

CooL4／CCAM 検討 SWG 共通
路側機センサー部インタフェース仕様
version 1.0.0

2023 年 12 月 26 日

自動運転レベル4等先進モビリティサービス
研究開発・社会実装プロジェクト テーマ4(CooL4)
ITS Japan 自動運転研究会 CCAM 検討 SWG

目次

1. 本仕様の目的と策定の考え方.....	2
1.1 本仕様の目的と策定の経緯.....	2
1.2 仕様策定の考え方(CooL4).....	2
1.3 仕様策定の考え方(CCAM 検討 SWG).....	2
2. 参照文献.....	4
3. 用語の定義.....	5
4. センサー部インタフェースの位置付けと構成.....	6
4.1 センサー部インタフェースの位置付け.....	6
4.2 センサー部インタフェース仕様の構成.....	7
5. データフォーマット仕様.....	8
5.1 データフォーマットの構成.....	8
5.2 データ項目の設計原則.....	8
5.3 メッセージのフォーマット.....	9
5.4 センサー情報のデータフォーマット.....	10
5.5 物標情報のデータフォーマット.....	12
5.6 直接検知 freespace 情報のデータフォーマット.....	15
6. 下位レイヤの規定.....	18
6.1 物理層・データリンク層.....	18
6.2 ネットワーク層・トランスポート層.....	18
6.3 データのエンコーディング方法.....	18
6.4 メッセージの送信タイミング.....	18
6.5 時刻同期.....	18
付録 A. データ型の定義.....	19
付録 B. Protocol Buffers のメッセージ定義.....	30
付録 C. Collective Perception Message (CPM) との関係.....	35
付録 D. バージョン履歴.....	37

1. 本仕様の目的と策定の考え方

1.1 本仕様の目的と策定の経緯

本仕様は、協調型自動運転に用いる路側機内で、センサーからデータを取得して認識処理を行うセンサー部と、複数のセンサー部や車両からの情報の統合や、車両に送信するデータの生成を行う路側データ連携モジュール(CCAM 検討 SWG の用語では制御部。以下、路側 DM という)との間のインタフェース仕様を定義するものである。協調型自動運転の実現を主目的としているが、協調型自動運転車のみを目的に路側機を設置するのはコスト的な負担が大きいため、手動運転車やその他のサービスにも有益なものとする 것을目指している。

経済産業省・国土交通省による自動運転レベル 4 等先進モビリティサービス研究開発・社会実装プロジェクト テーマ 4(以下、CooL4)は、協調型のレベル 4 自動運転サービスの実現を目指すプロジェクトで、2021 年に開始された。CooL4 では、路側に設置したセンサーや信号機からの情報を車両に共有するために、データ連携プラットフォーム(データ連携 PF)の開発を行っている。これまでに、データ連携アーキテクチャ(案)[1]とデータ連携 PF API 仕様(案)[2]の策定、データ連携 PF と路側機の開発・評価などを行ってきた。

ITS Japan 自動運転研究会 CCAM 検討 SWG(以下、CCAM 検討 SWG)は、協調型自動運転の社会実装を目指し、路車協調システムの考え方/ポリシーや要件などの協調領域に関する議論・検討を行っている。これまでに、実装ポリシーやユースケースの策定、それらを実現する機能要件や評価指標、評価方法等の検討を行ってきた。

両プロジェクトでは、協調型自動運転に用いる路側機の仕様検討を独立して実施していたが、路側機の開発コストを抑えるためには仕様の共通化が望ましいと考え、協議の結果、路側機内のセンサー部と路側 DM の間のインタフェース仕様を協力して策定することとなった。

以上の経緯から、2023 年 9 月から 12 月にかけて、CooL4 と CCAM 検討 SWG で集中的に検討する場を持ち、本仕様を取りまとめた。

1.2 仕様策定の考え方(CooL4)

本仕様を策定するにあたり、CooL4 で設定した考え方は次の通りである。

将来のユースケースに向けた拡張性を重視し、センサー部が取得・認識した有用と思われる情報を、なるべく落とさずに伝達できるインタフェースとする。このことから、現時点での V2I 通信の制約(通信帯域が狭いことなど)に縛られた仕様とはしない。ただし、現時点のユースケースを実現する際に高コストになることを避けるために、必須のデータ項目を最小限とする。

データの受信側は、高精度地図を持っていることを想定するが、高精度地図がシングルベンダであることは仮定しない。また、独自のデータ項目を追加可能なデータフォーマットとする。

路側機のセンサーからの情報と自己物標情報(車両等が発信する自分自身に関する情報)をフュージョンした情報を扱うことは、本仕様のスコープ外とする。

1.3 仕様策定の考え方(CCAM 検討 SWG)

本仕様を策定するにあたり、CCAM 検討 SWG で設定した考え方は次の通りである。

2025 年での社会実装を目指し、同年度までに技術的に実装可能なものを前提に検討し、自動運転車両側からの要望については可能な限り採用する。また、2025 年度以降、新たな技術が実用化された際に、機能拡張ができるように設計する。

CCAM 検討 SWG がユースケースを成立させるのに必要な情報と考える項目は必須とし、必ずしも必要ではないが性能の向上が期待できるものはオプションとする。また、CooL4 との間で意見が分かれる項目については、その議論の経緯を残した上でオプションとして採用する。

2. 参照文献

- [1] CooL4 データ連携アーキテクチャ(案)version 0.7.0(2023 年 3 月 14 日).
- [2] CooL4 データ連携 PF API 仕様(案)version 0.7.0(2023 年 3 月 14 日).
- [3] ETSI TS 102 894-2 V1.3.1 (2018-08). Intelligent Transport Systems (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary (2018 年 8 月).
- [4] ETSI TR 103 562 V2.1.1 (2019-12). Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2 (2019 年 12 月).
- [5] SAE J2735_202309. V2X Communications Message Set Dictionary(2023 年 9 月).
- [6] ITS Connect 推進協議会: ITS Connect システム 車車間通信メッセージ仕様 ITS Connect TD-001 1.0 版(2015 年 9 月 30 日).
- [7] ISO 23150:2023. Road vehicles – Data communication between sensors and data fusion unit for automated driving functions – Logical interface(2023 年 5 月).

