

次世代車載システム向け COM 外部仕様書

Ver.1.1.0

2016/03/31

Copyright (C) 2014-2016 by Center for Embedded Computing Systems

Graduate School of Information Science, Nagoya Univ., JAPAN

Copyright (C) 2014-2016 by AISIN COMCRUISE Co., Ltd., JAPAN

Copyright (C) 2014-2016 by FUJI SOFT INCORPORATED, JAPAN

Copyright (C) 2014-2016 by NEC Communication Systems, Ltd., JAPAN

Copyright (C) 2014-2016 by Panasonic Advanced Technology Development Co., Ltd., JAPAN

Copyright (C) 2014-2016 by SCSK Corporation, JAPAN

Copyright (C) 2014-2016 by Sunny Giken Inc., JAPAN

Copyright (C) 2015-2016 by SUZUKI MOTOR CORPORATION

Copyright (C) 2014-2015 by TOSHIBA CORPORATION, JAPAN

Copyright (C) 2014-2015 by Witz Corporation

上記著作権者は、以下の (1)~(3)の条件を満たす場合に限り、本ドキュメント（本ドキュメントを改変したものを含む。以下同じ）を使用・複製・改変・再配布（以下、利用と呼ぶ）することを無償で許諾する。

- (1) 本ドキュメントを利用する場合には、上記の著作権表示、この利用条件および下記の無保証規定が、そのままの形でドキュメント中に含まれていること。
- (2) 本ドキュメントを改変する場合には、ドキュメントを改変した旨の記述を、改変後のドキュメント中に含めること。ただし、改変後のドキュメントが、TOPPERS プロジェクト指定の開発成果物である場合には、この限りではない。
- (3) 本ドキュメントの利用により直接的または間接的に生じるいかなる損害からも、上記著作権者および TOPPERS プロジェクトを免責すること。また、本ドキュメントのユーザまたはエンドユーザからのいかなる理由に基づく請求からも、上記著作権者および TOPPERS プロジェクトを免責すること。

本ドキュメントは、AUTOSAR (AUTomotive Open System ARchitecture) 仕様に基いている。上記の許諾は、AUTOSAR の知的財産権を許諾するものではない。AUTOSAR は、AUTOSAR 仕様に基いたソフトウェアを商用目的で利用する者に対して、AUTOSAR パートナーになることを求めている。

本ドキュメントは、無保証で提供されているものである。上記著作権者および TOPPERS プロジェクトは、本ドキュメントに関して、特定の使用目的に対する適合性も含めて、いかなる保証も行わない。また、本ドキュメントの利用により直接的または間接的に生じたいかなる損害に関しても、その責任を負わない。

<目次>

1 概要	1
1.1 本文書の目的	1
1.2 関連文書	1
1.2.1 ベースとした文書	1
1.2.2 参考文書	2
1.3 凡例	3
1.3.1 仕様番号	3
1.3.2 API 仕様	4
1.3.3 注記	5
2 概念	6
2.1 機能一覧	6
2.2 主要概念	8
2.2.1 OSEK COM	8
2.2.2 初期化处理	10
2.2.2.1 IPDU グループの初期化	10
2.2.2.2 IPDU の初期化	10
2.2.2.3 シグナルの初期化	10
2.2.2.4 フィルタの初期化	10
2.2.3 未初期化处理	11
2.2.4 IPDU グループ管理	12
2.2.4.1 IPDU グループの定義	12
2.2.4.2 IPDU グループの起動	12
2.2.4.3 IPDU グループの停止	14
2.2.5 デッドラインモニタリング(DM)	16
2.2.5.1 送信デッドラインモニタリング	16
2.2.5.2 受信デッドラインモニタリング	18
2.2.5.2.1 受信 DM の有効化および無効化	20
2.2.6 最小遅延タイム(MDT)	21
2.2.7 受信完了通知タイミング	22
2.2.8 エンディアン変換	23
2.2.9 符号拡張	24
2.2.10 フィルタリング	25
2.2.10.1 フィルタアルゴリズム	27
2.2.11 送信モード	29
2.2.11.1 送信プロパティと送信モード	29
2.2.11.1.1 シグナル	29

2.2.11.1.2 シグナルグループ	3 0
2.2.11.2 TMS 判定	3 2
2.2.11.3 直接 N 回送信	3 4
2.2.11.4 周期送信	3 5
2.2.11.5 送信モードのユースケース	3 6
2.2.11.5.1 周期送信(送信モード変更なし)	3 7
2.2.11.6 VFB 仕様との関係	3 9
2.2.12 失敗送信要求リトライ	4 0
2.2.13 シグナル無効化	4 1
2.2.13.1 無効化されたシグナルの送信	4 1
2.2.13.2 無効化されたシグナルの受信	4 1
2.3 シグナルグループ(複合データ型の IPDU へのマッピング)	4 3
2.3.1 シグナルグループの初期化	4 3
2.3.2 シグナルグループの送信	4 3
2.3.3 シグナルグループの受信	4 3
2.3.4 シグナルグループ関連の通知	4 3
2.3.5 シグナルグループの属性	4 3
2.4 大型データ	4 5
2.4.1 大型データの送信	4 5
2.4.2 大型データの受信	4 5
2.5 可変長データ	4 7
2.5.1 可変長データの送信	4 7
2.5.2 可変長データの受信	4 7
2.6 アップデートビット	4 8
2.6.1 送信側のアップデートビット処理	4 8
2.6.2 受信側のアップデートビット処理	4 9
2.7 データシーケンス管理	5 0
2.7.1 送信側のデータシーケンス管理	5 0
2.7.2 受信側のデータシーケンス管理	5 0
2.8 通信保護	5 2
2.8.1 送信側の通信保護	5 2
2.8.2 受信側の通信保護	5 2
2.8.3 制約事項	5 3
2.9 シグナルゲートウェイ	5 4
2.9.1 シグナルに対するルーティング	5 4
2.9.2 シグナルグループに対するルーティング	5 4
2.9.3 タイムアウト発生時のルーティング	5 5

2.9.4 アップデートビット処理	5 5
2.9.5 シグナルゲートウェイの分離	5 6
2.10 エラー処理	5 7
2.10.1 エラー検出	5 7
2.10.2 エラー通知	5 7
2.11 関連モジュール	5 8
2.11.1 RTE	5 8
2.11.2 PduR	5 8
2.11.2.1 Com と PduR 間のインタフェース	5 8
2.12 COM 通信モデル	6 0
2.12.1 COM 送信モデル	6 2
2.12.2 COM 受信モデル	6 3
2.13 ファイル構成	6 4
2.14 コンフィギュレーション	6 5
2.14.1 コンフィギュレーションクラス	6 5
2.14.2 コンフィギュレーションルール	6 6
2.14.2.1 全般	6 6
2.14.2.2 シグナル	6 7
2.14.2.3 シグナルグループ	6 8
2.14.2.4 IPDU	6 8
2.14.2.5 シグナルゲートウェイ	6 8
2.14.2.6 フィルタ	6 8
2.14.2.7 周期関数呼び出し回数	6 9
2.15 本バージョン対象機能	7 0
3 API 仕様	7 2
3.1 API 実装方針	7 2
3.1.1 実装言語	7 2
3.1.2 再入可能性	7 2
3.1.3 引数チェック	7 2
3.1.4 デバッグ	7 2
3.2 データ型	7 3
3.2.1 外部からインポートするデータ型	7 3
3.2.2 内部で定義するデータ型	7 3
3.2.2.1 Com_StatusType	7 3
3.2.2.2 Com_SignalIdType	7 3
3.2.2.3 Com_SignalGroupIdType	7 3
3.2.2.4 Com_IpduGroupIdType	7 4

3.2.2.5 Com_IpduGroupVector	7 4
3.2.2.6 Com_ServiceIdType	7 4
3.2.2.7 Com_ConfigType	7 5
3.3 定数	7 6
3.3.1 IPDU 通信方向設定	7 6
3.3.2 IPDU 種別設定	7 6
3.3.3 シグナル型設定	7 6
3.3.4 エンディアン設定	7 7
3.3.5 フィルタアルゴリズム設定	7 8
3.3.6 送信モード設定	7 9
3.3.7 送信プロパティ設定	7 9
3.3.8 通信完了通知タイミング設定	8 0
3.3.9 アップデートビットクリアタイミング設定	8 0
3.3.10 エラー発生時処理設定	8 0
3.3.11 システムサービス返り値	8 1
3.3.12 無効値	8 1
3.4 返り値	8 2
3.5 エラーコード	8 2
3.5.1 標準エラー	8 2
3.5.2 拡張エラー	8 2
3.6 システムサービス	8 3
3.6.1 Com_Init	8 3
3.6.2 Com_DeInit	8 4
3.6.3 Com_IpduGroupControl	8 5
3.6.4 Com_ReceptionDMControl	8 6
3.6.5 Com_GetStatus	8 7
3.6.6 Com_GetConfigurationId	8 7
3.6.7 Com_GetVersionInfo	8 8
3.6.8 Com_ClearIpduGroupVector	8 9
3.6.9 Com_SetIpduGroup	8 9
3.6.10 Com_SendSignal	9 1
3.6.11 Com_SendDynSignal	9 2
3.6.12 Com_ReceiveSignal	9 4
3.6.13 Com_ReceiveDynSignal	9 5
3.6.14 Com_SendSignalGroup	9 6
3.6.15 Com_ReceiveSignalGroup	9 7
3.6.16 Com_InvalidateSignal	9 8

3.6.17 Com_InvalidateSignalGroup.....	9 9
3.6.18 Com_TriggerIPDUSend	1 0 0
3.6.19 Com_SwitchIpduTxMode	1 0 1
3.7 コールバック関数	1 0 2
3.7.1 Com_TriggerTransmit	1 0 2
3.7.2 Com_RxIndication	1 0 3
3.7.3 Com_TpRxIndication	1 0 4
3.7.4 Com_TxConfirmation	1 0 5
3.7.5 Com_TpTxConfirmation	1 0 6
3.7.6 Com_StartOfReception	1 0 7
3.7.7 Com_CopyRxData	1 0 9
3.7.8 Com_CopyTxData	1 1 0
3.8 周期関数	1 1 1
3.8.1 Com_MainFunctionRx	1 1 1
3.8.2 Com_MainFunctionTx	1 1 1
3.8.3 Com_MainFunctionRouteSignals	1 1 2
3.9 依存インタフェース	1 1 3
3.9.1 必須インタフェース	1 1 3
3.9.2 任意インタフェース	1 1 3
3.9.3 コンフィギュレーション可能なインタフェース	1 1 4
3.9.3.1 コールバック関数.....	1 1 4
3.9.3.1.1 Com_CbkTxAck	1 1 4
3.9.3.1.2 Com_CbkTxErr	1 1 4
3.9.3.1.3 Com_CbkTxTOut.....	1 1 5
3.9.3.1.4 Com_CbkRxAck.....	1 1 5
3.9.3.1.5 Com_CbkRxTOut	1 1 6
3.9.3.1.6 Com_CbkInv.....	1 1 6
3.9.3.1.7 Com_CbkCounterErr	1 1 7
3.9.3.2 コールアウト関数.....	1 1 8
3.9.3.2.1 Com_RxIpduCallout.....	1 1 8
3.9.3.2.2 Com_TxIpduCallout	1 1 9
3.10 コンテナおよびパラメータ	1 2 1
3.10.1 Com.....	1 2 1
3.10.2 ComGeneral.....	1 2 1
3.10.2.1 ComConfigurationUseDet	1 2 1
3.10.2.2 ComEnableMDTForCyclicTransmission	1 2 2
3.10.2.3 ComRetryFailedTransmitRequests	1 2 2

3.10.2.4 ComSupportedIPduGroups.....	1 2 2
3.10.2.5 ComVersionInfoApi.....	1 2 3
3.10.3 ComConfig.....	1 2 4
3.10.3.1 ComConfigurationId	1 2 4
3.10.4 ComTimeBase	1 2 4
3.10.4.1 ComTxTimeBase.....	1 2 5
3.10.4.2 ComRxTimeBase.....	1 2 5
3.10.4.3 ComGwTimeBase.....	1 2 6
3.10.5 ComIPduGroup	1 2 7
3.10.5.1 ComIPduGroupHandleId.....	1 2 7
3.10.5.2 ComIPduGroupGroupRef	1 2 7
3.10.6 ComIPdu.....	1 2 9
3.10.6.1 ComIPduHandleId.....	1 2 9
3.10.6.2 ComIPduType	1 3 0
3.10.6.3 ComIPduDirection	1 3 0
3.10.6.4 ComIPduSignalProcessing	1 3 1
3.10.6.5 ComIPduCancellationSupport.....	1 3 1
3.10.6.6 ComIPduCallout	1 3 1
3.10.6.7 ComIPduTriggerTransmitCallout	1 3 2
3.10.6.8 ComIPduGroupRef.....	1 3 2
3.10.6.9 ComIPduSignalRef	1 3 2
3.10.6.10 ComIPduSignalGroupRef	1 3 3
3.10.6.11 ComPduIdRef.....	1 3 3
3.10.7 ComTxIPdu	1 3 3
3.10.7.1 ComTxIPduUnusedAreasDefault.....	1 3 4
3.10.7.2 ComTxIPduClearUpdateBit	1 3 4
3.10.7.3 ComMinimumDelayTime	1 3 5
3.10.8 ComTxModeTrue.....	1 3 6
3.10.9 ComTxModeFalse.....	1 3 6
3.10.10 ComTxMode.....	1 3 7
3.10.10.1 ComTxModeMode	1 3 7
3.10.10.2 ComTxModeNumberOfRepetitions	1 3 7
3.10.10.3 ComTxModeRepetitionPeriod.....	1 3 8
3.10.10.4 ComTxModeTimeOffset	1 3 8
3.10.10.5 ComTxModeTimePeriod	1 3 9
3.10.11 ComIPduCounter.....	1 4 0
3.10.11.1 ComIPduCounterStartPosition.....	1 4 0

3.10.11.2 ComIPduCounterSize	1 4 0
3.10.11.3 ComIPduCounterThreshold.....	1 4 1
3.10.11.4 ComIPduCounterErrorNotification	1 4 1
3.10.12 ComIPduReplication	1 4 2
3.10.12.1 ComIPduReplicationQuorum	1 4 2
3.10.12.2 ComIPduReplicaRef.....	1 4 2
3.10.13 ComSignal	1 4 3
3.10.13.1 ComHandleId.....	1 4 3
3.10.13.2 ComSignalType	1 4 4
3.10.13.3 ComBitPosition	1 4 5
3.10.13.4 ComBitSize	1 4 6
3.10.13.5 ComSignalLength	1 4 6
3.10.13.6 ComUpdateBitPosition	1 4 7
3.10.13.7 ComSignalInitValue.....	1 4 7
3.10.13.8 ComSignalDataInvalidValue	1 4 8
3.10.13.9 ComSignalEndianness.....	1 4 8
3.10.13.10 ComTransferProperty	1 4 9
3.10.13.11 ComFirstTimeout.....	1 5 0
3.10.13.12 ComTimeout.....	1 5 0
3.10.13.13 ComDataInvalidAction	1 5 0
3.10.13.14 ComRxDataTimeoutAction	1 5 1
3.10.13.15 ComNotification	1 5 1
3.10.13.16 ComErrorNotification	1 5 1
3.10.13.17 ComInvalidNotification	1 5 2
3.10.13.18 ComTimeoutNotification.....	1 5 2
3.10.13.19 ComSystemTemplateSystemSignalRef	1 5 2
3.10.14 ComSignalGroup	1 5 3
3.10.14.1 ComHandleId.....	1 5 3
3.10.14.2 ComUpdateBitPosition	1 5 4
3.10.14.3 ComTransferProperty	1 5 4
3.10.14.4 ComFirstTimeout.....	1 5 5
3.10.14.5 ComTimeout.....	1 5 5
3.10.14.6 ComDataInvalidAction	1 5 5
3.10.14.7 ComRxDataTimeoutAction.....	1 5 6
3.10.14.8 ComNotification	1 5 6
3.10.14.9 ComErrorNotification	1 5 6
3.10.14.10 ComInvalidNotification	1 5 6

3.10.14.11 ComTimeoutNotification.....	1 5 7
3.10.14.12 ComSystemTemplateSignalGroupRef.....	1 5 7
3.10.15 ComGroupSignal.....	1 5 8
3.10.15.1 ComHandleId.....	1 5 8
3.10.15.2 ComSignalType.....	1 5 8
3.10.15.3 ComBitPosition.....	1 5 9
3.10.15.4 ComBitSize.....	1 5 9
3.10.15.5 ComSignalLength.....	1 6 0
3.10.15.6 ComSignalInitValue.....	1 6 0
3.10.15.7 ComSignalDataInvalidValue.....	1 6 1
3.10.15.8 ComSignalEndianness.....	1 6 1
3.10.15.9 ComTransferProperty.....	1 6 1
3.10.15.10 ComSystemTemplateSystemSignalRef.....	1 6 2
3.10.16 ComFilter.....	1 6 3
3.10.16.1 ComFilterAlgorithm.....	1 6 3
3.10.16.2 ComFilterMask.....	1 6 4
3.10.16.3 ComFilterX.....	1 6 5
3.10.16.4 ComFilterMin.....	1 6 5
3.10.16.5 ComFilterMax.....	1 6 6
3.10.16.6 ComFilterOffset.....	1 6 7
3.10.16.7 ComFilterPeriod.....	1 6 7
3.10.17 ComGwMapping.....	1 6 8
3.10.18 ComGwSource.....	1 6 8
3.10.19 ComGwDestination.....	1 6 8
3.10.20 ComGwSourceDescription.....	1 6 9
3.10.20.1 ComSignalType.....	1 6 9
3.10.20.2 ComBitPosition.....	1 7 0
3.10.20.3 ComBitSize.....	1 7 0
3.10.20.4 ComSignalLength.....	1 7 0
3.10.20.5 ComUpdateBitPosition.....	1 7 1
3.10.20.6 ComSignalEndianness.....	1 7 1
3.10.20.7 ComGwIPduRef.....	1 7 1
3.10.21 ComGwDestinationDescription.....	1 7 2
3.10.21.1 ComBitPosition.....	1 7 2
3.10.21.2 ComUpdateBitPosition.....	1 7 3
3.10.21.3 ComSignalInitValue.....	1 7 3
3.10.21.4 ComSignalEndianness.....	1 7 3

3.10.21.5 ComTransferProperty	1 7 4
3.10.21.6 ComGwIPduRef	1 7 4
3.10.22 ComGwSignal.....	1 7 6
3.10.22.1 ComGwSignalRef.....	1 7 6
4 リファレンス	1 7 7
4.1 データ型一覧	1 7 7
4.2 定数一覧	1 7 7
4.3 システムサービス一覧	1 7 9
4.4 コールバック関数一覧	1 8 0
4.5 周期関数一覧	1 8 0
4.6 エラーコード一覧	1 8 0
4.7 仕様番号一覧	1 8 1
4.7.1 削除した AUTOSAR 仕様一覧	1 8 1
4.7.2 変更した AUTOSAR 仕様一覧	1 8 1
4.7.3 追加した NCES 仕様一覧	1 8 2
4.7.4 本バージョンで対象外となる AUTOSAR 仕様一覧	1 8 5

図 2-1 受信 DM とアップデートビットの関係	1 8
図 2-2 ONE EVERY_N のフィルタリング例	2 8
図 2-3 他モジュールとの関係	5 8
図 2-4 時間的整合性が乱れる例	6 1
図 2-5 COM 送信モデル図	6 2
図 2-6 COM 受信モデル図	6 3
図 2-7 インクルードファイル構成	6 4

表 1-1 ベースとした文書一覧	1
表 1-2 参考文書一覧	2
表 1-3 仕様番号凡例	3
表 2-1 除外された OSEK COM の機能とその根拠	8
表 2-2 有効化された IPDU に対する振舞い	14
表 2-3 無効化された IPDU に対する振舞い	15
表 2-4 ComSignalType と ComFilterAlgorithms の対応表	26
表 2-5 送信モードおよび送信プロパティと IPDU 送信タイミングの関係	29
表 2-6 TMS 判定	32
表 2-7 ユースケース凡例	36
表 2-8 送信モードおよび送信プロパティと VFB 送信モードの関係	39
表 2-9 シグナルグループの属性	44
表 2-10 アップデートビットの状態とその意味	48
表 2-11 アップデートビットとルーティングの関係	56
表 2-12 IPDU 種別(ComIPduType) ごとの対応 API	59
表 2-13 本バージョン対象機能	70
表 2-14 本バージョン対象システムサービス	71
表 3-1 インポートするデータ型	73
表 3-2 COM システムサービスの返り値一覧	82
表 3-3 拡張エラーコード一覧	82
表 3-4 必須インタフェース	113
表 3-5 任意インタフェース	113
表 3-6 シグナルの IPDU 配置位置チェック概要	145
表 4-1 データ型一覧	177
表 4-2 定数一覧	177
表 4-3 システムサービス一覧	179
表 4-4 コールバック関数一覧	180
表 4-5 周期関数一覧	180
表 4-6 エラーコード一覧	180

1 概要

1.1 本文書の目的

本文書は「Requirements on Communication」に記述された要求事項を満たす COM の機能仕様を規定するものである。

本文書に含まれる仕様(以降、本仕様と略す)は、AUTOSAR Specification of Communication 仕様(以降、AUTOSAR 仕様と略す)をベースに、必要な拡張と修正を行ったものである。また、AUTOSAR 仕様は OSEK/VDX Communication 仕様(以降、OSEK 仕様と略す)をベースにしているため、本仕様も間接的に OSEK 仕様をベースにしている。

本文書に含まれる用語の定義は、「次世代車載システム向け COM 用語集」に記述している。

本文書は、COM に関する一般的な知識を持ったソフトウェア技術者が読むことを想定している。

OSEK 仕様や AUTOSAR 仕様に関する知識を持っていることが望ましいが、それを前提とせずに記述している。

1.2 関連文書

1.2.1 ベースとした文書

以下の表 1-1 は、本文書のベースとした文書であり、その内容は本文書内に含まれている。

ただし、AUTOSAR 仕様は OSEK 仕様の一部のみ含む〔COM013〕ため、AUTOSAR 仕様に含まない OSEK 仕様に関する内容は本文書に含まれていない。

表 1-1 ベースとした文書一覧

文書名	バージョン
OSEK/VDX Communication	Ver.3.0.3
AUTOSAR Specification of Communication	V4.2.0 (R4.0 Rev3)

1.2.2 参考文献

以下の表 1-2 参考文献一覧表 1-2 は、本文書から参照している文書、または本文書を理解するために必要な文書である。その内容は本文書に含まれていない。

表 1-2 参考文献一覧

文書名	バージョン
次世代車載システム向け COM 用語集	Ver.1.0.2
次世代車載システム向け RTOS 外部仕様書	Ver3.2.0
次世代車載システム向け RTE 外部仕様書	Ver1.0.0
AUTOSAR Layered Architecture	V3.2.0 (R4.0 Rev3)
Virtual Functional Bus	V2.2.0 (R4.0 Rev3)
General Requirements on Basic Software Modules	V3.2.0 (R4.0 Rev3)
List of Basic Software Modules	V1.6.0 (R4.0 Rev3)
Software Component Template	V4.2.0 (R4.0 Rev3)
Generic Structure Template	V3.2.0 (R4.0 Rev3)
Requirements on System Template	V3.1.0 (R4.0 Rev3)
Requirements on Debugging	V1.0.0 (R4.0 Rev1)
Requirements on Communication	V3.1.0 (R4.0 Rev3)
Requirements on Gateway	V2.2.0 (R4.0 Rev3)
Specification of BSW Module Description Template	V2.2.0 (R4.0 Rev3)
Specification of ECU Configuration	V3.2.0 (R4.0 Rev3)
Specification of Standard Types	V1.3.0 (R4.0 Rev1)
Specification of Communication Stack Types	V3.2.0 (R4.0 Rev3)
Specification of Communication Manager	V4.0.0 (R4.0 Rev3)
Specification of CAN Transport Layer	V4.0.0 (R4.0 Rev3)
Specification of FlexRay Transport Layer	V3.0.0 (R4.0 Rev3)
Specification of PDU Router	V3.2.0 (R4.0 Rev3)

1.3 凡例

1.3.1 仕様番号

本文書では、OSEK 仕様と、AUTOSAR 仕様と、名古屋大学大学院情報科学研究科附属組込みシステム研究センター(NCES)を中心とする次世代車載システム向け RTOS の仕様検討及び開発に関するコンソーシアム型共同研究(ATK2 コンソーシアム)で新規に規定した仕様とが混在しているため、以下に示す仕様番号を用いてこれらの仕様を区別して管理を行う。仕様番号は、要求事項にのみ付与することを基本とする。ただし、AUTOSAR 仕様において、概念の説明や補足事項に対しても仕様番号が付与されているものに関しては、そのまま付与する。また、本文書から ATK2 コンソーシアムで開発した A-COM の実装に依存する仕様は、参考情報として本文書に記載するため、仕様番号を付与する。

表 1-3 仕様番号凡例

仕様番号	内容
【CCOM _{xxx} 】	関連文書「OSEK/VDX Communication」で規定された仕様。
【COM _{xxx} 】	関連文書「AUTOSAR Specification of Communication」で規定された仕様。 AUTOSAR 仕様で記述されている COM 仕様番号を用いる。本仕様に採用しない AUTOSAR 仕様についても【COM _{xxx} 】で記載している。
【COM _{xxx} _Conf】	関連文書「AUTOSAR Specification of Communication」で規定された仕様。 AUTOSAR 仕様で記述されているコンフィギュレーション仕様番号を用いる。本仕様に採用しない AUTOSAR 仕様についても【COM _{xxx} _Conf】で記載している。
【COM _a xxx】	関連文書「AUTOSAR Specification of Communication」で規定された機能仕様のうち、仕様番号が付与されていないもの。AUTOSAR 仕様の要求事項であるため、本仕様において仕様番号を付与する。
【NCOM _{xxx} 】	ATK2 コンソーシアムで新規に規定した仕様。
[CCOM _{xxx}] [COM _{xxx}] [NCOM _{xxx}]	本文書で上記の仕様番号を内部参照する際に使用する。 仕様定義である【CCOM _{xxx} 】【COM _{xxx} 】【NCOM _{xxx} 】と区別するために左記のように記述する。
<COM _{xxx} >	本仕様で削除した AUTOSAR 仕様の仕様番号を記載する際に使用する。

1.3.2 API 仕様

データ型名

データ型名	＜データ型の ID 名を記載する＞
型	＜データ型の種類を記載する＞
範囲	＜そのデータ型の値の範囲を記載する＞
概要	＜データ型が示す内容を記載する＞

定数名

定数名	＜定数の ID 名を記載する＞
概要	＜定数が示す内容を記載する＞

システムサービス名

サービス ID		＜システムサービスの ID を記載する＞
C 言語 I/F		＜システムサービスの C 言語 I/F を記載する＞
パラメータ	[in]	＜システムサービスのパラメータで入力となるものを記載する＞
	[in/out]	＜システムサービスのパラメータで入力と出力となるものを記載する＞
	[out]	＜システムサービスのパラメータで出力となるものを記載する＞
返り値		＜システムサービスの返り値で標準エラーとなるものと，該当エラーが発生する条件を記載する．(W)が付いている場合は警告として扱っても良い＞
DET エラー		＜システムサービス内で通知する DET エラーのエラーコードと発生条件を記載する＞
同期/非同期		＜システムサービスの同期性を記載する＞
再入可能/再入不可能		＜システムサービスの再入可能性を記載する＞
機能		＜システムサービスの機能を記載する＞

メイン関数名

サービス ID	＜メイン関数の ID を記載する＞
C 言語 I/F	＜メイン関数の C 言語 I/F を記載する＞
起動タイミング	＜メイン関数の起動タイミングを記載する＞
DET エラー	＜システムサービス内で通知する DET エラーのエラーコードと発生条件を記載する＞
機能	＜システムサービスの機能を記載する＞

コンテナ名<コンテナ名称の末尾のみを記載する>

コンテナ名	<コンテナ名称をフルパスで記載する>
概要	<コンテナの概要を記載する>
多重度	<コンテナの多重度を記載する>
パラメータ	<コンテナが含むパラメータ名を列挙する>
サブコンテナ	<コンテナが含むサブコンテナ名を列挙する>

パラメータ<パラメータ名称の末尾のみを記載する>

パラメータ名	<パラメータ名称をフルパスで記載する>
概要	<パラメータの概要を記載する>
多重度	<パラメータの多重度を記載する>
型	<パラメータの型を記載する>
値の範囲	<パラメータの値の範囲を記載する>
デフォルト値	<パラメータのデフォルト値を記載する>
コンフィギュレーションクラス	
<プレコンパイルタイム, リンクタイム, ポストビルドタイムのいずれかを記載する>	
制限事項	
<パラメータの制限事項を記載する>	

1.3.3 注記

AUTOSAR 仕様との違い

削除や改変を行った AUTOSAR 仕様に対して、どのような差分があるかを説明する。

OSEK 仕様との違い

削除や改変を行った OSEK 仕様に対して、どのような差分があるかを説明する。

設計上の注意

本仕様に準拠した COM(以降, 本 COM と略す)を開発するユーザに対する注意事項または推奨事項を説明する。本 COM の使用者に向けた項目ではない。

使用上の注意

本 COM を使用してアプリケーションを開発するユーザに対する注意事項または推奨事項を説明する。本 COM の開発者に向けた項目ではない。

2 概念

2.1 機能一覧

本 COM が提供する機能の概略を以下に示す.

シグナル送受信

RTE に向けた, シグナル単位 of データ送受信を行う機能.

シグナルグループ送受信

RTE に向けた, シグナルグループ単位 of データ送受信を行う機能.

パッキング

送信シグナルを送信 IPDU へ梱包する機能.

アンパッキング

受信 IPDU から受信シグナルを解梱する機能.

通信制御

IPDU グループごとに, 通信の開始と停止を制御する機能.

最小遅延タイム(MDT)

IPDU の送信間隔における, 最小時間を確保する機能.

デッドラインモニタリング(DM)

シグナル送受信におけるタイムアウトを監視する機能.

送信側のデッドラインモニタリングを送信 DM と呼び, 受信側のデッドラインモニタリングを受信 DM と呼ぶ.

フィルタリング

フィルタアルゴリズムに従ってシグナルの有効無効を判定する機能.

エンディアン変換

指定されたエンディアンに従ってパッキングまたはアンパッキングを行う機能.

指定可能なエンディアンは以下の 3 種類.

- ・ リトルエンディアン
- ・ ビックエンディアン
- ・ オペーク

符号拡張

アンパッキングしたシグナルのデータを指定されたシグナル型に合わせる機能.

通知

以下を RTE へ通知する機能.

- ・ 要求された送信が完了したことを通知するための, 送信完了通知
- ・ 受信が発生したことを通知するための, 受信通知
- ・ DM で通信のタイムアウトが発生したことを通知するための, タイムアウト通知

TMS 判定

IPDU の送信モードを選択する機能.

IPDU の送信モードは 2 種類(ComTxModeTrue および ComTxModeFalse)持つことが可能であり, どちらのモードを選択するかを, シグナルの状態(TMC)によって判定する.

失敗送信要求リトライ

失敗した送信を再度 PDUR へ送信要求する機能.

シグナル無効化

シグナルの無効値(*ComSignalDataInvalidValue*)を送信することでシグナルを無効化する機能.

受信側では, 無効化されたシグナルを受信した場合の対応を行う.

シグナルゲートウェイ

受信 IPDU 内のシグナルまたはシグナルグループを RTE へ伝送せずに, COM 内部で他の IPDU に梱包して送信する機能.

可変長データ通信

可変長シグナルおよび可変長 IPDU の送受信を行う機能. TP を用いて実現する.

大型データ通信

大型シグナルおよび大型 IPDU の送受信を行う機能. TP を用いて実現する.

2.2 主要概念

2.2.1 OSEK COM

OSEK COM は、AUTOSAR COM の機能のベースである。

OSEK COM の機能のうち、COM013 において除外されたもの以外を AUTOSAR COM に実装する【COM010】[COM013]。OSEK 仕様と AUTOSAR 仕様との間で機能の定義が異なる場合は、AUTOSAR 仕様に従う【COM011】。OSEK 仕様には定義されていない機能が AUTOSAR 仕様に定義されている場合は、AUTOSAR 仕様に従って実装する【COM012】。OSEK COM の機能のうち、以下の表 2-1 に示す機能は AUTOSAR COM に実装しない【COM013】。

表 2-1 除外された OSEK COM の機能とその根拠

除外された OSEK COM の機能	除外された根拠
1:N 通信	RTE で実現するため
M:1 送信	RTE で実現するため
メッセージのキューイング	RTE で実現するため
通知メカニズム(TASK, FLAG, EVENT)	RTE で実現するため
ECU 内通信	RTE で実現するため
アプリケーションモード ・ GetComApplicationMode	不要な機能であるため
間接 NM へのインタフェース ・ I_MessageTransfer ・ I_MessageTimeOut	不要な機能であるため
ゼロサイズメッセージ ・ SendZeroMessage	不要な機能であるため (部分的に Com_TriggerTransmit でカバー)
送信側のシグナルフィルタリング	ユースケースが無い (ただし、送信モードの選択のためにフィルタリングのアルゴリズムを利用する)
IPDU 内のメッセージのオーバーラップ	ユースケースがなく、危険な概念であるため
OIL の使用	AUTOSAR 仕様ではコンフィギュレーション時に OIL を使用しないため
再入可能性	AUTOSAR 仕様の API のすべてが再入可能ではないため
周期メッセージの起動と停止 ・ StartPeriodic, ・ StopPeriodic	AUTOSAR 仕様では IPDU グループごとに起動と停止を行うため ・ Com_IPduGroupStart ・ Com_IpduGroupStop
ネットワークオーダーメッセージコールアウト	AUTOSAR 仕様では IPDU コールアウトのみ

除外された OSEK COM の機能	除外された根拠
CPU オーダーメッセージコールアウト	サポートするため
エラーフックルーチン ・ COMErrorHook ・ COMError_Name1_Name2 macros ・ COMErrorGetServiceId	AUTOSAR 仕様では DEM と DET の API を 直接利用してエラー通知するため
コールバックルーチンのためのインタフェース ・ COMCallback	AUTOSAR 仕様ではコールバック関数のシグ ネチャが明示的に定義されるため
スタートアップ処理のための API ・ StartCOM ・ StopCOM ・ StartCOMExtensions ・ InitMessage	以下の API に置き換えるため ・ Com_Init ・ Com_DeInit ・ Com_IpduGroupControl

2.2.2 初期化处理

本章では、Com_Init において行う処理について記述する。

COM システムサービスが呼び出される前に、C 言語のスタートアップ処理によってグローバル変数および静的変数が初期化されていることを前提条件とする。

COM は、C 言語のスタートアップ処理で初期化されない変数を、Com_Init において初期化する【COM128】。

Com_Init は、ECU 間通信の有効化を行わない【COM328】。

2.2.2.1 IPDU グループの初期化

IPDU グループの状態のデフォルトは停止状態であり、Com_Init によって開始状態に遷移することはない【COM444】。

2.2.2.2 IPDU の初期化

COM は、初期化時に送信側の IPDU バッファを送信 IPDU 未使用エリアデフォルト値 (ComTxIPduUnusedAreasDefault) の値で初期化する【COM015】。

COM は、初期化時にすべてのアップデートビットをクリアする【COM117】。

シグナルのサイズ(ComBitSize) に n が設定されたシグナルの初期値(ComSignalInitValue) の下位 n ビットを、IPDU バッファにコピーする【COM098】。

シグナルの型(ComSignalType) が UINT8_N のシグナルの初期値(ComSignalInitValue) を、下位バイトから順に、シグナルの長さ(ComSignalLength) の分だけ IPDU バッファにコピーする【COM483】。

COM は、以下の順で IPDU バッファを初期化する【COM217】。

- ・ 送信側の IPDU バッファを送信 IPDU 未使用エリアデフォルト値 (ComTxIPduUnusedAreasDefault) の値で初期化する【COM015】
- ・ シグナルの初期値(ComSignalInitValue) を IPDU バッファに配置する【COM098】【COM483】
- ・ アップデートビットをクリアする【COM117】

2.2.2.3 シグナルの初期化

COM は、シグナルバッファをシグナルの初期値(ComSignalInitValue) で初期化する【COM217】。

2.2.2.4 フィルタの初期化

COM は、初期化時に前回値(new_value)にシグナルの初期値(ComSignalInitValue) をコピーする【COM603】。

2.2.3 未初期化处理

本章では、Com_DeInit において行う処理について記述する。

Com_DeInit によって、すべての IPDU グループは停止され、ECU 間通信は無効化される [COM129]。
Com_DeInit の処理が終了した後、COM の状態は未初期化状態へ遷移する。

2.2.4 IPDU グループ管理

2.2.4.1 IPDU グループの定義

IPDU グループに対して、以下の規則が適用される。

1. IPDU は、任意の IPDU グループに所属することができる。
2. IPDU が所属している IPDU グループのいずれかが起動された場合、その IPDU は有効化される【COM771】。
3. サポートする IPDU グループの最大数はプリコンパイルタイムにおいて設定される。

規則の 1 と 3 は COM のコンフィギュレーション情報によってサポートされる。サポートする IPDU グループの最大数はサポート IPDU グループ数(*ComSupportedIpduGroups*) によって設定可能である【COM710_Conf】。

IPDU の通信を行うためには、その IPDU を有効化する必要がある。IPDU は所属 IPDU グループの起動によって有効化されるため、IPDU グループに所属していない IPDU の通信は行われない。

AUTOSAR 仕様との違い

AUTOSAR 仕様では IPDU グループのネスト所属がサポートされているが、本仕様ではサポートしない【NCOM009】。

2.2.4.2 IPDU グループの起動

IPDU グループのデフォルトの状態は停止状態である【COM444】。Com_IpduGroupControl の呼び出しによって、停止状態の IPDU グループが起動された場合、その IPDU グループの状態は起動状態に遷移する。

Com_IpduGroupControl の呼び出しによって IPDU が有効化された場合、COM はその IPDU に所属しているシグナルおよびシグナルグループに対する送信要求および受信要求を受け付ける【COM114】。

Com_IpduGroupControl の呼び出しによって IPDU が有効化された場合、COM はその IPDU に対する以下の初期化処理を常に行う【COM787】。

- ・ 送信モード(*ComTxModeMode*) が DIRECT または MIXED である場合、MDT リセット
- ・ 送信 DM および受信 DM のタイムアウト監視のためのタイマをリセット
- ・ IPDU 内の全アップデートビットをクリア
- ・ フィルタアルゴリズム(*ComFilterAlgorithm*) が ONE_EVERY_N のフィルタを持つシグナルの OCCURRENCE をリセット
- ・ 送信 IPDU のカウンタ値を 0 にする
- ・ 次に受け入れた受信 IPDU のカウンタ値を受け入れる

引数 Initialize に TRUE を指定した上で Com_IpduGroupControl を呼び出すことによって、IPDU が有効化された場合、COM はその IPDU に対する以下の初期化処理を常に行う【COM222】.

