**Project Report**

**on**

**Hexapod Spider Robot**

***Submitted in partial fulfillment of the***

***Requirements for the award of the degree***

***of***

BACHELORS OF ENGINEERING

in

MECHATRONICS ENGINEERING

**Submitted By: - Faculty Supervisor: -**

PANKAJ KUMAR (19BEM1010) Mr. INDERPREET SINGH  
SHIVESH PANDEY (19BEM1055)  
JASMEET SINGH (19BEM1018)  
MUKUL KUMAR (19BEM1039)  
HARSHUL AGGARWAL (19BEM1013)

**UNIVERSITY INSTITUTE OF ENGINEERING**

**CHANDIGARH UNIVERSITY**

**GHARUAN, MOHALI, PUNJAB, INDIA-140413**

**2021**

|  |  |  |
| --- | --- | --- |
| **INDEX** | | |
| **Sr. No** | **Content** | **Pg. No** |
| 1 | ABSTRACT | 4 |
| 2 | INTRODUCTION | 4 - 5 |
| 3 | OBJECTIVE | 5 |
| 4 | LITERATURE REVIEW | 6 |
| 5 | OVERVIEW  · Working Procedure  · Architecture | 6 - 27 |
| 6 | THEORETICAL CALCULATION | 28 |
| 7 | REQUIRED ELEMENTS | 29 – 31 |
| 8 | EQUIPMENTS / TOOLS NEEDED TO COMPLETE THIS PROJECT | 31 |
| 9 | Approximate Cost of project | 31 |
| 10 | Work plan and work distribution | 32 – 33 |
| 11 | PENDING WORK | 33 |
| 12 | CONCLUSION | 33 – 34 |
| 13 | ACKNOWLEDGEMENT | 34 |
| 14 | REFERENCE | 35 – 36 |

**ABSTRACT:**

This project (Hexapod Spider Robot) is mainly designed for military purpose uses this robot can go to different trains where many other vehicles and humans can't go, to reduce the number of casualties military persons they can send this robot first to scout the area and potentially finding the information they are looking for the same goes for any other organization that want to use it in remote areas due to its spider type design it can easily make his way through main difficult paths where normal wheel robots can't go.

**INTRODUCTION:**

Robots can be found all over. One of the most significant piece of a robot is its skeleton. There are a few essential suspension types: wheeled, followed and legged chassises. Wheeled chassises are quick, yet not appropriate for unpleasant landscape. Followed chassises are more slow, however more reasonable to rough landscape. Legged chassises are very lethargic and more hard to control, yet incredibly vigorous in harsh landscape. Legged chassises are fit to cross enormous openings and can work even in the wake of losing a leg. Many investigates were acted in this field in beyond couple of years, due to its enormous potential. Legged chassises are particularly great for space missions. There are likewise a few activities in military research. I expect to make a modest legged stage, which would permit examination and testing of strolling chassises. Make a framework with numerous sensors that permits the undercarriage any development or conduct. The robot ought to be driven from remotely associated PC and ought to send all accessible information from sensors, which will be shown on the PC in the UI program. This stage ought to be widespread, anybody could interface with the robot and drive it and anybody may associate and send his own information to the UI program of the control PC.

**OBJECTIVE:**

This new robot is prepared to do similar developments like business renditions and attempts to eliminate their negatives. Robot is made of plastic profiles via 3d designing the whole model due to their simple accessibility and adequate design we are able to operate the robot very preciously. This robot is mainly designed for military purpose uses this robot can go to different trains where many other vehicles and humans can't go, to reduce the number of casualties military persons they can send this robot first to scout the area and potentially finding the information they are looking for the same goes for any other organization that want to use it in remote areas due to its spider type design it can easily make his way through main difficult paths where normal wheel robots can't go. In contrast to business forms, this hexapod has likewise a wide assortment of sensors and equipment, Infrared sensor, LCD show, encoders. All the data from the sensors are shipped off the PC and showed in the UI program. It is additionally conceivable to utilize a library, which permits perusing of the multitude of information from the robot and sending orders to it. This robot can walk utilizing stand, wave or ripple step and is additionally fit for turn. Every leg is outfitted with a power touchy servomotor to determine joint's present position. IR can recognize impediments to keep away from impacts.

