

communication

目次 [Table of contents]

- 目次 [Table of contents]
- 図表目次 [Table of figures]
- communication [communication]
 - 機能概要 [Functional overview]
 - 機能詳細 [Functional detail]
 - 機能詳細説明
 - CAN / CAN
 - ユースケースとAPI一覧 [use-case and API lists]
 - 外部要因 ユースケース一覧 [outside factor use-case list]
 - 内部処理 エラーユースケース一覧 [internal processing error use-case list]
 - 起動/終了ユースケース一覧 [Start / end use-case list]
 - ユースケース communication_OnInitialization_001
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_OnInitialization_002
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_OnStart_001 OnStart
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_OnBackgroundStart_001 OnBackgroundStart
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_OnPreStart_001 OnPreStart
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_OnStop_001 OnStop
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_CANThreadStart_001 CANThreadStart
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_CANThreadStop_001 CANThreadStop
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_CAN_Start_Notify_001 CAN Start Notify
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_CanSend_001 [use-case communication_CanSend_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - シーケンス [Sequence]
 - ユースケース communication_CanSend_003 [use-case communication_CanSend_003]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_CanSend_006 [use-case communication_CanSend_006]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_CanSend_007 [use-case communication_CanSend_007]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_CanSend_010 [use-case communication_CanSend_010]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - シーケンス [Sequence]
 - ユースケース communication_CanSend_011 [use-case communication_CanSend_011]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_CanSend_012 [use-case communication_CanSend_012]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_CanSend_013 [use-case communication_CanSend_013]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケース communication_CanRecv_001 [use-case communication_CanRecv_001]
 - 概要 [Overview]

- シーケンス [Sequence]
- ユースケース communication_CanRecv_002 [use-case communication_CanRecv_002]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケース communication_CanRecv_003 [use-case communication_CanRecv_003]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - シーケンス説明 [Sequence Details]
- ユースケース communication_CanRecv_005 [use-case communication_CanRecv_005]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケース communication_CanRecv_009 [use-case communication_CanRecv_009]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケース communication_CanRecv_015 [use-case communication_CanRecv_015]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケース communication_CanRecv_016 [use-case communication_CanRecv_016]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケース communication_CanCommandCtrl_001 [use-case communication_CanCommandCtrl_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - シーケンス [Sequence]
 - シーケンス [Sequence]
 - シーケンス [Sequence]
- ユースケース communication_CanCommWatch_001 [use-case communication_CanCommWatch_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - シーケンス [Sequence]

図表目次 [Table of figures]

- 表.ユニット概要
 - 表. 外部要因ユースケース一覧 [Table. External factor use case lists]
 - 表. 内部処理 エラーユースケース一覧 [Table. internal processing error use-case list]
 - 表. 起動/終了ユースケース一覧 [Start / end use-case list]

communication [communication]

機能概要 [Functional overview]

表.ユニット概要

ユニット名[Unit Name]	コンポーネント名[Component Name]	ユニット概要[Description]	オーナーディレクトリ[Owner Dir]
communication	PeripheralService	<p>communicationは次の通信機能を持つ。</p> <p>(1) CAN [CAN]</p> <p>can_halと繋ぎ、CANデータの通信機能を提供する</p> <p><i>Support the CAN data communication function by can_hal.</i></p>	communication

機能詳細 [Functional detail]

communicationユニットは、SystemManagerより起動するプロセスである。

利用者に対して、CANは独立のAPIを提供する。

通信データと要求結果があれば、Native Service Frameworkを使用してメッセージの通信を行う。

communication unit is provided as a process. This is the first process to be started by init process and mainly provides launch function for the resident services.

communication provides CAN API for users.

If there is any data or result received, communication will send it to users by using Native Service Framework.

(1) CAN [CAN]

- can_halへCANデータを送信する。
Send CAN data to can_hal.
- can_halからCANデータを受信する。
Receive CAN data from can_hal.
- CANデータ通信途絶・復帰を監視する。
Check the can data communication suspension or resume.
- CANコマンド制御処理。

Transmission control of CAN command.
- CANサービスのAvailabilityを通知する。
Notifies the CAN service availability.
- 送信したCANデータを配送する(エコーバック)
Distribute CAN data transmitted by _CWORD86_.

機能詳細説明

本APIで提供する機能の詳細を以下に示す。

Function details of this guide are as follows.

CAN / CAN

No	機能 / functions	詳細説明 / Detail explanation
1	CANデータ送信 <i>Send the CAN data</i>	CANデータ / CANデータ(_CWORD30_, _CWORD29_)を送信する <i>Send the CAN data / CAN data(_CWORD30_, _CWORD29_)</i> CANデータの送信結果を返信する <i>Return the send status to user</i>
2	CANデータ配送登録処理 <i>Delivery registration of CAN data</i>	CANデータの配送を登録する <i>Delivery registration of CAN data</i>
3	CANデータ受信 <i>Receive the CAN data</i>	CANデータを受信する <i>Receive the CAN data</i>
4	CANデータ通信途絶・復帰を監視する <i>Check the can data communication suspension or resume</i>	通信途絶時間を設定し、通信途絶が発生する場合は、外部に通知する <i>Set the suspension time.If communication suspend is occured, notify the suspend event.</i> 通信途絶から復帰する場合は、外部に通知する <i>If communication resume from suspend, notify the resume event</i>

5	CANコマンド制御処理 <i>Transmission control of CAN command</i>	下記のコマンド制御を行う <ul style="list-style-type: none"> • CAN起動完了通知要求 / <i>StartUp Finish Request</i> • CANマスタリセット情報通知要求送信 / <i>CAN Master Reset information</i> • CANバージョン要求送信 / <i>CAN Version request</i> • CAN接続ノード通知要求送信 / <i>Request CAN connected node notification</i> • CAN区間燃費リセット要求配送 / <i>Can Fuel calculation request</i> • CAN区間燃費リセット応答送信 / <i>Can Fuel calculation request ACK</i>
---	---	--

ユースケースとAPI一覧 [use-case and API lists]

外部要因 ユースケース一覧 [outside factor use-case list]

表. 外部要因ユースケース一覧 [Table. External factor use case lists]

ユースケース番号 [Use Case Number]	機能カテゴリ [function category]	ユースケース名 [use-case name]	本ユースケースを実行可能な状態	ユースケースを実現するAPI [API that realizes use-case]	備考 [notes]
communication_CanSend_001	CanSend	CANデータ送信処理 <i>Start to send CAN data transmission</i>	Global_CAN_open	Canif_TransmissionStart CAN送信結果構造体 <i>CAN send result struct:</i> CAN_MSG_SENDSTS	receive message : CID : CID_CAN_TX_RESULT 配信元 / Source : LAN_SERVICE_CAN
communication_CanSend_003	CanSend	CANデータ(Bit)送信処理 <i>Start to send the CAN data (Bit)</i>	Global_CAN_open	Canif_TransStart CAN送信結果構造体 <i>CAN send result struct:</i> CAN_MSG_SENDSTS	receive message : CID : CID_CAN_TX_RESULT 配信元 / Source : LAN_SERVICE_CAN
communication_CanSend_006	CanSend	CANデータ (_CWORD30_)送信通知処理 <i>Start to send the CAN data (_CWORD30_)</i>	Global_CAN_open	Canif_CWORD30_Transmission CAN送信結果構造体 <i>CAN send result struct:</i> CAN_MSG_SENDSTS	receive message : CID : CID_CAN_TX_RESULT 配信元 / Source : LAN_SERVICE_CAN
communication_CanSend_007	CanSend	CANデータ (_CWORD30_)送信シーケンス終了通知処理 <i>Start to send the CAN data (_CWORD30_)</i>	Global_CAN_open	Canif_CWORD30_TxseqFinTransmission CAN送信結果構造体 <i>CAN send result struct:</i> CAN_MSG_SENDSTS	receive message : CID : CID_CAN_TX_RESULT 配信元 / Source : LAN_SERVICE_CAN

communication_CanSend_010	CanSend	<u>_CWORD29_データ送信通知処理</u> <i>Start to send the _CWORD29_data</i>	Global_CAN_open	Canif_Diag_CWORD29_Transmission <u>_CWORD29_データ送信結果構造体</u> <i>_CWORD29_data send result struct:</i> CAN_MSG_CWORD29_SENDDSTS	receive message : CID : CID_CAN_CWORD29_TX_RESULT 配信元 / Source : LAN_SERVICE_CAN
communication_CanSend_011	CanSend	<u>送信データエコーバック処理</u> <i>echo back functionality for can data transmit from _CWORD86_.</i>	Global_CAN_open	Canif_TransStart	
communication_CanSend_012	CanSend	<u>_CWORD30_送信データエコーバック処理</u> <i>echo back functionality for can data transmit from _CWORD86_.</i>	Global_CAN_open	Canif_CWORD30_Transmission Canif_CWORD30_TxsEqFinTransmission	
communication_CanSend_013	CanSend	<u>_CWORD29_送信データエコーバック処理</u> <i>echo back functionality for can data transmit from _CWORD86_.</i>	Global_CAN_open	Canif_CWORD29_Transmission	
communication_CanRecv_001	CanRecv	<u>CANデータ配送登録処理</u> <i>Delivery registration of CAN data</i>	Global_CAN_open	Canif_DeliveryEntry	
communication_CanRecv_002	CanRecv	<u>CANコマンド受信</u> <i>Receive the CAN command</i>	Global_CAN_open	Canif_DeliveryEntry CANコマンド受信構造体 <i>CAN command receive struct:</i> CAN_MSG_CANCMD	receive message : CID : CID_CAN_CMD_DELIVERY 配信元 / Source : LAN_SERVICE_CAN
communication_CanRecv_003	CanRecv	<u>CANデータ受信</u> <i>Receive the CAN data</i>	Global_CAN_open	Canif_DeliveryEntry CANデータ受信構造体 <i>CAN data receive struct:</i> CAN_MSG_CANDATA	receive message : CID : CID_CAN_DATA_DELIVERY 配信元 / Source : LAN_SERVICE_CAN
communication_CanRecv_005	CanRecv	<u>CANデータ(_CWORD30_)受信</u> <i>Receive the CAN data (_CWORD30_)</i>	Global_CAN_open	Canif_DeliveryEntry CANデータ受信構造体 <i>CAN data receive struct:</i> CAN_MSG_CANDATA	receive message : CID : CID_CAN_CWORD30_DELIVERY 配信元 / Source : LAN_SERVICE_CAN
communication_CanRecv_009	CanRecv	<u>配送登録先全消去</u> <i>Delete all the Delivery registration of CAN data</i>	Global_CAN_open	Canif_DeliveryEraseAll	

communication_CanRecv_015	CanRecv	<u>_CWORD29_データ受信</u> <i>Receive the _CWORD29_data</i>	Global_CAN_open	Canif_Diag_CWORD29_DeliveryEntry <u>_CWORD29_データ受信構造体</u> <u>_CWORD29_data receive struct:</u> CAN_MSG_CAN_CWORD29_DATA	receive message : CID : CID_CAN_CWORD29_DELIVERY 配信元 / Source : LAN_SERVICE_CAN
communication_CanRecv_016	CanRecv	<u>_CWORD29_データ配送登録処理</u> <i>Delivery registration of _CWORD29_data</i>	Global_CAN_open	Canif_Diag_CWORD29_DeliveryEntry	
communication_CanCommandCtrl_001	CanCommandCtrl	<u>CANコマンド制御処理</u> <i>Start to ctrl CAN command</i>	Global_CAN_open	Canif_CommandCtrl <ul style="list-style-type: none"> CANコマンド受信構造体 <i>CAN command receive struct:</i> CAN_MSG_CANCMD CAN送信結果構造体 <i>CAN send result struct:</i> CAN_MSG_SENDSTS 	receive message : <ul style="list-style-type: none"> CID : CID_CAN_CMD_DELIVERY 配信元 / Source : LAN_SERVICE_CAN CID : CID_CAN_TX_RESULT 配信元 / Source : LAN_SERVICE_CAN
communication_CanCommWatch_001	CanCommWatch	<u>CANデータ通信途絶監視処理</u> <i>Start to watch CAN data</i>	Global_CAN_open	Canif_CommWatch <ul style="list-style-type: none"> CAN通信途絶構造体 <i>CAN communication stop struct</i> CAN_MSG_COMM_WATCHSTS CAN通信復帰構造体 <i>CAN communication restore struct</i> CAN_MSG_COMM_WATCHSTS 	receive message : <ul style="list-style-type: none"> CID : CID_CAN_COMM_STOP 配信元 / Source : LAN_SERVICE_CAN CID : CID_CAN_COMM_RESTORE 配信元 / Source : LAN_SERVICE_CAN
communication_CanCommWatch_002	CanCommWatch	<u>CANデータ通信途絶</u> <i>CAN communication suspension</i>	Global_CAN_open	Canif_DeliveryEntry Canif_CommWatch CAN通信途絶構造体: <u>CAN communication stop struct:</u> CAN_MSG_COMM_WATCHSTS CAN通信復帰構造体: <u>CAN communication restore struct:</u> CAN_MSG_COMM_WATCHSTS CANデータ受信構造体: <u>CAN data receive struct:</u> CAN_MSG_CANDATA	

内部処理 エラーユースケース一覧 [internal processing error use-case list]

表. 内部処理 エラーユースケース一覧 [Table. internal processing error use-case list]

エラー番号 [error number]	機能カテゴリ [function category]	ユースケース名 [use-case name]	戻り値 [return value]	備考 [notes]
----------------------	----------------------------	-------------------------	--------------------	------------

communication_CAN_Error_001	CAN	内部処理エラー (_CWORD78_SendMsg /shm_open/mmap /McSend /_CWORD78_InvokeSync) <i>Internal processing error</i> (_CWORD78_SendMsg /shm_open/mmap /McSend /_CWORD78_InvokeSync)	CANIF_RET_ERROR_CANCEL	
-----------------------------	-----	---	------------------------	--

起動/終了ユースケース一覧 [Start / end use-case list]

表. 起動/終了ユースケース一覧 [Start / end use-case list]

ユースケース番号 [Use Case Number]	ユースケース名 [use-case name]	備考 [notes]
communication_OnInitialization_001	process create	-
communication_OnInitialization_002	OnInitialization	-
communication_OnStart_001	OnStart	-
communication_OnBackgroundStart_001	OnBackgroundStart	-
communication_OnPreStart_001	OnPreStart	-
communication_OnBackgroundStop_001	OnBackgroundStop	-
communication_OnPreStop_001	OnPreStop	-
communication_OnStop_001	OnStop	-
communication_CANThreadStart_001	CANThreadStart	-
communication_CANThreadStop_001	CANThreadStop	-
communication_CAN_Start_Notify_001	CAN Start Notify	-

