SIX MONTH INDUSTRIAL TRAINING

REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

Semester Software/Industrial Training & Project

at

CETPA INFOTECH

SUBMITTED BY
HARSH KAUSHIK (1809261)
JAN - JULY
2022

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
SHAHEED BHAGAT SINGH STATE UNIVERSITY
FEROZEPUR, PUNJAB (INDIA)

TO WHOM IT MAY CONCERN

I hereby certify that " **HARSH KAUSHIK** " Roll No **1809261** of Shaheed Bhagat Singh State University, Ferozepur has undergone Semester Software/Industrial Training & Project from **10/Jan/2022** to **10/Jun/2022** at CETPA INFOTECH to fulfill the requirements for the award of degree of B.Tech. (ECE). He worked on **VEHICLE TRACKING SYSTEM USING GPS AND GSM** project during the training under the supervision of **KARTIK GUPTA** During his tenure with us we found him sincere and hard working. Wishing him a great success in the future.

~

Signature of the SUPERVISOR

CANDIDATE'S DECLARATION

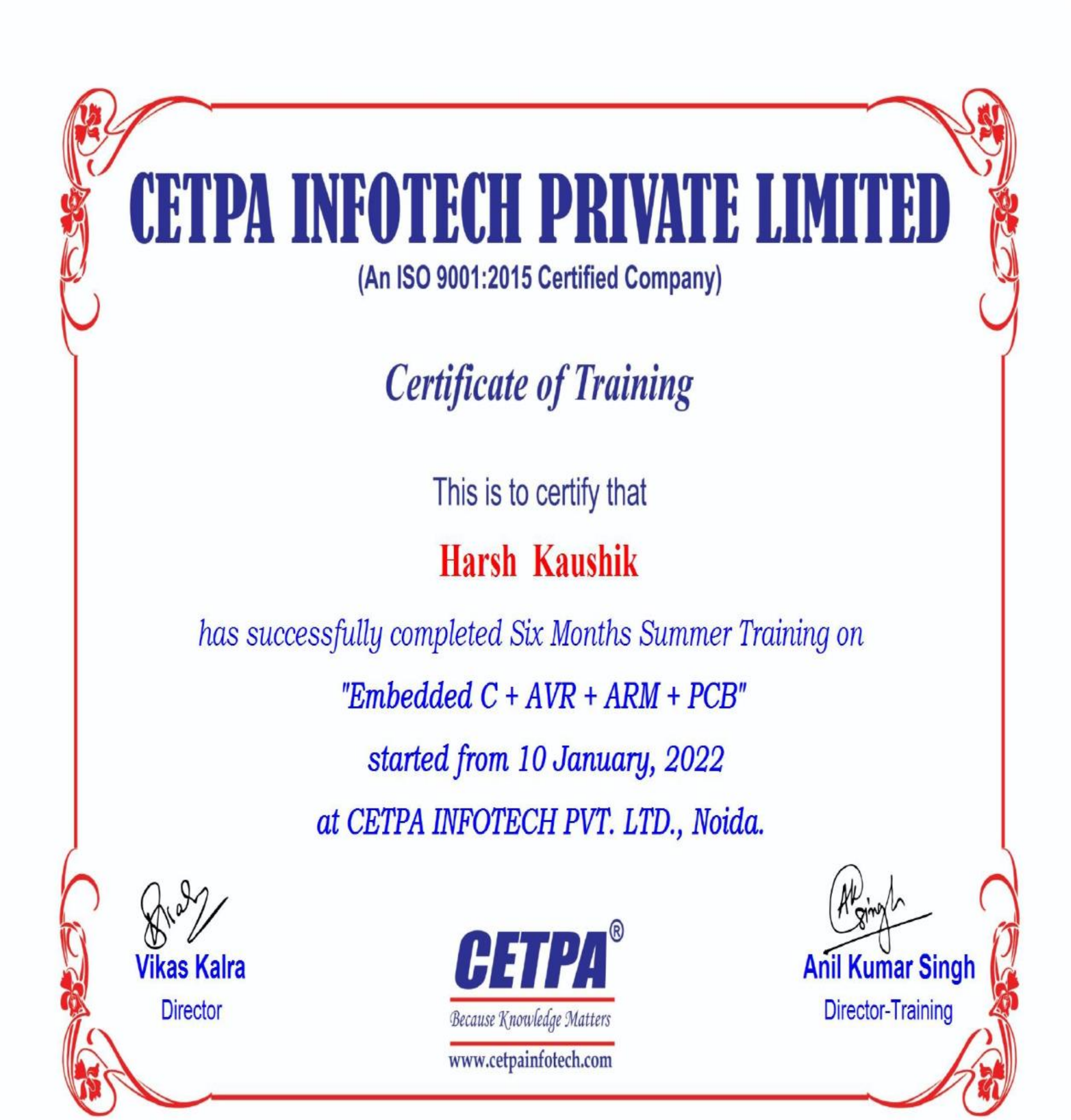
I hereby certify that the work which is being presented in the report entitled "Semester Software/Industrial Training & Project" by **HARSH KAUSHIK** University Roll No **1809261**, in partial fulfillment of requirement for the award of degree of B.TECH submitted in the "Department of ECE" at "Shaheed Bhagat Singh State University, Ferozepur " is an authentic record of my own work carried out during a period from **10/Jan/2022** to **10/Jun/2022,** under the supervision of **Mr. KARTIK GUPTA** The matter presented in this report has not been submitted in any other University/Institute for the award of B.Tech Degree.

Signature of the Student

ACKNOWLEDGEMENT

An Industrial training has an important role in exposing the real life situation in an industry. It was a great experience for me to work on training at **Cetpa infotech, Noida**

through which I could learn how to work in a professional challenging environment. Now, I would like to thank the people who guided me and have been a constant source of inspiration throughout the tenure of my industrial training. I wish my deep sense of gratitude to **Mr Kartik Gupta** whose affectionate guidance has enabled me to complete this training successfully. I also wish my deep sense of gratitude to **Dr. Vishal Sharma** (HOD: ECE Department at SBSSTU, FEROZEPUR) and Training & Placement Officer **MR. INDERJIT GILL** and other faculty members whose guidance and encouragement made my training successful.



CETPA INFOTECH PRIVATE LIMITED

(An ISO 9001:2015 Certified Company)

Certificate of Training

This is to certify that

Harsh Kaushik

has successfully completed Six Months Summer Training on

"Embedded C + AVR + ARM + PCB"

started from 10 January, 2022

at CETPA INFOTECH PVT. LTD., Noida.

Vikas Kalra
Director

CETPA®
Because Knowledge Matters
www.cetpainfotech.com

AK Singh
Anil Kumar Singh
Director-Training

Verify this Certificate by Registration Number ESEmbe10120226M2685131 at http://www.cetpainfotech.com

SOFTWARE TRAINING REPORT

IN

CETPA INFOTECH, NOIDA

NAME:-	**HARSH KAUSHIK**
BRANCH:-	Electronics and communication
UNIVERSITY ROLL NO. :-	1809261
SESSION:-	2018-2022
TRAINING AT:-	CETPA INFOTECH
TRAINING ON:-	Embedded C
TRAINING PERIOD	6 MONTHS

Chapter 1

Introduction of Institution

CETPA is a specialized training brand which has been awarded as a best IT and embedded training company for 5 consecutive years and award is delivered by Soha Ali Khan, Shekher Suman, Chetan Bhagat and Shashi Tharoor. CETPA is certified by ISO 9001:2008 and known for its quality training because CETPA believe in Quality and knowledge. CETPA is started before 12 years and continuously spread its wings in India and other country also. We have our branch in Germany and Ukraine. And in India we have our brand of education and training in Noida, Dehradun, Roorkee, Lucknow and Meerut. CETPA is a partner of Microsoft, Oracle, Autodesk, Panasonic and Nuvoton. CETPA have an experience to deliver our seminars and workshop in more than 500 colleges all over India.

CETPA provide certified and job oriented training. CETPA have an excellent placement. CETPA has specialized in 3 impotant domains that is, Training, Development and Consultancy. CETPA provide online training as well as tutorial training. CETPA experts helps student to prepare live project for students. CETPA have an excellent team of experts who understand student needs and guide them for their future and prepare them for their professional world. CETPA gives training to student according today industry demand. CETPA provide training to students as well as professionals too.

CETPA has specialization in 3 important domains namely: TRAINING, DEVELOPMENT and CONSULTANCY. The company provides specialized training in 50+ leading technologies like .NET, Java, PHP, Ethical Hacking, ANDROID, CCNA, AUTOCAD, VHDL, MATLAB, EMBEDDED SYSTEM, HVAC and many more.

CETPA, the Composition of "Computer Education & Technology Promotion Association", believes in quality not in quantity BECAUSE KNOWLEDGE MATTERS.

CETPA Offers –

Training -

CETPA is no 1 in the field of Software & Embedded Training programs for engineering & other students.

Consultancy -

CETPA provides job consultancy to the CETPA certified students.

Development -

CETPA development deals in software as well as embedded production development.

Brief Description about CETPA :-

CETPA INFOTECH PVT LTD is the leader in the "specialized training" brands of India certified by ISO 9001:2015 for its best quality. CETPA INFOTECH is the largest training service provider in various engineering domains for all engineering students as well as for the working professionals. It has an extensive experience of nurturing over 200000+ students in the past few years.

CETPA has been awarded as the "Best IT and Embedded Training Company" for 5 consecutive years for delivering high quality training and workshops at more than 500 colleges across India. CETPA is a trustworthy brand in Education and Training industry with its presence across several cities such as Noida, Roorkee, Lucknow, Dehradun and . The company was started 12 years back and it is continuously expanding having its overseas branches in Germany and Ukraine.

