|  |  |  |
| --- | --- | --- |
| **Author** | **Review** | **Approval** |
| SwD: J.Ze Ndi | STL: A.Diankouika | SW line manager: A.Vaché |
|  |  |  |
| **Distribution** | | |
| SW line manager: A.Vaché  Project leader: X. Biton  Design leader: E. Abid | SW designer: H. Zetti  SW designer: S.Bouazizi | SW designer: L. Medas  SW designer: J. Ze-Ndi |
|  |  |  |

# Table of content

1. General Information 3

1.1. Revision history \* 3

1.2. Purpose and Scope 3

1.3. Referenced documents 3

1.3.1. External documents 3

1.3.2. Internal Documents 3

1.4. Terminology and definitions 3

2. SW atomic architectural unit design 4

2.1. Overview 4

2.1.1. Interaction with other modules 4

2.2. Traceability 4

2.3. Files structure 5

3. FEATURES 6

3.1. Services 6

3.1.1. Exported services 6

3.1.2. Local services 12

3.2. Types 27

3.2.1. Global types 27

3.2.2. Local types 27

3.3. Variables 28

3.4. Constants 29

4. EEPROM 32

5. Configuration 32

6. Compilation Options 32

# General Information

## Revision history \*

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author(s)** | **Description/comment** |
| 1.1 | 22/12/2020 | Pierre-Olivier Pilot | Document creation |
| 1.2 | 13/12/2019 | Joseph Ze Ndi | Document updated |

*\* Template history is found in the CM tool used for templates*

## Purpose and Scope

The purpose of this document is to provide an overview of the ATM operation principle, and to present the implementation choices in terms of module and function splitting.

## Referenced documents

### External documents

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Reference** |
|  |  |  |
|  |  |  |
|  |  |  |

### Internal Documents

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Reference** |
|  | SW architecture design interface description | E1355904 in PTC |
|  | ECU parameters specification | DOORS/PP4G/DES/  TF-J |
|  | SBE\_4G\_NVP\_layout | PTC |
|  | To Perform Autotests | DOORS/PP4G/DES/TF-H |

## Terminology and definitions

|  |  |
| --- | --- |
| **Terminology** | **Meaning** |
| AAU | Atomic architectural unit |
| AEC | Autoliv Error Code |
| ATM | Auto-Tests Manager |
| SCH | Scheduler |
| MMG | Mode ManaGement |
| SW | Software |
| DEM | Diagnostic Event Manager |

# SW atomic architectural unit design

## Overview

The aim of the “ATM” (Auto-Tests Manager) component is to schedule auto-tests and manage the reported failures. ATM handles periodic or one shot test.

The ATM provides 5 test lists and a unique one for the startup auto-test. ATM main function is called by the SCH every 2ms.

The configurability of the ATM includes:

* Test types : cyclic or one shot
* List, period and slot/decade
* MMG inhibiting modes
* Callback function
* Enable/Disable switch

### Interaction with other modules

## Traceability

The traceability matrix is built from the system specification until the architecture document.

Then, refer to [Doc1] the get the traceability against the functional, design and safety requirements, related to this unit.

## Files structure

Below is the description of the files structure defined for this unit.

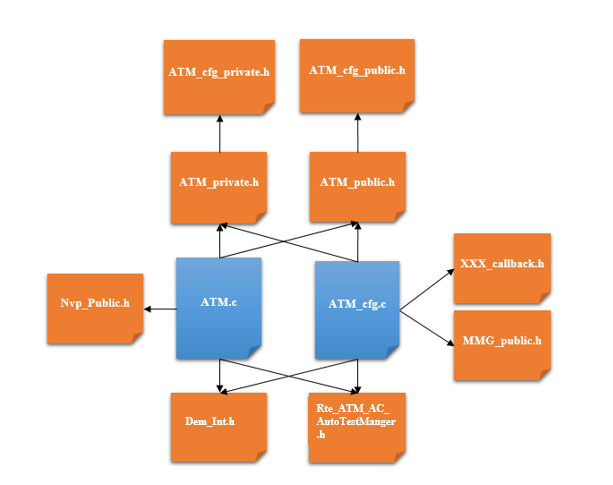


Figure : File structure

**ATM.c**

This file aims at:

* Initializing the whole component: ATM\_Init
* Scheduling auto-tests during startup and run phases: ATM\_runMainFunction
* Reporting test status to DEM: atm\_SetDemEvent
* Checking tests enable conditions: isAutoTestEnabled
* Check if auto test is compliant and execute it: executeAutoTest
* Returning last test execution report: ATM\_runGetTestResult
* Execute startup auto test phase: atm\_PlayOneShotTests
* Execute running auto test phase: atm\_PlayCyclicTests
* Returning the last critical auto test ID: ATM\_runGetLastCriticalAutotestId
* Returning status of the last critical auto test: ATM\_runGetCriticalAutoTestsReachedFlag

**ATM\_cfg.c**

This file contains the base ATM configuration:

* Defines used to configure and schedule each lists
* The configuration of the startup list
* The configuration of the 5 lists used during run time
* A table of structure containing information on each auto-test (period, MMG modes, callback, Event)

**ATM\_private.h**

It will gather the definition of all private constants, functions and variables

**ATM\_cfg\_private.h**

It will gather the definition of all private constants like all inhibiting mode.