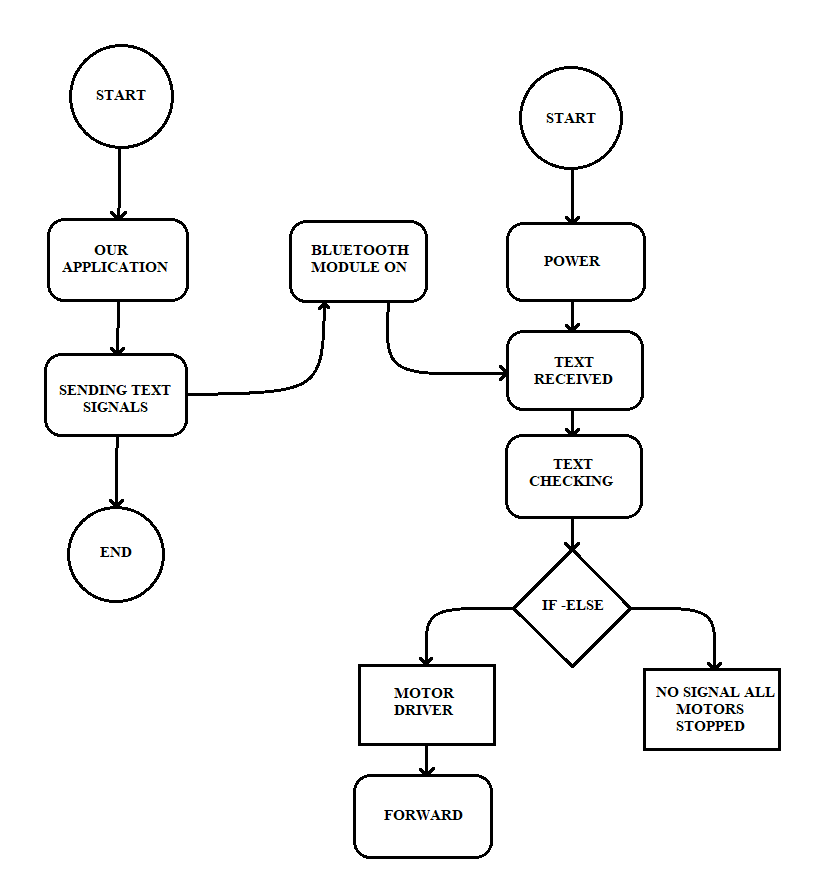
**LITERATURE REVIEW:**

There are a few organizations, which are delivering hexapod robot models. Name Lynxmotion or Trossen Robotics. The two organizations offer an assortment of leisure activity and exploration level robot packs and parts. They additionally offer a few kinds of hexapods. These Hexapod. This robot was planned and built during the task. It is furnished with sonars, camera, LCD show and more frill. robots contrast in the body shape and leg development. All robots accompany programming, which gives control of servomotors utilizing backwards kinematics and making custom steps. Robot units are sold for about $ 1,000, contingent upon the variant. Albeit a few arrangements as of now exist and have extraordinary potential, every last one of them has some weakness. The first is value, which is very high, about $ 1,000 a piece. One more burden is gear ofthe robots. The majority of the accessible robots have restricted development choices, such as missing foot sensors, which are hard to introduce later, or servomotor type with inadequate power or elements. Likewise, the batteries are frequently underlying the body and it is troublesome or even inconceivable to eliminate them. As a result of these weaknesses, we chose to fabricate a robot of our own.