3. 用語の定義

本仕様では、以下の用語を用いる。

物標

検知対象とする種別の物体。

移動物体

自分で移動する能力を持った物体。車両、歩行者など。

非固定物体

自分で移動する能力は持たないが、位置が固定されておらず、移動する可能性のある物体。
落下物、倒木など。

固定物体

位置が固定されており、移動することがない物体。樹木、壁、電柱など。

フリースペース

道路(歩道も含む)の移動可能領域上の物標が存在しない領域。

直接検知フリースペース

背景にある物体が検知できたことにより、物体がないことが検知できた領域。

オクルージョン

センサーで取得される情報が、物体に遮られることで、不完全になること。

誤検知

存在しない物標を存在するものと認識すること。

未検知

存在する物標を見落とすこと。

4. センサー部インタフェースの位置付けと構成

4.1 センサー部インタフェースの位置付け

図 1 に, CoolL4 データ連携アーキテクチャ(案)[1]を示す。また, この図の路側機周りをより詳細化したものを図 2 に示す。

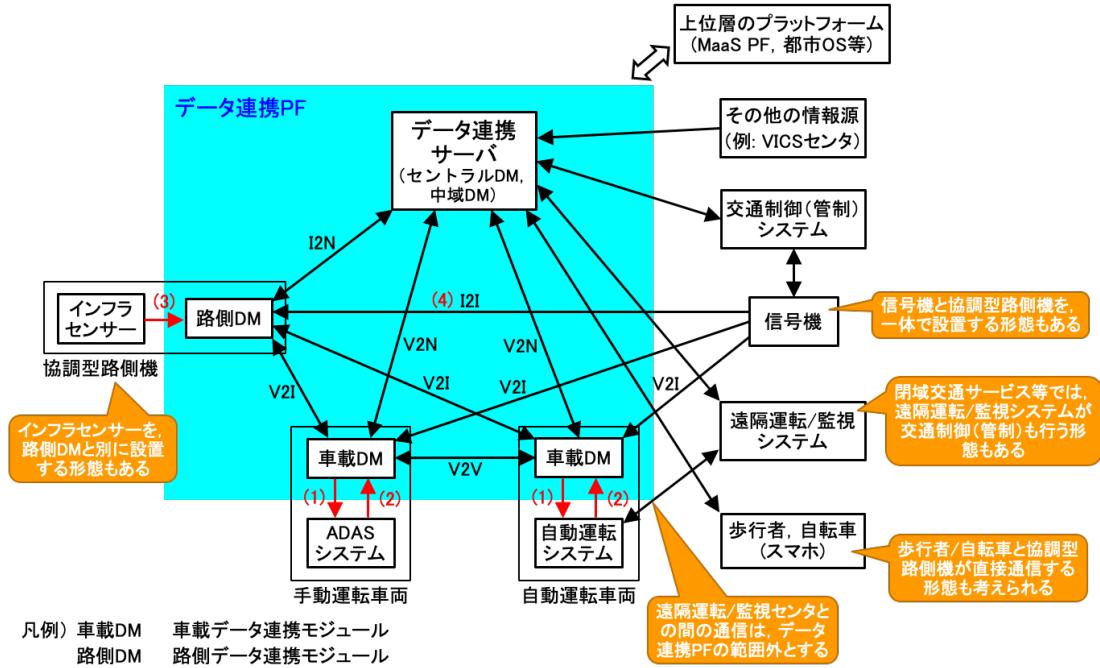


図 1. CoolL4 データ連携アーキテクチャ

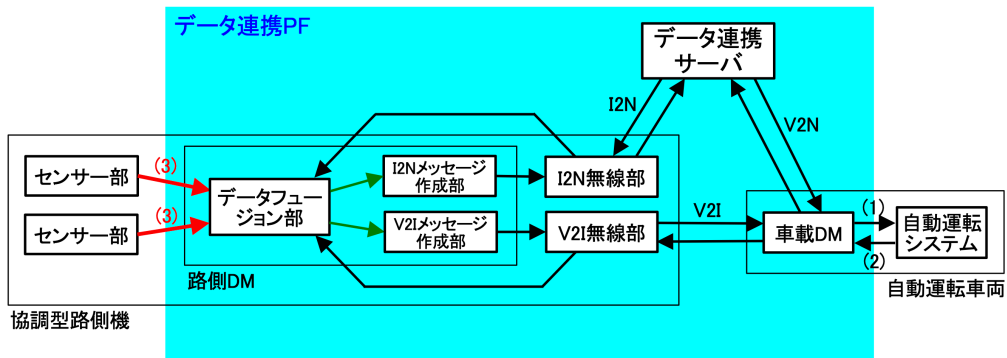


図 2. 路側機周りの詳細図

センサー部は, 路側に設置されたセンサーで取得したデータに対して認識処理を行い, 認識結果をデータフュージョン部に送信する。路側に複数のセンサーを設置する場合には, センサー毎にセンサー部があるものとしても良いし, 1 つのセンサー部が複数のセンサーを持ち, 複数のセンサーからのデータを統合(フュージョン)するものとしても良い。複数のセンサーを持つセンサー部が, センサーからのデータを統合せずにデータフュージョン部に送信する場合には, 論理的には複数のセンサー部がある(言い換えると, 1 つの装置が複数のセンサー部を含んでいる)ものと扱う。

データフュージョン部は, 複数のセンサー部からの情報や車両からの情報を統合し, 無線部を

通して、車両やサーバに送信する。

本仕様では、図 1 および図 2 の(3)のインタフェースを規定する。データフュージョン部が出力するデータのフォーマットや、V2I 通信により車両に送信されるデータのフォーマット等は、本仕様のスコープ外である。

4.2 センサー部インタフェース仕様の構成

本仕様では、(3)のインタフェースに流れるメッセージのデータフォーマットを中心に規定する。また、IP ネットワーク上でメッセージを転送する方法や、データのエンコーディング方法、メッセージの送信タイミング、センサー部と路側 DM の時刻同期の方法についても規定する。物理インタフェース(通信路の物理層・データリンク層)については、具体的な方式を規定するのではなく、要求を提示するにとどめる。また、路側 DM がセンサー部に動作指令を出す場合が考えられるが、そのような場合に用いる路側 DM からセンサー部へのメッセージ仕様については、本仕様のスコープ外とする。

5. データフォーマット仕様

5.1 データフォーマットの構成

センサー部から路側 DM に対して伝達することが必要な情報として、少なくとも、センサー部が検知した車両や歩行者などの物標情報がある。しかし、物標情報のみが伝達された場合、物標情報がない場所(領域)は、物標が存在しないのか、センサーから見えていないのかが判別できない。センサーから見えない理由としては、センサーの検知範囲外であることと、オクルージョンにより見えないことが考えられる。

物標が存在しないのか、センサーから見えていないのかを判別できるようにするには、物標が存在しない領域(フリースペース)を伝達するか、センサーから見えていない領域を伝達する必要がある。センサーから見えていない領域の内、オクルージョンにより見えていない領域は、センサーの設置位置と物体の位置および大きさから計算で求めることができる。

本仕様では、センサー部から路側 DM に、センサーの設置位置と検知領域を伝達することで、物標が存在しない領域とセンサーから見えていない領域を、受信側(路側 DM または車両)で求められるようにする。

ただし、背景にある物体(例えば道路面)をセンサーが検知したことにより検知されるフリースペース(直接検知フリースペース)については、それを受信側で求めるためには、検知した背景物体の情報を伝達する必要があるが、背景物体の形状を伝える必要があるなど、複雑なデータフォーマットになると考えられる。そこで、直接検知フリースペース情報についてはセンサー部で生成することとし、それを伝達するためのデータフォーマットを規定する。

以上より、センサー部から路側 DM には、センサー情報、物標情報、直接検知フリースペース情報の3種類の情報を伝達することとし、本仕様ではそれらのデータフォーマットを規定する。

路側機のセンサーがどのような種別の物体を検知する必要があるかは、ユースケースによって異なるため、本仕様では規定しない。具体的には、移動物体以外の物体については、検知が必須ではないユースケースが多いため、それらの検知はオプション機能とする。センサー部がどの種別の物体を検知できるかを示すために、検知可能物標種別のデータ項目をセンサー情報中に持たせる。

センサー部が複数のセンサーを持つ場合、複数のセンサーからのデータの統合(フュージョン)は、センサー部の役割である。センサー部から路側 DM に送信する物標情報と直接検知フリースペース情報は、複数のセンサーからのデータを統合して認識した結果とする。言い換えると、同じ物標に対する物標情報を、センサー毎に送信してはならない。

5.2 データ項目の設計原則

各情報のデータ項目の策定にあたって、以下の設計原則を設定する。

- 必須のデータ項目を最小限とする。これは、現時点のユースケースを実現する際に高コストになることを避けるためと、センサーの種類や検知状況によって得られるデータ項目が異なる(例えば、初回の検知では移動方向や速さの情報が得られないなど)ためである。なお、以下のデータフォーマットの記述で、「必須」と記載されていないデータ項目は、すべてオプションである。
- 必須のデータ項目以外のすべてのデータ項目に、そのデータ項目が設定されていないことを示す「不明」を表す値を入れられるようにする。ただし、データ項目が設定されていないことを他の方法で表せる場合(6.3 節で規定するエンコーディング方法はこれに該当する)には、

