FABRICATION OF THREE DEGREES OF FREEDOM ROBOTIC ARM

A Project report
Submitted in the partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology

in

Mechanical Engineering

By

I.RAJESH	186N1A0323
G.HARI TEJA	186N1A0316
A.S.NETHI KONDA	186N1A0302
B.RAKESH	186N1A0306
N.VASU	186N1A0337

Under the Guidance of

Mr. D.PAVAN KUMAR

Professor

SRINIVASA INSTITUTE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, New Delhi, Permanently affiliated to JNTUK, Kakinada)
(An ISO 9001:2015 Certified Institute, Accredited by NAAC with 'A' Grade)
(Recognized by UGC under sections 2(f) & 12(B))
NH-216, Amalapuram-Kakinada Highway, Cheyyeru (V), AMALAPURAM, E.G.Dt.-533216

2018-2022

SRINIVASA INSTITUTE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, New Delhi, Permanently affiliated to JNTUK, Kakinada)
(An ISO 9001:2015 Certified Institute, Accredited by NAAC with 'A' Grade)
(Recognized by UGC under sections 2(f) & 12(B))
NH-216, Amalapuram-Kakinada Highway, Cheyyeru (V), AMALAPURAM, E.G.Dt.-533216

Department of Mechanical Engineering

COLLEGE VISION

To develop the institute into a world class destination for technological education and research.

COLLEGE MISSION

- To impart high quality, industry relevant, career oriented, engineering education to rural students, to translate our vision into a reality.
- To provide the best of instructional and institutional infrastructure facilities.
- To have strategic linkages with industry and other institutions.
- To mould students to meet the challenges of life with ethics, courage and conviction.

SRINIVASA INSTITUTE OF ENGINEERING &TECHNOLOGY

(Approved by AICTE, New Delhi, Permanently affiliated to JNTUK,Kakinada)
(An ISO 9001:2015 Certified Institute, Accredited by NAAC with 'A' Grade)
(Recognized by UGC under sections 2(f) & 12(B))
NH-216, Amalapuram-Kakinada Highway, Cheyyeru (V), AMALAPURAM, E.G.Dt.-533216

Department of Mechanical Engineering

CERTIFICATE

This is to certify that the project entitled **"FABRICATION OF THREE DEGREES OF FREEDOM ROBOTIC ARM"** is being submitted by I.RAJESH (186N1A0323), G.HARI TEJA (186N1A0316), A.S.NETHI KONDA (186N1A0302), B.RAKESH (186N1A0306), N.VASU (186N1A0337) in partial fulfillment for the award of Bachelor of Technology in Mechanical Engineering to the Jawaharlal Nehru Technological University Kakinada is a record of bonafide work carried out by them under our guidance.

Mr.D.PAVAN KUMAR
PROJECT GUIDE

Mrs.CH.L.ASLESHA
HEAD ME

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

The Thesis entitled **"FABRICATION OF THREE DEGREES OF FREEDOM ROBOTIC ARM"** is a record of bonafide work carried out by me, submitted in partial fulfillment for the award of B.Tech in Mechanical Engineering to the Jawaharlal Nehru Technological University Kakinada.

Registration Numbers of the Candidates

186N1A0323
186N1A0316
186N1A0302
186N1A0306
186N1A0337

ACKNOWLEDGEMENT

There are very few people, whom we are always indebted for the completion of this work. It’s a dream coming true for us without their help it would have been impossible to complete this work. There are several persons who made us to realize this dream.

We wish to thank our project guide **Mr.D.PAVAN KUMAR**, Srinivasa Institute of Engineering and Technology for providing me the opportunity to do this work. Special thanks for his wise supervision, guidance, patience, and friendly encouragement.

We convey my special thanks to **Mrs. Ch.L. Aslesha**, H.O.D of Mechanical Engineering, for providing all facilities for carrying out this work.

We would also like to thank “**Dr. N. Seshaiah**” principal, for providing for excellent facilities & motivation to carry out this work.

We are very much thankful to Management of SRINIVASA INSTITUTE OF ENGINEERING AND TECHNOLOGY for providing various resources to complete this work

186N1A0323 I. RAJESH
186N1A0316 G. HARI TEJA
186N1A0302 A.S. NETHI KONDA
186N1A0306 B. RAKESH
186N1A0337 N. VASU

FABRICATION OF THREE DEGREES OF FREEDOM ROBOTIC ARM

Objectives:

1. The main objective of this project is the design and development automated gripper control for robotic arm.
2. Robotic arm with Bluetooth control.
3. Pick and place the objects with Arm with 180 Degrees rotations.
4. This project focuses to create and build more compact, useful and cheaper robotic arm to perform various functions where human is proven too dangerous to perform a specific task and also to eliminate human errors to get more precise work

Outcomes:

1. Design, Fabricated and Assembly done for Gripper control robotic arm.
2. Bluetooth app based contolled the moments.
3. 180 Degrees rotation both arm and base also
4. Assembled working model with various functions used in the recent trends

ABSTRACT

This paper discusses the design and fabrication of a pick and place robotic arm. The robotic arm is intended for educational purposes. In this project we are designing the robotic arm for improved accuracy by using servos to power the joints in the robotic arm. We are designing the robotic arm using Pro/Engineer software. In this project we are going to fabricate robotic arm which performs the pick and place operation. The project covers the procedure for selection of the servos used to power each joint of the arm in details. We are selecting plastic to fabricate the components of the robotic arm. Selecting a suitable servo controller and control software for the Robotic arm is developing using Microsoft's C programming language.

Table of Contents

LIST OF FIGURES

1 CHAPTER-1
INTRODUCTION

The first usage of the word ‘robot’ was in a 1921 Czech science fiction play – ‘Rossum’s Universal Robots’ – by Karel Capek. The robots were artificial people or androids and the word was derived from the word ‘Robata’, a Czech word for slave. A question of perpetual interest is to define a robot. Since the beginning of the study of robotics, there has been some controversy in the definition of a robot. So long as the evolution of robotics continues, the definition of the robot will change from time to time, depending on the technological advances in its sensory capability and level of intelligence. However, the most widely accepted definition of a robot was given by the Robotic Institute of America (RIA) in 1979.

Robotic manipulators resembling the human arm is known as robotic arms. They are constructed by a structure consisting of structurally robust links coupled by either rotational joints or translating joints. A robotic arm is thus a type of mechanically coupled or joined arm, run by programmable commands, with similar functions to a human arm. It may be the sum total of the mechanism links or may be part of a more complex sized robot.

1.1 Robotic Arm Components

A typical robotic arm has the following components:

- Links and joints
- Actuators
- Controller
- End-effector

1.1.1 Links and Joints

A link is considered as a rigid body that defines the relationship between two corresponding joint axes of a manipulator. Manipulators consist of rigid links, which are connected by joints that allow relative motion of corresponding links. The links move to position with the end-effector.

1.1.2 Actuators

Electric motor-driven actuators perform smoother movements, can be controlled very accurately, and are very reliable. However, these actuators cannot deliver as much power as hydraulic actuators of comparable mass. Nevertheless, for modest power actuator functions, electrical actuators are often preferred. The various types of electric motors used as actuators for robotic applications are direct current (DC) motors, stepper motors and servo motors.

1.1.3 Controller

The controller is the main part that processes information and carries out instructions in a robot. It is the robot's 'brain' and controls the robot's movements. It is usually a computer of some type which is used to keep information about the robot and the working process and execute programs which operate the robot. It contains programs, data algorithms, logic analysis and various other processing activities which enable the robot to perform its intended function.

1.1.4 End-effectors

End-effector is a device at the end of a robotic arm, designed to interact with the open world. The exact nature of performance of this device depends on the application of the robot. Typical functions of the end-effector consist grasping, pushing and pulling, twisting, using tools, performing insertions, welding and various types of assembly activities. Thus, the major types of robot end-effectors are:

Grippers: Grippers are the most commonly used type of end-effectors. They can use different gripping methods (such as vacuum or use of fingers). Material removal tools: These include cutting, drilling and deburring tools installed as robot tools.

1.2 Robotic Application

1.2.1 Welding torches

Welding is a widely using operation in robotic application. Welding torches have become very efficient end-effectors that can be controlled in a sophisticated way for optimized welding.

1.2.2 Tool changers

Tool changers are used when many different end effectors need to be used in sequence by one robot.

They are used to standardize the interface between the robot flange and the base of the tool. They can be manual or automatic Application of Robotic Arm The aim of the present work is to design and fabricating a robotic arm that can be used for demonstrating and educational purposes. It will be used to pick and place things which are having special properties or handle with care objects. Methodology This work covers material selection, design, programming and fabrication of a basic robotic arm system. It also covers the implementation of the kinematics of the arm but does not consider the details of the derivation of the kinematic equations

- Select a suitable material for the fabrication of a 2-DoF robotic arm
- Obtain suitable design parameters for the robotic arm
- Create a 3-d model of a robotic arm based on the design parameters
- Fabricate the robotic arm
- Calculate the torque required
- Calculate the power required

The availability of a robotic arm that can be used for demonstrating and educational purposes in the Department of Mechanical Engineering will go a broad way in stimulating the interest of students in robotics. It will provide a tool to use for learning and experimenting with robotics. Students with a flair for programming can reprogram the robot to adapt it to different tasks. In this chapter covers the detailed design and method of construction of the robotic arm and its controller. Choice of Material The following material properties were put into consideration during the material selection process: strength, lightness, availability and Ease of cutting. The material should possess sufficient strength so as to ensure that each link of the arm is able to bear the load imposed on it by motors, other attached links and the payload. Lightness of the material minimizes the torque requirement of the robotic actuators, thereby minimizing cost. The material also had to be readily available and easy to cut because the fabrication of some parts of a robotic arm involved the cutting of intricate shapes. Aluminium thick 5mm plates are satisfying all the above criteria and was thus selected for the project. It can be easily cut with readily available hand tools such as a cutting and hand drilling machines. In addition to this, it offers outstanding strength and stiffness. Which made it suitable for the project. It is also cheap and readily available. With density of 2.7 g/cm3, its lightness is a key advantage for any demonstrative or educational robotics project, because it minimizing the torque requirement of the robot actuators. Robotic Arm Structure As earlier stated in the introductory chapter of this report, the robotic arm has degrees of freedom with the inclusion of the gripper.

1.3 Kinematics

Kinematics is the branch of mechanics that studies the system of bodies without giving consideration to its mass or motion of a body or the forces acting on it. The kinematics of the robotic arm can be studied by two categories:

- Forward kinematics
- Inverse kinematics Forward Kinematics

The goal of the kinematic analysis of a robotic arm is to develop equations that can be used to determine the position of the end-effector given any set of joint values. The equations developed relate the end-effector position to the joint variables of the robotic arm. Forward kinematics refers to the use of the kinematic equations of the robotic arm to compute the position of the end-effector from specified values for the joint parameters. Inverse Kinematics A method of determining the pose of the end-effector of the robotic arm, given the joint coordinates has been discussed. “A problem of real practical need is the inverse problem: given the desired position of the end-effector, what are the required joint coordinates? For example, if we know the Cartesian pose of an object, what joint coordinates does the robot need in order to reach it.

1.4 Robotic Structures

1.4.1 Structure

The structure of a robot is usually mostly mechanical and can be called a kinematic chain. The chain is formed of links, actuators, and joints which can allow one or more degrees of freedom. Most contemporary robots use open serial chains in which each link connects the one before to the one after it. These robots are called serial robots and often resemble the human arm. Robots used as manipulators have an end effector mounted on the last link. This end effector can be anything from a welding device to a mechanical hand used to manipulate the environment.

1.4.2 Power source

At present mostly (lead-acid) batteries are used, but potential power sources could be:

- Image motor
- Pneumatic (compressed gases)
- Hydraulics (compressed liquids)
- Flywheel energy storage
- Organic garbage (through anaerobic digestion)
- Still untested energy sources (e.g., Nuclear Fusion reactors)

1.4.3 Actuation

Actuators are like the "muscles" of a robot, the parts which convert stored energy into movement. By far the most popular actuators are electric motors that spin a wheel or gear, and linear actuators that control industrial robots in factors. But there are some recent advances in alternative types of actuators, powered by electricity, chemicals, or compressed air.

1.4.4 Touch

Current robotic and prosthetic hands receive far less tactile information than the human hand. Recent research has developed a tactile sensor array that mimics the mechanical properties and touch receptors of human fingertips. The sensor array is constructed as a rigid core surrounded by conductive fluid contained by an elastomeric skin. Electrodes are mounted on the surface of the rigid core and are connected to an impedance-measuring device within the core. When the artificial skin touches an object the fluid path around the electrodes is deformed, producing impedance changes hat map the forces received from the object.

1.4.5 Vision

Computer vision is the science and technology of machines that see. As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences and views from cameras. In most practical computer vision applications, the computers are pre-programmed to solve a particular task, but methods based on learning are now becoming increasingly common. Computer vision systems rely on image sensors which detect electromagnetic radiation which is typically in the form of either visible light or infra-red light. The sensors are designed using solid-state physics. The process by which light propagates and reflects off surfaces is explained using optics. Sophisticated image sensors even require quantum mechanics to provide a complete understanding of the image formation process.

1.4.6 Manipulation

Robots which must work in the real world require some way to manipulate objects; pick up, modify, destroy, or otherwise have an effect. Thus the 'hands' of a robot are often referred to as end effectors, while the arm is referred to as a manipulator. Most robot arms have replaceable effectors, each allowing them to perform some small range of tasks. Some have a fixed manipulator which cannot be replaced, while a few have one very general-purpose manipulator, for example a humanoid hand. Mechanical Grippers: One of the most common effectors is the gripper. In its simplest manifestation it consists of just two fingers which can open and close to pick up and let go of a range of small objects. Fingers can for example be made of a chain with a metal wire run through it. Vacuum Grippers: Pick and place robots for electronic components and for large objects like car windscreens, will often use very simple vacuum grippers. These are very simple astrictive devices, but can hold very large loads provided the pretension surface is smooth enough to ensure suction.

2 CHAPTER-2

2.1 LITERATURE REVIEW

1. Kaushik Phasale, Praveen Kumar, Akshay Raut, Ravi Ranjan Singh, Amit Nichat, Design, Manufacturing and Analysis of Robotic Arm with SCARA Configuration.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 05 Issue: 04 | Apr-2018

This paper deals with the "Design, Manufacturing and Analysis of Robotic Arm with SCARA Configuration". In the modern world, robotics has become popular, useful and has achieved great successes in several fields of humanity. Every industrialist cannot afford to transform his unit from manual to semi-automatic or fully automatic as automation is not that cheap in India. The basic objective of this project is to develop a versatile and low cost robotic arm which can be utilized for Pick and Place operation. Here controlling of the robot has been done by using servo drives and Arduino microcontroller. This robot is having 4 DOF and controlled by android app with Bluetooth interfacing. This Robotic arm can be used in number of applications by changing the program of controller and end effector, so that it would be used mainly in the automatic assembly lines.

