WIRELESSLY CONTROLLED SERIAL MANIPULATOR

MINOR PROJECT REPORT

Submitted by

NAME OF THE CANDIDATES WITH REG No

ANIMESH GHOSHAL	RA1611018010025
SATYAPALSINH GOHIL	RA1611018010042
ATIF AKHTAR	RA1611018010099
SATYAM DUDHAGRA	RA1611018010102

Under the guidance of

DR.T.MUTHURAMALINGAM, PHD

(Associate Professor, Department of Mechatronics Engineering)

of

SEVENTH SEMESTER(15MH376L)

in

MECHATRONICS ENGINEERING

of

FACULTY OF ENGINEERING & TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Kancheepuram District
OCTOBER 2019

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled **"WIRELESSLY CONTROLLED SERIAL MANIPULATOR"** is the bonafide work of "**ANIMESH GHOSHAL, SATYAPALSINH GOHIL, ATIF AKHTAR AND SATYAM DUDHAGRA",** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
DR.T.MUTHURAMALINGAM
GUIDE
ASSOCIATE PROFESSOR
MECHATRONICS

SIGNATURE
Dr. G MURALI
HEAD OF THE DEPARTMENT
MECHATRONICS

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

The requirement of assistance in workplaces such as fabrication lab, workshops and small-scale industries is in, nowadays, a growing demand as can be seen with development of human assistive robots. Therefore, we aid to such requirement by designing a wirelessly controlled pick and place robot in industrial environment which will serve as an assistant and even encourage human-robot interaction. The robot manipulator will be a 3 degree of freedom manipulator (with RRR configuration). The task of assistance will be performed by end-effector such as picking or placing the object in the pre-defined place. The user will provide an input to the robot manipulator using a mobile application. The robot will perform the desired task. The robot is actuated electrically and it will be the task of microprocessor to process all the data and perform kinematics to achieve the desired task.

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank the members of SRM TEAM ROBOCON, for providing us an opportunity to do the project work in Robocon Lab, 5th Floor, Basic Engineering Lab and giving us support and guidance which helped in completing the project duly. We are extremely thankful to them for providing such a nice support and guidance, although he had busy schedule managing their corporate affairs.

We owe deep gratitude to our project guide Dr.T.Muthuramalingam, who took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Department of Mechatronics which helped us in successfully completing our project work. Also, I would like to extend our sincere esteems to all staff in laboratory for their timely support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 PROJECT BACKGROUND

As India enters the era of robots and artificial intelligence, most companies plan to automate jobs to compete on cost. But will replacing humans with machines really turn India into a global economic powerhouse? A new report suggests that a more effective strategy could be to train its workers for the future, giving them the skills, they need to work with sophisticated machines, data and algorithms. For businesses facing increasingly complex production processes and a tough global marketplace, cheap labour is no longer the main factor. Instead, the most successful employers are looking for talent and skills in new hires, using technology to enhance rather than automate existing jobs.

A pick and place robot are the one which is used to pick up an object and place it in the desired location. It can be a cylindrical robot providing movement in horizontal, vertical and rotational axes. The robots are faster and can get the work done in seconds compared to their human counterparts, they are flexible and have the appropriate design, they are accurate, they increase the safety of the working environment and actually never get tired.

A wirelessly controlled robotic manipulator as in our project deals with the human robot interaction in an industrial or human-restricted environment. The camera (OV7670) attached on the end effector provides a live-feedback that is achieved and controlled through an android mobile application.

The various components used in this project consist of Arduino Mega, ESP 32, OV7670, Servo Motor, DC Geared Motors, Potentiometers, LiPo Battery, Rubber Belt used for transmission, L298N Motor Driver and Power Board.

This robot can be used in human-restricted environments and fabrication lab where this can be helpful for simplifying arduous tasks and help in increasing production.

1.2 SERIAL AND PARALLEL MANIPULATORS

Serial Manipulators are the most common industrial robots and they are designed as a series of links connected by motor-actuated joints that extend from a base to an end-effector. Often they have an anthropomorphic arm structure described as having a "shoulder", an "elbow", and a "wrist". In its most general form, a serial robot consists of a number of rigid links connected with joints. Simplicity considerations in manufacturing and control have led to robots with only revolute or prismatic joints and orthogonal, parallel and/or intersecting joint axes (instead of arbitrarily placed joint axes). The reachable workspace of a robot's end-effector is the manifold of reachable frames. The dextrous workspace consists of the points of the reachable workspace where the robot can generate velocities that span the complete tangent space at that point, i.e., it can translate the manipulated object with three degrees of freedom, and rotate the object with three degrees of rotation freedom. The relationships between joint space and Cartesian space coordinates of the object held by the robot are in general multiple-valued: the same pose can be reached by the serial arm in different ways, each with a different set of joint coordinates. Hence the reachable workspace of the robot is divided in configurations (also called assembly modes), in which the kinematic relationships are locally one-to-one.

A parallel manipulator is a mechanical system that uses several computer-controlled serial chains to support a single platform, or end-effector. This device is called a Stewart platform or the Gough-Stewart platform in recognition of the engineers who first designed and used them. A parallel manipulator is designed so that each chain is usually short, simple and can thus be rigid against unwanted movement, compared to a serial manipulator. Errors in one chain's positioning are averaged in conjunction with the others, rather than being cumulative. Each actuator must still move within its own degree of freedom, as for a serial robot; however in the parallel robot the off-axis flexibility of a joint is also constrained by the effect of the other chains. It is this closed-loop stiffness that makes the overall parallel manipulator stiff relative to its components, unlike the serial chain that becomes progressively less rigid with more components.

