WEB BASED REMOTE TRIGGERED ROBOTIC MANIPULATOR USING OPEN SOURCE ARCHITECTURE

An Undergraduate project report submitted to Manipal University in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ECE Engineering

submitted by

Sheerin Sultana   1321008

Under the guidance of

Dr. Ramaprasad Poojary
Professor

Prof. Royson D'Souza
Assistant Professor

SCHOOL OF ENGINEERING & INFORMATION TECHNOLOGY

MANIPAL UNIVERSITY, DUBAI

June 2017

MANIPAL UNIVERSITY
DUBAI CAMPUS

SCHOOL OF ENGINEERING & IT
MANIPAL UNIVERSITY, DUBAI

DECLARATION

I hereby ***declare*** that the Project report entitled **WEB BASED REMOTE TRIGGERED ROBOTIC MANIPULATOR USING OPEN SOURCE ARCHITECTURE** which is being submitted to ***Manipal University Dubai*** in partial fulfillment of the requirements for the award of the Degree of ***Bachelor of Technology*** is a ***bona fide report of the project work carried out by me***. The material contained in this thesis has not been submitted to any University or Institution for the award of any degree.

Sheerin Sultana
Register No.: 1321008

Place: Manipal-Dubai
Date: 11.06.2017

SCHOOL OF ENGINEERING & IT

MANIPAL UNIVERSITY, DUBAI

CERTIFICATE

This is to *certify* that the Project entitled **WEB BASED REMOTE TRIGGERED ROBOTIC MANIPULATOR USING OPEN SOURCE ARCHITECTURE**, submitted by **Sheerin Sultana** (Register Number: 1321008) as the record of the project work carried out by her, is *accepted* as the *Project Thesis submission* in partial fulfillment of the requirements for the award of degree of ***Bachelor of Technology*** in **ECE ENGINEERING** of Manipal University, Dubai during the academic year 2016-17.

DR. RAMAPRASAD POOJARY	**PROF. ROYSON DSOUZA**	**DR. S. K. PANDEY**
Professor	Assistant Professor	Chairperson
School of Engineering and IT	School of Engineering and IT	

ACKNOWLEDGEMENT

I express my special thanks with gratitude and great respect to my valuable guides Dr. Ramaprasad Poojary and Prof. Royson DSouza for their keen interest in my case study, fruitful suggestion and valuable guidance with motivation along with constant encouragement to complete this report. I am also thankful to their great patience, constructive criticism and useful suggestion. I am thankful to all the staff members of Manipal University, Dubai for their support and encouragement. I would also like to thank my friends for extending their wholehearted support. I would like to express my gratitude to my family for their continuous support and help.

ABSTRACT

In recent years, the ease of access to various resources through the internet has become a major boon. Advancements in the field of Internet of Things has given rise to several smart systems and devices. This paper presents open source based remote triggered robotic manipulator which finds its major applications in virtual laboratory. The key purpose of this project is using open source resources to provide virtual platform for performing experiments with robotic arm remotely. The web interface built on Raspberry Pi provides user the control of robotic arm along with live streaming remotely at any place and time. Various experiments are performed on the robotic manipulator to study its statics and dynamics. Students can perform virtual simulation on the experiments and compare the same with the real-time values.
KEYWORDS: Raspberry Pi; IoT; open source; virtual lab; remote access; Python.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

DH	Denavit-Hartenberg
DOF	Degree of Freedom
GPIO	General-purpose input/output
HTML	Hypertext Markup Language
IoT	Internet of Things
PHP	Hypertext Preprocessor
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
PWM	Pulse Width Modulation
RL	Remote Laboratory
SoC	System on Chip

Chapter 1

INTRODUCTION

1.1 Introduction

Open source software and the Internet of Things (IoT) are two terms that have been vastly used these days. The project utilizes the two together to control a robotic arm through the internet. This chapter explains what open source software and IoT are and how they have a wide range of benefits. Their applications in recent years is discussed to demonstrate their features.

1.2 Open Source software

The advancement in technology has led to the development of many open source software. These software can be used by anyone at any place, making it convenient for projects or companies that are underfunded. They can be tweaked according to the requirements of the projects which makes it relevant for a variety of applications. The primary advantage of open source software is that it is free to use. It can be distributed online to everyone and modified easily which helps users to develop their coding skills. The fact that open source software can be easily downloaded, installed and run gives users an easy way to explore and experiment with the latest technologies on the market. This leads to more innovation as the community of user is much larger than those of proprietary software. As anyone can access the softwares code, users can fix the bugs in them, giving it more transparency. This makes the software more secure as compared to other paid software where only the company knows about the bugs they contain. They also do not contain any activation or licenses/serial number. If the company of the software fails, the code still exists and can be developed by the users. It can be updated and fixed faster than the proprietary software available in the market. It can operate and integrate as per the open protocol. It can be incorporated in all divisions such as law, health, education, research and manufacturing.

1.3 Internet of Things

With the increase in the reliance on the internet, there has been a growth in IoT is the interconnection of various electronic devices such as computers, smart phones, tablets through the internet [1]. IoT refers to the networked interconnection of everyday objects, equipped with intelligence. IoT can enable the interconnection and blend of the physical world and the web. The physical world can range from something as small as sensors and actuators to factories and automobiles. These devices are accessed through the internet using their unique IP addresses to transmit and receive data. Although the concept of connecting devices through the internet has existed for decades, this has turned into reality only in the past few years. In 1999, Kevin Ashton, a British tech pioneer used the term Internet of Things (IoT) to describe a system where objects in the physical world could be connected to the Internet by sensors. This term was used to showcase the features of the RFID (Radio-Frequency Identification) chip when it first came out.

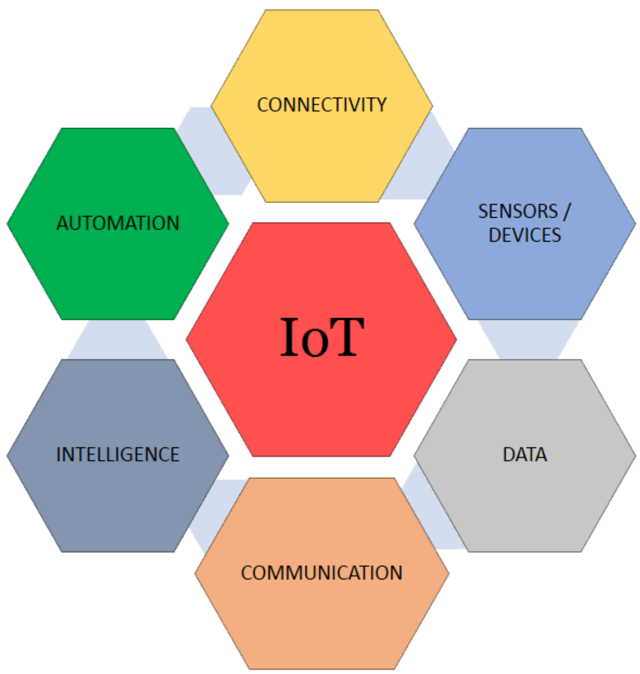

Figure 1.1: IoT characteristics

IoT makes use of various communication models which each have their own distinct feature. The common types of communication include:

- *Device-to-Cloud*
  In this the IoT device connects directly to the cloud service to transmit or exchange data. It can use Ethernet or Wi-Fi connections for transmission. The Cloud service gives the user remote access to the device.

- *Device-to-Gateway*
  In this model, the IoT devices connects to the Cloud service using an intermediary device. The intermediary device can be a hub or even a smartphone which behaves as a local gateway device which provides security and other features such as protocol translation.

- *Device-to-Device*
  It represents two or more devices which connect to each other and can communicate with each other. They can communicate over a variety of networks such as the Internet/ IP networks. This module is used in home automation systems to transfer low amounts of data between devices. Another example is a heart monitor teamed up with a smart watch.

- *Back-End Data-Sharing*
  This module extends the access of a lone device-to-cloud communication model so that the data and IoT devices can be used by third parties. The users can analyse and export the data from the Cloud service and combine this information with data from other sources and send this to other services for analysis.

These models showcase the flexibility in the way IoT devices can connect and give the user the desired experience. IoT has the following important characteristics:

- Common articles are instrumented. It implies that objects such as containers, tables, screws, foods and car tires can be independently tended to by method for being implanted with chip, RFID, standardize tag.

- Autonomic terminals are interconnected. The instrumented physical articles are associated as autonomic system terminals.

As a result of the quick growth in the IoT space, there are several competing tools, standards, policies, projects, organizations, and frameworks hoping to define how connected devices will communicate. Open source and open standards are important in ensuring that devices can interconnect properly and are needed to process the large volumes of data that these devices generate. Without the use of common protocols and open standards, the devices may not be able to communicate with one another. The lack of having a common basis in communication can lead to having your devices to stop functioning altogether. Common frameworks have been created by organizations such as AllSeen Alliance so that devices can communicate with one another irrespective of the manufacturers.

There are billions of devices that are connected to the internet and their number will only increase in the future. These devices generate an immense amount of data that need to be stored and processed. New techniques are required to process such data and retrieve them when required. Open source data tools make this possible along with the advances in machine learning, artificial intelligence and data mining. The security that comes along with open source code makes the devices easy to be tested and inspected. Linux is an example of a secure operating system along with other such systems which can be enhanced for embedded devices to keep the data and the device safe and secure.

1.4 Cloud computing

With the rise in IoT, cloud computing comes into picture. Although, they both seem similar, there are some significant differences between the two. Cloud computing refers to the transfer of data, pictures, documents and other digital material over the internet to data centres. Cloud computing and IoT go hand in hand. The vast amount of information and data generated by IoT devices need to be stored in some place and this is where cloud computing plays a role. Cloud computing is a web based service making it easily accessible to anyone online. Some companies of cloud computing include Dropbox and OneDrive from Microsoft. Cloud computing offers different features such as:

- Sharing information with only specified users.
- Enables multiple connectivity options such as through the laptops, tablet and smartphones.
- It is a measured device which means that the services you receive are based on how many you pay for them.
- It provides flexibility. You can increase storage space, modify your software setup and add/remove the number of users.

Cloud computing offers three types of services: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). Software as a service (SaaS) delivers applications which can be accessed through the web to the end-user. Platform as a service (PaaS) provides a place for users to develop their cloud applications. Infrastructure as a service (IaaS) delivers the software and hardware components on which the user can build a customized computing environment. Data storage resources, communications channel and computing resources are all linked to each other to provide stability to the applications which are used on cloud. Examples of this include smartphones, robots and ATMs [2].

Cloud computing deployment consists of four deployment models: private cloud, public cloud, community cloud and hybrid cloud [3].

- *Private cloud*
  The private cloud is for the sole use of a single organization which comprises of multiple users. They can be managed and owned by the organization or by a third party or both. They have firewall to protect the data and only authorized users can access them.
- *Public cloud*
  This is open to use to the public and is accessed by anyone online. Amazon and Google are examples of this.
- *Community cloud*
  This is managed and used by a certain groups or organizations which have a common interest. Banks are an example of this. They are hosted either internally or externally.
- *Hybrid cloud*
  This is a combination of two or more of the above-mentioned cloud deployment models. In this non-critical data is shared with the public and the important data is kept within the organization.

1.5 Remote triggered lab

A Virtual Learning Environment (VLE) is a framework for conveying learning materials to understudies by means of the web. These frameworks incorporate appraisal, understudy following, joint effort and specialized instruments. This empowers establishments to educate conventional full-time understudies as well as the individuals who can't frequently visit the grounds because of geographic or time confinements [4].

Designing courses can profit by utilizing remote research centers to bolster showing exercises and internet learning. A remote control framework can be a compelling instrument to be utilized as a part of commonsense classes and to improve understudies' trial aptitudes. Online experimentation additionally offers a critical support in building educating, and can be utilized to enhance the understudies' learning procedure [5].

A Remote Laboratory(RL) can be characterized as a situation whose capacity is to control a physical framework remotely, intending to teleoperate a genuine framework, to perform tests and to get to sensors information over the system. Both from an instructive and mechanical point of view, the idea is the same: remotely work plants. This includes pilot plants on account of instructive applications and full-scale plants for modern situations.

This sort of design adds to cooperative research since scientists found geologically indirectly can create control assignments over a similar plant. Besides, RLs are relied upon to encourage access to physical offices to individuals under circumstances of incapacity [6].

