Contents

CHAPTER-1 INTRODUCTION

1.1. INTRODUCTION:

Health is one of the global challenges for humanity. In the last decade the healthcare has drawn considerable amount of attention. The prime goal was to develop a reliable patient monitoring system so that the healthcare professionals can monitor the patients, who are either hospitalized or executing their normal daily life activities. Recently, the patient monitoring systems is one of the major advancements because of its improved technology. Currently, there is need for a modernized approach. In the traditional approach the healthcare professionals play the major role. They need to visit the patient's ward for necessary diagnosis and advising.

There are two basic problems associated with this approach. Firstly, the healthcare professionals must be present on site of the patient all the time and secondly, the patient remains admitted in a hospital, bedside biomedical instruments, for a period of time. In order to solve these two problems, the patients are given knowledge and information about disease diagnosis and prevention. Secondly, a reliable and readily available patient monitoring system (PMS) is required. In order to improve the above condition, we can make use of technology in a smarter way. In recent years, health care sensors along with Arduino play a vital role.

Wearable sensors are in contact with the human body and monitor his or her physiological parameters. We can buy variety of sensors in the market today such as ECG sensors, temperature sensors, pulse monitors etc. The cost of the sensors varies according to their size, flexibility and accuracy. The Arduino which is a cheap, flexible, fully customizable and programmable small computer board brings the advantages of a PC to the domain of sensor network. In our system we are measuring patient's parameters (ECG, temperature, heart rate, pulse, etc) with different available sensors

the device themselves and have it be cheap enough for them to afford without much debt.

OBJECTIVES:

- Internet of Things (IoT) is the emerging technology, which contains huge amount of smart object and smart devices connected to the internet for communicating with each other.
- In this project to analyze and compute the patient health we are using Arduino Uno, which is the heart of this project.
- The final results are displayed on the android device, on web server and also the results are sent to the user through SMS.
- These data results can be stored in data base centre which can be invoked from remote location at any time in an emergency case of patient without delaying the time.
- This project may play vital role in saving the patient life at emergency time since "Time is life"

1.2.PROBLEM DESCRIPTION AND FOCUS OF THE PROJECT

Paralysis is the inability to move muscles on your own and with purpose. It can be temporary or permanent. The most common causes are stroke, spinal cord injury, and multiple sclerosis. Paralysis can be a complete loss of movement known as plegia, or a significant weakness called paresis.

Paralysis is most often caused by damage in the nervous system, especially the spinal cord. Other major causes are stroke, trauma with nerve injury, poliomyelitis, cerebral palsy, peripheral neuropathy, Parkinson's disease, ALS, botulism, spina bifida, multiple sclerosis, and Guillain–Barré syndrome.For example, monoplegia/monoparesis is complete loss of movement or weakness of one limb. Hemiplegia/hemiparesis is complete loss of movement or weakness of arm and leg on same side of the body. Paraplegia/paraparesis is complete loss or weakening of both legs. Tetraplegia/tetraparesisor quadraplegia/quadraparesis is complete loss or weakness of both arms and both legs.

Paralysis is caused by injury or disease affecting the central nervous system (brain and spinal cord) which means that the nerve signals sent to the muscles is interrupted. Paralysis can also cause a number of associated secondary conditions, such as urinary incontinence and bowel incontinence.

Even though, there are innovative approaches for curing or treating paralysis patients, but the aim of treatment is to help a person adapt to life with paralysis by making them as independent as possible. Where we see a problem with these types of devices that are being developed is that they are very large and expensive machines.

They seem to be only available in hospitals and not able to be used at the patients home or at their convenience. Our goal is to make a device that will be able to retrain a patient‟s motion but have them be able to use.

1.3.RELATED WORK

Nowadays, telemedicine represents a very important research avenue. Significant effort has been deployed in assisting the patients in their everyday life through telehealth and telemedicine systems. In 1. The author describes a wireless patient monitoring system that could allow patients to be mobile in their environment. The developed system includes a pulse oximeter to measure blood oxygen concentration and the patient's pulse, as well as a temperature sensor to keep track of the patient's temperature.

The proof of concept was successful, and allowed for multiple patients at the same time on the same network the ability to add many more patients.data transmission within the system and network related issues and system integration in existing communication networks in the Internet-of-Things (IoT) pardigm. An early system that uses a cell phone application for remote monitoring that uses a GPS, an accelerometer and a light sensor is described in *2+. The prototype in*3+ allows physicians to remotely asses the patients health in and outdoors.Alongside sensors that acquire the data from the patient, their prototype also includes an alert system for emergency cases.

In the authors propose a monitoring system based on Embedded Network Gateway Servers (ENGS).These devices can serve the acquired data via an embedded web server and have modular design in order to be interfaced to a variety of sensors. The authors of detail the specific challenges posed by the design of wearable patient monitoring devices. They address in detail power consumption, data acquisition and filtering issues.

