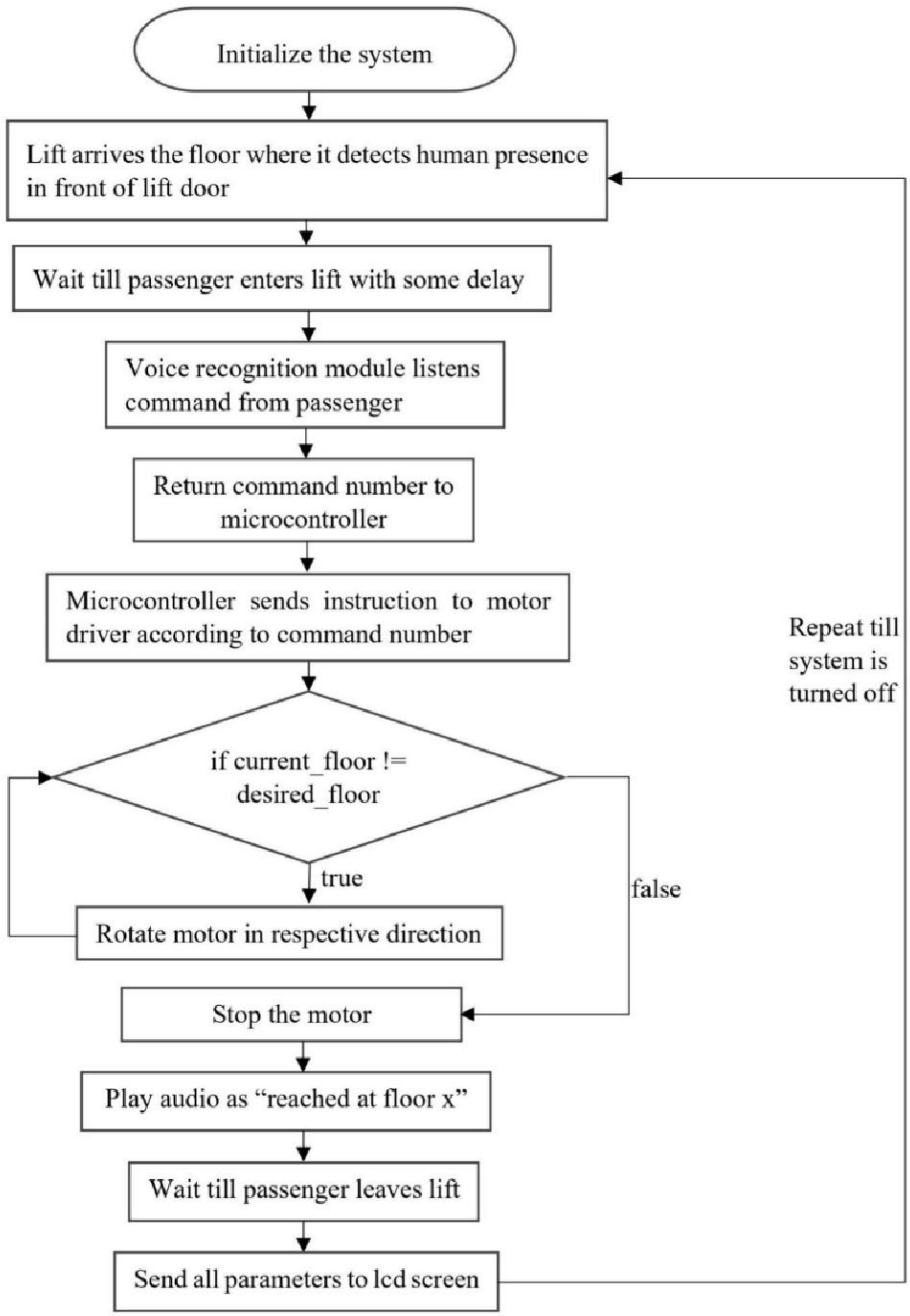

Fig 3.22: Flowchart of System

Flowchart Description:

1. Initialize the system.
2. Lift arrives the floor where it detects human presence in front of lift door.
3. User enters into lift.
4. Listen for a voice command from the user.
5. Compare the voice command with already stored commands and return command number using the voice recognition module.
6. Microcontroller determines which floor the user wants to go to base on the command value returned by voice recognition module.
7. Send a signal to the relay driver circuit to activate the DC motor in the desired direction.
8. Monitor the current floor number using the Ultrasonic sensor.
9. When the lift car reaches the desired floor, send a signal to the relay driver circuit to deactivate the DC motor.
10. Play an audio cue to inform the user that they have arrived at their destination.
11. While all this is working, the floor number, motor temperature, flame sensor, and load sensor parameters are sent to lcd screen for monitoring purpose.
12. Repeat steps 2-12 until the system is shut down.

