**Robotic Arm (Manipulator) of Two Degree of Freedom using Arduino**

*A project submitted in partial fulfilment of the requirements for the Degree of*

**Bachelor of Technology**

***in***

**Mechanical Engineering**

***By***

NAME ROLL NO.

Aman Kumar Arya **18564**

Shailendra Kumar Maurya **18550**

Vikas Kumar **18559**

Saurabh Mishra **18548**

Prasann Kumar Maurya **18537**

**Under the supervision of**

**Dr. A.K. Chauhan**



**DEPARTEMENT OF MECHANICAL ENGINEERING**

**KAMLA NEHRU INSTITUTE OF TECHNOLOGY, SULTANPUR (U.P.)**

Affiliated to

**Dr. APJ ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW (U.P.)**



# CERTIFICATE

TO WHOSOEVER IT MAY CONCERN

This isto certify that **Aman Kumar Arya (**18564**), Shailendra Kumar Maurya (**18550**), Vikas Kumar (**18559**), Saurabh Mishra (**18548**), Prasann Kumar Maurya (**18537**)** has carried out the project work in this report in entitled “Robotic Arm (Manipulator) of Two Degree of Freedom using Arduino”for the award of degree of Bachelor of technology In Mechanical Engineering, Kamla Nehru Institute of Technology, Sultanpur Affliated to Dr. A.P.J Abdul Kalam Technical University Lucknow( U.P) in our supervision.

This report is the record of the candidate's own work carried out by them under our supervision and guidance. This report is the part of their B. Tech. (Mechanical Engineering) curriculum. Their performance was excellent and we wish them good luck for their bright future endeavors.

DR. AK Chauhan Prof. S.P Kutar

(Project Supervisor) (H.O.D)

# DECLARATION

We hereby declare that this submission is our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree of the university or other institute of higher learning. except where due acknowledgement has been made in text**.**

Aman Kumar Arya (18564) Shailendra Kumar Maurya (18550)

Vikas Kumar (18559) Saurabh Mishra (18548)

Prasann Kumar Maurya (18537)

# ACKNOWLEDGEMENT

We would like to take this opportunity to thank KNIT Sultanpur for providing us such a vibrant and learning atmosphere.

First and foremost, we want to convey our most sincere gratitude to Dr. A.K Chauhan, Associate Professor in Mechanical Engineering, for taking out time from his hectic schedule and guiding us- all so in the most warm and friendly manner so that we can complete our project. We really appreciate his value of guidance and encouragement which encourage us to start and complete our project

We are also very grateful to Prof. A.K Chauhan and Prof. S.P Kutar Professor & Head of Department of Mechanical Engineering, without whom this report could not be in its shape. They always guided us through the difficulties and made us understand the concepts needed for the project. Their experimental and theoretical knowledge was indeed very helpful. Finally we feel privileged in conveying our thanks to all Faculty members of Mechanical Engineering Department for their encouragement and moral support.

We are also very grateful Professors of Mechanical Engineering who have enlightened us with their knowledge during our project Finally we would also like to thank our teammates. We would also like to thanks our batchmate for their mutual support in our project.

Aman Kumar Arya (18564)

Shailendra Kumar Maurya (18550)

Vikas Kumar (18559)

Saurabh Mishra (18548)

Prasann Kumar Maurya (18537)

# 

# ABSTRACT

This report presents a robotic manipulator which simulates movement to the grip and an object is also known as Robotic Arm. The Manipulator arm is made up of a finite number of individual rigid segments, each rigid segment is called as link. This arm is controlled by Servo Motors and Arduino , the robot can be implemented in the hazardous area for material handling and other productive work and can be also used in industries. To examine torque, we consider a prototype module for the robotic manipulator and simulate the task given to it. This paper concludes with some possible applications of 2 Degrees of freedom. The robotic manipulator mechanism is based on end effector to the robotic arm.

Keywords— Robotic Manipulator, Arduino UNO, Degree of Freedom, End Effector, Servo motor, Forward Kinematics, Reverse Kinematics

Table of Contents

[CERTIFICATE 2](#_Toc105590996)

[DECLARATION 3](#_Toc105590997)

[ACKNOWLEDGEMENT 4](#_Toc105590998)

[ABSTRACT 5](#_Toc105590999)

[CHAPTER – 1 8](#_Toc105591000)

[INTRODUCTION 8](#_Toc105591001)

[What are Robots? 8](#_Toc105591002)

[Types of Robots 8](#_Toc105591003)

[Robotic Arm (Manipulator) 11](#_Toc105591004)

[CHAPTER-2 12](#_Toc105591005)

[DESIGNING OF 2-DOF ROBOTIC ARM 12](#_Toc105591006)

[Degree of Freedom 12](#_Toc105591007)

[COMPONENTS: 14](#_Toc105591008)

[ARDUINO 14](#_Toc105591009)

[ARDIUNO MEGA 2560 BOARD 18](#_Toc105591010)

[SPECIFICATION OF MEGA 18](#_Toc105591011)

[Arduino Mega Pin Configuration 19](#_Toc105591012)

[SERVO MOTOR 22](#_Toc105591013)

[CONNECTING WIRES 25](#_Toc105591014)

[BREADBOARD 25](#_Toc105591015)

[LINKS 26](#_Toc105591016)

[JOINTS 26](#_Toc105591017)

[CHAPTER – 3 27](#_Toc105591018)

[KINEMATICS 27](#_Toc105591019)

[FORWARD KINEMATICS 27](#_Toc105591020)

[INVERSE KINEMATICS 29](#_Toc105591021)

[WORKSPACE OF PROJECT 30](#_Toc105591022)

[LITERATURE REVIEW 31](#_Toc105591023)

[CHAPTER – 4 34](#_Toc105591024)

[PROJECT ANALYSIS AND WORKING 34](#_Toc105591025)

[Programming of Arduino 34](#_Toc105591026)

[Program 35](#_Toc105591027)

[CHAPTER – 5 41](#_Toc105591028)

[APPLICATION 41](#_Toc105591029)

[CHAPTER – 6 42](#_Toc105591030)

[POSITIVES AND NEGATIVES OF A ROBOTIC ARM 42](#_Toc105591031)

[CHAPTER - 7 43](#_Toc105591032)

[CONCLUSION AND FUTURE SCOPE 43](#_Toc105591033)

***Table of Figures***

[**Figure 1 -Industrial Robots** 9](#_Toc105619901)

[**Figure 2-Medical Robot** 10](#_Toc105619902)

[**Figure 3-Service Robot** 10](#_Toc105619903)

[**Figure 4-Military Robot** 11](#_Toc105619904)

[**Figure 5-Space Robot** 11](#_Toc105619905)

[**Figure 6- Robotic Arm of 2 DOF** 12](#_Toc105619906)

[**Figure 7- Arduino UNO** 16](#_Toc105619907)

[**Figure 8 - Liliypad Arduino** 16](#_Toc105619908)

[**Figure 9-Arduino RedBoard** 17](#_Toc105619909)

[**Figure 10-Arduino Mega** 18](#_Toc105619910)

[**Figure 11-Arduino Leonardo** 18](#_Toc105619911)

[**Figure 12-Arduino Mega board Configuration** 20](#_Toc105619912)

[**Figure 13-Servo MG995** 23](#_Toc105619913)

[**Figure 14** 24](#_Toc105619914)

[**Figure 15-Breadboard** 27](#_Toc105619915)

[**Figure 16-Workspace** 32](#_Toc105619916)

# CHAPTER – 1

# INTRODUCTION

## What are Robots?

A Robot is a type of automated machine that can execute specific tasks with little or no human intervention and with speed and precision. It is basically an electro–mechanical machine that is controlled by means of computer and electronic programming. A robot is designed to perform tasks that are considered too dangerous to be performed by humans.