1.4.SYSTEM DESIGN AND IMPLEMENTATION

To overcome all the above drawbacks and meet the requirements of the system, we propose a system which mainly consists of a transmitter and a receiver section. In the transmitter section (at the patient side), a two axis accelerometer will be placed on the finger of the patient. This accelerometer is capable of measuring the static acceleration due to gravity and thus finding the angle at which the device is tilted with respect to the earth. Whenever patient needs any help he tilts the accelerometer in different directions. This acts as an input to the accelerometer while output of it is in volts that is connected to the controller board which acts as the processing unit. The output of the accelerometer depends on the tilt angles and is read by the controller. The controller maps the input voltages between 0 and 5 volts into integer values between 0 and 1023 as analog data from the range of 0-1023. This range provides a lot of sensitivity and a slight shift can lead to change in value.

To reduce the complexity and provide a simple way for the patients, we reduced its sensitivity by mapping it to 0-5 volts and then provided a range for front, back, forward and backward. These directions can be easily understood and used by any person using his/her thumb or any part of the body capable of moving in these directions. A predefined message catering to the basic needs of the patients and those required for emergency will be stored in the ranges assigned to a particular direction as mentioned above. For example: food/water is the message displayed when the patient moves his finger to the right.

So on tilting the accelerometer to the right, it will send its value to the controller. If this value lies between the range assigned to the right direction the predefined message that is food/water in this case will be sent to the next module that is the RF transmitter module. The RF transmitter becomes active when a message is sent from the controller for transmission. RF transmitter and receiver works on the frequency of 434 MHz. The accelerometer will be connected to each patient and each patient will have a controller board and transmitter for sending his messages.

TRANSMITTER

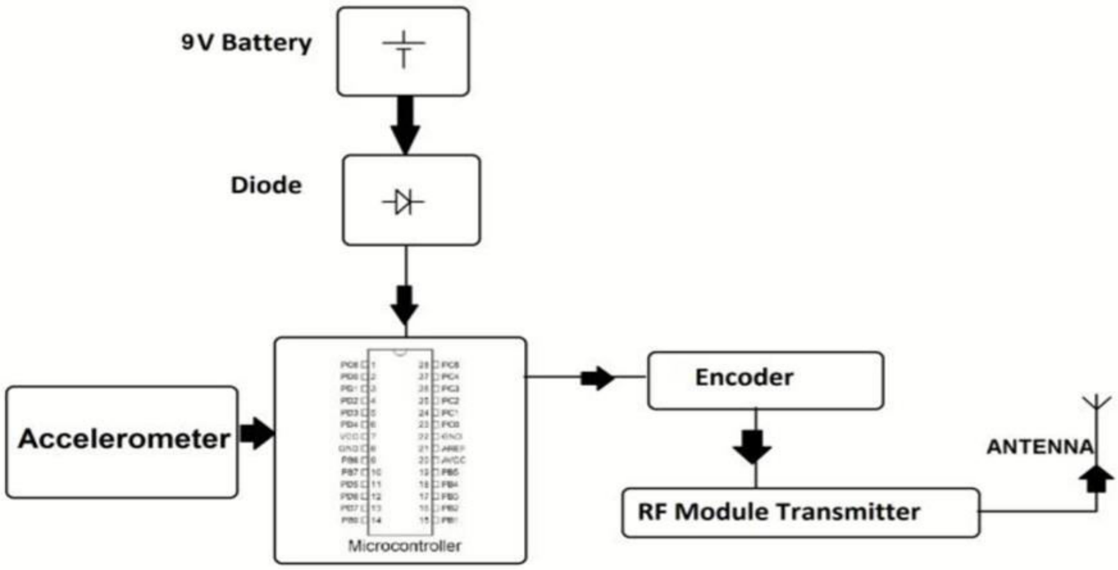

Fig.a. Transmitter block model

For identification of different patients their name or number is sent to the nurse. All these transmitters can be connected centrally to one RF receiver which works on the same frequency as the transmitter. Thus the proposed system will provide a many to one communication. At the receiver side, RF receiver will receive the message and send it to the controller board on the receiver side which will then display the message on the LCD. On reception of the message, nurse will remotely take the required action to cater to the needs of the message.

In case of emergency the patient has to just press a push button which will signal the processing board to send an emergency alarm to the receiver. The receiver will then signal the controller to activate the buzzer. This will help the nurse to take care of the emergency as soon as possible. Taking medication at the right time is a serious business, the Medicine Reminder is another feature of this device to prompt the nurse the time to give patients their medicines.

The Medicine Reminder is intended to be used by the nurse or caretaker so that a mistake is never made in giving the medicines. The medicine reminder is implemented using a real time clock.

