**SCHOOL OF ELECTRICAL ENGINEERING**

**VIT UNIVERSITY**

**CHENNAI CAMPUS**

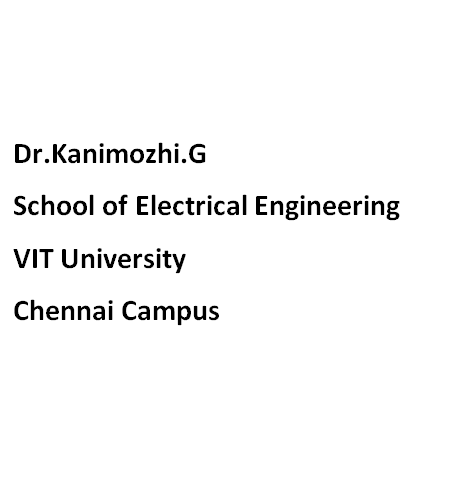
**PROJECT REPORT**

**MICROPROCESSORS AND MICROCONTROLLERS**

**(BEEE309L)**

**Home Automation using Hand Gestures for elderly people**

|  |  |
| --- | --- |
| Name | Dhanush Adithya L R- 21BEE1015 |
| Akash G - 21BEE1003 |
| Nivedha K - 21BEE1012 |
| Azhagu Meera M - 21BEE1224 |



# Objective:

The objective of implementing home automation using hand gestures for elderly people is to create a user-friendly and intuitive system that enables them to control various aspects of their home environment through simple hand movements. The system provides convenience, independence, and safety for elderly people who have limited mobility. By using hand gestures, the goal is to enhance their quality of life and promote a sense of empowerment in managing their living spaces.

**Table of Contents**

[Objective: 2](#_Toc139999380)

[List of Figures: 4](#_Toc139999381)

[Implementation Methods: 5](#_Toc139999382)

[Microcontroller Specification: 7](#_Toc139999383)

[● ATMega328P Processor 7](#_Toc139999384)

[➢ Memory 7](#_Toc139999385)

[➢ Security 7](#_Toc139999386)

[➢ Peripherals 7](#_Toc139999387)

[➢ Power 7](#_Toc139999388)

[Serial Communication: 8](#_Toc139999390)

[Python Code: 10](#_Toc139999391)

[Libraries Used: 10](#_Toc139999392)

[Code: 10](#_Toc139999393)

[Hardware Implementation: 12](#_Toc139999394)

[Conclusion: 13](#_Toc139999395)

# 

# List of Figures:

Fig 1: Hand Landmarks 5

Fig 2: Left hand recognistion 6

Fig 3: Right hand recognistion 6

Fig 4: ATMega328P Processor 7

Fig 5: Code interfacing Python and Arduino 8

Fig 6: Hardware Control code 9

Fig 7: Hardware Implementation 12

Fig 8: Hardware Implementation 12

# Implementation Methods:

For hand detection, we utilised the OpenCV library from Python’s comprehensive set of libraries. To enhance the hand tracking capabilities, we incorporated the CVZone library, along with the MediaPy, Serial, and Time libraries. Using CVZone, we accessed the webcam to capture video, which was subsequently processed through CVZone’s hand tracking module. MediaPy library was employed to identify and employed to identify and track each individual node within the fingers, enabling precise analysis and manipulation of hand movements.

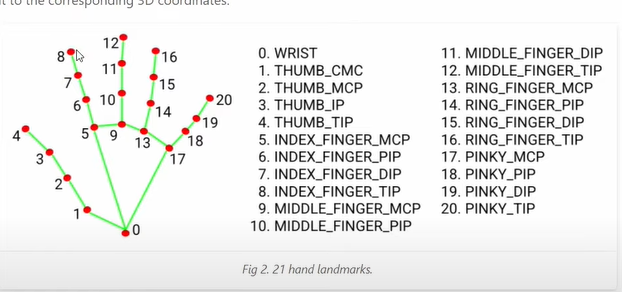


Fig 1: Hand Landmarks

The Mediapy library serves as a foundation for mapping hand nodes and projecting their respective locations as output. It facilitates the comparison of finger tip locations and finger middle point locations to determine whether a finger is open or closed. Once the number of open fingers is detected, the next step is to transmit this information to a Microcontroller for interaction with the physical world. Serial Communication is employed to establish a connection between Python and the microcontroller, enabling seamless interaction.

