Autonomous Smart Emergency Response Initiator for Women Safety

Project Report submitted in fulfillment

Of

Bachelor of Technology

In

Electronics and Telecommunications Engineering

By

Jas Patil (C041)
Varad Sheorey (C051)
Rishabh Singh (C055)

Under the supervision of

Prof. Sonal Parmar
(Asst. Professor, EXTC Department, MPSTME)

SVKM's NMIMS University
(Deemed-to-be University)

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING
Vile Parle (W), Mumbai-56

2020-21

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING

Vile Parle (W), Mumbai-56

2020-21

CERTIFICATE

This is to certify that the project entitled "Autonomous Smart Emergency Response Initiator for Women Safety" has been done by Mr. Jas Patil, Mr. Varad Sheorey and Mr. Rishabh Singh, under my guidance and supervision & has been submitted in partial fulfillment of degree of Bachelor of Technology in Electronics and Telecommunication of MPSTME, SVKM's NMIMS (Deemed-to-be University), Mumbai, India.

Prof. Sonal Parmar

(Project Mentor)

Date:

Place: Mumbai

Dr. Avinash More

(HOD – EXTC)

ACKNOWLEDGEMENT

We would like to express our most sincere appreciation and gratitude to all those who made it possible for us to turn this idea into a reality. We would like to give special thanks to our mentor Prof. Sonal Parmar, whose constant encouragement and suggestions helped us achieve our goal. Furthermore, we would also like to acknowledge with much appreciation the crucial role that was played by the project coordinators who constantly provided guidance and motivated us.

We would also like to give many thanks to our Head of Department, Dr. Avinash More whose efforts in guiding the team helped us in achieving our goal. We are thankful for our Project coordinators: Prof. Vipul Gohil, Prof. Ami Munshi, Prof. Amey Raut, Prof. Vidya Sawant, for their suggestions given by them during project presentations.

Jas Patil	Varad Sheorey	Rishabh Singh
B.Tech (EXTC)	B.Tech (EXTC)	B.Tech (EXTC)
C041	C051	C055

ABSTRACT

In today's world when there is equality in every field and women are having more opportunities than ever before they are still suffering to go out and prove themselves because of safety concerns. Moreover, the major problem is that the safety concerns are not bound to a specific range of age of women as is touches every range of women from adolescents to old aged. In this world where we can switch on our daily appliances from far away, we still have very less technological advancements in the field of women safety. The cases of women harassments before the pandemic were just increasing and it required a lockdown to impede those number of cases and put them into abeyance.

So, to keep this low rate steady this paper proposes a system using modern technologies which operates on itself when the pulse rate increases, which are aberrant when exposed to harassment in society. The security is provided by continuously monitoring the pulse rate. The pulse rate is detected by using a pulse sensor. When there is fluctuation in pulse rate that is not exhibiting normal rate, the device gets activated. A mobile app is used for location tracking. The software or application is pre-programmed in such a manner that whenever an emergency signal is sent, the app searches for the nearest available police station and an alert message along with the location co-ordinates are sent to them. The message is also forwarded along with the location details to the already added emergency contacts.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter accentuates the significance of the system designed, its importance, the motivation behind this for creating the system. The need of a multipurpose system that can be made available to the general public.

1.1 Background of the project

In today's world where there's equality in every field and women are having more opportunities than ever before. There is still a feeling of malaise that's holding them back. They are still hesitant to go out and prove themselves because of safety issues. Moreover, the major problem is that the safety concerns are not related to a specific age range of women it is touches every range of women from adolescents to old aged. In this world where we can switch on our daily appliances from far away, we still have very less technological advancements in the field of women safety.

Traditionally women were not allowed to go out on their own after dark which resulted in low crime rates. If we still keep the same mentality then we can save women from harassment but we cannot let them live fearlessly and independently. If we look at reports of the government the 1st and 2nd most common crime against women in India are Rape and Sexual Harassment and it is not located to a specific place the sad part is it is common in each part of the country. So, it can't be about traditions and culture that's causing this this is a common human mentality which is causing this. Just according to the annual report of 2018, 33,356 cases of rape were reported in India. We can't change the mentality of people in one day but we can at least boost the confidence of women with the help of technology giving them assurance of safety. In 21st century taking one's freedom is of the biggest crimes. And the system proposed will help in achieving that.

There are watches or other devices in the market that have been made specifically to limit these events. There is one thing in common in all these devices that they require physical triggering. Whenever someone finds themselves in such situation then they have to press a button located on any device like watch or phone to alert and send an SOS message. But it has been proven that during such scenarios the quality of thinking wisely drops and a person can't think effectively [1] [2]. If there is a system that thinks for them in such scenarios it would be helpful in alerting people so that the victim can have time to flee and not think about alerting people which can save lives.

The cases of women harassments before the pandemic were just increasing and it required a lockdown to impede those number of cases and put them into abeyance. So, to keep this low rate steady we are proposing a system using modern technologies which is a self-initiation device which will operate the pulse rate increases, which are aberrant when exposed to harassment in society. The security is provided by continuously monitoring the pulse rate. The pulse rate is detected by using pulse sensors. When the rate of pulse is increased more than that of normal rate, the device gets activated. A mobile app is used for location tracking. This app searches for the nearest available police station and an alert message along with the location details are sent to them. The message is also forwarded along with the location details to the already added emergency contacts.

1.2 Motivation and scope of the project

Every man has a woman in his life they care about. She can be a mother, a sister, a wife or his daughter. And the irony is women are afraid of men who doesn't treat them right and try to take advantage neglecting the fact that they are also someone's sister, wife, daughter or even mother. The most shocking part of this is the age factor, nowadays even the girls who haven't entered their teens are not safe. We all have read articles, headlines on newspaper and on our televisions sets that women are being mistreated and all of us get angry or frustrated for a brief period of time but after some hours we forget everything. Instead of just getting furious on the world and criticizing the world we can at least try to make changes and this project is just a small effort to start taking a step towards the change. And the motivation comes from this only as we can try to make the world a better place for our younger generations.

Violence against women in India is so common that on an average 848 Indian women are raped, harassed or killed every single day [4]. This number in itself is derogatory for a nation which cannot keep its people safe. India is named as World's biggest democracy and has the largest federal constitution. In the constitution they have 7 stated women specific legislation for their safety yet it has such poor record. Over 415,786 sexual violence cases have been reported in India between years 2001 and 2017. Indian Incorporated has also reported in 14% rise in sexual harassment complaints in just year 2019. The worst part is it is still very difficult to find justice to these victims as in many cases the police has been unable to track the assailants which can be seen in conviction rates, in 2017 the conviction rate was 32.2% not even 1/3rd which reduced further to 27.2% in 2018.

The elaborated study gave us certain insights on how to implement the design which can be safe and prove to be best for the user. The ways to implement this system are countless as the technologies are improving every day and everyday there is an update in sensors and apps which can be used to give better results in future. At present, the system we have designed has taken ideas from each paper that has been researched to make the best possible design achieved yet. We have known a new technique of EDR which is yet to be proven as it has many drawbacks. We will be using Bluetooth technology so we have found out the integration and prices of using it effectively [5,7]. To make system more up to date and allow more functionalities we will be incorporating an app which will also make the band compact [8].

1.3 Problem statement

In today’s world, violence against women is not uncommon. Everywhere in the world women are put through such emotional and mental and physical trauma that it’s just a grave violation of basic human rights. It negatively affects their well-being and prevents them from fully enjoying the pleasures that life has to offer. Violence not only has negative consequences to woman but it is also painful and unbearable for the family to see their loved one’s go through such trauma. India is considered as one of the world’s most dangerous country for women with rape being one of the most common crimes committed. Not just in India, this problem is present everywhere in the world. Global estimates published by WHO indicate that about 1 in 3 women (35%) have experienced either physical and/or sexual intimate partner violence or non-partner sexual violence in their lifetime.

Even though we live in an age where security is everything, from online security to personal security, can't we use this same technology which will make strides in the area of women safety, if proper resources are devoted to each other? The design of the device that can help prevent such abhorrent acts must be addressed. Thus, in this paper, proposed is the design of a Smart Emergency Response Initiator (SERI).

Chapter 2

Literature Survey

In the past several years, many researches were carried out in this sector of women safety. Many Apps were developed in-order to achieve a safer society for women, in India, development of women safety Apps came into picture after 2012 Delhi gang rape and murder of a 23 years old student, 'Nirbhaya'. The first thing any medical personnel is interested in knowing is the pulse and temperature of any patient, also pulses in the form of ECG and the body temperature. When setting up for an ECG, there is a need of a trained medical personnel which increases resource usage.

This chapter will help us understand the working of existing system and in the latter part of this chapter we will see how these systems are not as affective in certain situations. Many different systems are available for women safety such as VithUapp, Nirbhaya, SpotNSafe Feel Secure app.

The basic ideology behind all the available systems is the same, with these similarities they almost share the same limitations and drawbacks. Following is the basic outline of the available women safety app.

- Keep an emergency contact list, consisting of friends, family and police hotline.
- Track location of the user at all times or when it is changed over a threshold.
- A physical triggering in case of danger which sends a distress message to the emergency broadcast list.

Now if we must understand the aforementioned points to better understand the endless possibilities they can offer and what else can help us get a better system based on their understanding. When you're in danger, you call out for the people who are closest to you, whom you trust. It is important to broadcast a distress message to such a group of people who will take all the necessary steps to help you in an unnatural situation. When deciding on a broadcasting group, not only close people must be chosen but also people who keep a check on their phones more frequently. Many emergency hotlines have been started for people in need and thus such hotlines must also be considered. Knowing the location of a person can make it easier for the people searching for the person in danger. At times, when in an unavoidable situation making a call and letting them know about the location is highly unlikely and thus having hold of location of the victim can be very useful.

Finally, when in danger the victim must have a way of letting the people know about their location and letting them know that they might be in imminent danger. Making a call or at times even messaging can be difficult so there must be a way of doing the same but without elapsing a hefty amount of time. It can be by clicking a button on the phone or by using gestures on the screen.

The literature survey was done by studying and keeping the above-mentioned points. When the pulse rate of the user crosses the predefined threshold, the system sends an alert message to the emergency contact. The system is triggered by exceeding the pulse rate threshold which is controlled by Arduino NANO. This distress signal is pinged to the app with the help of a bluetooth module, HC-05. The entire system is autonomous and can be carried anywhere as explained by Pallavi Raj et al. in their paper "A Smart Band for Women Safety" [1].

The system proposed by Siddharth A. Kokalki et al. in their paper "Smart Health Band using IoT" [2]. Can be used for women safety with an introduction of testing feature. The authors have proposed an autonomous system for health monitoring. Two sensors, pulse and temperature sensors are used in the proposed system, the system is implemented using ESP8266 microcontroller that makes Wi-Fi implementation easier. The entire collected data is stored in cloud provided by "thinkSpeak". The same is also echoed on the app built by the authors.

In recent years the cases against women harassment has increased exorbitantly. There was no definite solution as all the devices available required a self-triggering or were bulky like stung gun. But there is an unfortunate fact that whenever a person is attacked, their ability of think logically is lowered. So, in this paper a wrist wearable device was design and implemented using a few sensors to sense anxiety or fear and a bank of capacitors to shock the assailant if they come close. The information about the attack is then directly sent to a few emergency numbers with a message which is connected to a smart phone.

