Project Report

On

Intelligent Notification System

Developed by: -

Drishti Patel (18162121007)

Guided by:-

Prof. Parth Parekh (Internal)

Mr Krishna Pal (External)

Submitted to

Department of Computer Science & Engineering

Institute of Computer Technology

Year: 2020

Certificate

This is to certify that the IOT Project work entitled **"Intelligent Notification System"** by Drishti Patel (Enrolment No.18162121007) of Ganpat University, towards the fulfilment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them in the CSE (CBA/BDA/CS) Department. The results/findings contained in this Project have not been submitted in part or full to any other University / Institute for award of any other Degree/Diploma.

Name and signature of IBM mentor

Name and signature of Internal Guide

Place:

Date:

Acknowledgement

Application Development project is a golden opportunity for learning and self-development. I consider myself very lucky and honoured to have so many wonderful people lead me through in completion of this project. First and foremost, I would like to thank **Dr. Hemal Shah,** Head of Department, Computer Science and Engineering, who gave us an opportunity to undertake this project. My grateful thanks to **Prof. Parth Parekh & Mr Krishna Pal (Internal & External Guides)** for their guidance in project work **Intelligent Notification System**, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where would have been without his/her help. CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Drishti Patel (Enrolment No. 18162121007)

Abstract

Studies have shown that the stress level of today's population is greater than that of the older generations and due to the stress, we might experience a varying list of diseases among which is heart related diseases. So to keep a check on those type of diseases we would require a monitor which would keep tab of our vitals and notify when there is any irregular functionality seen. We would require a device which is small enough be carried around by people without any inconvenience and having multiple functionalities. We can use the concept of big data to collect and analyse data about vitals from the general public for better predictions of irregularity, and use the concept of IOT to send notification to our personal mobile or laptop.

Chapter 1

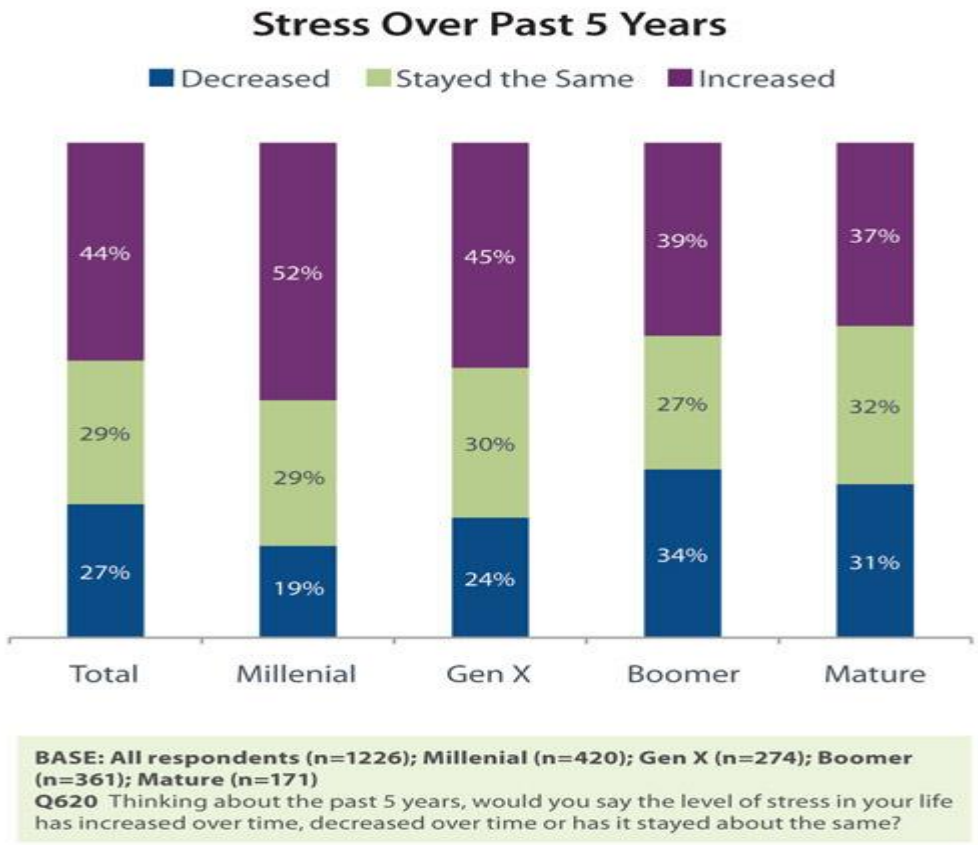
Introduction

1.1: Introduction

Science and technology by the way of inventions and innovations made life easier for everyone in all spheres of life. One of such is in medical science where people are now able to acquire vital medical data from patients. Two of the most important are heart rate and temperature. The first wireless ECG heart rate monitor was invented in 1977 by Polar Electro.

