IOT BASED VOICE CONTROLLED HOME ASSISTANCE HEXAPOD

NIKHIL SUBRAMANIAN|SHANJU SHRI MN| ROHITRAM V| DR SRIHARIPRIYA KC| SENSE

Introduction

The IoT based hexapod is designed in such a way that it acts as a companion to humans while performing tasks for them. Hexapod robots have 3 pairs of legs which is an advantage compared to wheeled robots since this robot can easily navigate even through rough and uneven surfaces. Google Assistant is connected to the system to accept voice commands from the user. Voice based navigation is an added advantage for old aged people. An important feature of this robot is surveillance, the camera attached to the hexapod helps in monitoring the home from anywhere using video streaming. Automation of electronics is done based on the commands given by user for which Adafruit and IFTTT platforms are used.

Motivation

The project aims at building an automated six legged hexapod that responds to voice commands to perform tasks.The movement of the hexapod is controlled by an arduino using the servo motors attached to each leg. The Lidar is used to collect data about the obstacles lying around the home while mapping the whole house.For mapping, the hexapod is manually controlled using a joystick. Google Assistant intakes the voice commands given and processes it before passing it to the NodeMCU which then pushes the data over to the Adafruit Platform by using the MQTT protocol..Adafruit along with the IFTTT platform is used to automate the electronics in the house based on the voice command received from the user. Live video is also streamed for surveillance purposes.

Methodology

A. Mapping

The shortest distance between the start point and the target point is calculated using a map based planning method in which the shortest distance is chosen based on the lowest movement cost. The detailed mapping of the house is done using a LIDAR mounted on the top of the hexapod. The LIDAR sensor works based on the pulsed light waves that it emits into the environment from a laser. These pulses then enter the sensor after bouncing off from nearby objects.The hexapod is manually operated using a joystick to navigate through the maze to draw a predefined map of the rooms as shown in Fig.5.

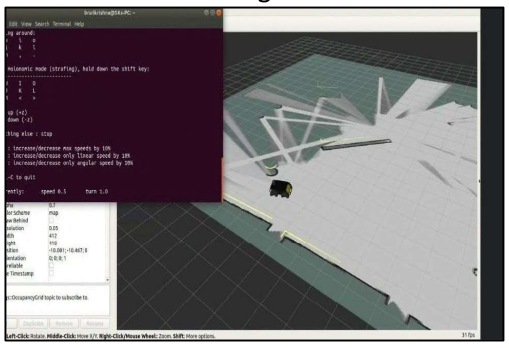

Fig. 5. Mapping the rooms by manually operating the hexapod.

Fig. 6. Image of dining room captured

Fig.7.Image of entrance of bedroom captured by the hexapod camera

B. Surveillance and live streaming

The surveillance is done using a Raspberry Pi camera module. The camera is mounted on top of the hexapod for a better Field of view. Camera module is enabled in the Raspberry Pi in order to use the camera. The camera is connected to the Pi CSI port. After setting up the camera, the video streaming web server is accessed at the Raspberry Pi's IP address to monitor the house from anywhere and at any time.

C. Voice navigation and automation

Fig. 8. Adafruit dashboard.

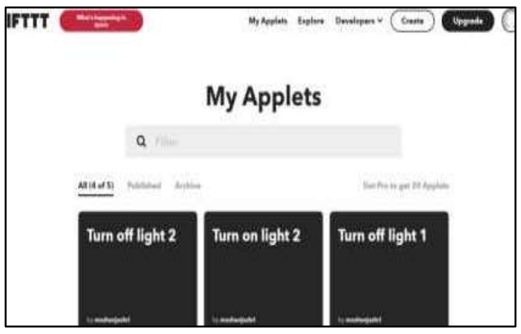

Fig. 9. Applets created in IFTTT.

In the Adafruit platform a new dashboard named 'Automation' is created and two triggers are added to it. The triggers are named Light 1 and Light 2 with values 0 and 1 for switching ON and switching OFF respectively. The Fig.8 shows the dashboard with two triggers. The Fig.9 shows the applets created in the IFTTT platform which triggers the functions to turn on and off the lights based on the given commands.

Result

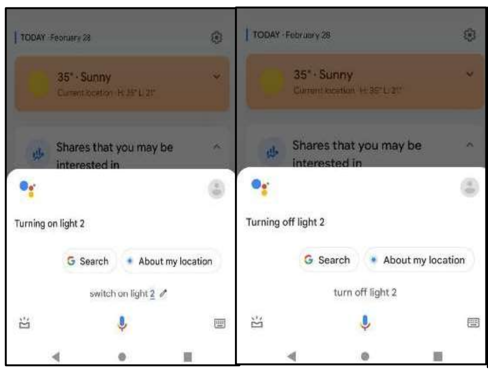

Fig. 15. Voice commands given to Google Assistant to turn on and off light 2.

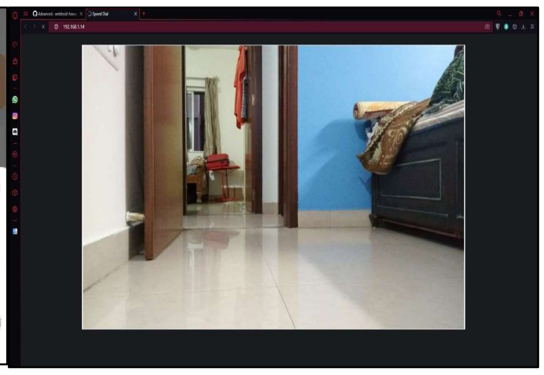

Fig.14 Live streaming

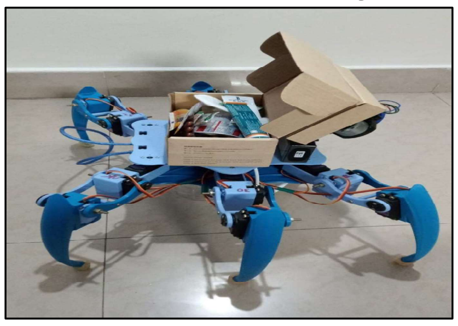

Fig.13 Hexapod carrying medicines to the user.

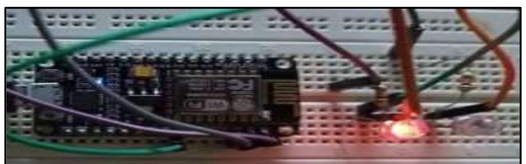

Fig. 16. When one of the LEDs is turned ON.

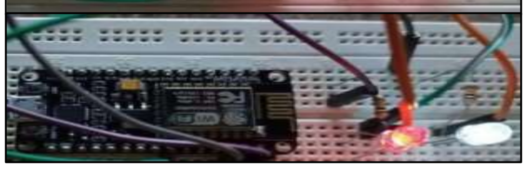

Fig. 17. When both the LEDs are turned ON.

- The hexapod functions successfully and expected results are obtained. The hexapod reaches the target destination to work on the task given as command. The hexapod moves easily through rough terrains in a stable way even when it's carrying medicines as shown in Fig. 13. Thus, it is able to carry things without them falling off the hexapod.
- The camera in the hexapod captures the images and video is streamed in the website. The following image Fig.14 represents the screenshot of the live streaming page. Thus the user can monitor the house from anywhere.
- The voice commands are given using the Google Assistant to turn on the lights in the room. Google Assistant passes the commands to the system and responds to the given commands as shown in Fig.15.
- LEDs represent the electronics in the rooms. When the user commands the LEDs to turn ON and OFF, the tasks are performed accordingly. Fig.16 and Fig.17 represent when light 1 is turned ON while light 2 is turned OFF and both the lights are turned ON respectively.
- Home Assistance Hexapod shows some similarities in developing and controlling the robot, but the features and application are vastly different from the previous works. This robot is specifically built to act as an assistant for people at home with features like carrying things, automating electronics and surveillance. Unlike the sensors used in other works, Lidar has a large measurement range and the accuracy is also very good.

Summary

The home assistance hexapod acts as a perfect companion and assistance for people at home, especially for sick and elderly people to meet their needs. The main objective of this system is achieved by integrating the Internet of Things with robotics successfully. The camera attached to the robot works effectively for monitoring the home even when the user is not at the site. Voice controlled automation proves to make the hexapod more user-friendly.This work can be further improved by integrating it with more sensors like moisture sensor, temperature sensor, gas sensor to detect the moisture level in the soil to notify the user regarding watering plants, to automate turning on/off air coolers, and to notify users in case there is a gas leak. Machine learning algorithms can also be applied on the captured videos to identify faces and allow people inside the house.