The proposed system consisted of two modules fear sensing system and android based app [3]. The fear sensing system consisted of a pulse sensor and a skin conductance sensor. The pulse sensor is linked with anxiety and fear as there is secretion of adrenaline here is a fluctuating of pulse which can be monitored using this sensor. Skin conductance sensor is used as duding such situation our body emits sweat which can also find out if a person is in danger as small changes in the conductance of the skin can be detected. After finding out that the person is in danger second module comes into play. Android based mobile app sends emergency message to a few emergency contacts along with that a few capacitors start charging to give electric shock to the assailant so that the person can get some crucial time for fight or flight. The concept of EDR used for skin conductance sensor and by doing this a bank of capacitors are placed in the band which can provide electrical shock in case of emergency. The heart of the system is a low power, high performance micro-controller [3]. The duty of MCU is to monitor the data from sensors and in case of emergency communicate wirelessly to the mobile phone using Bluetooth module. Instead of using a GSM separately and increase the bulkiness of the band they used mobile phone which everyone carries today. The data is sent from the micro-controller to the Bluetooth module via the UART.
The data flow is such that first the sensors detect any fluctuations in reading. If everything is the same nothing happens. If the values cross a threshold value then the micro-controller is triggered which commands the mobile phone. Along with that the capacitors are set to charge and are ready to give shock. Till that time the application in mobile phone will send location to the emergency contacts to make them aware of the situation.

Even in today's world the thing that haunts every woman is when they will be able to move freely without worrying about their security. 848 woes are harassed, raped and murdered every single day in India which is beyond huge number. As we all know humans cannot aptly react in critical situations, we need a device which can trigger automatically. They proposed a system which is the integration of multiple devices in which hardware components continuously communicate with the smart phone that has access to internet. The application consists of data which contains threshold values and this generates a signal which is transmitted to smart phone. The application has access to GPS and Messaging app which can send coordinates and message to nearly police station and emergency contacts.

G. C. Harikiran et al. proposed an advanced system can be built that can detect the location and health condition of person that will enable us to take action accordingly based on electronic gadgets like GPS receiver, body temperature sensor, GSM, Pulse rate sensor [4]. The heartbeat of a person in such situations is normally higher which helps make decisions along with other sensors like motion sensors to detect the abnormal motion of the women while she is victimized. The plan to use GSM and GPS of the mobile phone with can reduce the bulkiness of the device. This also enables in reduced power use and that the watch can be installed with Bluetooth 4.0 BLE (Bluetooth Low Energy) which comes in handy for several days on a single shot of charge. There were plenty of existing systems but the main drawback in those was self-triggering was required.

The proposed model was that a Smart phone can be connected to the smart band through BLE. A specific designed application can act as an interface between the phone and the band. In case of an abuse, the app can directly send message to family contacts, can send coordinates to nearby police station for immediate action. Control Unit collects information from smart wrist unit and GPS receiver. GSM module will then send all this information from control unit to base station. Wrist unit collects the data from human using body temperature sensor, pulse rate sensor and switches. RF module is used to send data from wrist unit to the control unit. A pulse sensor and temperature sensor are used for keeping track on vitals. A GSM is used to send data from control unit to base unit. A motion sensor is a device that detects moving objects. BLE is designed to connect devices with low power consumption. And GPS is used to find the location. The proposed system was well designed but it contained some flaws. The system is highly dependent on the sensors and in case of false triggering there is no way to cancel the message to send.

A paper titled "Android and Bluetooth Low Energy Device Based Safety System", proposed by P. Saikumar et al. proposed a system comprising of an Arduino NANO Controller, Bluetooth module, Taser and an Android versatile. When it is actuated, it tracks the location using GPS and send message using GSM through mobile portable application. This gadget comes along with a taser which can stun the assailant [5]. The proposed system will comprise of heartbeat and temperature sensor. During such situations it has been observed that the temperature and pulse of the women rises. When the sensors are initiated, they start sharing qualities to the dataset to be contrasted every 10 seconds.

The system will have a 3.7V LIPO battery and it is battery-powered which is associated with the charging circuit and it has a change to start the security gadget. The Emergency switch is utilized when the individual requires help and when the crisis catch is squeezed, it enacts the Taser which produces stun instrument and it likewise actuates the Bluetooth module to interface with Android portable Application which sends message and call utilizing the GSM in versatile to the SOS contacts. The SOS contact individuals will get the predefined message and current area of the individual who needs assistance utilizing the portable GPS in Android Mobile. The system will consist of the number of a police station and emergency contacts for sending the message in case of crisis. In this HC-05 Bluetooth Module has been used which is also alluded as SPP module. With the help of HC-05 it is very easy to interface the controller as it used sequential correspondence. HC-05 Bluetooth module gives exchanging mode and slave mode which implies it ready to utilize both accepting and transmitting information. They used a Linux based operating system for the android app.

The system proposed is a wrist band style model which continuously monitor the user's vitals to check for uneasy situations [6]. The advanced sensors such as temperature sensor, heart beat sensor, sweat sensor and an accelerometer sensor will continuously check the temperature, heart beat and sweat to check the emergency conditions. The system provides three switch buttons. Two switch buttons are used to give alert and the other one is used to cancel the alert. If any attack or emergency condition occurs the GPS module will track the location and send this location and emergency messages through GSM module to the emergency contacts that are predefined by the user. The device consists of an LCD display to display the message. A screaming buzzer and an LED light is also used to provide the alert at the time of emergency situations. A power supply unit is also provided for the device.

For establishing connection between the app and the hardware system that is responsible for all the processing, HC-05 interfacing with Arduino controllers go well together and the implementation is also not that complex and bluetooth can connect up to 8 devices simultaneously [7]. It is a low-power, RF operating at 2.4 GHz and up to 5 volts is required for powering the module. As this is a low power consuming module it can easily be used with a wide range of processors and sensors. Such an implementation is explained by P. L. Penmatsa et al. using Arduino Mega with sensors and transmitting it to other devices using bluetooth [8]. The authors suggested on using such a compact device that reduced the operating time by 50% and also reduced the requirement of medical personnel.

Inclusion of volunteer's real time mapping on the victim's app and echoing the real time location of the victim on the volunteer's app when a distress signal was issued by the victim [9]. The authors suggested to divide the map in multiple zones by which when a distress signal is issued by the victim, the volunteers in that particular zone would get the location of the victim in real time and so would the victim. In most of the papers we saw that the hardware was connected to phone using bluetooth via, an app. Making an app might be complex and very tedious but T. Munasinghe et al. suggested to use MIT App Inventor for analysing sensor data [10]. Using MIT App Inventor, many functionalities could be programmed into the app. Messaging can also be controlled by using MIT App Inventor and can also be easily configured for other Android configurations.

For implementing the project, we must take into account that the system is complaint with all the standards given out by IEEE SA which is Institute of Electrical and Electronics Engineers Standards Association;

1. IEEE 802.15.1 - IEEE Standard for Telecommunications and Information Exchange Between Systems.

2. IEEE 1118.1-1990 - IEEE Standard for Microcontroller System Serial Control Bus

These standards are laid down to make the interconnectivity, interoperability and uninterrupted functioning of different projects using similar or same technologies within them. To create uniformity, it is necessary to follow these standards to ensure smooth functioning of the implementation worldwide.

The first one in the list defines the standard for information exchange using Bluetooth. It is a protocol laid down for Bluetooth communication for serially receiving and sending information between two Bluetooth devices over 2.4 GHz wireless link [11].

Since, the project is built around a microcontroller and which be responsible for all the operations, the IEEE standard for Microcontroller System Serial Control Bus is referred to [12].

Chapter 3

Methodology and Implementation

In this chapter, block diagram for the proposed system is discussed along with the hardware description. The software implementation is discussed in the latter part of the chapter.

3.1 Block diagram

The block diagram of the proposed system is illustrated in Figure 3.1, the sensors, Pulse and temperature are input to the processor, which serially outputs the processed data to the Bluetooth module, which in turn transmits it to the App.

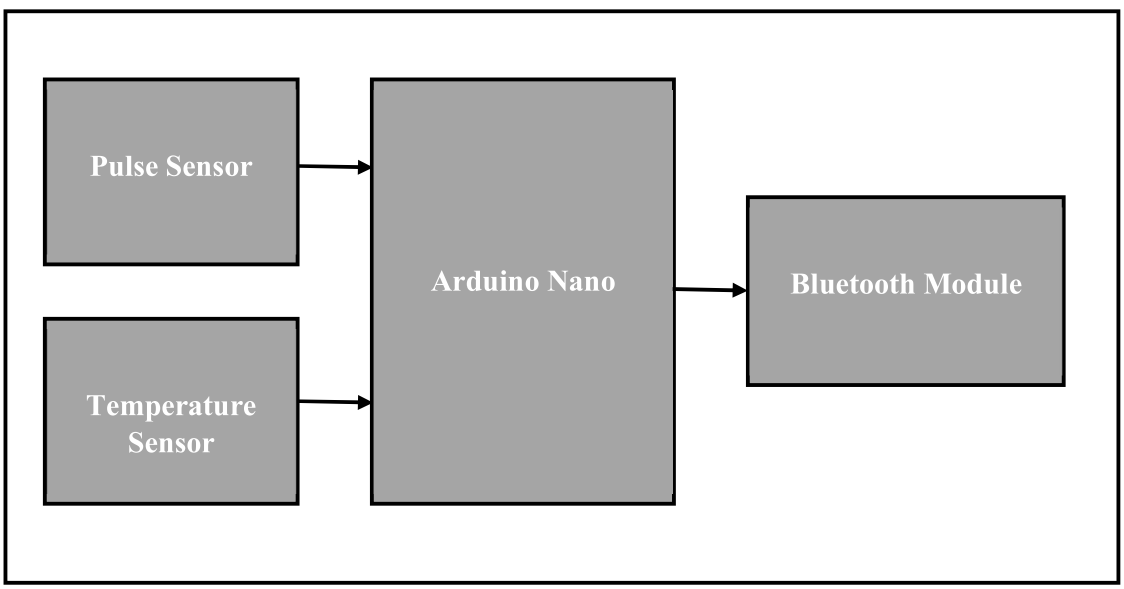


Figure 3.1 Block diagram of Designed System

The pulse sensor is used to get the pulse rate and gives an analog output on the analog pin of the Arduino Nano. This is in the form of a raw ECG which is then later used to obtain the pulse using software manipulation. Temperature is another parameter that is monitored and is done using DHT-11 sensor. This sensor returns two readings, the temperature and the environment humidity, however the humidity is not processed.

The processor used is an 8-bit microcontroller with 32KB flash memory, making it easier to work with large computation demanding processes. The controller is used for all the processing and working with all the components in unison. Bluetooth module is again a serial communication device that is connected to the controller and communicates with the app via a wireless link.

3.2 Hardware description

In this section we will understand the various components used for the implantation of the system.

3.2.1 Arduino Nano

The Arduino Nano board is a micro controller board which is based on ATmega328. It is a single chip micro controller and the architecture of Arduino Nano is customized Harvard architecture with an 8-bit RISC processor core. This board includes digital 14 Input/Output pins, a power jack, USB connection, an RST button, ceramic resonator of A16 MHz and a ICSP header. For the power supply to the board an AC to Dc adapter can be used or else a battery. The features of Arduino Nano are:

- The operating voltage of the board at 5V.
- The recommended input voltage should range between 7V to 12V.
- The input voltage ranges between 6V to 20V.
- There are 14 Digital input/output pins.
- There are 6 analog input pins.
- For each input/output pin the DC current is 40 mA.
- For a 3.3V pin the DC current is 50 mA.

- It supports a 32 KB of Flash Memory.
- Size of SRAM is 2 KB.
- Size of EEPROM is 1 KB.
- It offers a clock Speed of 16 MHz

3.2.1.1 Pin diagram

Figure 3.2 a and Fig. 3.3 b show the pin configuration of Arduino Nano, along with indication of different types of pins. For building an Arduino Nano board the components required are power pins, ATmega328, Reset Button, analog pins, CSP header, USB interface, power LED, digital pins, TX/RX pins, set led 13 and an external supply.

