**ROBOTIC AUTOMATED EXTERNAL DEFIBRILLATOR**

#### A PROJECT REPORT

***Submitted by***

## DINESH.K.R [REGISTER NO:211417104054] FAZIL.M [REGISTER NO:211417104063]

**HAZARATH SHRAVAN.R [REGISTER NO:211417104084]**

***in partial fulfillment for the award of the degree of***

## BACHELOR OF ENGINEERING

**IN**

#### COMPUTER SCIENCE AND ENGINEERING

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**ANNA UNIVERSITY: CHENNAI 600 025**

**APRIL 2021**

**BONAFIDE CERTIFICATE**

Certified that this project report **“ROBOTIC AUTOMATED EXTERNAL DEFIBRILLATOR”** is the bonafide work of **“DINESH.K.R (2017PECCS384), HAZARATH SHRAVAN.R (2017PECCS401), FAZIL.M (2017PECCS390)”**

who carried out the project work under my supervision.

#### SIGNATURE SIGNATURE

**Dr.S.MURUGAVALLI,M.E.,Ph.D., Mr.C.THYAGARAJAN, M.E., (Ph.D.) HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR,** DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on

#### INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors

**Tmt.C.VIJAYARAJESWARI**, **Dr. C.SAKTHIKUMAR, M.E., Ph.D.** and

**Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S. MURUGAVALLI , M.E., Ph.D.,**

for the support extended throughout the project.

We would like to thank my **Project Guide Mr.C.THYAGARAJAN, M.E.,(Ph.D.)** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

**ABSTRACT**

India is one of the most populous countries of the world. Due to over population, ignorance of health have been remained the major problems in India. For every one minute a death sw oops in because of heart attack.

Ambulance service plays a vital role in saving lives. Its primary purpose is to give first aid to the sick or injured people in the emergency scene. To save a life is auspicious as well as precious.

The idea here is to provide an intelligent smart health system using some sensors and microcontrollers; it will sense the body condition and send the data to the collaborated hospital’s database.

This proposed idea gives us the development of a wireless based system for pulse rate, blood pressure and temperature monitoring to be used in ambulance. By this, the real time information can be passed to nearby hospitals to alert them about the critical conditions over IOT.

This hardware device is fixed inside the ambulance to sense the patient’s health, collect the data in a wireless device called node MCU and immediately pass the database to the hospital’s server by the concept of IOT. This may intimate the hospital officials and may respond to the necessary actions to be taken to the person in emergency.

|  |
| --- |
| **LIST OF FIGURES** |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
| **4** | **SYSTEM DESIGN** |  |
|  | 4.1 ER-Diagram | 19 |
|  | 4.2 Dataflow Diagram | 20 |
|  | 4.3.1 List Of UML Diagram Types | 24 |
|  | 4.11 Use Case Diagram | 30 |
|  | 4.12 Activity Diagram | 32 |
|  | 4.13 Sequence Diagram | 33 |
|  | 4.12 Collaboration Diagram | 34 |
|  |  |  |
| **5** | **SYSTEM ARCHITECTURE** |  |
|  | 5.1 Architecture | 36 |
|  | 5.2.1.1 Temperature Sensor | 38 |
|  | 5.2.1.2 Pulse Sensor | 38 |
|  | 5.2.1.3 Heart Beat Sensor | 39 |
|  | 5.2.2.1 Arduino | 40 |
|  | 5.2.2.2 IOT Board | 41 |

|  |
| --- |
| **TABLE OF CONTENTS** |

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | **iv** |
| **1** | **INTRODUCTION** | **1** |
|  | * 1. Overview | 2 |
|  | * 1. Scope of the project | 3 |
| **2** | **LITERATURE SURVEY** | **4** |
| **3** | **SYSTEM ANALYSIS** | **10** |
|  | 3.1 Existing System | 11 |
|  | 3.1.1 Problem Definition | 11 |
|  | 3.2 Proposed System | 12 |
|  | 3.2.1 Advantages | 12 |
|  | 3.3 Requirement Analysis and Specification | 13 |
|  | 3.3.1 Introduction | 13 |
|  | 3.4 Hardware and Software Specification | 13 |
|  | 3.4.1 Hardware Requirements | 13 |
|  | 3.4.2 Software Requirements | 14 |
|  | 3.4.2.1 Java | 14 |

|  |  |  |
| --- | --- | --- |
|  | 3.4.2.1.1 Introduction To Java | 14 |
|  | 3.4.2.1.2 The Java Platform | 15 |
|  | 3.4.2.2 PHP | 15 |
|  | 3.4.2.2.1 Components Of PHP | 16 |
|  | 3.4.2.3 MySql | 16 |
|  | 3.2.2.4 Embedded C | 17 |
| **4** | **SYSTEM DESIGN** | **18** |
|  | 4.1 ER Diagram | 19 |
|  | 4.2 Data Flow Diagram | 20 |
|  | 4.3 UML Diagram | 22 |
|  | 4.3.1 List Of UML Diagram Types | 24 |
|  | 4.4 Structural Diagram | 25 |
|  | 4.5 Class Diagram | 25 |
|  | 4.6 Component Diagram | 26 |
|  | 4.7 Deployement Diagram | 26 |
|  | 4.8 Package Diagram | 26 |
|  | 4.9 Composite Structure Diagram | 27 |
|  | 4.10 Behavioral Diagram | 28 |
|  | 4.11 Use Case Diagram | 29 |
|  | 4.12 Activity Diagram | 31 |
|  | 4.13 Sequence Diagram | 32 |
|  | 4.14 Collaboration Diagram | 34 |
| **5.** | **SYSTEM ARCHITECTUE** | **35** |
|  | 5.1 Architecture | 36 |
|  | 5.2 System Module | 37 |
|  | 5.2.1 Data Collection From Patient’s By Using Sensor | 37 |
|  | 5.2.1.1 Temperature Sensor | 37 |
|  | 5.2.1.2 Pulse Sensor | 38 |
|  | 5.2.1.3 Heart Beat Sensor | 39 |
|  | 5.2.2 Transmit Data From Arduino to IOT board by use of UART Protocol | 39 |
|  | 5.2.2.1 Arduino | 40 |
|  | 5.2.2.2 IOT Board | 41 |
|  | 5.2.3 Transmit data from IOT board to server by use of MQQT protocol | 42 |
| **6.** | **TESTING** | **43** |
|  | 6.1 Testing Process | 44 |
|  | 6.1.1 Types Of Testing | 45 |
|  | 6.2 White Box and Black box Testing | 45 |
|  | 6.2.2 White box Testing | 45 |

|  |  |  |
| --- | --- | --- |
|  | 6.2.2 Black box Testing | 46 |
|  | 6.3 Software testing Strategies | 47 |
| **7.** | **CONCLUSION AND FUTURE ENHANCEMENT** | **48** |
|  | 7.1 Conclusion | 49 |
|  | 7.2 Future Enhancement | 49 |
|  | **SAMPLE CODE** | **50** |
|  | **REFERENCE** | **63** |

**CHAPTER 1 INTRODUCTION**

**1.1 OVERVIEW**

The traffic condition in India effect emergency services like Ambulance and Fire engine. Time is an important factor in case of emergencies; hence we are proposing a system which makes the patient gets right time treatment in ambulance.

