Document Title	SWS_CANStateManager: Complete Change Documentation 4.3.0 - 4.3.1
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	695

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Table of Contents

1 SWS_CANStateManager

1.1 Specification Item ECUC_CanSM_00127

Trace References:

none

Content:

Container Name	CanSMDemEventParameterRefsCanSMDemEventParameterRefs
Description	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
CANSM_E_BUS_OFF	ECUC_CanSM_00070
CANSM_E_MODE_REQUEST_TIMEOUT	ECUC_CanSM_00352

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76189: [CanSM] Reclassify CANSM_E_MODE_REQUEST_TIMEOUT as extended production error

  Problem description:

  CANSM_E_MODE_REQUEST_TIMEOUT has been "degraded" from a production error to a development error with AR 4.0.3 by RfC # 50140, even though comment # 7 of that RfC (RfC # 50140, comment # 7) stated that this error should stay a production error. And now RfC # 59085 aims at reclassifying this error as runtime error.

  From our point of view, supported by Toyota, CANSM_E_MODE_REQUEST_TIMEOUT should really be an extended production error, because it is a secondary (indirect) error (caused by a hardware

defect, but not immediately representing this defect), but there is no other means to detect a hardware defect on some platforms.

A typical hardware fault that can only be detected by CANSM_E_MODE_REQUEST_TIMEOUT is a CAN bus pulled constantly to high: Some controllers will wait forever to detect a low level before they try to send a message, so they will never go to bus-off.

Agreed solution:

Change CANSM_E_MODE_REQUEST_TIMEOUT to an extended production error:

Change SWS_CanSM_00385: New Text: If the CanSM module state machine was triggered with T_REPEAT_MAX (ref. to SWS_CanSM_00463, SWS_CanSM_00480, SWS_CanSM_00495, SWS_CanSM_00523, SWS_CanSM_00536), the CanSM module shall call the function Dem_SetEventStatus (ref. to chapter 8.5.1) with the parameters EventId := CANSM_E_MODE_REQUEST_TIMEOUT (ref. to chapter 7.3). (SRS_Can_01142)
Change SWS_CanSM_00654: Remove CANSM_E_MODE_REQUEST_TIMEOUT entry
New: SWS_CanSM_xxxx: Add "Extended Production Errors" to Chapter 7.3 with "CANSM_E_MODE_REQUEST_TIMEOUT" entry

Error Name: CANSM_E_MODE_REQUEST_TIMEOUT

Short Description: Mode request for a network failed more often than allowed by configuration

Long Description: The CAN State Manager was not able to change the mode of a CAN network after CanSMModeRequestRepetitionMax retries. It monitors the following CanIf services and the corresponding indications:
CanIf_SetControllerMode() -> CanSM_ControllerModeIndication()
CanIf_SetTrcvMode() -> CanSM_TransceiverModeIndication()
CanIf_CheckTrcvWakeFlag() -> CanSM_CheckTransceiverWakeFlagIndication()
CanIf_ClrTrcvWufFlag() -> CanSM_ClearTrcvWufFlagIndication()

Recommended DTC: Assigned by DEM

Detection Criteria:
Fail: When the CAN State Manager executed any of the CanIf services listed above without receiving the corresponding indication for CanSMModeRequestRepetitionMax times, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREFAILED to DEM.

Pass: When CAN State Manager receives any of the indications listed above, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREPASSED to DEM.

Secondary Parameters: None
Time Required: Depending on CanSMModeRequestRepetitionMax and CanSMMainFunctionTimePeriod.

Monitor Frequency: Continuously

MIL illumination: Assigned by DEM

Reference: SRS_BSW_00466

==========
ECUC XML:

- In container CanSMDemEventParameterRefs, add reference to DEM event named CANSM_E_MODE_REQUEST_TIMEOUT. Copy other fields from CANSM_E_BUS_OFF.
–Last change on issue 76189 comment 26–

BW-C-Level:

Application	Specification	Bus
1	3	1

1.2 Specification Item ECUC_CanSM_00335

Trace References:

none

Content:

Name	CanSMModeRequestRepetitionMaxCanSMConfiguration.CanSMModeRequestRepetitionMax	
Description	Specifies the maximal amount of mode request repetitions without a respective mode indication from the CanIf module until the CanSM module reports a Default Development Error to the Det and tries to go back to no communication.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Variant Value	true	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73570: No "default error" in AUTOSAR

  Problem description:

  The DET was renamed from development error tracer to default error tracer.