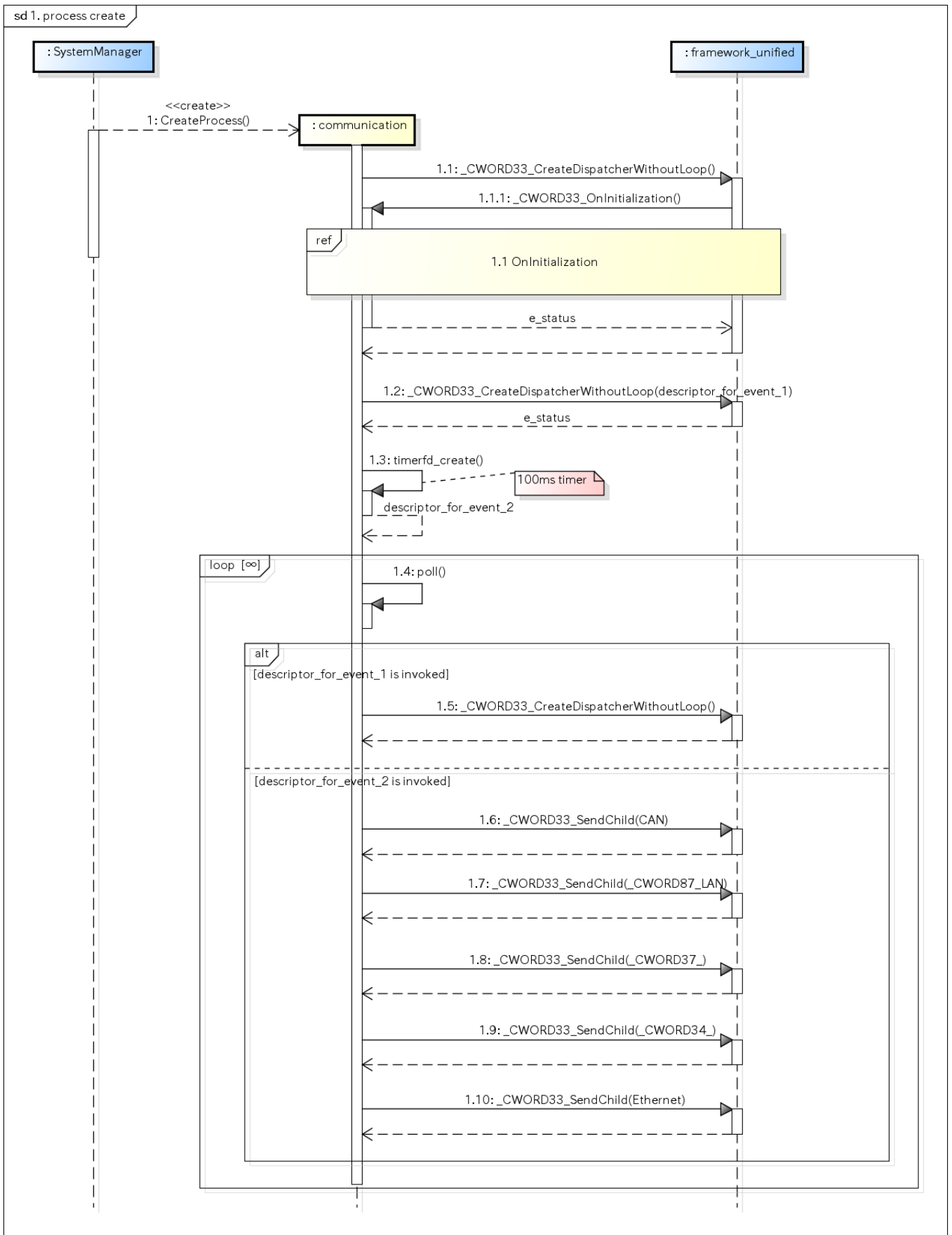
ユースケース communication_OnInitialization_001

概要 [Overview]

communicationプロセスの生成時の処理

[Processing when creating a communication process](#)

シーケンス [Sequence]

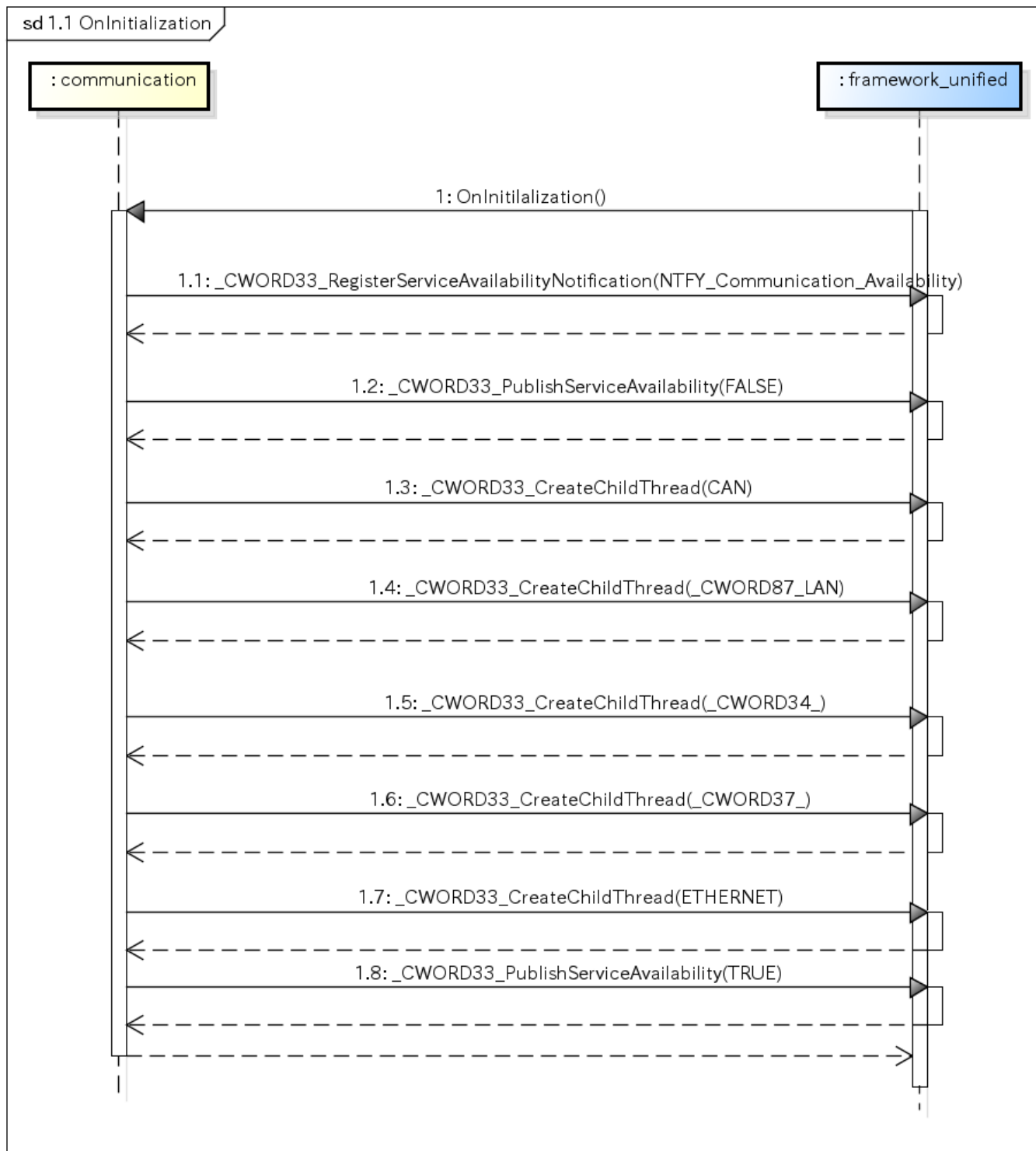


ユースケース communication_OnInitialization_002

概要 [Overview]

OnInitialization呼び出し時の処理
Processing when OnInitialization is called

シーケンス [Sequence]



ユースケース communication_OnStart_001 OnStart

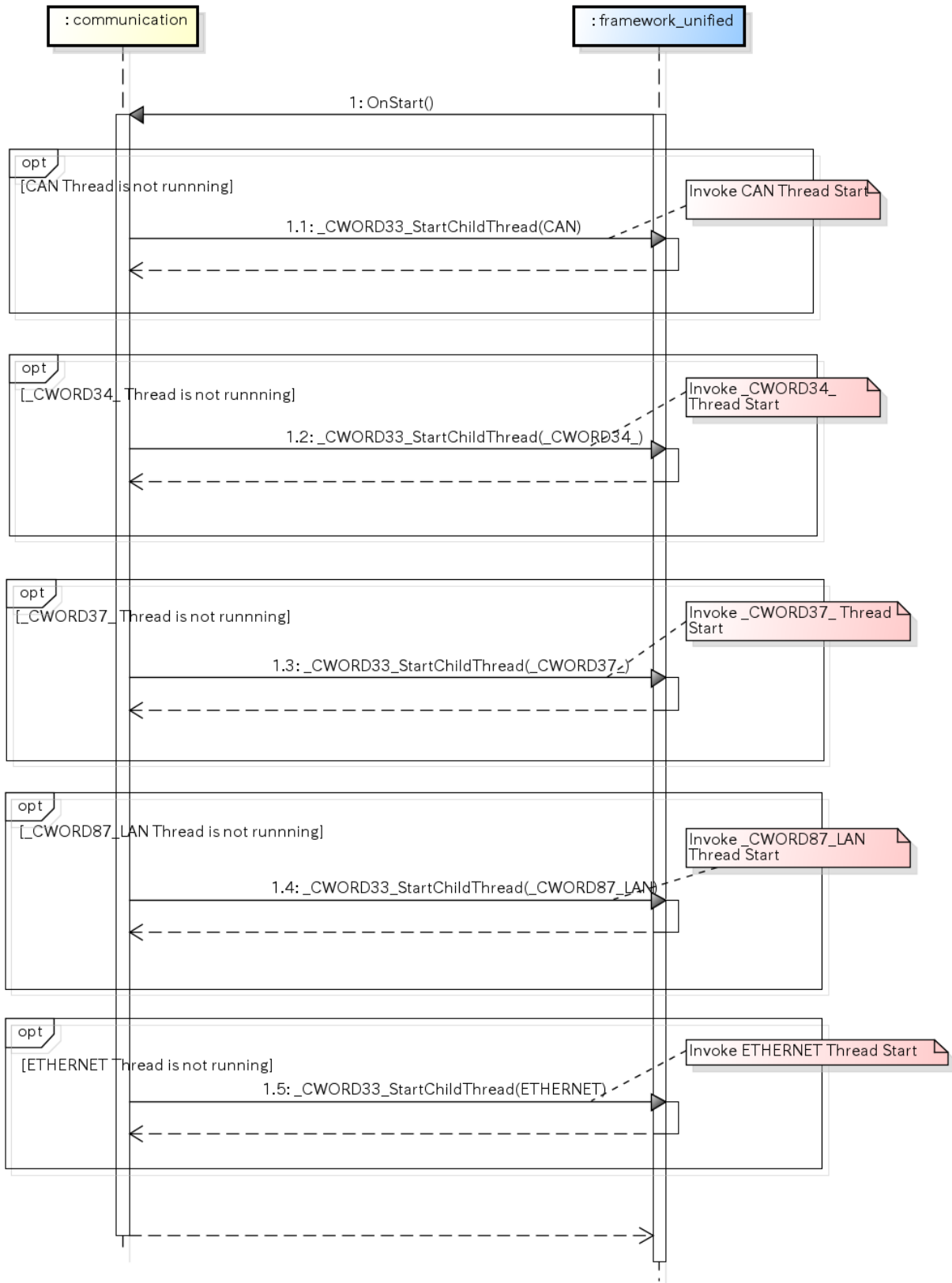
概要 [Overview]

通信路の管理スレッドをStartさせる

Start the communication path management thread

シーケンス [Sequence]

sd 2.1 OnStart



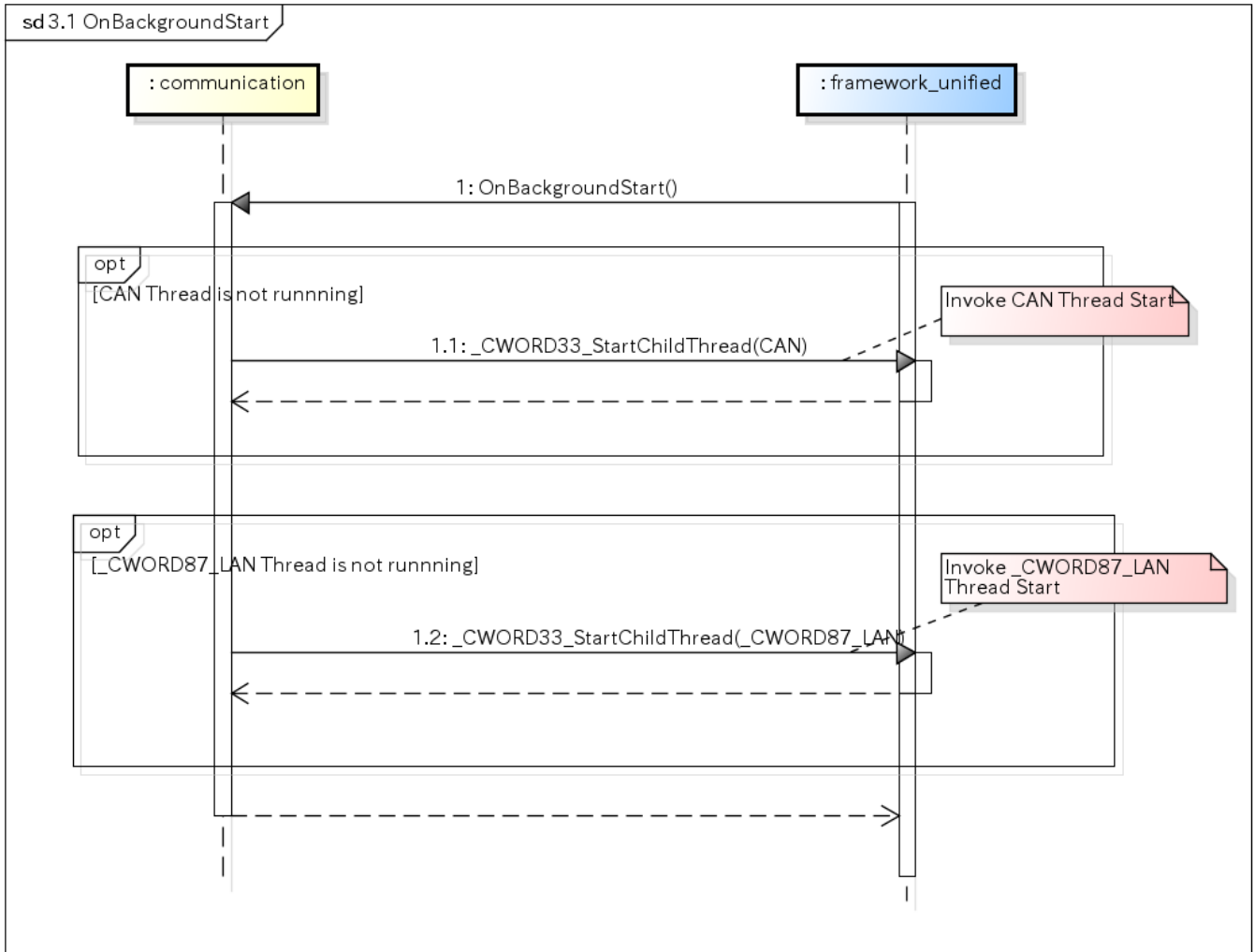
ユースケース communication_OnBackgroundStart_001 OnBackgroundStart