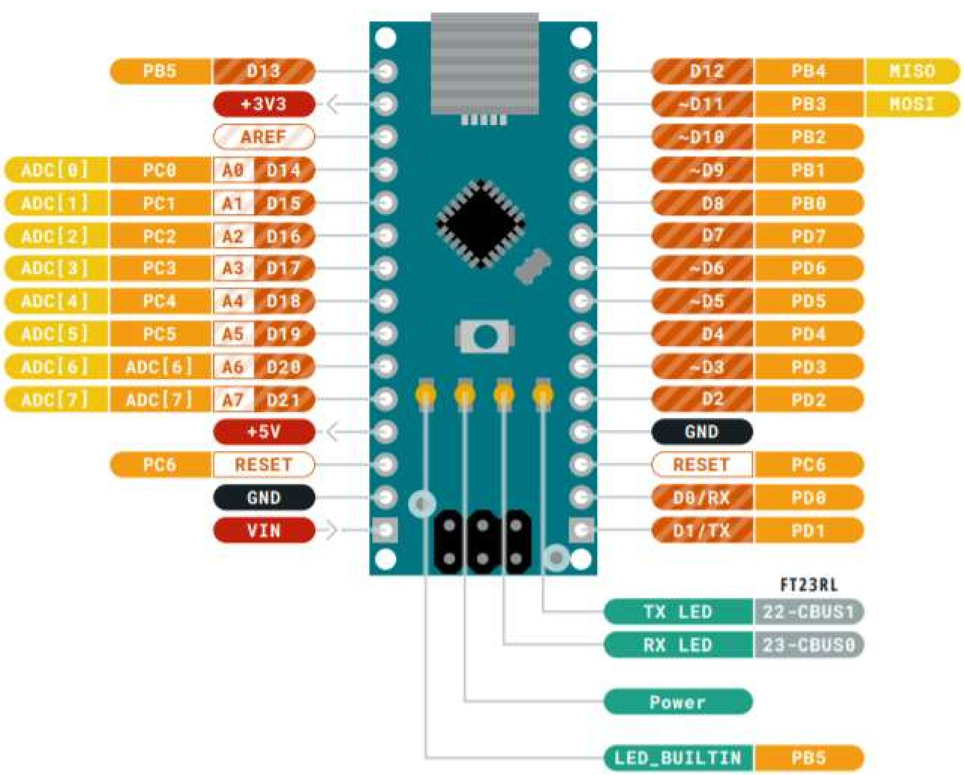


Figure 3.2 (a) Pin diagram of Arduino Nano (Top View)

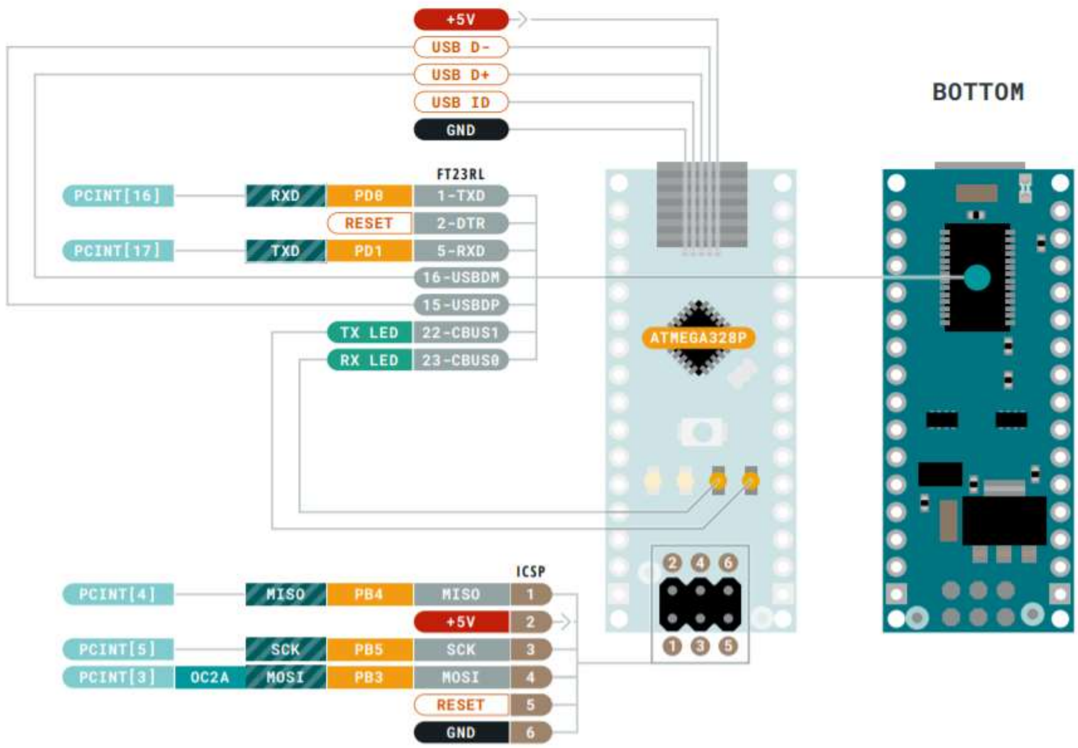


Figure 3.2 (b) Pin diagram of Arduino Nano (Bottom View)

A. **Power Supply:** To supply power to the Arduino Nano board it can be done with the help of an USB cable or an external power supply. AC to DC adapter or a battery mainly includes an external power supply. The connection of the adapter to the board can be made by plugging in a. power jack. n case of connecting a battery, battery leads are connected to Vin pin and GND pin the Power connector. The range for the suggested voltage should be 5 V.

B. **Input and Output:** There are 22 digital pins on the Arduino Nano board which can be used as input & output with the help of the functions like digitalWrite(), pinMode() & Digital Read().

C. **D1 (TX) & D0 (RX) (Serial):** These are used transmitting and receiving TTL serial data and are connected to the ATmega8U2 USB to TTL Serial chip equivalent pins.

D. **D2 & D3 (External Interrupts):** They can be connected to activate an interrupt over a low value or when there is a change in value.

E. **D3,D5, D6, D9, D10, & D11 (PWM):** These pins give 8-bit PWM output/power by the function of analogWrite().

F. **SPI Pins (D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK):** They maintain SPI-communication and even though they are offered by the fundamental hardware it is not presently included within the Arduino language.

G. **D-13(LED):** The inbuilt Light Emitting Diode can be connected to a digital pin (D13).As it is a High-value pin, the LED will be activated whenever the pin will be low.

H. **D4 (SDA) & D5 (SCL) (I2C):** They support TWI-communication with the help of a Wire library.

I. **AREF (Reference Voltage):** The analog inputs with analogReference() will be having a reference voltage.

J. **Reset Pin:** For resetting (RST) the micro controller this pin is used.

K. Memory: The memory of the Arduino Nano micro controller board includes a Flash memory of 32 KB which is used for storing the code. Apart for that it has a SRAM of 2 KB and EEPROM of 1 KB.

L. Communication: The Arduino Nano offers an UART TTL serial communication which is accessible on digital pins like TX (1) and RX (0). The Arduino has a software which has a serial monitor that permits easy data. Whenever the data is being broadcasted through the USB the two LEDs on the board will blink.

3.2.2 DHT-11

It is a simple, low cost digital humidity and temperature sensor. It functions by using a capacitive humidity sensor along with a thermistor to measure the surrounding air and after that it spits out the digital signal on the at a pin. It is very simple to use but for precise data it requires a bit time. Also, it can take account of new data once every 2 seconds. It can be a downside but the plus point of this sensor is that it is compact than other devices available so if we give it sufficient time it can give precise readings and for more readings, we need to use a different type of sensor because almost every temperature sensor takes a bit more time to read new data.

Following are the specifications of the DHT-11 sensor,

- The operating voltage of DHT11 is around 3.5 V to 5.5 V.
- In the case of operating current, it measures at 0.3 mA and the standby is 60 µA.
- Output is through Serial Data.
- The temperature range is from 0°C to 50°C which is sufficient for our project.
- It can measure humidity from 20% to 80% range.
- The accuracy of the temperature reading is approximately ±1°C.
- The accuracy of the humidity reading is approximately ±2%.

3.2.2.1 Pin identification

In Figure 3.2.2.1, a 3 pin DHT-11 sensor is show along with pin identification. Following gives the description of the various pins of the DHT-11 sensor;

- Vcc - Range of Power Supply should be between 3.5 V to 5.5 V.
- Data - This port outputs both humidity and temperature through serial Data.
- Ground - This pin is connected to the ground of the circuit.

Figure 3.3 DHT-11 pin identification

3.2.2.2 Communication process

The format that is used for communication is single bus data format and it is synchronized between micro controller and DHT11 sensor. Time taken for one communication process is about 4ms. Data consists of integral and decimal parts. When MCU sends start signal then DHT11 changes from low power consumption on running mode. After completing the start signal, DHT11 sends a response signal of 40-bit data and then triggers a signal acquisition. After that users can choose to collect (read) some of the data. If there is no start signal from MCU then DHT11 will not collect temperature & humidity information spontaneously. After collecting the data, DHT11 will change its mode to low-power-consumption until it receives a start signal from micro controller again and the process is again repeated.

3.2.3 Pulse sensor

Pulse sensors are wearable plug-and-play devices that can be used by athletes, students, game developers, anyone in the whole wide world. This simple device has made it easier to incorporate a user’s heart rate in different real-time projects. The sensor can be clipped onto the user’s finger-lobe or it can be connected to an Arduino Nano/Mega board or it can be directly connected to a breadboard.

The sensor needs around 5V to 3.3V of operating voltage with a current consumption of 4mA. The sensor has a led along with an ambient light sensor on one side and a noise cancelling and amplification circuit on the other. The working of the sensor is simple. The led should be placed exactly over a vein. When the blood flows then we can measure the heart rate. So, when the flow of blood is detected, the sensor will pick-up more light that is reflected back by the blood and this minor change in receiver light is measured over time to establish our heart rates. Figure 3.2.3 shows front and back view of a typical pulse sensor.

(a) (b)

Figure 3.4 Pulse sensor (a) front view, (b) back view

The pulse sensor has only 3 connections. Pin 1 is used for grounding the system. Pin 2 is used for connecting the sensor to a +5V to +3.3V supply voltage and Pin 3 is used for obtaining the pulsating output signal.

3.2.4 HC-06

HC-06 is a Bluetooth module designed for establishing short range wireless data communication between any two systems. It works on Bluetooth 2.0 communication protocol and it acts only as a slave device. The module can transmit files at the speed of 2.1Mb/s and is a cheaper and more flexible method of wireless data transmission.

The device works in the frequency range of 2.402 GHz to 2.480 GHz. It uses FHSS to avoid interference and have full duplex data transmission, i.e., it is capable of bi-directional network data transmissions at the same time.

The communication with the HC-06 module is completed using UART interface. SO, the module can be connected directly to a pc or microcontroller which has a RS232 port (for UART interface). After connecting the module, you have to program it in Arduino IDE to send and receive data to the module. To ensure a successful and complete wireless data transmission we need to take care of a few things:

- You need to set a default baud rate of UART serial communication to 9600. The value is a default setting of the module and can be changed within the program.
- This Bluetooth module only works as a slave so another device that works as a master is required in order to communicate with it.
- The master searches for slave and connects to it after a password is typed. A default password is set, i.e., '1234' which can be changed during the coding process.

- After authentication, the data that the master device sends can be used to perform simple tasks (by the slave), defined in the program.
- There are certain libraries available that make the communication between the master and slave device easier. Once these libraries are downloaded and called into the program, by using simple commands you can send or receive data by telling the Arduino what to do. If the module receives data from the master, it will be communicated to the Arduino board by UART serial communication.

