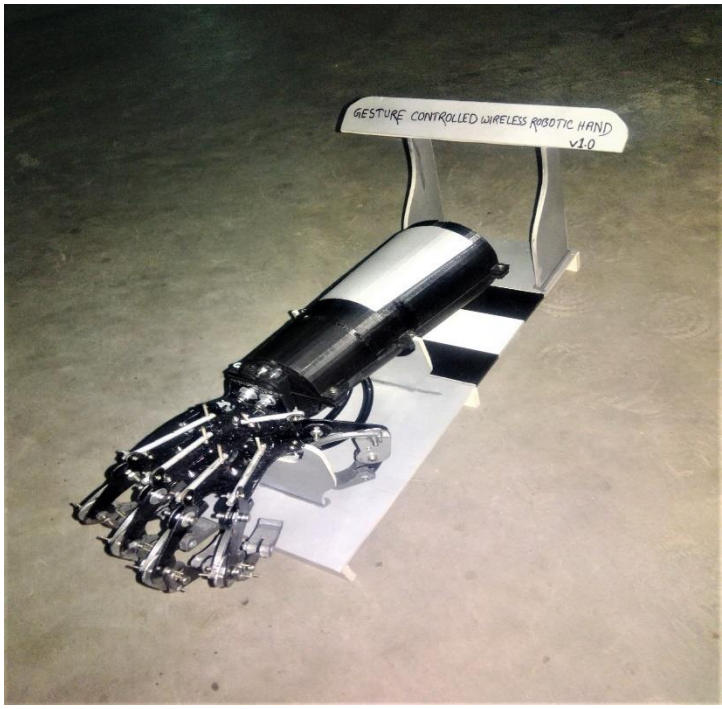
Course No. : ME 362

Instrumental and Measurement Sessional



Group : B_{13}

Khokon Chandra Sazzal	1510063
Enamul Hasan Rozin	1510070
Hasib Ahmed Prince	1510077
Walid hasan	1510084

GESTURE CONTROLLED WIRELESS ROBOTIC HAND

DEPARTMENT OF MECHANICAL ENGINEERING

BANGLADESH UNIVERSITY OF ENGINEERING & TECHNOLOGY

Table of Contents

Introduction

In today's life automation plays very important role. Robotic arm is called as robot manipulator which can perform various functions as human arm performs. Many industries use a robot for various functions where important part of any robot is Robotic arm or called as robot manipulator should be controlled precisely depending upon application. In industry or any application robot manipulator can be used for applications like welding, trimming; picking etc. advantage of such robotic arm is it can work in hazards area, which cannot be accessed by human. Many parameters of robot are designed according to requirement. There are different ways to control robotic arm like Voice Controlled, Keypad Control, Gesture Control, etc. Implemented system consists of transmitter & receiver. Transmitter is nothing but human hand with flex sensors & receiver is robot manipulator. Motion of transmitter is wirelessly transmitted to receiver through X-bee module. Robotic arm which is receiver is nothing but a mechanical system formed by different joints and end and effectors i.e. gripper movements of these fingers or gripper can be carried out using stepper motor or servo motor when user carry out motion of hand for any application at transmitter side same movement is copied by receiver as on transmitter there are flex sensors mounted on glove at transmitter which change its resistance depending on movement of user

Components

Figure 1:Mechanical Hand

1.Arduino Uno : Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button

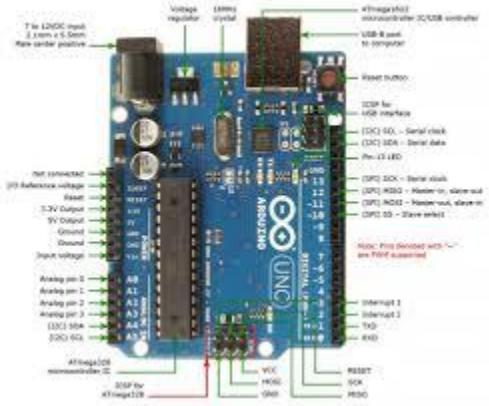


Figure 2:Arduino Uno

2.servo motor (x 5): A **servomotor** is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration.[1] It consists of a suitable motor coupled to a sensor for position feedback

.