According to research conducted by global consultancy firm, traffic in peak hours in major four cities - Delhi, Bengaluru, Chennai, Kolkata costs the economy Rs 1.47 lakh crore per annum.

Kolkata is worst among them followed by Bengaluru. In Bengaluru’s case, rail-based projects like Namma Metro or sub-urban rail, while effective, the capacity will not be sufficient to bring down congestion levels by the time they are fully operational.

The main reason behind this is increasing population which leads to increased number of vehicles, due to which emergency service like Ambulance get affected.

Because of this delay in ambulance service, patient may lose his life and number of these scenarios are increasing day by day.

When emergency vehicles are stuck in heavy traffic and cars are unable to pull over. In that time, the patients gets into dangerous stage. In ambulance the care taker have no idea treat the patient in sever stage it makes patient die or dangerous stage.This paper proposes a solution to make such services easily available to those in need.

## 1.2 SCOPE OF THE PROJECT

* If the condition is critical the information about the patient will be sent to the hospital monitoring database.
* It helps the doctors to prepare medical instruments in hospital if any emergency.
* To save a life is auspicious as well as precious.
* The idea here is to provide an intelligent smart health system using some sensors and microcontrollers; it will sense the body condition and send the data to the collaborated hospital’s database.
* Patient will get correct treatment at a time.

# CHAPTER 2 LITERATURE SURVEY

## Reference 1:

**Title:**

Development of Monitoring and Health Service Information System to Support Smart Health on Android Platform(2018)

## Author:

A’isya Nur Aulia Yusuf ; Fitri Yuli Zulkifli ; I Wayan Mustika

## Description:

This research proposes a design of health monitoring system named Mooble ,a system to monitor patient health condition and to prevent diseases as early as possible. Mooble consists of three subsystems: web application, database and API design, and mobile application on android platform. But this research will only focus on the design and development of the mobile application subsystem. This research consists of three main aspects: application design, development and testing. The system is developed using Rational Unified Process (RUP) framework. In the end, this research will result a mobile application to be used by patients.

## Drawback:

* + They used more code on Java than Objective-C.
  + The main areas of disadvantage in the Rational Unified Process software development cycle include its complexity, the disorganized development and applicability only to large software development projects.

## Reference 2:

**Title:**

An Android Application for Geolocation Based Health Monitoring, Consultancy and Alarm System (2018)

## Author:

Emre oner tartan, Cebrail ciflikli

## Description:

In this study they benefit the facilities provided by mobile technology and propose a geolocation-based heart rate monitoring system. The developed mobile application can send alarm message through notification, sms, mail and allows messaging with the health expert for consultancy. Hence if anomalies are observed in heart rate variability during the outdoor activities, emergency information can be delivered in the shortest time and the delays which have crucial affects can be prevented. The same framework can be extended to a more general system including different sensors for monitoring various physiological signals.

## Drawback:

* + There are many technical and non technical weaknesses to the way geolocation works that make this information unreliable at best.
  + A participant might be a local resident who is out of town for the day or uses a VPN to shield privacy, this should not detract from the value of their input.

## Reference 3:

**Title:**

Health Informatics-Personal health device communication- Device specialization-Medication monitor(2019)

## Author:

IEEE Std 11073-10472-2010

## Description:

Within the context of the ISO/IEEE 11073 family of standards for device communication, this standard establishes a normative definition of the communication between medication monitoring devices and managers (e.g., cell phones, personal computers, personal health appliances, set top boxes) in a manner that enables plug-and-play interoperability. It leverages appropriate portions of existing standards including ISO/IEEE 11073 terminology, information models and application profile. It specifies the use of specific term codes, formats, and behaviors in telehealth environments restricting ambiguity in base frameworks in favor of interoperability.

## Drawback:

* + They used more code on Java than Objective-C.
  + It is difficult to maintain.

## Reference 4:

**Title:**

Remote health, activity, and asset monitoring with wireless sensor networks(2017)

## Author:

M. Marzencki ; P. Lin T. Cho J. Guo B. Ngai , B. Kaminska

## Description:

They propose a system that employs a wireless mesh sensor network to provide the communication backbone for stationary and wearable sensors. The sensor network is interfaced with a PC application through a TCP/IP connection, which allows for remote control along with data visualization and storage. The proposed system is reliable, inexpensive, rapidly deployable by minimally qualified personnel, automatically reconfigurable, and completely autonomous. It provides simultaneous monitoring of environmental and personal health and activity data and the capability of combining both for improved situation assessment. They discuss the proposed architecture and present an example system built to demonstrate the efficacy of this concept.

## Drawback:

* Less accuracy rate
* Low performance

## Reference 5:

**Title:**

Monitoring Health Care System Using Internet of Things - An Immaculate Pairing(2017)

## Author:

Veena Tripathi , Faizan Shakeel

## Description:

The concept of connected health care system and smart medical devices bears enormous potential not just for companies, but also for the well-being of people in general. Hospitalized patients whose physiological status requires close attention can be constantly monitored using IoT-driven monitoring. This type of solution employs sensors to collect comprehensive physiological information and uses gateways and the cloud to analyze and store the information and then send the analyzed data wirelessly for further analysis and review. It replaces the process of having a health professional come by at regular intervals to check the patient's vital signs, instead providing a continuous automated flow of information. The main aim of this work is to give a comprehensive overview of this area of research and sensors used in health monitoring device, how the wearable health monitoring devices works, capture the data and generate report based on different parameters.

## Drawback:

* Implementing computers and the internet for ICT replace the convention education curriculums.
* Managing courses online is difficult.

