

**VDE Test Report**

Report No. .... :	223766-AS6-1	
VDE File No. .... :	5007383-4970-0007/223766	
Date of issue..... :	2016-04-28	
Laboratory ..... :	VDE Testing and Certification Institute	
Address ..... :	Merianstrasse 28 63069 Offenbach/Main; Germany	
Testing location/ address ..... :	VDE Prüf- und Zertifizierungsinstitut GmbH VDE Testing and Certification Institute Merianstrasse 28, 63069 Offenbach, Germany	
Applicant's name ..... :	Renesas Electronics Europe GmbH	
Applicant's address ..... :	Karl-Hammerschmidt-Straße 42; 85609 Aschheim-Dornach; Germany	
Applied standard(s) ..... :	DIN EN 60335-1 (VDE 0700-1):2012-10; EN 60335-1:2012 DIN EN 60335-1 Ber.1 (VDE 0700-1 Ber.1):2014-04; EN 60335-1:2012/AC:2014 EN 60335-1:2012/A11:2014 DIN EN 60730-1 (VDE 0631-1):2012-10; EN 60730-1:2011 IEC 60335-1(ed.5);am1 IEC 60730-1(ed.5) ;am1	
Test item description ..... :	Self-Diagnostic Routines for Micro controller Family S7	
Trade Mark ..... :	Renesas	
Type reference(s) ..... :	File Name	Revision
	cpu_test.c	1.x
	CPU_Test_Control.asm	1.x
	cpu_test_coupling.c	1.x
	CPU_Test_General_High.asm	1.x
	CPU_Test_General_Low.asm	1.x
	fpu_control.asm	1.x
	fpu_exten.asm	1.x
	fpu_test_coupling.c	1.x
	TestFPUCouplingEnd.asm	1.x
	TestFPUCouplingStart_A.asm	1.x
	TestFPUCouplingStart_B.asm	1.x
	TestGPRsCouplingEnd.asm	1.x
	TestGPRsCouplingStart_A.asm	1.x
	TestGPRsCouplingStart_B.asm	1.x

Report No.:	223766-AS6-1	Page	1	of	15
Disclaimer:					
This test report contains the result of a singular investigation carried out on the product submitted. A sample of this product was tested to found the accordance with the thereafter listed standards or clauses of standards resp.					
The test report does not entitle for the use of a VDE Certification Mark and considers solely the requirements of the specifications mentioned below.					
Whenever reference is made to this test report towards third party, this test report shall be made available on the very spot in full length.					



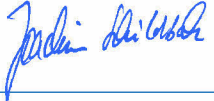

	TestFPUCouplingS0_S3_A.asm	1.x
	TestFPUCouplingS0_S3_B.asm	1.x
	TestFPUCouplingS4_S7_A.asm	1.x
	TestFPUCouplingS4_S7_B.asm	1.x
	TestFPUCouplingS8_S11_A.asm	1.x
	TestFPUCouplingS8_S11_B.asm	1.x
	TestFPUCouplingS12_S15_A.asm	1.x
	TestFPUCouplingS12_S15_B.asm	1.x
	TestFPUCouplingS16_S19_A.asm	1.x
	TestFPUCouplingS16_S19_B.asm	1.x
	TestFPUCouplingS20_S23_A.asm	1.x
	TestFPUCouplingS20_S23_B.asm	1.x
	TestFPUCouplingS24_S27_A.asm	1.x
	TestFPUCouplingS24_S27_B.asm	1.x
	TestFPUCouplingS28_S31_A.asm	1.x
	TestFPUCouplingS28_S31_B.asm	1.x
	TestGPRsCouplingR0_A.asm	1.x
	TestGPRsCouplingR0_B.asm	1.x
	TestGPRsCouplingR1_R3_A.asm	1.x
	TestGPRsCouplingR1_R3_B.asm	1.x
	TestGPRsCouplingR4_R6_A.asm	1.x
	TestGPRsCouplingR4_R6_B.asm	1.x
	TestGPRsCouplingR7_R9_A.asm	1.x
	TestGPRsCouplingR7_R9_B.asm	1.x
	TestGPRsCouplingR10_R12_A.asm	1.x
	TestGPRsCouplingR10_R12_B.asm	1.x
	clock_monitor.c	1.x
	crc.c	1.x
	CRC_Verify.c	1.x
	ramtest_march_c.c	1.x
	ramtest_march_c_HW.c	1.x
	ramtest_march_HW.c	1.x
	ramtest_march_x_wom.c	1.x
	ramtest_march_x_wom_HW.c	1.x
	test_adc12.c	1.x
Ratings .....	N/A	
Supplementary information:		