**ATM\_public.h**

It will gather the definition of all exported functions

**ATM\_cfg\_public.h**

It will gather the definition of all exported constants

# FEATURES

The purpose of this chapter is to only describe the internal implementation of the component.

For the description of the external implementation, please refer to [Doc1] (to get the list of services, types, variables and constants exported by this unit).

Actually, the description of the internal implementation is not necessary for this SW unit since its complexity is extremely low. Therefore, the current chapter shall not be treated. Then, refer to C-code implementation directly.

## Services

### Exported services

#### ATM\_Init

##### Function overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DSG\_ATM\_00010** | |  | | |
| Type | Design Requirement | | | |
| Status: | Proposed | | | |
| Covers: | ARCH\_SW\_ATM\_0010 | | | |
| Prototype | **void** **ATM\_Init**(**void**) | | | |
| Input Parameters | Name | | Type | Description |
| None | | - | - |
| Output Parameters | Name | | Type | Description |
| None | | - | - |
| Retuned value | Name | | Type | Description |
| None | | - | - |
| Object: | This function is used to initialize the ATM component | | | |
| Dynamic aspect: (Who: Callers) | This function shall be called by the MCU during ECU initialization sequence. | | | |
| Static aspect | Initialize all internal variables. Auto-tests status set to KU8\_ATM\_NOT\_DECIDED | | | |
| Constraint: | This function shall be called first | | | |

##### Function description

This function is in charge of initializing the different global variable of the main function. All autotests status are initialized to the default status KU8\_ATM\_TEST\_NOT\_DECIDED to avoid sending any other status before an autotest has been run.

##### Service diagram

Diagram

Description automatically generated

##### Local variables

|  |  |
| --- | --- |
| Status: | Proposed |
| Link Parent: | DSG\_ATM\_00010 |
| Name: | u8IdxAutotest |
| Object: | Variable used for auto test index. |
| Type: | u8AutoTestIdType |
| Unit: | NA |
| Range: | NA |
| Safety: | NA |
| Constraint: | NA |

#### ATM\_Shutdown

##### Function description

This function is not used in the current project context

##### Service diagram

This function is not used in the current project context

##### Local variables

No variable since the function is not used

#### ATM\_runMainFunction

##### Function overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DSG\_ATM\_00020** | |  | | |
| Type | Design Requirement | | | |
| Status: | Proposed | | | |
| Covers: | ARCH\_SW\_ATM\_0020 | | | |
| Prototype | **void** **ATM\_runMainFunction**(**void**) | | | |
| Input Parameters | Name | | Type | Description |
| None | | None | NA |
| Output Parameters | Name | | Type | Description |
| None | | None | NA |
| Retuned value | Name | |  | Type |
| None | |  | NA |
| Object: | This service oversees the scheduling of all auto-tests during the startup phase and in running mode. | | | |
| Dynamic aspect: (Who: Callers) | This function is called every 2ms in both cases (startup and run) by the SCH | | | |
| Static aspect | This function is split into two parts. One for the startup auto-tests and one for the ECU run phase. The sequence execution of each auto-tests is handled by this function. | | | |
| Constraint: | If a startup auto-test is critical (configured as ONE-SHOT ABORT or ONE SHOT RETRY), if it returns NOK, the whole sequence is frozen, and the function will never enter in the run phase.  If a startup auto-test is critical with ONE SHOT ABORT configuration, since its status is decided OK or NOK, it will never be executed any more.  If a startup auto-test is critical with ONE SHOT RETRY configuration, since its status is NOK, it will be tried again until reaching an OK status. | | | |

##### Function description

This function is periodically called every 2ms. It manages the schedule of autotests during start-up and running phases.

First start-up autotests are run sequentially, and periodic test are run only after start-up sequence.

##### Service diagram

Diagram

Description automatically generated

##### Local variables

None

#### ATM\_runGetTestResult

##### Function overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DSG\_ATM\_00040** | |  | | |
| Type | Design Requirement | | | |
| Status: | Proposed | | | |
| Covers: | ARCH\_SW\_ATM\_0020 | | | |
| Prototype | **void ATM\_runGetTestResult (u8AutoTestIdType** u8AutoTestId**, u8TestResultType \***u8TestResult **)** | | | |
| Input Parameters | Name | | Type | Description |
| u8AutoTestId | | u8AutoTestIdType | ATM auto-test ID provided in ATM\_cfg.c |
| Output Parameters | Name | | Type | Description |
| u8TestResult | | u8TestResultType | Possible values:   * KU8\_ATM\_TEST\_OK * KU8\_ATM\_TEST\_NOK * KU8\_ATM\_NOT\_DECIDED |
| Retuned value | Name | | | Type |
| None | | | NA |
| Object: | This service returns the last test result for a specific auto-test | | | |
| Dynamic aspect: (Who: Callers) | An RTE call is needed to access this service, the link shall be done at architecture level.  This function shall be called by any SW component. | | | |
| Static aspect | This function reads the test status from the local array. If the test ID provided in parameter is not valid, the status ATM\_KU8\_NB\_OF\_AUTOTEST is returned | | | |
| Constraint: | None | | | |

##### Function description

This function allows any module to get the test result (KU8\_ATM\_TEST\_OK, KU8\_ATM\_TEST\_NOK or KU8\_ATM\_NOT\_DECIDED) of any autotest identified by a specific ID.

It gets the last test result independently of the DEM event status.

