MIMIC – THE ADVANCED LEAD THROUGH ROBOT

A PROJECT REPORT

Submitted by

K.DINAKARAN	**511518114005**
B.RANJITH KUMAR	**511518114018**
V.SAKTHIVEL	**511518114020**
A.SANTHOSH KUMAR	**511518114023**

in partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

MECHANICAL ENGINEERING

P.T.Lee CHENGALVARAYA NAICKER COLLEGE OF ENGINEERING AND TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **"MIMIC – THE ADVANCED LEAD THROUGH ROBOT"** is the bonafide work of "**K.DINAKARAN (511518114005), B.RANJITH KUMAR (511518114018), V.SAKTHIVEL (511518114020) and A.SANTHOSH KUMAR (511518114023),who carried out the project work under my supervision.**

SIGNATURE	**SIGNATURE**
Dr.S.BOOPATHI.M.E.,Ph.D.,	Mr.N.PRAKASH .M.E
Head of the Department,	Project Guide,
Department of Mechanical Engineering,	Department of Mechanical Engineering,
P.T.Lee.CNCET,	P.T.Lee.CNCET,
Kanchipuram-631502	Kanchipuram-631502

Submitted for viva voce Examination held on __________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We would like to express my sincere gratitude of professor **Dr.P.PALANISAMY.M.E.,Ph.D.,** principle of our collage for giving the opportunity and facilities to carry out this project.

We would like to express my sincere gratitude of professor to professor **Dr.S.BOOPATHI.M.E.,Ph.D.,** Head of Department, of Mechanical Engineering, for giving the opportunity and facilities to carry out this project.

We express my sincere gratitude to my project coordinator **Mr.R.RAMESHBABU.M.E.,** Assistant professor in Department of Mechanical Engineering.

We are in debuted to out esteemed internal project guide **Mr.PRAKASH.,M.E.,** Assistant professor in Department of Mechanical Engineering, who has not only guide us ,but also gave constructive ideas at critical time, which helped as lot.

We would like to express my gratitude to all other faculty members of the Department of Mechanical Engineering for their immense help throughout this project.

ABSTRACT

The main objective of this project is to design a lead through robot for point to point robot programming using dimensional scaling of a real-time robot to a scaled down model. In this robot, we can record, save and playback the robot motion with accuracy and precision. This root the workers to not strain to program the robot by moving the nose of huge real-time robot but through the scale downed duplicate. This ensures the method of programing much easier to use. On the actual scenario of programing, the real-time robot offers resistance to motion which causes the operator to strain to move it to required positions. This issue can be overcome through this scale down method with greater reduction in latency of relocating the nose trajectory.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PWM	PULSE WIDTH MODULATION
PTP	POINT TO POINT
A_X	ANALOG PIN X
D_X	DIGITAL PIN X
TTL	TRANSISTOR TRANSISTOR LOGIC
CLK	CLOCK
SRAM	STATIC RANDOM-ACCESS MEMORY
I/O	INPUT
O/P	OUTPUT
mS	MILLI SECOND
µS	MICRO SECOND
FOS	FACTOR OF SAFETY

CHAPTER I

INTRODUCTION

1.1 OVERVIEW:

Recently robots are widely used in a various field particularly in the industry. Despite this fact robot still requires an undeniable amount of knowledge from the operators or workers who deal with them. As a result, robots cannot be easily programmed if the operator or the worker is not experienced in robotics field.

One of the programming methods that has been introduced to make programming task user friendly is lead-through robot programming. However, the existing lead-through programming methods still requires an amount of knowledge that is not available for most of the operators and workers.

The main objective of this project is to design a lead through method for point to point robot programming using incremental encoder feedback, which can record, save and playback the robot motion while considering the accuracy and precision of the robot. To validate the method, experiments were conducted in this project, where an operator manually moves a two DOF (degree of freedom) robotic arm on a white board while the encoder feedback was recorded and later played back by the robot. Then both recorded and playback trajectories were compared and analyzed.

1.2 OBJECTIVES OF THE PROJECT:

1. The main objective of this project is to build a robot that will be embedded with a robotic arm and can be controlled using new lead through method.
2. The robot can move to remote places and do the pick & place action of objects that are dangerous and harmful. The applications of this project is vast and can

be implemented in a lot of industries.

3. To create a pick and place robot with lead through method for point to point robot programming using angular and dimensional scaling of an real-time robot to a scaled down model.
4. To increase the pick-and-package global performance in terms of flexibility, dependability and error reduction.
5. Improvement of the working conditions of operators by a proper layout design and task allocation between worker and robot.

1.3 ORGANISATION OF THESIS:

Chapter 1 describes about motivation, objective and literature review ofthe project.

Chapter 2 describes about the block diagram and design of the proposedsystem.

Chapter 3 describes about the hardware that are used in the proposedsystem.

Chapter 4 describes about the Software that are used in the proposedsystem.

Chapter 5 briefly explains about the results of all components used inthe proposed system.

Chapter 6 describes about conclusion and scope for future.

CHAPTER II

REVIEW OF LITERATURE

2.1 PROGRAMMING BY DEMONSTRATION OF PICK-AND-PLACE TASKS FOR INDUSTRIAL MANIPULATORS USING TASK PRIMITIVES (2007 INTERNATIONAL SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE IN ROBOTICS AND AUTOMATION)

In this paper Alexander Skoglund; Boyko Iliev; Bourhane Kadmiry; Rainer Palm proposes the idea of an approach to Programming by Demonstration (PbD) to simplify programming of industrial manipulators. By using a set of task primitives for a known task type, the demonstration is interpreted and a manipulator program is automatically generated. A pick-and-place task is analyzed, based on the velocity profile, and decomposed in task primitives. Task primitives are basic actions of the robot/gripper, which can be executed in a sequence to form a complete a task. For modeling and generation of the demonstrated trajectory, fuzzy time clustering is used, resulting in smooth and accurate motions. To illustrate our approach, we carried out our experiments on a real industrial manipulator.

2.2 A CASE STUDY OF CYBER-PHYSICAL SYSTEM DESIGN: AUTONOMOUS PICK-AND-PLACE ROBOT)2018 IEEE 24TH INTERNATIONAL CONFERENCE ON EMBEDDED AND REAL-TIME COMPUTING SYSTEMS AND APPLICATIONS (RTCSA))

This paper by Pei-Chi Huang; Aloysius K. Mok presents a concept of

Although modern robots in warehousing systems can perform adequately in a goods-to-person model using hand-designed algorithms that are specialized to a particular environment, developing a robotic system that is capable of handling new products at an inexpensive cost remains a challenge. A conspicuous example of this challenge is seen in Amazon's use of autonomous robots to fetch customers' orders in their massive warehouses. To encourage advance in this technology, Amazon organized the competition, Amazon Picking Challenge that asked participants to develop their own hardware and software for the general task of picking a designated set of products from inventory shelves and then placing them at a target location (called a pick-and-place task). Current technology for pick-and-place tasks is still insufficient to meet the demand for low-cost automation. Handling awkward or oddly shaped object must still depend on hand-programming or specialized robotic systems, making manufacturing automation less flexible and expensive. In this paper, we shall present the design and implementation of a software system that is a step in advancing the technology toward full automation at reasonable costs. Our system integrates a set of state-of-the-art techniques in computer vision, deep-learning, trajectory optimization, visual servoing to create a library of skills that can be composed to perform a variety of robotic tasks. We demonstrate the capability of our system for performing autonomous pick-and-place tasks with an implementation using Hoppy, an industrial robotic arm in an environment similar to the Amazon Picking Challenge.

2.3 PROGRAMMING BY VISUAL DEMONSTRATION FOR PICK-AND-PLACE TASKS USING ROBOT SKILLS (2019 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND BIOMIMETICS (ROBIO))

This paper by Peng Hao; Tao Lu; Yinghao Cai; Shuo Wang propose the idea of a vision-based robot programming system for pick-and-place tasks that can generate programs from human demonstrations. The system consists of a detection network and a program generation module. The detection network leverages convolutional pose machines to detect the key-points of the objects. The network is trained in a simulation environment in which the train set is collected and auto-labeled. To bridge the gap between reality and simulation, we propose a design method of transform function for mapping a real image to synthesized style. Compared with the unmapped results, the Mean Absolute Error (MAE) of the model completely trained with synthesized images is reduced by 23% and the False Negative Rate FNR (FNR) of the model fine-tuned by the real images is reduced by 42.5% after mapping. The program generation module provides a human-readable program based on the detection results to reproduce a real-world demonstration, in which a longshort memory (LSM) is designed to integrate current and historical information. The system is tested in the real world with a UR5 robot on the task of stacking colored cubes in different orders.

2.4 A STUDY OF ROBOT CONTROL PROGRAMING FOR AN INDUSTRIAL ROBOTIC ARM (2019 6TH INTERNATIONAL CONFERENCE ON ADVANCED CONTROL CIRCUITS AND SYSTEMS (ACCS) & 2019 5TH INTERNATIONAL CONFERENCE ON NEW PARADIGMS IN ELECTRONICS & INFORMATION TECHNOLOGY (PEIT))

This paper by Mahmoud Abdelaal explains about the manufacturing and production industry today is still looking for improvement of their process. Programming of articulated industrial robots is a main field for manufacturing industry improvement. This work presents a study on the operation and movement control of a 6 degrees of freedom (DOF) robotic arm. A robot movement control and programming support system is presented for industrial use. Movement control programming of the robot unit is accomplished using MELFA-BASIC V which is the actual programming language for all modern Mitsubishi industrial Robots and as an industrial robot language it is very difficult to deal with. This article describes a design of programming algorithm for a movement control taking on consideration the speed and maximum possible accuracy. The presented experiment designs and analyzes speed optimizing motions on high accuracy motions. Circular motions and motions on straight line are required in the industry. In these types of motion the accuracy of the trajectory is very important.

2.5 FUZZY — ARDUINO BASED CONTROL STRATEGY FOR HUMAN SAFETY IN INDUSTRIAL ROBOTS (2017 IEEE INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING, INFORMATICS, COMMUNICATION AND ENERGY SYSTEMS (SPICES))

This paper presented by Chaitanya S. Gajbhiye; Megha G Krishnan; S. Kumaravel; S. Ashok focuses on many industries which uses various industrial robots for their production tasks like painting, wielding etc. In the majority industries human and robots shares the same work environment which results chances of accidents between human and robots if robots are not equipped with proper protection. In this paper, a new method is proposed to avoid such collisions between human and robot in automation industries. Most of time while externally adding this safety feature in the industrial robots you have to change either its programming or internal structure. The industrial robots like ABB robots which are widely used in industrial environment don't accept such changes. So this paper uses proximity IR sensors which are applied on ABB's IRB1200 pick and place robot. A controller is demonstrated using fuzzy algorithm implemented in MATLAB Simulink and Ardiuno. Finally RAPID based program helps robot to take proper action based on controller output.

2.6 HUMAN-IN-THE-LOOP APPROACH FOR TEACHING ROBOT ASSEMBLY TASKS USING IMPEDANCE CONTROL (2015 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA))

This paper by Luka Peternel; Tadej Petrič; Jan Babič propose a system that In this paper we propose a human-in-the-loop approach for teaching robots how to solve part assembly tasks. In the proposed setup the human tutor controls the robot through a haptic interface and a hand-held impedance control interface. The impedance control interface is based on a linear spring-return potentiometer that maps the button position to the robot arm stiffness. This setup allows the tutor to modulate the robot compliance based on the given task requirements. The demonstrated motion and stiffness trajectories are encoded using Dynamical Movement Primitives and learnt using Locally Weight Regression. To validate the proposed approach we performed experiments using Kuka Light Weight Robot and HapticMaster robot. The task of the experiment was to teach the robot how to perform an assembly task involving sliding a bolt fitting inside a groove in order to mount two parts together. Different stiffness was required in different stages of the task execution to accommodate the interaction of the robot with the environment and possible human-robot cooperation.

