RV COLLEGE OF ENGINEERING®

Rashtreeya Sikshana Samithi Trust

BENGALURU- 560059

(Autonomous Institution Affiliated to VTU, Belagavi)

Department of Electronics and Instrumentation Engineering

4th SEMESTER
SELF STUDY REPORT
"Smart Watch"

Submitted by

NAME	USN
Srinivas Prabhu B	1RV17EI056
Ruthvik SJ	1RV17EI044
Vinay Balamurali	1RV17EI063
Aisiri HR	1RV17EI004

Evaluation Table

Sl.No	Name of the Student	Relevance of the topic according to the subjects (5)	Report (15)	Presentation & Question Answer (20)	Demonstration (10)	Total (50)
1.	Srinivas					
2.	Ruthvik					
3.	Vinay					
4.	Aisiri					

Table of Contents

Acknowledgment

Firstly, we would like to thank the department of Electronics and Instrumentation, for giving us an opportunity to hone our skills in the fields of Sensors, Microcontrollers and Verilog programming, in the form of a Self Study project.

We would like to express our gratitude to all those who provided us, the necessary help during the course of this project. We extend our regards to Mr. Prasanna Kumar, Mr. Kendaganna Swamy and Mrs. Rajasree, whose constant guidance and encouragement helped us finish the project.

Abstract

In recent technological innovations in the field of disease prevention and maintenance of patient health have enabled the evolution of fields such as monitoring systems. Heart rate is a very vital health parameter that is directly related to the soundness of the human cardiovascular system.

Heart rate data is collected using direct observation method. Direct observation method using tools calls smart watch. This data is used for further research in heart rate time series. It can be measured either by the ECG waveform or by sensing the pulse - the rhythmic expansion and contraction of an artery as blood is forced through it by the regular contractions of the heart. The pulse can be felt from those areas where the artery is close to the skin.

In this project we have built a smart watch which can monitor and keep track of the heart beat of the user. The pulse is sensed by the artery lying underneath and is displayed continuously on OLED interfaced with the watch.

Introduction

Heart rate (HR) or pulse rate represents the number of times a heart beats each minute. The determination of what a normal heart rate is depends on many factors, such as age, body size, movement, exercise, and heart conditions. A normal heart rate can be between 70 and 100 beats per minute (bpm).
A heart rate monitor is a personal monitoring device that allows one to measure/display heart rate in real time or record the heart rate for later study. It is largely used to gather heart rate data while performing various types of physical exercise.

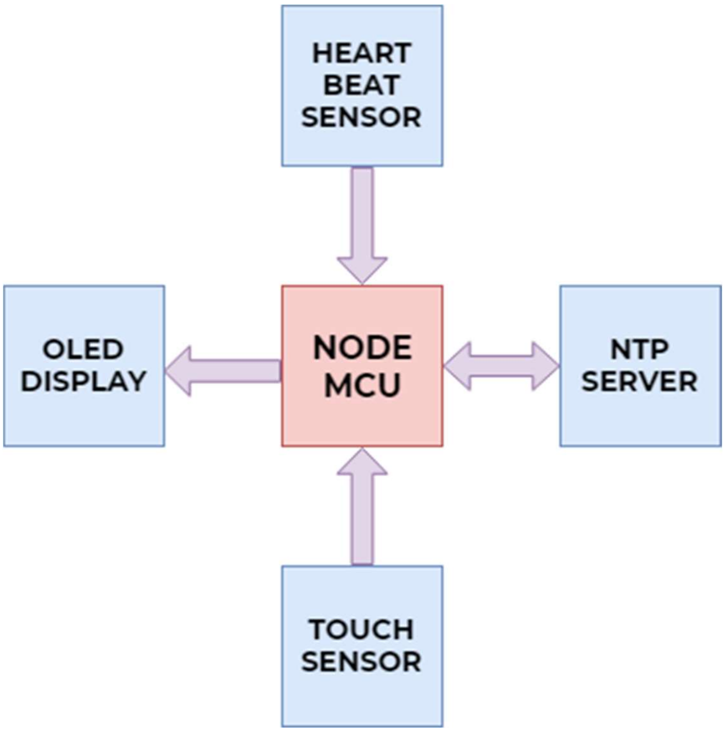
Application

1. Smart watches are being used as a platform for a variety of healthcare applications. Based on our review, the most common healthcare applications using smart watches focused on health monitoring or smart home environment for the elderly.
2. Another major application is with chronically ill patients needing medication adherence monitoring.
3. Monitoring Exercise Intensity.
4. These also prove useful to fitness enthusiasts. Heart rate monitors provide immediate feedback on how hard one is working out so that they can make adjustments to get the greatest benefit from their exercise regimen.
5. Preventing Overtraining.
6. Estimation of Maximal oxygen uptake (VOmax) and EE (Energy-Expenditure) .
7. It is well known that the relation between HR (Heart rate) and VOmax is linear. This relation is used in determining VOmax and Energy expenditure.]

