QUADRUPED MODEL USING ARDUINO AND SERVO MOTORS

A Major Project submitted in partial fulfillment
Of the requirements for the degree of

Bachelor of Engineering
(Electronics and Communication)

Under RGPV, Bhopal

Submitted by

Varun Burde (0875EC131111)

Under the Guidance
Of

Prof. Megha Motta

Department of Electronics and Communication
ACROPOLIS TECHNICAL CAMPUS, Indore, (M.P.)

2017

ACROPOLIS TECHNICAL CAMPUS, INDORE

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the below mentioned students have submitted a satisfactorily completed project work entitled as ***“Quadruped Model Using Arduino And Servo Motors”*** along with this report to the department of Electronics and Communication Engineering as a part of their degree program, Bachelor of Engineering in Electronics and Communication Engineering of **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, (M.P.).**

Varun Burde **(0875EC131111)**

Project Guide

Prof. Megha Motta

Project Coordinator

Mr. Nilesh Dubey

Head of Department

Dr. Rajesh Arya

DECLARATION

We declare that the project ***"Quadruped Model Using Arduino And Servo Motors"*** is our own work carried out under the supervision of Prof. Megha Motta, Electronics and Communication Engineering Department at ACROPOLIS TECHNICAL CAMPUS, Indore(M.P).

We further declare that the work submitted is our contribution and to the best of our knowledge, this report does not contain any part of work done by any other person/team without citation/acknowledgement.

Date: **Varun Burde** **(0875EC131111)**

ACKNOWLEDGEMENT

We are thankful to our institute Acropolis Technical Campus, Indore for providing us the opportunity to apply and convert our theoretical knowledge into practical skills through this project work.

Such work requires inputs, efforts and encouragement from all walks of life. We are indeed fortunate to get active and kind co-operation from many corners without which this endeavors wouldn’t have been a success.

We express our immense gratitude to our project supervisor Prof. Megha Motta, for his valuable guidance and support and his constantly coordinating with us that made us to complete this project to meet our objectives.

We also acknowledge the kind support rendered by Prof. Nilesh Dubey towards conducting the experimental studies and compilation of the project work.

At the same time, we also would like to thank Prof. Parag Parandkar for reviewing the dissertation of the project work and suggested corrective measures.

A special vote of thanks to our Head of the Department, Dr. Rajesh Arya for his constant vigilance, support and follow up on this project which made us to work on our toes during the course of this project.

Our deep sense of gratitude to each and every person involved directly or indirectly in this project work with special thanks to our classmates.

Varun Burde **(0875EC131111)**

Abstract

Quadrupedalism or pronograde posture is a form of terrestrial locomotion in animals using four limbs or legs. An animal or machine that usually moves in a quadrupedal manner is known as a quadruped, meaning "four feet" (from the Latin quattuor for "four" and pes for "foot"). Quadruped robots tend to look either animal-like or insect-like and are designed to help you get started with multi-legged motion. This project represents the same motion using Arduino and Servo Motors. The project is based on cybernetics which will show control engineering and Robotics. This four legged proposed model is capable to work on most of the terrain and are more successful than wheeled vehicle on different terrain. This model can find their applications in field of military to transport ammunition in during warzone, hurdling the various path obstacles. Can also be future household assistant to human with capabilities of traveling on stairs and other house paths. They can be a replacement of our former wheeled vehicle in the areas where they may fail in places like desert and random landmass. Moreover, it can be used to deeply understand the locomotion of real being and mimic on this model for research purpose.

List of Figures

List of Abbreviation

List Of Tables

Contents

CHAPTER 1
INTRODUCTION

We had made an electromechanical model which something looks like dog, and imitate the actions of a dog like sitting and walking. This projects tells about the locomotion of leg creature in general and how it can be performed in physical model. The model is made up of metal sheets, servomotors, and wood. We have use Arduino mega board to define the logic and perform actions. Servo motors and PWM motors so it need 3 terminals – power, ground and signal.

We have used the mpu6050 sensor which is 3 axis gyroscope and accelerometer, we take the input from the gyro sensor and made a feedback system for the stability of the model. The senor gives the inputs from all direction but we have utilized only two directions namely pitch and roll. The feedback loop system keeps the robot structure to stable with variation of motors angle and prevents it from falling.

1.1 Quadrupedalism

Quadrupedalism or pronograde posture is a form of terrestrial locomotion in animals using four limbs or legs. An animal or machine that usually moves in a quadrupedal manner is known as a quadruped, meaning "four feet" (from the Latin quattuor for "four" and pes for "foot"). Quadruped robots tend to look either animal-like or insect-like and are designed to help you get started with multi-legged motion.

Figure 1.1 Examples of Quadruped Robot

Legged robots are a type of mobile robot. They are somewhat a recent innovation in robotics. However, many or all bipedal models are not practical because they are cumbersome and slow. Most successful legged robots have four or six legs for further stability. This legs over wheels approach lends itself for use in all-terrain purposes because legs are more effective in an uneven environment than wheels.

1.2 Metal Gear Servo Motor(MG995)

A servo is a mechanical motorized device that can be instructed to move the output shaft attached to a servo wheel or arm to a specified position. Inside the servo box is a DC motor mechanically linked to a position feedback potentiometer, gearbox, electronic feedback control loop circuitry and motor drive electronic circuit.

A typical R/C servo looks like a plastic rectangular box with a rotary shaft coming up and out the top of the box and three electrical wires out of the servo side to a plastic 3 pin connector. Attached to the output shaft out the top of the box is a servo wheel or Arm. These wheels or arms are usually a plastic part with holes in it for attaching push / pull rods, ball joints or other mechanical linkage devices to the servo. The three electrical connection wires out of the side are V- (Ground), V+ (Plus voltage) and S Control (Signal). The control S (Signal) wire receives Pulse Width Modulation (PWM) signals sent from an external controller and is converted by the servo on board circuitry to operate the servo.

R/C Servos are controlled by sending pulse width signals (PWM) from an external electronic device that generates the PWM signal values, such as a servo controller, servo driver module or R/C transmitter and receiver. Pulse Width Modulation or PWM signals sent to the servo are translated into position values by electronics inside the servo. When the servo is instructed to move (Received a PWM signal) the on board electronics convert the PWM signal to a electrical resistance value and the DC motor is powered on. As the motor moves and rotates the linked potentiometer also rotates. Electrical resistance value from the moving potentiometer are sent back to the servo electronics until the potentiometer value matches the position value sent by the on-board servo electronics that was converted from the PWM signal. Once the potentiometer value and servo electronic signals match, the motor stops and waits for the next PWM signal input signal for conversion.

Figure 1.2 Servo motor (mg995)

Specifications

• Weight: 55 g

• Dimension: 40.7 x 19.7 x 42.9 mm approx.

• Stall torque: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)

• Operating speed: 0.2 s/60º (4.8 V), 0.16 s/60º (6 V)

• Operating voltage: 4.8 V a 7.2 V

• Dead band width: 5 µs

• Stable and shock proof double ball bearing design

• Temperature range: 0 ºC – 55 ºC

1.3 IMU (Inertia Measurement Unit) sensor (MPU 6050)

IMU sensors are one of the most inevitable type of sensors used today in all kinds of electronic gadgets. They are seen in smartphones, wearables, game controllers, etc. IMU sensors help us in getting the attitude of an object, attached to the sensor in three dimensional space. These values usually in angles, thus help us to determine its attitude. Thus, they are used in smartphones to detect its orientation. And also in wearable gadgets like the Nike fuel band or fit bit, which use IMU sensors to track movement. IMU sensors, thus have prolific number of applications. It is even considered to be an inexorable component in quadrotors.

