|  |  |  |  |
| --- | --- | --- | --- |
| Title | Software Subsystem Test Report <ServicePort Subsystem> | | |
| **Distribution** | Software Archive | | |
| **Author** | Hao Wu | **Date** | 2012-12-03 |
| **Review** | Paul Li  Xiaojin Kuang | **Date** | 2012-12-03 |
| **Approved** | Rock-RendongLi | **Date** | 2012-12-03 |
| **Remarks** |  | | |

Contents

[1 Introduction 3](#_Toc342292799)

[2 Static analyze Report 3](#_Toc342292800)

[2.1 PC\_Lint result from tools 3](#_Toc342292801)

[2.2 PC\_Lint Result check 4](#_Toc342292802)

[3 Module Test Report 5](#_Toc342292803)

[3.1 Module-Test Cases 5](#_Toc342292804)

[3.2 Module Test Result 5](#_Toc342292805)

[3.3 Module Test Result analyze 6](#_Toc342292806)

[References 6](#_Toc342292807)

[Revision History 6](#_Toc342292808)

[Review 6](#_Toc342292809)

**List of Figures**

xx

**List of Tables**

xx

# Introduction

This documents mainly describe the result of module test of ServicePort Subsystem. It has two parts:

1. Static analyze Report
2. Module Test Report

# Static analyze Report

## PC\_Lint result from tools

For the original documents, please refer RESULT.txt in Lint folder of ServicePort subsystem. I just copy into this doc.

*--- Module: ..\Layer2\ServicePort\_layer2\_HW.c*

*Path ..\Layer2\ServicePort\_layer2\_HW.c 136:*

*Warning 641 -- Converting enum 'E\_UNIT\_INITIALIZE\_CLASSES' to int*

*Path ..\Layer2\ServicePort\_layer2\_HW.c 137:*

*Warning 641 -- Converting enum 'E\_UNIT\_INITIALIZE\_CLASSES' to int*

*Path ..\Layer2\ServicePort\_layer2\_HW.c 138:*

*Warning 641 -- Converting enum 'E\_UNIT\_INITIALIZE\_CLASSES' to int*

*Path ..\Layer2\ServicePort\_layer2\_HW.c 139:*

*Warning 641 -- Converting enum 'E\_UNIT\_INITIALIZE\_CLASSES' to int*

*--- Module: ..\Layer2\ServicePort\_rx\_layer2.c*

*--- Module: ..\Layer2\ServicePort\_tx\_layer2.c*

*--- Module: ..\Layer1\ServicePort\_dio.c*

*--- Module: ..\Layer1\ServicePort\_uart.c*

*--- Module: ..\Layer7\ServicePort\_layer7.c*

*Path ..\Layer7\ServicePort\_layer7.c 180:*

*Note 960 -- Violates MISRA Required Rule 91, '#define' used within a block*

*Path ..\Layer7\ServicePort\_layer7.c 199:*

*Note 960 -- Violates MISRA Required Rule 91, '#undef' used within a block*

*Path ..\Layer7\ServicePort\_layer7.c 199:*

*Note 961 -- Violates MISRA Advisory Rule 92, use of '#undef' is discouraged*

*Path ..\Layer7\ServicePort\_layer7.c 349:*

*Info 731 -- Boolean argument to equal/not equal*

*Path ..\Layer7\ServicePort\_layer7.c 539:*

*Info 734 -- Loss of precision (assignment) (16 bits to 8 bits)*

*Path ..\Layer7\ServicePort\_layer7.c 104:*

*Warning 551 -- Symbol 'feAddress' (line 104, file ..\Layer7\ServicePort\_layer7.c) not accessed*

