

SAFE GUARD

Internship Project Report

Submitted by:

SHUBHAM RAIKWAR (0103IT171106)

VAISHALI HAYARAN (0103IT171121)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

at

LAKSHMI NARAIN COLLEGE OF TECHNOLOGY

KALCHURI NAGAR, RAISEN ROAD, BHOPAL (INDIA) - 462021

SESSION JANUARY MAY 2019

LIST OF FIGURES

1. Fig 3.1: Flowchart of Safe guard	15
2. Fig 3.2: Testing of safe guard	17
3. Fig 4.1: Devise in online mode	19
4. Fig 4.2: Email send to registered person.	19

TABLE OF CONTENTS

Title Page

Table of contents

1. INTRODUCTION	4
1.1 Problem Definition	5
1.2 Project Overview/Specifications	6
1.3 Hardware Specification	7
1.4 Software Specification	8
2. LITERATURE SURVEY	9
2.1 Existing System	10
2.2 Proposed System	11
2.3 Feasibility Study	12
3. SYSTEM ANALYSIS & DESIGN	13
3.1 Requirement Specification	14
3.2 Flowcharts / DFDs / ERDs	15
3.3 Algorithms and Pseudo Code	16
3.4 Testing Process	17
4. RESULTS / OUTPUTS	18
5. CONCLUSIONS / RECOMMENDATIONS	20
6. REFERENCES	22
7. APPENDICES	24
7.1 Details of software/simulator if any	25
7.2 Steps to execute/run/implement the project	25
7.3 Coding if any	25

CHAPTER – 1

INTRODUCTION

PROBLEM DEFINITION

There can be any sudden situation of panic. It could be because of an intruder entering our house or bad health status. Situations can be many for panicking and may vary from person to person. During such emergencies, we might be unable to intimate to the people around us. Sometimes situation arises in the life of elderly folks in which they require sudden help from their known ones. So, we come with a device for their convenience. .Using this device, life of people can be saved .

PROJECT OVERVIEW

Safe guard is an IOT based device which is portable and can comfortably used by the people. They are required to click a button present on the device and the message is sent to the registered person.

A **panic alarm** is an electronic device designed to assist in alerting somebody in emergency situations where a threat to persons or property exists. A panic alarm is frequently but not always controlled by a concealed **panic alarm button**. These buttons can be connected to a monitoring centre or locally via a silent alarm or a button.

This device has a following advantage:

- To save life during sudden stress condition.
- It is compact in size so easy to carry.
- It is user friendly so easy to use.

This device consists of an open source IOT platform named NodeMCU ESP8266 which includes firmware which runs on the ESP8266 Wifi Soc from Espressif Systems, and hardware which is based on the ESP-12 module and cloud services using ThingSpeak.

According to its developers, "**ThingSpeak** is an Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates".

HARDWARE SPECIFICATIONS

- RAM: 512MB or more
- Processor: Intel(R) core (TM)i-5-7200 CPU @2.50GHz 2.71GHz
- Hard Disk: up to 1TB
- NodeMCU ESP8266
- Jumper Wires
- Bread Board
- Led
- Button

SOFTWARE SPECIFICATIONS

- **Operating System:** An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner. An operating system is a software that manages the computer hardware. The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.
- **Thingier.io:** Although the Thingier.io platform is an Open Source platform for the Internet of Things, it provides a ready to use scalable cloud infrastructure for connecting things.
- **Arduino IDE:** The source code for the IDE is released under the GNU General Public License, version 2 The Arduino IDE supports the languages C and C++ using special rules of code structuring The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

CHAPTER – 2
LITERATURE SURVEY

EXISTING SYSTEM

FORBIX SEMICON wireless panic alarm system is a full proof method to secure a building. The alarm system consists of panic buttons and sirens. These panic buttons located at rather, strategic locations inside building premises. Moreover, high volume siren receivers spread across the building. While there is an event of an emergency, a person can alert the security personnel by the mere pressing of a button. The panic siren system also has provision for handheld remote-control units. Therefore, on the press of a remote controller, the building staff gets alerted. Then, respective panic button point shows up on the receiver console. It is possible to have more than one receiver consoles for the staff to know the panic point. Moreover, there is no limitation to the range of communication. And finally, boosters can help to enhance the signals kept at intermediate locations.

