TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta strojní

Design of control system of manipulator for manufacturing system with Industry 4.0 conception

Diplomová práce

Studijní program:	N2301 – Mechanical Engineering
Studijní obor:	2302T010 – Machines and Equipment Design
Autor práce:	**Kasi Viswanathan Puthucode Balakrishnan**
Vedoucí práce:	Ing. Radek Votrubec, Ph.D.

Liberec 2018

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechanical Engineering

Design of control system of manipulator for manufacturing system with Industry 4.0 conception

Master thesis

Study programme:	N2301 – Mechanical Engineering
Study branch:	2302T010 – Machines and Equipment Design
Author:	**Kasi Viswanathan Puthucode Balakrishnan**
Supervisor:	Ing. Radek Votrubec, Ph.D.

Liberec 2018

Technical University of Liberec
Faculty of Mechanical Engineering
Academic year: **2017/2018**

DIPLOMA THESIS ASSIGNMENT

(PROJECT, ART WORK, ART PERFORMANCE)

First name and surname:	**Kasi Viswanathan Puthucode Balakrishnan**
Study program:	**N2301 Mechanical Engineering**
Identification number:	**S16000468**
Specialization:	**Machines and Equipment Design**
Topic name:	**Design of control system of manipulator for manufacturing system with Industry 4.0 conception**
Assigning department:	**Department of Manufacturing Systems and Automation**

Rules for elaboration:

1. Describe basic principles of Industry 4.0.
2. Meet Arduino development board, its accessories and its shields.
3. Design a control system of a manipulator for smart factory model using Arduino board. This manipulator have to communicate with other members of factory via wifi.
4. Write an article with your supervisor about developed theme.

Scope of graphic works:	**as required**
Scope of work report (scope of dissertation):	**50-60**
Form of dissertation elaboration:	**printed/electronical**
Language of dissertation elaboration:	**English**

List of specialized literature:

[1] BEQUETTE, B. Process control: modeling, design, and simulation. Upper Saddle River, N.J.: Prentice Hall PTR, 2003. ISBN 0133536408.
[2] Arduino Learning: Getting Started with Arduino. In: Arduino [online]. 2014 [cit. 2015-01-09]. available from: http://arduino.cc/en/Guide/HomePage

Tutor for dissertation:	**Ing. Radek Votrubec, Ph.D.** Department of Manufacturing Systems and Automation
Date of dissertation assignment:	**15 November 2017**
Date of dissertation submission:	**15 May 2019**

prof. Dr. Ing. Petr Lenfeld
Dean

L.S.

Ing. Petr Zelený, Ph.D.
Head of Department

Liberec, dated: 15 November 2017

Declaration:

I hereby certify that I have been informed the Act 121/2000, the Copyright Act of the Czech Republic, namely § 60 - Schoolwork, applies to my master thesis in full scope.

I acknowledge that the Technical University of Liberec (TUL) does not infringe my copyrights by using my master thesis for TUL's internal purposes.

I am aware of my obligation to inform TUL on having used or Iicensed to use my master thesis; in such a case TUL may require compensation of costs spent on creating the work at up to their actual amount.

I have written my master thesis myself using literature listed therein and consulting it with my thesis supervisor and my tutor.

Concurrently 1 confirm that the printed version of my master thesis is coincident with an electronic version, inserted into the 15 5TAG.

Date: 21-05-2018

Signature:

Acknowledgment

It is with immense pleasure and gratitude that I acknowledge the support of my university for providing me this great opportunity to develop in the deepest manner my engineering skills for accomplishing this diploma thesis and help me to further develop the knowledge on the Industry concept 4.0 with the latest trends being used.

I am indebted firstly to thank **Ing. Radek Votrubec, Ph.D.** for his professional guidance, encouragement and good advice all along. This thesis is a much work better thanks to his supervision. It's always an immense pleasure to thank him for the skilled assistance he provided.

I would like to thank our head of the department **Ing. Petr Zelený Ph.D.** who has been a great support in every way to pursue our academics. Without his help, I might otherwise have never encountered successfully.

I have been greatly benefited from the guidance provided by **Ing. Iaroslav Kovalenko** and **Ing. Maryna Garan** in their seminars about thesis writing.

I would like to express my gratitude to The Department of Manufacturing System and Automation at the Technical university of Liberec for providing me the support I required, which gave me the opportunity to proceed freely with this work.

Finally, I am grateful to thank my parents and friends, who have been a moral support in all the situations and encouraging my works.

ABSTRACT:

This thesis supports the project of the department for teaching Industry 4.0. In this work, the construction of the manipulator - the robotic arm, which is part of the model of the production system with the concept of Industry 4.0, was made. The robot is one of the components that make up Internet things in the factory environment. With other elements it communicates via WIFI communication. It is necessary to choose an appropriate robot control design that is compatible with the operation we need to perform the required task. Part of the thesis is to find kinematic equations for movement of individual arms and their conversion to variables corresponding to positions of servomotors. The goal is to get the endpoint of the arm to the required coordinates. The programming of the robot is done using the control microprocessor and the Arduino software. Functions have been programmed in the control program to move the product in a model factory from one location to another.

ABSTRAKT:

Tato práce podporuje projekt katedry pro výuku koncepce Průmysl 4.0. V této práci byla provedena konstrukce manipulátoru - robotického ramene, které je součástí modelu výrobního systému s koncepcí Průmyslu 4.0. Robot je jedním z komponent vytvářejících Internet věcí v prostředí továrny. S ostatními prvky komunikuje prostřednictvím WIFI komunikace. Je nutné zvolit vhodný návrh řízení robota, který je kompatibilní s operací, kterou potřebujeme k provedení požadovaného úkolu. Součástí práce je nalezení kinematických rovnic pro pohyb jednotlivých ramen a jejich převedení na proměnné odpovídající polohám servomotorů. Cílem je dostat koncový bod ramene na požadované souřadnice. Programování robota se provádí pomocí řídicího mikroprocesoru a softwaru Arduino. V řídicím programu byly naprogramovány funkce, které umožní přemístit výrobek v modelové továrně z jednoho místa na jiné.

TABLE OF CONTENTS

TABLE OF FIGURES

List of Symbols

IOT - Internet of Things.

CPS - Cyber Physical System.

ITS - Intelligent Technical System.

AI - Artificial Intelligence.

AC - Alternating Current.

DC - Direct Current

1. INTRODUCTION

The world which is fast in growing and it is in a competitive environment which also comprises of day to day innovations and are also involved in seeing a lot of transformations in all aspects. From this we can say that the communications and transmissions of knowledge are being made real time through the usage of all the modern technologies. The developed countries has reached a position where they are able to frame the path of competitive industrial innovation through which the global market are developing in all the industrial sectors.

The main task of this work is to make an industrial arm or a robot arm work with the inputs and to make some calculations and to find an easy path for the robot arm to reach the target position which is mainly an **Educational Model** involved in the Industry 4.0. The working concept of the industry 4.0 is based upon the automation achieved through computational techniques. Industry 4.0 introduced the term called as the "smart factory" which is based upon the complete automation. Smart factories, which will be the main frame for all the industrial sectors operating in Industry concept 4.0, will take over the information and communication technology for an evolution in the supply chain and production line that would help to achieve much higher level of both automation and digitization.

The main aim of introducing the Industry Concept of 4.0 is to achieve the results which were not done using the others. This thesis work is mainly the concept of Industry 4.0 helping to create manipulator for the moving of objects within the working environment of the industry. The manipulator which is being used in this thesis is the robot arm, is being placed at one place in the Industry and helps to move the objects by usage of the grippers and movement takes place by the activation of each servo at different positions or angles of the arm. This operation is carried out by writing a arduino program to the robot arm feeding that as a input to the arduino board. The manipulator functions as a pick and place robot for placing the objects into the vehicle.

The reaching range of the arm is calculated with many methods and one simple method is finalized for the working of robot arm.

2. EVOLUTION OF THE INDUSTRY 4.0

The Industry 4.0 mainly stands for the fourth industrial revolution. Other related terms for the Industry 4.0 are Industrial Internet or the Digital Factory. The first revolution refers to the mechanical production which was mainly dependent on the water and steam power. The next comes the second revolution, this mainly concentrated on the Mass Production using the electrical energy. The Third revolution, the first programmable logic circuit (PLC) was used and the process was partially automated. Then comes the Industry 4.0 which is used in this generation were the technology is used to develop the organisation with the use of Internet of things and Cyber-Physical systems in order to make the product cycle with complete Automation Technologies. [1]

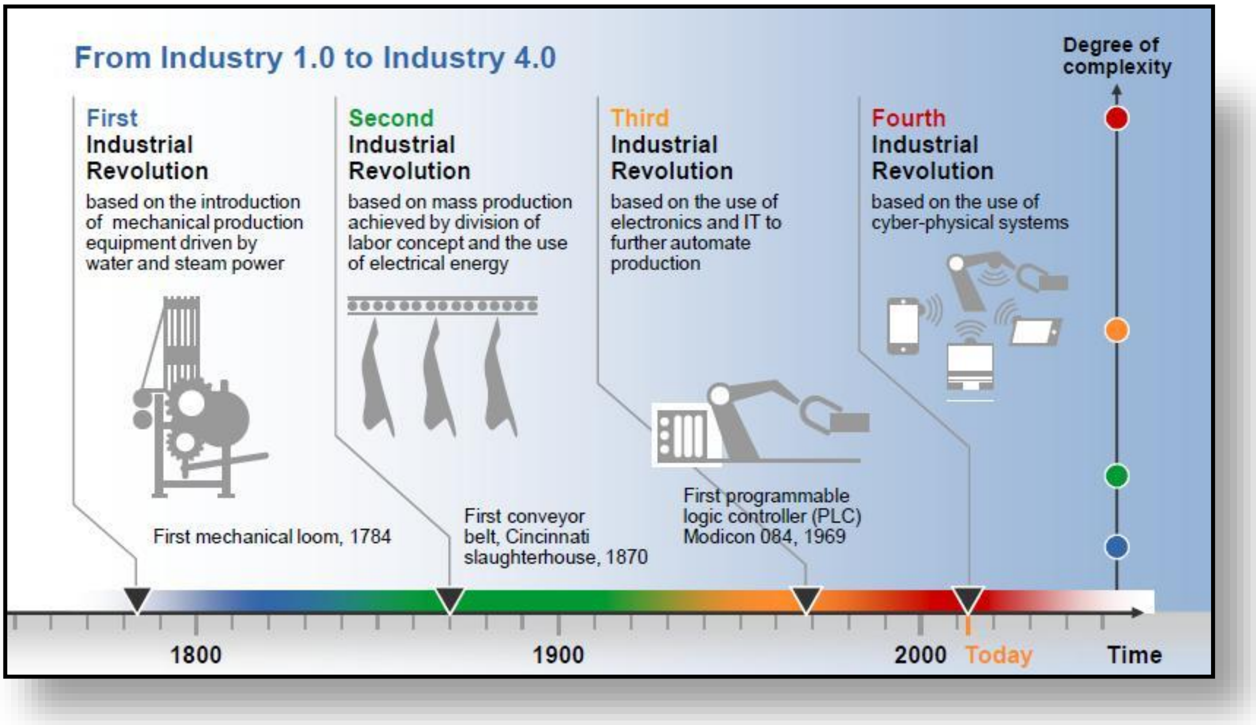

From Industry 1.0 to Industry 4.0
Degree of complexity
First Industrial Revolution
based on the introduction of mechanical production equipment driven by water and steam power
Second Industrial Revolution
based on mass production achieved by division of labor concept and the use of electrical energy
Third Industrial Revolution
based on the use of electronics and IT to further automate production
Fourth Industrial Revolution
based on the use of cyber-physical systems
First mechanical loom, 1784
First conveyor belt, Cincinnati slaughterhouse, 1870
First programmable logic controller (PLC) Modicon 084, 1969
1800
1900
2000
Today
Time

Figure 1 Evolution of industry 4.0 [1]

The productivity improvement benefits are mostly about reducing the costs of manufacturing and optimizing the project. Most of the projects are installed in order to completely satisfy the customer needs. If the production system is completely build up with sensors, software, IoT technologies, the quality of the products can be enhanced, the maintenance costs, labour costs can be reduced. Automation plays a very important role here and the typical components of Cyber-Physical Systems and the Internet of Things from which the quality can be checked in real-time and robots can easily decrease the errors. [2]

2.1 Major Principles of Industry 4.0

This concept of Industry 4.0 is mainly based on **FOUR Principles** which emphasis that the entire system should be computerized.

- Interoperability

This mainly refers to the capability of the machines and their related components to connect and interact with the customers through internet.

- Transparency in Information

This requires that the information systems should be able to create virtual copies of the physical world by converting the physical data into a sensor data. For achieving this sensor data which is a raw data has to be supported with compatible context data.

- Technical Assistance

This deals with the ability of the system to support humans through comprehensive gathering and vocalization of information for greater decision-making and spontaneous solution for all the problems. Technical Assistance also deals with the Cyber-enabled systems to support humans for handling various tasks that are considered to be time consuming.

- Decentralization of decisions

This refers to the capability of the cyber-enabled systems to autonomously come with customization of products in flexible manufacturing. [3]

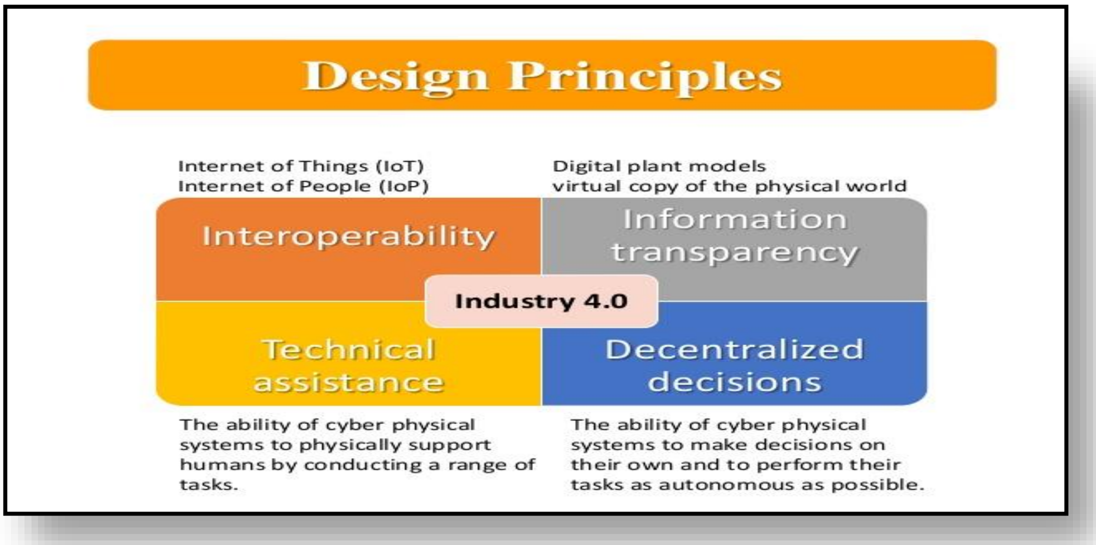

Design Principles
Internet of Things (IoT)
Internet of People (IoP)
Digital plant models
virtual copy of the physical world
Interoperability
Information transparency
Industry 4.0
Technical assistance
Decentralized decisions
The ability of cyber physical systems to physically support humans by conducting a range of tasks.
The ability of cyber physical systems to make decisions on their own and to perform their tasks as autonomous as possible.

Figure 2 principles of industry 4.0 [4]

3. SMART FACTORY

The virtual and physical worlds together gave rise to the Smart Factory. This includes artificial intelligence, machine learning, automation of work and communication between each machine with the manufacturing process. The Smart Factory will basically change how products are invented, produced and shipped. Simultaneously it will increase workers safety and will shield the environment by enabling fewer emissions. These advance technology were machines communicate and when the decision making moves from humans to technical machines then it can be said that the manufacturing process has become “smarter”. [5]

The Smart Factory concept unites many technologies to generate a new business model for industry. Among all the other things, it will be useful to produce highly customised products at affordable costs, with very low level of emissions and very less impact on environment. The outlook of the Smart Factory will be about the complex and extensive networks linking suppliers, manufacturers and customers. [6]

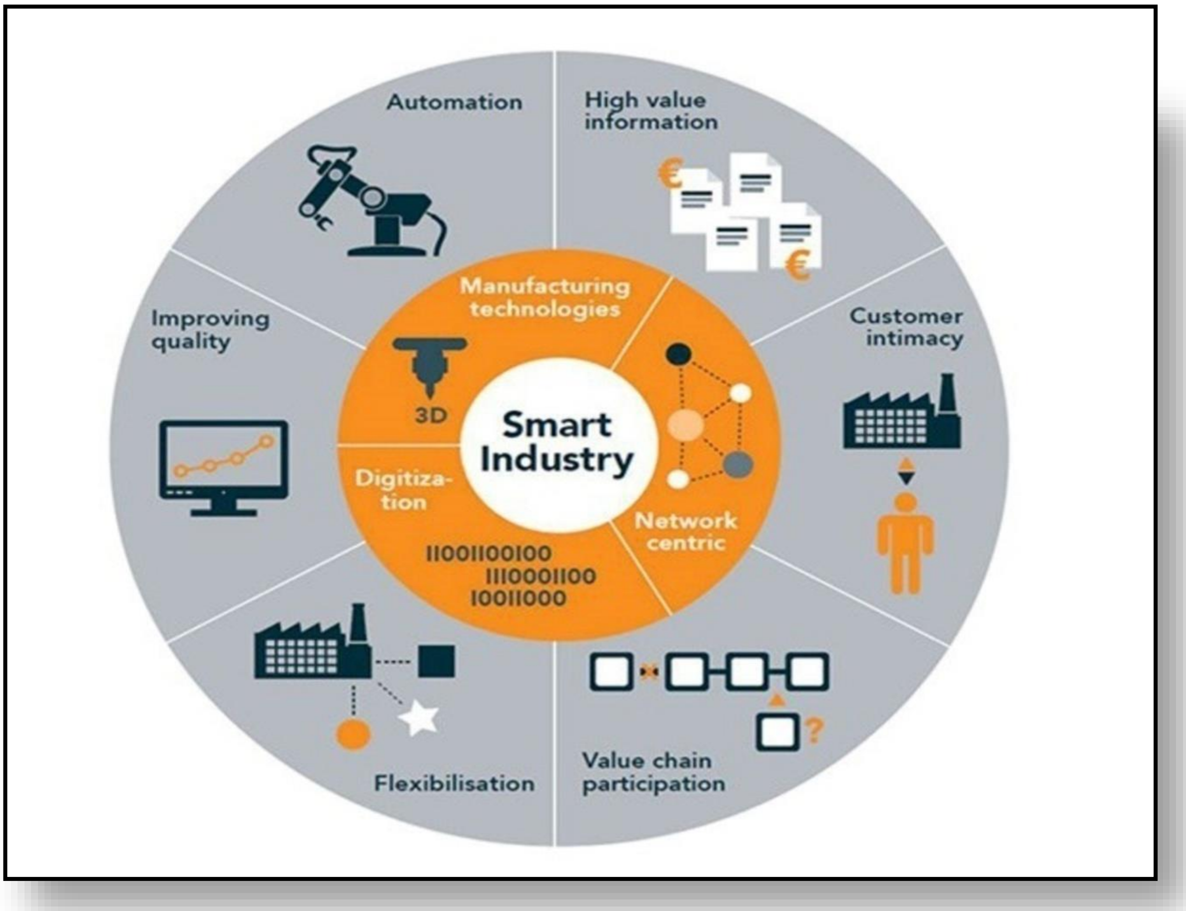


Figure 3 Smart Factory [7]

3.1 Cyber Physical Systems - New Generation of Production.

Cyber Physical System is the platform where a new generation of systems which are capable of integrating the abilities in a system with the computer. The industries which are using cyber physical system are able to provide the new revolution of Indutry 4.0, which is based on the way of access to the Internet and Physical Systems.

3.1.1 Platforms of Cyber Physical Systems [6]

Various basic platforms of Cyber Physical Systems are

- Integration of the Wireless System. [8]
- Wireless control System.
- Machine learning.
- Production - based on the sensors

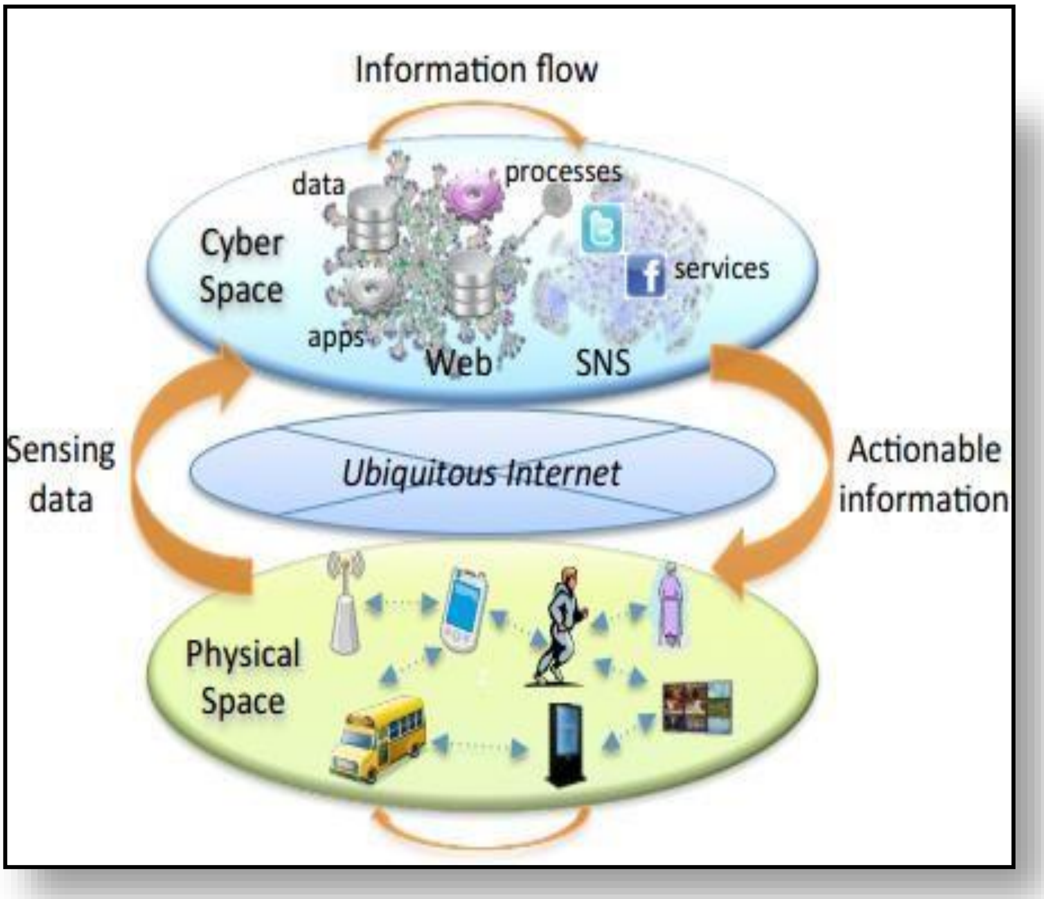


Figure 4 Flow of information between physical and cyber world [9]

4. ARDUINO-SINGLE BOARD MICROCONTROLLER

Arduino is defined as a single-board microcontroller to make use of electronics in multidisciplinary projects more accessible "A micro-controller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals". [13]

Arduino is an electronic platform which consists of Hardware and Software that can be easily used. Generally Arduino boards will be able to read all the inputs for examples a sensor light, a finger on a button and it will convert these into an output such as starting a motor, turning on a LED. We can instruct our board by giving a set of instructions and for this we will use an Arduino programming language and Arduino software. [14]

Arduino has many advantages over the other microcontrollers, Generally Arduino is much cheaper than the other micro controllers and a preassembled Arduino module can be purchased for $50., very simple and easily understandable.

