

AUTOSAR based Functional Development of Vehicle Road Lights

Swetha Patil

MECS, Dept. of E& IE.

Dayananda Sagar College of Engineering
Bangalore, India.

Mrs. Meharunnisa S P

Associate Prof, Dept. of E&IE,

Dayananda Sagar College of Engineering,
Bangalore, India

Abstract: The total number of Electronic Control Units (ECU) involved in building a vehicle is increasing day by day. Hence the AUTOSAR (AUTomotive Open System Architecture) consortium was developed to manage the growing electronics complexity in order to improve the cost and efficiency without any compromises with quality as well as reusability. Increasing software complexity is a major reason for the delay in a project and cost overrun. The AUTOSAR project is one of the challenging solutions in defining a methodology in order to reduce the efforts of software development. The main objective of the project is to develop a body control module for a platform independent vehicle using AUTOSAR and to protect the vehicle components using this ECU. This paper presents the development of a Vehicle Component taking Vehicle Road Lights as a case study using AUTOSAR methodology. The module is developed and verified in a MATLAB-SIMULINK environment.

Keywords—Model Based Design (MBD), AUTOSAR, Electronic Control Unit (ECU) and Vehicle Road Lights (VRL)

I. INTRODUCTION

Model Based Design (MBD) is a widely used and accepted approach for graphical execution of a system as it decreases the project development time and addresses the complexity of applications which is very important in this competitive automotive world. Capabilities such as automatic verification and validation and code generation are additional key advantages that make the development processes more effective and efficient. The design flow of MBD is shown in figure 1.

Standardization is needed in order to manage the increasing complexity of software and to improve modularity, flexibility, portability and reusability. AUTOSAR is one of the best and challenging ways to achieve this standardization. AUTOSAR consortium was developed with the objective of defining an open standard for an Electrical/Electronic (E/E) architecture. It addresses several issues such as managing the

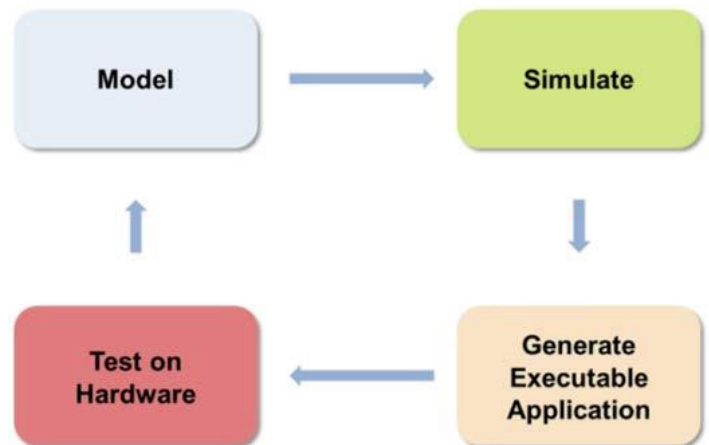


Figure 1: Model Based Design (MBD) flow

increasing complexity of E/E automotive systems, flexibility for product modification, update and upgrade scalability of solutions within and across product lines and maintaining the quality and reliability of the system. AUTOSAR has united more than 100 companies, automobile manufacturers, suppliers and tool vendors to develop and design a standard architecture for electronic control units.

Fuses are used to protect the wiring and electrical equipment in vehicles. AUTOSAR based ECU is a special type of fuse box used in off-highway vehicles which provides the protection of vehicle components effectively. As the AUTOSAR architecture allows the re-usage of the models hence it is introduced with this ECU in designing Vehicle Road light subsystem.

II. AUTOSAR METHODOLOGY

The main aim of the AUTOSAR is to develop encapsulated application software which is abstracted from the hardware and is totally independent of the communication technology and operating system. This software component can be located to any different ECU and it is reusable across different vehicle suppliers and

manufactures. The concept of AUTOSAR is displayed in figure 2.

The AUTOSAR software is classified into three parts) **1) Software Component:** It is the functional part of the ECU software which consists of software components that are hardware-abstracted. These software components together form the application layer.

2) AUTOSAR Run-time Environment (RTE): It is the middleware software that allows ECU functional development independent of the ECU hardware. This layer provides the communication between the software components that are present in a technology independent application layer and the basic software layer.

3) Basic Software (BSW): This layer consists of hardware-dependent parts of the software components as well as the operating system, communication drivers and AUTOSAR services.