RECIEVER

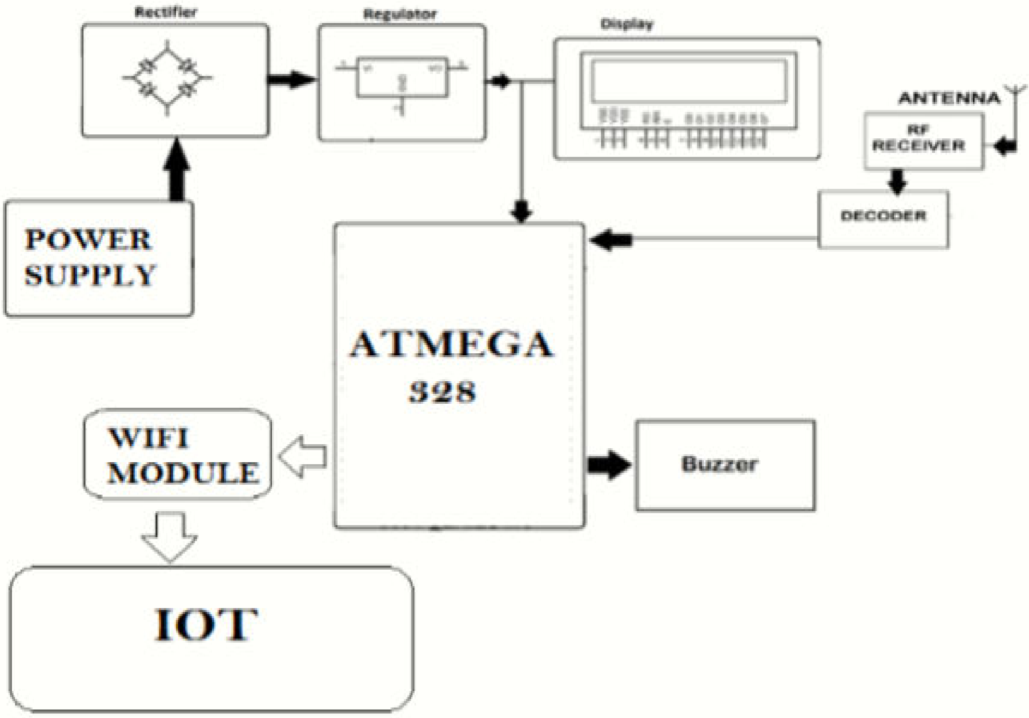

Fig.b. Receiver Block model

Usually for real time DS1307 Real time clock chip along with a battery is used, but this increases the requirement of hardware used and makes the device bulky. So we propose a system in which the time table of all the patients will be stored in the database and the nurse will be reminded automatically when its time to attend any patient according to the time table. This will be implemented by programming the controller board. On interfacing the LCD with the controller and feeding the code in the software, the real time clock runs the time on the display.

CHAPTER-2 HARDWARE SPECIFICATIONS AND CONNECTIONS

2.1.Components Required:-

1. Atmega Microcontroller Uno
2. Arduino Nano
3. Accelerometer & Gyro
4. RF Tx Rx
5. LCD Display
6. Resistors
7. Capacitors
8. Transistors
9. Cables and Connectors
10. Diodes
11. PCB and Breadboards
12. LED
13. Transformer/Adapter
14. Switch
15. IC
16. IC Sockets

2.2.Specifications:

Hardware Specification

1.Transmitter,2.Rectifier,3.Regulator,4.LCD,5.Antenna,6.RF Receiver,7.Buzzer,8.Atmega 328 Microcontroller,8051 Microcontroller,9.GSM Modem,10.Accelerometer,11.RF module Transmitter

Software specification

- Arduino Compiler
- MC Programming Language: C

Arduino Uno

The **Arduino Uno** is an open-source microcontroller board based on

the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. Fig3.1 shows arduino Genuine Uno which is Made in Italy.

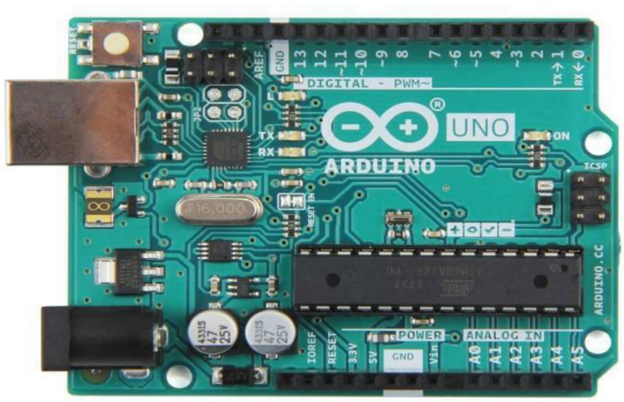

Fig2.1. Aduino Genuine Uno

ARDUINO NANO

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. 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. The ATmega328 has 32 KB, (also with 2 KB used for the bootloader. The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

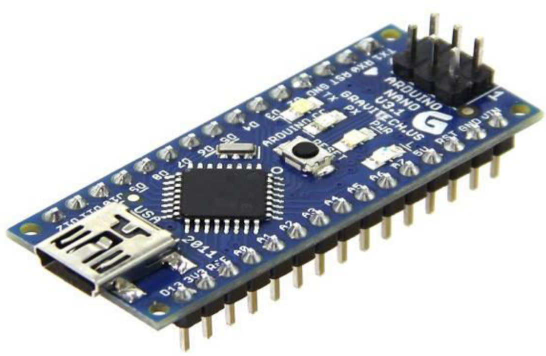

Fig.2.2.Arduino nano

BUZZER 5V DC

Buzzer is an audio signaling device that has many applications includes timers and alarm devices. Many types of buzzers are available mainly they are electromechanical, mechanical and electrical buzzers.