The workspace provided by parallel manipulator is far less compared to a serial manipulator and since, most of the applications in industries require a wide range of movements. Therefore, a Serial Manipulator is a better choice for performing a particular task. Another drawback of parallel manipulators is their nonlinear behaviour: the command which is needed for getting a linear or a circular movement of the end-effector depends dramatically on the location in the workspace and does not vary linearly during the movement.

1.3 CHALLENGES AND OPPORTUNITIES

In India the population of unskilled labour is high due to which very assemblage, control and maintenance of these systems becomes difficult. Therefore, wirelessly controlled manipulators have simple control structure which nullifies the various drawbacks as well as complexity.

Employing these systems in an industrial environment requires a huge amount of capital and availability of raw materials because these components are not manufactured within the country and acquiring them leads to high import tax which many manufacturing industries cannot yield. Thus, to counteract this demerit, Government of India in its budget 2018 allocated $480 Mn (INR 3,073 Cr) for Robotics, IOT and Artificial Intelligence/Machine Learning. It also promises skilled training in these domains which results in overall increase in production and economy.

Hence an approach has been carried out to simplify this manufacturing approach using wirelessly controlled serial manipulator that is suitable for the work environments of fabrication labs as well as human restricted environments. It also provides a simplified control which makes the unskilled labour adaptable to industrial environment.

1.4 MOTIVATION FOR WORK

In one of the biggest moves ever made to support the country's AI, ML, Robotics and IoT sectors, the Indian government has doubled its allocation to the 'Digital India' mission to around $480 Mn (INR 3,073 Cr).

As stated by **Aakrit Vaish, founder and CEO of Haptik**, "Allocation of significant fund and announcing efforts to enhance research in disruptive technologies like Artificial Intelligence (AI), Internet of Things (IoT) and Robotics implies that the importance of adoption of such technologies has finally been taken into consideration by the government."

He went on to say, "With NITI Aayog set to establish a national programme for artificial intelligence, this will not only significantly aid job creation but will also assist the government to move towards its Digital India vision."

The latest $480 Mn commitment points to the government's intentions to promote fifth generation technologies like AI, ML, IoT, Robotics. Consequently, India is expected to see more and more application of these cutting-edge technologies, which would in turn help bring the economy at par with developed countries.

Adaptation of such technologies will not only boost the ease of doing business, but will also help in accelerating the country's economic growth. The government's efforts in this regard seem to have fructified, given that the Indian Robotics market is expected to touch $15 Bn within the next two years.

Therefore, to become a part of this initiative, our project ensures a synergistic relationship between industry and 'Digital India' campaign by implementing a robotic system that can be used for performing difficult pick and place tasks and to encourage human robot interaction.

1.5 PROBLEM DEFINITION

Functional design of this Wirelessly controlled serial-manipulator is carried out by considering it as a 3R manipulator configuration, with the help of gear mechanisms and electronic actuators. Each joint is controlled using 12V motors of high torque and potentiometer and the end effector is controlled using servo motor. The whole system is powered by a 12V LIPO Battery of 6000mAh. A vision sensor OV7670(camera) is used for achieving the live feedback of the end effector so as to position it as per the required work environment. The camera is connected with ESP 32 which interacts with the Arduino mega. The whole system is controlled using a android mobile application.

1.6 OBJECTIVE

- To motivate human-robot interaction.
- Making the control user-friendly.
- Easing routine work in a potential working environment.
- Relieving humans from complex manual control.
- Wireless operation within a particular range.

CHAPTER 2

LITERATURE SURVEY

1. **Yusoff,M.A.K., Samin.R.E., Ibrahim.B.S.K Procedia Engineering Volume (2012),** in their paper titled,**"Wireless Mobile Robotic Arm"** portrays the development of a wireless robotic arm which is controlled using PS2 controller and Arduino Mega, which helps in picking up hazardous object that is far away from user.

2. **Minca, E., Iliescu,A., Voda,A., (2014),** in their paper titled,**"Modelling and Control of an assembly/disassembly mechatronics line served by mobile robot with manipulator"** presents an assembly line which consists of a robotic manipulator. This system can perform assembly and disassembly using synchronised hybrid petri nets(SHPN). This enables complete assembly and disassembly of parts.

3. **R.Correal., A.Jordan., A.Gimenez., C.Balaguer (2004),** in their paper titled**,"Wireless Teleportation of an Assistive Robot by PDA"** provides a climbing robot which supports in daily needs and activities of elderly individuals. It also presents a distributed architecture and the concept design of the HMI which handles the robotic system.

CHAPTER 3

PROJECT DESIGN AND IMPLEMENTATION

3.1 MECHANICAL DESIGN AND FABRICATION

Table 3.1: Mechanical Components and Specification

Sr No.	Components	Specification
1	Frame	Wood.
2	Gears	Plastic. Gear Ratio-: 1:2
3	Motor Coupler	Aluminium.
4	Motor Mounts	PLLA Material.
5	End Effector	2 Finger Claw.
6	Potentiometer Mount	Acrylic.
7	Belt Drive	Rubber
8	Caster Robot Ball Wheel	Qty.- 2
9	Pulley	PLLA Material Ratio-: 1:2

The CAD model shows the entire assembly of the system with proper dimensioning. As we see Fig 3.1 and 3.2 show the side view and top view of the system.

