

# Software Product Validation of Body Controller ECU's using Intelligent Automation Methods

Eshwari A Madappa, Varsha P Y, Nectar Jinil



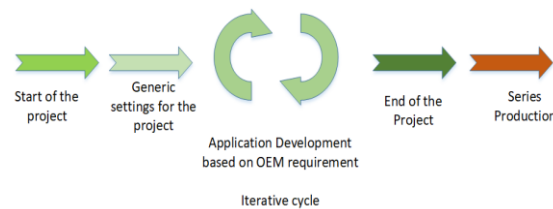
**Abstract:** In Continental the, Electronic Control Unit (ECU's) software consists of 2 parts, the basic software which is the generic part and the application software which does the product specific functionality. The project specific functionality depends upon the product type, like Body domain controller, immobilizer etc. Before the ECU is going to mass production, the controller has to be validated i.e., software and hardware. The main reason is to ensure, that once the ECU is fitted to the vehicle and there is a recall for ECU, it involves additional cost to the organization. The validation of the software includes, checking of the micro controller input, outputs, peripheral drivers, the initialization of the drivers, the timing measurement of the different components. All the parameters are controlled by the generic and application specific modules, and can be configured via configuration tool, manual configuration etc. Since the SW modules are complex and huge, so its need to perform validation of the SW modules manually. This is time consuming process, as it involves checking many c file, configuration files etc. The proposal is to automate the process using Python programming, which interacts with the C code, configuration files, debugger, interacting with the ECU's peripherals, components and validate the Software parameters before going for series production. In the proposed method the different steps can be automated for different micro-controllers i.e., Renesas RL78 and RH850 used in the ECU's and different architecture i.e., AUTOSAR and NON-AUTOSAR will be considered and the results will be compared with the manual validation effort and accuracy.

**Keywords:** Automation, Validation, ECU's

## I. INTRODUCTION

In Continental, there are many business units and one among them is Body and Security Business unit. The main product lines include Gateways, Intelligent body domain controller, Tire pressure monitoring systems, RF receivers, Immobilizers, Seat controllers and Comfort systems. Here the product is developed for different Original Equipment

Manufacturer (OEM's) based on the OEM requirement. To make the software development more generic it consisting of generic Software layer which deals with micro controller drivers, and related services. In Body and Security business unit, some projects that follow an automotive standard (AUTOSAR) and some projects that doesn't follow a specific automotive standard (NON-AUTOSAR)



**Fig. 1. Product Development Workflow**

The development workflow is shown in Fig.1. Before the software is going for series production series of validation of the ECU functionality needs to be done. This is done manually, which is time consuming and more tedious and erroneous. Any errors which is un-checked may lead to field issues which causes the vehicle to be recalled. This involves the organization credibility at stake and involves additional cost to the organization. The whole purpose of the proposed method is to automate the manual effort using more intelligent method to reduce the effort of the validation and reliable result.

## II. AUTOMOTIVE EMBEDDED DEVELOPMENT

The core of the automotive embedded development is the Electronic Control Unit which performs dedicated functionality. For example, in body domain controller ECU will perform the locking and unlocking functionality, the interior lighting functionality, wiper control logic, HVAC (Heating, Ventilation and cooling control logic). Similar to this there are many ECU's in the vehicle which perform a dedicated function. All these ECU's communicate and work together to form a complete vehicle architecture. Each of the ECU's internal component consists of micro controller. The micro controller used differs based on the application or functionality used. For time critical applications like engine management and airbags Infineon Tricore micro controller is used which has a better timing parameter. For the body domain controller, gateway ECU's which are not that time critical Renesas controllers like RH850 family, RL78 family are used. The main aim of the tier one supplier who will supply the ECU's to the OEM is to increase profitability and reduce the time to market. In the above example,

Revised Manuscript Received on June 10, 2020.

