QUADRUPED ROBOT - QUBO

Enrollment Nos.	08103500, 08103501, 08503856
Name of Students	Rachit Magon, Rashi Srivastava, Vidur Shailendra Bhatnagar.
Name of Supervisor	Mr. Manish Kumar Thakur

December – 2011

Submitted in partial fulfillment of Degree of

Bachelor of Technology

In

Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING &

INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY , NOIDA

TABLE OF CONTENTS

DECLARATION

We, Rachit Magon, Rashi Srivastava, Vidur Shailendra Bhatnagar, hereby declare that the project entitled "QUADRUPED ROBOT - QUBO" is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place : JIIT, NOIDA

Date : 09th Dec, 2011.

Signatures 1.

2.

3.

Names Rachit Magon

Rashi Srivastava

Vidur Shailendra Bhatnagar

Enroll. Nos. 08103500

08103501

08503856

CERTIFICATE

This is to certify that the project entitled **“QUADRUPED ROBOT - QUBO”** submitted by **Rachit Magon, Rashi Srivastava** and **Vidur Shailendra Bhatnagar,** is an authentic work carried out by them for partial fulfilment of award of degree of B.Tech (Computer Science & Engineering) of **JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA** during their 7th semester period from 15th July, 2011 to 9th Dec, 2011, under my guidance. The matter embodied in this project has not been submitted earlier for award of any degree or diploma in any other University or Institute.

Signature of Supervisor ……………………..

Name of Supervisor: **Mr. Manish Kumar Thakur**

Designation: Senior Lecturer, Dept. of CSE/IT

Date ……………………..

ACKNOWLEDGEMENT

We would like to express our gratitude to everyone involved with this project and making it a success thus far. We would like to thank the Department of Computer Science and Information Technology of JIIT, Noida for giving us the permission to commence this project in the first instance and for providing us the necessary resources specially the hardware components.

A sincere thanks to our project supervisor **Mr. Manish Kumar Thakur,** Senior Lecturer, Dept. of CSE/IT, for providing us with the opportunity to work under him and helping us learn, grow and benefit from his expertise and immense experience. We thank him for his patient hearing of our ideas and opening up our minds to newer horizons by pointing out our flaws, providing critical comments and suggestions to improve the quality of our work and appreciating our efforts.

DATE

PLACE

(Vidur Shailendra Bhatnagar)

(Rachit Magon)

(Rashi Srivastava)

SUMMARY

The Quadruped Robot **QUBO** is a Four Legged Voice controlled robot for entertainment and basic security purposes. We chose to build pet-type legged robot because we believe that dog-like and cat-like legged robots have major potential for future entertainment robotics markets for personal robots. However, numbers of challenges exists before any of such robot can be materialized in the real world. Robots have to be reasonably intelligent, maintain certain level of agility, and be able to engaged in some collaborative behaviors.

We divided the system into three main sub-divisions, namely Hardware & Controller, Computer Vision and Speech Interaction[9]. We would be using an 8-bit *Atmega 2560* micro-controller[2], to give two degrees of freedom to each leg, one degree of freedom to the tail and two to the head. The *Atmega 2560* would be controlled by an Intel Core i3 through Serial Data Transfer through *CC2500 wireless* modules, which would be receiving data from a CCTV camera and a microphones mounted on the "head" of the Quadruped Dog. The Quadruped would have two gaits: the slow *walking* gait and a speed *trotting* gait [3]. Both are symmetric gait patterns of quadruped animals the former one being a less complex and the trot gait requires movement of oscillator or center-of-mass balancing techniques [4].Along with basic walking techniques, we aim to develop Integrated Hand Mechanism for the robot, similar to the several mechanism proposed by [6] - [8].

We will be using the feature of speech and voice detection to communicate with the Quadruped Robot. We would be using the Hidden Markov Model [12] technique for speech recognition because they are a powerful statistical method for modeling speech signals, and they are the dominating approach in speech recognition today.

We will be using a marker based detection system for the reasons that it adds precision to our hardware design. We will use a set of markers to give instructions to qubo over the wireless module, so as to enable it to perform certain functions.

Signature of Students	**Signature of Supervisor**
____________________(Rachit Magon)	____________________
____________________(Rashi Srivastava)	(Mr. Manish Kumar Thakur)
____________________(Vidur Shailendra Bhatnagar)	

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ACRONYMS

QUBO	Quadruped Bot
UART	Universal asynchronous receiver/transmitter
TTL	Transistor–transistor logic
CapSense	Capacitor Sensor
RF Module	CC2500 Wireless Module
VRBOT	Voice Recognition for Robots
IDE	Integrated Development Environment
Mega2560	Atmel Mega 2560 Microcontroller

Chapter 1.
INTRODUCTION

1.1 General Introduction

Robot Entertainment is a new field for using autonomous robots in the entertainment industry. In order to demonstrate the feasibility of Robot Entertainment, we have developed a pet- type legged robot called QUBO (Fig.1), which has a head, neck, tail and 4 legs each of which has two degrees of freedom and is equipped with on-board sensors such as a micro-camera, a stereo microphone, and touch sensor. QUBO interacts with people by tonal sounds, and exhibits a large variety of complex behavioral patterns. QUBO is developed to investigate the feasibility of using robots as an entertainment tool. We consider entertainment applications an important target at this stage of both scientific and industrial development.

1.2 Problem Statement

To develop a Quadruped Robot System for Entertainment & Security which can follow commands and play with the owner, recognize a friend and bark at a stranger.

1.3 Empirical Study

1.3.1 Field Survey

Cost Analysis

The charts & diagrams below show the cost and the cost distribution between the components which have been used to develop Qubo.

COMPONENT	QTY.	COST	DETAILS	FINAL COST
Arduino Mega2560	0	-	Since the department has other Arduino boards we decided to use the ones available.	-
ArduinoDuimenilove	2	INR 1600	Issued to us for one semester from the CSE Department	-
Full Rotation Servo Motors	2	INR 1000	Issued to us for one semester from the CSE Department	-
Half Rotation Servo Motors	12	INR 6000	Issued to us for one semester from the CSE Department	-
VRBot Module	2	INR 1200	Ordered from Official Arduino Dealer	INR 2200
CCTV Camera	1	INR 1500	To be bought from a whole-seller in ChandiniChowk, New Delhi	INR 1500
TV Tuner Card	1	INR 500	Already Owned	-
Ultra Sonic Distance Sensors	1	INR 1500	Issued to us for one semester from the CSE Department	
Chasis Components	-	-	From Scrap Materials	-
	TOTAL	INR 11800		INR 3700

Table 1: Cost of Components

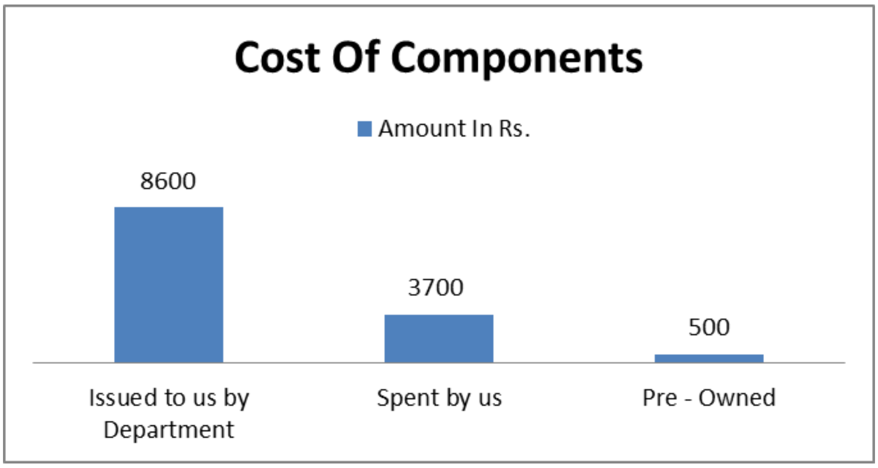

Fig 1: Cost of Components

1.3.2 Existing Tools

Embedded Programming

1. BASCOM: BASCOM accepts commands in BASIC and is very widely used. The disadvantage of using BASCOM is that it does not support any advanced data structures like double dimensional arrays or linked lists. Moreover, the programming is low-level and though it has more control it might get very difficult to program to control 14 actuators together.
2. AVR-Studio: Software made by the microcontroller giant Atmel which has been strongly recommended by Atmel corporation and is used for programming 8-bit microcontrollers in C. But we would need more hardware (either a Parallel Port or a USB Burner) to load the code into the microcontroller chip.
3. Processing based Arduino: Based on the groundbreaking software by MIT, uses the boot loader which comes pre-loaded on Arduino boards hence does not need a burner to burn the codes. It only requires a USB A-to-B cable and loads the desired codes into the microcontroller by Serial Communication.
4. Avr-Dude: It is used to burn HEX files onto the microcontroller, but since we are not generating HEX files, we do not need this software tool.
5. AT-Prog: Atmel-Avt-Microcontroller Programmer, it's primary purpose is, like AVR-DUDE, to burn HEX files onto the microcontroller chip, but again, it also needs a parallel port programmer.

   Hence, we would be using Processing based – *Arduino* software with no extra hardware, except a standard USB cable to load codes onto the IC, which would be recognized by its pre-installed boot loader.

6. *Computer Vision:* OpenCV libraries.
7. *Speech Interaction*
   - C# Speech Library: This Library saves a few pre-defined speech commands and when the user speaks it searches it's database for an appropriate match. The library is very basic and not robust, hence we would not use this library.
   - CMU – SPHINX: developed by the Carnegie Mellon University, this library has a training Module and a Test Module along with the real time running process. It needs intensive training (around 20 hours) and can recognize a dialect if proper training is given.

Analysis of Quadruped Robots Available in the market:

- Sony Aibo: The most efficient quadruped machine available in the market and certainly the most expensive. A second hand Sony Aibo would cost around 4,50,000 INR and comes loaded with technology with a 384 MHz 64-bit RISC processor developed by Sony Corporation specifically for the Aibo.
- Silverlit I-Cybie: Has been called the "economy Aibo", costs around 75,000 INR and is quite efficient but not comparable to the Aibo, it is approximately 12" tall and has been made up of 16 motors.
- Pleo the Dinosaur: Powered by Invo Labs "Life OS" this is a quadruped which evolves as the user plays with it. Pleo has four emotions which change according to the interaction level with humans. It is powered by a 32-bit ARM7 processor along with four 8-bit Atmel microcontrollers.

Below is a comparison of the three most popular quadruped robots available in the market, as of today, taken from the manufacturers' websites itself.

Comparison Factor	Aibo	I-Cybie	Pleo
Cost	4,50,000	75,000	70,000
Processor	64-bit Risc	32-bit	32-bit ARM 7
Motors	20	16	12
Camera	Yes	Yes	Yes
Speech Recognition	90% Accuracy	70% Accuracy	No Speech Interface
Touch Sensors	10	5	8
Display	150 LEDs display screen	31 LEDs/eye	5 multi-color LED/eye

Table 2: Comparative Study of Quadrupeds

1.3.3 Experimental Study

As part of experimental study, before starting with the project, a prototype of one leg was designed to see whether the Servo Motors would be able to handle the weight of the body or not, at different voltages.

The results of the prototype were as follows:

Motor Position	Given Potential Difference	Result
Thigh	4.5V	Very Slight Motion, current level exceeded resulting in the Arduino board to restart.
Knee	4.5V	Very Slight Motion, current level exceeded resulting in the Arduino board to restart.
Thigh	5V	Slight Motion, but not being able to give enough torque to pick up the Knee motor.
Knee	5V	Successful motion, but a buzzing sound was being heard from within the motor.
Thigh	6V	Successful Rotation
Knee	6V	Successful Rotation
Thigh	8V	Successful Rotation
Knee	8V	Successful Rotation

Table 3: Prototype Experimental Results

1.4 Approach To Problem In Terms Of Platform Used

Our entire project is based on Arduino open source development board. Using an Arduino simplifies the amount of hardware and software development needed to get a system running. The Arduino hardware platform already has the power and reset circuitry setup as well as circuitry to program and communicate with the microcontroller over USB. In addition, the I/O pins of the microcontroller are typically already fed out to sockets/headers for easy access (This may vary a bit with the specific model).

On the software side, Arduino provides a number of libraries to make programming the microcontroller much easier. The simplest of these are functions to control and read the I/O pins rather than having to fiddle with the bus/bit masks normally used to interface with the Atmega I/O (This is a fairly minor inconvenience). More useful are things such as being able to set I/O pins to PWM at a certain duty cycle using a single command or doing Serial communication.

The greatest advantage is having the hardware platform set up already, especially the fact that it allows programming and serial communication over USB. This saves the trouble of having to do our own PCB (which can cost more than an Arduino) or breadboarding. Either way it takes a lot of time to do and verify that everything is working correctly.

1.5 Significance of Problem

While primitive robot toy models only execute standardized pre-programmed routines, sometimes little more than a wind-up toy could do, advancing technology allows for interaction with the user and/or other environmental stimuli (e.g. sensor-detected obstacles), thus somewhat resembling a live playmate, but which has no feelings and will thus always remain inferior to a pet, while more convenient as it may be (ab)used with impunity and has low maintenance.

Nevertheless in the mind of some users the things can hold the loved place of a pet, as demonstrated by the fact that some even sleep with a metallic one instead of a plush cuddly toy. In fact manufacturers even found it pays to produce a toy that is essentially designed to be nurtured, rather like an egg in some 'parenting experience simulations', as proven by the success of the Japanese Tamagotchi.

Entertainment robots can take the form of interactive communications marketing tools at trade shows wherein promotional robots move about a trade show floor providing tongue and cheek interaction with trade show attendees in order to bring said attendees to a particular companies trade show booth.

Chapter 2.