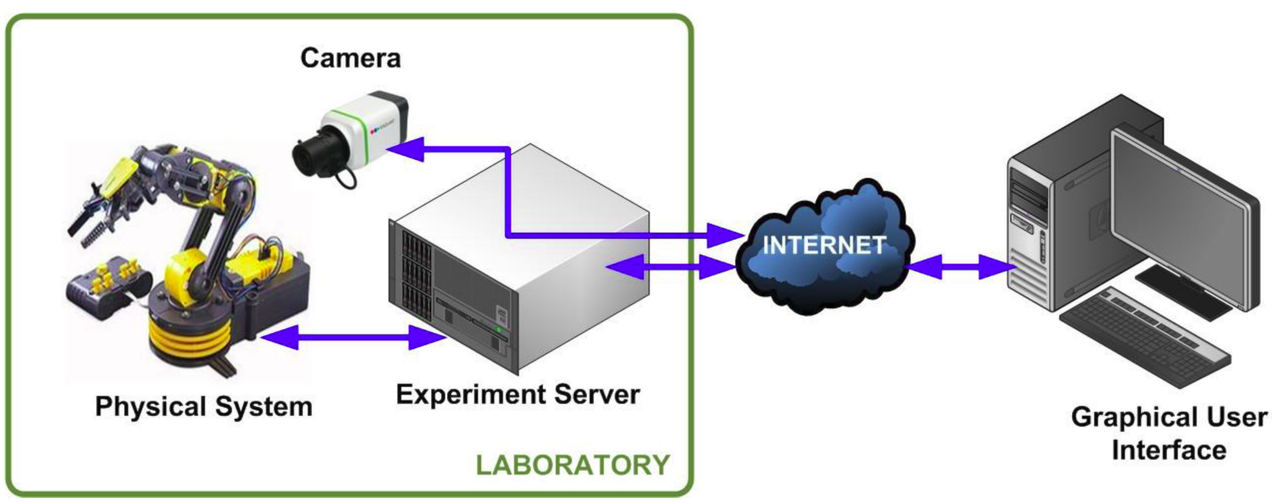

Figure 1.2: Remote Lab[6]

1.6 Applications of IoT

The application of IoT ranges from homes to the health care sector and factories. Some of the recent development are discussed below:

- **Smart homes**
  A smart home is one that consists of devices that can communicate which each other via the internet. They give the owners the ability to manipulate the controls of their home environment as per their desire. This increases the efficiency and security, thereby providing protection and reducing the cost. Companies like Amazon and Philips provide services in building a smart home. Air Quality egg is a sensor based device which monitors the levels of carbon monoxide and nitrogen dioxide in the house. These gases contribute to air pollution. Nest Learning Thermostat is an IoT based thermostat which saves up to 12% on heating and 15% on cooling in the house [7].

- **Wearables**
  Wearables are devices which monitor health and fitness of the user when they wear it. They have many features such as tracking of sleep patterns, activity tracking, food logging and heart rate tracking. Notifications are shown on the users phone and the data is synced on the computer [7].

- **Industry**
  In the construction industry, it is important to determine the quality of concrete. EDC (Embedded Data Collector) is a device which works with embedding sensors during the pouring and curing of concrete to provide information on the quality and strength of the concrete from the work station itself [7].

- **Transportation**
  Self-driving cars are slowing turning into a reality with the help of IoT. Olli, a self-driving shuttle bus created by IBM in 2016 makes use of cloud-based cognitive computing capability for analysing and learning from high quantities of transportation data which are produced by more than thirty sensors embedded all over the vehicle. Passengers can even make conversations with the vehicle on how it works and their destination [8].

- **Healthcare**
  Devices like UroSense help in measuring the core body temperature(CBT) and the urine output of patients on catheterization. This helps in avoiding infections while starting an early care for conditions such as heart failure, prostate cancer and diabetes. It wirelessly provides report data to the nursing stations anywhere [7].

1.7 Projects of IoT and Open source

- **Macchina.io**
  Macchina.io is an open source toolkit that is used to build fog computing and IoT edge appli-

cations which can connect sensors, cloud services and devices. It deals with building applications which run on Linux based devices such as Beaglebone, Raspberry Pi, RED Brick or Galileo/Edison. It can run on a Linux device of 32 MB of RAM as well as on Linux desktops and OS X. It provides a web-based extensible and modular JavaScript and C++ runtime environment for the development of IoT gateway applications which are run on Linux hacker boards. It supports a large variety of sensors and connection technologies. These include GPIO connected devices, GPS receivers and accelerometers [9].

- **GE Predix**
  Predix is a cloud based PaaS (Platform as a Service) software used for industrial IoT. It connects the industrial equipment, analyzes data and gives real-time insights. It helps in developing, deploying and operate industrial applications. Predix provides a secure connection with various machines from different manufacturers in the industrial area by using a varied mix of communication protocols and data to collect together data from these devices [10].

- **Node-RED**
  It is a software development tool which was created by IBM. It is a visual based programming tool for IoT based applications. It is a light-weight runtime which is built on Node.js and takes advantage of its event-driven, non-blocking model making it ideal for running at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud. The programs/flows that are created are stored on JSON and can be imported and exported easily. [11].

Chapter 2

OBJECTIVES AND WORK SCHEDULE

2.1 Objectives

- The primary aim of this project is to provide remote access to a robotic arm where the user can log in through our website and access the simulation to practice on and then move on to perform real time experiments on the robotic arm.
- The result of the project is to calculate the joint angles and the end effectors position using transformation matrix in real time.

2.2 Project Work Schedule

2.2.1 February 2017

- **Arm programming**
  A single motor is controlled using the Raspberry Pi and Python programming.
- **Web page design**
  Website is designed using HTML5, PHP and CSS

2.2.2 March 2017

- **Simulation**
  Blender (simulation software) is used to create the simulation.
- **Control of the six motors**
  The six servo motors are controlled through Python programming.

2.2.3 April-May 2017

- **Embedding the simulation model into the website**
  Simulation is converted to HTML5 format and embedded into the website.

- **Remote access of robotic arm**
  The arm is controlled remotely through the internet by operating the motors

2.2.4 June 2017

- **Documentation**
  Preparation of the thesis and report.

Chapter 3

LITERATURE REVIEW

3.1 Remote Monitoring and Control of Robotic Arm with Visual Feedback using Raspberry Pi

There are various ways to control this robot. It can be controlled from anywhere through the internet. This paper describes how to control the robotic arm in a virtual environment. It becomes an easy task to control the robotic arm from a remote end and anyone can use it for its various applications. Through a PC a webserver can be created and can be controlled from the remote end. Such robots that can be controlled through the internet can be used in many fields like the industry, automation, consumer applications (example: online retail stores) and household applications.

With the Raspberry Pi set as the web server, the users can access the data from the web pages hosted by the sever of the Raspberry Pi. The GPIO pins enable the client computers to communicate with different hardware parts using the 17 GPIO pins where some of them are multifunctional. The server is controlled by the web page that is hosted to control the robotic arm. The internet can be accessed through LAN or Wi-Fi. The Raspberry Pi is connected through the LAN and the PC can be connected through both. The Raspberry Pi requires a HDMI/VGA that can connect it to a LCD monitor.

There is a two-way communication so the data can be send and received from the webpage to the users PC. This executes the python codes and puts the robotic arm into action. Every webserver will have an IP assigned to it. When the client computer enters it, a demand is sent to the webserver for a similar space name [12].

3.1.1 Hardware

Raspberry Pi

The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), 256 megabytes of RAM, later upgraded to 512 MB. The modern webservers run on these combinations: Linux, Apache, MySQL and PHP. They are also called as the LAMP stack.

Robotic Arm

The arm has four rotational joints, which is known as the base, shoulder, elbow and wrist. The base turns the arm around the vertical and the other three rotate it around the x-axis. Since there are no rotations in the y-axis, this restricts the arm movements making the kinematic calculations easier. All joints have limitations while rotating in the backward and forward directions for all the joint parts. Therefore, it is important to know the angle for an application that requires the values of the angles. Through the rotating gear wheels, the gripper opens and closes and move parallel to each other.

3.1.2 Software

DEBIAN is an operating system consisting of a free software. It is the default distribution of the alpha boards. Apache is installed by executing the commands in terminal. This is similarly done to install PHP in the system. When everything is installed, the web server is restarted. The web server files are stored in the default directory /var/www/html. Therefore, whatever is changed, added or modified is placed in this directory which the client will receive upon request.

3.1.3 Implementation

The Raspberry Pi doesnt have ROM memory so a SD card is used for booting the raspberry pi with desired os in order to perform operations. The operating system is first installed on the sd card. The Popular operating systems for Raspberry Pi are Raspbian, Debian, Arch Linux, RISC OS, etc. The default directory /var/www/html can be changed in the apache configuration file. The preferred language that is used here is Python, but it also supports Java. Python programs are coded for the backward and forward direction of each joint and then linked with the PHP script.

3.1.4 Result

When the user presses the buttons, the code is executed for the movement of the joints. The Raspberry Pi therefore controls the robotic arm from the remote end. It can be controlled from any part of the world. There is a live video streaming so the user can view and control at the same time from the webpage.

3.2 Low Cost Automation for Sorting of Objects on Conveyor Belt

Object sorting robotic arm separates a particular object from the bulk of the objects. Doing it manually can be tiring especially when there are a lot of objects or the weight is greater for a person to carry. This kind of automation plays a major role. Raspberry Pi is used, a Linux based board. A Raspberry Pi has a lot of advantages with plenty of applications for a robotic system. The paper describes ways to be a cost-effective object sorting robotic system. The objective is to make it inexpensive, compact and reliable. A camera captures the image of the object which is crucial for the project. The processing of the image is made by a GNU Octave which is an open source language similar to MATLAB. It is portable. All data from the GNU Octave is sent to a microcontroller which in turn sends the information it received to control the robotic arm. The robotic arm then separates the objects to their respective places. Here objects are sorted according to their different shapes and colours. It achieves accuracy and operates at a faster rate which in turn saves time [13].

This arm has a 4-degree freedom of joints and an end effector. The movement of the arm to a particular direction are done by servo motors. Servo motors are in turn controlled by a pulse width modulation(PWM). They turn 90 degrees in both directions totaling to 180 degrees. 360 degrees servo motors are used for advanced applications. The pulse width modulation determines the position of the shaft, during its duration of the pulse it moves to its respective position. The power required for this robotic arm is 55V-7.2V and 8 Amps. Image processing is a form of signal processing. Here the image is the input and is processed and values are obtained.

Raspberry Pi

There are different components that can be used for the image processing but the Raspberry Pi is the most cost effective. Raspberry Pi is flexible and can perform many experiments. Raspberry Pi being open source can make as many changes or modification as needed. Different softwares can be installed for various requirements. It runs on Rasbian operating system and is coded by python and GNU Octave. When the object is captured by the IR sensors, the signal obtained is send to Raspberry Pi and the process of image processing begins. The camera is connected to the Pi and captures the object. To interface Raspberry Pi and servo motor control board an Arduino board is used.

Microcontroller

The microcontroller that is used is Atmega328 on the Arduino board is high performance. It operates with 1.8 to 5.5V.

Image processing

The image is obtained by the camera that is connected to the Raspberry Pi and is saved as jpg file with a resolution of 600x400. When the light is dim or dark then a flash light can be used so a good image can be captured for the processing.

3.2.1 Implementation

The hardware consists of a robotic arm, servo motor, Arduino, Raspberry Pi and a camera. The IR sensor will first sense if the object is present or absent. Once it senses the object it will switch on the camera which in turn captures the project. The saved image is then processed by the GNU Octave and will recognize the colour and shape of the captured object. The Raspberry Pi then sends the data to Arduino which uses the Python script and it further sends a serial message back to the Arduino. The Arduino being an interface between the Raspberry Pi and servo motors through servo controllers will control the robotic arm.

3.2.2 Result

After the processing, the detection of the colour and the shape is given with accuracy but if there is a dullness in the light it can give errors. Therefore, a flash light is used to prevent any future errors that can happen. The flash light is placed right above the object and will shine for 3 seconds while the photo is taken simultaneously. This helps in obtaining a good quality picture and accuracy. 20 to 30 objects can be done per minute and segregate them according to the colour and shape of the object.

3.3 Real time Image Processing based Robotic Arm Control Standalone System using Raspberry Pi

A robot equipped for observation and furthermore with another application in identifying and following a pre-specified object. The entire code for Object detection is written in MATLAB. Processing has been done on raspberry pi which works on Raspbian OS in view of Debian which is Linux OS [14].

To program the controlling of one arm robot utilizing Raspberry pi with no manual control. The program is made in MATLAB. A Simulink support package is used for raspberry pi hardware. The program includes capturing the object image, processing, identifying the green object and controlling of robot arm by using Raspberry Pi. Robot capable of surveillance and also with an alternate application in detecting and following a pre-specified object. Raspberry Pi is useful in real time projects.

The main aim of project is to make a versatile, remote robot fit for taking after a pre-specified object and also can be used in number of useful and versatile applications in robotic systems. The entire code is composed in MATLAB. Raspberry pi changes over the shading picture to a greyscale picture. From the computer vision four unique perspectives: shape, movement, position and introduction are obtained. A Raspbian operating system is used with Raspberry Pi. Open CV libraries takes care of Image processing. The image processing is required for controlling the robotic arm in the pick and place operation of the desired object.

3.3.1 Implementation

Raspberry Pi (B+ model)

Raspberry Pi has pc board with Linux or other working frameworks. It consists of Ethernet port, video and audio output, HDMI output ,2 USB port. The Raspberry Pi requires an external SD card to store the operating system and users data. Raspberry Pi is a powerful microcontroller that can perform almost any functions with the mouse, keyboard and the monitor being connected.