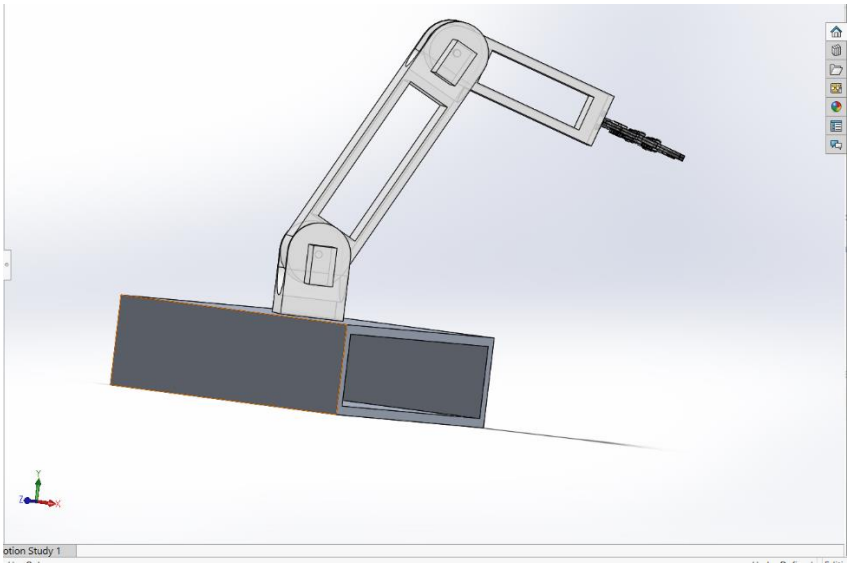

Fig 3.1: CAD Model Side View of the System.

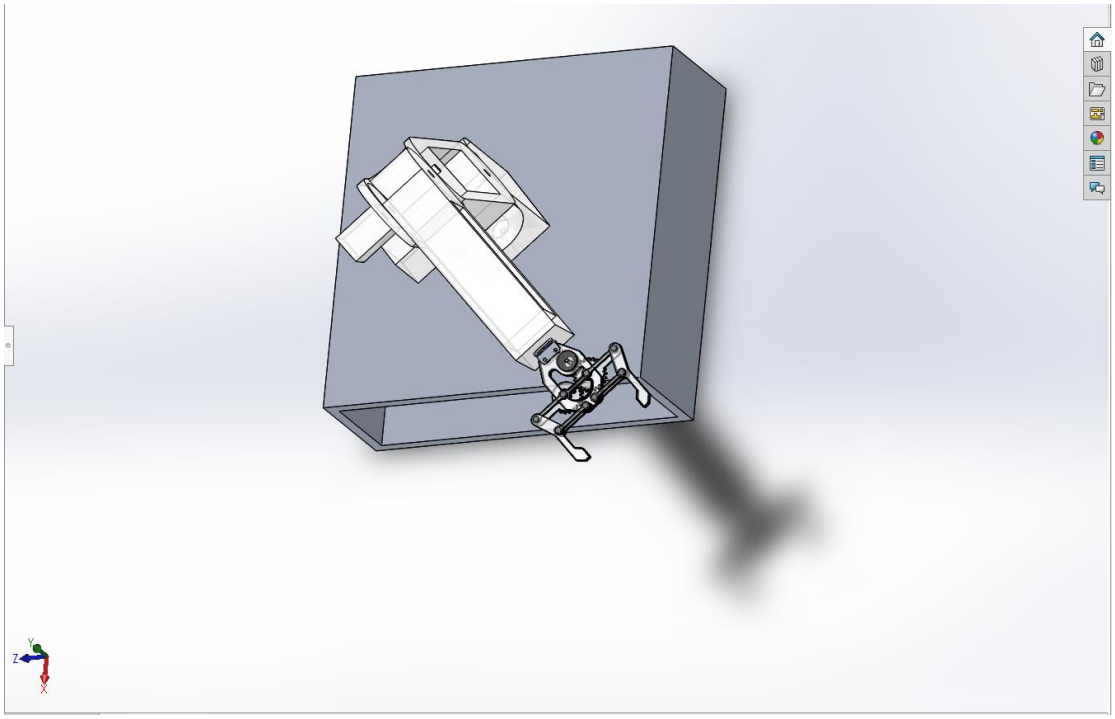

Fig 3.2: CAD model Top View of the System.

Fig 3.3: Frame

Fig 3.4: Gears

Fig 3.5: End Effector

Fig 3.6: Belt Drive

The robotic arm design is influenced by a number of variables such as geometry of the manipulator, dynamics involved, the structural characteristics of the linkage system (manipulator) and the actuator characteristics. The robotic arm resembles a human arm. The stationary part of the robot to which all other parts are attached is called its shoulder. The links are designed to be slender members to reduce its weight which is crucial in reducing the power consumption during its operation. The joints are simple revolute.

The hand of robot carry end-effector (not shown here), might be any tool or gripping mechanism. The design of end-effector is crucial to the satisfactory performance of the robotic arm and hence its design is dependent on the shape, size and weight of the object to be gripped.