Human cardiovascular system, organ system that conveys blood through vessels to and from all parts of the body, carrying nutrients and oxygen to tissues. Heart rate is one of the very important parameters of the cardiovascular system. The measurement of heart rate is used by medical professionals to assist in the diagnosis and tracking of medical conditions.

In today's world people are dealing with stress and with the stress comes a varying amount of diseases. According to studies, stress levels of high school students are comparable to the stress levels of mental patients of 1930's and 1940's. With the increase in stress levels there have been more cases of early onset of diseases such as heart disease, asthma, obesity, diabetes, headaches, depression and anxiety, gastro-intestinal diseases, Alzheimer, etc.



So as to keep check on the diseases we need a device which would check our vitals and inform us when any irregularity is detected. We have many devices in the market for doing that job but they are often too costly and inconvenient for general use. Our aim to is to build a device which is cost efficient and convenient with max utilities for the use of general public. Heart rate measurement is also used by individuals such as athletes.

Resting Heart Rate	
Infant up to age 1	100 – 160 bpm
Older Children ages 1-10	70 – 120 bpm
Teenage Children 11-17	60- 100 bpm
Adults	60 – 100 bpm
Average of Sex	
Male	70 bpm
Female	75 bpm
Active athletes	40 – 60 bpm

Table 1.1

The average heart beat is between 60 – 100 times per minute. If your heart beats below 60 times per minute, this is bradycardia. The rate of heartbeat is measured in beat per minute (bpm). Having bradycardia means that your heart beats very slowly.

1.2 Objective:

The objective of this experiment is to measure the heartbeat using pulse sensor and Arduino and display the reading via graphical representation. It should also give a notification in serial monitor.

Chapter 2

Project Scope

Due to the cost factor and the availability of present resources we have used arduino and heart beat sensor connected to it. Arduino is a microcontroller board used to send and receive data from the sensor and to present it in such a way that it is understandable. We have kept the size of the device to that of a hand held device so that it is convenient.

Both systems are controlled by Arduino Nano board, which connected to the Bluetooth module and need some programming works. For the software part As and Arduino IDE software have been used.

Chapter 3

Software and hardware requirements

For this project we have used arduino uno and heartbeat sensor.

❖ Required Hardware :

- Arduino UNO
- Pulse Sensor
- Bluetooth Module
- Connecting wires

❖ Required Software :

- Arduino IDE
- Serial Monitor

CHAPTER 4

Pin Description & Configuration

- **Arduino Uno**
  - Pin description

Pin Category	Pin Name	Details
Power	Vin,3.3V,5V,GND	Vin: Input voltage to Arduino when using an external power source 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/output pins	Digital pins 0-13	Can be used as input or output pins
Serial	0(Rx),1(Tx)	Used to receive and transmit TTL serial data
External interrupts	2,3	To trigger an interrupt
PWM	3,5,6,9,11	Provides 8 bit PWM output
SPI	10(SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication
Inbuilt LED	13	To turn on the inbuilt LED
TWI	A4(SDA),A5(SCA)	Used for TWI communication
AREF	AREF	To provide voltage for input voltage

Arduino uno Technical Specifications

MicroController	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12 V
Input Voltage Limits	6-20 V
Analog Input Pins	6(A0-A5)
Digital I/O pins	14(Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Boot loader)
SRAM	2 KB
EEPROM	1 KB
Frequency(Clock Speed)	16 MHz

- **Pulse Sensor Pin configuration**

Pin number	Pin name	Description
1	Ground	Connected to ground of the system
2	Vcc	Connected to +5V or +3.3V supply voltage
3	Signal	Pulsating output signal

Chapter 5

Details of the components:

ARDUINO UNO:

The **Arduino Uno** is an open source microcontroller based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "**uno**" **means** "one" in Italian and was chosen to mark the initial release of **Arduino** Software. The **Uno** board is the first in a series of USB-based **Arduino** boards; it and version 1.0 of the **Arduino** IDE were the reference versions of **Arduino**, which have now evolved to newer releases.