LITERATURE SURVEY

2.1 Summary of Relevant Papers

Title	Development of an Autonomous Quadruped Robot for Robot Entertainment
Authors	Masahiro Fujita & Hiroaki Kitano
Year of Publication	1998
Publishing Details	Published In Autonomous Robots, Volume 1 by Kluwer Academic Publishers
Summary	The authors present Robot Entertainment as a new field of the entertainment industry using autonomous robots. The authors have developed an autonomous quadruped robot, named MUTANT, as a pet-type robot. It has four legs, each of which has three degree-of-freedom, and a head which also has three degree-of-freedom. Micro camera, stereo microphone, touch sensors, and other sensor systems are coupled with a behavior generation system, which has emotion module as its major components, and generates high complex and interactive behaviors.
Web Link	http://www.springerlink.com/content/m766217640t629wk/

Title	Gait Analysis and Implementation of a Simple Quadruped Robot
Authors	Lianqing Yu Yujin Wang &Weijun Tao
Year of Publication	2010
Publishing Details	Published In Proceedings of 2010 2nd International Conference on Industrial Mechatronics and Automation
Summary	In this paper, the authors present an analysis of the various symmetric and asymmetric gait patterns of quadrupeds and implement the same in a Quadruped Robot to test the validity of the analysis. The authors perform a walking experiment and test it in accord with the mathematical model generated by them, hence proving the gait analysis as suitable for quadruped robots.
Web Link	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5538277

Title	Sony Legged Robot for RoboCup Challenge
Authors	H Kitano, M Fujita, S. Zrehen& K. Kageyama,
Year of Publication	1998
Publishing Details	Published In Proceedings of IEEE International Conference on Robotics and Automation, 1998.
Summary	The authors, working with Sony Corporation, present the schematics of Sony's legged robot system which participated in RoboCup 1998 – Paris.
Web Link	http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=680735

Title	Road Markers Recognition Based on Shape Information
Authors	Yunchong Li, Kezhong He, and Peifa Jia
Year of Publication	2009
Publishing Details	Proceedings of the IEEE, Computer Science and Information Engineering, 2009 WRI World Congress
Summary	This paper describes a shape-based road marker detection and recognition method in this paper. The 8-neighbor chain codes of close regions are abstracted from the plane image. Selected moment features and an improved minimal-error-rate classifier are utilized to recognized different lane markers and other road markers. The lengths and slopes of lane markers are fast calculated also using the moment features. The results show that the method can detect and recognize road markers effectively.
Web Link	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4290101

Title	Research of quickly identifying markers on Augmented Reality
Authors	Paul Viola and Michael J. Jones
Year of Publication	2003
Publishing Details	Published in the proceedings of the International Journal of Computer Vision 57(2), 137–154, 2004
Summary	For complicated calculation caused by marker unwrapping in two main visual marker tracking platforms in Augmented Reality, ARToolKit and ARTag, a new kind of markers based on ARToolKit is designed and implemented, which are called diagonal connected components markers. Those markers can be recognized without unwrapping, which just rely on numbers of white connected components of two diagonals of those markers. Experiment results show that compared with ARToolKit markers, the speed of recognition of those new markers is greatly increased.
Web Link	http://www.cs.cmu.edu/~efros/courses/AP06/Papers/viola-IJCV-01.pdf

Title	A Speech Interaction System based on Finite State Machine for Service Robot
Authors	Yong Tao, Hongxing Wei &Tianmiao Wing
Year of Publication	2008
Publishing Details	Published in the proceedings of the IEEE International Conference on Computer Science and Software Engineering, 2008
Summary	This paper discusses about t he human robot interaction based on finite state machines, where each step of the interaction is a state of the FSM. The control flow of the interaction is determined by the transitions in the FSM. The speech interaction is classified in 5 states: Speech Data Capture State, Speech recognition state, Logical conversion state, Speech response state and Interaction end state.
Web Link	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4721947

Title	English Digits Speech Recognition System Based on Hidden Markov Models
Authors	Ahmad A. M. Abushariah, Teddy S. Gunawan,Othman O. Khalifa
Year of Publication	2010
Publishing Details	Published in International Conference on Computer and Communication Engineering (ICCCE 2010), 11-13 May 2010, Kuala Lumpur, Malaysia
Summary	This paper aims to design and implement English digits speech recognition system using Matlab (GUI). This work was based on the Hidden Markov Model (HMM), which provides a highly reliable way for recognizing speech. The system is able to recognize the speech waveform by translating the speech waveform into a set of feature vectors using Mel Frequency Cepstral Coefficients (MFCC) technique This paper focuses on all English digits from (Zero through Nine), which is based on isolated words structure. Two modules were developed, namely the isolated words speech recognition and the continuous speech recognition.
Web Link	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5556819

Title	Vowel Recognition Based on Frequency Ranges Determined by Bandwidth Approach
Authors	Paulraj, M.P.;,Yaacob, S., Yusof, S.A.M.
Year of Publication	2008
Publishing Details	Published in the Audio, Language and Image Processing, 2008. ICALIP 2008, International Conference on Digital Object Identifier
Summary	Automatic speech recognition (ASR) has made great strides with the development of digital signal processing hardware and software especially using English as the language of choice. In this paper, a new feature extraction method is presented to identify vowels recorded from 80 Malaysian speakers. The features were obtained from vocal tract model based on bandwidth (BW) approach.
Web Link	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4590133

2.1 Integrated Summary of Literature Studied

Development on Quadruped as Robots started in 1994 wiyj Katsuhiko Inagaki & Hisato Kobayashi's paper on "Dynamical Motion Control for Quadruped Walking with Autonomous Distributed System" whereas advances and developments in speech recognition started in 1950 with Bell Labs inventing the first speech recognizer in the early 1950's. Using video for Face Recognition, which is also a very important part in home surviellance systems, received attention from engineer's in the 1960's.

In 1960,Bledsoe, along other researches, started Panoramic Research, Inc., in Palo Alto, California. The majority of the work done by this company involved AI-related contracts from the U.S. Department of Defense and various intelligence agencies. During 1964 and 1965, Bledsoe, along with Helen Chan and Charles Bisson, worked on using computers to recognize human faces.Because the funding of these researches was provided by an unnamed intelligence agency, little of the work was published.

By the 1970's research on Automatic Speech recognition as well as Face Detection & Recognition was in full swing since the development of the Hidden Markov Model & Haar coefficients.

During the early 1990's the ground breaking paper from Sony Corporation's Scientists H Kitano & M Fujitaon: "Sony Legged Robot for RoboCup Challenge" since then advancement in Quadruped Robot and the use of video and audio for control has been commonplace. The use of limb's as manipulation tools, however, was not realized until late 1990's when T. Arai Achi, H. Adachi, and K. Hormna published their paper on “Integrated Arm and Leg Mechanism".

Release of Quadruped Robots like the Aibo and the I-Cybie proves that this technology can be used for many purposes, the only drawback being that right now the technology is very expensive to be bought by the common man.

Qubo is using the Exoskeleton Version 3.0. Version 1.0 was built with hard- board which was fragile and soft but easy to shape. Version 2.0 was an Aluminium Variant but the size was bigger with all the interface cards and modules on the top. Version 3.0 has a more compact design with better weight distribution.

Each leg of Qubo has two degrees of freedom, while the neck, head and tail have one degree each. A single motor acts as a flexor-extensor pair muscle, flexing as well as extending a joint. The limbs are divided into Left & Right Limbs which are further divided into Hip, Knee, Shoulder & Elbow.

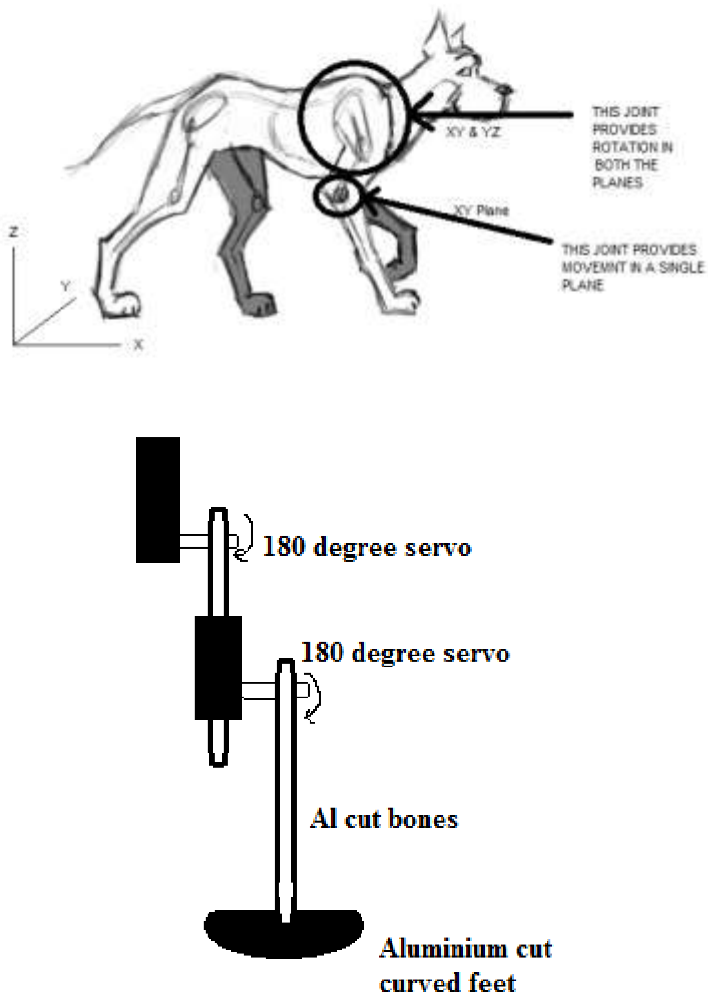

Fig 2: Qubo's Leg Structure

Marker Detection Algorithm

The first step is to do a 2D cross correlation on the image using a pre-defined mask. The output of this step is a series of threshold points, which may be part of markers. The next step is to group these points and to calculate the centroids of the markers. This gives a series of coordinate pares of possible markers.

The 2D cross correlation consists of placing a pre-defined mask on a certain image area. Each point of the mask is then multiplied with its corresponding point on the image. All these products are added to obtain the correlation value.

Before the different ways of doing the cross correlation are discussed, it is necessary to look at the design of the mask.

Below is an example of a marker image and the mask used to detect the marker:

0	0	2	2	0	0
0	2	2	2	2	0
2	2	2	2	2	2
2	2	2	2	2	2
0	2	2	2	2	0
0	0	2	2	0	0

Image of marker

-8	-8	4	4	-8	-8
-8	4	4	4	4	-8
4	4	4	4	4	4
4	4	4	4	4	4
-8	4	4	4	4	-8
-8	-8	4	4	-8	-8

Mask used to detect marker

0	0	8	8	0	0
0	8	8	8	8	0
8	8	8	8	8	8
8	8	8	8	8	8
0	8	8	8	8	0
0	0	8	8	0	0

Product of mask and marker image

Fig 3: Marker Detection

Assuming that the size of the area in which the markers are looked for, the search area, will be L x L pixels. Note that the search area in a practical setup may not be square. The size of the mask is M x M pixels. The mask must be smaller than the search area, therefore L>M. The image below shows the relationship between the image, search area and the mask.

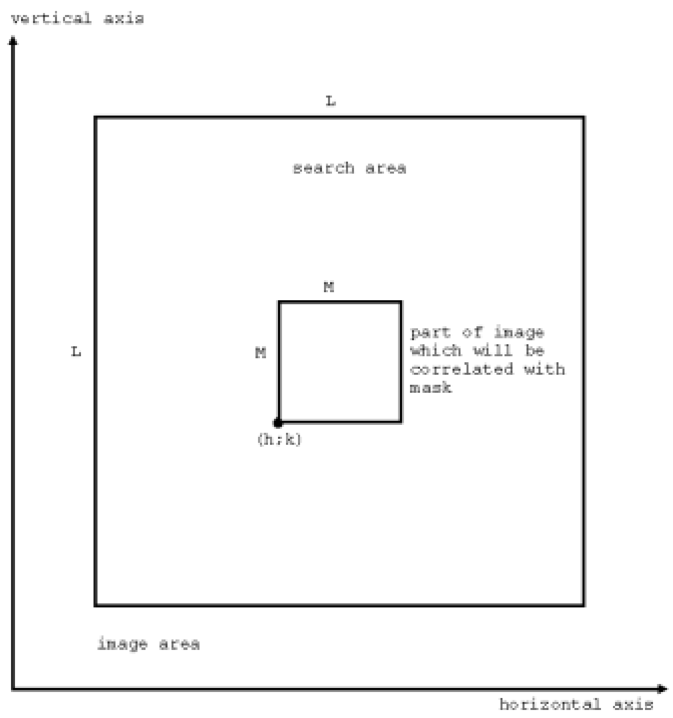

Fig 4: Marker Coordinates Evaluation

Chapter 3.

ANALYSIS DESIGN & MODELLING

3.1 Overall Description of the Project

In the recent years, the importance of Legged Robots has been realized greatly in terms of natural looks and better gait. Though wheeled Robot is faster legged robots can walk on different terrains and give a better feel to interact with. Qubo is a four-legged Robot which has been based on the biological exoskeleton of a dog.

Other than providing basic features playing features, which are present in many Quadruped available in the market, our project “QUBO” would also provide security for the owner. Using advanced database with computer vision, Qubo would detect markers (commands in form of digital markers) with the help of a wireless AV camera and respond to the markers by performing different operations. The major difference between the highly advanced Quadrupeds available in the market is the price required to develop Qubo, which is less than 10% of the Commercial Quadrupeds.

3.2 Specific Requirements

3.2.1 External Interface

Qubo has been developed on the concept of a Robotic Dog hence the Interface which Qubo provides, like any other dog, is the command-over-voice and vision interface. The master can give Qubo certain commands which would then be recognized by the System and the response is usually some mechanical movement or sometimes sounds. Similarly, the master can show different markers to Qubo each of them depicting some operation to be done by the Qubo.

- The master can call the Dog to which response should be visible
- After any command given to the Robot, it should start the coordinated movements while be ready for the trigger to get other commands.
- The master can easily stop an ongoing operation with the trigger (eg Double Clap) which can be defined by the master (like a whistle or a clap), or showing him a “Stop” marker.

