**REPORT**

**ON**

**“WIRELESS PATIENT HEALTH MONITORING AND ACCIDENTAL CRISIS MANAGEMENT SYSTEM’’**

**FROM**

**“BHARATI VIDYAPEETH COLLEGE OF ENGINEERING PUNE”**

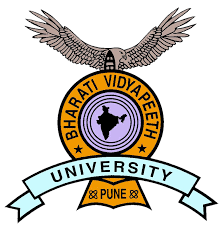
By

**RAM MISHRA**

**SIDDHANT KUMAR MISHRA**

Under the guidance of

**Prof. MEENA CHAVAN**



Department of Electronics Engineering

Bharati Vidyapeeth University

(Deemed to be)

College of Engineering,

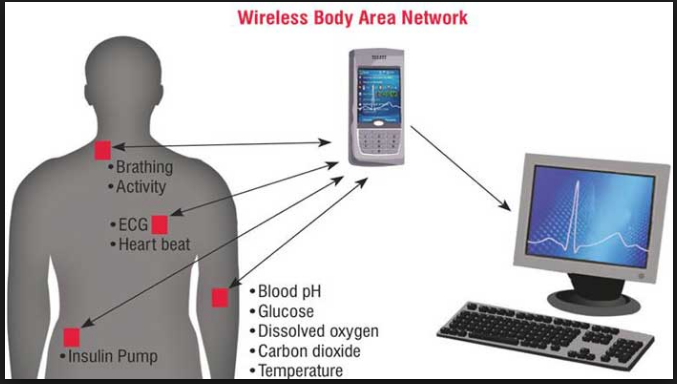
Pune-411043

**SYSTEM OVERVIEW:**

In this Wireless Patient Health Monitoring, we are monitoring various parameters of the patient using internet of things. In the patient monitoring system based on Internet of things project, the real-time parameters of patient’s health are sent to cloud using Internet connectivity. These parameters are sent to a remote Internet location so that user can view these details from anywhere in the world.

There is a major difference between [SMS based patient health monitoring](https://www.projectsof8051.com/patient-monitoring-through-gsm-modem/) and IOT based patient monitoring system. In IOT based system, details of the patient health can be seen by many users. The reason behind this is that the data needs to be monitored by visiting a website or URL. Whereas, in GSM based patient monitoring, the health parameters are sent using GSM via SMS.

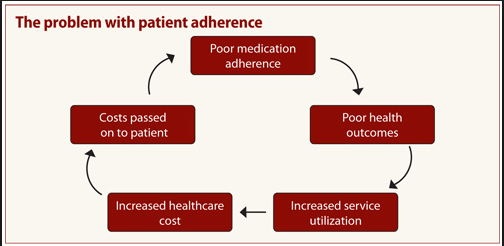
**INTRODUCTORY DIAGRAMS:**



**Figure 1: introduces the basic work chain network of wireless patient monitoring system.**

**PROBLEM STATEMENT:**

The facility for the patient is limited due to the typical instruments which are used for monitoring the real time parameters in medical field. Every time it is not possible for patients to go to the hospital and take a service. System introduced is helping us for taking care of patient by themselves as well as by doctors. Here patients not only can measure parameters related to body but also they can see it. At the same time hospital and doctor can also get patient’s health update and subsequently they can give related instructions after correct diagnosis. The wireless devices which are used these days have high performance and able to identify faults. Medical errors are eliminated due to the wireless devices. So the workload at the nurse station of the hospital is reduced.



**Fig 2. Problems encountered due to poor patient monitoring**

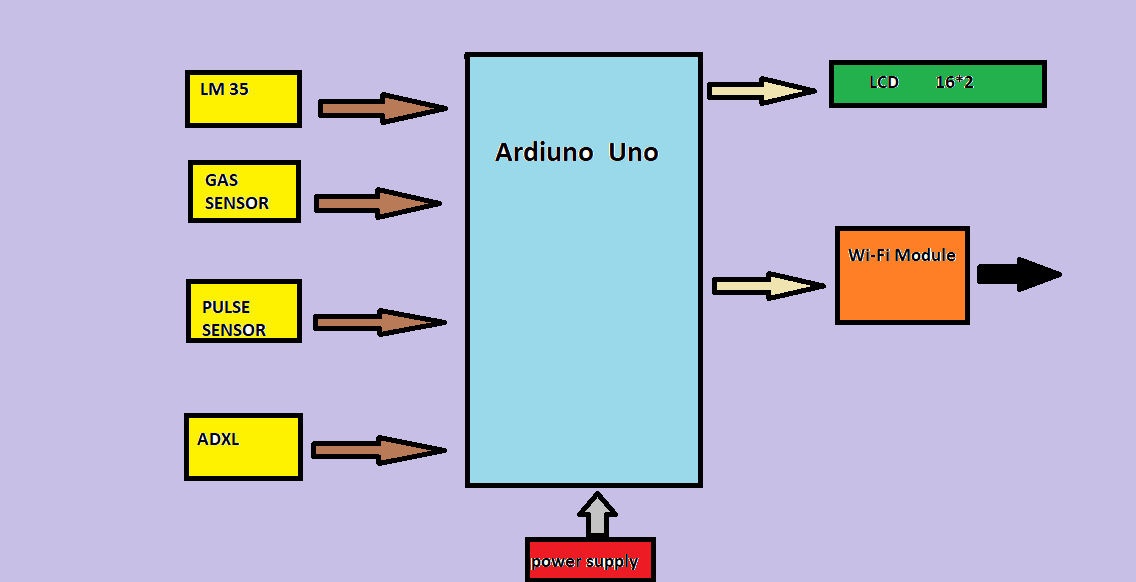
**MEDICAL SURVEILLANCE SYSTEM AS AN ANALYTICAL SOLUTIONS**