# CHAPTER 3 SYSTEM ANALYSIS

## 3.1 EXISTING SYSTEM

* Ambulance drivers are at increased risk of road accidents due to high driving speeds under emergency conditions.
* Inside an ambulance basic but important items such as blood pressure gauges, stethoscopes, thermometers, medical tapes, flashlights and blankets, to ensure the paramedics are prepared for almost any situation.
* But in sever condition we need doctor’s advice. When emergency vehicles are stuck in heavy traffic and cars are unable to pull over.
* In that time, the patients gets into dangerous stage. In ambulance the care taker have no idea treat the patient in sever stage it makes patient die or dangerous stage.

## 3.1.1 Problem Definition

* + Not receiving treatment have the right to refuse treatment or transfer to a medical facility via ambulance.
  + When emergency vehicles are stuck in heavy traffic and cars are unable to pull over.
  + 10 miles an hour over the posted speed limit and that is regardless of whether we are driving lights and sirens, traffic…etc. It makes accident, traffic etc

## 3.2 PROPOSED SYSTEM

* Developments in wireless sensors, communication and information network technologies have created a new era of the Internet of things (IoT).
* Our proposed system is to design a device in which the heart beat sensor will sense the heart beat and temperature sensor will sense the body temperature and pulse sensor will sense pulse rate.
* After sensing, sensors will send respective data to the microcontroller. After that microcontroller will sent it to server.
* In the collaborated hospital the patient’s heart rate, pulse and body temperature data will send through server.
* In hospital the respective doctor will continuous monitor patient’s health system. If that patient will be in critical condition then t the doctor will suggest the treatment to that patient in ambulance.
* In this project, the sensor in the ambulance will collect the patient’s information (temp, pulse, heartbeat) by using three Sensors.
* The patient’s information will be passed through the server. These information will be viewed by the doctor from server and suggest the treatment to the patients in the ambulance.

## 3.2.1 ADVANTAGES

* + IoT has applications in healthcare that benefit patients, families, physicians, hospitals and insurance companies.
  + They can track patients' adherence to treatment plans or any need for immediate medical attention.
  + IoT enables healthcare professionals to be more watchful and connect with the patients proactively.

## 3.3 REQUIREMENT ANALYSIS AND SPECIFICATION

**3.3.1 INTRODUCTION:**

* Traffic signals in India has a fixed time period to switch the signals. No changes for emergency vehicles.
* This makes the patient in ambulance to sever stage. To deal with this problem we designed this system.
* In this paper, we propose an IOT-based system for patients with the risk of heart attack and uneven body temperature, high pulse rate.
* If the condition is critical an alert notification will be sent to the hospital monitoring database.
* This system consists of various sensors which collects the patient's information and transmit those information to the server via IOT board.

## 3.4 HARDWARE AND SOFTWARE SPECIFICATION

**3.4.1 Hardware Requirements:**

* Microcontroller
* Wireless transmitter-node mcu
* Wireless receiver
* Heart rate sensor
* Pulse rate sensor
* Temperature sensor

## 3.4.2 Software Requirements:

* Arduino ide
* Mysql
* Embedded c
* Php
* java

**3.4.2.1 JAVA**

* Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems.
* The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

## 3.4.2.1.1 INTRODUCTION TO JAVA

* + Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project.
  + The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry.
  + The language that the team created was originally called Oak.
  + The first implementation of Oak was in a PDA-type device called Star Seven (\*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware.
  + The name \*7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other

phone in the office.

## 3.4.2.1.2 THE JAVA PLATFORM

* + A platform is the hardware or software environment in which a program runs.
  + The Java platform differs from most other platforms in that it’s a software-only platform that runs on top of other, hardware-based platforms.
  + Most other platforms are described as a combination of hardware and operating system.
  + The Java platform has two components :
    - The Java Virtual Machine (JVM)
    - The Java Application Programming Interface (Java API)
  + You’ve already been introduced to the JVM. It’s the base for the Java platform and is ported onto various hardware-based platforms.
  + The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets.
  + The Java API is grouped into libraries (packages) of related components.

## 3.4.2.2 PHP

* + PHP is a server side scripting language that is embedded in HTML.
  + It is used to manage dynamic content, databases, session tracking, even build entire ecommerce sites.
  + It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
  + PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side.
  + The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
  + PHP supports a large number of major protocols such as POP3, IMAP, and LDAP.
  + PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.

## 3.4.2.2.1 COMMON USES OF PHP

* + PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
  + PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
  + You add, delete, modify elements within your database through PHP.
  + Access cookies variables and set cookies.
  + Using PHP, you can restrict users to access some pages of your website.
  + It can encrypt data.

## 3.4.2.3 MySQL

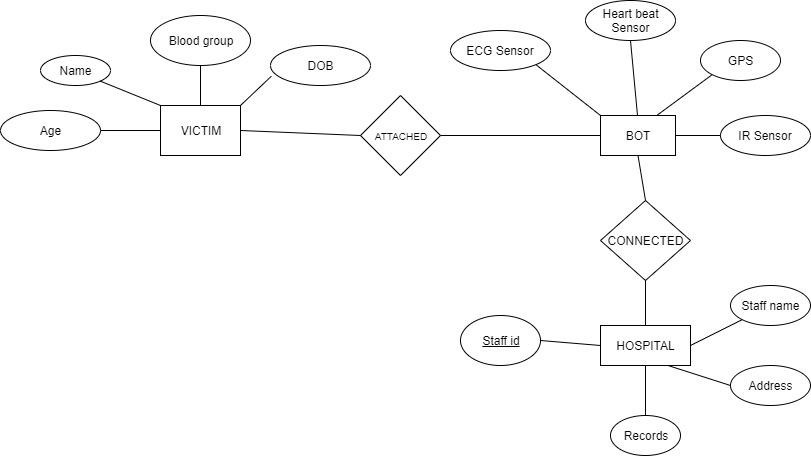
* + MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).
  + SQL is the most popular language for adding, accessing and managing content in a database.
  + It is most noted for its quick processing, proven reliability, ease and flexibility of use.
  + MySQL is an essential part of almost every open source PHP application.
  + The data in a MySQL database are stored in tables.
  + A table is a collection of related data, and it consists of columns and rows.