3.6 CODE OF SYSTEM :

```
const int trigPin = 25;
const int echoPin = 24;
long duration;
int distance;
int alert;
int in0, in1, in2;
//////////////////////////////////////
const int channelM0 = 5;
const int channelM1 = 6;
const int channelM2 = 7;
/////////////////////////////////////
const int irSensorPin3 = 47;
const int irSensorPin2 = 45;
const int irSensorPin1 = 26;
////////////////////////////////////////
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
/////////////////////////////////////////////////////////////////
int irSensorValue3;
int irSensorValue2;
int irSensorValue1;
/////////////////////////////////////////////////////////////////////////////////
byte pinsOut[] = { 30, 31, 32, 33, 34, 35, 36, 37 };
int m = 0;  // variable for storing the pin of APR33A3
//////////////////////////////////////////////////////////////////////////////////////
/**
*****************************************************************************
***
  * @file    vr_sample_control_led.ino
  * @author  JiapengLi
  * @brief   This file provides a demostration on
             how to control led by using VoiceRecognitionModule
```

```
*****************************************************************************
***
 * @note:
     voice control led
*****************************************************************************
***
 * @section  HISTORY

   2013/06/13    Initial version.
 */
#include <SoftwareSerial.h>
#include "VoiceRecognitionV3.h"
/**
 Connection
 Arduino    VoiceRecognitionModule
  2   ------->     TX
  3   ------->     RX
*/
VR myVR(10, 11);  // 2:RX 3:TX, you can choose your favourite pins.
uint8_t records[7]; // save record
uint8_t buf[64];
int led = 13;
#define ground (0)
#define first (1)
#define second (2)

/**
 @brief   Print signature, if the character is invisible,
          print hexible value instead.
 @param   buf     --> command length
          len     --> number of parameters
*/
void printSignature(uint8_t *buf, int len)
{
```

```
  int i;
  for(i=0; i<len; i++){
    if(buf[i]>0x19 && buf[i]<0x7F){
      Serial.write(buf[i]);
    }
    else{
      Serial.print("[");
      Serial.print(buf[i], HEX);
      Serial.print("]");
    }
  }
}

/**
  @brief   Print signature, if the character is invisible,
           print hexible value instead.
  @param   buf  -->  VR module return value when voice is recognized.
             buf[0]  -->  Group mode(FF: None Group, 0x8n: User, 0x0n:System
             buf[1]  -->  number of record which is recognized.
             buf[2]  -->  Recognizer index(position) value of the recognized record.
             buf[3]  -->  Signature length
             buf[4]~buf[n] --> Signature
*/
void printVR(uint8_t *buf) {
  Serial.println("VR Index\tGroup\tRecordNum\tSignature");
  Serial.print(buf[2], DEC);
  Serial.print("\t\t");
  if(buf[0] == 0xFF){
    Serial.print("NONE");
  }
  else if(buf[0]&0x80){
    Serial.print("UG ");
    Serial.print(buf[0]&(~0x80), DEC);
  }
  else{
```

```
    Serial.print("SG ");
    Serial.print(buf[0], DEC);
  }
  Serial.print("\t");
  Serial.print(buf[1], DEC);
  Serial.print("\t\t");
  if(buf[3]>0){
    printSignature(buf+4, buf[3]);
  }
  else{
    Serial.print("NONE");
  }
  Serial.println("\r\n");
}
///////////////////////////////////////////////////////////////////////////
const int ANALOG_PIN = A8;
int sensorValue;
////////////////////////////////////////////////////////////////////////////////
const int motor1A = 22;  // IN1
const int motor1B = 23;  // IN2
/////////////////////////////////////////////////////////////////////////
const int buttonPin1 = 50;
const int buttonPin2 = 53;
const int buttonPin3 = 52;
int flagup;
int flagdown;
int flagmedium;
int status;
/////////////////////////////////////////////////////////////////////////////////////////
#include <HX711_ADC.h>
#if defined(ESP8266) || defined(ESP32) || defined(AVR)
#include <EEPROM.h>
#endif
const int HX711_dout = 49;  //mcu > HX711 dout pin
const int HX711_sck = 48;   //mcu > HX711 sck pin
```

```
HX711_ADC LoadCell(HX711_dout, HX711_sck);
const int calVal_eepromAdress = 0;
unsigned long t = 0;
float i;
/////////////////////////////////////////////////////////////////
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 51
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float temperatureC;
/////////////////////////////////////////////////////////////////
void setup() {
  pinMode(channelM0, OUTPUT);
  pinMode(channelM1, OUTPUT);
  pinMode(channelM2, OUTPUT);
  digitalWrite(channelM0, HIGH);
  digitalWrite(channelM1, HIGH);
  digitalWrite(channelM2, HIGH);
  for (unsigned n = 0; n < 8; n++) {
    pinMode(pinsOut[n], OUTPUT);
    digitalWrite(pinsOut[n], HIGH);
  }
  Serial.begin(9600);
  pinMode(motor1A, OUTPUT);
  pinMode(motor1B, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  ///////////////////////////////////////////////////////////////////////
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(buttonPin3, INPUT);
  ///////////////////////////////////////////////////////////////////////
  sensors.begin();
  pinMode(irSensorPin1, INPUT);
