A Mini 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 QUARISHI (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 QUARISHI (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)

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 QUARISHI (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.

HEAD OF THE DEPARTMENT

DR. K. ANITHA SHEELA

(Professor of ECE department JNTUH-CEH)

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 CANDIDATE

We hereby declare that the project report title "**ANIMATRONIC ROBOT HAND USING IMAGE PROCESSING**" is a bonafide record of work done and submitted under the esteemed guidance of **Dr P.Chandra Sekhar Reddy**, Professor, Department of ECE, JNTUH-CEH, in partial fulfilment of the requirements for Mini project in Electronics and Communication Engineering at the Jawaharlal Nehru Technological University during the academic year 2021-22 is a bonafide work carried out by us and the results kept in the minor project has not been reproduced. The results have not been submitted to any other institute or university for the award of a degree or diploma.

By:

PATHA SAISRUJANA (19015A0416)

DULAPALLI NIHAL (18011A0413)

MANEESH KUMAR (18011A0421)

MUAZ HYDER QUARISHI (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 K. ANITHA SHEELA**, 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 QUARISHI (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 virtually created COM port then we prepare a schematic model in proteus which takes input from the created COM port and provides a real-time simulation model. This technique helps to efficiently translate hand gestures into a simulated virtual hand.

CONTENTS

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, cross-platform 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.

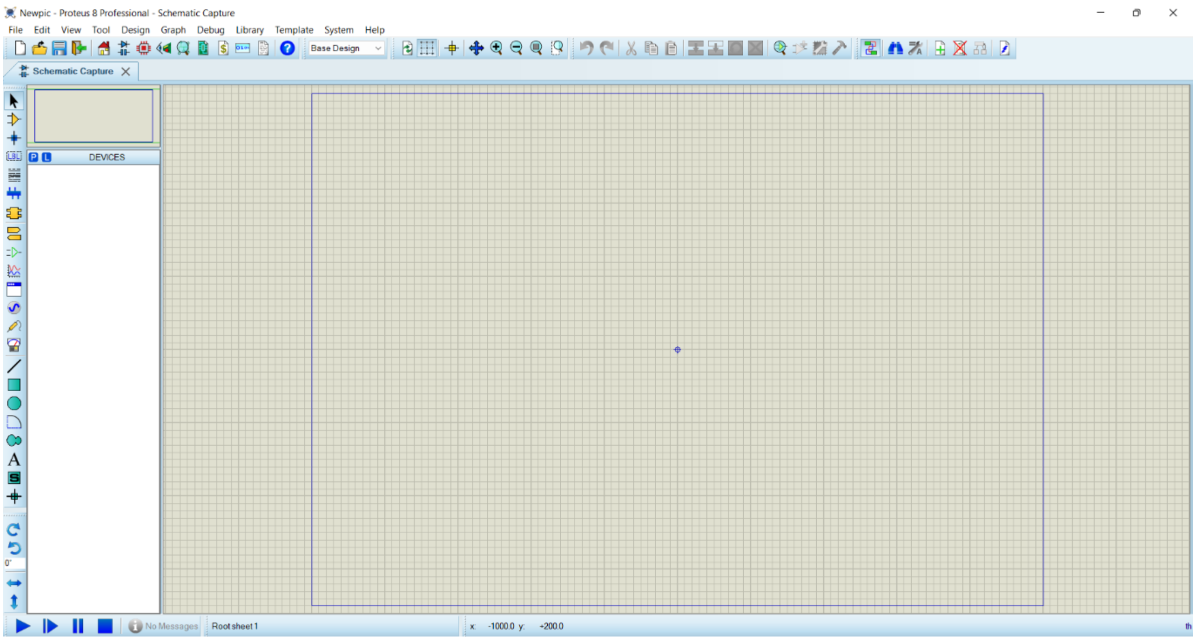
Furthermore, the scope of the project expands into Bionic Prosthetic.
185,0000 people have an amputation each year and around 6 Million people are limbless. A bionic arm is an electromechanical device that attaches to the human body and replicates the functionality of a natural arm/hand. It always consists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

PROJECT DESIGN

SOFTWARE USED

Proteus Design Suite

Proteus software is developed by Labcenter Electronics. Proteus is software used for electronic circuits, microprocessor-based circuits simulation and for designing printed circuit boards (PCB). The main feature of Proteus design software is its multiple system components. Proteus is a package that has ISIS Schematic Capture which is a tool used for circuit design and simulations, Printed Circuit Board (PCB) design known as ARES PCB Layout, VSM (Virtual System Modeling) which is embedded software with popular microcontrollers and hardware design. The designed circuits are simulated for analyzing the performance and behaviour of circuits before hardware implementation. In Proteus ISIS schematic capture, the most commonly used components of many leading manufacturers are listed with their part number and the ratings.



Proteus Software

Another main feature of Proteus is that it can be integrated with mikroBasic software. mikroBasic is developed by mickroelectronika, and mikroBasic PIC is a compiler used to build PIC microcontroller based projects. This software allows users to write and compile the programs and the programs compiled in mikroBasic can be directly uploaded into the microcontroller chip used in the Proteus design suite. So, the microcontroller-based circuit analysis is easy and simple in Proteus.

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.

Arduino Software

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.

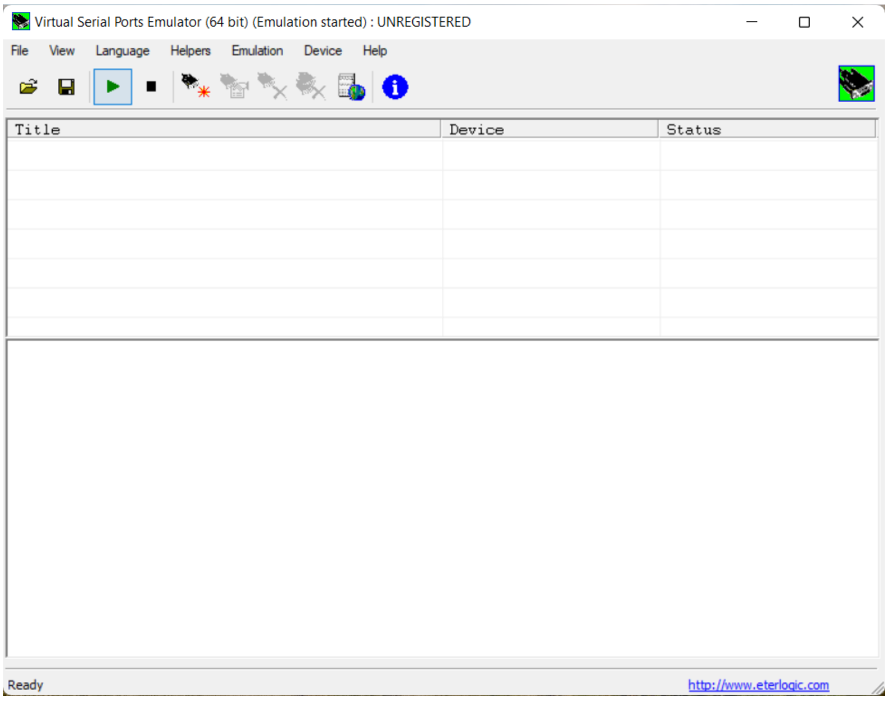
MediaPipe is a framework for building multimodal, cross-platform applied ML pipelines. With MediaPipe, a perception pipeline can be built as a graph of modular components, including, for instance, inference models (e.g., TensorFlow, TFLite) and media processing functions.