Like other academic and training company, at CETPA learner will be offer assortments programs but the trainers differ and make CETPA stand out from leftovers. CETPA has variety of knowledgeable and experienced trainers whose accession is different which learners can see everywhere. CETPA try its level best to boost their trainee's capability so that they stand out from others and whatever they grant to the corporate world inevitably becomes fruitful. Not only the fresher but also the professionals who are not adequate to deal with the upcoming technology and software are also helped here. CETPA try its aligned best to convey its services to every edge of the globe with the help of tailored education. CETPA's motto is to distribute the best utility to student and professionals and that is why CETPA have taken the personalized approach because CETPA does not want learners to adjust with their education.

Cetpa Infotech Private Limited Details

HEAD OFFICE (NOIDA)

D-58, Sector-2, Near Red FM.

Noida - 201301 , Uttar Pradesh

Phone No : 9212172602

ELECTRONIC SUPPORT

Toll Free : 0120-4535-353

query@cetpainfotech.com

ABSTACT

An embedded system is a special-purpose computer designed to perform one or a few dedicated functions, often with real time computing constraints. Embedded systems have become very important today as they control many of the common devices we use. An embedded micro controller is a chip, which has a computer processor and all its support functions, memory (Both program and data) and I/O (including bus interfaces) built within the device. The paper proposes a RCM (Remotely Controlled Model) which discusses significance of using Micro Controller to remotely control maximum number of devices so that the applications of Embedded System can be improved by connecting more devices to the Micro controller.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Chapter 2

Introduction to Embedded C

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems. So, in this article, we will see some of the Basics of Embedded C Program and the Programming Structure of Embedded C.

Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++, Python etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability

Embedded C Program: Basics

It is important to understand the basics of Embedded C programs.

Embedded C Program: Keywords

You cannot make use of any random words for defining any specific actions because the compiler won't be able to understand the word input provided by you. It is important for the compiler to first identify the word and then perform the specific action assigned (if any). For this to happen there are various predefined words that have a specific meaning defined in accordance with the compiler, known as keywords. Following is the table consisting of all the keywords that are used in Cx51 C Compiler (compilation of the programs that are developed for the 8051 based microcontroller is done by the respective compiler).

_at_	alien	bdata
code	compact	data
idata	interrupt	large
priority	reentrant	sbit
small	_task_	using

Embedded C Program: Data Types

When we want to declare any variables in the program we make use of data types. Following is the table consisting of all the data types that are used in Keil’s Cx51 Compiler.

Data Type	Bits(bytes)	Range
bit	1	0 or 1 (it is the bit addressable part of RAM)
signed bit	16(2)	-32768 to +32767
unsigned bit	16(2)	0 to 65535
signed char	8(1)	-128 to +127
unsigned	8(1)	0 to 255
float	32(4)	±1.175494E-38 to ±3.402823E+38
double	32(4)	±1.175494E-38 to ±3.402823E+38
sbit	1	0 or 1 (it is the bit addressable part of RAM)

What is the basic structure or a template of an Embedded C program?

It is important to have a basic framework in mind before writing any Embedded C program. Following is a practice template that can be used to write your initial Embedded C programs.

1. Multiline Comments
2. Single Line Comments
3. Pre-processor Directives
4. Global Variables
5. Function Declarations
6. Main Function
7. Local Variables
8. Function Calls
9. Infinite Loop
10. Statements
11. Function Definitions
12. Local Variables
13. Statements

Embedded C Program: Comments

The code of any program should be easily understood by the reader. Sometimes the code of a program can be so lengthy or complicated that it becomes for the developers as well as the reader to make any sense out of it. To eliminate the above stated problems comments are used. Comments are introduced in the code of a program to make it more readable. With the help of comments the reader can understand what is happening in the program.

For example:

```
E=x*y; /* the value of the multiplication of the two variables x and y is
stored in another variable E*
```

Embedded C Program: Single line Comments

When programming embedded systems there are many fractions of code that require proper explaining for future reference and to increase readability. When you want to only explain a certain fraction of a program it is best to use single line comments. Single line comments begin with (//) double slash, and have to be written in a single line because if the comment is continued in the next line the compiler will not ignore the comment and the program will fail to execute. Single line comments are also used to mention the beginning and ending of a certain block of program, promoting a proper structuring of the embedded program.

For example:

```
// the equation below defines the value of E variable

E= x+ y;
```

Embedded C Program: Multiline Comments

There are points in the program where explanation cannot suffice into a single line. Then you can use multiline comments. A multi line comment starts with a slash (/), then followed by a asterisk (/*) and ended by an asterisk (*) mark followed by a slash (*/). The compiler ignores the whole block of code where the multiline comments are written, meaning you can explain the code in multiple lines and these lines of explanation will be ignored by the compiler.

For example

E=x-y; /* the value of the subtraction of the two variables x and y is stored in another variable E*/

Embedded C Program: Directives of Processors

About Preprocessor Directive

Preprocessing is the first page of the C code compilation phase. The scanning of the provided code and the modifications of the scanned code is done by the preprocessor. The compiler stage receives the code from the processor after the completion of the scanning and modifying process. The commands that are given to the preprocessor are known as the C Preprocessor directives. Hash symbol (#) is used in the beginning of the preprocessor directives.

In the first step the preprocessor scans the complete code and identifies the token # (if any), when the preprocessor identifies a token it modifies the source code, these modifications are done by taking actions that are based on preprocessor directives. When the source code is modified, any preprocessor directives present are eliminated (the directives that begins with the hash symbol (#)). After the termination of the preprocessor directives the code is given to the compiler to complete the process of compilation.

Declaration of new preprocessor directives cannot be done by any programs as the set of directives names (valid) declared are fixed. Before an identifier (which is the name of the directive) hash symbols are used.

It is important to notice that there are no semicolons at the end of a preprocessor directive. If you want to declare multi line preprocessor directives then backslashes (\) are used for it.

What are the functions of pre-processor directives?

The pre-processor directive used to include files is #include.

Note: before the directive name the symbol hash (#) is used.

When including files from standard directory the pre-processor #include<filename.h> is used. The path of the standard directory is defined in IDE (Integrated Development Environment).

#include is identified as a command by the pre-processor, also the file's content is included in the C source code. Similarly, in embedded C the header files of the microcontroller are included. When the pre-processing process is completed the content from the header files are included in the course code.

For example

The most used program to test the microcontroller: The blinking of Led using 8051 microcontroller.

```
#include<reg51.h> void main()

{

unsigned int x;

for(;;)

{

P2=0x55;

for( x=0;x<400;x++); {
} P2=0xAA;

for( x=0;x<400;x++); {

}

}

}
```

OUTPUT- When the execution process of the program is completed, the Led will start to blink alternatively at odd and even positions, after some delay.

Factors for Selecting the Programming Language

The following are few factors that are to be considered while selecting the Programming Language for the development of Embedded Systems.

- **Size:** The memory that the program occupies is very important as Embedded Processors like Microcontrollers have a very limited amount of ROM (Program Memory).
- **Speed**: The programs must be very fast i.e., they must run as fast as possible. The hardware should not be slowed down due to a slow running software.
- **Portability:** The same program can be compiled for different processors.
- Ease of Implementation
- Ease of Maintenance
- Readability

The main features of the C language include the following.

- C language is software designed with different keywords, data types, variables, constants, etc.

- Embedded C is a generic term given to a programming language written in C, which is associated with a particular hardware architecture.
- Embedded C is an extension to the C language with some additional header files. These header files may change from controller to controller.

- The microcontroller 8051 #include<reg51.h> is used.

List a few known differences between C language and Embedded C

When talking about the differences between the C language and the Embedded C, there are not many to discuss except for a few extensions and the interface environment. They both have the same syntax meaning learning the syntax for any one language will automatically help to operate in the other language. Also, they have the same data types, functions, they make use of the same libraries, dictionaries etc.

It can be said that the Embedded C is an extension of the C programming language. Many features such as addressing I/O, fixed point arithmetic, multiple memory addressing and many more features are an addition to the artillery of the Embedded C.

Also, many desktop applications are developed using C programming language, on the other hand applications that are microcontroller based are developed using Embedded C.

Advantages of Embedded C programming language

- The time consumed by the development cycle of an embedded program reduced significantly.
- Developers were able to code easily.
- Modifications and updates were made easily, resulting in more successful maintenance cycles.

- Library functions were introduced that eliminated the need to write the same block of code again and again for each program.
- The compatibility of a particular program with various architectures increased as it only required a few modifications.
- The hardware cost used in embedded C programs is very low.
- The embedded applications that are designed are highly efficient for industrial usage.
- The complexity of the code is reduced significantly.

Disadvantages of Embedded C programming language

Following are the disadvantages of embedded C programming language.

- It executes one task at a time, it does not support multiple task executions.
- When we make any changes in the program then the appropriate changes are to be made in the hardware as well.
- Only hardware systems are supported.
- It cannot be scaled efficiently, there are still some issues faced while doing it.

Applications of Embedded C Program

The **applications of embedded c programming** include the following.

- Embedded C programming is used in industries for different purposes
- The programming language used in the applications is speed checker on the highway, controlling of traffic lights, controlling of

street lights, tracking the vehicle, artificial intelligence, home automation, and auto intensity control.

Embedded System

- A system incorporating both the hardware side as well as the software, which is designed to perform pre specified tasks are known as embedded systems. Examples of embedded systems that are widely used in day to day life are mobile phones, refrigerators, machine machines, etc.

- We might be exploiting these embedded systems without realizing that these embedded systems are designed to have their own hardware as well as the software to perform the assigned tasks properly and efficiently.

- There are various actions that are performed when you use your mobile phones such as you run different applications on a single phone, you use your phone to watch videos or maybe play music and many such other tasks. All the tasks that are being performed on a mobile device require a proper set of instruction and execution protocol to run smoothly.

- Also, embedded systems cannot be restricted only to stationary devices or stand alone devices, they can be scaled to relatively larger systems. For example airplanes have the required hardware assisted by a supportive software that is used by the pilots to perform various maneuvers, also providing the feature of auto pilot, etc. These actions are taken care of by the embedded systems that are installed on an airplane. These embedded systems provide safety as they operate on an incredibly low rate of errors, and these systems can be trained for any type of situations by exposing the system to the desired condition by using simulators.