「不明」を表す値は使用しない。

- 認識によって得られるすべての数値データ項目に、精度の情報を持たせるようにする。これは、複数の情報源からの情報を統合する時に必要となるためである。精度は数値で表現し、精度が x であるとは、誤差(真値と計測値の差)の絶対値が x 以下である確率が 95%になることを言う。例えば、物標情報に含まれる速さが 10m/s、速さの精度が 1m/s の場合、真の速さが 10m/s \pm 1m/s の区間に入る確率が 95%であることを表す。計測値が正規分布に従う場合、精度は、誤差の標準偏差の約 2 倍(いわゆる 2σ)となる。
ただし、センサーで取得したデータから、誤差の絶対値が x 以下である確率が 95%(以上)であることを「保証」できる x を求める手法は確立していない。各数値データの精度を、センサーから物標まで距離や物標を検知できた回数などの状況毎にあらかじめ計測して表にしておき、その表を引くといった方法で精度の値を生成することを想定している。
- 長さの単位は、原則として 0.01m(=1cm)とする。
- 方位角の単位は、原則として 0.0125 度とする。ただし、回転速度の単位は、0.01 度/s とする。

以降のデータフォーマットの規定には明示しないが、いずれの情報に対しても、独自のデータ項目を追加して良い。

5.3 メッセージのフォーマット

センサー部から路側 DM に送信するメッセージは、次の構成とする。

- メッセージヘッダ[必須]
- センサー情報のリスト[必須](リストのサイズは 1 以上)
- 物標情報のリスト(リストのサイズは 0 以上)
- 直接検知フリースペース情報のリスト(リストのサイズは 0 以上)

メッセージヘッダは、次のデータ項目で構成される。

- メッセージ ID[必須]
- プロトコルバージョン[必須]
- メッセージカウンタ[必須]
- センシング時刻[必須]
- センサー部異常通知
- センサー部異常コード

以下では、メッセージヘッダのそれぞれのデータ項目について説明する。

メッセージ ID

メッセージの種類を表す識別番号。本仕様に規定されたメッセージにおいては、1 を設定する。
メッセージ ID は、8 ビットの符号なし整数で表現する。

プロトコルバージョン

データフォーマットのバージョンを表す識別番号。本仕様に規定されたメッセージにおいては、1 を設定する。

プロトコルバージョンは、8 ビットの符号なし整数で表現する。

メッセージカウンタ

メッセージの順序を示す番号。メッセージを送信する度にインクリメントする。255 の次は 0 に戻す。

メッセージカウンタは、8 ビットの符号なし整数で表現する。

センシング時刻

センサーが情報を取得した時刻。

センサーが情報を取得するのに時間がかかる場合には、情報取得を開始してから終了するまでの間の任意の時刻を用いる。

センシング時刻は、DE_TimestampIts で表現する。

センサー部異常通知

センサー部の異常状態、サービス提供状態、機能停止準備中、要求動作、自己動作を表す。

センサー部異常通知は、DE_ErrorNotification で表現する。

センサー部異常コード

センサー部の異常内容を示すためのコード。コードの意味は、センサー部のメーカーが定義する。

センサー部異常コードは、24 ビットの符号なし整数で表現する。

5.4 センサー情報のデータフォーマット

センサー情報とは、センサー部が持つセンサーの設置位置や検知性能などに関する情報であり、メッセージ中には 1 つ以上のセンサー情報を含む。

センサー情報には、物標を検知することができる領域(検知領域)に関する情報を複数含むことができ、検知領域毎に、センサーの検知能力(検知できる物体の種別や検知信頼度)に関する情報を持つ。センサーの検知領域と検知能力を含むデータ単位を、検知性能情報と呼ぶ。センサー情報中に検知性能情報を複数持つことができるのは、センサーの検知領域が複数に分かれている場合や、検知領域によって検知能力が異なる場合に対応できるようにするためである。

センサーの検知領域には、検知対象でない物体によるオクルージョンで見えない領域は含めないものとする(図 3)。検知領域は頂点数が 16 以下の多角形で表現するが、検知領域が複雑な形状となりそれでは表現できない場合には、複数の検知領域に分割するか、小さくなる方向に近似する。

センサーの検知性能は、その時の状況(天候やセンサーのレンズの汚れなど)によって変化する動的な情報である。検知対象の物標の属性(大きさ、光や電波の反射率など)によっても検知領域や検知信頼度は変化するため、センサーの検知性能を正確に求めることは難しい。

検知可能物標種別

検知可能な物体の種別。四輪車, 列車, 自動二輪車, 軽車両, 人, 動物, 非固定物体, 固定物体が, それぞれ検知できるかどうかを表す。

検知可能物標種別は, `DE_DetectableClasses` で表現する。

検知領域

センサーが物標の存在を検知できる道路面の領域。

道路面の領域は多角形で表現する。多角形は頂点のリストで表現し, 最初の頂点と最後の頂点をつないだ内側の領域を表すものとする。各頂点の位置は, センサー設置位置からの相対座標で表現し, 頂点のリストのサイズは 3 以上 16 以下とする。各頂点の標高は静的地図からわかるため, 頂点の標高を表すデータ項目は持たない。

検知領域は, 頂点の位置を表す `DF_OffsetPointXY` のリストで表現する。

検知信頼度

センサーの検知領域内で, 検知限界物標サイズより大きい物標の存在を検知できる確率。言い換えると, 未検知率を 1 から減じた値。

検知信頼度を正確に求める手法は確立していない。ある信頼度が達成できない場合に, 情報を出力しなくなるセンサー部の場合, その信頼度の値(固定値)を用いれば良い。

検知信頼度は, `DE_Confidence` で表現する。

検知限界物標サイズ

センサーの検知領域内で, このサイズ以下の物標は見落とす可能性があるというサイズ。

検知限界物標サイズは, `DE_ObjectDimensionValue` で表現する。

センサー状態

センサーの稼働状態(正常稼働中か縮退稼働中か停止中か)と運用状態(運用中か試験中か)。

センサー状態は, `DE_SensorStatus` で表現する。

5.5 物標情報のデータフォーマット

物標情報とは, 道路上およびその周辺に存在する物体の位置や状態に関する情報であり, メッセージ中には任意の数の物標情報を含む。

物標情報は, 次のデータ項目で構成される。

- 物標 ID[必須]
- 情報取得時刻
- 物標種別のリスト(リストのサイズは 0 以上 4 以下)
- 存在信頼度
- 物標位置[必須]とその精度
- 物標参照位置
- 移動方向とその精度
- 速さとその精度

- 回転速度とその精度
- 前後加速度とその精度
- 物標の向きとその精度
- 物標のサイズとその精度
- 物標の静止状態
- トラッキング状態[必須]
- 検知回数
- 連続未検知回数
- 生存時間

以下では、上のそれぞれのデータ項目について説明する。

物標 ID

センサーが、物標の同一性を示すため(トラッキングのため)に付与する ID。物標 ID は、ある時刻においてはユニークになるように割り付ける(言い換えると、異なる物標に同じ ID 割り付けない)ものとする。

物標 ID は、16 ビットの符号なし整数で表現する。

情報取得時刻

物標情報を得た時刻。言い換えると、センサーが物標を観測した時刻。

情報取得時刻は、メッセージヘッダ中のセンシング時刻からの相対値で表現する。情報取得時刻がセンシング時刻と一致している場合には、情報取得時刻を設定する必要はない。

情報取得時刻は、物標の位置と整合していることが重要である。認識処理等で遅れが生じる場合には、遅れ分を補正しても良い。

情報取得時刻は、DE_TimeOfMeasurement で表現する。

物標種別のリスト

2 段階で表現した物標の種別と段階毎の分類信頼度のリスト。リストのサイズは 0 以上 4 以下とする。リストのサイズが 0 の場合、物標の種別が不明であることを表す。