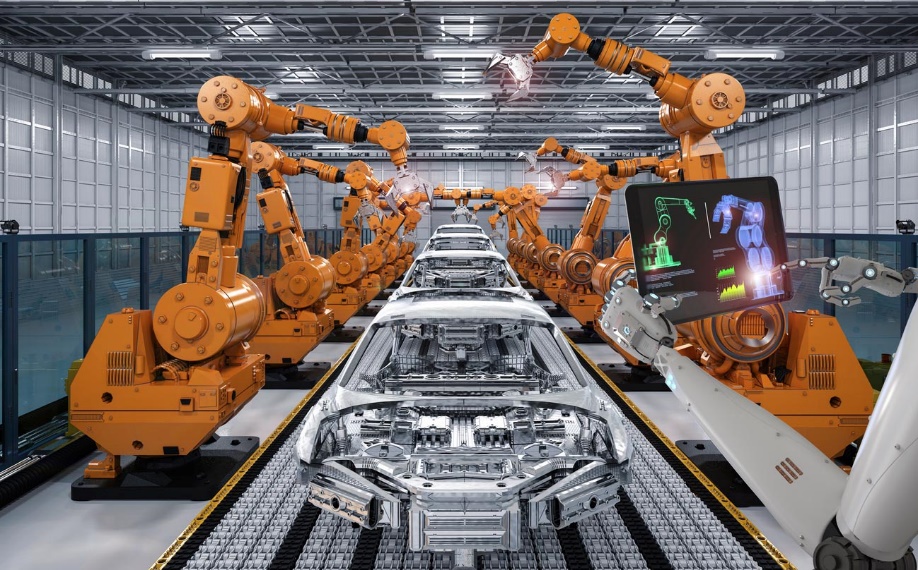
In the beginning it was a very tough task to define the term ‘Robot’ since it was continuously evolving. However the most widely accepted definition of robot was given by the Robotic Institute of America in 1979. It defined robot as a reprogrammable, multifunctional manipulator designed to move materials, parts, tools through various programmed motion to perform a variety of tasks.

## Types of Robots

There could be various bases to classify a robot for example – On the basis of their use or application, On the basis of their Kinematics.

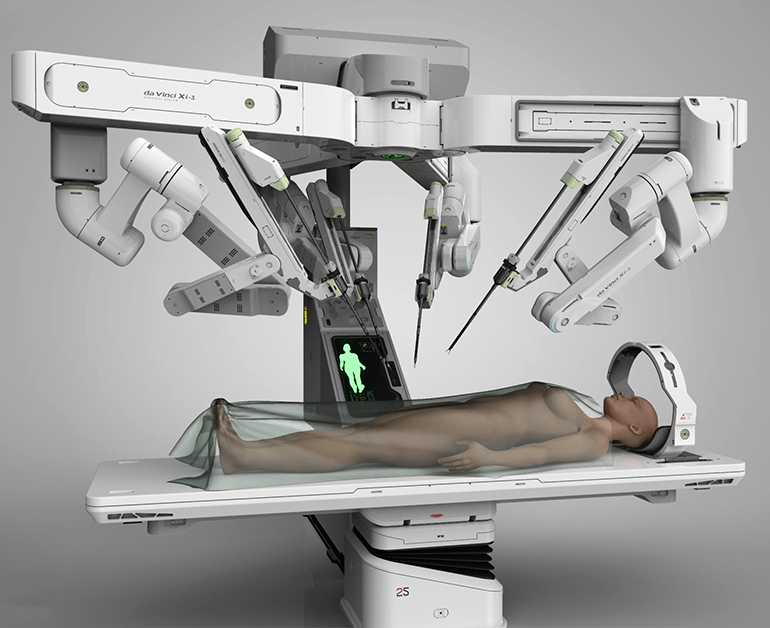
On the basis of their use and application robots can be classified as –

1. **Industrial Robots –** Robots employed in an industrialized manufacturing atmosphere. Typically they are robotic arms particularly created for application like material handling, painting and welding.



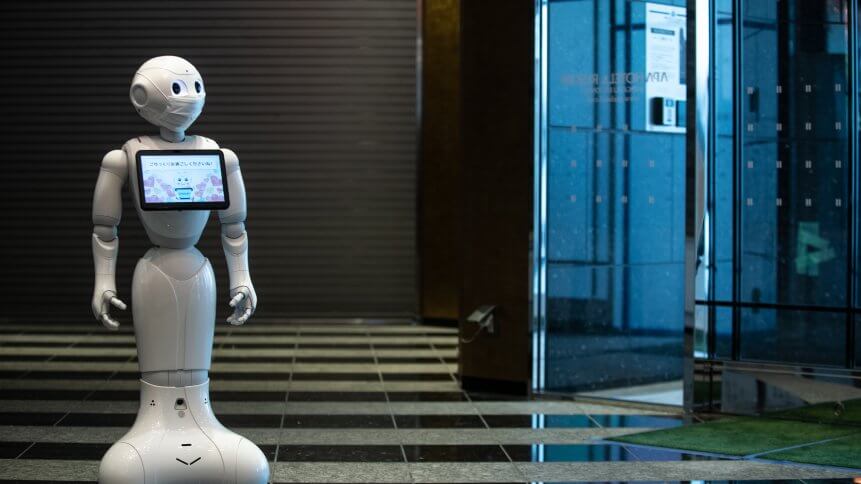
**Figure 1 -Industrial Robots**

1. **Medical Robots –** Robots employed in medicine and medicinal institutes. A typical example is the da Vinci Surgical System – a robotic surgical system.



**Figure 2-Medical Robot**

1. **Service Robots** – Robots which operate semi- or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations. Examples include lawn mowing robots, vacuum cleaning robots and sewer robots.