- ・ 【COM217】 で定義されている IPDU のデータを初期化
- ・ シグナルグループに含まれるシャドウバッファを初期化
- ・ フィルタリングのための前回値をシグナルの初期値(*ComSignalInitValue*) に置き換える
- ・ 送信モード(*ComTxModeMode*) が PERIODIC または MIXED である場合、送信周期のためのタイマをリセット

【COM787】 および 【COM222】 では、Com_IpduGroupControl によって初期化されるパラメータに関して説明されているが、送信モード(*ComTxModeMode*) が PERIODIC の IPDU が初期化なしで有効化された場合に、その IPDU の周期送信が再度オフセット時間から始まることは読み取れない。

したがって、送信モード(*ComTxModeMode*) が PERIODIC の IPDU が Com_IpduGroupControl で無効化された場合、その時点で周期カウンタを一旦停止し、その後、Com_IpduGroupControl が初期化なしで呼び出され、かつ IPDU が有効化された場合、周期カウンタを初期化せずにそのまま周期送信を再開する【COMa001】.

※ 周期送信の offset は送信開始タイミングを IPDU ごとに分けて、負荷分散するユースケースが考えられるため、本仕様が望ましい

※ MIXED における周期も同様である

Com_IpduGroupControl の呼び出しによって送信 IPDU が有効化された場合、COM はその IPDU に含まれるシグナルの状態(TMC)に従って TMS 判定し、送信モードを決定する【COM223】.

ただし、フィルタアルゴリズムが ONE_EVERY_N に設定されているシグナルの状態(TMC)は FALSE とする【NCOM053】.

TMS 判定による送信モード決定に関しては、「2.2.11.2 TMS 判定」を参照.

Com_IpduGroupControl の呼び出しによって送信 IPDU が有効化された場合、その時点でまだ更新されていないシグナルに対しては、そのシグナルの初期値(*ComSignalInitValue*) を利用して送信モードの決定を行う【COM228】.

Com_IpduGroupControl の呼び出しによって送信 IPDU が有効化された場合、その時点で既に更新されているシグナルに対しては、そのシグナルの最新の値を利用して送信モードの決定を行う【COM229】.

Com_IpduGroupControl の呼び出しによって IPDU が有効化された場合、その IPDU に含まれるシグナルに設定した DM を開始する【COM733】.

Com_Init 後に初めて Com_IpduGroupControl を呼び出す際に、引数 initialize が FALSE で指定された場合、TRUE を指定した場合と同じ結果になる【COM740】.

表 2-2 有効化された IPDU に対する振舞い

受信側	送信側
1. Com_IpduGroupControl のパラメータが TRUE の場合、受信 DM 用タイマを再初期化 • ComSignalFirstTimeout • ComSignalTimeout • ComSignalGroupFirstTimeout • ComSignalGroupTimeout 2. Com_RxIndication および Com_TpRxIndication の処理を行う 3. Com_ReceiveSignal および Com_ReceiveSignalGroup の処理を行う	1. 以下のシステムサービス内の処理を行う • Com_InvalidateSignal • Com_SendSignal • Com_SendSignalGroup • Com_SendSignalGroup 2. 次の送信要求を受け取るまではタイムアウト通知を行わない 3. Com_TxConfirmation および Com_TpTxConfirmation の処理を行う 4. Com_TriggerTransmit の処理を行う 5. 周期送信の周期をリセット

2.2.4.3 IPDU グループの停止

Com_IpduGroupControl の呼び出しによって、起動状態の IPDU グループが停止された場合、その IPDU グループの状態は停止状態に遷移する。

ある送信 IPDU を無効化して送信を停止するためには、その IPDU のすべての所属 IPDU グループの状態が停止状態でなければならない。したがって、listen-only-mode を実現するためには、送信 IPDU が所属している IPDU グループすべてを停止しなければならない。停止された IPDU グループに受信 IPDU が含まれている場合、その受信 IPDU も無効化される。

Com_SendSignal または Com_SendSignalGroup または Com_InvalidateSignal の呼び出しによる内部バッファの更新は、無効化された IPDU に対しても同じように行われる【COM334】。

無効化された送信 IPDU に含まれるシグナルの値が更新された際に、そのシグナルが送信トリガであったとしても、IPDU 送信は行わず、そのトリガの保存も行わない。よって、その送信 IPDU が有効化された場合でも、即時に送信されることはない。

Com_IpduGroupControl の呼び出しによって送信 IPDU が無効化された場合、COM はその送信 IPDU の送信 DM をキャンセルし、以降の送信完了通知を無視する【COM115】。

COM は無効化された IPDU に対する Com_TriggerTransmit の呼び出しを禁止することはできないが、呼び出された場合は E_NOT_OK を返す【COM001】。

Com_IpduGroupControl の呼び出しによって無効化された送信 IPDU に対して、届いていない送信完了通知が存在する場合、COM はエラー通知(ComErrorNotification)に指定されたコールバック関数を呼び出す【COM479】。

ただし、IPDU にエラー通知(*ComErrorNotification*) が指定されていない場合は何もしない。

Com_IpduGroupControl の呼び出しによって送信中の大型 IPDU が無効化された場合、COM はその大型 IPDU に対する送信処理を即時に停止する【COM714】。

Com_IpduGroupControl の呼び出しによって受信 IPDU が無効化された場合、COM はその受信 IPDU に対する受信処理を行わない【COM684】。

Com_IpduGroupControl の呼び出しによって受信中の大型 IPDU が無効化された場合、COM はその大型 IPDU に対する受信処理を行わず、受信途中であっても無視する【COM713】。

Com_IpduGroupControl の呼び出しによって受信 IPDU が無効化された場合、COM はその受信 IPDU の DM をキャンセルする【COM685】。

表 2-3 無効化された IPDU に対する振舞い

受信側	送信側
1. 受信 DM 停止 2. Com_RxIndication における受信処理不可 3. Com_ReceiveSignal および Com_ReceiveSignalGroup が呼び出された場合、内部バッファの値を受信シグナルのデータとして返し、戻り値を COM_SERVICE_NOT_AVAILABLE とする	1. 送信停止 2. 送信 DM 停止 3. Com_TxConfirmation において ・ タイムアウトを無視 ・ RTE での使用を無視 4. Com_SendSignal および Com_SendSignalGroup および Com_InvalidateSignal が呼び出された場合、内部バッファ値を更新し、戻り値を COM_SERVICE_NOT_AVAILABLE とする 5. 保留された送信要求をキャンセル 6. Com_TriggerTransmit が呼び出された場合、戻り値を E_NOT_OK とする

表 2-3 において、無効化された IPDU に対する振る舞いとして、「RTE での使用を無視」という記述がある。これは PduR への IPDU 送信後に、Com_TxConfirmation が呼び出される前に IPDU が無効化された場合、その後 Com_TxConfirmation を呼び出されても RTE への通知を行わないものと解釈する【COMa002】。

2.2.5 デッドラインモニタリング(DM)

シグナルグループに対する DM 処理が、シグナルに対する処理と同様な場合は特記しない。

ユーザは、DM の監視する時間として *初回タイムアウト(ComFirstTimeout)* および *タイムアウト(ComTimeout)* を指定することができ、タイムアウト発生時に RTE に通知するためのコールバック関数を *タイムアウト通知(ComTimeoutNotification)* に指定することができる。なお、これらのパラメータは *シグナル(ComSignal)* および *シグナルグループ(ComSignalGroup)* ごとに指定される。

シグナルまたはシグナルグループの *タイムアウト(ComTimeout)* に値を指定しない、または 0 を指定した場合、そのシグナルまたはシグナルグループのデッドラインモニタリングは行われず、*初回タイムアウト(ComFirstTimeout)* の値も無視される【COM333】。

2.2.5.1 送信デッドラインモニタリング

送信処理はシグナルのアップデートビット有無による影響を受けず、IPDU ごとに行われるため、送信 DM は IPDU ごとに行われる。ただし、送信タイムアウト発生時の *タイムアウト通知(ComTimeoutNotification)* はシグナルごとに行う。

シグナルの *送信プロパティ(ComTransferProperty)* に依らず、かつ送信タイムアウト時間が指定されている場合、COM は送信モードにも依らずに送信 DM を行う【NCOM100】。

シグナルの *送信プロパティ(ComTransferProperty)* が PENDING または TRIGGERED のどちらであっても、送信 DM の振舞いに違いはない。

AUTOSAR 仕様との違い

AUTOSAR 仕様では、*送信プロパティ(ComTransferProperty)* が TRIGGERED に設定されている場合、タイムアウト値が設定されている場合送信モードに依らず送信 DM を行うとされているが、本仕様では、送信プロパティに依らず送信 DM を開始することとする【NCOM100】。

IPDU に含まれるシグナルの *タイムアウト(ComTimeout)* のうち、最小値をその IPDU のタイムアウト時間として使用する【COM445】。

送信 DM は、送信完了通知をサポートしているバスにのみ設定されるべきものであり、それ以外の場合は必ず送信タイムアウトが発生してしまう。

送信モード(ComTxModeMode) が NONE のみである IPDU に送信タイムアウト時間が設定された場合、その IPDU が所属している IPDU グループの起動時に送信 DM を開始する【COM696】。

送信モード(ComTxModeMode) が NONE のみである IPDU に送信タイムアウト時間が設定された場合、Com_TxConfirmation によって送信完了通知を受け取った際に、その IPDU の送信 DM をリセットする【COM697】。

TRUE モード(*ComTxModeTrue*) または *FALSE* モード(*ComTxModeFalse*) の送信モード (*ComTxModeMode*) の一方が *NONE* であり、もう一方が *NONE* でない場合、*NONE* 側の送信モードである間は、その IPDU の送信 DM は行われない【COMa021】。

シグナルグループの場合、ユーザは、送信タイムアウト時間をグループシグナルごとではなく、シグナルグループに対してのみ指定することができる。

送信キャンセルサポート有無(*ComIPduCancellationSupport*) が *TRUE* である場合、IPDU 種別 (*ComIPduType*) が *NORMAL* である IPDU の送信タイムアウトが発生した際に、COM は *PduR_ComCancelTransmit* を呼び出して送信要求をキャンセルする【COM708】。

PduR_ComCancelTransmit の呼び出しは、通常のタイムアウト発生時処理と併せて行う。この際、COM は *PduR_ComCancelTransmit* の戻り値を無視する。*PduR_ComCancelTransmit* が *E_OK* 以外を返すのは拡張エラーが有効になっている場合のみであるため、特に問題はない。

PDUR からの送信完了通知がなく、送信タイムアウト時間を経過して送信タイムアウトが発生した場合、COM はタイムアウト通知(*ComTimeoutNotification*) を呼び出して RTE に通知する【COM304】。

送信タイムアウト時間を経過した場合、その理由に依らずにタイムアウト通知を行う。そのため、MDT または IPDU コールアウト関数によるフィルタリングによって送信が延期されていたとしても、送信タイムアウトは発生し、タイムアウト通知が行われる。

Com_SendSignal または *Com_SendSignalGroup* の呼び出し完了時に送信 DM が開始される。

本仕様では、*Com_SendSignal* または *Com_SendSignalGroup* の呼び出し後の次の *Com_MainFunctionTx* 実行時に送信 DM を開始する【NCOM110】[NCOM006]。

送信モード(*ComTxModeMode*) が *DIRECT* または *MIXED* の IPDU に対して送信 DM を行う場合、送信タイムアウトが発生する前に、送信要求繰り返し回数(*ComTxModeNumberOfRepetitions*) に指定された回数分の送信要求を PDUR に送り、すべての送信完了通知を PDUR から受け取らなければならない。送信タイムアウト時間を経過する前に、送信要求を繰り返した回数分の送信完了通知を受け取りきれなかった場合、送信タイムアウト時間が発生し、RTE に対するタイムアウト通知が行われる。

送信モード(*ComTxModeMode*) が *DIRECT* であり、かつ送信要求繰り返し回数 (*ComTxModeNumberOfRepetitions*) が 0 よりも大きい場合、送信要求を繰り返した回数分の送信完了通知を受け取った後に送信 DM をキャンセルする【COM308】。

送信要求を繰り返した回数分の送信完了通知を受け取り、送信 DM をキャンセルした後、送信完了通知が RTE に送られる【COM305】。

送信モード(*ComTxModeMode*) が DIRECT であり、かつ送信要求繰り返し回数(*ComTxModeNumberOfRepetitions*) が 0 よりも大きい場合、新しい送信要求が起こった際に送信 DM をリセットする【COM739】。

2.2.5.2 受信デッドラインモニタリング

受信 DM を有効にしている IPDU グループに所属している IPDU のみ、受信 DM が有効となり、それ以外の IPDU の受信 DM は無効となる【COM772】。

アップデートビットを持つシグナルまたはシグナルグループに受信タイムアウト時間が設定されている場合、シグナルまたはシグナルグループごとに受信 DM を行う【COM292】。

アップデートビットを持たないシグナルまたはシグナルグループに受信タイムアウト時間が設定されている場合、IPDU ごとに受信 DM を行う【COM290】。

アップデートビットを持たないシグナルまたはシグナルグループの初回タイムアウト(*ComFirstTimeout*) およびタイムアウト(*ComTimeout*) のうち、IPDU 内の最小値をそれぞれ用いて IPDU ごとの受信 DM を行う【COM291】。

受信 DM とアップデートビットの関係を以下の図に示す。

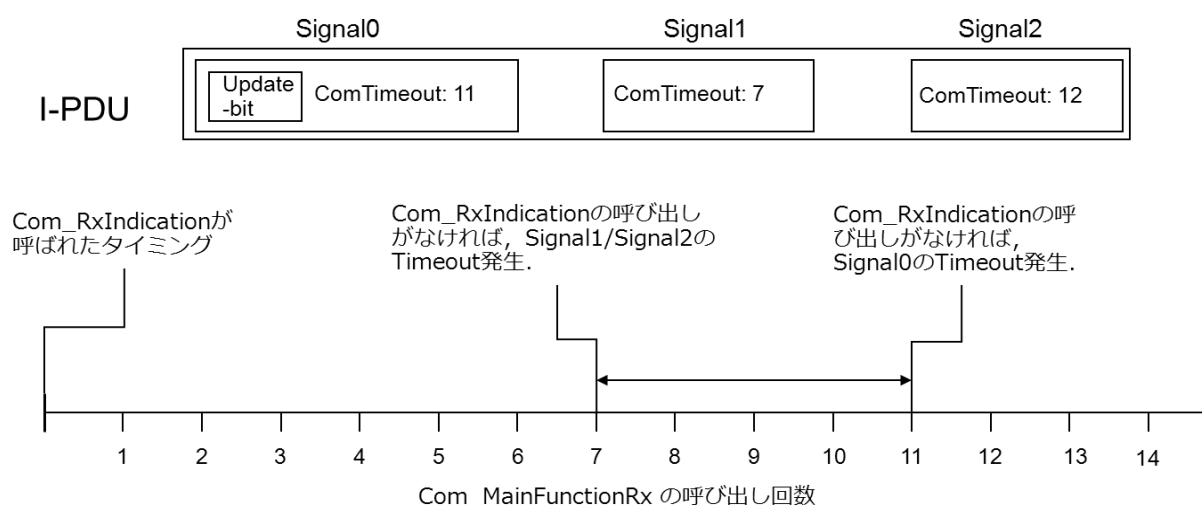


図 2-1 受信 DM とアップデートビットの関係

受信タイムアウトが発生した場合の処理は、受信タイムアウト時処理(*ComRxDataTimeoutAction*)で指定する。

受信タイムアウトが発生したシグナルの受信タイムアウト時処理(*ComRxDataTimeoutAction*) が REPLACE の場合、そのシグナルの値を初期値(*ComSignalInitValue*) に置き換える【COM470】。

シグナルの値を初期値(*ComSignalInitValue*) で置き換えた場合、最後に受信した値は失われるため、新しい値を受信するまでは初期値(*ComSignalInitValue*) がシグナルの値として扱われる。

受信タイムアウト発生によりシグナルまたはシグナルグループの値を初期値(*ComSignalInitValue*)に置き換えたとしても、フィルタリングのための前回値(*old_value*)は更新しない【COM393】。

受信タイムアウトが発生したシグナルグループの受信タイムアウト時処理(*ComRxDataTimeoutAction*) が REPLACE の場合、そのシグナルグループに含まれるすべてのグループシグナルの値を初期値(*ComSignalInitValue*) に置き換える【COM513】。

受信タイムアウトが発生したシグナルの受信タイムアウト時処理(*ComRxDataTimeoutAction*) が NONE または省略されている場合、そのシグナルの値の置き換えは行わない【COM500】。

Com_RxIndication または *Com_TpRxIndication* によって受信通知を受けた IPDU または大型 IPDU に受信 DM が設定されている場合、COM はその受信 DM タイマをリセットする【COM715】。

受信 DM では受信したシグナルの値を考慮しないため、無効値を受信した場合でも受信 DM タイマをリセットして受信 DM を再スタートする【COM738】。

アップデートビットを持つシグナルは、シグナルごとに受信 DM が行われる【COM292】。

アップデートビットがセットされているシグナルを受信した場合、そのシグナルの受信 DM タイマはリセットされる【NCOM031】。

シグナルまたはシグナルグループの初回タイムアウト(*ComFirstTimeout*) に 0 が指定されているまたは省略されている場合、所属 IPDU が有効化された後から初めて受信されるまでの間の受信 DM は行われない【COM716】。

受信 DM を開始するタイミングは周期関数の呼出しとは関係ないため、初回タイムアウト時間(*ComFirstTimeout*) に指定された時間を確実に待機するために、算出した周期関数呼出し回数に 1 を加えたタイミングで次の受信タイムアウトが検出される【NCOM033】。

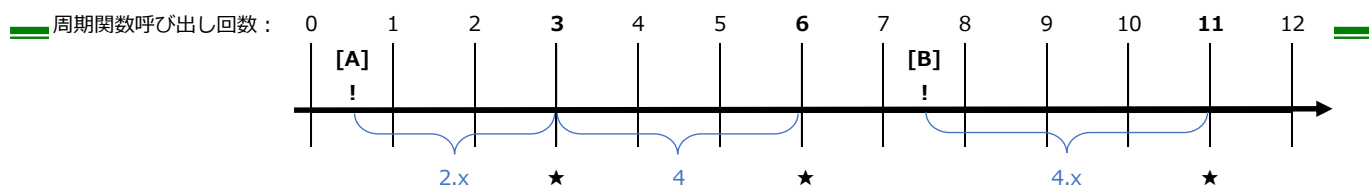
受信 DM タイマが *Com_RxIndication* によってリセットされた場合、受信 DM を再開するタイミングが周期関数の呼び出しとは関係なくなるため、タイムアウト時間(*ComTimeout*) に指定された時間を確実に経過するために、算出した周期関数呼出し回数に 1 を加えたタイミングで次の受信タイムアウトが検出される【NCOM034】。

Com_RxIndication が呼び出されず、周期関数において受信 DM タイマがリセットされた場合、算出した周期関数呼び出し回数のタイミングで次の受信タイムアウトが検出される【NCOM035】。

例

<i>ComRxTimeBase</i>	: 1.0
<i>ComFirstTimeout</i>	: 2.0 (周期関数呼出し回数は3(2+1)回)
<i>ComTimeout</i>	: 3.0 (周期関数呼出し回数は3回 or 4(3+1)回)

★…受信タイムアウト発生



[A]のタイミングで受信 DM が開始された場合、周期関数呼び出しの 2 回目で受信タイムアウトを検出してしまうと、1.x 秒で受信タイムアウトが発生したことになってしまう。そこで、周期関数呼び出しの 3 回目でタイムアウト検出を行う。この後、Com_RxIndication が呼び出されないままであれば、タイムアウト時間(ComTimeout) から算出した周期関数呼び出し回数の通り、3 回目の[6]でタイムアウト検出を行う。この後、[B]のタイミングで Com_RxIndication が呼び出され、受信 DM タイマが タイムアウト時間(ComTimeout) でリセットされた場合、2.x 秒でタイムアウト通知してしまうことを避けるために、その後の 4 回目の[11]でタイムアウト検出を行う。

一方、送信 DM を開始するタイミングは周期関数呼び出しのタイミングのみであるので、一律算出した周期関数呼び出し回数分でタイムアウト検出を行う。

2.2.5.2.1 受信 DM の有効化および無効化

ユーザは、Com_ReceptionDMControl を呼び出すことにより、IPDU グループごとに受信 DM の有効化および無効化を行うことができる。受信 DM が有効化された IPDU グループに所属している IPDU は、受信 DM が有効となり、それ以外の IPDU は受信 DM が無効となる【COM772】。

IPDU の受信 DM が無効な場合、受信タイムアウトが発生しても RTE への受信タイムアウト通知は行われない。ただし、受信 DM の無効化によって IPDU の受信が停止されることはない。

IPDU の受信 DM が有効な場合、受信タイムアウト発生時に RTE に対する受信タイムアウト通知が行われる。

Com_ReceptionDMControl により受信 DM の状態を無効から有効に切り替えた場合、初回タイムアウト(ComFirstTimeout) の値を受信 DM タイマに設定して受信 DM を開始する【COM224】。

Com_ReceptionDMControl において、受信 DM が既に有効な IPDU に対する、受信 DM 有効化の要求は無視される【COM486】。

Com_ReceptionDMControl の呼び出し時に、指定された IPDU グループに含まれている送信 IPDU は無視される【COM534】。

Com_ReceptionDMControl において、受信 DM が既に無効な IPDU に対する、受信 DM 無効化の要求は無視される【COM225】。

2.2.6 最小遅延タイム(MDT)

ユーザは、MDT によって送信の最小遅延時間を確保することができる。

Com_IpduGroupControl の呼び出しによって IPDU が有効化された際に、MDT はリセットされる [COM787]。したがって、MDT を停止後すぐに IPDU を有効化することにより、最小遅延時間を違反することができる。

なお、送信 DM は MDT による送信の遅延からの影響を受けない。

最小遅延時間(*ComMinimumDelayTime*) に値を指定しない、または 0 を指定した場合、MDT 監視は行われない【COM471】。

周期送信 MDT 有無(*ComEnableMDTForCyclicTransmission*) に FALSE を指定した場合、以下に対する MDT 監視は行われない【COM789】。

- ・ 送信モード(*ComTxModeMode*) に PERIODIC が指定された IPDU の周期送信
- ・ 送信モード(*ComTxModeMode*) に MIXED が指定された IPDU の周期送信
- ・ 送信モード(*ComTxModeMode*) に DIRECT が指定され、かつ送信要求繰り返し回数(*ComTxModeNumberOfRepetitions*) に 1 以上が指定された IPDU の直接 N 回送信

最小遅延時間(*ComMinimumDelayTime*) に 0 より大きい値を指定した場合、COM は PduR_ComTransmit を呼び出して PDUR へ IPDU を送信する度に MDT をリセットし、監視を開始する【COM698】。

最小遅延時間(*ComMinimumDelayTime*) に 0 より大きい値を指定した場合、COM は Com_TxConfirmation が呼び出された際に、既に監視が開始している MDT のカウンタに最小遅延時間(*ComMinimumDelayTime*) の値を再設定して、PduR_ComTransmit を呼び出した際に MDT 監視を開始する【COM469】。

IPDU の送信完了通知を受け取った際に最小遅延時間(*ComMinimumDelayTime*) を経過していない場合、監視中の MDT カウンタは再設定される。最小遅延時間(*ComMinimumDelayTime*) を経過しない限り、次の IPDU 送信は行われない。

ただし、Com_TriggerTransmit では最小遅延時間(*ComMinimumDelayTime*) を考慮しない【COM475】。

送信 DM においてタイムアウトが発生した場合、MDT カウンタは 0 で初期化し、MDT 待ちである場合は、即時に送信される【COMa006】。

2.2.7 受信完了通知タイミング

割込み駆動およびポーリングシステムの両方をサポートするために、受信完了通知を送るタイミングを指定することができる。ユーザは、IPDU のシグナル処理タイミング(*ComIPduSignalProcessing*) に IMMEDIATE または DEFERRED を指定する [COM119_Conf]。

Com_RxIndication または *Com_TpRxIndication* の呼び出しにより受信した IPDU のシグナル処理タイミング(*ComIPduSignalProcessing*) が IMMEDIATE の場合、COM は即時にアンパッキングおよび受信完了通知(*ComNotifications*) の呼び出しを行う 【COM300】。

Com_RxIndication または *Com_TpRxIndication* の呼び出しにより受信した IPDU のシグナル処理タイミング(*ComIPduSignalProcessing*) が DEFERRED の場合、COM は受信した IPDU のデータを内部バッファにコピーし、次の *Com_MainFunctionRx* のタイミングで、アンパッキングおよび受信完了通知(*ComNotifications*) の呼び出しを行う 【COM301】。

【COM301】のようにアンパッキングが遅延されており、シグナルバッファを更新する前に *Com_ReceiveSignal* が呼び出された場合、更新前のシグナルバッファの値を返す。

Com_RxIndication において、COM は受信した IPDU のデータの長さ(*PduInfoPtr->SduLength*)をチェックしてからアンパッキングし、完全に受信したシグナルに対してのみ受信完了通知を行う 【COM574】。

Com_RxIndication において、COM は受信した IPDU のデータの長さ(*PduInfoPtr->SduLength*)をチェックしてからアンパッキングし、完全に受信したシグナルグループに対してのみ受信完了通知を行う 【COM575】。

これにより、不整合な複合型(シグナルグループ)の受信を防ぐ。

受信した IPDU のデータの長さが期待値よりも短い場合、COM はその IPDU に対する受信処理を行わない 【COMa005】。

2.2.8 エンディアン変換

必要な AUTOSAR データ型(符号あり整数型, 符号なし整数型, ASCII, 列挙型, OPAQUE 型)をサポートするために, COM はすべての整数型に対するエンディアン変換をサポートする【COM007】. すなわち, 整数型以外の AUTOSAR データ型(ASCII, 列挙型, OPAQUE 型)のデータに関しては, そのデータの内容を解釈せずに扱う(AUTOSAR データ型の詳細は参考文書「次世代車載システム向け RTE 外部仕様書」を参照).

COM は以下のデータ型をサポートする【COM675】.

- BOOLEAN
- UINT8
- UINT16
- UINT32
- SINT8
- SINT16
- SINT32
- UINT8_N
- UINT8_DYN
- FLOAT32
- FLOAT64

AUTOSAR データ型のうち, OPAQUE 型のデータを, COM は UINT8_N または UINT8_DYN として解釈し, 常に n バイトの大きさのシグナルにマッピングする【COM472】.

ユーザは, OPAQUE 型のデータのエンディアン(*ComSignalEndianness*)には OPAQUE を指定しなければならない【COM553】【COM157_Conf】.

COM は, 符号あり整数型のデータに対しても, エンディアン変換を行う【COM674】.

COM は, 送信 IPDU のコールアウト関数を呼び出す前に, エンディアン変換を行う【COM221】.

シグナルの型(*ComSignalType*)が FLOAT32 または FLOAT64 の場合, COM はエンディアン変換以外の変換(分数・指数・記号・バイアス値の複雑な変換や正規化)をサポートしてはならない【COM580】.

2.2.9 符号拡張

COM は、受信したデータを受信シグナルの型(*ComSignalType*) の大きさに合わせるために符号拡張する【COM008】.

COM は、シグナルの初期値(*ComSignalInitValue*) を型(*ComSignalType*) の大きさに合わせるために符号拡張する【COM723】.

受信したシグナルの型(*ComSignalType*)が FLOAT32 または FLOAT64 の場合、符号拡張は行わない【COM579】.

COM は、受信したシグナルに対して、フィルタリングおよび通知検出を行う前に、符号拡張を行う【COM352】.

送信側では符号拡張を行わない【COM353】.

COM は、符号あり整数型のデータを符号拡張する場合、負の値も正しくマッピングする.

例：受信したシグナルのサイズ(*ComSignalSize*) が 5 ビットであり、その受信シグナルの型(*ComSignalType*) が SINT8 の場合の符号拡張

受信シグナル				1	1	1	0	1
↓ 符号拡張								
シグナルバッファ	1	1	1	1	1	1	0	1

※受信した値(-3)のまま保持する

2.2.10 フィルタリング

フィルタリング結果は TRUE または FALSE のいずれかとなる【COM694】.

フィルタリングは受信側でのみ行われる【COM695】.

受信側で行われたフィルタリング結果が FALSE となった場合, そのシグナルは破棄され, それ以降の処理は行われない【COM273】〔COM303〕.

送信側ではフィルタリングを行わないが, TMC 判定のためにフィルタを利用する【COM602】. TMC 判定については, 「送信モード」を参照.

COM はグループシグナルに対するフィルタリングをサポートしない【COM327】. ただし, TMC 判定のためにフィルタを利用する可能性はある〔COM326〕.

COM は, 符号ありのデータ型に対しても〔COM146_Conf〕で設定されたフィルタリングをサポートする【COM132】.

ユーザは, フィルタアルゴリズム(*ComFilterAlgorithms*)に以下を設定できる〔COM146_Conf〕.

- ALWAYS
- NEVER
- MASKED_NEW_EQUALS_X
- MASKED_NEW_DIFFERS_X
- MASKED_NEW_DIFFERS_MASKED_OLD
- NEW_IS_WITHIN
- NEW_IS_OUTSIDE
- ONE_EVERY_N

AUTOSAR 仕様では, フィルタリングの複雑度低減のために, OSEK 仕様で定義されているフィルタアルゴリズムの内の一部のみをサポートしている.

COM は, 〔COM146_Conf〕で設定されたフィルタアルゴリズムをサポートする【COM325】. ただし, 以下の例外を除く.

シグナルの型(*ComSignalType*)が UINT8_N または UINT8_DYN の場合, そのシグナルのフィルタアルゴリズム(*ComFilterAlgorithms*)は ALWAYS または NEVER のみサポートする【COM380】.

シグナルの型(*ComSignalType*)が BOOLEAN の場合, そのシグナルのフィルタアルゴリズム(*ComFilterAlgorithms*)は ALWAYS, NEVER, MASKED_NEW_EQUALS_X, MASKED_NEW_DIFFERS_X, MASKED_NEW_DIFFERS_MASKED_OLD, ONE_EVERY_N のみ

をサポートする【COM439】.

シグナルの型(*ComSignalType*) が FLOAT32 または FLOAT64 の場合, そのシグナルに対するフィルタリングはサポートしない【COM319】.

シグナルの型(*ComSignalType*) と フィルタアルゴリズム(*ComFilterAlgorithms*)のサポート関係を以下の表 2-4 に示す.

表 2-4 *ComSignalType* と *ComFilterAlgorithms* の対応表

型 (<i>ComSignalType</i>)	フィルタアルゴリズム(<i>ComFilterAlgorithms</i>)							
	ALWAYS	NEVER	MASKED_NEW_EQUALS_X	MASKED_NEW_DIFFERS_X	MASKED_NEW_DIFFERS_MASKED_OLD	NEW_IS_WITHIN	NEW_IS_OUTSIDE	ONE EVERY_N
BOOLEAN	○	○	○	○	○	×	×	○
UINT8	○	○	○	○	○	○	○	○
UINT16	○	○	○	○	○	○	○	○
UINT32	○	○	○	○	○	○	○	○
SINT8	○	○	○	○	○	○	○	○
SINT16	○	○	○	○	○	○	○	○
SINT32	○	○	○	○	○	○	○	○
FLOAT32	×	×	×	×	×	×	×	×
FLOAT64	×	×	×	×	×	×	×	×
UINT8_N	○	○	×	×	×	×	×	×
UINT8_DYN	○	○	×	×	×	×	×	×

あるフィルタリングにおいて, フィルタ対象シグナルの値を”今回値(new_value)”と呼び, 最後にフィルタを通過したシグナルの値を”前回値(old_value)”と呼ぶ.

前回値が必要なフィルタを持つシグナルに対して、COM は以下の処理を行う。

- ・ 初期化時、前回値にシグナルの初期値(*ComSignalInitValue*) をコピーする【COM603】
- ・ フィルタリング結果が TRUE となった場合、前回値に今回値をコピーする【COM302】
- ・ フィルタリング結果が FALSE となった場合、前回値を更新しない【COM303】

アプリケーションによって今回値が更新されるまでの間、COM は、シグナルの初期値(*ComSignalInitValue*) を今回値として使用する【COM604】。

2.2.10.1 フィルタアルゴリズム

各フィルタアルゴリズムのフィルタリング内容を以下に示す。

ALWAYS

フィルタアルゴリズム(*ComFilterAlgorithms*) が ALWAYS の場合、常に TRUE となる【CCOM2201】。

NEVER

フィルタアルゴリズム(*ComFilterAlgorithms*) が NEVER の場合、常に FALSE となる【CCOM2202】。

MASKED_NEW_EQUALS X

フィルタアルゴリズム(*ComFilterAlgorithms*) が MASKED_NEW_EQUALS_X の場合、マスクした今回値と *X* の値(*ComFilterX*) が一致していれば TRUE となり、異なっていれば FALSE となる【CCOM2203】。

MASKED_NEW_DIFFERS X

フィルタアルゴリズム(*ComFilterAlgorithms*) が MASKED_NEW_DIFFERS_X の場合、マスクした今回値と *X* の値(*ComFilterX*) が異なっていれば TRUE となり、一致していれば FALSE となる【CCOM2204】。

MASKED_NEW_DIFFERS MASKED_OLD

フィルタアルゴリズム(*ComFilterAlgorithms*) が MASKED_NEW_DIFFERS_MASKED_OLD の場合、マスクした今回値とマスクした前回値が異なっていれば TRUE となり、一致していれば FALSE となる【CCOM2205】。

NEW_IS WITHIN

フィルタアルゴリズム(*ComFilterAlgorithms*) が NEW_IS_WITHIN の場合、今回値が指定された範囲内(指定値を含む)であれば TRUE となり、範囲外であれば FALSE となる【CCOM2206】。

NEW_IS OUTSIDE

フィルタアルゴリズム(*ComFilterAlgorithms*) が NEW_IS_OUTSIDE の場合、今回値が指定された範囲外であれば TRUE となり、範囲内(指定値を含む)であれば FALSE となる【CCOM2207】.

ONE EVERY N

フィルタアルゴリズム(*ComFilterAlgorithms*) が ONE EVERY_N の場合、フィルタリングを N 回行う度に TRUE となり、それ以外は FALSE となる【CCOM2205】.

フィルタアルゴリズム(*ComFilterAlgorithms*) が ONE EVERY_N の場合、COM は以下の処理を行う【COM231】.

- ・ OCCURRENCE の値と PERIOD の値が一致した場合、OCCURRENCE を 0 にする
- ・ OCCURRENCE の値と OFFSET の値が一致した場合、フィルタリング結果を TRUE とする
- ・ フィルタリング処理を行った後、OCCURRENCE の値をインクリメントする

なお、Com_IpduGroupControl が呼び出されたことによってある IPDU が開始された場合、その IPDU に所属するシグナルが持つフィルタの OCCURRENCE は 0 にクリアされる【COM787】.

フィルタアルゴリズム(*ComFilterAlgorithms*) が ONE EVERY_N の場合のフィルタリング例を、図 2-2 に示す. なお、このコンフィギュレーション情報は以下のとおり.