IMU sensors usually consists of two or more parts. Listing them by priority, they are: accelerometer, gyroscope, magnetometer and altimeter. The MPU 6050 is a 6 DOF (Degrees of Freedom) or a six axis IMU sensor, which means that it gives six values as output. Three values from the accelerometer and three from the gyroscope. The MPU 6050 is a sensor based on MEMS (Micro Electro Mechanical Systems) technology. Both the accelerometer and the gyroscope is embedded inside a single chip. This chip uses I2C (Inter Integrated Circuit) protocol for communication.

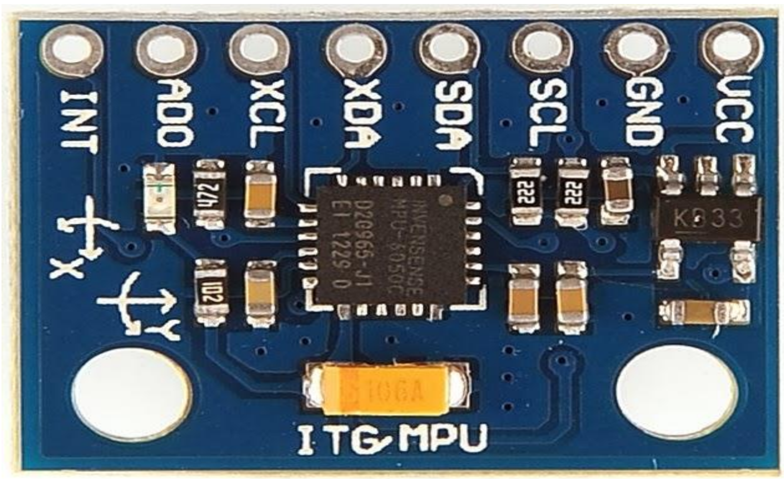


Figure 1.3 MPU 6050

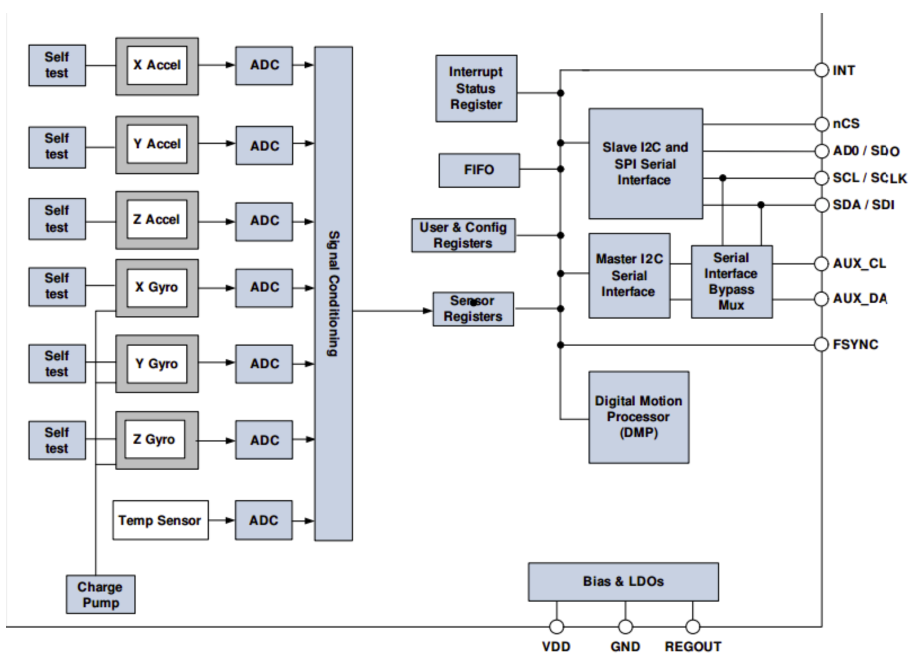


Figure 1.4 Block Diagram MPU6050

1.4 Arduino Mega (2560)

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, and 16 MHz crystal oscillator, a USB connection, a power jack and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimilar

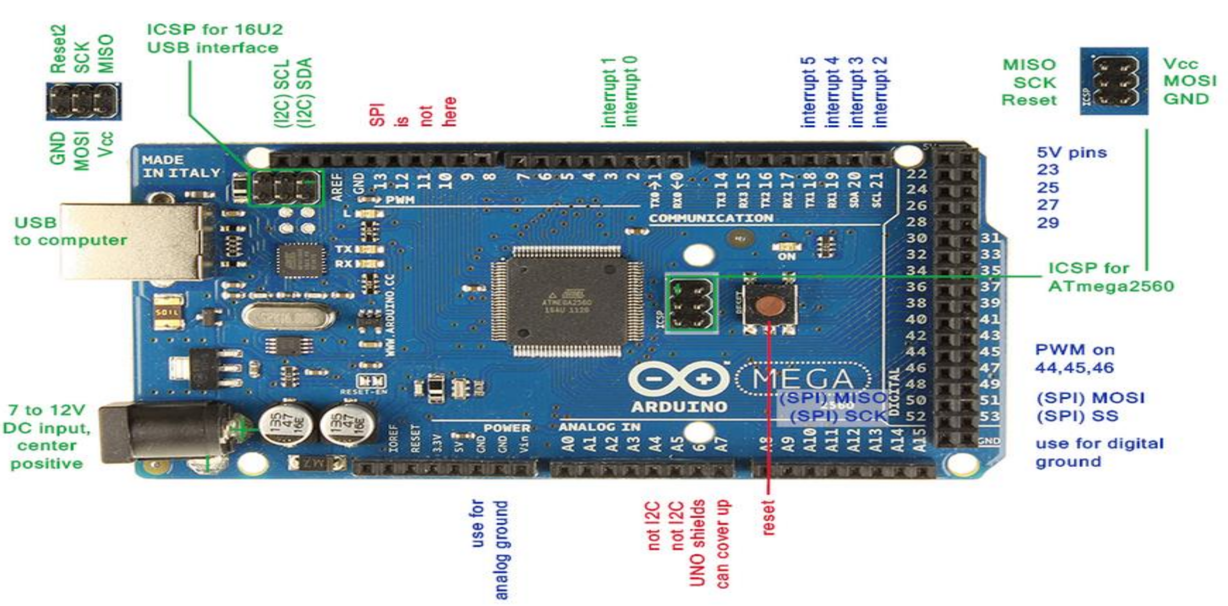


Figure.1.5 Arduino Mega (2560) Board

TECHNICAL DETAILS:

Microcontroller	*ATmega2560*
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE REVIEW

Robotic quadrupeds which can function as transportation machines present many advantages due to their high mobility and ability to traverse rugged terrain as opposed to conventional wheeled robots. However, the high degrees of freedom and large number of actuators associated greatly increase their energy expenditure.

A large group of researchers are already actively engaged in working on Limit Cycle Walking, as will be shown in the following section. However, there is a small difference between passive walkers and limit cycle walkers with the former being the subset of the latter. Many robots are derivatives of the so-called Passive Dynamic Walkers , a subgroup of Limit Cycle Walkers.

