**CHAPTER 1**

**INTRODUCTION**

**CHAPTER 1**

**INTRODUCTION**

**1.1 DESCRIPTION**

Robotics is the branch of technology that deals with the design, construction, operation, and application of robots, as well as computer systems for their control, sensory feedback, and information processing. Robots in laboratory, life science and pharmaceutical applications perform tasks at rates beyond human capability. It deals with the designation, construction, operation and application of numerous systems like mechanical, electrical, electronics, software, and information technology which are interrelated in a constrained approach so as to achieve the designated function of the system. Its interdisciplinary reputation has revolutionized numerous fields of its involvement. Warehouse management is addressed to be one such field that inherits its vast possibilities, versatility and virtues. It’s alone a rapidly expanding domain in storage field.

In the 20th century robotics technology is in its peak in terms of usage and opurtunities. These robots can function in potentially hazardous settings in proximity to biological dangers, the threat of radioactive contamination, and toxic chemotherapy compounds. Due to the potential hazards and high volumes, some industries and chemical factories are making use of automation to get things done. Many of the warehouses makes use of semiautomatic systems to get things done more faster and efficiently . The technology is available but regulation will pace the growth of robotic warehouses more than the technology.

Robotics can decrease the amount of human errors which can lead to many disruption in the workplace .With a fully automated system ,the amount of work done will be much higher and the amount of time consumed will be much lower, thus making the system for fast and reliant. The conventional systems such as Cartesian robots are used in most warehouses , this however can limit the flexibility of the system in general and makes it more harder to operate under several conditions. By addition of two 6 DOF robotic arm we ensure the flexibility of the system as well as the scope for future improvements. Automated systems mean such tracking can be made vastly more efficient. One problem, at least at this point, is that the capital to be invested in making a fully automated system is pretty higher than a semi-automated system. However, there is a general shift toward that as we are also encountering the pandemic and amount of human contact needs to be reduced even more day by day, which means robotics will be needed to satisfy these holes in the fields thus making this system more appealing.

The design of warehouse management system t uses engineering method. In sequence, the method is identification of the needs required. Then these needs are analyzed to get specific components. The operation of the robot can be classified into three main categories. They are detection, motion control and storing.

**CHAPTER 2**

**LITERATURE REVIEW**

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 JOURNAL REFERENCES**

A literature review was conducted to investigate the history and achievements of application of robotics in warehouse management field. The source investigated involves the following observations

**C.Liang[1]** looks at the introduction of automated robot picking solution that meets the requirements of automating the picking of items from shelves. The hardware of the proposed system comprises a lightweight robot manipulator, a low-cost commercially available 3D camera system and a custom-built robotic gripper. Several design requirements are proposed to address the above-mentioned challenges.

1) In an e-commerce fulfilment warehouse, the movement of products is often crossing path with human workers. Moving products in such environments remains a challenge or existing robots. Safety is the first critical issue to be concerned with. Therefore, human-friendly

lightweight robot with human robot interaction capability

should be used.

2) E-commerce business is a very fast moving customer-oriented business. Different services are needed for meeting different customers’ requirements. To keep up with the high volume of daily orders, the robot picking system should be versatile and adaptable

**SaahilGupt [2]** In this paper propose a robotic arm which is designed, synchronized with the working arm and would perform the task as the working arm does. The robotic arm can be designed to perform any desired task such as welding, gripping, spinning etc depending on the application. For example, robot arms in automotive assembly line perform a variety of tasks such as wielding and parts rotation and placement.

**Omijeh [3]** In this paper the design of a Remote Controlled Robotic Vehicle has been completed. A prototype was built and confirmed functional. This system would make it easier for man to unrivalled the risk of handling suspicious objects which could be hazardous in its present environment and workplace. Complex and complicated duties would be achieved faster and more accurately with this design

.

**Bodhisatwa Barma[4]** proposes an automated system which address the fulfilment of the objectives of automatic retrieval of queued books arrangement of returned books on the racks as well as automated updating of the library database.

**Jamshed Iqbal [5]** This paper develops the kinematic models a 6DOF robotic arm and analyzes its workspace. The proposed model makes it possiblr to control the manipulator to achieve any reachable position and orientation in an unstructured environment.

**Muhammad Bilal** **[6]** This project describe a mechanical system, design concept and prototype implementation of a 6 DOF robotic arm, which should perform industrial task such as pick and place of fragile objects operation. This robot arm being controlled by

micro-controller has base, shoulder, elbow, wrist rotation and a functional gripper. Gripper has been built as end-effector and is capable of grasping diverse objects within own workspace of the arm possible. PID controller is implemented on each motor. The microcontroller implement forward kinematics and position control of DC motors. The design aims to provide fine manipulation in performing pick and place task, while still

maintaining the simplicity of design

**CHAPTER 3**

**PROBLEM IDENTIFICATION AND SOLUTION**

**CHAPTER 3**

**PROBLEM IDENTIFICATION AND SOLUTION**

**3.1 PROBLEM IDENTIFICATION**

Existing conventional system contain a human element that plays a vital role in functioning of the warehouse system. The warehouse manager and workers are the human element present in the warehouse system. In case of small ecommerce warehouses it is the humans that does the most work. The worker then sorts and finds the corresponding items and then manually fetches it or stores it from the storage space. The identification of the product as well as the picking and placing of the items is done via human intervention.

This whole process requires knowledge of each product and its storage area and the ability to do it sufficiently fast with less number of errors. Hence the human element faces a set of drawbacks in this field. These drawbacks can be easily withdrawn by application of robotics.

The current warehouse and storage management systems are mostly manual and time consuming .Manual work has its own share of human errors which lead to many complications in the work site.Conventional System for warehouse management is also requires multiple human interventions and workers .Since human involvement is a factor in the process also tends to make some compromise in the reliability of the system due to inclusion of humanly factors such as tiredness, sleepiness, fatigue, etc. Inventory control is time consuming and highly prone to errors, resulting in the necessity of multiple and frequent counts to be done on a regular basis.Storage, picking, packing and all other order fulfillment management processes require a great deal of time.

**3.2 PROBLEM SOLUTION**

**3.2.1 METHODOLOGY**

Fig 3.1 shows the methodology adopted in development of the project. The proposed system contains two 6 DOF robotic arms along with an RFID sensor system which identifies each product uniquely and picks and places in the different storage space. Due to the addition of automation it can improve accuracy, reliability, and reduce the time consumption. It has unlimited possibilities of expanding the capabilities by adding better and advanced hardware components that can improve the capabilities like memory, inventory analysis, dynamic inventory containment, etc.

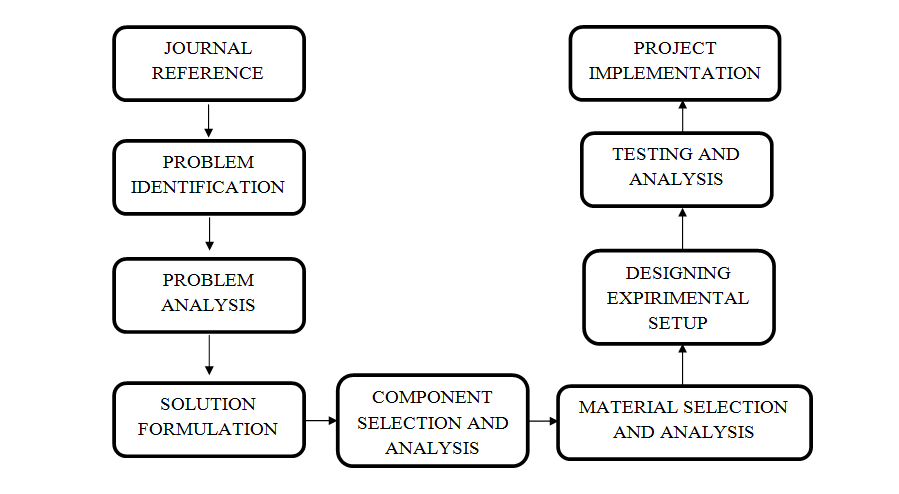
****

Fig 3.1 Methodology

Many number of journals related to the project were referred to in order to acquire enough data related to the problems faced in the warehouse management domain and the technological advancements up till date. The next step is to identify the drawbacks of the pre-existing conventional systems or also referred to as Problem identification. The identified drawbacks and demerits were analysed to determine the root of the cause and these root causes were determined and solutions were formulated. Next step is to list the components that are required to implement the project, their analysis, and selection. Similarly the raw materials required were also analysed and selected.

Next step is to design the experimental setup in 2D or 3D model and their testing and analysis is to be conducted. Any failure will be noticed and they will be rectified through study and analysis. Once the setup is perfected the implementation of the project is executed.