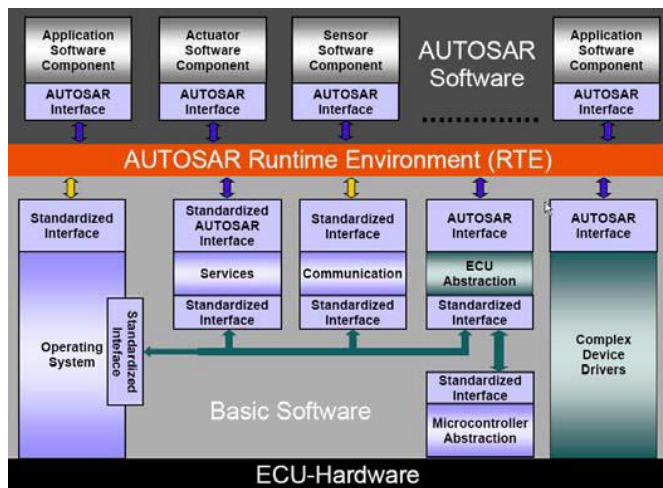


Figure 2: The AUTOSAR Software Architecture

III. WORKING OF ECU

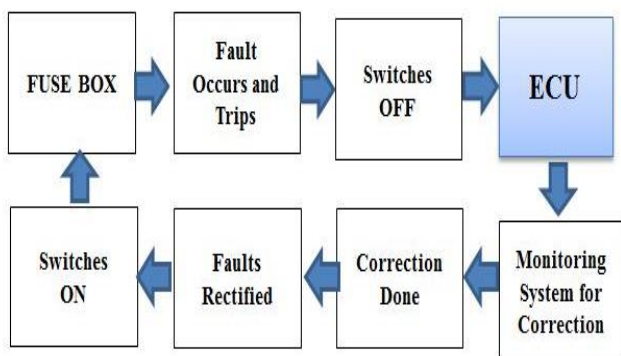


Figure 3: Working of ECU

In Normal Fuse Box, when fault occurs, it trips OFF and the Switches are forced to be OFF. But the advantage of implementing, Electronic Control Unit for Normal Fuse Box

is that, it checks for the faults and correction is made automatically and the switches regain normal operating conditions. Here in this Figure 3. Initially in the Fuse Box, fault occurs and the Fuse Box trips OFF and the switches are OFF. Now ECU comes into action by monitoring the system continuously, corrections are made by the ECU and faults are rectified due to which switches again get back to their normal operating conditions.

IV. VEHICLE ROAD LIGHTS(VRL)

The Vehicle Road Light (VRL) is an AUTOSAR application software component which controls the exterior lights of a vehicle such as Dipped/Main Beam Head Lights, Front Fog Light, Approach Light, Rear Fog Light, Stop Light, Emergency Brake Light, Reversing Light, Position Lights, Home Safe light and Auxiliary. The component receives the inputs from various sensors such as brake pedal switch, head light switch, beam control switch and other sensors. This software component is usually found in the body computer of an ECU. The functions include:

1. Dipped Beam Lights
2. High Beam Lights
3. License plate lamps
4. Beacon Lamps
5. Marker Lights

The objective is to execute the concept of AUTOSAR in a model-based development environment. It consists of all the executable models for the Application Layer and for simulatable RTE and OS. The Modular approach of the VRL system is given below which consists of three parts

1. Sensor part
2. Functional part
3. Actuators part

The inputs to the VRL module are called as Sensors (VRL-S), the body or the logical part of the module is known as the Functional part (VRL) and the outputs are called as the Actuators (VRL-A). The functional diagram of the VRL subsystem is given in figure 4. The modeling of the system is done in MATLAB Simulink.

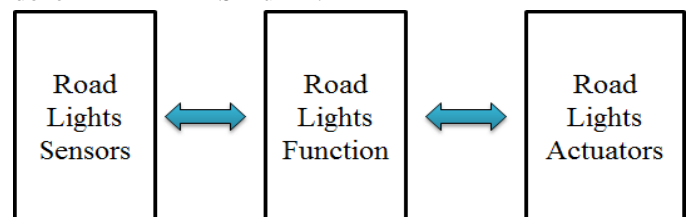


Figure 5: Modular Approach of VRL

The high beam lamps shall be controlled via a high beam switch and a momentary switch used for the flash to pass. This flash to pass switch signal shall turn on the high beam lamps regardless of the state of low beam lamps and will not affect the state of low beam lamps. The High Beam lamps may also be turned on via a received CAN message, the Parameter Group Number (PGN) will be determined at system design. Similarly the low beam lamps will be controlled via a low beam switch and may be turned on using a CAN bus message.