Literature Survey

TITLE OF PAPER	APPLICATION	APPROACH	SENSING METHOD
The use of smart watches to monitor heart rates in elderly people: A Complimentary approach	A knowledge based system to call for help during abnormal heart rates in elderly.	Through rule based reasoning .	No information given
Multi parameter health monitor watch	To build a watch to monitor multiple parameters to tackle chronic non communicable disease.	Continues monitoring of data from sensor.	Dry electrode in front of the watch . Reflective method for acquiring PPG signals.
Smart wrist watch	Building a data base of health parameters of an individual.	Continues monitoring and storage of data from sensors.	BMR is calculated by Harris Benedict equation. TMP75 for temp measurement.
Heart Rate Data Collection Using Smart Watch	Heart rate data collection using a built smart watch for future use in diagnosis	Data collection from already built smart watch.	----

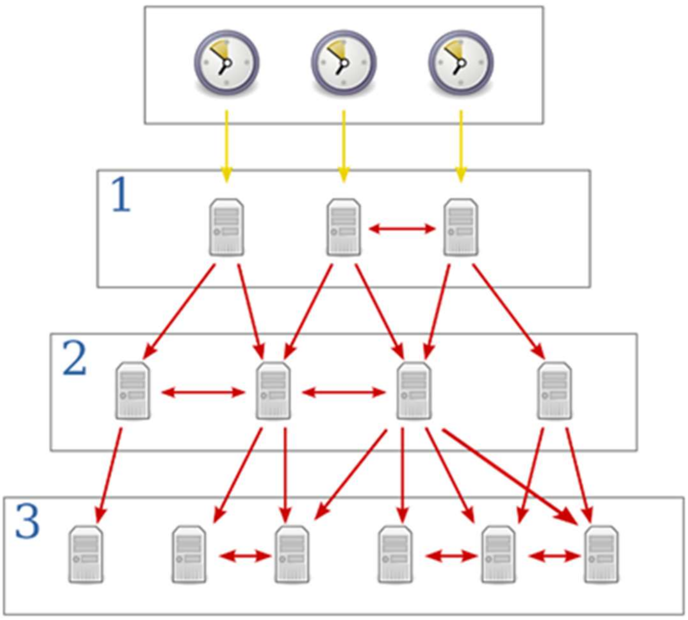
Block-diagram



NTP Server

NTP stands for Network Time Protocol, and it is an Internet protocol used to synchronize the clocks of computers to sometime reference. NTP is an Internet standard protocol originally developed by Professor David L. Mills at the University of Delaware.

NTP uses a hierarchical, semi-layered system of time sources. Each level of this hierarchy is termed a stratum and is assigned a number starting with zero for the reference clock at the top. A server synchronized to a stratum n server runs at stratum n + 1. The number represents the distance from the reference clock and is used to prevent cyclical dependencies in the hierarchy. Stratum is not always an indication of quality or reliability; it is common to find stratum 3 time sources that are higher quality than other stratum 2 time sources. A brief description of strata 0, 1, 2 and 3 is provided below.



Stratum 0

These are high-precision timekeeping devices such as atomic clocks, GPS or other radio clocks. They generate a very accurate pulse per second signal that triggers an interrupt and timestamp on a connected computer. Stratum 0 devices are also known as reference clocks.

Stratum 1

These are computers whose system time is synchronized to within a few microseconds of their attached stratum 0 devices. Stratum 1 servers may peer with other stratum 1 servers for sanity check and backup.[13] They are also referred to as primary time servers.

