Principe de gestion des configurations flotteur dans le décodeur Matlab

Cette note présente les mécanismes de gestion (collecte, suivi et stockage) des paramètres de configuration d’un flotteur dans le décodeur Matlab.

La gestion présentée ici concerne les flotteurs PROVOR à transmission Irdium Rudics (Remoceans) et Iridium SBD (FLBB indiens, Arvor deep, Arvor Iridium et PROVOR-DO Iridium).

[Principe de gestion des configurations flotteur dans le décodeur Matlab 1](#_Toc410639896)

[1. Problématique 1](#_Toc410639897)

[2. Structure de stockage des informations de configuration 2](#_Toc410639898)

[3. Déroulement des différentes étapes 3](#_Toc410639899)

[3.1. Etape initiale 3](#_Toc410639900)

[3.2. Traitement des données reçues 3](#_Toc410639901)

[3.2.1. Réception d’un message de configuration 3](#_Toc410639902)

[3.2.2. Traitement du buffer de décodage 4](#_Toc410639903)

[3.3. Etapes finales 4](#_Toc410639904)

[3.3.1. Etape 1 : génération de la configuration finale 5](#_Toc410639905)

[3.3.2. Etape 2 : affectation de la configuration finale aux données 5](#_Toc410639906)

[3.3.3. Etape 3 : génération du fichier META 5](#_Toc410639907)

# Problématique

Les conditions de réalisation de la mission d’un flotteur PROVOR sont définies par l’ensemble de ses paramètres de configuration.

Pour tous les types de flotteurs, ces paramètres de programmation sont définis avant la mise à l’eau ; pour les flotteurs à transmission Iridium, certains paramètres peuvent être modifiés ultérieurement.

Ainsi, dans un flotteur Argos, les paramètres de configuration sont toujours **statiques** (c'est-à-dire qu’ils ne sont pas modifiables une fois le flotteur lâché) alors que dans un flotteur Iridium, les paramètres de configuration peuvent être **statiques** ou **dynamiques** (c'est-à-dire qu’il existe des commandes permettant de les modifier durant la mission du flotteur et des messages permettant à ce dernier d’accuser réception de sa nouvelle programmation).

La configuration du flotteur étant indispensable au décodage de ses données et à la détermination des conditions de réalisation des mesures obtenues, le décodeur doit assurer le **suivi temps réel** de ces paramètres.

# Structure de stockage des informations de configuration

Toutes les informations nécessaires à la gestion des configurations sont stockées dans la structure globale g\_decPrv\_floatConfig.

**La configuration statique** est renseignée une seule fois et stockée dans :

* + g\_decPrv\_floatConfig.STATIC.NAMES : nom des paramètres,
  + g\_decPrv\_floatConfig.STATIC.VALUES : valeur des paramètres.

Les messages dans lesquels le flotteur informe des changements dans sa configuration (messages de type 251, 254 et 255) sont décodés et constituent la **configuration en attente**. Elle est stockée dans :

* + g\_decPrv\_floatConfig.DYNAMIC\_TMP.DATES : date de la configuration,
  + g\_decPrv\_floatConfig.DYNAMIC\_TMP.NAMES : nom des paramètres,
  + g\_decPrv\_floatConfig.DYNAMIC\_TMP.VALUES : valeur des paramètres.

**Les configurations dynamiques** (créées par mise à jour de la configuration courante avec la configuration en attente) sont stockées dans :

* + g\_decPrv\_floatConfig.DYNAMIC.NUMBER : numéros des configurations,
  + g\_decPrv\_floatConfig.DYNAMIC.NAMES : nom des paramètres,
  + g\_decPrv\_floatConfig.DYNAMIC.VALUES : valeurs des paramètres,
  + g\_decPrv\_floatConfig.DYNAMIC.IGNORED\_ID : liste des paramètres à ignorer lors de la recherche d’une configuration existante.