2.7 INCREMENTALLY ASSISTED KINESTHETIC TEACHING FOR PROGRAMMING BY DEMONSTRATION (2016 11TH ACM/IEEE INTERNATIONAL CONFERENCE ON HUMAN-ROBOT INTERACTION (HRI))

This paper by Martin Tykal; Alberto Montebelli; Ville Kyrki presents a vision- Kinesthetic teaching is an established method of teaching robots new skills without requiring robotics or programming knowledge. However, the inertia and uncoordinated motions of individual joints decrease the intuitiveness and naturalness of interaction and impair the quality of the learned skill. This paper proposes a method to ease kinesthetic teaching by combining the idea of incremental learning through warping several demonstrations into a common frame with virtual tool dynamics to assist the user during teaching. In fact, during a sequence of demonstrations the stiffness of the robot under Cartesian impedance control is gradually increased, to provide stronger assistance to the user based on the demonstrations accumulated up to that moment. Therefore, the operator has the opportunity to progressively refine the task's model while the robot more docilely follows the learned action. Robot experiments and a user study performed on 25 novice users show that the proposed approach improves both usability as well as resulting skill quality.

2.8 A ROBOT MANIPULATOR COMMUNICATIONS AND CONTROL FRAMEWORK (2008 IEEE INTERNATIONAL CONFERENCE ON MECHATRONICS AND AUTOMATION)

This paper by Christian Kohrt; Anthony Pipe; Gudrun Schiedermeier; Richard Stamp; Janice Kiely proposed an use of industrial scale experimental machinery robot systems such as the Mitsubishi RV-2AJ manipulator in research to experimentally prove new theories is a great opportunity. The robot manipulator communications and control framework written in Java simplifies the use of Mitsubishi robot manipulators and provides communication between a personal computer and the robot. Connecting a personal computer leads to different communication modes each with specific properties, explained in detail. Integration of the framework for scientific use is shown in conjunction with a graphical user-interface and within Simulink as a Simulink block. An example application for assisted robot program generation is described.

2.9 LEARNING BY WATCHING: EXTRACTING REUSABLE TASK KNOWLEDGE FROM VISUAL OBSERVATION OF HUMAN PERFORMANCE (IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION (VOLUME: 10, ISSUE: 6, DEC 1994))

This paper by Y. Kuniyoshi; M. Inaba; H. Inoue propose a novel task instruction method for future intelligent robots is presented, In our method, a robot learns reusable task plans by watching a human perform assembly tasks. Functional units and working algorithms for visual recognition and analysis of human action sequences are presented. The overall system is model based and integrated at the symbolic level. Temporal segmentation of a continuous task performance into meaningful units and identification of each operation is processed in real time by concurrent recognition processes under active attention control. Dependency among assembly operations in the recognized action sequence is analyzed, which results in a hierarchical task plan describing the higher level structure of the task. In another workspace with a different initial state, the system re-instantiates and executes the task plan to accomplish an equivalent goal. The effectiveness of our method is supported by experimental results with block assembly tasks.

2.10 OBJECT DETECTION AND RECOGNITION FOR A PICK AND PLACE ROBOT (IEEE ASIA-PACIFIC WORLD CONGRESS ON COMPUTER SCIENCE AND ENGINEERING 2014)

This paper by Rahul Kumar; Sunil Lal; Sanjesh Kumar; Praneel Chand proposes a technology Controlling a Robotic arm for applications such as object sorting with the use of vision sensors would need a robust image processing algorithm to recognize and detect the target object. This paper is directed towards the development of the image processing algorithm which is a pre-requisite for the full operation of a pick and place Robotic arm intended for object sorting task For this type of task first the objects are detected, and this is accomplished by feature extraction algorithm. Next, the extracted image (parameters in compliance with the classifier) is sent to the classifier to recognize what object it is and once this is finalized, the output would be the type of the object along with it's coordinates to be ready for the Robotic Arm to execute the pick and place task The major challenge faced in developing this image processing algorithm was that upon making the test subjects in compliance with the classifier parameters, resizing of the images conceded in the loss of pixel data. Therefore, a centered image approach was taken. The accuracy of the classifier developed in this paper was 99.33% and for the feature extraction algorithm, the accuracy was 83.6443%. Finally, the overall system performance of the image processing algorithm developed after experimentation was 82.7162%.

CHAPTER III

METHODLOGY

3.1 EXISTING SYSTEM

Programming an industrial robot is no longer just about coding instructions using a low-level programming language. As the technology behind robotics continues to evolve, new programming methods meant to make it easier for end-users have emerged. Today, three main methods are used to issue instructions to modern robots, namely:

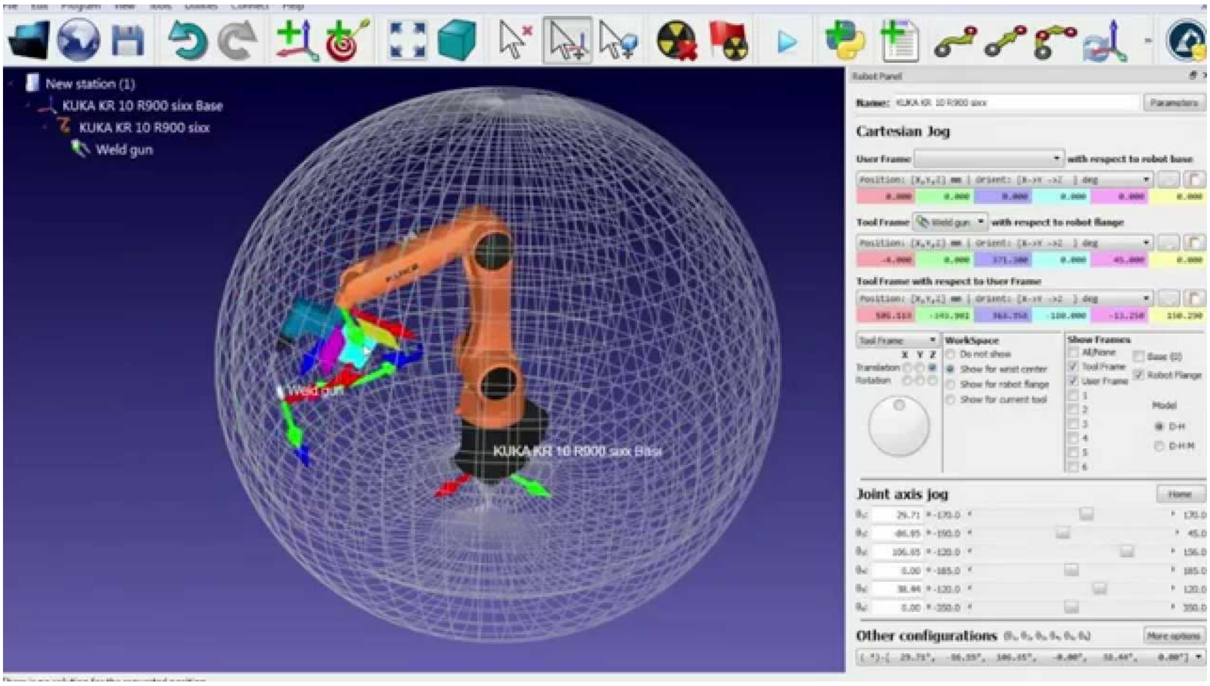
1. SIMULATION/OFFLINE PROGRAMMING

As the name suggests, this robot programming method involves writing instructions on a separate system and using virtual models of an industrial robot for testing. After the instructions have been written and tested, they are uploaded to the robot's memory. This method is preferred where a lot of instructions need to be written and tested before being deployed in real life.

Some of the advantages of offline programming include:

- Allows for more complexity
- Can lead to more efficient robot operation
- Supports precision
- Guarantees safety

Offline programming can be technical and therefore not suitable for non-engineers who do not understand low-level robotics programming or coding. All programming methods are very much in use today in most industries. The method you choose may come down to the type of robot you are programming and what you want to do with it.



Offline programming, or simulation, is most often used in robotics research to ensure that advanced control algorithms are operating correctly before moving them onto a real robot. However, it is also used in industry to reduce downtime and improve efficiency. It can be a particularly useful method for SMEs, as robots are more likely to be reconfigured multiple times than they are in mass production environments. Programming offline means that this does not interfere with production too much. Offline programming allows the robot to be programmed using a virtual mock-up of the robot and task. If the simulation software is intuitive to use, this can be a quick way to test an idea before moving it to the robot.

Some modern simulation packages are quite straightforward to use.

This video shows the RoboDK software with two robots performing pick and place from a conveyor belt. The simulation can be set up within a matter of minutes, once you are familiar with the software, using a library of common robots and objects.

Some simulators also allow you to enter a CAD part and the system will automatically generate the robot trajectories. This can improve the efficiency of programming even further.

Advantages of Offline Programming

- Reduces downtime required for robot programming. Programs are developed offline, so the robot only has to be halted while the new program is being downloaded and tested.
- Can be quite intuitive, especially if the robot can be moved around in a 3D CAD environment with drag and drop techniques.
- Easy to test many different approaches to the same problem, which would be inefficient for online programming methods.
- **Quicker robot integration:** Whether a new installation or a redeployment, time is always an essential factor. By creating, adjusting and testing the programming in a virtual environment, when the robot is installed, it is ready to operate, saving days or weeks compared to a teach pendant.
- **No downtime:** With teach pendant programming, the robot must be in teach mode. OLP can upload the programming to a robot while it is operating, which means less manufacturing delays.
- **Safety:** Whether faced with a safety concern due to robot configuration, or restricted access to the robot due to work from home protocols, OLP can be completed from a laptop in a safe location, and uploaded to the robot without the need to access the teach pendant. In addition, OLP allows for pre-engineering of the robot cell, providing the ability to implement the entire work cell, with fencing, external axis and other components to avoid delays at the time of installation.
- **Quick changeovers:** OLP can program a robot for Part B, while it is completing the Part A run. As soon as Part A is complete, the Part B

programming can be uploaded and the task begun, without turning off the robot. OLP allows for quicker cycle times, allowing for a higher mix of jobs with faster transitions.

- **Universal application:** Few manufacturing operations ascribe to brand loyalty, operating only one brand of robot or other tools. For most, the robot that accomplishes the task at the best price is the one that is installed, which leads to a variety of robots that need to be programmed. True OLP software should be “robot agnostic”: it has the ability to communicate across many platforms. Even if you are redeploying an older robot, OLP will interface with it.
- **Test new configurations:** Adding a robot into your manufacturing operation is a significant capital investment. Robotmaster’s intuitive user interface allows the user to design and test various configurations in 3D simulation before the robot is installed. You can engineer, identify, and solve issues before the robot arrives.
- **Ease of use:** The right OLP requires process knowledge, however, it does not require advanced programming or robotics expertise. OLP is built around the user and is designed to be a clean and elegant user experience.

Disadvantages of Offline Programming

While offline robot programming offers many benefits, including improved welding productivity and quality, limitations exist that industries and operators should be aware of when determining their offline programming software.

- Skilled Labor
- Manufacturers that utilize offline robot programming must employ programmers to make the best use of the technology. With a little bit of training, anyone can

have automated processes within their facility. The investment in technology and labor may ultimately seem insignificant compared to the increases in operational efficiency and productivity. By mimicking the real-world operating environment, offline robot programming reduces material waste, production downtime, and collision risk while strengthening the manufacturer's bottom line.

- Offline Robot Programming Software for Increased ROI
- After weighing the pros and cons of offline robot programming, it is safe to say that the manufacturers benefit considerably from the software's ability to improve their operational efficiency. For many years, robotic welding has enabled industries to adapt to various welding applications and requirements while adhering to industry standards. With offline robot programming, industries can make the optimum use of the welding robots and equipment to secure maximum returns from their investment.
- Virtual models will (probably) never be able to represent the real world with 100% accuracy. Programs may still need to be altered after they are applied to the real robot.
- Might take longer overall. Although offline programming reduces the downtime of the robot, it means that someone has to spend extra time developing the simulation, as well as testing it on the robot.
- Can sometimes end up wasting time sorting out simulator issues instead of solving production challenges. This could be related to the quality of the simulator.

2. TEACHING PENDANT / DRIVE THROUGH

A teach pendant device is needed to control an industrial robot remotely. The device allows its controller to work with robots without the need for tethering to a fixed terminal. Teach pendants offer a variety of settings to control robots and are also utilized to design new capabilities and features. Within the robotics repair industry, technicians not only repair the units themselves but use the device to test robotic equipment. The teach pendant is an essential component for industrial robots and utilized for application use, along with the repair and refurbishment process.