Stratum 2

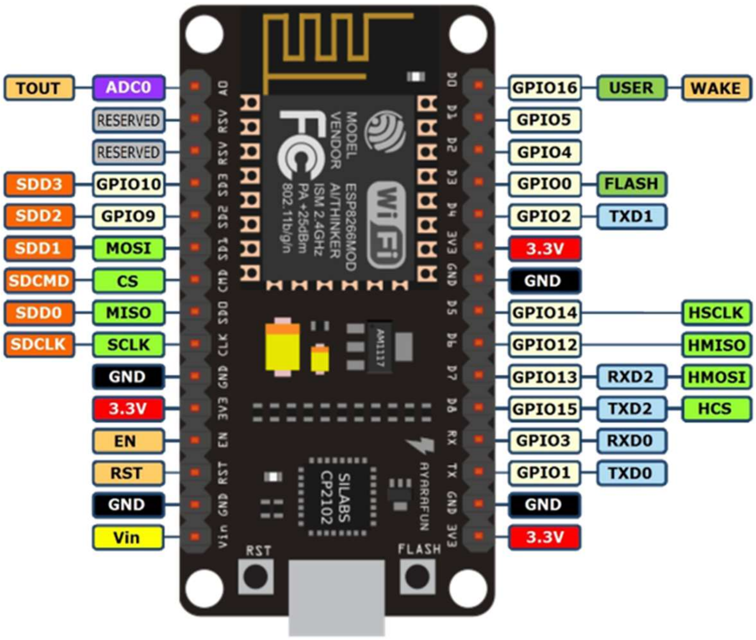
These are computers that are synchronized over a network to stratum 1 servers. Often a stratum 2 computer will query several stratum 1 servers. Stratum 2 computers may also peer with other stratum 2 computers to provide more stable and robust time for all devices in the peer group.

Stratum 3

These are computers that are synchronized to stratum 2 servers. They employ the same algorithms for peering and data sampling as stratum 2, and can themselves act as servers for stratum 4 computers, and so on.

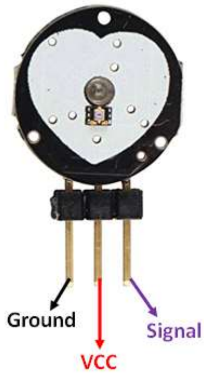
NodeMCU

NodeMCU is an open source Lua based firmware for the ESP8266 WiFi SOC from Espressif and uses an on-module flash-based SPIFFS file system. NodeMCU is implemented in C and is layered on the Espressif NON-OS SDK. The firmware was initially developed as is a companion project to the popular ESP8266-based NodeMCU development modules, but the project is now community-supported, and the firmware can now be run on any ESP.



Heart Beat Sensor

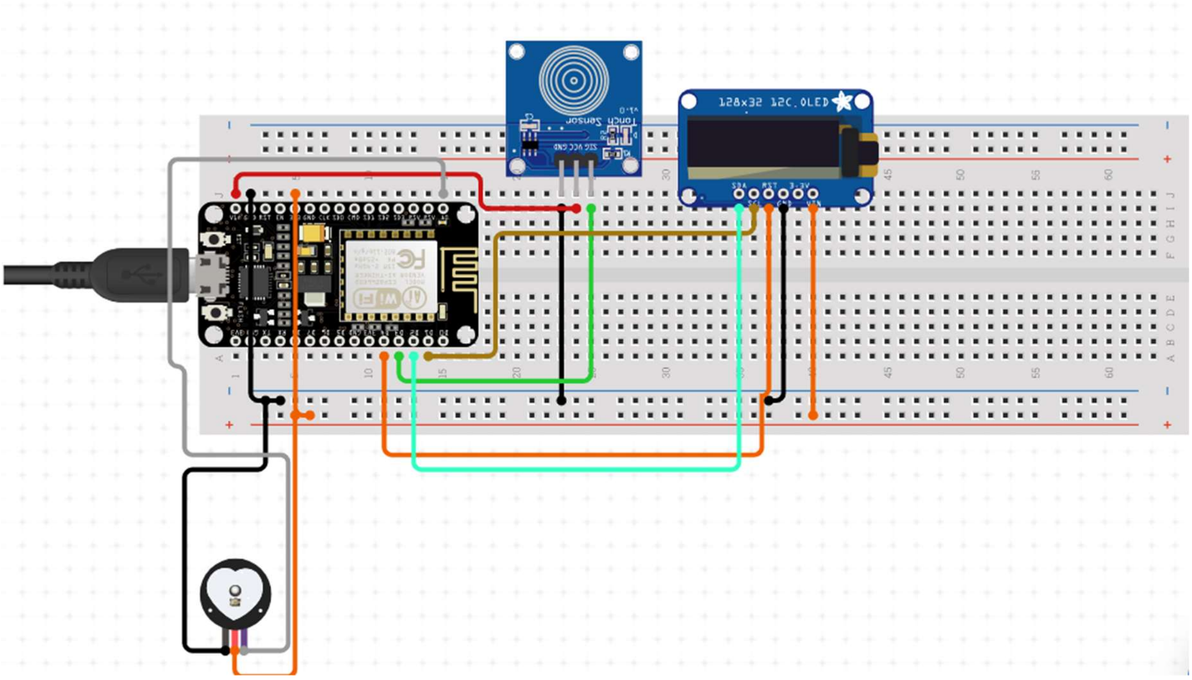
A heart rate monitor (HRM) is a personal monitoring device that allows one to measure/display heart rate in real time or record the heart rate for later study. It is largely used to gather heart rate data while performing various types of physical exercise. Measuring electrical heart information is referred to as Electrocardiography.