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.

VSPE

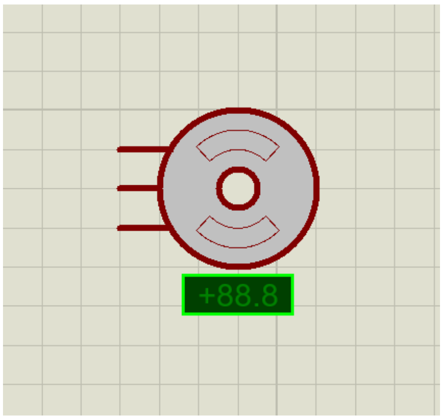
Virtual Serial Port Emulator (VSPE) is an easy-to-use virtual drive creation program that allows to create virtual serial port drivers and virtual serial ports. The virtual serial ports created by the VSPE program work exactly like real serial ports. By using this program, any number of virtual port pairs can be created. Unlimited number of virtual COM ports, virtual port creation, visibility of connection points, easy installation, real port configuration, VSPD, WDM, WMI, power management and PnP support are among the features of the program. COMPORT, TCP / IP, Telnet, IP and I2C connections can be done with the REALTERM terminal program.



SIMULATED HARDWARE

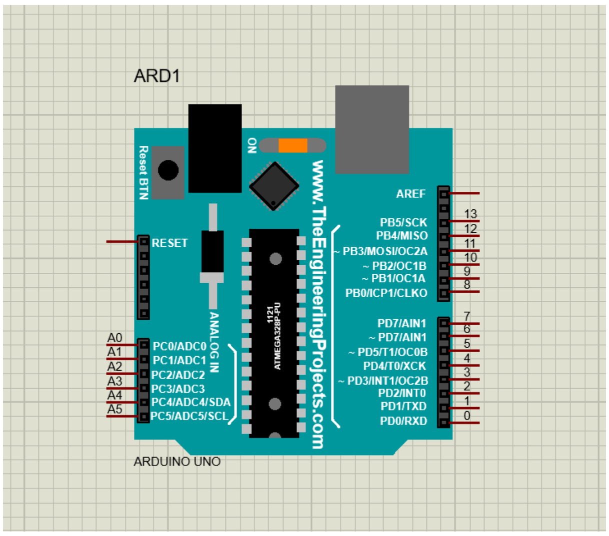
Servo Motor

A Servo motor consists of three major parts motor, a control board, potentiometer connected to the output shaft. The potentiometer, which controls the angle of the servo motor, allows the control circuitry to monitor the current angle of the shaft. The motor, through a series of gears, turns the output shaft and the potentiometer simultaneously.



Arduino Board

Arduino Uno is an 8-bit microcontroller based on AVR architecture. It is commonly used in many projects and autonomous systems because of its simplicity, low-cost. UNO is the best board to get started in the field of electronics and coding. UNO is the most robust board you can start working with. It is the most used and documented board in Arduino/Genuino family



TECHNICAL SPECS:

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

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)
```

```python
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)
```

Arduino Code

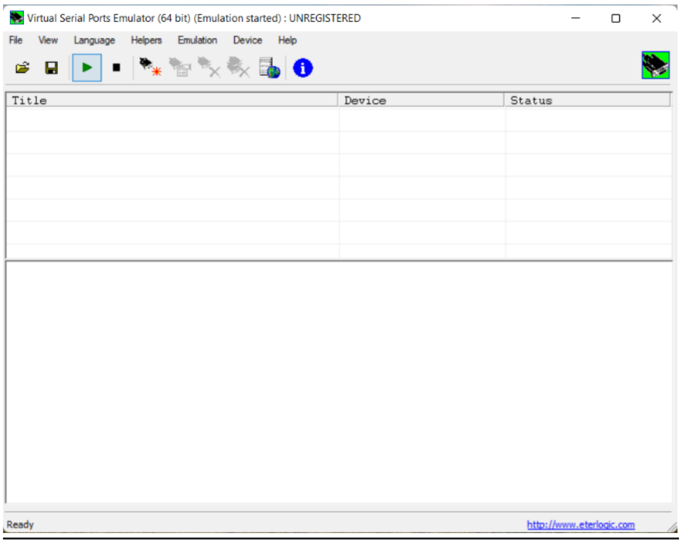
```cpp
#include <Servo.h>

