College of Engineering

Communication Engineering Program

Bachelor thesis

Graduation Project

Smart Wireless Alarm Safety System for Emergency Cases

Project Team

Amani N. Sharabati
Shatha M.F.Tebakhi
Sherin G. Abu Rajab
Sherin Y. Qutina

Project supervisor

Dr. Murad Abusbaih

Hebron-Palestine

May 2014

جامعة بوليتكنك فلسطين
الخليل - فلسطين
كلية الهندسة
دائرة الهندسة الكهربائية

اسم المشروع

Smart Wireless Alarm Safety System for Emergency Cases

أسماء الطالبات

اماني الشرباتي
شذا الطباخي
شيرين ابورجب
شيرين قطينة

بناء على نظام كلية الهندسة واشراف ومتابعة المشرف المباشر على المشروع ومتابعة اعضاء اللجنة المتحنة، تم تقديم هذا المشروع الى دائرة الهندسة الكهربائية،وذلك استكمالا لمتطلبات درجة البكالوريوس في تخصص هندسة الاتصالات والالكترونيات.

توقيع المشرف

توقيع اللجنة المناقشة

توقيع رئيس الدائرة

Contents

List of Figures

List of Tables

Dedication

This project is lovingly dedicated to our parents and families, thank you for your unconditional support with our studies. We are honored to have you as our families. Thank you for offering us the education we wanted, thank you for giving us a chance to prove and improve ourselves through all our work in this life, for giving all our need during the time we developed our system, and for teaching us that even the largest task can be accomplished if it is done one step at a time.

Without your love and support this project would not have been made possible. We would also like to dedicate this project to our friends. Thank you for believing in us, for your unlimited support, everlasting guidance and endless inspiration.

Acknowledgment

We would like to acknowledge Dr. Murad Abusbaih and the support and assistance given by instructors in the Electrical Engineering Department for their great impact in our education. We are grateful to many persons, who shared their memories and experiences, especially Eng-Sa'di Tamimi, Mr. Mohammed Tamimi, Eng. Mohammed Jabary, Eng. Ahmad Tamimi and Eng. Iman Sharabati for all their valuable advices.

We must also acknowledge Palestine Polytechnic University for giving us the opportunity to experience real engineering through this project, and for offering us to act with the laboratories in it.

Finally, we thank all those who assisted, encourage and support us during this project, Thank you all !

Abstract

We are living in the world of technology. Quick and efficient response to emergency is important. In this project, it is intended to design and implement a system to be used in emergency cases (Police, Fire brigade, Ambulance). The system can be used in houses, factories, schools, etc. The system enables fast calling of emergency services. Further, the details about location are delivered to proper station. The station determines the closest E-car; the responder will use a specific algorithm to find out the shortest route to this calling side.

The system is comprised of three switches, arduino microcontroller and GPRS/GPS shield. Switches are used for selecting the proper station to be called. GPS provides current coordinates that are sent to a server over GSM/GPRS network. The server uses specific algorithm and finds out the closest E-car. Necessary information is passed to this car, which uses another algorithm to find out the shortest path to reach the calling location.

1

Chapter One

Introduction

1.1 Overview

1.2 Motivation

1.3 Objectives

1.4 Approach

1.5 System Requirement

1.6 Litrature Review (Related Projects)

1.7 Problem Statement

1.8 System Cost

1.9 System Schedule

Chapter 1

Introduction

1.1 Overview

Emergency Services is something very important in the community to focus on it. Service providers need to reach the site of the accident in golden hour, so the life of people can be saved.

Our homes, factories, companies, etc. are susceptible to have a lot of accidents. Majority of people do not know the numbers of these services, other confused and cannot behave correctly.

Anyway, calling emergency to give them all needed information to reach the location take a lot of time which aggravate the problem, thus, the need arises to build a system that opens a data route to send the location of the accident, and identify some information includes name and SIM number to the appropriate station (Police, Fire or Emergency).

At the receiver side, the server will choose the closest car to the calling side and send the request to it. The chosen car will use the shortest path algorithm to reach accident location.

1.2 Motivation

According to the latest statistics published by civil defense in West Bank through August 2013, the financial losses due to fire accidents reached 2,216,917 NIS, and the number of injuries about 109, and the number of fire accidents reached 572, and so 235 of them got the right help, so it is noticed that the majority of the occurred accidents is fire or theft or human injuries [3].

From here the idea of developing a system that connecting a notification about the most happened accident with responsible sides in one device. This reduces the required time to reach these sides, so people can replace traditional way by this system and getting better result. On the other hand some institutions like factories will be more safety if a system like that will be always activated in it.

1.3 Objectives

- Design and implement a system that allows fast calling of police, ambulance, and fire-engine.
- Allow the police, ambulance, and fire brigade to reach the called location as quick as possible.

1.4 Approach

The system will be comprised of a main node that includes three switches to allow calling to (police, ambulance, and fire brigade) station. This node is equipped with a GSM and GPS elements. It allows sending of requests to proper destinations.

At the receiver side, requests are processed and the shortest route to the calling side is determined.

1.5 System Requirement

- The system will contain GSM module, which will send a call setup and other information through GPRS from the accident place to the emergency service.
- The system will contain GPS module, which will determine the accident location exactly.
- The system will contain a microcontroller, which communicates with the help of serial communication. First it takes the data from the GPS receiver and then sends the information to the station by using GPRS with help of GSM modem.
- Each main emergency center will have a server, the server will have an algorithm to determine the nearest emergency car to reach the location of accident after receiving the longitude and latitude data from GPS once the accident occurred.
- Each emergency car will have a smart phone that provide the map of the road, that it must follow according to the center analysis to reach the accident place as fast as possible.

1.6 Literature Review (Related Projects)

There are several related projects. However, the system is different and more efficient. Some of the related projects are listed. The differences to the system are presented.

1.6.1 GSM Based Emergency Calling System

This project presents an emergency calling system consisting of GSM module, solar panel which is connected with centralized control room by wireless communication (totally based on GSM).

In case of accidently emergency occurred this system will be very helpful. It uses different alarm circuits. So, if any attempt of theft occurs, vibration sensor will directly inform the control room by this attempt, and person from control room call on the system. Then, it automatically responses and sends message to the mobile.[4]

Advantages:

- Reduce the probability of attempt of theft.
- Continuously powered by solar panel.

Disadvantages:

- High delay, because the call must first send to a control room and then to destination.
- The use of SMS service is considered to be highly cost compared to other technologies such as GPRS.

Differences:

- We want to use GPRS instead of SMS.
- By pressing any of the buttons, the place of accident will be connected directly to appropriate destination without need for control room.

1.6.2 Design Of Intelligent Fire Alarm System Based On GSM Network

This project presents a solution to the problem of complex cabling, misdeclaration and missing alarm of traditional fire alarm system. An intelligent fire alarm system based on GSM network is designed.

This system adopts MSP430F149 as main control chip and the remote alarming and data exchanging are achieved by using GSM module TC35I.

The accuracy of fire alarm is improved by depending on smoke detector and temperature detector, and using a variable threshold alert algorithm with temperature compensation.[5]

Advantages:

- Supports a character of real-time and good reliability.

Disadvantages:

- It is only reliable in fire cases and doesn't useful in other like (theft, injuries).
- Depends on the sensitivity of detector. If it is low, all places will be fired and detector will not response.

Differences:

- In our project all emergency cases will be covered (fire, theft, injuries).
- Continuously tracking of moving objects and support them with short path to the accident location.

1.7 Problem Statement

The challenging aspect in this project can be listed as:

- Provide caller station in real time, accurate information about the calling side.
- Determine the closest point to the calling side.
- Determine the shortest route for the closest point to the caller.

1.8 System Cost

The following table describes the overall cost of the system:

Table 1.1: overall System Cost

Item	Cost (NIS)
Arduino Uno	850
SIM908 Arduino Shield	180
Emergency Switches	135
Resistors and led	20

1.9 System Schedule

The following tables describe the overall stages of the system:

Table 1.2: overall system timing table for the first semester

Task \ Week	First semester														
	1th	2th	3th	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th	15th
Choose the project and supervisor															
Collecting data about technologies used															
Collecting data about components															
Schematic Design															
Documentation															

Table 1.3: overall system timing table for the second semester

Task \ Week	Second semester														
	1th	2th	3th	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th	15th
Pay the hardware component															
Preparing software skills for the project															
Build up the software															
Testing the software															
Testing the hardware															
Testing the system															
Documentation writing															

2

Chapter Two

Theoretical Background

Chapter 2

Theoretical Background

2.1 Introduction

This chapter provides the basic theoretical information, some technologies, and some devices that will be used in the project. In order to use them in an appropriate way and take the desired advantage of them.

We will talk about the GPS technology, GSM/GPRS technology, software that will be used.

2.2 GSM Technology

GSM (global system for mobile communication) is a digital mobile phone technology used to transmit voice and data services, first launched in Finland in the early 1990s. GSM, the most popular of the three digital wireless technologies including GSM, TDMA and CDMA, is widely used in Europe, Asia and other parts of the world. In achieving voice communications between devices, GSM technology first digitizes and compresses data streams before transmitting them through a channel with two other user-specific data streams. [6]

In its 20 years of existence, GSM has evolved from a technology developed to handle simple voice calls into a comprehensive platform supporting advanced broadband and multimedia services. According to GSM World, the foremost association representing the interests of the GSM community, the technology is now present in 219 countries, serving more than 3 billion users worldwide. In 2010, there are nearly 800 GSM service providers all over the world, covering more than 80 percent of the world's land mass. The technology also supports more than 200 related companies in the broader ecosystem, including handset manufacturers, content developers, equipment providers and media organizations.[7]

2.2.1 The GSM Network

The GSM network is divided into three major systems: the switching system (SS), the base station system (BSS), and the operation and support system (OSS).[8] The basic GSM network elements are shown in Figure 2.1.

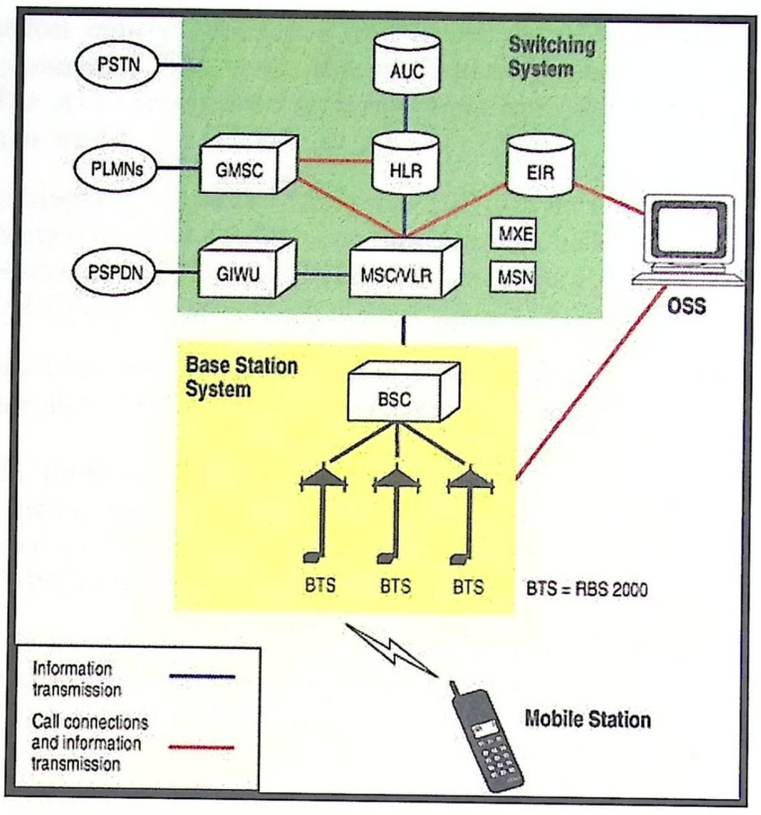


Figure 2.1: GSM network elements

The switching system (SS) is responsible for performing call processing and subscriber-related functions. The switching system includes the following functional units:

- **Home location registers (HLR)**: the HLR is a database used for storage and management of subscriptions. The HLR is considered the most important database, as it stores permanent data about subscribers, including a subscriber's service profile, location information, and activity status. When an individual buys a subscription from one of the PCS operators, he or she is registered in the HLR of that operator.

- **Mobile services switching center (MSC)**: the MSC performs the telephony switching functions of the system. It controls calls to and from other telephone and data systems. It also performs such functions as toll ticketing, network interfacing, common channel signaling, and others.

- **Visitor location registers (VLR)**: the VLR is a database that contains temporary information about subscribers that is needed by the MSC in order to service visiting subscribers. The VLR is always integrated with the MSC. When a mobile station roams into a new MSC area, the VLR connected to that MSC will request data about the mobile station from the HLR. Later, if the mobile station makes a call, the VLR will have the information needed for call setup without having to interrogate the HLR each time.

- **Authentication center (AUC)**: a unit called the AUC provides authentication and encryption parameters that verify the user's identity and ensure the confidentiality of each call. The AUC protects network operators from different types of fraud found in today's cellular world.

- **Equipment identity register (EIR)**: the EIR is a database that contains information about the identity of mobile equipment that prevents calls from stolen, unauthorized, or defective mobile stations. The AUC and EIR are implemented as stand-alone nodes or as a combined AUC/EIR node.

All radio-related functions are performed in *the Base Station System (BSS)*, which consists of base station controllers (BSCs) and the base transceiver stations (BTSs).

- **BSC**: the BSC provides all the control functions and physical links between the MSC and BTS. It is a high-capacity switch that provides functions such as handover, cell configuration data, and control of radio frequency (RF) power levels in base transceiver stations. A number of BSCs are served by an MSC.

- **BTS**: the BTS handles the radio interface to the mobile station. The BTS is the radio equipment (transceivers and antennas) needed to service each cell in the network. A group of BTSs are controlled by a BSC.

2.2.1.1 The Operation and Support System

The operations and maintenance center (OMC) is connected to all equipment in the switching system and to the BSC. The implementation of OMC is called the operation and support system (OSS). The OSS is the functional entity from which the network operator monitors and controls the system. The purpose of OSS is to offer the customer cost-effective support for centralized, regional and local operational and maintenance activities that are required for a GSM network. An important function of OSS is to provide a network overview and support the maintenance activities of different operation and maintenance organizations.

Additional Functional Elements

Other functional elements shown in Figure 2.1 are listed as follows [8]:

- **Message center (MXE)**: The MXE is a node that provides integrated voice, fax, and data messaging. Specifically, the MXE handles short message service, cell broadcast, voice mail, fax mail, email, and notification.

- **Mobile service node (MSN)**: The MSN is the node that handles the mobile intelligent network (IN) services.

- **Gateway mobile services switching center (GMSC)**: A gateway is a node used to interconnect two networks. The gateway is often implemented in an MSC. The MSC is then referred to as the GMSC.

- **GSM interworking unit (GIWU)**: The GIWU consists of both hardware and software that provides an interface to various networks for data communications. Through the GIWU, users can alternate between speech and data during the same call. The GIWU hardware equipment is physically located at the MSC/VLR.

2.2.2 GSM Subscriber Services

There are two basic types of services offered through GSM: telephony (also referred to as teleservices) and data (also referred to as bearer services).[8]

Telephony services are mainly voice services that provide subscribers with the complete capability (including necessary terminal equipment) to communicate with other subscribers. Data services provide the capacity necessary to transmit appropriate data signals between two access points creating an interface to the network.

In addition to normal telephony and emergency calling, the following subscriber services are supported by GSM:

- **Dual-tone Multifrequency (DTMF)**: DTMF is a tone signaling scheme often used for various control purposes via the telephone network, such as remote control of an answering machine. GSM supports full-originating DTMF.
- **Facsimile group III**: GSM supports CCITT Group 3 facsimile. As standard fax machines are designed to be connected to a telephone using analog signals, a special fax converter connected to the exchange is used in the GSM system. This enables a GSM-connected fax to communicate with any analog fax in the network.
- **Short Message Services**: a convenient facility of the GSM network is the short message service. A message consisting of a maximum of 160 alphanumeric characters can be sent to or from a mobile station. This service can be viewed as an advanced form of alphanumeric paging with a number of advantages. If the subscriber's mobile unit is powered off or has left the coverage area, the message is stored and offered back to the subscriber when the mobile is powered on or has reentered the coverage area of the network. This function ensures that the message will be received.
- **Cell broadcast**: a variation of the short message service is the cell broadcast facility. A message of a maximum of 93 characters can be broadcast to all mobile subscribers in a certain geographic area. Typical applications include traffic congestion warnings and reports on accidents.
- **Voice mail**: this service is actually an answering machine within the network, which is controlled by the subscriber. Calls can be forwarded to the subscriber's voice-mail box and the subscriber checks for messages via a personal security code.

- **Fax mail**: with this service, the subscriber can receive fax messages at any fax machine. The messages are stored in a service center from which they can be retrieved by the subscriber via a personal security code to the desired fax number.