\* Correspondence Author

**Eshwari A Madappa\***, assistant professor, Electronics and Communication Engineering, Jss Science and technological University, Mysuru, Karnataka, PIN-570008, India. E-mail: eshwarinaveen@sjce.ac.in

**Varsha P Y**, MTech, Automotive Electronics, Jss Science and technological University, Mysuru, Karnataka, PIN-570008, India. E-mail: varsha031096@gmail.com

**Nectar Jinil**, Senior manager, Continental automotive component private limited, Bangalore, India. Email: nectar.jinil@continental-corporation.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

RH850 microcontroller and RL78 microcontroller is used for wide variety of applications. For each of the different applications the tire one suppliers cannot go from the beginning, as it will increase the cost and time to market. To avoid this a modular approach is followed in the ECU development, which involves making maximum re-usability and developing a layered architecture so that there is more re-usability.

Also, the life time of a microcontroller is 5 years, and after that a new platform will be selected due to advancement in the microcontroller manufacturer. For this a modular approach is followed. One such approach is the AUTOSAR architecture. If AUTOSAR is not used many of the suppliers will follow a modular approach and the architecture will be in house, to increase re-usability and provide cost optimization. One such Layered architecture for a non-automotive project is shown in Fig 2.

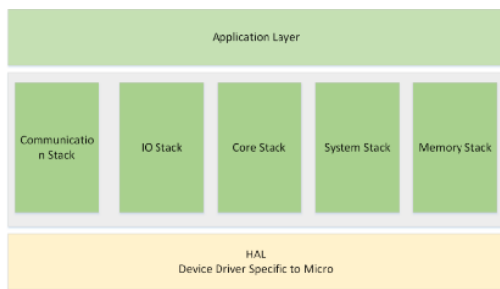


Fig. 2. Modular Architecture

Here in the figure the architecture is divided into 3 layers, Hardware Abstraction Layer (HAL), Service Layer and Application Layer.

The main advantage of this architecture is, if any microcontroller is changed only the HAL layer and few changes in the service layer is required and there is no change in application is required. This will ensure modularity in the development, increases re-usability and reduces cost. Another attribute that needs to be considered is that the microcontroller is used for different functionality as mentioned earlier, like BCM, Gate way, RF receiver etc. Each of these functionalities require different HAL, service layer parameters, needs to be configured. The configuration files are more like .xml files, which has nodes and that can be configured based on the function needs and using Java scripts. From the xml information .c, .h files can be generated. The Configuration file in xml helps to provide user interface to user for easy configuration of the micro controller modules like IO's, CAN cells, SPI cells, Peripheral devices, Memory access and memory allocation. This is one of the major concepts in AUTOSAR domain. It follows a standard across all the industry, and follows a common configuration.

III. PROPOSED METHOD

In this method, use of intelligent automation method is consider to automate the process of validation of ECU's functionality before it goes for series production. An overview of the complete process of the system to the problem of manual validation is shown in Fig 3.

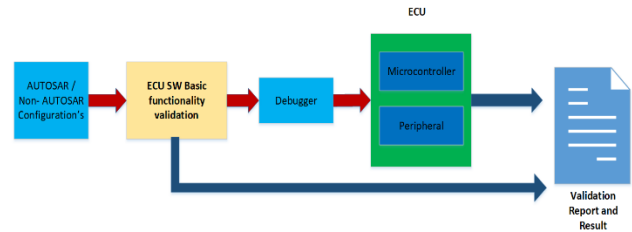


Fig. 3. The complete process of system

Consider a different input for different check phase. These inputs include the project configurations files, IO's initialization, etc. The parameters that are planned to validate includes the interrupt mapping from the generated binary file, EEPROM usage, IO initialization sequence. For the validation with the ECU the debugger has to be connected and controlled with the script. The debugger planned to use is systems debugger [1], and other measurement devices that can be controlled by Python [2] script. The project involves defining and coming up with algorithms and methods to automate the different validation steps in the software, and generating a detailed report out of it to provide information about the status of the validation and take corrective measures.

A.TEST AUTOMATION AND FLOW DESIGN

**Tool Information:** In this automation tool, user action gives accurate result as it is given by Graphical User Interface (GUI) is shown in Fig 4.