How do you program an embedded system?

We have made it clear by now that embedded systems are designed to have their unique hardware and software. For an embedded system considering that the respective system is of a simple design, the processor of the respective embedded system is considered to be the main hardware module of the system. The processor of an embedded system is considered to be the soul of the respective embedded system. For example the microcontroller, microprocessor, Complex Programmable Logic Device (CPLD), DSP or Field Programmable Gated Array (FPGA) can be considered as processors.

The above-mentioned embedded systems are similar to one another as they can be programmed easily, in simple words for any embedded system to perform certain predefined tasks you can write the program for it. This is considered to be the software side of an embedded system. All the instructions and the execution protocols are defined for an embedded system to perform properly.

There has to be a link or a connection or a well defined path for operating and communicating between the events happenings externally and internally of an embedded system, the programmed software also known as the embedded software is used for the above mentioned task as it operates the hardware to monitor and then analyze the external events and perform the specified actions by controlling the output of the respective embedded system. During the execution of the entire process, the embedded program usually interacts with the embedded hardware of the embedded system usually referred to as a processor to enable the system to perform the specified tasks.

Earlier Embedded Systems were developed mainly using Assembly Language. Even though Assembly Language is closest to the actual machine code instructions and produces small size hex files, the lack of portability and high amount of resources (time and man power) spent on developing the code, made the Assembly Language difficult to work with.

There are other high-level programming languages that offered the above mentioned features but none were close to C Programming Language. Some of the benefits of using Embedded C as the main Programming Language:

- Significantly easy to write code in C
- Consumes less time when compared to Assembly
- Maintenance of code (modifications and updates) is very simple
- Make use of library functions to reduce the complexity of the main code
- You can easily port the code to other architecture with very little modifications

Important Factors to consider for programming language selection

When selecting a programming language it is important to consider the follow factors:

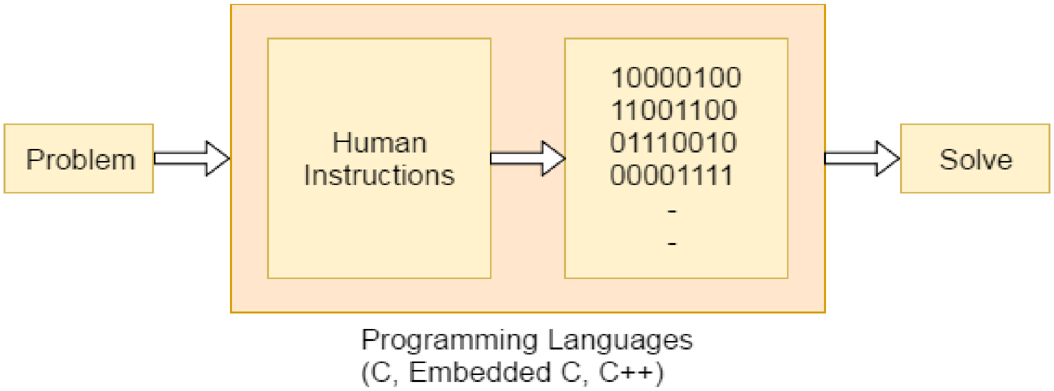
- Program memory plays a very important role in the smooth functioning of an embedded system. We are aware of the fact that microcontrollers have limited program memory (ROM). So, it is important for the embedded program to occupy limited memory.
- The speed at which the embedded software operates should be fast timed. The action time of a hardware should not be hampered by an embedded program that takes forever to execute actions. The embedded software should be continuously optimized to perform as fast as possible.

- In today's world time cannot be wasted to code a processor from scratch, the embedded software should be portable, meaning the programs that run on a particular processor should be easily compiled by another processor.
- The process of implementation should not be time consuming as it could negatively affect the development cycle.
- A proper maintenance process should be designed.

Embedded System Programming:

Basic Declaration

Let's see the block diagram of Embedded C Programming development:



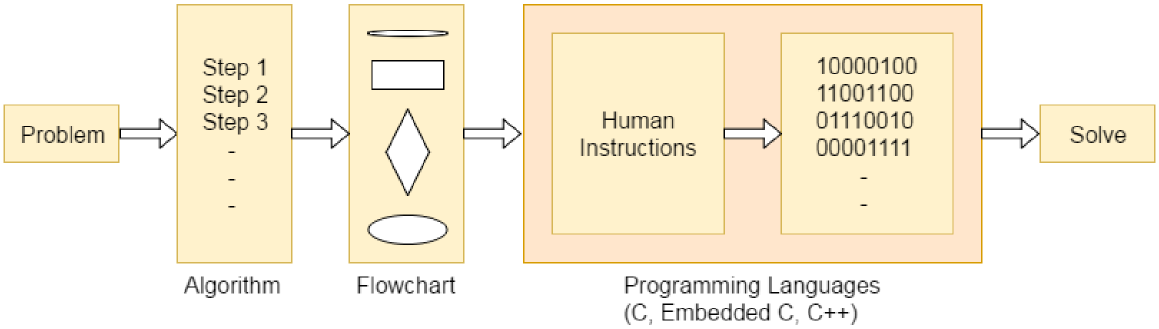
Function is a collection of statements that is used for performing a specific task and a collection of one or more functions is called a programming language. Every language is consisting of basic elements and grammatical rules. The C language programming is designed for function with variables, character set, data types, keywords, expression and so on are used for writing a C program.

The extension in C language is known as embedded C programming language. As compared to above the embedded programming in C is also have some additional features like data types, keywords and header file etc is represented by

```
1. #include<microcontroller name.h>
```

Basic Embedded C Programming Steps

Let's see the block diagram representation of Embedded C Programming Steps:



The microcontroller programming is different for each type of operating system. Even though there are many operating system are exist such as Windows, Linux, RTOS, etc but RTOS has several advantage for embedded system development.

The Embedded Environment

Microcontrollers used in development projects have very limited resources. You are working close to your target machine and you must be familiar with your target hardware construction and operation. A good quality C development environment incorporates tools which allow you to concentrate primarily on your applications and not on the hardware which runs them. However, you cannot ignore low-level details of your target hardware. The better you understand your run-time environment, the better you can take advantage of its limited capabilities and resources.

The Embedded Difference There are many aspects of embedded systems development which must be considered. These are:

Reliability : Embedded systems must be reliable. Personal computer programs such as word processors and games do not need to achieve the same standard of reliability that a microcontroller application must. Errors in programs such as word processors may result in errors in a document or loss of data. An error in a microcontroller application such as a television remote control or compact disc player will result in a product that does not work and consequently does not sell. An error in a microcontroller application such as an antilock braking system or autopilot could be fatal.

Efficiency : Issues of efficiency must be considered in real time applications. A real time application is one in which must be able to act at a speed corresponding with the occurrence of an actual process.

Cost : Many embedded systems must compete in a consumer market and cost is an important issue in project development.

Fabrication Techniques

CMOS

Complementary Metal Oxide Semiconductor (CMOS) is a technique commonly used to fabricate microcontrollers. CMOS requires less power and CMOS chips can be static which allows the implementation of a sleep mode. CMOS microcontrollers must have all inputs connected to something.

PMP

Post Metal Programming (PMP) allows ROM to be programmed after final metalization. This allows ROM to be programmed very late in the productions cycle. Memory Addressing and Types

Each microcontroller has a specific addressing range. An addressing range is the number of addresses a microcontroller can access. The addressing scheme used to access to these spaces varies from processor to processor, but the underlying hardware is similar.

RAM

Random access memory1 or RAM consists of memory addresses the CPU can both read from and write to. RAM is used for data memory and allows the CPU to create and modify data as it executes the application program. RAM is volatile, it holds its contents only as long as it has a constant power supply. If power to the chip is turned off, the contents of RAM are lost. This does not mean that RAM contents are lost during a chip reset. Vital state information or other data can be recorded in data memory and recovered after an interrupt or reset. 1 random access memory is used because the CPU can access any block of memory in RAM in the same amount of time. This differs from sequential storage such as tape where access time differs for different parts of the storage space. Memory Addressing and Types 25 Some chips provide an alternate RAM power supply so that memory contents can be maintained even when the rest of the chip is without power. This does not make RAM any less volatile, without a backup power source the contents would still be lost. This type of RAM is called battery backed-up static RAM.

ROM

ROM, read only memory, is typically used for program instructions. The ROM in a microcontroller usually holds the final application program. Maskable ROM is memory space that must be burned in by the manufacturer of the chip as it is constructed. To do this, you must provide the chip builder with the ROM contents you wish the chip to have. The manufacturer will then mask out appropriate ROM blocks and hardwire the information you have provided. Since recording chip ROM contents is part of the manufacturing process, it is a costly one-time expense. If you intend to use a small number of parts, you may be better off using chips with PROM. If you intend to use a large number of parts for your application, then the one-time expense of placing your program in ROM is more feasible.

PROM

Programmable ROM, or PROM, started as an expensive means to prototype and test application code before burning ROM. In recent years PROM has gained popularity to the point where many developers consider it a superior alternative to burning ROM. As microcontroller applications become more specialized and complex, needs for maintenance and support rise. Many developers use PROM devices to provide software updates to customers without the cost of sending out new hardware. There are many programmable ROM technologies available which all provide a similar service. A special technique is used to erase the contents of programmable ROM then a special method is used to program new instructions into the ROM. Often, the developer uses separate hardware to perform each of these steps.

EPROM

EPROM (erasable programmable ROM) is not volatile and is read only. Chips with EPROM have a quartz window on the chip. Direct exposure to ultra-violet The Embedded Environment 26 radiation will erase the EPROM contents. EPROM devices typically ship with a shutter to cover the quartz window and prevent ambient UV from affecting the memory. Often the shutter is a sticker placed on the window. Developers use an EPROM eraser to erase memory contents efficiently. The eraser bombards the memory with high-intensity UV light. To reprogram the chip, an EPROM programmer is used, a device which writes instructions into EPROM. The default, blank state for an EPROM device has each block of memory set. When you erase an EPROM you are really setting all memory blocks to 1. Reprogramming the device resets or clears the appropriate EPROM bits to 0. Because of the way EPROM storage is erased, you can not selectively delete portions of EPROM – when you erase the memory you must clear the entire storage space.

