

GESTURE CONTROLLED ROBOTIC ARM

*Project report submitted in partial fulfillment of the
requirement for the degree of*

BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING

By

Mudit Katiyar (181020)

UNDER THE GUIDANCE OF

DR. Emjee Puthooran



**JAYPEE UNIVERSITY OF INFORMATION
TECHNOLOGY, WAKNAGHAT**

June 2022

CONTENT

Declaration

Abstract

CHAPTER-1 INTRODUCTION

1.1 Robotics

1.2 Scope of Robotics

1.3 IFR Report

1.4 Application of Robotics

CHAPTER-2 TECHNICAL SPECIFICATION

2.1 Control Board

2.2 Actuator

2.3 Sensor

2.4 Wireless Module

CHAPTER-3 PROJECT DESIGN

3.1 Robotics Arm (Version – 1)

3.2 Control through Hand Gesture (Version – 1)

3.3 Robotic Arm (Version – 2)

3.4 Robotic Arm (Version – 3)

CHAPTER -4 ADDITIONAL COMPONENT

CHAPTER -5 CONCLUSION AND FUTURE SCOPE

REFERENCES

APPENDIX

CODE – 1

CODE – 2

DECLARATION

We hereby declare that the work reported in the B.Tech Project Report entitled Gesture Controlled Robotic Arm submitted at Jaypee University of Information Technology, Waknaghat, India is an authentic record of our work carried out under the supervision of Dr Emjee Puthooran. We have not submitted this work elsewhere for any other degree or diploma.

Mudit Katiyar (181020)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Emjee Puthooran

Date: 27/05/2022

INTRODUCTION

1.1 Robotics

Robot, any automatically operated machine that replaces human effort, though it may not resemble human beings in appearance or perform functions in a humanlike manner. A robot is a machine—especially one programmable by a computer—capable of carrying out a complex series of actions automatically. A robot can be guided by an external control device, or the control may be embedded within. Robotics is an interdisciplinary branch of the engineering and the Computer science. It involves design, construction, operation and use of the Robots [1].

In order to address the concern of Robotic harming a human being, Issac Asimov proposed three laws. He described a Robot as Humanoid, devoid of feelings and whose brain could be programmed by the human beings. These laws should be valid for the Real Robots [2].

- 1) *“A robot should not injure human being or, through inaction, allow a human to be harmed.”*
- 2) *“A robot must obey orders given by humans except when that conflicts with the First Law.”*
- 3) *“A robot must protect its own existence unless that conflicts with the First or the Second Law.”*

These were the general laws which can be applicable to the machines and other appliances. The Asimov laws were always been taken care during any designing of the Robot.

Types of Robots

There are different types of Robots which can be used for the different application. These Robots were design on the basis of the need, reducing time and human complexity. As they perform the task much more faster than the human and also avoid some error which cannot be seen with the naked eyes of the human.

Pre Programmed Robots-

These are the Robots which can be coded within a devices like Arduino, Raspberry pi etc. These Robots can be used to perform task like pick and place, manufacturing IC and other automotive industry. These can have feedback loop to provide more accuracy.



Figure 1.1.1 Pre-programmed Robot

Humanoid Robots

These robots must look like human or perform similar task like human. They can perform task like running, jumping or carrying object. Humanoid are sometimes designed to look like human. Because of these similarity, they can be called as humanoid. The most famous example for the humanoid is SOPHIA.



Figure 1.1.2 Sophia Robot

It is the most advanced robots which depicts the future of AI and advanced Robotics. The Robot itself is the unique combination of the science and engineering as well as the world first Robot citizen.

Autonomous Robots

These Robots must operate independently without the intervention of the human beings. Because they perform task with the help of the sensor to perceive the world around them and then take decision which can be optimum based on the situation. They require much more computation and data transmission.



Figure 1.1.3 Roomba Vacuum cleaner.

The best example for the Autonomous Robot can be Roomba that can be used as automatic home cleaner which also uses sensor to roam freely throughout a home.

1.2 Scope of Robotics

Robotics can be considered as the Field of combination of Robotic technology and the Artificial Intelligence. These Robotic Engineer can work with different organization like the ISRO, NASA etc. Even the developed countries were now trying to use Robots for various industry to increase their productivity. So this also account for increasing the Scope of Robotics in the Future. There are courses for the Robotic engineering in the institutes like IIT, NIT, BITS, VIT etc.

1.3 IFR Report

IFR stands for International Federation of Robotics is a non-profit organization aims at promoting the world of Robotics around the globe. It promotes the research, development and international cooperation in the entire field of Robotics [3].

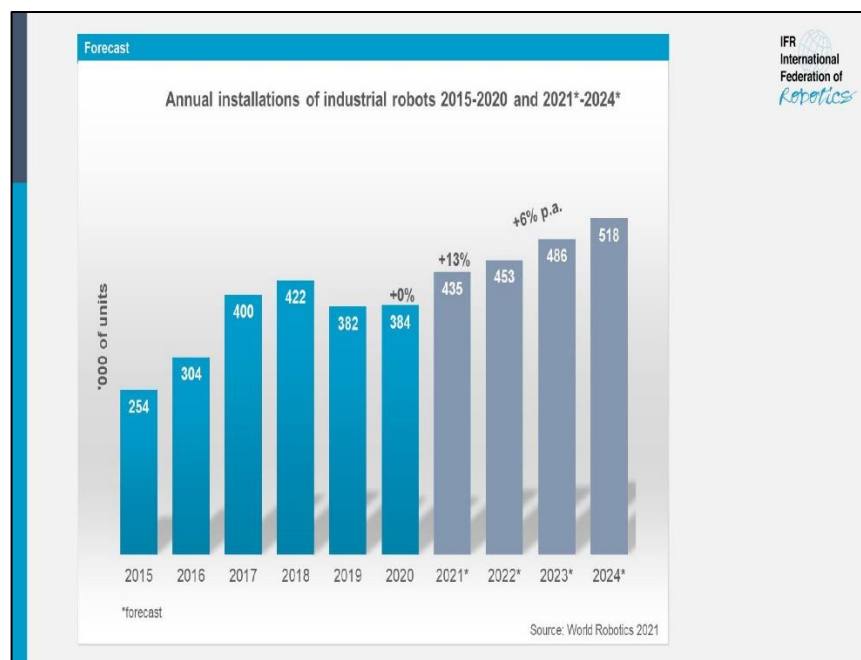


Figure 1.3.1 Installation Report by IFR

According to the report the number of installation of the Robotic technology around the globe is high at 2018. But there were certain decrease in at 2019 and 2020 due to the Covid19 crisis. In future there should be increase in the no of installation of the Robotic technology in the various manufacturing industries as well as for other application.

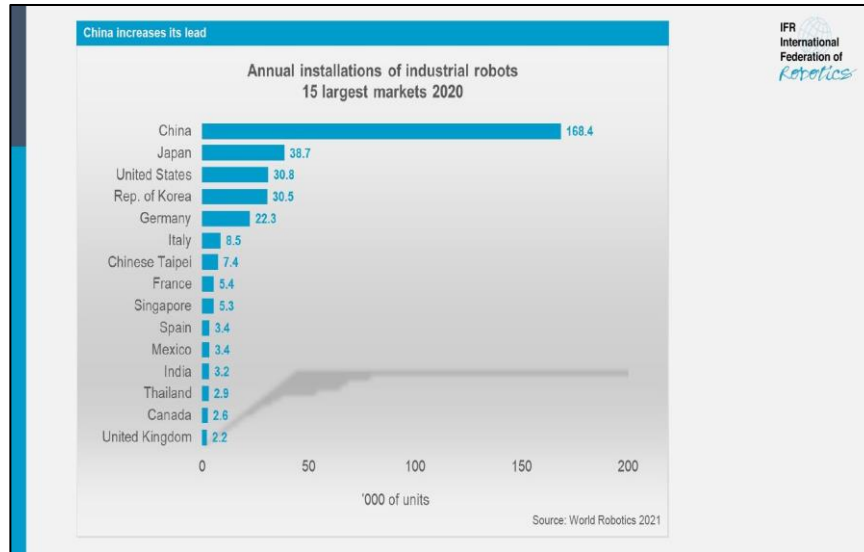


Figure 1.3.2 Installation Report Countries Wise

This report determine the top leading countries who are currently using the Robotic technology in various areas of their production. China is among the top followed by Japan. Instead of very advanced high technology of japan, China is among the top due to the establishment of the Robots in the manufacturing units like in chips, automobiles and other areas. Because of this china have the fastest growing economy in the world.