2.2.3 GSM Flexibility

GSM mobile phones use a SIM (subscriber identification module), a small, flat card inserted into a slot inside the phone. Unless a phone is locked by the company, a user may switch phones by simply taking off the SIM from one device and inserting it into another, without having to notify the service provider. Another flexibility feature offered by GSM companies is *roaming* GSM operators, under roaming agreements with foreign operators, can continue providing services to their customers, even when users leave the home network and travel abroad. Through roaming, a customer from the U.S. may use a "borrowed network" in India, provided the two operators have roaming agreements.[7]

In our project, the basic function of GSM/GPRS module opening a data route to send the location of accident and identify information, and also used to send the location of police car's permanently to determine the nearest one and then the shortest path.

2.3 GPRS Service

GPRS (general packet radio service) is a packet-based data bearer service for wireless communication services that is delivered as a network overlay for GSM, CDMA and TDMA (ANSI-I36) networks. GPRS applies a packet radio principle to transfer user data packets in an efficient way between GSM mobile stations and external packet data networks. Packet switching is where data is split into packets that are transmitted separately and then reassembled at the receiving end.

GPRS supports the world's leading packet-based Internet communication protocols, Internet protocol (IP) and X.25, a protocol that is used mainly in Europe. GPRS enables any existing IP orX.25 application to operate over a GSM cellular connection. Cellular networks with GPRS capabilities are wireless extensions of the Internet and X.25 networks.

GPRS gives almost instantaneous connection set-up and continuous connection to the Internet. GPRS users will be able to log on to an APN (Access Point Name) and have access to many services or an office network (without the need to dial-up) and remain continuously connected until they log off, only paying when data is actually transmitted. A physical end-to-end connection is not required because network resources and bandwidth are only used when data is actually transferred. This makes extremely efficient use of available radio bandwidth. Therefore, GPRS *packet-based* services should cost users less than *circuit-switched* services since communication channels are being shared and are on a "as-packets-are-needed" basis rather than dedicated to only one user at a time. It should also be easier to make applications available to mobile users because the faster data rate means that middleware currently needed to adapt

applications from fixed line rates to the slower speed of wireless systems will no longer be needed.

GPRS data speeds will range from 14.4 Kbit/s (using one radio timeslot) to 115kbit/s (by amalgamating timeslots) and offer continuous connection to the Internet for mobile phone and computer users. GPRS data speeds are likely to average at about 56 Kbit/s, with between 28 and 40 Kbit/s initially. The higher data rates will allow users to take part in video conferences and interact with multimedia web sites and similar applications using mobile handheld devices as well as notebook computers [9] .

2.3.1 Key features of GPRS Over GSM

- Higher bandwidth and, therefore, data speeds.
- Seamless, immediate and continuous connection to the internet "always on-line".
- New text and visual data and content services, such as email, chat, still and moving images, information services.
- Packet switching rather than circuit switching ,which mean that is higher radio spectrum efficiency because network resources and bandwidth are only used when data is actually transmitted even though it is always connected .
- Support for leading internet communication protocols- internet protocol (IP)and X.25.
- Different device (not GSM phone)-GPRS will be available for laptops or handheld computers or other devices, all of these devices have user interface that will allow users to utilize GPRS services.
- The first important step on the path to 3G.

2.3.2 GPRS Network Elements

GPRS technology brings many changes to the existing GSM network. Most of the modification made by adding new blocks rather than by modifying existing resources as shown in figure 2.2, a simplified view of this new hybrid network shows the elements introduced by GPRS.

GGSN

The gateway GPRS support node, is similar to the GSM gateway mobile switching center (GMSC) and provides a gateway between the GPRS network and the public packet data network (PDN) or other GPRS networks .The GGSN provides authentication and location management functions, connects to the home location register (HLR) by means of the Gc interface, and counts the number of packets transmitted for accurate subscriber billing.

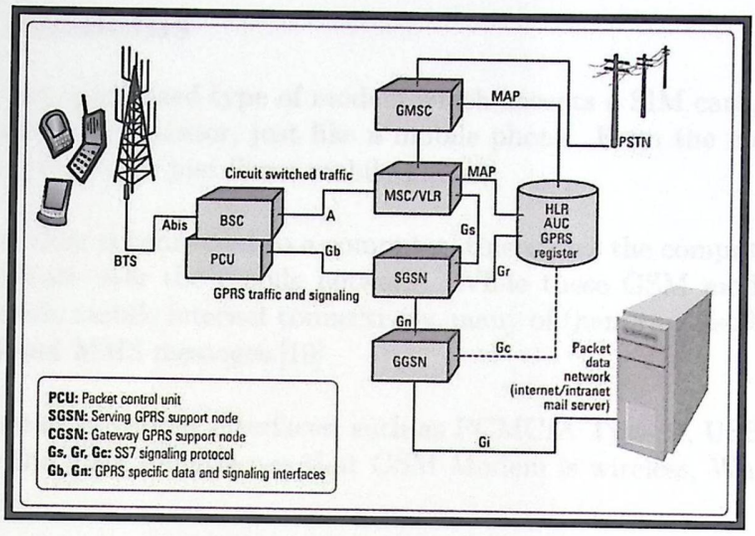


Figure 2.2: GPRS network architecture

SGSN

The serving GPRS support node, like the GSM mobile switching center and visitor location register (MSC/VLR), controls the connection between the network and the mobile station (MS). The SGSN provides session management and GPRS mobility management functions such as handovers and paging. It attaches to the HLR via the Gr interface and to the MSC/VLR via the Gs interface. It also counts the number of packets routed.

PCU

Functions of the packet control unit (PCU) include converting packet data into a format that can be transferred over the air interface, managing radio resources, and implementing quality of service (QoS) measurements.

Signaling Links

The signaling links between the GPRS nodes and the GSM blocks will be SS7 MAP interfaces. The signaling between GPRS nodes is defined by the GPRS specification. New physical interfaces include the Gb interface, which connects the SGSN to the PCU and is usually located in the base station subsystem (BSS); the Gn interface, which connects the GGSN and SGSN; and the Gc, Gr, and Gs interfaces, which carry SS7-based protocols.

In our project, we will use GPRS for sending the information of the accident to appropriate station. Further GPRS is used to send periodically the location of the police car's to the station, which determined by the GPS receiver.

2.4 GSM Modems

A GSM modem is a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. From the mobile operator perspective, a GSM modem looks just like a mobile phone.

When a GSM modem is connected to a computer, this allows the computer to use the GSM modem to communicate over the mobile network. While these GSM modems are most frequently used to provide mobile internet connectivity, many of them can also be used for sending and receiving SMS and MMS messages.[10]

GSM Modem comes in various interfaces, such as PCMCIA Type II, USB, and serial. GSM modem is however the main difference is that GSM Modem is wireless, While dial-up modem is wired.

Some GSM Modems also has GPRS feature that allows transmission of data over TCP/IP (internet). To transmit data using GSM Modem, there are various methods that can be used, such as:

- SMS.
- CSD or HSCSD.
- GPRS / UMTS.

Even though a normal mobile phone can be used as GSM Modem, it is highly recommended that a special industrial grade terminal to be used as a GSM Modem due to its stability, and reliability. A GSM Modem can be used to build the following applications [11] :

- SMS Gateway i.e. to send and receive SMS.
- Telemetric i.e. to collect data from remote terminals.
- call-back service for VOIP.
- SMS application, SMS solution, or SMS programmer.
- automatic reloading of pre-paid account with STK API.
- machine to machine communication.
- sending SMS from PC.
- automating business process.
- vehicle tracking with cell broadcast feature or with integrated GPS terminal.

2.4.1 GSM/GPRS MODEM Types

SIM300

SIM300 is a Tri-band GSM/GPRS engine as shown in figure 2.3 that works on frequencies EGSM 900 MHz DCS 1800 MHz and PCS1900 MHz, SIM300 provides GPRS multi-slot class 10 capabilities and support the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

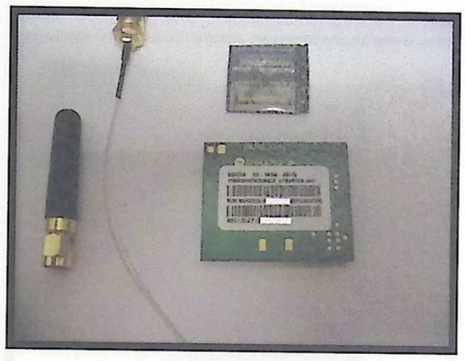


Figure 2.3: SIM 300

With a tiny configuration of 40mm x 33mm x 2.85 mm, SIM300 can fit almost all the space requirement in your application, such as Smart phone, PDA phone and other mobile device.

The physical interface to the mobile application is made through a 60 pins board-to-board connector, which provides all hardware interfaces between the module and customer's boards except the RF antenna interface.

The keypad and SPI LCD interface will give you the flexibility to develop customized applications, so with the two serial ports you can easily develop your applications, there is also two audio channels include two microphones inputs and two speaker outputs. This can be easily configured by AT command.

SIM300 provide RF antenna interface with two alternatives: antenna connector and antenna pad. The antenna connector is MURATA MM9329-2700. And customer's antenna can be soldered to the antenna pad.

The SIM300 is designed with power saving technique, the current consumption to as low as 2.5mA in SLEEP mode. The SIM300 is integrated with the TCP/IP protocol, Extended TCP/IP AT commands are developed for customers to use the TCP/IP protocol easily, which is very useful for those data transfer applications.[12]

SIM900B

This is a complete Quad-band GSM/GPRS module in a B2B type and designed with a very powerful single-chip processor integrating AMR926EJ-S core, allows getting a benefit from small dimensions and cost-effective solutions, as shown in figure 2.4. SIM900B delivers GSM/GPRS 850/900/1800/1900MHz performance for voice, SMS, Data, and Fax in a small form factor and with low power consumption. SIM900B can fit almost all requirements in your M2M applications. [13]

Figure 2.4: SIM900B Modem

The Features of this device are:

- Dimensions: 40*33*3 mm.
- Weight: 19g.
- Quad-Band 850 / 900 / 1800 / 1900 MHz.
- Control via AT commands according to GSM 07.05, 07.07, and SIMCOM enhanced AT Commands.
- Supply Voltage Range: 3,2 - 4,8 VDC.
- Temp. Range: -40°Cto +80°C.
- Software features: embedded TCP/UDP protocol , FTP/HTTP.
- Serial interface.
- Low power consumption: 1.0mA (sleep mode).

SIM908

Shield Arduino that can accept the modules from Simcom SIM900 (GSM/GPRS functionality) and SIM908 (functionality GSM / GPRS & GPS) and allows to make voice calls, send SMS or make connections to the Internet network.

The availability of new GSM/GPRS & GPS library, allow great flexibility. For example is possible reading a received text message or start a call and make all other operations performed by a standard mobile phone.

The board is compatible with Arduino Duemilanove, Arduino UNO, Arduino MEGA, etc. has two jacks for analog audio. With a microphone and a headset with 3.5 mm jack you can make a voice call. In case of receiving a call, through headphones, an alarm ring alerts you of incoming calls.

The shield includes a high capacitor dedicated to RTC (Real Time Clock) provided inside the SIM900 and SIM908. The capacitor allows to keep updated the time even in the absence of main power. The circuit operates with a supply voltage of 12 Vdc supplied directly from the Arduino board. The shield provides a connector for connecting an external battery lithium rechargeable only with SIM908: the module incorporates a dedicated charging circuit, the required voltage for charging the battery is taken directly from the Arduino board(from pin Vin).[14]

2.5 GPS Technology

2.5.1 Definition

The Global Positioning System (GPS) is a technical marvel made possible by a group of satellites in earth orbit that transmit precise signals, allowing GPS receivers to calculate and display accurate location, speed, and time information to the user. By capturing the signals from three or more satellites (among a constellation of 31 satellites available), GPS receivers are able to use the mathematical principle of trilateration to pinpoint your location.

GPS was originally created by the United States Department of Defense (DOD) as a military application. The system has been active since the early 1980s, but began to become useful to civilians in the late 1990s. Consumer GPS has since become a multi-billion dollar industry with a wide array of products, services, and Internet-based utilities.[15]

2.5.2 How does it Work

GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. GPS receivers take this information and use triangulation to calculate the user's exact location. Essentially, the GPS receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. Now, with distance measurements from a few more satellites, the receiver can determine the user's position and display it on the unit's electronic map, the following figure 2.5 illustrates the concept of GPS receiver.

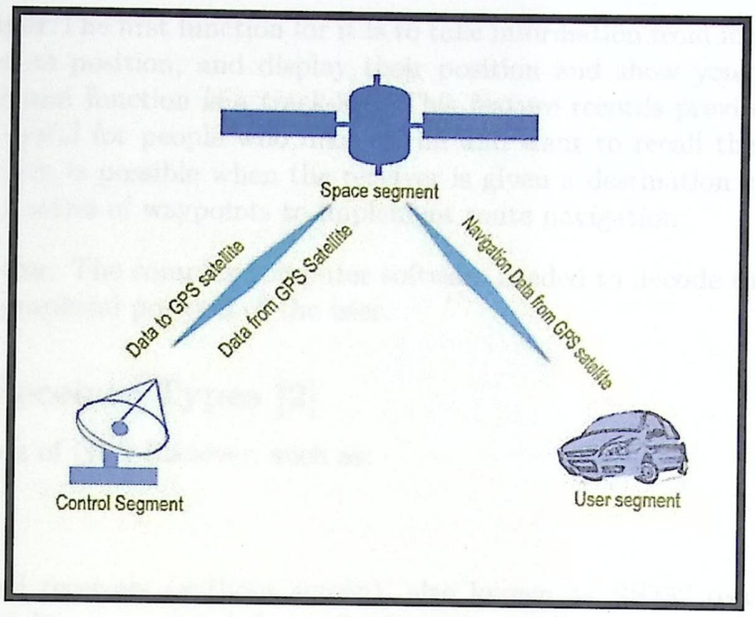


Figure 2.5: GPS Structure

A GPS receiver must be locked on to the signal of at least three satellites to calculate a 2D position (latitude and longitude) and track movement. With four or more satellites in view, the receiver can determine the user's 3D position (latitude, longitude and altitude). Once the user's position has been determined, the GPS unit can calculate other information, such as speed, bearing, track, trip distance, distance to destination, sunrise and sunset time and more.[15]

Information That GPS Provides

- Your current position: coordinates (latitude & longitude) and elevation.
- Direction to specified waypoints.
- Distance to specified waypoints.
- Your speed of travel.
- Your direction of travel.

2.5.3 GPS Components [1]

The three main components are :

1. **GPS Satellites**: Up to 30 GPS satellites fly, mostly in highly inclined (polar) orbits, at altitudes around 20,000 km. This means that there will be between four and eight of them reasonably high in the sky above any site on the Earth at any time.

2. **GPS Receiver**:The first function for it is to take information from four or more satellites to determine its position, and display their position and show you your location on a map. The second function is a track-log. This feature records previous positions and is particularly useful for people who hike or run and want to recall their route. Point-to-point navigation is possible when the receiver is given a destination coordinate. Finally, you can use a series of waypoints to implement route navigation.

3. **GPS Software**: The complex computer software needed to decode the signals and compute the geographical position of the user.

2.5.4 GPS Receiver Types [2]

There is many types of GPS Reciever, such as:

Type 1

Not-self-contained receivers (without screen), also known as RS232 receivers or also GPS mice. This type needs a computer (often a Pocket-PC or Palm PDA) and according program in order to visualize the actual position of the GPS receiver.

Type 2

Self-contained receivers (with screen). Here the computer is integrated in the GPS receiver. We have to distinguish between non-mapping devices (often handheld) and mapping devices which can be handheld or mounted in boat, car or plane (often with external antenna).

Type 3

The more sophisticated receivers for professional use in agriculture, mapping and GIS, military, oil and gas, public safety, survey etc. The principle of these receivers is the same as the above, but they often use extra receivers and special antennas for higher accuracy and can often store an enormous amount of in the field collected data, which can be treated later in the office.

Type 4

Very special applications, using GPS, as there are vehicle tracking systems (See our GPS Car Tracking page) and child-locator devices (See our GPS Locators page). Both use a GPS-chip to measure their location and some form of wireless phone system to communicate this information over great distances, eventually via Internet.

Type 5

Phones with incorporated GPS receiver. They could already be used for in-car navigation with street maps coming in through the air. End May 2005 Sylvan Ascent Inc. launched TopoPhone, which provides outdoor recreation enthusiasts with a better GPS solution by turning a cell

phone with built-in GPS into a complete Topographic mapping system. TopoPhone is superior to traditional GPS units, because it uses the phone to download maps automatically, and allows you to see where your friends and family are located. See also our GPS phones page.

2.5.5 GPS Receiver Modems

GR-213u GPS Receiver-Holux

GR-213U Smart GPS Receiver which is shown in figure 2.6; is a total solution GPS receiver, designed based on SiRF Star III Architecture, this positioning application meets strict needs such as car navigation, mapping, surveying, security, agriculture and so on, only clear view of sky and certain power supply are necessary to the unit. It communicates with other electronic utilities via compatible dual channel through USB interface and saves critical satellite data by built in backup memory. With low power consumption, the GR-213U tracks up to 20 satellites at a time, re-acquires satellite signals in 100 ms and updates position data every second.[16]

Figure 2.6: GR-213u GPS Receiver-Holux

The features of this device are:

- Tracks up to 20 satellites.
- Max update rate: 1 HZ.
- Antenna Type: Built in Patch Antenna.
- Minimum signal tracked: -159dBm.
- Power consumption: 80mA at 4.5- 5.5V input.
- Dimension: 2.54 × 1.65 × 0.7.