#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);}
}
```

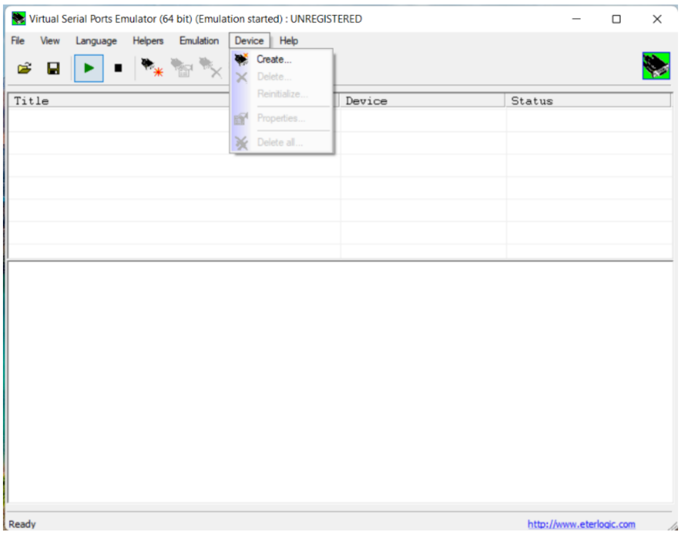
SETUP

Create a Virtual Port

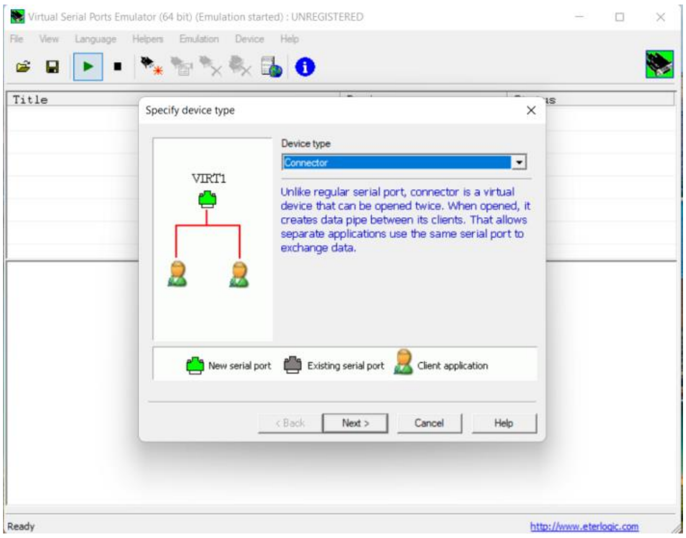
1) We use VSPE to create a virtual port.



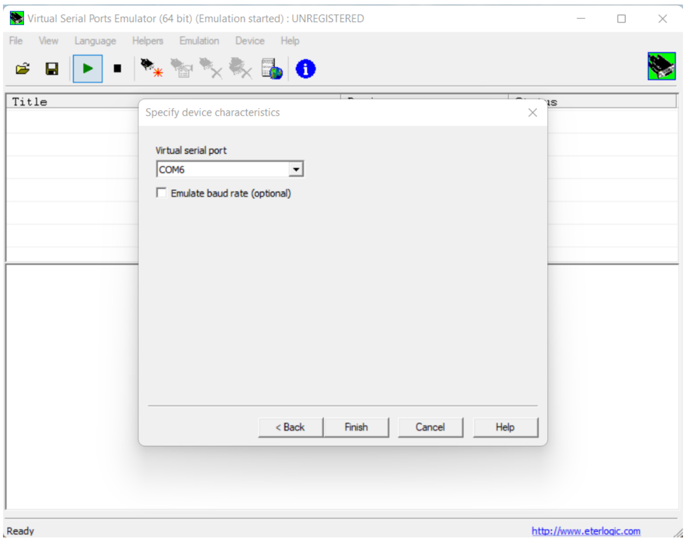
2) Go to Devices in Menu bar and Create



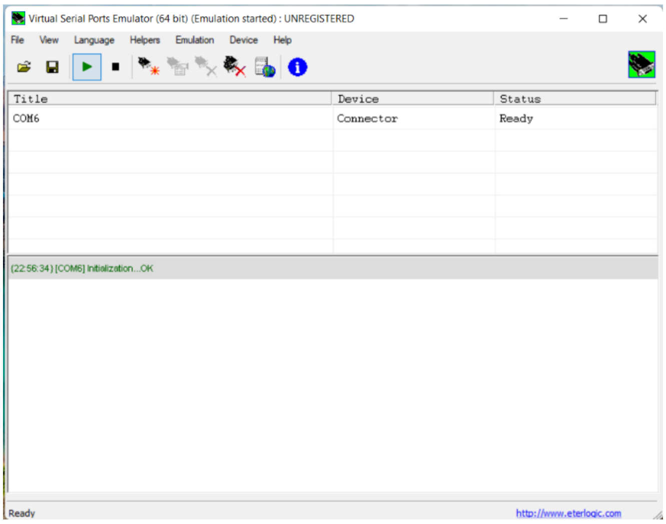
3) Select Connector as device type