1.4 Robotic Application

There are various area in which more precision is required and the same is more difficult for the human beings. So we can use these Robots in such area like Agriculture, Space and underwater exploration etc.

Space Exploration

As in the case of MARS, the astronauts have to collect samples for the soils, also they have to repair the space craft. These situation may create difficulty for the astronauts which can be overcome by the use of Robots. NASA uses Robotics Vehicles for free roam and collecting samples on MARS.

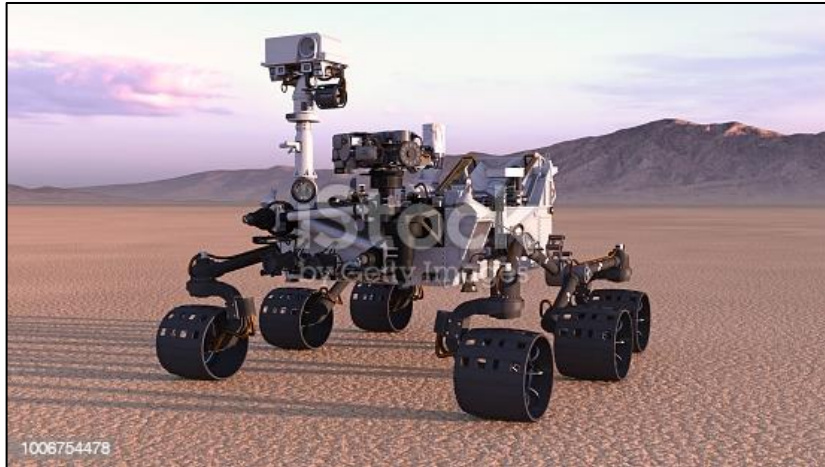


Figure 1.4.1 Space Robots.

Agriculture

Agriculture is the most important primary sector that provides a base to the economy of any country. It depends on the weather condition and the soil conditions. Robots can be used for harvesting the crops. The best example for removing weed in the farm is Eco-Robotix.

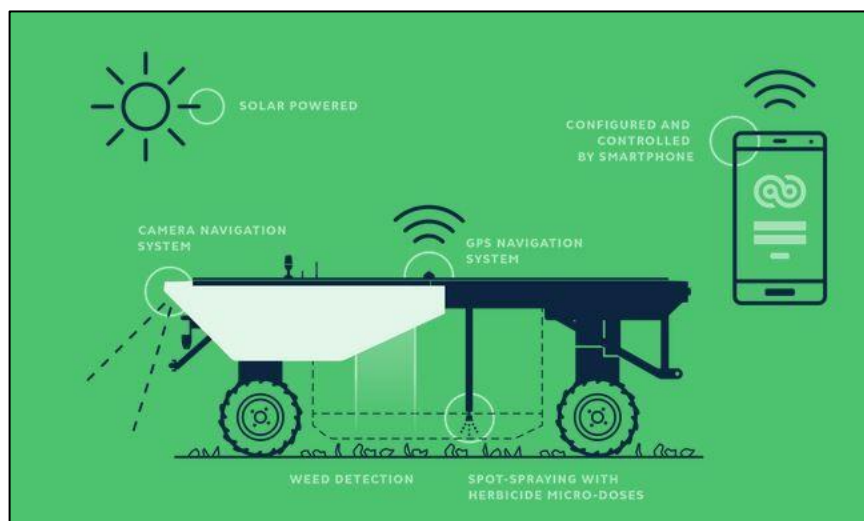


Figure 1.4.1 EcoRobotix

TECHNICAL COMPONENTS

2.1 CONTROL BOARD

2.1.1 Arduino-UNO

The controller which is used in this project is ARDUINO UNO. It is a microcontroller board which contain ATmega328p processor [4]. The pin out section as follows

- 1) 14 digital I/O pins.(out of which 16 can be used as PWM outputs)
- 2) 6 analog inputs.

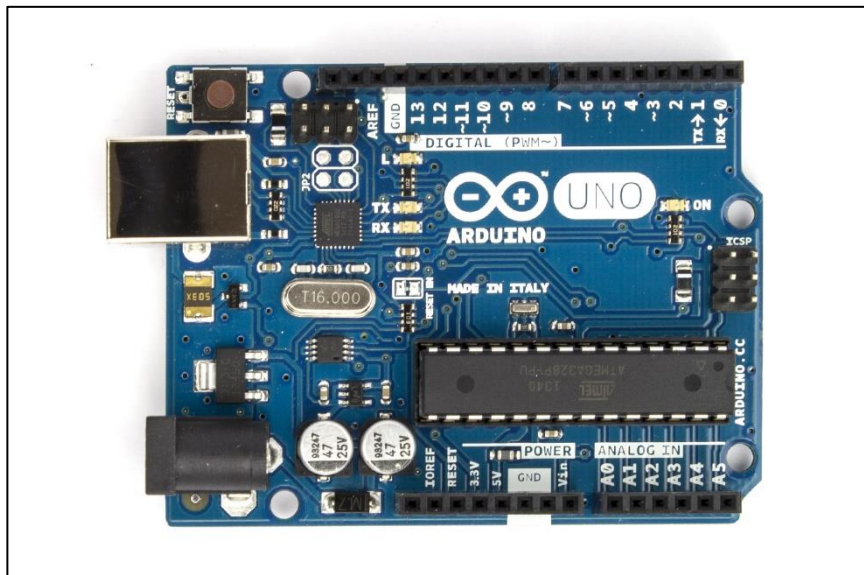


Figure 2.1.1 ARDUINO-UNO

It also contain 16MHz ceramic resonator, USB connection, power jack and a reset button. It must be connected to PC or some adapter for converting AC to DC for power. UNO means 1 in Italian.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Analog Input Pins	6
Digital I/O Pins	14
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Clock Speed	16 MHz

2.2.2 Arduino Nano

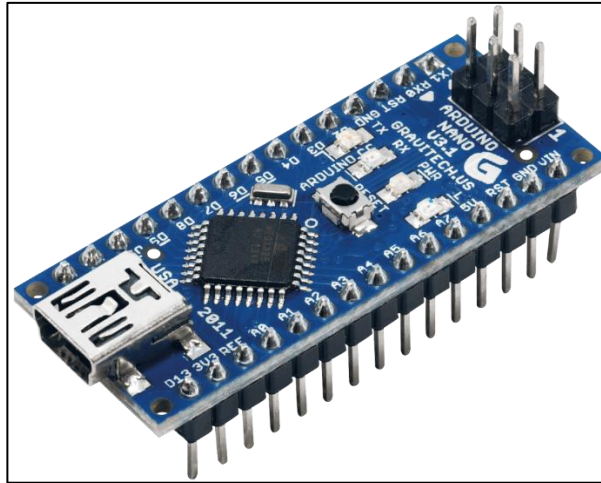


Figure 2.2.2 Arduino Nano

It has a Nano Board which consist of the Following-

- 1) 14 Digital I/O pin each with Current about 40mA.
- 2) 8 Analog pin.
- 3) 3V3 and 5V pin with Current about 50mA.
- 4) GND pin.
- 5) Microprocessor is ATmega 328p.

These pin which are specified are mainly used in this project. So this specification is completely related to the project. There are other pin also like Tx/Rx and RST, but those pin are not used in our project.

2.2 ACTUATOR

Servo in the robotic arm will work as actuator, similar to our muscles. These actuator help the arm of the robot to lift up and down as well as carry weight. This is the major component in the Robotic arm. There are various servo motor available in the market but we have to choose one of them according to the need.

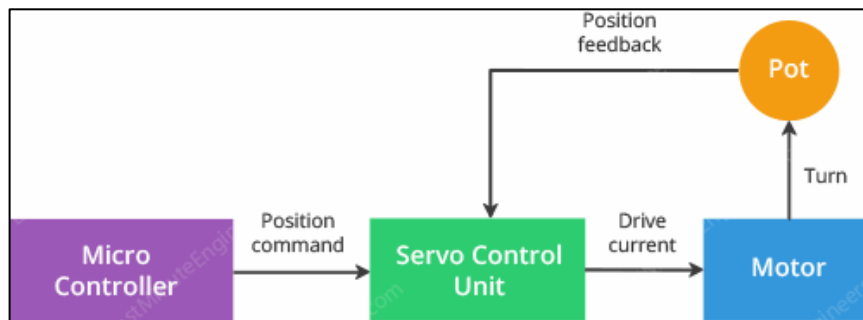


Figure 2.2.1 Block diagram of Servo Motor

Servo means the closed loop control system. These closed loop means the feedback loop for detecting and correcting error to achieve the desired result.

