Startup and Shutdown Sequence

Document Version: 1.0.0

Kindly refer the below mentioned points before you start with this document

* In the OS Configuration, Kindly configure an Autostart Task. Since some of the Functionalities of startup in this documents depends on this Task.
* This below mentioned startup or shutdown sequence is not full-fledged for all the modules.
* User can add the BSWM MODES , Rules , Action Items to match with the customer requirements. Following the below methods completely is not mandatory.

1. The Startup Sequence shall be executed Considering the **main()** function.
2. Within this following Functionalities shall be executed:
   1. MCU\_INIT ();**🡪 Mandatory.**
   2. PORT\_INIT (); 🡪**Mandatory.**
   3. DIO\_INIT ();🡪**Mandatory.**
   4. WATCHDOG\_INIT (); 🡪Optional.
   5. ECUM\_INIT ();🡪**Mandatory.**
3. Within the ECUM\_INIT(), we have mainly two functions:
   1. **EcuM\_AL\_DriverInitZero()** 🡪Added according to ECUM AUTOSAR Specification
   2. **EcuM\_AL\_DriverInitOne()🡪**Added according to ECUM AUTOSAR Specification
4. **EcuM\_AL\_DriverInitZero()** shall contain a callout function called:
   1. **EcuM\_Gen\_AL\_DriverInitZero()**

NOTE: The definition for **EcuM\_Gen\_AL\_DriverInitZero()** should be configured and generated in ECUM by the Configurator.

**EcuM\_Gen\_AL\_DriverInitZero()** shall basically contain as mentioned below:

* **Icu\_Init(NULL\_PTR) 🡪 Optional.**
* **IO\_Init()🡪 Optional.**

1. **EcuM\_AL\_DriverInitOne()** shall contain a callout function called:

**EcuM\_Gen\_AL\_DriverInitOne()**

NOTE: The definition for **EcuM\_Gen\_AL\_DriverInitOne()** should be configured and generated in ECUM by the Configurator.

* **Dem\_PreInit() 🡪 Mandatory**
* **Det\_Start() 🡪 Optional.**

1. After **EcuM\_AL\_DriverInitOne() ,** EcuM will call the below mentioned function:
   1. **StartOS()**
2. After the **StartOS()** is executed, the Autostart Task (TASK(OsTask\_AutoStart) )will be the first task to be executed and will have the functionality as mentioned below:
   1. Adc\_Init(Adc\_AdcConfigSet0); 🡪**Optional**
   2. FreeTimer\_Init();🡪**Optional**
   3. config\_select();**🡪Optional** : Used for selecting the Post Build configuration.

Note : This is also possible from ECUM Module.

* 1. **EcuM\_StartupTwo();** **🡪 Mandatory.**
  2. Cdd\_Init() 🡪 **Optional.**
  3. **Rte\_Start() 🡪Mandatory.**

1. In **EcuM\_StartupTwo()** the following Functionalities are done:
   1. **Schm\_Init() ->** Added according to ECUM AUTOSAR Specification
   2. **BswM\_Init() ->** Added according to ECUM AUTOSAR Specification

The **Schm\_Init()** will have the function calls to activate the OS Tasks and generated definition is generated by RTE.

* Example : setrelalarm(arguments) or setabsalarm(arguments)

NOTE: After the BSWM\_Init is done , BSWM will take control over Initializing the AUTOSAR COM STACK(Can, Canif, PduR, Com, IpduM), Memory STACK(NvM, Fee, Fls, EEL Drivers).A complete description will be provided in further description below.

1. After the Autostart Task Start is finalized. The Periodic tasks triggered by their alarms in **Schm\_Init()** shall start executing.
2. On **Rte\_Start()** the Initial State of the MODE SWITCH will be inBSWM \_STATE\_STARTUP\_ONE

BSWM Modes to be configured is as mentioned below:

* **BSWM \_STATE\_STARTUP\_ONE**
* **BSWM\_STATE\_STARTUP\_TWO**
* **BSWM\_STATE\_RUN**
* **BSWM\_STATE\_APP\_RUN**
* **BSWM\_STATE\_APP\_RUN**,
* **BSWM\_STATE\_APP\_POST\_RUN**
* **BSWM\_STATE\_PREP\_NO\_COMM.**
* **BSWM\_STATE\_NO\_COMM**
* **BSWM\_STATE\_PREP\_GODOWN.**

NOTE: Additional States can be added based on user requirements.

1. On **BSWM \_STATE\_STARTUP\_ONE** mode condition of **Rule\_ BSWM \_STATE\_STARTUP\_ONE** the following Autosar Modules shall be initialized as part of Action List Items:
   1. Can\_Init()
   2. Canif\_Init()
   3. NvM\_Init()
   4. Fee\_Init()
   5. Pdur\_Init()
   6. CanSM\_Init()
   7. ComM\_Init()
   8. IpduM\_Init()
   9. Com\_Init()
   10. CanTp\_Init()
   11. Dcm\_Init()
   12. **NvM\_ReadAll()**
   13. Schm\_Switch\_<NAME>( **BSWM\_STATE\_STARTUP\_TWO**)

NOTE: Application related initializations can also be done here or provide a Mode information for Application to Initialize it Modules.