Key Words: SCARA, Degree of Freedom (DOF), Arduino, Bluetooth Interfacing, Design, Manufacturing, Analysis.

2. 2. J.C. García Prada – M. Ceccarelli

A Multi-Objective Optimization of a Robotic Arm for Service Tasks.

In this paper, the main characteristics and peculiarities of service tasks are investigated in order to propose the design criteria in the form of computationally efficient objective functions. Then, these design criteria can be implemented in a multi-objective optimization algorithm by using ever commercial packages in order to obtain optimal design solutions for service robots. The proposed procedure has been applied to a robotic arm for service tasks. It is composed of two modules with four degrees of freedom (dofs). The main module with two dofs is called SIDEMAR (Spanish acronym of Mechatronic Design of an Integrated System for Service Robots). It has been designed and built at the Carlos III University of Madrid as a module for service robot applications. Results of the above-mentioned case study are reported to show both the soundness and engineering feasibility of the proposed multi-objective optimal design procedure. ©2010 Journal of Mechanical Engineering. All rights reserved. Keywords: robotics, robot design, service robots, optimal design

Strojniški vestnik - Journal of Mechanical Engineering 56(2010)5, 316-329 UDC 007.52

3. G Krishna Teja, K S Sandesh, K Saichand, V Kalyan

Modeling, Fabrication & Control of an Articulated Robotic Arm

Abstract: Modelling and Fabrication of robotic systems and their control for pick &place and maintenance tasks is highly complex activity involving coordination of various sub-systems. The entire design has four important modules:

i. CAD Modelling
ii. Control System Design
iii. Machine Vision and Image Processing
iv. Hardware Development and Testing

The five- axis articulated manipulator equipped with a vision camera in eye-to-hand configuration is designed for performing the pick and place operations of the defected tiles in a systematic manner. Dynamics of manipulator is required for design of model-based controllers. Interactive programs are developed in Matlab for kinematics and dynamics. Three-dimensional manipulator assembly configuration is modeled in Pro-E software. Motion analysis is conducted in Arduino software in order to compare the results obtained from the classical kinematics. The test set-up is developed using vision camera and microcontroller platform to guide the robot joint servos so as to perform defected object replacement activity. Presences of the coordinate of the region are indicated with the use of image-processing operations. Keywords: Arduino software, Motion analysis, Image Processing.

https://www.ijitee.org/wp-content/uploads/papers/v9i9/I7221079920.pdf

4. Veena C D, Sharath H K, Sree Rajendra, Shivashankara B S

Design and fabrication of plc and scada based robotic arm for material handling

Abstract - Picking and placing the object from the conveyor belt is the important task in the packing section of an industry. Pick and place manually, needs manual power and time. This is an attempt to design efficient mechanism for picking and placing by automating them by constructing the 3directional robotic arm using Pneumatic cylinders which are controlled by the PLC. The system consists of a PLC which controls the movements of the pneumatic cylinders based on the inputs coming from the sensors placed on the conveyor belt. The robotic arm is having the capability to move along the three axis (X, Y and Z). A mechanical gripper is placed at the end of the robotic arm which is used for holding the objects on the conveyor belt. Key words: PLC, SCADA, etc

https://www.irjet.net/archives/V5/i6/IRJET-V5I670.pdf

5. Anusha Ronanki, M. Kranthi

Design and Fabrication of Pick and Place Robot to Be Used in Library

ABSTRACT: The use of robots in library is becoming more popular in recent years. The trend seems to continue as long as the robotics technology meets diverse and challenging needs in educational purpose. The prototype consists of robotic arm along with grippers capable of moving in the three axes and an ATMEGA 8 microcontroller. Software such as AVR Studio is used for programming, PROTESUS is used for simulation and PROGISP is used for dumping the program. RFID is used for identifying the books and it has two IR Sensors for detecting the path. This robot is about 4 kg in weight and it is capable of picking and placing a book of weight one kgs.

KEYWORDS: Design, Fabrication, robot, and library.

International Journal of Innovative Research in Science, Engineering and Technology

Vol. 4, Issue 6, June 2015.

http://www.ijirset.com/upload/2015/june/56_10_Design.pdf

3 CHAPTER-3

3.1 Methodology

The Following document helps us understand the working methodology of our project. Let's start assembling them and make a circuit to build a remote-controlled robotic arm.

- Attach the Arduino Nano board on the breadboard. The Arduino will be powered through the positive and negative wire of the adapter.
- Place the Bluetooth module on the breadboard too. Power up the Bluetooth module through Arduino. Connect the Tx pin of the Bluetooth module to the Rx pin of the Arduino Nano board and connect the Rx pin of the Bluetooth module to the Tx pin of the Arduino Nano board.
- As we know that there are 4 stepper motors. Each one has a technical name. They are called Elbow, Shoulder, Base, and Gripper. The Vcc and Ground of all the motors will be common and connected to the positive and negative of the 6V adapter. The Signal pin of all the four motors will be connected to pin5, pin6, pin9, and pin11 of Arduino Nano.
- Make sure that the connections you have made are according to the following circuit diagram.

3.2 Getting Started with Arduino

- Download the latest version of Arduino IDE from Arduino.
- Connect your Arduino Nano board to your laptop and open the control panel. Then, click on Hardware and Sound. Now, click on Devices and Printers. Here, find the port to which your microcontroller board is connected. In my case it is COM14 but it is different on different computers.
- Click on the Tool menu and set the board to Arduino Nano from the drop-down menu.
- In the same Tool menu, set the port to the port number that you observed before in the Devices and Printers.
- In the same Tool menu, Set the Processor to ATmega328P (Old Bootloader)
- To write code to operate the servo motors, we need special library that will help us to write several functions for servo motors. This library is attached along with the code, in the link below. To include the library, click on Sketch > Include Library > Add ZIP. Library.
- Create the document and paste it into your Arduino IDE. Click on the upload button to burn the code on your microcontroller board.

3.3 Downloading the Application

As we have now assembled the whole circuitry and uploaded the code in the microcontroller board. let’s download a mobile app that will work as a remote control for the robotic arm. A free app is available on the google play store. The name of the app is the Little Arm Robot Control. To make a Bluetooth connection, turn on the Bluetooth on your mobile. Go to the settings and pair your mobile with the HC-05 module. After doing this, press the Bluetooth button in the app. If it turns green, it means that the app is now connected and ready to operate the robotic arm. There are sliders to set operate the robotic arm as desired.

3.4 Code working

1. In the start, A library is included to write a code to operate servo motors. Another library math.h is included to perform different mathematical operations in the code. Four objects are also initialized to be used for the four servo motors.

```
#include <Servo.h> //arduino library

#include <math.h> //standard c library


#define PI 3.141

Servo baseServo;

Servo shoulderServo;

Servo elbowServo;

Servo gripperServo;


int command;
```

2. Then a structure is declared to take values for the base, shoulder, and elbow servo motors.

```
struct jointAngle {// declaring a structure

int base;

int shoulder;

int elbow;

};
```

3. After this, some variables are initialized to store the desired grip, delay, and position of the servo motor. the speed is set to be 15, and an object is made to take the value of the angle in the structure.

```
int desiredGrip;
int gripperPos;
int desiredDelay;
int servoSpeed = 15;
int ready = 0;
struct jointAngle desiredAngle; //desired angles of the servos
```

4. Void setup () is a function that is used to set the pins of Arduino as INPUT or OUTPUT. Here in this function, we have declared that the pin of the motors will be connected to which pins of the Arduino. It is also ensured that the Arduino doesn't read serial input for too long. Initial Position and Baud Rate is also set in this function. Baud Rate is the speed by which the microcontroller board will communicate with the servos and Bluetooth module attached.

```
void setup()
{
Serial.begin(9600);
baseServo.attach(9); // attaches the base servo on pin 9 to the servo object
shoulderServo.attach(10); // attaches the shoulder servo on pin 9 to the servo object
elbowServo.attach(11);  // attaches the elbow servo on pin 9 to the servo object
gripperServo.attach(6);  // attaches the gripper servo on pin 9 to the servo object
Serial.setTimeout(50); //ensures the the arduino does not read serial for too long
Serial.println("started");
baseServo.write(90); //intial positions of servos
shoulderServo.write(150);
elbowServo.write(110);
ready = 0;
}
```

5. ServoParallelControl() is a function that is used to detect the current position of the robotic arm and move it according to the command given through the mobile app. If the current position is less than the actual, the arm will move up and vice versa. This function will return the value of the current position and the speed of the servo.

```
int servoParallelControl (int thePos, Servo theServo, int theSpeed )
{
 int startPos = theServo.read(); //read the current pos
 int newPos = startPos;
 //int theSpeed = speed;
 //define where the pos is with respect to the command
 // if the current position is less that the actual move up
 if (startPos < (thePos-5)){
  newPos = newPos + 1;
  theServo.write(newPos);
  delay(theSpeed);
  return 0;
  }
  else if (newPos > (thePos + 5)){
   newPos = newPos - 1;
   theServo.write(newPos);
   delay(theSpeed);
   return 0;
   }
   else {
    return 1;
   }
}
```

6. void loop() is a function that runs repeatedly in a loop. This function reads the data coming serially and store the angle of each servo in the structure. Initially, the status of all the servos is set to zero. Here a function servoParallelControl() is called and parameters are passed in it. this function will return the value and it will be stored in a variable of status.

```
void loop()
{
 if (Serial.available()){
 ready = 1;
 desiredAngle.base = Serial.parseInt();
 desiredAngle.shoulder = Serial.parseInt();
 desiredAngle.elbow = Serial.parseInt();
 desiredGrip = Serial.parseInt();
 desiredDelay = Serial.parseInt();
 if(Serial.read() == '\n'){
 Serial.flush(); //clear all other commands piled in the buffer
 //send completion of the command
 Serial.print('d');
 }
}

int status1 = 0;
int status2 = 0;
int status3 = 0;
int status4 = 0;
int done = 0 ;

while(done == 0 && ready == 1){
 //move the servo to the desired position
 status1 = servoParallelControl(desiredAngle.base, baseServo, desiredDelay);
 status2 = servoParallelControl(desiredAngle.shoulder, shoulderServo, desiredDelay);
```

```
status3 = servoParallelControl(desiredAngle.elbow, elbowServo, desiredDelay);
status4 = servoParallelControl(desiredGrip, gripperServo, desiredDelay);
 if (status1 == 1 & status2 == 1 & status3 == 1 & status4 == 1){
 done = 1
 }
}// end of while
}
```

Now, this was the whole procedure of making a robotic arm. After burning the code and downloading the app, the robot should work perfectly fine when the sliders on the app are moved. You can also program the arm to work autonomously to perform the desired task.

4 CHAPTER-4
COMPONENTS USED FOR ROBOTIC ARM

4.1 Arduino Nano

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.0) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech.

Schematic and Design

Arduino Nano 3.0 (ATmega328): schematic, Eagle files. Arduino Nano 2.3 (ATmega168): manual (pdf), Eagle files. Note: since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

Specifications

Microcontroller : Atmel ATmega168 or ATmega328

Operating Voltage (logic level) : 5 V

Input Voltage (recommended) : 7-12 V Input Voltage (limits) 6-20 V

Digital I/O Pins : 14 (of which 6 provide PWM output)

Analog Input Pins : 8 DC Current per I/O Pin 40 mA

Flash Memory :16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader SRAM 1 KB (ATmega168) or 2 KB (ATmega328) EEPROM 512 bytes (ATmega168) or 1 KB (ATmega328)

Clock Speed : 16 MHz

Dimensions : 0.73" x 1.70"

4.1.1 Power

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source. The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

4.1.2 Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

4.1.3 Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode() , digitalWrite() , and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions: Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details. PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Additionally, some pins have specialized functionality: I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library the Wiring website). There are a couple of other pins on the board: AREF. Reference voltage for the analog inputs. Used with analogReference (). Documentation on Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board. See also the mapping between Arduino pins and ATmega168 ports.

4.1.4 Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Nano's digital pins. The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation communication, please see the ATmega168 or ATmega328 datasheet.

4.1.5 Programming

To use the SPI The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Diecimila, Duemilanove, or Nano w/ ATmega168" or "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials. The ATmega168 or ATmega328 on the Arduino Nano comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

4.1.6 Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 or ATmega328 via a 100 Nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the boot loader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

4.2 Servo MG 995 motor

Figure 4.1 Servo Motor

MG995 Metal Gear Servo Motor is a high-speed standard servo can rotate approximately 180 degrees (60 in each direction) used for airplane, helicopter, RC-cars and many RC model. Provides 10kg/cm at 4.8V, and 12kgcm at 6V.

It is a Digital Servo Motor which receives and processes PWM signal faster and better. It equips sophisticated internal circuitry that provides good torque, holding power, and faster updates in response to external forces.

They are packed within a tight sturdy plastic case which makes them water and dust resistant which is a very useful feature in RC planes, Boats, and RC Monster Trucks etc. It equips 3-wire JR servo plug which is compatible with Futaba connectors too.

4.3 HC-05

HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration. HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds. This module works on 3.3 V. We can connect 5V supply voltage as well since the module has on board 5 to 3.3 V regulator. As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, no need to shift transmit level of HC-05 module. But we need to shift the transmit voltage level from microcontroller to RX of HC-05 module.

4.4 PCB

A printed circuit board (PCB) or printed wiring board (PWB) is a laminated sandwich structure of conductive and insulating layers. PCBs have two complementary functions. The first is to affix electronic components in designated locations on the outer layers by means of soldering. The second is to provide reliable electrical connections (and also reliable open circuits) between the component's terminals in a controlled manner often referred to as PCB design. Each of the conductive layers is designed with an artwork pattern of conductors (similar to wires on a flat surface) that provides electrical connections on that conductive layer. Another manufacturing process adds via, plated-through holes that allow interconnections between layers.