The most popular method of robot programming is probably the teach pendant. According to the British Automation and Robot Association, over 90% of robots are programmed using this method. The robot teaching pendant has changed a lot throughout its lifetime, but often consists of, what looks like, a giant handheld calculator. Early pendants were large, grey boxes with magnetic tape storage. The modern teach pendants are more like a touchscreen tablet, as the technology has developed to suit the ever evolving users. To program the robot, the operator moves it from point-to-point, using the buttons on the pendant to move it around and save each position individually. When the whole program has been learned, the robot can play back the points at full speed.

The “classic” option for robot programming is the teach pendant. This is a small console that comes packaged with the robot from the manufacturer.

Programming is usually done with the brand-specific programming language (e.g. RAPID for ABB robots, JBI for Motoman robots, etc). However, some robot brands (UR for example) do have a graphical user interface on their teach pendants.

Industrial robots are sophisticated machines capable of handling numerous manufacturing processes, even ones once thought only possible through human labor. In order for a robot to be capable of performing an application it first must be programmed. Robotic teach pendants are the most common device use to program industrial robots. Teach pendants are a vital component of a robotic system. Without a teach pendant a FANUC R-2000ib/165F would not be able to weld car frames, while the FANUC LR Mate 200id could not assemble computer keyboards, as they would lack information needed to complete each process.

Robotic teach pendants are handheld devices that may be wired or wireless. Pendants are typically included with the control system when purchasing an industrial robot. They may contain several buttons or switches or feature a touchscreen display as is the case with newer models. They also feature a display which shows the robot’s commands and allows for editing of those commands. In addition, the display can be used to recall the command history of the robot. Pendants utilize a keyboard for task input and easy programming. Another common feature of pendants is a large red button, which is the emergency stop button of the robotic system. It can be utilized should the industrial robot malfunction and will immediately stop operation once activated, ensuring the safety of any workers or manufacturing equipment around the robot.

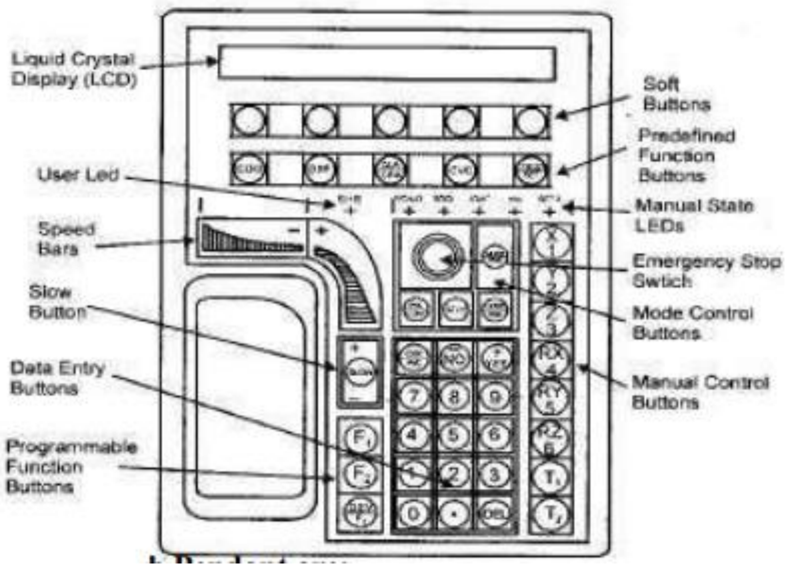
Robotic teach pendants provide a robot operator the ability to program applications and to control the robot’s motion remotely. Operators can program robots through pendants without needing to be connected to a fixed terminal. The remote capability allows operators to program or control robots safely out of reach of their workspace or hazardous environments. In addition to programming and controlling robots, pendants can be used for testing and troubleshooting robotic systems. The ability to test robot systems allows for easier integration. While troubleshooting is simplified by being able to recall the robot’s command history and functions to quickly identify an issue.

Industrial robots are best suited for performing repetitive tasks, making teach pendants ideal programming solutions as they are designed for teach and repeat programming. Through this technique operators program robots for specific application parameters, these may include the robot speed and range of motion. The controls on the pendant allow an operator to relay information to the robot about cycle times, velocity, functionality, and interactions needed with any additional machinery involved. For example, a teach pendant can be utilized to program a FANUC M-710ic/50 for a die-casting application. The pendant will relay information about each step needed in the application process including loading the die-cast machine with the mold, injecting the mold, and then removing the cast from the machine and mold once cooled. The use of a robot pendant allows the operator to safely program the application remotely, being spared of the harsh foundry environment. Teach pendants also simplify programming of a multi-step manufacturing process through inputting information with their keyboard instead of manually manipulating the robot or coding through a computer.

TEACH PENDANT FOR ROBOT SYSTEM

The teach pendant has the following primary functions:

Serve as the primary point of control for initiating and monitoring operations. Guide the robot or motion device, while teaching locations. Support application programs.The Teach Pendant is used with a robot or motion device primarily to teach. Robot locations for use: in application programs.The Teach Pendant is also used with c routine's that pause execution at specified points and allow an Operator to teach * re-teach the robot locations used by the program.There are two styles of Teach Pendants: the programmer‘s pendant, for use while an application is being pendant, which is designed for use during normal system operation.



Whenever this switch is released, arm power is removed from the motion device.To operate the Teach Pendant left hand is put through the opening on the left-hand side of the pendant and the left thumb is used to operate the pendant speed bars. The right hand is used for all the other function buttons.

The major areas of the Teach Pendant are:

1. Data Entry Buttons:

The data entry buttons are used to input data, normally in response to prompts that appear on the pendant display The data entry buttons include YES/NO, DEL, the numeric buttons, the decimal point and the REC/DONE button, which behaves like the Return or Enter key on a normal keyboard. In many cases, application programs have users press the REC/DONE button to signal that they have completed a task.

2. Emergency Stop Switch:

The emergency stop switch on the Teach Pendant immediately halts program execution and turns off arm power.

3. User LED:

The pendant is in background mode when the user LED is in not lit and none of the predefined functions are being used. The user LED is lit whenever an application program is making use of the Teach Pendant.

4. Mode Control Buttons:

The mode control buttons change the state being used to move the robot, switch control between the Teach Pendant and the application programs and enable arm power when necessary.

5. Manual Control Buttons:

When the Teach Pendant is in manual mode, these buttons select which robot joint will move, or the coordinate axis along which the robot will move.

6. Manual State LEDs:

The manual state LEDs indicates the type of manual motion that has been selected.

7. Speed Bars:

The speed bars are used totion. Pressing control the speed barthe rob near the outer ends will move the robot faster, while pressing the speed bar near the center will move the robot slower.

8. Slow Button:

The slow button selects between the two different speed ranges of the speed bars.

9. Predefined Function Buttons:

The predefined function buttons have specific, system- wide functions assigned to them, like display of coordinates, clear error, etc.

10. Programmable Function Buttons:

The programmable function buttons are used in custom application programs, and their functions will vary depending upon the program being run.

11. Soft Buttons:

The ―soft‖ buttons have different function or the selection made from the predefined function buttons.

ADVANTAGES OF A TEACHING PENDANT

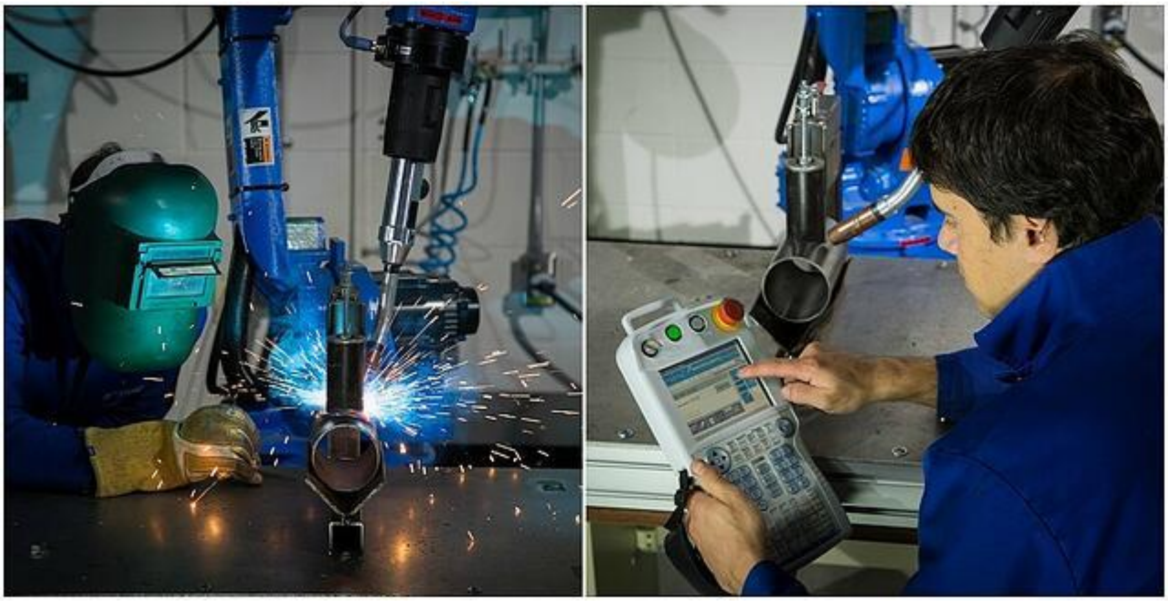
- Most traditional industrial robots come with a teach pendant, which makes them familiar to technicians.
- They allow precise positioning, as the robot can be programmed using numerical coordinates, in either world coordinates, robot coordinates or another coordinate system.
- Teach pendants are great for simple movements, such as painting in a straight line or over a large flat surface. The teach pendant is right next to the robot so it is very handy.
- It comes packaged with the robot so no extra hardware is required.
- As the software is developed by the manufacturer, it will make use of the robot's more "obscure" functionality.

DISADVANTAGES OF A TEACHING PENDANT

- Disruptive to the whole system due to robot downtime. The robot must be put into "teach mode" and all operations using the robot halted until it has been programmed.
- Requires training to learn and program.
- Might be difficult for skilled craftspeople who are unfamiliar with programming. As it is an online programming method, it increases downtime as the robot must be stopped for programming.
- Programmers must learn a completely different programming language for each robot brand.
- Requires more training and skilled robotics knowledge than more general-purpose, intuitive methods.

3. TEACHING BY DEMONSTRATION / LEAD THROUGH

Teaching by demonstration (and more specific methods like Kinetic teaching) offers an intuitive addition to the classic teach pendant. These methods **involve moving the robot around, either by manipulation a force sensor or a joystick attached to the robot wrist just above the end effector.** As with the teach pendant, the operator stores each position in the robot computer. Many collaborative robots have incorporated this programming method into their robots, as it is easy for operators to get started immediately using the robot with their applications.



Programming is vital to the success of any robotic system as it provides the set of instructions needed for an industrial robot to interact with its environment in order to perform specific tasks. Before a FANUC Arcmate 120ic can weld workpieces together, it must be programmed with the specific commands to be able to accurately complete the application. There are several programming methods used for industrial robots. The most common method is through teach pendants followed by offline programming. Another method that can be used for robotic programming is the lead through method.

The lead through method involves programming industrial robots through demonstration. This method is also referred to as hand guidance programming or the walk-through method. During programming a robot operator will physically move the robotic manipulator through the waypoints of a desired task. Some industrial robots have a joystick attached to their wrist above the EOAT which can also be used to move the robotic manipulator for lead through programming.

This programming method is best for robotic applications involving a continuous path such as welding automation or painting. Lead through programming has declined in use when it comes to traditional industrial robots. The size and weight of the robotic manipulators makes it difficult for robot operators to physically move them through an application path. However, lead through programming has started to make a comeback as many collaborative robots have incorporated the method as their main programming source. The lighter weight and smaller size of cobots makes it easier for operators to physically manipulate the robot arm. FANUC's Cr-15ia is a cobot featuring hand guidance programming. Universal’s cobots also feature this programming method, including their Universal UR10.