  This change was most of the time done automatically and unfortunately renamed "developement error" to "default error".

  "default error" should always be followed by "tracer", otherwise, "development error" is probably the right term.

  This could increase the impact (compared to my selection of impacted document, but formally, the configuration parameters *DevErrorDetect are not using the correct description:
  "Switches the Default Error Tracer (Det) detection and notification..."

  The parameter switches on/off the developement error detection. The DET does not need to be detected and can be present even when the parameter is set to false.

  Agreed solution:

  Rename "default error" to "development error" in all impacted documents, but not in an automated way (Do not change "default error tracer" to "developement error tracer"!)

  Blueprint/Example:
  - sub chapter is now called "7.x Default errors"
  - "[SWS_xxx_yyyyy]
  In case default error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected errors to the DET. ()"
  - "[SWS_xxx_yyyyy]
  If default error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the default error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"

- "In case default errors are enabled,..."
- "module raises the Default error XXX_E_TRANSITION"
- "The DET provides services to store default errors"
...

The correct text would be:
- sub chapter is called "7.x Development errors"
- "[SWS_xxx_yyyyy]
In case development error detection is enabled for the xxxx module: The xxxx module shall check API parameters for validity and report detected development errors to the DET. ()"
- "[SWS_xxx_yyyyy]
If development error detection is enabled: the function shall check that the service xxx_Init was previously called. If the check fails, the function shall raise the development error XXX_E_NOT_INITIALIZED otherwise (if DET is disabled) return E_NOT_OK. ()"
- "In case development errors are enabled,..."
- "module raises the development error XXX_E_TRANSITION"
- "The DET provides services to store development errors"

Solution for SWS_RTE:
– SWS_RTE —
- Change 4.8 Default errors to 4.8 Development errors
- Change "Errors which can occur at runtime in the RTE are classified as default errors" to "Errors which can occur at runtime in the RTE are classified as development errors"
- Remove [SWS_Rte_07676]
- Change [SWS_RTE_06611]"If a violation is detected the RTE shall report a default error to the DET." to "If a violation is detected the RTE shall report a development error to the DET."
- Change [SWS_Rte_06631]
[SWS_Rte_06631] d The RTE shall use the OS Application Identifier as the Instance Id to enable the developer to identify in which runtime section of the RTE the error occurs. This Instance ID is even unique across multi cores and so implicitly allows the development error to be traced to a specific core. c(SRS_BSW_00337)

SRS_Libraries:
- In chapter "3 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_SPALGeneral:
- In chapter "6.1.1.3.1 [SRS_SPAL_00157] ...": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "6.1.1.4.2 [SRS_SPAL_12448] ...": Rename "Development Error Tracer" to "Default Error Tracer"

SRS_FlashTest:
- In chapter "6.1 Functional Requirements": Rename "Development Error Tracer" to "Default Error Tracer"
- In chapter "7 References":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_MFXLibrary:
- In chapter "2 Acronyms and abbreviations": Rename "Development Error Tracer" to "Default Error Tracer"

SWS_MemoryAbstractionInterface:
- In chapter "3.1 Input documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_FlexRayNetworkManagement:
- In chapter "3.3 Related AUTOSAR documents":
Rename "Development Error Tracer" to "Default Error Tracer"
Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_CANStateManager:
- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_PDURouter:
- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"

SWS_EEPROMDriver:
- In chapter "3.1 Input documents": Rename "AUTOSAR_SWS_DevelopmentErrorTracer" to "AUTOSAR_SWS_DefaultErrorTracer"
–Last change on issue 73570 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item ECUC_CanSM_00352

Trace References:

none

Content:

Name	CANSM_E_MODE_REQUEST_TIMEOUTCanSMDemEventParameterRefs.CANSM_E_MODE_REQUEST_TIMEOUT		
Description	Reference to configured DEM event to report bus off errors for this CAN network.		
Multiplicity	0..1		
Type	Symbolic name reference to [DemEventParameter]		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: Dem		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76189: [CanSM] Reclassify CANSM_E_MODE_REQUEST_TIMEOUT as extended production error

Problem description:

CANSM_E_MODE_REQUEST_TIMEOUT has been "degraded" from a production error to a development error with AR 4.0.3 by RfC # 50140, even though comment # 7 of that RfC (RfC # 50140, comment # 7) stated that this error should stay a production error. And now RfC # 59085 aims at reclassifying this error as runtime error.

