Heartbeat Sensor using Arduino (Heart Rate Monitor)

Heartbeat Sensor is an electronic device that is used to measure the heart rate i.e. speed of the heartbeat. Monitoring body temperature, heart rate and blood pressure are the basic things that we do in order to keep us healthy.

In order to measure the body temperature, we use thermometers and a sphygmomanometer to monitor the Arterial Pressure or Blood Pressure.

Heart Rate can be monitored in two ways: one way is to manually check the pulse either at wrists or neck and the other way is to use a Heartbeat Sensor.

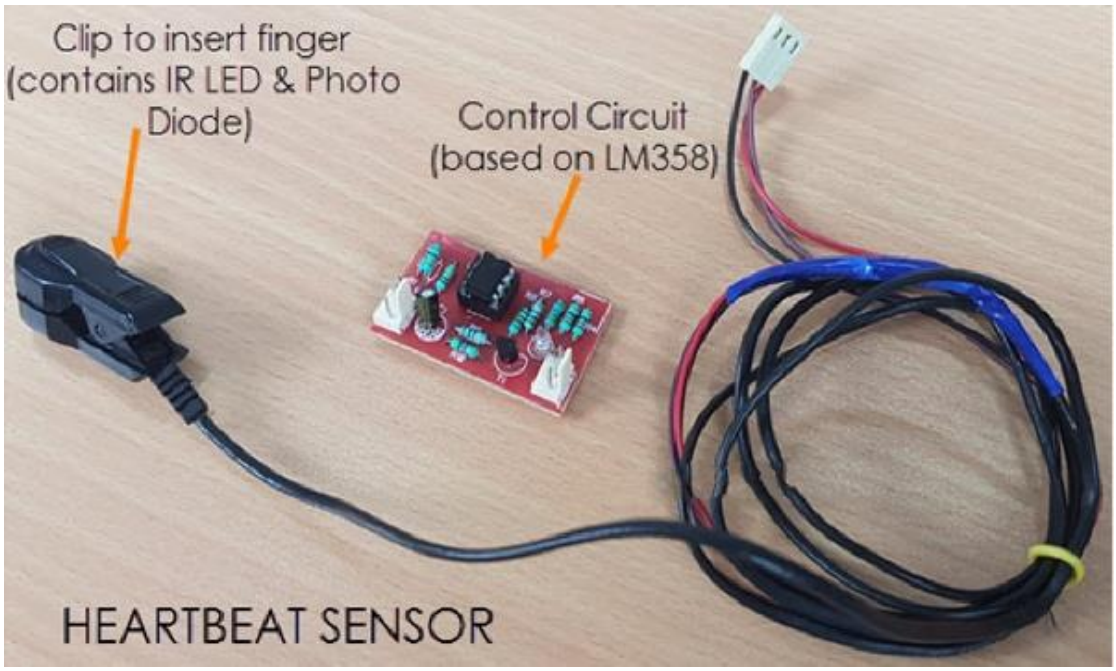
In this project, we have designed a Heart Rate Monitor System using Arduino and Heartbeat Sensor. You can find the Principle of Heartbeat Sensor, working of the Heartbeat Sensor and Arduino based Heart Rate Monitoring System using a practical heartbeat Sensor.

Introduction to Heartbeat Sensor

Monitoring heart rate is very important for athletes, patients as it determines the condition of the heart (just heart rate). There are many ways to measure heart rate and the most precise one is using an Electrocardiography

But the more easy way to monitor the heart rate is to use a Heartbeat Sensor. It comes in different shapes and sizes and allows an instant way to measure the heartbeat.

Heartbeat Sensors are available in Wrist Watches (Smart Watches), Smart Phones, chest straps, etc. The heartbeat is measured in beats per minute or bpm, which indicates the number of times the heart is contracting or expanding in a minute.



Principle of Heartbeat Sensor

The principle behind the working of the Heartbeat Sensor is Photo plethysmograph. According to this principle, the changes in the volume of blood in an organ is measured by the changes in the intensity of the light passing through that organ.

Usually, the source of light in a heartbeat sensor would be an IR LED and the detector would be any Photo Detector like a Photo Diode, an LDR (Light Dependent Resistor) or a Photo Transistor.

