Fault Tolerant Automotive Safety Engineering using MEMS and System on Chip

CHANDRASEKARAN SUBRAMANIAM[1], SRIRAM BADRI[2], NARENDRA VARUN K[3], RAJESH KUMAR S[4], SOORYA NIVEDHA A[5]
[1]Computer Science and Engineering, [3,4]Mechanical Engineering,
[2,5]Electronics and Communication Engineering
[1,3,4]Kumaraguru College of Technology, [2]Sri Venkateswara College of Engineering,
[5]K.P.R Institute of Engineering Technology
Anna University, Chennai, INDIA
[1]chandrasekaran_s@msn.com, [2]b-sriram@hotmail.com, [3]narendravarun@hotmail.com,
[4]rajeshsugumaran@hotmail.com, [5]sooryagupta12@gmail.com

Abstract: - The objective of the paper is to propose a Fault Tolerant Automotive Safety Engineering focusing on the context awareness features, user actions and unexpected reaction from the environment with the help of MEMS sensors and micro controller. The safety aspect in the design and development of an automotive software is considered in the system level and also in the detailed hardware and software component level. The model is checked using MuSMV model checker against the design faults. The safety due to multiple functionalities and behaviour of the critical modules like tyre pressure monitoring, wiper control, battery power level detection, seat belt and headlight monitoring are considered in the work and implemented through interacting automata towards system and software safety standards. A prototype of the proposed safety automotive model is implemented with all sensors and a programmable system on chip micro controller to validate the safety quality through timing and sensing fault tolerant characteristics.

Key-Words: - Fault tolerance, architectural model, safety design, interacting automata, safety standards, programmable controllers,

1 Introduction

Design of fault tolerant electronics has become a standard requirement in the automotive sector these days. These systems increases the overall automotive and passenger safety by liberating the driver from handling routine tasks and also assisting the driver during critical situations. Fault tolerance techniques used in the design of automotive software and how they help in improving the overall reliability and dependability of the system is investigated. Common design techniques used in the design of fail-safe sensors and actuators is presented. Advancements in the field of automotive electronics have helped in realizing the potential of sophisticated vehicular control systems. In addition to liberating the driver from routine tasks, such systems assist the driver during critical situations, thereby enhancing vehicular safety and performance. Ensuring fault tolerance in automotive software is an active area of research. Safety-critical systems such as X-by-wire systems and most ECUs typically use a lot of sensors for performing their functions. Hence sensors and actuators, which form the backbone of most commonly used electronic systems, need to be fault tolerant as well. Fault tolerant communication systems are built so they are tolerant to defective circuits, line failures etc., and constructed using redundant hard- and software architectures to ensure reliable communication between different sub-systems in a vehicle. Safety-critical automotive applications, such as steer-by-wire systems, are in most cases control systems with hard real time requirements. Such systems typically have a number of sensors (inputs) connected to them, whose values are processed in order to produce the control actions. Consequently the sensors are the first in the flow of information and control computations rely of these values. Therefore it is important that the sensors can be trusted. To avoid costly and numerous wires, ECUs have to be placed close to the sensors, and communication between ECUs has to be multiplexed. A fault-tolerant sensor configuration should be at least fail-operational for one sensor fault. This can be obtained by hardware redundancy with the same type of sensors or by analytical redundancy with different sensors and process models. Sensor systems with static redundancy are realized with a triplex system and a voter. A configuration with dynamic redundancy needs at least two sensors and