## 3.4.2.4 EMBEDDED C

* + In every embedded system based projects, Embedded C programming plays a key role to make the microcontroller run & perform the preferred actions.
  + At present, we normally utilize several electronic devices like mobile phones, washing machines, security systems, refrigerators, digital cameras, etc.
  + The controlling of these embedded devices can be done with the help of an embedded C program. For example in a digital camera, if we press a camera button to capture a photo then the microcontroller will execute the required function to click the image as well as to store it.
  + Embedded C programming builds with a set of functions where every function is a set of statements that are utilized to execute some particular tasks.
  + Both the embedded C and C languages are the same and implemented through some fundamental elements like a variable, character set, keywords, data types, declaration of variables, expressions, statements.
  + All these elements play a key role while writing an embedded C program.
  + The embedded system designers must know about the hardware architecture to write programs.
  + These programs play a prominent role in monitoring and controlling external devices.
  + They also directly operate and use the internal architecture of the microcontroller, such as interrupt handling, timers, serial communication, and other available features.

# CHAPTER 4 SYSTEM DESIGN

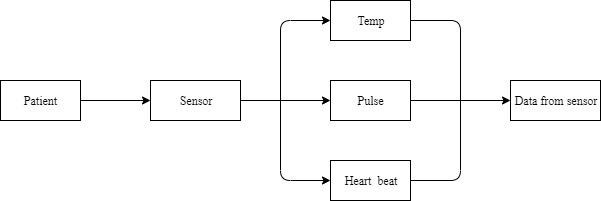
## 4.1 ER DIAGRAM

* ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database.
* In other words, ER diagrams help to explain the logical structure of databases.
* ER diagrams are created based on three basic concepts: entities, attributes and relationships.
* ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

#### Fig.4.1 ER diagram for robotic automated external defibrillator

**4.2 DATA FLOW DIAGRAM**

* + A picture is worth a thousand words. A Data Flow Diagram (DFD) is traditional visual representation of the information flows within a system.
  + A neat and clear DFD can depict a good amount of the system requirements graphically.
  + It can be manual, automated, or combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored.
  + The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.
  + It is usually beginning with a context diagram as the level 0 of DFD diagram, a simple representation of the whole system.



#### Fig.4.2.1 dfd level 0 diagram

* + To elaborate further from that, we drill down to a level 1 diagram with lower level functions decomposed from the major function of the system.

Untitled Diagram (7).jpg

#### Fig.4.2.2 dfd level 1 diagram

* + This could continue to evolve to become a level 2 diagram when further analysis is required.

Untitled Diagram (8).jpg

#### Fig.4.2.3 dfd level 2 diagram

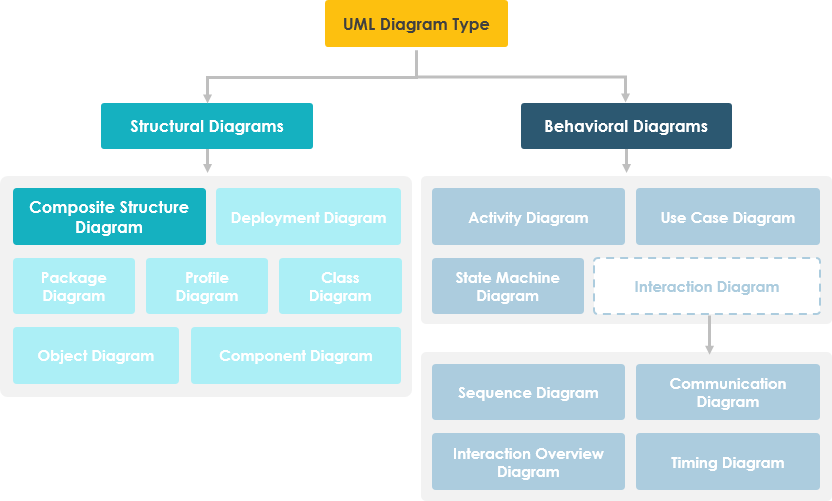
* + Progression to level 3, 4 and so on is possible but anything beyond level 3 is not very common. Please bear in mind that the level of details for decomposing function really depending on the complexity that function.

## 4.3 UML DIAGRAMS

* + UML stands for Unified Modeling Language.
  + It’s a rich language to model software solutions, application structures, system behavior and business processes.
  + There are 14 UML diagram types to help you model these behaviors.
  + Unified Modeling Language™ (UML®) is a standard visual modeling language intended to be used for modeling business and similar processes, analysis, design, and implementation of software-based systems.
  + UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.
  + Specification explained that process:
    - provides guidance as to the order of a team’s activities,specifies what artifacts should be developed,
    - directs the tasks of individual developers and the team as a whole, and
    - offers criteria for monitoring and measuring a project’s products and activities.
  + UML is intentionally process independent and could be applied in the context of different processes.
  + Still, it is most suitable for use case driven, iterative and incremental development processes
  + An example of such process is Rational Unified Process (RUP).
  + UML is not complete, and it is not completely visual.
  + Given some UML diagram, we can't be sure to understand depicted part or behavior of the system from the diagram alone.
  + Some information could be intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams.
  + For example, semantics of multiplicity of actors and multiplicity of use cases on use case diagrams is not defined precisely in the UML specification and could mean either concurrent or successive usage of use cases.
  + Name of an abstract classifier is shown in italics while final classifier has no specific graphical notation, so there is no way to determine whether classifier is final or not from the diagram.

## 4.3.1 LIST OF UML DIAGRAM TYPES

* + So, what are the different UML diagram types?
  + There are two main categories;
    - structural diagrams
    - behavioral diagrams.



## 4.4 STRUCTURAL DIAGRAMS

* + Structure diagrams show the things in the modeled system.
  + In a more technical term, they show different objects in a system.
  + Behavioral diagrams show what should happen in a system.
  + They describe how the objects interact with each other to create a functioning system.

## 4.5 CLASS DIAGRAM

* + Class diagrams are the main building block of any object-oriented solution.
  + It shows the classes in a system, attributes, and operations of each class and the relationship between each class.
  + In most modeling tools, a class has three parts.
    - Name at the top,
    - attributes in the middle and
    - operations or methods at the bottom.
  + In a large system with many related classes, classes are grouped together to create class diagrams.
  + Different relationships between classes are shown by different types of arrows**.**

## 4.6 COMPONENT DIAGRAM

* A component diagram displays the structural relationship of components of a software system.
* These are mostly used when working with complex systems with many components.
* Components communicate with each other using interfaces.
* The interfaces are linked using connectors.