PCB Front side

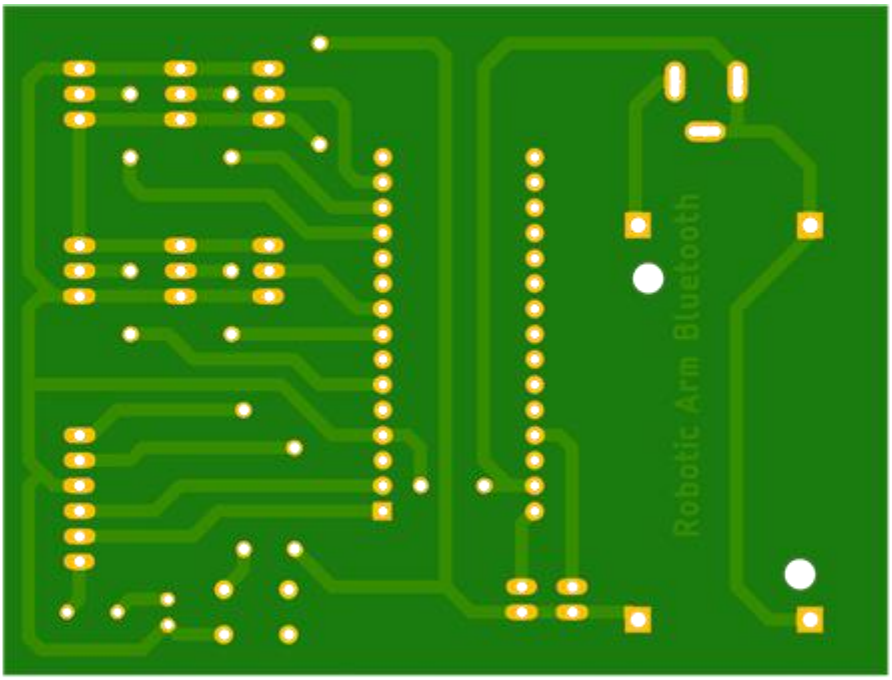


Figure 4.2 PCB front side view without assembly

PCB back side

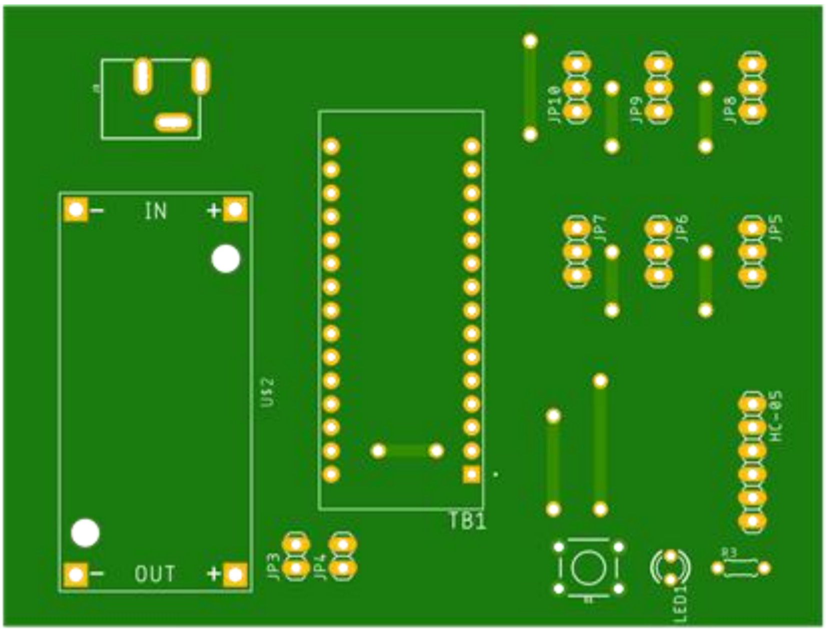


Figure 4.3 PCB Back side view without assembly

4.5 Battery

12V 7Ah batteries are very popular deep cycle and general-purpose batteries, commonly used for powering medical equipment, security systems, UPS and other emergency systems, toys, scooters, fish finders, etc. For a long time, 12V 7Ah batteries were built only as Sealed Lead Acid (SLA) batteries. In recent times, deep cycle Lithium Iron Phosphate ($LiFePO_4$) batteries are offering great weight savings, a larger number of supported charging/discharging cycles, etc. But, SLA AGM (Absorbent Glass Mat) batteries are still going on strong.

Figure 4.4 Battery

5 CHAPTER-5
CLASSIFICATION OF ROBOTS

Industrial robots are found in a variety of locations including the automobile and manufacturing industries. Robots cut and shape fabricated parts, assemble machinery and inspect manufactured parts. Some types of jobs robots do: load bricks, die cast, drill, fasten, forge, make glass, grind, heat treat, load/unload machines, machine parts, handle parts, measure, monitor radiation, run nuts, sort parts, clean parts, profile objects, perform quality control, rivet, sand blast, change tools and weld. Outside the manufacturing world robots perform other important jobs. They can be found in hazardous duty service, CAD/CAM design and prototyping, maintenance jobs, fighting fires, medical applications, military warfare and on the farm.

5.1 Types of robots as per Applications

Nowadays, robots do a lot of different tasks in many fields. And this number of jobs entrusted to robots is growing steadily. That's why one of the best ways how to divide robots into types is a division by their application.

5.1.1 Industrial robots

Robots today are being utilized in a wide variety of industrial applications.

Figure 5.1 Industrial robot model

Any job that involves repetitiveness, accuracy, endurance, speed, and reliability can be done much better by robots, which is why many industrial jobs that used to be done by humans are increasingly being done by robots.

5.1.2 Mobile robots

Also known as Automated Guided Vehicles, or AGVs, these are used for transporting material over large sized places like hospitals, container ports, and warehouses, using wires or markers placed in the floor, or lasers, or vision, to sense the environment they operate in. An advanced form of the AGV is the SGV, or the Self Guided Vehicle, like PatrolBot Gofer, Tug, and Speci-Minder, which can be taught to autonomously navigate within a space.

5.1.3 Agriculture robots

Although the idea of robots planting seeds, ploughing fields, and gathering the harvest may seem straight out of a futuristic science fiction book, nevertheless there are several robots in the experimental stages of being used for agricultural purposes, such as robots that can pick apples.

5.1.4 Telerobots

These robots are used in places that are hazardous to humans, or are inaccessible or far away. A human operator located at a distance from a telerobot controls its action, which was accomplished with the arm of the space shuttle.

Figure 5.2 Telerobots model

Telerobots are also useful in nuclear power plants where they, instead of humans, can handle hazardous material or undertake operations potentially harmful for humans.

5.1.5 Service robots

The Japanese are in the forefront in these types of robots. Essentially, this category comprises of any robot that is used outside an industrial facility, although they can be sub-divided into two main types of robots: one, robots used for professional jobs, and the second, robots used for personal use. Amongst the former type are the above-mentioned robots used for military use, and then there are robots that are used for underwater jobs, or robots used for cleaning hazardous waste.

5.2 Types of Robots by Locomotion & Kinematics

As you can understand, robot's application alone does not provide enough information when talking about a specific robot. For example, an industrial robot - usually, when talking about industrial robots we think of stationary robots in a work cell that do a specific task.

5.2.1 Cartesian robot /Gantry robot

Used for pick and place work, application of sealant, assembly operations, handling machine tools and arc welding. It's a robot whose arm has three prismatic joints, whose axes are coincident with a Cartesian coordinator.

5.2.2 Cylindrical robot

Used for assembly operations, handling at machine tools, spot welding, and handling at die-casting machines. It's a robot whose axes form a cylindrical coordinate system.

5.2.3 Spherical/Polar robot

Used for handling at machine tools, spot welding, die-casting, fettling machines, gas welding and arc welding. It's a robot whose axes form a polar coordinate system.

5.2.4 SCARA robot

Used for pick and place work, application of sealant, assembly operations and handling machine tools. It's a robot which has two parallel rotary joints to provide compliance in a plane.

5.2.5 Articulated robot

Used for assembly operations, die-casting, fettling machines, gas welding, arc welding and spray painting. It's a robot whose arm has at least three rotary joints.

5.2.6 Parallel robot

One use is a mobile platform handling cockpit flight simulator. It's a robot whose arms have concurrent prismatic or rotary joints.

5.3 Tasks

The various tasks which a pick and place robot can perform are as follows: -

5.3.1 Robot pick-and-place

The use of robots for placing products in cartons and transfer of cartons and products between different stations in the packaging lines is very common in all industries. High speed pick-and-place robots for placing small items like candy and cookies in packages are often combined with a visual observation system for identifying products.

5.3.2 Handling of flexible packages

Flexible packaging material is the generic term for soft packages made of film, foil or paper sheeting. Popular forms are stand-up pouches, bags, sachets and envelopes. These packages are often formed, filled and sealed in a vertical or horizontal form-fill-seal machine. The package is then finally put into a case by top loading.

5.3.3 Cartooning machines

Cartooning machines erect boxes from flat sheets of corrugated material. The erected boxes are then filled with products or individual cartons and are then prepared for the palletizing process. As with most packaging machines, vacuum cups, vacuum pumps and other pneumatic components are an essential part of the cartooning.

5.3.4 Rotary cartoners

Rotary cartoner is one of the most popular types of cartooning machines. These machines use a series of vacuum bars equipped with suction cups that move in a continuous rotary motion. Rotary cartoners utilize a "pick-and-carry" motion to move cartons.

5.3.5 Palletizing and depalletizing

Palletizing is the process of placing packages on a pallet alternatively removing them from a pallet (depalletizing). Palletizing machines use vacuum pumps, suction cups and other pneumatic components. These machines typically pick up multiple boxes at a time and place them on a stack (or remove them from a stack).

5.3.6 Automated pick & place robots

The use of specialized machines for high speed pick-and-place of small items with suction cups is very common in the electronics and consumer industries. This application is typically characterized by short cycle times, high acceleration forces and large variations on the parts to be handled.

5.3.7 Seal machines

During the pouch/bag forming phase vacuum is often applied to transport belts that help provide a grip on both sides of the pouch/bag material. The vacuum belt moves the pouch material from a web roll into position to receive the product from the filler. Holes in the belt allow vacuum to hold the pouch while the belt is rotating and the pouch is been removed

5.3.8 Bag opening

Vacuum and suction cups are used to pick and open paper and plastic bags. Suction cups with stiffer bellows and a soft sealing lip are preferred in these quite often high-speed applications.

6 CHAPTER-6
SELECTION OF TASK

From the various tasks which can be done using the pick and place robots we have particularly meshed the two processes of picking & placing along with pallezting process.

We have decided to pick an Automobile Battery (Dimensions 45x45x65mm. Weight 250 grams) from the conveyor.

Then placing it at the packing center, also picking a packed battery from the packing station and moving towards the Box-packing center.

Placing of Battery at Box-packing center and again movement to the conveyor to pick an unpacked battery.

So both the picking & placing along with the packing procedure can be accompanied using this pick and place robot.

- Why pick & place robots
- We have selected the pick and place robots for this particular process due to the following reasons:-
- Using of Human labour for the loading and unloading of the Batteries and also for packing purpose will consume more time.
- Even though Number of laborers is required more, the loading and unloading time should include allowances if laborers are considered.
- Moreover the work can be done easily using a single pick and place robot, which is used for both loading and unloading and pallezting purpose.

6.1 Defining work station

The work station for this operation of pick & place and pallezting is been designed in such a way that:-

- The unpacked battery coming from the belt conveyor is been sensed by a sensor and the moment of the conveyor is been controlled by the sensor.
- As one by one the battery comes, the Robot picks one battery and moves towards the packing station, keeps the battery on the conveyor there.
- Then picks the Packed Battery from there and moves towards the Box-packing center and places the Battery for Box-packaging.
- Further Robot movement continuous towards the return journey takes a Battery from conveyor and again the above procedure is been carried out.

6.2 Working of Robot

This work is able to successfully accomplish the defined functionality. A sample robot which can rotate, magnetize an object, lower and raise its arm, by being controlled by the ESP 32 There are four buttons being controlled by the control unit at the base of the arm:

1) ON/OFF: the ON button puts on the system while the OFF button puts off the system

2) START/STOP: the START button starts the movement of the whole arm from its reset point, while the STOP button takes the arm back to its reset button after completion of its movement.

3) RIGHT-LEFT/LEFT-RIGHT: when this button is switched to the RIGHT-LEFT part it causes movement from right to left, while the LEFT-RIGHT part causes movement from left to right.

4) 4.180/90: when the button is on 180, it causes a rotation of 180 degree of the base stepper motor, but when put on 90 degrees, it causes rotation of 90 degrees.

DESIGN PROCEDURE FACTORS TO BE CONSIDERED

The various factors to be considered while designing of pick and place robots are been discussed as follows. The factors are all important while designing procedure of the robot.

6.2.1 Controls

The mechanical structure of a robot must be controlled to perform tasks. The control of a robot involves three distinct phases - perception, processing, and action. Sensors give information about the environment or the robot itself (e.g. the position of its joints or its end effector). This information is then processed to calculate the appropriate signals to the actuators (motors) which move the mechanical.

The processing phase can range in complexity. At a reactive level, it may translate raw sensor information directly into actuator commands. Sensor fusion may first be used to estimate parameters of interest (e.g. the position of the robot's gripper) from noisy sensor data. An immediate task (such as moving the gripper in a certain direction) is inferred from these estimates. Techniques from control theory convert the task into commands that drive the actuators.

At longer time scales or with more sophisticated tasks, the robot may need to build and reason with a "cognitive" model. Cognitive models try to represent the robot, the world, and how they interact. Pattern recognition and computer vision can be used to track objects. Mapping techniques can be used to build maps of the world. Finally, motion planning and other artificial intelligence techniques may be used to figure out how to act. For example, a planner may figure out how to achieve a task without hitting obstacles, falling over, etc.

6.2.2 Autonomy levels

Control systems may also have varying levels of autonomy. Direct interaction is used for hap tic or teleoperated devices, and the human has nearly complete control over the robot's motion. Operator-assist modes have the operator commanding medium-to-high-level tasks, with the robot automatically figuring out how to achieve them. An autonomous robot may go for extended periods of time without human interaction. Higher levels of autonomy do not necessarily require more complex cognitive capabilities. For example, robots in assembly plants are completely autonomous, but operate in a fixed pattern. Another classification takes into account the interaction between human control and the machine motions.

1. Teleportation: -

A human control each movement; each machine actuator change is specified by the operator.

2. Supervisory: -

A human specifies general moves or position changes and the machine decides specific movements of its actuators.

3. Task-level autonomy: -

The operator specifies only the task and the robot manages itself to complete it.

4. Full autonomy: -

The machine will create and complete all its tasks without human interaction

6.3 Basic methods of Programming Robots

There are three basic methods for programming Industrial robots but currently over 90% are programmed using the teach method.

6.3.1 A Teach Method

The logic for the program can be generated either using a menu-based system or simply using a text editor but the main characteristic of this method is the means by which the Robot is taught the positional data. A teach pendant with Controls to drive the robot in a number of different co-ordinate systems is used to manually drive the robot to the desired locations. These locations are then stored with names that can be used within the robot program. The coordinate systems available on a standard jointed arm robot are:

6.3.2 Joint Co-ordinates

The robot joints are driven independently in either direction.

6.3.3 Global Co-ordinates

The tool centre point of the robot can be driven along the X, Y or Z axes of the Robots global axis system. Rotations of the tool around these axes can also be performed.

6.3.4 Tool Co-ordinates

Similar to the global co-ordinate system but the axes of this one are attached to the tool centre point of the robot and therefore move with it. This system is especially useful when the tool is near to the work piece.

6.3.5 Work piece Co-ordinates

With many robots it is possible to set up a co-ordinate system at any point within the working area. These can be especially useful where small adjustments to the program are required as it is easier to make them along a major axis of the co-ordinate system than along a general line. The effect of this is similar to moving the position and orientation of the global co-ordinate system.

