5 DEGREE OF FREEDOM ROBOTIC ARM

MD. RASHED IQBAL-12321022
UMME AYSHA-12321028
TANJILA ROBYAT-12121179

DEPARTMENT OF ELECTRICAL AND ELECTRINIC ENGENEERING
APRIL, 2016

Inspiring Excellence
BRAC UNIVERSITY
DHAKA, BANGLADESH

5 DEGREE OF FREEDOM ROBOTIC ARM

A DISSERTATION SUBMITTED TO
DEPARTMENT OF ELECTRICAL AND ELECTRINIC ENGENEERING
OF
BRAC UNIVERSITY

BY
MD. RASHED IQBAL-12321022
UMME AYSHA-12321028
TANJILA ROBYAT-12121179

IN PARTIALFULFILLMENT OF THE REQUIREMENT FOR THE DEGREE
BACHELOR OF SCIENCE

ACKNOWLEDGEMENT

Firstly, we would like to thank our thesis adviser Dr. KhalilurRhaman, Associate Professor, Dept. of Computer Science and Engineering (CSE), BRAC University; for his supportive guidance and feedbacks towards our project for its completion. Secondly we would like to thank Md. Monir for his support in making and joining various parts in our project and others without whom our project would not be a successful one.

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of Dr. KhalilurRhaman, Department of Computer Science and Engineering, BRAC UNIVERSITY, Dhaka, Bangladesh. The work was spread over final year courses, EEE400: Thesis/Project, in accordance with the course curriculum of the Department for the Bachelor of Science in Electrical and Electronic Engineering program.

---------------------------------	-------------------------------	-------------------------------
MD. RASHED IQBAL	**UMME AYSHA**	**TANJILA ROBYAT**

Countersign by

Dr. KhalilurRhaman
Associate Professor
Department of Computer Science and Engineering
BRAC University

ABSTRACT

A robotic arm is one type of mechanical arm which is similar to a human arm and can do most of the works that a human arm can do. It can do various works which is very risky for a human. A robotic arm is programmable and can do the works through instructions. In today’s world robotic arms are used in industries, rescue purpose and various research works. Considering the facts we thought of designing and making some improvements in robotic arm. Our robotic arm is an arduino controlled six degree freedom robot arm which can move 360 degree and in 6 directions and can do almost all the works of gripping. First of all, the unique part of our robotic arm is, a counter weight is used here to balance the arm while some object is gripped. Secondly, it is able to move to a specific point of its workspace through x, y and z axis values automatically if the points are given. For giving command we are using Bluetooth serial communication system through an Application. This work presents a 6 degree robotic arm with a gripper, controlled with an android application and Arduino MEGA via Bluetooth to carry or load materials**.** Here, the app searches for the Bluetooth connection, if the Bluetooth device connected with robot is open for connection, then the android app connects with it. Then the app sends command to the Arduino MEGA connected with the robot which is the brain of the robot, this command is fetched by the Arduino MEGA and according to the command it moves the robot forward, backward, left turn, right turn, stop and it expands the gripper to pick a material, squeezes the gripper when a material is picked and it can also move the gripper in left or right direction according to need.

TABLE OF CONTENT

TABLE OF FIGURE CONTENT PAGE

CHAPTER1: MOTIVATION

While the field of robotics is continuously expanding at a remarkable rate and better performing robots are created every year, robotics still remains out of reach for many students and researchers. The main reasons for this difficulty are the high complexity of the hardware and software of robots, and their typically high cost. We believe that the computing power, sensing capabilities and intuitive programming interfaces of modern smart phones afford an inexpensive yet highly capable robotic platform. Smartphone based robots are becoming increasingly popular, with many exciting applications emerging in both academia and industry. As a case in point, we provide a detailed description of a simple robotic platform based on this approach.

The Android Mobile Phone Platform by Google becomes more and more popular among software developers, because of its powerful capabilities and open architecture. As it’s based on the java programming language, its ideal lecture content of specialized computer science courses or applicable to student projects. We think it is a great platform for a robotic system control, as it provides plenty of resources and already integrates a lot of sensors. The java language makes the system very attractive to apply state-of-the-art software engineering techniques, which is our main research topic. The unsolved issue is to make the android device interoperate with the remaining parts of the robot: actuators, specialized sensors and maybe coprocessors.

Using an Android device to stream data and control the robot is an opportunity to focus on fundamental research and the problems occurred during the development phase while the time spent to embed different technologies to replace the smart phone functions is now very small.

With a modular platform and flexible enough to support a wide range of sensors that can be incorporated and relocated very easily, you can use the same robot for different applications of indoor and outdoor terrains.

An important advantage of using a smart phone for an onboard computer is that the size of a robot can be kept relatively small, yet still have great features. Its cost can also be minimal since the phone itself can handle computation, sensing and battery power. Many different phones are now available on the market. Before purchasing an Android phone to be used as an onboard computer for a robot, one has to consider the uses and needs of that particular robot.

Smartphone, a small yet powerful device is rapidly changing the traditional ways of human machine interaction. Modern smart phones are embedded with accelerometer sensor, Bluetooth module and are powered by different operating systems such as Symbian, Bada, Android, OS etc. Among all available mobile operating systems Android OS has gained significant popularity after being launched in 2008, overtaking all previous competitors due to its open architecture. Android. Researchers from around the world have shown keen interest in gesture technology and its possibilities in various fields making it a powerful tool for humans. Smartphones have proved to be of much more aid than being a device just for making calls. The large world is merging together into the palms of humans in the form of a smartphone. (A. Valera, December 12-15, 2005.)

The motivation towards working on this research and development program was originated from the view that a lot of students are making projects on robotic arm but our main point was to make a rescue robot which will be able to defuse a bomb by picking it and putting it in some place where the bomb cannot harm people. Then again, our project is not limited to just picking and putting it another place it will be able to reach any point if the values of x, y and z axis are given. Again to make it more unique we added a counter eight at the back portion. (Hibbeler, 2012).

Platform has revolutionized the application development field for cellphone, opening new doors for technical exploration. The smartphone can be freely rotated in space, temporarily varying 3-dimensional signal data is obtained from the phone's 3-axis accelerationsensor. This data is transmitted to a robot via Bluetooth module of smartphone using an android app. Further, it is processed by a microcontroller embedded on the robot for its desirable motions. In this context, a robot is an analogy for any machine that is controlled by man varying from a simple toy to heavy machinery. Robots have evenreplaced humans in performing various tasks that they are unable to perform due to physical disability, size limitation or extreme environments. For past two decades their interactions.

.

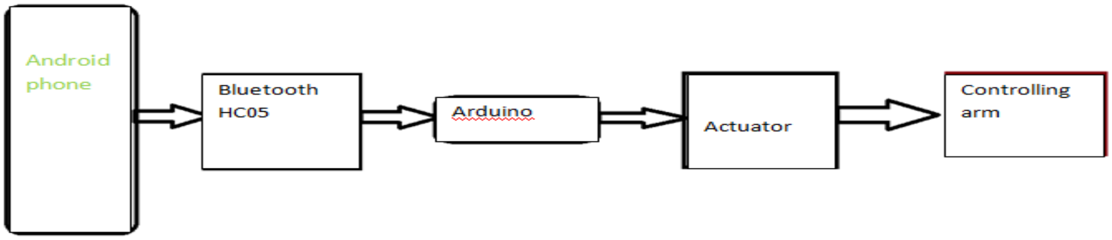


Figure 1.1: Main components of the Android based 5 degree freedom robotic arm platform and

Here, the robotic platform gets sensory input from the phone’s internal sensors, as well as external sensors via the Arduino Mega. The Android phone sends commands to the robot’s actuators via the Arduino Mega. The Android phone interacts with the Arduino Mega through a wireless Bluetooth connection.

CHAPTER 2: INTRODUCTION

2.1: Introduction to the Robotic Arms:

The end effector, or robotic hand, 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 lines perform a variety of tasks such as welding and parts rotation and placement during assembly. In some circumstances, close emulation of the human hand is desired, as in robots designed to conduct bomb disarmament and disposal.

As various researches are taking place there is a lot of variety in robotic arm. A few types of robotic arms are given here to have a small overview.

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

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

Spherical Robot / Polar Robot (such as the Unimate): Used for handling at machine tools, spot welding, die-casting, fettling machines, gas welding and arc welding. It's a robot whose axes form a polar coordinate system.

SCARA robot: Used for pick and place work, application of sealant, assembly operations and handling machine tools. This robot features two parallel rotary joints to provide compliance in a plane.

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

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

Anthropomorphic Robot: Similar to the robotic hand Luke Skywalker receives at the end of The Empire Strikes Back. It is shaped in a way that resembles a human hand, i.e. with independent fingers and thumbs.

2.2 Impact of Robotic Arms in the Economy

There have been many discussions regarding Americans losing jobs to overseas manufacturers and to increased use of robotic automation, and with fair wage employment on the rise in many other countries, there has also been an increase in the use of robots there. Amid these discussions and trends, recent studies show that, in fact, industrial robotics actually positively impacts jobs and the economy.

Creating New Jobs: Between2008 and 2011, an estimated 500,000 to 750,000 new jobs were created because of the robotics industries. This growth is attributed to the increased level of productivity in manufacturing companies. While the robotic solution has previously displaced a number of workers, the increased efficiency also led to increased productivity. Historically, this improved output has led to lower prices and a greater demand for products with more circulation. Entirely new market segments have resulted as more people work together to deliver products to consumers.

As these automated systems continue to provide value in the manufacturing industry, the demand for more, newer, and better systems continues to rise. This, in turn, increases the demand for qualified people in the robotics industry. At the same time, these manufacturing companies require people to manage, maintain, and program the new material handling robots, conveyor systems, as well as other automated solutions.

By some estimates, the robotics industry on its own generates around 170,000 jobs worldwide. The operators and technicians who use and deploy these robots account for tens of thousands more, and those figures don’t include jobs that are indirectly created due to increased productivity.

The indirect jobs that benefit from robotic automation include everything from retailers to distributors – conceivably millions of jobs. There are also places and industries that simply could not exist if they didn’t have the right robotic toolsbut as we continue to develop effective solutions, the job market can continue to evolve and grow.

2.3 Impact of Robotic Arms in the Economy of Bangladesh

In a developing country like Bangladesh the use of robotics or embedded system is not that much. But now a day's the use of embedded system is increasing day by day in offices, irrigation and the industries. On the other hand the direct use of robotics is not that much popular in the economic aspects. If it happens the level of Bangladesh in the economy will also be increased. On the other hand if the use of direct robotics is increased, may people will be jobless and unemployment problem will be increased as well as there will be some new job opportunities like robot controlling. That is how the economy will be affected by the use of robotics.

2.4 History of Existing Arms

In 1961 the first industrial robot, Unimate, joined the assembly line at a General Motors plant to work with heated die-casting machines. Unimate took die castings from machines and performed welding on auto bodies; tasks that are unpleasant for people. Obeying step-by-step commands stored on a magnetic drum, the 4,000-pound arm is versatile enough to perform a variety of tasks. An industry was spawned and a variety of other tasks were also performed by robots, such as loading and unloading machine tools. Unimate industrial robots are among the most widely used industrial robots in the world. With over 20 years of continued improvement they are highly reliable, easy-to-use robots. The UNIMATE robots feature up to six fully programmable axes of motion and are designed for high-speed handling of parts weighing up to 500 lbs. The dedicated electronic control is regarded as one of the simplest controllers available in the industry today for teaching and operating industrial robots. Unimate was conceived in 1956 at a meeting between inventors George Devol and Joseph Engelberger, where they discussed the writings of science fiction. Together they made a serious commitment to develop a real, working robot.

Fig: 2.1 previous Robotic Arm

2.5 Introducing Smart phone in Robotics

A person may have many options for a robot brain like an 8051 microcontroller, Arduino, Raspberry Pi. But all of these robotic brains must be equipped with memory, sensors, batteries, communication boards, and many more functions to be closed by the functionality of a smart phone.

A mobile device like a smart phone provide hardware and software resources including sensing capabilities, communication options, a smart device is able to run several applications and has intuitive programming applications. Using a laptop computer you can communicate with the robot through wireless application installed on the smartphone, while streaming video/images or receiving a huge amount of information it becomes a simple task.

A smart phone is used for control, communication, streaming video/images, and is powered by a Linux-based operating system.

Regarding the specifications of a smart device, there is a remarkable list with powerful processors with two/four/.. cores, graphical processing units, high definition video cameras, high definition microphones, a set of sensors (such as GPS, accelerometer), a long life battery, a wide range of communication options (3G, 4G, Bluetooth, Wi-Fi), run modern operating systems (Android, Windows, iOS), and you can build an application in minutes using an intuitive Software Development Kit (SDK).

2.6 Project Overview

The key concept of this project is an auto moving robotic arm with counter balance through one instruction. The arm will reach in one point when the instruction is given and will pick something and keep it in another point. It is something like pick and place automatically through one instruction.

At the beginning of our project we were just given the base in which the arm will be placed. This was used in previous project works. Then we made a solid works software design using that base design through which we need not do the calculations of trial and error. The calculations were done in the software. The whole structure was done through the software.

The second work was to calculate the counter weight to balance the arm when it picks some load. The calculations were done by reading books and taking help from the internet. The counter weight was placed at the back side of the arm.

The third phase was the serial communication and coding of the Arduino according to our planned project. It was the main part as we were about to make the movement automatic through one instruction to pick and place some objects.

The fourth phase was connecting all the things together in one project. Making the circuits and connect all the actuator, arduino with motor controller and also connect the Bluetooth module for serial communication with the app and Arduino code.

The final phase was to make it work properly through testing the connection. It was important to check whether it works properly or not.

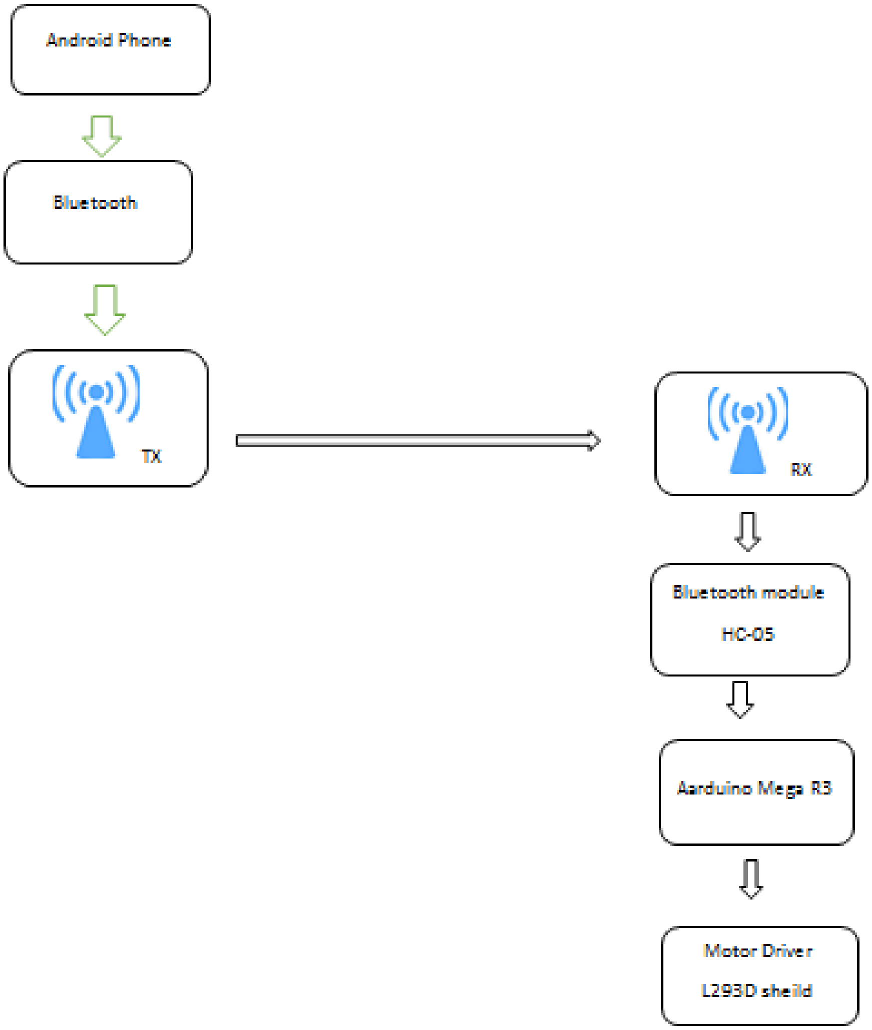


Fig 2.2; Overall Block Diagram

CHAPTER 3: METHODOLOGY

3.1 What Is Methodology

This chapter is focused on the methodology process which is the sets of methods to fabricate and design that been used. The design that been implement are need to specify certain criteria to achieve project objectives. The information in the literature review is interpreted to select the suitable design for the Robotic Arm. The design need to be specified in CAD software model. The steps that involve in fabricating and programming of the model are stated.

3.2 Flow Chart of Methodology

To achieve the objectives, the methodology has been constructed (Figure 3.1). The function of the flow chart is to give guideline and direction to accomplish the main goal of the project. The following paragraph is the summary of the flow chart.

First of all, the project start with the review about related topic on Robotic Arm and Automatic Guided Conveyor. After reviewing on related topic, the design of Robotic Arm is sketched on the paperwork. If the design characteristic is still not good, the sketching process will be continue until meet the desire design of Robotic Arm. Then, based on the design sketch, the CAD MODEL of Robotic arm using solidworks is made. The final process are fabricating and finishing.

