CS321 PROJECT REPORT

Group 19 MEMBERS:

1. Hemanth Gandham – 140101027

2. Mohd Imran – 140101041

3. Arun Chandh - 140101075

SMART HEALTH PROJECT FOR (ELDER PEOPLE/PHYSICALLY CHALLENGED/PATIENTS).

OBJECTIVE

The project is aimed towards patients/elder people who stay in home either after post treatment or those people who require detailed medical attention. And our idea is to continuously monitor their health statistics, vitals, etc. through the internet and the storage of the data which would serve for self-assessment.

FEATURES:

- Parameters like temperature, heart beat are checked continuously and are sent to the caretaker and the doctor who see these stats once they come online.
- Warning for abnormal variations will be sent to them.
- In cases of emergency such as heart attack or the body temperature gets too cold we send an emergency alert to the care taker, doctor and few related people who can reach him as fast as possible. (BY ONLINE AND BY CALL and MESSAGE means).

- This prototype can be extended and used as a dashboard in hospitals to monitor the health of the patients by adding additional health parameters.
- This device uses IoT since pi is connected to Internet (wirelessly) and can be extended to use MQTT protocol.

Equipment Used:

1. Arduino x 2: For receiving data from the sensors used and make the output available for reading.
2. Raspberry Pi 2 : For reading the arduino data and accelerometer's readings.
3. Pulse rate Sensor : Reading the pulse rate.
4. GSA modem : For calling(missed call)/messaging with the message during cases of emergency /high caution.
5. Temperature and Humidity Sensor: Read body temperature
6. Accelerometer (or) PIR motion Sensor : Detecting movement
7. Buzzer : Initially used as a prototyoe for the GSM modem
8. Arduino Base Shield: To connect multiple sensor to arduino with ease making the circuit simple and compact.

BRIEF DESCRIPTION OF IMPLEMENTATION

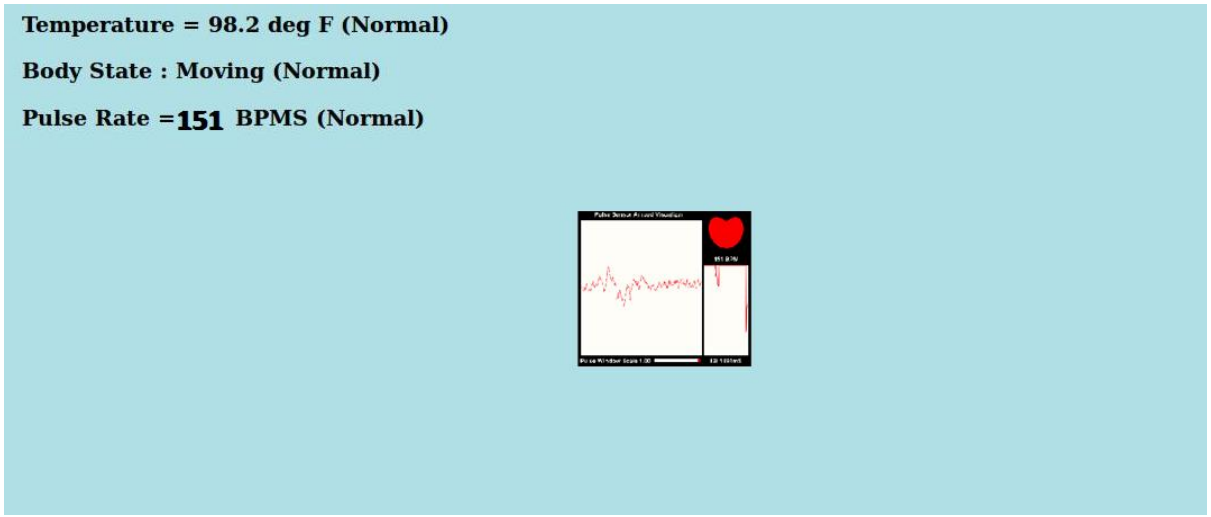
The product is a wearable device consisting of all the above components.

Inside the wearable device are the two arduinos connected to the pi and the accelerometer connected with the pi.Also present is the GSM module.