LEAD THROUGH PROGRAMMING METHODS

In lead through programming, the robot is moved through the desired motion path in order to record the path into the controller memory.
There are two ways of accomplishing lead through programming:

1. Powered lead through
2. Manual lead through

The powered lead through method makes use of a teach pendant to control the various joint motors, and to power drive the robot arm and wrist through a series of points in space.

Each point is recorded into memory for subsequent play back during the work cycle. The teach pendant is usually a small handheld control box with combinations of toggle switches, dials, and buttons to regulate the robot's physical movements and programming capabilities.

Among the various robot programming methods, the powered lead through method is probably the most common today.

It is largely limited to point-to-point motions rather than continuous movement because of the difficulty in using the teach pendant to regulate complex geometric motions in space.

A large number of industrial robot applications consist of point-to-point movements of the manipulator. These include part transfer tasks, machine loading and unloading, and spot welding.

The manual lead through method (also sometimes called the 'walkthrough' method) is more readily used for continuous-path programming where the motion cycle involves smooth complex curvilinear movements of the robot arm.

The most common example of this kind of robot application is spray painting, in which the robot's wrist, with the spray painting gun attached as the end effector, must execute a smooth, regular motion pattern in order to apply the paint evenly over the entire surface to be coated.

Continuous arc welding is another example in which continuous path programming is required and this is sometimes accomplished with the manual lead through method.

In the manual lead through method, the programmer physically grasps the robot arm (and end effector) and manually moves it through the desired motion cycle.

If the robot is large and awkward to physically move, a special programming apparatus is often substituted for the actual robot.

This apparatus has basically the same geometry as the robot, but it is easier to manipulate during programming.

A teach button is often located near the wrist of the robot (or the special programming apparatus) which is depressed during those movements of the manipulator that will become part of the programmed cycle.

This allows the programmer the ability to make extraneous moves of the arm without their being included in the final program.

The motion cycle is divided into hundreds or even thousands of individual closely spaced points along the path and these points are recorded into the controller memory.

The control systems for both lead through procedures operate in either of two modes: teach mode or run mode.

The teach mode is used to program the robot and the run mode is used to execute the program.

The two lead through methods are relatively simple procedures that have been developed and enhanced over the last 20 years to teach robots to perform simple, repetitive operations in factory environments.

The skill requirements of the programmers are relatively modest and these procedures can be readily applied in the plant.

ADVANTAGES

The main advantage of lead through programming is it simplifies the robotic programming process. Anyone regardless of robotic experience can program using this method. It is considered to be faster than teach pendant programming since it eliminates having to key-in instructions. Users can quickly setup, program, and implement their articulated robot or cobot. Lead through programming is also more intuitive than other robotic programming methods. Since the operator physically moves the robotic arm, applications are demonstrated in the same manner as how a human would perform the task.

- Quicker than traditional teach pendants. It removes the need for multiple button pressing, allowing the operator to simply move the robot to the desired position.
- More intuitive than both traditional teach pendants and simulation programs, as the task is programmed in almost the same way a human operator would perform it. This makes it simple for operators to learn. Generally, this method requires no knowledge of programming concepts or being familiar with 3D CAD environments (as simulation does).
- Very good for detailed tasks which would require many lines of code to achieve the same effect, such as welding or painting of intricate shapes.

DISADVANTAGES

While lead through programming is simple, fast, and easy to learn, it does have some drawbacks. Lead through programming is not suitable for all robot types. As mentioned earlier, traditional six axis robots are moving away from this programming method as their size and weight makes it difficult to physically demonstrate applications. Another disadvantage is this programming method does involve downtime since the actual robot is involved. This means robots must be taken out of production to be programmed which could cause manufacturing delays. This programming method is also difficult for applications requiring precise coordinates or multiple movements in different directions. Moving the robot manually can cause inaccuracies, especially if movements need to be precise. For such applications it is best if coordinates can be keyed-in numerically to ensure the most precise robot operation.

- As with traditional a teach pendant, this method uses the physical robot for programming. This means that it does not reduce downtime, as much as offline programming.
- Moving the robot to precise coordinates is not as straightforward as with the other methods. This is especially true with some joystick based systems, where there is no way of entering a numerical value. Kinetic teaching combines these features by allowing for the entering of exact numerical coordinates along with positioning based coordinates.
- Not so good for tasks which are "algorithmic" in nature. For example, if a robot had to paint a flat surface by moving horizontally along the surface, then move down an inch, move horizontally in the opposite direction, etc. Moving the robot by hand would be arduous and inaccurate for such a task.
- One of the main disadvantages of programming with teach pendants is it can be time consuming, especially when programming complex applications. The more commands an application has the longer programming will take since each step must be manually entered. Another disadvantage of teach pendants is robots must be taken out of production during programming, causing downtime for operations.

- Disadvantages of lead through programming include lack of versatility, precision, and downtime. Lead through programming is mostly limited to robots as most traditional industrial robots no longer support this method. It is also difficult for applications requiring precise coordinates as programming by hand can cause inconsistencies and be inaccurate. Like teach pendants, lead through programming also involves downtime since the robot must be taken out of operation and placed in teach mode for programming.

3.2 NEW SYSTEM : ADVANCED LEAD THROUGH ROBOT - ALGORITHM

1. The user controls the small model for the job he planned to execute.
2. There is a serial communication between the small model and the real time robot. The setup gets the input destination from the user via speechthrough a microphone.
3. Whenever the small model moves, the position is determined by the micro controller through map function.
4. The angles determined from the small model is feed back to the servo motors to operate it live.
5. In order to save the positions of the robot, the save button is pressed so that the current position of all the potentiometers is saved.
6. The saved positions are a point functions where the servo angles are travel to each points. Not like path function, it doesn't follow the irregular trajectory.
7. After the process of saving, the RAM inside the micro controller which carries the angular data can be retrieved at any time with ease of use.
8. When we need to play the robot motion, we need to press the play button so that the saved points are given to servo motors of individual joints to attain the duplicative positions of the small model.
9. This loop will continue forever until there is a power supply. When we want to reset the program, we have to press the reset button to reset the entire program so we can add new programs to work with.

10. We can stop in case of emergency if we press the emergency button. This will reboot the entire robot to stop the motion. Thus the Robot programing is simplified and the objective is achieved.

3.3 BLOCK DIAGRAM

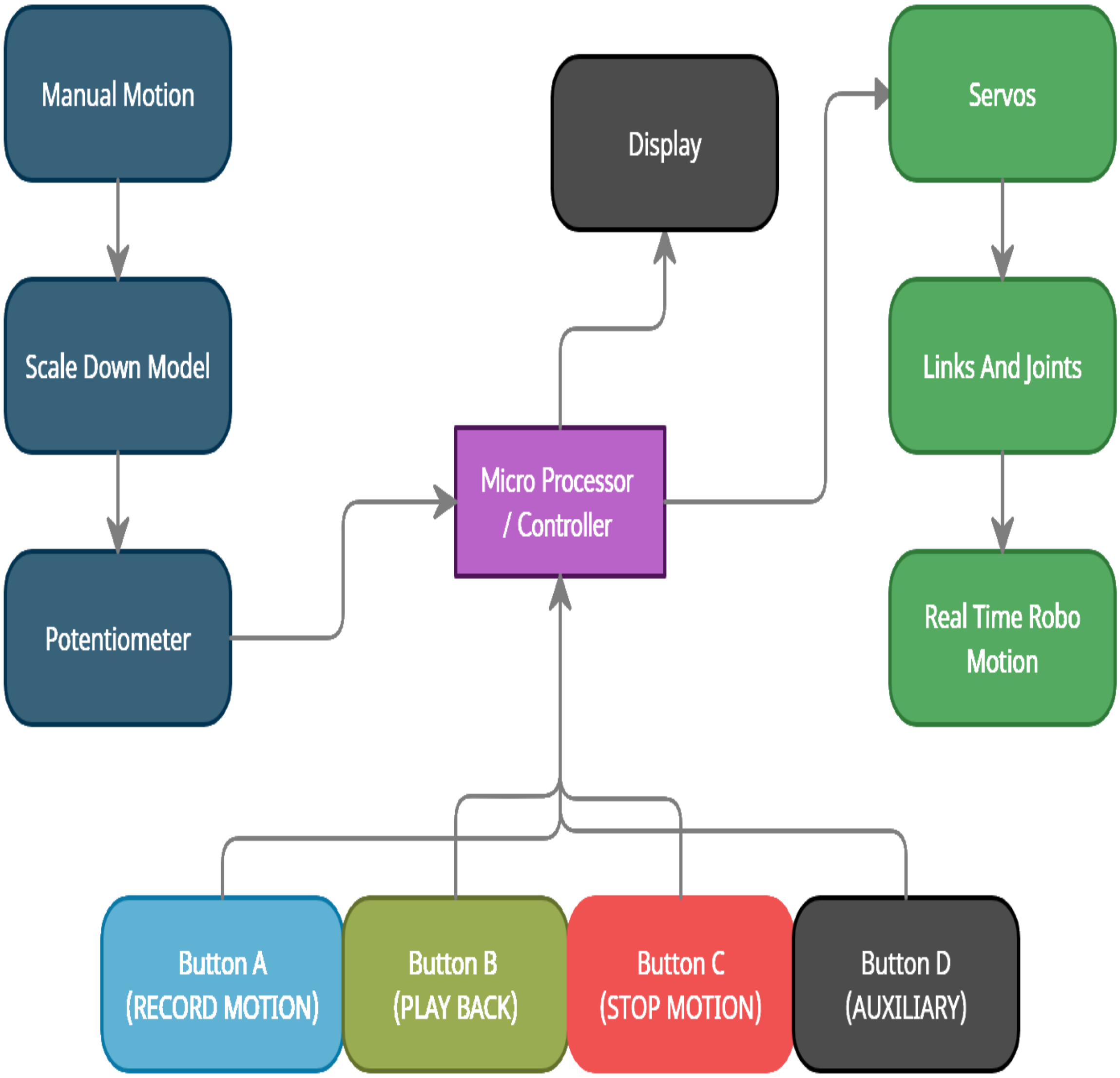


Fig 3.1: Block diagram of the proposed system

3.4 DETAILED WORKING & DESIGN

Step-1: We going to move the scale down model.

Step-2: The potentiometer is attached to the model will gives the output voltage with reference to the power supply and the angle of the scale down model.

Step-3: These output voltages from each joints of the scale down model are then converted to angles through micro controller.

Step-4: In the real-time robot, we will have servo motors for angular twist of the arm. So we will control the servo through the angles obtained from the microcontroller with data from the potentiometer of the scale down model.

So through programing, we can lively control the real-time robot with a small scale down model.

Later, when we need to record some required motion, we have switches for saving the angles, replay the position and to reset the program memory.

Through display device like computer, we can lively monitor the actions performed by the robot with clear data that can be saved and documented for future references.