1. In **Rule\_ BSWM \_STATE\_STARTUP\_TWO** the following Autosar Modules shall be initialized as part of Action List Items:
   1. Dem\_Init()🡪 Optional
   2. Dem\_SetOperationCycleState(ID,START).
   3. Schm\_Switch\_<NAME>(**BSWM\_STATE\_RUN**)
2. Action list items of **Rule\_ BSWM \_STATE\_STARTUP\_TWO** shall be executed only if the Current BSWM Mode is BSWM\_STATE\_STARTUP\_TWO **AND** **NvM\_ReadAll** process is completed (Either E\_OK or E\_NOT\_OK).

NOTE: Applications which are dependent on EEPROM related Information can be initialized in Action list items of **Rule\_ BSWM \_STATE\_STARTUP\_TWO**.

NOTE: If NvM is not present in the Stack , then “**NvM\_ReadAll** process is completed (Either E\_OK or E\_NOT\_OK)” condition can be avoided.

1. In **Rule\_ BSWM \_STATE\_RUN** shall be used to Request ComM for Full communication as part of Action List Items:
   1. ComM\_CommunicationAllowed(ChannelID,TRUE(1))
   2. ComM\_RequestComMode(ChannelID, FULL\_COMMUNICATION)
   3. Schm\_Switch\_<NAME>(**BSWM\_STATE\_APP\_RUN**)
2. Action list items of **Rule\_ BSWM \_STATE\_RUN** shall be executed only if the Current BSWM Mode is **BSWM\_STATE\_RUN.**
3. The **Rule\_BSWM\_STATE\_APP\_RUN** shall be used to Enable the Tx and Rx IPDU Groups with respect to COM as part of the Action List:
   1. ENABLE TX RX IPDU Groups of COM Module.
   2. Schm\_Switch\_<NAME> (**BSWM\_STATE\_APP\_POST\_RUN**).
4. Action List items of **Rule\_BSWM\_STATE\_APP\_RUN** shall be executed only if the Current Mode is **BSWM\_STATE\_APP\_RUN.**

The Shutdown Sequence shall be executed followed by a trigger from an external environment as mentioned below:

* 1. Diagnostic Request with an ECU Reset or a Jump to Boot.
  2. Application has initiated a Shutdown.

1. In **Rule\_ BSWM\_STATE\_PREP\_NO\_COMM** the following Action Items shall be executed:
   1. DISABLE TX and RX Ipdu Groups.
   2. ComM\_CommunicationAllowed(ChannelID,FALSE(0))
   3. ComM\_RequestComMode(ChannelID, NO\_COMMUNICATION)
   4. Schm\_Switch\_<NAME>( **BSWM\_STATE\_NO\_COMM**)
2. The BSWM Rule **Rule\_ BSWM\_STATE\_PREP\_NO\_COMM** shall be executed onlywhen the Mode Changes to **BSWM\_STATE\_PREP\_NO\_COMM** from Application **OR** Dcm Mode has been changed to DCM\_ECURESET or JUMPTOBOOT.
3. The BswM Rule **Rule\_ BSWM\_STATE\_NO\_COMM** shall have the following Action Items:
   1. ComM\_DeInit()
   2. Dem\_SetOperationCycleState(ID,END)
   3. Dem\_Shutdown()
   4. NvM\_WriteAll()
   5. Schm\_Switch\_<NAME>( **BSWM\_STATE\_PREP\_GODOWN)**
4. Rule **Rule\_ BSWM\_STATE\_NO\_COMM** shall be executed only when the Current BSWM Mode is **BSWM\_STATE\_NO\_COMM.**
5. The BswM Rule **Rule\_ BSWM\_STATE\_PREP\_GODOWN** shall have the following Action Items:
   1. Rte\_Stop()
   2. **ECUM\_SelectShutdownTarget(TARGET, MODE)**
   3. EcuM\_GoDown(caller).
6. The **Rule\_ BSWM\_STATE\_PREP\_GODOWN** shall be executed only when BswM Mode has been changed to **BSWM\_STATE\_PREP\_GODOWN AND NvM\_Writeall** Process has been completed with E\_OK or E\_NOT\_OK.
7. In the ECUM\_GoDown, the following Functionalities are executed:
   1. BswM\_DeInit() 🡪Added according to ECUM AUTOSAR Specification
   2. SchM\_DeInit() 🡪Added according to ECUM AUTOSAR Specification
   3. ShutdownOS 🡪Added according to ECUM AUTOSAR Specification

NOTE:

* After this Rule, all the Tasks are stopped by EcuM\_GoDown() which calls SchM\_DeInit()
* SchM\_DeInit will cancel all the OS alarms.
* EcuM will take control from BswM.

1. ShutdonwOS will have a call to Os\_ShutdownCore.
2. Os\_ShutdownCore will call shutdownHook in which we call the below function:
   1. EcuM\_Shutdown()
3. In EcuM\_Shutdown(), based on the MODE value of **ECUM\_SelectShutdownTarget()**

I.e ECUM\_STATE\_RESET or ECUM\_STATE\_OFF,below mentioned callbacks are provided by ECUM Module:

* 1. EcuM\_AL\_SwitchOff() 🡪 for ECUM\_STATE\_OFF Mode value
  2. EcuM\_AL\_Reset() 🡪 for ECUM\_STATE\_RESET Mode value

1. In EcuM\_AL\_Reset() , Mcu\_Reset() function will be executed which in turn resetting the ECU.

NOTE: This document is a Draft version and includes on with respect to Basic Startup and Shutdown.

The Following Points are still pending:

1. BSWM Rules for NM Interaction is not available.
2. Diagnostic Interaction with BSWM is not available.
3. …….