Contact details

rohitram.v2018@vitstudent.ac.in, nikhil.subramanian2018@vitstudent.ac.in, mn.shanjushri2018@vitstudent.ac.in

Reference

[1]Richik Ray, Rishita Shanker et al. "Automated Quadruped Robot Simulation using Internet of Things and MATLAB", Journal of Physics: Conference Series, 2021.
[2]Mostafa Khazaee, Majid Sadedel and Atoosa Davarpanah, "Behavior-Based Navigation of an Autonomous Hexapod Robot Using a Hybrid Automaton", Journal of Intelligent & Robotic Systems, May 2021.
[3]Abdelrahman Sayed Sayed, Hossam Hassan Ammar and Rafaat Shalaby, "Centralized Multi-agent Mobile Robots SLAM and Navigation for COVID-19 Field Hospitals", 2nd Novel Intelligent and Leading Emerging Sciences Conference, October 2020.
[4]Md. Imran Uddin, Md. Shahriar Alamgir et al. "Multitasking Spider Hexapod Robot", IEEE International Conference on Robotics, Automation, Artificial Intelligence and Internet-of-Things (RAAICON), 2019.

IoT based Voice Controlled Home Assistance Hexapod

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

by

Rohitram V

18BEC0811

Nikhil Subramanian

18BEC0711

Under the guidance of

Dr. Sriharipriya K C

School of Electronics Engineering

VIT, Vellore.

May, 2022

DECLARATION

We hereby declare that the thesis entitled “IoT based Voice Controlled Home Assistance Hexapod” submitted by us, for the award of the degree of *Bachelor of Technology in Electronics and Communication Engineering* to VIT is a record of bonafide work carried out by us under the supervision of Dr. Sriharipriya K C.

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 01-05-2022

Nikhil Subramanian

Type text here

Rohitram V

Signature of the Candidates

CERTIFICATE

This is to certify that the thesis entitled “IoT based Voice Controlled Home Assistance Hexapod” submitted by Nikhil Subramanian (18BEC0711), SENSE, VIT and Rohitram V (18BEC0811), for the award of the degree of *Bachelor of Technology in Electronics and Communication Engineering*, is a record of bonafide work carried out by her under my supervision during the period, 03. 01. 2022 to 30.04.2022, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : 01-05-2022

Dr. Sriharipriya K C

Signature of the Guide

Internal Examiner

External Examiner

Dr. Prakasam P

Head of the Department

Electronics and Communication Engineering

ACKNOWLEDGEMENT

We hereby acknowledge that we have worked on the thesis, “IoT based Voice Controlled Home Assistance Hexapod” in its entirety. First and foremost, we offer my sincerest gratitude to my teammates who have worked throughout the project with us. We are grateful to Dr.Sriharipriya, Associate Professor, Department of Embedded Technology, SENSE, our project guide, for her enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped me tremendously at all times in our research and writing of this thesis.

Nikhil Subramanian

Rohitram V

Student Name

Executive Summary

IoT-assisted robotics applications are a palpable fact of our forthcoming future, thanks to the ongoing Internet of Things revolution and the rapid dispersion of robots in many activities of daily life. IoT based Hexapod is one such application which can be used for various purposes from hospitals to the military which has serious upsides in the current and future generations. This work aims to design a hexapod for helping people, especially old-aged or sick people at home with automation, voice based navigation and surveillance. Sensors are used to detect obstacles and the hexapod can choose a different path. Electronic devices in each room are controlled through voice commands. The robot is also designed to help the user by carrying objects like medicines, files, etc., to various areas in the home based on the given commands. The home is monitored and the video will be live streamed for the user to monitor their home from anywhere at any time.

CONTENTS

List of Figures

List of Abbreviations

IoT	Internet of Things
IoRT	Internet of Robotic Things
ROS	Robot Operating System
MQTT	MQ Telemetry Transport
IFTTT	If This Then That
LIDAR	Light Detection and Ranging

1. INTRODUCTION

1.1 Objective

The IoT based hexapod is designed in such a way that it acts as a companion to humans while performing tasks for them. Hexapod robots have 3 pairs of legs which is an advantage compared to wheeled robots since this robot can easily navigate even through rough and uneven surfaces. Each leg has 3 servo motors which adds up to 18 servo motors overall, which are controlled by an arduino.

Google Assistant is connected to the system to accept voice commands from the user. Voice based navigation is an added advantage for old aged people. Another important feature of this robot is surveillance, the camera attached to the hexapod helps in monitoring the home from anywhere using video streaming. Lidar enables the hexapod to identify the presence of an entity and helps in avoiding human-robot collision. It also helps in mapping and path planning. Automation of electronics is done based on the commands given to the hexapod by the user for which Adafruit and IFTTT platforms are used. The data is given as input to the Adafruit platform which supports MQTT protocol using IFTTT and commands are executed.

1.2 Background

Robots are replacing humans in almost every sector, they get the job done more precisely and quickly. It is expected that the stock of robots could reach almost 20 million by 2030, with automated workers taking upto 51 million jobs in the next decade. They will eventually become integral parts of our daily routines. Similarly, the Internet of Things is also gaining popularity in recent times.

For most of the technical history, the focus has been on making technologies connect with humans or using devices to communicate with other humans. From switching on the appliances to controlling the room temperatures, the Internet of Things concept is applied. The internet is utilized to send emails to other humans, keep tabs on them on social media, and buy items from them. The goal of the Internet of Things is to reduce human involvement in the communication network. The internet becomes a network of gadgets talking to one another, rather than a network of persons connected through technologies. The Internet of Things is essentially a network of things that can communicate with one another.

Robots can be linked into the Internet of Things infrastructure, allowing for connections between various entities via various communication protocols. The Internet of Robotic Things refers to robots that are connected to Internet of Things technology and can communicate with other objects through the Internet. Intelligent devices can use IoRT to monitor events, combine sensor data from several sources, and employ both local and distributed intelligence to choose the best course of action. This is followed by a physical act of controlling or manipulating objects.

A lot of beneficial aspects are driving the worldwide Internet of Robotic Things market. The use of the internet of robotic things is being boosted by the expansion of the e-commerce business and the expansion of application fields due to the integration of robots with various technologies. This market is being driven by the increasing usage of smart devices, automation in manufacturing processes, and the growth of e-commerce. One such application of Internet of Robotic Things is home assistant hexapod. The hexapod is used for different purposes from hospital to military.

1.3 Literature review

Abdelrahman Sayed Sayed et al. have worked on a fully autonomous centralized multi robot system specifically for Covid-19 field hospitals. This system is designed in such a way that it performs tasks like disinfecting, detecting remote body temperature.., A master PC is used as a publisher and Raspberry Pi mounted on robots acts as a subscriber. A Kinect sensor is being used to map the hospital area which is also used as a thermal camera to detect people with high body temperature.

Abhishek C et al. integrated a robotic system with an embedded system of digital image processing. They have equipped the hexapod with a camera for capturing videos in real-time and a distance sensor to detect and avoid obstacles. The Yolo algorithm is being used to detect objects.

Anita Chaudhari et al. have developed a prototype for a quadruped robot using IoT for delivering medicines and essentials to Covid-19 patients. Ultrasonic sensors are used for detecting obstacles. The movement of the robot will be controlled using an android application which can be accessed by doctors and other medical staff. Fernando Gomez-Bravo et al. proposed a hybrid legged wheeled mobile robot named R3HC. This system was designed in such a way that it calculates optimum path planning trajectories based on the size of the obstacle and power consumption efficiency. Similarly, hybrid control architecture tends to increase body stability.

G.Anandravisekar, et al. have built a surveillance robot with metal detectors and a camera for live streaming. Cayenne software is being used to control the robot both in manual and automatic mode. FireBird V robot is the 5th in the FireBird robot series. Mahesh Pawaskar et al. preferred wireless communication between Firebird-V based hexapod robot and the user using Raspberry Pi through WiFi. The activities of the robot are controlled by a webpage through Raspberry Pi and the information about the position of the joystick is fed to Raspberry Pi to move the position of the camera for surveillance purposes.

Md. Imran Uddin et al. focused on improving climbing techniques (by using suction cups), walking angles and payload. This model is specifically designed for human safety purposes during disasters like earthquake, fire incidents etc., Infrared sensors are used for detecting dark and light surfaces and a sound sensor module is used for detecting sound. The robot also streams live videos in Raspberry Pi's IP address.

Mostafa Khazaee, Majid Sadedel and Atoosa Davarpanah have worked on behavior based autonomous hexapod with goal attraction and obstacle avoidance as its main behaviors. They have achieved the results using hybrid automaton. Richik Ray et al designed a quadruped robot using IoT and MATLAB. Webcam for surveillance, operational voice control, checking the health status, battery, direction of the bot, speed check are few features of this robot.

Vaibhav Malhotra has proposed a system for helping aged people in airports with their luggages. Self Designed belts are used for recognizing the owner of the luggage using Yolo algorithm. Carrier will be connected wirelessly to the user's phone app by scanning QR code on it.