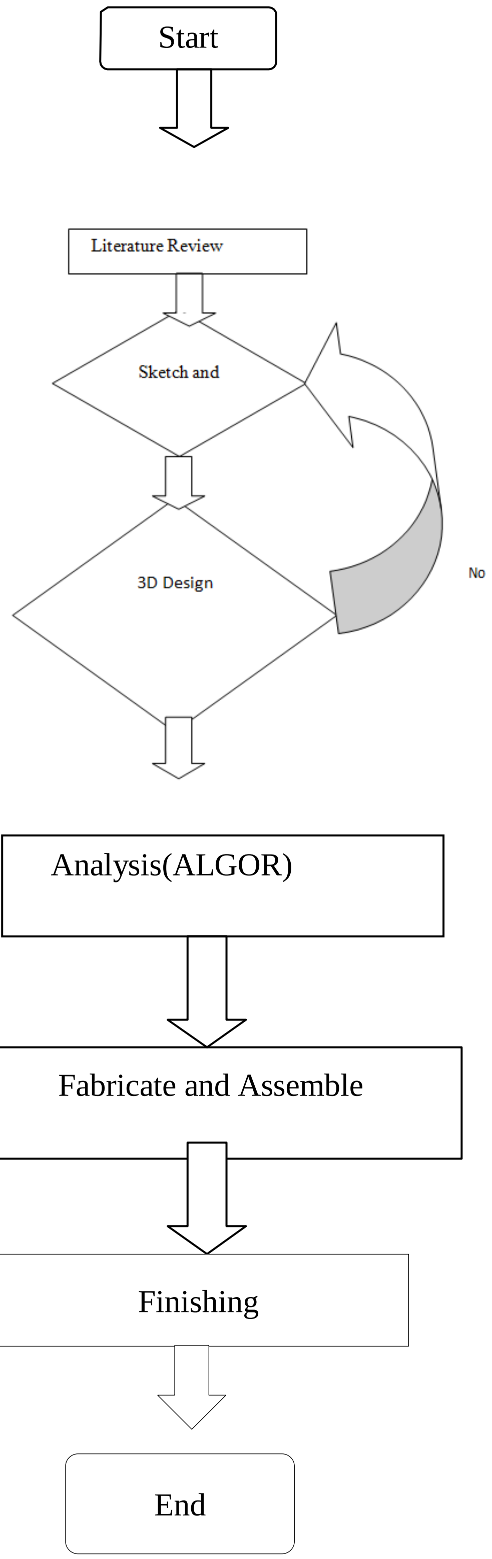


Fig 3.1, Flow Chart

3.3 Proposed Robotic Arm Design

There are several consider at ion should be known when designing the Robotic Arm including mechanical parts and electrical components to sensor technology ,computer programming system and artificial technology that influenced the overall robotic arm performance. The mechanical and electrical components include grippers and the body of the robot, while the electrical components consist of microcontroller or servocontroller,voltage regulator, and sensing system. The overall design of robotic arm can be categorized into two parts, which are the mechanical design and the electrical components. The mechanical design such as gripper and the robotic arm or body must be designed as accurate as possible to keep away from any problem when the robot starts its movement. The electrical components also must be chosen wisely to make sure the electronic can be performing perfectly and easy to attach to mechanical part of the robotic arm. The body must be designed as accurate as possible to keep away from any problem when the robot starts its movement. The electrical components also must be chosen wisely to make sure the electronic can be performing perfectly and easy to attach to mechanical part of the robotic arm robot can move smoothly or can operate. The mechanical parts is designed by choosing aluminum because it strong and light material compare to Perspex or other material. Aluminum is difficult to break due to its quality. On the other hand, Aluminum material al so easily of in and the price is cheap.

3.4 Drawing Consideration

The drawing of the Robotic Arm must achieve the entire desired objective. The drawings must be complete as fast as possible and then it can be fabricated perfectly without making any mistake in the middle of the fabrication process. The consideration for the drawing is for it firstly to be moveable, in sketching; the measurement of the robotic arm link length must be taken to get the suitable size for the robotic arm. This is to make sure the loading and unloading mechanism can lift and move the load. The robotic arm depends on degree of freedom angle or translation limitation, and the arm link, the angle. The configuration of the robotic arm should be considered. Besides that, all the motors used for the robotic arm must achieve the required motor torque.

In this project, the software used to draw the model is Solid Works. Solid Works software is mechanical design automation as a computer added software (CAD). This software was chosen as it is possible to sketch idea of the design, experiment with features and dimensions and produce models. Solid works able to draw 3D design based on the component. There are some steps that must be follow which is:

(i). **Sketches:** Draw some design sketches, dimensioning, where to apply the objectives needed and so on

(ii). **Features:** Select the appropriate features; determine the best features to be applied and so on.

(iii). **Assemblies:** Select the components to be mate, what types of mate to apply in the drawing and so on. Mate is the mating process in Solidworks.

After the overall design finish, the next step which is fabricating process can be continuing. All the components to be fabricated must be label first to avoid repetition process.

3.5 Robotic Arm Design

Robotic arm design is the mechanical component sketch before fabricating process. Overall design of the robotic arm use simple arm link, base and linear actuator. The important thing about the robotic arm design is the dimension. The design must achieve the required dimension to match with the linear actuator torque.

Basically, mechanical part of robotic arm can be bought at fabricating company which allowing any design from Solidworks to be fabricated. Figure 3.2 show the dimension of the linear actuator. Figure 3.3 show the dimension of counter weight.

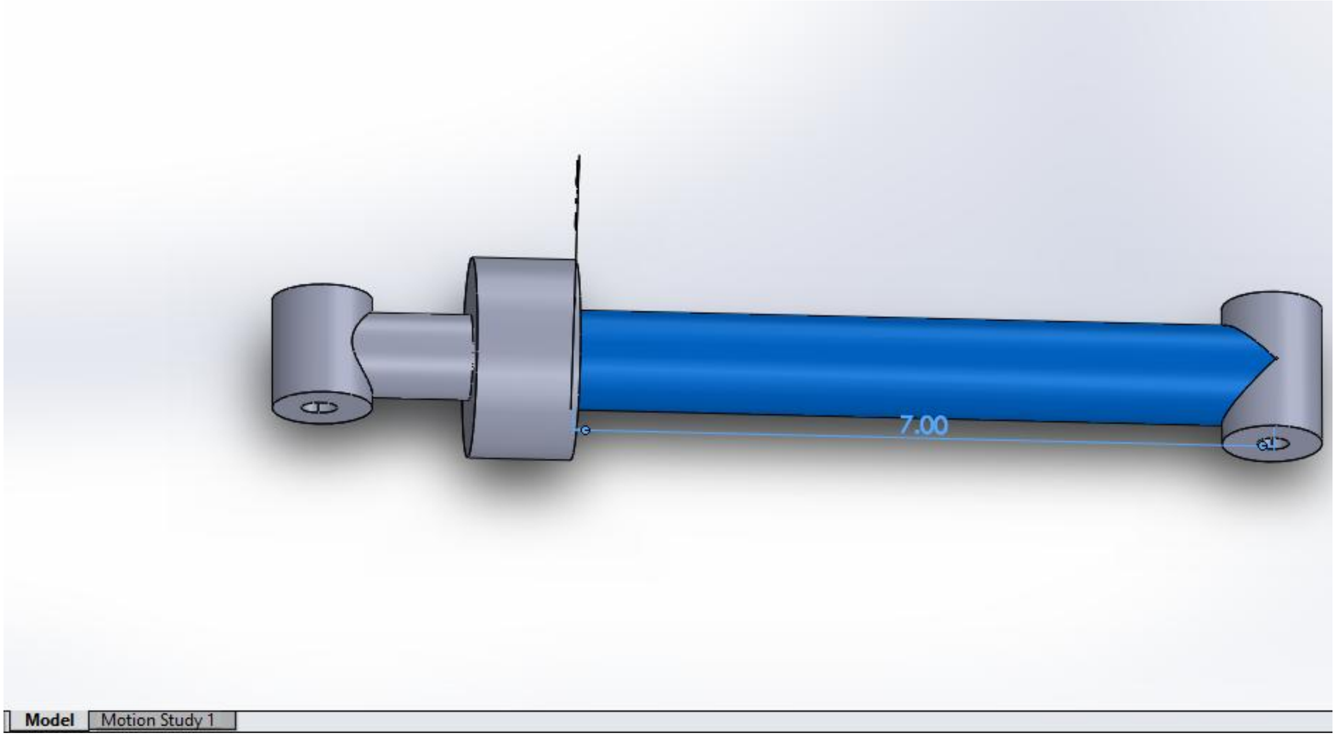


Fig 3.2, Linear Actuator (Solid works)

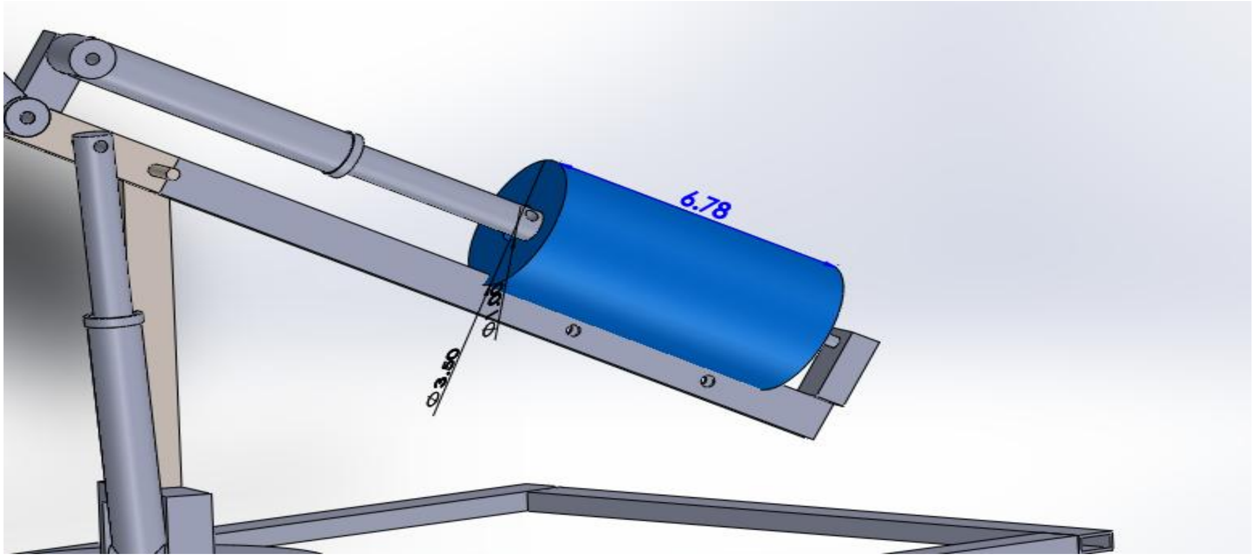


Fig 3.3, Joint and Dimension of Counter Weight (Solid Works)

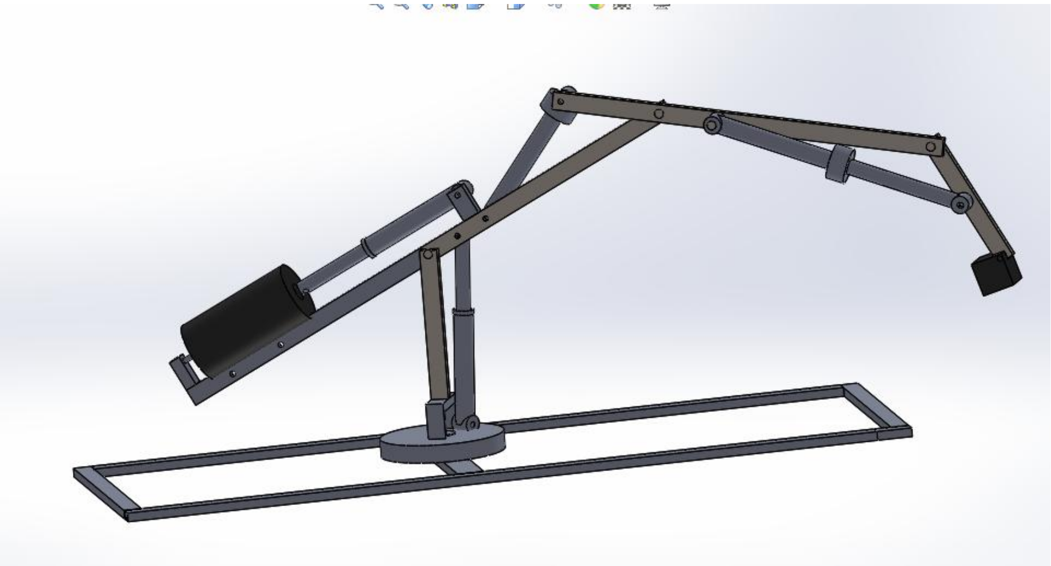


Fig3.4, Overall Model of Arm (Solid Works)

3.6 Robotic Arm Degree of Freedom

Robotic arm includes a drive assembly and an articulated arm assembly pivotally connected to the drive assembly. The articulated arm includes a pivoting base link system, a wrist link system, and a first elbow link system rotatable connected to the base link system by a pair of upper arms and connected to the wrist link system by a pair of forearms, a second elbow link system rotatable connected to the base link system by another at least one upper arm and connected to the wrist link system by another at least one forearm, wherein the drive assembly is connected to at least one of the upper arms and the base link system to provide three degrees of freedom by driving the at least one of the upper arms and pivoting the pivoting base link system to position the wrist link system at a given location with a predetermined skew relative to an axis of translation.

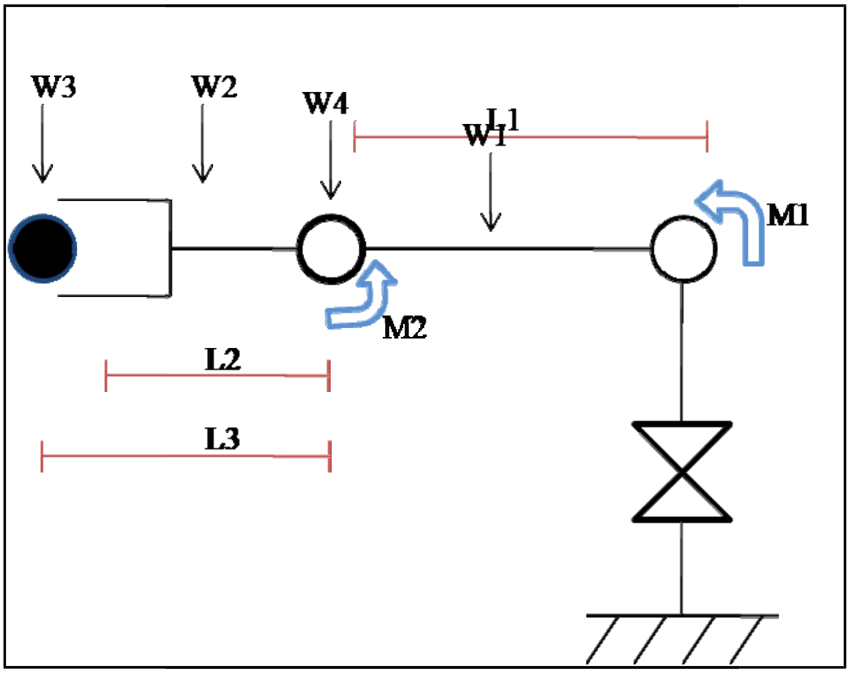


Fig 3.5 Force Calculation of Joints

Based on Figure 3.5, the desire force calculation of joint at can be calculated by using moment of Inertia formula.

(i). Torque about Joint 1

$$M1 = (L1/2)*(W1) + L1*(W4) + (L1+L2/2)*(W2) + (L1+L3)*W3 \qquad (3.1)$$

(ii). Torque about Joint 2

M2= (L2/2)*W2+ (L3)*(W3)

(3.2)

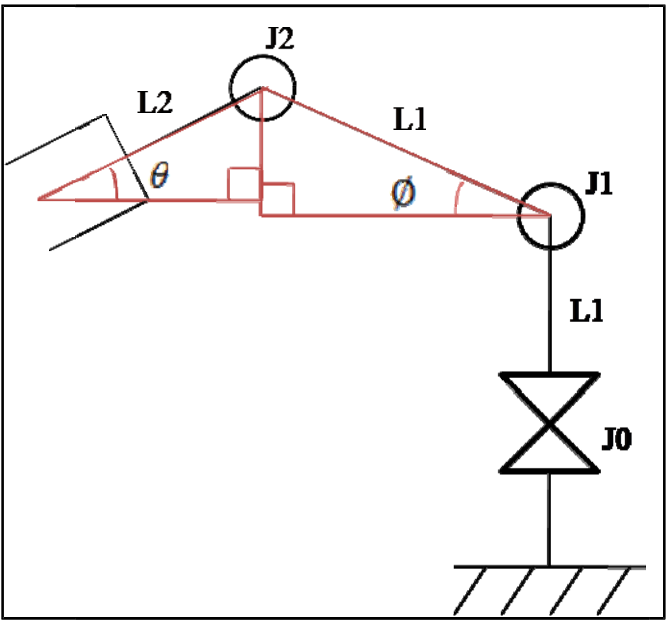


Fig 3.6; Forward kinematics

Based on figure 3.5, forward kinematic of the robotic arm can be calculated. Forward kinematics is the method to solve the orientation and position of the end effectors, given the joint angles and link lengths of the robotic arm.To Calculate Forward Kinematics, the Knowledge about geometry and algebra can be refer to figure 3.6, assume the base located at x = 0 and y = 0.The first Step is to locate x and y of each point.

Joint 0 (with x and y at base equaling zero):

$$X_o = 0 \tag{3.3}$$

$$Y_o = 0 \tag{3.4}$$

Joint 1 (with x and y J1 equaling 0):

$$\cos\Phi = x1/L1 \tag{3.5}$$

$$x1=L1\ (\mathrm{Cos}\Phi) \quad (3.6)$$

$$\mathrm{Sin}\Phi = y1/L1 \quad (3.7)$$

$$y1= L1\ (\mathrm{Sin}\Phi) \quad (3.8)$$

Joint 2 (with x and y at J2 equaling 0):

$$\mathrm{Sin}\theta = x2/L2 \quad (3.9)$$

$$x2 = =L2(\mathrm{Sin}\theta) \quad (3.10)$$

$$\mathrm{Cos}\theta = y2/L2 \quad (3.11)$$

$$y2 = L2(\mathrm{Cos}\theta) \quad (3.12)$$

End Effectors Location:

$$Xo+x1+x2 \quad (3.13)$$

$$Yo+y1+y2 \quad (3.14)$$

Z equals Alpha in Cylindrical Co-ordinates.