## 4.7 DEPLOYMENT DIAGRAM

* + A deployment diagram shows the hardware of your system and the software in that hardware.
  + Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration.

## 4.8 PACKAGE DIAGRAM

* As the name suggests, a package diagram shows the dependencies between different packages in a system.
* Check out this wiki article to learn more about the dependencies and elements found in package diagrams.

## 4.9 COMPOSITE STRUCTURE DIAGRAM

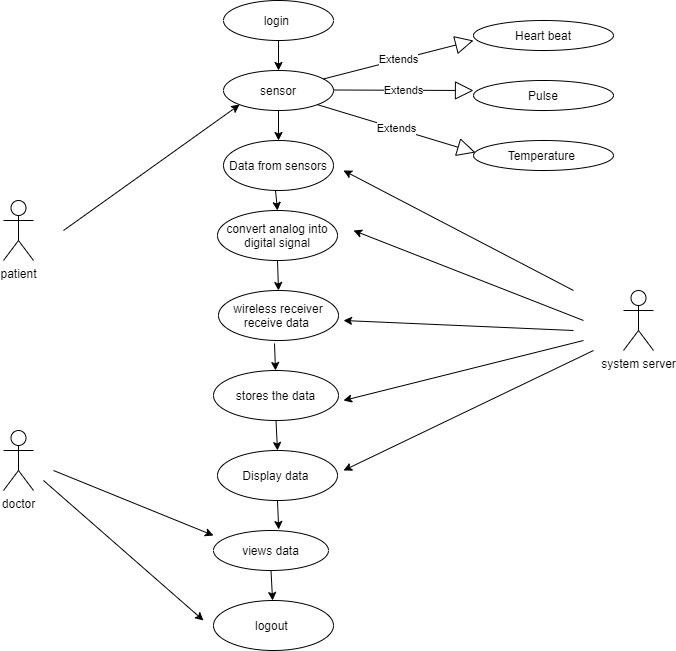
* A composite structure diagram is a UML structural diagram that contains classes, interfaces, packages, and their relationships, and that provides a logical view of all, or part of a software system.
* It shows the internal structure (including parts and connectors) of a structured classifier or collaboration.
* A composite structure diagram performs a similar role to a class diagram, but allows you to go into further detail in describing the internal structure of multiple classes and showing the interactions between them.
* You can graphically represent inner classes and parts and show associations both between and within classes.
* Composite structure diagrams are used to show the internal structure of a class.

## 4.10 BEHAVIORAL DIAGRAMS

* In a behavior diagram, individual aspects of a system and their changes are displayed at runtime.
* A behavior diagram is intended to provide clarity, for example, about internal processes, business processes or the interaction of different systems.
* Depending on the diagram used, a selected aspect is shown.
* In the Unified Modeling Language (UML), objects are modeled that can change their states through behavior.
* Basically, UML distinguishes between two different behavior specifications:
  + comon behaviors model
  + communication model
* In the Comon Behaviors Model, the basic occurrence of behavior is called “BehaviorPerformance”.
* Each behavior is triggered by an object in the role of the “Invoker”. BehaviorEmergence” results from the interaction of the objects involved.
* The communication model distinguishes between sender and receiver.
* The sending event is an “InvokerEvent” and the receiving event is the “ReceivingEvent”.
* The result of a send event is a “RequestObject”, which is transmitted to the receiver.
* If it arrives at the receiver, it causes a receiver event there that can initiate a behavior.
* The UML knows the following behavior diagrams:
  + [use case diagram](https://t2informatik.de/en/smartpedia/use-case-diagram/).
  + activity diagram.
  + [state machine diagram](https://t2informatik.de/en/smartpedia/state-diagram/).
  + communication diagram.
  + sequence diagram.
  + interaction overview diagram.
  + timing diagram.
* The last four types are also referred to as interaction diagrams.

## 4.11 USE CASE DIAGRAM

* As the most known diagram type of the behavioral UML diagrams, use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions interact.
* It’s a great starting point for any project discussion because you can easily identify the main actors involved and the main processes of the system.
* A Use Case describes a sequence of actions that provided something of unmeasurable value to an actor and is drawn as a horizontal ellipse.
* An actor is a person, organization or external system that plays a role in one or more interaction with the system.



#### Fig.4.11 Use case diagram for robotic automated external defibrillator

**4.12 ACTIVITY DIAGRAM**

 Activity diagrams represent workflows in a graphical way.

 They can be used to describe the business workflow or the operational workflow of any component in a system.

 An activity diagram shows the overall flow of control.

 Sometimes activity diagrams are used as an alternative to State machine diagrams.

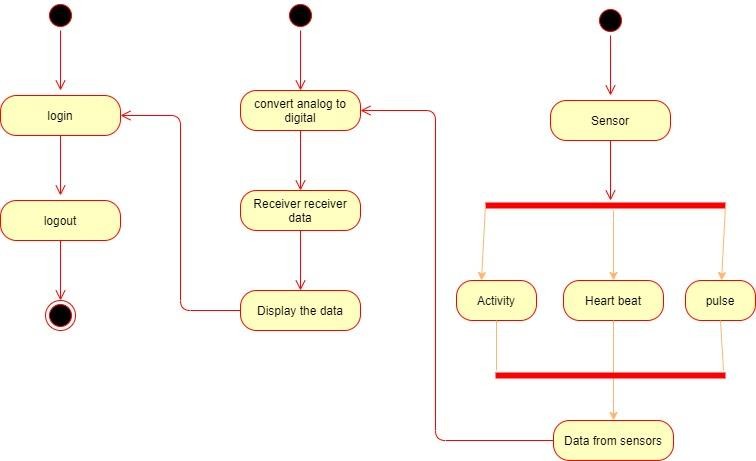
 Rounded rectangles represent activities.

 Diamonds represent decisions.

 Bars represent the start or end of concurrent activities.

 A black circle represents the start of the workflow.

 An encircled circle represents the end of the workflow



#### Fig.4.12 Activity diagram for robotic automated external defibrillator

**4.13 SEQUENCE DIAGRAM**

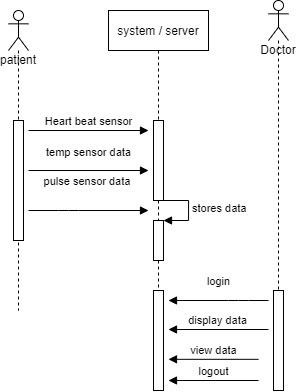
 Sequence diagrams in UML show how objects interact with each other and the order those interactions occur.

