ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 6, Issue 5, May 2017

Degrees Of Failure Strategy for Electronics Stability Program

Punya Maller[1] Jayanthi P N[2] Harish Kunatharaju[3]

***Abstract*—Automotive Engineering has become predominant nowadays due to its service to mankind as it integrates several components that provides assistance to the driver. In general, the system should be fault free for proper functioning of all the components. This paper discusses the fault detection and degradation strategy that can be applied to automotive. The work proposes developing an environment for Diagnostic system that is robust, reliable and efficient to handle any faults occurring in the system. The paper also discusses regarding the Test suite for DSM components in ABS ECU system.**

***Keywords*—Fault detection; Degradation strategy; Diagnosis and system management(DSM); antilock Braking system (ABS); ECU; Test suite.**

I. Introduction

Automotives nowadays have been equipped with both active safety and passive safety systems. Active safety systems are those systems which provide safety before accidents happens (preventing) and minimize the effects of the crash. Passive safety systems are those systems which help to reduce the effects of an accident.

There are two different ESP's for active and passive safety systems. ESP of passive safety systems consists of air bags, seat belts etc. ESP of active safety systems consists of Antilock Braking Systems (ABS), Traction Control Systems (TCS), Vehicle Dynamics Control (VDC), Power Train (PT), and many Value Added Functions (VAF's) like Hill Hold Control (HHC), Hill Descent Control (HDC), All Terrain Progress Control (ATPC) and many others. High End cars like Mercedes Benz, Audi, Jaguar, Land rover, Porsche etc. will have the above functionalities. Electronic stability Control (ESC) was first introduced by Bosch in 1995 for the Mercedes Benz S-Class sedan. Since then, worldwide more than ten million systems are produced and marketed as ESP- Electronic Stability Program by Bosch.

Antilock Braking System (ABS), Traction control system (TCS), Vehicle Dynamics Control (VDC) and some Value added Functions (VAFs) are integral components of an ESP system. ESP is a holistic system that can control entire movement of a vehicle.

Studies about automobile accidents show that more than 30% of automobile accidents are caused by skidding, which usually occur on the vehicle with conventional brake system. Major cause for this is Wheel lock up during braking. Electronic stability control is a technology that improves a vehicle's stability by detecting and reducing skidding [1]. When ESC detects loss of steering control, it automatically applies the brakes to help "steer" the vehicle where the driver intends to go. Braking is automatically applied to each of the wheels individually, such as the front wheel that is outside to overcome oversteer or the rear wheel on the inner side to counter understeer.

Whenever there is an ESC failure, it needs to be handled and the system must be degraded accordingly due to its interdependency on other components of the automotive. This requires diagnosis of the system[2] and hence we have a (diagnosis) DSW component in the car.

II. Wheel Speed Sensors of ABS

There are two types of wheel speed sensors basically: active and passive WSS [3]. Active WSS require an external power supply to produces an output whereas passive WSS does not exclusively require a power supply.

A. Passive Wheel speed Sensors

This class of WSS is installed directly connected to the wheel hub to which a pulse wheel is attached. It consists of a pole pin wounded by a coil and is connected to a permanent magnet. This produces a magnetic field that extends up to the pulse wheel. This change the magnetic flux as the pulse wheel rotates due to the successive alternation of the wheel teeth and the gap. This in turn induces an alternating voltage which is measured. The frequency and amplitude of the voltage depend on the speed of the wheel.

B. Active Wheel speed Sensors

This class of WSS may use either a pulse wheel or a multipolar ring. Magneto-resistors are used in detecting the magnetic field generated by the multipolar ring. The associated sensor electronics generate a digital signal. The signal sent to the control unit is a pulse width modulated signal. When using a pulse wheel a magnet is externally placed over the sensor. Hall sensors can also be used instead of magneto-resistors as they can detect close to zero speed as well as they can react to smallest change in magnetic field.

C. Requirement and implementation

The Wheel speed information is one of the primary signals for the ESP system. Wheel-speed Sensors are used for calculation of the wheel speed and vehicle speed. It is used within the application software to estimate a vehicle reference speed and to calculate the slip of each wheel. As mentioned, signals are generated by impulse wheels or magnets which are located directly on the wheel. Signal is transmitted to the ECU via cable.

