Project Report on

PET HEALTH MONITORING SYSTEM

Submission to THE ROBOTICS CLUB - SNIST as a part of INDUCTION'23

TEAM NO - 04

THE ROBOTICS CLUB

Integrating Knowledge...

THE ROBOTICS CLUB-SNIST

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY

(AUTONOMOUS)

(Affiliated to JNTU University, Hyderabad)
Yamnampet, Ghatkesar, Hyderabad – 501301.

2023

CERTIFICATE

This is the project work titled "**Pet's Health Monitoring System"** by '**Jerry K Paul, P. Ajay, Veluri Chandra Shekhar Pradeep, T. Hima Avinash, Tanmay Sugandhi, R.N. Rohit Chand, P. Bhagya Sree, Ashritha Ravula, Manichand'** under the mentorship of '**SaiGanesh, Priya**' and is a record of the project work carried out by them during the year 2023-2024 as part of INDUCTION'23 under the guidance and supervision of

Mr. D P NAGA AJAY KUMAR
&
MS. TAPASWINI
Technical head

Mr. T BHARAT KUMAR
The President of
THE ROBOTICS CLUB

Dr. A. PURUSHOTHAM
Faculty Advisor
Mechanical Department

DECLARATION

The project work reported in the present thesis titled **"Pet's Health Monitoring System"** is a record work done by **Team 04** in **THE ROBOTICS CLUB** as a part of **INDUCTION-2023.**

No part of the thesis is copied from books/ journals/ Internet and wherever the portion is taken, the same has been duly referred in the text. The report is based on the project work done entirely by TEAM 04 and not copied from any other source.

ACKNOWLEDGMENT

This project report is the outcome of the efforts of many people who have driven our passion to explore into implementation of **"Pet's Health Monitoring System".** We have received great guidance, encouragement and support from them and have learned a lot because of their willingness to share their knowledge and experience.

We thank our technical heads **Mr. D P NAGA AJAY KUMAR & Ms. P TAPASWINI"** fo r being with us till the end of the project completion.

We thank all the members of **Steering Body**, **Executive Body, Technical Advisory Board, Club's Incubation and Competence Committee** of **The Robotics Club** for helping us with crucial parts of the project. We are deeply indebted to **Mr. T Bharat Kumar –** The President, **Mr. N Abhinav-** The Vice President and **Mr. Jayanth Siva Madhav** – SAB Chairman and **Mr. S V REDDY** – General Secretary **THE ROBOTICS CLUB** respectively and also every other person who spared their valuable time without any hesitation whenever we wanted.

We also thank our faculty advisor **Dr. A. PURUSHOTHAM**, Professor, Mechanical Department, who encouraged us during this project by rendering his help when needed.

Contents **Pg. No**

Chapter1	Introduction:	
1.1	Problem Statement	1
1.2	Introduction of the project	1
1.3	Literature survey	1
1.4	Organization of project	2

Chapter2	Architecture:	
2.1	List of figures	2
2.2	Components used	2
2.3	Hardware	2
2.4	Software	4

Chapter3	Implementation and working:	
3.1	Block diagram	5
3.2	Flowchart	5
3.3	Working	5
3.4	Circuit diagram	6

Chapter4	Experimental Results and Conclusions:	
4.1	Results	6
4.2	Future enhancements	6
4.3	Conclusions	6
4.4	References	6
4.5	Source code	6-8
4.6	List of expenses	9