3.7 Analysis

The purpose of linear static stress analysis of robotic is to determine the displacement and stress resulting under loads representing normal operating conditions. The analysis of linear static stress will be done by using. ALGOR software is a general- purpose metaphysics finite element analysis software package develops by ALGOR. After analysis is completed using the software, we will got the information if the aluminum profile which is used to carry the object is strong enough or not. The capabilities of robotic arm are rely on these results whether the element can carry on the project or to withstand the object to be loaded.

3.8 Testing Process

To assemble mechanical components of this robotic arm, the method of fabricating parts must be sure first because the part to be assembled has different dimensions and usage. Gripper is used to hold an object. The gripper is assembling along with the linear actuator using coupling to give the movement. The suitable lift mechanism is important in robotic arm because it’s function as an affecter to hold the object. The methods that have been used to lift the object are robotic arm mechanism which gripper functions as the holder.

Besides assembling mechanical components, electric components used for this project must be considered. There are many electronic for robotic arm out there such as motor controller, regulator, Aarduino and computer. Motor controller used to control many servos simultaneously by using computer as a host. Host of the servo controller can be computer or microcontroller. For this project, computer functions as the operation system to control all motors using software provided by the manufacturer.

3.9 Expected outcome

The expected outcome after the project is completed, Robotic arm able to:

i. Load and unload the object
ii. The movement of the robotic arm
iii. Electric components decided

CHAPTER 4: BACKGROUND STUDY

4.1Existing Robotic Platforms for Education and Research

This paper presents a mobile robot laboratory. The laboratory is based on the LEGO Mind storms programmed with Legos. With this environment a wide variety of robot activities can be developed due to its flexibility, power, and simple use.

For the communication between the computer and the LEGO controller, the standard solution is the use of an infrared tower. With this kind of communication, several problems can occur because long distances between the emitter and the receiver are not allowed. In addition, the communication can fail if there is some obstacle between them. In order to avoid these limitations, the paper proposes a new communication system based on a Bluetooth chip.

In 1998 LEGO released the first Mindstorms set, the Robotics Invention System (RIS 1.0). It is an educational toy for children aged 12 years and older. Apart from those familiar beams, bricks and gears, the kit contains dc motor actuators, a range of sensors, and, most importantly, the RCX component. The RCX is the LEGO's programmable brick that allowed models not just to move, but to sense and respond to their environment. It is based on a Hitachi H8 series microprocessor. This 8-bit CPU provides most of the control logic for the RCX, including serial I/O, Analog to Digital Converter and built–in timers. It even contains 16KB of internal ROM and 33KB of static RAM. In addition, there is an interface to three actuators and three sensors, an infrared (IR) communications interface (to communicate with a desktop computer or another RXC) as well as an LCD, four pushbuttons and a small speaker.

The RCX component was initially developed as an educational tool through the collaboration of LEGO and MIT. On the first version of the RCX allowed six input and six output blocks to be connected to the H8 microprocessor. When LEGO developed the commercial version of the RCX, the number of inputs and outputs was reduced to three of each. Although this change reduced the flexibility of the RCX utilization, it reduced the drain of the batteries that power the system.(A. Valera, December 12-15, 2005.)

The components of the LEGO Mindstorms Robotic Invention System are described in detail for this article, components are categorized under the headings of sensors, actuators, and information technology.

The initial intended use of the RCX system was for research and educational activities. Combining the versatile LEGO construction blocks with the easy-to-use programming and I/O interfacing of the RCX provided a fast prototyping system to support these activities.

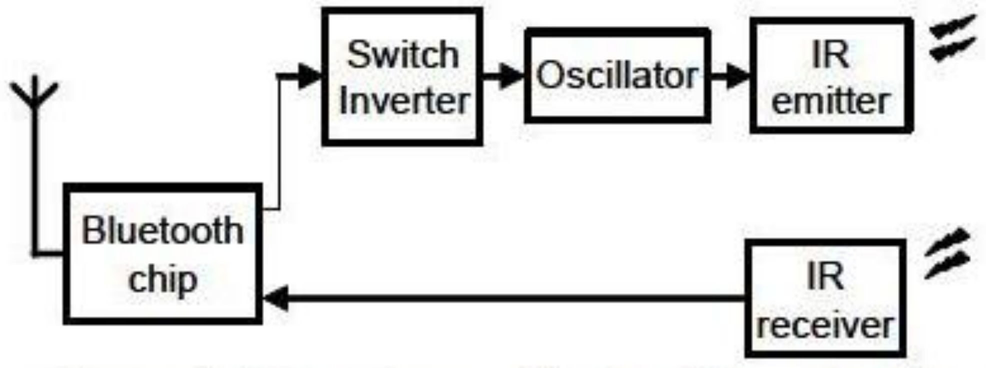


Figure 4.1: Communication system architecture. The system has the Bluetooth chip and the emitter and the receiver stages.

Although the commercialization of this product has focused on the recreational and K-12 educational markets, the flexible and expanding world of LEGO Mindstorms is widely accepted as a tool for research and higher education.

In particular, the kit is relatively cheap, robust, reconfigurable, reprogrammable, and induces enthusiasm and innovation in students.

Drawbacks of this system

- With this kind of communication, several problems can occur because long distances between the emitter and the receiver are not allowed. In addition, the communication can fail if there is some obstacle between them.

4.2 Towards intelligent robot with smartphones

Mobile phones are one of the top selling mobile devices in the world. Due to their large production, their prices have a high cost/benefit ratio. Current smartphones have a variety of built-in sensors that can be explored to build robots. For example, many of them have accelerometer, camera, Wi-Fi, Bluetooth, battery, speakers, microphone, Global Positioning System (GPS) receiver and compass. Some have even stereo camera for 3D imaging and gyroscopes.

Most robot designers use microcontrollers or computers in order to implement the robot’s control algorithms and off the shelf sensors are purchased to build the robot’s sensing systems. Buying all the sensors typically embedded in a mobile phone would certainly be much more expensive than buying a new and powerful smartphone. Moreover, current smart- phones are powered with processors faster than 1GHz, often multi-core and 1GB of RAM memory or more. Many net-books don’t have such rich configuration.

One common use for phones and other mobile devices is as robots' remote controllers, such as described by Naskaretal.In the past few years, many projects were created aiming to use smartphones as the robot's main control unit. In fact, mobile phones with 300MHz processors are already capable of executing complex robotics algorithms such as the Extended Kalman Filter, particle filter and potential fields with soft real time constraints. Most solutions consist in connecting the mobile phone to a microcontroller in the robot using serial, USB or Bluetooth connections. These solutions, although flexible, need microcontrollers, making the project more expensive and complex.

In order to build simpler yet powerful robots, several authors proposed the use of a universal interface present in any mobile device: the audio channel (accessible via head- phones connectors). With such approaches, the mobile phone can directly control the robot's motors without any intermediate processing. The problem with these approaches is that only open loop control is possible.

Using a smartphone as the "brain" of a robot is already an active research field with several open opportunities and promising possibilities. In this paper they present a review of current robots controlled by mobile phones and discuss a closed loop control mechanism that they have developed to control mechatronics systems using audio channels of mobile devices, such as phones and tablet computers. In our work, actuators commands are sent via audio and sensors reading are received by the phone also via audio using only analog electronics and no intermediate processing units.

Although they have already proposed such closed loop control technique elsewhere, they have now added the mobile device’s compass, commonly a 3-axis magnetometer, in the control loop so that a mobile robot would be able walk with accurate heading without the use of wheels’ encoders information, or using single wheel odometry. Unfortunately, they have found that the motors cause magnetic field distortions, leading to incorrect heading measurements done by the compass, especially because the mobile phone is placed near the motors. One alternative solution is using mobile phones with gyroscopes.

Figure 4.3: Robot architecture

The purpose of our research is to provide simpler robot’s hardware architectures but with powerful computational platforms so that robots' designers can focus on their research and tests instead of sensors and actuators connection infrastructure. These simple architectures are also useful for educational robotics, because students can build their own robots with low cost and use them as platforms for experiments in several courses.

Drawbacks of this System

- The device must be attached with the vehicle always which is not a very efficient design.
- Two or more smartphone maybe needed.

CHAPTER 5: PROPOSED OVERALL DESIGN

5.1 Objectives

The goal of our project was to build a 6 degree freedom robotic arm with a gripper to hold or grab materials brained by Arduino MEGA using a Bluetooth module (HC -05) as well as an android application to control the kinetics of that robot and the movement of the gripper through any android device and to minimize both expenses and time spent on building robots, allowing users to focus on more fundamental research and robotic problems. The platform also had to be modular and flexible enough to support different sensors and actuators that could be incorporated and relocated very easily. The process involved in building the robot includes the assembling of a chassis used for the robot and programming the Arduino MEGA as well as the interface for the android device.

We’ve also analyzed on android platform and its communication method with Arduino and Bluetooth Module (HC-05).We also learned how to develop efficient android application that can communicate with hardware and provide command via Bluetooth.

We’ve also studied Arduino microcontroller and its interfacing with HC-05, Servos, glass motor, motor controller and android platform.

The outcome of the project is a combination of embedded computing and Programming.

We believe that three main off-the-shelf components can be used in order to fulfill these requirements:

1. A smart phone running the Android operating system used as operating an robotic arm with 3D space rotation within their working space.

2. An electronic board (Arduino MEGA) used to interact with peripheral devices such as servos, sensors and motors not included in the phone.
3. It may move freely automatic system within their workspace by controlling co-ordinate system like x, y, z axis.

Due to the variability in complexity of these components, the total cost to build such a robotic platform can change, especially depending on the phone and vehicle used.

This Project delivers an arm demonstration that can move forward, left, right, backward, up and down or carry materials commanded via an Android Device.

In the following sections, we will describe a modular and flexible robotic platform that can be built for approximately $150 (excluding the phone).

5.2 Hardware Overview

In this section we will describe each of the equipment that we've used in our project and how to setup Arduino and it's interfacing with Servos, L293D, DC motors, HC-05 etc.

5.2.1 Aarduino MEGA

It is the brain of the project. It controls the whole robot and the gripper.

The Arduino software is free and open source. The programming platform is based on the popular Wiring language. The IDE is based on Processing, which is a well-known language among designers and prototypes. Unlike most microcontroller interfaces, Arduino is cross-platform; it can be run on Windows, Linux and Macintosh OS X. Because the language is based on well-used frameworks, Arduino can interact with other software on the computer like Flash or even web APIs like Twitter.

The project Arduino first began in 2005 at Interaction Design Institute Ivrea (IDII) but the dawn of Arduino began in year 2002 when Massimo Banzi (Massimo Banzi 2012) co-founder of Arduino was appointed as an associate professor to teach the students of IDII to promote modern ways of interactive design. (David Kushner 2011) Banzi wanted to offer his students something modern and inexpensive so everybody could carry their works without many obstacles.

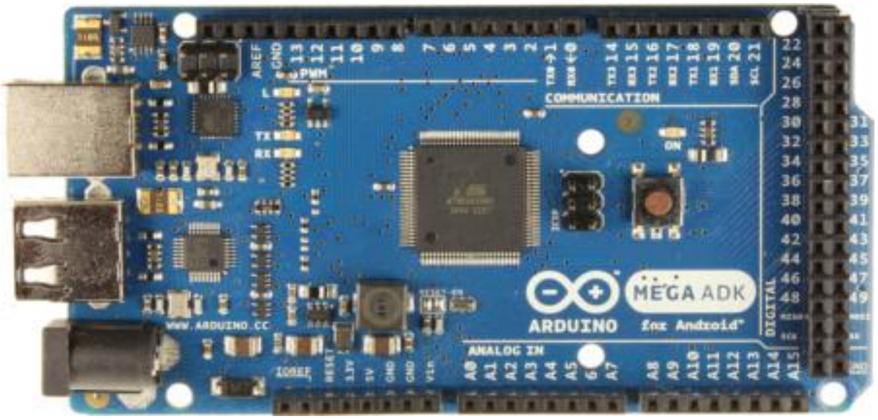


Figure 5.1: Arduino MEGA ADK

By then, the most used tool in the market was BASIC Stamp (Parallax 2012), which was expensive. So as an alternative Banzi wanted to develop something better. Banzi was also involved in processing (Processing 2012), the processing language. So with the help of a Colombian student Hernando Barragán (Barragan Studio 2012) who was working on a wiring (Wiring 2012) platform, they tried to make processing for hardware and make it simpler and easier to use after working on the project. They came up with a prototype, which was the birth of Aarduino.

The Arduino Mega is a microcontroller board based on the ATmega1280. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

5.2.1.1 Summary of Aarduino MEGA

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
USB Host Chip	MAX3421E

5.2.1.2 Memory of Aarduino MEGA

The MEGA has 256 KB of flash memory for storing code (of which 8 KB is used for the boot loader), 8 KB of SRAM and 4 KB of EEPROM5

5.2.1.3 Power of Aarduino MEGA

The Arduino MEGA can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 5.5 to 16 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

5.2.1.4 Input and Output of Arduino MEGA

Each of the 50 digital pins on the MEGA can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 Kohms. In addition, some pins have specialized functions:

5.2.1.5 Communication of Aarduino MEGA

The Arduino MEGA has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software 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 ATmega8U2/16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the MEGA digital pins.

The ATmega2560 also supports TWI and SPI communication. The Arduino software includes a Wire library to simplify use of the TWI bus; see the Wire library for details. For SPI communication, use the SPI library.

The USB host interface given by MAX3421E IC allows the Arduino MEGA ADK to connect and interact to any type of device that have a USB port. For example, allows you to interact with many types of phones, controlling Canon cameras, interfacing with keyboard, mouse and games controllers as Wiimote and PS3.

5.2.1.6 USB Overcurrent Protection of Arduino MEGA

The Arduino MEGA ADK has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

5.2.1.7 Pin Mapping of Arduino Mega ADK

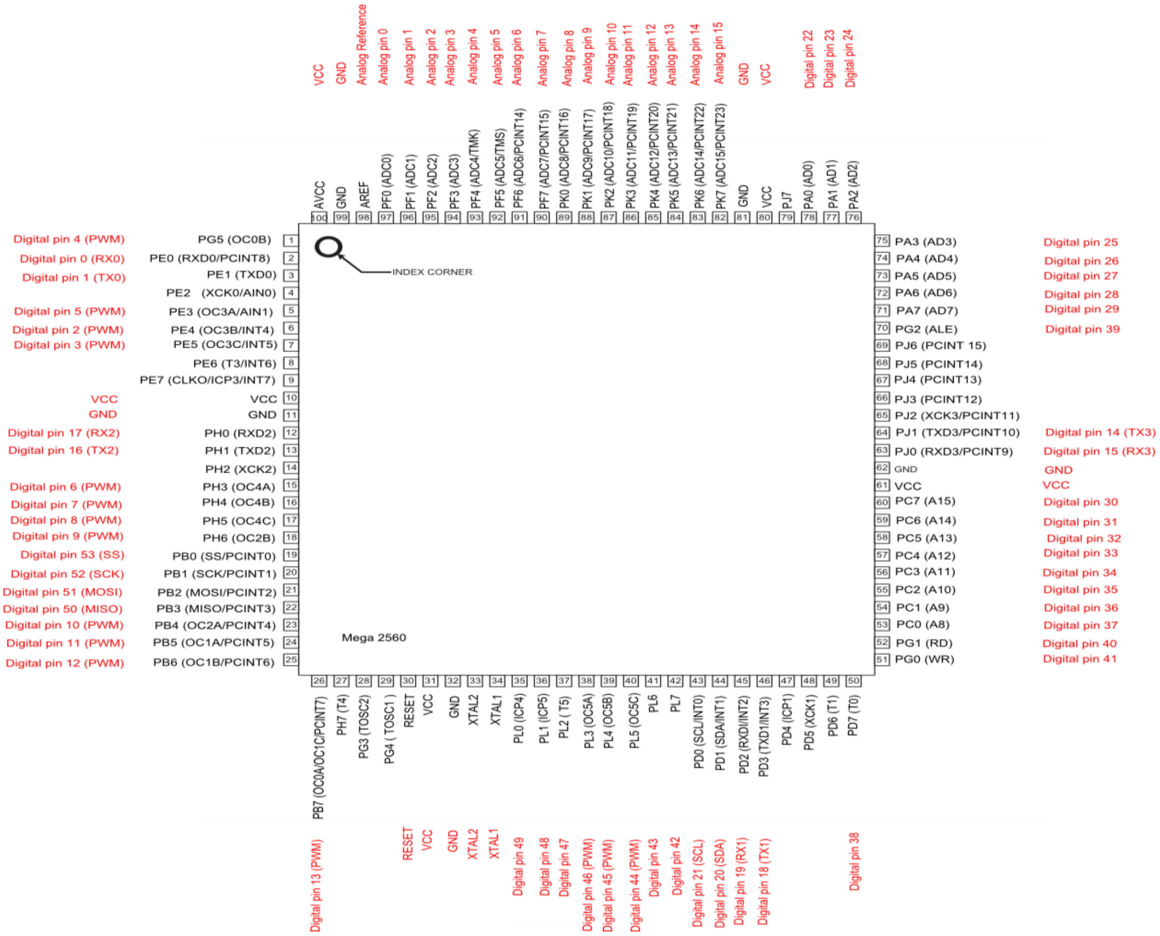


Figure 5.2: Pin Mapping of Arduino Mega ADK.