2. PROJECT DESCRIPTION AND GOALS

The project aims at building an automated six legged hexapod that responds to voice commands to perform tasks. The detailed block diagram of the project is shown in Fig.1. The movement of the hexapod is controlled by an arduino using the servo motors attached to each leg. The Lidar is used to collect data about the obstacles lying around the home while mapping the whole house. This data is then used to prepare a detailed map of the rooms. For mapping, the hexapod is manually controlled using a joystick. Once the mapping is done, the hexapod is ready to move around the house and identify the obstacles on its way without bumping into anyone or anything.

Google Assistant intakes the voice commands given and processes it before passing it to the NodeMCU which then pushes the data over to the Adafruit Platform by using the MQTT protocol. The entire process is carried out using WiFi as the wireless technology.

Adafruit along with the IFTTT platform is used to automate the electronics in the house based on the voice command received from the user. Live video is also streamed for surveillance purposes.

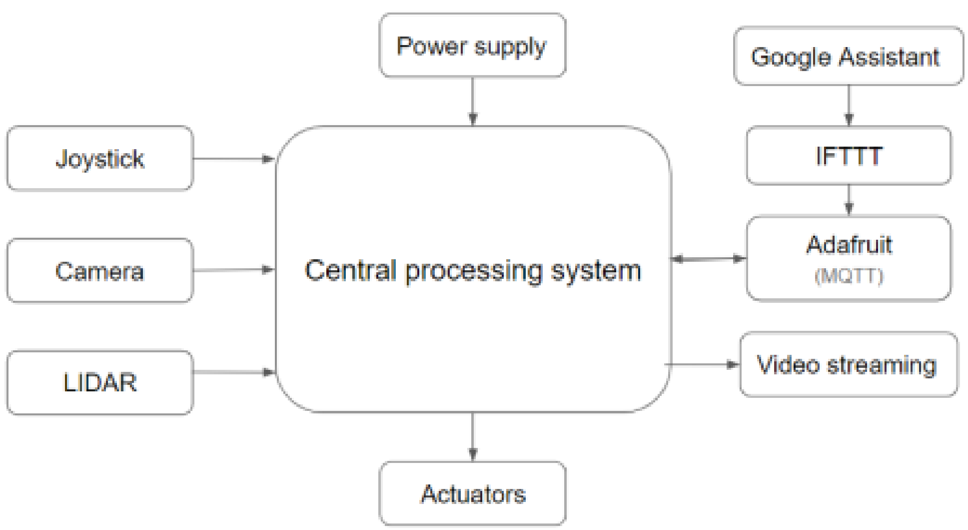

Fig.1 Block diagram of the six legged robot

3. TECHNICAL SPECIFICATION

3-D printing, also known as additive manufacturing, is a method of creating three-dimensional objects from a CAD model or a digital 3D model. To create the parts of the hexapod, 3-D printing process is used. The parts are built firm through 3-D printing.

Table.1 List of 3D printed parts

Part	Layer Height	Infill	Support	Number of Pieces	Plastic (g)	Material
Thigh	0.2	40%	Yes	6	189	PLA
Femur	0.25	40%	No	6	30	PLA
Femur symmetry	0.25	40%	No	6	30	PLA
Separator	0.2	40%	No	6	9	PLA
Feet	0.2	40%	Yes	6	342	PLA
Tap servo	0.1	50%	Yes	18	128	PLA
Base inferior	0.2	25%	No	1	156	PLA
Base superior	0.2	25%	No	1	158	PLA

Body separator	0.2	40%	No	2	19	ABS
Camera support	0.2	40%	No	1	20	ABS

Table 2. List of parts required to join and hold the parts together

Component	**Used in**	**Number**	**Type**
Bolt M2.5x20 countersunk head	Femur separator	12	A
Bolt M3x20 hexagonal head	Servomotor	72	B
Nut M4 auto blocking	Modified cap	18	C
Bolt 2.2x6.5 self tapping	Horn/Arduino/RPi	80	D
Bolt M4x16 hexagonal head	Modified cap	18	E
Nut M3 auto blocking	Pi	6	F
Washer M4	Modified cap	36	G
Bearing 4x10x4	Femur	12	H

Table 3. List of technical components

Component	Number
Towerpro MG996R 12kg Servo	18
Rubber End Cap D10mm	6
Lipo Battery	1
XT60 Connector (Male)	1
YEP 20A SBEC	1
Arduino Mega 2560 r3	1
Arduino Mega Proto Shield R3	1
Male Pin Strip 0.1"	60
DC 7V-24V To DC 5V 3A USB	1
RPi 3 B+	1
32GB SD Card	1
Raspberry Pi Camera Board	1

4. DESIGN APPROACH AND DETAILS

4.1 Design approach

4.1.1. Assembling the hexapod and calibration

The parts of the six-legged robot like femur, base, thigh etc., are 3D printed which are designed in CATIA V5r21 with a full parametric design. Each leg is fitted with 3 MG996R servo motors which are controlled by the arduino. The legs are attached to the body of the hexapod in such a way that the servo motors do not get damaged. A protoshield is included with the Arduino MEGA, which makes attaching the servos to the microcontroller much easier. Each servomotor is connected to the protoshield using 18 tri terminal connectors welded directly to the PCB. The Arduino MEGA comes with an A-B USB cable that connects it to the Raspberry Pi. A 20A UBEC is used to power the servos and the protoshield as shown in Fig.2. A Lidar is integrated with the robot for mapping purposes.

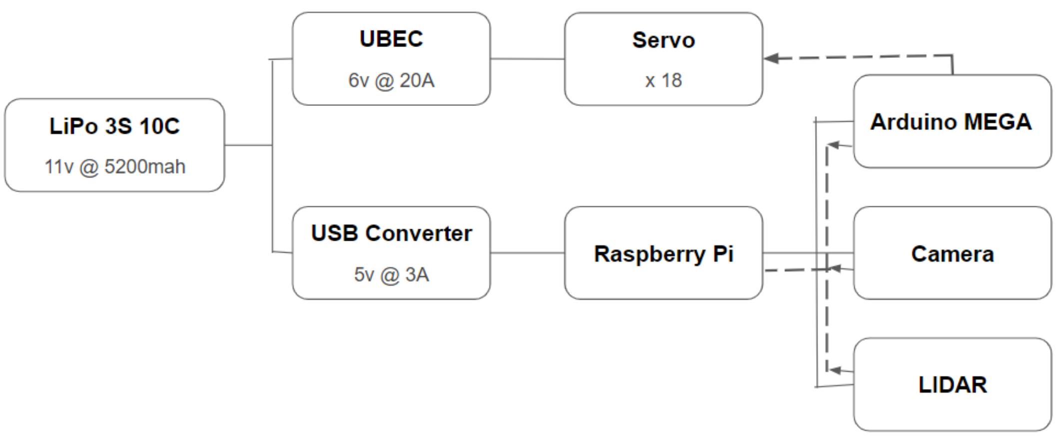

Fig. 2. Electrical system of the hexapod.

Servo calibration is done to stop the hexapod from assuming default calibration values and to reduce error. The hexapod will be in walk position by default when the calibration mode is started after the initial calibration. That suggests the angles aren't the ones used for calibration. If this option is not disabled, the associated servo will move to the desired place every time calibration is done. This mode can be disabled if the movement of the servo is not desirable while calibrating it. The 3 walking gaits, tripod gait, wave gait, ripple gait are adjusted.

Raspbian and the ROS framework are installed in the Raspberry Pi, and the OS is configured in order to remotely connect the hexapod. A 32 GB SD card is used for this purpose. After installing the Arduino IDE, the board is plugged in the Raspberry Pi and required packages are installed. The camera in the Raspberry Pi is also enabled.

