A Major Project Report On

# ANIMATRONIC ROBOT HAND USING IMAGE PROCESSING

**Submitted in Fulfilment of the Requirements for the Award of Degree**

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

**Submitted By**


**PATHA SAISRUJANA (19015A0416)**

**DULAPALLI NIHAL (18011A0413)**

**KORADA MANEESH KUMAR (18011A0421)**

**MUAZ HYDER QURAISHI (18011A0462)**


**Under the esteemed guidance of**


**DR.P.CHANDRA SEKHAR REDDY**

**Professor in ECE Department**

**Department Of Electronics and Communication Engineering**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**College of Engineering Hyderabad**

Autonomous

(Kukatpally-Hyderabad-500085)


**2021-22**

**Presented to the Faculty Of**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY-COLLEGE OF ENGINEERING HYDERABAD-500085**

**In Partial Fulfilment of the Requirements for the Degree of**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

## CERTIFICATE BY THE SUPERVISOR

This is to certify that the project entitled **"ANIMATRONIC ROBOT HAND USING IMAGE PROCESSING"** being submitted by

**PATHA SAISRUJANA (19015A0416)**
**DULAPALLI NIHAL (18011A0413)**
**KORADA MANEESH KUMAR (18011A0421)**
**MUAZ HYDER QURAISHI (18011A0462)**

In partial fulfilment of the requirements for the award of degree in Bachelors of Technology in Electronics and Communication Engineering at the Jawaharlal Nehru Technological University during the academic year, 2021-22 is a bonafide work carried out under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or institute for the award of any degree or diploma.

Project Supervisor

**DR. P. CHANDRA SEKHAR REDDY**

(Professor of ECE department, JNTUHCEH)

**Presented to the Faculty Of**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY-COLLEGE OF ENGINEERING HYDERABAD-500085**

**In Partial Fulfilment of the Requirements for the Degree of**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

## CERTIFICATE BY THE HEAD OF THE DEPARTMENT

This is to certify that the project report entitled "**ANIMATRONIC ROBOT HAND USING IMAGE PROCESSING**"
Submitted by

**PATHA SAISRUJANA (19015A0416)**
**DULAPALLI NIHAL (18011A0413)**
**KORADA MANEESH KUMAR (18011A0421)**
**MUAZ HYDER QURAISHI (18011A0462)**

In partial fulfilment of the requirements for the award of degree in **Bachelor of Technology in Electronics and Communication Engineering** at the Jawaharlal Nehru Technological University during the academic year, 2021-22 is a bonafide work carried out by them under guidance and supervision. The results are verified and found satisfactory. The result embodied in this project report has not been submitted to any other University or Institute for the award of any Degree or Diploma.

HEAD OF THE DEPARTMENT
**DR. A RAJANI**

(Professor of ECE department, JNTUHCEH)

**Presented to the Faculty Of**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY-COLLEGE OF ENGINEERING HYDERABAD-500085**

**In Partial Fulfilment of the Requirements for the Degree of**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

# DECLARATION OF THE CANDIDATES

We Hereby Declare The Major Project Entitled "**ANIMATRONIC ROBOT HAND USING IMAGE PROCESSING**" is a bonafide record work done and submitted under the esteemed guidance of **Dr. P CHANDRA SEKHAR REDDY**, Professor of ECE department, JNTU, Hyderabad , in partial fulfillment of the requirements for the award of the degree of **Bachelor Of Technology (Regular)** in **Electronics and Communication Engineering** during the academic year 2021-2022.This record is a bonafide work carried out by us and the results kept in the major project have not been reproduced. The results in the project have not been submitted in any other university for the award of degree or diploma.

By:

**PATHA SAISRUJANA (19015A0416)**

**DULAPALLI NIHAL (18011A0413)**

**MANEESH KUMAR (18011A0421)**

**MUAZ HYDER QURAISHI (18011A0462)**

## ACKNOWLEDGEMENT

This project titled "**ANIMATRONIC ROBOT HAND USING IMAGE PROCESSING**" was carried out by us. We are grateful to **Dr P. CHANDRA SEKHAR REDDY,** Professor and **Dr A RAJANI**, Professor and Head of the Department of Electronics and Communication Engineering, JNTUH College of Engineering, for their guidance while pursuing this project.

We take this precious opportunity to acknowledge our internal project guide

**Dr P. CHANDRA SEKHAR REDDY**, Professor of Electronics and Communication Engineering, JNTUH College of Engineering, Hyderabad for his timely advice, effective guidance and encouragement throughout the completion of our mini-project work.

We also owe deep respect of gratitude to our parents and friends for their cheerful encouragement and valuable suggestions, without whom this work would not have been completed in this stipulated time.

We would like to articulate our heartfelt gratitude to the authorities of JNTU for their help throughout our project work. A few lines of acknowledgement do not fully express our gratitude and appreciation for all those who guided and supported us throughout this project. Last but not the least, we acknowledge the help received from many journals and websites.

Finally, we thank one and all who helped us directly or indirectly throughout our project work.

By:

**PATHA SAISRUJANA (19015A0416)**

**DULAPALLI NIHAL (18011A0413)**

**MANEESH KUMAR (18011A0421)**

**MUAZ HYDER QURAISHI (18011A0462)**

# ABSTRACT

A robot hand, especially one with multiple fingers, is a kinematics chain of rigid links interconnected by movable joints, necessary for conducting various tasks in daily life. Realization of the robot hand-making flexible human-like motions is the design inspired by the human musculoskeletal system. It is not only used as a hand of a humanoid robot but also used as a prosthetic hand. However, constructing a model and applying a conventional controller to it is difficult, due to its complicated structure. Therefore, one of the obstacles in developing a robot hand is designing a controller. In the present study, a bioinspired controller is used to convey information or interact with the environment to control a five-fingered robot hand with a musculoskeletal system.

Developing and training a model from scratch works better but typically requires a large amount of data to produce accurate results.

To overcome this problem we use the Mediapipe package. In this package, pre-trained models such as HANDS is used as a starting point.

In this project we detect hand and send that data to the Bluetooth then we prepare a Hardware model of Hand with Arduino which takes input from the Bluetooth module and provides a real-time working Hand model.

# CONTENTS

# LIST OF FIGURES

# INTRODUCTION

Animatronics is a hybrid of animation and electronics. It can be pre-customized (programming) or remotely controlled. It is the utilization of link pulled gadgets or actuators to quicken a reproduction of a human or a creature or carry similar attributes, in any case, lifeless thing.

Robot hands are intended to realize the same dexterous and versatile manipulation that we humans can do. Thus, for robot-hand research, understanding the anatomy and motion of the human hand is fundamental. On the other hand, from the point of view of human-hand research, describing the mechanisms and mechanics behind hand posture and motion helps to understand how we realize such dexterous and versatile manipulations. However, there is a wide gap between robotic and human informatics, and it is difficult to interchange the diverse knowledge accumulated in each research field directly.

We have been involved in digital-hand research for more than a decade. The goal of this project is to simulate the animatronic hand using Image processing for promoting ergonomic product design, taking the individual differences in hand properties into account. Interestingly, this robot hand is applicable to not only ergonomic product design as previously described but to interface the flexible human-like motions. For example, a simple animatronic model with image processing libraries enabled us to reproduce the motion of the human hand.
The 5 finger movements chosen for this project were:
1)Thumb open 2) Index finger open 3) Middle finger open 4) Ring finger open 5) Pinky finger
Open and many combinations.

In this project, we used OpenCV for hand recognition. OpenCV is free for use, crossplatform library of programming functions that mainly work on real-time computer vision.

Along with OpenCV, we use The Proteus Design Suite which is a proprietary software tool suite used primarily for electronic design simulation where we simulated Arduino UNO which is an open-source microcontroller board based on an 8-bit ATMega328P microcontroller with an operating voltage of 5V and clock speed of 16MHz. It has 6 analog I/O pins and 14 digital I/O pins and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB Cable.

PWM Servo Motor is simulated to represent each finger in proteus. The PWM sent to the motor determines the position of the shaft, and based on the duration of the pulse sent via the control wire; the rotor will turn to the desired position which represents the status of the finger.

# BACKGROUND

The hand is one of the most complex and beautiful pieces of natural engineering in the human body. It gives us a powerful grip but also allows us to manipulate small objects with great precision. This versatility sets us apart from every other creature on the planet. The hand has one of the strangest arrangements of muscles in the body. Most of its movements are controlled by muscles that aren't located in the hand at all but in the forearm. The human opposable thumb is longer, compared to finger length than any other primate thumb. This long thumb and its ability to easily touch the other fingers allow humans to firmly grasp and manipulate objects of many different shapes.

The gestures produced by the motion of the human hand are undoubtedly natural. They may often prove more efficient and powerful as compared to various modes of interaction. The gestures are communicative, meaningful body motions i.e, physical movement of fingers, hands, arms or body to convey information or interact with the environment. To increase the use of robots where conditions are not certain such as rescue operations we can make robots that follow the instruction of a human operator and perform the task, in this way decisions are taken according to the working condition by the function of the task is performed by robots. In this project we will mainly develop an animatronic robot hand that can work faster and more accurate than any human worker and improves the production capacity, in addition, they do not require any breaks and can work more intelligent.