6.4 Principles

A program in Karel is used to control a simple robot that lives in an environment consisting of a grid of streets and avenues. Karel understands five basic instructions:

1. move (Karel moves by one square in the direction he is facing),

2. turn left (Karel turns 90 ° left),

3. put beeper (Karel puts a beeper on the square he is standing at),

4. pick beeper (Karel lifts a beeper off the square he is standing at),

5. Turnoff (Karel switches himself off, the program ends).

Safety requirements

The various safety requirements which were considered while designing the robot are decided as follows:

1. The Robot should not be programmed such that it should damage the Battery while holding it in its gripper.

2. Correct holding position should be set as if it not set then while movement of the Robot it may drop the Lead Batteries which can arise a Hazardous situation in the industry.

3. The Robot should be interfaced properly with the sensors been placed near the Belt conveyor so as to know when the belt conveyor is to be stopped or to be started to move the batteries ahead.

4. Load carrying capacity should be maintained as it should be always more than the default load which is to be shifted.

6.5 Product Design

6.5.1 Selection of product

From the number of products available we selected the Battery of automobiles for been used in our project. We had number of options for the selection of product, as per our requirement the Battery was matching the conditions. The other products which we considered were as follows: -

6.5.1.1 Bearing

Due to radial cross section of the bearing, it would be little bit difficult for the Robot Gripper to hold the bearing in it and transport from one place to another holding it. So we rejected this product.

6.5.1.2 Bags of iron ore

The fines bagging system was pre-decided but due to the weight limit we switched over the other products.

6.5.1.3 Cell phone packing:

As due to the light and sensitive parts of the Cell phones we also skipped it as there are chances of causing damage to the Cell phones while holding in the grippers of the Robots.

6.5.1.4 Bottle packing: -

The radial shape of the bottles was not able to grip inside the grippers of the robots. Though pick and place robots are used in bottle packing industries but they are been designed very precisely and are costly so as the grippers are to be such that it can hold the bottles and move towards the decided target.

6.6 Designing of Workspace

The designing of work space have been done by keeping following points in mind:-

1. It should utilize Minimum time for doing the job.

2. No obstructions should be there in between the workspace envelope.

3. Idle time should be reduced as much as possible.

4. Efficient and safe transportation of the Batteries should be under gone.

The design of work space includes a Belt conveyor which brings the charged batteries from the plant and it is been transferred to the Packing centre Using the Robotic arm. There is moment of 90 degrees; the robot picks a packed Battery from the packed centre after placing the unpacked Battery. Then the robots proceed towards the Box packing centre where it unloads the Battery and further moves towards the Belt conveyor to repeat the same procedure.

6.6.1 Works to be done

6.6.1.1 Selection of parts

Various components of appropriate specifications should be selected so as to complete the fabrication and assembly of the Robot. If the selection is not done properly then the proper working of the robot cannot be obtained. It includes the parts like selection of actuators, motors, sensors etc. Thus, the selection procedure of various components is also an important issue for the project work.

6.6.1.2 Completion of model

Future work is to fabricate and manufacture the complete body structure of the robot, then the assembly of all the manufactured parts are to be done so that the required load is lifted and been transported to the targeted place.

6.7 Programming

Programming of the Pick and place Robot is to be done using a suitable Programming Language. The Robot is to been interfaced with the computer by the programmed software, which will guide the robot to do its job for which it is been programmed. There are numbers of various programming languages available now a days in the market, so the appropriate programming language is to be selected for the programming purpose and the programming is to be done.

6.7.1 6.7.1 Program used

```
#include <Servo.h>


#define SERVO_BASE          2

#define SERVO_SHOULDER      3

#define SERVO_ELBOW         4

#define SERVO_GRIPPER       5


Servo myservo_1;  // create servo object to control a servo

Servo myservo_2;

Servo myservo_3;

Servo myservo_4;


unsigned char Data_String[25],Data_Index = 0,New_Data_Rec_Flag = 0;

unsigned int Received_Servo_Value[4],Final_Servo_Val[4];

unsigned char Index_i = 0,Index_j = 0,Counter_to_Refresh = 0;


void setup() {

  Serial.begin(9600);

  myservo_1.attach(SERVO_BASE);       // attaches the servo on pin 2 to the servo object

  myservo_2.attach(SERVO_SHOULDER);   // attaches the servo on pin 3 to the servo object

  myservo_3.attach(SERVO_ELBOW);      // attaches the servo on pin 4 to the servo object

  myservo_4.attach(SERVO_GRIPPER);    // attaches the servo on pin 5 to the servo object
```

```
  myservo_1.write(90);
  delay(200);
  myservo_2.write(90);
  delay(200);
  myservo_3.write(90);
  delay(200);
  myservo_4.write(90);
  delay(200);

  Received_Servo_Value[0] = 90;     // Default values
  Received_Servo_Value[1] = 90;
  Received_Servo_Value[2] = 90;
  Received_Servo_Value[3] = 45;

  Final_Servo_Val[0] = 90;         // Default values
  Final_Servo_Val[1] = 90;
  Final_Servo_Val[2] = 90;
  Final_Servo_Val[3] = 45;
}

void loop()
{
  if(New_Data_Rec_Flag==0)
  {
    if (Serial.available())     // check whether bluetooth data is available
    {
      // read incoming serial data:
      char inChar = Serial.read();    // read bluetooth data one by one
      //Serial.print(inChar);
```

```
    if(inChar==0x0A)              // End character of bluetooth data
    {
      Data_String[Data_Index] = inChar;
      Data_Index = 0;
      New_Data_Rec_Flag = 1;
    }
    else
    {
     if(inChar!=0x2C)             // removing ascii value, except for 0x2C which is ascii for ,
     {
       inChar = inChar - 0x30;   // removing Ascii value of 0, so that we can get exat value
     }
     Data_String[Data_Index] = inChar;
     Data_Index++;
    }
  }
}

if(New_Data_Rec_Flag==1)
{
   Received_Servo_Value[0] = 0;
   Received_Servo_Value[1] = 0;
   Received_Servo_Value[2] = 0;
   Received_Servo_Value[3] = 0;

   for(Index_i = 0,Index_j = 0;;)
   {
       if(Data_String[Index_j]==0x2C)
       {
```

```
            Index_j++;
            Index_i++;
            Serial.print("A ");
          }
          else if(Data_String[Index_j]==0x0A)
          {
              New_Data_Rec_Flag = 0;
              Serial.print("B ");
              break;
          }
          else
          {
            Received_Servo_Value[Index_i] = Received_Servo_Value[Index_i] * 10 +
Data_String[Index_j];
            Index_j++;
            Serial.print("C ");
          }
      }
      Serial.print(Received_Servo_Value[0]);
      Serial.print("  ");
      Serial.print(Received_Servo_Value[1]);
      Serial.print("  ");
      Serial.print(Received_Servo_Value[2]);
      Serial.print("  ");
      Serial.println(2*Received_Servo_Value[3]);
  }

  Counter_to_Refresh++;
  delay(1);
```

```
 if(Counter_to_Refresh >= 10)    // delay of 10 msec = 1 msec * 10, this will allow smooth
movement of servos
 {
    Counter_to_Refresh = 0;
    if(Received_Servo_Value[0]!=Final_Servo_Val[0])
    {
       if(Received_Servo_Value[0]>Final_Servo_Val[0])
       {
          Final_Servo_Val[0]++;
       }

       if(Received_Servo_Value[0]<Final_Servo_Val[0])
       {
          Final_Servo_Val[0]--;
       }
       myservo_1.write(180 - Final_Servo_Val[0]);      // adjuted as per app
    }

    if(Received_Servo_Value[1]!=Final_Servo_Val[1])
    {
       if(Received_Servo_Value[1]>Final_Servo_Val[1])
       {
          Final_Servo_Val[1]++;
       }

       if(Received_Servo_Value[1]<Final_Servo_Val[1])
       {
          Final_Servo_Val[1]--;
       }
```

```
  myservo_2.write(Final_Servo_Val[1]);
}

if(Received_Servo_Value[2]!=Final_Servo_Val[2])
{
  if(Received_Servo_Value[2]>Final_Servo_Val[2])
  {
    Final_Servo_Val[2]++;
  }

  if(Received_Servo_Value[2]<Final_Servo_Val[2])
  {
    Final_Servo_Val[2]--;
  }
  myservo_3.write(Final_Servo_Val[2]);
}

if(Received_Servo_Value[3]!=Final_Servo_Val[3])
{
  if(Received_Servo_Value[3]>Final_Servo_Val[3])
  {
    Final_Servo_Val[3]++;
  }

  if(Received_Servo_Value[3]<Final_Servo_Val[3])
  {
    Final_Servo_Val[3]--;
  }
  myservo_4.write(180 - (2*Final_Servo_Val[3]));  // adjuted as per app
```

```
    }
  }
}
```

6.7.2 Interfacing with the computer

In the industrial design field of human-machine interaction, the user interface is where interaction between humans and machines occurs. The goal of interaction between a human and a machine at the user interface is effective operation and control of the machine, and feedback from the machine which aids the operator in making operational decisions. A user interface is the system by which people (users) interact with a machine. The user interface includes hardware (physical) and software (logical) components. User interfaces exist for various systems, and provide a means of: • Input, allowing the users to manipulate a system, Output, allowing the system to indicate the effects of the users' manipulation

After completion of the model of the pick and place robot and selection of programming language both should be interfaced. The interfacing of robot and computer using the software is the most important thing in the project. It should be interfaced using trial and error method, and then final movement should be set using the software's. The movement of robot should be precisely managed causing no harm to the operator, and also the batteries which are to be moved from one station to another.

7 CHAPTER-7
INTRODUCTION TO CAD

Throughout the history of our industrial society, many inventions have been patented and Whole new technologies have evolved. Perhaps the single development that has impacted. Manufacturing more quickly and significantly than any previous technology is the digital computer.

Computers are being used increasingly for both design and detailing of engineering components in the drawing office. Computer-aided design (CAD) is defined as the application of computers and graphics Software to aid or enhance the product design from conceptualization to documentation.

CAD is most commonly associated with the use of an interactive computer graphics system, referred to as a CAD system. Computer-aided design systems are powerful tools and, in the mechanical, design and geometric modeling of products and components.

There are several good reasons for using a CAD system to support the engineering design function:

- To increase the productivity
- To improve the quality of the design
- To uniform design standards
- To create a manufacturing data base
- To eliminate inaccuracies caused by hand-copying of drawings and inconsistency between drawings

7.1 CAD/CAM Software

Software allows the human user to turn a hardware configuration into a powerful design and manufacturing system. CAD/CAM software falls into two broad categories,2-D and 3-D, based on the number of dimensions are called 2-D representations of 3-D objects is inherently confusing. Equally problem has been the inability of manufacturing personnel to properly read and interpret complicated 2-D representations of objects. 3-D software permits the parts to be viewed with the 3-D planes-height, width, and depth-visible. The trend in CAD/CAM is toward 3-D representation of graphic images. Such representation approximates the actual shape and appearance of the object to be produced; therefore, they are easier to read and understand.

7.1.1 Applications of CAD/CAM

The emergence of CAD/CAM has had a major impact on manufacturing, by standardizing product development and by reducing design effort, tryout, and prototype work; it has made possible significantly reduced costs and improved productivity.

AutoCAD is a computer-aided drafting and design system implemenented on a personal computer. It supports a large number of devices. Device drivers come with the system and include most of the digitizers, printer/plotters, video display boards, and plotters available on the market.

AutoCAD supports 2-D drafting and 3-D wire-frame models. The system is designed as a single-user CAD package. The drawing elements are lines, polylines ofany width, arcs, circles, faces, and solids. There are many ways to define a drawing element. For example, a circle canbe defined by center and its radius, three points, and two end points of its diameter. The system alwaysprompts the user for all options.Of course, the prompt can be turned off by advanced users. Annotation and dimensioning are also supported. Text and dimension symbols can be placed on anywhere on the drawing, at any angle, and at any size. A variety of fonts and styles are also availble.

7.1.2 Some typical applications of CAD/CAM are as follows:

Programming for NC, CNC, and industrial robots;

Design of dies and molds for casting, in which, for example, shrinkage allowances are preprogrammed;

Design of tools and fixtures and EDM electrodes;

Quality control and inspection----for instance, coordinate-measuring machines programmed on a CAD/CAM workstation;

Process planning and scheduling.

8 CHAPTER-8
INTRODUCTION TO PRO/ENGINEER

Pro/ENGINEER, PTC's parametric, integrated 3D CAD/CAM/CAE solution, is used by discrete manufacturers for mechanical engineering, design and manufacturing.

Created by Dr. Samuel P. Geisberg in the mid-1980s, Pro/ENGINEER was the industry's first successful parametric, 3D CAD modeling system. The parametric modeling approach uses parameters, dimensions, features, and relationships to capture intended product behavior and create a recipe which enables design automation and the optimization of design and product development processes.

This powerful and rich design approach is used by companies whose product strategy is family-based or platform-driven, where a prescriptive design strategy is critical to the success of the design process by embedding engineering constraints and relationships to quickly optimize the design, or where the resulting geometry may be complex or based upon equations. Pro/ENGINEER provides a complete set of design, analysis and manufacturing capabilities on one, integral, scalable platform. These capabilities, include Solid Modeling, Surfacing, Rendering, Data Interoperability, Routed Systems Design, Simulation, Tolerance Analysis, and NC and Tooling Design.

Companies use Pro/ENGINEER to create a complete 3D digital model of their products. The models consist of 2D and 3D solid model data which can also be used downstream in finite element analysis, rapid prototyping, tooling design, and CNC manufacturing. All data is associative and interchangeable between the CAD, CAE and CAM modules without conversion. A product and its entire bill of materials(BOM) can be modeled accurately with fully associative engineering drawings, and revision control information. The associativity in Pro/ENGINEER enables users to make changes in the design at any time during the product development process and automatically update downstream deliverables. This capability enables concurrent engineering — design, analysis and manufacturing engineers working in parallel — and streamlines product development processes.

Pro/ENGINEER is an integral part of a broader product development system developed by PTC. It seamlessly connects to PTC’s other solutions including Windchill, ProductView, Mathcad and Arbortext.