Fig 3.7: Assembled Robot Front View

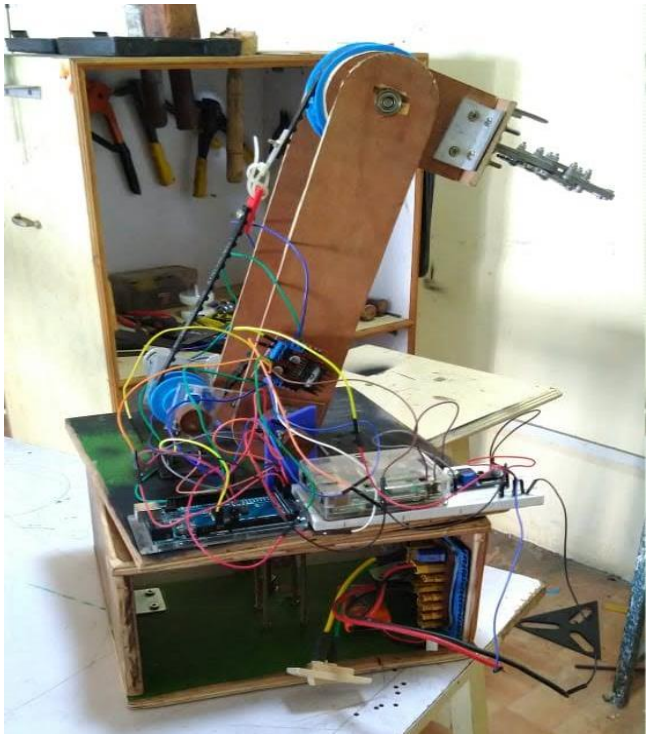

Fig 3.8: Assembled Robot Side View

CHAPTER 4

ELECTRONICS CONTROL AND CODING

4.1 COMPONENTS SELECTION

Table 4.1: Electronic Components and Specification

Sr No.	Components	Specification
1	ESP 32	
2	Arduino Mega	54 digital i/o pins,16 analog pins
3	L298N Motor driver	12V,2A
4	DC motor	
5	Servo Motor	
6	Battery	12V,7Amp hr.
7	Power Board	
8	Potentiometer	
9	Jumper Wires	

The electrical motors were chosen of about 12V with 22kg-cm torque value along with a 12V 2kg-cm motor. Motor drivers were selected based on the compatibility of the motors with it. Vision sensor is used for obtaining live feedback for executing the required task.

4.2 ESP 8266

The **ESP8266** is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by manufacturer Espressif Systems in Shanghai, China.

The chip first came to the attention of Western makers in August 2014 with the **ESP-01** module, made by a third-party manufacturer Ai-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands. However, at first there was almost no English-language documentation on the chip and the commands it accepted. The very low price and the fact that there were very few external components on the module, which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, chip, and the software on it, as well as to translate the Chinese documentation.

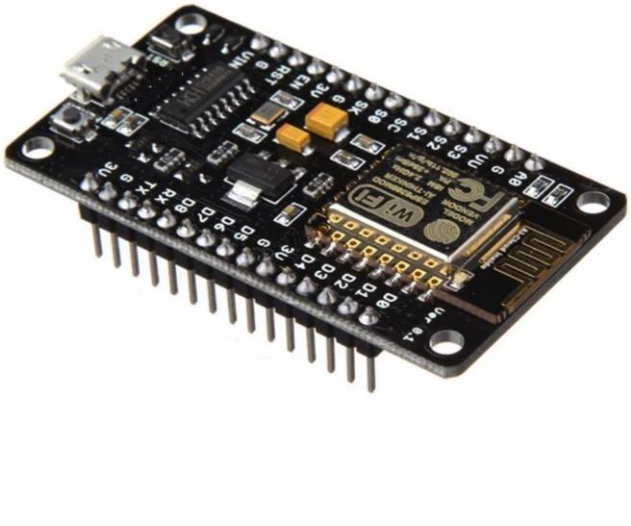

Fig 4.1: ESP 8266

4.3 ARDUINO MEGA

Arduino mega 2560 is the main controller for our project containing 54 digital I/O pins out of which 14 can be used as pwm pins and 16 analog pins 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

Features:

- Clock Speed 16MHz
- SRAM 8KB
- EEPROM 4KB
- Flash Memory 256 KB of which 8 KB used by bootloader

In our project we are connecting the two motors which is supposed to be driving our wheels pwm pins are connected to 3^{rd} and 5^{th} pin and direction pins of both the motors are connected to 9^{th} and 6^{th} pin. Motor for the brushes whose pwm pins are connected to 11^{th} and 13^{th} pin and direction pins are connected to 24^{th},26^{th},$32^{nd}$and 34^{th} pins respectively. Ultrasonic sensor (front and back) trigger pins are connected to 4^{th} and 10^{th} pin and echo pins are connected to 7^{th} and 8^{th}.

