# SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLGY(Autonomous)

**Batch No** **:** 21

**PROJECT NAME** **:**Touch-Free Switch Control Panel for Elevators

**GROUP MEMBERS:** A.Srihari Reddy(21781A0406)

G.Venugopal Reddy(21781A0474)

G.Sreekanth Reedy(21781A0476)

H.Pavankalyan (21781A0479)

## Problem Statement No 4

# Touch-Free Switch Control Panel for Elevators

### AIM OF THE PROJECT:

Design and develop a touch-free switch control panel for elevators to enhance user experience, accessibility, and hygiene.

### PROBLEM STATEMENT AND SOLUTION:

Problem: Traditional elevator control panels require physical contact, posing health risks and accessibility barriers.

Solution: Develop a touch-free control panel using gesture and voice recognition technologies.

### PROJECT DESIGN SPECIFICATION:

### Components:

1. User-friendly interface
2. Gesture and voice recognition capabilities
3. Integration with existing elevator systems
4. Power supply: 12V DC
5. Dimensions: 12" x 12" x 2"

## 2. Functional Requirements:

### 1.User-friendly interface:

In the mystical land of technology, where buttons and screens once ruled, a new magic emerged: touch-free user interfaces. These interfaces allowed mere mortals to command their devices without physical contact. No more grubby fingerprints on elevator buttons just elegant hand gestures.

### 2.Gesture and voice recognition capabilities:

Gesture recognition is a touchless technology that allows devices to understand and respond to human movements as if they were magical commands. Picture this: you raise your hand, and the elevator obeys no buttons, no fuss.

Close your eyes and imagine: you speak, and the elevator listens. Voice recognition transforms spoken words into actionable commands.

### 3.Integration with existing elevator systems:

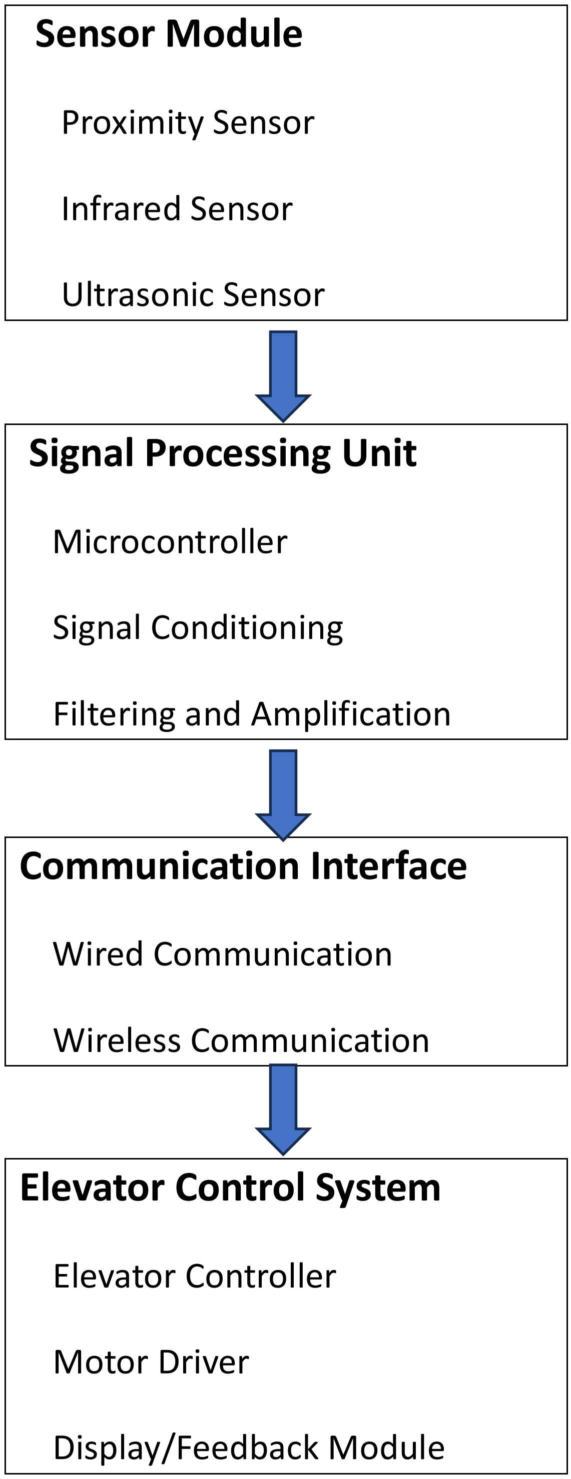
Integrating an elevator into an existing building or incorporating one into a new construction project involves a fascinating blend of architectural design, engineering, and technological wizardry.