Raspberry Pi NoirCamera

The Raspberry Pi NoirCamera does everything like the regular camera module. It doesnt have the ability to see in the dark because of the lack of infrared lighting. A 5 mega pixel resolution image and a 1080 HD video can be obtained.

Relay

A relay is used as it acts like a switch making it cheaper than replacing a switch. They control one electrical circuit by opening and closing in an another circuit.

3.3.2 Result

Processing the image and for the identification of green object and to track the green object programming with respect to time and distance by varying the threshold value.

3.4 Web-Controlled Surveillance Robot with A Robotic Arm

A web controlled surveillance robot is a robot that can be controlled through the internet wirelessly. There are also cameras mounted on the robot for video surveillance. This can be used as a multimachine. The core objective of the robotic arm is to be observe the international borders for any illegal and suspicious activities and other violations [15].

This project uses a Raspberry Pi with a Linux based ARM11 processor and Rasbian OS. It is user friendly, compact and portable. The reason behind the selection of the Raspberry Pi is with respect of its speed as well as being inexpensive. It can perform various application. A microcontroller is used for providing the PWM output that helps the servo motors to control the motion of the robotic arm in the desired direction and the DC motors are controlled by the Raspberry Pi. Robotic arm mimics the actions of a human arm. These motions are possible with the help of 5 motors -3 servo motors and two DC motors. The robotic arm consists of an elbow, wrist, shoulder pivot and an assembly of a similar looking human finger.

3.4.1 WebIOPi

WebIOPi is a software used to control, debug and uses the Raspberry Pis GPIOs from anywhere in the world and from any browser through the internet. It is first installed by typing sudo python-m webiopi. The pins can be accessed by entering the IP address of the Raspberry Pi from any browser. All the GPIOs are used as output ports.

3.4.2 Motion

Motion is a streaming software and is used to obtain live feed video using the Raspberry Pi. It is installed using sudo apt-get install motion. To start the webcam server sudo service motion start is entered in the terminal. The live feed can be accessed from any browser by entering The Raspberry Pi address it has obtained in the URL bar.

3.4.3 Motors

The motor driver IC L293d has 2 H-bridge circuits that can drive 2 motors at the same time. The input decides the direction of the motor. Here there are 6 DC motors, 4 of which is attached as wheels for the motion of the robot. One DC motor is used for the rotation of the robotic arm and one for the picking and placing operations of the robotic arm. The motors that are placed at the same side of the robot receives the same signal at any given time. Therefore, it only controlling two motors at the same time by giving the same signal. The shoulder, elbow and wrist of the robotic arm that uses the servo motors for motion are attached to the pins of the microcontroller.

3.4.4 Conclusion

The robotic arm is multifunctional and can be used for picking and placing objects. It can used in automation and heavy machinery industries and it can also have other purposes like spying and bomb defusing.

3.5 Internet controlled robotic arm

The main objective of the robots are not limited in industries, they can be used to help humans in their work in industries or day to day chores. Robots are increasingly replacing human beings because the perform more efficiently and accurately with repetitive tasks [16].

This paper describes the movement of robot over the internet. The robot is controlled by Arduino Uno and uses an Arduino Ethernet shield to interface through the internet. By entering the desired position and the robotic arm will move to the desired direction of the inputted value. With the click of the button the robotic arm can also move to an already preprogrammed movement. The robot does not require any prior trainings as it completely manually controlled by the user.

A webserver is developed using HTML so that the user can operate the robotic arm through the internet.

3.5.1 Design

The robot consists of a base, shoulders, elbow, wrist and a gripper. The power source for the robot is a Lead Acid battery(12v/1.2Ah) and has a lot of advantages. The power is regulated to 5 volts with a voltage regulator LM78XX. A servo motor is used with a feedback and controls the motion of the robotic arm. Here a servo motor 180 is used because the project requires high torque.

Acrylic is used as the base of the robot because it is inexpensive, robust and can take the weight of the motor. The gripper and the servo brackets of the robotic arm is made of aluminium as thy are light weight and rigid.

3.5.2 HTML-Kit

Using HTML-Kit program a webserver can be created with HTML coding. The full featured HTML editor edits, formats, validates, previews and publishes web pages in HTML,XHTML and in XML languages.

3.5.3 Implementation

There are two parts in this system the computer system and the robotic arm. The Arduino Uno controls the entire system. The Arduino Uno interfaces the internet through the Arduino Ethernet Shield, then the Arduino Ethernet Shield enables the Arduino Uno to connect to the internet through LAN cable. The user can access the robot and control it through internet. A relay infrastructure is used to connect the robotic arm to the internet and it enables secure web access to the embedded systems behind a firewall. The servo motor is first positioned and the Arduino Uno waits for the command.

3.5.4 Conclusion

The paper describes how a robot is not limited to industrial work but also can be used for household activities. The robot can be accessed by anyone and be controlled via the internet. The hardware operations of the robot consists of controlling through servo motors and the development of the robotic arm. The software development consists of the development of the web server and programming the Arduino Uno.

3.6 Web Based Simulation and Remote Triggered Laboratory for Robots

A remote triggered virtual lab is developed using industrial robots such as Movemaster RM-501 and Puma 560. The programming languages used LabVIEW along with open source software such as JavaScript and Python. The objective of the project is to enable users to get an understanding in robotics as well as control systems. The concepts in these two fields are demonstrated in the virtual labs through a website [17].

3.6.1 Software

The development of the virtual lab made use of many different tool. These include: MATLAB/MATLAB Builder JA, Solidworks, WAMP AND LAMP, JSP, ZoneMinder, Python, servlets and JavaScript. The kinematics of robots in a 3-dimensional simulation is shown by first developing the models in Solidworks. This is then imported into MATLAB. The simulation of the model is then established using functions in MATLAB. The functions are constructed to output the result in the form of text and images. For the web interface, J2EE and Java programming have been used. ZoneMinder is used for video feed where the video can be recorded, captured and analyzed.

3.6.2 Hardware

Puma 560 is a 6-degree freedom robot. Each of its joints are separately controlled through brushed DC servomotors. The feedback generated from the servos are retrieved from a potentiometer and 500-1000 count 3-channel encoder. Through the feedback the robots global position can be identified. The Mitsubishi Movemaster RM-501 robot is an arm with 5 rotational joints. They are controlled through DC motors. Each motor has two encoders. Both Puma 560 and RM501 can be controlled remotely through the web. The website shows the simulation along with the real robot being controlled.

3.6.3 Kinematics

Forward Kinematics involves calculation of the end effector position using the joint angles. In the simulation, the initial position i.e. the base position is known. The user selects any one of the joint angles and specifies its value for the simulation. Using the value of the angle, the transformation matrix is generated and shown on the website. The end effector position changes as per the matrix and this is also shown on the website. In inverse kinematics, the joint angles are calculated using the end effector position.

3.6.4 Implementation

MATLAB Builder JA has many features such as: sorting data, data conversion and array shape. In MATLAB, the fundamental concepts in robotics are programmed and then saved as functions. The functions generate an image or text as an output. The two robots used are controlled by using Python, WAMP and JavaScript. Serial port RS232 is used to create a connection between the robot and the server. Using this, the robot can be actuated. Serial read and serial write on the robots controller is done using Python scripts. The website has PHP, JavaScript and Ajax embedded into it. The Python script at the server is called when the user clicks on a button or command is typed out on the website.

3.6.5 Result

This project uses MATLAB Builder JA and J2EE technology in creating a virtual lab where users gain insight on robotic concepts like kinematics. The robots give a real-time view of they work. The lab gives user flexibility and enough time to analyze the concepts by using the learning material provided.

3.7 Smart Phone Based Robotic Arm Control Using Raspberry Pi, Android and Wi-Fi

This paper describes how to control a robotic arm using an application android platform. They are connected through Wi-Fi. The android application is the command center of the robotic arm. Android applications are composed in the Java programming language and utilize Java core libraries [18].

Robots are separated into two categories industrial and service robots. Service robots perform semi or fully autonomous operations. The robot body is manufacture mechanically and electrical parts were likewise used to fabricate the automated arm. For the most part, the web controlled robots are wired and these wired robots have some space limitation. So to avoid the restriction, the mechanical control is made remote that is, connected via WiFi.

The automated arm has four rotational joints, the base, shoulder, elbow and wrist. The base turns the arm around the vertical z-axis, while the other three turn it around the x-axis. The arm's rotational speed is most certainly not steady for example, turning the shoulder joint downwards will take less time than turning it upwards by the same amount. As battery fades so does the robotic arms speed.

A signal is generated on clicking a specific button on the android application which enables the raspberry pi to make the arm move as indicated by the predefined program. Android application is an application work in the android platform. All smart phones broadly use the android application. Android application being the command center of the arm as it summons the arm to move or grab. The command is passed to the arm through the android java and the Raspberry Pi. It has all elements which is accessible in PCs and portable workstations. Through the wifi the Raspberry is connected to the android but the app cant receive or send information. Therefore, two methods can be followed to connect to the Raspberry Pi:

- Router method: Through a common router the Raspberry Pi and android phone are connected.

- Wifi Direct method: Here the information are directly interchanged via wifi between the Raspberry Pi and android phone.

This arm consists of 5 motors and 3 motor driver ICs which controls the robotic arm. The motor driver ICS used is an L293D that interfaces with the GPIOs pins of the Raspberry Pi. Each driver IC can control 2 motors. 15 GPIO pins of Raspberry Pi are used to control the robotic arm. All motors use 5V to work the robotic arm.

3.7.1 Conclusion

The Raspberry Pi can be utilized for the control of a Robotic Arm with Smartphone from a remote zone. The present situation web controlled robot has a few inconveniences for example, wired limitations and server issues. All these issues are solved here as wifi is the fastest usage of the internet.

3.8 From remote experiments to web-based learning objects: an advanced telelaboratory for robotics and control systems

This paper describes the advancement of the telelaboratory facilities. This virtual environment web based access can be used as simulation environment and in physical experiments. The tremendous effect of data advances on regular day to day existence has profoundly influenced the conventional teaching and learning processes. Students have an advantage of distance learning and exploit 3-D tools, web video and audio streaming. The huge number of technological devices and the spreading of networking allow remote users to access the equipment and control remote and virtual environment. That a realistic virtual condition can have a few advantages over a remote lab as far as preparing execution. Experiments such as pick and place are conducted in these telelaboratory. It is proposed to utilize MATLAB as the standard control configuration device [19].

While the particular applications, might be extremely different, a typical component of the vast majority of the telelaboratories created inside the academic environment is that they contain an accumulation of tests accessible to the users is free, informal and is not in structured way. This is done to help students to learn through trial and error method and have a hands-on learning experience.

Clients are made a request to execute controllers for single inputsingle output (SISO) and multiple inputmultiple output (MIMO) plants, that are simulated with Real Time Application Interface Linux threads.

Each test must be a secluded, compact, and independent LO. The move from a (virtual or real) experiment to an independent instructional unit (i.e., LO), represents extra demand on the association of the telelaboratory. The web cam that enables students to see the real-time output of the examination.

3.8.1 Conclusion

For the student, an environment is created to improve their skills and learn through trial and error method. All the templates are structured the same. An essential issue is that the student does not need to sit around idly learning the robot proprietary language to move the arm, yet can just utilize exceptionally straightforward symbol programming rules. Students are shown the forward and inverse kinematic of a serial controller using an automated arm. Students are not required to learn any programming such as MATLAB or Scicos, for example. They can however can simply rotate a few handles and see the output in real time.

3.9 Current trends in remote laboratories

Hands-on research facilities are the most well-known forms of such laboratory environment, offering students openings of experimentation with systems related to academic material. The difference between the physical and the web based simulator is not made aware to the user via internet. The Remote-laboratory architecture relies on the clientserver where Internet advances are in the focal point of current advancement systems[20].

3.9.1 Remote laboratory Applications for Engineering Education and Research

- *Electronics and Microelectronics*
  The recent developments in technologies raise the standards for education and research especially in the field of electronics and microcontrollers. Even though breadboards and other wiring are considered a part of basic training the change in skills in digital design that is more effective with its use in time and other forms of training.

- *Robotics*
  They consist of a combination of software and hardware platforms with commonly used programs like MATLAB, LabVIEW, C/C++ and Java. All of them involving the mobile robotics to enhance the teaching and learning experience. The abilities of the web based robotics to create 2D and 3D simulations to achieve the grippers position that is inputted according to the user. Basic and advanced experiments can be performed like forward and reverse kinematics.

- *Power Electronics and Electrical Drives*
  Education in this field has a foundation in an extensive variety of disciples, like physics, electrical circuits, analog and digital electronics. Most power electronics laboratory that are remote or physical use the electrical drives. These have gained more attraction among the students and academics due to the growing need of alternative energy resources.

3.9.2 Conclusion

This paper describes the remote laboratories and related technologies with particular areas of the industrial electronics. There are many benefits that pertain to the given remote laboratory for all academics. Providing access for developing countries is also a major goal that has to be met.

