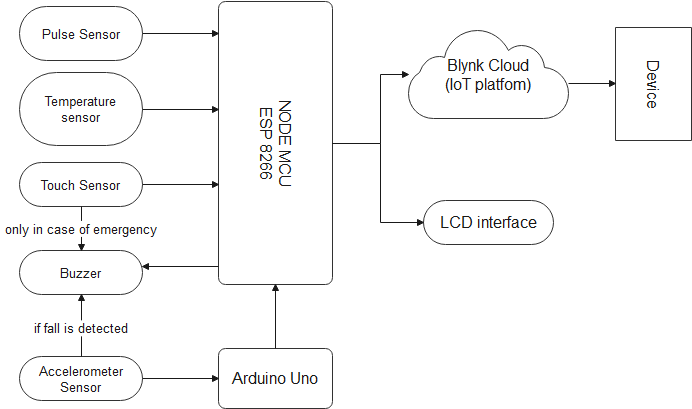
**Health Monitoring System**

**Component used: -**

|  |  |  |
| --- | --- | --- |
| S.N. | Name of component | Quantity |
| 1 | Node Mcu (ESP8266) | 1 |
| 2 | Arduino uno board | 1 |
| 3 | Pulse sensor 1 | 1 |
| 4 | Touch sensor (proximity sensor) | 3 |
| 5 | Buzzer | 2 |
| 6 | Accelerometer (ADXL 345) | 1 |
| 7 | LCD display (16 X 2) | 1 |

**Connections: -**



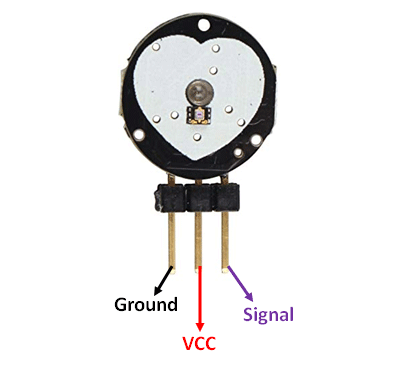
Pulse sensor: - Connect to the ESP8266 on pin A0 as it is analog pin.

Temperature sensor: - We used DHT11 sensor for temperature measurement. It is connected to ESP8266 to digital pin 2.

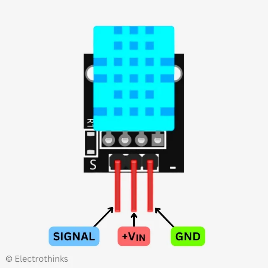
Touch sensor: - We used 3 proximity sensors at pins 12, 13, 14.

Buzzer: - We used two buzzers one for emergency purpose and other for fall detection. Emergency purpose buzzer connected to ESP8266 at digital pin 16 and fall detection buzzer is connected to Arduino uno board at pin 7.

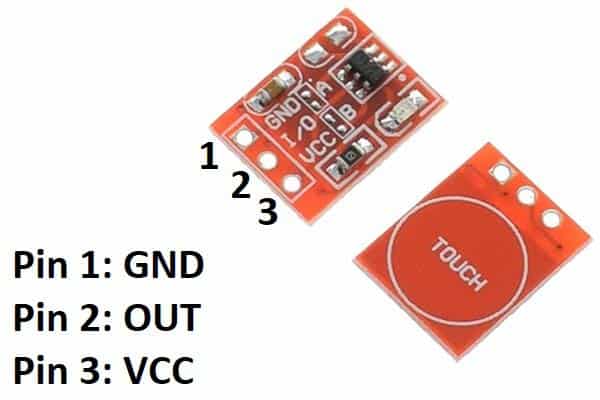
Accelerometer: - We used ADXL345 digital accelerometer connected to Arduino uno board at pins SCL and SDA.



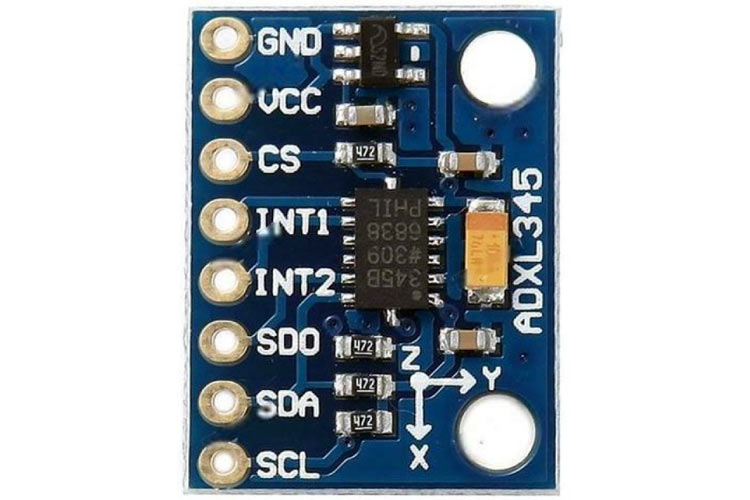
Pulse sensor



DHT11 sensor



Touch sensor



ADXL345 sensor

After connecting the circuit as per the instruction above, finally connect Arduino uno board and ESP8266. In which, ESP8266 acts as a master microcontroller and Arduino acts as a slave microcontroller. Connect LCD display to see the monitored data, which will be connected to ESP8266 by SCL and SDA pins.

To make the monitored data available on the remote device we will make use of Blynk IoT platform. To start with the Blynk IoT platform follow the following steps: -

1. Open the Blynk IoT platform on the web and login with E-mail ID.
2. Create a dashboard to display temperature and heartrate continuously.
3. Manage notifications as the proximity sensor is activated and display it with respective messages.
4. Use the Auth token and Token ID of Blynk dashboard in ESP8266 code.

**Now you are ready with your project!!**