Unit Testing with VectorCAST and AUTOSAR

Vector TechDay – Software Testing with VectorCAST
Agenda

- **Introduction**
  - Unit Testing Demo
  - Working with AUTOSAR Generated Code
  - Unit Testing AUTOSAR SWCs
  - Conclusions
Introduction

Types of Testing

**Unit Testing**
Individual units or modules are tested. It involves testing of source code by developers.

**Integration Testing**
Individual modules are grouped together and tested. The purpose is to determine that modules are working as expected once they are integrated.

**System Testing**
Testing is performed on the whole system by checking whether the system or application meets the requirement specification document.
Introduction

How Does Unit Testing Work?

Source Code

Test Harness

- Test Driver
  - Unit(s) Under Test
  - Stubbed Dependents
  - Real Dependents (.obj files)

- Test Harness
  - Unit(s) Under Test
  - Stubbed Dependents
  - Real Dependents (.obj files)
Introduction

What is VectorCAST?
Introduction

How Does VectorCAST Help You?

**Time to Test**
- **Weeks** → **Minutes**

**Frequency of Test**
- **Per Release** → **Per Change**

**Who Runs Tests**
- **QA** → **Everyone**

**Level of Automation**
- **Manual** → **Automatic**
How Does VectorCAST Enable Unit Testing?

- Creates an automated and repeatable process for developers
- Automatically generates your Test Harness including Smart Stubs
- Provides an intuitive user interface for generating test cases
- Contains a Command Line Interface (CLI) for headless test execution
- Supports C, C++, and Ada
Introduction

VectorCAST Enables Multi-Target Testing?

**Source Code**
- parse raw source code
- auto generate all drivers and stubs

**Host Environment**
- VectorCAST RSP
  - Test Harness
  - Tests
- Test Reports
  - PASS
  - FAIL
  - PASS
  - 100% Coverage

**Target/Simulator**
- Execute Tests
- Pass/Fail Results and Code Coverage

Ethernet, Serial Link, JTAG
Introduction

VectorCAST Supports Your Embedded Tool Chain

- Push button on-target and simulator test execution
- Built-in support for more than 30 compiler families
  - Hundreds of chip and runtime combinations
- Full-featured integrations with:
  - Cross Compiler Language Extensions
  - Embedded Debuggers
  - JTAG Probes
  - Real-Time Operating Systems

What do all these companies have in common?

None of these companies offer a test automation platform.

VectorCAST provides test automation to all of these environments.
What is AUTOSAR?

AUTOSAR (AUTomotive Open System ARchitecture) is a worldwide development partnership of vehicle manufacturers, suppliers, service providers and companies from the automotive electronics, semiconductor and software industry.

Learn More About AUTOSAR

https://www.autosar.org/
Introduction

Automotive Trends

Cloud / Backend

- support of high performance processors
- high bandwidth
- service based architectures
- open source, agile development
- dynamic and updatable
- internet

Embedded Systems

- safe
- secure
- embedded integration and debugging
- automotive supply chain
- automotive communication protocols
- automotive diagnostics

AUTOSAR Classic
Adaptive – best of two worlds

**Cloud / Backend**
- system and mobility strategies
- deep learning

**On board Supercomputers**
- multipurpose computing servers
- connectivity, gateways, HMI
- automated driving
  - mastered by OEM

**AUTOSAR Adaptive**
- safe
- secure
- embedded integration and debugging
- automotive supply chain
- automotive communication protocols
- automotive diagnostics

**AUTOSAR Classic**
- high bandwidth
- service based architectures
- open source, agile development
- dynamic and updatable
- internet
Introduction

Being Prepared for the Next-Generation of ECUs

Adaptive MICROsAR is a complete basic software solution up to ASIL D

Seamless interoperability with classic AUTOSAR ECUs

Additional, high performance ECUs hosting applications for upcoming use cases

Applications installed and started during runtime

Development of applications in the ecosystem of POSIX-based OS (Linux, PikeOS, QNX, Integrity, ...)
Introduction

AUTOSAR Product Comparison

AUTOSAR Classic Platform - CP

<table>
<thead>
<tr>
<th>Application Software Component</th>
<th>Actuator Software Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAR Interface</td>
<td>AUTOSAR Interface</td>
</tr>
</tbody>
</table>

Application Layer

- System Services
- Memory Services
- Communication Services
- I/O Hardware Abstraction
- Onboard Device Abstraction
- Microcontroller Drivers
- Memory Drivers
- Communication Drivers
- I/O Drivers