##### Service diagram

Diagram

Description automatically generated

##### Local Variables

None

#### ATM\_runGetLastCriticalAutotestId

##### Function overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DSG\_ATM\_00050** | |  | | |
| Type | Design Requirement | | | |
| Status: | Proposed | | | |
| Covers: |  | | | |
| Prototype | **void** **ATM\_runGetLastCriticalAutotestId (UInt8** \*u8AutoTestId**)** | | | |
| Input Parameters | Name | | Type | Description |
| None | | NA | NA |
| Output Parameters | Name | | Type | Description |
| U8AutoTestId | | UInt8 | ATM auto-test ID provided ATM\_cfg.c |
| Retuned value | Name | | Type | Description |
| None | | NA | NA |
| Object: | This service returns the last critical Auto-test Id | | | |
| Dynamic aspect: (Who: Callers) | None | | | |
| Static aspect |  | | | |
| Constraint: | None | | | |

##### Function description

This function aims at returning the last critical autotest ID.

##### Service diagram



##### Local Variables

None

#### ATM\_runGetLastCriticalAutotestsReachedFlag

##### Function overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DSG\_ATM\_00060** | |  | | |
| Type | Design Requirement | | | |
| Status: | Proposed | | | |
| Covers: |  | | | |
| Prototype | **void** **ATM\_runGetCriticalAutoTestsReachedFlag** (boolean \* pb8CriticalAutoTestsFlag) | | | |
| Input Parameters | Name | | Type | Description |
| None | | NA | NA |
| Output Parameters | Name | | Type | Description |
| pb8CriticalAutoTestsFlag | | Boolean | Critical auto-test status |
| Retuned value | Name | | Type | Description |
| None | | NA | NA |
| Object: | This service checks the status of the las critical auto test | | | |
| Dynamic aspect: (Who: Callers) | None | | | |
| Static aspect |  | | | |
| Constraint: | None | | | |

##### Function description

This function aims at returning status of the last critical auto test.

##### Service diagram



### Local services

#### atm\_SetDemEvent

##### Function overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DSG\_ATM\_00060** | |  | | |
| Type | Design Requirement | | | |
| Status: | Proposed | | | |
| Link parents: |  | | | |
| Covers: |  | | | |
| Prototype | **void** **atm\_SetDemEvent (**const **uint8** cu8AutotestId, **Dem\_EventStatusType** EventStatus**)** | | | |
| Input Parameters | Name | | Type | Description |
| cu8AutotestId | | uint8 | Index of the test in the internal configuration array |
| EventStatus | | Dem\_EventStatusType | Status of the autotest to be set |
| Output Parameters | Name | | Type | Description |
| None | | None | NA |
| Retuned value | Name | | | Type |
| None | | | NA |
| Object: | Set EventStatus for a specific auto-test | | | |
| Dynamic aspect: (Who: Callers) | This function shall be Called every time a test needs to be executed by ATM | | | |
| Static aspect |  | | | |
| Constraint: | None | | | |

##### Function description

The goal of this function is to set via the Dem, the status an event defined by the parameter *EventStatus* for its related autotest.

This shall be applied in case of the autotests present in the castATMConfig list.

##### Service diagram

A picture containing computer

Description automatically generated

##### Local variables

None

#### executeAutoTest

##### Function overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DSG\_ATM\_00070** | |  | | |
| Type | Design Requirement | | | |
| Status: | Proposed | | | |
| Link parents: |  | | | |
| Covers: |  | | | |
| Prototype | **void** **executeAutoTest (**const **uint8** cu8AutotestId**)** | | | |
| Input Parameters | Name | | Type | Description |
| cu8AutotestId | | uint8 | Index of the test in the internal configuration array |
| Output Parameters | Name | | Type | Description |
| None | | None | NA |
| Retuned value | Name | | | Type |
| None | | | NA |
| Object: | Execute a test by calling its callback. Send the test status to DEM | | | |
| Dynamic aspect: (Who: Callers) | This function shall be Called every time a test needs to be executed by ATM | | | |
| Static aspect | Notify the DEM according to the test result. | | | |
| Constraint: | None | | | |

##### Function description

This is a sub-function of the ATM\_runMainFunction(). The goal of this function is to check if an autotest is compliant to the ECU mode got through the autotest system context.

* When compliant, the autotest is executed. After the autotest execution this function updates its related event status according to the test result.
* When not compliant, the autotest is simply ignored

##### Service diagram

Diagram

Description automatically generated

##### Local variables

|  |  |
| --- | --- |
| Status: | Proposed |
| Link Parent: | DSG\_ATM\_00070 |
| Name: | u8ReturnedAutoTestStatus |
| Object: | Variable used to store the autotest results before notification and storage |
| Type: | u8TestResultType |
| Unit: | NA |
| Range: | [0; 255] |
| Dynamic aspect: | NA |
| Constraint: | None |

|  |  |
| --- | --- |
| Status: | Proposed |
| Link Parent: |  |
| Name: | u8InhibModesIndex |
| Object: | Index of the inhibiting modes combination |
| Type: | uint8 |
| Unit: | NA |
| Range: | [0; 255] |
| Dynamic aspect: | NA |
| Constraint: | None |