ABSTRACT
THE ROBOTICS CLUN-SNIST
INDUCTION'23
TEAM-04

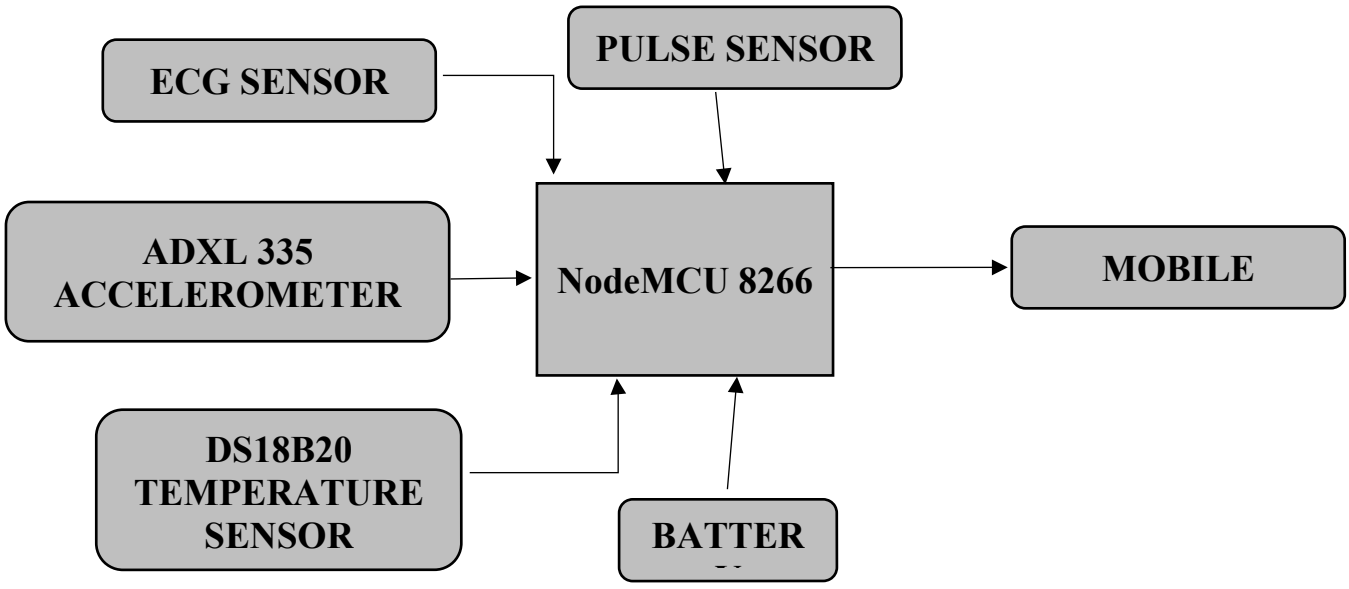
THE PROBLEM:

Every living creature on this earth has equal importance within the ecosystem. But at present, the lives of the animals are in risk**.** There is an increasing number of issues regarding various animal health condition and movements. And in recent era, pets have become an integral part of a human life. For many of us, pets are real family members, and we take care of them, their health, and wellbeing. At present to detect the health status of animal we required to wait for veterinary expertise which take long time for its arrival.

TEAM'S APPROACH TO SOLVE THE PROBLEM:

Pet's health monitoring system is a belt which can be used to check all the vital parameters of their pets in real-time. Such belts can measure heart rate, temperature etc. By using ECG sensor, we can measure heart rate and with temperature sensor we can measure the pet's body temperature. If any serious changes occur in any parameters that can be taken as signs of illness, it will send an alert to a pet owner. Physical activity is very important for pets. Modern applications can analyse walking distance, time of activity per day etc. We can use ADXL accelerometer to count the steps during walking. If a pet is not active enough during the day, an owner will be notified. To control all the sensors, we use NodeMCU8266 microcontroller.

BLOCK DIAGRAM:



TITLE OF THE PROJECT: - PET'S HEALTH MONITORING SYSTEM

What inspired you to select the problem?

These days technology is approaching to greater heights, many people are using smart watches for monitoring of health and compact version of the mobile. We thought to monitor the pets help of IoT technology. using few sensors, the owners of the pets can monitor the health condition of their pets.

What do you feel is the most innovative part of the problem?

I think the most innovative part of this problem is that inserting all the electronics stuff on the body of dog is a great task. Also, transferring the data to the mobile for user understanding.

PET's HEALTH MONITORING SYSTEM

1. Jerry K Paul, 2. P. Ajay, 3. Veluri Chandra Shekhar Pradeep, 4. T. Hima Avinash, 5. Tanmay Sugandhi, 6. R.N. Rohit Chand, 7. P. Bhagya Sree, 8. Ashritha Ravula, 9. Manichand
Sreenidhi Institute of Science and Technology, Yamnampet, Ghatkesar

Abstract— Every living creature on this earth has equal importance within the ecosystem. But at present, the lives of the animals are in risk**.** There is an increasing number of issues regarding various animal health condition and movements. And in recent era, pets have become an integral part of a human life. For many of us, pets are real family members, and we take care of them, their health, and wellbeing. At present to detect the health status of animal we required to wait for veterinary expertise which take long time for its arrival. Pet's health monitoring system is a belt which can be used to check all the vital parameters of their pets in real-time. Such belts can measure heart rate, temperature etc. By using ECG sensor, we can measure heart rate and with temperature sensor we can measure the pet's body temperature. If any serious changes occur in any parameters that can be taken as signs of illness, it will send an alert to a pet owner. Physical activity is very important for pets. Modern applications can analyse walking distance, time of activity per day etc. We can use ADXL accelerometer to count the steps during walking. If a pet is not active enough during the day, an owner will be notified. To control all the sensors, we use NodeMCU8266 microcontroller.