Auxiliary lamps may be used in order to provide high intensity light so that the driver is able to see at longer range than the vehicle's high beam headlamps. The VRL shall provide an I/O to drive an additional pair of Auxiliary (also known as Flip-Up) low beam headlamps using the Auxiliary switch as input to select between regular low beam and auxiliary low beam. The Auxiliary Low Beam Headlamps shall be controlled using the combination of auxiliary switch and Low Beam switch. If auxiliary switch is TRUE, auxiliary lamps take the place of the regular lamps. Similar procedure is carried for high beam auxiliary lamps.

The beacon lamps shall be controlled via a switch. The beacon lamps may also be turned on via a received CAN message whose PGN will be determined at system design. The auxiliary beacon shall be controlled by the beacon lamp switch if equipped.

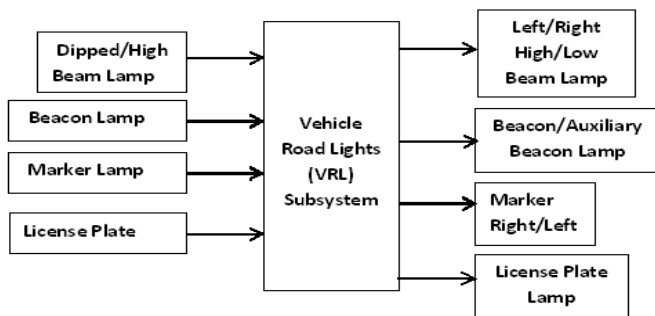


Figure 5: VRL Functional Block Diagram

The VRL Subsystem shall be supplied a status signal that identifies whether the configuration is NASO/ISO standard which shall be used to determine the marker lamp functionality. When the Headlamps low beam switch OR the marker lamp switch are turned on, the marker lamps shall be turned on.

V. TESTING & VALIDATION

Once the modeling of the VRL subsystem is completed, the model is verified by a UNIT and Model In Loop (MIL) testing methods. Unit testing is a method where the individual units of a source code or sets of one or more

program modules together with associated control data, usage procedures, and operating procedures are tested to find out whether they are fit for use. MIL is a method to verify or to test whether it is implementable.

Following the UNIT & MIL testing, Bench testing is to be carried out which consists of composing of AUTOSAR'S Virtual Function Bus (VFB) where the application software components are integrated into VFB and then the Run Time Environment (RTE) is generated where the outputs from the VRL module are given to it.

VIII. CONCLUSION

A model-based design process with AUTOSAR compliance enables the development of standardized software with high quality and efficiency. Such tools are also evolving in market to create AUTOSAR-compliant software. The low-cost tools made in-house can also be efficiently used for the development of AUTOSAR-compliant software components. In this study, we proposed a methodology where the model-based design method will be integrated into standardized software architecture. The application software will be modeled using runnable and software components. The VRL subsystem will be developed by the proposed software development methodology and the feasibility of the methodology can also be verified.

IX. ACKNOWLEDGMENT

I offer my sincere thanks to my beloved project guide Mrs. Meharunnisa S P, Assoc. Prof., Department of Electronics and Instrumentation Engineering, for her valuable suggestions, expert advice, unending support and constant guidance that helped me in completing the paper successfully. I express my deep sense of gratitude and profound feeling of admiration to Mr. J.S Rajashekhar, HOD, and Department of Instrumentation Technology for his continuous encouragement.

REFERENCES

- [1] L. Vitkin, T.K. Jestin, "Incorporating Autocode Technology into Software Development Process", ICSE 2004, pp.51-57
- [2] MathWorks Automotive Advisory Board (MAAB) "Controller style guidelines for production intent using MATLAB®, Simulink® and Stateflow® Version 2.0", Jan 29th, 2007
- [3] Andreas Köhler and Tillman Reck "AUTOSAR Compliant Functional Modeling with MATLAB®, Simulink®, Stateflow® and Real-Time Workshop® Embedded Coder of a Serial Comfort Body Controller"
- [4] //www.autosar.org