*--- Module: ..\Source\ServicePort\_constructor.c*

*--- Module: ..\Source\ServicePort\_execute.c*

*Path ..\Source\ServicePort\_execute.c 137:*

*Warning 641 -- Converting enum '\_T\_SERVICE\_PORT\_IDX' to int*

*Path ..\Source\ServicePort\_execute.c 137:*

*Warning 641 -- Converting enum 'E\_ATTRIB\_IDX' to int*

*Path ..\Source\ServicePort\_execute.c 159:*

*Warning 641 -- Converting enum 'T\_DATA\_OBJ\_RETURNCODES' to int*

*--- Module: ..\Source\ServicePort\_overload.c*

*Path ..\Source\ServicePort\_overload.c 81:*

*Warning 641 -- Converting enum 'E\_UNIT\_INITIALIZE\_CLASSES' to int*

*Path ..\Source\ServicePort\_overload.c 86:*

*Warning 641 -- Converting enum 'E\_UNIT\_INITIALIZE\_CLASSES' to int*

*Path ..\Source\ServicePort\_overload.c 90:*

*Warning 641 -- Converting enum 'E\_UNIT\_INITIALIZE\_CLASSES' to int*

*Path ..\Source\ServicePort\_overload.c 93:*

*Warning 641 -- Converting enum '\_E\_SERVICE\_PORT\_DC\_INDEXES' to int*

*Path ..\Source\ServicePort\_overload.c 97:*

*Warning 641 -- Converting enum 'E\_UNIT\_INITIALIZE\_CLASSES' to int*

*Path ..\Source\ServicePort\_overload.c 142:*

*Warning 641 -- Converting enum 'T\_DATA\_OBJ\_RETURNCODES' to int*

*Path ..\Source\ServicePort\_overload.c 159:*

*Warning 641 -- Converting enum '\_T\_SERVICE\_PORT\_IDX' to int*

*--- Global Wrap-up*

*:*

*Note 900 -- Successful completion, 20 messages produced*

## PC\_Lint Result check

From the result of PC\_Lint:

*Path ..\Layer7\ServicePort\_layer7.c 180:*

*Note 960 -- Violates MISRA Required Rule 91, '#define' used within a block*

*Path ..\Layer7\ServicePort\_layer7.c 199:*

*Note 960 -- Violates MISRA Required Rule 91, '#undef' used within a block*

*Path ..\Layer7\ServicePort\_layer7.c 199:*

*Note 961 -- Violates MISRA Advisory Rule 92, use of '#undef' is discouraged*

The *#define* and *#undef* is used in a static function and is designed intentionally. The following warnings:

*Path ..\Layer7\ServicePort\_layer7.c 349:*

*Info 731 -- Boolean argument to equal/not equal*

*Path ..\Layer7\ServicePort\_layer7.c 539:*

*Info 734 -- Loss of precision (assignment) (16 bits to 8 bits)*

*Path ..\Layer7\ServicePort\_layer7.c 104:*

*Warning 551 -- Symbol 'feAddress' (line 104, file ..\Layer7\ServicePort\_layer7.c) not accessed*

We have checked the relative source code and think the warning can be ignored.

For the rest of warnings, they are caused by the code generated by EntryTool. And those warnings would not affect the design, and should be ignored too.

# Module Test Report

## Module-Test Cases

Table 1: Module Test Functions and Test Cases

|  |  |  |
| --- | --- | --- |
| **Test Functions** | **Test Cases** | **Test Pass** |
| servicePort.unit.Initialize (const T\_UNIT SLOW\*me, TUSIGN8 typeOfStartUp) | case 1~case 3 | OK |
| servicePort.ServPortGetActivationStatusSRV() &  servicePort.ServPortUpdateActivationStatusSRV() | case 1~case 4 | OK |
| **Data link layer & Application layer** | case1~case 32 | OK |

## Module Test Result

The following are messages printed by the Module test project.

