

## Secured Luxuries Transporting System

<sup>1</sup>R.Venkatesh, <sup>2</sup>M.Chandra Sekhar, <sup>3</sup> K.V.N.S.Sandeep

1. Final year B.Tech ECE student, St. Ann's ENGINEERING COLLEGE, Chirala, A.P.

2. Final year B.Tech ECE student, St. Ann's ENGINEERING COLLEGE, Chirala, A.P.

3. Final year B.Tech ECE student, St. Ann's ENGINEERING COLLEGE, Chirala, A.P.

---

**Abstract:** In Today's world safety and security plays an important role, hence we tend to provide a good safety and security system while travelling. Vehicles are important in today's fast paced society. Hence, acquiring a vehicle now a days is considered a necessity, compared to the past where it is considered as a luxury. In this thriving society, more and more vehicles are produced to meet the increasing demands of people and business from all the corners of the world. Here comes the necessity to provide more and more safety and security features to them. Hence this project aim is to design an embedded system for vehicle cabin safety and security by modifying and integrating the existing modules. This monitors the level of the toxic gases such as Co, LPG and alcohol inside the vehicle and provides alert information in the form of alarm during the critical situations. And also send SMS to the authorized person through the GSM. An IR Sensor is used to detect the static obstacle in front of the vehicle and the vehicle gets stopped if any obstacle is detected. This may avoid accidents due to collisions of vehicles with any static obstacle. This includes vehicle reversing mechanism too.

**Keywords:** DC, GSM, Embedded System, Microcontrollers, Sensors.

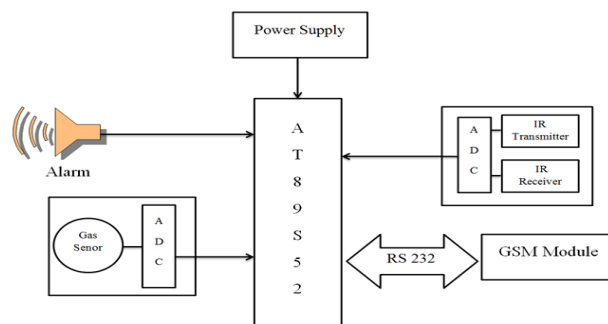
---

### I. Introduction:

Vehicle is the first place where safety starts. Hence we must try to equip it with the latest technologies and measures to make it as a machine and also to keep our self and our loved one safe. Always remember that safety starts and ends with the person who drives the vehicle. The cars and trucks found today have high range of built in safety accessories to protect their passengers. Before it used to be just seat belts, but now more features have been included which are more advanced and efficient than seat belts. Warning alert and alarms are other security systems incorporated in the cars and trucks to alert us about various factors like exceeding speed limiter, Smoke alarms. These are designed to make the passengers aware of crossing the limitation which is important in most of the time and in most cases. In the same way here an embedded system has been designed to make the journey of the passengers inside a vehicle safe and secure with various recently found safety and security measures.

### II. Hardware Description:

In our proposed system an obstacle sensor i.e. the IR module is mounted in front of the car, such that the distance between the obstacles in front of vehicle is normally of 3ft. is sensed by the sensor module and instructs the microcontroller about the obstacles ahead. The Microcontroller then alerts the driver with an alarm and control the vehicle by stopping. In the same way a gas sensor is mounted inside the vehicle such that it senses the presence of the gas inside the vehicle cabin and informs the microcontroller if there is any increase in the level of the toxic gases. The microcontroller then alerts the passengers and the driver inside the vehicle with an alarm and also sends a SMS to the authorized user through GSM. The block diagram of the proposed system is shown in the figure.1 the proposed system is developed using Atmel AT89S52 microcontroller. The system has two modules namely Gas sensing module and the obstacle detection module and they are interfaced with the microcontroller. An ADC is provided in these two modules for converting the analog acquired data into the digital data for processing by the microcontroller. The GSM module is connected with the microcontroller through RS232 and the alarm is interfaced with the microcontroller to raise the alarm during the critical situations.



## 2.1. The Gas Sensing Module

The gas sensing module is used to sense the presence of gases such as CO, LPG, ALCOHOL, and other toxic gases inside the vehicle. If critical levels of gases were found, that is if the CO exceeds 20PPM and the level of LPG exceeds 10,000PPM and the presence of alcohol is detected then the digital data from the gas sensing module is sent to the microcontroller which displays the information about the gas leakage inside the vehicle and produces an alarm to alert the persons inside the vehicle. It also sends a text message to the authorized person through the GSM modem connected to the microcontroller such that remedy measures could be taken by the authorized person and to give proper medical treatment to them if required [10].

### 2.1.1. Gas Sensors

A gas detector is a device which detects the presence of various gases with in an area usually a part of a system to warn about gases which might be harmful to humans or animals. Gas detectors can be used to detect combustible, flammable and toxic gases and also oxygen depletion [10]. This type of device is used widely in industry and can found in a variety of locations such as an oil rings, to monitor the manufacturing process and in emerging technologies such as photovoltaic. They may also be used in firefighting and the gas detectors are usually battery operated. The architecture of the gas sensing module is shown in figure.2

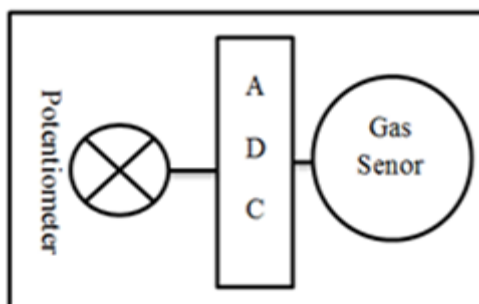


Fig 2. Architecture of Gas Sensing Module

The sensing module transmits warning via a series of audible and visible signals such as alarms and flashing lights, when dangerous levels of a gas vapors are detected. As detectors measure a gas concentration, the sensor responds to a calibration gas, which serves as the reference point or scale. As a sensor's detection exceeds a pre-set alarm level, the alarm or signal will be activated. As units, gas detectors are produced as portable or stationary devices. Originally, detectors were produced to detect a single gas, but modern units may detect several toxic or combustible gases, or even a combination of both types.

### 2.1.2. MQ-7-Gas Sensor

A MQ-7 gas sensor is used in this system which is a device that detects the presence of the carbon monoxide (CO) and LPG and several of their gases in the environment. CO is a colorless and odorless compound produced by incomplete combustion. It is often referred to as the "silent killer" because it is virtually undetectable without using detection technology. Elevated levels of CO can be dangerous to humans depending on the amount present and length of exposure. Smaller concentration can be harmful over longer periods of time while increasing concentration required diminishing exposure times to be harmful. MQ-7 sensors are designed to measure CO levels over a time and sounds an alarm before dangerous levels of CO accumulate in an environment, and therefore giving the people adequate warning to safely ventilate the area or evacuate. Some system connected detectors also alert a monitoring service that can dispatch emergency services if necessary.

The interfacing of the sensor with the micro controller and the developed gas sensor module is shown in fig3 (a) and fig3 (b).