Furthermore, the scope of the project expands into Bionic Prosthetic.
185,0000 people have an amputation each year and around 6 Million people are limbless. A bionic arm is an electromechanical device that attaches to the human body and replicates the functionality of a natural arm/hand. It always consists of a bionic hand depending on the level of amputation, may also include a powered wrist, elbow, and/or shoulder.

# THEORETICAL INFRASTRUCTURE

## Arduino UNO Microcontroller

Arduino UNO is an open-source microcontroller word based on the microchip ATmega328P microcontroller and developed by Arduino cc. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The microcontroller was introduced in the electronics industry with the intention of making tasks easy. It is based on arduino's past wiring and processing projects. Processing is written for nonprogramming users. Arduino wiring is produced on the basis of programming language.

*Figure 1.Arduino UNO*

Arduino can be used to communicate with a computer, another Arduino board or other microcontrollers. The ATmega328P microcontroller provides UART TTL (5V) serial communication which can be done using digital pin 0 (Rx) and digital pin 1 (Tx). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The ATmega16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. There are two RX and TX LEDs on the arduino board which will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328P also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

## TECHNICAL SPECIFICATIONS:

This table describes the technical specifications of the Arduino used in the project.

| | |
|---|---|
| Microcontroller | ATmega328P |
| Operating voltage | 5V |
| Input voltage (recommended) | 7-12V |
| Input voltage (limit) | 6-30V |
| Digital I/O pins | 14(of which 6 provide PWM output) |
| PWM digital I/o pins | 6 |
| Analog input pins | 6 |
| DC current per I/o pins | 20mA |
| DC current for 3.3V pin | 50Ma |
| Flash memory | 32KB |
| SRAM | 2KB |
| EEPROM | 1KB |
| Clock speed | 16MHz |
| LED_BUILTIN | 13 |
| Length | 68.6mm |
| Width | 53.4mm |
| weight | 25g |

## Servo Motors

It is a type of motor that can rotate an object at some specific angles. It is made up of simple motor which runs through a servo mechanism. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, robotics, etc.

A DC motor along with servomechanism (closed-loop control system) acts as a servo motor which is basically used as a mechanical transducer in the automation industry. Based on its accurate closed-loop control, it has versatile applications used in many industries. The **DC servo motor definition** is, a motor that is used in servo systems is known as a servo motor. A servo system is a closed-loop system where the feedback signal (position, velocity, acceleration, etc.) drives the motor. This signal acts as an error and based on controller, accurate position or velocity is achieved. The motors are coupled to an output shaft (load) through a gear train for power matching. Servo motor acts as a mechanical transducer as they convert an electrical signal to an angular velocity or position.


*Figure 2.DC Servo Motor*

## Specifications of SG90 Servo

| | |
|---|---|
| Modulation | Analog |
| Torque | **4.8V:** 25.00 oz-in (1.80 kg-cm) |
| Speed | **4.8V:** 0.12 sec/60° |
| Weight | 0.32 oz (9.0 g) |
| Dimensions | **Length:**0.91 in (23.0 mm)<br>**Width:**0.48 in (12.2 mm)<br>**Height:**1.14 in (29.0 mm) |

## HC-05 Wireless Bluetooth Serial Transceiver

This is a module which can add two-way wireless functionality to your projects. It is used to communicate between microcontrollers or communicate with any device with Bluetooth functionality. This is considered as the perfect choice if you are looking for a wireless module that could transfer data from your computer or mobile phone to a microcontroller or vice versa.

We need wireless communication in this project, so we will use Bluetooth technology and for that module that will be used is HC-05. This module has several programmable baud rates but the default baud rate is 9600 bps. It can be configured as either master or slave, whereas another module HC-06 can work only in slave mode. This module has four pins. One for VCC (5V) and the remaining three for GND, TX, and RX. The default password of this module is **1234** or **0000**. If we want to communicate between two microcontrollers or communicate with any device with Bluetooth functionality like a Phone or Laptop HC-05 helps us to do that. Several Android applications are already available which makes this process a lot easier.

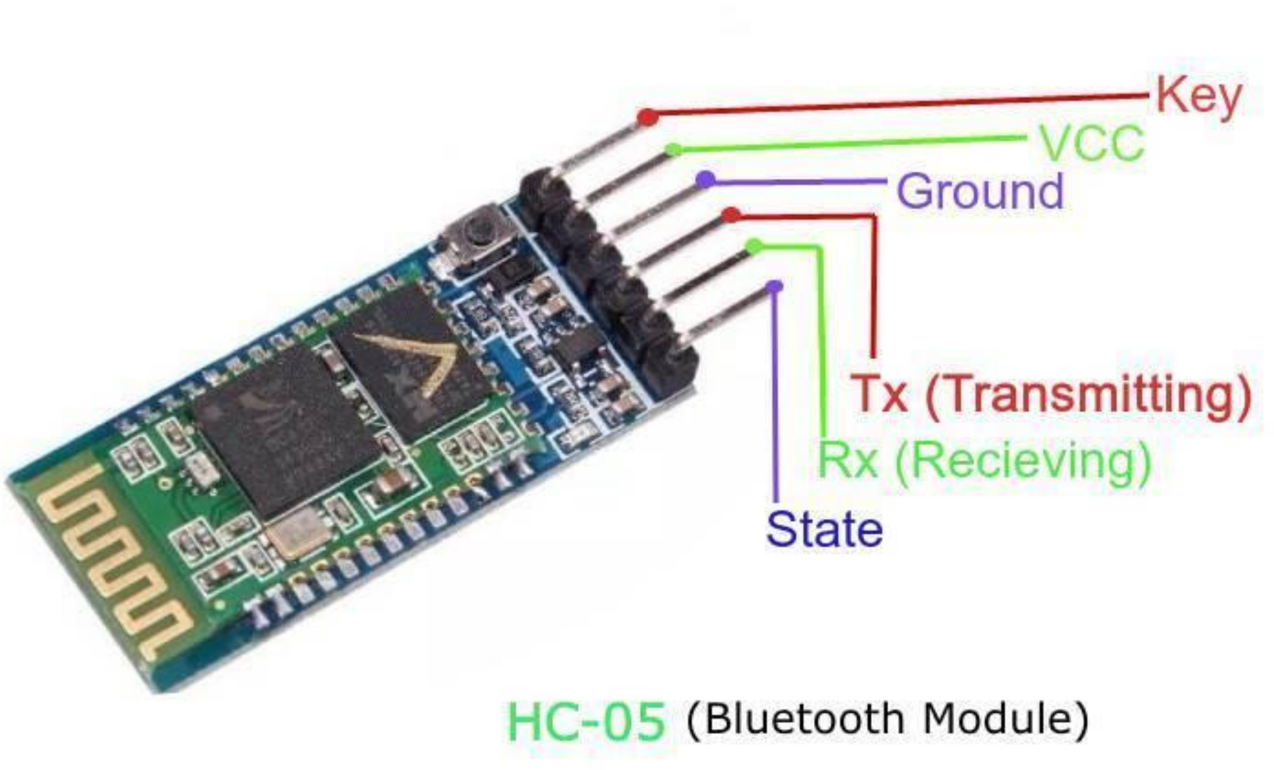
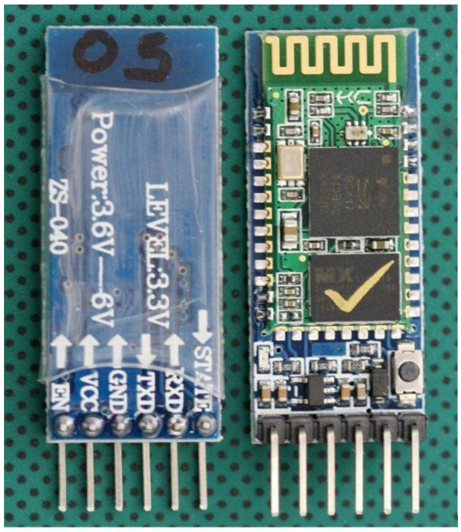



*Figure 3.HC-05 Bluetooth Transceiver Pin Description*




HC-05

*Figure 4.HC-05 Bluetooth Transceiver IC*

<u>HC-05 Technical Specifications</u>

- Serial Bluetooth module for Arduino and other microcontrollers
- Operating Voltage: 4V to 6V (Typically +5V)
- Operating Current: 30mA
- Range: <100m
- Works with Serial communication (USART) and TTL compatible
- Follows IEEE 802.15.1 standardized protocol

- Uses Frequency-Hopping Spread spectrum (FHSS)
- Can operate in Master, Slave or Master/Slave mode
- Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

## Power Supply

A power supply takes the AC from the AC source then produces a DC output of 2A and 5V. The power supply chosen to supply the servomotors control circuit is capable of supplying the same current even though all though all the synchronous servomotors are working.

