exida DEVELOPMENT

Safety Library

The Safety Library is a set of C modules implementing Safety Integrity Functions also called Safety Mechanisms to protect against systematic and random failures.

- **FlwCtl** – Control Flow Monitor is a module used to detect if the elements of a program are processed in a wrong sequence.
- **TemCtl** – The Temporal Control module is devoted to supervise the reliability of program execution in consideration of periodicity and timing constraints of periodicity.
- **QeGAnC** – This module generates the question (challenge) and verifies the obtained answer (response) in a question/answer based safety concept.
- **AnsGen** – This module generates the answer (response) in a question/answer based safety concept. The answer is created based on the results of the diagnostic modules.
- **CfgReq** – This module is for managing and monitoring the CPU configuration registers in order to detect an unwanted change. The mechanisms adopted in this module are: input parameter check, anti illegal preemption mechanism, protection of internal variables and protection of configuration registers
- **E2ELib** – This module defines a minimal safety protocol to ensure safety communication over CAN, LIN, SPI ect.
- **NvmPrt** – This module detects corruption of data in Non Volatile Memory (Flash, EEPROM, etc..) and, depending on the algorithm used, allows the error correction.
- **PrtBuf** – This module provides protected buffer able to detect corruption due to interference by other non safety modules.
- **ADCDrv** – The ADC module initializes and controls the internal Analog to Digital converter; it protects the configuration registers and provides an internal state machine to avoid illegal operations on a BUSY channel. Other mechanisms adopted are input parameter check and protection of internal variables.
- **MpuDrv** – This module handle the Memory Protection (MPU) to detect unwanted memory accesses. The mechanisms adopted in this module are: input parameter check, anti illegal preemption mechanism, protection of internal variables and protection of configuration registers

- **PORT** – Autosar compliant PORT driver which manages the configuration of the Digital I/O channels and protects them against unwanted changes. Other mechanisms adopted are input parameter check, anti illegal preemption mechanism and protection of internal variables.

- **DIO** – Autosar compliant DIO Driver which provides services for reading and writing to/from I/O channels, ports and channel groups. It shall be associated with the PORT driver to ensure the channels protection. The mechanisms adopted in this module are input parameter check, anti illegal preemption mechanism, protection of internal variables and protection of configuration registers

- **StkPrt** – The Stack Protection module allows to detect a stack overflow and underflow. Some particulars patterns are located before and after the stack space in memory; these patterns are cyclically checked for their integrity.

- **PWMDrv** – The PWM driver provides functions for initialization and control of the microcontroller internal PWM (Pulse Width Modulation), by modifying the duty cycle and the signal period time. It provides internal mechanisms for the protection of PWM configuration registers, input parameters check, anti illegal preemption mechanism and protection of internal variables

- **SftITC** – This module groups a set of functions to ensure safe/protected inter-task communication.

- **DblStr** – This module is used to detect data corruption of safety critical variables by using the Double Inverted mechanism.

- **CpuTst** – The Core Test module performs a cyclic test of the core instruction set. This HW dependent SIF shall be developed for the specific µC.

- **RamTst** – This module executes (at start-up, online or shut-down) selected RAM tests to verify the integrity of the Memory.

- **SIFECo** – Safety Integrity Function Events Collector is the module that stores the integrity results of all other modules. It can be queried to verify the system integrity status.

- **MskCnt** – The purpose of this module is to create counters related to events of safety mechanisms. These counters are incremented in case of error up to a maximum threshold, beyond which the error is reported to SIFECo.

- **GPTDrv** – This module initializes and controls the internal General Purpose Timers of the µC. It provides services for starting and stopping HW timers, for getting their values and for controlling interrupt notifications. The mechanisms adopted in this module are: input parameter check, anti illegal preemption mechanism, protection of internal variables and protection of configuration registers

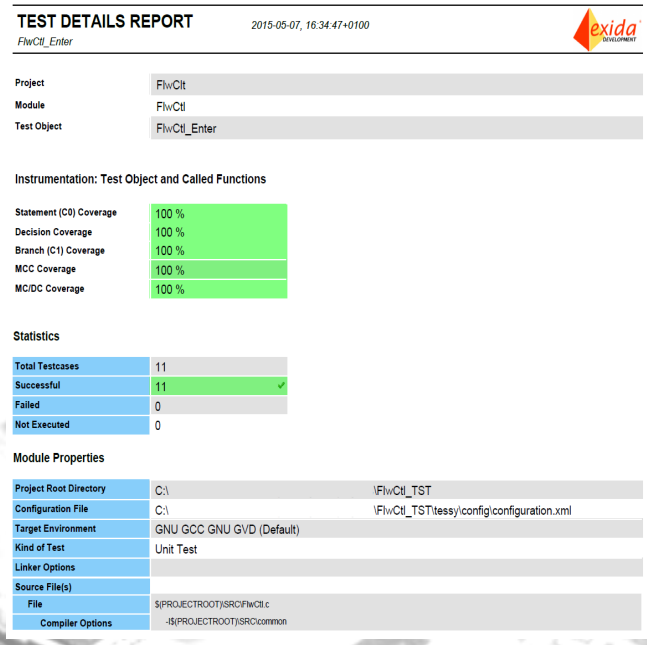
Note(s) :