#### isAutoTestEnabled

##### Function overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DSG\_ATM\_00080** | |  | | |
| Type | Design Requirement | | | |
| Status: | Proposed | | | |
| Component: | ATM | | | |
| Link parents: |  | | | |
| Covers: | ARCH\_SW\_ATM\_0010 | | | |
| Prototype | **boolean** **isAutoTestEnabled (const** uint8 cu8AutotestId**)** | | | |
| Input Parameters | Name | | Type | Description |
| cu8AutotestId | | Uint8 | Index of the test in the internal configuration array |
| Output Parameters | Name | | Type | Description |
| isAutoTestEnabled | | boolean |  |
| Retuned value | Name | | Type | Type |
| bReturn | | boolean | * B\_TRUE when the test is enabled and can be executed * B\_FALSE otherwise |
| Object: | This service check if an auto-test can be executed | | | |
| Dynamic aspect: (Who: Callers) | This function shall be Called every time a test needs to be executed by ATM | | | |
| Static aspect | Check if a callback is configured and if the test is enabled by configuration. When a test is disabled, the DEM is notified with a PASSED state to avoid freezing the complete sequence. | | | |
| Constraint: | Test ID must be in the correct range. | | | |

##### Function description

This is a sub-function of the ATM\_runMainFunction(). It aims to check if a test is enabled according to 2 criteria :

* the presence of a no null call back function in the configuration
* the NVP configuration

In the last case, the event status is updated as if the test was passed.

NB: it is possible to enable or disable a test via two ways:

* *From configuration:*

Sometimes during certain integration tests campaign, it is necessary to disable a test which will qualify a fault before the one you want to check. To do this, an NVP block is used: *NVP\_au8AutoTestsActivation:*

* A test is enabled if its state = 0xAA
* A test is disabled if its state = 0x55

The correspondence between the table index and the test ID can be found in the file ATM\_cfg.c.

* *During run time*:

It happens that all conditions are not fulfilled to execute a test. A wrong condition could be one of these:

* Battery not stable
* Car cranking situation
* Motor activation
* Etc.

##### Service diagram

Diagram

Description automatically generated

##### Local variables

|  |  |
| --- | --- |
| Status: | Proposed |
| Link Parent: |  |
| Name: | bReturn |
| Object: | Return the result |
| Type: | boolean |
| Unit: | NA |
| Range: | B\_TRUE  B\_FALSE |
| Dynamic aspect: | None |
| Constraint: | None |

#### atm\_PlayOneShotTests

##### Function overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DSG\_ATM\_0001** | |  | | |
| Type | Design Requirement | | | |
| Status: | Proposed | | | |
| Component: | ATM | | | |
| Link parents: |  | | | |
| Covers: | ARCH\_SW\_ATM\_0010 | | | |
| Prototype | **void** **atm\_PlayOneShotTests (void)** | | | |
| Input Parameters | Name | | Type | Description |
| None | | - | - |
| Output Parameters | Name | | Type | Description |
| None | | - | - |
| Retuned value | Name | | Type | Type |
| None | | - | - |
| Object: | Sub function of ATM\_runMainFunction  Treats the “one shot” sequence according to the table passed as argument which concerns either the start-up cyclic tests, or the running cyclic tests. | | | |
| Dynamic aspect: (Who: Callers) | ATM\_runMainFunction | | | |
| Static aspect | None | | | |
| Constraint: | The table passed as argument is expected to be constant. | | | |

##### Function description

This is a sub-function of the ATM\_runMainFunction(). Its goal is to execute autotests during startup phase.

ATM module implements two different behavior: “one shot” and “cyclic” executions.

During the startup phase, “critical” auto-tests are executed once sequentially (“one shot” execution). If one of them fails, the whole startup sequence is frozen and the RUN phase is not reachable. However, when all startup auto-test are passed, the RUN state is entered.

The “one shot” execution list contains all auto-tests that need to be executed before entering the run phase. Only one test is executed by function call. The function is called every 2ms.

This list configuration is stored respectively in *castStartupSequentialList* located in ATM\_cfg.c. It is described in chapter 3.4

* *Critical autotests*

The sequence defined by the “one shot” list contains CRITICAL auto tests: they are the auto tests which are in the *castStartupSequentialList*. They are configured as ONE SHOT, ONE SHOT DECIDED, ONE SHOT ABORT or ONE SHOT RETRY.

The test order is important because the next test is called only if the previous one was passed. They are executed sequentially. When a test fails during this sequence, two strategies can be applied:

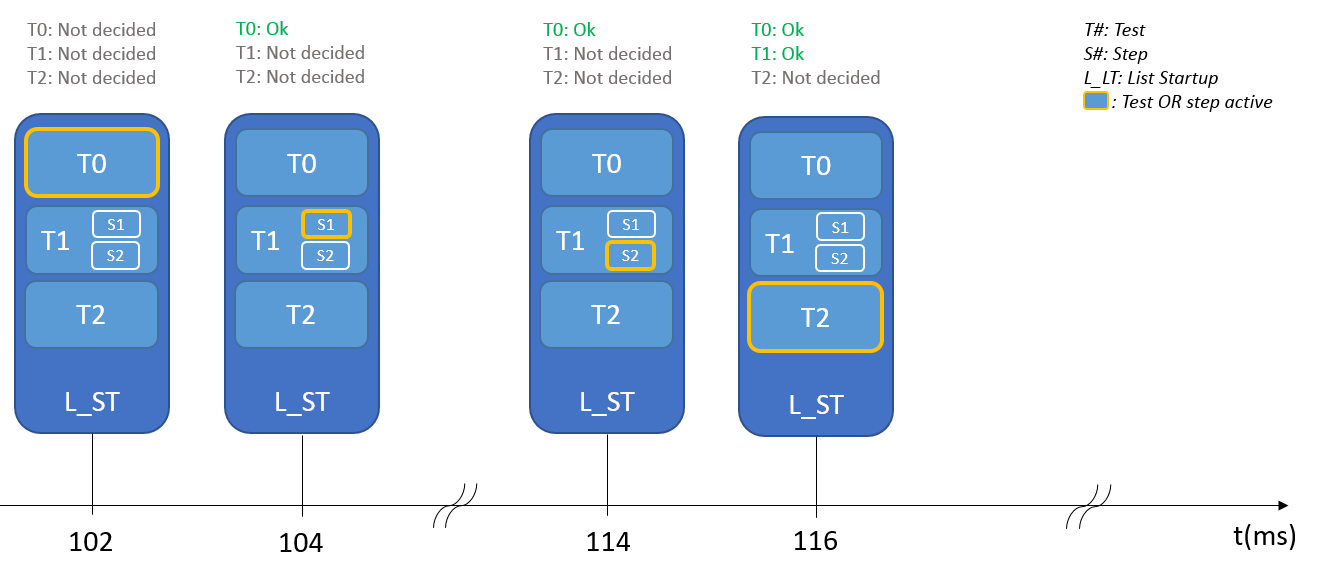
* The test is configured as a ONE SHOT ABORT auto-test: if its status is NOK, the test is never played again and the whole sequence is stopped.
* The test is configured as a ONE SHOT RETRY auto-test: since its status is still NOK, the test is replayed again on the next task occurrence. Some of these tests are designed to be stopped until a maximum number of tries, but this mechanism is designed inside the algorithm of the tests.

Sometimes, a test needs more than one call to be fully passed. In this case, all steps are scheduled on their correct period derived from the 2ms cyclic call. When the last step is executed, the state machine moves to the next test.

* The test is configured as a ONE SHOT auto-test: the test is played only once, so the next test is played
* The test is configured as a ONE SHOT DECIDED auto-test: the test is not retried, so the next test is played

A brief example of a startup list containing 3 tests is presented below:

* Test 0 and 2 are one shot test
* Test 1 is composed of 2 steps separated by 10ms.



##### Service diagram

Diagram

Description automatically generated

##### Local variables

|  |  |
| --- | --- |
| Status: | Proposed |
| Link Parent: |  |
| Name: | u8AutotestID |
| Object: | ID of the autotest to be executed |
| Type: | uint8 |
| Unit: | NA |
| Range: | [0; 255] |
| Dynamic aspect: | None |
| Constraint: | None |

#### atm\_PlayCyclicTests

##### Function overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DSG\_ATM\_0001** | |  | | |
| Type | Design Requirement | | | |
| Status: | Proposed | | | |
| Component: | ATM | | | |
| Link parents: |  | | | |
| Covers: | ARCH\_SW\_ATM\_0010 | | | |
| Prototype | **void** **atm\_PlayCyclicTests (stCyclicListEltType \***cpstCyclicList**)** | | | |
| Input Parameters | Name | | Type | Description |
| cpstCyclicList | | stCyclicListEltType | Head (first element) of the array containing the 5 sub-list to be played. |
| Output Parameters | Name | | Type | Description |
| None | | - | - |
| Retuned value | Name | | Type | Type |
| None | | - | - |
| Object: | Sub function of ATM\_runMainFunction  Treats the cyclic auto test scheduling according to the table passed as argument which concerns either the start-up cyclic tests, or the running cyclic tests. | | | |
| Dynamic aspect: (Who: Callers) | ATM\_runMainFunction | | | |
| Static aspect | None | | | |
| Constraint: | The table passed as argument is expected to be constant. | | | |

##### Function description

This is a sub-function of the ATM\_runMainFunction(). Its goal is to execute autotest from cyclic lists.

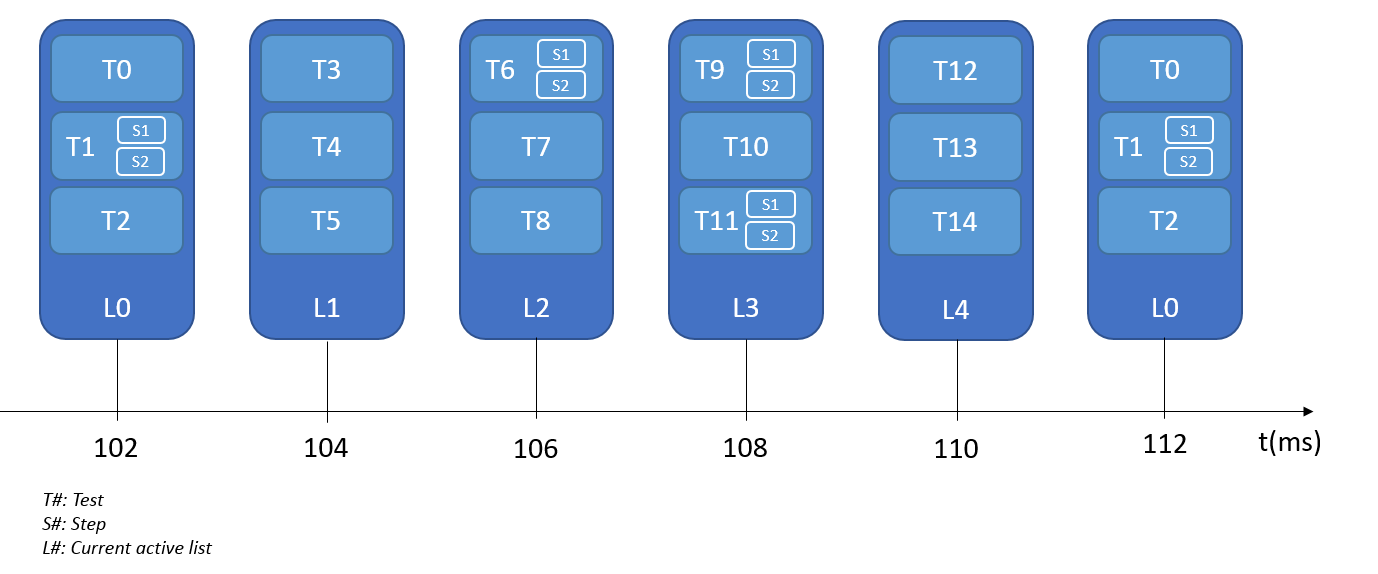
Two cyclic lists coexist, to be played respectively during the “one shot” sequence, and after the “one shot” sequence completion in the case where all critical auto tests result could be “passed”.