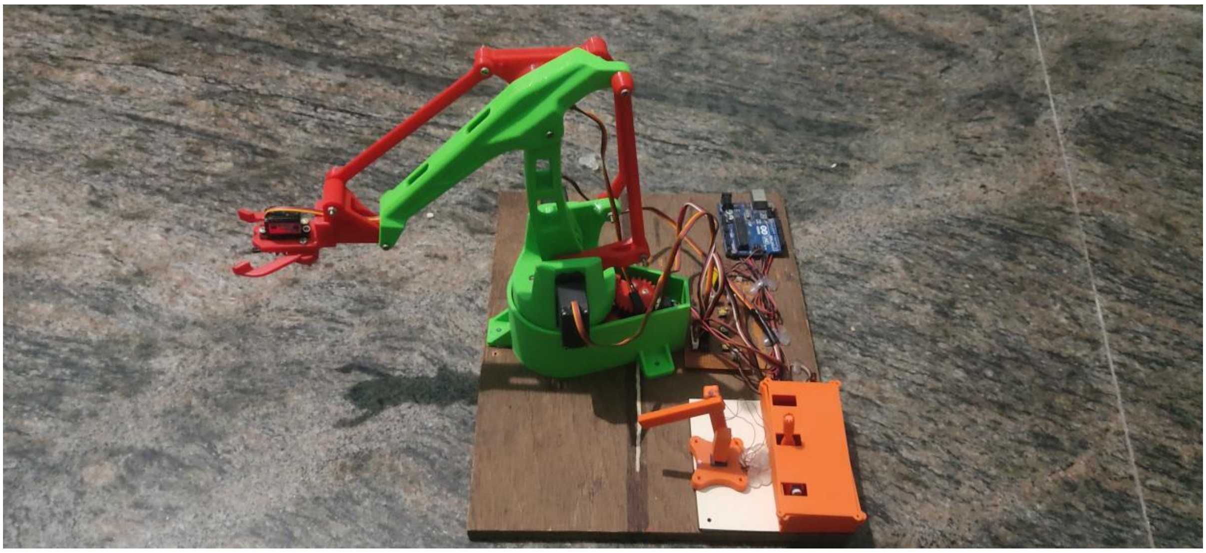
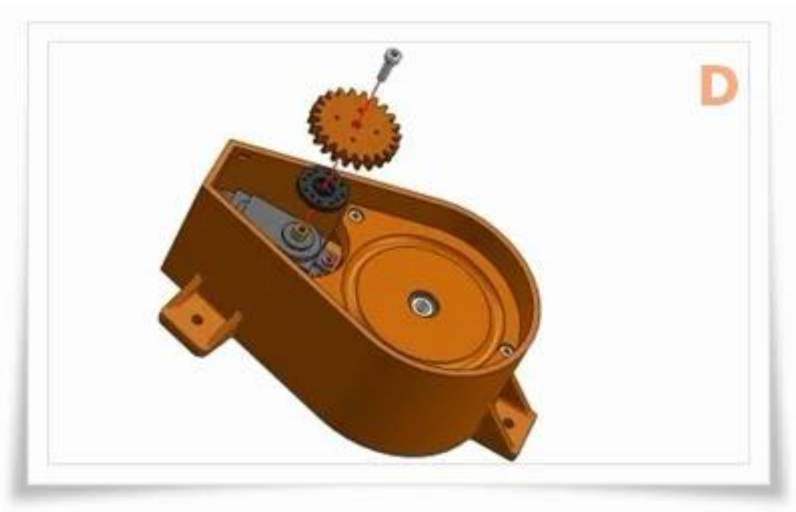


Fig 3.2: Design of the project

A - Put in position an MG946 servo with the driving shaft aligned forward. Fix the servo to the main base using the self-tapping screws supplied with it.

B - Insert 3 M3 nuts in the receptacles of the main base as shown.

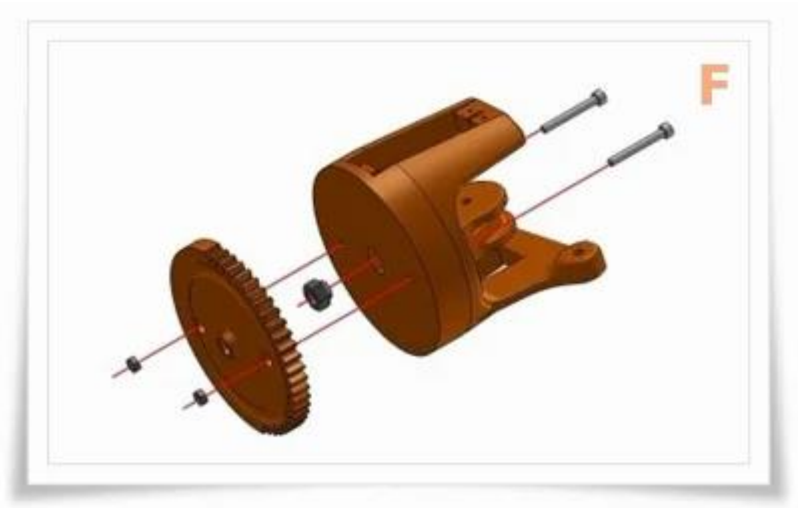
C - Insert the 606 bearings in its housing and attach the plate to the main base using 3 M3 screws. Verify the freedom of movement of the bearing.



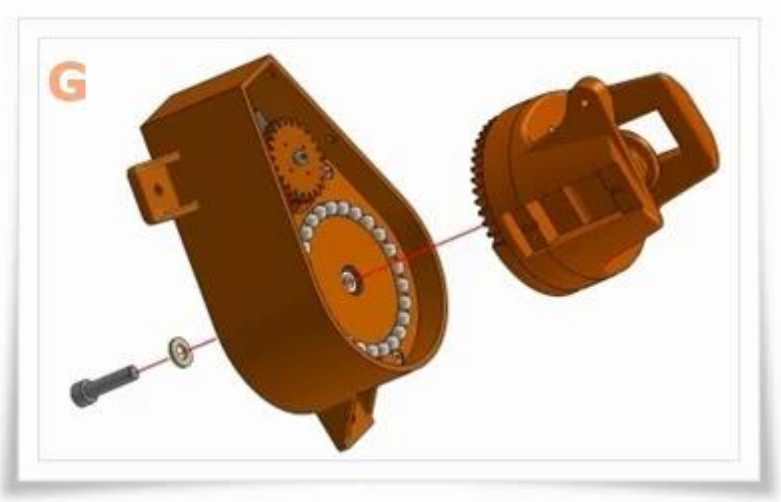
D - Position the drive plate on the splined shaft and upper the driving printed gear. Add one or two small self-tapping screw to connect plate and the gear. (There are two driving gears available one has 22 teeth and the other 25. I

Made two because during printing of the base I've got some deformation and the two axis distance became smaller)

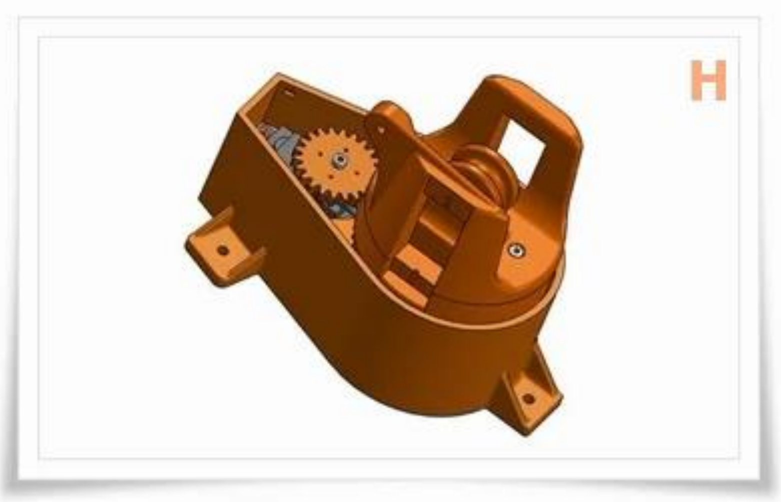
E - Fill the path using about 25 spheres with a diameter of 6mm.



F - Insert an M6 self-locking nut in the receptacle of the swivel base then place in position le geared base and fix it using a couple of M3 screws and nuts as shown.

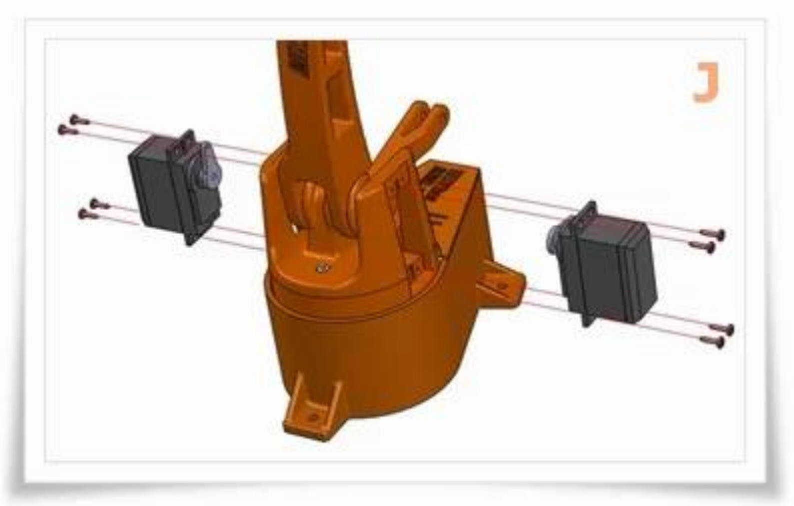


G - Keeping the main base flat and the swivel element in contact with it, connect the two elements using an M6 screw.

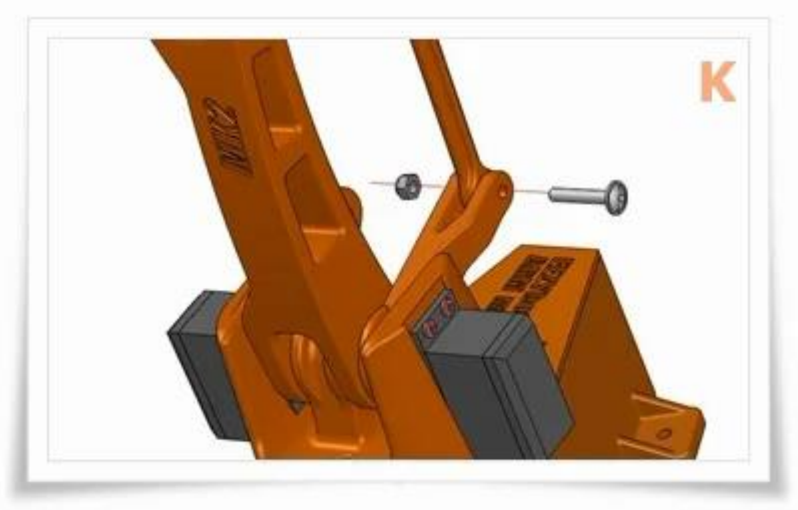


H - Now the main base is finished.

I - Put in position the main arm and the vertical drive lever, connect them with the main base horizontal axis using a 4mm dia rod 33 mm long.



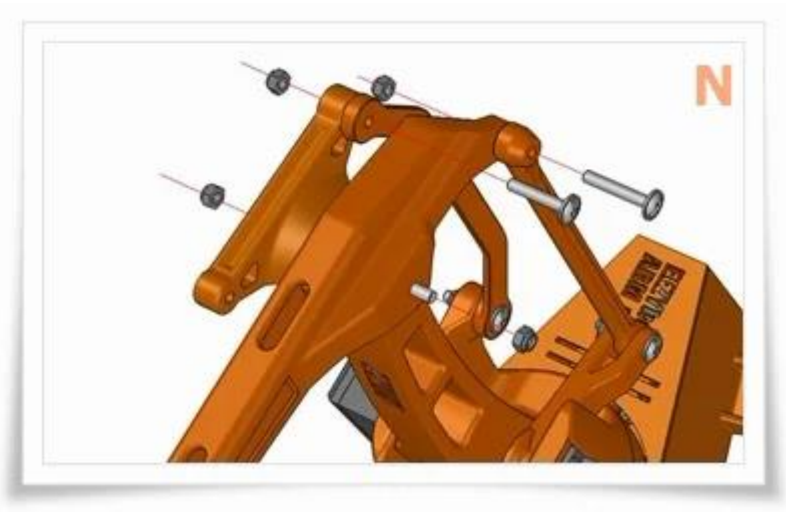
J - Fix in position the two servo and hold in place using eight selftapping screws. To drive the arms use the sigle horns supplied with the servos. Make sure that the mid position of the servos are aligned with the housing of the arms



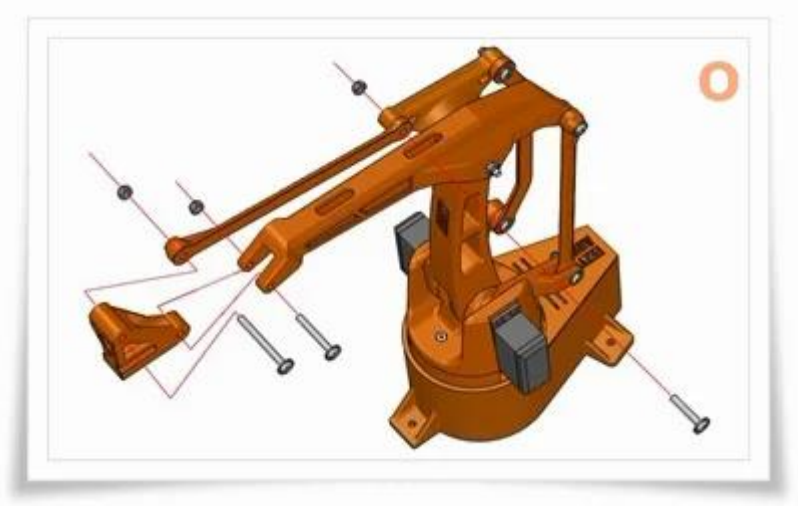
K - Connect the lower end of straight lever to the driving arm.