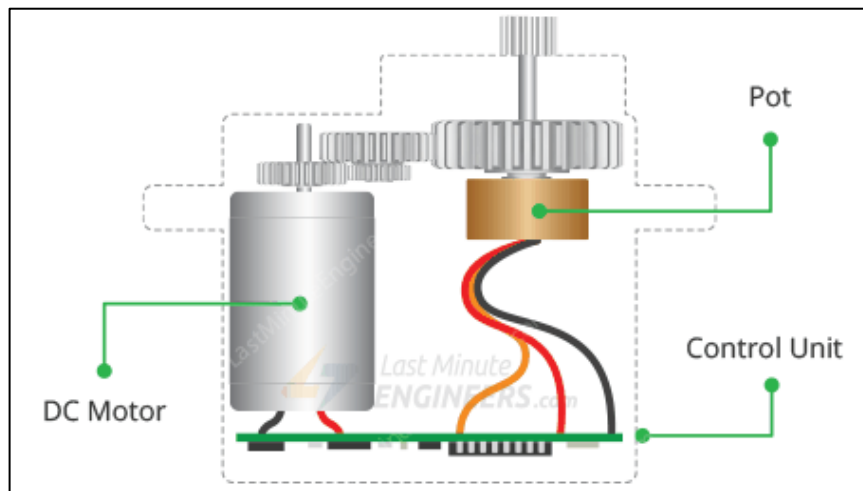


Figure 2.2.2 Servo motor internal diagram.

Working Principle - It has a dc motor which will convert a direct current into mechanical motion. A potentiometer is connected to provide a feedback to the control unit about the current position of the

shaft of the servo motor. The control unit correct the error in the rotation of the shaft based on the feedback.

For choosing the servo motor in the Robotic arm, we must look at the torque and speed of the motor. There are two types of Motor used in this project as follows –

- 1) Mg90 mini servo [6].
- 2) MG995 Servo
- 3) Sg90 mini servo [5].



Figure 2.2.3 Mg90 servo

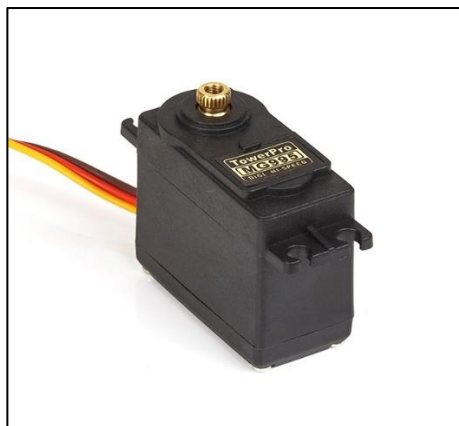


Figure 2.2.4 Mg995 Servo

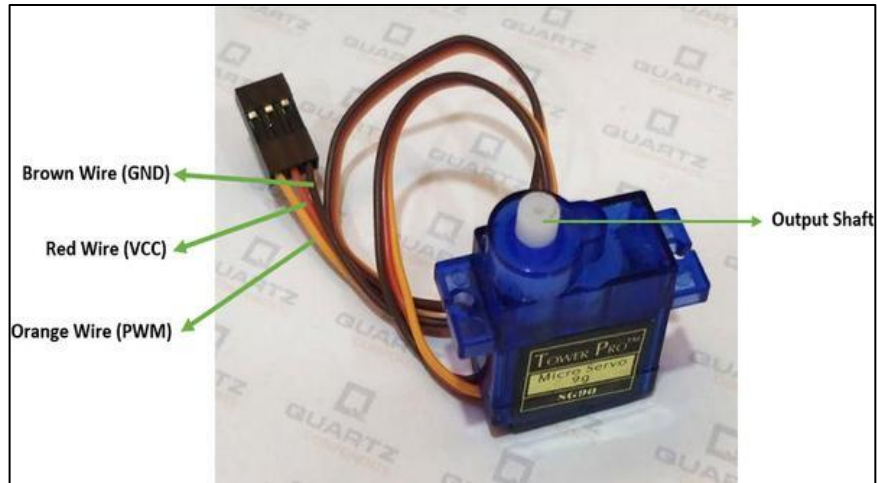


Figure 2.2.5 Sg90 servo

These servo have 3 Pins –

Vcc	-	Red wire
GND	-	Brown
PWM	-	Yellow/Orange

Vcc is used to connect the servo with the 5V supply or greater (depends on the type of servo). It is required to make the servo motor work for a particular task.

GND is the ground pin.

PWM stands for pulse width modulation. This signal through this pin have a total symbol width of 20ms (50Hz).

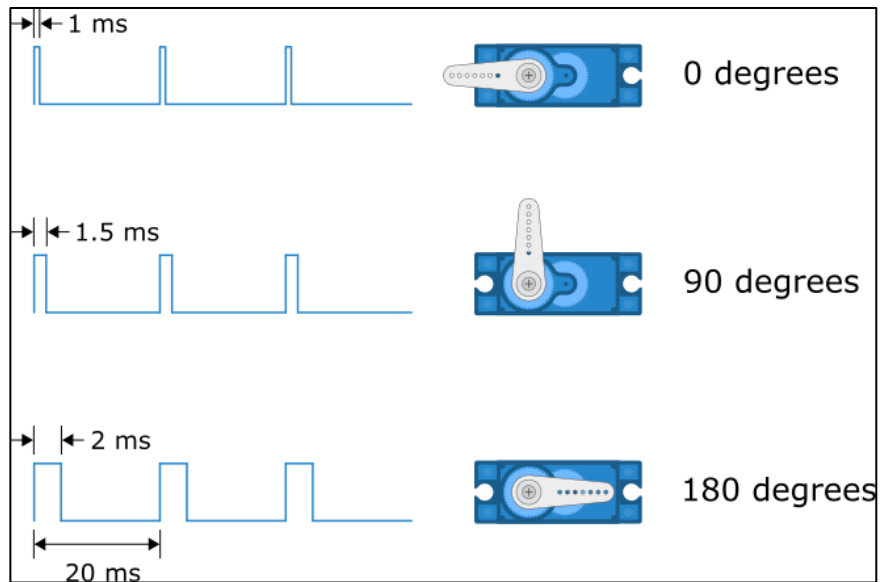


Figure 2.2.5 PWM signal

By controlling the PWM signal, we can easily control the position of the servo motor. The arduino has PWM pins connection through which it produces signal but the signal width is limited to 50Hz or 20ms.

Servo Interface with Arduino-UNO.

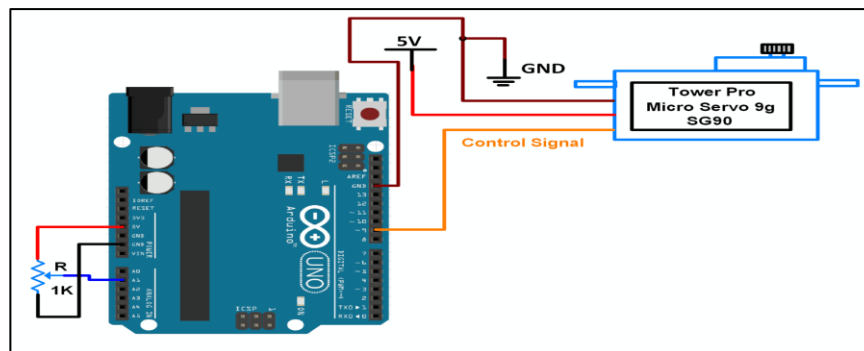


Figure 2.2.6 Circuit diagram

2.3 Sensor

There are two type of sensor that are used in this project –

- 1) Flex sensor.
- 2) MPU 6050 sensor

2.3.1 Flex sensor

Flex sensor is also called a blend sensor or it is flexible potentiometer. There are two sizes that are available in the flex sensor 2.2” and 4.5” (inches) [7].

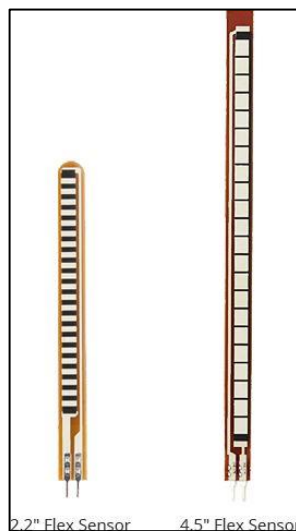


Figure 2.3.1.1 Flex sensor

Construction – flex sensor consist of phenolic resin substrate with conductive ink deposited on its surface. This conductive ink contain Segmented conductor on the top of it. It provides a flexible potentiometer in which resistance change upon bending.