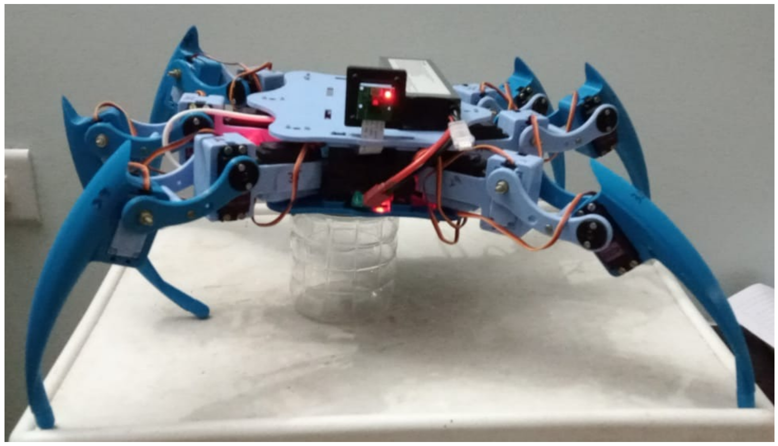

Fig. 3. Six legged hexapod when it’s turned on

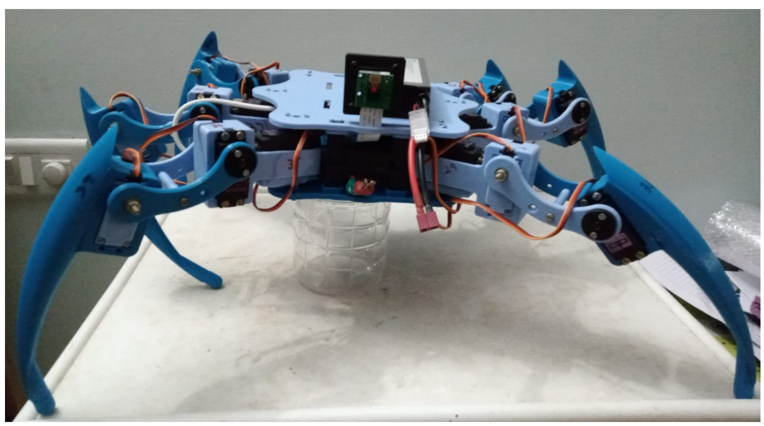

Fig. 4. Six legged hexapod when it’s turned off.

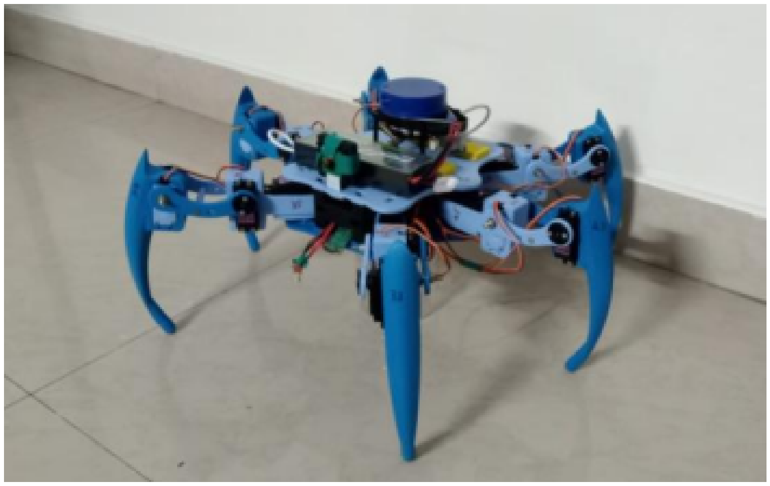

Fig.5 Fully assembled hexapod.

4.1.2. Mapping

The shortest distance between the start point and the target point is calculated using a map based planning method in which the shortest distance is chosen based on the lowest movement cost. The detailed mapping of the house is done using a LIDAR mounted on the top of the hexapod. The LIDAR sensor works based on the pulsed light waves that it emits into the environment from a laser. These pulses then enter the sensor after bouncing off from nearby objects. It is different from IR sensors and ultrasonic sensors which are generally used. This sensor works in the frequency range of 5000Hz with a maximum wide ranging range of 10 meters. The hexapod is manually operated using a joystick to navigate through the maze to draw a predefined map of the rooms as shown in Fig.5. The hexapod will then map the room to identify the obstacles in each room and a path is drawn in such a way that it avoids collision during navigation.

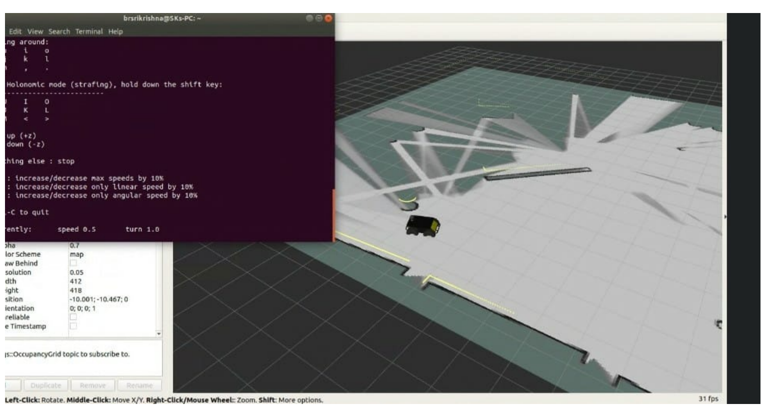

Fig. 6. Mapping the rooms by manually operating the hexapod.

4.1.3. Surveillance and live streaming

The surveillance is done using a Raspberry Pi camera module. The camera is mounted on top of the hexapod for a better Field of view. Camera module is enabled in the Raspberry Pi in order to use the camera. The camera is connected to the Pi CSI port. After setting up the camera, the video streaming web server is accessed at the Raspberry Pi's IP address to monitor the house from anywhere and at any time.

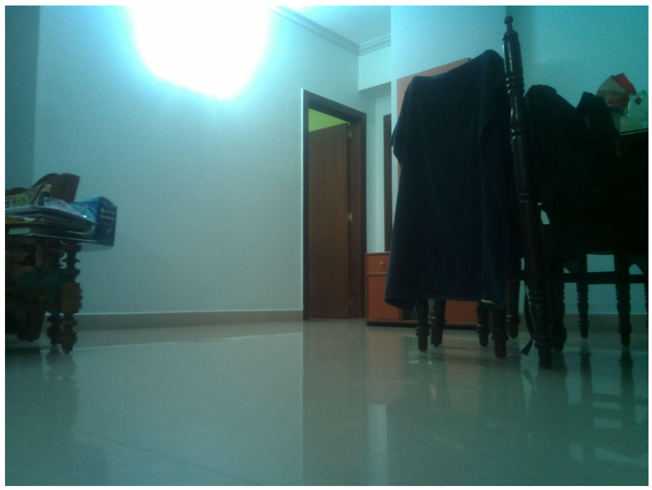

Fig. 7. Image of dining room captured by the hexapod camera

Fig.8.Image of entrance of bedroom captured by the hexapod camera

4.1.4. Voice navigation and automation

Google Assistant linked to the user's account is used for giving voice commands to the robot. A speaker is connected to the Raspberry Pi for the hexapod to respond to the voice commands. The system deals with listening to the audio and performing the actions when the triggering word is stated.

If the user says the predetermined phrase, the function to turn on/off the lights or other electronics is satisfied and the output is either turning on or off the light bulb in the room from which the voice command is given, depending on the command. The feedback is sent in the form of sounds from the speaker.

In the Adafruit platform a new dashboard named ‘Automation’ is created and two triggers are added to it. The triggers are named Light 1 and Light 2 with values 0 and 1 for switching ON and switching OFF respectively. The Fig.8 shows the dashboard with two triggers. The key is then used to link the Adafruit platform with the IFTTT platform.

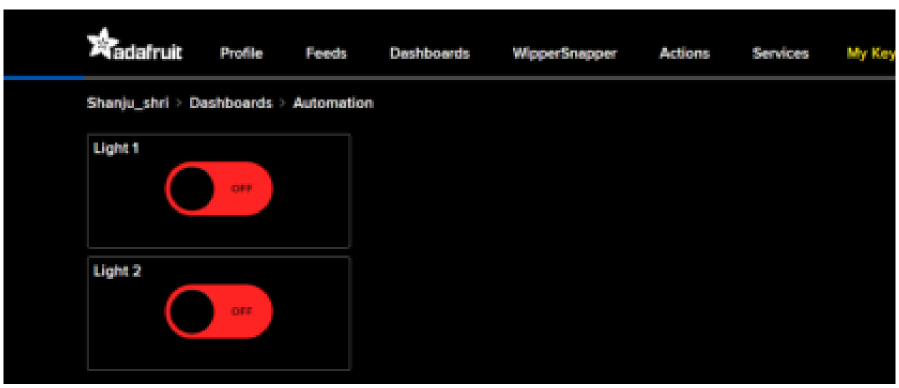

Fig. 9. Adafruit dashboard.

The NodeMCU is connected to the LEDs which represent the electronics in the rooms. The data in the form of voice is then sent to the Adafruit platform through IFTTT. This data is then used to automate the electronics connected to the system. The Fig.9 shows the applets created in the IFTTT platform which triggers the functions to turn on and off the lights based on the given commands.