### 4. Power supply: 12V DC :

When it comes to ensuring elevator reliability during power outages, battery backup systems step onto the stage. These systems provide a safety net, allowing elevators to gracefully descend or reach a designated floor even when the main power falters.

# PROJECT ARCHITECTURE & METHODOLOGY:

## Touch-Free Switch Control Panel Interface



**Sensor Module**

Proximity Sensor

Infrared Sensor

Ultrasonic Sensor

**Signal Processing Unit**

Microcontroller

Signal Conditioning

Filtering and Amplification

**Communication Interface**

Wired Communication

Wireless Communication

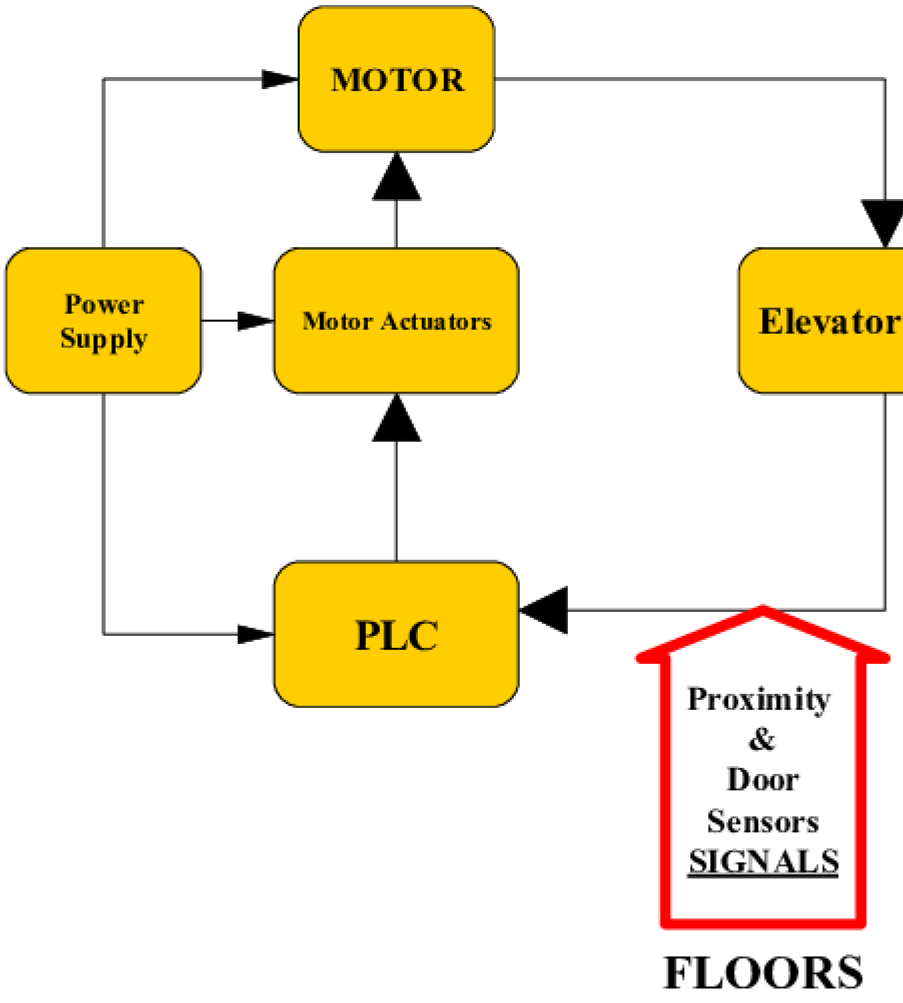
**Elevator Control System**

Elevator Controller

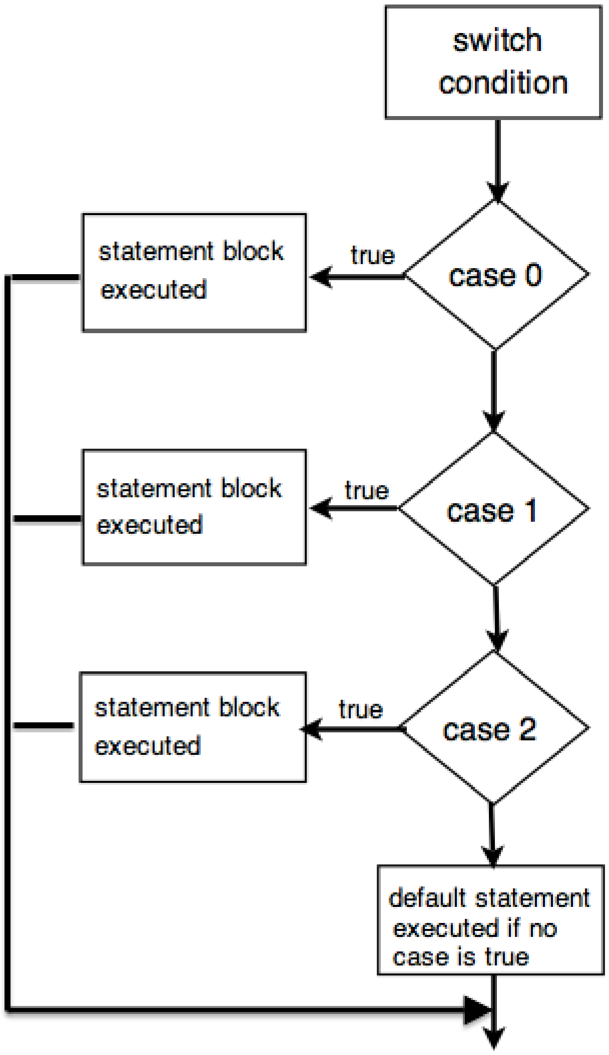
Motor Driver

Display/Feedback Module

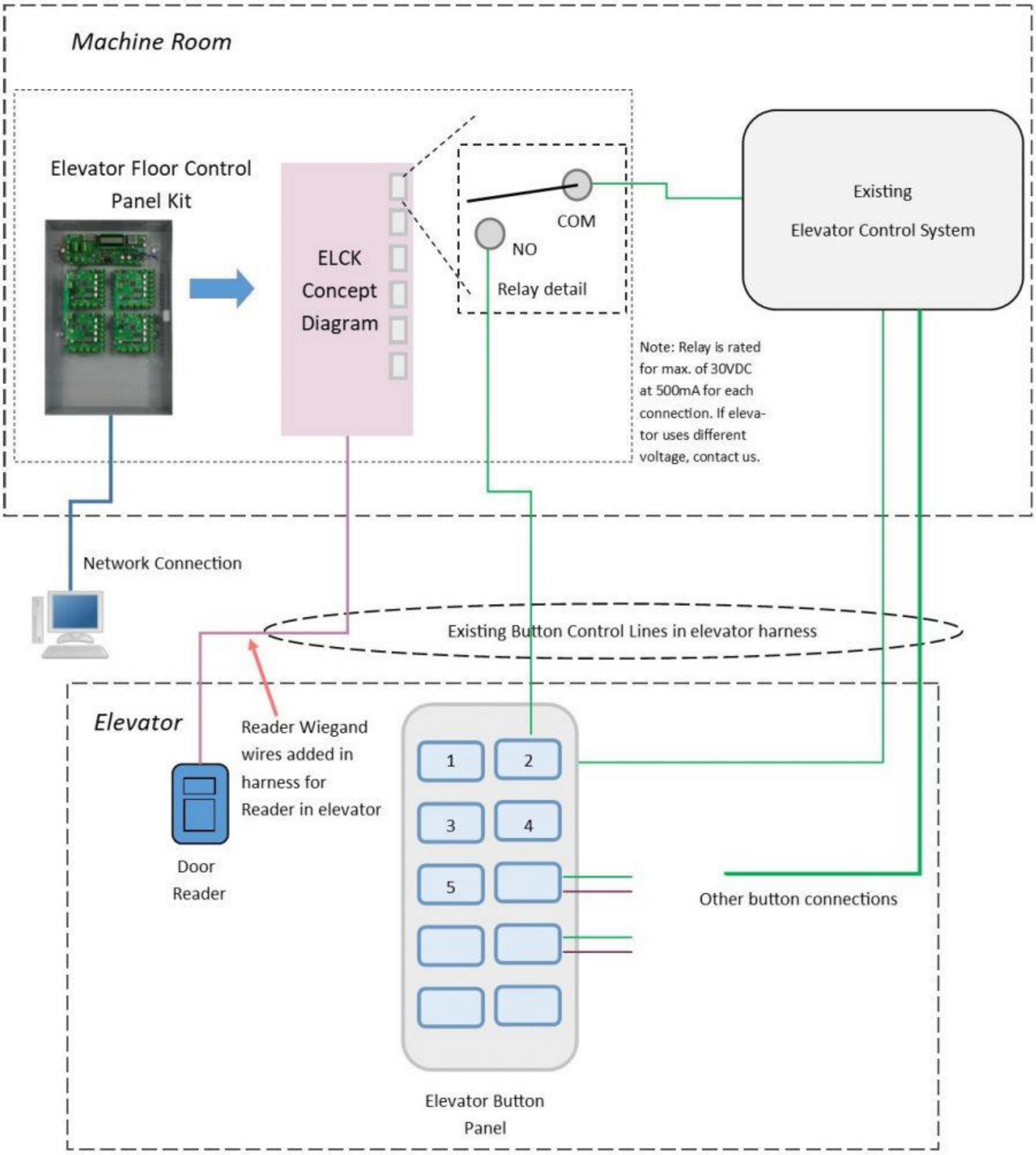
## BLOCK DIAGRAM:



## FLOW EXPLANATION:



# CIRCUIT DIAGRAM:


Machine Room
Elevator Floor Control Panel Kit
ELCK Concept Diagram
COM
NO
Relay detail
Existing Elevator Control System
Note: Relay is rated for max. of 30VDC at 500mA for each connection. If elevator uses different voltage, contact us.
Network Connection
Existing Button Control Lines in elevator harness
Elevator
Reader Wiegand wires added in harness for Reader in elevator
Door Reader
1
2
3
4
5
Other button connections
Elevator Button Panel

## COMPONENTS WORKING PRINCIPLE/ FUNCTIONALITY:

1. Gesture Recognition Module:  It uses infrared sensors to detect hand gestures.

2. Voice Recognition Module: It uses microphone and speech recognition algorithms.

3.LED Indicators: It display floor selection and status.

4.Microcontroller: It processes inputs and controls elevator functions.