With these two i.e. a light source and a detector, we can arrange them in two ways: A Trans missive Sensor and a Reflective Sensor.

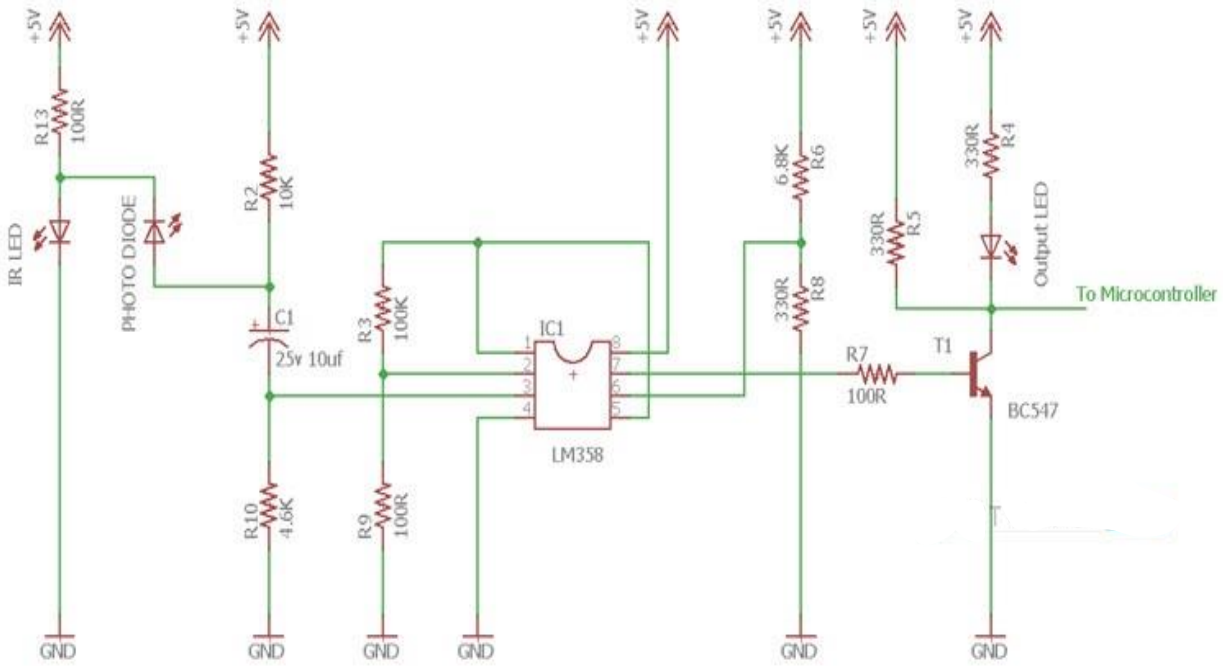
In a Trans missive Sensor, the light source and the detector are place facing each other and the finger of the person must be placed in between the transmitter and receiver.

Reflective Sensor, on the other hand, has the light source and the detector adjacent to each other and the finger of the person must be placed in front of the sensor.

Working of Heartbeat Sensor

A simple Heartbeat Sensor consists of a sensor and a control circuit. The sensor part of the Heartbeat Sensor consists of an IR LED and a Photo Diode placed in a clip.

The Control Circuit consists of an Op-Amp IC and few other components that help in connecting the signal to a Microcontroller. The working of the Heartbeat Sensor can be understood better if we take a look at its circuit diagram.



