LGE VC [Jaguar LandRover] [TCU4 VDC]

**Software Detailed Design**

[Special Mode Application]

***Notice:***

***Information contained in this document is classified LG Confidential Proprietary. No person outside the LG Group shall have access to the information contained in this document unless business needs dictate, otherwise. It is the responsibility of the person knowing the information contained in this document to ensure confidentiality of information contained in it and securing unauthorized access to this document at all times.***

About This Document

Document Information

|  |  |
| --- | --- |
| Issuing authority | LGE VC Telematics Team |
| Configuration ID | LGE\_JLR\_VDC\_SDD\_SpecialModeApplication |
| **Status of document** | In Progress / Approved / Released 중 선택 |

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Comment | Author | Approver |
| 0.1 | 2017-12-22 | Initial Release | luc.nguyen | tap.pham |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Purpose

This document specifies the Software Detailed Design (SDD) for the Special Mode application including the static design, dynamic design, and algorithm design.

Scope

This document identifies the class consisting of each component from SAD, and describes the behaviors of those classes to accomplish the requirements upon them.

Audience

The target audience of this document is:

* Software architect who will evaluate the design of the software
* Component developer who will implement the design in actual code
* Native application developers who need to interact with Special Mode application
* JLR TCU4 VDC project participants who want to understand the low level design of the Special Mode application
* Test engineers who verify this component

Related Documents

* SAD (Software Architectural Design)

Conventions

This document remarks “NOTE” as follows

NOTE

Prefix: SM\_ ~ Special Mode

Ex: SMApplication means SpecialModeApplication

Prefix: DM\_ ~ Demo Mode

Ex: DMBaseProcess means DemoModeBaseProcess

Acronyms

|  |  |
| --- | --- |
| Acronym | Description |
| SAD | Software Architectural Design |
| SDD | Software Detailed Design |
| PSIM | Personal SIM |
| TSIM | Telematics SIM |
| TSP | Telematics Service Provider |
| SM | Special Mode (also known as Demo Mode) |

Glossary (Optional)

|  |  |
| --- | --- |
| Glossary | Description |
| TCU | Telematics Control Unit |
| TSP | Telematics Service Provider |
| NGTP | Next Generation Telematics Pattern (Protocol) |
| MCU | Micro Controller Unit |
| TSDP | Telematics Service Data Provider |
| TOPS | Telematics Over Wi-Fi |

Table of Contents

[1 Introduction 7](#_Toc501719568)

[2 Special Mode Component 9](#_Toc501719569)

[2.1 External Design 9](#_Toc501719570)

[2.2 Internal Design 10](#_Toc501719571)

[2.2.1 Static Design 10](#_Toc501719572)

[2.2.2 Dynamic Design 11](#_Toc501719573)

[2.2.2.1 State Design 11](#_Toc501719574)

[2.2.2.2 Interaction Design 12](#_Toc501719575)

[2.2.3 Algorithm Design 12](#_Toc501719576)

[2.2.3.1 SpecialModeApplication class 13](#_Toc501719577)

[2.2.3.1.1 SpecialModeApplication list variables 13](#_Toc501719578)

[2.2.3.1.2 SpecialModeApplication method specifications 13](#_Toc501719579)

[2.2.3.2 SpecialModeBaseProcess class 14](#_Toc501719580)

[2.2.3.2.1 SpecialModeBaseProcess list variables 14](#_Toc501719581)

[2.2.3.2.2 SpecialModeBaseProcess method specifications 14](#_Toc501719582)

[2.2.3.3 SpecialModeHandler class 14](#_Toc501719583)

[2.2.3.3.1 SpecialModeHandler list variable 14](#_Toc501719584)

[2.2.3.3.2 SpecialModeHandler list method 15](#_Toc501719585)

[2.2.3.4 SpecialModeReceiverManager class 15](#_Toc501719586)

[2.2.3.4.1 SpecialModeReceiverManager list variable 15](#_Toc501719587)

[2.2.3.4.2 SpecialModeReceiverManager list method 15](#_Toc501719588)

[2.2.3.5 SpecialModeDebugReceiver class 15](#_Toc501719589)

[2.2.3.5.1 SpecialModeDebugReceiver list variable 15](#_Toc501719590)

[2.2.3.5.2 SpecialModeDebugReceiver list method 15](#_Toc501719591)

[2.2.3.6 SpecialModeDiagReceiver class 15](#_Toc501719592)

[2.2.3.6.1 SpecialModeDiagReceiver list variable 16](#_Toc501719593)

[2.2.3.6.2 SpecialModeDiagReceiver list method 16](#_Toc501719594)

[2.2.3.7 DemoModeProcess class 16](#_Toc501719595)

[2.2.3.7.1 DemoModeProcess list variable 16](#_Toc501719596)

[2.2.3.7.2 DemoModeProcess list method 16](#_Toc501719597)

[2.2.3.8 DemoModeTimeHandler class 16](#_Toc501719598)

[2.2.3.8.1 DemoModeTimeHandler list variable 16](#_Toc501719599)

[2.2.3.8.2 DemoModeTimeHandler list method 16](#_Toc501719600)

[2.2.3.9 DemoModeTimerSet class 16](#_Toc501719601)

[2.2.3.9.1 DemoModeTimerSet list variable 17](#_Toc501719602)

[2.2.3.9.2 DemoModeTimerSet list method 17](#_Toc501719603)

[2.2.3.10 InControlLightProcess class 17](#_Toc501719604)

[2.2.3.10.1 InControlLightProcess list variable 17](#_Toc501719605)

[2.2.3.10.2 InControlLightProcess list method 17](#_Toc501719606)

[2.2.3.11 SpecialModeServiceManager class 17](#_Toc501719607)