Complex Drivers

Runtime Environment

AUTOSAR Adaptive Platform - AP

<table>
<thead>
<tr>
<th>SWC</th>
<th>SWC</th>
<th>AUTOSAR Runtime Environment for Adaptive Applications</th>
<th>SWC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARA</td>
<td>ARA</td>
<td>AUTOSAR Interface</td>
<td>ARA</td>
</tr>
</tbody>
</table>

AUTOSAR Interface

- Actuator Software Component
- Sensor Software Component
- Application Software Component

<table>
<thead>
<tr>
<th>API</th>
<th>API</th>
<th>Adaptive AUTOSAR Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Management</td>
<td>Execution Management</td>
<td>Service</td>
</tr>
<tr>
<td>Operating System</td>
<td>Persistence</td>
<td>Service</td>
</tr>
<tr>
<td>Bootloader</td>
<td></td>
<td>Service</td>
</tr>
</tbody>
</table>

Adaptive AUTOSAR Foundation

- (Virtual) Machine / Hardware
- Platform Health Management
- Logging and Tracing
- Hardware Acceleration
- Communication Management
- Update Configuration Management
- Security Management
- Diagnostics

Real Time Requirements

Safety Critical

Computing Power
Agenda

Introduction

- **Unit Testing Demo**
  - Working with AUTOSAR Generated Code
  - Unit Testing AUTOSAR SWCs
  - Conclusions
Complete Ethernet Demo

The demo has several ECUs with different tasks. The central ECU is MyECU. MyECU is connected to one Ethernet channel Ethernet1 and to two CAN channels CAN1 and CAN2. On each CAN channel there are two ECUs connected, USDnode01, USDnode02 on CAN1 and USDnode03, USDnode04 on CAN2.

On Ethernet side there are seven further ECUs designed for the different demo cases.

- DownloadUDSnode01 to DownloadUDSnode04 are used for flash simulation of UDSnodes0x on CAN
- TimeSync being the Slave node to receive synchronized global time from MyECU
- AvTPAudio to show AVB behavior
- Terminal for the SOME/IP demo.
MyECU is the central ECU of the Ethernet Demo. It is running MICROSOAR4 BSW and implements following functionality:

- Provider of SOME/IP services
  - Maths including methods Add, Divide, Multiply and Subtract as well as the event Triangle. These methods implement a simple version of a calculator, and the event generates a triangle signal.
- Many additional included features not described here...
We will select the AUTOSAR Software Component (SWC) SWC_service_Math_Server.c from the application source folder to unit test.
Unit Testing Demo

Sample function to be tested

```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv) {
    int a = 5;
    int b = 7;
    int result = service_Math_Multiply(a, b);
    printf("The result is: %d\n", result);
    return 0;
}

int service_Math_Multiply(int x, int y) {
    return x * y;
}
```

The function `service_Math_Multiply` is a sample function to be tested as part of a unit testing demo.
Create Requirements

- SOME/IP Maths Requirements:

<table>
<thead>
<tr>
<th></th>
<th>Key</th>
<th>ID</th>
<th>Module</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>RQ1000</td>
<td>MS001</td>
<td>SWC_service_Math_Server</td>
<td>&quot;Add&quot;</td>
<td>&quot;Return the sum of A + B&quot;</td>
</tr>
<tr>
<td>3</td>
<td>RQ1001</td>
<td>MS002</td>
<td>SWC_service_Math_Server</td>
<td>&quot;Divide&quot;</td>
<td>&quot;Return the division of A / B&quot;</td>
</tr>
<tr>
<td>4</td>
<td>RQ1002</td>
<td>MS003</td>
<td>SWC_service_Math_Server</td>
<td>&quot;Multiply&quot;</td>
<td>&quot;Return the multiplication of A * B&quot;</td>
</tr>
<tr>
<td>5</td>
<td>RQ1003</td>
<td>MS004</td>
<td>SWC_service_Math_Server</td>
<td>&quot;Subtract&quot;</td>
<td>&quot;Return the subtraction of A - B&quot;</td>
</tr>
<tr>
<td>6</td>
<td>RQ1004</td>
<td>MS005</td>
<td>SWC_service_Math_Server</td>
<td>&quot;Generate Triangle&quot;</td>
<td>&quot;Generate a triangle signal and return the next value of the signal on each call&quot;</td>
</tr>
</tbody>
</table>
Agenda

- Introduction
- Unit Testing Demo
  - Working with AUTOSAR Generated Code
  - Unit Testing AUTOSAR SWCs
  - Conclusions
Working with AUTOSAR Generated Code

What do I Unit Test?