Chapter 4

METHODOLOGY

4.1 Introduction

The robot is controlled over a wireless network through a web browser. The six servos attached to the arms joints uses Python scripts for the movements. Robotics simulation software is used in 3D modeling and rendering in a virtual environment that imitate the real environment of the robot. To achieve communication between the computer and the robot the solution has a hardware interface consisting of the Raspberry pi development board. This system is flexible enough to give students experience with many levels of motion control. The Raspberry pi provides platform to set up a web server and provide control through the HTML page. The 3D model simulation of the robotic arm is built on Blender.

4.2 Architecture

The proposed project work comprises of following components:

- Robotic Arm: Experiments are performed by operating 6 stepper motors individually with 5V, 1A specification.
- Raspberry Pi-3B: Apache server is used for executing programs based on which the required control signals are given to servo motor. Python 3.4 IDE is used for coding.
- Servo motor driver: Used for providing 5V, 1A to each servo motor.
- USB camera: Providing live feed to enable remote access.

The design architecture consists of a robot consisting of 6 servo motors providing 6 degrees of freedom(DOF), connected to Raspberry Pi. The power for each motor is supplied from Braccio [21] motor driver with individual control signals from Raspberry Pi. A web camera is connected to Raspberry pi provides live feed for real time experimentation. The registered users have access to the robot through the virtual platform and control the actions and direction of the robot along with various features that are embedded in the webpage.

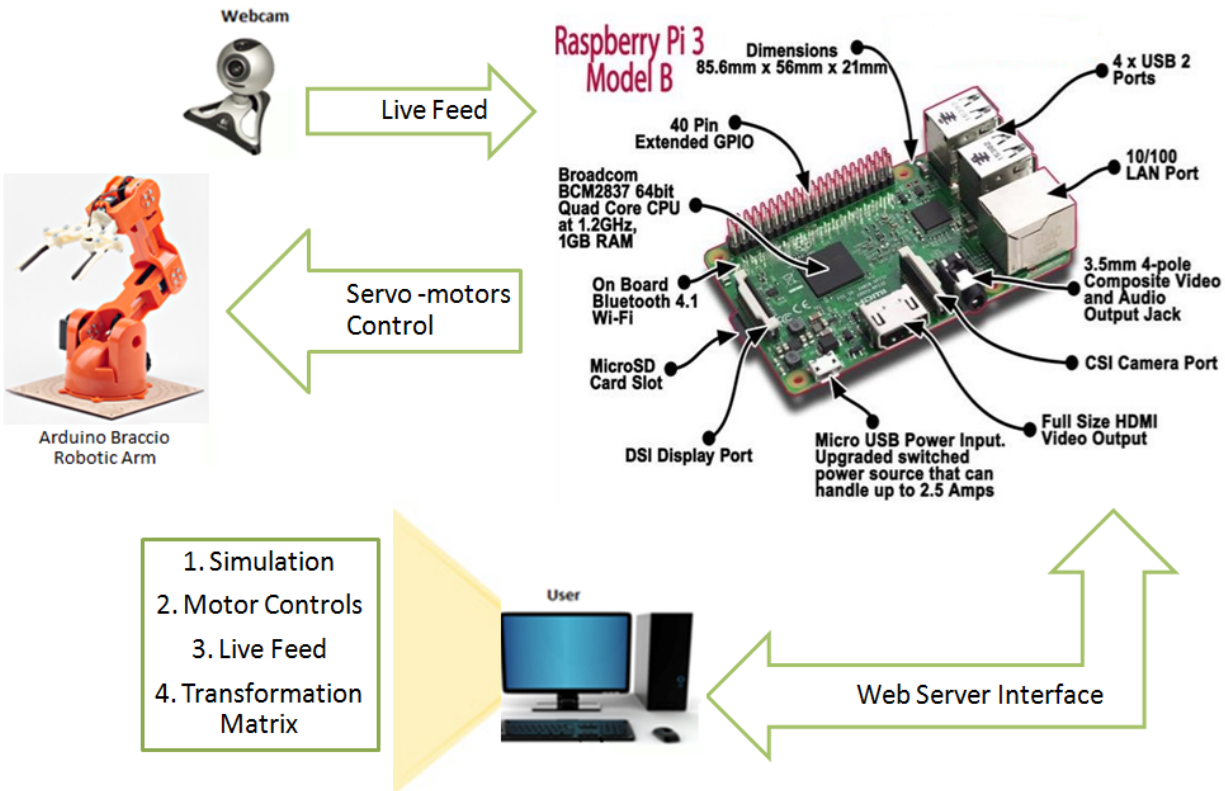

Figure 4.1: Architecture

4.3 Phases of Project

The following concepts and softwares were used in process of undertaking this project. Following are the phases followed for project:

1. Introduction to Python 3.47 IDE, Blender, HTML, PHP
2. Programming robotic arm with python and PHP
3. Interfacing driver module for servo motor with Raspberry Pi-3B
4. Construction of simulated blender model
5. Deploying server on Raspberry Pi-3B
6. Webpage development
7. Integration of simulation and hardware controls and troubleshooting

4.3.1 Introduction to Python 3.4 IDE, PHP, Blender

- Python is easy to learn, fast to code, and readable than C. Module time is imported for proving delay to the motors between change in pulse width. It is used for controlling the servo motors. Python code can run everywhere since its a free software. Further it works on Windows, Linux and OS X.
- PHP (Hypertext Pre-processor) used for creating buttons on web page and executing python program to provide servo motor control.
- Blender is the simulator used to load design files, animate any design, add effects and visualize robot projects. Blender uses real physics with animations to simulate your robot so you can verify that it operates as expected. It allows project modeling and programming robotic arm. It is used for creating simulated model and performing experiments in simulation.

4.3.2 Programming robotic arm with Python and PHP

1. **PHP code:**

```
<?php
if (isset($_POST['M1R']))
{
exec('sudo python /var/www/html/program/m1r.py');
}
if (isset($_POST['M1L']))
{
exec('sudo python /var/www/html/program/m1l.py');
}
?>
```

The PHP coding given below executes the specific python program:

```
<?php
exec(program.py)
?>
```

2. **Python code for increasing angle:**

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12,GPIO.OUT)
```

```
GPIO.setwarnings(False)
z = GPIO.PWM(12,50)
file = open("/home/pi/Documents/values/test.txt","r")
m = file.read()
file.close()
n = float(m)
n = n + 0.6
if(n>12):
n = 12
s = str(n)
file = open("/home/pi/Documents/values/test.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(2)
GPIO.cleanup()
```

3. **Python code for decreasing angle:**

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(12,50)
file = open("/home/pi/Documents/values/test.txt","r")
m = file.read()
file.close()
n = float(m)
n = n - 0.4
if (n<2):
n = 2
s = str(n)
file = open("/home/pi/Documents/values/test.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(2)
GPIO.cleanup()
```

- Modules imported in script:
  - import Rpi.GPIO : module provides access to configure Raspberry Pis I/O pins.
  - import time: to provide sufficient delay
- Pulse width modulation is done by varying duty cycle:
  - To read the duty cycle value from text file
    - f=open(file.txt,r)
    - file.read()
  - To write the duty cycle value into the file
    - f=open(file.txt,w)
    - file.write()
  - float(value/variable) : conversion from string to int/float to vary duty cycle
  - str(value/variable) : conversion from int/float to string to write into file

4.3.3 Interfacing driver module with servo motors and Raspberry Pi-3b

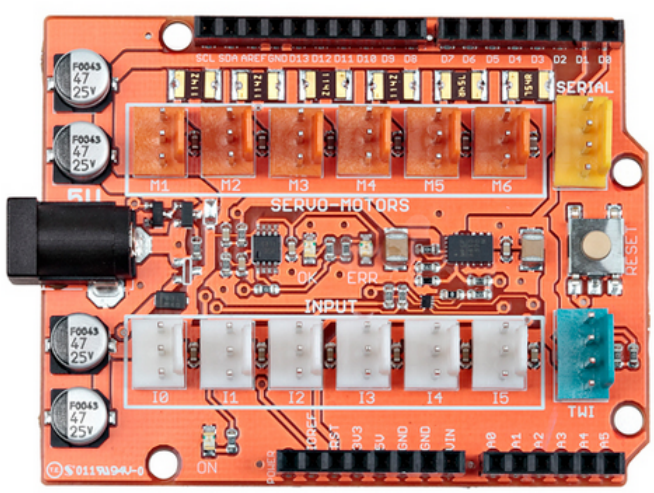

Figure 4.2: Driver module

Each driver port consists of 3 pins, 5V, ground and control signal. Motor ports 1-4 provides output of 5V, 1.1A and motor port 5-6 provides output of 5V, 750mA. Control signal is generated by Raspberry Pi. The following GPIO board pins are used as control pins:

- motor 1 - pin 35
- motor 2 - pin 36

- motor 3 - pin 37
- motor 4 - pin 38
- motor 5 - pin 16
- motor 6 - pin 40

4.3.4 Construction of simulated model in Blender

Blender being a free and open source 3D creation suite supports the entirety of the 3D pipelinemodeling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation [22]. It supports python programming. As a result, programming can be done to create modification in the model.

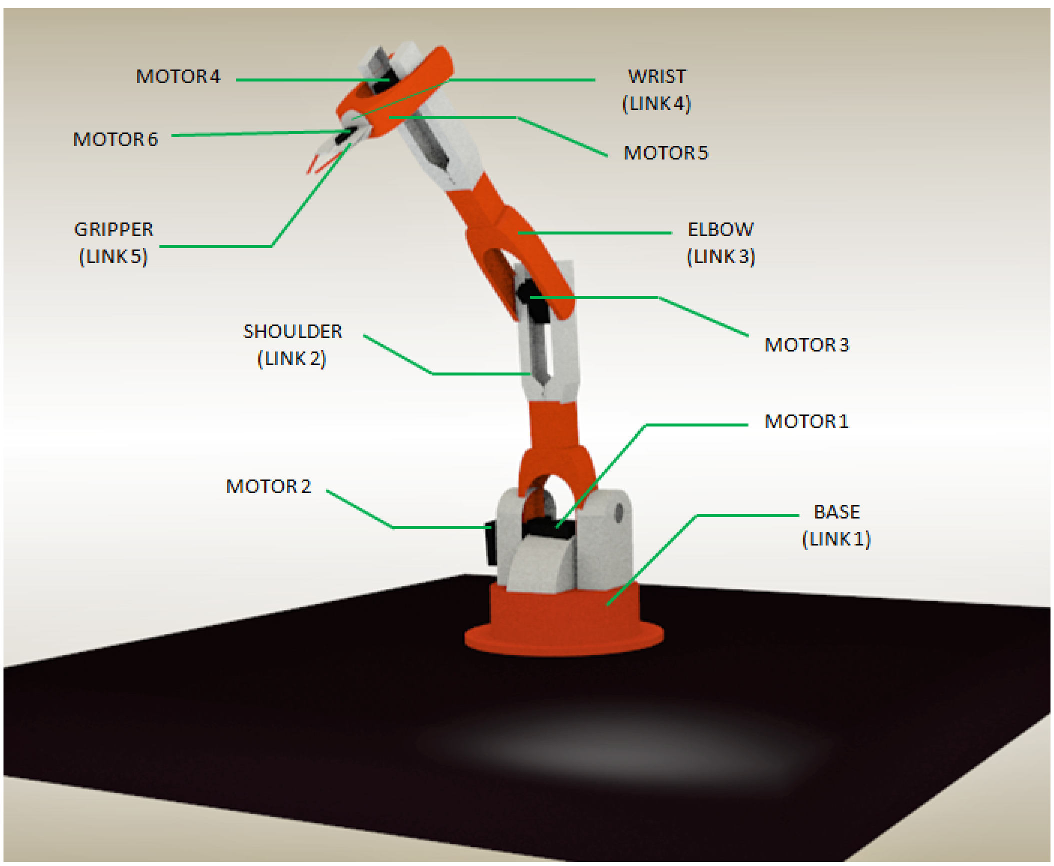

Figure 4.3: Blender render model

Following are the modifiers used:

- Duplicate
- Boolean

- Mirror
- Extrude

Only one object can be selected at a time and can be edited in Edit Mode. The model can be transformed using scale, rotate and size. The default shapes used were: plane, cube, circle, UV sphere and cylinder. By combining these with the modifiers the robots structure was created. The colours of the parts and background can be changed. The background colour is used to enhance the lighting on the model and increase visibility to the user.