2.5.5.1 GP-635T GPS Receiver

The GP-635T which is shown in figure 2.7; this a slim GPS module with -161dBm tracking sensitivity and only 27 second cold start time. The slim design makes it ideal for applications where you don't have a lot of space to work in. Really it's quite small. This 50-channel GPS module, based on the uBlox-6 chipset, has an antenna on board and connects to your system via TTL serial. The 1Hz update rate is fast enough for the majority of applications (and can be increased to 5Hz if you need) so whether you're tracking a pet or building an autonomous car, the GP-635T has you covered. [17]

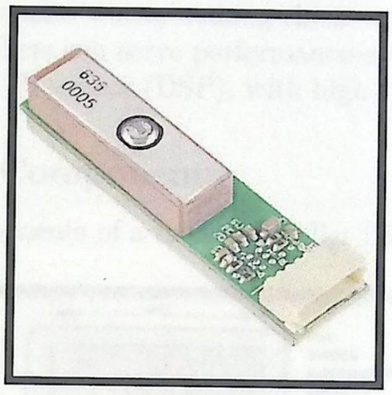


Figure 2.7: GP-635T GPS Receiver

The features of this device are:

- carrying the "T" option of this module which is TTL-UART and not USB.
- This module works with 6-pin 1mm pitch JST type cables and connectors which you can find in the related items below.
- Dimensions: 35 x 8 x 6.5 mm.

In the proposed design, GPS will be used in this system to determine the location of the accident happened. When the shortest path is specified, the driver will have a navigator device to show him the path to be followed.

It is also used in determining the location of the cars as nodes on the roads, to locate the nearest car to control the accident.

2.6 Microcontroller

A microcontroller is the type of small computer on a single integrated circuit. Which contains a processor core, programmable input/ output and memory peripherals? Often program memory is in the form of OTP ROM or NOR flash included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for enclosed applications, in contrast to the microprocessors which are used in personal computers and other applications.[18]

Microcontrollers are being used in automatically controlled products and accessories, as in automobile engine control systems, implantable medical devices, remote controls, appliances, office machines, toys and power tools. By minimizing the cost and size compared to a design that uses different microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes.

Microcontrollers may use Four-bit words and operate at clock rate frequencies as low as 4000 HZ, for low power consumption. Generally, microcontrollers have the ability to withhold functionality while waiting for an event such as a button press or other interrupt; while sleeping, power consumption may be just nano watts, making them well suited for long lasting battery applications. Other microcontrollers can serve performance-critical roles, where they may need to act more like a Digital Signal Processor (DSP), with high speed and power consumption.

2.6.1 Microcontroller Components

Figure 2.8 shows the basic components of a microcontroller [19].

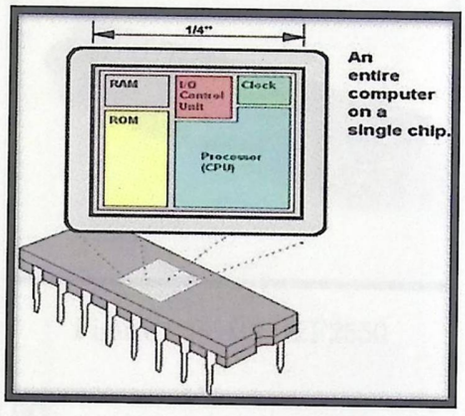


Figure 2.8: Microcontroller Component

- A processor is the logic circuitry that responds to and processes the basic instructions that drive a computer, the term processor has generally replaced the term central processing unit (CPU). The processor in a personal computer or embedded in small devices is often called a microprocessor.

- Memory is the electronic holding place for instructions and data that your computer's microprocessor can reach quickly. When your computer is in normal operation, its memory usually contains the main parts of the operating system and some or all of the application programs and related data that are being used. Memory is often used as a shorter synonym for random access memory (RAM). This kind of memory is located on one or more microchips that are physically close to the microprocessor in your computer. The more RAM you have, the less frequently the computer has to access instructions and data from the more slowly accessed hard disk form of storage.

- A peripheral is any computer device that is not part of the essential computer (the processor, memory, and data paths) but is situated relatively close by. A near synonym is input/output (I/O) device. Some peripherals are mounted in the same case with the main part of the computer as are the hard disk drive, CD-ROM drive, and NIC. Other peripherals are outside the computer case, such as the printer and image scanner, attached by a wired or wireless connection.

2.6.2 Types of Microcontrollers

PIC18F2550

PIC18F2550, as shown in figure 2.9, provides high computational performance at an economical price, with the addition of high endurance, enhanced flash program memory. In addition it introduces design enhancements. [20]

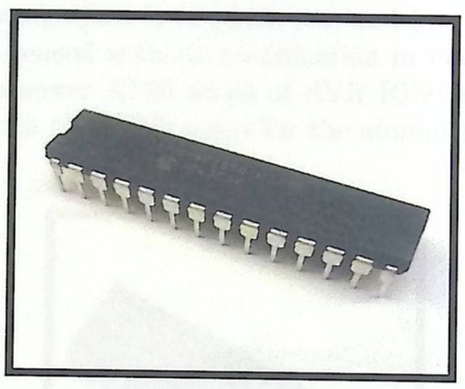


Figure 2.9: PIC18F2550

The features of this device are:

- High-Performance.
- 28-pin Low Power Microcontroller.
- Flash Program Memory: 32 Kbytes.
- SRAM Data Memory: 2048 bytes.
- I/O Pins: 24.
- Timers: One 8-bit / three 16-Bit.
- USB: USB V2.0 Compliant with High and Low Speed Operation.
- Enhanced USART: RS-485, RS-232 and LIN 2.0.

Arduino Uno

The Arduino Uno; is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.[21]

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

AT89S52- 8 bit Microcontroller

The AT89 series as shown in figure 2.10; remains very popular as general purpose microcontrollers, due to their industry standard instruction set, and low unit cost. This allows a great amount of legacy code to be reused without modification in new applications. While considerably less powerful than the newer AT90 series of AVR RISC microcontrollers, new product development has continued with the AT89 series for the aforementioned advantages.[22]

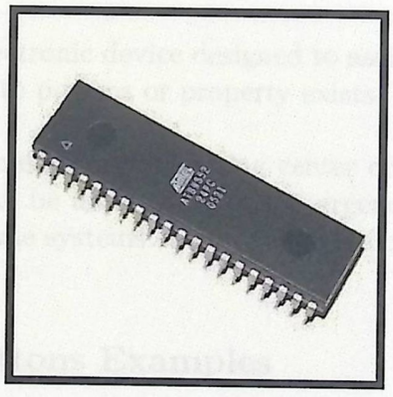


Figure 2.10: AT89S52-8bit Microcontroller

The features of this device are:

- Compatible with MCS-51 Products.
- 8K Bytes of In-System Reprogrammable Flash Memory.
- Fully Static Operation: 0 Hz to 33 MHz.
- Three-level Program Memory Lock.
- 256 x 8-bit Internal RAM.

- 32 Programmable I/O Lines.
- Three 16-bit Timer/Counters.
- Programmable Serial Channel.
- Low-power Idle and Power-down Modes.
- 4.0V to 5.5V Operating Range.
- Full Duplex UART Serial Channel.
- Interrupt Recovery from Power-down Mode.
- Fast Programming Time.
- Flexible ISP Programming (Byte and Page Mode)

In this design, microcontroller will take the data from the GPS receiver and then sends the information to the station by using GPRS with help of GSM modem.

2.7 Emergency Buttons

An emergency button is an electronic device designed to assist in alerting somebody in emergency situations where a threat to persons or property exists.

These buttons can be connected to a monitoring center or locally via a silent alarm or an audible bell/siren. The alarm can be used to request emergency assistance from local security, police or emergency services. Some systems can also activate closed-circuit television to record or assess the event.

2.7.1 Alarm Panic Buttons Examples

- A button in a critical system (such as a nuclear weapons system) used to quickly activate an extreme measure to mitigate an emergency situation.
- A red button integral to key fobs which activates a car alarm's siren.
- A device given to elderly individuals in order to maintain their independence outside of an aged Care Facility, while still affording them a means of summoning help should they require it (i.e. a medical emergency that renders them immobile, like a fall, injury or illness). Such a device can also be referred to as an Emergency Medical Alert (EMA) button and can be fitted as either a pendant or bracelet to be worn by the user. MAB's are usually wirelessly connected to a call centre. When the alarm is raised, an operator will call the individual's home to ensure a false alarm has not occurred; if there is no answer, the operator will alert either family members, emergency services, or both.

- A button similar to the above, which is used indoors in self-sufficient houses for elderly people, where it alerts someone inside the house, who will then first check for a false alarm by phoning the person, and if there is no false alarm, will enter the person's flat to check what is the problem.

- A button used in convenience stores, gas station, or other establishments staffed with a single employee during late hours. Often located under the counter near the cash register or safe, the button can be pressed in times of distress (Such as robbery, disruptive or threatening behavior, or a situation which may warrant assistance), triggering a silent alarm. If the button alarms a private security company, a fee may be charged for each time the button is used. This prevents mis-use, and often aids in the employees judgment of the situation; whether or not it warrants the fee to have help to deal with the situation.[23]

Panic buttons can be the wired type or wireless type. Wired panic buttons are connected using cabling. Wired panic buttons are very reliable, but it can sometimes be difficult and costly to get cable to each panic button location.

Wireless panic buttons use a short-range radio transmitter, similar to that used with a garage door opener. Wireless panic buttons are easy to install and can be placed just about anywhere. Wireless panic buttons can also be carried around on the premises by employees if desired. Wireless panic buttons require batteries and need to be tested frequently to assure that they are operating properly.

Panic Buttons Are Faster Than Cell Phones; panic buttons need to be small, wireless, and easily accessible to be useful to everyone. They can activate an audible or silent alarm as soon as an intruder or threat is encountered. Although dialing the emergency number is easy on a cell phone, it takes some time to place the call and can alert an intruder. Panic buttons are often kept in a convenient pocket, on a belt loop, or even around the neck, and a single push initiates the call for help.

2.8 Mobile Phone (Cellular Phone)

Mobile phone is one of the most sophisticated and modern techniques that we use, it transformed from simple use to make calls, to be used in many complex useful applications nowadays.

Many updates and developments are quickly continued on mobile phones, and the updates still until now, mobile phones become a very smart, efficient, and small device that puts the whole world in your hands.[24]

2.8.1 Mobile Operating System (Mobile Software Platform)

Like Microsoft windows, Apple's Macintosh, and Linux, and many other desktop operating systems, the development that has occurred on mobile phones makes us more control on the

mobile devices, application, and services, with their special operating systems as the computer operating systems.
Most common mobile operating systems:

- Android from Google Inc.(open source, Apache).
- Symbian OS from Symbian Foundation.(open public license).
- Windows phone from Microsoft (closed source, Proprietary).
- BlackBerry OS from RIM (closed source, Proprietary).

These operating systems and many other ones, representing the operating systems for our mobile phones, and by using the appropriate programming language for each operating system, we become enable to program these phones, so we can build a new and different applications and services we want from these mobiles.

In this system Android is the operating system that we want to use in order to support our project objective requirements .[25]

Android Operating System

In recent years Open Handset Alliance (OHA) led by Google has developed neat, versatile, powerful and elegant platform which was android. This platform spread widely and its market growing quickly, because it is supported by large set of hardware, software, and network carriers.

Android is open source and Google releases the code under the Apache License(open source license), Android operating system is a stack of software components which is roughly divided into five sections and four main layers: Linux kernel, Libraries, Android runtime, Application Framework, Applications.[26]

Android Architecture

The architecture of Android operating system is described as following figure 2.11, with briefly explains [26]

mobile devices, application, and services, with their special operating systems as the computer operating systems.
Most common mobile operating systems:

- Android from Google Inc.(open source, Apache).
- Symbian OS from Symbian Foundation.(open public license).
- Windows phone from Microsoft (closed source, Proprietary).
- BlackBerry OS from RIM (closed source, Proprietary).

These operating systems and many other ones, representing the operating systems for our mobile phones, and by using the appropriate programming language for each operating system, we become enable to program these phones, so we can build a new and different applications and services we want from these mobiles.

In this system Android is the operating system that we want to use in order to support our project objective requirements .[25]

Android Operating System

In recent years Open Handset Alliance (OHA) led by Google has developed a powerful and elegant platform which was android. This platform spread widely growing quickly, because it is supported by large set of hardware, software,

Android is open source and Google releases the code under the Apache license), Android operating system is a stack of software components into five sections and four main layers: Linux kernel, Libraries, Framework, Applications.[26]

Android Architecture

The architecture of Android operating system is described explains [26]

called Dalvik Virtual Machine which is a and optimized for Android. The Dalvik memory management and multi-threading, which the Dalvik VM enables every Android application to instance of the Dalvik virtual machine. The Android libraries which enable Android application developers using standard Java programming language.

work layer provides many higher-level services to applications in Application developers are allowed to make use of these services

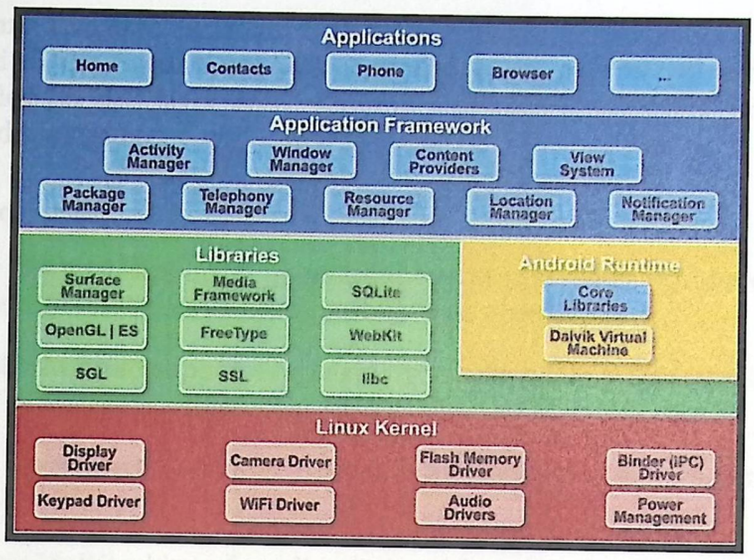


Figure 2.11: Android Architecture

- **Linux kernel**
  At the bottom of the layers is Linux - Linux 2.6 with approximately 115 patches. This provides basic system functionality like process management, memory management, device management like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

- **Libraries**
  On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

- **Android Runtime**
  This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android. The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine. The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

- **Application Framework**
  The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

- **Applications**
  You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, and Games etc.

Android Features

There are many features supported by android operating system, and here we mention some of them which give a helpful point for our system objective [27] :

- **Connectivity**:Android supports connectivity technologies including GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
- **Additional hardware support**:Android can use video/still cameras, touch screens, GPS, etc.
- **Multitasking**:Multitasking of applications, with unique handling of memory allocation, is available.
- Android supports capturing a screenshot by pressing the power and volume-down buttons at the same time.

this project, a mobile phone will be used as mainly device in this project; in order to download the map in the emergency cars (ambulance, police and fire) then choose the shortest path to reach the accident location as fast as possible.

2.8.2 Programming language

Servers in stations will be programmed using Java programming language where it is an important language, that is created by James Gosling from Sun Microsystems (Sun) in 1991. The first publicly available version of Java (Java 1.0) was released in 1995.

Sun Microsystems was acquired by the Oracle Corporation in 2010. Oracle has now the steermanship for Java. Over time new enhanced versions of Java have been released. The current version of Java is Java 1.7 which is also known as Java 7.

From the Java programming language the Java platform evolved. The Java platform allows software developers to write program code in other languages than the Java programming language and still runs on the Java virtual machine. The Java platform is usually associated with the Java virtual machine and the Java core libraries.

Java Architecture

Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended

to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2014, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers.

Like any programming language, the Java language has its own structure, syntax rules, and programming paradigm. The Java language's programming paradigm is based on the concept of object-oriented programming (OOP), which the language's features support.

The Java language is a C-language derivative, so its syntax rules look much like C's: for example, code blocks are modularized into methods and delimited by braces (and), and variables are declared before they are used.[28]

TCP/IP Sockets in Java

A socket is an abstraction through which an application may send and receive data, in much the same way as an open file handle allows an application to read and write data to stable storage. A socket allows an application to plug in to the network and communicate with other applications that are plugged in to the same network. Information written to the socket by an application on one machine can be read by an application on a different machine and vice versa. Different types of sockets correspond to different underlying protocol suites and different stacks of protocols within a suite. This book deals only with the TCP/IP protocol suite. The main types of sockets in TCP/IP today are stream sockets and datagram sockets. Stream sockets use TCP as the end-to-end protocol (with IP underneath) and thus provide a reliable byte-stream service. A TCP/IP stream socket represents one end of a TCP connection. Datagram sockets use UDP (again, with IP underneath) and thus provide a best-effort datagram service that applications can use to send individual messages up to about 65,500 bytes in length. [29]

TCP Sockets

Java provides two classes for TCP: Socket and ServerSocket. An instance of Socket represents one end of a TCP connection. A TCP connection is an abstract two-way channel whose ends are each identified by an IP address and port number. Before being used for communication, a TCP connection must go through a setup phase, which starts with the client's TCP sending a connection request to the server's TCP. An instance of ServerSocket listens for TCP connection requests and creates a new Socket instance to handle each incoming connection. Thus, servers handle both ServerSocket and Socket instances, while clients use only Socket.