We place the DHT11 temperature sensor and pulse sensor at their respective appropriate locations on the patient's body.(Pulse senor tied to the fingertip and temperature sensor near the armpit region). Of the 2 arduinos one is solely dedicated for the use of pulse sensor while the other takes care of the remaining sensors except the accelerometer (connected to Pi).

These values are sent to Pi from Arduino which sends it to the server we host from which the doctor/caretaker can see the data of our vitals.

A graph of the heart rate is plotted using the values obtained from the pulse sensor and on open software Processing 3 of which screenshots will be sent to the page for every 1 minute.



simple webserver displaying data

During emergency situations like:

a) Pulse rate falls below certain rate or exceeds a certain rate .

b) Temperature rate exceeds certain value or is below a certain value.

c) Accelerometer detects sudden decrease or a sharp spike and becomes constant later.

A message is sent to the doctor and people concerned with the patient. A missed call is also given to them.

Setting up the Hardware:

1. Raspberry Pi set up:

(a) Boot the Raspbian Operating System on to the memory card from a PC.

(b) Set a static IP on to Raspberry pi Add this code in the end of /etc/dhcpcd.conf file

```
interface eth0
static ip_address=192.168.0.10/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1
```

interface = This defines which network interface you are setting the configuration for.

static ip_address = This is the IP address that you want to set your device to. (Make sure you leave the /24 at the end)

static routers = This is the IP address of your gateway (probably the IP address or your router) static domain_name_servers = This is the IP address of your DNS (probably the IP address of your router).

(c) Accessing Pi Remotely:

Connect Raspberry pi and your PC with a Lan Cable and directly ssh it with the static IP SSH COMMAND :

ssh -X pi@static_Ip

Software Assembly:

1. Setting up the internet in pi:

Add this lines in the /etc/apt/apt.conf file

```
Acquire::http::proxy "http://username:password@proxy:port/";
Acquire::https::proxy "https://username:password@proxy:port/";
```

2.Installing LAMP and Arduino software in pi:

It is same as installing LAMP in Ubuntu .Follow the instructions from this website. Similarly, install the Arduino software (Linux version)

https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu

3. Reading the data of Arduino and sending it to a webpage.

Arduino sends the data to serial monitor. The following python script takes this data from the serial monitor and stores it in a file (*data.txt* in our case).

```
import serial

ser = serial.Serial('/dev/ttyUSB0', 9600)

# Send data (a string)

sending_string = str.encode("what")

ser.write(sending_string)


# Read data

while True:

   data = bytearray()

   data += ser.read()

   received_str = data.decode()

   print(received_str)

   file = open("data.txt", "a")

   file.write(str(received_str))

   file.close()
```

The following PHP code writes the data from data.txt file to a webpage dynamically (you need not refresh the webpage. Whenever

any data is added or changed, it gets changed automatically on the webpage).

```
<?php
    echo "\n Data : ";
    $fh = fopen('data.txt','r');
    $data = "";
   while ($line = fgets($fh)) {
        echo($line);
        $data += ($line + "<br></br><br></br>"); }
       fclose($fh);
       echo str_replace(PHP_EOL, '<br></br><br></br>', $output);
?>
```

Create a new folder in /var/www/html in pi and add the above two files in it along with the *index.html*

ARDUINO

ARDUINO 1

The DHT11 sensor , PIR sensor and buzzer sensors are connected to it and its output is connected to pi. The Serial rate set in this Arduino is 9600.

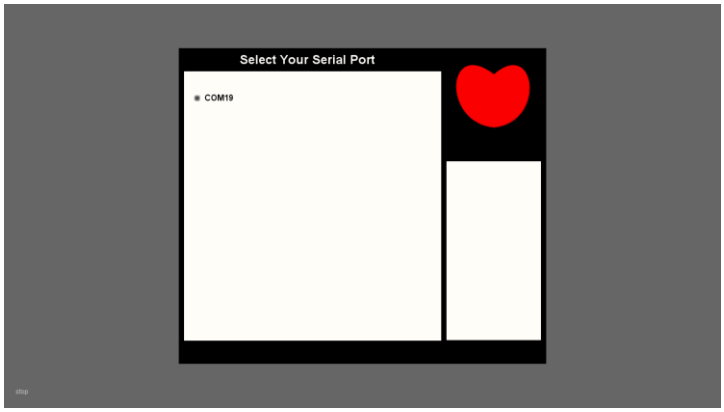
Sensors were connected through the usage of data shield and GSM ‘s (TX and RX pins were connected carefully).

