Extensive Verification of Activation Timing of In-Vehicle Software

Akira KANAZAWA*, Ken FURUTO and Tatsuji MATSUMOTO

To keep vehicle embedded software quality high, products must be verified on every event timing including unexpected timing.Especially important is the timing of power supply switching. In general, in the time region just after power supply is switched on, the possibility of occurrence of software error is higher than in the steady state. This is because the operating conditions at the time of switch-on of a power supply are different from those in the steady state, such as higher communication load.Continuous efforts are being made to verify product quality thoroughly, but if software scale and complexity continue to increase, the verification work might be extremely larger in the future than in these days. The authors have developed a testing tool that allows the verification to be performed at high time resolution at the timing of power supply switching. And the authors have also implemented automatic testing functions to the tool. Thus, they have constructed efficiency verification environment. The authors have applied this "power switching timing test tool" in the verification of prototype ECU. About 17,000 test cases were prepared and tested about prototype ECU. Such testing is impossible without the "power switching timing test tool". As a result, we have demonstrated the tool to be effective.In addition, we developed the automatic judge function to the tool so that testing operation efficiency is also improved considerably.

1. Introduction

In recent years, functions such as door locking or room lamps of vehicles are becoming highly sophisticated in pursuit of usability and comfort even in compact cars as well as in luxury cars. As an in-vehicle electronic control unit (hereinafter referred to as "ECU") that controls these functions also becomes more sophisticated, the size of embedded software has increased 1,000-fold in the last 20 years as shown in **Fig. 1**. Meanwhile, the software is becoming even more complex, causing a concern for a potential quality loss due to insufficient design and verification.

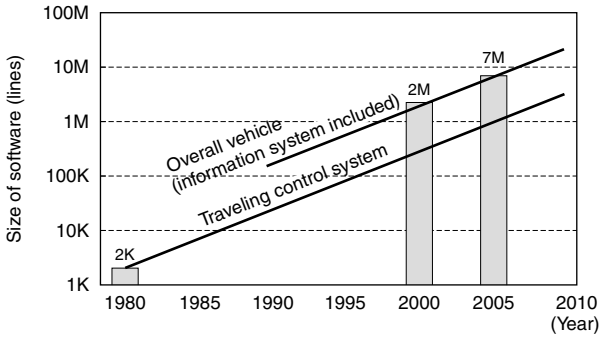


Fig. 1. Transition in size of in-vehicle ECU software[1]

In general, possible preventive measures against software problems would be as follows:

[1] Improving from a project management level by improving processes[2];
[2] Securing quality from upstream processes by improving design techniques;
[3] Performing technical reviews by members versed in software development;
[4] Conducting verification tests with an actual unit under as many operating conditions as possible;
[5] Using a simulation environment to conduct operation verification of timings that are difficult to verify with an actual unit[3].

This paper, with a focus on the method [4] mentioned above, reports a tool the authors developed which enables more efficient verification by using an actual unit under far more detailed operating conditions than before.

2. Purpose

In order to secure software quality, the basic method is to conduct sufficient reviews and verification at a design phase and eliminate problems completely. However, because design or verification of design involves a human intervention, it is unlikely to be able to remove 100 percent of errors. Therefore, it is required to actually operate the software to conduct tests to detect and correct lurking design problems. Before releasing products, it is necessary to eliminate the problems through the tests.

Indicated in **Fig. 2** is a typical V-shaped flow chart of software development procedure. Among the processes, the system test process becomes the focal point of verification with which the software is actually operated for verification. This paper introduces a tool the authors

developed, with paying a special attention to the system test process, which allows efficient verification of whether or not abnormal behavior occurs depending on the activation timing of in-vehicle ECU.