Fig.2.3Buzzer

RF TRANSMITTER

This wireless data is the easiest to use, lowest cost RF link we have ever seen! Use these components to transmit position data, temperature data, and even current program register values wirelessly to the receiver. These modules have up to 500 ft range in open space. The transmitter operates from 2-12V. The higher the Voltage, the greater the range .We have used these modules extensively and have been very impressed with their ease of use and direct interface to an MCU. The theory of operation is very simple. What the transmitter 'sees' on its data pin is what the receiver outputs on its data pin. If you can configure the UART module on a uC, you have an instant wireless data connection. The typical range is 500ft for open area.

This is an ASK transmitter module with an output of up to 8mW depending on power supply voltage. The transmitter is based on SAW resonator and accepts digital inputs, can operate from 2 to 12 Volts-DC, and makes building RF enabled products very easy.

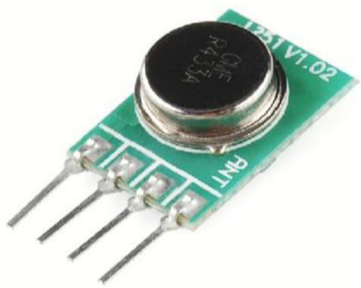

Fig.2.4.Transmitter

RF RECIEVER

This receiver type is good for data rates up to 4800bps and will only work with the 434MHz o 315 MHz transmitter. Multiple 434MHz or 315MHz receivers can listen to one 434MHz transmitter or 315 MHz transmitter.

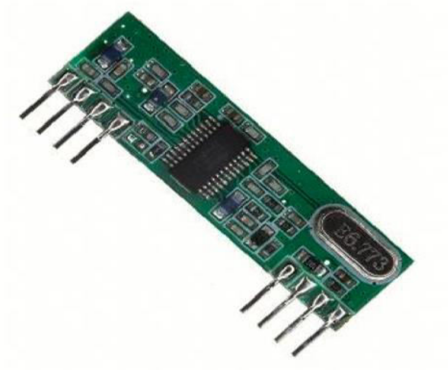

Fig.2.5.Reciever

This wireless data is the easiest to use, lowest cost RF link we have ever seen! Use these components to transmit position data, temperature data, and even current program register values wirelessly to the receiver. These modules have up to 500 ft range in open space. The receiver is operated at 5V. We have used these modules extensively and have been very impressed with their ease of use and direct interface to an MCU.

The theory of operation is very simple. What the transmitter 'sees' on its data pin is what the receiver outputs on its data pin. If you can configure the UART module on a uC, you have an instant wireless data connection. Data rates are limited to 4800bps. The typical range is 500ft for open area. This receiver has a sensitivity of 3uV. It operates from 4.5 to 5.5 volts-DC and has digital output. The typical sensitivity is -103dbm and the typical current consumption is 3.5mA for 5V operation voltage.

ACCELEROMETER & GYRO

The ADXL345-EP is an extended performance version of the ADXL345, which is a small, thin, ultralow power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 g. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I2C digital interface.

The ADXL345-EP is well suited for extended temperature range industrial and aerospace equipment. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (3.9 mg/LSB) enables measurement of inclination changes less than 1.0°.

Fig.2.5.Accelerometer

Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion by comparing the acceleration on any axis with user-set thresholds. Tap sensing detects single and double taps in any direction. Freefall sensing detects if the device is falling. These functions can be mapped individually to either of two interrupt output pins. An integrated memory management system with a 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor activity and lower overall system power consumption. Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation. The ADXL345-EP is supplied in a small, thin, 3 mm × 5 mm × 1 mm, 14-lead, enhanced plastic package.

BREAD BOARD

A **breadboard** is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate.