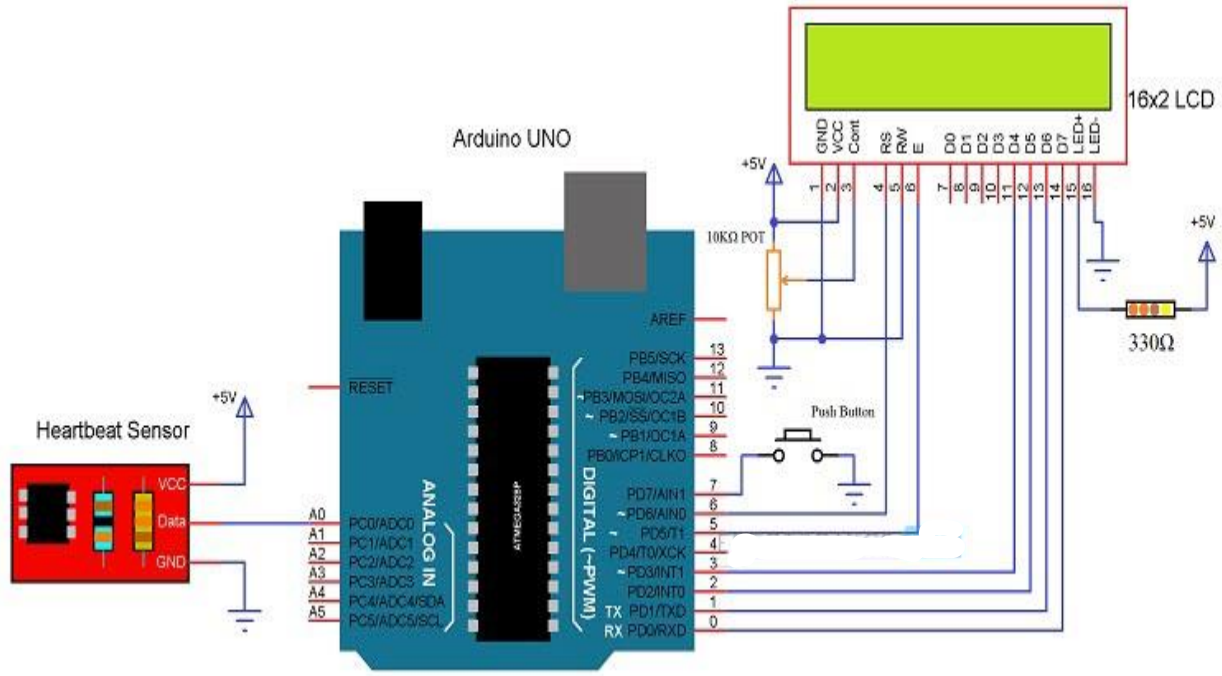
The above circuit shows the finger type heartbeat sensor, which works by detecting the pulses. Every heartbeat will alter the amount of blood in the finger and the light from the IR LED passing through the finger and thus detected by the Photo Diode will also vary.

The output of the photo diode is given to the non – inverting input of the first op – amp through a capacitor, which blocks the DC Components of the signal. The first op – amp cats as a non – inverting amplifier with an amplification factor of 1001.

The output of the first op – amp is given as one of the inputs to the second op – amp, which acts as a comparator. The output of the second op – amp triggers a transistor, from which, the signal is given to a Microcontroller like Arduino.

The Op – amp used in this circuit is LM358. It has two op – amps on the same chip. Also, the transistor used is a BC547. An LED, which is connected to transistor, will blink when the pulse is detected.

Circuit of Arduino based Heart Rate Monitor using Heartbeat Sensor
The following image shows the circuit diagram of the Arduino based Heart Rate Monitor using Heartbeat Sensor. The sensor has a clip to insert the finger and has three pins coming out of it for connecting VCC, GND and the Data.



Components Required

- Arduino UNO x 1
- 16 x 2 LCD Display x 1
- 10KΩ Potentiometer
- 330Ω Resistor (Optional – for LCD backlight)
- Push Button
- Heartbeat Sensor Module with Probe (finger based)