 It’s important to note that they show the interactions for a scenario.

 The processes are represented vertically, and interactions are shown as arrows.

 This article explains the purpose and the basics of Sequence diagrams.

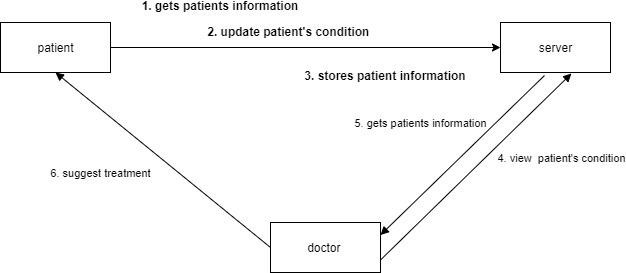
* It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



#### Fig 4.13 Sequence diagram for robotic automated external defibrillator

**4.14 COLLABORATION DIAGRAM**

* UML Collaboration Diagrams illustrate the relationship and interaction between software objects.
* They require use cases, system operation contracts and domain model to already exist.
* The collaboration diagram illustrates messages being sent between classes and objects.



#### Fig 4.14 Collaboration diagram for robotic automated external defibrillator

**CHAPTER 5**

**SYSTEM ARCHITECTURE**

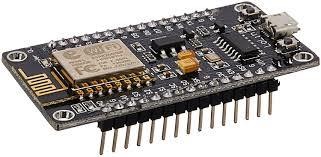
**5.1 ARCHITECTURE**



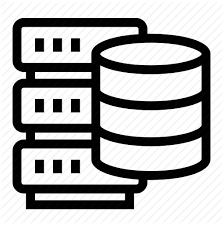
Temp sensor



Pulse sensor

IOT Board

Arduino Microcontroller



Heartbeat sensor

Server

#### Fig.5.1 System architecture for robotic automated external defibrillator

 System architecture is the conceptual model that defines the structure, behavior, and more views of a system.

 An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

## 5.2 SYSTEM MODULE

**5.2.1 DATA COLLECTION FROM PATIENT’S BY USING SENSOR**

 Sensor is a devices to measure temperature, pulse, heart beat readings through electrical signals.

 Temperature is the most common physical measurement type in industrial applications.

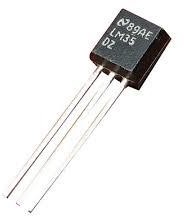
 Pulse sensors use the photoelectric method.

 The heart rate sensor measures your heart rate in Beats per Minute using an optical LED light source and an LED light sensor.

 The sensor will get the information of patient’s temperature, pulse, heartbeat and transmit it into microcontroller.

## 5.2.1.1 TEMPERATURE SENSOR

* The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly- proportional to the Centigrade temperature.
* The LM35 device is rated to operate over a −55°C to 150°C temperature range, while the LM35C device is rated for a −40°C to 110°C range (−10° with improved accuracy).



#### Fig.5.2.1.1 Temperature sensor

**5.2.1.2 PULSE SENSOR**

* Pulse Sensor is a well-designed plug-and-play heart-rate sensor for Arduino.
* 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.



#### Fig.5.2.1.2 Pulse sensor

**5.2.1.3 HEART BEAT SENSOR**

* Heartbeat Sensor is an electronic device that is used to measure the heart rate i.e. speed of the heartbeat.
* We can find the Principle of Heartbeat Sensor, working of the Heartbeat Sensor and Arduino based Heart Rate Monitoring System using a practical heartbeat Sensor.



#### Fig.5.2.1.3 Heart beat sensor

**5.2.2 Transmit data from ardunio to IOT board by use of UART protocol.**

* + Microcontrollers can take inputs from the device they controlling and retain control by sending the device signals to different parts of the device.
  + Microcontroller will transmit data from arduino to IOT board by using UART protocol.
  + It provides transmit data and a receive data and then send it to the server.
  + And it will be stored in server and display that information to the doctor.

## 5.2.2.1 ARDUINO

### Arduino is an open source microcontroller board based on the ATmega328 which is used to upload(burn) a program to the microcontroller using a USB cable.

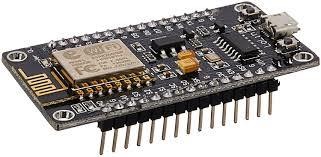
* + It also has a 14 input and output pins and regulated power of 5V.
  + Reset button which is present on the board can be used to reset the Arduino microcontroller.



#### Arduino

**5.2.2.2 IOT BOARD**

* + IoT has evolved from the convergence of wireless technologies, microelectromechanical systems (MEMS) and the Internet.
  + The concept may also be referred to as the Internet of Everything.
  + IoT board is featured with SIM900 GPRS modem to activate internet connection and also equipped with a controller to process all input UART data to GPRS based online data.
  + Data may be updated to a specific site or a social network by which the user can able to access the data over the internet.



#### IOT board

**5.2.3 Transmit data from IOT board to server by use of MQQT protocol**

* + IOT board will transmit the data to server by use of MQQT protocol.
  + IoT prototyping kits and development boards combine microcontrollers and processors.
  + MQ Telemetry Transport is an extremely simple and lightweight messaging protocol (subscribe and publish) designed for limited devices and networks with high latency, low bandwidth or unreliable networks. By using the data received from controller the doctor can suggest the treatment to the patient.
  + As it reduces the human effort then it definitely saves out time.

# CHAPTER 6 TESTING

## 6.1 TESTING PROCESS

* + Testing is performed to identify errors. It is used for quality assurance.
  + Testing is an integral part of the entire development and maintenance process.
  + The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself.
  + For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected.
  + The testing approach document is designed for Information and Technology Services’ upgrades to PeopleSoft.
  + The document contains an overview of the testing activities to be performed when an upgrade or enhancement is made, or a module is added to an existing application.
  + Testing is one of the important steps in the software development phase.
  + Testing checks for the errors, as a whole of the project testing involves the following test cases:
    - Static analysis is used to investigate the structural properties of the Source code.
    - Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

## 6.1.1 TYPES OF TESTING :

* UNIT TESTING
* INTEGRATION TESTING
* STRESS TESTING
* SMOKE TESTING
* STRUCTURE TEST
* PERFORMANCE TEST
* REGRESSION TESTING
* SECURITY TESTING
* PROGRAM TESTING
* VALIDATION TESTING
* USER ACCEPTANCE TESTING

**6.2 WHITE BOX AND BLACK BOX TESTING**

**6.2.1 WHITE BOX TESTING**

* + This testing is also called as Glass box testing.
  + In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function.
  + It is a test case design method that uses the control structure of the procedural design to derive test cases.
  + Basis path testing is a white box testing.
  + Basis path testing:
* Flow graph notation.
* Cyclometric complexity.
* Deriving test cases.
* Graph matrices Control.