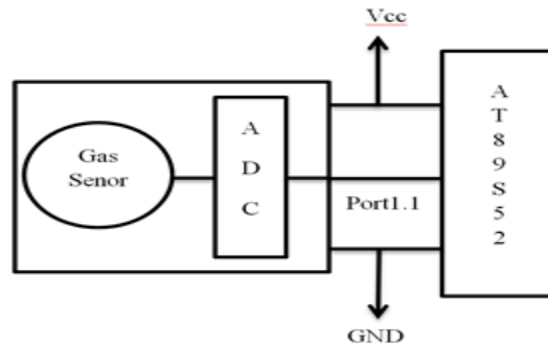


FIG.3 (a)

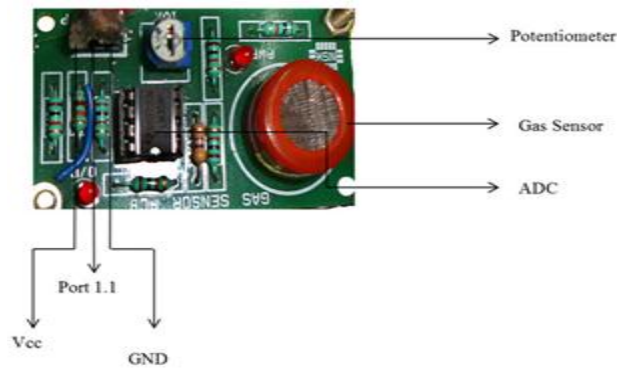


FIG.3 (b)

Fig. (a) Interfacing Gas Sensor with Micro controllers.  
(b) The developed Gas Sensor Module

The MQ-7 sensor also detects the presence of a dangerous LPG leak in our home or in a service station, storage tank environment and even in vehicle which uses LPG gas as its fuel. This unit can be easily incorporated into an alarm unit, sound an alarm or provide a visual indication of the LPG concentration. The sensor's excellent sensitivity combined with a quick response time. When the target combustible gas exists, the sensor's conductivity is higher along with the gas concentration raising. A simple electronic circuit is used to convert the change in conductivity to its corresponding output signal of a gas concentration. This module may be used in various application areas such as in Home, Industry and in a Car [17]. The features of the sensor are,

- High Sensitivity to Carbon monoxide, LPG, propane etc.
- Stable and Long life

### 3.2. Obstacle Sensing Module

The obstacle sensing module is used to sense the static obstacles in front of the vehicle such that, accidents due to unwanted parking of the vehicles and collision with trees and other objects especially during the night time could be avoided. These obstacles could be detected using various methods such as ultrasonic sensors etc. [6]. The working principle of the obstacle sensor is shown in the fig (4).

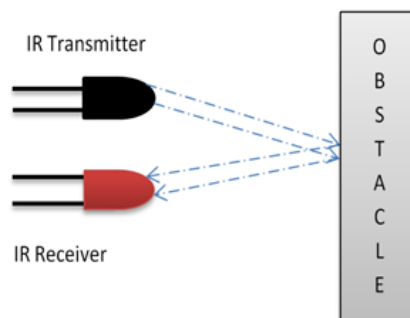


Fig.4 IR Sensing Module

An IR module is used to detect the vehicle in front obstacles, it consists of an IR transmitter and a Receiver which combines to form an obstacle sensing module. The architecture of IR module is shown in fig. 5.

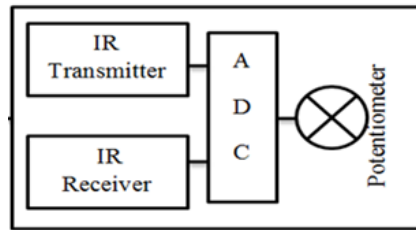
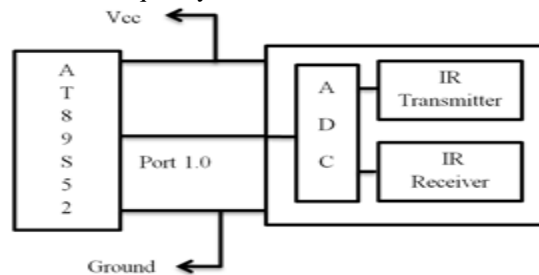


Fig.5 Architecture of the IR module

**2.2.1. IR Transmitter**

In general, the basic building block of any IR transmitter modulation of the information signal with the carrier signals, because the receiver modules which are available off-the-shelf are made for a particular carrier frequency. So it is clear that when we choose a particular IR receiver module, we also need to transmit the modulated wave with the same carrier frequency of an IR receiver module.



**2.2.2. IR Receiver**

It is quite simple to construct an IR receiver with readily available off-the-shelf modules. These modules are nothing but the IC packages, referred as TSOP (Thin small-outline package). In this work, the receiver is designed for 38 kHz carrier signal; hence the selected IC should work for the same frequency. The IC TSOP4838 will serve as a receiver module, which is compatible with both TTL and CMOS logic. This means that we can directly get digital signal from the receiver module and then connect it to the microcontroller. The interfacing of IR module with the microcontroller and the developed module is shown in the fig. 6(a) and fig6 (b).

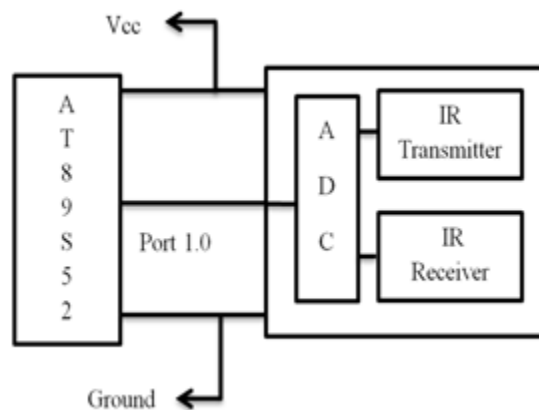


Fig. 6(a)

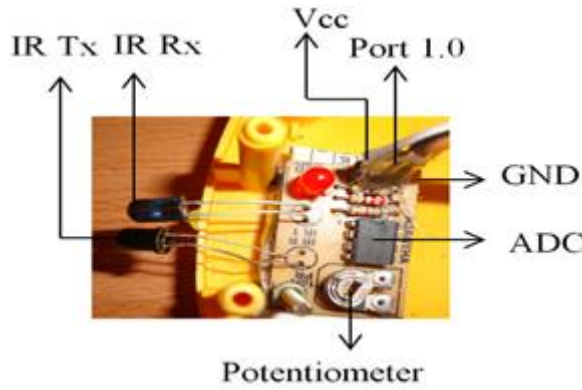


Fig.6 (b)

Fig.6 (a) Interfacing IR sensor with AT89S52. (b) The developed Obstacle detection module.

### 2.3. Microcontroller

The microcontroller forms the heart of an embedded system. The AT89S52 is the microcontroller used here which is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel’s high-density non-volatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly- flexible and cost-effective solution to many embedded control applications. The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes.

The pin diagram of the AT89S52 is shown in the fig. 7. The microcontroller is always in the active mode, if the gas sensor senses a gas and finds a critical situation then the information is sent to the microcontroller. The microcontroller first provokes an alarm to alert the passengers inside the vehicle and also an alert message to the authorized user is sent in the form of SMS through GSM. The microcontroller uses AT+CMGS command to send the SMS where the GSM is connected to the microcontroller through RS232 cable.

(T2) P1.0	1	40	VCC
(T2 EX) P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
(MISO) P1.5	6	35	P0.4 (AD4)
(MISO) P1.6	7	34	P0.5 (AD5)
(SCK) P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	EA/VPP
(TXD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(WR) P3.6	16	25	P2.4 (A12)
(RD) P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

Fig. 7 Pin Description of the AT89S52

### 2.4. GSM Module

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. A GSM modem exposes an interface that allows applications such as SMS to send and receive messages over the modem interface. The mobile operator charges for this message sending and receiving as if it was performed directly on a mobile phone. To perform these tasks, a GSM modem must support an “extended AT command set” for sending/receiving SMS messages, as defined in the ETSI GSM 07.05 and 3GPP TS 27.005 specifications. Due to some compatibility issues that can exist with mobile phones, using a

dedicated GSM modem is usually preferable to a GSM mobile phone. It should also be noted that not all phones support the modem interface for sending and receiving SMS messages. In particular, most smart phones, including Blackberries, iPhone, and Windows Mobile devices, do not support this GSM modem interface for sending and receiving SMS messages at all. Additionally, Nokia phones that use the S60 (Series 60) interface, which is Symbian based, only support sending SMS messages via the modem interface, and do not support receiving SMS via the modem interface [15]. The interfacing between the GSM and the Microcontroller and the developed module is shown in the fig. 8(a) and fig.8 (b).