ARDUINO 2

Pulse sensor connected to it.

Works at Serial Rate of 115200.

Its serial port is listened to by the Processing software.

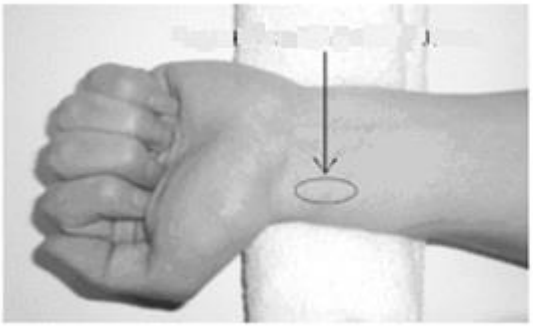


Outputs the BPM (beats per minute) to the Pi which it obtains from Processing.

PULSE SENSOR

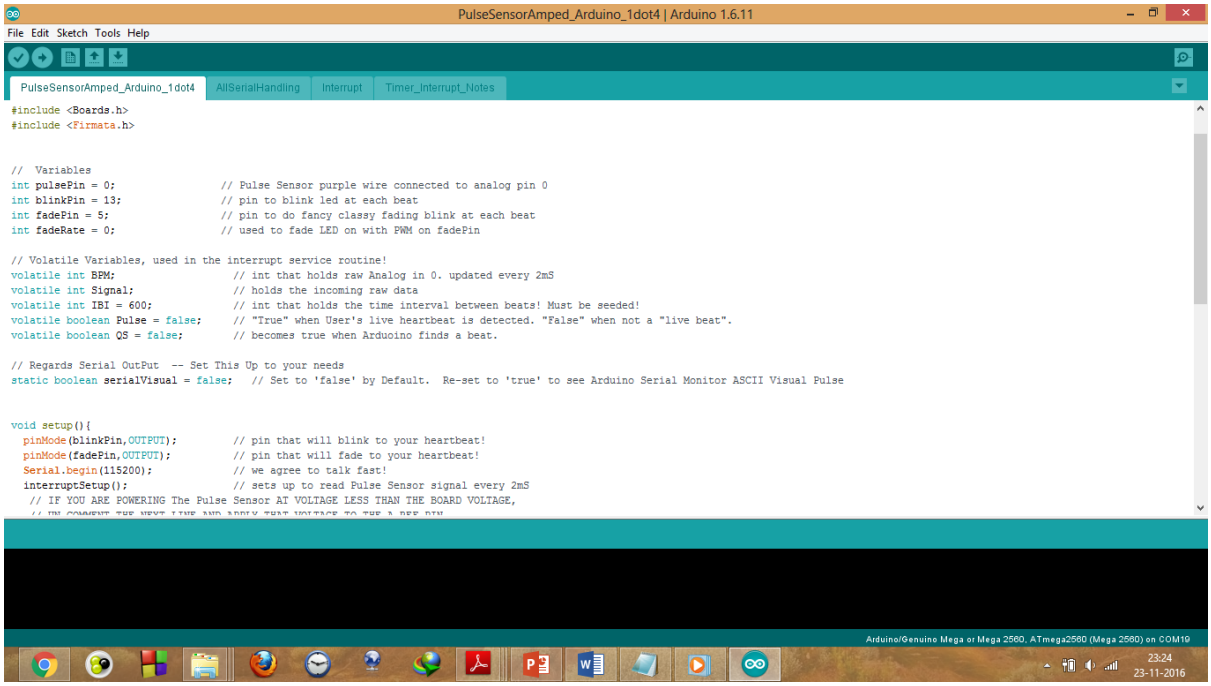
Contains 3 pins of which one is connected to the analog pin of the Arduino board one to the ground.