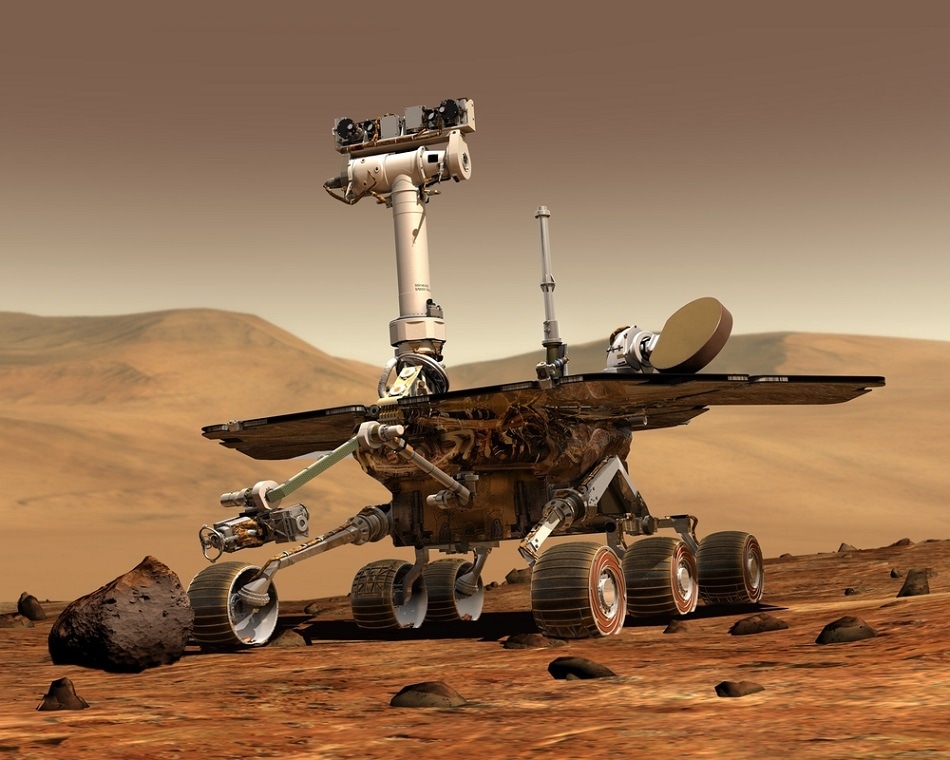
**Figure 3-Service Robot**

1. **Military Robots** – Robots brought into play in military and armed forces. They consist of bomb discarding robots, shipping robots and exploration drones.



**Figure 4-Military Robot**

1. **Space Robots** – Robots employed in space exploration and other space activities. A Common example is the mars exploration rover (International Federation of Robotics, 2003).



**Figure 5-Space Robot**

## Robotic Arm (Manipulator)

A Robotic arm is a robotic manipulator, usually programmable, with similar functions to a human arm. Of all robotic mechanical systems, Robotic manipulators have always been given special attention for various reasons. A Robotic arm can be used for various tasks such as welding, drilling, spraying and many more. It may be the sum total of the mechanism or may be part of a more complex robot. The arms are made of the links and joints that are controlled with the help of servo motors and a programmable electronic device called Arduino.

A typical robotic arm has the following components:

* Links and joints
* Actuators
* Controller
* End-effector
* Sensor (not present in some robots)



**Figure 6- Robotic Arm of 2 DOF**

# CHAPTER-2

# DESIGNING OF 2-DOF ROBOTIC ARM

The Robotic Arm is designed using the Arduino i.e. ATMEGA328p Micro-controller using Arduino programming. This process works on the principle of interfacing servos. This task is achieved by using Arduino board. Arduino boards are able to read inputs light on a sensor and turn it into an output, activating a motor, turning on a LED

We are using 2 DoF Robotic Arm which consist of 3 links and 2 lower pairs

Degree of Freedom : Degree of Freedom (DoF) is anindependent joint that can provide freedom of movement of the manipulator, either in a rotational or translational (linear) sense.

Mathematically We can Calculate DoF by Kutzbach Equation:

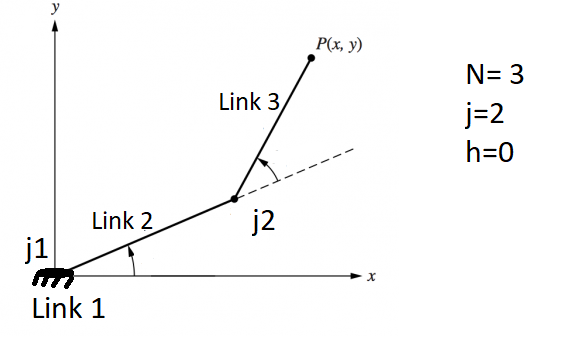
**DOF**= 3(N-1)-2j-h

Where;

N= Number of Links

j = Number of Lower Pairs

h= Number of Higher Pairs



DOF = 3 (3-1) - (2\*2) – 0

DOF = 6 – 4

DOF = 2

## COMPONENTS:

### ARDUINO

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.[1]

**What Does it Do?**

The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-TV.

the Arduino can be used as the brains behind almost any electronics project.

#### TYPES OF ARDUINO

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. Some of them are as following

##### ARDUINO UNO

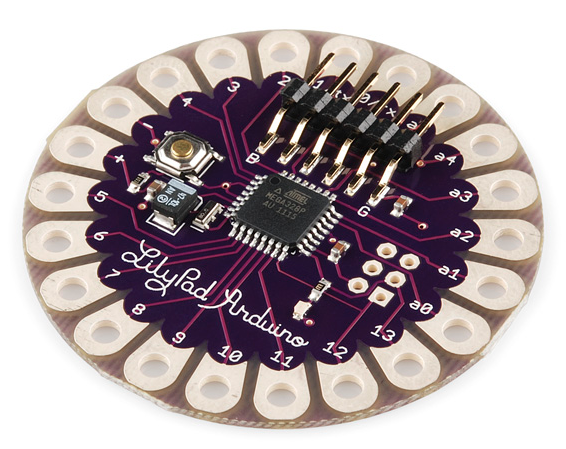
It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection, a power jack, a reset button and more. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

[](https://cdn.sparkfun.com/assets/9/1/e/4/8/515b4656ce395f8a38000000.png)

**Figure 7- Arduino UNO**

##### [LILYPAD ARDUINO](https://www.sparkfun.com/products/9266)

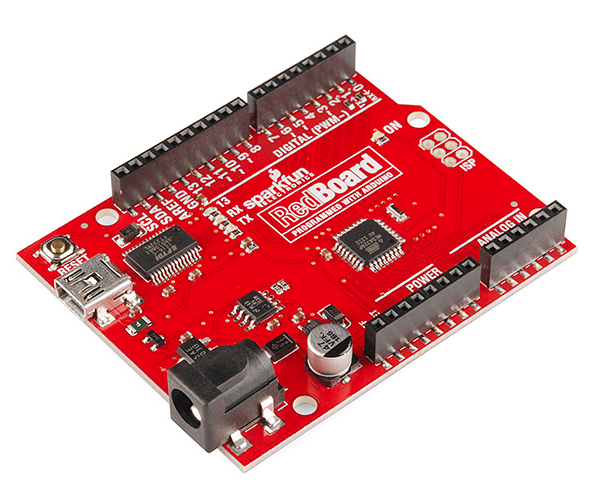
The LilyPad Arduino is considered as other Arduino board type that is designed for integrating with wearable projects and e-textile projects. This board comes in round shape that helps to decrease the snagging and can be easily connected to other devices. This board uses the Atmega328 microcontroller and Arduino bootloader in it. This board uses very less external component in it that makes the design easy and compatible