- ・ ComFilterOffset【COM313_Conf】 : 5
- ・ ComFilterPeriod【COM312_Conf】 : 8

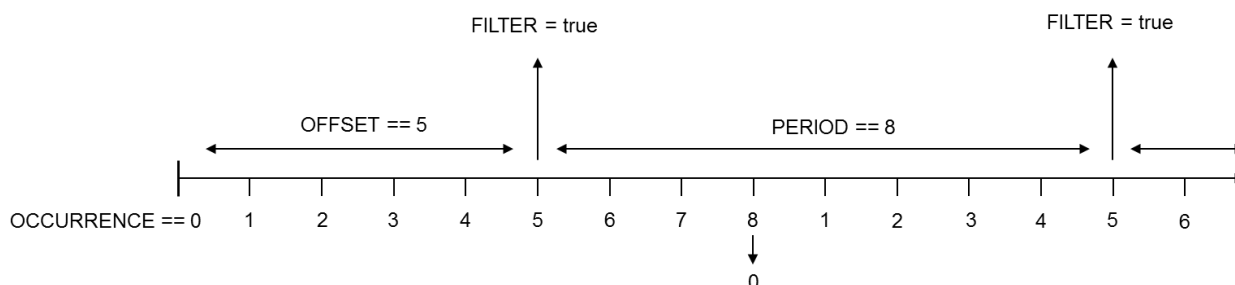


図 2-2 ONE EVERY_N のフィルタリング例

2.2.11 送信モード

2.2.11.1 送信プロパティと送信モード

2.2.11.1.1 シグナル

ユーザは、シグナルごとに送信プロパティ(*ComTransferProperty*)を設定することが可能であり、IPDU ごとに送信モード(*ComTxModeMode*)を設定することが可能である。Com_SendSignal によって送信要求を受けたシグナルの送信プロパティ(*ComTransferProperty*)と、所属 IPDU 送信モード(*ComTxModeMode*)の組合せによる、IPDU 送信タイミングの違いを以下の表 2-5 に示す。

表 2-5 送信モードおよび送信プロパティと IPDU 送信タイミングの関係

送信モード (<i>ComTxModeMode</i>)	送信プロパティ(<i>ComTransferProperty</i>)				
	PENDING	TRIGGERED	TRIGGERED_ON_CHANGE	TRIGGERED_WITHOUT_REPETITION	TRIGGERED_ON_CHANGE_WITHOUT_REPETITION
DIRECT	送信しない	直接 N 回送信	バッファ値を変更した場合 直接 N 回送信	直接 1 回送信	バッファ値を変更した場合 直接 1 回送信
MIXED	送信しない (周期送信のみ)	直接 N 回送信 (+周期送信)	バッファ値を変更した場合 直接 N 回送信 (+周期送信)	直接 1 回送信 (+周期送信)	バッファ値を変更した場合 直接 1 回送信 (+周期送信)
PERIODIC	送信しない(周期送信のみ)				
NONE	送信しない(トリガ送信のみ)				

送信要求を受けたシグナルの送信プロパティ(*ComTransferProperty*)が TRIGGERED であり、所属 IPDU の送信モード(*ComTxModeMode*)が DIRECT または MIXED である場合、COM はその IPDU の直接送信を *N* 回(*ComTxModeNumberOfRepetitions*)繰り返して行う【COM330】。

送信要求を受けたシグナルの送信プロパティ(*ComTransferProperty*)が TRIGGERED_WITHOUT_REPETITION であり、所属 IPDU の送信モード(*ComTxModeMode*)が DIRECT または MIXED である場合、COM はその IPDU の直接送信を 1 回行う【COM767】。

送信要求を受けたシグナルの送信プロパティ(*ComTransferProperty*)が TRIGGERED_ON_CHANGE であり、所属 IPDU の送信モード(*ComTxModeMode*)が DIRECT または MIXED である場合、送信シグナルの値とシグナルバッファの値が異なっていれば、COM はその IPDU の直接送信を *N* 回(*ComTxModeNumberOfRepetitions*)繰り返して行う【COM734】。

送信要求を受けたシグナルの送信プロパティ(*ComTransferProperty*)が TRIGGERED_ON_CHANGE_WITHOUT_REPETITION であり、所属 IPDU の送信モード(*ComTxModeMode*)が DIRECT または MIXED である場合、送信シグナルの値とシグナルバッファの

値が異なっていれば、COM はその IPDU の直接送信を 1 回行う【COM768】。

送信モード(*ComTxModeMode*) が NONE の IPDU の送信を開始しない【COM135】。

送信モード(*ComTxModeMode*) が NONE の IPDU に対しては、Com_TriggerTransmit によって送信を要求することができる。

送信プロパティ(*ComTransferProperty*) が TRIGGERED であるシグナルの送信中に、TRIGGERED_WITHOUT_REPETITION であるシグナルを送信した場合、COM はその IPDU の直接送信を N 回繰り返して行う【NCOM098】。

送信要求を受けたシグナルの送信プロパティ(*ComTransferProperty*) が PENDING であり、所属 IPDU の送信モード(*ComTxModeMode*) が DIRECT または MIXED である場合、COM はその IPDU の送信を開始しない【NCOM099】。

2.2.11.1.2 シグナルグループ

ユーザは、シグナルグループおよびグループシグナルごとに送信プロパティ(*ComTransferProperty*)を設定することが可能であり、IPDU ごとに送信モード(*ComTxModeMode*)を設定することが可能である。

送信要求を受けたシグナルグループの送信プロパティ(*ComTransferProperty*) が TRIGGERED であり、所属 IPDU の送信モード(*ComTxModeMode*) が DIRECT または MIXED である場合、COM はその IPDU の直接送信を N 回(*ComTxModeNumberOfRepetitions*) 繰り返して行う【COM741】。

送信要求を受けたシグナルグループの送信プロパティ(*ComTransferProperty*) が TRIGGERED_WITHOUT_REPETITION であり、所属 IPDU の送信モード(*ComTxModeMode*) が DIRECT または MIXED である場合、COM はその IPDU の直接送信を 1 回行う【COM769】。

送信要求を受けたシグナルグループの送信プロパティ(*ComTransferProperty*) が TRIGGERED_ON_CHANGE であり、そのシグナルグループに付属しているグループシグナルすべてに送信プロパティ(*ComTransferProperty*) が設定されておらず、所属 IPDU の送信モード(*ComTxModeMode*) が DIRECT または MIXED である場合、そのシグナルグループのグループシグナルの値のいずれか一つでもグループシグナルバッファの値と異なっていれば、COM はその IPDU の直接送信を N 回(*ComTxModeNumberOfRepetitions*) 繰り返して行う【COM742】。

送信要求を受けたシグナルグループの送信プロパティ(*ComTransferProperty*) が TRIGGERED_ON_CHANGE であり、そのシグナルグループに付属しているグループシグナルのいずれかに送信プロパティ(*ComTransferProperty*) が設定されており、所属 IPDU の送信モード(*ComTxModeMode*) が DIRECT または MIXED である場合、送信プロパティ(*ComTransferProperty*) が TRIGGERED_ON_CHANGE であるグループシグナルの値がグループシグナルバッファの値と異なっていれば、COM はその IPDU の直接送信を N 回(*ComTxModeNumberOfRepetitions*) 繰り返して行う【COM743】。

送信要求を受けたシグナルグループの送信プロパティ(*ComTransferProperty*) が TRIGGERED_ON_CHANGE_WITHOUT_REPETITION であり、所属 IPDU の送信モード(*ComTxModeMode*) が DIRECT または MIXED である場合、送信グループシグナルの値がグループシ

グナルバッファの値と異なっていれば、COM はその IPDU の直接送信を 1 回行う【COM770】.

2.2.11.2 TMS 判定

COM は下位層との通信を IPDU 単位で行い、IPDU には複数のシグナルを付属できる。ユーザは、送信 IPDU に異なる 2 つの送信モードを設定できる [COM032]。COM は、送信 IPDU に付属するシグナルの状態から、送信モードを選択する。このとき、シグナルの状態を TMC と呼び、送信モード選択結果を TMS と呼ぶ。

送信モードを選択するための条件は、コンフィギュレーション時に静的に設定される [COM676]。

COM は、送信シグナルのフィルタリング結果を TMC として扱う [COM602]。

フィルタアルゴリズム(*ComFilterAlgorithms*) が MASKED_NEW_DIFFERS_MASKED_OLD であるフィルタを持つシグナルに対しては、シグナルのサイズ(*ComBitSize*) を考慮した TMC 判定を行うが、他のフィルタを持つシグナルに対しては、シグナルの型(*ComSignalType*) の大きさを考慮した TMC 判定を行う [COM763]。

TMS 判定は IPDU ごとに行う [COM605]。

RTE が *Com_SendSignal* または *Com_SendSignalGroup* を呼び出した際に、COM は、更新したシグナルまたはシグナルグループが所属する IPDU の TMS 判定を行う [COM245]。

IPDU(*ComIPdu*) に付属するシグナル(*ComSignal*) のいずれかがフィルタ(*ComFilter*) を持つ場合、COM はフィルタを持つシグナルの TMC を用いて TMS 判定を行う [COM676]。

IPDU(*ComIPdu*) に付属するシグナル(*ComSignal*) のいずれもフィルタ(*ComFilter*) を持たない場合、COM はその IPDU の TMS を常に TRUE とする [COM677]。

IPDU に付属するいずれかのシグナルの TMC が TRUE である場合、COM はその IPDU の TMS を TRUE とする [COM678]。

IPDU に付属するすべてのシグナルの TMC が FALSE である場合、COM はその IPDU の TMS を FALSE とする [COM679]。

TMS 判定の条件および結果の概略を以下の表 2-6 に示す。

表 2-6 TMS 判定

条件		TMS
IPDU に付属するシグナルのすべてがフィルタを持たない		TRUE
IPDU に付属するシグナルのいずれかがフィルタを持つ	フィルタを持つシグナルのいずれかの TMC が TRUE	TRUE
	フィルタを持つシグナルのすべての TMC が FALSE	FALSE

COM は、TMS が TRUE となった場合は *TRUE* モード(*ComTxModeTrue*) の送信モード(*ComTxModeMode*) を使用し、TMS が FALSE となった場合は *FALSE* モード(*ComTxModeFalse*) の

送信モード(*ComTxModeMode*) を使用する 【COM032】.

COM は、TMS 判定により決定した送信モードと、各シグナルの送信プロパティに従って、IPDU の送信タイミングを決定する 【COM238】.

TMS 判定により IPDU の送信モードが変更された場合、COM は変更後の送信モードを即時に適用し、必要に応じて *PduR_ComTransmit* を呼び出す 【COM239】.

TMS 判定により IPDU の送信モードが変更された場合、COM は周期送信のための周期タイマをリセットして再スタートする 【COM244】.

RTE が *Com_SendSignal* および *Com_SendSignalGroup* を呼び出した際に、TMS 判定によって既に開始されている IPDU の送信モードが PERIODIC または MIXED に変更された場合、COM は即時に *PduR_ComTransmit* を呼び出し、変更後の送信モードを適用して新しい送信サイクルを開始する. この送信サイクルは、シグナルまたはシグナルグループの送信プロパティに関わらず、送信モードの変更により開始される. ただし、MDT および送信オフセット(*ComTxModeTimeOffset*) は考慮する 【COM495】.

TMS 判定により送信モードが DIRECT に変更された場合、即時に直接(N 回)送信が開始される. ただし、MDT は考慮する 【COM582】.

1 回の *Com_MainFunctionTx* の処理内では、同じ IPDU の送信を複数回行わない 【COM478】.

COM は、グループシグナルの状態から、設定された条件に従って TMS 判定を行う 【COM326】. このとき、送信モードの選択自体はシグナルの場合と同様に行われる.

2.2.11.3 直接 N 回送信

送信モード(*ComTxModeMode*) が DIRECT または MIXED である IPDU の送信要求を繰り返す回数は、送信要求繰り返し回数(*ComTxModeNumberOfRepetitions*) [COM281_Conf] によって設定される。

失敗送信要求リトライ有無(*ComRetryFailedTransmitRequests*) に FALSE が設定されており、IPDU の送信モード(*ComTxModeMode*) が DIRECT または MIXED であり、かつ送信要求繰り返し回数(*ComTxModeNumberOfRepetitions*) が 0 の場合、COM は PduR_ComTransmit を 1 回のみ呼び出し、送信コンファメーションの結果を考慮しない【COM467】。

IPDU の送信要求を繰り返している途中に、RTE から新しい送信要求を受け取った場合、COM は実行中の送信要求の繰り返し処理をキャンセルし、新しい送信要求に対する処理を即時に開始する【COM279】。

RTE が Com_SendSignal または Com_SendSignalGroup を呼び出すことにより送信することになった IPDU の送信モード(*ComTxModeMode*) が DIRECT または MIXED であり、送信要求繰り返し回数(*ComTxModeNumberOfRepetitions*) が 1 以上の場合、COM は PDUR への送信と RTE への送信完了通知を、以下の順に実行する【COM305】。

1. 送信要求繰り返しカウンタに送信要求繰り返し回数(*ComTxModeNumberOfRepetitions*) を代入する。
2. 送信要求繰り返しカウンタの値が 0 でない限り、COM は送信要求繰り返し周期(*ComTxModeRepetitionPeriod*) に従って PduR_ComTransmit を呼び出して IPDU の送信要求を送る。
3. PDUR から送信完了通知を受け取った際に、送信要求繰り返しカウンタの値が 0 でない場合、COM は送信要求繰り返しカウンタの値をデクリメントする。
PDUR から送信完了通知を受け取った際に、送信要求繰り返しカウンタの値が 0 である場合、COM はその送信完了通知を無視する。
4. 送信要求繰り返しカウンタの値が 0 になった場合、COM は RTE に対して送信完了通知を送り、その IPDU の送信 DM をキャンセルする【COM392】。

【COM305】は COM の実装を完全に定義してはいない。しかし、COM を実装する際には上記の定義に従って送信完了通知を実装しなければならない。これにより、ある極限状態における送信完了通知が周期から遅れて到着した場合は、周期を違反しても良いことになる。これは、CAN が LPDU のためのキューを持たないことを前提としている。特殊な状況においては、ハードウェアと CAN と CANIF の間で競合状態が発生し、過剰な送信を引き起こす可能性がある。

IPDU の送信要求を繰り返している途中に、PDUR が E_NOT_OK を返して送信失敗を通知したとし

ても、COM はその通知を無視する。COM は送信完了通知を受け取った場合にのみ送信要求繰り返しカウンタの値をデクリメントするため、失敗した送信要求を安全に無視することができる。

送信モード(*ComTxModeMode*) が MIXED である IPDU の直接 N 回送信と周期送信のタイミングが重なった場合、COM はその送信要求を直接 N 回送信のものとしてカウントする【COM494】。

直接 N 回送信が完了する前に、送信 DM のタイムアウトが発生した場合、COM はその IPDU の送信要求をそれ以上繰り返さない【COM392】。

2.2.11.4 周期送信

Com_IpduGroupControl によって起動された IPDU グループに含まれている IPDU の送信モード(*ComTxModeMode*) が PERIODIC または MIXED である場合、その IPDU の周期送信が開始される。周期送信が開始されてから最初の送信が行われるまでの間をオフセット時間として待機することが可能であり、そのオフセット時間は周期送信オフセット(*ComTxModeTimeOffset*) に指定される。最初の送信が行われて以降は、送信周期(*ComTxModeTimePeriod*) に指定された時間間隔で周期送信が行われる。

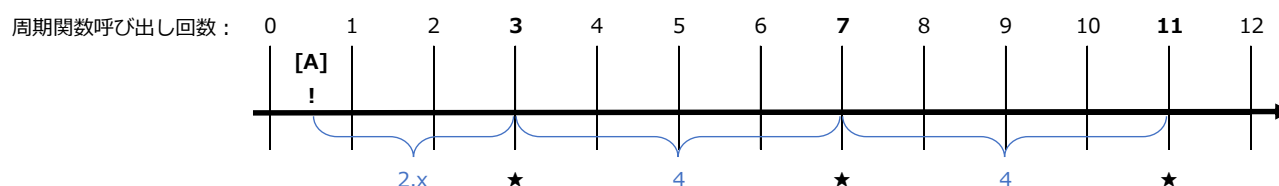
送信オフセット(*ComTxModeTimeOffset*) および送信周期(*ComTxModeTimePeriod*) は送信タイムベース(*ComTxTimeBase*) に指定された値から周期関数呼び出し回数に変換される【NCOM005】。

周期送信が開始されるタイミングは周期関数の呼び出しとは関係ないため、送信オフセット(*ComTxModeTimeOffset*) に指定された時間を確実に待機するために、算出した周期関数呼び出し回数に 1 を加えたタイミングで送信が行われる【NCOM020】。

例：周期送信のオフセット時間

ComTxTimeBase	: 1.0
ComTxModeTimeOffset	: 2.0 (周期関数呼び出し回数は3(2+1)回)
ComTxModeTimePeriod	: 4.0 (周期関数呼び出し回数は4回)

★…PduR_ComTransmitによる送信



上記において、[A]のタイミングで周期送信が開始された場合、周期関数呼び出しの 2 回目で送信してしまうと、オフセット時間が 1.x 秒になってしまう。そこで、周期関数呼び出しの 3 回目で最初の送信を行うことにより、指定された 2.0 秒を確実に待機する。以後の周期 4.0 秒に対しては、算出した周期関数呼び出し回数分の周期関数実行後に送信を行うことにより、指定した周期を実現する。

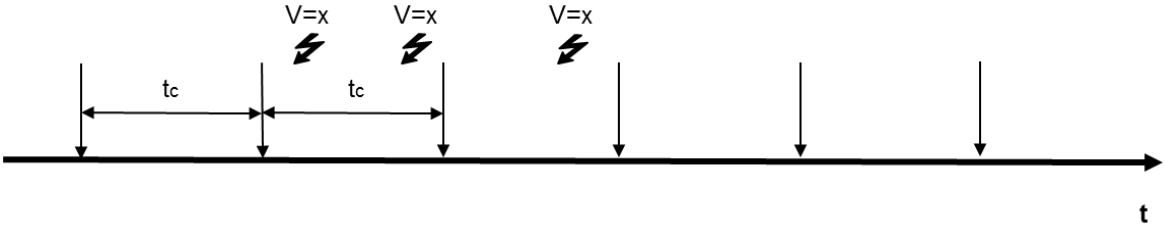
2.2.11.5 送信モードのユースケース

本章では、送信モードごとの代表的なユースケースについて記載する。ユースケースの凡例を以下の表 2-7 に示す。

表 2-7 ユースケース凡例

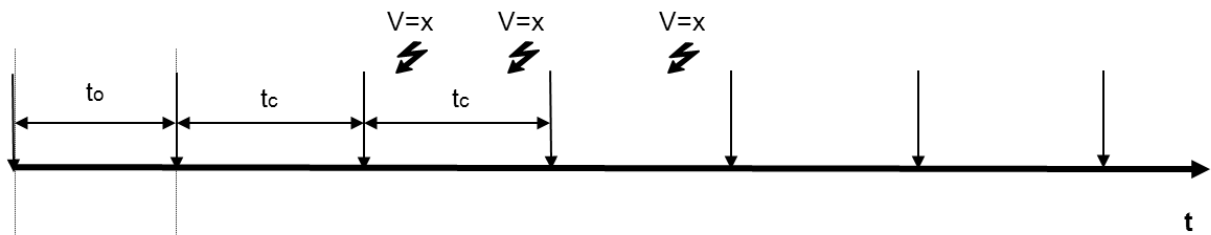
記号	説明
t_c, t_{c1}, t_{c2}	周期(<i>ComTxModeTimePeriod</i>)
t_d	送信要求繰り返し回数(<i>ComTxModeNumberOfRepetitions</i>)
t_o	周期オフセット(<i>ComTxModeTimeOffset</i>)
$t_{r,min}$	COM の最小ソフトウェア応答時間(内部の周期時間など)
V	通信データの値 x : 任意の値, またはその値の範囲, $a..w$: ユーザが定義した値, またはその値の範囲 $a <> b$: a と b の値の範囲は連続しない
	RTE から COM への要求
	COM から PDUR への要求
	保留された COM から PDUR への要求 (要因 : RTE からの新しい要求, MDT による遅延など)
dt	PDUR への送信要求の最小遅延時間(MDT) (I-PDU ごとに設定可能 : 最小遅延時間(<i>ComMinimumDelayTime</i>))
w/o TMS switch	TMS 判定による送信モードの変更なし
W TMS switch	TMS 判定による送信モードの変更あり

2.2.11.5.1 周期送信(送信モード変更なし)
 周期送信オフセットなし



ComSignal	
ComTransferProperty	PENDING
ComFilter	
ComFilterAlgorithm	ALWAYS
ComIPdu	
ComIPduDirection	SEND
ComTxModeTrue	
ComTxMode	
ComTxModeMode	PERIODIC
ComTxModeNumberOfRepetitions	—
ComTxModeRepetitionPeriod	—
ComTxModeTimeOffset	0
ComTxModeTimePeriod	tc

周期送信オフセットあり



ComSignal	
ComTransferProperty	PENDING
ComFilter	
ComFilterAlgorithm	ALWAYS
ComIPdu	
ComIPduDirection	SEND
ComTxModeTrue	
ComTxMode	
ComTxModeMode	PERIODIC
ComTxModeNumberOfRepetitions	—
ComTxModeRepetitionPeriod	—
ComTxModeTimeOffset	t_o
ComTxModeTimePeriod	t_c

2.2.11.6 VFB 仕様との関係

SWC 実装時は、どの通信媒体を使用するのかは不明である。そのため、その SWC からの送信要求に対する、バス固有の繰り返しおよびバス固有のタイミング処理は、COM(または適切なバスインタフェースおよびドライバ)によって隠蔽化する。

COM は、送信プロパティと送信モードの組み合わせにより、送信要求の繰り返しおよびバス固有のタイミング処理を実現する。

送信モードおよび送信プロパティと、VFB の送信モードの関係を以下の表 2-8 送信モードおよび送信プロパティと VFB 送信モードの関係表 2-8 に示す。

表 2-8 送信モードおよび送信プロパティと VFB 送信モードの関係

送信モード(ComTxModeMode)	送信プロパティ(ComTransferProperty)	
	TRIGGERED / TRIGGERED_ON_CHANGE	PENDING
DIRECT	DIRECT	—
MIXED	DIRECT	PERIODIC
PERIODIC	—	PERIODIC
NONE	—	—

2.2.12 失敗送信要求リトライ

失敗送信要求リトライ有無(*ComRetryFailedTransmitRequests*) が TRUE の場合、COM は PduR_ComTransmit の戻り値を評価しなければならない。PduR_ComTransmit の戻り値が E_OK でない場合、COM は、次の Com_MainFunctionTx のタイミングで PduR_ComTransmit を呼び出して送信失敗した IPDU の送信要求を再度行う【COM773】。

失敗送信要求リトライを行う前に IPDU バッファが更新された場合、最新のデータが送られる【COMa020】。

直接 N 回送信の間に失敗送信要求リトライが来た場合、繰り返し送信タイミングはリトライの影響を受けない【NCOM097】。

失敗送信要求リトライは送信 DM と独立した機能であり、失敗送信要求リトライによる IPDU の送信要求は送信 DM に影響を与えない【COM774】。

失敗送信要求リトライ有無(*ComRetryFailedTransmitRequests*) が TRUE であり、かつ、ある IPDU において送信 DM のタイムアウトが発生した場合、COM は、その IPDU に対する保留中の送信要求をすべて無効化する【COM775】。

周期送信のための周期タイマは、最初の送信を行った際に常に開始される【COM776】。

COM は、停止中の IPDU に対しては失敗送信要求リトライを行わない【COM777】。

失敗送信要求リトライ有無(*ComRetryFailedTransmitRequests*) が TRUE であり、かつ、Com_IpduGroupControl の引数 Initialize を TRUE に指定して呼び出されたことによってある IPDU が開始された場合、COM は、その IPDU に対する保留中の送信要求をすべて無効化する【COM778】。

失敗送信要求リトライ有無(*ComRetryFailedTransmitRequests*) が TRUE である場合、COM は、PduR_ComTransmit の戻り値を【COM773】に従って評価し、そうでない場合は【COM305】に従って評価する【COM779】。

2.2.13 シグナル無効化

送信側が有効なシグナルの値を提供できない場合、COM は無効値(*ComSignalDataInvalidValue*)を送信することでシグナルを無効化することができる。送信側で無効化されたシグナルを受信した場合、COM は無効値受信時処理(*ComDataInvalidAction*)の設定に従って対処する。

2.2.13.1 無効化されたシグナルの送信

COM は、*Com_InvalidateSignal* の内部で *Com_SendSignal* を呼び出すことにより無効値(*ComSignalDataInvalidValue*)を送信する【COM099】。

無効化されたシグナルを含む IPDU を下位層に送るタイミングは、送信モード(*ComTxModeMode*)および送信プロパティ(*ComTransferProperty*)によって決定される。*Com_InvalidateSignal* の内部で *Com_SendSignal* を呼び出すことにより、無効化されたシグナルに対しても TMS 判定が行われる。

グループシグナルごとに無効値を設定することが可能であるため、グループシグナルの無効化は *Com_InvalidateSignalGroup* または *Com_InvalidateSignal* により実現できる【COM520_Conf】。

VFB は複合データ型という属性のみを定義する。複合データ型の無効化とシグナルグループの無効化との最適なマッピング方法は、シグナルグループに付属するすべてのグループシグナルを無効化することである。RTE は *Com_InvalidateSignalGroup* を呼び出すことにより、シグナルグループを完全に無効化することが可能である。

2.2.13.2 無効化されたシグナルの受信

受信シグナルの値が無効値(*ComSignalDataInvalidValue*)と等しく、かつ、そのシグナルの無効値受信時処理(*ComDataInvalidAction*)が NOTIFY である場合、COM は、無効値受信通知(*ComInvalidNotification*)で定義されたコールバック関数を呼び出すことによって RTE へ通知し、そのシグナルに対するフィルタリングなどの受信処理や受信完了通知を行わない【COM680】。

ただし、無効化されたシグナルを受信した場合でも、受信 DM のタイマは再スタートされる。

受信シグナルの値が無効値(*ComSignalDataInvalidValue*)と等しく、かつ、そのシグナルの無効値受信時処理(*ComDataInvalidAction*)が NOTIFY である場合、COM は、シグナルバッファを更新しない【COM717】。

受信シグナルの値が無効値(*ComSignalDataInvalidValue*)と等しく、かつ、そのシグナルの無効値受信時処理(*ComDataInvalidAction*)が REPLACE である場合、COM は、受信シグナルの値を初期値(*ComSignalInitValue*)に置き換えて、有効な値を受信したかのように受信処理や受信完了通知を行う【COM681】。

無効値受信時処理(*ComDataInvalidAction*) が設定されていないシグナルを受信した場合、COM は常に有効な値を受信したかのように受信処理や受信完了通知を行う【COM736】。

受信シグナルグループに付属するいずれかのグループシグナルの値が無効値(*ComSignalDataInvalidValue*) と等しく、かつ、そのシグナルグループの無効値受信時処理(*ComDataInvalidAction*) が NOTIFY である場合、COM は、無効値受信通知(*ComInvalidNotification*) で定義されたコールバック関数を呼び出すことによって RTE へ通知し、そのシグナルグループに対するフィルタリングなどの受信処理や受信完了通知を行わない【COM682】。

受信シグナルグループに付属するいずれかのグループシグナルの値が無効値(*ComSignalDataInvalidValue*) と等しく、かつ、そのシグナルグループの無効値受信時処理(*ComDataInvalidAction*) が NOTIFY である場合、COM は、グループシグナルバッファを更新しない【COM718】。

グループシグナルを受信する前に *Com_ReceiveSignalGroup* が呼び出された場合、グループシグナルバッファの値は初期化された状態から更新されていないため、初期値(*ComSignalInitValue*) が返される。

受信シグナルグループに付属するいずれかのグループシグナルの値が無効値(*ComSignalDataInvalidValue*) と等しく、かつ、そのシグナルグループの無効値受信時処理(*ComDataInvalidAction*) が REPLACE である場合、COM は、そのシグナルグループに付属するすべてのグループシグナルの値を初期値(*ComSignalInitValue*) に置き換えて、有効な値を受信したかのように受信処理や受信完了通知を行う【COM683】。

無効値受信時処理(*ComDataInvalidAction*) が設定されていないシグナルグループを受信した場合、COM は常に有効な値を受信したかのように受信処理や受信完了通知を行う【COM737】。

2.3 シグナルグループ(複合データ型の IPDU へのマッピング)

AUTOSAR の複合データ型をサポートするために、COM はシグナルグループを提供する。シグナルグループの整合性を考慮して通信を行うことにより、複合データ型に必要な整合性を保証する。COM は、シグナルグループの整合性を考慮した通信を行うために、シャドウバッファを使用する。RTE はシャドウバッファに格納されたグループシグナルにアクセスして通信を行う。RTE は `Com_SendSignalGroup` または `Com_ReveiveSignalGroup` を呼び出すことにより、シャドウバッファと IPDU を同期することができる。

シグナルグループはコンフィギュレーション時に静的に指定され、それぞれのショートネームを利用したシンボル名が与えられる [COM345_Conf] [COM044]。

2.3.1 シグナルグループの初期化

`Com_Init` により、送信側のシグナルグループのシャドウバッファが初期化される [COM484]。

2.3.2 シグナルグループの送信

グループシグナルを指定して `Com_SendSignal` または `Com_InvalidateSignal` が呼び出された場合、COM はシャドウバッファの更新のみを行う。このとき、IPDU バッファは更新されないため、TMS 判定などの送信処理は不要である [COMa010]。

`Com_SendSignalGroup` により、シャドウバッファの値が IPDU バッファにパッキングされる [COM050]。

2.3.3 シグナルグループの受信

`Com_ReceiveSignalGroup` により、受信 IPDU のバッファ内にあるシグナルグループのデータがシャドウバッファにアンパッキングされた後、`Com_ReceiveSignal` によるシャドウバッファからグループシグナルの受信が可能となる。

`Com_ReceiveSignalGroup` により、IPDU バッファ内にあるシグナルグループのデータがシャドウバッファにアンパッキングされる [COM051]。

2.3.4 シグナルグループ関連の通知

送信側および受信側のどちらにおいても、完全なシグナルグループに対してのみ、グループシグナルおよび通知を指定することができる。

完全なシグナルグループに対して、送受信または無効化が行われた場合、COM は RTE に対して通知を行う [COM053] [COM345_Conf] [COM520_Conf]。

2.3.5 シグナルグループの属性

シグナルグループおよびグループシグナルに関するコンフィギュレーション情報および処理の一覧と、それぞれの対象について以下に示す。

表 2-9 シグナルグループの属性

属性	シグナルグループごと	グループシグナルごと
アップデートビット (ComUpdateBitPosition)	○ [COM257_Conf]	×
通知(ComNotification)	○ [COM498_Conf]	×
エラー通知 (ComErrorNotification)	○ [COM499_Conf]	×
タイムアウト通知 (ComTimeoutNotification)	○ [COM552_Conf]	×
無効値受信通知 (ComSignalDataInvalidValue)	○ [COM391_Conf]	×
送信要求	○ [COM200], [COM557]	○ [COM197], [COM203]
TMS 判定	×	○
受信要求	○ [COM201]	○ [COM198]
フィルタリング	×	×

2.4 大型データ

※大型データ通信は本バージョン対象外の機能であるため、本章の内容は対象外となる。

大型シグナルとは、下位層の通信プロトコルで扱われる LPDU に納まりきらない大きさのシグナルのことである。大型シグナルが所属している IPDU のことを大型 IPDU と呼び、これらを総括して大型データと呼ぶ。例えば、CAN の場合は 8 バイトを超えるシグナルは大型シグナルとなる。FlexRay の場合、1 つのフレームに複数の LPDU を含むことができるため、状況は複雑になる。したがって、FlexRay のフレームサイズを超えない大きさのシグナルを大型シグナルと定義しても良い。

大型シグナルは大型 IPDU に所属し、大型 IPDU は TP を介して下位層に送信される。大型 IPDU に 1 つ以上の大型シグナルが含まれている必要はない。

大型 IPDU および大型シグナルの大きさは 4095 バイトに制限される [COM437_Conf]。大型シグナルの長さは固定であり、実行時に変更することはできない。

大型シグナルの型(*ComSignalType*) は UINT8_N でなければならない。

COM は下位層のバス特性を認識しないため、ある IPDU が単一の LPDU で送信できるかどうかをコンフィギュレーション情報から取得する必要がある [COM761_Conf]。

また、COM は下位層のバスに依存しない PDUR のインタフェースを使用しているため、下位層の LPDU にとって大きすぎる IPDU を送信していないかどうかを検出して防ぐことができない。

しかし、COM は TP によって通信可能な最大サイズを許可しなければならないため、IPDU の長さに対するこれ以上の制限を行わない。

2.4.1 大型データの送信

大型 IPDU を送信するために、COM は PDUR システムサービスを使用して TP 送信を行う。

大型 IPDU を送信する場合のシーケンスは以下のとおり。

1. PduR_ComTransmit : 送信要求のための PDUR システムサービス
2. Com_CopyTxData : 送信データコピーのための COM コールバック関数(1 回以上)
3. Com_TpTxConfirmation : TP 送信完了通知のための COM コールバック関数

PduR_ComTransmit により大型 IPDU の送信要求を行った後、Com_TpTxConfirmation により TP 送信完了が通知されるまで、または送信中止が通知されるまでの間、COM はその大型 IPDU の内部バッファを更新しない [COM662]。

2.4.2 大型データの受信

大型 IPDU を受信するために、COM は PDUR システムサービスを使用して TP 受信を行う。

大型 IPDU を受信する場合のシーケンスは以下のとおり.