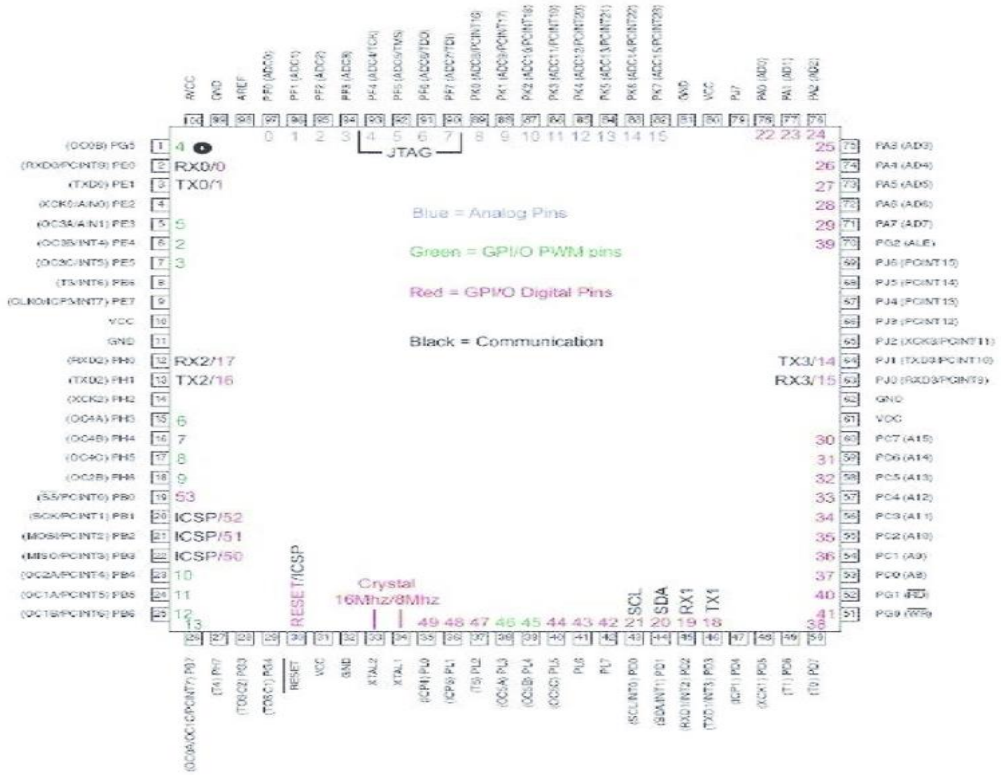

Fig 4.2: Arduino Mega Pin Diagram

4.4 L298N MOTOR DRIVER

It's an h-bridge driver which is used in our project to control the brushes of the motor. This allows you to control the speed and direction of two DC motors, or control one bipolar stepper motor with ease. The L298N H-bridge module can be used with motors that have a voltage of between 5 and 35V DC. There is also an onboard 5V regulator, so if your supply voltage is up to 12V you can also source 5V from the board.

This 5V is used to power up the Arduino in our project.

Consider the following fig 4.6 - match the numbers against the list below the image:

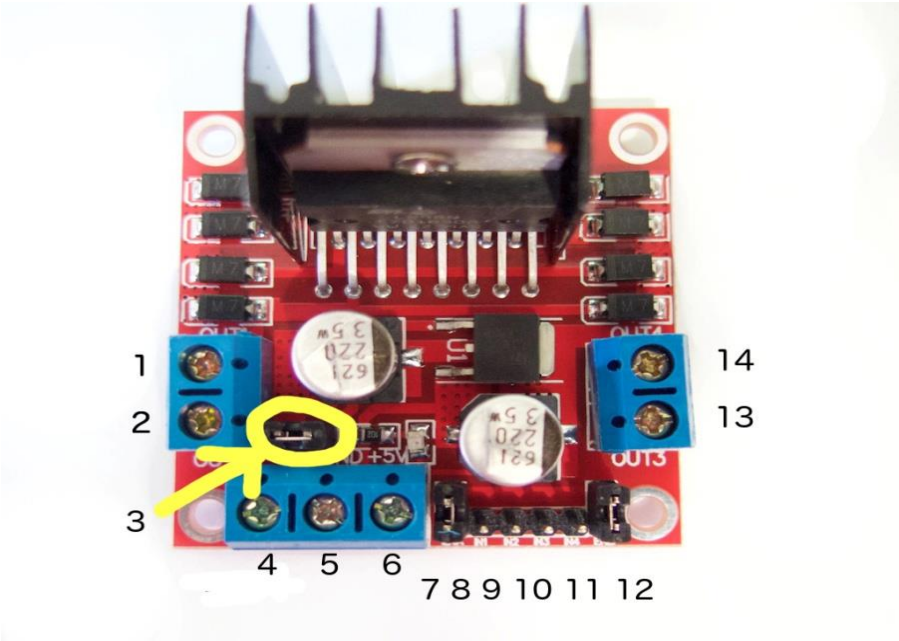

Fig 4.3: L298n Motor Driver 12V

1. DC motor 1 "+" or stepper motor A+
2. DC motor 1 "-" or stepper motor A-
3. 12V jumper - remove this if using a supply voltage greater than 12V DC. This enables power to the onboard 5V regulator
4. Connect your motor supply voltage here, maximum of 35V DC. Remove 12V jumper if >12V DC
5. GND
6. 5V output if 12V jumper in place, ideal for powering your Arduino (etc)
7. DC motor 1 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
8. IN1
9. IN2
10. IN3

11. IN4

12. DC motor 2 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.

13. DC motor 2 "+" or stepper motor B+

14. DC motor 2 "-" or stepper motor B-

To control one or two DC motors is quite easy. First connect each motor to the A and B connections on the L298N module. If we are using two motors for a robot (etc) ensured that the polarity of the motors is the same on both inputs. Otherwise we may need to swap them over when we set both motors to forward and one goes backwards!

Next, we connected our power supply - the positive to pin 4 on the module and negative/GND to pin 5. If our supply is up to 12V we can leave in the 12V jumper (point 3 in the image above) and 5V will be available from pin 6 on the module.

This can be fed to our Arduino's 5V pin to power it from the motors' power supply. We didn’t forget to connect Arduino GND to pin 5 on the module as well to complete the circuit. Now we will need six digital output pins on your Arduino, two of which need to be PWM (pulse-width modulation) pins. PWM pins are denoted by the tilde ("~") next to the pin number.