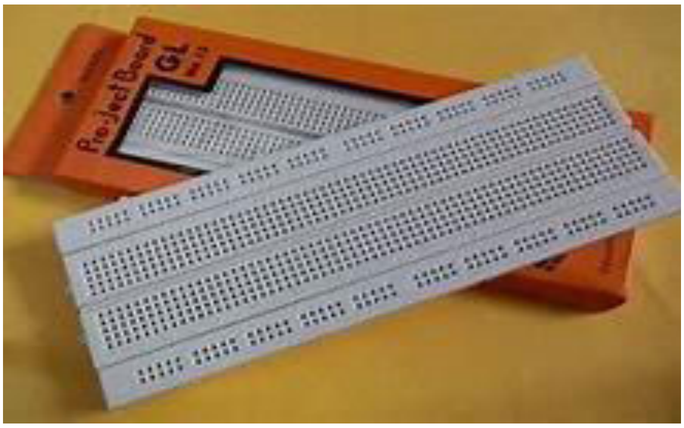

Fig.2.6.Bread board

CIRCUIT DIAGRAM & CONNECTIONS

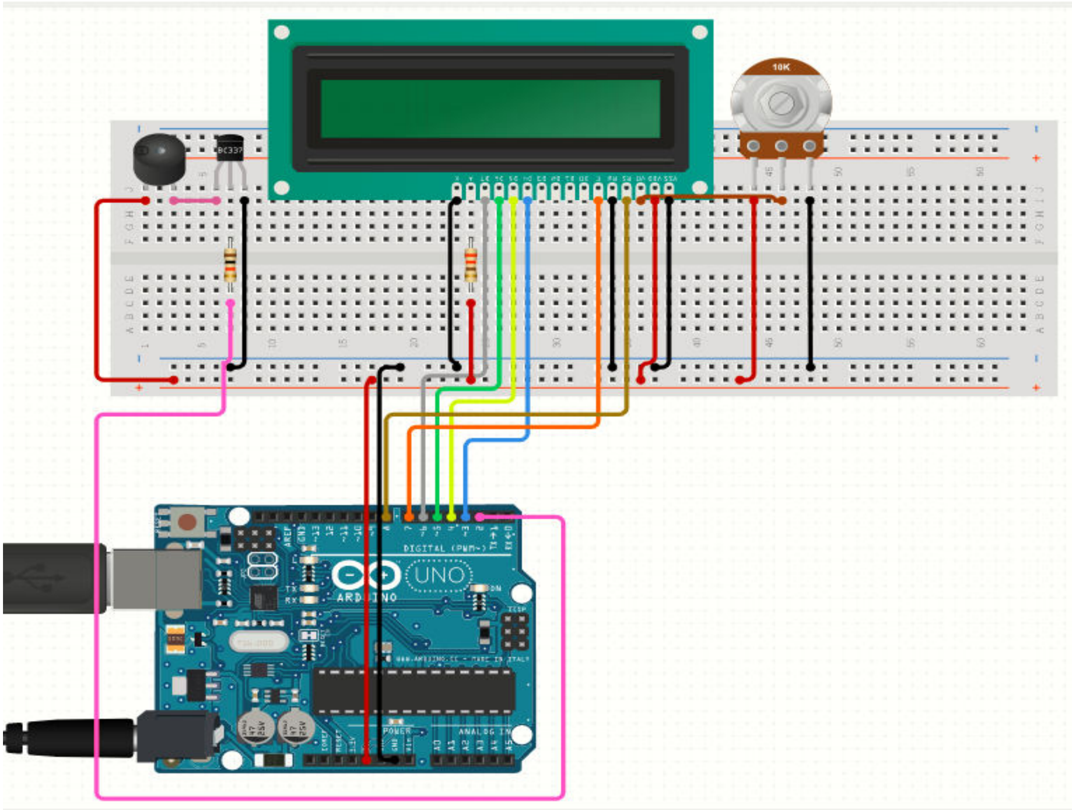

Fig.2.7.Reciever circuit

Similarly connect the VCC Pin of OLED Display to Arduino 3.3V pin and GND to GND. Connect a buzzer to Arduino digital pin 8 and other pin to GND.

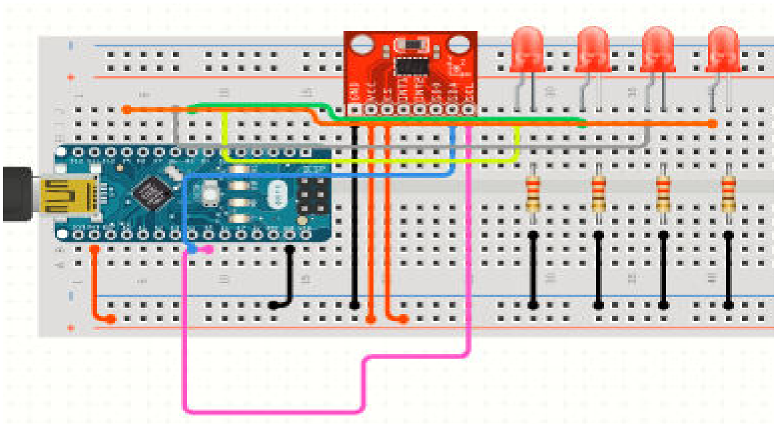

Fig.2.8.Transmitter circuit

Connect accelerometer sensor VCC pin to Arduino 5V Pin and GND to GND. Connects its signal pin to Arduino Analog pin A0. Connect its SDA & SDL pins to Arduino A4 & A5 pins respectively