Medical health surveillance with the latest electronic technology like IOT is the need of the hour  and it is the "the continuous, systematic collection, analysis and interpretation of health-related data needed for the planning, implementation, and evaluation of [public health](https://en.wikipedia.org/wiki/Public_health) practice."

Medical Health surveillance systems can be passive or active. A passive surveillance system consists of the regular, ongoing reporting of diseases and conditions by all health facilities in a given territory. An active surveillance system is one where health facilities are visited and health care providers and medical records are reviewed in order to identify a specific disease or condition.  Passive surveillance systems are less time consuming and less expensive to run but risk under-reporting of some diseases. Active surveillance systems are most appropriate for epidemics or where a disease has been targeted for elimination.

**BLOCK DIAGRAM & DESCRIPTION**

Here is the basic overview of the block diagram (figure 3) of wireless patient monitoring system.



**Figure 3: represents the basic block diagram of wireless patient monitoring system.**

**With reference to figure: 3, following are the important outcomes:**

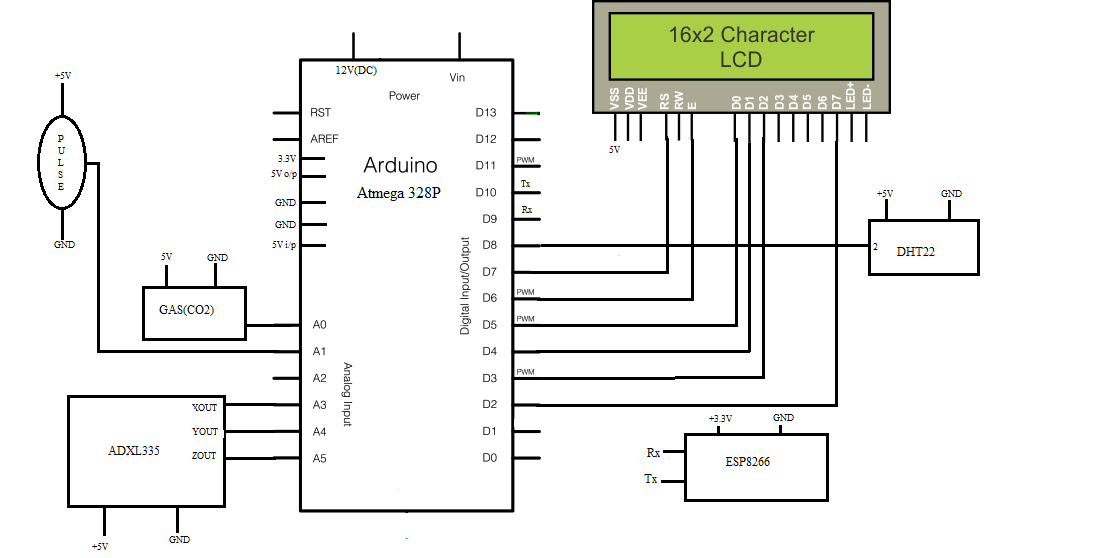
**There are 4 sensors working as input to the Arduino.**

* ADXL= ACCELEROMETER SENSOR FOR STABILITY MEASURMENT.
* PULSE SENSOR= HEART BEAT SENSOR FOR MEASURING HEART BEATS.
* GAS SENSOR FOR MEASURING THE RELATIVE CO2 DENSITY.
* DHT OR LM35 SENSOR FOR MEASURING RELATIVE TEMPERATURE & HUMIDITY.