I. INTRODUCTION

As the healthcare industry continues to embrace the digital era and advance remote patient monitoring, it's no surprise that this technology is also transforming the pet health industry. With everything from wearable devices to smart health monitoring platforms, owners can now use a variety of wearable and non-wearable pet accessories to track almost every area of their pet's health. the wireless pet wearables industry has expanded from location trackers to include solutions powered by IoT technologies that help owners monitor their pet's health. PET's HEALTH MONITORING SYSTEM is a wearable always-on health tracker that monitors a pet's health and activity levels to generate a wellness score, providing the owner with a big picture of the animal's overall health. While some wearable trackers are wireless tags attached to a normal pet collar, the Pet Pace Collar is an actual collar. Designed for both cats and dogs, it continuously monitors and measures their vital signs, including activity, pulse, temperature, and number of steps. It provides a convenient and accurate way to monitor pets with existing medical issues while aiding with preventative care. Owners and veterinarians can get comprehensive, accurate, and immediate information about an animal's health via the website and smartphone app.

II. PROBLEM STATEMENT

Pet's health monitoring system is a belt which can be used to check all the vital parameters of their pets in real-time. Such belts can measure heart rate, temperature etc. By using ECG sensor, we can measure heart rate and with temperature sensor we can measure the pet's body temperature. If any serious changes occur in any parameters that can be taken as signs of illness, it will send an alert to a pet owner. Physical activity is very important for pets. Modern applications can analyse walking distance, time of activity per day etc. We can use ADXL accelerometer to count the steps during walking. If a pet is not active

enough during the day, an owner will be notified. To control all the sensors, we use NodeMCU8266 microcontroller.

III. LITERATURE SURVEY

We all discussed about the problem statement and put out our own ideas and changes to bring this small-scale project which can be made fast. We saw multiple videos and had many discussions on how to implement our idea in an efficient and what are the measures we need to take while doing the project.

IV. ARCHITECTURE

COMPONENT REQUIRED: -

Hardware:

1. **ECG Sensor:** The EPIC sensor can be used either in a contact or non-contact mode. The contact mode is used to measure for example bio-electric signals such as an ECG or an EEG from the human body, and the non-contact mode is used to measure the disruption of an electric field caused by a human body movement. An electrical heart rate sensor, also called an ECG, measures your heart rate by measuring electrical signals in your blood. Based on these electrical signals, an ECG heart rate monitor measures the timing and strength of your heart rate. An ECG heart rate monitor is more accurate than an optical heart rate. The basic principle of the ECG is that stimulation of a muscle alters the electrical potential of the muscle fibres.

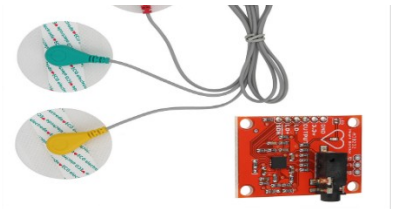


Fig. 4.1. ECG Sensor

2. **DS18B20 Temperature Sensor:** The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with non-volatile user-programmable upper and lower trigger points 8 The main advantage of DS18B20 Temperature sensor is its size of the sensor, It is so small and compact that it can be used almost everywhere. Secondly, the metal covering which protects it from environmental factors like heat direct heat, and water also acts as a conducting part for the overall temperature sensor.

Fig. 4.2. DS18B20 Temperature Sensor

3. **ADXL335 Sensor:** The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 g. The principle that when the acceleration is applied to the sensor, the capacitance inside the sensor changes. This change in capacitance is then used to measure the acceleration of the object. Accelerometer sensors have the ability to alter obtained physical acceleration from motion.