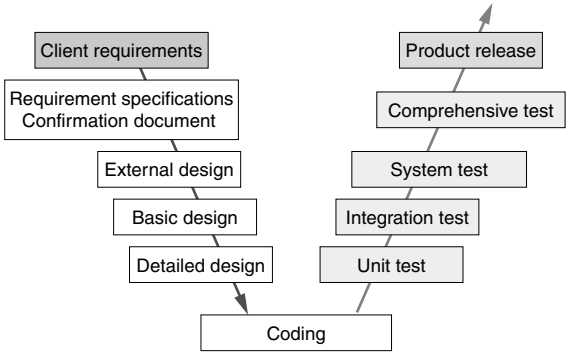


Fig. 2. V-shaped flow chart of software development procedure

Embedded software, such as the one used for in-vehicle ECU, works very differently depending on the occurrence timing of various input events such as switches, sensors, or communications. In order to secure software quality, an extensive verification is required to ensure that no abnormal behavior occurs even at unexpected, exceptional timings as well as with typical operation patterns and normal operation timings described in specifications. In particular, the timing of turning on the power of ECU where an initialization program and routine processing start to work concurrently is known from previous samples as well as from past experiences that an omission of timing design tends to lurk. This is due to the fact that operating conditions become different from a steady state by increased communication, etc. Traditionally, product quality has been secured through various efforts described in the Introduction. However, as the trend of larger size and more complex software proceeds further in the future, there is a concern that, although quality can still be secured, man-hours required for verification process may exponentially increase with the conventional evaluation method.

In response to the above, the authors have developed a tool with which a PC is used to control power activation timings of in-vehicle ECU at a finely divided time interval with a detailed time resolution, allowing efficient verification of the in-vehicle ECU behavior of when the control is performed.

3. Issues

First, the following is the summary of the currently adopted system testing methods and their issues.

To conduct a test on different power activation timings, the power source must be repeatedly turned ON/OFF to verify the operation.

However, because the activation test is performed manually, the following issues are present:

① Testing is difficult with which the activation timings at the accuracy level of 100 ms or less are targeted. Thus, in order to conduct the targeted timing test, the test may be required to be performed repeatedly for number of times.

② Recording of test results is also performed manually. Because the manual test operation and recording intensifies the load on workers, it is necessary to consider providing breaks and shift rotation for the workers. Consequently, a large number of man-hours will become necessary, resulting in an increase of development cost.

All these issues are caused due to dependency on manual work. Thus, to solve them, an automation tool must be introduced.

Previously, the authors used a commercially-available general-purpose test tool for automated testing. For example, tests were conducted on timings, such as the concurrent occurrence of multiple events that is likely to generate design-related problems. The authors have also considered applying commercially-available tools to control ON/OFF operation of a power source; however, due to limitations of tools, it was unable to conduct timing verifications with an accuracy of 500 µs or less. Although quality have been ensured through the manual cyclic activation tests thus far, a concern has arisen that man-hours required for verification process may increase exponentially; as the trend of larger and more complex software proceeds further in the future, the number of cyclic activation tests that are required to secure the quality also increases.

The authors have therefore decided to newly develop a special tool that enables the control of ON/OFF operation of a power source at 1-µs accuracy level to efficiently and extensively verify operations at power activation timings.

4. Principle

The configuration of the devices is shown in **Fig. 3**. Also, the overview of the specification is described in **Table 1**.

What the authors developed this time is a power distribution/input-output box and a control software for this power distribution/input-output box that works on PC.

DC power from an external source is supplied to an ECU through a power distribution/input-output box. The PC and the power distribution/input-output box

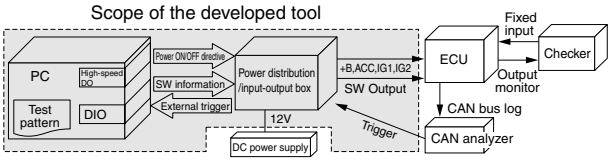


Fig. 3. Device configuration diagram

Table 1. Specification overview

- Possible to independently supply 16 types of power output (4 ECU categories x 4 power types (+B, ACC, IG1, IG2))
  +B: A power type that is always kept ON when connected to a battery.
  ACC, IG1 and IG2: Power types that are turned ON depending on the position of an ignition key.
- Possible to control timings with high accuracy.
  Time resolution: 1μs
  (However, the reproducing capability relies on the relay capability of power distribution/input-output box.)
- The current capacity per output is 40 A.
- Test patterns can be easily created using a general-purpose text editor.
- TTL can be externally output. (Input to ECU via port is available.)
- Linked start function by external trigger is available.

are connected via a high-speed digital output I/F board (high-speed DO in the figure) and a digital input-output I/F board (DIO in the figure). The ON/OFF operation of a relay in the power distribution/input-output box is controlled by the software operating on PC so that ON/OFF of the DC power supplied to the ECU can be controlled.

Using a high-speed digital output I/F board for outputting of power source signals from PC enables achieving a highly accurate time resolution of 1 μs as a specification of the software. However, including the reproduction test, the overall capacity of tool depends on the accuracy of the relay mounted in the power distribution/input-output box. Although the difference in individual response capabilities of relay can be corrected by the software, a high-speed relay needs to be equipped in order to enhance the overall response capability of the tool.