## Circuit Diagram

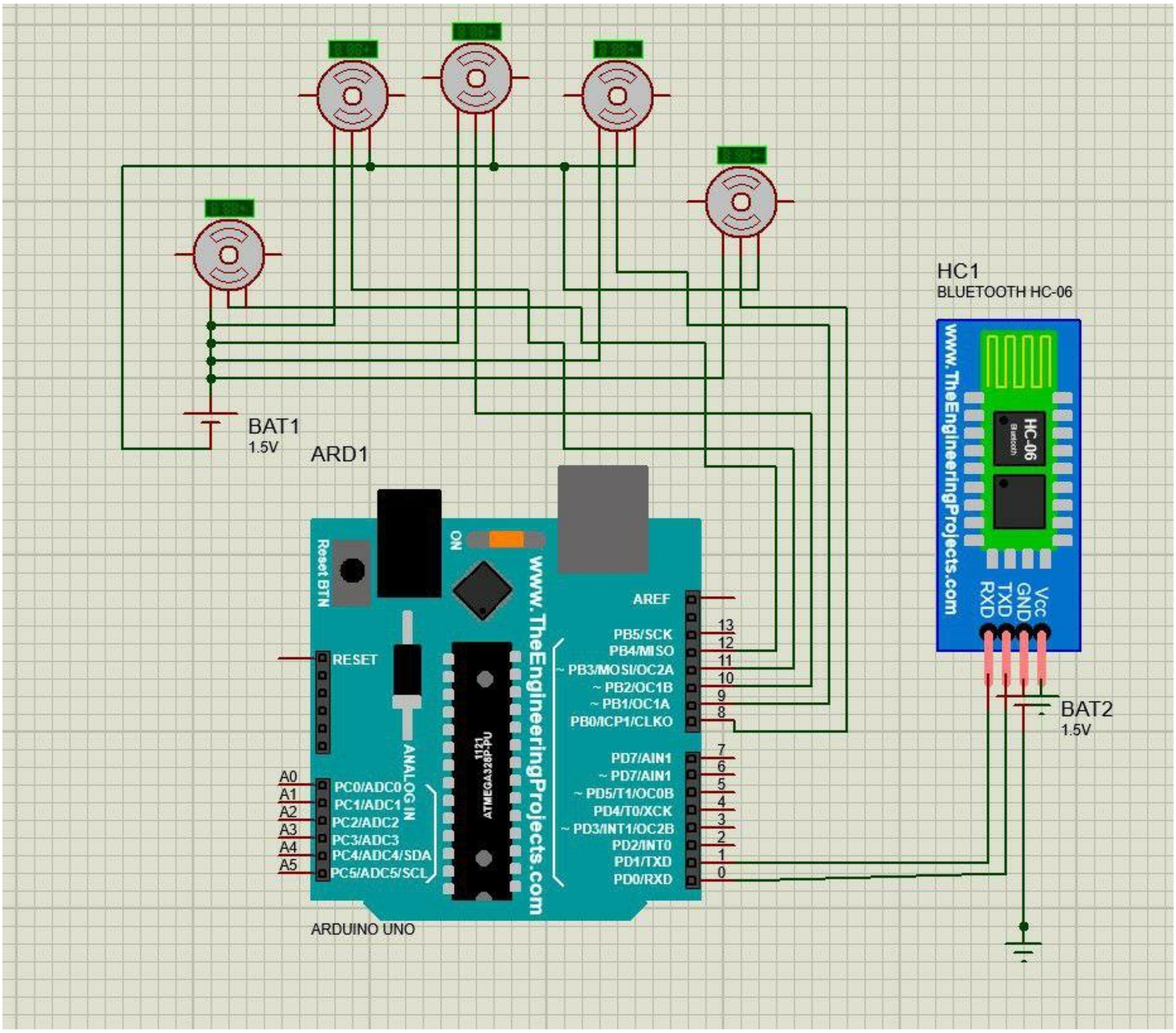



*Figure 5.Schematic Diagram*

# SOFTWARE USED

## Arduino

Arduino is an open-source platform used for constructing and programming electronics. It can receive and send information to most devices. it uses hardware called Arduino circuit boards and software programme (Simplified C++) to programme the board.


*Figure 6.Arduino IDE*

The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'

## OpenCV Mediapipe

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Here we specifically use the Mediapipe library built on OpenCV and other ML pipelines.

MediaPipe is a framework for building multimodal, cross-platform applied ML pipelines. With MediaPipe, a perception pipeline can be built as a graph of modular components, including, for instance, inference models (e.g., TensorFlow, TFLite) and media processing functions.

MediaPipe Hands utilizes an ML pipeline consisting of multiple models working together: A palm detection model that operates on the full image and returns an oriented hand bounding box. A hand landmark model that operates on the cropped image region defined by the palm detector and returns high-fidelity 3D hand keypoints.

The pipeline is implemented as a MediaPipe graph that uses a hand landmark tracking subgraph from the hand landmark module and renders using a dedicated hand renderer subgraph. The hand landmark tracking subgraph internally uses a hand landmark subgraph from the same module and a palm detection subgraph from the palm detection module.

# SYSTEM FLOW

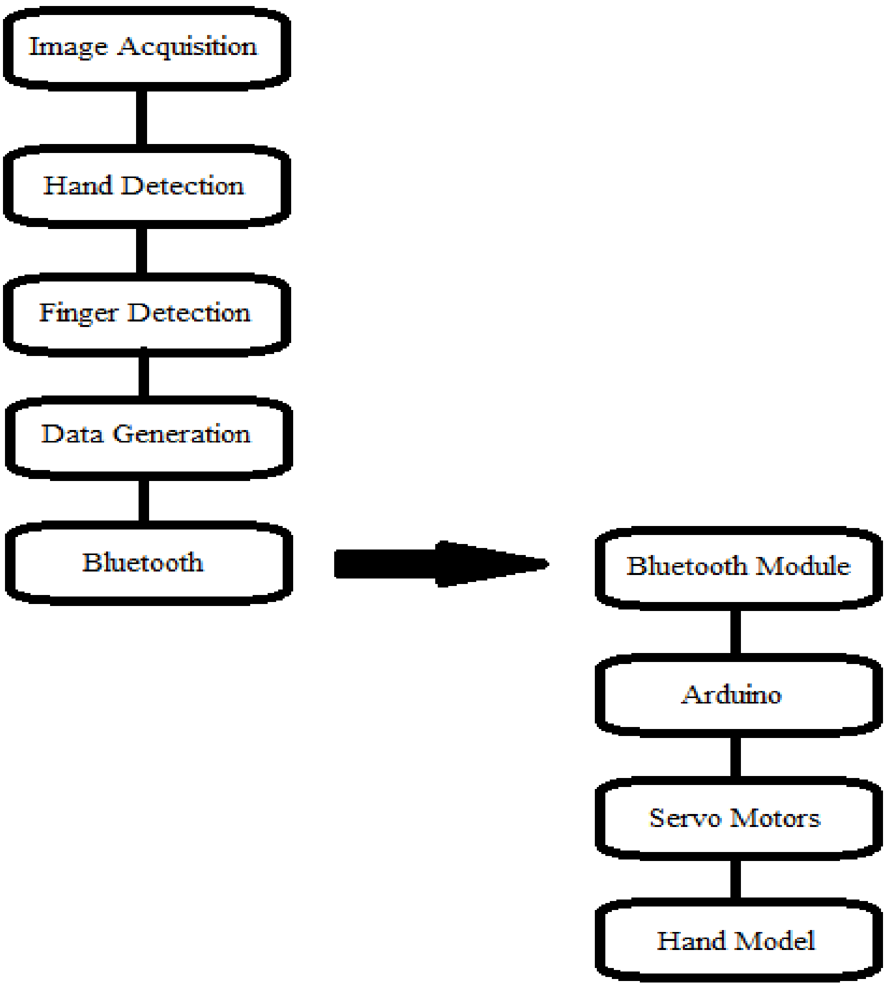



*Figure 7.System Flow Model*

The system generates data from detecting hands and the data is passed to the simulated hand in proteus.

Initially for hand tracking Mediapipe module is used.

Hand tracking module utilizes ML pipeline consisting of two models

- Palm detector
- Hand landmark

Palm detector will provide accurately cropped image to the hand landmark model.

Hand landmark model will try to locate the 21 key hand points.

This drastically reduce data augmentation. (cropping, scale etc.. )

## Palm detector

To detect initial hand locations, we employ a single shot detector (SSD) model for real-time application.

The core of SSD is predicting category scores and box offsets for a fixed set of

default bounding boxes using small convolutional filters applied to feature maps.

SSD has two components: a **backbone** model and **SSD head**. *Backbone* model usually is a pre-trained image classification network as a feature extractor.

The backbone model is based on a standard architecture used for high quality image classification, the SSD head is where the hand is detected.

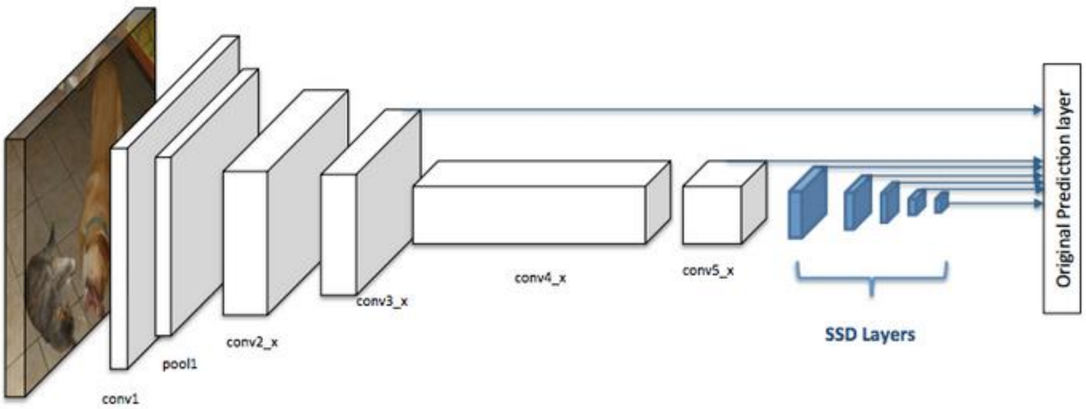



*Figure 8.VGG Network Model*

The backbone uses a typical ResNet structure (VGG) based on ImageNet generating feature maps.