Figure 3.5 Depicts HC-06 Bluetooth module,

Figure 3.5 HC-06 module

Table 1 shows pin description of HC-06 Bluetooth module, HC-06 is a 6-pin module that can be seen in Figure 3.6,

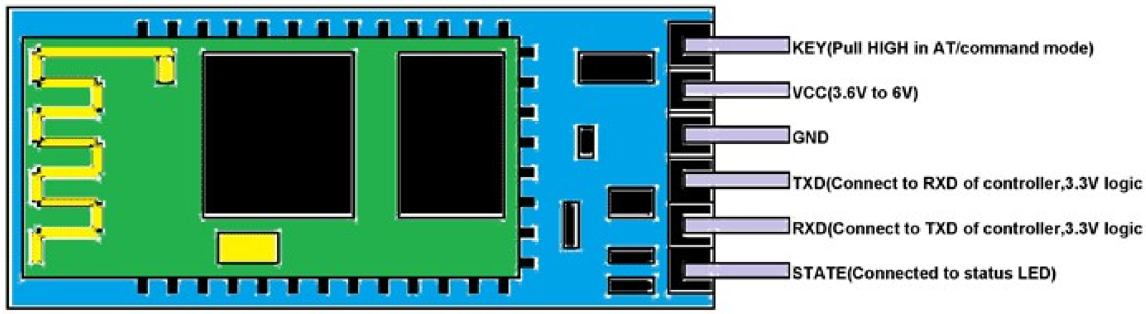


Figure 3.6 Pin identification of HC-06

Pin	Name	Function
1	Key	The pin determines whether the module works in AT command mode or normal mode [High = AT commands receiving mode (Commands response mode), Low or NC = Bluetooth module normal working]
2	Vcc	+5V Power Supply needs to be given to this pin for powering the module

3	Gnd	Connect to ground
4	TXD	Serial data is transmitted by the module through this pin (at 9600 bps by default), 3.3 V logic
5	RXD	Serial data is received by the module through this pin (at 9600 bps by default), 3.3 V logic
6	State	The pin is connected to the LED on the board to represent the state of the module

Table 1 Pin description of HC-06

3.3 Software

Arduino IDE is used to program the Arduino Nano. The Arduino IDE or the Integrated Development Environment is a cross-platform Software which is provided by Arduino to program Microcontrollers. It is an IDE built around C and C++ and thus the functionalities can be developed using C and C++ and could be used to program the controllers.

Apart from Arduino IDE, MIT App Inventor 2 was used to develop App for the android phone. App Inventor is a web application IDE provided by Google and not maintained by Massachusetts Institute of Technology. It is a good application for beginners to create an application software for Android. It is a free and open-source software which makes it easier for community collaborations.

3.3.1 Libraries

While designing the program for the controller we used the following libraries for better functionalities.

A. SoftwareSerial

When Arduino NANO is connected to the PC terminal, it continuously communicates with the PC terminal with its serial communication bus which is also pins 1 and 2 named Tx and Rx respectively. When another serial communicating device is connected to the NANO board, it can sometimes interrupt communication between the board and the external device. To resolve this, we use SoftwareSerial library that defines virtual or software serial pins.

B. DHT

To process the data from DHT-11 sensor, which is conversion of various output parameters, DHT library is used. This library can covert the temperature readings to Fahrenheit or Celsius based on users' requirements.

C. PulseSensor Playground

Output of a pulse sensor is an analog type data that is in a raw form and needs to be processed before it can determine the pulse rate. This transformation is done using PulseSensor Playground library.

3.3.2 Software flow description

Figure 3.7 shows the software flow of the proposed system. Once the System is turned on, the sensors keep reading the users' vitals and keeps processing the data, on sensing abrupt or abnormal pulses, a trigger signal is sent to the app which start the five second timer for the user in and gives them a chance to cancel the SOS message in case of a false trigger. If the timer goes off and still the user hasn't cancelled the SOS request, the app them autonomously sends a message that contains the GPS locations of the user to the users' friend list.

The different sensors consume power when detect the respective signal, temperature for DHT-11 and user's pulse for pulse sensor. These sensors are transducers, which means, they convert the detected physical quantity to its corresponding digital equivalent. The system is designed in such a way that these sensors detect the readings every 3 second and thus will transmit these readings to the software every 3 seconds.

By various tests under different conditions and scenario, we deduced that 10 seconds were ample in order to cancel the false triggered SOS signal. If the user clicks “CANCEL” within these 10 seconds, no distress message will be sent and will be sent otherwise.

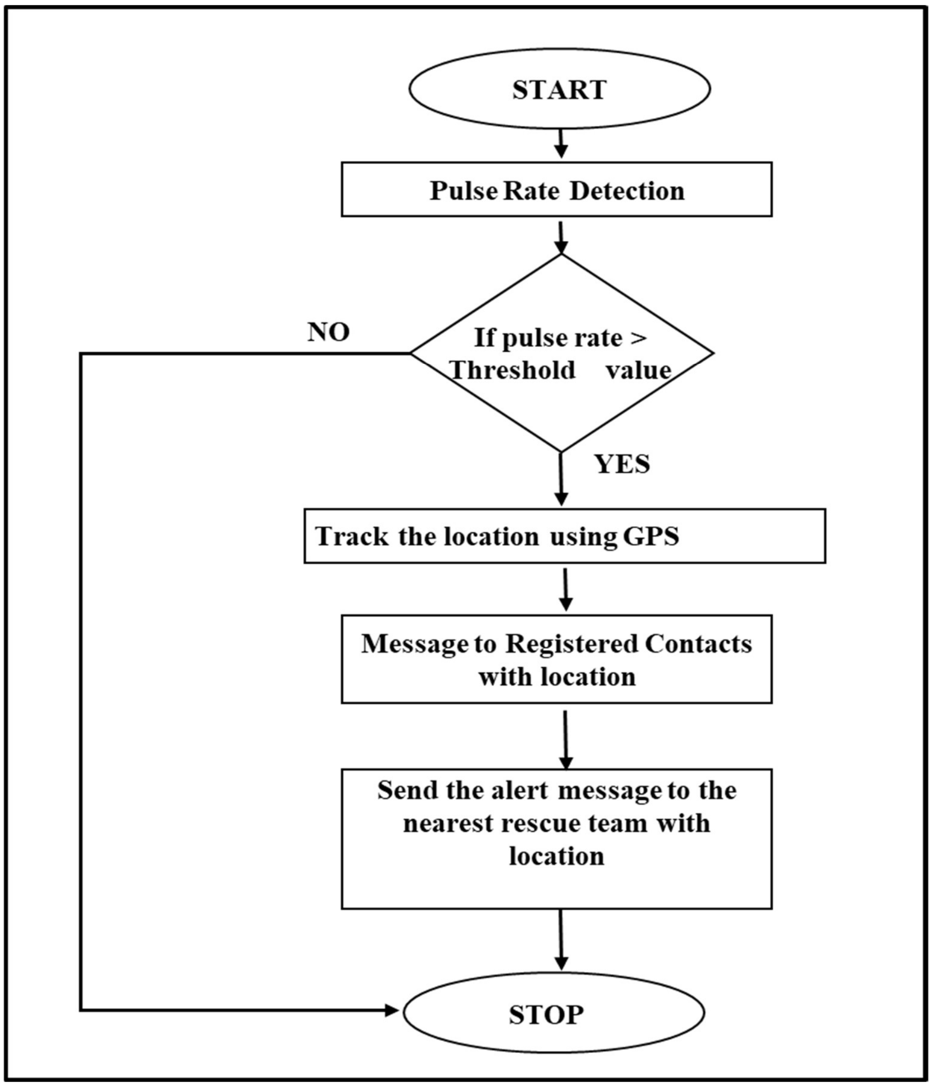


Figure 3.7 Software flow

Chapter 4

Results and Analysis

In this chapter we will analyze the results obtained from our implementation. Discussion of the results is outlined in an orderly fashion with the help of some visual data.

Figure 4.1 shows the implemented hardware connections based around Arduino Nano. The sensors along with the Bluetooth module are powered by the microcontroller itself. Hardware is designed to be modular which means, the maintenance of the system is easy, if either one of the components becomes unresponsive then, it can simply be changed accordingly. Arduino Nano is equipped with 3.3V and 5V outputs that are used by the sensors and the Bluetooth module. The pulse sensor is given an independent power supply because it is sensitive to current change that maybe caused by sharing components.

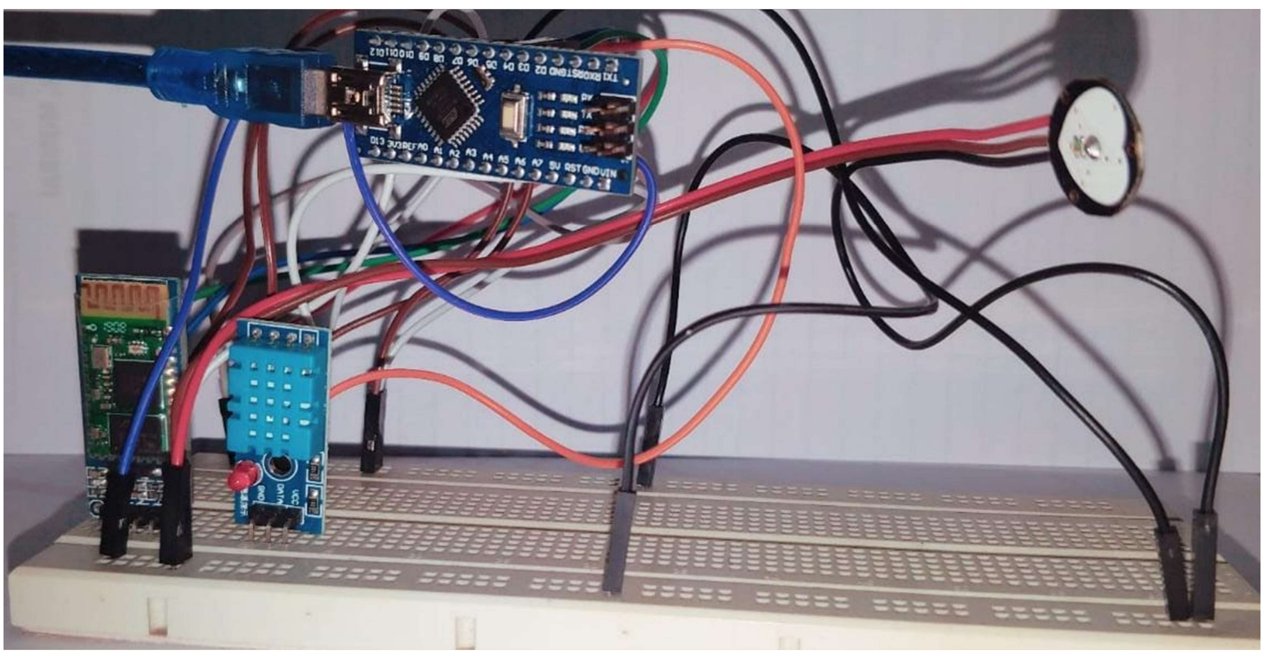


Figure 4.1 (a) System Hardware (Component view)