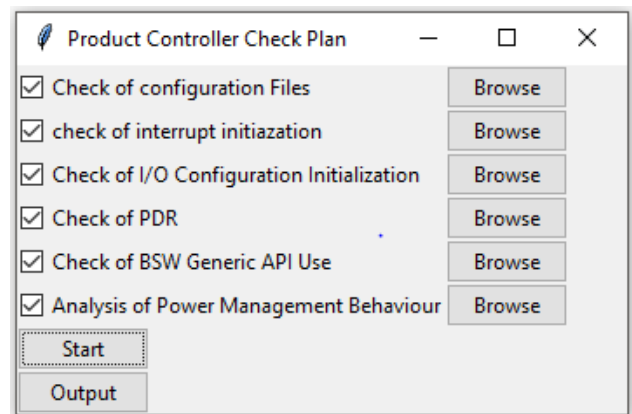


Fig. 4. Main GUI of customized test automation tool  
Software test automation method is described which allows the user to test in faster and more reliable way as shown in Fig 5.

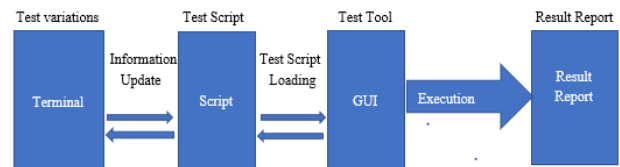


Fig. 5. Process of test generation tool

Testing process has considered 6 criteria and corresponding testing techniques. Where, C1 – Check of configuration files, C2 - Check of Interrupt Initialization, C3 - Check of I/O Configuration Initialization, C4 - Check of PDR, C5 - Check of BSW Generic API Use,

C6 - Analysis of Power Management Behaviour. C1-Check of configuration files: Check incorrect configuration data handling due to mistakes in configuration data processing. Here considering Extensible Markup Language (XML) format files to store data. The Configuration file in xml helps to provide user interface to user for easy configuration of the micro controller modules like IO's, CAN cells, SPI cells, Peripheral devices, Global setting, Core field, System setting and compare the data with the Hardware Software Interface (HSI).

C2-Check of Interrupt Initialization: Consider s19 and map files in these criteria. S19 Contains low-level machine code and addresses of a Assembly program or compiled C, which represents hexadecimal coded binary data used by the Freescale series of microcontrollers. Map files indicates the relative offsets of functions for a given version of compiles binary. The hardware vector table values of two bytes is obtained from the vector table address(0000H-007FH) in s19 file. Using the vector table address the interrupt services routine (ISR) address of 2/3 byte is fetched based on the s-record data format in s19 file. If the map file contains the virtual vector table then the isr is mapped with the map file.

C3 - Check of I/O Configuration Initialization: Consider win idea software and configuration tool. Ports configured as defined in HWSW interface sheet. Put a break point when the first ASW task is called. Extract from debugger the status of register and verify the value of all the configuration registers of the I/O (use SFR window).

C4 - Check of PDR: Goal is to ensure that the compilation, link options and mappings are right. Verify the LOCATE and UTIP sections of the PDR of the project. Check the last version of compiler and toolset to project. And also verify that the blocks mappings are correct depending of the memory size.

C5 - Check of BSW Generic API Use: The test is to have an extract of API use in order to analyses some criteria of Basic Software API. Provide the list of interesting Basic Software API used by the project.

C6 - Analysis of Power Management Behaviour: Check the proper configuration and safe low power system setting with a power manager specialist from SWP Project. Here consider LPM callouts, LPM tasks, LPM interrupts to list the low power tasks.

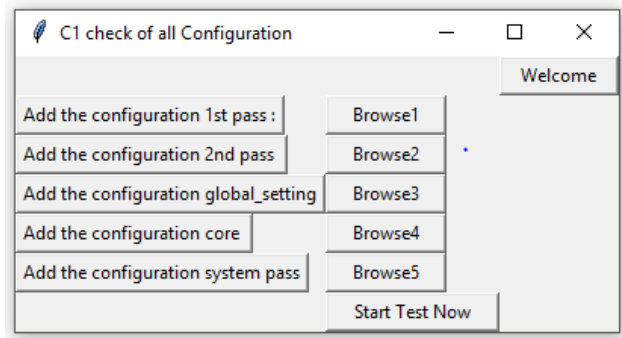


Fig. 6.GUI of input selection for C1.