Each panic button needs to mount at strategic locations inside a building. However, in a situation of emergency or panic, a person needs to press the nearby alarm button. As a result, the whole building and security will get an alert, with various sirens and alarms. Thus, it gives a sense of security and satisfaction in the mind of residents. Moreover, the system runs on wireless technology. Therefore, it becomes possible to have many panic buttons without wiring. The signal range is 20-30 meters indoors and rather 100-120m outdoors.

PROPOSED SYSTEM

Safe guard is an IOT based device which is portable and can comfortably use by the people. They are required to click a button present on the device and the message is sent to the registered person.

A **panic alarm** is an electronic device designed to assist in alerting somebody in emergency situations where a threat to persons or property exists. A panic alarm is frequently but not always controlled by a concealed **panic alarm button**.

This device consists of an open source IOT platform named NodeMCU ESP8266 which includes firmware which runs on the ESP8266 Wifi Soc from Espressif Systems, and hardware which is based on the ESP-12 module and cloud services using ThingSpeak.

According to its developers, "**ThingSpeak** is an Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates”.

Our safe guard will be a sort of a wrist watch which is portable and comfortable.

Safe guard has the following advantages:

- To save life during sudden stress condition.
- It is compact in size so easy to carry.
- It is user friendly so easy to use.

A button will be present on the wrist watch. Whenever critical situation arises for a person then, he or she can press the button and the notification in the form of call or message will be sent to the registered person on the cloud services.

This is how life of a person can be saved in the emergency conditions.

FEASIBILITY STUDY

A feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable. A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

Technical Feasibility:

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Our projects aim at maintaining technical feasibility. The components or hardware used in safeguard are feasible enough.

Economic Feasibility:

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide. Our aims at maintaining the cost of the device so that it can easily be used by the poor people.

Operational Feasibility:

Safe guard satisfies the requirements identified in the requirements analysis phase of system development. Our project determines whether—and how well—the organization's needs can be met by completing the project.

CHAPTER – 3
SYSTEM ANALYSIS AND DESIGN

REQUIREMENT SPECIFICATION

System requirements are expressed in a software requirement document. The Software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirements definition and the requirement specification. The software requirement document is not a design document. It should set out what the system should do without specifying how it should be done. The requirement set out in this document is complete and consistent.

The software specification document satisfies the following: -

1. It specifies the external system behaviours.
2. It specifies constraints on the implementation.
3. It is easy to change.
4. It serves as reference tool for system maintainers.
5. Its records forethought about the life cycle of the system.
6. It characterizes acceptable response to undesired events.

USER CLASS AND CHARACTERISTICS

There are two types of users :

1. General public: They can see the product and the prices but they cannot buy it.
2. Patients : Those patients who require emergent help from their known ones can buy it and use it.

Various requirements are:

- Portability
- Usability
- Efficient
- Cheap
- Durability

FLOW CHART

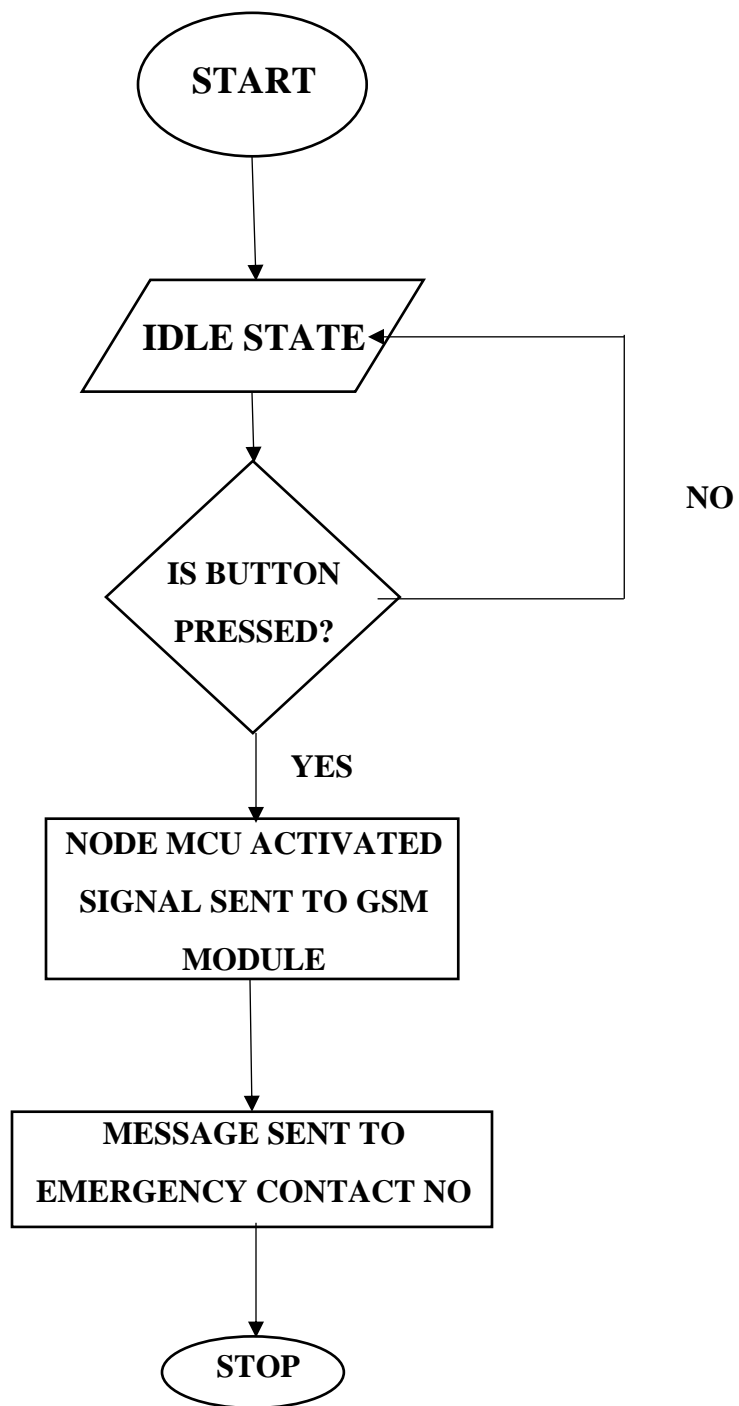


Fig 3.1: Flowchart of Safe guard

PSEUDO CODE

```
#include <SPI.h>
#include <ESP8266WiFi.h>
#include <ThingyESP8266.h>
#define USERNAME "shubhamraikwa6"
#define DEVICE_ID "NM1"
#define DEVICE_CREDENTIAL "oxjXRt2gQ6Xx"
#define SSID "Creator"
#define SSID_PASSWORD "vaishu24"
ThingyESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
int pushPin = D5;
int ledPin = D2;
int val = 0;
void setup() {
    pinMode(ledPin, OUTPUT); // declare LED as output
    pinMode(pushPin, INPUT_PULLUP); // declare pushbutton as input
    Serial.begin(9600);
    thing.add_wifi(SSID, SSID_PASSWORD);
    Serial.println("entering the gates"); }
void loop(){
    val = digitalRead(pushPin);
    Serial.println(val); // read input value
    if (val == LOW) { // check if the input is HIGH (button released)
        digitalWrite(ledPin, HIGH); // turn LED OFF
        delay(1000);
        digitalWrite(ledPin, LOW);
        thing.handle();
        thing.call_endpoint("email");
        delay(4000);
    } else {
        digitalWrite(ledPin, LOW);
    }
}
```

All the header files are included in the code, then the important information is given for the connection with Thingy.io service

Pushpins are given for led button resp. then this void setup() function is used for making connection from net to NodeESP8266 .