## CODE FOR SOLUTION:

Here is a sample code in Python for gesture and voice recognition, and elevator control:

Gesture Recognition

```
import cv2

import media pipe as mp

# Initialize gesture recognition

mp_hands = mp.solutions.hands

hands = mp_hands.Hands(min_detection_confidence=0.5, min_tracking_confidence=0.5)

# Define gestures

gestures = {

    "up": [1, 0, 0, 0, 0],  # thumb up

    "down": [0, 1, 0, 0, 0],  # thumb down

    "left": [0, 0, 1, 0, 0],  # index finger left
    "right": [0, 0, 0, 1, 0],  # index finger right
}
# Capture video from camera
cap = cv2.VideoCapture(0)
while True:
    success, image = cap.read()
    if not success:
        break
    # Convert image to RGB
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    # Process image for gesture recognition
    results = hands.process(image)
    # Check for gestures
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            gesture = []
            for finger in hand_landmarks.landmark:
                gesture.append(finger.y > 0.5)
            gesture = tuple(gesture)
            if gesture in gestures.values():
                print("Gesture recognized:", list(gestures.keys())[list(gestures.values()).index(gesture)])
    # Display image
    cv2.imshow("Gesture Recognition", image)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break
cap.release()
cv2.destroyAllWindows()
```

Voice Recognition

```
import speech_recognition as sr
# Initialize voice recognition
r = sr.Recognizer()
# Define voice commands
commands = {
    "go up": "up",
    "go down": "down",
    "go to floor": "floor",
}
while True:
    with sr.Microphone() as source:
        audio = r.listen(source)
        try:
            command = r.recognize_google(audio, language="en-US")
            print("Voice command recognized:", command)
            if command in commands.keys():
                print("Command executed:", commands[command])
        except sr.UnknownValueError:
            print("Voice command not recognized")
```

Elevator Control

```
import RPi.GPIO as GPIO
# Initialize elevator control
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)  # up button
GPIO.setup(23, GPIO.OUT)  # down button
GPIO.setup(24, GPIO.OUT)  # floor select
def control_elevator(direction):
    if direction == "up":
        GPIO.output(17, GPIO.HIGH)
    elif direction == "down":
        GPIO.output(23, GPIO.HIGH)
    elif direction == "floor":
        GPIO.output(24, GPIO.HIGH)
# Integrate with gesture and voice recognition
def main():
    # Gesture recognition
    gesture_recognition_code_here
    # Voice recognition
    voice_recognition_code_here
    # Elevator control
    control_elevator(gesture_or_voice_command)
if __name__ == "__main__":
    main()
```

## PROJECT OUTCOME:

The project outcomes for a touch-free switch control panel interface using hand gestures can be truly magical:

**Seamless Elevator Interaction:**

Imagine stepping into an elevator and merely gesturing with your hand—no buttons, no physical contact.The touch-free switch control panel interprets your hand movements as commands: “Ascend,” “Descend,” or even “Pause for dramatic effect.”Elevator journeys become intuitive, efficient, and, dare I say, enchanting.

**Hygiene and Safety**:

In our pandemic-conscious world, minimizing surface contact is crucial.Touch-free interfaces reduce the risk of spreading germs and viruses.Elevator buttons, once potential hotspots, transform into touchless spells.

### User-Friendly Magic:

No prerequisite knowledge required. No deciphering cryptic symbols.Anyone—from seasoned wizards to first-time riders—can use the touch-free interface.It’s like teaching a wandless spell to Muggles.

## CONCLUSION:

Control Panel Interface represents a significant advancement in the realm of user interaction with technology. By leveraging gesture recognition and motion sensing, this innovative interface not only enhances user convenience but also contributes to a more hygienic and intuitive control experience. Its ability to minimize physical contact makes it particularly valuable in environments where cleanliness and efficiency are paramount, such as healthcare settings, public spaces, and high-tech homes.

The adoption of touch-free technology opens doors to a future where interactions are more fluid and less constrained by traditional physical interfaces. As the technology continues to evolve, we can anticipate even more sophisticated systems that will further integrate with smart environments, offering seamless and adaptable control solutions.

Overall, the Touch-Free Switch Control Panel Interface stands as a testament to the potential of emerging technologies to transform everyday tasks, providing a glimpse into a more streamlined, responsive, and user-friendly future.