The extensible software feature has added a major benefit for users, who can use a different programming language. Arduino can also be expanded with the other programming language like C and C++. [14]

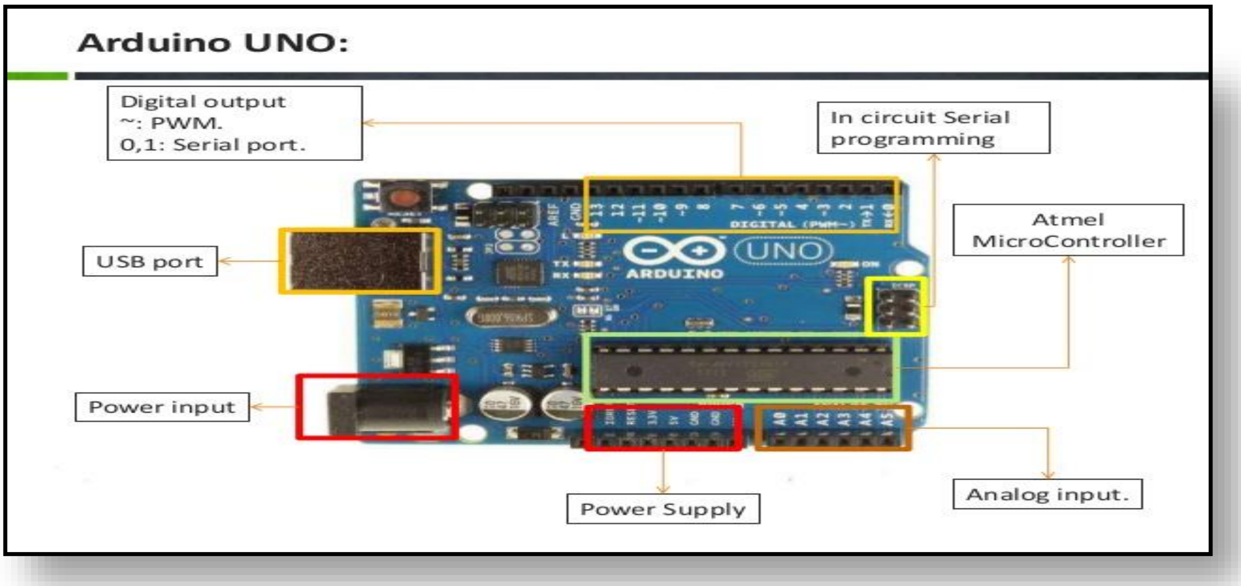


Figure 5 Basic Layout of Arduino Board [15]

4.1. Arduino Mega 2560

The conception of Arduino Mega actually originated from the ATmega2560. There are 54 digital input/output pins, out of which 15 pins can be used as PWM outputs, 16 pins as analog inputs, a 16 MHz crystal oscillator, a port for USB connection, a slot for power jack, an ICSP header and a reset button. The power for the board is given by using the AC - DC adapter or a battery. The main advantage of the Arduino Mega 2560 is that, it is compatible with almost all the shields designed for all other boards. [14]

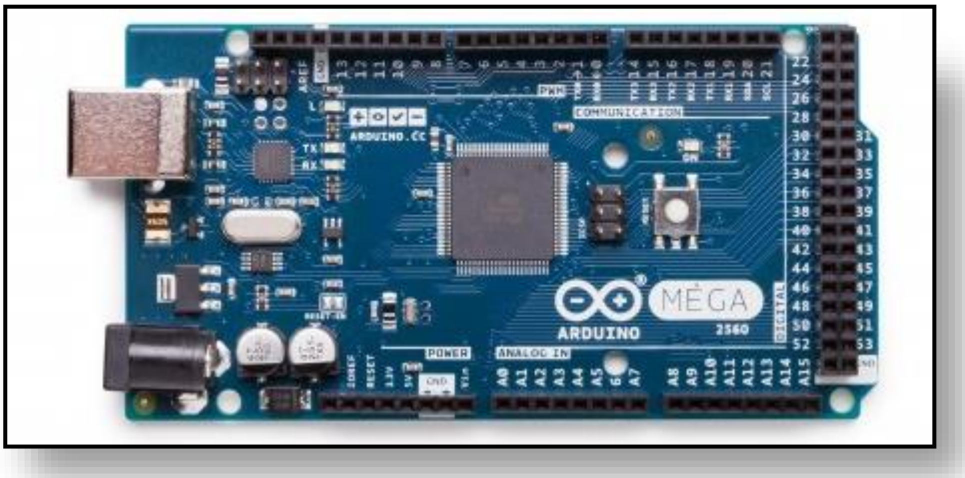
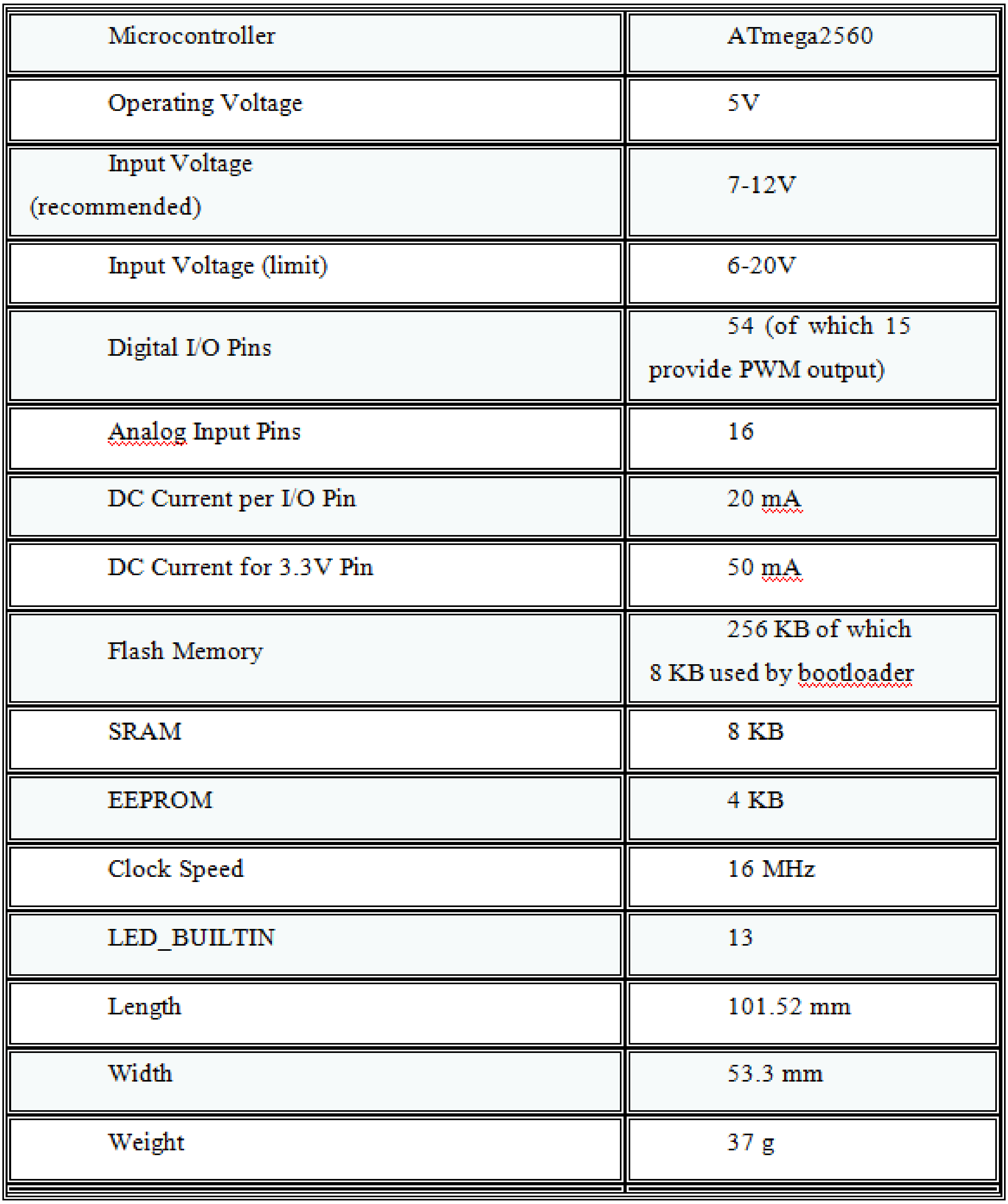


Figure 6 Arduino Mega 2560 [14]

The programs can be uploaded in the Arduino by using the Arduino Software. Additional advantage of the ATMega2560 is that, it is installed with a boot loader which can help the users in uploading new codes without using any external hardware programme. There is also a safety feature in the ATmega2560, that it is accompanied with a resettable poly fuse which will help in protecting the USB port of the computer. When the voltage exceeds 500mA the safety feature breaks the connection, until the high voltage or short circuit is removed. [14]

The power can be supplied to the Mega2560 by using extension wire which is connected to the PC’s USB port and the Mega2560 can also support the power given by an external power source. The Mega2560 has an advantage of detecting whether the power supply is being given from the USB port or it is an external source for instance by using an AC-DC adapter or battery. The power supply given to the board for operation can be varied from 6 to 20 volts [14]

.



Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Figure 7 Technical Specifications of Arduino Mega2560 [14]

The above table is all about the technical specification of the Arduino Mega 2560. Where some voltages like the Output voltage and the input voltage range is mentioned. The total number of pins in the board is which is mentioned and in those like 15 of them which gives PWM output. There are totally 16 input pins. The DC current per I/O pin is 20 mA.

The total memory of the Arduino board Mega 2560 is 256 KB out of which 8 KB used by bootloader. The total clock speed of the board is 16 MHz. The last information is about some physical dimensions of the board where the length is 101.52mm and the width 53.3mm and the total weight of the board is 37 g. [14]

4.1.1. Input Output Pins of Mega 2560

The advantage of Mega2560 is that all the 54pins can be used as the Input or Output pins by using certain commands like digitalwrite(), pinmode(), digitalread() functions.. The safety value or the maximum operating value should not exceed 40mA. The upper end range of the pin can be changed by using the AREF pin, even though the default measuring is from 5v to Ground. [14]

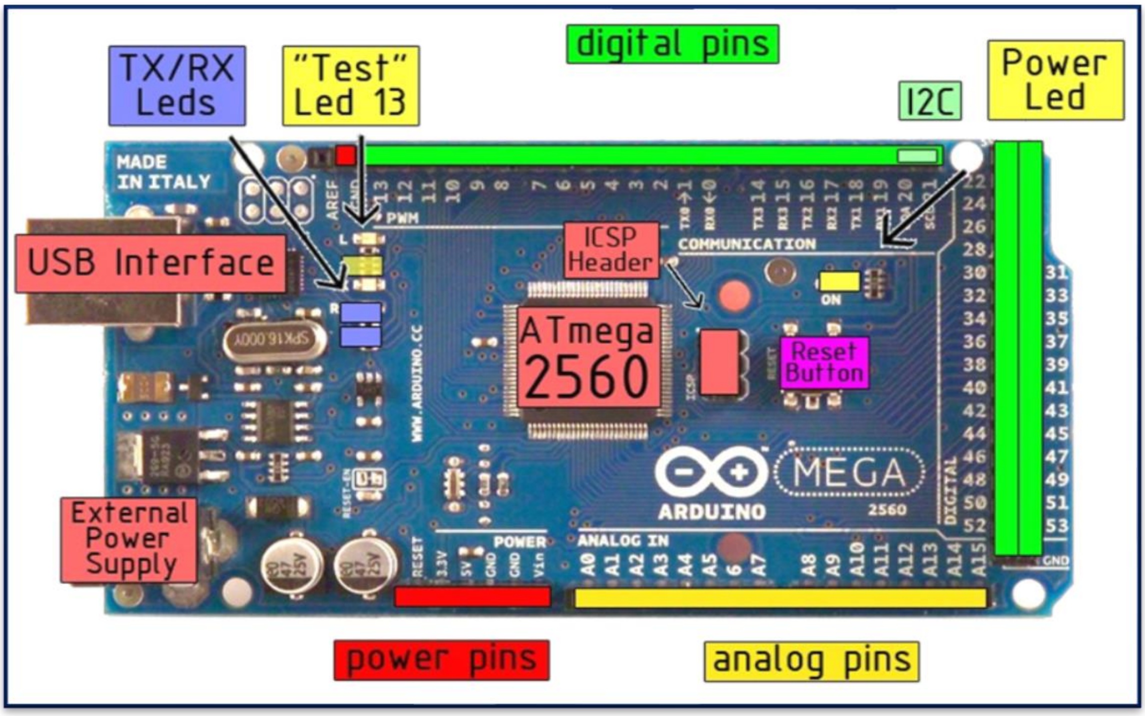


Figure 8 Pins OF Arduino Mega 2560 [14]

- V_{IN} - When we use an external power source like USB or via a power jack, this pin can be used for supplying the voltage.
- 5V pin - The purpose of the pin is to maintain output of 5V from the regulator in the Arduino board. The power is supplied to the board through a DC power jack or through a USB connector but not through the 5V pin, the board will be damaged if the power is given through 5v pin.
- 3.3V - The maximum current drawn from this pin is 50mA.
- GND pins - These pins are used for grounding. [14]

4.1.2. Dimensions of The Board

- Overall Length of the Mega 2560 board is 4 inches.
- The Width of the Mega 2560 board is 2.1inches.
- The power jack and USB port are extending beyond the length of the board.
- The distances between each pin is less than 0.16 inch.
- Serial Peripheral Interface (SPI) is also available through ICSP header in the Mega 2560.
- The Mega 2560 does not use the reset button before uploading, it is designed in such a way that it can be reset only by the software that is running in the system. When the Mega 2560 is connected to the computer through the USB port it automatically resets itself. The auto reset can be disabled by connecting a 110 ohm resistor from the 5V to the reset line. [14]

5. ROBOT ARM AND ITS COMPONENTS

The robot arm used in our smart factory is from the company AREXX engineering. This arm is capable of working and handling objects and things in an environment. These works are carried out by the robot mainly by the program that is been uploaded by using the Arduino software. The main tasks of the robot in our Industry 4.0 are to pick and place the objects.

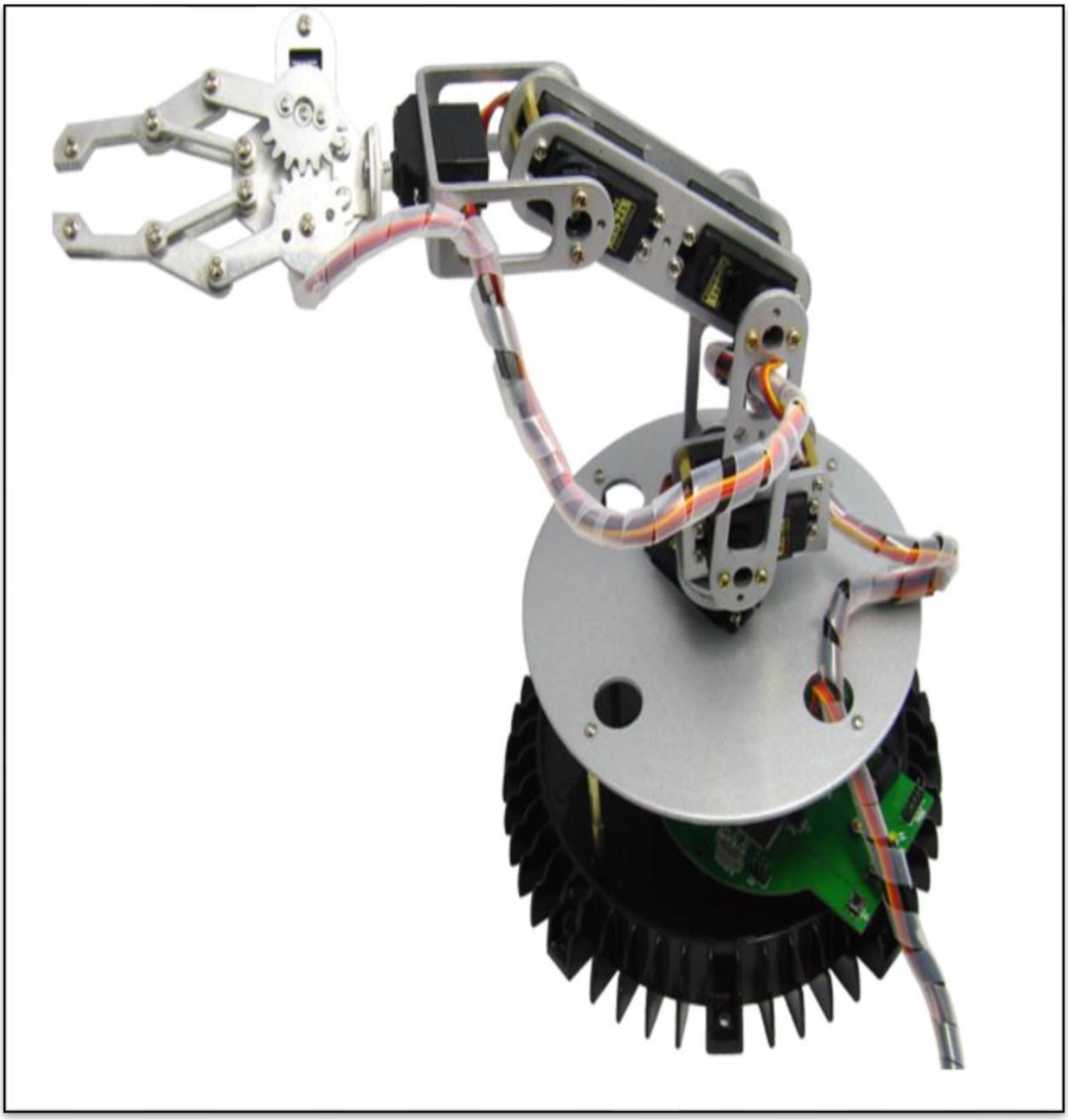


Figure 9 The Robot Arm In Industry 4.0

5.1 Specification of Robot Arm

- Processor used – Arduino Mega 2560
- Number of Servo Motors used – **6 Motors**
- Arm length – **390mm**.
- Material of the Arm – **Metal.**
- Height of the robot – **460mm.**
- Base Diameter – **210mm.**
- Power Supply – **9-14 Volts.** [16]

5.2 Pictorial Representation of All the Components of Robot Arm

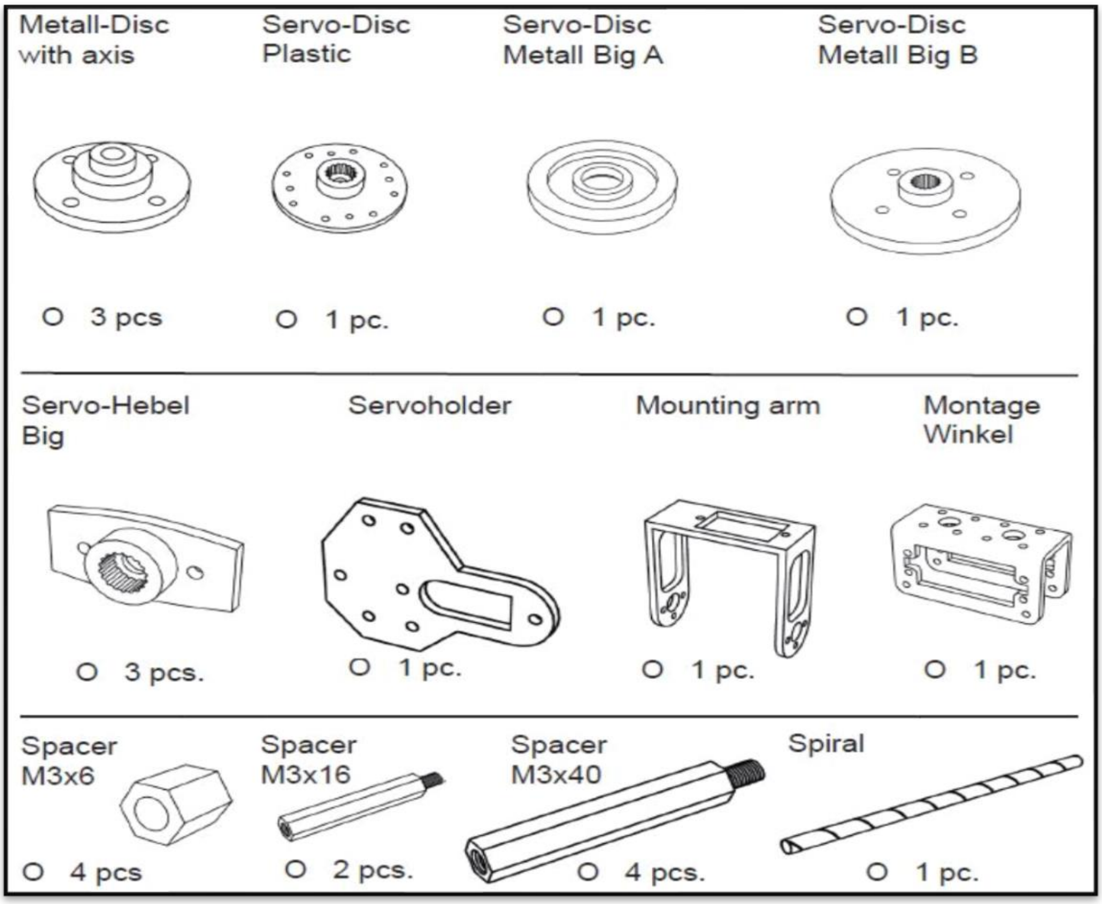


Figure 10 Mounting Parts Of The Robot Arm [16]

These parts are general which are used for mounting the robot arm for performing the functions. Mainly this image consists of parts that are required for mounting the servo of the arm.

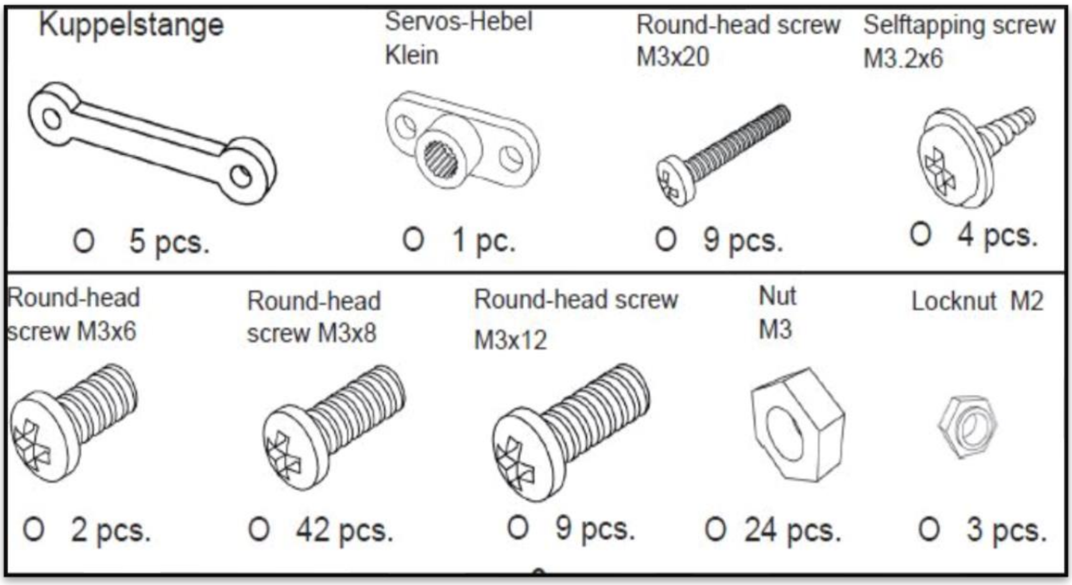


Figure 11 Fasteners Used in the Robot Arm [16]

Various kinds of Fasteners are used in the arm in various places just to fix each and every part of the arm. The image above shows the numbers of required fasteners of each type.

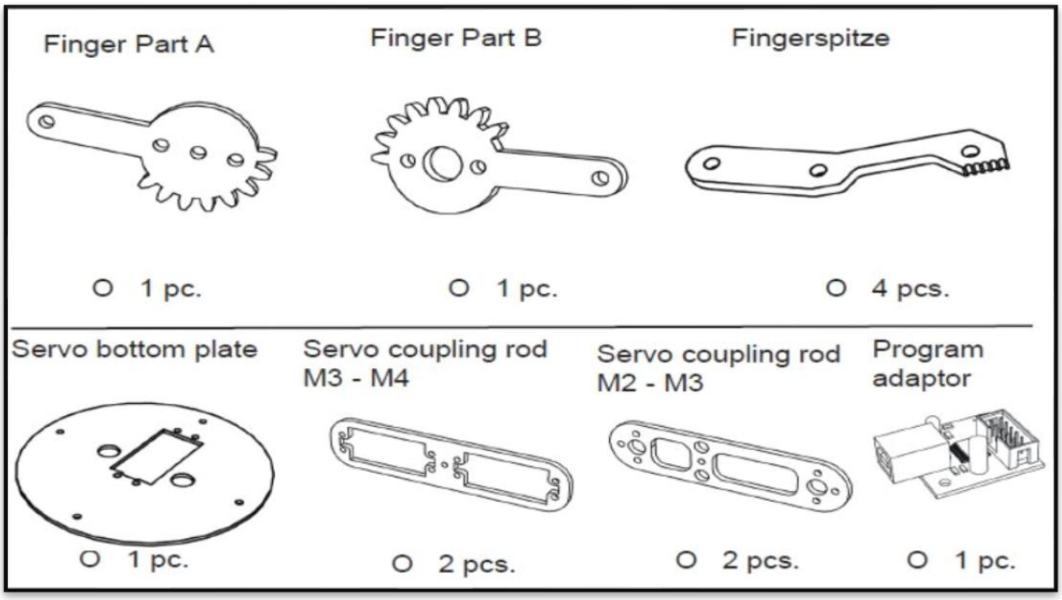


Figure 12 Parts of Gripper and Mounting Parts of Servo Motor [16]

The above picture shows the images of parts of end effectors and the parts for mounting the servo motor.

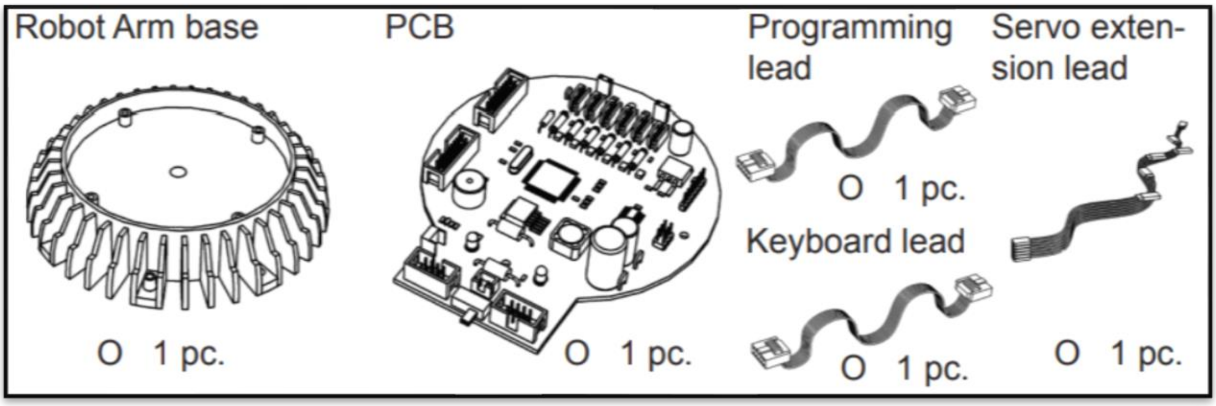


Figure 13 Essential Parts of Robot Arm [16]

In this figure there are some parts which are considered as the most essential parts of the robot arm. The Base part is where the entire arm is installed, PCB (Printed Circuit Board) and some leads for connecting some important parts of the robot arm. [16]

5.3 Steps in Assembly

The first step in the assembly of the robot arm is to set up the base plate and to mount the arm in the base plate. The base plate is installed with a servo motor for the rotational movement and for mounting the servo arm in the base plate we require some parts those are,

1 pc. Bottom plate of robot
1 pc. Servo Arm
2 pcs. Spacer **M3x16**
4 pcs. Round-head screw **M3x8**
2 pcs. Self trapping screw **M3.2x8** [16]

The servo motor is attached to each part of the arm for the purpose of producing the rotational movements. In the base plate the servo is mounted in a slot that is created with respective to the size of the servo used. This robot consists of **6 degrees of freedom** and we totally use 6 servo motors for each joint.