Les informations permettant de lier les données aux configurations sont stockées dans :

* + g\_decPrv\_floatConfig.USE.CYCLE : numéro de cycle flotteur,
  + g\_decPrv\_floatConfig.USE.PROFILE : numéro de profil flotteur,
  + g\_decPrv\_floatConfig.USE.CYCLE\_OUT : numéro de cycle Argo,
  + g\_decPrv\_floatConfig.USE.CONFIG : numéro de configuration associé (au numéro de cycle flotteur et de profil flotteur concerné).

**Remarque :** La fonction *print\_config\_in\_csv\_file\_ir\_rudics\_sbd2*, disséminée (en commentaire) aux endroits cruciaux du code, permet de suivre le contenu des différents tableaux liés aux configurations et stockés via la variable globale g\_decPrv\_floatConfig.

# Déroulement des différentes étapes

## Etape initiale

Cette étape a lieu lors de l’initialisation d’une session de décodage de toutes les données reçues d’un flotteur.

Elle se fait dans la fonction *init\_float\_config\_ir\_rudics*.

Les données de configuration du flotteur au lâcher, stockées dans son fichier JSON de méta-données sont utilisées pour :

1. Construire la configuration statique qui ne sera jamais mise à jour car ces paramètres ne sont pas modifiables par une commande (initialisation de g\_decPrv\_floatConfig.STATIC.NAMES et g\_decPrv\_floatConfig.STATIC.VALUES),
2. Construire la liste des paramètres à ne pas prendre en compte lorsque l'on cherchera si une configuration donnée existe déjà ou non (renseignement de la liste g\_decPrv\_floatConfig.DYNAMIC.IGNORED\_ID). Dans le cas du Remocean, il s'agit des paramètres PM3 à PM52 ; en effet les 5 paramètres associés à chacun des 10 profils possibles sont recopiés dans PM03 à PM07 en fonction du profil concerné.
3. Construire la première configuration avec les paramètres dynamiques du flotteur. **Cette configuration porte le numéro 0**, c'est la configuration au lâcher et elle ne sera jamais référencée en tant que configuration autre, c'est-à-dire qu’elle sera utilisée uniquement pour renseigner les informations LAUNCH\_CONFIG\_PARAMETER\_\* du fichier META (initialisation de g\_decPrv\_floatConfig.DYNAMIC.NUMBER, g\_decPrv\_floatConfig.DYNAMIC.NAMES et g\_decPrv\_floatConfig.DYNAMIC.VALUES).
4. Construire la première configuration en attente avec les paramètres dynamiques du flotteur, elle est datée de la date de mise à l’eau du flotteur (initialisation de g\_decPrv\_floatConfig.DYNAMIC\_TMP.DATES, g\_decPrv\_floatConfig.DYNAMIC\_TMP.NAMES et g\_decPrv\_floatConfig.DYNAMIC\_TMP.VALUES).

## Traitement des données reçues

### Réception d’un message de configuration

Lorsque l’on reçoit un message de changement de configuration (message de type 251, 254 ou 255). Ce message est décodé et les informations contenues sont prises en compte.

Cela se fait dans la fonction *update\_float\_config\_ir\_rudics\_sbd2*.

Les informations décodées sont utilisées pour créer une nouvelle configuration en attente, datée de la date du message reçu. Cette nouvelle configuration est initialisée avec la dernière configuration en attente puis mise à jour avec les informations décodées.

### Traitement du buffer de décodage

Lorsque le contenu du buffer est décodé (soit parce qu’il est complet, soit parce qu’il est trop vieux et doit être décodé) et que le décodage indique qu’au moins un paquet de données (type 0) a été reçu (même s’il est vide), il est nécessaire de figer la configuration courante et de l’associer aux données.

**Attention :** le décodage du contenu du buffer ne concerne que les messages de type 0, 250, 252, et 253 **mais surtout pas** les messages de type 251, 254 ou 255 qui concernent la configuration qu’il faudra prendre en compte pour traiter les cycles à venir.