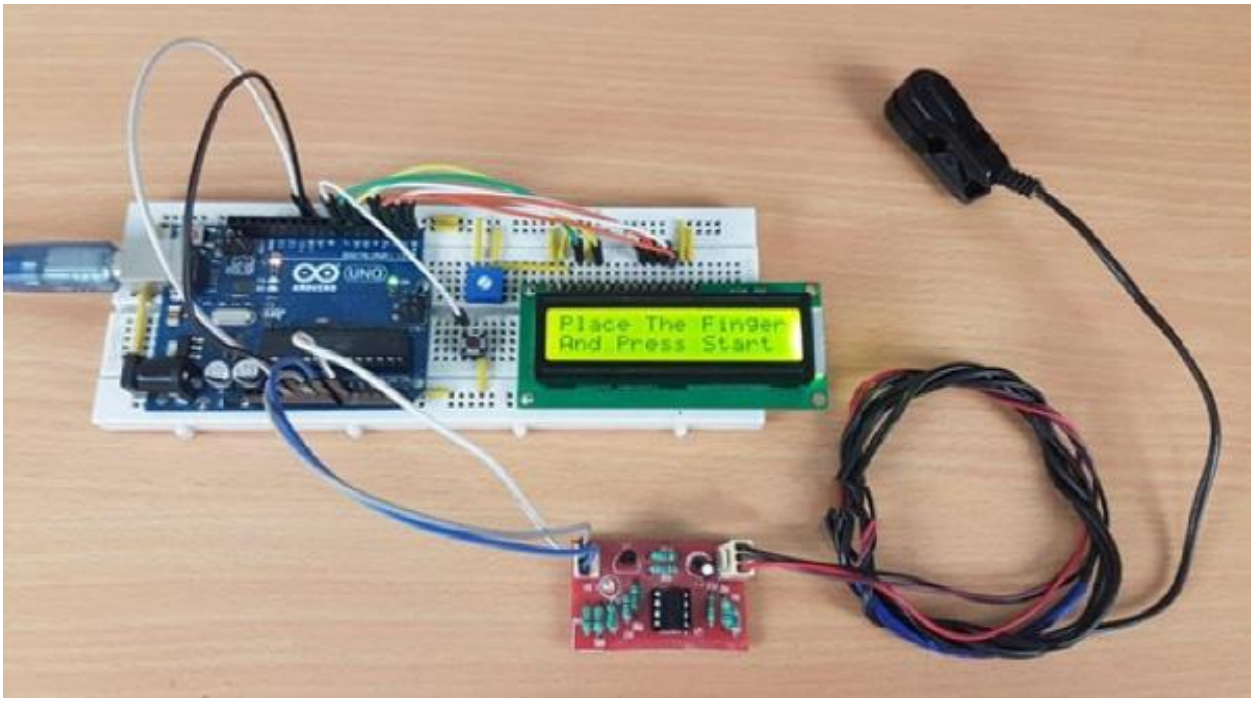
- Mini Breadboard
- Connecting Wires

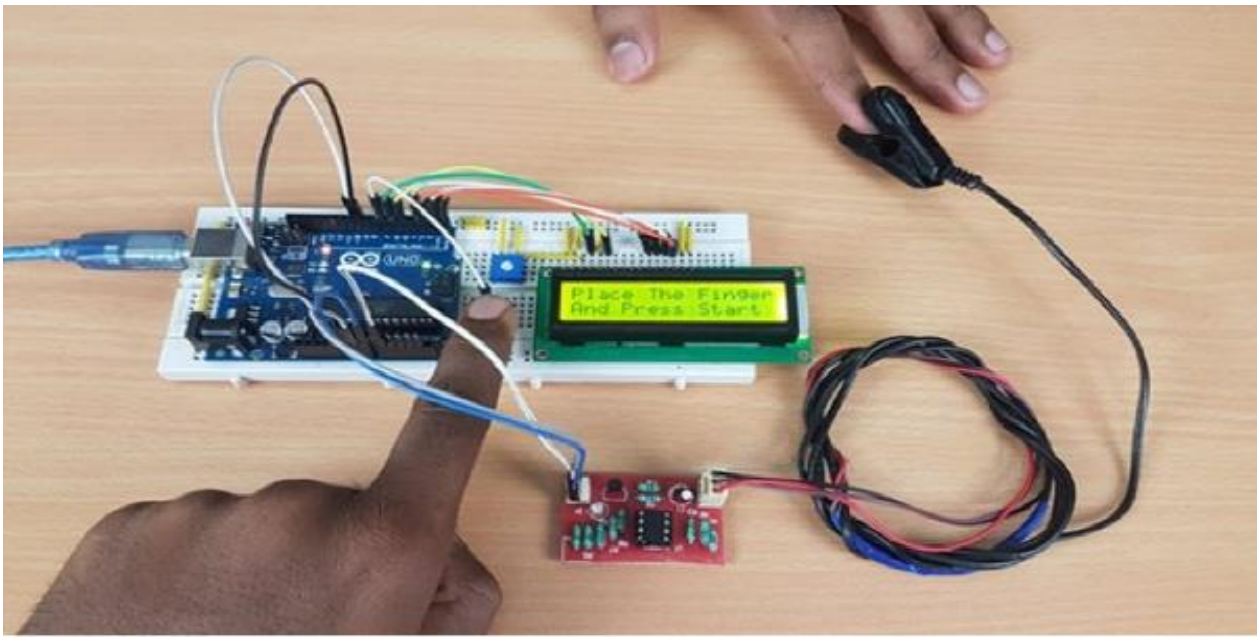
Circuit Design of Interfacing Heartbeat Sensor with Arduino