CHAPTER-3 METHODOLOGY

TRANSMITTER CODE

```
#include <Wire.h>

#include <Adafruit_Sensor.h>

#include <Adafruit_ADXL345_U.h>

#include <RH_ASK.h> // Include RadioHead Amplitude Shift Keying Library

#include <SPI.h> // Include dependant SPI Library

/* Assign a unique ID to this sensor at the same time */

Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);

// Create Amplitude Shift Keying Object

RH_ASK rf_driver;

void displaySensorDetails(void)

{

  sensor_t sensor;

  accel.getSensor(&sensor);

  Serial.println("------------------------------------");

  Serial.print  ("Sensor:       "); Serial.println(sensor.name);

  Serial.print  ("Driver Ver:   "); Serial.println(sensor.version);

  Serial.print  ("Unique ID:    "); Serial.println(sensor.sensor_id);

  Serial.print  ("Max Value:    "); Serial.print(sensor.max_value); Serial.println(" m/s^2");

  Serial.print  ("Min Value:    "); Serial.print(sensor.min_value); Serial.println(" m/s^2");

  Serial.print  ("Resolution:   "); Serial.print(sensor.resolution); Serial.println(" m/s^2");

  Serial.println("------------------------------------");

  Serial.println("");

  delay(200);

}
```

```
void displayDataRate(void)
{
  Serial.print  ("Data Rate:    ");
  switch(accel.getDataRate())
  {
    case ADXL345_DATARATE_3200_HZ:
      Serial.print  ("3200 ");
      break;
    case ADXL345_DATARATE_1600_HZ:
      Serial.print  ("1600 ");
      break;
    case ADXL345_DATARATE_800_HZ:
      Serial.print  ("800 ");
      break;
    case ADXL345_DATARATE_400_HZ:
      Serial.print  ("400 ");
      break;
    case ADXL345_DATARATE_200_HZ:
      Serial.print  ("200 ");
      break;
    case ADXL345_DATARATE_100_HZ:
      Serial.print  ("100 ");
      break;
    case ADXL345_DATARATE_50_HZ:
      Serial.print  ("50 ");
      break;
```

```
case ADXL345_DATARATE_25_HZ:
  Serial.print  ("25 ");
  break;
case ADXL345_DATARATE_12_5_HZ:
  Serial.print  ("12.5 ");
  break;
case ADXL345_DATARATE_6_25HZ:
  Serial.print  ("6.25 ");
  break;
case ADXL345_DATARATE_3_13_HZ:
  Serial.print  ("3.13 ");
  break;
case ADXL345_DATARATE_1_56_HZ:
  Serial.print  ("1.56 ");
  break;
case ADXL345_DATARATE_0_78_HZ:
  Serial.print  ("0.78 ");
  break;
case ADXL345_DATARATE_0_39_HZ:
  Serial.print  ("0.39 ");
  break;
case ADXL345_DATARATE_0_20_HZ:
  Serial.print  ("0.20 ");
  break;
case ADXL345_DATARATE_0_10_HZ:
  Serial.print  ("0.10 ");
```

```
      break;
    default:
      Serial.print  ("???? ");
      break;
  }
  Serial.println(" Hz");
}
void displayRange(void)
{
  Serial.print  ("Range:         +/- ");
  switch(accel.getRange())
  {
    case ADXL345_RANGE_16_G:
      Serial.print  ("16 ");
      break;
    case ADXL345_RANGE_8_G:
      Serial.print  ("8 ");
      break;
    case ADXL345_RANGE_4_G:
      Serial.print  ("4 ");
      break;
    case ADXL345_RANGE_2_G:
      Serial.print  ("2 ");
      break;
    default:
      Serial.print  ("?? ");
```

```
      break;
  }
  Serial.println(" g");
}
void setup(void)
{
 pinMode(9,OUTPUT);
 pinMode(5,OUTPUT);
 pinMode(6,OUTPUT);
 pinMode(8,OUTPUT);
 pinMode(7,OUTPUT);
#ifndef ESP8266
  while (!Serial); // for Leonardo/Micro/Zero
#endif
  Serial.begin(9600);
  Serial.println("Accelerometer Test"); Serial.println("");
  /* Initialise the sensor */
  if(!accel.begin())
  {
    /* There was a problem detecting the ADXL345 ... check your connections */
    Serial.println("Ooops, no ADXL345 detected ... Check your wiring!");
    while(1);
  }
  /* Set the range to whatever is appropriate for your project */
  accel.setRange(ADXL345_RANGE_16_G);
  // accel.setRange(ADXL345_RANGE_8_G);
```

```
  // accel.setRange(ADXL345_RANGE_4_G);
  // accel.setRange(ADXL345_RANGE_2_G);
  /* Display some basic information on this sensor */
  displaySensorDetails()
  /* Display additional settings (outside the scope of sensor_t) */
  displayDataRate();
  displayRange();
  Serial.println("");
  // Initialize ASK Object
  rf_driver.init();
  // Setup Serial Monitor
  Serial.begin(9600);
}
void loop(void)
{
  /* Get a new sensor event */
  sensors_event_t event;
  accel.getEvent(&event);
  /* Display the results (acceleration is measured in m/s^2) */
  Serial.print("X: "); Serial.print(event.acceleration.x); Serial.print("  ");
  Serial.print("Y: "); Serial.print(event.acceleration.y); Serial.print("  ");
  Serial.print("Z: "); Serial.print(event.acceleration.z); Serial.print("  ");Serial.println("m/s^2 ");
  delay(200);
  if (event.acceleration.x>7)
  {
    digitalWrite(5,HIGH);
```

```
    digitalWrite(9,HIGH);
    delay(200);
    digitalWrite(9,LOW);
    Serial.println("Need Water");
    const char *msg = "Need Water";
    rf_driver.send((uint8_t *)msg, strlen(msg));
    rf_driver.waitPacketSent();
{
// Message Transmitted
Serial.println("Message Transmitted: ");
delay(200);
  }
  }
  else
  { Serial.println("");
  digitalWrite(5,LOW);
  }
   if (event.acceleration.y>7)
  {
    digitalWrite(6,HIGH);
    digitalWrite(9,HIGH);
    delay(200);
    digitalWrite(9,LOW);
    Serial.println("Need food");
     const char *msg = "Need food";
    rf_driver.send((uint8_t *)msg, strlen(msg));
```

```
    rf_driver.waitPacketSent();
{
// Message Transmitted
Serial.println("Message Transmitted: ");
delay(200);
  }
  }
  else
  {Serial.println("");
  digitalWrite(6,LOW);}
  if (event.acceleration.x<-7)
  {
    digitalWrite(8,HIGH);
    digitalWrite(9,HIGH);
    delay(200);
    digitalWrite(9,LOW);
    Serial.println("Need medic");
     const char *msg = "Need medic";
    rf_driver.send((uint8_t *)msg, strlen(msg));
    rf_driver.waitPacketSent();
{
// Message Transmitted
Serial.println("Message Transmitted: ");
delay(200);
  }
  }
```

```
  else
  { Serial.println("");
  digitalWrite(8,LOW);}
if (event.acceleration.y<-7)
  {
    digitalWrite(7,HIGH);
    digitalWrite(9,HIGH);
    delay(200);
    digitalWrite(9,LOW);
    Serial.println("Need help");
     const char *msg = "Need help";
    rf_driver.send((uint8_t *)msg, strlen(msg));
    rf_driver.waitPacketSent();
{
// Message Transmitted
Serial.println("Message Transmitted: ");
delay(200);
  }
  }
  else
  {Serial.println("");
  digitalWrite(7,LOW);}
  }
```