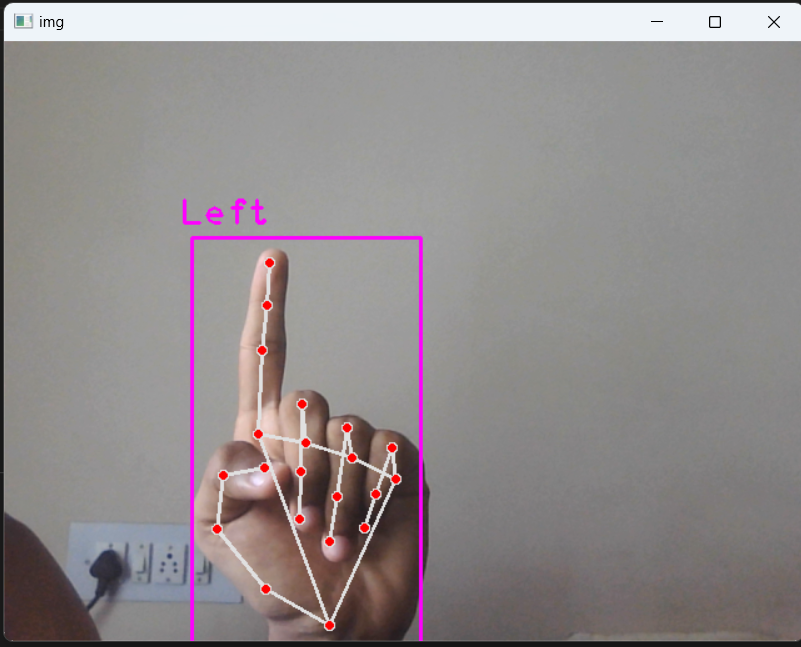
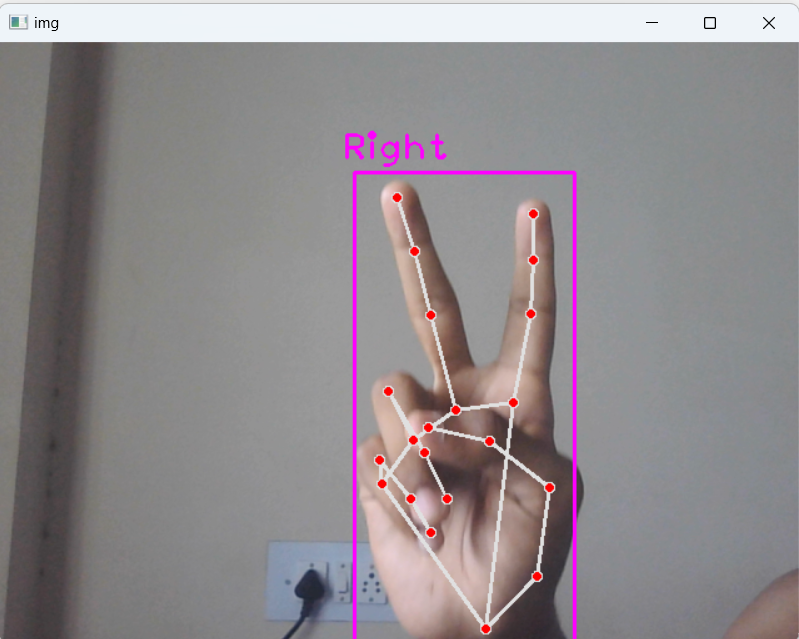
 

Fig 2: Left hand recognition

Fig 3: Right hand recognition

# 

# Microcontroller Specification:

## ATMega328P Processor

### Memory

* CLK frequency 16 MHz
* 32KB Flash
* 2KB SRAM
* 1KB EEPROM

### Security

* Power On Reset(POR)
* Brown Out Detection(BOD)