In the first section, Passive Dynamic Walking robots will be addressed, followed by actuated Limit Cycle Walkers. Most robots using this concept are referred to as `Passive-based' walking robots, but it’s important to note that Limit Cycle Walking is a much more accurate description for them than the former.

The Arduino quadruped robot Stompy[4], a quadruped using eight servos, with each leg consisting of a hip joint and an ankle joint (no knee). Stompy uses a “trot” type gait where the front right leg is synchronized with the back left leg and these two limbs move in antiphase with the other two limbs. The result is a stomping type of walk, but it gets the job done. Controlling all the servos requires eight PWM pins and standard Arduinos only have six, so I decided to use an Arduino Mega. Stompy uses an IR distance sensor to keep from bumping into things and is capable of autonomous movement.

Popular existing Quadrupeds

1. StarlETH from ETH Zurich[2]:

This project, initiated by Christian David Remy from ETH Zurich aims at designing and constructing an energy efficient robotic quadruped which is capable of vast multitudes of dynamic motions such as walking, trotting, etc. While the work is currently in progress, the objective is to come up with fundamentally new control strategies that incorporate the advantageous characteristics of passive dynamics into an optimal controller. This quadruped has highly compliant elastic legs.

2. BigDog from Boston Dynamics[3]:

BigDog is a rough-terrain robot that walks, runs, climbs and carries heavy loads. The power is derived from an engine that drives a hydraulic actuation system. It has four legs that are articulated like an animal's and also has consists of compliant elements to absorb shock and subsequently, recycle energy from one step to the next. However, the most appealing factor is the size: about 3 feet long, 2.5 feet tall and weighs 240 lbs.

The main disadvantage like any other powered/actuated quadruped robot is its inherent incapability to utilize the natural dynamics and incorporate an efficient gait. Even though it has the ability to walk 20miles without refueling, yet that feat would most certainly go to its bulky engine and excessive performance specifications which are not all efficient.

CHAPTER 3

PROBLEM STATEMENT AND SOLUTION DOMAIN

3.1 PROBLEM STATEMENT

We have been using the wheel model (vehicle) for a long time of commuting and transportation. They were quite good on even land but, it was difficult to travel on uneven land with them. The problem with them was, travelling with them on random surface, can be quite difficult and unstable. And also there are some places where it’s impossible for wheeled robot to climb or pass the hurdle the obstacles. This model is key how it can be solved.

3.2 SOLUTION DOMAIN

Legged robots, unlike wheeled robots, have the potential to access nearly all of the earth's land mass, enabling robotic applications in areas where they are currently infeasible. The solution come in our mind looking at the Quadruped creature around us like horse, camel etc. all these creatures around us are capable of travelling different terrain across landmass successfully. This ability of them can be observed and can be implemented on model to sole above problem.

Legged robots are a type of mobile robot. They are somewhat a recent innovation in robotics. However, many or all bipedal models are not practical because they are cumbersome and slow. Most successful legged robots have four or six legs for further stability. This legs over wheel approach lends itself for use in all-terrain purposes because legs are more effective in an uneven environment than wheels.

The advantage of using this model is that they can travel to variety of terrain with even as well as uneven surface with more stability and confidence. Some of the big advantage of having such model is that they can go to places where wheel machine can’t go as well as the places where human entrance is vulnerable. This model is capable of hurdling the various path obstacles like stairs, spits, holes, slopes with ease and must be implemented for promising future.

CHAPTER 4

DESIGN OF SYSTEM

The chapter describes the design and construction of the system i.e. the connection of Arduino to Servos Mg995 and Arduino to MPU 6050. The basic connections and the components used are described below:

4.1DESIGN OF SERVO MOTOR MG995

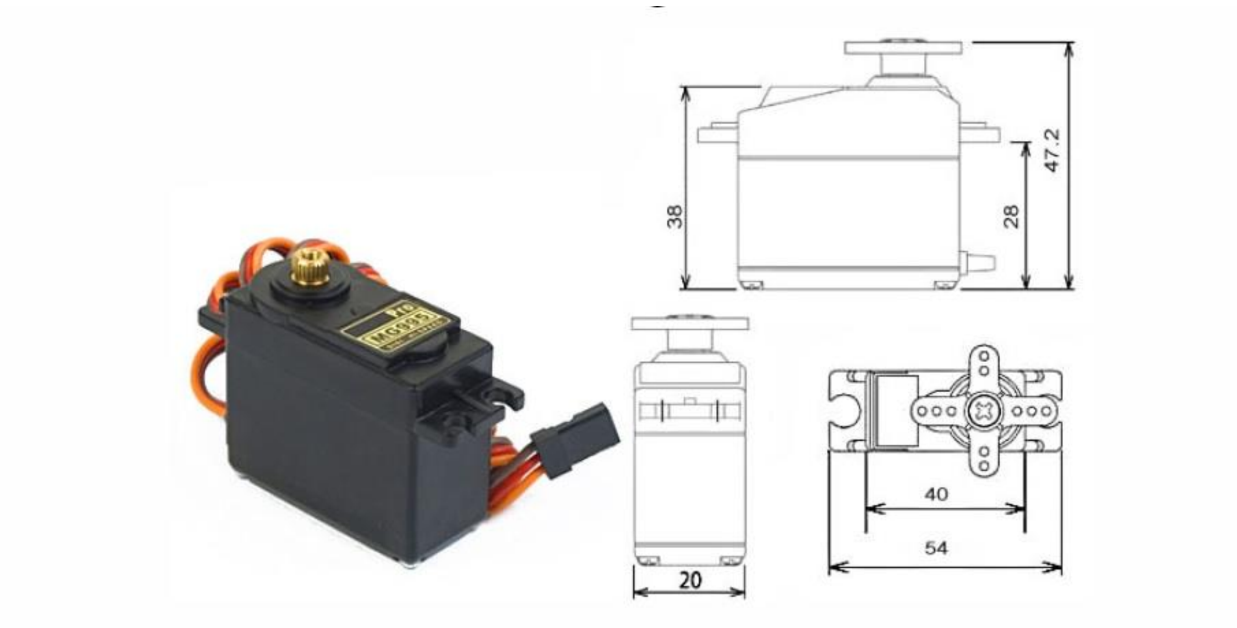


Figure 4.1 Design of Servo Mg995

The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-speed standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos. We can make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG995 Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast

4.2 INTERFACING MG995 WITH ARDUINO

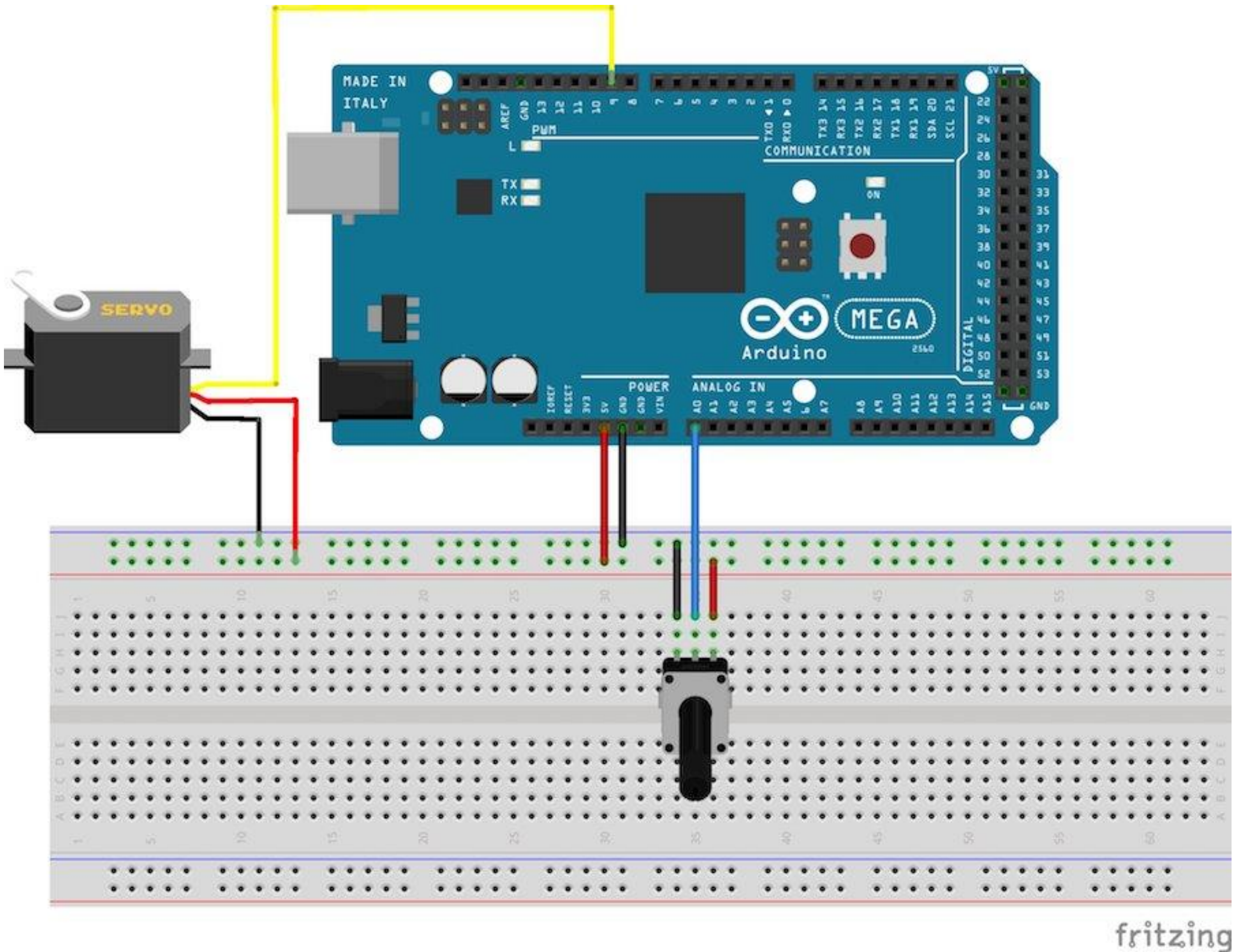


Figure 4.2 Interfacing Mg995 With Arduino

Table 1 Connecting Arduino to servo motors

Arduino pin	Servo pin
3	servo motor 1 - Signal
4	servo motor 2 - Signal
5	servo motor 3 -Signal
6	servo motor 4 -Signal
7	servo motor 5 -Signal
8	servo motor 6 -Signal
9	servo motor 7 -Signal
10	servo motor 8 -Signal

The diagram clearly shows the connection of the motor with the Arduino

4.3 PIN DESCRIPTION OF SERVO MOTOR

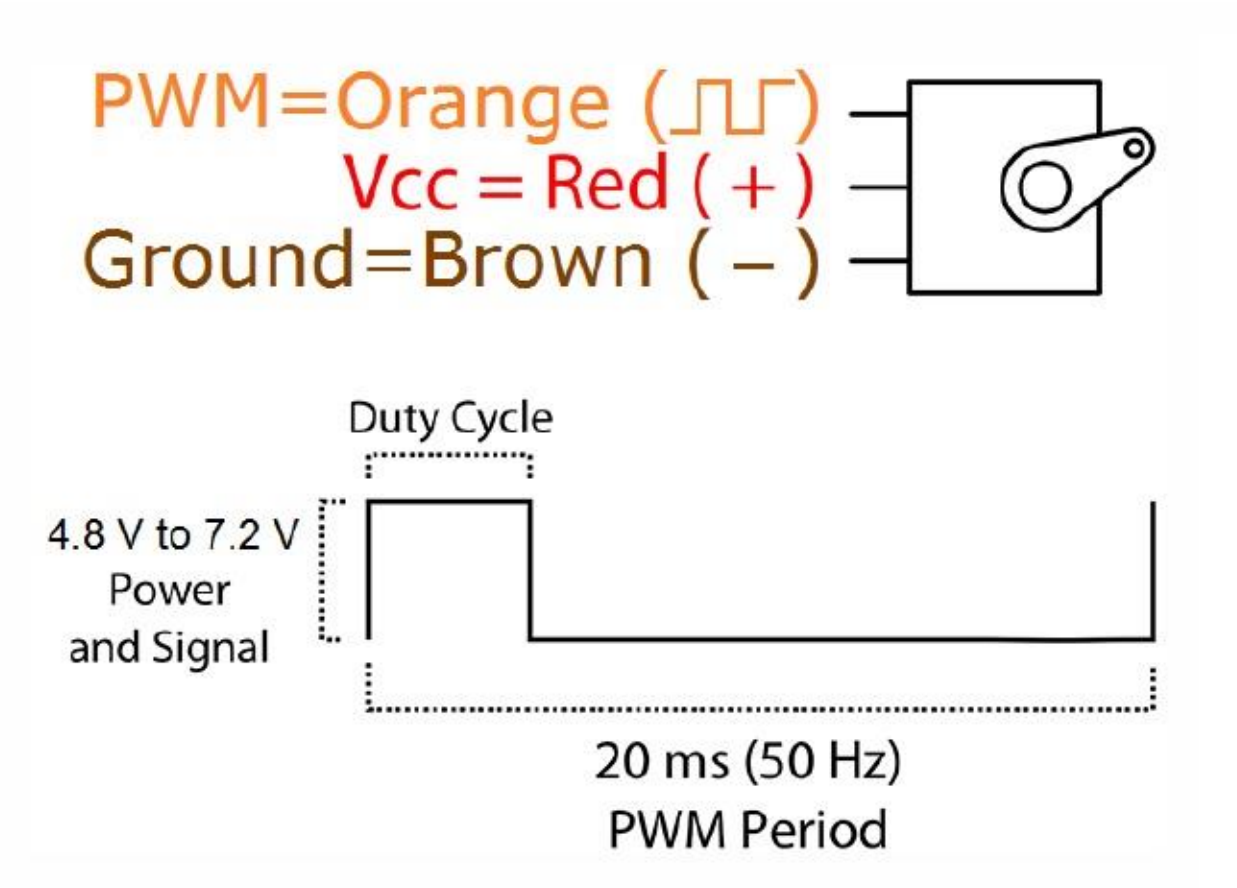


Figure 4.3 Pin Description of Mg995

Orange: This wire is for pwm signal.

Red: this wire is for power supply, recommended voltage is 4-6 volt.

Brown: This wire is for ground.