Each motor is parented to control other parts of the robot as shown below:

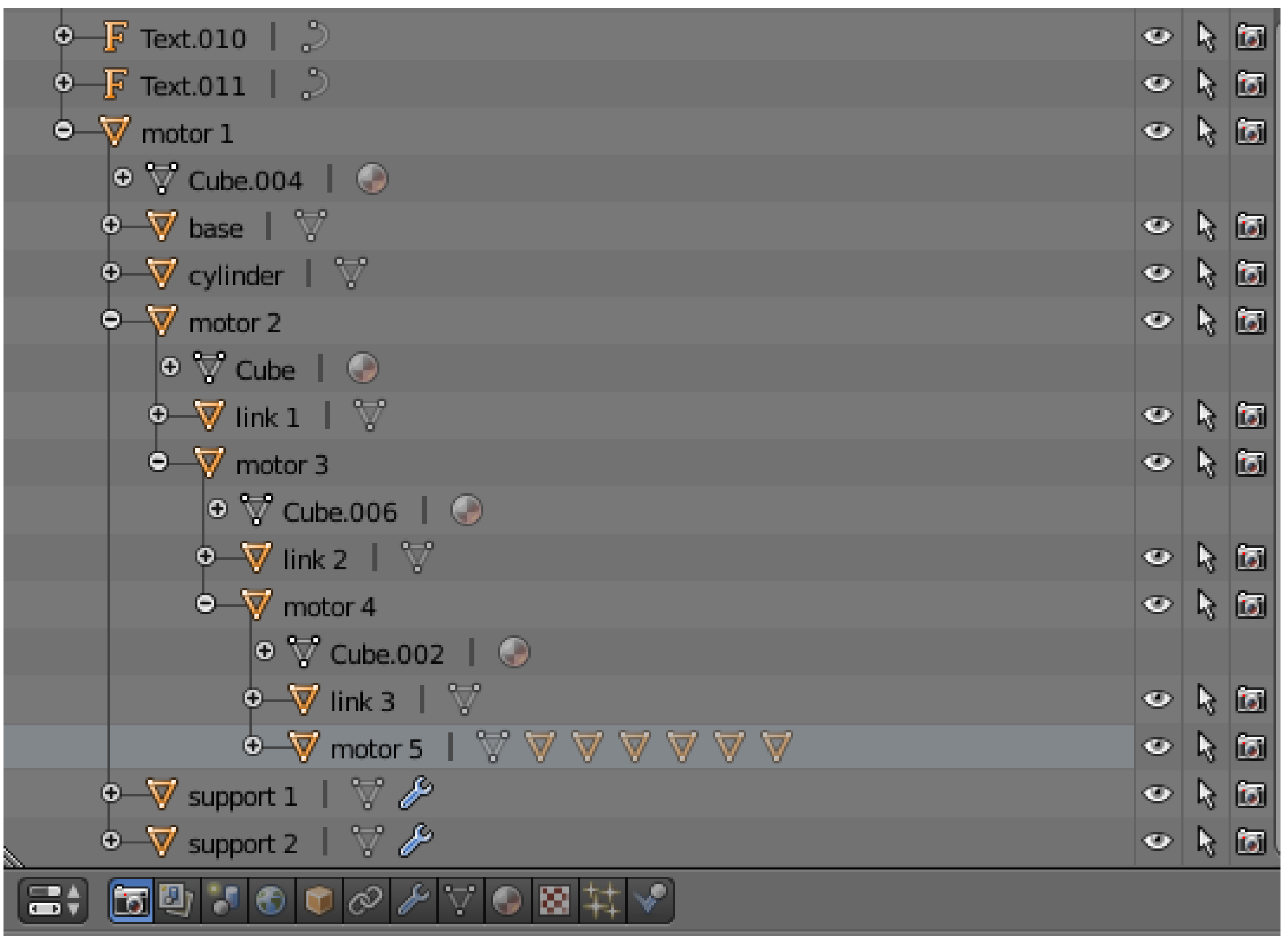

Figure 4.4: Parent Hierarchy

The model is controlled using buttons made in Blender. Each motor is linked to two controls. One control to move in the forward/upwards direction and the other to move it in the reverse direction. The motors are linked through a node tree in the Node Editor mode. In this the axis of movement is specified.

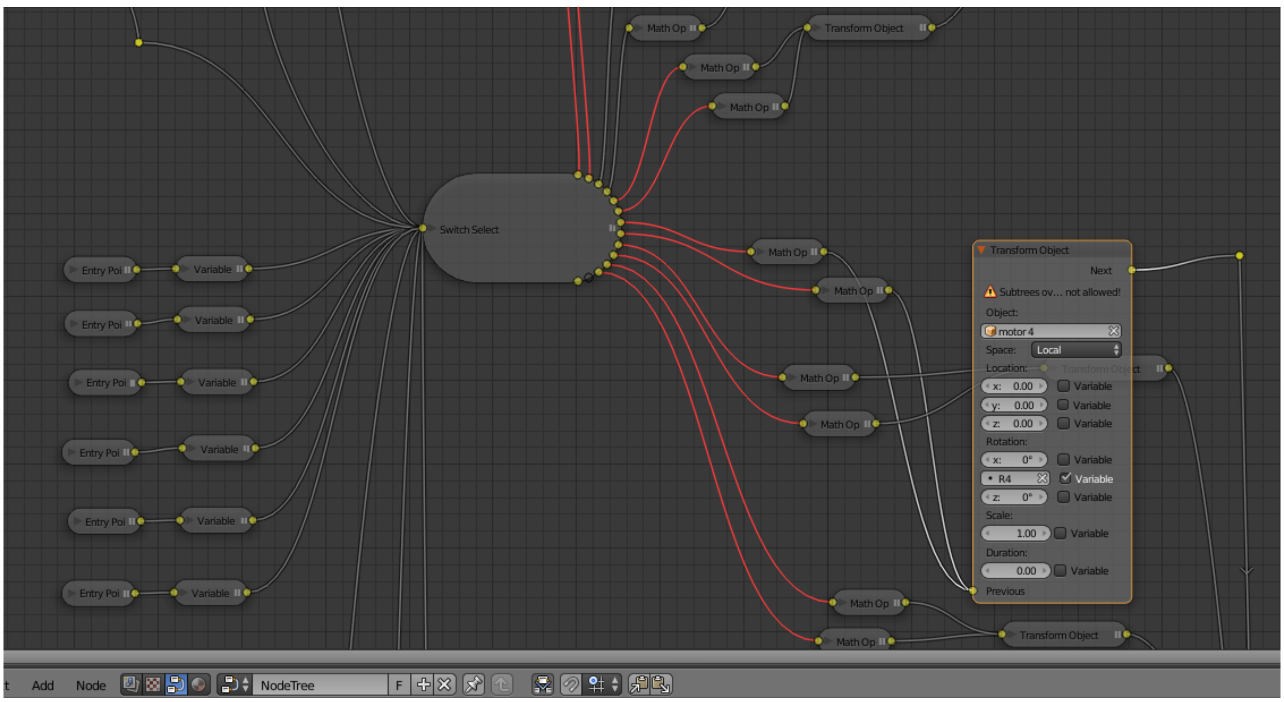

Figure 4.5: Node Tree

4.3.5 Deploying server on Raspberry Pi-3B

Apache HTTP Server, colloquially called Apache, is free and open-source cross-platform web server software. Apache and PHP Installation steps:

```
sudo apt-get update && sudo apt-get upgrade
```

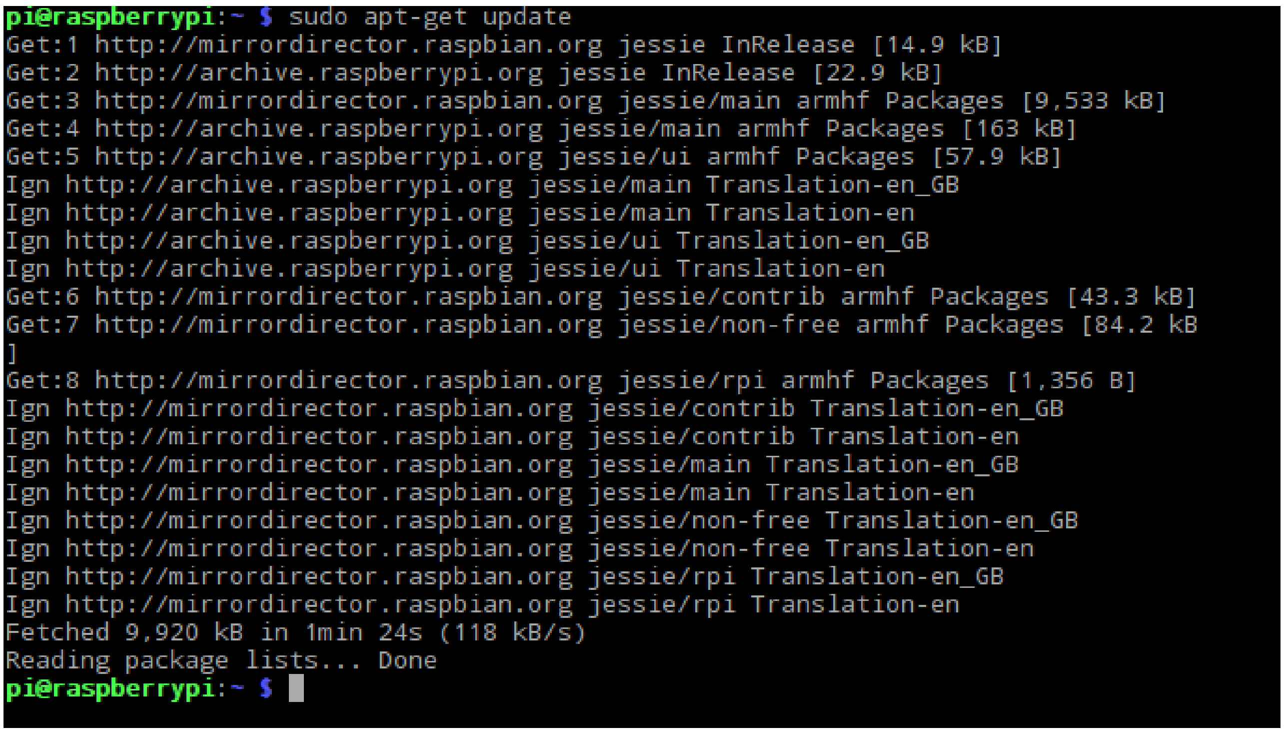

Figure 4.6: Step-1

```
sudo apt-get install apache2 -y
```

```
pi@raspberrypi:~ $ sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,467 kB/1,759 kB of archives.
After this operation, 5,258 kB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ jessie/main apache2-bin armhf
 2.4.10-10+deb8u8 [901 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ jessie/main apache2-utils arm
hf 2.4.10-10+deb8u8 [195 kB]
Get:3 http://mirrordirector.raspbian.org/raspbian/ jessie/main apache2-data all
2.4.10-10+deb8u8 [162 kB]
Get:4 http://mirrordirector.raspbian.org/raspbian/ jessie/main apache2 armhf 2.4
.10-10+deb8u8 [208 kB]
Fetched 1,308 kB in 10s (122 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libapr1:armhf.
(Reading database ... 112842 files and directories currently installed.)
Preparing to unpack .../libapr1_1.5.1-3_armhf.deb ...
Unpacking libapr1:armhf (1.5.1-3) ...
Selecting previously unselected package libaprutil1:armhf.
Preparing to unpack .../libaprutil1_1.5.4-1_armhf.deb ...
Unpacking libaprutil1:armhf (1.5.4-1) ...
Selecting previously unselected package libaprutil1-dbd-sqlite3:armhf.
Preparing to unpack .../libaprutil1-dbd-sqlite3_1.5.4-1_armhf.deb ...
Unpacking libaprutil1-dbd-sqlite3:armhf (1.5.4-1) ...
Selecting previously unselected package libaprutil1-ldap:armhf.
Preparing to unpack .../libaprutil1-ldap_1.5.4-1_armhf.deb ...
Unpacking libaprutil1-ldap:armhf (1.5.4-1) ...
Selecting previously unselected package liblua5.1-0:armhf.
Preparing to unpack .../liblua5.1-0_5.1.5-7.1_armhf.deb ...
Unpacking liblua5.1-0:armhf (5.1.5-7.1) ...
Selecting previously unselected package apache2-bin.
Preparing to unpack .../apache2-bin_2.4.10-10+deb8u8_armhf.deb ...
Unpacking apache2-bin (2.4.10-10+deb8u8) ...
```