3.2.2 Functional Requirements

- Response to Trigger: While performing (or not performing) any operation, Qubo should respond to the Trigger which has been defined by the master. The response to the Trigger must be visible and Qubo should be ready to accept Commands.
- Stand: If Qubo is called for, it should get to this position and get ready to accept any command.
- Bend: This should be a low-energy consumption state (at-ease position) for Qubo where it should bend in a way to keep standing but not consume a lot of power.
- Walk/Move Forward: When the master asks Qubo to move, the Robot Dog should perform a well-coordinated action to walk forward. Moreover, Qubo should be careful of not to hit a wall (or obstacle) while walking.
- Hello: To greet the master, Qubo should sit in a rest position and lift it’s hand up while waiting for the master to touch his hand, once the masters’ hand touches Qubo’s hand it should shake the hand till the master releases it.
- Look: This should be a low-energy but active position where Qubo should look the whole room and hunt for any movement or a stranger to warn the master of.
- Dance: Qubo should be able to dance while not moving much from its position so as to entertain friends and family.
- Detect markers: Qubo should recognize markers and perform operations accordingly.

3.2.3 Performance Requirements

The main Performance Requirements for the dog are:

- Qubo should be able to easily operate in indoor conditions under any lighting and any indoor flooring with respectable balancing while walking.
- Qubo should handle any discrepancies in recognition of markers and voice commands and not act in any unintended manner. It should consume very less power on modes on which it has to be kept for long hours together.
- Qubo should be a fun to play while providing security features to the master.

3.2.4 Logical Database Requirements

- The system requires database for the commands for which it has been trained to operate.
- Moreover, a constantly increasing database of stored markers will have to maintained when recognizing and storing different markers.

3.2.5 Design Constraints

As with any Legged Robot the Design Constraints, other than price, are:

- Stability: Since balancing a quadruped is very important, the design should be stable enough to handle various kinds of gait and action without falling as well as without exceeding the power limit.
- Weight: The Design should be light weight and strong. The modules should be independent and complete model should be much more than the sum of parts.

3.2.6 System Attributes

Software Requirements:

- Processing-based Arduino-0022 IDE
- VRBot Training Binary Executable
- Visual Studio 2008/2010
- Windows Operating System (Windows 7 or higher)

Hardware Requirements:

- Arduino Mega2560 with Atmel Atmega 2560
- VRBot Interface
- Standard Servo Motors & UltraSonic Sensors
- Wireless 1.25GHz AV Camera
- Standard Motor (9V) & Sensors (5V) Interface Cards
- 9V 3A Light Weight Battery Pack
- Qubo Exoskeleton Version 3.0

3.3 Design Diagrams

3.3.1 Use Case Diagrams

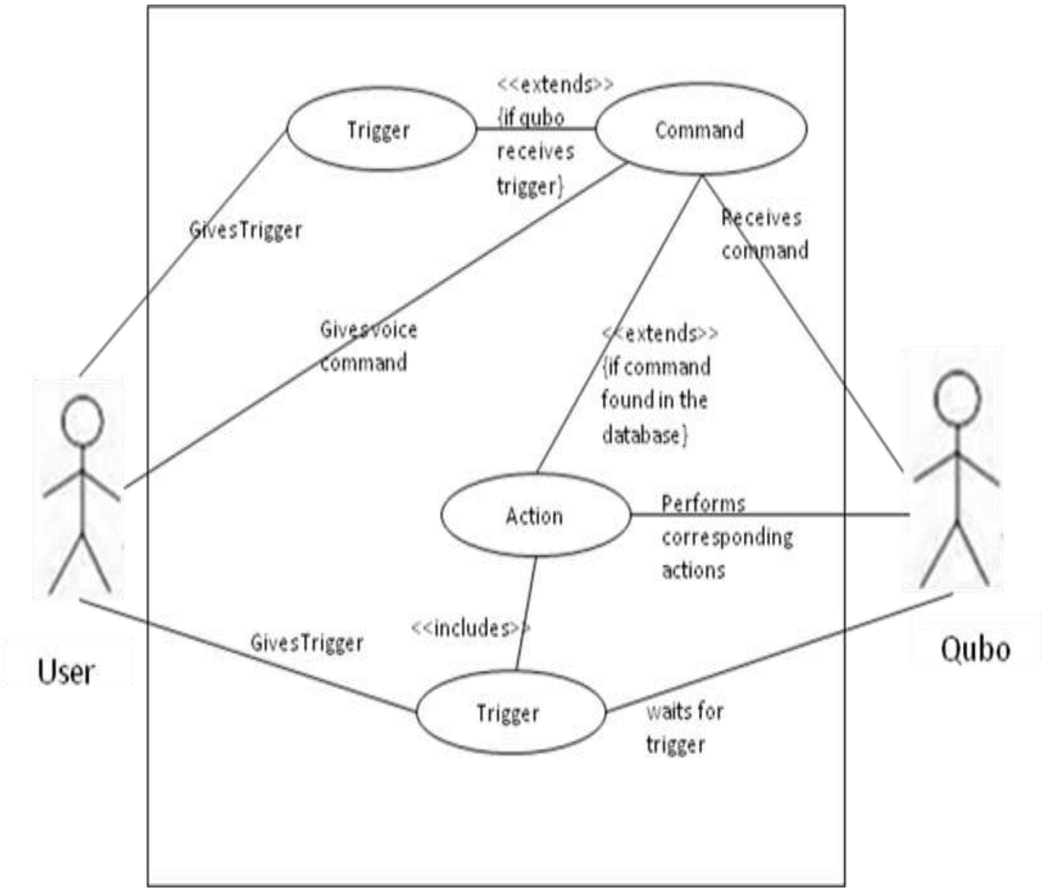

Fig 5: Use Case diagram for voice trigger and command

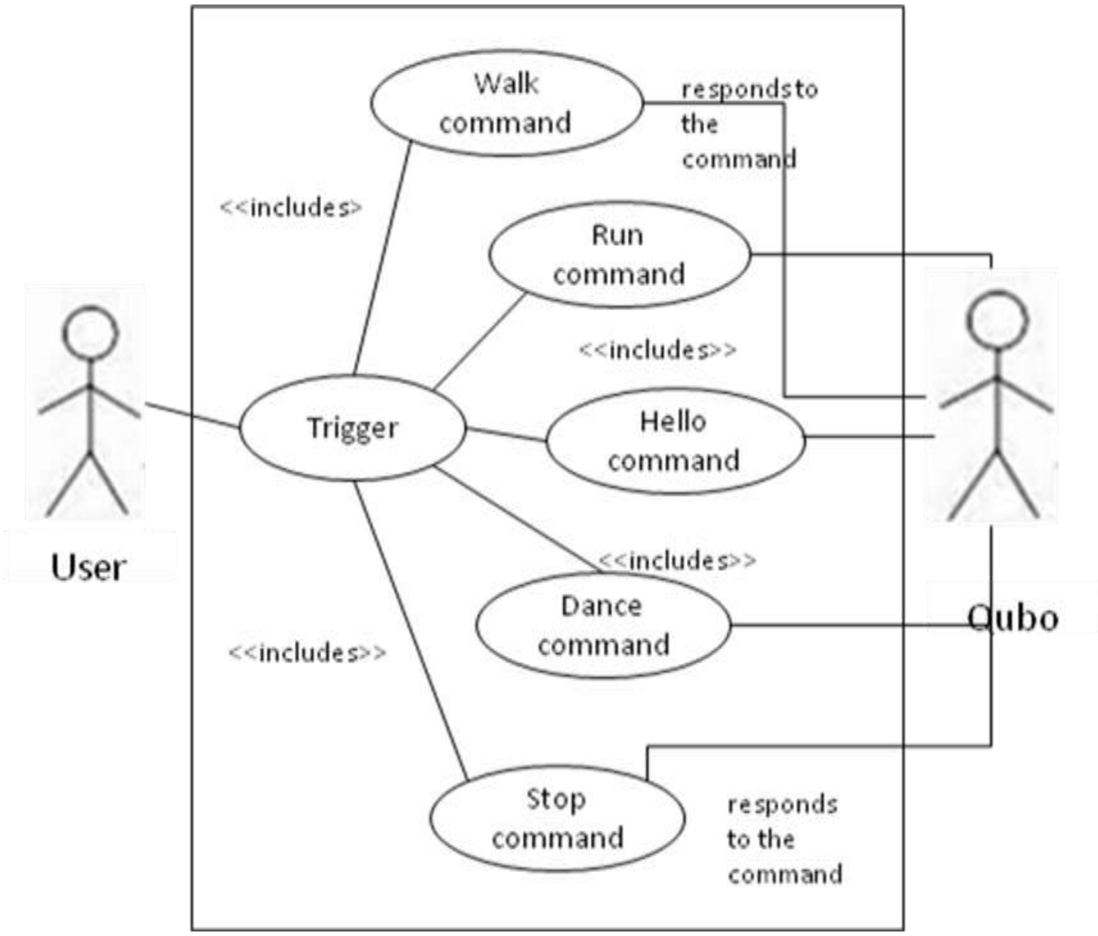

Fig 6: Use Case diagram for voice commands and corresponding actions

3.3.2 Control flow diagrams

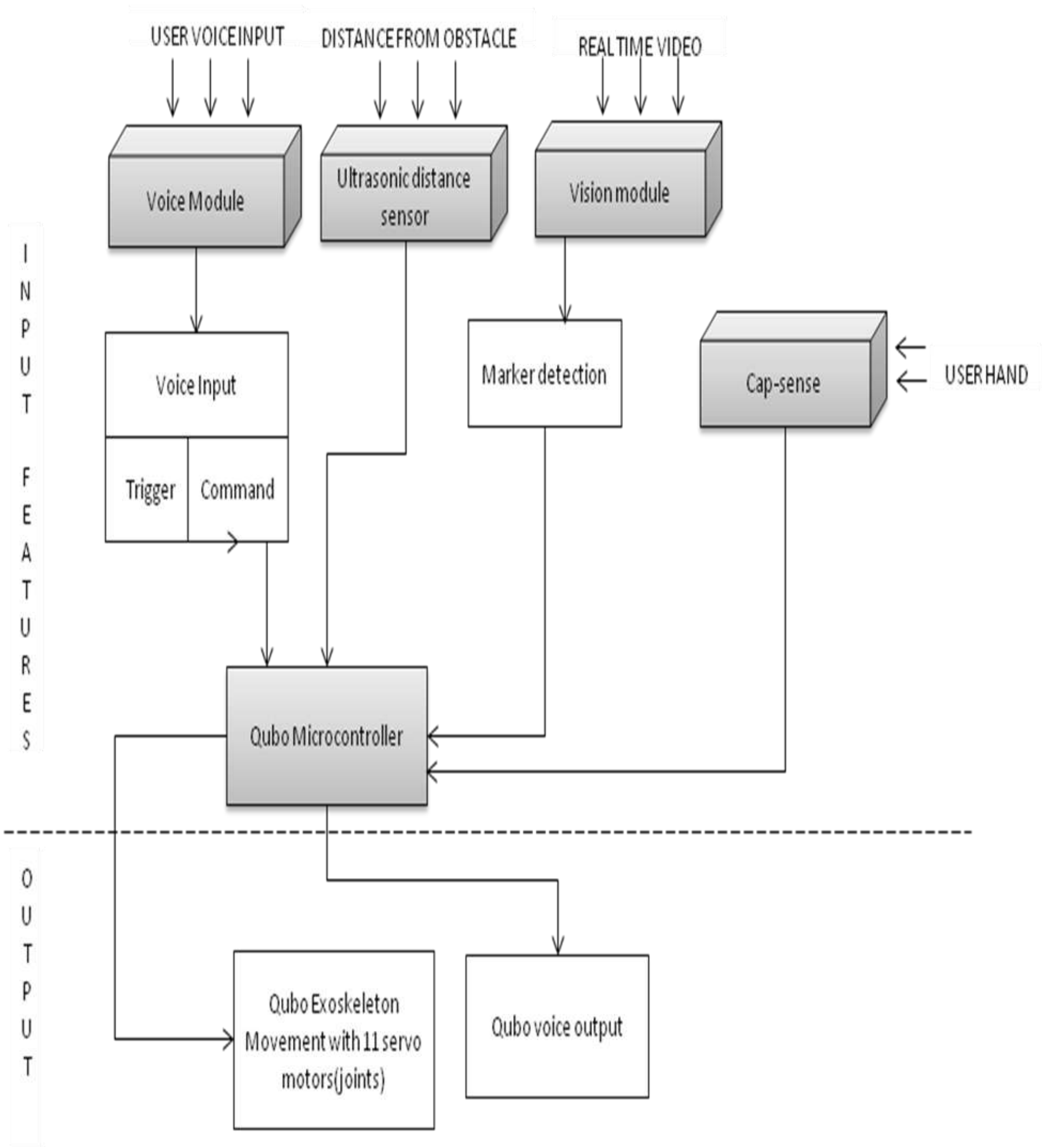

Fig 7: Basic Control Flow Diagram

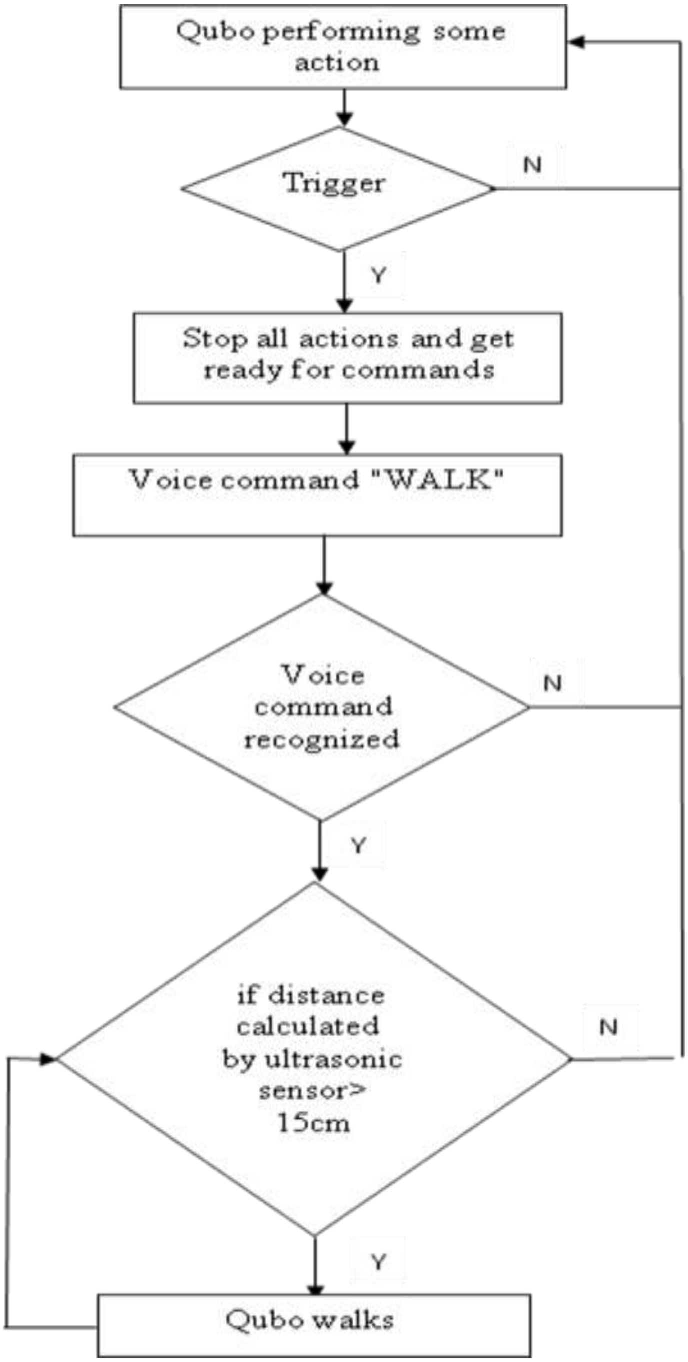

Fig 8: Control Flow Diagram for "Walk Command"

