PRPJECT-3

WELLNESS ANALYSING SYSTEM

**NIVEDA V-18BEE052**

**PRIYADHARSHINI R-18BEE036**

# PROBLEM STATEMENT:

In times of COVID we have special Covid 19 Quarantine centers setup in order to treat covid patients. Since covid is highly infectious it is very important to quarantine covid patients but at the same time doctors need to monitor health of covid patients too. With the increasing number of cases it is becoming difficult to keep a track on the health conditions of many quarantined patients.

The problems here are:  
• Doctors need to regularly monitor patient health.  
• There are increasing number of patients for the doctors to monitor.  
• The doctors are at risk of infection just for monitoring purpose.

**OBJECTIVE OF PROBLEM STATEMENT:**

* To Solve this issue we here design a remote based  wellness analysing system that allows for remptely monitoring of multiple covid patients.
* The system monitors patient heartbeat, temperature ,blood oxygen and blood pressure.
* The data  can be able  to analyse and monitor by doctors as well as the concerned users.

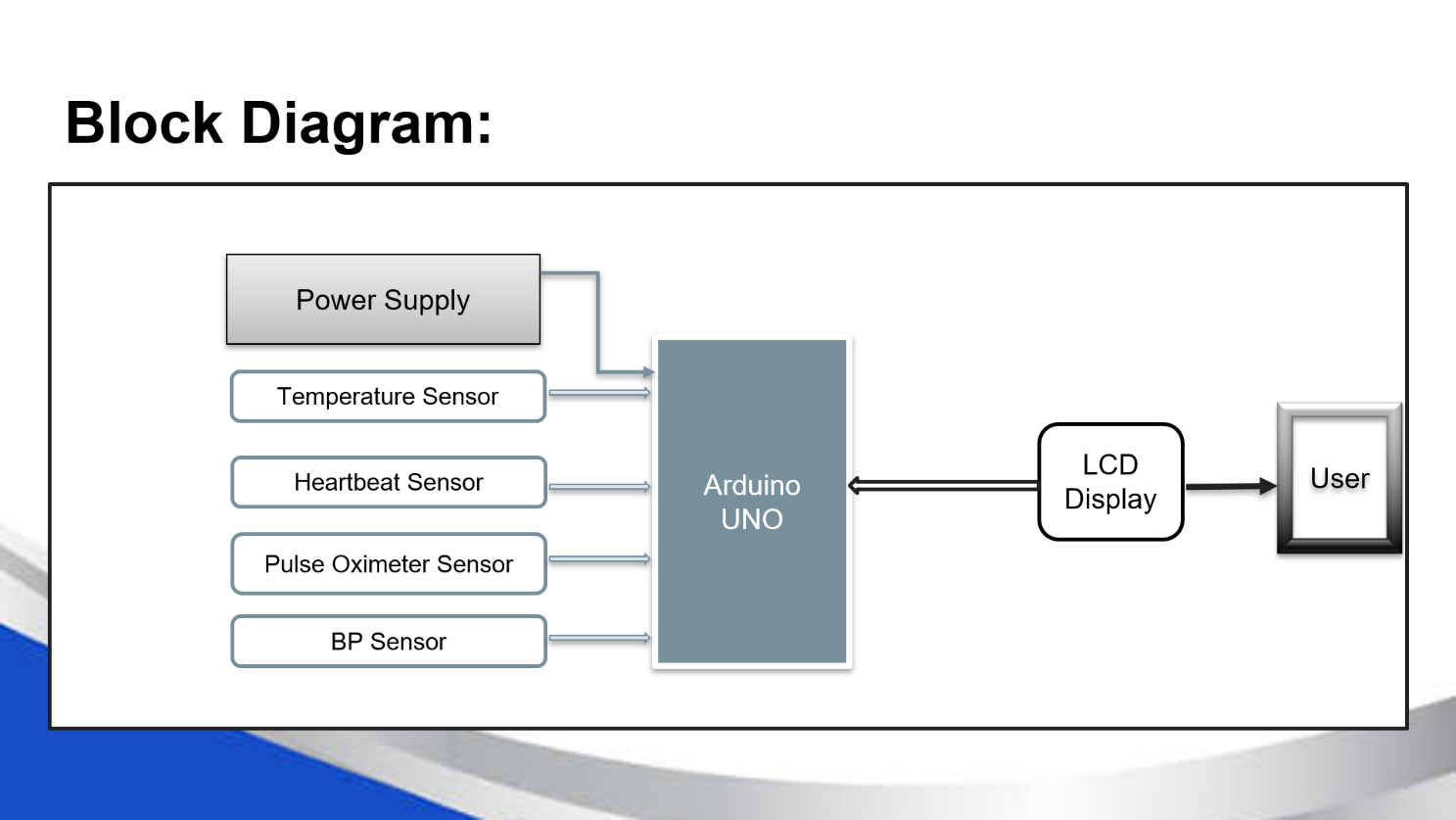
**SOLUTION:**

Our proposed system will solve the problem in the following methods,

* Provide alternative to traditional management of patients reducing of hospitalization.
* Reduce cost of formal health care.
* Early detection of disease , reduce suffering and medical cost.
* Easy continuous analyzing and monitoring