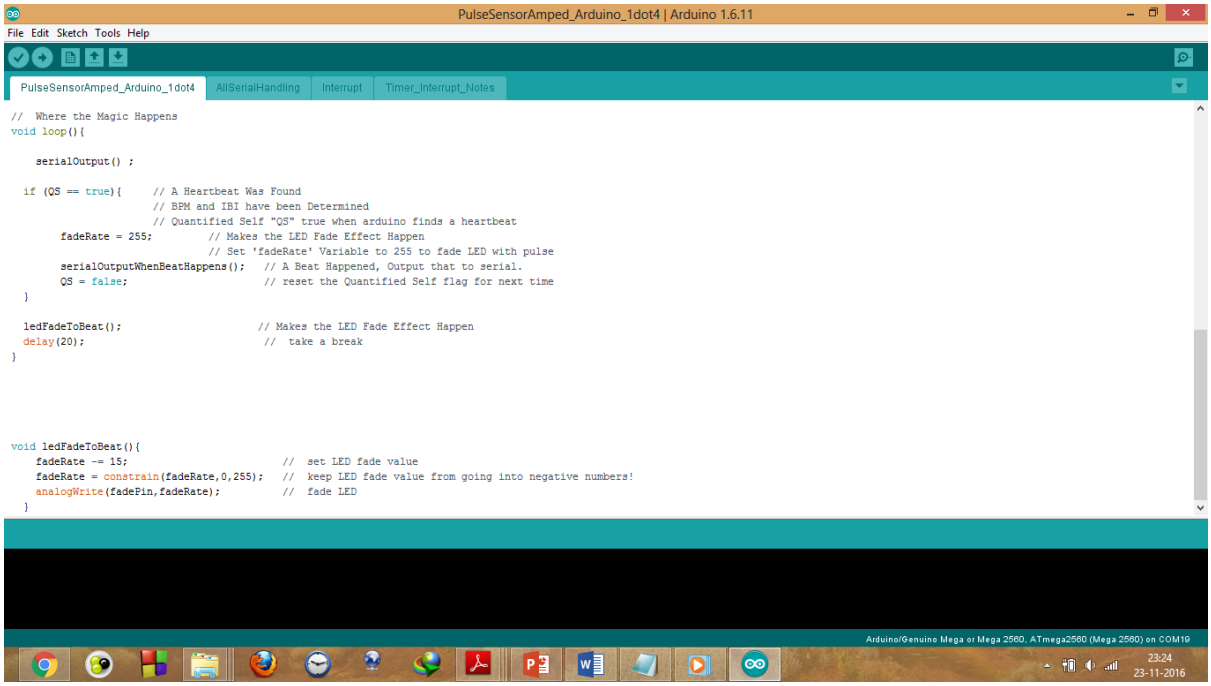
It is connected to the patient in the following way.



Emergency measeures are taken when the pulse is out of the given range(120 – 180) bpm.

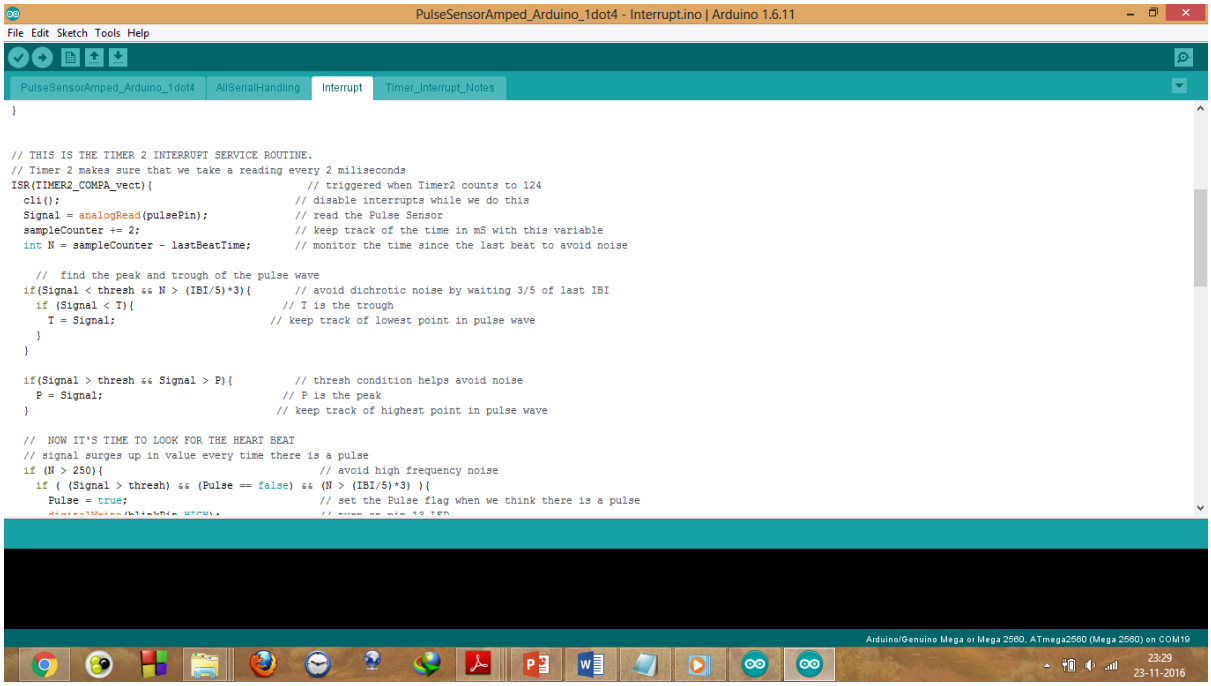
Code (for reading pulse sensor ‘s data)