**3.2.2 SOLUTION**

Robotics and automation is the most effective methodology that can be employed as a solution to the challenges faced by the conventional warehouse management system. Robots have the ability to store and retrieve data from its memory regarding the product ,their storage space, their size and their manufacturers. Robots can distinguish the type of items simply by identifying the RFID tags which is given uniquely to each product. Robots have high repeatability and do not experience fatigue due to prolonged functioning of the system. A proper record of the inventory of items in the warehouse management system can be observed dynamically. A seamless and quick picking , identifying and storing can be performed.

**CHAPTER 4**

**HARDWARE DESCRIPTION**

**CHAPTER 4**

**HARDWARE DESCRIPTION**

* 1. **ARDUINO UNO**

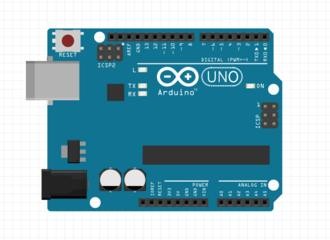
**INTRODUCTION**

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328 microcontroller and developed by Arduino.c. 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 pins, 6 Analog pins, and 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 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nan and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The Uno board is the first in a series of USB-based Arduino boards, and it and version 1.0 of the Arduino IDE were the reference versions of Arduino, now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to- serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to

version R2) programmed as a USB-to-serial converter. Fig 4.1 shows the image of Arduino Uno.



**Fig 4.1** Image of Arduino UNO

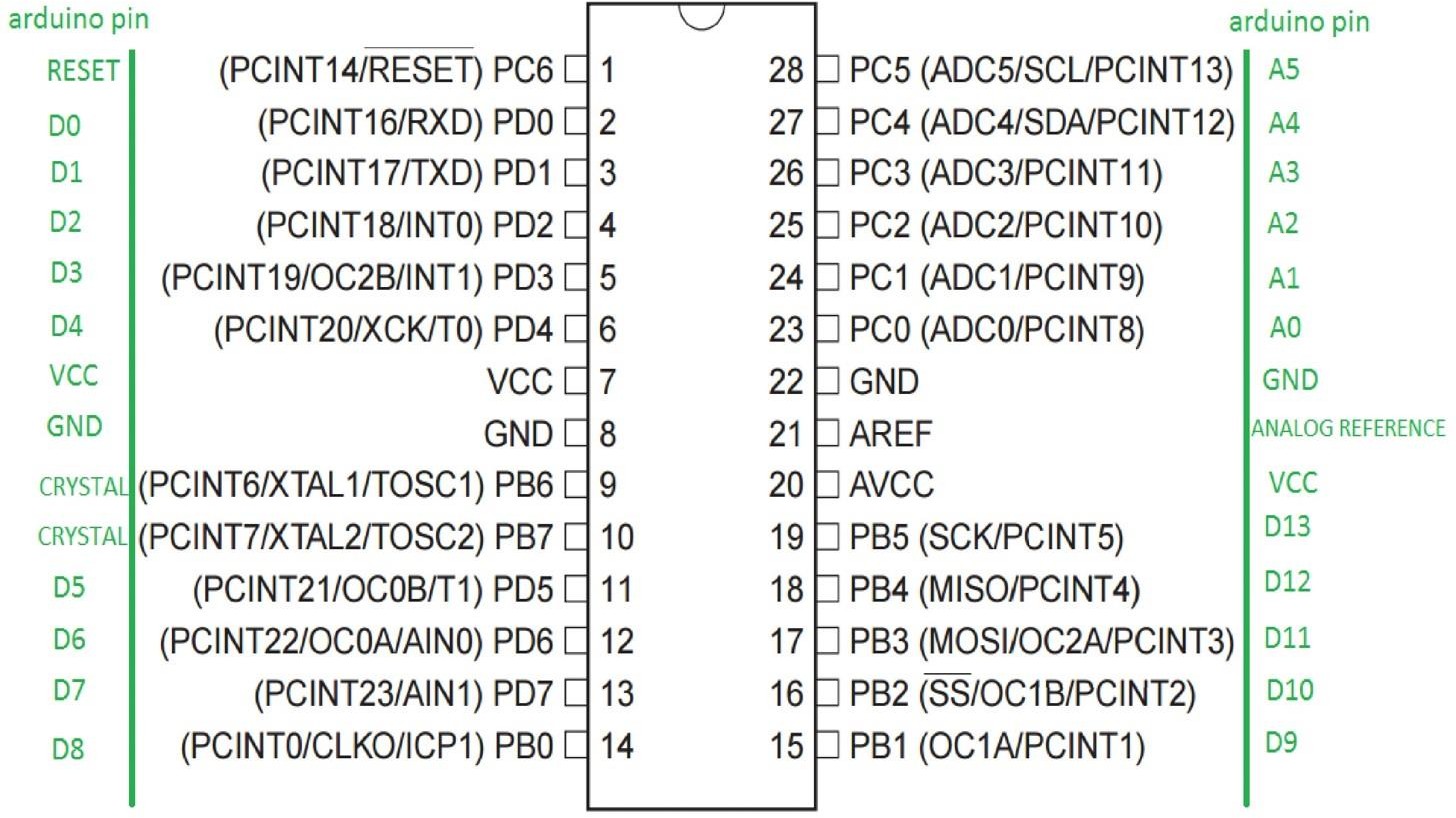
The Arduino project started at the Interaction Design Institute Ivrea (IDII) in [Ivrea](https://en.wikipedia.org/wiki/Ivrea), Italy. At that time, the students used a [BASIC](https://en.wikipedia.org/wiki/BASIC_Stamp) [Stamp](https://en.wikipedia.org/wiki/BASIC_Stamp) microcontroller at a cost of $100, a considerable expense for many students. In 2003 Hernando Barragan created the development platform [wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) as a Master's thesis project at IDII, under the supervision of Massimo Baozi and Casey Rees, who are known for work on the [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)) language. The project goal was to create simple, low-cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a [printed circuit board](https://en.wikipedia.org/wiki/Printed_circuit_board) (PCB) with an [ATmega](https://en.wikipedia.org/wiki/ATmega)168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller. In 2003, Massimo Baozi, with David Mallis, another IDII student, and David Cuatrilloes, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on

Wiring, they [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) the project and renamed it Arduino. Early [Arduino](https://en.wikipedia.org/wiki/Arduino) boards used the FTDI USB-to-serial driver chip and an [ATmega](https://en.wikipedia.org/wiki/ATmega)168. The Uno differed from all preceding boards by featuring the ATmega328P microcontroller and an ATmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

# TECHNICAL SPECIFICATIONS

* [Microcontroller](https://en.wikipedia.org/wiki/Microcontroller): [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P)
* Operating Voltage: 5 Volts
* Input Voltage: 7 to 20 Volts
* Digital I/O Pins: 14 (of which 6 provide PWM output)
* Analog Input Pins: 6
* DC Current per I/O Pin: 20 mA
* DC Current for 3.3V Pin: 50 mA
* [Flash Memory](https://en.wikipedia.org/wiki/Flash_Memory): 32 KB of which 0.5 KB used by [bootloader](https://en.wikipedia.org/wiki/Booting#BOOT-LOADER)
* [SRAM](https://en.wikipedia.org/wiki/Static_random-access_memory): 2 KB
* [EEPROM](https://en.wikipedia.org/wiki/EEPROM): 1 KB
* Clock Speed: 16 MHz
* Length: 68.6 mm
* Width: 53.4 mm
* Weight: 25 g

Fig 4.2 gives the pin diagram of Arduino Uno.



**Fig 4.2** Pin Diagram of Arduino UNO

# PINS

* **LED**: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it's off.
* **VIN**: The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
* **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
* **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* **GND**: Ground pins.
* **IOREF**: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
* **Reset**: Typically used to add a reset button to shields which block the one on the board.

**SPECIAL PIN FUNCTIONS**

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.

In addition, some pins have specialized functions:

* **Serial** / [UART](https://en.wikipedia.org/wiki/UART): pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
* **External interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* [**PWM**](https://en.wikipedia.org/wiki/Pulse-width_modulation) (pulse-width modulation): 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite() function.
* [**SPI**](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface) (Serial Peripheral Interface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
* **TWI** (two-wire interface) / [I²C](https://en.wikipedia.org/wiki/I%C2%B2C): A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
* **AREF** (analog reference): Reference voltage for the analog inputs.

# COMMUNICATIONS

The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows serial communication on any of the Uno's digital pins.

# AUTOMATIC RESET

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the

hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip.

This setup has other implications. When the Uno is connected to a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.