**There are 2 modes of retrieving the output from the Arduino.**

* OUTPUT THROUGH 16\*2 LCD SCREEN.
* IOT BASED OUTPUT THROUGH WI-FI MODULE**.**

**HARDWARE DESIGN AND INTERFACING DIAGRAM:**



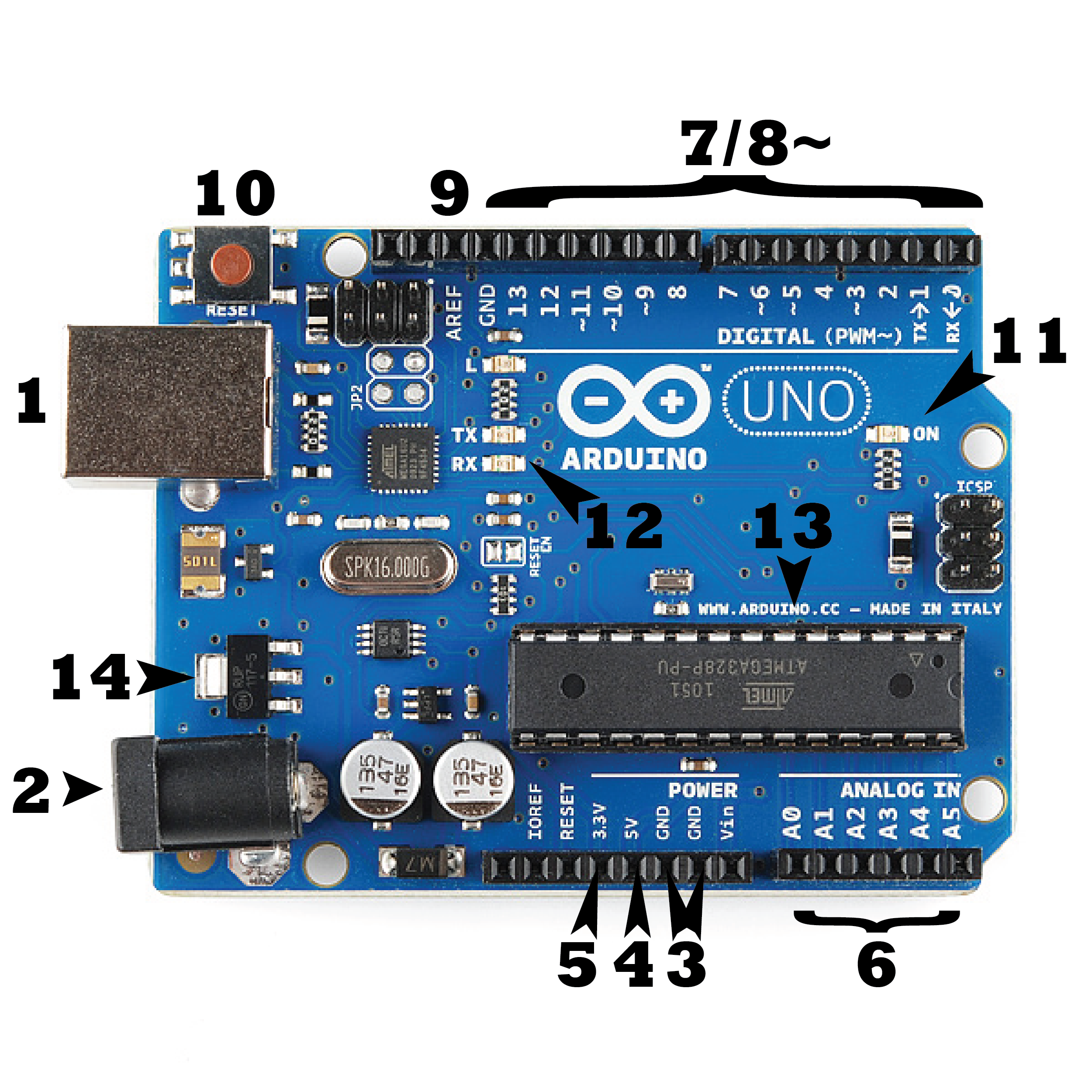
**Figure 4: represents the hardware interfacing block diagram of wireless patient monitoring system.**

With reference to the **figure 4**, we can retrieve following outcomes:

* Gas sensor is connected to the A0 pin of the Arduino.
* Accelerometer sensor is connected to A3, A4, A5 pins of the Arduino.
* Pulse sensor is connected to the A1 pin of the Arduino.
* DHT sensor is connected to D8 pin of the Arduino.
* LCD is connected to D2-D7 pins of the Arduino.

**COMPONENTS SCHEMATICS**

* **Arduino-Uno**



**Figure 5: represents pin configuration of arduino-uno**

Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

* **POWER:**

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

* **Memory**

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

* **Input and Output ports**

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode (), digital Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms

* **Programming**

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

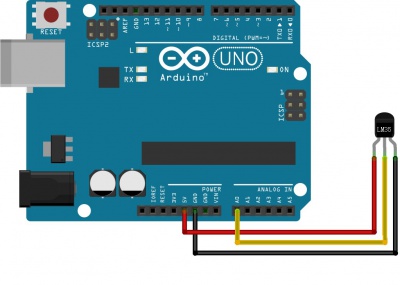
The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

* On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
* On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.
* **TEMPERATURE SENSOR**

**LM 35**



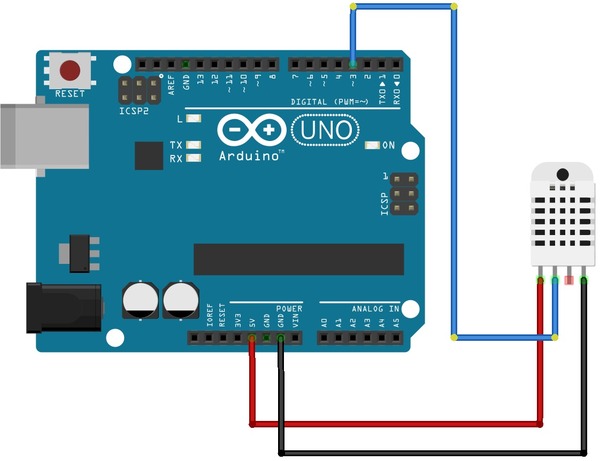
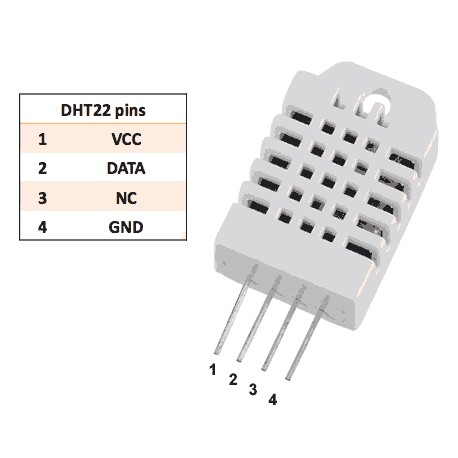
**Fig 6: connection of arduino with LM35**