From our point of view, supported by Toyota, CANSM_E_MODE_REQUEST_TIMEOUT should really be an extended production error, because it is a secondary (indirect) error (caused by a hardware defect, but not immediately representing this defect), but there is no other means to detect a hardware defect on some platforms.

A typical hardware fault that can only be detected by CANSM_E_MODE_REQUEST_TIMEOUT is a CAN bus pulled constantly to high: Some controllers will wait forever to detect a low level before they try to send a message, so they will never go to bus-off.

Agreed solution:

Change CANSM_E_MODE_REQUEST_TIMEOUT to an extended production error:

Change SWS_CanSM_00385: New Text: If the CanSM module state machine was triggered with T_REPEAT_MAX (ref. to SWS_CanSM_00463, SWS_CanSM_00480, SWS_CanSM_00495, SWS_CanSM_00523, SWS_CanSM_00536), the CanSM module shall call the function Dem_SetEventStatus (ref. to chapter 8.5.1) with the parameters EventId := CANSM_E_MODE_REQUEST_TIMEOUT (ref. to chapter 7.3). (SRS_Can_01142)
Change SWS_CanSM_00654: Remove CANSM_E_MODE_REQUEST_TIMEOUT entry
New: SWS_CanSM_xxxx: Add "Extended Production Errors" to Chapter 7.3 with "CANSM_E_MODE_REQUEST_TIMEOUT" entry

Error Name: CANSM_E_MODE_REQUEST_TIMEOUT

Short Description: Mode request for a network failed more often than allowed by configuration

Long Description: The CAN State Manager was not able to change the mode of a CAN network after CanSMModeRequestRepetitionMax retries. It monitors the following CanIf services and the corresponding indications:
CanIf_SetControllerMode() -> CanSM_ControllerModeIndication()
CanIf_SetTrcvMode() -> CanSM_TransceiverModeIndication()

CanIf_CheckTrcvWakeFlag() -> CanSM_CheckTransceiverWakeFlagIndication()
CanIf_ClrTrcvWufFlag() -> CanSM_ClearTrcvWufFlagIndication()

Recommended DTC: Assigned by DEM

Detection Criteria:
Fail: When the CAN State Manager executed any of the CanIf services listed above without receiving the corresponding indication for CanSMModeRequestRepetitionMax times, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREFAILED to DEM.
Pass: When CAN State Manager receives any of the indications listed above, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREPASSED to DEM.

Secondary Parameters: None
Time Required: Depending on CanSMModeRequestRepetitionMax and CanSMMainFunctionTimePeriod.

Monitor Frequency: Continuously

MIL illumination: Assigned by DEM

Reference: SRS_BSW_00466

==========
ECUC XML:

- In container CanSMDemEventParameterRefs, add reference to DEM event named CANSM_E_MODE_REQUEST_TIMEOUT. Copy other fields from CANSM_E_BUS_OFF.
–Last change on issue 76189 comment 26–

BW-C-Level:

Application	Specification	Bus
1	3	1

1.4 Specification Item SWS_CanSM_00385

Trace References:

SRS_Can_01142

Content:

If the CanSM module state machine was triggered with T_REPEAT_MAX (ref. to SWS_CanSM_00463, SWS_CanSM_00480, SWS_CanSM_00495, SWS_CanSM_00523, SWS_CanSM_00536), the CanSM module shall call the function Det_ReportError with the ErrorId parameter Dem_SetEventStatus with the parameter EventId EventId := CANSM_E_MODE_REQUEST_TIMEOUT (ref. to chapter [REF]).