[2.2.3.11.1 SpecialModeServiceManager list variable 17](#_Toc501719608)

[2.2.3.11.2 SMServiceManager list method 18](#_Toc501719609)

[2.2.3.12 SpecialModeType <<Enumeration>> 18](#_Toc501719610)

[2.2.3.13 SpecialModeEvent <<Enumeration>> 18](#_Toc501719611)

Figures

[Figure 1 Software Architectural Design 7](#_Toc501719612)

[Figure 2: Sequence Diagram for External design 9](#_Toc501719613)

[Figure 3 Component Design for Special Mode 10](#_Toc501719614)

[Figure 4 Simple Class diagram for Special Mode component 10](#_Toc501719615)

[Figure 5: State Diagram 11](#_Toc501719616)

[Figure 7 Detail Class Diagram 13](#_Toc501719617)

**Tables**

[Table 1 Software Component Descriptions 8](#_Toc501719618)

[Table 2: Simple Class identified 11](#_Toc501719619)

[Table 3 Description of each state 12](#_Toc501719620)

[Table 4 State transitions 12](#_Toc501719621)

[Table 5 Interface design for 12](#_Toc501719622)

[Table 6: SpecialModeApplication list variables. 13](#_Toc501719623)

[Table 7: SpecialModeApplication method specifications. 13](#_Toc501719624)

[Table 8: SpecialModeBaseProcess list variables. 14](#_Toc501719625)

[Table 9: SpecialModeBaseProcess method specifications. 14](#_Toc501719626)

[Table 10: SpecialModeHandler list variable. 14](#_Toc501719627)

[Table 11: SpecialModeHandler method specifications. 15](#_Toc501719628)

[Table 12: SpecialModeReceiverManager list variable. 15](#_Toc501719629)

[Table 13: SpecialModeReceiverManager method specifications 15](#_Toc501719630)

[Table 14: SpecialModeDebugManager list variable. 15](#_Toc501719631)

[Table 15: SpecialModeDebugManager method specifications 15](#_Toc501719632)

[Table 16: SpecialModeDiagReceiver list variables. 16](#_Toc501719633)

[Table 17: SpecialModeDiagReceiver method specifications. 16](#_Toc501719634)

[Table 18: DemoModeProcess list variable. 16](#_Toc501719635)

[Table 19: DemoModeProcess method specifications. 16](#_Toc501719636)

[Table 20: DemoModeTimeHandler list variable 16](#_Toc501719637)

[Table 21: DemoModeTimeHandler method specifications 16](#_Toc501719638)

[Table 22: DemoModeTimerSet list variables 17](#_Toc501719639)

[Table 23: DemoModeTimerSet method specifications 17](#_Toc501719640)

[Table 24: InControlLightProcess list variable 17](#_Toc501719641)

[Table 25: InControlLightProcess method specifications. 17](#_Toc501719642)

[Table 26: SpecialModeServiceManager list variables 17](#_Toc501719643)

[Table 27: SpecialModeServiceManager method specification. 18](#_Toc501719644)

[Table 28: Enumeration: SpecialModeType 18](#_Toc501719645)

[Table 29: Enumeration SpecialModeEvent 18](#_Toc501719646)

# Introduction

This is the software architectural design of JLR Telematics VM in VDC called as JLR TCU4. Telematics VM consists of eCall, bCall, CS, Special Mode (Demo Mode), RVM, RVC, EVT, RDC, PC, ADTA, LT and Provisioning.