We use Interrupts while using this sensor so we cant use functions like tone other time related functions,etc.

Usage of Timer interrupt service:



Precautions:

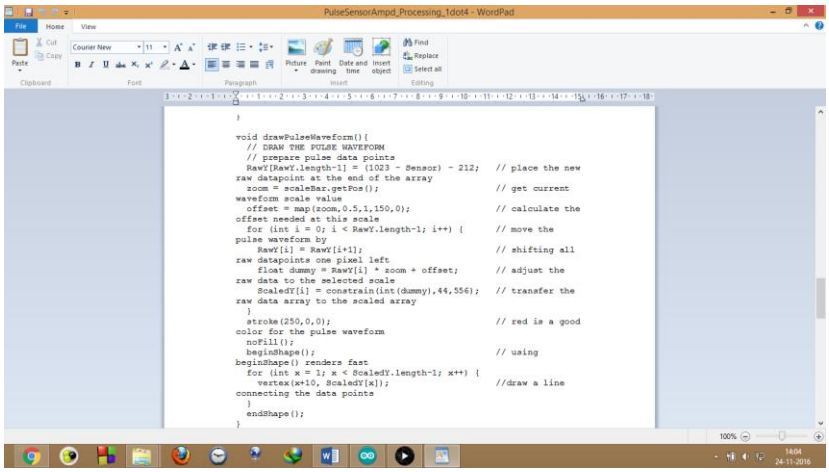
1) The back portion of the pulse sensor should be insulated as the moisture affects its values and also it would remain longer for usage.

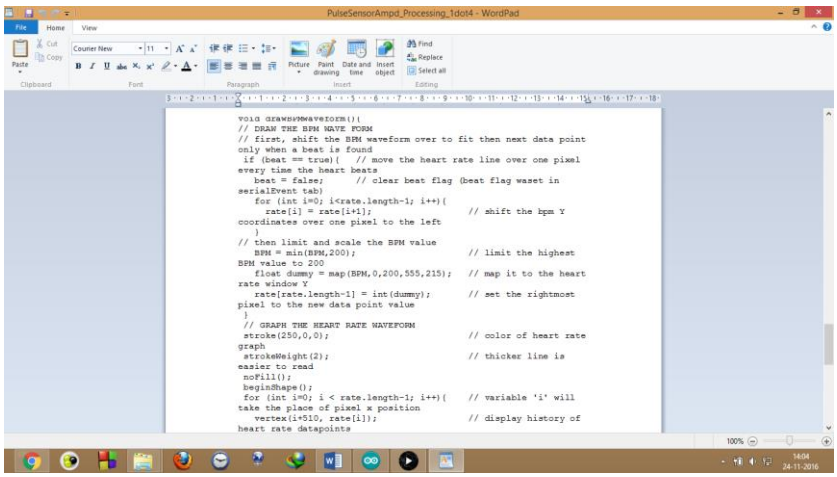
2) Connection should be done in the proper fashion. For this sake a LED has been provided in the sensor to indicate the correct connection/arrangement.

PROCESSING :

Using the Pulse sensor we receive data that is rather complicated but is useful in plotting graphs by using open software like this one.

Software listens to the serial data from the Arduino and plots graphs based on the algorithm it uses.