8.1 Different modules in PRO/ENGINEER:

- PART DESIGN
- ASSEMBLY
- DRAWING
- SHEETMETAL
- MANUFACTURING

8.1.1 Design of present work:

The present work is aimed at designing a Robotic arm for pick and place. the total system can be sub-divided into different parts, such as:

1. Motor
2. Node MCU
3. Gear pick and place
4. Wires

8.2 3D Images in PRO/ENGINEER software

8.2.1 8.2.1 Robot Assembly

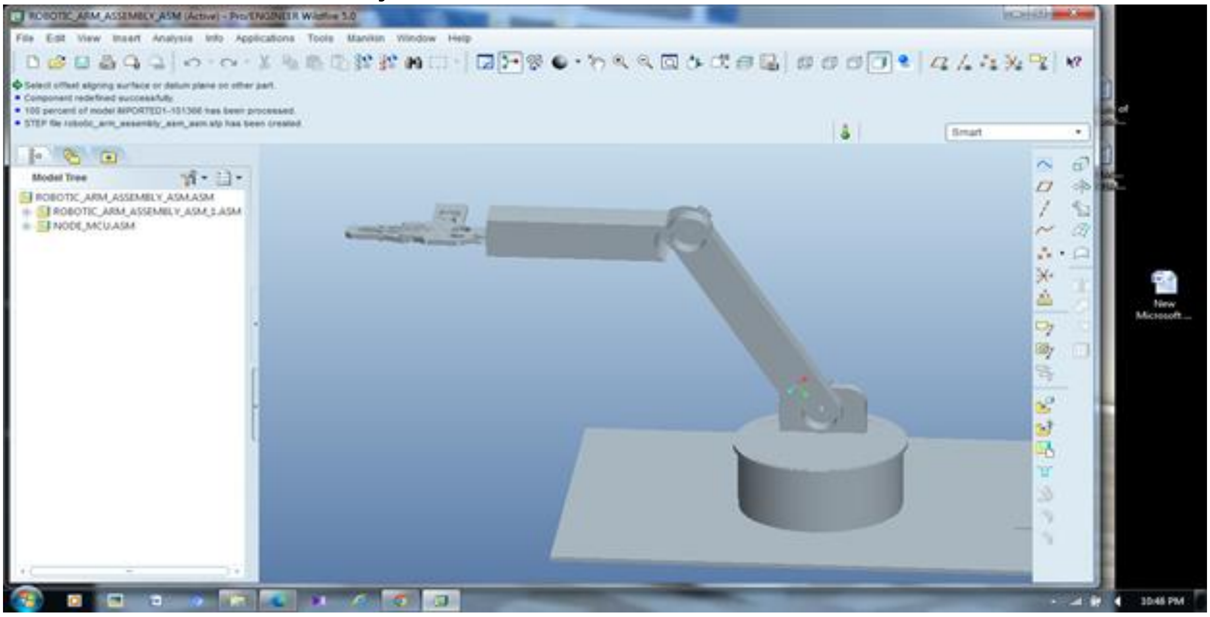


Figure 8.1 robot 3D assembly model.

8.2.2 Node MCU

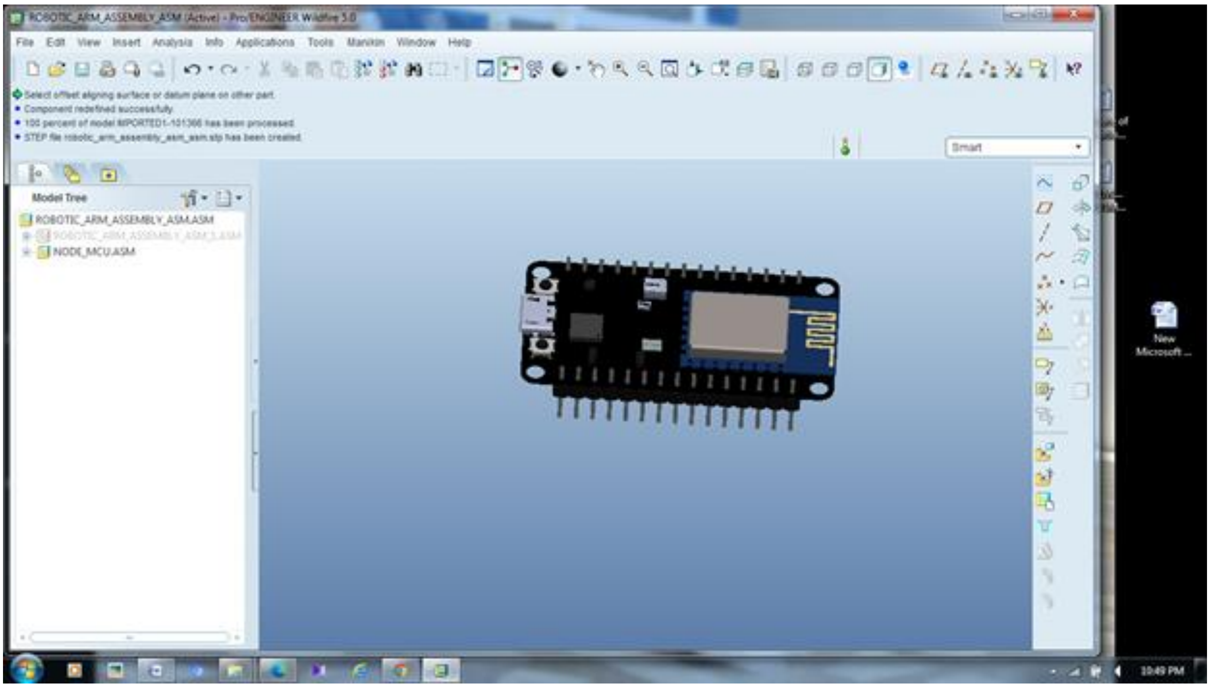


Figure 8.2 Node MCU 3D model.

8.2.3 Holding Jaw

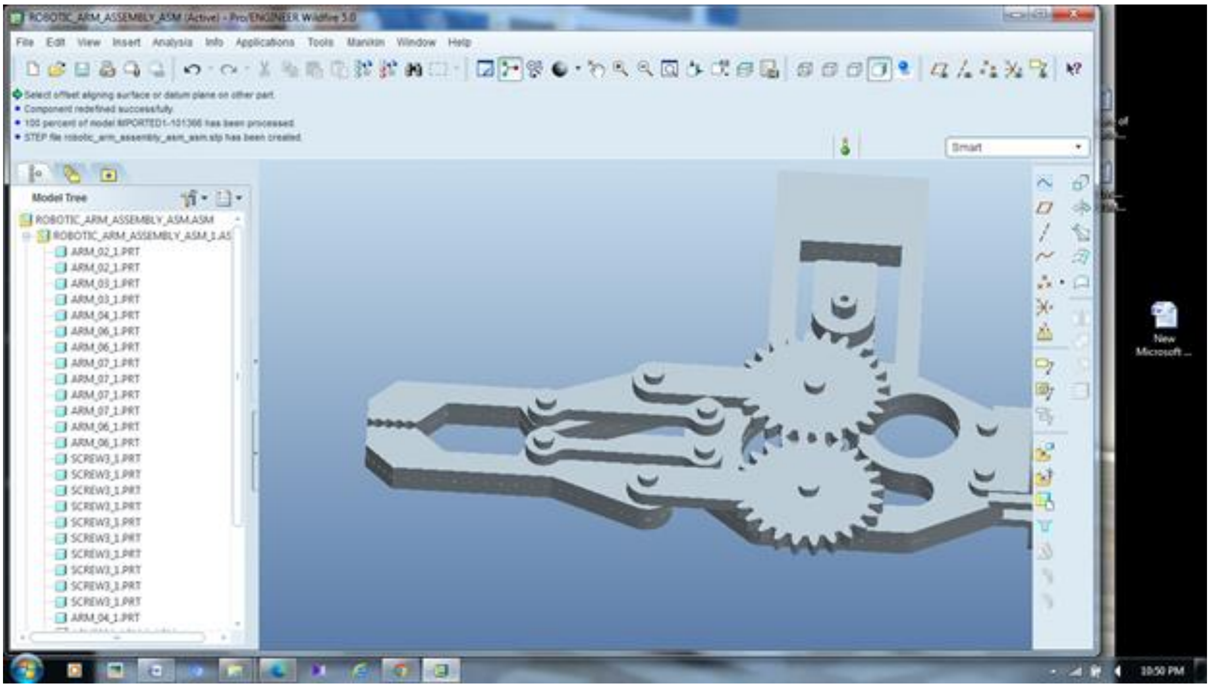


Figure 8.3 Holding Jaw 3D model.

8.2.4 Link 1

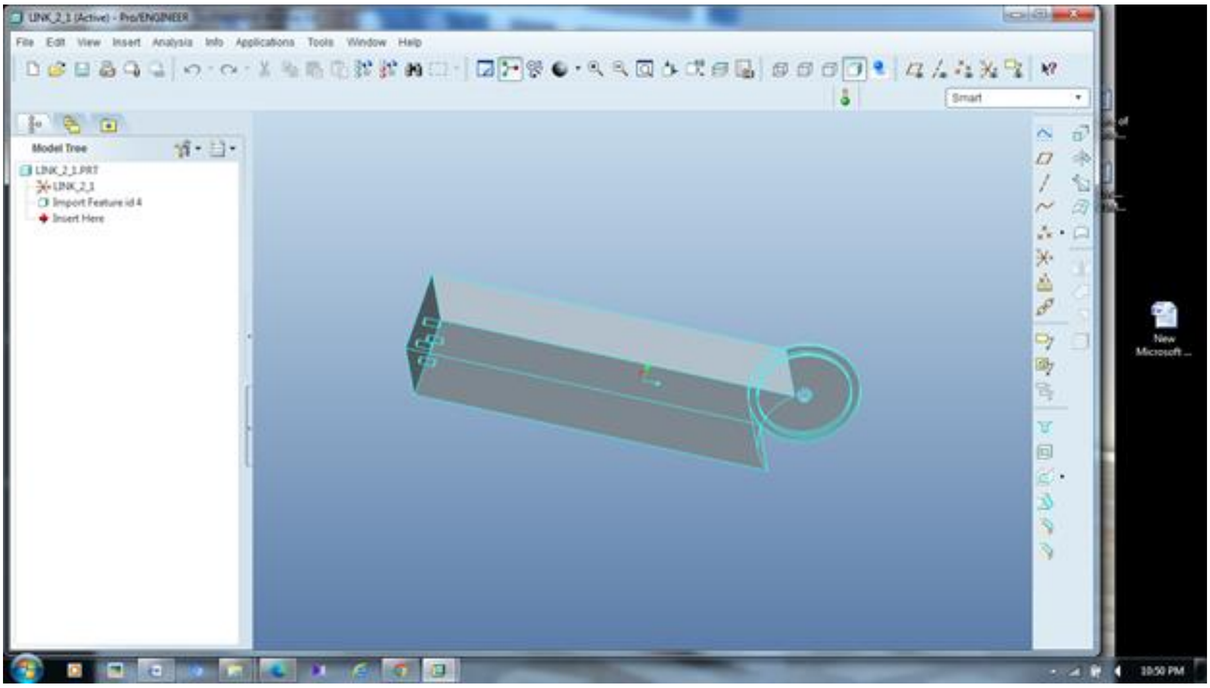


Figure 8.4 Link 1 3D model.

8.2.5 Link 2

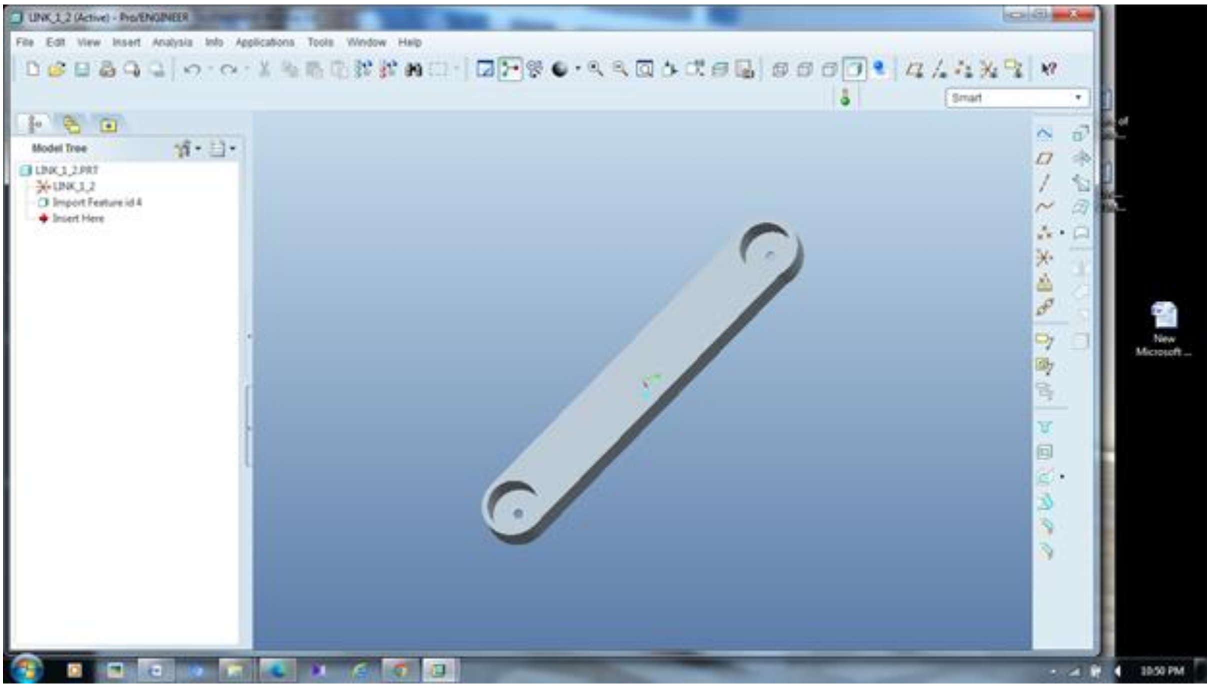


Figure 8.5 Link 2 3D model.

8.2.6 Main base attach

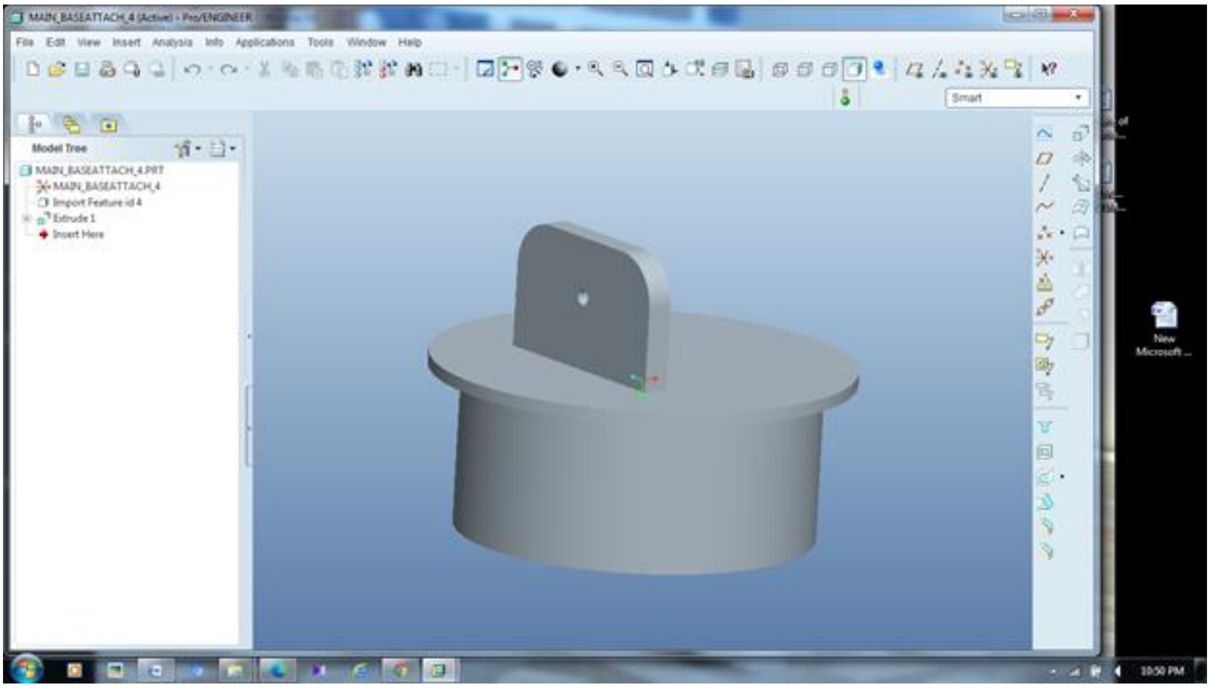


Figure 8.6 Main base attach 3D model.