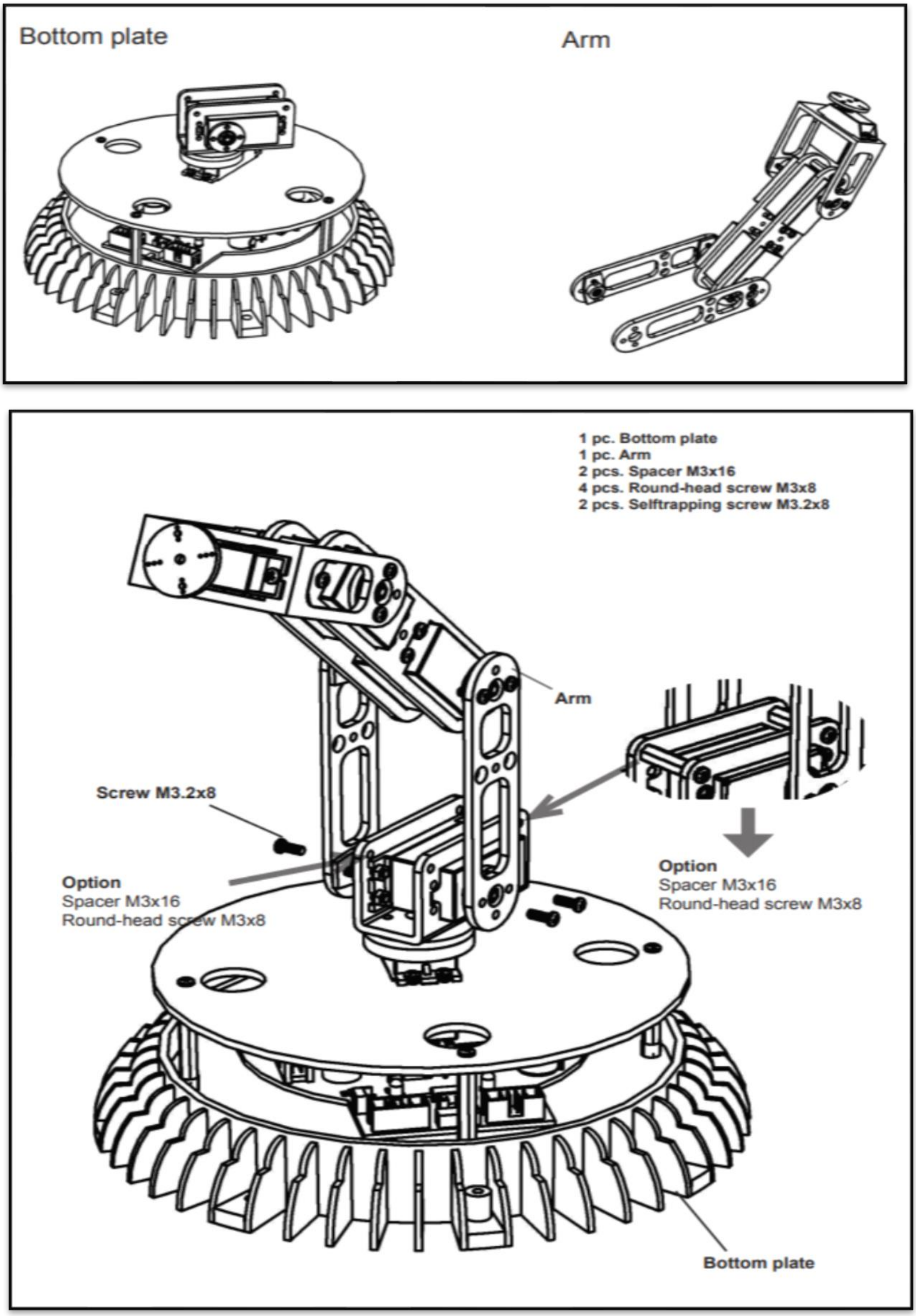


Figure 14 Assembly of Base and Arm Of the Robot [16]

The second part of the assembly is the Gripper which is also called as end effectors. This gripper has a claw in the end which performs the work like picking and placing the objects. The gear arrangement of the robot arm is also done by using the sets of screws and the gears for the better precision and accuracy of material handling during the operation of the arm. The power for the Gripper is supplied through the Servos, which are placed at the joints. The most important part of the robot arm gripper this is because this is the part which will perform the necessary action or command that being imported to the robot through programs. [16]

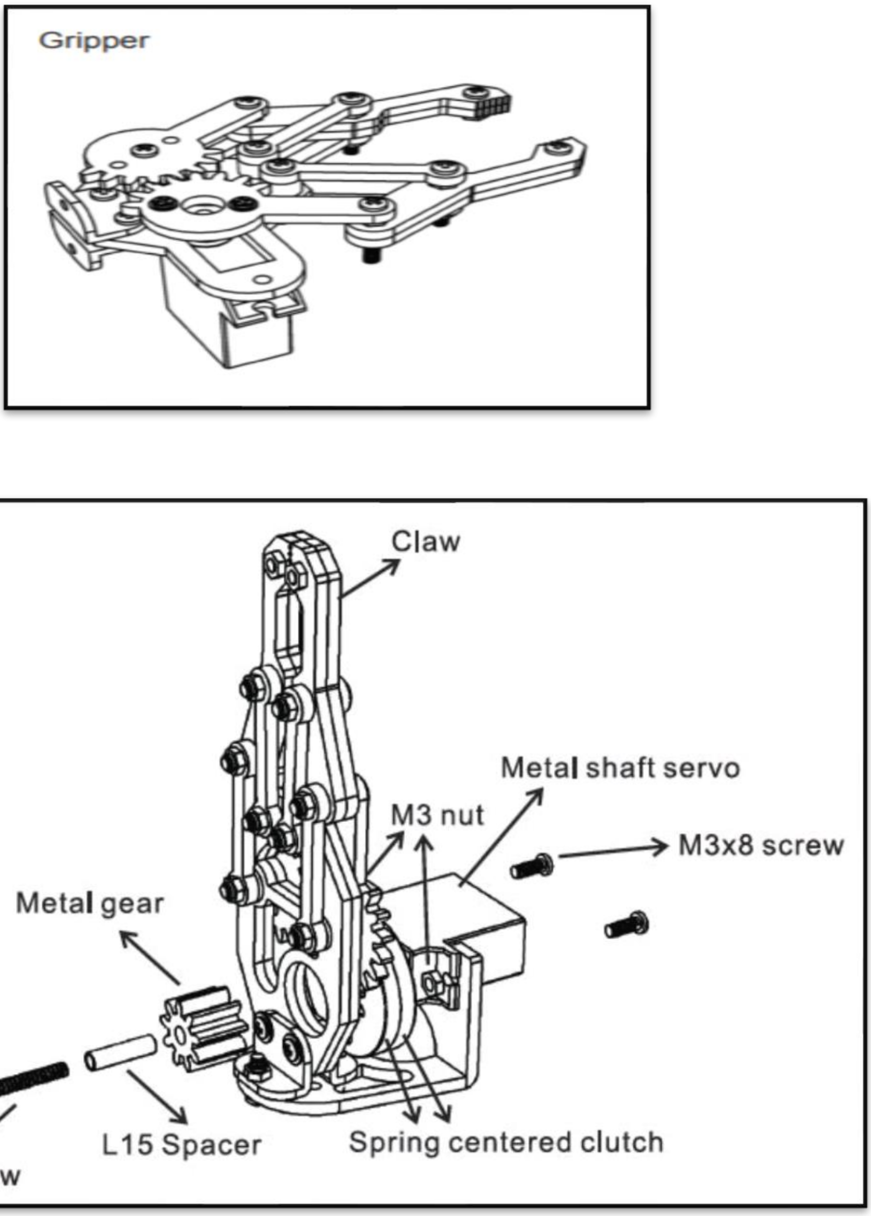


Figure 15 Assembly of Gripper Mechanism [16]

5.4 Printed Circuit Board of Robot Arm

The PCB is considered as the ‘Brains’ of the powered devices. They not only supply power to a device they also support mechanics because with the power they even route the signals to various components. The thickness of the PCB board is directly proportional to the complexity of the devices. A PCB generally consists of 1 to 10 layers. They are considered as the physical components of the electronic device. The board is made up of typically plastic or resin. There is a silkscreen on the top which show the connections.

To design and manufacture a PCB few steps are followed,

- Schematic design of the board.
- Board design with EDA software.
- Manufacturing.
- Assembling in a robot.
- Testing the board.

First the schematic design is created by the designer. It is a crucial step before designing and it should never be overlooked. The schematic contains series of symbols such as circuit, switch, diodes, resistors, nodes and more.

After the circuit schematic design it is been translated to an Electronic Design Automation (EDA). Then it has to be exported to an industrial format and the result acts a set of instructions for production phase of PCB.

Once when the design automation is over then the PCB is ready for the manufacturing process. The dimensions are based on the number of layers and complexity of the project.

So the final step is the white box testing which makes sure that the PCB is structured properly after its been fabricated. The result is then compared black box testing which only tests input and output data’s. [16]

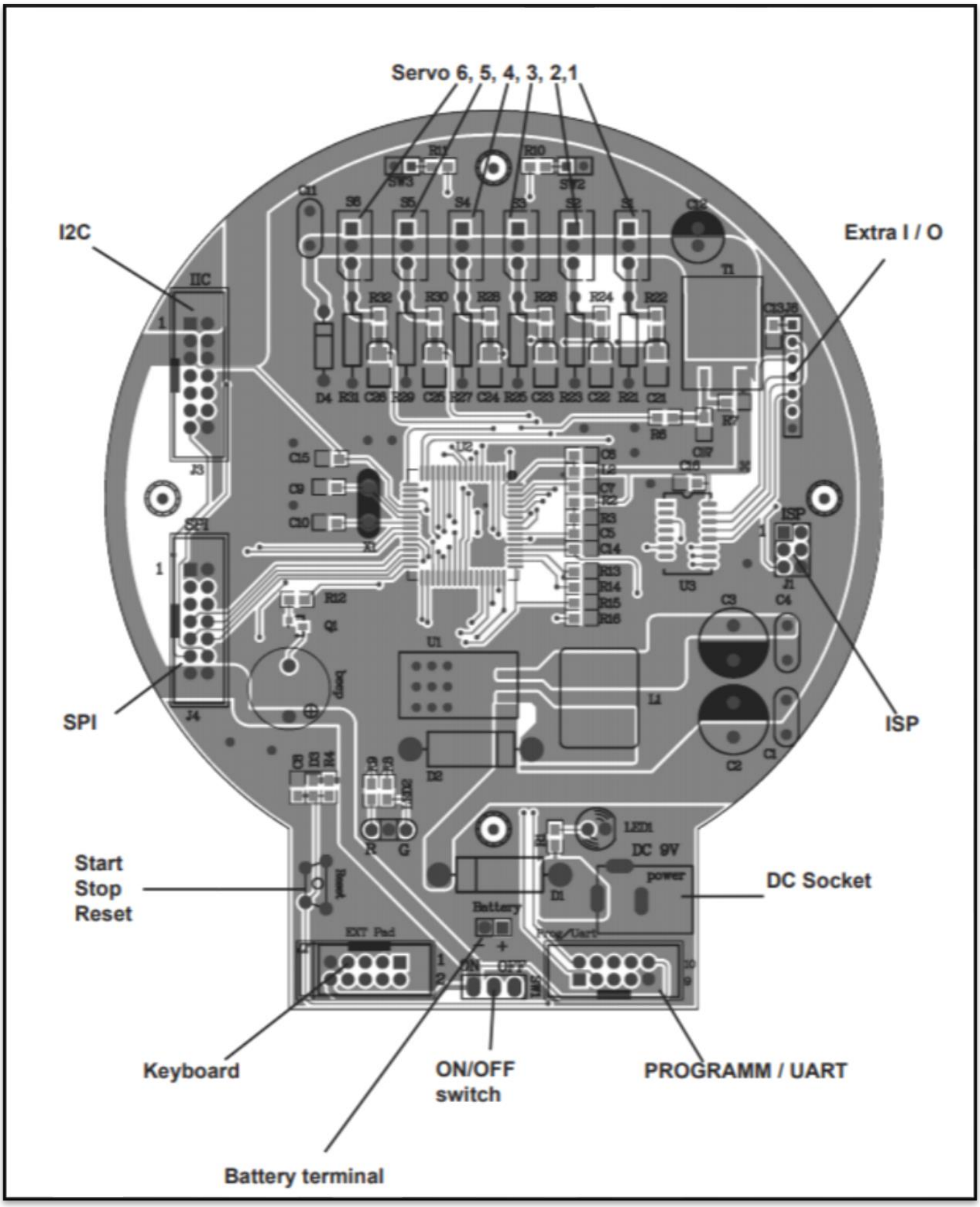


Figure 16 PCB of Robot Arm [16]

5.5 Servo Motor Used in Our Robot

A Servo Drive can also be referred to as an amplifier since it takes the signal from the controller and amplifies it to deliver a certain voltage and current. An Electric Servo Motor is an Electric Machine that **converts Electrical Energy into Mechanical Energy**. Most Electric Motors operate through the interaction between an Electric Motor’s Magnetic Field and Winding Currents to generate Force within the Motor.

Servomechanism is a type of control system in which a small signal or a small force is used to control a much larger force and in which output accurately follows the input even if it is varying rapidly. The **system constantly compares the input and the output until the error signal becomes zero.**

5.5.1 Functions of Servo Mechanism

- The Primary Task of a servomechanism is to maintain the output of a system at the desired value in the presence of disturbances.
- Accurate control of motion **without** the need for **human attendants** (**automatic control**).
- **Maintenance of accuracy** with mechanical load variations, changes in the environment, power supply fluctuations, and aging and deterioration of components (regulation and self-calibration).
- Control of a high-power load from a low-power command signal (**Power Amplification**).
- In many applications, servomechanisms allow high-powered devices to be controlled by signals from devices of much lower power. [17]

5.5.2 Construction of Electric Drive System:

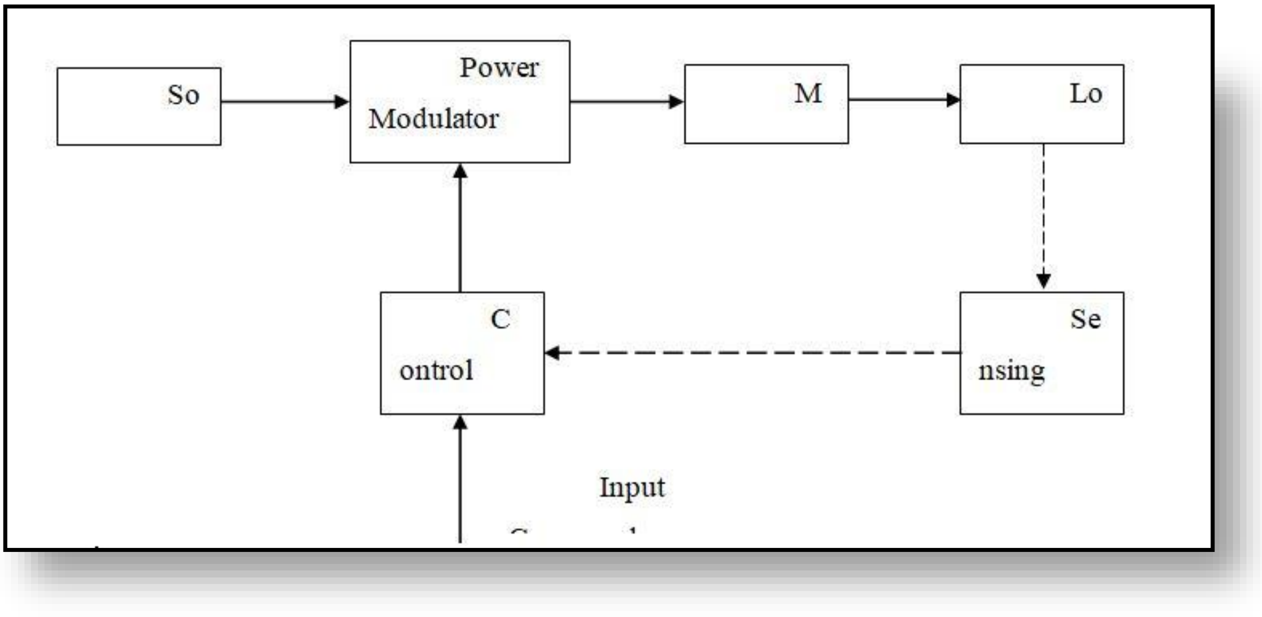


Figure 17 Block Diagram of Electrical Drives [17]

Mainly, an Electrical Drive System has Electrical Motors, Load, Power Modulator, Sources, Control Unit and Sensing Unit.

ELECTRICAL MOTORS:

Most commonly used Electrical Motors are **DC Motors** (Shunt, Series, Compound and Permanent Magnet), **Induction Motors** (Squirrel Cage, Wound Rotor and Linear), **Synchronous Motors** (Wound Field and Permanent Magnet), **Brushless DC Motors**, **Stepper Motors**, **Ring Motors**, etc.,.

- POWER MODULATORS:

Some Drives may employ more than one of these Modulators. It can be classified into

- ➢ Converters (AC to DC)
- ➢ Invertors (DC to AC)
- ➢ AC Voltage Controllers (AC to AC)
- ➢ DC Choppers (DC to DC)
- ➢ Cyclo Convertors (Frequency Conversion) [18]

- ELECTRICAL SOURCES:

In India, Single Phase and Three Phase 50 Hz, AC Supplies are readily available in most locations. Very Low Power Drives are generally fed from Single Phase Source. Rest of the Drives are powered from 3 Phase Source. Low and Medium Power Motors are fed from 400 V supply. For higher ratings, motors may be rated at 3.3 kV, 6.6 kV, 11 kV. Some Drives are powered from a batter, battery voltage may have 24 V, 48 V and 110 V DC.

- SENSING UNIT:

  This can be majorly categorized as

  - Speed Sensing
  - Current Sensing

Speed Sensing is required for implementation of closed loop speed control schemes. Speed is usually sensed by using **Tachometers**. When very high speed accuracies required, as in computer peripherals and paper mills etc., Digital Tachometers are used.

Two commonly used methods of the sensing the current (a) use of **current sensor employing Hall Effect** (ii) It involves the use of a non-inductive resistance shunt in conjunction with an isolation amplifier which has an arrangement for an amplification and isolation between power and control circuits.

- CONTROL UNIT:

Controls for a Power Modulator are provided in the Control Unit. Nature of the Control Unit for a particular Drive depends on the Power Modulator that is used.

When Semiconductor convertor are used, the Control Unit will consist of firing circuits, which employ Linear and **Digital Integrated Circuits and Transistors**, and a **Microprocessor** when sophisticated control is required.

5.5.3 Servo Motors:

Servo Motors are nothing but a simple Electrical Motor, controlled with the help of servomechanism. If the motor as controlled device, associated with servomechanism is DC motor, then it is commonly known DC Servo Motor. If the controlled motor is operated by AC, it is called AC Servo Motor. [17]

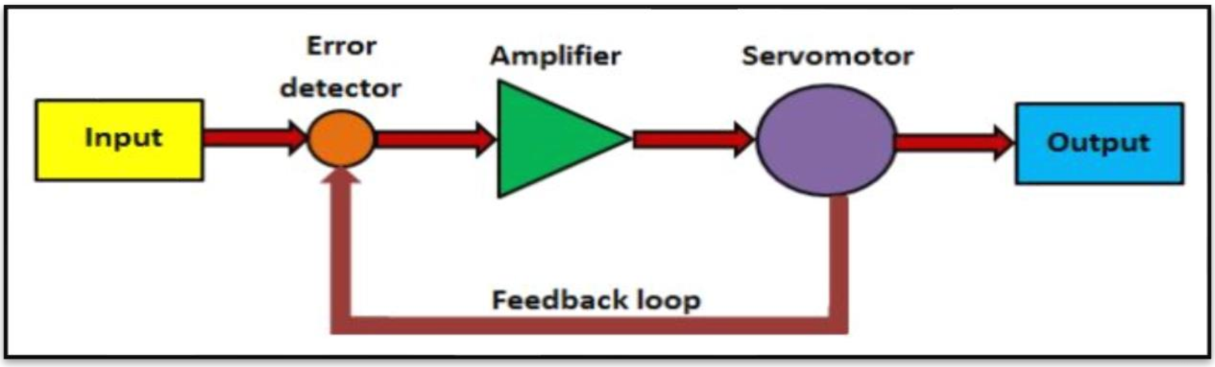


Figure 18 Servo Motor [17]

Servo Motor is a special type of motor which is automatically operated up to certain limit for a given command with help of error-sensing feedback to correct the performance.

The main reason behind using a servo is that **it provides angular precision**, i.e. it will only rotate as much we want and then stop and wait for next signal to take further action. This is unlike a normal electrical motor which starts rotating as and when power is applied to it and the rotation continues until we switch off the power.

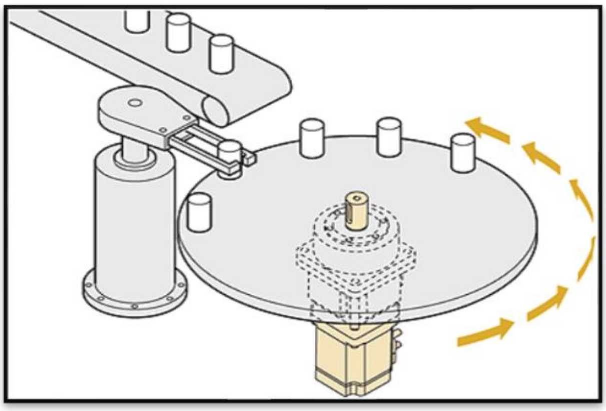


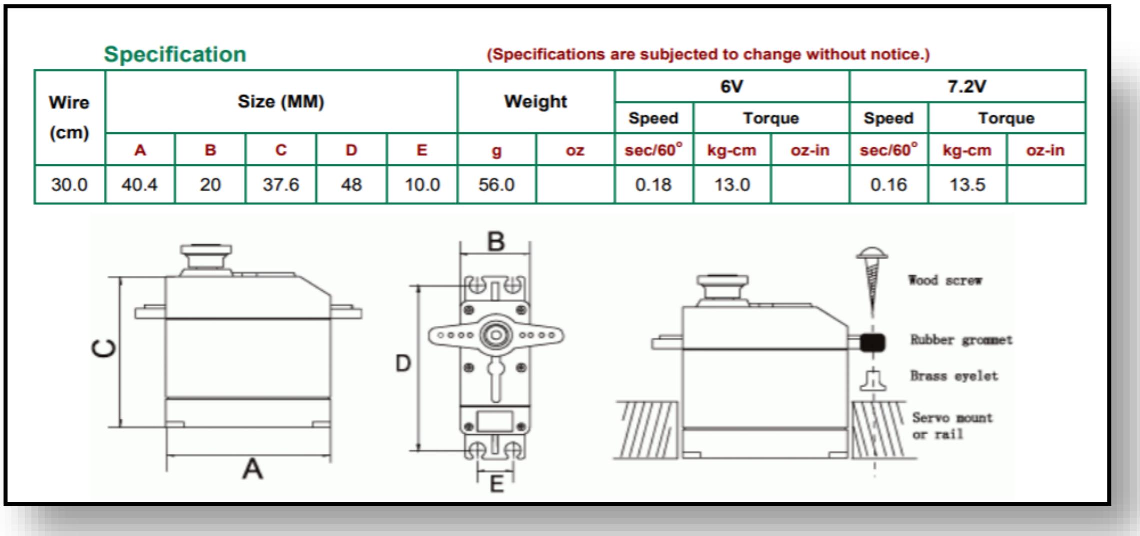
Figure 19 Application OF Servo Motor

The Servo Motor fitted inside the table gives angular motion in a periodic time interval. At the Mean Time, the Robot Arm shifts the object from the Table to the Conveyor Belt. The servo motor is specialized for high-response, high-precision positioning. As a Motor, capable of accurate rotation angle and speed control, it can be used for a variety of equipment.

The main difference between the Servo Motor and Stepper Motor is Power Consumption. A servomotor consumes power as it rotates to the commanded position but then the servomotor rests until the next Input Signal. Stepper motors continue to consume power to lock in and hold the commanded position. [17]

5.5.4 Servo Motor Used in Our Industrial Robot Arm

There actually 6 servo motors are used in our robot arm. The name of the servo is **DG Servo S07NF STD.** Specification of the motor is shown in the figure. Some things should be noted for the servo motor, Specify the connector type when you purchase the servo. For some applications rubber must be used for reducing vibration. Correct model of servo should be selected. Overloading Torque will damage the servo mechanism. It should be kept clean and away from corrosive gases. [16]



Specification (Specifications are subjected to change without notice.)

Wire (cm)	Size (MM)					Weight		6V			7.2V		
								Speed	Torque		Speed	Torque	
	A	B	C	D	E	g	oz	sec/60°	kg-cm	oz-in	sec/60°	kg-cm	oz-in
30.0	40.4	20	37.6	48	10.0	56.0		0.18	13.0		0.16	13.5	

C
A
B
D
E
Wood screw
Rubber grommet
Brass eyelet
Servo mount or rail

Figure 20 Specification of DG Servo S07NF STD [16]

5.5.5 Schematic arrangement of Servo and amplifier

This arrangement below shows the need of the amplifier is to step up or increase the current that is being supplied to the servos, with this increase in the voltage the servos are able to work effectively, to catch or release the object based upon the needs of the operation to be done. The resistors, optical diodes are all connected in series on the breadboard.

This show all the connections for the 6 servo motors present in the arm and the amplifier which is used to amplify the signal that is sent to the servo motor. In this each servo as its own function there are totally 3 arms and three servos are used for the motion of these three arms and one is used for the base rotation and rest two is used for the end effectors.

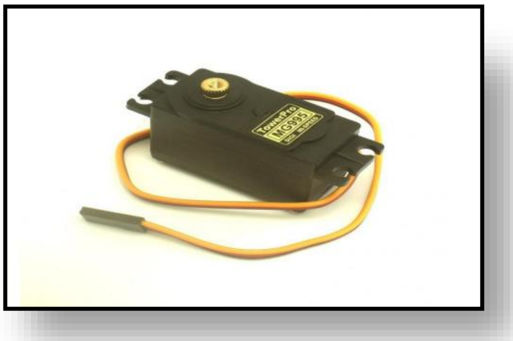


Figure 21 Servo Motor [19]

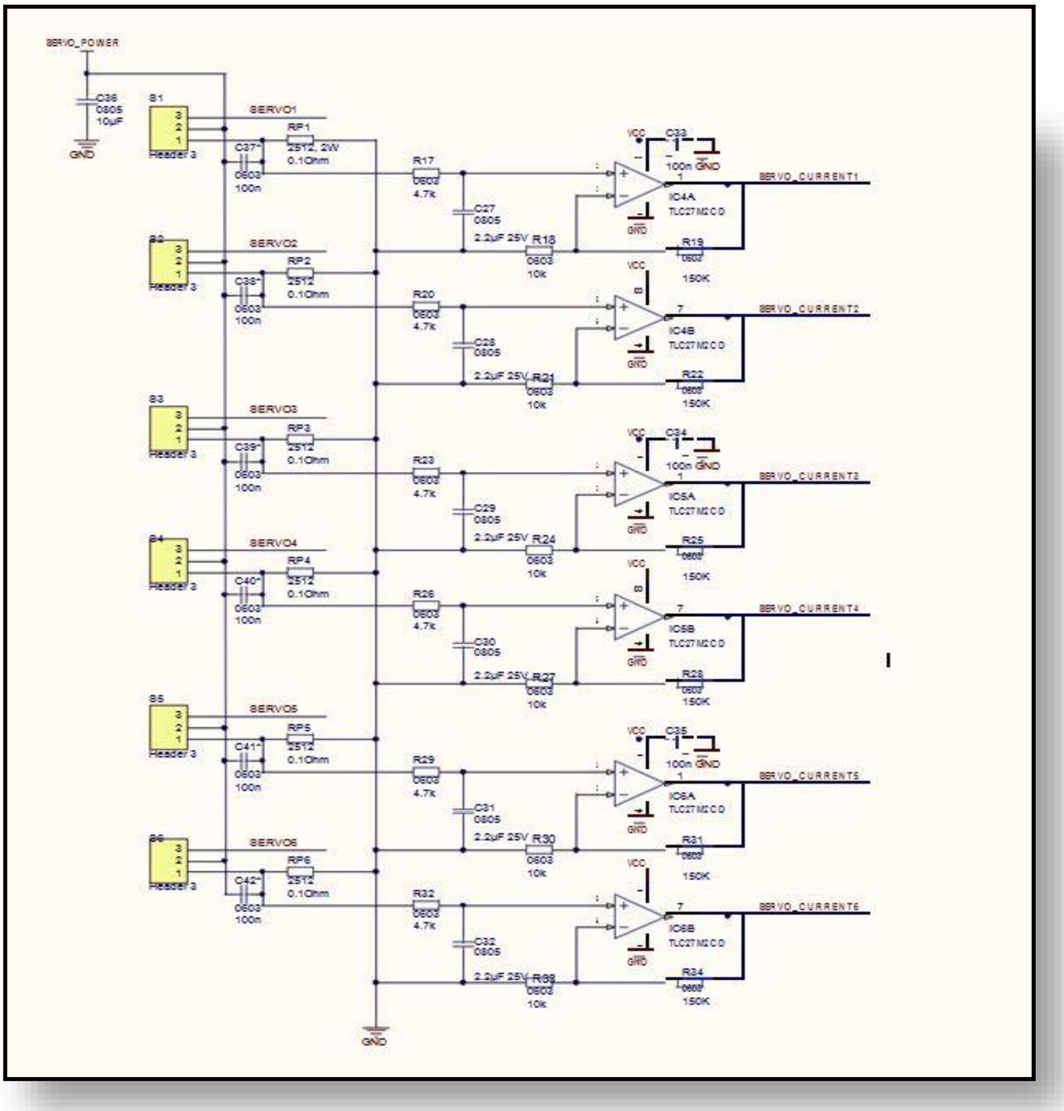


Figure 22 Schematic Diagram of the Servo and Amplifier [16]

5.5.6 Programming of servo motor.

Here is a sample code for the servo motor which is used in the robot arm. We totally use 6 servo motors in the robot arm and each motor serves for each arm in the robot. [14]