INITIALIZATION TEST:

Subystem Null pointer...assert in File=C:\works\project\Application\ServicePort\Source\ServicePort\_overload.c Line=74

Tasks Initialization...OK

Data Initialization...OK

PRESENCE PIN TEST:

Presence Pin startup check...OK

Presence Pin disabled...OK

Presence Pin enabled...OK

Service Port activation...OK

RECEPTION TEST:

Send a correct Frame (1) - Long Address...OK

Send a correct Frame (2) - Short Address...OK

Send a correct Frame (3) - Long Address & expansion byte...OK

Send a correct Frame (4) - Short Address & expansion byte...OK

Send a correct Frame (5) - Long Address & expansion byte...OK

Send a correct Frame (6) - Long Address & number of bytes = 0...OK

Send a valid Frame with uart error on preambles...OK

Send a valid Frame with uart error on delimiter...OK

Send a valid Frame with a slave to master request...OK

Send an a valid Frame with incorrect preamble in the delimiter status...OK

Send an valid Frame with short address and uart error in the address status...OK

Send a valid Frame with long address and uart error in the address status...OK

Send an invalid Frame with short address and command = 1..OK

Send a valid Frame with long address and uart error in the number of byte status...OK

Send an invalid Frame with long address and buffer overflow...OK

READ OBJECT TEST:

Read value (1) - Attribute index WHOLE\_OBJECT...OK

Read value (2) - Attribute index != WHOLE\_BOJECT...OK

Read value (3) - Invalid sybsystem index...OK

Read value (4) - Invalid object index...OK

Read value (7) - Read on an incorrect Board index...OK

WRITE OBJECT TEST:

Write value (1) - Attribute index WHOLE\_OBJECT...OK

Write value (2) - Invalid sybsystem index...OK

Write value (3) - Invalid object index...OK

Write value (5) - Write on an incorrect Board index...OK

READ MEMORY TEST:

Read memory (1) - Correct reading...OK

Read memory (2) - Fe board reading...OK

Read memory (3) - Buffer overflow...OK

WRITE MEMORY TEST:

Write memory (1) - Correct writing...OK

Write memory (3) - Buffer overflow...OK

OTHER FRAME CHECK TEST:

Too few data...OK

Command not implemented...OK

Crc error...OK

## Module Test Result analyze

From the printed messages by the Module Test project, we can know that Module Test passed. The review work should start and next futher level test plan need to be prepared.

References

|  |  |
| --- | --- |
| **Ref.** | **Document** |
| [1] |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev.** | **Description of Version/Changes** | **Primary Author(s)** | **Date** |
| 00 | First Release | Hao Wu | 2012-12-03 |
|  |  |  |  |

Review

**First Review**

|  |  |
| --- | --- |
| **Document Revision:** | 00 |
| **Review Date:** | 2012-12-03 |

**Review-Participant:**

|  |  |  |
| --- | --- | --- |
| *Dept.* | *Name* |  |
| R&D ShangHai | Paul Li  Xiaojin Kuang | 2012-12-03 rev.00 |

**Decision of the Review:**

|  |  |  |
| --- | --- | --- |
|  | *Decision* | *next steps* |
| **X** | Inspection passed ***without restrictions*** | Phase finished |
|  | Inspection passed ***with restrictions*** | some changes must be done |
|  | Inspection ***not*** passed | Inspection must be repeated |

**Changes Are Proved:**

The Reviewer confirms that all changes are done:

|  |  |  |
| --- | --- | --- |
| proved Rev: | Date: | Reviewer: |
| 00 | 2012-12-03 | Paul Li  Xiaojin Kuang |

**Check list:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | yes | no |
| 1. | The Module Tests are passed successful and documented in a proper way. | **X** |  |
| 1. | The Integration Tests are passed successful and documented in a proper way. |  | **X** |
| 2. | The Code Reviews are passed successful and documented in a proper way. | **X** |  |
| 3. | All safety requirements (Safety Function/Integrity/Measures) are considered. |  | **X** |
| 4. | The accomplished Test Procedures are sufficient to start the Product System Test. | **X** |  |
| 5. | Are all open issues transferred to the defects table? | **X** |  |

**Remarks:**

**Defects**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Checkpoint | Description | Major Defect | Done date |
| 1 |  | Need to update | N | 2012-12-03 |