# DIGITAL SERVO MOTOR (MG996R)

**INTRODUCTION**

A servo is an electronic device that is used to operate RC cars, robots, electronic toys, and electrical appliances by determining the motion direction.

The 25KG digital servo features HV high torque, metal gear, fast heat dissipation, sensitivity and responsiveness. With high quality and performance, it is tailored for RC hobbyist to provide different experience and delight.

# FUNCTION

In general, a servo contains a DC motor, circuit, gear train, output shaft, and potentiometer. The potentiometer is a position sensor that is connected to the output shaft. The shaft is attached to the control wheels by gears. The movement and its direction are controlled by an electrical signal.

Servo motors function based on the waves of pulses sent by a command, like from a transmitter, to create a mechanical motion. The specification of a servo is normally being set by the turning angle per time at 60 degrees (speed) and the load carrying capacities (torque).

This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package.The gearing and motor have also been upgraded to improve dead bandwith and centering. High-torque standard servo can rotate approximately 120 degrees (60 in each direction).Provides great feedback and accuracy.

# WORKING

Servos have three wires that extend from the casing .Each of these wires serves a specific purpose. These three wires are for the control, power, and ground. The control wire is responsible for supplying the electrical pulses. The motor turns to the appropriate direction as commanded by the pulses.

When the motor rotates, it changes the resistance of the potentiometer and ultimately allows the control circuit to regulate the amount of movement and direction. When the shaft is at the desired position, the supply power shuts off. The power wire provides the servo with the power needed to operate, and the ground wire provides a connecting path separate from the main current. Fig 4.3 shows the image of a digital servo motor.



**Fig 4.3** Digital Servo Motor

A Digital RC Servo has a different way of sending pulse signals to the servo motor.If the analog servo is designed to send a constant 50 pulse voltage per second, the digital RC servo is capable of sending up to 300 pulses per second.

With this rapid pulse signals, the speed of the motor will increase significantly, and the torque will be more constant; it decreases the amount of deadband.

As a result, when the digital servo is used, it provides quicker response and faster acceleration to the RC component.Also, with fewer deadband, the torque also provides a better holding capability.

**4.3 SG90 MICRO SERVO MOTOR**

**INTRODUCTION**

It is tiny and lightweight with high output power. This servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. It comes with a 3 horns (arms) and hardware.

**SPECIFACTIONS**

* Operating voltage: 4.8 V (~5V)
* Operating speed: 0.1 s/60 degree
* Stall torque: 1.8 kgf·cm
* Dead band width: 10 µs
* Temperature range: 0 ºC – 55 ºC

### HARDWARE AND SOFTWARE REQUIRED

* SG90 Micro Servo motor
* Arudino Uno
* Arduino IDE(1.0.6V)



**Fig 4.4** SG90 Servo Motor



**Fig 4.5** SG90 Wire Connection

**HARDWARE CONNECTION**

The SG90 micro servo motor has 3 wire interface in which the connections should made as follows:

* Red wire-5V
* Brown wire-Ground
* Yellow wire-digital pin 9

**4.4 RFID SYSTEM**

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. An RFID tag consists of a tiny radio transponder; a radio receiver and transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device, the tag transmits digital data, usually an identifying inventory number, back to the reader. This number can be used to inventory goods.

There are two types of RFID Tags.  
1. Passive Tags  
2. Active Tags

Passive tags are powered by energy from the RFID reader's interrogating radio waves.

Active tags are powered by a battery and thus can be read at a greater range from the RFID reader; up to hundreds of meters.

Unlike a barcode, the tag doesn't need to be within the line of sight of the reader, so it may be embedded in the tracked object. RFID is one method of automatic identification and data capture. RFID tags are used in many industries. For example, an RFID tag attached to an automobile during production can be used to track its progress through the assembly line; RFID-tagged pharmaceuticals can be tracked through warehouses; and implanting RFID microchips in livestock and pets enables positive identification of animals.

An RFID System consists of 2 parts; namely the Tag and the reader.

**RFID TAG:**

RFID tags are made out of three pieces: a micro chip, an antenna for receiving and transmitting the signal and a substrate. The RFID tag includes either fixed or programmable logic for processing the transmission and sensor data, respectively.

RFID tags can be passive, active or battery-assisted passive. An active tag has an on-board battery and periodically transmits its ID signal. A battery-assisted passive has a small battery on board and is activated when in the presence of an RFID reader. A passive tag is cheaper and smaller because it has no battery; instead, the tag uses the radio energy transmitted by the reader. To operate a passive tag, it must be illuminated with a power level roughly a thousand times stronger than an active tag for signal transmission. That makes a difference in interference and in exposure to radiation.

Tags may either be read-only, having a factory-assigned serial number that is used as a key into a database, or may be read/write, where object-specific data can be written into the tag by the system user. Field programmable tags may be write-once, read-multiple. "Blank" tags may be written with an electronic product code by the user.

**READER:**

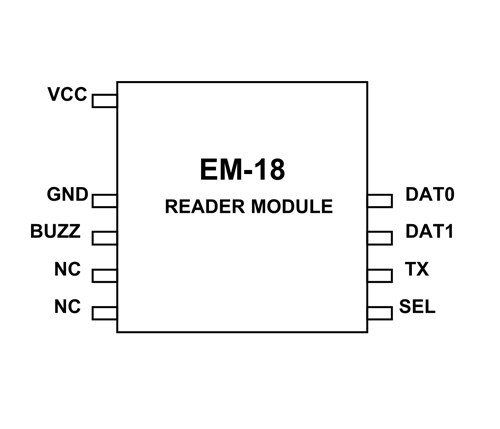
RFID systems can be classified by the type of tag and reader.

* A Passive Reader Active Tag system has a passive reader which only receives radio signals from active tags (battery operated, transmit only). The reception range of a PRAT system reader can be adjusted from 0–600 m, allowing flexibility in applications such as asset protection and supervision.
* An Active Reader Passive Tag system has an active reader, which transmits interrogator signals and also receives authentication replies from passive tags.
* An Active Reader Active Tag system uses active tags awoken with an interrogator signal from the active reader. A variation of this system could also use a Battery-Assisted Passive tag which acts like a passive tag but has a small battery to power the tag's return reporting signal.



The system we used is the EM-18 RFID Module with a reader and a tag as shown in figure 4.9. It is an Active Reader Passive Tag system. The Pin diagram of Em-18 Module is as shown in figure 4.10.

**Figure 4.6** EM-18 RFID Reder Module and Tags

  
**Figure 4.7** EM-18 RFID Reader Module Pin Diagram

The Pin Description of EM-18 RFID Reader Module is given in Table 4.1.

**Table 4.1** Pin Description of EM-18 RFID Reader Module

|  |  |
| --- | --- |
| **Pin Number** | **Description** |
| VCC | Should be connected to positive of power source. |
| GND | Should be connected to ground |
| BUZZ | Should be connected to BUZZER |
| NC | No Connection |
| NC | No Connection |
| SEL | SEL=1 then o/p =RS232  SEL=0then o/p=WEIGAND |
| TX | DATA is given out through TX of RS232 |
| DATA1 | WEIGAND interface DATA HIGH pin |
| DATA0 | WEIGAND interface DATA LOW pin |

**EM-18 FEATURES AND SPECIFICATIONS:**

* Operating voltage of EM-18: +4.5V to +5.5V
* Current consumption:50mA
* Can operate on LOW power
* Operating temperature: 0ºC to +80ºC
* Operating frequency:125KHz
* Communication parameter:9600bps
* Reading distance: 10cm, depending on TAG
* Integrated Antenna

**4.5 VARIABLE RHEOSTAT**

A rheostat is a variable resistor which is used to control current. They are able to vary the resistance in a circuit without interruption. The construction is very similar to the construction of a potentiometers. It uses only two connections, even when 3 terminals are present. The first connection is made to one end of the resistive element and the other connection to the wiper which is a sliding contact.  In contrast to potentiometers, rheostats have to carry a significant current. Therefore they are mostly constructed as wire wound resistors. Resistive wire is wound around an insulating ceramic core and the wiper slides over the windings.



**Fig 4.8** Variable Rheostat

**SPECIFICATIONS**

* Variable diameter – 6mm
* 2M LOG – 2 Mega ohm with a logarithmic track.

# 4.6 LEAD-ACID BATTERY

# Introduction

The **lead–acid battery** was invented in 1859 by French physicist Gaston Planté and is the earliest, yet still most widely used, type of rechargeable battery. Despite having a very low energy-to-weight ratio and a low energy-to-volume ratio, its ability to supply high surge currents means that the cells have a relatively large power-to-weight ratio. These features, along with their low cost, make them attractive for use in motor vehicles to provide the high current required by automobile starter motors.