4.5 PROGRAM FLOW CHART

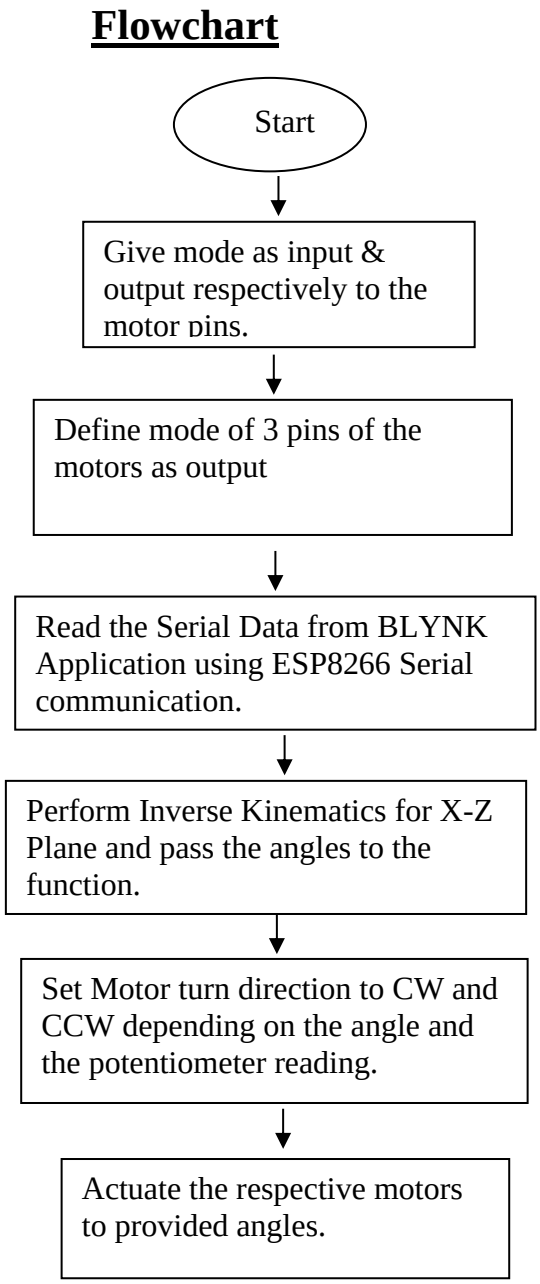

Fig 4.4: Flowchart of Uploaded ARDUINO MEGA Code

4.6 COMPLETE ASSEMBLY OF WIRELESSLY CONTROLLED SERIAL MANIPULATOR.

The system was completely assembled by placing all electrical and electronic components in place. Circuit connections were done using wires and jumpers. The sensors and motor drivers were placed on the mount and frame respectively.