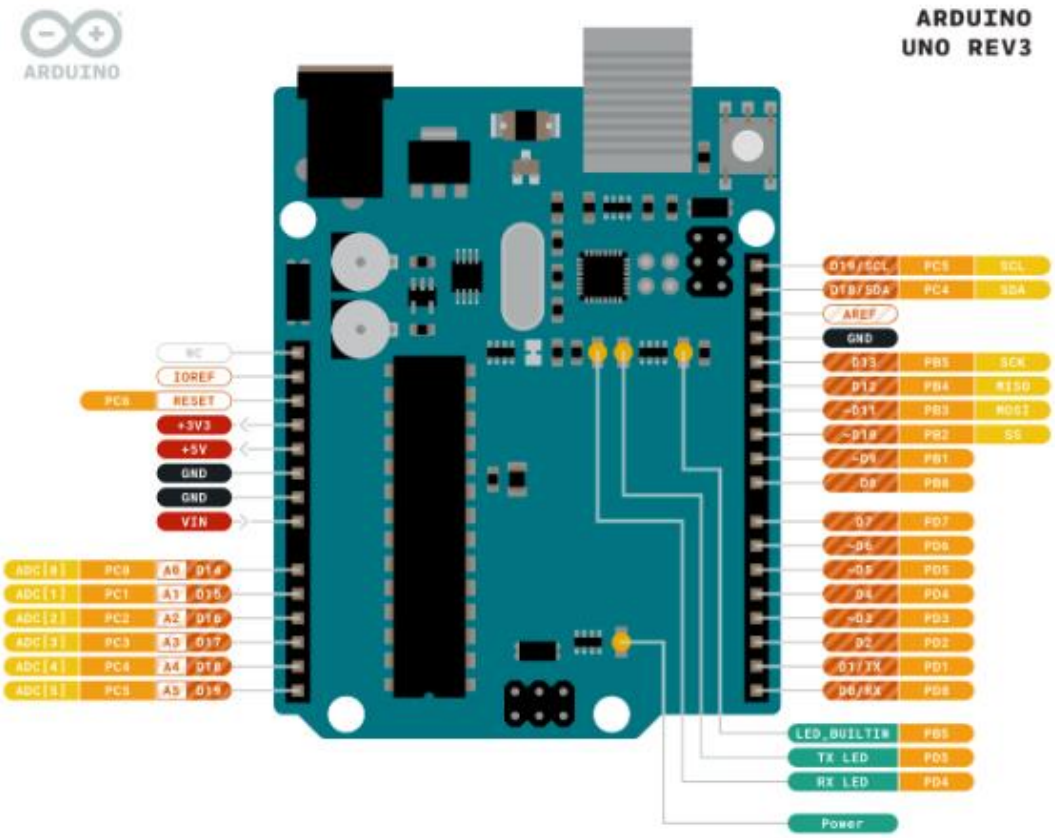
The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

ARDUINO specification:

POWER SUPPLY:

- An Arduino Uno board (Rev 3)
- Standard A-B USB cable
- AC to DC Adapter (7-12V)
- Batteries (9V) with a battery connector

PINOUT DIAGRAM:



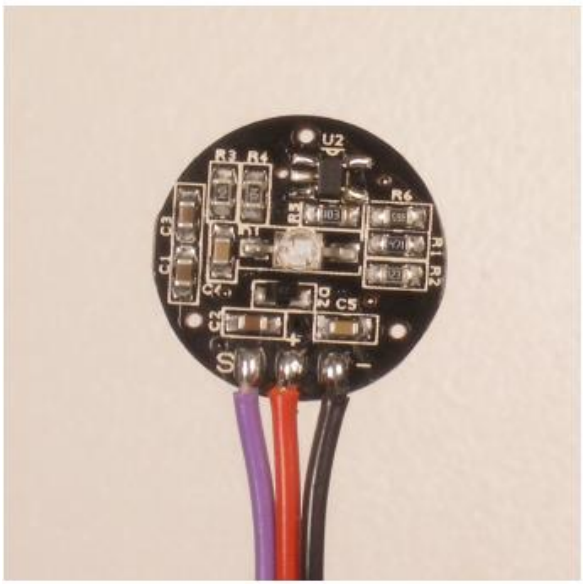
PULSE SENSOR:

Pulse Sensor is a well-designed plug-and-play heart-rate sensor for Arduino. It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart rate data into their projects. The sensor clips onto a fingertip or earlobe and plugs right into Arduino with some jumper cables. It also includes an open-source monitoring app that graphs your pulse in real time.