The “cyclic” execution is performed during both startup and run phases. Each phase uses its own set of test scheduling data.

The first list is necessary to identify a fault explaining why a critical “one shot” test has failed (eg. Low voltage being monitored as cyclic test in parallel with critical autotests). The second list includes also tests that require the critical auto tests to be passed first.

Each periodic list is divided into five sub-lists to cover 10ms and more test period. This cyclic execution design allows to better share the CPU load between 5 execution points by 10ms windows.

The execution order is fixed and cannot be changed.



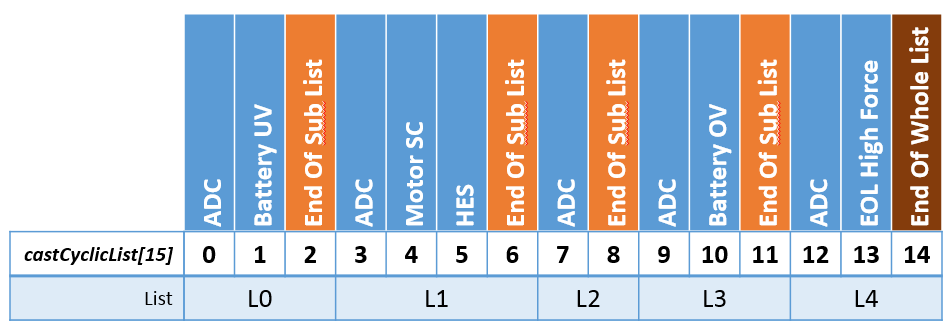
These list configurations are stored respectively in *castStartupCyclicList* and *castRunningCyclicList* located in ATM\_cfg.c. They are described in chapter 3.4

These arrays are split into five sub-lists separated by a delimiter. They are passed as argument to the atm\_PlayCyclicTests function. Thanks to the sub-list separator, atm\_PlayCyclicTests only loops through one table and do not have to switch between five.

Two types of delimiter are used:

* KU8\_END\_OF\_SUB\_LIST is used between lists. It has an offset of 1 indication the ATM main function to continue with its current indexes.
* KU8\_END\_OF\_THE\_WHOLE\_LIST terminate the list. It has an offset of 0 indicating the ATM main function to loop back to the first list and reset all its indexes.

Tests shall be added between these delimiters like in the diagram presented below:



As the delimiters cannot be removed, total array size shall be increased by 5:

##### Service diagram

Diagram

Description automatically generated

##### Local variables

|  |  |
| --- | --- |
| Status: | Proposed |
| Link Parent: |  |
| Name: | u8AutotestID |
| Object: | ID of the autotest to be executed |
| Type: | uint8 |
| Unit: | NA |
| Range: | [0; 255] |
| Dynamic aspect: | None |
| Constraint: | None |

## Types

### Global types

### Local types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Status: | | Proposed | | |
| Link Parent: | | DSG\_ATM\_00020 | | |
| Name | Structure | | Object | Constraints |
| stCyclicListEltType | uint16 u16Period | | Test period in number of ATM execution occurrences | NA |
| uint8 u8Offset | | offset (in number of ATM execution occurrences) | NA |
| uint8 u8TestIndex | | Test Index |  |
| stStartupEltType | uint16 u16StepPeriod | | Period (in number of ATM execution occurrences) | NA |
| uint8 u8Type | | Autotest type | NA |
| uint8 u8TestIndex | | Test Index | NA |

## Variables

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Status: | | Proposed | | | | |
| Link Parent: | | DSG\_ATM\_00040 | | | | |
| Name | Object | | Type | Range | Dynamic aspect | Constraints |
| au8AtStatuses | This array contains the status of each test in RAM | | uint8 | KU8\_ATM\_TEST\_OK  KU8\_ATM\_TEST\_NOK  KU8\_ATM\_NOT\_DECIDED | This array is written only by ATM\_runTest to update the auto-test status.  This table is read by a call to ATM\_runGetTestResult. The call to this function is done asynchronously. | None |
| u16StartupPeriodCnt | Counter used to count the period between each step of a test | | uint16 | [0 ; 65535] | During the startup phase, the mainfunction is called every 2ms, this counter is used to create the correct period call for each test step. | None |
| u8CurrentTask | Counter indicating which task needs to be activated | | uint8 | [0 ; 4] | Evaluated during each mainfunction call in the RUN phase | None |
| u16CurrentDecade | Decade counter used to dispatch the CPU load on a current timeslot | | uint16 | [0 ; 59999] | Circular counter, incremented by 1 at the end of the whole list | None |
| u8StartupTestIndex | Index used in “castStartupList” during the startup test sequence | | uint8 | [0 ; 255] | Incremented by one after each new executed test | Shall be equal to 15 when all startup auto-test are passed |
| u8CyclicTestIndex | Index used in “castCyclicList” during the RUN sequence | | uint8 | [0 ; 255] | Incremented by one at the end of a test | Reset when the last test of the list is played |