[](https://cdn.sparkfun.com/assets/3/b/6/2/6/515b5139ce395f314e000000.png)

**Figure 8 - Liliypad Arduino**

##### ARDUINO [RED BOARD](https://www.sparkfun.com/products/11575)

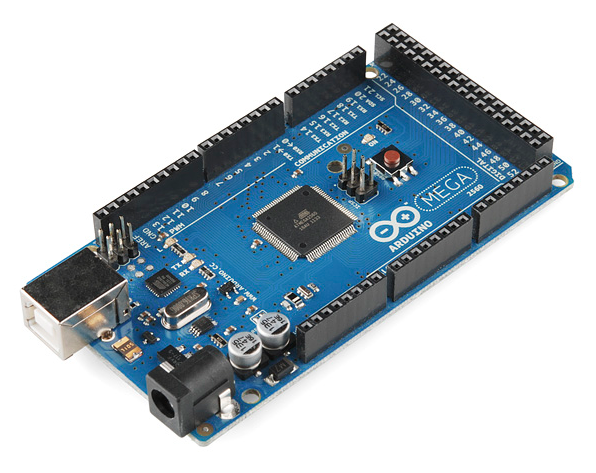
The Arduino Red board is another type of Arduino board that uses the mini USB cable for getting programmed and the Arduino IDE is used for this purpose. as The Red board uses the FTDI chip and USB chip for the connection to other device. As the design of red board is very simple it can be easily integrate with other projects. The only requirement if to plug the red board and select appropriate option and can upload program in no time. The barrel jack can be used to control the USB cable of the Arduino Red board.

[](https://cdn.sparkfun.com/assets/0/9/4/7/3/515b53d7ce395fc351000000.png)

**Figure 9-Arduino RedBoard**

##### [Arduino Mega (R3)](https://www.sparkfun.com/products/11061)

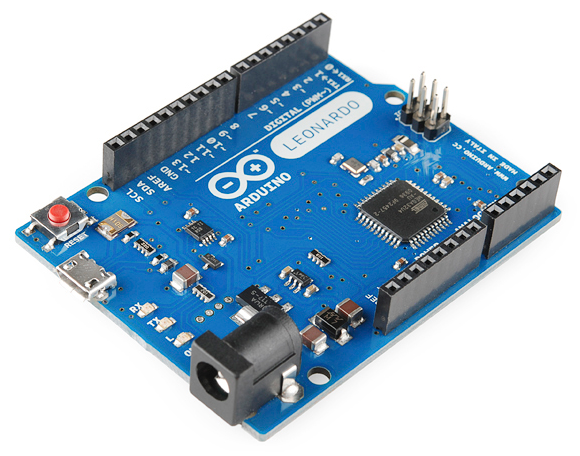
The microcontroller board like “Arduino Mega” depends on the ATmega2560 microcontroller. It includes digital input/output pins-54, where 16 pins are analog inputs, 14 are used like PWM outputs hardware serial ports ([UARTs](https://www.elprocus.com/basics-of-uart-communication-block-diagram-applications/)) – 4, a [crystal oscillator](https://www.elprocus.com/crystal-oscillator-circuit-and-working/)-16 MHz, an ICSP header, a power jack, a USB connection, as well as an RST button. This board mainly includes everything which is essential for supporting the microcontroller. So, the power supply of this board can be done by connecting it to a PC using a USB cable, or battery or an AC-DC adapter. This board can be protected from the unexpected electrical discharge by placing a base plate.

[](https://cdn.sparkfun.com/assets/9/2/3/f/f/515b547fce395f853c000007.png)

**Figure 10-Arduino Mega**

##### [Arduino Leonardo](https://www.sparkfun.com/products/11286)

The Leonardo is Arduino's first development board to use one microcontroller with built-in USB. This means that it can be cheaper and simpler. Also, because the board is handling USB directly, code libraries are available which allow the board to emulate a computer keyboard, mouse, and more!

[](https://cdn.sparkfun.com/assets/3/2/0/f/1/515b5745ce395fc83c000001.png)

**Figure 11-Arduino Leonardo**

##### SHIELDS

Additionally, there are these things called **shields** -- basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities -- [controlling motors](https://www.sparkfun.com/products/9815), [connecting to the internet](https://www.sparkfun.com/products/9026), [providing cellular](https://www.sparkfun.com/products/9607) or [other wireless communication](https://www.sparkfun.com/products/11018), [controlling an LCD screen](https://www.sparkfun.com/products/9363), and [much more](https://www.sparkfun.com/products/9595).

## ARDIUNO MEGA 2560 BOARD

Arduino board is an open-source microcontroller board which is based on AT-mega 2560 microcontroller. The growth environment of this board executes the processing or wiring language. These boards have recharged the automation industry with their simple to utilize platform wherever everybody with small otherwise no technical backdrop can start by discovering some necessary skills to program as well as run the [Arduino board](https://www.elprocus.com/arduino-basics-and-design/). These boards are used to extend separate interactive objects otherwise we can connect to software on your PC like MaxMSP, Processing, and Flash.

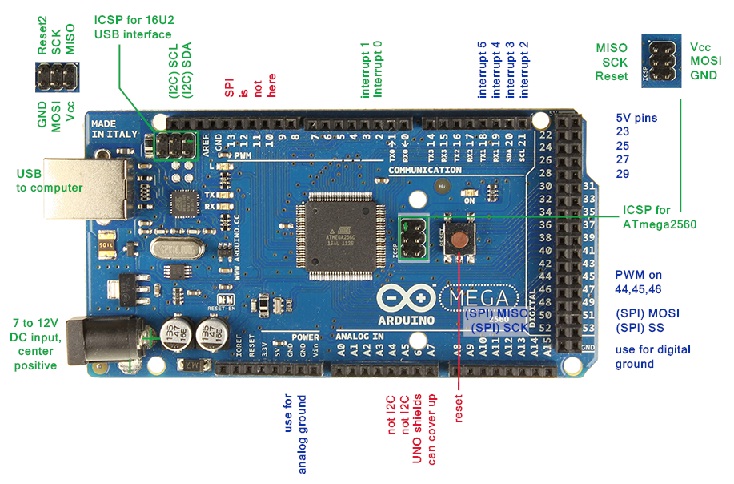
### SPECIFICATION OF MEGA

The specifications of Arduino Mega include the following.

