

Spécification Technique de Réalisation pour le module de Communication

Version : 0.3

Date de version : 14/05/2025

|  |  |
| --- | --- |
| Une image contenant signe, extérieur, ciel, rue  Description générée automatiquement | Opération réalisée avec le concours des Investissements d’avenir de l’Etat confiés à l’ADEME |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Projet | Zone | Lot | Phase | Générations | Type de document | Emetteur | Numéro Chrono | Version | Indice |
|  | | | | | | | | | |

Informations du document

Périmètre de diffusion : interne

Type : xxx

Date prévue de livraison : 14/05/2024

Statut : xxx

Lieu de stockage du document : xxxx

Auteurs :

|  |  |  |
| --- | --- | --- |
| Pilote(s) | Organisation | Rôle dans le projet |
| R.POTARUSOV | 09/01/2025 | Création |
| B. YAHIAOUI | 14/05/2025 | Sprint2 |
|  |  |  |
| Contributeurs | Organisation | Rôle dans le projet |
|  |  |  |
|  |  |  |
|  |  |  |

Table de révision

|  |  |  |
| --- | --- | --- |
| Version | Date | Contenu de la modification |
| 0.1 |  |  |
| 0.2 |  |  |
| 0.3 | 14/05/2025 | Adaptation à la version des messages au format bytes |
| 0.4 |  |  |
| 0.5 |  |  |
| 0.6 |  |  |
|  |  |  |
|  |  |  |

Table des matières

[Informations du document 2](#_Toc198617709)

[Table de révision 2](#_Toc198617710)

[Table des matières 3](#_Toc198617711)

[1. Caractéristique logiciel 4](#_Toc198617712)

[2. Description du document 4](#_Toc198617713)

[3. Environnement Software et Hardware 5](#_Toc198617714)

[4. Exigences implémentées 5](#_Toc198617715)

[5. Dispositions particulières 5](#_Toc198617716)

[5.1. Liste des paramètres nécessaires au lancement 5](#_Toc198617717)

[5.2. Descriptif des entêtes de messages 7](#_Toc198617718)

[6. Schéma Global du Module AD 8](#_Toc198617719)

[7. Brique MsgDecoder 9](#_Toc198617720)

[7.1. Entrées du sous-module Décode : 9](#_Toc198617721)

[7.2. Sorties du sous-module Decode: 9](#_Toc198617722)

[7.3. Description des principaux algorithmes 10](#_Toc198617723)

[8. Brique MsgEncoder 11](#_Toc198617724)

[8.1. Entrées du sous-module Encode 11](#_Toc198617725)

[8.2. Sorties du sous-module Encode 12](#_Toc198617726)

[8.3. Description des principaux algorithmes 12](#_Toc198617727)

# Caractéristique logiciel

|  |  |
| --- | --- |
| Fonction | FP3.3.1 : Le module de communication (SW + HDW) doit assurer la transmission des messages de l’AD vers la supervision et vice versa |
| Nom du logiciel | Module de communication |
| Version | 1 |
| Dates des livrable | 15/12/2024 |
| Date de validation | 15/12/2024 |
| Utilisateur | SICEF |
| Equipe/service | AD |
| Responsables | B. YAHIAOUI |

# Description du document

Ce document est une description détaillée du module de communication, en particulier ses briques des messageries ADçèSupervision, qui sérialise et désérialise des données définies dans l’architecture présentée dans le document : "1.4.9.Architecture des messages mécanismes de sécurité et protocole d’échanges\_FM-TZ-L4-TTP-G2-LIV-SIC-00118-v0.4.1". Ce dernier document contient le détail de définition de chaque message avec le nombre d’octets qu’occupe chaque valeur et son "offset" dans la trame. Le livrable issu à la réalisation de ce module s’appuie sur les choix suivants :

* La réalisation du livrable associé a été validé avec un protocole de communication en TCP,
* La valeur de la taille maximale d’une trame a été définie à 1000 octets,
* Compatibilité en MIL, SIL et test sur véhicule,
* L’interprétation des messages se fait en BigEndian.

Le module de Communication comporte deux briques de messagerie : MsgDecoder (réception de messages) et MsgEncoder (envoi de messages). Le principe est que l’on reçoit un à l’entrée du décodeur un vecteur en uint8 ainsi que sa taille et que l’on fournit à la sortie de l’encodeur le même type de donné (vecteur et sa taille).