Void loop() function is used for sending Email through the thingy.io services

When the input goes to 0 message send to the registered person.

TESTING

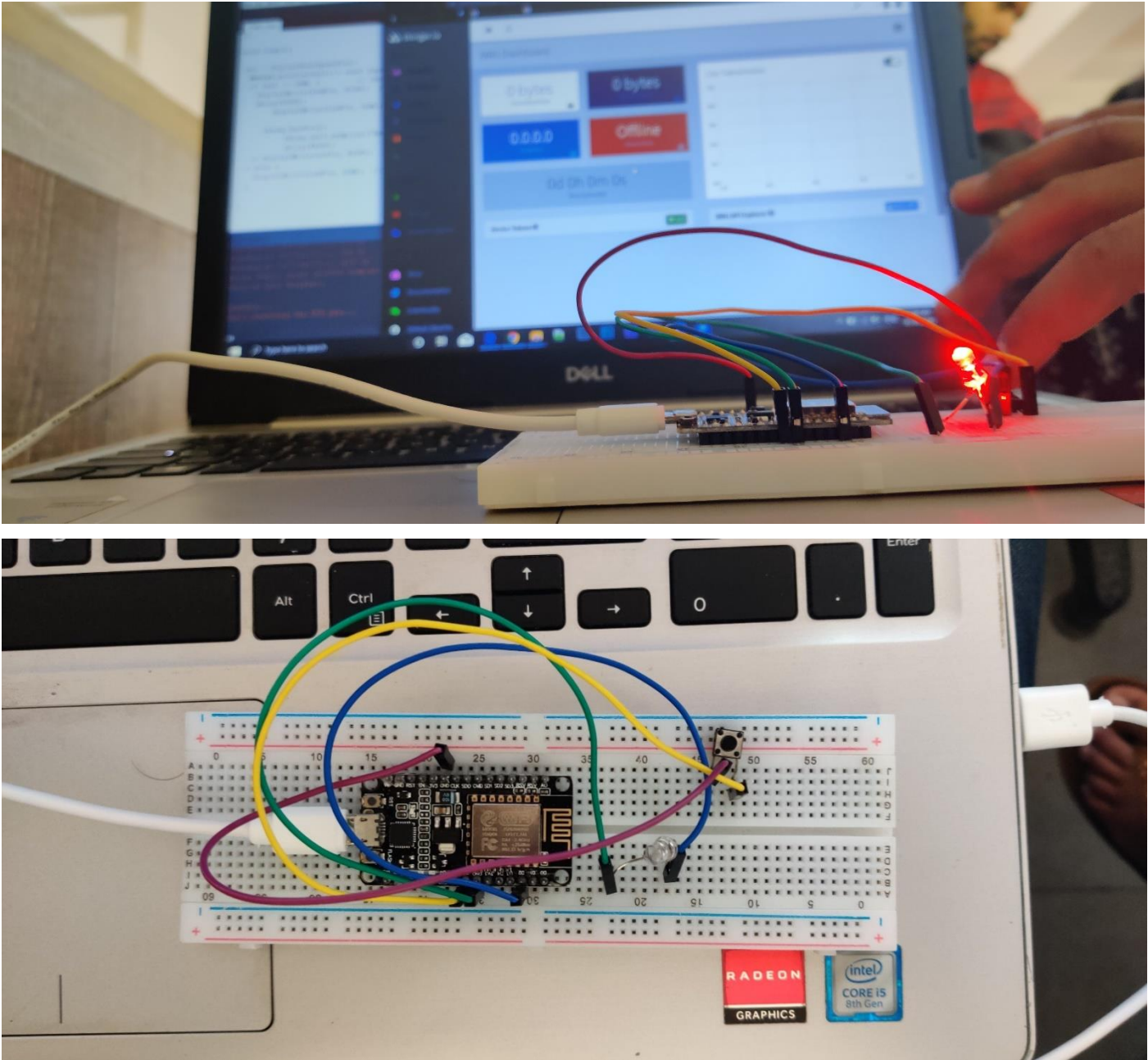


Fig 3.2 Testing of safe guard

Firstly, connections are made for testing in the breadboard. we connected nodeMCU ,a button and led bulb . We connected led bulb and pushpin onto d5 and d2. When nodeMCU gets connected, a light is blinked and hence it is ready to send an email.

When a button is pressed console gives 0 and message is sent to the registered person.