These feature maps are further passed to the convolutional filters in the head.

These filters will reduce size progressively and outputs the union of multiple layer bounding box.

Mediapipe uses only square bounding boxes reducing computation (reducing anchor boxes).

# Hand Landmarking

After running palm detection over the whole image, our subsequent hand landmark model performs precise landmark localization of 21 coordinates inside the detected hand regions via regression.

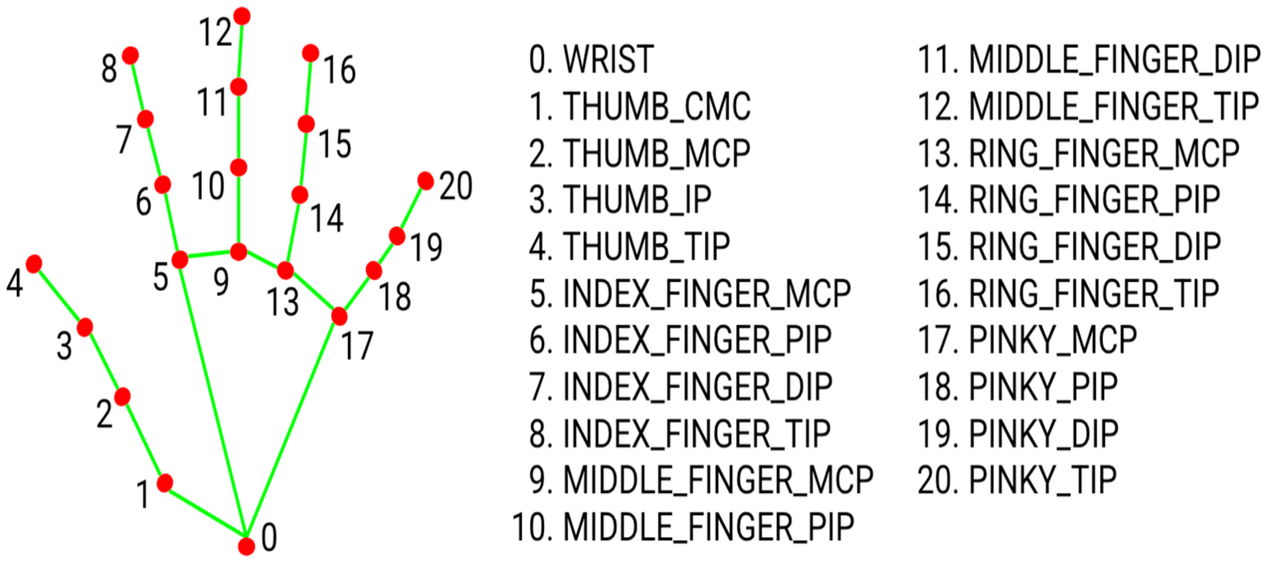



*Figure 9.Hand Landmarking Model*

Handmarking model uses Multi view bootstrapping to generate key points even for occluded images

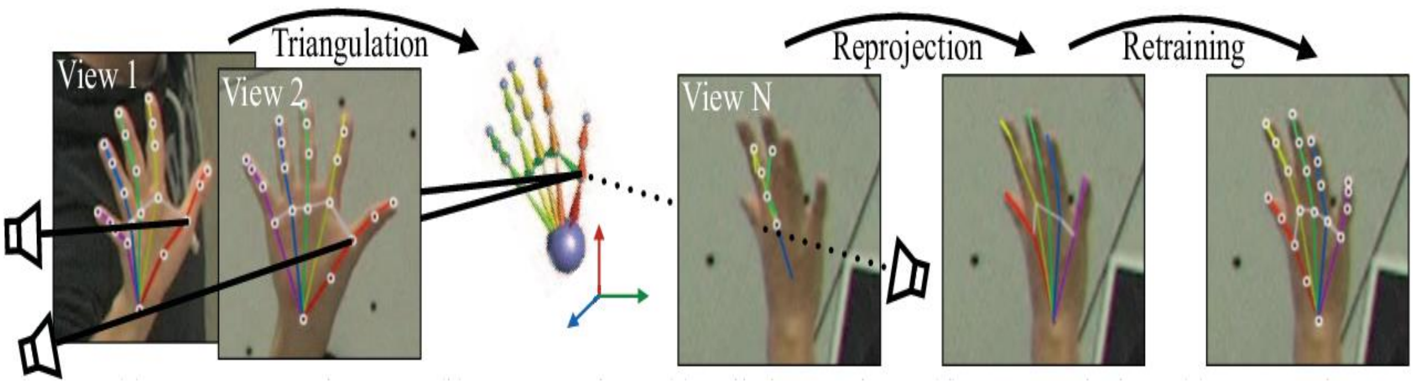



*Figure 10.Bootstrapping Model*

The points are detected using typical detector, then further of our hand moving a map is generated through triangulation.

If any prediction has low confidence value (fails) the previously projected map is used to reproject and this projection is retrained for further predictions.

This technique will also detect self occluded hands.

# DETECTOR FLOW

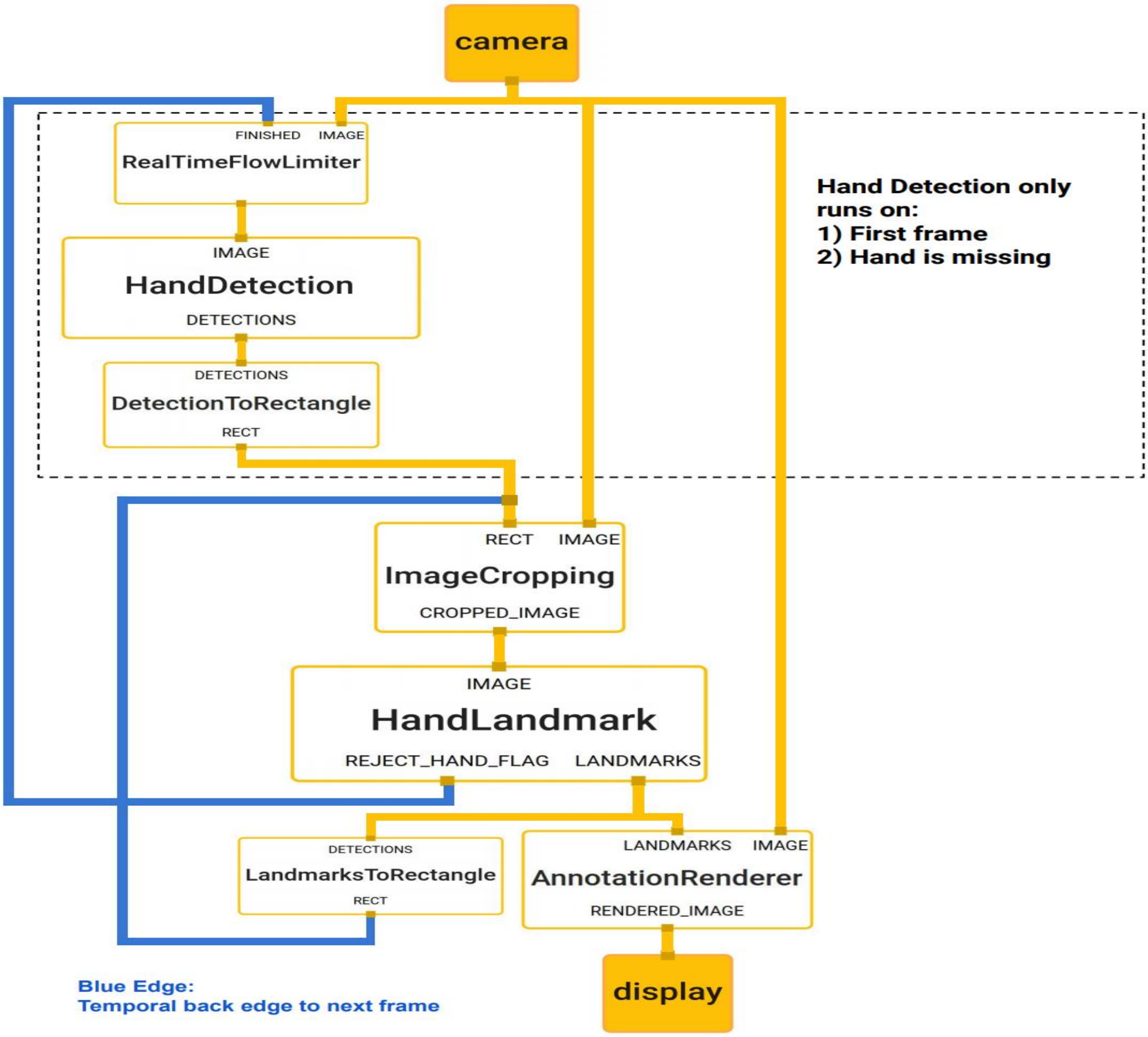



*Figure 11.Detector Flow Model*

The coordinates of the hand key points are obtained, through distance calculations we can generate hand data according to its state.

To pass this hand data to the animatronic hand physically we use output communication port of Bluetooth module.