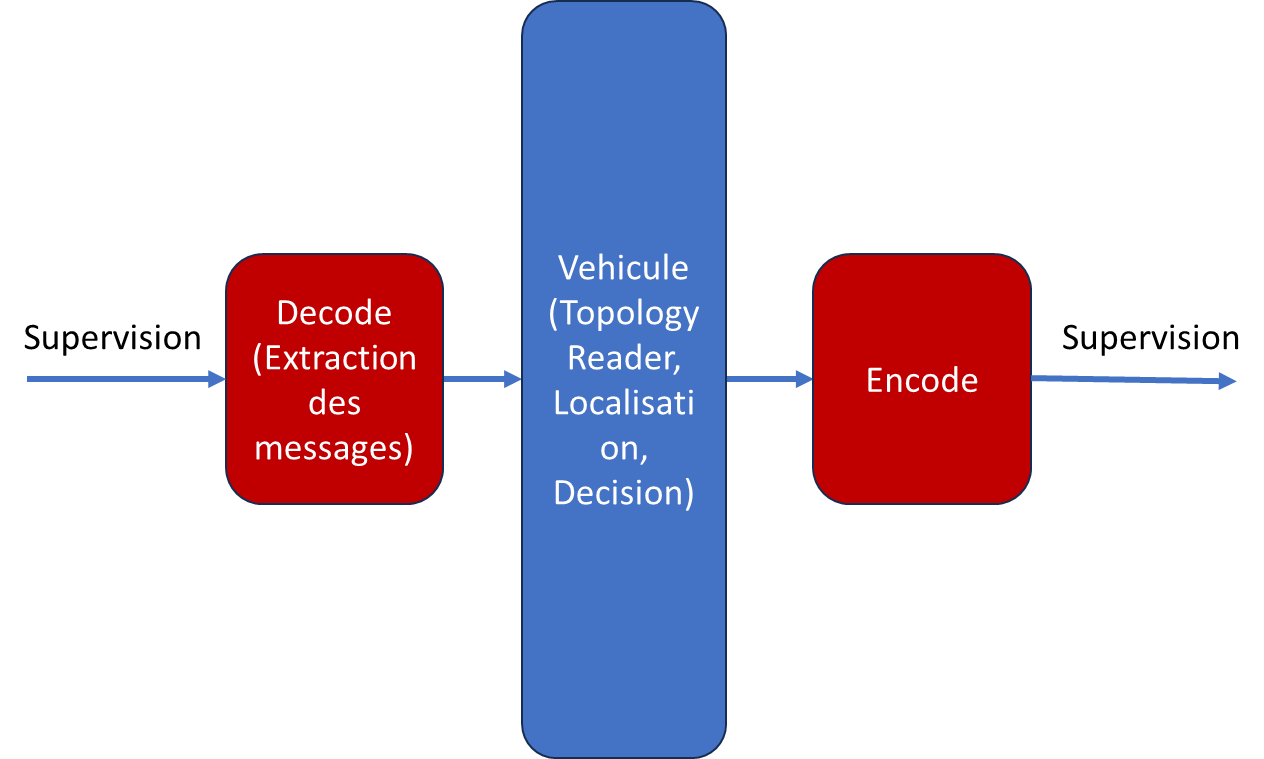


Figure 1. Schéma bloque du module de messagerie dans son contexte

Le Schéma ci-dessus comporte les briques de messagerie en rouge. Le bloque en bleu représente l’essentiel des briques AD.

# Environnement Software et Hardware

Ressources disponibles sur le microcontrôleur :

|  |  |
| --- | --- |
| **Feature** | **Detail** |
| Micro Control Core | 32-bit Infineon TC297TP |
| Maximum Frequency | 300 MHz |
| Flash | 8M |
| SRAM (Static Random Access Memory) | 728K |
| EEPROM (Electrically Erasable Programmable Read-Only Memory) | 128K |
| SBC (System Basic Chip) | TLF35584 |

Logiciels :

* Matlab R2024a
* La suite de logiciels Ecotron : EcoCoder, EcoCAL et EcoFlash

# Exigences implémentées

|  |  |  |
| --- | --- | --- |
| Exigences | Version | Commentaires |
| EX- FP1.4.9.1 | **1** |  |
| EX- FS1.4.9.1.1 | **1** |  |
| EX- FS1.4.9.1.2 | **1** |  |
| EX- FS1.4.9.1.3 | **1** |  |
| EX- FS1.4.9.1.4 | **1** |  |
| EX- FS1.4.9.1.5 | **1** |  |
| EX- FS1.4.9.1.6 | **1** |  |
| EX- FP1.4.9.2 | **1** | **A l’exception de message :**  **ISAcknowledge et ResumeMission** |
| FS1.4.9.2.1 | **1** |  |
| EX- FS1.4.9.2.2 | **1** |  |
| EX- FS1.4.9.2.3 | **1** |  |
| EX- FS1.4.9.2.4 | **1** |  |
| EX- FS1.4.9.2.5 | **1** |  |
| EX- FS1.4.9.2.6 | **1** |  |
| EX- FS1.4.9.2.7 | **1** |  |
| EX- FS1.4.9.2.8 | **1** |  |