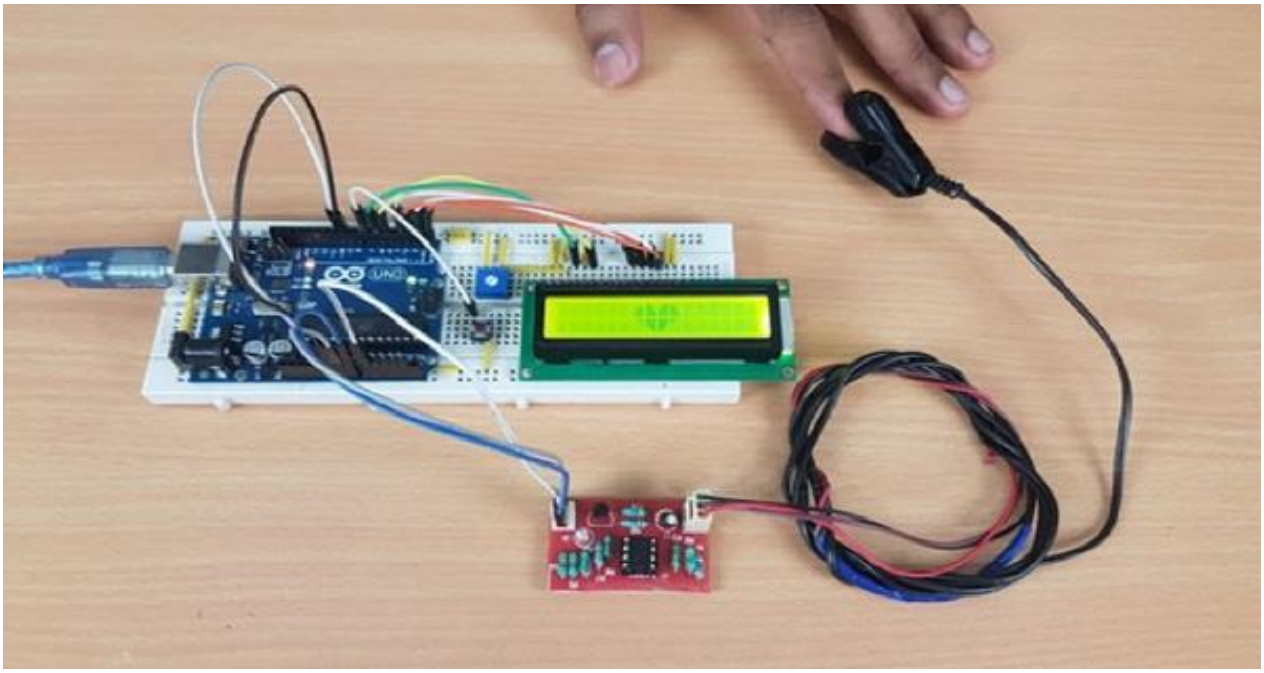
The circuit design of Arduino based Heart rate monitor system using Heart beat Sensor is very simple. First, in order to display the heartbeat readings in bpm, we have to connect a 16×2 LCD Display to the Arduino UNO.