Applications: Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

4.Radio link (nRf2401):

NRF24L01 transceiver module uses the 2.4 GHz band and it can operate with baud rates from 250 kbps up to 2 Mbps. If used in open space and with lower baud rate its range can reach up to 100 meters.

3.Veroboard

4.lots of Nut & Bolt

Gloves with Flex Sensor

1.Flex Sensor : A **flex sensor** or **bend sensor** is a sensor that measures the amount of deflection or bending. Usually, the sensor is stuck to the surface, and resistance of sensor element is varied by bending the surface. Since the resistance is directly proportional to the amount of bend it is used as goniometer, and often called flexible potentiometer.

Figure : Flex Sensor

2.Arduino Nano: The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package.

Arduino Nano V3.0

It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one.The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated

external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

3.Gyro-sensor: In recent years vibration gyro sensors have found their way into camera-shake detection systems for compact video and still cameras, motion sensing for video games, and vehicle electronic stability control (anti-skid) systems, among other things.

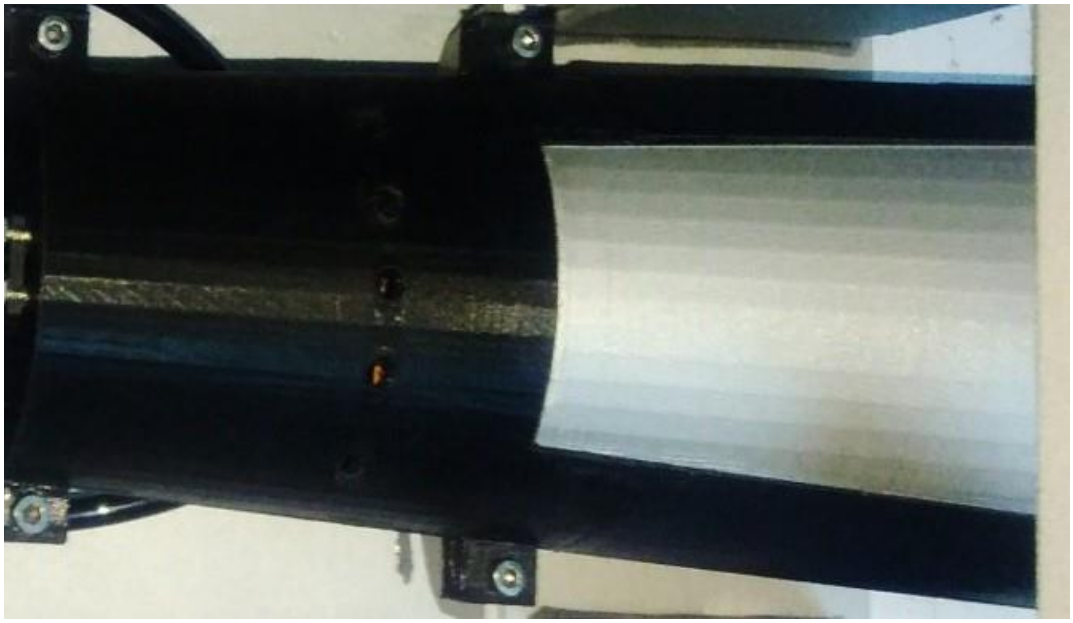
4.Hand Gloves.

5.Radio link (nRf 2401l).

6.Bread Board .