The Number „1“ represents the executable code.

“x” will be a number from 0 to 99 and indicates changes on comments or labels inside the file or modifications of development tools or environmental routines.

Test sample condition .....	<input checked="" type="checkbox"/>	Non-damaged sample
	Remark:	N/A
Sample entry date .....	2016-04-27	
Date (s) of performance of tests .....	2016-04-27 to 2016-04-28	

Tested by .....		
Name, Signature .....	J. Schildbach (Authorization of test report)	
Function .....	Testing engineer	
Verified by .....		
Name, Signature .....	R. Schwab	
Function .....	Reviewer	

Factory(ies) .....	Renesas Electronics Europe GmbH Karl-Hammerschmidt-Straße 42; 85609 Aschheim-Dornach; Germany
--------------------	--

Possible test case verdicts:	
Test case does not apply to the test object :	N/A
Test object does meet the requirement .....	P (Pass)
Test object does not meet the requirement :	F (Fail)

<b>Final Verdict:</b>	<input checked="" type="checkbox"/> <b>P</b>	<input type="checkbox"/> <b>F</b>
Remark .....	N/A	



Environmental conditions (if applicable)	Ambient temperature	Atmospheric pressure	Relative humidity
Rated values..... :	15-35 °C	860-1060 hPa	30-60 %
Verified values ..... :	N/A	Range confirmed by: Deutscher Wetterdienst (Meteorological service)	N/A



Performed tests						
TABLE R.1 / Table H.1 for software class R.1 / B – GENERAL FAULT / ERROR CONDITIONS						
Component <sup>1)</sup>	Fault/error	Acceptable measures <sup>2) 3) 4)</sup>	Definitions	Document reference for applied measure	Document reference for applied test	Verdict
1 CPU						
1.1 Register	Stuck at	Functional test, or	H.2.16.5			—
		periodic self-test using either:	H.2.16.6	SWD_003_1_PA015_1.0_SW Design Documentation_for_IEC60730	SWV_002_PA015_1.0_SW Verification report_for_IEC60730	P
		– static memory test, or	H.2.19.6			
		– word protection with single bit redundancy	H.2.19.8.2			N/A
1.2 Void						
1.3 Programme counter	Stuck at	Functional test, or	H.2.16.5	Related to application		N/A
		periodic self-test, or	H.2.16.6			
		independent time-slot monitoring, or	H.2.18.10.4			
		logical monitoring of the programme sequence	H.2.18.10.2			
2 Interrupt handling and execution	No interrupt or too frequent interrupt	Functional test; or	H.2.16.5	Related to application		N/A
		time-slot monitoring	H.2.18.10.4			
3 Clock	Wrong frequency (for quartz synchronized clock: harmonics/subharmonics only)	Frequency monitoring, or	H.2.18.10.1	SWD_003_1_PA015_1.0_SW Design Documentation_for_IEC60730	SWV_002_PA015_1.0_SW Verification report_for_IEC60730	P
		time slot monitoring	H.2.18.10.4			N/A
4 Memory						
4.1 Invariable memory	All single bit faults	Periodic modified checksum; or	H.2.19.3.1	SWD_003_1_PA015_1.0_SW Design Documentation_for_IEC60730	SWV_002_PA015_1.0_SW Verification report_for_IEC60730	P
		multiple checksum, or	H.2.19.3.2			N/A
		word protection with single bit redundancy	H.2.19.8.2			N/A
4.2 Variable memory	DC fault	Periodic static memory test, or	H.2.19.6	SWD_003_1_PA015_1.0_SW Design Documentation_for_IEC60730	SWV_002_PA015_1.0_SW Verification report_for_IEC60730	P
		word protection with single bit redundancy	H.2.19.8.2			N/A
Report No.:	223766-AS6-1			Page	5	of 15