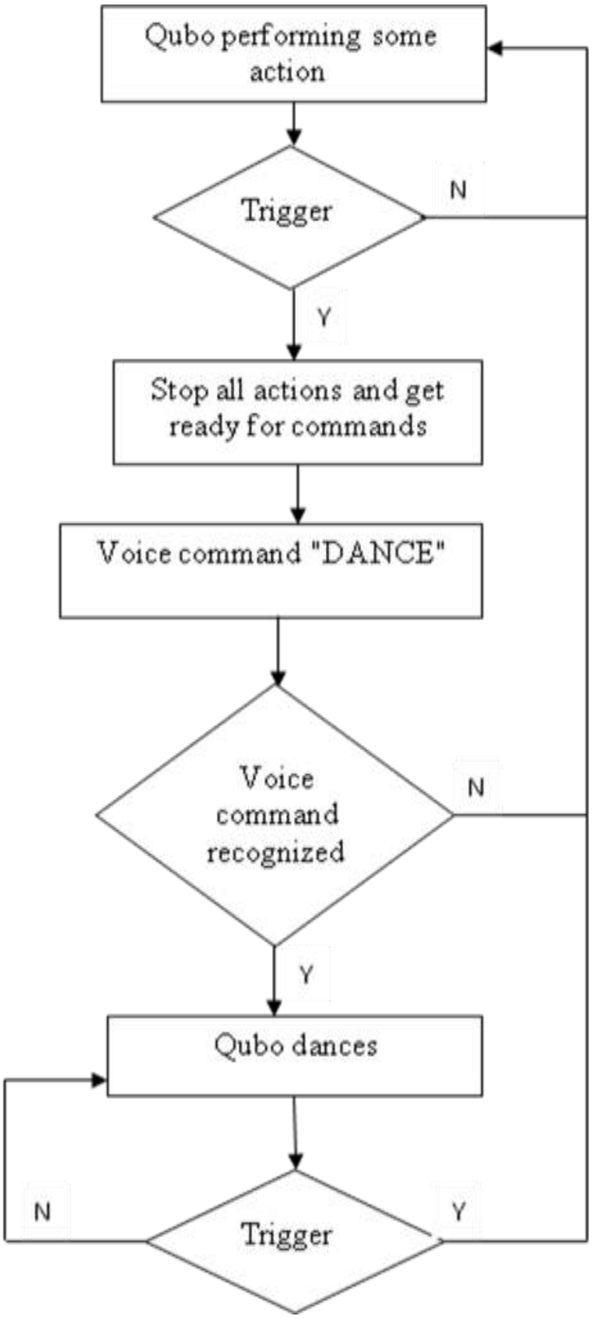

Fig 9: Control Flow Diagram for "Dance" voice command

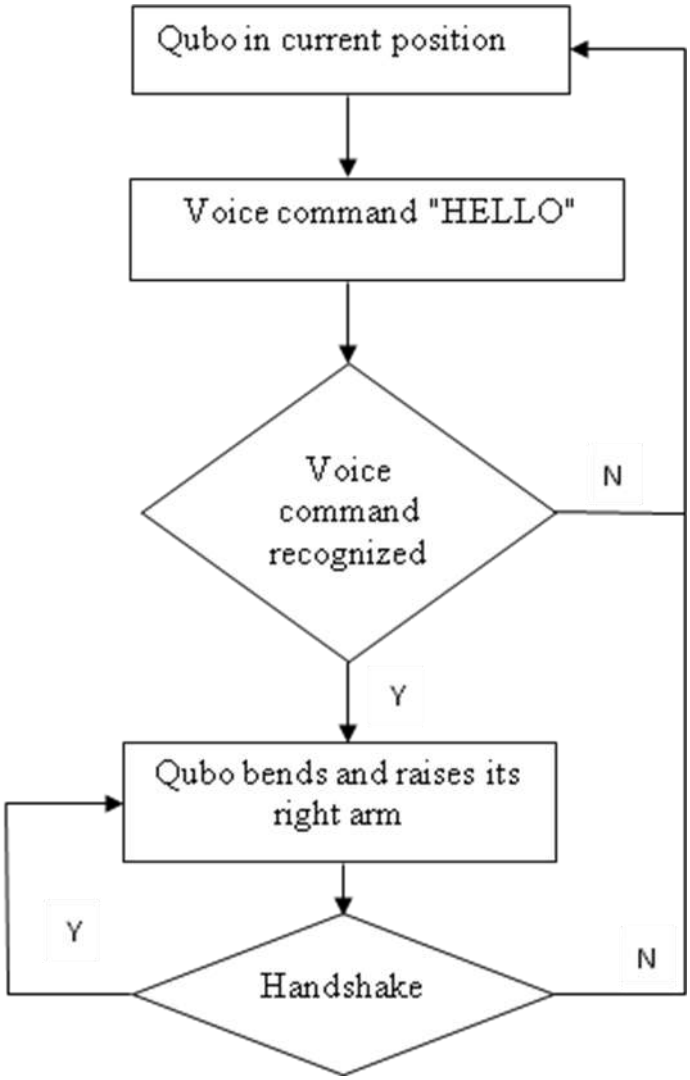

Fig 10: Control Flow Diagram for "Hello" voice command

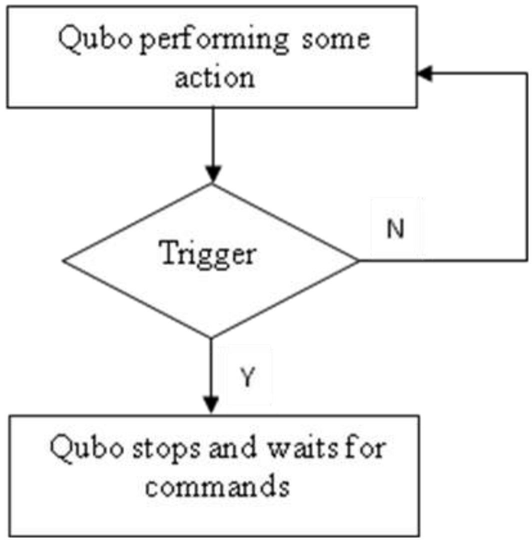

Fig 11: Control Flow Diagram to Stop Qubo

3.3.3 Sequence Diagrams

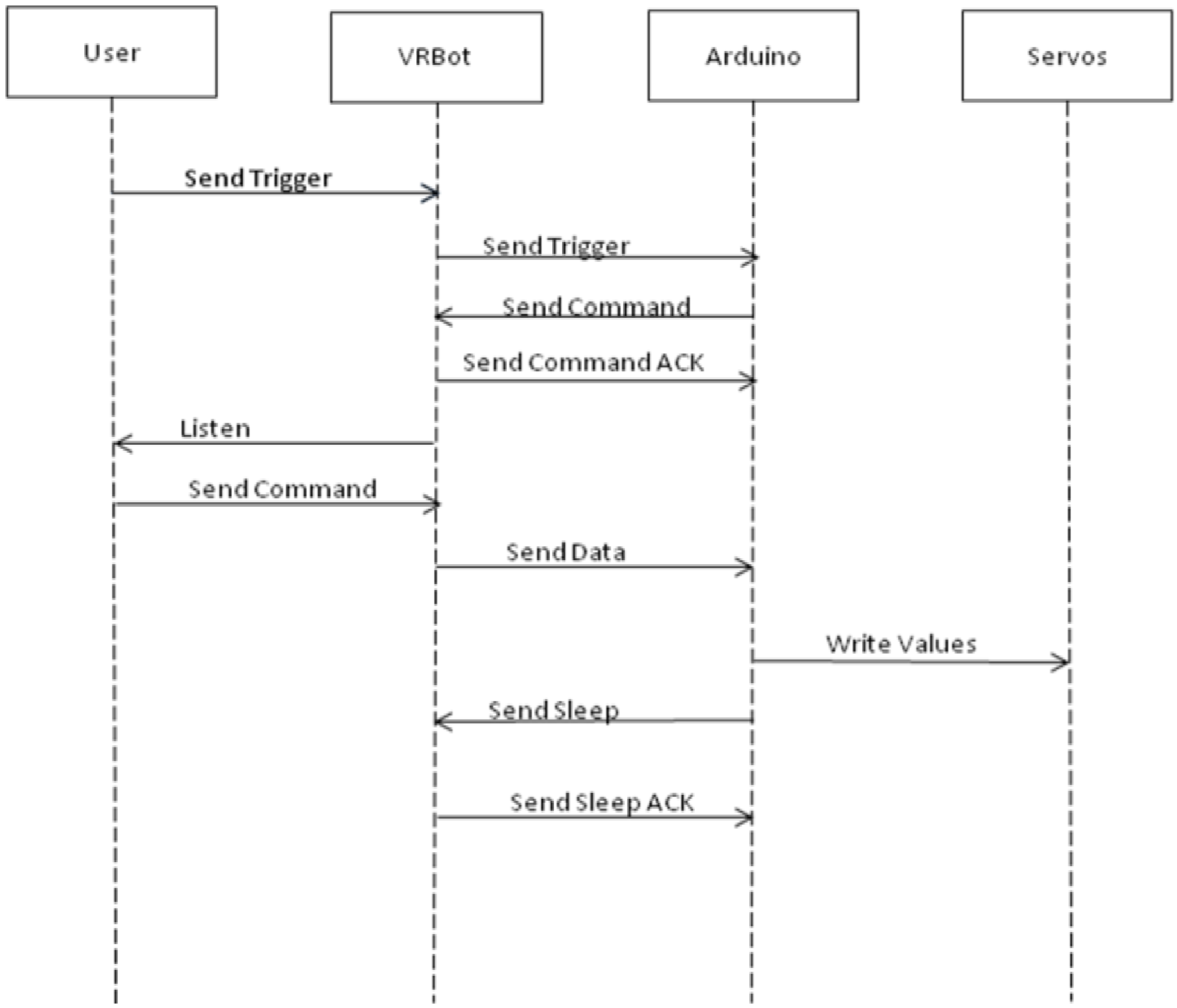

Fig 12: Sequence Diagram for Audio Command Recognition

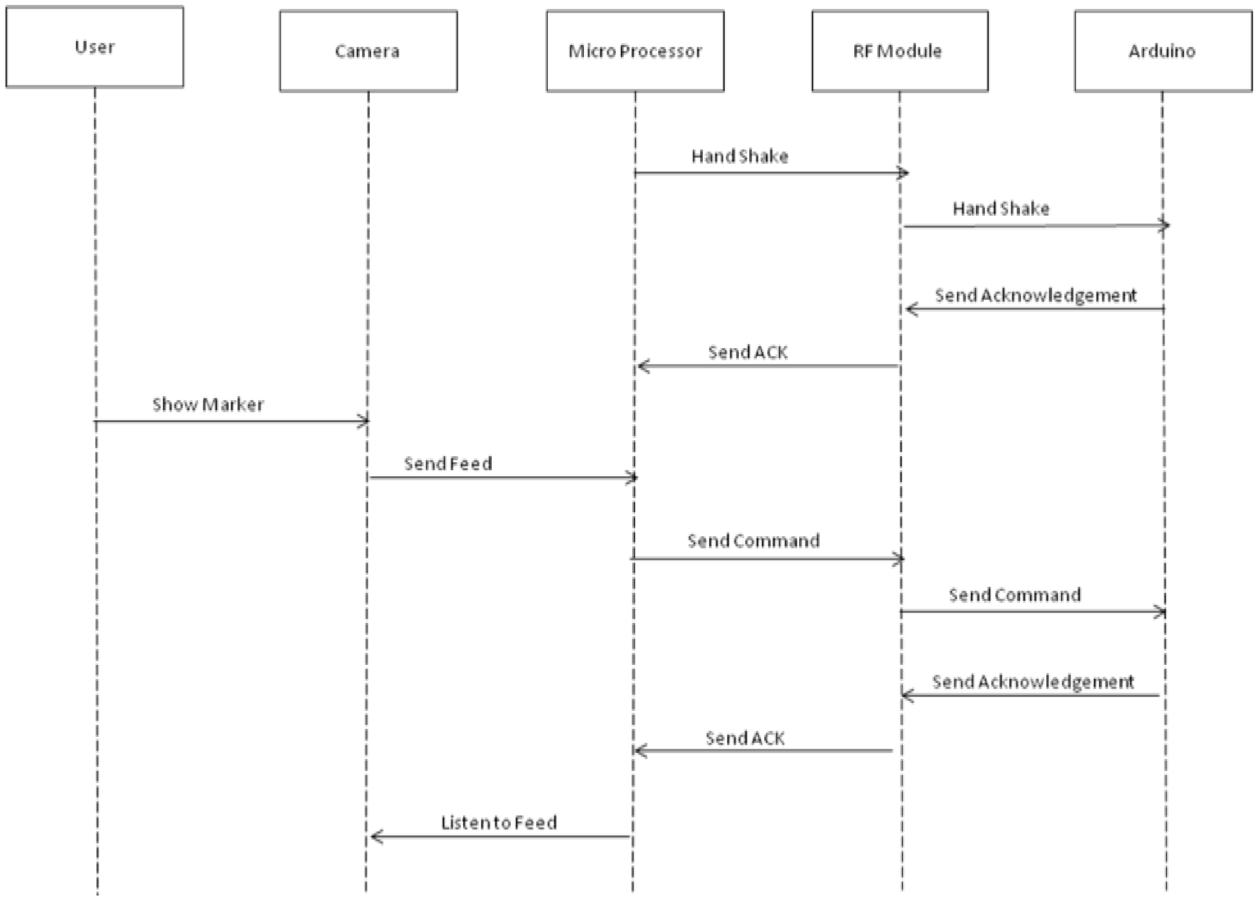

Fig 13: Sequence for Microcontroller and Camera Interface for Handshake Operation