1. Com_StartOfReception : 受信要求のための COM コールバック関数
2. Com_CopyRxData : 受信データコピーのための COM コールバック関数(1 回以上)
3. Com_TpRxIndication : 大型 IPDU 受信完了通知

2.5 可変長データ

※可変長データ通信は本バージョン対象外の機能であるため、本章の内容は対象外となる。

可変長シグナルとは、実行時にデータの長さを変更することができるシグナルのことである。可変長シグナルの最大長は、コンフィギュレーション時に指定されなければならない。可変長シグナルを含む IPDU は可変長 IPDU と呼ばれ、可変長 IPDU の通信は TP を介して行われる。

可変長シグナルの型(*ComSignalType*) は `UINT8_DYN` でなければならない【COM753】。

【COM753】により、可変長シグナルの長さはバイト単位でしか指定できない。

可変長シグナル以外の全シグナル(大型シグナルも含む)は、可変長シグナルよりも先に IPDU へパッキングされる【COM754】【COM755】【COM756】。したがって、IPDU 内の可変長シグナルの長さはエンコードしない。受信側において、受信した可変長シグナルの長さは、受信した IPDU のデータの長さから算出することができる。

ユーザは、可変長シグナルにアップデートビットを設定することができる。この場合、アップデートビットは可変長シグナルのデータの前に配置する必要がある【COM755】。

あるシグナルグループ内に大型シグナルと可変長シグナル両方を含めることができる。

2.5.1 可変長データの送信

可変長 IPDU は PDUR システムサービスによって TP を介して送信される。

RTE は、送信する可変長シグナルの実際の長さを指定して `Com_SendDynSignal` を呼び出すことによって、可変長シグナルの送信要求を COM に渡す【COM627】。

`Com_SendDynSignal` の引数に指定された可変長シグナルの長さによって可変長 IPDU の長さが決められるため、可変長シグナルのデータの実際の長さと等しくなければならない【COM757】。

2.5.2 可変長データの受信

RTE は、可変長シグナルを受信する際に、その実際の長さを知る必要がある。

`Com_StartOfReception` によって可変長 I-PDU の受信が開始され、可変長 IPDU に含まれている可変長シグナルの長さが以下のように算出される(単位はバイト)【COM758】。

＜可変長シグナルの長さ＞

$$= \text{＜受信した IPDU の長さ＞} - \text{＜可変長シグナルのビット位置(ComBitPosition)＞} / 8$$

`Com_ReceiveDynSignal` は、【COM758】によって算出された可変長シグナルの実際の長さを返す。

2.6 アップデートビット

ユーザは、シグナルまたはシグナルグループのデータが更新されているかどうかを事前に識別できるようにするために、アップデートビットの指定が可能である。アップデートビットは IPDU バッファ内に含まれており、IPDU バッファにパッキングされたシグナルおよびシグナルグループのデータと共に PDUR へ送信される。

送信モードが DIRECT であり、かつ送信要求繰り返し回数(*ComTxModeNumberOfRepetitions*) が 1 以上である場合、アップデートビットはサポートされない [COM310]。

アップデートビットはシグナルまたはシグナルグループごとに指定可能であり、アップデートビット位置(*ComUpdateBitPosition*) に従って、シグナルまたはシグナルグループの所属 IPDU の内部バッファに配置される。

アップデートビットはシグナルまたはシグナルグループの内部バッファには配置されず、COM の内部のみで処理される【COM055】。

よって、AUTOSAR SW-C がアップデートビットに直接アクセスすることはできない。

COM はアップデートビットを以下の表 2-10 ように解釈する【COM059】。

表 2-10 アップデートビットの状態とその意味

状態	意味
0	アップデートビットはクリアされている。つまり、データは更新されていない。
1	アップデートビットはセットされている。つまり、データは更新されている。

初期化時にすべてのアップデートビットがクリアされる [COM117]。

アップデートビットを持つシグナルの DM については、「2.2.5 デッドラインモニタリング(DM)」を参照。

2.6.1 送信側のアップデートビット処理

Com_SendSignal によりシグナルの値が更新された場合、そのシグナルのアップデートビットはセットされ、Com_SendSignalGroup によりシグナルグループの値が更新された場合、そのシグナルグループのアップデートビットはセットされる【COM061】。

AUTOSAR 仕様では、Com_SendSignal によってシグナルが送信された場合、前回送信時の値と変化がなくても、アップデートビットをセットするかどうかの規定がない。しかし、受信 DM においてアップデートビットの値で受信タイムアウト判定を行うため、Com_SendSignal ではシグナルの値に関わらず一律アップデートビットをセットする【COMa003】。

アップデートビットクリアタイミング(*ComTxIPduClearUpdateBit*) が Transmit の場合、シグナルおよびシグナルグループの所属 IPDU が PduR_ComTransmit によって下位層に送信され、かつ E_OK

が返された後に、そのシグナルおよびシグナルグループのアップデートビットはすべてクリアされる【COM062】。

アップデートビットクリアタイミング(*ComTxIPduClearUpdateBit*) が Confirmation の場合、シグナルおよびシグナルグループの所属 IPDU が *PduR_ComTransmit* によって下位層に送信され、かつ *E_OK* が返された後、*Com_TxConfirmation* が呼び出された際に、そのシグナルおよびシグナルグループのアップデートビットはすべてクリアされる【COM577】。

送信モード(*ComTxModeMode*) が DIRECT であり、かつ送信要求繰り返し回数(*ComTxModeNumberOfRepetitions*) が 1 以上である IPDU の送信完了通知処理は、【COM305】で定義された通りに行われる。

アップデートビットクリアタイミング(*ComTxIPduClearUpdateBit*) が *TriggerTransmit* の場合、シグナルおよびシグナルグループの所属 IPDU が *Com_TriggerTransmit* によってトリガ送信された後に、そのシグナルおよびシグナルグループのアップデートビットはすべてクリアされる【COM578】。

2.6.2 受信側のアップデートビット処理

アップデートビットを持つシグナルまたはシグナルグループを受信した際に、そのアップデートビットがセットされている場合、そのシグナルまたはシグナルグループのデータは更新されたと判断される【COM067】。

受信したシグナルまたはシグナルグループにアップデートビットが指定されている場合、アップデートビットがセットされていれば受信処理(フィルタリング、通知、シグナルゲートウェイ、エンディアン変換)が行われ、アップデートビットがクリアされていれば受信したシグナルまたはシグナルグループは破棄される【COM324】。

アップデートビットを持つシグナルを受信した際に、そのアップデートビットがクリアされている場合、そのシグナルのデータは破棄され、シグナルゲートウェイも行われず。アップデートビットがクリアされているシグナルに対する受信処理は行われなため、シグナルの内部バッファも更新されない。よって、アップデートビットがクリアされているシグナルに対して RTE が受信要求を出した場合、そのシグナルの初期値または最後に受信した値が返される。

2.7 データシーケンス管理

※データシーケンス管理は本バージョン対象外の機能であるため、本章の内容は対象外となる。

ユーザは、IPDU カウンタによりデータシーケンス管理を行うことができる。IPDU カウンタは IPDU ごとに指定することができる [COM592_Conf]。

2.7.1 送信側のデータシーケンス管理

通信方向(*ComIPduDirection*) が SEND である IPDU が *IPDU カウンタ(ComIPduCounter)* を持つ場合、Com_Init または引数 Initialize が true に指定された Com_IpduGroupControl において、IPDU カウンタの値に 0 が設定される【COM687】。

PduR_ComTransmit によって IPDU が送信される前に、IPDU カウンタの値はインクリメントされる。ただし、PduR_ComTransmit の返り値が E_OK ではない場合、IPDU カウンタの値はデクリメントされる【COM688】。

2.7.2 受信側のデータシーケンス管理

通信方向(*ComIPduDirection*) が RECEIVE である IPDU が *IPDU カウンタ(ComIPduCounter)* を持つ場合、Com_Init または引数 Initialize が true に指定された Com_IpduGroupControl において初期化された後、IPDU カウンタの値に関わらず IPDU が受信される【COM587】。

IPDU カウンタ(ComIPduCounter) を持つ IPDU を受信した場合、その IPDU カウンタの期待値が設定される【COM588】。

例：IPDU カウンタのサイズが 4 ビットの場合

サイズが 4 ビットの IPDU カウンタは 0 から 15 までの値を持つことが出来るため、受信した IPDU の IPDU カウンタの値が 15 である場合、次の期待値は 0 である。受信した IPDU の IPDU カウンタ値が期待値と一致しているかどうかに関わらず、期待値は更新される。

IPDU カウンタ(ComIPduCounter) を持ち、かつ IPDU 複製が設定されていない IPDU を受信し、[COM588] に従って次の期待値が設定された後に、IPDU カウンタの値が以下のいずれかである場合、その IPDU は廃棄される【COM590】。

- ・ 受信した IPDU カウンタの値 < IPDU カウンタの期待値
- ・ 受信した IPDU カウンタの値 >

IPDU カウンタの期待値 + IPDU カウンタの閾値(*ComIPduCounterThreshold*)

受信した IPDU カウンタの値と期待値の不一致を検出した場合、*IPDU カウンタエラー通知(ComIPduCounterErrorNotification)* に指定されたコールバック関数を呼び出すことによって RTE へ通知される【COM727】 [COM726]。

2.8 通信保護

※通信保護は本バージョン対象外の機能であるため、本章の内容は対象外となる。

ユーザは、安全に関わる IPDU のデータの破損や損失を防ぐために、複数の LPDU に複製することができる。PDUR において IPDU の複製が行われ、COM において複製された IPDU の比較と判定が行われる。複製された IPDU は、比較のためのチェックサムまたはシグネチャを持っていないため、受信側で完全な IPDU の保存と比較を行う必要がある。

複製 IPDU (*ComIPduReplicaRef*) が参照する IPDU が複製される [COM601_Conf]。IPDU を複製する数は複製数 (*ComIPduReplicationQuorum*) に指定され、指定可能な値は 1 から 3 のいずれかである [COM600_Conf]。

PDUR は TP を介した通信における 1:n ルーティングおよび送信のための複数フレームはサポートしない。したがって、大型 IPDU および可変長 IPDU の複製はサポートしない。

受信側で行う通信保護判定は IPDU カウンタに依存するため、複製する IPDU が IPDU カウンタを持っていることを保証しなければならない。

2.8.1 送信側の通信保護

IPDU の複製自体は PDUR によって行われるため、COM の送信側の通信保護として行う処理は特にない。

2.8.2 受信側の通信保護

受信側では、複製された IPDU の比較と判定が行われ、正常に受信した IPDU に含まれるシグナルのみを通過させる。

複製される IPDU は IPDU カウンタを持たなければならない。ただし、その IPDU カウンタの閾値 (*ComIPduCounterThreshold*) は 0 でなければならない [COM595_Conf]。IPDU カウンタの処理は複製された IPDU それぞれに対して行われる。

複製される IPDU は、IPDU 種別 (*ComIPduType*) が NORMAL でなければならない。

複製数 (*ComIPduReplicationQuorum*) に指定された数の複製 IPDU がすべて同一な場合のみ、その IPDU に含まれるシグナルおよびシグナルグループを受信する [COM596]。

[COM596] でシグナルおよびシグナルグループを正常に受信した場合、その受信完了通知は、複製数 (*ComIPduReplicationQuorum*) に関わらず、一度のみ RTE へ通知される [COM597]。

IPDU は特別なチェックサムを持たないため、複製された IPDU と比較する場合はデータをバイトごとの比較によって実現されることが期待される。

2.8.3 制約事項

既存機能を再利用するアプローチとして PDUR において IPDU の複製を行うため、システムのセットアップは注意して行わなければならない。

PDUR としては、異なるバスでの IPDU 複製も可能である。COM の受信側では複製された IPDU は同じ ID となるが、Com_RxIndication はこれらの IPDU に対してリエントラントに呼び出してはならない。

これを防ぐための簡単な方法としては、すべての IPDU が同じインタフェースモジュールで受信されるようにすることにより、同じバスでの IPDU 複製とすることである。なお、これは AUTOSAR が推奨する方法であり、システムインテグレータは他の方法を選択することも可能である。

さらに、上位層からの IPDU の更新を防ぐ機能は存在しないため、通信保護としては、Com_TriggerTransmit による IPDU トリガ送信と組み合わせることはできない。

2.9 シグナルゲートウェイ

※シグナルゲートウェイは本バージョン対象外の機能であるため、本章の内容は対象外となる。

ユーザは、シグナルおよびシグナルグループに対して、1:n のシグナルゲートウェイを利用できる。シグナルゲートウェイでは、シグナルおよびシグナルグループの ID を識別するための定数を用いてルーティングを行う。ルーティング元とルーティング先はコンフィギュレーション時に設定される。

すべてのシグナルおよびシグナルグループにおいてシグナルゲートウェイが不要な場合、COM はシグナルゲートウェイの機能を無効化する【COM370】。

シグナルゲートウェイは COM に閉じた機能であり、周期タスク呼び出しを除き、他のモジュールからアクセスすることはできない。

シグナルゲートウェイのコンフィギュレーション情報はゲートウェイマッピング情報 (*ComGwMapping*) というコンテナに集約されている【COM544_Conf】。シグナルゲートウェイのルーティング先はシグナルおよびシグナルグループそれぞれに独立して設定できる。

ゲートウェイ先情報 (*ComGwDestinationDescription*) 内の GW シグナル (*ComGwSignal*) にルーティング先のシグナルまたはシグナルグループが指定される【COM549_Conf】【COM551_Conf】。

シグナルゲートウェイを持つシグナルまたはシグナルグループは、ルーティング先のシグナルまたはシグナルグループにコピーされる【COM377】。

なお、シグナルゲートウェイは大型 IPDU に含まれているシグナルおよびシグナルグループはサポートしない【COM598】。

2.9.1 シグナルに対するルーティング

シグナルゲートウェイを持つシグナルは、受信 IPDU から送信 IPDU にルーティングされる【COM357】。

受信したシグナルとルーティング先のシグナルのエンディアンが異なる場合、エンディアン変換が行われる【COM360】。

2.9.2 シグナルグループに対するルーティング

シグナルゲートウェイを持つシグナルグループは、受信 IPDU から送信 IPDU にルーティングされる【COM361】。

シグナルグループのデータの整合性を保つために、シグナルグループごとにルーティングされる【COM383】。

シグナルグループに含まれるグループシグナルの一部をまとめたシグナルグループへの転送は

整合性を保ってルーティングされる【COM735】.

受信したグループシグナルとルーティング先のシグナルのエンディアンが異なる場合、エンディアン変換が行われる【COM362】.

2.9.3 タイムアウト発生時のルーティング

シグナルゲートウェイを持つシグナルまたはシグナルグループに対して、受信タイムアウトが発生した場合でも、ルーティングは行われる【COM701】.

つまり、シグナルゲートウェイを持つシグナルまたはシグナルグループの受信が発生しない場合でも、ルーティングは行われる.

2.9.4 アップデートビット処理

受信したシグナルまたはシグナルグループがアップデートビットを持ち、かつ、ルーティング先のシグナルまたはシグナルグループもアップデートビットを持つ場合、アップデートビットがセットされているシグナルまたはシグナルグループに対してルーティングが行われ、ルーティング先のシグナルまたはシグナルグループが送信された後にアップデートビットがクリアされる【COM702】.

受信したシグナルまたはシグナルグループがアップデートビットを持ち、かつ、ルーティング先のシグナルまたはシグナルグループもアップデートビットを持つ場合、アップデートビットがクリアされているシグナルまたはシグナルグループに対するルーティングは行われない【COM703】.

受信したシグナルまたはシグナルグループはアップデートビットを持ち、かつ、ルーティング先のシグナルまたはシグナルグループはアップデートビットを持たない場合、アップデートビットがセットされているシグナルまたはシグナルグループに対してルーティングが行われる【COM704】.

受信したシグナルまたはシグナルグループはアップデートビットを持ち、かつ、ルーティング先のシグナルまたはシグナルグループはアップデートビットを持たない場合、アップデートビットがクリアされているシグナルまたはシグナルグループに対するルーティングは行われない【COM705】.

受信したシグナルまたはシグナルグループはアップデートビットを持たず、かつ、ルーティング先のシグナルまたはシグナルグループはアップデートビットを持つ場合、ルーティング後にアップデートビットはセットされ、送信後にアップデートビットはクリアされる【COM706】.

表 2-11 アップデートビットとルーティングの関係

ルーティング先 ルーティング元		アップデートビットを持つ	アップデートビットを持たない
アップデートビットを持たない		ルーティングされる.	ルーティングされる.
アップデートビットを 持つ	セット	ルーティング先のアップデート ビットはルーティング後にセッ トされ, 送信後にクリアされる.	
	クリア	ルーティングされない.	

2.9.5 シグナルゲートウェイの分離

IPDU 受信に使われる割込み処理にかかる時間を短縮するために, シグナルゲートウェイを割込み処理から分離する必要がある.

シグナルゲートウェイ処理はすべて Com_MainFunctionRouteSignals にて行われる【COM359】
 [COM400].

Com_MainFunctionRouteSignals では, 以下の順で送信プロパティおよび送信モードの判定も行われる【COM466】.

1. ルーティングされたシグナルを IPDU の内部バッファにコピーする
2. ルーティングされたシグナルの値から TMC 判定を行う
3. コピーした IPDU の TMS 判定を行う

Com_MainFunctionRouteSignals が 1 回呼び出されるごとに, 最多 1 回の IPDU 送信が行われる【COM539】.

2.10 エラー処理

COM は標準エラーを定義しないため、本章では拡張エラーに関してのみ記述する。

COM は、*DET* 使用有無 *ComConfigurationUseDet* に TRUE が設定された場合は拡張エラーを有効化し、FALSE が設定された場合は拡張エラーを無効化する【COM028】〔COM141_Conf〕。

2.10.1 エラー検出

COM における拡張エラーの検出は、プリコンパイル時に有効化および無効化の設定が可能である〔COM028〕〔COM141_Conf〕。

2.10.2 エラー通知

COM は、拡張エラーを検出した際に、DET システムサービスの *Det_ReportError* を呼び出して以下を通知することにより、拡張エラーを通知する【COM442】。

- COM のモジュール ID (50)
- インスタンス ID (0)
- 拡張エラーを検出した COM システムサービスの ID (*Com_ServiceIdType* 参照)
- エラーコード〔COM707〕

2.11 関連モジュール

COM は、他モジュールに向けて機能を提供し、また、他モジュールから提供される機能を利用する。COM と他モジュールとの関係を以下の図 2-3 他モジュールとの関係図 2-3 に示す。

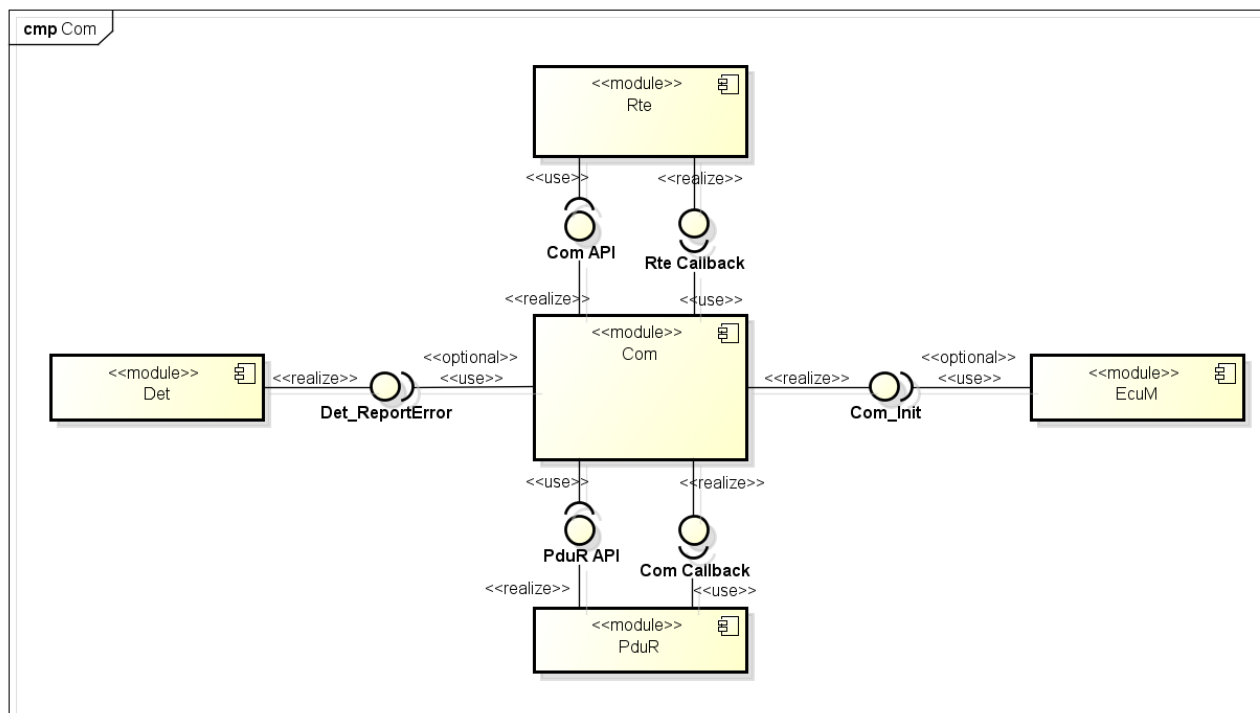


図 2-3 他モジュールとの関係

2.11.1 RTE

RTE は、COM の上位に位置するモジュールであり、シグナル送受信のために COM の機能を利用する。RTE の詳細は参考文書「次世代車載システム向け RTE 外部仕様書」を参照。

2.11.2 PduR

PduR は、COM の下位に位置するモジュールであり、IPDU の送受信のために COM の機能を利用する。PduR の詳細は参考文書「Specification of PDU Router」を参照。

PduR は COM に対して以下の機能を要求する。

- ・ IPDU の受信を通知するためのインタフェース(受信インディケーション)
- ・ IPDU の送信完了を通知するためのインタフェース(送信コンファメーション)
- ・ トリガ送信を引き起こすためのインタフェース
- ・ TP 通信のためのバッファリング

2.11.2.1 Com と PduR 間のインタフェース

COM の下位に位置するモジュールは PDUR のみである。COM は、通信プロトコル固有の TP を介

した通信を行うのか、通常の通信を行うのかが異なる 2 種類の IPDU の通信を PDUR システムサービスの利用により実現する。IPDU が TP を介した通信を行うかどうかは、*IPDU 種別(ComIPduType)* によって指定される。

COM は、PduR_ComTransmit を呼び出すことにより IPDU を送信する【COM138】。

IPDU 種別(ComIPduType) が NORMAL の場合、TP 通信を使用しないモジュールに向けて PDUR を使う【COM759】。

この場合、PDUR は以下の COM コールバック関数を使用する。

- Com_TxConfirmation
- Com_RxIndication
- Com_TriggerTransmit

IPDU 種別(ComIPduType) が TP の場合、TP 通信を使用するモジュールに向けて PDUR を使う【COM760】。

この場合、PDUR は以下の COM コールバック関数を使用する。

- Com_TpTxConfirmation
- Com_TpRxIndication
- Com_StartOfReception
- Com_CopyTxData
- Com_CopyRxData

表 2-12 IPDU 種別(ComIPduType) ごとの対応 API

API 名	<i>IPDU 種別(ComIPduType)</i>	
	NORMAL	TP
PduR_ComTransmit	○	○
Com_TxConfirmation	○	×
Com_RxIndication	○	×
Com_TriggerTransmit	○	×
Com_TpTxConfirmation	×	○
Com_TpRxIndication	×	○
Com_StartOfReception	×	○
Com_CopyTxData	×	○
Com_CopyRxData	×	○

2.12 COM 通信モデル

本章では、COM の通信機能のモデル図を記載する。モデル図にはすべての主要機能が記載されているが、実際に使用する際には、一部の必要な機能のみで構成される可能性がある。

ある受信シグナルに対してフィルタリングおよび無効値検出および通知が構成されている場合、これらの機能は以下の順序で実現される【COM396】。

1. 無効値検出
2. フィルタリング
3. 通知

シグナルまたはシグナルグループに、受信タイムアウト処理(*ComRxDataTimeoutAction*) および受信タイムアウト通知(*ComTimeoutNotification*) の両方が設定されている場合、これらの機能は以下の順序で実現される【COM744】。

1. 受信タイムアウト処理
2. 受信タイムアウト通知

シグナルバッファはシグナルおよびグループシグナルごとに所属しているバッファで IPDU と同期している。送信側ではエンディアン変換、アップデートビットセット前のデータが格納されている。受信側ではアップデートビット判定、エンディアン変換、符号拡張、フィルタリングが実施された結果が格納される。

シャドウバッファはグループシグナルごとに所属しているバッファで、シグナルグループに所属する全てのグループシグナルの時間的整合性を保証する。

シャドウバッファが存在しない場合、全てのグループシグナルを受信する前に *Com_RxIndication* または *Com_MainFunctionRx* によって IPDU バッファが更新されてしまった場合、シグナルグループに所属するグループシグナル間で時間的なズレが生じる。

グループシグナルの時間的整合性が乱れるケースを図 2-4 に示す。なお、時間的に乱れる順序は以下のとおり。

- グループシグナル A に対して *Com_ReceiveSignal* を発行して受信
- グループシグナル B に対して *Com_ReceiveSignal* を発行して受信
- *Com_RxIndication/Com_MainFunctionRx* によって IPDU バッファ更新
- グループシグナル C に対して *Com_ReceiveSignal* を発行して受信

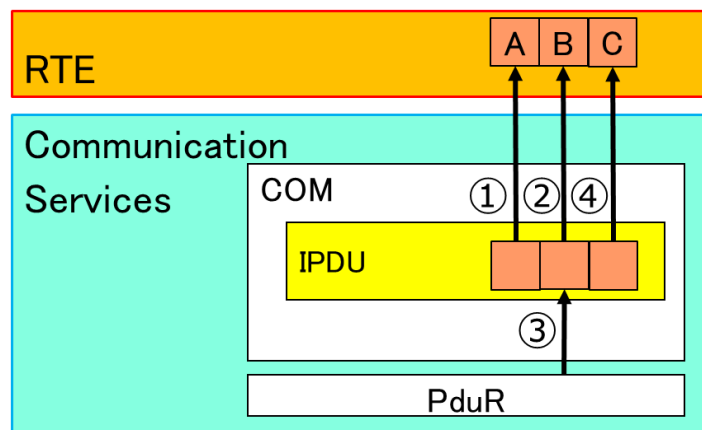


図 2-4 時間的整合性が乱れる例

2.12.1 COM 送信モデル

COM 送信モデル図を以下の図 2-5 に示す。

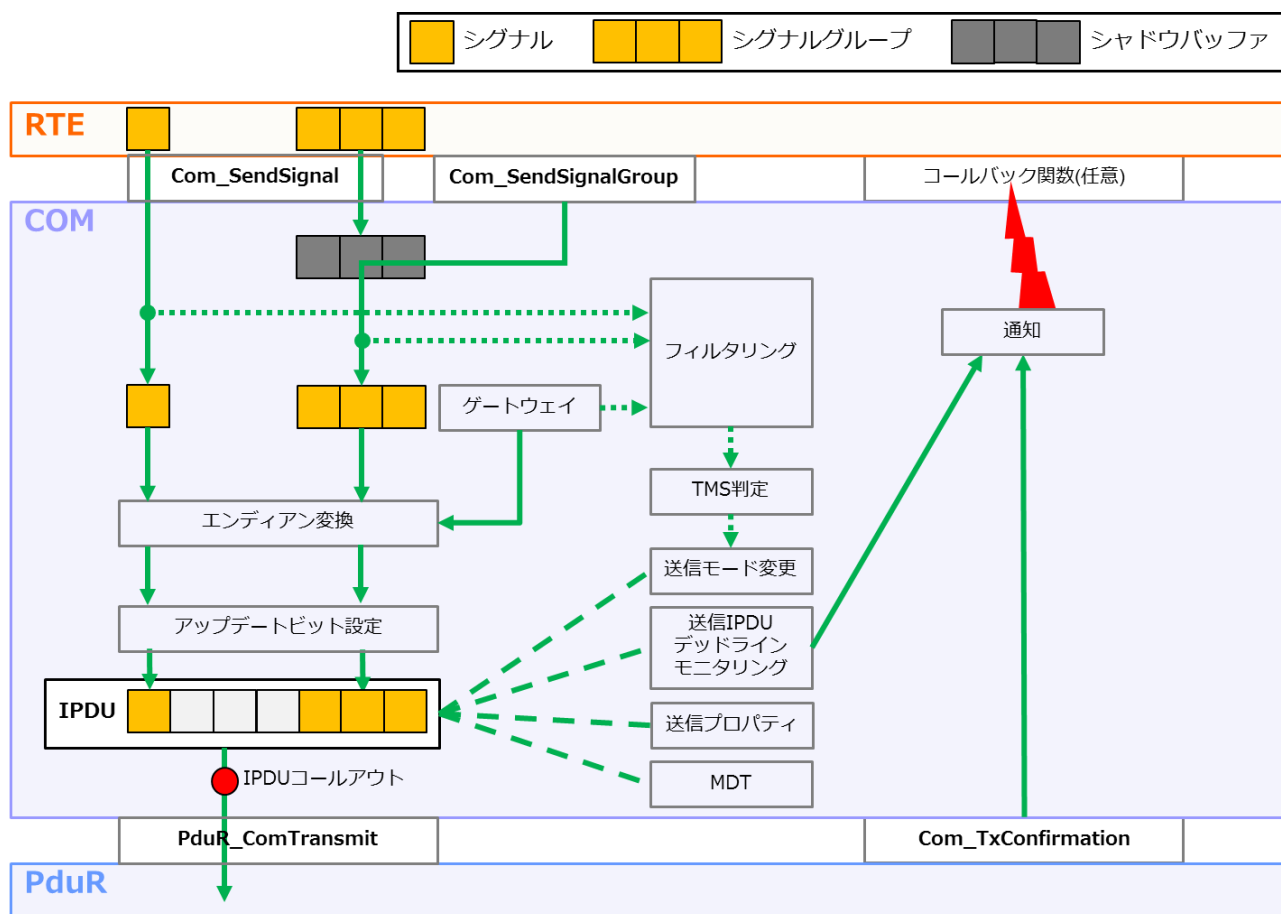


図 2-5 COM 送信モデル図

COM はシグナルに対して **Com_SendSignal** を発行した場合、シグナルバッファの更新、エンディアン変換、アップデートビットのセットを行って、IPDU バッファにパッキングを行う。

COM はグループシグナルに対して **Com_SendSignal** を発行した場合、シャドウバッファの更新のみを行い、シグナルグループに対して **Com_SendSignalGroup** を発行した場合、シグナルグループに所属しているグループシグナルのシャドウバッファのデータでシグナルバッファの更新、エンディアン変換、アップデートビットのセットを行って、IPDU バッファにパッキングを行う。

2.12.2 COM 受信モデル

COM 受信モデルを以下の図 2-6 に示す。

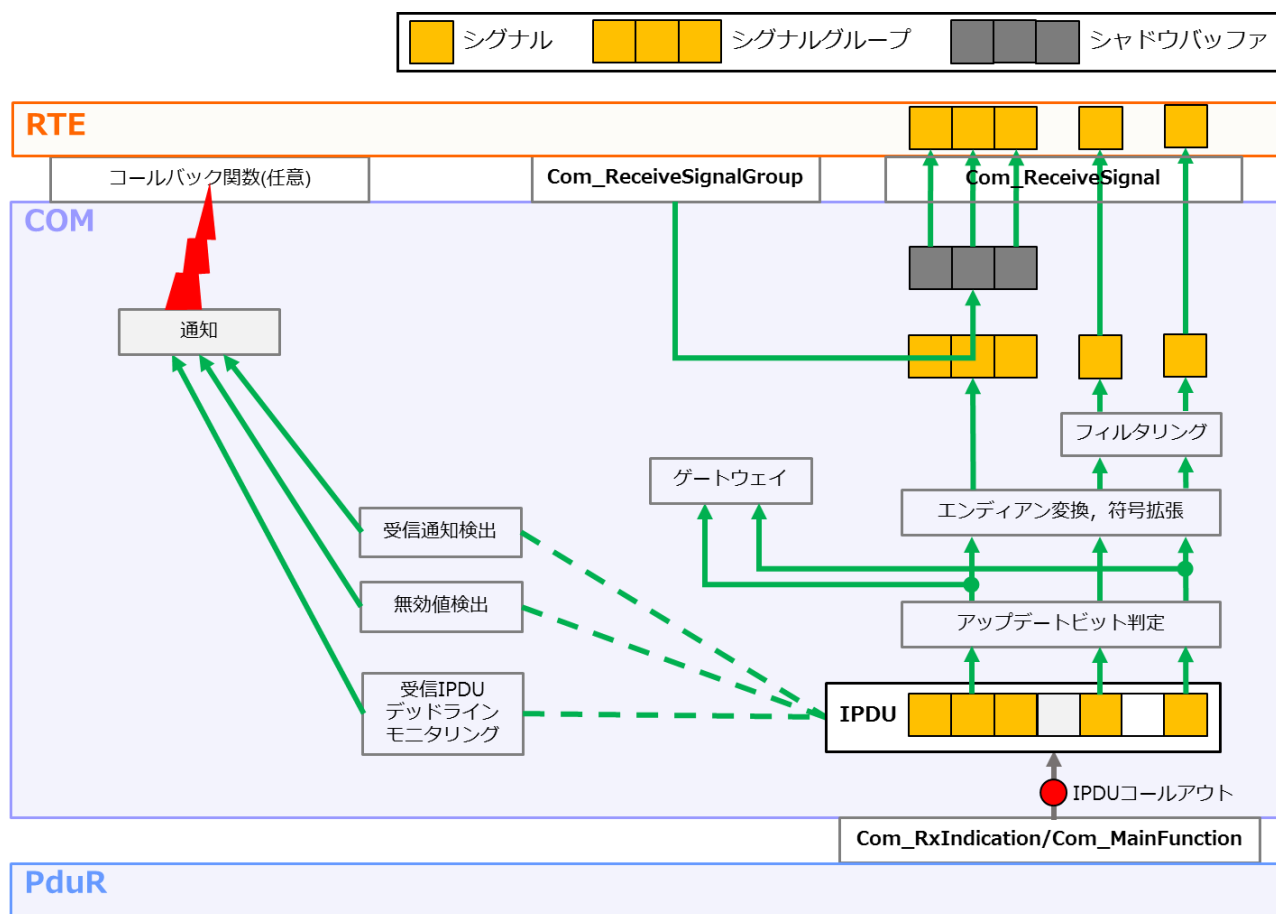


図 2-6 COM 受信モデル図

COM は Com_RxIndication または Com_MainFunctionRx によって IPDU バッファの更新、アップデートビット判定、エンディアン変換、符号拡張、フィルタリングを行い、シグナルバッファを更新する。

COM はシグナルに対して Com_ReceiveSignal を発行した場合、シグナルバッファに格納されているデータを取得する。

COM はシグナルグループに対して Com_ReceiveSignalGroup を発行した場合、シグナルバッファに格納されているデータをシャドウバッファに格納し、グループシグナルに対して Com_ReceiveSignal を発行した場合、シャドウバッファに格納されているデータを取得する。

2.13 ファイル構成

本文書ではファイルの構成を完全には定義しない。本章では、ファイル構成に含まれる必要のあるファイルの名称と概要について記述する。

COM のコードファイルの構成を以下に示す。

- Com.c : COM の API の実態を含む【COM584】。
- Com_Cfg.c : プレコンパイル時に設定可能なパラメータを含む【COM585】。
- Com_Lcfg.c : リンク時に設定可能なパラメータを含む【COM583】。
- Com_PBcfg.c : ポストビルド時に設定可能なパラメータを含む【COM583】。

COM のヘッダファイルの構成を以下に示す。

- Com.h : COM の API の定義を含む【COM005】。
: 外部モジュールへ公開するパラメータを含む【COM786】。
- Com_Cbk.h : COM のコールバック関数とコールアウト関数を含む【COM731】。

COM に関連するファイルは、図 2-7 に示すようなインクルード構成とする【COM220】。

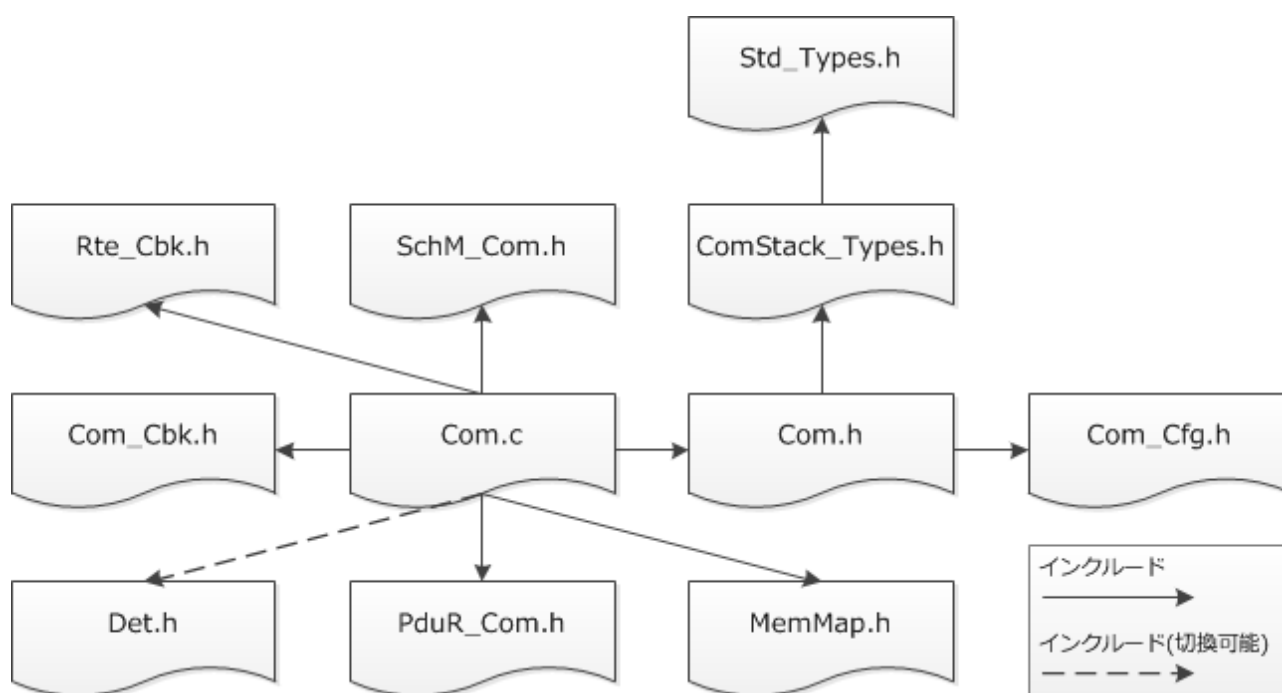


図 2-7 インクルードファイル構成

AUTOSAR 仕様との違い

AUTOSAR 仕様では、PDU の ID や長さを示すデータ型のサイズを、COM スタックジェネレータで生成し、ComStack_Cfg.h に定義すると規定されている [EcuC041_Conf] [EcuC042_Conf] [COMTYPE029]。しかし、これらはジェネレータで生成するほどの内容で無いため、本仕様では、ComStack_Cfg.h は使用せず、ComStack_Types.h に PDU 関連のデータ型を直接定義する【NCOM025】。

2.14 コンフィギュレーション

AUTOSAR アーキテクチャの全体像については、参考文書「AUTOSAR Layered Architecture」および参考文書「Specification of ECU Configuration」を参照。

コンフィギュレーション情報は XML 形式で記述されなければならない【COM006】。

すべてのコンフィギュレーションセットは ID によって識別される【COM374】[COM394_Conf]。

COM ジェネレータで生成するコンフィギュレーションデータは、"com_config"という変数名の配列で、ComConfig コンテナごとに定義する【NCOM024】。

したがって、Com_Init では"**Com_Init(&com_config[0]);**"のように指定する。添字はコンフィギュレーション ID (*ComConfigurationId*) で指定した値となる。

2.14.1 コンフィギュレーションクラス

各パラメータは以下のいずれかのコンフィギュレーションクラスを持つ。

- VARIANT-PRE-COMPILE
- VARIANT-LINK-TIME
- VARIANT-POST-BUILD

VARIANT-PRE-COMPILE はプレコンパイルタイムにコンフィギュレーション可能なパラメータであることを示し、#define されるなど、多くの場合はソースコードとして提供される【COM606】。

VARIANT-LINK-TIME はリンクタイムまたはプレコンパイルタイムにコンフィギュレーション可能なパラメータであることを示し、多くの場合はオブジェクトコードとして提供される【COM607】。

VARIANT-POST-BUILD はポストビルドタイムまたはリンクタイムまたはプレコンパイルタイムにコンフィギュレーション可能なパラメータであることを示し、多くの場合はオブジェクトコードとして提供される【COM608】。

ポストビルドタイムにコンフィギュレーション可能なパラメータは、そのパラメータが使用されていないタイミングに更新される【COM373】。

ポストビルドタイムのコンフィギュレーションされるコンフィギュレーション情報は、一意な ID で識別される【COM487】。

2.14.2 コンフィギュレーションルール

コンフィギュレーションルールに反するディスクリプションファイルを入力された場合、COM ジェネレータはエラーを検出する【NCOM007】。

2.14.2.1 全般

以下のパラメータのショートネームは重複してはならない【COM401】。

- ComSignal
- ComSignalGroup
- ComGroupSignal
- ComIPdu
- ComIPduGroup

以下のパラメータに指定する文字列は、【COM401】に挙げられたパラメータのショートネームとは異なる文字列でなければならない【COM732】。

- ComNotification
- ComErrorNotification
- ComTimeoutNotification
- ComInvalidNotification
- ComIPduCallout

以下は C 言語の識別子ルールに従って記述されなければならない【COM402】。

- ComSignal のショートネーム
- ComSignalGroup のショートネーム
- ComGroupSignal のショートネーム
- ComIPdu のショートネーム
- ComIPduGroup のショートネーム
- ComNotification の設定値
- ComErrorNotification の設定値
- ComTimeoutNotification の設定値
- ComInvalidNotification の設定値
- ComIPduCallout の設定値