概要 [Overview]

通信路の管理スレッドをStartさせる

Start the communication path management thread

シーケンス [Sequence]



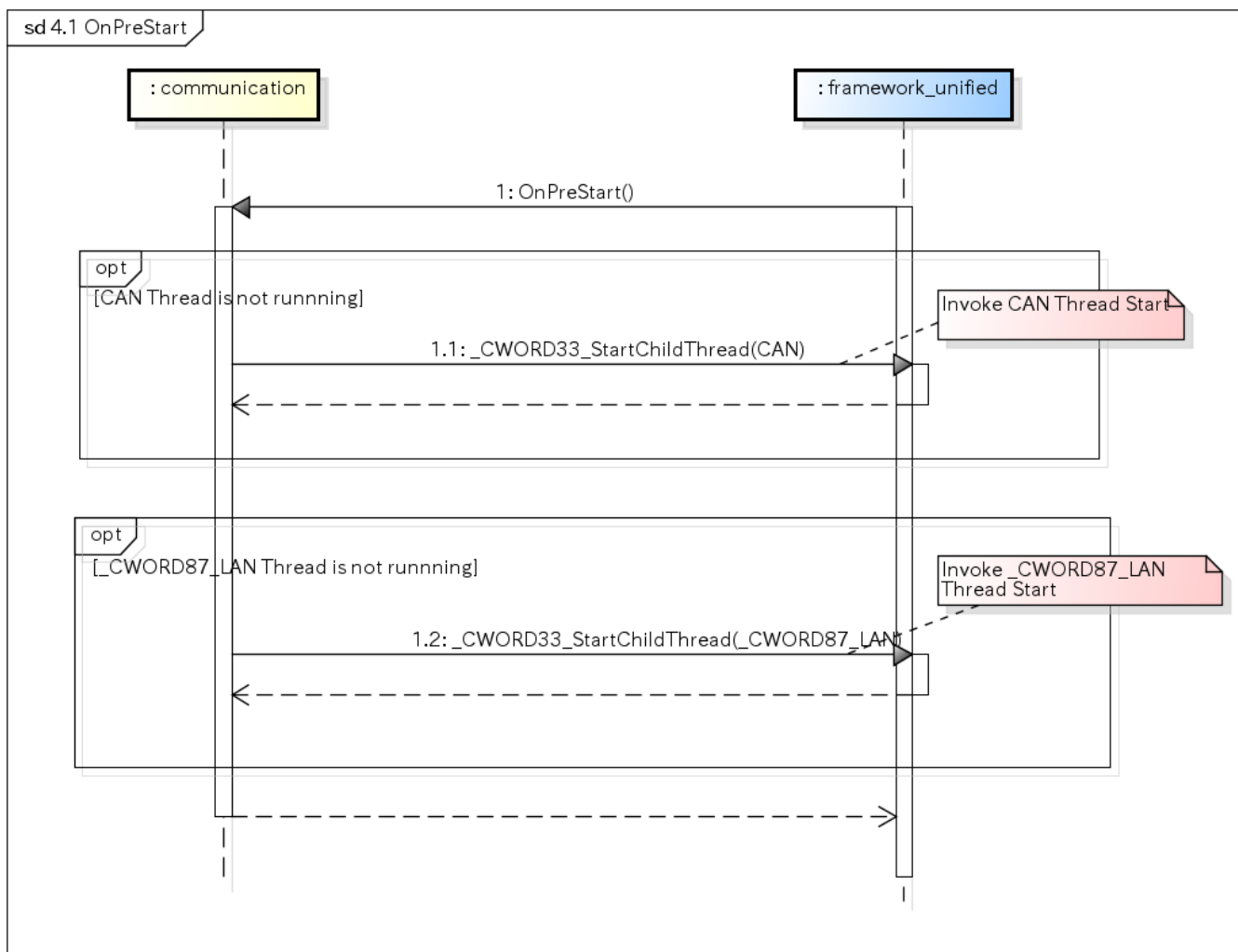
ユースケース communication_OnPreStart_001 OnPreStart

概要 [Overview]

通信路の管理スレッドをStartさせる。

Start the communication path management thread

シーケンス [Sequence]



ユースケース communication_OnStop_001 OnStop

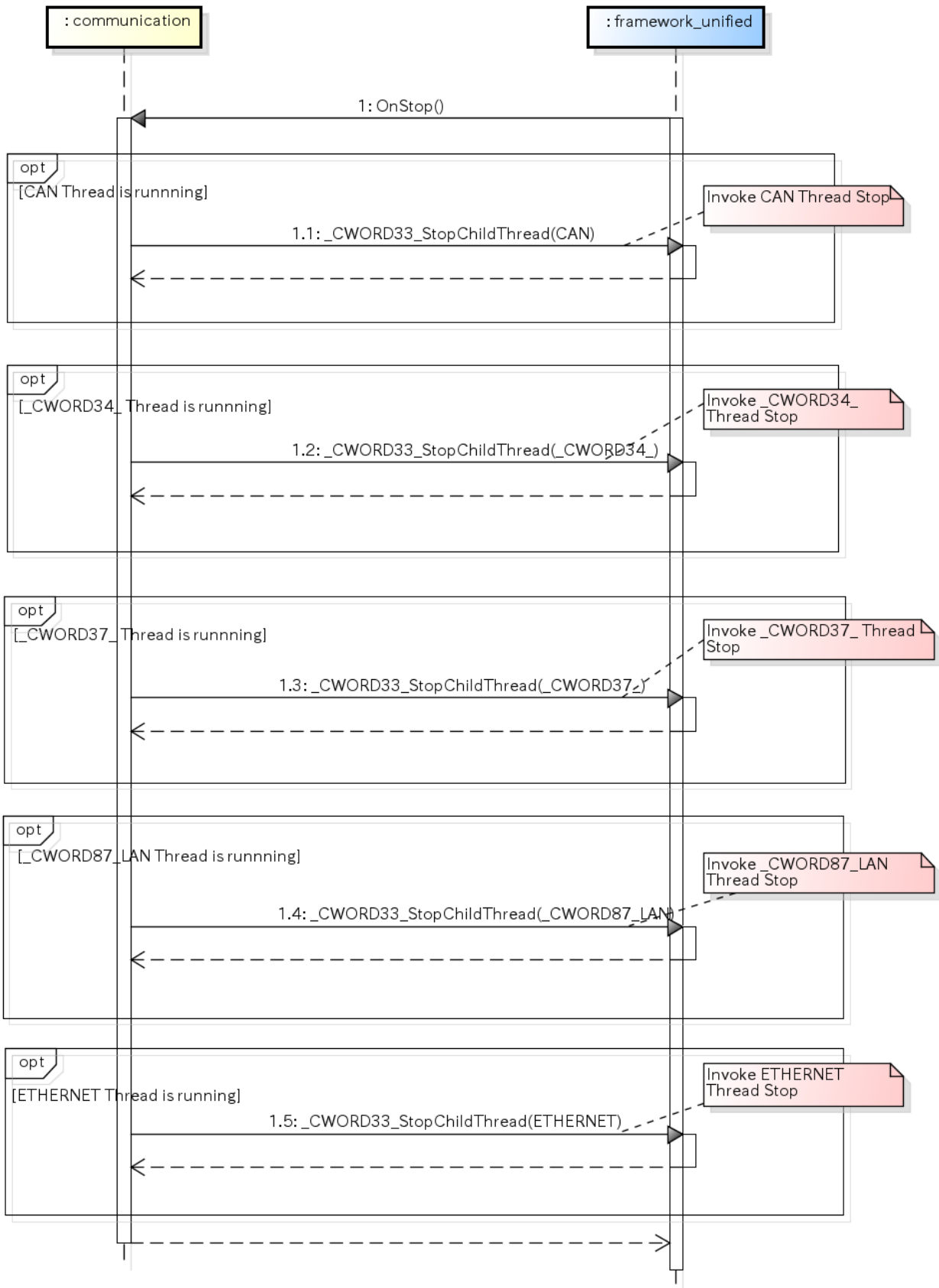
概要 [Overview]

な通信路の管理スレッドをStopさせる

Stop the communication path management thread

シーケンス [Sequence]

sd7.1 OnStop



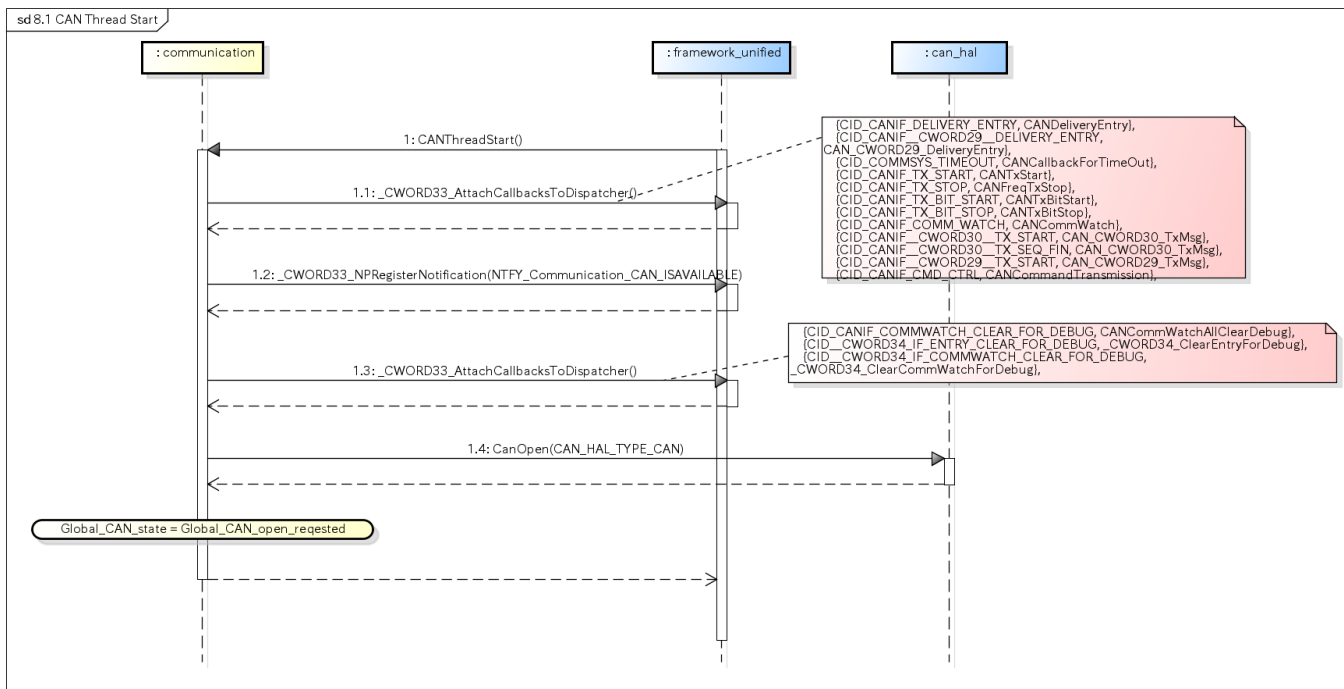
ユースケース communication_CANThreadStart_001 CANThreadStart

概要 [Overview]

CAN Threadの開始処理

CAN Thread start processing

シーケンス [Sequence]



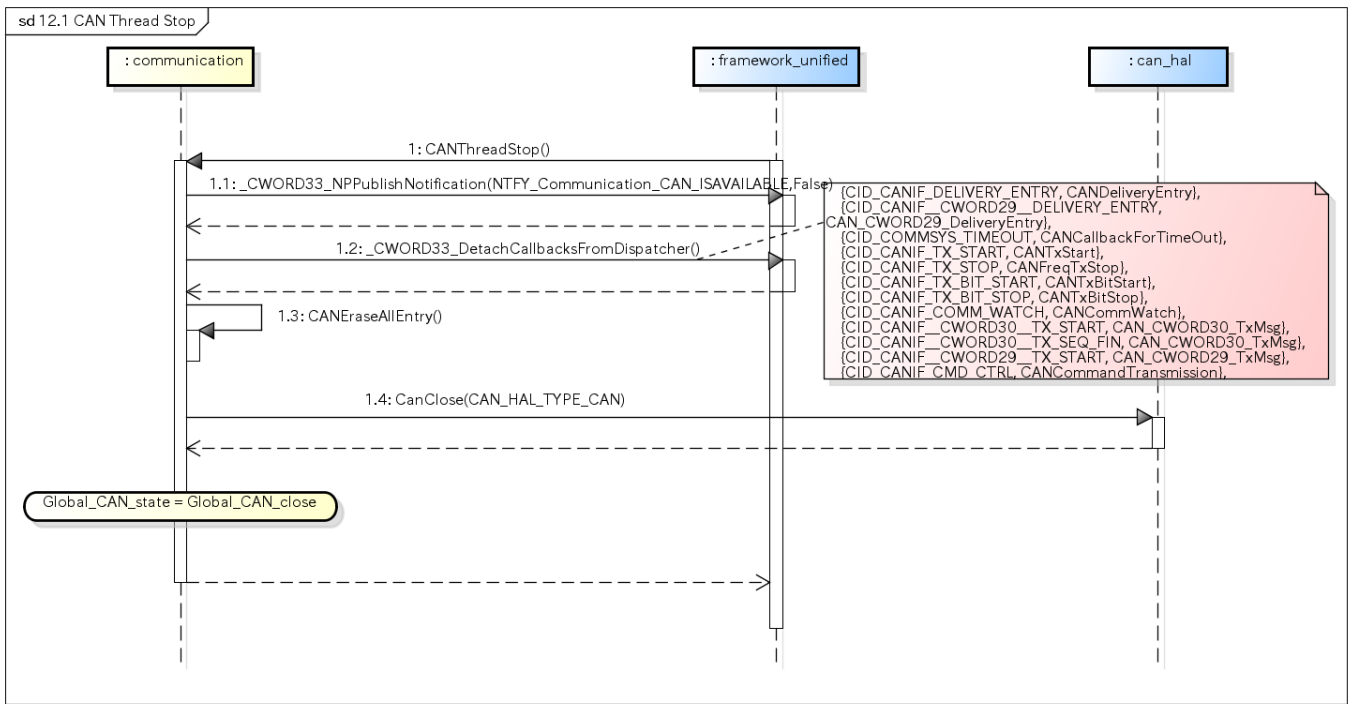
ユースケース communication_CANThreadStop_001 CANThreadStop