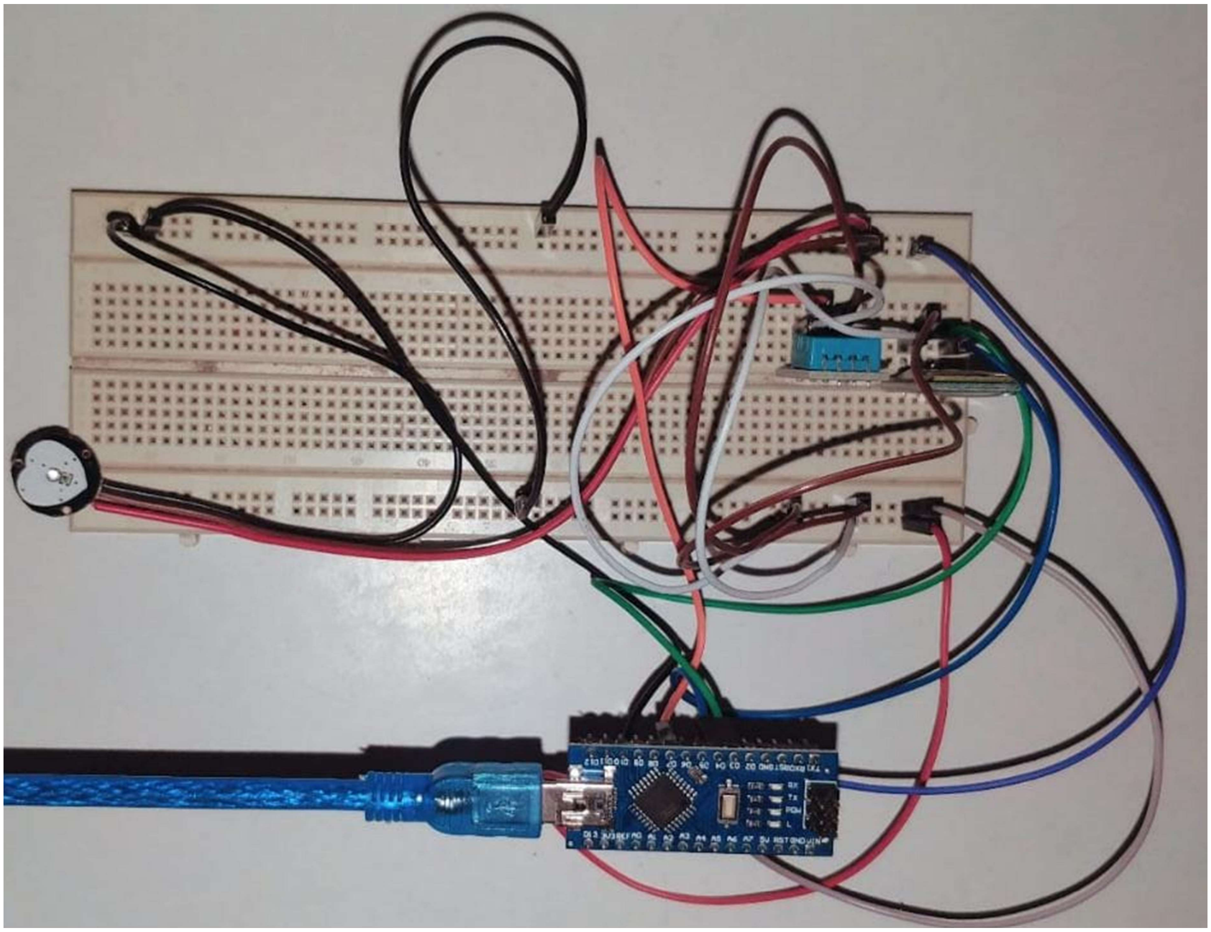


Figure 4.1 (b) System Hardware (Connection view)

As, it can be seen in Figure 4.1 (b) only pulse sensor is given a power of 5V (Connected in the lower horizontal connection lines). Many combinations we tried for the available power outlets and only by isolating the pulse sensor, the readings were accurate which made them reliable. As the output was more stable, the rate of false triggering was under 30%.

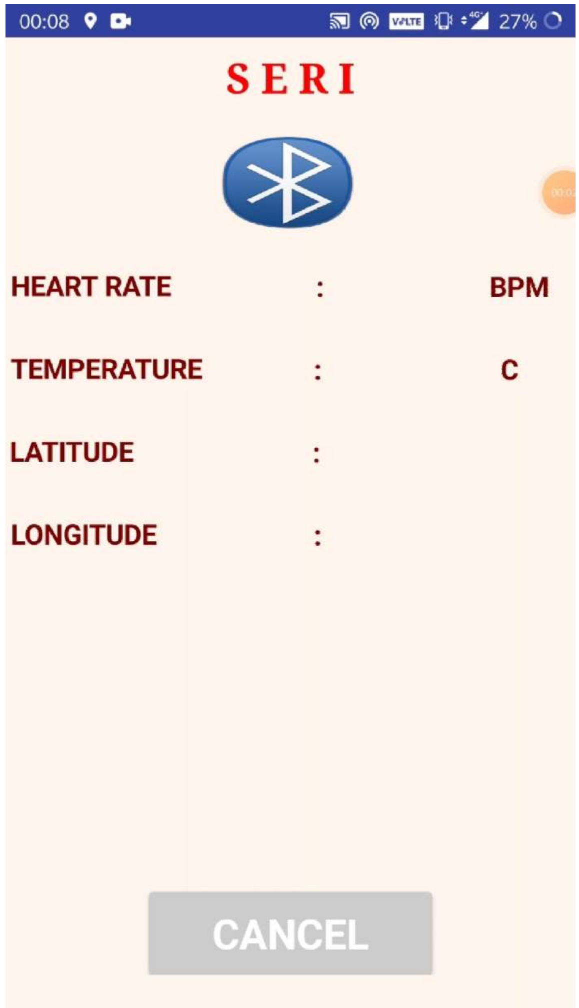


Figure 4.2 SOS SERI landing page

App completes the system and it also binds the entire system together, “SOS SERI” the app developed by us and the hardware itself are linked together via Bluetooth technology. As soon as the App is launched, user is presented with the landing page which is depicted in figure 4.2. Landing Page of the app has four output fields, for Heat rate, temperature and the remaining two for the GPS Co-ordinates i.e., latitude and longitude.

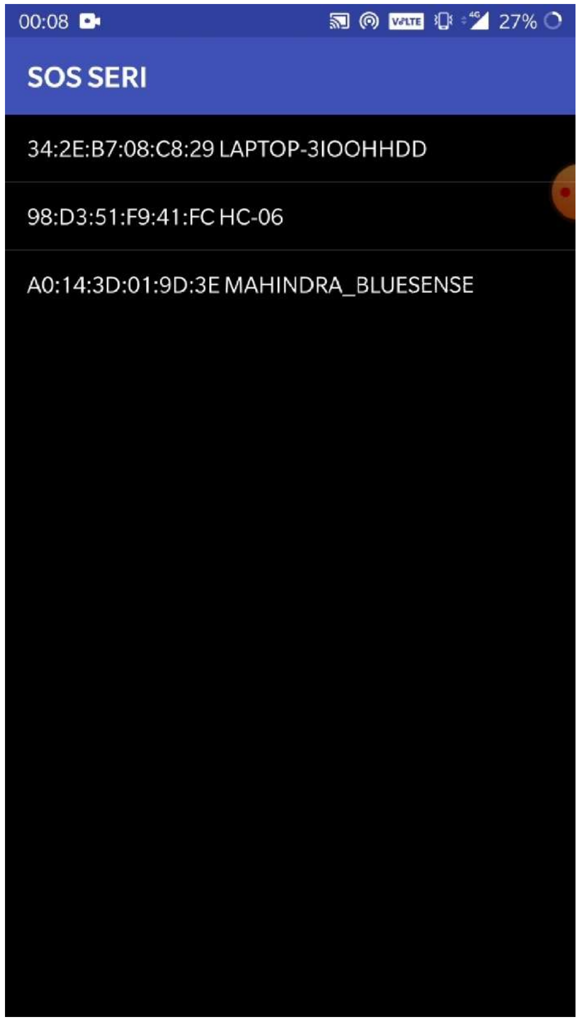


Figure 4.3 Bluetooth List Picker

Until the system is not connected to the "SOS SERI" app, the app is of no use. Thus, the user has to connect the hardware with the app by taping on the Bluetooth logo after which the user has to tap on "HC-06" which is shown in figure 4.3 The Bluetooth logo is nothing but a list picker that allows the app to access the available devices in the phone's proximity. once the communication link is established, the background of the logo header is turns green (refer fig.4.4).

Thus, the user has to connect the hardware with the app by taping on the Bluetooth logo after which the user has to tap on "HC-06" which is shown in figure 3.10, once the communication link is established, the background of the logo header is turns green and if there is no link is established, there will be no change.

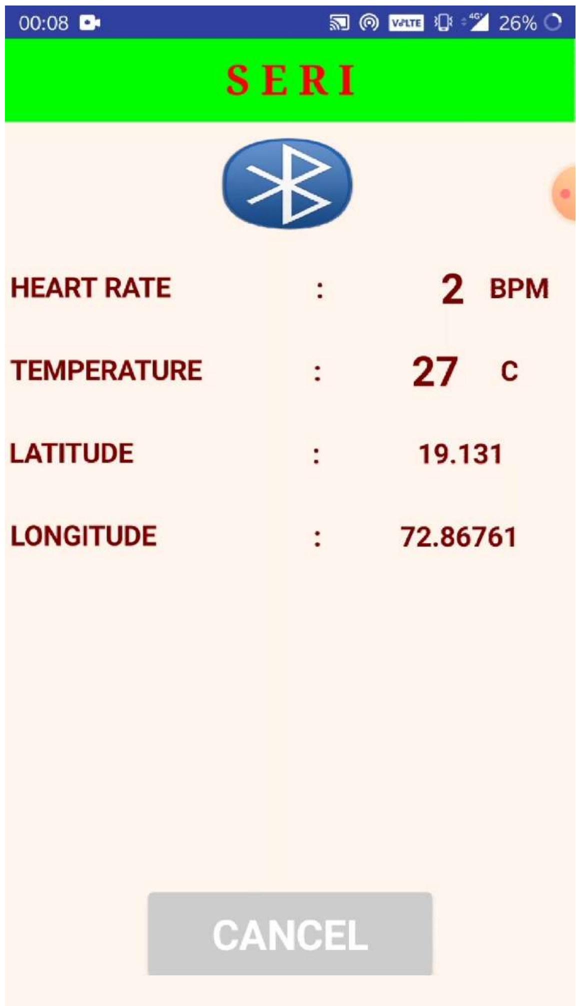


Figure 4.4 Connection acknowledgement

As the two are linked, the readings are sent from the hardware to the app via Bluetooth, App uses the phone's location sensor. This reduces load on the microcontroller and there is no need for large battery to power up a GPS module.

The pulse sensor will keep monitoring user's pulses and if not connected it will echo an output of 220+ BPM. DHT-11 provides user's skin temperature, when there is no surface contact, it reads the environment temperature. However, it is recommended to let the system be active on the user for at least 10 seconds before it is connected to the app. On, reception of a heart rate greater than a threshold, will trigger a SOS request and a 10 seconds timer will be set off as shown in fig. 4.5

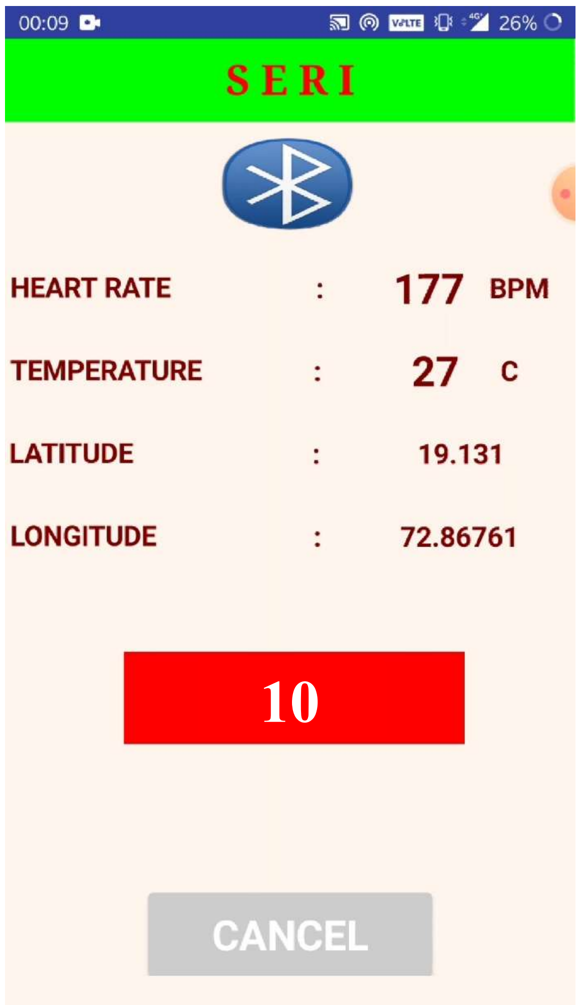


Figure 4.5 SOS triggered page

The system is designed in such a way that, in case of a false trigger, the user has an option of freely cancelling the false SOS trigger. In that case, if the user clicks “CANCEL” then, the app will close and no message is broadcasted, keeping in mind that the user will only have time till the timer goes off. In situations when the trigger is true and the user doesn’t do anything, the app broadcasts an autonomous message to the user’s emergency contact list. Such a case is shown in figure 4.6.