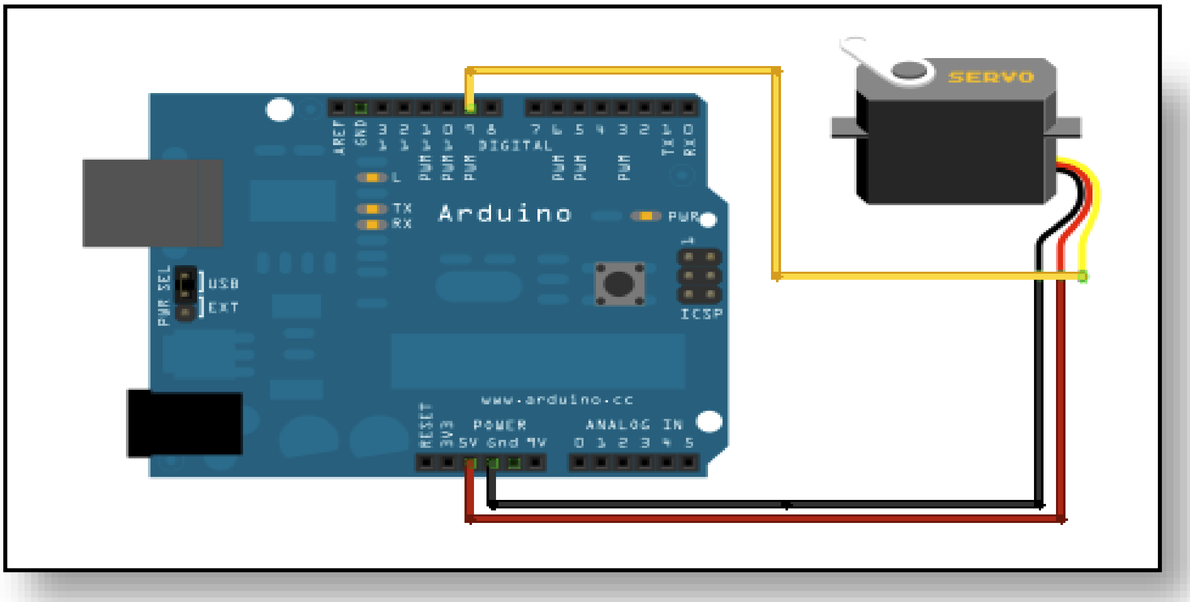


Figure 23 Connection of servomotor and arduino [14]

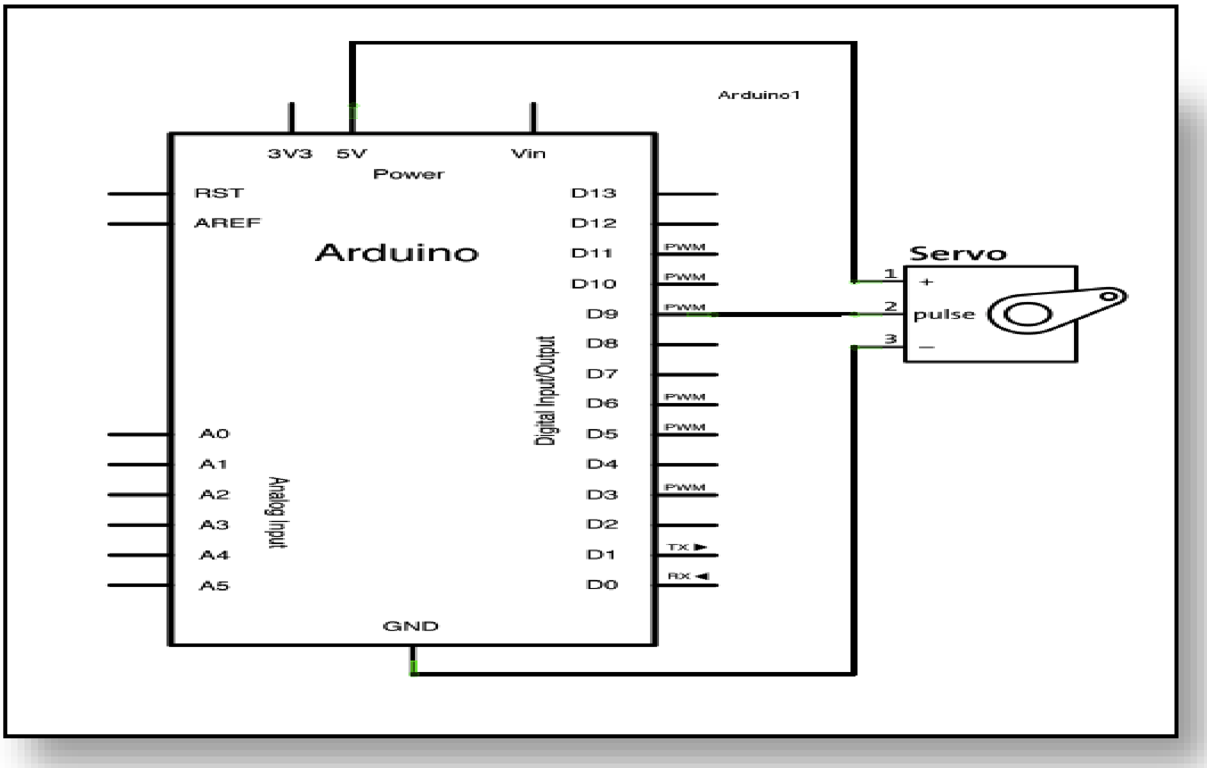
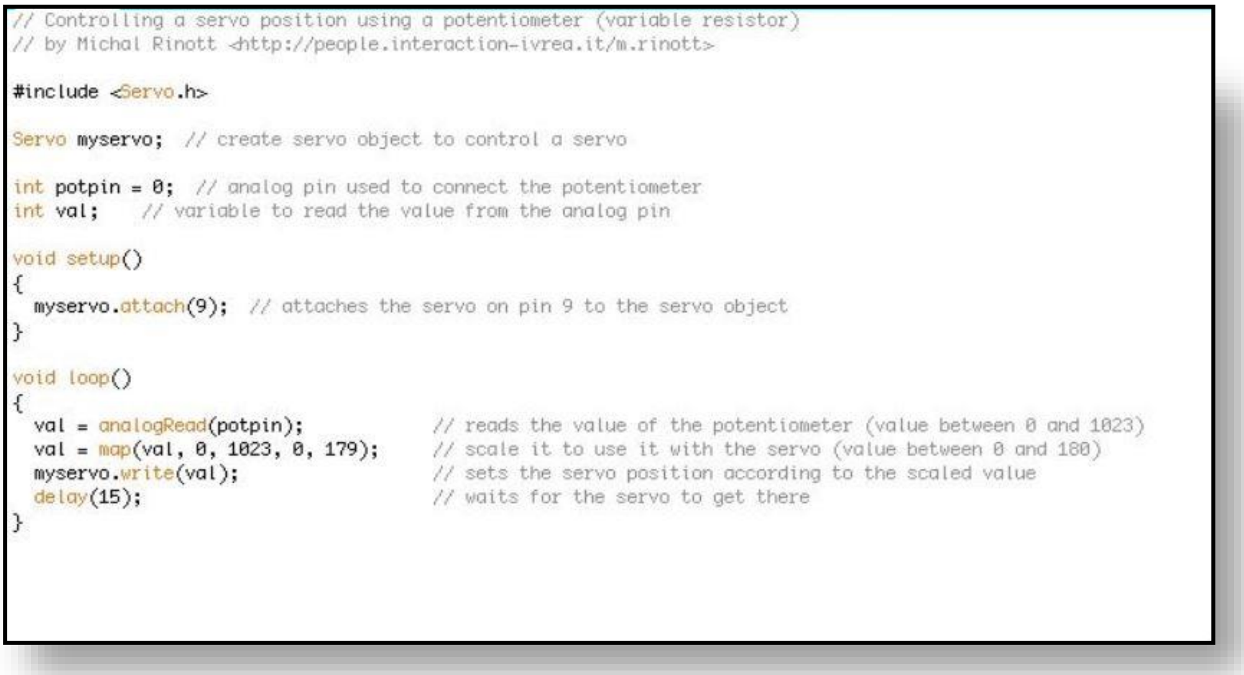


Figure 24 sample connection of Servo motor [14]



```
// Controlling a servo position using a potentiometer (variable resistor)
// by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>

#include <Servo.h>

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

int potpin = 0;  // analog pin used to connect the potentiometer
int val;    // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin);            // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 179);     // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val);                  // sets the servo position according to the scaled value
  delay(15);                           // waits for the servo to get there
}
```

Figure 25 Sample program of Servo motor [14]

5.6 WIFI Module

Arduino WiFi module is a wireless module that allows communication between Arduinos when they are used in different places. The working frequency is 2.4 GHz and this frequency is generally used for connecting computers or phones with wireless routers. The power requirement for wireless module is approximately 3.3 volts. The transmission power can be set at four levels from MIN to MAX, but for HIGH and MAX it is suggested to use an external power source of 3.3 V power supply because maximum current is no longer sufficient for these transmitting power, who can put a stabilizer on Arduino boards.

This wireless module consumes hundreds of mill amperes for a short time when transmitting and receiving datas, so it is always adviced to connect a 10 micro Farad capacitor between the 3.3V supply voltage and the ground, in addition to an external 3.3V supply.

The wifi module consists of 8 pins and the connection to arduino is very specific. The wiring of the wifi module to the arduino is shown diagrammatically below. The wifi module must not be connected to a 5v supply as this may result in the malfunction of the module and hence extra care must be taken while giving the connections.

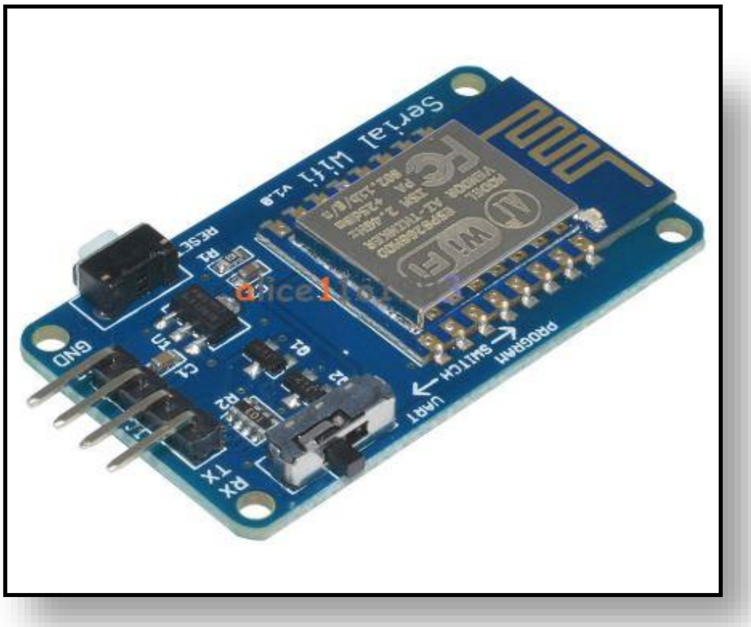


Figure 26 WIFI Module [14]

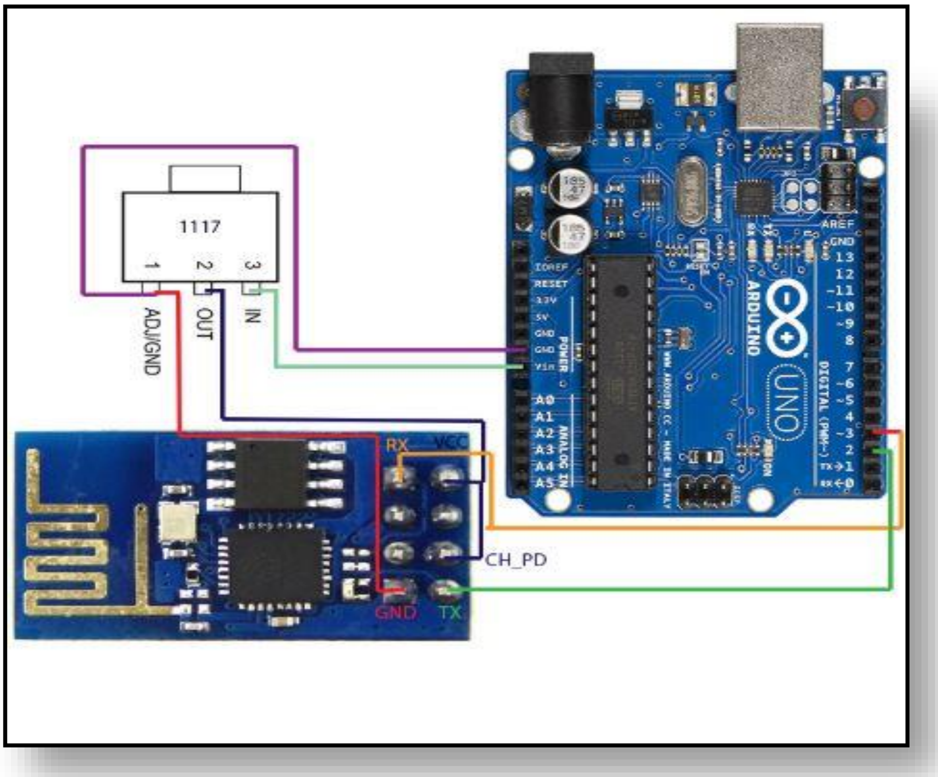


Figure 27 Basic Connection Of WIFI module [14]

5.7 Power Supply Convertor

This chip is used for converting the input signals corresponding output signal which is required for the robot. In personal testing I measured the lowest output voltage is around 1.23V for the input supply of 5V and 12V. [14]

Step-down converter with LM2596 - Power supply, converter

- Step-Down Inverter with Integrated Circuit LM2596.
- Efficiency: up to 92% (the higher the input voltage, the higher the efficiency).
- Switching frequency: 150kHz.
- Short circuit protection.
- Operating temperature: -40 to +85 ° C (output power 10W or less).
- Input voltage: 4.5 - 40V.
- Output voltage: 1.5 - 35V.
- Output current: 2A without additional cooler, 3A with cooler.
- Dimensions: 43 x 20 x 14mm. [14]

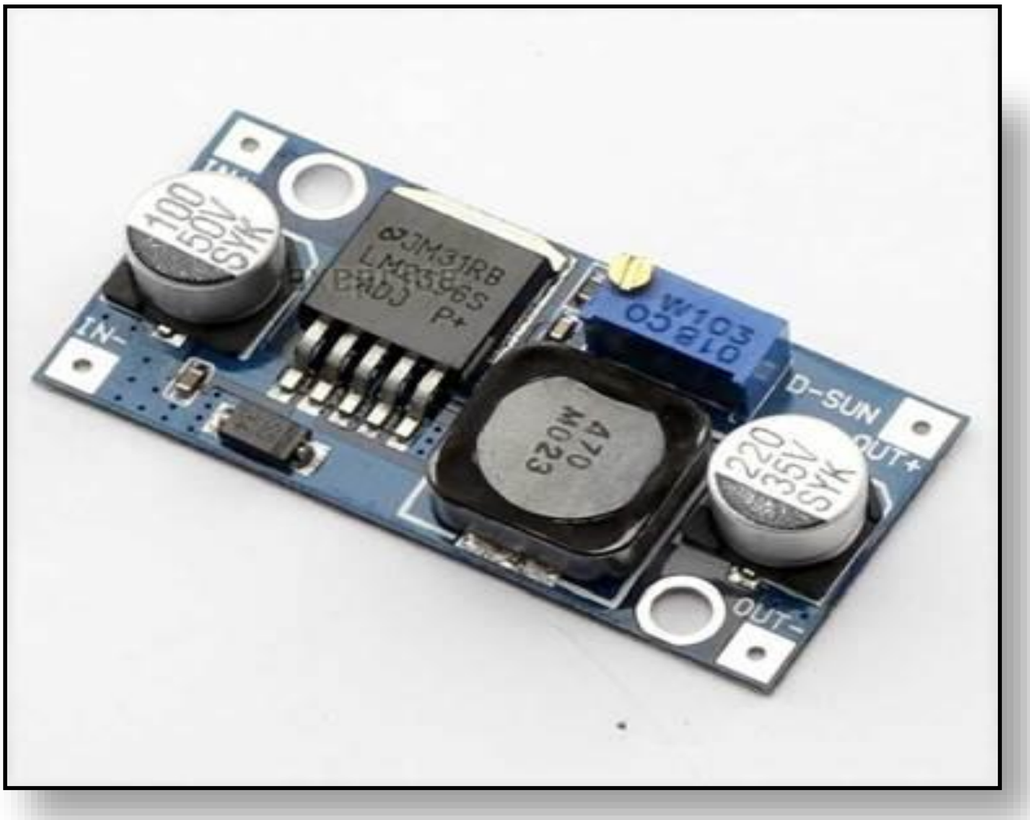


Figure 28 Power Supply Convertor [14]

6. METHOD OF SOLVING FOR THE ANGLES OF SERVO MOTORS.

This is the main part of the project work where we should find the relations between the angles of the servo and the target coordinate point the robot arm. The main aim is to make the simplest equations as the result which can bring more accurate results. I tried them in three methods one is inverse kinematic method of solving the equations, the second one is the simple geometry means of constructing the geometry and finding relations and the third and the successful method was the using some vectors and some trigonometric equations and solving and finding the simplest equation.

The equations which was found in the inverse kinematics method is in the degree of of 6 and 8 which is impossible to import in the arduino board the example is mentioned below so this method because of the big equations this method was excluded. The second method is the geometrical method which didn't give the result that is required by us. The third method gave us a result with a equation which has two degree of freedom and the simplest form of equation which can be given to arduino as input.

Now there is discussion about each method and the method of solving is explained with the help of figures.

6.1 First Method of Solving:

Our robot arm has 4 degrees of freedom excluding the end effectors.
The first approach for the robot arm to reach the desired position is considering,

a,b,c = lengths of the arms in the robot manipulator.
o_1,o_2,o_3, m = points of the joints from the diagram
Points m,o_1,o_2,o_3 are in the same plate P.
α_o serve to rotation of this plate P.
Every point may be reached or described with α_o and 2 coordinates points in plate P.
World System : M(x,y,z)
Robot system : M (α_o, x_p, y_p)

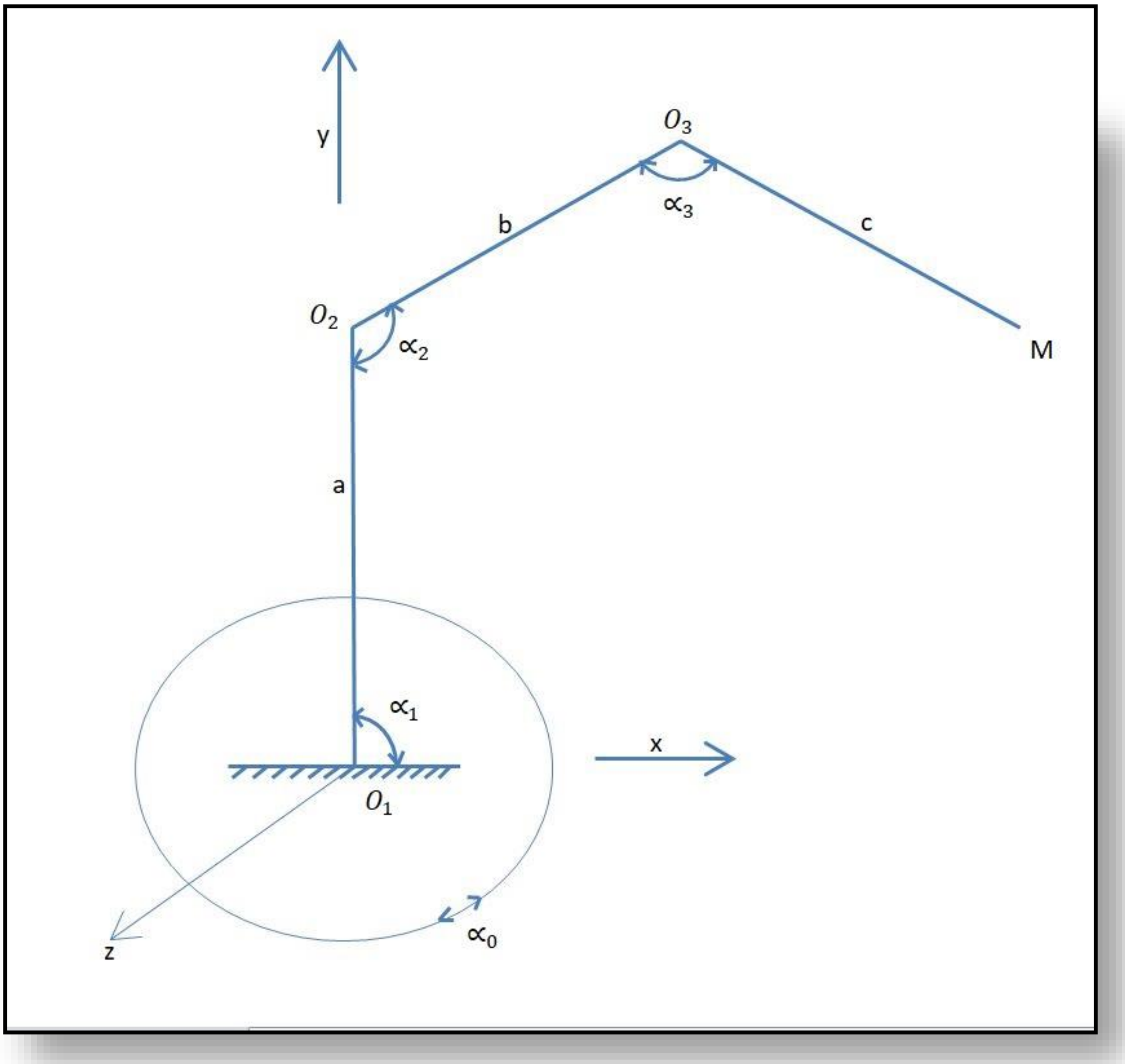


Figure 29 Schematic Robot Arm

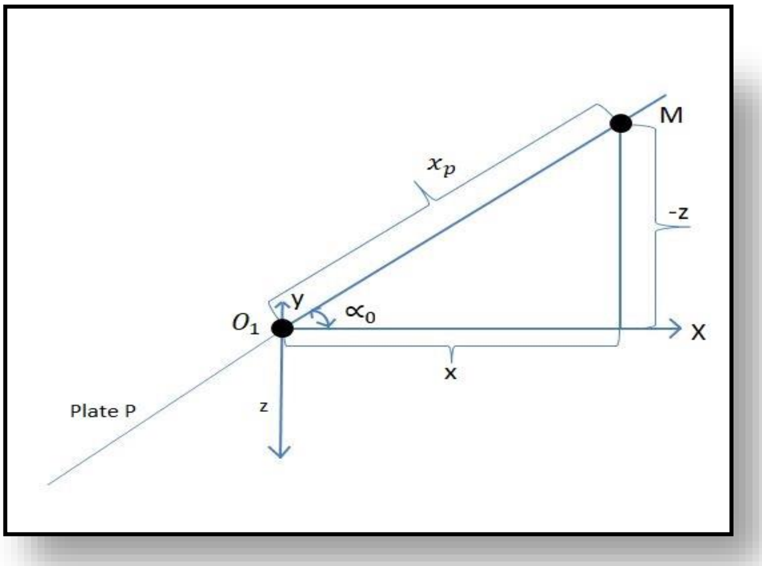


Figure 30 Angle and Target point

Where,

X_p = Sqrt (x.x+t.t) (1)

Y_p = y (2)

For the calculations of angles we use,

sin $\alpha_o = \frac{-z}{xp}$ or Cos $\alpha_o = \frac{x}{xp}$ (3)

or Tan $\alpha_o = \frac{-z}{xp}$ -> then $\alpha_o = Tan^{-1}(\frac{-z}{x})$ (4)

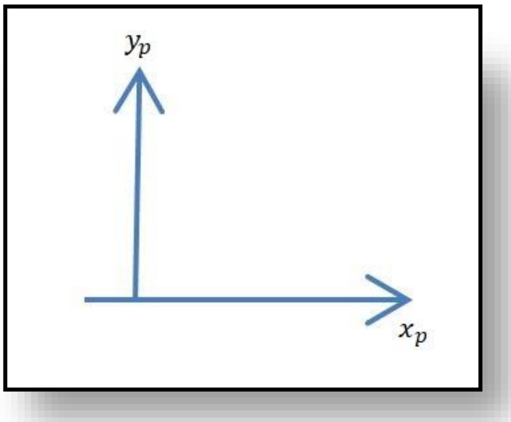


Figure 31 Two Main Axis of the Arm

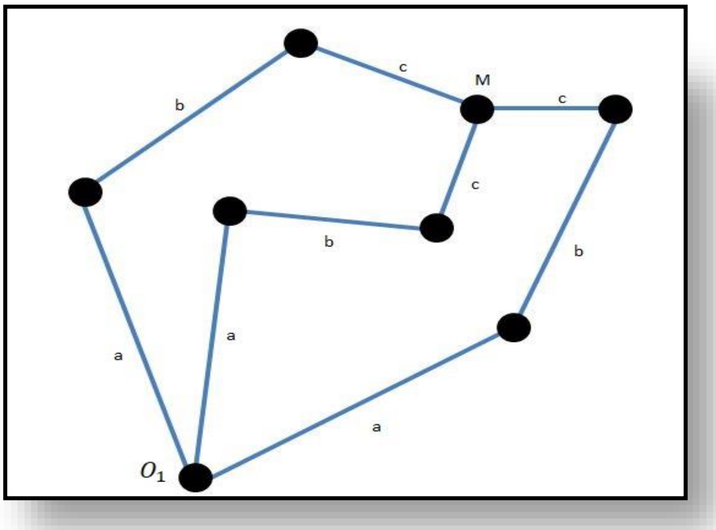


Figure 32 Example of Different Results

The count of solutions for this method is infinity and we have to make some choices.

There is an increasing of circle

At this point you have to write about possibilities of choice accounting to the and energy to reach new desired position, side angels can be for bidden for the robot etc.

6.1.1 Example with one of the above solution

Radius R is arbitrary fixed it has 0,1,2, or 4 solutions if number of solutions ≥ 1 choose one of then

If it has no solution try with another radius R.

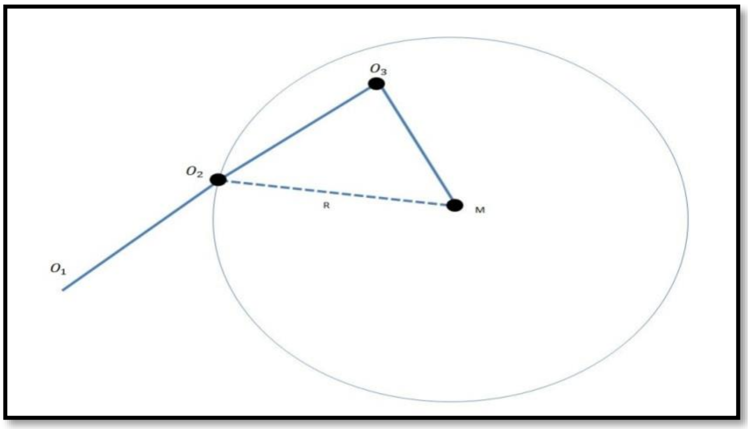


Figure 33 Considering Two circles method

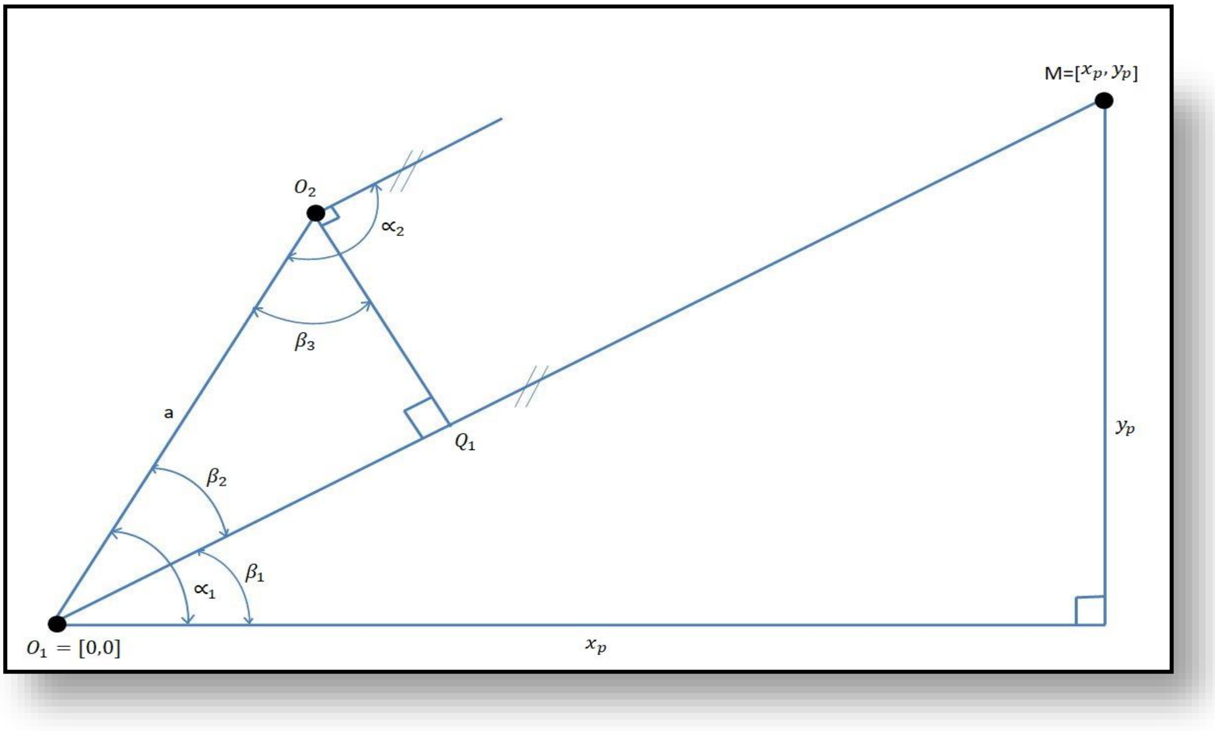


Figure 34 Denotion of Angles to reach the target point

|O1,m| = $\sqrt{x_p{}^2 + y_p{}^2}$ (5)

|Q$_1$Q$_2$| = b

|O$_1$O$_2$| = a

ß1 = tan$\frac{Y_p}{X_p}$ (6)

ß $_2$ = tan$\frac{|O_2Q_1|}{|O_1Q_1|}$, (7)

We have to determine this

α$_1$ = ß$_1$+ß$_2$. (8)

This is generally determined by relations of geometrical equations and we substitute the values to determine the position.