As they are inexpensive compared to newer technologies, lead–acid batteries are widely used even when surge current is not important and other designs could provide higher energy densities. In 1999 lead–acid battery sales accounted for 40–45% of the value from batteries sold worldwide (excluding China and Russia), equivalent to a manufacturing market value of about $15 billion. Large-format lead–acid designs are widely used for storage in backup power supplies in cell phone tower, high-availability settings like hospitals, and stand-alone power systems. For these roles, modified versions of the standard cell may be used to improve storage times and reduce maintenance requirements. Gel-cells and absorbed glass- mat batteries are common in these roles, collectively known as VRLA (valve-regulated lead–acid) batteries.

The electrical energy produced by a discharging lead–acid battery can be attributed to the energy released when the strong chemical bonds of water (H2O) molecules are formed from H+ ions of the acid and O2− ions of PbO2. Conversely, during charging the battery acts as a water-splitting device, and in the charged state the chemical energy of the battery is stored in the potential difference between the pure lead at the negative side and the PbO2 on the positive side, plus the Sulphuric Acid in aqueous condition.

**CHAPTER 5**

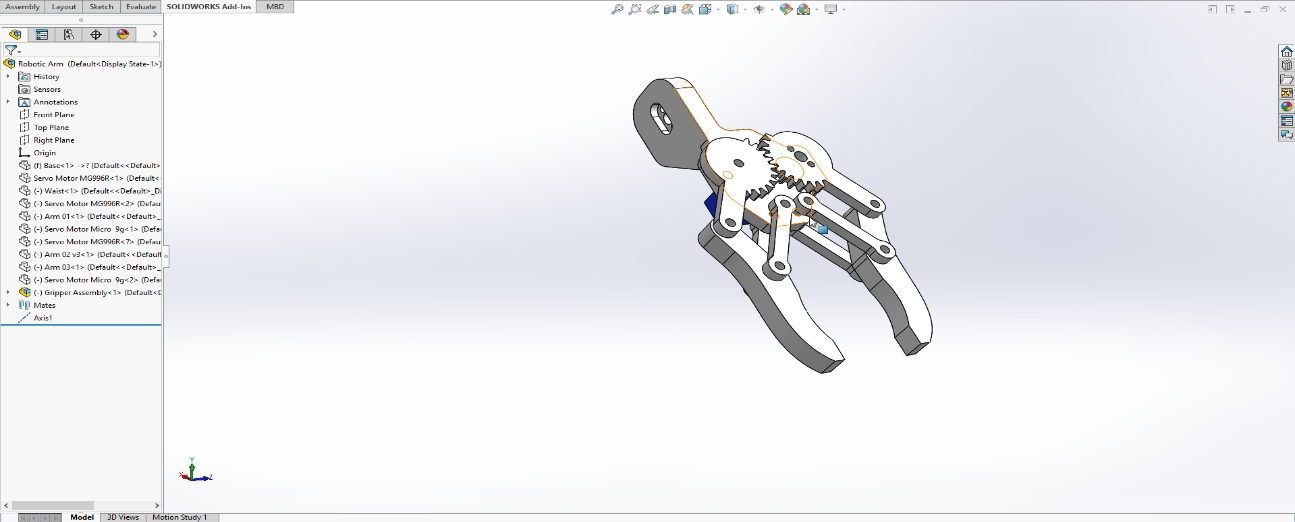
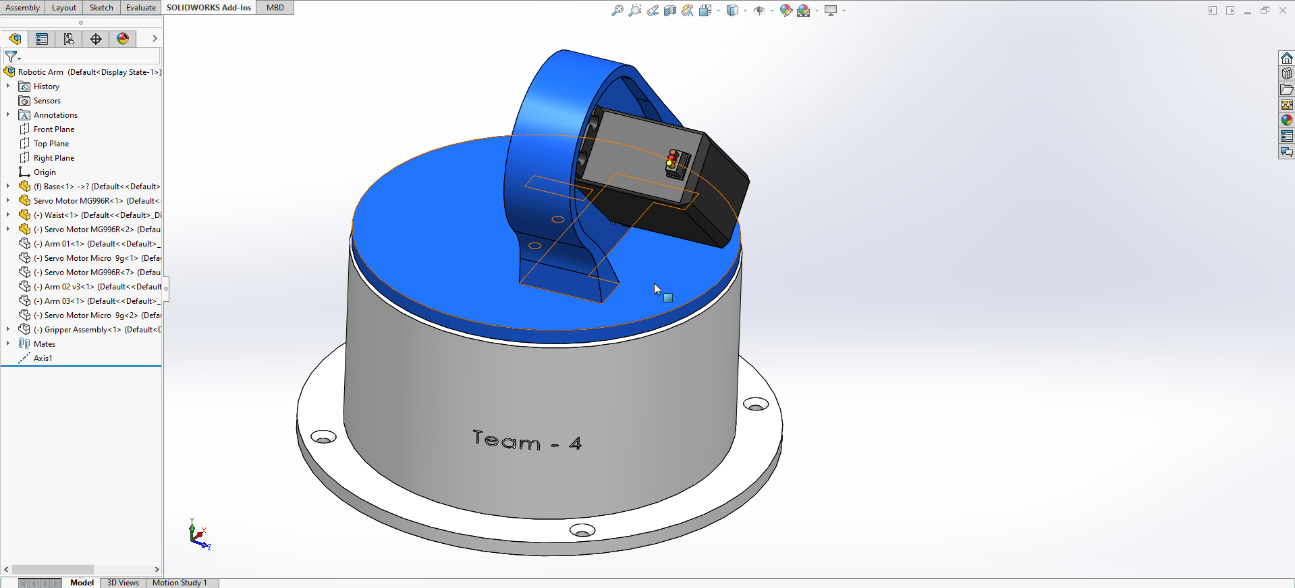
**PROJECT DESIGN**