We will use PC in emergency stations as mainly device in our project; in order to determine the car's location permanently using GPRS then determine the closest one to the accident location.

2.9 Shortest Path Algorithms

Heuristic refers to experience-based techniques for problem solving, learning, and discovery that give a solution which is not guaranteed to be optimal. Where the exhaustive search is impractical, heuristic methods are used to speed up the process of finding a satisfactory solution via mental shortcuts to ease the cognitive load of making a decision. Examples of this method include using a rule of thumb, an educated guess, an intuitive judgment, stereotyping, or common sense.

In more precise terms, heuristics are strategies using readily accessible, though loosely applicable, information to control problem solving in human beings and machines.

In engineering, a heuristic is an experience-based method that can be used as an aid to solve process design problems, varying from size of equipment to operating conditions. By using heuristics, time can be reduced when solving problems. Several methods are available to engineers. These include Failure mode and effects analysis and Fault tree analysis. The former relies on a group of qualified engineers to evaluate problems, rank them in order of importance and then recommend solutions. The methods of forensic engineering are an important source of information for investigating problems, especially by elimination of unlikely causes and using the weakest link principle. Because heuristics are fallible, it is important to understand their limitations. They are aids that facilitate quick estimates and preliminary process designs.[30]

2.9.1 Algorithms

Dijkstra's Algorithm

Dijkstra's algorithm is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms.

For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. For example, if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path first is widely used in network routing protocols, most notably IS-IS and OSPF (Open Shortest Path First).[31]

A* search Algorithm

A* uses a best-first search and finds a least-cost path from a given initial node to one goal node (out of one or more possible goals). As A* traverses the graph, it follows a path of the

lowest expected total cost or distance, keeping a sorted priority queue of alternate path segments along the way.[32]

It uses a knowledge-plus-heuristic cost function of node x (usually denoted f(x)) to determine the order in which the search visits nodes in the tree shown in figure 2.12; the cost function is a sum of two functions:

- the past path-cost function, which is the known distance from the starting node to the current node x (usually denoted g(x)).
- A future path-cost function, which is an admissible "heuristic estimate" of the distance from x to the goal (usually denoted h(x)).

The h(x) part of the f(x) function must be an admissible heuristic; that is, it must not overestimate the distance to the goal. Thus, for an application like routing, h(x)might represent the straight-line distance to the goal, since that is physically the smallest possible distance between any two points or nodes.

If the heuristic h satisfies the additional condition for every edge (x, y) of the graph (where d denotes the length of that edge), then his called monotone, or consistent. In such a case, A* can be implemented more efficiently-roughly speaking, no node needs to be processed more than once (see closed set below) and A* is equivalent to running Dijkstra's algorithm with the reduced cost

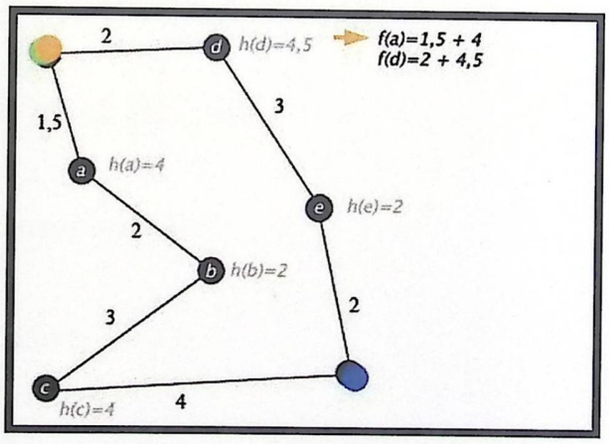


Figure 2.12: A* Algorithm tree

2.10 Application programming interface (API)

API is a tool in SDK manager, it can be added to android library, it will be used to build shortest path algorithm. Application programming interface (API) specifies how some software components should interact with each other. In addition to accessing databases or computer

hardware, such as hard disk drives or video cards, an API can be used to ease the work of programming graphical user interface components. In practice, many times an API comes in the form of a library that includes specifications for routines, data structures, object classes, and variables.[33]

An API specification can take many forms, including an International Standard such as POSIX, vendor documentation such as the Microsoft Windows API, the libraries of a programming language, e.g., Standard Template Library in C++ or Java API.

3

Chapter Three

Conceptual Design

Chapter 3

Conceptual Design

This chapter describes the system main parts and the design concepts in some details; it will talk about system general block diagram, the system main components, system main flow chart.

3.1 General System Block Diagram

As shown in figure 3.1, the general block diagram; the main parts of the system are: GSM network which is the connection medium through which data will be transferred, GPS receiver, where data and coordinates will be created, microcontroller which is the interface between GPS and GSM/GPRS modems, Servers and android devices where the data will be received and processed.

The main components of the system are the GPS Navigator (Samsung galaxy phone), microcontroller, GSM modem and server. This section will describe all of them in more details and the options for these components will be described.

3.1.1 GPS Receiver

SIM908

In this project, at the transmitter side, SIM908 Arduino Shield is used to determine the location of the accident, as shown in figure 3.2 .

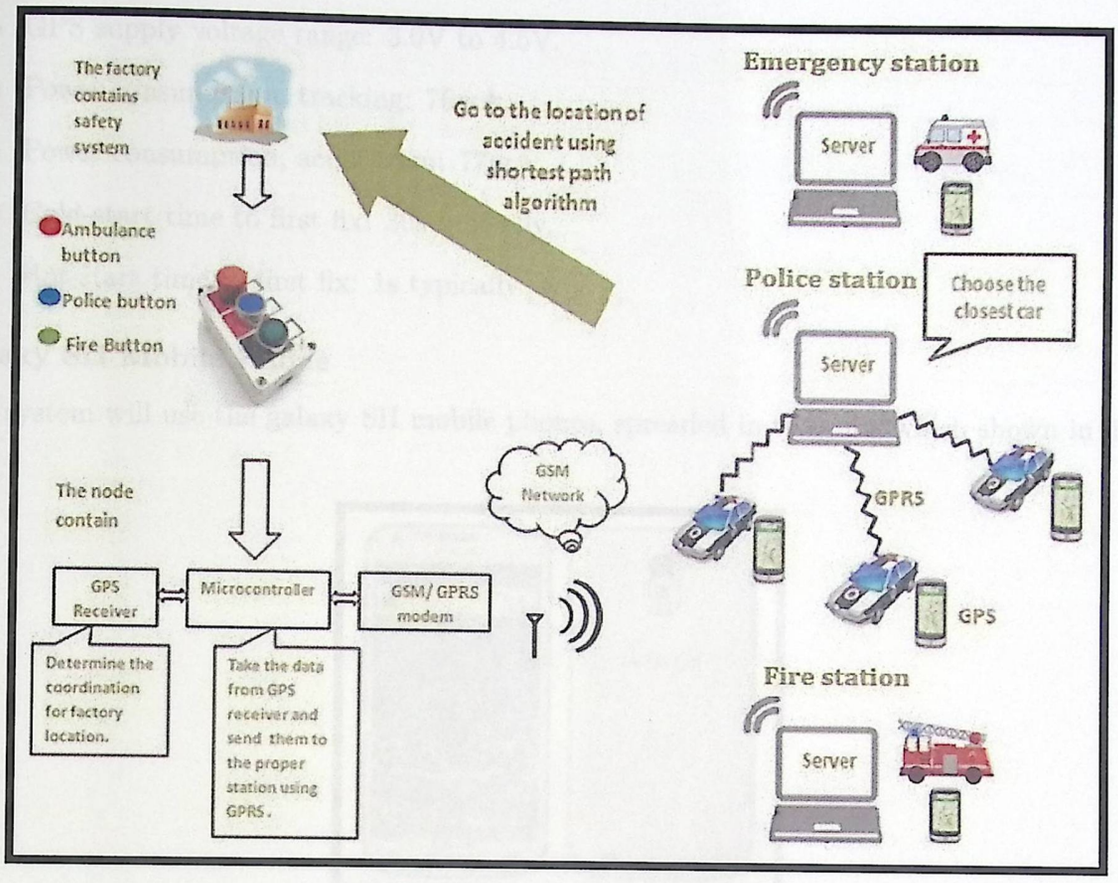


Figure 3.1: General Block Diagram

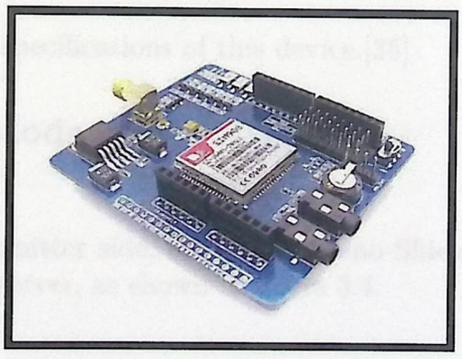


Figure 3.2: SIM 908 Arduino Shield

The features of this device are:

- 42-channel receiver.
- Tracking sensitivity: -160 dBm.

- GPS supply voltage range: 3.0V to 4.5V.
- Power consumption, tracking: 76mA.
- Power consumption, acquisition: 77mA.
- Cold-start time to first fix: 30s typically.
- Hot-start time to first fix: 1s typically.[34]

Galaxy SII Mobile Phone

The system will use the galaxy SII mobile phones, spreaded in the cars, which shown in figure 3.3.

Figure 3.3: Galaxy SII mobile

Table 3.4 shows the main specifications of this device.[35]

3.1.2 GSM/GPRS Modem

SIM908 Arduino Shield

In this project, at the transmitter side, SIM908 Arduino Shield is used to send the location of the accident to the proper server, as shown in figure 3.4.

Table 3.1: Galaxy SII Mobile phone Specifications

Specifications	
CPU	Dual-core,1.5 GHz Scorpion
OS	Android,OS, v2.3.5 (Gingerbread), upgradable to v4.1.2 (Jelly Bean)
General	2G Network,GSM 850 / 900 / 1800 / 1900 3G Network,HSDPA 850 / 1700 / 1900 / 2100
SIM Card	Mini-SIM
Connectivity	GPRS/EDGE/WIFI/BLUETOOTH
GPS	with,A-GPS support
Memory	Card slot micro SD, up to 32 GB, 8 GB included,Internal,16/32 GB storage, 1 GB RAM
WLAN	Wi-Fi,802.11 a/b/g/n, dual-band, DLNA, Wi-Fi Direct, Wi-Fi
3G	BUILT IN,SUPPORT PHONE CALLS AND 3 G NETWORK
Data Transfer	USB cable/ Bluetooth
Speed	HSDPA,,42 Mbps; HSUPA, 5.76 Mbps

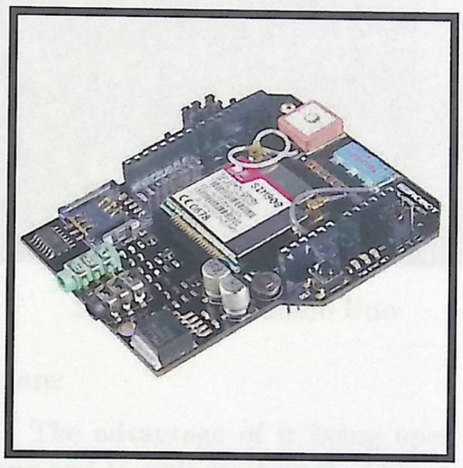


Figure 3.4: SIM 908 Arduino Shield

The features of this device are:

- Quad-band: 850/900/1800/1900 MHz.
- GPRS multi-slot class 10.
- Controlled via AT commands (GSM 07.07, 07.05 and SIMCom Enhanced AT Commands).
- GPRS supply voltage range: 3.2V to 4.8V.
- Low power consumption.
- Operating temperature: -40°Cto +85°C.

- Serial interface and debug interface for GSM/GPRS.
- Debug interface for GPS NMEA output.
- Separate U.FL antenna connectors: one for GSM/GPRS and one for GPS.[34]

3.1.3 Arduino Uno

In this project, at the transmitter side, Arduino Uno is used to interact with SIM908 shield, as shown in figure 3.5.

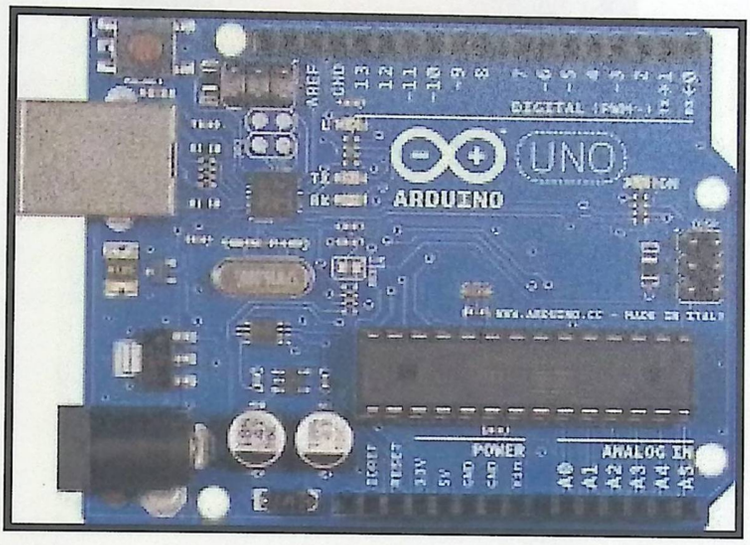


Figure 3.5: Arduino Uno

The features of this device are:

- An open source design. The advantage of it being open source is that it has a large community of people using and troubleshooting it. This makes it easy to find someone to help you debug your projects.
- An easy USB interface. The chip on the board plugs straight into your USB port and registers on your computer as a virtual serial port.
- Very convenient power management and built-in voltage regulation. You can connect an external power source of up to 12v and it will regulate it to both 5v and 3.3v. It also can be powered directly off of a USB port without any external power.
- An easy-to-find and relatively cheap.
- A 16 MHz clock. This makes it not the speediest microcontroller around, but fast enough for most applications.
- A button to reset the program on the chip.[36]

3.1.4 Panic Buttons in Our Project

This type of buttons as shown in figure 3.6; will be used in the system in order to let the person to activate it when he or she needs help. Therefore, the alarm reaches the station to give the quick help to the calling location.[37]

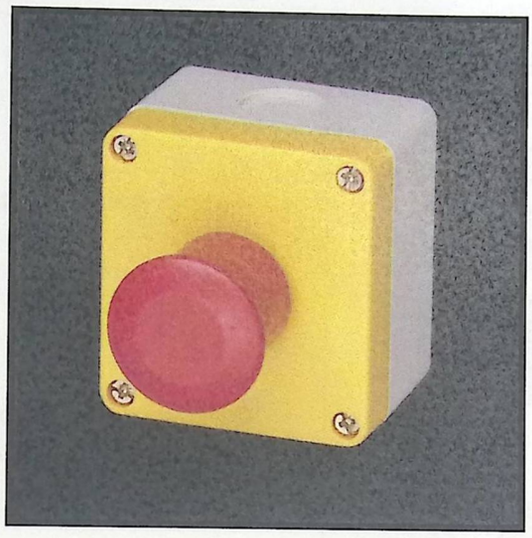


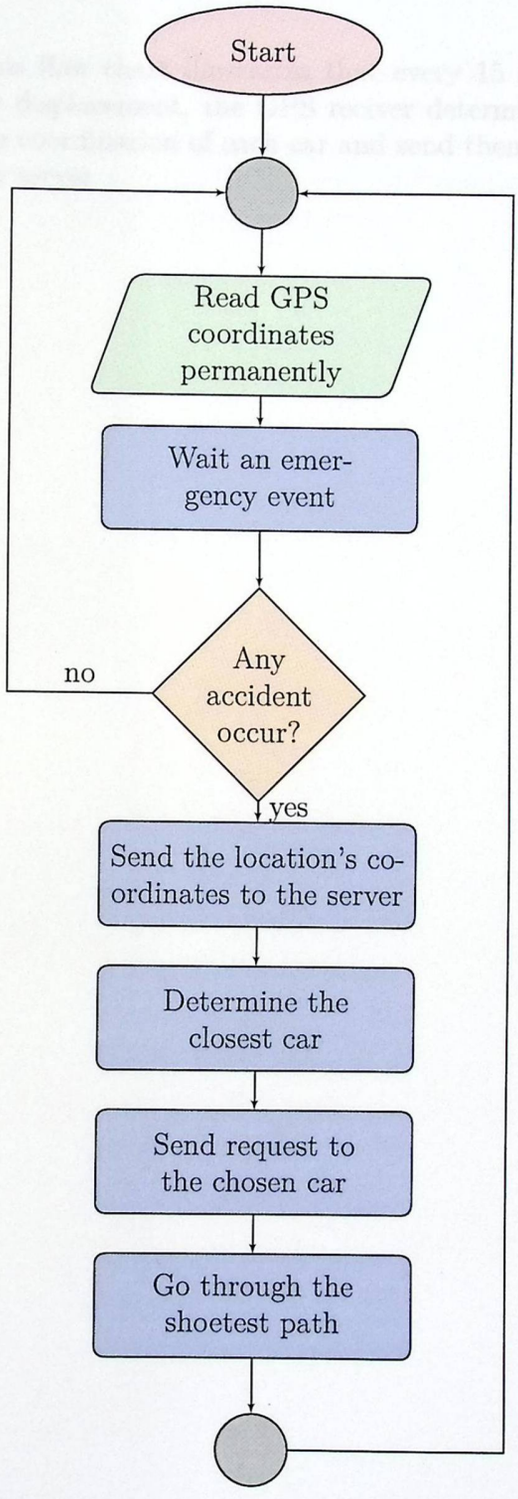
Figure 3.6: Panic button

3.2 System Flow Chart

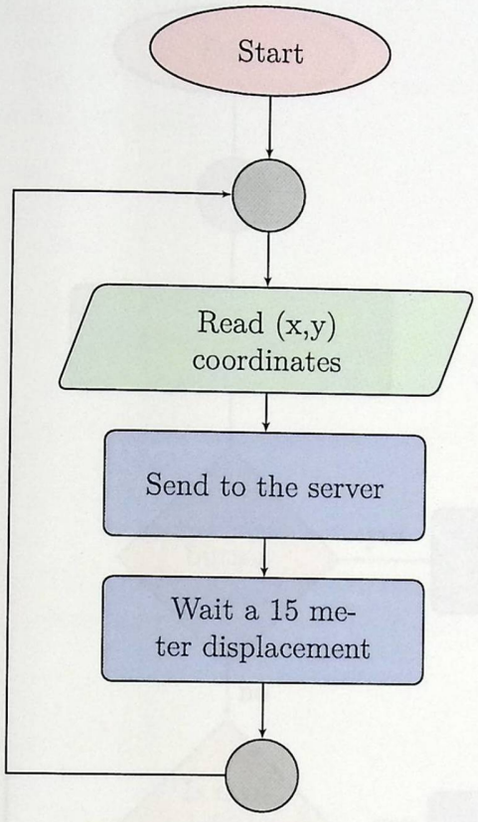
This flow chart illustrates the system main operation.