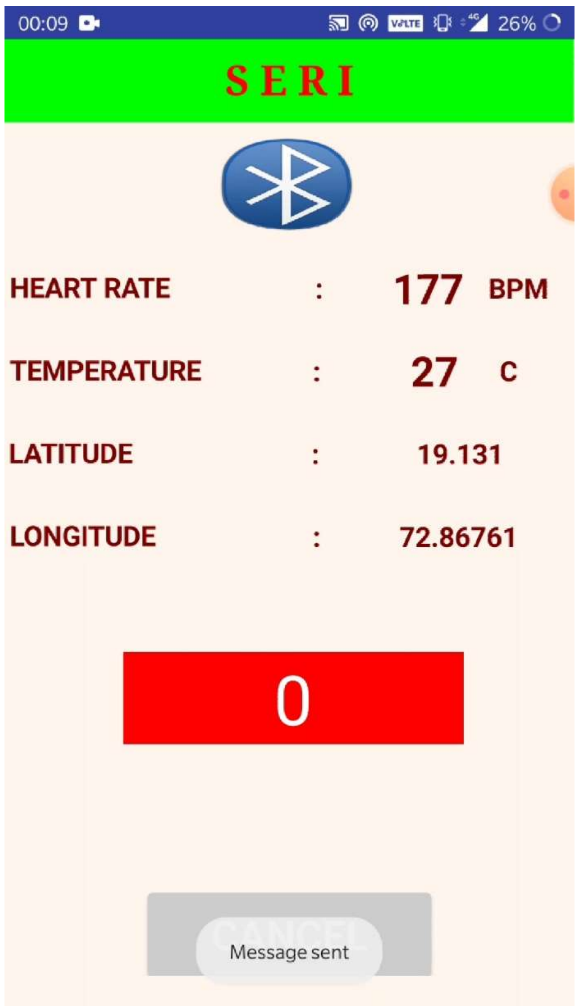


Figure 4.6 Message acknowledgement

Finally, figure 4.7 shows a specimen distress mesasge which was autonomously sent by the app to the user's emergency list. The message is constructed in such a way that, it has locality or street address which is human readable and a GPS location along with it, which gives the last know location of the user. Not one but 5 messages are sent out, this is done to get the attention of the recepiant.

Figure 4.7 Received message

Chapter 5

Advantages, Limitations and Application

This chapter discusses the pros and cons of the designed system along with some of the applications are also suggested later in the chapter:

5.1 Advantages

There are many advantages of the said device which ultimately benefits society and women in particular. This project concerns with the safety of women in general. In this world with such an advanced technological breakthrough, there is still a void for something which will help women to step out of their homes fearlessly. Our project tackles this problem and makes it easier for emergency services to reach the victim effortlessly.

- The proposed system is fully autonomous and can function without any human intervention.
- The sensors used in the band detects any abnormalities in the user's vital signs such as the heart rate and temperature within few seconds. The device then communicates with the application through a Bluetooth module, the latter broadcasting an emergency message containing the location of the user to their emergency contacts and the nearest emergency services.
- As there are no moving parts, the maintenance of the system becomes easier.
- In case of false triggering, user is given an option of cancelling the SOS request also time given for the same is sufficient.
- By using phone's location sensor, the need for incorporating a specific module for location tracking is eliminated, resulting in reduction of size of the system.
- Up to 50 numbers can be added in emergency contact list, which can be altered later

5.2 Limitations

Although the system is designed by considering most of the parameters but there can be some areas of improvement;

- Firstly, the device won't work if there is a battery gets discharged and even though the components don't have a high-power rating, battery drainage will always be an issue.
- Second, the two halves of the system, hardware and the phone itself have to be within a few meters range of each other which is the limit of the Bluetooth technology.

4.3 Applications

With no to very little changes in the source code, system can be employed under various sectors and fields;

- Post-surgery monitoring
- Psychiatric Institutions
- Paralysis Treatment Center
- Cardiorespiratory Fitness

These are some of the suggested areas that can use the proposed system. As it can be seen, the majority of disciplines are from medical sector, sensitivity and accuracy of sensors are very crucial and thus better grade sensors must be employed.

Chapter 6

Conclusion and Future Scope

This chapter summarizes the discussions made earlier about the designed system and work that can make this system evolve to a better system.

6.1 Conclusion

India is a growing economy and women too contribute towards that growth, around 20.3% of total work force of India are Women. If we want that number to go higher then we must do ensure that our cities, states and the entire nation is safe for women. We have to ensure that not only the women of the cities are safe but also the ones in rural areas. The proposed system might not the only solution but it's one of many steps that we need to take in order to make a better and a safer society for everyone especially for women. In rural areas not everyone has smart devices, in that case the hardware can be added with a GSM module and that could be used for communication.

6.2 Future Scope

The project can be further extended to create a whole new system of connected smart health bands so that everyone will be monitored and given proper treatment at right time. With more advanced and reliable sensors, the health band can be more efficient. This project can be a life saying band for people who do not have sufficient time to take care of their health or people who live alone and do not have someone to look after. It can be achieved by adding some extra hardware like camera and sound recorder which can give live updates to the emergency contact whenever the system is in emergency state which can act as an additional safety measure and also help the law enforcement to catch the felon. Now as our project has two parts one which rests on the users wrist that is hardware and other is the app which is installed in the phone, with the advancement of technology and updates in the sensors the two parts can be accommodated in one single unit as a smart watch but with safety measures.

The designed system can be used for autistic children as a means to track their pulse rates and notify his/her parents about significant changes which may indicate that the child is suffering from a stroke or not. It can be used by outpatients so that the doctors can keep an eye on their vitals post-treatment. This reduces the hospital visits done by the patients.

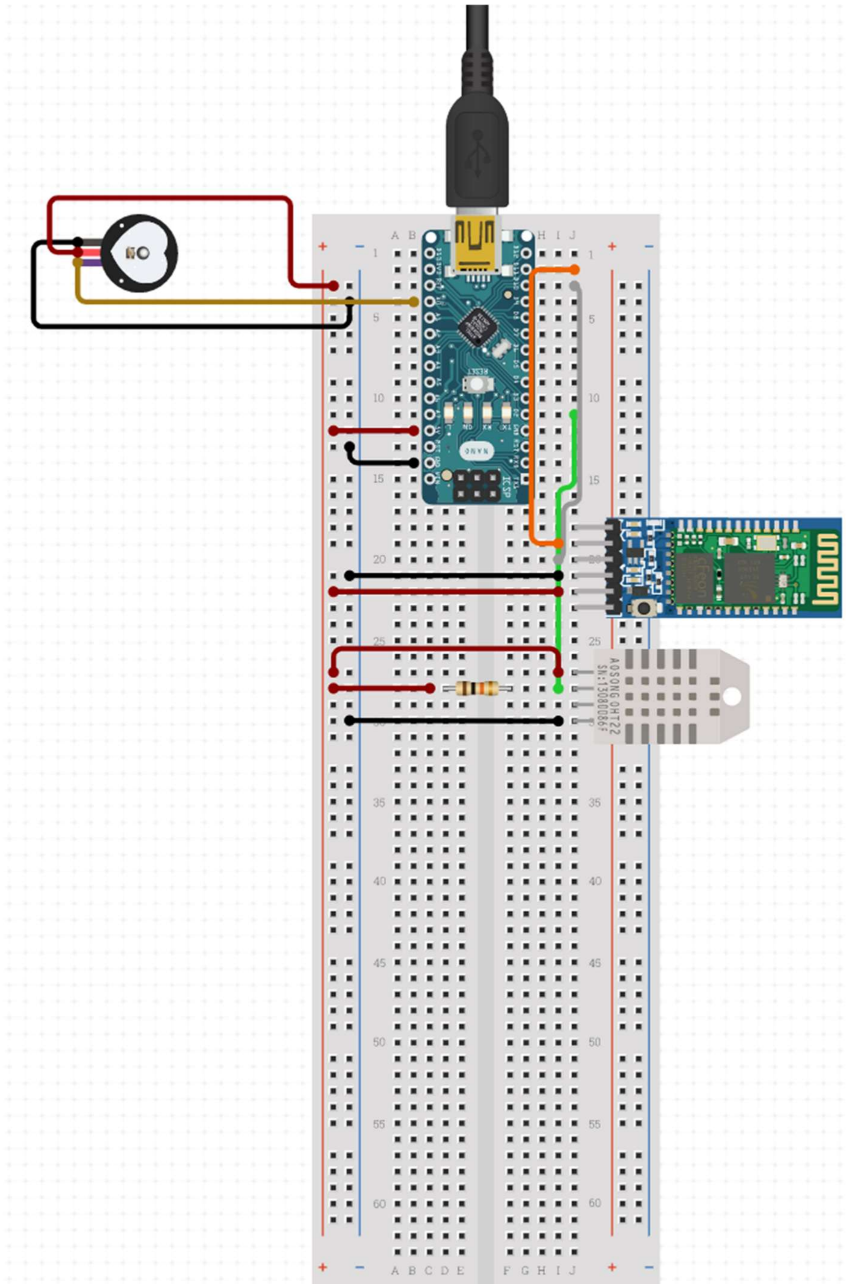
References

1. Pallavi Raj, Saikrishna P, Solly Ann Varghese, Unnikrishnan R, “A SMART BAND FOR WOMEN SAFETY”, International Research Journal of Engineering and Technology (IRJET), Vol: 05, Issue: 04, Apr-2018.
2. Siddharth A. Kokalki, Akshay R. Mali, Pawan A. Mundada, Ritesh H. Sontakke, “Smart Health Band Using IoT,” Department of Computer Science and Engineering, DKTE Society’s Textile and Engineering Institute, Ichalkaranji.
3. V. P. N, S. S, Suhas R C, R. B. S, Raksha Ramakrishna, and Roopa J, “Design and Implementation of Smart Wrist band for Safety Measures in Emergency,” Unpublished, 2013.
4. G. C. Harikiran, K. Menasinkai, and S. Shirol, “Smart security solution for women based on Internet of Things (IOT),” presented at the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Mar. 2016.

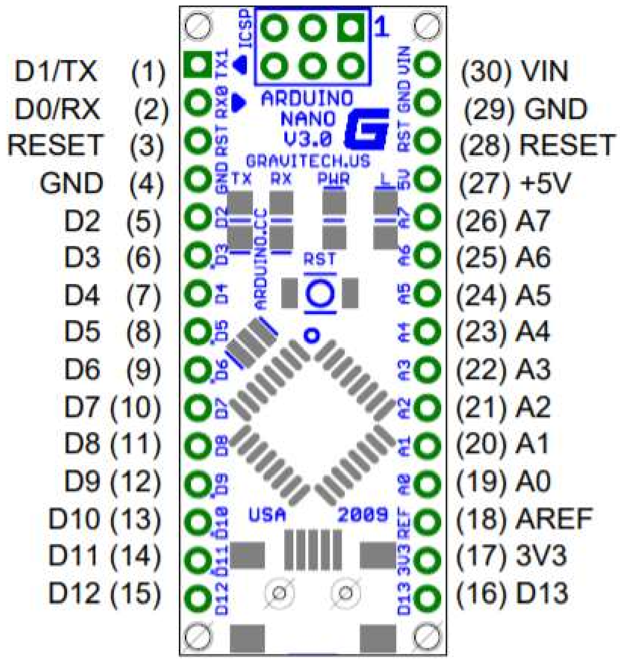
5. P. Saikumar, P. Bharadwaja and J. Jabez, "Android and Bluetooth Low Energy Device Based Safety System," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 1180-1185.
6. Sonisha Soman, Sreelakshmi G, Aswani Asok, Sreedharan Embrandhiri, Sree Narayana Guru Institute of Science and Technology, “Intelligent Multipurpose Safety Wrist Band for Women using Arduino”, IJIIE-International Journal of Innovations & Implementations in Engineering 2017 APRIL Edition Volume 1.
7. Anisha Cotta, Naik Trupti Devidas, Varda Kalidas Naik Ekoskar,”WIRELESS COMMUNICATION USING HC-05 BLUETOOTH MODULE INTERFACED WITH ARDUINO”, International Journal of Science, Engineering and Technology Research (IJSETR) Volume 5, Issue 4, April 2016.
8. P. L. Penmatsa and D. V. R. K. Reddy, "Smart Detection and Transmission of Abnormalities in ECG via Bluetooth," 2016 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, 2016, pp. 41-44.
9. A. Z. M. Tahmidul Kabir, A. M. Mizan and T. Tasneem, "Safety Solution for Women Using Smart Band and CWS App," *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Phuket, Thailand, 2020, pp. 566-569.