* Developing a web page,they can easily get the basics information from the app.

[](http://nevonprojects.com/wp-content/uploads/2020/10/IOT-Quarantine-center-health-checker-3d-view-3.jpg)

Self Quarantine Method

****

**CONFIGURATION OF THE PROJECT:**

The system structure represents the connection of the individual components,

* Power Supply
* Temperature Sensor
* Heartbeat Sensor
* Pulse Oximeter Sensor
* BP sensor
* LCD Display

Hardware Simulation-Proteus Software

Software Coding-Arduino IDE

**SOFTWARE**-Web page developed in Sublime Text Editor

Languages Used for developing a Website are,

* HTML

-Hypertext Markup Language is standard language for documents designed to be displayed in a browser.

* CSS

-Cascading Style Sheets is used describing the presentation of Web page,including colors,layout and fonts.

* JAVASCRIPT

-Javascipt defines the functionality of the Web page.

**DESCRIPTION OF COMPONENTS:**

* **Temperature Sensor**

**-**LM35**,**Calibrated in degree celsius, High accuracy, measures –5degree to 150degree Celsius

* **Heartbeat Sensor:**

-Heart beat sensor that will tell us the number of pulses in a minute

* **Pulse Oximeter Sensor:**

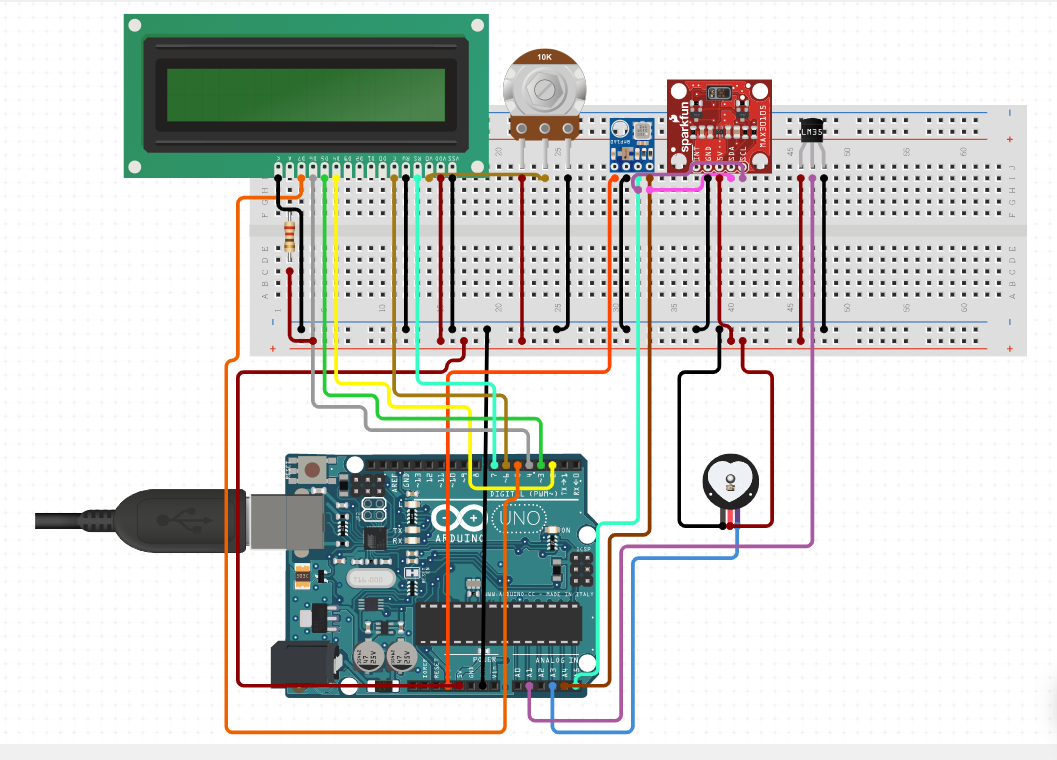
**-**The MAX30100 is an integrated pulse oximetry and heartrate monitor sensor,internally rechargeable,operates from 1.8V and 3.3V

* **BP Sensor:**

**-**The MAX30100 is an integrated pulse oximetry and heartrate monitor sensor,internally rechargeable,operates from 1.8V and 3.3V.