### Peripherals

* 2x 8-bit Timer/Counter with a dedicated period register and compare channels
* 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
* 1x USART with fractional baud rate generator and start-of-frame detection
* 1x controller/peripheral Serial Peripheral Interface (SPI)
* 1x Dual mode controller/peripheral I2C
* 1x Analog Comparator (AC) with a scalable reference input
* Watchdog Timer with separate on-chip oscillator
* Interrupt and wake-up on pin change

### Power

* 2.7 - 5.5 volts



## Fig 4: ATMega328P Processor

# Serial Communication:

Serial communication is utilised to facilitate data transfer from Python to an Arduino Microcontroller. The controller remains in a state of readiness, awaiting the arrival of data sent from Python through a USB cable. Upon receiving the data serially, the microcontroller proceeds to process the information and executes the programmed instructions accordingly. This mechanism enables seamless interaction between Python and the microcontroller, allowing for effective control and operation. The baud rate of the serial communication used here is 9600.

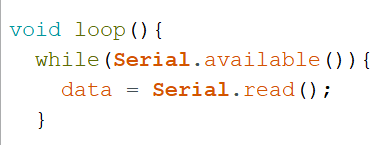


Fig 5: Code interfacing Python and Arduino

Here the void loop function will be waiting for incoming data .Once the data is received, it is then stored in a variable named “data”. This continuous loop ensures that the program remains responsive, ready to process and utilise the received information as soon as it becomes available.

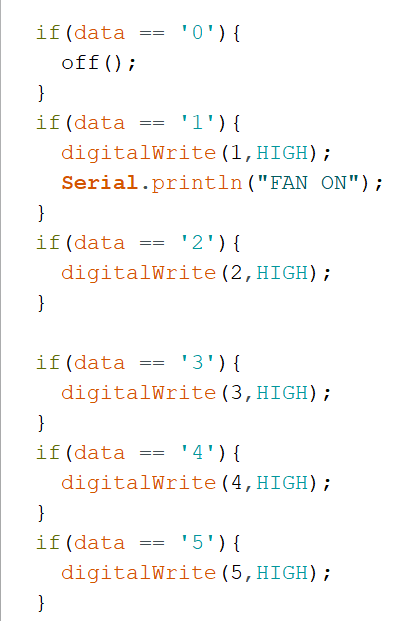


Fig 6: Hardware Control code

This program is designed to govern the relay modules that are connected to specific digital pins of the microcontroller. It provides control and regulation over the functioning of these relay modules, allowing for effective management of connected devices or systems.

# Python Code:

## Libraries Used:

* OpenCV
* CVZone
* Mediapy
* Serial
* Time

## Code:

import cv2 as cv

import time

import serial

from cvzone.HandTrackingModule import HandDetector

# for serial communication

# arduino = serial.Serial(port = 'COM7', timeout=0)

cap = cv.VideoCapture(0)

time.sleep(1) #to produce a delay of 1 second

detector = HandDetector(maxHands=1)

cap.set(3, 640) #orientation of video display

cap.set(4, 480)

while True:

isTrue, img = cap.read()

\_\_, img = detector.findHands(img)

lmlist = detector.findHands(img, draw=False)

# print(lmlist)

hnd = []

l = []

for i in lmlist:

l = i['lmList']

if (l[4][0] > l[3][0]): # for thumb

hnd.append(0)

else:

hnd.append(1)

if (l[8][1] > l[6][1]): # for index finger

hnd.append(0)

else:

hnd.append(1)

if (l[12][1] > l[10][1]): # for middle finger

hnd.append(0)

else:

hnd.append(1)

if (l[16][1] > l[14][1]): # for ring finger

hnd.append(0)

else:

hnd.append(1)

if (l[20][1] > l[14][1]): # for little finger

hnd.append(0)

else:

hnd.append(1)

print(hnd)

sum = 0

for i in hnd:

sum = sum + i

print(sum)

arduino.write(str.encode(str(sum)))

cv.imshow("img", img)

