ELDERLY CARE

Peripherals Lab Project

Batch R

Abhishek Kumar	150101003
Shashank Garewal	150101067
Shubham Jindal	150101071
Soumik Roy	150101074

Contents:

Click *here* for project Video

1. Introduction

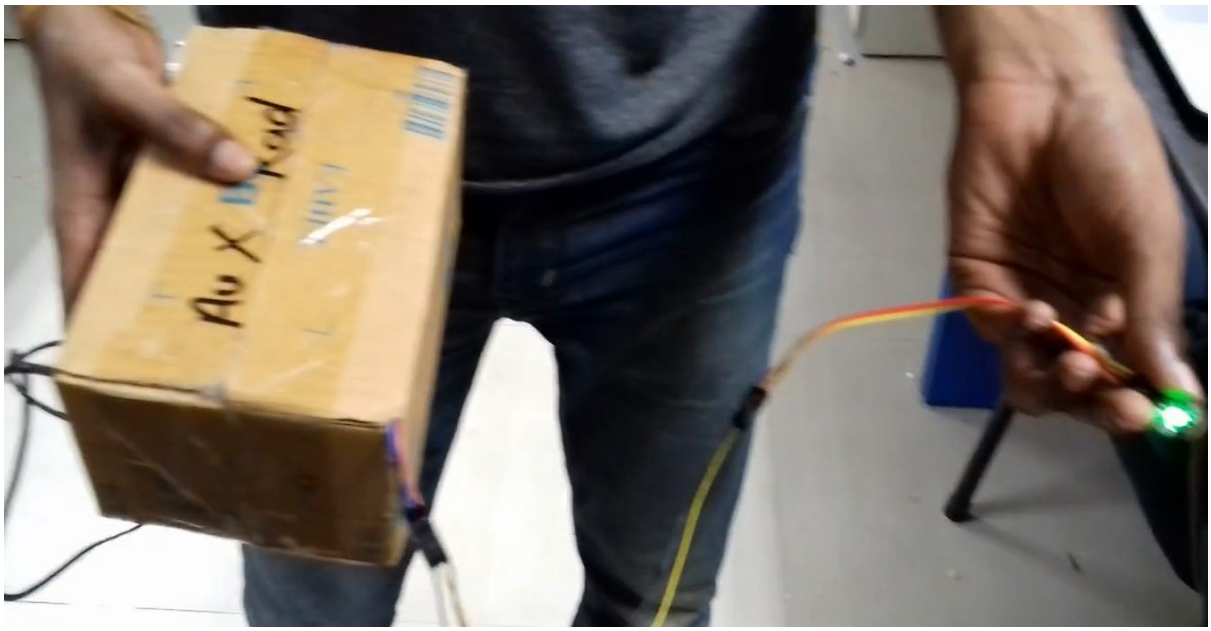
The aim of this project is to provide smart solutions for the daily needs of an elderly person. We have added sensors to monitor health and well being of the elderly person. Also, some smart home solutions are added.

2. Modules

We have divided our project into two modules. Both the module have different functionalities.

The modules are the modules:

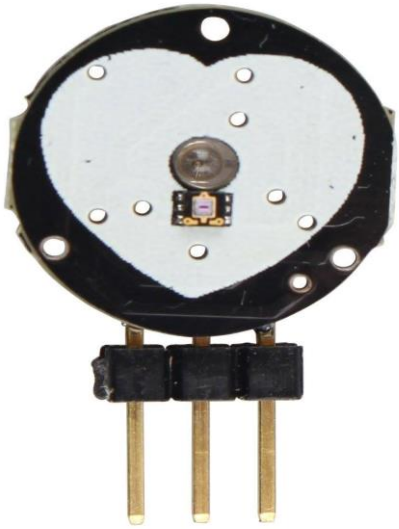
2.1. Wearable



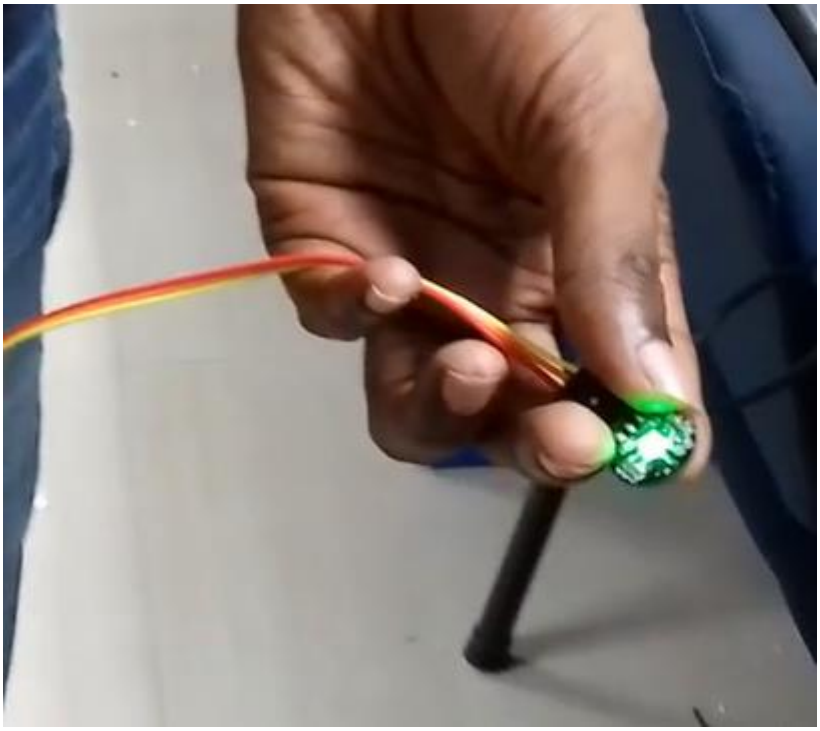
This module is supposed to be worn by the elderly person around his/her waist. This module has an Arduino in it that does all the processing. The wearable has an extension that contains the pulse rate sensor which attaches to the fingertip of the person. For mobility this sensor is powered by a 9V battery or power bank. All the data from this module is wirelessly sent to the other module to be forwarded to the server.

This module consists of the following sensors:

2.1.1. Pulse Rate Sensor



This sensor is attached to the finger tip of the person. It works on the principle of different optical properties of the oxygenated and deoxygenated blood. A strong light source is in the sensor. With every heartbeat there is a spike in arterial (oxygenated) blood which is depicted in change in absorbance or reflectance of the light.



This sensor gives analog data. In our code we did sampling of the sensor data for every 25 milli seconds. Whenever the value crosses a threshold value for oxygenated blood and comes back below we count one heartbeat. This process we continue for 60 seconds and then send the heartbeat count to the server.

Code:

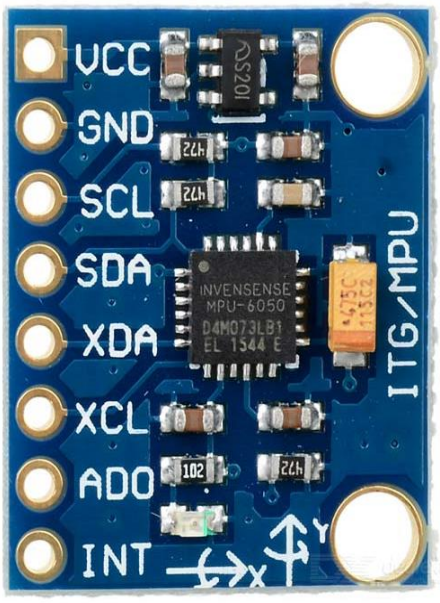
a. For counting

```
void Cal_heartbeat()
{
  int val = analogRead(Pulsepin);
//  Serial.println(val);
  pulse = val;
  if(oldpulse < Threshold && pulse >= Threshold)
    goingup = 1;
  if(oldpulse >= Threshold && pulse < Threshold)
    goingdown = 1;
  if(goingup == 1 && goingdown == 1)
  {
    heartbeat++;
    goingup = 0;
    goingdown = 0;
  }
  oldpulse = val;
}
```

b. For sending count on xbee

```
void printheartbeat()
{
  Serial.print("No. of Heart Beats are: ");
  Serial.println(heartbeat);
  xbee.write("No. of Heart Beats are: ");
  //xbee.write(heartbeat);
  int hb=heartbeat;
  int i=3;
  char c[3];
  for(i=3; i>0; i--){
    int a  = hb%10;
    hb = hb/10;
    c[i-1] = a+'0';

  }

  for(i=0; i<3; i++)
    xbee.write(c[i]);
  xbee.write("\n");
  if(heartbeat < 20)
  {
    Serial.println("Emergency!! Possible Heart Failure");
    xbee.write("Emergency!! Possible Heart Failure\n");
  }
  if(heartbeat > 100)
  {
    Serial.println("Emergency!! Possible Heart Attack");
    xbee.write("Emergency!! Possible Heart Attack\n");
```

2.1.2. Accelerometer



We are using MPU 6050 as our accelerometer. This sensor is placed inside the wearable box. The sensor is placed with is z axis facing upwards and x axis facing towards the front.