L - Connect the lower end of angled lever to the fixed end on the base.

M - Use a threated M4 rod to connect the horizontal arm and the triangle to the upper part of the main arm.



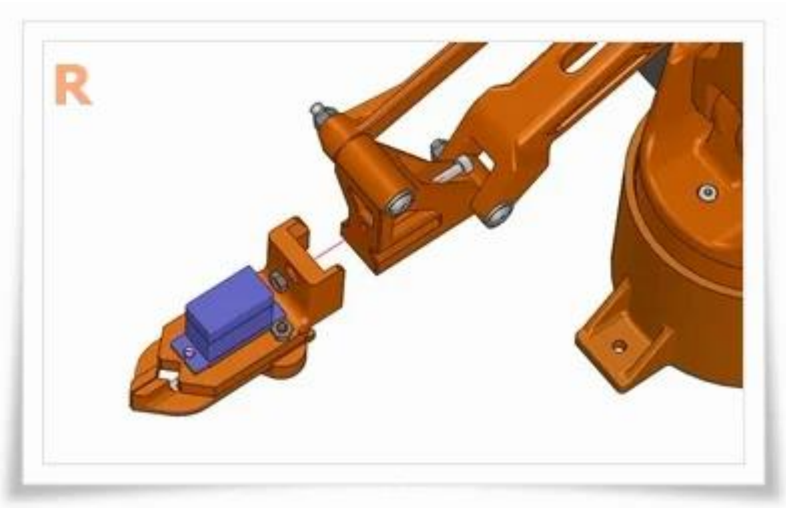
N - Connect the straight rod to the main arm and angled to the triangle.



O - Attach the rod and the fast release of the gripper, to the front part of the horizontal arm.

P - The robotic arm is now assembled. You can now proceed with the gripper assembly or you can use your own gripper design

Q - Assembly the gripper as shown in the image "Q".



R - Attach the gripper to the fast release at the end of the arm and make the servo wire pass in the internal space of the horizontal arm.

3.5 INTERCONNECTION OF COMPONENTS:

The five potentiometers are connected from the 5-analog pins A1-A5. The five servos are connected to 5-digital with PWM output pins and the four switches are connected to the digital pins 8, 12 & 13.

The display of results are obtained through the serial monitor through Type-B data cable to display monitor. The board is powered with 7.4V DC made of Li-ion cells in arrangement 2S3P. As each cells are with specification 3.7V, 2.2AH. So the 2S3P will gives the output of 7.4V 6.6AH

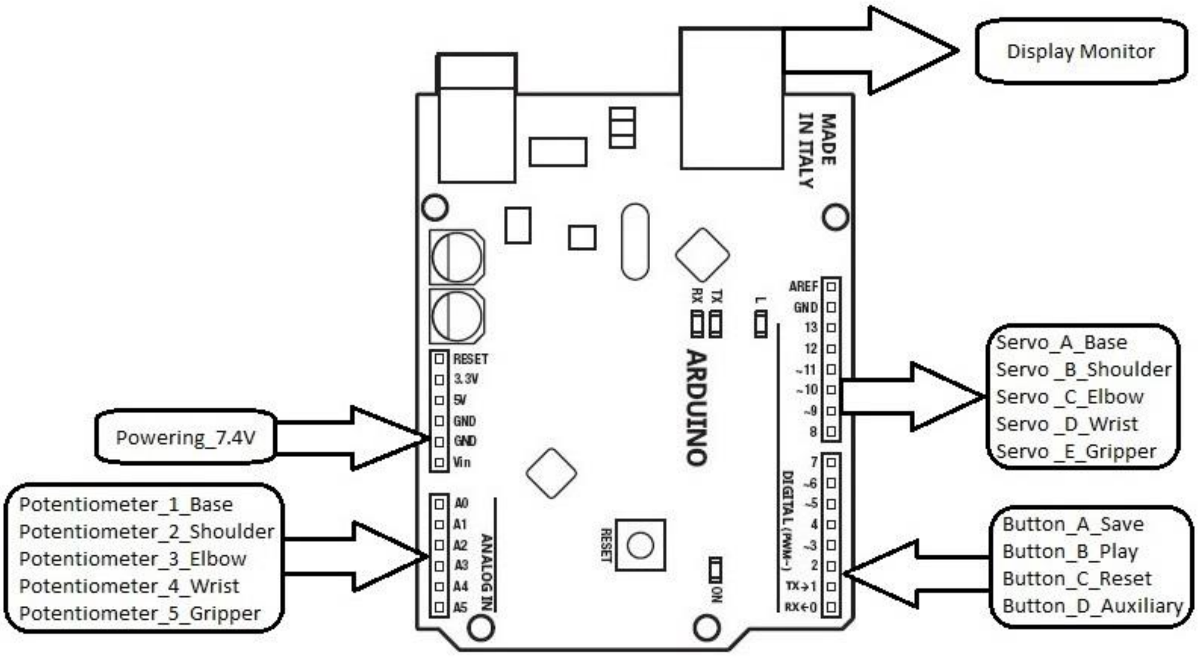


Fig 3.3: Interconnection diagram

CHAPTER IV
HARDWARE

4.1 ARDUINO UNO:

An open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external supply.

4.1.1 FEATURES:

- The operating voltage is 5V
- The recommended input voltage will range from 7v to 12V
- Digital input/output pins are 14
- Analog i/p pins are 6
- DC Current for each input/output pin is 40 mA
- DC Current for 3.3V Pin is 50 mA
- Flash Memory is 32 KB
- SRAM is 2 KB
- EEPROM is 1 KB
- CLK Speed is 16

Fig 4.1 Arduino UNO

4.2 HIGH TORQUE SERVO MOTOR:

A rotary 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. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor, although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system. The servo circuitry is built right inside the motor unit and has a position able shaft, which usually is fitted with a gear. The motor is controlled with an electric signal which determines the amount of movement of the shaft. Inside there is a set-up: a small DC motor, potentiometer, and a control circuit. The motor is attached by gears to the control wheel.

As the motor rotates, the potentiometer's resistance changes, so the control circuit can precisely regulate how much movement there is and in which direction. When the shaft of the motor is at the desired position, power supplied to the motor is stopped. If not, the motor is turned in the appropriate direction.

The desired position is sent via electrical pulses through the signal wire. The motor's speed is proportional to the difference between its actual position and desired position. So if the motor is near the desired position, it will turn slowly, otherwise it will turn fast. This is called proportional control. This means the motor will only run as hard as necessary to accomplish the task at hand, a very efficient little guy.

4.2.1 APPLICATIONS:

- position control surfaces
- operating grippers
- In food services and pharmaceuticals
- in-line manufacturing
- Automatic doors
- Printing presses
- Solar array and antenna positioning
- Cameras

4.2.2 FEATURES:

- Weight: 55g
- Dimension: 40.7×19.7×42.9mm
- Operating Voltage: 4.8-6.6 V

- Stall torque @4.8V : 13kg-cm
- Stall torque @6.6V : 15kg-cm
- Operating speed @ 4.8V : 0.19sec/60degree
- Operating speed @ 6.6V : 0.15sec/60degree
- Angle of rotation: 180 degrees
- Temperature Range: 0- 55º
- Servo wire length: 32cm
- Pulse cycle: 1 mS
- Gear Type: Metal gear
- Power Supply: Through External

4.2.3 COMPARISON TABLE:

MG995 Metal Gear Servo Motor Vs MG90S Metal Gear Servo Motor

SPECIFICATION	**MG995 Metal Gear Servo Motor**	**MG90S Metal Gear Servo Motor**
Weight	55 gm	13.4g
Dimension	40.7×19.7×42.9mm	22.8 × 12.2 × 28.5 mm
Operating Speed (4.8V no load)	0.19sec/60degree	15sec / 60 deg
Operating Speed (6.6V no load)	0.15sec/60degree	13sec / 60 deg
Stall torque @4.8V	13kg-cm	1.8kg-cm
Stall torque @ 6.6V	15kg-cm	2.2kg-cm

Dead band width	5µs	5µs
Temperature range	0- 55º	0- 55º
Operation Voltage	4.8-6.6 V	4.8-6.6 V

Fig.4.2: High Torque Servo Motor

4.3 LM7805 VOLTAGE REGULATOR IC:

Voltage regulators are very common in electronic circuits. They provide a constant output voltage for a varied input voltage. In our case the 7805 IC is an iconic regulator IC that finds its application in most of the projects. The name 7805 signifies two meaning, “78” means that it is a positive voltage regulator and “05” means that it provides 5V as output. So our 7805 will provide a +5V output voltage.

The output current of this IC can go up to 1.5A. But, the IC suffers from heavy heat loss hence a Heat sink is recommended for projects that consume more current. For example if the input voltage is 12V and you are consuming 1A, then (12-5) * 1 = 7W. This 7 Watts will be dissipated as heat.

4.3.1 PIN DIAGRAM:

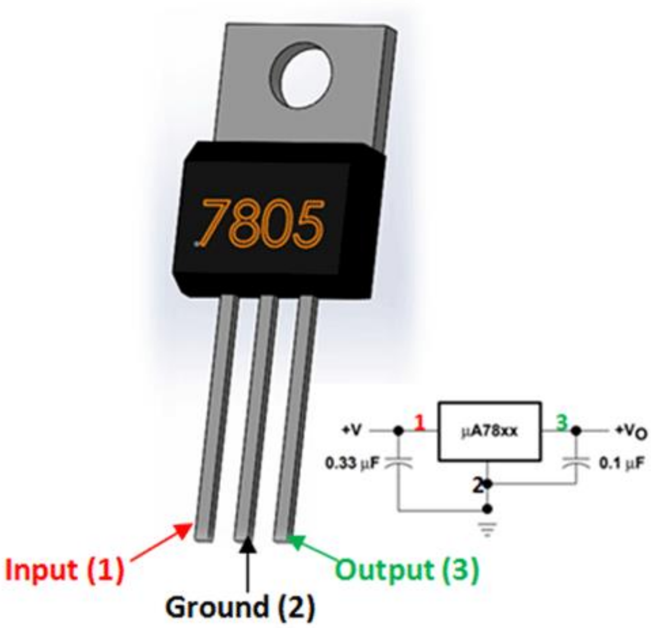


Fig. 4.3 Pin diagram of LM7805 IC

s, the 7805 Voltage Regulator IC. A regulated power supply is very much essential for several electronic devices due to the semiconductor material employed in them have a fixed rate of current as well as voltage. The device may get damaged if there is any deviation from the fixed rate.

One of the important sources of DC Supply are Batteries. But using batteries in sensitive electronic circuits is not a good idea as batteries eventually drain out and loose their potential over time. Also, the voltage provided by batteries are typically 1.2V, 3.7V, 9V and 12V. This is good for circuits whose voltage requirements are in that range. But, most of the TTL IC's work on 5V logic and hence we need a mechanism to provide a consistent 5V Supply. Here comes the 7805 Voltage Regulator IC to the rescue. It is an IC in the 78XX family of linear voltage regulators that produce a regulated 5V as output. 7805 is a three terminal linear voltage regulator IC with a fixed output voltage of 5V which is useful in a wide range of applications. Currently, the 7805 Voltage Regulator IC is manufactured by Texas Instruments, ON Semiconductor, STMicroelectronics, Diodes incorporated, Infineon Technologies, etc.

4.4 ROTARY POTENTIOMETER:

A three-terminal resistor with a rotating contact that forms an adjustable voltage divider. Potentiometers consist of a resistive element, a sliding contact (wiper) that moves along the element, If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat. The measuring instrument called a potentiometer is essentially a voltage divider used for measuring electric potential (voltage). Potentiometers operated by a mechanism can be used as position transducers.