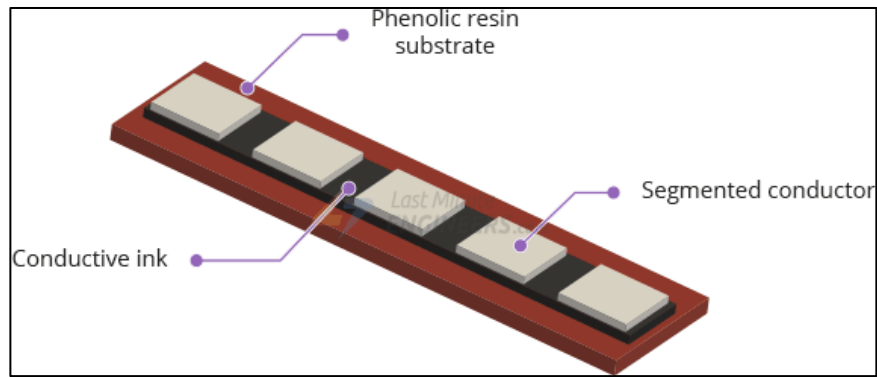


Figure 2.3.1.2 Component on Flex sensor.

Working – A resistance of the flex sensor will change according to the bend in the sensor.

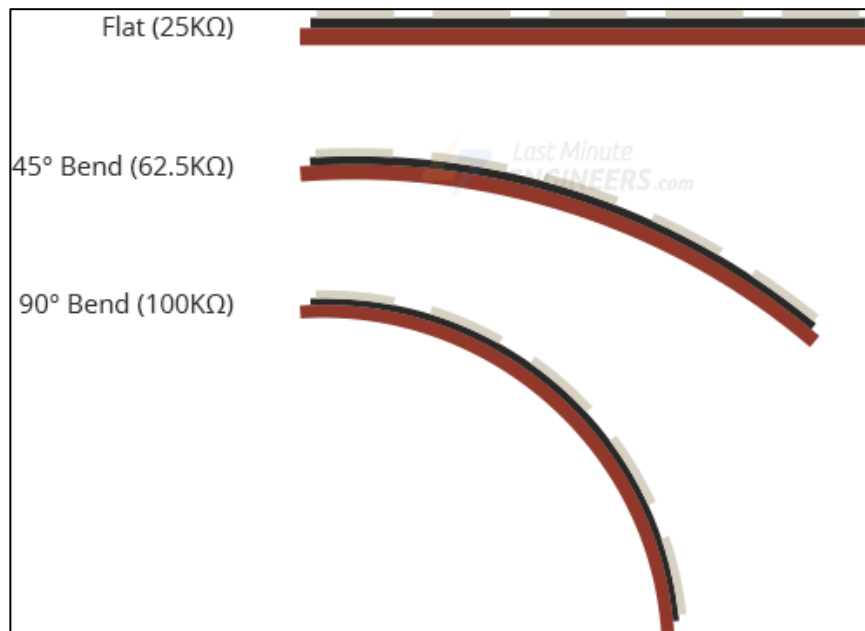


Figure 2.3.1.3 Bend Resistance.

2.3.2 MPU6050

MPU6050 module is MEMS which stands for Micro Electro- Mechanical System which provide 3 axis Accelerometer and 3 axis Gyroscope. This can be used to measure acceleration, orientation, velocity and displacement. We will use ROLL, PITCH and YAW values to determine the position and orientation of a body at 115200 or 9600 baud rate for the serial monitor. It has a default address 0x68 for the

communication b/w Arduino and MPU 6050. There are three 16 bit ADC converter for each Accelerometer and Gyroscope value [8].

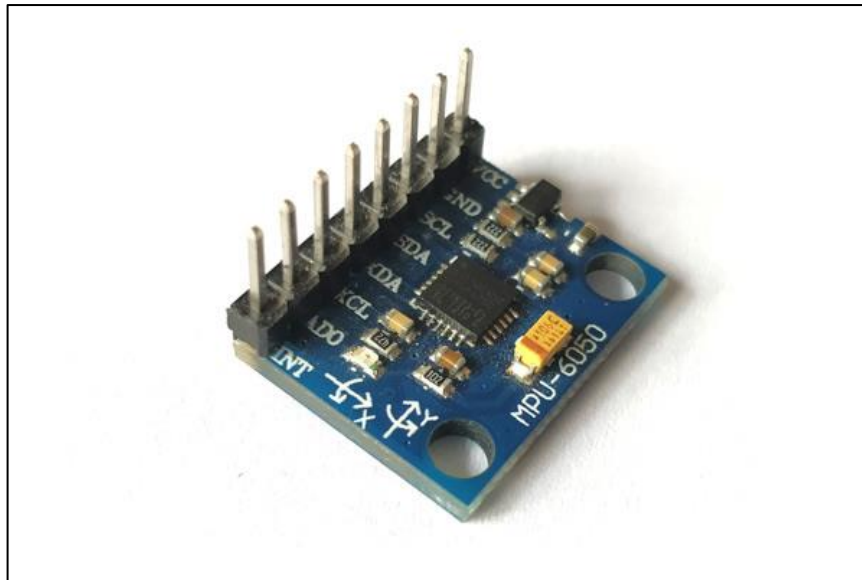


Figure 2.3.2.1 MPU6050.

The pin configuration are as follows –

VCC	-	+5V is used for power supply.
GND	-	GND of Arduino.
SCL	-	Serial clock pulse for I2C communication.
SDA	-	Serial Data communication with I2C.
XDA	-	Connect to other I2C module for data.
XCL	-	Connect to other I2C module for clock.
AD0	-	Pin can be used to vary default address.
INT	-	Interrupt pin for data.

Interfacing with ARDUINO-UNO

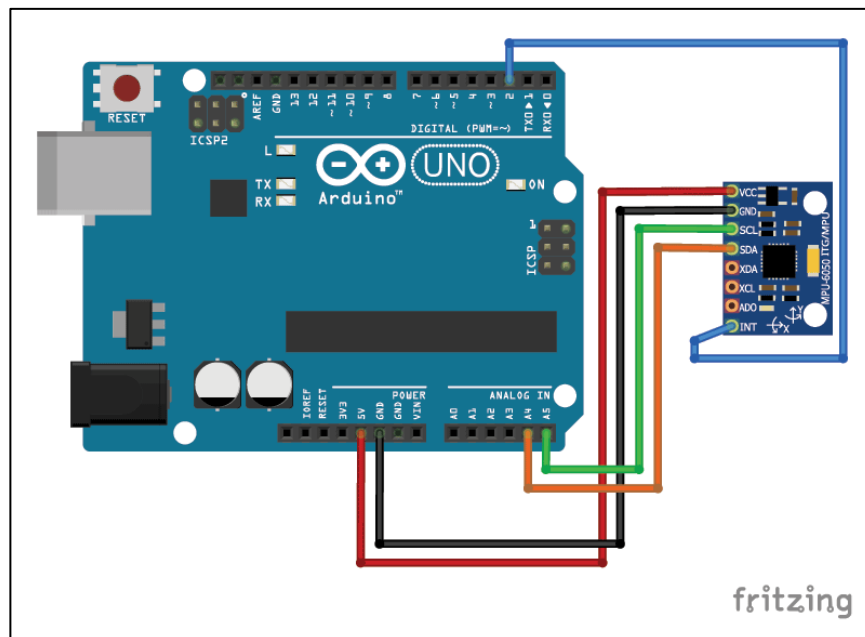


Figure 2.3.2.2 MPU6050 Circuit diagram

2.4 NRF24L01 Module

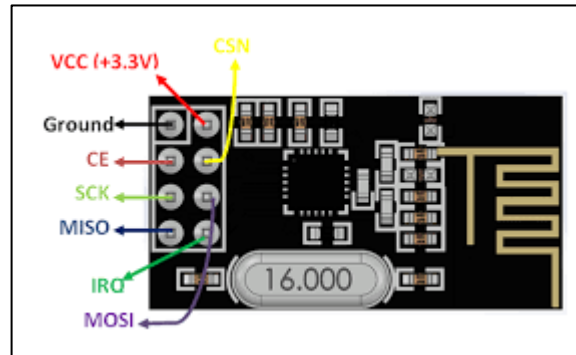


Figure 2.4.1 Nrf24L01

This Module [9] is used for the Wireless Communication between the two modules of same type. This can be used to transmit the data of MPU6050 and Flex sensor which is fitted in the Gloves to the Module which is on the wooden board. This module is connected to the Arduino at 3.3v source which draws a current about 12 mA during transmission. The module operates on the ISM band (Industrial Scientific and Medical band) which is mostly used by the lower power device for the unlicensed user. The operating Frequency is about 2.4 GHz. The module will communicate with the Arduino in the SPI interface. This have an advantage of transmitting the data with any limit in the data which has to be transferred. The configuration in which they are connected is like a Master-Slave Configuration.