Screenshots from the Graph Plortting Alogorithm

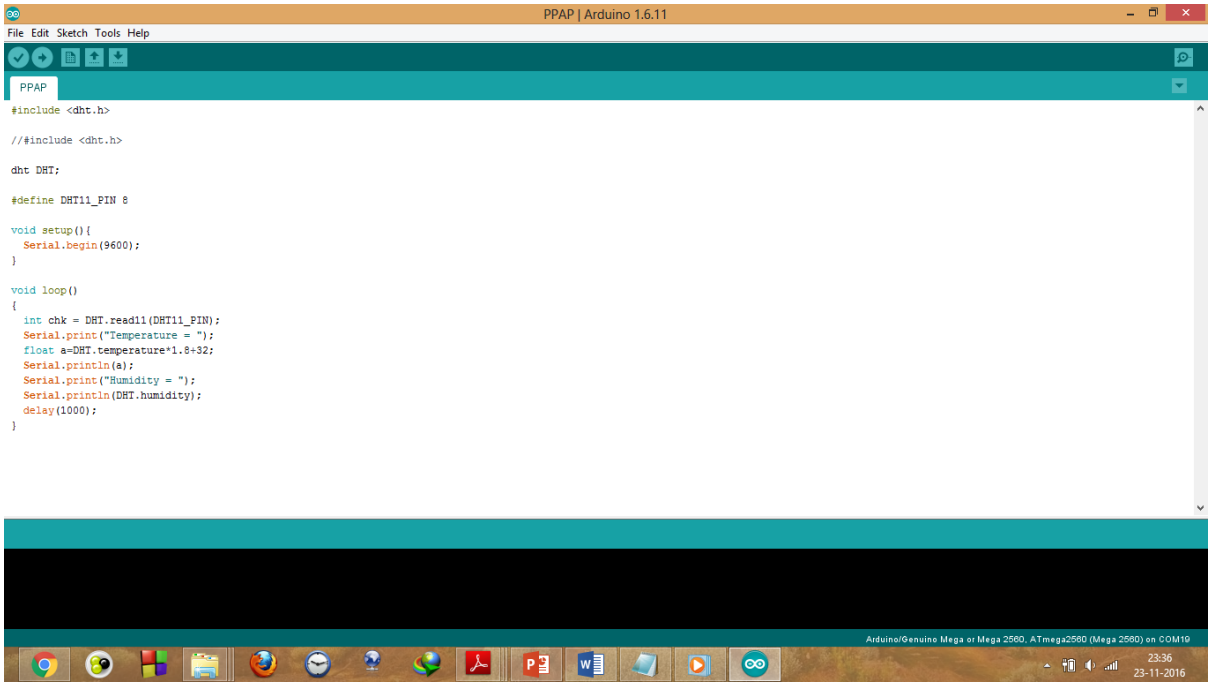
DHT11- Humidity and Temperature Sensor

This DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. It is highly reliable and has excellent long-term stability. Connected to the digital pin 8.

The DHT11 temperature and humidity sensor has the measurement range of 20-90%RH and 0-50 degree Celsius. And in terms of accuracy it gives ±5%RH and ±2 degree Celsius.

DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status.

(Individual) Code which involves Temperature reading part.



Libraries used : DHTLib available in Arduino playground.

Precautions

DHT11 should be mounted at the place as far as possible from parts that may generate heat.

The quality of connection wires will affect the quality and distance of communication and high quality shielding-wire is recommended.

SIM900A GSM Modem

Used for communicating through offline means in cases of emergency.(as mentioned above).The module supports communication in 900MHz band.