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76189: [CanSM] Reclassify CANSM_E_MODE_REQUEST_TIMEOUT as extended production error

  Problem description:

  CANSM_E_MODE_REQUEST_TIMEOUT has been "degraded" from a production error to a development error with AR 4.0.3 by RfC # 50140, even though comment # 7 of that RfC (RfC # 50140, comment # 7) stated that this error should stay a production error. And now RfC # 59085 aims at reclassifying this error as runtime error.

  From our point of view, supported by Toyota, CANSM_E_MODE_REQUEST_TIMEOUT should really be an extended production error, because it is a secondary (indirect) error (caused by a hardware defect, but not immediately representing this defect), but there is no other means to detect a hardware defect on some platforms.

  A typical hardware fault that can only be detected by CANSM_E_MODE_REQUEST_TIMEOUT is a CAN bus pulled constantly to high: Some controllers will wait forever to detect a low level before they try to send a message, so they will never go to bus-off.

  Agreed solution:

  Change CANSM_E_MODE_REQUEST_TIMEOUT to an extended production error:

  Change SWS_CanSM_00385: New Text: If the CanSM module state machine was triggered with T_REPEAT_MAX (ref. to SWS_CanSM_00463, SWS_CanSM_00480, SWS_CanSM_00495, SWS_CanSM_00523, SWS_CanSM_00536), the CanSM module shall call the function Dem_SetEventStatus (ref. to chapter 8.5.1) with the parameters EventId := CANSM_E_MODE_REQUEST_TIMEOUT (ref. to chapter 7.3). (SRS_Can_01142)
  Change SWS_CanSM_00654: Remove CANSM_E_MODE_REQUEST_TIMEOUT entry

New: SWS_CanSM_xxxx: Add "Extended Production Errors" to Chapter 7.3 with "CANSM_E_MODE_REQUEST_TIMEOUT" entry

Error Name: CANSM_E_MODE_REQUEST_TIMEOUT

Short Description: Mode request for a network failed more often than allowed by configuration

Long Description: The CAN State Manager was not able to change the mode of a CAN network after CanSMModeRequestRepetitionMax retries. It monitors the following CanIf services and the corresponding indications:
CanIf_SetControllerMode() -> CanSM_ControllerModeIndication()
CanIf_SetTrcvMode() -> CanSM_TransceiverModeIndication()
CanIf_CheckTrcvWakeFlag() -> CanSM_CheckTransceiverWakeFlagIndication()
CanIf_ClrTrcvWufFlag() -> CanSM_ClearTrcvWufFlagIndication()

Recommended DTC: Assigned by DEM

Detection Criteria:
Fail: When the CAN State Manager executed any of the CanIf services listed above without receiving the corresponding indication for CanSMModeRequestRepetitionMax times, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREFAILED to DEM.
Pass: When CAN State Manager receives any of the indications listed above, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREPASSED to DEM.

Secondary Parameters: None
Time Required: Depending on CanSMModeRequestRepetitionMax and CanSMMainFunctionTimePeriod.

Monitor Frequency: Continuously

MIL illumination: Assigned by DEM

Reference: SRS_BSW_00466

==========
ECUC XML:

- In container CanSMDemEventParameterRefs, add reference to DEM event

named CANSM_E_MODE_REQUEST_TIMEOUT. Copy other fields from CANSM_E_BUS_OFF.
–Last change on issue 76189 comment 26–

BW-C-Level:

Application	Specification	Bus
1	3	1

1.5 Specification Item SWS_CanSM_00395

Trace References:

SRS_Can_01145

Content:

If the CanSM module has to deny the request CanSM_RequestComMode, because of a pending mode indication (ref. to CANSM388), then this function shall call the function Det_ReportError with the ErrorId parameter CANSM_E_WAIT_MODE_INDICATION (ref. to chapter REF).