2.4.1 Power Efficient

This module is very efficient for power consumption which consume only 12 mA during the transmission at 0 dBm. During Standby mode it consumes only 26 μ A.

2.4.2 Working

The module operates in Radio Frequency range. Transmission of data requires the channel b/w the two module. This channel is in range of about 2400- 2525 MHz , which mean that it have about 125 channel to communicate. In this project we have used 250 Kbps for uploading the data onto the channel. This information has bandwidth less than 1 MHz which make the air data rate 1 Mbps. It can use multiple module for transmitting and a single receiver. The protocol used by it as follows

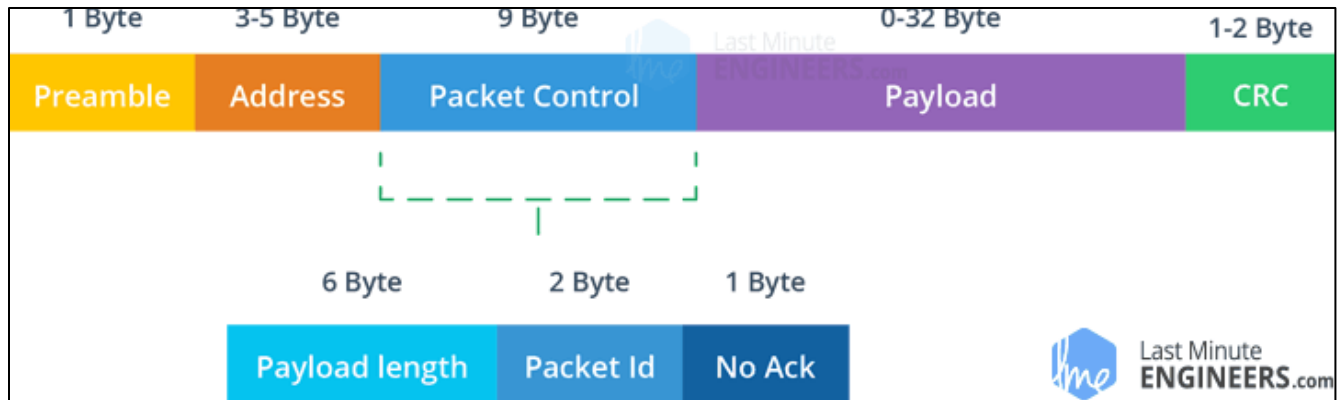


Figure 2.4.2 Enhanced Shock burst Protocol

- **Preamble** will determine the introduction of the protocol.
- **Address** will determine to which address the data is sent to.
- **Packet length** will determine and vary the length of the packet between 1-32 byte.
- **Packet ID** will give individual id to each packet for differentiation.
- **NO Acknowledgement** will be used for acknowledgement purpose to the transmitter to know whether the receiver received the packet or not.
- **Payload** is the actual information transmitted.
- **CRC** is Cyclic Redundancy check for determine the error in the information.

2.4.3 Summary

Operating Voltage	3.6V
Operating Current	13 mA
Standby Current	26 uA
Communication Range	800 m
Frequency Range	2.4 GHz
Modulation Format	GFSK

Table 2.4

2.4.4 Interfacing with Arduino

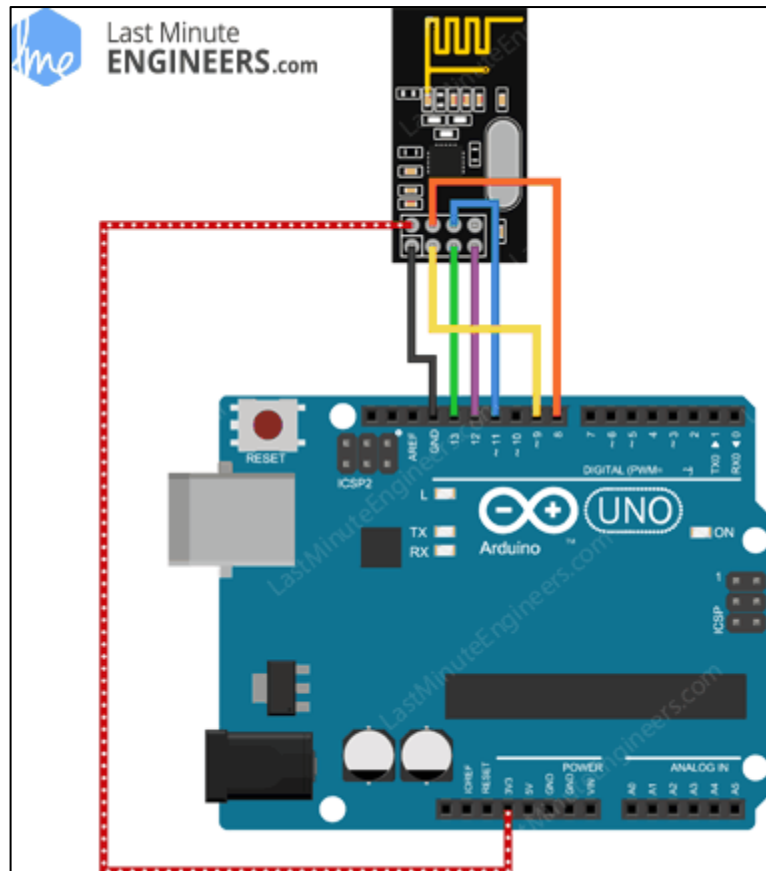


Figure 2.4.2 Nrf with Arduino

PROJECT DESIGN

3.1 ROBOTIC ARM

The Robotic Arm of this project consist of various component.

- 1) End Effector.
- 2) Actuator.
- 3) Sensor.
- 4) Controller.



Figure 3.1.1 Robotic Arm

End-Effector – It is the important component of the Robot that can be used to pick and place the object or thing around that Robot.

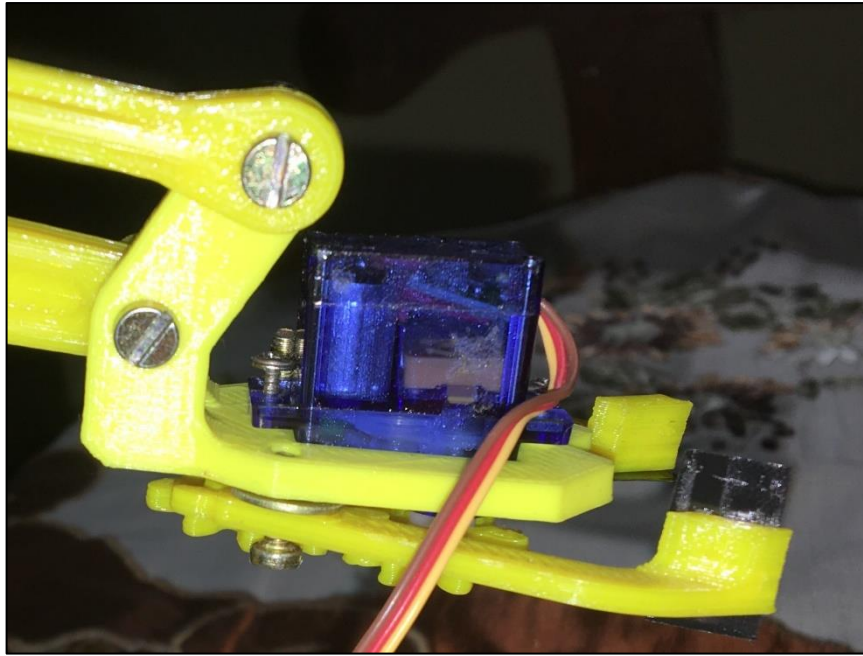


Figure 3.1.2 End Effector.