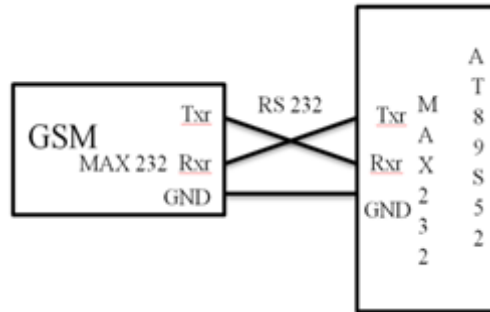


Fig.8 (a)Interfacing GSM with AT89S52



Fig8. (b) The developed GSM module

### III. Software description

#### 3.1. Embedded C

An embedded system is the one which is designed to perform a specific task and the embedded software rules the entire system. This software for a particular embedded system could be developed using various embedded programming languages. But embedded C is the well-known embedded programming language.

Use of C in embedded systems is driven by following advantages

It is small and reasonably simpler to learn, understand, program and debug.

- C Compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.
- Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/ microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems.
- As C combines functionality of assembly language and features of high level languages, C is treated as a ‘middle-level computer language’ or ‘high level assembly language’.
- It is fairly efficient.
- It supports access to I/O and provides ease of management of large embedded projects.

Many of these advantages are offered by other languages also, but what sets C apart from others like Pascal, FORTRAN, etc. is the fact that it is a middle level language; it provides direct hardware control without sacrificing benefits of high level languages. Compared to other high level languages, C offers more flexibility because C is relatively small, structured language; it supports low-level bit-wise data manipulation. Compared to assembly language, C Code written is more reliable and scalable, more portable between different platforms (with some changes) [13]. Moreover, programs developed in C are much easier to understand, maintain and debug. Also, as they can be developed more quickly, codes written in C offers better productivity. C is based on the philosophy ‘programmers know what they are doing’; only the intentions are to be stated explicitly. It is easier to write good code in C and convert it to an efficient assembly code (using high quality compilers) rather than writing an efficient code in assembly itself. Benefits of assembly language programming over C are negligible when we compare the ease with which C programs are developed by programmers.



### 3.2. Keil Compiler

After developing the software, it must be downloaded to the microcontroller through any one of the downloading tools such as universal programmer. Hence the program should be cross compiled before downloading it into the microcontroller; the keil compiler comes into act at this place. Keil Software provides us with software development tools for the 8051 family of microcontrollers. With these tools, we can generate embedded applications for the multitude of 8051 derivatives. Keil provides following tools for 8051 family development.

1. C51 Optimizing C Cross Compiler,
2. A51 Macro Assembler,
3. 8051 Utilities (linker, object file converter, library manager),
4. Source-Level Debugger/Simulator,
5.  $\mu$  Vision for Windows Integrated Development Environment.

The keil 8051 tool kit includes three main tools, assembler, compiler and linker.

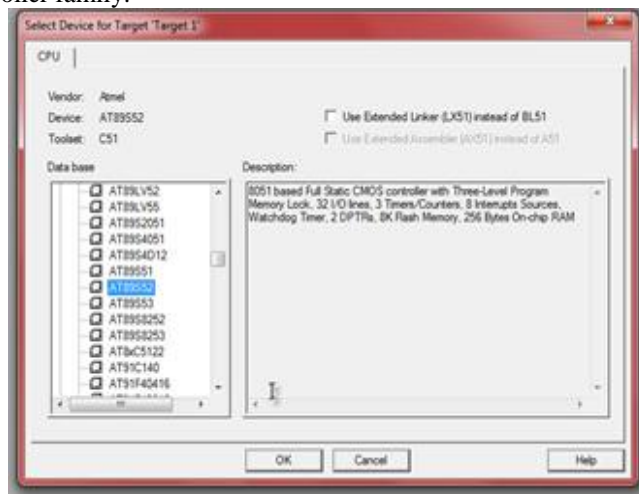
- An assembler is used to assemble the 8051 assembly program
- A compiler is used to compile the C source code into an object file
- A linker is used to create an absolute object module suitable for the in-circuit emulator.

A project development cycle: - these are the steps to develop a project using keil

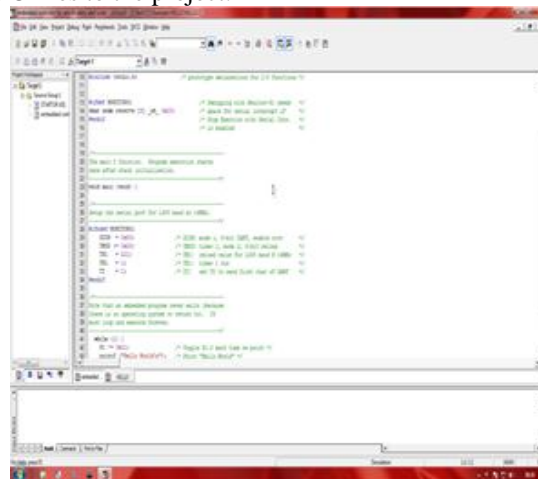
1. Create source files in C or assembly.
2. Compile or assemble source files.
3. Correct errors in source files.
4. Link object files from compiler and assembler.
5. Test linked application.

Programming and Downloading the Hex file to the microcontroller,

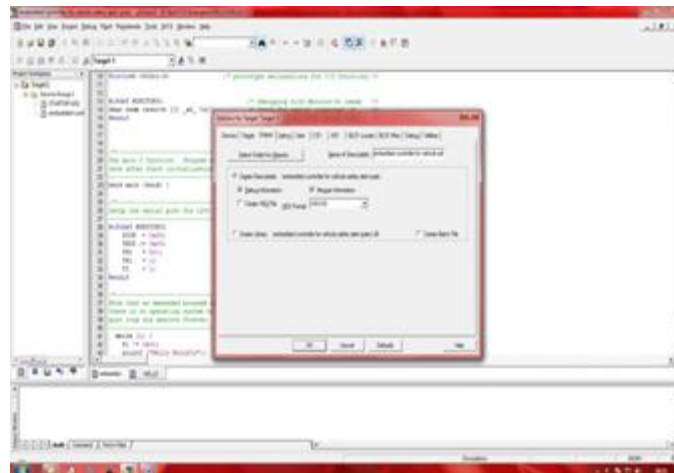
1. Select the microcontroller in which the hex file have to be downloaded. In this project AT89S52 is selected from the 8051 microcontroller family.



2. Add the source files i.e. the C files to the project.



3. Select the option create hex file to create the hex file while running the program.



4. Download hex files to the respective microcontroller.

#### IV. Implementation

An embedded controller is implemented by using two set of modules, i.e. the obstacle sensing module and the gas detection module. The obstacle detection module is mounted in front of the vehicle such that it searches for the static obstacles. The IR transmitter transmits the IR signals continuously to the maximum level of 3 ft. when the IR signals are reflected back by an object then the signal is received by the IR receiver, thus the module senses the obstacle. It then sends the corresponding signals to the microcontroller such that an alarm is provoked by the microcontroller as the first measure and stops the vehicle. The data flow diagram of the obstacle sensing module is shown in the fig.9.

