



O3MCANインターフェース

このドキュメントでは、O3Mxxx センサー ファミリーの CAN インターフェイスについて説明します。主に J1939 メッセージと UDS 機能 (オンライン パラメーター化とグラフィック プリミティブ (O3M2xx のみ) など) について説明します。CANopen の仕様についても簡単に説明します。

V07 – 2020-05-19

1. コンテンツ

2.	出力メッセージ	3
2.1.	DBC ファイル	3
2.2.	メッセージ ID に関する注意事項	3
2.3.	すべての O3M バリエーションに共通のメッセージ	3
2.3.1.	バージョン情報	3
2.3.2.	グローバル情報	4
2.3.3.	仮想スイッチ	5
2.3.4.	2D カメラのキャリブレーションデータ	6
2.4.	DI バリエーション専用のメッセージ	7
2.4.1.	基本機能パラメータ	7
2.4.2.	基本機能の結果	10
2.5.	OD バリエーション専用のメッセージ	12
2.6.	LG バリエーション専用のメッセージ	17
3.	入力メッセージ	22
3.1.	すべての O3M バリエーションに共通のメッセージ	22
3.1.1.	エゴデータ	22
3.1.2.	仮想スイッチ	23
3.1.3.	UDS 転送プロトコルに関する注意事項	23
3.1.4.	2D オーバーレイ入力	25
3.1.5.	オンラインパラメータ化	33
3.2.	メッセージの説明	36
3.3.	DI バリエーション専用のメッセージ	37
3.3.1.	DI バリエーションに固有のオンラインパラメータ	37
3.4.	OD バリエーション専用のメッセージ	39
3.4.1.	OD バリエーションに固有のオンラインパラメータ	39
3.5.	LG バリエーション専用のメッセージ	45
3.5.1.	LG バリエーションに固有のオンラインパラメータ	45
3.6.	スタンバイモード	50
3.6.1.	スタンバイモードに入る	50
3.6.2.	スタンバイモードを解除する	50



3.7。	教える.....	51
3.7.1.	ティーチコマンド.....	51
3.7.2.	アンティーチコマンド.....	51
3.8.	O3M リセット.....	51
3.9.	UDS 応答コード	51
3.10.	UDS 診断情報	51
4.	開けられる	53
5.	バージョン履歴	54



2. 出力メッセージ

2.1. DBCファイル

O3M のすべてのファームウェア リリースには、末尾が「.dbc」のファイルが含まれています。このファイルには、エンコーディングと説明を含む、CAN 通信で使用するすべてのメッセージと信号が記述されています。このドキュメントの目的は、信号に含まれる情報の使用方法を説明することです。便宜上、信号の符号化について簡単に説明します。信号エンコーディングの詳細については、ファームウェアリリースに付属のファイルを参照してください。

2.2. メッセージ ID に関するメモ

O3M メッセージ ID は、J1939 ID 仕様に従って生成されます。したがって、各出力メッセージの ID のバイト 0 は O3M の J1939 ID であり、各入力メッセージのバイト 1 は O3M の J1939 ID です。このドキュメントでは、O3M の J1939 ID に依存するメッセージ ID の部分が強調表示されています。赤、別の J1939 ID を持つ O3M と通信するには、それに応じてメッセージ ID を調整してください。ID の末尾の文字「x」は、拡張識別子の使用を示します。

例: メッセージ名: Global_Information (O3M 出力 メッセージ)

DBCファイル (データベース定義) によるID: 0x4FF01EFx (拡張識別子)

バイト3	バイト2	バイト1	バイト0
0x04	0xFF	0x01	0xEF (O3MのJ1939 ID。デフォルトは0xEF)

例: メッセージ名: VS_MixedInput (O3M 入力 メッセージ)

DBCファイル (データベース定義) によるID: 0x4EFEFFEx (拡張識別子)

バイト3	バイト2	バイト1	バイト0
0x04	0xEF	0xEF (O3MのJ1939 ID。デフォルトは0xEF)	FE

2.3. すべての O3M バリエーションに共通のメッセージ

2.3.1. バージョン情報

メッセージ **DBC_ファイル_バージョン** O3M の通信マトリックスのバージョンを照会するために使用できます。バージョンはすべての DBC ファイルの一部であり、ファイル コンテンツの最初の行に表示されます (例: バージョン「DI 3.8.0」)。O3M との通信が正しく行われるようにするには、O3M の通信プロトコルのバージョンが、通信機能によって期待されるバージョンと一致することを確認することをお勧めします。

メッセージ DBC_File_Version は、要求時にのみ出力されます。O3M が使用する通信プロトコルのバージョンを照会するには、次の手順に従ってください。

1) 送信する O3M へのリクエストメッセージ

次の仕様を使用してメッセージを送信します。

名前 (.dbc ファイル)	RQST
ID	0x18EAEFF1x
ダウンロードコンテンツ	3



ペイロード	バイト0	バイト1	バイト2
	0xA0	0xFF	0x00

2) 受け取る O3Mからの返答

応答メッセージを受け取る

名前 (.dbc ファイル)	DBC_ファイル_バージョン		
ID	0x4FFA0EFバツ		
ダウンロードコンテンツ	3		
ペイロード	バイト0	バイト1	バイト2
	バリエーション	DBC バージョン メジャー	DBC バージョン マイナー

プロトコルバージョン (DI_4.21など) のパッチ番号の出力はありません。7)。パッチ バージョンの増分は、ファームウェア パッチまたはドキュメントの更新にのみ使用されるため、実際のプロトコルは変更されず、通信には影響しません。

バリエーション バイトは O3M バリエーションを示します。

バリエーション バイト	O3M バリエーション
0x00	DI
0x01	外径
0x02	LG

注: O3M は ID 18E8FF のメッセージも出力します。EFx で RQST メッセージを確認します。そのメッセージの最初のバイトが 0 でない場合、要求は失敗し、DBC_File_Version メッセージは出力されません。

2.3.2. グローバル情報

グローバル情報メッセージには O3M ステータス情報が含まれており、O3M で発生した問題に気付いて対応できるように、継続的に監視する必要があります。

特に、信号 SwCtrl_OpMode を使用して O3M の動作モードを監視し、リセットまたはスタンバイ モードの場合を除き、それが 0x22 (RUN) であることを確認します。

さらに、メッセージ カウンターを使用して、通信がまだ実行されていることを確認できます。

名前 (.dbc ファイル)	グローバル_インフォメーション
ID	0x4FF01EFバツ
ダウンロードコンテンツ	7

グローバル情報メッセージの信号:

ポジション	名前	単位	範囲	意味
ビット 0 – 31	Glob_master_時間	μs	[0-4294967295]	現在のタイムスタンプ (μs)。71分後にラップアラウンド。
ビット 32 – 39	Glob_sensor_利用可能	ビットフィールド	-	すべてのビットは、O3M 出力の品質に影響を与える可能性のある 1 つの機能のステータスを示します。このような場合でも出力は利用できますが、品質が低下する可能性があります。ビット「TRACKING_ERROR」は、O3M の OD バリエーションでのみ使用されます。 INTERFERENCE_DETECTED (1u) SPRAY_DETECTION (2u) TRACKING_ERROR (4u) INVALID_CAM_ORIENTATION (8u) SIGNAL_PATH_MONITORING (16u)



				INTERNAL_ERROR (32u) BLOCKAGE_DETECTED (64u) FORCE_CALIBRATION_RESET (128u)
ビット 40 – 47	閉塞_統計 我ら	%	[0 .. 100]	O3M の 3D カメラ ウィンドウのうち、泥や霧などで覆われているように見える割合。ビジョン アシスタントを使用して、閉塞検出を設定してください。 値 0xFE は、O3M で閉塞検出が利用できないことを示します。 値 0xFF は、一般的なエラーを示します。
ビット 48 – 53	SwCtrl_OpMo で	-	列挙	O3M 動作モードの現在のステータス。0x11: 初期化 0x12: 起動 0x13: DSP_BOOT 0x14: セルフテスト 0x15: WIAT_DSP_BOOTED 0x17: パラメータ化 0x21: 制限付き実行 0x22: 実行 0x23: スタンバイ 0x31: 緊急 通常動作時の動作モードは 0x22 (RUN) である必要があります。診断サービスを使用して O3M がスタンバイモードに設定されている場合（以下を参照）、動作モードは 0x23 (STANDBY) である必要があります。0x31 は、O3M が自動的に回復できない緊急モードです。
ビット 54 – 55	Global_Inform ation_cnt	-	[0 .. 3]	メッセージカウンター; 値 0、1、2、3 を循環します。メッセージが送信されるたびに更新されます。

2.3.3. 仮想スイッチ

O3M には、結果値と CAN 経由で入力された値を使用して、単純な論理演算を実行する機能があります。ロジックの設定については、ifm Vision Assistant を参照してください。

ロジックへの入力として使用するために CAN を介して O3M に値を送信するには、セクション「O3M 入力メッセージ」で説明されているメッセージ VS_Mixed_Input を使用してください。

論理演算の結果は、VS_MixedOutput、VS_AnalogOutput、VS_DigitalOutput の 3 つの CAN メッセージを使用して出力されます。ifm Vision Assistant を使用して、これらのメッセージの送信を構成します。

メッセージ VS_MixedOutput は、2 つのアナログ出力と 38 のデジタル出力を提供し、ほとんどのアプリケーションで十分です。さらに出力を使用する必要がある場合は、メッセージ VS_AnalogOutput で 4 つの追加のアナログ出力を使用でき、メッセージ VS_DigitalOutput で 62 の追加のデジタル出力を使用できます。

すべてのデジタル出力は、CAN メッセージ内の単一ビットとしてエンコードされます。すべてのアナログ出力は、12 ビットを使用してエンコードされます。値 0x000 から 0xFA0 は、ビットあたり 1/4000 の解像度で値 0 から 1 をエンコードするために使用されます。0xFFD は、アナログ出力にマッピングされた結果が 0 より小さいことを示し、0xFFE は、1 より大きいことを示し、0xFFF は、出力にマッピングされた値がないことを示します。

出力メッセージのレイアウトは次のとおりです。

名前 (.dbc ファイル)	VS_MixedOutput		
ID	0x4FF52EF バツ		
ダウンロードコンテンツ	8		
ペイロード	ビット 0 ~ 11	アナログ出力 00	
	ビット 12 ~ 23	アナログ出力 01	



	ビット 24～61	デジタル出力 0～37、ビットごとに 1 つのデジタル出力
	ビット 62～63	メッセージカウンタ; 値 0、1、2、3 を循環します。メッセージが送信されるたびに更新されます。

名前 (.dbc ファイル)	VS_Analog出力	
ID	0x4FF50EF バツ	
ダウンロードコンテンツ	8	
ペイロード	ビット 0～11	アナログ出力02
	ビット12～23	アナログ出力03
	ビット 24～35	アナログ出力04
	ビット 36～47	アナログ出力05
	ビット 48～49	メッセージカウンタ; 値 0、1、2、3 を循環します。メッセージが送信されるたびに更新されます。

名前 (.dbc ファイル)	VS_デジタル出力	
ID	0x4FF51EF バツ	
ダウンロードコンテンツ	8	
ペイロード	ビット 0～61	デジタル出力 38～99、ビットごとに 1 つのデジタル出力
	ビット 62～63	メッセージカウンタ; 値 0、1、2、3 を循環します。メッセージが送信されるたびに更新されます。

2.3.4. 2Dカメラキャリブレーションデータ

O3M25x および O3M 26x の場合、O3M の 2D カメラのキャリブレーション情報を CAN に出力できます。この動作は、ifm Vision Assistant を使用して構成できます。

2D カメラのキャリブレーション情報には、O3M によって出力された 3D 値を 2D 画像にキャリブレーションするために必要なすべての情報が含まれています。

メッセージ **定数_2D_Calib_Data** Bouguetモデルを使用した2Dカメラレンズの光学モデルが含まれています。帯域幅を維持するために、操作サイクルごとに 1 つのパラメータのみが送信されます。合計 13 のパラメータがあるため、O3M の 2D カメラの完全な光学モデルを作成するには、すべてのパラメータを収集するのに 13 の操作サイクルが必要です。

名前 (.dbc ファイル)	定数_2D_Calib_Data				
ID	0x4FF0AEF バツ				
ダウンロードコンテンツ	5				
ペイロード	バイト0	バイト1	バイト2	バイト3	バイト4
	マルチプレックス信号	リトル エンディアンを使用した float32 データ型としての光学モデルの 1 つのパラメータ (intel) バイトオーダー。			
		マルチプレックス信号値		パラメータ	
		0x00		Fx	
		0x01		ふう	
		0x02		MX	
		0x03		じぶんの	
		0x04		アルファ	
		0x05		K1	
		0x06		K2	
		0x07		K5	
		0x08		K3	
		0x09		K4	
		0x0A		センターオフセットX	
		0x0B		センターオフセットY	
		0x0C		センターオフセットZ	



これらのパラメータの説明は、次の場所にあります。

http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html

メッセージ **Dynamic_2D_Calib_Data** O3M ワールド座標での 2D カメラの位置 (中心と回転角度) に関する情報が含まれています。メッセージのIDは0x4FF09EF。動的キャリブレーションまたはオンラインパラメータサービスを使用して O3M の位置と回転が変更されると、値が変更される可能性があります。したがって、値は連続して出力されません。メッセージのバイトオーダーは

リトルエンディアン (インテル)。

メッセージのレイアウトは次のとおりです。

ポジション	名前	単位	できること 物理的な	範囲	特別 値	意味
ビット 0 ~ 11	ExtrCalib_2D_rot_x	ラッド	$P = 0.1 \cdot \text{パイ} / 180 \cdot C - 1.1 \cdot \text{円周率}$	$[-1.1 \cdot \text{pi} .. 1.1 \cdot \text{円周率}]$	*	2D カメラとワールド座標間の xAxis 周りの回転角度 システム
ビット 12 ~ 19	ExtrCalib_2D_delta_tx	メートル	$\Delta = 0.01 \cdot C - 1.2$	$[-1.2 .. 1.2]$	**	2D カメラの中心と x 方向のワールド座標系の間の追加オフセット。静的オフセットは、メッセージ内の信号「中心オフセット X」です。 「Constant_2D_Calib_Data」
ビット 20 ~ 31	ExtrCalib_2D_rot_y	ラッド	$P = 0.1 \cdot \text{パイ} / 180 \cdot C - 1.1 \cdot \text{円周率}$	$[-1.1 \cdot \text{pi} .. 1.1 \cdot \text{円周率}]$	*	2D カメラとワールド座標間の yAxis 周りの回転角度 システム
ビット 32 ~ 39	ExtrCalib_2D_delta_ty	メートル	$\Delta = 0.01 \cdot C - 1.2$	$[-1.2 .. 1.2]$	**	y方向の追加オフセット。見る ExtrCalib_2D_delta_tx.
ビット 40 ~ 51	ExtrCalib_2D_rot_z	ラッド	$P = 0.1 \cdot \text{パイ} / 180 \cdot C - 1.1 \cdot \text{円周率}$	$[-1.1 \cdot \text{pi} .. 1.1 \cdot \text{円周率}]$	*	2D カメラとワールド座標間の zAxis 周りの回転角度 システム
ビット 52 ~ 59	ExtrCalib_2D_delta_tz	メートル	$\Delta = 0.01 \cdot C - 1.2$	$[-1.2 .. 1.2]$	**	z方向の追加オフセット。見る ExtrCalib_2D_delta_tx.
ビット 62 ~ 63	Dynamic_2D_Calib_Data_cnt	-	-	$[0 .. 3]$	-	メッセージカウンター; 値 0、1、2、3 を循環します

*: 0xFFD = 最小値よりも小さい値、0xFFE = 最大値よりも大きい値、0xFFF = エラー

** *: 0xFD = 最小値よりも小さい値、0xFE = 最大値よりも大きい値、0xFF = エラー

座標系と回転角度の詳細については、O3M のドキュメントを確認してください。

2.4. DIバリエーション専用のメッセージ

O3MXX1 では、3D 値の最小値、最大値、中央値、または特定のパーセンタイルがすべての領域に対して計算される関心領域 (ROI) を定義することができます。領域は、ifm ビジョン アシスタントを使用して定義されます。

2.4.1. 基本機能パラメータ

定義された領域の出力値を理解しやすくするために、領域定義自体も CAN に出力されます。負荷を最小限に抑えるために、1 つのメッセージのみが使用され、すべてのパラメータを送信するためのさまざまなレイアウトがあります。受信者は、ROI の正確な定義を知るために、すべての情報を収集し、それらを組み合わせる必要があります。