3.4 Risk Analysis

Risk id	Classification	Description	Risk Area	Probability	Impact	RE(P*I)
A1a	Class: Product Engineering Element: Requirements Attribute: Scale	This attribute covers both technical and management challenges presented by large complex systems development.	Scaling of Project is possible as it will not require grass root changes but will drastically increase the cost.	1 (Low)	3 (Medium)	3
A2b	Class: Product Engineering Element: Design Attribute: Difficulty	The risk refers to functional or design requirements that may be extremely difficult to realize.	Balancing the quadruped without using an accelerometer and gyrometer.	3 (Medium)	5 (High)	15
A2c	Class: Product Engineering Element: Design Attribute: Interfaces	The risk involving the techniques for defining and managing all hardware and software interfaces.	The voice control interface may not respond efficiently in noisy environment.	3 (Medium)	5 (High)	15

Table 4: Risk Analysis

A2d	Class: Product Engineering Element: Design Attribute: Performance	The performance attribute refers to time-critical performance risk involving response time and throughput requirements.	Voice Processing and issuing commands does not involve heavy computation.	1 (Low)	3 (Medium)	3
A2f	Class: Product Engineering Element: Design Attribute: Hardware Constraints	Covers target hardware with respect to the dependence on hardware to meet system and software performance requirements.	The voice recognition accuracy depends on the ambient conditions.	1 (Low)	3 (Medium)	3
A3a	Class: Product Engineering Element: Code and Unit Test Attribute: Feasibility	The risk in coding and testing arising from inherently difficult implementation needs.	Unavailabilit y of multi threading libraries in Arduino Mega 258	3 (Medium)	3 (Medium)	9

Table 4: Risk Analysis (Contd.)

A4a	Class: Product Engineering Element: Integration and Test Attribute: Environment	Risks related to adequacy of the environment to enable integration in a realistic environment or to fully test all functional and performance requirements.	The environment chosen may not be adequate to completely integrate the components of our hardware.	3 (Medium)	5 (High)	15
A5f	Class: Product Engineering Element: Engineering Specialties Attribute: Specifications	Specifications for the system, hardware, software, interface, or test requirements or design at any level with respect to feasibility of implementation.	Our system needs specific hardware requirements to engineer high sensitivity and good quality of results.	3 (Medium)	3 (Medium)	9

Table 4: Risk Analysis (Contd.)

A3a	Class: Product Engineering Element: Code and Unit Test Attribute: Feasibility	The risk in coding and testing arising from inherently difficult implementation needs.	Unavailability of multi threading libraries in Arduino Mega 258	3 (Medium)	3 (Medium)	9
A4a	Class: Product Engineering Element: Integration and Test Attribute: Environment	Risks related to adequacy of the environment to enable integration in a realistic environment or to fully test all functional and performance requirements.	The environment chosen may not be adequate to completely integrate the components of our hardware.	3 (Medium)	5 (High)	15
A5f	Class: Product Engineering Element: Engineering Specialties Attribute: Specifications	Specifications for the system, hardware, software, interface, or test requirements or design at any level with respect to feasibility of implementation	Our system needs specific hardware requirements to engineer high sensitivity and good quality of results.	3 (Medium)	3 (Medium)	9

Table 4: Risk Analysis (Contd.)

B1e	Class: Development Environment Element: Development Process Attribute: Product Control	Risks due to dependency of product control on traceability of requirements from the source specifications.	Our product is not ready to support change control depending on changing requirements from the source.	1 (High)	5 (High)	5
B2d	Class: Development Environment Element: Development System Attribute: Familiarity	Risks due to no prior use of the system by project personnel as well as inadequate training for new users.	The platform used to develop the product is completely new to us, there is no prior knowledge.	3 (Medium)	5 (High)	15
C1d	Class: Program Constraints Element: Resources Attribute: Facilities	Risk depending on the adequacy of the program facilities for development, integration, and testing of the product.	Hardware availability in given budget is a constraint, installation adds time constraints.	3 (Medium)	3 (Medium)	1

Table 4: Risk Analysis (Contd.)

Interrelationship Graph

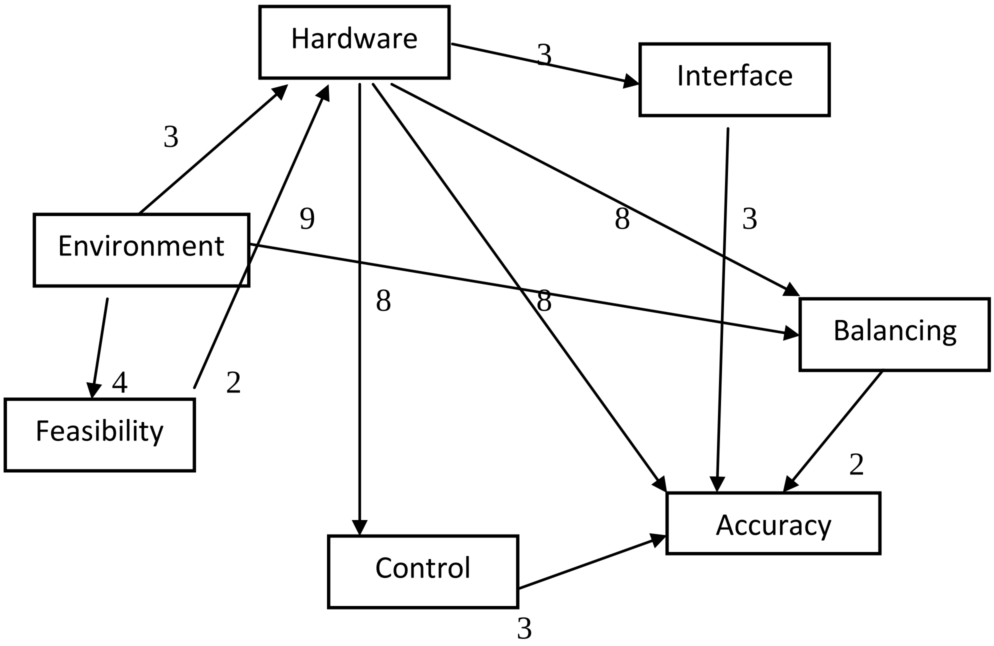

Fig 14: Interrelationship Graph

Risk Classification via Graph

S.N.	Risk Area	#of Risk Statements	Weights (In+ out)	Total weight	Priority
1	Hardware	8	3+9+8+8+8+3	39	1
2	Interface	6	3+3	6	7
3	Control	6	8+3	11	3
4	Accuracy	8	2+3+8	13	5
5	Balancing	3	3+8+8+2	21	2
6	Feasibility	2	4+2	6	6
7	Environment	5	3+9+4	16	4

Table 5: Risk Classification via Graph

Risk Description

Risk Statement	Risk Area	Priority
The hardware is very delicate and might need to be adjusted every now and then	Hardware	1
Any difference in weigh distribution might tend to disrupt the balancing while in motion	Balancing	2
Light and noise conditions change the control precision	Control	3
Qubo might respond differently in different terrains and light conditions	Environment	4

Table 6: Risk Description

3.5 Risk Mitigation

Risk ID	Mitigation Plan	Date started	Date completed
1.	The project will be well planned before starting with the implementation. All the steps to be taken will be well framed and well understood before beginning with the project.	July’11	December’11
2.	The Hardware will be planned and implemented with blueprint.	August’11	September’11
3.	The development board and other components/sensors will be well checked before beginning with the project and we will try to avoid most of the constraints that may occur.	August’11	December’11
4.	The interfaces will be developed so that actions are synchronized properly for the efficient working of my project.	September’11	December’11
5	Sensors will be debugged thoroughly before the actual implementation	2 August’11	20 August’11
6.	The quality of the project will be ensured according to the requirements.	October’11	December’11

Table 7: Risk Mitigation

7.	The quality of the code, documentation, design will be ensured so that we can develop a well manageable code.	August'11	December'11
8.	The project will be well documented and communicated to both the group members at every step so as to avoid uncorrelated project.	November'11	December'11

Table 7: Risk Mitigation (Contd.)

Chapter 4.