# Dispositions particulières

## Liste des paramètres nécessaires au lancement

Que ce soit pour une simulation ou pour une compilation pour flashage sur ECU, il est nécessaire de définir les paramètres suivants :

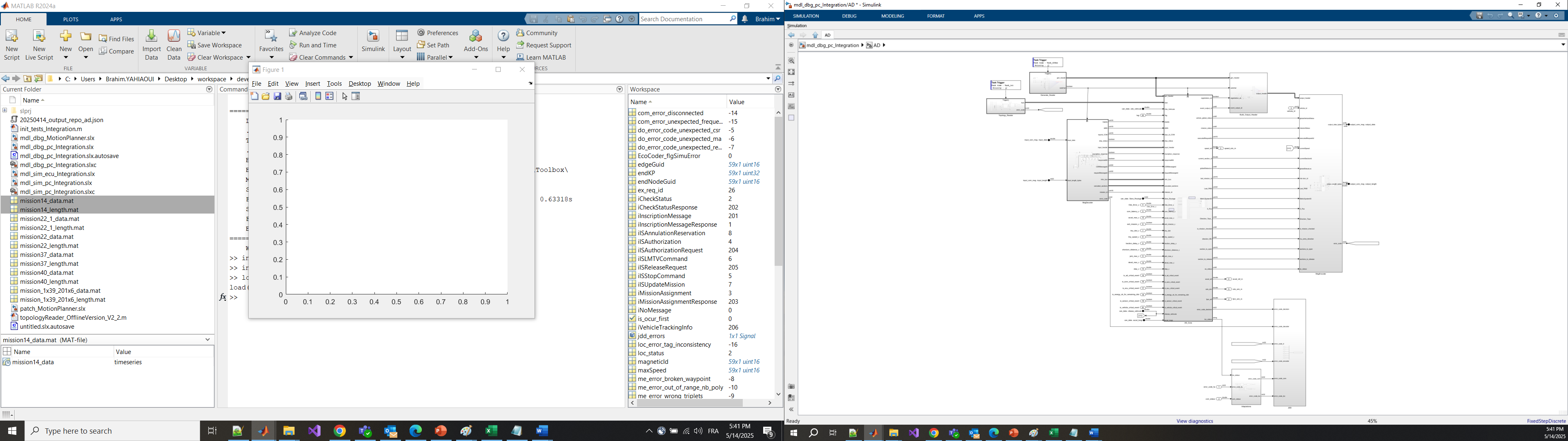
|  |  |  |
| --- | --- | --- |
| Nom | Valeurs | Description |
| **Constants** |  |  |
| null\_id | 65535 | L'identifiant concidéré comme l'élément nul |
| **Parameters** |  |  |
| msg\_version | 51 | Version du message |
| tcp\_msg\_size | 1330 | Taille maximale des trames tcp |
| nb\_max\_triplets | 32 | Nombre maximal de polylines |
| registration\_timeout | 1 | Délais de demande en boucle de l'inscription à la supervision |
| section\_entery\_timeout | 1 | Délais de demande en boucle d'autorisation pour occuper une section |
| nb\_max\_nodes\_mission | 100 | Nombre maximal de nœud dans une mission |
| nb\_max\_edges\_topo | 59 | Nombre maximal d'arc dans un fichier topo |
| nb\_poly\_sc | 16 | Nombre maximal de polynomes dans le MotionPlanner |
| nb\_corr\_sc | 1000 | Nombre maximal de corrections dans le cas d'une dégénéréscence d'une courbe dans le MotionPlanner |
| ex\_req\_id | 26 | Obsolète |
| loc\_status | 2 | Obsolète |
| **Precalculs** |  |  |
| nb\_max\_edges\_mission | 99 | nb\_max\_nodes\_mission - 1 |
| **Codes erreurs** |  |  |
| msg\_error\_invalid\_message | -1 | Impossible reçu est invalide ou peut être décalé. |
| msg\_error\_invalid\_message\_version | -2 | La version du message reçue est incompatible avec la version du decoder. |
| msg\_error\_unexpected\_request | -3 | Le message reçu est inattendu. |
| msg\_error\_unexpected\_response | -4 | La réponse à envoyer n'est pas connu par le module du encoder. |
| do\_error\_code\_unexpected\_csr | -5 | check status response dans la mission assignment est non compatible avec la dernière demande check status. |
| do\_error\_code\_unexpected\_ma | -6 | L'ouverture de section reçue est non compatible avec celle qui a été demandée. |
| do\_error\_code\_unexpected\_request | -7 | requête inattendue au moment où on attend une nouvelle mission |
| me\_error\_broken\_waypoint | -8 | Chemin reçu par la supervision non compatible avec la topo dans le calculateur. |