## 

## Constants

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Status: | | Proposed | | | | |
| Link Parent: | | DSG\_ATM\_00040 | | | | |
| Name | Object | | Type | Value | Unit | Constraints |
| castATMConfig | Table of structure containing the configuration of each auto-test | | stAutoTestConfigurationType | See ATM\_cfg.c | NA | All members shall be filled with correct values |
| cau32InhibitingModes | Table of inhibition modes bit-fields combinations used in the configuration above. | | u32ModeMaskType | See ATM\_cfg.c | NA | Each element shall be different to each other to preserve memory space. |
| castStartupSequentialList | Execution sequence and configuration of one shot auto-test | | stStartupEltType | See ATM\_cfg.c | NA | No empty member allowed |
| castStartupCyclicList | Configuration of the five sub-lists used for cyclic execution of tests during the startup phase | | stCyclicListEltType | See ATM\_cfg.c | NA | No empty member allowed.  Sub list delimiter shall not be moved or removed |
| castRunningCyclicList | Configuration of the five sub-lists used for cyclic execution of tests during the running phase | | stCyclicListEltType | See ATM\_cfg.c | NA | No empty member allowed.  Sub list delimiter shall not be moved or removed |
| ATM\_KU8\_NB\_OF\_AUTOTEST | Number of auto tests | | uint8 | 36U | NA | NA |
| KU8\_ATM\_TEST\_NON\_EXISTENT | Value returned in case of a nonexistent ID of an autotest | | uint8 | 8U | NA | NA |
| KU8\_ZERO | Macro used for value 0 | | uint8 | 0x00 | NA | NA |
| KU16\_ZERO | Macro used for value 0 | | uint16 | 0x00 | NA | NA |
| KU16\_ONE |  | | uint16 | 0x0001 | NA | NA |
| KU16\_MAX |  | | uint16 | 65535 | NA | NA |
| KU16\_CFG\_MAX\_DECADE | Event status is Prepassed | | uint16 | 60000U | NA | NA |
| KU16\_CFG\_MAX\_TASK |  | | uint16 | 5U | NA | NA |
| KU8\_ATM\_TEST\_OK | Auto test status when test is OK | | uint8 | 1U | NA | NA |
| KU8\_ATM\_TEST\_NOK | Auto test status when test is NOK | | uint8 | 2U | NA | NA |
| KU8\_ATM\_TEST\_NOT\_DECIDED | Auto test status when test is not complete | | uint8 | 4U | NA | NA |
| KU8\_MODE\_STATUS\_ON |  | | uint8 | 170U | NA | NA |
| ATM\_KU8\_NB\_OF\_STARTUP\_AUTO\_TESTS | Number of startup auto tests | | uint8 | 21U | NA | NA |
| KU16\_CFG\_MAX\_TASK | Auto test status | | uint8 | 5U | NA | NA |
| B\_TRUE |  | | boolean | 0xAA | NA | NA |
| B\_FALSE |  | | boolean | 0x55 | NA | NA |
| ATM\_KU8\_ID\_TEST\_PRELIMINARY\_BOOST | Id of the Preliminary boost autotest | | uint8 | 8U | NA | NA |
| DEM\_EVENT\_STATUS\_PREPASSED | Event status is prepassed | | uint8 | 2U | NA | NA |
| DEM\_EVENT\_STATUS\_PREFAILED |  | | uint8 | 3U | NA | NA |
| KU8\_AUTO\_TEST\_CONFIG\_TYPE\_ONE\_SHOT\_ABORT |  | | uint8 | 0x02 | NA | NA |
| KU8\_AUTO\_TEST\_CONFIG\_TYPE\_ONE\_SHOT |  | | uint8 | 0x08 | NA | NA |
| KU8\_AUTO\_TEST\_CONFIG\_TYPE\_ONE\_SHOT\_DECIDED |  | | uint8 | 0x10 | NA | NA |
| KU8\_ATM\_TEST\_BAD\_CONDITIONS |  | | uint8 | 16U | NA | NA |
| ATM\_KU8\_ID\_TEST\_ADC | Number of auto tests | | uint8 | 0U | NA | NA |
| ATM\_KU8\_ID\_TEST\_HIGH\_SIDE\_SWITCH | High side switch autotest ID | | uint8 | 1U | NA | NA |
| ATM\_KU8\_ID\_TEST\_DRIVER\_COMMAND | Driver command autotest ID | | uint8 | 2U | NA | NA |
| ATM\_KU8\_ID\_TEST\_MOSFET\_HIGH\_SHORT\_CIRCUIT | Mosfet High SC autotest ID | | uint8 | 3U | NA | NA |