Features:

- Biometric Pulse Rate or Heart Rate detecting sensor
- Plug and Play type sensor
- Operating Voltage: +5V or +3.3V

- Current Consumption: 4mA
- Inbuilt Amplification and Noise cancellation circuit.
- Diameter: 0.625”
- Thickness: 0.125” Thick



How Pulse Sensor Works:

The working of the **Pulse/Heart beat sensor** is very simple. The sensor has two sides, on one side the LED is placed along with an ambient light sensor and on the other side we have some circuitry. This circuitry is responsible for the amplification and noise cancellation work. The LED on the front side of the sensor is placed over a vein in our human body. This can either be your Finger tip or you ear tips, but it should be placed directly on top of a vein.

Now the LED emits light which will fall on the vein directly. The veins will have blood flow inside them only when the heart is pumping, so if we monitor the flow of blood we can monitor the heart beats as well. If the flow of blood is detected then the ambient light sensor will pick up more light since they will be reflect ted by the blood, this minor change in received light is analysed over time to determine our heart beats.

How to use Pulse sensor:

To use the sensor simply power it using the Vcc and ground pins, the sensor can operate both at +5V or 3.3V system. Once powered connect the Signal pin to the ADC pin of the microcontroller to monitor the change in output voltage. If you are using a development board like Arduino then you can use the readily available code which will make things a lot easier. Refer the datasheet at the bottom of the page for more information on how to interface the sensor with Arduino and how to mount it.

Chapter 6

Project Plan

List of Major Activities:

1. To understand the use of Arduino & Pulse sensor.

2. Establish connections using Arduino.

3. Develop a program to read data from arduino and display the rate in bpm.

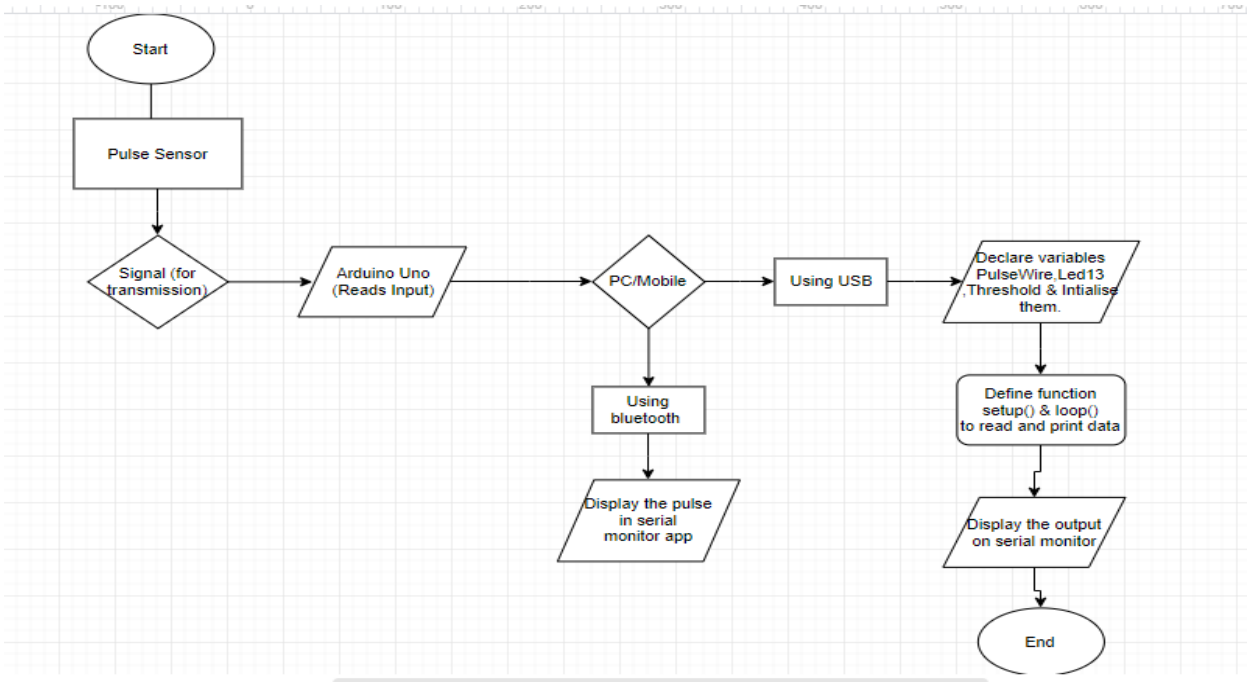
4. To present the data in graphical form.