In the same way the gas detection module which is placed inside the vehicle continuously senses the presence of gas in the vehicle cabin. When excess levels of toxic gases such as CO greater than 20ppm, LPG higher than 10,000ppm and the presence of alcohol are found, then the corresponding signals are sent to the microcontroller which in turn provokes an alarm to alert the passengers in the vehicle. And in turn commands the GSM module to send an SMS to the authorized user about the alarming situation inside the vehicle using the AT+CMGS command. The flow diagram of the gas sensing module is shown in the fig.11. The GSM module is connected to the microcontroller by the means of RS232 cable as the communication between the GSM and the microcontroller is synchronized through the MAX232. The system architecture of the proposed system is shown in the fig. 10.

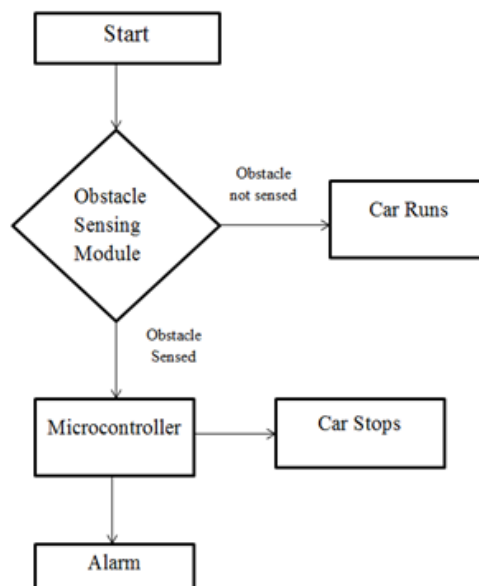


Fig.9 Flow Chart for Working of Obstacle Sensing Module



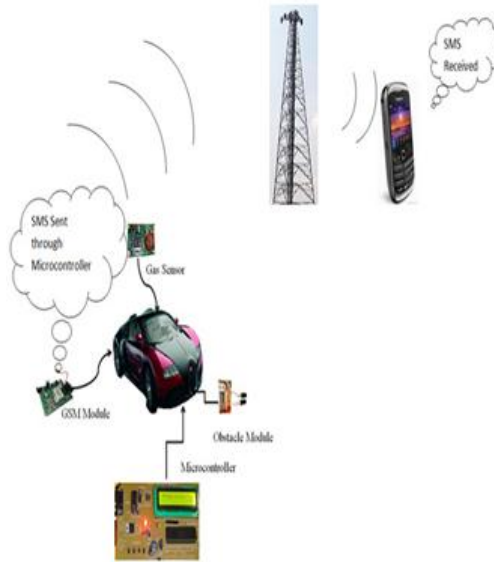
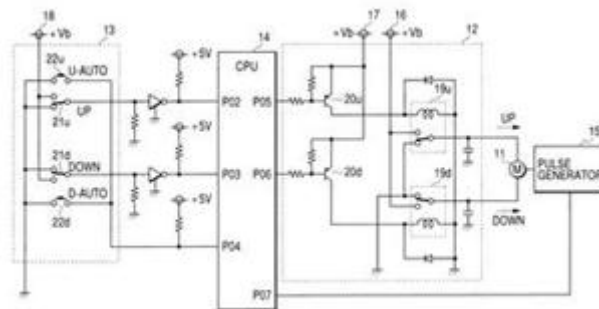


Fig.10 System Architecture of the Proposed System

**AUTOMATIC DOOR OPENING SYSTEM:**

In the present day the transportation is very complex process and the accident occurrence ratio in day today life is going on increasing .most of the recent accidents in the road ways are due to fuel leakage and fire explosions have been increasing. To avoid these accidents we are developing the new software and hardware combinations. The circuit diagram is given below.

**BLOCK DIAGRAM:**



**Door lock opener:**

**Description:**

- This circuit can be used to operate an electric strike or an electromagnetic lock on a door. It is not the door being opened/closed, but a small electromagnetic strike which unlocks the door. The opener has the following features currently programmed in software:

Automatic operation when a guest pushes the doorbell, the strike is operated immediately - can be set by a switch (SW)

- manual remote operation - by pushing a button (BUTTON), the strike is operated immediately
- timer delayed operation by pushing the same button longer than 500 ms - the strike is operated after a pre-set time
- programmable timer delays for relay operating time and delay before timer opening
- two-color LED display shows current state

**Usage:**

After power-on the circuit will be operating and the LED will be constant RED when the auto switch is in OFF state and will be slowly changing between red and green when the switch is in AUTO state. Shortly pushing the BUTTON will operate the door relay for about 3 seconds (this delay can be reprogrammed). Holding down the PUSH button for a bit longer will operate the door relay after about 20 seconds. This timer delay is also reprogrammable. The display LED lights in constant YELLOW while the delay timer is running.

**Delay programming:**

WARNING: entering inappropriate codes for the timers will result in strange operator delays and you may have to break the power for the device to reset and start again.

To enter the delay programming mode, hold down the BUTTON and change the state of the SWITCH. The LED display will turn OFF until the release of the BUTTON. In this mode the software reads the values for the two timers in the form of 4 hex numbers. The first two numbers make up an 8-bit value for the timer of the door opening time and the second two numbers make up an 8-bit value for the delay timer. The values are given in a rough 10 ms unit.

- entered digits: D1 D2 D3 D4
- door opening time:  $D1 * 16 + D2$
- opening delay time:  $D3 * 16 + D4$
- to program 3 seconds opening time and 20 seconds delay time you would enter digits 30c8 (3,0,12,8)
- To return to the hard-coded original delay times, break the power to the device for a few seconds.

Entering digits: press the BUTTON the number of times the digit you want to enter. E.g. you want to enter digit 8, press the button eight times. To finish entering a digit, wait for about 1 second and the LED color will blink, indicating that the code is stored and you can proceed entering the next digit (if there are more). Not pushing the button for 1 second in this mode will store the digit 0 and proceed with a LED blink. After all 4 digits are stored the program will restart with the new values.

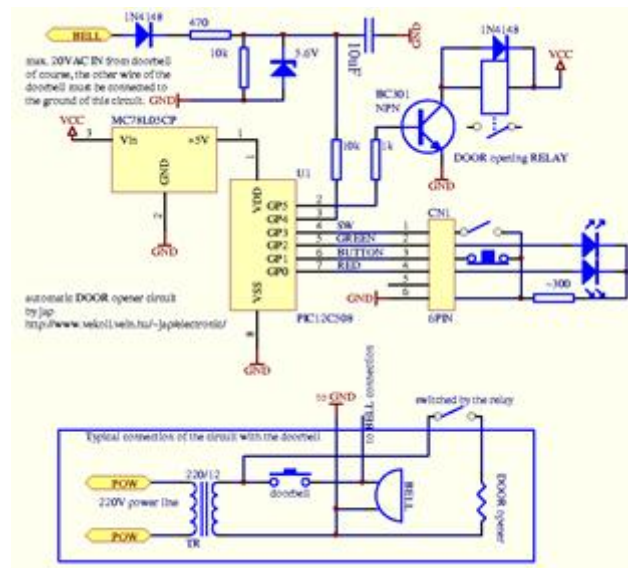
**Software:**

The software for the microcontroller is available. It is developed under the Microchip MPLAB development environment, and you will need it or some other PIC compiler to build the object code for the device. If you program the object code, check the fuse settings and make sure that all the I/O pins are programmed to be I/O lines and not OSC or MCLR. In the beginning of the source you find #define line which sets the PIC chip under which the code will compile. It is a good idea to develop on a 16F84 device and when everything is working, compile the code for the one-time programmable 12C508.