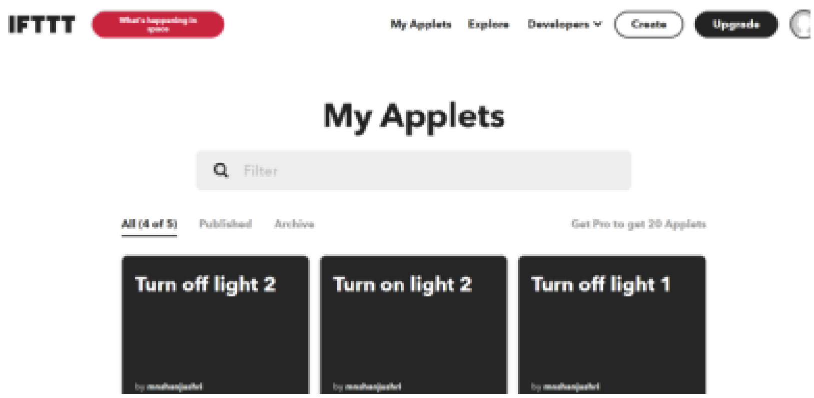

Fig. 10. Applets created in IFTTT.

4.2 Codes and Standards

4.2.1 Configuration

Networking - The best option to connect Hexapod is to create its own Wi-Fi network and connect to it. To get it working we will need to create an Ad-Hoc network.

In order to set up the network we will need to know if our driver supports it.

An EP-NG8508GS adapter was used which uses an rt8192cu driver. This driver does not work with Ad-Hoc mode and 8192cu needs to be installed.

Interface configuration

To create the network we need to change `/etc/network/interfaces`:

```
auto wlan0

iface wlan0 inet static

  address 192.168.100.1

  netmask 255.255.255.0

  wireless-channel 1

  wireless-essid Hexapod

  wireless-mode ad-hoc
```

This enables an Ad-Hoc network with ESSID Hexapod and establishes the IP 192.168.100.1 for the device.

DHCP server configuration

To get an IP automatically DHCP server needs to be configured.

To do this we need the following package:

```
sudo apt-get update

sudo apt-get install isc-dhcp-server
```

And it needs to be configured editing `/etc/dhcp/dhcpd.conf`:

```
ddns-update-style interim;

default-lease-time 600;

max-lease-time 7200;

authoritative;

log-facility local7;

subnet 192.168.100.0 netmask 255.255.255.0 {

  range 192.168.100.5 192.168.100.150;

}
```

After this, the network needs to be restarted.

To work with Raspbian:

```
sudo service networking restart; sudo service isc-dhcp-server restart
```

4.2.2 Calibration

If calibration is not done hexapod cannot be operated.

Default position

If the hexapod is started without calibrating it completely, the hexapod will write default values to servo calibration positions.

Table 4. Default calibration values

	Left	Right
Thigh	90º	90º
Femur	130º	50º
Feet	90º	90º

Starting calibration

To start a calibration Hexapod is started and it will go to the default calibration position. After that, the angles are changed as desired to send a Calibrate message:

```
Args [Leg, member, angle]
```

Table 5. Calibration

	From	To
Leg	0	5
Member	0	2
Angle	0	180

Completed calibration

If servos are all in desired neutral position just send a Calibrate message with angle = 255:

```
$ rostopic pub /antfirm/calibrate hexapod_msgs/Calibrate 0 2 255
```

Calibration will be successfully completed and now the hexapod will be able to run.

Current Calibration

To see current calibration stored in EEPROM a Calibrate message with angle = 254 needs to be sent:

```
$ rostopic pub /antfirm/calibrate hexapod_msgs/Calibrate 3 1 254
```

And calibration will be listed.

Move servos while calibrating

By default, when starting the calibration mode after the first calibration, the hexapod will be in walk position. That means that the angles are not the calibration ones. To see the real angles, a Calibrate message needs to be sent with [angle = 253].

```
$ rostopic pub /antfirm/calibrate hexapod_msgs/Calibrate 0 0 253
```

The hexapod's servos will go to the calibration angles. If this mode is not turned off, every time a new Calibration message is sent the corresponding servo will go to the desired location.

4.2.3 PS3 drivers in ROS

- Dependencies

```
$ sudo apt-get install libspnav-dev python-cwiid libusb-dev

$ cd ~/catkin_ws/src

$ git clone https://github.com/ros/diagnostics

$ cd ~/catkin_ws

$ catkin_make
```

- Source installation

```
$ cd ~/catkin_ws/src

$ git clone https://github.com/ros-drivers/joystick_drivers.git

$ cd ~/catkin_ws
```

```
$ catkin_make
```

Bluetooth settings

Insert the bluetooth USB dongle. Check if the board recognises it.

```
$ lsusb
```

To see if the dongle is working :

```
$ hciconfig
```

If the bluetooth is turned off :

```
$ sudo hciconfig hci0 up
```

```
$ sudo hcitool scan
```

4.2.4 Change gait

To change gait a Gait message needs to be sent.

Gait message form: `Args [Type, leg0, leg1, leg2, leg3, leg4, leg5]`

Table 6. Type and Sequence in changing gait

	Type	**Sequence**
Tripod	1	{0, 1, 1, 0, 0, 1}
Ripple	2	{0, 2, 3, 1, 2, 0}

Sequence explanation

Sequence is a vector that coincides its member with leg and its value means the turn in which the leg is going to move.

For example tripod gait with sequence: *{0, 1, 1, 0, 0, 1}*

Table 7. Turn to move for each leg

Leg	Turn to move
0	0
1	1
2	1
3	0
4	0
5	1

So, first move is *legs{0,3,4}* and second move involves *legs{1,2,5}*

4.2.5 Installing Raspbian on Raspberry Pi

```
$ sudo umount /dev/sdb
```

```
$ sudo dd bs=1M if=2014-12-24-wheezy-raspbian.img of=/dev/sdb
```

Remote connection with Raspi

Raspbian needs to be prepared to auto connect the wifi network and use a static IP to find it easily.

To do this SD Card needs to be introduced with raspbian on a pc.

Mount sdX2 to /mnt :

```
$ sudo mount /dev/sdb2 /mnt
```

And edit these files :

```
$ sudo nano /mnt/etc/wpa_supplicant/wpa_supplicant.conf
```

If the image is not mounted in /mnt, the file needs to be edited in

```
$ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev

update_config=1

network={

        ssid="NAME_WIFI_NETWORK"

        psk="PASSWORD"

        proto=RSN

        key_mgmt=WPA-PSK

        pairwise=CCMP

        auth_alg=OPEN

}

$ sudo nano /mnt/etc/network/interfaces

auto lo

iface lo inet loopback

iface eth0 inet dhcp

allow-hotplug wlan0

iface wlan0 inet manual

wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf

iface default inet static

   address 192.168.0.9
```

```
   netmask 255.255.255.0

   network 192.168.0.0

   gateway 192.168.0.1

   dns-nameservers 8.8.8.8 8.8.4.4
```

Now raspberry can be accessed using ssh like this:

User/password: pi/raspberry

```
$ ssh pi@192.168.0.9
```

First boot on raspberry

```
$ sudo raspi-config
```

Watchdog

Its purpose is to automatically restart RPi if it becomes unresponsive. To install it:

```
$ sudo apt-get install watchdog

$ sudo modprobe bcm2708_wdog

$ sudo su

$ echo 'bcm2708_wdog' >> /etc/modules
```

ROS indigo

Installing ROS on raspbian

Prerequisites

Setup ROS Repositories

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu wheezy main" >
/etc/apt/sources.list.d/ros-latest.list'

$ wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -O - |
```

```
sudo apt-key add -
```

Now, it needs to be made sure that the Debian package index is up-to-date:

```
$ sudo apt-get update

$ sudo apt-get upgrade
```

Install bootstrap dependencies

```
$ sudo apt-get install python-setuptools

$ sudo easy_install pip

$ sudo pip install -U rosdep rosinstall_generator wstool rosinstall
```

Initializing rosdep

```
$ sudo rosdep init

$ rosdep update
```

Create a catkin Workspace

In order to build the core packages, a catkin workspace is needed. To create one:

```
$ mkdir ~/ros_catkin_ws

$ cd ~/ros_catkin_ws
```

Next we will want to fetch the core packages so we can build them. We will use wstool for this. We will install ROS-Comm (ROS package, build, and communication libraries without GUI tools)+ Rosserial :

```
$ rosinstall_generator ros_comm diagnostics joystick_drivers robot_upstart
bond_core dynamic_reconfigure nodelet_core rosserial class_loader image_common
vision_opencv image_transport_plugins --rosdistro indigo --deps --wet-only
--exclude roslisp --tar > indigo-hexapod-wet.rosinstall

$ wstool init -j8 src indigo-hexapod-wet.rosinstall
```

Resolve Dependencies

```
$ mkdir ~/ros_catkin_ws/external_src

$ sudo apt-get install checkinstall cmake
```

libconsole-bridge-dev: Install with the following:

```
$ cd ~/ros_catkin_ws/external_src

$ sudo apt-get install libboost-system-dev libboost-thread-dev

$ git clone https://github.com/ros/console_bridge.git

$ cd console_bridge

$ cmake .