Moreover, the power distribution/input-output box is equipped with a function to externally output TTL-level signals, so the software can also control the output of TTL-level signals just like ON/OFF of the power source. This function also enables outputting the ON/OFF status of a switch to ECU and others. For example, this can be applied to a test case where the statuses of switch and power source are changed simultaneously.

Also, a trigger to start a test can be input from outside to the power distribution/input-output box. The control software can recognize the external trigger and start to control ON/OFF operation of a power or switch in coordination with the trigger.

In an example of device configuration indicated in **Fig. 3**, a CAN bus signal output from the ECU is input to a bus data analyzer (CAN analyzer in the figure), and when a specific frame is received, a trigger will be output to the power distribution/input-output box. The CAN (acronym for Controller Area Network) is a serial communication protocol developed for being mounted on vehicles.

In coordination with this trigger, ON/OFF operation of four types of power sources, +B (power type that is always kept ON when connected to a battery) and ACC, IG1, IG2 (power types that are turned ON depending on the position of an ignition key), which are actual power systems in a vehicle, will be controlled while ON/OFF signals of a switch are controlled.

5. How to Use

The procedure for using this tool is as follows:
① Prepare test patterns.
② Input the test patterns into the tool and configure them.
③ Conduct a test using the tool.

In test patterns, each line shall contain one command. For command descriptions, a parameter shall come after the command with a comma delimiting them.

The test patterns are in a highly versatile text format. The description format of the test patterns is shown in **Table 2**.

Table 2. Test pattern description format

Command	Parameter	Description example
OutReq OutPut	Bit signal name, Relative output time, Output status	OutReq, ACC-1, 00000100ms, ON
WaitTime	Relative output time	WaitTime, 15s
WaitTrig	Bit signal name	WaitTrig, +B-1
Nop ExecOut	(None)	ExecOut

For example, in order to verify the ECU behavior of when the ACC power supply ON timing is delayed than the point where 500 ms have elapsed since the +B power supply ON timing, a test as shown in **Fig. 4** will be conducted. A test pattern example is described in **Fig. 5**. This example shows a test case where ACC power activation is delayed by 1 ms each.

Because the test patterns are written in a text format, the patterns can be edited using a general-purpose text editor. However, if test patterns, with which timings are varied extensively as shown in **Fig. 4**, are created manually, the efficiency will be affected. In response to the above, the authors have developed a separate tool that automatically creates test patterns to improve work efficiency. Concerning this tool, detailed descriptions shall be omitted due to limitations of space.

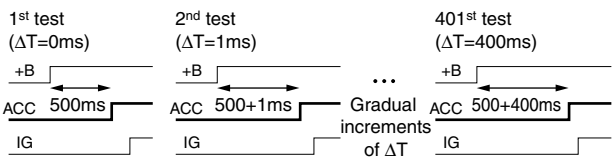


Fig. 4. Test example with delayed ACC power activation

```
'1st test (Δt=0ms)
OutReq, +B-1,              0ms, ON
OutReq, ACC-1,           500ms, ON
OutReq, IG-1,           1000ms, ON
ExecOut
                :
WaitTime,                    5s
'
'2nd test (Δt=1ms)
OutReq, +B-1,              0ms, ON
OutReq, ACC-1,           501ms, ON
OutReq, IG-1,            999ms, ON
ExecOut
                :
                :
WaitTime,                    5s
'
'401st test (Δt=400ms)
OutReq, +B-1,              0ms, ON
OutReq, ACC-1,           900ms, ON
OutReq, IG-1,            600ms, ON
ExecOut
                :
```

Fig. 5. Test pattern example

When this tool is started, and created test patterns are loaded, a screen shown in **Fig. 6** appears. Because the power's ON/OFF status is represented in a wave form for each signal, a user can immediately recognize whether or not the power ON/OFF timing for the created test patterns are suitable for achieving the objectives.

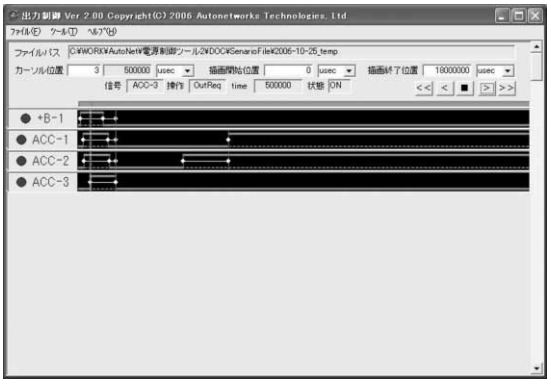


Fig. 6. Tool execution screen

One of the applied usages of this tool would be to connect to an actual vehicle test bench. An example of the connected actual vehicle test bench is indicated in **Fig. 7**.

