MATLAB EXPO 2019

Simulink 기반 Legacy C/C++ Code 통합, 결과 시각화 및 검증 방안

유성재
Model-Based Design

Systematic use of models throughout the development process
Legacy Code? Hand-written code?

Hand Code

Internal Libraries

Verified Components

Vendor Libraries

Device Drivers

Components
Under Development
MBD with Legacy Code!

Textual Requirements
Executable Specification
Model used for production code generation
Generated C/C++ code
Object code

Requirements Authoring
Modelling
Code Generation
Compilation and Linking

Software architecture and unit design
S-functions from Handed C/C++ code

Coverage analysis / Test case generation
Code review and Static analysis

Unit(Integration) testing

For Model-Based Design
For legacy code development

MATLAB EXPO 2019
Challenge?
Agenda

- Legacy code integration using Simulink
- Visualization using Simulink
- Verification with legacy code
How to Import Legacy Code

- Legacy Code Tool
- Legacy code integration in Stateflow
- C Caller Block
What legacy C code integration in Simulink means?

- Legacy Code Tool enables existing C code to be used in Simulink models
How to use Legacy Code Tool?

- General procedure for using Legacy Code Tool

1. Initialize LCT data structure
2. Populate LCT data structure
3. Generate S-function source file
4. Compile S-function source file
5. Create masked S-Function block

Modeling with S-Function

S-function block
Maintenance Problem…

Legacy code

Wrapper code

Modeling

If any changes in code

Script file

MATLAB EXPO 2019
Introducing C Caller Block

C Caller Block makes it easier to call C Functions in Simulink → It works for simulation and Code Generation
Key Features

- Automate the process

- Synchronize with custom code changes

  - Tedious
  - Error prone
  - Hard to maintain

Define Block Interface → Build Simulation MEX → Write Codegen TLC

Automate

C/C++ Code

MATLAB EXPO 2019
Demo: Simple C Caller
Demo: Integrate C code with Simulink Model
C Caller workflow

1. Add "c call" annotation
2. Insert custom C code in generated header file
3. Specify function name and parameters
4. Set function name in Port specification
Agenda

- Legacy code integration using Simulink
- Visualization using Simulink
- Verification with legacy code
Visualize Simulation Data
Visualize Simulation Data
Visualize Simulation Data

1. Component Implementing a Limit
2. Configuration Parameters: MultiinstanceModelExample/Configuration (Active)
3. Help
Tune and Visualize Your Model with Dashboard Blocks

- Custom Gage
- Flight Instrument Gauges - Aerospace Blockset -
Tune and Visualize Your Model with Dashboard Blocks

Inputs

- Engine Speed (rpm)
  - 300
  - 700

Fault Injection

- Throttle Angle
  - NORMAL
  - FAIL
- Engine Speed
  - NORMAL
  - FAIL
- EGO
  - NORMAL
  - FAIL
- MAP
  - NORMAL
  - FAIL

Fuel (g/s)

Air/Fuel Ratio

Normal Range

Full Range
DEMO : Test harness creation with Dashboard
DEMO : Visualize Input and Output data
Visualization Summary
Agenda

- Legacy code integration using Simulink
- Visualization using Simulink
- Verification with legacy code
Test Case generation

Existing test coverage

Test Cases

Design Logic

Test case generation

100% coverage
Systematic Functional Testing with Simulink Test

Test Case

Inputs
- MAT file (input)
- Group 1: Signal 1
- Signal Builder
- Test Sequence
- and more!

Main Model

Test Harness

Assessments
- MAT file (baseline)
- MATLAB Unit Test
- Test Assessment
- and more!
DEMO: Code Coverage with C caller
DEMO : Coverage Report

Tests

<table>
<thead>
<tr>
<th>Test#</th>
<th>Started execution</th>
<th>Ended execution</th>
</tr>
</thead>
</table>

Summary

<table>
<thead>
<tr>
<th>File Contents/Complexity</th>
<th>Decision</th>
<th>Condition</th>
<th>MCDC</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>RejectDoublePress.c</td>
<td>2.67%</td>
<td>67%</td>
<td>17%</td>
<td>86%</td>
</tr>
<tr>
<td>RejectDoublePress.c</td>
<td>2.67%</td>
<td>67%</td>
<td>17%</td>
<td>86%</td>
</tr>
</tbody>
</table>

Details
DEMO : Test Case Generation

Coverage Options

- Decision : on
- Condition : on
- MCDC : on
- MCDC mode : masking
- Relational Boundary : off

Tests

<table>
<thead>
<tr>
<th>Test#</th>
<th>Started execution</th>
<th>Ended execution</th>
</tr>
</thead>
</table>

Summary

<table>
<thead>
<tr>
<th>File Contents/Complexity</th>
<th>Decision</th>
<th>Condition</th>
<th>MCDC</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>RejectDoublePress.c</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Key takeaways

Bring C/C++ code into Simulink

- With full integration into Model-Based Design workflows
- To visualize simulation result
- Enable unified verification environment
MATLAB EXPO 2019

데모 부스와 상담부스로 질문 하시기 바랍니다.

감사합니다