以下のパラメータは 0 から連番で指定されなければならない【NCOM008】。

- ComConfigurationId
- ComIPduHandleId
- ComIPduGroupHandleId
- ComHandleId

【NCOM008】により、処理を高速化することができる。なお、ComConfig が複数定義されている場合、ComConfig ごとに 0 から連番とする。各 ID はコンフィギュレーションツールにより自動付番される想定であるため、この制約が問題になるとは考えにくい。

以下の数値変換できないパラメータが指定された場合、エラーとする【NCOM104】。

- ・ 10 進数、16 進数(0x)以外の記法で書かれた数値
- ・ 「aaa」のように数値と関係のない文字列
- ・ 空文字列
- ・ 32bit で表現できない桁数の数値

2.14.2.2 シグナル

シグナルおよびグループシグナルは複数の IPDU に所属してはならない【COM101】。

シグナルおよびグループシグナルのデータはオーバーラップしてはならない【COM102】。

ユーザは、送信側および受信側のシグナル無効値(*ComSignalDataInvalidValue*)を一致させなければならない。シグナル無効値(*ComSignalDataInvalidValue*)にアプリケーション内で有効となる値を指定してはならない。

シグナルのサイズ(*ComBitSize*)は、そのシグナルの型(*ComSignalType*)が取り得る最大値を超えてはならない【COM785】。

シグナルおよびグループシグナルの型(*ComSignalType*)が `UINT8_N` または `UINT8_DYN` である場合、そのサイズは N バイトでなければならない【COM443】。

シグナルおよびグループシグナルの型(*ComSignalType*)が `UINT8_N` または `UINT8_DYN` である場合、そのエンディアン(*ComSignalEndianness*)は `OPAQUE` でなければならない【COM553】。

ある可変長 IPDU に所属している可変長シグナルの数は最大 1 つでなければならない【COM754】。

可変長シグナルは可変長 IPDU 内の一番後ろに配置されなければならない【COM755】。

可変長シグナルの長さ(*ComSignalLength*)には、取り得る最大の長さが指定されなければならない【COM756】。

送信モード(*ComTxModeMode*)が `DIRECT` または `MIXED` であり、かつ送信要求繰り返し回数(*ComTxModeNumberOfRepetitions*)が 1 以上である場合、アップデートビットが指定されてはならない【COM310】

シグナルの初期値(*ComSignalInitValues*)および無効値(*ComSignalDataInvalidValue*)に指定された値は、整数型およびブーリアン型として解釈される【COM790】。

シグナルの型(*ComSignalType*)が `UINT8_N` の場合、その初期値(*ComSignalInitValues*)および無効値(*ComSignalDataInvalidValue*)に指定された値は、空白で区切られた 10 進の ASCII コード表記

の文字列として解釈される【COM791】。例えば，“97 98 99”と指定した場合，“abc”と解釈される。

シグナルの型(*ComSignalType*)が FLOAT32 または FLOAT64 の場合、そのシグナルはフィルタを持ってはならない【COM319】。よって、シグナルの型(*ComSignalType*)が FLOAT32 または FLOAT64 の場合、そのシグナルの値は TMS 判定には用いられない。

2.14.2.3 シグナルグループ

シグナルゲートウェイを持つシグナルグループの場合、ルーティング元およびルーティング先に対して同じ型が指定されなければならない【COM365】。

シグナルおよびシグナルグループの IPDU への配置に関して特に制約は無く、シグナルとシグナルグループを交互に配置しても良い。

2.14.2.4 IPDU

送信 IPDU(*ComTxIPdu*)は TMS 判定の結果により選択され得る送信モードのコンフィギュレーション情報(*ComTxModeTrue* または *ComTxModeFalse*)を持たなければならない【COM465】。

アップデートビットを持つシグナルが所属している送信 IPDU のアップデートビットクリアタイミング(*ComTxIPduClearUpdateBit*)が省略されている場合、COM ジェネレータによってデフォルト値の Transmit で補完される【NCOM030】。

2.14.2.5 シグナルゲートウェイ

ルーティング元およびルーティング先のシグナルのサイズ(*ComBitSize*)は等しくなければならない【COM384】。

最適化の問題：ルーティング元およびルーティング先のシグナルの配置順序およびエンディアンが等しい場合、ルーティング先の所属 IPDU ごとにルーティングできる【COM386】。

IPDU 種別(*ComIPduType*)が TP の場合、その IPDU に含まれるシグナルおよびシグナルグループは、シグナルゲートウェイ(*ComGwMapping*)を持ってはならない【COM598】。

2.14.2.6 フィルタ

フィルタアルゴリズム(*ComFilterAlgorithm*)が ONE_EVERY_N の場合、フィルタオフセット(*ComFilterOffset*)にはフィルタ周期(*ComFilterPeriod*)よりも小さい値が指定されなければならない【COM535】。

フィルタアルゴリズム(*ComFilterAlgorithm*)が NEW_IS_OUTSIDE または NEW_IS_WITHIN の場合、そのフィルタを持つシグナルの型(*ComSignalType*)の範囲内の値をフィルタ最小値(*ComFilterMin*)およびフィルタ最大値(*ComFilterMax*)に指定しなければならない【NCOM045】。

フィルタアルゴリズム(*ComFilterAlgorithm*)が NEW_IS_OUTSIDE または NEW_IS_WITHIN の場合、そのフィルタのフィルタ最小値(*ComFilterMin*)はフィルタ最大値(*ComFilterMax*)よりも小さい値が指定されなければならない【NCOM046】。

2.14.2.7 周期関数呼び出し回数

送信周期や DM のタイムアウト時間は秒単位で指定される。COM においては、一定間隔ごとに呼び出される周期関数の呼び出し回数によって、これらの時間の計測を行う。また、周期関数の呼び出し間隔も秒単位で指定される。

本仕様では、各パラメータに指定された秒を使用して以下の方法で周期関数の呼び出し回数の算出を行う【NCOM005】。

1. 対象時間が周期関数呼び出し間隔時間で割り切れる場合

$$\text{周期関数呼び出し回数} = \text{対象時間} / \text{周期関数呼び出し間隔時間}$$

(例) ComTxModeTimePeriod: 5.0, ComTxTimeBase: 1.0 の場合,
MF 呼び出し回数は 5 回

2. 対象時間が MF 呼び出し間隔時間で割り切れない場合

$$\text{周期関数呼び出し回数} = (\text{対象時間} / \text{周期関数呼び出し間隔時間}) + 1$$

(例) ComTxModeTimePeriod: 5.0, ComTxTimeBase: 2.0 の場合,
MF 呼び出し回数は 3 回 (切り上げた(ceil)整数値)

2.15 本バージョン対象機能

本章では，通信プロトコルを CAN のみに限定する．

本バージョン対象の COM 機能を以下の表 2-13 に示す．

表 2-13 本バージョン対象機能

分類	機能名	対象
通信 管理	初期化	○
	未初期化	○
	IPDU グループ管理	○
	シグナル無効化	○
	データシーケンス管理	×
	通信保護	×
	DET エラー通知	○
送信	送信処理	○
	送信モード	○
	MDT	○
	送信 DM	○
	失敗送信要求リトライ	○
	下位層からのトリガ送信	×
受信	受信処理	○
	フィルタリング	○
	受信 DM	○
通信 形式	シグナル通信	○
	シグナルグループ通信	○
	可変長データ通信	×
	大型データ通信	×
	TP 通信	×
	シグナルゲートウェイ	×

本バージョン対象の COM システムサービスを以下の表に示す.

表 2-14 本バージョン対象システムサービス

システムサービス名	対象
Com_Init	○
Com_DeInit	○
Com_GetStatus	○
Com_GetConfigurationId	○
Com_GetVersionInfo	○
Com_IpduGroupControl	○
Com_ReceptionDMControl	○
Com_SetIpduGroup	○
Com_ClearIpduGroupVector	○
Com_SendSignal	○
Com_SendDynSignal	×
Com_SendSignalGroup	○
Com_ReceiveSignal	○
Com_ReceiveDynSignal	×
Com_ReceiveSignalGroup	○
Com_InvalidateSignal	○
Com_InvalidateSignalGroup	○
Com_TriggerIPDUSend	○
Com_SwitchIpduTxMode	○

3 API 仕様

3.1 API 実装方針

3.1.1 実装言語

システムサービスのインタフェースは ISO/ANSI-C に基づいて定義する【CCOM2901】.

通常はシステムサービスを関数として実装するが、必要に応じて C 言語のプリプロセッサマクロで実装しても良い【COM434】.

3.1.2 再入可能性

あるシステムサービスが再入不可能である場合、そのシステムサービスの呼び出し側は、そのシステムサービスが実行されている間に再度呼び出さないことを保証しなければならない【COM320】.

再入不可能なシステムサービスは、再入可能なシステムサービスから呼ばれているかどうかを確認する必要はない【COM321】.

AUTOSAR 仕様との違い

COM システムサービスには再入不可能と規定されているものがあるが、これは RTE 側で排他することを想定しているものである. 本仕様では、直接システムサービスを使用された場合のデータ整合性を考慮し、すべてのシステムサービスを再入可能として実装するものとする.

排他エリア用の API を RTE から提供されることを前提として、排他エリアのショートネームを <Reentrant>とした場合、COM の実装上では以下の API を記述する【NCOM032】.

- SchM_Enter_Com_<Reentrant>()
- SchM_Exit_Com_<Reentrant>()

なお、ショートネーム<Reentrant>の命名規則は実装依存とする. ネスト呼出しされる可能性があるシステムサービスに対しては、通常の排他エリアとは異なる排他エリアを用意する必要がある.

3.1.3 引数チェック

COM のシステムサービスの引数で渡された入力パラメータの妥当性を確認しなければならない. なお、この引数チェックは開発中のソースコードでのみ有効となり、製品には含まれない【COM024】.

3.1.4 デバッグ

デバッグの際に参照する必要がある変数をグローバル変数として定義する【COM745】.

デバッグの際に参照する必要がある変数の型を Com.h に定義する【COM746】.

Com.h 内で定義された型のサイズは C 言語の sizeof により取得可能とする【COM747】.

デバッグの際に参照する必要がある変数については、本文書内に記載される【COM748】.

3.2 データ型

3.2.1 外部からインポートするデータ型

COM は、表 3-1 に示すデータ型をインポートして使用する【COM609】。

表 3-1 インポートするデータ型

ファイル名	データ型
ComStack_Types.h	BufReq_ReturnType
	NotifResultType
	ComStack_Types
	PduIdType
	PduInfoType
	PduLengthType
	RetryInfoType
Dem.h	Dem_EventIdType
	Dem_EventStatusType
Std_Types.h	Std_ReturnType
	Std_VersionInfoType

3.2.2 内部で定義するデータ型

3.2.2.1 Com_StatusType

データ型名	Com_StatusType【COMa011】	
型	列挙型	
値の範囲	COM_UNINIT	COM は初期化されておらず、使用不可能な状態。 リセット後のデフォルト値であり、値は 0 となる。
	COM_INIT	COM は初期化されており、使用可能な状態。
概要	COM 状態型。Com_GetStatus の返り値に利用される。	

3.2.2.2 Com_SignalIdType

データ型名	Com_SignalIdType【COMa012】	
型	uint16	
値の範囲	0..<シグナル ID の最大値>	0 から連番となる整数値。
概要	シグナル ID 型。	

3.2.2.3 Com_SignalGroupIdType

データ型名	Com_SignalGroupIdType【COMa013】	
型	uint16	

値の範囲	0..<シグナルグループ ID の最大値>	0 から連番となる整数値.
概要	シグナルグループ ID 型.	

3.2.2.4 Com_IpduGroupIdType

データ型名	Com_IpduGroupIdType 【COMa014】	
型	uint16	
値の範囲	0..<IPDU グループ ID の最大値>	0 から連番となる整数値. <IPDU グループ ID の最大値>は, <i>ComSupportedIPduGroups</i> [COM710_Conf] に指定した値よりも小さくなくてはならない.
概要	IPDU グループ ID 型.	

3.2.2.5 Com_IpduGroupVector

データ型名	Com_IpduGroupVector 【COMa015】	
型	uint8[(<i>ComSupportedIPduGroups</i> - 1) / 8 + 1]	
値の範囲	ビットフィールド	
概要	IPDU グループベクタ型. IPDU グループの起動状態とデッドラインモニタリング状態を指定するために使われる.	

3.2.2.6 Com_ServiceIdType

データ型名	Com_ServiceIdType 【COMa016】	
型	uint8	
値の範囲	COMServiceId_Init	0x01
	COMServiceId_DeInit	0x02
	COMServiceId_IpduGroupControl	0x03
	COMServiceId_ReceptionDMControl	0x06
	COMServiceId_GetStatus	0x07
	COMServiceId_GetConfigurationId	0x08
	COMServiceId_GetVersionInfo	0x09
	COMServiceId_SendSignal	0x0A
	COMServiceId_ReceiveSignal	0x0B
	COMServiceId_SendSignalGroup	0x0D
	COMServiceId_ReceiveSignalGroup	0x0E
	COMServiceId_InvalidateSignal	0x10
	COMServiceId_TriggerIPDUSend	0x17
	COMServiceId_MainFunctionRx	0x18

	COMServiceId_MainFunctionTx	0x19
	COMServiceId_MainFunctionRouteSignals	0x1A
	COMServiceId_InvalidateSignalGroup	0x1B
	COMServiceId_ClearIpduGroupVector	0x1C
	COMServiceId_SetIpduGroup	0x1D
	COMServiceId_TpRxIndication	0x1E
	COMServiceId_SendDynSignal	0x21
	COMServiceId_ReceiveDynSignal	0x22
	COMServiceId_CopyRxData	0x23
	COMServiceId_CopyTxData	0x24
	COMServiceId_StartOfReception	0x25
	COMServiceId_TpTxConfirmation	0x26
	COMServiceId_SwitchIpduTxMode	0x27
	COMServiceId_TxConfirmation	0x40
	COMServiceId_TriggerTransmit	0x41
	COMServiceId_RxIndication	0x42
概要	システムサービス ID 型.	

3.2.2.7 Com_ConfigType

データ型名	Com_ConfigType 【COMa017】
型	構造体型
値の範囲	実装仕様
概要	コンフィギュレーションデータ型.

3.3 定数

3.3.1 IPDU 通信方向設定

SEND

定数名	SEND
概要	送信側であることを示す定数

RECEIVE

定数名	RECEIVE
概要	受信側であることを示す定数

3.3.2 IPDU 種別設定

NORMAL

定数名	NORMAL
概要	通常通信を行う IPDU であることを示す定数

TP

定数名	TP
概要	TP 通信を行う IPDU であることを示す定数

3.3.3 シグナル型設定

BOOLEAN

定数名	BOOLEAN
概要	ブーリアン型であることを示す定数

UINT8

定数名	UINT8
概要	8 ビットの符号なし整数型であることを示す定数

UINT16

定数名	UINT16
概要	16 ビットの符号なし整数型であることを示す定数

UINT32

定数名	UINT32
概要	32 ビットの符号なし整数型であることを示す定数

SINT8

定数名	SINT8
概要	8 ビットの符号あり整数型であることを示す定数

SINT16

定数名	SINT16
概要	16 ビットの符号あり整数型であることを示す定数

SINT32

定数名	SINT32
概要	32 ビットの符号あり整数型であることを示す定数

UINT8_N

定数名	UINT8_N
概要	8 ビットの符号なし整数の配列型であることを示す定数

UINT8_DYN

定数名	UINT8_DYN
概要	8 ビットの符号なし整数の可変長配列型であることを示す定数

FLOAT32

定数名	FLOAT32
概要	32 ビットの浮動小数点型であることを示す定数

FLOAT64

定数名	FLOAT64
概要	64 ビットの浮動小数点型であることを示す定数

3.3.4 エンディアン設定

BIG_ENDIAN

定数名	BIG_ENDIAN
概要	ビッグエンディアンであることを示す定数

LITTLE_ENDIAN

定数名	LITTLE_ENDIAN
概要	リトルエンディアンであることを示す定数

OPAQUE

定数名	OPAQUE
概要	オpaqueであることを示す定数

3.3.5 フィルタアルゴリズム設定

ALWAYS

定数名	ALWAYS
概要	常に通過するフィルタであることを示す定数

NEVER

定数名	NEVER
概要	常に通過しないフィルタであることを示す定数

MASKED_NEW_DIFFERS_MASKED_OLD

定数名	MASKED_NEW_DIFFERS_MASKED_OLD
概要	マスクした今回値とマスクした前回値を比較して、値が異なる場合のみ通過するフィルタであることを示す定数

MASKED_NEW_DIFFERS_X

定数名	MASKED_NEW_DIFFERS_X
概要	マスクした今回値と X の値を比較して、値が異なる場合のみ通過するフィルタであることを示す定数

MASKED_NEW_EQUALS_X

定数名	MASKED_NEW_EQUALS_X
概要	マスクした今回値と X の値を比較して、値が一致する場合のみ通過するフィルタであることを示す定数

NEW_IS_OUTSIDE

定数名	NEW_IS_OUTSIDE
概要	今回値が指定された範囲外の値である場合のみ通過するフィルタであることを示す定数

NEW_IS_WITHIN

定数名	NEW_IS_WITHIN
-----	---------------

概要	今回値が指定された範囲内の値である場合のみ通過するフィルタであることを示す定数
----	---

ONE EVERY N

定数名	ONE EVERY_N
概要	N 回フィルタリングされる度に 1 回のみ通過するフィルタであることを示す定数

3.3.6 送信モード設定

DIRECT

定数名	DIRECT
概要	直接(N 回)送信であることを示す定数

PERIODIC

定数名	PERIODIC
概要	周期送信であることを示す定数

MIXED

定数名	MIXED
概要	直接(N 回)送信と周期送信を並行して行うことを示す定数

3.3.7 送信プロパティ設定

PENDING

定数名	PENDING
概要	送信トリガではないことを示す定数

TRIGGERED

定数名	TRIGGERED
概要	送信トリガであることを示す定数

TRIGGERED ON CHANGE

定数名	TRIGGERED_ON_CHANGE
概要	値の変更がある場合のみ送信トリガとなることを示す定数

TRIGGERED ON CHANGE WITHOUT REPETITION

定数名	TRIGGERED_ON_CHANGE_WITHOUT_REPETITION
-----	--

概要	値の変更がある場合のみ繰り返しなしの送信トリガとなることを示す定数
----	-----------------------------------

TRIGGERED WITHOUT REPETITION

定数名	TRIGGERED_WITHOUT_REPETITION
概要	繰り返しなしの送信トリガとなることを示す定数

3.3.8 通信完了通知タイミング設定

DEFERRED

定数名	DEFERRED
概要	通信完了通知を周期関数で通知することを示す定数

IMMEDIATE

定数名	IMMEDIATE
概要	通信完了通知を即時に通知することを示す定数

3.3.9 アップデートビットクリアタイミング設定

Confirmation

定数名	Confirmation
概要	送信完了通知を受け取った際にアップデートビットをクリアすることを示す定数

Transmit

定数名	Transmit
概要	PduR_ComTransmit を呼び出した際にアップデートビットをクリアすることを示す定数

TriggerTransmit

定数名	TriggerTransmit
概要	Com_TriggerTransmit により IPDU がフェッチされた際にアップデートビットをクリアすることを示す定数

3.3.10 エラー発生時処理設定

NOTIFY

定数名	NOTIFY
概要	コールバック関数を呼び出して RTE へ通知することを示す定数

REPLACE

定数名	REPLACE
概要	バッファの値を初期値に置き換えることを示す定数

NONE

定数名	NONE
概要	特に何もしないことを示す定数

3.3.11 システムサービス返り値

COM_SERVICE_NOT_AVAILABLE

定数名	COM_SERVICE_NOT_AVAILABLE
概要	システムサービスが利用不可であることを示す定数

COM_BUSY

定数名	COM_BUSY
概要	リソース使用中によるシステムサービス異常終了であることを示す定数

3.3.12 無効値

COM_INVALID_UINT8

定数名	COM_INVALID_UINT8
概要	8 ビットの符号なし整数型のための無効値(0xFF)

COM_INVALID_UINT16

定数名	COM_INVALID_UINT16
概要	16 ビットの符号なし整数型のための無効値(0xFFFF)

COM_INVALID_UINT32

定数名	COM_INVALID_UINT32
概要	32 ビットの符号なし整数型のための無効値(0xFFFFFFFF)

3.4 返り値

COM はシステムサービスの返り値のためのデータ型を定義しない。COM システムサービスは Std_ReturnType の値、または、エラー通知のための定数を返す。

COM システムサービスの返り値を以下の表 3-2 に示す【COM459】。

表 3-2 COM システムサービスの返り値一覧

返り値名	値	ファイル名	概要
E_OK	0x00	Std_Types.h	システムサービス正常終了
E_NOT_OK	0x01	Std_Types.h	システムサービス異常終了
COM_SERVICE_NOT_AVAILABLE	0x80	Com.h	システムサービス利用不可
COM_BUSY	0x81	Com.h	リソース使用中によるシステムサービス異常終了

3.5 エラーコード

3.5.1 標準エラー

COM は標準エラーを定義しない。そのため、COM は DEM へのエラー通知を行わない。ただし、今後のバージョンにおいて標準エラーを定義した場合、COM は DEM へのエラー通知を行わなければならない。

3.5.2 拡張エラー

COM は以下の表 3-3 で示す拡張エラーを定義する【COM707】。

表 3-3 拡張エラーコード一覧

エラーコード	値	概要
COM_E_PARAM	0x01	不正な値を引数に与えて COM システムサービスを呼び出した
COM_E_UNINIT	0x02	COM が未初期化状態(Com_Init 呼び出し前または Com_Deinit 呼び出し後)の間に COM システムサービスを呼び出した
COM_E_PARAM_POINTER	0x03	NULL ポインタを引数に与えて COM システムサービスを呼び出した

3.6 システムサービス

3.6.1 Com_Init

サービス ID		0x01	
C 言語 I/F		void Com_Init(const Com_ConfigType* config) 【COM432】	
パラメータ	[in]	config	コンフィギュレーション情報へのポインタ
	[in/out]	－	
	[out]	－	
返り値		－	
DET エラー		COM_E_PARAM	引数 config が不正 〔COM433〕〔NCOM056〕
			既に初期化されている 〔COM373〕
同期/非同期		同期	
再入可能/再入不可能		再入可能	
機能		本システムサービスは、COM の初期化を行う。	

機能仕様

本システムサービスは、引数 config に NULL ポインタが指定された場合、コンフィギュレーション ID(ComConfigurationId) が 0 のコンフィギュレーション情報をデフォルトとして使用する【NCOM040】。

引数に指定された config が、設定されたコンフィギュレーション情報以外が指定された場合を不正とする【NCOM056】。

本システムサービスは、引数 config に与えられたコンフィギュレーション情報が不正であり、かつ、DET エラーが有効である場合に、DET モジュールの Det_ReportError を利用してエラーコード COM_E_PARAM を報告する【COM433】【COM024】。

ポストビルドタイムパラメータは、未使用時(つまり、未初期化時)にのみ更新できる【COM373】。

本システムサービスの詳細な機能に関しては、「2.2.2 初期化処理」を参照。

設計上の注意

本システムサービスは、他の COM システムサービスの機能を実現してはならない。

使用上の注意

本システムサービスが呼び出されて正常終了しない限り、他の COM システムサービスは利用不可能である。なお、本システムサービスを呼び出しただけでは、ECU 間通信は開始されない。

3.6.2 Com_DeInit

サービス ID		0x02	
C 言語 I/F		void Com_DeInit(void) 【COM130】	
パラメータ	[in]	—	
	[in/out]	—	
	[out]	—	
返り値		—	
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM068】
同期/非同期		同期	
再入可能/再入不可能		再入可能	
機能		本システムサービスは, COM の未初期化を行い, ECU 間通信を終了させる.	

機能仕様

本システムサービスは, すべての IPDU グループを停止する 【COM129】.

設計上の注意

本システムサービスは, 他の COM システムサービスの機能を実現してはならない.

3.6.3 Com_IpduGroupControl

サービス ID		0x03	
C 言語 I/F		void Com_IpduGroupControl(Com_IpduGroupVector ipduGroupVector, boolean initialize) 【COM751】	
パラメータ	[in]	ipduGroupVector	全 IPDU グループに対して起動または停止を指定するための IPDU グループベクタ (ベクタ値：停止 = 0, 起動 = 1)
		initialize	新たに起動された IPDU の初期化を要求するフラグ
	[in/out]	—	
	[out]	—	
返り値		—	
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM069】
		COM_E_PARAM_POINTER	引数に NULL ポインタが指定された 【NCOM091】
同期/非同期		同期	
再入可能/再入不可能		再入可能	
機能		本システムサービスは、全 IPDU グループの起動および停止行う。	

機能仕様

本システムサービスは、引数 ipduGroupVector で指定された値に従って、IPDU グループの状態を起動状態または停止状態に設定する 【COM612】。

本システムサービスは、【COM612】により状態が変更された IPDU グループに対して、その状態に応じて IPDU グループの起動および停止を行う 【COM613】。

本システムサービスは、【COM613】により起動した IPDU グループに所属している IPDU に対して、【COM787】および【COM222】で定義されている(再)初期化処理を行う 【COM614】。

本システムサービスは、【COM612】により状態が変更されなかった IPDU グループに対して、処理を行わない 【COM615】。

3.6.4 Com_ReceptionDMControl

サービス ID		0x06	
C 言語 I/F		void Com_ReceptionDMControl(Com_IpduGroupVector ipduGroupVector) 【COM752】	
パラメータ	[in]	ipduGroupVector	全 IPDU グループに対して受信 DM の有効または無効を指定するための IPDU グループベクタ (ベクタ値：無効 = 0, 有効 = 1)
	[in/out]	—	
	[out]	—	
返り値		—	
DET エラー		COM_E_UNINIT	初期化未実施【NCOM070】
		COM_E_PARAM_POINTER	引数に NULL ポインタが指定された【NCOM092】
同期/非同期		同期	
再入可能/再入不可能		再入可能	
機能		本システムサービスは、全 IPDU グループの受信 DM 無効化および有効化を行う。	

機能仕様

本システムサービスは、引数 ipduGroupVector で指定された値に従って、IPDU グループの受信 DM 状態を有効状態または無効状態に設定する【COM616】。

本システムサービスは、【COM616】により受信 DM 状態が変更された IPDU グループに対して、受信 DM の有効化および無効化を行う【COM617】。

本システムサービスは、【COM616】により受信 DM 状態が変更されなかった IPDU グループに対して、処理を行わない【COM618】。

3.6.5 Com_GetStatus

サービス ID		0x07	
C 言語 I/F		Com_StatusType Com_GetStatus(void) 【COM194】	
パラメータ	[in]	—	
	[in/out]	—	
	[out]	—	
返り値		COM_UNINIT	COM 未初期化状態
		COM_INIT	COM 初期化状態
DET エラー		—	
同期/非同期		同期	
再入可能/再入不可能		再入可能	
機能		本システムサービスは、COM の現状態を返す。	

3.6.6 Com_GetConfigurationId

サービス ID		0x08	
C 言語 I/F		uint32 Com_GetConfigurationId(void) 【COM375】	
パラメータ	[in]	—	
	[in/out]	—	
	[out]	—	
返り値		ConfigurationId	コンフィギュレーション ID
DET エラー		—	
同期/非同期		同期	
再入可能/再入不可能		再入可能	
機能		本システムサービスは、現在有効なコンフィギュレーション ID を返す。	

機能仕様

初期化前に本システムサービスが呼び出された場合、コンフィギュレーション ID が確定していないため、正しい値を返せない。

Com_Init が呼び出される前に Com_GetConfigurationId が呼び出された場合、Com_GetConfigurationId は COM_INVALID_UINT32 (0xFFFFFFFF)を返す 【NCOM001】。

設計上の注意

コンフィギュレーション ID はコンフィギュレーション時に静的に指定され、COM によって変更することはできない 【COM619】。

3.6.7 Com_GetVersionInfo

サービス ID		0x09	
C 言語 I/F		void Com_GetVersionInfo(Std_VersionInfoType* versioninfo) 【COM426】	
パラメータ	[in]	—	
	[in/out]	—	
	[out]	versioninfo	バージョン情報
返り値		—	
DET エラー		COM_E_PARAM_POINTER	引数に NULL ポインタが指定された [NCOM039]
同期/非同期		同期	
再入可能/再入不可能		再入可能	
機能		本システムサービスは、COM のバージョン情報を返す。	

機能仕様

本システムサービスは、COM のバージョン情報として以下を返す 【COM424】.

- ・ ベンダ ID
- ・ モジュール ID
- ・ SW メジャーバージョン
- ・ SW マイナーバージョン
- ・ SW パッチバージョン

本システムサービスは、COM_VERSION_INFO_API の値によってプレコンパイル時に有効化または無効化される 【COM425】.

引数 versioninfo に NULL ポインタが指定され、かつ DET エラーが有効である場合、DET エラー COM_E_PARAM_POINTER を通知する 【NCOM039】.

設計上の注意

本システムサービスの呼び出し元と呼び出し先のソースコードが利用可能である場合、本システムサービスはマクロとして Com.h に定義される.

3.6.8 Com_ClearIpduGroupVector

サービス ID		0x1c	
C 言語 I/F		void Com_ClearIpduGroupVector(Com_IpduGroupVector ipduGroupVector) 【COM749】	
パラメータ	[in]	—	
	[in/out]	ipduGroupVector	クリアされる IPDU グループベクタ
	[out]	—	
返り値		—	
DET エラー		COM_E_PARAM_POINTER	引数に NULL ポインタが指定された 【NCOM093】
同期/非同期		同期	
再入可能/再入不可能		再入可能	
機能		本システムサービスは、IPDU グループベクタの全ビットを 0 クリアする。	

3.6.9 Com_SetIpduGroup

サービス ID		0x1d	
C 言語 I/F		void Com_SetIpduGroup(Com_IpduGroupVector ipduGroupVector, Com_IpduGroupIdType ipduGroupId, boolean bitval) 【COM750】	
パラメータ	[in]	ipduGroupId	設定する IPDU グループの ID
		bitval	設定するビット値
	[in/out]	ipduGroupVector	設定される IPDU グループベクタ
	[out]	—	
返り値		—	
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM071】
		COM_E_PARAM	引数に不正な値が指定された 【NCOM057】
		COM_E_PARAM_POINTER	引数に NULL ポインタが指定された 【NCOM094】
同期/非同期		同期	
再入可能/再入不可能		再入可能	
機能		本システムサービスは、IPDU グループベクタのうち、IPDU グループ ID に対応する位置のビット値を、指定されたビット値に設定する。	

機能仕様

本システムサービスは、引数 ipduGroupVector で渡された IPDU グループベクタのうち、引数 ipduGroupId で指定された IPDU グループ ID に対応する位置のビット値を、引数 bitval で指定されたビット値に設定する 【COM623】。

引数に指定された IPDU グループ数が, コンフィギュレーション時に設定された IPDU グループ数と等しいか, それより大きい場合を不正とする **【NCOM057】**.

3.6.10 Com_SendSignal

サービス ID		0x0a	
C 言語 I/F		uint8 Com_SendSignal(Com_SignalIdType SignalId, const void* SignalDataPtr) 【COM197】	
パラメータ	[in]	SignalId	送信するシグナルの ID
		SignalDataPtr	送信するシグナルデータへのポインタ
	[in/out]	—	
	[out]	—	
返回值		E_OK	送信成功
		COM_SERVICE_NOT_AVAILABLE	対応する IPDU グループが停止している (または, DET エラー発生により異常終了した)
		COM_BUSY	大型データ処理のため, TP バッファがロックされている
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM072】
		COM_E_PARAM	引数に不正な値が指定された 【NCOM058】
		COM_E_PARAM_POINTER	引数に NULL ポインタが指定された 【NCOM073】
同期/非同期		非同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは, シグナルを送信する.	

機能仕様

本システムサービスは, 引数 **SignalId** で指定されたシグナル ID に対応するシグナルバッファを, 引数 **SignalDataPtr** で指定されたシグナルデータで更新する 【COM624】.

【COM624】で更新されたシグナルの *ComTransferProperty* に **TRIGGERED** が指定されており, かつ, そのシグナルが所属している IPDU の *ComTxModeMode* に **DIRECT** または **MIXED** が指定されている場合, 本システムサービスは, 他の COM システムサービスによって遅延されない限り, その IPDU の送信を即時に行う 【COM625】 【NCOM006】.

引数に指定されたシグナル ID が, コンフィギュレーション時に設定された IPDU に所属するシグナル数と等しいか, それより大きい場合を不正とする 【NCOM058】.

設計上の注意

本システムサービスに関連する機能として, MDT やオフセット送信が挙げられる.

大型データ通信は本バージョン対象外の機能であるため, 返回值 **COM_BUSY** は対象外となる.

3.6.11 Com_SendDynSignal

サービス ID		0x21	
C 言語 I/F		uint8 Com_SendDynSignal(Com_SignalIdType SignalId, const void* SignalDataPtr, uint16 Length) 【COM627】	
パラメータ	[in]	SignalId	送信するシグナルの ID
		SignalDataPtr	送信するシグナルデータへのポインタ
		Length	送信するシグナルの長さ
	[in/out]	—	
	[out]	—	
返り値		E_OK	送信成功
		E_NOT_OK	引数 Length で指定された値が、対象シグナルの <i>ComSignalLength</i> で指定された値よりも大きい
		COM_SERVICE_NOT_AVAILABLE	対応する IPDU グループが停止している (または、DET エラー発生により異常終了した)
		COM_BUSY	大型データ処理のため、TP バッファがロックされている
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM085】
		COM_E_PARAM	引数に不正な値が指定された 【NCOM086】
		COM_E_PARAM_POINTER	引数に NULL ポインタが指定された 【NCOM087】
同期/非同期		非同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは、可変長シグナルを送信する.	

機能仕様

本システムサービスは、引数 **SignalId** で指定されたシグナル ID に対応するシグナルバッファを、引数 **SignalDataPtr** で指定されたシグナルデータで更新する 【COM628】.

【COM628】で更新されたシグナルの *ComTransferProperty* に **TRIGGERED** が指定されており、そのシグナルが所属している IPDU の *ComTxModeMode* に **DIRECT** または **MIXED** が指定されている場合、本システムサービスは、その IPDU の送信を即時に行う 【COM629】.

【COM628】で更新されたシグナルの *ComTransferProperty* に **PENDING** が指定されている場合、本システムサービスは、そのシグナルが所属している IPDU の送信は行わない 【COM630】.

引数に指定されたシグナル ID が、コンフィギュレーション時に設定された IPDU に所属するシグナル数と等しいか、それより大きい場合を不正とする 【NCOM086】.

設計上の注意

可変長シグナル通信は本バージョン対象外の機能であるため、本システムサービスは対象外である.

使用上の注意

可変長シグナルの長さの最大値は *ComSignalLength* に指定した値を超えてはならないが、最小値に対する制限はない。したがって、可変長シグナルの長さをゼロに指定することも可能である。

3.6.12 Com_ReceiveSignal

サービス ID		0x0b	
C 言語 I/F		uint8 Com_ReceiveSignal(Com_SignalIdType SignalId, void* SignalDataPtr) 【COM198】	
パラメータ	[in]	SignalId	受信するシグナルの ID
	[in/out]	—	
	[out]	SignalDataPtr	受信したシグナルデータの格納場所へのポインタ
返り値		E_OK	受信成功
		COM_SERVICE_NOT_AVAILABLE	対応する IPDU グループが停止している (または, DET エラー発生により異常終了した)
		COM_BUSY	大型データ処理のため, TP バッファがロックされている
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM074】
		COM_E_PARAM	引数に不正な値が指定された 【NCOM059】
		COM_E_PARAM_POINTER	引数に NULL ポインタが指定された 【NCOM075】
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは, シグナルを受信する.	

機能仕様

本システムサービスは, 引数 **SignalId** で指定されたシグナル ID に対応するシグナルバッファのデータを, 引数 **SignalDataPtr** で指定された位置に格納する 【COM631】.

引数に指定されたシグナル ID が, コンフィギュレーション時に設定された IPDU に所属するシグナル数と等しいか, それより大きい場合を不正とする 【NCOM059】.

本システムサービスが呼び出された場合, 対応する IPDU 内を引数 **SignalDataPtr** のデータで常に更新する. よって, その IPDU の受信があった場合は最新の受信データで更新し, その IPDU の受信がなかった場合はシグナルの初期値 **ComSignalInitValue** で更新する. この更新は, たとえ IPDU が停止しており, **COM_SERVICE_NOT_AVAILABLE** が返された場合でも行う 【COMa019】.

設計上の注意

大型データ通信は本バージョン対象外の機能であるため, 返り値 **COM_BUSY** は対象外となる.

使用上の注意

本システムサービスは, シグナルバッファの更新を行わない.

3.6.13 Com_ReceiveDynSignal

サービス ID		0x22	
C 言語 I/F		uint8 Com_ReceiveDynSignal(Com_SignalIdType SignalId, void* SignalDataPtr, uint16* Length) 【COM690】	
パラメータ	[in]	SignalId	受信するシグナルの ID
	[in/out]	Length	in : 受信可能な長さの最大値 out : 受信したシグナルの長さ
	[out]	SignalDataPtr	受信したシグナルデータの格納場所へのポインタ
返り値		E_OK	受信成功
		E_NOT_OK	引数 Length で指定された値が、受信したシグナルの長さよりも小さい
		COM_SERVICE_NOT_AVAILABLE	対応する IPDU グループが停止している (または、DET エラー発生により異常終了した)
		COM_BUSY	TP バッファがロックされている
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM088】
		COM_E_PARAM	引数に不正な値が指定された 【NCOM089】
		COM_E_PARAM_POINTER	引数に NULL ポインタが指定された 【NCOM090】
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは、可変長シグナルを受信する.	