5.2.2 Linear Actuator

Four 12V (torque, force) actuators were used in our project. Among them one is for movement in x axis, one is for z axis; one is with the gripper which also a move through x axis and the last one is with the counter weight.

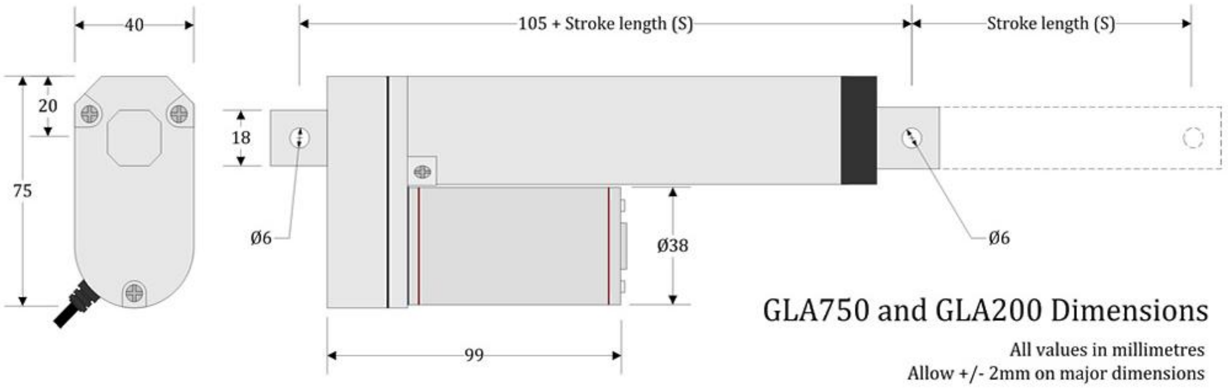
A linear actuator is an actuator that creates motion in a straight line, in contrast to the circular motion of a conventional electric motor. Linear actuators are used in machine tools and industrial machinery, in computer peripherals such as disk drives and printers, in valves and dampers, and in many other places where linear motion is required.

Figure5.3: Linear actuator

The GLA750 is a compact and lightweight 12V DC (direct current) linear actuator available in six different stroke lengths (travel distances) of 30mm (1 3/16"), 50mm (2"), 100mm(4"), 150mm(6"), 200mm(8"),300mm(12").

Despite its small size the GLA750 is able to provide a pushing or pulling force of up to 750N (76kg, 168lbs) at the rated voltage (and as with any DC motor operation at other voltages is possible with modified operating characteristics).

Need a faster actuator? Then please take a look at our GLA200 model with a 32mm/s no-load speed. Alternatively if you need a larger actuator capable of a much greater force then the GLA4000 model may be suitable for you.



5.2.2.1 Operating Characteristics:

Motor: 12V DC 25W

Operating voltage: 3-18V

No Load speed: 10mm/s

No Load Current: 0.8A

Full Load Speed: 8mm/s

Maximum load: 750N/76kg

Current for maximum dynamic load: 4A

Stall current: 5.2A

Maximum static load: 2300N/234kg

Maximum duty cycle: 40%

(The proportion of time that the actuator may be in operation for, up to 2minutes during any continuous stretch. The duty cycle should be reduced below 20% for average loads >300N)

5.2.2.2 Actuator Build Detail

Installation length: 105mm + Stroke length (distance from front to rear mounting hole centers). For example the installation length for a 150mm stroke is 150 + 105 = 255mm.

Mounting: Two 6mm holes. One is towards the end of the piston rod, the other on the rear of the gearbox housing

Body material: Aluminum piston rod and main sleeve & zinc plated steel gearbox and motor housing. Rated to IP64, suitable for most outdoor use

Certification: This is a CE and RoHS compliant product

Gearbox type: 30:1, three-stage spur gears, all-metal

Screw type: ACME, 29° thread angle

Limit switches: Installed at the ends of travel. Click here to see a circuit diagram

Weight:

30mm: 700g

100mm: 820g

200mm: 960g 50mm: 730g

150mm: 890g

300mm: 1100g

5.2.2.3 Leads & Connector Options

The actuator comes with a 0.8m PVC insulated lead as standard (black external sleeve with two internal cores, one red and one black) with uncovered lead ends.

An option is provided for a DC power connector to be added (2.1mm internal diameter, 5.5mm external) on a 300mm lead, bringing the total lead length to 1.1m. The connector is designed to be directly compatible with our manual switch harness and DC extension lead, as well as our female screw-terminal to DC connectors.

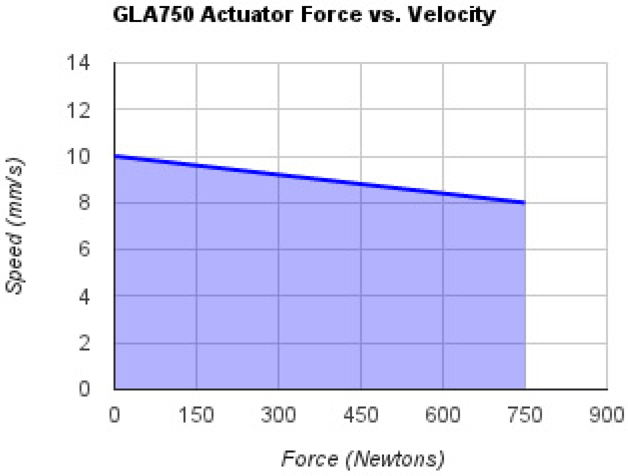


Fig 5.4 Force vs Speed curve

5.2.3 Motor controller

For controlling the motors with Aarduino we used L298 motor controller modules.L298N H-bridge IC can allow controlling the speed and direction of two DC motors, or controlling one bipolar stepper motor with ease.

The L298N H-bridge module can be used with motors that have a voltage of between 5 and 35V DC. With the module used in this tutorial, there is also an onboard 5V regulator, so if your supply voltage is up to 12V you can also source 5V from the board.

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver de-signed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

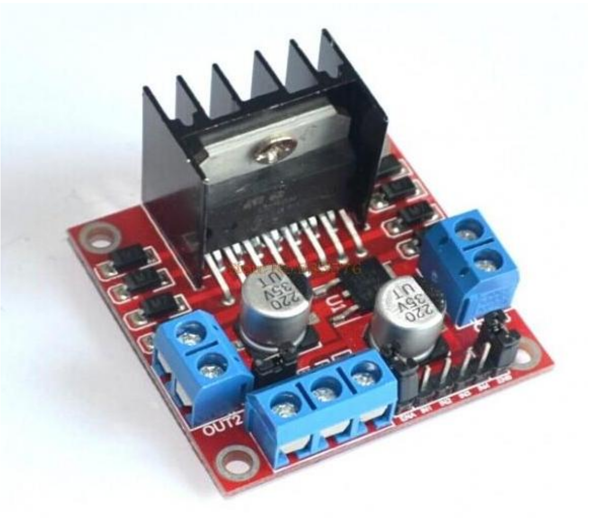
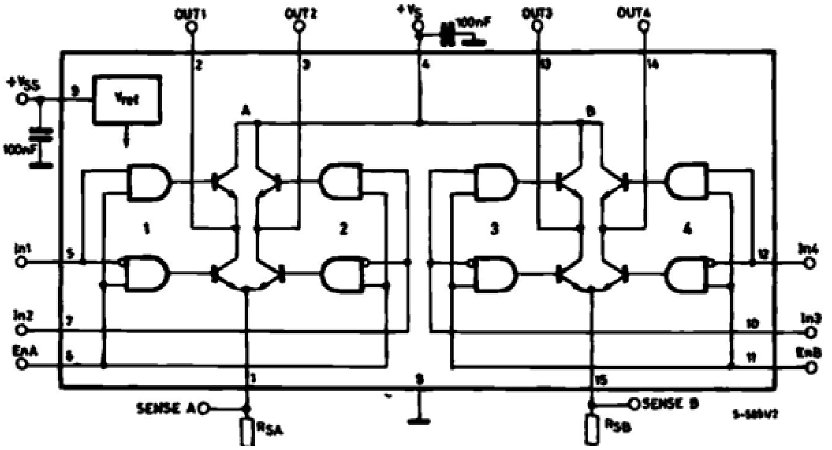


Fig5.5 L293 Motor controller shield



5.2.3.1 Absolute maximum rating

Symbol	Parameter	Value	Unit
V_s	Supply Voltage	36	V
V_{ss}	Logic supply voltage	36	V
V_i	Input Voltage	7	V
V_{en}	Enable Voltage	7	V
I_o	Peak output current	1.2	A
P_{tot}	Total Power Dissipation at Tpins = 90 °C	4	W
T_{stg}, T_i	Storage and Junction Temperature	-40 to 150v	^{0}C

Table5.1 maximum Ratings

5.2.3.2 Pin Functionality

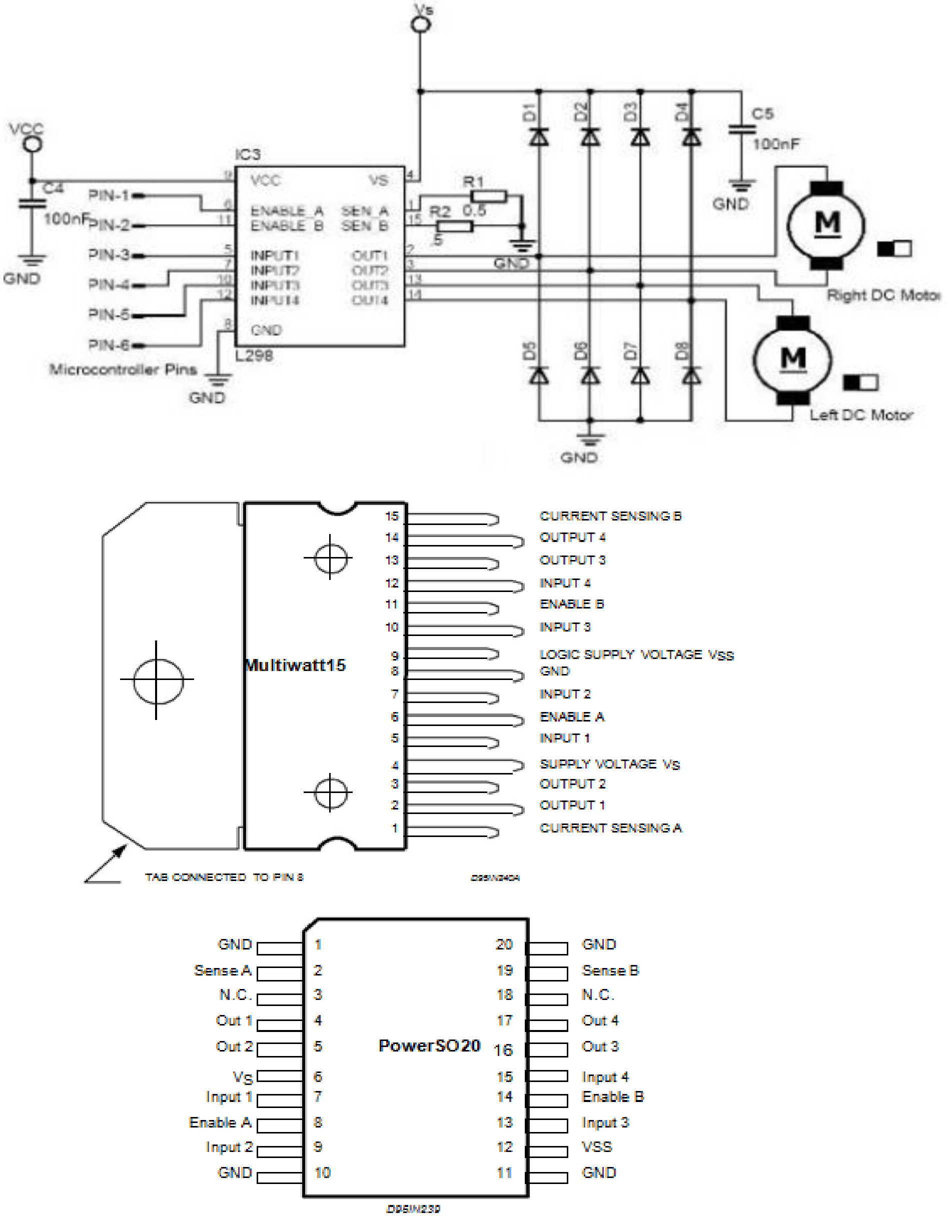


Fig 5.6 Pin Connections

Pin Functions

MW.15	**Power SO**	**Name**	**Function**
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to Control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load Connected between these two pins is monitored at pin 1.
4	6	VS	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this Pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (Enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be Connected between this pin and ground.

10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

Table 5.2 Pin Functions

5.2.3.3 Voltage Specification

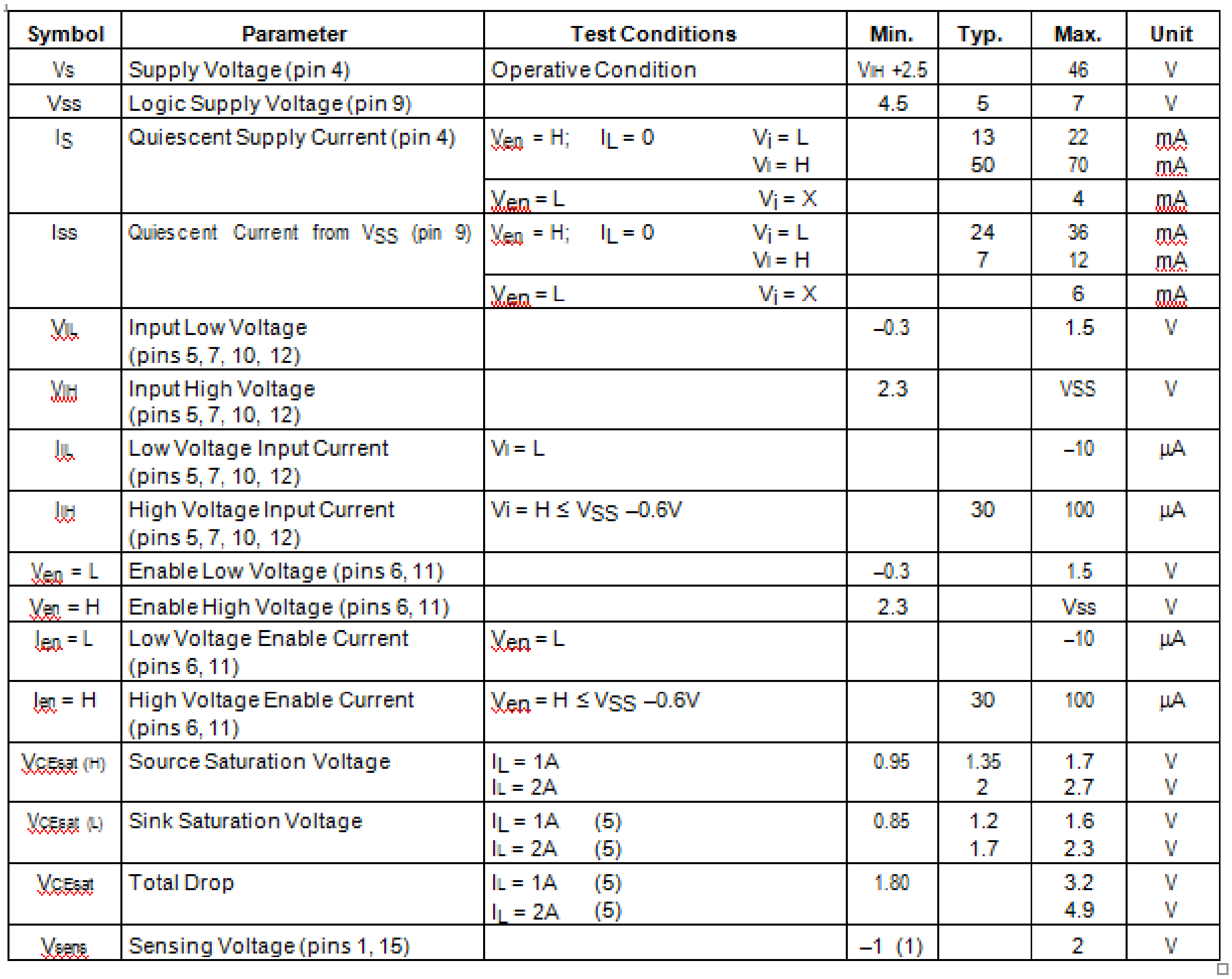
VCC is the voltage that it needs for its own internal operation 5v; L293D will not use this voltage for driving the motor. For driving the motors it has a separate provision to provide motor supply VSS (V supply). L293d will use this to drive the motor. It means if you want to operate a motor at 9V then you need to provide a Supply of 9V across VSS Motor supply.

The maximum voltage for VSS motor supply is 36V. It can supply a max current of 600mA per channel. Since it can drive motors Up to 36v hence you can drive pretty big motors with this l293d.VCC pin 16 is the voltage for its own internal Operation. The maximum voltage ranges from 5v and up to 36v.

TIP: Don’t Exceed the Vmax Voltage of 36 volts or it will cause damage.