4.3 Addressing (relevant to variable and invariable memory)	Stuck at	Word protection with single bit parity including the address	H.2.19.18.2	Covered by 1.1; 3; 4.1; 4.2 and	P
5 Internal data path					—
5.1 Data	Stuck at	Word protection with single bit redundancy	H.2.19.8.2	Related to application	N/A
5.2 Addressing	Wrong address	Word protection with single bit redundancy including the address	H.2.19.8.2		
6 External communication					—
6.1 Data	Hamming distance 3	Word protection with multi-bit redundancy, or CRC – single word , or	H.2.19.8.1 H.2.19.4.1	Related to application	N/A
		transfer redundancy, or	H.2.18.2.2		
		protocol test	H.2.18.14		
6.2 Void					—
6.3 Timing	Wrong point in time	Time-slot monitoring, or scheduled transmission	H.2.18.10.4 H.2.18.18	Related to application	N/A
		Time-slot and logical monitoring, or	H.2.18.10.3		
		comparison of redundant communication channels by either:		—	
	Wrong sequence	– reciprocal comparison	H.2.18.15	Related to application	N/A
		– independent hardware comparator	H.2.18.3		
		Logical monitoring, or	H.2.18.10.2		
time-slot monitoring, or	H.2.18.10.4				
scheduled transmission (same options as for wrong point in time)	H.2.18.18				
7. Input/output					—
7.1 Digital I/O	Fault conditions specified in 19.11.2	Plausibility check	H.2.18.13	Related to application	N/A
7.2 Analog I/O					—
Report No.:	223766-AS6-1			Page	6
				of	15



7.2.1 A/D- and D/A- converter	Fault conditions specified in 19.11.2	Plausibility check	H.2.18.13	SWD_003_1_PA015 _1.0_SW Design Documentation_for_ IEC60730	SWV_002_PA015 _1.0_SW Verification report_for_ IEC60730	P
7.2.2 Analog multiplexer	Wrong addressing	Plausibility check	H.2.18.13	Related to application		N/A
8. Void						—
9 Custom chips. ASIC, GAL, Gate array	Any output outside the static and dynamic functional specification	Periodic self-test	H.2.16.6	Related to application		N/A
Supplementary information: */*						
To 1.1 Registers: The routines include stuck-at and coupling failure detection. The user can set stuck-at detection only or stuck plus coupling failure detection.						
To 1.3 Program Counter In the routines for register test 1.1 a small routine for testing of program counter is integrated. This routine does not cover completely the requirement of standard. It is a support for measures referenced in the table above.						

Additional measures	Details	Reference	Verdict
Watch Dog test	Fail Trigger and Rest Source Monitoring	SWD_003_1_PA015 _1.0_SW Design Documentation_for_ IEC60730	P
Stack Pointer Register and Stack Memory Test	Write-read-verify with pattern for register and March-C for memory	and SWV_002_PA015 _1.0_SW Verification report_for_ IEC60730	P

