

Intelligent ECU End of Line Testing to Support ISO26262 Functional Safety Requirements

Jin Seo Park
Infineon Technologies Korea Co. Ltd.

Il hong Suh
Hanyang Univ.

Chang Yul Choe and Minho Ro
Hyundai KEFICO

Simon P. Brewerton
Infineon Technologies UK Ltd

ABSTRACT

The recent adoption of the ISO26262 Functional Safety Standard has lead to the need for a greater degree of rigor in the technical, organizational and process aspects of electronic ECU engineering. One new facet of this standard also covers (in part 9.7) the analysis of dependent failures at manufacturing time, not only the microcontroller, but also for the plethora of connected system ASICs, input circuits, output drivers and communication devices in the PCB of the ECU. This paper will describe the CAN based end of line ECU self test system that was implemented at a major tier 1 supplier to address the issues of efficiently gaining a high degree of diagnostic coverage of single point faults and latent faults in highly integrated automotive ECUs. It will outline the requirements and demonstrate how a flexible system solution allows not only efficient test of the new design but also addresses the degree of rigor required to be demonstrated when testing a production volume ECU to ISO26262 functional safety standard.

CITATION: Park, J., Suh, I., Choe, C., Ro, M. et al., "Intelligent ECU End of Line Testing to Support ISO26262 Functional Safety Requirements," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 6(1):2013, doi:10.4271/2013-01-0403.

MOTIVATION AND ADVANTAGES OF ISO26262 BASED END OF LINE TESTING

Since the introduction of the ISO26262 automotive functional safety standard, there has been much closer focus on the implementation and verification of functions that support functional safety in ECUs. However the ISO26262 also mandates a single point fault metric and a multiple latent fault metric at different levels for different ASILs. It also defines processes for engineering electronic systems to support specific safety goals in a requirements based engineering development flow. In the context, end of line programming and testing is mentioned in Part 9, Section 9.7 - Analysis of Dependent Failures. Specifically, in part 9.7.4.4 there is a list of topics that are applicable:

- Random Hardware Failures (Common blocks such as clocks, test logic, power supplies, MCUs and ASICs)
- Development Faults (faulty requirements, design faults, implementation faults, faults due to new technologies, and faults from modifications)
- Manufacturing faults (faults due to process, procedures, training, faults in control plans and monitoring of special characteristics, faults relating to software flashing and end-of-line programming)
- Installation Faults (wire harness connections, interchangeability of parts, failure of adjacent items or elements)
- Repair Faults (processes, procedure, training, troubleshooting, and substitute parts)

- Environmental Factors (Temperature, vibration, humidity, pollution, corrosion, contamination, EMC, ESD)
- Failures of Common External Resources (Power connections, input data, buses and communications)
- Stresses due to Specific Situations (wear, aging)

So from this list it can be seen that end of line testing and programming now fall within the scope of the ISO26262, so a degree of rigor must be applied to the engineering processes and architectures of the testing methodology.

There are normally several specific tests required at the end-of-line to verify that the final ECU is capable of reaching specific safety goals derived from the safety analysis performed at a system level for the ECU functions in the context of their intended operation. These tests are to verify and monitor 'Special Characteristics'. These can be any measurable aspect of the ECU operation, but typically include such things as clock accuracy, voltage regulator accuracy, ADC accuracy and processor core operation.

TECHNIQUES TO DETECT END OF LINE FAILURES

The application of end-of-line tests provides the final verification and sign-off that a specific ECU is functioning exactly as required. This implies that all critical aspects can be (and are) tested. There are 2 different opportunities for End of Line testing; once at the ECU manufacturing plant and once at the final vehicle assembly plant. The ECU manufacturing plant end-of-line test environment permits many varied stimulus to be generated and measurements to be taken simultaneously, however the limiting factor is test time. The longer a test runs the better the expected diagnostic coverage may be. This is typically a law of diminishing returns, where 80% coverage can be obtained by 20% of the test vectors, but the other 20% of coverage cannot be achieved even with a further 80% more test vectors (equivalent to test time). The test time is broken up into 4 distinct periods:

- Load Device Under Test (DUT) - install the ECU on the test fixture
- Download the basic software and flash the program memory
- Download and execute test patterns to check ECU functions
- Program calibrations
- Label then remove the passing DUT ready for next test cycle

Each operation has to be as efficient as possible as test time is often the bottleneck in the manufacturing process. EOL testers are very expensive pieces of equipment to purchase and maintain, so full utilization and capacity must be carefully matched.

The End of Line test at the vehicle manufacturer is much more limited in scope due to the restricted stimulation possibilities to fully exercise the ECU functionality. The ISO 26262 however demands that the system be proved safe by being free from manufacturing defects. These defects can be due to incorrect assembly, missing harness connectors, wrong ECU version in the wrong vehicle, wrong software loaded or a multitude of other potential issues. The ISO 26262 requires that all the possible failures of assembly process are examined in the FMEDA and that measures are put in place to detect or prevent such occurrences. There is very little time on the vehicle build line to perform much but the specific tests required for the detection of correct assembly and connection of all essential elements such as sensors, power supplies and actuators. In high integrity systems (ASILC and ASILD) the ECUs application software already provides the capability to detect latent failures by the start-up tests when the system boots, so these tests do not need to be repeated by the End of Line programming station, but a final check must be made of vehicle diagnostic trouble codes to ensure that no detected and logged faults are present when the vehicle assembly is completed.