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.

We have used accelerometer for fall detection. As falling can be a very dangerous situation for any elderly person. A sudden fall in the direction sends notifies the server of this emergency.

Code:

```
void falldetect()
{
  Vector rawAccel = mpu.readRawAccel();
  Activites act = mpu.readActivites();
  if(act.isFreeFall)
  {
    Serial.println("Emergency!! your grandparent fell");
    xbee.write("Emergency!! your grandparent fell\n");
  }
  else
  {
    //Serial.println("Emergency!! your grandparent is enjoying life");
//    xbee.write("Emergency!! your grandparent is enjoying life\n");
  }
  if (freefallDetected)
  {
    ledState = !ledState;

    digitalWrite(4, ledState);

    freefallBlinkCount++;

    if (freefallBlinkCount == 20)
    {
      freefallDetected = false;
      ledState = false;
      digitalWrite(4, ledState);
    }
```

2.2. Home

This is the stationary module of our system. This module is directly connected to the server. This module provides a little bit of home automation for assistance to the elderly person. There is an Arduino in this module for control operations.

This module has the following sensors:

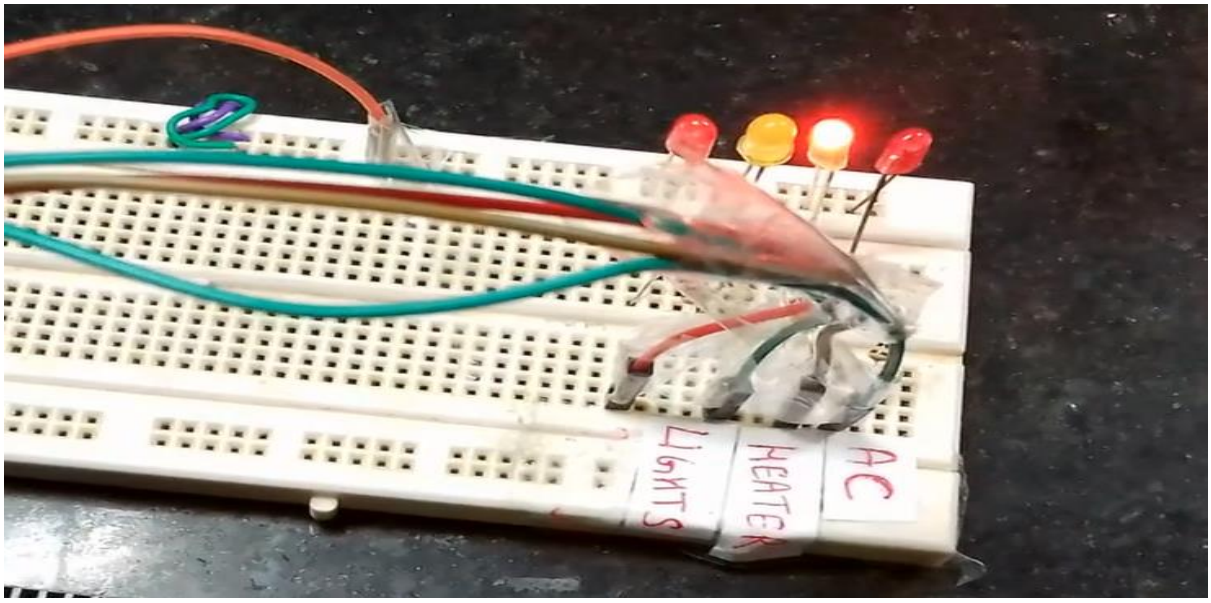
2.2.1. Temperature and Humidity Sensor

This sensor is placed inside the room of the elderly person. This sensor monitors room temperature and humidity.

Two thermal sensors conduct electricity based upon the humidity of the surrounding air. One sensor is encased in dry nitrogen while the other measures ambient air. The difference between the two measures the humidity and the air temperature.