$ sudo checkinstall make install
```

liblz4-dev: Install with the following: Install with the following:

```
$ cd ~/ros_catkin_ws/external_src

$ wget
http://archive.raspbian.org/raspbian/pool/main/l/lz4/liblz4-1_0.0~r122-2_armhf.
deb

$ wget
http://archive.raspbian.org/raspbian/pool/main/l/lz4/liblz4-dev_0.0~r122-2_armh
f.deb

$ sudo dpkg -i liblz4-1_0.0~r122-2_armhf.deb liblz4-dev_0.0~r122-2_armhf.deb
```

Resolving Dependencies with rosdep

The remaining dependencies should be resolved by running rosdep:

```
$ cd ~/ros_catkin_ws

$ rosdep install --from-paths src --ignore-src --rosdistro indigo -y -r
--os=debian:wheezy
```

Invoke catkin_make_isolated:

```
$ sudo ./src/catkin/bin/catkin_make_isolated --install
-DCMAKE_BUILD_TYPE=Release --install-space /opt/ros/indigo
```

Now ROS should be installed :

```
$ echo 'source /opt/ros/indigo/setup.bash' >> ~/.bashrc
```

Then reboot or execute :

```
$ source /opt/ros/indigo/setup.bash
```

Installing Arduino-1.0.5

```
$ sudo apt-get install arduino

$ cd /tmp

$ wget http://downloads.arduino.cc/arduino-1.0.5-linux32.tgz

$ tar zxvf arduino-1.0.5-linux32.tgz

$ cd arduino-1.0.5

$ rm -rf hardware/tools

$ sudo su

$ cp -ru lib /usr/share/arduino

$ cp -ru libraries /usr/share/arduino

$ cp -ru tools /usr/share/arduino

$ cp -ru hardware /usr/share/arduino

$ cp -ru examples /usr/share/doc/arduino-core

$ cp -ru reference /usr/share/doc/arduino-core
```

Once the Arduino IDE is installed, the board can be plugged in the Raspberry.

```
$ ls -l /dev
```

Unplug the Arduino from the Raspberry and run the line again. One of the devices will no longer be available. Now, it needs to be started by listing the permissions of the Arduino. Plug the Arduino in again and choose one of the lines below according to device, where the X is the number of the device.

```
$ ls -l /dev/ttyACMX

$ ls -l /dev/ttyUSBX
```

If XX is rw: the Arduino is configured properly. If XX is --: the Arduino is not configured properly and one of the below lines needs to be executed :

```
$ sudo chmod a+rw /dev/ttyACMX

$ sudo chmod a+rw /dev/ttyUSBX
```

Create catkin_ws

To create catkin workspace:

```
$ mkdir -p ~/catkin_ws/src

$ cd ~/catkin_ws/src

$ catkin_init_workspace
```

The workspace can be built without packages :

```
$ cd ~/catkin_ws/

$ catkin_make

$ echo 'source ~/catkin_ws/devel/setup.bash' >> ~/.bashrc
```

Compiling hexapod packages

```
$ cd ~/catkin_ws/src

$ git clone https://github.com/ros-drivers/rosserial.git
```

Compiling

To compile all :

```
$ cd ~/catkin_ws
```

```
$ catkin_make
```

Uploading firmware to arduino

```
$ catkin_make hexapod_antfirm_firmware_upload
```

If Arduino board was connected to the Raspberry through the ttyUSB port, then the following lines need to be changed in the code :

```
$ sudo nano ~/catkin_ws/sr/hexapod/hexapod_antfirm/firmware/CMakeLists.txt
```

Replace the line

```
$ PORT /dev/ttyACM0
```

```
by
```

```
$ PORT /dev/ttyUSB0
```

Now run the line

```
$ sudo nano ~/catkin_ws/src/hexapod/hexapod_antfirm/launch/antfirm.launch
```

Replace the line:

```
$ <param name="~port" value="/dev/ttyACM0" />
```

```
by
```

```
$ <param name="~port" value="/dev/ttyUSB0" />
```

Finally, run:

```
$ sudo nano ~/catkin_ws/src/hexapod/hexapod_antfirm/launch/antfirm_debug.launch
```

Replace the line:

```
$ <param name="~port" value="/dev/ttyACM0" />
```

by

```
$ <param name="~port" value="/dev/ttyUSB0" />
```

Executing

```
$ roscore
```

To initialize Arduino open a new session and execute:

```
$ rosrun rosserial_python serial_node.py /dev/ttyACM0
```

4.2.6 Upgrading

If updates are available it can be checked with:

```
$ cd ~/catkin_ws/src/hexapod

$ git pull
```

If there are any changes the firmware needs to be recompiled :

```
$ cd ~/catkin_ws

$ catkin_make

$ catkin_make hexapod_antfirm_firmware_upload

$ rm -rf ~/catkin_ws/build/hexapod/hexapod_antfirm
```

4.2.7 Enable RaspiCam

First of all the cam in Raspberry Pi needs to be enabled.

To do this :

```
$ sudo raspi-config
```

Activate option -> *5: Enable Camera*

Finish and reboot raspberry.

Ros package image_common

```
$ sudo apt-get update

$ sudo apt-get install libyaml-cpp-dev libpoco-dev

$ cd ~/catkin_ws/src

$ git clone https://github.com/ros/class_loader.git

$ git clone https://github.com/ros/pluginlib

$ git clone https://github.com/ros-perception/image_common.git

$ git clone https://github.com/ros-perception/image_transport_plugins

$ cd ~/catkin_ws

$ catkin_make
```

Ros package image_transport_plugins

```
$ sudo apt-get update

$ sudo apt-get install libopencv-dev python-opencv libtheora-dev

$ cd ~/catkin_ws/src

$ git clone https://github.com/ros-perception/vision_opencv

$ git clone https://github.com/ros/dynamic_reconfigure

$ git clone https://github.com/ros/nodelet_core

$ git clone https://github.com/ros/bond_core

$ git clone https://github.com/ros-perception/image_transport_plugins

$ rm -r vision_opencv/opencv_apps  #examples of opencv usage

$ cd ~/catkin_ws

$ catkin_make
```

Adding ARM libraries for interfacing to Raspberry Pi GPU

```
$ git clone https://github.com/raspberrypi/userland.git /home/pi/userland
```

Install raspicam node

```
$ cd ~/catkin_ws/src

$ git clone https://github.com/fpasteau/raspicam_node.git raspicam

$ cd ~/catkin_ws

$ catkin_make
```

Ros robot_upstart

This package from Clearpath Robotics allows us to create a suite of scripts to assist with launching background ROS processes on Ubuntu Linux PCs.

Installing packages

Install Upstart

```
$ sudo apt-get install upstart
```

Updating the workspace

```
$ cd ~/ros_catkin_ws

$ rosinstall_generator robot_upstart --rosdistro indigo --deps --wet-only
--exclude roslisp --tar > indigo-custom_ros.rosinstall

$ wstool merge -t src indigo-custom_ros.rosinstall

$ wstool update -t src

$ rosdep install --from-paths src --ignore-src --rosdistro indigo -y -r
--os=debian:wheezy

$ sudo ./src/catkin/bin/catkin_make_isolated --install
-DCMAKE_BUILD_TYPE=Release --install-space /opt/ros/indigo
```

Rebuild catkin_ws

```
$ cd ~/catkin_ws/
```

```
$ catkin_make
```

Reboot system before using the upstart service.

Usage

To create a service with the hexapod robot the following command needs to be run :

```
$ rosrun robot_upstart install hexapod_bringup/launch --interface wlan0
```

Camera permissions

```
$ sudo su
```

```
$ echo 'SUBSYSTEM=="vchiq",GROUP="dialout",MODE="0660"' >
/etc/udev/rules.d/10-vchiq-permissions.rules
```

```
$ exit
```

After that it is necessary to change user principal group to serial privileges group with command:

```
$ sudo usermod -g dialout pi
```

4.2.8 Installation using Ubuntu

Installing ROS packages

Once ROS repositories are added, basics ROS packages needs to be installed which are ROS dependencies for Hexapod:

```
$ sudo apt-get install ros-indigo-ros-comm ros-indigo-diagnostics
ros-indigo-joystick-drivers ros-indigo-rosserial ros-indigo-rosserial-server
ros-indigo-rosserial-arduino ros-indigo-robot-upstart ros-indigo-bond-core
ros-indigo-dynamic-reconfigure ros-indigo-nodelet-core ros-indigo-class-loader
ros-indigo-image-common ros-indigo-vision-opencv
ros-indigo-image-transport-plugins
```

Environment setup

This allows us to access ROS commands:

```
$ echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
```

Additional dependencies