* **LCD Display:**

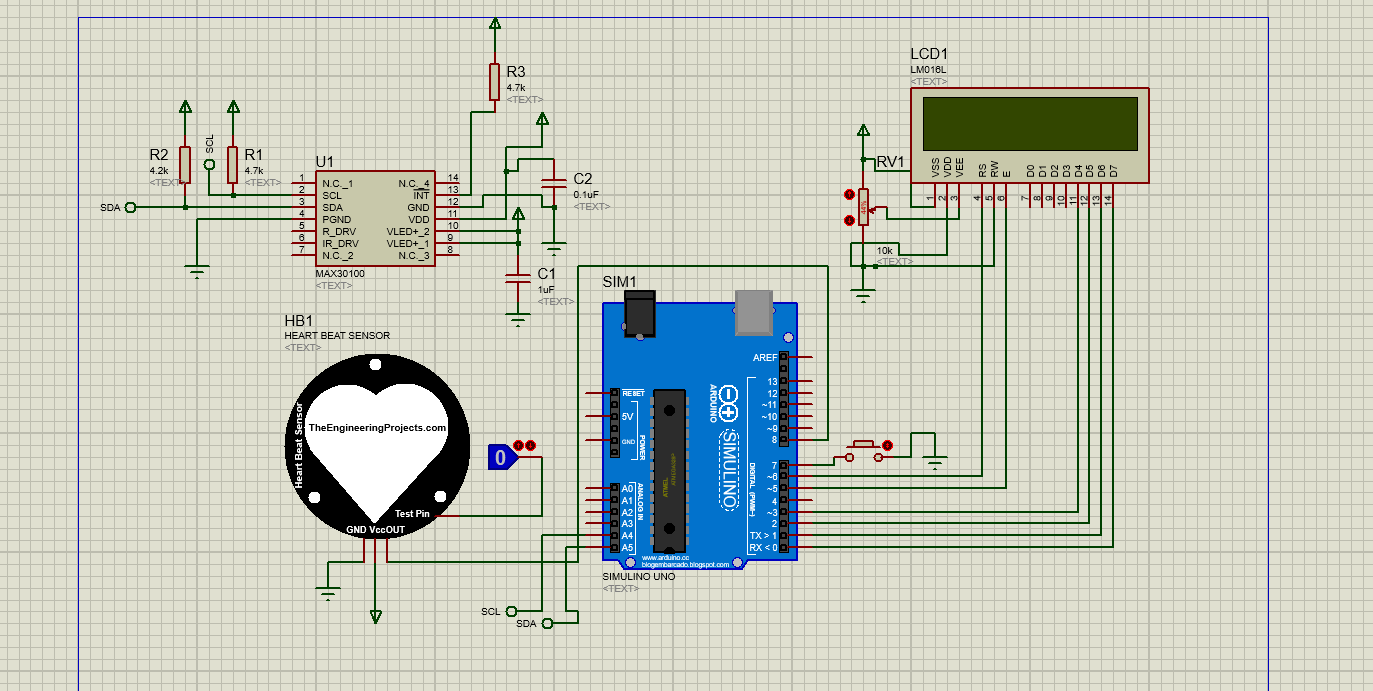
**-**16\*2 LCD Display consists of two rows and each row can print 16 characters, each character is build by a 5×8 pixel box.