This sensor also give analog data which we are sampling per 100 millisecond. This data is used to regulate the air conditioning of the room. If the temperature of the room falls below a certain level (in our case 20 degree) the

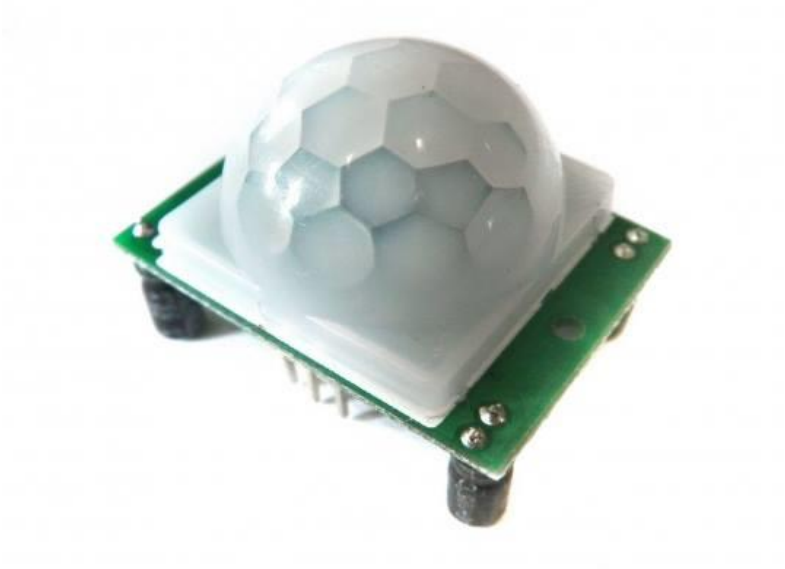
heating is turned on and AC is turned on if the temperature goes above 30 degrees. We are showing this using LEDs.



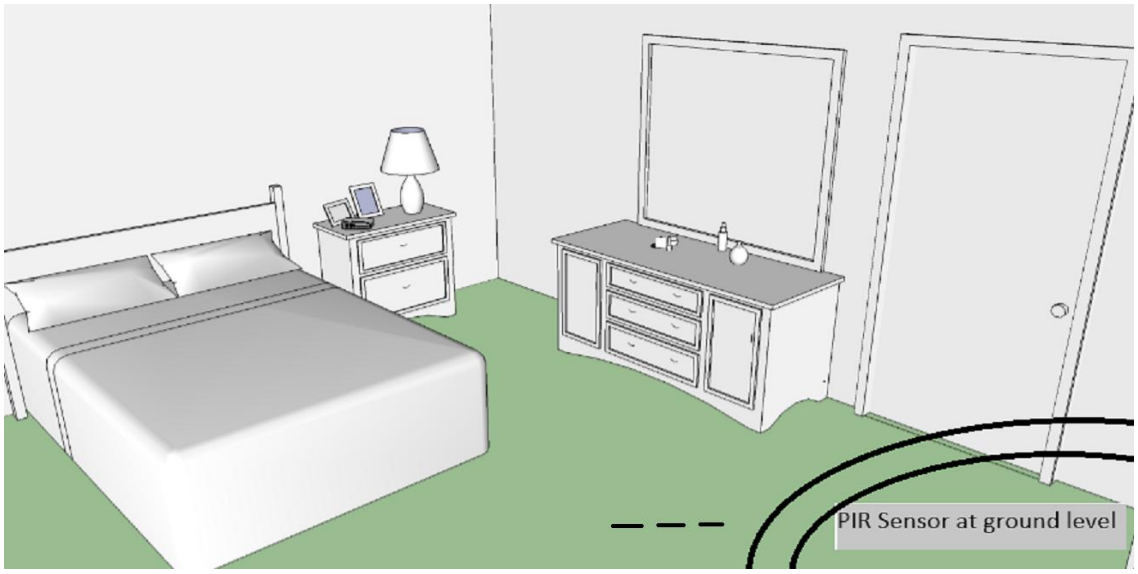
Code:

```
void printTempHumi()
{
  if(t > 30.0)
  {
    Serial.println("Turning On AC");
    stateLED1 = HIGH ;
    if(stateLED2 == HIGH)
    {
      stateLED2 = LOW;
    }
    digitalWrite(LED2, stateLED2);
    digitalWrite(LED1, stateLED1);
    Serial.println("AC Turned On");
  }
  if(t < 29.0)
  {
    Serial.println("Turning On Heater");
    xbee.write("Turning On Heater");
    stateLED2 = HIGH ;
    if(stateLED1 == HIGH)
    {
      stateLED1 = LOW;
    }
    digitalWrite(LED1, stateLED1);
    digitalWrite(LED2, stateLED2);
    Serial.println("Heater Turned On");
    xbee.write("Heater Turned On");
  }
```