4) Select the name of the virtual port (Note the port name)

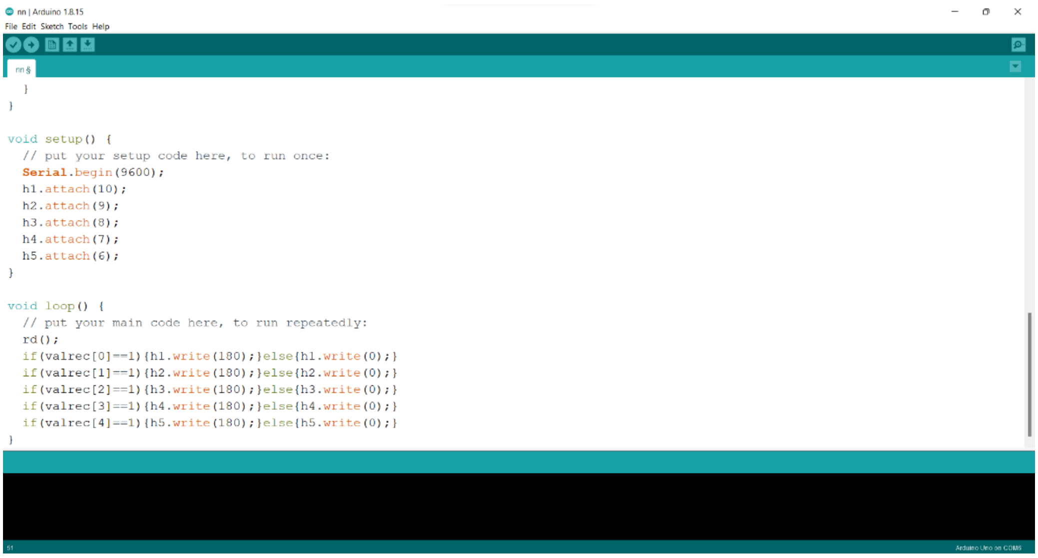


5) And make sure the port is ready and running

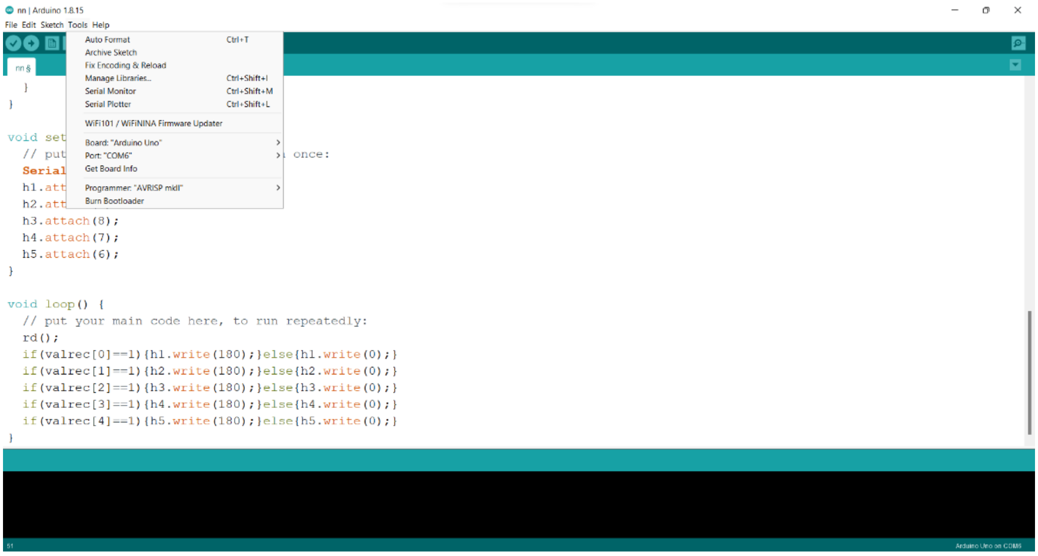


Arduino Setup

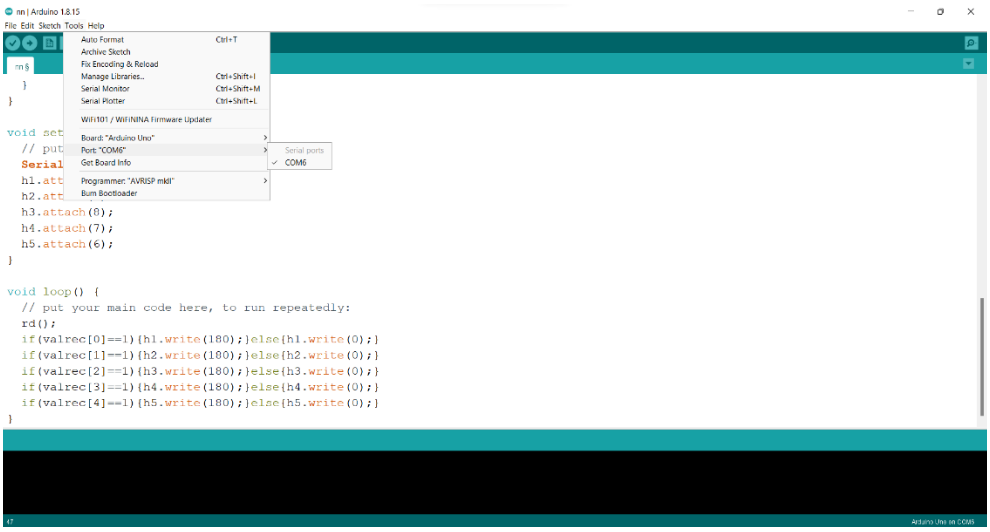
1) Write the code in the IDE



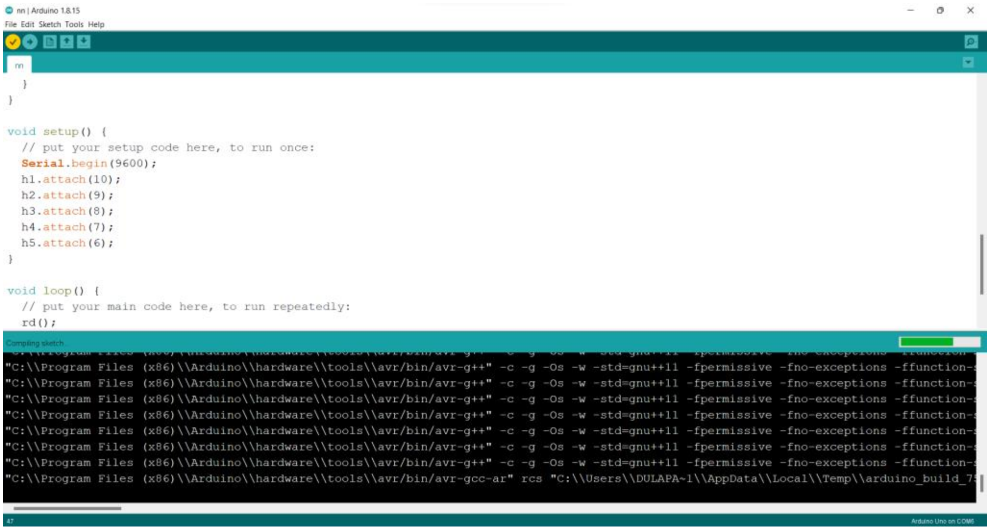
2) Go to Tools in the menu bar



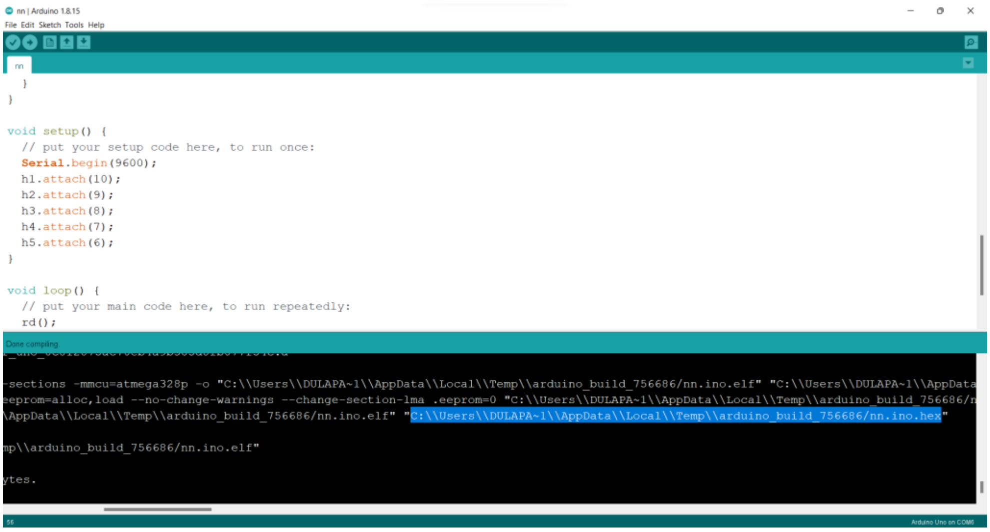