Arduino gets data from Bluetooth transceiver which makes the motors respond accordingly with the gestures given at the input.

# EXPERIMENTAL ANALYSIS

Create a module naming cvzone using OpenCV and Mediapipe

```
import cv2
import mediapipe as mp
import math


class HandDetector:

    def __init__(self, mode=False,maxHands=2, detectionCon=0.5, minTrackCon=0.5):
        self.mode = mode
        self.maxHands = maxHands
        self.detectionCon = detectionCon
        self.minTrackCon = minTrackCon

        self.mpHands = mp.solutions.hands
        self.hands = self.mpHands.Hands(static_image_mode=self.mode,
max_num_hands=self.maxHands,min_detection_confidence=self.detectionCon,
min_tracking_confidence = self.minTrackCon)
        self.mpDraw = mp.solutions.drawing_utils
        self.tipIds = [4, 8, 12, 16, 20]
        self.fingers = []
        self.lmList = []

    def findHands(self, img, draw=True, flipType=True):
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.hands.process(imgRGB)
        allHands = []
        h, w, c = img.shape
        if  self.results.multi_hand_landmarks:
            for handType,handLms in
zip(self.results.multi_handedness,self.results.multi_hand_landmarks):
                myHand={}
                ## lmList
                mylmList = []
                xList = []
                yList = []
                for id, lm in enumerate(handLms.landmark):
                    px, py = int(lm.x * w), int(lm.y * h)
                    mylmList.append([px, py])
                    xList.append(px)
                    yList.append(py)
                xmin, xmax = min(xList), max(xList)
```

```
                ymin, ymax = min(yList), max(yList)
                boxW, boxH = xmax - xmin, ymax - ymin
                bbox = xmin, ymin, boxW, boxH
                cx, cy = bbox[0] + (bbox[2] // 2), \
                         bbox[1] + (bbox[3] // 2)

                myHand["lmList"] = mylmList
                myHand["bbox"] = bbox
                myHand["center"] =  (cx, cy)

                if flipType:
                    if handType.classification[0].label =="Right":
                        myHand["type"] = "Left"
                    else:
                        myHand["type"] = "Right"
                else:myHand["type"] = handType.classification[0].label
                allHands.append(myHand)

                ## draw
                if draw:
                    self.mpDraw.draw_landmarks(img, handLms,self.mpHands.HAND_CONNECTIONS)
                    cv2.rectangle(img, (bbox[0] - 20, bbox[1] - 20),(bbox[0] + bbox[2] +
20, bbox[1] + bbox[3] + 20),(255, 0, 255), 2)
                    cv2.putText(img,myHand["type"],(bbox[0] - 30, bbox[1] -
30),cv2.FONT_HERSHEY_PLAIN,2,(255, 0, 255),2)
        if draw:
            return allHands,img
        else:
            return allHands

    def fingersUp(self,myHand):
        myHandType =myHand["type"]
        myLmList = myHand["lmList"]
        if self.results.multi_hand_landmarks:
            fingers = []
            # Thumb
            if myHandType == "Right":
                if myLmList[self.tipIds[0]][0] > myLmList[self.tipIds[0] - 1][0]:
                    fingers.append(1)
                else:
                    fingers.append(0)
            else:
                if myLmList[self.tipIds[0]][0] < myLmList[self.tipIds[0] - 1][0]:
                    fingers.append(1)
                else:
                    fingers.append(0)
            # 4 Fingers
            for id in range(1, 5):
                if myLmList[self.tipIds[id]][1] < myLmList[self.tipIds[id] - 2][1]:
                    fingers.append(1)
                else:
```

```
                fingers.append(0)
        return fingers
```

We can install cvzone package to get the module in prior.

## Create the main file and import the above module

```
import cv2
import cvzone.SerialModule
from cvzone.HandTrackingModule import HandDetector
import serial
cap = cv2.VideoCapture(0)
cap.set(3,1280)
cap.set(4,720)

detect = HandDetector(detectionCon=0.8,maxHands=1)
ms=cvzone.SerialModule.SerialObject("COM2",9600,1)

while True:
    stat,img = cap.read()
    img = cv2.flip(img, 1)
    hand,img = detect.findHands(img,flipType=False)
    if hand:
        h=hand[0]
        f = detect.fingersUp(h)
        if f[0]==1:
            f[0]=0
        else:
            f[0]=1
        ans=""
        for i in f:
            ans+=str(i)
        print(ans)
        ms.sendData(ans)
    cv2.imshow("image",img)
    cv2.waitKey(1)
```

## Record Code

```
import cv2
import cvzone.SerialModule
from cvzone.HandTrackingModule import HandDetector
import serial
```

```
fi = open('record.txt', 'x')


cap = cv2.VideoCapture(0)
cap.set(3,1280)
cap.set(4,720)

detect = HandDetector(detectionCon=0.8,maxHands=1)
ms=cvzone.SerialModule.SerialObject("COM6",9600,1)

while True:
    stat,img = cap.read()
    img = cv2.flip(img, 1)
    hand,img = detect.findHands(img,flipType=False)
    if hand:
        h=hand[0]
        f = detect.fingersUp(h)
        if f[0]==1:
            f[0]=0
        else:
            f[0]=1
        ans=""
        for i in f:
            ans+=str(i)
        print(ans)
        fi.write(ans+"\n")
    cv2.imshow("image",img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

A File named record.txt will be created. Content being the data generated. Only the data will be recorded but not the frames of the given input.

## Playback Code

```
import time
import serial
import cvzone.SerialModule
import cv2

ms=cvzone.SerialModule.SerialObject("COM6",9600,1)

with open('readme.txt', 'r') as f:
    for i in range(100):
        s=f.readline()
        print(s)
        ms.sendData(s)
        time.sleep(0.5)
```

# Arduino Code

```
#include <Servo.h>

#define n 5
#define digits 1

Servo h1;
Servo h2;
Servo h3;
Servo h4;
Servo h5;

int valrec[n];
int slen=n*digits+1;
int count=0;
bool c=false;

String rec;

void rd(){
  while(Serial.available()){
    char c=Serial.read();
    if(c=='$'){
      c=true;
    }
    if(c){
      if(count<slen){
        rec = String(rec+c);
        count++;
      }
      if(count>=slen){
        for(int i=0;i<n;i++){
          int num=(i*digits)+1;
        valrec[i]=rec.substring(num,num+digits).toInt();
        }
        rec="";
        count=0;
        c=false;
      }
    }
  }
}