CHAPTER 4

RESULTS / OUTPUTS

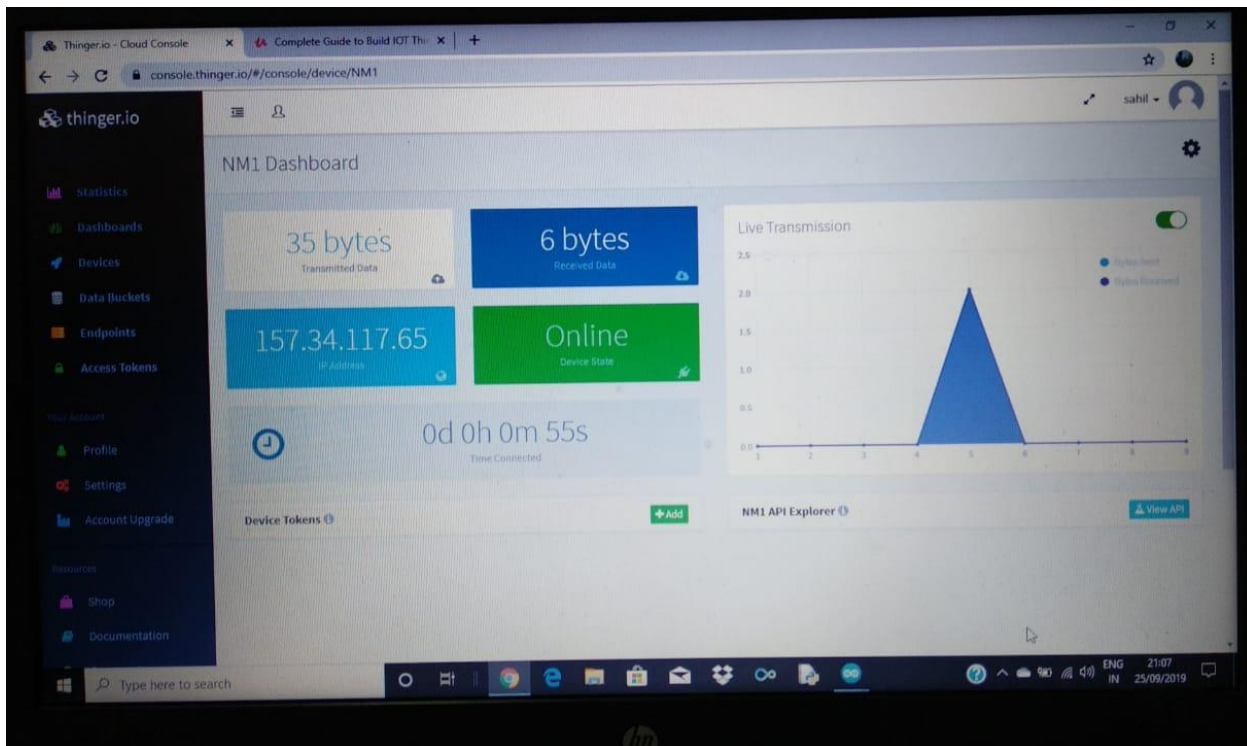


Fig 4.1: Device in online mode

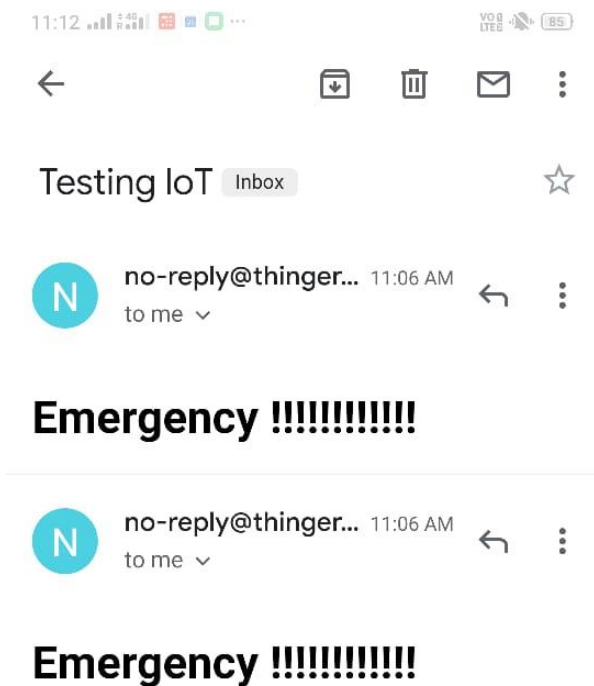


Fig 4.2: Email sent to registered person.

CHAPTER 5

CONCLUSION

CONCLUSION

Sometimes situation arises in the life of elderly folks in which they require sudden help from their known ones. So our device aims at designing a portable, usable device which helps in saving the life. A panic button is an electronic device designed to assist in alerting somebody in emergency situations where a threat to persons or property exists. A panic alarm is frequently but not always controlled by a concealed alarm button. These buttons can be connected to a monitoring centre or locally via a silent alarm or a button. This device is user-friendly and portable.

CHAPTER 6

REFERENCES

REFERENCES

1. Paul Gantt, Ron Gantt, “Disaster Psychology”, Professional Safety, August 2012.
2. Priyanka Shinde, Pranita Taware, Snehal Thorat, Tejashree Waghmare, “Emergency Panic Button (Android App)”, International Journal of Scientific and Engineering Research (ISSN 2229-5518), Volume 3- Issue 3, March 2012.
3. www.udemy.com.
4. International Journal of Engineering and Techniques - Volume 4, Issue 3, May - June 2018 ISSN: 2395-1303
5. <http://www.ijetjournal.org>
