Rq ID	RAM TEST MAIN FUNCATION
1	The RamTst module consists of a RamTst_MainFunction() for background testing, the APIs for foreground testing, several configuration and status APIs (Application Programming Interface), and several configuration containers.
6	RamTst_Allow Permit the RamTst_MainFunction() to perform testing at its next scheduled call.
36	RamTst_Stop Prohibit the RamTst_MainFunction() from performing tests at its next scheduled call. When RamTst_Stop is called, testing stops after the current atomic sequence. Test status is retained, but test parameters (block number, loop count, etc.) are discarded.
37	RamTst_Suspend Temporarily prohibit the RamTst_MainFunction() from performing tests at its next scheduled call. When RamTst_Suspend is called, testing stops after the current atomic sequence. Test status and test parameters are retained.
38	RamTst_Resume Permits the RamTst_MainFunction() to continue testing at the point where it was suspended, at its next scheduled call. Testing continues according to the saved test parameters.
2	RamTst_MainFunction() is the scheduled function for background testing.
3	For background testing, RamTst_MainFunction() is called periodically by a scheduler, and is interruptible. One complete test consists of testing with one algorithm over the memory space defined by the currently selected configuration. This complete test is split up over many scheduled calls.
4	The state “test running” does not necessarily mean that testing is continuously being performed. For foreground testing, it does mean that the test is directly performed by an API call and RamTst_MainFunction() is not scheduled. For background testing, it only means that RamTst_MainFunction() is permitted to test a small portion of the RAM when it is called periodically by the scheduler.
39	RamTst_Suspend causes a state change to “test suspended” at the end of the current RamTst_MainFunction() atomic sequence if RamTst_MainFunction() is actively testing.
11	RamTst_Allow is called to permit the RamTst_MainFunction() to test when called, it does not initiate any test itself.
40	RamTst_Stop must first be called before selecting another configuration parameter set by RamTst_SelectAlgParams.
29	RamTst_ChangeNumberOfTestedCells operates at the end of the current RamTst_MainFunction() atomic sequence if RamTst_MainFunction() is actively testing. For a foreground test,RamTst_ChangeNumberOfTestedCells is not relevant.
30	The scheduler calls MainFunction periodically, testing the RamTstNumberOftestedCellds at each scheduled call.
14	A complete RAM test is defined as running one algorithm over all of that algorithm’s configured memory blocks
15	After completing a RAM test, RamTst_MainFunction() begins a new test at the next scheduled time, starting at the START_ADDRESS of the first RAM block
16	In the example below, the test is performed using RAMTST_ALG_PARAMS_ID1 over its preconfigured memory block definitions (BLOCK_ID’s A through n), in z calls to MainFunction().
31	RamTstMaxNumberOfTestedCells for each RamTstAlgParams container. The integrator must carefully select RamTstMaxNumberOfTestedCells such that it puts an upper limit on the run time of RamTst_MainFunction() in a background task according to the system needs for throughput
17	No matter how many blocks or partial blocks are tested in one RamTst_MainFunction() scheduled call, test status information must be maintained for each block separately.
32	RamTst_MainFunction() to test a RamTstNumberOfTestedCells of memory using the selected algorithm, until the entire defined area of RAM is tested.
41	RamTst_MainFunction() can be interrupted at the end of each atomic sequence during a scheduled call.
42	RamTst_MainFunction(): Is made up of one or more atomic (i.e. uninterruptable) pieces of code. The number of cells that can be tested in one atomic sequence is considered as implementation specific, thus it is not determined by any (standardized) configuration parameter. However, it is expected that at least
44	RamTstMinNumberOfTestedCells are completely tested during one atomic sequence. It should be noted, that in general the detection of coupling faults between cells is limited to those cells which are tested together in the same atomic sequence.
45	At the end of each atomic piece, internal flags are checked to see if an OS task has changed any parameter of the state chart, and to respond to question-type APIs.
18	RamTst_MainFunction(): Knows inherently: - which algorithm it is using; - which memory blocks must be tested for this algorithm, - start and end addresses of each block; - number of cells to test at each call - further parameters for the test
19	RamTst_MainFunction(): Remembers: - which block it is in; - which address to start at in the next call; - status of the test; - overall test results; - test results for each block.
19-	When RamTstNumberOfTestedCells is reached, RamTst_MainFunction() ends testing for that scheduled call, and starts testing in the next scheduled call at the next (saved) address.
46	When the end of a block is reached during a scheduled call, RamTst_MainFunction() continues testing at the beginning of the next block, and continues until RamTstNumberOfTestedCells is reached. (Note: The atomic test sequence should be careful to take into account any issues regarding crossing into the next block.)
19--	When all blocks are fully tested, RamTst_MainFunction() issues a notification and repeats testing at the first block.
54	If there is an error during testing, RamTst_MainFunction() issues a notification (if configured) and continues testing.
34	For background testing, the ECU State Manager or the BSW Scheduler must schedule the RAM Test main function. The number of cells tested in one cycle is set as a default at pre-compile or link time based upon the needs of the scheduler.
55	The function RamTst_MainFunction shall update the overall test result status and error is updated based on the current status of RamTstAlgParams.
52	RamTst_MainFunction() tests memory blocks at its next scheduled call. In case of successful test result report Pass. In case of an error always report Fail.