The LM35 series are preciseness integrated-circuit temperature sensors, whose output voltage is

linearly proportional to the urologist (Centigrade) temperature. The LM35 so has a plus over linear temperature sensors graduated in ° Kelvin, because the user isn't needed to work out an outsized constant voltage from its output to get convenient Centigrade scaling. As shown in ***figure* 6** The LM 35 doesn't need any external activity or trimming to produce typical accuracies of ±1⁄4°C at temperature and ±3⁄4°C over a full −55 to +150°C temperature vary.

* **TEMPERATURE & HUMIDITY SENSOR**

**DHT 22**

**Fig 7: Connection of arduino with DHT22**

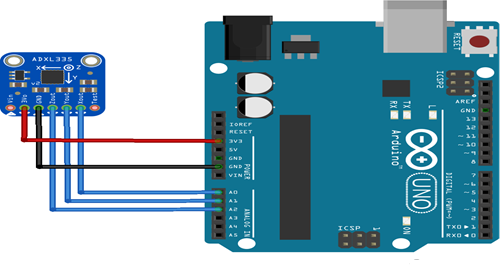
As shown in ***figure 7***, DHT22 capacitive humidity sensing digital temperature and humidity module is one that contains the compound has been calibrated digital signal output of the temperature and humidity sensors. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability. .  
The sensor includes a capacitive sensor wet components and a high-precision temperature measurement devices, and connected with a high-performance 8-bit microcontroller. The product has excellent quality, fast response, strong anti-jamming capability, and high cost.  
Standard single-bus interface, system integration quick and easy. Small size, low power consumption, signal transmission distance up to 20 meters, making it the best choice of all kinds of applications and even the most demanding applications.

* **COMBUSTIBLE GAS SENSOR**

Sensitive material of MQ-9 gas sensor is SnO2, which with lower conductivity in clean air. It make detection by method of cycle high and low temperature, and detect CO when low temperature (heated by 1.5V). The sensor’s conductivity is higher along with the gas concentration rising. When high temperature (heated by 5.0V), it detects Methane, Propane etc combustible gas and cleans the other gases adsorbed under low temperature. Please use simple electro circuit, Convert change of conductivity to correspond output signal of gas concentration. MQ-9 gas sensor has high sensitivity to Carbon Monoxide, Methane and LPG. The sensor could be used to detect different gases contains CO and combustible gases, it is with low cost and suitable for different application.

* **ACCELEROMETER**

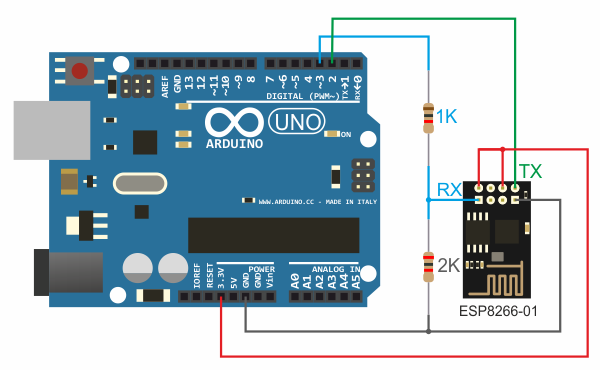
**(ADXL335)**



**Fig 8: Connection of Arduino with ADXL335**

With reference to the ***figure 8*** ,The ADXL335 is a small, thin, low power, complete 3-axis accel--erometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

* **WI-FI (ESP8266EX)**



**Fig 9: Connection Of Arduino With Esp8266ex**

ESP8266EX delivers highly integrated Wi-Fi SOC solution to meet users continuous demands for efficient power usage, compact design and reliable performance in the Internet of Things industry.

As shown in figure 9, it can work with the complete and self-contained Wi-Fi networking capabilities, ESP8266EX can perform either as a standalone application or as the slave to a host MCU. When ESP8266EX hosts the application, it promptly boots up from the flash. The integrated high speed cache helps to increase the system performance and optimize the system memory.

Also, ESP8266EX can be applied to any microcontroller design as a Wi-Fi adaptor through SPI / SDIO or I2C / UART interfaces.