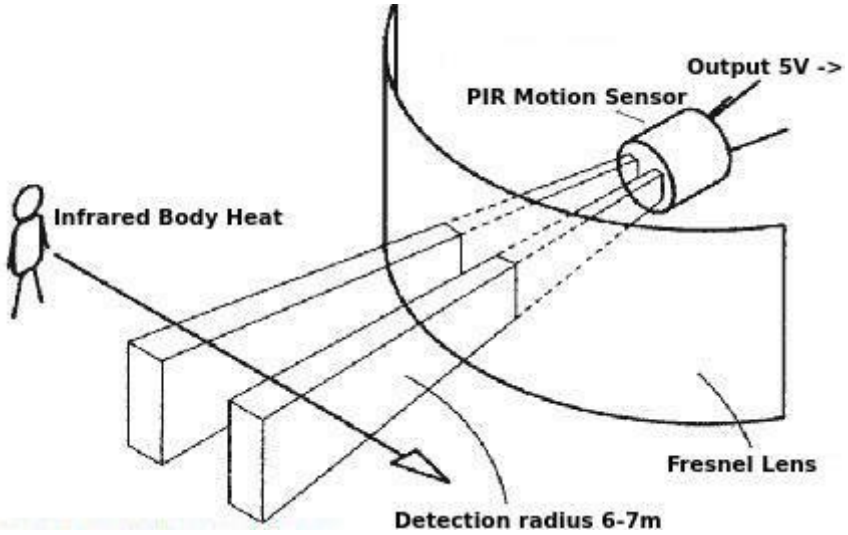
2.2.2. PIR Motion Sensor



This sensor is to be placed on the ground level of the room as shown in the following diagram.



An individual PIR sensor detects changes in the amount of infrared radiation impinging upon it, which varies depending on the temperature and surface characteristics of the objects in front of the sensor. The sensor detects motion only in the field shown in the following image.



This sensor gives digital data. 0 in case of no motion and 1 if motion is detected. We are sampling the data from this sensor for every 20 milliseconds. If we receive 0s continuously for a fixed period of time the lights are turned off in the room. Placing the sensor gives us an additional benefit of not detecting the motion on the bed and not switching lights on during the sleep.

Code:

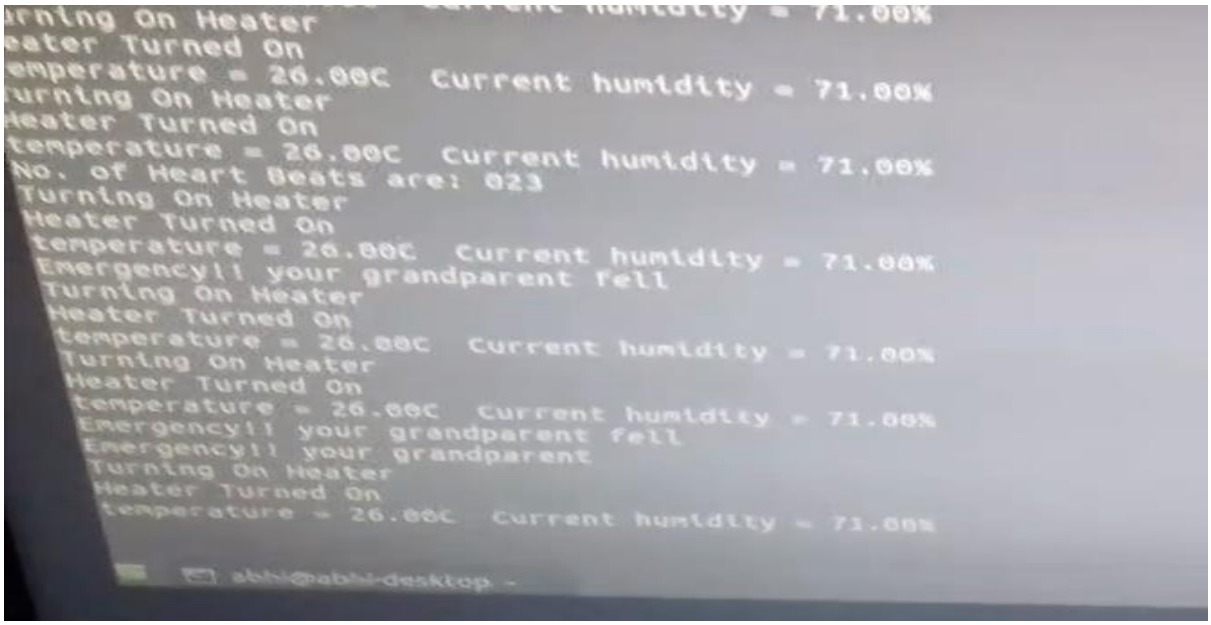
```
int motiontimer = 0;
int pir = 13;
int flag=0;
void lightsoff()
{
  int val = digitalRead(pir);
//  Serial.print(val);
  if(val == 0)
    motiontimer += 25;
  else
    motiontimer = 0;

  if(motiontimer >= 10000)
  {
    digitalWrite(light, 0);//setting lights off
    lightvalue = 0;
  }
  if(lightvalue== 0 && val == 1){
    digitalWrite(light,1);
    lightvalue = 1;
  }
}
```