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76442: [CanSM] unnecessary requirement SWS_CanSM_00395 + dev. error CANSM_E_WAIT_MODE_INDICATION

  Problem description:

  # 55033 self-contained CPR solution tells that:
  - Remove limitations for usage of CanSM_RequestComMode to request a transition from FULL_COMMUNICATION to NO_COMMUNICATION
  - Remove DET error, which is related to the limitation of CanSM_RequestComMode

  However, following descriptions are still on CP R4.3.0 SWS CanSM:
  - corresponding requirement [SWS_CanSM_00395]
  - corresponding development error CANSM_E_WAIT_MODE_INDICATION in [SWS_CanSM_00654]
  –Last change on issue 76442 comment 12–

  Agreed solution:

  Remove [SWS_CanSM_00395]
  Remove "CANSM_E_WAIT_MODE_INDICATION " from [SWS_CanSM_00654]
  ->
  Type or error Relevance Related error code Value [hex]

API service used without module initialization Development CANSM_E_UNINIT 0x01
API service called with wrong pointer Development CANSM_E_PARAM_POINTER 0x02
API service called with wrong parameter Development CANSM_E_INVALID_NETWORK_HANDLE 0x03
API service called with wrong parameter Development CANSM_E_PARAM_CONTROLLER 0x04
API service called with wrong parameter Development CANSM_E_PARAM_TRANSCEIVER 0x05
Mode request for a network failed more often as allowed by configuration Development CANSM_E_MODE_REQUEST_TIMEOUT 0x0A
DeInit API service called when not all CAN networks are in state CANSM_NO_COMMUNICATION Development CANSM_E_NOT_IN_NO_COM 0x0B
–Last change on issue 76442 comment 5–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.6 Specification Item SWS_CanSM_00654

Trace References:

SRS_BSW_00337

Content:

Type or error	Relevance	Related error code	Value [hex]
API service used without module initialization	Development	CANSM_E_UNINIT	0x01
API service called with wrong pointer	Development	CANSM_E_PARAM_POINTER	0x02
API service called with wrong parameter	Development	CANSM_E_INVALID_NETWORK_HANDLE	0x03
API service called with wrong parameter	Development	CANSM_E_PARAM_CONTROLLER	0x04
API service called with wrong parameter	Development	CANSM_E_PARAM_TRANSCEIVER	0x05
Network mode request during pending indication	Development	CANSM_E_WAIT_MODE_INDICATION	0x07
Mode request for a network failed more often as allowed by configuration	Development	CANSM_E_MODE_REQUEST_TIMEOUT	0x0A

Type or error	Relevance	Related error code	Value [hex]
DeInit API service called when not all CAN networks are in state CANSM_NO_COMMUNICATION	Development	CANSM_E_NOT_IN_NO_COM	0x0B

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76189: [CanSM] Reclassify CANSM_E_MODE_REQUEST_TIMEOUT as extended production error

  Problem description:

  CANSM_E_MODE_REQUEST_TIMEOUT has been "degraded" from a production error to a development error with AR 4.0.3 by RfC # 50140, even though comment # 7 of that RfC (RfC # 50140, comment # 7) stated that this error should stay a production error. And now RfC # 59085 aims at reclassifying this error as runtime error.

  From our point of view, supported by Toyota, CANSM_E_MODE_REQUEST_TIMEOUT should really be an extended production error, because it is a secondary (indirect) error (caused by a hardware defect, but not immediately representing this defect), but there is no other means to detect a hardware defect on some platforms.

  A typical hardware fault that can only be detected by CANSM_E_MODE_REQUEST_TIMEOUT is a CAN bus pulled constantly to high: Some controllers will wait forever to detect a low level before they try to send a message, so they will never go to bus-off.

  Agreed solution:

  Change CANSM_E_MODE_REQUEST_TIMEOUT to an extended production error:

  Change SWS_CanSM_00385: New Text: If the CanSM module state machine was triggered with T_REPEAT_MAX (ref. to SWS_CanSM_00463, SWS_CanSM_00480, SWS_CanSM_00495, SWS_CanSM_00523, SWS_CanSM_00536), the CanSM module shall call the function Dem_SetEventStatus (ref. to chapter 8.5.1) with the parameters EventId := CANSM_E_MODE_REQUEST_TIMEOUT (ref. to chapter 7.3). (SRS_Can_01142)
  Change SWS_CanSM_00654: Remove CANSM_E_MODE_REQUEST_TIMEOUT entry
  New: SWS_CanSM_xxxx: Add "Extended Production Errors" to Chapter 7.3 with "CANSM_E_MODE_REQUEST_TIMEOUT" entry

Error Name: CANSM_E_MODE_REQUEST_TIMEOUT

Short Description: Mode request for a network failed more often than allowed by configuration