機能仕様

本システムサービスは、引数 SignalId で指定されたシグナル ID に対応するシグナルバッファのデータを、引数 SignalDataPtr で指定された位置に格納する 【COM711】.

本システムサービスは、引数 SignalId で指定されたシグナル ID に対応するシグナルバッファのデータの長さを、引数 Length で指定された位置に格納する 【COM712】.

本システムサービスは、引数 Length で指定された受信可能な長さの最大値よりも受信したシグナルの長さの方が長い場合、【COM711】と【COM712】の処理を行わない 【COM724】.

引数に指定されたシグナル ID が、コンフィギュレーション時に設定された IPDU に所属するシグナル数と等しいか、それより大きい場合を不正とする 【NCOM089】.

設計上の注意

可変長シグナル通信は本バージョン対象外の機能であるため、本システムサービスは対象外である.

3.6.14 Com_SendSignalGroup

サービス ID		0x0d	
C 言語 I/F		uint8 Com_SendSignalGroup(Com_SignalGroupIdType SignalGroupId) 【COM200】	
パラメータ	[in]	SignalGroupId	送信するシグナルグループの ID
	[in/out]	—	
	[out]	—	
返り値		E_OK	送信成功
		COM_SERVICE_NOT_AVAILABLE	対応する IPDU グループが停止している (または, DET エラー発生により異常終了した)
		COM_BUSY	大型データ通信のため, TP バッファがロックされている
DET エラー		COM_E_UNINIT	初期化未実施【NCOM076】
		COM_E_PARAM	引数に不正な値が指定された【NCOM060】
同期/非同期		非同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは, シェドウバッファのデータで IPDU を更新する.	

機能仕様

本システムサービスは, 引数 SignalGroupId で指定されたシグナルグループのシェドウバッファのデータで IPDU を更新する【COM635】.

引数に指定されたシグナルグループ ID が, コンフィギュレーション時に設定された IPDU に所属するシグナルグループ数と等しいか, それより大きい場合を不正とする【NCOM060】.

使用上の注意

本システムサービスを使用する場合, コンフィギュレーション時にシグナルグループを設定する必要がある【COM637】[COM345_Conf].

本システムサービスを呼び出す前に, Com_SendSignal を呼び出して対象のシグナルグループのシェドウバッファを更新しなければならない【COM636】.

大型データ通信は本バージョン対象外の機能であるため, 返り値 COM_BUSY は対象外となる.

3.6.15 Com_ReceiveSignalGroup

サービス ID		0x0e	
C 言語 I/F		uint8 Com_ReceiveSignalGroup(Com_SignalGroupIdType SignalGroupId) 【COM201】	
パラメータ	[in]	SignalGroupId	受信するシグナルグループの ID
	[in/out]	—	
	[out]	—	
返り値		E_OK	送信成功
		COM_SERVICE_NOT_AVAILABLE	対応する IPDU グループが停止している (または, DET エラー発生により異常終了した)
		COM_BUSY	大型データ通信のため, TP バッファがロックされている
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM077】
		COM_E_PARAM	引数に不正な値が指定された 【NCOM061】
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは, IPDU 内のデータでシャドウバッファを更新する.	

機能仕様

本システムサービスは, 引数 `SignalGroupId` で指定されたシグナルグループが所属する IPDU からデータを取り出し, 対応するシャドウバッファを更新する 【COM638】.

本システムサービスを使用する場合, コンフィギュレーション時にシグナルグループを設定する必要がある 【COM639】 【COM345_Conf】.

本システムサービスが呼び出された場合, 対応する IPDU 内のデータでシャドウバッファを常に更新する. よって, その IPDU の受信があった場合は最新の受信データで更新し, その IPDU の受信がなかった場合はグループシグナルの初期値 `ComSignalInitValue` で更新する. この更新は, たとえ IPDU が停止しており, `COM_SERVICE_NOT_AVAILABLE` が返された場合でも行う 【COM461】.

引数に指定されたシグナルグループ ID が, コンフィギュレーション時に設定された IPDU に所属するシグナルグループ数と等しいか, それより大きい場合を不正とする 【NCOM061】.

使用上の注意

RTE が `Com_ReceiveSignal` を呼び出してシャドウバッファのデータを取り出すためには, 事前に本システムサービスを呼び出す必要がある.

大型データ通信は本バージョン対象外の機能であるため, 返り値 `COM_BUSY` は対象外となる.

3.6.16 Com_InvalidateSignal

サービス ID		0x10	
C 言語 I/F		uint8 Com_InvalidateSignal(Com_SignalIdType SignalId) 【COM203】	
パラメータ	[in]	SignalId	無効化するシグナルの ID
	[in/out]	—	
	[out]	—	
返り値		E_OK	無効化成功
		COM_SERVICE_NOT_AVAILABLE	対応する IPDU グループが停止している, または, 指定されたシグナルに無効値が設定されていない (または, DET エラー発生により異常終了した)
		COM_BUSY	大型データ処理のため, TP バッファがロックされている
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM078】
		COM_E_PARAM	引数に不正な値が指定された 【NCOM062】
同期/非同期		非同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは, シグナルを無効化する.	

機能仕様

本システムサービスは, 引数 SignalId で指定されたシグナルを無効化した後, 内部で Com_SendSignal を呼び出すことで無効値を送信する 【COM642】 【COM099】.

本システムサービスは, 引数 SignalId で指定されたシグナル(ComSignal)に無効値 (ComSignalDataInvalidValue)が設定されていない場合, COM_SERVICE_NOT_AVAILABLE を返す 【COM643】.

引数に指定されたシグナル ID が, コンフィギュレーション時に設定された IPDU に所属するシグナル数と等しいか, それより大きい場合を不正とする 【NCOM062】.

設計上の注意

大型データ通信は本バージョン対象外の機能であるため, 返り値 COM_BUSY は対象外となる.

3.6.17 Com_InvalidateSignalGroup

サービス ID		0x1b	
C 言語 I/F		uint8 Com_InvalidateSignalGroup(Com_SignalGroupIdType SignalGroupId) 【COM557】	
パラメータ	[in]	SignalGroupId	無効化するシグナルグループの ID
	[in/out]	—	
	[out]	—	
返り値		E_OK	無効化成功
		COM_SERVICE_NOT_AVAILABLE	対応する IPDU グループが停止している, または, 指定されたシグナルグループに所属しているグループシグナルのいずれかに無効値が設定されていない (または, DET エラー発生により異常終了した)
		COM_BUSY	大型データ通信のため, TP バッファがロックされている
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM079】
		COM_E_PARAM	引数に不正な値が指定された 【NCOM063】
同期/非同期		非同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは, シグナルグループを無効化する.	

機能仕様

本システムサービスは, 引数 SignalGroupId で指定されたシグナルを無効化した後, 内部で Com_SendSignalGroup を呼び出すことで無効値を送信する 【COM645】

引数に指定されたシグナルグループ ID が, コンフィギュレーション時に設定された IPDU に所属するシグナルグループ数と等しいか, それより大きい場合を不正とする 【NCOM063】.

設計上の注意

大型データ通信は本バージョン対象外の機能であるため, 返り値 COM_BUSY は対象外となる.

3.6.18 Com_TriggerIPDUSend

サービス ID		0x17	
C 言語 I/F		void Com_TriggerIPDUSend(PduIdType PduId) 【COM348】	
パラメータ	[in]	PduId	トリガ送信する IPDU の ID
	[in/out]	—	
	[out]	—	
返り値		—	
DET エラー	COM_E_UNINIT	初期化未実施 【NCOM080】	
	COM_E_PARAM	引数に不正な値が指定された 【NCOM064】	
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは、IPDU をトリガ送信する.	

機能仕様

本システムサービスは、IPDU 送信時に MDT のみを考慮し、必要に応じて送信の延期や MDT タイマのリセットを行う。送信モードは考慮しない 【COM388】.

本システムサービスは、引数 PduId で指定された IPDU にコールアウト関数が設定されている場合は、そのコールアウト関数を呼び出す 【COM492】 【COM387_Conf】.

引数に指定された IPDU 数が、コンフィギュレーション時に設定された IPDU 数と等しいか、それより大きい場合を不正とする 【NCOM064】.

使用上の注意

本システムサービス実行中に、再度本システムサービスの呼び出す場合、アプリケーションプログラマは無限再帰を作成しないように注意すること。本システムサービスを、同じ ComPduId で呼び出すことによる直接再帰と、長い再帰パスを持つ間接再帰は、理論上可能である。

3.6.19 Com_SwitchIpduTxMode

サービス ID		0x27	
C 言語 I/F		void Com_SwitchIpduTxMode(PduIdType PduId, boolean Mode) 【COM784】	
パラメータ	[in]	PduId	送信モードを変更する IPDU の ID
		Mode	送信モード
	[in/out]	—	
	[out]	—	
返り値		—	
DET エラー	COM_E_UNINIT		初期化未実施【NCOM081】
	COM_E_PARAM		引数に不正な値が指定された【NCOM065】
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは、IPDU の送信モードを変更する.	

機能仕様

本システムサービスは、引数 PduId で指定された IPDU の送信モードを、引数 Mode で指定された送信モードで更新する。送信モードが変更された場合、新しい送信モードが即時に有効となる。なお、引数 PduId で指定された IPDU の送信モードが、引数 Mode で指定された送信モードと既に同じ場合、本システムサービスは何もしない。【COMa018】

引数に指定された IPDU 数が、コンフィギュレーション時に設定された IPDU 数と等しいか、それより大きい場合を不正とする【NCOM065】。

使用上の注意

ユーザは、本システムサービスを呼び出すことによる送信モード変更と、Com_SendSignal または Com_SendSignalGroup の呼び出しによる送信モード変更(TMS)を併用する場合、より直近に変更された送信モードが有効となることを考慮して使用しなければならない。

3.7 コールバック関数

3.7.1 Com_TriggerTransmit

サービス ID		0x41	
C 言語 I/F		Std_ReturnType Com_TriggerTransmit(PduIdType TxPduId, PduInfoType* PduInfoPtr) 【COM001】	
パラメータ	[in]	TxPduId	トリガ送信する IPDU の ID
	[in/out]	—	
	[out]	PduInfoPtr	送信する SDU データの格納場所へのポインタ
返り値	E_OK		SDU のトリガ送信成功
	E_NOT_OK		SDU のトリガ送信失敗 (PduInfoPtr は NULL ポインタとなる可能性がある)
DET エラー	COM_E_PARAM		引数に不正な値が指定された
	COM_E_PARAM_POINTER		引数に NULL ポインタが指定された
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは、下位層のモジュールからの送信要求を受けて、SDU をトリガ送信する.	

機能仕様

本システムサービスは、引数 TxPduId で指定された IPDU のデータのうち、SDU データを引数 PduInfoPtr で指定された場所の PduInfoPtr->SduDataPtr に格納し、その SDU データの長さを PduInfoPtr->SduLength に格納する 【COM647】.

本システムサービスは、MDT を考慮せず、MDT タイマのリセットも行わない 【COM475】.

引数 TxPduId で指定された IPDU が所属している IPDU グループが停止している場合、本システムサービスは E_NOT_OK を返すが、【COM647】で要求されている処理は行う。したがって、本システムサービスを呼び出した下位層のモジュール(PduR)は、E_NOT_OK が返された場合の PduInfoPtr の取り扱いを判断する必要がある。

本システムサービスは、引数 TxPduId で指定された IPDU の *IPDU 種別 ComIPduType* が NORMAL である場合、送信モードに関わらず処理を行う 【COM260】.

本システムサービスは、割込みにより呼び出される可能性がある 【COM648】.

設計上の注意

本バージョン対象外の機能であるため、本システムサービスは対象外である。

3.7.2 Com_RxIndication

サービス ID		0x42	
C 言語 I/F		void Com_RxIndication(PduIdType RxPduId, PduInfoType* PduInfoPtr) 【COM123】	
パラメータ	[in]	RxPduId	受信した IPDU の ID
		PduInfoPtr	受信した IPDU データへのポインタ
	[in/out]	—	
	[out]	—	
返り値		—	
DET エラー		COM_E_UNINIT	初期化未実施【NCOM082】
		COM_E_PARAM	引数に不正な値が指定された【NCOM066】
		COM_E_PARAM_POINTER	引数に NULL ポインタが指定された【NCOM083】
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは, 下位層のモジュール(PduR)からの IPDU 受信通知を受け付ける.	

機能仕様

本システムサービスは割込みにより呼び出される可能性があるため, データの整合性を保証する【COM649】.

引数に指定された IPDU 数が, コンフィギュレーション時に設定された IPDU 数と等しいか, それより大きい場合を不正とする【NCOM066】.

3.7.3 Com_TpRxIndication

サービス ID		0x1e		
C 言語 I/F		void Com_TpRxIndication(PduIdType PduId, NotifResultType Result) 【COM650】		
パラメータ	[in]	RxPduId	受信した IPDU の ID	
		Result	NTFRSLT_OK	IPDU 受信成功し、受信バッファにデータが格納されている
			上記以外	IPDU 受信失敗 (受信バッファのロック解除が可能)
	[in/out]	—		
	[out]	—		
返り値		—		
DET エラー		—		
同期/非同期		同期		
再入可能/再入不可能		再入可能.		
機能		本システムサービスは、下位層のモジュール(PduR)からの大型 IPDU または可変長 IPDU の受信通知を受け付ける.		

機能仕様

本システムサービスは割込みにより呼び出される可能性があるため、データの整合性を保証する【COM651】.

設計上の注意

大型データ通信および可変長データ通信は本バージョン対象外の機能であるため、本システムサービスは対象外である.

3.7.4 Com_TxConfirmation

サービス ID		0x40	
C 言語 I/F		void Com_TxConfirmation(PduIdType TxPduId) 【COM124】	
パラメータ	[in]	TxPduId	送信した IPDU の ID
	[in/out]	－	
	[out]	－	
返り値		－	
DET エラー		COM_E_UNINIT	初期化未実施 【NCOM084】
		COM_E_PARAM	引数に不正な値が指定された 【NCOM067】
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは、下位層のモジュール(PduR)からの IPDU 送信完了通知を受け付ける.	

機能仕様

本システムサービスは割込みにより呼び出される可能性があるため、データの整合性を保証する【COM652】.

引数に指定された IPDU 数が、コンフィギュレーション時に設定された IPDU 数と等しいか、それより大きい場合を不正とする 【NCOM067】.

3.7.5 Com_TpTxConfirmation

サービス ID		0x26	
C 言語 I/F		void Com_TpTxConfirmation(PduIdType PduId, NotifResultType Result) 【COM725】	
パラメータ	[in]	PduId	送信した IPDU の ID
		Result	IPDU 送信結果
	[in/out]	－	
	[out]	－	
返り値		－	
DET エラー		－	
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは、下位層のモジュール(PduR)からの大型 IPDU または可変長 IPDU の送信完了通知を受け付ける.	

設計上の注意

大型データ通信および可変長データ通信は本バージョン対象外の機能であるため、本システムサービスは対象外である.

3.7.6 Com_StartOfReception

サービス ID		0x25	
C 言語 I/F		BufReq_ReturnType Com_StartOfReception(PduIdType ComRxPduId, PduLengthType TpSduLength, PduLengthType* RxBufferSizePtr) 【COM691】	
パラメータ	[in]	ComRxPduId	TP で受信する IPDU の ID
		TpSduLength	TP で受信する IPDU の長さ
	[in/out]	—	
	[out]	RxBufferSizePtr	内部 TP の受信バッファサイズの格納場所へのポインタ
返り値		BUFREQ_OK	接続が確立された(RxBufferSizePtr には受信可能なバッファサイズが格納されている)
		BUFREQ_E_NOT_OK	接続が拒否された(RxBufferSizePtr は更新されない)
		BUFREQ_E_OVFL	PDU 長(PduLength)が TpSduLength より小さい
		BUFREQ_E_BUSY	受信バッファが新しい受信を受け付けられない (実装依存)
DET エラー		—	
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは、内部 TP の受信バッファサイズを返す.	

機能仕様

本システムサービスは、引数 ComRxPduId で指定された IPDU が大型 IPDU であり、かつ、引数 TpSduLength で指定された値がコンフィギュレーション時の IPDU の長さより大きい場合、BUFREQ_E_OVFL を返す【COM654】.

本システムサービスは、引数 ComRxPduId で指定された IPDU が可変長 IPDU であり、かつ、引数 TpSduLength で指定された値がコンフィギュレーション時の IPDU の長さより大きい場合、BUFREQ_E_OVFL を返す【COM655】.

本システムサービスは、引数 ComRxPduId で指定された IPDU が可変長 IPDU である場合、引数 TpSduLength を用いて、その IPDU に所属している可変長シグナルの長さを計算する【COM656】.

本システムサービスは、COM 内部の TP バッファがロックされておらず、かつ、引数 TpSduLength で指定された値がコンフィギュレーション時の IPDU の長さより小さい場合、引数 RxBufferSizePtr で指定された場所に引数 ComRxPduId で指定された IPDU のバッファサイズを格納する【COM657】.

本システムサービスは、引数 ComRxPduId で指定された IPDU が所属している IPDU グループが停止している場合、BUFREQ_E_NOT_OK を返す【COM721】.

設計上の注意

大型データ通信および可変長データ通信は本バージョン対象外の機能であるため、本システムサービスは対象外である。

3.7.7 Com_CopyRxData

サービス ID		0x23	
C 言語 I/F		BufReq_ReturnType Com_CopyRxData(PduIdType PduId, const PduInfoType* PduInfoPointer, PduLengthType* RxBufferSizePtr) 【COM692】	
パラメータ	[in]	PduId	TP で受信する IPDU の ID
		PduInfoPointer	TP で受信する IPDU データへのポインタ(受信する長さ(SduLength)と受信するデータ(SduDataPtr)が含まれる) SduLength にゼロを指定した場合, 使用可能な受信バッファのサイズが取得できるが, PduInfoPtr は不正なデータとなる
	[in/out]	—	
	[out]	RxBufferSizePtr	TP バッファのサイズへのポインタ
返り値		BUFREQ_OK	受信データのコピー成功
		BUFREQ_E_NOT_OK	受信データのコピー失敗
		BUFREQ_E_BUSY	受信バッファが使用不可能 (実装依存)
DET エラー		—	
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは, 受信データを TP バッファにコピーする.	

機能仕様

本システムサービスは, 割込みにより呼び出される可能性があるため, データの整合性を保証する【COM658】.

本システムサービスは, 引数 PduId で指定された IPDU の所属 IPDU グループが停止している場合, BUFREQ_E_NOT_OK を返し, データはコピーしない【COM782】.

設計上の注意

大型データ通信および可変長データ通信は本バージョン対象外の機能であるため, 本システムサービスは対象外である.

使用上の注意

本システムサービスは, Com_StartOfReception の呼び出しにより接続が有効化されている場合のみ有効となる.

3.7.8 Com_CopyTxData

サービス ID		0x24	
C 言語 I/F		BufReq_ReturnType Com_CopyTxData(PduIdType PduId, PduInfoType* PduInfoPtr, RetryInfoType* RetryInfoPtr, PduLengthType* TxDataCntPtr) 【COM693】	
パラメータ	[in]	PduId	TP で送信する IPDU の ID
		PduInfoPtr	TP で送信する IPDU データへのポインタ(送信するデータ(SduDataPtr)とその長さ(SduLength)が含まれる)である。SduLength にゼロを指定した場合は送信可能なデータサイズが取得できるが、PduInfoPtr は不正なデータとなる。
		RetryInfoPtr	大型 IPDU の送信が完了または中止されるまで、常に完全なバッファを保持しているため、COM はこのポインタの値を無視する
	[in/out]	—	
	[out]	TxDataCntPtr	残りの送信データサイズ
戻り値		BUFREQ_OK	送信データのコピー成功
		BUFREQ_E_NOT_OK	送信データのコピー失敗
		BUFREQ_E_BUSY	TP バッファが使用不可能 (実装依存)
DET エラー		—	
同期/非同期		同期	
再入可能/再入不可能		再入可能.	
機能		本システムサービスは、大型 IPDU の送信データを送信 TP バッファにコピーして、TxDataCntPtr に残りの送信データのサイズを格納する。	

機能仕様

本システムサービスは、以下を考慮する 【COM663】.

- ・ 割込みにより呼び出される可能性があるため、データの整合性が取れていること
- ・ ComPduId が PduR_ComTransmit の ID と一致していること

本システムサービスは、引数 PduId で指定された IPDU の所属している IPDU グループが停止している場合、BUFREQ_E_NOT_OK を返し、データはコピーしない 【COM783】.

3.8 周期関数

3.8.1 Com_MainFunctionRx

サービス ID	0x18
C 言語 I/F	void Com_MainFunctionRx(void) 【COM398】
起動タイミング	FIXED_CYCLIC
DET エラー	—
機能	本システムサービスは、受信処理を周期的に行う。

機能仕様

事前に Com_Init が呼び出されて COM の初期化が完了しない限り、本システムサービスは処理を何もせずに終了する 【COM664】。

3.8.2 Com_MainFunctionTx

サービス ID	0x19
C 言語 I/F	void Com_MainFunctionTx(void) 【COM399】
起動タイミング	FIXED_CYCLIC
DET エラー	—
機能	本システムサービスは、送信処理を周期的に行う。

機能仕様

事前に Com_Init が呼び出されて COM の初期化が完了しない限り、本システムサービスは処理を何もせずに終了する 【COM665】。

AUTOSAR 仕様との違い

AUTOSAR 仕様では、即時送信対象のシグナルに対して Com_SendSignal が発行された場合に、Com_SendSignal の中で PduR_ComTransmit を呼び出すか、次回の Com_MainFunctionTx 実行時に呼び出すかは、実装依存となっている。

本仕様では、次回の Com_MainFunctionTx 実行時に、PduR_ComTransmit を呼び出すこととする 【NCOM006】。

よって、ユーザが Com_MainFunctionTx の起動周期(ComTxTimeBase) に指定した時間によっては、Com_SendSignal を呼び出してから PduR_ComTransmit が呼び出されるまでの間隔が開いてしまう場合もあるため、注意すること。

3.8.3 Com_MainFunctionRouteSignals

サービス ID	0x1a
C 言語 I/F	void Com_MainFunctionRouteSignals(void) 【COM400】
起動タイミング	FIXED_CYCLIC
DET エラー	—
機能	本システムサービスは、シグナルゲートウェイの処理を周期的に行う。

機能仕様

事前に Com_Init が呼び出されて COM の初期化が完了しない限り、本システムサービスは処理を何もせずに終了する 【COM666】。

本システムサービスは、以下を考慮する 【COM667】。

- ・ 起動周期(おそらく従来のタスクまたはスレッドのサイクル)は直接シグナルゲートウェイの待ち時間に影響を与えること
- ・ 非干渉受信割込みおよびシグナルゲートウェイタスクのための呼び出しを挿入すること

本システムサービスを周期的に起動するための周期タスクを設定する必要がある 【COM668】
[COM186_Conf]。

設計上の注意

シグナルゲートウェイは本バージョン対象外の機能であるため、本システムサービスは対象外となる。

3.9 依存インタフェース

3.9.1 必須インタフェース

本 COM の機能を実現するために、表 3-4 で示すインタフェースが必要となる【COM669】.

表 3-4 必須インタフェース

システムサービス名	概要
PduR_ComTransmit	PduR に対して、IPDU の送信を要求するためのシステムサービス.

AUTOSAR 仕様との違い

AUTOSAR 仕様では、〔COM669〕において Dem_ReportErrorStatus も必須インタフェースと定義している. しかし、COM は標準エラーを持たないため、DEM への報告が不要である. よって、必須インタフェース一覧から Dem_ReportErrorStatus を削除した.

3.9.2 任意インタフェース

本 COM の任意の機能を実現するために、表 3-5 で示すインタフェースが必要となる【COM670】.

表 3-5 任意インタフェース

システムサービス名	概要
Det_ReportError	BSW モジュールから報告されたイベントをキューに入れるためのシステムサービス(BSW モジュールのみが使用する). 報告されたイベントに対する処理自体は Dem のメイン関数内で行う.
PduR_ComCancelTransmit	PduR に対して、IPDU の送信を要求するためのシステムサービス.

Det_ReportError は、*DET* 使用有無 *ComConfigurationUseDet* に TRUE が設定された場合に有効となり、FALSE が設定された場合に無効となる〔COM141_Conf〕.

PduR_ComCancelTransmit は、送信キャンセルサポート *ComTxIPduCancelTransmitSupport* に TRUE が設定された IPDU が存在する場合に有効となり、そうでない場合に無効となる〔COM709_Conf〕.

3.9.3 コンフィギュレーション可能なインタフェース

3.9.3.1 コールバック関数

コールバック関数は割込みレベルまたはタスクレベルで実行されるため、それぞれのレベルの OS 制約が適用される。

3.9.3.1.1 Com_CbkTxAck

C 言語 I/F		void Com_CbkTxAck(void) 【COM468】
パラメータ	[in]	—
	[in/out]	—
	[out]	—
返り値		—
DET エラー		—
同期/非同期		同期
再入可能/再入不可能		考慮しない
機能		本コールバック関数は、IPDU 送信成功を通知する。

コンフィギュレーション仕様

本コールバック関数は、送信側でのみ呼び出される。

本コールバック関数は、シグナルおよびシグナルグループごとに設定可能であり、RTE が生成した Rte_COMCbkTxAck_<sn>または Rte_COMCbkTxAck_<sg>と対応している。

本コールバック関数の名前は通知 *ComNotification* で定義される [COM498_Conf]。

3.9.3.1.2 Com_CbkTxErr

C 言語 I/F		void Com_CbkTxErr(void) 【COM491】
パラメータ	[in]	—
	[in/out]	—
	[out]	—
返り値		—
DET エラー		—
同期/非同期		同期
再入可能/再入不可能		考慮しない
機能		本コールバック関数は、IPDU 送信失敗を通知する。

コンフィギュレーション仕様

本コールバック関数は、送信側でのみ呼び出される。

本コールバック関数は、シグナルおよびシグナルグループごとに設定可能であり、RTE が生成した

Rte_COMCbktTxErr_<sn>または Rte_COMCbktTxErr_<sg>と対応している.

本コールバック関数の名前はエラー通知 *ComErrorNotification* で定義される [COM499_Conf].

3.9.3.1.3 Com_CbkTxTOut

C 言語 I/F		void Com_CbkTxTOut(void) 【COM554】
パラメータ	[in]	—
	[in/out]	—
	[out]	—
返り値		—
DET エラー		—
同期/非同期		同期
再入可能/再入不可能		考慮しない
機能		本コールバック関数は、送信タイムアウト発生を通知する.

コンフィギュレーション仕様

本コールバック関数は、送信側でのみ呼び出される.

本コールバック関数は、シグナルおよびシグナルグループごとに設定可能であり、RTE が生成した Rte_COMCbktTxTOut_<sn>または Rte_COMCbktTxTOut_<sg>と対応している.

本コールバック関数の名前はタイムアウト通知 *ComTimeoutNotification* で定義される [COM552_Conf].

3.9.3.1.4 Com_CbkRxAck

C 言語 I/F		void Com_CbkRxAck(void) 【COM555】
パラメータ	[in]	—
	[in/out]	—
	[out]	—
返り値		—
DET エラー		—
同期/非同期		同期
再入可能/再入不可能		考慮しない
機能		本コールバック関数は、IPDU 受信成功を通知する.

コンフィギュレーション仕様

本コールバック関数は、受信側でのみ呼び出される.

本コールバック関数は、シグナルおよびシグナルグループごとに設定可能であり、RTE が生成した Rte_COMCbktRxAck_<sn>または Rte_COMCbktRxAck_<sg>と対応している.

本コールバック関数の名前は通知 *ComNotification* で定義される [COM498_Conf].

3.9.3.1.5 Com_CbkRxTOut

C 言語 I/F		void Com_CbkRxTOut(void) 【COM556】
パラメータ	[in]	—
	[in/out]	—
	[out]	—
返り値		—
DET エラー		—
同期/非同期		同期
再入可能/再入不可能		考慮しない
機能		本コールバック関数は、受信タイムアウト発生を通知する.

コンフィギュレーション仕様

本コールバック関数は、受信側でのみ呼び出される.

本コールバック関数は、シグナルおよびシグナルグループごとに設定可能であり、RTE が生成した Rte_COMCbkRxTOut_<sn>または Rte_COMCbkRxTOut_<sg>と対応している.

本コールバック関数の名前はタイムアウト通知 *ComTimeoutNotification* で定義される [COM552_Conf].

3.9.3.1.6 Com_CbkInv

C 言語 I/F		void Com_CbkInv(void) 【COM536】
パラメータ	[in]	—
	[in/out]	—
	[out]	—
返り値		—
DET エラー		—
同期/非同期		同期
再入可能/再入不可能		考慮しない
機能		本コールバック関数は、無効値の受信を通知する.

コンフィギュレーション仕様

本コールバック関数は、受信側でのみ呼び出される.

本コールバック関数は、シグナルおよびシグナルグループごとに設定可能であり、RTE が生成した Rte_COMCbkInv_<sn>または Rte_COMCbkInv_<sg>と対応している.

本コールバック関数の名前は無効値通知 *ComInvalidNotification* で定義される [COM315_Conf].

3.9.3.1.7 Com_CbkCounterErr

C 言語 I/F		void Com_CbkCounterErr(PduIdType ComPduId, uint8 ExpectedCounter, uint8 ReceivedCounter) 【COM726】	
パラメータ	[in]	ComPduId	カウンタエラーが発生した IPDU の ID
		ExpectedCounter	カウンタ期待値
		ReceivedCounter	実際に受信したカウンタ値
	[in/out]	—	
	[out]	—	
返り値		—	
DET エラー		—	
同期/非同期		同期	
再入可能/再入不可能		考慮しない	
機能		本コールバック関数は、カウンタエラー発生を通知する。	

コンフィギュレーション仕様

本コールバック関数は、IPDU ごとに設定可能であり、受信側でのみ呼び出される。

本コールバック関数の名前は *IPDU* カウンタエラー通知 *ComIPduCounterErrorNotification* で定義される [COM003_Conf]。

3.9.3.2 コールアウト関数

COM は、IPDU のコールアウト関数のみをサポートする。IPDU の *IPDU 通信種別(ComIPduType)* が NORMAL または TP の場合、その IPDU のコールアウト関数をサポートする。

IPDU のコールアウト関数が FALSE を返した場合、COM は、その IPDU に対する処理をそれ以降行わない【COMa004】。

3.9.3.2.1 Com_RxIpduCallout

C 言語 I/F		boolean Com_RxIpduCallout(PduIdType Id, const uint8* IpduData) 【COM700】	
パラメータ	[in]	Id	受信した IPDU の ID
		IpduData	受信した IPDU のデータへのポインタ
	[in/out]	－	
	[out]	－	
返回值		TRUE	IPDU は正常に処理される
		FALSE	IPDU はこれ以上処理されない
DET エラー		－	
同期/非同期		同期	
再入可能/再入不可能		考慮しない	
機能		本コールアウト関数は、ユーザ定義の受信フィルタリングを行う。	

コンフィギュレーション仕様

本コールアウト関数の名前は *IPDU* コールアウト *ComIPduCallout* で定義される【COM387_Conf】。

機能仕様

TP によって大型 IPDU を受信した場合、本コールアウト関数は、Com_TpRxIndication の処理中に呼び出される【COM720】。

本コールアウト関数は、受信した全てのデータを IPDU に格納する【NCOM096】。

3.9.3.2.2 Com_TxIpduCallout

C 言語 I/F		boolean Com_TxIpduCallout(PduIdType Id, uint8* IpduData) 【COM346】	
パラメータ	[in]	Id	送信した IPDU の ID
		IpduData	送信した IPDU のデータへのポインタ
	[in/out]	—	
	[out]	—	
返回值		TRUE	IPDU は正常に処理される
		FALSE	IPDU はこれ以上処理されない
DET エラー		—	
同期/非同期		同期	
再入可能/再入不可能		考慮しない	
機能		本コールアウト関数は、ユーザ定義の送信フィルタリングまたは事前送信処理を行う。	

コンフィギュレーション仕様

本コールアウト関数の名前は *IPDU* コールアウト *ComIPduCallout* または *IPDU* トリガ送信コールアウト *ComIPduTriggerTransmitCallout* で定義される [COM387_Conf] [COM765_Conf]。

機能仕様

本コールアウト関数は、以下の COM システムサービスを呼び出すことが可能である 【COM381】。

- Com_TriggerIPDUSend
- Com_SendSignal
- Com_SendSignalGroup

本コールアウト関数によって更新される可能性があるシグナルおよびシグナルグループおよびグループシグナルの送信プロパティ(*ComTransferProperty*) には、TRIGGERED または TRIGGERED_ON_CHANGE を設定してはならない 【COM780】。

本コールアウト関数によって更新される可能性があるシグナルおよびグループシグナルにフィルタ(*ComFilter*) が設定される場合、そのフィルタアルゴリズム(*ComFilterAlgorithm*) には、ALWAYS または NEVER のどちらかを設定しなければならない 【COM781】。

Com_TriggerTransmit は、*IPDU* トリガ送信コールアウト *ComIPduTriggerTransmitCallout* が設定された *IPDU* を指定して呼び出された場合、そのコールアウト関数を呼び出す 【COM766】。

Com_TriggerTransmit は、*IPDU* トリガ送信コールアウト *ComIPduTriggerTransmitCallout* に設定されたコールアウト関数の返回值を無視する 【COM395】。

送信 IPDU に *IPDU* コールアウト *ComIPduCallout* が設定されている場合、その送信 IPDU を *PduR_ComTransmit* で送信する前に、本コールアウト関数を呼び出す【COM719】.

3.10 コンテナおよびパラメータ

3.10.1 Com

コンテナ名	/AUTOSAR/EcuDefs/Com 【COM540_Conf】
概要	COM トップコンテナ
多重度	1
パラメータ	—
サブコンテナ	ComConfig
	ComGeneral

3.10.2 ComGeneral

コンテナ名	/AUTOSAR/EcuDefs/Com/ComGeneral 【COM541_Conf】
概要	全般的な設定を含むコンテナ
多重度	1
パラメータ	ComConfigurationUseDet
	ComEnableMDTForCyclicTransmission
	ComRetryFailedTransmitRequests
	ComSupportedIPduGroups
	ComVersionInfoApi
サブコンテナ	—

3.10.2.1 ComConfigurationUseDet

パラメータ名	/AUTOSAR/EcucDefs/Com/ComGeneral/ComConfigurationUseDet 【COM141_Conf】	
概要	DET の使用有無	
多重度	0..1	
型	EcucBooleanParamDef	
値の範囲	TRUE	DET エラー有効
	FALSE	DET エラー無効
デフォルト値	FALSE 【NCOM103】	
コンフィギュレーションクラス		
プレコンパイルタイム		

コンフィギュレーション仕様

COM ジェネレータは、本パラメータの値を定義するための定数 COM_DEV_ERROR_DETECT を生成する [COM028]。

DET は開発時に明確に発行する意図がある場合のみ発行されるべきもので、ユーザに対しては発行すべきではないため、デフォルト値は FALSE とする [NCOM103]。

3.10.2.2 ComEnableMDTForCyclicTransmission

パラメータ名	/AUTOSAR/EcuDefs/Com/ComGeneral/ ComEnableMDTForCyclicTransmission 【COM788_Conf】		
概要	周期送信における MDT の使用有無		
多重度	0..1		
型	EcucBooleanParamDef		
値の範囲	TRUE	周期送信における MDT 有効	
	FALSE	周期送信における MDT 無効	
デフォルト値	TRUE		
コンフィギュレーションクラス			
プレコンパイルタイム			

3.10.2.3 ComRetryFailedTransmitRequests

パラメータ名	/AUTOSAR/EcucDefs/Com/ComGeneral/ ComRetryFailedTransmitRequests 【COM780_Conf】	
概要	失敗送信要求のリトライの使用有無	
多重度	0..1	
型	EcucBooleanParamDef	
値の範囲	TRUE	失敗した送信要求のリトライ有効
	FALSE	失敗した送信要求のリトライ無効
デフォルト値	TRUE	
コンフィギュレーションクラス		
プレコンパイルタイム		

3.10.2.4 ComSupportedIPduGroups

パラメータ名	/AUTOSAR/EcucDefs/Com/ComGeneral/ComSupportedIPduGroups 【COM710_Conf】
概要	IPDU グループ数の最大値
多重度	1
型	EcucIntegerParamDef
値の範囲	0..65535
デフォルト値	ー
コンフィギュレーションクラス	
プレコンパイルタイム	

3.10.2.5 ComVersionInfoApi

パラメータ名	/AUTOSAR/EcucDefs/Com/ComGeneral/ComVersionInfoApi 【COM438_Conf】	
概要	Com_GetVersionInfo 使用有無	
多重度	1	
型	EcucBooleanParamDef	
値の範囲	TRUE	Com_GetVersionInfo 有効
	FALSE	Com_GetVersionInfo 無効
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム		

3.10.3 ComConfig

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig 【COM337_Conf】
概要	コンフィギュレーションごとの設定を含むコンテナ
多重度	1..* 【NCOM002】
パラメータ	ComConfigurationId
サブコンテナ	ComTimeBase
	ComIPduGroup
	ComIPdu
	ComSignal
	ComSignalGroup
	ComGwMapping

AUTOSAR 仕様との違い

AUTOSAR 仕様では、本コンテナの多重度を 1 としているが、ポストビルド対応により複数のコンフィギュレーション情報に対応するため、多重度を 1..*とする [NCOM002]。

3.10.3.1 ComConfigurationId

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComConfigurationId 【COM394_Conf】
概要	コンフィギュレーション ID
多重度	1
型	EcucIntegerParamDef
値の範囲	0..4294967295
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.4 ComTimeBase

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComTimeBase 【COM730_Conf】
概要	タイムベースの設定を含むコンテナ
多重度	1
パラメータ	ComTxTimeBase
	ComRxTimeBase
	ComGwTimeBase
サブコンテナ	—

3.10.4.1 ComTxTimeBase

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComTimeBase/ComTxTimeBase 【COM186_Conf】
概要	Com_MainFunctionTx の起動周期[秒]
多重度	0..1
型	EcucFloatParamDef
値の範囲	$0 < x \leq 3600$ 【NCOM101】 $x : \text{ComTxTimeBase}$
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

コンフィギュレーション仕様

ComIPduDirection が SEND である ComIPdu が 1 つ以上存在している場合, ComTxTimeBase に値を指定しなければならない 【NCOM049】.