1 段階目の種別では、物標が不明、四輪車、列車、自動二輪車、軽車両、人、動物、非固定物体、固定物体のいずれに分類されるかを表す。不明以外の場合には、2 段階目の種別で、さらに詳しい種別を表す。

分類信頼度は、1 段階目、2 段階目のそれぞれに対して、パーセント値で表現する。

2 段階目の種別に、物標の群を表す種別があるが、この種別の必要性和使い方は次の通りである。センサーで多くの車両や人が検知できた場合に、それらを個別の物標情報として表すとメッセージサイズが大きくなり、通信帯域の制約などにより送信できなくなる場合が考えられる。このような場合に、送信する物標情報を間引くと、受信側では、本来は物標が存在する場所をリースペースと誤認する可能性がある。これを防ぐためには、物標情報を間引くのではなく、類似の動きをする複数の物標情報を統合する方法が望ましい。複数の物標を統合した物標情報は、物標種別を物標の群を表す種別とすることで区別する。

物標種別を物標の群を表す種別とした場合、物標 ID は、物標の群に対して付与する。物標位置、物標参照位置、物標の向き、物標のサイズの各データ項目で、物標の群全体のバウンデ

イングボックスを表す。移動方向、速さ、回転速度、前後加速度は、物標の群に属する物標で、これらのデータがほぼ同じ値であればその値を用い、そうでなければ不明(または値を設定しない)とする。

物標種別は、DF_ObjectClass で表現する。

存在信頼度

物標情報が表す物体が実際に存在する確率。言い換えると、誤検知率を 1 から減じた値。

存在信頼度を正確に求める手法は確立していない。ある信頼度が達成できない場合に、物標情報を出力しなくなるセンサー部の場合、その信頼度の値(固定値)を用いれば良い。

存在信頼度は、DE_Confidence で表現する。

物標位置とその精度

物標の位置とその精度。

物標位置は、日本測地系 2011 (JGD2011)に基づく経度緯度標高で表現する。

位置の精度は、水平方向位置精度を表すための 95%の確率で正しい位置を含む楕円(95%信頼楕円)と、標高の精度で表現する。楕円は、長半径、短半径、長軸の方位角で表現する。

物標位置とその精度は、DF_Position で表現する。

物標参照位置

物標位置が、物標のどの場所を表しているか。

認識した物標の位置をできる限り正確に伝えるために、位置を最も正確に特定できる場所を参照位置とする。具体的には、センサーから見えている場所を参照位置とすることを想定している。認識した物標の全体が見えている場合には、車両の場合は前面の中心の底辺位置、人の場合は中央の底辺位置を使う。どの面が見えているかわからない場合には、見えている面を前面であるものと扱う。

物標参照位置は、DE_RefPoint で表現する。

移動方向とその精度

物標の移動方向とその精度。移動方向は、WGS84 測地系における方位角で表現する。

移動方向は DE_AzimuthValue、移動方向の精度は DE_AzimuthAccuracy で表現する。

速さとその精度

物標の移動の速さとその精度。

速さが負の値の場合は、物標が後退していることを表す。

速さは DE_SpeedValue、速さの精度は DE_SpeedAccuracy で表現する。

回転速度とその精度

物標の回転速度(ヨーレート)とその精度。

回転速度は DE_YawRateValue、回転速度の精度は DE_YawRateAccuracy で表現する。

前後加速度とその精度

物標の前後加速度(速さの時間微分)とその精度。

前後加速度は DE_AccelerationValue、前後加速度の精度は DE_AccelerationAccuracy で表

現する。

物標の向きとその精度

物標の向きとその精度。向きは、WGS84 測地系における方位角で表現する。

物標の向きは、移動方向と一致するとは限らない。向きが明確でない物標に対しては、物標のサイズを表現するために用いる情報と位置付ける。

向きは DE_AzimuthValue, 向きの精度は DE_AzimuthAccuracy で表現する。

物標のサイズとその精度

物標のサイズとその精度。具体的には、物標の向きに対する物標のバウンディングボックスの長さ、長さの精度、幅、幅の精度、高さ、高さの精度。

長さ、幅、高さはそれぞれ DE_ObjectDimensionValue, 長さの精度、幅の精度、高さの精度はそれぞれ DE_ObjectDimensionAccuracy で表現する。

物標の静止状態

物標が静止しているか、静止している場合には、どれだけの時間静止しているかを示す。

物標位置の変化が、センサーの位置認識誤差以内である場合に、物標が静止しているものとする。

静止状態は、DE_StaticStatus で表現する。

トラッキング状態

最新のセンシング状態(物標を検知したかどうか)、検知しなかった場合にその理由、物標情報の削除予告、物標情報が融合または分割されたかを表す。トラッキング状態は必須のデータ項目であるが、最新のセンシング状態以外はオプションとする。

トラッキング状態は、DE_TrackingStatus で表現する。

検知回数

物標を検知した合計回数。物標を複数のセンサーで検知した場合は、センサー毎に検知回数を増加させる。

検知回数は、DE_DetectionCount で表現する。

連続未検知回数

物標を連続して検知できなかった回数。

連続未検知回数は、DE_LostCount で表現する。

生存時間

物標が最初に検知されてからの経過時間。

生存時間は、DE_ObjectAge で表現する。

5.6 直接検知 freespace 情報のデータフォーマット

直接検知 freespace 情報とは、センサー部が背景にある物体を検知したことにより検知された freespace に関する情報であり、メッセージ中には任意の数の直接検知 freespace 情報を含む。

路側センサーは、道路の上方に設置することが一般的であるため、道路面が認識できれば、そ

の上には物体がないこと(フリースペースであること)を高い信頼度で検知することができる。

規定値よりも小さい直接検知フリースペースは、検知不要とする。具体的には、フリースペース領域が直径 5m の円に収まるサイズの直接検知フリースペースは、検知する必要がないものとする。

なお、直接検知フリースペース情報を自動運転車両の制御に活用する方法やその有用性については、現時点では明らかになっていない。

直接検知フリースペース情報は、次のデータ項目で構成される。

- 情報取得時刻
- フリースペース領域[必須]とその精度
- 存在信頼度
- 検知限界物標サイズ

なお、フリースペースをトラッキングすることはないため、フリースペースには ID を付与しない。

以下では、上のそれぞれのデータ項目について説明する。

情報取得時刻

直接検知フリースペース情報を得た時刻。言い換えると、センサーがフリースペースを観測した時刻。

情報取得時刻は、メッセージヘッダ中のセンシング時刻からの相対値で表現する。情報取得時刻がセンシング時刻と一致している場合には、情報取得時刻を設定する必要はない。

情報取得時刻は、フリースペースの位置と整合していることが重要である。認識処理等で遅れが生じる場合には、遅れ分を補正しても良い。

情報取得時刻は、DE_TimeOfMeasurement で表現する。

フリースペース領域とその精度

物体が存在しない道路面の領域とその精度。

道路面の領域は多角形で表現する。多角形は頂点のリストで表現し、最初の頂点と最後の頂点をつないだ内側の領域を表すものとする。最初の頂点の位置は、日本測地系 2011 (JGD2011) に基づく経度緯度標高で表現し、2 番目以降の各頂点の位置は、最初の頂点の位置からの相対座標で表現する。2 番目以降の頂点のリストのサイズは 2 以上 15 以下とする。2 番目以降の各頂点の標高は静的地図からわかるため、頂点の標高を表すデータ項目は持たない。

フリースペース領域の精度は、各頂点に個別に持つのではなく、すべての頂点に対して 1 つのみ持つ。具体的には、各頂点の水平方向位置精度を表すための 95%信頼楕円をすべて含む楕円と、各頂点の標高の精度の最大値で表現する。楕円は、長半径、短半径、長軸の方位角で表現する。