Long Description: The CAN State Manager was not able to change the mode of a CAN network after CanSMModeRequestRepetitionMax retries. It monitors the following CanIf services and the corresponding indications:
CanIf_SetControllerMode() -> CanSM_ControllerModeIndication()
CanIf_SetTrcvMode() -> CanSM_TransceiverModeIndication()
CanIf_CheckTrcvWakeFlag() -> CanSM_CheckTransceiverWakeFlagIndication()
CanIf_ClrTrcvWufFlag() -> CanSM_ClearTrcvWufFlagIndication()

Recommended DTC: Assigned by DEM

Detection Criteria:
Fail: When the CAN State Manager executed any of the CanIf services listed above without receiving the corresponding indication for CanSMModeRequestRepetitionMax times, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREFAILED to DEM.
Pass: When CAN State Manager receives any of the indications listed above, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREPASSED to DEM.

Secondary Parameters: None
Time Required: Depending on CanSMModeRequestRepetitionMax and CanSMMainFunctionTimePeriod.

Monitor Frequency: Continuously

MIL illumination: Assigned by DEM

Reference: SRS_BSW_00466

==========
ECUC XML:

- In container CanSMDemEventParameterRefs, add reference to DEM event named CANSM_E_MODE_REQUEST_TIMEOUT. Copy other fields from CANSM_E_BUS_OFF.
–Last change on issue 76189 comment 26–

BW-C-Level:

Application	Specification	Bus
1	3	1

- RfC #76442: [CanSM] unnecessary requirement SWS_CanSM_00395 + dev. error CANSM_E_WAIT_MODE_INDICATION

  Problem description:

  # 55033 self-contained CPR solution tells that:
  - Remove limitations for usage of CanSM_RequestComMode to request a transition from FULL_COMMUNICATION to NO_COMMUNICATION
  - Remove DET error, which is related to the limitation of CanSM_RequestComMode

  However, following descriptions are still on CP R4.3.0 SWS CanSM:
  - corresponding requirement [SWS_CanSM_00395]
  - corresponding development error CANSM_E_WAIT_MODE_INDICATION in [SWS_CanSM_00654]
  –Last change on issue 76442 comment 12–

  Agreed solution:

  Remove [SWS_CanSM_00395]
  Remove "CANSM_E_WAIT_MODE_INDICATION " from [SWS_CanSM_00654]
  ->
  Type or error Relevance Related error code Value [hex]
  API service used without module initialization Development CANSM_E_UNINIT 0x01
  API service called with wrong pointer Development CANSM_E_PARAM_POINTER 0x02
  API service called with wrong parameter Development CANSM_E_INVALID_NETWORK_HANDLE 0x03
  API service called with wrong parameter Development CANSM_E_PARAM_CONTROLLER 0x04
  API service called with wrong parameter Development CANSM_E_PARAM_TRANSCEIVER 0x05
  Mode request for a network failed more often as allowed by configuration Development CANSM_E_MODE_REQUEST_TIMEOUT 0x0A
  DeInit API service called when not all CAN networks are in state CANSM_NO_COMMUNICATION Development CANSM_E_NOT_IN_NO_COM 0x0B
  –Last change on issue 76442 comment 5–

  BW-C-Level:

Application	Specification	Bus
1	4	1

1.7 Specification Item SWS_CanSM_00664

Trace References:

SRS_BSW_00466

Content:

Error Name:	CANSM_E_MODE_REQUEST_TIMEOUT	
Short Description:	Mode request for a network failed more often than allowed by configuration	
Long Description:	The CAN State Manager was not able to change the mode of a CAN network after CanSMModeRequestRepetitionMax retries. It monitors the following CanIf services and the corresponding indications: CanIf_SetControllerMode() -> CanSM_ControllerModeIndication() CanIf_SetTrcvMode() -> CanSM_TransceiverModeIndication() CanIf_CheckTrcvWakeFlag() -> CanSM_CheckTransceiverWakeFlagIndication() CanIf_ClrTrcvWufFlag() -> CanSM_ClearTrcvWufFlagIndication()	
Recommended DTC:	Assigned by DEM	
Detection Criteria:	Fail	When the CAN State Manager executed any of the CanIf services listed above without receiving the corresponding indication for CanSMModeRequest RepetitionMax times, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREFAILED to DEM.
	Pass	When CAN State Manager receives any of the indications listed above, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREPASSED to DEM.
Secondary Parameters:	None	
Time Required:	Depending on CanSMModeRequestRepetitionMax and CanSMMainFunctionTime Period.	
Monitor Frequency	Continuous	
MIL illumniation:	Assigned by DEM	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76189: [CanSM] Reclassify CANSM_E_MODE_REQUEST_TIMEOUT as extended production error