The following steps show the details of flowchart entities:

- Track emergency cars location permanently by the server.
- The appropriate button will be pressed if any accident happens.
- GSM/GPRS network deliver the location determined by GPS receiver.
- The server will determine the closest emergency car to the calling side.
- A request will be sent to the chosen car.
- The chosen car will follow the map provided by the android navigator device to reach the accident location through the shortest path.

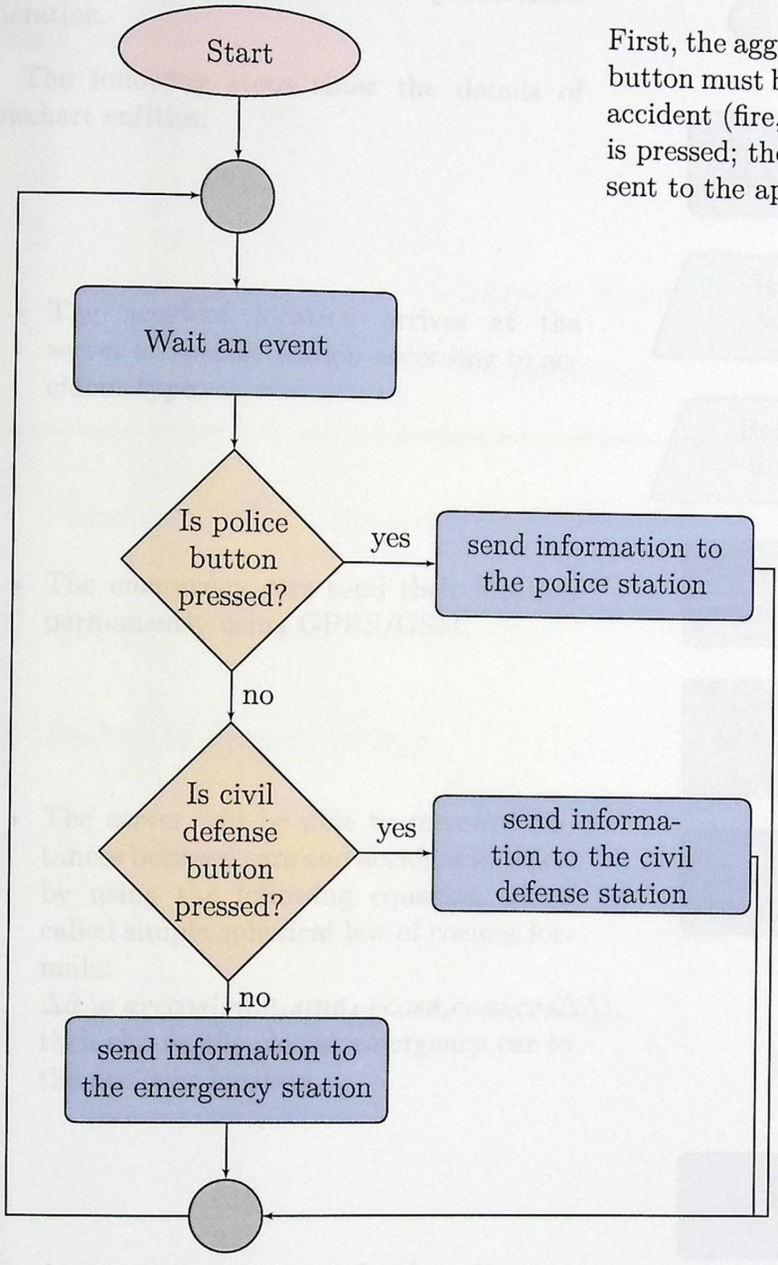


3.2.1 Tracking System



This flow chart illustrates that every 15 meter displacement, the GPS reciver determines the coordination of each car and send them to the server.

3.2.2 Transmitter Response

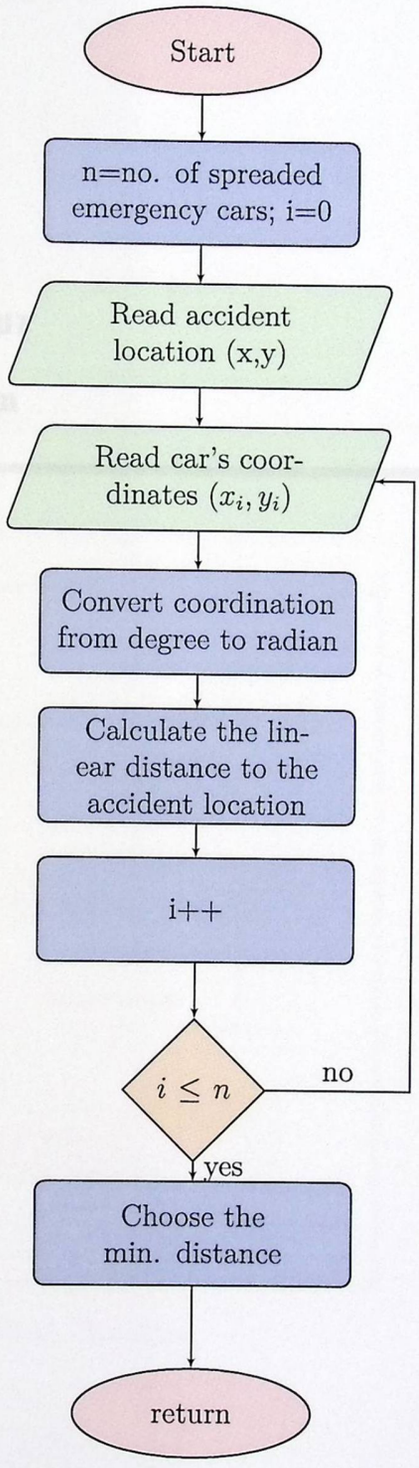


First, the aggrieved person should decide which button must be pressed according to the kind of accident (fire, theft, injures). when the button is pressed; the accident location information is sent to the appropriate server.

3.2.3 Server Response

This flow chart illustrates the system main operation.

The following steps show the details of flowchart entities:



- The accident location arrives at the server at specific station according to accident type.

- The emergency cars send their location permanently using GPRS/GSM.

- The server will be able to measure distances between cars and accident location by using the following equation which called simple spherical law of cosines formula:
  $\Delta\hat{\sigma} = arccos(sin\emptyset_s sin\emptyset_f + cos\emptyset_s cos\emptyset_f cos\Delta\Lambda)$,
  then choose the closest emergency car to the accident location.

- At the chosen car, an android application will be programmed using Google map to determine the shortest path to reach the accident and provide help as fast as possible.

4

Chapter Four

Detailed Design

4.1 Functional Block Diagram

4.2 Hardware System Design

4.3 Software System Design

4.4 AT Command for GPRS and GPS

Chapter 4

Detailed Design

After explaining the main block diagram and flowchart of the system in the previous chapter, this chapter explains the specific details of the hardware and pseudo code of the system implementation.

Figure 4.1 shows in details the interconnection between the system elements.

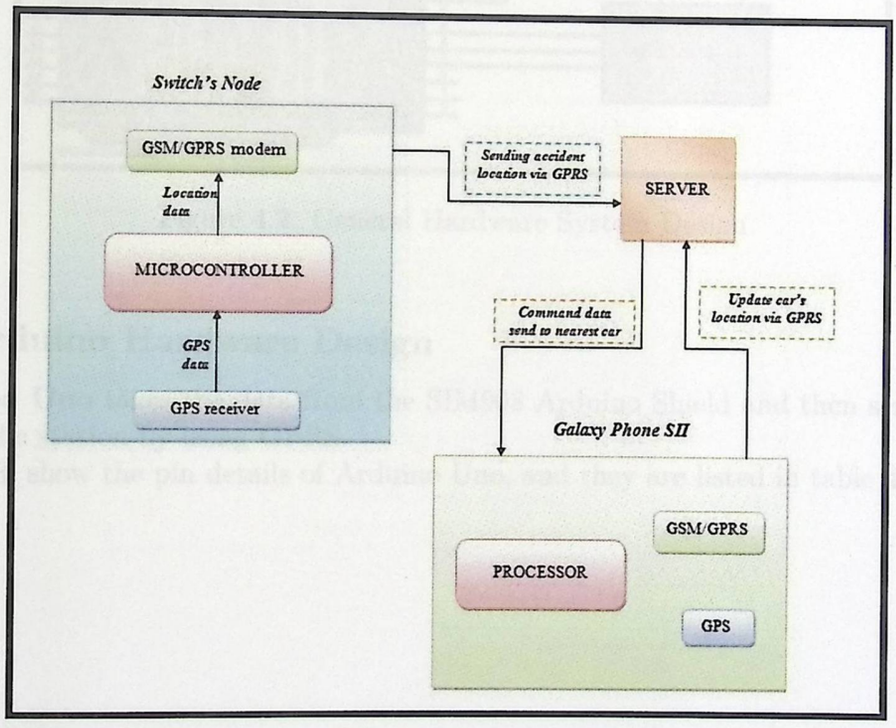


Figure 4.1: Functional Block Diagram

4.1 Hardware System Design

Figure 4.2 represents the general pin connection for the integrated circuits that will be used in this project.

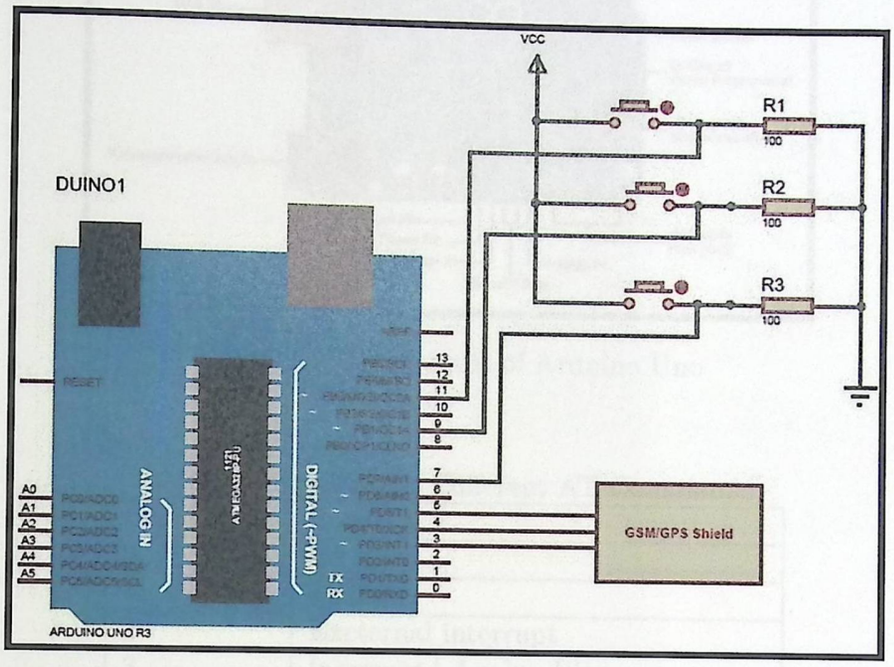


Figure 4.2: General Hardware System Design

4.1.1 Arduino Hardware Design

The **Arduino Uno** takes the data from the SIM908 Arduino Shield and then sends the coordination to the station by using GPRS.

Figure 4.3; show the pin details of Arduino Uno, and they are listed in table 4.1.[38]

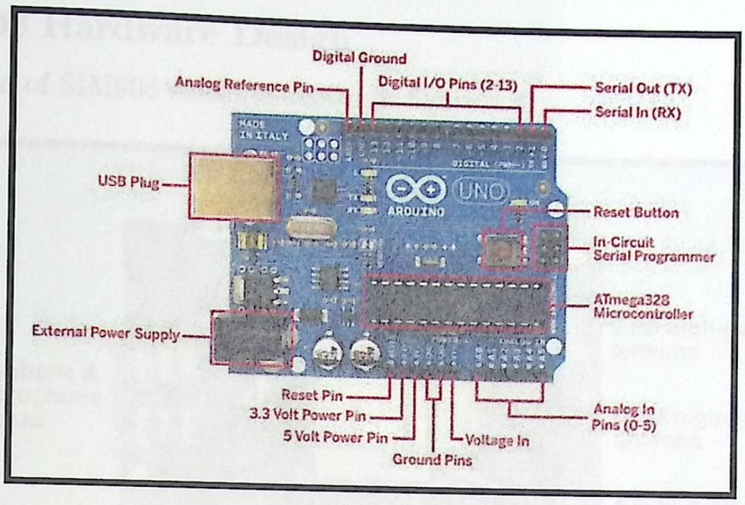


Figure 4.3: Pin Details of Arduino Uno

Table 4.1: Response of Different AT Commands

Pin number	Function
0	Rx
1	Tx
2	Exeternal interrupt
3	Interrupt/ Analog Pin
4, 7, 8	NC
5, 6, 9	Analog Pin (PWM)
10	SS: slave select/ Analog pin
11	MOSI: Master In Slave Out,Analog pin
12	MISO: Slave In Master Out
13	Built in Led / SCK serial clock

4.1.2 SIM908 Hardware Design

The pin connection of SIM908 shield is shown in figure 4.4.