(Xp,Yp) is our target position of the robot arm.

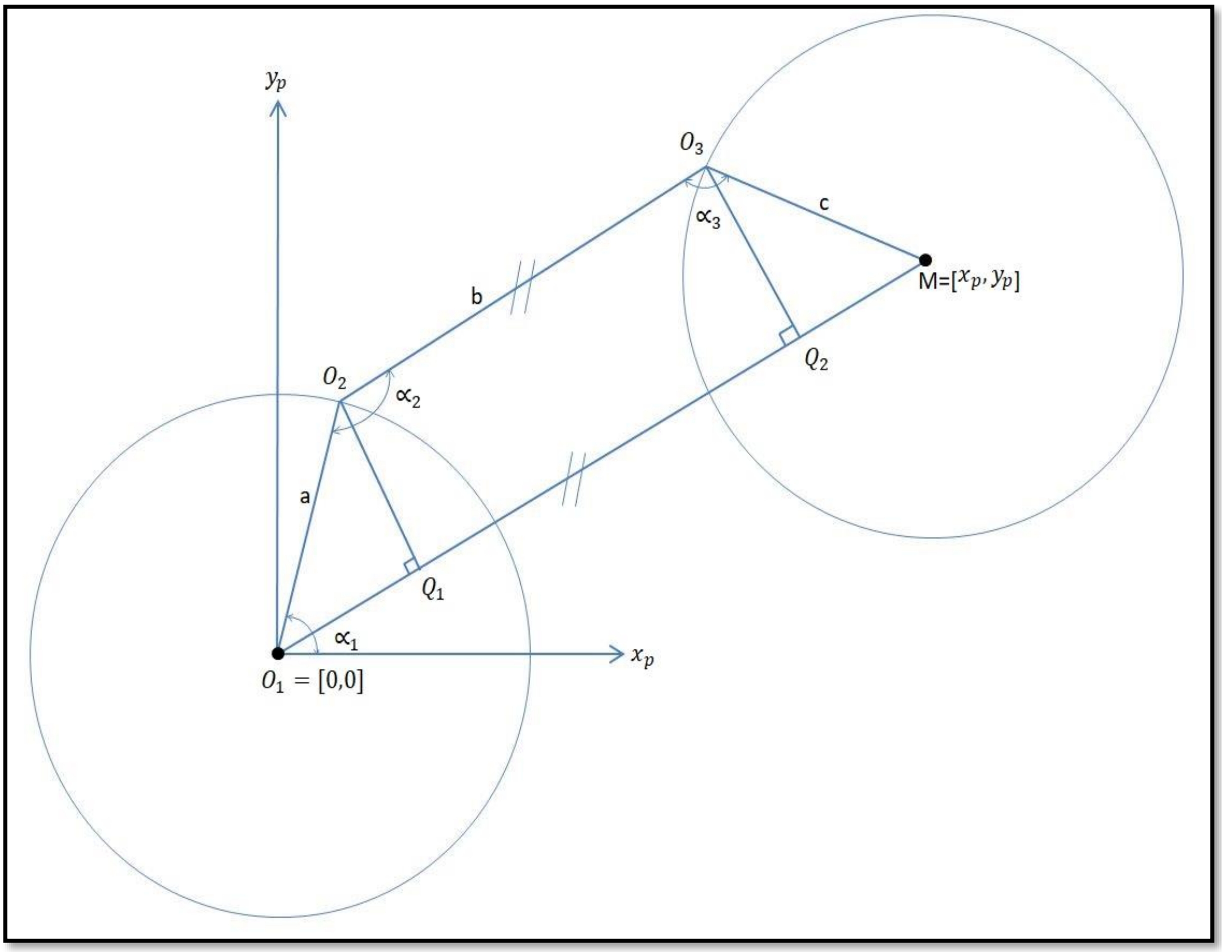


Figure 35 The Two circles and the angles to reach the point

We know : $O_1 = [0,0]$

$M = [x_n, y_n]$

a,b,c

we should find : $O_2 = [x_2,y_2]$=?

$O_3 = [x_3,y_3]$ = ?

Equation of circle 1 : $(X_p\text{-}O)^2 + (Y_p\text{-}0)^2 = a^2$ (9)

Equation of circle 2 : $(X_p\text{-}X_m)^2 + (Y_p\text{-}Y_m)^2 = C^2$ (10)

Point O2 belongs to circle 1 and point O3 belongs to circle 2.

$(x_2)^2 + (y_2)^2 = a^2$ and $(x_3 - x_n)^2 + (y_3 - y_n)^2 = c^2$ (11)

Distance between points O_2 and O_3 is b

b = $\sqrt{|x_3 - x_2|^2 + |y_3 - y_2|^2}$ (12)

B has the same tangent as $|O_1\text{-}m|$=d Equation of line is $y_p=kX_p+c_1$

For b: O_2 and O_3 belong to line b :$y_2 = Kx_2 + q_c$ (13)

$y_3 = Kx_3 + q_c$

For $|O_1m|$: O = K.0+qd =>qd = o (14)

$Y_m = Kx_m + 0$

So solving with this concept we arrived with a set of equations :

Circle 1 => $(x_2\text{-}0)^2 + (yp\text{-}0)^2$ (15)

Circle 2 => $(x_p\text{-}x_m) + (y_p\text{-}y_m)^2$ = c2 (16)

O2 =>$x_2{}^2 + y_2{}^2 = a^2$ (17)

$(x_{3-}x_m)^2 + (y_p - y_m)^2 = c^2$ (18)

Between O_1& O_2

b = $\sqrt{(x_0 - x_2)^2 + (y_0 - y_2)^2}$ (19)

We know that

$Y = kx + qc$ (20)

Point O_2 => $y_2 = kx_2 + q_c$

$y_8 = kx_3 + q_c$ (21)

Qd = O

M => ym = kxm +O (22)

Now we know,

Xp^2 + yp^2 = a^2 (23)

$(x_p - x_m) + (y_p - y_m) = c^2$ (24)

$({x^2}_3 + {y_2}^2)$ = a^2 (25)

$(x_3 - x_m)^2 + (y_3 - y_m)^2 = c^2$ (26)

$where\, y_3 = kx_3 + q_c$ (27)

$y_2 = kx_2 + q_c$ (28)

$b = \sqrt{(x_3 - x_3)^2 + (y_3 - y_3)^2}$ (29)

$ym = kxm$ (30)

$From$ (2)& (3)

${x_p}^2 + {y_p}^2 = {x_p}^2 + {y_p}^2$ (31)

From (2) & (4)

$(x_p - x_m)^2 + (y_p - y_m)^2 = (x_3 - x_m)^2 + (y_3 - y_m)^2$ (32)

$({x_p}^2 + {x_m}^2 + 2x_p\, x_m) + ({y_p}^2 + {y_m}^2 - y_p y_m)$ (33)

$({x_3}^2 + {x_m}^2 - 2x_3 x_m) + ({y_p}^2 + {y_m}^2 - y_p y_m)$ (34)

$(xp^{\,2} + yp^2) + xm^2 + ym^2 - 2\left(x_p x_m + y_p y_m\right)$ (35)

$= ({x_3}^2 + {y_3}^2) + ({x_m}^2 + {y_m}^2) - 2\,(x_2 x_m + y_3 y_m)$

$a^2 - (x_3^2 + y_3^2) = -2\,(x_3 x_m + y_3 y_m) + 2\,(x_p x_m + y_p y_m)$ (36)

= -$2x_3 x_m - 2y_3 y_m + 2x_p x_m + 2y_p y_m$

= -$2x_m\left(x_p . x_3\right) + 2y_m\left(y_p - y_3\right)$

= $2x_m\left(x_p . x_3\right) + 2kx_m\left(y_p - kx_3\right)$

= $2x_m[\left(x_p . x_3\right) + k\left(y_p - kx_3 - q_3\right)$

$\frac{a^2 - \left(x_3^2 + y_3^2\right)}{2\,[\left(x_p - x_3\right) + k\,\left(y_p - kx_3 - q_c\right)} = xm$ (37)

$\frac{a^2 - \left(x_3^2 + y_3^2\right)}{2\,[\left(x_p - x_3\right) + \left(ky_p - k^2 x_3 - q_c\right)} = xm$ (38)

$\frac{{x_p}^2 + \left(y_p^2 - x_3^2 - y_3^2\right)}{2\,[\left(x_p - x_3\right) + \left(ky_p - k^2 x_3 - q_c\right)} = xm$ (39)

$\frac{{x_p}^2 - x_3^2 + y_p^2 - y_3^2}{2\,\left(x_p - x_3\right) + 2\left(ky_p - k^2 x_3 - q_c\right)} = xm$ (40)

$$\frac{(x_p + x_3)(x_p - x_3) + (y_p^2 - y_3^2)}{2 + 2(ky_p - k^2 x_3 - q_c)} \quad = xm \tag{41}$$

$We\ know\ that : \ ym = \ kxm.$

$$= \boldsymbol{k} \ \times \left[\frac{\boldsymbol{(x_p + x_3) + (y_p^2 - y_3^2)}}{\boldsymbol{2+2\ (ky_p - k^2 x_3 - k_{qc}}}\right] \tag{42}$$

In this same method we tried solving the equations in MATLAB and we approached a very big solution which is not supportive for the Arduino software and the result was not accurate the result is shown below.

$Xp^2 + yp^2 = a^2$ (43)

$(x_p - x_m) + (y_p - y_m) = c^2$ (44)

$(x^2{}_3 + y_2{}^2) = a^2$ (45)

$(x_3 - x_m)^2 + (y_3 - y_m)^2 = c^2$ (46)

In this we considered,

A = X_2,

B = Y_2,

C = X_3,

D = Y_3,

X = X_m,

Y = Y_m.

So we have a set of equations now which we have to upload in the MATLAB software and wait for the result it took a very long time and did not produce the accurate result so this method was eliminated.

$A^2 + B^2 – a^2$ (47)

$B - \frac{y}{x} - A - D + \frac{y}{x} - C$ (48)

$C^2 – 2x\text{-}C + x^2 + d^2 – 2y\text{-}D + y^2 – c^2$

$C^2 – 2A\text{-}C+A^2+D^2\text{-}2\text{-}B\text{-}D+B^2\text{-}b^2$ (49)

B from (2) is D + $\left(D + \frac{A-y}{x} - \frac{C-y}{x}\right)$insert B to (1) and (4)

Reduced to 3 equation with unknown variables A,C,D

$A^2 + \left(D + \frac{A-y}{x} - \frac{C-y}{x}\right)^2 - a^2$ (50)

C^2-2x-C + x^2 + D^2 – 2y-D + y^2-c^2

C^2-2A-C+A^2+D^2–2-$\left(D+\frac{A-y}{x}-\frac{C-y}{x}\right)-D+\left(D+\frac{A-y}{x}-\frac{C-y}{x}\right)^2-a^2$

1B) A^2 + $\left(D+\frac{A-y}{x}-\frac{C-y}{x}\right)^2-a^2$

C^2-2x-C + x^2 + D^2 – 2y-D + y^2-c^2

(4B)C^2-2A-C+A^2+D^2–2-$\left(D+\frac{A-y}{x}-\frac{C-y}{x}\right)-D+\left(D+\frac{A-y}{x}-\frac{C-y}{x}\right)^2-b^2$ (51)

A from (1B)

$$\left[\frac{x^2-\left(\frac{\sqrt{2-C-D-x-y-C^2-Y^2-D^2-x^2+a^2-x^2+a^2-y^2}}{x}+\frac{2-C-y^2-2-D-x-y}{2-x^2}\right)}{\frac{x^2-\left(\frac{\sqrt{2-C-D-x-y-C^2-Y^2-D^2-x^2+a^2-y^2}}{x}-\frac{2-C-y^2-2-D-x-y}{2-x^2}\right)}{x^2+y^2}}\right] \quad (52)$$

C form (3)$\begin{bmatrix} x+\sqrt{(D+c-y)-(c-D+y)} \\ x-\sqrt{(D+c-y)-(c-D+y)} \end{bmatrix}$

D form (3) $\begin{bmatrix} y+\sqrt{(C+c-x)-(c-C+x)} \\ y-\sqrt{(C+c-x)-(c-C+x)} \end{bmatrix}$

A form (4B) $$\left[\frac{\frac{x^2.\left(\frac{2.C.x^2+2.C.y^2}{2-x^2}+\frac{b-\sqrt{x^2+y^2}}{x}\right)}{x^2+y^2}}{\frac{x^2.\left(\frac{b-\sqrt{x^2+y^2}}{x}-\frac{2-C-x^2+2\,C-y^2}{2x^2}\right)}{x^2+y^2}}\right] \quad (53)$$

C from (4B) $$\left[\frac{\frac{x^2-\left(\frac{2-A-x^2+2-A-y^2}{2-x^2}+\frac{b-\sqrt{x^2+y^2}}{x}\right)}{x^2+y^2}}{\frac{x^2-\left(\frac{b-\sqrt{x^2+y^2}}{x}-\frac{2-A-x^2+2-A-y^2}{2-x^2}\right)}{x^2+y^2}}\right] \quad (54)$$

Insert D from (3) to (1B) and (4B) – 2 solutions reduce to 2 equation with variable A,C

First

A^2+ $\left[y+\sqrt{(C+c-x)(c-C+x)}+\frac{A-y}{x}-\frac{C-y}{x}\right]^2-a^2$ (55)

$$C^2 - 2AC + A^2 + [y + \sqrt{(C + c - x) - (c - C + x)}]^2$$
$$- \; 2\left[y + \; \sqrt{(C + c - x)(c - C + x)} + \frac{A - y}{x} - \; \frac{C - y}{x}\right][\, y$$
$$+ \; \sqrt{(C + c - x) - (c - C + x)}\Big]$$
$$+ \left[y + \; \sqrt{(C + c - x) - (c - C + x)} + \frac{A - y}{x} - \; \frac{C - y}{x}\right]^2$$

Second

$\mathrm{A}^2 + \left[\, y - \sqrt{(C + c - x) - (c - C + x)} \; + \; \frac{A-y}{x} - \; \frac{C-y}{x}\right]^2 - \; a^2$

$\mathrm{C}^2 \quad – \quad 2\mathrm{AC}+\mathrm{A}^2 \quad + \quad \left[\, y - \sqrt{(C + c - x) - (c - C + x)}\right]^2 - \; 2 \; \Big[y - \sqrt{(C + c - x)(c - C - x)} + \; \frac{A-y}{x} - \; \frac{C-y}{x}\Big] [\, y - \; \sqrt{(C + c - x)(c - C - x)} \; + \Big[y + \sqrt{(C + c - x)(c - C - x)} \; + \; \frac{A-y}{x} - \; \frac{C-y}{x}\Big]$

First solution continues:

A from (5) is (56)

$$\frac{\dfrac{x^2\left(\dfrac{\sqrt{x^4 - 2.C.x^3 + c^2.x^2 - c^2.x^2 + a^2x^2 + a^2.y^2 - c^2x^2 + 2cy^2 - 2x^2.y\,\sqrt{2Cx - C^2 + C^2 - x^2} + 2Cxy\,\sqrt{2Cx - C^2 + C^2 - x^2}}}{x} - \dfrac{2xy^2 - 2cy^2 + 2xy\,\sqrt{2Cx}}{2x^2}\right)}{x^2 + y^2}}{\dfrac{x^2\left(\dfrac{2xy^2 - 2cy^2 + 2xy\sqrt{2Cx - C^2 + c^2 - x^2}}{2x^2} + \dfrac{\sqrt{x^4 - 2Cx^3 + C^2x^2 - c^2y^2 + a^2x^2 - a^2y^2 - C^2x^2 - y^2x^2 + 2Cxy^2 - 2x^2y\,\sqrt{2Cx - C^2 + C^2 - x^2} + 2C.x.y.\sqrt{2Cx}}}{x}\right)}{x^2 + y^2}}$$

Second solutions continue

A from (7) is (57)

$$\frac{\dfrac{x^2\left(\dfrac{\sqrt{x^4 - 2.C.x^3 + c^2.x^2 - c^2.x^2 + a^2x^2 + a^2.y^2 - c^2x^2 + 2cy^2 - 2x^2.y\,\sqrt{2Cx - C^2 + C^2 - x^2} + 2Cxy\,\sqrt{2Cx - C^2 + C^2 - x^2}}}{x} - \dfrac{2xy^2 - 2cy^2 + 2xy\,\sqrt{2Cx - C^2 + C^2 - x^2}}{2x^2}\right)}{x^2 + y^2}}{\dfrac{x^2\left(\dfrac{2xy^2 - 2cy^2 + 2xy\sqrt{2Cx - C^2 + c^2 - x^2}}{2x^2} + \dfrac{\sqrt{x^4 - 2Cx^3 + C^2x^2 - c^2y^2 + a^2x^2 - a^2y^2 - C^2x^2 - y^2x^2 + 2Cxy^2 - 2x^2y\,\sqrt{2Cx - C^2 + C^2 - x^2} + 2C.x.y.\sqrt{2Cx - C^2 + x^2}}}{x}\right)}{x^2 + y^2}}$$

This is the solution that we got which is very big and this cannot be imported in the Arduino program so we would like to find a simple solution for our robot to reach the desired solution. The above all was regarding our first method of solving but we were not satisfied with the results.

6.2 Second Method of Solving

The second method we used was the kinematic solution generally a robots calculation is always based on the kinematics and even we were interested to try it and you can see the results that we obtained. [20]

6.2.1 Solving the Inverse kinematics

The second method we used was the kinematic solution generally a robots calculation is always based on the kinematics and even we were interested to try it and you can see the results that we obtained

In this method I try to solve the equations with inverse kinematics method considering the robots 6 degrees of freedom. Short description of the calculations is shown below. [20]

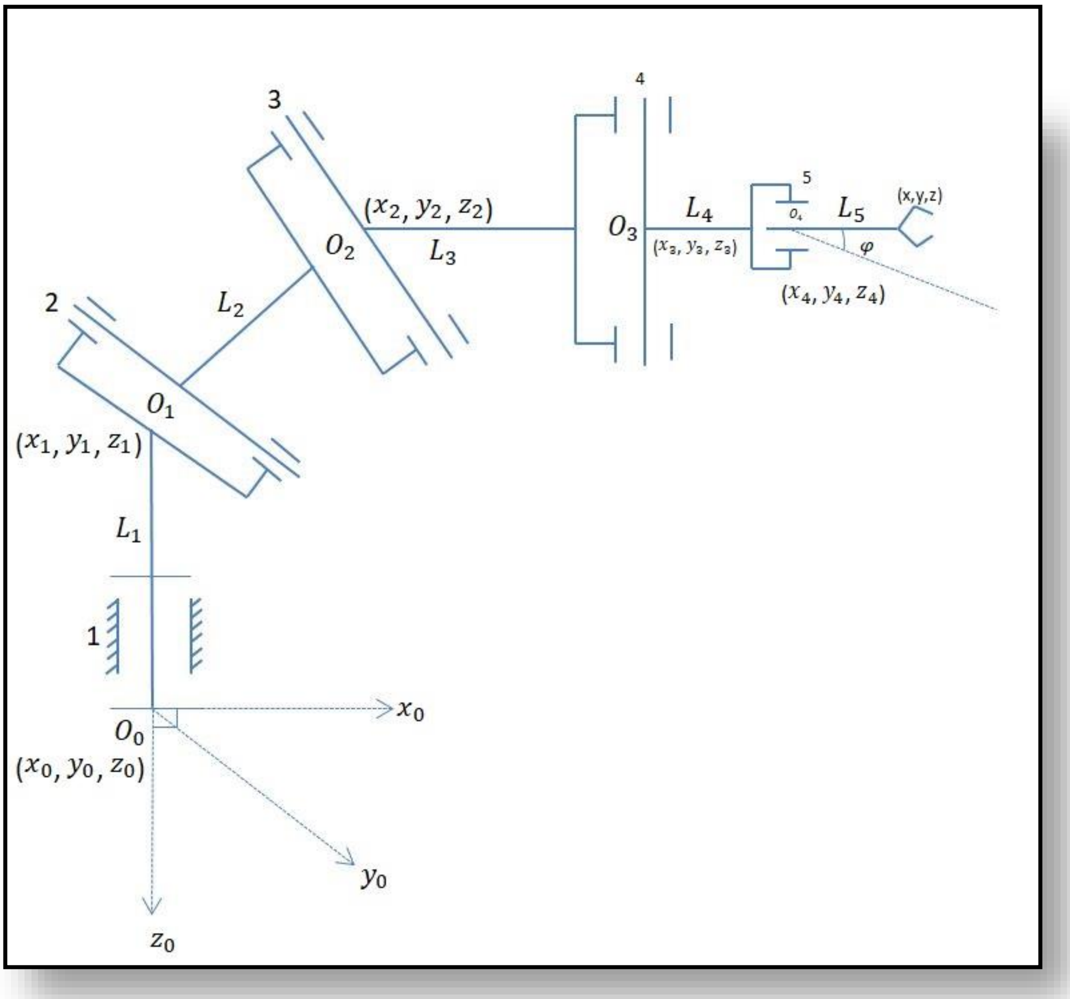


Figure 36 Skeleton View of Robot Arm

The above diagram is the skeleton of the robot arm which is used for inverse kinematic calculation.

$Y = [L_2\cos\theta_2 + L_3\cos(\theta_2+\theta_3) + L_4 \cos [(\theta_2+\theta_3+\theta_4) + L_5] \text{Sin}\theta$ (59)

$Z = L_1 + L_2 \sin\theta_2 + L_3 \sin(\theta_2+\theta_3)\ L_4 \text{Sin}(\theta_2+\theta_3+\theta_4) + L_5$ (60)

$x_4 = x\text{-}L_5 \sin\theta^1$ (61)

$y_4 = y\text{-}L_5\sin\theta^1$ (62)

$z_4 = z\text{-}L_5$ (63)

substituting x,y,z into x4,y4,z4

$x_4 = L_2 \text{Cos}\theta_1 \text{Cos}\theta_1 + L_3 \text{Cos}\theta_1\cos(\theta_2+\theta_2) + L_4\cos\theta_1 + \theta_5 + L_5\cos\theta_1 – L3 \cos\theta_1$ (64)

$y_4 = L_2 \sin\theta_1 \text{Cos}\theta_2 + L_3 \sin\theta_1\cos(\theta_2+\theta_2) + L_4\sin\theta_1 + \text{Cos}\theta_1 +\theta_3+\theta_4\ L_5\cos\theta_1 – L_3 \cos\theta 1$ (65)

$z_4 = L_1 + L_2\text{sinc}\theta_1+ L_2\sin(\theta_1+\theta_2) + L_4 \sin\theta_1 \cos(\theta_1 + \theta_1) + L4 – L5$ ---(3)

$$x_4{}^2 + y_4{}^2 + (Z_4 + L_1)^2 = L_4{}^2cos^2\theta_1 + sin^2\theta_1 sin^2\theta_2 + \theta_3 + L^2{}_2cos^2\theta_1 cos^2 (\theta_2 + \theta_3) + L^2{}_2sin^2\theta_1cos^2\theta_2 + l_3{}^2sin^2\theta_1cos^2\theta_1cos^2(\theta_2+\theta_3) + l_4{}^2sin^2\theta_1cos^2(\theta_2+\theta_3+\theta_4) + l_2{}^2sin^2\theta_2 + l_3{}^2sin^2(\theta_2+\theta_3) + l_4{}^2sin^2(\theta_2+\theta_3+\theta_4)$$

$$x_4{}^2 + y_4{}^2 + (Z_4 - L_1)^2 = L_4{}^2cos^2\theta_1 + cos^2\theta_1sin^2\theta_1 + \theta_3 + L^2{}_3(\theta_2 + \theta_3)\{cos^2\theta_1sin^2\theta_1\} + L^2{}_4cos^2[\theta_2+\theta_3+\theta_4]\{cos^2\theta_1 + sin^2\theta_1\} + L^2{}_2sin^2\theta_2 + L^2{}_3sin^2(\theta_2+\theta_3) + L^2{}_4sin^2(\theta_2+\theta_3) + L^2{}_4sin^2(\theta_2+\theta_3+\theta_4) \qquad (66)$$

$x_4{}^2 + y_4{}^2 = l_4{}^2 + l_3{}^2 + l_4{}^2$

CosACosB – Sin A sin B = Cos (A+B) (67)

$2\angle 2\angle 3 \cos(\theta_2 - \theta_2 - \theta_3) + 2l_3l_4\cos(\theta_2 - \theta_3 - \theta_4) + 2l2l4 \cos(\theta_2 - \theta_3 - \theta_4)$

$2\angle 2\angle 3 \cos(\text{-}\theta_3) + 2l_3l_4\cos(-\theta_4) + 2l2l4 \cos(-\theta_3 - \theta_4)$

$\text{-}2 [l_2l_3 \cos\theta_3 + l_3l_4 \cos\theta_4 + l_2l_4 \cos(\theta_3 + \theta_4)]$

$\text{-}2 [l_2l_3 \cos\theta_3 + l_3l_4 \cos\theta_4 + l_2l_4 \cos\theta_3\ cos\ \theta_4 - l_2l_4 \sin\theta_3 - \sin\theta_3 \sin\theta_4]$

$\text{-}2 [l_2l_3 \cos\theta_3 + l_3l_4 \cos\theta_4 + l_2l_4 \cos\theta_3\ cos\ \theta_4 - l_2l_4 \sin\theta_3 \sin\theta_4]$