IMPLEMENTATION & TESTING

4.1 Implementation Details & Issues

4.1.1 Implementation Snapshots

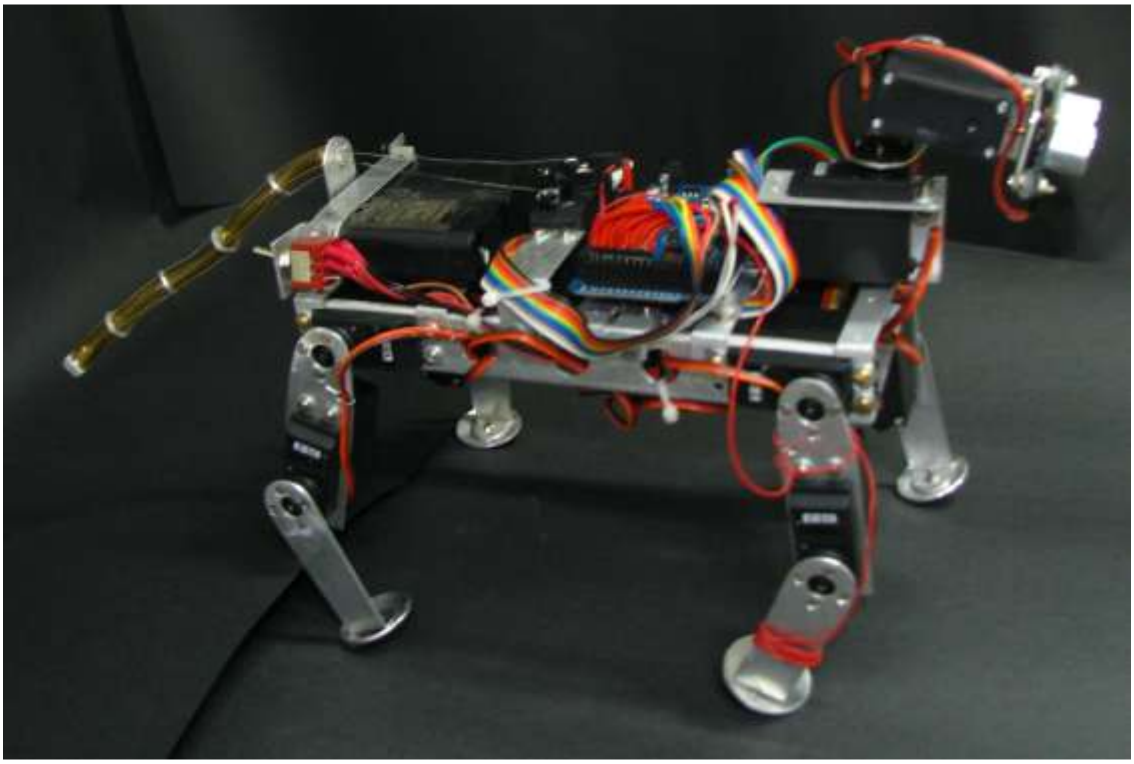

Fig 15: QUBO WALKING

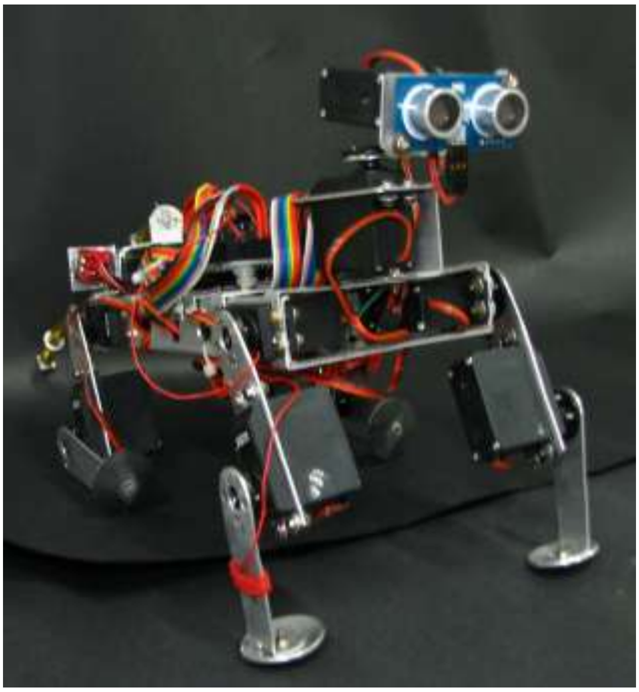

Fig 16: QUBO LYING PROFILE

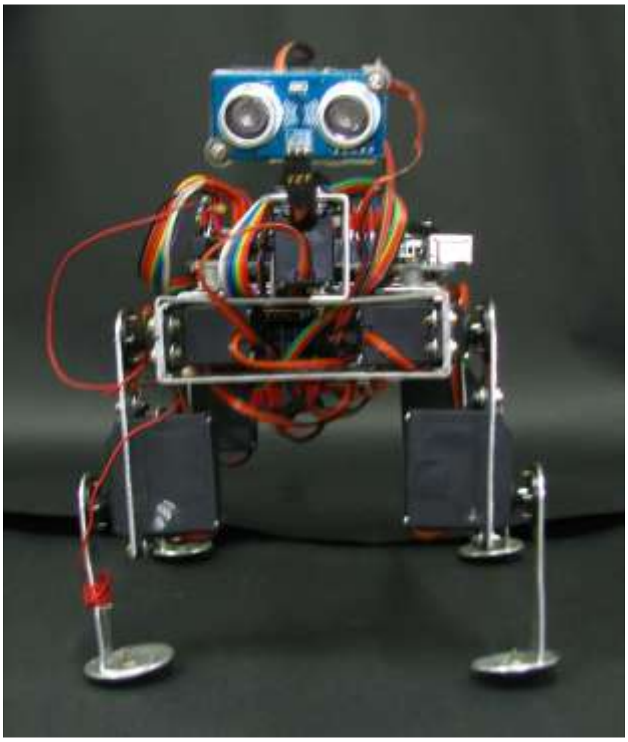

Fig 17: QUBO LYING FRONT

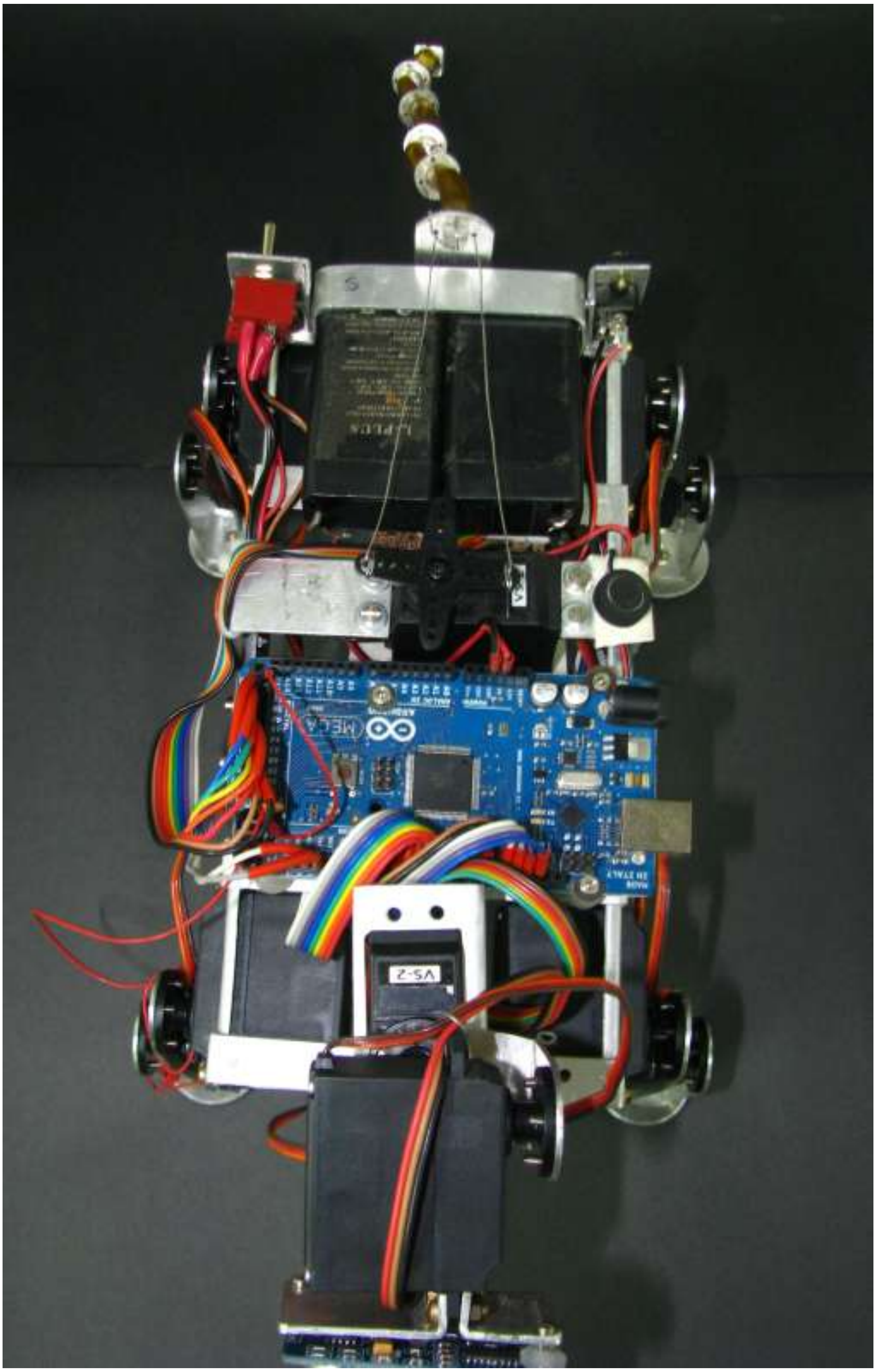

Fig 18: QUBO TOP

Implementation of Hardware

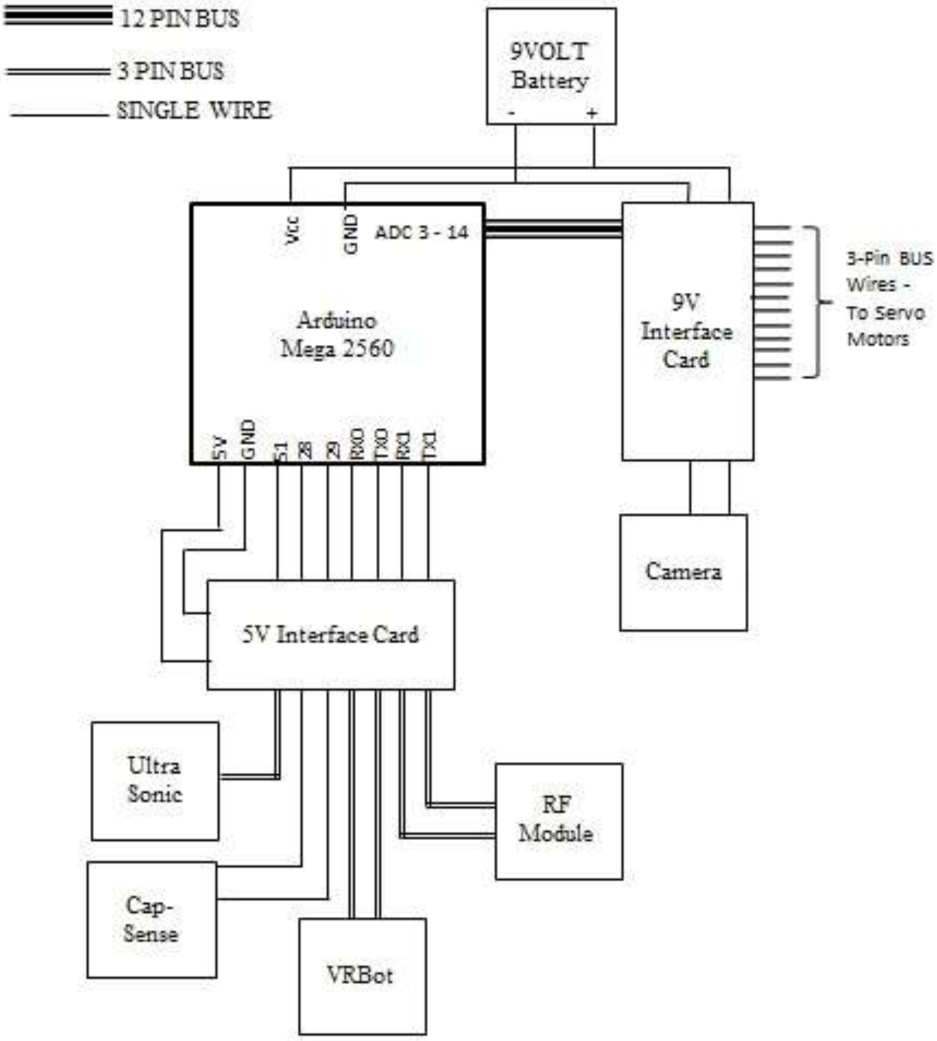

Fig 19: Hardware Connections

4.1.2 Debugging

RF Module

Debugging for the RF Module was done with a LOOP hardware setup, Hyper Term and the Processing Serial Monitor:

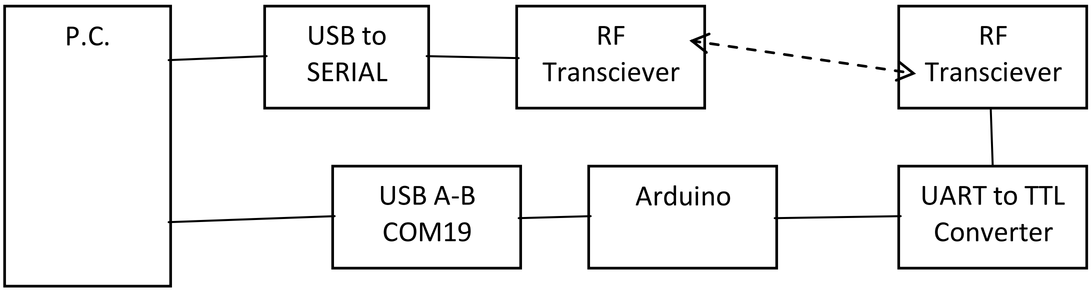

Fig 20: RF Module Working

The Data transmitted was “qwertyuiop”(LEFT) which was successfully received at the COM19 Terminal Monitor (RIGHT).

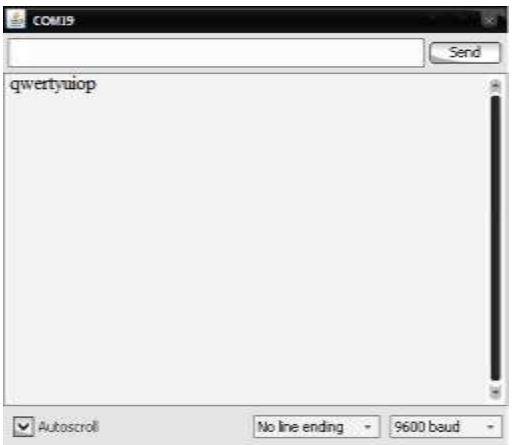

Fig 21: Debugging RF Module Using Hyperterminal & Serial Monitor

Ultra Sonic Sensor

The UltraSonic distance sensor was connected with the ArduinoDuemunilove for testing. The board was coded to print the values on the Serial port.

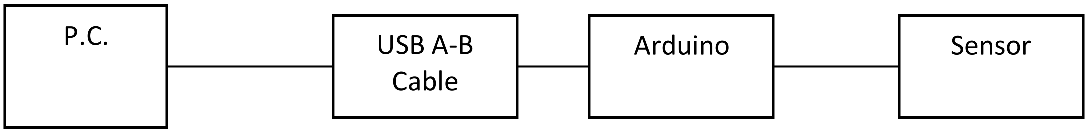

Fig 22: Debugging Ultra Sonic Sensor Using Serial Monitor

The sensor was tested for values, from 10cm to 250cm. It was found that the Sensor stops responding for distances above 200cm and has an error of (-+)3cm.

Code Snippet:

```
void loop()
{
long duration, inches, cm;
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingPin, LOW);
pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);
inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(100);
}
```

4.1.3 Error & Exception Handling

Incorrect Marker Detection

Due to the motion of the camera and disturbance in the feed, the artoolkit recognizes faulty markers (recognizes one marker as the other, or recognizes an arbitrary square object as a marker). This Error is overcome by taking the value of markers for a minimum of THREE trails. A marker is considered recognized ONLY if it occurs in three consecutive turns, else it is not considered. This accuracy-speed tradeoff has to be dealt with to avoid unwanted behaviors of Qubo.

Code Snippet:

```
if (recog_id!=-1)
{
val_recog=recog_id;
done_flag = 0;
}
elseif ((recog_id==-1)&&(no_of_times<MAX_TIMES))
{
no_of_times++;
}
elseif ((no_of_times == MAX_TIMES)&&(done_flag==0))
{
no_of_times=0;
done_flag = 1;
recog_id=val_recog;
……
}
```

Training of Voice

QUBO has been trained for THREE different voices instead of just one. This results in easy recognition of commands.

4.2 Risk Management

1. Balancing :

Response : With the help of eight motors (2 on each limb) placed accurately, correct balancing of Qubo is achieved.

Effectiveness : Motions like walking, running and dancing becomes very simple and smooth if the balancing is perfect.

2. Hardware and components study:

Response : All the datasheets and related information were studied properly before starting the project.

Effectiveness: No wrong current/voltage input/output were used.

3. Debugging / Compilation :

Response : We used hyper terminal and serial monitor to send and receive data to and from the computer

Effectiveness : Although additional time was required to debug the sensors before testing but it saved us from searching for benchmarks.

4.3 Testing

4.3.1 Testing Plan

Type Of Test	Will Test Be Performed?	Comments/Explanations	Components
Requirements Test	Yes	The requirements-based testing (RBT) process would validate that the requirements are correct, complete, unambiguous, and logically consistent.	-
Unit Testing	Yes	Testing of individual Components of the Robot. Modules like the Voice Module & Vision Module which works independently.	All Sensors can be tested for correct values.
Integration Test	Yes	Make sure that the interaction of two or more components produces results that satisfy functional requirement. Checks for Analog (Voltage) and Digital Output of various sensors to be properly integrated to the main board. Checks for interface misuse and interface misunderstandings.	Integrated Vision CapSense, UltraSonic sensor with microcontroller
Performance	Yes	Performing tasks repeatedly by testers to test the robot against its performance.	Complete Robot specially w.r.t different tasks again and again
Stress	Yes	Qubo would be tested on different terrains and with different obstacles to test the gait.	-

Table 8: Testing Plans

Compliance	No	There is no standard which exists for quadruped robots as of now.	-
Load	Yes	Qubo would be tested to perform operation for long durations.	Can be done by leaving one single command and not calling for the Robot for long.
Bugs	Yes	Code Development by keeping track of bugs in a multi-project, multi-developer environment, for a clean, simple, and fast interface.	Embedded Programs
Security	No	Not Applicable on Qubo.	-

Table 8: Testing Plans (Contd.)

