```
  #define USE_ARDUINO_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM
math.
#include <PulseSensorPlayground.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
//  Variables
const int PulseWire = 0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0
const int LED13 = 13;          // The on-board Arduino LED, close to PIN 13.
int Threshold = 550;           // Determine which Signal to "count as a beat" and which to ignore.
                               // Use the "Gettting Started Project" to fine-tune Threshold Value beyond default
setting.
                               // Otherwise leave the default "550" value.

//const int rs = 12, en = 13, d4 = A2, d5 = A3, d6 = A4, d7 = A5;
//LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

//LiquidCrystal lcd(12, 7, A1, A2, A3, A4);

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

PulseSensorPlayground pulseSensor;  // Creates an instance of the PulseSensorPlayground object
called "pulseSensor"

SoftwareSerial mySerial(10, 9);  //10-->GsmTX    11-->GSMRx


void SendMessage()
{
  Serial.println("low bp detected");

  Serial.println("Sending Message");

  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode

  delay(1000);  // Delay of 1000 milli seconds or 1 second

  mySerial.println("AT+CMGS=\"+919481437413\"\r"); // Replace x with mobile number

   //gsmSerial.println("AT+CMGS=\"9880107078\"\r"); // Replace x with mobile number

  delay(1000);

  mySerial.println("Consult Doctor: low bp  Detected");// The SMS text you want to send

  delay(100);

 mySerial.println((char)26);// ASCII code of CTRL+Z

  delay(1000);



}





void SendMessagelowbp()

{


  Serial.println("low BPM rate detected");

  Serial.println("sending message");

  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode

  delay(1000);  // Delay of 1000 milli seconds or 1 second

  mySerial.println("AT+CMGS=\"+919481437413\"\r"); // Replace x with mobile number
```

```
   //gsmSerial.println("AT+CMGS=\"7204099395\"\r"); // Replace x with mobile number
  delay(1000);
  mySerial.println("drink salt water immediately");// The SMS text you want to send
  delay(100);
 mySerial.println((char)26);// ASCII code of CTRL+Z
  delay(1000);



}

void setup() {
lcd.begin(16, 2);

   mySerial.begin(9600);   // Setting the baud rate of GSM Module
   Serial.begin(9600);    // Setting the baud rate of Serial Monitor (Arduino)

  // Configure the PulseSensor object, by assigning our variables to it.
  pulseSensor.analogInput(PulseWire);
//  pulseSensor.blinkOnPulse(LED13);       //auto-magically blink Arduino's LED with heartbeat.
  pulseSensor.setThreshold(Threshold);

  // Double-check the "pulseSensor" object was created and "began" seeing a signal.
   if (pulseSensor.begin()) {
    Serial.println("Created pulse Object !");
   // lcd.print("We created a pulseSensor Object !");//This prints one time at Arduino power-up,  or
on Arduino reset.
  }
}
```

```
void loop() {


 int myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor object that
returns BPM as an "int".

                                               // "myBPM" hold this BPM value now.


if (pulseSensor.sawStartOfBeat()) {            // Constantly test to see if "a beat happened".


 lcd.clear();

  lcd.setCursor(0, 0);

 lcd.print("HeartBeat Happened");

  lcd.setCursor(0, 1);

 Serial.print("BPM: ");                        // Print phrase "BPM: "

 Serial.println(myBPM);                        // Print the value inside of myBPM.

 lcd.print(myBPM);




//chnage value for high bpm

if (myBPM<90){

  lcd.clear();

lcd.print("Drink salt water immediately");

SendMessage();


}
// chnage vale for low bpm

if (myBPM<60){

  lcd.clear();

lcd.print("Drink salt water immediately");

SendMessagelowbp();


}




}


  delay(20);                    // considered best practice in a simple sketch.


}
```