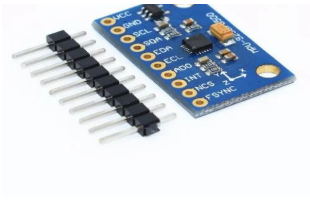


Fig. 4.3. ADXL335 Sensor

4. **Pulse sensor:** A pulse wave is the change in the volume of a blood vessel that occurs when the heart pumps blood, and a detector that monitors this volume

change is called a pulse sensor. The PPG uses 7 a pulse sensor on the fingertip that measures the changes in light absorption and reflection onto the skin to measure blood flow. Heart beat sensors are designed to give digital output heart beat when a finger is placed on it. When the heart beat detector starts working, the light emitting detector blinks simultaneously for every heartbeat.

Fig. 4.4. Pulse sensor

5. **NodeMCU:** NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Esp Systems, and hardware which is based on the ESP-12 module. The NodeMCU Development Board can be easily programmed with Arduino IDE since it is easy to use Programming NodeMCU with the Arduino IDE will hardly take 5-10 minutes. All you need is the Arduino IDE, a USB cable and the NodeMCU board itself. You can check this Getting Started Tutorial for NodeMCU to prepare your Arduino IDE for Node MCU.

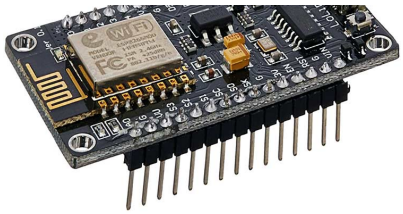


Fig. 4.5. NodeMCU

6. **Battery**: Lithium-ion batteries are smaller and more powerful than other batteries. Lithium-ion batteries stand up well to repeated charging and discharging. Lithium-ion batteries can be fast-charging. Lithium-ion batteries support wireless charging.

Fig. 4.6. LI-ION Batteries

7. **Jumper wires:** A jumper wire is an electrical wire, or group of them in a cable, with a connector or pin at each end which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Fig. 4.7. Jumper Wires

8. **Dog Chest Belt:** Screw blades commonly known as screw conveyor blades, these are single segments which, when connected together, form spirals or shaftless screws. The production process starts with specialized software, which defines the parameters of laser firing cutting and blade bending operation. The CNC machine ensures precision and high quality of the cold bending process.

Fig. 4.8. Dog Chest Belt

9. **Vector Board:** General purpose pre-punched insulating punch board with variety of hole sizes and grid patterns; unplanted copper clad or unclad.

Fig. 4.9. Vector board

V. SOFTWARE RERQUIREMENTS

a. **Arduino IDE**: Arduino is an open-source hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices.

Fig. 5.1. Arduino IDE

b. **Proteus Software:** The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

Fig. 5.2. Proteus Software

c. **Auto CAD:** CAD (computer aided design) design is used in almost every industry, in projects as wide-ranging as landscape design, bridge construction, office building design, and movie animation. With 2D or 3D CAD programs, you can perform a variety of tasks: you can create a 3D model of a design, apply material and light effects, and document the design with dimensions and other annotations. With features like point clouds, you can add real-life context to your drawings to create a digital twin or recreate physical objects in your designs.

Fig. 5.3. AutoCAD

d. **Thinker CAD:** Tinker cad is a free web app for 3D design, electronics,

and coding, trusted by over 50 million people around the world. Build STEM confidence by bringing project-based learning to the classroom.

Fig. 5.4. Thinker CAD

e. **Blynk**: Blynk is a software company that provides infrastructure for the Internet of Things. In 2014 Blynk pioneered the no-code approach to IoT app building and gained global popularity for its mobile app editor.

Fig. 5.5. Blynk

VI. IMPLEMENTATION AND WORKING

i. Initially the belt which is made through the idea is attached to the pets.
ii. Then if the supply is connected the product start working.
iii. The sensors which are attached will senses the body of the pets a show them in the mobile application which is developed.
iv. So that the user monitors the health of the pets.
v. If the readings are less than the threshold then the owners can make necessary steps.