Under the same environment as that of real car, the operations to turn the power ON/OFF can be performed at various timings for multiple ECUs. This will allow users to verify, while being able to imagine the vehicle behavior, the operations of a developed ECU which vary depending on the timing of power activation.

Fig. 7. Example of connected actual vehicle test bench

6. Verification of Effectiveness

In order to verify the effectiveness of this tool, the authors have conducted an evaluation by actually applying it to a prototype ECU. The configuration diagram is shown in **Fig. 8**. The prototype ECU is a system that has a function to interactively relay communication data between two CAN buses.

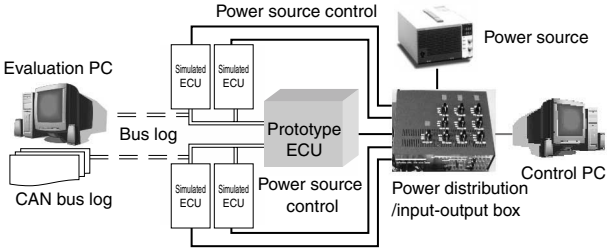


Fig. 8. Example of connection to prototype ECU

The two CAN buses of prototype ECU are connected with two ECUs respectively, four in total, that simulate other in-vehicle ECUs (hereinafter referred to as simulated ECUs). The power source's ON/OFF of prototype ECU and of four "simulated ECUs" connected to prototype ECU, five units in total, is controlled by a control PC via the power distribution/input-output box. An evaluation PC (equipped with a function to analyze CAN bus data) collects data from the two CAN buses, records them in a log, and then analyzes this log to verify the behavior of prototype ECU.

Two units of simulated ECUs were connected to the two CAN buses, respectively, that are connected with prototype ECU. Then, the authors verified that all cases work normally even when the power activation timing of each simulated ECU is variously changed based on the test specification of prototype ECU.

In specific terms, the authors conducted evaluations on scenarios of 16 patterns in total, including different patterns based on 12 kinds of activation

sequences, by changing timings at an interval of 1 ms. The number of scenarios created totaled 11,020. Furthermore, the authors conducted additional evaluations using 6,400 scenarios of activation timings at an interval of 0.1 ms; with these scenarios, the activation timing range is restricted to 20-ms time duration before and after the on-time timing defined in the specification. With these 11,020 test scenarios at 1-ms level and 6,400 test scenarios at 0.1-ms level, totaling the number of test scenarios to approximately 17,000, the authors conducted operation verification tests.

Regarding the execution result, the data of two CAN busses were collected using the evaluation PC, and recorded in a log. Then, the authors checked the result by analyzing the data and comparing it with the specification.

For analyzing the log, the authors developed a log comparison tool as shown in **Fig. 9**. The tool was designed to detect differences from the design specification by automatically comparing the execution results (CAN data) with reference pattern data prepared beforehand (data output timing patterns strictly reflecting the original design intention).

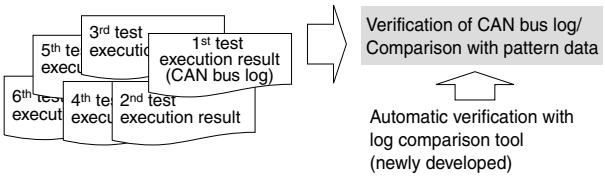


Fig. 9. Automatic verification using log comparison tool

7. Consideration

For the application to the operation verification of prototype ECU, the authors have created approximately 17,000 extensive test patterns, and attempted to improve efficiency by automating the operation verification of power activation timings.

If the operation is to be performed manually, the required man-hours for the operation can be calculated as follows. Here, it shall be assumed that two workers share the duties of power supply switch operation as well as of the check, verification, and recording of test results. Assuming that 30 seconds are required for one test pattern, approximately 140 hours are necessary to execute 17,000 test patterns. Since two workers are performing the operation, approximately 280 man-hours are needed in total.