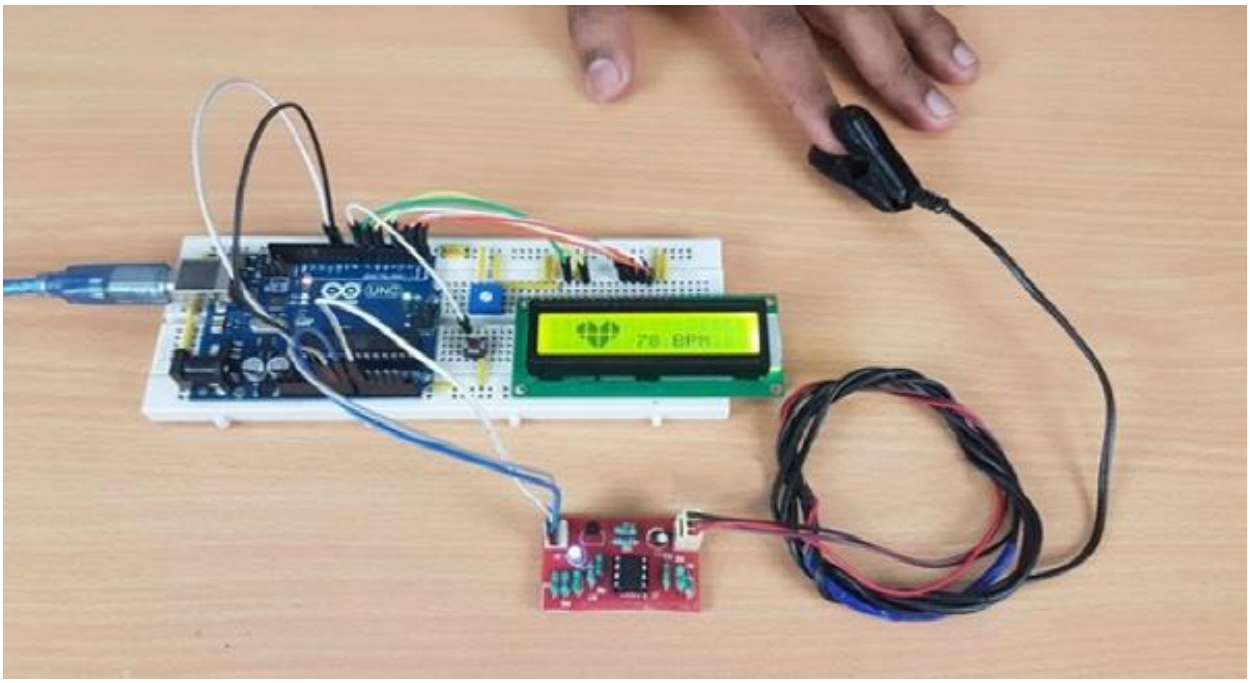
The 4 data pins of the LCD Module (D4, D5, D6 and D7) are connected to Pins 1, 1, 1 and 1 of the Arduino UNO. Also, a 10KΩ Potentiometer is connected to Pin 3 of LCD (contrast adjust pin). The RS and E (Pins 3 and 5) of the LCD are connected to Pins 1 and 1 of the Arduino UNO.

Next, connect the output of the Heartbeat Sensor Module to the Analog Input Pin (Pin 1) of Arduino.




Place The Finger
And Press Start





Working of the Circuit

Upload the code to Arduino UNO and Power on the system. The Arduino asks us to place our finger in the sensor and press the switch.

Place any finger (except the Thumb) in the sensor clip and push the switch (button). Based on the data from the sensor, Arduino calculates the heart rate and displays the heartbeat in bpm.

While the sensor is collecting the data, sit down and relax and do not shake the wire as it might result in a faulty values.

After the result is displayed on the LCD, if you want to perform another test, just push the rest button on the Arduino and start the procedure once again.

CODE

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(6, 5, 3, 2, 1, 0);
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);
}
```

Applications of Heart Rate Monitor using Arduino

- A simple project involving Arduino UNO, 16×2 LCD and Heartbeat Sensor Module is designed here which can calculate the heart rate of a person.
- This project can be used as an inexpensive alternative to Smart Watches and other expensive Heart Rate Monitors.