**OVERVIEW:**

**WORKING PROCEDURE:**

As described earlier in this paper, this robot is designed used in military operations collect information and scout the areas where normal wheel robots can’t go due to different terrains and this robot can also be used for many purposes because of its unique spider design and the mobility it provides is far better than conventional bots. it can also be used for research purposes where we can't risk any life form, using this robot will be way better in these kind of operations. our programming did not limit the upgrades and modifications this robot, the Block Diagram of this project is given below.



Block Diagram

**CODING**

// Include the Servo library

#include <Servo.h>

// IR sensor

#define LS 9 // left sensor

#define RS 12 // right sensor

// Ultrasonic sensor pins

const int trigPin = 8;

const int echoPin = 10;

// Declare the Servo pin

int servoPin1 = 0;

int servoPin2 = 1;

int servoPin3 = 2;

int servoPin4 = 3;

int servoPin5 = 4;

int servoPin6 = 5;

int servoPin7 = 6;

int servoPin8 = 7;

int servoPinu = 11;

int flag;

int cnt;

Servo Servo1,Servo2,Servo3,Servo4,Servo5,Servo6,Servo7,Servo8,Servou;

void setup() {

pinMode(LS, INPUT);

pinMode(RS, INPUT);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

Servo1.attach(servoPin1);

Servo2.attach(servoPin3);

Servo3.attach(servoPin5);

Servo4.attach(servoPin7);

Servo5.attach(servoPin2);

Servo6.attach(servoPin4);

Servo7.attach(servoPin6);

Servo8.attach(servoPin8);

flag=0;

cnt=1;

}

void loop(){

long duration, inches, cm;

if(flag==0)//moving Leg A backwards for first time alone

{

Servo5.write(50);

delay(50);

Servo1.write(150);

delay(100);

Servo5.write(90);

delay(50);

}

// The sensor is triggered by a HIGH pulse of 10 or more microseconds.

// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:

pinMode(trigPin, OUTPUT);

digitalWrite(trigPin, LOW);

delayMicroseconds(10000);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10000);

digitalWrite(trigPin, LOW);

// Read the signal from the sensor: a HIGH pulse whose

// duration is the time (in microseconds) from the sending

// of the ping to the reception of its echo off of an object.

pinMode(echoPin, INPUT);

duration = pulseIn(echoPin, HIGH);

// convert the time into a distance

inches = microsecondsToInches(duration);

cm = microsecondsToCentimeters(duration);

if(cm>=26)//when obstacle is not present

{

if(!digitalRead(LS) & !digitalRead(RS) ) // Move Forward

{

Servo6.write(50);

delay(50);

Servo2.write(30);

delay(100);

Servo6.write(90);

delay(50);

Servo5.write(50);

delay(50);

Servo1.write(90);

delay(100);

Servo5.write(90);

delay(50);

//main

Servo6.write(120);

delay(50);

Servo8.write(120);

delay(50);

Servo4.write(60);

delay(100);

Servo2.write(60);

delay(100);

Servo6.write(90);

delay(50);

Servo8.write(90);

delay(50);

Servo7.write(50);

delay(50);

Servo3.write(150);

delay(100);

Servo7.write(90);

delay(50);

Servo8.write(50);

delay(50);

Servo4.write(90);

delay(100);

Servo8.write(90);

delay(50);

//main

Servo7.write(120);

delay(50);

Servo5.write(120);

delay(50);

Servo1.write(120);//changed

delay(100);

Servo3.write(60);

delay(100);

Servo7.write(90);

delay(50);

Servo5.write(90);

delay(50);

flag=1;

}

//when quadruped deviates the line in left side, then it should move rightwards

else if(!digitalRead(LS) & digitalRead(RS))//right

{

Servo6.write(50);

delay(50);

Servo2.write(30);

delay(100);

Servo6.write(90);

delay(50);

Servo5.write(50);

delay(50);

Servo1.write(90);

delay(100);

Servo5.write(90);

delay(50);