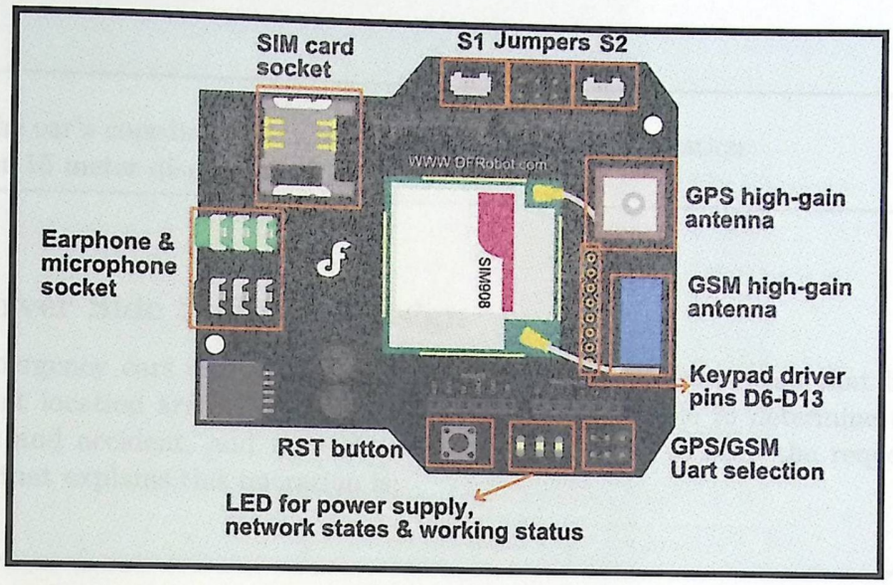


Figure 4.4: Pins Connection of SIM908 shield

4.2 Software System Design

4.2.1 Transmitter Side Software Design

First, wait the accident event. If accident happens, the person should choose the appropriate switch according to the kind of accident (fire, theft, injures). Then the information about the accident location is sent to appropriate station. The pseudo code that explains this operation is:

Wait event ...

if $S1 = 1$ **then**
    Send data to Police station
else
    if $S2 = 1$ **then**
        Send data to Emergency station
    else
        if $S3 = 1$ **then**
            Send data to Fire Brigade station

4.2.2 Tracking System

The emergency cars send their location permanently to the server, the pseudo code for this operation is:

---

do

Send the car's coordination (latitude, longitude) for it's location

while after 15 meter displacement

4.2.3 Server Side Software Design

First, the emergency cars send their location permanently to the station that belongs to it. When accident location arrives to station, the server will be able to determine the distances between cars and accident, and then choose the shortest one to send the request to it, the pseudo code that explains this operation is:

Start run ..
Read the location of accident as (s,f)
Read cars location as (x_i, y_i)
n= number of cars on the street

for $(i = 0; i < n; i++)$ **do**
{
Convert (s, f, x_i, y_i) from degree to angular using equation:
$\Delta\hat{\sigma} = arccos(sin\emptyset_s sin\emptyset_f + cos\emptyset_s cos\emptyset_f cos\Delta\Lambda)$
determine the distance between each car and the sender side
Determine the min.
Send request to the chosen car
}

4.2.4 Shortest Path Algorithm

Dijkstra algorithm will be used to determine the shortest path for the chosen car to reach the location of the accident. By applying this algorithm, the street will be divided into sub segments, and a node (vertex) will be put at every crossroad. The cost will be the distance of the segment in the algorithm.[39] The following map illustrates the idea.

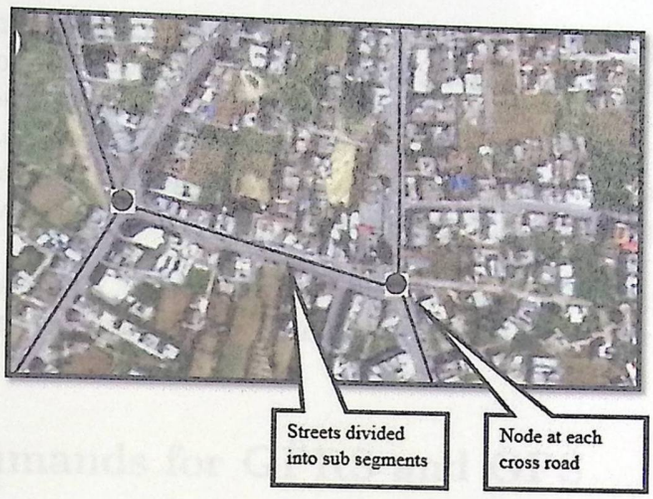


Figure 4.5: Illustration map

The pseudo code of Dijkstra algorithm is:

$Q = 0$ ▷ first, Q is empty vertices group
Define u as the source node
$Q = u$ ▷ put u in Q group.
For all nodes v
{
if v is neighboring of u **then**
$D(v) = d(u, v)$ ▷ determine the distance between neighbors u and v
else
$D(v) = \infty$ ▷ distance $= \infty$ if node v is not a neighbor
do
Find w not in Q such that $D(w)$ is the min.
Add w to Q
Update $D(v)$ for each neighbor v of w and not in Q
$D(v) = min.[D(v), D(w) + d(w, u)]$ ▷ choose the min. distance and calculate it
while $Q \neq v$
}

Google Maps Android API

With the Google Maps Android API, you can add maps to your app that are based on Google Maps data. The API automatically handles access to Google Maps servers, data downloading, map display, and touch gestures on the map. You can also use API calls to add markers, polygons and overlays, and to change the user's view of a particular map area.

The key class in the Google Maps Android API is Map View. A Map View displays a map with data obtained from the Google Maps service. When the Map View has focus, it will capture keypresses and touch gestures to pan and zoom the map automatically; including handling network requests for additional maps tiles. It also provides all of the UI elements necessary for users to control the map. Your application can also use Map View class methods to control the map programmatically and draw a number of overlays on top of the map.

The Google Maps Android APIs are not included in the Android platform, but are available on any device with the Google Play Store running Android 2.2 or higher, through Google Play services.[33]

4.3 AT Commands for GPRS and GPS

In this system, GPRS/GPS Modem are the devices that involve machine-to-machine communication, AT commands will be used to control the GPRS/GPS modem.

AT commands (are used to control MODEMs). AT is the abbreviation for Attention. These commands come from Hayes commands that were used by the Hayes smart modems. The Hayes commands started with AT to indicate the attention from the MODEM. The dial up and wireless MODEMs (devices that involve machine to machine communication) need AT commands to interact with a computer.
AT commands with GSM/GPRS MODEM or mobile phone can be used to access the following services [40]:

1. Data and voice link over mobile network.
2. SMS service
3. MMS service
4. Fax service

Table 4.2 shows the types of AT command and responses.

Table 4.2: Response of Different AT Command

Test command	AT+<T>=?	The Mobile Equipment Returns The List Of Parameters And Value Ranges Set.
Read command	AT+<T>?	Return The Currently Set Value Of The Parameter.
Write command	AT+<T>=<..>	This Command Sets The User-Definable ParameterValues.
Execution command	AT+<T>	The execution command,reads non-variable parameter affected by internal processes in the GSM engine.

AT commands used for GPS service activation is shown in table 4.3.

Table 4.3: AT Commands for GPS Support

Command	Description
AT+CGPSIPR=9600	Set the baud rate
AT+CGPSPWR=1	Turn on GPS power supply
AT+CGPSRST=1	Reset GPS in autonomy mode
AT+CGPSINF=0	Get the coordination

AT commands used for GPRS service activation is shown in table 4.4.

Table 4.4: AT Commands for GPRS Support

Command	Description
AT+CGATT = 1	Attach or Detach From GPRS Service
AT+CDGCONT=1,"IP","Wataniya_internet"	Define PDP Context
AT+CGACT = 1,1	PDP Context Activate Or Deactivate
AT+CIFSR	Get local IP address
AT+CSTT= "Wataniya_internet"	Define APN
AT+CIICR	Bring GPRS connection
AT+CIPSTART="TCP","195.3.191.27","12346"	Define server's public IP & port
AT+CIPSEND	Send the data

5

Chapter Five

System Implementation

5.1 Introduction

5.2 Hardware System Implementation

5.3 Software Implementation

Chapter 5

System Implementation

This chapter describes the hardware and software implementation of the system. The system can be divided into two main parts: first is the implementation of the software functions, objects, and methods: at arduino microcontroller, server, and mobile. The second part is the implementation of an entire interconnected set of components and software for the purpose of determining proper functions and achieving the desired goals of the system.

5.1 Hardware System Implementation

Figure 5.1 shows the hardware components before connected together. SIM908 Arduino Shield used to Capture GPS coordinates and sends them via GPRS using the data SIM (WATANIYA). USB cable used to connect Arduino Uno with the computer for programming, while Arduino Uno is connected with SIM908 and switches. The android application runs on a Galaxy phone. The computer is the server which receives the coordination from Galaxy phone and SIM908 .

Figure 5.1: Electrical Subsystem Components

Figure 5.2 illustrates the Arduino Uno and shield connection with the three switches. Each switch is connected with a resistor for protection and a led to ensure connectivity. The combination between Microcontroller and shield was programed such as every switch alarms for the

appropriate emergency case. The digital pin number 3 is for GSM communication, and pin 4 is for the GPS.

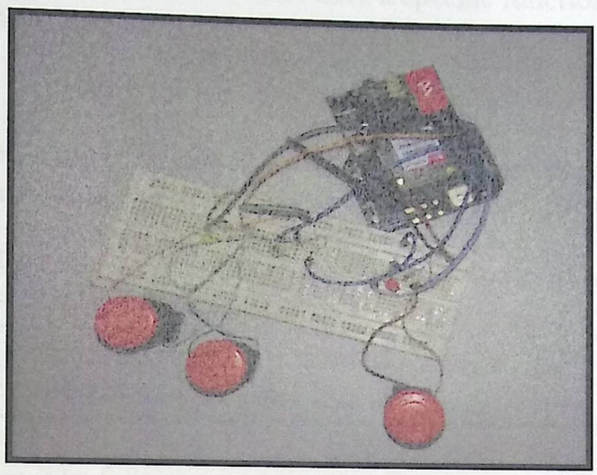


Figure 5.2: Arduino Connection

5.2 Software implementation

5.2.1 Server implementation

To make the server communicate with other devices there are two options available; the first is public IP, the second is the 3G modem, the first choice is prefered to reduce the cost of the project.

Server application has already been built using java programming language; there are five main operation in the system:

1. Open TCP socket on specific port.
2. Server waits the clients (GSM/GPS shield , Android devices) to send a connection request.
3. After the connection is established, the server reads the cars coordination continuously, and waits for the shield information.
4. Once the shield sends the accident location, the server calculates distance between it and the latest coordination from cars.
5. Server decides which is the minimum distance, then write on a socket to the intended car.

A detailed description for the server will be implemented by describing its main methods and interfaces as used in this system

5.2.1.1 Server main methods

The server program contains five classes, each have a specific function. A detailed description for each class is determined below.

CreatPort Class

This class is responsible for calling the other classes, so it uses partial programming in java called (Threads). It is used to perform the main operation, which includes:

1. Open socket: A TCP/IP socket is being used in order to make a connection between the server and both E-box and E-cars on specific port.

```
ServerSocket serverSocket = new ServerSocket(port);
server = serverSocket.accept();
```

2. Read incoming data: The following statements allow the server to read the incoming data from the E-car and E-box through the socket.

```
DataInputStream in = new DataInputStream(server.
getInputStream());
String message;
message = in.readLine();
```

3. Calculation to find distances: To return the values of the computed distances, this class must call the (Distance class).

```
Distance homecar1 = new Distance(home.getX(),
home.getY(), car1.getX(), car1.getY());
double homecar1d = homecar1.getdist();
```

4. Write on socket: In this procedure, the minimum distance is already computed and ready to be sent. The appropriate car moves by calling the (Alert class).

```
DataOutputStream out = new DataOutputStream(server.
getOutputStream());
out.writeUTF(dataforclosest);
```

Coordination Class

This class updates the incoming coordination by getting and setting (X=Latitude)&(Y=longtitude) every time received.

```
public void setall(double x, double y) {
        setX(x);
        setY(y);
}
```

Distance Class

This class used to calculate the distance between geographic information (latitudes & longitudes), in order to use it, a conversion must be used between degree and radian during the calculation according to the following equation:

$$\Delta\hat{\sigma} = arccos(sin\emptyset_s sin\emptyset_f + cos\emptyset_s cos\emptyset_f cos\Delta\Lambda)$$

```
public Distance(double lon1, double lat1, double lon2,
  double lat2) {

  double x1 = Math.toRadians(lat1);
  double y1 = Math.toRadians(lon1);
  //x2, y2 car
  double x2 = Math.toRadians(lat2);
  double y2 = Math.toRadians(lon2);
  //Distance between home and car
  double sec1 = Math.sin(x1) * Math.sin(x2);
  double dl = Math.abs(y1 - y2);
  double sec2 = Math.cos(x1) * Math.cos(x2);
  //sec1,sec2,dl are in degree, need to
  convert to radians
  double centralAngle = Math.acos(sec1 +
  sec2 * Math.cos(dl));
   //Radius of Earth: 6378.1 kilometers
   distance = centralAngle * 6378.1;
}

}
```

Alert Class

This class is used to check out the status of socket in order to write on it.

```
public static int alertPort = 0;
public static boolean alertF = false;
```

5.2.1.2 Server interfaces

The following interfaces appears once the server's program runs.

Welcome interface

Once the server application starts running, a greeting screen appears as shown in figure 5.4, the object **Start Button** is used in this form to enter the login window.

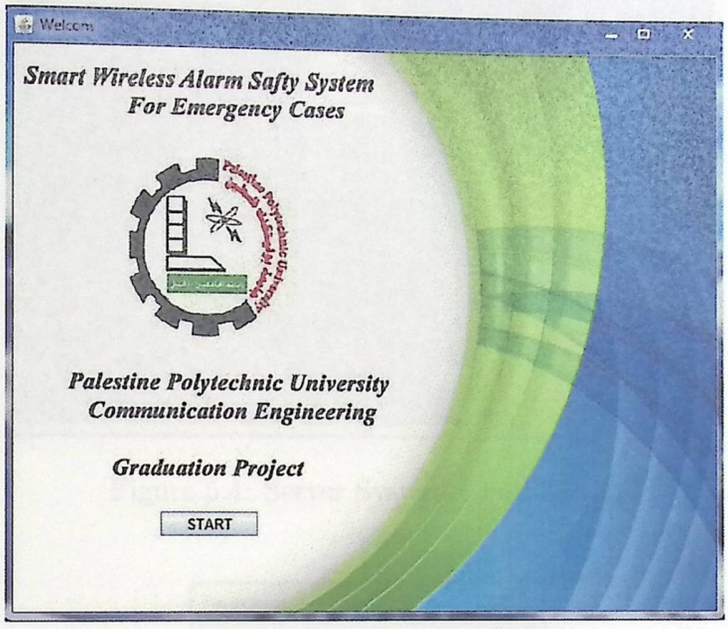


Figure 5.3: Server System Welcome Screen

Login interface

This form is used to enter the Username and Password to sign in to the system as shown in figure 5.5. A warning message appears if there is a wrong insertion in any of the field, as shown in figure 5.6, the object **Login** is used to get the username, password and sign in to the system; and that leads to the monitor form.

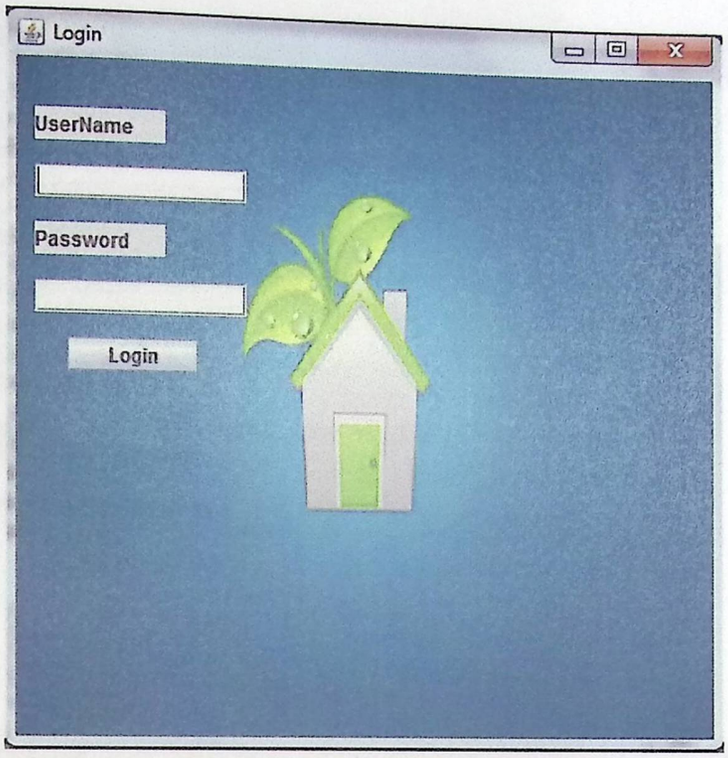


Figure 5.4: Server System Login Screen

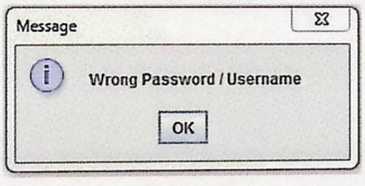


Figure 5.5: Warning Message

Monitor Form

Figure 5.7 shows the main form in the server system, which allows the observer to monitor all events occur.