| me\_error\_wrong\_triplets | -9 | les données triplets reçus sont invalides, incontinuité des triplets détectés. |
| me\_error\_out\_of\_range\_nb\_poly | -10 | Le nombre de polynome maximale prédéfinit ne suffit pas pour contenir toute la mission |
| mp\_error\_failed\_gen | -11 | Itération de correction maximale atteinte. Résultat invalide. |
| mp\_error\_failed\_pcs | -12 | Impossible de valider ou de produire des abscisses curvilignes valides. Résultat invalide. |
| tr\_error\_wrong\_topo | -13 | Erreur dans la topologie au moment de la construction de la table des raccourcis. |
| com\_error\_disconnected | -14 | Connexions non établie ou perdue. |
| com\_error\_unexpected\_frequency | -15 | Deux messages reçus à une fréquence plus petite que 100ms. |
| loc\_error\_tag\_inconsistency | -16 | Deux messages reçus à une fréquence plus petite que 100ms. |
| **Identifiant des messages** |  |  |
| iNoMessage | uint8(0) | Valeur nul pour les identifiants de message |
| **>Supervision** |  |  |
| iInscriptionMessageResponse | uint8(1) | Valeur correspondant à la spec de communication |
| iCheckStatus | uint8(2) | Valeur correspondant à la spec de communication |
| iMissionAssignment | uint8(3) | Valeur correspondant à la spec de communication |
| iISAuthorization | uint8(4) | Valeur correspondant à la spec de communication |
| iISStopCommand | uint8(5) | Valeur correspondant à la spec de communication |
| iISLMTVCommand | uint8(6) | Valeur correspondant à la spec de communication |
| iISUpdateMission | uint8(7) | Valeur correspondant à la spec de communication |
| iISAnnulationReservation | uint8(8) | Valeur correspondant à la spec de communication |
| **>AD** |  |  |
| iInscriptionMessage | uint8(201) | Valeur correspondant à la spec de communication |
| iCheckStatusResponse | uint8(202) | Valeur correspondant à la spec de communication |
| iMissionAssignmentResponse | uint8(203) | Valeur correspondant à la spec de communication |
| iISAuthorizationRequest | uint8(204) | Valeur correspondant à la spec de communication |
| iISReleaseRequest | uint8(205) | Valeur correspondant à la spec de communication |
| iVehicleTrackingInfo | uint8(206) | Valeur correspondant à la spec de communication |