A potentiometer is a simple knob that provides a variable resistance, which we can read into the Arduino board as an analog value. In this example, that value controls the rate at which an LED blinks. e connect three wires to the Arduino board. The first goes to ground from one of the outer pins of the potentiometer. The second goes from 5 volts to the other outer pin of the potentiometer. The third goes from analog input 2 to the middle pin of the potentiometer.

By turning the shaft of the potentiometer, we change the amount of resistance on either side of the wiper which is connected to the center pin of the potentiometer. This changes the relative "closeness" of that pin to 5 volts and ground, giving us a different analog input. When the shaft is turned all the way in one direction, there are 0 volts going to the pin, and we read 0. When the shaft is turned all the way in the other direction, there are 5 volts going to the pin and we read 1023. In between, analogRead() returns a number between 0 and 1023 that is proportional to the amount of voltage being applied to the pin.

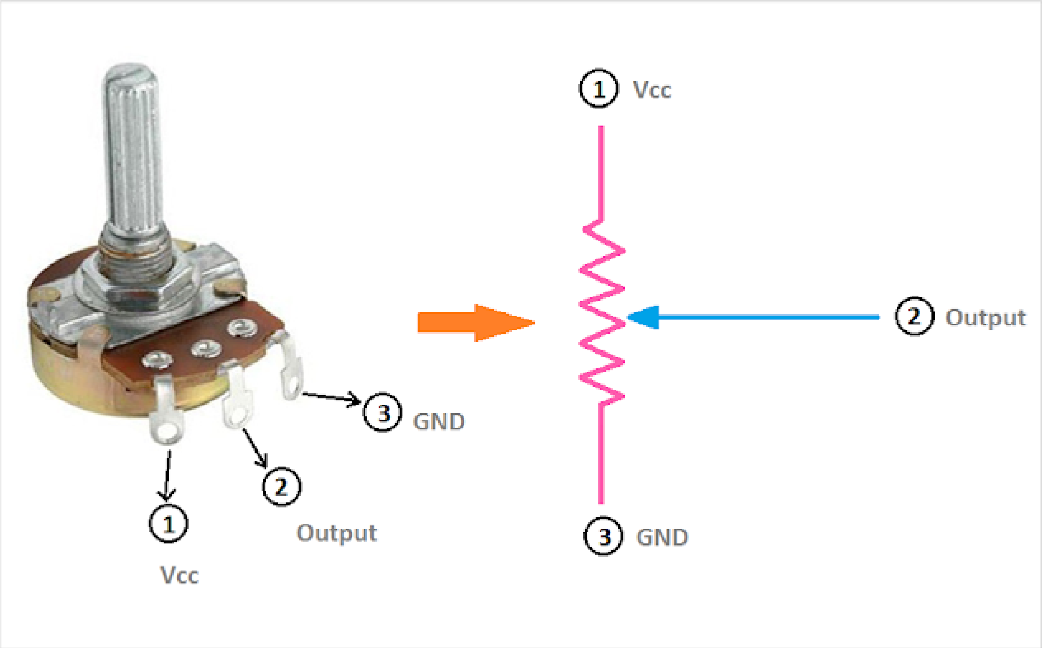


Fig.4.4: Rotary Potentiometer

4.5 USB TO TTL CONVERTER:

A TTL-USB converter is essentially required for the direct interfacing of modules to the PC, without an intermediate microcontroller or similar platform. We generally use TX and RX pins for communication. We using a Type-2 RS232 TTL signal converter (Tx, Rx, +5V, Gnd). It is a serial with 2-pin connectors for mounting on a microcontroller board. Where, 2-pin connector for power and another 2-pin connector for data. USB to TTL / USB-TTL /STC microcontroller programmer / PL2303 in nine upgrades plate with a transparent cover Compatible with ARDUINO, RASPBERRY PI, AVR, PIC, 8051, etc. The USB to RS232 module based TTL provides the best and convenient way to connect your RS232 TTL Devices or demo board to your PC via the USB port. Adopt imported controller PL2303HX, which can stabilize the flash with high speed, 500mA self-recovery fuse for protection. Two data transmission indicator can monitor data transfer status in real time. Reserve 3.3V and 5V pin interface, easy for the DDWRT of different voltage system that need power. The entire board is coated by high quality transparent heat-shrinkable sleeve, making the PCB in insulation state from outside, so that the board won't burnt down by material short cut. Electrostatic package, insures the board will not be damaged before use.

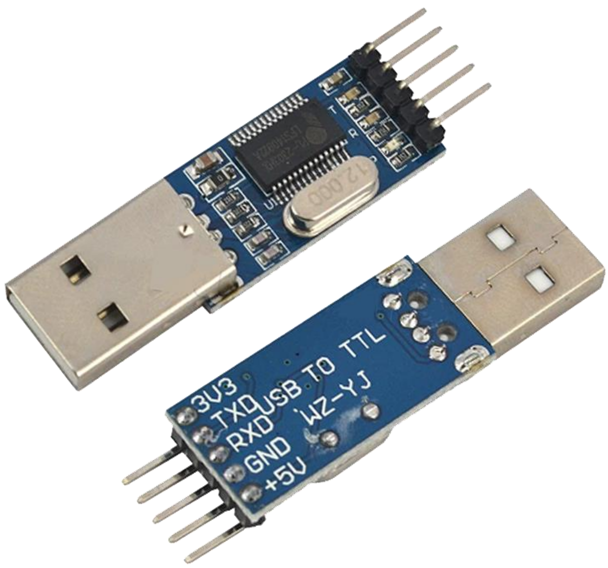


Fig.4.5: USB to TTL Converter

4.6 18650-LITHIUM-ION BATTERY:

A lithium-ion battery or Li-ion battery is a type of rechargeable battery. Lithium-ion batteries are commonly used for portable electronics and electric vehicles and are growing in popularity for military and aerospace applications. Lithium ions move from the negative electrode through an electrolyte to the positive electrode during discharge, and back when charging. Li-ion batteries use an intercalated lithium compound as the material at the positive electrode and typically graphite at the negative electrode. The batteries have a high energy density, no memory effect (other than LFP cells) and low self-discharge.

Research areas for lithium-ion batteries include extending lifetime, increasing energy density, improving safety, reducing cost, and increasing charging speed, among others. Research has been under way in the area of non-flammable electrolytes as a pathway to increased safety based on the flammability and volatility of the organic solvents used in the typical electrolyte. During discharge, an oxidation half-reaction at the anode produces positively charged lithium ions and negatively charged electrons. The oxidation half-reaction may also produce uncharged material that remains at the anode. Lithium ions move through the electrolyte, electrons move through the external circuit, and then they recombine at the cathode (together with the cathode material) in a reduction half-reaction. The electrolyte and external circuit provide conductive media for lithium ions and electrons, respectively, but do not partake in the electrochemical reaction. During discharge, electrons flow from the negative electrode (electrons move from the positive electrode to the negative electrode through the external circuit. To charge the cell the external circuit has to provide electric energy. This energy is then stored as chemical energy in the cell (with some loss, e. g. due to coulombic efficiency lower than 1).

Fig.4.6: 18650-Lithium-ion battery

CHAPTER V

SOFTWARE

5.1 ARDUINO IDE:

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards.

5.2 WIRED COMMUNICATION:

Wired communication refers to the transmission of data over a wire-based communication technology. Wired communication is also known as wireline communication. Examples include telephone networks, cable television or internet access, and fiber-optic communication. Most wired networks use Ethernet cables to transfer data between connected PCs. Also waveguide (electromagnetism), used for high-power applications, is considered wired line. Local telephone networks often form the basis for wired communications and are used by both residential and business customers in the area. Many networks today rely on the use of fiber optic communication technology as a means of providing clear signaling for both inbound and outbound transmissions and are replacing copper wire transmission. Fiber optic technology is capable of accommodating far more signals than copper wiring while still maintaining the integrity of the signal over longer distances.

5.2.1 INFRASTRUCTURE:

The Robots will become a part of the fabric of everyday life. It will become part of our overall life just like water, electricity, telephone, TV and most recently the Internet. Whereas the current Internet typically connects full-scale computers, the Internet of Things (as part of the future internet) will connect everyday objects as well as robots with a strong integration into the physical world.

1. Plug and control integration:

If we look at advanced lead through technology, we can just make the connection between the real-time robot ad the scale down model. We can lively control them and record the path functions to follow in production.

2. Infrastructure functionality:

The infrastructure needs to support the robot applications in finding the work volume. Robots may run anywhere once the path to follow instructions are thought. But if we have proper infra structure, we can make use of the effective work volume with well-defined work functions.

3. Physical location and position:

As the robot is strongly rooted in the physical world, the motion parameters and positions are very important, especially for finding exact angles of joints, it is recommendable to use high sensitive hall Effect gimbals in place of potentiometers.

4. Security and Protection:

In addition, an infrastructure needs to provide support for security and privacy functions including identification, integrity, confidentiality and authorization. The biometry systems and encryption should be deployed in the control station that enhances the limits of the robot.

5. Data management:

It is a crucial aspect in robot programing when considering a world of objects interconnected and constantly exchanging all types of information, the volume of the generated data and the process involved in handling those data becomes critical. We can see the actions performed by the robot in detail through the serial monitor. Also we can record the data and make it a s a log for future reference or to plot the graph of actions performed by that robot graphically. the opportunities of data management are,

- Data collection and analysis
- Interval calculation
- Multi robot networking
- Virtual process plan
- Complex operation plan

5.2.2 BENEFITS OF ADVANCED LEAD THROUGH PROGRAMING:

The Advancement in Lead through programing offers a number of benefits to organizations, enablingthem to:

- Need not to monitor their overall actions
- improve the facility
- save time and money
- enhance employee productivity
- integrate and adapt business models
- make better business decisions
- generate more revenue

This encourages companies to rethink the ways they approach their businesses, industries and markets and gives them the tools to improve their business strategies.

5.2.3 APPLICATIONS:

- Small scale industries
- Low accuracy high precision manufacturing industries
- Hazardous locations
- Waste management
- Incident management system
- Object Tracking
- Hospital management

- Human assistance
- Cooking and serving
- Quality control
- Industry 4.0

5.3 C & C++ LANGUAGE:

The Arduino IDE supports the languages C and C++ using special rules of code structuring. These are an open source programming language which is interpreted and are required to be compiled to run. If the programmer needs to change the code, they can quickly see the results in the serial monitor. It is a high-level language and writing time is not to be considered since after single time perfect programing, we are not going to look back and alter the program. These are flexible and dynamic language that you can use in different ways. It can be used interactively when ones simply want to test a code or a statement.

CHAPTER VI

RESULT AND DISCUSSION

6.1 RESULT:

The robot was tested in three different purposes – one for finding the accuracy and precision another to check the live control and another for pick and place robot programing. The robot was capable of performing any operations that can be achieved inside the work volume with selective end effector and joint drivers. Providing navigation instructions to the robot that learns .the autonomous action of the robot was achieved in these tests.

The programing was executed and the real-time robot is capable of learning the job and to follow the path we instructed through the small model.