**CHAPTER 5**

**PROJECT DESIGN**

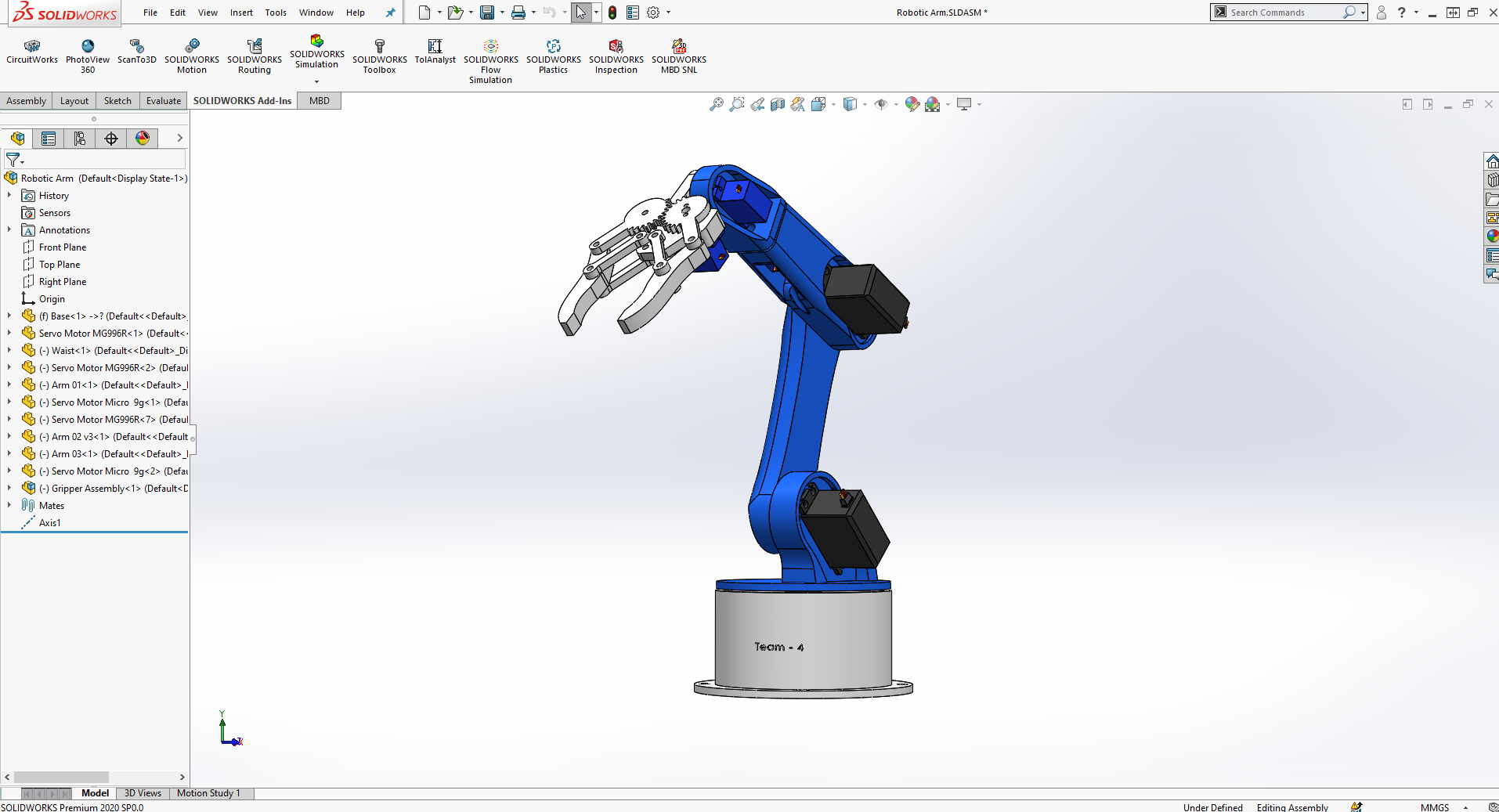
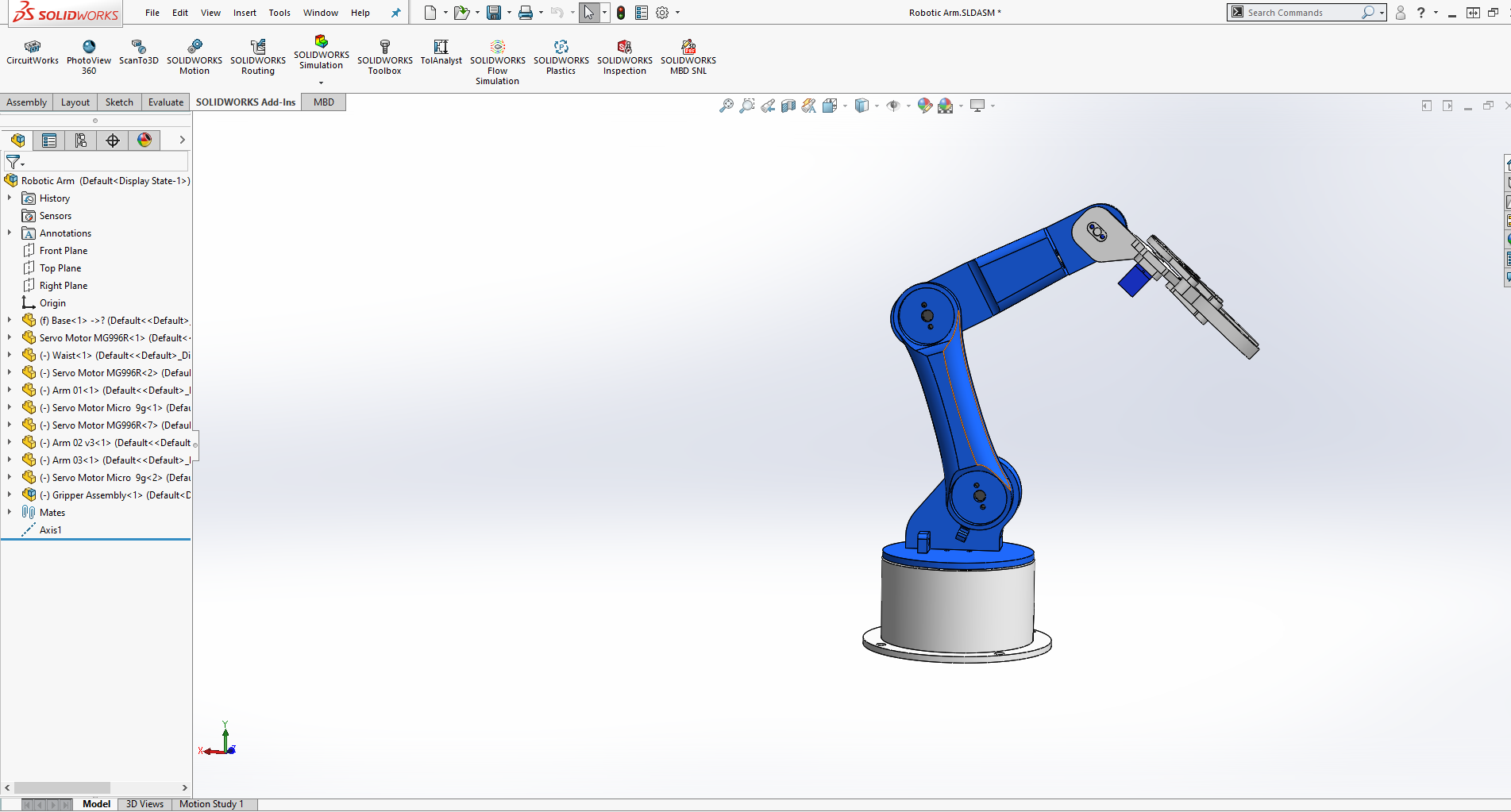
**5.1 PROJECT MODEL**

The design shown fig 5.1 shows the CAD representation of the robot.

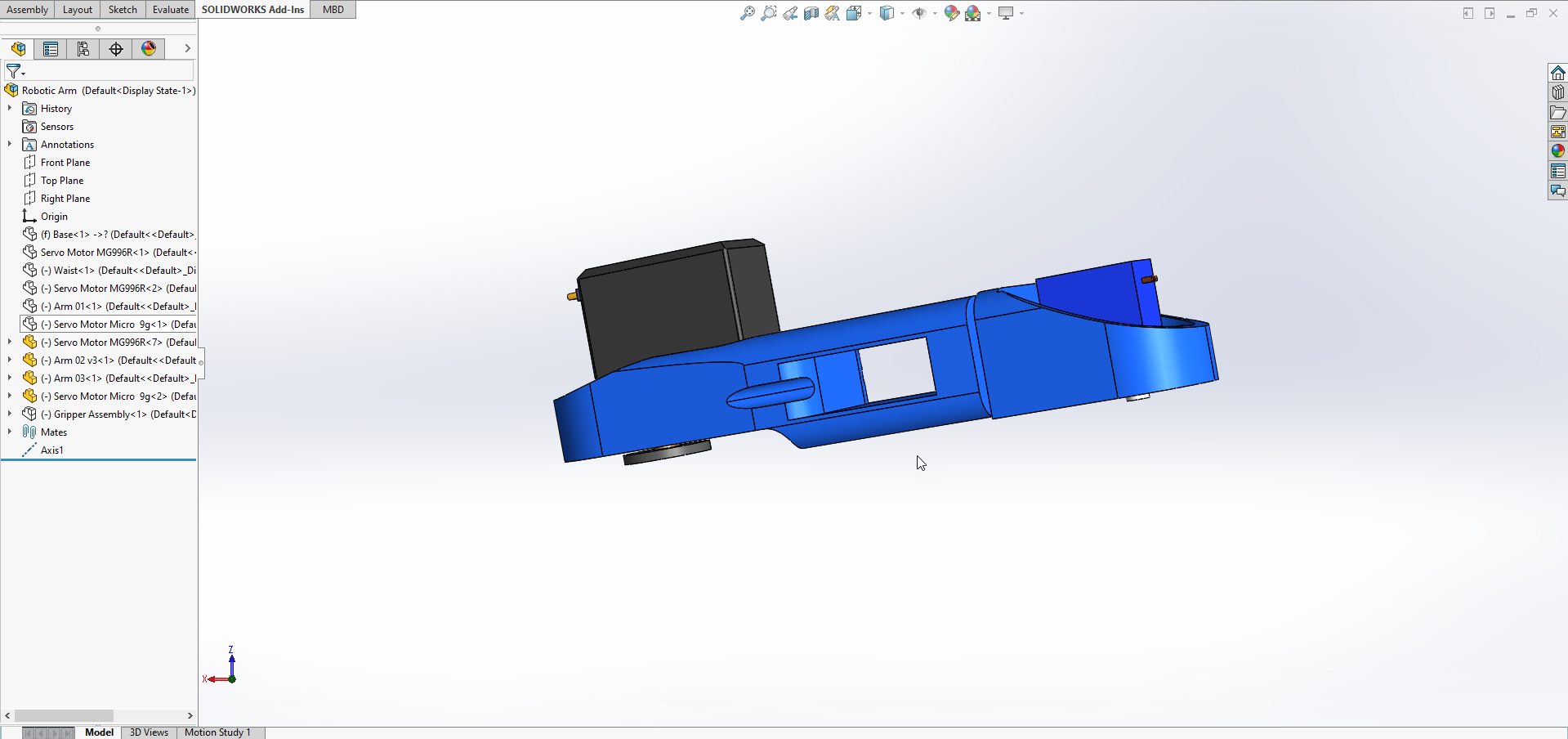
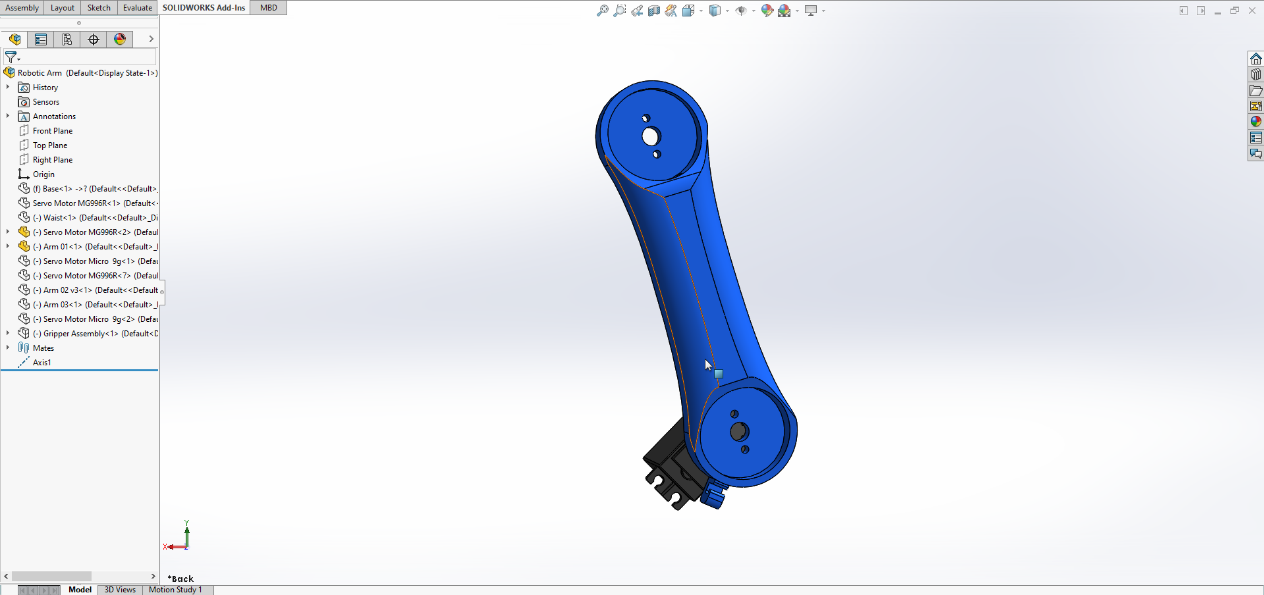