Materials

1. 3d printed_plastic :

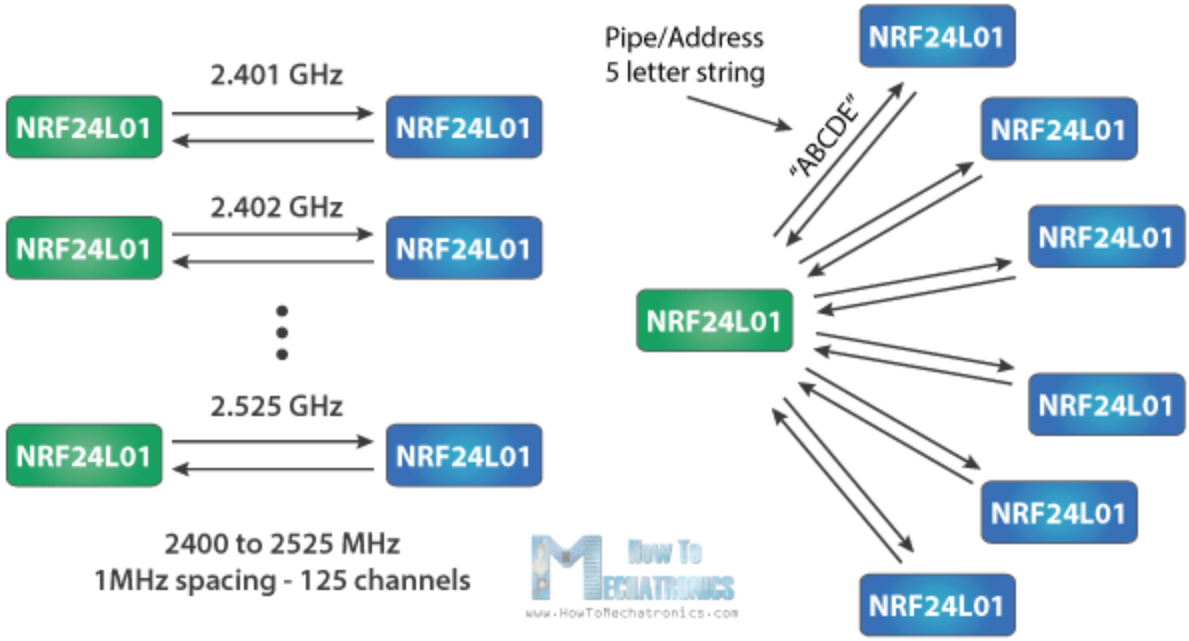


2. stainless steel:

Working Principle

nRf 2401l :

The module can use 125 different channels which gives a possibility to have a network of 125 independently working modems in one place. Each channel can have up to 6 addresses, or each unit can communicate with up to 6 other units at the same time.