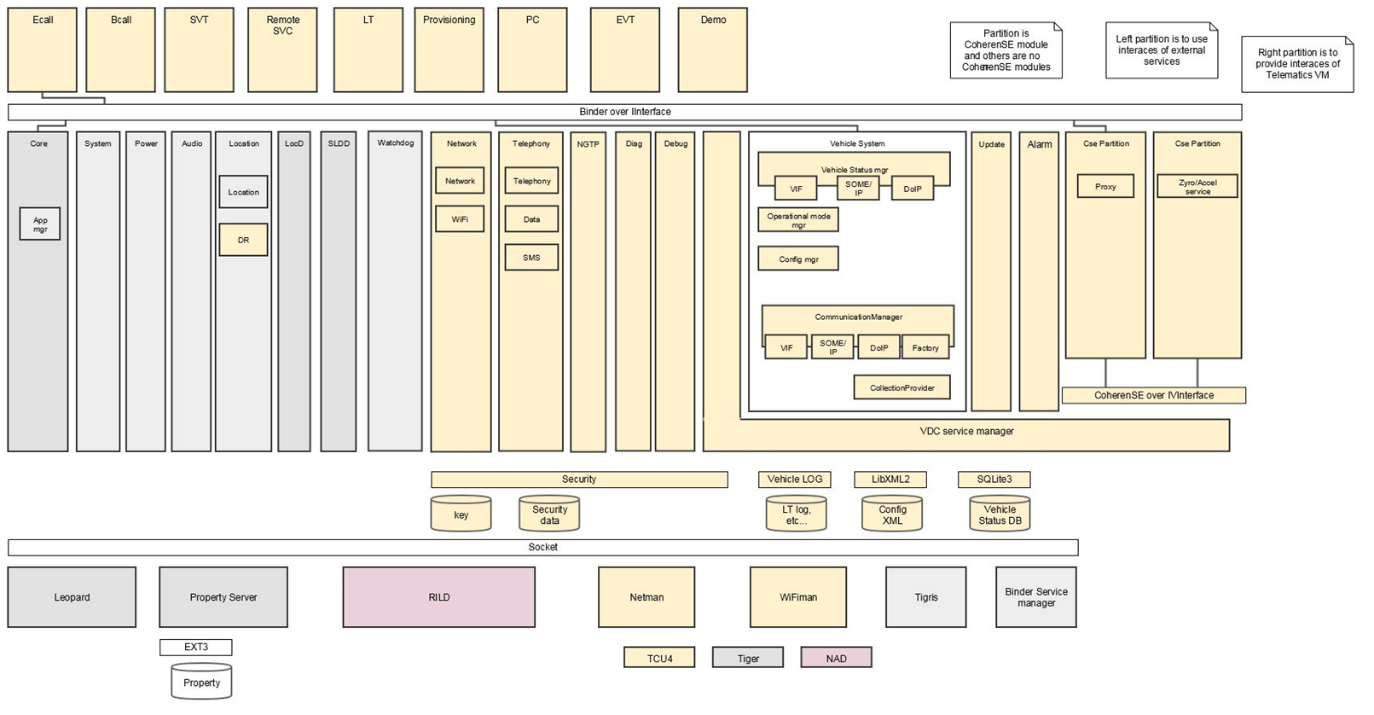
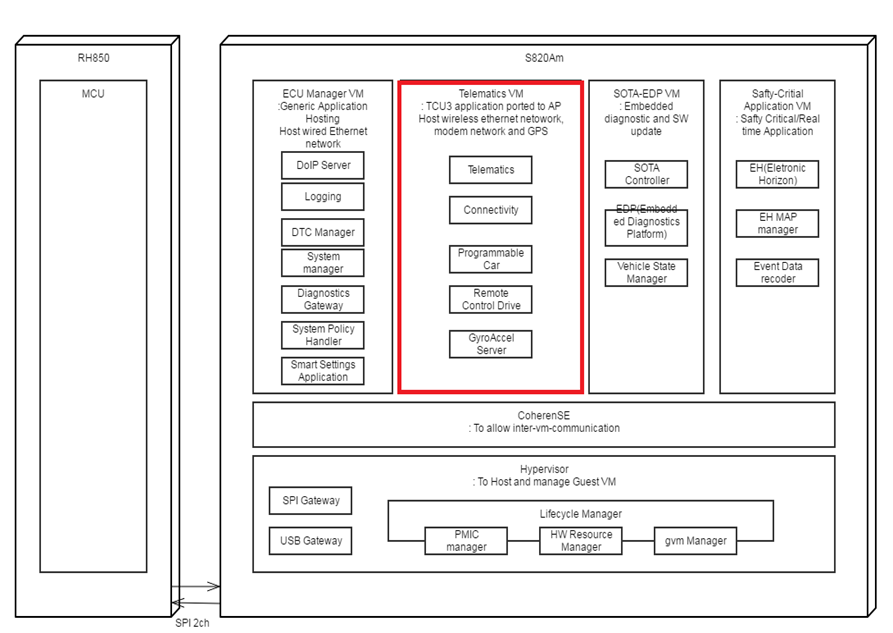


Figure 1 Software Architectural Design

Below table describes the software components of Telematics VM.

Table 1 Software Component Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| SW Component ID | SW Component Name | Description | SRS ID |
| SW Component ID | Special Mode (known as Demo Mode) | Mode using for demo telematics function in car show. | SRD\_TELEM\_CORE7200\_v1  1.15.1.1 General Requirements  SRD\_TELEM\_CORE7300\_v1  1.15.1.2 Demo Mode Activation  SRD\_TELEM\_CORE7301\_v1  1.15.1.2 Demo Mode Activation  SRD\_TELEM\_CORE7302\_v1  1.15.1.2 Demo Mode Activation  SRD\_TELEM\_CORE7400\_v1  1.15.1.3 Demo Mode Functionality  SRD\_TELEM\_CORE7401\_v1  1.15.1.3 Demo Mode Functionality  SRD\_TELEM\_CORE7402\_v1  1.15.1.3 Demo Mode Functionality  SRD\_TELEM\_CORE7404\_v1  1.15.1.3 Demo Mode Functionality  SRD\_TELEM\_CORE7405\_v1  1.15.1.3 Demo Mode Functionality.  SRD\_TELEM\_CORE7500\_v1  Demo Mode Termination  SRD\_TELEM\_CORE7501\_v1  Demo Mode Termination |

# Special Mode Component

## External Design

Special mode in which TCU stays awake regardless of CAN network activity and maintains a network connection via Wi-Fi Client for any telematics data traffic.

This is done to avoid a potential problem with cellular coverage at car shows and features demo.