概要 [Overview]

CAN Threadの終了処理

CAN Thread start processing

シーケンス [Sequence]



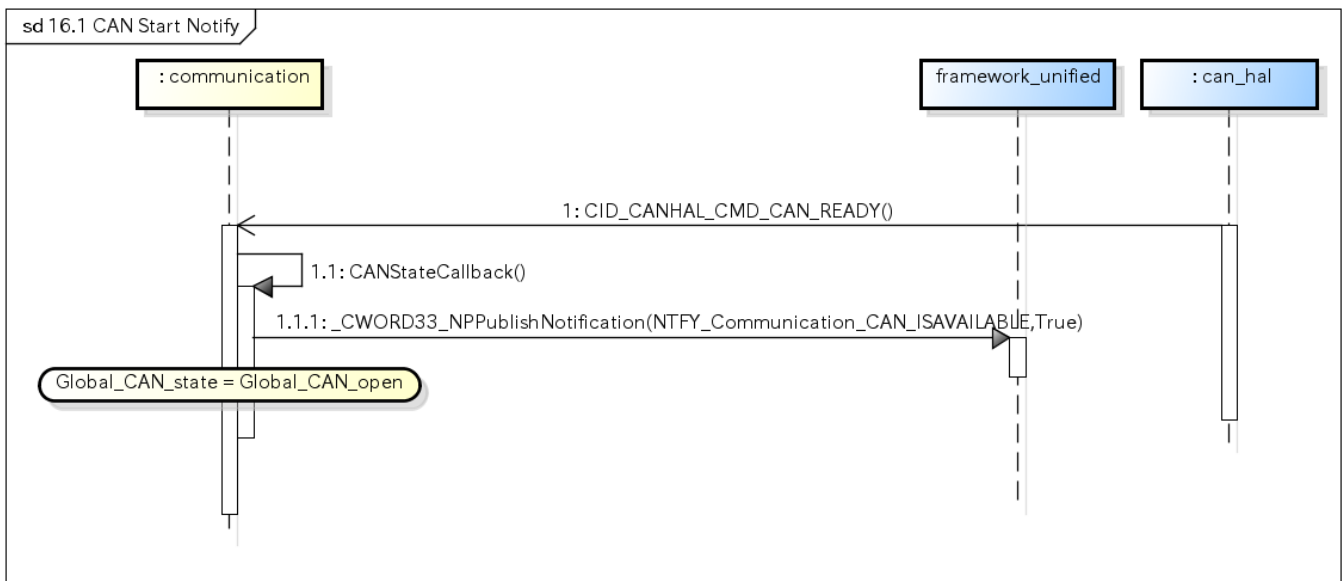
ユースケース communication_CAN_Start_Notify_001 CAN Start Notify

概要 [Overview]

CANの利用可通知受信

CAN availability notification received

シーケンス [Sequence]

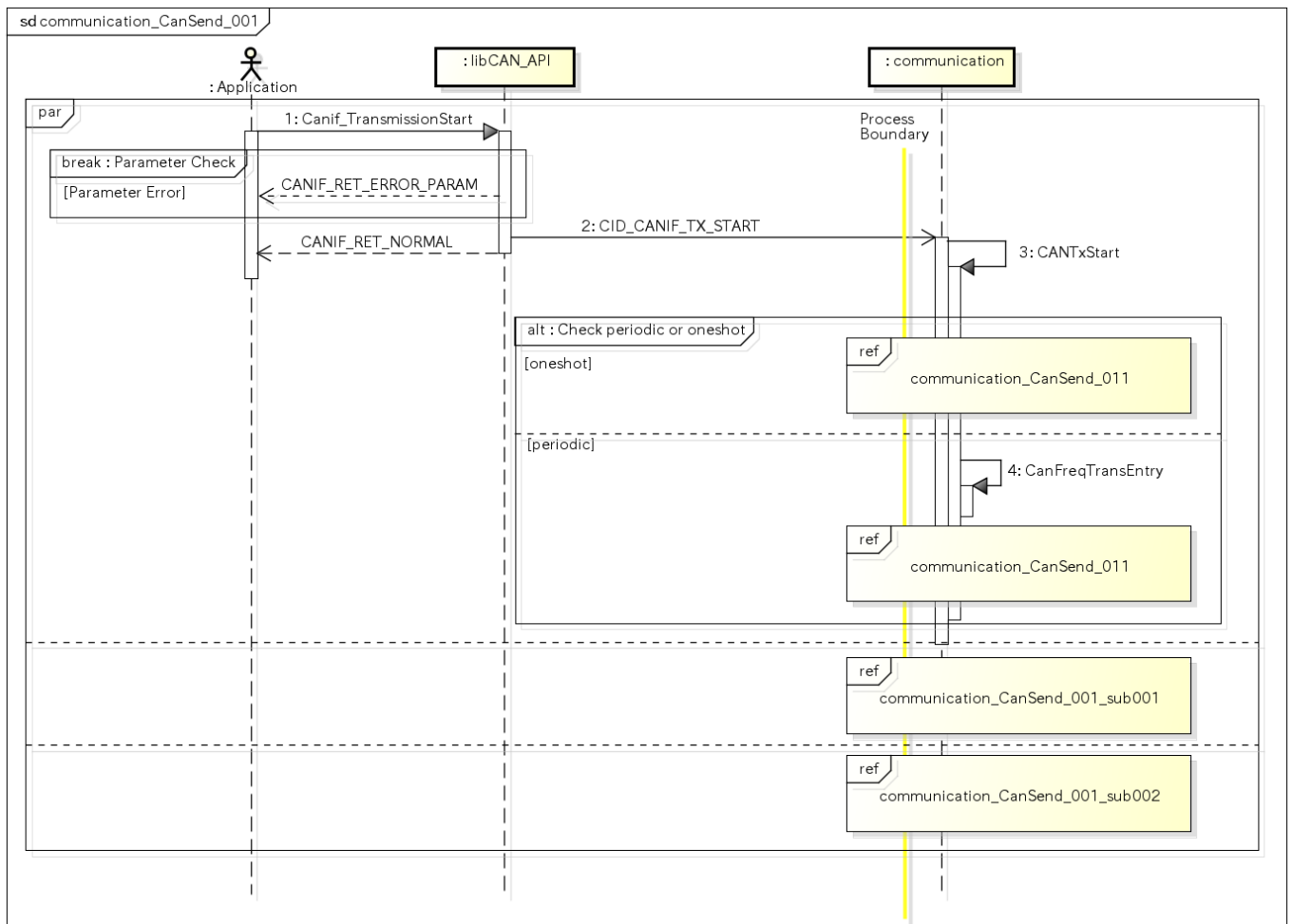


ユースケース communication_CanSend_001 [use-case communication_CanSend_001]

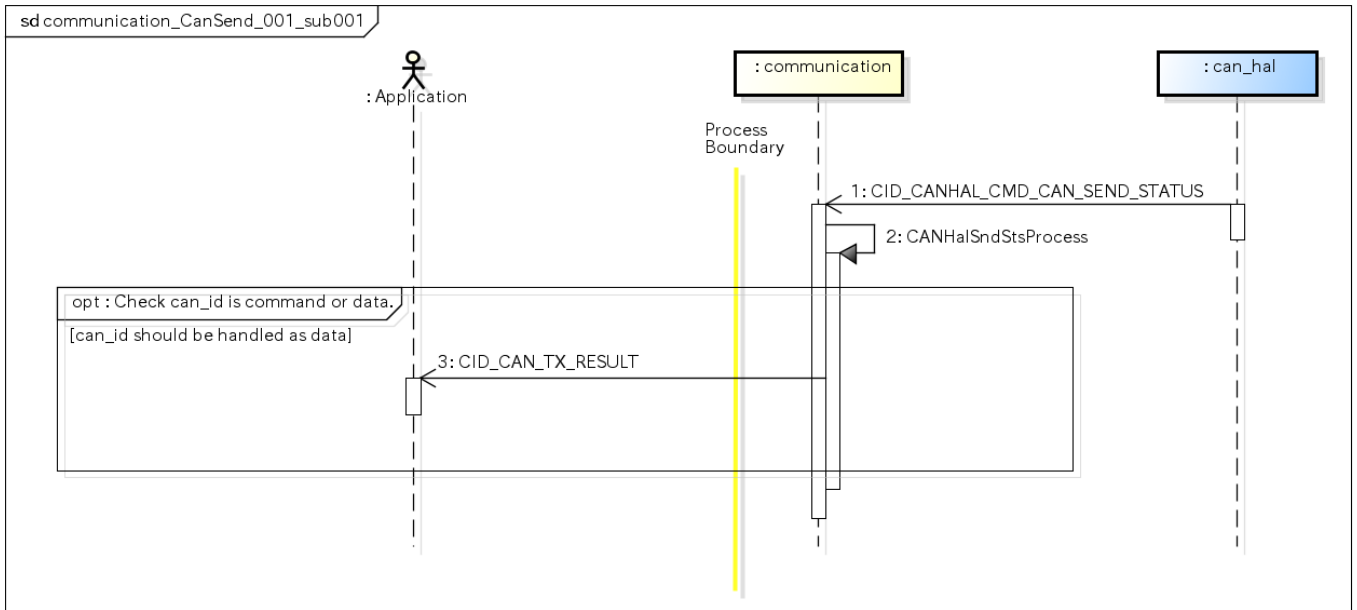
概要 [Overview]

- CANデータ送信処理
Start to send CAN data transmission
- 送信結果を取得する
Get the result of transmission

シーケンス [Sequence]



シーケンス [Sequence]

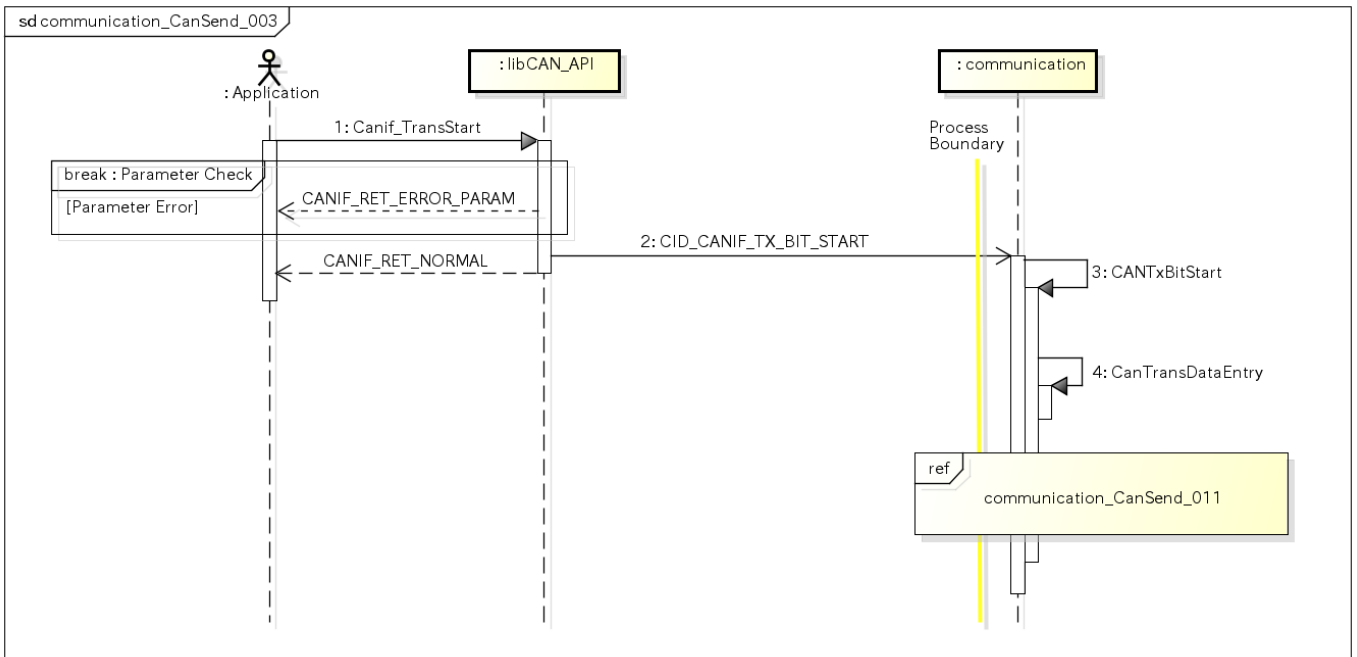


ユースケース communication_CanSend_003 [use-case communication_CanSend_003]

概要 [Overview]

- CANデータ(Bit)送信処理
Start to send the CAN data (Bit)
- 送信結果を取得する
Get the result of transmission

シーケンス [Sequence]

