Virtual based Automated Testing for automotive Body ECUs

QTronic User Conference 2019
Virtual ECUs and Applications
2nd of December, Berlin, Germany

Hiroshi Ueda
Ubiquitous AI Corporation
SPQA Division

Kazumasa Matoba
AISIN SEIKI Co., Ltd.
Software Engineering Dept.
Agenda

1. Introduction
2. Background
3. HILS by Silver
4. Python Scripts Management
5. Summary
Agenda

1. Introduction
2. Background
3. HILS by Silver
4. Python Scripts Management
5. Summary
1-1. AISIN Group Main Products

**ICT & Electronics**
- Sensor
- Car navigation system
- ECU

**Powertrain**
- Automatic transmission
- CVT
- Electric water pump

**Body**
- Power sliding door
- Sunroof
- Power seat

**Vehicle Safety System**
- Parking assist system
- Power tilt & telescopic
- Electric parking brake
1-2. Features of the Body System

Features of the body system
- There is a direct operation request from the user.
- The number of actuators to be driven is relatively small. (1 ~ 3 pieces)
- The requirement for responsiveness is low
Agenda

1. Introduction
2. Background
3. HILS by Silver
4. Python Scripts Management
5. Summary
2-1. Testing Embedded Software

- Increase in test cases due to the development of larger-scale in-vehicle software
- Increased demand for simulation testing
  (Reduced hardware and other provisioning costs
  /easy and replicable test environments
  /to advance the testing schedule)

Requirements analysis

Architecture design

Unit design

Unit test

Coding

Integration test

System test

Software Development V-Model
1. Silver was introduced as a tool for simulation of product ECU, and SILS environment was constructed.

2. TestWeaver Light was introduced to automatically generate reports and automatically judge for simulation results.

2 Test Automation

SILS

1 Control Loop

Product Software dll

Plant Model dll

Python Module dll

Test Code py

Watch

Product Software c

Plant Model slx

Test Instrument py, rml

Test Report html
- SILS enables efficient testing on PCs
- Increasing the ratio of SILS testing can improve the efficiency of the entire test process.

BUT... In the development of embedded software, we can not eliminate tests requiring verification of actual machines.

ex.) Interrupt/Timing/Processing Load

- > HILS capable of efficient automatic testing in actual machines is required
2-4. Issue in Test Assets Management

Good management practices for test assets are needed to keep testing costs lower.

Management Cost

There is a big difference in maintenance costs depending on the management method.
Agenda

1. Introduction
2. Background
3. HILS by Silver
4. Python Scripts Management
5. Summary
HILLS can do automated testing with scripts.

Plant models can be easily fail-safe tested.

In-house HILLS already exists, but there are issues.
- Must use low-level in-house language for test scenario creation.
- Language specifications are not maintained and operated.
3-2. HILS Details

**Benefits**
- Common test interface with SILS
- Easily run Simulink models in real time
- Reduce in-house software components
- Can use Python as a scripting language

**Implementation Function**
- Digital In/Out
- Analog Out
- CAN Communication
- LIN Communication
3-3. Benefit: Common Test IF with SILS

Common Test IF with SILS = Aim to reduce costs through reuse
- Keep the environment (Tools, scripts, and project-related files) used in the SILS tests
3-4. Easily run Simulink models in RT

Flexible support for MBD by designing the HILS Plant Model Interface so that it can be imported from Simulink Model. Model development is performed externally using Simulink, and by providing the model, real time operation can be easily performed.
3-5. Other Benefits

<Reduce in-house software components>
- Cultivate successors through in-house software refactoring.
- Smaller in-house software components

<table>
<thead>
<tr>
<th>Component</th>
<th>Conventional</th>
<th>Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.PC</td>
<td>Outside</td>
<td>Outside</td>
</tr>
<tr>
<td>2.Test Scripts</td>
<td>In-house</td>
<td>In-house</td>
</tr>
<tr>
<td>3.GUI</td>
<td>In-house</td>
<td>Outside</td>
</tr>
<tr>
<td>4.Interpreter</td>
<td>In-house</td>
<td>Outside</td>
</tr>
<tr>
<td>5.Plant Model</td>
<td>In-house</td>
<td>Internal</td>
</tr>
<tr>
<td>6.Controller</td>
<td>In-house</td>
<td>In-house(new)</td>
</tr>
<tr>
<td>7.Device</td>
<td>Outside</td>
<td>Outside(used)</td>
</tr>
<tr>
<td>8.I/F Box</td>
<td>In-house</td>
<td>In-house(used)</td>
</tr>
<tr>
<td>9 Relay Board</td>
<td>In-house</td>
<td>In-house</td>
</tr>
<tr>
<td>10.Target ECU</td>
<td>In-house</td>
<td>In-house</td>
</tr>
</tbody>
</table>

<Can use Python as a scripting language>
- Accelerate script coding by moving from the original low-level to the widely used high-level languages
- Advantages in learning costs, external resources, information, and serviceability (by means of structure)
Agenda

1. Introduction
2. Background
3. HILS by Silver
4. Python Scripts Management
5. Summary
4-1. Test Asset

Maintenance Cost (intangible)

- Test Scenario
- Test Specification
- Test Configuration
- Plant Model
- Other

Thousands more!!
4-2. Test Framework

Advantages of the framework
- Reduce code volume
- Prevent unintended variations
- Fewer bugs

Before
Python module
Control
Coding
API

After
Python module
Framework
Control
Coding
API

Maintenance Target

Expected effect

50% more efficient

New
Reuse
Costs
5-1. Test Script Architecture Design

Design Test Script Architecture by Python for

The Python module API Interface

The Python module API

Simulation environment Interface

for each product

for each function

for each scenario

(SUT Interface)
Agenda

1. Introduction
2. Background
3. HILS by Silver
4. Python Scripts Management
5. Summary
7. Summary

- Develop HILS System by Silver
  Tests using real ECU are now possible in terms of test such as timing that cannot be confirmed by SILS

- Framework design for Python module of Silver
  Can create and maintain test scenarios with minimal cost, even with a large number of tests
Thank you for your attention!!