VII. CAD MODEL

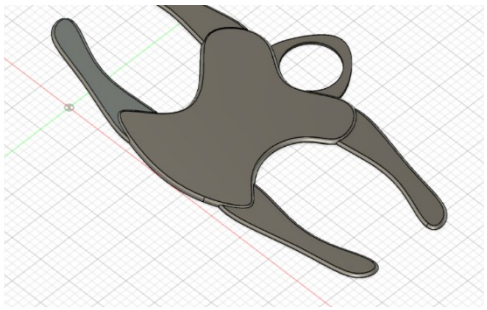


Fig. 7.1. CAD Design

VIII. BLOCK DIAGRAM

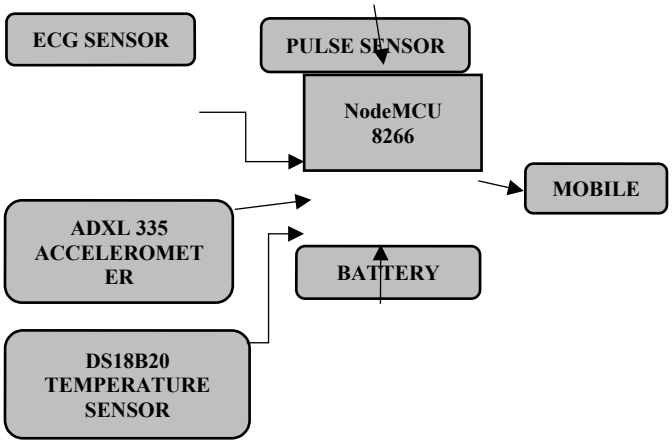


Fig. 8.1. Block Diagram

IX. CIRCUIT DIAGRAM

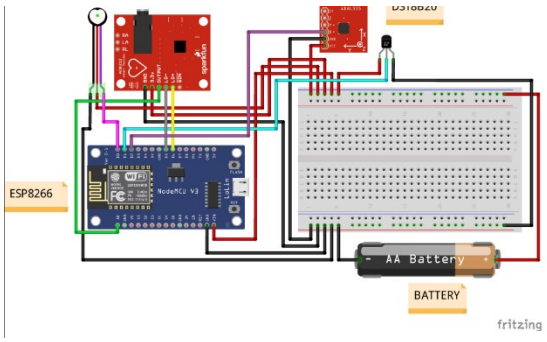


Fig. 9.1. Circuit Diagram

X. RESULT

From the project we can conclude that the project monitors the health of the pets like dog and cats by recording the values such as temperature, ECG, pulse and counting the number of steps. These readings can be monitored by owner through mobile application.

XI. FUTURE ADVANCEMENTS

i. Can make a Mobile Application
ii. Can place a food remainder.
iii. Task remainder.
iv. Number of calories burnt.
v. Placing GPS tracker and GSM to track the location.

XII. REFERENCES

Smart Devices for Pet Health Monitoring. (2022, November 28). Ambiq. https://ambiq.com/blog/smart-devices-for-pet-health-monitoring/

Kumari, S. (2018). *Development of IoT Based Smart Animal Health Monitoring System Using Raspberry Pi*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3315327#:~:text=A%20prototype%20of%20smart%20animal,and%20humidity%20has%20been%20developed.

Unold O, Nikodem M, Piasecki M, Szyc K, Maciejewski H, Bawiec M, Dobrowolski P, Zdunek M. IoT-Based Cow Health Monitoring System. Computational Science – ICCS 2020. 2020 May 25; 12141:344–56. doi: 10.1007/978-3-030-50426-7_26. PMCID: PMC7302546.