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  h1.attach(10);
  h2.attach(9);
  h3.attach(8);
  h4.attach(7);
  h5.attach(6);
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  rd();
if(valrec[0]==1){h1.write(180);}else{h1.write(0);}
if(valrec[1]==1){h2.write(180);}else{h2.write(0);}
if(valrec[2]==1){h3.write(180);}else{h3.write(0);}
if(valrec[3]==1){h4.write(180);}else{h4.write(0);}
if(valrec[4]==1){h5.write(180);}else{h5.write(0);}
}
```

# SETUP

First connect PC to the Bluetooth module HC-05, Go to Bluetooth settings to add new device. Search for HC-05.

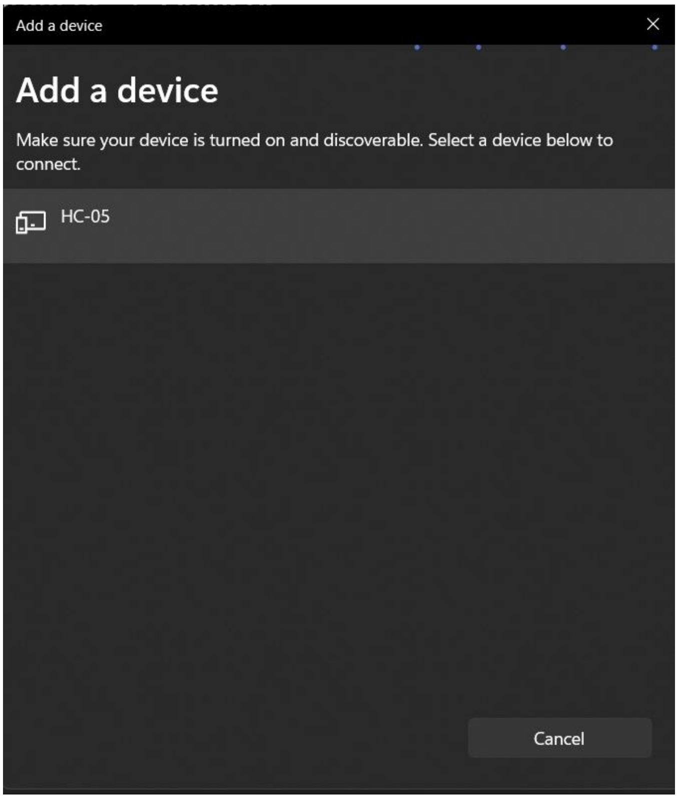



*Figure 12. add device*

Connect PC to HC-05 by selecting. Then we need to provide password. The default password will either be 0000 or 1234.

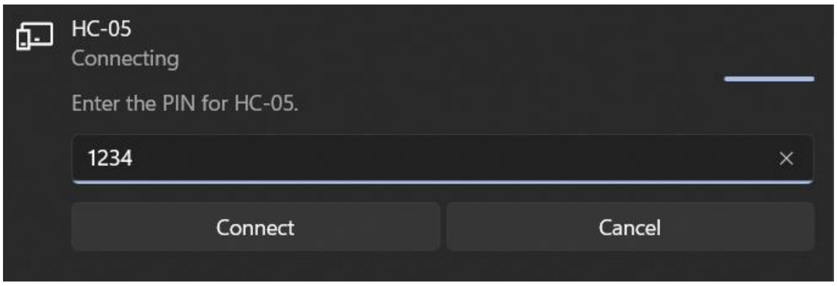
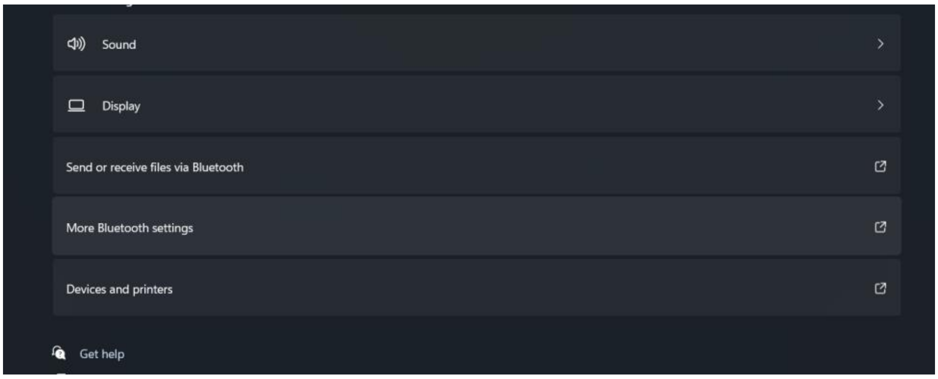



*Figure 13. Password*

Once our PC is connected to Bluetooth module we need to find the COM port assigned to it. Go to More Bluetooth settings and COM ports. HC-05 Serial port will be visible. The port which is outgoing should be used.



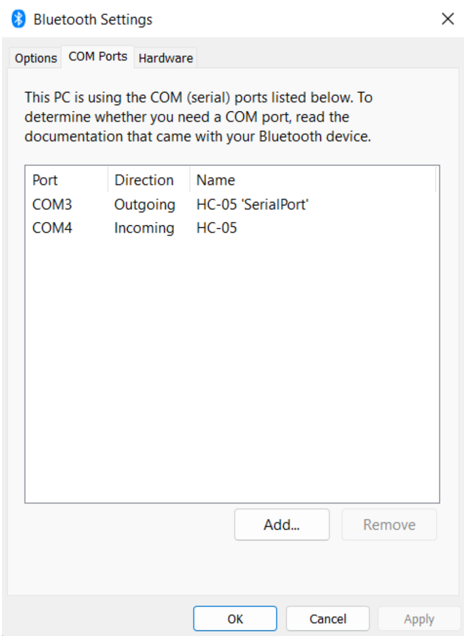




*Figure 14. COM ports*

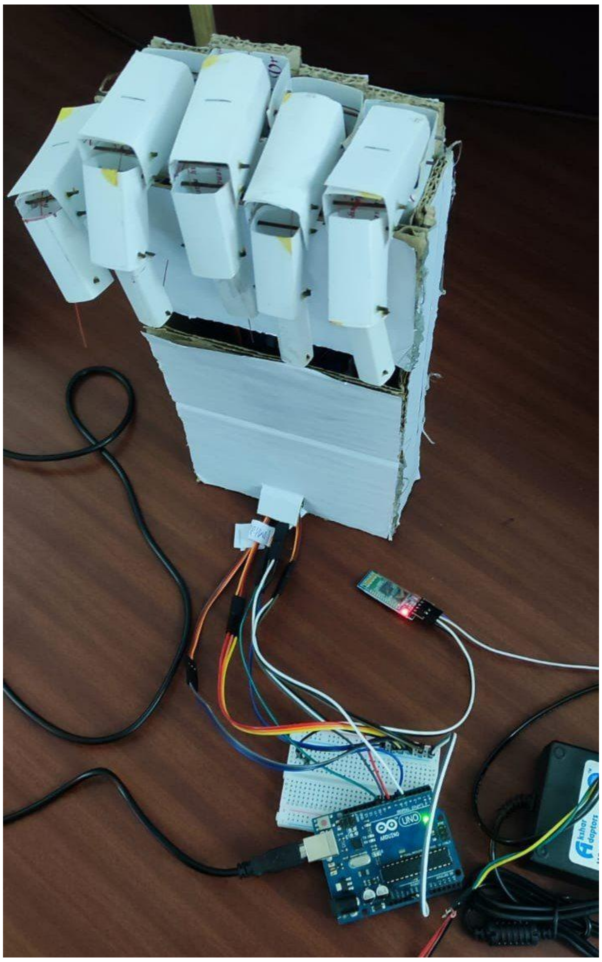
The outgoing COM port should be mentioned at the line 9 inside SerialObject and the Baud rate should be 9600 which is ideal for Arduino.