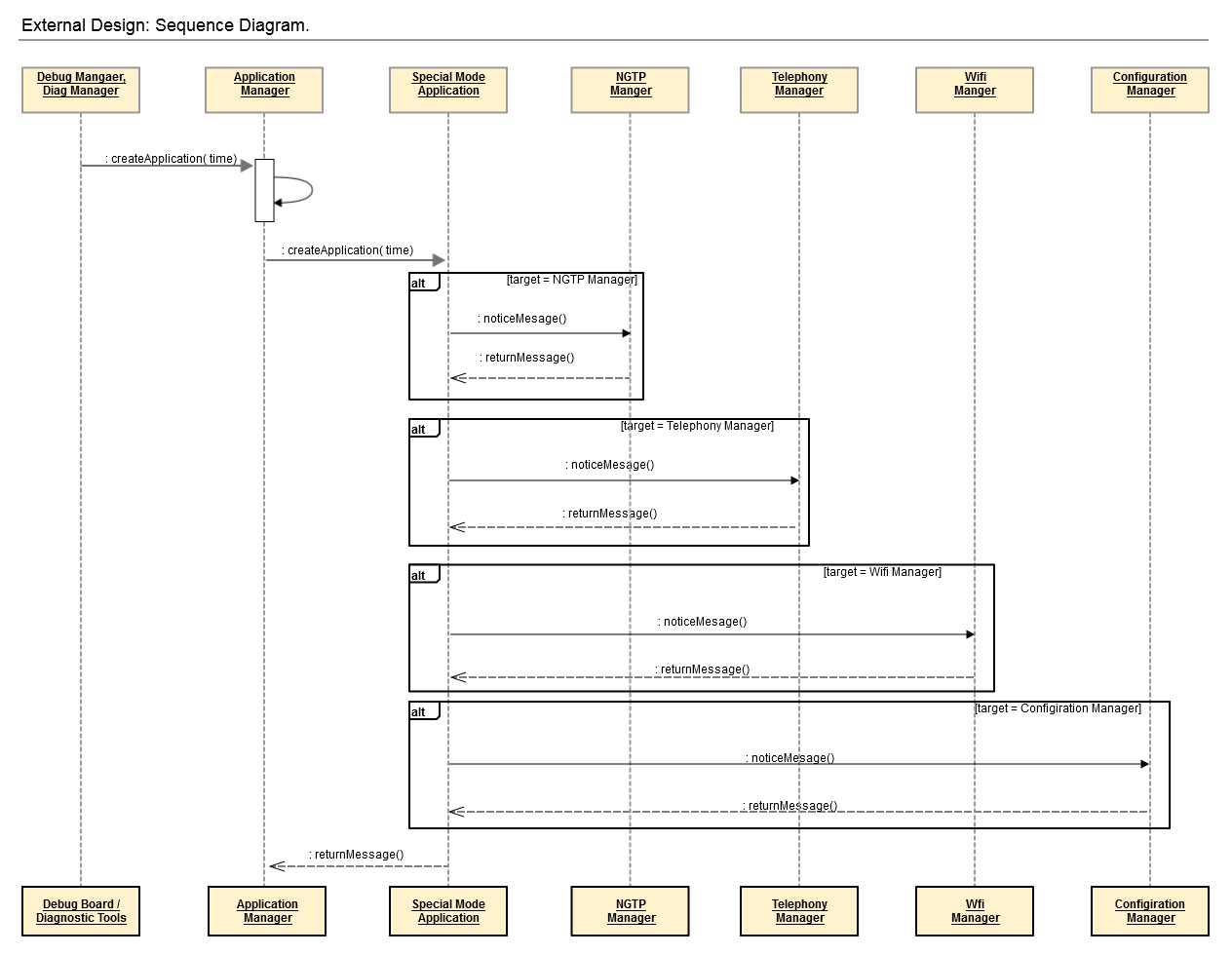


Figure 2: Sequence Diagram for External design

* Step 1.0: When user wants to active Special Mode (Demo Mode) via Debug Board.
* Step 1.1: When user wants to active Special Mode (Demo Mode) via Diagnostic Routine.
* Step 2.0, 2.1, 2.2, 2.3 and 2.4: When Special Mode notices status of other Service Manager.

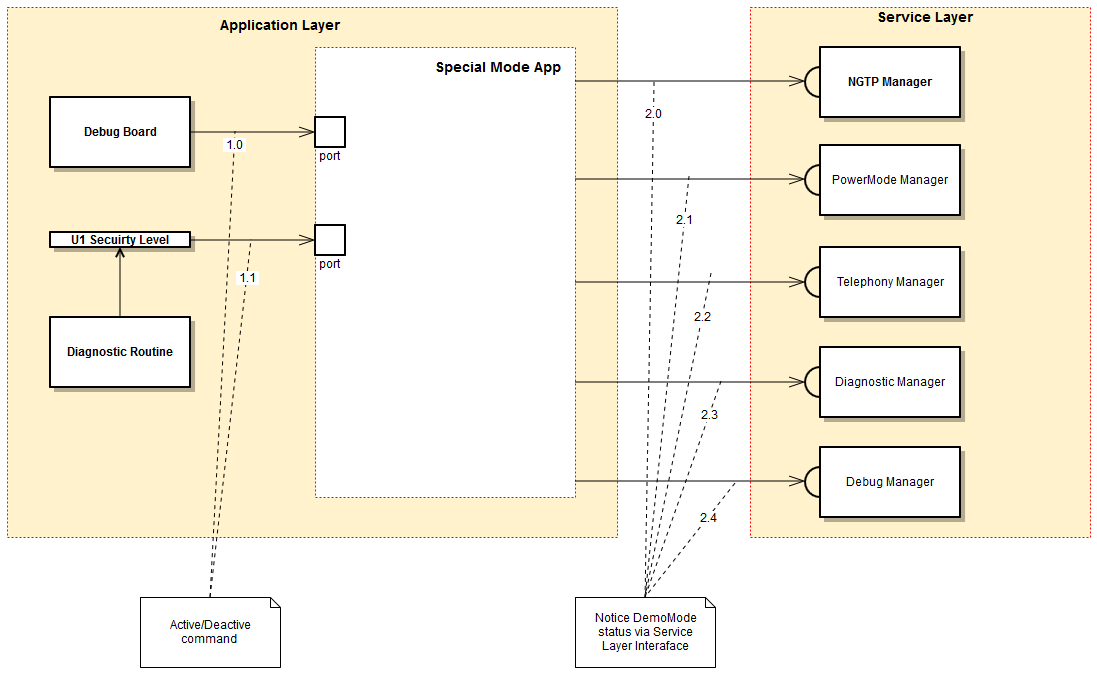


Figure 3 Component Design for Special Mode

## Internal Design

### Static Design

The Special Mode Application supports the operational states listed below:

* Active Demo Mode
* Terminate Demo Mode

The class diagram for the Special Mode component is shown below:

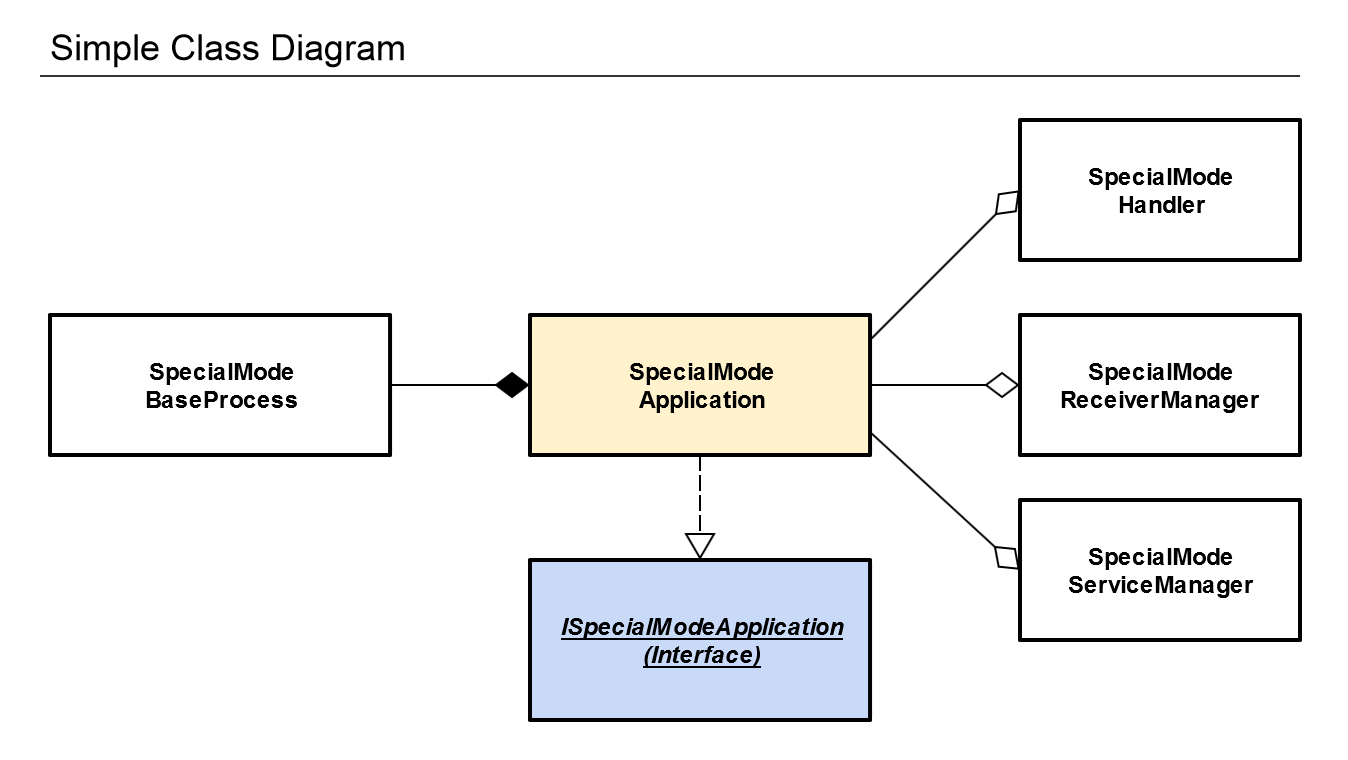


Figure 4 Simple Class diagram for Special Mode component

The classes identified above are described in the below table:

Table 2: Simple Class identified

|  |  |  |
| --- | --- | --- |
| Class | Class ID | Descriptions |
| SpecialModeApplication | SMA\_SpecialModeApplication | The class that is called at Special Mode app launch in the Application Manager. |
| SpecialModeHandler | SMH\_SpecialModeHandler | Handler class for event handling of SpecialModeApplication object |
| SpecialModeReceiverManager | SMRM\_SpecialModeReceiverManger | Receiver class for communication with external interfaces |
| ISpecialModeApplication | ISMA\_ISpecialModeApplication | Interface |
| SpecialModeServiceManager | SMSM\_SpecialModeServiceManager | Get state of other service manager. |
| SpecialModeProcessManager | SMPM\_SpecialModeProcessManager | Observer class for observing state change |

### Dynamic Design

#### State Design

The Special Mode State Machine class shall configure to the following state:

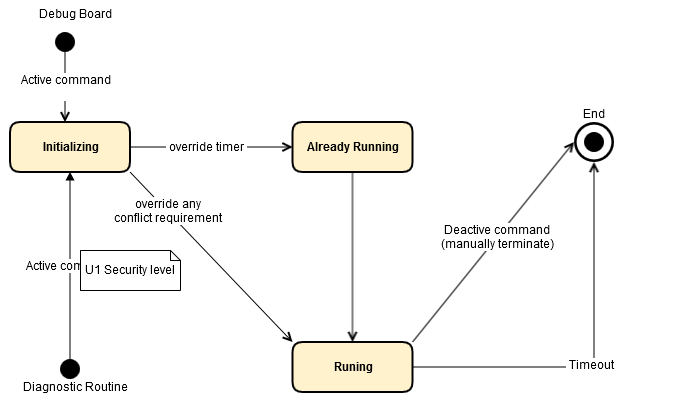


Figure 5: State Diagram

The descriptions for each state are as follows:

Table 3 Description of each state

|  |  |
| --- | --- |
| States | Descriptions |
| Initializing | This is the initial state. |
| Already running | TCU already running in Demo Mode when receiver a new active command from Diagnostic Routine or from Debug Board. |
| Running | TCU running in Demo Mode. |
| End | Terminate Demo Mode ( TCU change to other mode) |

The state transitions for the state diagram are as follows:

Table 4 State transitions

|  |  |  |  |
| --- | --- | --- | --- |
| Current State | Event/Action | Next State | Descriptions |
| Start | Active Command | Initializing | Initialize app, register service |
| Initializing | Override any conflict requirements. | Running | While TCU4 running in other mode (normal, sleep, listen), it’s must change to Demo Mode and override other conflict request to TCU4 |
| Initializing | Override current time value | Already Running | While Demo Mode is already active (running), Telematics shall reset DemoModeTimeout timer value to the value specified in the activation command. |
| Running | Timeout | End | Auto end Demo Mode when time out. |
| Running | Deactivate Command | End | Manually end Demo Mode using Debug board or Diagnostic Tools. |

#### Interaction Design

Table 5 Interface design for

|  |  |  |  |
| --- | --- | --- | --- |
| SW Component | Interface Name | Type | Parameters |
| Special Mode Application | ISpecialModeApplication  (Interface Special Mode Application) | call | TBD |

Sequence diagram for Special Mode Application Interface: TBD

### Algorithm Design

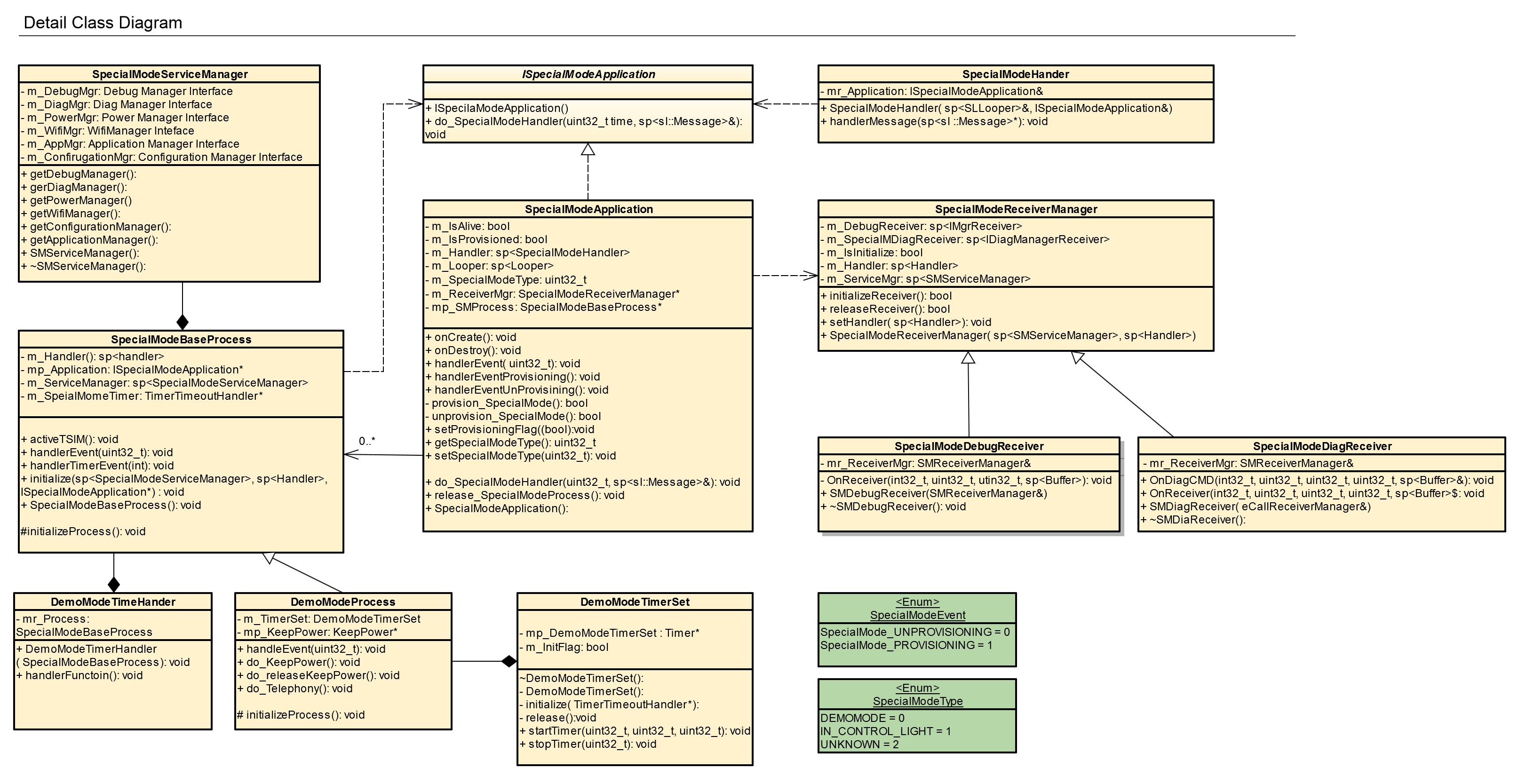


Figure 7 Detail Class Diagram

#### SpecialModeApplication class

#### SpecialModeApplication list variables

Table 6: SpecialModeApplication list variables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| m\_IsAlive | private | bool | Check TCU is already running under Special Mode |
| m\_IsProvisioned | private | bool | True if provisioned |
| m\_Handler | private | sp<Handler> | TBD |
| m\_Looper | private | sp<Looper> | Using for send message to Looper of |
| m\_SpecialModeType | private | SpecialModeType | enum: type of special mode |
| m\_ReceiverMgr | private | SpecialModeReceiverManager\* | Receive message from other component. |
| m\_SpecialModeProcess | private | SpecialModeBaseProcess | Base process when TCU running in Special Mode |