$x_4{}^2 + y_4{}^2 (z_4 - l_2)^2 = l_2{}^2 + l_3{}^2 + l_1{}^2 - 2 [l_2l_3 \cos\theta_2 + l_3l_4cos\theta_4 + l_2l_4 \cos(\theta_3 + \theta_4]$

$$x^2 + {y_3}^2 + \left(z_3 - l_1\right)^2 = {l_2}^2 cos^2\theta_1 cos^2\theta_2 + {l_3}^2 cos^2\theta_1 cos^2\left(\theta_2 + \theta_3\right) + {l_4}^2 cos^2\theta_1 cos^2(\theta_2$$

$$+ \theta_3 + \theta_4) + {l_4}^2 cos^2 + 2l_2 cos^2\theta_1 cos^2\theta_2 l_3\, cos^2(\theta_2 + \theta_3)$$

$$+ 2l_3 l_4\, cos\theta_1\, cos\theta_1 \cos(\,\theta_2$$

$$+ 2l_3 cos^2\theta_1 \cos(\theta_2 + \theta_3) l_4\, cos\theta_1 \cos(\theta_2 + \theta_3 + \theta_4)$$

$$- 2l_2\, cos\theta_1\, cos\theta_2 l_4\, cos\theta_4\, cos\theta_2$$

$$- 2l_3 cos\theta_1 \cos(\theta_2 + \theta_3) + 2l_4\, cos\theta_1 \cos(\theta_2 + \theta_3 + \theta_4) l_2 \cos\theta_4$$

$$+ {l_2}^2 sin^2\theta_2 cos^2\theta_2 + {l_3}^2 sin^2\theta_1 cos^2(\theta_2 + \theta_4) + {l_2}^4 sin^2\theta_1 cos^2(\theta_2 + \theta_3 + \theta_4)$$

$$+ {l_4}^2 cos^2\theta_4$$

$$+ 2l_2\, sin\theta_1\, cos\theta_2 l_3 \sin\theta_1 \cos(\theta_2 + \theta_3)$$

$$+ 2l_2\, sin\theta_1 cos\theta_4 l_4\, sin\theta_1 \cos(\theta_2 + \theta_3)$$

$$+ 2l_3\, sin\theta_1 \cos(\theta_2 + \theta_3) l_4\, sin\theta_1 \cos(\theta_2 + \theta_3) - 2l_3\, sin\theta_1 \cos(\theta_2$$

$$+ \theta_3) l_4\, cos\theta_4\, sin\theta_1 - 2l_2 sin\theta_1 \cos(\theta_2 + \theta_3 + \theta_4)\, l_2\, cos\theta_4 + l^2{}_2 sin^2\theta_2$$

$$+ {l_2}^2 sin^2(\theta_2 + \theta_4) + {l_2}^2 sin^2\theta_4$$

$$+ {l_2}^2 sin\theta_2 l_3 \sin(\theta_2 + \theta_3) + 2l_2 sin\theta_2 l_4 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$+ 2l_2 sin\theta_2 l_3 sin(\theta_2 + \theta_3) + 2l_2 sin\theta_2 l_4 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$+2l_3 \sin(\theta_2 + \theta_3)\, l_4 \sin(\theta_2 + \theta_3 + \theta_4) - 2l_2 sin\theta_2 l_2 sin\theta_4$$

$$-2l_2 \sin(\theta_2 + \theta_3) l_4 sin\theta_4 - 2l_2 \sin(\theta_2 + \theta_4 + \theta_4) l_2 sin\theta_4$$

$$x^2{}_3 + y^2{}_3 + (z_3 - l_2)^2$$

$$= l^2{}_2 cos^2\theta_a(1) + l^2{}_3 cos^2(\theta_2 + \theta_3)(1) + l^2{}_4 cos^2(\theta_2 + \theta_3 + \theta_4)(1)$$

$$+ l^2{}_4 cos^2\theta_3(1)$$

$$+2l_2 l_3\, cos\theta_2\, \cos(\theta_2 + \theta_3)(1) + 2l_2 l_4\, cos\theta_2\, \cos(\theta_2 + \theta_3 + \theta_4)(1)$$

$$+ 2l_3 l_4 \cos\theta_2\, (\theta_2 + \theta_3) \cos(\theta_2 + \theta_3 + \theta_4)\, (1) - 2\, l_2 l_4 \cos\theta_2\, \cos\theta_4\, (1)$$

$$- 2l_3 l_4 \cos(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3) + \cos(\theta_2 + \theta_3 + \theta_4)(1) - 2l_3 l_4 \cos\theta_3 \cos\theta_4\, (1)$$

$$+{l_2}^2 sin^2(\theta_2 + \theta_3) \cos\theta_4(1) - 2{l_2}^2 sin^2(\theta_2 + \theta_3 + \theta_4) + {l_2}^4 sin^2\theta_4$$

$$+2l_2 sin^2\theta_2 l_3 \sin(\theta_2 + \theta_3) + 2l_2 sin\theta_2(\theta_2 + \theta_3 + \theta_4) + {l_2}^2 sin^2\theta^4$$

$$+2l_3 sin\, (\theta_2 + \theta_3)\, l_4 \sin(\theta_2 + \theta_3 + \theta_4) - 2\, l_3 sin\theta_2 l_4 \sin\theta_4$$

$$-2l_3 \sin(\theta_2 + \theta_3) l_3 \sin(\theta_2 + \theta_3 + \theta_4) - 2l_3 sin\theta_2 + L_4\, sin\theta_4$$

$$x^2{}_3 + y^2{}_3 + (z_3 - l_2)^2 = l^2{}_2(1) + l^2{}_3\, (1) + l^2{}_4\, (1) + l^2{}_4(1)$$

$$+2l_2 l_3 \cos\left(\theta_2 - \theta_2 - \theta_3\right) + 2l_2 l_4 \cos(\theta_2 - \theta_2 - \theta_3 - \theta_4)$$

$$- 2l_3 l_4 \cos(\theta_2 + \theta_3 - \theta_2 - \theta_3 - \theta_4) - 2{l_4}^2 \cos(\theta_2 + \theta_3 + \theta_4 - \theta_4)$$

$$- 2l_3 l_4 \cos(\theta_2 + \theta_3 - \theta_2 - \theta_3 - \theta_4) - 2{l_4}^2 \cos(\theta_3 - \theta_4)$$

$$x^2{}_3 + y^2{}_3 + (z_3 - l_1)^2 = l^2{}_2 + l^2{}_3 + 2l^2{}_4 - 2l_2 l_3 \cos(\theta_3 + \theta_4)$$

- $2l_3l_4\, cos\, \theta_4 - 2l_2l_4\, cos\, (\theta_2 - \theta_4) - 2l_3l_4\cos(\theta_2 + \theta_3 - \theta_4) - 2{l_4}^4\cos(\theta_2 + \theta_2)$

.

.

This continues, the kinematics solution was very big and the result which I approached did not seem accurate so there was a need to change the method of solving.

6.3 Third Method of Solving

Basics Vector Method. So at last we came out with a result for the robot to reach the desired target point by calculating the angles for the servos

This method is the final method of my work where we used some vector methods and trigonometric equations to find the final relation for the angles of the servo and the target position. The corresponding equations of x and y should be written from the figure which is the first step in this method. This equations are squared and simplified using some trigonometric functions and arrived with a final equation with two degrees of freedom.

Where,

a, b, c - length of each arms of robot

φ - Angle corresponding to first arm with length **a.**

ψ - Angle corresponding to first arm with length **b.**

γ - Angle corresponding to first arm with length **c.**

(x,y) - It is the target position of robot on the other words we can say it the input that is given by us to the robot.

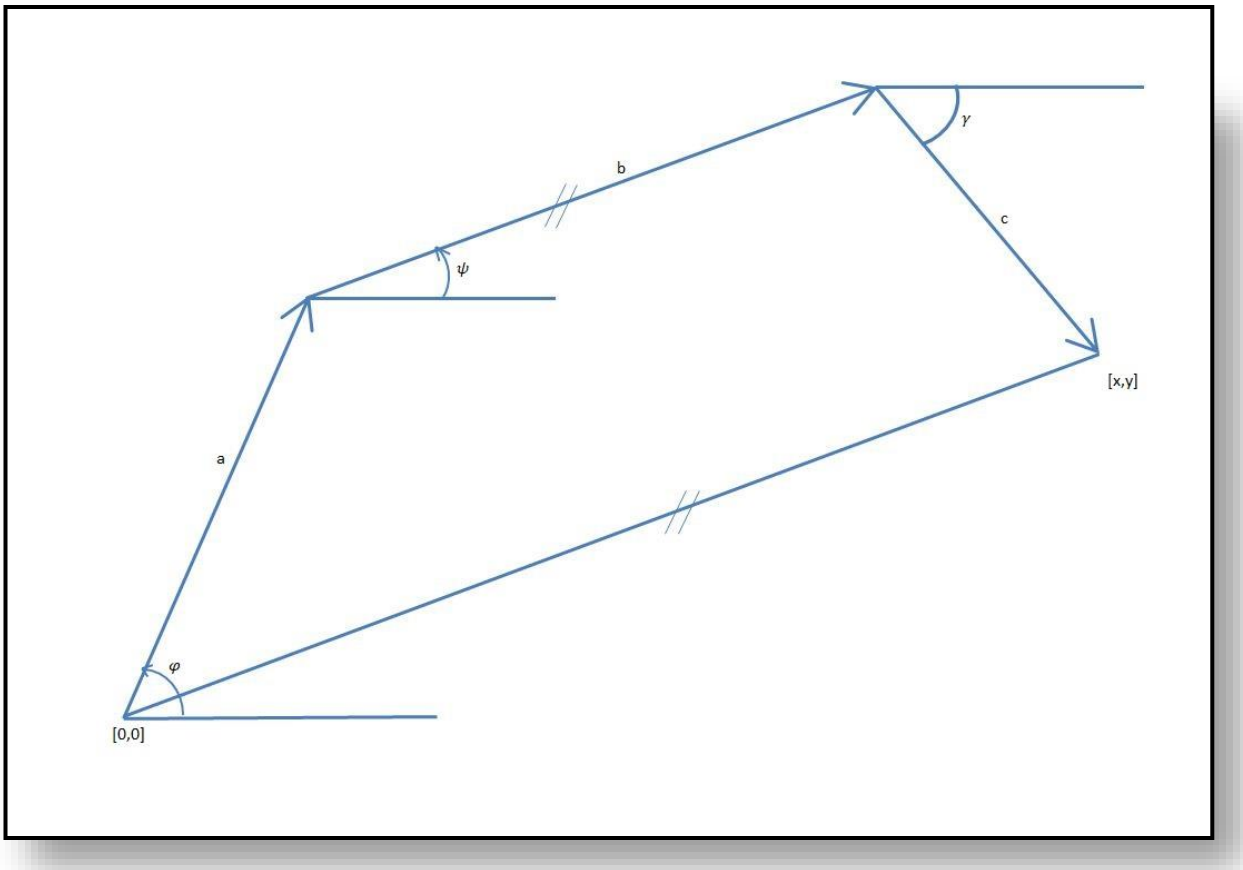


Figure 37 The three arms and target point

$$a \cos \varphi + b \cos \psi + c \cos \gamma = x \quad (67)$$

$$a \sin \varphi + b \sin \psi - c \sin \gamma = y \quad (68)$$

squaring both equation

$$a^2\cos^2\varphi = [(x - b\ cos\varphi) - c \cos\gamma]^2 \quad (69)$$

$$a^2\sin^2\varphi = [(y - b\ sin\ \varphi) + c \sin\gamma]^2 \quad (70)$$

$$a^2\cos^2\varphi = x^2 - 2bx\cos\psi + b^2\cos^2\psi - 2c \cos\gamma\ (x - b\cos\psi) + c^2\cos^2\gamma \quad (71)$$

$$a^2\sin^2\varphi = y^2 - 2by\sin\psi + b^2\sin^2\psi - 2c \sin\gamma\ (y - b\sin\psi) + c^2\sin^2\gamma \quad (72)$$

adding (3) + (4)

$$a^2 = x^2+y^2 - 2b\ x \cos\psi - 2by \sin\psi + b^2 - 2c \cos\gamma\ (x - b\cos\psi) - 2c\sin\gamma(y - b\sin\psi) + c^2$$

$$x^2 + y^2 + c^2 + b^2 - a^2 = 2b\ [x \cos\psi + y \sin\psi] + 2c[\cos\gamma\ (x - b\cos\psi) + \sin\gamma\ (y - b\sin\psi)] \quad (73)$$

$$\tan \psi = {}^{y}/_{x}$$

$$\psi = \tan^{-1}({}^{y}/_{x}) \quad (74)$$

$$\gamma_{12} = \text{Arc sin}\,\frac{\boldsymbol{Q}}{\pm\sqrt{\boldsymbol{M}^2 + \boldsymbol{N}^2}} - \boldsymbol{arc}\,\mathbf{tan}(\frac{\boldsymbol{M}}{\boldsymbol{N}}) \qquad (75)$$

After finding the **ψ, γ12** we substitute the values of these in the 1st equation and we can find φ.

$$\text{Where } \varphi = \cos^{-1}\left[\frac{x - bcos\psi - \text{c}\cos\gamma}{a}\right] \qquad (76)$$

6.3.1 Solving in Vector Method

Here is an example for the above equations where a set of values are assumed and it is first drawn geometrically and the angles are measured which is then calculated manually using these formulas and these are values are compared in order to check whether this equation is correct or not.

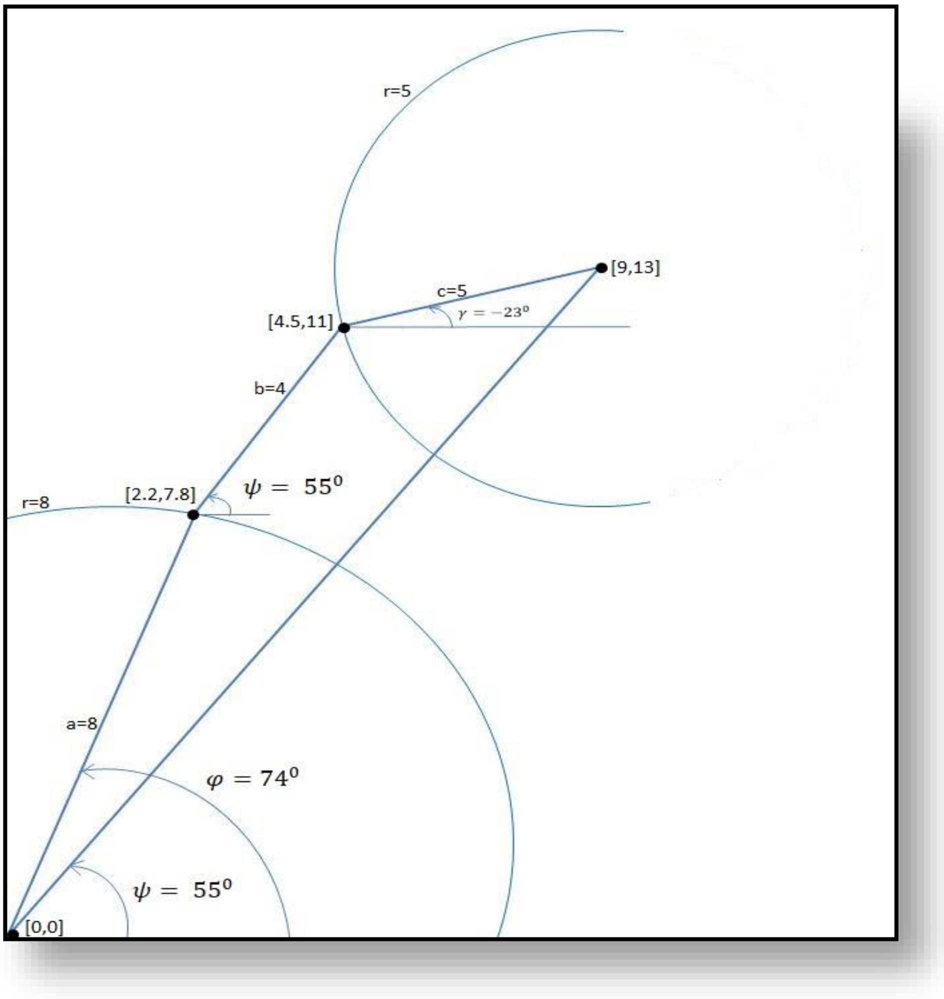


Figure 38 Geometrical calculation of the Angles

This figure is drawn as per scale and it is used for the checking the values that we got for the angles of the servos using the formulas.

As per the drawing we got,

$\psi = 55^0$

$\varphi = 74^0$

$\gamma_{12} = -23^0$

Then we have to continue with the manual result and check at least we get approximate values of these results.

Consider,

x = 9	a = 8
y = 13	b = 4
	c = 5

$\tan\psi = {}^{y}/_{x}$

$\psi = \tan^{-1} ({}^{13}/_{9})$

$\psi = 55^0$

Substituting the values of a, b, c, x, y and ψ in equation (5)

We get

$10.04 = 6.704 \cos\boldsymbol{\gamma} + 9.724 \sin\gamma$

$$6.704\, cos\gamma + \; 9.724 \sin\gamma \; - \; 10.04 = 0$$

The above equation is in the form

$M \cos\gamma + n \sin\gamma + Q = 0$

This relation can be written as,

$$\gamma_{12} = \text{Arc sin} \frac{\boldsymbol{Q}}{\pm\sqrt{\boldsymbol{M^2} + \boldsymbol{N^2}}} - \boldsymbol{arc}\,\mathbf{tan}\,\frac{\boldsymbol{M}}{\boldsymbol{N}}$$

In the above equation we substitute the values and we can the value for gamma

$$\gamma_{12} = \text{Arc sin}\left(\frac{-10.04}{\sqrt{(6.704)^2 + (9.724)^2}}\right) - arc\tan\left(\frac{6.704}{9.724}\right)$$

$\gamma_{12} = -23.61^0$

Next we have to find angle φ so we have now the other 2 angles so we can substitute it in the 1st equation and we can find the angle φ

$a^2\cos^2\varphi \; = (x\text{-}b \cos\varphi)^2 - 2\,(x\text{-}b\cos\psi)\,(c \cos\gamma) + c^2\cos^2\gamma$

Substituting all the values

$64 \cos^2\varphi = 8.55$

$\mathrm{Cos}^2\varphi = 0.13$

$\mathrm{Cos}\ \varphi = \sqrt{0.13} = 0.316$

$\varphi = \mathrm{Cos}^{-1}\ (0.316)$

$\varphi = 72^0$

6.3.2 Parametric equations for line

To show the parametric equation of the line we consider a small example of a vector function. This is done because first the robot reaches the point Xm and Ym and the it as to reach a other point Xm1 and Ym1 which lies in the same line.

$$\vec{r}(t) = \langle t, 1 \rangle$$

A Vector function takes all the function but in some case they also return a function. It can also be a function of 2 or more variables. But the graph in this case will not be the same line.

Now, we want to determine the graph of the vector function above. In order to find the graph of our function we'll think of the vector that the vector function returns as a position vector for points on the graph.

$\vec{v} = \langle a, b \rangle$, is a vector that starts at the origin and

So, to get a graph of vector function we should plug in values of variables and the plot the point that corresponds to each position and we should connect the dots. Here are some evaluations for our example.

$$\vec{r}(-3) = \langle -3, 1 \rangle \qquad \vec{r}(-1) = \langle -1, 1 \rangle$$

So each these are the positions,

$$(-3,1) \qquad (-1,1) \qquad (2,1)$$

are all points that lie on the graph of our vector function.

If we do some more evaluations and plot all the points we get the following sketch.

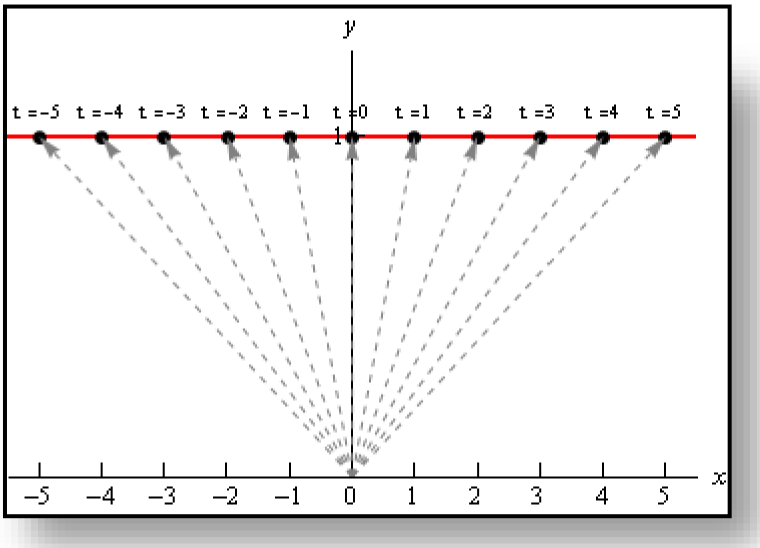


Figure 39 Parametric Equation of Line

7. WORKING OF OUR ROBOT AND ITS PROGRAMMING.

The initial position of robot is where all the angles of the servos are 90 degrees so the robot arm is completely straight end effector is pointing the sky.

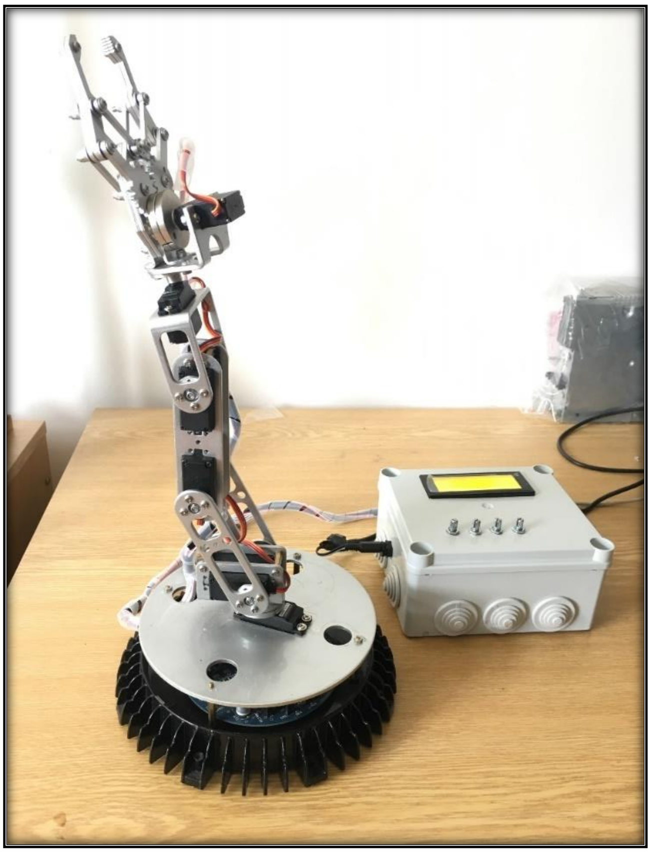


Figure 40 Initial Position of Robot

The control box consists of the power modulator, Arduino, Potentiometer and the LCD display which is in the outer side of the box. The connections of each component is shown in the figure below.

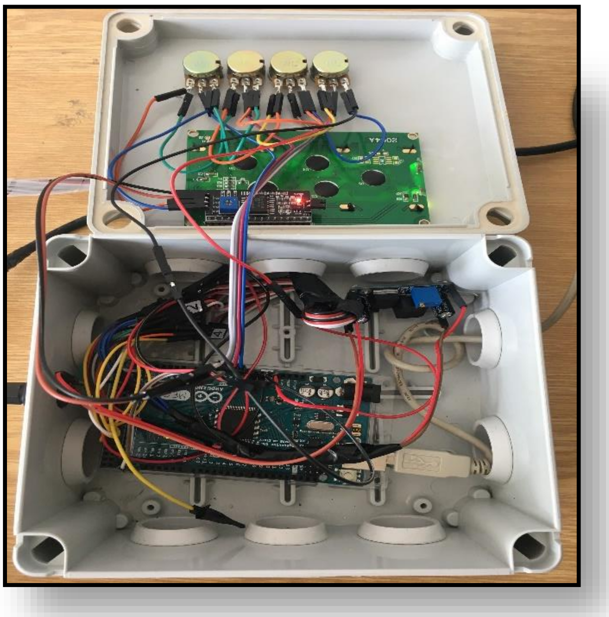


Figure 41 Connections in the Control Box

The first step in programming is to maintain the values of all the servo angle to 90 degrees. As shown in the figure below the rotation, Arm1, Arm2, Arm3 all the angles are set to 90 degrees and the other values shown in the figure such as A0, A1, A2, A3 are the readings of the potentiometers.

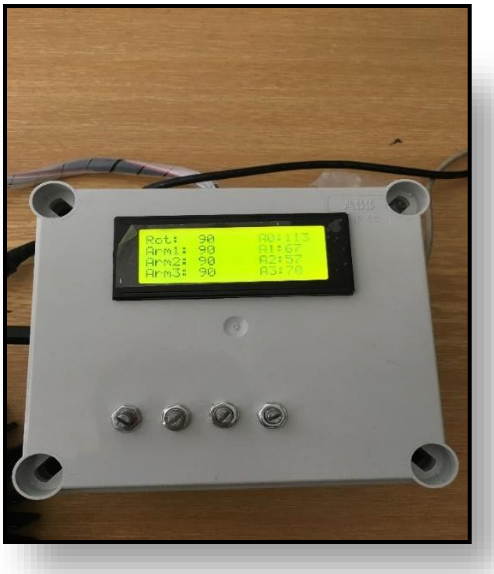
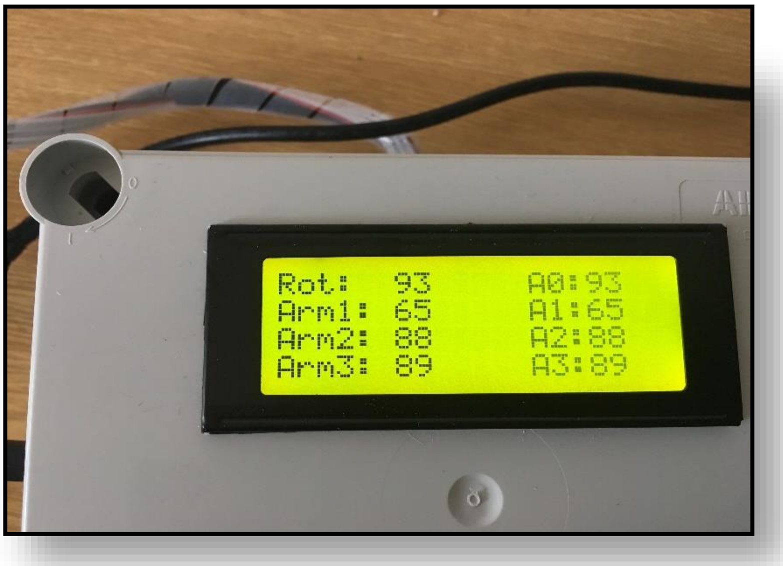


Figure 42 Control Box of Robot Arm.

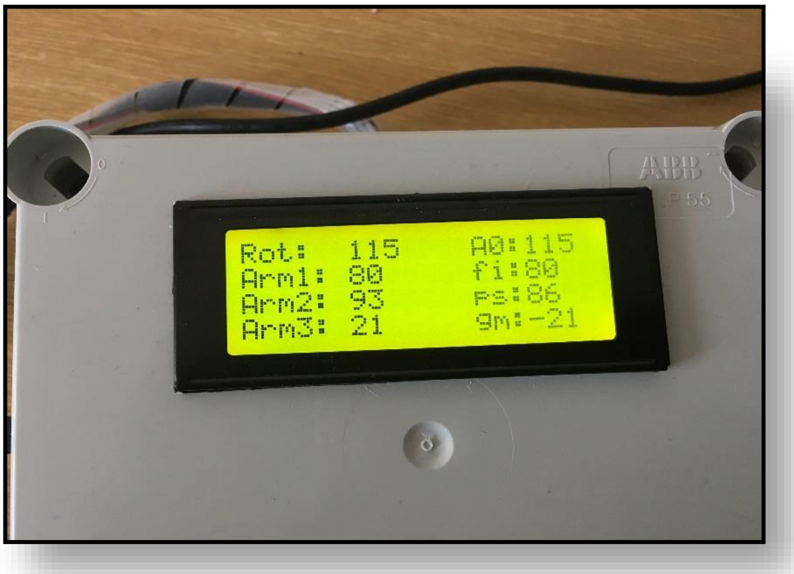
The next step is to coordinate the readings of the potentiometers to the values of the servo motors. Now both are directly proportional so change in the value of potentiometer is same as the value of the servo.



```
Rot:   93      A0:93
Arm1:  65      A1:65
Arm2:  88      A2:88
Arm3:  89      A3:89
```

Figure 43 Coordinating values of servo and potentiometer

Now the values of the potentiometer are named as the angles of each arm namely they are Fi, Psi, Gamma. So program is edited as required for calculating the values of each angle corresponding to each servo position.