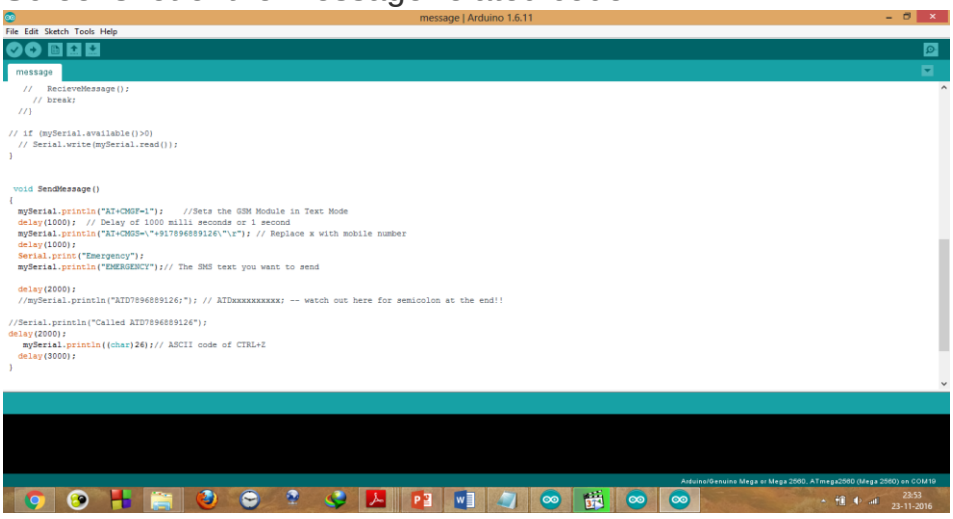
CONNECTING WITH ARDUINO:

Connection with Arduino is to be done with a great deal of care because our GSM module uses a power supply of 2 ampleres and 12 volts and if power taken from Arduino or any incorrect connections will damage the Arduino. An adaptor is used(the one provided with Arduino).

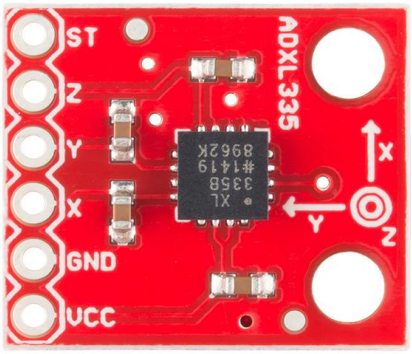
The communication between Arduino and GSM module is serial. So we are supposed to use serial pins of Arduino (Rx and Tx). We may connect the Tx pin of GSM module to Rx pin of Arduino and Rx pin of GSM module to Tx pin of Arduino BUT we faced a

PROBLEM->The problem with this connection is while programming. Arduino uses serial ports to load program from the Arduino IDE. If these pins are used in wiring, the program will not be loaded successfully to Arduino. So you have to disconnect wiring in Rx and Tx each time you burn the program. Once the program is loaded successfully, you can reconnect these pins and have the system working! To avoid this difficulty, we use an alternate method in which two digital pins of arduino are used for serial communication. We need to select two PWM enabled pins of arduino for this method. So I choose pins 9 and 10 (which are PWM enabled pins). This method is made possible with the SoftwareSerial Library of Ardunio. SoftwareSerial is a library of Arduino which enables serial data communication through other digital pins of Arduino. The library replicates hardware functions and handles the task of serial communication.

Screenshot of the message related code:



ACCELEROMETER :



In our project we use this to detect motion, that means whenever there is a sudden motion something like falling down out of balance control or some fainting, we get the reading for the same.

RANGE :