```

```
pinMode(irSensorPin2, INPUT);
pinMode(irSensorPin3, INPUT);
///////////////////////////////////////////////////////////////////////
LoadCell.begin();
//LoadCell.setReverseOutput(); //uncomment to turn a negative output value to positive
float calibrationValue;    // calibration value (see example file "Calibration.ino")
calibrationValue = 696.0;  // uncomment this if you want to set the calibration value in the sketch
#if defined(ESP8266) || defined(ESP32)
//EEPROM.begin(512); // uncomment this if you use ESP8266/ESP32 and want to fetch the calibration value from eeprom
#endif
EEPROM.get(calVal_eepromAdress, calibrationValue);  // uncomment this if you want to fetch the calibration value from eeprom
unsigned long stabilizingtime = 2000;  // preciscion right after power-up can be improved by adding a few seconds of stabilizing time
boolean _tare = true;  //set this to false if you don't want tare to be performed in the next step
LoadCell.start(stabilizingtime, _tare);
if (LoadCell.getTareTimeoutFlag()) {
  Serial.println("Timeout, check MCU>HX711 wiring and pin designations");
  while (1);
} else {
  LoadCell.setCalFactor(calibrationValue);  // set calibration value (float)
  Serial.println("Startup is complete");
}
/////////////////////////////////////////////////////////////
myVR.begin(9600);
Serial.println("Elechouse Voice Recognition V3 Module\r\nControl LED sample");
if (myVR.clear() == 0) {
  Serial.println("Recognizer cleared.");
} else {
  Serial.println("Not find VoiceRecognitionModule.");
  Serial.println("Please check connection and restart Arduino.");
  while (1);
}
```

```
if (myVR.load((uint8_t)ground) >= 0) {
  Serial.println("ground loaded");
}
if (myVR.load((uint8_t)first) >= 0) {
  Serial.println("first loaded");
}
if (myVR.load((uint8_t)second) >= 0) {
  Serial.println(" second loaded");
}
/*
digitalWrite(pinsOut[7], LOW);
delay(1000);
digitalWrite(pinsOut[7], HIGH);
delay(1000) ;
digitalWrite(pinsOut[8], LOW);
delay(1000);
digitalWrite(pinsOut[8], HIGH);
delay(1000) ;
digitalWrite(pinsOut[0], LOW);
delay(1000);
digitalWrite(pinsOut[0], HIGH);
delay(1000) ;
*/
lcd.init();
// Print a message to the LCD.
lcd.backlight();
lcd.clear();
lcd.setCursor(3, 0);
lcd.print("Hello, world!");
lcd.setCursor(2, 1);
lcd.print("Ywrobot Arduino!");
lcd.clear();
}
void loop() {
  lcd.setCursor(5, 0);
```

```
lcd.print("Welcome");
lcd.setCursor(0, 1);
lcd.print("Temp= ");
lcd.setCursor(6, 1);
lcd.print(temperatureC);
flame_lift();
temp_lift();
if (temperatureC >= 50 || sensorValue <= 200) {
  lift_stop();
  digitalWrite(pinsOut[1], LOW);
  delay(1000);
  digitalWrite(pinsOut[1], HIGH);
  delay(1000);
}
else {
  int ret;
  ret = myVR.recognize(buf, 50);
  if (ret > 0) {
    switch (buf[1]) {
      case ground:
        in0 = 1;
        break;
      case first:
        in1 = 1;
        break;
      case second:
        in2 = 1;
        break;
      default:
        Serial.println("Record function undefined");
        break;
    }
    /** voice recognized */
    printVR(buf);
  }
```

```
int last_floor = digitalRead(buttonPin1);
int second_Floor = digitalRead(buttonPin2);
int ground_floor = digitalRead(buttonPin3);
irSensorValue1 = digitalRead(irSensorPin1);
irSensorValue2 = digitalRead(irSensorPin2);
irSensorValue3 = digitalRead(irSensorPin3);
if (last_floor == HIGH || irSensorValue3 == LOW) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("plz wait ");
  lcd.setCursor(0, 1);
  lcd.print("lift load check");
  delay(1000);
  lift_load_check();
  flame_lift();
  temp_lift();
  status = 2;
  flagup = 1;
}
if (second_Floor == HIGH || irSensorValue2 == LOW) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("plz wait ");
  lcd.setCursor(0, 1);
  lcd.print("lift load check");
  delay(1000);
  lift_load_check();
  flame_lift();
  temp_lift();
  flagmedium = 1;
}
if (ground_floor == HIGH || irSensorValue1 == LOW) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("plz wait ");
```

```
  lcd.setCursor(0, 1);
  lcd.print("lift load check");
  delay(1000);
  lift_load_check();
  flame_lift();
  temp_lift();
  flagdown = 1;
  status = 1;
}
////////////////////////////////////////////////////////////////////////////////
if (flagdown == 1 || in0 == 1) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Lift Call");
  lcd.setCursor(0, 1);
  lcd.print("Ground Floor");
  delay(1000);
  lcd.clear();
  digitalWrite(pinsOut[2], LOW);
  delay(500);
  digitalWrite(pinsOut[2], HIGH);
  delay(500);
  lift_down_ground();
}
if (flagup == 1 || in2 == 1) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Lift Call");
  lcd.setCursor(0, 1);
  lcd.print("Second Floor");
  delay(1000);
  lcd.clear();
  digitalWrite(pinsOut[3], LOW);
  delay(500);
  digitalWrite(pinsOut[3], HIGH);
```

```
    delay(500);
    lift_up_last();
  }
  if (flagmedium == 1 || in1 == 1) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Lift Call");
    lcd.setCursor(0, 1);
    lcd.print("First Floor");
    delay(1000);
    lcd.clear();
    digitalWrite(pinsOut[4], LOW);
    delay(500);
    digitalWrite(pinsOut[4], HIGH);