//main

Servo6.write(120);

delay(50);

Servo8.write(120);

delay(50);

Servo4.write(30);

delay(100);

Servo2.write(60);

delay(100);

Servo6.write(90);

delay(50);

Servo8.write(90);

delay(50);

Servo7.write(50);

delay(50);

Servo3.write(150);

delay(100);

Servo7.write(90);

delay(50);

Servo8.write(50);

delay(50);

Servo4.write(90);

delay(100);

Servo8.write(90);

delay(50);

//main

Servo7.write(120);

delay(50);

Servo5.write(120);

delay(50);

Servo3.write(60);

delay(100);

Servo1.write(120);

delay(100);

Servo7.write(90);

delay(50);

Servo5.write(90);

delay(50);

flag=1;

}

//when quadruped deviates the line in right side, then it should move leftwards

else if(digitalRead(LS) & !digitalRead(RS))//left

{

Servo6.write(50);

delay(50);

Servo2.write(30);

delay(100);

Servo6.write(90);

delay(50);

Servo5.write(50);

delay(50);

Servo1.write(90);

delay(100);

Servo5.write(90);

delay(50);

//main

Servo6.write(120);

delay(50);

Servo8.write(120);

delay(50);

Servo4.write(30);

delay(100);

Servo2.write(90);

delay(100);

Servo6.write(90);

delay(50);

Servo8.write(90);

delay(50);

Servo7.write(50);

delay(50);

Servo3.write(150);//changed

delay(100);

Servo7.write(90);

delay(50);

Servo8.write(50);

delay(50);

Servo4.write(90);

delay(100);

Servo8.write(90);

delay(50);

//main

Servo7.write(120);

delay(50);

Servo5.write(120);

delay(50);

Servo1.write(120);

delay(100);

Servo3.write(60);

delay(100);

Servo7.write(90);

delay(50);

Servo5.write(90);

delay(50);

flag=1;

}

}

//obstacle is detected, so taking complete right turn

else

{

// just turning the head right side and again to usual position

Servou.write(30);

delay(100);

Servou.write(150);

delay(100);

Servou.write(90);

delay(500);

if(digitalRead(LS) & !digitalRead(RS)) // if left sensor is outside black line, repeat right turn 4 times so that after turning it will be in correct position in line to walk

{

makeRight(4);

}

else if (!digitalRead(LS) & !digitalRead(RS))

{

makeRight(3); // if both sensor are in black line, repeat right turn 3 times

}

else

{

makeRight(2); // if right sensor is outside black line, repeat right turn 2 times

}

}

}

void makeRight(int j) //to turn when there is an obstacle

{

while(j>=0)

{

j--;

Servo6.write(50);

delay(100);

Servo2.write(30);

delay(100);

Servo6.write(90);

delay(100);

Servo5.write(60);

delay(100);

Servo1.write(90);

delay(100);

Servo5.write(90);

delay(100);

Servo6.write(120);

delay(100);

Servo8.write(120);

delay(100);

Servo4.write(130);

delay(100);

Servo2.write(130);

delay(200);

Servo6.write(90);

delay(100);

Servo8.write(90);

delay(100);

Servo7.write(50);

delay(100);

Servo3.write(130);

delay(100);

Servo7.write(90);

delay(100);

Servo8.write(50);

delay(100);

Servo4.write(90);

delay(100);

Servo8.write(90);

delay(100);

Servo7.write(120);

delay(100);

Servo5.write(130);

delay(100);

Servo1.write(130);

delay(100);

Servo3.write(60);

delay(200);

Servo7.write(90);

delay(100);

Servo5.write(90);

delay(100);

}

}

long microsecondsToInches(long microseconds)

{

// According to Parallax's datasheet for the PING))), there are

// 73.746 microseconds per inch (i.e. sound travels at 1130 feet per

// second). This gives the distance travelled by the ping, outbound

// and return, so we divide by 2 to get the distance of the obstacle.