**CIRCUIT DIAGRAM:**

**SIMULATION RESULTS:**

**Chart

Description automatically generatedInterfacing with Heartbeat Sensor:**

**Interfacing with MAX30100 with Heartbeat Sensor:**

A screenshot of a computer

Description automatically generated**Interfacing with LM35 and BMP180:**

**CODING PART:**

**-part 1**

#include <LiquidCrystal.h>

LiquidCrystal lcd(0,1,2,3,5,6);

int data=A0;

int start=7;

int count=0;

unsigned long temp=0;

byte customChar1[8] = {0b00000,0b00000,0b00011,0b00111,0b01111,0b01111,0b01111,0b01111};

byte customChar2[8] = {0b00000,0b11000,0b11100,0b11110,0b11111,0b11111,0b11111,0b11111};

byte customChar3[8] = {0b00000,0b00011,0b00111,0b01111,0b11111,0b11111,0b11111,0b11111};

byte customChar4[8] = {0b00000,0b10000,0b11000,0b11100,0b11110,0b11110,0b11110,0b11110};

byte customChar5[8] = {0b00111,0b00011,0b00001,0b00000,0b00000,0b00000,0b00000,0b00000};

byte customChar6[8] = {0b11111,0b11111,0b11111,0b11111,0b01111,0b00111,0b00011,0b00001};

byte customChar7[8] = {0b11111,0b11111,0b11111,0b11111,0b11110,0b11100,0b11000,0b10000};

byte customChar8[8] = {0b11100,0b11000,0b10000,0b00000,0b00000,0b00000,0b00000,0b00000};

void setup()

{

lcd.begin(16, 2);

lcd.createChar(1, customChar1);

lcd.createChar(2, customChar2);

lcd.createChar(3, customChar3);

lcd.createChar(4, customChar4);

lcd.createChar(5, customChar5);

lcd.createChar(6, customChar6);

lcd.createChar(7, customChar7);

lcd.createChar(8, customChar8);

pinMode(data,INPUT);

pinMode(start,INPUT\_PULLUP);

}

void loop()

{

lcd.setCursor(0, 0);

lcd.print("Place The Finger");

lcd.setCursor(0, 1);

lcd.print("And Press Start");

while(digitalRead(start)>0);

lcd.clear();

temp=millis();

while(millis()<(temp+10000))

{

if(analogRead(data)<100)

{

count=count+1;

lcd.setCursor(6, 0);

lcd.write(byte(1));

lcd.setCursor(7, 0);

lcd.write(byte(2));

lcd.setCursor(8, 0);

lcd.write(byte(3));

lcd.setCursor(9, 0);

lcd.write(byte(4));

lcd.setCursor(6, 1);

lcd.write(byte(5));

lcd.setCursor(7, 1);

lcd.write(byte(6));

lcd.setCursor(8, 1);

lcd.write(byte(7));

lcd.setCursor(9, 1);

lcd.write(byte(8));

while(analogRead(data)<100);

lcd.clear();

}

}

lcd.clear();

lcd.setCursor(0, 0);

count=count\*6;

lcd.setCursor(2, 0);

lcd.write(byte(1));

lcd.setCursor(3, 0);

lcd.write(byte(2));

lcd.setCursor(4, 0);

lcd.write(byte(3));

lcd.setCursor(5, 0);

lcd.write(byte(4));

lcd.setCursor(2, 1);

lcd.write(byte(5));

lcd.setCursor(3, 1);

lcd.write(byte(6));

lcd.setCursor(4, 1);

lcd.write(byte(7));

lcd.setCursor(5, 1);

lcd.write(byte(8));

lcd.setCursor(7, 1);

lcd.print(count);

lcd.print(" BPM");

temp=0;

while(1);

}

**Part-2**

#include <Wire.h>

#include <SFE\_BMP180.h>

#include<LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 10, 11, 12, 13);//RS,EN,D4,D5,D6,D7

float temp;

char PRESSURESHOW[4];// initializing a character of size 4 for showing the result

char TEMPARATURESHOW[4];// initializing a character of size 4 for showing the temparature result

SFE\_BMP180 bmp;

float alt = 5.0; // Altitude of current location in meters

void setup() {

pinMode(A0, INPUT); //sensor

lcd.begin(16, 2);

// Print a logo message to the LCD.

lcd.print(" BLOOD PRESSURE ");

lcd.setCursor(0, 1);

delay (2500);

lcd.clear();//clear display

Serial.begin(9600);

analogReference(INTERNAL);

bool success = bmp.begin();

if (success) {

Serial.println("BMP180 init success");

}

}

void loop() {

char status;

double T, P, seaLevelPressure;

bool success = false;

status = bmp.startTemperature();

if (status != 0) {

delay(1000);

status = bmp.getTemperature(T);

if (status != 0) {

status = bmp.startPressure(3);

if (status != 0) {

delay(status);

status = bmp.getPressure(P, T);

if (status != 0) {

seaLevelPressure = bmp.sealevel(P, alt);

lcd.print("Pressure= ");

lcd.print(PRESSURESHOW);

lcd.print(" hPa ");

delay(1000);

lcd.clear();

temp = analogRead(A1); //temp = temp \* 0.48828125; //temp=temp\*(5.0/1023.0)\*100;

temp=temp\*1100/(1024\*10);

lcd.print("TEMP: ");

lcd.print(temp);

lcd.println("\*C");

delay(500);

lcd.clear();