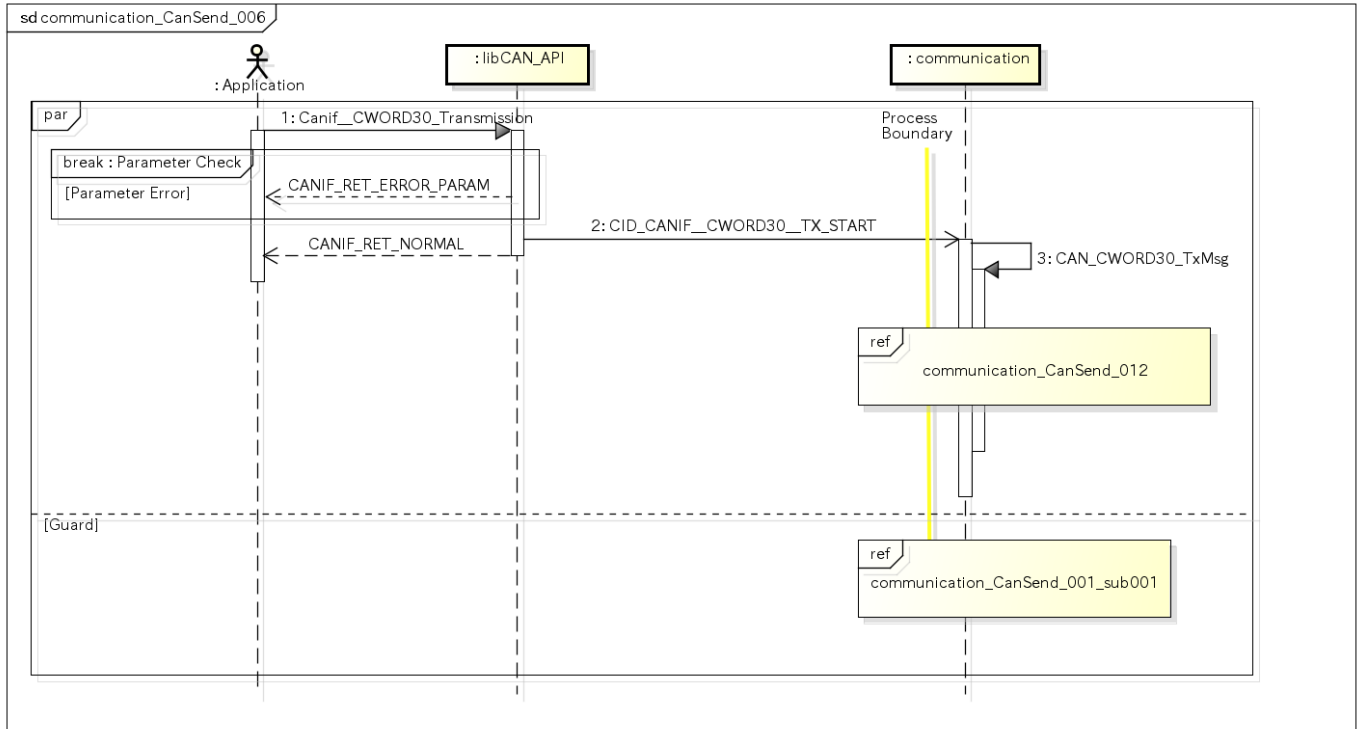


ユースケース communication_CanSend_006 [use-case communication_CanSend_006]

概要 [Overview]

- CANデータ(_CWORD30_)送信通知処理
Start to send the CAN data (_CWORD30_)
- 送信結果を取得する
Get the result of transmission

シーケンス [Sequence]

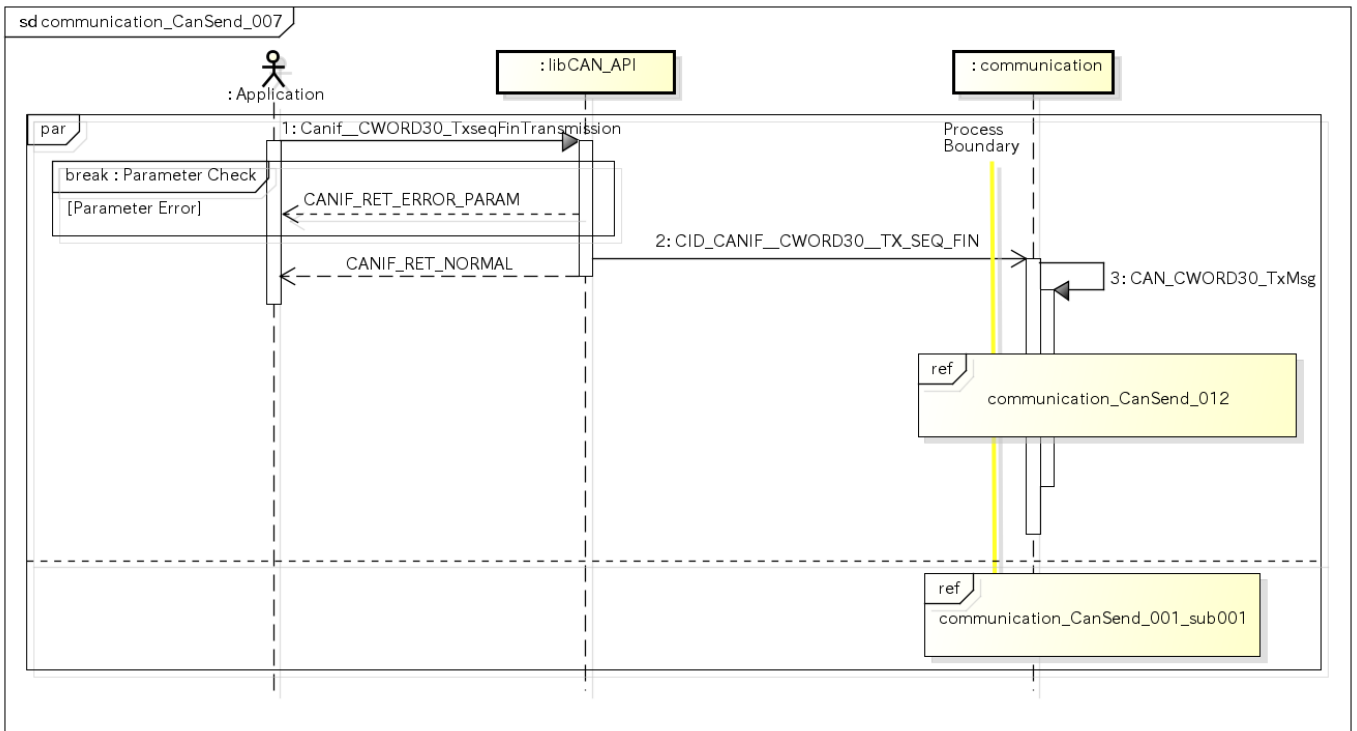


ユースケース communication_CanSend_007 [use-case communication_CanSend_007]

概要 [Overview]

- CANデータ(_CWORD30_)送信シーケンス終了通知処理
Stop to send the CAN data (_CWORD30_)
- 送信結果を取得する
Get the result of transmission

シーケンス [Sequence]

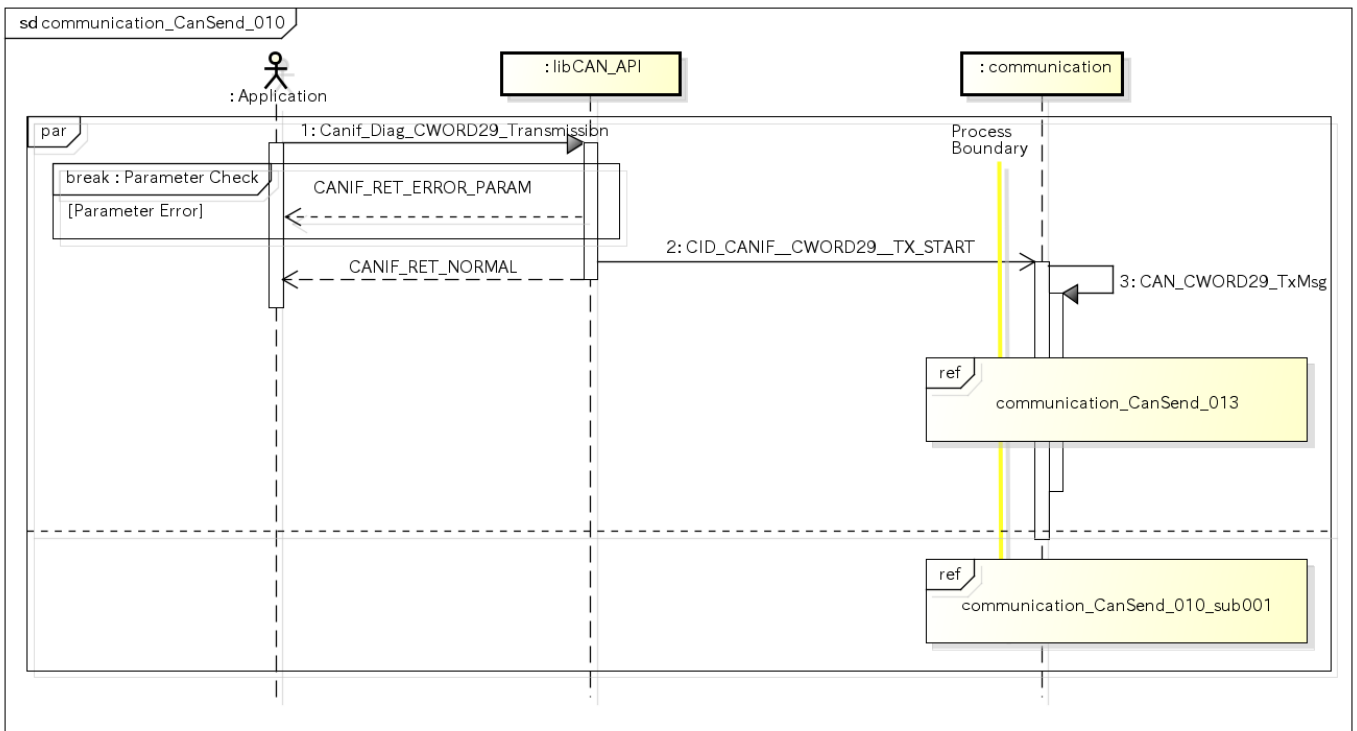


ユースケース communication_CanSend_010 *[use-case communication_CanSend_010]*

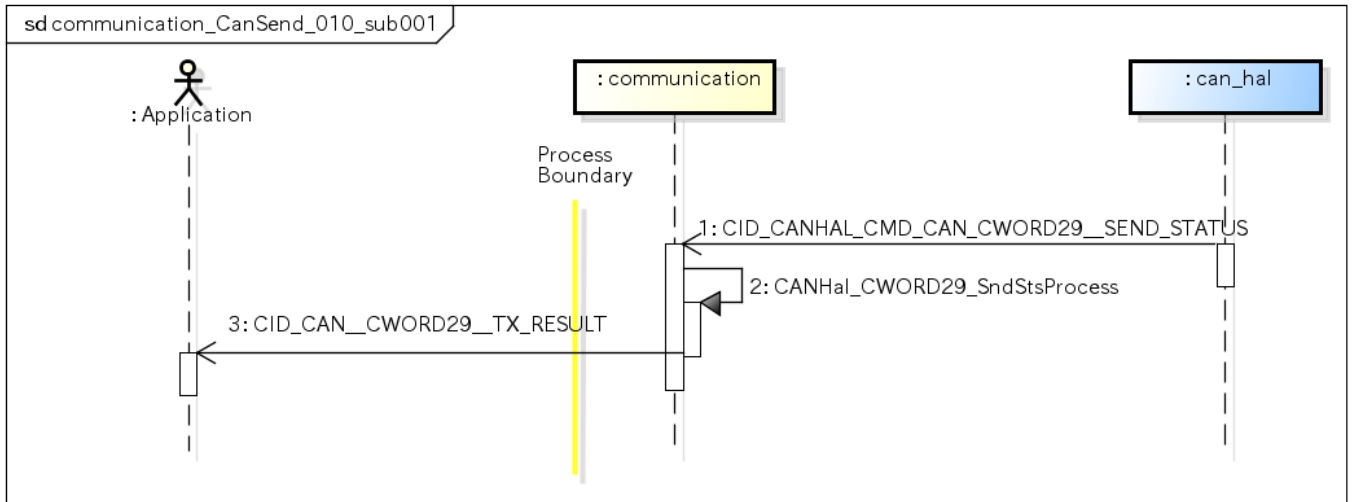
概要 *[Overview]*

- *_CWORD29_データ送信通知処理*
Start to send the _CWORD29_data
- *送信結果を取得する*
Get the result of transmission

シーケンス *[Sequence]*



シーケンス [Sequence]

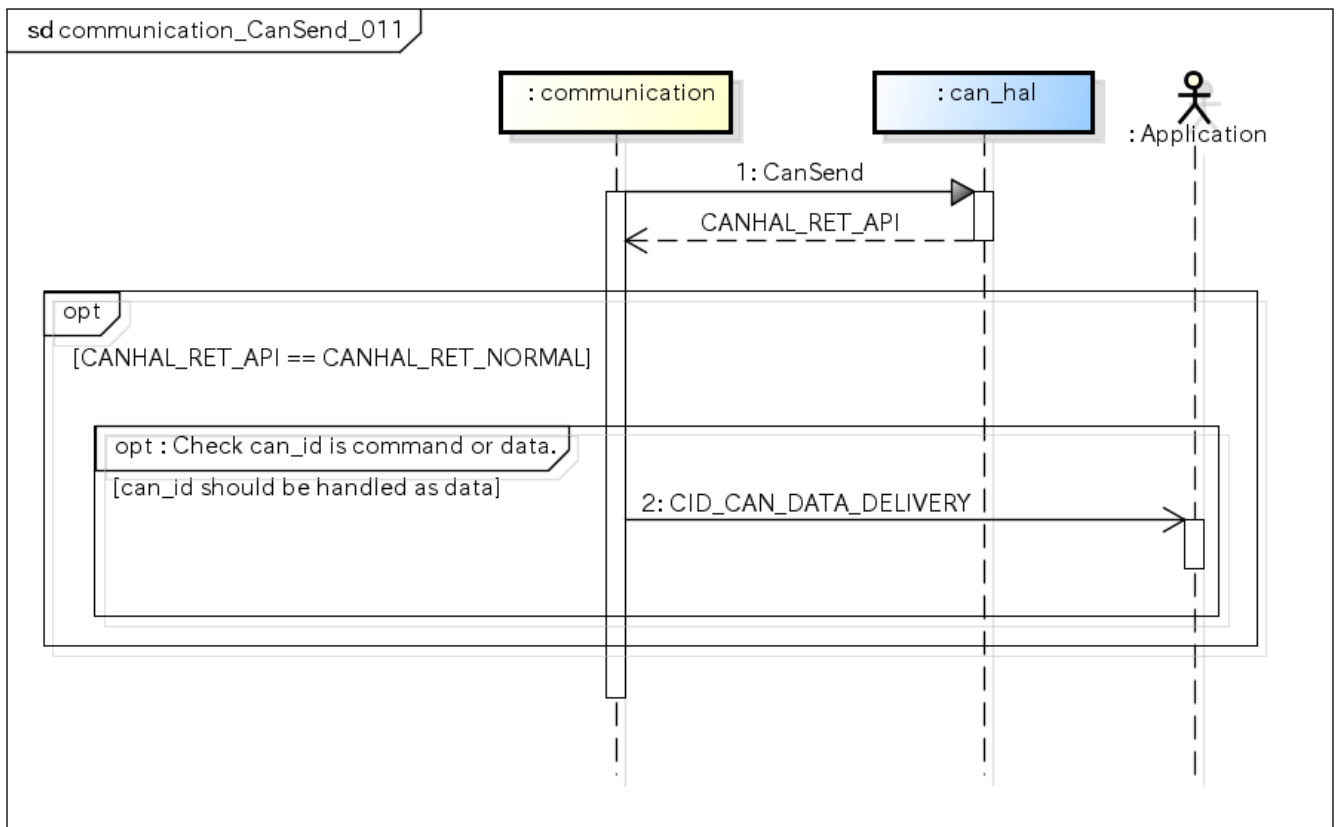


ユースケース communication_CanSend_011 [use-case communication_CanSend_011]

概要 [Overview]

- 送信データエコーバック処理 (TECHWG-645)
echo back functionality for can data transmit from _CWORD86_.