1. The whole safety library is written in ANSI C;
2. Some module are HW dependent and the availability for a given micro needs to be checked

Delivery Includes

- **Module Detailed Specification** – Module level detailed requirements specification
- **Module Level Design** – A detailed design model in UML (see example below) is included for each module both as a .pdf file and .eap (Enterprise Architect ®) file.
- **Static Code Analysis** – A static code analysis report showing the conformance to MISRA-C 2004 and the Cyclomatic Complexity Analysis is provided for each modules. Only exceptions are modules written in assembly.
- **Module Level Implementation** – For each module of the safety library acquired, the Binary code or the obfuscated source code is provided depending on the agreement.
- **Unit Tests** – Unit test specifications and reports are included for each module of the library.
- **Code Coverage** – Code Coverage reports are included to show the achieved branch coverage of the specified tests.

The above items are selectable for each module.



TEST DETAILS REPORT
FlwCtl_Enter
2015-05-07, 16:34:47+0100

Project	FlwClt
Module	FlwCtl
Test Object	FlwCtl_Enter

Instrumentation: Test Object and Called Functions

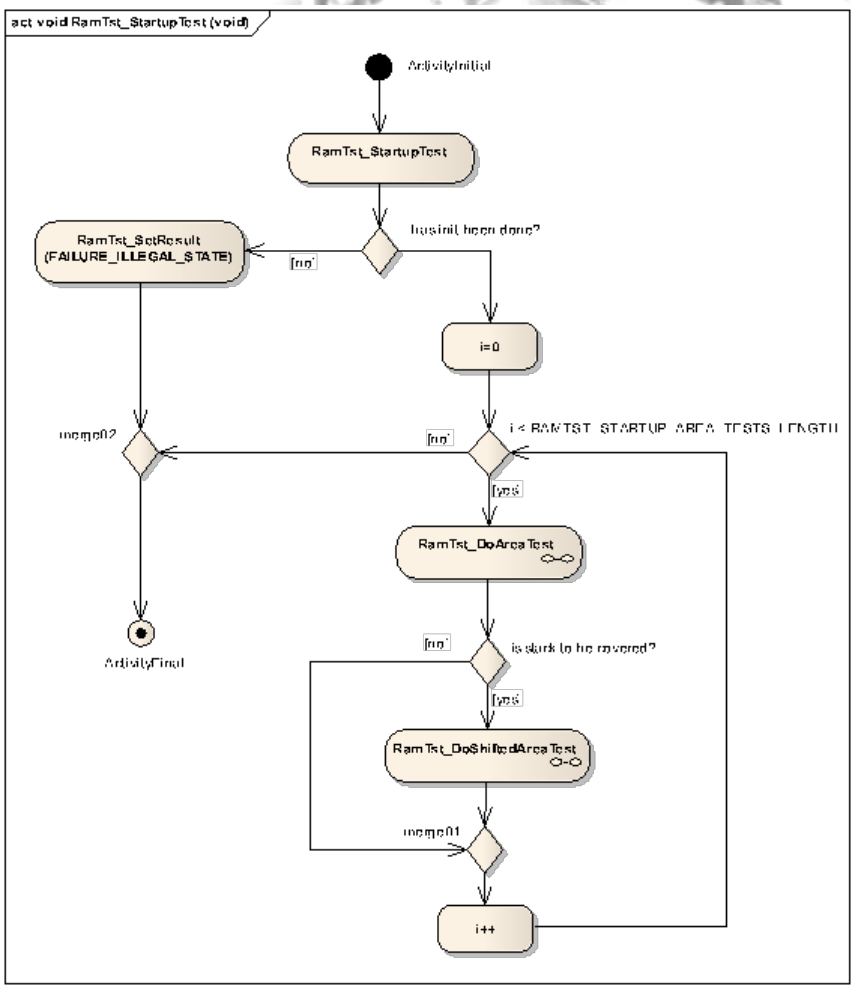
Statement (C0) Coverage	100 %
Decision Coverage	100 %
Branch (C1) Coverage	100 %
MCC Coverage	100 %
MC/DC Coverage	100 %

Statistics

Total Testcases	11
Successful	11
Failed	0
Not Executed	0

Module Properties

Project Root Directory	C:\	\FlwCtl_TST
Configuration File	C:\	\FlwCtl_TST\tessy\config\configuration.xml
Target Environment	GNU GCC GNU GVD (Default)	
Kind of Test	Unit Test	
Linker Options		
Source File(s)		
File	$(PROJECTROOT)/SRC/FlwCtl.c	
Compiler Options	-I$(PROJECTROOT)/SRC\common	



Related Services:

Integration – Integration services are also provided on demand.

Configuration – Configuration services are also provided on demand.

Customization – The library can be customized to the specific needs of a project.

For More Info Contact:

Dr. Giovanni Dallara
exida development S.r.l.
Via Ribes 5,
10010 - Colleretto Giacosa (TO)
Italy

Tel.: +39 01251925223
E-mail: gdallara@exida.com
Web: www.exida.eu