// See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf

return microseconds / 74 / 2;

}

long microsecondsToCentimeters(long microseconds)

{

// The speed of sound is 340 m/s or 29 microseconds per centimeter.

// The ping travels out and back, so to find the distance of the

// object we take half of the distance travelled.

return microseconds / 29 / 2;

}

**ARCHITECTURE AND CONCEPT DESIGN 3D MODEL:**



**THEORETICAL CALCULATION:**

We are using 18 MG996R 180° Servo motors

Motor Specs:

4.8V-7.2V

500mA-900mA (6 V)

Stall Current 2. A

To Calculate 2C LIPO 3.7 V Battery Run Time at Full Load

7.2V x 900mA = 6.48W

For 8 Motors:

6.48 x 8 = 51.84W (52W Approx)

Battery Run Time:

7.4 V, 2000 mAH

52W/7.4V = 7.027 A

2 AH/7.027 A = 0.2846 H

17.07 Min

70% = 11.9 Min

So with 8 servo motors running at full load our battery will last approximately 12 minutes and we use that information to test the calculations, robot testing and further battery calculations.

**REQUIRED ELEMENTS AND PARTS LIST:**

In this project we used 3D printing builder the physical body of the robot. we have used 18 MG996R Servo Motors with Arduino Mega and motor driver shield so all the Motors will stay in control. The robot has six legs in total and each leg consists of three Servo Motors. The Motors are powered by 11.1 v 2C LIPO batteries. we have also used Bluetooth module to communicate with the robot through our Android application. when we press any button from our application it will send the signal to the robot via Bluetooth and the Arduino I will move the robot according to the programming. An IR sensor is also there to detect the objects and motion gestures.

1. **Arduino:** We used Arduino Mega for this robot to control all the servo motors individually. The Arduino Mega board is based on the ATmega2560 microcontroller.
2. **Shields:** We used 2 Shield boards to manage the high number of servo motors. That also helps to provide a different power source to the motors.
3. **Power distribution module:** Due to high number of servo motors we used a power distribution module.
4. **Motors:** We used MG996R 180° servo motors as they are perfect for this robot. We used 18 servo motors in total 3 for each leg.
5. **Lithium-Polymer Battery:** We used a lithium-polymer battery to power the robot. It was a 2C battery that provides 7.4 V and 2.2 AH of current.
6. **Bluetooth module:** Bluetooth module is utilized for correspondence among robot and regulator. It gives remote sequential correspondence. It is additionally utilized for ace slave correspondence.
7. **CONNECTING WIRES:** Wires are a vital piece of each framework and they are utilized for electrical connections between different electronic parts.

**EQUIPMENT / TOOLS NEEDED TO COMPLETE THIS PROJECT**

1) Soldering Iron

-Soldering Flux

-Soldering Wire

2.) 3D Printer for making body

3.) Super Glue

4.) Other Miscellaneous materials like nuts, bolts and screws

**Approximate Cost of project:**

Arduino Mega 1000

Shield 2 X 400 = 800

Power Module 150

MG996R Servo Motors 18 X 250 = 4500

3D Printing 5000

Battery 600

Bluetooth Module 250

Wires 100

**TOTAL COST Rs.12400**

**Work plan and work distribution:**

Pankaj Kumar:

* Coding
* Program Development
* Idea generation
* Feature enhancement
* Physical assembly
* PPT formation
* Research Paper

Shivesh Pandey:

* OS Design
* Application Development
* Idea generation
* Feature enhancement
* Physical assembly
* PPT formation
* Research Paper

Jasmeet Singh:

* 3D Modeling
* Design & Build
* Soldering
* Idea generation
* Feature enhancement
* Project Report
* Project Synopsis
* PPT modification
* Research Paper

Mukul Kumar:

* Kinematics Calculations
* Name generation
* Feature enhancement
* Project Report
* Project Synopsis
* PPT modification