4.3.2. Features to be Tested

Testing Id	Type of Test	Classification	Risk Area	Description	Level of Risk (1-Low, 2-Medium, 3-High)
1a	Requirement Testing	Power supplied	Current	Low current resets the Arduino.	2
2a	Unit Testing	VRBot	Training	Extra noise while training the VRBot hampers the training process.	1
2b	Unit Testing	Wireless 1.25GHz AV Camera	Range	Out of range causes a probable loss of signal.	2
2c	Unit Testing	Wireless 1.25GHz AV Camera	Obstruction	Obstruction causes disturbance in the output of the camera.	1
3a	Unit Testing	Wireless module CC250	Range & Obstruction	Out of range causes probable loss in data received or transmitted.	3
3b	Unit Testing	Ultrasonic Distance Sensor	Angle of Incidence	Change in the angle of incidence produces error	2
4a	Integration Testing	Voice and vision module	Synchronization	Processing is slowed down when voice and vision command at given at the same time	1
4b	Integration Testing	Wireless module and Arduino	Delay	Delay occurs when Arduino is performing some other function	1

Table 9: Features To be Tested

5a	Performance Testing	Walk operation	Different terrains	Speed might vary greatly from very high to negligible depending upon friction of terrain	2
5b	Performance Testing	Walk operation	Obstruction or presence of wall	The Robot tries to avoid the wall by moving right.	1
5c	Performance Testing	Voice Recognition	Signal and Noise	If the noise is above the maximum level, the robot would not detect the signal.	3
5d	Performance Testing	Voice Recognition	Small room and echo	In the case of a small room, the echo hampers in the voice detection	2
5e	Performance Testing	Marker Recognition	Ambient Lighting	Low levels of light result in fault marker recognition	2
5f	Performance Testing	Marker Recognition	Marker Color	Any other color than absolute black may not be recognized	3
5g	Performance Testing	Marker Recognition	Marker Size	Small Markers are not easily recognized	2
5h	Performance Testing	Marker Recognition	Distance	Markers should not be very far from the Robot	1
5i	Performance Testing	Getting up from any sit Position	Terrain	Very low friction coefficient may result in slipping	1

Table 9: Features To be Tested (Contd.)

5j	Performance Testing	Hello	Terrain	May result in misbalance if the terrain is not smooth	3
6a	Stress & Load	Continuous Working	Time Duration	Very long period of continuous working may result in over-heating.	1
6b	Stress & Load	Weight	Max Weight	Can't bare a weight more than 5kgs	2

Table 9: Features To be Tested (Contd.)

4.3.3 Features Not be Tested

- Voltage over runs can not be tested due to the fact that it burns out the components.
- Power supplied by the batteries: Due to the unavailabity of batteries and their high costs, the batteries are not to be included in this release of Qubo.
- Sit operation of Qubo: Low risk is involved in this operation since current rating is low and testing is already done for "Hello" operation, which is almost same as "Sit" operation.
- Open door: Actual implementation of opening the door is not included in the release of the version.

4.3.4 Approach Taken for Testing

Testing Id	Classification	Description of Risk Area	Type of Testing
1a	Power supplied	Low current resets the Arduino.	Black Box Testing
2a	VRBot	Extra noise while training the VRBot hampers the training process.	Black Box Testing
2b	Wireless 1.25GHz AV Camera	Out of range causes a probable loss of signal.	Black Box Testing
2c	Wireless 1.25GHz AV Camera	Obstruction causes disturbance in the output of the camera.	Black Box Testing
3a	Wireless module CC250	Out of range causes probable loss in data received or transmitted.	Black Box Testing
3b	Ultrasonic Distance Sensor	Change in the angle of incidence produces error	White Box Testing
4a	Voice and vision module	Processing is slowed down when voice and vision command at given at the same time	White Box Testing
4b	Wireless module and Arduino	Delay occurs when Arduino is performing some other function	White Box Testing
5a	Walk operation	Speed might vary greatly from very high to negligible depending upon friction of terrain	Black Box Testing
5b	Walk operation	The Robot tries to avoid the wall by moving right.	Black Box Testing

Table 10: Approach Taken for Testing

5c	Voice Recognition	If the noise is above the maximum level, the robot would not detect the signal.	Black Box Testing
5d	Voice Recognition	In the case of a big room, the echo hampers in the voice detection	Black Box Testing
5e	Marker Recognition	Low levels of light result in faulty marker recognition	Black Box Testing
5f	Marker Recognition	Any other color than absolute black may not be recognized	Black Box Testing
5g	Marker Recognition	Markers should not be very far from the Robot	Black Box Testing
5h	Getting up from any sit Position	Very low friction coefficient may result in slipping	Black Box Testing
5j	Hello	May result in misbalance if the terrain is not smooth	White Box Testing
6a	Continuous Working	Very long period of continuous working may result in over-heating.	Black Box Testing
6b	Weight	Can’t bare a weight more than 5kgs	Black Box Testing

Table 10: Approach Taken for Testing (Contd.)

4.3.5 Item Pass/Fail Criteria

- Unit Test Level

Testing Id	**Unit to be tested**	**Area of Testing**	**Pass/ Fail Criteria**
1	VRBot	Testing of VRBot under various conditions of noise and echo 1. Open Area 2. Small room (Conference room) 3. Large empty room (SAC)	Pass : Atleast 40% recognition efficiency Fail : <40 % Efficiency
2	Wireless 1.25GHz AV Camera	Out of range causes a probable loss of signal.	Pass : Video Feed without flicker Fail : Otherwise
3	Wireless 1.25GHz AV Camera	Obstruction causes disturbance in the output of the camera.	Pass : Video Feed without flicker Fail : Otherwise
4	Wireless module CC250	Out of range causes probable loss in data received or transmitted.	Pass : Video Feed without flicker Fail : Otherwise
5	Ultrasonic Distance Sensor	Change in the angle of incidence produces error 1. Two walls perpendicular to each other 2. Two/ more than two objects at an angle other than 90 degree	Pass : Distance less than 200 cms Fail : Otherwise
6	Marker Detection	Detection of markers in varying light intensity	Pass : Atleast 80% recognition efficiency Fail : <80 % Efficiency

Table 11: Item Pass/Fail Criteria (Unit Test Level)

- Master Test Plan Level

Testing Id	Module to be tested	Area of Testing	Pass/ Fail Criteria
1	Walk operation	Speed might vary greatly from very high to negligible depending upon friction of terrain 1. Smooth Floor(Tiles) 2. Rough Floor (Hostel Floor) 3. Soft Terrain (Ground) 4. Very Rough Terrain (Cardboard)	Pass : Should budge from its place at a constant speed Fail : Gets stuck in its motion
2	Getting up from any sit Position	Very low friction coefficient may result in slipping 1. Smooth Floor(Tiles) 2. Rough Floor (Hostel Floor) 3. Soft Terrain (Ground	Pass : Squats back without resetting or falling Fail : Falls or resets
3	Hello	May result in misbalance if the terrain is not smooth	Pass : Sits without resetting or falling Fail : Falls or resets
4.	Dance	May slip and result in misbalance on some terrains	Pass : Dances without resetting or falling Fail : Falls or resets

Table 12: Item Pass/Fail Criteria (Master Test Level)

4.3.6 Test Cases

- Black Box Testing

1. Power supplied (Voltage)

Test case Id	Input	Expected Output	Status
1a	3V-5V	No output	PASS
1b	>9V	Works fine	PASS

2. Power supplied (Current)

Test case Id	Input	Expected Output	Status
2a	1A	Reset	PASS
2b	2A	Reset	FAIL
2c	3A	Works fine	PASS

3. VRBot Training and commands

Test case Id	Input	Expected Output	Status
3a	Noise level at 10dB	No Data Loss	PASS
3b	Noise Level at 20dB	No Data Loss	PASS
3c	Noise level at 30dB	Data Loss	FAIL
3d	Command given at 20dB	Command not recognized	FAIL
3e	Command given at 60dB	Command recognized	PASS

4. Range of Wireless 1.25GHz AV Camera

Test case Id	Input	Expected Output	Status
4a	Range ~ 30inch	Disturbance	PASS
4b	Range ~ 50inch	Some disturbance	PASS
4c	Range ~ 1 m	No Disturbance	PASS
4d	Range ~ 2 m	No disturbance	FAIL

5. Wireless module CC2500 range and baud rates

Test case Id	Input	Expected Output	Status
5a	Range ~ 1m and baud rate 9600	No data loss	PASS
5b	Range ~ 5m and baud rate 9600	Some data loss	PASS
5c	Range ~ 5m and baud rate 19200	No data loss	PASS

6. Marker Recognition in different light condition

Test case Id	Input	Expected Output	Status
6a	Low light conditions	Markers not recognized	PASS
6b	Proper light conditions	Markers recognized	PASS

7. Distance of the markers from Qubo

Test case Id	Input	Expected Output	Status
7a	Range ~50 cm	Markers recognized	PASS
7b	Range ~ 1m	Markers sometimes not recognized	FAIL
5c	Range ~ 2m	Markers not recognized	PASS

8. Speed of walk operation on different terrains

Test case Id	Input	Expected Output	Status
8a	Very Smooth Floor	1cm per second	PASS
8b	Rough Floor	6cm per second	PASS
8c	Soft terrain (Sand)	2 cm per second	PASS

9. Speed of run operation on different terrains

Test case Id	Input	Expected Output	Status
9a	Very Smooth Floor	1cm per second	PASS
9b	Rough Floor	14cm per second	PASS
9c	Soft terrain (Sand)	5 cm per second	PASS

10. Getting up from lie down/ sit position on different terrains

Test case Id	Input	Expected Output	Status
10a	Very Smooth Floor	Not possible	FAIL
10b	Rough Floor	Possible	PASS
10c	Soft terrain (Sand)	High risk of tilting	PASS

11. Stress/ Weight on Qubo

Test case Id	Input	Expected Output	Status
11a	No Weight	Operates normally	PASS
9b	Weight ~2 kg	Current rate increases	FAIL
9c	Weight ~ 6kg	Arduino board resets	PASS

- White Box Testing

1. HELLO Command

```
void qubo_hello()
{
  qubo_bend();
  pos_rhip=-35;
  pos_rkn=110;                    (A)
  pos_lhip=-35;
  pos_lkn=110;
  servos_write(7);
  pos_rsh=85;
  servos_write(7);                  (B)
  while(cs_29_28.capSense(30)<300            (C)
  && !Serial1.available()                  (D)
  &&!Serial2.available());                 (E)
  while(!Serial1.available()               (F)
  &&cs_29_28.capSense(30)>=300     (G)
  &&!Serial2.available())                  (H)
  {
    pos_rsh=85;
    servos_write(7);                           (I)
    pos_rsh=60;
    servos_write(7);
  }


  int bend_angle=15;
  pos_rsh=bend_angle;
  pos_rel=-bend_angle;                       (J)
  pos_lsh=bend_angle;
  pos_lel=-bend_angle;
  servos_write(3);
}
```

Control Flow Graph

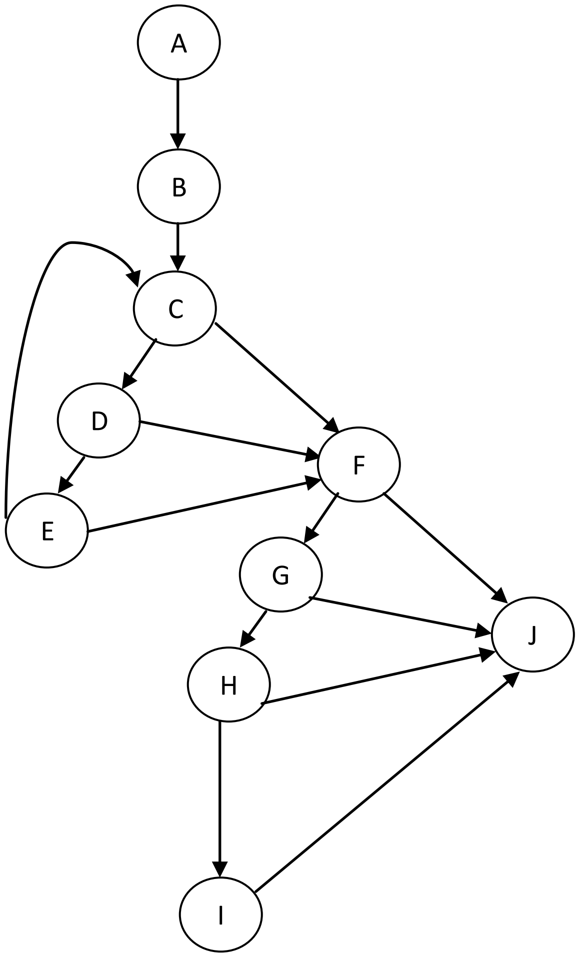

Fig 23: Control Flow Graph

Cyclomatic complexity: M= E-N+2P, where E-no of edges, N- no of nodes, P- no of connected components

M= 10-15+2*15= 25

2. Serial transfer of commands after marker detection

```
static void send_serial( int recog_id)
{
if (recog_id!=-1)                              (A)
{
val_recog=recog_id;
done_flag = 0;                                 (B)
}
else if ((recog_id==-1)                        (C)
&&(no_of_times<MAX_TIMES))                     (D)
{
no_of_times++;                                 (E)
}
else if ((no_of_times == MAX_TIMES)            (F)
&&(done_flag==0))                              (G)
{
no_of_times=0;
done_flag = 1;                                 (H)
recog_id=val_recog;
printf("%d\n",recog_id);
if (recog_id == 0)                             (I)
fprintf( port, "Z" ) ;                         (J)
else if (recog_id==1)                          (K)
fprintf( port, "F" ) ;                         (L)
else if (recog_id==2)                          (M)
fprintf( port, "!" ) ;                         (N)
}
fclose( port ) ;                               (O)
draw();
}
```