ComTxTimeBase に 0 を指定した場合, エラーとする 【NCOM101】.

3.10.4.2 ComRxTimeBase

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComTimeBase/ComRxTimeBase 【COM728_Conf】
概要	Com_MainFunctionRx の起動周期[秒]
多重度	0..1
型	EcucFloatParamDef
値の範囲	$0 < x \leq 3600$ 【NCOM102】 $x : \text{ComRxTimeBase}$
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

コンフィギュレーション仕様

ComIPduDirection が RECEIVE である ComIPdu が 1 つ以上存在している場合, ComRxTimeBase に値を指定しなければならない 【NCOM050】.

ComRxTimeBase に 0 を指定した場合, エラーとする 【NCOM102】.

3.10.4.3 ComGwTimeBase

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComTimeBase/ComGwTimeBase 【COM729_Conf】
概要	Com_MainFunctionRouteSignals の起動周期[秒]
多重度	0..1
型	EcucFloatParamDef
値の範囲	0..3600
デフォルト値	－
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.5 ComIPduGroup

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPduGroup 【COM341_Conf】
概要	IPDU グループの設定を含むコンテナ
多重度	1..* 【NCOM028】
パラメータ	ComIPduGroupHandleId
	ComIPduGroupGroupRef
サブコンテナ	—

AUTOSAR 仕様との違い

AUTOSAR 仕様では、多重度を 0..*としているが、本パラメータを設定しないユースケースが想定されないため、多重度を 1..*とする [NCOM028].

コンフィギュレーション仕様

COM ジェネレータは、本コンテナのショートネームと IPDU グループ ID (*ComIPduGroupHandleId*) を関連付けるための定数を Com_Cfg.h に生成する 【COM126】.

【COM126】で生成された定数は、Com_IpduGroupControl などの IPDU グループ ID を指定するシステムサービスを呼び出す際に使用できる.

3.10.5.1 ComIPduGroupHandleId

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPduGroup/ ComIPduGroupHandleId 【COM184_Conf】
概要	IPDU グループ ID (定数生成)
多重度	1
型	EcucIntegerParamDef
値の範囲	0..3600
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.5.2 ComIPduGroupGroupRef

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPduGroup/ ComIPduGroupGroupRef 【COM185_Conf】
概要	所属 IPDU グループ参照
多重度	0..*
型	/AUTOSAR/EcuDefs/ComConfig/ComIPduGroup への参照
値の範囲	—

デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

AUTOSAR 仕様との違い

本仕様では, IPDU グループのネスト所属をサポートしないため, *IPDU* グループ所属グループ (*ComIPduGroupGroupRef*) を指定した場合, エラーとする【NCOM009】.

3.10.6 ComIPdu

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu 【COM340_Conf】
概要	IPDU の設定を含むコンテナ
多重度	1..* 【NCOM027】
パラメータ	ComIPduHandleId
	ComIPduType
	ComIPduDirection
	ComIPduSignalProcessing
	ComIPduCancellationSupport
	ComIPduCallout
	ComIPduTriggerTransmitCallout
	ComIPduGroupRef
	ComIPduSignalRef
	ComIPduSignalGroupRef
サブコンテナ	ComPduIdRef
	ComTxIPdu
	ComIPduCounter
	ComIPduReplication

AUTOSAR 仕様との違い

AUTOSAR 仕様では、本コンテナの多重度を 0..*としているが、本コンテナを設定しないユースケースが想定されないため、多重度を 1..*とする [NCOM027].

コンフィギュレーション仕様

COM ジェネレータは、本コンテナのショートネームと *IPDU ID(ComIPduHandleId)* を関連付けるための定数を Com_Cfg.h に生成する 【COM174】.

〔COM174〕で生成された定数は、PduR_ComTransmit などの IPDU ID を指定するシステムサービスを呼び出す際に使用できる.

IPDU 通信方向(ComIPduDirection) に SEND が設定された場合、サブコンテナ ComTxIPdu の設定は必須となる 【COM497】.

3.10.6.1 ComIPduHandleId

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduHandleId 【COM175_Conf】
概要	IPDU の ID (受信通知または送信通知の対象 IPDU を識別するために使用される)

多重度	1【NCOM003】
型	EcucIntegerParamDef
値の範囲	0..65535
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

AUTOSAR 仕様との違い

AUTOSAR 仕様では、本パラメータの多重度を 0..1 としているが、送信通知および受信通知のどちらも起こらない IPDU は存在しないため、多重度を 1 とする [NCOM003].

3.10.6.2 ComIPduType

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComIPduType 【COM761_Conf】	
概要	IPDU 通信の種別	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	NORMAL	通常通信のための IPDU
	TP	TP 通信のための IPDU
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム，ポストビルドタイム		

3.10.6.3 ComIPduDirection

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComIPduDirection 【COM493_Conf】	
概要	IPDU 通信の方向	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	SEND	送信 IPDU
	RECEIVE	受信 IPDU
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム，ポストビルドタイム		

3.10.6.4 ComIPduSignalProcessing

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComIPduSignalProcessing 【COM119_Conf】	
概要	シグナル通知処理タイミング	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	DEFERRED	周期タスク(メイン関数)でシグナルの送信完了通知および受信通知を行う
	IMMEDIATE	Com_TxConfirmation および Com_RxIndication でシグナルの送信完了通知や受信通知を行う
デフォルト値	ー	
コンフィギュレーションクラス		
プレコンパイルタイム, リンクタイム, ポストビルドタイム		

3.10.6.5 ComIPduCancellationSupport

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComIPduCancellationSupport 【COM709_Conf】	
概要	IPDU 通信要求キャンセルの使用有無	
多重度	0..1	
型	EcucBooleanParamDef	
値の範囲	TRUE	IPDU 通信要求キャンセル有効
	FALSE	IPDU 通信要求キャンセル無効
デフォルト値	FALSE	
コンフィギュレーションクラス		
プレコンパイルタイム		

使用上の注意

IPDU 通信種別(ComIPduType) が NORMAL の場合, 送信要求キャンセルをサポートし, IPDU 通信種(ComIPduType) が TP の場合, 送信要求キャンセルと受信要求キャンセルをサポートする.

3.10.6.6 ComIPduCallout

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduCallout 【COM387_Conf】	
概要	IPDU 通信前に呼び出されるコールアウト関数名	
多重度	0..1	
型	EcucFunctionNameDef	

値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム	

3.10.6.7 ComIPduTriggerTransmitCallout

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ ComIPduTriggerTransmitCallout 【COM765_Conf】
概要	IPDU トリガ送信時に呼び出されるコールアウト関数名
多重度	0..1
型	EcucFunctionNameDef
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム	

3.10.6.8 ComIPduGroupRef

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduGroupRef 【COM206_Conf】
概要	所属している IPDU グループへの参照
多重度	1..* 【NCOM004】
型	/AUTOSAR/EcuDefs/ComConfig/ComIPduGroup への参照
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

AUTOSAR 仕様との違い

AUTOSAR 仕様では、本パラメータの多重度を 0..*としているが、IPDU グループに所属しない IPDU を開始する方法が存在しないため、多重度を 1..*とする [NCOM004].

3.10.6.9 ComIPduSignalRef

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduSignalRef 【COM518_Conf】
概要	所持しているシグナルへの参照
多重度	0..*

型	/AUTOSAR/EcuDefs/ComConfig/ComSignal への参照
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.6.10 ComIPduSignalGroupRef

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ ComIPduSignalGroupRef 【COM519_Conf】
概要	所持しているシグナルグループへの参照
多重度	0..*
型	/AUTOSAR/EcuDefs/ComConfig/ComSignalGroup への参照
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.6.11 ComPduIdRef

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComPduIdRef 【COM711_Conf】
概要	対応する PDU への参照
多重度	1
型	/AUTOSAR/EcuDefs/EcuC/EcuPduCollection/Pdu への参照
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

コンフィギュレーション仕様

Com に構成されている全ての IPDU に対応する PduRsrcPdu を持たない PduR と構成された場合エラーとする 【NCOM052】.

3.10.7 ComTxIPdu

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComTxIPdu 【COM496_Conf】
概要	送信 IPDU の設定を含むコンテナ

多重度	0..1
パラメータ	ComTxIPduUnusedAreasDefault
	ComTxIPduClearUpdateBit
	ComMinimumDelayTime
サブコンテナ	ComTxModeTrue
	ComTxModeFalse

コンフィギュレーション仕様

IPDU に属するシグナルのフィルタアルゴリズムにより、以下のように不要となるコンテナが指定されている場合、警告とする【NCOM041】.

- ・ IPDU に属するシグナルのフィルタアルゴリズムがすべて ALWAYS の場合、TMS の判定結果は常に true となるため、ComTxModeFalse の設定は不要
- ・ IPDU に属するシグナルのフィルタアルゴリズムがすべて NEVER の場合、TMS の判定結果は常に false となるため、ComTxModeTrue の設定は不要

3.10.7.1 ComTxIPduUnusedAreasDefault

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ComTxIPduUnusedAreasDefault 【COM017_Conf】
概要	IPDU 未使用領域のためのデフォルト値(バイトパターン)
多重度	1
型	EcucIntegerParamDef
値の範囲	0..255
デフォルト値	－
コンフィギュレーションクラス	
プレコンパイルタイム， リンクタイム	

3.10.7.2 ComTxIPduClearUpdateBit

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ComTxIPduClearUpdateBit 【COM576_Conf】	
概要	送信 IPDU が所持しているシグナルおよびシグナルグループの、アップデートビットをクリアするタイミング	
多重度	0..1	
型	EcucEnumerationParamDef	
値の範囲	Confirmation	送信完了通知を受け取った際にクリア
	Transmit	PduR_ComTransmit を呼び出した際にクリア
	TriggerTransmit	Com_TriggerTransmit により IPDU がフェッチされ

		た際にクリア
デフォルト値	TRANSMIT [NCOM030]	
コンフィギュレーションクラス		
プレコンパイルタイム, リンクタイム, ポストビルドタイム		

3.10.7.3 ComMinimumDelayTime

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ComMinimumDelayTime 【COM181_Conf】
概要	最小遅延時間(MDT) [秒]
多重度	0..1
型	EcucFloatParamDef
値の範囲	0..3600
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.8 ComTxModeTrue

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeTrue 【COM455_Conf】
概要	TMS が TRUE の場合の送信モード
多重度	0..1
パラメータ	—
サブコンテナ	ComTxMode

3.10.9 ComTxModeFalse

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeFalse 【COM454_Conf】
概要	TMS が FALSE の場合の送信モード
多重度	0..1
パラメータ	—
サブコンテナ	ComTxMode

3.10.10 ComTxMode

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeTrue/ComTxMode /AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeFalse/ComTxMode 【COM351_Conf】
概要	送信モードの設定を含むコンテナ
多重度	1
パラメータ	ComTxModeMode
	ComTxModeNumberOfRepetitions
	ComTxModeRepetitionPeriod
	ComTxModeTimeOffset
	ComTxModeTimePeriod
サブコンテナ	—

3.10.10.1 ComTxModeMode

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode/ComTxModeMode /AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode/ComTxModeMode 【COM137_Conf】	
概要	送信モード	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	DIRECT	直接(N 回)送信
	PERIODIC	周期送信
	MIXED	DIRECT+PERIODIC
	NONE	送信しない
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム，ポストビルドタイム		

3.10.10.2 ComTxModeNumberOfRepetitions

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeTrue/ComTxMode/ComTxModeNumberOfRepetitions /AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeFalse/ComTxMode/ComTxModeNumberOfRepetitions 【COM281_Conf】
--------	--

概要	直接 N 回送信の送信回数 (送信モードが DIRECT または MIXED の場合に有効)
多重度	0..1
型	EcucIntegerParamDef
値の範囲	0..255
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.10.3 ComTxModeRepetitionPeriod

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeTrue/ComTxMode/ComTxModeRepetitionPeriod /AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeFalse/ComTxMode/ComTxModeRepetitionPeriod 【COM282_Conf】
概要	直接 N 回送信の送信間隔[秒] (直接 N 回送信の送信回数が 1 以上であり, 送信モードが DIRECT または MIXED の場合に有効)
多重度	0..1
型	EcucFloatParamDef
値の範囲	$0 < x \leq 3600$ 【NCOM054】 x : ComTxModeRepetitionPeriod
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

コンフィギュレーション仕様

ComTxModeRepetitionPeriod に 0 を指定した場合, エラーとする [NCOM054].

3.10.10.4 ComTxModeTimeOffset

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeTrue/ComTxMode/ComTxModeTimeOffset /AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeFalse/ComTxMode/ComTxModeTimeOffset 【COM180_Conf】
概要	周期送信のオフセット時間[秒] (送信モードが PERIODIC または MIXED の場合に有効)

多重度	0..1
型	EcucFloatParamDef
値の範囲	0..3600
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.10.5 ComTxModeTimePeriod

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeTrue/ComTxMode/ComTxModeTimePeriod /AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu/ComTxIPdu/ ComTxModeFalse/ComTxMode/ComTxModeTimePeriod【COM178_Conf】
概要	周期送信の周期時間[秒] (送信モードが PERIODIC または MIXED の場合に有効)
多重度	0..1
型	EcucFloatParamDef
値の範囲	$0 < x \leq 3600$ 【NCOM055】 x : ComTxModeTimePeriod
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

コンフィギュレーション仕様

ComTxModeTimePeriod に 0 を指定した場合, エラーとする [NCOM055].

3.10.11 ComIPduCounter

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduCounter 【COM592_Conf】
概要	PDU カウンタの設定を含むコンテナ(任意)
多重度	0..1
パラメータ	ComIPduCounterThreshold
	ComIPduCounterStartPosition
	ComIPduCounterSize
	ComIPduCounterErrorNotification
サブコンテナ	—

設計上の注意

データシーケンス管理は本バージョン対象外の機能であるため、本コンテナは対象外となる。

3.10.11.1 ComIPduCounterStartPosition

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduCounter/ ComIPduCounterStartPosition 【COM594_Conf】
概要	IPDU カウンタ 開始位置
多重度	1
型	EcucIntegerParamDef
値の範囲	0..2031
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム	

3.10.11.2 ComIPduCounterSize

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduCounter/ ComIPduCounterSize 【COM593_Conf】
概要	IPDU カウンタサイズ[bit]
多重度	1
型	EcucIntegerParamDef
値の範囲	0..8
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム	

3.10.11.3 ComIPduCounterThreshold

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduCounter/ ComIPduCounterThreshold 【COM595_Conf】
概要	IPDU カウンタ 閾値
多重度	0..1
型	EcucIntegerParamDef
値の範囲	0..255
デフォルト値	0
コンフィギュレーションクラス	
プレコンパイルタイム	

3.10.11.4 ComIPduCounterErrorNotification

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduCounter/ ComIPduCounterErrorNotification 【COM003_Conf】
概要	IPDU カウンタエラー通知コールバック関数(Com_CbkCounterErr)の名称
多重度	0..1
型	EcucFunctionNameDef
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.12 ComIPduReplication

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduReplication 【COM599_Conf】
概要	IPDU 複製の設定を含むコンテナ
多重度	0..1
パラメータ	ComIPduReplicationQuorum
	ComIPduReplicaRef
サブコンテナ	—

設計上の注意

通信保護は本バージョン対象外の機能であるため、本コンテナは対象外となる。

3.10.12.1 ComIPduReplicationQuorum

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduReplication/ ComIPduReplicationQuorum 【COM600_Conf】
概要	IPDU 複製数
多重度	1
型	EcucIntegerParamDef
値の範囲	1..3
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム	

3.10.12.2 ComIPduReplicaRef

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu/ComIPduReplication/ ComIPduReplicaRef 【COM601_Conf】
概要	複製する IPDU のレプリカ PDU への参照
多重度	1..2
型	/AUTOSAR/EcuDefs/EcuC/EcuPduCollection/Pdu への参照
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム	

3.10.13 ComSignal

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal 【COM344_Conf】
概要	シグナルの設定を含むコンテナ
多重度	0..* 【NCOM029】
パラメータ	ComHandleId
	ComSignalType
	ComBitPosition
	ComBitSize
	ComSignalLength
	ComUpdateBitPosition
	ComSignalInitValue
	ComSignalDataInvalidValue
	ComSignalEndianness
	ComTransferProperty
	ComFirstTimeout
	ComTimeout
	ComDataInvalidAction
	ComRxDataTimeoutAction
	ComNotification
	ComErrorNotification
	ComInvalidNotification
	ComTimeoutNotification
	ComSystemTemplateSystemSignalRef
サブコンテナ	ComFilter

コンフィギュレーション仕様

COM ジェネレータは、本コンテナのショートネームとシグナル ID(*ComHandleId*) を関連付けるための定数を Com_Cfg.h に生成する 【COM163】。

【COM163】で生成された定数は、Com_SendSignal などのシグナル ID を指定するシステムサービスを呼び出す際に使用できる。

3.10.13.1 ComHandleId

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComHandleId 【COM165_Conf】
概要	シグナルの ID
多重度	1

型	EcucIntegerParamDef
値の範囲	0..65535
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

3.10.13.2 ComSignalType

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignal/ComSignalType 【COM127_Conf】	
概要	シグナルの型	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	BOOLEAN	boolean
	UINT8	符号なし整数(8bit)
	UINT16	符号なし整数(16bit)
	UINT32	符号なし整数(32bit)
	SINT8	符号あり整数(8bit)
	SINT16	符号あり整数(16bit)
	SINT32	符号あり整数(32bit)
	FLOAT32	浮動小数点(32bit)
	FLOAT64	浮動小数点(64bit)
	UINT8_N	配列(8bit)
UINT8_DYN	可変長配列(8bit)	
デフォルト値	ー	
コンフィギュレーションクラス		
プレコンパイルタイム, リンクタイム, ポストビルドタイム		

コンフィギュレーション仕様

可変長データ通信は本バージョン対象外の機能であるため、ComSignalType に UINT8_DYN を指定した場合エラーとする 【NCOM095】.

3.10.13.3 ComBitPosition

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComBitPosition 【COM259_Conf】
概要	シグナルのビット位置
多重度	1
型	EcucIntegerParamDef
値の範囲	0..2031
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

コンフィギュレーション仕様

シグナルの型(*ComSignalType*) が UINT8_N の場合に, ComBitPosition が 8 の倍数でない場合, エラーとする【NCOM019】.

シグナルが IPDU バッファの外に配置されてしまうことを避けるため, COM ジェネレータは以下の場合をエラーとする.

- シグナルの型(*ComSignalType*) が UINT8_N の場合, $\text{ComBitPosition} + (\text{ComSignalLength} * 8)$ が, $(\text{PduLength} * 8)$ より大きい【NCOM015】
- シグナルの型(*ComSignalType*) が UINT8_N 以外, かつリトルエンディアンの場合, $\text{ComBitPosition} + \text{ComBitSize}$ が, $(\text{PduLength} * 8)$ より大きい【NCOM016】
- シグナルの型(*ComSignalType*) が UINT8_N 以外かつビッグエンディアンの場合, $(\text{PduLength} - \text{ComBitPosition} / 8 - 1) * 8 + \text{ComBitPosition} \% 8 + 1$ が, ComBitSize より小さい【NCOM111】

COM ジェネレータは, IPDU バッファに配置されないシグナルまたはシグナルグループをエラーとする【NCOM051】.

表 3-6 シグナルの IPDU 配置位置チェック概要

型 (<i>ComSignalType</i>)	エンディアン (<i>ComSignalEndiannes</i>)	エラー条件
UINT8_N	OPAQUE	$\text{ComBitPosition} + (\text{ComSignalLength} * 8) > (\text{PduLength} * 8)$ 【NCOM015】
UINT8_N 以外	LITTLE_ENDIAN	$\text{ComBitPosition} + \text{ComBitSize} > (\text{PduLength} * 8)$ 【NCOM016】
	BIG_ENDIAN	$(\text{PduLength} - \text{ComBitPosition} / 8 - 1) * 8 + \text{ComBitPosition} \% 8 + 1 < \text{ComBitSize}$ 【NCOM111】

3.10.13.4 ComBitSize

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComBitSize 【COM158_Conf】
概要	シグナルのビットサイズ (シグナル型が UINT8_N または UINT8_DYN の場合は無効)
多重度	0..1
型	EcucIntegerParamDef
値の範囲	1..64
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

コンフィギュレーション仕様

シグナルの型(*ComSignalType*) に指定したデータ型のビットサイズより大きい値をシグナルのサイズ(*ComBitSize*) に指定した場合, エラーとする【NCOM010】[COM785].

例 : *ComSignalType* が UINT8 のシグナルに対して, *ComBitSize* を 10 とした場合, エラー

シグナルの型(*ComSignalType*) が UINT8_N および UINT8_DYN 以外のシグナルに対して, シグナルのサイズ(*ComBitSize*) を指定していない場合, エラーとする【NCOM011】.

シグナルの型(*ComSignalType*) が UINT8, UINT16, UINT32, SINT8, SINT16, SINT32, **UINT8_N** のシグナルに対して, シグナルのサイズ(*ComBitSize*) より大きいビットサイズの値を初期値(*ComSignalInitValue*) または無効値(*ComSignalDataInvalidValue*) に指定した場合, エラーとする【NCOM038】.

例 : *ComSignalType* が SINT8, *ComBitSize* が 7 のシグナルに対して, *ComSignalInitValue* に 85(1010101)を指定した場合, エラー

※1010101 を指定したい場合は, -43(11010101)を指定するか, *ComSignalType* を UINT8 に変更して 85 を指定する

型(*ComSignalType*) が UINT8_N のシグナルに, サイズ(*ComBitSize*) を指定した場合, 警告とする【NCOM021】.

型(*ComSignalType*) が FLOAT32 または FLOAT64 のシグナルに, サイズ(*ComBitSize*) を指定した場合, 警告とする【NCOM022】.

3.10.13.5 ComSignalLength

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComSignalLength 【COM437_Conf】
--------	---

概要	シグナルの長さ[byte] (シグナル型が <code>UINT8_N</code> または <code>UINT8_DYN</code> の場合のみ有効)	
多重度	0..1	
型	EcucIntegerParamDef	
値の範囲	1..8	CAN または LIN 通信する IPDU に所属する場合
	1..254	FlexRay 通信する IPDU に所属する場合
	1..4095	TP 通信する IPDU に所属する場合
デフォルト値	—	
コンフィギュレーションクラス		
プレコンパイルタイム, リンクタイム, ポストビルドタイム		

コンフィギュレーション仕様

型(*ComSignalType*) が UINT8_N のシグナルに, 長さ(*ComSignalLength*) を指定しない場合, または長さ(*ComSignalLength*) に 0 を指定した場合, エラーとする【NCOM012】.

型(*ComSignalType*) が UINT8_N 以外のシグナルに, 長さ(*ComSignalLength*) を指定した場合, 警告とする【NCOM023】.

3.10.13.6 ComUpdateBitPosition

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignal/ComUpdateBitPosition 【COM257_Conf】	
概要	シグナルのアップデートビット位置	
多重度	0..1	
型	EcucIntegerParamDef	
値の範囲	0..63	CAN または LIN 通信する IPDU に所属する場合
	0..2031	FlexRay 通信する IPDU に所属する場合
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム，ポストビルドタイム		

3.10.13.7 ComSignalInitValue

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComSignalInitValue 【COM170_Conf】	
概要	シグナルの初期値	
多重度	0..1	
型	EcucStringParamDef	
値の範囲	シグナルの型ごとに異なる	

デフォルト値	0
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

コンフィギュレーション仕様

シグナルの型(*ComSignalType*) が UINT8_N の場合に, 初期値(*ComSignalInitValue*) に指定した文字列に含まれるデータ数が, シグナルの長さ(*ComSignalLength*) と一致しない場合, エラーとする【NCOM013】.

例 : *ComSignalLength* が 3 であり, *ComSignalInitValue* を”96 97 98 100”とした場合, エラー

3.10.13.8 ComSignalDataInvalidValue

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ ComSignalDataInvalidValue 【COM391_Conf】
概要	シグナルの無効値
多重度	0..1
型	EcucStringParamDef
値の範囲	シグナルの型ごとに異なる
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

コンフィギュレーション仕様

シグナルの型(*ComSignalType*) が UINT8_N の場合に, 無効値(*ComSignalDataInvalidValue*) に指定した文字列に含まれるデータ数が, シグナルの長さ(*ComSignalLength*) と一致しない場合, エラーとする【NCOM014】.

例 : *ComSignalLength* が 3 であり, *ComSignalDataInvalidValue* を”1 2”とした場合, エラー

3.10.13.9 ComSignalEndianness

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComSignalEndianness 【COM157_Conf】	
概要	シグナルのエンディアン	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	LITTLE_ENDIAN	リトルエンディアン
	BIG_ENDIAN	ビッグエンディアン
	OPAQUE	オペーク

デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

コンフィギュレーション仕様

シグナルの型(*ComSignalType*) が UINT8_N および UINT8_DYN 以外であり, かつ, エンディアン (*ComSignalEndianness*) が OPAQUE である場合, エラーとする【NCOM026】.

3.10.13.10 ComTransferProperty

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignal/ComTransferProperty 【COM232_Conf】	
概要	シグナルの送信プロパティ	
多重度	0..1	
型	EcucEnumerationParamDef	
値の範囲	PENDING	シグナルの書き込みが発生しても、所属 IPDU の送信(繰り返しあり)はトリガしない
	TRIGGERED	シグナル書き込みが発生した場合、所属 IPDU の送信(繰り返しあり)をトリガする
	TRIGGERED_WITHOUT_REPETITION	シグナルの書き込みが発生した場合、所属 IPDU の送信(繰り返しなし)をトリガする
	TRIGGERED_ON_CHANGE	シグナルの書き込みが発生し、バッファの値が前回値と異なる場合、所属 IPDU の送信(繰り返しあり)をトリガする
	TRIGGERED_ON_CHANGE_WITHOUT_REPETITION	シグナルの書き込みが発生し、バッファの値が前回値と異なる場合、所属 IPDU の送信(繰り返しなし)をトリガする
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム、リンクタイム、ポストビルドタイム		

コンフィギュレーション仕様

送信シグナルは送信プロパティ(*ComTransferProperty*) に値を指定しなければならない【NCOM042】.

3.10.13.11 ComFirstTimeout

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComFirstTimeout 【COM183_Conf】
概要	シグナルの初回タイムアウト時間[秒] 初回タイムアウト以降(Com_RxIndication コール後かタイムアウト発生後)は ComTimeout 値を使用して DM 監視を行う。
多重度	0..1
型	EcucFloatParamDef
値の範囲	0..3600
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.13.12 ComTimeout

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComTimeout 【COM263_Conf】
概要	シグナルのタイムアウト時間[秒]
多重度	0..1
型	EcucFloatParamDef
値の範囲	0..3600
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.13.13 ComDataInvalidAction

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignal/ComDataInvalidAction 【COM314_Conf】	
概要	シグナルの無効値を受信した場合の処理	
多重度	0..1	
型	EcucEnumerationParamDef	
値の範囲	NOTIFY	無効値受信通知用コールバック関数を呼び出す
	REPLACE	初期値に置き換える
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム，ポストビルドタイム		

コンフィギュレーション仕様

ComDataInvalidAction を NOTIFY に設定し、ComInvalidNotification を設定しない場合、本仕様では何もしない【NCOM109】.

3.10.13.14 ComRxDataTimeoutAction

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignal/ ComRxDataTimeoutAction 【COM412_Conf】	
概要	シグナルの受信タイムアウトが発生した場合の処理	
多重度	0..1	
型	EcucEnumerationParamDef	
値の範囲	NONE	何もしない
	REPLACE	初期値に置き換える
デフォルト値	ー	
コンフィギュレーションクラス		
プレコンパイルタイム, リンクタイム		

3.10.13.15 ComNotification

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignal/ComNotification 【COM498_Conf】
概要	シグナルの送信通知または受信通知のためのコールバック関数名
多重度	0..1
型	EcucFunctionNameDef
値の範囲	－
デフォルト値	－
コンフィギュレーションクラス	
プレコンパイルタイム，リンクタイム	

3.10.13.16 ComErrorNotification

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignal/ComErrorNotification 【COM499_Conf】
概要	シグナルのエラー通知のためのコールバック関数名
多重度	0..1
型	EcucFunctionNameDef
値の範囲	－
デフォルト値	－
コンフィギュレーションクラス	

プレコンパイルタイム, リンクタイム, ポストビルドタイム

3.10.13.17 ComInvalidNotification

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ ComInvalidNotification 【COM315_Conf】
概要	シグナルの無効値受信を通知するためのコールバック関数名
多重度	0..1
型	EcucFunctionNameDef
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.13.18 ComTimeoutNotification

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ ComTimeoutNotification 【COM552_Conf】
概要	シグナルのタイムアウト発生を通知するためのコールバック関数名
多重度	0..1
型	EcucFunctionNameDef
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.13.19 ComSystemTemplateSystemSignalRef

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ ComSystemTemplateSystemSignalRef 【COM002_Conf】
概要	システムテンプレートのシグナルへの参照
多重度	0..1
型	EcucStringParamDef
値の範囲	I-SIGNAL-TO-I-PDU-MAPPING への参照
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.14 ComSignalGroup

コンテナ名	/AUTOSAR/EcuDefs/Com/Config/ComSignalGroup 【COM345_Conf】
概要	シグナルグループの設定を含むコンテナ
多重度	0..*
パラメータ	ComHandleId
	ComUpdateBitPosition
	ComTransferProperty
	ComFirstTimeout
	ComTimeout
	ComDataInvalidAction
	ComRxDataTimeoutAction
	ComNotification
	ComErrorNotification
	ComInvalidNotification
	ComTimeoutNotification
	ComSystemTemplateSignalGroupRef
サブコンテナ	ComGroupSignal

コンフィギュレーション仕様

COM ジェネレータは、本コンテナのショートネームとシグナルグループ ID(*ComHandleId*) を関連付けるための定数を Com_Cfg.h に生成する 【COM044】.

【COM044】で生成された定数は、Com_SendSignalGroup などのシグナルグループ ID を指定するシステムサービスを呼び出す際に使用できる.

3.10.14.1 ComHandleId

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ComHandleId 【COM165_Conf】
概要	シグナルグループの ID
多重度	1
型	EcucIntegerParamDef
値の範囲	0..65535
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

3.10.14.2 ComUpdateBitPosition

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComUpdateBitPosition 【COM257_Conf】	
概要	シグナルグループのアップデートビット位置	
多重度	0..1	
型	EcucIntegerParamDef	
値の範囲	0..63	CAN または LIN 通信する IPDU に所属する場合
	0..2031	FlexRay 通信する IPDU に所属する場合
デフォルト値	ー	
コンフィギュレーションクラス		
プレコンパイルタイム, リンクタイム, ポストビルドタイム		

3.10.14.3 ComTransferProperty

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ ComTransferProperty 【COM232_Conf】	
概要	シグナルグループの送信プロパティ	
多重度	0..1	
型	EcucEnumerationParamDef	
値の範囲	PENDING	シグナルの書き込みが発生しても、所属 IPDU の送信(繰り返しあり)をトリガしない
	TRIGGERED	シグナルの書き込みが発生した場合、所属 IPDU の送信(繰り返しあり)をトリガする
	TRIGGERED_WITHOUT_REPETITION	シグナルの書き込みが発生した場合、所属 IPDU の送信(繰り返しなし)をトリガする
	TRIGGERED_ON_CHANGE	シグナルの書き込みが発生し、バッファの値が前回値と異なる場合、所属 IPDU の送信(繰り返しあり)をトリガする
	TRIGGERED_ON_CHANGE_WITHOUT_REPETITION	シグナルの書き込みが発生し、バッファの値が前回値と異なる場合、所属 IPDU の送信(繰り返しなし)をトリガする
デフォルト値	ー	
コンフィギュレーションクラス		
プレコンパイルタイム、リンクタイム、ポストビルドタイム		

コンフィギュレーション仕様

送信シグナルは送信プロパティ (*ComTransferProperty*) に値を指定しなければならない【NCOM042】.

3.10.14.4 ComFirstTimeout

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComFirstTimeout 【COM183_Conf】
概要	シグナルグループの初回タイムアウト時間[秒] 初回タイムアウト以降(Com_RxIndication コール後かタイムアウト発生後) は ComTimeout 値を使用して DM 監視を行う。
多重度	0..1
型	EcucFloatParamDef
値の範囲	0..3600
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.14.5 ComTimeout

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ComTimeout 【COM263_Conf】
概要	シグナルグループのタイムアウト時間[秒]
多重度	0..1
型	EcucFloatParamDef
値の範囲	0..3600
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.14.6 ComDataInvalidAction

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ ComDataInvalidAction 【COM314_Conf】	
概要	シグナルグループの無効値を受信した場合の処理	
多重度	0..1	
型	EcucEnumerationParamDef	
値の範囲	NOTIFY	無効値受信通知用コールバック関数を呼び出す
	REPLACE	初期値に置き換える
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム，ポストビルドタイム		

3.10.14.7 ComRxDataTimeoutAction

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ ComRxDataTimeoutAction 【COM412_Conf】	
概要	シグナルの受信タイムアウトが発生した場合の処理	
多重度	0..1	
型	EcucEnumerationParamDef	
値の範囲	NONE	何もしない
	REPLACE	初期値に置き換える
デフォルト値	ー	
コンフィギュレーションクラス		
プレコンパイルタイム， リンクタイム		

3.10.14.8 ComNotification

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ComNotification 【COM498_Conf】
概要	シグナルグループの送信通知または受信通知のためのコールバック関数名
多重度	0..1
型	EcucFunctionNameDef
値の範囲	－
デフォルト値	－
コンフィギュレーションクラス	
プレコンパイルタイム，リンクタイム	

3.10.14.9 ComErrorNotification

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ ComErrorNotification 【COM499_Conf】
概要	シグナルグループのエラー通知のためのコールバック関数名
多重度	0..1
型	EcucFunctionNameDef
値の範囲	－
デフォルト値	－
コンフィギュレーションクラス	
プレコンパイルタイム，リンクタイム，ポストビルドタイム	

3.10.14.10 ComInvalidNotification

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/	
--------	--	--

	ComInvalidNotification 【COM315_Conf】
概要	シグナルグループの無効値受信を通知するためのコールバック関数名
多重度	0..1
型	EcucFunctionNameDef
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.14.11 ComTimeoutNotification

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ ComTimeoutNotification 【COM552_Conf】
概要	シグナルグループのタイムアウト発生を通知するためのコールバック関数名
多重度	0..1
型	EcucFunctionNameDef
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.14.12 ComSystemTemplateSignalGroupRef

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ ComSystemTemplateSignalGroupRef 【COM001_Conf】
概要	システムテンプレートのシグナルグループへの参照
多重度	0..1
型	EcucStringParamDef
値の範囲	I-SIGNAL-TO-I-PDU-MAPPING への参照
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.15 ComGroupSignal

コンテナ名	/AUTOSAR/EcuDefs/Com/Config/ComSignalGroup/ComGroupSignal 【COM520_Conf】
概要	グループシグナルの設定を含むコンテナ
多重度	1..*
パラメータ	ComHandleId
	ComSignalType
	ComBitPosition
	ComBitSize
	ComSignalLength
	ComSignalInitValue
	ComSignalDataInvalidValue
	ComSignalEndianness
	ComTransferProperty
	ComSystemTemplateSystemSignalRef
サブコンテナ	ComFilter

コンフィギュレーション仕様

COM ジェネレータは、本コンテナのショートネームとグループシグナル ID(*ComHandleId*) を関連付けるための定数を Com_Cfg.h に生成する【COM521】。

【COM521】で生成された定数は、Com_SendSignal などのグループシグナル ID を指定するシステムサービスを呼び出す際に使用できる。

3.10.15.1 ComHandleId

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComHandleId 【COM165_Conf】
概要	グループシグナルの ID
多重度	1
型	EcucIntegerParamDef
値の範囲	0..65535
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

3.10.15.2 ComSignalType

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/
--------	--

	ComGroupSignal/ComSignalType 【COM127_Conf】	
概要	グループシグナルの型	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	BOOLEAN	boolean
	UINT8	符号なし整数(8bit)
	UINT16	符号なし整数(16bit)
	UINT32	符号なし整数(32bit)
	SINT8	符号あり整数(8bit)
	SINT16	符号あり整数(16bit)
	SINT32	符号あり整数(32bit)
	FLOAT32	浮動小数点(32bit)
	FLOAT64	浮動小数点(64bit)
	UINT8_N	配列(8bit)
UINT8_DYN	可変長配列(8bit)	
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム，ポストビルドタイム		

3.10.15.3 ComBitPosition

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComBitPosition 【COM259_Conf】
概要	グループシグナルのビット位置
多重度	1
型	EcucIntegerParamDef
値の範囲	0..2031
デフォルト値	ー
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.15.4 ComBitSize

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComBitSize 【COM158_Conf】	
概要	グループシグナルのビットサイズ (グループシグナルの型が UINT8_N または UINT8_DYN の場合は無効)	
多重度	0..1	

型	EcucIntegerParamDef
値の範囲	1..64
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.15.5 ComSignalLength

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComSignalLength【COM437_Conf】	
概要	グループシグナルの長さ[byte] (グループシグナルの型が UINT8_N または UINT8_DYN の場合のみ有効)	
多重度	0..1	
型	EcucIntegerParamDef	
値の範囲	1..8	CAN または LIN 通信する IPDU に所属する場合
	1..254	FlexRay 通信する IPDU に所属する場合
	1..4095	TP 通信する IPDU に所属する場合
デフォルト値	ー	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム，ポストビルドタイム		

コンフィギュレーション仕様

型(*ComSignalType*) が UINT8_N のシグナルに, 長さ(*ComSignalLength*) を指定しない場合, または長さ(*ComSignalLength*) に 0 を指定した場合, エラーとする 【NCOM012】.