**Schematic diagram:**

Capacitors on the VCC and VDD lines are not shown, but recommended. Place 100 nF close to the PIC pins VCC & VSS, place 100 nF and 100 uF close to the 78L05 pins VCC & GND. The two LED diodes on the schematic are packaged together in a two-color green-red LED with 3 leads. The button is a break-on-push type for fast sensitive operation.

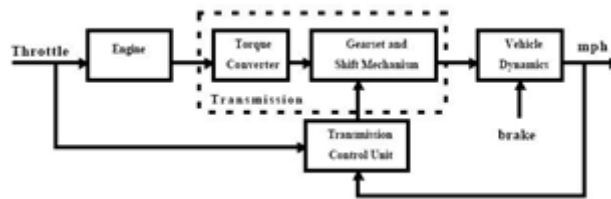
Fig (b): Door lock opener



**AUTOMATIC WINDOW OPENER:**

In order to escape from fire accidents if the door doesn't open then with the help of this circuit and technology we can prevent the loss of one's precious life from fire accidents. In the below circuit we use the transformers for the purpose input power supplying and this passes power to the circuit. The switch is used to perform the applications for the switching purpose the door opener applying the IN4148 diode is used to provide the LED purpose and acts as a SENSOR. by using the CN1 6 pin IC the LED the opening function is carried out. The equivalent circuit diagram is shown in the figure above.

**BLOCK DIAGRAM:**



The other circuit and block diagrams are shown in the figure below. There is a feedback loop between the ENGINE & TRANSMISSION CONTROL UNIT of the vehicle. The engine output is given to the TORQUE CONVERTER & to the GREASE&SHIFT MECHANISM and the resulting output is given to the VEHICLE DYNAMICS., where the BRAKE is applied. The whole operation is controlled by the TRANSMISSION CONTROL UNIT. The block diagram is shown above.

**SIMULINK DESIGN VERIFIER:**

Power Window Controller Temporal Properties Using a power window controller model, this demonstration will help you model temporal system requirements for property

Proving and test case generation using Simulink® Design Verifier™ Temporal Operator blocks. Contents

[Temporal Operators](#)

[Power window Controller](#)

[Property Specification](#)

[Test case Generation for Property Validation](#)

[Clean Up](#)

Temporal Operators

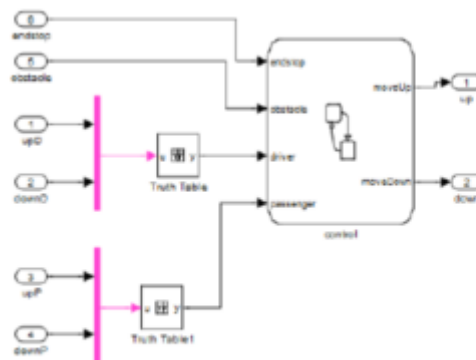
The Simulink® Design Verifier™ library provides three basic temporal operator blocks that can be used to model temporal properties. The intent of the temporal operators is to support the specification of temporal requirements, such that the modeled property has a closer co-relation to the actual English requirement. These blocks are low-level building blocks for constructing more complex temporal properties. Power window Controller. The power window controller responds to the driver and passenger commands by giving the commands for moving the window up or down. It also responds to an obstacle and to reaching the end of the window frame in either direction. Consider the following two requirements for the power window controller:

Requirement 1 (Obstacle Response)

Whenever an obstacle is detected, the controller shall give the down command for 1 second.

Requirement 2 (Auto down feature)

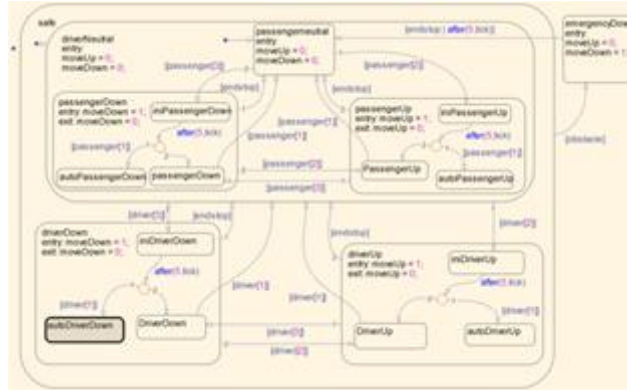
If the driver presses the down button for less than 1 second, the controller keeps giving the down command until the end has been reached or the driver presses the up button.



**%Model of the power window controller:**

Open\_system ('sldvdemo\_powerwindowController')

Open\_system ('sldvdemo\_powerwindowController/control')

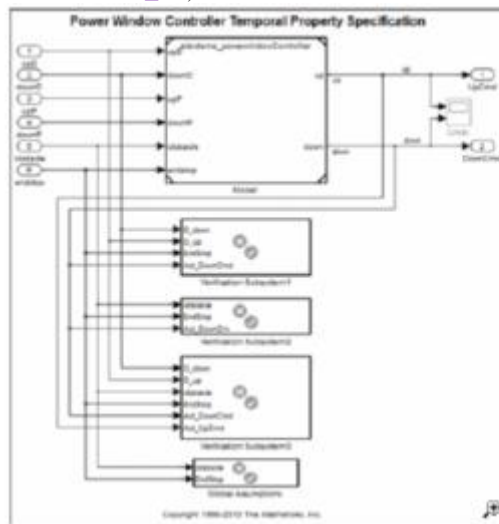


Property Specification

The power window verification system is the top-level model that contains a model reference to the power window controller model specifying the controller behavior and the modeled requirements.

**%Model of the top-level verification system:**

```
Open_system('sldvdemo_powerwindow_vs')
```



Global Assumptions: The power window controller is an open system. This makes the environment controlled inputs, obstacle and end stop (end of the window frame) to occur freely. To constrain the environment, add two global

Assumptions for your controller model.

- 1) The obstacle and the end stop inputs never become true at the same time.
- 2) The obstacle does not occur multiple times within the following 1-second interval.

For the temporal assumption on obstacle, use a Detector block with output type of Delayed Fixed Duration to Capture the fixed duration of 1 second (5 time steps with 0.2 sample times).

**% Global Assumptions:**

```
Open_system('sldvdemo_powerwindow_vs/Global Assumptions')
```



Now consider the first controller requirement for Obstacle Response.

**% Obstacle Response:**

```
open_system('sldvdemo_powerwindow_vs/Verification Subsystem2')
```

Here, use the Detector block with output type of Delayed Fixed Duration for the property specification. After detection of the obstacle, construct a fixed interval of 4 steps. Note that the input is not observed during the output construction phase for the Detector with Delayed Fixed Duration output type. In the case where the

obstacle can occur freely in absence of the assumption, you might wish to observe all the intermediate occurrences of the obstacle. This can be achieved through an Extender block with Finite extension duration of 4 time steps. Now consider the Auto Down feature of the power window controller. For illustration, consider this property specification in smaller parts:

1. The first temporal duration of interest driver presses the down button for less than 1 second is captured by Detector1. At sample rate of 0.2, the 1-second interval is broken down into 5 time steps. On detection of the down signal, construct a 5-step fixed temporal duration at the output of Detector1, which you will subsequently use, in combination of other constraints.

2. For the Auto Down feature, you know that the down signal cannot be pressed for more than 1 second, or 5 time steps. Thus, you want to ensure that the either both driver up and down are true or both are false in less than 5 steps after down is pressed. By having an AND block connect this driver neutral to the other two Detector outputs, enforce the constraint that the driver down can be pressed for any number of consecutive time steps less than 5.

3. You also need to ensure that during this period other signal, such as Obstacle, End Stop, or Driver Up are not true, since these will take the controller out of responding to the down press. This is captured in Detector2 enforcing that NOT (Halt Down) is true for 5 time steps. This is achieved by using a Detector block with Delayed Fixed

Duration, with the Time steps for input detection = 5 and Time steps for output duration = 1. Output.