TYPICAL TEST SETUP

To illustrate the issues associated with end-of-line testing, a real production system has been used as an example. The application outlined in this paper is a generic test architecture, capable of programming and testing a family of powertrain and transmission control ECUs based on the Infineon 32bit TriCore microcontroller (TC1782). Although some specific implementations are outlined, these only represent one possible approach to the problem, and there are many other possible solutions that are also valid for different architectures.

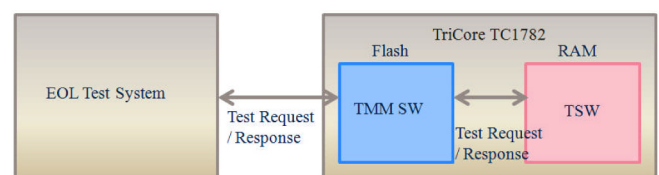


Figure 1. Block Diagram of the basic EOL Test Architecture

The end-of-line test architecture is comprised of 3 major parts:

End-Of-Line Test System (EOL TS): The EOL TS is the independent tester device which provides the electrical conditions and the communication commands to test the connected Electronic Control Unit (ECU) hardware. The EOL TS can provide stimuli as communication bus messages, PWM triggers, and analog voltages, and in parallel can log bus messages, voltages, currents and digital voltages (PWM). The ECU is typically connected via the main product

connector, so special software is required else only normal functionality would be possible.

Test Management Mode Software (TMM SW): The TMM SW is the part of production release Start Boot software which resides in the internal flash memory of the production ECU. When ECU system boots directly after reset, the TMM SW checks test management mode entry conditions. If the correct sets of conditions are met, then the ECU goes to test management mode ready to begin the test sequence, else it boots normally.

Test Software (TSW): The TSW is the group of specific test routines which execute a test case for each functional block of the ECU hardware. If Test Management Mode is successfully entered after reset, then the EOL TS will download the required TSW file to perform the demanded test. The TSW is a binary file format image of the test program, which is compiled and linked to execute from the internal SRAM memory of the microcontroller. Whenever EOL TS requests a test, a section of the TSW runs. Depending on the requested command from EOL TS, the requested test case in TSW runs and the test result is sent back for test reporting.

TEST FLOW

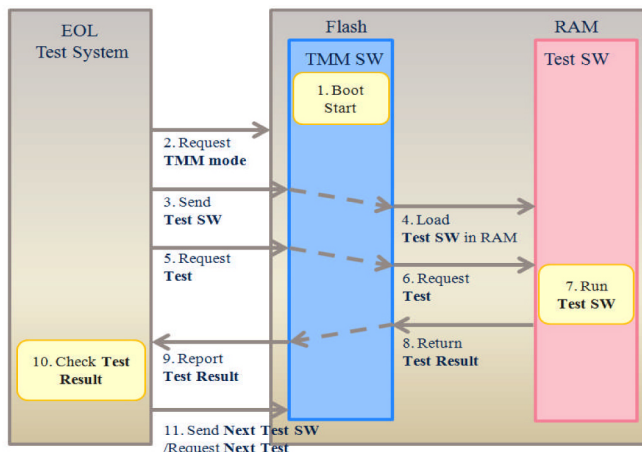


Figure 2. Diagram of the Test Flow between the EOL TS and the ECU microcontroller.

The sequence of performing an initial test is outlined in Figure 2. This consists of the following steps:

1. **Boot Start:** ECU is powered on and Reset is released so that the Microcontroller starts booting from Flash.
2. **Request TMM mode:** The EOL TS requests TMM mode to the ECU by generating a specific set of signal conditions on the product connector directly after reset. The Boot software measures the inputs and decides to boot into TMM mode or perform an application (normal mission mode) boot. If conditions are met it enters the TMM mode.

3. On entering TMM mode the ECU sends an acknowledgement message to the EOL TS via CAN.
4. Upon receipt of the acknowledgement, the EOL TS sends the TSW to the ECU SRAM (also via CAN). Once downloaded, the ECU calculates a checksum and sends the checksum back to the EOL TS.
5. EOL TS checks the validity of the download and can log an error if the download fails. Else the EOL TS sends a test request command.
6. The ECU receives the test request and
7. Executes the specific test (under control of the EOL TS).
8. The ECU calculates the test results and
9. Returns the test results (via CAN) to the EOL TS.
10. The EOL TS checks the results (pass / fail) and decides which is the next test to run or if the test is complete.
11. If the test is complete then the ECU is depowered and removed, else the next test package is downloaded and the cycle repeats from step 4

TEST PATTERN STRUCTURE

Memory

The TSW is used only for EOL testing, so it is not necessary or desirable to store the TSW in Flash. When a test is performed the TSW is downloaded and executed from internal SRAM in the TriCore Program Memory Interface (PMI) with the parameters and variable held in the SRAM of the Data Memory Interface (DMI). If the TSW were to be stored and executed from Program Memory Unit (PMU) flash, then the test vectors would occupy flash memory in production devices, which has several drawbacks:

- The flash storage increases the cost of every device
- The test vector might accidentally get executed in case of a fault in the microcontroller and therefore violate a safety goal (with the implication that the test code might be safety critical and therefore expensive to build)
- The test code executes slower from flash than from SRAM

It is therefore strongly recommended that any test code is only ever downloaded and executed and that the mechanisms used to enter the special test mode use very contrived entry conditions that can never exist in the real application environment.