The model is an exact copy of the original prototype

**Fig 5.1** CAD drawing of gripper and base

**Figure 5.2** full body of the 6dof arm

****

**Fig 5.2** different parts of the robotic arm

**5.2 FABRICATION PROCESS**

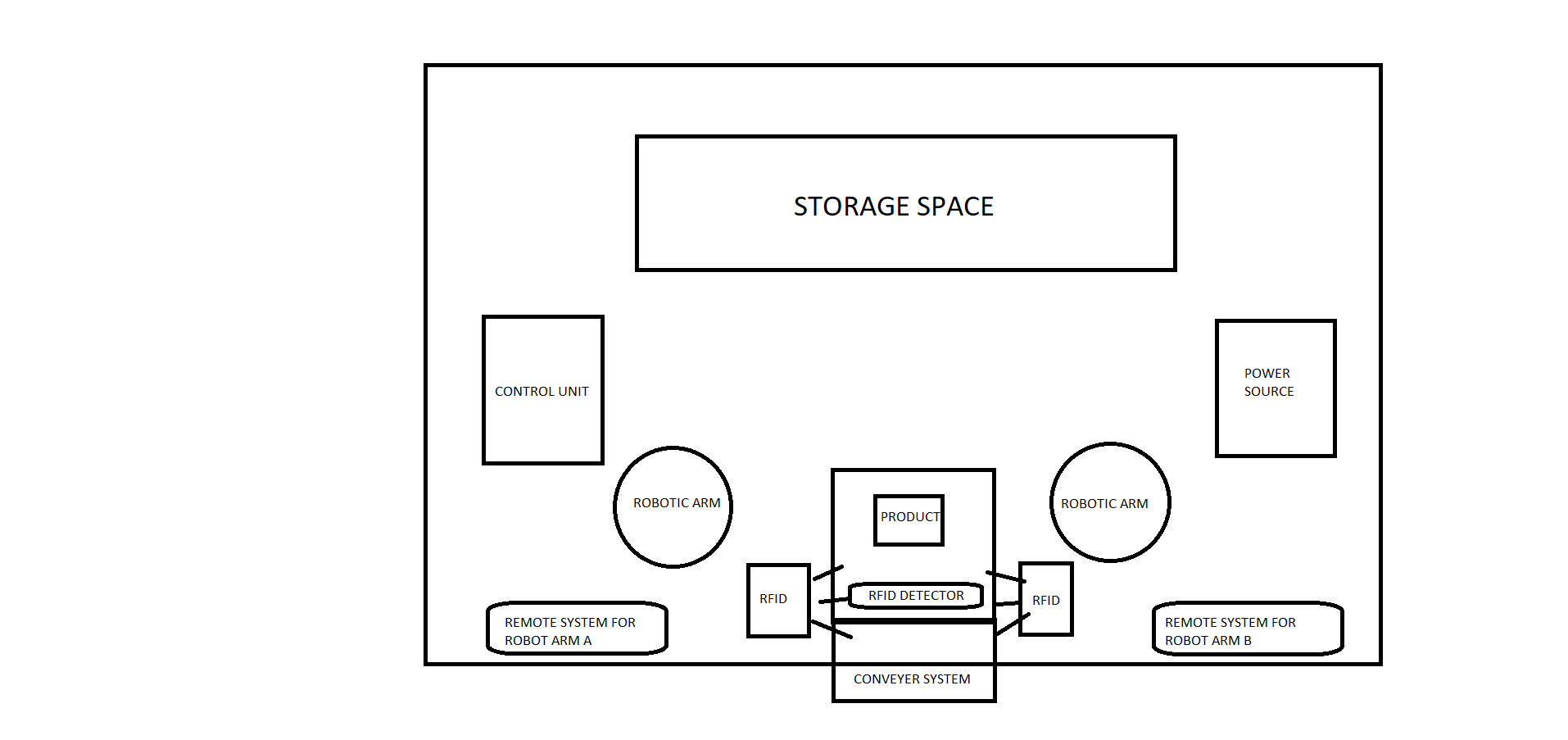
The problems which were discussed in earlier chapters are resolved and fabricated based on the following techniques:

* Used PLA filaments for 3D printing the body of the robotic arm and and gripper.
* The system base is made from a solid wood platform or 1X 1.5m dimensions.
* Both MG996R and SG90 motor provides speed, decent accuracy and high torque giving much agility and strength.
* The RFID reader module helps in verifying that the item picked by the robot and helps in checking the storage space available for the item..
* This speeds up the process of identifying, picking and placing the item in the specified storage area thus making the whole system more efficient and repeatable.

**5.3 WORKING PRINCIPLE**

The system is fully automated and only manual process is the input of the prescription and the room specified to the medicine. The following are the steps involved in the working of the robot:

**MODEL DIAGRAM**



* **DETECTION:**

Initially the every product is given a unique rfid tag and assigned a specific storage location. Now when the product is placed in the conveyor system the conveyor carries the product to the rfid detection system. Once the product reaches RFID detection system the tag is scanned and detected .

* **IDENTIFICATION:**

Now the RFID tags are uniquely identified and the corresponding signal is send to the Arduino system. Arduino board then checks for the storage space available in the storage area assigned to the product and checks if it is available. Through the infrared sensors placed in the storage area the processor is able to check the storage space available.

* **PICK AND PLACE OPERATION:**

The next step is to pick the object from the conveyor and place it in the storage space available. Our system consists of two 6 dof robotic arms to do the processor alternatively thus making the system more efficient and fast. The robotic arm comes with a control panel consisting of 6 variable rheostats for control of the 6 motors. A teaching pendant is added to the robotic arm so that the operator can teach the robot a specific task and make the robot repeat the task endlessly without inaccuracies. This additional feature allows the whole system to be more flexible and do many other tasks other than warehouse management. The advantage of this system over conventional is its effectiveness and fully autonomous feature which allows this system to be able to work under minimum interferences and interventions.