シーケンス [Sequence]

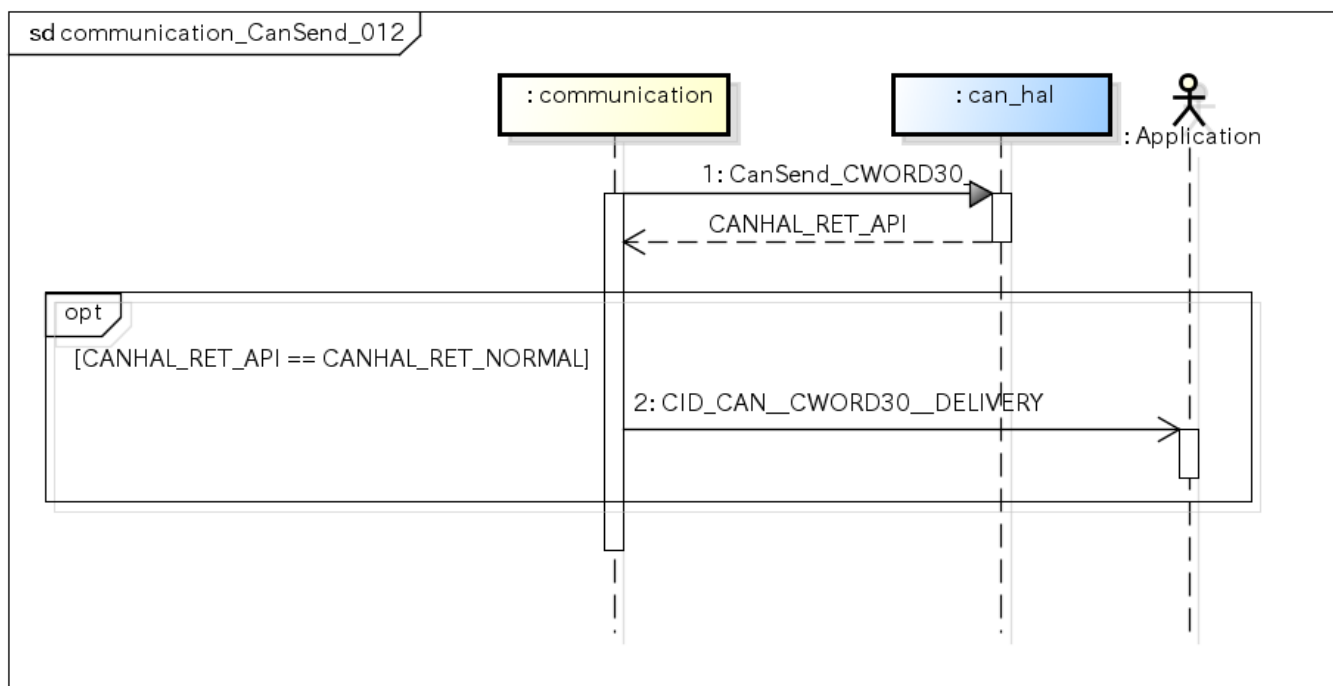


ユースケース communication_CanSend_012 [use-case communication_CanSend_012]

概要 [Overview]

- _CWORD30_ 送信データエコーバック処理 (TECHWG-645)
echo back functionality for can data transmit from _CWORD86_.

シーケンス [Sequence]

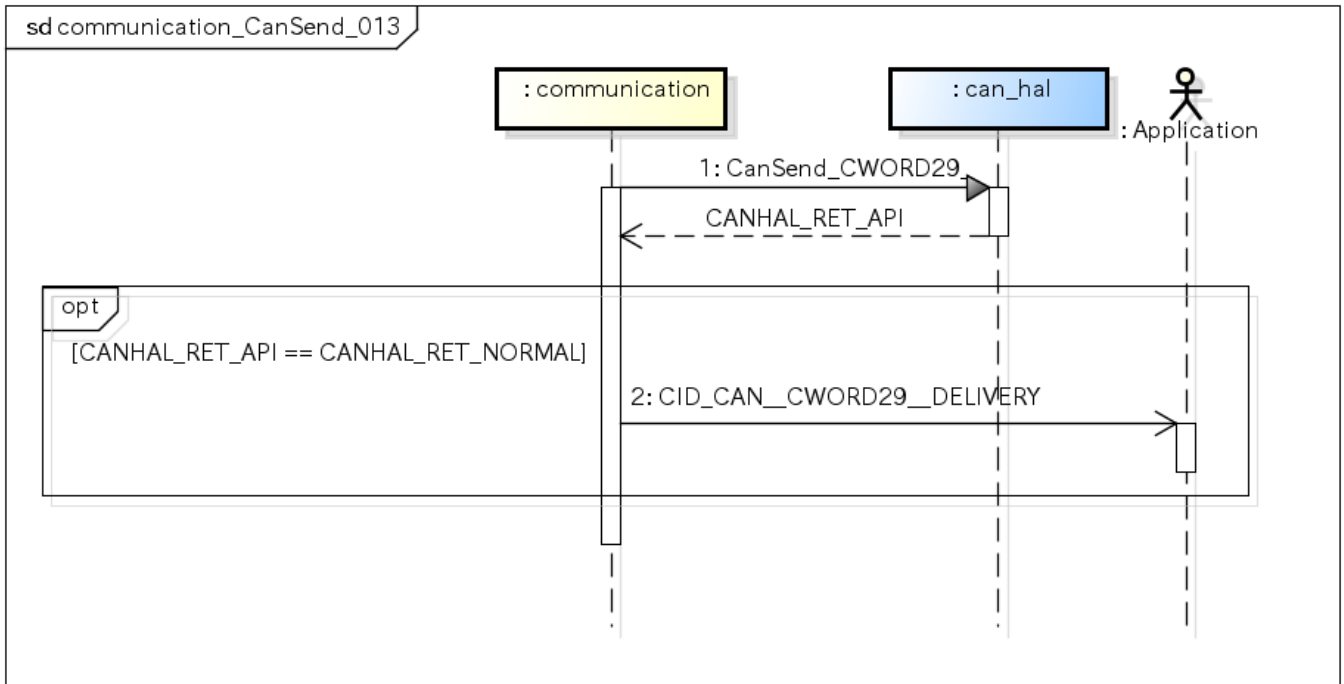


ユースケース communication_CanSend_013 [use-case communication_CanSend_013]

概要 [Overview]

- _CWORD29_ 送信データエコーバック処理 (TECHWG-645)
echo back functionality for can data transmit from _CWORD86_.

シーケンス [Sequence]



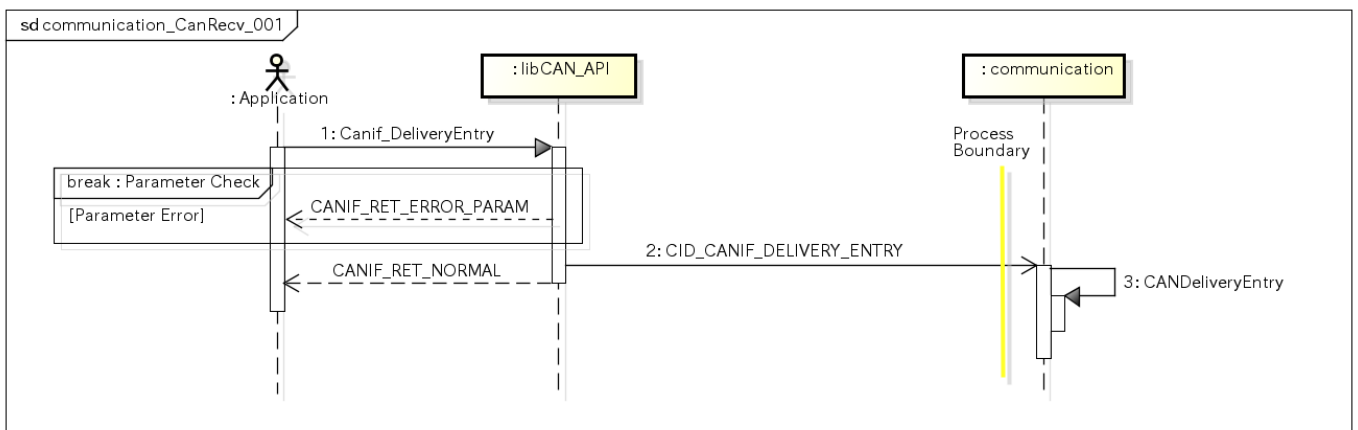
ユースケース communication_CanRecv_001 *[use-case communication_CanRecv_001]*

概要 *[Overview]*

CANデータ配送登録処理。送信データエコーバック処理対象のCANフレーム(communication_CanSend_011、 communication_CanSend_012)も本ユースケースを利用し配送登録する。

Delivery registration of CAN data. Can data transmit from _CWORD86_ is handled same as Can data received from can_hal.

シーケンス *[Sequence]*



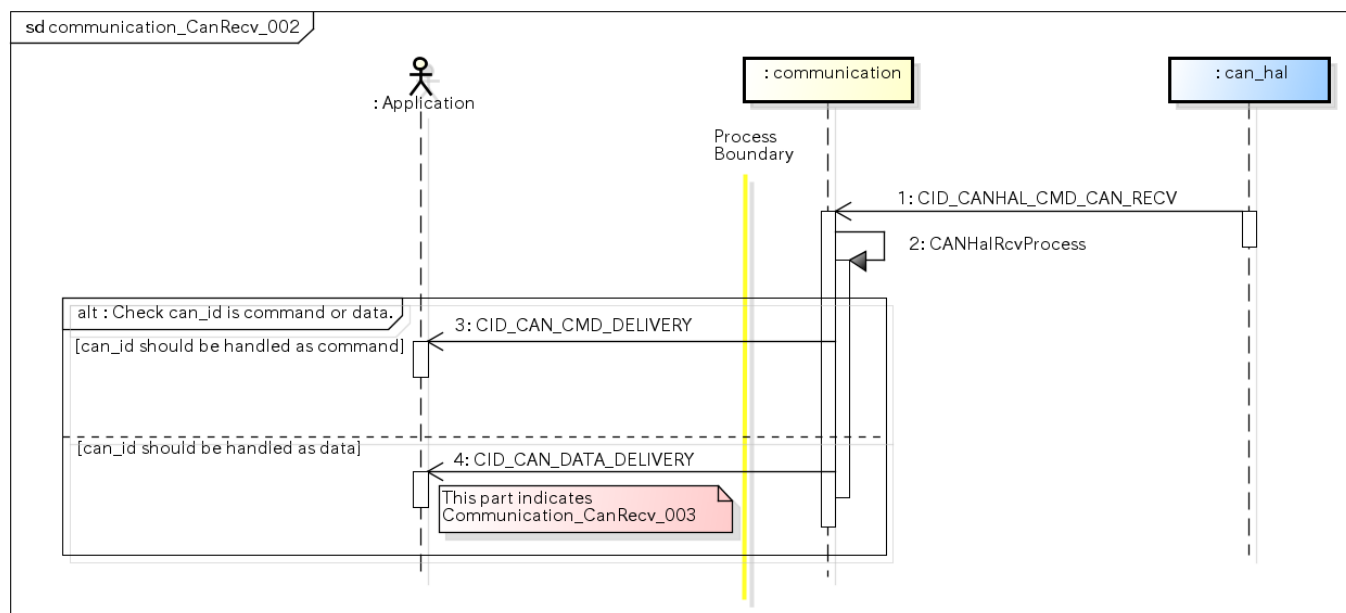
ユースケース communication_CanRecv_002 *[use-case communication_CanRecv_002]*

概要 *[Overview]*

CANコマンド受信

Receive the CAN command

シーケンス [Sequence]



ユースケース communication_CanRecv_003 [use-case communication_CanRecv_003]

概要 [Overview]

CANデータ受信

Receive the CAN data

シーケンス [Sequence]

can_idで分岐するためCommunication_CanRecv_002と同等のシーケンスとなる。従ってCommunication_CanRecv_002を参照。

シーケンス説明 [Sequence Details]

