Heart Rate Monitoring using Fog Computation

In this project we are going to make a **Heart Rate Detection and Monitoring System using Arduino** that will detect the heart beat using the Pulse Sensor and will show the readings in BPM (Beats Per Minute) on the LCD connected to it. It will also send the readings to ThingSpeak server using the Wi-Fi module ESP8266, so that Heart Beats can be monitored from anywhere in the world over the internet. **ThingSpeak** is a great source for displaying the data online and you can access the data from ThingSpeak at any time and at any place.

Along with this, we are using Fog computing, which moves computation and networking closer to the network edge, reduces the need to communicate via the cloud and thus decreases latency.

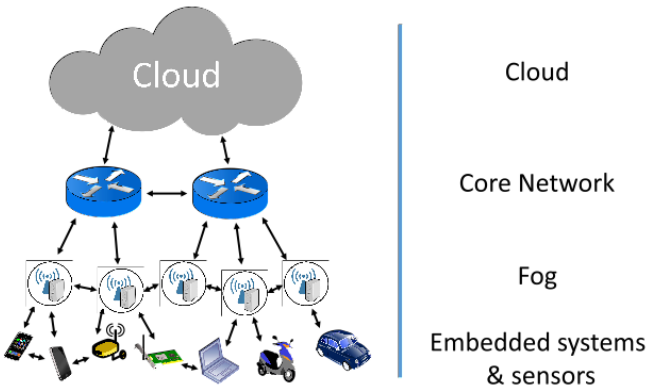


Figure 1. Generic Distributed Fog Computing Infrastructure.

Components Required:

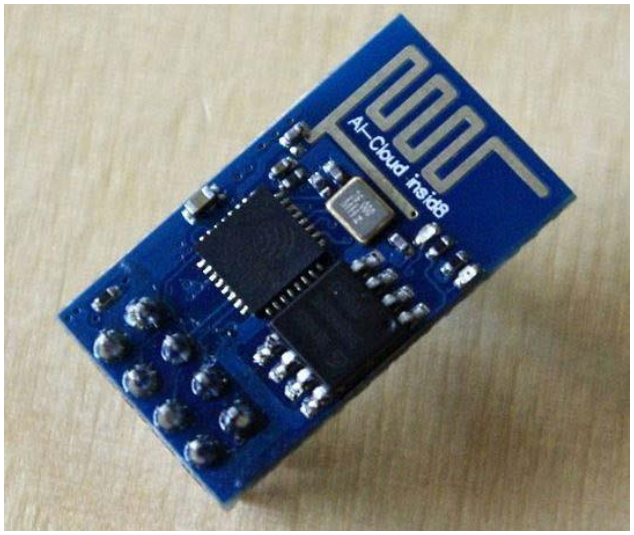
- Pulse sensor
- Wi-Fi module ESP8266
- Arduino Uno
- LCD
- Bread Board
- 10k potentiometer
- 1k resistors
- 220 ohm resistors
- LED
- Connecting wires

Circuit Diagram and Explanation:

First of all we will connect the ESP8266 with the Arduino. ESP8266 runs on 3.3V and if you will give it 5V from the Arduino then it won't work properly and it may get damage. Connect the VCC and the CH_PD to the 3.3V pin of Arduino. The RX pin of ESP8266 works on 3.3V and it will not communicate with the Arduino when we will connect it directly to the Arduino. So, we will have to make a voltage divider for it which will convert the 5V into 3.3V. This can be done by connecting three resistors in series like we did in the circuit. Connect the TX pin of the ESP8266 to the pin 9 of the Arduino and the RX pin of the ESP8266 to the pin 10 of Arduino through the resistors.

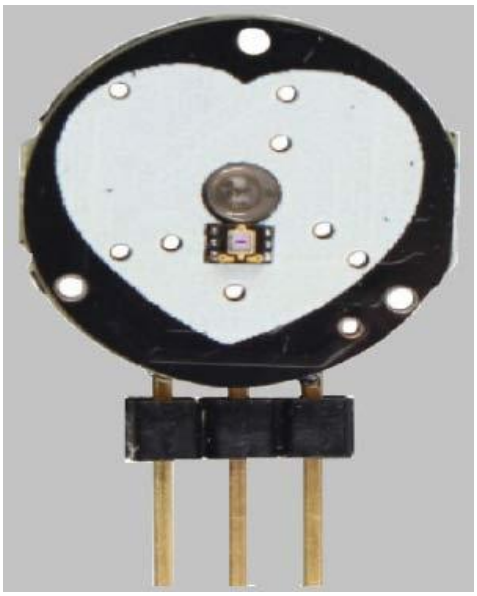
ESP8266 Wi-Fi module gives your projects access to Wi-Fi or internet. It can communicate with any microcontroller and it is the most leading devices in IOT platform

.