XIII. SOURCE CODE

```
// code for PET'S HEALTH MONITORING SYSTEM

// Declarations
#define USE_ARDUINO_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.
#include <OneWire.h>
#include <DallasTemperature.h>
#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.
int ONE_WIRE_BUS=0;   // Data wire is connected to digital pin 0 on the NodeMCU
OneWire oneWire(ONE_WIRE_BUS);  // Setup a oneWire instance to communicate with any OneWire device
DallasTemperature sensors(&oneWire);  // Pass oneWire reference to DallasTemperature library


const int PulseWire = 1;       // PulseSensor PURPLE WIRE connected to digital PIN 1
const int LED = 4;          // 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.

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

const int xpin = 2; //Data wire is connected to digital pin 2 on the NodeMCU
int ECG_pin=A0;     //Data wire is connected to analog pin 0 on the NodeMCU
const int ypin = 0;
const int zpin = 1;
float threshold = 6;
float xval[100] = {0};
float yval[100] = {0};
float zval[100] = {0};
float xavg, yavg, zavg;
int steps, flag = 0;

void setup()
{
  // Pin declerations
  pinMode(3, INPUT); //ECG Setup for leads off detection LO +
  pinMode(2, INPUT); //ECG Setup for leads off detection LO -
  calibrate();
```

```
  pulseSensor.analogInput(PulseWire);
  pulseSensor.blinkOnPulse(LED);       //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("We created a pulseSensor
Object !");  //This prints one time at Arduino power-up,
or on Arduino reset.
  }
  sensors.begin();    // Start up the library
  Serial.begin(9600);
}

void loop()
{

// temparature sensor
  sensors.requestTemperatures();

   //print the temperature in Celsius
  Serial.print("Temperature: ");
  Serial.print(sensors.getTempCByIndex(0));
  Serial.print((char)176);  //shows degrees character
  Serial.print("C  |  ");

  //print the temperature in Fahrenheit
  Serial.print((sensors.getTempCByIndex(0) * 9.0) / 5.0
+ 32.0);
  Serial.print((char)176);  //shows degrees character
  Serial.println("F");

  delay(500);

// Pulse sensor
if (pulseSensor.sawStartOfBeat()) {            // Constantly
test to see if "a beat happened".
int myBPM = pulseSensor.getBeatsPerMinute();  //
Calls function on our pulseSensor object that returns
BPM as an "int".
                                    // "myBPM" hold this
BPM value now.
 Serial.println("♥  A HeartBeat Happened ! "); // If test
is "true", print a message "a heartbeat happened".
 Serial.print("BPM: ");                        // Print phrase
"BPM: "
 Serial.println(myBPM);                        // Print the value
inside of myBPM.
}

  delay(20);

// ADXL 355
  int acc = 0;
  float totvect[100] = {0};
  float totave[100] = {0};
  float xaccl[100] = {0};
  float yaccl[100] = {0};
  float zaccl[100] = {0};

  for (int a = 0; a < 100; a++)
  {
    xaccl[a] = float(analogRead(xpin) - 345);
    delay(1);
    yaccl[a] = float(analogRead(ypin) - 346);
    delay(1);
    zaccl[a] = float(analogRead(zpin) - 416);
    delay(1);

    totvect[a] = sqrt(((xaccl[a] - xavg) * (xaccl[a] -
xavg)) + ((yaccl[a] - yavg) * (yaccl[a] - yavg)) +
((zval[a] - zavg) * (zval[a] - zavg)));
    totave[a] = (totvect[a] + totvect[a - 1]) / 2 ;

    Serial.println("totave[a]");
    Serial.println(totave[a]);

    delay(100);

    if (totave[a] > threshold && flag == 0)
    {
      steps = steps + 1;
      flag = 1;
    }
    else if (totave[a] > threshold && flag == 1)
    {
      // Don't Count
    }
    if (totave[a] < threshold   && flag == 1)
    {
      flag = 0;
    }
    if (steps < 0) {
      steps = 0;
    }

    Serial.println('\n');
    Serial.print("steps: ");
    Serial.println(steps);
    delay(1000);
  }
  delay(1000);
}

void calibrate()
{

  float sum = 0;
  float sum1 = 0;
  float sum2 = 0;
  for (int i = 0; i < 100; i++) {
    xval[i] = float(analogRead(xpin) - 345);
    sum = xval[i] + sum;
  }

  delay(100);

  xavg = sum / 100.0;
  Serial.println(xavg);
  for (int j = 0; j < 100; j++)
  {
    yval[j] = float(analogRead(ypin) - 346);
    sum1 = yval[j] + sum1;
  }

  yavg = sum1 / 100.0;
  Serial.println(yavg);
  delay(100);
  for (int q = 0; q < 100; q++)
  {
```

```
    zval[q] = float(analogRead(zpin) - 416);
    sum2 = zval[q] + sum2;

  }

  zavg = sum2 / 100.0;
  delay(100);
  Serial.println(zavg);

// ECG sensor

  if((digitalRead(10) == 1)||(digitalRead(11) == 1)){
    Serial.println('!');
  }
  else{
    // send the value of analog input 0:
      Serial.println(analogRead(ECG_pin));
  }
  //Wait for a bit to keep serial data from saturating
  delay(1000);
}
```

RECORD OF EXPENSES:

Component	**Quantity**	**Price**
NodeMCU ESP8266	1	**Rs 260/-**
Pulse sensor	1	**Rs 480/-**
ECG sensor	1	**Rs 800/-**
Accelerometer sensor	1	**Rs.600/-**
Battery holders	1	**Rs 35/-**
Temperature sensor	1	**Rs 400/-**
Batteries	4	**Rs 340/-**
Jumper wires	few	**Rs 50/-**
Vector board	1	**Rs 50/-**
Dog belt	1	**Rs 400/-**
	TOTAL: Rs 3415/-	