OLED

An organic light-emitting diode (OLED) is a light-emitting diode (LED) in which the emissive electroluminescent layer is a film of organic compound that emits light in response to an electric current. This organic layer is situated between two electrodes; typically, at least one of these electrodes is transparent. OLEDs are used to create digital displays in devices such as television screens, computer monitors, portable systems such as smartphones, handheld game consoles and PDAs. A major area of research is the development of white OLED devices for use in solid-state lighting applications

Circuit Diagram



Working

1. NodeMCU which is the firmware for esp8266 gets powered up and in turn powers the touch sensor, oled, heartbeat sensor.
2. ESP8266 connects to the internet through the WiFi credentials given by the user.
3. Once a stable connection is established, ESP8266 pings the NTP server.
4. Using the NTP library, we collect the required date and time, which is displayed on the OLED(the time is updated every second).
5. When the touch sensor is pressed i.e when its value is read high, the OLED screen displays the BPM value which is read from the Heart beat sensor.
6. The BPM is calculated based on the analog values read from the Heart Rate sensor.
7. BPM value is displayed for 10 seconds and then display goes back to the default screen i.e time and date.

Embedded C CODE

```
#include <ESP8266WiFi.h>
#include <time.h>
#include <NTPClient.h>
// change next line to use with another board/shield
#include <ESP8266WiFi.h>
//#include <WiFi.h> // for WiFi shield
//#include <WiFi101.h> // for WiFi 101 shield or MKR1000
#include <WiFiUdp.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET LED_BUILTIN  //4
#define pin_button D8
#if (SSD1306_LCDHEIGHT != 32)
#error("Height incorrect, please fix Adafruit_SSD1306.h!");
#endif
int button_state;
Adafruit_SSD1306 display(OLED_RESET);
int i;
const char* ssid = "virus";
const char* password = "XFD1048576";
String daystamp;
String timestamp;
String formattedDate;
double alpha=0.75;
int ledPin = 13;
int dst = 0;

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "europe.pool.ntp.org", 3600, 60000);

// You can specify the time server pool and the offset (in seconds, can be
// changed later with setTimeOffset() ). Additionaly you can specify the
// update interval (in milliseconds, can be changed using setUpdateInterval() ).
void setup()
{

    pinMode(A0,INPUT);
    pinMode(pin_button, INPUT);
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
```

```
      .
      display.clearDisplay();
      display.display();

      pinMode(ledPin,OUTPUT);
      digitalWrite(ledPin,LOW);

      Serial.begin(115200);

      display.setTextSize(1);
      display.setTextColor(WHITE);

      display.setCursor(0,0);
      display.println("Wifi connecting to ");
      display.println( ssid );

      WiFi.begin(ssid,password);

      display.println("\nConnecting");

display.display();

        while( WiFi.status() != WL_CONNECTED )
            {
            delay(500);
            display.print(".");
            display.display();
            }

timeClient.begin();
timeClient.setTimeOffset(19800);

    // Clear the buffer.
    display.clearDisplay();
    display.display();
    display.setCursor(0,0);

    display.println("Wifi Connected!");
    display.println("Initializing Setup");
    // display.println(WiFi.localIP() );

    display.display();
display.println("\nWaiting for NTP...");//
```

```
while(!time(nullptr))
            {
            Serial.print("*");
            delay(1000);
            }
    display.println("Time response....OK");
    display.println("watch is ready");
    display.display();
    delay(1000);

    display.clearDisplay();
    display.display();
}

void loop()
{
    button();
    timer();
    Serial.println(button_state);
    while(button_state==HIGH)
    {
    for(i=0;i<70;i++)
    {
    BPM();
    }
    button_state = LOW;
    }
}

void BPM()
{
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(0,10);
  display.print("BPM : ");
  display.setTextSize(2);
  display.setCursor(60,10);
  static double oldValue = 0;
  int sensor=analogRead(A0);
  double value = (alpha * oldValue) + ((1 - alpha) * sensor);
  int x=value/7;
  if(x>=65 && x<=90)
```

```
    {
    Serial.print("BPM: ");
    Serial.println(x);
    display.print(x);
    }
    oldValue=value;
    delay(200);
    display.display();
}

int button()
{
     button_state = digitalRead(pin_button);
     // int flag = button_state;
     delay(100);
     return button_state;
}

void timer()
{
    display.clearDisplay();
    timeClient.update();
    timestamp = timeClient.getFormattedTime();
    Serial.println(timeClient.getFormattedTime());
formattedDate = timeClient.getFormattedDate();
    Serial.println(formattedDate);
int splitT=formattedDate.indexOf("T");
    daystamp = formattedDate.substring(0,splitT);
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
display.print(timestamp);
    display.setTextSize(1);
    display.setCursor(0,25);
    display.print(daystamp);
display.display();
    delay(500); // update every 0.5 sec
}
```