## Descriptif des entêtes de messages

Les messages, quel que soit le type, ont un ensemble de champs commun qui sont les suivants : **TypeMessage**, **messageId**, **timeStamp** et **sequenceNumber**. L’idée dans la réalisation est d’unifier ces champs sous le même bus de données. Ce qui permet une modularité efficace du code développé. L’objet constituant dans l’ordre les champs est appelé **« header »**. Pour cela, on a remonté le besoin d’unifier les tailles des messagesId, on nous propose 24bits.

Il est à noter qu’un bloc spécifique permet aujourd’hui de générer l’entête de sortie en calculant champ par champ.

# Schéma Global du Module AD

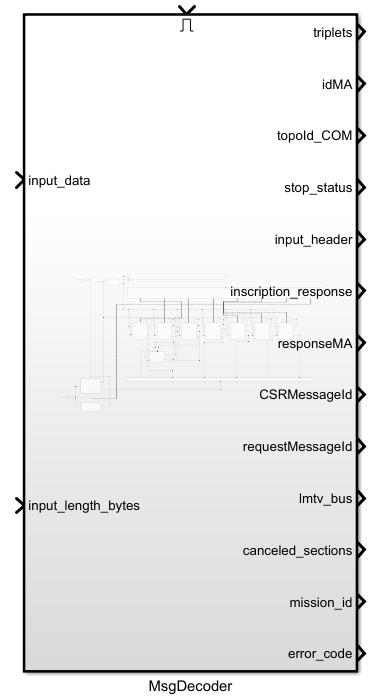


MsgDecoder

MsgEncoder

Figure 1. Modèle Simulink : vue globale

# Brique MsgDecoder



## Entrées du sous-module Décode :

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Description | Type | Provenance |
| input\_msg | Vecteur de bytes. | uint8 | Supervision |
| input\_length\_bytes | Taille du vecteur | int32 |  |

## Sorties du sous-module Decode:

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Description | Type | Destination |
| triplets | Défini un chemin décomposé en un ensemble de « sous-chemins » appelés triplets. Chaque triplet correspond à trois identifiants de nœud : le premier, le second et le dernier. Le triplet a pour particularité de ne contenir de nœud de bifurcation qu’a ses bords. | Tableau de 3x32 en uint16. | Décision |
| idMA | Identifiant de la section pour laquelle le véhicule demande l’autorisation | uint16 | Décision |
| topoId\_COM | Identifiant du référentiel topologique utilisé par la Supervision | uint16 | Décision |
| stop\_status | 1 si l’arrêt, 0 sinon. | boolean | Décision |
| input\_header | Entête du message d’entrée. | Bus (voir header [Descriptif des entêtes de messages](#_Descriptif_des_entêtes)) | Décision |
| inscription\_response | Vaut 1 si l’inscription vient d’être accordée par la supervision. | Boolean | Décision |
| responseMA | Vaut 1 si l’autorisation d’ouverture de section vient d’être acceptée. | boolean | Décision |
| CSRMessageId | L’identifiant de message envoyé à la supervision au moment d’un CheckStatusResponse et retourné par le MissionAssignment pour assurer la traçabilité de la demande de mission. | uint16 | Décision |
| requestMessageId | L’identifiant de message envoyé à la supervision au moment d’une demande d’autorisation d’ouverture de section et retourné par ISAuthorization pour assurer la traçabilité de la demande de la demande. | Uint16 | Décision |
| lmtv\_bus | Données pour limiter la vitesse sur un intervalle définies par la supervision. Cette sortie est composée deux trois données : la vitesse de limitation, l’identifiant du nœud de départ de la limitation et celui du nœud de la fin. | Bus{  uint16,  uint16,  uint16  } | Décision |
| canceled\_sections | Non utilisée. Incohérence remontée. | Bus{  uint16,  uint16,  uint16  } | Décision |
| mission\_id | Identifiant de mission. | uint16 | Décision |
| error\_code | Codes erreurs :   * Message invalide * Version de message invalide * Type de message inattendu | Vecteur de int32 | Brique de journalisations des codes erreurs |

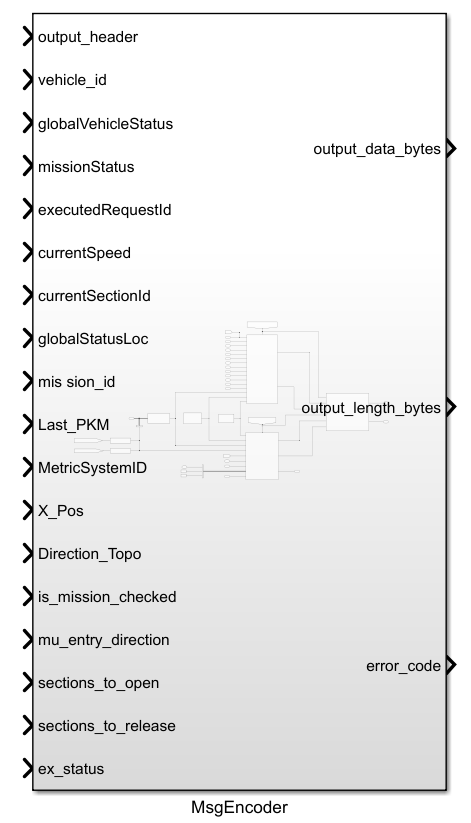
## Description des principaux algorithmes

La brique a pour but de lire le contenu en byte reçu et le traduire en des variables utiles pour les briques nécessitant des informations provenant de la supervision. Pour se faire, la brique commence par extraire l’entête de type header (voir [Descriptif des entêtes de messages](#_Descriptif_des_entêtes)). Le premier octet correspond au type de message et ainsi on applique un switch case qui permet de définir quel type de message on doit décoder. Si ce dernier est un message nul, il est ignoré, sinon si c’est un type qui n’est pas pris en compte, un code erreur est remonté. La brique a la capacité de vérifié si la version du message est valide où pas et informe la brique de journalisation des briques d’erreurs.