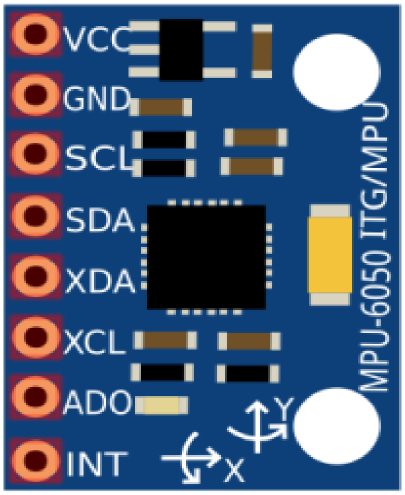
4.4 DESIGN OF IMU (INERTIA MEASUREMENT UNIT) SENSOR (MPU 6050)

IMU (Inertia Measurement Unit) sensor (MPU 6050) is controlled with the I²C protocol.

I²C (Inter-Integrated Circuit), pronounced I-Squared-C, is a multi-master, multi-slave, packet switched, single-ended, serial computer bus invented by Philips Semiconductor (now NXP Semiconductors). It is typically used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance, intra-board communication.

I²C uses only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL), pulled up with resistors. Typical voltages used are +5 V or +3.3 V although systems with other voltages are permitted.

Mpu 6050 pinout



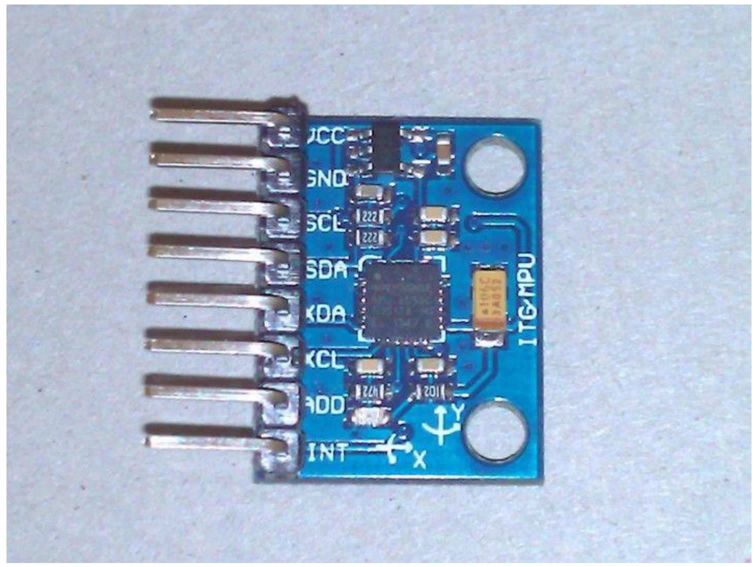


Figure 4.4 pinout of MPU6050

Interfacing the Arduino MPU 6050

The MPU 6050 communicates with the Arduino through the I2C protocol. The MPU 6050 is connected to Arduino as shown in the following diagram. If your MPU 6050 module has a 5V pin, then you can connect it to your Arduino’s 5V pin. If not, you will have to connect it to the 3.3V pin. Next, the GND of the Arduino is connected to the GND of the MPU 6050.

Table 2 Connection of MPU6050 to Arduino

Arduino pin	MPU6050 Pin
3.3V	Vcc
Gnd	Gnd
Digital pin 2	Int
SDA 20	SDA
SCL 21	SCL

The program we will be running here, also takes advantage of the Arduino’s interrupt pin. Connect your Arduino’s digital pin 2 (interrupt pin 0) to the pin labeled as INT on the MPU 6050. Next, we need to set up the I2C lines. To do this, connect the pin labeled SDA on the MPU 6050 to the Arduino’s analog pin 20(SDA) and the pin labeled as SCL on the MPU 6050 to the Arduino’s analog pin 21 (SCL). That’s it, you have finished wiring up the Arduino and MPU 6050.

CHAPTER 5
METHODOLOGY AND SOFTWARE

5.1 Block diagram

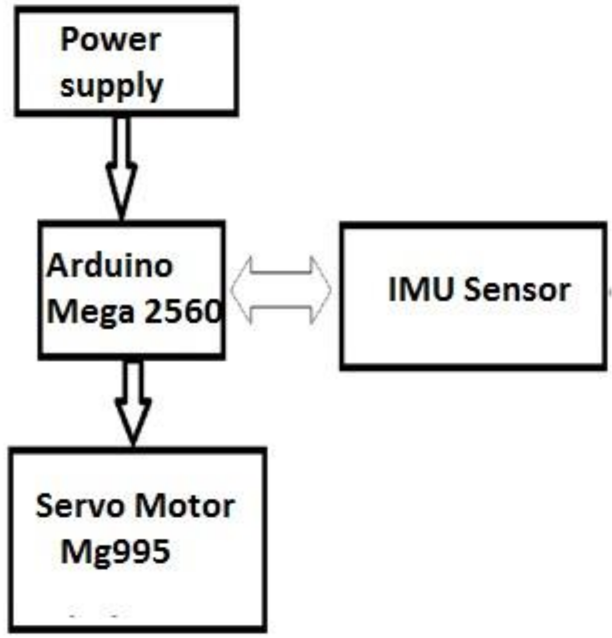


Figure.5.1 Block Diagram

Explanation:

Block Diagram of project contain Power supply unit in this we use 4-6volt power supply, it can be either battery or power adapter which can provide minimum current to drive Servo motor. It will power the Arduino, Servo motors and IMU sensor. Arduino Mega 2560 Board this board contain Microcontroller from Avr family which will store the code and runs all the logics provided in the code. The servo motor and IMU sensor are interceded with the Arduino with the help of fly leads. IMU sensor is our input unit of the system which consist of Gyroscope and accelerometers which will get values in Yaw, Pitch and Roll direction. We will be taking use of Pitch and roll values for coding. And the Servo motor Mg995 which will be our output unit which will drive our robot and give the precise movement with the provided signal by the Arduino.

5.2 Methodology

IMU sensor 6050

IMU sensors usually consist of two or more parts. Listing them by priority, they are the accelerometer, gyroscope, magnetometer, and altimeter. The MPU 6050 is a 6 DOF (Degrees of Freedom) or a six-axis IMU sensor, which means that it gives six values as output. Three values from the accelerometer and three from the gyroscope. The MPU 6050 is a sensor based on MEMS (Micro Electro Mechanical Systems) technology. Both the accelerometer and the gyroscope are embedded inside a single chip. This chip uses I2C (Inter-Integrated Circuit) protocol for communication.