* The ATmega2560 is a Microcontrolle
* The operating voltage of this microcontroller is 5volts
* The recommended Input Voltage will range from 7volts to 12volts
* The input voltage will range from 6volts to 20volT
* The digital input/output pins are 54 where 15 of these pins will supply PWM o/p.
* Analog Input Pins are 16
* DC Current for each input/output pin is 40 mA
* DC Current used for 3.3V Pin is 50 mA
* Flash Memory like 256 KB where 8 KB of flash memory is used with the help of bootloader
* The static random access memory (SRAM) is 8 KB
* The electrically erasable programmable read-only memory (EEPROM) is 4 KB
* The clock (CLK) speed is 16 MHz
* The USB host chip used in this is MAX3421E
* The length of this board is 101.52 mm
* The width of this board is 53.3 mm
* The weight of this board is 36 g

### Arduino Mega Pin Configuration

The pin configuration of this **Arduino mega 2560** board is shown below. Every pin of this board comes by a particular function which is allied with it. All analog pins of this board can be used as digital I/O pins. By using this board, the Arduino mega projected can be designed. These boards offer flexible work memory space is the more & processing power that permits to work with different types of sensors without delay. When we compare with other type of arduino these boards are physically superior.



**Figure 12-Arduino Mega board Configuration**

**Pin 3.3V & 5V**

These pins are used for providing o/p regulated voltage approximately 5V. This [RPS (regulated power supply)](https://www.elprocus.com/regulated-power-supply-circuit-working-applications/) provides the power to [the microcontroller](https://www.elprocus.com/arm7-based-lpc2148-microcontroller-pin-configuration/) as well as other components which are used over the Arduino mega board. It can be attained from Vin-pin of the board or one more regulated voltage supply-5V otherwise [USB cable](https://www.elprocus.com/can-interface-to-usb/), whereas another voltage regulation can be offered by 3.3V0-pin. The max power can be drawn by this is 50mA.

**GND Pin**

The Arduino mega board includes 5-GND pins where one of these pins can be used whenever the project requires.

**Reset (RST) Pin**

The RST pin of this board can be used for rearranging the board. The board can be rearranged by setting this pin to low.

**Serial Communication**

The serial pins of this board like TXD and RXD are used to transmit & receive the serial data. Tx indicates the transmission of information whereas the RX indicates receive data.

**LED**

This Arduino board includes a [LED](https://www.elprocus.com/mains-operated-led/) and that is allied to pin-13 which is named as digital pin 13. This LED can be operated based on the high and low values of the pin. This will give you to modify the programming skills in real time.

**Analog Pins**

There are 16-analog pins included on the board which is marked as A0-A15. It is very important to know that all the analog pins on this board can be utilized like digital I/O pins. Every analog pin is accessible with the 10-bit resolution which can gauge from GND to 5 volts. But, the higher value can be altered using AREF pin as well as the function of analog Reference ().

**I2C**

The [I2C communication](https://www.elprocus.com/i2c-bus-protocol-tutorial-interface-applications/) can be supported by two pins namely 20 & 21 where 20-pin signifies Serial Data Line (SDA) which is used for holding the data & 21-pin signifies Serial Clock Line (SCL ) mostly utilized for offering data synchronization among the devices

**SPI Communication**

The term SPI is a serial peripheral interface which is used to transmit the data among the controller & other components. Four pins like MISO (50), MOSI (51), SCK (52), and SS (53) are utilized for the communication of SPI.

**PWM PINS**

PWM(PULSE WITH MODERATION) are the pins which convert digital input to analog output . there are 15 pins of PWM in MEGA 2560 .

**External Interrupts**

The external interrupts can be formed by using 6-pins like interrupt 0(0), interrupt 1(3), interrupt 2(21), interrupt 3(20), interrupt 4(19), interrupt 5(18). These pins produce interrupts by a number of ways i.e. Providing LOW value, rising or falling edge or changing the value to the interrupt pins.

**Vin Pin**

The range of supplied input voltage to the board ranges from 7volts to 20volts. The voltage provided by the power jack can be accessed through this pin. However, the output voltage through this pin to the board will be automatically set up to 5V.

**PROGRAMMING**

The programming of an Arduino Mega 2560 can be done with the help of an IDE (Arduino Software), and it supports C-programming language. Here the sketch is the code in the software which is burned within the software and then moved to the Arduino board using a USB cable.

### SERVO MOTOR

A servomotor (or servo motor) is **a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration.** It consists of a suitable motor coupled to a sensor for position feedback. Servomotors are used in applications such as robotics, [CNC machinery](https://www.engineeringchoice.com/cnc-machining/), or [automated manufacturing.](https://www.engineeringchoice.com/additive-manufacturing/)[2]



**Figure 13-Servo MG995**

#### WORKING

It consists of three parts:

1. Controlled device
2. Output sensor
3. Feedback system

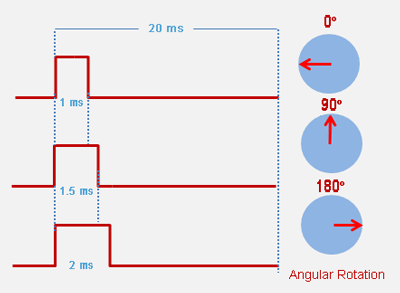
It is a closed-loop system where it uses a positive feedback system to control motion and the final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

Here reference input signal is compared to the reference output signal and the third signal is produced by the feedback system. And this third signal acts as an input signal to the control the device. This signal is present as long as the feedback signal is generated or there is a difference between the reference input signal and reference output signal. So the main task of servomechanism is to maintain the output of a system at the desired value at presence of noises.

#### **Controlling Servo Motor:**

Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

Servo motor works on **PWM (Pulse width modulation)** principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears**. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.



**Figure 14**

#### TYPES

1. **Positional rotational servo motor**

Positional rotation servo motor is the most common type of servo motor. The shaft’s o/p rotates in about 180o. It includes physical stops located in the gear mechanism to stop turning outside these limits to guard the rotation sensor. These common servos involve in radio-controlled water, radio-controlled cars, aircraft, robots, toys and many other applications.

1. **Continuous Rotation Servo Motor**

Continuous rotation servo motor is quite related to the common positional rotation servo motor, but it can go in any direction indefinitely. The control signal, rather than set the static position of the servo, is understood as the speed and direction of rotation.

1. **Linear Servo Motor**

Linear servo motor is also similar to the positional rotation servo motor discussed above, but with an extra set of gears to alter the o/p from circular to back-and-forth. These servo motors are not simple to find, but sometimes you can find them at hobby stores where they are used as actuators in higher model airplanes.