This End effector is attached with the link and joint to the Arm of the Robot. It consist of Sg90 mini servo that have **torque** of **1.3kg/cm** which can be controlled using the Flex sensor via Arduino. It shaft will rotate from 0 to 180 degree, when the motor is connected to 5V and having a current of 10mA (idle) and 100mA during the rotation. This S90 servo can have **dead bandwidth** of 8us.



Figure 3.1.3 Shaft and teeth.

Actuator – The arm of the Robot is used to lift the object picked with the end effector. The Arm can move in the upward / downward and Forward / Reverse direction. This can be controlled using the MPU

6050 sensor. For this coding in arduino is required to convert the values obtained from the signal and then converted into the degree.



Figure 3.1.4 Robotic Arm back and Side view.

This Arm contain Mg 90 tower pro servo which has a **torque of 1.8kgf-cm** and operated at a voltage of 5V. Each side of Robot consist of Mg90 servo with the torque as follows

*Torque due to link 1 = (mass of Servo * length of the link) + (stationary link weight * stationary link)*

$$\text{Torque (link1)} = 9g * 7.5 \text{ cm} + 10 * 8.5 = 153 \text{ gf-cm}$$

*Torque due to link 2 = (dist,) mass of Servo * length of the link*

$$\text{Torque (link 2)} = 4.5g * 15.5 \text{ cm} = 70 \text{ gf-cm}$$

NOTE –This torque is only a rough approximation of the Robotic arm because we did not consider link and joint weight.

The exact calculation is difficult due to the mass is being distributed. But this will give an idea for the calculation of the torque and also for choosing the servo motor.

Sensor

This project uses 2- types of sensor

- 1) Flex sensor.
- 2) MPU 6050.

Already discussed in the component specification.

Controller

Arduino is used as the controller for processing all the value of the MPU 6050, Flex sensor and Servo to provide proper functioning of each component within the arm of the Robot. Arduino is already discussed in the component specification.

3.2 Control through Hand Gesture

In this we attach MPU 6050 and flex sensor on the Gloves. By moving the hand in any direction will causes the movement in the arm of the Robot.

LEFT/RIGHT – This left and Right motion in the Robotic arm can be done by tilting the hand in the Right and left direction respectively.

Accuracy – In this case there can be a delay of **1 second** in the motion of the Arm in the respective direction. So to reduce the delay, we have to tilt the Gloves slowly.

FORWARD/REVERSE – This forward and Reverse motion can be done by tilting the sensor in the forward and the Reverse direction respectively.

Accuracy – If we move the hand very frequently then there should be delay and the Arm will not move. The sensor will send too much data to the arduino which causes the motor to jam. So to minimize this issue we need to move the arm slowly so that the data could be properly processed by the arduino.

END EFFECTOR – The end effector will move by bending the flex sensor. So that the motor on the End effector will rotate the gear attached to it.



Figure 3.2.1 MPU6050 on glove



Figure 3.2.2 Flex sensor attached to Glove

3.3 ROBOTIC ARM (VERSION – 2)

This Robotic Arm structure consist of 3 - MG 995 servo motor which can be controlled with the help of 3 – Flex sensor. These Flex Sensor must be fiited on the gloves. Each flex sensor must have different function to perform as follows –

1. **Flex-1** on forefinger can be used to control the gripper motor to grip the object.
2. **Flex-2** on the middle finger can be used to control the upward / downward motion of the Robotic Arm with servo motor.
3. **Flex-3** on the Ring fingure can be used to contol the Left / Right motion of the body of the Robotic Arm.



Figure 3.3.1 Side View

3.3.1 Accuracy

The accuracy of this model is very less because while bending one flex sensor, the other flex sensor also bends which causes movement in other servo motor also. This created a problem in holding up of an object precisely. Also due to variation in the value of the flex sensor continously, the motor used to move or give a error of about 3 to 4 degree.

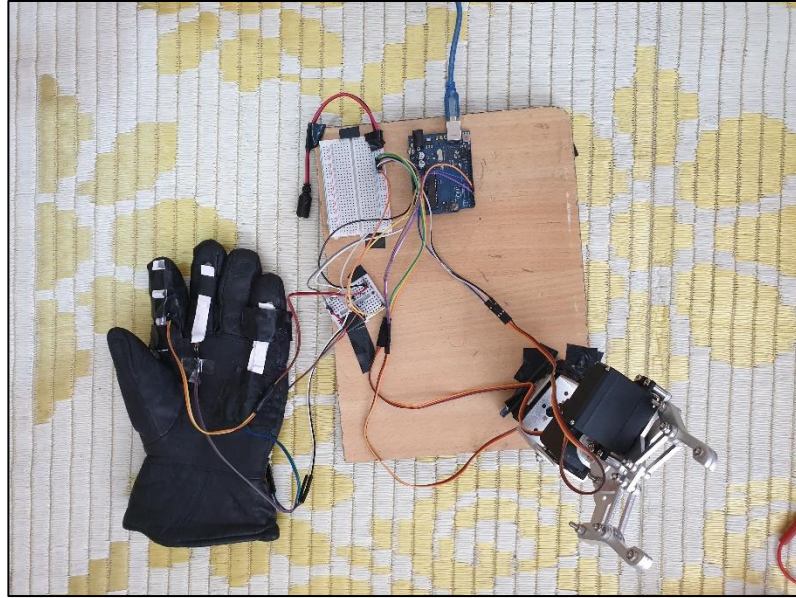


Figure 3.3.2 Top View

3.4 ROBOTIC ARM WIRELESS (VERSION – 3)

In this version, The performance and accuracy of both the version is combined. Completely depending on either MPU6050 or flex sensor will causes more problem in controlling the servo motor. So the combination use of MPU6050 with the flex sensor to control the servo motor can give it more flexibility. Also this time the connect is much more wireless with the use of NRF24L01 module which can be used to transmit the data from arduino nano to Arduino Uno. The Arduino nano is connect to collect the information from the flex and Mpu6050 sensor. This information is transmitted to the arduino uno which is connect to the servo motor for controlling the whole arm of the robot.

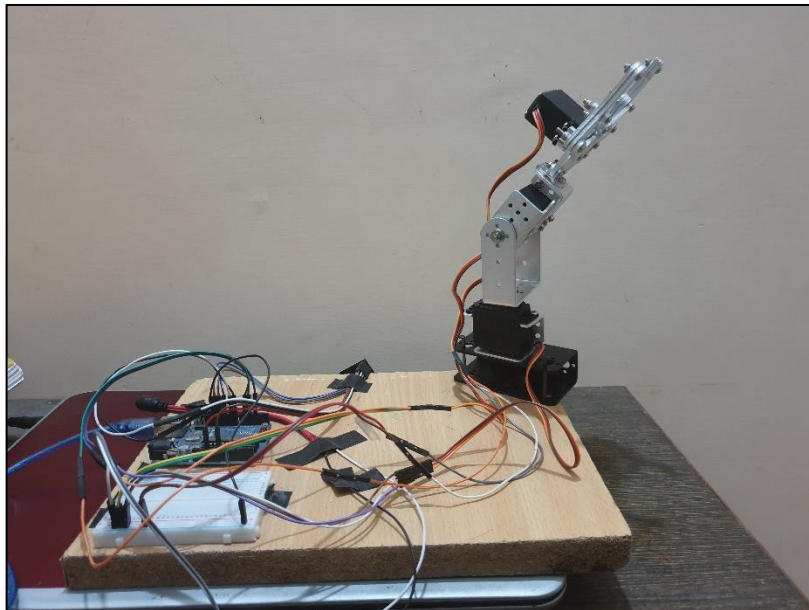


Figure 3.4.1 Side View

3.4.1 Experimental Analysis

Coding this type of model will require pre requisite knowledge of servo rotation upto a desired angle well calibrated with flex and mpu6050 sensor. The coding part also require some knowledge of wireless communication and nrf R24 library which uses certain command “ *radio.openWritingPipe()* “ and “*radio.openReadingPipe()*”.