The wheel speed is calculated using the basic formula, Speed = Distance/Time

The WSS failure word task in the project is a 120ms task which means that every 120ms a check is made to see if there are any issues with each of the sensors.

Three different conditions were reported as faulty conditions where a fault was required to be logged as specified by the BOSCH customer.

1. Monitoring of wheel speed differences-1. Faulty wheel Front Left.

Monitoring algorithm-If the difference between maximum and minimum wheel speed related to a reference speed exceeds 5% a wheel specific wheel speed sensor fault is set if the faulty wheel is always the same or the wheel speed is near standstill (default value 5 kilometers per hour) or when one of the wheel shows a different speed than the remaining three wheels. For eg, if three wheel show a speed of 100kmph and one of the wheel shows a speed of 90kmph then this fault is logged..

Detection time- within 70 sec.

2. Monitoring of wheel speed differences-2. Faulty wheel not assignable

Monitoring algorithm-If the difference between maximum and minimum wheel speed related to a reference speed exceeds 5%. A generic wheel speed sensor fault is set if no wheel specific fault could be determined i.e. the wheels on the front axle have different speeds that the wheel on the rear axle. During system mode ABS and backup-EBD (Electr. Brake Force Distribution) and below 20 kilometers per hour the deviation is 6 kilometers per hour for velocities < 100 kilometers per hour and 6% for a velocity > 100 kilometers per hour.

Detection time- within 70 sec.

3. Wss fault due to missing wheel speed

Monitoring algorithm-If the difference between maximum speed shown by three wheels and minimum speed shown by one of the wheel exceeds 5%. The monitoring checks for the signals from the wheels. When the vehicle is moving, if there is sudden loss of signal from a wheel i.e. the wheel shows zero speed, this fault is logged, since the speeds received from individual wheels speed sensors is used to calculate the vehicle speed. During system mode ABS and backup-EBD (Electr. Brake Force Distribution) and below 20 kilometers per hour the deviation is 6 kilometers per hour for velocities < 100 kilometers per hour and 6% for a velocity > 100 kilometers per hour.

Detection time- Depends on driving situation.

The vehicle speed is usually the average of the speed received from the four sensors. Hence if any of the sensors are faulty, a fault needs to be logged and should be informed to the driver Also the ABS will be unavailable since these sensors are one of the main components in it.

III. SYSTEM MODE MANAGER (SMM)

The system modes are controlled by the system mode manager (SMM). The inputs are different system wakeup sources (requester). The priority handler calculates the highest system mode request of all wakeup sources. These system modes may be requests by the project specific requester (e.g. wakeup unit, network management). At the system start all requester request INIT till they finished their initialization. The state machine determines the new system mode depending on the current and requested mode. Depending on the current system mode and the highest requested system mode the state machine calculates the new current system mode.

A. SMM states

The System Mode Manager defines different fix "System Modes":
INIT
OFF
NET OFF
NET ON
NORMAL

System mode "INIT"
Action of system: The system is in init phase (after a soft- or hardware reset). The mode "INIT" is usually not requested by any requester.
System mode "OFF":
There are no external and internal requests. It is not necessary to keep the ECU alive. Allowed transition to NET OFF.
System mode "NET OFF"
The ECU wakes up because of external conditions (requester requests NET OFF), but there is no communication on the network necessary. The ECU stays alive because of internal hold conditions (e.g. EEPROM Handler busy). If one requester did not finish his initialization (stays in "INIT") the system mode transition to "OFF" will not be allowed.
System mode "NET ON"
The ECU wakes up because of external conditions and sends information to the network. The ECU stays alive because of hold conditions (e.g. internal condition: vehicle speed).
System mode "NORMAL"
The ECU is in "NORMAL" mode if at least one requester requests this mode. Usually occurs only if ignition is on as requested by wake up unit (WAU).