Communication_CanRecv_002を参照

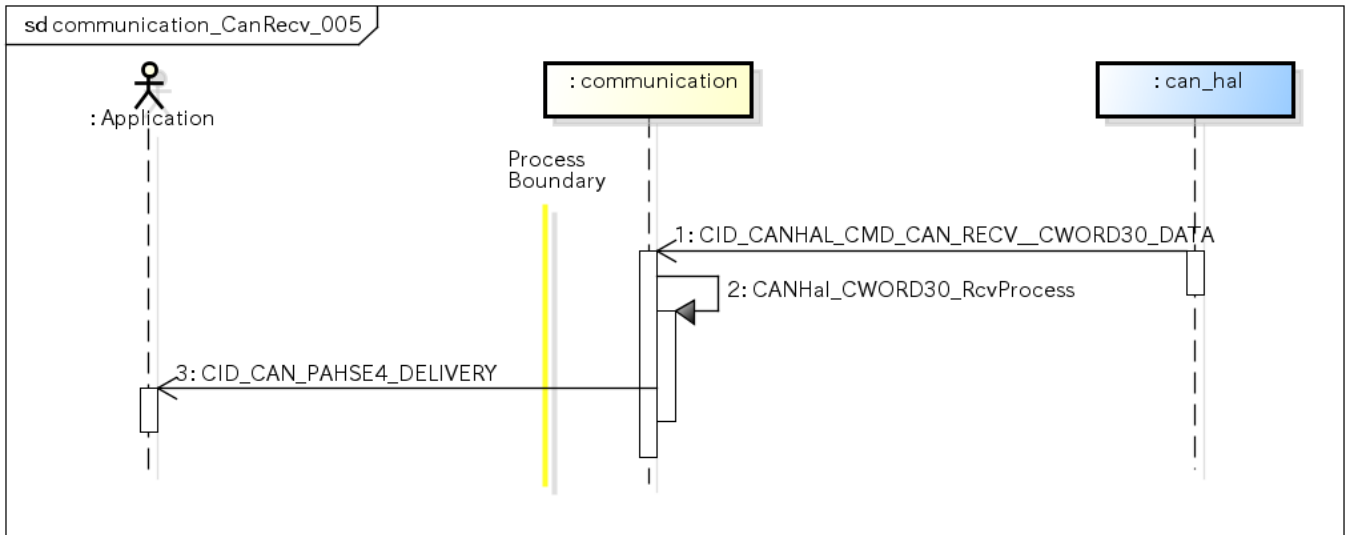
ユースケース communication_CanRecv_005 [use-case communication_CanRecv_005]

概要 [Overview]

CANデータ(_CWORD30_)受信

Receive the CAN data (_CWORD30_)

シーケンス [Sequence]



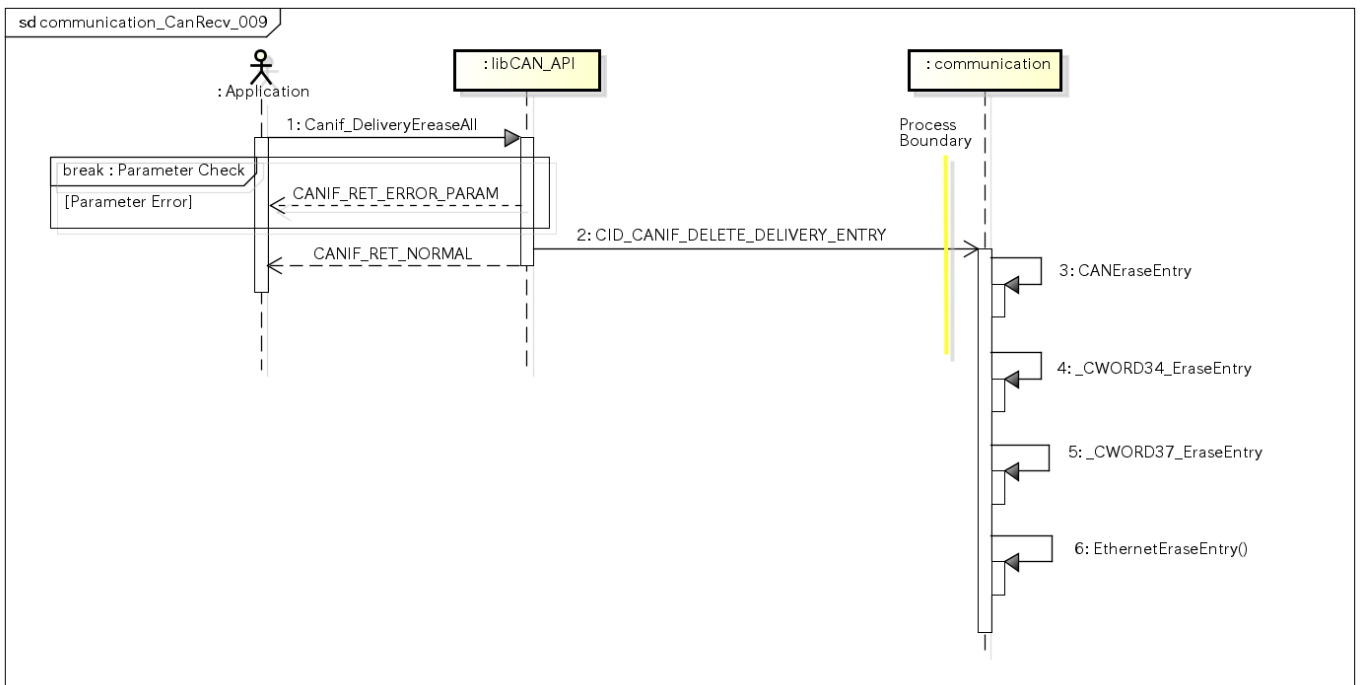
ユースケース communication_CanRecv_009 [use-case communication_CanRecv_009]

概要 [Overview]

配送登録先全消去

Delete all the Delivery registration of CAN data

シーケンス [Sequence]

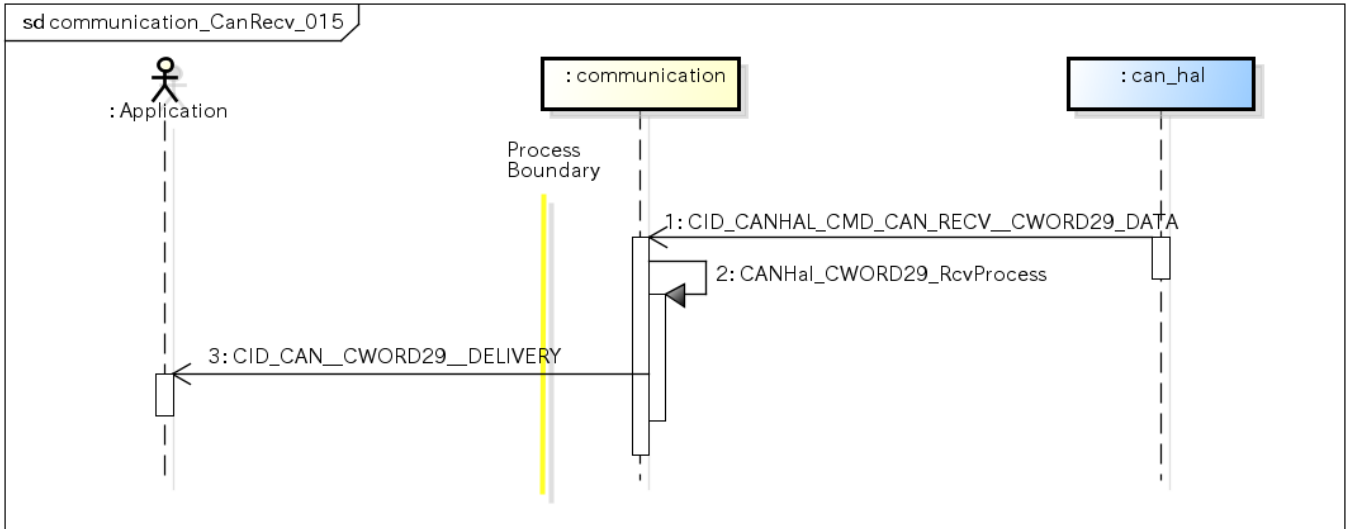


ユースケース communication_CanRecv_015 [use-case communication_CanRecv_015]

概要 [Overview]

_CWORD29_データ受信

シーケンス [Sequence]



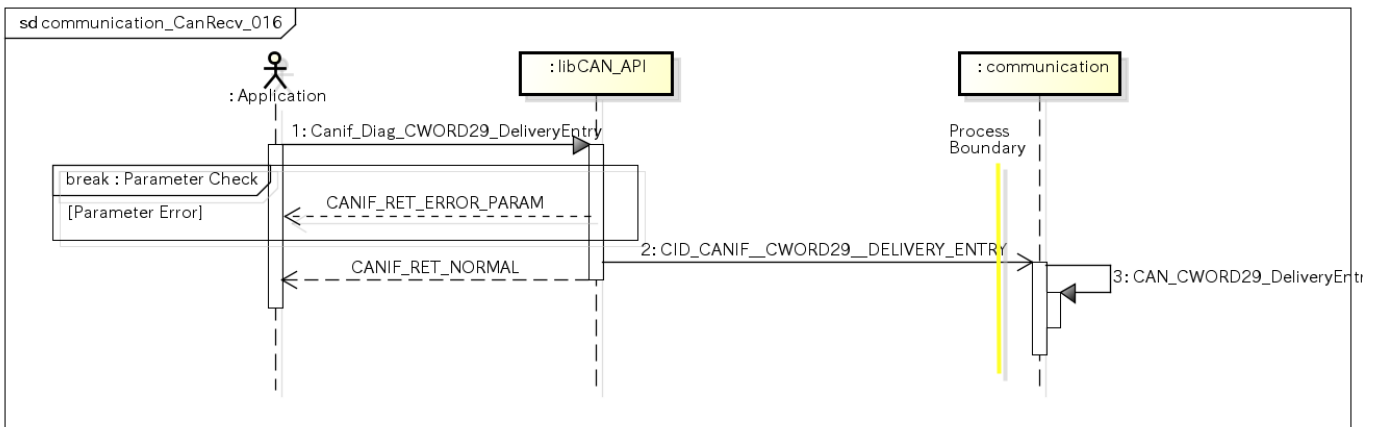
ユースケース communication_CanRecv_016 [use-case communication_CanRecv_016]

概要 [Overview]

_CWORD29_データ配送登録処理。送信データエコーバック処理対象のCANフレーム(communication_CanSend_013)も本ユースケースを利用し配送登録する。

Delivery registration of _CWORD29_data. Can data transmit from _CWORD86_ is handled same as Can data received from can_hal.

シーケンス [Sequence]



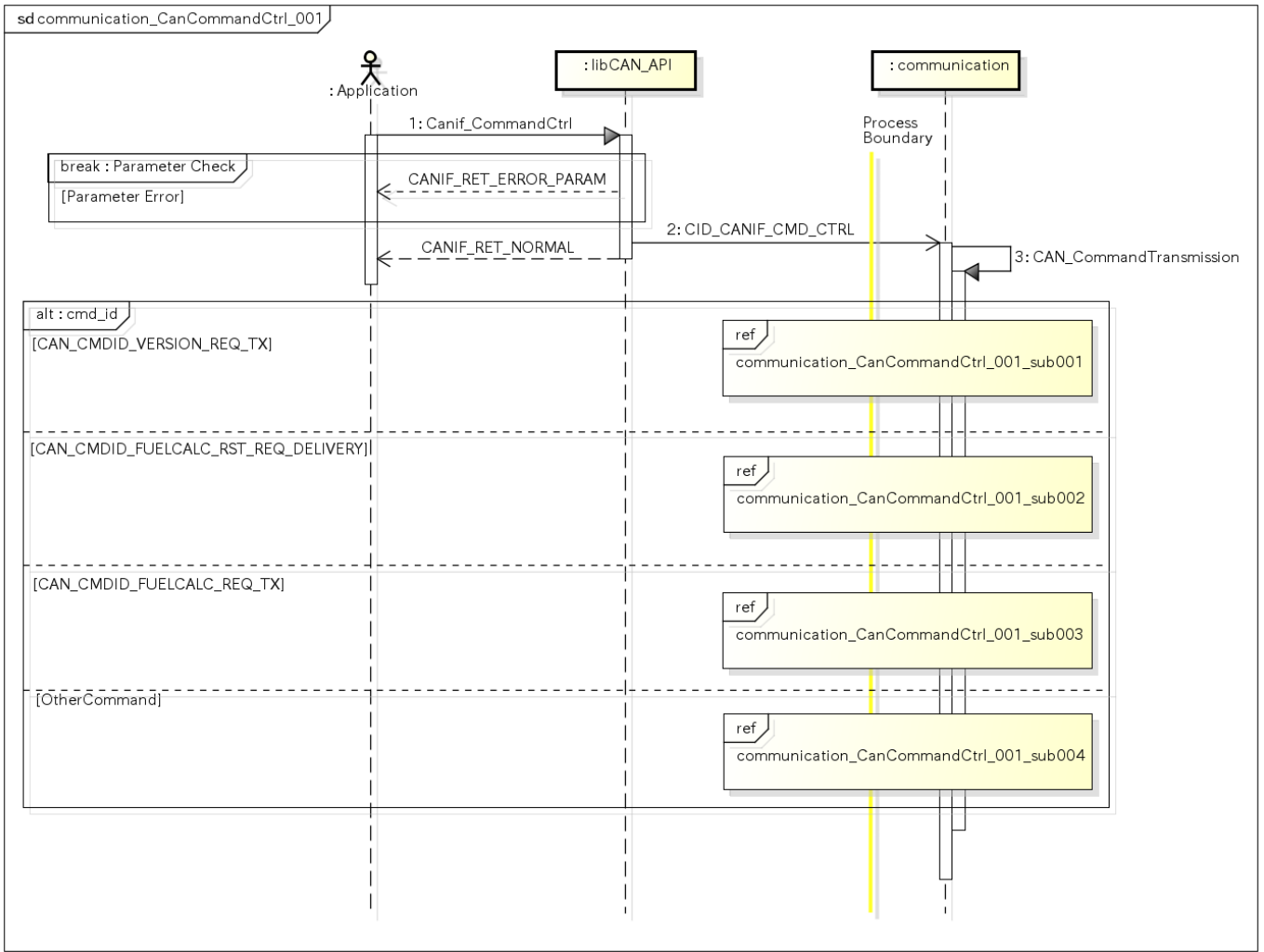
ユースケース communication_CanCommandCtrl_001 [use-case communication_CanCommandCtrl_001]

概要 [Overview]