4種類のメッセージレイアウトがあります。レイアウト0には、基本的な機能構成に関する一般的な情報が含まれており、最初に1回送信されます。レイアウト1から3には、1つのROIの構成に関する情報が含まれており、定義されたROIごとに連続して送信されます。最後のROIに対してレイアウト3が送信された後、プロセスは再びレイアウト0から開始されます。したがって、合計すると、レイアウトの順序は0 - (1 - 2 - 3) - (1 - 2 - 3) - (1 - 2 - 3) - ... - 0 - (1 - ... (1 - 2 - 3) とともに、定義されたROIごとに送信されます。

ROIのプロパティはifm ビジョン アシスタントを使用して設定されるため、この情報を受け取って解釈する必要はありません。ただし、O3M で ROI 定義が変更されたときにアプリケーションを調整する必要がないように、アプリケーションのROI定義の変更に自動的に対応すると便利な場合があります。したがって、ROI 定義値は常に送信されます。

CANで。

名前 (.dbc ファイル)	BF_Global_Parameters		
ID	0x4FF02EFバツ		
ダウンロードコンテンツ	7		
ペイロード	ビット0～2	ROI_Param_mode: レイアウト インジケータ	
	ビット3～55	ROI_Param_mode 値に応じたペイロード	

このレイアウトは最初に1回送信され、基本機能の一般的な構成が含まれます。

特徴。

ROI_Param_mode = 0	すべてのROIに共通の定義
ビット2～4	出力モード ROIごとにCANで送信される値を定義します。 出力値が少ない出力モードを選択すると、CANの負荷が軽減されます。0 = × 1 = y 2 = z 3 = xyz 4 = アンプ 5 = xyz+ampl 6 = 出力不可
ビット8～14	Number_of_groups 定義されたROIグループの数。ROIがグループに結合されている場合、グループごとに1つの出力のみが作成されます。ROIが結合されていない場合、ROIごとに1つのグループが使用されます。
ビット16～22	Number_of_rois 定義されたROIの合計数。 この数値は、ROIごとに定義が送信されるため、(1 - 2 - 3) ROI_Param_mode サイクルが送信される頻度も示します。
ビット24～29	TemporalWindowSize 出力値のフィルタの設定を示します。このパラメータの詳細については、ifm ビジョン アシスタントを参照してください。

次のレイアウトは、定義されたROIごとに1回送信され、それぞれの詳細なパラメータ化が含まれています。

ROI。

ROI_Param_mode = 1	ROI 構成値 - パート 1	
ビット2～7	ROI_number メッセージ内の値が属するROIを示します。この信号は、各レイアウトメッセージで繰り返されます。	
ビット8～14	ROIグループ ROIが属するROIグループを示します。	
ビット16～30	ROI_minimum_cartesian_x 3D ROIの最小x座標を示します。2D ROIの場合、値は-10000です。	
	単位	CM
	範囲	[-10000～10000]



	物理的にすることができます	$P = C - 10000$
	特別な価値	0x7FFD = 最小値よりも小さい
		0x7FFE = 最大値よりも大きい
		0x7FFF = エラー
ビット 32 ~ 46	ROI_maximum_cartesian_x 3D ROI の最大 x 座標を示します。 2D ROI の場合、値は 10000 単位です。	
	す。	CM
	範囲	[-10000~10000]
	物理的にすることができます	$P = C - 10000$
	特別な価値	0x7FFD = 最小値よりも小さい
		0x7FFE = 最大値よりも大きい
		0x7FFF = エラー
ビット 47 ~ 49	ROI_Grp_reference_domain ROI グループの参照ドメイン値を示します。についてはifmビジョンアシスタントをご参照ください。 詳細。 値:	
	0x0	すべての値は独立しています
	0x1	バツ
	0x2	よ
	0x3	Z
	0x4	振幅
ビット 50 ~ 55	ROI_upper_left_x 2D ROI の左上隅のピクセル列を示します。列は 0 から 63 まで数えられ、0 が左端の列になります。 3D ROI の場合、値は 0 です。	

ROI_Param_mode = 2	ROI 構成値 - パート 2	
ビット 2 ~ 7	ROI_number メッセージ内の値が属する ROI を示します。この信号は、各レイアウトメッセージで繰り返されます。	
ビット 8 ~ 9	ROI_Grp_output_value_type ROI グループの出力値のタイプ。 値:	
	0x0	最小
	0x1	最大
	0x2	平均
	0x3	パーセンタイル
ビット 10 ~ 15	ROI_lower_right_x 2D ROI の右下隅のピクセル列を示します。列は 0 から 63 まで数えられ、0 が左端の列になります。 3D ROI の場合、値は 63 です。	
ビット 16 ~ 30	ROI_minimum_cartesian_y 3D ROI の最小 y 座標を示します。 2D ROI の場合、値は -10000 単位です。	
	す。	CM
	範囲	[-10000~10000]
	物理的にすることができます	$P = C - 10000$
	特別な価値	0x7FFD = 最小値よりも小さい
		0x7FFE = 最大値よりも大きい
		0x7FFF = エラー
ビット 32 ~ 46	ROI_maximum_cartesian_y 3D ROI の最大 y 座標を示します。 2D ROI の場合、値は 10000 単位です。	
	す。	CM
	範囲	[-10000~10000]
	物理的にすることができます	$P = C - 10000$



	特別な値	0x7FFD = 最小値よりも小さい
		0x7FFE = 最大値よりも大きい
		0x7FFF = エラー

ROI_Param_mode = 3	ROI 構成値 - パート 3
ビット 2～7	ROI_number メッセージ内の値が属する ROI を示します。この信号は、各レイアウトメッセージで繰り返されます。
ビット 8～11	ROI_lower_right_y 2D ROI の右下隅のピクセル行を示します。行は 0 から 15 まで数えられ、0 が一番上の行になります。 3D ROI の場合、値は 15 です。
ビット 12～15	ROI_upper_left_y 2D ROI の左上隅のピクセル行を示します。行は 0 から 15 まで数えられ、0 が一番上の行になります。 3D ROI の場合、値は 0 です。
ビット 16～30	ROI_minimum_cartesian_z 3D ROI の最小 z 座標を示します。 2D ROI の場合、値は -10000 単位です。
	す。
	範囲
	物理的にすることができます
	特別な値
ビット 32～46	ROI_maximum_cartesian_z 3D ROI の最大 z 座標を示します。 2D ROI の場合、値は 10000 単位です。
	す。
	範囲
	物理的にすることができます
	特別な値

2.4.2. 基本機能の結果

基本機能機能の結果は、定義された ROI グループごとに CAN に出力されます。

出力値には、選択した OutputMode 値に応じて、x、y、z、および振幅値が含まれる場合があります。ROI および ROI グループ自体と同様に、出力モードも ifm ビジョン アシスタントを使用して構成されます。

すべての出力が選択されていない場合の負荷を軽減するために、出力メッセージは ROI グループではなく出力値によって配置されます。このように、出力値が 1 つだけ構成されている場合、他のメッセージを送信する必要はありません。

一方、同じ ROI グループのすべての出力値が必要な場合は、すべてのメッセージを受信し、一致する値を結合する必要があります。

複数のメッセージからの値を組み合わせるときは常に、すべての値が同じ O3M サイクルからのものであることが重要です。そうしないと、x、y、z の値が同じピクセルに属していない可能性があります。

この目的のために、すべてのメッセージにはメッセージ カウンター信号 (ビット 62～63) があります。新しく受信した 2 つのメッセージが同じメッセージ カウンター値を持つ場合、それらは同じ O3M サイクルに属し、両方のメッセージからの情報を組み合わせることができます。



情報を結合する前に、最後に受信した各タイプの2つのメッセージをバッファリングし、メッセージカウンタを比較して一致させることをお勧めします。ここではいくつかの例を示します。

例 1:

	最新メッセージカウンタ	前のメッセージ カウンタ 1
メッセージ1	2	
メッセージ2	2	1
処置: 両方に最新のものを使用してください		

例 2:

	最新メッセージカウンタ	前のメッセージ カウンタ 1
メッセージ1	2	
メッセージ2	3	2
処置: 両方にメッセージ・カウンタ2のメッセージを使用してください。次回のために最新の Message2 データを保持します。		

例 3:

	最新メッセージカウンタ	前のメッセージ カウンタ 1
メッセージ1	2	
メッセージ2	0	3
これは起こりえません。この場合、メッセージは失われます。受信機の改良が必要です。		

例 4:

	最新メッセージカウンタ	前のメッセージ カウンタ 3
メッセージ1	0	
メッセージ2	0	3
処置: 両方に最新のものを使用してください。メッセージ カウンタは3から直接0に戻ります。		

3つ以上のメッセージからの情報を結合する必要がある場合、最大で2つの異なる O3M サイクルからの情報であるため、すべてのメッセージには2つの異なるメッセージカウンタ値しか存在できません。したがって、すべてのメッセージについて、最後に受信した2つのデータを保存しておけば、常に十分です。情報を組み合わせるには、すべてのメッセージで利用できるメッセージカウンタの値を使用します。

すべてのメッセージには、4つの ROI グループの結果値が含まれています。すべての ROI グループが定義されていない場合、値は無効になります。ROI グループが定義されていない限り、出力値を使用しないでください。

ROI グループが定義されていないメッセージは、O3M によって出力されません。

たとえば、5つの ROI グループが定義され、出力モードが3(x,y,z)に設定されている場合、次のようになります。

メッセージが送信されます:

メッセージ	ID	コンテンツ
BF_x_output_0	0x4FF10EF バツ	ROI グループ 1 ~ 4 の X 値 ROI グループ
BF_y_output_0	0x4FF11EF バツ	1 ~ 4 の Y 値 ROI グループ 1 ~ 4 の Z 値
BF_z_output_0	0x4FF12EF バツ	
BF_x_output_1	0x4FF14EF バツ	ROI グループ 5 の X 値 (その他のビットは無効)
BF_y_output_1	0x4FF15EF バツ	ROI グループ 5 の Y 値 (その他のビットは無効)
BF_z_output_1	0x4FF16EF バツ	ROI グループ 5 の Z 値 (その他のビットは無効)

利用可能なすべてのメッセージと ID の完全なリストを次に示します。

グループ	バツ	よ	Z	振幅
グループ 1 - 4	0x4FF10EF バツ	0x4FF11EF バツ	0x4FF12EF バツ	0x4FF13EF バツ



グループ 5 – 8	0x4FF14EF バツ	0x4FF15EF バツ	0x4FF16EF バツ	0x4FF17EF バツ
グループ 9 – 12	0x4FF18EF バツ	0x4FF19EF バツ	0x4FF1AEF バツ	0x4FF1BEF バツ
グループ 13 – 16	0x4FF1CEF バツ	0x4FF1DEF バツ	0x4FF1EEF バツ	0x4FF1FEF バツ
グループ 17 – 20	0x4FF20EF バツ	0x4FF21EF バツ	0x4FF22EF バツ	0x4FF23EF バツ
グループ 21 – 24	0x4FF24EF バツ	0x4FF25EF バツ	0x4FF26EF バツ	0x4FF27EF バツ
グループ 25 – 28	0x4FF28EF バツ	0x4FF29EF バツ	0x4FF2AEF バツ	0x4FF2BEF バツ
グループ 29 – 32	0x4FF2CEF バツ	0x4FF2DEF バツ	0x4FF2EEF バツ	0x4FF2FEF バツ
グループ 33 – 36	0x4FF30EF バツ	0x4FF31EF バツ	0x4FF32EF バツ	0x4FF33EF バツ
グループ 37 – 40	0x4FF34EF バツ	0x4FF35EF バツ	0x4FF36EF バツ	0x4FF37EF バツ
グループ 41 – 44	0x4FF38EF バツ	0x4FF39EF バツ	0x4FF3AEF バツ	0x4FF3BEF バツ
グループ 45 – 48	0x4FF3CEF バツ	0x4FF3DEF バツ	0x4FF3EEF バツ	0x4FF3FEF バツ
グループ 49 – 52	0x4FF40EF バツ	0x4FF41EF バツ	0x4FF42EF バツ	0x4FF43EF バツ
グループ 53 – 56	0x4FF44EF バツ	0x4FF45EF バツ	0x4FF46EF バツ	0x4FF47EF バツ
グループ 57 – 60	0x4FF48EF バツ	0x4FF49EF バツ	0x4FF4AEF バツ	0x4FF4BEF バツ
グループ 61 – 64	0x4FF4CEF バツ	0x4FF4DEF バツ	0x4FF4EEF バツ	0x4FF4FEF バツ

メッセージ レイアウトはすべてのメッセージで同じです。

ビット 0 ~ 14	メッセージの第 1 グループの信号 メッ
ビット 15 ~ 29	セージの第 2 グループの信号 メッセージ
ビット 32 ~ 46	の第 3 グループの信号 メッセージの第 4
ビット 47 ~ 61	グループの信号 メッセージ カウンタ信号
ビット 62 ~ 63	

CAN値から物理値を取得するには、次の式を使用します。

値型	式	特別な価値
X (単位はメートル)	$P = 0.01 * C - 100$	0x7FFD = -100m より小さい値 0x7FFE = 100m より大きい値 0x7FFF = エラー
Y (単位はメートル)	$P = 0.01 * C - 100$	0x7FFD = -100m より小さい値 0x7FFE = 100m より大きい値 0x7FFF = エラー
Z (単位はメートル)	$P = 0.01 * C - 100$	0x7FFD = -100m より小さい値 0x7FFE = 100m より大きい値 0x7FFF = エラー
振幅	$P = 2 * C$	0x0 = エラー

2.5. ODバリエーション専用のメッセージ

O3MXX1 では、物体や反射体の検出に最適化されたファームウェアをインストールすることができます。このソフトウェア バリエーションには、Crash Predictor (CP) と呼ばれる衝突状況を予測する機能も含まれています。

OD ソフトウェアは、最大 20 個のオブジェクトとリフレクターを検出および追跡できます。各オブジェクトには、追跡に使用される一意の識別子があります。オブジェクトは、定義された車両と相互作用する可能性が最も高いオブジェクトがインデックス 0 であるという意味で並べられます。モードが「リフレクター検出」に設定されている場合、リフレクター オブジェクトは配列内の通常のオブジェクトの前にソートされます。CAN に出力されるオブジェクトの数、車両の定義、衝突予測機能のパラメータ化は、ifm Vision Assistant を使用して定義できます。

O3M は、オブジェクトの最も近いエッジの 2D ライン モデルを使用して、オブジェクトの位置を推定します。したがって、オブジェクトの位置は、2つの点 (x1,y1) と (x2,y2) と、オブジェクトの最小範囲と最大範囲 (zMin と zMax) を使用して定義されます。オブジェクト定義の詳細については、ifm Vision Assistant を参照してください。



クラッシュプレディクター機能の結果は、CANメッセージを使用してCANに出力されます
 Crash_Predictor_Info (ID 0x4FF02EF/バス) :

名前 (.dbc ファイル)	Crash_Predictor_Info			
ID	0x4FF02EF/バス			
ダウンロードコンテンツ	8			
ペイロード	信号名	単位	CANから物理的な変換	説明
ビット 0 ~ 7	CP_object_id	-	P=C	衝突が予想される物体のID
ビット 8 ~ 15	CP_ttc	の	$P = 0.04 * C$	衝突までの予測時間。インテリジェント モードでのみ使用できます。詳細については、ifm Vision Assistant のドキュメントを参照してください。
ビット 16 ~ 24	CP_impact_velocity	MS	$P = 0.1 * C$	予測される衝撃速度
ビット 25 ~ 27	CP_crash_predicted	列挙型	$P = C - 2$	衝突が予測されるかどうかに関する情報: 0x-2 = クラッシュ プレディクターが無効になっています 0x-1 = クラッシュ プレディクターは利用できません (クラッシュ 後) 0x0 = クラッシュ プレディクタが有効になっています。墜落は予測されていない 0x1 = クラッシュ 予測 0x2 = エラー
ビット 28 ~ 29	CP_重要度	列挙型	P=C	インテリジェント/ダイナミック クラッシュ プレディクターによって検出された、予測されたクラッシュの重大性について説明します。 値は、時間距離パラメータに従って、どの「タイム ゾーン」が対象であるかを示します。 0x0 = クラッシュ 予測なし 0x1 = タイム ゾーン 1 のクラッシュ 予測 0x2 = タイム ゾーン 2 のクラッシュ 予測 0x3 = タイム ゾーン 3 のクラッシュ 予測 ゾーン ベース モードの場合
ビット 30	CP_Minimum_zone_triggered	-	P=C	合: オブジェクトが検出されたすべてのゾーンの最小値を示します。たとえば、ゾーン 1 と 3 で物体が検出された場合、値は 1 になります。インテリジェント/動的衝突予測モードの場合: 物体が最小ゾーンで検出されたことを示します。これは、物体が最小許容距離よりも近くにあることを意味します。最小ゾーンは、ゾーンベースのクラッシュのゾーンとは異なる必要があります 予測、それは別の定義です。詳細については、「最小許容距離」パラメータに関するifm Vision Assistantのドキュメントを参照してください。
ビット 32 ~ 39	CP_obj_ID_zone1	-	P=C	ゾーン 1 で検出されたオブジェクトの ID。そのゾーンにオブジェクトがある場合にのみ有効です (信号 CP_Zone_1 P=1 または C=2)。
ビット 40 ~ 47	CP_obj_ID_zone2	-	P=C	ゾーン 2 で検出されたオブジェクトの ID。そのゾーンにオブジェクトがある場合にのみ有効です (信号 CP_Zone_2 P=1 または C=2)。
ビット 48 ~ 55	CP_obj_ID_zone3	-	P=C	ゾーン 3 で検出されたオブジェクトの ID。そのゾーンにオブジェクトがある場合にのみ有効です (信号