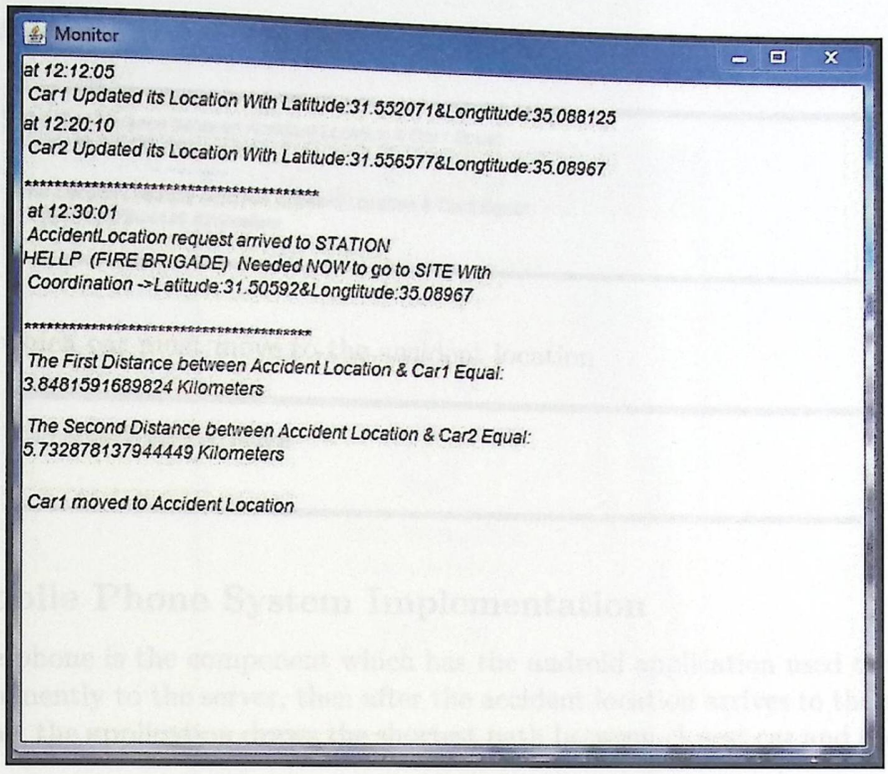
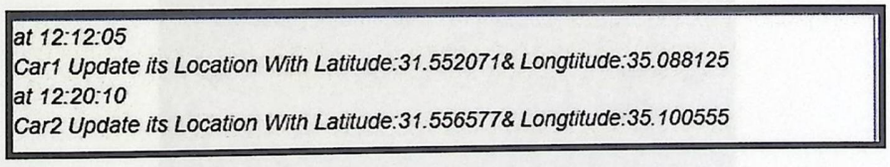


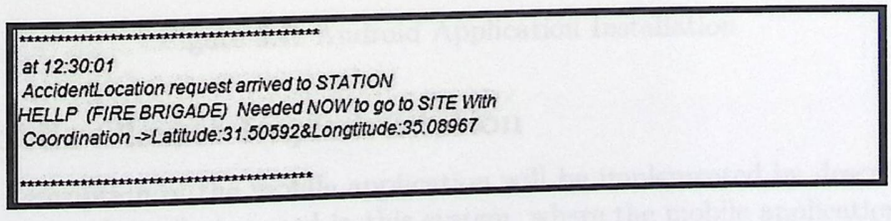
Figure 5.6: Main form Monitoring Screen

Monitor form events and processes are listed below in details:

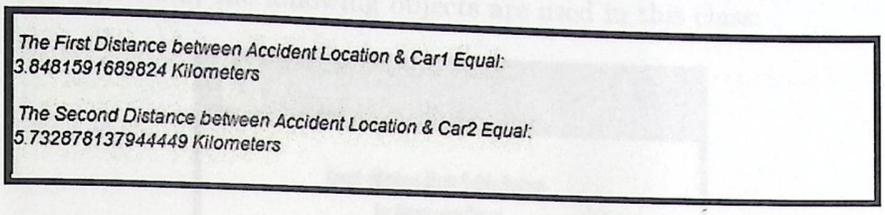
1. Displays the cars location updates: it is an indication for cars location update with specific times.



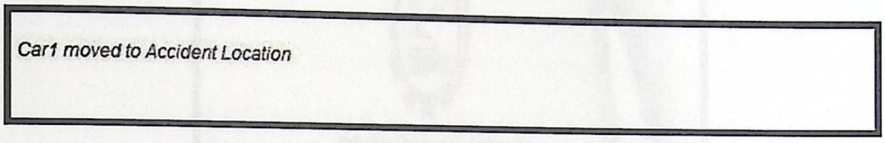
2. Displays the request from E-Box once accident occurs in a certain time. The server recognizes the type of accident depending on the chosen button, so the following clauses appear to the observer.



3. Calculates the distances then provides the results.



4. Decide which car must move to the accident location



5.2.2 Mobile Phone System Implementation

The mobile phone is the component which has the android application used to send the cars location permanently to the server, then after the accident location arrives to the application of the nearest car; the application draws the shortest path between closest car and the calling side.

Mobile application has already been built using the Eclipse environment, to use this application it must be uploaded on mobile, which will be performed by connecting the mobile phone through a USB cable to the PC. From Run Configuration window at eclipse choose the mobile phone, as shown in figure 5.8. The application must be installed on smart phone to be ready to use by the driver.

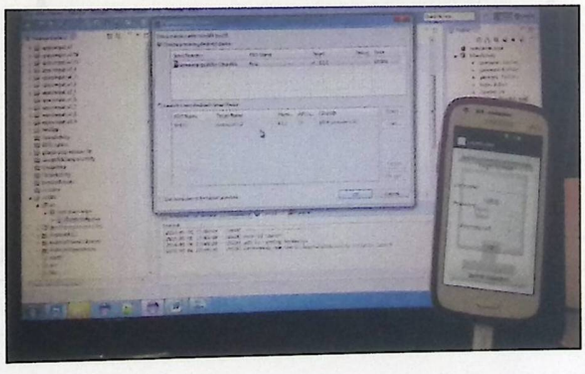


Figure 5.7: Android Application Installation

5.2.3 Mobile Classes Implementation

A detailed description of the mobile application will be implemented by describing its main classes, methods, and attributes used in this system, where the mobile application classes are:

1. **Welcome Class:** This class will be used when starting the application, as greeting screen, as shown in figure 5.9, the following objects are used in this class:

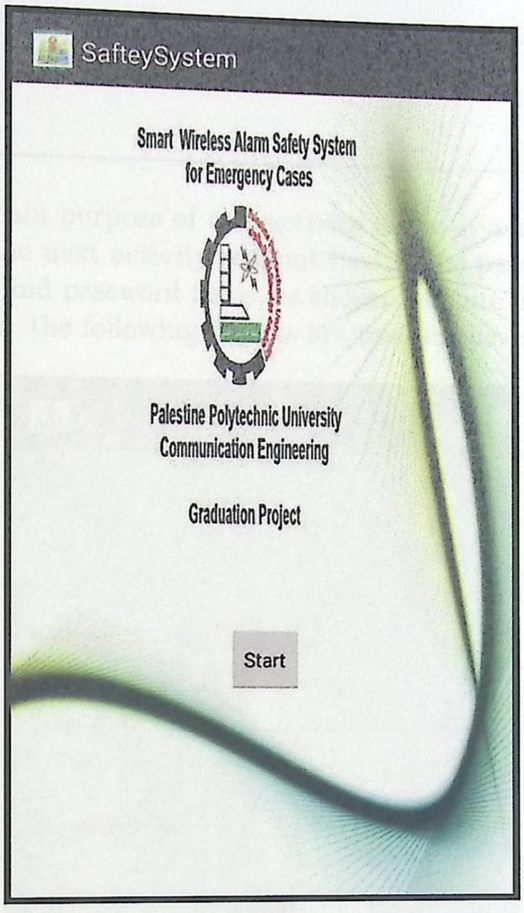


Figure 5.8: Welcome Activity

OnCreate

It is called when the activity is first created.

```
super.onCreate(savedInstanceState);
```

Start Button

This button is used to enter the next activity of this application, by activating the intent which will lead to the Login Class.

```
1    public void start(View v){

3    Intent intent=new Intent(Welcome.this,Login.class);
     startActivity(intent);

5     }
```

2. **Login Class:** The main purpose of this activity is the system and application security, no one can move to the next activity without having the password. The driver must fill the correct username and password fields, as shown in figure 5.10. Then he can enter the Update Location class, the following objects are used in this class:

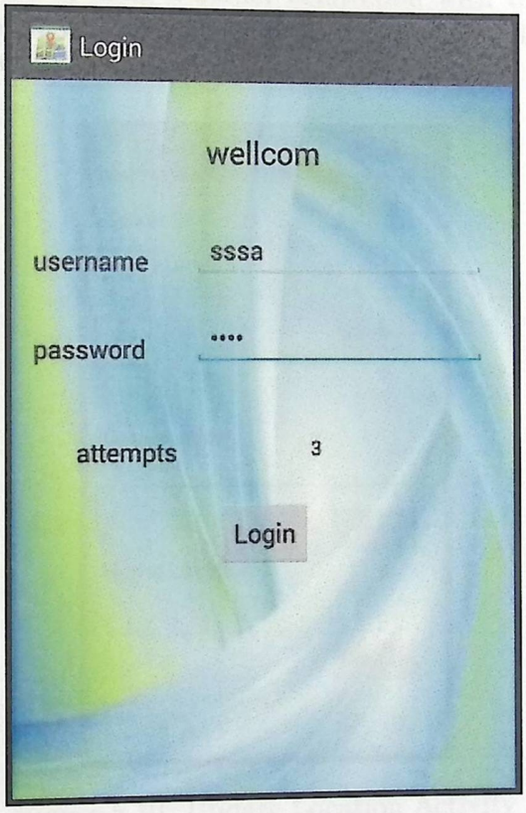


Figure 5.9: Login Activity

Attempts Counter

The driver has just three attempts to fill the fields, after that the activity closes.

```
   int counter = 3;
2  if(counter==0)
```

```
login.setEnabled(false);
```

Login Button

This button is used to enter the update location class, by using the intent method .

```
1 Intent intent1=new Intent(Login.this,
  UpdateLocation.class);
3 startActivity(intent1);
```

3. **Update Location Class:** This Class is the core of the android application, because it has the functionality of opening a socket connection with the server, and updates the location to send it permanently; in addition, it has the ability to receive the accident location from the server, as shown in figure 5.11.

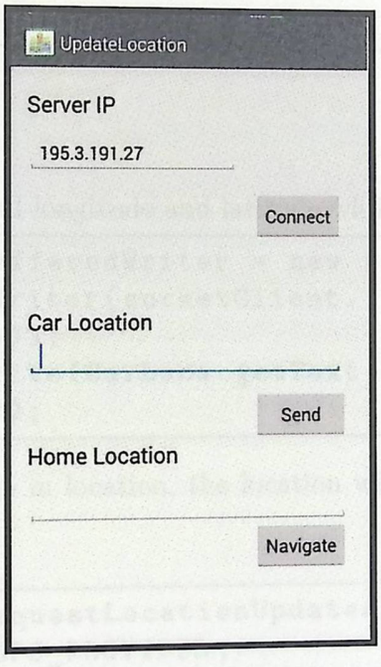


Figure 5.10: Update Location Activity

The main objects used in this class are:

Connect Button

This button is used to connect with the server, through opening the socket connection with the public IP. Then the rest of the fields will be valid.

```
private boolean connected = false;
private String server = "";
private static final int PORT NO. = variable;
connect.setOnClickListener(new OnClickListener() {

public void onClick(View v) {
try {
if (connected == false) {
socketClient=newSocket(server, PORT NO.);
connected = true;
connect.setText("Disconnect");
CarLoca.setEnabled(true);
HomeLoca.setEnabled(true);
Send.setEnabled(true); }
}
};
```

Send Button

This button sends the initial longitude and latitude while the car stands.

```
BufferedWriter bufferedWriter = new BufferedWriter(
new OutputStreamWriter(socketClient.
getOutputStream()));
bufferedWriter.write(CarLoca.getText().
toString() + "\n");
```

Whenever there is a change in location, the location will be sent to the server every 15 meters.

```
locationManager.requestLocationUpdates
(LocationManager.GPS_PROVIDER,
MINIMUM_TIME_BETWEEN_UPDATES,
MINIMUM_DISTANCE_CHANGE_FOR_UPDATES,
new MyLocationListener());
```

requestLocationUpdates() method; takes four parameters:

- Provider: The name of the provider with which we register. In this case, GPS is the provider.
- MinTime: The minimum time interval for notifications, in milliseconds.

- MinDistance: The minimum distance interval for notifications, in meters (15 meters) .
- Listener: An object whose **onLocationChanged()** method will be called for each location update.

The **MyLocationListener** class implements the LocationListener abstract class. Four methods needed in this implementation:

- OnLocationChanged (Location location): Called when the location has changed.
- OnProviderDisabled (String provider): Called when the provider is disabled by the user.
- OnProviderEnabled (String provider): Called when the provider is enabled by the user.
- OnStatusChanged (String provider, int status, Bundle extras): Called when the provider status changes.

Navigate Button

This button is used to catch the accident and car locations as an origin and destination points, then passes them to the final activity which is the shortest path.

```
public void navigate(View v)
  {
   Intent intent = new Intent(UpdateLocation.
   this,ShortestPath.class);
   intent.putExtra("GPS", homeloc);
   intent.putExtra("currentlocation", this.
   coordinatetoserver);
   startActivity(intent);
  }
```

4. Shortest Path Class: The main purpose of this class is to access Google Map, then coordinates the location from the previous activity, and draws the shortest path between current location and the accident location, as shown in figure 5.12.

Figure 5.11: Shortest Path activity

The following methods are used in this class:

API Key

In order to use Google Maps in Android application, we ensure to check the Google APIs as a build target, because Google Maps are not part of the standard Android SDK. To obtain this key for the application, the following steps are followed:

- Retrieve information about the application's certificate.
- Register a project in the Google APIs Console and add the Maps API as a service for the project.
- Request one or more keys.
- Add the key to the application and begin development.

```
<meta-data
 android:name="com.google.android.maps.v2.API_KEY"
 android:value="OUR.API.KEY" />
```

Intent

It is used to catch the coordination "current and accident locations" from the previous activity.

```
Intent intent = getIntent();
GPS = intent.getExtras().getString("GPS");
currentlocation = intent.getExtras().
getString("currentlocation");
```

5.2.4 Arduino software

The Arduino microcontroller is connected with SIM908(GPRS/GPS shield) which is called the E-Box; then it is programmed using the arduino AT command.

Initialize the pin modes and power configuration

For these initialization, the setup function will only runs once; after each power up or reset the Arduino board.

```
void setup()
  {
   digitalWrite(5,HIGH);
   delay(1500);
   digitalWrite(5,LOW);
   Serial.begin(9600); // set the baud rate
   digitalWrite(3,LOW); // enable GSM TX, RX
   digitalWrite(4,HIGH);
  }
```

Send the GPS coordination

AT commands used activate the GPS are listed in the following code.

```
   void send_GPS()
   {
   Serial.println("AT+CGPSIPR=9600");
   Serial.println("AT+CGPSPWR=1");
   Serial.println("AT+CGPSRST=1");
   Serial.println("AT+CGPSINF=0");
```

```
    }
```

Read Response from serial

A **read String** function is used to read data from serial; which it is the respond of AT commands.

```
void read String() {
index=0;
while(Serial.available() != 0)
{
if(index < 199)
{
inChar = Serial.read();
inData[index] = inChar;
index++;
}}}
```

Send location

GPS Reciever sends the location once a button is pressed; a void loop function is used to allow the program to execute and respond.

```
void loop()
 {
Serial.println("AT+CGATT=1");
Serial.println("AT+CSTT=\"Wataniya_internet\"");
Serial.println("AT+CIICR");
Serial.println("AT+CIPSTART=\"TCP\",\"195.3.191.27
\",\"12346\"");
Serial.println("AT+CIPSEND");
Serial.println("\r");
Serial.print(latitude);
Serial.println(longitude);
Serial.write(26);
 }
```

6

Chapter Six

Result and performance

6.1 Introduction

6.2 Testing Scenarios

Chapter 6

Results and performance

Measuring and testing the performance of the system is the final stage to complete the project. This chapter gives a detailed description of the project testing and performance evaluation.

6.1 Testing Scenarios

In this section, a demo will describe the scenario of starting the whole system, and several tests done for examination.

6.1.1 Demo

This demo describes the system startup with details in each part of it. **For the mobiles side**, the following steps are done for configuration issue.

- Download the mobile application and open it, then give it to the driver.
- The GPS and the mobile data must be turned on.
- Start the application then fill the correct username and password.
- Connect the E-cars and the server applications together; the mobiles at E-cars will start receiving the GPS signals and calculate its coordination, then send it to the server.
- When the driver move more than 15 meters the mobile application sends the new coordinates to the server.
- Once the accident location arrives, the navigation of the shortest path determined between accident and car locations.

Figure 6.1 shows the application activities and classes that will be shown on the mobile.

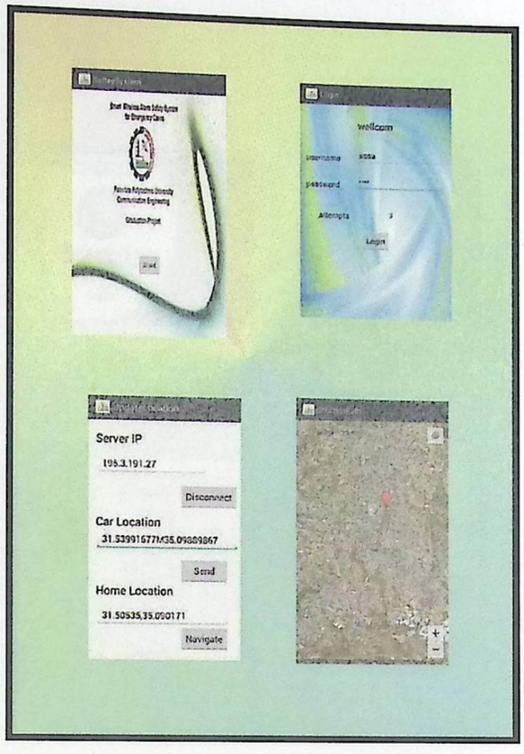


Figure 6.1: Mobile Phone Application Activities

For the server side, the following steps are done for configuration issue.