ISBN: 978-1-61804-146-3

fault detection for each sensor. The fault detection can be performed by self-tests [1]. Single-chip solutions cannot be divided up into fault containment regions, i.e. they are subject to common mode failures. Common mode failures occur because of faults whose occurrence causes the failure of all replicas. For instance, duplication cannot provide any protection against a common mode failure that stops both CPUs, e.g. a failure on the power supply. In a single-chip implementation, usual sources of common mode failures are the clock tree, the power supply and the silicon substrate [2]. The increasing use of software is closely related to the increasing occurrence of system accidents. Software usually controls the interactions among components and allows almost unlimited complexity in component interactions and coupling compared to the physical constraints imposed by the mechanical linkages replaced by computers. The constraints on complexity imposed by nature in physical systems do not exist for software and must be imposed by humans on their design and development process [3]. The safety in the design of automotive software indicates that errors which could reduce the safety of the system have been eliminated or controlled. The safe design with minimum errors can be achieved only through systematic design and validation not only in the architectural perspective but also in the timely action perspective [4]. To see if a system meets real-time requirements, schedulability analysis is used and this methodology is well known for single-processor or multiprocessor operating systems [5]. Generally, model checkers are formal verification tools that evaluate a model to determine if it satisfies a given set of properties. Modern symbolic model checkers use logical representations of sets of states, such as BDDs (Binary Decision Diagrams), to represent regions of the state space, which satisfy the properties being evaluated [6]. The use of multiple functional safety analysis techniques can work together to make each more efficient than if applied independently [7]. Drive-by-wire systems can be defined as electronic or electrical systems or subsystems, which have direct control of the vehicle. This can be implemented to control a particular function, e.g. braking, or can be a global system strategy as in full-vehicle drive-by-wire systems [8]. Mixed discrete-continuous systems exhibit different modes in which they operate continuously, and modes are switched in a non-continuous-i.e., discrete-manner [GKS00]. Approaches to specifying hybrid systems include hybrid automata, hybrid Petri nets, and equation-based approaches. In practice, extensions of the Matlab Simulink languages (State flow) are most commonly used [9]. The sensors have triple redundancy with a voting mechanism in place. There are also redundant copies of some of the signals that are essential to ensure safety, such as displacement and force measurements of the brake pedal. Other redundant subsystems include power supply and ECUs[10]. The organization of the paper is as follows: Section 2 in this paper introduces a architectural model that is implemented in a safety critical automotive application. The various sensors that are used in the safety model are described in Section 3. Model verification to ensure that the model is present in all its states is done using a symbolic model verifier, NuSMV. The model that is simulated before the actual hardware implementation to verify whether the inputs given are correct and if the outputs obtained are as per estimation. Section 4 explores the simulation that is done using MATLAB for the design verification of various sub processes and the sample output is given. Section 5 discusses the hardware implementation of a prototype using the sensors and programmable system on chip micro controller and concludes with the experimental results obtained and the scope for future works.