    delay(500);
    switch (status) {
     case 1:
      lift_medium_up();
      Serial.println("Sensor value is 1");
      break;
     case 2:
      lift_medium_ground();
      Serial.println("Sensor value is 2");
      break;
     default:
      Serial.println("Sensor value is not 1, 2, or 3");
      break;
    }
  }
 }
}

void ultra() {
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
```

```
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0351 / 2;
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
}

void lift_up_last() {
  if (temperatureC >= 50 || sensorValue <= 200) {
    lift_stop();
  }
  else {
    ultra();
    if (distance >= 10) {
      digitalWrite(motor1A, HIGH);
      digitalWrite(motor1B, LOW);
      status = 2;
    }
    else {
      digitalWrite(motor1A, LOW);
      digitalWrite(motor1B, LOW);
      digitalWrite(channelM2, LOW);
      delay(300);
      digitalWrite(channelM2, HIGH);
      flagup = 0;
      in0 = 0;
      in1 = 0;
      in2 = 0;
    }
  }
}
```

```
void lift_down_ground() {
 if (temperatureC >= 50 || sensorValue <= 200) {
  lift_stop();
 }
 else {
  ultra();
  if (distance <= 40) {
   digitalWrite(motor1A, LOW);
   digitalWrite(motor1B, HIGH);
   status = 1;
  }
  else {
   digitalWrite(motor1A, LOW);
   digitalWrite(motor1B, LOW);
   digitalWrite(channelM0, LOW);
   delay(300);
   digitalWrite(channelM0, HIGH);
   flagdown = 0;
   in0 = 0;
   in1 = 0;
   in2 = 0;
  }
 }
}

void lift_medium_up() {
 if (temperatureC >= 50 || sensorValue <= 200) {
  lift_stop();
 }
 else {
  ultra();
  if (distance >= 25) {
   digitalWrite(motor1A, HIGH);
   digitalWrite(motor1B, LOW);
  }
```

```
    else {
      digitalWrite(motor1A, LOW);
      digitalWrite(motor1B, LOW);
      digitalWrite(channelM1, LOW);
      delay(300);
      digitalWrite(channelM1, HIGH);
      flagmedium = 0;
      in0 = 0;
      in1 = 0;
      in2 = 0;
      status = 0;
    }
  }
}

void lift_medium_ground() {
  if (temperatureC >= 50 || sensorValue <= 200) {
    lift_stop();
  }
  else {
    ultra();
    if (distance <= 25) {
      digitalWrite(motor1A, LOW);
      digitalWrite(motor1B, HIGH);
    }
    else {
      digitalWrite(motor1A, LOW);
      digitalWrite(motor1B, LOW);
      flagmedium = 0;
      in0 = 0;
      in1 = 0;
      in2 = 0;
      status = 0;
    }
  }
```

```
}

void lift_stop() {
 digitalWrite(motor1A, LOW);
 digitalWrite(motor1B, LOW);
 delay(1000);
 in0 = 0;
 in1 = 0;
 in2 = 0;
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Waiting ");
 lcd.setCursor(0, 1);
 lcd.print("Temp= ");
 lcd.setCursor(6, 1);
 lcd.print(temperatureC);
 lcd.clear();
}

void temp_lift() {
 sensors.requestTemperatures();
 temperatureC = sensors.getTempCByIndex(0);
 if (temperatureC != DEVICE_DISCONNECTED_C) {
  Serial.print("Temperature: ");
  Serial.print(temperatureC);
  Serial.println("°C");
 }
 else {
  Serial.println("Error reading temperature!");
 }
 if (temperatureC <= 50) {
  Serial.println("temp ok");
  alert = 0;
 }
 else {
```

```
    Serial.println("temp HIGH");
    alert = 1;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Lift Temp");
    lcd.setCursor(5, 1);
    lcd.print("HIGH");
  }
}

void flame_lift() {
  sensorValue = analogRead(ANALOG_PIN);
  Serial.print("Sensor Value: ");
  Serial.println(sensorValue);
  if (sensorValue >= 200) {
    Serial.println("no fire");
    alert = 0;
  }
  else {
    Serial.println("Fire detected");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Lift Fire");
    lcd.setCursor(0, 1);
    lcd.print("detected");
    digitalWrite(pinsOut[5], LOW);
    delay(500);
    digitalWrite(pinsOut[5], HIGH);
    delay(500);
    alert = 1;
  }
}

void weightCheck() {
  static boolean newDataReady = 0;
```

```
const int serialPrintInterval = 0;  //increase value to slow down serial print activity
// check for new data/start next conversion:
if (LoadCell.update()) newDataReady = true;
// get smoothed value from the dataset:
if (newDataReady) {
  if (millis() > t + serialPrintInterval) {
    i = LoadCell.getData();
    Serial.print("Load_cell output val: ");
    Serial.println(i);
    newDataReady = 0;
    t = millis();
  }
}
// receive command from serial terminal, send 't' to initiate tare operation:
if (Serial.available() > 0) {
  char inByte = Serial.read();
  if (inByte == 't') LoadCell.tareNoDelay();
}
// check if last tare operation is complete:
if (LoadCell.getTareStatus() == true) {
  Serial.println("Tare complete");
}
}

void lift_load_check() {
 for (int i = 1; i <= 500; i++) {
  weightCheck();
  Serial.println(i);
 }
 if (i < 1) {
  Serial.println("lift not Overload");
  lcd.setCursor(0, 0);
  lcd.print(" lift NOT Overload ");
  lcd.setCursor(5, 1);
  lcd.print(" Overload ");
```

```
    lcd.clear();
  }
  else {
    lcd.setCursor(0, 0);
    lcd.print(" lift Overload ");
    lcd.clear();
    Serial.println("lift Overload");
    digitalWrite(pinsOut[6], LOW);
    delay(500);
    digitalWrite(pinsOut[6], HIGH);
    delay(500);
    flagup = 0;
    flagmedium = 0;
    flagdown = 0;
    lift_stop();
    delay(1000);
  }
}
```

