

# Application Acceptance Test Report

Release Name: SYS-E350-I3.1.1-P51.5

Project: zFAS Series

---

<b>Author:</b>	Cedomir Jovanovic
<b>Security:</b>	Confidential
<b>Document number:</b>	-
<b>Version:</b>	3.50.0
<b>Date:</b>	2019-09-10
<b>Status:</b>	Released
<b>SW-C:</b>	CtApACFStateMachineSlave

---

## TTTech Automotive GmbH

Schoenbrunner Str. 7, A-1040 Vienna, Austria, Tel. + 43 1 585 34 34-0, Fax +43 1 585 34 34-90, office@tttech-automotive.com

No part of the document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the written permission of TTTech Automotive. Company or product names mentioned in this document may be trademarks or registered trademarks of their respective companies. TTTech Automotive undertakes no further obligation in relation to this document.

Copyright © 2018, TTTech Automotive GmbH. All rights reserved.

Subject to change and corrections

Document generation timestamp: 2019-09-10 17:20:09

# Table Of Contents

Revision Chart	3
1. Application Acceptance Test Result	4
1.1 SW-C Overall Test Result & Integration Recommendation	4
1.2 Statistics	4
1.3 Test Case Results	5
2 Test Artefact Information	10
2.1 Test Input Artefacts provided by the SWC-Supplier	10
2.2 Test Output Artefacts generated by the Integrator	10
3 Test Environment Information	12
3.1 Test Management	12
3.2 AAT Test Framework Information	12
3.3 Additional Software Tools	12
3.4 Test PC Software Image	12
Guidelines	13

## Revision Chart

Version	Date	Responsible Person	Description
3.50.0	2019-09-10	Cedomir Jovanovic	Automatic creation of the document





# 1. Application Acceptance Test Result

This chapter documents the overall test results of the performed Application Acceptance Test.

## 1.1 SW-C Overall Test Result & Integration Recommendation

SWC Name	Version	Integration Recommendation
CtApACFStateMachineSlave	SWC-Z322-I3.1.1-P43.0_20181024081651-S4.0_20190215165115	Passed, integrate

Table 1 Application Acceptance Test Result & Integration Recommendation

	Passed, integrate all test cases were without errors or only priority C failures occurred
	Failed, do not integrate at least one priority A failure occurred
	Failed, but still integrate at least one priority B failure occurred
	Not Executed

## 1.2 Statistics

	Priority A		Priority B		Priority C		Total	
Overall number of test cases	6	100%	22	100%	7	100%	35	100%
Executed test cases	6	100%	21	95.5%	7	100%	34	97.1%
Not executed test cases	0	0%	1	4.5%	0	0%	1	2.9%
Passed test cases	6	100%	21	95.5%	6	85.7%	33	94.3%
Failed test cases	0	0%	0	0%	1	14.3%	1	2.9%

Table 2 Statistics

## 1.3 Test Case Results

TC ID	Test Case	Test Case Description	Priority	Executed at	Result <sup>1</sup>	Explanation/Comment	Bug References
1	SWC limits check	Reads and checks all limits from Architectursteckbriefe. It creates temporary file swc_limits.csv used by other processes	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
2	Folder content check	Checks if delivered SWC contains all mandatory subfolders	PRIO_A	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
3	Release notes check	Check Release Notes document against the delivered SWC content	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
4	Supplier's AITR check	Verifies and checks the supplier's AITR document against the delivered SWC content.	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
5	Check 02_libs content (Libraries delivered)	Checks the existence of the libraries in 02_libs folder. At least one library should exist in this folder. Existence should be checked by expected file extension	PRIO_A	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
6	Check for mandatory MAP file	Checks if mandatory MAP file exists in 05_data subfolder. There must be exactly one MAP file	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
7	Check for mandatory MISRA file	Checks if mandatory MISRA file exists in 05_data subfolder. There must be exactly one MISRA file	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
8	Check for mandatory	Checks if one or more build log files are delivered in 05_data subfolder	PRIO_B	2019-09-10 17:15 -	Passed, integrate		No bug references.