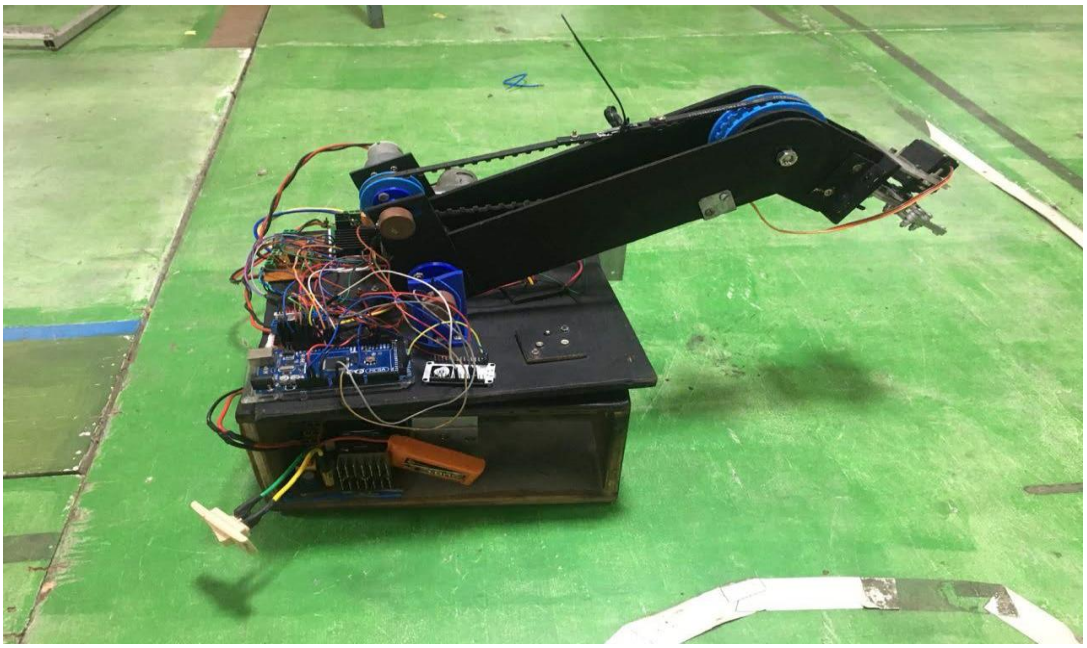

Fig 4.5: Complete Assembly of the Serial Manipulator

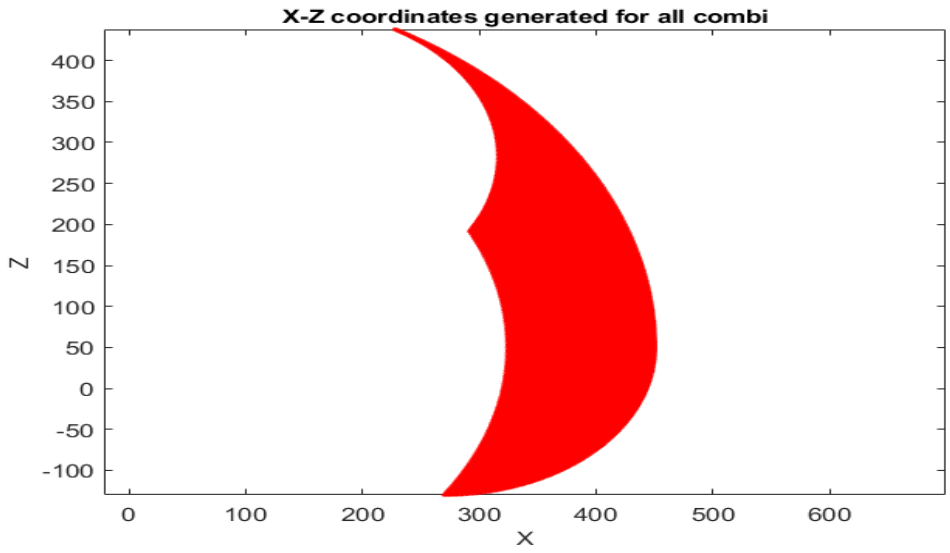

Fig 4.6: Workspace of the Serial Manipulator

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

The assembly of the system was carried out efficiently and precision was given utmost importance. We ran the code and an inference was drawn from the results. The code ran perfectly without any error. The mechanical system responded to the code and the potentiometer and camera sensors that were used, provided good response time. Our hypothesis was that the system will respond to the code and after successful testing sessions we can say that our results do support our hypothesis. I think the tests went smoothly and we had no problems, except for the fact that we had to adjust the position of joint motors accordingly in order to provide more speed and efficient and synchronous movement of the system.

The system that we have made is a working prototype which demonstrates our mechanical design along with its durability and feasibility. Further future study in this project might involve positioning the serial manipulator on mobile platform and using the vision sensor to map its positioning and move around a working environment. Further advancement can be to make this system voice controlled, where the system responds to the individual voice commands given by the users.

CHAPTER 6

REFERENCES

[1] Mohd Ashiq Kamaril Yusoff, Reza Ezuan Samin, Babul Salam Kader Ibrahim "Wireless Mobile Robotic Arm" Procedia Engineering, Volume 41, 2012, Pages 1072-1078.

[2] Minca, E., Iliescu,A., Voda,A., "Modelling and Control of an assembly/disassembly mechatronics line served by mobile robot with manipulator" , Control Engineering Practice, Volume 31, October 2014, Pages 50-62.

[3] R.Correal., A.Jordan., A.Gimenez., C.Balaguer ., "Wireless Teleportation of an Assistive Robot by PDA"., IFAC Telematics Applications in Automation and Robotics, Espoo, Finland, 2004, Pages 105-110.

[4] How to Use OV7670 Camera Module with Arduino
https://circuitdigest.com/microcontroller-projects/how-to-use-ov7670-camera-module-with-arduino

[5] L298n Motor Driver
https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/