EEPROM

EEPROM (electrically erasable programmable ROM) devices have a significant advantage over EPROM devices as they allow selective erasing of memory sections. EEPROM devices use high voltage to erase and re-program each memory block. Some devices require an external power source to provide the voltage necessary for erasing and writing and some have an onboard pump which the chip can use to build up a charge of the required voltage.

Flash Memory

Flash memory is an economical compromise between EEPROM and EPROM technology. As with EEPROM high voltage is applied to erase and rewrite flash memory. However, unlike EEPROM, you can not selectively erase portions of flash memory – you must erase the entire block as with EPROM devices. Many manufacturers are turning to flash memory. It has the advantages of not requiring special hardware and being inexpensive enough to use in quantity.

Registers

The CPU maintains a set of registers which it uses to store information. Registers are used to control program execution and maintain intermediate values needed to perform required calculations. Some microcontrollers provide access to CPU registers for temporary storage purposes. This can be extremely dangerous as the CPU can at any time overwrite a register being used for its designated purpose.

TIMER

A TIMER interrupt occurs when a timer overflow is detected. For example, In the Microchip PIC16C74 there is a TMR0 interrupt which is generated when the TMR0 8 bit timer overflows. An overflow occurs when the timer goes from 1111 1111 to 0000 0000. The timer is usually incremented every instruction cycle.

Difference between Software Testing and Embedded Testing

Software Testing

Software testing is the process of verification and validation for a software. It ensures that a software or application is free from defects and viruses. It also ensures that software fulfills the requirements of end user as designed and developed. It makes sure that software meets the user requirements effectively and efficiently and also handles all the exceptional and boundary cases.

Embedded Testing:

Embedded testing is the process of verification and validation of both software and hardware. It ensures the defect free whole system including software and hardware. It is basically performed on hardware in order to find the defects. It also ensures that system meets the end user's requirements.

ARM processor and its Features

ARM is considered to be family of Central Processing Units that is used in music players, smartphones, wearables, tablets and other consumer electronic devices.

The architecture of **ARM processor** is created by **Advanced RISC Machines**, hence name ARM. This needs very few instruction sets and transistors. It has very small size. This is reason that it is perfect fit for small size devices. It has less power consumption along with reduced complexity in its circuits.
They can be applied to various designs such as 32-bit devices and embedded systems. They can even be upgraded according to user needs.

The main features of ARM Processor are mentioned below :

1. **Multiprocessing Systems –**
   ARM processors are designed so that they can be used in cases of multiprocessing systems where more than one processors are used to process information. First AMP processor introduced by name of ARMv6K had ability to support 4 CPUs along with its hardware.

2. **Tightly Coupled Memory –**
   Memory of ARM processors is tightly coupled. This has very fast response time. It has low latency (quick response) that can also be used in cases of cache memory being unpredictable.

3. **Memory Management –**
   ARM processor has management section. This includes Memory Management Unit and Memory Protection Unit. These management systems become very important in managing memory efficiently.

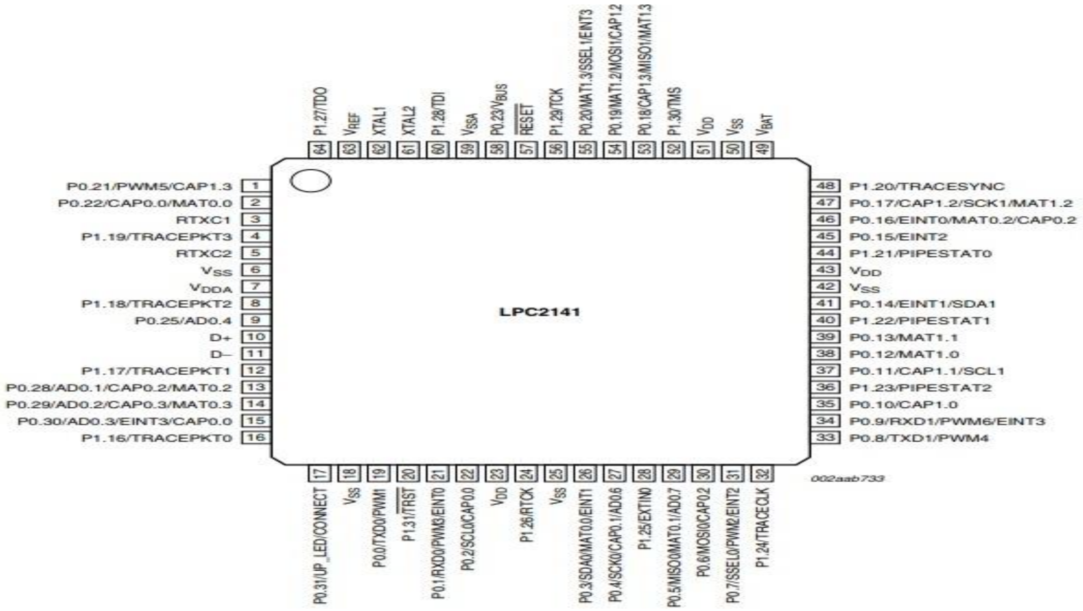
4. **Thumb-2 Technology –**
   Thumb-2 Technology was introduced in 2003 and was used to create variable length instruction set. It extends 16-bit instructions of initial Thumb technology to 32-bit instructions. It has better performance than previously used Thumb technology.

5. **One cycle execution time –**
   ARM processor is optimized for each instruction on CPU. Each instruction is of fixed length that allows time for fetching future instructions before executing present instruction. ARM has CPI (Clock Per Instruction) of one cycle.

6. **Pipelining –**
   Processing of instructions is done in parallel using pipelines. Instructions are broken down and decoded in one pipeline stage. The pipeline advances one step at a time to increase throughput (rate of processing).

7. **Large number of registers –**
   Large number of registers are used in ARM processor to prevent large amount of memory interactions. Registers contain data and addresses. These act as local memory store for all operations.

Pins Details



Pin1 (P0.21/PWM5/AD1.6/Cap1.3): It is a general-purpose pin and it could be used for four multiple ways such as it could be as input output data pin, as a pulse width modulation generator, as an analog to digital converter and as a capture input for timer l channel 3.

Pin 2 (P0.22/AD1.7/CAP0.0/MAT0.0): It could be for used for four purposes. First one P0.21, it could be used for input output data pin, second one AD1.7 , it could be used as analog to digital converter with ADC 1, input 7. Third one CAP0.0,it could be used as a capture input for timer 0 and channel 0. Fourth one MAT 0.0, it could be used as match output for timer 0 and channel 0.

Pin 3 (RTC X1): Pin3 is used as input pin for RTC oscillator circuit.

Pin 4 (P1.19/TRACEPKT3): Pin 4 could be as GPIO pin as well as with 3 bit input output pin for inner pull up.
Pin 5 (RTCX2): Pin 4 is used as output pin for RTC oscillator circuit.

Pin(6,18,25,42,50): These pins are used as references pins for grounding the microcontroller.

Pin7(VDDA): This pin is used as voltage source pin with 3.3 Volts. These voltages could be useful for on chip digital to analog converter and analog to digital converter.

Pin8(P1.18/TRACEPKT2): This pin is used as GPIO pin and 2 bit input output pin for inner pull up.

Pin9(P0.25/AD0.4/AOUT): This pin is used as GPIO pin, as input 4 for AD0 and as output pin for digital to analog converter.

Pin(10,11): Pin10 is used for D+ line bidirectional USB. Similarly the pin11 is used for D-line bidirectional USB.

Pin12(P1.7/TRACEPKT1): This pin is used as a GPIO pin and as a standard input/output port for inner pull up.

Pin13 (P0.28/AD0.1/CAP0.2/MAT0.2): This pin is used as a GPIO pin, analog to digital converter pin for ADC-0 input 1, capture input pin for timer 0 channel 2 and as a match output pin for timer 2 channel 1.

Pin14(P0.29/AD0.2/CAP0.3/MAT0.3): This pin could be used as a GPIO pin, converter input pin for ADC-0 input 2,capature input for timer 0 channel 3 and as a match output pin for timer 0 channel 3.

Pin15(P0.30/AD0.3/CAP0.3/EINT3/CAP0.0): This pin could be used as GPIO pin, converter pin for ADC-0 timer input 3,external interrupt with input 3 and as capture input pin for timer 0 channel 0.

Pin(17,19,20,21): All these pins are used as GPIO pins. Pin17 is used as UP_LED pin, means it is used as indicator pin. Pin19 is used as a transmitter output for UART0 and as a pulse width modulator for output 1. Pin20 is used as a reset pin for JATG interface. Similarly the pin21is uses as receiver input for UART0,as PWM generator for output 3 and as external interrupt with input 0.

Pin(22,24,26,27,28,29,30): These are GPIO pins. Pin22 is used as clock input output and capture input pin, pin 24 is used as CLK output during JATG interface. Pin 26 is used as matched output for timer 0 channel 0 and as external interrupt for input1.Pin 27 is used as a serial clock for transferring data from master bus to slave bus and as a digital converter ADC-0.6 for input 6. Pin 28 could be used as external trigger input with inner pullup. Pin 29 is used as MISO for transferring data form mater to slave bus and used as a converter ADC-0 with input 7.Pin 30 is also used as MISO output and as a capture input for timer 0 channel 2.

Pin(23,43,51): These pins are used for suppling input voltages to internal core and input output ports.

AVR MICROCONTROLLER

AVR microcontroller is an electronic chip manufactured by Atmel, which has several advantages over other types of microcontroller.

We can understand microcontroller by comparing it with Personal Computer (PC), which has a motherboard inside it. In that motherboard a microprocessor (AMD, Intel chips) is used that provides the intelligence, EEPROM and RAM memories for interfacing to the system like serial ports, display interfaces and disk drivers. A microcontroller has all or most of these features built into a single chip, therefore it doesn’t require a motherboard and any other components.