}}}}}

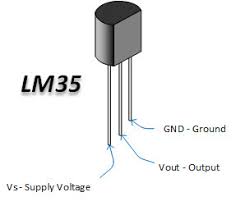
Graphical user interface, text, application

Description automatically generated**WEB PAGE:**

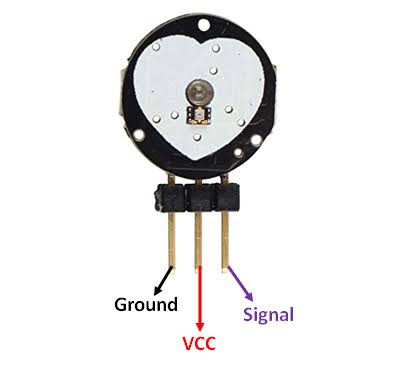
Graphical user interface, text, application, email

Description automatically generated

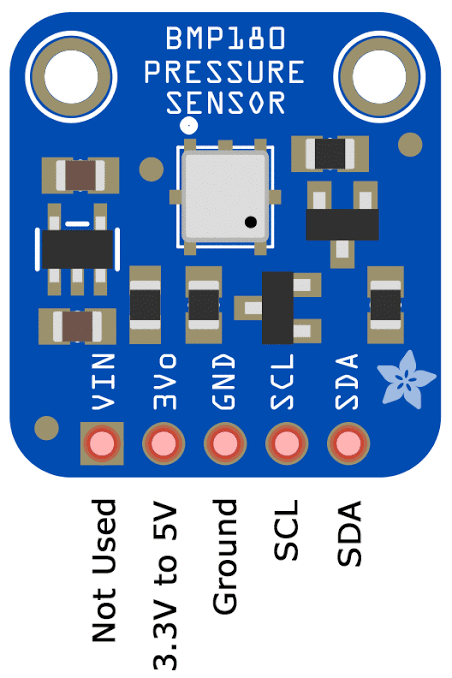
**SENSORS AND ITS SPECIFICATIONS:**

**1.LM35 Temperature Sensor**

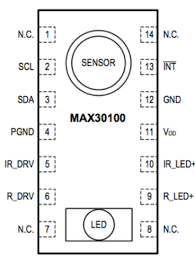
* Minimum and Maximum Input Voltage is 35V and -2V respectively.
* Can measure temperature ranging from -55°C to 150°C.
* Output voltage is directly proportional (Linear) to temperature (i.e.) there will be a rise of 10mV (0.01V) for every 1°C rise in temperature.
* ±0.5°C Accuracy.

**2.Heartbeat Sensor**

* Biometric **Pulse** Rate or **Heart Rate** detecting **sensor**.
* Plug and Play type **sensor**.
* Operating Voltage: +5V or +3.3V.
* Current Consumption: 4mA.
* Inbuilt Amplification and Noise cancellation circuit.

**3.BMP180-Pressure Sensor**

* Operating voltage of **BMP180**: 1.3V – 3.6V.
* Input voltage of BMP180MODULE: 3.3V to 5.5V.
* Peak current : 1000uA.
* Consumes 0.1uA standby.
* Maximum voltage at SDA , SCL : VCC + 0.3V.
* Operating temperature: -40ºC to +80ºC.

**4.Pulse Oximeter Sensor-MAX30100**

* Operating Voltage: 1.8v – 5.5v.
* Interface Type: I2C Serial Interface.
* Module **Dimensions**: 18.8mm (L) x 14.4mm (W) x 3.0mm (H)
* Module **Weight**: 1.2g (Header + module)

**APPLICATION:**

* Doctors to monitor patients remotely without risk of infection
* A single doctor over 500 patients at a time.
* Doctor gets instant alert in case of health fluctuations of emergency.
* The system is mounted at patient bedside and constantly transmits patient health data over the internet so that doctors can monitor multiple patients remotely and attend the desired patient urgently when needed.

**FUTURE SCOPE:**

* Data can be analysed using trained databases and the required action can be performed
* A band can be developed which is very compact to use
* App can be made to send a consolidated report to the assigned Doctor