3.7 SYSTEM BUDGET ANALYSIS:

Sr. No	Components	Price (₹)
1	Arduino Mega	1600
2	Voice recognition module (V3)	3540
3	Audio Player module (APR33A3)	590
4	Speaker	200
5	16×2 LCD display	130
6	I2C serial interface adapter module	80
7	Motor driver (L293)	90
8	DC motor	200
9	Ultrasonic sensor (HC-SR04)	58
10	IR sensor (IR-08H) (3 pcs)	315
11	Load cell & hx711 sensor	354
12	Flame sensor (YL 38)	55
13	Temperature sensor (ds18b20)	177
14	Push button (4 pcs)	60
15	Transformer (12-0-12)	236
16	Diode (1N4007) (4 pcs)	4
17	Capacitor (2 pcs)	10

18	Resistor (4 pcs)	4
19	Voltage regulator (LM-7812)	10
20	Wires	332
21	Lift structure	1180
	TOTAL	9225

Table 4 : Budget Table

CHAPTER 4
DISCUSSION ON RESULTS

4.1 Description of All Results :

The voice-operated lift control system represents a significant milestone in the evolution of elevator technology, offering a blend of convenience, safety, and user-friendliness that surpasses traditional control mechanisms. At the core of this innovation is the utilization of voice commands for floor selection, a feature that not only simplifies the user experience but also caters to individuals with physical disabilities or those carrying heavy loads. By simply speaking their desired floor, users can initiate the elevator's movement, eliminating the need for manual button presses or keypad entries. This seamless interaction between humans and machines not only enhances efficiency but also fosters a sense of inclusivity within the building environment.