if cv.waitKey(20) & 0xFF == ord('d'): #to stop the code

break

# Hardware Implementation:

Fig 7: Hardware Implementation

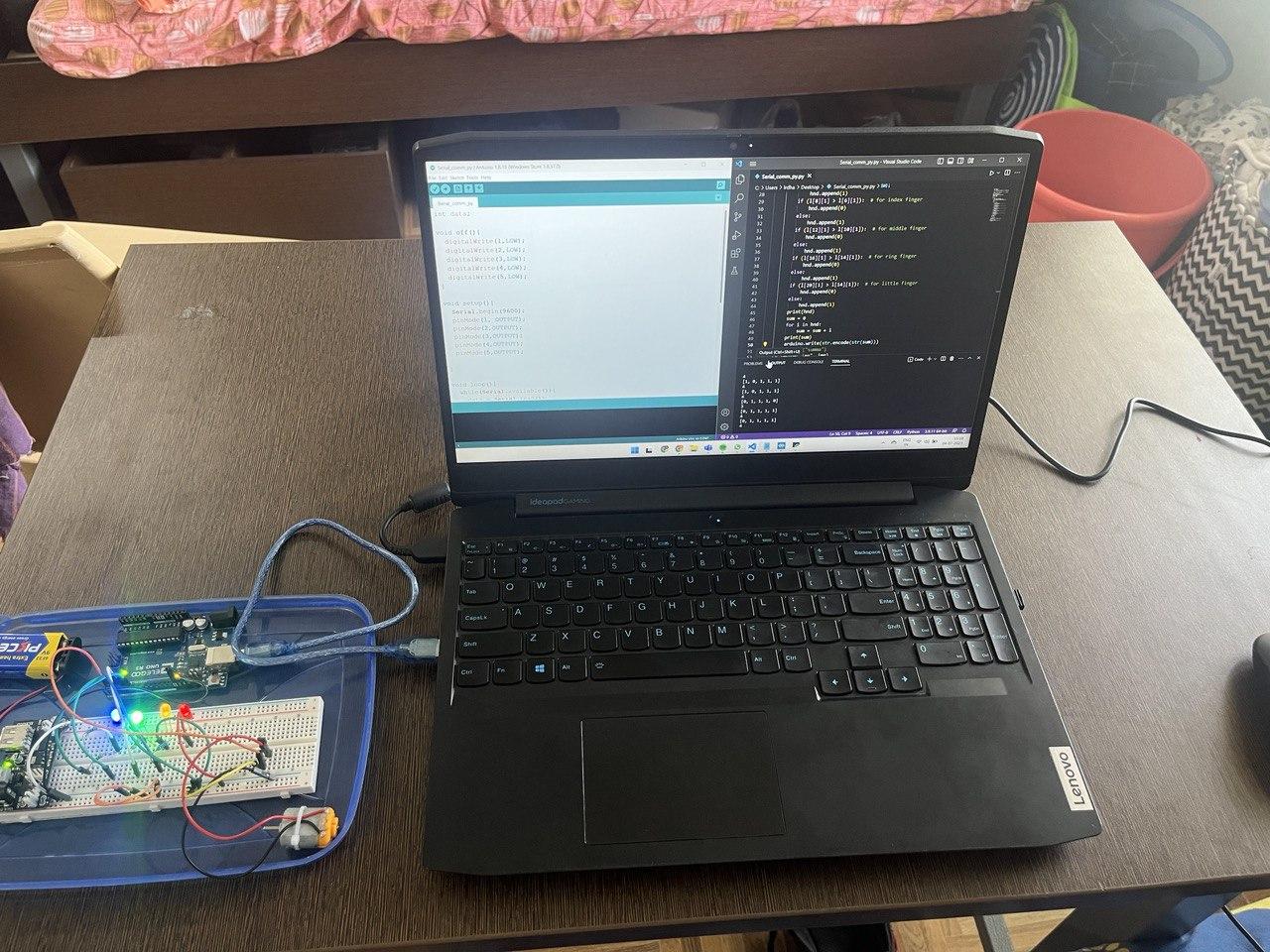


Fig 8: Hardware Implementation

# Conclusion:

Therefore, in conclusion, implementing home automation using hand gestures for elderly people offers a user-friendly and intuitive solution that empowers them to control their home environment effortlessly. This technology promotes convenience, independence, and safety by catering to the needs of individuals with limited mobility. By incorporating hand gestures as the interface, the system enhances the quality of life for the elderly, enabling them to manage their living spaces with ease and fostering a sense of empowerment in their daily lives.