Additional hardware features	Protection	Reference	Verdict
Ram parity error detection	Stuck at or illegal modification	r01um0001eu0080_synergy_s7g2.pdf	P
Invalid memory access detection function	<ul style="list-style-type: none"> <li>- any access to undefined memory</li> <li>- write access to invariable memory (ROM)</li> <li>- instruction fetch from special predefined memory areas</li> </ul>		P
Window Watch dog with independent clock	<ul style="list-style-type: none"> <li>- Loss of clock of the arithmetic logical unit (ALU) or the complete micro controller</li> <li>- permanent execution of an undefined endless loop</li> <li>- permanent undefined code execution („runaway software“)</li> <li>- time slot monitoring</li> </ul>		P
Port Output Enable	Set PWM outputs to High-Impedance when failure is indicated from external or detected by software		P
On Chip Temperature	Over Temperature for Silicon Device		P
Voltage Monitoring	Under voltage detection to avoid unstable operation		P



Clause	Requirement + Test	Result – Remark	Verdict
R	ANNEX R (NORMATIVE) (60335-1) SOFTWARE EVALUATION		—
	Programmable electronic circuits requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2 validated in accordance with the requirements of this annex	Self-test routines for software of class R.1	P
R.1	Programmable electronic circuits using software		—
	Programmable electronic circuits requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2 constructed so that the software does not impair compliance with the requirements of this standard		P
R.2	Requirements for the architecture		—
	Programmable electronic circuits requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2 use measures to control and avoid software-related faults/errors in safety-related data and safety-related segments of the software		P
R.2.1.1	Programmable electronic circuits requiring software incorporating measures to control the fault/error conditions specified in table R.2 have one of the following structures:		—
	- single channel with periodic self-test and monitoring		N/A
	- dual channel (homogenous) with comparison		N/A
	- dual channel (diverse) with comparison		N/A
	Programmable electronic circuits requiring software incorporating measures to control the fault/error conditions specified in table R.1 have one of the following structures:		—
	- single channel with functional test		P
	- single channel with periodic self-test		P
	- dual channel without comparison		N/A
R.2.2	Measures to control faults/errors		—
R.2.2.1	When redundant memory with comparison is provided on two areas of the same component, the data in one area is stored in a different format from that in the other area		N/A
R.2.2.2	Programmable electronic circuits with functions requiring software incorporating measures to control the fault/error conditions specified in table R.2 and that use dual channel structures with comparison, have additional fault/error detection means for any fault/errors not detected by the comparison		N/A
R.2.2.3	For programmable electronic circuits with functions requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2, means are provided for the recognition and control of		N/A





	errors in transmissions to external safety-related data paths		
R.2.2.4	For programmable electronic circuits with functions requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2, the programmable electronic circuits incorporate measures to address the fault/errors in safety-related segments and data indicated in table R.1 and R.2 as appropriate		P
R.2.2.5	For programmable electronic circuits with functions requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2, detection of a fault/error occur before compliance with clause 19 is impaired	Self-test routines only; compliance to clause 19 has to be insured by the user of the self-test routines	N/A
R.2.2.6	The software is referenced to relevant parts of the operating sequence and the associated hardware functions		P
R.2.2.7	Labels used for memory locations are unique		P
R.2.2.8	The software is protected from user alteration of safety-related segments and data		P
R.2.2.9	Software and safety-related hardware under its control is initialized and terminates before compliance with clause 19 is impaired	Self-test routines only; compliance to clause 19 has to be insured by the user of the self-test routines	N/A
R.3	Measures to avoid errors		—
R.3.1	General		—
	For programmable electronic circuits with functions requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2, the following measures to avoid systematic fault in the software are applied		—
	Software that incorporates measures used to control the fault/error conditions specified in table R.2 is inherently acceptable for software required to control the fault/error conditions specified in table R.1	Class R.1 only	N/A
R.3.2	Specification		—
R.3.2.1	Software safety requirements:	Software Id: 1.x	P
	The specification of the software safety requirements includes the descriptions listed		P
R.3.2.2	Software architecture		—
R.3.2.2.1	The specification of the software architecture includes the aspects listed - techniques and measures to control software faults/errors (refer to R.2.2); - interactions between hardware and software; - partitioning into modules and their allocation to the specified safety functions;	See table R.1	P