  Problem description:

  CANSM_E_MODE_REQUEST_TIMEOUT has been "degraded" from a production error to a development error with AR 4.0.3 by RfC # 50140, even though comment # 7 of that RfC (RfC # 50140, comment # 7) stated that this error should stay a production error. And now RfC # 59085 aims at reclassifying this error as runtime error.

  From our point of view, supported by Toyota, CANSM_E_MODE_REQUEST_TIMEOUT should really be an extended production error, because it is a secondary (indirect) error (caused by a hardware defect, but not immediately representing this defect), but there is no other means to detect a hardware defect on some platforms.

  A typical hardware fault that can only be detected by CANSM_E_MODE_REQUEST_TIMEOUT is a CAN bus pulled constantly to high: Some controllers will wait forever to detect a low level before they try to send a message, so they will never go to bus-off.

  Agreed solution:

  Change CANSM_E_MODE_REQUEST_TIMEOUT to an extended production error:

  Change SWS_CanSM_00385: New Text: If the CanSM module state machine was triggered with T_REPEAT_MAX (ref. to SWS_CanSM_00463, SWS_CanSM_00480, SWS_CanSM_00495, SWS_CanSM_00523, SWS_CanSM_00536), the CanSM module shall call the function Dem_SetEventStatus (ref. to chapter 8.5.1) with the parameters EventId := CANSM_E_MODE_REQUEST_TIMEOUT (ref. to chapter 7.3). (SRS_Can_01142)
  Change SWS_CanSM_00654: Remove CANSM_E_MODE_REQUEST_TIMEOUT entry
  New: SWS_CanSM_xxxx: Add "Extended Production Errors" to Chapter 7.3 with "CANSM_E_MODE_REQUEST_TIMEOUT" entry

  Error Name: CANSM_E_MODE_REQUEST_TIMEOUT

  Short Description: Mode request for a network failed more often than allowed by configuration

  Long Description: The CAN State Manager was not able to change the mode of a CAN network after CanSMModeRequestRepetitionMax retries. It monitors the following CanIf services and the corresponding indications:

CanIf_SetControllerMode() -> CanSM_ControllerModeIndication()
CanIf_SetTrcvMode() -> CanSM_TransceiverModeIndication()
CanIf_CheckTrcvWakeFlag() -> CanSM_CheckTransceiverWakeFlagIndication()
CanIf_ClrTrcvWufFlag() -> CanSM_ClearTrcvWufFlagIndication()

Recommended DTC: Assigned by DEM

Detection Criteria:
Fail: When the CAN State Manager executed any of the CanIf services listed above without receiving the corresponding indication for CanSMModeRequestRepetitionMax times, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREFAILED to DEM.
Pass: When CAN State Manager receives any of the indications listed above, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREPASSED to DEM.

Secondary Parameters: None
Time Required: Depending on CanSMModeRequestRepetitionMax and CanSMMainFunctionTimePeriod.

Monitor Frequency: Continuously

MIL illumination: Assigned by DEM

Reference: SRS_BSW_00466

==========
ECUC XML:

- In container CanSMDemEventParameterRefs, add reference to DEM event named CANSM_E_MODE_REQUEST_TIMEOUT. Copy other fields from CANSM_E_BUS_OFF.
–Last change on issue 76189 comment 26–

BW-C-Level:

Application	Specification	Bus
1	3	1

1.8 Specification Item SWS_CanSM_00666

Trace References:

none

Content:

Error Name:	CANSM_E_BUS_OFF (ref. to ECUC_CanSM_00070)	
Short Description:	Bus-off detection	
Long Description:	The bus-off recovery state machine of a CAN network has detected a certain amount of sequential bus-offs without successful recovery	
Recommended DTC:	Assigned by DEM	
Detection Criteria:	Fail	PRE_FAILED when CanSM_Controller BusOff is called (T_BUS_OFF/E_BUS_OFF), debouncing to be defined by OEM in DEM
	Pass	After successful transmission of a CAN frame (G_BUS_OFF_PASSIVE/E_BUS_OFF_PASSIVE)
Secondary Parameters:	None	
Time Required:	PRE_FAILED immediately (in error interrupt context), FAILED depending on debounce configuration of DEM	
Monitor Frequency	Continuous	
MIL illumniation:	Assigned by DEM	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76189: [CanSM] Reclassify CANSM_E_MODE_REQUEST_TIMEOUT as extended production error

  Problem description:

  CANSM_E_MODE_REQUEST_TIMEOUT has been "degraded" from a production error to a development error with AR 4.0.3 by RfC # 50140, even though comment # 7 of that RfC (RfC # 50140, comment # 7) stated that this error should stay a production error. And now RfC # 59085 aims at reclassifying this error as runtime error.

  From our point of view, supported by Toyota, CANSM_E_MODE_REQUEST_TIMEOUT should really be an extended pro-

duction error, because it is a secondary (indirect) error (caused by a hardware defect, but not immediately representing this defect), but there is no other means to detect a hardware defect on some platforms.

A typical hardware fault that can only be detected by CANSM_E_MODE_REQUEST_TIMEOUT is a CAN bus pulled constantly to high: Some controllers will wait forever to detect a low level before they try to send a message, so they will never go to bus-off.

Agreed solution:

Change CANSM_E_MODE_REQUEST_TIMEOUT to an extended production error:

Change SWS_CanSM_00385: New Text: If the CanSM module state machine was triggered with T_REPEAT_MAX (ref. to SWS_CanSM_00463, SWS_CanSM_00480, SWS_CanSM_00495, SWS_CanSM_00523, SWS_CanSM_00536), the CanSM module shall call the function Dem_SetEventStatus (ref. to chapter 8.5.1) with the parameters EventId := CANSM_E_MODE_REQUEST_TIMEOUT (ref. to chapter 7.3). (SRS_Can_01142)
Change SWS_CanSM_00654: Remove CANSM_E_MODE_REQUEST_TIMEOUT entry
New: SWS_CanSM_xxxx: Add "Extended Production Errors" to Chapter 7.3 with "CANSM_E_MODE_REQUEST_TIMEOUT" entry

Error Name: CANSM_E_MODE_REQUEST_TIMEOUT

Short Description: Mode request for a network failed more often than allowed by configuration

Long Description: The CAN State Manager was not able to change the mode of a CAN network after CanSMModeRequestRepetitionMax retries. It monitors the following CanIf services and the corresponding indications:
CanIf_SetControllerMode() -> CanSM_ControllerModeIndication()
CanIf_SetTrcvMode() -> CanSM_TransceiverModeIndication()
CanIf_CheckTrcvWakeFlag() -> CanSM_CheckTransceiverWakeFlagIndication()
CanIf_ClrTrcvWufFlag() -> CanSM_ClearTrcvWufFlagIndication()

Recommended DTC: Assigned by DEM

Detection Criteria:
Fail: When the CAN State Manager executed any of the CanIf services listed above without receiving the corresponding indication for CanSMModeRequestRepetitionMax times, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status

DEM_EVENT_STATUS_PREFAILED to DEM.
Pass: When CAN State Manager receives any of the indications listed above, it shall report the extended production error CANSM_E_MODE_REQUEST_TIMEOUT with event status DEM_EVENT_STATUS_PREPASSED to DEM.

Secondary Parameters: None
Time Required: Depending on CanSMModeRequestRepetitionMax and CanSMMainFunctionTimePeriod.

Monitor Frequency: Continuously

MIL illumination: Assigned by DEM

Reference: SRS_BSW_00466

==========
ECUC XML:

- In container CanSMDemEventParameterRefs, add reference to DEM event named CANSM_E_MODE_REQUEST_TIMEOUT. Copy other fields from CANSM_E_BUS_OFF.
–Last change on issue 76189 comment 26–

BW-C-Level:

Application	Specification	Bus
1	3	1