21	RAM test for RamTst_MainFunction function shall Fail if test result of at least one block is RAMTST_RESULT_NOT_OK.] ()
22	RAM test for RamTst_MainFunction function shall Pass if test result of all the blocks is RAMTST_RESULT_OK.]

12	After a RamTst_Stop call, RamTst_MainFunction shall not begin testing again when called by the scheduler until after a RamTst_Allow call. ⌋ ()
12-	The function RamTst_Resume shall permit the RamTst_MainFunction to continue testing at the point where it was suspended, at its next scheduled call. Testing continues according to the saved test states. The function RamTst_Resume shall change the execution status to RAMTST_EXECUTION_RUNNING if it has been RAMTST_EXECUTION_SUSPENDED. ⌋ ()
5	RamTst_MainFunction is not scheduled during the foreground test.
23	If the RAM Test execution status is RAMTST_EXECUTION_RUNNING, the function RamTst_MainFunction shall continue to test the RAM blocks defined in the selected RamTstAlgParams. ⌋ (SRS_RamTst_13809)
24	If the RAM Test execution status is RAMTST_EXECUTION_RUNNING and if no blocks have yet been tested (first call of the function), then the function RamTst_MainFunction shall start testing with the first configured RAM block in the selected RamTstAlgParams. ⌋ ()
13	If the RAM Test execution status is not RAMTST_EXECUTION_RUNNING when this API is called, the function RamTst_MainFunction shall return immediately without any actions.
25	The function RamTst_MainFunction shall update the test result status of single blocks according to The test result for a specific block (identified in the given RamTstAlgParams) – shall be · RAMTST_RESULT_NOT_TESTED if this block is considered as not yet tested. · RAMTST_RESULT_UNDEFINED if a test on this block is running. · RAMTST_RESULT_OK if all memory cells in this block have been tested sucessfully. · RAMTST_RESULT_NOT_OK if a failure has been detected for at least one memory cell in this block. ⌋ ()
53	The function RamTst_MainFunction shall update the overall test result status according to
	The overall test result – for the set of blocks in the current RamTstAlgParams – shall be · RAMTST_RESULT_NOT_TESTED if no test was started yet (after reset or deinit). · RAMTST_RESULT_UNDEFINED if a test was started, not all blocks have yet been tested and no block result is RAMTST_RESULT_NOT_OK. · RAMTST_RESULT_OK if all blocks have been tested with result status RAMTST_RESULT_OK. · RAMTST_RESULT_NOT_OK if at least one block test result is RAMTST_RESULT_NOT_OK regardless whether all blocks have been already tested or not. ⌋ () If at least one block test result is RAMTST_RESULT_NOT_OK, then the function shall report the production error RAMTST_E_RAM_FAILURE to the DEM. ⌋
27	After the function RamTst_MainFunction has completed testing all RAM blocks configured in the selected RamTstAlgParams, the next call of the function RamTst_MainFunction shall restart the test from the beginning. ⌋ ()
35	The function RamTst_MainFunction shall test the defined number of RAM cells within one call. The defined number is specified by the function RamTst_ChangeNumberOfTestedCells or by initialization.
28	Updating the test results will overwrite the result from a previous test of the current block and the overall test result, including the case that the background test was resumed after a partial foreground test of the current block.
56	If the configuration parameter RamTstTestPolicy for a block is set to RAMTEST_NON_DESTRUCTIVE, the test algorithm shall restore the original memory content of the tested cells of this block after the test (given that no error is detected). ⌋ (SRS_RamTst_13811) Hint: For a transparent test algorithm, this behavior is automatically fulfilled without additional overhead. For a non-transparent test algorithm, this option means overhead in runtime/memory in order to save and restore the content. [SWS_RamTst_00201] ⌈ If the configuration parameter RamTstTestPolicy for a block is set to RAMTEST_DESTRUCTIVE, the test algorithm shall fill the tested cells after the test with the bit pattern defined for this block by parameter RamTst_FillPattern (given that no error is detected). ⌋ (SRS_RamTst_13812) This requirement shall ensure reproducible behavior.
57	⌈ If the configuration parameter RamTstTestPolicy for a block is set to RAMTEST_DESTRUCTIVE, the test algorithm shall fill the tested cells after the test with the bit pattern defined for this block by parameter RamTst_FillPattern (given that no error is detected). ⌋ (SRS_RamTst_13812) This requirement shall ensure reproducible behavior. Hint: For a transparent test algorithm, specifying this option would mean runtime overhead. For a non-transparent algorithm, the runtime overhead can be minimized, if the fill pattern corresponds to a constant value left behind by the algorithm anyhow.
47	In general, the actual test algorithm within one call of RamTst_MainFunction must be performed within one or more atomic sequences. Only within one atomic sequence, the memory written by the algorithm is allowed to be corrupted during the test. This means, that the algorithm can be applied only to those cells accessed within one atomic sequence, so that the detection of coupling faults between cells (by background test) is restricted to those cells which are included in one atomic sequence.
58	RamTst_MainFunction (Examples) The first example sequence shows the initialization of the RAM Test module, a foreground Run Full Test request, error notification, and the cyclic call background testing.
48	A cyclic background task called by a scheduler consists of several small atomic sequences in succession. At the end of each atomic sequence, the command variables are checked to see if any command has been received, and corresponding actions are taken.
49	The stop request is handled following the currently running atomic sequence of the main routine, or at the next cyclic call of the main routine if it is not currently running. The allow request is handled at the next cyclic call of the main routine.