2.2.3. Alarm

We have made an alarm system in the house module for reminding the elderly person for taking medicine. We have implemented this as LED blinking and buzzer alarm system. After every fixed interval (used 10 seconds for the demo) the LED will blink and buzzer will sound.

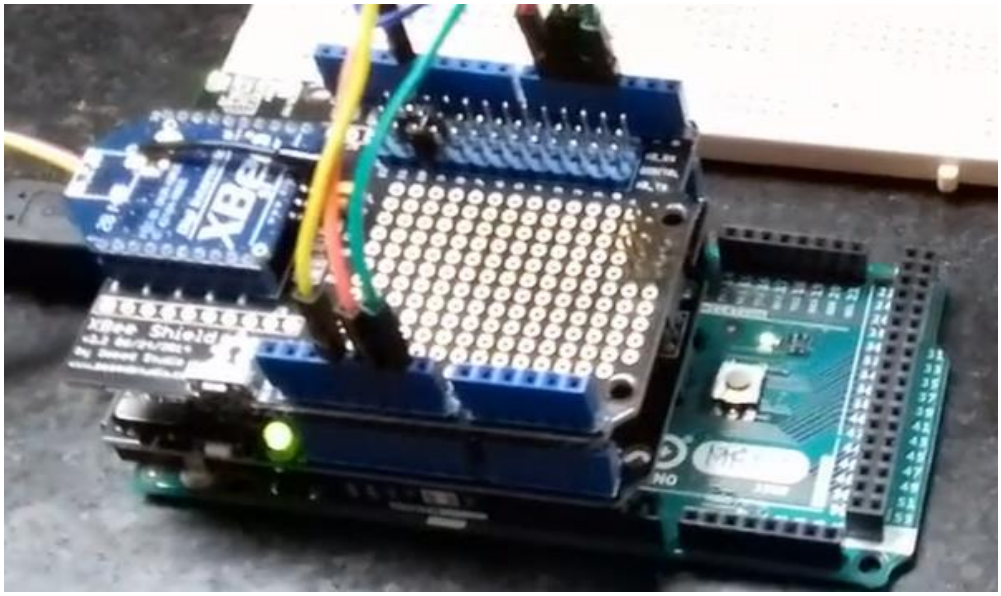
2.3. Server

We have used a Raspberry Pi as our server. The server is connected by a usb cable to the home module of the Arduino. It collects data from all the modules and displays it. From the server we have the capability to send data to other devices.



3. Communication

We are using xbee devices for communication between the modules. One of the xbee devices is connect to the wearable module and the other one is connected to the home module using xbee shields. The xbee provides wireless communication between the modules. We have calibrated the xbees such that the wearable module is sender and the home module is the receiver. Using the xbee device provides mobility to the wearable module. Now the wearable module no longer needs to be connected to the server to send data.



The home module Arduino sends data to the server via USB cable. We are running a python program on the Raspberry Pi to collect the data coming from the USB port. The data is then displayed on the monitor connected to the Pi.