2 Proposed Safety Architecture

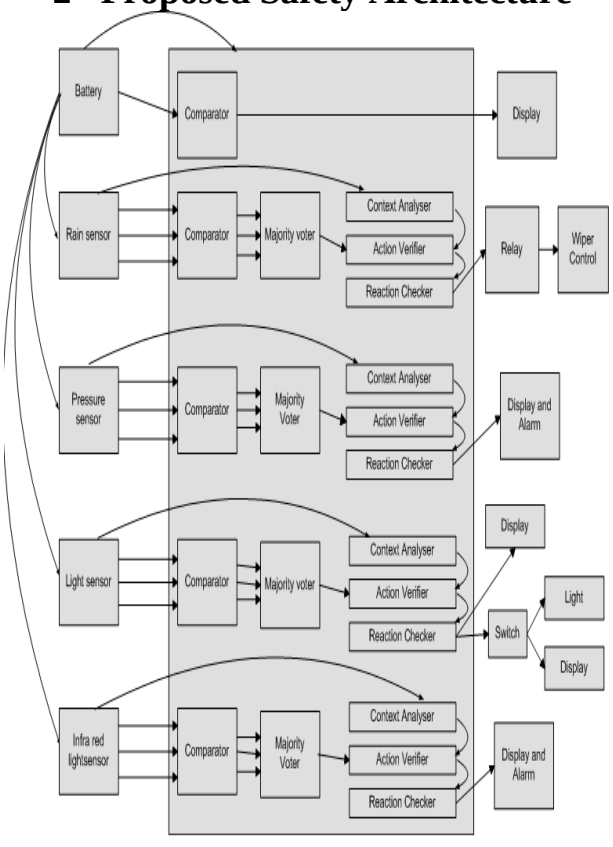

Fig.1 Safety architecture with MEMS-IO controllers

The existing architectures of safety critical modules in automobiles take in to consideration the necessary sensor input values and the micro controller used calculates the necessary reaction according to the input. The context in which the values are obtained and the correctness of the value forms an integral part in a safety critical system. Taking surrounding environment as an example, we can say that the sensor value varies in varied weather conditions as every sensor has an operating temperature. As the surrounding temperature varies, the values from the sensors vary. Therefore the context in which the sensor is worked needs to be taken into account while implementing a safety critical sensor network module. Moreover when the input from the sensor is directly fed into the micro controller, the correctness of the value obtained cannot be determined. Sometimes, the sensor may give incorrect values due to errors present in the hardware or the surrounding disturbances. If these values are used by the controller to determine the reaction, it may lead to wrong decision which can results in accidents. The proposed architecture not only takes into account the action and reaction component involved in sensor networks, but also proposes an additional context analyzer to assess the environment in which the senor is implemented as shown in Figure 1. The model consists of five Micro electro-mechanical systems (MEMS) sensors to read values from five safety critical automotive components. The sensors continuously monitor the various parameters and update the micro controller with the acquired information. The values are then given to a comparator to compare with the estimated values. For the safety engineering module to be fault tolerant, values from the safety critical module is obtained thrice. On comparing the three values to the estimated values, the three signals are given to a majority voter. The values that are passed through a majority voter component ensure that the values are fault tolerant. The output of the majority voter is given to a context analyzer and action verifier. The context analyzer and action verifier are a part of the micro controller. The context analyzer determines the environment in which the sensor values are obtained. The action verifier ensures that the steps decided to be taken by the micro controller is correct. The combination of context analyzer and action verifier makes the system to be fail safe and fault tolerant. The reaction checker verifies whether the reaction to be taken on the safety critical module is correct. A warning signal is displayed to ensure timely rectification of the errors in the various safety modules.

3 Micro Electro-mechanical Systems

A micro electromechanical system (MEMS) is a technology which consists of many small electrical and mechanical devices integrated into one system. MEMS technology uses semiconductor fabrication techniques and here classical physics theory may not be applicable. The various steps involved in fabrication of MEMS devices are deposition of semiconductor layers, photo lithography and etching. MEMS find useful application in a variety of fields. Some of the important applications of MEMS are in sensor networks, actuators and accelerometers. The commercial use of MEMS can be found in ink-jet printers, accelerometers in cars, a number of business phones and digital cameras and various sensors. MEMS as sensors find applications in numerous safety critical application especially in automobiles. Sensors may be used to monitor safety critical modules in automobiles such as tire pressure monitoring, headlight monitoring, seat belt monitoring, wiper control and battery monitoring systems. The pressure sensor consists of a flexible diaphragm that deforms in the presence of a pressure difference. Piezo resistors are etched across the edges where the diaphragm is micro machined. Top side of the diaphragm is exposed to the environment. On application of pressure, the diaphragm is deformed downwards that changes the resistance of piezo resistors. On- chip electronics measure the change in resistance, which causes a corresponding voltage change to appear across the output pin. The MEMS sensor senses the ambient light to automatically control the head lights. It is a photo resistor made up of a high resistance semiconductor. The resistance depends on the intensity of the incident light. As the intensity of light increases, the resistance decreases. When the frequency of incident light is high, the photons give enough energy to the electrons to jump from the valence bond to the conduction band. These free electrons conduct electricity thereby lowering the resistance. The output pin of the sensors is a voltage value that depends on the resistance value in the sensor. The infra red light sensor consists of an infra red light receiver. The receiver is a voltage divider circuit which provides a voltage at the anode. The voltage across the cathode increases as the intensity of IR light falling on the receiver increases. An op-amp is used to amplify the detected voltage. This sensor may be used in seat belt monitoring as well as wiper control system. The intensity of light falling on the receiver is used to detect the presence of rain or whether the seat belt is locked or not.

ISBN: 978-1-61804-146-3

4 Model Verification using Symbolic Model Verifier- NuSMV

In safety critical model, the model needs to be verified for its correctness in all its sates. NuSMV is a symbolic model checker that allows checking finite state machines by using specifications in Computational Temporal Logic (CTL) and Linear Temporal Logic (LTL). The NuSMV input language allows specifying synchronous or asynchronous systems at gate or behavioural level. The basic idea behind NuSMV is to provide the relations between the various transitions that occur in a finite Kripke structure. This provides great flexibility at the same time it can also lead to inconsistency in the results obtained. Since NuSMV is used to define finite state machines, the data types that can be used are finite ones i.e. Boolean, scalar, bit vector and arrays of basic data types. The NuSMV code written below describes the state transitions that occur in the proposed model. The various systems present in the safety critical application are each assigned to the variable type *state. State* itself is defined as a module which consists of three variables of Boolean type. The three variables represent each of the readings obtained from the sensor. Module *Condition* defines the condition that the three input values need to satisfy for each system to be in *Safe* state. *Flag* mentioned in module *Condition* represents the result obtained by verifying the inputs from the sensors. The results so obtained are stored in separate variables which may be used to check the correctness of the input.