#### APPLICATION OF SERVO MOTOR

* The servo motor is used in robotics to activate movements, giving the arm to its precise angle.
* The Servo motor is used to start, move and stop conveyor belts carrying the product along with many stages. For instance, product labeling, bottling and packaging
* The servo motor is built into the camera to correct a lens of the camera to improve out of focus images.
* The servo motor is used in robotic vehicle to control the robot wheels, producing plenty torque to move, start and stop the vehicle and control its speed.
* The servo motor is used in solar tracking system to correct the angle of the panel so that each solar panel stays to face the sun  
  The Servo motor is used in metal forming and cutting machines to provide specific motion control for milling machines
* The Servo motor is used in Textiles to control spinning and weaving machines, knitting machines, and looms
* The Servo motor is used in automatic door openers to control the door in public places like supermarkets, hospitals, and theatres

### CONNECTING WIRES

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with [breadboards](https://blog.sparkfuneducation.com/what-is-a-breadboard) and other prototyping tools in order to make it easy to change a circuit as needed. All jumper wires are basically same inspite of their colour. There are four type of jumper wires:

1. Male to Male Type
2. Male to Female Type
3. Female to Female type

**Male to Male type**

This type of wires are used in connecting female header pin of any development board (like Arduino) to other development board or breadboard. Also you can combined it with our Female jumper wire to create Male to Female jumper wire.

**Male to Female type**

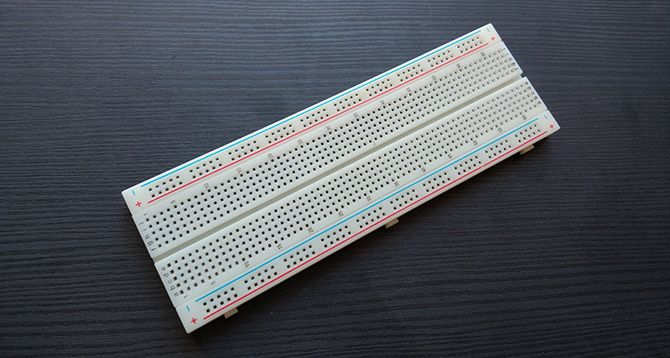
It is used in connecting female header pin of any development board (like Arduino) to other development board having male connector.

**Female to Female Type**

Itcan be used in between two male to male wires to make connection I large distances. And ca also be used in establishing quick connections.

### BREADBOARD

A breadboard is a simple device designed to let you create circuits without the need for soldering. They come in various sizes, and the design can vary, in general rule they look something like this:[3]



**Figure 15-Breadboard**

### LINKS

Links is a resistant body that constitutes part of the machine, connecting other parts which have motion relative to it.

**Types of Links**

1. **Rigid Links**

A rigid link is one that does not undergo any deformation while transmitting motion. Links, in general, are elastic in nature. They are considered rigid if they do not undergo appreciable deformation while transmitting motion.

1. **Flexible Link**

A flexible link is one which while transmitting motion is partly deformed in a manner not to affect the transmission of motion.

1. **Fluid Link**

A fluid link is one that is deformed by having fluid in a closed vessel and the motion is transmitted through the fluid by pressure.

1. **Floating Link**

A floating link is one that is not connected with the frame.

### JOINTS

A joint is a section which is used to connect two or more mechanical parts in a system. It can be classified as :

1. Lower Pair Joints
2. Higher Pair Joints

**Lower pair joints**

In Lower pair joints there is contact between the surfaces or areas of members.

Examples – Hinge joints/Revolute Joints

Slider Joints

**Higher pair joints**

**In** Higher Pair Joints there is contact of points or Line between members.

Examples – Cylindrical Roller

Cam Pair

# CHAPTER – 3

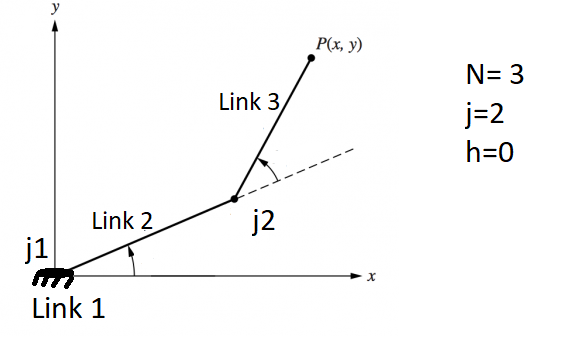
# KINEMATICS

A Robotic arm is designed to function in two different situation. One In which the end effector reaches at position by the rotation of servomotor and joints. In the second case a desired location is set and rotation of the links and joints are calculated. First one is called Forward Kinematics and Second is called Inverse Kinematics.

## FORWARD KINEMATICS

Forward kinematics refers to the use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters.

In a Cartesian co-ordinate system, let there is a link of length l1 is joint at origin (0,0), and rotated at an angle theta1 from x – axis and a second link of length of l2 joint at the end of first link and rotated at an angle of theta2 from its initial position. So now we have to calculate the end point of second link i.e. position of end effector. Hence in the diagram below –



In Tringle OAB In Tringle OAB

OA = OC=

OB = PC=

X = OA + OC

**=**

Y = OB + PC

=

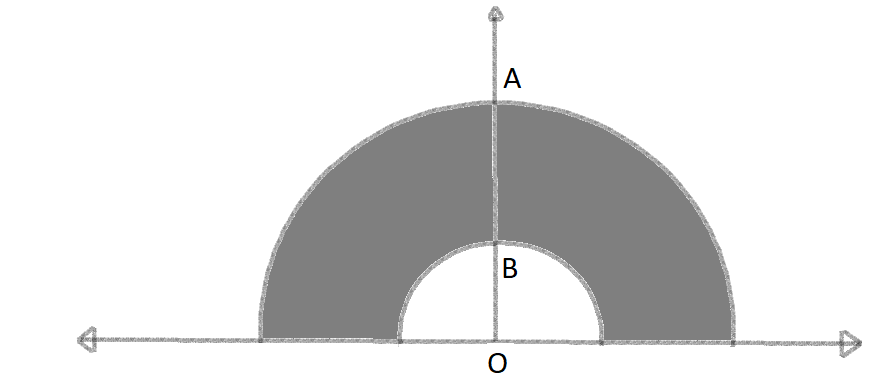
## INVERSE KINEMATICS

**inverse kinematics** is the mathematical process of calculating the variable joint parameters needed to place the end of a [kinematic chain](https://en.wikipedia.org/wiki/Kinematic_chain), such as a robot manipulator or [animation character's skeleton](https://en.wikipedia.org/wiki/Skeletal_animation), in a given position and orientation relative to the start of the chain. Given joint parameters, the position and orientation of the chain's end,

## WORKSPACE OF PROJECT

Length of first link=l1

Length of second link = l2



**Figure 16-Workspace**

Where OA is l1+l2  and OB is l1-l2

The area in which the Robotic Arm will perform its function is called Workspace of the Robotic Arm.

So Area between the circles of Radius l1+l2 and l1 – l2 will be our workspace.

< P (X,Y) <

Shaded Area is workspace of our project.