AVR microcontroller comes in different configuration, some designed using surface mounting and some designed using hole mounting. It is available with 8-pins to 100-pins, any microcontroller with 64-pin or over is surface mount only.

Some mostly used AVR microcontrollers are:-

- ATmega8 microcontroller
- ATmega16 microcontroller
- ATmega32 microcontroller
- ATmega328 microcontroller

Key Features:

Consider some general features of ATmega32 microcontroller is:-

- 2 Kilo bytes of internal Static RAM
- 32 X 8 general working purpose registers
- 32 Kilo bytes of in system self programmable flash program memory.
- 1024 bytes EEPROM
- Programmable serial USART
- 8 Channel, 10 bit ADC
- One 16-bit timer/counter with separate prescaler, compare mode and capture mode.
- Available in 40 pin DIP, 44-pad QFN/MLF and 44-lead QTFP
- Two 8-bit timers/counters with separate pre scalers and compare modes
- 32 programmable I/O lines
- In system programming by on-chip boot program
- Master/slave SPI serial interface
- 4 PWM channels

Chapter 3

PROJECT WORK

VEHICLE TRACKING SYSTEM USING GPS AND GSM

Abstract

Initially the GPS continuously takes input data from the satellite and stores the latitude and longitude values in AT89s52 microcontroller's buffer. If we have to track the vehicle, we need to send a message to GSM device, by which it gets activated. It also gets activated by detecting accident on the IR sensor, by detecting fire on the temperature sensor, by detecting theft connected to vehicle. Parallely deactivates GPS with the help of relay .Once GSM gets activated it takes the last received latitude and longitude positions values from the buffer and sends a message to the particular number or laptop which is predefined in the program. Once message has been sent to the predefined device the GSM gets deactivated and GPS gets activated.

INTRODUCTION

Vehicle tracking system main aim is to give Security to all vehicles. Accident alert system main aim is to rescuing people in accidents. This is improved security systems for vehicles. The latest like GPS are highly useful now a days, this system enables the owner to observe and track his vehicle and find out vehicle movement and its past activities of vehicle. This new technology, popularly called vehicle Tracking Systems which created many wonders in the security of the vehicle. This hardware is fitted on to the vehicle in such a manner that it is not visible to anyone who is inside or outside of the vehicle. Thus it is used as a covert unit which continuously or by any interrupt to the system, sends the location data to the monitoring unit. When the vehicle is stolen, the location data from tracking system can be used to find the location and can be informed to police for further action. Some Vehicle tracking System can even detect unauthorized movements of the vehicle and then alert the owner. This gives an edge over other pieces of technology for the same purpose. This accident alert system in it detects the accident and the location of the accident occurred and sends GPS coordinates to the specified mobile, computer etc. The fire detector circuit in it is used to detect fire in the vehicle, if the temperature inside the vehicle goes above a certain limit then a warning will be automatically send to the intende receiver. The infrared sensor which is additionally interfaced to the microcontroller is used to detect the obstacles and accidents, in any case if any mishap occurs then its warning will be directly send to the intended receiver. When a request by user is sent to the number at the modem, the system automatically sends a return reply to that particular mobile indicating the position of the vehicle in Page | 11 terms of latitude and longitude. A Program has been developed which is used to locate the exact position of the vehicle and also to navigated track of the moving vehicle on Google Map.

APPLICATIONS:

The project that has been introduced here can be used for variety of applications –

1. Car navigation

2. Fleet management/tracking

3. Palmtop, Laptop, PDA, and Handheld

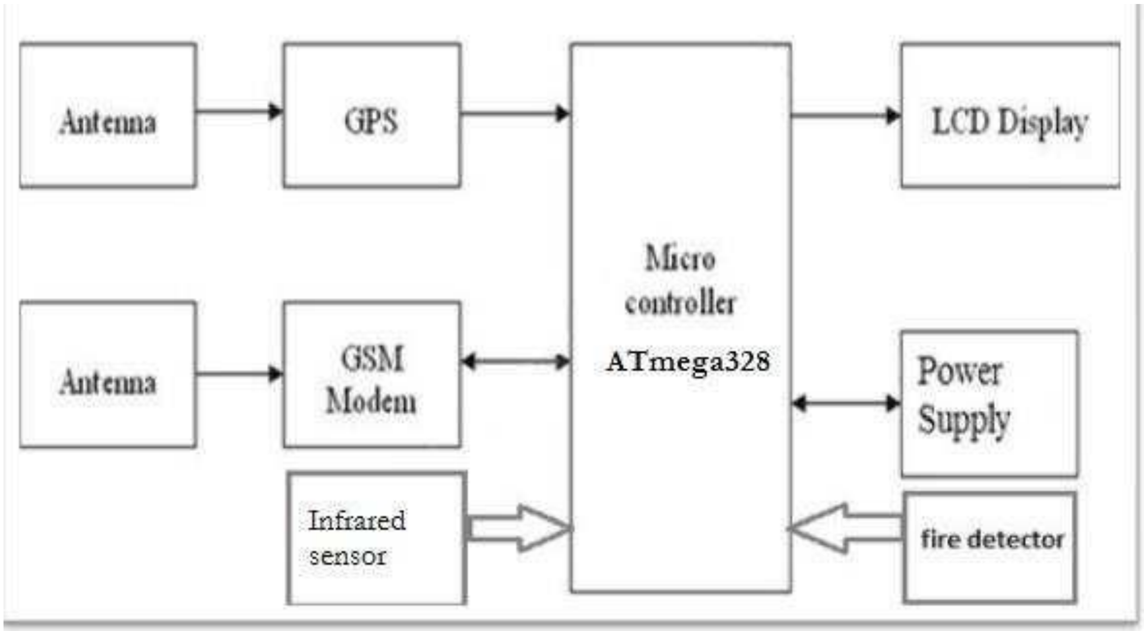
4. Location Based Services enabled devices

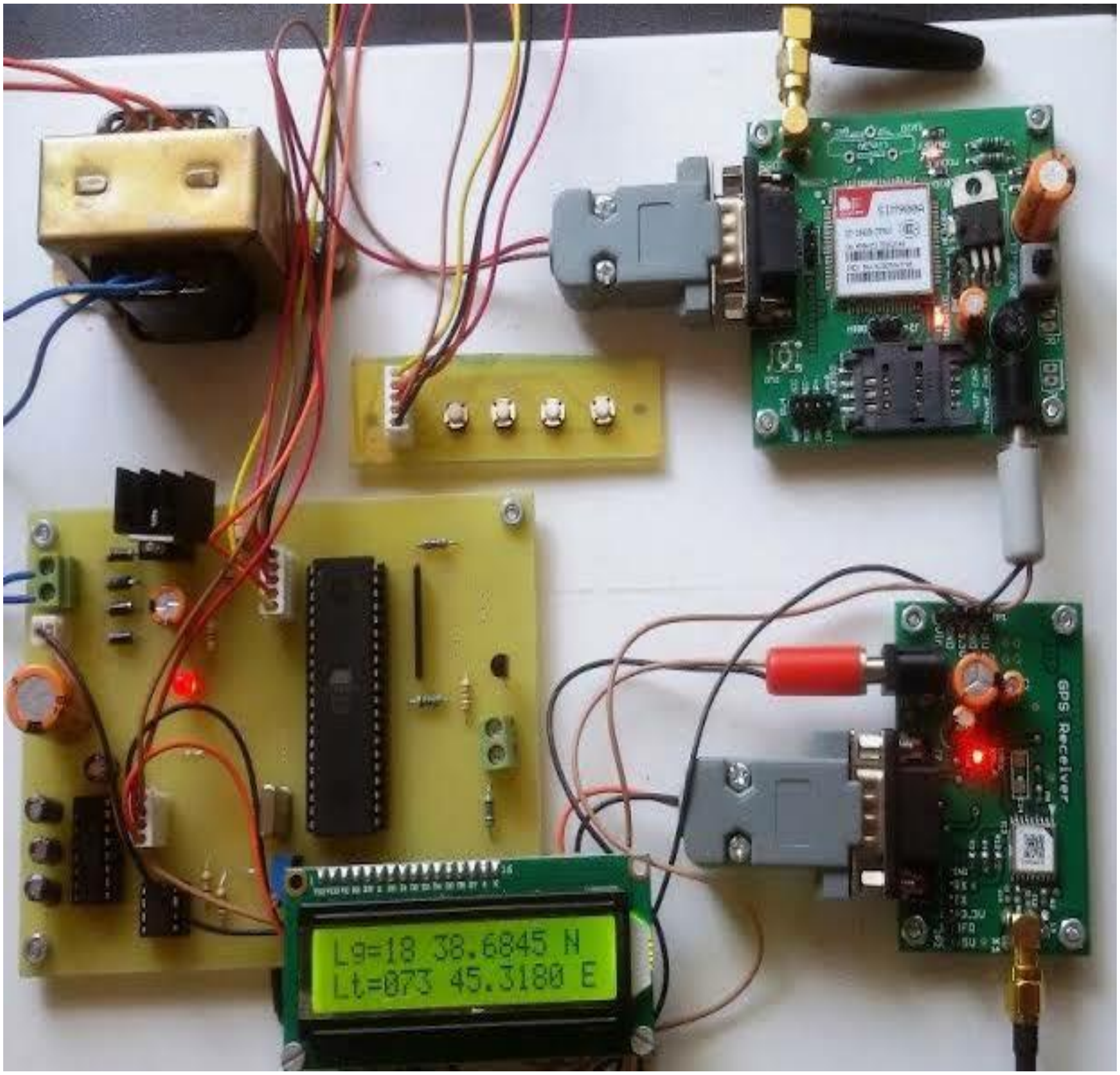
WORKING

The project consists of GPS receiver and GSM modem with a micro controller. The whole system is attached to the vehicle. In the other end (main vehicle station) one GSM mobile phone is attached to the computer with VB application. So the GPS system will send the longitudinal and altitude values corresponding to the position of vehicle to GSM Modem. Imagine the bus has left Bangalore at 6 o clock in the morning. If the officer in charge for that vehicle wants to know where the vehicle is, he will come to the computer and click on the vehicle number on the VB program .The VB program will send an SMS to the vehicle number. The SMS sent would come through the GSM service provider and then reach the vehicle, which is traveling, because the vehicle has a GSM device with sim card. This GSM modem will receive the SMS and send to the microcontroller in the vehicle. The microcontroller will receive this SMS and compare the password and the command. If every thing matches then it will perform the request required by the office. A place name is assigned for each longitude & latitude. The GSM receiver in the vehicle office receives these data & gives to the PC through serial port. The VB program in the PC checks this data with its database & displays the details of the vehicle on the screen. The device is password controlled i.e. person who knows the device password only able to operate. In case of any mishaps such as fire , theft or obstacle ,th e device will automatically will send an alert to the registered number,i.e, the number that is feeded into the memory of microcontroller .