APPENDIX 1: ARDUINO CODE

```
#include <math.h>
#include <Servo.h>
#include <String.h>
String content;
String chk, x1, y1, a1;
int xx,yy,aa;
char character;
double theta1, theta2;
int theta0;
void setup(){
Serial.begin(9600);
Serial1.begin(9600);
delay(2000);
}
int potPin0=A5;
int potPin1=A1;
int potPin2=A3;
//ANGULAR CONSTANTS
#define POT_VALUE_MAX0 900
#define POT_VALUE_MIN0 580
#define PERM_ERROR0 5
#define MAX_ANGLE0 80
#define POT_VALUE_MAX1 520
#define POT_VALUE_MIN1 110
#define PERM_ERROR1 5
#define MAX_ANGLE1 100
#define POT_VALUE_MAX2 830

#define POT_VALUE_MIN2 200


#define PERM_ERROR2 5
#define MAX_ANGLE2 90
class DCMotor{
private:
int M_pin10, M_pin20, M_PWMPin0, M_pin11, M_pin21, M_PWMPin1,
M_pin12, M_pin22, M_PWMPin2;
int M_Speed0, M_Speed1, M_Speed2;
int turnDirection0;
int turnDirection1;
int turnDirection2;
enum turnDirection0{right0, left0};
enum turnDirection1{right1, left1};
enum turnDirection2{right2, left2};
public:
DCMotor(int p1, int p2, int p3, int p4, int p5, int p6, int
p7, int p8, int p9)
```

```
{
M_pin10=p1;
M_pin20=p2;
M_PWMPin0=p3;
M_pin11=p4;
M_pin21=p5;
M_PWMPin1=p6;
M_pin12=p7;
M_pin22=p8;
M_PWMPin2=p9;
pinMode(M_pin10,OUTPUT);
pinMode(M_pin20,OUTPUT);
pinMode(M_PWMPin0,OUTPUT);
pinMode(M_pin11,OUTPUT);

pinMode(M_pin21,OUTPUT);


pinMode(M_PWMPin1,OUTPUT);


pinMode(M_pin12,OUTPUT);


pinMode(M_pin22,OUTPUT);
pinMode(M_PWMPin2,OUTPUT);
}
void SetTurnDirection0(int dir0)
{
turnDirection0=dir0;
switch(turnDirection0)
{
case right0:
digitalWrite(M_pin10,HIGH);
digitalWrite(M_pin20,LOW);
break;
case left0:
digitalWrite(M_pin10,LOW);
digitalWrite(M_pin20,HIGH);
break;
}
}
void SetTurnDirection1(int dir1)
{
turnDirection1=dir1;
switch(turnDirection1)
{
case right1:
digitalWrite(M_pin11,HIGH);
digitalWrite(M_pin21,LOW);
break;
```

```
case left1:
digitalWrite(M_pin11,LOW);
digitalWrite(M_pin21,HIGH);
break;
}
}
void SetTurnDirection2(int dir2)
{
turnDirection2=dir2;
switch(turnDirection2)
{
case right2:
digitalWrite(M_pin12,HIGH);
digitalWrite(M_pin22,LOW);
break;
case left2://turning Left
//motor moves CCW
digitalWrite(M_pin12,LOW);
digitalWrite(M_pin22,HIGH);
break;
}
}
void SetTurnSpeed0(int s0)
{
M_Speed0=s0;
}
void SetTurnSpeed1(int s1)
{
M_Speed1=s1;
}
void SetTurnSpeed2(int s2)
{
M_Speed2=s2;
}
void Turn0()
{
analogWrite(M_PWMPin0,M_Speed0);
}
void Turn1()
{
analogWrite(M_PWMPin1,M_Speed1);
}
void Turn2()
{
analogWrite(M_PWMPin2,M_Speed2);
}
void Stop0()
```

```
{
analogWrite(M_PWMPin0,0);
}
void Stop1()
{
analogWrite(M_PWMPin1,0);
}
void Stop2()
{
analogWrite(M_PWMPin2,0);
}
void GoToAngle(double target0, int howFast0, double target1,
int howFast1, double target2, int howFast2)
{
int
currentAngle0=((float)analogRead(potPin0)-POT_VALUE_MIN0)/(POT
_VALUE_MAX0-POT_VALUE_MIN0)*MAX_ANGLE0;
int
currentAngle1=((float)analogRead(potPin1)-POT_VALUE_MIN1)/(POT
_VALUE_MAX1-POT_VALUE_MIN1)*MAX_ANGLE1;
int
currentAngle2=((float)analogRead(potPin2)-POT_VALUE_MIN2)/(POT
_VALUE_MAX2-POT_VALUE_MIN2)*MAX_ANGLE2;
if(currentAngle0<target0)
{SetTurnDirection0(right0); }
else if(currentAngle0>target0)
{SetTurnDirection0(left0); }
if(currentAngle1<target1)
{SetTurnDirection1(right1); }
else if(currentAngle1>target1)
{SetTurnDirection1(left1); }
if(currentAngle2<target2)
{SetTurnDirection2(right2); }
else if(currentAngle2>target2)
{SetTurnDirection2(left2); }
SetTurnSpeed0(howFast0);
while(abs(currentAngle0-target0)>=PERM_ERROR0)
{ Turn0();
delay(1);
currentAngle0=((float)analogRead(potPin0)-POT_VALUE_MIN0)/(POT
_VALUE_MAX0-POT_VALUE_MIN0)*MAX_ANGLE0;
}
Stop0();
delay(100);
SetTurnSpeed1(howFast1);
while(abs(currentAngle1-target1)>=PERM_ERROR1)
{ Turn1();
delay(1);
```

```
currentAngle1=((float)analogRead(potPin1)-POT_VALUE_MIN1)/(POT
_VALUE_MAX1-POT_VALUE_MIN1)*MAX_ANGLE1;
}
Stop1();
delay(100);
SetTurnSpeed2(howFast2);
while(abs(currentAngle2-target2)>=PERM_ERROR2)
{
Turn2();
delay(1);
currentAngle2=((float)analogRead(potPin2)-POT_VALUE_MIN2)/(POT
_VALUE_MAX2-POT_VALUE_MIN2)*MAX_ANGLE2;
}
Stop2();
delay(100);
}
};
int motor_p10=13;
int motor_p20=12;
int pwmPin0=11;
int motor_p11=4;
int motor_p21=7;
int pwmPin1=6;
int motor_p12=9;
int motor_p22=8;
int pwmPin2=10;
DCMotor customServo(motor_p10, motor_p20, pwmPin0, motor_p11,
motor_p21, pwmPin1, motor_p12, motor_p22, pwmPin2);
void loop(){
while(Serial1.available()) {
character = char(Serial1.read());
content.concat(character);
if(character=="!"){
content="";
break;
}
delay(100);
}
chk = content[0];
if (chk == "x"){
x1 = content.substring(2);
xx = x1.toDouble();
Serial.println(xx);
}
else if(chk == "y"){
y1 = content.substring(2);
yy = y1.toDouble();
Serial.println(yy);
```

```
}
else if(chk == "a"){
a1 = content.substring(2);
aa = a1.toDouble();
Serial.println(aa);
}
content = "";
double a1 = 270;
double a2 = 180;
double x = xx;
double y = yy;
double th2d = acos((pow(x,2) + pow(y,2) - pow(a1,2) - pow(a2,
2))/(2*a1*a2));
double th2 = M_PI/2 - th2d;
double th1 = atan(y/x) + atan((a2*sin(th2d))/(a1 +
a2*cos(th2d)));
theta0 = aa;
theta1 = th1*180/M_PI;
theta2 = th2*180/M_PI;
Serial.println(theta1);
Serial.println(theta2);
customServo.GoToAngle(theta0 , 230, theta1, 50, theta2, 80);
delay(100);
```

APPENDIX 1: ESP8266 CODE

```
#define BLYNK_PRINT Serial
#include <Servo.h>
Servo myservo;
int pos;
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "F3mML5opX5dvKukNeyW6xO359yZVvL8D";
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "harsh";
char pass[] = "abcd@1234";
// This function will be called every time Slider Widget
// in Blynk app writes values to the Virtual Pin 1
BLYNK_WRITE(V0)
{
int x = param.asInt();
Serial.print("x:");
Serial.print(x);
Serial.println("!");
}
BLYNK_WRITE(V1)
{
int y = param.asInt();
Serial.print("y:");
Serial.print(y);
Serial.println("!");
}
BLYNK_WRITE(V2)
{
int z = param.asInt();
Serial.print("a:");
Serial.print(z);
Serial.println("!");
}
BLYNK_WRITE(V3)
{
int pos = param.asInt();
myservo.attach(14);
myservo.write(pos);
}
void setup()
{
Serial.begin(9600);
Blynk.begin(auth, ssid, pass);
}
void loop()
{
Blynk.run();
}
```