```
4   #import serial
5   cap = cv2.VideoCapture(0)
6   cap.set(3,1280)
7   cap.set(4,720)
8   detect = HandDetector(detectionCon=0.8,maxHands=1)
9   ms=cvzone.SerialModule.SerialObject("COM3",9600,1)
10  while True:
11      stat,img = cap. (variable) img: Any
12      img = cv2.flip(img, 1)
13      hand,img = detect.findHands(img,flipType=False)
14      if hand:
```

The blinking frequency will change if the Bluetooth module is connected to the PC.

## RESULT AND DISCUSSION

A research on possible designs and basic information about the robotic arm was initially made. Using 5 servo motors, the robot Hand could move in different directions. For the optimal control of the robot Hand, the microcontroller Arduino UNO is used. This microcontroller was preferably used since it is low cost and good for beginners especially in writing the program. The program code was written in c+ + language which is one of the most popular and fundamental programming languages. The Bluetooth module will be the medium of communication between the robot Hand and the PC.



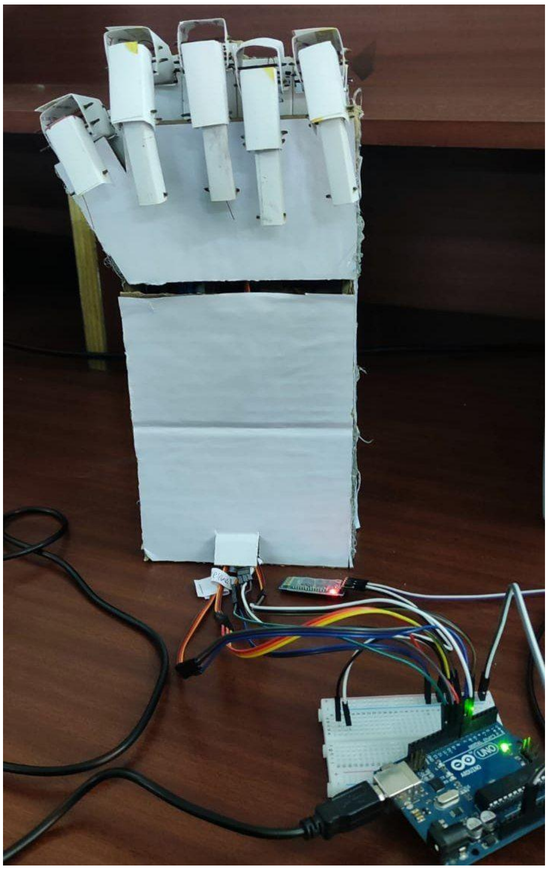


*Figure 15. Physical Model*

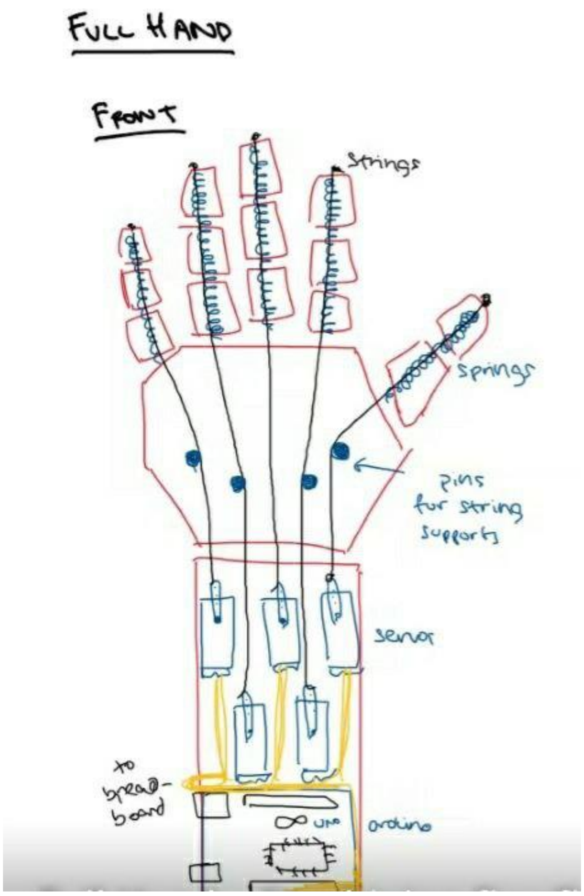
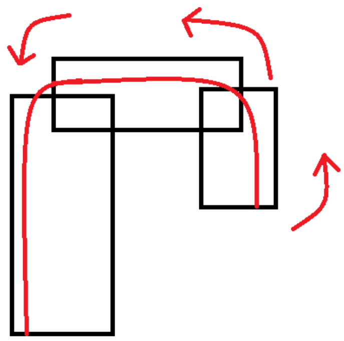
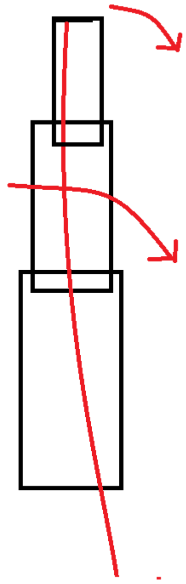



*Figure 16. Mechanical Design*

The mechanics in the hand is that each finger has a dedicated servo motor and a string attached to the very tip of the finger.



By default the fingers will be closed. A string attached to each finger at one end at the other end attached to the servo motor which will control the mechanism. If the finger need to be raised the servo motor will rotate resulting in pulling the finger.



If the finger need to pulled down the servo will loose the string which brings the finger to the default position.

| S No | Finger | Servo Max. Angle (degrees) |
|---|---|---|
| 1 | Thumb | 135 |
| 2 | Index | 180 |
| 3 | Middle | 175 |
| 4 | Ring | 140 |
| 5 | Little | 135 |

Run the code in cmd for real time data visualisation.

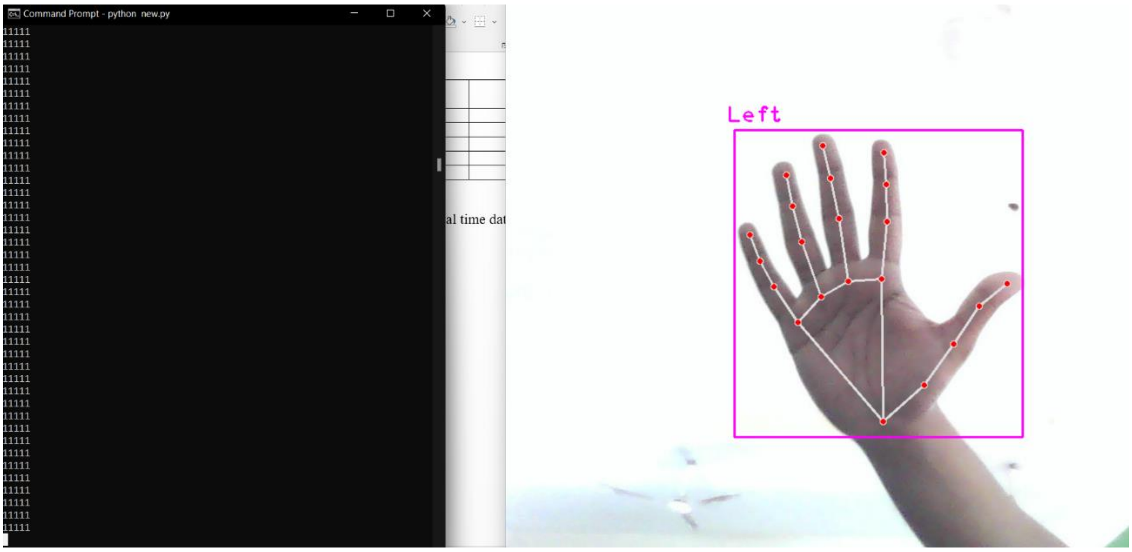
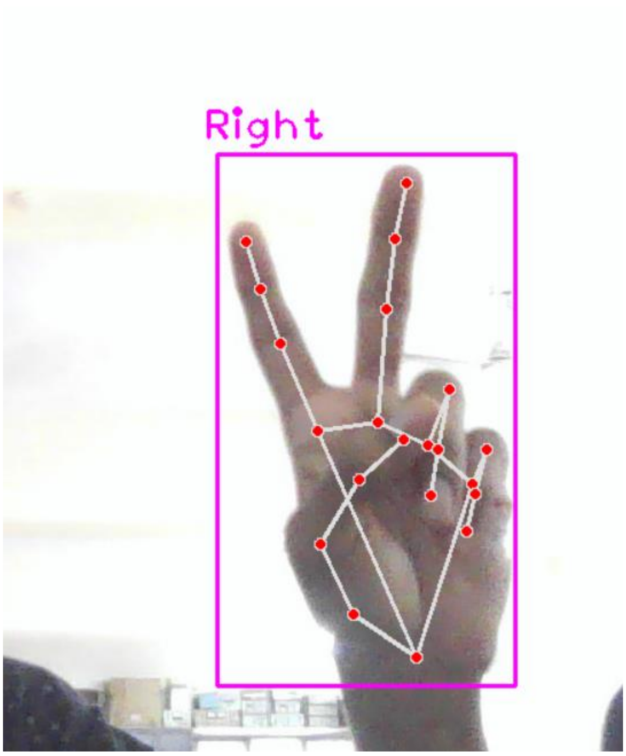
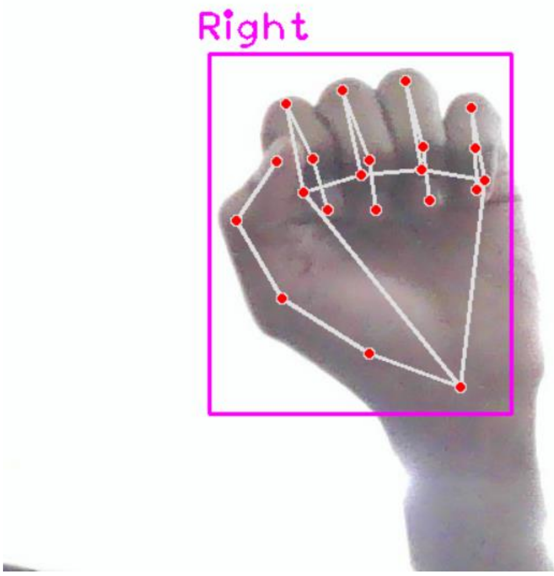
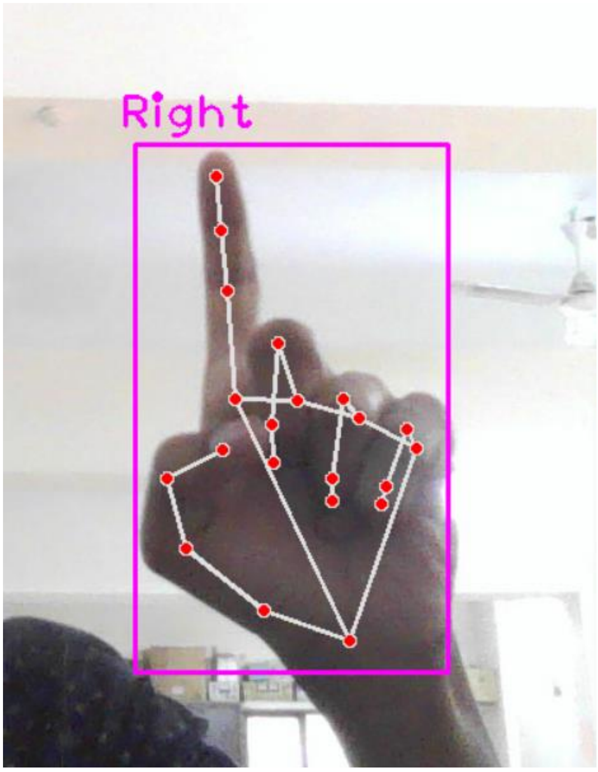


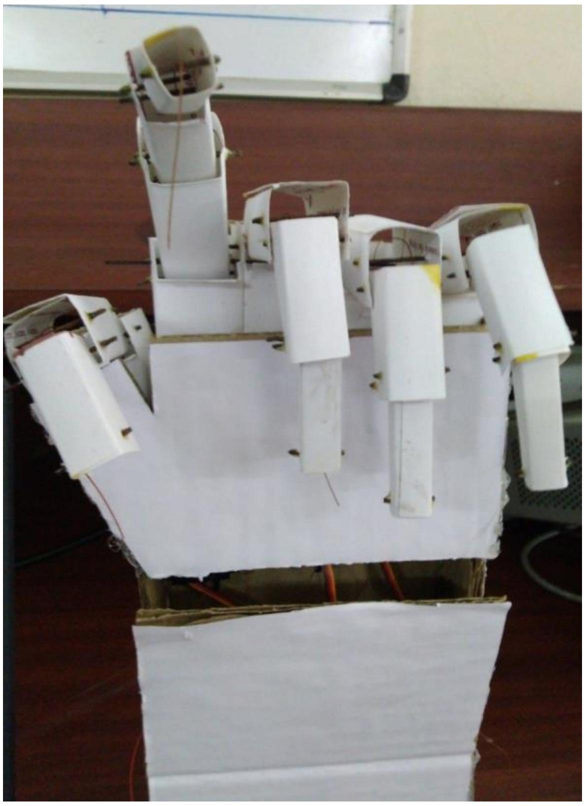
*Figure 17. Real Time*

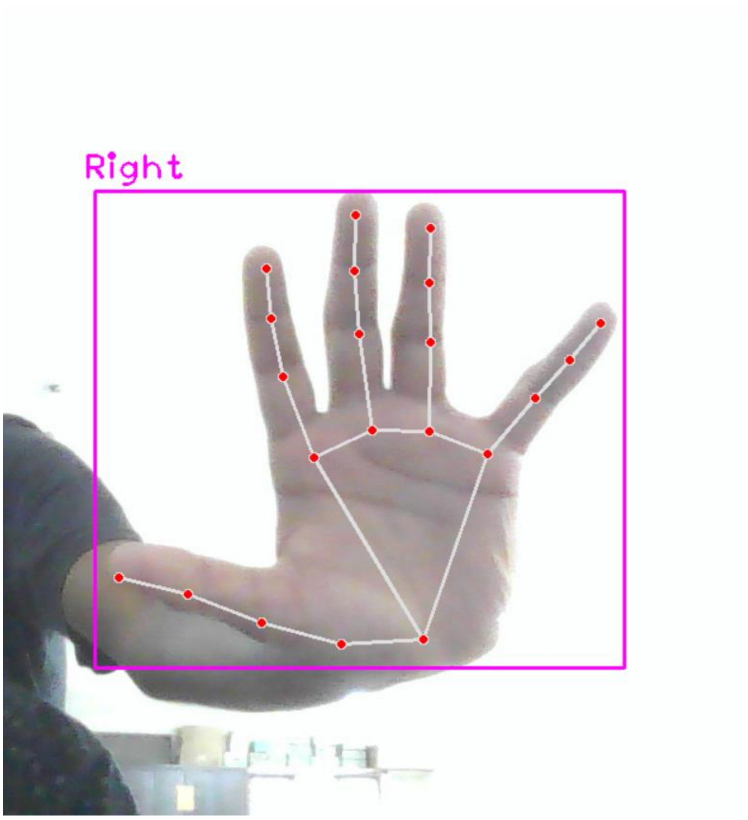
0 represents that the finger is closed and 1 represents the finger is open. The servo rotates as per mentioned maximum angles for each finger.

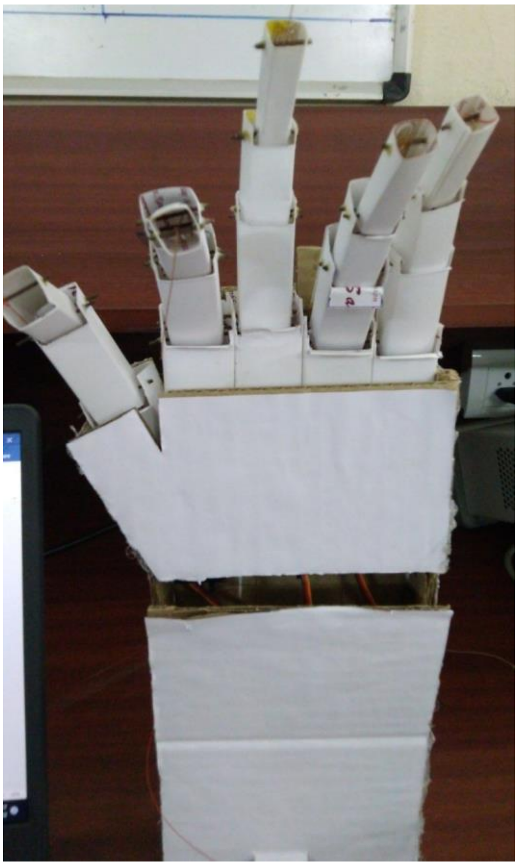


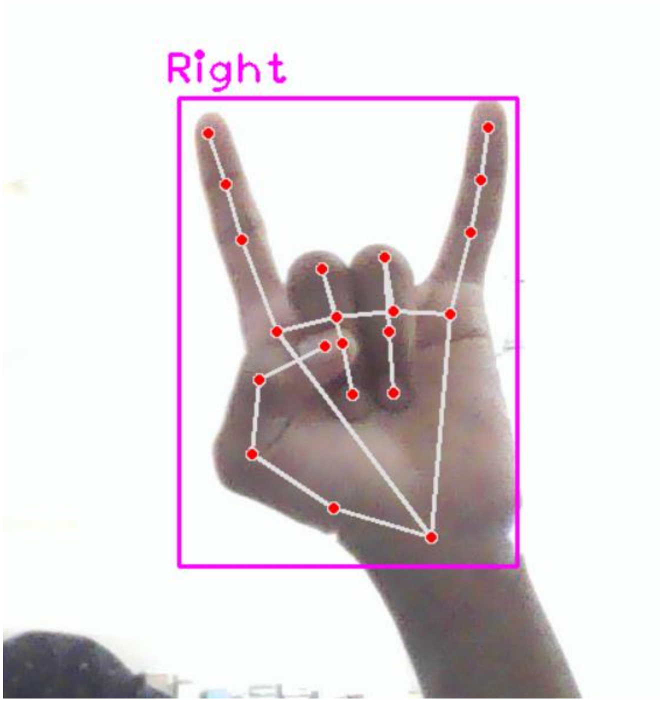


Right

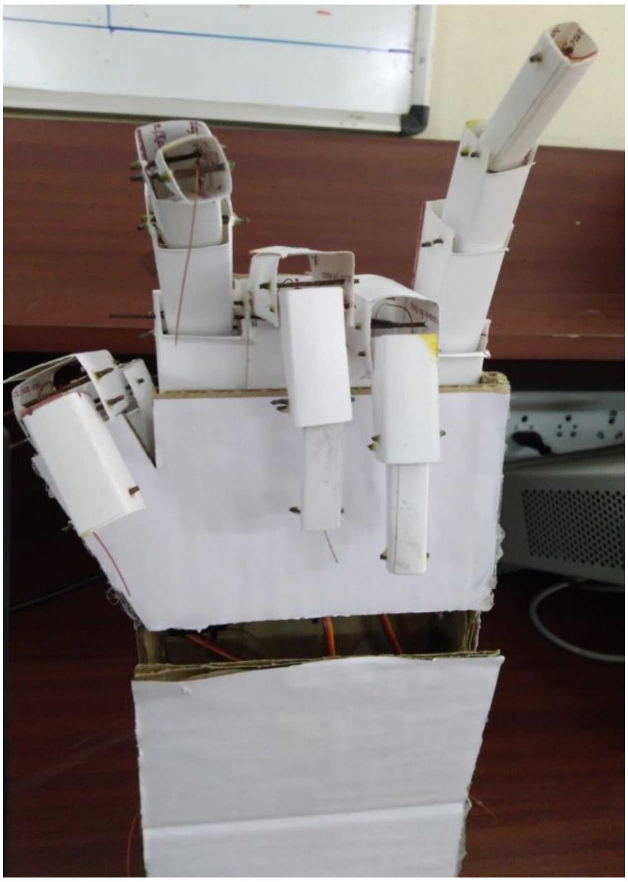

Right




Right




Right



# CONCLUSION

This Project presents a design of the Robot hand using Animatronics which mimics the natural movements of the human hand. The servos, controlled by an Arduino, are used to control the movement of the finger. The overall system performs responsibly well. The robot hand is even capable to carry out normal routine activities as the human hand does. The microcontroller accepts inputs and generates the proper control signals based on those inputs. Here, we have provided two techniques for giving gesture input, one is using image processing and the other is using Arduino control signals in which finger count specifies the command for the robot to navigate in a specific direction in the environment.