- Most of the time your Application SWCs
- Possibly Complex Device Drivers (CDDs) or Microcontroller Abstraction Layer (MCAL) Drivers
- Maybe custom BSW components
Working with AUTOSAR Generated Code

Tips & Techniques

- From the Project View right click your environment | Properties
Tips & Techniques

- Monitor your build times
- Tools | Options | GUI | Message window timestamps
Tips & Techniques

- Disable comments on the Preprocessor command for your compiler
- For Visual Studio 2013 remove "/C"

- One sample environment reduce
  - Build time from 9:03 to 6:11 (32%)
  - Build size from 1.5GB to 322MB (79%)

- Why?
  - See next slide
Working with AUTOSAR Generated Code

Tips & Techniques

- By design AUTOSAR multiply includes many header files
- VectorCAST flattens the include chain during the harness generation process
- This results in very large source files
- Unit testing does not need source code comments
- Use the Preprocess comment to strip them out
- MemMap.h tends to be the biggest contributor
Working with AUTOSAR Generated Code

Tips & Techniques

- AUTOSAR Macros such as FUNC, P2VAR, and defines get expanded in the coverage viewer

- Source code

- Coverage viewer
Agenda

Introduction
Unit Testing Demo
Working with AUTOSAR Generated Code

- **Unit Testing AUTOSAR SWCs**

Conclusions
Unit Testing AUTOSAR SWCs

Start with a VectorCAST Project

- File | New | VectorCAST Project | Empty Project
Unit Testing AUTOSAR SWCs

Create Unit Test Environments

- Under Group_1 | Create Unit Test Environment | Interactive
Unit Testing AUTOSAR SWCs

Import Requirements

- In three steps:
  - Set options, get fields

- Map attributes to fields, import

- View
Unit Testing AUTOSAR SWCs

Write Test Cases

- Insert Test Case
- Set Input & Expected Values
- Execute
Unit Testing AUTOSAR SWCs

Increase Your Coverage
Unit Testing AUTOSAR SWCs

Link Requirements & Execute Tests
Export Results

Specify CSV export settings:
- Environment name
- Test case name
- Pass/fail status

Append to file: D:\Y\C\D\12300984D\7\external\DemoSomeIpDemo\VC-PRJ_UT\SOMEIP_Math_Results.csv

Appending test data to file: D:\Y\C\D\12300984D\7\external\DemoSomeIpDemo\VC-PRJ_UT\SOMEIP_Math_Results.csv
File closed

************* starting export *************

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1000</td>
<td>&quot;SWC_SERVICE_MATH_SERVER&quot;</td>
<td>&quot;service_Math_Methods.Add.001.001&quot;</td>
<td>&quot;pass&quot;</td>
</tr>
<tr>
<td>RQ1001</td>
<td>&quot;SWC_SERVICE_MATH_SERVER; SWC_SERVICE_MATH_SERVER&quot;</td>
<td>&quot;service_Math_Methods_Divide.001; service_Math_Methods_Divide.002&quot;</td>
<td>&quot;pass; pass&quot;</td>
</tr>
<tr>
<td>RQ1002</td>
<td>&quot;SWC_SERVICE_MATH_SERVER&quot;</td>
<td>&quot;service_Math_Methods_Multiply.001&quot;</td>
<td>&quot;pass&quot;</td>
</tr>
<tr>
<td>RQ1003</td>
<td>&quot;SWC_SERVICE_MATH_SERVER&quot;</td>
<td>&quot;service_Math_Methods_Substract.001&quot;</td>
<td>&quot;pass&quot;</td>
</tr>
<tr>
<td>RQ1004</td>
<td>&quot;SWC_SERVICE_MATH_SERVER&quot;</td>
<td>&quot;service_Math_MainFunction.001&quot;</td>
<td>&quot;pass&quot;</td>
</tr>
</tbody>
</table>
Workflow

- Start with a VectorCAST Project
- Create Unit Test Environments
- Import Requirements
- Write Test Cases
- Increase Your Coverage
- Link Requirements & Execute Tests
- Export Results

Live Demo
Do your AUTOSAR Unit Testing Faster with VectorCAST

- Fast building and executing of your unit test harnesses on host or your ECU
- Quickly create test cases and increase your test coverage with the GUI
- Easy requirement to test case linking and exporting test results
More Information!
Visit our Website for:
> News
> Products
> Demo Software
> Support
> Workshops
> Contact Addresses

www.vector.com
Author:
Krueger, Kurt
Vector North America