最初の頂点の位置とフリースペース領域の精度は DF_Position で、2 番目以降の頂点の位置は DF_OffsetPointXY のリストで表現する。データフォーマット上では、フリースペース領域の精度は、最初の頂点の位置の精度として表す。

存在信頼度

直接検知フリースペース情報が表す領域内に、検知限界物標サイズより大きい物体が実際に存在しない確率。言い換えると、存在する物体を見落とす確率を 1 から減じた値。

存在信頼度を正確に求める手法は確立していない。ある信頼度が達成できない場合に、直

接検知 freespace 情報を出力しなくなるセンサー部の場合、その信頼度の値(固定値)を用い
れば良い。

存在信頼度は、DE_Confidence で表現する。

検知限界物標サイズ

freespace 領域内で、このサイズ以下の物標は見落とす可能性があるというサイズ。

検知限界物標サイズは、DE_ObjectDimensionValue で表現する。

6. 下位レイヤの規定

6.1 物理層・データリンク層

センサー部と路側 DM を接続する通信路の物理層・データリンク層は、本仕様では規定しない。ただし、IP 通信が可能で、おおよそ 10Mbps 以上の転送レートを持つものであることを要する。

6.2 ネットワーク層・トランスポート層

ネットワーク層には IP (IPv4 または IPv6)、トランスポート層には UDP を使用する。使用する IP アドレスやポート番号は、本仕様では規定しない。

6.3 データのエンコーディング方法

1 回のセンシングで認識された情報 (センサー情報、物標情報、直接検知 freespace 情報) を入れたメッセージを、1 つの UDP データグラムに入れて送信する。メッセージが、UDP データグラムの最大長 (約 64KB) を超えるケースは想定していない。

メッセージを UDP データグラムに入れるためのエンコーディングには、Protocol Buffers (バージョン 3.15 以降) を使用する。

Protocol Buffers のメッセージ定義を付録 B に示す。Protocol Buffers のメッセージ定義において、オプションのデータ項目を optional フィールドとすることで、オプションのデータ項目を設定しないことを optional フィールドに値を設定しないことで表せるようにしている。そのため、各データ項目の「不明」を表す数値は使用しない。ただし、物標種別の 2 段階目が「不明」であることを表す数値は使用する。

本仕様で規定するデータフォーマットに独自のデータ項目を追加する場合には、付録 B に示すメッセージ定義に対して、独自のフィールドを追加する。その場合、独自のフィールドに対するフィールド番号を、1000 以上の値とすること。

6.4 メッセージの送信タイミング

センサー部から路側 DM へは、周期的にメッセージを送信する。メッセージの送信周期は、本仕様では規定しない。

静止している物標については、物標情報を送る間隔を長くしても良い。物標情報中の静止状態は、物標情報を送る間隔を決めるために用いることができる。具体的な方法は、本仕様では規定しない。

6.5 時刻同期

センサー部は、路側 DM または無線部をマスターとして、PTP (Precision Time Protocol, IEEE 1588-2008) を用いた時刻同期を行う。

付録 A. データ型の定義

ここでは、データフォーマット中で使用するデータ型について規定する。

DE_TimestampIts

UTC(協定世界時)で 2004 年 1 月 1 日 0 時 0 分 0 秒からのミリ秒単位の経過時間。

このデータ型は閏秒も数えるため、UTC との差は、閏秒が挿入されるたびに 1 秒ずつ大きくなる。そのため、UTC や UNIX 時間(UTC で 1970 年 1 月 1 日 0 時 0 分 0 秒からのミリ秒単位の経過時間。閏秒を数えない)と変換する場合には、閏秒がいつ挿入されたかの情報が必要になる。GPS 時刻(UTC で 1980 年 1 月 1 日 0 時 0 分 0 秒からの経過時間)とは、機械的に変換できる。

42 ビットの符号なし整数で表現できる。

このデータ型は、文献[3]のものを採用した。

DE_ErrorNotification

センサー部の異常状態、サービス提供状態、機能停止準備中、要求動作、自己動作を表すデータ型。それぞれの状態を表現する値のビット毎論理和で表す。

センサー部の異常状態の具体的な表現値は以下の通り。

- 0: 正常動作中
- 0x01: 異常発生中

センサー部のサービス提供状態の具体的な表現値は以下の通り。

- 0: サービス提供中
- 0x02: サービス縮退中
- 0x04: サービス停止中

センサー部が機能停止準備中であることの表現値は以下の通り。

- 0: 通常動作中
- 0x08: 機能停止準備中

センサー部の要求動作の具体的な表現値は以下の通り。

- 0: 要求なし
- 0x10: 電源 OFF/ON 要求
- 0x20: リセット指示要求
- 0x30: 状態変更指示要求

センサー部の自己動作の具体的な表現値は以下の通り。

- 0: 通常動作
- 0x40: 自己リセット予告
- 0x80: 自己状態変更予告
- 0xc0: 自己復旧中

DE_SensorType

センサーの種類を表すデータ型。

具体的な表現値は以下の通り。

- 0: 不明(数値で表現する場合)

- 1: レーダー
- 2: LiDAR
- 3: 単眼カメラ
- 4: ステレオカメラ
- 5: ナイトビジョン
- 6: 超音波センサー
- 7: PMD センサー
- 8: 複数のセンサーのフュージョン
- 9: 誘導ループ
- 10: 球面カメラ

このデータ型は、文献[4]のものに修正を加えたものである。

DF_DetectCapability

検知性能情報を表すデータ型で、検知可能物標種別、検知領域、検知信頼度、検知限界物標サイズで構成される。

検知可能物標種別は DE_DetectableClasses で、検知領域は DF_OffsetPointXY のリストで、検知信頼度は DE_Confidence で、検知限界物標サイズは DE_ObjectDimensionValue で表現する。

DE_DetectableClasses

検知できる物標種別の集合を表すデータ型。

8 ビットの符号なし整数の各ビットで、それぞれ、四輪車、列車、自動二輪車、軽車両、人、動物、非固定物体、固定物体が検知できるかどうかを示す。

具体的なビットと物標の種別の対応は以下の通り。

- ビット 0 (LSB): 四輪車
- ビット 1: 列車
- ビット 2: 自動二輪車
- ビット 3: 軽車両
- ビット 4: 人
- ビット 5: 動物
- ビット 6: 非固定物体
- ビット 7 (MSB): 固定物体

DF_OffsetPointXY

平面上での相対位置を表すデータ型で、WGS84 測地系における東西方向の距離(dx, 東が正)と南北方向の距離(dy, 北が正)で構成される。

東西方向と南北方向の距離は、それぞれ DE_DistanceValue で表現する。

DE_DistanceValue

相対位置を示すための距離を 0.01m 単位で表すデータ型。

具体的な表現値は以下の通り。

- 132768: 不明(数値で表現する場合)
- 132767: -1327.67m 以下
-

-1:	-0.01m
0:	0m
1:	0.01m
.....	
132767:	1327.67m 以上

このデータ型は、文献[4]のものに修正を加えたものである。

DE_SensorStatus

センサーの稼働状態と運用状態を表すデータ型。それぞれの状態を表現する値のビット毎論理和で表す。

センサーの稼働状態の具体的な表現値は以下の通り。

0:	正常稼働中
0x1:	縮退稼働中
0x2:	停止中

センサーの運用状態の具体的な表現値は以下の通り。

0:	運用中
0x4:	試験中

DE_TimeOfMeasurement

基準時刻との時間差をミリ秒単位で表すデータ型。時刻を相対値で表すために用いる。

具体的な表現値は以下の通り。

-1500:	基準時刻の 1500ms 前
.....	
-1:	基準時刻の 1ms 前
0:	基準時刻
1:	基準時刻の 1ms 後
.....	
1500:	基準時刻の 1500ms 後