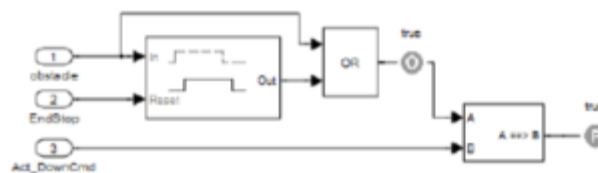
4. Take the AND of these constructed durations.

5. For the AutoDown feature, you do not want to limit the number of time steps that the controller gives the down

Command. You know that you want the controller to keep giving the down command as long as the driver does not

Again press an up or down command, or you hit an obstacle or the physical end of the window frame. This behavior can be captured by the Extender block with Infinite extension period and an external reset signal that encodes the condition to end the extension.

6. The final piece is an implies block that takes the temporal duration constructed as explained above and checks if the controller down command is true for every time step of this duration. Once you have the initial property specification, you can use it for property proving using Simulink Design Verifier. You immediately get a counterexample for the above property. The counterexample shows a scenario where the down command is given when the controller was in the emergency down state due to response to an earlier detected obstacle. When you add a constraint to avoid this, you get another counterexample. If the down button is pressed when previously the up command was being given, the auto down feature is disabled and the down command is given only as long as the down button is pressed. Looking at these counterexamples and observing the model, you see a pattern that the auto down feature is only enabled when the controller is in a neutral state to begin with when the driver presses the down button. Incorporate this constraint by forcing the controller output to be neutral - neither up nor down command is true - as a precondition for the auto down property. This property proves valid.



**% Valid Auto down:**

```
open_system('sldvdemo_powerwindow_vs/Verification Subsystem3')
```

Test case Generation for Property Validation

Once the properties are specified, in addition to property proving, you can run Simulink Design Verifier to automatically generate test cases that exercise various conditions in the property. This can be achieved by placing custom test objective blocks at appropriate locations in the property. One such place is to put a test objective block (true value) on the signal feeding into the first input of the implies block (as shown in the above property). On running test generation, when the test objective is satisfied, you can get a test case exercising the various constraints encoded in the property. Consider the valid Auto down property specified above. On running test generation, Simulink Design Verifier satisfies this test objective and creates a test harness. The Signal Builder block with relevant signals is shown below. One can now simulate this test case and see how the temporal durations are created in the property by placing a scope that displays the input and output values of the two Detector blocks, and No\_Cmd.

Manually inspecting the test case values enables you to see if the property specified behaves as intended.

There is an additional advantage to having such a test objective on the implies block. During property proving, you can Ensure that the implies is not getting trivially true making the property valid; in other words, the output

of the implies block becoming true because the first input is always false. When you get a test case satisfying this objective, you know that there is at least one case where the first input to implies becomes true. This exercise can help you validate your property specifications by manually inspecting the test cases automatically generated by Simulink Design Verifier.

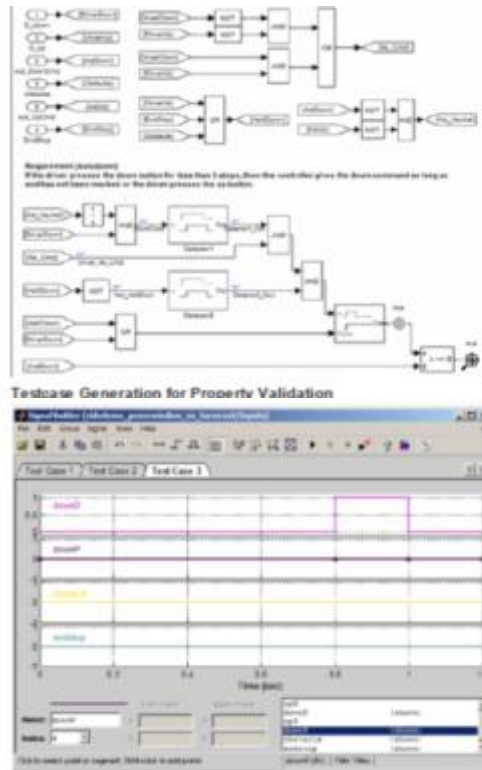
Clean Up

To complete the demonstration, close all the opened models.

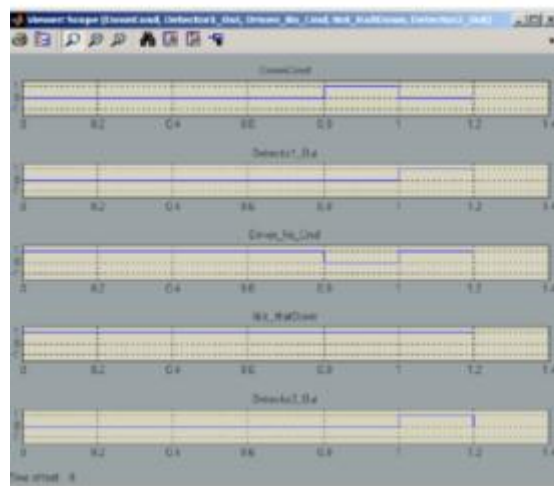
```
Close_system ('sldvdemo_TOBlocks', 0);
```

```
close_system ('sldvdemo_powerwindowController', 0);
```

```
close_system ('sldvdemo_powerwindow_vs', 0);
```



One can now simulate this test case and see how the Temporal durations are created in the property by placing a scope that displays the input and output values of the two Detector blocks, and No\_Cmd.



Manually inspecting the test case values enables you to see if the property specified behaves as intended.



There is an additional advantage to having such a test objective on the implies block. During property proving, you can ensure that the implies is not getting trivially true making the property valid; in other words, the output of the implies block becoming true because the first input is always false. When you get a test case satisfying this objective, you know that there is at least one case where the first input to implies becomes true. This exercise can help you validate your property specifications by manually inspecting the test cases automatically generated by Simulink Design Verifier.

Clean Up

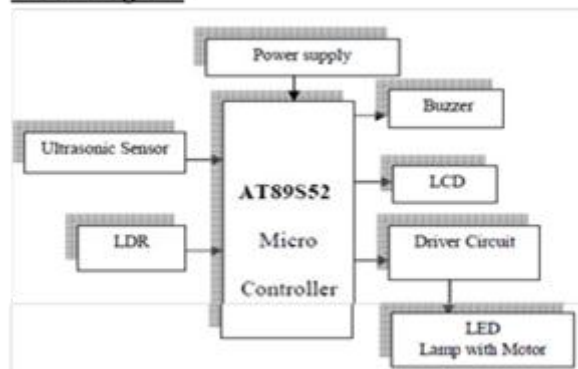
To complete the demonstration, close all the opened models.

```
close_system('sldvdemo_TOBlocks', 0);
close_system('sldvdemo_powerwindowController', 0);
close_system('sldvdemo_powerwindow_vs', 0);
```

**VEHICLE REVERSING CONTROL:**

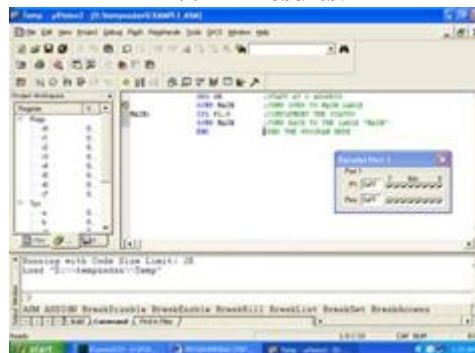
In this circuit we are using the LDR for time sensing purpose, whether the time is day or night. So the LDR will sense the time, if it is night when the car is get reversed the ultrasonic sensor will sense the obstacles which are the behind the vehicle and it will rotate the LED lamp with motor according to the distance. If it is low distance it will rotate the lamp with that much angle it will flash the light on humans and alert the pedestrian's. So the pedestrian come to know that the vehicle is coming behind so he can get alert. so it can alert the driver by using the buzzer and the obstacle information also displayed on LCD display. If the obstacle is very nearer to the car then Break will be applied, so that the speed of the car is decreases and automatic break also performed

**Block diagram :**



**Block diagram of Vehicle reversing**

**V. Results:**



Output obtained while executing the program developed by using keil software

**Advantages**

⌚ **Automatic vehicle control**

The vehicle is automatically stops when it detects the obstacle are very nearer.