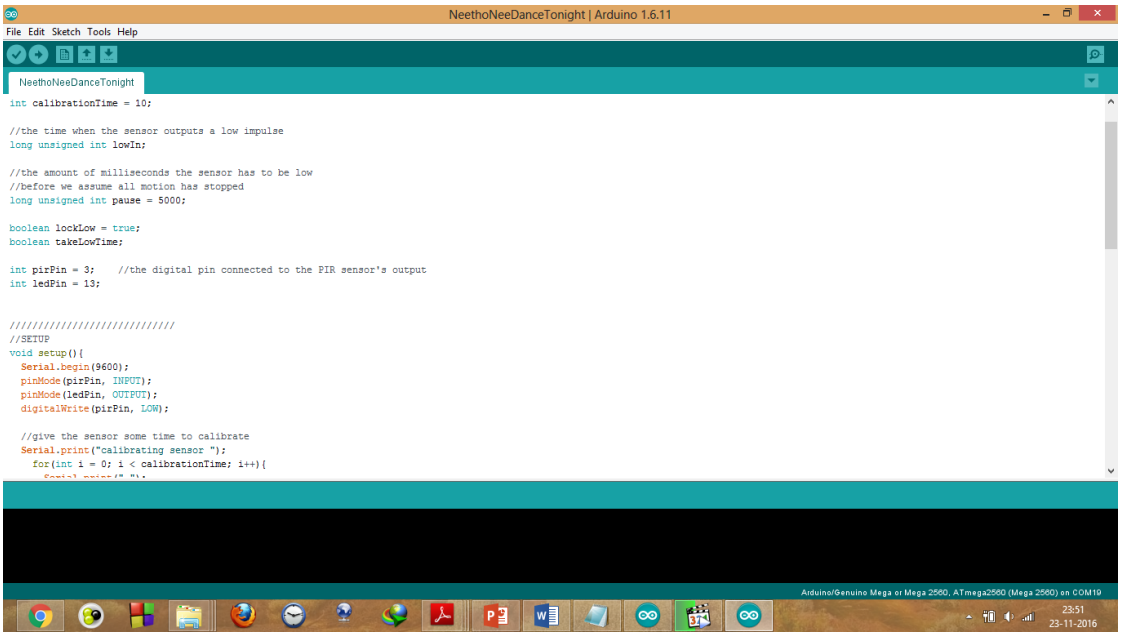
Most accelerometers will have a selectable range of forces they can measure. These ranges can vary from ±1g up to ±250g. Typically, the smaller the range, the more sensitive the readings will be from the accelerometer. Precautions:

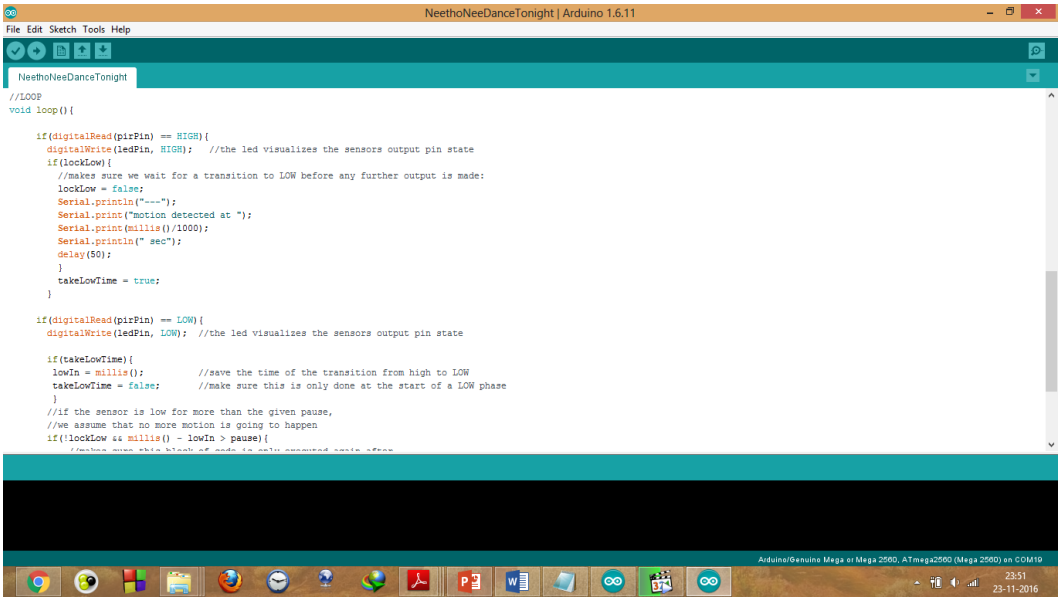
1.Look out for the sharp spikes.

2. care is to be taken while setting the default parameters of the three parameters.

PIR Motion SENSOR:

In our project initially we used this sensor for our need of motion detection, but later shifted to sparkfun accelerometer as this sensor is too sensitive and detects every motion that happens In its range specified. The sensor itself is housed in a hermetically sealed metal can to improve noise/temperature/humidity immunity.(Fresnel lens) Behind the window are the two balanced sensors.





Also used was the buzzer which served as an prototype for the GSM modem implementation. Beep (with the tone function).

SCOPE FOR FUTURE IMPLEMENTATION:

1. Include more sensors like Blood Pressure sensor and Breath sensor, muscle sensor etc. which couldn't be included due to their high cost and complexity.
2. Increase the scope of the webpage and create a connected system between the doctors and patients.
3. Automated assessment and analysis of data.