# FUTURE SCOPE

Robot hands have a wide scope of development. In the near future, the hands will be able to perform every task as humans in a much better way. Imagination is the limit for its future applications. It can be a real boon for handicapped people, who are paralyzed or lost their hands in some accident. The hand can be trained to listen to the command from a human and perform that task. A Precise gesture-controlled system is also possible.

Brain-Computer Interface(BCI) is an emerging field of research. BCI can be used to acquire signals from the human brain and control the hand. The system can work in the same way as the human hand.  A person who may lose his hand in an accident can resume his life like previous with such artificial hands. These Robot hands are versatile and have enormous ways of implementation.

# APPLICATIONS

Robotic arms can be used to automate the process of placing goods or products onto pallets. By automating the process, palletizing becomes more accurate, cost-effective,

and predictable. The use of robotic arms also frees human workers from performing tasks that present a risk of bodily injury.
These arms can be used as prosthetics to limbless people bringing no change to their daily life. High strength and durability are the additional features of the arm.

Humanoid robots for the household environment can be developed for assistance for the household activities

The rehabilitation robotics system can be used by disabled people, for instance, with upper limb impairments, to act more autonomously in daily life as well as in the working environment.

This is technological approach can be used in developing other parts of the body enlarging the scope of implementation

# REFERENCES


- Applications of a Fluidic Artificial Hand in the Field of Rehabilitation August 2007 OlegIvlev, Christian Pylatiuk, Tamim Asfour

- Hand-based tracking animatronics interaction October 2013 Cesar Guerrero-Rincon , Alvaro Uribe-Quevedo , Hernando Leon-Rodriguez , Jong-Oh Park2

- Animatronic Hand Controller December 2013 Sabrina Chowdhury

- Prosthesis control via Mirru App using MediaPipe hand tracking by Google Developers

- Creating and Testing Virtual Hardware (Port) Y. Ege, H. Çıtak, M. Çoramık

- Wu, X. H., Su, M. C., & Wang, P. C. (2010, October). A hand-gesture-based control interface for a car-robot. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 4644-4648). IEEE

- Zaman, H. U., Joy, A. A., Akash, K. M., & Fayad, S. T. (2017, June). A simple and effective way of controlling a robot by hand gesture. In 2017 International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 330-333). IEEE.