Harshul Aggarwal:

* Idea generation
* Name generation
* Design & Build
* Feature enhancement
* Physical assembly
* PPT modification

**PENDING WORK and FUTURE SCOPE:**

Obviously, the entire world has confronted the Coronavirus circumstance. So it required some investment for us to meet and give the project its last touch. Some colleagues who are close by the college attempted to do the work on the actual model and others collaborated through web-based mode. Due to the Coronavirus circumstance, in the future we can add different sensors that will help to measure various things that can eventually help the robot to be more accurate and precise, and we can also implement a camera that will provide vision and information to the operator.

**CONCLUSION:**

This robot is mainly designed for military purpose uses this robot can go to different trains where many other vehicles and humans can't go, to reduce the number of casualties military persons they can send this robot first to scout the area and potentially finding the information they are looking for the same goes for any other organization that want to use it in remote areas due to its spider type design it can easily make his way through main difficult paths where normal wheel robots can't go.

**ACKNOWLEDGEMENT**

It gives us proud privilege to complete this project work. This is the only page where I have the opportunity to express my emotions and gratitude from the core of my heart.

It gives us great pleasure in expressing sincere and deep gratitude towards my guide Mr Inderpreet Singh for his valuable and firm suggestions, guidance and constant support throughout the completion of project named “Hexapod Spider Robot”. I am thankful to Chandigarh University for providing me various resources infrastructure facilities.

I also offer my most sincere thanks to my team members and staff members of Mechatronics Department, University Institute of Engineering, Chandigarh University for cooperation provided by them in many ways.

Pankaj Kumar

Shivesh Pandey

Jasmeet Singh

Mukul Kumar

Harshul Aggarwal

(Group 4, student BE. Mechatronics)

**REFERENCES:**

[1] Uluc Saranlı. Dynamic locomotion with a hexapod robot. PhD thesis, The University of Michigan, 2002.

[2] NASA. All-terrain hex-limbed extra-terrestrial explorer. http: //athlete.jpl.nasa.gov/, 2009.

[3] Franco Tedeschi and Giuseppe Carbone. Design issues for hexapod walking robots. Robotics, 3(2):181–206, 2014.

[4] Boston Dynamics. Boston dynamics: Dedicated to the science and art of how things move. http://www.bostondynamics.com/ robot petman.html, 2015.

[5] Boston Dynamics. Boston dynamics: Dedicated to the science and art of how things move. http://www.bostondynamics.com/robot bigdog.html, 2015.

[6] Boston Dynamics. Boston dynamics: Dedicated to the science and art of how things move. http://www.bostondynamics.com/robot ls3.html, 2015.

[7] Wikipedia. Microcontroller — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Microcontroller&oldid=654251428, 2015.

[8] Lynxmotion. Lynxmotion robot kits. http://www.lynxmotion.com/, 2015.

[9] Trossen Robotics. Trossen robotics. http://www.trossenrobotics.com/, 2015.

[10] EZ Moore and M Buehler. Stable stair climbing in a simple hexapod robot. Technical report, DTIC Document, 2001.

[11] Xilun Ding, Alberto Rovetta, JM Zhu, and Zhiying Wang. Locomotion analysis of hexapod robot. INTECH Open Access Publisher, 2010.

[12] S. Manoiu-Olaru, M. Nitulescu, and V. Stoian. Hexapod robot. mathematical support for modeling and control. In System Theory, Control, and Computing (ICSTCC), 2011 15th International Conference on, pages 1–6, Oct 2011.

[13] Roland Siegwart. Introduction to autonomous mobile robots. MIT Press, Cambridge, Mass, 2004.

[14] Raspberry Pi Foundation. Respberry pi. http: //www.raspberrypi.org/, 2014.

[15] Atmel. Atmega2560. http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-MicrocontrollerATmega640-1280-1281-2560-2561 atasheet.pdf, 2015.