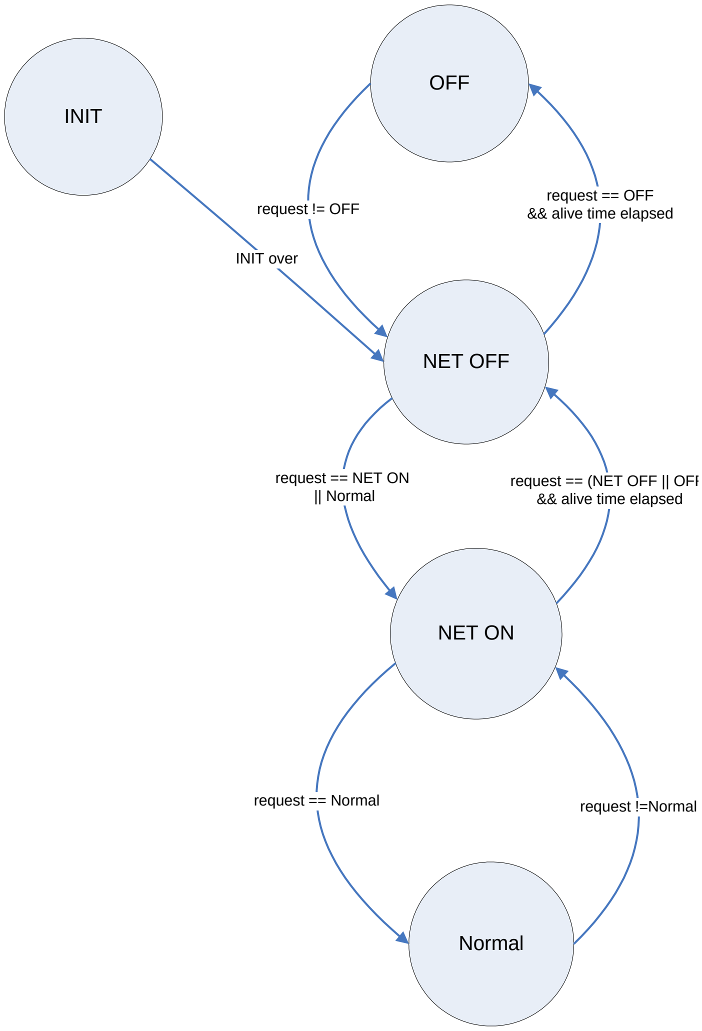

Fig 1. SMM states

B. Implementation

When ignition is switched off, then all the components in system should finish their close up activities such as saving data in EEPROM through pdm and request system to OFF state. System will not shut down until all the components request OFF state, this can happen only when component finishes its activities or else it will continue requesting system to be ON state and it will never shuts down. The system will wait for max time of 15min after ignition off to finish these activities. This time is called maxafterrun time. If still components such as APB, TCM and VM still requests system to be in Normal state then timeout will be logged. DSW component should store the timeouts individually for each specified component such as APB, VM and TCM.

DSWs role is to store this timeout data in a byte. This byte value should be initialized by byte info from Diagnosis Communication component (DCOM), which will be having info about last 5 timeouts. If timeout occurs then DSW byte will be shifted left and new timeout presence will be updated in LSB and this will be stored in EEPROM through PDM and Dcom will update accordingly. It's required to consider only last five timeouts info from latest

DSW should do this process of updating timeout details and storing in EEPROM though PDM with in 20sec maximum. This process usually starts after maxafterrun and system wait for 20sec to finish this update activity. If still it takes more than 20sec then DSW should forcefully shut down the system irrespective of any component is requesting for SMM Normal or Net On.

IV. ASCET

ASCET is developed by ETAS and used by BOSCH since 1990. ASCET-MD (Modelling and Design) tool is used to model and design the software according to the customer requirements.With ASCET-MD software module components can be specified both on the physical level with block diagrams, state machines and textually in the high-level languages ESDL or C. ASCET also supports the development of AUTOSAR software components.

Functionalities of ASCET

- Modeling of AUTOSAR software components
- Integrated code generator for floating point and fixed-point arithmetic (PC simulation)
- Easy integration of C code, library functions and specific arithmetic services
- Simulation of software components

V. TEST SUITE

Test suite is basically a process of performing static analysis of the code. Test suite is a method of DSW program debugging that is done by examining the code without executing the program. This is done to ensure that the code meets proper coding standards (often called Code Review). The main advantage of performing this is it finds issues with the code even before the integration phase.

As every new generation of car has new features being added, the systems are getting even more complex and so verification is not only important at the component level but also at the system level. Testing is done at the early stages of development, during system integration, prior to product launch and regular operation of car during maintenance and diagnosis[4].