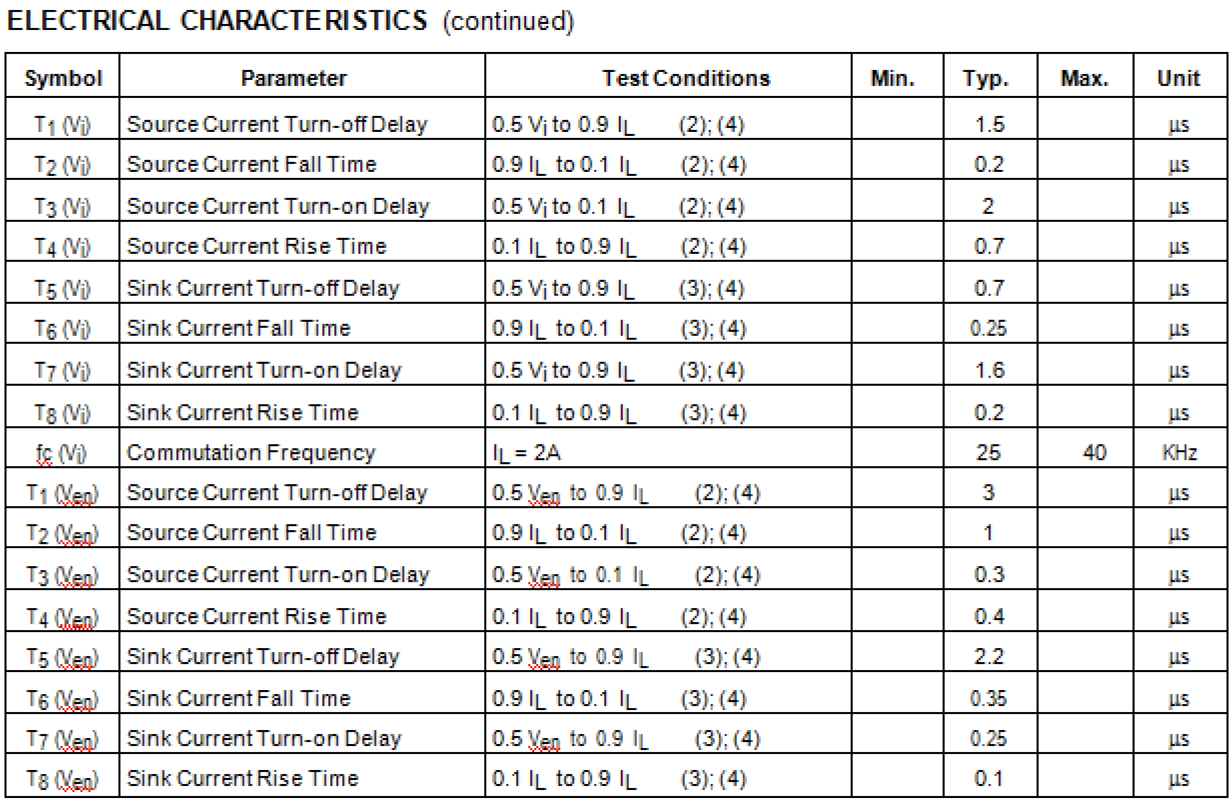
5.2.3.4 ABSOLUTE MAXIMUM RATINGS and ELECTRICAL CHARACTERISTICS (VS = 42V; VSS = 5V, Tj = 25□ C; unless otherwise specified)

Symbol	**Parameter**	**Value**	**Unit**
V_S	Power Supply	50	V
VSS	Logic Supply Voltage	7	V
VI,Ven	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100□ s) –Repetitive (80% on –20% off; ton = 10ms) –DC Operation	 3 2.5 2	 A A A
Vsens	Sensing Voltage	–1 to 2.3	V
Ptot	Total Power Dissipation (Tcase = 75□ C)	25	W
Top	Junction Operating Temperature	–25 to 130	□ C
Tstg, Tj	Storage and Junction Temperature	–40 to 150	□ C



Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_s	Supply Voltage (pin 4)	Operative Condition	V_{IH} +2.5		46	V
V_{ss}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0 V_i = L V_i = H		13 50	22 70	mA mA
		V_{en} = L V_i = X			4	mA
I_{ss}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0 V_i = L V_i = H		24 7	36 12	mA mA
		V_{en} = L V_i = X			6	mA
V_{iL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V_{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		VSS	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_i = L			–10	µA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	V_i = H ≤ V_{SS} –0.6V		30	100	µA
V_{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)		2.3		V_{ss}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L			–10	µA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} –0.6V		30	100	µA
V_{CEsat} (H)	Source Saturation Voltage	I_L = 1A I_L = 2A	0.95	1.35 2	1.7 2.7	V V
V_{CEsat} (L)	Sink Saturation Voltage	I_L = 1A (5) I_L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	I_L = 1A (5) I_L = 2A (5)	1.80		3.2 4.9	V V
V_{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

Table 5.3 Direction Table of Dc Motor



ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V_i)	Source Current Turn-off Delay	0.5 V_i to 0.9 I_L (2); (4)		1.5		µs
T_2 (V_i)	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		0.2		µs
T_3 (V_i)	Source Current Turn-on Delay	0.5 V_i to 0.1 I_L (2); (4)		2		µs
T_4 (V_i)	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.7		µs
T_5 (V_i)	Sink Current Turn-off Delay	0.5 V_i to 0.9 I_L (3); (4)		0.7		µs
T_6 (V_i)	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.25		µs
T_7 (V_i)	Sink Current Turn-on Delay	0.5 V_i to 0.9 I_L (3); (4)		1.6		µs
T_8 (V_i)	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.2		µs
f_c (V_i)	Commutation Frequency	I_L = 2A		25	40	KHz
T_1 (V_{en})	Source Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (2); (4)		3		µs
T_2 (V_{en})	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		1		µs
T_3 (V_{en})	Source Current Turn-on Delay	0.5 V_{en} to 0.1 I_L (2); (4)		0.3		µs
T_4 (V_{en})	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.4		µs
T_5 (V_{en})	Sink Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (3); (4)		2.2		µs
T_6 (V_{en})	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.35		µs
T_7 (V_{en})	Sink Current Turn-on Delay	0.5 V_{en} to 0.9 I_L (3); (4)		0.25		µs
T_8 (V_{en})	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.1		µs

5.2.3.5 Clockwise and Anti-clockwise direction

High Left	High Right	Low Left	Low Right	Description
On	Off	Off	On	Motor runs clockwise
Off	On	On	Off	Motor runs anti-clockwise
On	On	Off	Off	Motor stops or decelerates
Off	Off	On	On	Motor stops or decelerates

Table 5.4: Clock and Anti-clockwise direction in terms of HIGH and LOW power

***5.2.4 Glass motor*:**

We used three wiper motor or glass motors. One is for the rotation through y axis. The motor is calculated through the angle it moves. Another motor is used with the gripper to open and close to hold something. The other one is used for moving the arm in y axis. The speed is normally adjustable, with several continuous speeds and often one or more "intermittent" settings. Here the speed is controlled through adjusting the RPM of the motors. Otherwise the speed was high for our project

Figure 5.6: Glass motor

The DC motors don’t have enough torque to drive a robot directly by connecting wheels in it. Gears are used to increase the torque of dc motor on the expense of its speed**.**

5.2.4.1 Mathematical interpretation:

(a) Calculating Mechanical Power Requirements

(b)Torque - Speed Curves

(c)Numerical Calculation

(d)Sample Calculation

(e)Thermal Calculations

(a)Calculating Mechanical Power Requirements

Physically, power is defined as the rate of doing work. For linear motion, power is the product of force multiplied by the distance per unit time. In the case of rotational motion, the analogous calculation for power is the product of torque multiplied by the rotational distance per unit time.

$P_{rot} = M * \omega$

Where:

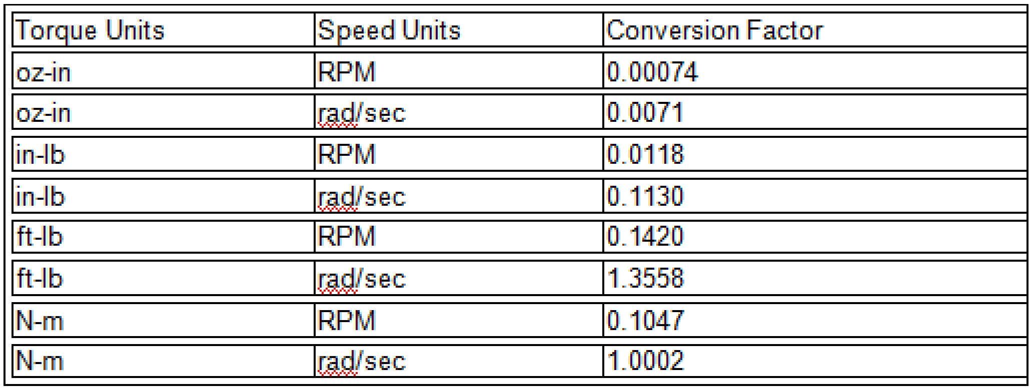
P_{rot} = rotational mechanical power M = torque

ω = angular velocity

The most commonly used unit for angular velocity is rev/min (RPM). In calculating rotational power, it is necessary to convert the velocity to units of rad/sec. This is accomplished by simply multiplying the velocity in RPM by the constant (2 x π) /60:

$^{\omega}rad/\sec^{=\omega}rpm*(2\pi/60)$

It is important to consider the units involved when making the power calculation. A reference that provides conversion tables is very helpful for this purpose. Such a reference is used to convert the torque-speed product to units of power (Watts). Conversion factors for commonly used torque and speed units are given in the following table. These factors include the conversion from RPM to rad/sec where applicable.



Torque Units	Speed Units	Conversion Factor
oz-in	RPM	0.00074
oz-in	rad/sec	0.0071
in-lb	RPM	0.0118
in-lb	rad/sec	0.1130
ft-lb	RPM	0.1420
ft-lb	rad/sec	1.3558
N-m	RPM	0.1047
N-m	rad/sec	1.0002

(b)Torque - Speed Calculation:

Rotational power (Pr) is given by:

Pr =Torque (T) * Rotational Speed (ω)

Thus,

$$\mathbf{T}= \frac{\boldsymbol{Pr}}{\boldsymbol{w}}$$

Pr is constant for DC motor for a constant input electrical power. Thus torque (T) is inversely Proportional speed (ω).

$$T \ \alpha \ \frac{1}{\omega}$$

Thus to increase the value of torque we have to loose speed.

TORQUE (oz-in)	SPEED (rpm)	CURRENT (mA)	POWER (Watts)	EFFICIENCY (%)
-----------	---------	------------	----------	-------------
0.025	11247.65	0.024	0.208	0.10
0.05	10786.3	0.048	0.399	71.87
0.075	10324.95	0.072	0.573	75.27
0.1	9863.6	0.096	0.730	74.99
0.125	9402.25	0.120	0.870	73.25
0.15	8940.9	0.144	0.992	70.78
0.175	8479.55	0.168	1.098	67.89
0.2	8018.2	0.192	1.187	64.73
0.225	7556.85	0.217	1.258	61.40
0.25	7095.5	0.241	1.313	57.95
0.275	6634.15	0.265	1.350	54.41
0.3	6172.8	0.289	1.370	50.80
0.325	5711.45	0.313	1.374	47.14
0.35	5250.1	0.337	1.360	43.44
0.375	4788.75	0.361	1.329	39.71
0.4	4327.4	0.385	1.281	35.95
0.425	3866.05	0.409	1.216	32.17
0.45	3404.7	0.433	1.134	28.37
0.475	2943.35	0.457	1.035	24.56
0.5	2482.0	0.481	0.918	20.74
0.525	2020.65	0.505	0.785	16.90
0.55	1559.3	0.529	0.635	13.05
0.575	1097.95	0.553	0.467	9.20
0.6	636.6	0.577	0.283	5.34
0.625	175.25	0.602	0.081	1.47

Torque - Speed Curve:

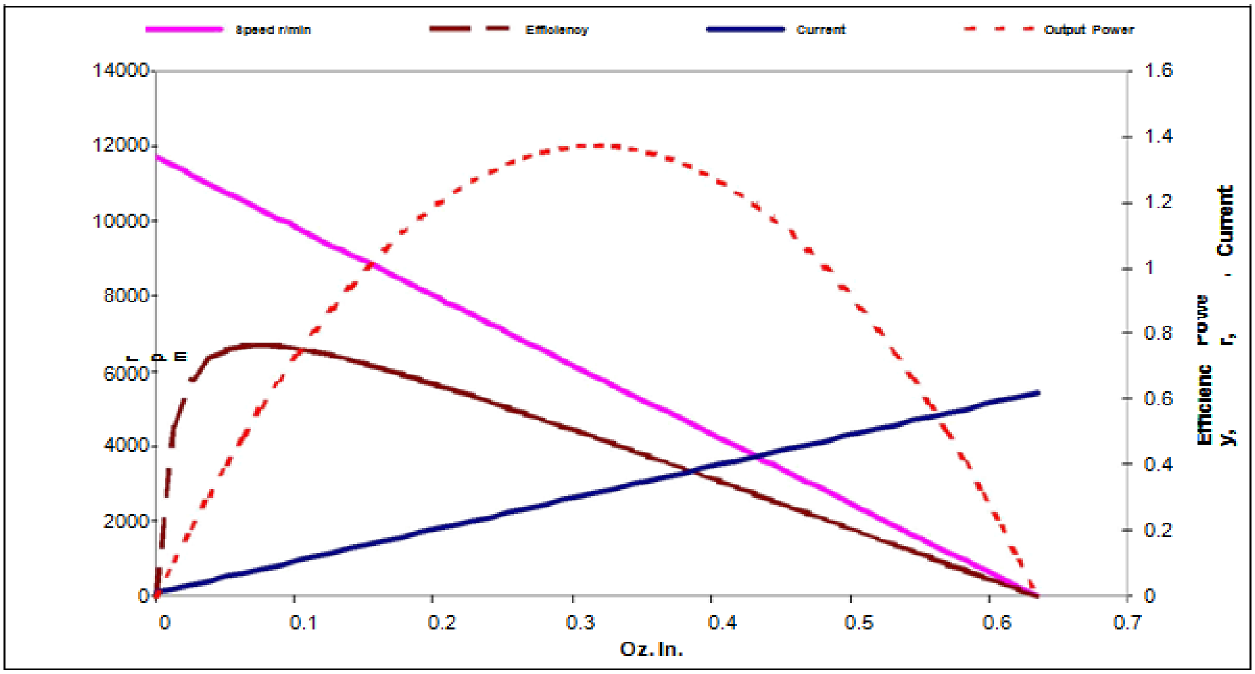


Fig5.7: Torque vs. speed graph

(c)Numerical Calculation

For an iron-less core, DC motor of relatively small size, the relationships that govern the behavior of the motor in various circumstances can be derived from physical laws and characteristics of the motors themselves. Kirchhoff’s voltage rule states, "The sum of the potential increases in a circuit loop must equal the sum of the potential decreases." When applied to a DC motor connected in series with a DC power source, Kirchhoff’s voltage rule can be expressed as "The nominal supply voltage from the power source must be equal in magnitude to the sum of the voltage drop across the resistance of the armature windings and the back EMF generated by the motor.":

$V_O = (I \text{ x} R) V_e$

Where:

V_O = Power supply (Volts)

I = Current (A)

R = Terminal Resistance (Ohms)

V_e = Back EMF (Volts)

The back EMF generated by the motor is directly proportional to the angular velocity of the motor. The proportionality constant is the back EMF constant of the motor.

$V_e = \omega k_e$

Where:

ω= angular velocity of the motor

K_e = back EMF constant of the motor

Therefore, by substitution:

$V_o = (I \text{ x } R) + (\omega k_e)$

The torque produced by the rotor is directly proportional to the current in the armature windings. The proportionality constant is the torque constant of the motor.

$M_o = I \text{ x} k_M$

Where: $_{Mo}$ = torque developed at rotor k_M = motor torque constant

Substituting this relationship:

$V = ^{(M/R)} + (\omega k_{e)} k_M$

The torque developed at the rotor is equal to the friction torque of the motor plus the resisting torque due to external mechanical loading:

$M_o = M_l + M_f$

Where:M_f = motor friction torque M_l = load torque

Assuming that a constant voltage is applied to the motor terminals, the motor velocity will be directly proportional to sum of the friction torque and the load torque. The constant of proportionality is the slope of the torque-speedcurve.

Our motor is to be operated with 12 volts applied to the motor terminals. The torque load is 0.2 oz-in. Find the resulting motor speed, motor current, efficiency, and mechanical power output.

From the motor data sheet, it can be seen that the no-load speed of the motor at 12 volts is 11,700 rpm. If the torque load is not coupled to the motor shaft, the motor would run at this speed.

The motor speed under load is simply the no-load speed less the reduction in speed due to the load. The proportionality constant for the relationship between motor speed and motor torque is the slope of the torque vs. speed curve, given by the motor no-load speed divided by the stall torque. In this example, the speed reduction caused by the 0.2 oz -in torque load is:

0.2 oz-in x (11,700 rpm/.634 oz-in) = -3690 rpm The motor speed under load must then be:

11,700 rpm - 3690 rpm = 8010 rpm

The motor current under load is the sum of the no-load current and the current resulting from the load. The proportionality constant relating current to torque load is the torque constant (k_M), in this case, 1.039 oz -in/A. In this case, the load torque is 0.2 oz-in, and the current resulting from the load must be:

I = 0.2 oz-in x 1 amp/1.039 oz -in = 192 mA

The total motor current must be the sum of this value and the motor no-load current. The data sheet lists the motor no-load current as 11 mA. Therefore, the total current is:

192 mA + 11 mA = 203 mA

The mechanical power output of the motor is simply the product of the motor speed and the torque load with a correction factor for units (if required). Therefore, the mechanical power output of the motor in this application is:

Output power = 0.2 oz-in x 8010 rpm x .00074 = 1.18 Watts

The mechanical power input to the motor is the product of the applied voltage and the total motor current in Amps. In this application:

Input power = 9 volts x .203 A = 1.82 Watts

Since efficiency is simply power out divided by power in, the efficiency in this application is:

So our motor efficiency yet to be get with full power of battery= 1.18 Watts / 1.82 Watts = .65 = 65%

5.2.5 HC-05 Bluetooth Module

This Bluetooth module is a software configurable Master/Slave mode, which can be selected by simple AT commands. This module is a simple replacement for wired serial connections. You can use it simply for converting a normal serial port into a wireless serial port to establish connection between MCU, GPS, and robot like devices with PC or embedded board.

This module comes with 3.3V regulator on board; module can be powered by 3.6V -6V supply. RXD TXD pins are 3.3V level so connect MAX3232 for connecting it with PC serial port. Onboard LED is for status. Default pairing password is 1234.