#### SpecialModeApplication method specifications

Table 7: SpecialModeApplication method specifications.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| onCreate | SMA\_onCreate | do something when app start | TBD | TBD |
| onDestroy | SMA\_onDestroy | do something when terminate | TBD | TBD |
| onPostReceiver | SMA\_onPostReceiver | TBD | sp<Post> | TBD |
| handlerEvent | SMA\_handlerEvent | TBD | uint32\_t | TBD |
| handlerEventProvisionning | SMA\_handlerEvent Provisioning | call when success provision Special Mode, start SpecialModeProcess | TBD | TBD |
| handlerEventUnprovisioning | SMA\_handlerEventUnProvisioning | call when provisioning false. Release SpecialModeProcess | TBD | TBD |
| provisionSpecialMode | SMA\_provisionSpecialMode |  | TBD | bool |
| getSpecialModeType | SMA\_getSpecialModeType | <<enum>> SpecialModeType | TBD | uint32\_t |
| doSpecialModeHandler | SMA\_do\_SpecialModeHandler | Handler message | uint32\_t,  sp< sl::Message>& | TBD |
| releaseSpecialModeProcess | SMA\_release\_SpecialModeHandler | Delete special mode process | TBD | TBD |
| SMApplication | SMA\_SpecialModeApplication | Constructor | TBD | TBD |
| ~SMApplication | SpecialModeA\_SMApplication | Destructor | TBD | TBD |

#### SpecialModeBaseProcess class

#### SpecialModeBaseProcess list variables

Table 8: SpecialModeBaseProcess list variables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| m\_Handler | private | sp<Handler> | TBD |
| m\_Application | private | ISpecialModeApplication\* (Special Mode App Interface) | Single instance of Special Mode running |
| m\_ServiceManager | private | sp<SpecialModeServiceManager> | Management other service (i.e. NGTP) |
| m\_SpecialModeTimer | private | TimerTimeoutHandler | Current time value. |

#### SpecialModeBaseProcess method specifications

Table 9: SpecialModeBaseProcess method specifications.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| SpecialModeBaseProcess | SMBP\_SpecialModeBaseProcess | Constructor | TBD | TBD |
| checkStatusTSIM | SMBP\_checkStatusTSIM | Check status of Telematics SIM | TBD | int |
| activeTSIM | SMBP\_activeTSIM | Active Telematics SIM | TBD | void |
| handlerEvent | SMBP\_handlerEvent | TBD | uint32\_t | TBD |
| handlerTimer | SMBP\_handlerTimer | Time management | int | TBD |
| initialize | SMBP\_initialize | Initialize some process when start TCU4 change to Special Mode | sp<ServiceManager>,  sp<Handler>,  ISpecialModeApplication\* | TBD |

#### SpecialModeHandler class

#### SpecialModeHandler list variable

Table 10: SpecialModeHandler list variable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| m\_Application | private | ISMApplication\* | TBD |

#### SpecialModeHandler list method

Table 11: SpecialModeHandler method specifications.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| SpecialModeHandler | SMH\_SpecialModeHander | Constructor | sp<SLLooper>& looper,  ISpecialModeInterfac& service | TBD |
| ~SpecialModeHandler | SMH\_~SpecialModeHandler | Destructor | TBD | TBD |
| handlerMessage | SMH\_handlerMessage | TBD | sp<sl::Message>& message | TBD |

#### SpecialModeReceiverManager class

#### SpecialModeReceiverManager list variable

Table 12: SpecialModeReceiverManager list variable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| m\_IsInitialize | public | bool | Check receiver manager is already initialize |
| m\_DiagReceiver | public | sp<DiagReceiver> | Management signal of Diagnostic tool. |
| m\_Handler | public | sp<Handler> | TBD |
| m\_ServiceManager | public | sp<ServiceManager> | Management communicate with other service layer (NGTP, WIFI, Telephony…) |
| m\_DebugReceiver | public | sp<DebugReceiver> | Management signal from Debug board. |

#### SpecialModeReceiverManager list method

Table 13: SpecialModeReceiverManager method specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| initializeReceiver | SMRM\_initializeReceiver | TBD | TBD | bool |
| releaseReceiver | SMRM\_releaseReceiver | TBD | TBD | bool |
| setHandler | SMRM\_setHandler | TBD | sp<Handler> | TBD |
| specialModeReceiverManager | SMRM\_specialModeReceiverManager | TBD | sp<SMReceiverManager> | TBD |

#### SpecialModeDebugReceiver class

#### SpecialModeDebugReceiver list variable

Table 14: SpecialModeDebugManager list variable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| mr\_ReceiverMgr | private | SMReceiverManager& | Receiver management |

#### SpecialModeDebugReceiver list method

Table 15: SpecialModeDebugManager method specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| onReceiverMgr | SMDR\_onReceiverMgr | Call when received a debug message | uint32\_t,  uint32\_t,  uint32\_t,  sp<Buffer> | TBD |
| SpecialModeDebugReceiver | SMDR\_specialModeDebugReceiver | Constructor | TBD | TBD |
| ~SpecialModeDebugReceiver | SMDR\_specialModeDebugReceiver | Destructor | TBD | TBD |

#### SpecialModeDiagReceiver class

#### SpecialModeDiagReceiver list variable

Table 16: SpecialModeDiagReceiver list variables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| mr\_ReceiverMgr | private | SpecialModeReceiverManager& | receiver management |

#### SpecialModeDiagReceiver list method

Table 17: SpecialModeDiagReceiver method specifications.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| onDiagCMD | SMDR\_onDiagCMD | Listen Diagnostic Command | uint32\_t,  uint32\_t,  uint32\_t,  sp<Buffer> | TBD |
| onReceiver | SMDR\_onReceiver | Call when received a diagnostic message | uint32\_t,  uint32\_t,  uint32\_t,  sp<Buffer> | TBD |
| ~SpecialModeDiagReceiver | SMDR\_~SMDiagReceiver | Destructor | TBD | TBD |
| SpecialModeDiagReceiver | SMDR\_SMDiagReceiver | Constructor | TBD | TBD |