				CP_Zone_3 P=1 または C=2)。
ビット 56～57	CP_ゾーン_1	-	P=C-1	0x0 = 無効、ゾーンが定義されていない 0x1 = ゾーンが占有されていない 0x2 = ゾーンが占有されている 0x3 = 予約済み
ビット 58～59	CP_Zone_2	-	P=C-1	0x0 = 無効、ゾーンが定義されていない 0x1 = ゾーンが占有されていない 0x2 = ゾーンが占有されている 0x3 = 予約済み
ビット 60～61	CP_ゾーン_3	-	P=C-1	0x0 = 無効、ゾーンが定義されていない 0x1 = ゾーンが占有されていない 0x2 = ゾーンが占有されている 0x3 = 予約済み
ビット 62～63	Crash_Predictor_Info_cnt	-	P=C	メッセージカウンター。範囲は 0..3 です。この値を使用して、各 O3M サイクルのメッセージを結合します。詳細については、「基本機能の結果」セクションを参照してください。

O3M は、最大 20 個のオブジェクトの詳細情報を送信します。CAN に出力されるオブジェクトの数は、ifm Vision Assistant を使用して構成されます。

各オブジェクトの情報は、2 つの CAN メッセージ (A および B) に分割されます。「基本関数の結果」セクションで説明されているように、メッセージ カウンターが一致するメッセージのみを組み合わせることが重要です。オブジェクトは重要度によって順序付けられているため、メッセージの順序はサイクルごとに変化し、配列インデックス内のオブジェクトが 1 から変化する可能性があるためです。次へ循環します。両方のメッセージが同じオブジェクトを説明している場合でも、異なる年齢の情報を再度結合することは望ましくないため、メッセージの再結合には十分注意する必要があります。

メッセージ ID は次のとおりです。

オブジェクトインデックス	IDメッセージA	IDメッセージB
0	0x4FF10EF バツ	0x4FF11EF バツ
1	0x4FF12EF バツ	0x4FF13EF バツ
2	0x4FF14EF バツ	0x4FF15EF バツ
3	0x4FF16EF バツ	0x4FF17EF バツ
4	0x4FF18EF バツ	0x4FF19EF バツ
5	0x4FF1AEF バツ	0x4FF1BEF バツ
6	0x4FF1CEF バツ	0x4FF1DEF バツ
7	0x4FF1EEF バツ	0x4FF1FEF バツ
8	0x4FF20EF バツ	0x4FF21EF バツ
9	0x4FF22EF バツ	0x4FF23EF バツ
10	0x4FF24EF バツ	0x4FF25EF バツ
11	0x4FF26EF バツ	0x4FF27EF バツ
12	0x4FF28EF バツ	0x4FF29EF バツ
13	0x4FF2AEF バツ	0x4FF2BEF バツ
14	0x4FF2CEF バツ	0x4FF2DEF バツ
15	0x4FF2EEF バツ	0x4FF2FEF バツ
16	0x4FF30EF バツ	0x4FF31EF バツ
17	0x4FF32EF バツ	0x4FF33EF バツ
18	0x4FF34EF バツ	0x4FF35EF バツ
19	0x4FF36EF バツ	0x4FF37EF バツ

両方のメッセージを組み合わせると、オブジェクトに関するすべての関連情報が利用可能になります。

A タイプのメッセージからの情報 (* はオブジェクトのインデックスを表します):



ペイロード	信号名	単位	できること 物理的な 変換	説明
ビット 0～6	Obj_*_vx	MS	$P = C \cdot 0.5 - 30$	x軸に沿ったオブジェクトの相対速度。特別な値: 0x7D = -30 m/s 未満 0x7E = 30 m/s を超える 0x7F = エラー
ビット 7	Obj_*_タイプ	列挙型	P=C	タイプ識別子。 0x0 = 通常のオブジェクト 0x1 = レトリフレクター
ビット 8～14	Obj_*_vy	MS	$P = C \cdot 0.5 - 30$	y 軸に沿ったオブジェクトの相対速度。特別な値: 0x7D = -30 m/s 未満 0x7E = 30 m/s を超える 0x7F = エラー
ビット 15	Obj_*_測定された	列挙型	P=C	オブジェクトが現在のフレームで測定 (または外挿) されたかどうかを示すフラグ。 0x0 = 外挿 0x1 = 測定済み
ビット 16～20	Obj_*_ay	m/s ²	$P = C - 10$	y 軸に沿ったオブジェクトの相対加速度。特別な値: 0x1D = -10 m/s ² 未満 0x1E = 10 m/s ² を超える 0x1F = エラー
ビット 21～23	Obj_*_ep	列挙型	P=C	列挙として提供される存在確率。値が大きいくほど、オブジェクトの存在に関する信頼度が高いことを示します。 0x0: [0.00..0.25]% 0x1:]0.25..0.50]% 0x2:]0.50..0.75]% 0x3:]0.75..0.85]% 0x4:]0.85..0.90]% 0x5:]0.90..0.95]% 0x6:]0.95..1.0]%
ビット 24～28	Obj_*_ax			X 軸に沿ったオブジェクトの相対加速度。特別な値: 0x1D = -10 m/s ² 未満 0x1E = 10 m/s ² を超える 0x1F = エラー
ビット 29～31	Obj_*_qvx	列挙型	P=C	列挙として提供される x 軸に沿った速度の質。値が大きいくほど、速度の値に関する信頼度が高いことを示します。 0x0: [0.00..0.25]% 0x1:]0.25..0.50]% 0x2:]0.50..0.75]% 0x3:]0.75..0.85]% 0x4:]0.85..0.90]% 0x5:]0.90..0.95]% 0x6:]0.95..1.0]%
ビット 32～35	Obj_*_az	m/s ²	$P = C - 5$	Z 軸に沿ったオブジェクトの相対加速度。特別な値: 0xD = -5 m/s ² 未満 0xE = 5 m/s ² を超える 0xF = エラー
ビット 36～37	Obj_*_TrackAge	列挙型	P=C	オブジェクトが O3M によって追跡されたフレーム数。 0x0 = [0..2] フレーム 0x1 = [3..12] フレーム



				0x2 = [13..25] フレーム 0x3 = 25 フレーム以上
ビット 38～45	Obj_*_ID	-	P=C	オブジェクトの一意の ID。0 は、オブジェクトが検出されなかったことを示します。有効な値の範囲は 1～255 です。この値を使用して、次のフレームで同じオブジェクトを認識します。オブジェクトの最小
ビット 46～56	Obj_*_zMin	メートル	LP = 0.02*C-10	z 座標。 特別な値: 0x7FD = -10 m 未満 0x7FE = 30 m を超える 0x7FF = エラー
ビット 57～61	Obj_*_vz	MS	P = C*0.5 - 6	Z 軸に沿ったオブジェクトの相対速度 特別な値: 0x1D = -6 m/s 未満 0x1E = 6 m/s を超える 0x1F = エラー
ビット 62～63	Obj_*_A_cnt	-	P=C	メッセージカウンター。範囲は 0..3 です。この値を使用して、各 O3M サイクルのメッセージを結合します。詳細については、「基本機能の結果」セクションを参照してください。

B タイプ メッセージからの情報:

ペイロード	信号名	単位	できること 物理的な 変換	説明
ビット 0～7	Obj_*_dz	メートル	LP = 0.02*C	オブジェクトの高さ (zMax = zMin + dz)。特別な値: 0xFD = 0 未満 0xFE = 5m 以上 0xFF = エラー
ビット 8～19	Obj_*_dy	メートル	LP = 0.02*C - 40	y 軸に沿った両方のオブジェクトの端点間の距離: y2 = y1 + dy 特別な値: 0xFFD = -40 m 未満 0xFFE = 40 m を超える 0xFFF = エラー
ビット 20～31	Obj_*_dx	メートル	LP = 0.02*C - 40	x 軸に沿った両方のオブジェクトの端点間の距離: x2 = x1 + dx 特別な値: 0xFFD = -40 m 未満 0xFFE = 40 m を超える 0xFFF = エラー
ビット 32～44	Obj_*_x1	メートル	LP = 0.02*C - 80	最初のオブジェクトのコーナー座標の x 位置。特別な値: 0x1FFD = -80 m 未満 0x1FFE = 80 m を超える 0x1FFF = エラー
ビット 45～57	Obj_*_y1	メートル	LP = 0.02*C - 80	最初のオブジェクトのコーナー座標の y 位置。特別な値: 0x1FFD = -80 m 未満 0x1FFE = 80 m を超える 0x1FFF = エラー
ビット 58～60	Obj_*_qvy	列挙型	P=C	列挙として提供される、y 軸に沿った速度の品質。値が大きいほど、速度の値に関する信頼度が高いことを示します。 0x0: [0.00..0.25]% 0x1:]0.25..0.50]% 0x2:]0.50..0.75]% 0x3:]0.75..0.85]%



				0x4:]0.85..0.90]% 0x5:]0.90..0.95]% 0x6:]0.95..1.0]%
ビット61	Obj_*_歴史	-	P=C	255の異なるオブジェクトIDしかないため、O3Mはある時点で、新しいオブジェクトに対して同一意の識別子の再利用を開始する必要があります。これが発生するたびに、履歴ビットが変更されます。したがって、ID 0x10 および hist = 0 のオブジェクトが見られて失われた場合、後で新しいオブジェクトに ID 0x10 および hist = 1 が割り当てられます。その時点で、古いオブジェクトのトラックは破棄されます。
ビット62～63	Obj_*_B_cnt	-	P=C	メッセージカウンター。範囲は0..3です。この値を使用して、各O3Mサイクルのメッセージを結合します。詳細については、「基本機能の結果」セクションを参照してください。

2.6. LG バリエーション専用のメッセージ

O3MXX1 では、作物の端や地面の長い山を検出するために最適化されたファームウェアをインストールすることができます。O3M では両方のオブジェクトを「線」と呼んでいます。したがって、「ヒープ」と「ライン」の両方の名前が、CAN インターフェイスで検出されたオブジェクトを表すために使用されます。この機能の詳細については、O3M のドキュメントを参照してください。

ライン ガイダンス (LG) ファームウェアは、最大5つのヒープを検出および追跡できます。デフォルト設定では、3回線のみが使用されます。ただし、CAN インターフェイス上の通信は最大5ヒープ用に設計されているため、以下の説明には5つのヒープすべてが含まれていますが、センサー出力は現在3つに制限されています。

繰り返しますが、ヒープ情報を生成するには、同じ O3M サイクルからのメッセージのみを結合する必要があることに注意することが重要です。受信したメッセージを結合するメカニズムについては、「基本機能の結果」の章を参照してください。

基準点：

ライン ガイダンス出力のすべての位置は、O3M が動的に生成するライン ガイダンス基準点に関連しています。したがって、他のすべての位置は基準点に関連しているため、最も重要なメッセージは基準点情報を含むメッセージです。

名前 (.dbc ファイル)		参考_ポイント_情報	
ID		0x4FF02EFバツ	
ダウンロードコンテンツ		5	
ペイロード	ビット 0 ～ 10	基準点 X 値、単位 m (メートル) 基準点	
	ビット 11 ～ 21	Y 値、単位 m (メートル) 基準点 Z 値、	
	ビット 22 ～ 30	単位 m (メートル)	
	ビット 38 ～ 39	メッセージカウンター。この値を使用してメッセージを結合します	
CANから物理へ 価値変換	X と Y の値: P = C*0.01 - 10 Z-値: P = C*0.01 - 2		
特別な価値	X と Y の値: 0x7FF = エラー 0x7FE = 10 m より大きい値 0x7FD = -10 m より小さい値 Z 値: 0x1FF = エラー 0x1FE = 2 m より大きい値 0x1FD = -2 m より小さい値		



ヒープ:

O3M によって検出されたヒープのプロパティは、ヒープごとに2つのメッセージを使用して出力されます。「基本機能の結果」の章で説明されているメカニズムを使用して、両方のメッセージを結合してください。これは、行の位置を解釈するときに、両方のメッセージの情報が同じ測定値に属することが非常に重要であるためです。

ヒープの正確なプロパティを記述するために、次の信号が使用されます。

名前	できること 物性値 変換	説明	特別な価値
Line_ID	P=C	回線の一意識別子	
Line_alpha	$P = (0.1 * C - 25) * \text{円周率} / 180$	ラジアン単位の線の中心線の方 向	0x1FD = $25/180 * \pi$ より小 さい 0x1FE = $25/180 * \pi$ より 大きい 0x1FF = エラー
Line_curvature	$P = 0.0005 * C - 0.05$	ヒープの中心線の曲率半径。 単位は1 / mです。 0は直線、 高い絶対値は、きつい曲線を示し ます。正の値は左に曲率の方向を 示し、負の値は右に曲率の方向を 示します。	0xFD = より小さい - 0.05 0xFE = より大きい 0.05 0xFF = エラー
Line_quality	P=C	中心線検出の品質。値の範囲は0 ～ 100 です。値が大きいほど、品 質が高いことを示します。カット エッジの高さ	0x7D = 0 より小さい 0x7E = 100 より大きい 0x7F = エラー
Line_zStepDetectionH 8	$P = 0.01 * C$		0x1D = 0 より小さい 0x1E = 5 より大きい 0x1F = エラー
Line_zStepDetectionH エイトバリッド	-	Z段検出高さ信号の有効フラグ	-
Line_foresight	P=C	検出の予測範囲。単位は メートル	0x1D = 0 より小さい 0x1E = 20 より大きい 0x1F = エ ラー
Line_offset	$P = 0.01 * C - 10$	y 方向の参照点からのヒープの中 心線のオフセット。単位はmで す。	0x7FD = 10 未満 0x7FE = 10 より大きい 0x7FF = エラー
Line_Type	-	行の種類 0: ヒー プ 1: カットエッジ	-
ライン_測定	-	現在のフレームでラインが(外 挿ではなく) 測定されたことを 示すフラグ。	-
Line_History	-	同じ ID で新しく作成された回線の ビットを切り替えます。新しい行 が以前に検出されたものと同じ一 意の識別子を取得するたびに、こ のビットが変更されて、前の行が 破棄されたことを示します。	-



Line_centerOfGravityOffset	$P = 0.05 * C - 5$	ヒープの重心からヒープの中心線までの横方向のオフセット。単位はmです。	0xFD = -5より小さい0xFE = 5より大きい0xFF = エラー
Line_centerOfGravityOffsetValid	-	重心オフセット信号の有効フラグ	-
Line_maxHeightOffset	$P = 0.05 * C - 5$	ヒープの中心線に対する最大高さ位置の横方向のオフセット。単位はmです。	0xFD = -5より小さい0xFE = 5より大きい0xFF = エラー
Line_maxHeightOffset有効	-	最大高さオフセット信号の有効なフラグ	-
Line_heapHeight	$P = 0.01 * C$	検出されたヒープの高さ。単位はmです。	0x1FD = 0 未満 0x1FE = 5 より大きい 0x1FF = エラー
Line_heapHeightValid	-	ヒープ高さ信号の有効なフラグ	-
Line_heapWidth	$P = 0.01 * C$	検出されたヒープの幅。単位はmです。	0x3FD = 0 より小さい 0x3FE = 10より大きい 0x3FF = エラー
Line_heapWidthValid	-	ヒープで覆われた yz 平面上の信号領域	-
Line_heapArea	$P = 0.01 * C$	域を持つヒープの有効なフラグ。単位はm ² です。	0x3FD = 0 より小さい 0x3FE = 10より大きい 0x3FF = エラー
Line_heapAreaValid	-	ヒープ領域信号の有効フラグ	-

各ヒープのプロパティは、次のレイアウトを使用して2つのメッセージに分割されます。

Message_Line_A (DLC = 8):