# LITERATURE REVIEW

In the early 1950s, investigators at the University of California scrutinized detailed drawings from da Vinci’s notebooks which together form a tome exceeding 1,119 pages dating from 1480 to 1518 and therefore referred to, like the great Atlantic ocean, as the Codex Atlanticus.

Modern investigations increasingly make it clear that he singularly pursued knowledge of everything known to these ancient scholars. He, in effect, was following in the footsteps of such figures as Hero of Alexandria, Philon, and Cstebius who were all reported to be interested in mechanically simulating motion and human attributes

da Vinci began a systematic method of devising and building the sophisticated mechanical device that was 500 years ahead of its time. His first robotic design was in December 1478, at the age of 26, before he moved to Milan.

In the Codex Atlanticus, folio 812, is a power mechanism that features a front wheel drive, rack-and-pinion automobile. Impressive as it is, it was also fully programmable, with the ability to control its own motion and direction. It is now thought that this “base” would form the basis of his ultimate goal, an fully functional automaton.

In 1495, at about the time he was working on his method of painting on wet plaster and the Last Supper, da Vinci designed and probably built the first of several programmable humanoid robots.

It is possible that in the recent history of the world only wars have had a more dramatic impact upon our society than expositions. The first industrial exposition occurred in Paris in 1798 and enabled the public to witness progress and technology that could change the lives of everyone. This process continued into the nineteenth century when the extraordinary potential of remote-controlled robotic devices was clearly demonstrated to an unsuspecting public at the 1898 Electrical Exhibition in Madison Square Garden, New York City. Nicola Tesla was at the height of his inventive prowess when he brought upon the unprepared world, a fully automated, remote-controlled robotic submersible boat. “Teleautomata will ultimately be produced, capable of acting as if possessed of their own intelligence, and their advent will create a revolution.” (Tesla, 1898 [14]).

Now, with the advent of electronics and the incorporation of solid-state transistors instead of vacuum tubes, the evolution of the microcircuit and more rapid computer systems, the stage was set for early modern robotic arm evolution. The first “position controlling apparatus” was patented in 1938 by Willard Pollard (Fig. 8). This was a spray finishing robotic arm that had five degrees-of-freedom and an electrical control system. Although Pollard [17] never built his arm, his design and interest in an industrial application for automated robotic arms would spur on the ingenuity of others. Harold A. Roselund [18], working for De Vilbiss, developed another sprayer that was indeed manufactured. Both arms were very sophisticated for their time, and each solved movement at the respective joints in unique ways; the electronic controller systems lacked the fidelity required to make them broadly utilizable, however. The modern era of robotics was launched by the intrepid use of these two, little known arms developed in the late 1930s.

Unimate introduced its first robotic arm in 1962 (Fig. 8) [19]. The arm was invented by George Devol and marketed by Joseph Engelberger. The first industrial arm was installed at the General Motors plant in Ternstedt, New Jersey, for automated diecasting. Ultimately, approximately 8,500 units were sold. Industrial robots graduated from the laboratory to the factory [20]. It is interesting that in this process the robotic arm’s movements and the degrees-of-freedom incorporated nautical terms for robotics—pitch, yaw, and roll.

The five principal types of robotic arm are: rectangular coordinate, polar coordinate, cylindrical coordinate, revolute coordinate, and self compliant automatic robot assembly (SCARA). Two more recent additions are called serpentine and anthropomorphic [21]. These arms can be subdivided by the types and complexity of each of their joints and control systems. The evolution of robotic arms is rapidly developing, however, and such schemes probably do more for organizing information than in defining the actual product. Applications to medicine, and surgery in particular, are ripe for companies, because classic fields of application, for example nuclear reactor work, have declined. In the past 40 years radical improvements have been made and more degrees-of-freedom are now possible. Downsizing and cost reduction will follow. Hand technologies will rapidly advance as computer-control issues improve and work at universities will find fruitful applications in industry and medicine. “Haptics” and other sensory systems will be added to advanced surgical robotics as this technology evolves.

It is necessary to discuss the two additional categories of robotic arm a bit further, because they may become more important to medical applications. The first is the serpentine robots. These devices were originally designed on the basis of the kinesiology of another complex biological joint, the spine (Fig. 11). The purpose of making serpentine robots was to produce a device with more degrees-of-freedom than the normal human arm. As computer-control algorithms advance and the means to control the complex maneuvers of >10–20 or 30 degrees-of-freedom become available, these systems have become increasingly complex. The first such systems were called “serpentine” because it was necessary for the robotic arms to “snake” through passages and pipes to inspect nuclear reactors, fuel tank baffles, and wing spars. To overcome the multiple-joint-control issues and prevent restrictive backlash, Miyake in 1986 described innovative solutions in control [22]. In 1968 the US Navy funded a spine-like arm for ocean exploration; this has been called the Scripps tensor arm. Another such ultrahigh-dexterity robotic arm, called the Articulating Mechanism, was developed by Ralph Mosher in 1969. It was a modular and low-cost alternative to the Scripps design, but was not as precise. Many of the space arms used on the United States Space Shuttle were serpentine. The arm designed by Frederick Wells in 1970 at the Marshall Space Flight Center in Huntsville, Alabama, was such a device. This arm has continued to evolve with improvements by Iwatsuka, in 1986, and by Wuenecher. Wuenecher called his device the remote control manipulator intended to aid astronauts. The Spine Robot is a Swedish-made serpentine robotic arm invented by Ove Larson and Charles Davidson in 1983. It consists of stacked ovoidal discs controlled by opposable cables. There are now many versions of this design which use bellows, U-joints, and pressurized capillary systems (Scheinman). In 1984, Motohiko Kuura designed expandable and contractible arms for serpentine applications. The final addition in this series is the 1991 modular robotic joint (MJR) arm invented by Mark Rosheim. The advantages of this system are that it has more degrees-of-freedom than the human arm, increases modularity, and is fault tolerant, if one joint fails another is capable of providing the mobility needed to accomplish its task. Why would surgeons be at all interested in more degrees-of-freedom, you might wonder? Another coming technological tour-de-force is woundless surgery. This type of complex surgery is also called peroral, transgastric endoscopic surgery, and cholecystectomies, appendectomies, and tubal bandings have already been performed [23, 24]. To achieve more complex tasks, for example nephrectomy or radical prostatectomy, an ultrahigh-dexterity robotic arm will be necessary.

# CHAPTER – 4

# PROJECT ANALYSIS AND WORKING

We are designing a robotic arm of 2 DOF with the help of some mechanical and electronic components such as

1. Two Servo Motor (MG995R)
2. Arduino Mega 2560
3. Two Kinematic Links (l1 = 12cm & l2 =9cm)
4. A Centimeter Grid or Graph Paper)
5. Jumper Wires (M-M, M-F, F-F)
6. Breadboard

## Programming of Arduino

To Program an Arduino, Firstly we need to install a software -Arduino Environment (IDE). After installing we connect our Arduino to computer/laptop. To program Arduino we write code on the IDE – verify it and upload it to Arduino. If any program is already uploaded or implemented then we can reset it using reset button on the Arduino board.