2.The script for check of interrupt initialization is being written. The GUI for C2 is shown in Fig7, which consists of option to browse the input s19 file (binary file) and map file. Once the files are chosen, then start button is activated to run the script and the results is generated.

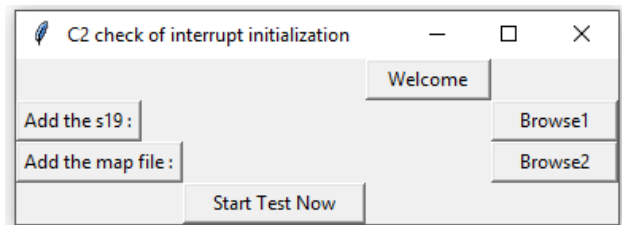


Fig. 7. GUI of input selection for C2.

3.Script is written for C3 the microcontroller IO pins initialization is generated in an HTML file format. Fig8 and Fig 9 shows the GUI and the outputs of C3 test.

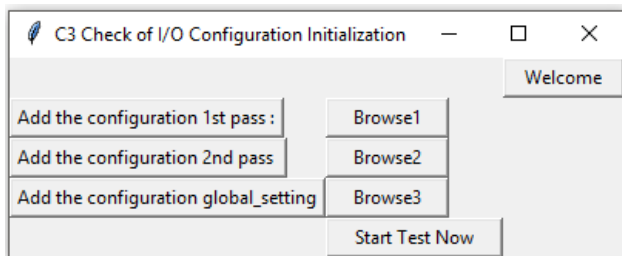


Fig. 8. GUI of input selection for C3.

IV. RESULTS

1.Configurations of microcontroller pins is done using the config tool. The pin configuration varies from one variant to the other from 100pins to 144 pins. The unused pins have default values in the config tool. Few pins are used for specific functions and the values for configuration like the type of port, initial values, directions, polarity are configured by the information provided in the HSI sheets.

Due to more pins and manual effort, the configuration being done is not always 100% correct. The verification of each configuration pins takes more time and human effort. In this paper the verification of this configuration is done by the automation method. The xml file generated is being used as input for the automation test. The generated output is in the html or excel format for readability, which is being compared with the HSI sheets. The results of C1 GUI input selection is shown in Fig 6.



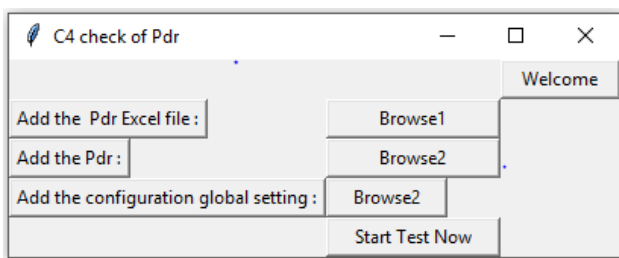
**C3 : IO Initialization test report**

if the compared value is NOK check the configuration value

RegName	before_IO_init	after_IO_init	from_IO_Config	Result
PM0	0xff	0xfe	0xfe	OK
PM1	0xff	0xc3	0xc3	OK
PM3	0xff	0xe4	0xe4	OK
PM4	0xff	0x75	0x75	OK
PM5	0xff	0x2	0x2	OK
PM6	0xff	0x6	0x6	OK
PM7	0xff	0x1	0x1	OK
PM8	0xff	0xef	0xef	OK
PM9	0xff	0xd8	0xd8	OK
PM12	0xff	0xbf	0xbf	OK
PM14	0xff	0xfe	0xfe	OK

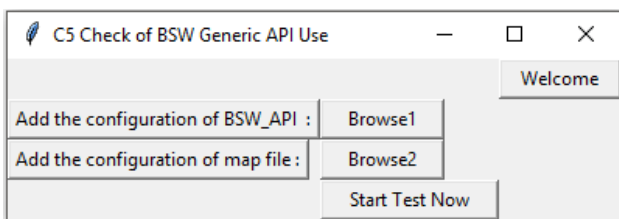
**Fig. 9. Output of C3 check of IO initialization.**

4.The script is being written for check of pdr which contains the EEPROM usage details. The generated output is in the html or excel format for readability. The results of C4 GUI input selection is shown in Fig 10.