DSM has huge code that supports various variants in car. This part of code is unique to each variant. In this Project, CMP is the customer main path. This is the methodology of maintaining various projects in single module for better maintenance and avoids rework for common code functionalities. Here code is maintained to be compatible for all the projects in CMP.It supports all the projects in CMP of various functionalities with handling of fault monitoring,, state management etc.With all this merging of projects in CMP, DSM is now a component with huge code changes. So for every release of software, static analysis of the code should be performed as per the standards.

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 6, Issue 5, May 2017

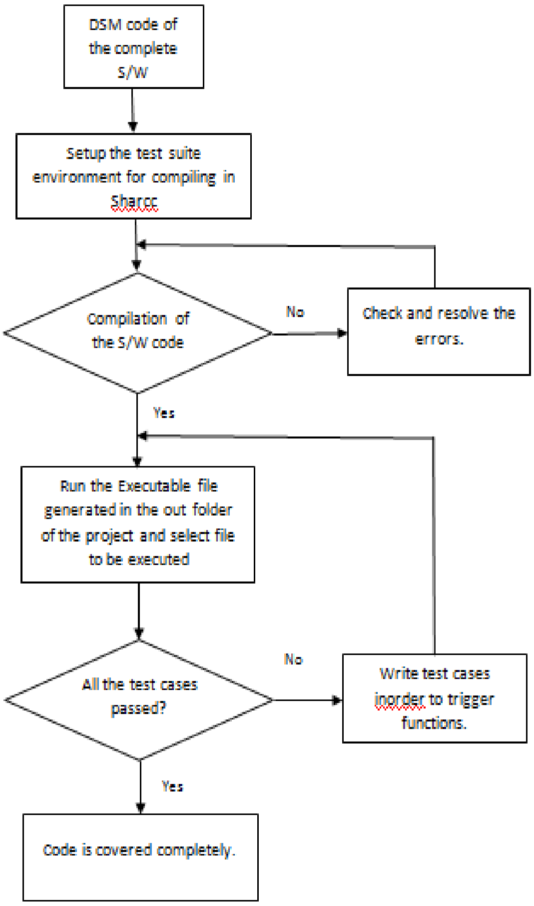

Fig 2 Flowchart showing the methodology for test suite.

A. *Methodology for Test Suite*

- There are several modules in DSM and each module has code. Here test files need to be maintained with all the possible test cases for each module separately and these test files needs to be compiled along with complete code in test suite mode.
- The tool SharCC is used for compilation of the whole project.
- The initial configuration to use the DSWtestsuite mode is done. For this we use two files namely makeoption and Fileselection
- The project for which the test suite is been done is selected through the MTC buildoptions window and put for compilation.
- Once compilation is through then gen folder will be generated with variant/project number (each project need to compile separately).
- In gen folder if we execute the executable in the out folder, it will execute the code module by module separately without test cases in test files and generate a report mentioning the no. of test cases passed or failed.
- For the test cases that failed we need to write the test cases to trigger those functions and trying executing again.
- Here the main aim is to make sure every module test cases is passed and 100% code coverage is achieved.

VI. RESULTS

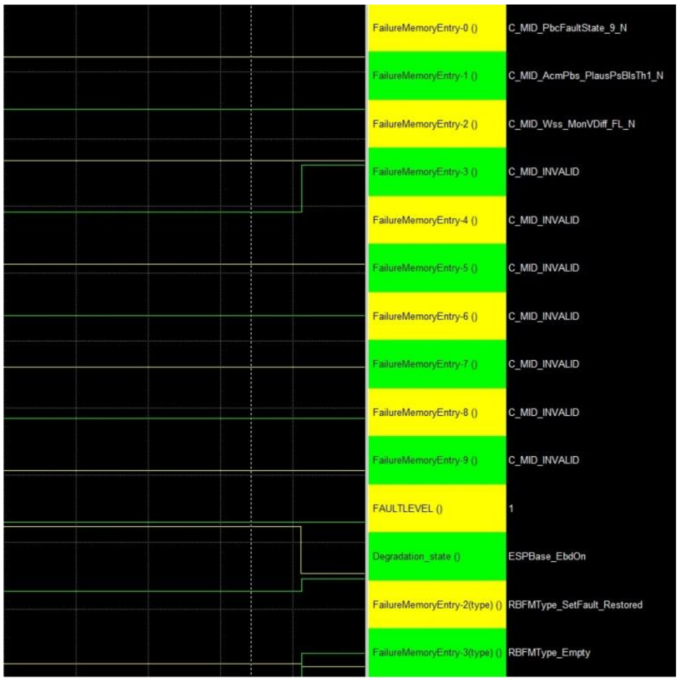

Fig:3 Logging of fault when one of the wheels has slightly different speed than the others.