We need to give Input for our project, for that we have to use Serial Monitor where we give input several things like Length of Links, Forward Kinematics or Backward Kinematics and serial monitor will display the output according to the case.

### Program

Program to calculate the Position of the end effector or to send it to a specific Location –

#include <Servo.h>

Servo motor1;

Servo motor2;

// angles in degree and radian

// length of links of robot arm

double L1;

double L2;

double pi = 3.14159265359;

double getUserInputBlocking(const char \* message) {

double result = 0;

bool receiving = true;

bool acquisitionStarted = false;

Serial.println(message);

do {

int r = Serial.peek();

if (r != -1) { // got something

if (isdigit(r)) {

acquisitionStarted = true;

Serial.read(); // remove the byte from the incoming stream

result = 10 \* result + (r - '0'); // do the math, might overflow.

} else {

if (acquisitionStarted) {

receiving = false; // we are done

} else {

Serial.read(); // ignore that byte and remove it from the incoming stream

}

}

}

} while (receiving);

return result;

}

void setup() {

motor1.attach(11);

motor2.attach(3);

Serial.begin(9600);

motor1.write(0);

motor2.write(0);

// Serial.println("Enter the length of first arm ");

L1 = getUserInputBlocking("Enter the length of first arm ");

Serial.println(L1);

// Serial.println("Enter the length of second arm ");

L2 = getUserInputBlocking("Enter the length of second arm ");

Serial.println(L2);

Serial.println("Enter Number 1 for ForwardKinematics");

Serial.println("Enter Number 0 for inverseKinematics");

}

void loop() {

if (Serial.available()) {

char choice = Serial.read();

if (choice == '1'){ //Forwad kinematics Case

ForwardKinematics();

}

else if(choice == '0'){

inverseKinematics(); //inverse kinematics case

}

}

}

void ForwardKinematics(){

double angle1 ;

double angle2 ;

double rad\_angle1;

double rad\_angle2;

double x;

double y;

angle1 = getUserInputBlocking("Enter the angle1 in degree ");

Serial.println(angle1);

angle2 = getUserInputBlocking("Enter the angle2 in degree ");

Serial.println(angle2);

rad\_angle1 = (angle1\*pi)/180;

rad\_angle2 = (angle2\*pi)/180;

motor1.write(angle1);

delay(1000);

motor2.write(angle2);

x = L1 \* cos(rad\_angle1) +L2 \* cos(rad\_angle1 + rad\_angle2);

y = L1 \* sin(rad\_angle1) +L2 \* sin (rad\_angle1 + rad\_angle2);

delay(1000);

Serial.print("L1 = ");

Serial.println(L1);

Serial.print("L2 = ");

Serial.println(L2);

Serial.print("x = ");

Serial.println(x);

Serial.print("y = ");

Serial.println(y);

Serial.print("angle1 is ");

Serial.println(angle1);

Serial.print("angle2 is ");

Serial.println(angle2);

Serial.println("Enter Number 1 for ForwardKinematics");

Serial.println("Enter Number 0 for inverseKinematics");

}

void inverseKinematics(){

double angle1 ;

double angle2 ;

double rad\_angle1;

double rad\_angle2;

double x;

double y;

x = getUserInputBlocking("Enter the value x ");

y = getUserInputBlocking("Enter the value y ");

rad\_angle2 = acos((sq(x)+ sq(y) - sq(L1) - sq(L2)) / (2\*L1\*L2));

rad\_angle1= atan(y / x) - atan((L2\*sin(rad\_angle2)) / (L1+ L2\*cos(rad\_angle2)));

delay(1000);

angle1= (rad\_angle1\*180)/pi;

angle2= (rad\_angle2\*180)/pi;

Serial.print("x is ");

Serial.println(x);

Serial.print("y is ");

Serial.println(y);

Serial.print("angle1 is ");

Serial.println(angle1);

Serial.print("angle2 is ");

Serial.println(angle2);

Serial.println("Enter Number 1 for ForwardKinematics");

Serial.println("Enter Number 0 for inverseKinematics");

motor1.write(angle1);

motor2.write(angle2);

delay(2000);

}

CONNECTION

# CHAPTER – 5

# APPLICATION

Robotic arm is used in different fields like in mechanical industry, medical field, agriculture, transport

1. Assembly line

Robotic manipulators used in assembly lines in Mechanical & Automobile Industries. It replaces any object or product from one Assembly Line to another. Hence, it reduces time of

1. Welding Operations

Welding is one of the most common application of Robotic Arm in industrial sector. Sometimes the welders need to supervise the process. Example –: Arc welding, Spot welding etc.

1. Painting
2. Material Handling
3. Loading & Unloading
4. Mechanical Cutting
5. Gluing, Adhesive Sealing and Spraying Materials

# CHAPTER – 6

# POSITIVES AND NEGATIVES OF A ROBOTIC ARM

**Positives**

* Improvement in Productivity and Efficiency
* Reduction of Working Cost
* Provide good returns on investment
* Ability to work in risky ways and make it more safe
* Use of equipment efficiently
* Improvement in working condition in factory

**Negatives**

* Cause unemployment for manual workers
* High initial Cost
* Technical Limitations
* Requires constant monitoring
* Can perform only specific tasks

# CHAPTER - 7

# CONCLUSION AND FUTURE SCOPE

The objective of this project was to design a robotic arm of 2 DOF with the help of Arduino and to study its various application in different fields. This Arm can perform different types of task with modification in Arduino Programming and by changing the end effector according to required function. Thus it is very useful in different industrial environment mainly in manufacturing, transportation, Packaging and many more.

There is always a possibility of improvement in the design and functioning in the robotics. It has reached to very advance levels in developed countries but it is still not available or underperforming in developing countries. In the era of AI and IOT it is very important to advance with these new technologies otherwise our advancement would not be visible to this world.

**FUTURE SCOPE**(5\_6061965933050594449.pdf)

* The number of degrees of freedom of the robotic arm can be increased in order to expand its workspace, thereby making it more versatile.
* Motors with higher torque ratings can be used to power the joints so as to ensure that the robotic arm remains in position even when electric current is not supplied to the motors.
* Wireless control of the robotic arm using Bluetooth™ or Wi-Fi™ (IEEE 802.11x) technology can be implemented.
* Object detection and collision avoidance can be implemented by adding proximity sensors to the robotic arm

[1] “Getting Started with Arduino products | Arduino.” https://www.arduino.cc/en/Guide (accessed Jun. 08, 2022).

[2] “Servo Motor - Types, Working Principle & Applications.” https://www.watelectrical.com/servo-motor-types-and-working-principle/ (accessed Jun. 08, 2022).

[3] “Breadboard - an overview | ScienceDirect Topics.” https://www.sciencedirect.com/topics/engineering/breadboard (accessed Jun. 08, 2022).