Figure 4.7: Step-2

```
sudo apt-get install libapache2-mod-php5 php5
```

```
pi@raspberrypi:~ $ sudo apt-get install php5 libapache2-mod-php5 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libonig2 libperl4-corelibs-perl libqdbm14 lsof php5-cli php5-common
  php5-json php5-readline
Suggested packages:
  php-pear php5-user-cache
The following NEW packages will be installed:
  libapache2-mod-php5 libonig2 libperl4-corelibs-perl libqdbm14 lsof php5
  php5-cli php5-common php5-json php5-readline
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 5,087 kB of archives.
After this operation, 18.8 MB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ jessie/main libonig2 armhf 5.
9.5-3.2 [101 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ jessie/main libperl4-corelibs
-perl all 0.003-1 [43.6 kB]
Get:3 http://mirrordirector.raspbian.org/raspbian/ jessie/main lsof armhf 4.86+d
fsg-1 [321 kB]
Get:4 http://mirrordirector.raspbian.org/raspbian/ jessie/main php5 all 5.6.30+d
fsg-0+deb8u1 [1,318 B]
Get:5 http://mirrordirector.raspbian.org/raspbian/ jessie/main libqdbm14 armhf 1
.8.78-5+b1 [86.0 kB]
Get:6 http://mirrordirector.raspbian.org/raspbian/ jessie/main php5-common armhf
 5.6.30+dfsg-0+deb8u1 [721 kB]
Get:7 http://mirrordirector.raspbian.org/raspbian/ jessie/main php5-json armhf 1
.3.6-1 [16.9 kB]
Get:8 http://mirrordirector.raspbian.org/raspbian/ jessie/main php5-cli armhf 5.
6.30+dfsg-0+deb8u1 [1,911 kB]
Get:9 http://mirrordirector.raspbian.org/raspbian/ jessie/main libapache2-mod-ph
p5 armhf 5.6.30+dfsg-0+deb8u1 [1,874 kB]
Get:10 http://mirrordirector.raspbian.org/raspbian/ jessie/main php5-readline ar
mhf 5.6.30+dfsg-0+deb8u1 [11.2 kB]
Fetched 5,087 kB in 39s (128 kB/s)
Selecting previously unselected package libonig2:armhf.
(Reading database ... 113530 files and directories currently installed.)
Preparing to unpack .../libonig2_5.9.5-3.2_armhf.deb ...
Unpacking libonig2:armhf (5.9.5-3.2) ...
Selecting previously unselected package libperl4-corelibs-perl.
Preparing to unpack .../libperl4-corelibs-perl_0.003-1_all.deb ...
Unpacking libperl4-corelibs-perl (0.003-1) ...
Selecting previously unselected package lsof.
```

Figure 4.8: Step-3

Motion web server is used for live streaming. Motion server Installation steps:

```
wget https://github.com/ccrisan/motioneye/wiki/precompiled/
ffmpeg_3.1.1-1_armhf.deb
dpkg -i ffmpeg_3.1.1-1_armhf.deb
```

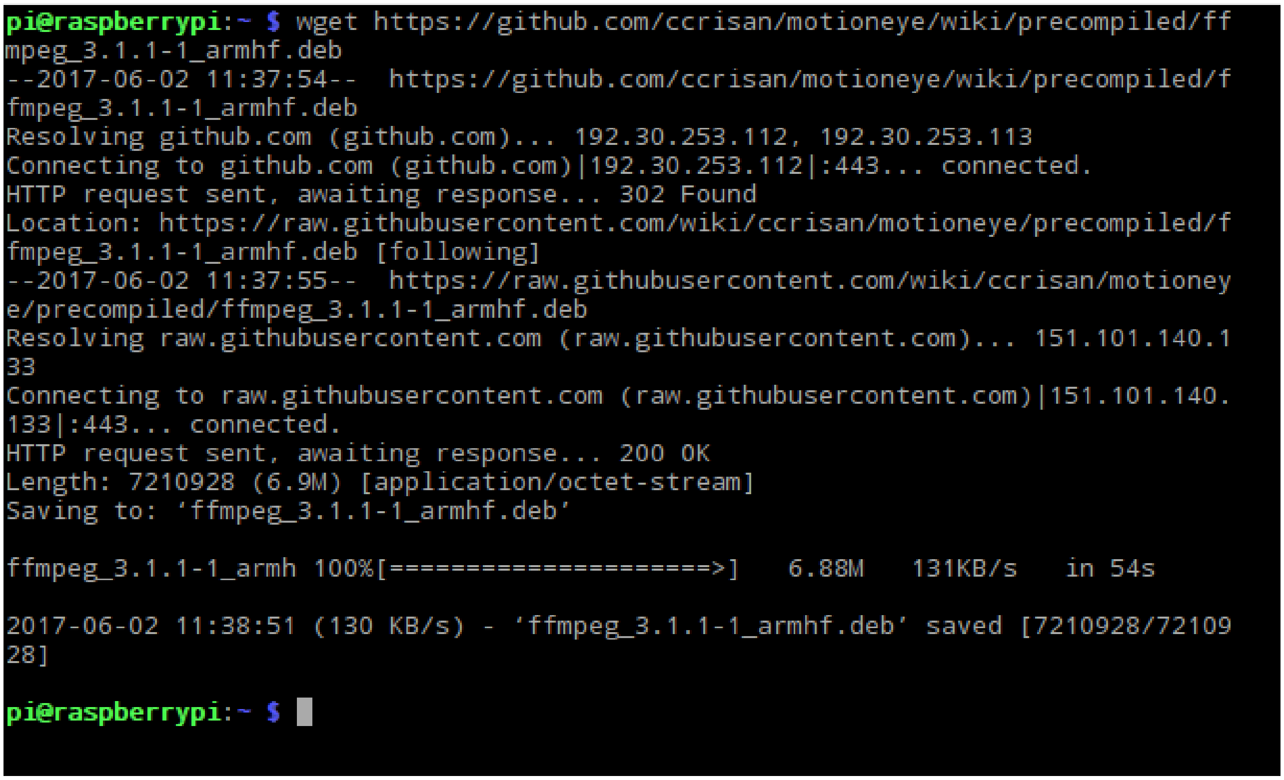

Figure 4.9: Step-1

```
sudo dpkg -i ffmpeg_3.1.1-1_armhf.deb
```

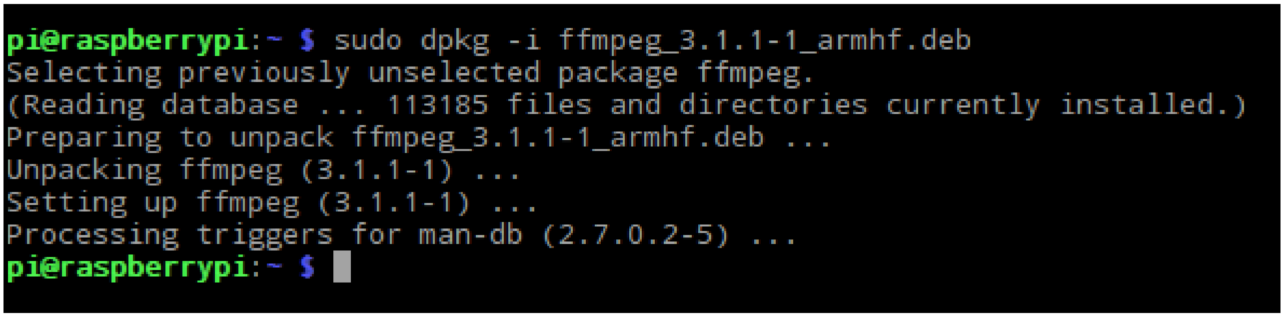

Figure 4.10: Step-2

```
sudo apt-get install curl libssl-dev libcurl4-openssl-dev
libjpeg-dev libx264-142 libavcodec56 libavformat56
libmysqlclient18 libswscale3 libpq5
```

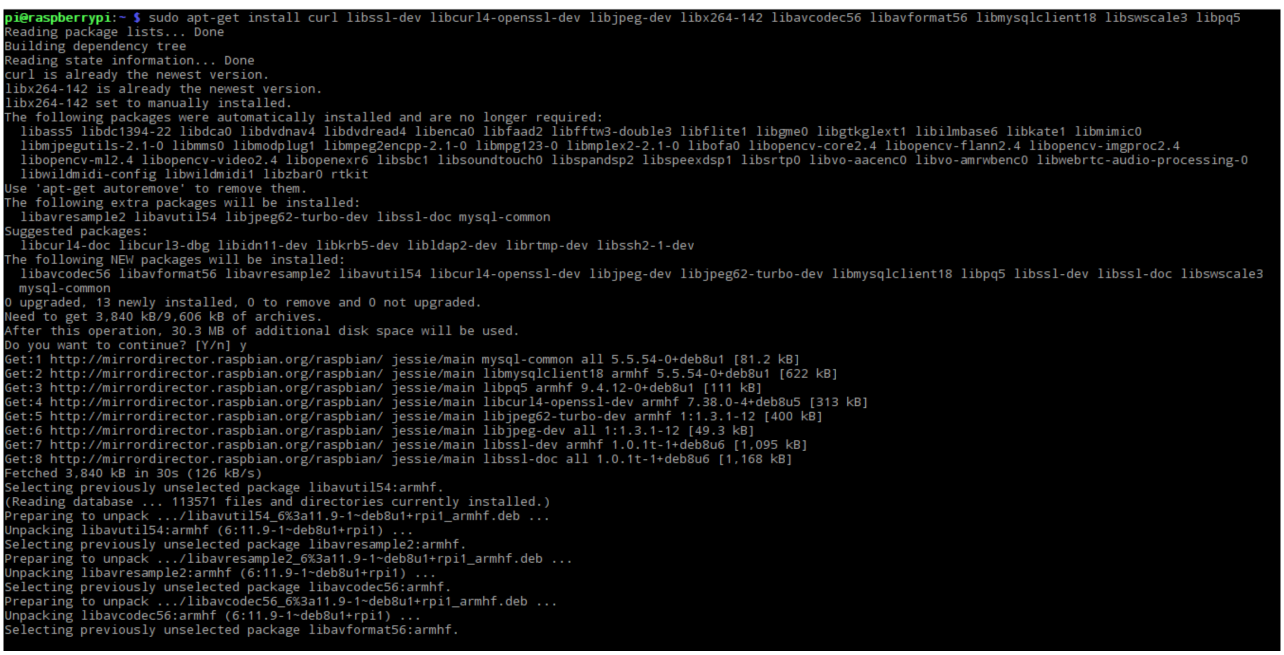

Figure 4.11: Step-3

```
wget https://github.com/Motion-Project/motion/releases/
download/release-4.0.1/pi_jessie_motion_4.0.1-1_armhf.deb
```

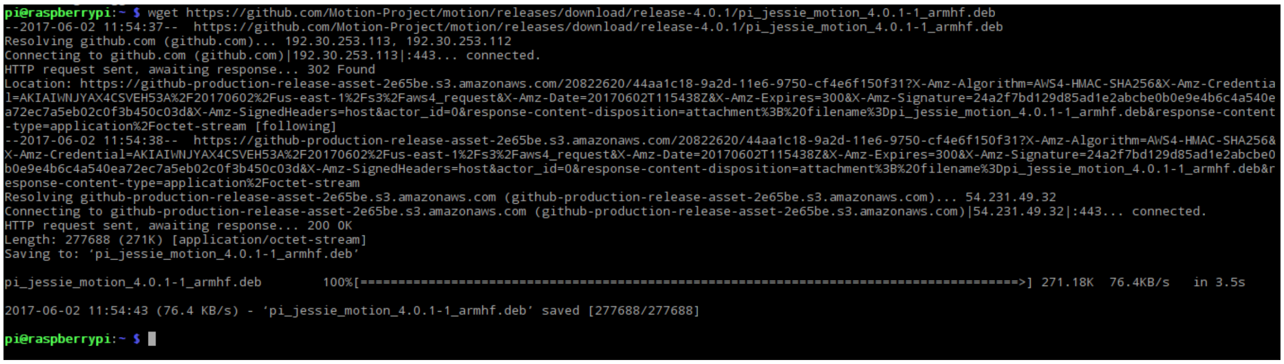

Figure 4.12: Step-4

```
sudo dpkg -i pi_jessie_motion_4.0.1-1_armhf.deb
```

4.3.6 Web page development

The website is designed using Adobe Dreamweaver CS6. The HTML5 programming, Cascading Style Sheets (CSS) and JavaScript in the code mode as well as the design mode was used together to build the website. HTML5 is used to give the overall structure of the website. CSS is used to customize the website according to preferred colours and fonts.

HTML code:

```
<html>
<body>
<img src="arm.png" alt="Arm" style="width:540px;height:428px; ">
<form method="post">
  <table
 style="width: 75%; text-align: left; margin-left: auto; margin
 -right: auto;"
 border="0" cellpadding="2" cellspacing="2">
<tbody>
   <tr>
     <td style="text-align: center;">Rotate Left</td>
     <td style="text-align: center;">Rotate Right</td>
  </tr>
  <tr>
<td style="text-align: center;"><button name="M1L">motor1
decrease</button></td>
```

```
<td style="text-align: center;"><button name="M1R">motor1
increase </button></td>
  </tr>
</tbody>
  </table>
</form>
</body>
<iframe width=’370’ height=’305’
img src=http://172.17.42.155:8085 ></iframe>
</html>
```

Figure 4.13: Control buttons

Controls from web inputs is provided through control buttons that were created. These are used to increase or decrease duty cycle.

The simulation model is embedded into web page by converting it into a HTML file using Blend4Web which is an open source software [23]. It behaves as an add-on in Blender and thus, makes the transition to the web page much easier. The model can be previewed in JSON format and can be exported into a HTML file. The user can move around the scene using the arrow keys on the keyboard. WASD keys are used to zoom in and zoom out as well as rotate. Other features can be accessed by clicking the gear icon. M1, M2, M3, M4, M5, M6 represent the motor controls.