3) Select the Arduino Uno board and the virtual port created



4) Compile the code

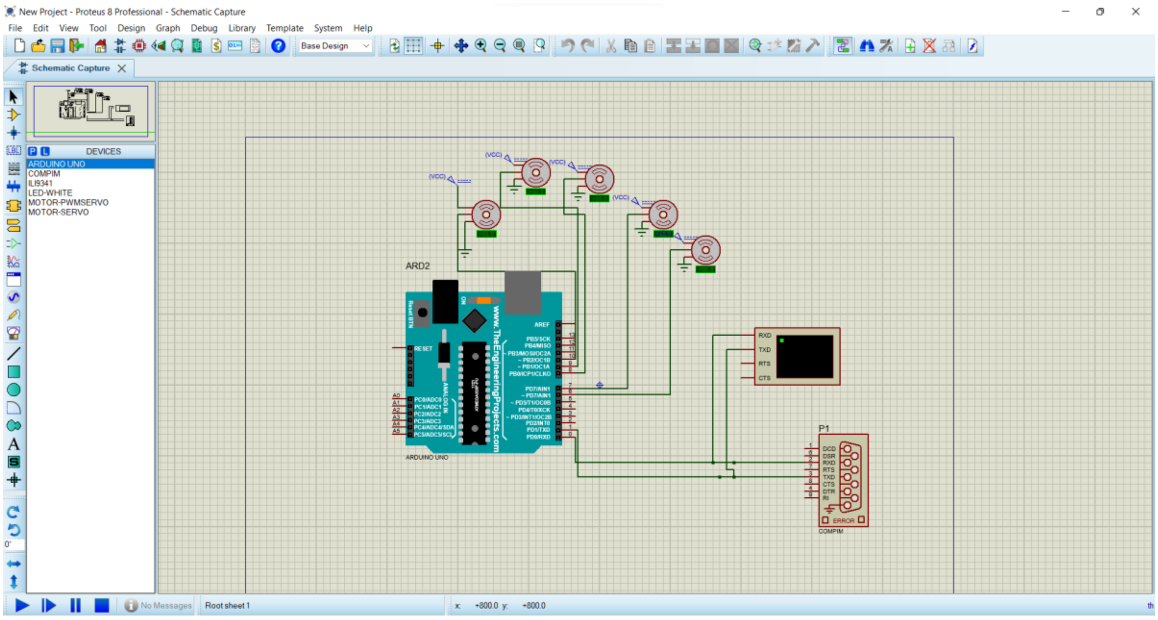


5) Note the path of the hex file created after compilation

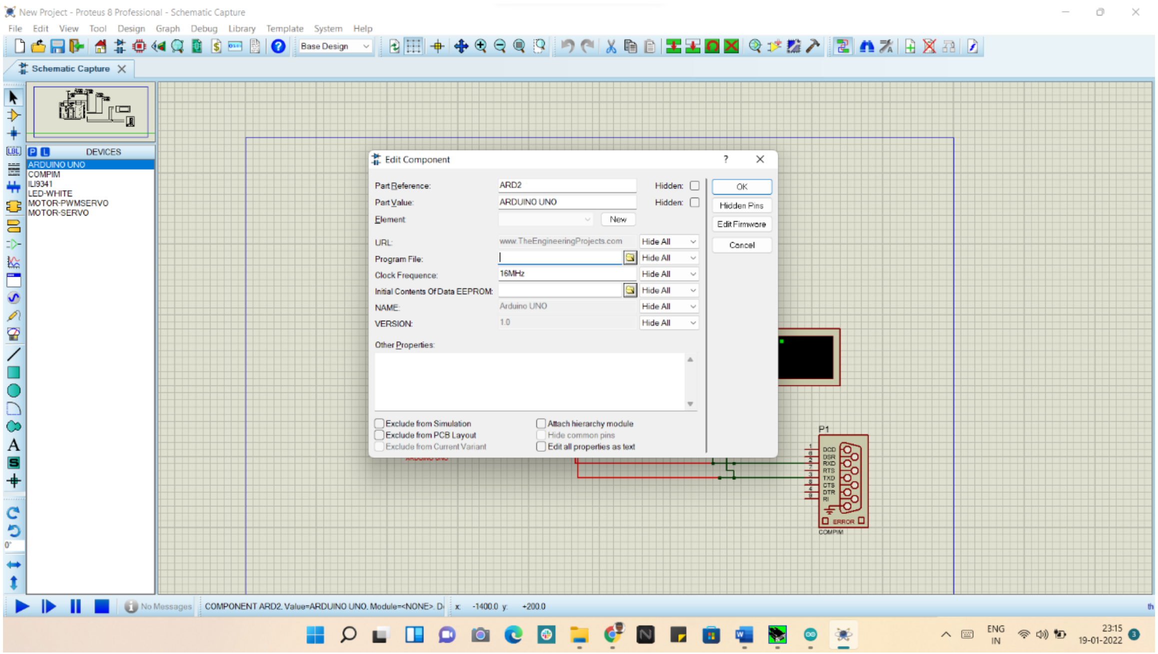


Proteus Setup

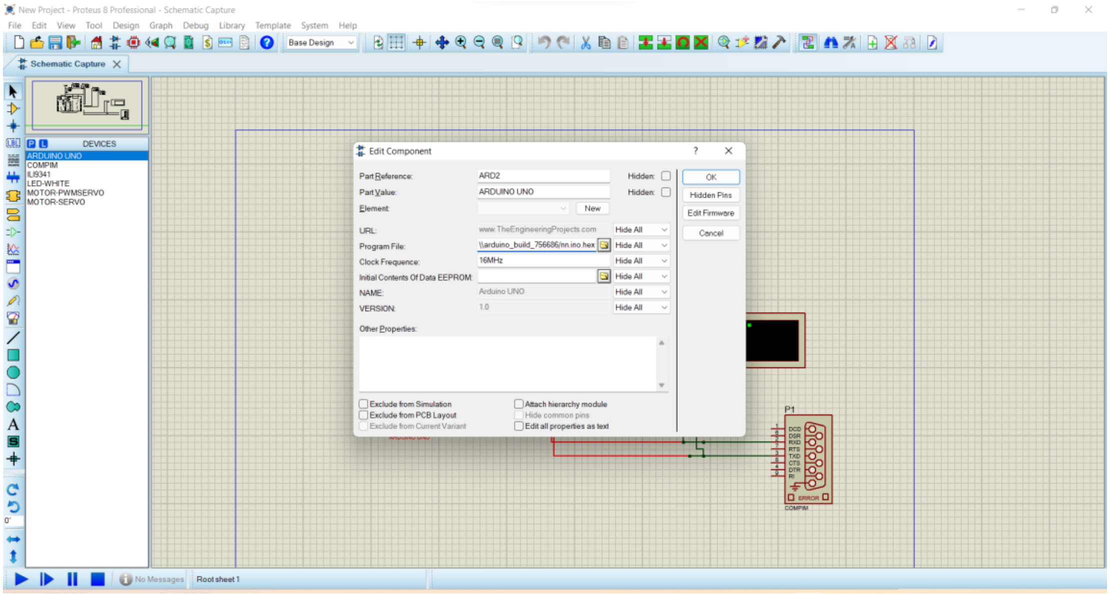
1) Using the virtual Hardware components design the circuit



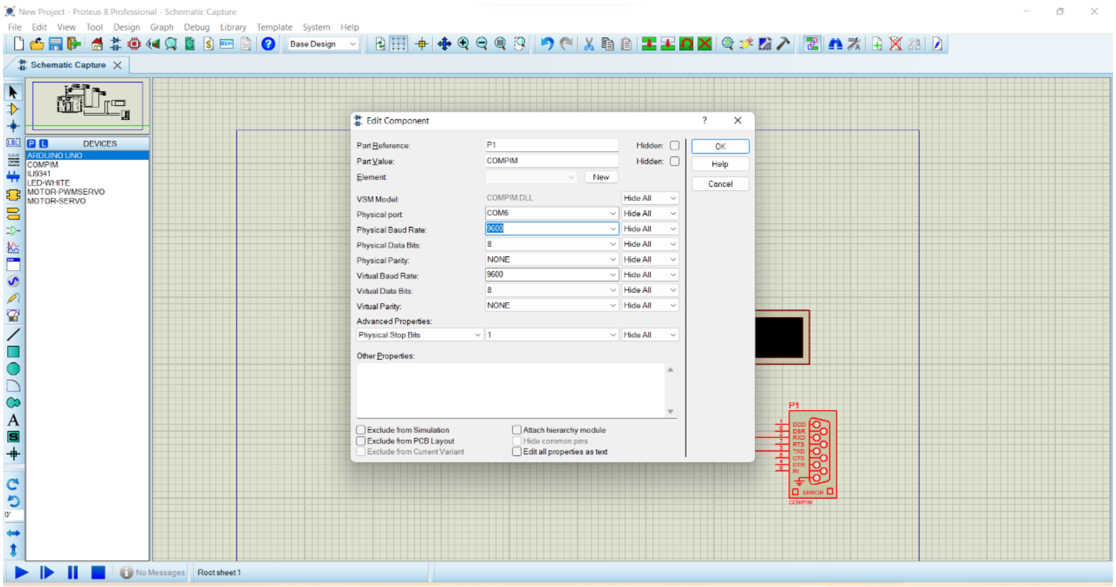