Control Flow Graph

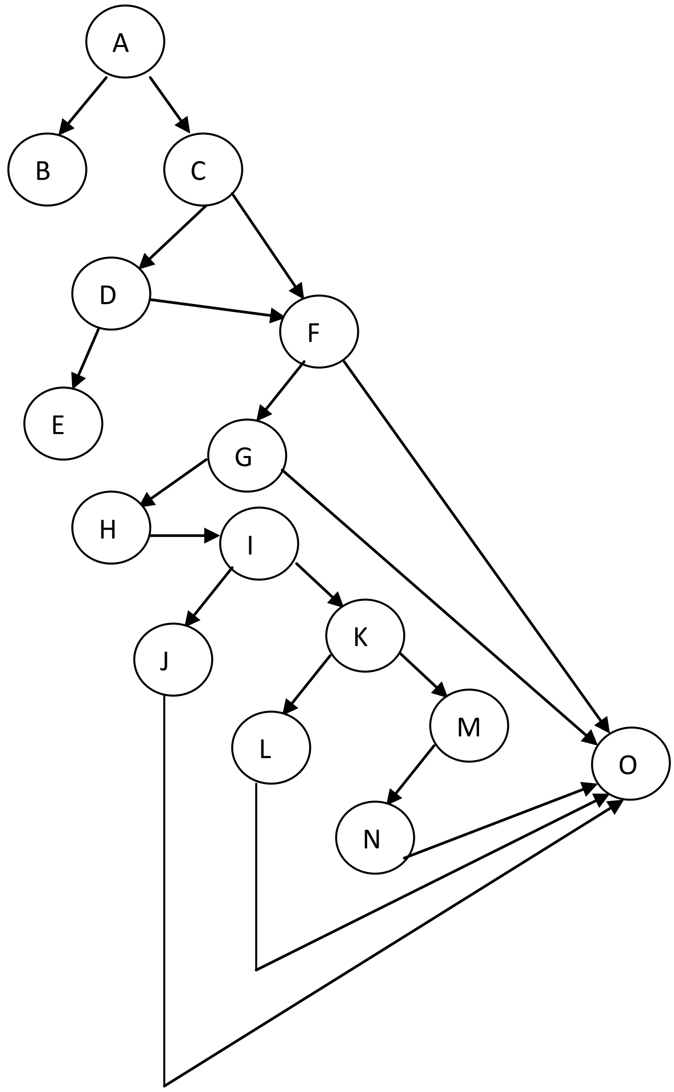

Fig 24: Control Flow Graph

Cyclomatic complexity: M= E-N+2P, where E-no of edges, N- no of nodes, P- no of connected components

M= 20-15+2*15=35

Chapter 5.

FINDINGS AND CONCLUSIONS

5.1 FINDINGS

A robot for entertainment requires a completely autonomous physical agent. A fully autonomous pet robot focusing on specific perceptual functions such as speech and vision, promotes and accelerates research activities involving the integration of subsystems.

A robot for entertainment can be effectively designed using various state-of-the-art technologies, such as speech recognition and vision, even though these technologies may not be mature enough for applications where they perform a critical function. While there exists special and difficult requirements in entertainment applications themselves, limited capabilities in the speech and vision systems may turn out to be an interesting and attractive feature for appropriately designed entertainment robots.

5.2 CONCLUSIONS

5.2.1 Watching Motions of a Robot:

Many people enjoy watching motions, gestures, and behaviors of animals in a zoo. Recently, computer controlled dinosaur robots have been very popular in theme parks. The ability to perform movements of a certain level of complexity is important for robots to really be entertaining.

5.2.2. Interacting with a Robot:

We can interact with the dog by cue cards and voices. In addition to watching the motions of the pet animal, the interaction enhances the owner's enjoyment. Interaction is an important feature of entertainment robot.

5.2.3. Controlling a Robot:

"Action" games are also popular in video game software. Many people enjoy controlling a car to compete for a time record, or a character to fight with another character controlled by another player. Robot soccer, for example, would be one of the most promising targets of applications. While RoboCup (Kitano, 1997b) focuses of research issues, consumers would certainly be interested in an easier way to play soccer with robots. Thus, controlling a robot will be one of the major application categories in Robot Entertainment.

5.2.4. Creating a Robot:

We believe that the creation of a robot is itself entertaining. Recently, many robot contests have opened, where participants had to create robots under some constraints such as weight, cost, and so on. These participants enjoy creating robots implemented with their ideas, and enjoy observing the behavior of the robots they have designed. We believe that not only technically-trained people but also laymen can enjoy creating a robot, if we can provide a friendly development environment.

5.3 FUTURE WORK

We believe that we will be able to create a completely new market in the near future by introducing this kind of robot products sharply focused on entertainment applications. We strongly believe that after the gold rush of the Internet and cyber-space, people will eagerly seek *real* objects to play with and touch. Robot Entertainment provides tangible physical agents and an unquestionable sense of *reality*. It is expected that various kinds of entertainment applications will be developed for Robot Entertainment systems. The possibility of entertainment using such an autonomous robot is as follows:

REFERENCES

1. Fujita & Kitano, Development of an Autonomous Quadruped Robot for Robot Entertainment, in*Autonomous Robots, Volume 1, 1998,* Kluwer Academic Publishers, Boston.
2. Atmel Corporation, Atmega 2560, in *8-bit Atmel Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash,* San Jose, USA.
3. Lianqing Yu Yujin Wang&Weijun Tao, Gait Analysis and Implementation of a Simple Quadruped Robot, in *2010 2nd International Conference on Industrial Mechatronics and Automation,* China
4. Katsuhiko Inagaki &Hisato Kobayashi, Dynamical Motion Control for Quadruped Walking with Autonomous Distributed System in *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems 1994.'Advanced Robotic Systems and the Real World', IROS '94,* Tokyo, Japan.
5. H Kitano, M Fujita, S. Zrehen& K. Kageyama, Sony Legged Robot for RoboCup Challenge in *Proceedings of IEEE International Conference on Robotics and Automation, 1998.*
6. T. Arai Achi, H. Adachi, and K. Hormna, "IntegratedArm and Leg Mechanism and Its Kinematic Analysis," *Proceedings of the IEEE International Conference on Robotics and Automation, 1995*
7. N. Koyachi, H. Arai, K Asami, and Y. Itoh, "Hexapod with Integrated Limb Mechanism of Leg and Arm,"*Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 1952-1957, 1995.*
8. S. Hirose and K. Kato, "Development of Quadruped Walking Robot with the Mission of Mine Detection and Removal - Proposal of Shape-Feedback Master-Slave Arm - ," Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 1713-1718, 1998.
9. Yong Tao, Hongxing Wei &Tianmiao Wing, "A Speech Interaction System based on Finite State Machine for Service Robot", Proceedings of the IEEE International Conference on Computer Science and Software Engineering, 2000.
10. CristineEnglund "Speech recognition in the JAS 39 Gripen aircraft- adaption to speech at different G-loads" Master thesis in Speech Technology
11. Yong Tao, Hongxing Wei, Tianmiao Wang "A speech interaction system based on finite state machines for Service Robot" 2008 International Conference on Computer Science and Software Engineering

12. Ahmad A.M.Abushariah, Teddy S. Gunawan& Othman O. Khalifa "English digit speech recognition system based on Hidden Markov Model" International Conference on Computer and Communication Engineering (ICCCE 2010), 11-13 May 2010, Kuala Lumpur, Malaysia

13. M.P. Paulraj, S. Yaacob, S. A. Mohd Yusof "Vowel Recognition based on Frequency Ranges determined by Bandwidth Approach" Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on Digital Object Identifier

APPENDIX A

The work break down structure of the project is as follows:

Task A: Research Paper Survey, Hardware Survey and Library Study

Task B: Surveying for existing products and their prices

Task C: Collecting information on available motors and control boards, etc.

Task D: Checking the available hardware and ensuring a rough draw to use the same.

Task E: Creating a prototype out of hard board to test the feasibility of the project and test the weight carrying capabilities of the servo motors.

Task F: Started with Code development to simultaneously use several motors and measure the required current and voltage values.

Task G: Building the exoskeleton version 2.0 with aluminum as base.

Task H: Attaching the different modules, one after another, such that centre of balance is always maintained.

Task I: Developing a generic read , write, map and remap function for the motor set so as to ease the manual work of finding the required angle and arbitrarily setting a zero value for each motor on the quadruped.

Task J: Developing basic actions like stand, bend, lie, tail wag.

Task K: Developing the Dance action.

Task L: Developing Gait by studying actual quadruped walk from videos.

Task M: Developing the Hello action.

Task N: Understanding and implementing the Distance Sensor Module.

Task O: Understanding and implementing the Voice Recognition Module.

Task P: Training the voice recognition Module for different users and adding Voice Password.

Task Q: Building the exoskeleton version 3.0 with a smaller body base so as to improve torque to weight ratio.

Task R: Developing a capacitive touch sensor and a CapSense Library.

Task S: Tweaking motor parameters to achieve better quality of result and lower power consumption.

Task T: Installing Wireless Modules

Task U: Developing marker detection modules.

\

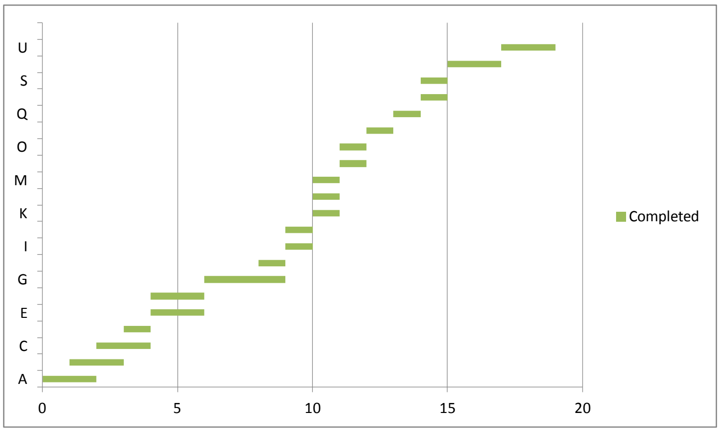

Fig 25: Gantt Chart

APPENDIX B

Description of Tools

Processing Based Arduino Alpha Version 0022

The open-source Arduino environment makes it easy to write code and upload it to the I/O board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing, avr-gcc, and other open source software. It is based on processing by Caset Reas & Ben Fry

Web Link: http://www.arduino.cc

Developers: Massimo Banzi, David Cuartielles and Tom Igoe.

ARToolkit

ARToolKit is a computer tracking library for recognizing Markers and then developing AR applications. To do this, it uses video tracking capabilities that calculate the real camera position and orientation relative to square physical markers in real time. Once the real camera position is known a virtual camera can be positioned at the same point and 3D computer graphics models drawn exactly overlaid on the real marker. We are using the View Point Tracking feature of artoolkit to detect and track Markers in Qubo's Environment.

Web Link: http://www.hitl.washington.edu/artoolkit/

Developers: Hirokazu Kato of Nara Institute of Science and Technology

HyperTerminal

Hyper Terminal is a Windows 7 Terminal Interface program capable of connecting to systems through TCP/IP Networks, Dial-Up Modems, and COM ports.

Web Link: http://www.hilgraeve.com/hyperterminal/

Developers: Hilgraeve

Microsoft Visual Studio 2010

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services .

Web Link: http://www.microsoft.com/visualstudio/en-us

Developers: Microsoft Corporation

Honestech TVR 2.0

Honestech TVR 2.0 encodes and compresses video data in real-time from video devices connected to a PC: TV Tuner, VCR, or analog camcorder. The TVR Software is being used to set the Receiver of the TVR which in-turn would be connected to the Wireless Camera.

Web Link: http://www.honestech.com/main/TVR20.asp

Developers: Honest Technology

Robosoft Systems CC2500 Configuration Utility

This High Speed CC2500 Configuration Utility allows engineers of all skill levels to quickly and cost-effectively add wireless capabilities to the CC2500 Module.

Web Link: http://www.robosoftsystems.co.in/roboshop/cc2500-wireless-module.html

Developers: RoboSoft Systems

APPENDIX C

Bio-Data of Students

Name	Vidur Shailendra Bhatnagar
Areas of Interest	Software Development in Java or C/C++, Unix Shell Scripting, Networking Programming.
Working Experience	Developed a "Complete Office Automation" software using Java and Swings, working autonomously as an independant software developer in Rohits & Co. India Pvt. Ltd., Roorkee, Uttarakhand from 1st July, 2011 till 1st August, 2011.
Internship	Completed a billing system using Java and Swings, in BHEL, Haridwar from 1st June, 2011 till 30th June, 2011.
Projects Undertaken	1. "BLOOM"- Bluetooth Messenger for Symbian Phones A comprehensive mobile application for S40 mobile phones with features like instant Bluetooth messenger, file transfer, remote calling and remote access to PC. 2. WORD SCRAM- Word formation game in C++ Player has to form maximum possible words through keyboard input from a random selection of 6 English letters. The project uses the concept of "Dynamic Programming" to achieve a linear time complexity. 3. UMESSENGER- LAN MESSENGER in Unix UMessenger, based on shell scripting intertwined with C programming in GCC, is a LAN chat application for Unix servers which uses sockets as its communication gateway.
Contact Details	+91-9650843773 vidur.bhatnagar@gmail.com

Name	Rachit Magon
Areas of Interest	Robotics & Embedded System Programming Web Technologies C/C++ Programming
Internship	Completed a 4 weeks project in JSP and Android Application Development with Tata Consultancy Services, Lucknow in June-July 2011.
Projects Undertaken	1. Wireless FM Feedback Motion Sensor Rifle: A Wireless gun which can be controlled by the movement of the hands after wearing a pair of gloves. 2. Smart Walking Stick: An audio-enabled walking stick for the visually impaired. 3. IGNITE (Internet Generic Networking & Information Terminal): Developed a networking application with SMS, and information crawling support in JSP.
Contact Details	+91-9582141137 rachit.magon@gmail.com

Name	Rashi Srivastava
Areas of Interest	Robotics & Embedded System Programming Web Technologies C/C++ Programming
Internship	Completed a 4 weeks project in JSP and Android Application Development with Tata Consultancy Services, Lucknow in June-July 2011.
Projects Undertaken	1. Wireless FM Feedback Motion Sensor Rifle: A Wireless gun which can be controlled by the movement of the hands after wearing a pair of gloves. 2. Smart Walking Stick: An audio-enabled walking stick for the visually impaired. 3. IGNITE (Internet Generic Networking & Information Terminal): Developed a networking application with SMS, and information crawling support in JSP.
Contact Details	+91-9582141128 rashi.srv89@gmail.com