```
$ sudo apt-get install git arduino
```

Raspicam node

The Raspberry Pi has a lot of features and one of them is its incredible camera.

4.3 Constraints

Technical issues like mismatch of versions of OS and ROS software, and grip at the feet of the pod are sorted out by using older versions and making DIY feet for extra grip respectively. The IoT platform used is also comparatively user friendly.

5. SCHEDULE, TASKS AND MILESTONES

5.1 Project Execution stages

● The initial stage of the project started in January, so I looked for works that are necessary during these tough times . On choosing the problem statement, I gathered literature that aimed at solving similar issues in order to understand the present work and technology used, that I could build on in the work.

● Further, I surveyed the collected literature to understand the pros and cons of each proposed work and started working on the paper. I looked into various conferences that are suitable for this project.

● With this, in February 3D parts are designed and printed and then assembled to complete the hexapod. ROS was installed and calibrations of servo motors were done. Dashboard was created in Adafruit platform and integrated with IFTTT platform and Google Assistant for giving voice commands. The paper was completed and was submitted to the IEEE conference at the end of the month.

● In the beginning of April, the movement of the robot and mapping was the main point of focus. The IoT platform was built and integrated with the hexapod. The paper got approved for the notification and a conference was held. The paper was selected as the best paper in the IEEE conference and certificates were issued.

5.2 Gantt chart

Table 8. Gantt chart

Month (2022)	January				February				March				April				May			
Week and Task	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Problem statement and brainstorming																				
Literature survey																				
Design of process automation and topic discussion																				
1st Review																				
Designing 3D printed model of hexapod																				
Assembled the 3D model and worked with servo motor interface																				
Review 2																				
Worked in robot movement and integrating NODE-MCU																				
Developed a applet in IFTTT and attended ICDCECE conference for publication																				
Integration of module																				
Results and Conclusion																				
IEEE paper, Poster work, Report																				
Review 3																				

6. PROJECT DEMONSTRATION

Fig. 11 - 13. Hexapod in motion after voice command is given.

Fig. 14 Image captured by hexapod's camera.

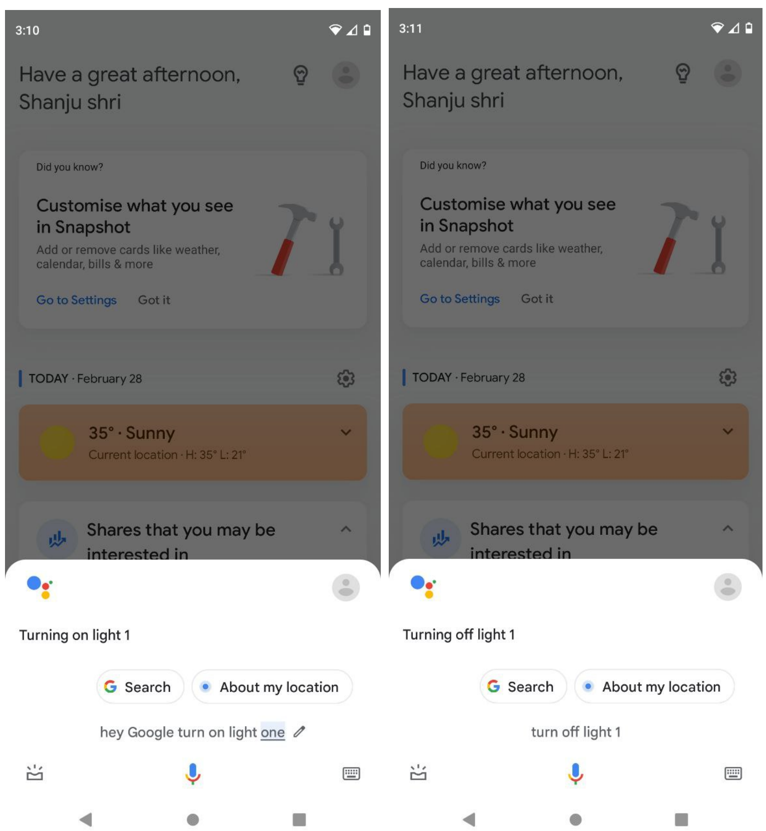

Fig. 15-16 Voice commands given to turn on and turn off light 1.

7. RESULT & DISCUSSION

The hexapod functions successfully and expected results are obtained. The hexapod reaches the target destination to work on the task given as command. The hexapod moves easily through rough terrains in a stable way even when it's carrying medicines as shown in Fig. 13. Thus, it is able to carry things without them falling off the hexapod.

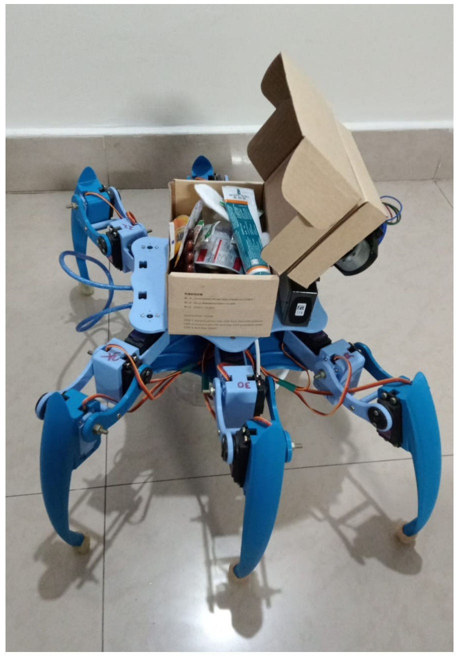

Fig.17 Hexapod carrying medicines to the user.

The camera in the hexapod captures the images and video is streamed in the website. The following image Fig.14 represents the screenshot of the live streaming page. Thus the user can monitor the house from anywhere.

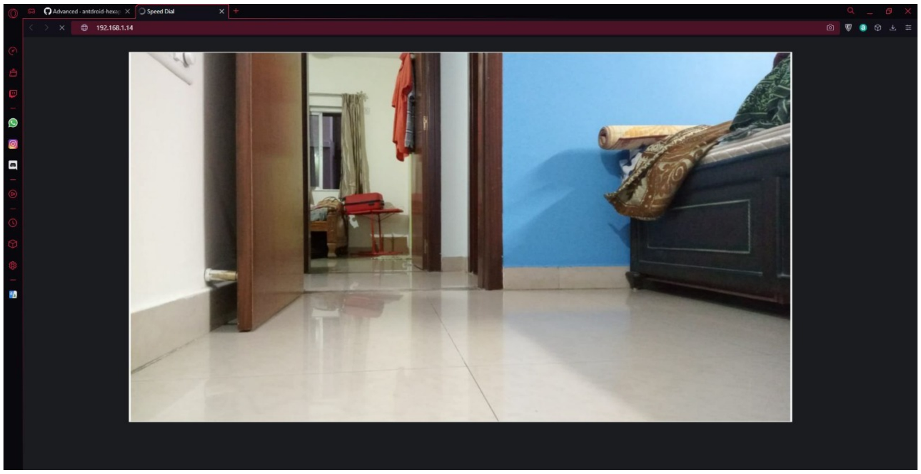

Fig.18 Live streaming

The voice commands are given using the Google Assistant to turn on the lights in the room. Google Assistant passes the commands to the system and responds to the given commands as shown in Fig.15.

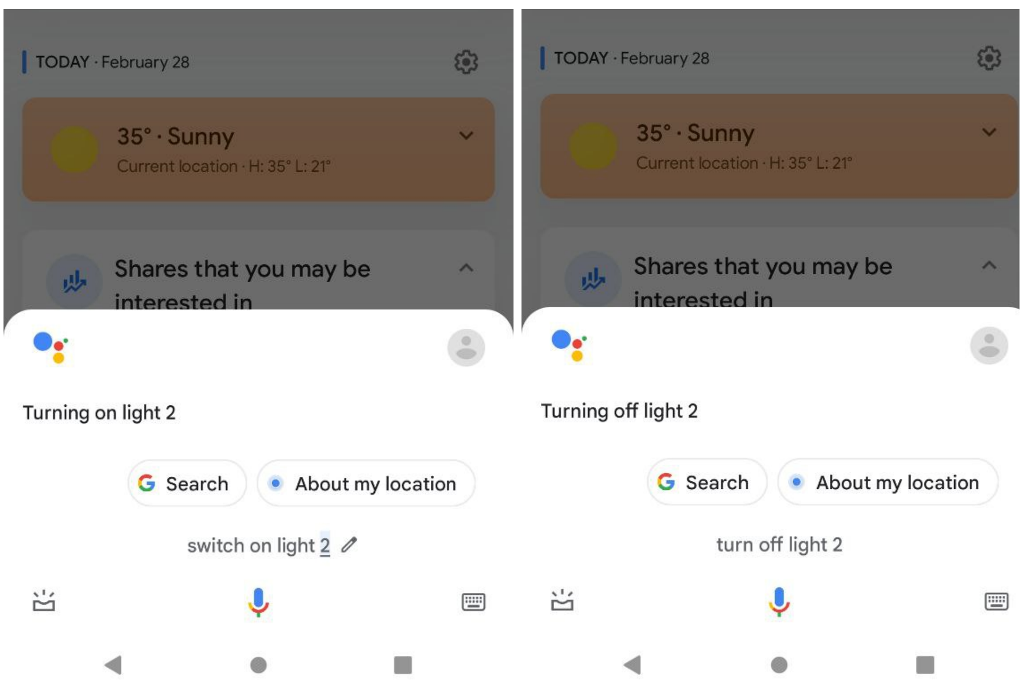

Fig. 19-20. Voice commands given to Google Assistant to turn on and off light 2.

LEDs represent the electronics in the rooms. When the user commands the LEDs to turn ON and OFF, the tasks are performed accordingly. Fig.16 and Fig.17 represent when light 1 is turned ON while light 2 is turned OFF and both the lights are turned ON respectively.

Fig. 21. When one of the LEDs is turned ON.

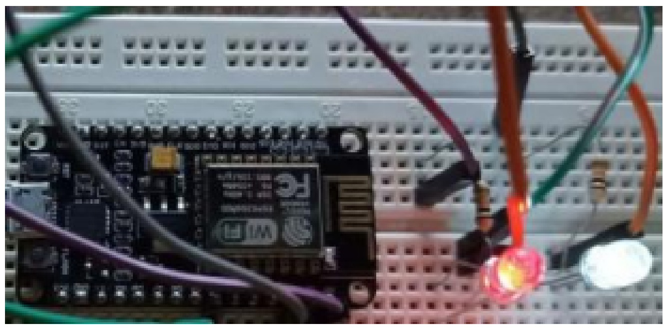

Fig. 22. When both the LEDs are turned ON.

Home Assistance Hexapod shows some similarities in developing and controlling the robot, but the features and application are vastly different from the previous works. This robot is specifically built to act as an assistant for people at home with features like carrying things, automating electronics and surveillance. Unlike the sensors used in other works, Lidar has a large measurement range and the accuracy is also very good.

8. SUMMARY

8.1 Conclusion

The home assistance hexapod acts as a perfect companion and assistance for people at home, especially for sick and elderly people to meet their needs. The main objective of this system is achieved by integrating the Internet of Things with robotics successfully. The camera attached to the robot works effectively for monitoring the home even when the user is not at the site. Voice controlled automation proves to make the hexapod more user-friendly.

8.2 Scope for future work

This work can be further improved by integrating it with more sensors like moisture sensor, temperature sensor, gas sensor to detect the moisture level in the soil to notify the user regarding watering plants, to automate turning on/off air coolers, and to notify users in case there is a gas leak. Machine learning algorithms can also be applied on the captured videos to identify faces and allow people inside the house.