BLOCK-DIAGRAM

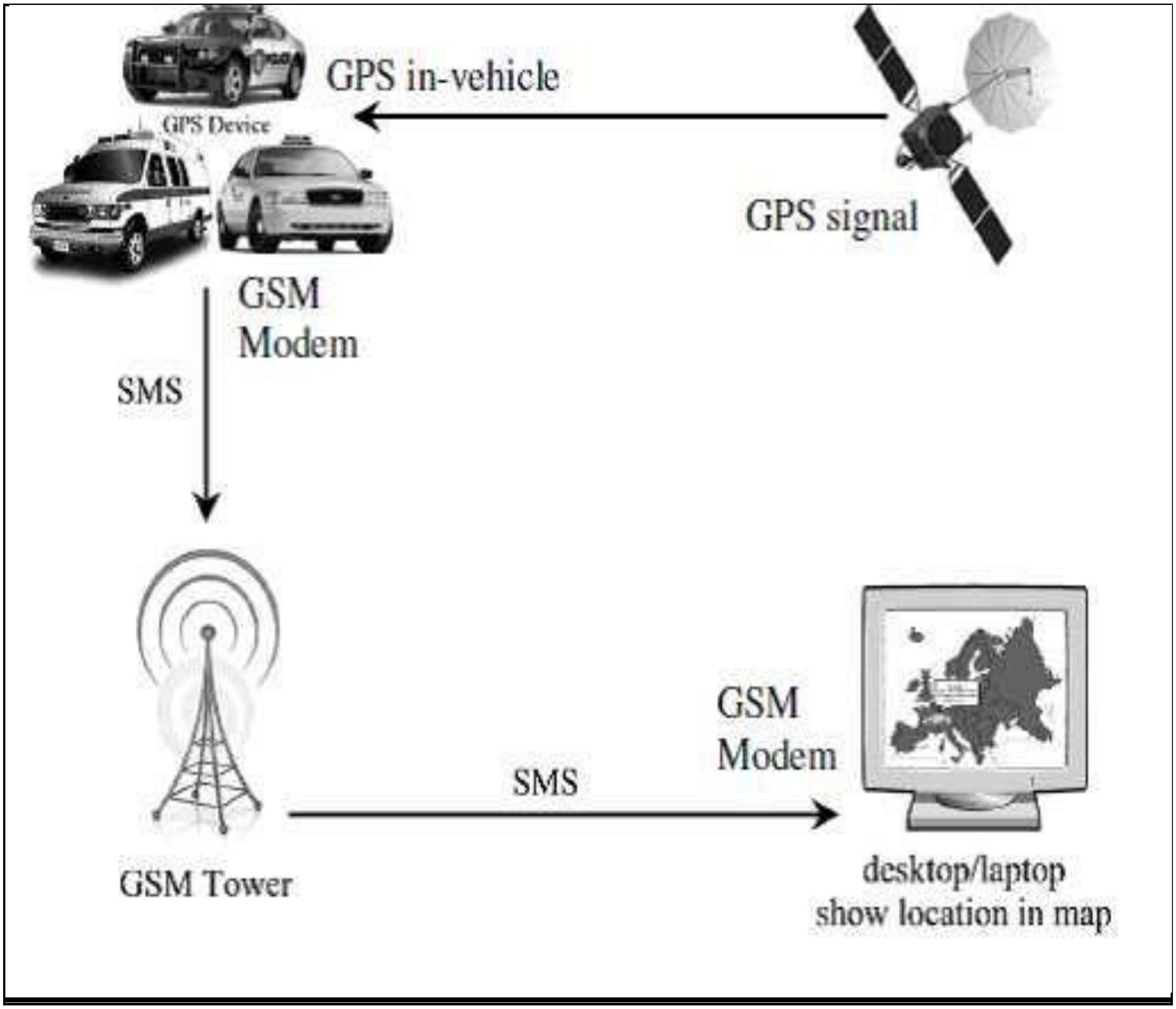
The block diagram of the vehicle tracking system is shown below. The block diagram shows the overall view of the system. The blocks that are connected here are Microcontroller, LCD display, GPS, GSM, Power supply, Infrared sensor, Fire detector.



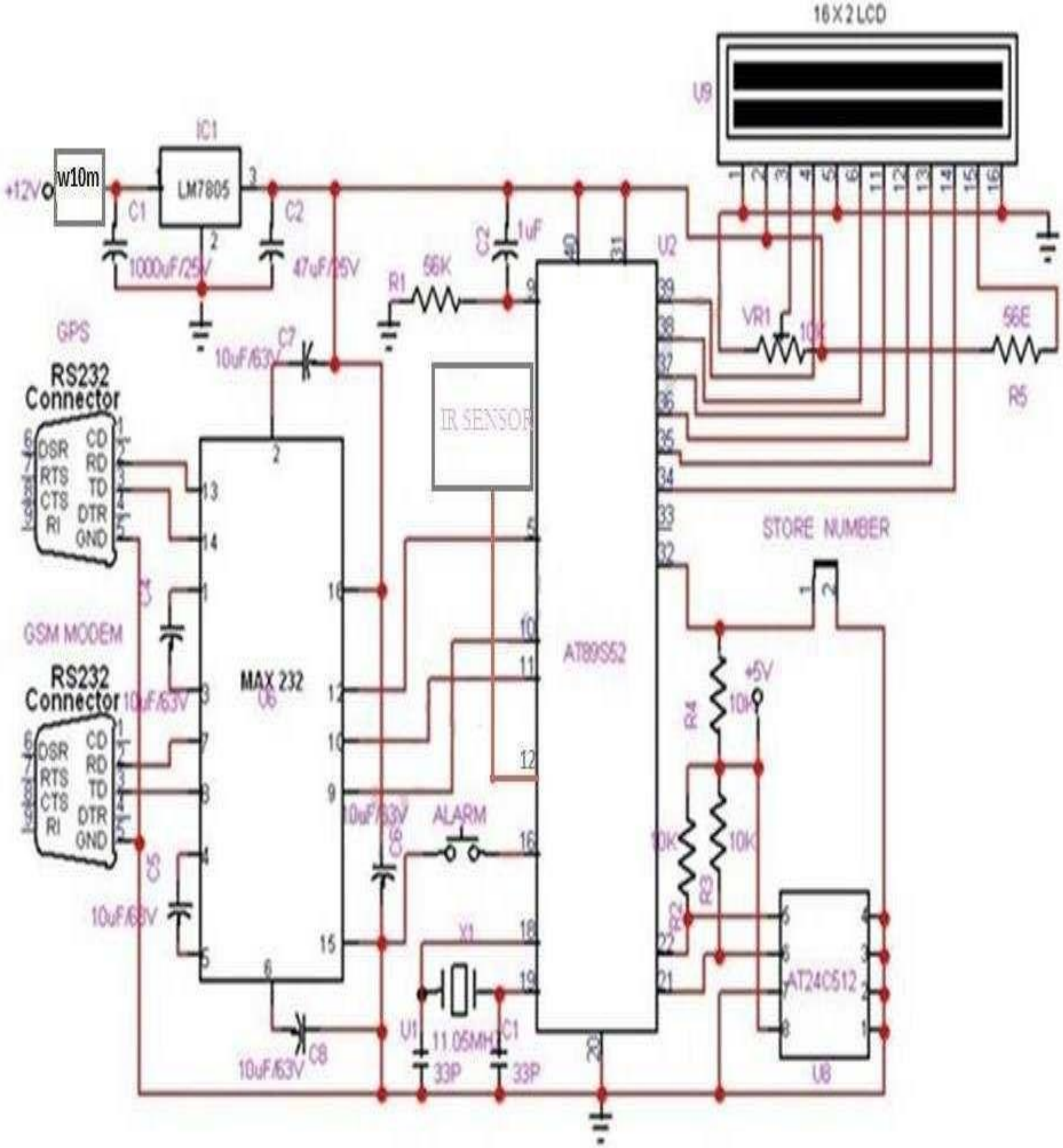


CONCEPT AND OVERVIEW

This vehicle tracking system takes input from GPS and send it through the GSM module to desired mobile/laptop using mobile communication. Vehicle Tracking System is one of the biggest technological advancements to track the activities of the vehicle. The security system uses Global Positioning System GPS, to find the location of the monitored or tracked vehicle and then uses satellite or radio systems to send to send the coordinates and the location data to the monitoring center. At monitoring center various software's are used to plot the Vehicle on a map. In this way the Vehicle owners are able to track their vehicle on a real-time basis. Due to real-time tracking facility, vehicle tracking systems are becoming increasingly popular among owners of expensive vehicles.