#### DemoModeProcess class

#### DemoModeProcess list variable

Table 18: DemoModeProcess list variable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| m\_TimerSet | public | DemoModeTimerSet\* | Set life time (h) for TCU4 run in Special Mode. |
| m\_KeepPower | public | KeepPower\* | Prevent TCU change to Listen Mode or Sleep Mode. |

#### DemoModeProcess list method

Table 19: DemoModeProcess method specifications.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| handleEvent | DMP\_handlerEvent | TBD | uin32\_t | TBD |
| doKeepPower | DMP\_do\_KeepPower | Prevent TCU enter Sleep Mode or Listen Mode | TBD | TBD |
| doReleaseKeepPower | DMP\_do\_releaseKeepPower | Enable TCU can enter other power mode | TBD | TBD |
| initializeProcess | DMP\_initializeProcess | Call when start DemoMode process | TBD | TBD |

#### DemoModeTimeHandler class

#### DemoModeTimeHandler list variable

Table 20: DemoModeTimeHandler list variable

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| m\_Process | private | SpecialModeBaseProcess | Current time process |

#### DemoModeTimeHandler list method

Table 21: DemoModeTimeHandler method specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| demoModeTimerHandler | DMTH\_demoModeTimerHandler | TBD | SpecialModeBaseProcess | TBD |
| handlerFunction | DMTH\_handlerFunction | TBD | TBD | TBD |

#### DemoModeTimerSet class

#### DemoModeTimerSet list variable

Table 22: DemoModeTimerSet list variables

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| m\_DemoModeTimerSet | private | Timer\* | Timer pointer |
| m\_InitFlag | private | bool | True if initialize success Timer |

#### DemoModeTimerSet list method

Table 23: DemoModeTimerSet method specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| DemoModeTimerSet | DMTS\_DMTimerSet | Constructor | TBD | TBD |
| ~DemoModeTimerSet | DMTS\_~DMTimerSet | Destructor | TBD | TBD |
| initialize | DMTS\_initialize | Initialize time value for new process of Special Mode | TimerTimeoutHandler\* | TBD |
| release | DMTS\_release | Call onDestroy Special Mode |  | TBD |
| startTimer | DMTS\_startTimer | TBD | uint32\_t, (hour)  uint32\_t, (minute)  uint32\_t, (second) | TBD |
| stopTimer | DMTS\_stopTimer | TBD | uint32\_t | TBD |

#### InControlLightProcess class

#### InControlLightProcess list variable

Table 24: InControlLightProcess list variable

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| TBD | TBD | TBD | TBD |

#### InControlLightProcess list method

Table 25: InControlLightProcess method specifications.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| InControlLightProcess | ICLP\_InControlLightProcesss | Constructor | TBD | TBD |
| ~InControlLightProcess | ICLP\_~InControlightProcess | Destructor | TBD | TBD |
| handlerEvent | ILCP\_handlerEvent | TBD | utin32\_t | TBD |
| handlerTimerEvent | ILCP\_handlerTimerEvent | TBD | int | TBD |
| initializeProcess | ILCP\_initializeProcess | Call before enter InControlLightProcess | TBD | TBD |
| doSpecialModeHandler | ILCP\_do\_SpecialMode | TBD | TBD | TBD |

#### SpecialModeServiceManager class

#### SpecialModeServiceManager list variable

Table 26: SpecialModeServiceManager list variables

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Accessibility** | **Type** | **Description** |
| m\_DebugMgr | private | IDebugManager | Communicate with Debug Manager |
| m\_DiagMgr | private | IDiagManager | Communicate with Diagnostic Manager |
| m\_PowerMgr | private | IPowerManger | Communicate with Power manager |
| m\_WifiMgr | private | IWifiManager | Communicate with Wi-Fi Manager |
| m\_AppManager | private | IApplicationManager | Communicate with Application Manager |
| m\_ConfigurationMgr | private | IConfigurationManager | Communicate with Configuration Manager |

#### SMServiceManager list method

Table 27: SpecialModeServiceManager method specification.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Method ID** | **Description** | **Input** | **Output** |
| getDebugManager | SMSM\_getDebugManager | Get service debug manager service | TBD | TBD |
| getDiagManager | SMSM\_getDiagManager | Get diagnostic manager service | TBD | TBD |
| getPowerManager | SMSM\_getPowerManager | Get power manager service | TBD | TBD |
| getWifiManager | SMSM\_getWifiManager | Get Wi-Fi manager service | TBD | TBD |
| getApplicationManager | SMSM\_getApplicationManager | Get application manager service | TBD | TBD |
| getConfigurationManager | SMSM\_getConfigurationManager | Get configuration manager service | TBD | TBD |
| SpecialModeServiceManager | SMSM\_SpecialModeServiceManager | Constructor | TBD | TBD |
| ~SpecialModeServiceManager | SMSM\_~SpecialModeServiceManager | Destructor | TBD | TBD |

#### SpecialModeType <<Enumeration>>

Table 28: Enumeration: SpecialModeType

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| E\_DEMO\_MODE | 0 | TBD |
| E\_IN\_CONTROL\_LIGHT | 1 | TBD |
| E\_UNKNOWN | 2 | TBD |

#### SpecialModeEvent <<Enumeration>>

Table 29: Enumeration SpecialModeEvent

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| E\_SPECIALMODE\_PROVISIONING | 0 | TBD |
| E\_SPECIALMODE\_UNPROVISIONNING | 1 | TBD |