Pour décoder les messages, on utilise, selon le cas adapté, les opérations bit à bit ou des blocs permettant de convertir des tableaux de uint8 en entier.

Ce bloque a pour rôle aussi de fournir les données à une fréquence de 10Hz (100ms) aux restes des blocs qui en dépendent. Les entrées ont pour fréquence 20Hz (50ms) où on s’assure en amont que deux trames ne sont pas reçues en rafale ou entrelacées.

# Brique MsgEncoder



## Entrées du sous-module Encode

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Description | Type | Provenance |
| output\_header | Entête de message de sortie. | Bus (voir header [Descriptif des entêtes de messages](#_Descriptif_des_entêtes)) | Générateur de header (voir [Descriptif des entêtes de messages](#_Descriptif_des_entêtes)) |
| vehicle\_id | Identifiant du véhicule. | Uint16 | Calibration |
| globalVehicleStatus | Vaut 0 si aucun état critique n’a été retourné.  Module dépends à ce jour essentiellement de valeurs non calculées comme l’état de la batterie ou celle des pneus. | boolean | Décision |
| missionStatus | Vaut 1 si une mission est en cours. | boolean | Décision |
| executedRequestId | Manque de description dans la spec remontée. | uint16 | Décision |
| currentSpeed | Vitesse de véhicule courante. | uint16 | CAN |
| currentSectionId | Identifiant de l’identifiant de section en cours. Vaut 0 sin pas de section. | uint16 | Décision |
| globalStatusLoc | Manque de clarté sur la souce de cette donnée. Problème remonté. | uint8 | Décision |
| mission\_id | Identifiant de la dernière mission exécutée ou celle qui est en cours d’exécution. | uint16 | Décision |
| Last\_PKM | PRM du dernier TAG rencontré. Vaut 0 si ce n’est pas encore le cas. | uint32 | Localisation |
| MetricSystemID | Identifiant du système métrique en cours. | uint16 | Localisation |
| X\_Pos | Coordonnée du véhicule en abscisse curviligne relativement à la métrique en cours dans le repère topologique. | uint32 | Localisation |
| Direction\_Topo | Direction topologique du véhicule relativement à la métrique en cours dans le référentielle topologique. | uint8 | Localisation |
| is\_mission\_checked | Vaut 1 si la mission envoyée pas la supervision est validée. | boolean | Décision |
| mu\_entry\_direction | Direction topologique de la prochaine section fermée. | uint8 | Décision |
| sections\_to\_open | Identifiant de la section à demander pour lancer un ISAuthorizationRequest. | uint16 | Décision |
| sections\_to\_release | Identifiant de la section à libérer pour lancer un ISReleaseRequest. | uint16 | Décision |
| ex\_status | Manque d’explication pour cette variable. Demande remontée. | uint8 | Décision |

## Sorties du sous-module Encode

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Description | Type | Destination |
| output\_msg | Vecteur de bytes. | uint8 | Supervision |
| output\_length\_bytes | Taille du vecteur. | int32 | Supervision |
| error\_code | Code d’erreur qui se manifeste si un évènement envoyé par la décision est incompatible avec la messagerie. | int32 | Brique de journalisations des codes erreurs |

## Description des principaux algorithmes

Le rôle de ce bloque est de produire la totalité de la messagerie à envoyer par l’AD à la supervision. On distingue deux types de messages : les messages périodiques VehiculeTrakingInfo et les messages évènementiels qui correspondent au reste des messages envoyés à la supervision (CheckStatusResponse, MissionAssignment, ISAuthorizationRequest…ets.). Ce bloque tourne à la fréquence de 20Hz (50ms) : Une fois sur deux, il envoie un message périodique de type VehiculeTrackingInfo, sinon il envoie un message de type évènementiel si celui-ci est levé par la brique décision. Grace à ce mécanisme, on parvient à envoyer dans le même port les deux types de messages.

Pour encoder les messages, on utilise comme pour le décodeur, selon le cas adapté, les opérations bit à bit ou des blocs permettant de convertir des entiers en tableaux de uint8.

Le seul code erreur géré par le bloque est celui où un évènement inattendu est envoyé par la décision. Cela pour prévenir éventuellement à des problèmes d’incompatibilité de versions.