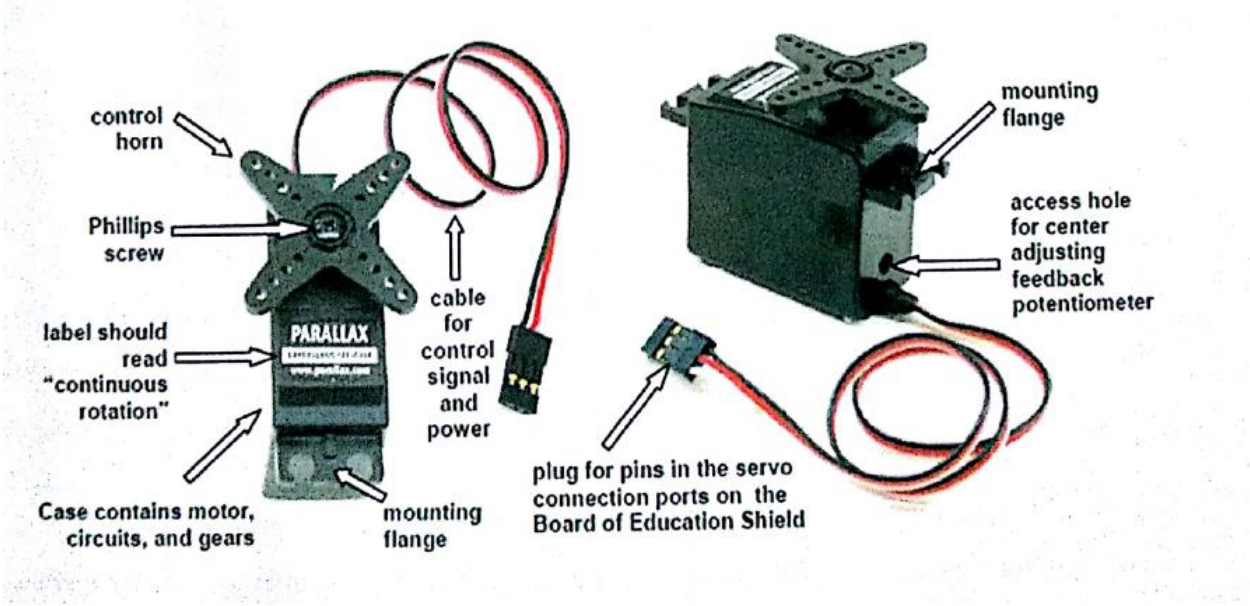


The power consumption of this module is just around 12mA during transmission, which is even lower than a single LED. The operating voltage of the module is from 1.9 to 3.6V, but the good thing is that the other pins tolerate 5V logic, so we can easily connect it to an Arduino without using any logic level converters.

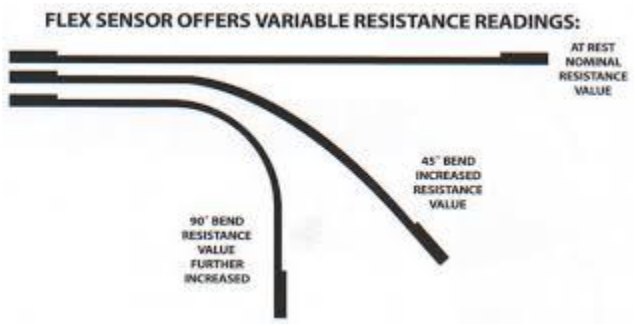
Servo motor :

. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

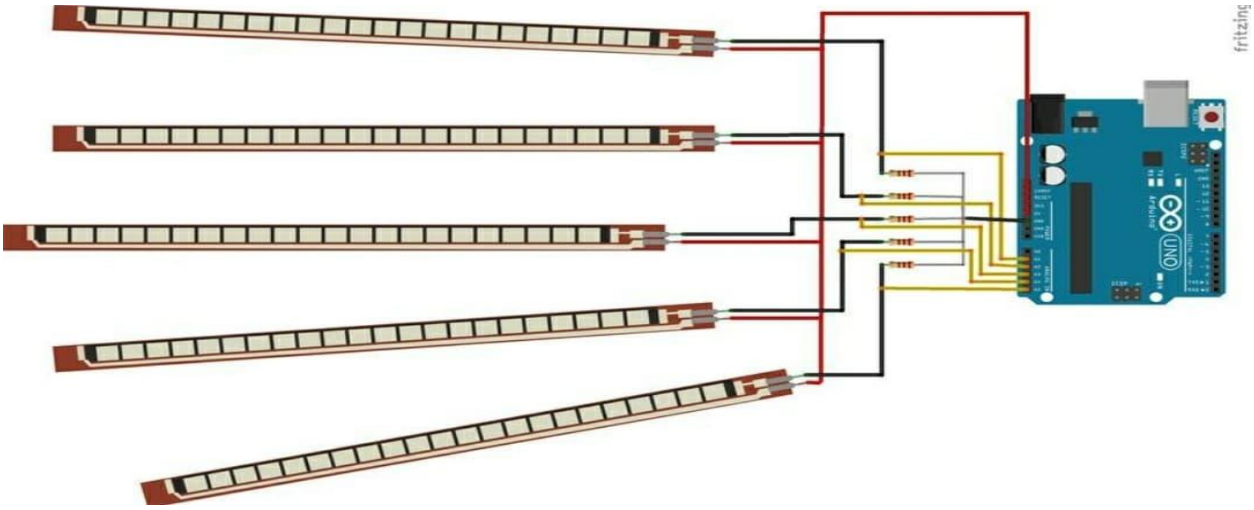
Servomotors are not a specific class of motor although the term *servomotor* is often used to refer to a motor suitable for use in a closed-loop control system



Flex sensor: One side of the **sensor** is printed with a polymer ink that has conductive particles embedded in it. When the **sensor** is straight, the particles give the ink a resistance of about 30k Ohms. ... By measuring the resistance, you can determine how much the **sensor**is being bent.