2) Click on the Arduino for edit component Window



3) At the program file place, the hex file created of Arduino



4) Make sure the baud rate of COMPIM is the same as mentioned in the Arduino code.



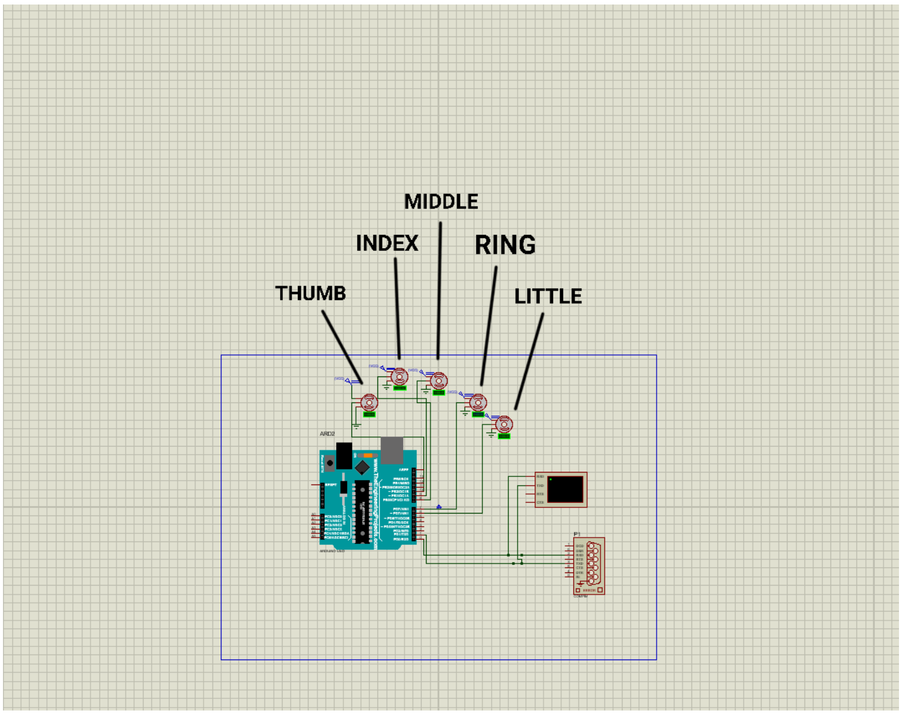
RESULT

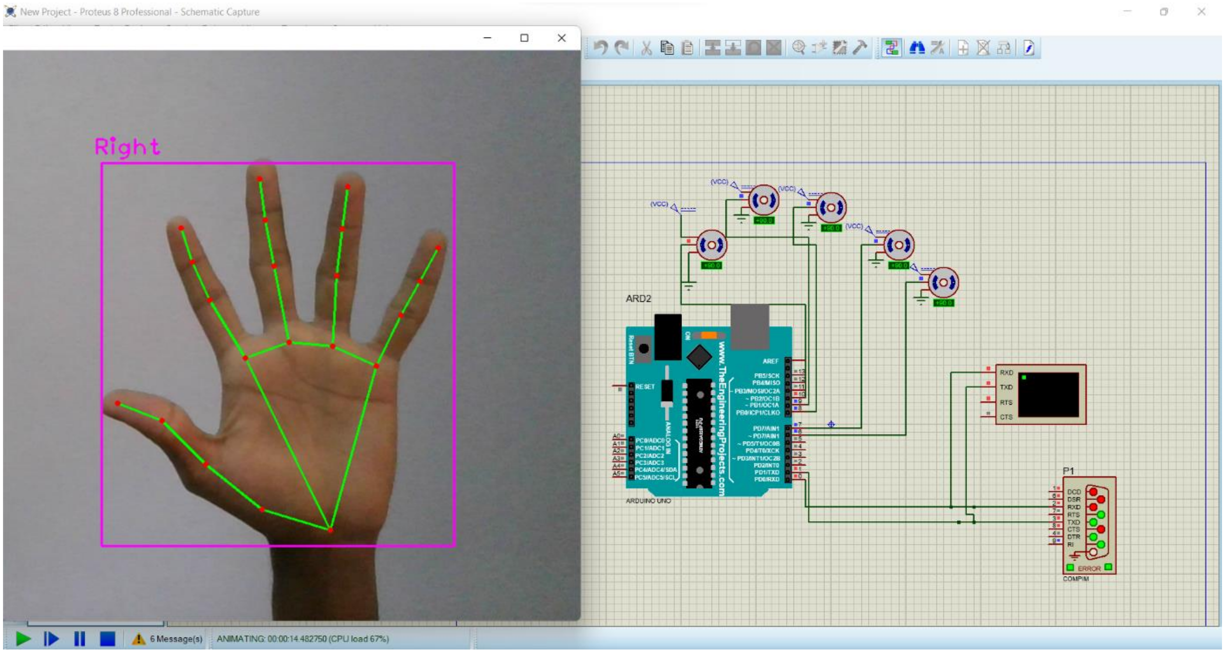
In the main file of python code make sure the serial port is mentioned with the created port at line 10 of the code.