ビット位置	信号
ビット0～6	Line_ID
ビット7～15	Line_alpha
ビット16～23	Line_curvature
ビット24～30	Line_quality
ビット31～39	Line_zStepDetectionHeight
ビット40～44	Line_foresight
ビット45～55	Line_offset
ビット56	Line_Type
ビット57	ライン_測定
ビット58	Line_zStepDetectionHeightValid
ビット59	Line_History
ビット62～63	メッセージカウンター。範囲は0..3です。この値を使用して、各O3Mサイクルのメッセージを結合します。詳細については、「基本機能の結果」セクションを参照してください。

Message_Line_B (DLC = 7):

ビット位置	信号
ビット0～7	Line_centerOfGravityOffset
ビット8～15	Line_maxHeightOffset
ビット16～24	Line_heapHeight
ビット25～34	Line_heapWidth



ビット 35 ～ 44	Line_heapArea
ビット45	Line_maxHeightOffsetValid
ビット46	Line_heapHeightValid
ビット47	Line_heapWidthValid
ビット48	Line_heapAreaValid
ビット49	Line_centerOfGravityOffsetValid
ビット 54 ～ 55	メッセージカウンター。範囲は 0..3 です。この値を使用して、各 O3M サイクルのメッセージを結合します。詳細については、「基本機能の結果」セクションを参照してください。

各ヒープの 2 つのメッセージのメッセージ ID は次のとおりです。

行番号	メッセージAのID	メッセージBのID
ライン0	0x4FF10 EF バツ	0x4FF11 EF バツ
ライン1	0x4FF12 EF バツ	0x4FF13 EF バツ
2行目	0x4FF14 EF バツ	0x4FF15 EF バツ
3行目	0x4FF16 EF バツ	0x4FF17 EF バツ
4行目	0x4FF18 EF バツ	0x4FF19 EF バツ

曲率コマンド:

検出されたヒープ/トリミング エッジのプロパティに加えて、O3M は各行を追跡するために必要な曲率コマンドも出力します。曲率コマンドは、選択した線をたどるのに必要な半径を示します。追加のヒープ プロパティと考えることもできますが、別の CAN メッセージを使用して出力されるため、ここでは別のオブジェクトとして説明します。

名前	CANから物理へ 値変換	説明	特別な値
曲率_ コマンド	$P = C * 0.025 - 803.2$	曲率半径 利用可能なものに従うためには必要 ライン。単位は1/km	0xFFFC = 曲率コマンドではない 0xFFFD = -803.2 より小さい 0xFFFE = 803.2 より大きい 0xFFFF = エラー

5 つのヒープすべての曲率コマンド信号は、5 つのメッセージを使用して出力されます。出力されるヒープが 5 未満の場合、未使用のヒープの曲率コマンド メッセージは O3M によって送信されません。

曲率コマンド メッセージのレイアウトは次のとおりです。

名前 (.dbc ファイル)	MoCa_曲率_A	
ID	0x4FF20 EF バツ	
ダウンロードコンテンツ	7	
ペイロード	ビット 0 ～ 15	Curvature_Command_0、ライン 0 の曲率コマンド
	ビット 16 ～ 22	Line_0_Id、ライン 0 の一意の ID
	ビット 24 ～ 30	Line_1_Id、ライン 1 の一意の ID Curvature_Command_1、ラ
	ビット 32 ～ 47	イン 1 の Curvature コマンド メッセージ カウンター。値 0、1、
	ビット 54 ～ 55	2、3 を循環します。メッセージが送信されるたびに変更されます。このカウンターを使用して、メッセージを他の O3M 出力メッセージと組み合わせてください。

名前 (.dbc ファイル)	MoCa_曲率_B	
ID	0x4FF21 EF バツ	
ダウンロードコンテンツ	7	
ペイロード	ビット 0 ～ 15	Curvature_Command_2、ライン 2 の曲率コマンド
	ビット 16 ～ 22	Line_2_Id、ライン 2 の一意の ID



	ビット 24 ～ 30	Line_3_Id、ライン 3 の一意の ID Curvature_Command_3、ラ
	ビット 32 ～ 47	イン 3 の Curvature コマンド メッセージ カウンター。値 0、1、
	ビット 54 ～ 55	2、3 を循環します。メッセージが送信されるたびに更新されま す。このカウンターを使用して、メッセージを他の O3M 出力 メッセージと組み合わせてください。

名前 (.dbc ファイル)	MoCa_Curvature_C	
ID	0x4FF22EF バツ	
ダウンロードコンテンツ	7	
ペイロード	ビット 0 ～ 15	Curvature_Command_4、ライン 4 の Curvature コマンド
	ビット 16 ～ 22	Line_4_Id、ライン 4 の一意の ID
	ビット 24 ～ 30	予約済み
	ビット 32 ～ 47	予約済み
	ビット 54 ～ 55	メッセージカウンター; 値 0、1、2、3 を循環します。メッセー ジが送信されるたびに更新されます。このカウンターを使用し て、メッセージを他の O3M 出力メッセージと組み合わせてく ださい。



3. 入力メッセージ

3.1. すべての O3M バリエーションに共通のメッセージ

3.1.1. エゴデータ

OD および LG モードでは、O3M は車両のエゴ データに関する情報を使用して、検出されたオブジェクトの動きと線の推定の品質を向上させることができます。

エゴデータの使用は、ifm ビジョンアシスタントを使用して O3M で構成する必要があります。3 つの自我データ モードがあります。

1. エゴデータを使用しない (デフォルト)
2. CAN からの速度のみを使用する
3. CAN からの速度とヨー レートを使用する

エゴデータモード2または3を使用するには、CANで必要な入力情報を提供する必要があります。速度とヨー レートは多くの車両で利用でき、ゲートウェイを使用して O3M に転送できます。ステアリング角度のみが利用できる場合、幾何学的関係を使用してステアリング角度と車両寸法からヨー レートの推定値を計算できます。

速度とヨー レートは、標準の J1939 メッセージを使用して O3M に送信されます。

O3M に速度情報を送信するには、2 つの J1939 標準 CAN メッセージが必要です: 車両の現在の速度を転送するために使用される信号 Wheel_BasedVehicleSpeed を含むメッセージ EBS21 (ID 0x1803FFFE_x)、および信号を含むメッセージ TCO1 (ID 0xCFE6CFE_x)進行方向が前方か後方かを示す DirectionIndicator。

新しい値が利用可能になると、両方のメッセージを頻繁に送信する必要があります。

名前 (.dbc ファイル)	EBS21	
ID	0x1803FFFE _x	
ダウンロードコンテンツ	8	
ペイロード	ビット 16 ~ 31	Wheel_BasedVehicleSpeed 単 位: km/h 変換: $C = P * 256$ (これは、1 km/h または 0.0039 km/h あたり 8 ビットの解像度があることを意味します) 値 の範囲: [0 .. 251] J1939 による説明: スリップの影響を受け、5 Hz から 20 Hz の 周波数範囲でフィルタリングされた 1 つの車軸の車輪速度の平 均として計算された、車両の実際の速度 (前後の速度の正の 値)。前後の速度) スリップの影響を受け、5 Hz から 20 Hz の周 波数範囲でフィルタリングされた 1 つの車軸の車輪速度の平均 として計算されます。

名前 (.dbc ファイル)	TCO1	
ID	0xCFE6CFE _x	
ダウンロードコンテンツ	8	
ペイロード	ビット 30 ~ 31	方向指示器 値の範囲: [0 .. 3] 意味: 0x0 = フォワード 0x1 = 後方 0x2 = エラー 0x3 = 利用不可



	J1939 による説明: 車両の方向を示します。
--	--------------------------

ヨー レート情報を O3M に送信するには、J1939 標準 CAN メッセージ VDC2 (ID 0x18F009FE) のヨー レート信号が使用されます。この場合も、新しい値が利用可能になるたびにメッセージを送信する必要があります。

名前 (.dbc ファイル)	VDC2	
ID	0x18F009FE	
ダウンロードコンテンツ	8	
ペイロード	ビット 24 ~ 39	ヨーレート 単位: ラジアン/秒 換算: $C = (P + 3.92) * 8192$ 単位は必ず rad/s を使用してください。deg/s から rad/s に変換するには、180 で割って π を掛けます。値の範囲: [-3.92 .. 3.92] J1939 による説明: 垂直軸を中心とした回転を示します。

3.1.2. 仮想スイッチ

O3M には、結果値と CAN 経由で入力された値を使用して、単純な論理演算を実行する機能があります。ロジックの設定については、ifm Vision Assistant を参照してください。

メッセージ VS_Mixed_Input は、ロジックへの入力として使用するために、CAN を介して O3M に値を送信するために使用されます。2 つの異なるメッセージ レイアウトを使用します。最初のレイアウトでは、4 つのアナログ入力と 14 のデジタル入力を利用できます。2 番目のレイアウトには、追加の 2 つのアナログ入力が含まれています。両方のレイアウトを交互に使用できますが、2 つの連続するメッセージの間に少なくとも 5 ミリ秒の間隔を空けるように注意する必要があります。

すべてのデジタル入力は、CAN メッセージ内の単一ビットとしてエンコードされます。すべてのアナログ出力は、12 ビットを使用してエンコードされます。値 0x000 から 0xFA0 は、ビットあたり 1/4000 の解像度で値 0 から 1 をエンコードするために使用されます。0xFFF は、信号が利用できないことを示します。0xFA1 から 0xFFE までのすべての値は予約されているため、使用しないでください。

名前 (.dbc ファイル)	VS_MixedInput	
ID	0x4EFEフェス	
ダウンロードコンテンツ	8	
ペイロード	ビット 0～1 は入力 ID セクターです。この信号の値が「0」の場合、メッセージにはアナログ入力 0～3 およびデジタル入力 0～13 が想定されています。値が「1」の場合、アナログ入力 4～5 が想定されます。	
	入力 ID セクター 0:	
	ビット 2～13	アナログ入力 00
	ビット 14～25	アナログ入力 01
	ビット 26～37	アナログ入力 02
	ビット 38～49	アナログ入力 03
	ビット 50～63	デジタル入力 0～13、ビットごとに 1 つのデジタル入力
	入力 ID セクター 1:	
	ビット 2～13	アナログ入力 04
	ビット 14～25	アナログ入力 05

3.1.3. UDS 転送プロトコルに関する注意事項

O3M に長いメッセージを送信するには、ISO14229 で定義されている UDS 転送プロトコルが使用されます。UDS 転送プロトコルの完全な説明はここでは達成できません。したがって、この



セクションでは、パラメータを O3M に送信するために必要なものを簡単に説明することしかできません。不完全であることをお許しください。詳細については、ISO14229 を参照してください。

次のメッセージを送信する前に、O3M から送信された応答メッセージを常に待って、すべてのコマンドが処理されていることを確認してください。

メッセージID

UDS プロトコルは2つのメッセージを使用します。O3M に送信される1つの要求メッセージ (RQST) と応答 O3M から送信されたメッセージ (RSP)。

メッセージ	ID
RQST	0x18DAEF1x
RSP	0x18DAF1EF バツ

メッセージ内容

UDS メッセージは、1 バイトのサービス識別子とペイロードで構成されます。ペイロードの長さは 4096 バイトに制限されています。O3M が使用するすべてのパラメータは、それよりもはるかに短いです。

プロトコル

使用される 2 つの異なるプロトコルがあります: 6 バイトのペイロードを持つ単一のメッセージ プロトコル (プラス サービス識別子) と、より長いペイロードを持つ複数のメッセージ プロトコル。

単一メッセージ プロトコル

ペイロードが 6 バイト以下の場合、1 回のメッセージ転送が使用されます。

リクエストメッセージ (O3M に送信)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
サービスを含むバイト 単位のデータサイズ 識別子	サービス識別子	ペイロード - 最大 6 バイト。 DLC のみを転送するように調整する必要があります。 使用されるバイト。					

応答メッセージ (O3M から送信)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
バイト単位のデータサイズ	サービス識別子 リクエストの。 応答が肯定的な場合、0x40 は 追加しました。	追加情報 (例: エラーコード) 情報の最後のバイト (byte0 のデータサイズで示されるバイト) の値が 0x78 の場合、O3M はコマンドを一度に完了できません。この場合、別のメッセージをお待ちください。					

複数のメッセージ プロトコル

ペイロードが 6 バイトを超える場合、複数のメッセージを使用して転送する必要があります。この目的のために、O3M との間で送受信される最初のメッセージは、転送を確立するために使用され、データの転送には連続したメッセージが使用されます。

最初の要求メッセージ (O3M に送信)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x10 + データサイズの上 位バイト データサイズは最大 4095 バイトまで可能なため、 データサイズの上位バイト は最大 0xF まで可能です。	の下位バイト データサイズ	サービス 識別子	データの最初の 5 バイト				



最初の応答メッセージ、フロー制御 (O3M から送信)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x30 は OK を意味します。 送信を続けます。その他の 値はエラーです。	タイミング情報						

連続したリクエスト メッセージを送信する前に、最初の応答メッセージを待ってください。
2 つのメッセージ間の 5 ミリ秒の間隔よりも速くリクエスト メッセージを送信します。

連続メッセージ (O3M に送信)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
部品識別子。 最初の連続 メッセージには 0x21 があ ります。次のメッセージ数 0x2F まで、0x20 までラッ プアROUNDします。	最大7バイトのデータ						

完全なデータが受信された後でのみ、O3M から応答メッセージが送信されます。最終的な応答メッセージは、単一メッセージ転送の場合と同じです。

UDS 転送プロトコルは、次の O3M 機能に使用されます。

2Dオーバーレイ入力
オンラインパラメータ化

ペイロード コンテンツの詳細については、これらのトピックを参照してください。

3.1.4. 2Dオーバーレイ入力

O3M2xx を使用すると、特定の幾何学的特徴を描画するコマンドを送信できます。プリミティブ 2D ビデオ画像へのオーバーレイとして。これらのコマンドには UDS 転送プロトコルが使用されます。

プリミティブごとに、次の手順に従う必要があります。

- O3Mがアイテムを描画するために必要なコマンドを作成する コマンドを
- O3Mに送信するためのUDPメッセージ構造を作成する UDP通信を処理する
-

一般情報

すべてのプリミティブにはプリミティブ番号が必要です。O3M は、すべてのプリミティブをプリミティブ番号で管理します。プリミティブ番号は、プリミティブをオフにするために必要です。したがって、プリミティブごとに異なるプリミティブ番号を使用することをお勧めします。

プリミティブのタイプに応じて、プリミティブのさまざまなプロパティを設定する必要があります。使用可能なプリミティブ型のプロパティについては、以下で説明します。

すべての位置は、画像の左上隅 (位置 (0/0)) から数えられ、x が列位置、y が行位置です。

すべてのプリミティブは **UDS サービス識別子 0x2E**。

アイコンプリミティブ

アイコン プリミティブ コマンドは、O3M メモリの特定の場所に既にあるアイコンを表示します。すべてのアイコンは、事前に ifm ビジョンアシスタントツールを使用して O3M にロードする必要があります。アイコンを O3M にロードする方法の詳細については、ifm ビジョンアシスタントを参照してください。



アイコン プリミティブのプロパティ:

名前	データ・タイプ	説明
プリミティブ番号	Uint8	プリミティブの一意の識別子。送信するプリミティブごとに異なる識別子を使用してください。
アイコンID	Uint16	表示するアイコン画像のID。ID はアイコンをロードするときのビジョンアシスタント ア
posX	Uint12	アイコンの列はピクセル単位で表示されます アイコンの
ボックスY	Uint12	行はピクセル単位で表示されます

コマンド レイアウト:

	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
バイト0	0xFD							
バイト1	0x23							
バイト2	プリミティブ番号							
バイト3	iconID 下位バイト							
バイト4	iconID 上位バイト							
バイト5	posX、下位 8 ビット							
バイト6	posY、下位 4 ビット				posX、上位 4 ビット			
バイト7	posY、上位 8 ビット							

UDS コマンド フロー:

1. O3M (ID 0x18DAFF1x、DLC = 8) Byte0 Byte1 Byte2 Byte3 Byte4

					バイト5	バイト6	バイト7
0x10	0x09	0x2E	Byte0 - コマンドの Byte4				

2. 応答O3Mからのメッセージ (ID 0x18DAF1EFx) バイト 0 バイト 1

バイト 2	バイト 3	バイト 4			バイト5	バイト6	バイト7
0x30	タイミング情報 (0x00、0x0A、0x78、0x00、0x00、0x00、0x00)						

3. O3M (ID 0x18DAFF1x、DLC = 4)

バイト0	バイト1	バイト2	バイト3	
0x21	Byte4 - コマンドの Byte7			

4. 応答O3Mからのメッセージ (ID 0x18DAF1EF バッ)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x03	0x6E	0xFD	0x23	0x00	0x00	0x00	0x00