CIRCUIT DIAGRAM :



COMPONENT

- Power supply
- Microcontroller ATMEGA328
- GPS module
- GSM module
- Max232
- RS232
- LCD display
- LED
- Infrared sensor
- Fire detector

CODING

Coding is as follows….

```
int8_t
answer;
int
onModuleP
in= 2;
char
aux_strin
g[30];
int flag
```

```
= 0; char
number
[20];
char
realnumbe
r[9];
char
mynumber[
9]; int
a=0; int
b=0; int
c=0;
//Your
phone
number
char phone_number[]="9977514948";
 char data[100]; int
data_size;  char
aux_str[30]; char aux;
int x = 0; char N_S,W_E;
char url[] =
"pruebas.libelium.com";
char frame[200];
 char
latitude
[15];
char
longitud
e[15];
char
altitude
[6];
char
date[16]
; char
```

```
time[7];
char
satellit
es[3];
char
speedOTG
[10];
char
course[1
0]; void
setup(){
mynumber
[0]='9';
mynumber
[1]='9';
mynumber
[2]='7';
mynumber
[3]='7';
mynumber
[4]='5';
mynumber
[5]='1';
mynumber
[6]='4';
mynumber
[7]='9';
mynumber
[8]='4';
mynumber
[9]='8';

    pinMode(onModulePin, OUTPUT);
    Serial.begin(115200);
```

```
power_o
n();
power_o
nGPS();
power_o
nSMS();

    delay(5000);

sendATcommand("AT+CPIN=****", "OK",
2000);

delay
(3000
);
 while( (sendATcommand("AT+CREG?", "+CREG:
    0,1", 1000) || sendATcommand("AT+CREG?",
"+CREG: 0,5", 1000)) == 0 );
     sendATcommand("AT+CLIP=1",
"OK", 1000);
     while (
    start_GPS() ==
    0);
 while (sendATcommand("AT+CREG?", "+CREG: 0,1",
    2000) ==
0);

        // sets APN , user name and password
sendATcommand("AT+SAPBR=3,1,\"Contype\",\"GPRS\"
",
"OK", 2000);
sendATcommand("AT+SAPBR=3,1,\"APN\",\"*******\"",
"OK",
2000);
sendATcommand("AT+SAPBR=3,1,\"USER\",\"*******
\"",
"OK", 2000);
sendATcommand("AT+SAPBR=3,1,\"PWD\",\"*******\"",
"OK",
2000);

    // gets the GPRS bearer while
    (sendATcommand("AT+SAPBR=1,1", "OK", 20000)
    == 0)
    {
        delay(5000);
    }
delay(1000);
while(Serial.available(
    ) != 0)
    {
      Serial.read();

}
}
void loop(){       answer =
sendATcommand("", "+CLIP", 1000);
    //Detect
    incomming call
    if (answer ==
    1)
    {
        Serial.println("Incoming call");
           if (
          flag
          ==
          0){
             for (int i=0;
              i<19; i++){
                    // read the incoming byte:
               while
(Serial.available() == 0)
{
delay (50);
               }
                    //Stores phone number
               number[i] = Serial.read();

              }

Serial.flush();
flag = 1;
          }
          //Stores phone calling
          number for (int i=0;
          i<=14; i++){
            if(number[i]==
'"'){            i++;
realnumber[0]=number[i];
i++;
realnumber[1]=number[i];
i++;
realnumber[2]=number[i];
i++;
realnumber[3]=number[i];
i++;
realnumber[4]=number[i];
i++;
realnumber[5]=number[i];
i++;
realnumber[6]=number[i];
i++;
realnumber[7]=number[i];
i++;
realnumber[8]=number[i];
              break;
             }
    }
    //Check phone
    number for
    (int
    i=0;i<9;i++){
      if (realnumber[i] == mynumber[i]){
          a++;
          if(
          a==9
          ){
             Serial.println("Correct
number");             sendATcommand("ATH",
               "OK", 1000); if(b==1){
```

```
b=0;
}else{
b=1;
c=1;


                    }
                    b
                    r
                    e
                    a
                    k
                    ;
                  }
                }else{
                  Serial.println("Wrong
                  number"); break;

}
}
a=0;
answer=0
;
flag =
0;
}
//Send SMS once and
position to HTTP
if (b==1){
get_GPS();
send_HTTP();
    delay(500);
if (c==1){
      sendSMS();
delay(100);
      c=0;
      }
     }
}
void
power
_on()
{

uint8_t
answer=0;

digitalWrite(onModulePin
,HIGH);     delay(3000);
digitalWrite(onModulePin
,LOW);
 while(answer == 0){     // Send AT every two
    seconds
and wait for the answer        answer
= sendATcommand("AT", "OK", 2000);
    }
}  int8_t sendATcommand(char*
ATcommand, char*
expected_answer, unsigned int
timeout){
     uint8_t
x=0,  answer=0;
    char

response[100];
unsigned long
previous;
```

```
     memset(response, '\0', 100);     //
Initialice the string        delay(100);
     while( Serial.available() > 0)
    Serial.read();     // Clean the input
buffer
     if
    (ATcommand[0]
    != '\0')
    {
        Serial.println(ATcommand);    // Send
the AT command

}

x
=
0
;
    previous = millis();

    // this loop waits for the answer do{
    if(Serial.available() != 0){    // if
        there are
data in the UART input buffer, reads it and
checks for the asnwer            response[x] =
Serial.read();
            //Serial.print(response[x]);
            x++; if (strstr(response,
            expected_answer) != NULL)    //
check if the desired answer (OK) is in the
response of the module            {
                answer = 1;
            }
        }
    }while((answer == 0) && ((millis() -
previous) < timeout));    // Waits for the
asnwer with time out
 return

    ans
    wer
    ;
}  void
power_on
GPS(){

uint8_t
answer=0;

        // checks if the module is
started     answer =
    sendATcommand("AT", "OK",
2000); if (answer == 0)
    {
        // power on pulse

digitalWrite(onModulePin,HIGH);
delay(3000);
digitalWrite(onModulePin,LOW);

        // waits for an answer from the module
        while(answer == 0){
//            Send AT every two seconds and wait
for                                        the
answer            answer =
sendATcommand("AT", "OK", 2000);
        }
    }
```

```
}
int8_t
start_GP
S(){
 unsigned long
    previous
     previous
= millis();
        // starts the GPS
sendATcommand("AT+CGPSPWR=1", "OK",
2000);     sendATcommand("AT+CGPSRST=0",
"OK", 2000);

    // waits for fix GPS while((
    (sendATcommand("AT+CGPSSTATUS?", "2D
    Fix",
5000) ||
sendATcommand("AT+CGPSSTATUS?", "3D Fix", 5000))
==
0 ) &&
        ((millis() - previous) < 90000));
 if ((millis() - previous)
    < 90000)
    {
        r
        e
        t
        u
        r
        n
        1
        ;
    }
    else
    {
        return 0;
    }
}
int8_t
get_GP
S(){
     int8_t
counter, answer;
    long
previous;

    // First get the NMEA string //
    Clean the input buffer while(
    Serial.available() > 0)
    Serial.read();
        // request Basic string
sendATcommand("AT+CGPSINF=0",
"AT+CGPSINF=0\r\n\r\n",
2000);      counter = 0;      answer = 0;
memset(frame, '\0', 100);     // Initialize the
string      previous = millis();
    // this loop waits for the NMEA
    string do{
 if(Serial.available() !=
        0){
            frame[counter] = Serial.read();
counter++;
            // check if the desired answer is in
            the
response of the module if
            (strstr(frame, "OK")
            != NULL)
            {
                answer = 1;
            }
        }
        // Waits for the asnwer with time out
        } while((answer == 0) && ((millis() -
    previous) < 2000));

frame[counter-3]
= '\0';

        // Parses the string     strtok(frame,
",");     strcpy(longitude,strtok(NULL, ","));
// Gets longitude
strcpy(latitude,strtok(NULL, ",")); // Gets
latitude     strcpy(altitude,strtok(NULL,
".")); // Gets altitude     strtok(NULL, ",");
strcpy(date,strtok(NULL, ".")); // Gets date
strtok(NULL, ",");     strtok(NULL, ",");
strcpy(satellites,strtok(NULL, ",")); // Gets
satellites     strcpy(speedOTG,strtok(NULL,
","));  // Gets speed over ground. Unit is
knots.
    strcpy(course,strtok(NULL, "\r")); // Gets
course

convert2Degrees(lati
tude);
convert2Degrees(long
itude);


    ret
    urn
    ans
    wer
    ;
}

/* convert2Degrees ( input ) - performs the
conversion from input
* parameters in  DD°MM.mmm’ notation to
  DD.dddddd° notation.
 *
* Sign '+' is set for positive
  latitudes/longitudes (North, East)
* Sign '-' is set for negative
  latitudes/longitudes
(South, West)
*  */ int8_t convert2Degrees(char* input){
 float


    deg
    ;
    flo
    at
    min
    ute
    s;
    boolean neg = false;

    //auxiliar
    variable char
    aux[10];
 if (input[0]
    == '-')
    {        neg = true;
strcpy(aux, strtok(input+1,
"."));
    } else     {
    strcpy(aux, strtok(input,
"."));
    }

        // convert string to integer and add it
to final float variable
    deg = atof(aux);
     strcpy(aux,
strtok(NULL, '\0'));
minutes=atof(aux);
minutes/=1000000; if (deg
    < 100)
    {
minutes +=
deg;
deg = 0;


    }
    e
    l
    s
    e
    {        minutes
+= int(deg) % 100;
deg = int(deg) / 100;
    }

        // add minutes to degrees
deg=deg+minutes/60;
 if (neg
    ==
true)
{
deg*=-
1.0;
    }
neg =
false
;
 if(
    deg
    < 0
    ){
        neg =
true;
deg*=-1;
    }
     float
    numberFloat=deg;  int
    intPart[10]; int
    digit;  long


    newNumber=(long)number
    Float;   int size=0;

    w
    h
    i
    l
    e
    (
    1
    )
    {
        size=size+1;
digit=newNumber%10;
newNumber=newNumber/10
;
        intPart[size-
            1]=digit;
if (newNumber==0){
break;
        }
    }



    i
    n
    t
    i
    n
    d
    e
    x
    =
    0
    ;
    i
    f
    (
    n
    e
    g
    )
    {
        index++;
input[0]='-';
    } for (int i=size-1;
    i >= 0; i--)
    {
input[index]=intPart[i]+'
0';         index++;
    }
input[inde
x]='.';
index++;
     numberFloat=(numberFloat-
(int)numberFloat); for (int i=1;
    i<=6 ; i++)
    {
numberFloat=numberFloat*10;
digit=
(long)numberFloat;
numberFloat=numberFloat-
digit;
input[index]=char(digit)+48
;         index++;      }
input[index]='\0';
}
void
send_H
TTP(){
```

```
uint8_t
answer=0;
            // Initializes HTTP service
answer = sendATcommand("AT+HTTPINIT", "OK",
    10000); if (answer == 1)
    {
                    // Sets CID parameter
answer =
sendATcommand("AT+HTTPPARA=\"CID\",1",
"OK", 5000);
        if
(answer == 1)
{
            // Sets url
            sprintf(aux_str,
"AT+HTTPPARA=\"URL\",\"http://%s/demo_sim908.p
hp?", url);             Serial.print(aux_str);
sprintf(frame,
"visor=false&latitude=%s&longitude=%s&altitude=%
s&time=%s&s atellites=%s&speedOTG=%s&course=%s",
latitude, longitude, altitude, date, satellites,
speedOTG, course);
Serial.print(frame);            answer =
sendATcommand("\"", "OK", 5000); if (answer ==
            1)
            {
                // Starts GET action
                answer =
sendATcommand("AT+HTTPACTION=0",
"+HTTPACTION:0,200",

                30000
                ); if
                (answw
                er ==
                1)
                {

                    Serial.println(F("Done!"));


                }
                e
                l
                s
                e
                {
                    Serial.println(F("Error
getting url"));
                }



            }
            e
            l
            s
            e
            {
                Serial.println(F("Error setting
the url"));             }


        }
        e
        l
        s
        e
        {
            Serial.println(F("Error setting the
CID"));
        }

    
    }
    e
    l
    s
    e
    {
        Serial.println(F("Error
initializating"));
    }
sendATcommand("AT+HTTPTERM", "OK",
5000);
}  void
power_on
SMS(){

uint8_t
answer=0;

        // checks if the module is
started     answer =
    sendATcommand("AT", "OK",
2000); if (answer == 0)
    {
                // power on pulse
digitalWrite(onModulePin,HIGH);
delay(3000);
digitalWrite(onModulePin,LOW);

        // waits for an answer from the module
        while(answer == 0){     // Send AT every
        two
```

```
seconds and wait for the answer
answer = sendATcommand("AT", "OK", 2000);
        }
    }

}  void sendSMS(){
sendATcommand("AT+CPIN=****",
"OK", 2000);



delay(
3000);
    Serial.println("Connecting to the network...");
 while( (sendATcommand("AT+CREG?", "+CREG: 0,1",
    500) ||
            sendATcommand("AT+CREG?", "+CREG:
0,5", 500))
=
=
0
)
;
    Serial.print("Setting SMS mode...");
    sendATcommand("AT+CMGF=1", "OK", 1000);
// sets the SMS mode to text
    Serial.println("Sending SMS");
         sprintf(aux_string,"AT+CMGS=\"%s\"",
phone_number);      answer =
sendATcommand(aux_string, ">", 2000);     //
send the SMS number if (answer == 1)
        {
        Serial.print("Theft Alert:");

              
              Serial.print("
              Latitude: ");
              int i = 0;

while(latitude[i]!=0){
Serial.print(latitude[i]);
i++;
              }
              Serial.print(" /
Longitude: ");               i =
              0;
while(longitude[i]!=0){
              Serial.print(longitude[i]);
              i++;
              }
        Serial.write(0x1A);
answer = sendATcommand("", "OK",
        20000); if (answer == 1)
        {
            Serial.print("Sent ");

        }
        e
        l
        s
        e
        {
            Serial.print("error ");
        }

    }
    e
    l
    s
    e
    {
        Serial.print("error ");
        Serial.println(answer, DEC);
    }

}
```