NuSMV Code

```
MODULE main
VAR
TPMS: state;
HLMS: state;
SBMS: state;
BMS:  state;
RMS:  state;
R_Pressure: condition(TPMS);
R_Head:    condition(HLMS);
R_Seat:    condition(SBMS);
R_Battery:  condition(BMS);
R_Rain:       condition(RMS);

MODULE state
VAR
Reading1: boolean;
Reading2: boolean;
Reading3: boolean;

MODULE condition( c )
DEFINE
flag  :=  ((c.Reading1&c.Reading2)  |
(c.Reading2&c.Reading3)  |
(c.Reading3&c.Reading1));
```

5 Simulation using MATLAB

The automotive system has several sub systems, like tire pressure monitoring, battery monitoring, head light control and wiper control sub systems. Each sub system has to be modeled and designed to protect the safety feature not in their functionalities but also in their timely behavior. MatLab block diagram language is the versatile platform to realize the safety aspects by verifying the system invariants and transitions during the respective inside and outside events.

5.1 Tire Pressure Monitoring Sub System

The tire pressure monitoring system continuously observes the pressure in the tire with reference value which is 30 psi. The monitored value is displayed in a LCD. In cases when the pressure is less or falls below a safe value i.e. 20 psi an alarm circuit is activated. Mathematical calculations are carried out with the voltage obtained from the sensor to convert the value into the actual pressure present in the tire before being displayed as shown in Figure 2. The circuit so designed is verified by feeding in different input pressure values and verifying with the expected output.

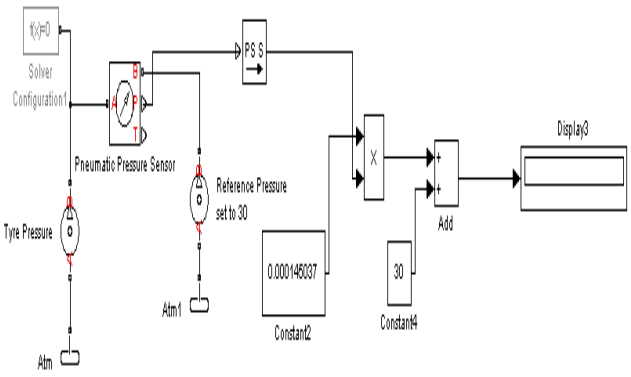

Fig.2 Tire Pressure Monitoring System

5.2 Battery Monitoring Sub System

Battery Monitoring System continuously monitors the battery voltage with the reference value. The battery voltage may vary between 12.7V at full charge and 10.8V when fully discharged. In instances where the voltage falls below 10.8V, a warning text is displayed as depicted in Figure 3.

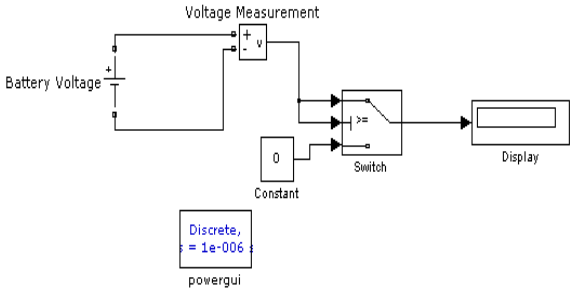

Fig.3 Battery Monitoring System

5.3 Head Light Monitoring Sub System

The output from the ambient light sensor is compared with the reference value. In case the sensor output, as shown in Figure 4 is more than the reference value the, the output of the comparator activates a relay. The intensity of light for the head light connected to the relay is thus varied according to the ambient light sensed. In case the sensor output is less than the reference value the headlight is switched off.