**5.4 MODEL CALCULATION**

**5.4.1 GRIPPING FORCE UNDER NORMAL   
 CONDITION:**

**GRIPPING FORCE CALCULATION**:

F : Gripping force [N] …… Sum of push forces

μ : Coefficient of static friction between the finger attachment and the work part

m : Work part weight [Kg]

g : Gravitational acceleration [= 9.8m/s2 ]

A condition in which a work part does not drop when the work part is gripped statistically:

Fμ > W

F > mg μ

Necessary gripping force as the recommended safety factor of 2 in normal transportation:

F > mg x 2/ μ (safety factor)

When the friction coefficient μ is between 0.1 and 0.2:

F > mg/ (0.1~0.2) x 2 = (10~20) x mg

**OUR MODEL:**

T=1.6 kg/cm

R=10cm

Hence f= 0.16 N

**TOTAL LOAD ON JOINTS:**

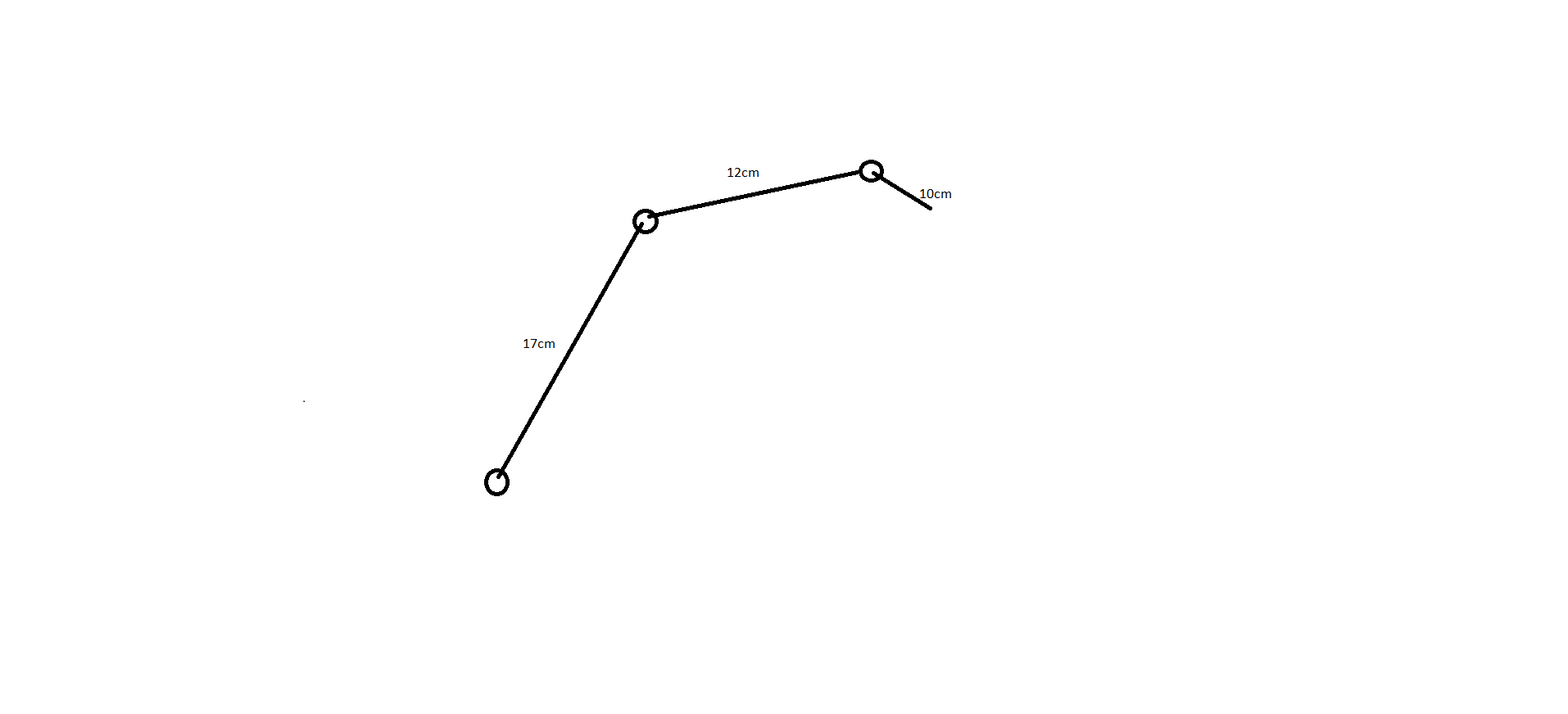
Torque on bottom joint can be calculated as follows

**τmax**=∑ mi⋅g⋅ri

=g ∑ mi⋅ri

=g⋅(ml⋅r1+m2⋅r2+m3⋅r3+mpayload⋅rpayload)

Similarly for each joint the torque can be calculated



**OUR MODEL:**

Torque when a box of dimesion 8 cm with a mass of 90 g can be calculated as follows

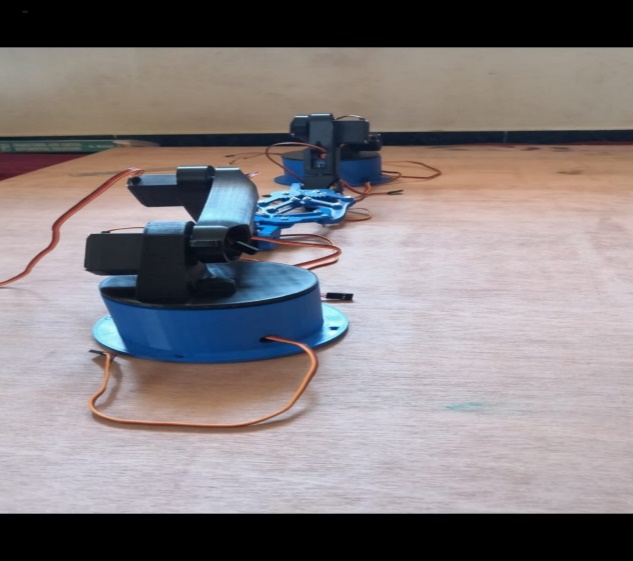
**Τmax =** 9.18(0.055Kg x 17cm+ 0.055kgx 12cm+ 0.027kg x 10cm+ 0.090kgx 8cm)

=9.995 kg/cm

Which is less than that of the stall torque of the motor ie 11kg/cm.Thus the system will easily be able hold an object of 90g mass.

We can also find the max value for payload using this formula.

**5.5 FINAL ASSEMBLY OF PROJECT**

**  
  
Fig 5.3** Final Project Model

**CHAPTER 6**

**BUDGET**

**CHAPTER 6**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 6.1** Budget Representation   |  |  |  | | --- | --- | --- | | **Materials Required** | **Quantity** | **Cost in (Rs.)** | | Controller | 1 | 1500 | | Potentiometer | 4 | 400 | | RFID Kit | 2 | 3000 | | Servo Motor | 4 | 4000 | | Battery | 1 | 800 | | 3D Printing | 1 | 7000 | | Fabrication cost | - | 5000 | | Arduino Kit | 1 | 1000 | | TOTAL = Twenty Two Thousand And Seven Hundred Rupees Only | * Rs.22,700/- | | |

**BUDGET**

**CHAPTER 7**

**MERITS AND DEMERITS**

**CHAPTER 7**

**MERITS AND DEMERITS**

**7.1 MERITS**

* Does not require any human efforts and responsibilities .
* Great economic and safety benefits .
* Failure of one arm does not stop the system from working. It only slows the process.
* Eliminates the risk of human errors.
* Remains competitive in a rapidly evolving industry
* Meets the demands of high work volumes .
* The right item is placed in the right workspace without any confusions or errors.

**7.2 DEMERITS**

* Complex design due to involvement of electronics, mechanical and programming parts.
* Even a minor error in the mechanical design can result in failure of the entire operation.
* Requires high knowledge of programming to ensure that all the six motors collaborate with each other for the efficient working of the arm.
* Maintenance cost can be higher due to the system complexity under part failure..
* Under power failure, the entire system fails if there is no back up. Then human operation would be necessary and should later update the data into the system.
* Even a small shift in the position of the arm or the storage space can cause improper working of the entire system.

**CHAPTER 8**

**APPLICATIONS**

**CHAPTER 8**

**APPLICATIONS**

* Applied in warehouse management systems .
* Can be used in many ecommerce storage applications.
* Can be used to function in Bio-Hazardous environments.
* Can be used to function under less human surveillance.