10. T. Munasinghe, E. W. Patton and O. Seneviratne, "IoT Application Development Using MIT App Inventor to Collect and Analyze Sensor Data," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 6157-6159.
11. IEEE 802.15.1-2002 - IEEE Standard for Telecommunications and Information Exchange Between Systems.
12. . IEEE 1118.1-1990 - IEEE Standard for Microcontroller System Serial Control Bus.

Appendix I: Circuit diagram



Appendix II: Data Sheet



Arduino Nano Pin Layout

Arduino Nano	Specifications
Microcontroller	ATmega328P
Architecture	AVR
Operating Voltage	5 Volts
Flash Memory	32 KB of which 2 KB used by Bootloader
SRAM	2KB
Clock Speed	16 MHz
Analog I/O Pins	8

Arduino Nano	Specifications
EEPROM	1 KB
DC Current per I/O Pins	40 milliAmps
Input Voltage	(7-12) Volts
Digital I/O Pins	22
PWM Output	6
Power Consumption	19 milliAmps
PCB Size	18 x 45 mm
Weight	7 gms

Arduino Nano Specifications

Arduino Nano Pin	Pin Name	Type	Function
1	D1/TX	I/O	Digital I/O Pin Serial TX Pin
2	D0/RX	I/O	Digital I/O Pin Serial RX Pin
3	RESET	Input	Reset (Active Low)
4	GND	Power	Supply Ground
5	D2	I/O	Digital I/O Pin
6	D3	I/O	Digital I/O Pin
7	D4	I/O	Digital I/O Pin
8	D5	I/O	Digital I/O Pin

Arduino Nano Pin	Pin Name	Type	Function
9	D6	I/O	Digital I/O Pin
10	D7	I/O	Digital I/O Pin
11	D8	I/O	Digital I/O Pin
12	D9	I/O	Digital I/O Pin
13	D10	I/O	Digital I/O Pin
14	D11	I/O	Digital I/O Pin
15	D12	I/O	Digital I/O Pin
16	D13	I/O	Digital I/O Pin
17	3V3	Output	+3.3V Output (from FTDI)
18	AREF	Input	ADC reference
19	A0	Input	Analog Input Channel 0
20	A1	Input	Analog Input Channel 1
21	A2	Input	Analog Input Channel 2
22	A3	Input	Analog Input Channel 3
23	A4	Input	Analog Input Channel 4
24	A5	Input	Analog Input Channel 5
25	A6	Input	Analog Input Channel 6
26	A7	Input	Analog Input Channel 7
27	+5V	Output or Input	+5V Output (From On-board Regulator) or +5V (Input from External Power Supply
28	RESET	Input	Reset (Active Low)
29	GND	Power	Supply Ground
30	VIN	Power	Supply voltage

Arduino Nano Pin Description

ATMEGA328P – Simplified Features	
CPU	8-bit AVR
Number of Pins	28
Operating Voltage (V)	+1.8 V TO +5.5V
Number of programmable I/O lines	23
Communication Interface	Master/Slave SPI Serial Interface(17,18,19 PINS) [Can be used for programming this controller] Programmable Serial USART(2,3 PINS) [Can be used for programming this controller] Two-wire Serial Interface(27,28 PINS)[Can be used to connect peripheral devices like Servos, sensors and memory devices]
ADC Module	6channels, 10-bit resolution ADC
Timer Module	Two 8-bit counters with Separate Prescaler and compare mode, One 16-bit counter with Separate Prescaler,compare mode and capture mode.
Analog Comparators	1(12,13 PINS)
PWM channels	6
External Oscillator	0-4MHz @ 1.8V to 5.5V 0-10MHz @ 2.7V to 5.5V 0-20MHz @ 4.5V to 5.5V
Internal Oscillator	8MHz Calibrated Internal Oscillator
Program Memory or Flash memory	32Kbytes[10000 write/erase cycles]
CPU Speed	1MIPS for 1MHz
RAM	2Kbytes Internal SRAM
EEPROM	1Kbytes EEPROM
Watchdog Timer	Programmable Watchdog Timer with Separate On-chipOscillator
Power Save Modes	Six Modes[Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby]
Operating Temperature	-40°C to +105°C(+105 being absolute maximum, -40 being absolute minimum)

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

DHT-11 Technical Specifications

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

DHT-11 Electrical Specifications

Product Attribute	Attribute Value
Manufacturer:	Olimex Ltd.
Product Category:	Bluetooth Modules (802.15.1)
Class:	Class 2
Interface Type:	Serial
Output Power:	6 dBm
Data Rate:	3 Mb/s
Frequency:	2.4 GHz
Operating Supply Voltage:	3.1 V to 4.2 V
Minimum Operating Temperature:	- 25 C
Maximum Operating Temperature:	+ 75 C
Dimensions:	27 mm x 13 mm x 2 mm
Height:	2 mm
Length:	27 mm
Product:	Bluetooth Modules
Sensitivity:	- 80 dBm
Type:	USB Adapter
Width:	13 mm
Brand:	Olimex Ltd.
Mounting Style:	SMD/SMT

HC-06 Specifications

Absolute Maximum Ratings	Minimum	Maximum	Unit
Operating Temperature Range	-40	+85	Celcius
Input Voltage Range	3	5.5	V
Output Voltage Range	0.3	Vdd	V
Supply Current	3	4	mA

Pulse Sensor Specifications

Appendix III: List of Components

Sensors

- PulseSensor PLSNSR1
- Temperature sensor DHT-11

Microcontroller

- Arduino Nano (ATmega328P)

Communication Device

- Bluetooth module HC-06

Appendix IV: Hardware Source Code

```
#include <SoftwareSerial.h>

#include "DHT.h"

int pulsePin = A0;                 // Pulse Sensor purple wire connected to analog pin A0

int blinkPin = 13;// pin to blink led at each beat

#define DHTPIN 7

#define DHTTYPE DHT11

int RX_pin = 3;

int TX_pin = 4;

DHT dht(DHTPIN, DHTTYPE);

SoftwareSerial BTSerial(RX_pin, TX_pin);

// Volatile Variables, used in the interrupt service routine!
volatile int BPM;                  // int that holds raw Analog in 0. updated every 2mS
volatile int Signal;               // holds the incoming raw data
volatile int IBI = 600;            // int that holds the time interval between beats! Must be
seeded!
volatile boolean Pulse = false;    // "True" when User's live heartbeat is detected. "False"
when not a "live beat".
volatile boolean QS = false;       // becomes true when Arduoino finds a beat.
```

```
static boolean serialVisual = true;   // Set to 'false' by Default.  Re-set to 'true' to see Arduino
Serial Monitor ASCII Visual Pulse


volatile int rate[10];                    // array to hold last ten IBI values

volatile unsigned long sampleCounter = 0;          // used to determine pulse timing

volatile unsigned long lastBeatTime = 0;           // used to find IBI

volatile int P = 512;                      // used to find peak in pulse wave, seeded

volatile int T = 512;                     // used to find trough in pulse wave, seeded

volatile int thresh = 525;                // used to find instant moment of heart beat, seeded

volatile int amp = 100;                   // used to hold amplitude of pulse waveform, seeded

volatile boolean firstBeat = true;        // used to seed rate array so we startup with reasonable
BPM

volatile boolean secondBeat = false;      // used to seed rate array so we startup with
reasonable BPM



void setup()

{


  pinMode(blinkPin,OUTPUT);         // pin that will blink to your heartbeat!

  Serial.begin(9600);             // we agree to talk fast!

  interruptSetup();               // sets up to read Pulse Sensor signal every 2mS

                                  // IF YOU ARE POWERING The Pulse Sensor AT VOLTAGE LESS
THAN THE BOARD VOLTAGE,


  BTSerial.begin(9600);

  dht.begin();                            // UN-COMMENT THE NEXT LINE AND APPLY THAT
VOLTAGE TO THE A-REF PIN

                                  //   analogReference(EXTERNAL);

}
```

```
//  Where the Magic Happens
void loop()
{
  serialOutput();
  float tempC = dht.readTemperature();

  if (QS == true) // A Heartbeat Was Found
    {
      // BPM and IBI have been Determined
      // Quantified Self "QS" true when arduino finds a heartbeat
      serialOutputWhenBeatHappens(); // A Beat Happened, Output that to serial.
      QS = false; // reset the Quantified Self flag for next time
    }

  delay(3000); //  take a break for 3 seconds
}


void interruptSetup()
{
  // Initializes Timer2 to throw an interrupt every 2mS.
  TCCR2A = 0x02;     // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO INTO
CTC MODE
  TCCR2B = 0x06;     // DON'T FORCE COMPARE, 256 PRESCALER
  OCR2A = 0X7C;      // SET THE TOP OF THE COUNT TO 124 FOR 500Hz SAMPLE
RATE
  TIMSK2 = 0x02;     // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND
OCR2A
  sei();             // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
}
```

```
void serialOutput()
{   // Decide How To Output Serial.
 if (serialVisual == true)
  {
    arduinoSerialMonitorVisual('-', Signal);   // goes to function that makes Serial Monitor
Visualizer
  }
 else
  {
     sendDataToSerial('S', Signal);     // goes to sendDataToSerial function
   }
}

void serialOutputWhenBeatHappens()
{


 int flag = 0;
 float tempC = dht.readTemperature();


if (serialVisual == true) //  Code to Make the Serial Monitor Visualizer Work
   {


 if (BPM > 150)


     {
      flag = 1;
      }
    else
     {
      flag = 0;
```

```
    }

   BTSerial.print(int(BPM));
   BTSerial.print("|");
   BTSerial.print(int(tempC));
   BTSerial.print("|");
   BTSerial.println(flag);
   Serial.print(int(BPM));
   Serial.print("|");
   Serial.print(int(tempC));
   Serial.print("|");
   Serial.println(flag);


  }

 else
  {
   sendDataToSerial('B',BPM);   // send heart rate with a 'B' prefix
   sendDataToSerial('Q',IBI);   // send time between beats with a 'Q' prefix

  }

}

void arduinoSerialMonitorVisual(char symbol, int data )
{
 const int sensorMin = 0;      // sensor minimum, discovered through experiment
 const int sensorMax = 1024;    // sensor maximum, discovered through experiment
 int sensorReading = data; // map the sensor range to a range of 12 options:
```