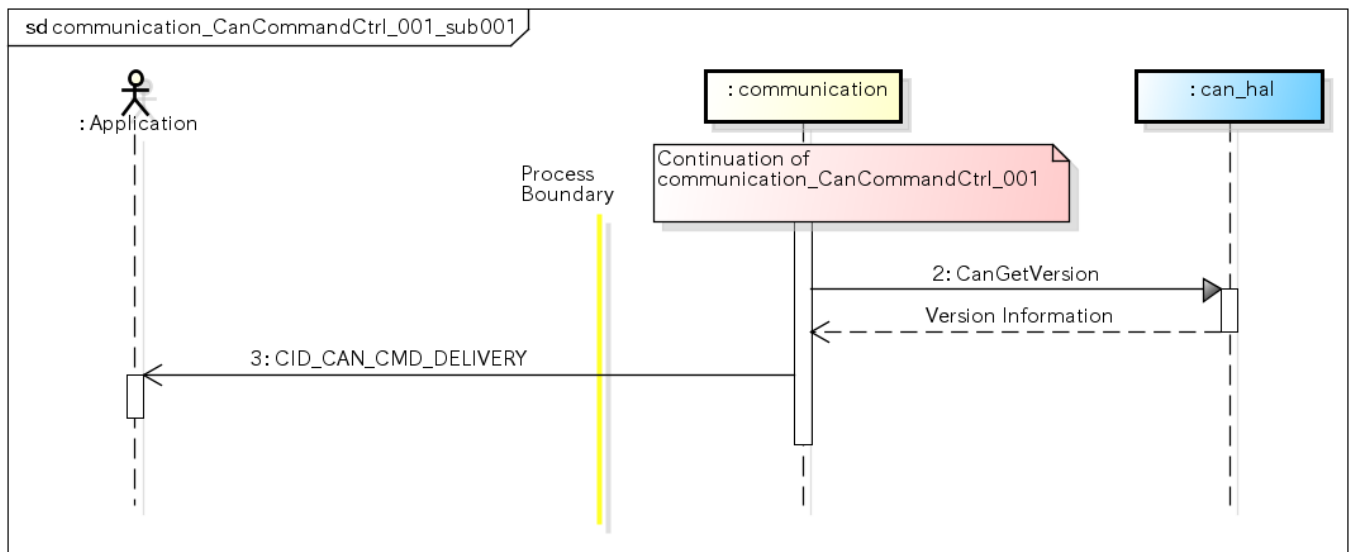
CANコマンド制御処理

Start CAN command control

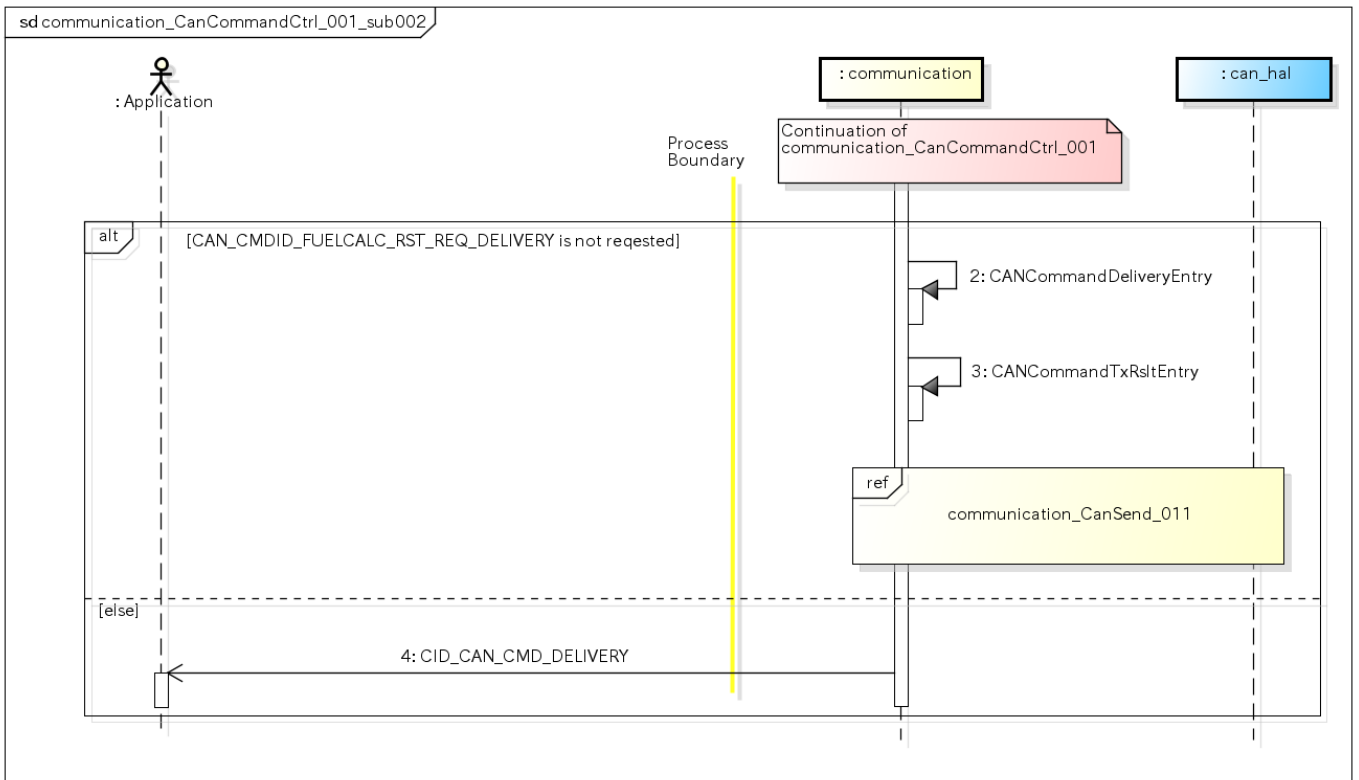
シーケンス [Sequence]



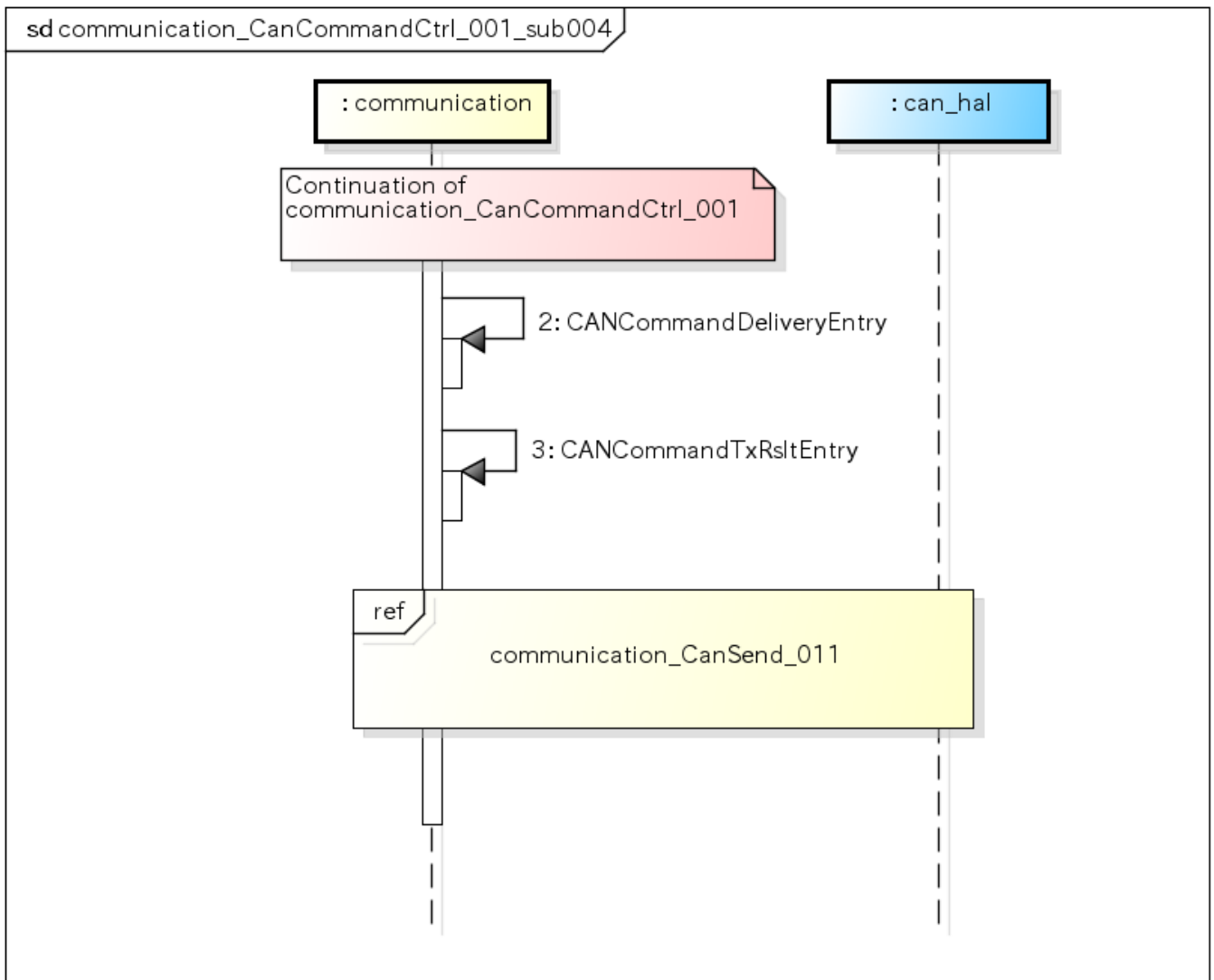
シーケンス [Sequence]



シーケンス [Sequence]



シーケンス [Sequence]



ユースケース communication_CanCommWatch_001 *[use-case communication_CanCommWatch_001]*

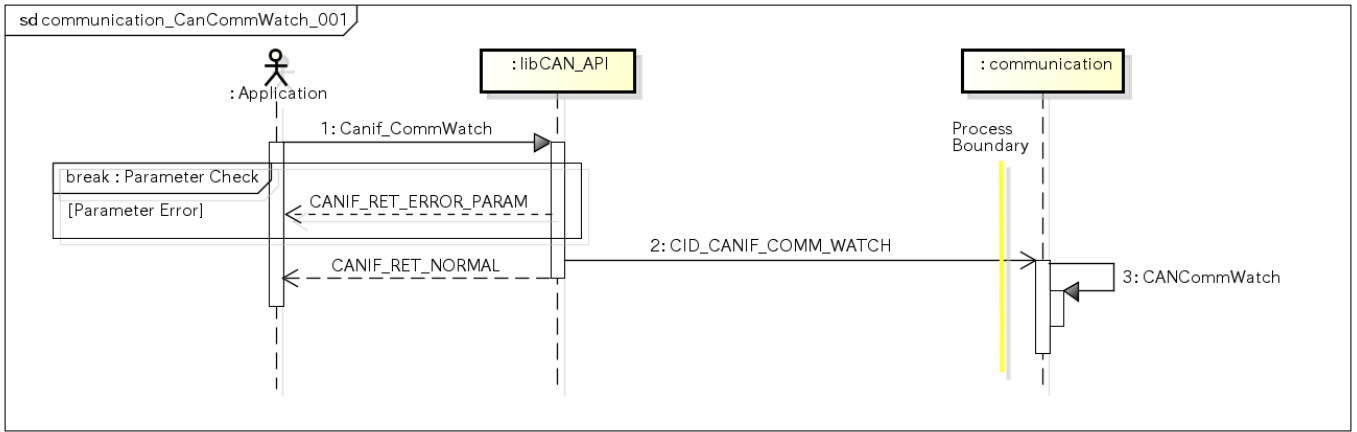
概要 *[Overview]*

CANデータ通信途絶監視処理

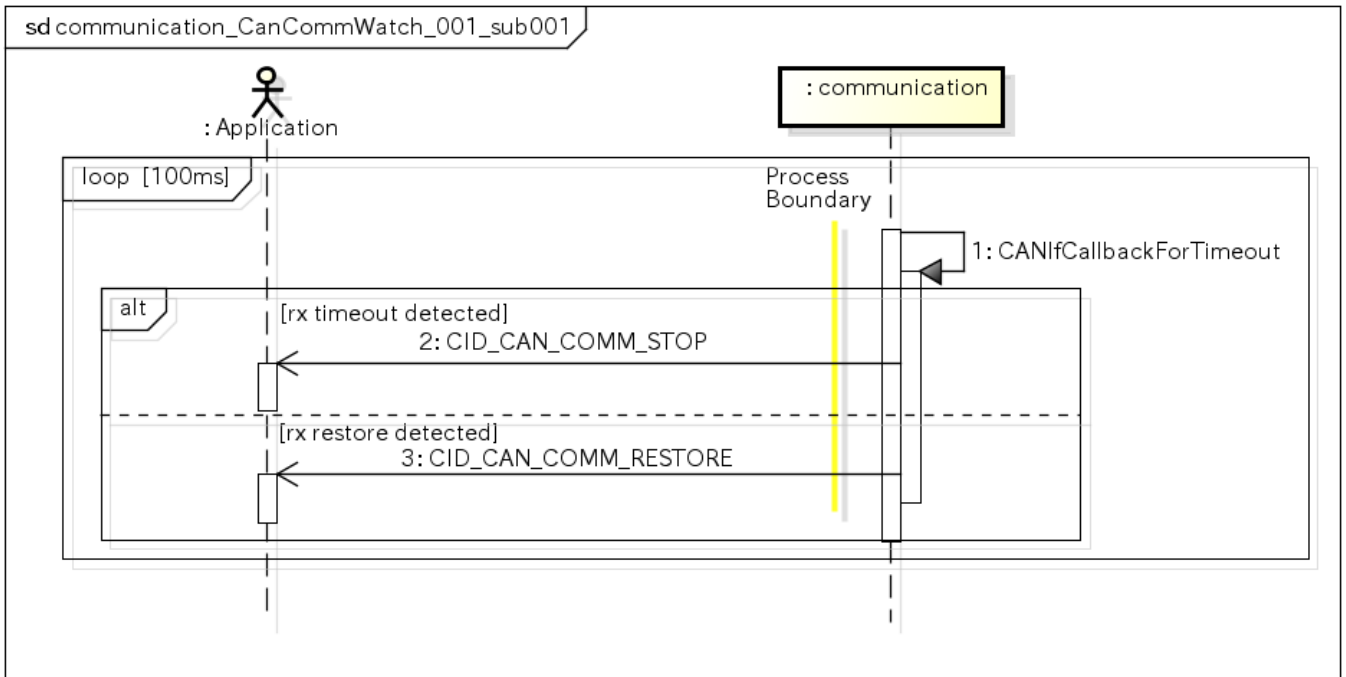
Start to watch CAN data

- 監視時間内でCANデータが受信していない場合は途絶として、途絶のイベントを外部に通知する
If there is no candata in the period of watch time,the status will be suspend and notify it to the user
- 途絶状態でCANデータが受信したら復帰する。復帰イベントを外部に通知する
If there is any can data receivee when suspend status,it will return to resume and notify it to the user

シーケンス *[Sequence]*



シーケンス [Sequence]



x