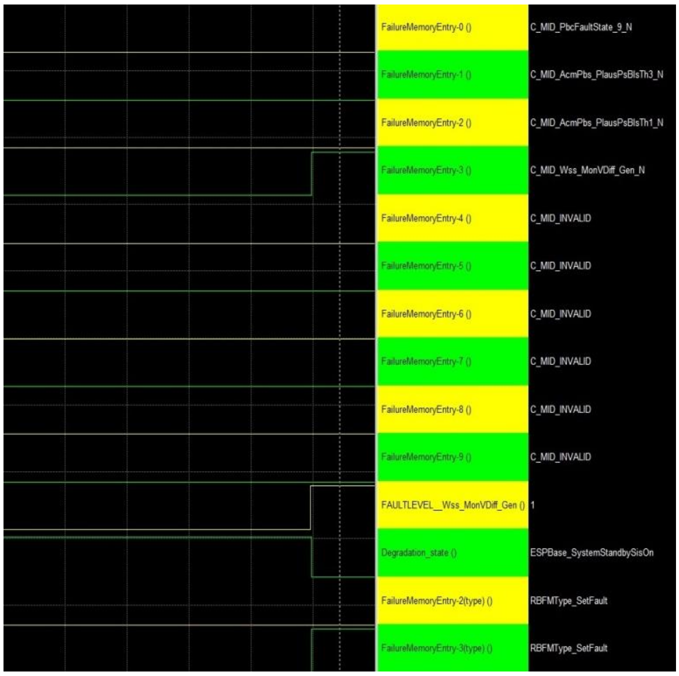

Fig:4 Logging of fault when the wheels on the front axle has slightly different speed than the other axle.

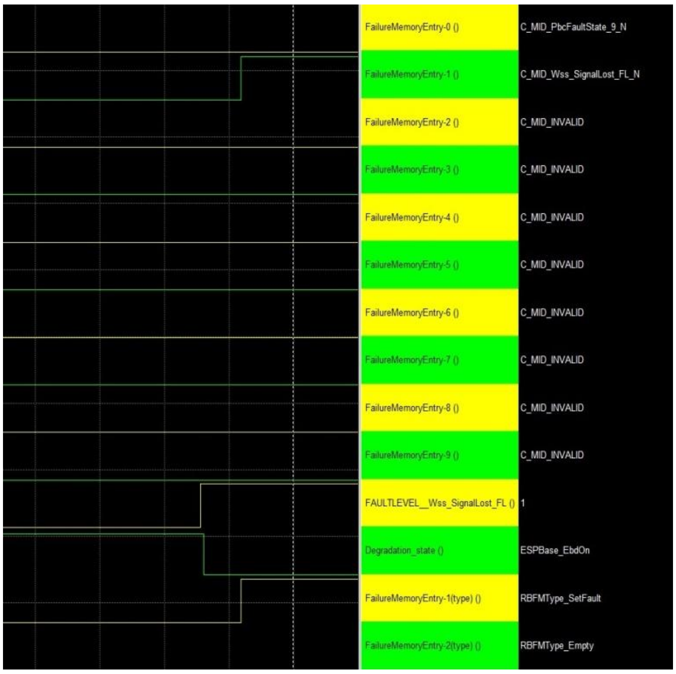

Fig:5 Logging of fault when one of the wheel shows zero speed and all others show some speed.

These tests are run on a labcar which is a real-time PC. LabCar testing assures the quality of ECU. LABCAR is part of the standardized ETAS tool family.

The car model software is flashed on to the ECU that is interfaced to the labcar. The labcar software provides GUI for the user to simulate the environment to check the results.