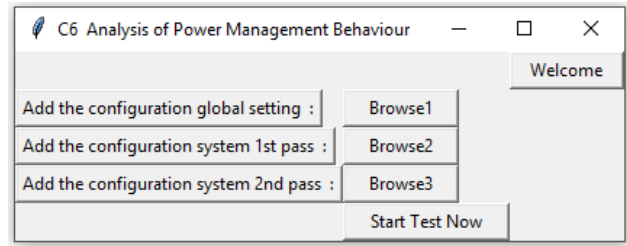
**Fig. 10. GUI of input selection for C4.**

5.Script is being written to check the API used for design and the description of the API is added. The results of C5 GUI input selection is shown in Fig 11.



**Fig. 11. GUI of input selection for C5.**

6.A separate script is being written for C6 check of analysis of power management. The results of C6 GUI input selection is shown in Fig 12.



**Fig. 12. GUI of input selection for C6.**

## V. CONCLUSION

In this paper, the validation of body controller ECU's is automated using python programming. The validation of ECU's has several test cases from C1-C6 for both AUTOSAR and NON-AUTOSAR. The work is carried for basic software. Test automation proposed in this paper increases the efficient verification by generating the test result report. The input for validation of ECU's includes configuration files(XML files), binary files(.s19 files), pdr files, .c files. Some test depends on hardware setup to check the initialization values (configuration of microcontroller pins) of ECU's for both AUTOSAR and NON-AUTOSAR. The result of an automation process overcomes the manual effort and reduces time for testing. The script is being written for six test cases using python packages i.e., lxml, xlwt, xlrd, tkinter, isystem connect and the outcome is generated in HTML and Excel.

## ACKNOWLEDGMENT

My sincere thanks to our Principal, HOD (Department of ECE) of JSS Science and Technological University for the support given to do the project in the company. I would like to thank my guide Mrs. Eshwari A Madappa for guidance and support provided. I would like to thank managers of Continental Automotive Components Private Limited Mr. Nectar Jinil and Mr. Sanjaya Dash for the encouragement and support provided. I thank the whole team for the support of technical guidance.

## REFERENCES

1. <https://www.isystem.com/downloads/isystem-connect-sdk.html>
2. <https://www.edn.com/electronics-blogs/embeddedbasics/4442206/Python-s-role-in-developing-real-time-embeddedsystems>
3. Rajesh Kumar and Vivek Kumar" Process Optimization for Testing of Domain Specific Languages in Industrial Automation" 2015
4. Jae-Hoon Lim and Jae-Jeong Kim" Build-up of customized test automation solutions for efficient verification of chassis systems in vehicle HILS" 2017 17th International Conference on Control, Automation and Systems (ICCAS 2017)
5. Karuturi Sneha and Malle Gowda M" Research on Software Testing Techniques and Software Automation Testing Tools", International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017)

## AUTHORS PROFILE



**Eshwari A Madappa** Assistant Professor at sri jayachamarajendra college of engineering Mysuru, Dept Electronics and Communication. JSS Science and Technology University. She has completed her Bachelor of Engineering in JSSATE Bangalore and Master of Technology in Power Electronics stream in RVCE Bangalore.



**Varsha P Y** MTech student at Sri jayachamarajendra college of engineering Mysuru in Automotive Electronics, Dept Electronics and Communication. JSS Science and Technology University. She has completed her Bachelor of Engineering in Dr Ambedkar institute of technology Bangalore



**Nectar Jinil** Senior manager at continental automotive component private Limited. He has completed his Bachelor of Engineering in Noorul Islam College of Engineering Chennai and Master of Technology in Anna university of Technology, Coimbatore.