**Code for Wi-Fi Module Interfacing :[12]**

**#define BLYNK\_PRINT Serial**

**#include <ESP8266WiFi.h>**

**#include <BlynkSimpleEsp8266.h>**

**#include <SimpleTimer.h>**

**char auth[] = "5330eba2fff54130b846443b89e7b013";**

**int a;**

**int b;**

**int c;**

**int d;**

**int e;**

**int f;**

**int g;**

**String inData;**

**char recieved;**

**unsigned long previousMillis;**

**SimpleTimer timer;**

**void myTimerEvent()**

**{**

**Blynk.virtualWrite(V0, millis() / 1000);**

**}**

**WidgetLCD lcd(V1);**

**int x;**

**int y;**

**void setup()**

**{**

**Serial.begin(9600);**

**Blynk.begin(auth, "AD", "ad12345678");**

**timer.setInterval(1000L, myTimerEvent);**

**}**

**void loop()**

**{**

**Blynk.run(); // Initiates Blynk**

**timer.run(); // Initiates SimpleTimer**

**rec();**

**}**

**void rec()**

**{**

**unsigned long currentMillis = millis();**

**if (currentMillis - previousMillis >= 1500)**

**{**

**if (Serial.available() > 0)**

**{**

**char recieved = Serial.read();**

**inData += recieved;**

**if (recieved == 'A')**

**{**

**a= inData.toInt();**

**inData = " ";**

**lcd.print(0,0,"H:");**

**lcd.print(2,0,a);**

**lcd.print(4,0," ");**

**}**

**if (recieved == 'B')**

**{**

**b= inData.toInt();**

**inData = " "; // Clear recieved buffer**

**lcd.print(8,0,"T:");**

**lcd.print(10,0,b);**

**lcd.print(12,0," ");**

**if(b > 36)**

**{**

**Blynk.email("Mishra.siddhant003@gmail.com", "patient alert", "TEMP over 36!,Take emergency measures");**

**Blynk.notify("patient alert- TEMP over 36");**

**}**

**}**

**if (recieved == 'C')**

**{**

**c= inData.toInt();**

**inData = " "; // Clear recieved buffer**

**Blynk.virtualWrite(V2, c);**

**if(c > 140)**

**{**

**Blynk.email("Mishra.siddhant003@gmail.com", "patient alert", "patient is in unstable state ,Take emergency measures");**

**Blynk.notify("patient is in unstable state");**

**}**

**}**

**if (recieved == 'D')**

**{**

**d= inData.toInt();**

**inData = " "; // Clear recieved buffer**

**Blynk.virtualWrite(V3, d);**

**if(d > 140)**

**{**

**Blynk.email("Mishra.siddhant003@gmail.com", "patient alert", "patient is in unstable state ,Take emergency measures");**

**Blynk.notify("patient is in unstable state");**

**}**

**}**

**if (recieved == 'E')**

**{**

**e= inData.toInt();**

**inData = " "; // Clear recieved buffer**

**Blynk.virtualWrite(V4, e);**

**if(e > 140)**

**{**

**Blynk.email("Mishra.siddhant003@gmail.com", "patient alert", "patient is in unstable state ,Take emergency measures");**

**Blynk.notify("patient is in unstable state");**

**}**

**}**

**if (recieved == 'F')**

**{**

**f= inData.toInt();**

**inData = " ";**

**lcd.print(0,1,"G:");**

**lcd.print(2,1,f);**

**lcd.print(5,1," ");**

**}**

**if (recieved == 'G')**

**{**

**g= inData.toInt();**

**inData = " ";**

**lcd.print(8,1,"BPM:");**

**lcd.print(12,1,g);**

**lcd.print(15,1," ");**

**if(g > 90)**

**{**

**Blynk.email("Mishra.siddhant003@gmail.com", "patient alert", "BPM over 90!,Take emergency measures");**

**Blynk.notify("patient alert- BPM over 90");**

**}**

**}**

**}**

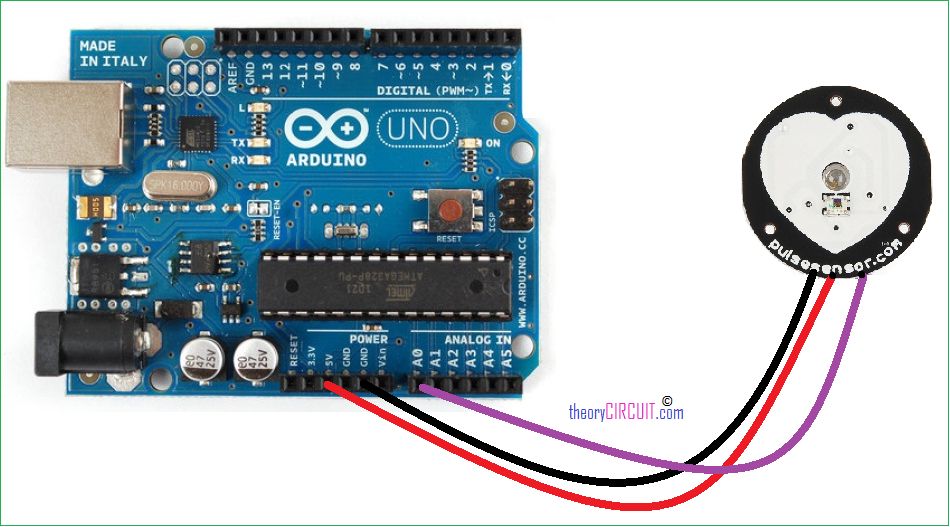
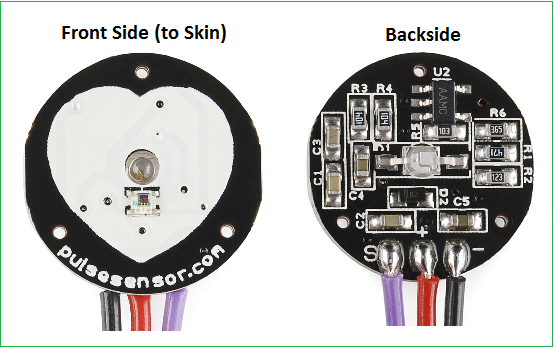
**recieved =' ';**