8.2.7 Rotate Attachment

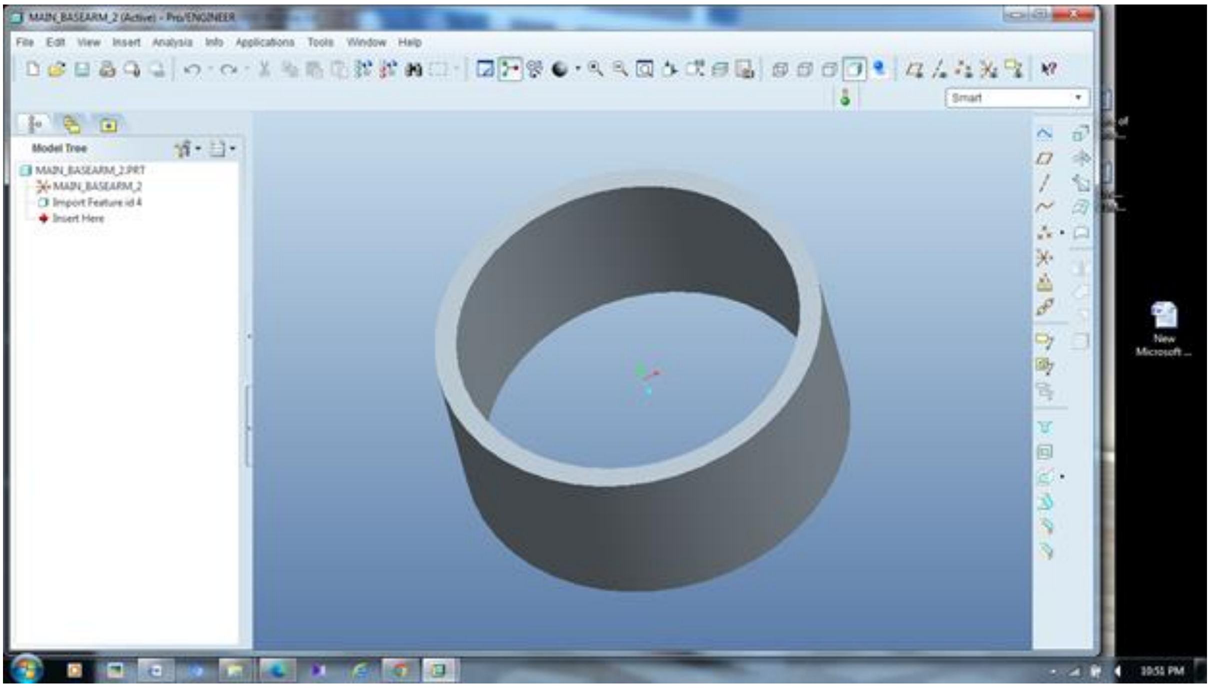


Figure 8.7 rotate attach 3D model.

8.2.8 Base

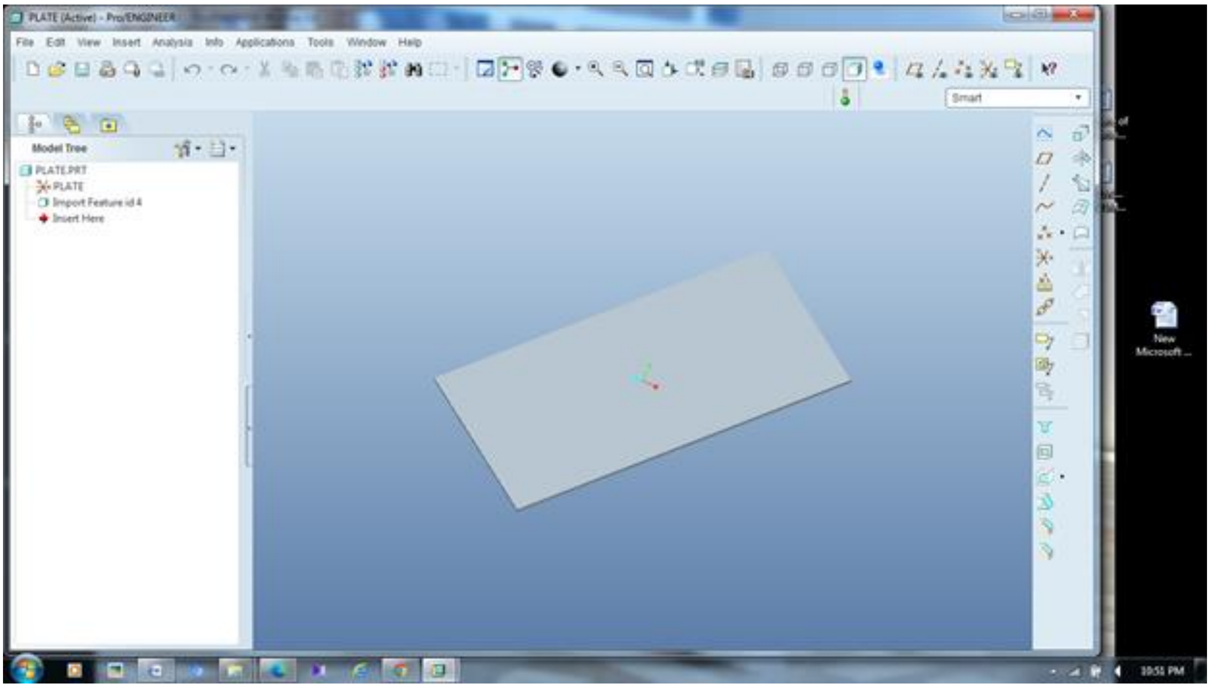


Figure 8.8 base 3D model.

8.2.9 PCB

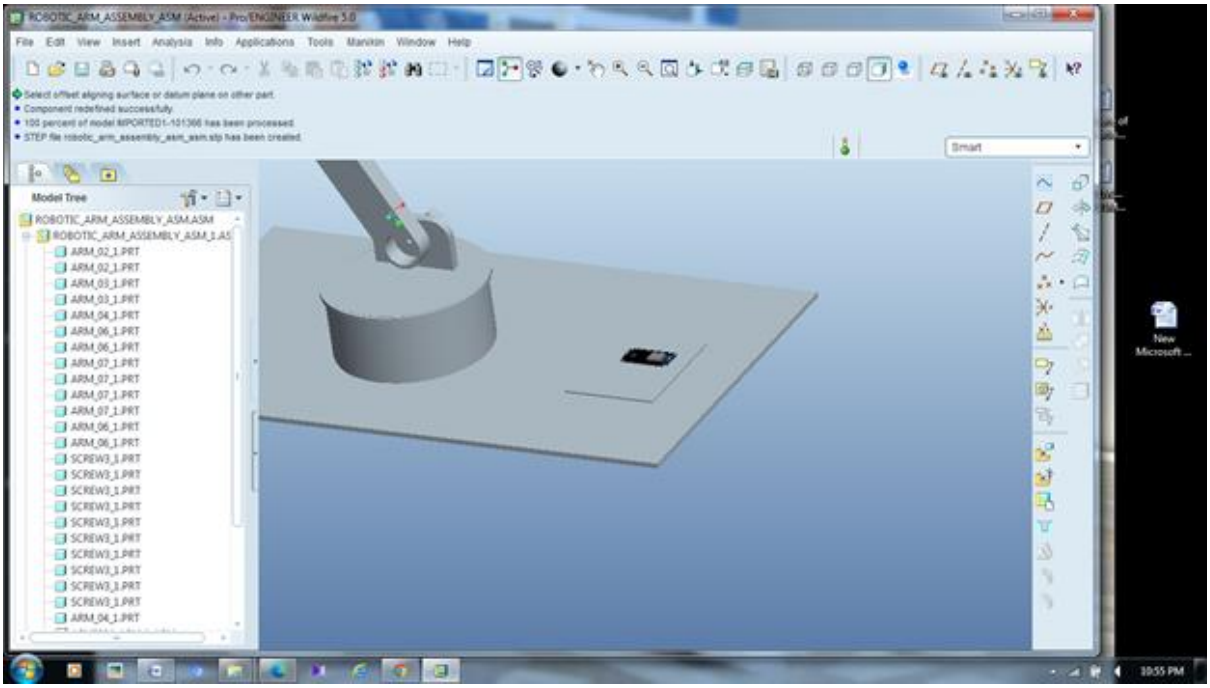


Figure -8.9 PCB 3D model.