Chapter 7

Implementation Details

1. **Connect Arduino UNO with pulse sensor.**
2. **Connect Ground, Vcc and signal pin of pulse sensor with respective pins of Arduino.**
3. **Connect the Bluetooth module (Ground, Vcc, Tx & Rx pins) with respective (Ground, Vcc, 10 & 11 pins) of Arduino.**
4. **Install Arduino version 1.8.11 and import the Pulse sensor library.**
5. **Verify the code.**
6. **Open the Serial Monitor to display the heartbeat.**

FLOWCHART:



CODE :

```
#include <SoftwareSerial.h>

#define USE_ARDUINO_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.

#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.

SoftwareSerial BTSerial(10, 11); // RX | TX

//  Variables

const int PulseWire = A0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0

const int LED13 = 13;          // 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.

// SoftwareSerial ourSerial(PIN_RX, PIN_TX);

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



void setup() {


  Serial.begin(9600);          // For Serial Monitor //Speed to transfer data

BTSerial.begin(38400);
```

```
  // Configure the PulseSensor object, by assigning our variables to it.
  pulseSensor.analogInput(PulseWire);
  pulseSensor.blinkOnPulse(LED13);       //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.
  }
}




void loop() {


 int myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor object that returns BPM as an "int".
                                               // "myBPM" hold this BPM value now.


if (pulseSensor.sawStartOfBeat()) {
  // Constantly test to see if "a beat happened".
 Serial.println("♥  A HeartBeat Happened ! "); // If test is "true", print a message "a heartbeat happened".
```

```
 Serial.print("BPM: ");                        // Print phrase "BPM: "
 Serial.println(myBPM);
 BTSerial.println("♥  A HeartBeat Happened ! "); // If test is "true", print a message "a heartbeat happened".
 BTSerial.print("BPM: ");
 BTSerial.println(myBPM);
}// Print the value inside of myBPM.


//  if (BTSerial.available())



  delay(20);                  // considered best practice in a simple sketch.//for controller 1s.


}
```

Pictures:

```
♥  A HeartBeat Happened !
BPM: 64
♥  A HeartBeat Happened !
BPM: 90
♥  A HeartBeat Happened !
BPM: 90
♥  A HeartBeat Happened !
BPM: 88
♥  A HeartBeat Happened !
BPM: 87
♥  A HeartBeat Happened !
BPM: 86
♥  A HeartBeat Happened !
BPM: 87
♥  A HeartBeat Happened !
BPM: 88
♥  A HeartBeat Happened !
BPM: 87
♥  A HeartBeat Happened !
```

Graphical Representation :

Precautions/Necessary Measures:

- During placing the finger on Pulse sensor, extra care should be taken so that it remains in proper place to read data.
- Wires should not be loosely connected.

Relative Background:

- We have used the concept of IOT (Internet of things) which is a system of interrelated computing devices, mechanical and digital machines, objects that are provided with unique identifiers and the ability to transfer the data over a network without requiring human interaction.
- We have used some basic components of IOT such as Arduino Uno, Sensors (Heartbeat and temperature sensors) and connected all of them to get the desired output**.**

Chapter 8

Conclusion and Future work

Conclusion

To conclude, we would revisit all the covered points.

i. Increase in stress levels and corresponding diseases
ii. Need of a device to monitor the vitals
iii. Send a notification in case of any irregularity
iv. Device should be affordable, convenient and have max functionality

Future scopes:

Monitoring device that could detect heart beat anomalies of physically challenged individuals without hands.

It can be supported using mobile technology**.**

My contribution :

1.Establishing connection using Arduino.

2.Connection in mobile(via bluetooth module) & laptop.

3.Pivotal Tracker & Github.