**}**

**}**

* **PULSE SENSOR**

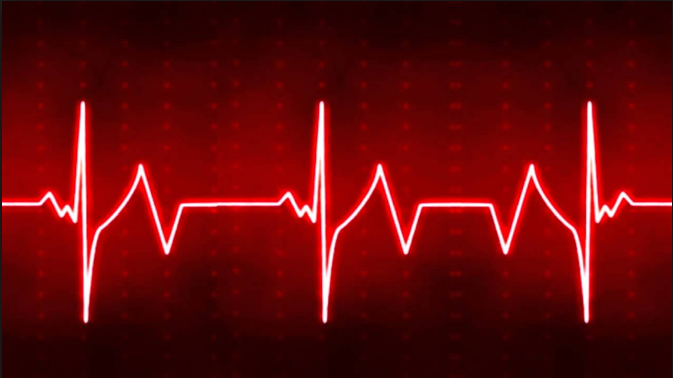
**(APDS-9008):**

**Fig 10 : Connection Of Arduino With Apds-9008:**

The APDS-9008 is a low cost analog-output ambient light photo sensor in miniature chip LED lead-free surface mount package. It consists of a spectrally suited photo sensor, which provides excellent responsivity that is close to the response of the human eyes. **Referring figure 10,** The APDS-9008 is ideal for applications in which the measurement of ambient light is used to control display backlighting. Mobile appliances such as the mobile phones and PDAs that draw heavy current from display backlighting will benefit from incorporating these photo sensor products in their designs by reducing power consumption significantly.

**FEATURES:**

 **Figure 11: Represent the Features Of a Normal Heartbeat Wave [10]**

• Excellent responsivity

Close responsivity to the human eye

• Miniature ChipLED Leadfree surface-mount package

Height – 0.55 mm

Width – 1.60 mm

Depth – 1.50 mm

• Low sensitivity variation across various light sources

• Operating temperature : -40°C to 85°C

• Vcc supply 1.6 to 5.5V

• Lead-free package, RoHS compliance

• Output linearity across wide illumination range

• High output saturation voltage

* **LCD DISPLAY:**

As shown in figure 14,It has 16 pins and the first one from left to right is the Ground pin. The second pin is the VCC which we connect the 5 volts pin on the Arduino Board. Next is the Vo pin on which we can attach a potentiometer for controlling the contrast of the display.

Next, The RS pin or register select pin is used for selecting whether we will send commands or data to the LCD. For example if the RS pin is set on low state or zero volts, then we are sending commands to the LCD like: set the cursor to a specific location, clear the display, turn off the display and so on. And when RS pin is set on High state or 5 volts we are sending data or characters to the LCD.

**16\*2 LCD Type**



**Figure 12: represents the pin configuration of a 16\*2 LCD**

Next comes the R / W pin which selects the mode whether we will read or write to the LCD. Here the write mode is obvious and it is used for writing or sending commands and data to the LCD. The read mode is used by the LCD itself when executing the program which we don’t have a need to discuss about it in this tutorial.

Next is the E pin which enables the writing to the registers, or the next 8 data pins from D0 to D7. So through this pins we are sending the 8 bits data when we are writing to the registers or for example if we want to see the latter uppercase A on the display we will send 0100 0001 to the registers according to the ASCII table.

And the last two pins A and K, or anode and cathode are for the LED back light.

**ADVANTAGES**

1. Long distance communication due to radio frequency used**.**
2. Remote monitoring and access.
3. Here the system is reconfigurable.
4. Lots of functions done by a single arduino .
5. Low cost and high performance speed.



**APPLICATIONS:**

Industrial area

By doing some changes we can use same system in electric furnace to control the operation of plants. Only the thing we have to do is change the sensors.

Medical field

In big hospitals it is not easy to see physiological parameters of patient each and every time. This system gives alert to doctor as well as staff. S it is easy to give quick service to particular patient by particular doctor.

**RESULTS AND ANALYSIS**

**Result on Hardware:**



**Fig:13 RESULT IMAGE 1: REPRESENTS THE OUTPUT ON THE LCD SCREEN**

**WITH REFRENCE TO THE RESULT IMAGE 1, THE FOLLOWING OUTCOMES CAN BE RETRIEVED:**

H= RELATIVE HUMIDITY. ( NORMAL VALUE = 40-60)