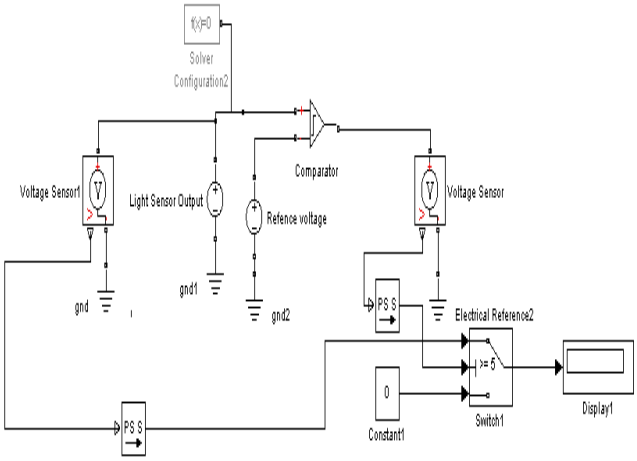

Fig.4 Head Light Monitoring System

5.4 Automatic Wiper Control Sub System

Wiper Control System consists of an IR transmitter placed inside the wind screen and an IR receiver on the outside. The intensity of the light received is compared with the reference value using a comparator. Any change in the intensity received indicates rain drops on the wind screen. The comparator output is given to a driver which in turn drives a stepper motor. The direction signal gives the direction in which the motor should rotate. The direction signal is adjusted depending on the size of the wind screen. The motor is powered using a battery as shown in Figure 5. Figure 6 shows a simulation of this circuit. The first two signals indicate the voltage and current produced by the comparator depending on the sensors input. The signal at the bottom is the directional signal which gives the direction in which the motor should rotate either clockwise or anticlockwise; this is indicated by the change in phase of the signal.

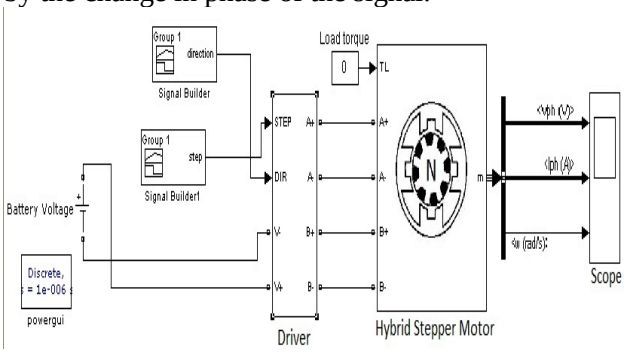

Fig.5 Automatic Wiper Control System

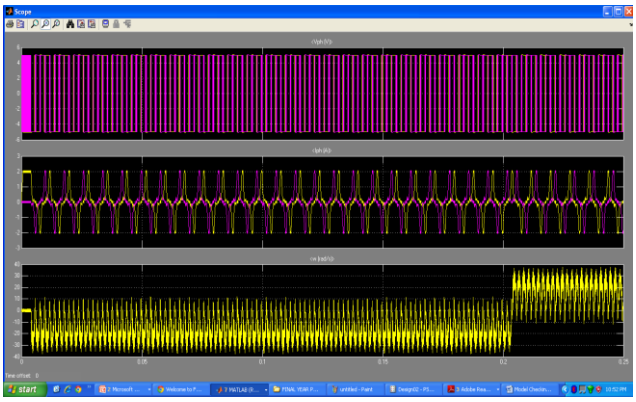

Fig.6 Matlab simulation

6 Implementation using SoC

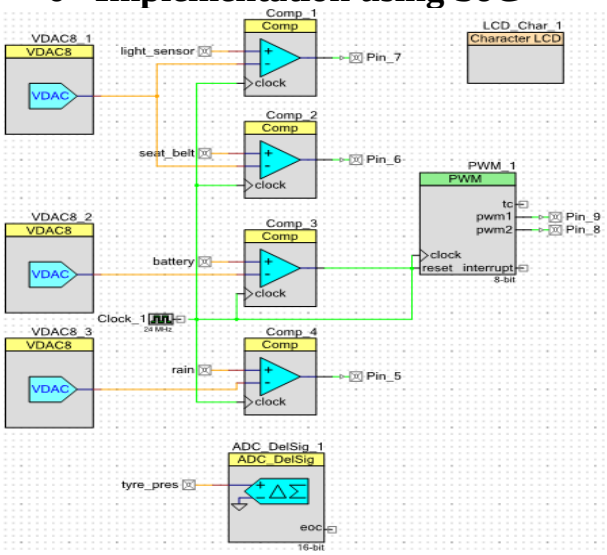

Fig.7 System Built on Chip

The programmable system on chip micro controller is an advanced mixed signal controller that uses an 8051 core. The intellectual properties are embedded as blocks within the micro controller. The controller

consists of analog, digital as well as mixed signal blocks which can be configured both at the system as well as chip level. The controller finds use in a number of safety critical applications as it is easy to re-configure and debug. The model implemented using this controller consists of three DAC blocks, four comparator blocks, one PWM control, an ADC and a Liquid Crystal Display as shown in Figure 7. The DACs are used to generate analog references that are given to the comparators. A clock is used to synchronize the working of all the blocks present in the system. The comparators are used to compare the inputs from the senor to the reference value generated. The outputs of the comparators are given to external physical components that perform the necessary action. An ADC is used in the pressure monitoring system to convert the analog voltage value to a digital component and later into a numeric value. The Liquid crystal display is used to print the entire necessary warning signal depending on the output of the comparators.