| ATM\_KU8\_ID\_TEST\_MOSFET\_LOW\_SHORT\_CIRCUIT | Mosfet Low SC autotest ID | | uint8 | 4U | NA | NA |
| ATM\_KU8\_ID\_TEST\_MOTOR\_DISCONNECTION | Motor disconnection autotest ID | | uint8 | 5U | NA | NA |
| ATM\_KU8\_ID\_TEST\_DRIVER\_SELF\_PROTECTION | HW self-protection autotest ID | | uint8 | 6U | NA | NA |
| ATM\_KU8\_ID\_TEST\_MOSFET\_OPEN\_CIRCUIT | Mosfet OC autotest ID | | uint8 | 7U | NA | NA |
| ATM\_KU8\_ID\_TEST\_PRELIMINARY\_BOOST | Preliminary boost autotest ID | | uint8 | 8U | NA | NA |
| ATM\_KU8\_ID\_TEST\_VBOOST | Boost autotest ID | | uint8 | 9U | NA | NA |
| ATM\_KU8\_ID\_PFLASH\_MEMORY\_CORRUPTION | P\_Flash autotest ID | | uint8 | 10U | NA | NA |
| ATM\_KU8\_ID\_TEST\_HALL\_EFFECT\_SENSORS | Hall Effect sensors autotest ID | | uint8 | 11U | NA | NA |
| ATM\_KU8\_ID\_TEST\_MOTOR\_CURRENT | Motor current autotest ID | | uint8 | 12U | NA | NA |
| ATM\_KU8\_ID\_TEST\_MOTOR\_BLOCKED | Motor blocked autotest ID | | uint8 | 13U | NA | NA |
| ATM\_KU8\_ID\_TEST\_MOTOR\_TEMPERATURE | SW self-protection autotest ID | | uint8 | 14U | NA | NA |
| ATM\_KU8\_ID\_TEST\_SENSOR\_TEMPERATURE | Temperature sensor autotest ID | | uint8 | 15U | NA | NA |
| ATM\_KU8\_ID\_TEST\_MOTOR\_SHORT\_CIRCUIT | Motor SC autotest ID | | uint8 | 16U | NA | NA |
| ATM\_KU8\_ID\_TEST\_PWM\_ORDER | Motor order autotest ID | | uint8 | 17U | NA | NA |
| ATM\_KU8\_ID\_TEST\_UNDER\_VOLTAGE\_POWER | UV power autotest ID | | uint8 | 18U | NA | NA |
| ATM\_KU8\_ID\_TEST\_OVER\_VOLTAGE\_POWER | OV power autotest ID | | uint8 | 19U | NA | NA |
| ATM\_KU8\_ID\_TEST\_UNDER\_VOLTAGE\_TENSIONING\_POWER | UV tensioning autotest ID | | uint8 | 20U | NA | NA |
| ATM\_KU8\_ID\_TEST\_OVER\_VOLTAGE\_LOGIC | OV logic autotest ID | | uint8 | 21U | NA | NA |
| ATM\_KU8\_ID\_TEST\_UNDER\_VOLTAGE\_LOGIC | UV logic autotest ID | | uint8 | 22U | NA | NA |
| ATM\_KU8\_ID\_TEST\_IMPLAUSIBLE\_BATTERY\_LEVEL | Implausible battery level autotest ID | | uint8 | 23U | NA | NA |
| ATM\_KU8\_ID\_NVM\_NOT\_PROG | NVM\_NOT\_PROG autotest ID | | uint8 | 25U | NA | NA |
| ATM\_KU8\_ID\_NVM\_MEMORY\_CORRUPTION | NVM\_Memory\_Corruption autotest ID | | uint8 | 26U | NA | NA |
| ATM\_KU8\_ID\_RAM\_MEMORY\_CORRUPTION | RAM\_Memory\_Corruption autotest ID | | uint8 | 27U | NA | NA |
| ATM\_KU8\_ID\_RESET\_CAUSE | Warm reset autotest ID | | uint8 | 29U | NA | NA |
| ATM\_KU8\_ID\_END\_OF\_LIFE\_LOW\_FORCE | EOL low force autotest ID | | uint8 | 30U | NA | NA |
| ATM\_KU8\_ID\_END\_OF\_LIFE\_HIGH\_FORCE | EOL high force autotest ID | | uint8 | 31U | NA | NA |
| ATM\_KU8\_ID\_END\_OF\_LIFE\_COMFORT | EOL comfort autotest ID | | uint8 | 32U | NA | NA |
| ATM\_KU8\_ID\_END\_OF\_LIFE\_HAPTIC\_WARNING | EOL haptic warning autotest ID | | uint8 | 33U | NA | NA |
| ATM\_KU8\_ID\_TEST\_BATTERY\_SURVEY | Battery survey autotest ID | | uint8 | 34U | NA | NA |
| ATM\_KU8\_ID\_TEST\_ADC\_CALIB | ADC calibration autotest ID | | uint8 | 35U | NA | NA |
| ATM\_KU8\_ID\_EXT\_WDG\_OUT\_OF\_ORDER | Ext watchdog autotest ID | | uint8 | 24U | NA | NA |

# 

# EEPROM

Values of all NVP parameters presented below can be found in [Doc3].

* Table used to enable/disable auto-test
  + NVP\_au8AutoTestsActivation

The EEPROM parameters are all specified in [Doc2].

Refer to this document for more details.

# Configuration

See ATM\_cfg.c

# Compilation Options

None