3.10.15.6 ComSignalInitValue

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComSignalInitValue 【COM170_Conf】
概要	グループシグナルの初期値
多重度	0..1
型	EcucStringParamDef
値の範囲	グループシグナルの型ごとに異なる
デフォルト値	0
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.15.7 ComSignalDataInvalidValue

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComSignalDataInvalidValue 【COM391_Conf】
概要	グループシグナルの無効値
多重度	0..1
型	EcucStringParamDef
値の範囲	グループシグナルの型ごとに異なる
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.15.8 ComSignalEndianness

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComSignalEndianness 【COM157_Conf】	
概要	グループシグナルのエンディアン	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	LITTLE_ENDIAN	リトルエンディアン
	BIG_ENDIAN	ビッグエンディアン
	OPAQUE	オペーク
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム，ポストビルドタイム		

3.10.15.9 ComTransferProperty

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComTransferProperty 【COM560_Conf】	
概要	グループシグナルの送信プロパティ	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	PENDING	グループシグナルの書き込みが発生しても, 所属しているシグナルグループの送信プロパティに影響を与えない
	TRIGGERED_ON_CHANGE	グループシグナルの書き込みが発生し, バッファの値が前回値と異なる場合, 所属しているシグナルグループの送信プロパティに影響を与える

デフォルト値	－
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.15.10 ComSystemTemplateSystemSignalRef

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComSystemTemplateSystemSignalRef【COM002_Conf】
概要	システムテンプレートのグループシグナルへの参照
多重度	0..1
型	EcucStringParamDef
値の範囲	I-SIGNAL-TO-I-PDU-MAPPING への参照
デフォルト値	－
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.16 ComFilter

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComFilter /AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComFilter 【COM339_Conf】
概要	フィルタの設定を含むコンテナ
多重度	0..1
パラメータ	ComFilterAlgorithm
	ComFilterMask
	ComFilterMin
	ComFilterMax
	ComFilterOffset
	ComFilterPeriod
サブコンテナ	ComFilterX
	—

3.10.16.1 ComFilterAlgorithm

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComFilter/ ComFilterAlgorithm /AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComFilter/ComFilterAlgorithm 【COM146_Conf】	
概要	フィルタのアルゴリズム ・ 今回値 : フィルタ対象のシグナル値, ・ 前回値 : 前回フィルタを通過したシグナル値 ・ X の値 : ComFilterX に設定した値	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	ALWAYS	フィルタ結果は常に TRUE
	NEVER	フィルタ結果は常に FALSE
	MASKED_NEW_ DIFFERS_MASKED_ _OLD	マスクした今回値とマスクした前回値を比較 ・ 同じ場合 : FALSE ・ 異なる場合 : TRUE
	MASKED_NEW_ DIFFERS_X	マスクした今回値と X の値を比較 ・ 同じ場合 : FALSE ・ 異なる場合 : TRUE
	MASKED_NEW_ EQUALS_X	マスクした今回値と X の値を比較 ・ 同じ場合 : TRUE

		・異なる場合：FALSE
	NEW_IS_OUTSIDE	今回値が範囲外である場合，TRUE
	NEW_IS_WITHIN	今回値が範囲内である場合，TRUE
	ONE_EVERY_N	N 回フィルタされる度に 1 回のみ TRUE
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム		

コンフィギュレーション仕様

ComFilterAlgorithm が MASKED_NEW_DIFFERS_MASKED_OLD か, MASKED_NEW_DIFFERS_X か MASKED_NEW_EQUALS_X に設定されている場合, ComFilterMask を設定していなければエラーとする【NCOM105】.

ComFilterAlgorithm が MASKED_NEW_DIFFERS_X か MASKED_NEW_EQUALS_X に設定されている場合, ComFilterX を設定していなければエラーとする【NCOM106】.

ComFilterAlgorithm が ONE_EVERY_N に設定されている場合, ComFilterOffset と ComFilterPeriod を設定していなければエラーとする【NCOM107】.

ComFilterAlgorithm が NEW_IS_OUTSIDE か NEW_IS_WITHIN に設定されている場合, ComFilterMin と ComFilterMax を設定していなければエラーとする【NCOM108】.

3.10.16.2 ComFilterMask

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComFilter/ ComFilterMask /AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComFilter/ComFilterMask【COM235_Conf】
概要	フィルタのマスク値 (フィルタアルゴリズム <i>ComFilterAlgorithm</i> が MASKED_NEW_DIFFERS_MASKED_OLD または MASKED_NEW_DIFFERS_X または MASKED_NEW_EQUALS_X の場 合, 有効となる)
多重度	0..1
型	EcucIntegerParamDef
値の範囲	-2147483648..4294967295【NCOM047】
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

AUTOSAR 仕様との違い

AUTOSAR 仕様では、ComFilterMask の値の範囲が 0..18446744073709551615 と規定されているが、フィルタアルゴリズム(*ComFilterAlgorithm*) が MASKED_NEW_DIFFERS_MASKED_OLD または MASKED_NEW_DIFFERS_X または MASKED_NEW_EQUALS_X のフィルタを持つシグナルが符号ありの型である可能性もある。そして、浮動小数点型のシグナルに対するフィルタリングはサポートされないため、64 ビットの範囲を考慮する必要がない。したがって、本仕様では ComFilterMask の値の範囲を-2147483648..4294967295 とする [NCOM047]。

3.10.16.3 ComFilterX

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComFilter/ComFilterX /AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComFilter/ComFilterX 【COM147_Conf】
概要	フィルタの X の値 (フィルタアルゴリズム <i>ComFilterAlgorithm</i> が MASKED_NEW_DIFFERS_X または MASKED_NEW_EQUALS_X の場 合、有効となる)
多重度	0..1
型	EcucIntegerParamDef
値の範囲	-2147483648..4294967295 【NCOM048】
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

AUTOSAR 仕様との違い

AUTOSAR 仕様では、ComFilterX の値の範囲が 0..18446744073709551615 と規定されているが、フィルタアルゴリズム(*ComFilterAlgorithm*) が MASKED_NEW_DIFFERS_X または MASKED_NEW_EQUALS_X のフィルタを持つシグナルが符号ありの型である可能性もある。そして、浮動小数点型のシグナルに対するフィルタリングはサポートされないため、64 ビットの範囲を考慮する必要がない。したがって、本仕様では ComFilterX の値の範囲を-2147483648..4294967295 とする [NCOM048]。

3.10.16.4 ComFilterMin

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComFilter/ ComFilterMin /AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComFilter/ComFilterMin 【COM318_Conf】
--------	--

概要	フィルタを通過できる最小値 (フィルタアルゴリズム <i>ComFilterAlgorithm</i> が NEW_IS_OUTSIDE または NEW_IS_WITHIN の場合、有効となる)
多重度	0..1
型	EcucIntegerParamDef
値の範囲	-2147483648..4294967295 【NCOM043】
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

AUTOSAR 仕様との違い

AUTOSAR 仕様では, ComFilterMin の値の範囲が 0..18446744073709551615 と規定されているが, フィルタアルゴリズム(*ComFilterAlgorithm*) が NEW_IS_OUTSIDE または NEW_IS_WITHIN のフィルタを持つシグナルが符号ありの型である可能性もある. そして, 浮動小数点型のシグナルに対するフィルタリングはサポートされないため, 64 ビットの範囲を考慮する必要がない. したがって, 本仕様では ComFilterMin の値の範囲を -2147483648..4294967295 とする [NCOM043].

3.10.16.5 ComFilterMax

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComSignal/ComFilter/ ComFilterMax /AUTOSAR/EcucDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComFilter/ComFilterMax 【COM317_Conf】
概要	フィルタを通過できる最大値 (フィルタアルゴリズム <i>ComFilterAlgorithm</i> が NEW_IS_OUTSIDE または NEW_IS_WITHIN の場合、有効となる)
多重度	0..1
型	EcucEnumerationParamDef
値の範囲	-2147483648..4294967295 【NCOM044】
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

AUTOSAR 仕様との違い

AUTOSAR 仕様では, ComFilterMax の値の範囲が 0..18446744073709551615 と規定されているが, フィルタアルゴリズム(*ComFilterAlgorithm*) が NEW_IS_OUTSIDE または NEW_IS_WITHIN のフィルタを持つシグナルが符号ありの型である可能性もある. そして, 浮動小数点型のシグナルに対する

フィルタリングはサポートされないため、64 ビットの範囲を考慮する必要がない。したがって、本仕様では ComFilterMax の値の範囲を-2147483648..4294967295 とする [NCOM044]。

3.10.16.6 ComFilterOffset

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComFilter/ ComFilterOffset /AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComFilter/ComFilterOffset 【COM313_Conf】
概要	フィルタのオフセット (フィルタアルゴリズム <i>ComFilterAlgorithm</i> が ONE_EVERY_N の場合, 有効となる)
多重度	0..1
型	EcucIntegerParamDef
値の範囲	0..18446744073709551615
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

3.10.16.7 ComFilterPeriod

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal/ComFilter/ ComFilterPeriod /AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup/ ComGroupSignal/ComFilter/ComFilterPeriod 【COM312_Conf】
概要	フィルタの通過周期 (フィルタアルゴリズム <i>ComFilterAlgorithm</i> が ONE_EVERY_N の場合, 有効となる)
多重度	0..1
型	EcucIntegerParamDef
値の範囲	0..18446744073709551615
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム	

3.10.17 ComGwMapping

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping 【COM544_Conf】
概要	シグナルゲートウェイの設定を含むコンテナ
多重度	0..*
パラメータ	—
サブコンテナ	ComGwSource
	ComGwDestination

設計上の注意

シグナルゲートウェイは本バージョン対象外の機能であるため、本コンテナおよびサブコンテナは対象外となる。

3.10.18 ComGwSource

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ComGwSource 【COM545_Conf】
概要	ゲートウェイ元シグナルの設定を含むコンテナ
多重度	1
パラメータ	—
サブコンテナ	ComGwSourceDescription
	ComGwSignal

3.10.19 ComGwDestination

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ ComGwDestination 【COM546_Conf】
概要	ゲートウェイ先シグナルの設定を含むコンテナ
多重度	1..*
パラメータ	—
サブコンテナ	ComGwDestinationDescription
	ComGwSignal

3.10.20 ComGwSourceDescription

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription 【COM548_Conf】
概要	ゲートウェイ元シグナルの記述の設定を含むコンテナ
多重度	0..1
パラメータ	ComSignalType
	ComBitPosition
	ComBitSize
	ComSignalLength
	ComUpdateBitPosition
	ComSignalEndianness
	ComGwIPduRef
サブコンテナ	—

3.10.20.1 ComSignalType

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription/ComSignalType 【COM127_Conf】	
概要	ゲートウェイ元シグナルの型	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	BOOLEAN	boolean
	UINT8	符号なし整数(8bit)
	UINT16	符号なし整数(16bit)
	UINT32	符号なし整数(32bit)
	SINT8	符号あり整数(8bit)
	SINT16	符号あり整数(16bit)
	SINT32	符号あり整数(32bit)
	FLOAT32	浮動小数点(32bit)
	FLOAT64	浮動小数点(64bit)
	UINT8_N	配列(8bit)
UINT8_DYN	可変長配列(8bit)	
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム，リンクタイム，ポストビルドタイム		

3.10.20.2 ComBitPosition

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription/ComBitPosition 【COM259_Conf】
概要	ゲートウェイ元シグナルのビット位置
多重度	1
型	EcucIntegerParamDef
値の範囲	0..2031
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.20.3 ComBitSize

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription/ComBitSize 【COM158_Conf】
概要	ゲートウェイ元シグナルのビットサイズ (シグナル型が UINT8_N または UINT8_DYN の場合は無効)
多重度	0..1
型	EcucIntegerParamDef
値の範囲	0..64
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.20.4 ComSignalLength

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription/ComSignalLength 【COM437_Conf】	
概要	ゲートウェイ元シグナルの長さ[byte] (シグナル型が UINT8_N または UINT8_DYN の場合のみ有効)	
多重度	0..1	
型	EcucIntegerParamDef	
値の範囲	0..8	CAN または LIN 通信する IPDU に所属する場合
	0..254	FlexRay 通信する IPDU に所属する場合
	0..4095	TP 通信する IPDU に所属する場合
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム, リンクタイム, ポストビルドタイム		

3.10.20.5 ComUpdateBitPosition

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription/ComUpdateBitPosition 【COM257_Conf】	
概要	ゲートウェイ元シグナルのアップデートビット位置	
多重度	0..1	
型	EcucIntegerParamDef	
値の範囲	0..63	CAN または LIN 通信する IPDU に所属する場合
	0..2031	FlexRay 通信する IPDU に所属する場合
デフォルト値	ー	
コンフィギュレーションクラス		
プレコンパイルタイム, リンクタイム, ポストビルドタイム		

3.10.20.6 ComSignalEndianness

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription /ComSignalEndianness 【COM157_Conf】		
概要	ゲートウェイ元シグナルのエンディアン		
多重度	1		
型	EcucEnumerationParamDef		
値の範囲	LITTLE_ENDIAN	リトルエンディアン	
	BIG_ENDIAN	ビッグエンディアン	
	OPAQUE	オペーク	
デフォルト値	ー		
コンフィギュレーションクラス			
プレコンパイルタイム，リンクタイム，ポストビルドタイム			

3.10.20.7 ComGwIPduRef

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription /ComGwIPduRef 【COM550_Conf】
概要	ゲートウェイ元のシグナルの所属 IPDU への参照
多重度	1
型	/AUTOSAR/EcucDefs/Com/ComConfig/ComIPdu への参照
値の範囲	－
デフォルト値	－
コンフィギュレーションクラス	
プレコンパイルタイム，リンクタイム，ポストビルドタイム	

3.10.21 ComGwDestinationDesctiption

コンテナ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription 【COM549_Conf】
概要	ゲートウェイ先シグナルの記述の設定を含むコンテナ
多重度	0..1
パラメータ	ComBitPosition
	ComUpdateBitPosition
	ComSignalInitValue
	ComSignalEndianness
	ComTransferProperty
	ComGwIPduRef
サブコンテナ	ComFilter

3.10.21.1 ComBitPosition

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComBitPosition 【COM259_Conf】
概要	ゲートウェイ先シグナルのビット位置
多重度	1
型	EcucIntegerParamDef
値の範囲	0..2031
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.21.2 ComUpdateBitPosition

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ ComGwDestination/ComGwDestinationDescription/ ComUpdateBitPosition 【COM257_Conf】	
概要	ゲートウェイ先シグナルのアップデートビット位置	
多重度	0..1	
型	EcucIntegerParamDef	
値の範囲	0..63	CAN または LIN 通信する IPDU に所属する場合
	0..2031	FlexRay 通信する IPDU に所属する場合
デフォルト値	ー	
コンフィギュレーションクラス		
プレコンパイルタイム, リンクタイム, ポストビルドタイム		

3.10.21.3 ComSignalInitValue

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComGwMapping/ ComGwDestination/ComGwDestinationDescription/ComSignalInitValue 【COM170_Conf】
概要	ゲートウェイ先シグナルの初期値
多重度	0..1
型	EcucStringParamDef
値の範囲	シグナルの型ごとに異なる
デフォルト値	0
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

3.10.21.4 ComSignalEndianness

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ ComGwDestination/ComGwDestinationDescription/ ComSignalEndianness 【COM157_Conf】	
概要	ゲートウェイ先シグナルのエンディアン	
多重度	1	
型	EcucEnumerationParamDef	
値の範囲	LITTLE_ENDIAN	リトルエンディアン
	BIG_ENDIAN	ビッグエンディアン
	OPAQUE	オペーク
デフォルト値	—	

コンフィギュレーションクラス
プレコンパイルタイム, リンクタイム, ポストビルドタイム

3.10.21.5 ComTransferProperty

パラメータ名	/AUTOSAR/EcucDefs/Com/ComConfig/ComGwMapping/ ComGwDestination/ComGwDestinationDescription/ ComTransferProperty 【COM232_Conf】	
概要	ゲートウェイ先シグナルの送信プロパティ	
多重度	0..1	
型	EcucEnumerationParamDef	
値の範囲	PENDING	シグナルの書き込みが発生しても、所属 IPDU の送信(繰り返しあり)はトリガしない
	TRIGGERED	シグナル書き込みが発生した場合、所属 IPDU の送信(繰り返しあり)をトリガする
	TRIGGERED_WITH OUT_REPETITION	シグナルの書き込みが発生した場合、所属 IPDU の送信(繰り返しなし)をトリガする
	TRIGGERED_ON_C HANGE	シグナルの書き込みが発生し、バッファの値が前回値と異なる場合、所属 IPDU の送信(繰り返しあり)をトリガする
	TRIGGERED_ON_C HANGE_WITHOUT _REPETITION	シグナルの書き込みが発生し、バッファの値が前回値と異なる場合、所属 IPDU の送信(繰り返しなし)をトリガする
デフォルト値	－	
コンフィギュレーションクラス		
プレコンパイルタイム、リンクタイム、ポストビルドタイム		

3.10.21.6 ComGwIPduRef

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ ComGwDestination/ComGwDestinationDescription/ComGwIPduRef 【COM550_Conf】
概要	ゲートウェイ先シグナルの所属 IPDU への参照
多重度	1
型	/AUTOSAR/EcuDefs/Com/ComConfig/ComIPdu への参照
値の範囲	－
デフォルト値	－
コンフィギュレーションクラス	

プレコンパイルタイム, リンクタイム, ポストビルドタイム

3.10.22 ComGwSignal

コンテナ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ComGwSource/ ComGwSignal /AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ ComGwDestination/ComGwSignal 【COM551_Conf】
概要	ゲートウェイするシグナルへの参照を含むコンテナ
多重度	0..1
パラメータ	ComGwSignalRef
サブコンテナ	—

3.10.22.1 ComGwSignalRef

パラメータ名	/AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ComGwSource/ ComGwSignal/ComGwSignalRef /AUTOSAR/EcuDefs/Com/ComConfig/ComGwMapping/ ComGwDestination/ComGwSignal/ComGwSignalRef 【COM547_Conf】
概要	ゲートウェイ対象シグナルへの参照
多重度	1
型	/AUTOSAR/EcuDefs/Com/ComConfig/ComSignal への参照 または /AUTOSAR/EcuDefs/Com/ComConfig/ComSignalGroup への参照 または /AUTOSAR/EcuDefs/Com/ComConfig/ComGroupSignal への参照
値の範囲	—
デフォルト値	—
コンフィギュレーションクラス	
プレコンパイルタイム, リンクタイム, ポストビルドタイム	

4 リファレンス

4.1 データ型一覧

表 4-1 データ型一覧

データ型名	概要
Com_StatusType	COM 状態型.
Com_SignalIdType	シグナル ID 型
Com_SignalGroupIdType	シグナルグループ ID 型
Com_IpduGroupIdType	IPDU グループ ID 型
Com_IpduGroupVector	IPDU グループベクタ型
Com_ServiceIdType	サービス ID 型
Com_ConfigType	COM のコンフィギュレーション型

4.2 定数一覧

表 4-2 定数一覧

定数名	概要
SEND	送信側であることを示す定数
RECEIVE	受信側であることを示す定数
NORMAL	通常通信を行う IPDU であることを示す定数
TP	TP 通信を行う IPDU であることを示す定数
BOOLEAN	ブーリアン型であることを示す定数
UINT8	8 ビットの符号なし整数型であることを示す定数
UINT16	16 ビットの符号なし整数型であることを示す定数
UINT32	32 ビットの符号なし整数型であることを示す定数
SINT8	8 ビットの符号あり整数型であることを示す定数
SINT16	16 ビットの符号あり整数型であることを示す定数
SINT32	32 ビットの符号あり整数型であることを示す定数
UINT8_N	8 ビットの符号なし整数の配列型であることを示す定数
UINT8_DYN	8 ビットの符号なし整数の可変長配列型であることを示す定数
FLOAT32	32 ビットの浮動小数点型であることを示す定数
FLOAT64	64 ビットの浮動小数点型であることを示す定数
BIG_ENDIAN	ビッグエンディアンであることを示す定数
LITTLE_ENDIAN	リトルエンディアンであることを示す定数
OPAQUE	オpaqueであることを示す定数
ALWAYS	常に通過するフィルタであることを示す定数

定数名	概要
NEVER	常に通過しないフィルタであることを示す定数
MASKED_NEW_DIFFERS_MASKED_OLD	マスクした今回値とマスクした前回値を比較して、値が異なる場合のみ通過するフィルタであることを示す定数
MASKED_NEW_DIFFERS_X	マスクした今回値と X の値を比較して、値が異なる場合のみ通過するフィルタであることを示す定数
MASKED_NEW_EQUALS_X	マスクした今回値と X の値を比較して、値が一致する場合のみ通過するフィルタであることを示す定数
NEW_IS_OUTSIDE	今回値が指定された範囲外の値である場合のみ通過するフィルタであることを示す定数
NEW_IS_WITHIN	今回値が指定された範囲内の値である場合のみ通過するフィルタであることを示す定数
ONE_EVERY_N	N 回フィルタリングされる度に 1 回のみ通過するフィルタであることを示す定数
DIRECT	直接(N 回)送信であることを示す定数
PERIODIC	周期送信であることを示す定数
MIXED	直接(N 回)送信と周期送信を並行して行うことを示す定数
PENDING	送信トリガではないことを示す定数
TRIGGERED	送信トリガであることを示す定数
TRIGGERED_ON_CHANGE	値の変更がある場合のみ送信トリガとなることを示す定数
TRIGGERED_ON_CHANGE_WITHOUT_REPETITION	値の変更がある場合のみ繰り返しなしの送信トリガとなることを示す定数
TRIGGERED_WITHOUT_REPETITION	繰り返しなしの送信トリガとなることを示す定数
DEFERRED	通信完了通知を周期関数で通知することを示す定数
IMMEDIATE	通信完了通知を即時に通知することを示す定数
Confirmation	送信完了通知を受け取った際にアップデートビットをクリアすることを示す定数
Transmit	PduR_ComTransmit を呼び出した際にアップデートビットをクリアすることを示す定数
TriggerTransmit	Com_TriggerTransmit により IPDU がフェッチされた際にアップデートビットをクリアすることを示す定数
NOTIFY	コールバック関数を呼び出して RTE へ通知することを示す定数
REPLACE	バッファの値を初期値に置き換えることを示す定数
NONE	特に何もしないことを示す定数
COM_SERVICE_NOT_AVAIL	システムサービスが利用不可であることを示す定数

定数名	概要
LABLE	
COM_BUSY	リソース使用中によるシステムサービス異常終了であることを示す定数
COM_INVALID_UINT8	8 ビットの符号なし整数型のための無効値(0xFF)
COM_INVALID_UINT16	16 ビットの符号なし整数型のための無効値(0xFFFF)
COM_INVALID_UINT32	32 ビットの符号なし整数型のための無効値(0xFFFFFFFF)

4.3 システムサービス一覧

表 4-3 システムサービス一覧

システムサービス名	概要
Com_Init	初期化
Com_DeInit	未初期化
Com_GetStatus	状態取得
Com_GetConfigurationId	現在有効なコンフィギュレーション情報の ID 取得
Com_GetVersionInfo	COM のバージョン情報取得
Com_IpduGroupControl	IPDU グループ管理
Com_ReceptionDMControl	受信 DM 管理
Com_SetIpduGroup	IPDU グループベクタ設定
Com_ClearIpduGroupVector	IPDU グループベクタクリア
Com_SendSignal	シグナル送信
Com_SendDynSignal	可変長シグナル送信
Com_SendSignalGroup	シグナルグループ送信
Com_ReceiveSignal	シグナル受信
Com_ReceiveDynSignal	可変長シグナル受信
Com_ReceiveSignalGroup	シグナルグループ受信
Com_InvalidateSignal	シグナル無効化
Com_InvalidateSignalGroup	シグナルグループ無効化
Com_TriggerIPDUSend	IPDU トリガ送信
Com_SwitchIpduTxMode	IPDU 送信モード変更

4.4 コールバック関数一覧

表 4-4 コールバック関数一覧

コールバック関数名	概要
Com_TxConfirmation	送信完了通知
Com_TpTxConfirmation	TP 送信完了通知
Com_RxIndication	受信完了通知
Com_TpRxIndication	TP 受信完了通知
Com_TriggerTransmit	トリガ送信
Com_StartOfReception	受信開始
Com_CopyTxData	送信データコピー
Com_CopyRxData	受信データコピー

4.5 周期関数一覧

表 4-5 周期関数一覧

周期関数名	概要
Com_MainFunctionTx	送信処理を行う周期関数
Com_MainFunctionRx	受信処理を行う周期関数
Com_MainFunctionRouteSignals	ルーティング処理を行う周期関数

4.6 エラーコード一覧

表 4-6 エラーコード一覧

エラーコード名	概要
COM_E_PARAM	引数に不正な値が渡された
COM_E_UNINIT	初期化が未実施である
COM_E_PARAM_POINTER	引数に NULL ポインタが渡された

4.7 仕様番号一覧

4.7.1 削除した AUTOSAR 仕様一覧

仕様タグ	削除理由
<COM199>	非推奨なシステムサービス(Com_UpdateShadowSignal)である
<COM632>	非推奨なシステムサービス(Com_UpdateShadowSignal)である
<COM633>	非推奨なシステムサービス(Com_UpdateShadowSignal)である
<COM634>	非推奨なシステムサービス(Com_UpdateShadowSignal)である
<COM202>	非推奨なシステムサービス(Com_ReceiveShadowSignal)である
<COM640>	非推奨なシステムサービス(Com_ReceiveShadowSignal)である
<COM641>	非推奨なシステムサービス(Com_ReceiveShadowSignal)である
<COM286>	非推奨なシステムサービス(Com_InvalidateShadowSignal)である
<COM288>	非推奨なシステムサービス(Com_InvalidateShadowSignal)である
<COM644>	非推奨なシステムサービス(Com_InvalidateShadowSignal)である
<COM489>	送信側と受信側の無効値が一致していることはユーザが保証する必要がある
<COM671>	コールバック関数の実装方法に関する説明であり，要求仕様ではない
<COM673>	COM のバージョンチェックはユーザが行う(※R4.1.1 で削除される)
<COM762>	ComBitSize の設定可能範囲を 1..64 とする
<COM764>	ComBitSize の設定可能範囲を 1..64 とする
<COM999>	要求仕様ではない

4.7.2 改変した AUTOSAR 仕様一覧

仕様タグ	改変内容
[NCOM002]	ComConfig の多重度変更 [COM337_Conf]
[NCOM003]	ComIPduHandleId の多重度変更 [COM175_Conf]
[NCOM004]	ComIPduGroupRef の多重度変更 [COM206_Conf]
[NCOM009]	IPDU グループのネスト所属は非サポート [COM185_Conf]
[NCOM025]	PDU 関連のデータ型の定義規則 [COM220]
[NCOM027]	ComIPdu の多重度変更 [COM340_Conf]
[NCOM028]	ComIPduGroup の多重度変更 [COM341_Conf]
[NCOM029]	ComSignal の多重度変更 [COM344_Conf]
[NCOM043]	ComFilterMin の値の範囲変更 [COM318_Conf]
[NCOM044]	ComFilterMax の値の範囲変更 [COM317_Conf]
[NCOM047]	ComFilterMask の値の範囲変更 [COM235_Conf]
[NCOM048]	ComFilterX の値の範囲変更 [COM147_Conf]
[NCOM100]	送信プロパティの依らず送信 DM を行う [COM481]

4.7.3 追加した NCES 仕様一覧

仕様タグ	概要
[NCOM001]	初期化前に Com_GetConfigurationId が呼び出された場合の返り値
[NCOM005]	周期関数の呼出し回数の算出方法
[NCOM006]	PduR_ComTransmit 呼出しタイミング
[NCOM007]	COM ジェネレータでのエラー検出
[NCOM008]	ID 系パラメータは 0 から連番
[NCOM010]	ComSignalType と ComBitSize の整合チェック
[NCOM011]	ComSignalType と ComBitSize の整合チェック
[NCOM012]	ComSignalType と ComSignalLength の整合チェック
[NCOM013]	ComSignalInitValue と ComSignalLength の整合チェック
[NCOM014]	ComSignalDataInvalidValue と ComSignalLength の整合チェック
[NCOM015]	シグナル配置のチェック
[NCOM016]	シグナル配置のチェック
[NCOM019]	ComSignalType と ComBitPosition の整合チェック
[NCOM020]	周期送信のオフセット時間算出方法
[NCOM021]	ComSignalType と ComBitSize の整合チェック
[NCOM022]	ComSignalType と ComBitSize の整合チェック
[NCOM023]	ComSignalType と ComSignalLength の整合チェック
[NCOM024]	コンフィギュレーションデータの変数名の定義規則
[NCOM026]	ComSignalType と ComSignalEndianness の整合チェック
[NCOM030]	ComTxIPduClearUpdateBit のデフォルト値
[NCOM031]	DM タイマのリセットタイミング
[NCOM032]	COM システムサービスで用いる排他エリア
[NCOM033]	タイムアウト検出タイミング
[NCOM034]	タイムアウト検出タイミング
[NCOM035]	タイムアウト検出タイミング
[NCOM038]	ComBitSize と ComSignalInitValue または ComSignalDataInvalidValue の整合性チェック
[NCOM039]	NULL ポインタ指定時の振舞い
[NCOM040]	NULL ポインタ指定時の振舞い
[NCOM041]	ComTxModeTrue および ComTxModeFalse の整合性チェック
[NCOM042]	送信シグナルには ComTransferProperty を指定しなければならない
[NCOM045]	ComFilterMin および ComFilterMax の整合性チェック
[NCOM046]	ComFilterMin および ComFilterMax の整合性チェック
[NCOM049]	ComTxTimeBase 制限事項

仕様タグ	概要
[NCOM050]	ComRxTimeBase 制限事項
[NCOM056]	Com_Init の引数範囲指定
[NCOM057]	Com_SetIpduGroup の引数範囲指定
[NCOM058]	Com_SendSignal の引数範囲指定
[NCOM059]	Com_ReceiveSignal の引数範囲指定
[NCOM060]	Com_SendSignalGroup の引数範囲指定
[NCOM061]	Com_ReceiveSignalGroup の引数範囲指定
[NCOM062]	Com_InvalidateSignal の引数範囲指定
[NCOM063]	Com_InvalidateSignalGroup の引数範囲指定
[NCOM064]	Com_TriggerIPDUSend の引数範囲指定
[NCOM065]	Com_SwitchIpduTxMode の引数範囲指定
[NCOM066]	Com_RxIndication の引数範囲指定
[NCOM067]	Com_TxConfirmation の引数範囲指定
[NCOM068]	Com_DeInit の初期化チェック
[NCOM069]	Com_IpduGroupControl の初期化チェック
[NCOM070]	Com_ReceptionDMControl の初期化チェック
[NCOM071]	Com_SetIpduGroupId の初期化チェック
[NCOM072]	Com_SendSignal の初期化チェック
[NCOM073]	Com_SendSignal の引数が NULL_PTR
[NCOM074]	Com_ReceiveSignal の初期化チェック
[NCOM075]	Com_ReceiveSignal の引数が NULL_PTR
[NCOM076]	Com_SendSignalGroup の初期化チェック
[NCOM077]	Com_ReceiveSignalGroup の初期化チェック
[NCOM078]	Com_InvalidateSignal の初期化チェック
[NCOM079]	Com_InvalidateSignalGroup の初期化チェック
[NCOM080]	Com_TriggerIPDUSend の初期化チェック
[NCOM081]	Com_SwitchIpduTxMode の初期化チェック
[NCOM082]	Com_RxIndication の初期化チェック
[NCOM083]	Com_RxIndication の引数が NULL_PTR
[NCOM084]	Com_TxConfirmation の初期化チェック
[NCOM085]	Com_SendDynSignal の初期化チェック
[NCOM086]	Com_SendDynSignal の引数範囲指定
[NCOM087]	Com_SendDynSignal の引数が NULL_PTR
[NCOM088]	Com_ReceiveDynSignal の初期化チェック
[NCOM089]	Com_ReceiveDynSignal の引数範囲指定

仕様タグ	概要
〔NCOM090〕	Com_ReceiveDynSignal の引数が NULL_PTR
〔NCOM091〕	Com_IpduGroupControl の引数が NULL_PTR
〔NCOM092〕	Com_ReceptionDMControl の引数が NULL_PTR
〔NCOM093〕	Com_ClearIpduGroupVector の引数が NULL_PTR
〔NCOM094〕	Com_SetIpduGroup の引数が NULL_PTR
〔NCOM095〕	ComSignalType に UINT8_DYN を指定した場合エラー
〔NCOM096〕	Com_RxIpduCallout の振る舞い
〔NCOM097〕	N 回送信中にリトライ要求が起こった場合の振る舞い
〔NCOM098〕	1 回送信要求と N 回送信要求が同時に来た場合の振る舞い
〔NCOM099〕	送信プロパティが PENDING の場合の振る舞い
〔NCOM101〕	ComTxTimeBase の範囲を指定
〔NCOM102〕	ComRxTimeBase の範囲を指定
〔NCOM103〕	ComConfigurationUseDet のデフォルト値を指定
〔NCOM104〕	数値変換できないパラメータが指定された場合
〔NCOM105〕	ComFilterAlgorithm のパラメータチェック
〔NCOM106〕	ComFilterAlgorithm のパラメータチェック
〔NCOM107〕	ComFilterAlgorithm のパラメータチェック
〔NCOM108〕	ComFilterAlgorithm のパラメータチェック
〔NCOM109〕	無効値処理を NOTIFY に設定し無効値通知を設定しない場合の処理
〔NCOM110〕	送信 DM 開始タイミングについて
〔NCOM111〕	シグナル配置のチェック

4.7.4 本バージョンで対象外となる AUTOSAR 仕様一覧

データシーケンス管理

[COM587], [COM588], [COM590], [COM687], [COM688], [COM726], [COM727],
[COM003_Conf], [COM592_Conf], [COM593_Conf], [COM594_Conf], [COM595_Conf]

通信保護

[COM596], [COM597], [COM599_Conf], [COM600_Conf], [COM601_Conf]

可変長データ通信

[COM627], [COM628], [COM629], [COM630], [COM690], [COM711], [COM712], [COM724],
[COM753], [COM754], [COM755], [COM756], [COM757], [COM758]

大型データ通信

[COM662], [COM713], [COM714], [COM720]

TP 通信

[COM650], [COM651], [COM654], [COM655], [COM656], [COM657], [COM658], [COM663],
[COM691], [COM692], [COM693], [COM721], [COM725], [COM760], [COM782], [COM783]

シグナルゲートウェイ

[COM357], [COM359], [COM360], [COM361], [COM362], [COM365], [COM370], [COM377],
[COM383], [COM384], [COM386], [COM400], [COM466], [COM539], [COM598], [COM666],
[COM667], [COM668], [COM701], [COM702], [COM703], [COM704], [COM705], [COM706],
[COM735], [COM544_Conf], [COM545_Conf], [COM546_Conf], [COM547_Conf],
[COM548_Conf], [COM549_Conf], [COM550_Conf], [COM551_Conf], [COM729_Conf]

以下の仕様の内、シグナルゲートウェイに関する部分は対象外となる。

[COM127_Conf], [COM157_Conf], [COM158_Conf], [COM170_Conf], [COM232_Conf],
[COM257_Conf], [COM259_Conf], [COM437_Conf]

下位層からのトリガ送信

[COM001], [COM260], [COM395], [COM475], [COM578], [COM647], [COM648], [COM696],
[COM697], [COM766], [COM765_Conf]

その他

[COM001_Conf], [COM002_Conf]

変更履歴

バージョン	変更日	変更内容	変更者
1.0.0	2014/12/02	新規作成	NCES
1.0.1	2015/03/31	誤記修正	NCES
1.1.0	2016/03/31	シグナルグループ対応	NCES