C Coding in PSoC

```
#include <device.h>
void main()
{          int MV;
           int sb, hl, bat;
   /* Place your initialization/startup code here (e.g.
MyInst_Start()) */
          ADC_DelSig_1_Start();
          ADC_DelSig_1_theACLK_Start();
          Clock_1_Start();
          Comp_1_Start();
          Comp_2_Start();
          Comp_3_Start();
          Comp_4_Start();
          LCD_Char_1_Start();
          PWM_1_Start();
          VDAC8_1_Start();
          VDAC8_2_Start();
          VDAC8_3_Start();
          VDAC8_1_SetRange_4V();
          VDAC8_2_SetRange_4V();
          VDAC8_3_SetRange_4V();
   for(;;)
   {      ADC_DelSig_1_StartConvert();
if(ADC_DelSig_1_IsEndConversion(ADC_DelSig_1
_RETU          RN_STATUS)==0x01)
{
ADC=ADC_DelSig_1_GetResult32();
MV=ADC_DelSig_1_CountsTo_mVolts(ADC);
if(MV>=4098)
{
LCD_Char_1_PrintString("WARNING!      Pressure
above Limits");
LCD_Char_1_PrintNumber(MV);
}
else
{
LCD_Char_1_PrintString("SAFE! Pressure within
Limits");
LCD_Char_1_PrintNumber(MV);
}
}
hl=Comp_1_GetCompare();
sb=Comp_2_GetCompare();
bat=Comp_4_GetCompare();
if(hl==0)
LCD_Char_1_PrintString("Headlight      switched
off");
else
LCD_Char_1_PrintString("Headlight      switched
on");
if(sb==0)
LCD_Char_1_PrintString("SAFE!          Seatbelt
Locked");
else
LCD_Char_1_PrintString("UNSAFE!  Please  put
Seatbelt");
if(bat==0)
LCD_Char_1_PrintString("WARNING!       Battery
Low");
else
LCD_Char_1_PrintString("SAFE! Battery safe");
}}
```

6 Conclusion

This paper consists of a proposed model that is applied in safety critical automotive applications. The model is verified using a symbolic model verifier and the verified model is implemented using a programmable system on chip micro controller. The implemented model is verified for its fault tolerant behavior. As a part of further research, we are working towards proposing an intelligent architecture that uses bio inspired MEMS sensors to detect changes in values and an artificial intelligent controller to assess the values obtained from the sensors.

References:

[1] Xi Chen, “Requirements and concepts for future automotive electronic architectures from the view of integrated safety”, PhD Thesis, Universitätsverlag Karlsruhe, 2008.

[2] C. Lu, Jean-Charles Fabre, Marc-Olivier Killijian, “An approach for improving Fault-Tolerance in Automotive Modular Embedded Software”, Proc. of the 17th International Conference on Real-Time and Network Systems, 2009.

[3] R. Baumann. The impact of technology scaling on soft error rate performance and limits to the

efficacy of error correction. In Digest of the International Electron Devices Meeting IEDM'02, pages 329–332, 2002.

[4] S.Chandrasekaran et.,al., "CAR Based Safety Model in Automotive Software Engineering", In the Proceedings of Recent Researches in Software Engineering, Parallel and Distributed Systems",,University of Cambridge, U.K. Feb 2011,pp 206-211.

[5] Juan R. Pimentel, "Designing safety-critical systems: A Convergence of Technologies", Kettering University, USA.

[6] Miroslav Popovic and Ilija Basicevic, " Formal verification of embedded software based on software compliance properties and explicit use of time", INTERNATIONAL JOURNAL OF COMPUTERS Issue 3, Volume 5, 2011.pp 423-430.

[7] Hongkun Zhang, Wenjun Li, and Jun Qin, "Model-based Functional Safety Analysis Method for Automotive Embedded System Application", International Conference on Intelligent Control and Information Processing, China 2010, pg-761-765.

[8] Emmanuel Touloupis, James A Flint and David D Ward, "Safety-Critical Architectures for Automotive Applications", Electronic Systems and Control Division Research, Loughborough University, 2003, pg-47-49.

[9] A. Pretschner, M. Broy, I.H. Kruger, T. Stauner, "Software Engineering for Automotive Systems: A Roadmap", Proc. of Future of Software Engineering, pp. 55-71, 2007.

[10] D. Jhalani, S. Dhir, "Survey of Fault Tolerant Techniques in Automotives", University of Wisconsin Madison.