A key aspect of the system's functionality lies in its real-time monitoring capabilities, facilitated by an array of sensors strategically placed within the lift infrastructure. These sensors continuously track parameters such as load, temperature, and the presence of any potential fire hazards, ensuring that the operating environment remains safe at all times. Should any anomalies be detected, the system promptly responds with appropriate error handling procedures, including displaying relevant error messages and halting lift operation if necessary. This proactive approach to safety underscores the system's commitment to user well-being, mitigating potential risks before they escalate into emergencies.

Fig.4.1 Start Screen

The system's ability to precisely interpret voice commands and translate them into actionable instructions for the elevator motor enables smooth and efficient navigation between floors. Users can rest assured that their desired destinations will be accurately reached, thanks to the system's robust control algorithms and responsive motor operation. Furthermore, the inclusion of manual input handling mechanisms provides an additional layer of flexibility, allowing users to override voice commands in emergency situations or when alternative control methods are preferred.

Fig.4.2 Floor description

The integration of a comprehensive feedback mechanism, displayed in real-time on an LCD screen within the lift cabin, further enhances the user experience by providing essential information such as current floor status, motor temperature, and sensor readings. This transparency not only instills confidence in the system's reliability but also empowers users to make informed decisions while utilizing the lift. In essence, the voice-operated lift control system represents a paradigm shift in elevator design, prioritizing user-centricity, safety, and efficiency in equal measure. As this technology continues to evolve, it holds the potential to redefine the future of vertical transportation, creating smarter, more accessible, and safer urban environments for all.