O3M 2D 画像の指定した位置にアイコンが表示されます。

楕円プリミティブ

楕円プリミティブ コマンドは、2D 画像内の指定された位置に、指定されたエッジと塗りつぶしの色を持つ楕円を表示します。色と位置の希望する設定を見つけるには、使用するのが最善です

ifmビジョンアシスタントオーバーレイエディター。

楕円プリミティブのプロパティ:

名前	データ・タイプ	説明
プリミティブ番号	Uint8	プリミティブの一意の識別子。送信するプリミティブごとに異なる識別子を使用してください。楕円の線
LineWidth	Uint8	(ピクセル) 線の赤色の値 (0x00 .. 0xFF) 線の緑色の値
LineColorRed	Uint4	(0x00 .. 0xFF) 線の青色の値 (0x00 .. 0xFF) 線のアル
LineColorGreen	Uint4	ファ値 (0x00 .. 0xFF)。0x00 は透明、0xFF は不透明で
LineColorBlue	Uint4	す。
LineColorAlpha	Uint4	
塗りつぶし色赤	Uint4	顔の赤色の値 (0x00 .. 0xFF) 顔の緑色の値
塗りつぶし色緑	Uint4	(0x00 .. 0xFF)



フィルカラーブルー	Uint4	顔の青色の値 (0x00 .. 0xFF) 顔のアルファ値
フィルカラーアルファ	Uint4	(0x00 .. 0xFF)。0x00 は透明、0xFF は不透明です。
posX_leftUpCorner	Uint12	左上隅の列位置 左上隅の行位置 右下隅の
posY_leftUpCorner	Uint12	列位置 右下隅の行位置
posX_rightDownCorner	Uint12	
posY_rightDownCorner	Uint12	

コマンド レイアウト:

	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
バイト0	0xFD							
バイト1	0x22							
バイト2	プリミティブ番号							
バイト3	LineWidth							
バイト4	LineColorGreen				LineColorRed			
バイト5	LineColorAlpha				LineColorBlue			
バイト6	塗りつぶし色緑				塗りつぶし色赤			
バイト7	フィルカラーアルファ				フィルカラーブルー			
Byte8	posX_leftUpCorner、下位 8 ビット							
バイト9	posY_leftUpCorner、下位4ビット				posX_leftUpCorner、上位 4 ビット			
バイト10	posY_leftUpCorner、上位 8 ビット							
バイト11	posX_rightDownCorner、下位 8 ビット							
バイト12	posY_rightDownCorner、下位 4 ビット				posX_rightDownCorner、上位 4 ビット			
Byte13	posY_rightDownCorner、上位 8 ビット							

UDS コマンド フロー:

1. O3M (ID 0x18DAEFF1x、DLC = 8) Byte0 Byte1 Byte2 Byte3 Byte4

					バイト5	バイト6	バイト7
0x10	0x0F	0x2E	Byte0 - コマンドの Byte4				

2. 応答O3Mからのメッセージ (ID 0x18DAF1EFx) バイト0 バイト1

バイト2	バイト3	バイト4		バイト5	バイト6	バイト7
0x30	タイミング情報 (0x00、0x0A、0x78、0x00、0x00、0x00、0x00)					

3. O3M (ID 0x18DAEFF1x、DLC = 8)

バイト0	バイト1	バイト2	バイト3	バイト4		バイト5	バイト6	バイト7
0x21	Byte5	コマンドの Byte11						

4. O3M (ID 0x18DAEFF1x、DLC = 3) Byte0 Byte1 Byte2

0x22	Byte12	コマンドの Byte13

5. 応答O3Mからのメッセージ (ID 0x18DAF1EF バツ)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x03	0x6E	0xFD	0x22	0x00	0x00	0x00	0x00

楕円は、指定された色とスタイルで、指定された位置に O3M 2D 画像上に表示されます。

ポリラインプリミティブ

ポリライン プリミティブ コマンドは、2D 画像内に指定された線の色と最大 22 の点 (頂点) で線を表示します。色と位置に必要な設定を見つけるには、ifm ビジョン アシスタント オーバーレイ エディターを使用するのが最善です。

パフォーマンスを最適化するために、使用されたポイントの位置のみが転送されます。したがって、ペイロードのサイズと O3M に送信するメッセージの数は、ラインの必要な頂点の数によって異なります。



ポールの性質

ylene プリミティブ:

名前	データ・タイプ	説明
プリミティブ番号	Uint8	プリミティブの一意の識別子。送信するプリミティブごとに異なる識別子を使用してください。楕円（ピクセル）の線で
LineWidth	Uint8	線の点（頂点）の数。ラインを作るには少なくとも2つ必要です。最大22（0x16）がサポートされます。
NumberOfPoints	Uint8	
LineColorRed	Uint4	ラインの赤色の値 (0x00 .. 0xFF) ラインの緑色の値 (0x00 .. 0xFF) ラインの青色の値 (0x00 .. 0xFF) ラインのアルファ値 (0x00 .. 0xFF)。0x00 は透明、0xFF は不透明です。
LineColorGreen	Uint4	
LineColorBlue	Uint4	
LineColorAlpha	Uint4	
posX_01	Uint12	1点目の列位置 1点目の列位置
posY_01	Uint12	
...	Uint12	ポイント 2 ~ 22 の列と行の位置

コマンド レイアウト:

	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
バイト0	0xFD							
バイト1	0x20							
バイト2	プリミティブ番号							
バイト3	LineWidth							
バイト4	NumberOfPoints							
バイト5	LineColorGreen				LineColorRed			
バイト6	LineColorAlpha				LineColorBlue			
バイト7	posX_01、下位 8 ビット							
Byte8	posY_01、下位 4 ビット				posX_01、上位 4 ビット			
バイト9	posY_01、上位 8 ビット							
バイト10	posX_02、下位 8 ビット							
バイト11	posY_02、下位 4 ビット				posY_02、下位 4 ビット			
バイト12	posY_02、上位 8 ビット							
...	posX_01 および posY_01 で説明したように、X および Y 位置のポイントごとに3バイト							

コマンドの合計バイト数は $7 + 3 * (\text{NumberOfPoints})$ です。

UDSコマンドフロー（10ポイントの例）：

コマンドのバイト数は $7 + 3 * 10 = 37$ (Byte0 から Byte36) です。最初の要求メッセージでは 5 バイトが転送され、連続するメッセージごとに最大 7 バイトが転送されます。したがって、合計 6 つの要求メッセージが必要です。最後のメッセージのペイロードは $37 - 5 - 4 * 7 = 4$ です。したがって、パーツ識別子に最初のバイトが必要なため、DLC は 5 です。

さらにリクエスト メッセージが必要な場合は、メッセージごとにパーツ識別子を増やしてください。

O3M 応答は、最後の要求メッセージが完了した後に O3M によって送信されます。

1. O3M (ID 0x18DAFF1x、DLC = 8)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x10	0x26 (データ サイズ = 37 バイトのペイロード + 1 バイトのサービス識別子)	0x2E	Byte0 - コマンドの Byte4				

2. 応答O3Mからのメッセージ (ID 0x18DAF1EFx) バイト 0 バイト 1

バイト 2	バイト 3	バイト 4	バイト 5	バイト 6	バイト 7
0x30	タイミング情報 (0x00、0x0A、0x78、0x00、0x00、0x00、0x00)				

3. O3M (ID 0x18DAFF1x、DLC = 8)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
------	------	------	------	------	------	------	------



0x21	バイト5 - コマンドのバイト11						
4. O3M (ID 0x18DAEFF1x、DLC = 8)							
バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x22 Byte12 - コマンドの Byte18							
5. O3M (ID 0x18DAEFF1x、DLC = 8)							
バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x23 Byte19 - コマンドの Byte25							
6. O3M (ID 0x18DAEFF1x、DLC = 8) Byte0 Byte1 Byte2 Byte3 Byte4							
					バイト5	バイト6	バイト7
0x24 Byte26 - コマンドの Byte32							
7. O3M (ID 0x18DAEFF1x、DLC = 5)							
バイト0	バイト1	バイト2	バイト3	バイト4			
0x25 Byte33 - コマンドの Byte36							
8. 応答O3Mからのメッセージ (ID 0x18DAF1EFバツ)							
バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x03	0x6E	0xFD	0x20	0x00	0x00	0x00	0x00

これで、指定された色とスタイル、指定された数のコーナーとコーナーの位置で、線が O3M 2D 画像に表示されます。

ポリゴンプリミティブ

ポリゴンプリミティブコマンドは、指定された線の色と2D画像内の最大22ポイント（頂点）で塗りつぶされたまたは塗りつぶされていないポリゴンを表示します。色と位置に必要な設定を見つけるには、ifm ビジョン アシスタント オーバーレイ エディターを使用するのが最善です。

パフォーマンスを最適化するために、使用されたポイントの位置のみが転送されます。したがって、ペイロードのサイズと O3M に送信するメッセージの数は、ポリゴンの頂点の数によって異なります。

ポリゴン プリミティブは、ポリライン プリミティブによく似ています。2つの大きな違いがあります

1. ラインは自動的に閉じて、最後のポイントの後の最初のポイントに戻ります。
2. 指定した色でポリゴンが塗りつぶされます。

ポリゴン プリミティブのプロパティ:

名前	データ・タイプ	説明
プリミティブ番号	UInt8	プリミティブの一意的識別子。送信するプリミティブごとに異なる識別子を使用してください。楕円（ピクセル）の線で
LineWidth	UInt8	線の点（頂点）の数。ラインを作るには少なくとも2つ必要です。最大22（0x16）がサポートされます。
NumberOfPoints	UInt8	線の点（頂点）の数。ラインを作るには少なくとも2つ必要です。最大22（0x16）がサポートされます。
LineColorRed	UInt4	ラインの赤色の値 (0x00 .. 0xFF) ラインの緑色の
LineColorGreen	UInt4	値 (0x00 .. 0xFF) ラインの青色の値 (0x00 ..
LineColorBlue	UInt4	0xFF) ラインのアルファ値 (0x00 .. 0xFF)。0x00
LineColorAlpha	UInt4	は透明、0xFF は不透明です。
塗りつぶし色赤	UInt4	内部の赤色の値 (0x00 .. 0xFF) 内部の緑色の値
塗りつぶし色緑	UInt4	(0x00 .. 0xFF) 内部の青色の値 (0x00 .. 0xFF) 内部の
フィルカラーブルー	UInt4	アルファ値 (0x00 .. 0xFF)。0x00 は透明、0xFF は不
フィルカラーアルファ	UInt4	透明です。
posX_01	UInt12	1点目の列位置 1点目の列位置
posY_01	UInt12	
...	UInt12	ポイント 2 ~ 22 の列と行の位置



コマンド レイアウト:

	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
バイト0	0xFD							
バイト1	0x21							
バイト2	プリミティブ番号							
バイト3	LineWidth							
バイト4	NumberOfPoints							
バイト5	LineColorGreen				LineColorRed			
バイト6	LineColorAlpha				LineColorBlue			
バイト7	塗りつぶし色緑				塗りつぶし色赤			
Byte8	フィルカラーアルファ				フィルカラーブルー			
バイト9	posX_01、下位 8 ビット							
バイト10	posY_01、下位 4 ビット				posX_01、上位 4 ビット			
バイト11	posY_01、上位 8 ビット							
バイト12	posX_02、下位 8 ビット							
Byte13	posY_02、下位 4 ビット				posY_02、下位 4 ビット			
バイト14	posY_02、上位 8 ビット							
...	posX_01 および posY_01 で説明したように、X および Y 位置のポイントごとに3バイト							

コマンドの合計バイト数は $9 + 3 * (\text{NumberOfPoints})$ です。

UDS コマンド フロー (4 ポイントの例):

コマンドのバイト数は $9 + 3 * 4 = 19$ (Byte0 から Byte18) です。最初の要求メッセージでは 5 バイトが転送され、連続するメッセージごとに最大 7 バイトが転送されます。したがって、合計 3 つの要求メッセージが必要です。最後のメッセージのペイロードは $19 - 5 - 1 * 7 = 7$ です。したがって、パーツ識別子に最初のバイトが必要なため、DLC は 8 です。

さらにリクエスト メッセージが必要な場合は、メッセージごとにパーツ識別子を増やしてください。

O3M 応答は、最後の要求メッセージが完了した後に O3M によって送信されます。

1. O3M (ID 0x18DAFF1x、DLC = 8)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x10	0x14 (データサイズ = 19 バイトのペイロード + 1 バイトサービス識別子)	0x2E	Byte0 - コマンドの Byte4				

2. 応答O3Mからのメッセージ (ID 0x18DAF1EFx) バイト0 バイト1

バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x30	タイミング情報 (0x00、0x0A、0x78、0x00、0x00、0x00、0x00)				

3. O3M (ID 0x18DAFF1x、DLC = 8) Byte0 Byte1 Byte2 Byte3 Byte4

					バイト5	バイト6	バイト7
0x21	Byte5 - コマンドの Byte11						

4. O3M (ID 0x18DAFF1x、DLC = 8)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x22	Byte12 - コマンドの Byte18						

5. 応答O3Mからのメッセージ (ID 0x18DAF1EF バツ)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x03	0x6E	0xFD	0x21	0x00	0x00	0x00	0x00

指定された色とスタイル、指定された数のコーナーとコーナーの位置で、ポリゴンが O3M 2D 画像に表示されます。

テキストプリミティブ

テキスト プリミティブ コマンドは、2D 画像内に指定されたテキストと背景色とサイズでテキスト ポリゴンを表示します。テキストには、utf-16 エンコーディングを使用して最大 40 文字をエンコードできます。色と位置に必要な設定を見つけるには、ifm ビジョン アシスタント オーバーレイ エディターを使用するのが最善です。



パフォーマンスを最適化するために、使用される文字のみ。したがって、ペイロードのサイズと O3M に送信するメッセージの数は、テキストの文字数によって異なります。

テキスト プリミティブのプロパティ:

名前	データ・タイプ	説明
プリミティブ番号	Uint8	プリミティブの一意の識別子。送信するプリミティブごとに異なる識別子を使用してください。テキストの
LineColorRed	Uint4	赤色の値 (0x00 .. 0xFF) テキストの緑色の値 (0x00 ..
LineColorGreen	Uint4	0xFF) テキストの青色の値 (0x00 .. 0xFF) テキストの
LineColorBlue	Uint4	アルファ値 (0x00 .. 0xFF)。0x00 は透明、0xFF は不透
LineColorAlpha	Uint4	明です。
塗りつぶし色赤	Uint4	背景の赤色の値 (0x00 .. 0xFF) 背景の緑色の値 (0x00 ..
塗りつぶし色緑	Uint4	0xFF) 背景の青色の値 (0x00 .. 0xFF) 背景のアルファ値
フィルカラーブルー	Uint4	(0x00 .. 0xFF)。0x00 は透明、0xFF は不透明です。
フィルカラーアルファ	Uint4	
NumberOfChars	Uint8	テキストの文字数 (0x11~0x40)
フォントサイズ	Uint8	フォントサイズのインデックス。通常は 1~4 を使用しま
TextPosX	Uint12	テキスト表示の列位置 テキスト表示の
TextPosY	Uint12	行位置 テキスト文字 1、utf-16 エンコー
TextChar01	Int16	ド 追加のテキスト文字
...	Int16	

コマンド レイアウト:

	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
バイト0	0xFD							
バイト1	0x24							
バイト2	プリミティブ番号							
バイト3	LineColorGreen				LineColorRed			
バイト4	LineColorAlpha				LineColorBlue			
バイト5	塗りつぶし色緑				塗りつぶし色赤			
バイト6	フィルカラーアルファ				フィルカラーブルー			
バイト7	NumberOfChars							
Byte8	フォントサイズ							
バイト9	TextPosX、下位 8 ビット							
バイト10	TextPosY、下位 4 ビット				TextPosX、上位 4 ビット			
バイト11	TextPosY、上位 8 ビット							
バイト12	TextChar01、下位 8 ビット							
Byte13	TextChar01、上位 8 ビット							
...	追加のテキスト文字、1 文字あたり 2 バイト							

コマンドの合計バイト数は $12 + 2 * \text{NumberOfChars}$ です。

UDS コマンド フロー (8 文字の例、「O3M テキスト」):

コマンドのバイト数 $12 + 2 * 8 = 28$ (Byte0 から Byte27)。最初の要求メッセージでは 5 バイトが転送され、連続するメッセージごとに最大 7 バイトが転送されます。したがって、合計 5 つの要求メッセージが必要です。最後のメッセージのペイロードは $28 - 5 * 7 = 2$ です。したがって、パーツ識別子に最初のバイトが必要なため、DLC は 3 です。

さらにリクエスト メッセージが必要な場合は、メッセージごとにパーツ識別子を増やしてください。

O3M 応答は、最後の要求メッセージが完了した後に O3M によって送信されます。



1. O3M (ID 0x18DAFF1x、DLC = 8)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x10	0x1D (データサイズ=28 バイトのペイロード+1 バイトサービス識別子)	0x2E	Byte0 – コマンドの Byte4				