When this tool is used, it takes approximately 4 hours for the creation of test scenarios, approximately 49 hours for the performing of tests, and approximately 20 hours for the evaluation of results. Among the above time allocation, the tests are conducted automatically by the tool; therefore, the required man-hours is 24 man-hours in total.

From the above, a reduction of 256 (280 minus 24) man-hours, i.e., 32 man-days, becomes possible compared with the conventional method.

Furthermore, potential applied usages of this tool would be as follows:

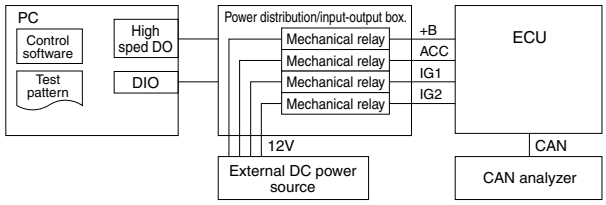
① Instantaneous interruption test

② Bus short-circuit

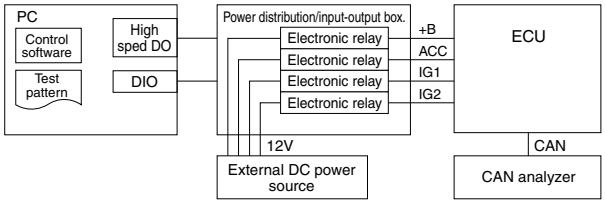
An instantaneous interruption indicates a temporary OFF status of ECU's power source, which could be caused by poorly connected electrical wires in a vehicle environment. When an instantaneous interruption occurs, an ECU must perform a predetermined abnormal-condition handling process. For the purpose of verifying whether or not this abnormal-condition handling process is appropriately performed, an instantaneous interruption test is conducted.

In the instantaneous interruption test, it only needs to create test data that turns the power OFF instantaneously (10 μs, for example). However, the tool developed this time adopts a mechanical relay for a relay in a power distribution/input-output box as shown in **Fig. 10 (a)**. Consequently, if the relay's following capability is not appropriate, the relay may not react to an instantaneous turning OFF of power, resulting in a non-occurrence of instantaneous interruption. Also, when the instantaneous interruption test is used for the cyclic testing, the relay contact may melt, causing a trouble to occur to the equipment. Therefore, a mechanical relay is not suitable for the instantaneous interruption test.

Hence, by replacing the relays in the power distribution/input-output box with electronic relays (FET, etc.) as shown in **Fig. 10 (b)**, and then by increasing the accuracy level of timings to handle repeated ON/OFF operation at high speed, it becomes possible to conduct tests including the instantaneous interruption test.



(a) Current configuration



(b) Configuration after improvement (example)

Fig. 10. Pre- and post-improvement configurations (example)

A bus short-circuit indicates an abnormal condition where a communication line comes in contact with a power line or ground wire, causing communication failure. Also for this problem, this equipment can be applied to generate a bus short-circuit from a PC at a designated timing, which can possibly contribute to extensive verification of operating conditions.

8. Conclusion

This time, the authors have successfully developed the software that enables timing verification to be conducted at the accuracy level of 1 μs instead of 500 μs, previously possible maximum capacity, and the hardware to support this software.

In conducting a verification using a prototype ECU, the authors have attempted to further improve the efficiency by using an automatic generation tool of test data as well as automatic analyzing tool of test results, thereby demonstrating the effectiveness of these tools.

Their potential applications in the future would include verification of abnormal-condition handling process such as for instantaneous interruptions or bus short-circuits. The authors strive to continuously contribute more than ever to the development of efficient, high-quality software for in-vehicle ECU by continuously developing wide-ranging test tools.

References

(1) "Improving the efficiency of in-vehicle software development" by Keisuke Ogawa, November 2007 Issue of Nikkei Automotive Technology, pp82-97

(2) "Construction of CMM Level 3 Software Development Process for Automotive ECUs" by Satoshi Terakubo et al., The 166th Issue of SEI Technical Review, pp45-50

(3) "Development of Cross-Development Environment Tool for Automotive ECU Software" by Tatsuji Matsumoto et al., The 165th Issue of SEI Technical Review, pp10-14

Contributors (The lead author is indicated by an asterisk (*)).

A. KANAZAWA*
- Software Development Center, AutoNetworks Technologies, Ltd.

K. FURUTO
- Manager, Software Development Center, AutoNetworks Technologies, Ltd.

T. MATSUMOTO
- General Manager, Software Development Center, AutoNetworks Technologies, Ltd.