## 6.2.2 BLACK BOX TESTING

* In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised.
* It fundamentally focuses on the functional requirements of the software.
* The steps involved in black box test case design are:
  + Graph based testing methods.
  + Equivalence partitioning.
  + Boundary value analysis.
  + Comparison testing.

## 

## 6.3 SOFTWARE TESTING STRATEGIES

* A software testing strategy provides a road map for the software developer.
* Testing is a set activity that can be planned in advance and conducted systematically.
* For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:
  + Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
  + Different testing techniques are appropriate at different points in time.

# CHAPTER 7

**CONCLUSION AND FUTURE ENHANCEMENT**

## 7.1 CONCLUSION

* In this paper, we propose an IOT-based system for patients with the risk of heart attack and uneven body temperature or any other sever condition in ambulance.
* If the condition is critical the information about the patient will be sent to the hospital database.
* This paper proposes a system to update patient data to hospital servers for analyzing.
* The doctor can view patient details, before the patient entering in to the hospital.
* We approach a system which is used to connect the ambulance and hospitals.

## 7.2 FUTURE ENHANCEMENT

* Our future work will be integrated with aadhar card and patients details.
* It will help to doctors to analyze the doctor for past history of the patients.
* The processes will include the creating a data base for every patients.

## SAMPLE CODE

#include<htc.h>

\_\_CONFIG(HS&WDTDIS&LVPDIS);

#define \_XTAL\_FREQ 10000000

void Key\_pad\_san();

void delay(unsigned int);

unsigned char flg,ke\_press;

#define RS RE0

#define RW RE1

#define EN RE2

#define LCD PORTD

void conv(unsigned int value);

void lcdinit(void);

void lcdcmd(unsigned char cmd);

void lcdchar(unsigned char chr);

void lcdstr(unsigned char \*str);

void conv\_st(unsigned int value);

void Ir\_dir();

unsigned char mot\_driv,flg;

void lcdinit()

{

lcdcmd(0x38);

\_\_delay\_ms(10);

lcdcmd(0x0E);

\_\_delay\_ms(10);

lcdcmd(0x01);

\_\_delay\_ms(10);

lcdcmd(0x06);

\_\_delay\_ms(10);

lcdcmd(0x0C);

\_\_delay\_ms(10);

lcdcmd(0x80);

\_\_delay\_ms(10);

}

void lcdcmd(unsigned char cmd)

{

LCD=cmd;

RS=0;

RW=0;

EN=1;

\_\_delay\_ms(50);

EN=0;

}

void lcdchar(unsigned char chr)

{

LCD=chr;

RS=1;

RW=0;

EN=1;

\_\_delay\_ms(1);

EN=0;

}

void lcdstr(unsigned char \*str)

{

for(;\*str;)

{

//\_\_delay\_ms(40);

LCD=\*str++;

RS=1;

RW=0;

EN=1;

\_\_delay\_ms(5);

EN=0;

}

}

void main()

{

PORTB=0xff; // port

TRISA=0xff;

TRISB=0xFf; //portb out

TRISC=0x80; //portc out

TRISD=0x00; //portc out

TRISE=0x00; //portc out

ADCON1=0x82;

ADCON0=0x81;

RBPU=0;

TRISC=0x80;

lcdinit();

lcdcmd(0x80);

lcdstr(" init,,, ");

delay(100);

lcdcmd(0xC0)

delay(100);

flg=0;

while(1)

{

if(RB0==0&&RB1==1&&RB2==0&&RB3==0 && flg==0)

{

mot\_driv=1;

flg=1;

}

else if(RB0==0&&RB1==0&&RB2==1&&RB3==0 && flg==0)

{

mot\_driv=2;

flg=1;

}

else if(RB0==0&&RB1==1&&RB2==1&&RB3==0 && flg==0)

{

mot\_driv=3;

flg=1;

}

else if(RB0==0&&RB1==0&&RB2==0&&RB3==1 && flg==0)

{

mot\_driv=4;

flg=1;

}

else

{

RC0=0;

RC1=0;

RC2=0;

RC3=0;

mot\_driv=0;

flg=0;

}

delay(100);

lcdcmd(0xC0)

delay(100);

switch(mot\_driv)

{

case 0:

Ir\_dir();

break;

case 1:

lcdstr("v=");

RC0=1;

RC1=0;

RC2=1;

RC3=0;

mot\_driv=0;

break;

case 2:

lcdstr("i=");

RC0=1;

RC1=0;

RC2=0;

RC3=0;

\_\_delay\_ms(2000);

RC0=0;

RC1=0;

RC2=0;

RC3=0;

\_\_delay\_ms(1000);

RC0=1;

RC1=0;

RC2=1;

RC3=0;

mot\_driv=0;

break;

case 3:

lcdstr("p=");

RC0=0;

RC1=0;

RC2=1;

RC3=0;

//\_\_delay\_ms(2000);

//\_\_delay\_ms(2000);

//\_\_delay\_ms(2000);

RC0=0;

RC1=0;

RC2=0;

RC3=0;

\_\_delay\_ms(1000);

RC0=1;

RC1=0;

RC2=1;

RC3=0;

mot\_driv=0;

break;

case 4:

RC0=0;

RC1=1;

RC2=0;

RC3=1;

mot\_driv=0;

break;

}

}

lcdstr(" i ");

delay(100);

lcdcmd(0xC0)

delay(100);

void delay(unsigned int t\_d)

{

unsigned int i;

for(i=0;i<=t\_d;i++)

{

\_\_delay\_ms(10);

}

}

void Ir\_dir()

{

if(RB7==0 && RB6==1 && RB5==1)

/RB7----input1, RB6----input2, RC3 Intput 3

{

RC0=0;

RC1=1;

RC2=1;

RC3=0;

\_\_delay\_ms(1000);

\_\_delay\_ms(1000);