Fig.4.3 Load check Screen

Fig.4.4 Fire detection screen

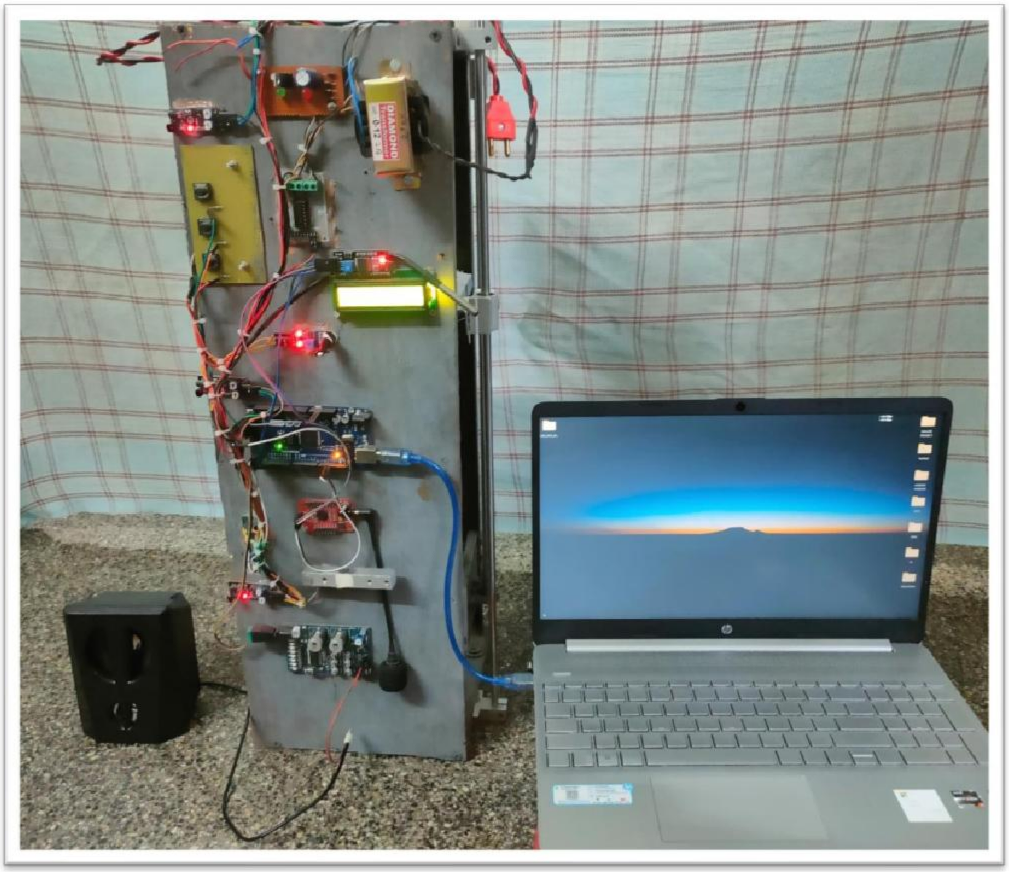

Fig.4.5 Prototype image

CHAPTER 5
SYSTEM OVERVIEW

5.1 ADVANTAGES:

1. Enhanced Accessibility and Hygiene: Voice control simplifies operation, making the elevator user-friendly for all, while reducing germ transmission by eliminating the need for button presses.
2. Safety: The system includes safety features such as load sensors, IR sensors, flame sensors, temperature sensors, and ultrasonic sensors, which can help to prevent accidents.
3. Advanced Safety: Integrated sensors monitor critical parameters, ensuring immediate error handling and preventing accidents, guaranteeing a secure journey for passengers.
4. Clear Communication and Emergency Response: Auditory feedback provides floor updates, while voice-activated emergency features enable swift communication with building management or emergency services, enhancing safety and convenience.
5. Future-Proof Technology: The integration of voice control ensures adaptability to future advancements, prolonging the system's relevance and maximizing return on investment for building owners.

5.2 DISADVANTAGES:

1. Complexity: Implementing voice recognition and multiple sensors can add complexity to the system and can also be little bit hard to program such complex hardware.
2. Maintenance: Advanced features may require more maintenance and troubleshooting and annual audit to check the working of lift.
3. Cost: The additional components and technology can increase the overall cost of the system, especially voice recognition modules are costly as well as they require high end hardware
4. Voice recognition: Currently voice recognition is user dependent, so for user-independent voice recognition high end module is needed which increases system cost further.
5. User Adaptation: Users may need time to adapt to voice commands and auditory feedback.

5.3 APPLICATIONS

1. Commercial Buildings: Implementing the voice-operated lift system in office buildings, shopping malls, and hotels can improve accessibility and user experience for employees, customers, and guests while ensuring their safety and providing clear communication on floor updates.
2. Hospitals and Healthcare Facilities: In healthcare settings, where hygiene and accessibility are paramount, the voice-operated lift can assist patients, visitors, and healthcare staff in navigating the facility safely and efficiently, while also minimizing the risk of germ transmission.
3. Residential Complexes: Installing the voice-operated lift system in residential buildings and apartment complexes can enhance convenience for residents, especially those with mobility issues or disabilities, while also providing peace of mind through advanced safety features and emergency response capabilities.
4. Transport Hubs: Airports, train stations, and bus terminals can benefit from the voice-operated lift system to facilitate the movement of passengers with luggage or mobility restrictions, while also improving efficiency and communication in busy transit environments.
5. Government Buildings: Government offices, courthouses, and municipal buildings can enhance accessibility and safety for citizens and employees by implementing the voice-operated lift system, ensuring compliance with accessibility regulations, and providing clear communication during emergencies.
6. Educational Institutions: Universities, colleges, and schools can improve accessibility and safety for students, faculty, and visitors by installing the voice-operated lift system in campus buildings, while also promoting hygiene and reducing the risk of accidents through advanced safety features.
7. Retail Stores: Retailers can enhance the shopping experience for customers by installing the voice-operated lift system in multi-level stores, improving accessibility for shoppers of all abilities, and providing clear communication on floor locations and promotions.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion:

Voice-controlled elevators are a long-term solution that can be used by anyone, including people with disabilities. They have the potential to make life easier for everyone and reduce the spread of germs. In addition to the benefits mentioned above, voice-controlled elevators could also be used to improve security and convenience. For example, authentication could be used to restrict access to certain floors, and sensors could be used to reduce the need for users to give specific commands. Voice-controlled elevators have a wide range of potential applications, including in homes, offices, hospitals, hostels, and public places. Overall, voice-controlled elevators are a promising technology with the potential to improve our lives in many ways.

6.2 Future Scope:

This system paves the way for a future where interacting with elevators feels as natural as having a conversation. Imagine effortlessly telling the lift where you're headed using complete sentences, or the system recognizing your voice and whisking you to your usual floor without needing to press a button. But this isn't just about convenience. Advanced safety features could analyse sensor data to predict potential problems before they occur, preventing breakdowns and ensuring a smooth ride. In case of emergencies, a two-way voice communication system would connect you directly with a designated responder, providing immediate assistance.

The auditory experience within the lift would also become more dynamic and personalized. Gone are the days of repetitive elevator music. The system could adapt to your preferences, offering calming music for a relaxing journey or informative announcements about upcoming events on your destination floor. Integration with smart devices would allow you to seamlessly call the lift using voice commands on your phone or wearable, further streamlining the process.

Accessibility would be a core principle in this future. Users with visual impairments could navigate the lift with ease using voice commands, eliminating the need for physical buttons. The lift could even integrate with wheelchairs or other mobility aids, automatically adjusting its height to ensure a safe and comfortable entry and exit. Additionally, clear audio descriptions would detail all lift features and controls, catering to users with various needs.