RECIEVER CODE

```
/*

Liquid crystal library

  The circuit:

 * LCD RS pin to digital pin 13

 * LCD Enable pin to digital pin 12

 * LCD D4 pin to digital pin 5

 * LCD D5 pin to digital pin 4

 * LCD D6 pin to digital pin 3

 * LCD D7 pin to digital pin 2

 * LCD R/W pin to ground

 * LCD VSS pin to ground

 * LCD VCC pin to 5V

 * 10K resistor:

 * ends to +5V and ground

 * wiper to LCD VO pin (pin 3)

#include <RH_ASK.h> // Include RadioHead Amplitude Shift Keying Library

#include <SPI.h> // Include dependant SPI Library

// Create Amplitude Shift Keying Object

RH_ASK rf_driver;

// include the library code:

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(13,12, 5, 4, 3, 2);

int buz = 9;

const int lowrange = 2000;
```

```
const int highrange = 4000;
void setup()
{
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Initialize ASK Object
rf_driver.init();
// Setup Serial Monitor
Serial.begin(9600);
pinMode(buz,OUTPUT);
}
void loop()
{
  lcd.setCursor(0,0);
// Set buffer to size of expected message
uint8_t buf[10];
uint8_t buflen = sizeof(buf);
// Check if received packet is correct size
if (rf_driver.recv(buf, &buflen))
{
// Message received with valid che-cksum
Serial.println("Message Received: ");
Serial.println((char*)buf);
lcd.println((char*)buf);
     for (int i = lowrange; i <= highrange; i++)
    {
```

```
      tone (buz, i, 200);
    }
    for (int i = highrange; i >= lowrange; i--)
    {
      tone (buz, i, 200);
    }
for (int i = lowrange; i <= highrange; i++)
    {
      tone (buz, i, 200);
    }
    for (int i = highrange; i >= lowrange; i--)
    {
      tone (buz, i, 200);
    }
    for (int i = lowrange; i <= highrange; i++)
    {
      tone (buz, i, 200);
    }
    for (int i = highrange; i >= lowrange; i--)
    {
      tone (buz, i, 200);
    }
delay(3000);
lcd.clear();
}
}
```