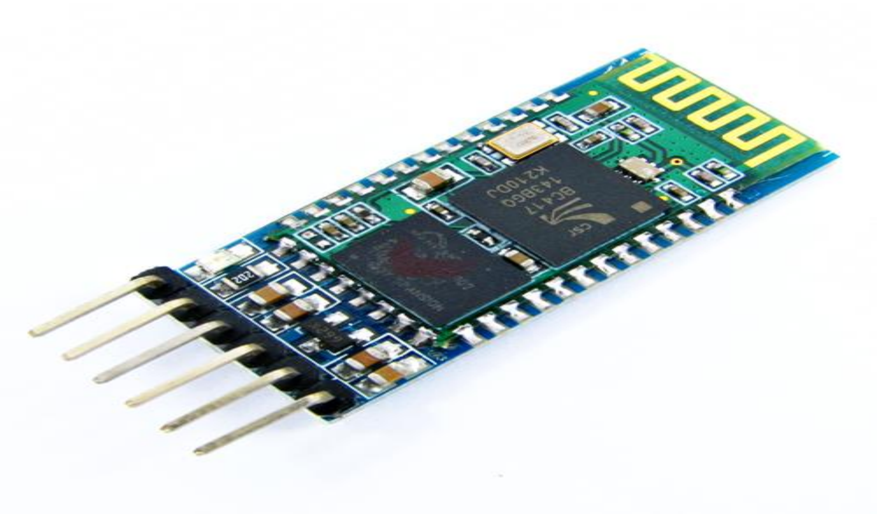


Figure 5.8: HC-05 BLUETOOTH MODULE

5.2.5.1 Specifications

- Bluetooth protocol: Bluetooth Specification v2.0+EDR

- Frequency: 2.4GHz ISM band
- Modulation: GFSK(Gaussian Frequency Shift Keying)
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3VDC 50mA
- Dimensions: 15.2x35.7x5.6mm
- Chipset CSR BC417143
- Bluetooth version V2.0+EDR
- Flash 8Mbit
- RXD and TXD are 3.3V level Brand new and high quality.
- Suitable for Bluetooth adapter, Bluetooth phone use, master-slave can also be used for both.
- Can be set for the module control parameters and control commands issued via AT commands.
- Provide 7 input and output ports, scalable user IO resources.
- Great for embedded wireless serial transmission alternatives.
- USB protocol: USB v1.1/2.0.
- Frequency: 2.4GHz ISM band.
- Modulation: GFSK (Gaussian Frequency Shift Keying).
- Sensitivity: not more than -84dBm at 0.1% BER.
- Rate: Asynchronous: 2.1Mbps (Max) / 160 kbps, Synchronous: 1Mbps/1Mbps.

- Security features: Authentication and encryption.
- Support profiles: Bluetooth serial port (master & slave).
- Power Supply: 3.3V - 6V DC, 50mA.
- Working temperature: –5°C to 45°C.
- Max. Serial baud rate: 1382400bps, support for hardware flow control transfer.

5.2.5.2Software features

1. Default Baud rate: 38400, Data bits: 8, Stop bit: 1, Parity: No parity, Data control has Supported.

Baud rate: 9600,19200,38400,57600,115200,230400,460800

2. Given a rising pulse in PIO0, device will be disconnected.

3. Status instruction port PIO1: low-disconnected, high-connected.

PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are Paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.

4. Auto-connect to the last device on power as default.

5. Permit pairing device to connect as default.

6. Auto-pairing PINCODE:”1234” as default

7. Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

5.2.5.3Schematic

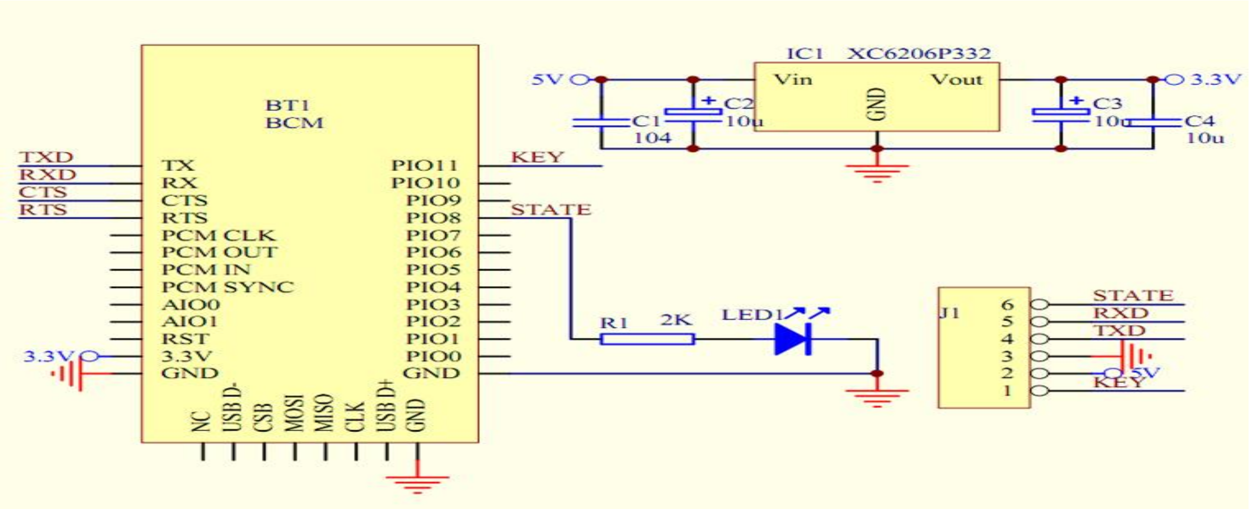


Figure 5.9: Schematic of HC-05

5.2.5.4 Pin definition

Pin	**#**	**Pad Type**	**Description**
KEY	1	CMOS input with weak internal pull-up	HC-05 for into AT mode, HC-06 for clear paired information
VCC	2	3.3V	Integrated 3.3V(+)supply with on-chip linear regulator output within 3.15-3.3V
GND	3	VSS	Ground port
TXD	4	CMOS output, tri-stable with weak internal pull-up	Asynchronous data output (UART Transmit)

RXD	5	CMOS input with weak internal pull-up	Asynchronous data input (UART Receive)
STATUS	6	CMOS output with weak internal pull-up	LED output status, and Bluetooth is not connected then output pulse, If Bluetooth connectivity, the output is high. It to judge state by the MCU

Table 5.5: Pin definition of HC-05

5.2.6 Battery

5.2.6.1 9V battery

Figure 5.10: Battery

For powering we’ve used energizer, alkaline, SONY 14V, 3800 mAh battery to power up the four dc motors to move the wheels and to power up the L293D to control those DC motors. We need to power up the Arduino MEGA with a battery voltage in the range of 3V TO 9V. Arduino MEGA can’t be power up in more than 10V-12V, otherwise it can damage the Arduino MEGA. One 12V, and 20Ah Rechargeable battery was used to provide Voltage to the Actuators. This was a lead-acid battery weighing 7.1 kgs and 181 X 77 X 171 mm in dimension. The battery was accommodated outside. With full charge each battery shows 12.9 volts across their terminals. The battery can provide 2.1A max.

5.2.6.2 Led acid Battery

Lead-acid batteries store energy using a reversible chemical reaction between lead plates and dilute sulphuric acid (electrolyte). There are three basic types of lead acid battery - starter batteries: used to start engines in cars etc., **deep-cycle batteries**: used in renewable energy applications and camping etc., and marine batteries: used both for starting and for deep cycle applications.

Fig 5.11: Led acid battery

5.2.6.1 (a) Different Types of Lead Acid Battery

Starter batteries have many thin lead plates which enables them to discharge a lot of energy very quickly - i.e. to start a vehicle. However, if a starter battery is **discharged deeply** (more than 20-25% depth of charge), its plates can be permanently damaged and the lifetime of the battery greatly reduced.

Deep cycle batteries have fewer thicker lead plates, and so cannot discharge energy so quickly, but can be cycled deeply and recharged many times without damaging the battery. Deep cycle batteries are designed to provide a steady current over a long period of time.

5.2.6Counter weight:

When our am is in mean position (0, 0, 0); the weight in the position where center on gravity in on the A point.

Moment/torque = force X distance

$\Sigma F = 0$

$\Sigma \text{Moments}=0$

When load P is removed

$\Sigma MA=0$,$Qx=20(5+1)$

So, $Qx=117$pounds…..equation (1)

When load P is applied

$\Sigma MB=0$, $Q(x+5)=20(1)+20(10)$; $Qx+5Q=182$,

As we know $Qx=117$, so$117+5Q=182$; $Q=13$ pounds

As 1pound=.453592kg

Substitute Q = 5.89kg to Equation (1)

18x=117;

X=6.5in;

As we are changing the arm for 6 degree free rotation along with changing the the three axis rotation so our CG will be changed so to keep CG on A position we need to keep that on position by getting a counter weight which keep the arm ob balanced.

As we already get our counter weight

So the formula is Q(counter)*D1(distance counter weight)=Forward weight*D2

When Y axis -=0 and X-axis and Z axis is changing

P(x,y,z)	No. of points	Q	D1	Forward weight	D2
(0,0,0)	1.(off)	5.89kg	6.5in	38.285kg	1
(2,0,0)	2(A)	5.89kg	7.4in	21.79kg	2
(5,0,3)	3(B)	5.89kg	8in	9.42kg	5
(8,0,4)	4(C)	5.89kg	8.79	6.47kg	8
(11,0,8)	5(D)	5.89kg	9.77ini	5.23kg	11
(13,0,9)	6(E)	5.89kg	10.56in	4.78kg	13
(20,0,11)	7(F)	5.89kg	12.89in	3.79kg	20
(20,0,13)	8(G)	5.89kg	13.43in	3.95kg	20
(20,0,15)	9(5)	5.89kg	14.22in	4.18kg	20

When Y axis -=1and X-axis and Z axis is changing same as before

P(x,y,z)	No. of points	Q	D1	Forward weight	D2
(0,1,0)	1.(off)	5.89kg	6.5in	38.285kg	2
(5,1,3)	2(A)	5.89kg	7.4in	21.79kg	5
(8,1,4)	3(B)	5.89kg	8in	9.42kg	8
(11,1,8)	4(C)	5.89kg	8.79	6.47kg	11
(13,1,9)	5(D)	5.89kg	9.77ini	5.23kg	13
(20,1,11)	6(E)	5.89kg	10.56in	4.78kg	20
(21,1,13)	7(F)	5.89kg	12.89in	3.79kg	21

(22,1,15)	8(G)	5.89kg	13.43in	3.95kg	22
(22,1,17)	9(5)	5.89kg	14.22in	4.18kg	22
(22,1,17.5)	10(0)	5.86kg	14.22kg		22

When Y axis -=2and X-axis and Z axis is changing same as before

P(x,y,z)	No. of points	Q	D1	Forward weight	D2
(0,2,0)	1.(off)	5.89kg	6.5in	12.76kg	3
(5,2,3)	2(A)	5.89kg	7.4in	10.89kg	4
(8,2,4)	3(B)	5.89kg	8in	5.24kg	9
(11,2,8)	4(C)	5.89kg	8.79in	4.3kg	12
(13,2,9)	5(D)	5.89kg	9.77in	3.13kg	18
(20,2,11)	6(E)	5.89kg	10.56in	2.96kg	21
(21,2,13)	7(F)	5.89kg	12.89in	3.45kg	22
(22,2,15)	8(G)	5.89kg	13.43in	3.43kg	23
(22,2,17)	9(5)	5.89kg	14.22in	3.64kg	23
(22,2,17.5)	10(0)	5.89kh	14.22in	3.46kg	23

When Y axis -=3and X-axis and Z axis is changing same as before

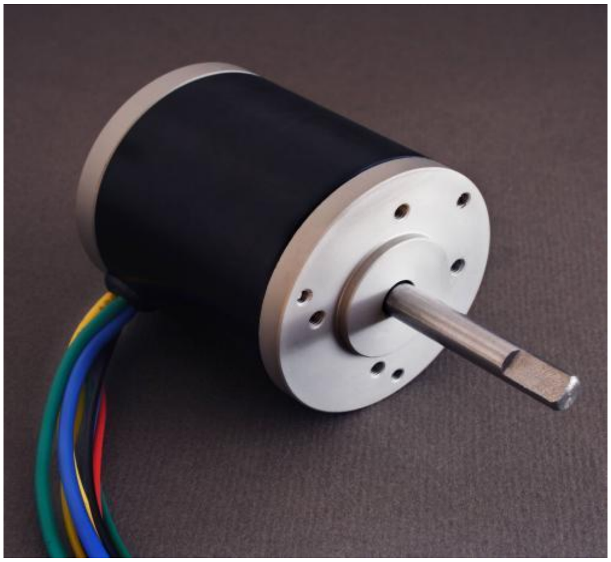
P(x,y,z)	No. of points	Q	D1	Forward weight	D2	D2
(0,3,0)	1.(off)	5.89kg	6.5in	9.5	4	4
(5,3,3)	2(A)	5.89kg	7.4in	4.84kg	9	2
(8,3,4)	3(B)	5.89kg	8in	3.43kg	12	5
(11,3,8)	4(C)	5.89kg	8.79	3.45kg	15	8
(13,3,9)	5(D)	5.89kg	9.77ini	3.19kg	18	11
(20,3,11)	6(E)	5.89kg	10.56in	2.82kg	22	13
(21,3,13)	7(F)	5.89kg	12.89in	3.3kg	23	20
(22,3,15)	8(G)	5.89kg	13.43in	3.29kg	24	20
(22,3,17)	9(5)	5.89kg	14.22in	3.48kg	24	20
(22,3,17.5)	10(0)	5.89kg	14.22in	3.48kg	24	

When Y axis -=4and X-axis and Z axis is changing same as before

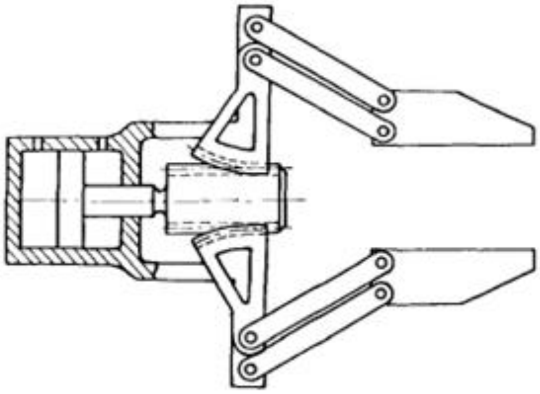
P(x,y,z)	No. of points	Q	D1	Forward weight	D2
(0,4,0)	1.(off)	5.89kg	6.5in	7.6kg	5
(5,4,3)	2(A)	5.89kg	7.4in	3.96kg	11
(8,4,4)	3(B)	5.89kg	8in	3.36kg	14
(11,4,8)	4(C)	5.89kg	8.79	3.04kg	17

(13,4,9)	5(D)	5.89kg	9.77ini	2.74kg	21
(20,4,11)	6(E)	5.89kg	10.56in	2.5kg	24
(21,4,13)	7(F)	5.89kg	12.89in	33.04kg	25
(22,4,15)	8(G)	5.89kg	13.43in	3.16kg	25
(22,4,17)	9(5)	5.89kg	14.22in	3.16kg	25
(22,4,17.5)	10(0)	5.89kg	14.22in	3.16kg	25

5.2.7 DC motor in Gripper



Gripper Assembly mates easily with one dc motor which have shaft to open and close the fingers, and also acts as the gripper's "wrist".



5.3 Software Overview

5.3.1 What is Android?

Android is one of the most popular OS. Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast—every day another million users power up their Android devices for the first time and start looking for apps, games, and other digital content.

Android gives us a world-class platform for creating apps and games for Android users everywhere, as well as an open marketplace for distributing to them instantly.

5.3.3 Android IDEs

For Android Development purpose and IDE is required. Focusing on Developers comfort and flexibility some IDEs give the facility of Android Development. They are:

1. Eclipse (Juno Version)
2. IntelliJ IDEA
3. Android Studio

CHAPTER 6: REVERSE ENGINEERING

*6.**1 Solid works design***

The most important part of our project was to design a Solid Works structure as it is easy to implement the structure easily if there is a perfect design. Through the designed structure it is quite easy to implement without any trial and error and it takes less time. In our project everything of the structure was designed in the software even the counter weight also.

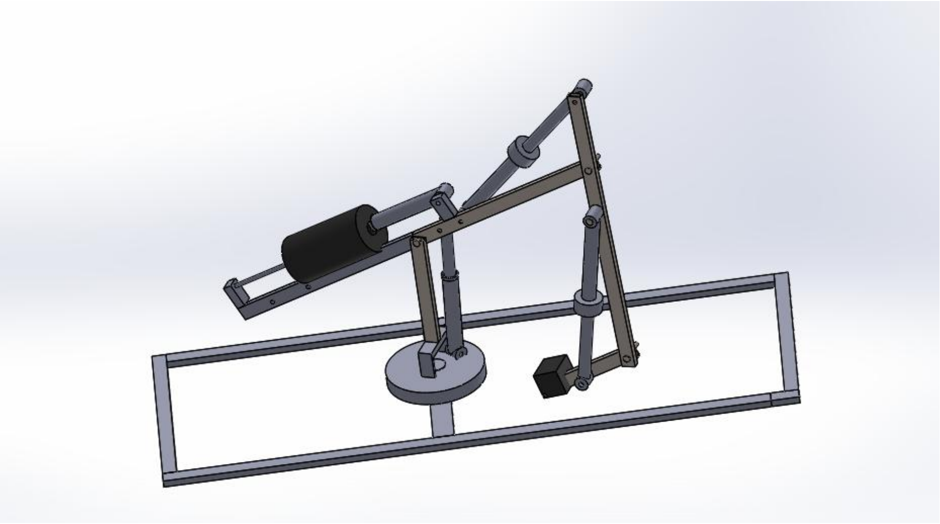


Fig 6.1 Solid Works 3D Model

As ewe firstly used block to measure the counter weight in solid work design with three cubes so the calculation we find from solid work helps us to find 5.89kg as counter weight to fix the CG.