	<ul style="list-style-type: none"> <li>- hierarchy and call structure of the modules (control flow);</li> <li>- interrupt handling;</li> <li>- data flow and restrictions on data access;</li> <li>- architecture and storage of data;</li> <li>- time-based dependencies of sequences and data</li> </ul>		
R.3.2.2.2	The architecture specification is validated against the specification of the software safety requirements by static analysis		P
R.3.2.3	Module design and coding		—
R.3.2.3.1	Based on the architecture design, software is suitably refined into modules		P
	Software module design and coding is implemented in a way that is traceable to the software architecture and requirements		P
R.3.2.3.2	Software code is structured		P
R.3.2.3.3	Coded software is validated against the module specification by static analysis		P
	The module specification is validated against the architecture specification by static analysis	Reviews and source code walk through	P
R.3.3.3	Software validation		—
	The software is validated with reference to the requirements of the software safety requirements specification		P
	Compliance is checked by simulation of:		—
	- input signals present during normal operation		P
	- anticipated occurrences		P
	- undesired conditions requiring system action		P

H.11.12.3	Measures to avoid errors (60730-1)		
H.11.12.3.1	For controls with software Class B or C the V-model for the software life cycle should be applied	Class B Remark: Software self-diagnostics are made of functions to be executed one by one in series, there are no complex relationships and interactions to consider.	P
	Measures used for software class C are inherently acceptable for software class B		N/A
	Other methods are possible if they incorporate disciplined and structured processes including design and test phases		N/A

H.11.12.3.2	Specification		P
H.11.12.3.2.1	Software safety requirements		N/A
H.11.12.3.2.1.1	The specification of the software safety requirements includes:		N/A
	<ul style="list-style-type: none"> <li>A description of each safety related function to be implemented, including its response time(s): <ul style="list-style-type: none"> <li>functions related to the application including their related software classes</li> <li>functions related to the detection, annunciation and management of software or hardware faults</li> </ul> </li> </ul>		N/A
	<ul style="list-style-type: none"> <li>A description of interfaces between software and hardware</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>A description of interfaces between any safety and non-safety related functions</li> </ul>		N/A
H.11.12.3.2.2	Software architecture		
H.11.12.3.2.2.1	The description of software architecture shall include the following aspects:		
	<ul style="list-style-type: none"> <li>Techniques and measures to control software faults/errors (refer to H.11.12.2)</li> </ul>		P
	<ul style="list-style-type: none"> <li>Interactions between hardware and software</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>Partitioning into modules and their allocation to the specified safety functions</li> </ul>		P
	<ul style="list-style-type: none"> <li>Hierarchy and call structure of the modules (control flow)</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>Interrupt handling</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>Data flow and restrictions on data access</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>Architecture and storage of data</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>Time based dependencies of sequences and data</li> </ul>		N/A
H.11.12.3.2.2.2	The architecture specification shall be verified against the specification of the software safety requirements by static analysis. Acceptable methods are:		
	<ul style="list-style-type: none"> <li>Control flow analysis</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>Data flow analysis</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>Walk-throughs / design reviews</li> </ul>		P
H.11.12.3.2.3.1	Based on the architecture design, software is suitably refined into modules. Software module design and coding are implemented in a way that is traceable to the software architecture and requirements		N/A
H.11.12.3.2.3.2	Software code is structured		N/A
H.11.12.3.2.3.3	Coded software is verified against the module specification, and the module specification is verified against the architecture specification by static analysis		N/A
H.11.12.3.2.4	Design and coding standards	MISRA	P
	Program design and coding standards is consequently used during software design and maintenance		P
	Coding standards specify programming practice, proscribe unsafe language features, and specify		P