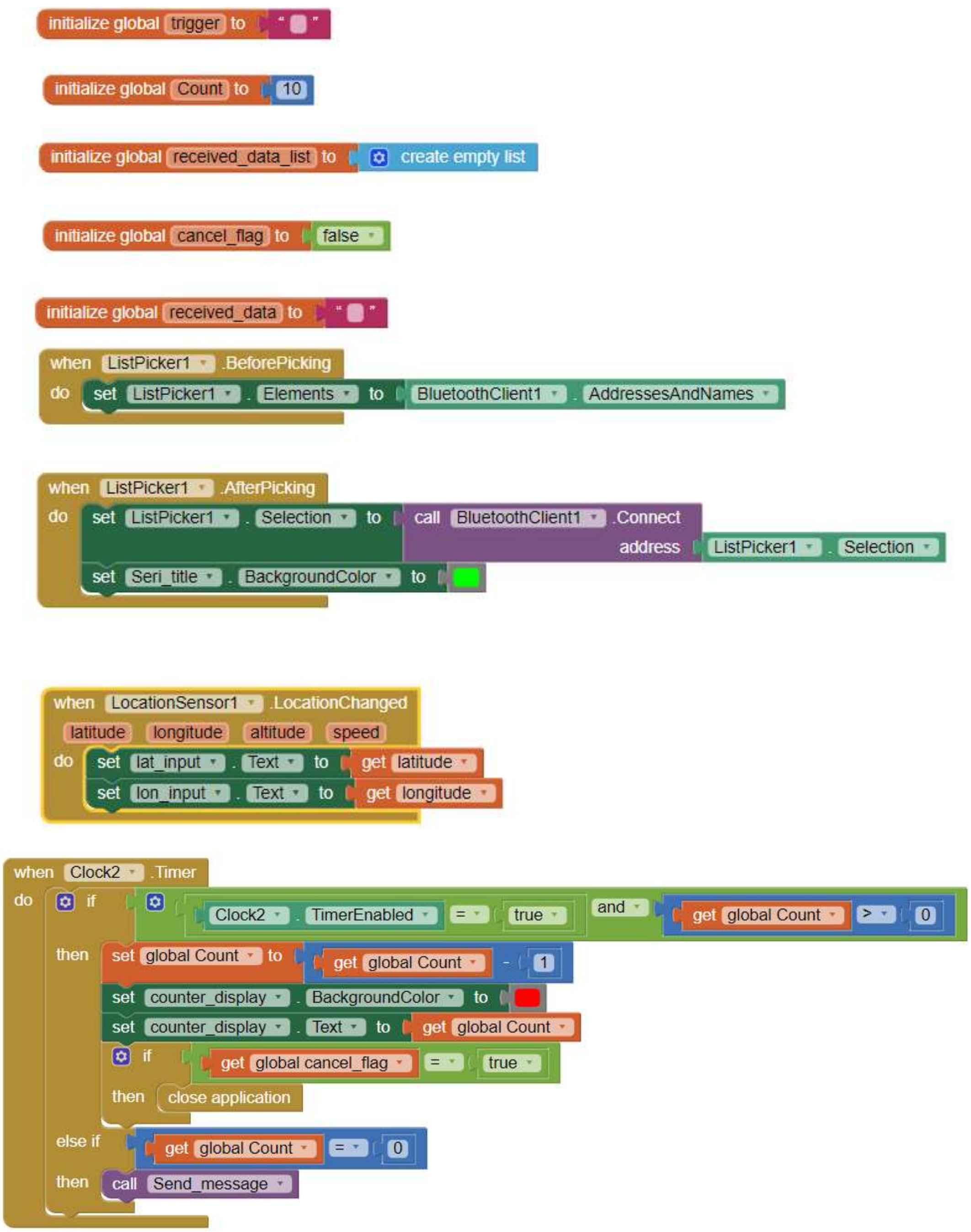
```
  int range = map(sensorReading, sensorMin, sensorMax, 0, 11);
  // do something different depending on the
  // range value:
}


void sendDataToSerial(char symbol, int data )
{
   Serial.print(symbol);
   Serial.println(data);
}

ISR(TIMER2_COMPA_vect) //triggered when Timer2 counts to 124
{
  cli();                                      // disable interrupts while we do this
  Signal = analogRead(pulsePin);              // read the Pulse Sensor
  sampleCounter += 2;                         // keep track of the time in mS with this variable
  int N = sampleCounter - lastBeatTime;       // monitor the time since the last beat to avoid
noise
                                              //  find the peak and trough of the pulse wave
  if(Signal < thresh && N > (IBI/5)*3) // avoid dichrotic noise by waiting 3/5 of last IBI
    {
      if (Signal < T) // T is the trough
      {
        T = Signal; // keep track of lowest point in pulse wave
      }
    }

  if(Signal > thresh && Signal > P)
    {         // thresh condition helps avoid noise
      P = Signal;                             // P is the peak
```

```
  }                                   // keep track of highest point in pulse wave

 //  NOW IT'S TIME TO LOOK FOR THE HEART BEAT
 // signal surges up in value every time there is a pulse
 if (N > 250)
 {                               // avoid high frequency noise
  if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3) )
    {
     Pulse = true;                            // set the Pulse flag when we think there is a pulse
     digitalWrite(blinkPin,HIGH);               // turn on pin 13 LED
     IBI = sampleCounter - lastBeatTime;         // measure time between beats in mS
     lastBeatTime = sampleCounter;              // keep track of time for next pulse

     if(secondBeat)
     {                      // if this is the second beat, if secondBeat == TRUE
      secondBeat = false;                  // clear secondBeat flag
      for(int i=0; i<=9; i++) // seed the running total to get a realisitic BPM at startup
      {
        rate[i] = IBI;
      }
     }

     if(firstBeat) // if it's the first time we found a beat, if firstBeat == TRUE
     {
      firstBeat = false;                  // clear firstBeat flag
      secondBeat = true;                  // set the second beat flag
      sei();                             // enable interrupts again
      return;                             // IBI value is unreliable so discard it
     }
    // keep a running total of the last 10 IBI values
```

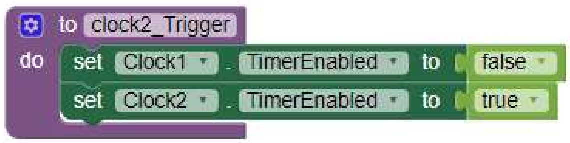
```
    word runningTotal = 0;                // clear the runningTotal variable



  for(int i=0; i<=8; i++)
      {                // shift data in the rate array
        rate[i] = rate[i+1];                  // and drop the oldest IBI value
        runningTotal += rate[i];              // add up the 9 oldest IBI values
      }

    rate[9] = IBI;                          // add the latest IBI to the rate array
    runningTotal += rate[9];                // add the latest IBI to runningTotal
    runningTotal /= 10;                     // average the last 10 IBI values
    BPM = 60000/runningTotal;               // how many beats can fit into a minute? that's
BPM!
    QS = true;                              // set Quantified Self flag
    // QS FLAG IS NOT CLEARED INSIDE THIS ISR
   }
  }

  if (Signal < thresh && Pulse == true)
   {   // when the values are going down, the beat is over
    digitalWrite(blinkPin,LOW);            // turn off pin 13 LED
    Pulse = false;                         // reset the Pulse flag so we can do it again
    amp = P - T;                           // get amplitude of the pulse wave
    thresh = amp/2 + T;                    // set thresh at 50% of the amplitude
    P = thresh;                            // reset these for next time
    T = thresh;
   }

  if (N > 2500)
   {                          // if 2.5 seconds go by without a beat
```

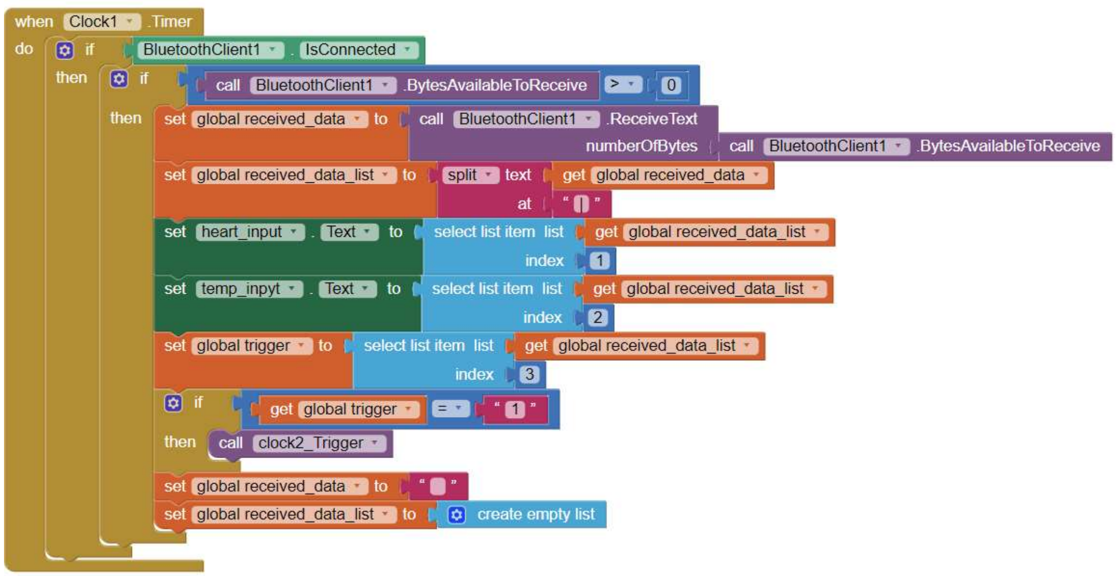
```
    thresh = 512;                        // set thresh default
    P = 512;                             // set P default
    T = 512;                             // set T default
    lastBeatTime = sampleCounter;        // bring the lastBeatTime up to date
    firstBeat = true;                    // set these to avoid noise
    secondBeat = false;                  // when we get the heartbeat back
  }

 sei();                                  // enable interrupts when youre done!
}// end isr
```

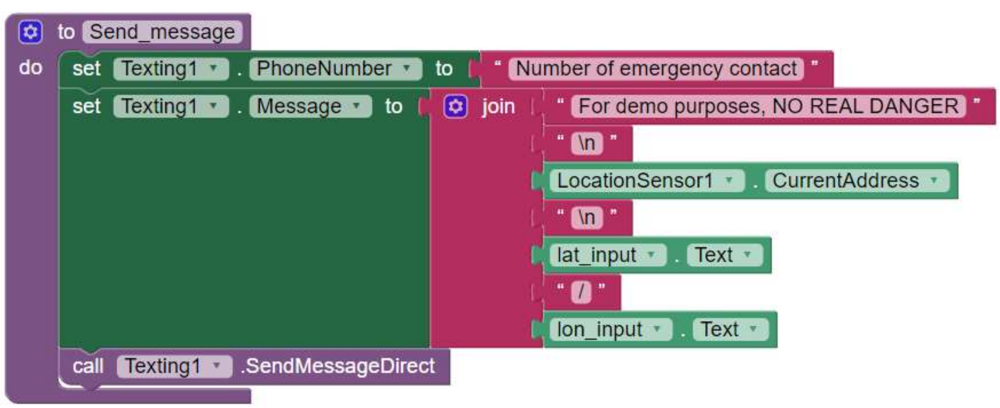
Appendix V: App Source Code




when Cancel .Click
do set global cancel_flag to true
set Cancel . BackgroundColor to


to clock2_Trigger
do set Clock1 . TimerEnabled to false
set Clock2 . TimerEnabled to true


when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then if call BluetoothClient1 .BytesAvailableToReceive > 0
then set global received_data to call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
set global received_data_list to split text get global received_data
at " | "
set heart_input . Text to select list item list get global received_data_list
index 1
set temp_inpyt . Text to select list item list get global received_data_list
index 2
set global trigger to select list item list get global received_data_list
index 3
if get global trigger = " 1 "
then call clock2_Trigger
set global received_data to " "
set global received_data_list to create empty list


to Send_message
do set Texting1 . PhoneNumber to " Number of emergency contact "
set Texting1 . Message to join " For demo purposes, NO REAL DANGER "
" \n "
LocationSensor1 . CurrentAddress
" \n "
lat_input . Text
" / "
lon_input . Text
call Texting1 .SendMessageDirect