VERILOG HDL CODE

```
module timer(clk,rst,a,b,c,d,y);
input clk,rst,a,b,c,d;
reg a,b,c,d;
output reg [3:0] y;
reg [26:0] clkd;
always@(posedge clk) clkd=clkd+1;
always@(posedge clkd[20])
begin

case(a)
0: y[0]=8'b00000011;
1: y[0]=8'b10011111;
2: y[0]=8'b00100101;
3: y[0]=8'b00001101;
4: y[0]=8'b10011001;
5: y[0]=8'b01001001;
6: y[0]=8'b01000001;
7: y[0]=8'b00011111;
8: y[0]=8'b00000001;
9: y[0]=8'b00001001;
default: y[0]=8'b00000000;
endcase

if(b==0)
begin
a=a-1'd1;
b=4'd9;
end
else b=4'd9;

case(b)
0: y[1]=8'b00000010;
1: y[1]=8'b10011110;
2: y[1]=8'b00100100;
3: y[1]=8'b00001100;
4: y[1]=8'b10011000;
5: y[1]=8'b01001000;
6: y[1]=8'b01000000;
7: y[1]=8'b00011110;
8: y[1]=8'b00000000;
9: y[1]=8'b00001000;
```

```
default: y[1]=8'b00000000;
endcase

if(c==0&&d==0)
begin
if(b==0)
begin
a=a-1'd1;
b=4'd9;
c=4'd5;
d=4'd9;
end
else
begin
b=b-1'd1;
c=4'd5;
d=4'd9;
end
end

case(c)
0: y[2]=8'b00000011;
1: y[2]=8'b10011111;
2: y[2]=8'b00100101;
3: y[2]=8'b00001101;
4: y[2]=8'b10011001;
5: y[2]=8'b01001001;
6: y[2]=8'b01000001;
7: y[2]=8'b00011111;
8: y[2]=8'b00000001;
9: y[2]=8'b00001001;
default: y[2]=8'b00000000;
endcase
if(d==0)
begin
c=c-1'd1;
d=4'd9;
end

case(d)
0: y[3]=8'b00000011;
1: y[3]=8'b10011111;
3: y[3]=8'b00100101;
3: y[3]=8'b00001101;
4: y[3]=8'b10011001;
```

```
5: y[3]=8'b01001001;
6: y[3]=8'b01000001;
7: y[3]=8'b00011111;
8: y[3]=8'b00000001;
9: y[3]=8'b00001001;
default: y[3]=8'b00000000;
endcase
```

Future Enhancement

1. Size reduction
2. Interfacing other sensors
3. Improving functionality
4. Use of rechargeable battery or increasing battery life

Relevance

SENSORS AND INSTRUMENTATION:

Interfacing sensor with other modules to build application based project .

Reading and displaying of data from heart sensor to continuously monitor heart rate.

Realizing a capacitive touch sensor for human interrupts.

MICROCONTROLLER:

Applied concepts learnt in the subject to code NodeMCU.

DIGITAL SIGNAL DESIGN:

Verilog code for timer was written and tested.

References

[1] Majid H. Alsulami, Anthony S. Atkins, Russell J Campion (2016), The Use of Smart Watches to Monitor Heart Rates in Elderly People: A Complementary Approach.

[2] Lipeng Fang Xianxiang Chen , Zhen Fang, Kai Tong , Jiankang Liu Zhengling He, Junxia LiMulti, “parameter health monitoring watch”. 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services

[3] Najuka Jagtap, Jagannath Wadgaonkar, Kalyani Bhole , Smart Wrist Watch, 2016 IEEE Students’ Conference on Electrical, Electronics and Computer Science

[4] H. Mansor, M. H. A. Shukor, S. S. Meskam, N. Q. A. M. Rusli, and N. S. Zamery, “Body temperature measurement for remote health monitoring system,” in Smart Instrumentation, Measurement and Applications (ICSIMA), 2013 IEEE International Conference on, pp. 1–5, IEEE, 2013.