	procedures for source code documentation as well as for data naming conventions		
H.11.12.3.3	Testing		
H.11.12.3.3.1	Module design (software system design, software module design and coding)		P
H.11.12.3.3.1.1	A test concept with suitable test cases is defined based on the module design specification.		P
H.11.12.3.3.1.2	Each software module is tested as specified within the test concept		P
H.11.12.3.3.1.3	Test cases, test data and test results are documented		P
H.11.12.3.3.1.4	Code verification of a software module by static means includes such techniques as software inspections, walk-throughs, static analysis and formal proof		P
	Code verification of a software module by dynamic means includes functional testing, white-box testing and statistical testing		P
H.11.12.3.3.2	Software integration testing		
H.11.12.3.3.2.1	A test concept with suitable test cases is defined based on the architecture design specification		N/A
H.11.12.3.3.2.2	The software is tested as specified within the test concept		N/A
H.11.12.3.3.2.3	Test cases, test data and test results are documented		N/A
H.11.12.3.3.3	Software validation		
H.11.12.3.3.3.1	A validation concept with suitable test cases is defined based on the software safety requirements specification		P
H.11.12.3.3.3.2	The software is validated with reference to the requirements of the software safety requirements specification as specified within the validation concept.		P
	The software is exercised by simulation or stimulation of:		P
	• input signals present during normal operation		P
	• anticipated occurrences		P
	• undesired conditions requiring system action		P
H.11.12.3.3.3.4	Test cases, test data and test results are documented		P
H.11.12.3.4	Other Items		
H.11.12.3.4.1	Tools, programming languages are assumed to be suitable if they comply with "increased confidence from use" according to IEC 61508-7, C.4.4		P
H.11.12.3.4.2	Management of software versions: All versions are uniquely identified for traceability		P
H.11.12.3.4.3	Software modification		
H.11.12.3.4.3.1	Software modifications are based on a modification request which details the following:		N/A
	• the hazards which may be affected		N/A



	<ul style="list-style-type: none"> <li>the proposed change</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>the reasons for change</li> </ul>		N/A
H.11.12.3.4.3.2	An analysis is carried out to determine the impact of the proposed modification on functional safety.		N/A
H.11.12.3.4.3.3	A detailed specification for the modification is generated including the necessary activities for verification and validation, such as a definition of suitable test cases		N/A
H.11.12.3.4.3.4	The modification are carried out as planned		N/A
H.11.12.3.4.3.5	The assessment of the modification is carried out based on the specified verification and validation activities. This may include:		N/A
	<ul style="list-style-type: none"> <li>a reverification of changed software modules</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>a reverification of affected software modules</li> </ul>		N/A
	<ul style="list-style-type: none"> <li>a revalidation of the complete system</li> </ul>		N/A
H.11.12.3.4.3.6	All details of modification activities are documented		N/A
H.11.12.3.5	For class C control functions: One of the combinations (a–p) of analytical measures given in the columns of table H.9 is used during hardware development .....		N/A

Supplementary information:

The self-diagnostic routines mentioned under I are foreseen for following measures of table R.1 / H.1 of.