8.3 Drawing images

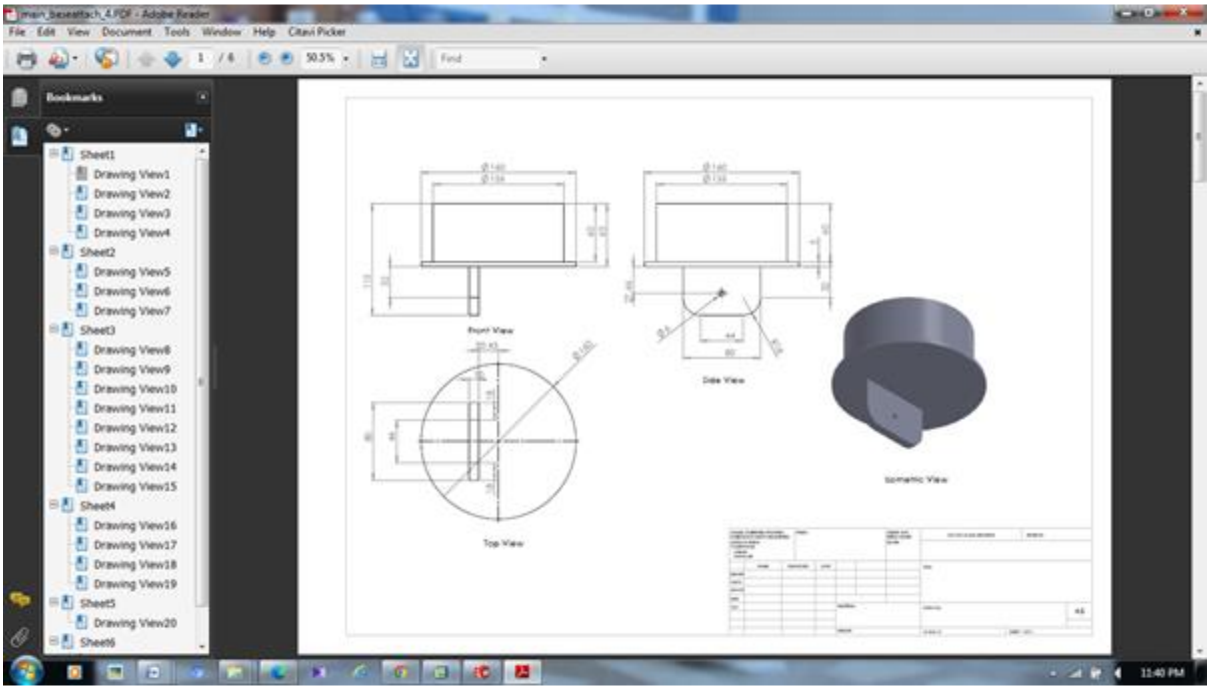


Figure 8.10 Main base attach drawing image

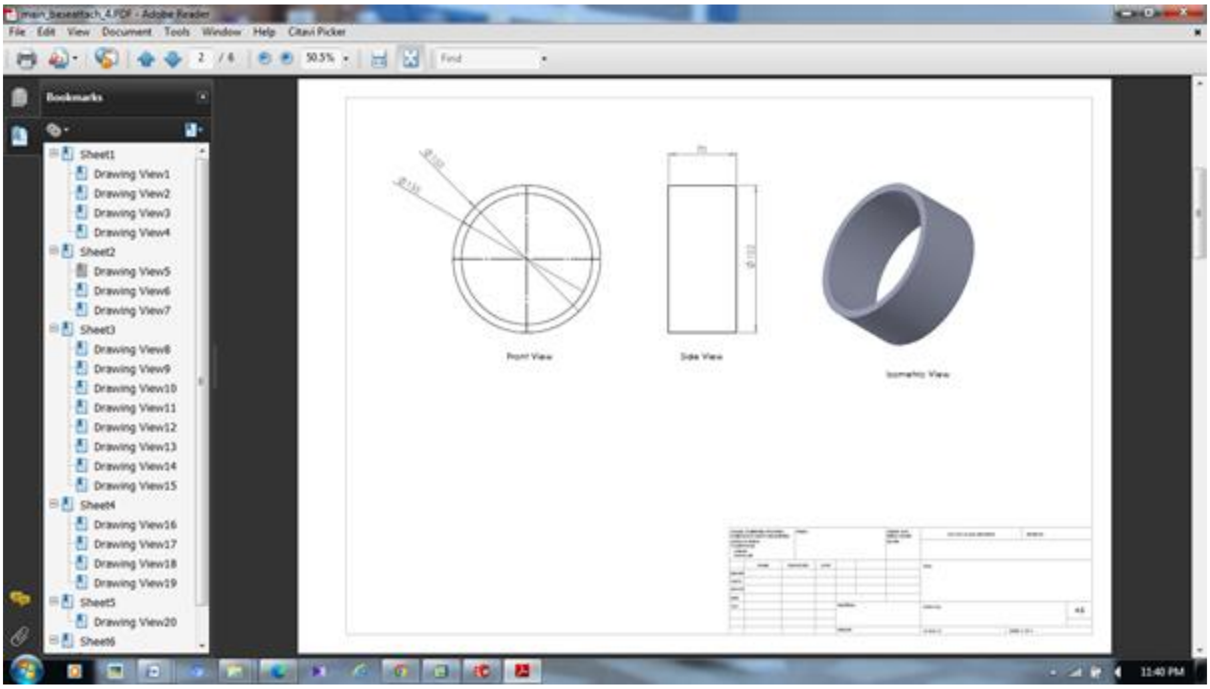


Figure 8.11 Rotate attach drawing image

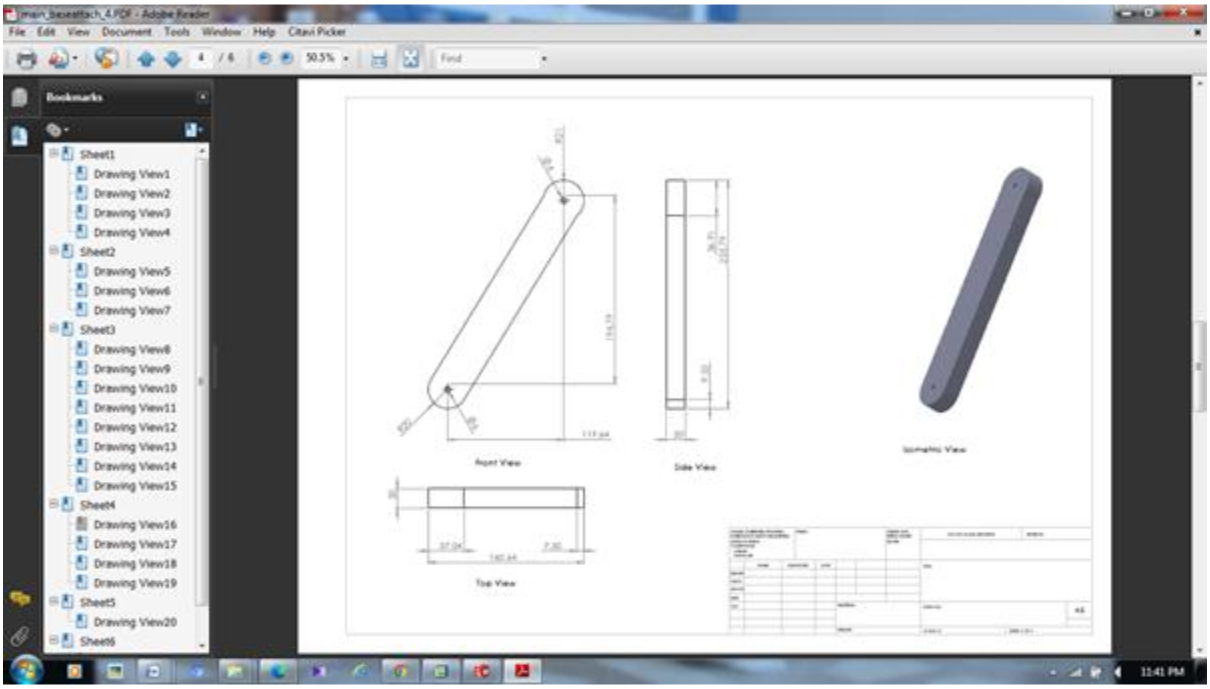


Figure 8.12 Link 2 drawing image

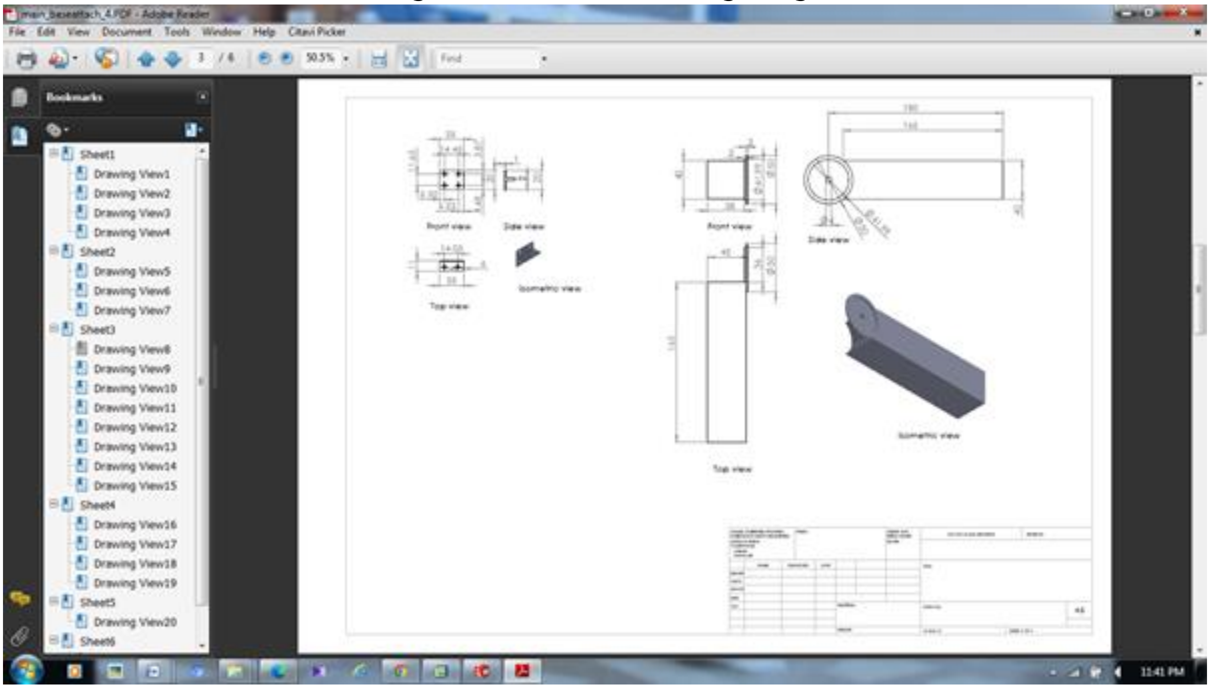


Figure 8.13 Link 1 drawing image

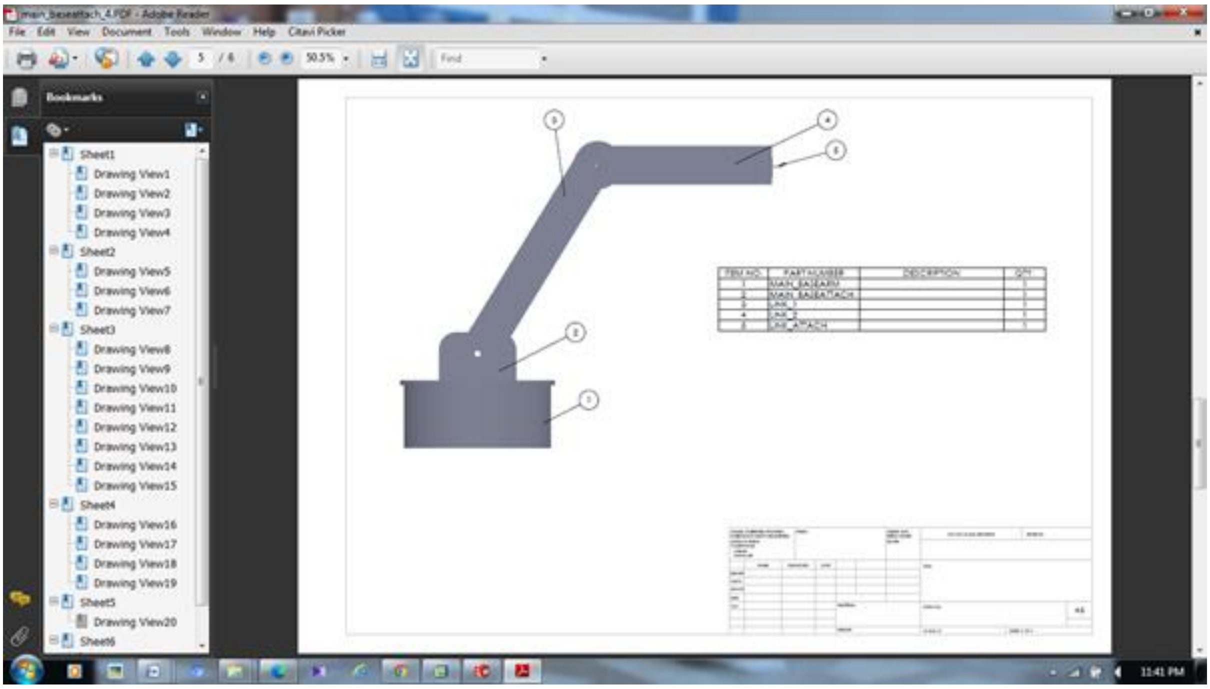


Figure 8.14 Robot 3D assembly drawing image

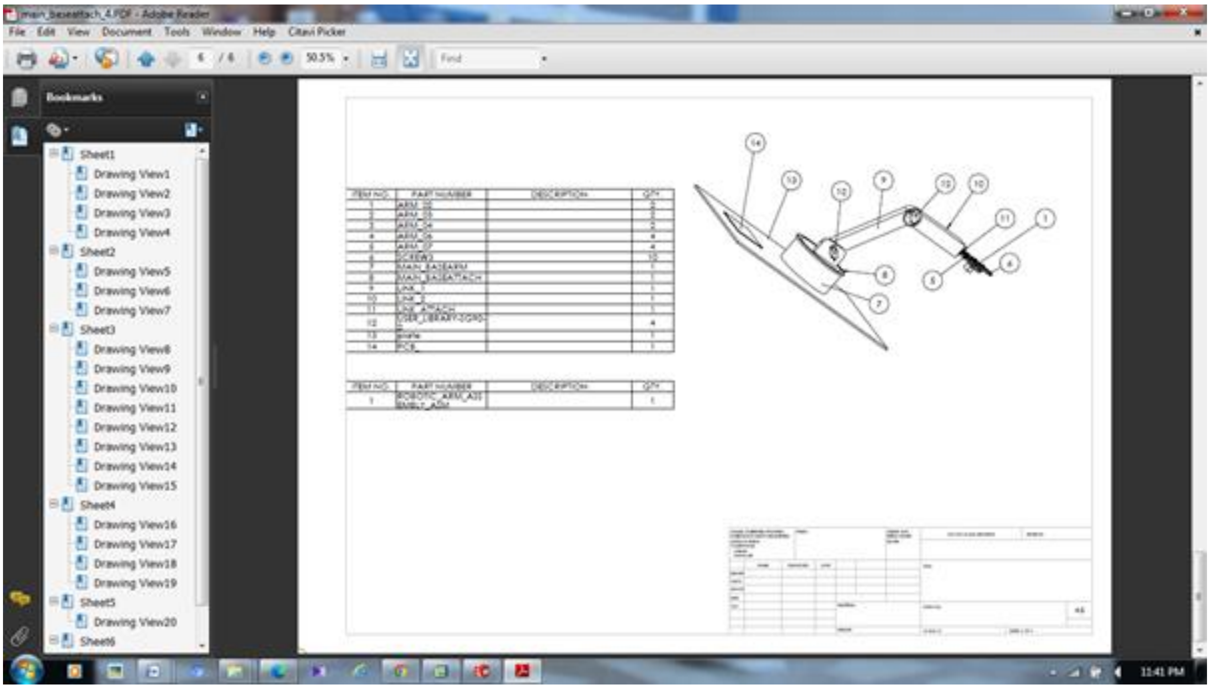


Figure -8.15 Parts specification drawing image

Figure -8.10 to 8.15 drawing images in Pro/Engineer software.

9 CHAPTER-9
MANUFACTURING PROCESS

9.1 Assembling the Base of Robotic Arm

Take the bottom outer bearing disc and slide it over standoffs. Mount bottom inner bearing disc on stand-offs. Align the disc such that smaller holes are in line with the stand-offs. Put the bearing on top of bottom inner bearing disc. Observe how disc supports the bearing from beneath. Next, place the two middle inner bearing discs. These will perfectly fit inside the bearing.

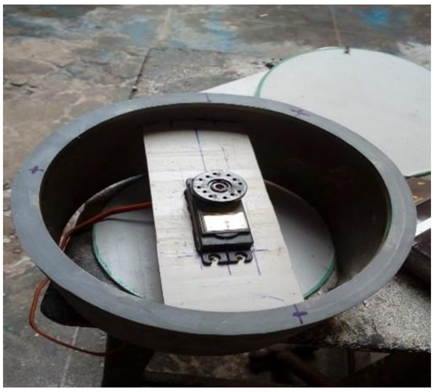


Figure 9.1 Base

9.2 Attaching Base Servo Horn

Mount servo horn on top inner bearing disc using self-threading M2 (“M2” represents a diameter of 2mm) screws of 8mm length. These screws can be found in servo accessories. Be careful about holes you are mounting the horn on (use holes closer to disk centre).Seat the “servo horn & top inner bearing disc” assembly on top of bearing as shown. Make sure all smaller holes in these discs are aligned since we will fasten them all together using 20mm M3 bolts.

9.3 Completing The Base Assembly

Place the middle outer bearing discs first and then top outer bearing disc as shown. The middle discs will have a larger inner diameter compared to the top and bottom ones. The inner diameters of the middle discs will be equal to the bearing’s outermost diameter. Slide up the bottom outer bearing disc, align the holes in the discs and stack them. Thus, sandwiching the bearing as the top and bottom bearing discs have a smaller inner diameter compared to the bearing’s outermost diameter. Place 20 mm stand-offs on the top and fasten the M3 bolts of 20mm length from the bottom. Your base is completed now.

Figure 9.2 Completing Base Assembly

9.4 Attaching Servo To The Link 1

Place servo motor into the slot available on racquet-shaped link 1. Be careful about the positioning and the orientation of the servo. Fasten the servo to the link using M4 bolts of 16mm length and nuts. The head of the servo pointing upwards. Don’t let this confuse you; the head actually goes into the slot in the horn. Attach M3 bolts of 8mm length and 20mm standoffs to the link.

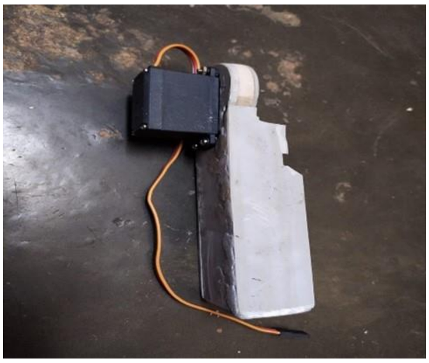


Figure 9.3 Attaching servo to link 1

9.5 Orientation of the Link

This is the orientation in which we will place the completed link upon our base. Note the free holes through which we will drive the screws to fix it to the base.

9.6 Adding Link 1 To Base

Place the assembly of the Link on the base as shown & fasten using M3 bolts of 8mm length. Keep in mind that you’ve set servo at 90 degrees before assembling (connect the servo to evive, go to control > servo and set servo at 90 degrees). Observe how the head of the servo goes into the servo horn. We have to ensure proper locking between the servo head and servo horn. To do this, drive a bolt into the centre of the horn (find this in servo accessories) through the bottom of the base (through the large hole) and fasten it.

9.7 Completing The Assembly of the First Link

Push the servo in the slot provided on the racquet shaped link 2 as shown and fasten it using M4 bolts of 16mm length and M4 nuts. Notice the difference between this step and what we did for the sketching robot. Place this on the base and fasten using M3 bolts of 8mm length.