⌚ **Fast response**

The sensors sense the obstacle very fast.

⌚ **High reliable**

⌚ **System will be stable for a long time**

The sensors senses the obstacle continuously and gives the distance between the vehicle and obstacle



**Applications**



**Future Scope**

In future the extension of our project is an autonomous car which is in research.

An **autonomous car**, also known as **robotic** or informally as **driverless**, is an autonomous vehicle capable of fulfilling the human transportation capabilities of a traditional car. As an autonomous vehicle, it is capable of sensing its environment and navigating on its own. A human may choose a destination, but is not required to perform any mechanical operation of the vehicle.

Autonomous cars are not in widespread use, but their introduction could produce several direct advantages:

- Fewer crashes, due to the autonomous system's increased reliability compared to human drivers
- Increased roadway capacity due to reduced need of safety gaps and the ability to better manage traffic flow.
- Relief of vehicle occupants from driving and navigation chores.
- Removal of constraints on occupant's state - it would not matter if the occupants were too young, too old or if their frame of mind were not suitable to drive a traditional car. Furthermore, disabilities would no longer matter.
- Elimination of redundant passengers - humans are not required to take the car anywhere, as the robotic car can drive empty to wherever it is required.
- Alleviation of parking scarcity as cars could drop off passengers, park far away where space is not scarce, and return as needed to pick up passengers.

Indirect advantages are anticipated as well. Adoption of robotic cars could reduce the number of vehicles worldwide, reduce the amount of space required for vehicle parking, and reduce the need for traffic police and vehicle insurance.

Autonomous vehicles sense the world with such techniques as laser, radar, lidar, GPS and computer vision. Advanced control systems interpret the information to identify appropriate navigation paths, as well as obstacles and relevant signage. Autonomous vehicles typically update their maps based on sensory input, such that they can navigate through uncharted environments.

**PRINCIPLE OF THE DEVICE**

The automatic alarming device is comprised of a controller, acceleration sensors and angle sensors module, a GPS module (Global Positioning System), and a GSM module (the Global System for Mobile Communications). The design scheme of

The automatic alarming device is shown in Fig. 1.

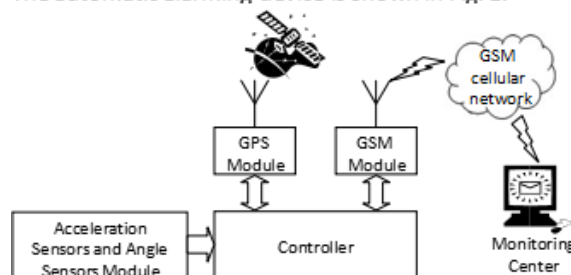


Figure 1. Design scheme of the automatic alarming device.

The controller obtains acceleration and angle data from the acceleration sensors and angle sensors module. The detection algorithm built in the controller detects whether an accident occurs. If the algorithm detects an accident, the controller sends a command to the GPS module, to obtain the location of the current vehicle, and then sends a command to the GSM module. The GSM module sends a short message of alarm to monitoring center through GSM cellular network to complete the alarm function. The contents of an alarming short message include the latitude and longitude of the location of the current vehicle, time, date, crash type, license plate, name of driver, and mobile phone numbers of driver.

When the automatic alarming device detects an accident, the buzzer of the device sounds an alarm to remind the driver. If the driver does not manually cancel the alarm after 60seconds, the short message of the alarm will be sent to monitoring center. If the driver cancels the alarm, the automatic alarming device starts to detect whether an accident occurs again. Even if the driver loses consciousness in an accident, the driver can also be rescued in time, as shown in Fig. 2.

The scheme research is supported by the National 863 Projects of China (2009.01-2012.08). We express our acknowledgment hereby.

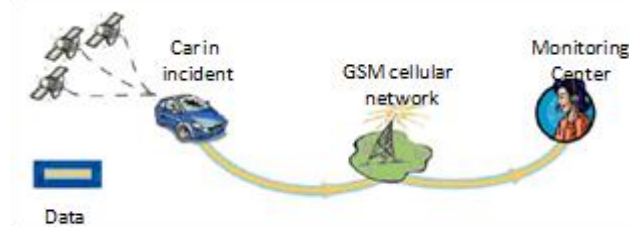


Figure 2. Process of alarming.

**DETECTION ALGORITHM**

Body injury usually results from impact acceleration or overload effects on human body in an accident. It causes pain, transient loss of consciousness and a variety of mechanical damage, even death. The damage degree relates to peak of overload, action time, rate of overload, direction of overload, position of human body, and other factors [5].

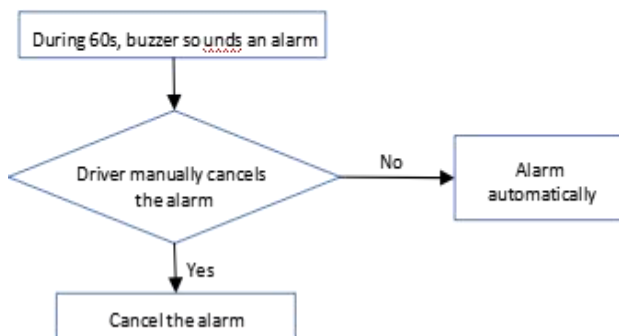


Figure 4. Detection algorithm of flowchart.

Because each component of vehicle has different acceleration on impact, the sensors of the automatic alarming device are placed at the center of gravity of vehicle in order to evaluate the acceleration of driver in an accident. It is different

from most of emergency call systems.

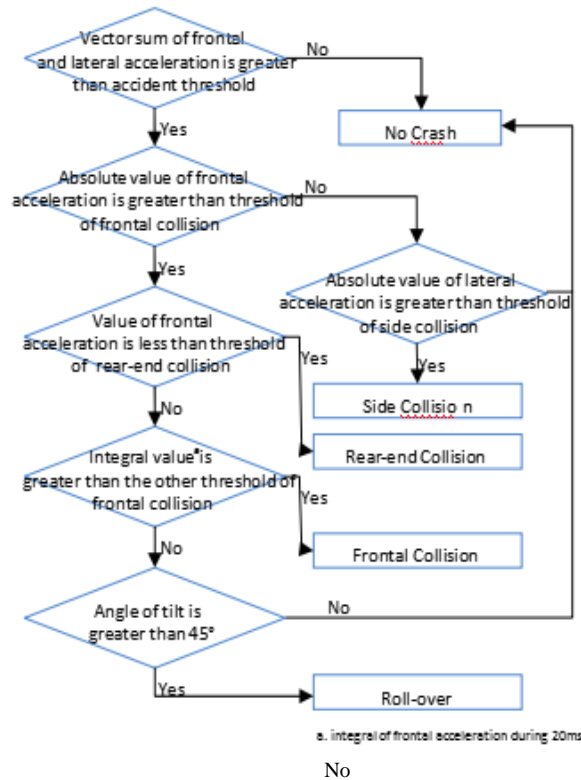


Figure 3. Detection algorithm of flowchart.