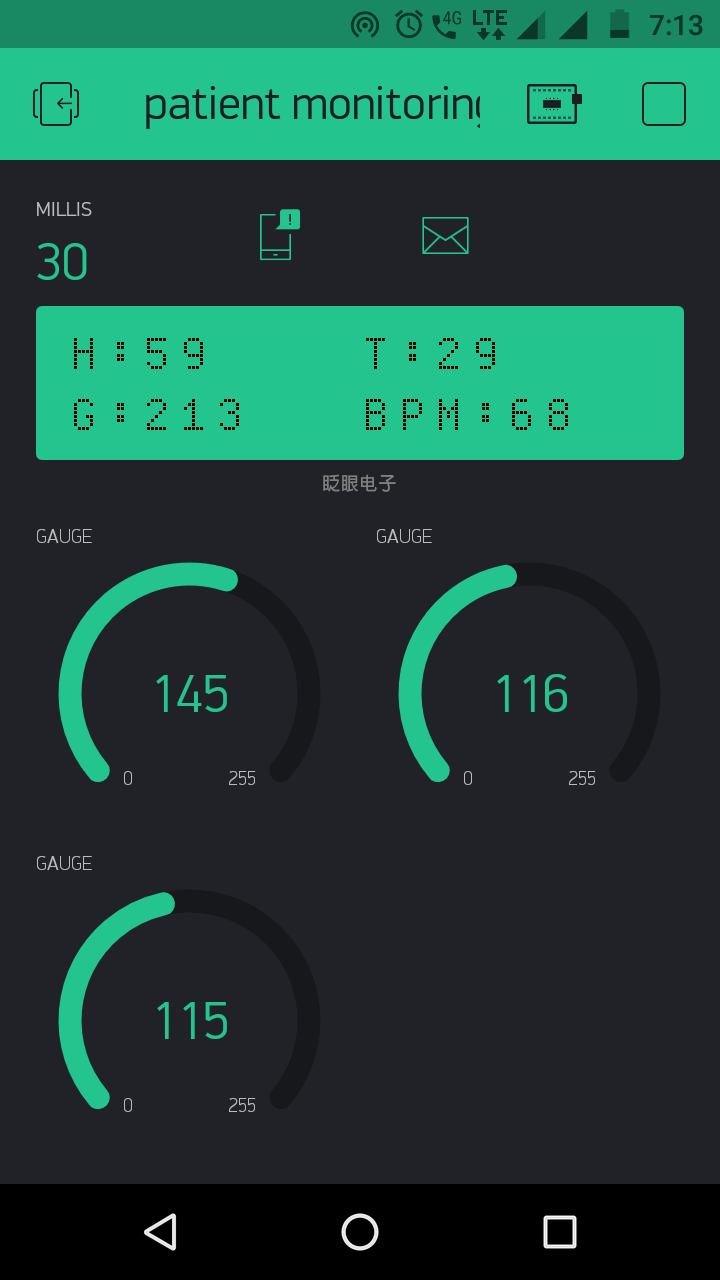
g = RELATIVE VALUE OF SPO2 IN ATMOSPHERE (NORMAL VALUE = >115)

T = RELATIVE ROOM TEMPERATURE ( NORMAL VALUE= 30.00-39.00)

**Virtual Results:**

**Here we have made use of the blynk mobile application to display the virtual results.**

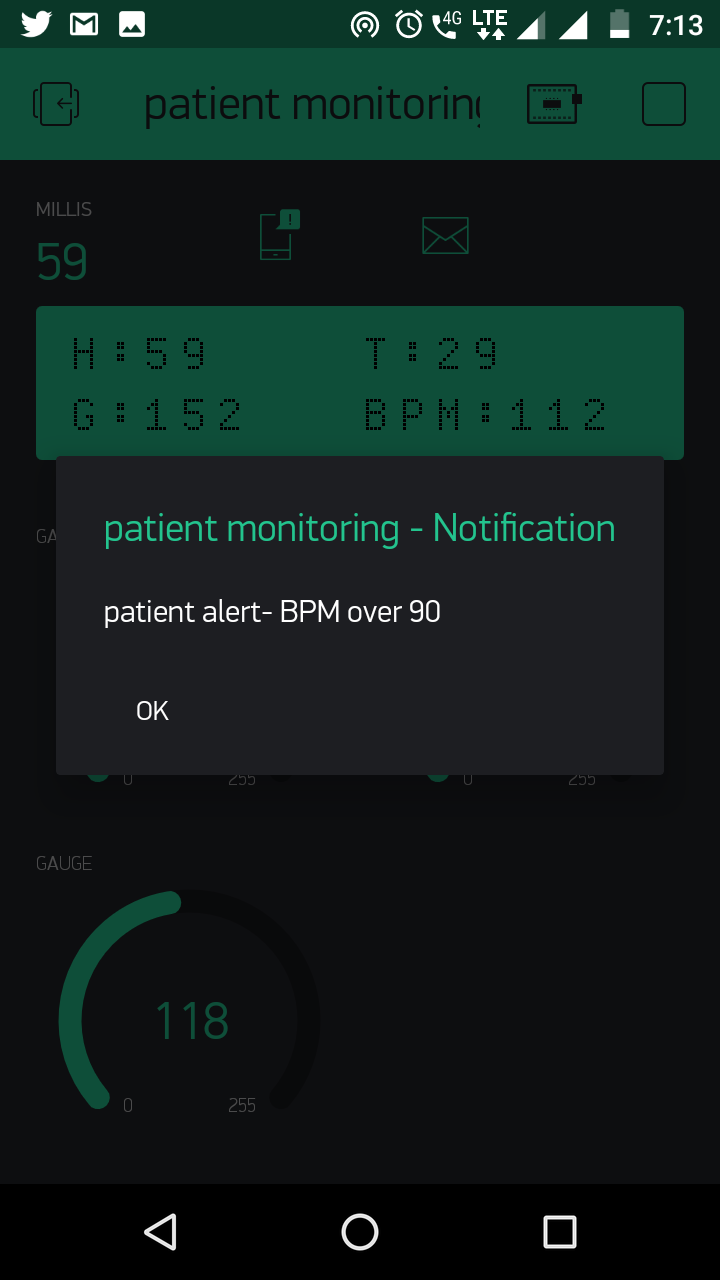
**Blynk is a App** for iOs and Android to connect Arduino, Raspberry Pi and similar hardware.[13]