Then connect the Pulse Sensor with the Arduino. The connections of the pulse sensor are very easy. Pulse sensor has three pins. Connect 5V and the ground pin of the pulse sensor to the 5V and the ground of the Arduino and the signal pin to the A0 of Arduino.

Then connect the LED to pin 13 of Arduino. You do not have to connect a resistor with because the Arduino has built in resistor at pin 13.



In last, we will connect the LCD with Arduino. The connections of the LCD are as follows

- Connect pin 1 (VEE) to the ground.
- Connect pin 2 (VDD or VCC) to the 5V.
- Connect pin 3 (V0) to the middle pin of the 10K potentiometer and connect the other two ends of the potentiometer to the VCC and the GND. The potentiometer is used to control the screen contrast of the LCD. Potentiometer of values other than 10K will work too.
- Connect pin 4 (RS) to the pin 12 of the Arduino.
- Connect pin 5 (Read/Write) to the ground of Arduino. This pin is not often used so we will connect it to the ground.
- Connect pin 6 (E) to the pin 11 of the Arduino. The RS and E pin are the control pins which are used to send data and characters.
- The following four pins are data pins which are used to communicate with the Arduino.

Connect pin 11 (D4) to pin 5 of Arduino.

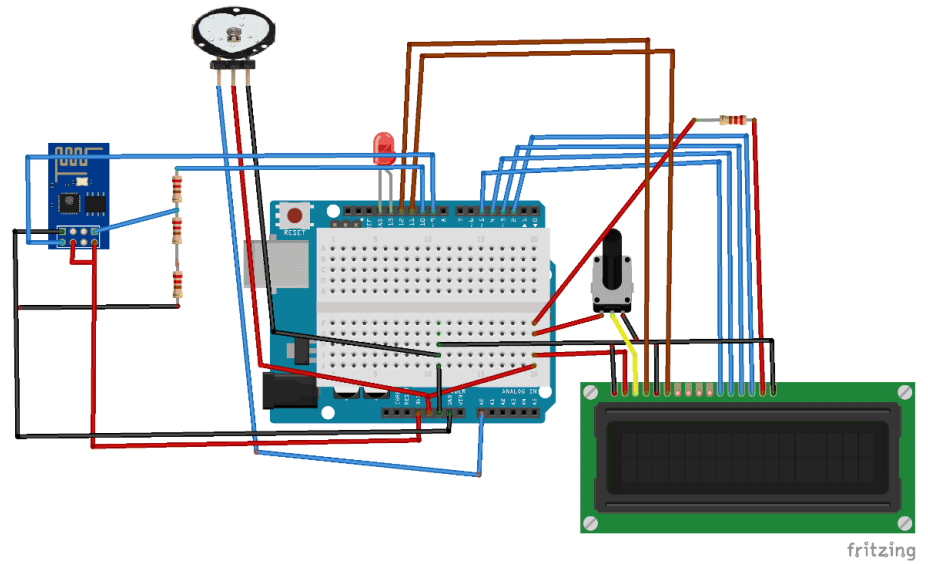
Connect pin 12 (D5) to pin 4 of Arduino.

Connect pin 13 (D6) to pin 3 of Arduino.

Connect pin 14 (D7) to pin 2 of Arduino.

- Connect pin 15 to the VCC through the 220 ohm resistor. The resistor will be used to set the back light brightness. Larger values will make the back light much more darker.
- Connect pin 16 to the Ground.

- After this we will do the ThingSpeak setup and take the reading and make the graphs.
- We will setup a parameter for Trigger and then accordingly we can implement Fog for emergency situations.



Working Explanation:

First we need to attach the Pulse Sensor to any organ of body where it can detect the pulse easily like finger, check the video below. Then the Pulse Sensor will measure the change in volume of blood, which occurs when every time heart pumps blood in the body. This change in volume of blood causes a change in the light intensity through that organ. The Arduino will then convert this change into the heart beat per minute (BPM). The LED connected at pin 13 will also blink according the Heart Beat.

The ESP8266 will then communicate with the Arduino and will send the data to ThingSpeak. The ESP8266 will connect the network of your router that you will provide in the code and will send the data of the sensor online. This data on the ThingSpeak will be shown in a Graph form showing the past readings too and can be accessed from anywhere over internet. The LCD connected will also show you the BPM.