TestGroup: DIA - SFB

- **TestClass: SFB CheckFailureStateTransitions**
  Number of tested objects: 460. All tests in this class passed. (CTC = 20616)

- **TestClass: SFB CheckPrestorageOfFreezeFrames**
  Number of tested objects: 460. All tests in this class passed. (CTC = 32075)

All tests in this group passed. (GTC = 52691)

TestGroup: SMM

- **TestClass: SMM Requester**
  Number of tested objects: 6. All tests in this class passed. (CTC = 72)

- **TestClass: SMM PrioHandler**
  Number of tested objects: 1. All tests in this class passed. (CTC = 498)

- **TestClass: SMM StateHandler**
  Number of tested objects: 1. All tests in this class passed. (CTC = 196)

All tests in this group passed. (GTC = 766)

Fig:6 Test Suite Results

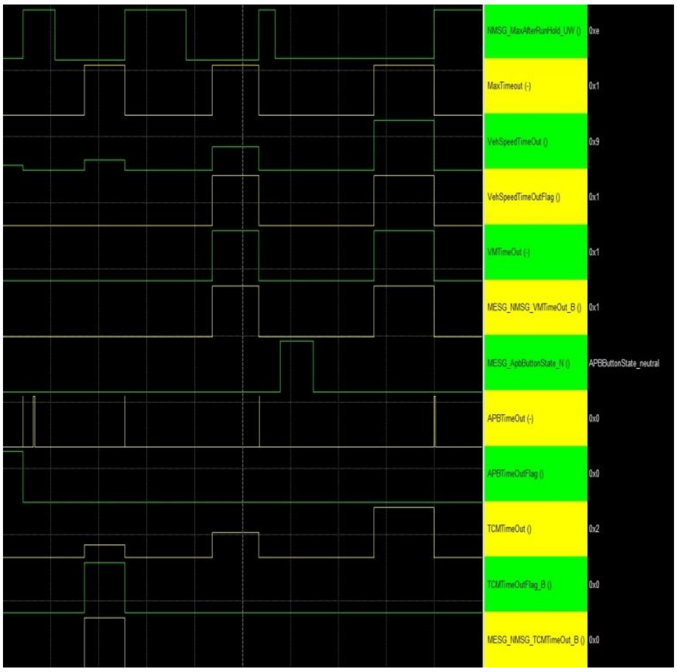

Fig:5 SMM 20sec implementation result

CONCLUSIONS

This paper shows how diagnosis as a component plays a very important role in automotive. Thus the paper describes the fault detection and degradation strategy for WSS and SMM in a car. Three faulty conditions together with the mode management that were mentioned important by the customer were developed. The work here makes contribution to detect faults with WSS under real time operation and to manage the mode for better performance of the system were considered.

The paper also shows a method for performing the test suite for checking the code coverage of the DSM component.

References

[1] Chih-Keng Chen et. al. "A compensated-yaw-moment-based vehicle stability controller",*2008 Chinese control and Decision Conference (CCDC 2008).*

[2] J. A. F. Vinsonneaut et. al. **"**Improved SI engine modelling techniques with application to fault detection", *Proceedings of the2002 IEEE International Conference on Control Applications, September 18-20, 2002 Glasgow, U.K.*

[3] William J. Fleming,New Automotive Sensors—A Review, *IEEEsensors journal, vol. 8, no. 11, november 2008*.

[4] Oliver Praprotnik et. al. "A Test Suite for System Tests of Distributed Automotive Electronics", *2009 Second International Conference on Advances in Circuits, Electronics and Micro-Electronics.*

ACKNOWLEDGEMENT

The authors here would like to thank the RV college of engineering ,Electronics department (RVCE) and Robert Bosch Engineering and Business Solution Pvt. Ltd. for their extended support.

BIOGRAPHIES

Punya Maller[1], received B.E in Electronics and Communication Engineering from Jain College of Engineering, Belgaum, India in 2015. She is currently pursuing M.Tech in Communication systems from RV College of engineering, Bangalore. Her subjects of interests are Embedded system designs and wireless communication.

Jayanthi P.N[2], received B.E in Electronics and Communication from Ambedkar institute of Technology, Bangalore and M.Tech in Digital Electronics and Communication from Dayananda Sagar College of Engineering. She is currently pursuing Ph.D in communication systems. Her subjects of interests are Embedded system design, wireless communication and MIMO detection algorithms.

Harish Kunatharaju[3], received B.E in Electronics and Communication from Sri Vasavi Engineering college, Tadepalligudem, India. He is currently working as Senior Software engineer at Robert Bosch Engineering and Business Solution Pvt. Ltd. His subject of interests are Embedded systems, C and C++ programming and LINUX.