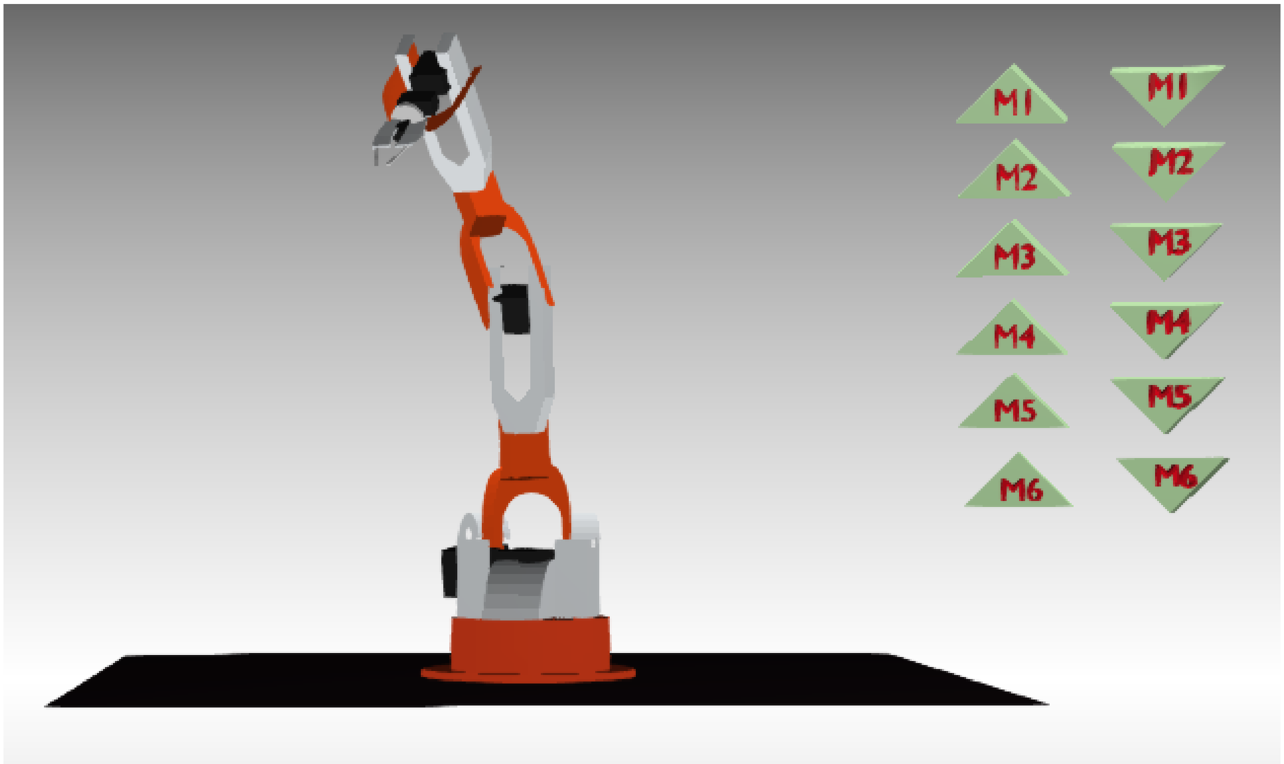

Figure 4.14: Simulation model in HTML file

4.4 Mathematical modelling

It is necessary to study the kinematic performance of a robot when creating its mathematical model. The kinematics models deal with the robots movements while eliminating the forces generated during its movement. Kinematics of a robot involves the study of how the different links of the robot move with respect to each other as well as with time. The spatial relationship of the robot is analyzed which involves the relation between the orientation and position of the end-effector and its joint variables. Kinematics has two sub-divisions: Forward kinematics(FK) and Inverse Kinematics(IK). In Forward Kinematics, the joint angles are known and the end-effector position is calculated. In Inverse Kinematics, the end-effector position is known and the joint angles are calculated.

4.4.1 Kinematic Model

The robotic platform used in the work is a 6 degree of freedom(DOF) robotic manipulator developed by Arduino. The arm has been extensively used in research, development and teaching. It is essentially a serial manipulator where all its joints are revolute. The arm has a geometrical configuration is made up of elbow, waist, shoulder, and wrist. Except for the wrist, the remaining joints have a single DOF. The wrist can move in two planes (roll and pitch), thereby making the end-effector more flexible in terms of object manipulation. The arm is fully-actuated with each DOF achieved by a precise servo motor. The end-effector is a two-state gripper having rubber pads.

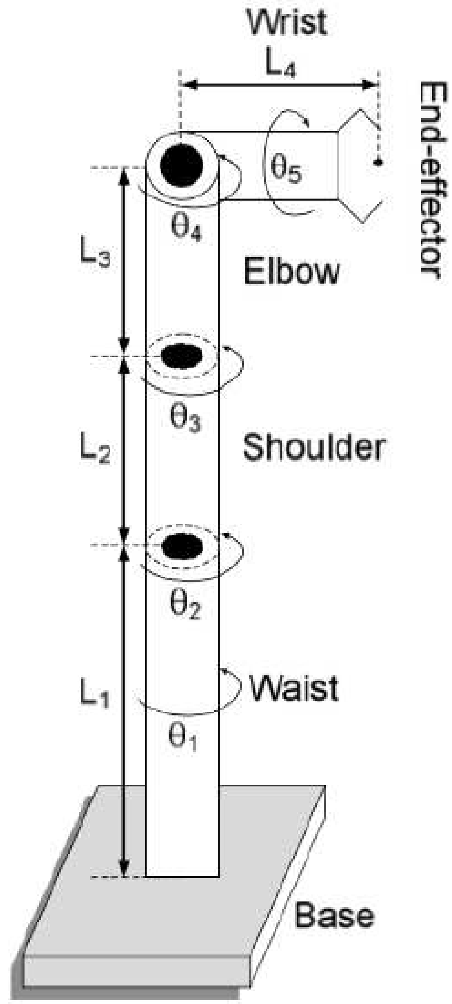

Figure 4.15: Kinematic model

Table 4.1: Manipulator Link Lengths

Joint	Symbol	Link length(cm)
Waist	L_1	17
Shoulder	L_2	17
Elbow	L_3	8.5
Wrist	L_4	13.5

4.4.2 Forward Kinematic Model

The study of kinematic problem of a robot can be performed using various methods. Two of the widely used methods are based on Denavit-Hartenberg (DH) parameters and successive screw displacements. Both methods are systematic in nature and more suitable for modeling serial manipulators. DH method has been used to develop the kinematic model of the robot in this work because of its versatility and acceptability for modeling of any number of joints and links of a serial manipulator regardless of complexity.

A simplified kinematic model of the robotic arm is illustrated in Figure 4.15. The first three joints are used to move the tool point to its desired position, while the last two joints adjust the orientation of the end-effector. Table 1 shows the link lengths mentioned in Figure 4.15.
Transformation Matrix = R$_{z,\theta i}, Trans_{z,di}, Trans_{x,ai}, R_{x,\alpha i}$

$$\begin{bmatrix} C_{\theta i} & -S_{\theta i} & 0 & 0 \\ S_{\theta i} & C_{\theta i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_{\alpha i} & -S_{\alpha i} & 0 \\ 0 & S_{\alpha i} & C_{\alpha i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Table 4.2: Servo Angle

Servo	Position	Angle
1	Base	0°-180°
2	Waist	15°-165°
3	Shoulder	0°-180°
4	Elbow	0°-180°
5	Wrist	0°-180°
6	Gripper	10°-73°

Table 4.3: DH Parameters

Symbol	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
$_{i-1}$	0	-90°	0	0	-90°	0
a$_{i-1}$	0	0	L$_2$	L$_3$	0	0
d$_i$	L$_1$	0	0	0	0	L$_4$
θ_i	θ_1	$\theta_2 - 90°$	θ_3	θ_4	θ_5	0

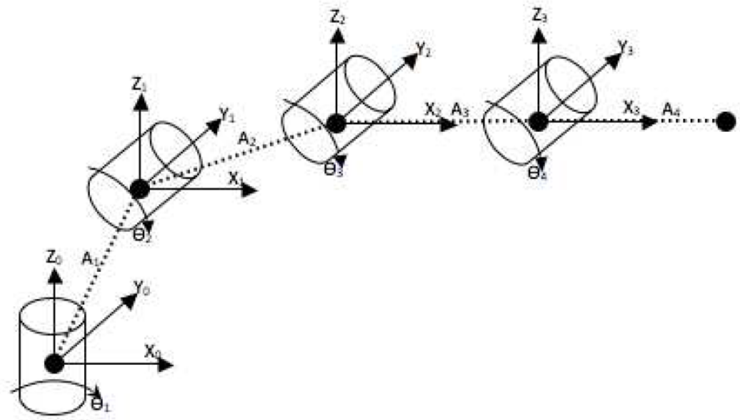

Figure 4.16: Coordinate assignment

$$\text{Transformation Matrix T} = \begin{bmatrix} C_1C_5S_{234} + S_1S_5 & -C_1S_5S_{234} + S_1C_5 & C_1C_{234} & C_1A \\ -S_1C_5C_{234} - C_1S_5 & S_1C_5C_{234} + C_1C_5 & S_1C_{234} & S_1A \\ C_5C_{234} & -S_5C_{234} & -S_{234} & B \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A = $L_2S_2 + L_3S_{23} + L_4C_{234}$
B = $L_1 + L_2C_2 + L_3C_{23} - L_4S_{234}$

The 3X3 matrix comprising of first three rows and first three columns is the rotation while the last column represents the position (x, y, z) of the end-effector with respect to base.

4.5 Work envelope

A space on which a robot can move and work its wrist end is called as a work volume. It is also known as the work envelope or the work space. For building up an efficient work, a portion of the physical attributes of a robot should be carefully looked after:

- The structures of different robots
- The maximum life for moving a robot joint
- The size of the robot segments like wrist, arm, and body

The working envelope refers to the working volume which can be reached to by some point toward the finish of the Robot arm, this point is generally the focal point of the end effector mounting plate. It limits any devices or workpiece which the end effector may hold. There are territories inside the working envelope which can't come to the end of the Robot arm. Such territories are named dead zones. The greatest cited payload limit can be accomplished at certain arm traverses, this may not really be at the most extreme reach.

The work volume of a jointed arm arrangement robot is complicated. The wrist and elbow of a controller are swept in a horizontal and vertical position. It works practically like a human arm. The outcome accomplished from the jointed arm and cylindrical design robots are same. The real benefit of the jointed arm arrangement robot is that it can move every way effectively and adaptably. It is utilized for performing machine stacking and emptying operations in the CNC machines.

4.5.1 Revolute Configuration

The Braccio arm design has an expansive working envelope in respect to the floor space it possesses. The state of the working envelope relies on the individual plan. This design helps in obtaining a sphere.

Chapter 5

RESULT AND FUTURE SCOPE

5.1 Result

The completion of the project has resulted in the following:

- The Raspberry Pi has been configured as per the requirements of the project. It is interfaced with the Braccio arm along with the servo motors.
- A website has been created for the user to register and log in to access the robotic arm through the site.
- The Python programs are executed in the website and the user is able to control the arm virtually.
- A live feed is generated simultaneously alongside the arm controls to help the user to see the robot and control it.
- The virtual simulation of the arm is embedded into the website.
- The website can be accessed on PCs and phones from anywhere in the world.

5.2 Conclusion

This project aims to be of aid to those who have an interest in robotics and want to have a better understanding of how a robot works. Students will be benefitted from this the most as the web-interface gives you a real-time experience. Those who live in a place where access to a working robot is unavailable will also benefit from this project.

5.3 Future scope

The future scope of this project will be:

- To control the robotic arm using the simulation model instead of the Python scripts in real time.
- Generate graphs to show the speed of the arms movements and the distance it covers each time it moves.
- Incorporate colour detection and sort objects by colour.

References

[1] A. Deendayal Patil, H. Imtiyaz Kadiri, A. Shriram Joshi and A. B Wani, ”IOT Based Remote Access Human Control Robot Using MEMS Sensor”, *IJCSMC*, vol. 5, no. 3, pp. .816 826, 2016.

[2] [Online]. Available: https://web.archive.org/web/20150317032921/cloud.cio.gov/topics/cloud-computing-service-models. [Accessed: 04- Jun- 2017].

[3] [Online]. Available: https://www.ibm.com/developerworks/community/blogs/722f6200-f4ca-4eb3-9d64-8d2b58b2d4e8/entry/4_Types_of_Cloud_Computing_Deployment_Model_You_Need_to_Know1?lang=en. [Accessed: 04- Jun- 2017].

[4] [Online]. Available: http://global.oup.com/uk/orc/learnvle/. [Accessed: 17- Jun- 2017].

[5] A. Cardoso, V. Sousa and P. Gil, ”Demonstration of a remote control laboratory to support teaching in control engineering subjects”, *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 226-229, 2016.

[6] I. Gonzlez, A. Caldern, A. Mejas and J. Andjar, ”Novel Networked Remote Laboratory Architecture for Open Connectivity Based on PLC-OPC-LabVIEW-EJS Integration. Application in Remote Fuzzy Control and Sensors Data Acquisition”, *Sensors*, vol. 16, no. 12, p. 1822, 2016.

[7] [Online]. Available: https://www.ibm.com/blogs/internet-of-things/iot-applications-industries/. [Accessed: 04- Jun- 2017].