このデータ型は、文献[4]のものを採用した。

DF_ObjectClass

物標種別を表すデータ型で、物標の 1 段階目の種別(不明, 四輪車, 列車, 自動二輪車, 軽車両, 人, 動物, 非固定物体, 固定物体), 1 段階目の分類信頼度, 2 段階目の種別, 2 段階目の分類信頼度で構成される。

1 段階目の種別は、2 段階目のどの種別に関するデータがあるかで表現する。2 段階目の種別は、1 段階目が四輪車の場合は DE_VehicleSubclassType で、列車の場合は DE_TrainSubclassType で、自動二輪車の場合は DE_MotorcycleSubclassType で、軽車両の場合は LightVehicleSubclassType で、人の場合は DE_PersonSubclassType で、動物の場合は DE_AnimalSubclassType で、非固定物体の場合は DE_NfoSubclassType で、固定物体の場合は DE_FoSubclassType で表現する。2 段階目の種別に関するデータがない場合には、1 段階目の種別が不明であることを示す。また、分類信頼度は、DE_ClassConfidence で表現する。

このデータ型は、文献[4]のものに拡張・修正を加えたものである。

DE_VehicleSubclassType

車両の種別を表すデータ型。具体的な表現値は以下の通り。

- 0: 不明(数値で表現する場合)
- 1: 乗用車
- 2: バス
- 3: 小型トラック
- 4: 大型トラック
- 5: 牽引車(牽引されている車両)
- 6: 特殊車両
- 7: 緊急車両
- 8: 農業用車両
- 9: 四輪車群

このデータ型は、文献[4]のものに修正を加えたものである。

DE_TrainSubclassType

列車の種別を表すデータ型。具体的な表現値は以下の通り。

- 0: 不明(数値で表現する場合)
- 1: 路面電車
- 2: その他の列車

DE_MotorcycleSubclassType

自動二輪車の種別を表すデータ型。具体的な表現値は以下の通り。

- 0: 不明(数値で表現する場合)
- 1: 原付
- 2: 自動二輪車
- 3: 自動二輪車群

DE_LightVehicleSubclassType

軽車両の種別を表すデータ型。具体的な表現値は以下の通り。

- 0: 不明(数値で表現する場合)
- 1: 自転車
- 2: 人力車
- 3: 荷車
- 4: キックボード
- 5: 軽車両群

DE_PersonSubclassType

人の種別を表すデータ型。具体的な表現値は以下の通り。

- 0: 不明(数値で表現する場合)
- 1: 歩行者
- 2: 車椅子
- 3: シニアカー
- 4: ベビーカー(とそれを動かしている人)
- 5: スケート/スケートボードに乗っている人

6: 人の群

このデータ型は、文献[4]のものに修正を加えたものである。

DE_AnimalSubclassType

動物の種別を表すデータ型。具体的な表現値は以下の通り。

0: 不明(数値で表現する場合)

このデータ型は、文献[4]のものを採用した。

DE_NfoSubclassType

非固定物体の種別を表すデータ型。具体的な表現値は以下の通り。

0: 不明(数値で表現する場合)

DE_FoSubclassType

固定物体の種別を表すデータ型。具体的な表現値は以下の通り。

0: 不明(数値で表現する場合)

DE_ClassConfidence

分類の信頼度をパーセント値で表現したもの。

0: 不明(数値で表現する場合)

1: 1%

.....

100: 100%

このデータ型は、文献[4]のものに修正を加えたものである。

DE_Confidence

信頼度(検知信頼度, 存在信頼度)を表すデータ型。

信頼度を表す確率から、以下の式で求めた値を切り上げた整数値で表現する。ただし、値が101以上となる場合には、101とする。また、不明を数値で表現する場合には、0を用いる。

$$-10 \cdot \log_{10}(1 - \text{信頼度を表す確率})$$

具体的には、DE_Confidence の値と信頼度を表す確率の関係は以下のようになる。

0: 不明(数値で表現する場合)

1: ~20.6%

2: ~36.9%

3: ~49.9%

.....

9: ~87.4%

10: ~90.0%

11: ~92.06%

.....

99: ~99.999999987%

100: ~99.999999990%

101: それより大きい

DF_Position

位置とその精度を表すデータ型。

経度, 緯度, 標高, 水平方向位置精度楕円の長半径(長軸の長さの半分), 短半径(短軸の長さの半分), 長軸の方位角, 標高の精度で構成される。

経度は DE_Longitude, 緯度は DE_Latitude, 標高は DE_AltitudeValue, 水平方向位置精度楕円の長半径と短半径は DE_SemiAxisLength, 長軸の方位角は DE_AzimuthValue, 標高の精度は DE_AltitudeAccuracy で表現する。

DE_Longitude

経度を 0.1 マイクロ度単位で表すデータ型。

具体的な表現値は以下の通り。

-1800000000:	西経 180 度
.....	
-1:	西経 0.1 マイクロ度
0:	本初子午線上
1:	東経 0.1 マイクロ度
.....	
1800000000:	東経 180 度
1800000001:	不明(数値で表現する場合)

このデータ型は, 文献[3]のものを採用した。

DE_Latitude

緯度を 0.1 マイクロ度単位で表すデータ型。

具体的な表現値は以下の通り。

-900000000:	南緯 90 度(南極)
.....	
-1:	南緯 0.1 マイクロ度
0:	赤道上
1:	北緯 0.1 マイクロ度
.....	
900000000:	北緯 90 度(北極)
900000001:	不明(数値で表現する場合)

このデータ型は, 文献[3]のものを採用した。

DE_AltitudeValue

標高を 0.01m 単位で表すデータ型。

具体的な表現値は以下の通り。

-100000:	-1000m 以下
.....	
-1:	-0.01m
0:	準拋楕円体上
1:	0.01m
.....	

80000: 8000m 以上
 80001: 不明(数値で表現する場合)

このデータ型は、文献[3]のものを採用した。

DE_AltitudeAccuracy

標高の精度(誤差がそれ以下である確率が 95%になる値)を 0.01m 単位で表すデータ型。

具体的な表現値は以下の通り。

1: 0.01m

 20000: 200.0m 以上
 20001: 不明(数値で表現する場合)

DE_SemiAxisLength

精度を表す楕円の半径を 0.01m 単位で表すデータ型。

具体的な表現値は以下の通り。

1: 0.01m

 4093: 40.93m
 4094: 40.94m 以上
 4095: 不明(数値で表現する場合)

このデータ型は、文献[3]のものを採用した。

DE_AzimuthValue

WGS84 測地系における方位角を 0.0125 度単位で表すデータ型。

具体的な表現値は以下の通り。

0: 北

 7200: 東

 14400: 南

 21600: 西

 28800: 不明(数値で表現する場合)

このデータ型は、文献[5]の DE_Angle を採用した。

DE_AzimuthAccuracy

WGS84 測地系における方位角の精度(誤差がそれ以下である確率が 95%になる値)を 0.0125 度単位で表すデータ型。

具体的な表現値は以下の通り。

1: 0.0125 度

 80: 1.00 度

7199:	89.9875 度
7200:	90.00 度以上
7201:	不明(数値で表現する場合)

DE_RefPoint

物標位置が、物標のどの場所を表しているかを表すデータ型。

具体的な表現値は以下の通り。

0:	不明(数値で表現する場合)
1:	中央(バウンディングボックスの中央)の底面位置
2:	前面の中心の底面位置
3:	右前角の底面位置
4:	右側面の中心の底面位置
5:	右後角の底面位置
6:	背面の中心の底面位置
7:	左後角の底面位置
8:	左側面の中心の底面位置
9:	左前角の底面位置