```
6    cap.set(3,1280)
7    cap.set(4,720)
8
9    detect = HandDetector(detectionCon=0.8,maxHands=1)
10   ms=cvzone.SerialModule.SerialObject("COM6",9600,1)
11
12   while True:
13       stat,img = cap.read()
14       img = cv2.flip(img, 1)
```

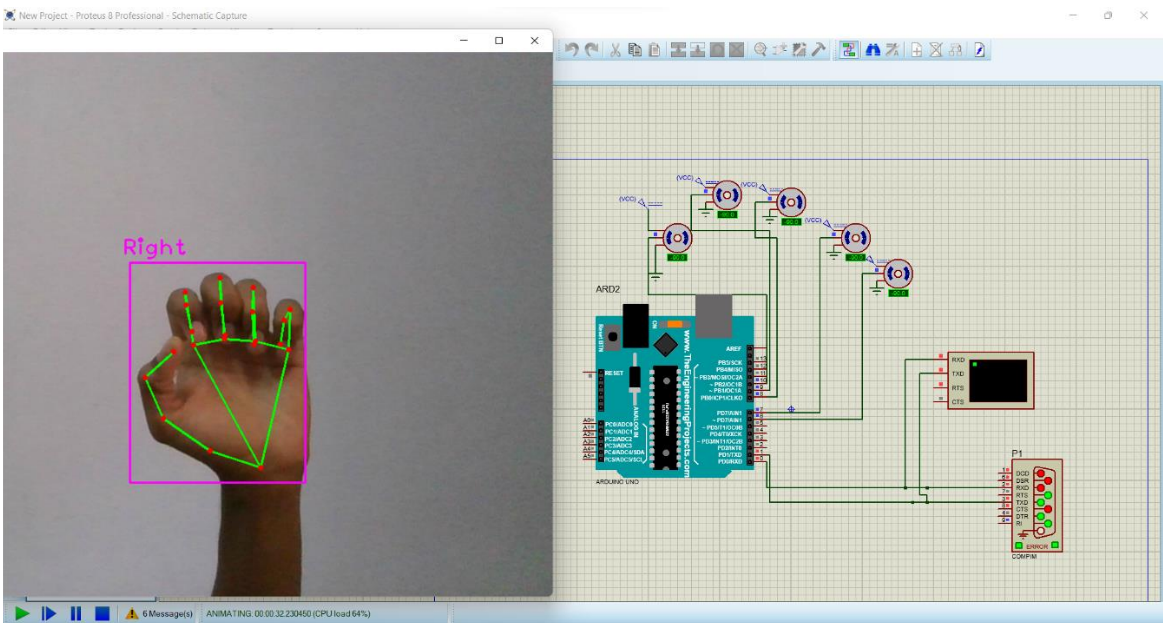
The created virtual port should be active all the time. Now run the circuit in Proteus and python code simultaneously.

The virtual hand is represented with each servo controlling their respective finger.

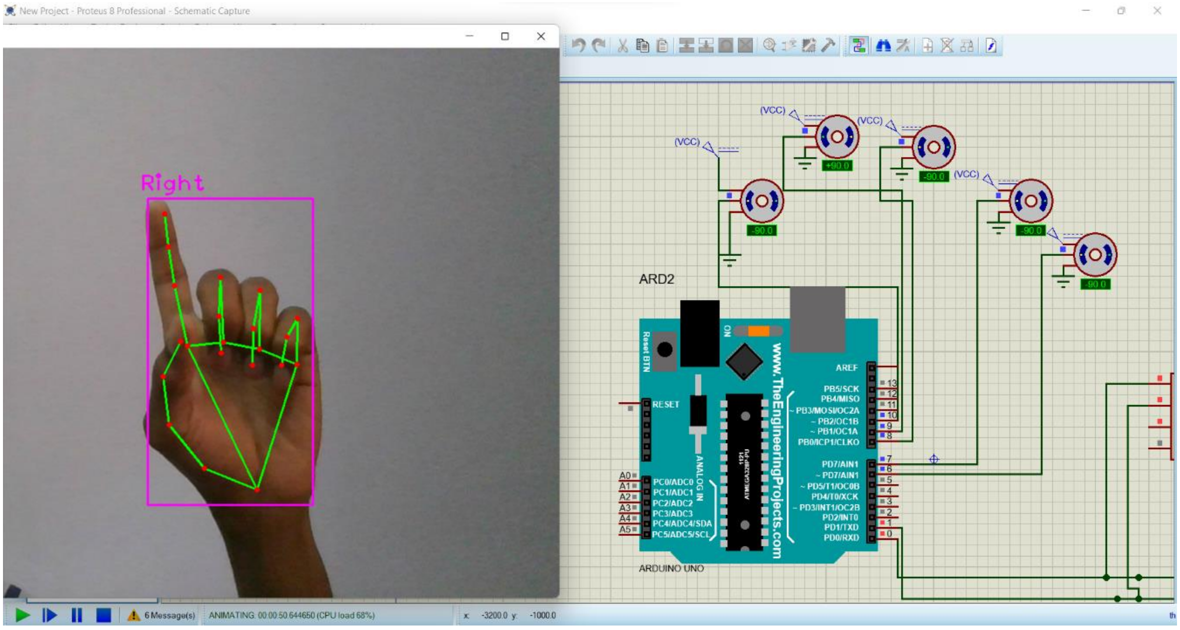


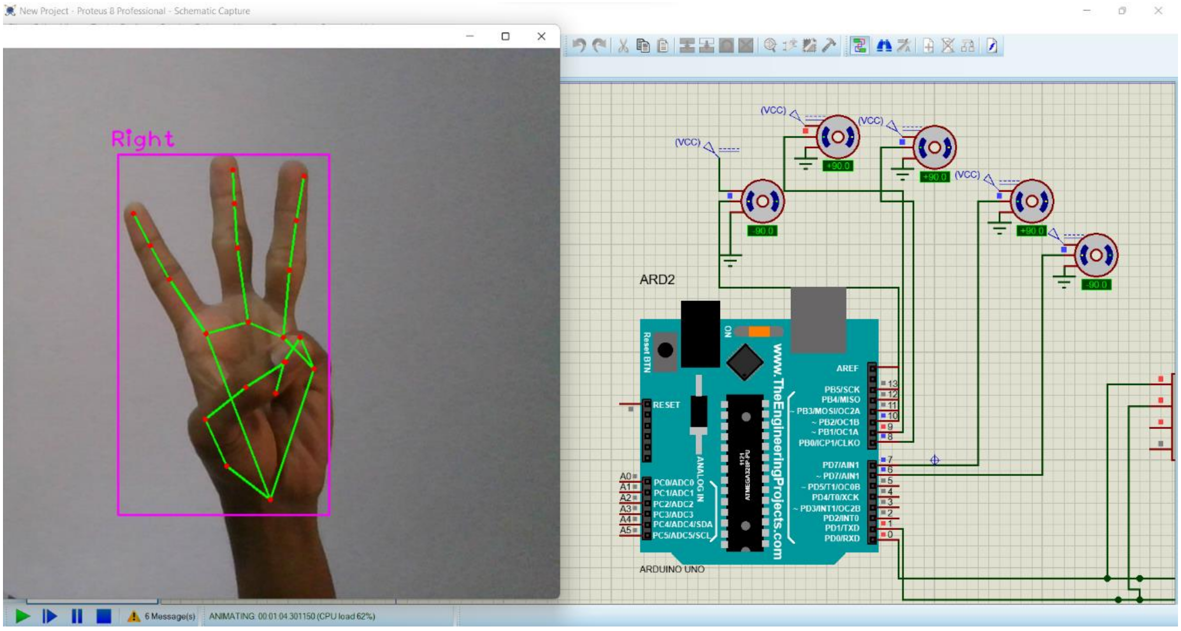


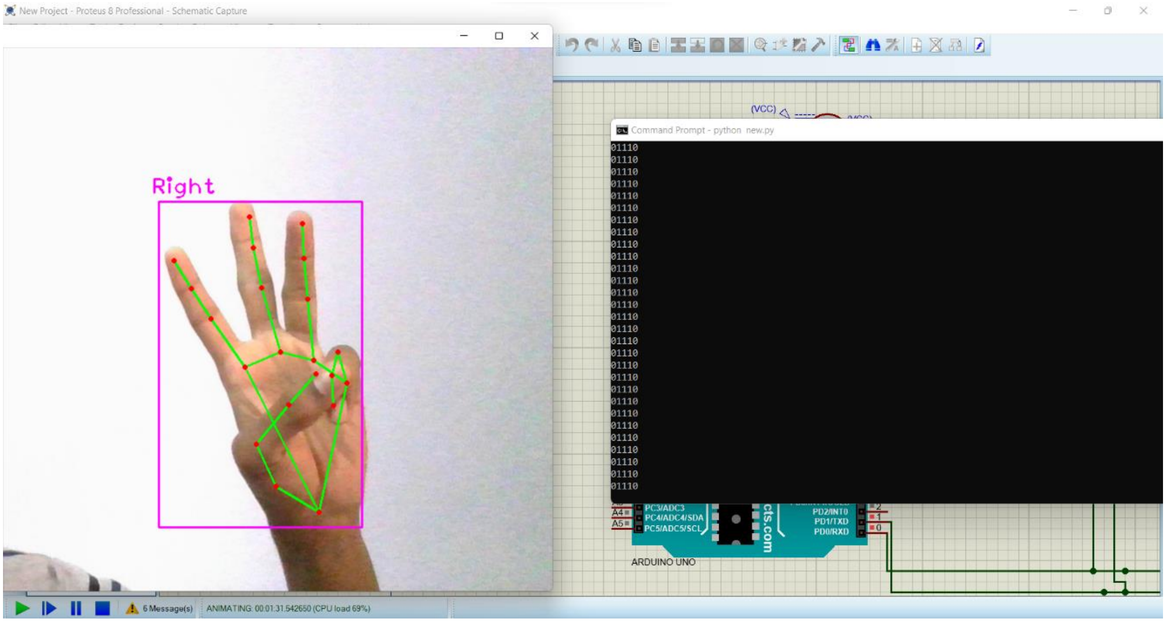
+90 degrees in servo represents the finger is open



-90 degrees in servo represents the finger is close


New Project - Proteus 8 Professional - Schematic Capture
Right
ARD2
ARDUINO UNO
www.TheEngineeringProjects.com
(VCC)


New Project - Proteus 8 Professional - Schematic Capture
Right
ARD2
ARDUINO UNO
www.TheEngineeringProjects.com
(VCC)



We get real-time data in the command prompt as a string with 0’s and 1’s. With format as Thumb, index, middle, ring, little.

“0” represents that the finger is close and “1” represents that the finger is open.

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