Circuit Diagram And Setup :



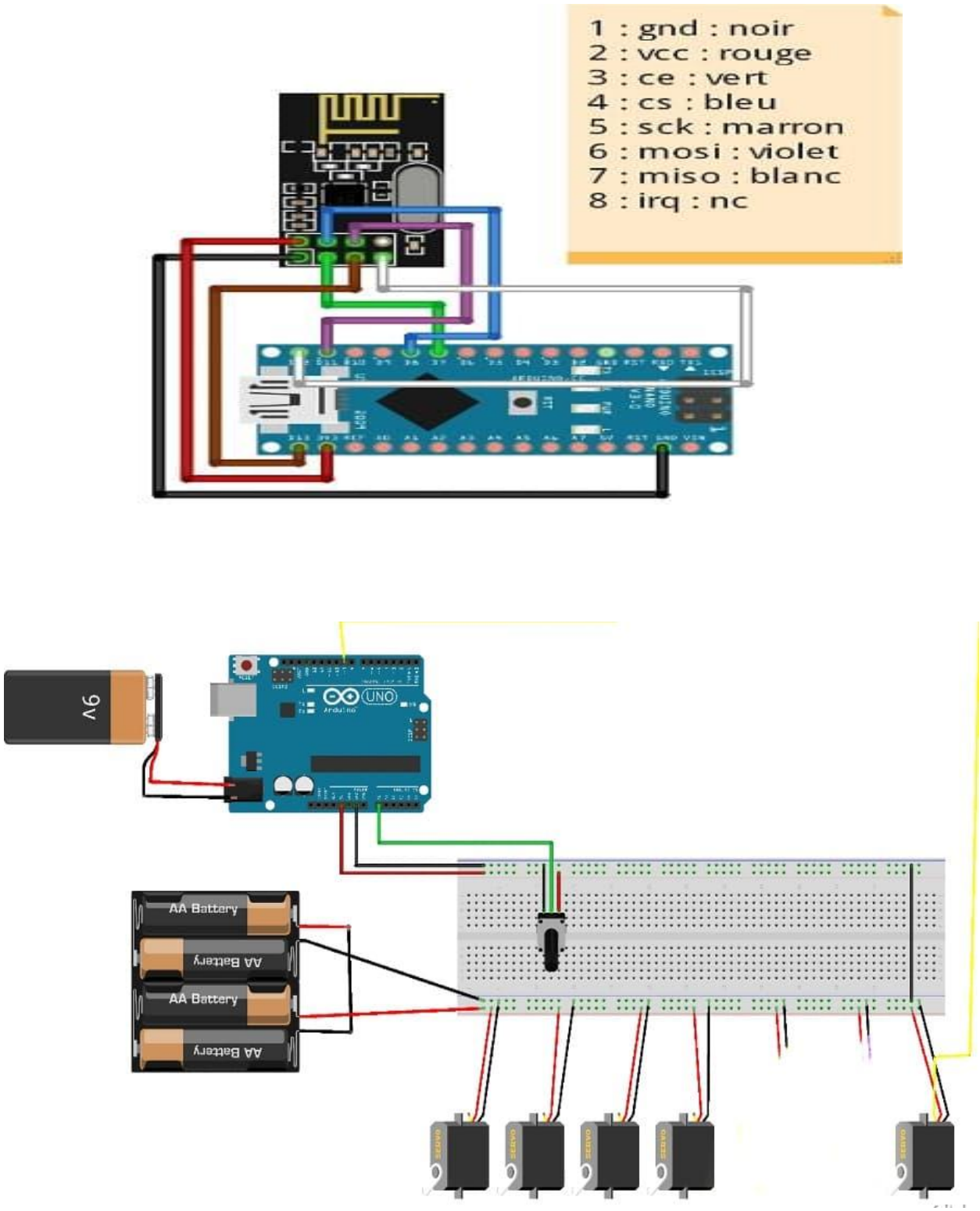


Figure: Flex sensor , Servo motor & radio link circuit

Programming and algorithm development:

Code of transmit

```
#include <SPI.h>
#include "RF24.h"
RF24 radio(9, 10); // Establish a new RF24 object
const uint64_t send_pipe=0xB01DFACECEL;//This will be this device
const uint64_t recv_pipe=0xDEADBEEFF1L;//This will be the other device

//flex sensor
const int FLEX_PIN[5] = {A0,A3,A4,A5,A6};
const float VCC[5] = {4.74,3.38,4.76,4.77,4.73};
const float R_DIV[5] = {9550.0,9550.0,9550.0,9500.0,9500.0};
const float STRAIGHT_RESISTANCE[5] =  {4888.0,23600.0,2400.0,24000.0,1141.0};
const float BEND_RESISTANCE[5] = {1700.0,0.0,4000.0,4000,5000.0};
float sensor_data_angle[5];
int flexADC[5];
float flexR[5];
float flexV[5];

void setup()
{
  Serial.begin(9600); // Set up communcations with the serial monitor in the arduino IDE

  Serial.println("Start");
  radio.begin();// Basically turn on communications with the device
  radio.setPALevel(RF24_PA_LOW);//RF24_PA_MAX is max power
  radio.setRetries(15,15);//This will improve reliability
  radio.openWritingPipe(recv_pipe);//Set up the two way communications with the named device
```

```
  radio.openReadingPipe(1,send_pipe);
  radio.stopListening();
}
void loop()
{
   for (int i=0;i<5;i++)
   {
     flexADC[i] = analogRead(FLEX_PIN[i]);
     Serial.println(flexADC[i]);
     flexV[i] = flexADC[i] * VCC[i] / 1023.0;
     flexR[i] = R_DIV[i] * (VCC[i] / flexV[i] - 1.0);
     Serial.println(String(i) + "------Resistance: " + String(flexR[i]) + " ohms\n");
     sensor_data_angle[i] = map(flexR[i], STRAIGHT_RESISTANCE[i], BEND_RESISTANCE[i],0, 90.0);
     Serial.println(String(i)+ "-----Bend: " + String(sensor_data_angle[i]) + " degrees");
     Serial.println();
   }

    if(!radio.write(sensor_data_angle, sizeof(5*sensor_data_angle[0])))// Send the message_code and
check to see if it comes back false
    {
      Serial.println("Failed");
    }
    else
    {
      Serial.println("data sent by radiolink\n");
    }

    delay(2000);
}
```