地面に接している物体においては、底面位置とは地面位置のことである。

DE_SpeedValue

速さを 0.01m/s 単位で表すデータ型。

具体的な表現値は以下の通り。

-16382:	-163.82m/s 以下(=-589.752km/h 以下)
-16381:	-163.81m/s
.....	
-1:	-0.01m/s(=1cm/s)
0:	静止
1:	0.01m/s(=1cm/s)
.....	
16381:	163.81m/s
16382:	163.82m/s 以上(=589.752km/h 以上)
16383:	不明(数値で表現する場合)

このデータ型は、文献[4]の DE_SpeedValueExtended に修正を加えたものである。

DE_SpeedAccuracy

速さの精度(誤差がそれ以下である確率が 95%になる値)を 0.01m/s 単位で表すデータ型。

具体的な表現値は以下の通り。

1:	0.01m/s(=1cm/s)
.....	
16381:	163.81m/s
16382:	163.82m/s 以上(=589.752km/h 以上)
16383:	不明(数値で表現する場合)

DE_YawRateValue

回転速度(ヨーレート)を 0.01 度/s 単位で表すデータ型。

具体的な表現値は以下の通り。

-32766:	右へ 327.66 度/s 以上
-32765:	右へ 327.65 度/s
.....	
-1:	右へ 0.01 度/s
0:	直進
1:	左へ 0.01 度/s
.....	
32765:	左へ 327.65 度/s
32766:	左へ 327.66 度/s 以上
32767:	不明(数値で表現する場合)

このデータ型は、文献[3]のものを採用した。

DE_YawRateAccuracy

回転速度(ヨーレート)の精度(誤差がそれ以下である確率が 95%になる値)を 0.01 度/s 単位で表すデータ型。

具体的な表現値は以下の通り。

1:	0.01 度/s
.....	
32765:	327.65 度/s
32766:	327.66 度/s 以上
32767:	不明(数値で表現する場合)

DE_AccelerationValue

前後加速度を 0.01m/s² 単位で表すデータ型。

具体的な表現値は以下の通り。

-2000:	20.00m/s ² 以上で減速
-1999:	19.99m/s ² で減速
.....	
-1:	0.01m/s ² で減速
0:	一定速度
1:	0.01m/s ² で加速
.....	
1999:	19.99m/s ² で加速
2000:	20.00m/s ² 以上で加速
2001:	不明(数値で表現する場合)

DE_AccelerationAccuracy

前後加速度の精度(誤差がそれ以下である確率が 95%になる値)を 0.01m/s² 単位で表すデータ型。

具体的な表現値は以下の通り。

1:	0.01m/s ²
.....	
999:	9.99m/s ²
1000:	10.00m/s ² 以上
1001:	不明 (数値で表現する場合)

DE_ObjectDimensionValue

物標の一辺の長さを 0.01m 単位で表すデータ型。

具体的な表現値は以下の通り。

1:	0.01m
.....	
65534:	655.34m
65535:	不明 (数値で表現する場合)

DE_ObjectDimensionAccuracy

物標の一辺の長さの精度 (誤差がそれ以下である確率が 95%になる値) を 0.01m 単位で表すデータ型。

具体的な表現値は以下の通り。

1:	0.01m
.....	
65534:	655.34m
65535:	不明 (数値で表現する場合)

DE_StaticStatus

物標の静止状態を表すデータ型。

具体的な表現値は以下の通り。

0:	移動している (静止している時間が 1 秒未満)
1:	1~2 秒間静止している
2:	2~3 秒間静止している
.....	
3599:	3599~3600 秒間静止している
3600:	3600 秒間以上静止している
3601:	移動が観測されたことがない
3602:	不明 (数値で表現する場合)

DE_TrackingStatus

最新のセンシング状態 (物標を検知したかどうか), 検知しなかった場合にその理由, 物標情報の削除予告, 物標情報が融合または分割されたかを表すデータ型。それぞれの状態を表現する値のビット毎論理和で表す。

最新のセンシング状態の具体的な表現値は以下の通り。

0:	物標を検知した
0x01:	物標を検知しなかった

検知しなかった理由の具体的な表現値は以下の通り。

0:	物標を検知した, または, 検知しなかった理由が不明
----	----------------------------

0x02: センサーの検知範囲外

0x04: オクルージョン

物標情報の削除予告の具体的な表現値は以下の通り。

0: 削除予告なし, または不明

0x08: 削除予告あり(予告物標情報を削除する N 周期前からセットする)

物標情報が融合されたかの具体的な表現値は以下の通り。

0: 融合なし, または不明

0x10: 他の物標と融合された(物標の融合後, N 周期の間セットする)

物標情報が分割されたかの具体的な表現値は以下の通り。

0: 分割なし, または不明

0x20: 分割して生成された(物標の分割後, N 周期の間セットする)

N の具体的な値は, 本仕様では規定しない。

DE_DetectionCount

物標を検知した合計回数を表すデータ型。

具体的な表現値は以下の通り。

0: 不明(数値で表現する場合)

1: 1 回

.....

65534: 65534 回

65535: 65535 回以上

DE_LostCount

物標を連続して検知できなかった回数を表すデータ型。

具体的な表現値は以下の通り。

0: 物標を検知した, または不明(数値で表現する場合)

1: 1 回

.....

254: 254 回

255: 255 回以上

DE_ObjectAge

物標が最初に検知されてからの経過時間を表すデータ型。

具体的な表現値は以下の通り。

0: 0 秒(物標がはじめて検知された)

1: 0.1 秒

.....

35999: 3599.9 秒

36000: 3600.0 秒以上

36001: 不明(数値で表現する場合)