**CHAPTER 9**

**CONCLUSION**

**CHAPTER 9**

**CONCLUSION**

* Complex design due to involvement of electronics, mechanical and programming parts.
* Even a minor error in the mechanical design can result in failure of the entire operation.
* Requires high knowledge of programming to ensure that all the six motors collaborate with each other for the efficient working of the arm.
* Maintenance cost can be higher due to the system complexity under part failure..
* Under power failure, the entire system fails if there is no back up. Then human operation would be necessary and should later update the data into the system.
* Even a small shift in the position of the arm or the storage space can cause improper working of the entire system.

**CHAPTER 10**

**FUTURE DEVELOPEMENT**

**CHAPTER 10**

**FUTURE DEVELOPMENT**

* Addition of rails or wheels below the arm can help in achieving more complicated tasks.
* New innovative robotic arms with different degrees of freedom can also help in storage as well as manipulation.
* More complicated actions can be taught to the robotic arm by making use of its record and play option to do much more tasks.
* Better servo motors can be used for more load carriage and efficiency.
* Better processors can be added to increase the range of the whole system to do various other tasks .

**CHAPTER 11**

**REFERENCES**

**CHAPTER 11**

**REFERENCES**

1. Design Analysis of a Remote Controlled “Pick and Place” Robotic Vehicleby B.O. Omijeh [1] International Journal of Engineering Research and Development Volume 10, Issue 5 (May 2014)
2. Design and Operation of Synchronized Robotic Arm by GoldyKatal, SaahilGupt IJRET: International Journal of Research in Engineering and Technology .
3. R.B. Gillespie, J. E. Colgate, M. A. Peshkin, “A general framework for robot control”; IEEE Transactions on Robotics and Automation, 17,4, 391-401, 2001.
4. Modeling and Analysis of a 6 DOF Robotic Arm Manipulator Canadian Journal on Electrical and Electronics Engineering Vol. 3, No. 6, July 2012

**CHAPTER 12**

**APPENDIX**

**CHAPTER 12**

**APPENDIX**

#include <Servo.h> //Servo header file

//Declare object for 5 Servo Motors

Servo Servo\_0;

Servo Servo\_1;

Servo Servo\_3;

Servo Servo\_4;

Servo Servo\_5;

Servo Gripper;

//Global Variable Declaration

int S0\_pos, S1\_pos, S3\_pos,S5\_pos,S4\_pos, G\_pos;

int P\_S0\_pos, P\_S1\_pos, P\_S3\_pos,P\_S4\_pos,P\_S5\_pos, P\_G\_pos;

int C\_S0\_pos, C\_S1\_pos, C\_S3\_pos,C\_S4\_pos,C\_S5\_pos, C\_G\_pos;

int POT\_0,POT\_1,POT\_3,POT\_4,POT\_5,POT\_6;

int saved\_data[700]; //Array for saving recorded data

int array\_index=0;

char incoming = 0;

int action\_pos;

int action\_servo;

void setup() {

Serial.begin(9600); //Serial Monitor for Debugging

//Declare the pins to which the Servo Motors are connected to

Servo\_0.attach(8);

Servo\_1.attach(9);

Servo\_3.attach(10);

Servo\_4.attach(11);

Servo\_5.attach(12);

Gripper.attach(13);

//Write the servo motors to initial position

Servo\_0.write(90);

Servo\_1.write(90);

Servo\_3.write(90);

Servo\_4.write(90);

Servo\_5.write(90);

Gripper.write(90);

Serial.println("Press 'R' to Record and 'P' to play"); //Instruct the user

}

void Read\_POT()

//Function to read the Analog value form POT and map it to Servo value

{

POT\_0 = analogRead(A0); POT\_1 = analogRead(A1); POT\_3 = analogRead(A3); POT\_4 = analogRead(A4); POT\_5 = analogRead(A5);POT\_6 = analogRead(A6); //Read the Analog values form all five POT

S0\_pos = map(POT\_0,0,1024,10,170); //Map it for 1st Servo (Base motor)

S1\_pos = map(POT\_1,0,1024,10,170); //Map it for 2nd Servo (Hip motor)

S3\_pos = map(POT\_3,0,1024,10,170);

S4\_pos = map(POT\_4,0,1024,10,170);

S5\_pos = map(POT\_5,0,1024,10,170);//Map it for 4th Servo (Neck motor)

G\_pos = map(POT\_6,0,1024,10,170); //Map it for 5th Servo (Gripper motor)

}

void Record() //Function to Record the movements of the Robotic Arm

{

Read\_POT(); //Read the POT values for 1st time

//Save it in a variable to compare it later

P\_S0\_pos = S0\_pos;

P\_S1\_pos = S1\_pos;

P\_S3\_pos = S3\_pos;

P\_S4\_pos = S4\_pos;

P\_S5\_pos = S5\_pos;

P\_G\_pos = G\_pos;

Read\_POT(); //Read the POT value for 2nd time

if (P\_S0\_pos == S0\_pos) //If 1st and 2nd value are same

{

Servo\_0.write(S0\_pos); //Control the servo

if (C\_S0\_pos != S0\_pos) //If the POT has been turned

{

saved\_data[array\_index] = S0\_pos + 0; //Save the new position to the array. Zero is added for zeroth motor (for understading purpose)

array\_index++; //Increase the array index

}

C\_S0\_pos = S0\_pos; //Saved the previous value to check if the POT has been turned

}

//Similarly repeat for all 5 servo Motors

if (P\_S1\_pos == S1\_pos)

{

Servo\_1.write(S1\_pos);

if (C\_S1\_pos != S1\_pos)

{

saved\_data[array\_index] = S1\_pos + 1000; //1000 is added for 1st servo motor as differentiator

array\_index++;

}

C\_S1\_pos = S1\_pos;

}

if (P\_S3\_pos == S3\_pos)

{

Servo\_3.write(S3\_pos);

if (C\_S3\_pos != S3\_pos)

{

saved\_data[array\_index] = S3\_pos + 3000; //3000 is added for 3rd servo motor as differentiater

array\_index++;

}

C\_S3\_pos = S3\_pos;

}

if (P\_S4\_pos == S4\_pos)

{

Servo\_4.write(S4\_pos);

if (C\_S4\_pos != S4\_pos)

{

saved\_data[array\_index] = S4\_pos + 1000;

//1000 is added for 1st servo motor as differentiator

array\_index++;

}

C\_S4\_pos = S4\_pos;

}

if (P\_S5\_pos == S5\_pos)

{

Servo\_5.write(S5\_pos);

if (C\_S5\_pos != S5\_pos)

{

saved\_data[array\_index] = S5\_pos + 1000; //1000 is added for 1st servo motor as differentiator

array\_index++;

}

C\_S5\_pos = S5\_pos;

}

if (P\_G\_pos == G\_pos)

{

Gripper.write(G\_pos);

if (C\_G\_pos != G\_pos)

{

saved\_data[array\_index] = G\_pos + 4000;

//4000 is added for 4th servo motor as differentiator

array\_index++;

}

C\_G\_pos = G\_pos;

}

//Print the value for debugging

Serial.print(S0\_pos); Serial.print(" "); Serial.print(S1\_pos); Serial.print(" "); Serial.print(S3\_pos); Serial.print(" "); Serial.print(S4\_pos); Serial.print(" "); Serial.print(S5\_pos); Serial.print(" "); Serial.println(G\_pos);

Serial.print ("Index = "); Serial.println (array\_index);

delay(100);

}

void Play()

//Functon to play the recorded movements on the Robotic ARM

{

for (int Play\_action=0; Play\_action<array\_index; Play\_action++) //Navigate through every saved element in the array

{

action\_servo = saved\_data[Play\_action] / 1000; //The fist character of the array element is split for knowing the servo number

action\_pos = saved\_data[Play\_action] % 1000; //The last three characters of the array element is split to know the servo postion

switch(action\_servo){ //Check which servo motor should be controlled

case 0: //If zeroth motor

Servo\_0.write(action\_pos);

break;

case 1://If 1st motor

Servo\_1.write(action\_pos);

break;

case 3://If 3rd motor

Servo\_3.write(action\_pos);

break;

case 4://If 4th motor

Servo\_4.write(action\_pos);

break;

case 5://If 1st motor

Servo\_5.write(action\_pos);

break;

case 6://If 1st motor

Gripper.write(action\_pos);

break;

}

delay(50);

}

}

void loop() {

if (Serial.available() > 1) //If something is received from serial monitor

{

incoming = Serial.read();

if (incoming == 'R')

Serial.println("Robotic Arm Recording Started......");

if (incoming == 'P')

Serial.println("Playing Recorded sequence");

}

if (incoming == 'R') //If user has selected Record mode

Record();

if (incoming == 'P') //If user has selected Play Mode

Play();

}