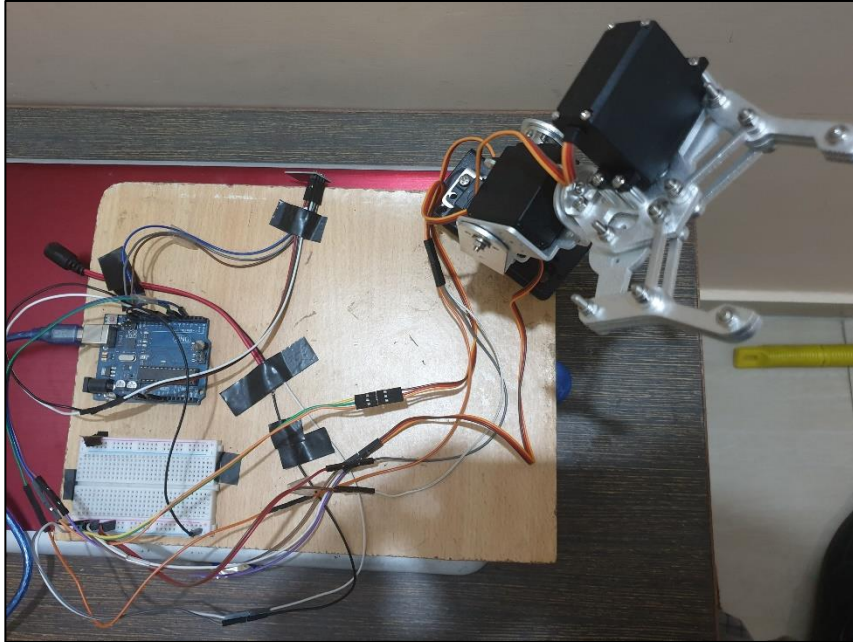


Figure 3.4.2 Top View

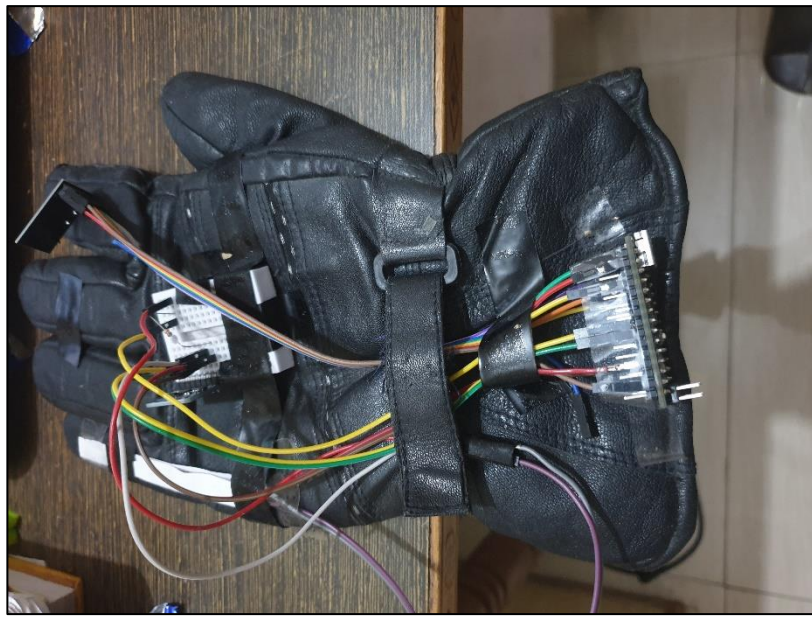


Figure 3.4.3 Hand Glove

ADDITIONAL COMPONENTS

1) Breadboard.

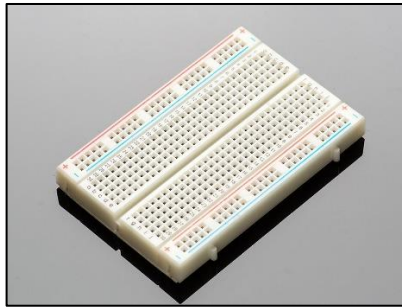


Figure 4.1 Breadboard

2) Jumper wire.

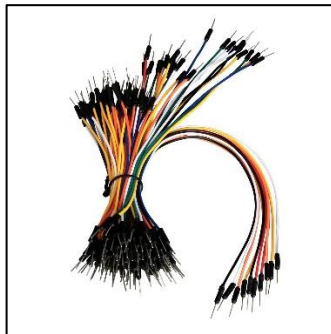


Figure 4.2 Wire

3) Two - 3.7V battery.



Figure 4.3 3.7V Battery

4) 7805 IC voltage regulator – It is used in this project to regulate the 7.4V battery or greater. Because the Arduino is not capable of handling 4 servo motor at the same time. So we require external 5V supply for proper functioning of the servo [10].



Figure 4.4 7805 IC

CONCLUSION

With help of the Flex and MPU 6050 sensor we can control the servo motor of the End effector and the Arm of the Robot. These device must be properly connected so that it can be perform very efficiently. This project can also be used for pick and place of the object and required a microcontroller to function properly. This is only a prototype to bring the clear objective and the application of this type of Robotic Arm. In actual it can be very costly and we require good amount of actuator and joints for creating a 6 Degree of Freedom Robot.

Accuracy

For accuracy in movement we can use Digital servo Motor.

Reason

Producing torque from rest mode makes the initial time sluggish. -*Analog servo.*

- Due to small microprocessor to receive and direct action at high Frequency. -*Digital servo.*
- Digital servo can send 6 times tha amount of pulse.
- Due to high frequency of signal the digital servo's output doesn't lag.
- Operating Voltage 4.8-6.6V.
- Stall torque 18kg-cm.
- Metal gear.

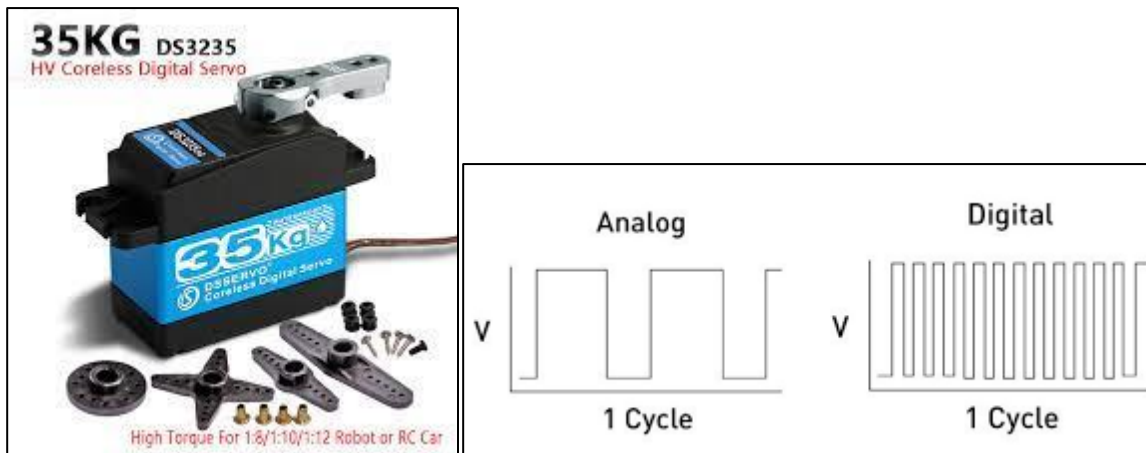


Figure 5.1 Digital Servo

REFERENCE

1. Robotics Definition - <https://www.britannica.com/technology/robot-technology>
2. LAWS - <https://www.scientificamerican.com/article/asimovs-laws-wont-stop-robots-from-harming-humans-so-weve-developed-a-better-solution/>
3. IFR report - <https://ifr.org/free-downloads/>
4. <https://www.farnell.com/datasheets/1682209.pdf> - ARDUINO UNO datasheet.
5. http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf – Sg90 servo datasheet.
6. https://www.electronicoscaldas.com/datasheet/MG90S_Tower-Pro.pdf - Mg90 servo datasheet.
7. Flex sensor datasheet - <https://www.spectrasymbol.com/wp-content/uploads/2019/07/flexsensordatasheetv2019revA.pdf>
8. MPU -6050 sensor datasheet - <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf>
9. NRF24L01 module - https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/#google_vignette
10. 7807 IC - <https://www.electronicsforu.com/technology-trends/learn-electronics/7805-ic-voltage-regulator>

APPENDIX

CODE – 1

//Code for Gesture Controlled Robotic ARM (Arduino UNO & MPU6050)

//Circuit Digest

#include<Wire.h> //I2C Wire Library

#include<Servo.h> //Servo Motor Library

Servo servo_1;

Servo servo_2;

Servo servo_3;