RC0=1;

RC1=0;

RC2=1;

RC3=0;

}

else if(RB7==1 && RB6==0 && RB5==1 )

{

RC0=1;

RC1=0;

SRC2=0;

RC3=1;

\_\_delay\_ms(1000);

\_\_delay\_ms(1000);

RC0=1;

RC1=0;

RC2=1;

RC3=0;

}

else if(RB7==1 && RB6==1 && RB5==0)

{

RC0=0;

RC1=1;

RC2=1;

RC3=0;

\_\_delay\_ms(1000);

\_\_delay\_ms(1000);

RC0=1;

RC1=0;

RC2=1;

RC3=0;

}

}

lcdinit();

RCIE=1;

//INTE=1;

PEIE=1;

GIE=1;

lcdcmd(0x80);

lcdcmd(0xC0);

delay(100);

put\_str("AT\r\n");

\_\_delay\_ms(1);

put\_str("AT+CMGF=1\r\n");

\_\_delay\_ms(1);

put\_str("AT+CNMI=2,2,0,0,0\r\n");

\_\_delay\_ms(1);

flge1=0;

lcdcmd(0x01);

lcdcmd(0x80);

lcdstr("T:");

lcdcmd(0x85);

lcdstr("L:");

lcdcmd(0x8A);

lcdstr("H:");

T\_count=50;

cmd\_gsm=0;

while(1)

{

T\_count++;

ADCON0=0x81;

\_\_delay\_ms(10);

GODONE=1;

while(GODONE==1);

{

vlt=ADRESL;

vlt+=(ADRESH<<8);

vr=((vlt\*0.00488)/0.01);;

lcdcmd(0x82);

conv(vr);

\_\_delay\_ms(10);

ADCON0=0x89;

\_\_delay\_ms(10);

GODONE=1;

while(GODONE==1);

vlt1=ADRESL;

vlt1+=(ADRESH<<8);

vr1=(vlt1);

lcdcmd(0x87);

conv(vr1);

\_\_delay\_ms(10);

ADCON0=0x91;

\_\_delay\_ms(10);

GODONE=1;

while(GODONE==1);

vlt2=ADRESL;

vlt2+=(ADRESH<<8);

vr2=(vlt2);//\*20\*0.00488);//\*100\*0.00488);

lcdcmd(0x8c);

conv(vr2);

\_\_delay\_ms(10);

ADCON0=0x99;

\_\_delay\_ms(10);

GODONE=1;

while(GODONE==1);

vlt3=ADRESL;

vlt3+=(ADRESH<<8);

vr3=(vlt3);//\*20\*0.00488);//\*100\*0.00488);

}

lcdcmd(0xC4);

conv(vr3);

\_\_delay\_ms(10);

if(RB0==1)

{

lcdcmd(0xC0);

lcdstr("G:HIGH");

vr3=255;

}

else

{

lcdcmd(0xC0);

lcdstr("G:Low ");

vr3=0;

}

if(vr>45)

{

// RC1=1;

flge1=1;

}

else if(vr<45)

{

//RC1=0;

flge1=0;

}

if(vr1>0&&vr1<20)

{

flge2=1;

}

else

{

flge2=0;

}

if(T\_count>=100)

{

put\_str("AT+CMGS=\"sms\"\r\n");

\_\_delay\_ms(1000);

conv\_st(vr);

\_\_delay\_ms(1000);

put\_str("h:");

\_\_delay\_ms(1000);

conv\_st(vr1);

\_\_delay\_ms(1000);

put\_str("Le:");

\_\_delay\_ms(1000);

conv\_st(vr2);

\_\_delay\_ms(1000);

\_\_delay\_ms(10);

\_\_delay\_ms(1000);

put\_str("uploading");

\_\_delay\_ms(1000);

conv\_st(vr3);  
 \_\_delay\_ms(1000);

\_\_delay\_ms(1000);

put\_char(0x1a);

\_\_delay\_ms(1000);

\_\_delay\_ms(1000);

T\_count=0;

fun\_mo++;

flg=1;

}

lcdcmd(0xCc);

lcdchar(var[1]);

lcdchar(var[2]);

lcdchar(var[3]);

lcdchar(var[4]);

if(var[0]=='M'&&var[1]==' '&&var[2]=='O'&&var[3]=='N'&&gsm\_rec==1)

{

lcdcmd(0xC0);

lcdstr("data");

lcdcmd(0xCc);

lcdstr("rec1");

cmd\_gsm=1;

gsm\_rec=0;

RC0=1;

T\_count=0;

//M\_count=50;

flg=0;

}

else if(var[0]=='M'&&var[1]=='O'&&var[2]=='F'&&var[3]=='F'&&gsm\_rec==1)

{

lcdcmd(0xC0);

lcdstr(" ");

RC0=0;

lcdcmd(0xCc);

lcdstr("rec2");

cmd\_gsm=0;

gsm\_rec=0;

RC0=0;

T\_count=0;

}

if(flge1==1&&flge2==1&&cmd\_gsm==0)

{

RC0=1;

}

else if(flge1==0&&flge2==1&&cmd\_gsm==0)

{

RC0=1;

}

else if(flge1==1&&flge2==0&&cmd\_gsm==0)

{

RC0=1;

}

else if(flge1==0&&flge2==0&&cmd\_gsm==0)

{

RC0=0;

}

}

## REFERENCES

1. H. Ning et al.,“From Internet to smart world,” IEEE Access, vol. 3, pp. 1994-1999, Oct. 2015.

1. R. Jalali, K.El-khatib, and C. McGregor, “Smart city architecture for community level services through the Internet of Things,” in Proc. IEEE 18th Int. Conf. Intell.Next Generat. Netw. (ICIN), Feb. 2015, pp. 108-113.
2. M. Arif, H.Samani, C.-Y. Yang, and Y.-Y. Chen, “Adaptation of mobile robots to intelligent vehicles,” in Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on. IEEE,2013.

**4.** [1] F. Matsuno and S. Tadokoro, “Rescue robots and systems in japan in Robotics and Biomimetics, 2004. ROBIO2004. IEEE International Conference on. IEEE, 2004. [5] C.-P. Lam, C.-T. Chou, K.-H. Chiang and L.-C.Fu, “Human-centered robot navigation towards a harmoniously human–robot coexisting environment, ”Robotics, IEEE Transactions on, vol. 27, no. 1, pp. 99– 112,2011.