La création d’une nouvelle configuration et son association aux données décodées sont faites dans la fonction *set\_float\_config\_ir\_rudics*, au cours des étapes suivantes :

1. Recherche de la configuration courante,
2. Mise à jour de cette configuration avec les informations de la configuration en attente. Cette phase comporte une duplication des paramètres PM associés au profil courant dans les paramètres PM03 à PM07.

**Attention :** dans les premières versions logicielles du flotteur, les messages #254 et #255 sont à prendre en compte à l'issue de chaque profil flotteur pendant un cycle flotteur ; dans les nouvelles versions logicielles du flotteur, les messages #254 et #255 sont à prendre en compte uniquement à l'issue d'un cycle flotteur.

1. Recherche de la configuration ainsi créée parmi les configurations existantes (en tenant compte de la liste g\_decPrv\_floatConfig.DYNAMIC.IGNORED\_ID) :
   1. Si elle n'existe pas, on crée une nouvelle configuration,
   2. Si elle existe mais porte le numéro 0, c'est la configuration au lâcher qui est dupliquée pour ne pas être référencée dans les données PROF ou TRAJ.
2. Affectation de la configuration trouvée (ou nouvellement créée) aux données du cycle flotteur et profil flotteur que l'on vient de décoder.

## Etapes finales

Lorsque le traitement de toutes les données reçues du flotteur est achevé, on génère, affecte puis sauvegarde la configuration finale qui figurera dans le fichier META et sera référencée dans les fichiers PROF et TRAJ.

Cette opération est réalisée en trois étapes.

### Etape 1 : génération de la configuration finale

La génération de la configuration finale se fait dans la fonction *create\_output\_float\_config\_ir\_rudics* au cours des étapes suivantes :

1. Création des numéros de cycle Argo, pour cela :
   1. On construit la liste des cycles flotteur et profils flotteur probablement transmis (mais non nécessairement reçus), i.e. en comblant les éventuels trous de numéros de cycles flotteur ou de profils flotteur,
   2. On numérote de 1 à N le tableau ainsi obtenu.
2. Création de la configuration finale qui figurera dans le fichier META, pour cela :
   1. On renomme les PM03 à PM07 en PM3 à PM7 afin qu’ils soient correctement identifiés lors de l’affectation du label NetCDF associé à chaque paramètre,
   2. On supprime les paramètres qui ne concernent pas cette version de flotteur,
   3. On supprime les paramètres que l'on ne souhaite pas diffuser (informations Iridium confidentielles),
   4. On affecte à chaque paramètre son label NetCDF Argo,
   5. On stocke le résultat dans la configuration finale.

### Etape 2 : affectation de la configuration finale aux données

L’affectation de la configuration finale aux données se fait dans la fonction *add\_configuration\_number\_ir\_rudics\_sbd2*.

Pour cela le numéro de cycle Argo est ajouté à la structure des données décodées (profil ou trajectoire).

### Etape 3 : génération du fichier META

La génération du fichier NetCDF de méta-données est réalisée dans la fonction *create\_nc\_meta\_file\_3\_1*, au cours des opérations suivantes :

1. Récupération de la liste des paramètres que l'on considère obligatoires.

Pour chaque version de flotteur, on a préalablement établi une liste de paramètres de configuration qui seront visibles dans les tableaux CONFIG\_PARAMETER\_\* et CONFIG\_MISSION\_\* même si leur valeur n'est jamais modifiée par rapport à celle des tableaux LAUNCH\_CONFIG\_PARAMETER\_\*.

1. Construction du tableau final des configurations en concaténant les configurations statique et dynamique,
2. Construction de la configuration au lâcher, qui sera stockée dans les tableaux LAUNCH\_CONFIG\_PARAMETER\_\*, en prenant en compte la configuration statique et la première configuration dynamique (affectée du numéro 0),
3. Construction des configurations qui seront stockées dans les tableaux CONFIG\_PARAMETER\_\* et CONFIG\_MISSION\_\*.

Lorsqu’un paramètre (non obligatoire) n’a pas été modifié depuis sa valeur au lâcher du flotteur, il est supprimé de la liste constituant la configuration finale.