Mass properties of arm total

Configuration: Default

Coordinate system: -- default --

Mass = 60.30 pounds

Volume = 241.81 cubic inches

Surface area = 2603.10 square inches

Center of mass: (inches)

X = 6

Y = 3.33

Z = 6.17

Principal axes of inertia and principal moments of inertia: (pounds * square inches)

Taken at the center of mass.

Ix = (0.02, 0.99, -0.12) Px = 5561.60

Iy = (-0.02, 0.12, 0.99) Py = 19545.27

Iz = (1.00, -0.01, 0.02) Pz = 21903.90

Moments of inertia: (pounds * square inches)

Taken at the center of mass and aligned with the output coordinate system.

Lxx = 21898.62 Lxy = 255.94 Lxz = -80.52

Lyx = 255.94 Lyy = 5770.58 Lyz = -1679.32

Lzx = -80.52 Lzy = -1679.32 Lzz = 19341.57

Moments of inertia: (pounds * square inches)

Taken at the output coordinate system.

Ixx = 24858.89 Ixy = 1228.51 Ixz = 1722.28

Iyx = 1228.51 Iyy = 9480.95 Iyz = -442.33

Izx = 1722.28 Izy = -442.33 Izz = 21426.33

5.2 Coordinate system geometry for the six-axis robot

The design of the robot is a six degree-of-freedom, articulated arm (Figure 1). Joints one and two are concurrent and are located at the base of the robot, the shoulder.

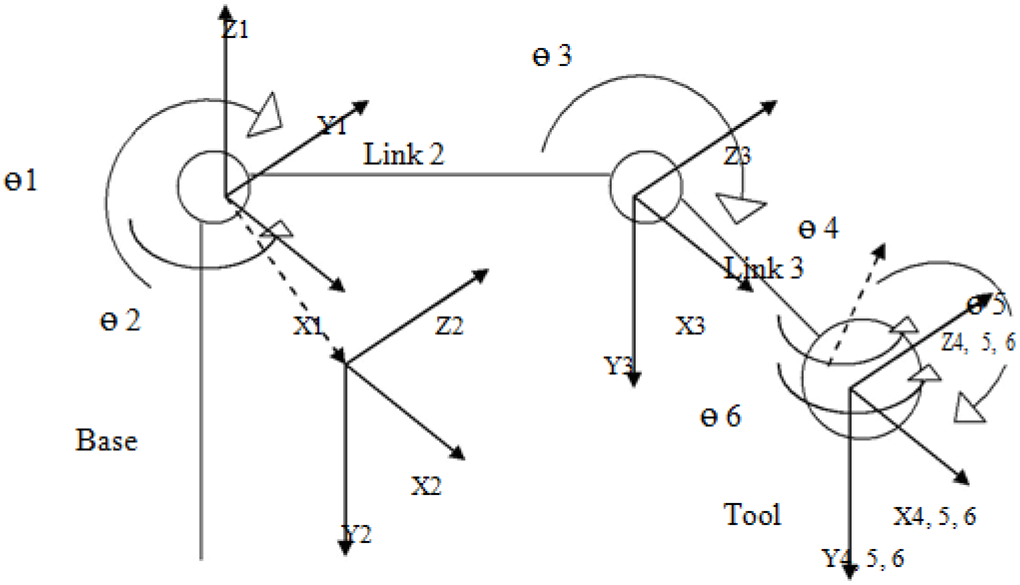


Figure 6.2: Coordinate system geometry for the six-axis robot

Joint 3 is located along a link 20 inches from the base and acts as an elbow. The fourth, fifth and sixth joints are also concurrent and are located 14 inches from joint three; these joints act as a wrist. The wrist has a roll axis, followed by a pitch axis, followed by another roll axis. Having three concurrent joints simplifies the motion of the wrist. For example, a rotation of joint four does not result in a translation of joints five and six. This minimizes the complexity of the equations of motion of the wrist.

6.2.Denavit&Hartenberg (D-H) notation

The definition of a manipulator with four joint-link parameters for each link and a systematic procedure for assigning right-handed orthonormal coordinate frames, one to each link in an open kinematic chain, was proposed by Denavit&Hartenberg ,so is known as Denavit -Hartenberg (DH) notation. Figure 2 shows a pair of adjacent links, link(i-1) and link i, their associated joints, joint (i-1), i and (i+1), and axis (i-2), (i-1) and i respectively.

A frame {i} is assigned to link i as follows:

i. The Zi -1 lies along the axis of motion of the ith joint.

ii. The Xi axis is normal to the Zi-1 axis, and pointing away from it.

iii. The Yi axis completes the right – handed coordinate system as required.

The DH representation of a rigid link depends on four geometric parameters associated with each links. These four parameters completely describe any revolute or prismatic joint as follows:

i.Link twist (αi) – angle between zi-1 and zi axes measured about xi-axis in the right hand sense.

ii. Joint distance (di) - distance measured along zi-1 axis from the origin of frame {i-1} to intersection of xi axis with zi-1 axis.

iii.Joint angle (θi) – angle between xi-1 and xi axes measured about the zi-1 axis in the right hand

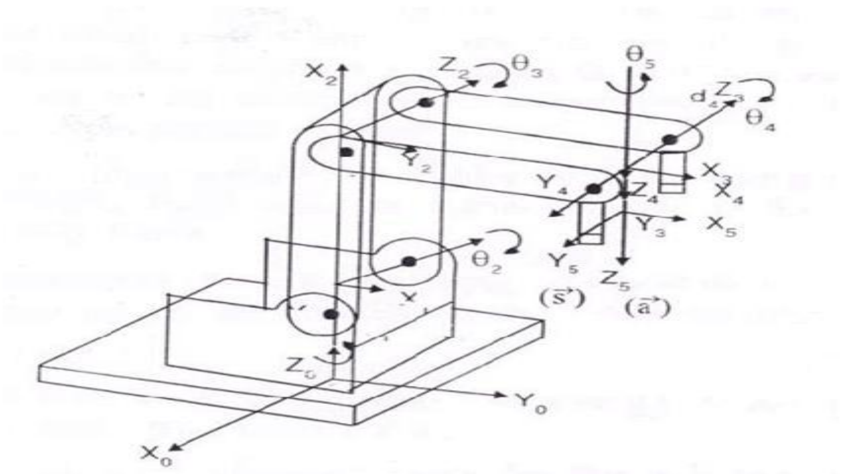


Figure 6.3. D-H representation of Pravak Robot Arm

The kinematic model is shown in Figure 3 with frame assignments according to the Denavit&Hartenberg (D-H) notations. The kinematic parameters according to this model are given in Table 2.

Table 2. D-H Parameter for Pravak Robot Arm

Joint	Θ_i ( )	α_i ( )	a_i(mm)	d_i(mm)
1	Ɵ1	-90	0	226
2	Ɵ2	0	179	0
3	Θ_3	0	177	0
4	Ɵ4	-90	0	0
5	Ɵ5	0	0	80

6.3 Kinematic relationship between adjacent links

Once the DH coordinate system has been established for each link, a homogeneous transformation matrix can easily be developed considering frame {i-1} and frame {i}. This transformation consists of four basic transformations as shown in Figure 4 and the joint link parameter as given in Table 1.

i. A rotation about zi-1 axis by an angle θi

ii. Translation along zi-1 axis by distance di

iii. Translation by distance ai along xi axis and

iv. Rotation by angle αi about xi axis i-1

Ti = Tz(θi) Tz(di) Tx(ai) Tx(αi)

$$^{i-1}T_i = \begin{bmatrix} C\theta i & -S\theta i & 0 & 0 \\ S\theta i & C\theta i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & di \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & ai \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha i & -S\alpha i & 0 \\ 0 & S\alpha i & C\alpha i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{i-1}T_i = \begin{bmatrix} C\theta i & -S\theta i C\alpha i & S\theta i S\alpha i & aiC\theta i \\ S\theta i & C\theta i C\alpha i & -C\theta i S\alpha i & aiS\theta i \\ 0 & S\alpha i & C\alpha i & di \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where, Te is end-effector transformation matrix.

This kinematic model can also be expressed by 12 equations as: nx = c1*c234*c5+s1*s5

ny = s1*c234*c5-c1*s5 nz = s234*c5

ox = c1*c234*s5+s1*c5 oy = -s1*c234*s5-c1*c5 oz = -s234*s5

ax = c1*s234 ay = s1*s234 az = -c234

px = c1*(s234*d5+a3*c23+a2*c2+a1) py = s1*(s234*d5+a3*c23+a2*c2+a1) pz = -c234*d5+a3*s23+a2*s2+d1

6.3.1 RESULT

For the home position, a program in MATLAB 2007 is made and its output is compared with the experimental result as follows.

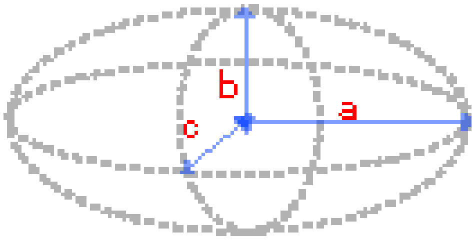
Experimental

Result: [0 177 325 1]

Matlab Program Output: [0 177.00 325.00 1]

6.4 Developing arm equation:

We used ellipsoid to develop the arm equation. An ellipsoid is a closed quadric surface that is a three-dimensional analogue of an ellipse. The standard equation of an ellipsoid centered at the origin of a Cartesian coordinate system and aligned with the axes is.



$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1$$

The volume of an ellipsoid is given by the following formula: 4/3*(pi*a*b*c)

The surface area of a general ellipsoid.

$$S \approx 4\pi\left[(a^p b^p + a^p c^p + b^p c^p)/3\right]^{\frac{1}{p}}$$

$$S \approx 4\pi\left[(a^p b^p + a^p c^p + b^p c^p)/3\right]^{\frac{1}{p}}$$

Where p=1.6075

Calculating Ellipsoid

Semi-axis a: .3556m

Semi-axis b: .6096m

Semi-axis c: .3048m

Volume: 0.27676m^3

Surface area: 2.18761 m^2

Then place the arm with this ellipsoid. It will established system with DC motor the to reach the targeted point

The “shoulder” joint has two motors allowing the arm to rotate in the range $(0,\pi/2)$ in the *xz*plane and in the range(0,□ π/2)in the yzplane. The “elbow” joint has one motor allowing the arm to rotate in the range(-π ,π)in the yz plane. All points within a sphere of radius r=L1+L2 , where L1,L2 are the lengths of the two parts of the arm. The centre of this sphere is located on the “shoulder” coordinates The “shoulder” joint has two motors allowing the arm to rotate in the range (0,pi/2) in the xzplane and in the range(0,pi/2) in the yz plane. The “elbow” joint has one motor allowing the arm to rotate in the range $(-\pi,\pi)$ in the *yz* plane. All points within a sphere of radius $r=L_1+L_2$, where L_1,L_2 are the lengths of the two parts of the arm. The centre of this sphere is located on the “shoulder” coordinates

$$x=\rho\sin\kappa\cos\phi$$

$$y=\rho\sin\kappa\sin\phi,\forall\rho\in(0,r)$$

$$z=\rho\cos\kappa$$

$$x^2+y^2+z^2=\rho^2,\forall\rho\in(0,r)$$

here $\rho=$ the radius of the of the ellipse sphere

$\kappa=$ the ellipsoid constant =90

$\phi=$ *the angle which* is created between x-y axis

$\theta_1=$ the angle between x-z axis with the arm length when it is rotating

$x_m = [L_1 \sin\theta_1 + L_2 \sin(\theta_1, \theta_2)] \cos\phi$

$y_m = [L_1 \sin\theta_1 + L_2 \sin(\theta_1, \theta_2)] \sin \varphi$

$z_m = [L_1 \cos\theta_1 + L_2 \cos(\theta_1, \theta_2)] \cos\theta 1$

here L1=16in, L2=15in

so at mean position to find arm kinematics we get, $\theta 1=90$, and $\theta 2= 167$, $\phi=90$

so we get $xm=0$, $ym=1m$, $zm=0$

Th The angles can be calculated geometrically using the equations

$$\tan\phi = \frac{y_m}{x_m}$$

So tan $\phi=0$;

As ym=1 inch ,zm=0

$\cos\kappa = zm/r$

so
k=acos(zm/r)

K=90

Now calculate the target point we have point when x axis is changing but z is unchanged. It creats*(ρ=2)*

No. of points	θ_1	, θ_2	ϕ	x	y	z	P(x,y,z)
1.(A)	90	167	0	2	0	0	(2,0,0)
2(B)	85	167	0	5	0	3	(5,0,3)
3(C)	82.5	167	0	8	0	4	(8,0,4)
4(D)	76.23	167	0	11	0	8	(11,0,8)
5(E)	70.6	167	0	13	0	9	(13,0,9)
6(F)	60	167	0	17	0	11	(20,0,11)
7(G)	49.65	167	0	19	0	13	(20,0,13)
8(5)	35	167	0	22	0	15	(20,0,15)

Now calculate the target point Y=1 we have point when x axis is changing but z is unchanged. It creates□ *ρ=5*

No. of points	θ_1□1	,□ $\theta_2$2	ϕ	x	y	z	P(x,y,z)
1.(A)	90	167	30	2	1	3	(5,1,3)
2(B)	85	167	30	5	1	4	(8,1,4)
3(C)	82.5	167	30	8	1	8	(11,1,8)
4(D)	76.23	167	30	11	1	9	(13,1,9)
5(E)	70.6	167	30	13	1	11	(20,1,11)
6(F)	60	167	30	17	1	13	(21,1,13)
7(G)	49.65	167	30	19	1	15	(22,1,15)
8(5)	35	167	30	22	1	17	(22,1,17)
9(0)	33	167	30	22	1	17.5	(22,1,17.5)

Now calculate the target pointy=2 we have point when x axis is changing but z is unchanged. It creates□ $\rho=2$

No. of points	$\theta_1$1	,□ $\theta_2$2	ϕ	x	y	z	P(x,y,z)
1.(A)	90	167	60	2	2	3	(5,2,3)

2(B)	85	167	60	5	2	4	(8,2,4)
3(C)	82.5	167	60	8	2	8	(11,2,8)
4(D)	76.23	167	60	11	2	9	(13,2,9)
5(E)	70.6	167	60	13	2	11	(20,2,11)
6(F)	60	167	60	17	2	13	(21,2,13)
7(G)	49.65	167	60	19	2	15	(22,2,15)
8(5)	35	167	60	22	2	17	(22,2,17)
9(0)	33	167	60	22	2	17.5	(22,2,17.5)

Now calculate the target point we have point when x axis is changing but z is unchanged. It creates□ $\rho=5$

No. of points	θ_1	$,\theta_2$	ϕ	x	y	z	P(x,y,z)
1.(A)	90	167	90	5	3	3	(5,3,3)
2(B)	85	167	90	8	3	3	(8,3,4)

3(C)	82.5	167	90	11	3	4	(11,3,8)
4(D)	76.23	167	90	13	3	8	(13,3,9)
5(E)	70.6	167	90	20	3	9	(20,3,11)
6(F)	60	167	90	21	3	11	(21,3,13)
7(G)	49.65	167	90	22	3	13	(22,3,15)
8(5)	35	167	90	22	3	15	(22,3,17)
9(0)	35	167	90	22	3	17.5	(22,3,17.5)

Now calculate the target point we have point when x axis is changing but z is unchanged. It creats$\rho=5$

No. of points	θ_1	,θ_2	ϕ	x	y	z	P(x,y,z)
1.(A)	90	167	90	5	3	3	(5,4,3)
2(B)	85	167	90	8	3	3	(8,4,4)

3(C)	82.5	167	90	11	3	4	(11,4,8)
4(D)	76.23	167	90	13	3	8	(13,4,9)
5(E)	70.6	167	90	20	3	9	(20,4,11)
6(F)	60	167	90	21	3	11	(21,4,13)
7(G)	49.65	167	90	22	3	13	(22,4,15)
8(5)	35	167	90	22	3	15	(22,4,17)
9(0)	35	167	90	22	3	17.5	(22,4,17.5)

CHAPTER 7: CONTROL ALGORITHM AND APPLICATION

7.1 Interfacing the Dc motors with motor controller

The L293D is a Dual H bridge Motor Driver IC.It can control two dc motors simultaneously in either directions.