9.8 Getting Started With The Second Link

Take the supporting link and attach a servo horn to it as shown. Fasten the horn with M2 bolts of 8mm length. Fit the servo holder into the rectangular slots provided on the supporting link. Insert M3 nuts into the slots provided in the servo holder and fasten with M3 bolts of 8mm length. Slide the micro servo into the servo holder as shown and then fix it in place using M2 bolts of 12mm length.

9.9 Assembling The Gripper

Insert another micro servo into the slot on the gripper plate as shown, then affix it there using M2 bolts of 8mm length and M2 nuts. Fasten 15mm standoffs to the plate using M3 bolts of 8mm length. Insert gripper part 1 into the thin slot provided on the gripper plate. Note that this part is not symmetric and the small square slots are nearer to one end than to the other. Slide gripper part 2 (with a thin slit) onto the protrusion on gripper part 1. Notice that gripper part 2 has holes 3mm in diameter so that threaded part of standoffs may pass through it. Fasten these parts using M3 nuts. Insert gripper part 3 into the rectangular holes of gripper part 1 as shown and place M3 nuts and fasten using M3 bolts of 12mm length. Now attach a micro servo horn on gripper part 3 with its holes aligned with the holes on gripper part 3 and fasten with self-threading M2 bolts of 8mm length (from micro servo accessories).

Figure 9.4 Gripper Assembly

9.10 Attaching The Gripper Claw

Bring together gripper claw 1, gripper link and a one-sided micro servo horn in the configuration shown. First, fasten gripper link to the one-sided micro servo horn using the self-threading M2 bolt from micro servo accessories. Keep in mind, protruded end of servo horn should be facing the servo. Next, insert the M3 bolt of 12mm length to gripper claw 1. Hold the “gripper link and servo horn” assembly intact. Then insert gripper link 1 onto this bolt and fasten an M3 lock nut on the same. Ensure that the links are movable about the bolt but they shouldn’t be too loose. Now lock the one-sided micro servo horn and the free servo head using the self-threading bolt from micro servo accessories. Ensure the micro servo is at 90 degrees before assembling. Now, note that there is a free hole on the gripper plate and that gripper claw 1 is in front of the other hole. Intuitively, place gripper claw 2 on the other hole. Now, M4 bolts of 16mm length in both these holes and fasten using M4 lock nuts. Keep in mind that claws aren't too loose but are free to rotate about these bolts.

9.11 Attaching Gripper Assembly to the Link

The micro servo horn on the assembly is still free. We left a micro servo head free too in step 9, where we started making the gripper. Use bolts from micro servo accessories to lock both of these together (ensure micro servo is at 90 degrees before fixing), hence fixing the gripper to the supporting link through the micro servo. Now, attach 20mm standoffs to supporting link and fasten M3 bolts of 8mm length, subsequently bring link 2, align it with the standoffs and fasten M3 bolts of 8mm length.

9.12 Attaching The Second Link To The First Link

Now, set the servo angle to 180 degrees. There is another horn at the bottom of the supporting link which has to be connected to a servo head. Also, recall that there is a free servo head in the first link. Simply lock the two with a bolt from servo accessories and the robot will be complete.

Figure 9.5 Assembling

9.13 Robotic Arm Controller

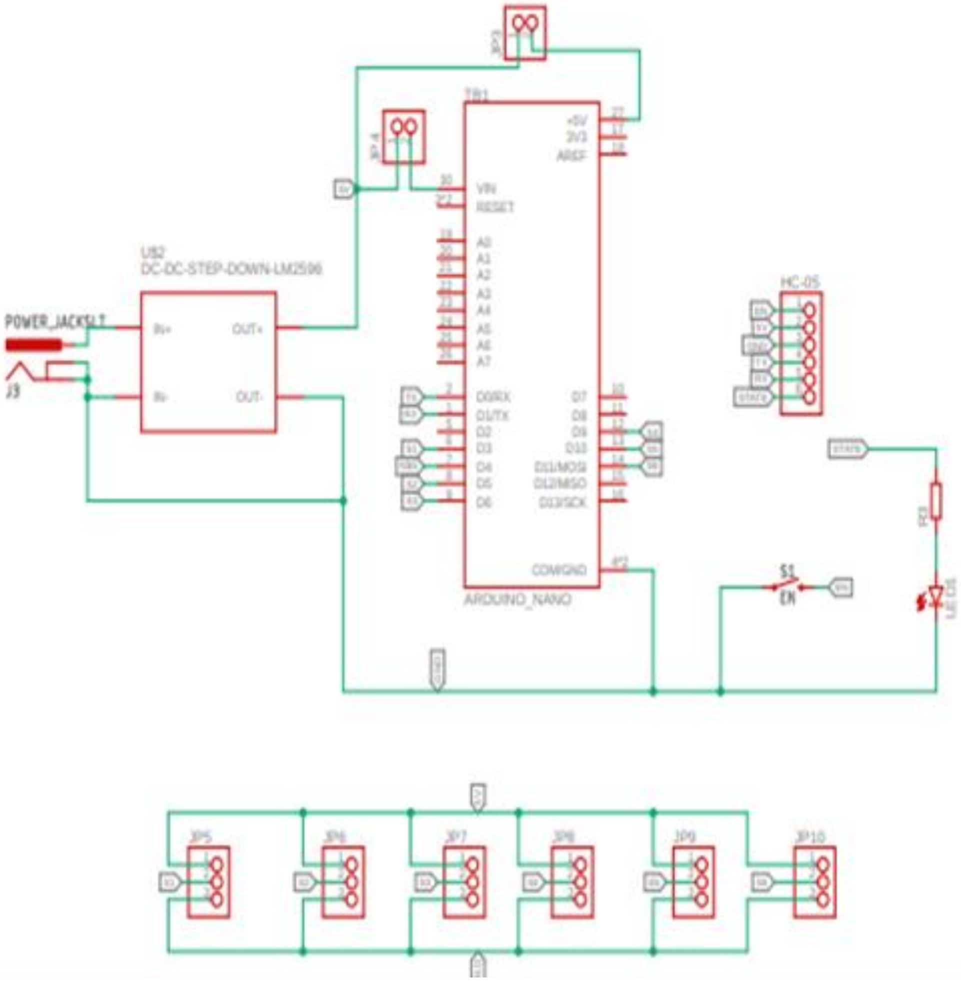


Figure 9.6 Robotic arm Controller layout

10 CHAPTER 10
RESULTS AND DISCUSSION

Result from different working operating condition and effects in return are given below:

Mechanical advantages in movement:

As metallic servos are used in case of functioning the arm, a servo can work with in the range of angle from 0 to 180 degree. But the program coded in the arduino allows the motor to move at some accurate angle.

Angle	**Range**
Elbow	0-120
Wrist	0-40
Arm	0-80

Current consumption:

The current consumption depends on the load and the position of the object.

Load	Current consumption(mA)
15gm	Low (0-200)
35gm	Normal (200-500)
45gm	Normal (500-800)
55gm	High (800-900)
75gm	Overloaded (above 900)
95gm	Overloaded (above 900)

The overall system design is then implemented according to the system block diagram. The system that has been created is then tested to determine its performance. Tests were carried out on servo motors and pick and place missions based on inverse kinematics.

Servo Motor Testing

Servo motor testing is carried out to determine whether the motor is functioning properly or not. The test is carried out on each servo motor at each joint. Servo motor testing is done by measuring the voltage when the servo motor is given a PWM signal and when the servo motor is not given a PWM signal. No-load voltage (VNL) is the voltage when the servo motor is not given a PWM signal. Full load voltage (VFL) is the voltage when the servo motor is given a PWM signal which will make the servo motor have a load.

No Joint VNL (V) VFL (V)

1 Base 5.07 4.99

2 Shoulder 5.06 4.98

3 Elbow 5.04 4.93

4 Gripper 5.00 4.73

Average 5.04 4.90

From the results, it can be seen that the VNL and the VFL have an average difference of 0.08 V. This is because there is power dissipation when the servo motor operates. The existence of power dissipation will affect the position of the end-effector inaccurately.

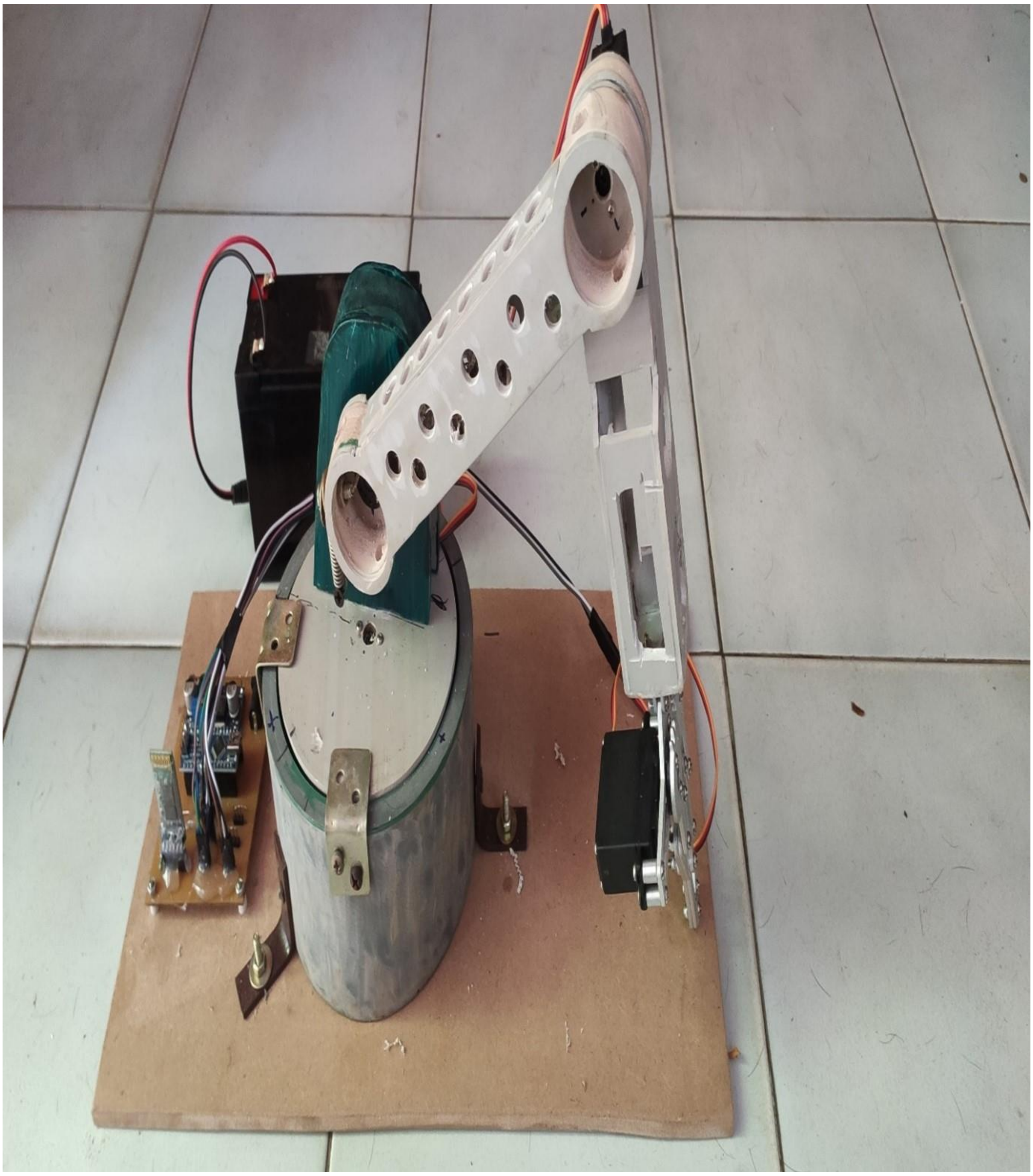
11 CONCLUSION

This Project represents the design, development and implementation of robot arm in terms of industrial automation, which has the talent to accomplish simple tasks, such as it can sense any material and capable of light material handling. The robot arm was designed and built from acrylic material where servo motors and stepper motor were used to perform links between arms and execute arm movements. The servo motors include encoder so that no controller was implemented; how- ever, the rotation range of the motor is less than 180Âº span, which greatly decreases the region reached by the arm and the possible positions. But stepper motors which can rotate 360 degree at 120 steps make the arm easier to solve the problem. The design of the robot arm was limited to four degrees of freedom since this design allows most of the necessary movements and keeps the costs and the complexity of the robot competitively. The end effector is not included in the design because a commercially available gripper is used since it is much easier and economical to use a commercial one than build it. During design, we faced some difficulties due to the way of joining thin acrylic parts strongly. A mechanical junction based on screws and nuts is used and in order to accomplish that, a small feature was designed which allowed fastening the bolts with the nuts without having to screw in the thin acrylic layer. To control the robot an autonomous computer code is used and serial communication method give an access to the autonomous process.

12 REFERENCES

1. Sanjay Lakshmi Narayan, Shweta Patel, “Position Control of Pick and Place Robotic Arm”, Department of Electrical Engineering MS Rakaia Technology, Bangalore, India.
2. Design and fabrication of pick and place robotic arm by Dr Sunil Kumar Talakayala, Research gate 343738510.
3. Nikku, S., “Introduction to Robotics” USA: John Wiley & Sons, 2011.
4. Haidari, A. M. A., Benachiba, C., Zahir, M., “Software Interfacing of Servo Motor with Microcontroller”, Journal of Electrical Systems (JES), pp.84-99, 2013.
5. International federation of robotics (IFR), “World Robotics”, Switzerland: United Nation, 2005.
6. Gieras, F. J., Piech, J. Z., Tomczuk, I. “Linear Synchronous Motors: Transportation and Automation Systems, Second Edition” USA: Taylor and Francis Group, LLC, pp.407-413, 2012.
7. V. Ramya, B. Palaniappan, T. Akilan, Embedded System for Robotic Arm Movement Control Using Web Server and Zigbee Communication, International Conference on Research Trends in Computer Technologies, 2013, pp. 30-34.
8. Pradeep Kumar, et al. design and implementation is based on pneumatic principles. International journal of scientific and technology research volume 2, issue 8, august 2013. 3. Mie Sajjid Hussain Talpur , et al, implementation of micro controller based on reliable and high performance robotic system for food manufacturing line. 978-1-4577-1139-8/12 (c) 2012 IEEE.
9. Mohhammad Hafizuddin.B.A.S et al., use of programmable logic controller to control all the robot movement. April 2008 5. R.V.Sharma , "work piece recognition and location of pick and place robot in SFMS" I.J. Image ,graphics and signal processing , 2014,4,9-17.vol2, issue 3, 2011

13 PHOTO GALLERY



PROTOTYPE OF FABRICATION OF THREE DEGREES OF FREEDOM ROBOTIC ARM


Blow molding
Description :
In 1851, S.T. Armstrong, U.S. patent reference to blow molding a plastic material other than glass. Ferngren and Kopitke produced a blow molding machine and sold it to Hartford Empire Company. This was the beginning of the commercial blow molding process.
S.T. Armstrong
Process of Blow molding
OKAYA