Code of Receiver

```
#include <SPI.h>

#include "RF24.h" // This is the RF24 library that may need to be installed through the Manage Libraries feature in the IDE.

#include <Servo.h>


//initialize a servo object for the connected servo

Servo servo_test0;

Servo servo_test1;

Servo servo_test2;

Servo servo_test3;

Servo servo_test4;

Servo servo_test5;

float angle = 0;

const int pin[6]={2,3,4,5,6,7};


RF24 radio(9, 10);//Create a commuications object for talking to the NRF24L01

const uint64_t send_pipe=0xB01DFACECEL;//These are just arbitrary 64bit numbers to use as pipe identifiers

const uint64_t recv_pipe=0xDEADBEEFF1L;//They must be the same on both ends of the communciations


void setup()

 {

  Serial.begin(9600);//Set up comm with the IDE serial monitor


    servo_test0.attach(pin[0]);

  servo_test1.attach(pin[1]);
```

```
 servo_test2.attach(pin[2]);
 servo_test3.attach(pin[3]);
 servo_test4.attach(pin[4]);
 Serial.println("pins are connected");
Serial.println("listening");
 radio.begin();//Start up the radio object
 radio.setRetries(15,15);//This will improve reliability of the module if it encounters interference
 radio.setPALevel(RF24_PA_LOW);//This sets the power low. This will reduce the range. RF24_PA_MAX would increase the range
 radio.openWritingPipe(send_pipe);//Thses are the reverse of the transmit code.
 radio.openReadingPipe(1,recv_pipe);
 radio.startListening();//Give the module a kick

}
float sensor_data_angle[5];

void loop()
{
  if( radio.available())//Keep checking on each loop to see if any data has come in
 {
   while(radio.available())//Loop while there is incoming data. The packets are one unsigned long in total so it shoudl only loop once
   {
    radio.read(sensor_data_angle, sizeof(5*sensor_data_angle[0]));//Stuff the incoming packet into the motor_code variable

      servo_test0.write(2*int(sensor_data_angle[0]));
      servo_test1.write(2*int(sensor_data_angle[0]));
      servo_test2.write(2*int(sensor_data_angle[0]));
      servo_test3.write(2*int(sensor_data_angle[0]));
```

```
      servo_test4.write(2*int(sensor_data_angle[0]));
        }
   delay(10);
 }
}
```

Application

1. Wirelessly controlling any robot in a fully manual way .
2. Helping physically disabled persons.
3. Protecting human body in dangerous jobs where parameters like temperature,radiation is so high .
4. Controlling machines from remote places.
5. Using dangerous weapons from remote places
6. Performing as a medical assistance in surgery cases.
7. Assembling critical machine components more precisely.

Limitations

1. High 3D printing cost.
2. Servo motors that we used are not that much efficient.
3. Flex sensors were short that needed to be a bit longer.
4. Weight of the hand is little heavier .
5. It cannot lift heavier objects as it needs more torque.

Conclusion

We compared the performance of our mechanical hand with flex sensor with how the hand moved when we control it using the gloves . It works almost exactly the same way . our flex sensors are able to attain the appropriate as the controller .

At the beginning it was decided to use plastic to make mechanical hand . we found it as a difficult task for the purpose of designing matter . That's why we use stailess steel and gave shape it in workshop.Then we made our hand plastic and wrist is made of stainless steel.

Although there are a lot of limitations we can say it was a successful project in such a short time . This project provide us a great opportunity to implement theoretical knowledge obtained specially at ME 361 course . It was a great pleasure to achieve the experience of practical life . The project introduced us with reality & related our theoretical knowledge with practical life . Experiences gathered here is certainly going to be very helpful in days to come .

Recommendations (Further Improvements)

- With better sensing capabilities it can be made to be used in search and retrieve missions for chemical spillage and biological agents.
- With proper insulation it can be used to work in extreme temperature conditions
- With proper modifications and precision it can be used to define bombs in wars
- A wireless camera can be used. This feature will enable to maneuver the carrier without actually being physically able to see it. This provides great strategic advantage for the controller
- A thermal sensor could also be attached. This would add extra sensing capability which would help us to determine the temperature of the working environment
- Attaching a pressure sensor with mechanical hand so that it can pick up fragile and sensitive materials

- The driving wheel system could be replaced by caterpillar tracks if possible. It would ensure power transmission all the wheels more efficiently and also make it more robust and powerful

References

1.Shamsil,A.,(2010).Motion Tracking Glove for Human Machine Interaction: Grasp & Release , Department of Electrical & Computer Engineering , McMaster University,Ontario, Canada.

2. Nair ,A., Shete ,D., Somanathan,N., & Sardesai,R., (2010-2011).Motion controlled Robotic Arm Using Flex Sensors, Department of Instrumentation Engineering , Smt.

Indira Gandhi College of Engineering ,Mumbai,India.