TC ID	Test Case	Test Case Description	Priority	Executed at	Result <sup>1</sup>	Explanation/Comment	Bug References
	BUILD LOG files			2019-09-10 17:17			
9	Check for mandatory test cases in the delivery	Checks if all mandatory test cases are delivered in 04_test subfolder	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
10	Test Case naming convention check	Checks if all delivered Test cases (mandatory and additional) follows the naming convention	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
11	Mandatory Test Case folder content check	Checks if every mandatory test case folder contains all mandatory files.	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
12	Additional Test Case folder content check	Checks if additional test case folder (if any) contains all mandatory files.	PRIO_C	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
13	Heap allowed but not used	Checks if SWC is allowed to use Heap but heap is never been used.	PRIO_C	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
14	Heap used but not allowed	Checks if SWC using the heap but it is not allowed to use.	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
15	Host check	Check if SWC's host stated in Release Notes document is correct (the same as the one stated in Architecture Model)	PRIO_C	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.

TC ID	Test Case	Test Case Description	Priority	Executed at	Result <sup>1</sup>	Explanation/Comment	Bug References
16	ASIL level check	Check if SWC's ASIL level stated in Release Notes documents is correct (the same as the one stated in Architecture Model)	PRIO_C	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
17	Supplier's AITR limits check	Checks the limits stated in supplier's AITR document against the limits in Architektursteckbriefe	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
18	Mandatory test case result check	Checks the result of the mandatory test cases stated in supplier's AITR document	PRIO_B	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
19	Additional test case result check	Checks the result of the additional test cases stated in supplier's AITR document (if any)	PRIO_C	2019-09-10 17:15 - 2019-09-10 17:17	Passed, integrate		No bug references.
20	Interface check	Checks symbols used by delivered libraries against the allowed symbol set - white list (RTE symbols from Contract header and specific list of symbols from the Architecture Model)	PRIO_B	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate	0 forbidden symbols found. See unagreed symbols in unagreed_symbols_CtApACFStateMachineSlave.txt	No bug references.
21	RTE interface usage check	Checks if all RTE interface are used	PRIO_C	2019-09-10 17:16:00 - 2019-09-10 17:17	65	85.00% of RTE interfaces not used, 15.00% used.  See unused RTE interfaces in unused_rte_CtApACFStateMachineSlave.txt See RTE symbols statistics table in used_categories_stats_CtApACFStateMachineSlave.txt Maybe not all the RTE interfaces from the Model are needed to be used, see SW-C's Release Notes.	No bug references.
22	Resource consumption check	Checks the resource consumption from the delivered MAP file against the limits in Architektursteckbriefe	PRIO_B		Not Executed	Since this is the APH component and delivered Release version is not the same as the current Release version, test cannot be executed.	
23	XML content check	Checks if Release Notes document and supplier's AITR document (xml	PRIO_B	2019-09-10 17:16:00 -	Passed, integrate		No bug references.

TC ID	Test Case	Test Case Description	Priority	Executed at	Result <sup>1</sup>	Explanation/Comment	Bug References
		format)are valid XML file by the xml standards and by the XML Schema.		2019-09-10 17:17			
24	AIT version check	Checks if AIT version stated in delivered supplier's AITR exists in the list of allowed AIT versions by Preintegration environment (Architecture).	PRIO_B	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.
25	Release version check	Checks if the Release label stated in delivered Release Notes exists in the lists of Acceptable Releases by Preintegration environment(Architecture).	PRIO_A	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.
26	Check allowed compiler flags	Reads BUILD LOG files and checks compiler and linker flags found there against the allowed ones stated in the Preintegration environment (Architecture).	PRIO_A	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.
27	Compiler version check	Checks if Compiler version stated in delivered Release Notes is accepted by the Preintegration environment (Architecture).	PRIO_B	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.
28	MISRA level check	Reads MISRA measurements from supplier's AITR file and checks if level is correct. If MISRA measurements not delivered within AITR file test fails.	PRIO_B	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.
29	Is SWC buildable	Checks the result of Integrator's build process.	PRIO_A	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate	Build successful.	No bug references.
30	Memory Mapping check	Checks if all symbols used in component are in correct memory section.	PRIO_B	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.