CHAPTER-4 RESULTS AND DISCUSSIONS

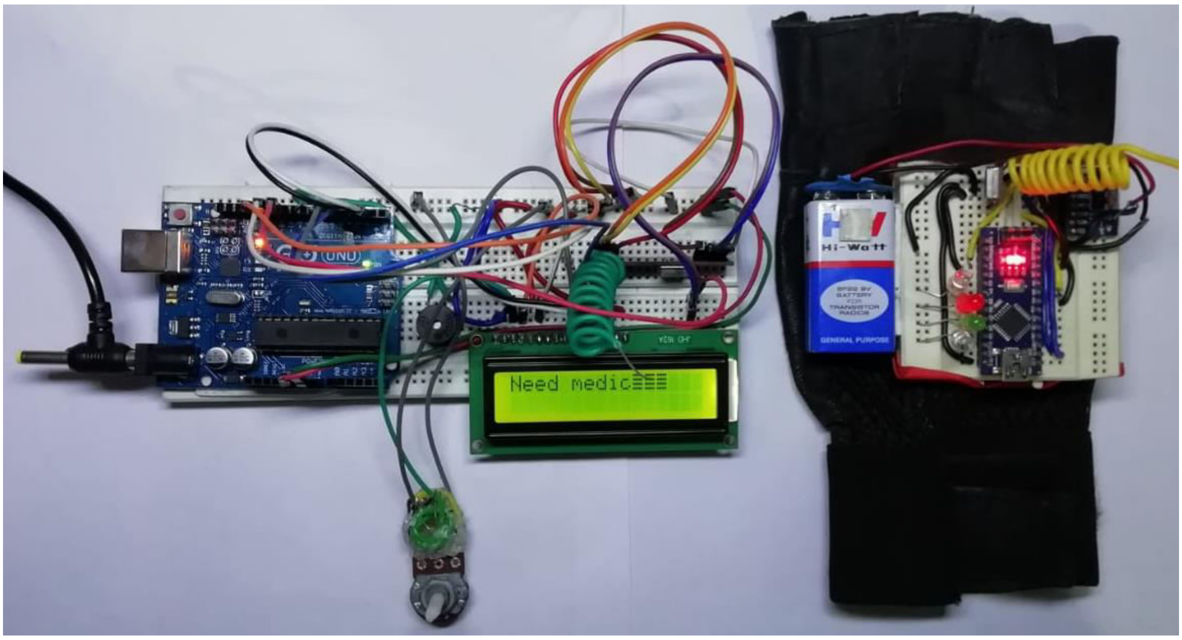

Fig.5.1.Expriment results

The user now just needs to tilt the device in a particular angle to convey a message. Tilting the device in different directions conveys a different message. Here we use accelerometer in order to measure the statistics of motion. It then passes on this data to the microcontroller. The microcontroller processes the data and displays the particular message as per input obtained.

The microcontroller now displays the associated message on the LCD screen. It also sounds a buzzer along with message as soon as it receives motion signal from the accelerometer. If there was no one to attend to the message displayed on the LCD, the patient can choose to tilt the device for some more amount of time which will trigger an SMS to be sent through a GSM modem to the registered care taker of the patient with the message that the patient wants to convey.

In this way the Automated Paralysis Patient Care System truly automates the care taking ability of the patient which ensures a timely attention to the patient and thus for a good health of the patient.

CHAPTER-5 CONCLUSION AND FUTURE SCOPE

As health care services are important part of our society, automating these services lessen the burden on humans and eases the measuring process. Also the transparency of this system helps patients to trust it. When threshold value is reached, the alarm system that consists of buzzer and LED alerts the doctors and he can act more quickly. The objective of developing monitoring systems is to reduce health care costs by reducing physician office visits, hospitalizations, and diagnostic testing procedure. The GSM technology helps the server to update the patient data on website. Many further improvements can be made in our system to make it better and easily adaptable such as adding more advanced sensors. The biometric information of the patient which is stored and published online can be given to scientists and researchers of medical fields to analyse the value and find patterns or for other research work. To simplify the hardware and reduce wiring we can use wireless sensors.

According to the availability of sensors or development in biomedical trend more parameter can be sensed and monitored which will drastically improve the efficiency of the wireless monitoring system in biomedical field. A graphical LCD can be used to display a graph of rate of change of health parameters over time. The whole health monitoring system which we have framed can be integrated into a small compact unit as small as a cell phone or a wrist watch. This will help the patients to easily carry this device with them wherever they go. In addition with medical application we can use our system in industrial and agricultural application by using sensors like humidity sensors, fertility check sensors, etc.

By taking a overall survey, it can be found that there are many problems existing for the paralyzed people such as paralysis in their leg, hand, vocal tract and also in other body parts. There are systems existing for their comforts individually. But, this system will help to monitor the needs of paralytic patients when needed. This system helps patient overcome barriers to convey their needs without putting efforts. Moreover this can be modified to be used for several purposes where person’s mobility is affected.

CHAPTER-6 REFERENCES