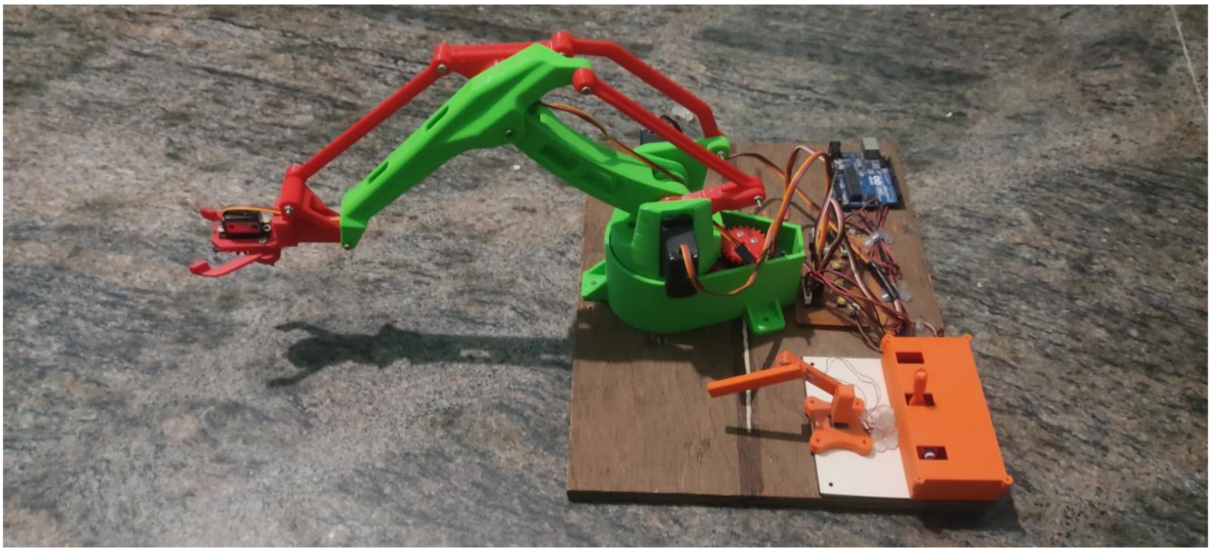


Fig 6.1: Proposed system

6.1.1 COMPARISON TABLE:

CONVENTIONAL PROGRAMING VS ADVANCED LEAD THROUGH PROGRAMING:

	Conventional Programing	Advanced lead through programing
Live control	Not-possible	Possible
Live control latency	-	20mS
Save position delay	150mS	1000mS
Driver type	Stepper/Servo	Servo
PWM frequency	Based on driver	490 Hz
Robot cost	>60L	<25L
Emergency reset	<50mS	<100mS
Accuracy	1 micron	800 microns
Precision	< 0.5mm	<0.70mm
Controller	Joystick/PC	Scale down model
Suitable Application	Job machining	Job handling

The entire system is powered with Li-ion cells. Convientional programing like offline,teachpendent and leadthrough methods are compared with this new method on the table above.

This shows the difference between conventional programing and advanced lead through programing. As per our result, this programing method is highly suitable for low investment material handling purpose that ensures high accuracy and precision parameters.

6.2 DISCUSSION:

The system performed the desired functions and the objectives stated were met. With all these operations being performed, it is indeed a low-cost sustainable investment for building the industrial robots for the people and making the quality and variety products with user desirable manner that leads to people to be a better entrepreneur there by the economy will be sustained in industry 4.0.

CHAPTER VII

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION:

After implementation of the advanced lead through programing, the automation meets the defined objectives. It is easy to use and provides simple user interface. It requires low maintenance, which is also an additional advantage. The Programing method was mainly developed in the aim of a low-cost project, and this can be developed as a real time huge robots and can serve for the purpose of small and medium scale industries for ease of use with greater FOS.

We hope that this project enables the implementation of robots in small scale industries as the programing simplified.

7.2 FUTURE SCOPE:

The MIMIC program has successful outputs, but integration with the systemwirelessly remains as the future work of this project.

REFERENCES

1. Alexander Skoglund,Boyko Iliev,Bourhane Kadmiry,Rainer Palm, ‘Programming by Demonstration of Pick-and-Place Tasks for Industrial Manipulators using Task Primitives’, 2007 International Symposium on Computational Intelligence in Robotics and Automation.

2. Pei-Chi Huang,Aloysius K. Mok, ‘A Case Study of Cyber-Physical System Design: Autonomous Pick-and-Place Robot’ , 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA).

3. Peng Hao; Tao Lu; Yinghao Cai; Shuo Wang, ‘Programming by Visual Demonstration for Pick-and-Place Tasks using Robot Skills’, 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)

4. Mahmoud Abdelaal,‘A Study of Robot Control Programing for an Industrial Robotic Arm’ 2019 6th International Conference on Advanced Control Circuits and Systems (ACCS) & 2019 5th International Conference on New Paradigms in Electronics & information Technology (PEIT)

5. Chaitanya S. Gajbhiye; Megha G Krishnan; S. Kumaravel; S. Ashok‘Fuzzy — Arduino based control strategy for human safety in industrial robots, 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)

6. Luka Peternel; Tadej Petrič; Jan Babič, ‘Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface’, 2015 IEEE International Conference on Robotics and Automation (ICRA).

7. Martin Tykal; Alberto Montebelli; Ville Kyrki, ‘Incrementally assisted kinesthetic teaching for programming by demonstration’, 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)

8. Christian Kohrt; Anthony Pipe; Gudrun Schiedermeier; Richard Stamp; Janice Kiely, ‘A robot manipulator communications and control framework’, 2008 IEEE International Conference on Mechatronics and Automation

9. Y. Kuniyoshi; M. Inaba; H. Inoue, ‘Learning by watching: extracting reusable task knowledge from visual observation of human performance IEEE Transactions on Robotics and Automation (Volume: 10, Issue: 6, Dec 1994)

10. Rahul Kumar; Sunil Lal; Sanjesh Kumar; Praneel Chand,’ Object detection and recognition for a pick and place Robot Asia-Pacific World Congress on Computer Science and Engineering, IEEE Asia-Pacific World Congress on Computer Science and Engineering 2014

APPENDIX

MIMIC CODE:

```
#include <Servo.h>

Servo servo_a;  //Gripper

Servo servo_b;  //Shoulder

Servo servo_c;  //Elbow

Servo servo_d;  //Base

int pot1;   //Gripper

int pot2;   //Shoulder

int pot3;   //Elbow

int pot4;   //Base

int servo1[10];

int servo2[10];

int servo3[10];

int servo4[10];

int angle1;  //Gripper

int angle2;  //Shoulder

int angle3;  //Elbow

int angle4;  //Base
```

```
int savebutton = 8;
int playbutton=12;
int resetbutton = 13;
int switchpoint =0;
int led=7;
boolean ledon=false;
boolean lastbutton=LOW;
boolean currentbutton=LOW;
int cycle = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(savebutton, INPUT);
  pinMode(playbutton, INPUT);
  pinMode(resetbutton, INPUT);
  pinMode(7,OUTPUT);

  servo_a.attach(9);//Base
  servo_b.attach(10);//Shoulder
  servo_c.attach(11);//Elbow
```

```
  servo_d.attach(5);//Gripper
 /*
  Serial.println("Hi, I am MIMIC!");
  delay(2000);
  Serial.println("You can Control, Record & Play back the motion you
needed.");
  delay(2000);
  Serial.println("");
  Serial.println("CAUTION - Please follow the instructions Below...");
  delay(3000);
  Serial.println("");
  Serial.println("Press Button-A to Record currect position of Mimic. (Must
Record 10 Positions to Playback)");
  delay(5000);
  Serial.println("Press Button-B to Playback the recorded motion.");
  delay(3000);
  Serial.println("Press & Hold Button-C to Reset the motion. (Must complete a
play cycle)");
  delay(5000);
  Serial.println("Press Button-D for Emergency stop and Program reboot.");
  delay(3000);
  Serial.println("");
```

```
  Serial.print("Connecting.");
  delay(400);
  Serial.print(".");
  delay(400);
  Serial.print(".");
  delay(400);
  Serial.print(".");
  delay(400);
  Serial.print(".");
  delay(400);
  Serial.println(".");
  delay(400);
  Serial.println("Live Control Is Activated.");
  delay(2000);
  */
}


boolean debounce (boolean last)
{
  boolean current = digitalRead(playbutton);
  if(last!=current)
```

```
  {
    delay(5);
    current= digitalRead(playbutton);
  }
  return current;
}

void(* resetFunc) (void) = 0;

void  loop()
{
  currentbutton=debounce(lastbutton);
  if(lastbutton==LOW&&currentbutton==HIGH)
  {
    ledon=!ledon;
  }
  lastbutton=currentbutton;
  digitalWrite(led,ledon);


//Base
```

```
pot1=analogRead(A1);
angle1=map(pot1,0,1023,0,180);
servo_a.write(angle1);

//Shoulder
pot2=analogRead(A2);
angle2=map(pot2,0,1023,0,180);
servo_b.write(angle2);

//Elbow
pot3=analogRead(A3);
angle3=map(pot3,0,1023,0,180);
servo_c.write(angle3);

//Gripper
pot4=analogRead(A4);
angle4=map(pot4,0,1023,180,0);
servo_d.write(angle4);
    Serial.println(angle1);
    Serial.println(angle2);
    Serial.println(angle3);
```

```
      Serial.println(angle4);




if(digitalRead(savebutton) == HIGH)
  {
  switchpoint++;
  switch(switchpoint)
    {
     case 1:
      servo1[0] = angle1;
      servo2[0] = angle2;
      servo3[0] = angle3;
      servo4[0] = angle4;
      Serial.println("");
      Serial.println("Position 1 out of 10 is Saved");
      delay(1000);
      break;

     case 2:
      servo1[1] = angle1;
```

```
servo2[1] = angle2;
servo3[1] = angle3;
servo4[1] = angle4;
Serial.println("Position 2 out of 10 is Saved");
delay(1000);
break;

case 3:
servo1[2] = angle1;
servo2[2] = angle2;
servo3[2] = angle3;
servo4[2] = angle4;
Serial.println("Position 3 out of 10 is Saved");
delay(1000);
break;

case 4:
servo1[3] = angle1;
servo2[3] = angle2;
servo3[3] = angle3;
servo4[3] = angle4;
```

```
Serial.println("Position 4 out of 10 is Saved");
delay(1000);
break;

case 5:
servo1[4] = angle1;
servo2[4] = angle2;
servo3[4] = angle3;
servo4[4] = angle4;
Serial.println("Position 5 out of 10 is Saved");
delay(1000);
break;

case 6:
servo1[5] = angle1;
servo2[5] = angle2;
servo3[5] = angle3;
servo4[5] = angle4;
Serial.println("Position 6 out of 10 is Saved");
delay(1000);
break;
```

```
case 7:

servo1[6] = angle1;

servo2[6] = angle2;

servo3[6] = angle3;

servo4[6] = angle4;

Serial.println("Position 7 out of 10 is Saved");

delay(1000);

break;


case 8:

servo1[7] = angle1;

servo2[7] = angle2;

servo3[7] = angle3;

servo4[7] = angle4;

Serial.println("Position 8 out of 10 is Saved");

delay(1000);

break;


case 9:

servo1[8] = angle1;
```

```
servo2[8] = angle2;
servo3[8] = angle3;
servo4[8] = angle4;
Serial.println("Position 9 out of 10 is Saved");
delay(1000);
break;

case 10:
servo1[9] = angle1;
servo2[9] = angle2;
servo3[9] = angle3;
servo4[9] = angle4;

Serial.println("Position 10 out of 10 is Saved");
Serial.println("");
Serial.print("Preparing.");
delay(400);
Serial.print(".");
delay(400);
Serial.print(".");
delay(400);
```

```
      Serial.print(".");

      delay(400);

      Serial.println(".");

      delay(400);

      Serial.println("Ready to play.");

      delay(1000);

      break;

    }

 }



if(digitalRead(led) == HIGH)

{

  Serial.println("");

  Serial.print("Play Cycle-");

  cycle = cycle+1 ;

  Serial.println(cycle);


  for(int i = 0; i < 10; i++)

     {

     Serial.print("Possition ");
```

```
Serial.print(i+1);
Serial.print(" ---> ");
{
 servo_a.write(servo1[i]);
 Serial.print("Base->");
 Serial.print(servo1[i]);
 Serial.print("   ");

 servo_b.write(servo2[i]);
 Serial.print("Shoulder->");
 Serial.print(servo2[i]);
 Serial.print("   ");

 servo_c.write(servo3[i]);
 Serial.print("Elbow->");
 Serial.print(servo3[i]);
 Serial.print("   ");

 servo_d.write(servo4[i]);
 Serial.print("Gripper->");
 Serial.print(servo4[i]);
```

```
      Serial.print("   ");
     }delay(1000);
    }
}

if(digitalRead(resetbutton) == HIGH)
 {
  delay(100);
  Serial.println("");
  Serial.print("Processing...");
  Serial.print(".");
  delay(500);
  Serial.print(".");
  delay(500);
  Serial.println(".");
  delay(500);
  Serial.println("reset complete");
  Serial.println("");
  delay(1000);
  resetFunc();
 }
```

```
}
```