The detection algorithm is used to accurately detect an accident and judge crash types as shown in Fig. 3 and Fig.4. Crash types include frontal collision, side collision, rear-end collision and roll-over. The detection algorithm is suitable for different types of vehicles. The automatic detection of an accident is achieved by permanent listening to the crash sensor. It keeps transmitting data to controller. The controller keeps scanning the data. If it is found that there exists a design parameter value exceeds a given threshold value, the buzzer of the device sounds an alarm to remind the driver. After 60 seconds, if the driver does not manually cancel the alarm, the device automatically alarms.

### EXPERIMENT RESULTS

In order to evaluate the automatic alarming device performance, we experiment on two aspects: testing whether the device may trigger false alarm when the vehicle is during normal driving, and sled test of vehicle frontal impact simulation.

#### A Testing whether the Device may Trigger False Alarm

The automatic alarming device is mounted on a vehicle, to test whether the device may trigger false alarm, when the vehicle is during normal driving. Crash sensors mounted on the roof can get higher acceleration and angle of tilt than placed in the center of gravity of the vehicle, and it is easier to detect whether the device may trigger the false alarm, as shown in Fig.5.



Figure 5. Test whether the device may trigger false alarm.

The vehicle is respectively during normal driving, passing through speed control humps and braking at the speed of 40, 50, 60, 70km/h, as shown in Fig. 6. We test three times at each speed. If the automatic alarming

device produces a false alarm, the short message with alarming will be sent to the mobile phone of testing personnel. The testing personnel can judge whether the device triggers a false alarm.



Figure 6. Testing ground.

The analysis of experimental data shows that the automatic alarming device does not cause false alarms at the speed of 40 to 70 km/h range. The automatic alarming device has achieved the design requirements.

**B. Sled Test of Vehicle Frontal Impact Simulation**

We test the automatic alarming device by sled test of vehicle frontal impact simulation. We want to know whether the automatic alarming device generates a condition that the automatic alarming device has no alarm, but should alarm. We contrast the acceleration data from the automatic alarming device with the acceleration data from sled test system to verify the alarm effect of the automatic alarming device. The sensors of the automatic alarming device are placed as shown in Fig. 7.



Figure 7. Sled test of vehicle frontal impact simulation.



Figure 8. Alarm short message.

We made a sled test of vehicle frontal impact simulation at the speed of 50km/h. The mobile phone of testing personnel successfully received the alarm short message, and the judgment of crash type was accurate as shown in Fig. 8.

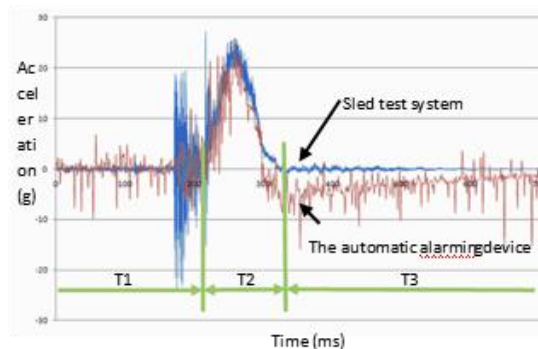


Figure 9. Contrast the acceleration data.

We contrasted the acceleration data from the automatic alarming device with the acceleration data from sled test system as shown in Fig. 9. The acceleration data are not filtered. The blue curve represents the change of acceleration from sled test system; the red curve represents the change of acceleration from the automatic alarming device. The changes of accelerations are similar. Both of curves are divided into 3 stages. The difference in T1 and T3 between these two curves is mainly due to the different choices of sampling frequency and sensors. T2 is the main part of the impact. Both of these peaks in T2 are basically similar about 25g. Both of the durations in T2 are basically similar about 100ms.

## VI. Conclusion:-

With the help of this above technical presentation by equipping these equipments in our vehicle provides a support and it can be modeled as a transporting system that avoids and prevents accidents, if at all accident occurs, your condition will be messaged up to the service centre and ambulance will come to the that place without manual interruption easily. Accidents due to any gas leakages will also be avoided easily. Also fire accidents prevention measures like automatic windowing and door opening mechanisms is also included in this paper.

## References:

- [1] Hitoshi Miyata, Makoto Ohki, Yasuyuki Yokouchi, Masaaki Ohkita, "Control of the autonomous mobile robot DREAM-1 for a parallel parking," Department of Electrical and Electronic Engineering, Faculty of Engineering, Tottori University, 4-101, Koyama-Minami, Tottori 680, Japan, *Mathematics and Computers in Simulation* 41 (1996) 129-138
- [2] Nikolaj Zimic, Miha Mraz, "Decomposition of a complex fuzzy controller for the truck-and-trailer reverse parking problem," University of Ljubljana, Faculty of Computer and Information Science, Trzaskacesta 25, SI-1000 Ljubljana, Slovenia, *Mathematical and Computer Modeling* 43 (2006) 632-645
- [3] Massaki Wada, Student Member, "Development of Advanced Parking Assistance System," IEEE, Kang Sup Yoon, Member, IEEE, and Hideki Hashimoto, Member, IEEE, *IEEE Transactions on Industrial Electronics*, VOL. 50, NO. 1, Feb 2003
- [4] Tsung-hua Hsu, Jing-Fu Liu, Pen-Ning Yu, Wang-Shuan Lee and Jia-Sing Hsu, "Development of an Automatic Parking System for Vehicle," Automotive Research and Testing Center, Changhua County, Taiwan, R.O.C., IEEE Vehicle Power and Propulsion Conference (VPPC), September 3-5, 2008, Harbin, China
- [5] F. Gomez-Bravo, F. Cuesta, A. Ollero, "Parallel and diagonal parking in nonholonomic autonomous vehicles," Departamento Ingeniera de Sistemas y Automatic, Escuela Superior de Ingenieros, Universidad de Sevilla, Camino de los Descubrimientos, E-41092 Seville, Spain, *Engineering Applications of Artificial Intelligence* 14 (2001) 419-434
- [6] Pan Chaofeng, Zhou Kongkang, Li Zhongxing, "Developmental review of vehicle emergency call technology," *China Safety Science Journal*, vol.18, pp. 17-21, 2008.
- [7] Rosmos D. Lindholm, "A pan European automatic emergency call (ecall)," Alborg University, dissertation, December 2004.
- [8] Howard R. Champion, Augenstein JS, Blatt AJ, "Reducing highway deaths and disabilities with automatic wireless transmission of serious injury probability ratings from vehicles in crashes to EMS," *Proceedings of the 18th International Technical Conference on the Enhanced Safety of Vehicles*, Paper Number 406, June 2002.
- [9] Pan Chaofeng, "Study on vehicle emergency call technology and vehicle crash severity assessment," Jiangsu University, 2008.
- [10] R. Al-Ali, Member, IEEE, Imran Zualkernan, and Fadi Aloul, Senior Member, IEEE, "A mobile gprs-sensors array for Air pollution monitoring" vol. 6, pp. 410-422, Oct. 2010.
- [11] Da-Jeng Yao, "A gas sensing system for indoor air quality control and polluted environmental monitoring," pp. 11-14, 2009.
- [12] Chen Peijiang, Jiang Xuehua, "Design and Implementation of Remote monitoring system based on GSM," vol.42, pp.167-175. 2008.
- [13] LIU Zhen-ya, WANG Zhen-dong, CHEN Rong, "Intelligent Residential Security Alarm and Remote Control System Based on Single Chip Computer," vol. 42, pp. 143-166, 2008.
- [14] M.Gao, F.Zhang, and J.Tian, "Environmental monitoring system with wireless mesh network based on embedded system," in *Proc. 5th IEEE Int. Symp. Embedded Computing*, 2008, pp. 174-179.