ADVANTAGES:

Commercial fleet operators are by far the largest users of vehicle tracking systems. These systems are used for operational functions such as routing, security, dispatch and collecting on-board information.

These are also used for fire detector in large vehicles like train, bus etc. because the vehicle like train contains large number of people and the sending alert of fire accident can save many lives.

The applications for this project are in military, navigation, automobiles, aircrafts, fleet management, remote monitoring, remote control, security systems, tele services, etc.

- Fleet monitoring
- Vehicle scheduling
- Route monitoring
- Driver monitoring
- Accident analysis
- Geo-fencing geo-coding

These are just a few advantages of the project that has been introduced in this report . We can interface more number of sensors in order to serve multiple purposes. The microcontroller that has been used in this project have inbuit ADCs and hence the controller is capable of accepting analog inputs, which is the biggest advantage. Since all real world signals are analog in nature, by incorporating different sensors required purpose can be served.

FUTURE SCOPE :

- We can use the EEPROM to store the previous Navigating positions up to 256 locations and we can navigate up to N number of locations by increasing its memory.

- We can reduce the size of the kit by using GPS+GSM on the same module.

- We can increase the accuracy up to 3m by increasing the cost of the GPS receivers.

- We can use our kit for detection of bomb by connecting to the bomb detector.

- With the help of high sensitivity vibration sensors we can detect the accident. whenever vehicle unexpectedly had an accident on the road with help of vibration sensor we can detect the accident and we can send the location to the owner, hospital and police.

- We can use our kit to assist the traffic. By keeping the kits in the entire vehicles and byknowing the locations of all the vehicles.

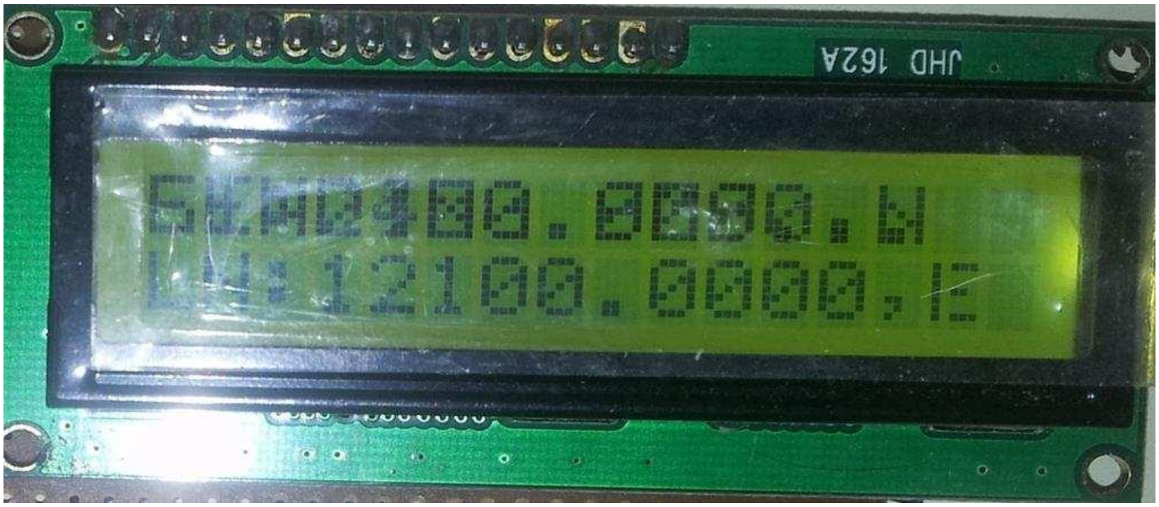
- If anybody steals our car we can easily find our car around the globe. By keeping vehicle positioning vehicle on the vehicle.

RESULT:

- Whenever accident or theft of the vehicle is occurred then the

device sends message to given mobile device.

- **Message for theft** : “Theft alert latitude: 2400.0090, N longitude: 12100.0000, E time: 12:00”
-
- **Message for accident** : “Accident alert latitude: 2400.0090, N longitude: 12100.0000, E time: 12:00”
- This system shows the location of vehicle on the lcd connected to it also just to make sure the working condition of the microcontroller.



2 CONCLUSION :

Vehicle tracking system makes better fleet management and which in turn brings large profits. Better scheduling or route planning can enable you handle larger jobs loads within a particular time. Vehicle tracking both in case of personal as well as business purpose improves safety and security, communication medium, performance monitoring and increases productivity. So in the coming year, it is going to play a major role in our day-to-day living.

Main motto of the project is to incorporate different types of sensors so that they help in decrease the chances of losing life in such accident which we can't stop from occurring. Whenever accident is alerted the paramedics are reached to the particular location to increase the chances of life. This device invention is much more useful for the accidents occurred in deserted places and midnights. This vehicle tracking and accident alert feature plays much more important role in day to day life in future.

REFERENCES :

[] R.S Gaonkar, Microprocessor Architecture Programming and Application , Wiley Eastern Ltd, New Delhi.

[] Krishna Kant, Microprocessor and microcontroller , Eastern Company Edition , New Delhi 2007.

[] Daniel. W. Lewis, Fundamental of embedded software , prentice hall of India, 2004.

[] William Stalling, Wireless Communication and Networks , nd edition, prentice hall of India, 2005.

[]Chen, H., Chiang, Y. Chang, F. H. Wang, Toward Real-Time Precise Point Positioning: Differential GPS Based on IGS Ultra Rapid Product, SICE Annual Conference, The Grand Hotel, Taipei, Taiwan August 18-21, (2010).

[] Asaad M. J. Al-Hindawi, Ibraheem Talib, Experimentally Evaluation of GPS/GSM Based System Design , Journal of Electronic Systems, Volume Number, 2 June, 2012.

[] Kunal Maurya , Mandeep Singh, Neelu Jain, Real Time Vehicle Tracking System using GSM and GPS Technology- An Anti-theft Tracking System ,)nternational Journal of Electronics and Computer Science Engineering, ISSN 2277-1956/V1N3-1103-1107 .

[] Vikram Kulkarni & Viswaprakash Babu, embedded smart car security system on face detection , special issue of)JCCT,)SSN Online : 22310371, ISSN(Print):0975-7449,volume-3, issue-1.