For a touch of futuristic elegance, the lift could incorporate touchless controls alongside traditional buttons, offering a more hygienic option. Real-time floor information displayed within the lift and announcements for upcoming stops would keep you informed throughout your journey. Finally, the system would prioritize energy efficiency, potentially utilizing features like regenerative braking or smart lighting to minimize the lift's environmental impact.

By focusing on these advancements, voice-operated lifts are poised to revolutionize how we travel within buildings. They won't just be a means of transportation, but an intelligent and user-friendly extension of the smart building experience.

Voice Operated Lift Control System with Safety

ORIGINALITY REPORT

17%	13%	07%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.irjmets.com Internet Source	6%
2	forum.arduino.cc Internet Source	3%
3	Submitted to Savitribai Phule Pune University Student Paper	2%
4	www.ijser.org Internet Source	1%
5	st.robu.in Internet Source	1%
6	Submitted to Midlands State University Student Paper	1%
7	rifkifirdaus152060.blogspot.com Internet Source	1%
8	www.ijraset.com Internet Source	1%
9	robu.in Internet Source	1%

10	www.jetir.org Internet Source	<1%
11	vertrektnaso.com Internet Source	<1%
12	Submitted to Gitam University Student Paper	<1%
13	Submitted to Technological University Dublin Student Paper	<1%
14	ijaems.com Internet Source	<1%
15	Submitted to University of Technology Bahrain Student Paper	<1%
16	ten-tools.com Internet Source	<1%
17	www.i-scholar.in Internet Source	<1%
18	forum.mysensors.org Internet Source	<1%
19	fabacademy.org Internet Source	<1%
20	lambdageeks.com Internet Source	<1%
21	Submitted to St Mary MacKillop College	

	Student Paper	<1%
22	Submitted to Universiti Teknologi Malaysia Student Paper	<1%
23	es.scribd.com Internet Source	<1%
24	Submitted to Somerset College of Arts and Technology, Somerset Student Paper	<1%
25	fastercapital.com Internet Source	<1%
26	Submitted to Australian National University Student Paper	<1%
27	Submitted to CSU, San Jose State University Student Paper	<1%
28	community.blynk.cc Internet Source	<1%
29	electronicsworkshops.com Internet Source	<1%
30	lists.infradead.org Internet Source	<1%
31	ijariie.com Internet Source	<1%

32	Submitted to Institute of Technology Blanchardstown Student Paper	<1%
33	Submitted to University of Mauritius Student Paper	<1%
34	pdfcoffee.com Internet Source	<1%
35	Submitted to Kenyatta University Student Paper	<1%
36	yantrabazaar.com Internet Source	<1%
37	Submitted to Kampala International University Student Paper	<1%
38	Submitted to Macquarie University Student Paper	<1%
39	Agus Kurniawan. "Beginning Arduino Nano 33 IoT", Springer Science and Business Media LLC, 2021 Publication	<1%
40	Naoya Onizawa, Daisaku Katagiri, Warren J. Gross, Takahiro Hanyu. "Analog-to-stochastic converter using magnetic-tunnel junction devices", Proceedings of the 2014 IEEE/ACM International Symposium on Nanoscale Architectures, 2014	<1%

Publication

41	Submitted to Universitas Diponegoro Student Paper	<1 %
42	www.elprocus.com Internet Source	<1 %
43	Submitted to Universidad TecMilenio Student Paper	<1 %
44	Submitted to University of Wolverhampton Student Paper	<1 %
45	berlmathges.de Internet Source	<1 %
46	www.intersil.com Internet Source	<1 %
47	www.suntarapornorg.com Internet Source	<1 %
48	deepai.org Internet Source	<1 %
49	www.indiamart.com Internet Source	<1 %
50	Submitted to De Montfort University Student Paper	<1 %
51	Submitted to University of Kent at Canterbury Student Paper	<1 %
52	Submitted to University of Portsmouth	

Student Paper

<1%

53	m.blog.naver.com	Internet Source	<1%
54	sallyfitzgibbonsfoundation.com	Internet Source	<1%
55	vdocuments.mx	Internet Source	<1%
56	www.highfrequencytransformer.net	Internet Source	<1%
57	www.seekic.com	Internet Source	<1%
58	Subathra, Arul Prakash A, Vijay Ramalingam, S. Vignesh. "Arduino Uno-Powered Parking Guidance System with Ultrasonic Sensors", 2023 1st International Conference on Optimization Techniques for Learning (ICOTL), 2023	Publication	<1%
59	Tej Prakash Sharma, Akansha Solanki, Tarun Jain, Aavaig Malhotra, Monica Bhutani. "Wi-Fi based quadcopter drone with battery monitoring and optimization using crazyflie platform", International Journal of Information Technology, 2023	Publication	<1%

60	repositorio.unitau.br:8080 Internet Source	<1%
61	vman.cc Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		