2. 応答O3Mからのメッセージ (ID 0x18DAF1EFx) バイト 0 バイト 1

バイト 2	バイト 3	バイト 4	バイト 5	バイト 6	バイト 7
0x30	タイミング情報 (0x00、0x0A、0x78、0x00、0x00、0x00、0x00)				

3. O3M (ID 0x18DAFF1x、DLC = 8)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x21	Byte5 – コマンドの Byte11						

4. O3M (ID 0x18DAFF1x、DLC = 8)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x22	Byte12 – コマンドの Byte18						

5. O3M (ID 0x18DAFF1x、DLC = 8)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x23	Byte19 – コマンドの Byte25						

6. O3M (ID 0x18DAFF1x、DLC = 3)

バイト0	バイト1	バイト2
0x24	Byte26 – コマンドの Byte27	

7. 応答O3Mからのメッセージ (ID 0x18DAF1EF バツ)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x03	0x6E	0xFD	0x24	0x00	0x00	0x00	0x00

テキストは、指定された色とスタイルで、指定された位置に O3M 2D 画像上に表示されます。

プリミティブの変更

色や位置など、既存のプリミティブのパラメータを変更するには、同じプリミティブ番号と新しい必要な値を使用してプリミティブを再送信します。

プリミティブの削除

既存のプリミティブを削除するには、プリミティブ コマンド「プリミティブのクリア」を使用します。

プリミティブのクリア コマンドのプロパティ:

名前	データ・タイプ	説明
プリミティブ番号	Uint8	削除するプリミティブを表す一意識別子。の作成に使用したのと同じ値を使用してください 原始的な。

コマンド レイアウト:

	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
バイト0	0xFD							
バイト1	0x27							
バイト2	プリミティブ番号							

UDS コマンド フロー:

1. O3M (ID 0x18DAFF1x、DLC = 5) Byte0 Byte1 Byte2 Byte3 Byte4

0x04	0x2E	Byte0 – コマンドの Byte2		
------	------	---------------------	--	--

2. 応答O3Mからのメッセージ (ID 0x18DAF1EFx) バイト 0 バイト 1

バイト 2	バイト 3	バイト 4	バイト 5	バイト 6	バイト 7
0x03	0x6E	0xFD	0x27	0x00	0x00



指定されたプリミティブ番号のプリミティブは、O3M 2D 画像から削除されます。

3.1.5. オンラインパラメータ化

オンラインパラメータ化により、一部のパラメーターを永続的に書き込まずにすぐに変更できます。

パラメータを送信してから適用する必要があります。

一部のパラメータはリトルエンディアンのバイトオーダーで送信する必要があり、一部のパラメータはビッグエンディアンのバイトオーダーで送信する必要があることに注意してください。

短いパラメータを適用せずに設定する

このコマンドは、最大4バイトのサイズのパラメータを変更するために使用されます。含まれていません
コマンドを適用します。

O3Mへのリクエストメッセージ (ID 0x18DAE FF1x)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
長さ	コマンドIDが高い 0x4Bバイト		IDが低い バイト	ペイロード バイト1	ペイロード バイト2	ペイロード バイト3	ペイロード バイト4

長さは3+ペイロードバイト数です。したがって、1バイトのパラメーターの場合は4、4バイトのパラメーターの場合は7です。

applyで短いパラメータを設定する

このコマンドは、最大4バイトのサイズのパラメータを変更するために使用されます。適用コマンドが含まれています。
この適用は、適用なしで送信された以前のすべてのパラメータにも適用されます。

O3Mへのリクエストメッセージ (ID 0x18DAE FF1x)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
長さ	コマンドIDが高い 0x4Dバイト		IDが低い バイト	ペイロード バイト1	ペイロード バイト2	ペイロード バイト3	ペイロード バイト4

長さは3+ペイロードバイト数です。したがって、1バイトのパラメーターの場合は4、4バイトのパラメーターの場合は7です。

パラメータIDのリストは以下にあります。

適用のない大きなパラメータブロック

このコマンドは、任意のサイズのパラメータを変更するために使用されます。1つのコマンドで変更できるのは4080バイトまでなので、パラメータが大きい場合は複数のコマンドが必要です。このコマンドは適用コマンドが含まれていません。

O3Mへのリクエスト (ID 0x18DAE FF1x)、最初のメッセージ

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
ビット7..4=0001 ビット3..0= 長さビット 11..8	長さ ビット 7..0	コマンド 0x4c	ID高 バイト	IDが低い バイト	オフセットビット 31..24	オフセットビット 23..16	オフセット ビット15..8



2番目のメッセージ

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x21	オフセット ビット 7..0	サイズ ペイロードビット 31..24ビット	サイズ ペイロード 23..16	サイズ ペイロード ビット 15..8	サイズ ペイロード ビット 7..0	最初のバイト の パラメータ	二番目 のバイト パラメータ

次のメッセージは 0x22 で始まり、7 バイトのパラメーターが含まれています。次のメッセージでは、最初のバイトが 0x2f までカウントされ、0x20 から再開されます。

パラメータ ID については、以下を参照してください。

オフセットは、パラメータのオフセットです。これは通常 0 ですが、サイズが >4080 バイトの場合は、いくつかの大きなパラメータ ブロック コマンドを送信し、それに応じてオフセットを変更する必要があります。

適用のある大きなパラメータ ブロック

このコマンドは、任意のサイズのパラメーターを変更するために使用されます。1 つのコマンドで変更できるのは 4080 バイトまでなので、パラメータが大きい場合は複数のコマンドが必要です。このコマンドには、apply コマンドが含まれています。

コマンドが 0x4E になったことを除いて、メッセージの内容は「適用しない大きなパラメータ ブロック」と同じです。

コマンドを適用

このコマンドは、以前に送信された変更を適用します。

O3Mへのリクエストメッセージ (ID 0x18DAFF1x、DLC = 3)

バイト0	バイト1	バイト2
0x02	コマンド 0x02 0x4A	

コマンドをクリア

このコマンドは、まだ適用されていないすべての変更をクリアします。

O3Mへのリクエストメッセージ (ID 0x18DAFF1x、DLC = 3)

バイト0	バイト1	バイト2
0x02	コマンド 0x01 0x4A	

ダーティステータス取得コマンド

ダーティ ステータスは、まだ適用されていない変更があるかどうか、および適用されている変更があるかどうかを示します。

O3Mへのリクエストメッセージ (ID 0x18DAFF1x、DLC = 3)

バイト0	バイト1	バイト2
0x02	コマンド 0x06 0x4A	

O3Mからの回答 (ID 0x18DAF1EFx、DLC = 8):

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
------	------	------	------	------	------	------	------



0x03	0x8A	状態 ワーキング コピー	状態 影 コピー	バイトを埋める	バイトを埋める	バイトを埋める	バイトを埋める
------	------	--------------------	----------------	---------	---------	---------	---------

状態の作業コピー:

0 = 変更は適用されません

1 = 変更が適用されました

状態のシャドウ コピー:

0 = 未適用の変更なし

1 = 適用されていない変更があります

フラッシュコマンドに戻る

このコマンドは、オンラインパラメータ化によって行われた変更を元に戻し、元の設定を復元します
パーマネントパラメータメモリから。

O3Mへのリクエストメッセージ (ID 0x18DAEF1x、DLC = 3)

バイト0	バイト1	バイト2
0x02	コマンド0x05 0x4A	

オンラインパラメータ化の回答メッセージ

応答OK O3M からのメッセージ (ID 0x18DAF1EFバツ)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x01	コマンド0x00 + 0x40		0x00	0x00	0x00	0x00	0x00

コマンドが 0x4A の場合、byte1 は 0x8A になります。

お待ちください O3M からのメッセージ (ID 0x18DAF1EFバツ)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x03	0x7F	コマンド0x78		0x00	0x00	0x00	0x00

コマンドはリクエストからのものです。

このメッセージは、O3M がまだコマンドを処理していることを意味します。これらのいくつかを送信する可能性があります
メッセージ。

エラー O3M からの応答メッセージ (ID 0x18DAF1EF×、DLC=8)

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
0x03	0x7F	コマンドエラーコード0x00		0x00	0x00	0x00	0x00

エラーリスト:



エラーコード	説明
0x10	一般的な拒否
0x11	サポートされていないサービス
0x12	サポートされていないサブ関数
0x13	無効な形式
0x14	応答が長すぎます
0x21	BusyRepeat リクエスト
0x22	条件が正しくありません。これは、オンラインパラメータ化が無効になっていることを意味します。
0x24	リクエストシーケンスエラー
0x31	範囲外アクセスのリクエスト
0x33	が拒否されました
0x35	無効キー
0x36	試行回数を超えました 期限が切れて
0x37	いません
0x70	アップロード ダウンロード 受け入れられない転送データ
0x71	サスペンド 一般的なプログラミングエラー 間違ったブロック
0x72	クシーケンス カウンター サブ機能 アクティブセッション
0x73	でサポートされていないサービス アクティブセッションで
0x7E	サポートされていない
0x7F	

フロー制御メッセージ

フロー制御メッセージは、基礎となる TP プロトコルの一部です。詳細については、https://en.wikipedia.org/wiki/ISO_15765-2 を参照してください。

フロー制御は、O3M がマルチパート送信の最初のメッセージを受信した後に送信されます。これは、送信者に送信できる速度とブロックサイズを通知します。

オンラインパラメータ化からの TP メッセージは次のようになります。

30 00 0A 00 01 00 00 00

送信者に、別のフロー制御メッセージなしですべてのデータを送信する可能性があり、メッセージを 10 ミリ秒離しておく必要があることを伝えます。

3.2. メッセージの説明

説明：

名前：パラメーターの名前

工場出荷時の値: 出荷時またはファームウェア更新後のコンテンツ

データ型：uint = 符号なし整数/ sint = 符号付き整数/ float = IEEE754 浮動小数点。数字は使用ビット数を示します。

Min: 最小許容値。最大: 最大許容値。

適用には特別なアクションが必要です: 「y」は、適用時に内部的に何らかの再初期化が行われることを意味します。これに時間がかかる場合、その間に実行モードが変更され、センサーは短時間使用できなくなります。ここでの「n」は、パラメータが再初期化なしで適用されることを意味します。

Parameter Number: パラメータを識別する番号 Description: パラ

メータの説明

バイトオーダー: バイトオーダー。これは、値 0x12 34 56 78 を 78 56 34 12 (リトルエンディアン) または 12 34 56 78 (ビッグエンディアン) として指定する必要があることを意味します。



3.3. DIバリエント専用のメッセージ

3.3.1. DI バリエントに固有のオンラインパラメータ

名前	Factory 値	Dataタイプ	最小	マックス	パラメータ します 特定番号 アル アクション	パラメータ記述の 必要性を適用	バイトオーダー
イーサネット出力構成 0		uint8	0	1	n	0x4805 0は標準出力、1 デバッグ出力	少し エンディアン
PMDExtrCalib_camCal_trans 0 X		float32	- 30	30	よ	0x4807 の外部キャリブレーション カメラ: X トランスレーション [m]	少し エン
PMDExtrCalib_camCal_trans 0 Y		float32	- 30	30	よ	ディアン 0x4808 少しの外部キャリブレーション カメラ: Y 変換 [m]	エンディアン
PMDExtrCalib_camCal_trans 1 Z		float32	- 30	30	よ	0x4809 少しの外部校正 カメラ: Z 変換 [m]	エンディアン
PMDExtrCalib_camCal_rotX - 1,5707 96327		float32	- 3,1	3,14	よ	0x480A 外部校正 カメラ: Y 回転 [rad]	少し エンディアン
PMDExtrCalib_camCal_rotY 1, 5707 96327		float32	- 3,1	3,14	よ	0x480B の外部キャリブレーション カメラ: Y 回転 [rad]	少し エンディアン
PMDExtrCalib_camCal_rotZ 0		float32	- 3,1	3,14	よ	0x480C の外部キャリブレーション カメラ: Z 回転 [rad]	少し エンディアン
PMDExtrCalib_IlluCal_transX -0,085		float32	- 30	30	よ	0x480D の外部キャリブレーション カメラ: X トランスレーション [m]	少し エンディアン
PMDExtrCalib_IlluCal_transY 0,052		float32	- 30	30	よ	0x480E 少しの外部校正 カメラ: Y 変換 [m]	エンディアン
PMDExtrCalib_IlluCal_transZ 0,047		float32	- 30	30	よ	0x480F 少しの外因性校正 カメラ: Z 変換 [m]	エンディアン
PMDExtrCalib_IlluCalibIsRelat 1 ive		uint8	0	1	よ	0x4810 illuが小さいかどうかを示すフラグ キャリブレーションは、カメラに対する ディアンまたは絶対で与えられます 世界座標。	対 エン
DistImageCust_sprayRemoval 0 感度		uint8	0	3	n	0x4811 スプレー除去 カスタマイズ	少し エンディアン
DistImageCust_pixelPlausibili 1 zationThresholds		uint8	0	3	n	0x4812 ピクセルの正当化 カスタマイズ	少し エンディアン
DistImageCust_blockageSens 0 itivity		uint8	0	3	n	0x4813 ブロッキングのカスタマイズが少ない	エンディアン
DistImageCust_spatialFilterX -100 分		float32	- 100	100	n	0x4814 空間フィルタ デカルト座標、 最小 X	少し エンディアン
DistImageCust_spatialFilterX 100 最 大		float32	- 100	100	n	0x4815 空間フィルタ デカルト座標、 最大 X	少し エンディアン
DistImageCust_spatialFilterY -100 分		float32	- 100	100	n	0x4816 空間フィルタ デカルト座標、 最小 Y	少し エンディアン
DistImageCust_spatialFilterY 100 Max		float32	- 100	100	n	0x4817 空間フィルタ デカルト座標、 最大 Y	少し エンディアン
DistImageCust_spatialFilterZ -100 分		float32	- 100	100	n	0x4818 空間フィルタ デカルト座標、 最小 Z	少し エンディアン



DistImageCust_spatialFilterZ 100 大	最	float32	- 100	100	n	0x4819	空間フィルタ デカルト座標、 最大Z	少し エンディアン
DistImageCust_reflectorThreshholdValue	0.1	float32	0	1	n	0x481A	設定値 再帰反射体を検出するた めの反射率のしきい値	少し エンディアン
DistImageCust_reflectorClose 0 範囲		uint8	0	1	n	0x481B	リフレクター近距離	少し エンディアン
DistImageCust_bfActive	1	sint32	0	1	n	0x481C	BF切り替えフラグ 機能のオン/オフ、デフォルト: オン	少し エンディアン
DistImageCust_bfROIs_numberOfROIs	1	sint32	1	64	n	0x4822	使用された ROI の数	少し エンディアン
DistImageCust_bfTemporalWindowSize	1	sint32	1	50	n	0x482B	時間の経過とともにウィンドウをフィルタリングする	エンディアン
DistImageCust_bfOutputMode	5	sint32	0	5	n	0x482C	CANの出力モード 出力 0: x のみ; 1: y のみ; 2: z のみ; 3: xyz のみ。4: 増幅 のみ。5: xyz+ampl	少し エンディアン
AutoCalibParam_numberOfPatterns 3つ	0	uint8	0	8	n	0x482D	少ないパターン数 自動校正エンディアンに使用 (0,1: 自動校正 無効)	
triggeredStreetCalibration	0	uint8	0	1	n	0x4834	かどうかを示すフラグ ストリートプレーンでのトリガーキャリブレーションベースのエンディアン 見積もりが有効です	少し
num_Frames_Averaging	3	sint32	0	50	n	0x4836	のウィンドウサイズ pmd エンディアン生データの時間的フィルタ リング (0: フィルタリングなし)	少し
pixelPlausiReflectivityThreshold	2.1d	float32	0	20	n	0x4837	の反射率のしきい値 ピクセルブラウシバイゼーション (0: エンディアンフィル タリングなし)	少し
Modulation_Frequency_Mode	3	uint8	0	3	よ	0x4838	0 = ランダム 13 頻度少し ホッピング (デフォルト) 1 = 固定トリプル 1 2 = 固定トリプル 2 3 = 固 定トリプル 3 0x483B の	エンディアン
amplThresholdFactor	1.0	float32	0.1	10	n		振幅係数 異常なエンディアンのカメラ/照 明の組み合わせの補正	少し
ROICanOutputOnOff	1	uint8	0	1	n	0x440B	ROI ビッグエンディアンのオンとオフを切り替えます CAN メッセージ。	
回転画像90	0	uint8	0	1	よ	0x483D	Auto 出力を少しずつ回転させる 2D オーバーレイでエンディアンを表示す る場合は 90 度	
Output2D_OnOff	1	uint8_t	0	1	n	0x5C03	1: 2D 出力は少しオンにする必要があります 0: 2D 出力はオフエンディアンとする	
回転画像90	0	uint8_t	0	1	よ	0x5C06	アナログビデオを少しずつ回転させる 90度時計回りのエンディアン	
Overlay3DOnOff	1	uint8	0	1	n	0x6001	0: GP ビッグエンディアンに送信されたオーバーレイなし 1: オーバーレイを計算し、オー バーレイ データを GP に送信し ます	
FixedGraphicPrimitivesOnOff	1	uint8	0	1	n	0x6003	ショー FixedGraphicPrimitives =1、または表示しない=0	ビッグエンディアン