Servo servo_4;

const int MPU_addr=0x68; //MPU6050 I2C Address

int16_t axis_X,axis_Y,axis_Z;

int minVal=265;

int maxVal=402;

int a;

double x;

double y;

double z;

```

void setup()

{

  Serial.begin(9600);


  Wire.begin();           //Initilize I2C Communication

  Wire.beginTransmission(MPU_addr); //Start communication with MPU6050

  Wire.write(0x6B);       //Writes to Register 6B

  Wire.write(0);          //Writes 0 into 6B Register to Reset

  Wire.endTransmission(true); //Ends I2C transmission


  servo_1.attach(3); // Forward/Reverse_Motor

  // servo_2.attach(5); // Up/Down_Motor

  servo_3.attach(6); // Gripper_Motor

  servo_4.attach(9); // Left/Right_Motor

}


void loop()

{


  Wire.beginTransmission(MPU_addr);

  Wire.write(0x3B);       //Start with regsiter 0x3B

  Wire.endTransmission(false);

  Wire.requestFrom(MPU_addr,14,true); //Read 14 Registers

```

```

axis_X=Wire.read()<<8|Wire.read();           //Reads the MPU6050 X,Y,Z AXIS Value

axis_Y=Wire.read()<<8|Wire.read();

axis_Z=Wire.read()<<8|Wire.read();


int xAng = map(axis_X,minVal,maxVal,-90,90);   // Maps axis values in terms of -90 to +90

int yAng = map(axis_Y,minVal,maxVal,-90,90);

int zAng = map(axis_Z,minVal,maxVal,-90,90);


x= RAD_TO_DEG * (atan2(-yAng, -zAng)+PI);      //Formula to convert into degree

y= RAD_TO_DEG * (atan2(-xAng, -zAng)+PI);

z= RAD_TO_DEG * (atan2(-yAng, -xAng)+PI);


int flex_sensorip = analogRead(A0);           //Reads flex sensor output


a= map(flex_sensorip,785,880,0,180);

servo_3.write(a);

delay(1000);


//Writes gripper value to 3rd servo motor


Serial.println(flex_sensorip);

```

```
if(x >=0 && x <= 60)

{

    int mov1 = map(x,0,60,0,180);

    Serial.print("Movement in F/R = ");

    Serial.print(mov1);

    Serial.println((char)176);

    servo_1.write(mov1);

}
```

```
/* else if(x >=300 && x <= 360)

{

    int mov2 = map(x,360,300,0,90);

    Serial.print("Movement in Up/Down = ");

    Serial.print(mov2);

    Serial.println((char)176);

    servo_2.write(mov2);

    delay(1000);

}*/
```

```
if(y >=0 && y <= 60)

{

    int mov3 = map(y,0,60,90,180);

    Serial.print("Movement in Left = ");

    Serial.print(mov3);

    Serial.println((char)176);

}
```

```
servo_4.write(mov3);  
  
}  
  
else if(y >=300 && y <= 360)  
{  
    int mov3 = map(y,360,300,90,0);  
    Serial.print("Movement in Right = ");  
    Serial.print(mov3);  
    Serial.println((char)176);  
    servo_4.write(mov3);  
}  
}
```

CODE – 2

Transmitter Code

```
#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

#include <Wire.h>


//create an RF24 object

RF24 radio(9, 8); // CE, CSN


//address through which two modules communicate.

const byte address[6] = "00001";

const int MPU_addr=0x68;    //MPU6050 I2C Address

int16_t axis_X,axis_Y,axis_Z;

int minVal=265;

int maxVal=402;


const int flexPin=A0;

int flexValue;


//double x;

//double y;

double z;

double text[3];
```

```

void setup()

{

  Serial.begin(9600);

  radio.begin();


  //set the address

  radio.setDataRate(RF24_250KBPS);

  radio.openWritingPipe(address);


  //Set module as transmitter

  //radio.stopListening();


  Wire.begin();           //Initilize I2C Communication

  Wire.beginTransmission(MPU_addr); //Start communication with MPU6050

  Wire.write(0x6B);       //Writes to Register 6B

  Wire.write(0);          //Writes 0 into 6B Register to Reset

  Wire.endTransmission(true); //Ends I2C transmission
}

void loop()

{

```

```
//Data from MPU6050 for X,Y,Z coordinates
```

```
Wire.beginTransmission(MPU_addr);
```

```
Wire.write(0x3B);           //Start with register 0x3B
```

```
Wire.endTransmission(false);
```

```
Wire.requestFrom(MPU_addr,14,true);    //Read 14 Registers
```

```
axis_X=Wire.read()<<8|Wire.read();    //Reads the MPU6050 X,Y,Z AXIS Value
```

```
axis_Y=Wire.read()<<8|Wire.read();
```

```
axis_Z=Wire.read()<<8|Wire.read();
```

```
int xAng = map(axis_X,minVal,maxVal,-90,90);    // Maps axis values in terms of -90 to +90
```

```
int yAng = map(axis_Y,minVal,maxVal,-90,90);
```

```
int zAng = map(axis_Z,minVal,maxVal,-90,90);
```

```
text[0]= RAD_TO_DEG * (atan2(-yAng, -zAng)+PI);    //Formula to convert into degree
```

```
text[1]= RAD_TO_DEG * (atan2(-xAng, -zAng)+PI);
```

```
z= RAD_TO_DEG * (atan2(-yAng, -xAng)+PI);
```

```
//Reading Flex Value from Arduino Nano
```

```
flexValue= analogRead(flexPin);
```



```
Serial.println("Flex Value is:");  
  
if(flexValue>957){  
  
    text[2]=1;  
  
}  
  
else{  
  
    text[2]=0;  
  
}  
  
Serial.println(text[2]);
```

```
//Sending Data over Nrf Module
```

```
  
radio.write(text, sizeof(text));  
  
Serial.println("X value is ");  
  
Serial.println(text[0]);  
  
Serial.println("Y vlaue is ");  
  
Serial.println(text[1]);  
  
}
```

Reciever Code –

```
//Include Libraries  
  
#include <SPI.h>  
  
#include <nRF24L01.h>  
  
#include <RF24.h>
```

```
#include<Servo.h>           //Servo Motor Library
```

```
Servo servo_1;
```

```
Servo servo_2;
```

```
Servo servo_3;
```

```
int x;
```

```
int y;
```

```
int Flex;
```

```
//create an RF24 object
```

```
RF24 radio(9, 8); // CE, CSN
```

```
//address through which two modules communicate.
```

```
const byte address[6] = "00001";
```

```
double text[3];
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600);
```

```
    radio.begin();
```

```
    radio.setDataRate(RF24_250KBPS);
```

```
//set the address

radio.openReadingPipe(1, address);


//Set module as receiver

radio.startListening();


servo_1.attach(3); // Forward/Reverse_Motor

servo_2.attach(5); // Gripper_Motor

servo_3.attach(6); // Left/Right_Motor
}


void loop()

{

//Read the data if available in buffer

if (radio.available())

{


radio.read(text, sizeof(text));

x=text[0];

y=text[1];

Flex=text[2];

Serial.println(text[0]);

Serial.println(text[1]);

Serial.println(text[2]);
```

```
// MPU6050 Code
```

```
if(x >=0 && x <= 60)
```

```
{
```

```
int mov1 = map(x,0,60,60,0);
```

```
Serial.print("Movement in F/R = ");
```

```
Serial.print(mov1);
```

```
Serial.println((char)176);
```

```
servo_1.write(mov1);
```

```
}
```

```
if(y >=0 && y <= 45)
```

```
{
```

```
int mov3 = map(y,45,0,90,0);
```

```
Serial.print("Movement in Right = ");
```

```
Serial.print(mov3);
```

```
Serial.println((char)176);
```

```
servo_3.write(y);
```

```
}
```

```
else if(y >=300 && y <= 345)
```

```
{
```

```
int mov3 = map(y,345,300,90,180);
```

```
    Serial.print("Movement in left = ");

    Serial.print(mov3);

    Serial.println((char)176);

    servo_3.write(mov3);

}

// flex sensor

if(Flex==1){

    servo_2.write(180);

}

else{

    servo_2.write(150);

}

}

else{

    Serial.println("No radio available");

}

delay(500);

}
```

GESTURE CONTROLLED ROBOTIC ARM

ORIGINALITY REPORT

22%

SIMILARITY INDEX

22%

INTERNET SOURCES

8%

PUBLICATIONS

22%

STUDENT PAPERS

PRIMARY SOURCES

1

circuitdigest.com

Internet Source

13%

2

Submitted to Sogang University

Student Paper

2%

3

forum.arduino.cc

Internet Source

2%

4

lastminuteengineers.com

Internet Source

1%

5

Submitted to Middle East College of
Information Technology

Student Paper

1%

6

news.jevtonline.org

Internet Source

1%

7

Submitted to Jaypee University of Information
Technology

Student Paper

1%

8

Submitted to Nelson Mandela Metropolitan
University

Student Paper

1%