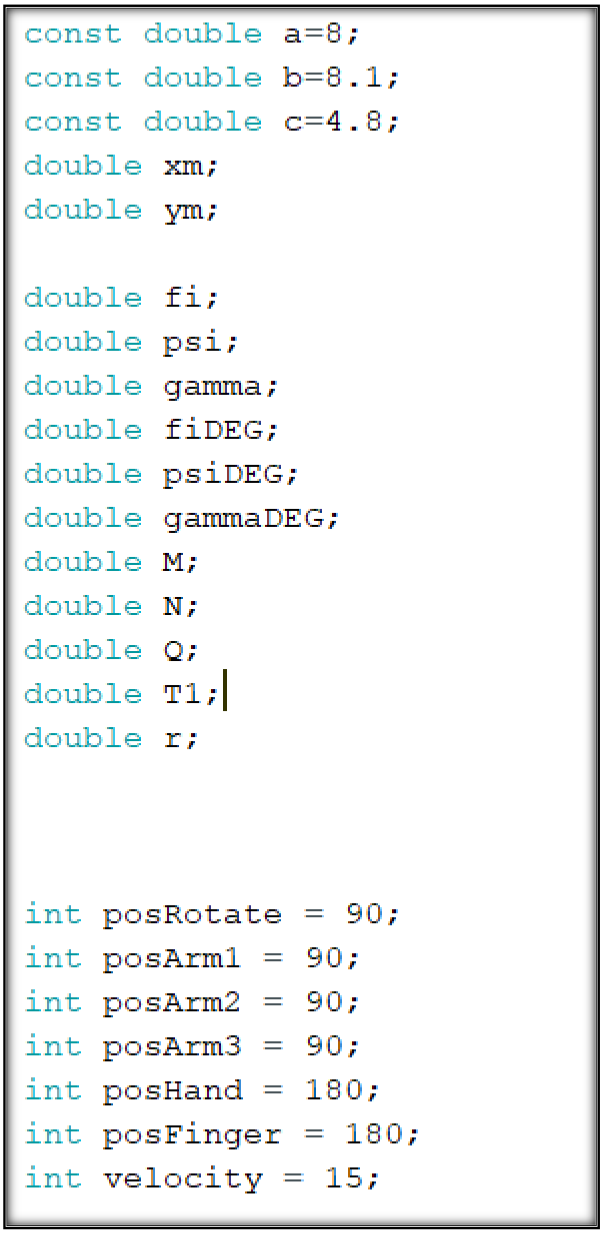
```
Rot:   115     A0:115
Arm1:  80      fi:80
Arm2:  93      ps:86
Arm3:  21      gm:-21
```

Figure 44 Values of angle and Servo position

7.1 Programming the Robot

There were 5 programs each program is for each example for instance the potentiometer and the servo are coordinated which is made in the program all the programs are attached in this thesis.

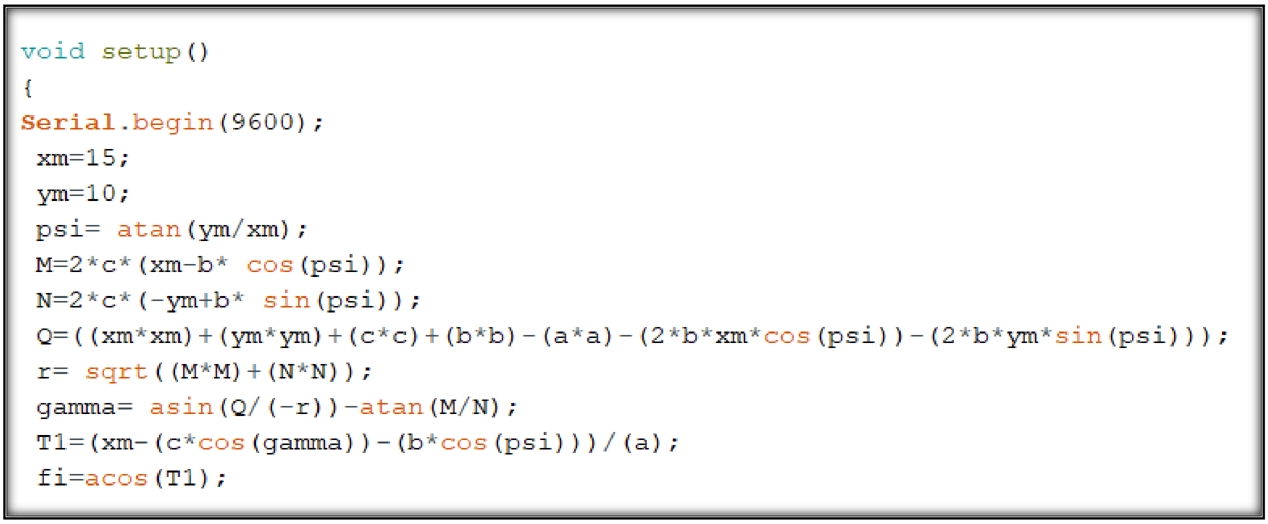
The program for making the robot work by giving the coordinates as the inputs is considered as the main task where the formulas (i.e.) the relation between the Servo angles and the coordinates in written in the Arduino code as Arduino variables.



```
const double a=8;
const double b=8.1;
const double c=4.8;
double xm;
double ym;

double fi;
double psi;
double gamma;
double fiDEG;
double psiDEG;
double gammaDEG;
double M;
double N;
double Q;
double T1;
double r;



int posRotate = 90;
int posArm1 = 90;
int posArm2 = 90;
int posArm3 = 90;
int posHand = 180;
int posFinger = 180;
int velocity = 15;
```

Figure 45 Input varibles

The above figure shows the input variable such as the length of the each arms which are respectively a, b, c which are in centimetres. The xm, ym are the target point which is given as the input to the program so the Arduino calculates and will reach the target point. Then comes the angle of each arm of the robot which is namely Psi, Fi, Gamma. Then it is converted from radians to degrees.

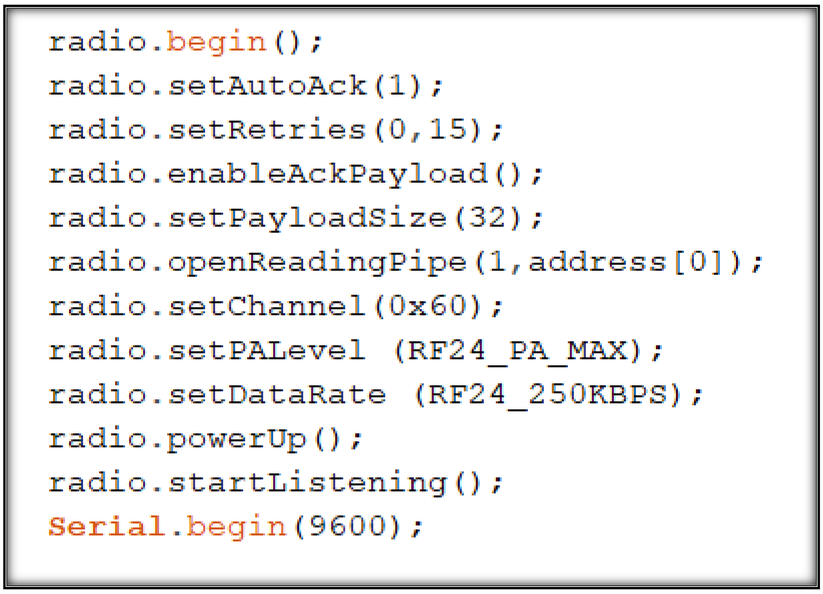


```
void setup()
{
Serial.begin(9600);
 xm=15;
 ym=10;
 psi= atan(ym/xm);
 M=2*c*(xm-b* cos(psi));
 N=2*c*(-ym+b* sin(psi));
 Q=((xm*xm)+(ym*ym)+(c*c)+(b*b)-(a*a)-(2*b*xm*cos(psi))-(2*b*ym*sin(psi)));
 r= sqrt((M*M)+(N*N));
 gamma= asin(Q/(-r))-atan(M/N);
 T1=(xm-(c*cos(gamma))-(b*cos(psi)))/(a);
 fi=acos(T1);
```

Figure 46 Formula Part of the Program

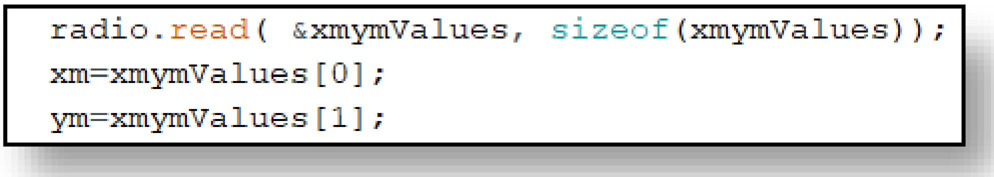
The figure shows the important part of robot arms programming where the input is the xm and ym and the rest are the formulas for the angles. M, N and Q are the constant which is used in the formula for calculating angle gamma.

The WIFI module is added which is also included in the program. Where the xm and ym values are given as input from the module for the robot.



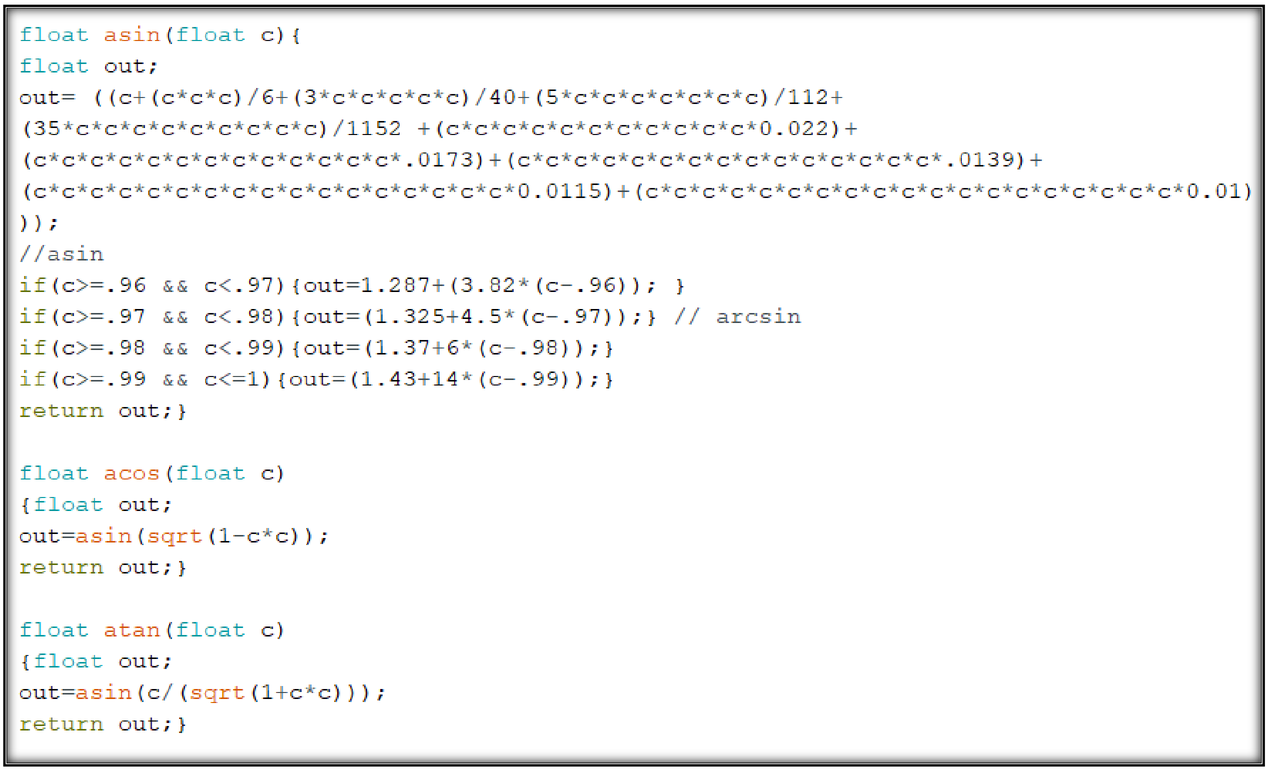
```
radio.begin();
radio.setAutoAck(1);
radio.setRetries(0,15);
radio.enableAckPayload();
radio.setPayloadSize(32);
radio.openReadingPipe(1,address[0]);
radio.setChannel(0x60);
radio.setPALevel (RF24_PA_MAX);
radio.setDataRate (RF24_250KBPS);
radio.powerUp();
radio.startListening();
Serial.begin(9600);
```

Figure 47 WIFI module program



```
radio.read( &xmymValues, sizeof(xmymValues));
xm=xmymValues[0];
ym=xmymValues[1];
```

Figure 48 Array for collecting xM and yM values

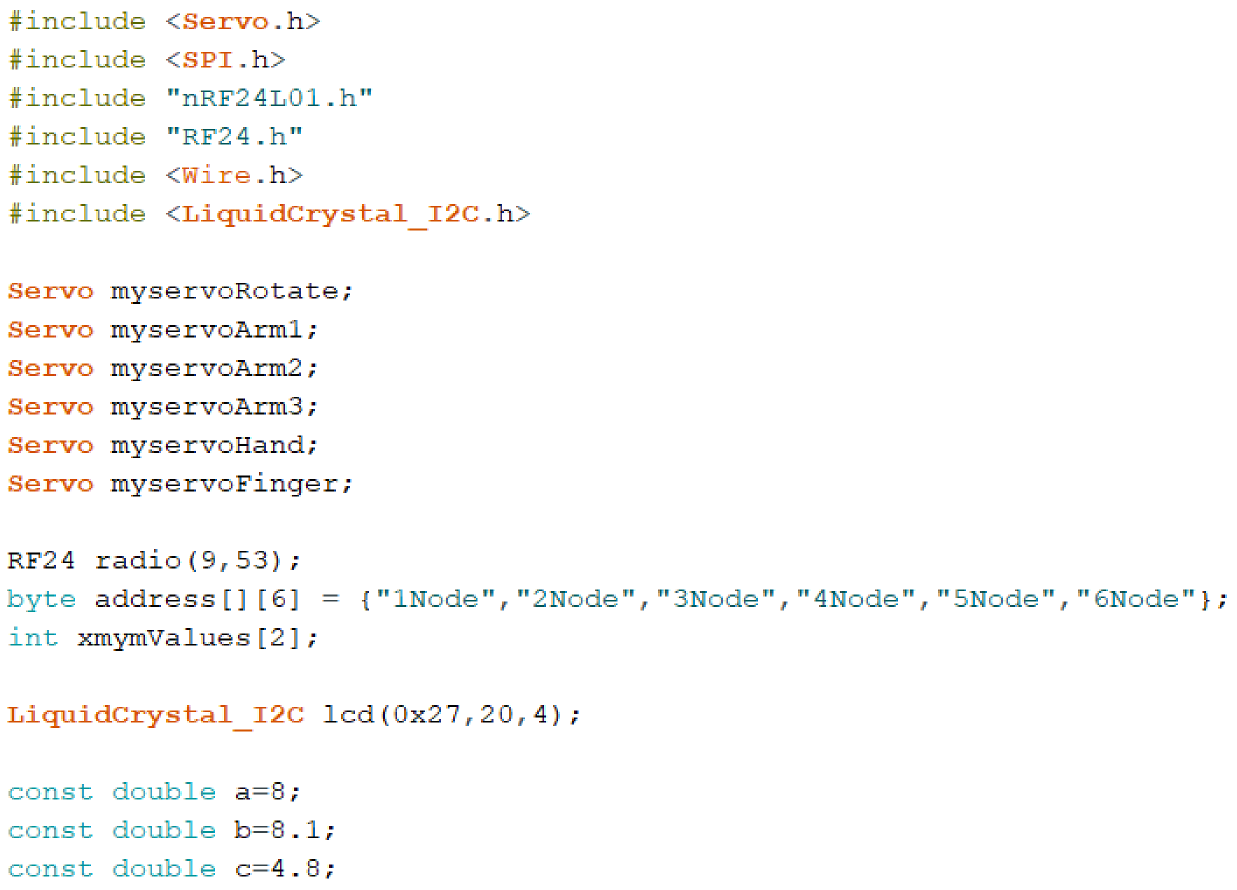


```
float asin(float c){
float out;
out= ((c+(c*c*c)/6+(3*c*c*c*c*c)/40+(5*c*c*c*c*c*c*c)/112+
(35*c*c*c*c*c*c*c*c*c)/1152 +(c*c*c*c*c*c*c*c*c*c*c*0.022)+
(c*c*c*c*c*c*c*c*c*c*c*c*c*.0173)+(c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*.0139)+
(c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*0.0115)+(c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*0.01)
));
//asin
if(c>=.96 && c<.97){out=1.287+(3.82*(c-.96)); }
if(c>=.97 && c<.98){out=(1.325+4.5*(c-.97));} // arcsin
if(c>=.98 && c<.99){out=(1.37+6*(c-.98));}
if(c>=.99 && c<=1){out=(1.43+14*(c-.99));}
return out;}

float acos(float c)
{float out;
out=asin(sqrt(1-c*c));
return out;}

float atan(float c)
{float out;
out=asin(c/(sqrt(1+c*c)));
return out;}
```

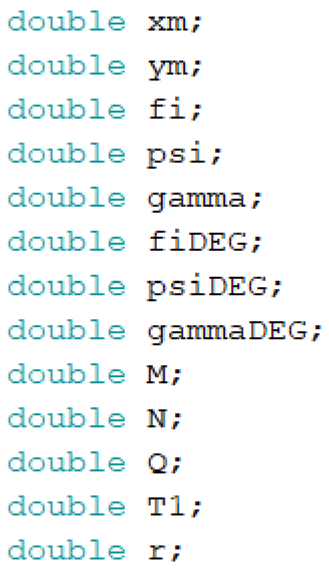
Figure 49 Taylors Series for Arc Tan and Arc sin.

The arc sin and arc tan are calculated by using Taylor's series which is shown in the figure where the format for calculating the series is as the same which is done in mathematical method. The final program is shown below. In the program first the calculated values are made as trial run the calculation is show in the above topic considering some dummy values for the length of the arms and with a imagined coordinates the program is checked for its accuracy. The values are checked and then it was executed for the real values of the robot arms with the length and some imaginary coordinate target points.

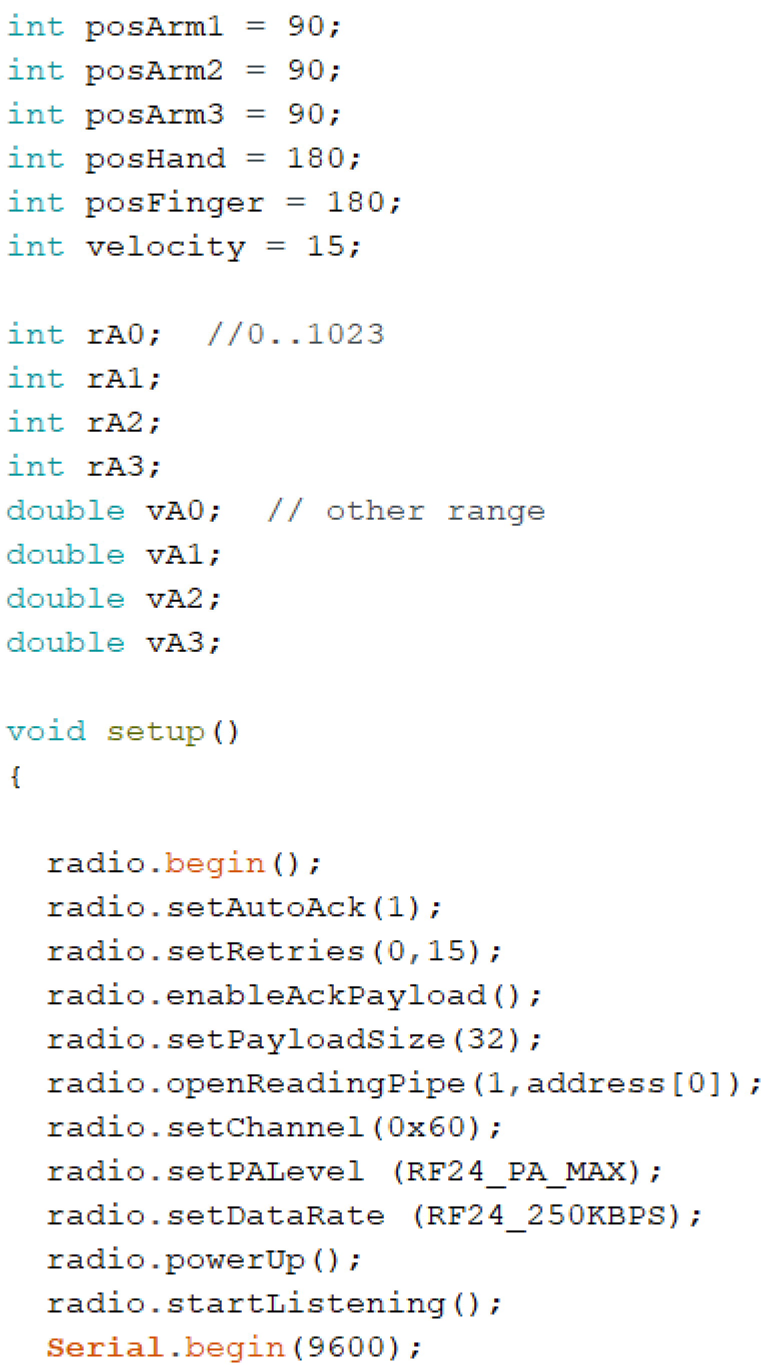
The final program for the robot arm is as follows:



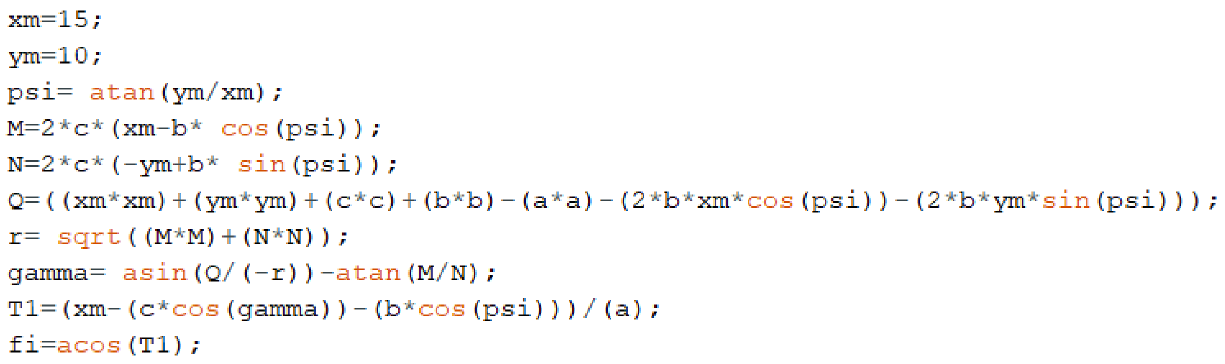
```
#include <Servo.h>
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

Servo myservoRotate;
Servo myservoArm1;
Servo myservoArm2;
Servo myservoArm3;
Servo myservoHand;
Servo myservoFinger;

RF24 radio(9,53);
byte address[][6] = {"1Node","2Node","3Node","4Node","5Node","6Node"};
int xmymValues[2];

LiquidCrystal_I2C lcd(0x27,20,4);

const double a=8;
const double b=8.1;
const double c=4.8;
```



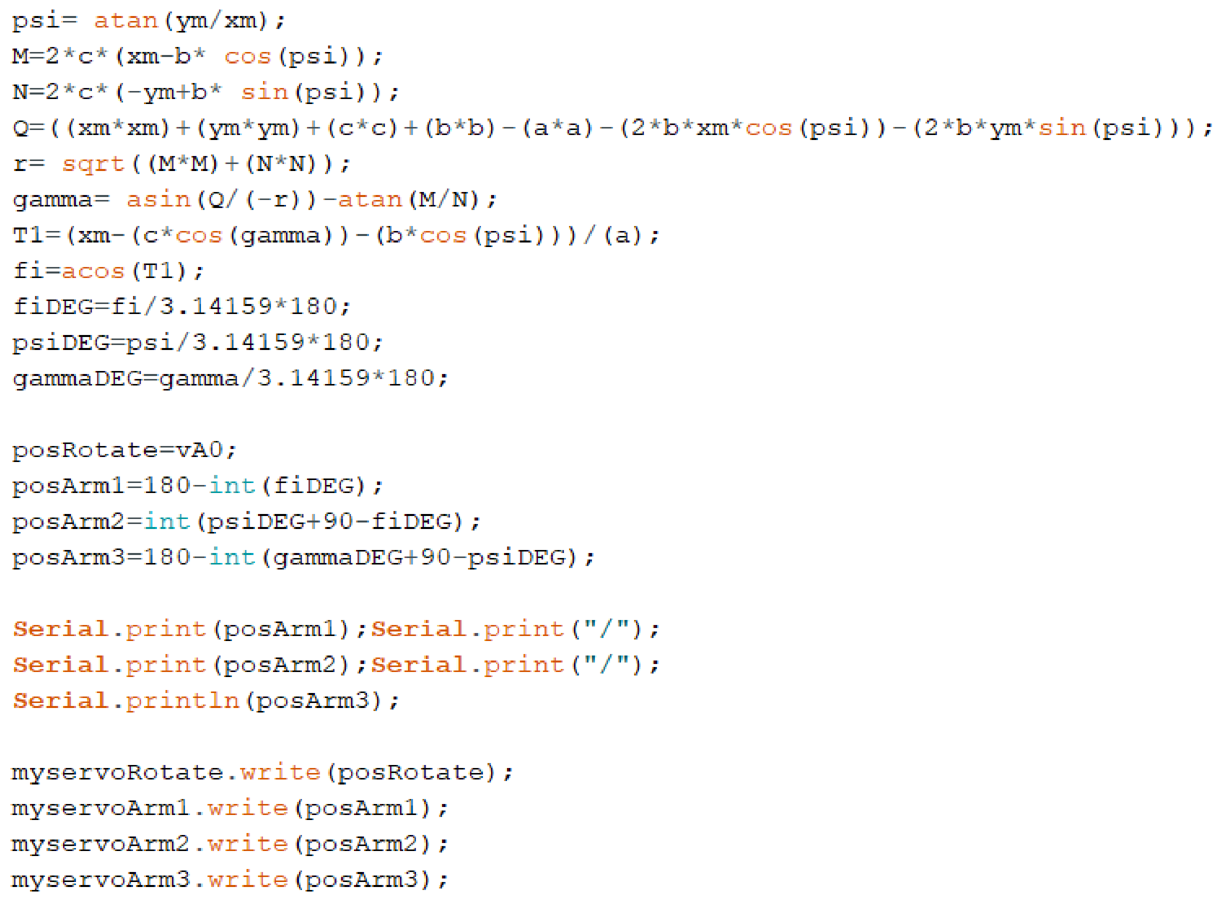
```
double xm;
double ym;
double fi;
double psi;
double gamma;
double fiDEG;
double psiDEG;
double gammaDEG;
double M;
double N;
double Q;
double T1;
double r;
```