- Confirm the private IP address to get the public IP from the server.
- Open the server application from NetBeans IDE 8.0 program.
- Run the program to start the application then fill the correct username and password.
- After running; a TCP socket is opened to connect each of the E-cars and E-box with the server.
- The server receives the E-cars coordination periodically.
- Once the server receives the accident location from the E-Box; it calculates the linear distance between last coordination of each E-car and the accident location.
- The server sends the accident coordinates to the E-Car with the closest distance.

Figure 6.2 shows the application interfaces that will be shown on the server.

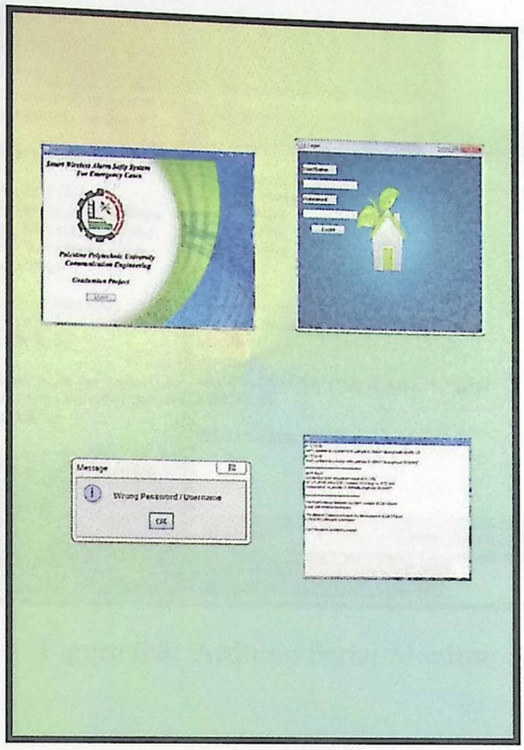


Figure 6.2: Server Application Interfaces

For the E-box side, which is a combination of Arduino microcontroller, a GPS/GPRS shield called SIM908, and the three emergency buttons, the following steps are done for configuration issue.

- Open the E-box program from Arduino 1.0.5-r2.
- With an USB cable connect the computer with the E-Box, to download the program on it.
- Remove the USB cable; and connect the power supply.
- Every button is responsible for a specific accident type; Once a button pressed, the location will be sent to the server.

Figure 6.3 shows the serial monitor screen when the program runs.

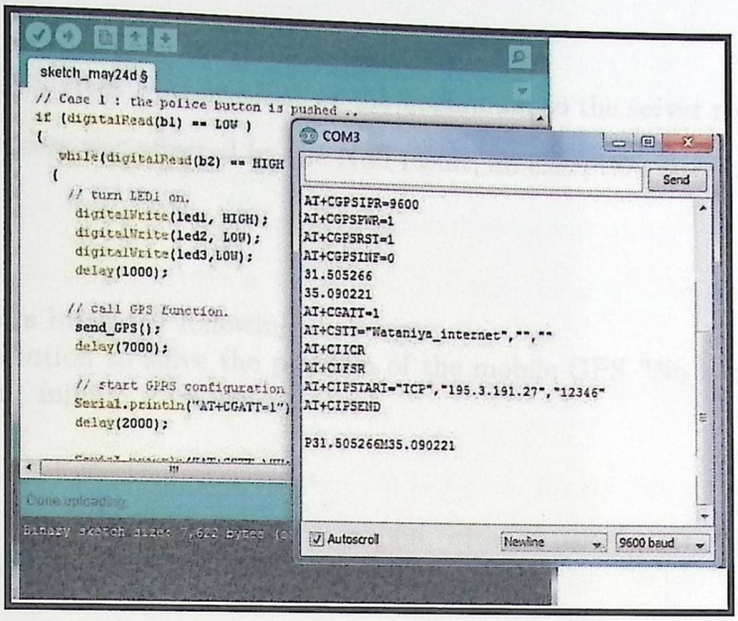


Figure 6.3: Arduino Serial Monitor

6.1.2 Tests

6.1.2.1 Test1

The first test is conducted over a small area; when the mobile phones and E-Box where close to the server, and without using cars. A person carries the mobile, then opens the application and starts walking.

Results

- Test time: the time is approximately 5 minutes.
- Mobile GPS: after 1 minute the location appears.
- E-Box GPS: no GPS error in this test, the coordinations were correct.
- Number of times that the person exeeds a 15 meter distance: 12 times; the distance of travel was 180 meter.
- Time that E-Box sends the location when button pressed: it takes 2 minutes approximately.
- Server response: the server connected with E-Box & E-cars, receives there coordination and do the processing to respond in a total time of 1 minute.

Errors

- The mobile GPS gives No Location! at the beginning; so the server receives Null !
- The server process was affected by the Null result, an exception error occurs at the server monitor.

Challenges

The mentioned errors bring the following challenges:
To give a perfect solution to solve the problem of the mobile GPS "No Location"; it must be turned on before one minute from using.

6.1.2.2 Test2

The second test is applied over a medium region, without using a car. A person carries the mobile and don't move, then opens the application.

Results

- Test time: the time is approximately 2 minutes.
- Mobile GPS: the location appears immediately.
- E-Box GPS: no GPS error in this test, the coordination was correct.
- Number of times that the person exceeds a 15 meter distance: 0 time; the test done in the case of stand.
- Time that E-Box sends the location when button pressed: it takes 1minutes approximately.
- Server response: the server connected with E-Box & E-cars, receives there coordination and do the processing to respond in a total time of one minute.

Errors

When accident occured, the server analyzes all coordinates received, then determines the closest car; but no request was received to the android applications.

Challenges

Determine the root of the problem if it is from the server code in transmitting part, or in the android application code in receiving part, and it is solved by many trials on both codes.

6.1.2.3 Test3

The final test is applied within Hebron city near downtown, two cars were used. A driver carries the mobile, open the application and start driving.

Results

- Test time: the time is approximately one minute.
- Mobile GPS: the location appears immediately.
- E-Box GPS: no GPS error in this test, the coordinates was correct.
- Number of times that the person exceeds a 15 meter distance: 100 times; the distance of travel was 2km meter approximately.
- Time that E-Box sends the location when button pressed: it takes one minute approximately.
- Server response: the server connected with E-Box & E-cars, receives there coordination and do the processing to respond in a total time of few nanosecond.
- The shortest path is drawn on the screen of the closest car's mobile.

Errors

In this test, there is no errors, and the system works properly.

7

Chapter Seven

Conclusion and recommendations

7.1 Introduction

7.2 Problems

7.3 Acquired learning outcomes

7.4 Recommendations for future work

Chapter 7

Conclusion and recommendations

This chapter describes the real learning outcomes that have been acquired during the work on the project, and recommendations and suggestions for future work and development.

7.1 Problems

Many problems, challenges, and issues have been raised during the work on the project. Many experiments, suggestions, ideas and researches have been carried out to deal with the different situations. Some of these problems are:

1. The Israeli restrictions on granted the 3G spectrum, which stops the projects depending on mobile data.
2. The dynamic IP of the data SIM, which introduces a problem in the process of sending and receiving.
3. Many problems have been raised while receiving GPS signals on Arduino shield, due to restrictions on the GPS in Palestine, and difficulty in using GPS over mobile networks, which are costy.
4. GPS coordination length is varied according to region; so a problem with splitting the GPS coordination and deal with them occurs at the server.

7.2 Acquired learning outcomes

After accomplishing the project tasks, many abilities have been achieved as:

1. We have learnt Android programming language that is used to program the mobile phone application, and how to deal with android developer tools program, which we depend on it to build a mobile application to send the cars locations, and receive the accident location, then apply the shortest path algorithm on Google map.

2. We have learnt Java programming language that is used to program the server at stations, and how to deal with TCP socket to send and receive coordination, then do processing on them.

3. We have learnt the Arduino programming language that is used to program GPS/GPRS shield in order to determine coordination then send them to the server.

4. We have learnt how to provide a quick, advanced and modern succor, when emergency situation occurs.

5. We have developed our abilities in troubleshooting and problem solving.

7.3 Recommendations for future work

At the end, some ideas can be given to develop the system or extend its duties and functions, and some recommendations can be given to avoid the problems that may happen in the future as:

1. Instead of using the shortest path algorithm with the Google map, the android code can be linked with the MAPS application that exists in any smart phone, which provide extra features.

2. Develop the shortest path algorithm so that taking into account the congestion at streets.

3. Support the E-Box with extra sensors for motion and fire instead of switches only.

4. Use the system over a high data rate cellular system will add to the efficiency and enhance the performance.

5. Use up to date, and accurate GPS system with minimal connection establishment, and response delay.

6. apply the shortest path algorithm on the server.

Bibliography

[1] GPS Component. White paper. http://geology.isu.edu/geostac/Field_Exercise/GPS/GPS_basics_u_blox_en.pdf.

[2] GPS receiver types. White paper. http://www.erpsoftwarebusiness.com/Software/GPS_Type.html.

[3] Civil Defense Reports. White paper. http://www.pcd.ps/new/sites/default/files/shhr_8.pdf.

[4] Patel Himaanshu. Gsm based emergency calling system. *International Journal Of Electronics And communication Engineering and Technology (IJECET)*, 4(1):35–44, February 2013.

[5] Sarika B.Kale. [design of intelligent ambulance and traffic control]. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2(5), April 2013.

[6] Asha Mehrotra. *GSM System Engineering.* Artech House, USA, 1997.

[7] GSM Specifications. White paper. http://www.ehow.com/about_6365030_gsm-technology.html?ref=Track2&utm_source=ask.

[8] GSM Network Architecture. White paper. http://edu.eap.gr/pli/pli23/documents/Parallila_Keimena/GSM.pdf.

[9] Timo Halonen. *GSM, GPRS and EDGE Performance: Evolution Towards 3G/UMTS.* John Wiley & Sons Ltd, England, 2003.

[10] GSM Modem Definition. White paper. http://www.nowsms.com/faq/what-is-a-gsm-modem.

[11] GSM Modem Specifications. White paper. http://www.mobitek-system.com/Products/GSM_Modem.html.

[12] SIM300 Features. White paper. http://probots.co.in/Manuals/SIM300.pdf.

[13] SIM900 Descriptions. White paper. http://wm.sim.com/producten.aspx?id=1022.

[14] SIM908 Features. White paper. http://wmhttp://store.open-electronics.org/GSM_GPRS_GPS_SHIELD.

[15] GPS Definition. White paper. http://gps.about.com/od/beforeyoubuy/a/howgpsworks.html.

[16] GR 213u GPS Receiver Holux Description & Features. White paper. http://www.holux.com/JCore/en/products/products_content.jsp?pno=273.

[17] GP-635T GPS Receiver Description & Features. White paper. https://www.sparkfun.com/products/11571.

[18] Microcontroller Definition. White paper. http://electrapk.com/what-is-a-microcontroller/.

[19] Microcontroller components. White paper. http://electrapk.com/microcontrollers-microcontroller-basics/.

[20] PIC18F2550 Description. White paper. http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010280.

[21] Arduino Uno Description & Specifications. White paper. http://arduino.cc/en/Main/arduinoBoardUno.

[22] AT89S52-8 bit Microcontroller. White paper. http://www.atmel.com/devices/at89s52.aspx.

[23] Alarm panic buttons. White paper. http://compnetworking.about.com/od/securitycontrol/a/what-is-a-panic-button.htm.

[24] John Agar. *Constant Touch: A Global History of the Mobile Phone.* Icon Books Ltd, London, 2013.

[25] Mobile operating system Definition and Types. White paper. http://www.gsmarena.com/glossary.php3?term=os.

[26] Android operating system & Architecture. White paper. http://www.tutorialspoint.com/android/android_architecture.htm.

[27] Mark L. Murphy. *Android Programming Tutorials.* CommonsWare, LLC, USA, 2010.

[28] James Gosling Ken Arnold. *THE Java Programming Language.* Addison Wesley Professional, USA, 2005.

[29] Kenneth L.Calvert. *TCP/IP Sockets in java.* Morgan Kaufmann, USA, 2008.

[30] Heuristic Background. White paper. http://whatis.techtarget.com/definition/heuristic.

[31] Donald B. Johnson. A note on dijkstra's shortest path algorithm. *International Journal Of Electronics And communication Engineering and Technology (IJECET)*, 20(3):385–388, July 1973.

[32] A* algorithm description. White paper. http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html.

[33] application programming interface. White paper. https://developer.android.com/guide/topics/location/index.html.

[34] SIM908 Arduino Shield Features. White paper. http://microcontrollershop.com/product_info.php?products_id=6010.

[35] Samsung Galaxy S II Specifications. White paper. http://www.gsmarena.com/samsung_i9100_galaxy_s_ii-3621.php.

[36] Arduino Uno Features. White paper. http://www.instructables.com/id/Intro-to-Arduino/step2/Arduino-Uno-Features/.

[37] Panic Switch. White paper. http://compnetworking.about.com/od/securitycontrol/a/what-is-a-panic-button.htm.

[38] Pin Configuration of Arduino Uno. White paper. http://arduino.cc/en/Main/ArduinoBoardUno.

[39] Ross Kurose. *Computer Networking.* Addison-Wesley Longman Publishing Co, 2008.

[40] AT commands with GPRS and GPS. White paper. http://www.dfrobot.com/image/data/TEL0051/3.0/SIM908_AT20Command20Manual_V1.01.pdf.

Abbreviation

A

APN: Access Point Name

AT command: Abbreviation of Attention

AUC: Authentication Center

API: Application Programming Interface

B

BSS: Base Station System

BTSs: Base Transceiver Stations

BSCs: Base Station Controllers

C

CDMA: Code Division Multiple Access

CS: Coding Schemes

CPU: Central Processing Unit

CD-ROM: Compact Disc, Read-Only-Memory

CLI: Common Language Infrastructure

CLR: Common Language Runtime

CMOS: Complementary metal–oxide–semiconductor

CEP: Circular Error Of Probability

D

DTMF: Dual-tone Multifrequency

DSP: Digital Signal Processor

E

EIR: Equipment identity registers

EMA: Emergency Medical Alert

EDGE: Enhanced Data rates for GSM Evolution

ECMA: European Computer Manufacturers Association

EV-DO, EV, EVDO: Enhanced Voice-Data Optimized or Enhanced Voice-Data Only

E-Car: Emergency Car

E-Box: Emergency Box

F

FTP/HTTP: File Transfer Protocol / Hypertext Transfer Protocol

FTDI: Future Technology Devices International

G

GSM: Global System for Mobile Communication

GPS: Global Positioning System

GPRS: General Packet Radio Service

GMSC: Gateway Mobile Services Switching Center

GIWU: GSM Interworking Unit

GGSN: The Gateway GPRS Support Node

GMSC: GSM Gateway Mobile Switching Center

GIS: Geographic Information System

H

HLR: Home Location Registers

HSCSD: High-speed circuit-switched data

I

IN: Intelligent Network

IP: Internet protocol

I/O: input/output

ICSP header:

IDEN: Integrated Digital Enhanced Network

IL: intermediate language

IS-IS: Intermediate System to Intermediate System

J

JIT: Just In Time

M

MXE: Message Center

MSC: Mobile Services Switching Center

MSN: Mobile Service Node

MS: Mobile Station

MMS: Multimedia Messaging Service

MISO: Master in Slave Out

MOSI: Slave in Master Out

N

NMEA: National Marine Electronics Association

NOR flash: Non-Volatile Computer Storage Flash memory

O

OSS: Operation and Support System

OMC: Operations and Maintenance Center

OTP ROM: One-Time Programmable Read-Only Memory

OHA: Open Handset Alliance

OS: Operating System

OSPF: Open Shortest Path First

P

PDN: Public Packet Data Network

PCU: Packet Control Unit

PCMCIA: Personal Computer Memory Card International Association

PDA: personal digital assistant

PWM: Pulse-width modulation

PPS: Pulse per Second

PC: Personal Computer

Q

Qos: Quality of Service

R

RF: Radio Frequency

RAM: Random Access Memory

S

SIM: Subscriber Identification Module

SMS: Short Message Services

SS: Switching System

SGSN: Serving GPRS Support Node

SS7: Signaling System No. 7

SPI LCD: Serial Peripheral Interface liquid-crystal display

SQL: Structured Query Language

SSL libraries: Secure Sockets Layer

SRAM: Static random-access memory

SCK: Serial Clock

SS: Slave Select

T

TDMA: Time Division Multiple Access

TCP/IP: Transmission Control Protocol / Internet Protocol

TTL serial: Transistor–transistor logic

U

USB: Universal Serial Bus

UMTS: Universal Mobile Telecommunications System

UART or USART: Universal Asynchronous Receiver/Transmitter

V

VLR: Visitor Location Registers

VM: Virtual Machine

VOIP: Voice over Internet Protocol

W

WLAN: Wireless Local Area Network

WAAS/EGNOS: Wide Area Augmentation System/ European Geostationary Navigation Overlay Service

WiMAX: Worldwide Interoperability for Microwave Access