[8] [Online]. Available: https://www-03.ibm.com/press/us/en/pressrelease/49957.wss. [Accessed: 04- Jun- 2017].

[9] [Online]. Available: https://macchina.io/. [Accessed: 04- Jun- 2017].

[10] [Online]. Available: https://www.predix.io. [Accessed: 04- Jun- 2017].

[11] [Online]. Available: http://nodered.org/. [Accessed: 04- Jun- 2017].

[12] R. OommenThomas and K. Rajasekaran, ”Remote Monitoring and Control of Robotic Arm with Visual Feedback using Raspberry Pi”, *International Journal of Computer Applications*, vol. 92, no. 9, pp. 29-32, 2014.

[13] V. Pereira, V. Amsdem Fernandes and J. Sequeira, "Low Cost Automation for Sorting of Objects on Conveyor Belt", *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 5, pp. 195-200, 2016.

[14] P. Hemalatha and C. Lakshmi, "Real time Image Processing based Robotic Arm Control Standalone System using Raspberry pi", *SRG - IJECE Journal*, vol. 2, no. 8, 2015.

[15] A. Kaluskar, K. Avalaskar and H. Gandhi, "Web-Controlled Surveillance Robot with A Robotic Arm", *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. 9, pp. 8614-8629, 2017.

[16] W. Kadir, R. Samin and B. Ibrahim, "Internet Controlled Robotic Arm", *Procedia Engineering*, vol. 41, pp. 1065-1071, 2012.

[17] A. Navaraja, N. Jain, D. Sengupta and C. Kumar, "Web Based Simulation and Remote Triggered Laboratory for Robots", in *CAD/CAM, Robotics and Factories of the Future*, 2016, pp. 665–677.

[18] K. Premkumar and K. Gerard Joe Nigel, "Smart phone based robotic arm control using raspberry pi, android and Wi-Fi", 2015 *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2015.

[19] A. Balestrino, A. Caiti and E. Crisostomi, "From Remote Experiments to Web-Based Learning Objects: An Advanced Telelaboratory for Robotics and Control Systems", *IEEE Transactions on Industrial Electronics*, vol. 56, no. 12, pp. 4817-4825, 2009.

[20] L. Gomes and S. Bogosyan, "Current Trends in Remote Laboratories", *IEEE Transactions on Industrial Electronics*, vol. 56, no. 12, pp. 4744-4756, 2009.

[21] [Online]. Available: http://www.arduino.org/products/tinkerkit/arduino-tinkerkit-braccio. [Accessed: 04- Jun- 2017].

[22] [Online]. Available: https://docs.blender.org/manual/. [Accessed: 04- Jun- 2017].

[23] [Online]. Available: https://www.blend4web.com/doc/en/index.html . [Accessed: 04- Jun- 2017].

ANNEXURES

RASPBERRY PI TECHNICAL SPECIFICATIONS :

- Processor:
  - Broadcom BCM2387 chipset.
  - 1.2GHz Quad-Core ARM Cortex-A53 (64Bit)
- 802.11 b/g/n : Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE):
  - IEEE 802.11 b/ g/ n Wi-Fi. Protocol: WEP, WPA WPA2, algorithms AES-CCMP (maximum key length of 256 bits), the maximum range of 100m.
  - IEEE 802.15 Bluetooth, symmetric encryption algorithm Advanced Encryption Standard(AES) with 128-bit key, the maximum range of 50 meters.
- GPU:
  - Dual Core Video Core IV Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated Open VG, and 1080p30 H.264 high-profile decode.
  - Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
- Memory: 1GB LPDDR2
- Operating System: Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
- Dimensions: 85 x 56 x 17mm
- Power: Micro USB socket 5V1, 2.5A
- Ethernet: 10/100 BaseT Ethernet socket
- Video Output:
  - HDMI (rev 1.3 1.4)

  - Composite RCA (PAL and NTSC)

- Audio Output:
  - Audio Output 3.5mm jack
  - HDMI
  - USB 4 x USB 2.0 Connector

- GPIO Connector:
  - 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip
  - Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines

- Camera Connector: 15-pin MIPI Camera Serial Interface (CSI-2)

- Display Connector: Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane

- Memory Card Slot: Push/pull Micro SDIO

BRACCIO ROBOTIC ARM:

- Maximum operating distance: 80 cm
- Maximum Height: 52 cm
- Base Width: 14 cm
- Gripper Width: 9 cm
- Cable Length: 40 cm
- Maximum Load Capacity/Weight at 32 cm operating distance: 150 g
- Maximum weight at the base Braccio configuration: 400g
- Servo Motors: 2 x SR 311, 4 x SR 431
- Total Weight: 0.792 Kg

SERVO MOTOR SPRINGRC SR431 - DUAL OUTPUT SERVO:

- Control Signal: PWM Analog
- Torque:

- At 4.8V: 169.5 oz-in (12.2 kg-cm)
- At 6.0V: 201.4 oz-in (14.5 kg-cm)

- Weight: 2.19 oz (62.0 g)
- Dimensions: 1.650.811.56 in (42.020.539.5 mm)
- Speed:
  - At 4.8V: 0.20 sec/60
  - At 6.0V: 0.18 sec/60
- Rotation Support: Dual Bearings
- Gear Material: Metal
- Rotation Range: 180
- Connector Type: J

SERVO MOTOR SPRINGRC SR311:

- Control Signal: PWM Analog
- Torque:
  - At 4.8V: 43.13 oz-in (3.1 kg-cm)
  - At 6.0V: 52.86 oz-in (3.8 kg-cm)
- Weight: 0.95 oz (27.0 g)
- Dimensions: 1.230.651.13 in (31.316.528.6 mm)
- Speed:
  - At 4.8V: 0.14 sec/60
  - At 6.0V: 0.12 sec/60
- Rotation Support: Dual Bearings
- Gear Material: Metal
- Rotation Range: 180
- Connector Type: J

THE BRACCIO SHIELD:

- Operating Voltage: 5V
- Power Consumption: 20mW
- Max current:
  - 1.1A from M1 to M4 connectors
  - 750mA from M5 and M6 connectors

HTML CODE:

```
<html>
<body>
<img src="arm.png" alt="Arm" style="width:540px;height:428px;">
<form method="post">
  <table
 style="width: 75%; text-align: left; margin-left: auto; margin
 -right: auto;"  border="0" cellpadding="2" cellspacing="2">
<tbody>
   <tr>
     <td style="text-align: center;">Rotate Left</td>
     <td style="text-align: center;">Rotate Right</td>
  </tr>
  <tr>
<td style="text-align: center;"><button name="M1L">motor1
decrease</button></td>
<td style="text-align: center;"><button name="M1R">motor1
increase </button></td>
  </tr>
  <tr>
<td style="text-align: center;"><button name="M2L">motor2
decrease</button></td>
<td style="text-align: center;"><button name="M2R">motor2
increase</button></td>
  </tr>
  <tr>
<td style="text-align: center;"><button name="M3L">motor3
decrease</button></td>
<td style="text-align: center;"><button name="M3R">motor3
increase</button></td>
  </tr>
```

```
 <tr>
<td style="text-align: center;"><button name="M4L">motor4
decrease</button></td>
<td style="text-align: center;"><button name="M4R">motor4
increase</button></td>
  </tr>
 <tr>
<td style="text-align: center;"><button name="M5L">motor5
decrease</button></td>
<td style="text-align: center;"><button name="M5R">motor5
increase</button></td>
  </tr>
 <tr>
<td style="text-align: center;"><button name="M6L">motor6
decrease</button></td>
<td style="text-align: center;"><button name="M6R">motor6
increase</button></td>
  </tr>

</tbody>
  </table>

</form>

</body>
<iframe width='370' height='305'
img src=http://172.17.42.155:8085 ></iframe>

</html>
```

PHP CODE:

```
<?php

if (isset($_POST['M1R']))
{
exec('sudo python /var/www/html/program/m1r.py');
}
if (isset($_POST['M1L']))
{
exec('sudo python /var/www/html/program/m1l.py');
```

```
}
if (isset($_POST['M2R']))
{
exec('sudo python /var/www/html/program/m2r.py');
}
if (isset($_POST['M2L']))
{
exec('sudo python /var/www/html/program/m2l.py');
}
if (isset($_POST['M3R']))
{
exec('sudo python /var/www/html/program/m3r.py');
}
if (isset($_POST['M3L']))
{
exec('sudo python /var/www/html/program/m3l.py');
}
if (isset($_POST['M4R']))
{
exec('sudo python /var/www/html/program/m4r.py');
}
if (isset($_POST['M4L']))
{
exec('sudo python /var/www/html/program/m4l.py');
}
if (isset($_POST['M5R']))
{
exec('sudo python /var/www/html/program/m5r.py');
}
if (isset($_POST['M5L']))
{
exec('sudo python /var/www/html/program/m5l.py');
}
if (isset($_POST['M6R']))
{
exec('sudo python /var/www/html/program/m6r.py');
}
if (isset($_POST['M6L']))
{
exec('sudo python /var/www/html/program/m6l.py');
}

?>
```

MOTOR 1 CODE:

- Decrease angle :

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(35,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(35,50)
file = open("/home/pi/Documents/values/m1.txt","r")
m = file.read()
file.close()
n = float(m)
n = n - 0.4
if (n<2):
n = 2
s = str(n)
file = open("/home/pi/Documents/values/m1.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(0.5)
```

- Increase angle:

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(35,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(35,50)
file = open("/home/pi/Documents/values/m1.txt","r")
m = file.read()
file.close()
n = float(m)
n = n + 0.6
if(n>12):
n = 12
s = str(n)
file = open("/home/pi/Documents/values/m1.txt","w")
```

```
file.write(s)
file.close()
z.start(n)
time.sleep(0.5)
```

MOTOR 2 CODE:

- Decrease angle :

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(36,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(36,50)
file = open("/home/pi/Documents/values/m2.txt","r")
m = file.read()
file.close()
n = float(m)
n = n - 0.4
if (n<2):
n = 2
s = str(n)
file = open("/home/pi/Documents/values/m2.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(0.5)
```

- Increase angle :

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(36,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(36,50)
file = open("/home/pi/Documents/values/m2.txt","r")
m = file.read()
file.close()
n = float(m)
```

```
n = n + 0.6
s = str(n)
file = open("/home/pi/Documents/values/m2.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(0.5)
```

MOTOR 3 CODE:

- Decrease angle:

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(37,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(37,50)
file = open("/home/pi/Documents/values/m3.txt","r")
m = file.read()
file.close()
n = float(m)
n = n - 0.4
if (n<2):
n = 2
s = str(n)
file = open("/home/pi/Documents/values/m3.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(0.5)
```

- Increase angle:

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(37,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(37,50)
```

```
file = open("/home/pi/Documents/values/m3.txt","r")
m = file.read()
file.close()
n = float(m)
n = n + 0.6
if(n>12):
n = 12
s = str(n)
file = open("/home/pi/Documents/values/m3.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(0.5)
```

MOTOR 4 CODE:

- Decrease angle:

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(38,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(38,50)
file = open("/home/pi/Documents/values/m4.txt","r")
m = file.read()
file.close()
n = float(m)
n = n - 0.4
if (n<2):
n=2
s = str(n)
file = open("/home/pi/Documents/values/m4.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(1)
```

- Increase angle:

```
import time
import RPi.GPIO as GPIO
```

```
GPIO.setmode(GPIO.BOARD)
GPIO.setup(38,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(38,50)
file = open("/home/pi/Documents/values/m4.txt","r")
m = file.read()
file.close()
n = float(m)
n = n + 0.6
if (n>12):
n = 12
s = str(n)
file = open("/home/pi/Documents/values/m4.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(1)
```

MOTOR 5 CODE:

- Decrease angle:

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(16,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(16,50)
file = open("/home/pi/Documents/values/m5.txt","r")
m = file.read()
file.close()
n = float(m)
n = n - 0.4
if (n<2):
n = 2
s = str(n)
file = open("/home/pi/Documents/values/m5.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(0.5)
```

- Increase angle:

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(16,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(16,50)
file = open("/home/pi/Documents/values/m5.txt","r")
m = file.read()
file.close()
n = float(m)
n = n + 0.6
if(n>12):
n = 12
s = str(n)
file = open("/home/pi/Documents/values/m5.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(0.5)
```

MOTOR 6 CODE:

- Decrease angle:

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(40,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(40,50)
file = open("/home/pi/Documents/values/m6.txt","r")
m = file.read()
file.close()
n = float(m)
n = n - 0.4
if (n<2.8):
n = 2.8
s = str(n)
file = open("/home/pi/Documents/values/m6.txt","w")
file.write(s)
```

```
file.close()
z.start(n)
time.sleep(0.5)
```

- Increase angle:

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(40,GPIO.OUT)
GPIO.setwarnings(False)
z = GPIO.PWM(40,50)
file = open("/home/pi/Documents/values/m6.txt","r")
m = file.read()
file.close()
n = float(m)
n = n + 0.5
if (n> 5.7):
n = 5.7
s = str(n)
file = open("/home/pi/Documents/values/m6.txt","w")
file.write(s)
file.close()
z.start(n)
time.sleep(0.5)
```