9. REFERENCES

[1] Richik Ray, Rishita Shanker et al. "Automated Quadruped Robot Simulation using Internet of Things and MATLAB", *Journal of Physics: Conference Series*, 2021.

[2] Mostafa Khazaee, Majid Sadedel and Atoosa Davarpanah, "Behavior-Based Navigation of an Autonomous Hexapod Robot Using a Hybrid Automaton", *Journal of Intelligent & Robotic Systems,* May 2021.

[3] Abdelrahman Sayed Sayed, Hossam Hassan Ammar and Rafaat Shalaby, "Centralized Multi-agent Mobile Robots SLAM and Navigation for COVID-19 Field Hospitals", *2nd Novel Intelligent and Leading Emerging Sciences Conference*, October 2020.

[4] Md. Imran Uddin, Md. Shahriar Alamgir et al. "Multitasking Spider Hexapod Robot", *IEEE International Conference on Robotics, Automation, Artificial Intelligence and Internet-of-Things (RAAICON)*, 2019.

[5] Fernando Gomez-Bravo, Pablo Villadoniga, and Giuseppe Carbone, "Design and Operation of a Novel Hexapod Robot for Surveillance Tasks", *Advances in Service and Industrial Robotics*, 2018.

[6] Mahesh Pawaskar, Ankit Khandeparkar, Adesh C Hardas, Sonia Aneesh, "Surveillance and Controlling of Firebird-V Based Hexapod using Raspberry Pi", *IEEE International Conference for Innovation in Technology (INOCON)*, 2020.

[7] Vaibhav Malhotra, "AUTONOMOUS LUGGAGE CARRIER HEXAPOD", *International Journal of Engineering Applied Sciences and Technology*, Volume 5, Issue 7, 2020.

[8] G. Anandravisekar et al. "IOT Based Surveillance Robot ", *International Journal of Engineering Research & Technology (IJERT)*, Vol. 7 Issue 03, March 2018.

[9] Abhishek C et al. "Development of Hexapod Robot with Computer Vision", *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, Vol. 9, Volume 8, August 2021.

[10] Anita Chaudhari, Jeet Thakur and Pratiksha Mhatre, "PROTOTYPE FOR QUADRUPED ROBOT USING IoT TO DELIVER MEDICINES AND ESSENTIALS TO COVID-19 PATIENT ", *International Journal of Advanced Research in Engineering and Technology (IJARET)*, Vol. 12, Issue 5, May 2021.

[11] Marko Bjelonic , Timon Homberger et al. "Autonomous Navigation of Hexapod Robots With Vision-based Controller Adaptation", *International Conference on Robotics and Automation*, 2017.

[12] Srividya C. R, Shreya Adiga H, et al. "AI-based hexapod military robot", *International Journal of Advance Research, Ideas and Innovations in Technology*, Volume 7, Issue 4, 2021.

[13] Z. Iklima, T M Kadarina, "Distributed Path Planning Classification with Web-based 3D Visualization using Deep Neural Network for Internet of Robotic Things", *Journal of Science and Technology*", Vol 13, No 2, 2021.

[14] Joana Coelho, Fernando Ribeiro, Bruno Dias, Gil Lopes and Paulo Flores, "Trends in the Control of Hexapod Robots: A Survey", *Robotics*, 2021.

[15] Pieter Simoens, Mauro Dragone and Alessandro Saffiotti, "The Internet of Robotic Things: A review of concept, added value and applications" , *International Journal of Advanced Robotic Systems,* 2018.

[16] Mindaugas Luneckas, Tomas Luneckas et al. "A hybrid tactile sensor-based obstacle overcoming method for hexapod walking robots", *Intelligent Service Robotics*, October 2020.

[17] Mohit S. Mhaske, Shivam S. Raut, Harshali Chavan, "An IoT based Human Detection Quadruped Bot for Rescue", *International Research Journal of Engineering and Technology (IRJET)*, Vol. 7 Issue 04, April 2020.

[18] Uma S , R Eswari et al. "IoT based Voice/Text Controlled Home Appliances", International Conference on Recent Trends in Advanced Computing (ICRTAC), 2019.

[19] Feng Zhang, Shidong Zhang, Qian Wang , Yujie Yang and Bo Jin, "Straight Gait Research of a Small Electric Hexapod Robot", Applied Sciences, 2021.

[20] Wei Cheah et al. "Grid-based Motion Planning Using Advanced Motions for Hexapod Robots", International Conference on Intelligent Robots and Systems (IROS), 2018.

[21] Liang Ding, et al. "Footstep Planning for Hexapod Robots Based on 3D Quasi-static Equilibrium Support Region", Journal of Intelligent and Robotic Systems, September 2021.

[22] Marek Zak, and Jaroslav Rozman, "Design, construction and control of hexapod walking robot", IEEE 13th International Scientific Conference on Informatics, 2015.

[23] Jing Qi, et al. "Fusing Hand Postures and Speech Recognition for Tasks Performed by an Integrated Leg-Arm Hexapod Robot", Applied Sciences, 2020.

[24] Svitlana Maksymova et al. "Software for Voice Control Robot: Example of Implementation", Open Access Library Journal, Volume 4, No.8, August 2017.

[25] Abhilash Krishna et al. "Design and Fabrication of a Hexapod", International Conference on Embedded Systems (ICES), 2014.

[26] Aqeel-ur-Rehman, Royda Arif and Hira Khursheed, "Voice Controlled Home Automation System for the Elderly or Disabled People", Journal of Applied Environmental and Biological Sciences, 2014.

[27] Pankaj Bhardwaj, et al. “a review paper on smart home automation”, International Journal of Scientific Research and Management Studies (IJSRMS), Volume 3 Issue 7.

[28] Ribil Mary Roy et al. “Voice Controlled Home Automation System”, Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII), 2021.

[29] Ling Fang and Feng Gao, “Type Design and Behavior Control for Six Legged Robots”, Chinese Journal of Mechanical Engineering, July 2018.