max_number_of_algo_primitives_for_short_primitives	96	uint8	0	96	n	0x6004	すべての「ビッグエンディアンの最大数」の合計 * 短いプリミティブ	
max_number_of_algo_primitives_for_long_primitives	0	uint8	0	10	n	0x6005	「ビッグエンディアンの最大数」すべての合計 * 長いプリミティブ	
max_number_of_CAN_input_10 プリミティブ_for_short_primitives		uint8	0	10	n	0x6006	すべての「ビッグエンディアンの最大数」の合計 * 短いプリミティブ	
max_number_of_CAN_input_4 プリミティブ_for_long_primitives		uint8	0	10	n	0x6007	すべての「ビッグエンディアンの最大数」の合計 * 長いプリミティブ	
max_number_of_fixed_primitives_for_short_primitives	10	uint8	0	24	n	0x6008	「ビッグエンディアンの最大数」の合計 * 短いプリミティブ	
max_number_of_fixed_primitives_for_long_primitives	8	uint8	0	10	n	0x6009	すべての「ビッグエンディアンの最大数」の合計 * 長いプリミティブ	
ROIGraphicPrimitivesOnOff	1	uint8	0	1	n	0x600C	ショー ROIGraphicPrimitives=1、または表示しない=0	ビッグエンディアン
Display3DFOV	0	uint32	0	3	n	0x600D	0 - 3D FOV ビッグ エンディアンを表示しない オーバーレイ 1 - 3D FOV オーバーレイの長方形を表示 2 - 「3D FOV」テキストで 3D FOV オーバーレイの長方形を表示します。 3 - 3D FOV オーバーレイの四角形を表示し、灰色のオーバーレイで外側にクリップアルゴプリミティブを表示します。	
BlinkingOnDuration	10	uint16	1	100	n	グラフィック	ビッグエンディアンの点滅はプリミティブ、これは「オン」の持続時間です。 単位は 100ms です。	0x600F
BlinkingOffDuration	10	uint16	1	100	n	グラフィック	ビッグエンディアンの点滅はプリミティブ、これは「オフ」期間です。 単位は 100ms です。	0x6010

3.4. ODバリエーション専用のメッセージ

3.4.1. OD バリエーションに固有のオンラインパラメータ

名前	事実 オリ ヴァ レ	データ・タイプ	最小	最大	申し込む ニーズ スペ シア ル ア ク シ ョ ン	パラメータ の 説明 エ ー ス 数	パラメータ の 説明	バイト 注文
イーサネット出力設定 イグレーション	0	uint8	0	1	n	0x8805	0は標準出力、1はデバッグ出力です。	少し エンディアン
VehicleDim_xMin	- 1	float32	- 20	20	よ	0x8808	車両寸法 説明 (世界で 座標)、軸	少し エンディアン



							グランドプレーン上の平行ボックス	
VehicleDim_xMax	1	float32	- 20	20	よ	0x8809	車両寸法説明 (世界で座標)、軸-グランドプレーン上の平行ボックス	少し エンディアン
VehicleDim_yMin	- 1	float32	- 20	20	よ	0x880A	車両寸法説明 (世界で座標)、軸-グランドプレーン上の平行ボックス	少し エンディアン
VehicleDim_yMax	1	float32	- 20	20	よ	0x880B	車両寸法説明 (世界で座標)、軸-グランドプレーン上の平行ボックス	少し エンディアン
VehicleDim_zMax	2	float32	0	10	よ	0x880C	車両寸法説明 (世界で座標)、軸-グランドプレーン上の平行ボックス	少し エンディアン
PMDExtrCalib_camCal_transX	0	float32	- 30	30	よ	0x880D	カメラの外部キャリブレーション: X 変換 [メートル]	少し エンディアン
PMDExtrCalib_camCal_transY	0	float32	- 30	30	よ	0x880E	カメラの外部キャリブレーション: Y 変換 [メートル]	少し エンディアン
PMDExtrCalib_camCal_transZ	1	float32	0.5	30	よ	0x880F	カメラの外部キャリブレーション: Z 変換 [メートル]	少し エンディアン
PMDExtrCalib_camCal_rotX	- 1,570796327	float32	- 3,1	3,14	よ	0x8810	カメラの外部キャリブレーション: Y 回転 [rad]	少し エンディアン
PMDExtrCalib_camCal_rotY	1,570796327	float32	- 3,1	3,14	よ	0x8811	カメラの外部キャリブレーション: Y 回転 [rad]	少し エンディアン
PMDExtrCalib_camCal_rotZ	0	float32	- 3,1	3,14	よ	0x8812	カメラの外部キャリブレーション: Z 回転 [rad]	少し エンディアン
PMDExtrCalib_illuCal_transX	- 0,085	float32	- 30	30	よ	0x8813	カメラの外部キャリブレーション: X 平行移動 [メートル]	少し エンディアン
PMDExtrCalib_illuCal_transY	0,052	float32	- 30	30	よ	0x8814	カメラの外部キャリブレーション: Y 変換 [メートル]	少し エンディアン
PMDExtrCalib_illuCal_transZ	0,047	float32	- 30	30	よ	0x8815	カメラの外部キャリブレーション: Z 変換 [メートル]	少し エンディアン
PMDExtrCalib_illuCaliblsRelative	1	uint8	0	1	よ	0x8816	illuキャリブレーションがカメラに対して行われるのか、世界で絶対的に行われるのかを示すフラグ座標。	少し エンディアン



ObjectListCust_sprayRemovalSensitivity	0	uint8	0	3	n	0x8817	スプレー除去 カスタマイズ	少し エンディアン
ObjectListCust_pixelPlausibilizationThreshold	1	uint8	0	2	n	0x8818	ピクセルの妥当化 カスタマイズ	少し エンディアン
ObjectListCust_blockageSensitivity	0	uint8	0	3	n	0x8819	閉塞のカスタマイズ	少し エンディアン
ObjectListCust_spatialFilterXMin	- 100	float32	- 100	100	n	0x881A	デカルト座標上の空間 フィルター、 最小X	少し エンディアン
ObjectListCust_spatialFilterXMax	100	float32	- 100	100	n	0x881B	デカルト座標上の空間 フィルター、 最大X	少し エンディアン
ObjectListCust_spatialFilterYMin	- 100	float32	- 100	100	n	0x881C	デカルト座標上の空間 フィルター、 最小Y	少し エンディアン
ObjectListCust_spatialFilterYMax	100	float32	- 100	100	n	0x881D	デカルト座標上の空間 フィルター、 最大Y	少し エンディアン
ObjectListCust_spatialFilterZMin	- 100	float32	- 100	100	n	0x881E	デカルト座標上の空間 フィルター、 最小Z	少し エンディアン
ObjectListCust_spatialFilterZMax	100	float32	- 100	100	n	0x881F	デカルト座標上の空間 フィルター、 最大Z	少し エンディアン
ObjectListCust_reflectorThresholdValue	0,1	float32	0	1	n	0x8820	再帰反射体を検出するための 反射率のしきい値を 設定するための値	少し エンディアン
ObjectListCust_auto校正モード	0	uint8	0	42	n	0x8821	自動校正モード (0：無効、1：有効、42： テストモード)	少し エンディアン
ObjectListCust_objectDetectionVariant	1	uint8	0	1	n	0x8822	オブジェクト検出バリエーションの 選択 (0: 通常 オブジェクトリスト+クラッシュ プレディクター、1: レトリフレクターモード)	少し エンディアン
ObjectListCust_ObjectDetectionZMin	0,5	float32	- 10	10	n	0x8823	最小z座標 通常のオブジェクトの場合 検出	少し エンディアン
ObjectListCust_ObjectDetectionZMax	2	float32	- 10	10	n	0x8824	最大z座標 通常のオブジェクトの場合 検出	少し エンディアン
ObjectListCust_CrashPredictorSensitivity	1	uint8	0	2	n	0x8825	クラッシュのカスタマイズ 予測子 - 0: 低感度 (偽陽性率が低いように最適 化されています) - 1: 中 感度 --2：高感度 (真陽性率が高くなるよ うに最適化)	少し エンディアン



ObjectListCust_Ego モーションダイナミクス	2	uint8	0	2	n	0x8826	のパラメータ化 よそよそしい クラッシュの可能性 回避計算。 - 0: 回避なし モデル化された (例: 鉄道) - 1: 低サイドステッピング ダイナミクス (例: トラック、 0.1 ラジアン/秒) - 2: 高 ダイナミクスを回避する (例: 車、0.3 rad/s)	少し エンディアン
ObjectListCust_acc ブレーキ	5	float32	0	30	n	0x8827	ブレーキの定義 衝突時の減速 予測子。	少し エンディアン
ObjectListCust_delay ブレーキ	1	float32	0	3	n	0x8828	衝突予測器のブレーキ 遅れの定義 ObjectListCust_delayBr ake3 >= ObjectListCust_delayBr ake2 >= ObjectListCust_delayBr あける	少し エンディアン
ObjectListCust_ego VMin	0	float32	- 40	40	n	0x8829	最小限の定義 の自我速度 クラッシュイベントの	少し エンディアン
ObjectListCust_ego VMax	18	float32	- 40	40	n	0x882A	計算 最大の定義 の自我速度 衝突イベントの計算 自	少し エンディアン
ObjectListCust_Ego データモード	2	uint8	0	2	よ	0x882B	我速度を可能にする 計算とヨーレート入力 0: 自我速度とヨー レートを取り込む アカウント; 1: 自我速度の み。2: データを使用しませ ん。	少し エンディアン
ObjectListCust_cpD eactivateTimeAfterT リガー	10	float32	0	60	n	0x882C	間の最小時間 2つの予測されたクラッシュ。	少し エンディアン
ObjectListCust_cpM axCrashObjectDista せ	25	float32	0	50	n	0x882D	の最大距離 関連するオブジェクト。	少し エンディアン
ObjectListCust_cpM inDistAllowed	0,5	float32	0	5	n	0x882E	この値を下回る距離のオブ ジェクトは、予測された衝 突につながります。	少し エンディアン
ObjectListCust_cpA 活性化する	0	uint8	0	1	n	0x882F	衝突検出のフラグを有効に します。	少し エンディアン
AutoCalibParam_nu mberOfPatterns	0	uint8	0	8	n	0x8830	使用パターン数 自動キャリブレーション (0,1: 自動校正 無効)	少し エンディアン
トリガーされたStreetCalib 配給	0	uint8	0	1	n	0x8837	キャリブレーションがトリガー されたかどうかを示すフラグ ストリート プレー ンの推定に基づく アクティブ	少し エンディアン



num_Frames_Aver エージング	0	sint32	0	50	n	0x8838	pmd 生データの時間的フィルタリングのためのウィンドウサイズ (0: フィルタリングなし) ピク	少し エンディアン
pixelPlausiReflectivityThreshold	2.1	float32	0	20	n	0x8839	セル妥当性化のための反射率のしきい値 (0: フィルタリングなし)	少し エンディアン
Modulation_Frequency_Mode	3	uint8	0	3	よ	0x883A	0 = ランダム 13 周波数ホッピング (デフォルト) 1 = 固定トリプル 1 2 = 固定トリプル 2 3 = 固定トリプル 3 の振	少し エンディアン
amplThresholdFact または	1.0	float32	0.1	10	n	0x883B	幅係数の補償 珍しいカメラ/イル 組み合わせ	少し エンディアン
ObjectListCust_delay ブレーキ2	0	float32	0	3	n	0x883D	クリティカルレベル 2 のクラッシュ プレディクターのブレーキ遅延 (秒) 有効条件 ObjectListCust_delayBrake3 >= ObjectListCust_delayBrake2 >= ObjectListCust_delayBrake3	少し エンディアン
ObjectListCust_delay ブレーキ3	0	float32	0	3	n	0x883E	クリティカルレベル 3 のクラッシュ プレディクターのブレーキ遅延 (秒) 有効条件 ObjectListCust_delayBrake3 >= ObjectListCust_delayBrake2 >= ObjectListCust_delayBrake3	少し エンディアン
CANMaxNumberOf オブジェクト	8	uint8	0	20	n	0x840A	の構成 最大数 CAN のオブジェクトリスト内のオブジェクト	大きい エンディアン
回転画像90	0	uint8	0	1	よ	0x8843	出力を時計回りに 90 度回転させます。 2Dでのプレゼンテーション	少し エンディアン
Output2D_OnOff	1	int8_t	0	1	n	0x9C03	1: 2D 出力をオンにする 0: 2D 出力をオフにする	少し エンディアン
回転画像90	0	uint8_t	0	1	よ	0x9C06	アナログビデオを時計回りに90度回転します	少し エンディアン
Overlay3DOnOff	1	uint8	0	1	n	0xA001	0: GP に送信されるオーバーレイなし 1: オーバーレイを計算し、オーバーレイ データを GP に送信します	大きい エンディアン
FixedGraphicPrimitivesOnOff	1	uint8	0	1	n	0xA003	を表示します FixedGraphicPrimitives =1 または表示しない=0	大きい エンディアン
max_number_of_al go_primitives_for_s hort_primitives	96	uint8	0	96	n	0xA004	マックスアルゴシース プリミティブ	大きい エンディアン



							1つから生成 アルゴリズム結果フレーム	
max_number_of_algo_primitives_for_long_primitives	0	uint8	0	10	n	0xA005	マックスアルゴショート 1つから生成できる プリミティブ アルゴリズム結果フレーム	大きい エンディアン
max_number_of_CAN_input_primitives_for_short_primitiveの	10	uint8	0	10	n	0xA006	マックスCANショート 1フレームに表示できる プリミティブ	大きい エンディアン
max_number_of_CAN_input_primitives_for_long_primitives	4	uint8	0	10	n	0xA007	最大CANロング 1フレームに表示できる プリミティブ Max	大きい エンディアン
max_number_of_fixed_primitives_for_short_primitives	10	uint8	0	24	n	0xA008	Fixed short 1フレームに表示できる プリミティブ Max	大きい エンディアン
max_number_of_fixed_primitives_for_long_primitives	8	uint8	0	10	n	0xA009	Fixed long 1フレームに表示できる プリミティブ Algoオブ	大きい エンディアン
MaxAlgoObjectsToドロー	10	uint8	0	20	n	0xA00A	ジェクトオーバーレイの プリミティブとして処理 できるオブジェクトの最大 数	大きい エンディアン
CAOnOff	1	uint8	0	1	n	0xA00C	を表示します CAGraphicPrimitives =1 または表示しない=0	大きい エンディアン
OD3DOnオフ	1	uint8	0	1	n	0xA00E	を表示します OD3DGraphicPrimitives =1 または表示しない=0	大きい エンディアン
ODReflectiveOnOff	1	uint8	0	1	n	0xA010	を表示する OD3DGraphicPrimitives =1 または表示しない=0 0 -	大きい エンディアン
displayOnlyMostCriticalゾーン	0	uint8	0	1	n	0xA011	クラッシュを表示 の予測対象オブジェクト 各ゾーン 独立して 1 - クラッシュを表示 最も危険なゾーンのための予 測オブジェクト。他の2つ のゾーンは表示されていま せん	大きい エンディアン
Display3DFOV	0	uint32	0	3	n	0xA012	0 - 3D FOV オーバーレイ を表示しない 1 - 3D 視野を表示 オーバーレイ長方形 2 - 3D 視野を表示 長方形をオーバーレイ 「3D FOV」テキスト 3 - 3D 視野を表示 オーバーレイ長方形と Algo プリミティブのクリップ 灰色の外 オーバーレイ	大きい エンディアン



BlinkingOnDuration	10	uint16	1	100	n	0xA014	グラフィックプリミティブの点滅の場合、これは「オン」の持続時間です。単位は100msです。	大きい エンディアン
BlinkingOffDuration	2	uint16	1	100	n	0xA015	グラフィックプリミティブの点滅の場合、これは「オフ」期間です。単位は100msです。	大きい エンディアン