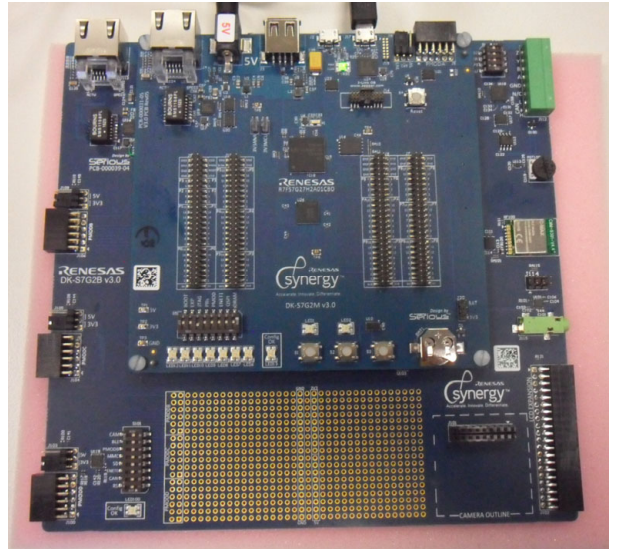
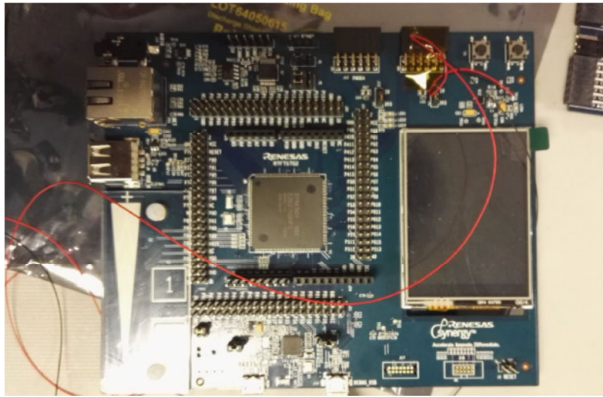
File Name	Measure
cpu_test.c	1.1 CPU Register
CPU_Test_Control.asm	
cpu_test_coupling.c	
CPU_Test_General_High.asm	
CPU_Test_General_Low.asm	
fpu_control.asm	
fpu_exten.asm	
fpu_test_coupling.c	
TestFPUCouplingEnd.asm	
TestFPUCouplingStart_A.asm	
TestFPUCouplingStart_B.asm	
TestGPRsCouplingEnd.asm	
TestGPRsCouplingStart_A.asm	
TestGPRsCouplingStart_B.asm	
TestFPUCouplingS0_S3_B.asm	
TestFPUCouplingS4_S7_A.asm	
TestFPUCouplingS4_S7_B.asm	
TestFPUCouplingS8_S11_A.asm	



TestFPUCouplingS8_S11_B.asm	
TestFPUCouplingS12_S15_A.asm	
TestFPUCouplingS12_S15_B.asm	
TestFPUCouplingS16_S19_A.asm	
TestFPUCouplingS16_S19_B.asm	
TestFPUCouplingS20_S23_A.asm	
TestFPUCouplingS20_S23_B.asm	
TestFPUCouplingS24_S27_A.asm	
TestFPUCouplingS24_S27_B.asm	
TestFPUCouplingS28_S31_A.asm	
TestFPUCouplingS28_S31_B.asm	
TestGPRsCouplingR0_A.asm	
TestGPRsCouplingR0_B.asm	
TestGPRsCouplingR1_R3_A.asm	
TestGPRsCouplingR1_R3_B.asm	
TestGPRsCouplingR4_R6_A.asm	
TestGPRsCouplingR4_R6_B.asm	
TestGPRsCouplingR7_R9_A.asm	
TestGPRsCouplingR7_R9_B.asm	
TestGPRsCouplingR10_R12_A.asm	
TestGPRsCouplingR10_R12_B.asm	
clock_monitor.c	3. Clock
crc.c	4.1 invariable memory
CRC_Verify.c	
ramtest_march_c.c	4.2 variable memory
ramtest_march_c_HW.c	
ramtest_march_HW.c	
ramtest_march_x_wom.c	
ramtest_march_x_wom_HW.c	
test_adc12.c	7.2.1 A/D- and D/A- converter

Photo documentation:

Test Setup



Testing and measuring equipment:

Editor:	IAR Embedded Workbench for ARM, v. 7.40 IAR Embedded Workbench Common Components, v. 7.2
Compiler/Linker:	IAR Embedded Workbench for ARM, v. 7.40 IAR Embedded Workbench Common Components, v. 7.2
Debugger:	IAR Embedded Workbench for ARM, v. 7.40 IAR Embedded Workbench Common Components, v. 7.2
Hardware:	Renesas DK-S7G2 Development Kit for Synergy S7

Uncertainty of measurement (optional according to sub-clause 5.10.3.1.c of IEC 17025):

N/A

**END OF TEST REPORT**