6. Huang Huang, Shide Xiao, Xiangyin Meng, Ying Xiong (April 24 - 25, 2010) “A remote home security system based on wireless sensor network and GSM technology.
7. J.D. Jara, L. Caldas-Calle, E. Barbecho, J. Bravo-Torres, J.P. Bermeo, P. Gallegos (July 2017) “Development and Design of the Panic Button System for Community Security in Rural Areas of Pucará-Ecuador” Jour of Adv Research in Dynamical & Control Systems. Special Issue.
8. Choudhury B., Choudhury T. S, Pramanik A, Arif W, Mehedi J, “Design and implementation of an SMS based home security system”, IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), pp. 1-7, 2015

CHAPTER 6

APPENDICES

APPENDICES

Softwares Used :

ARDUINO 1.8.1

The open-source **Arduino** Software (IDE) makes it **easy to write code** and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in **Java** and based on Processing and other open-source software. This software can be used with any **Arduino** board.

NODEMCUESP8266

NodeMCU is an open-source firmware and development kit that helps you to prototype or build IoT product. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266.

Steps to implement a project

- Connect NodeMCU in the breadboard with jumper wire and led bulb.
- Provide a button in the testing circuit.
- Whenever a button is pressed , the device gets connected to the thinger service and email is sent to the registered person.