Things needed:

- motor controller
- 2 DC Motors (3v each)

- Arduino (Mega)
- Jumper wiresr
- Breadboard
- battery
- Battery holder

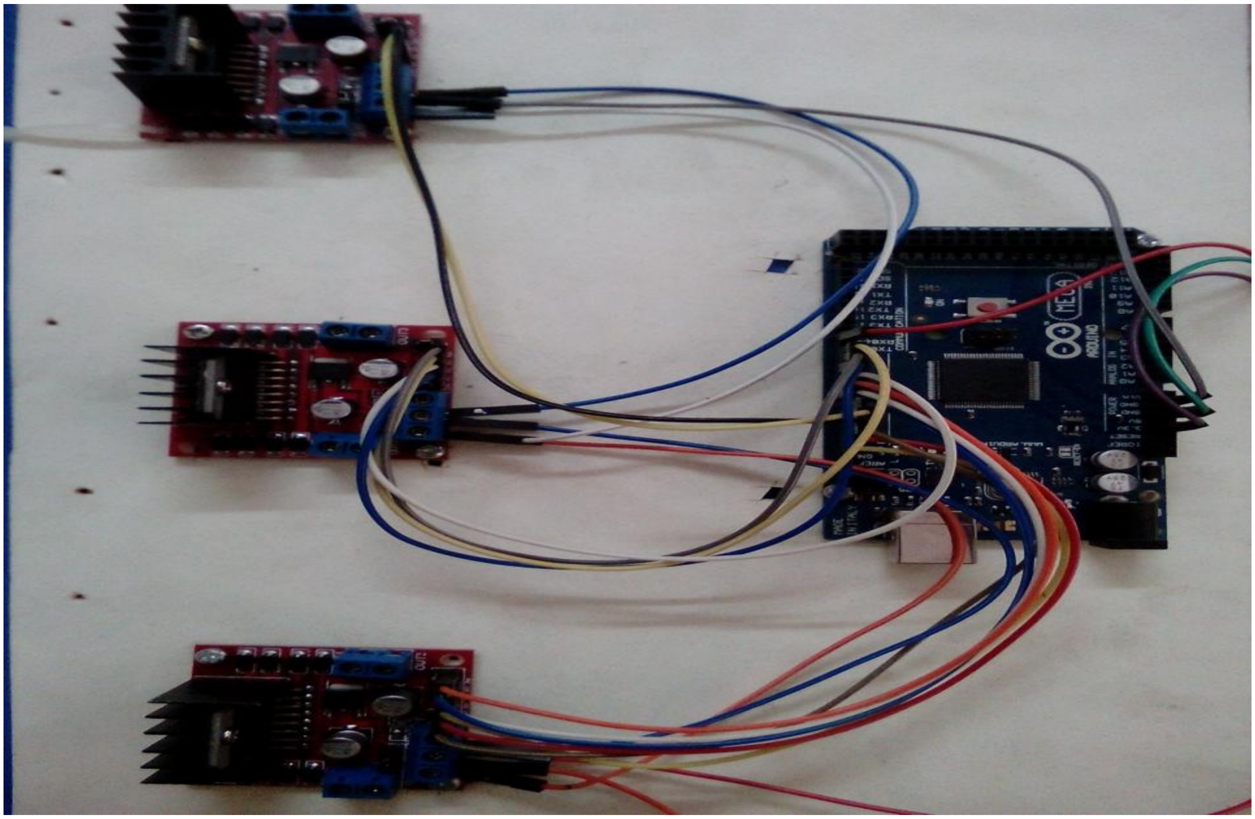


Figure 7.1: Circuit with motor controller and Arduino

7.2 Robot and gripper coding

Now that we know how Arduino coding works with hardware let’s move on to the main part, our project robot car coding.

Since we are using actuators and dc motors we use of the analogWrite() (PWM) functionality on pins 9 and 10, whether or not there is a Servo on those pins. On the Mega, up to 12 servos can be used without interfering with PWM functionality; use of 12 to 23 motors will disable PWM on pins 11 and 12.

Then we define the pin numbers of all of our wheels that are connected with dc motors internally to the Arduino Mega ADK board. As we have four wheels and each wheel have two properties 1.Move forward 2. Move backward. So we need to define 8 pin to control four wheels to move in both direction in forward and backward.

```
intLeftMotorForward = 2;
intLeftMotorReverse = 3;
intRightMotorForward = 4;
intRightMotorReverse = 5;

int LeftMotor2Forward = 6;
int LeftMotor2Reverse = 7;
int RightMotor2Forward = 8;
int RightMotor2Reverse = 9;
```

```
Servo myservo, myservo1; // create servo object to control a servo. (Maximum 8)
```

To control the position of the gripper we define two pos variable such as intpos = 0, pos1 = 0 to zero.

Then in the setup () function we put all the pin mode () function for those 8 Motor variables with the behavior of an output.

```
pinMode(RightMotorForward, OUTPUT);
pinMode(RightMotorReverse, OUTPUT);
pinMode(LeftMotorForward, OUTPUT);
pinMode(LeftMotorReverse, OUTPUT);
pinMode(RightMotor2Forward, OUTPUT);
pinMode(RightMotor2Reverse, OUTPUT);
pinMode(LeftMotor2Forward, OUTPUT);
pinMode(LeftMotor2Reverse, OUTPUT);
```

To attach the servos we need to attach it to an Arduino pin. Since we have two servos in the gripper we do this by:

```
myservo.attach(10);
myservo1.attach(11);
```

Here servo number one is attached in Arduino pin 10 and servo number two is attached in Arduino pin 11.

Now to check if the Bluetooth device is picking up data we did this:

```
if(Serial.available() > 0)  // if data is available to read
int data =Serial.read(); // read it and store it in 'val'
```

Now to drive the robot in we have given value 1,2,3……etc for each function Driveforward(), Reverse(), Rightturn(), Leftturn(), Allstop(), expand(),shrink() etc like this:

```
if(data=='1')
   {
Driveforward();
   }
else if(data=='2')
   {
Reverse();
   }
else if(data=='3')
   {
Rightturn();
   }
else if(data=='4')
     {
Leftturn();
     }
else if(data=='5')
     {
Allstop();
```

```
    }
if(data=='6')
    {
expand();
    }
else if(data=='7')
    {
shrink();
    }
else if(data=='8')
    {
expand_one();
    }
else if(data=='9')
    {
shrink_one();}
```

These values are matched in the android coding, if arduino receives 1 from the android application then it will move the robot forward. All the other functions have same working procedure between Arduino and Android.

We have coded the Driveforward() function like this:

```
voidDriveforward()
  {
```

```
digitalWrite(RightMotorForward, HIGH);
digitalWrite(LeftMotorForward, HIGH);
digitalWrite(RightMotor2Forward, HIGH);
digitalWrite(LeftMotor2Forward, HIGH);
    }
```

Here all the four motors are HIGH, that means all are active. All move in the forward direction.

To turn the car into right we coded that by this way:

```
voidRightturn(){
digitalWrite(LeftMotorForward, HIGH);
digitalWrite(RightMotorForward, LOW);
digitalWrite(LeftMotor2Forward, HIGH);
digitalWrite(RightMotor2Forward, LOW);
 }
```

Here both the left sided wheels are HIGH/active and both the right sided wheels are LOW/inactive. That mechanism makes the robot turn to the right side.

To turn the car into left we coded that by this way:

```
voidLeftturn()
   {
digitalWrite(LeftMotorForward,LOW);
digitalWrite(RightMotorForward, HIGH);
digitalWrite(LeftMotor2Forward, LOW);
digitalWrite(RightMotor2Forward, HIGH);  }
```

Here both the right sided wheels are HIGH/active and both the left sided wheels are LOW/inactive. That mechanism makes the robot turn to the left side.

Now, to move the robot in the reverse direction we did the coding like this:

```
void Reverse()
   {
digitalWrite(RightMotorReverse, HIGH);
digitalWrite(LeftMotorReverse, HIGH);
digitalWrite(RightMotor2Reverse, HIGH);
digitalWrite(LeftMotor2Reverse, HIGH);
   }
```

Here, all four reverse variable for all four wheels are HIGH/active. This mechanism moves the robot in the backward direction.

To stop the robot we need to stop all the four wheels stop. We can do this by this way:

```
voidAllstop()
  {
digitalWrite(LeftMotorReverse, LOW);
digitalWrite(RightMotorReverse, LOW);
digitalWrite(LeftMotor2Reverse, LOW);
digitalWrite(RightMotor2Reverse, LOW);
digitalWrite(LeftMotorForward,LOW);
digitalWrite(RightMotorForward, LOW);
digitalWrite(LeftMotor2Forward,LOW);
```

```
digitalWrite(RightMotor2Forward, LOW);
  }
```

Here, we have made all the 8 variables that move the robot LOW/inactive. This mechanism completely stops the robot.

Now, to expand the finger servo of the gripper to make space for picking up materials we have coded a function called expand().

```
void expand(){
pos = pos+6.0;
myservo.write(pos);
  }
```

Here, we have expanded the finger servo to 6 degree each time we pressed the button assigned to operate this finger servo in the android application.

Now, to squeeze or shrink the finger servo of the gripper to remove space to grab materials we have coded a function called shrink().

```
void shrink(){
pos=pos-6.0;
myservo.write(pos);
    }
```

Here, we have shrinked the finger servo by reducing to 6 degree each time we pressed the button assigned to operate this finger servo in the android application.

Now, to move the wrist servo in one direction either left or right to position the servo so that the materials don’t touch the ground and prevent it from being damaged we have coded a function called expand_one().

```
voidexpand_one(){
   pos1 = pos1+5.0;
myservo1.write(pos1);
}
```

Here, we have turned the wrist servo in one side(left or right) to 5 degree each time we pressed the button assigned to operate this wrist servo in the android application.

Now, to move the wrist servo in other direction either left or right to position the servo so that the materials don’t touch the ground and prevent it from being damaged we have coded a function called shrink_one().

```
voidshrink_one()
{
     pos1=pos1-5.0;
myservo1.write(pos1);
}
```

Here, we have turned the wrist servo in the other side by reducing to 5 degree each time we pressed the button assigned to operate this wrist servo in the android application.

This is all for the Programming of Arduino MEGA ADK to control the robot and the gripper.

Next we’ll see the Android part of our Project.

7.3Communication between Android and Arduino

Android is very popular now, especially ADK (Android Open Accessory Development Kit), which allows external 0pen source hardware to connect with Android system by USB and interact with an Android-powered device in a special “accessory” mode.

Through this ADK Arduino can make communication with Android devices through various wireless media (e.g: Bluetooth, Wi-Fi connection etc). In our Project we have chosen Bluetooth as a wireless medium. An Android App will send data to the Arduino .Internal program of Arduino will decode the data sent by the App and will execute instructions logicwise.

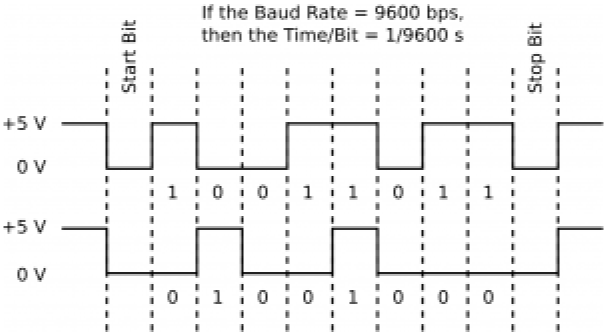
7.3.1 Sending Data: WITH THE HELPING OF ASCII CODE

ASCII (American Standard Code for Information Interchange) is the most common format for text files in computers and on the Internet. In an ASCII file, each alphabetic, numeric, or special character is represented with a 7-bit binary number (a string of seven 0s or 1s). 128 possible characters are defined.

One important concept to understand is that serial information is sent out of and read into the Arduino one byte at a time. Moreover, each byte that is read into memory corresponds to the decimal ASCII encoding for the ASCII character being received. So, if the character A is sent, the value stored in myByte would be 65 since this is the decimal code for the ASCII character A. For more on ASCII characters and serial communications.

Serial communication is a method for two computers or microcontrollers (or one computer and a microcontroller) to talk to one another. Arduino uses the transistor-transistor logic (TTL) serial protocol. This protocol sends bits using voltages of zero (for a zero bit) and five (for a one bit).

Another popular serial protocol is RS-232 which uses +13 V to represent zeros and -13 V to represent ones. In both cases, serial communication involves sending bytes one bit at a time starting with the least significant bit (LSB). Each bit is preceded by a start bit (0 V for TTL serial communication) and immediately followed by a stop bit (5 V for TTL serial communication). Therefore, the figure below shows the signals that would be generated if the bytes 11011001 (top) and 00010010 (bottom) where sent via a serial channel.



The figure above incorrectly shows the stop bit as 0V (low). It should be 5V (high).

In order for two devices to communicate via a serial signal, both must be configured with the same baud rate or bits per second (bps) rate. This tells both machines the interval of time between individual bits so they can synchronize. The most common baud rate for Arduino is 9600. In fact, this is the default baud rate when a new serial monitor is opened from the Arduino IDE. Other baud rates are sometimes necessary. For example, communication with GPS modules often uses a baud rate of 4800.

The following sketch demonstrates serial communication between an Arduino Uno and a the serial monitor on a computer. When a character is sent to the Arduino from the serial monitor on a computer, the Arduino reads this byte and immediately will send it back. The Serial.begin(), Serial.available(), Serial.read(), Serial.print(), and Serial.write() commands are used to start serial communication, test to see if any bytes are available in the serial buffer, read in the next byte in the serial buffer, print a decimal number to the serial monitor, and print the ASCII character corresponding to this decimal number, respectively. The comments in the code should help you figure out what each component does.

.

CHAPTER:8 CONCLUSION

8.1 Discussion

We succeeded in developing an android based 4 wheeler robot with a gripper, controlled with an android application and Arduino MEGA microcontroller via Bluetooth that can move forward, backward, left turn, right turn, stop and carry or load materials through the gripper.

We have studied the communication between Bluetooth device HC-05 and its interfacing with Arduino MEGA ADK and Android application, interfacing of L293D with Arduino MEGA ADK and with Dc motors.

The experiment has integrated a high performance Smartphone with robotics and utilized the Bluetooth technology as a fast, secure and reliable connection between them.

We believe that Android based robotic platforms will be used extensively for multidisciplinary, innovative and affordable projects in research and education. In addition, this approach to robotics will stimulate the creativity of electrical, mechanical and software engineers, as well as robotics students and hobbyists. People with physical limitations such as handicapped people could use the feature from this project to compensate their abilities.

7.2 Present & Future Use of the Arm

We have a plan to add a video surveillance system feature in the robot to capture images and record video on the go. We also have a plan to add voice command in the robot

8.2 Present and Future Work.

8.2.1 In the industries

As it is a robotic arm and can pick and place any material it will be very useful and efficient in the industries. In industries there are some works which is very risky for human being but for a robot to do this work it is quite easy. For example, in big machineries if something stuck it becomes difficult for a human to reach there. On the other hand a robot arm can pick it quite easily.

8.2.2 Moving heavy materials

For moving heavy materials from one place to another a robotic hand can be used as it is quite easy and can be done very quickly.

8.2.3 for collecting fossils

For collecting coal and other fossils human beings need to go there which is very risky for their life sometimes. As a robotic hand has no life and can pick any kind of material which is quite easy for it, the government can use robotic arm to do this type of risky works.

8.2.4In Space

The Shuttle's robotic arm has performed many kinds of tasks over the years. It has set satellites into orbit and retrieved others for repair. The first time Canadarm was used in one of many International Space Station (ISS) assembly missions was for Mission STS-88, December 1998.

The Canadarm of Space Shuttle Discovery with the inspection boom and laser camera system to inspect hard-to-reach areas for damage and help ensure the safety of the astronauts.

So it will provide for future research to continue this project by more researched. Then it could help to explore more flexible arm which could be used in the spacecraft or could be used in many space rover project like:- Mars Rover project etc .Future more research should be benefited by performing flawlessly. It could supports astronauts during spacewalks. If its elbow and wrist joint cameras have provided visual inspection of the Shuttle and payload then it could knocked ice off the Shuttle's wastewater dumping vents and loosened a jammed solar array panel. So future more researched could be benefited in space work also.

8.2.5Human benefit

The robotic technology used provides humanlike dexterity here on Earth in a variety of environments. These may include servicing nuclear power stations, welding and repairing pipelines on the ocean floor, remote servicing of utility power lines, or cleaning up radioactive and other hazardous wastes.

For example, MacDonald, Dettwiler and Associates Ltd. (MDA), which developed Canadarm, has also designed a Light Duty Utility Arm system to inspect and analyze radioactive waste in underground storage tanks. This system consists of a modular, seven-joint manipulator attached to a telescopic vertical positioning mast. A mobile system deploys the manipulator in the tank.

Remotely operated robotic systems have enjoyed wide application in industry and other fields. In medicine, Canada has been a leader in the development of techniques involving robotic surgery operated from a remote location.

8.2.6 A Medical Benefit

The Seaman Magnetic Resonance Centre in Calgary has teamed up with MDA Space Missions to adapt space robotics for use in surgery. The benefits are improved accuracy, efficiency, and the quality of patient care. "NeuroArm" uses miniaturized tools such as laser scalpels with pinpoint accuracy and it can also perform soft tissue manipulation, needle insertion, suturing, and cauterization.

The Centre for Minimal Access Surgery at McMaster University's St. Joseph's Hospital is a tele surgery pioneer. In a successful pilot project, the surgeon directed medical staff from a remote, high technology, operating console and conducted robotic surgical operations on patients hundreds of kilometers away in a hospital in North Bay, Ontario.

References

A. Valera, M. V. (December 12-15, 2005.). *Development of Bluetooth Communications for LEGO-BasedMobile Robot Laboratories.* Spain: 44th IEEE Conference on Decision and Control, and the European Control Conference.

Analysis of Residual Stresses and Distortions in Girth-Welded Carbon Steel Pipe. (May,2013). *International Journal of Recent Technology and Engineering (IJRTE)*, Volume-2, Issue-2.

Anton, H., Bivens, I. C., & Davis, S. (n.d.). *Calculus.* Wiley.

Boldea, L. (1997). *Linear Electric Actuators and Generators.* Canada: The Woodbridge Company Ltd.

Bolton, C., Cantos, M., & Tang, P. (n.d.). *Real-time Control of Robot Arm Based on Hand Tracking Using Leap Motion.* Shanghai: Shanghai University.

Bolton, W. (2011). *Mechatronics: Electronic control systems in mechanical and electrical engineering.*

Goldenberg, A. A., & Fenton, R. G. (March,1985). Goldenberg, Andrew A. A Complete Generalized Solution to the Inverse Kinematics of Robots. *IEEE*, Goldenberg, Andrew A. B. Benhabib; Fenton, Robert G. A Complete Generalized Soluti Vol RA-1. NO.1.

Hibbeler, R. C. (2012). *Engineering Mechanics Dynamics.* Prentice Hall.

Hubert, C. (2002). *Electric Machines: Theory, Operation,Applications,Adjustment & Control.* US: Pearson.

Robert Resnick, J. W. (January 1,1970). *Fundamentals of Physics.* United States of America: Wiley.