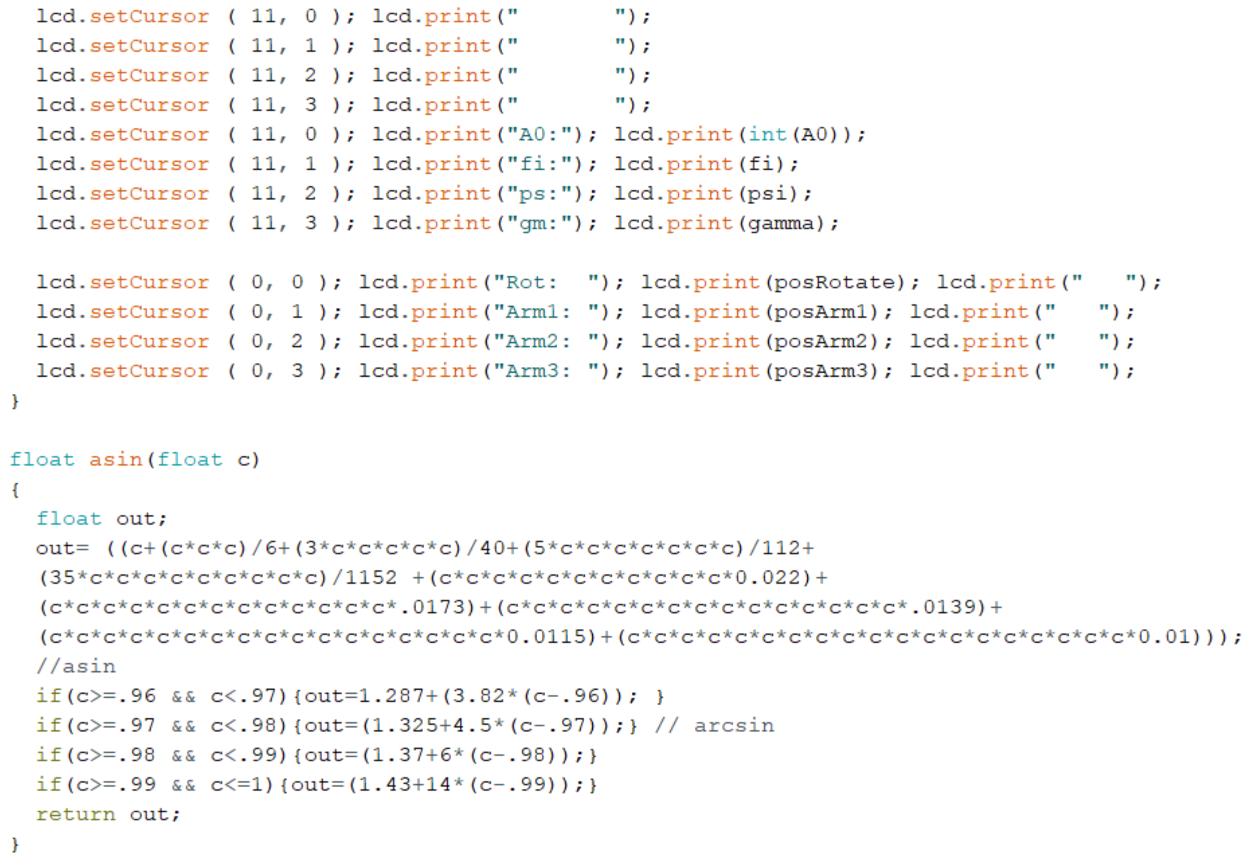
```
int posArm1 = 90;
int posArm2 = 90;
int posArm3 = 90;
int posHand = 180;
int posFinger = 180;
int velocity = 15;

int rA0;  //0..1023
int rA1;
int rA2;
int rA3;
double vA0;  // other range
double vA1;
double vA2;
double vA3;

void setup()
{

  radio.begin();
  radio.setAutoAck(1);
  radio.setRetries(0,15);
  radio.enableAckPayload();
  radio.setPayloadSize(32);
  radio.openReadingPipe(1,address[0]);
  radio.setChannel(0x60);
  radio.setPALevel (RF24_PA_MAX);
  radio.setDataRate (RF24_250KBPS);
  radio.powerUp();
  radio.startListening();
  Serial.begin(9600);
```



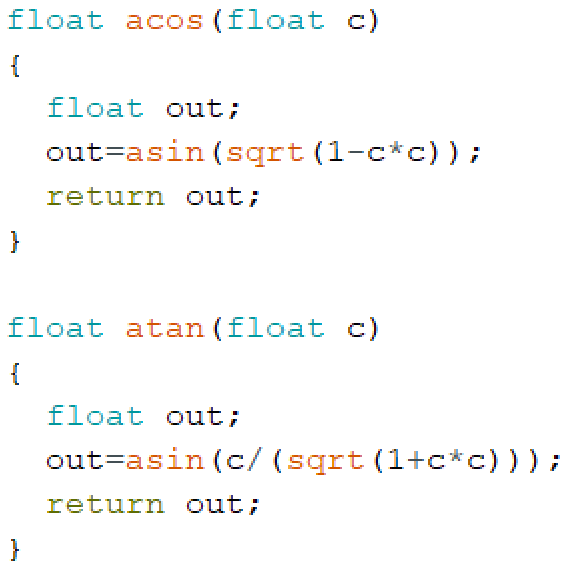
```
  xm=15;
  ym=10;
  psi= atan(ym/xm);
  M=2*c*(xm-b* cos(psi));
  N=2*c*(-ym+b* sin(psi));
  Q=((xm*xm)+(ym*ym)+(c*c)+(b*b)-(a*a)-(2*b*xm*cos(psi))-(2*b*ym*sin(psi)));
  r= sqrt((M*M)+(N*N));
  gamma= asin(Q/(-r))-atan(M/N);
  T1=(xm-(c*cos(gamma))-(b*cos(psi)))/(a);
  fi=acos(T1);
  Serial.print("psi=");  Serial.println(psi,5);
  Serial.print("M=");    Serial.println(M,5);
  Serial.print("N=");    Serial.println(N,5);
  Serial.print("Q=");    Serial.println(Q,5);
  Serial.print("r=");    Serial.println(r,5);
  Serial.print("gamma=");Serial.println(gamma,5);
  Serial.print("T1=");   Serial.println(T1,5);
  Serial.print("fi=");   Serial.println(fi,5);

  myservoRotate.attach(36);
  myservoArm1.attach(35);
  myservoArm2.attach(34);
  myservoArm3.attach(33);
  myservoHand.attach(32);
  myservoFinger.attach(31);

  myservoRotate.write(posRotate);
  myservoArm1.write(posArm1);
  myservoArm2.write(posArm2);
  myservoArm3.write(posArm3);
  myservoHand.write(posHand);
  myservoFinger.write(posFinger);
  delay(3000);

  lcd.init(); // initialize the lcd
  lcd.backlight();
  lcd.setCursor ( 0, 0 ); lcd.print("Rot:  "); lcd.print(posRotate); lcd.print("   ");
  lcd.setCursor ( 0, 1 ); lcd.print("Arm1: "); lcd.print(posArm1); lcd.print("   ");
  lcd.setCursor ( 0, 2 ); lcd.print("Arm2: "); lcd.print(posArm2); lcd.print("   ");
  lcd.setCursor ( 0, 3 ); lcd.print("Arm3: "); lcd.print(posArm3); lcd.print("   ");

}

void loop()
{
  delay(50);
  rA0 = analogRead(A0);
  rA1 = analogRead(A1);
  rA2 = analogRead(A2);
  rA3 = analogRead(A3);
  vA0 = rA0/5.67;
  vA1 = rA1/5.67;
  vA2 = rA2/5.67;
  vA3 = rA3/5.67;

  radio.read( &xmymValues, sizeof(xmymValues));
  xm=xmymValues[0];
  ym=xmymValues[1];
```



```
  psi= atan(ym/xm);
  M=2*c*(xm-b* cos(psi));
  N=2*c*(-ym+b* sin(psi));
  Q=((xm*xm)+(ym*ym)+(c*c)+(b*b)-(a*a)-(2*b*xm*cos(psi))-(2*b*ym*sin(psi)));
  r= sqrt((M*M)+(N*N));
  gamma= asin(Q/(-r))-atan(M/N);
  T1=(xm-(c*cos(gamma))-(b*cos(psi)))/(a);
  fi=acos(T1);
  fiDEG=fi/3.14159*180;
  psiDEG=psi/3.14159*180;
  gammaDEG=gamma/3.14159*180;

  posRotate=vA0;
  posArm1=180-int(fiDEG);
  posArm2=int(psiDEG+90-fiDEG);
  posArm3=180-int(gammaDEG+90-psiDEG);

  Serial.print(posArm1);Serial.print("/");
  Serial.print(posArm2);Serial.print("/");
  Serial.println(posArm3);

  myservoRotate.write(posRotate);
  myservoArm1.write(posArm1);
  myservoArm2.write(posArm2);
  myservoArm3.write(posArm3);
```



```
  lcd.setCursor ( 11, 0 ); lcd.print("       ");
  lcd.setCursor ( 11, 1 ); lcd.print("       ");
  lcd.setCursor ( 11, 2 ); lcd.print("       ");
  lcd.setCursor ( 11, 3 ); lcd.print("       ");
  lcd.setCursor ( 11, 0 ); lcd.print("A0:"); lcd.print(int(A0));
  lcd.setCursor ( 11, 1 ); lcd.print("fi:"); lcd.print(fi);
  lcd.setCursor ( 11, 2 ); lcd.print("ps:"); lcd.print(psi);
  lcd.setCursor ( 11, 3 ); lcd.print("gm:"); lcd.print(gamma);

  lcd.setCursor ( 0, 0 ); lcd.print("Rot:  "); lcd.print(posRotate); lcd.print("   ");
  lcd.setCursor ( 0, 1 ); lcd.print("Arm1: "); lcd.print(posArm1); lcd.print("   ");
  lcd.setCursor ( 0, 2 ); lcd.print("Arm2: "); lcd.print(posArm2); lcd.print("   ");
  lcd.setCursor ( 0, 3 ); lcd.print("Arm3: "); lcd.print(posArm3); lcd.print("   ");
}

float asin(float c)
{
  float out;
  out= ((c+(c*c*c)/6+(3*c*c*c*c*c)/40+(5*c*c*c*c*c*c*c)/112+
  (35*c*c*c*c*c*c*c*c*c)/1152 +(c*c*c*c*c*c*c*c*c*c*c*0.022)+
  (c*c*c*c*c*c*c*c*c*c*c*c*c*.0173)+(c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*.0139)+
  (c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*0.0115)+(c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*0.01)));
  //asin
  if(c>=.96 && c<.97){out=1.287+(3.82*(c-.96)); }
  if(c>=.97 && c<.98){out=(1.325+4.5*(c-.97));} // arcsin
  if(c>=.98 && c<.99){out=(1.37+6*(c-.98));}
  if(c>=.99 && c<=1){out=(1.43+14*(c-.99));}
  return out;
}
```



```
float acos(float c)
{
  float out;
  out=asin(sqrt(1-c*c));
  return out;
}

float atan(float c)
{
  float out;
  out=asin(c/(sqrt(1+c*c)));
  return out;
}
```

8. CONCLUSION

Study of Industry 4.0 was made and then started with a scheme where the manipulator is part of the Industry. The control of the manipulator was tried in different methods before achieving the final solution. This method was selected because of its simple calculation and because of its accuracy.

So, Design of control of manipulator was successfully done by using the vector method and by calculating the angles of servos using some formulas and then verifying it by using some simple values for the length of the arms and some imaginary coordinate points. The Values of the coordinates are given to the robot through the WIFI module from the other robots which are the part of Industry 4.0. This Manipulator is used for picking and placing objects from one place to another in the scheme. The Arduino in the robot arm will be programmed in such a way that it can receive the information from the other robots Arduino. There are even several study programs tried with this robot because it’s an education model. This was an improvised version since first this robot was made work with the joystick but it is working with the WIFI module.

SCOPE OF THE THESIS:

The Manipulator in the smart factory can be further improvised to the requirements needed for the other smart factories.

- The Formulas and calculation made for the arm can be made more accurate by calculating it in some software’s like MATLAB which can increase the precision of the robot’s movement.
- The Robot arm can be having some sensors in the grippers and the other vehicle which sends the command through the WIFI module also can have a sensor so that the arm can accurately reach the position or coordinate point.

9. REFERENCES

[1] D.-I. Roman., “On the road to Industry 4.0,” It's OWL Cluster management GmbH. Solutions from the Leading-Edge Cluster it's, 01 November 2014. [Online]. Available: http://www.its-owl.com/fileadmin/PDF/Informationsmaterialien/. [Accessed 03 January 2018].

[2] “ISCOOP-Better quality products real time monitoring IoT enabled quality improvement and cobots,” [Online]. Available: https://www.i-scoop.eu/industry-4-0/#Better_quality_products_real-time_monitoring_IoT-enabled_quality_improvement_and_cobots. [Accessed 03 January 2018].

[3] “Kingstar,” [Online]. Available: https://kingstar.com/industry-4-0-key-design-principles/. [Accessed 05 January 2018].

[4] “Industry 4.0,” [Online]. Available: https://image.slidesharecdn.com/industry4-161112133940/95/industry-40-8-638.jpg?cb=1478958592. [Accessed 05 January 2018].

[5] “SCOR,” [Online]. Available: https://www.scor.com/en/file/20206/download?token=KjT5rxfI. [Accessed 07 January 2018].

[6] E. HOZDIĆ, “International Journal of Modern Manufacturing Technologies” Smart factory for industry 4.0 A review [online]. 2016 [cit. 2017-01-05].

[7] “Hubspot,” [Online]. Available: http://cdn2.hubspot.net/hubfs/687152/featured_images/smart_industry.jpg. [Accessed 07 January 2018].

[8] P. WRIGHT, “Cyber -physical product manufacturing,” [online]. 2014. [Online]. Available: https://www.researchgate.net/publication/262879033_Cyber-physical_product_manufacturingwww.researchgate.net. [Accessed 15 January 2018].

[9] “Time- Rockwell Automation: What is Smart Manufacturing,” [online]. 2009. [Online]. Available: https://www.rockwellautomation.com/resources/downloads/rockwellautomation/pdf/about-us/company-overview/TIMEMagazineSPMcoverstory.pdf. [Accessed 10 January 2018].

[10] B. K. H.-A. Y. S. a. L. J. BAGHERI, “Cyber-physical Systems Architecture for Self-Aware Machines in Industry 4.0 Environment,” IFAC (International Federation of Automatic Control), Hosting by Elsevier Ltd, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896315005571.

[11] J. T. C. D. R. a. P. GAUSEMEIER, Universität. Der Weg zu Intelligenten Technischen System 2013. [Online]. Available: http://www.gito.de/homepage/im/imhp.nsf/0/B0BB1B1F4355A4ABC1257B09.

[12] P. D. H. A. S. S. W. KAGERMANN, "SMART SERVICE WELT Umsetzungsempfehlungen für das Zukunftsprojekt," Internetbasierte Dienste für die Wirtschaft, 2014. [Online]. Available: http://www.acatech.de/filead. [Accessed 17 January 2018].

[13] [Online]. Available: https://www.introtoarduino.com/downloads/IntroArduinoBook.pdf. [Accessed 22 January 2018].

[14] "Arduino Learning: Getting Started with Arduino," Arduino, 2018. [Online]. Available: https://www.arduino.cc. [Accessed 22 January 2018].

[15] [Online]. Available: https://image.slidesharecdn.com/introductiontoarduino-130219180141-phpapp01/95/introduction-to-arduino-7-638.jpg?cb=1423564632. [Accessed 25 January 2018].

[16] KLEISS, "Ronalden and WINKOOP, Henk van. AREXX Engineering," 2005. [Online]. Available: http://www.arexx.nl/arexx.php?cmd=goto&cparam=home. [Accessed 06 February 2018].

[17] J. G. a. S. C. Selvi, in A Text Book of Electrical Drives and Controls, Anuradha Publications, 2012.

[18] [Online]. Available: https://www.electrical4u.com/electrical-drives/. [Accessed 20 February 2018].

[19] [Online]. Available: https://www.google.cz/search?q=servo+motor+images&safe=active&source=lnms&tbm=isch&sa=X&ved=0ahUKEwif_ayp5ILbAhXFaFAKHUzpCgEQ_AUICigB&biw=1366&bih=623#imgrc=RU1-nMAmXWankM:. [Accessed 15 March 2018].

[20] M. P.Jaganathan M.E, "Inverse Kinematics," in Robotics, Lakshmi Publications, 2015.

10.APPENDICES INDEX

10.1. Appendix A - Study Program 1

```
#include <Servo.h>
Servo myservoRotate;
Servo myservoArm1;
Servo myservoArm2;
Servo myservoArm3;
Servo myservoHand;
Servo myservoFinger;
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);

int posRotate = 90;
int posArm1 = 90;
int posArm2 = 90;
int posArm3 = 90;
int posHand = 180;
int posFinger = 180;
int velocity = 1;

void setup()
{
  myservoRotate.attach(36);
  myservoArm1.attach(35);
  myservoArm2.attach(34);
  myservoArm3.attach(33);
  myservoHand.attach(32);
  myservoFinger.attach(31);

  myservoRotate.write(posRotate);
  myservoArm1.write(posArm1);
  myservoArm2.write(posArm2);
  myservoArm3.write(posArm3);
  myservoHand.write(posHand);
  myservoFinger.write(posFinger);

lcd.init(); // initialize the lcd
lcd.backlight();
lcd.setCursor ( 0, 0 ); lcd.print("Rot:  "); lcd.print(posRotate); lcd.print("   ");
lcd.setCursor ( 0, 1 ); lcd.print("Arm1: "); lcd.print(posArm1); lcd.print("   ");
lcd.setCursor ( 0, 2 ); lcd.print("Arm2: "); lcd.print(posArm2); lcd.print("   ");
lcd.setCursor ( 0, 3 ); lcd.print("Arm3: "); lcd.print(posArm3); lcd.print("   ");

}

void loop()
{
  for(posRotate = 90; posRotate <= 180; posRotate++)  //ROTATE+
  {
    myservoRotate.write(posRotate);
    lcd.setCursor ( 0, 0 ); lcd.print("Rot:  "); lcd.print(posRotate); lcd.print("   ");
    delay(velocity);
  }

  delay(1000);
  for(posRotate = 180; posRotate >= 0; posRotate--) //ROTATE-
  {
    myservoRotate.write(posRotate);
    lcd.setCursor ( 0, 0 ); lcd.print("Rot:  "); lcd.print(posRotate); lcd.print("   ");
    delay(velocity);
  }

  delay(1000);
  for(posRotate = 0; posRotate <= 90; posRotate++)  //ROTATE+
  {
```

```
  myservoRotate.write(posRotate);
  lcd.setCursor ( 0, 0 ); lcd.print("Rot:  "); lcd.print(posRotate); lcd.print("   ");
  delay(velocity);
}

delay(1000);
for(posArm1 = 90; posArm1 <= 180; posArm1++)       // Arm1+
{
  myservoArm1.write(posArm1);
  lcd.setCursor ( 0, 1 ); lcd.print("Arm1: "); lcd.print(posArm1); lcd.print("   ");
  delay(velocity);
}

delay(1000);
for(posArm1 = 180; posArm1 >= 0; posArm1--)       //Arm1-
{
  myservoArm1.write(posArm1);
  lcd.setCursor ( 0, 1 ); lcd.print("Arm1: "); lcd.print(posArm1); lcd.print("   ");
  delay(velocity);
}

delay(1000);
for(posArm1 = 0; posArm1 <= 90; posArm1++)       // Arm+
{
  myservoArm1.write(posArm1);
  lcd.setCursor ( 0, 1 ); lcd.print("Arm1: "); lcd.print(posArm1); lcd.print("   ");
  delay(velocity);
}

delay(1000);
for(posArm2 = 90; posArm2 <= 180; posArm2++)       // Arm2+
{
  myservoArm2.write(posArm2);
  lcd.setCursor ( 0, 2 ); lcd.print("Arm2: "); lcd.print(posArm2); lcd.print("   ");
  delay(velocity);
 }

 delay(1000);
 for(posArm2 = 180; posArm2 >= 0; posArm2--)       //Arm2-
 {
  myservoArm2.write(posArm2);
  lcd.setCursor ( 0, 2 ); lcd.print("Arm2: "); lcd.print(posArm2); lcd.print("   ");
  delay(velocity);
 }

 delay(1000);
 for(posArm2 = 0; posArm2 <= 90; posArm2++)       // Arm2+
 {
  myservoArm2.write(posArm2);
  lcd.setCursor ( 0, 2); lcd.print("Arm2: "); lcd.print(posArm2); lcd.print("   ");
  delay(velocity);
 }

 delay(1000);
 for(posArm3 = 90; posArm3 <= 180; posArm3++)       // Arm3+
 {
  myservoArm3.write(posArm3);
  lcd.setCursor ( 0, 3 ); lcd.print("Arm3: "); lcd.print(posArm3); lcd.print("   ");
  delay(velocity);
 }

 delay(1000);
 for(posArm3 = 180; posArm3 >= 0; posArm3--)       //Arm3-
 {
  myservoArm3.write(posArm3);
  lcd.setCursor ( 0, 3 ); lcd.print("Arm3: "); lcd.print(posArm3); lcd.print("   ");
  delay(velocity);
 }
```

```
  delay(1000);
  for(posArm3 = 0; posArm3 <= 90; posArm3++)       // Arm3+
  {
    myservoArm3.write(posArm3);
    lcd.setCursor ( 0, 3); lcd.print("Arm3: "); lcd.print(posArm3); lcd.print("   ");
    delay(velocity);
  }
```

10.2. Appendix B - Study Program 2

```
#include <Servo.h>
Servo myservoRotate;
Servo myservoArm1;
Servo myservoArm2;
Servo myservoArm3;
Servo myservoHand;
Servo myservoFinger;
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);

int posRotate = 90;
int posArm1 = 90;
int posArm2 = 90;
int posArm3 = 90;
int posHand = 180;
int posFinger = 180;
int velocity = 15;

int rA0;
int rA1;
int rA2;
int rA3;
double vA0;
double vA1;
double vA2;
double vA3;

void setup()
{
  myservoRotate.attach(36);
  myservoArm1.attach(35);
  myservoArm2.attach(34);
  myservoArm3.attach(33);
  myservoHand.attach(32);
  myservoFinger.attach(31);

  myservoRotate.write(posRotate);
  myservoArm1.write(posArm1);
  myservoArm2.write(posArm2);
  myservoArm3.write(posArm3);
  myservoHand.write(posHand);
  myservoFinger.write(posFinger);
  delay(3000);

lcd.init(); // initialize the lcd
lcd.backlight();
lcd.setCursor ( 0, 0 ); lcd.print("Rot:  "); lcd.print(posRotate); lcd.print("   ");
lcd.setCursor ( 0, 1 ); lcd.print("Arm1: "); lcd.print(posArm1); lcd.print("   ");
lcd.setCursor ( 0, 2 ); lcd.print("Arm2: "); lcd.print(posArm2); lcd.print("   ");
lcd.setCursor ( 0, 3 ); lcd.print("Arm3: "); lcd.print(posArm3); lcd.print("   ");

}
void loop()
{
  delay(2);
  rA0 = analogRead(A0);
  rA1 = analogRead(A1);
  rA2 = analogRead(A2);
  rA3 = analogRead(A3);
  vA0 = rA0/5.67;
  vA1 = rA1/5.67;
  vA2 = rA2/5.67;
  vA3 = rA3/5.67;
  posRotate=vA0;
  posArm1=vA1;
  posArm2=vA2;
  posArm3=vA3;
  myservoRotate.write(posRotate);
  myservoArm1.write(posArm1);
  myservoArm2.write(posArm2);
  myservoArm3.write(posArm3);
  lcd.setCursor ( 13, 0 ); lcd.print("       ");
  lcd.setCursor ( 13, 1 ); lcd.print("       ");
  lcd.setCursor ( 13, 2 ); lcd.print("       ");
  lcd.setCursor ( 13, 3 ); lcd.print("       ");
  lcd.setCursor ( 13, 0 ); lcd.print("A0:"); lcd.print(int(vA0));
  lcd.setCursor ( 13, 1 ); lcd.print("A1:"); lcd.print(int(vA1));
  lcd.setCursor ( 13, 2 ); lcd.print("A2:"); lcd.print(int(vA2));
  lcd.setCursor ( 13, 3 ); lcd.print("A3:"); lcd.print(int(vA3));

  lcd.setCursor ( 0, 0 ); lcd.print("Rot:  "); lcd.print(posRotate); lcd.print("   ");
  lcd.setCursor ( 0, 1 ); lcd.print("Arm1: "); lcd.print(posArm1); lcd.print("   ");
  lcd.setCursor ( 0, 2 ); lcd.print("Arm2: "); lcd.print(posArm2); lcd.print("   ");
  lcd.setCursor ( 0, 3 ); lcd.print("Arm3: "); lcd.print(posArm3); lcd.print("   ");
```

10.3. Appendix C - Study Program 3

```
#include <Servo.h>
Servo myservoRotate;
Servo myservoArm1;
Servo myservoArm2;
Servo myservoArm3;
Servo myservoHand;
Servo myservoFinger;
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);

const double a=80;
const double b=81;
const double c=48;

int posRotate = 90;
int posArm1 = 90;
int posArm2 = 90;
int posArm3 = 90;
int posHand = 180;
int posFinger = 180;
int velocity = 15;

int rA0;
int rA1;
int rA2;
int rA3;
double vA0;
double vA1;
double vA2;
double vA3;
double fi;
double psi;
double gamma;

void setup()
{
  myservoRotate.attach(36);
  myservoArm1.attach(35);
  myservoArm2.attach(34);
  myservoArm3.attach(33);
  myservoHand.attach(32);
  myservoFinger.attach(31);

  myservoRotate.write(posRotate);
  myservoArm1.write(posArm1);
  myservoArm2.write(posArm2);
  myservoArm3.write(posArm3);
  myservoHand.write(posHand);
  myservoFinger.write(posFinger);
  delay(3000);

lcd.init(); // initialize the lcd
lcd.backlight();
lcd.setCursor ( 0, 0 ); lcd.print("Rot:  "); lcd.print(posRotate); lcd.print("   ");
lcd.setCursor ( 0, 1 ); lcd.print("Arm1: "); lcd.print(posArm1); lcd.print("   ");
lcd.setCursor ( 0, 2 ); lcd.print("Arm2: "); lcd.print(posArm2); lcd.print("   ");
lcd.setCursor ( 0, 3 ); lcd.print("Arm3: "); lcd.print(posArm3); lcd.print("   ");

}
void loop()
{
  delay(2);
  rA0 = analogRead(A0);
  rA1 = analogRead(A1);
  rA2 = analogRead(A2);
  rA3 = analogRead(A3);
  vA0 = rA0/5.67;
  vA1 = rA1/5.67;
  vA2 = rA2/5.67;
  vA3 = rA3/5.67;
    fi=vA1;
  psi=vA2;
  gamma=-vA3;

  posRotate=vA0;
  posArm1=fi;
  posArm2=180-psi;
  posArm3=-gamma;

  myservoRotate.write(posRotate);
  myservoArm1.write(posArm1);
  myservoArm2.write(posArm2);
  myservoArm3.write(posArm3);

  lcd.setCursor ( 13, 0 ); lcd.print("       ");
  lcd.setCursor ( 13, 1 ); lcd.print("       ");
  lcd.setCursor ( 13, 2 ); lcd.print("       ");
  lcd.setCursor ( 13, 3 ); lcd.print("       ");
  lcd.setCursor ( 13, 0 ); lcd.print("A0:"); lcd.print(int(vA0));
  lcd.setCursor ( 13, 1 ); lcd.print("fi:"); lcd.print(int(fi));
  lcd.setCursor ( 13, 2 ); lcd.print("ps:"); lcd.print(int(psi));
  lcd.setCursor ( 13, 3 ); lcd.print("gm:"); lcd.print(int(gamma));

  lcd.setCursor ( 0, 0 ); lcd.print("Rot:  "); lcd.print(posRotate); lcd.print("   ");
  lcd.setCursor ( 0, 1 ); lcd.print("Arm1: "); lcd.print(posArm1); lcd.print("   ");
  lcd.setCursor ( 0, 2 ); lcd.print("Arm2: "); lcd.print(posArm2); lcd.print("   ");
  lcd.setCursor ( 0, 3 ); lcd.print("Arm3: "); lcd.print(posArm3); lcd.print("   ");
}
```