**Fig:14 Result image 2: represents the virtual LCD screen and values of accelerometer sensor used in the project.**

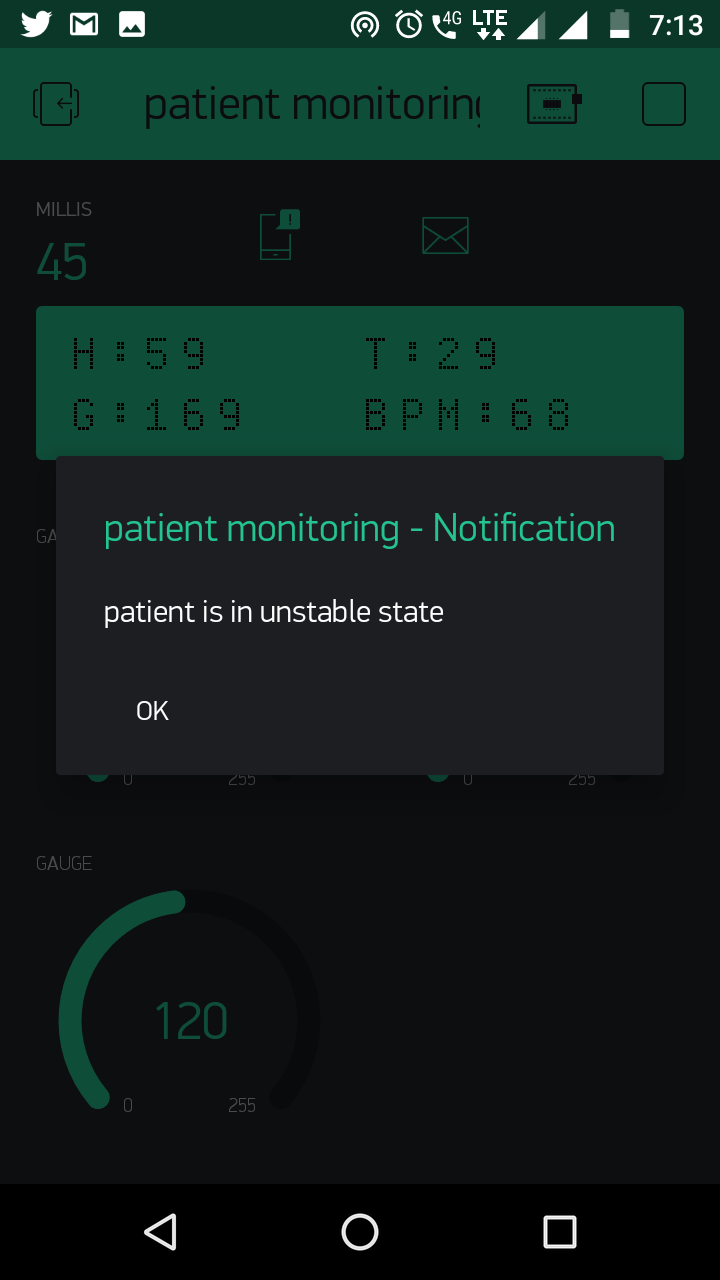
With reference to result image 2, the values of the three quadrants varies as per the change in the posture of the patient’s body.

**Warning results:**



**Fig15: Result image 3: represents the warning when the patient’s pulse rate is more than 90 BPM.**

With reference to result image 3, a warning message can be seen alerting the doctor or the family member of the patient to take appropriate actions.



**Fig16: Result image 4: represents the warning state du to the inappropriate posture of the patient.**

With reference to the result image 4, a warning message is displayed to show that the patient has either fell down to alert the doctors or the family members to take appropriate actions.

**FUTURE SCOPES:**

1. In future for day to day life everyone can use the system to take care of them. They can measure live data, send it to their doctor and they can receive instant reply about “what they have to do next”. Life of people will be saved by giving primary help to them through message.
2. Wireless Sensor Actor Networks (WSANs) are emerging as a new generation of sensor networks. In Wireless Sensor Actor Networks, sensors gather information about the physical world, while actors take decisions and then perform appropriate actions upon the environment, which allows remote, automated interaction with the environment. The presence of a single actuator in sensor networks eliminates the need for coordination and communication between actuators and a sparsely connected network eliminates the need for location management. The fourth extension is to include WSANs in remote monitoring of patients to automatically actuate the devices such as the defibrillator, drug delivery system, muscle stimulator, etc., when life threatening events occur. Our current work can also extend further to monitor sports personalities and patients affected by other specific diseases during their normal routine activities.