Accelerometer

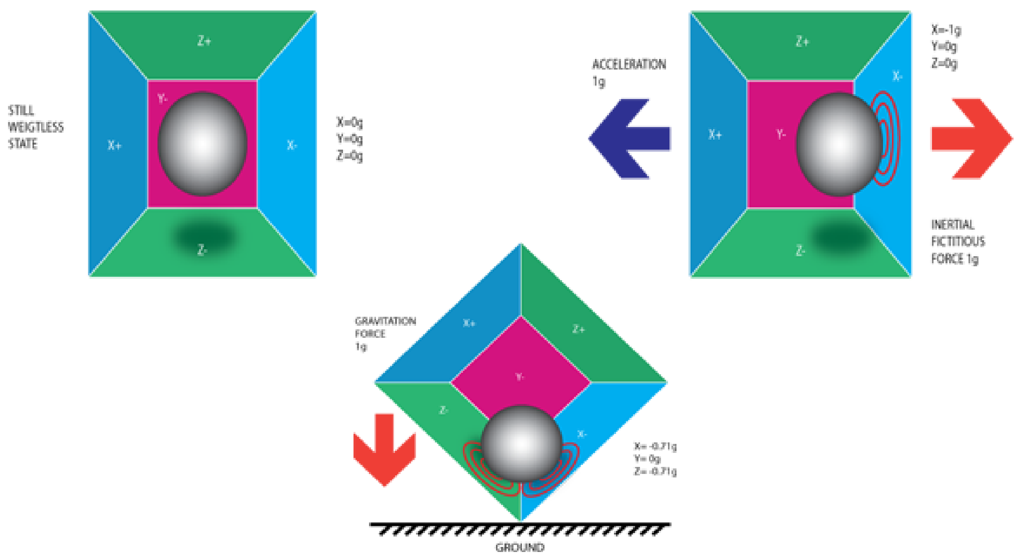


Figure 5.2Piezo Electric Accelerometer

An accelerometer works on the principle of the piezoelectric effect. Here, imagine a cuboidal box with a small ball inside it, like in the picture above. The walls of this box are made with piezoelectric crystals. Whenever you tilt the box, the ball is forced to move in the direction of the inclination, due to gravity. The wall that the ball collides with creates tiny piezoelectric currents. There are three pairs of opposite walls in a cuboid. Each pair corresponds to an axis in 3D space: X, Y, and Z axes. Depending on the current produced from the piezoelectric walls, we can determine the direction of inclination and its magnitude.

Gyroscope

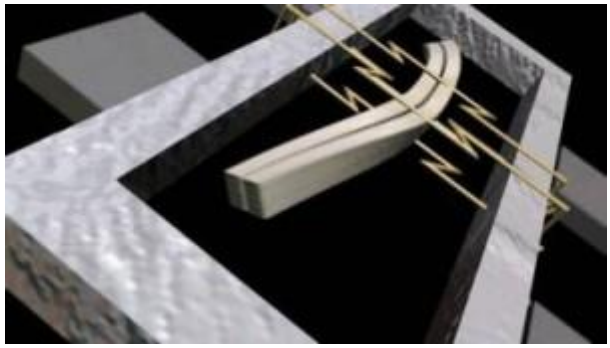


Figure 5.3 Piezo Electric Gyroscope

Gyroscopes work on the principle of Coriolis acceleration. Imagine that there is a fork-like structure that is in a constant back and forth motion. It is held in place using piezoelectric crystals. Whenever you try to tilt this arrangement, the crystals experience a force in the direction of inclination. This is caused as a result of the inertia of the moving fork. The crystals thus produce a current in consensus with the piezoelectric effect, and this current is amplified. The values are then refined by the host microcontroller.

Servo motor Mg995

To fully understand how the servo works, you need to take a look under the hood. Inside there is a pretty simple set-up: a small DC motor, potentiometer, and a control circuit. The motor is attached by gears to the control wheel. As the motor rotates, the potentiometer's resistance changes, so the control circuit can precisely regulate how much movement there is and in which direction.

When the shaft of the motor is at the desired position, power supplied to the motor is stopped. If not, the motor is turned in the appropriate direction. The desired position is sent via electrical pulses through the signal wire. The motor's speed is proportional to the difference between its actual position and desired position. So if the motor is near the desired position, it will turn slowly, otherwise it will turn fast. This is called proportional control. This means the motor will only run as hard as necessary to accomplish the task at hand, a very efficient little guy.

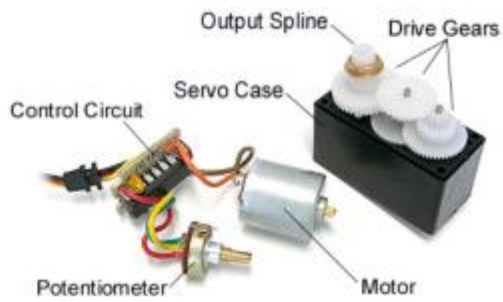


Figure 5.4 The guts of a servo motor (L) and an assembled servo (R)

Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually only turn 90° in either direction for a total of 180° movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the motor determines position of the shaft, and based on the duration of the pulse sent via the control wire; the rotor will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position. Shorter than 1.5ms moves it in the counter clockwise direction toward the 0° position, and any longer than 1.5ms will turn the servo in a clockwise direction toward the 180° position.

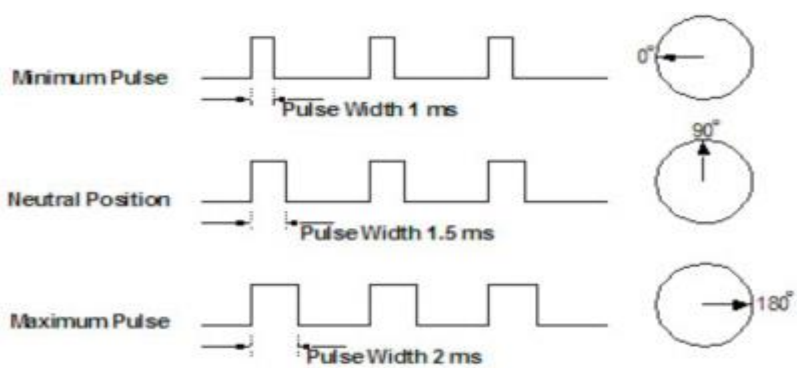


Figure 5.5 Variable Pulse width control servo position

When these servos are commanded to move, they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is called the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

5.3 Process of Uploading the Code and Testing the Arduino MPU 6050

To test the Arduino MPU 6050, first download the Arduino library for MPU 6050, developed by Jeff Rowberg. You can find the library here. Next, you have to unzip/extract this library and take the folder named “MPU6050” and paste it inside the Arduino’s “library” folder. To do this, go to the location where you have installed Arduino (Arduino –> libraries) and paste it inside the libraries folder. You might also have to do the same thing to install the I2C dev library if you don’t already have it for your Arduino. Do the same procedure as above to install it, you can find the file here: I2Cdev library.

If you have done this correctly, when you open the Arduino IDE, you can see “MPU6050” in File –> Examples. Next, open the example program from: File –> Examples –> MPU6050 –> Examples –> MPU6050_DMP6.

Figure 5.6 Arduino MPU 6050 DMP code

Next, you have to upload this code to your Arduino. After uploading the code, open up the serial monitor and set the baud rate as 115200. Next, check if you see stuff like "Initializing I2C devices…" on the serial monitor. If you don't, just press the reset button. Now, you'll see a line saying "Send any character to begin DMP programming and demo." Just type in any character on the serial monitor and send it and you should start seeing the yaw, pitch, and roll values coming in from the MPU 6050. Like so:

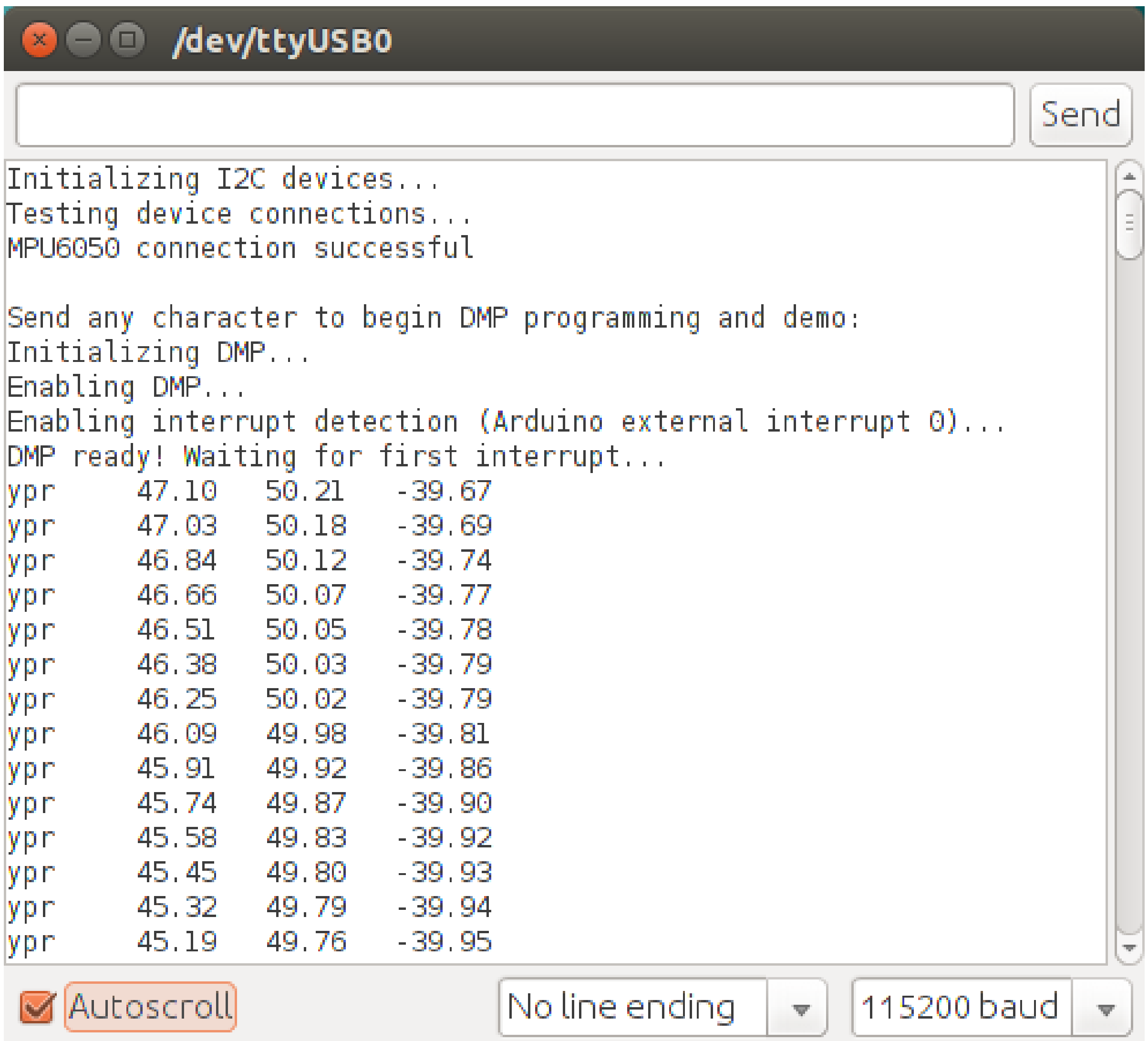


Figure 5.7Arduino MPU 6050 Serial Monitor

DMP stands for Digital Motion Processing. The MPU 6050 has a built-in motion processor. It processes the values from the accelerometer and gyroscope to give us accurate 3D values.

Process of testing servo with Arduino

Use any programs from the servo examples

We have used sweep to check the working of servo, make sure to properly connect the pin given in the code

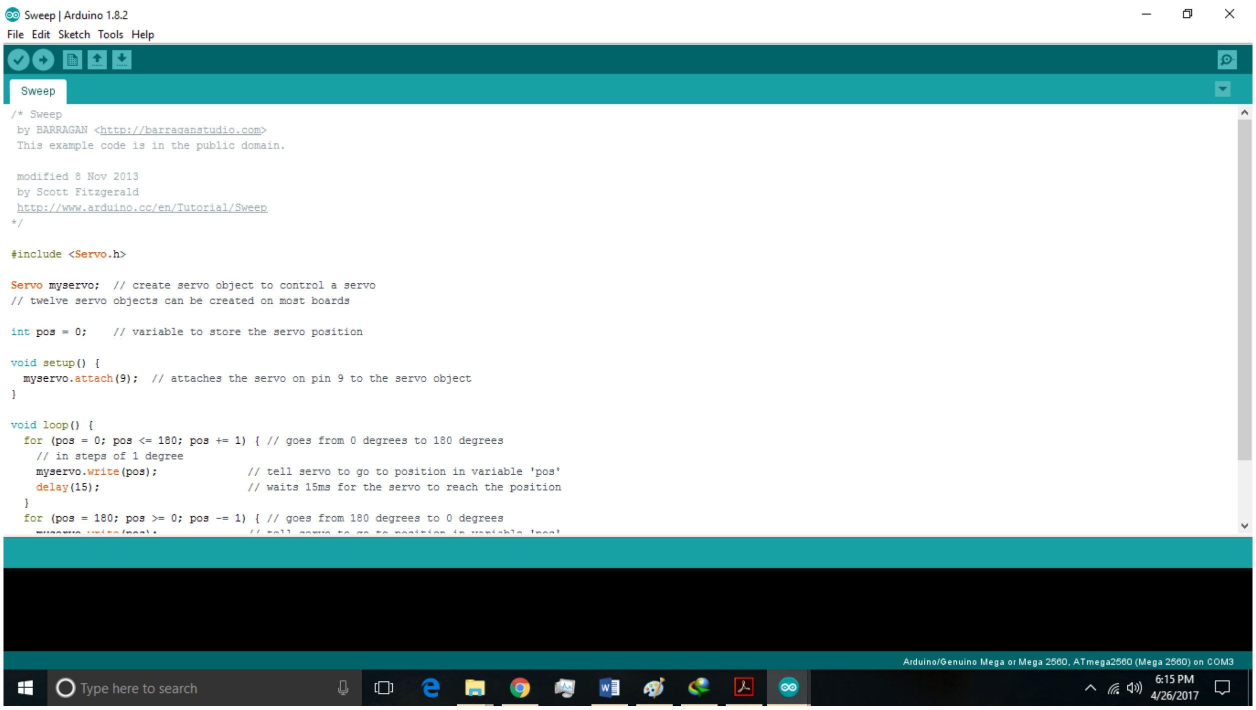


Figure 5.8 Servo Code

Then check for the Servo response, whether it’s working or not.

CHAPTER 6

WORKING AND TROUBLESHOOTING

6.1 Working

1. First we interfaced servo motor with Arduino with help of fly leads. After connection, we upload the program of test code found in the downloaded library in Arduino Mega Board by the help of Arduino IDE software.
2. Second we designed our chassis with wood. We screwed down our servo motor on plywood. We made a Limb by cutting the wood of correct dimension and fix with the attachment provided with the servo.