3.5. LG バリエント専用のメッセージ

3.5.1. LG バリエントに固有のオンラインパラメータ

名前	工場 値	データ タイプ	最小	マックス	単位 （ スケーラブル ）	パラメータID	説明	バイト 注文	
イーサネット出力会社 構成	0	uint8	0	1	n	0xC805	5	0 が標準です 出力、1 デバッグ 出力	少し エンディアン
PMDExtrCalib_ca mCal_transX	0	float32	- 30	30	よ	0xC808	8	外因性 の校正 カメラ：× 翻訳 [m]	少し エンディアン
PMDExtrCalib_ca mCal_transY	0	float32	- 30	30	よ	0xC809	9	外因性 の校正 カメラ：Y 翻訳 [m]	少し エンディアン
PMDExtrCalib_ca mCal_transZ	1	float32	0.5	30	よ	0xC80A	10	外因性 の校正 カメラ：Z 翻訳 [m]	少し エンディアン
PMDExtrCalib_ca mCal_rotX	- 1,5707963 27	float32	- 3,1	3,14	よ	0xC80B	11	外因性 の校正 カメラ：Y回転 [rad]	少し エンディアン
PMDExtrCalib_ca mCal_rotY	1,5707963 27	float32	- 3,1	3,14	よ	0xC80C	12	外因性 の校正 カメラ：Y回転 [rad]	少し エンディアン
PMDExtrCalib_ca mCal_rotZ	0	float32	- 3,1	3,14	よ	0xC80D	13	外因性 の校正 カメラ：Z回転 [rad]	少し エンディアン
PMDExtrCalib_Illu Cal_transX	- 0,085	float32	- 30	30	よ	0xC80E	14	外因性 の校正 カメラ：× 翻訳 [m]	少し エンディアン
PMDExtrCalib_Illu Cal_transY	0,052	float32	- 30	30	よ	0xC80F	15	外因性 の校正 カメラ：Y 翻訳 [m]	少し エンディアン



PMDExtrCalib_Illu Cal_transZ	0,047	float32	- 30	30	よ	0xC810	16	外因性 の校正 カメラ：Z 翻訳 [m]	少し エンディアン
PMDExtrCalib_Illu CalibIsRelative	1	uint8	0	1	よ	0xC811	17	示すフラグ 光るかどうか 校正が行われます カメラを基準に または絶対に 世界座標。	少し エンディアン
LineGuidanceCus t_sprayRemovalS 好感度	0	uint8	0	3	n	0xC812	18	スプレー除去 カスタマイズ	少し エンディアン
LineGuidanceCus t_pixelPlausibiliza しきい値	1	uint8	0	2	n	0xC813	19	ピクセル 正当化 カスタマイズ	少し エンディアン
LineGuidanceCus t_blockageSensiti 虚無	0	uint8	0	3	n	0xC814	20	閉塞 カスタマイズ	少し エンディアン
LineGuidanceCus t_spatialFilterXMi n	- 100	float32	- 100	100	n	0xC815	21	デカルトの空間 フィルター コーディネート、 最小X	少し エンディアン
LineGuidanceCus t_spatialFilterXMa バツ	100	float32	- 100	100	n	0xC816	22	デカルトの空間 フィルター コーディネート、 最大X	少し エンディアン
LineGuidanceCus t_spatialFilterYMi n	- 100	float32	- 100	100	n	0xC817	23	デカルトの空間 フィルター コーディネート、 最小Y	少し エンディアン
LineGuidanceCus t_spatialFilterYMa バツ	100	float32	- 100	100	n	0xC818	24	デカルトの空間 フィルター コーディネート、 最大Y	少し エンディアン
LineGuidanceCus t_spatialFilterZMi n	- 100	float32	- 100	100	n	0xC819	25	デカルトの空間 フィルター コーディネート、 最小Z	少し エンディアン
LineGuidanceCus t_spatialFilterZMa バツ	100	float32	- 100	100	n	0xC81A	26	デカルトの空間 フィルター コーディネート、 最大Z	少し エンディアン
LineGuidanceCus t_reflectorThresho ldValue	0,1	float32	0	1	n	0xC81B	27	設定値 反射率 検出するしきい値 再帰反射器	少し エンディアン
LineGuidanceCus t_lineGuidanceMo で	0	uint8	0	1	n	0xC81C	28	検出されるラインの 種類: 0 - ヒープ検 出、1 - カットエッ ジ	少し エンディアン
LineGuidanceCus t_EgoDataMode	2	uint8	0	2	よ	0xC81D	29	エゴデータの使用 モード - 0: 統合 速度と ヨーレート入力 - 1: 使用のみ 速度 情報	少し エンディアン



								- 2: 速度なし そしてヨーレート 情報	
LineGuidanceCus t_steeringXMin	10	float32	0,3	30	n	0xC81E	30	最小 x のコーディネート 操舵 計算	少し エンディアン
LineGuidanceCus t_steeringXMax	20	float32	0,3	30	n	0xC81F	31	最大 x のコーディネート 操舵 計算	少し エンディアン
LineGuidanceCus t_steeringTTC	2,5	float32	0,3	10	n	0xC820	32	予測時間 [s] 操舵 計算	少し エンディアン
LineGuidanceCus t_maxLineAngle	0,3490658 5	float32	0	3,14	n	0xC821	33	最大 オリエンテーション (傾斜) ラジアン of 線の	少し エンディアン
LineGuidanceCus t_xMin	0	float32	- 100	100	n	0xC822	34	行のみ これと交わる 残るエリアと として与えられる 出力	少し エンディアン
LineGuidanceCus t_xMax	10	float32	- 100	100	n	0xC823	35	行のみ これと交わる 残るエリアと として与えられる 出力	少し エンディアン
LineGuidanceCus t_yMin	- 5	float32	- 100	100	n	0xC824	36	行のみ これと交わる 残るエリアと として与えられる 出力	少し エンディアン
LineGuidanceCus t_yMax	5	float32	- 100	100	n	0xC825	37	行のみ これと交わる 残るエリアと として与えられる 出力	少し エンディアン
LineGuidanceCus t_minHeight	0,3	float32	0,2	2	n	0xC826	38	の最小高さ ヒープ/カットエッジ/... [メートル]	少し エンディアン
LineGuidanceCus t_minWidth	0,25	float32	0	10	n	0xC827	39	の最小幅 ヒープ/カットエッジ/... [メートル]	少し エンディアン
LineGuidanceCus t_maxWidth	3	float32	- 1	10	n	0xC828	40	最大幅 ヒープ/カットエッジ/... [m] (負の 値 <=> 無制限)	少し エンディアン
LineGuidanceCus t_skipStreetPlane 推定	0	uint8	0	1	n	0xC829	41	かどうかを決定します ストリートプレーン 推定 プロセスがスキップされます <=> 平面有効 フラグは 0 に設	少し エンディアン
LineGuidanceCus t_referencePointX 位置	0	float32	- 100	100	n	0xC82A	42	定されます x の値 基準点 ロケーション。	少し エンディアン



LineGuidanceCus t_referencePointY 位置	0	float32	- 100	100	n	0xC82B	43	の y 値 基準点 <small>ロケーション。</small>	少し エンディアン
LineGuidanceCus t_referencePointZ 位置	0	float32	- 100	100	n	0xC82C	44	の z 値 基準点 <small>ロケーション。</small>	少し エンディアン
AutoCalibParam_ numberOfPattern の	0	uint8	0	8	n	0xC82D	45	の数 なるパターン <small>のために使用される</small> 自動校正 (0,1: 自動校正 無効)	少し エンディアン
トリガーされたStreetCal イブレーション	0	uint8	0	1	n	0xC834	52	かどうかを示すフラグ トリガーされた 校正ベース 道路面の推定値は <small>アクティブ</small>	少し エンディアン
lineQualityThresh 古い	0	float32	0	1	n	0xC835	53	適用 閾値 回線品質測定	少し エンディアン
num_Frames_Av 老化する	3	sint32	0	50	n	0xC836	54	ウィンドウサイズ 一時的な <small>pmdのフィルタリング 生データ (0: なし フィルタリング)</small>	少し エンディアン
pixelPlausiReflecti vityThreshold	2.1	float32	0	20	n	0xC837	55	反射率 ピクセルのしきい値 妥当化 (0: <small>フィルタリングなし)</small>	少し エンディアン
変調_周波数 ency_Mode	3	uint8	0	3	よ	0xC838	56	0 = ランダム 13 周波数ホッピング <small>(デフォルト)</small> 1 = 固定トリプル 12 = 固定トリプ ル 23 = 固定トリ	少し エンディアン
amplThresholdFa 制御器	1.0	float32	0.1	10	n	0xC839	57	ブル 3 振幅係数 補償のために 異常な カメラ/照明 組み合わせ	少し エンディアン
LineGuidanceCus t_lineOutputAvera ゲフィルタ	0	float32	0	1	n	0xC83D	61	パラメータは ユーザーが線を滑らかに できるようにする 説明出力	少し エンディアン
CANMaxNumber の線	1	uint8	0	5	n	0xC40A	10	の構成 最大数 <small>のオブジェクトの 上のオブジェクトリスト できる</small>	大きい エンディアン
回転画像90	0	uint8	0	1	よ	0xC83E	62	出力を回転 90度で 時計回りに <small>2Dでのプレゼンテーション</small>	少し エンディアン
Output2D_OnOff	1	int8_t	0	1	n	0xDC03	3	1 : 2D出力がオンにな ります	少し エンディアン



								0: 2D 出力をオフにする	
回転画像90	0	uint8_t	0	1	よ	0xDC06	6	アナログを回転させるビデオ 90 時計回りに度	少し エンディアン
Overlay3DOnOff	1	uint8	0	1	n	0xE001	1	0: GP に送信されるオーバーレイなし 1: 計算するオーバーレイして送信データを GP にオーバーレイする	大きい エンディアン
StartupIconOnOff	1	uint8	0	1	n	0xE002	2	1: アイコンを表示するそして背景色 (スプラッシュスクリーン) で起動 0: 表示しない起動時に何でも	大きい エンディアン
StartupDisplayDuration エーション	2000年	uint32	0	0xFFFF	n	0xE003	3	アイコンを表示する時間と背景色 (スプラッシュスクリーン) で起動	大きい エンディアン
StartupIconID	1	uint8	0	40	n	0xE004	4	どのアイコンにするか起動時に表示 (イニシャル: ifm ロゴ)	大きい エンディアン
FixedGraphicPrimitivesOnOff	1	uint8	0	1	n	0xE006	6	を表示します FixedGraphicPrimitives =1 または表示しない=0	大きい エンディアン
max_number_of_algo_primitives_for_short_primitives	96	uint8	0	96	n	0xE008	8	マックスアルゴショートできるプリミティブから生成される 1つのアルゴリズムの結果 フレーム	大きい エンディアン
max_number_of_algo_primitives_for_long_primitives	0	uint8	0	10	n	0xE009	9	マックスアルゴショートできるプリミティブから生成される 1つのアルゴリズムの結果 フレーム	大きい エンディアン
max_number_of_CAN_input_primitives_for_short_primitives	10	uint8	0	10	n	0xE00A	10	マックスCANショートできるプリミティブに表示される 1フレーム	大きい エンディアン
max_number_of_CAN_input_primitives_for_long_primitives	4	uint8	0	10	n	0xE00B	11	最大CANロングできるプリミティブに表示される 1フレーム	大きい エンディアン
max_number_of_fixed_primitives_for_short_primitives	10	uint8	0	24	n	0xE00C	12	マックスフィックスショートできるプリミティブに表示される 1フレーム	大きい エンディアン
max_number_of_fixed_primitives_for_long_primitives	8	uint8	0	10	n	0xE00D	13	最大固定長できるプリミティブに表示される 1フレーム	大きい エンディアン



DisplayEgoSpeed オンオフ	1	uint8	0	1	n	0xE00E	14	クラッシュの大きな予測オブジェクト のフォーマット: エンディアン 1: 自分のペロシ ティを書く 0: 書き込まない	
DisplayEgoYawra てオンオフ	1	uint8	0	1	n	0xE00F	15	クラッシュビギンのフォーマット 予測されるオブジェクト: 1: 自分のヨーレー トを書く 0: 書き込まない	エンディアン
Display3DFOV	0	uint32	0	3	n	0xE010	16	0 - 表示しない 3D FOV オーバーレイ 1 - 3D 視野を表示 オーバーレイ長方形 2 - 3D 視野を表示 オーバーレイ長方形 「3D FOV」テキスト付 き 3 - 3D FOV を表示 オーバーレイ長方形 とクリップアルゴ 外のプリミティブ グレーのオーバーレイ付き	大きい エンディアン
点滅持続時間 n	10	uint16	1	100	n	0xE012	18	グラフィック プリミティ ブの点滅については、 これは「オン」の持 続時間です。 単位は 100ms です。	大きい エンディアン
点滅オフ持続時間 n	10	uint16	1	100	n	0xE013	19	グラフィック プリミティ ブの点滅については、 これは「オフ」期 間です。 単位は 100ms です。	大きい エンディアン

3.6. スタンバイモード

O3Mセンサーにはスタンバイモードがあります。このモードでは、照明が無効になり、消費電力が削減されます。このモードではオンライン キャリブレーションを使用できません。
可能な応答コードは、「UDS 応答コード」にリストされています。

3.6.1. スタンバイモードに入る

O3Mへのリクエストメッセージ (ID 0x18DA~~FF~~1x、DLC=3)

バイト0	バイト1	バイト2
0x02	コマンド 0x03 0x4A	

3.6.2. スタンバイモードを終了

O3Mへのリクエストメッセージ (ID 0x18DA~~FF~~1x、DLC=3)

バイト0	バイト1	バイト2
0x02	コマンド 0x04 0x4A	



3.7. 教える

O3M がティーチ コマンドを受信すると、ユーザー定義ロジックがデータ値を不揮発性メモリに保存します。unteach コマンドを受信すると、それらを消去します。考えられる応答コード「UDS 応答コード」にリストされています。

3.7.1. コマンドを教える

O3Mへのリクエストメッセージ (ID 0x18DAFF1x、DLC = 3)

バイト0	バイト1	バイト2
0x02	コマンド 0x07 0x4A	

3.7.2. アンティーチコマンド

O3Mへのリクエストメッセージ (ID 0x18DAFF1x、DLC = 3)

バイト0	バイト1	バイト2
0x02	コマンド 0x08 0x4A	

3.8. O3Mリセット

O3Mへのリクエストメッセージ (ID 0x18DAFF1x)、DLC = 8

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
2	コマンド1 0x11		00 埋める バイト	00 バイトを埋める	00 バイトを埋める	00 バイトを埋める	00 バイトを埋める

3.9. UDS 応答コード

OK

バイト0	バイト1
0x01	0x8A

お待ちください

バイト0	バイト1	バイト2	バイト3
0x03	0x7F	0x4A	0x78

エラー

バイト0	バイト1	バイト2	バイト3
0x03	0x7F	0x4A	エラー コード

3.10. UDS 診断情報

UDS サービス 0x22 (ReadDataByIdentifier) を使用します。



識別子 0x07 0x01 を使用して、カメラユニットの温度と電圧を摂氏/ボルトで読み取ります。

- 応答: 0x62 0x07 0x01 プラス 52 データ バイト:

[0:0] 供給電圧

[44:0] PCB の温度

[48:0] フロントエンドの温度

識別子 0x07 0x02 を使用して、照明ユニットの温度と電圧を摂氏で読み取ります /
ボルト。

- 肯定応答: 0x62 0x07 0x02 プラス 12 データ バイト:

[0 : 0] Umin (変調中の電圧)

[4:0] Unom (電圧外部変調)

[8:0] 温度



4. 開けられる

サブディレクトリ「Specs」のファームウェア リリースには、CANopen モードでの O3M のインターフェース仕様を含む EDS ファイルがあります。

ただし、データメッセージのパイロードはJ1939の場合と同じです。単一の信号だけを受信することはできません。つまり、`crash_predictor_info` メッセージの 8 バイト全体を受信し、`CP_object_id` を取得する必要があります。これらのメッセージのパイロードのデータ構造は、EDS ファイルで指定されていません。メッセージのデータ構造 (ビットオフセット、長さなど) については、このドキュメント (J1939 セクション) または DBC ファイルを参照してください。



5. バージョン履歴

バージョン	日付	変更点
V01	2017-09-07	初回リリース
V02	2017-12-04	- Crash Predictor Info メッセージの間違ったオフセットを修正 - CAN/Open に関する短い章を追加
V03	2018-01-11	- LeaveStandbyコマンドの間違ったIDを修正 - 追加されたティーチング/アンティーチ - UDS 応答コードを追加
V04	2018-03-14	- crash_predictor_info のいくつかの修正
V05	2018-10-08	- 「適用のない大きなパラメータ ブロック」のバイナリ形式を修正
V06	2019-01-28	- 一般的なフォーマットの変更 - CANopenセクションの修正
V07	2020-05-19	- 小さな修正を加えたレビュー - 目次を修正 - UDS 診断の章を追加