付録 B. Protocol Buffers のメッセージ定義

```
syntax = "proto3";

message SensingMessage {
  uint32 message_id = 1;
  uint32 protocol_version = 2;
  uint32 message_counter = 3;
  uint64 sensing_time = 4;
  uint32 error_notification = 5;
  uint32 error_code = 6;
  repeated SensorInformation sensor_info = 7;
  repeated ObjectInformation object_infos = 8;
  repeated PerceivedFreeSpaceInformation freespace_infos = 9;
}

message SensorInformation {
  optional SensorType type = 1;
  sint32 latitude = 2;
  sint32 longitude = 3;
  sint32 altitude = 4;
  repeated DetectCapability detect_capabilities = 5;
  uint32 sensor_status = 6;
}

enum SensorType {
  ST_UNKNOWN = 0;
  ST_RADAR = 1;
  ST_LIDAR = 2;
  ST_MONOVIDEO = 3;
  ST_STEREOVISION = 4;
  ST_NIGHTVISION = 5;
  ST_ULTRASONIC = 6;
  ST_PMD = 7;
  ST_FUSION = 8;
  ST_INDUCTIONLOOP = 9;
  ST_SPHERICALCAMERA = 10;
}

message DetectCapability {
  uint32 detectable_classes = 1;
  repeated OffsetPointXY poly_points = 2;
  optional uint32 confidence = 3;
}
```

```
    optional uint32 detectable_size = 4;
}

message OffsetPointXY {
    sint32 dx = 1;
    sint32 dy = 2;
}

message ObjectInformation {
    uint32 object_id = 1;
    optional sint32 time_of_measurement = 2;
    repeated ObjectClass object_classes = 3;
    optional uint32 confidence = 4;
    Position position = 5;
    optional RefPoint ref_point = 6;
    optional uint32 heading = 7;
    optional uint32 heading_accuracy = 8;
    optional sint32 speed = 9;
    optional uint32 speed_accuracy = 10;
    optional sint32 yaw_rate = 16;
    optional uint32 yaw_rate_accuracy = 17;
    optional sint32 acceleration = 18;
    optional uint32 acceleration_accuracy = 19;
    optional uint32 orientation = 20;
    optional uint32 orientation_accuracy = 21;
    optional uint32 length = 22;
    optional uint32 length_accuracy = 23;
    optional uint32 width = 24;
    optional uint32 width_accuracy = 25;
    optional uint32 height = 26;
    optional uint32 height_accuracy = 27;
    optional uint32 static_status = 11;
    optional uint32 tracking_status = 12;
    optional uint32 detection_count = 13;
    optional unit32 lost_count = 14;
    optional unit32 object_age = 15;
}

message ObjectClass {
    oneof subclass_type {
        VehicleSubclassType vehicle_subclass_type = 1;
        TrainSubclassType train_subclass_type = 2;
        MotorcycleSubclassType motorcycle_subclass_type = 3;
    }
}
```

```
LightVehicleSubclasType light_vehicle_subclass_type = 4;
PersonSubclasType person_subclass_type = 5;
AnimalSubclasType animal_subclass_type = 6;
NfoSubclasType nfo_subclass_type = 7;
FoSubclasType fo_subclass_type = 8;
}
optional uint32 class_confidence = 9;
optional uint32 subclass_confidence = 10;
}

enum VehicleSubclasType {
    VSCT_UNKNOWN = 0;
    VSCT_PASSENGER_CAR = 1;
    VSCT_BUS = 2;
    VSCT_LIGHT_TRUCK = 3;
    VSCT_HEAVY_TRUCK = 4;
    VSCT_TRAILER = 5;
    VSCT_SPECIAL_VEHICLES = 6;
    VSCT_EMERGENCY_VEHICLE = 7;
    VSCT_AGRICULTURAL = 8;
    VSCT_GROUP = 9;
}

enum TrainSubclasType {
    TSCT_UNKNOWN = 0;
    TSCT_TRAM = 1;
    TSCT_OTHER_TRAIN = 2;
}

enum MotorcycycSubclasType {
    MSCT_UNKNOWN = 0;
    MSCT_MOPED = 1;
    MSCT_MOTORCYCLE = 2;
    MSCT_GROUP = 3;
}

enum LightVehicleSubclasType {
    LVSCT_UNKNOWN = 0;
    LVSCT_BICYCLE = 1;
    LVSCT_RICKSHAW = 2;
    LVSCT_CART = 3;
    LVSCT_KICKBOARD = 4;
    LVSCT_GROUP = 5;
}
```



```
}

enum PersonSubclassType {
    PSCT_UNKNOWN = 0;
    PSCT_PEDESTRIAN = 1;
    PSCT_WHEELCHAIR = 2;
    PSCT_SENIOR_CAR = 3;
    PSCT_STROLLER = 4;
    PSCT_SKATES = 5;
    PSCT_GROUP = 6;
}

enum AnimalSubclassType {
    ASCT_UNKNOWN = 0;
}

enum NfoSubclassType {
    NFOSCT_UNKNOWN = 0;
}

enum FoSubclassType {
    FOSCT_UNKNOWN = 0;
}

message Position {
    sint32 latitude = 1;
    sint32 longitude = 2;
    sint32 altitude = 3;
    optional uint32 semi_axis_length_major = 4;
    optional uint32 semi_axis_length_minor = 5;
    optional uint32 semi_orientation = 6;
    optional uint32 altitude_accuracy = 7;
}

enum RefPoint {
    RP_UNKNOWN = 0;
    RP_CENTER_BOTTOM = 1;
    RP_FRONT_MIDWIDTH_BOTTOM = 2;
    RP_FRONT_RIGHT_BOTTOM = 3;
    RP_MIDLENGTH_RIGHT_BOTTOM = 4;
    RP_REAR_RIGHT_BOTTOM = 5;
    RP_REAR_MIDWIDTH_BOTTOM = 6;
    RP_REAR_LEFT_BOTTOM = 7;
}
```

```
RP_MIDLENGTH_LEFT_BOTTOM = 8;  
RP_FRONT_LEFT_BOTTOM = 9;  
}  
  
message PerceivedFreeSpaceInformation {  
  optional sint32 time_of_measurement = 1;  
  Position position = 2;  
  repeated OffsetPointXY poly_points = 3;  
  optional uint32 confidence = 4;  
  optional uint32 detectable_size = 5;  
}
```

付録 C. Collective Perception Message (CPM) との関係

ETSI(欧州電気通信標準化機構)による協調認識サービスに関するテクニカルレポート[4]には、ITS ステーション(車両や路側機)間で、検知した物標の情報を共有するための Collective Perception Message (CPM) が規定されている。CPM は、ITS ステーション間のメッセージであるため、本仕様で定義するセンサー部インタフェースのデータフォーマットとは位置付けが異なるが、目的が似ているために、共通する部分が多い。以下では、本仕様のデータフォーマットと CPM との主な相違点について述べる。

CPM には、管理情報、ステーションデータ情報、センサー情報、認識物標情報、 freespace 追加情報などが含まれる。管理情報とステーションデータ情報には、メッセージの発信元である車両や路側機に関する情報が含まれる。特に車両からのメッセージの場合には、車両の現在位置に加えて、移動状況(移動方向、速さ、回転速度、加速度など)が含まれる。本仕様のデータフォーマットには、これに対応する情報は含んでいない。

CPM のセンサー情報と本仕様のセンサー情報には、以下の相違がある。

- CPM のセンサー情報は、センサーID のデータ項目を持つ。本仕様では、センサーID は路側 DM 側で付与することを想定しており、センサー情報はセンサーID のデータ項目を持たない。
- CPM では、1 つのセンサー情報に対して、検知領域は 1 つしか持てない。
- CPM では、センサーの検知領域は、放射状の領域、円、楕円、長方形でも表現できる。多角形による検知領域の表現では、センサー設置位置の情報は持っていない。
- CPM には、検知可能物標種別、検知限界物標サイズ、センサー状態の各データ項目はない。

CPM の認識物標情報と本仕様の物標情報には、以下の相違がある。

- CPM の認識物標情報は、物標の向きは判別できないという前提でデータ項目が設計されている。そのため、物標の位置、速度、加速度は、 x , y , z の 3 軸方向の値で表現されている。これと比較すると、本仕様の物標情報では、鉛直方向の速さと、前後方向以外の加速度は表現できない。逆に CPM では、回転速度のデータ項目を持たない。また、物標参照位置の表現方向も異なる。
- CPM には、本仕様の物標情報に含まれないデータ項目として、物標を認識したセンサーのリスト、物標の存在するレーン ID と縦方向の位置(`matchedPosition`)がある。
- CPM には、トラッキング状態、検知回数、連続未検知回数の各データ項目はない。

CPM の freespace 追加情報は、センサーの検知領域と物標情報から算出できる freespace が実際の freespace と一致しない場合に、追加的に送信する情報である。それに対して、本仕様の直接検知 freespace 追加情報は、センサーが直接検知した freespace に関する情報を表すものであり目的が異なるが、両者のデータフォーマットには類似性が高い。CPM の freespace 追加情報情報と本仕様の直接検知 freespace 情報のデータフォーマットには、以下の相違がある。

- CPM の freespace 追加情報には、本仕様の直接検知 freespace 情報に含まれないデータ項目として、物標を認識したセンサーのリストと、 freespace 領域内でオクルージョンを考える必要があるかないかを示すフラグ(`shadowingApplies`)がある。
- CPM では、 freespace 領域は、円、楕円、長方形でも表現できる。

- CPM には、情報取得時刻と検知限界物標サイズの各データ項目はない。
- この他に、各データ項目の表現方法にも多くの違いがある。比較的重要なものを以下に示す。
- CPM では、メッセージ全体に対して管理情報の中に基準位置の情報を持っており、認識物標情報に含まれる物標の位置は、そこからの相対位置で表現されている。
 - CPM では、信頼度(検知信頼度、存在信頼度)を表す値の意味を定めていない。本仕様では、値の意味を決めているが、値を求める手法には言及していない。

付録 D. バージョン履歴

2023 年 12 月 26 日 version 1.0.0 最初のリリース版