TC ID	Test Case	Test Case Description	Priority	Executed at	Result <sup>1</sup>	Explanation/Comment	Bug References
31	Extended Version check	Compares SW-C version with the release label from the build process. Both data is given by supplier in Release Notes.	PRIO_C	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.
32	Availability of dataset files check	The SWC delivery is checked. If it contains the .hex files in the RN (compares RN against the delivery)	PRIO_B	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.
33	Dataset version check	Checks the version found in every hex file against the version stated in RN	PRIO_A	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.
34	Dataset filesize check	Checks the size found in every hex file against the size projected by the Architecture Model.	PRIO_B	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.
35	Check mandatory compiler/linker flags	Reads BUILD LOG files and checks compiler and linker flags found there against the mandatory ones stated in the Preintegration environment (Architecture).	PRIO_B	2019-09-10 17:16:00 - 2019-09-10 17:17	Passed, integrate		No bug references.
				Sum <sup>2</sup>	1/34		

Table 3 Test Case Results

<sup>1</sup>If test case fails, number of errors are shown in this column

<sup>2</sup>Ratio of test cases: failed/executed

	Passed
	Failed
	Not Executed

## 2 Test Artefact Information

The following describes the artefacts which were tested at the integrator and delivered by the SW-C supplier during Application Acceptance Testing

### 2.1 Test Input Artefacts provided by the SWC-Supplier

<b>Name</b>	CtApACFStateMachineSlave
<b>Version</b>	SWC-Z322-I3.1.1-P43.0_20181024081651-S4.0_20190215165115
<b>ASIL Level</b>	D
<b>Host</b>	APH
<b>Description</b>	

Table 4 Tested SWC

Name	Version	Comment
libCtApACFStateMachineSlave_BDL.a		
libCtApACFStateMachineSlave_Implementation.a		
libCtApACFStateMachineSlave_ServerRunnables.a		

Table 5 Tested Release Content

### 2.2 Test Output Artefacts generated by the Integrator

Name	Version	Comment
AATR_TTTech_CtApACFStateMachineSlave.xml	2.0.2	automatic creation
unused_rte_CtApACFStateMachineSlave.txt		File containing list of unused RTE interfaces with its categories
used_rte_CtApACFStateMachineSlave.txt		File containing list of used RTE interfaces with its categories
used_symbols_CtApACFStateMachineSlave.txt		File containing all used symbols in swc libraries
AAT_release_notes20190910171546.log		File containing all information about checks in AAT process
bad_memory_sections_CtApACFStateMachineSlave.txt		File containing information about bad memory sections
unagreed_symbols_CtApACFStateMachineSlave.txt		File containing all unagreed symbols in swc libraries

used_categories_stats_CtApACFStateMachineSlave.txt		File containing statistic table of used RTE symbols in swc libraries
white_CtApACFStateMachineSlave.dat		Whitelist file containing all allowed symbols specific to the SWC.

**Table 6 Generated Test Artefacts**

## 3 Test Environment Information

### 3.1 Test Management

PTC Integrity Baseline Label	-
PTC Integrity Test Session ID	3302346
PTC Integration Test Element	785317

Table 7 Test Management Table

### 3.2 AAT Test Framework Information

Application Integration Test Environment has not been changed.

AAT Test Framework Version	C8_REL_REV176895
Change description	No changes
Effects of Change	No effects

Table 8 AAT Test Framework Information

### 3.3 Additional Software Tools

{NO DATA}

### 3.4 Test PC Software Image

{NO DATA}

# Guidelines

Guidelines:

- \* Delivery location: [https://zfas.sourcecontrol.in.audi.vwg/svn\\_zfas/SW/branches/0420/0300\\_ApplicationLayer/0320\\_Release/CtApACFStateMachineSlave](https://zfas.sourcecontrol.in.audi.vwg/svn_zfas/SW/branches/0420/0300_ApplicationLayer/0320_Release/CtApACFStateMachineSlave)
- \* used buildconfiguration: `python .\build.py -s CtApACFStateMachineSlave update_release`; Also see [DeliveryManual\\_StateMachineSlave.pdf](#) chapter "Erlaeuterung zu den Eigenschaften der SWC-Version" for information about the used buildconfiguration
- \* used additionally builddefines: None; Also see [DeliveryManual\\_StateMachineSlave.pdf](#) chapter "Erlaeuterung zu den Eigenschaften der SWC-Version" for information about the used builddefines and flags