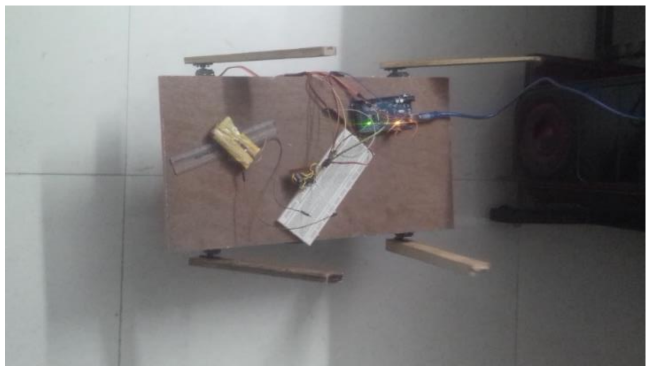


Figure 6.1 First Model

Figure 6.2 Tools

3. Unfortunately, our first model wasn’t a success and we had to made some major modification for betterment of chassis.

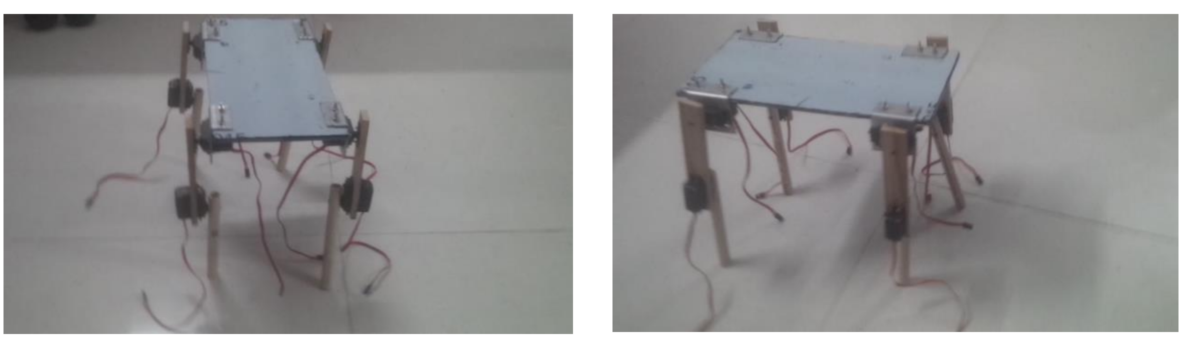


Figure 6.3 Second Model

4. Then we interface IMU 6050 sensor with Arduino, Upload a test code from Arduino library. Use the calibration program to Get the offset and make it further accurate.
5. Connected Servo motors and IMU sensor simultaneously and written a code for IMU for input and Servo motor as output device, and then uploaded the test code

Figure 6.4 Test Code

6. Finally, the whole hardware is assembled, interfaced and mounted on the chassis, and the result is noted down.

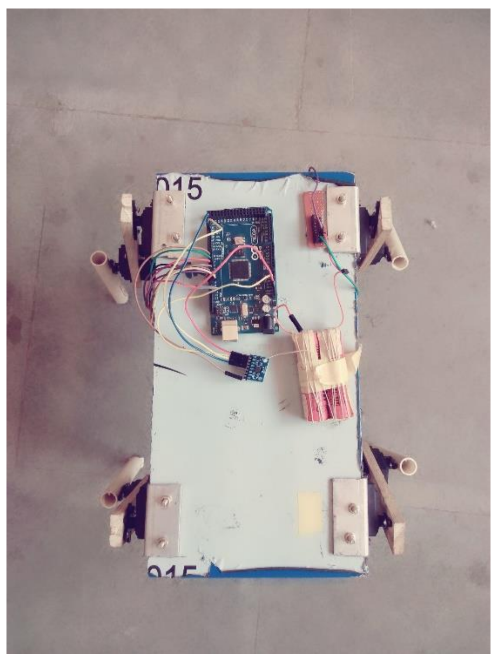


Figure 6.5 Final Product

6.2 Troubleshooting

The structure part was the most time consuming part. It took weeks to make a perfect the structure.

Problem that we have faced during experimentation were:

1. We were unable to cut out perfect L shape angle, due to which motors weren't able to fit in perfect position.
2. Making of limb required lot of skill to cut out for motors to fit inside cavity.

The coding part was also time consuming part. It took weeks to make a perfect coding.

Problem that we have faced during experimentation were:

1. We tried many codes to just make it walk, not all the codes could make it move forward.
2. We had to change code many times just to adjust perfect angle position of servos.

CONCLUSION AND FUTURE WORK

CONCLUSION AND FUTURE WORK

All in all, we consider this project to be a success. We have implemented a elecromechanical model wich can mimic movemnt of animals. As almost everything described already for this design, we would like to say there are still numerous kinds of enhancements one can implement on this project to make it even more convenient.

It will be more convenient to use highly precise servo motor of better motion. The Chassis can be built with the help of CNC machine or parts can be made out of 3d printer for less error. More efficient chassis can include the 12 motors servo assembly which will work more real alike. will include hip movement as well. The motion can be enhanced and could made more stable by adding suspension and so it would be more stable during jerks and fall down.

In future the problem of the problem of traveling to mountain and other unexplored land by wheel, will be replaced by this kind of model. And these kind of model will find application where Human reach is vulnerable Legged robots, unlike wheeled robots, have the potential to access nearly all of the earth's land mass, enabling robotic applications in areas where they are currently infeasible. However, the current control software for legged robots is quite limited, and does not let them realize this potential. In the Learning Locomotion project, we seek to develop software that significantly advances the state of the art in robotic quadruped locomotion over rough terrain. This model can find their applications in field of military to transport ammunition in during warzone, hurdling the various path obstacles. Can also be future household assistant to human with capabilities of traveling on stairs and other house paths. They can be a replacement of our former wheeled vehicle in the areas where they may fail in places like desert and random landmass. Moreover, it can be used to deeply understand the locomotion of real being and mimic on this model for research purpose.

REFRENCES

REFRENCES

[1] McGeer, Tad. "Passive dynamic walking." *the international journal of robotics research* 9.2 (1990): 62-82.

[2] Hutter, M. A. R. C. O., et al. "StarlETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion." *Int. Conf. on Climbing and Walking Robots (CLAWAR)*. 2012.

[3] Raibert, Marc, et al. "Bigdog, the rough-terrain quadruped robot." *Proceedings of the 17th World Congress*. 2008.

[4] Stompy the Arduino quadruped robot on Instructables by joesinstructables
http://www.instructables.com/id/Arduino-Quadruped-Robot/

APPENDIX

APPENDIX 1

SOFTWARE PROGRAM

-------------Code Started------------

```
#include <Servo.h>

Servo frontlu, frontld, frontru, frontrd, backlu, backld, backru, backrd  ; // create servo
object to control a servo


int pos = 120;    // variable to store the servo position

void setup() {
  backlu.attach(2);
  backld.attach(3);
  backru.attach(4);
  backrd.attach(5);
  frontlu.attach(6);
  frontld.attach(7);
  frontru.attach(8);
  frontrd.attach(9);
  backlu.write(0);//120
  backld.write(180);//60
  backru.write(180);//60
  backrd.write(0);//120
  frontlu.write(90);//120
  frontld.write(0);//60
  frontru.write(90);//60
  frontrd.write(180);//120
  delay(5000);
}

void loop()  {
  for (pos = 90; pos <= 120; pos += 1)
  { // goes from 0 degrees to 180 degrees
    frontlu.write(pos);
    backru.write(180 - pos);
```

```
    frontrd.write(pos);
    backld.write(180 - pos);

    delay(10);
  }
  delay(500);
  for (pos = 90; pos <= 120; pos += 1)
  { // goes from 180 degrees to 0 degrees
    backlu.write(pos);
    frontru.write(180 - pos);
    backrd.write(pos);
    frontld.write(180 - pos);
    delay(10);
  }
   delay(500);
}
```