# Mise en œuvre du décodeur Matlab pour les flotteurs Remocean

Cette note concerne le décodeur Matlab des flotteurs NKE PROVOR ProvBio II Remocean.

Numéro de version Coriolis : V5.9.

Numéro de décodeur Matlab : 105.

## Table des matières

[1. Mise en œuvre du décodeur Matlab pour les flotteurs Remocean 1](#_Toc367782216)

[1.1. Table des matières 1](#_Toc367782217)

[1.2. Description de la livraison 1](#_Toc367782218)

[1.2.1. Répertoire « matlab » : 1](#_Toc367782219)

[1.2.2. Répertoire « json » : 2](#_Toc367782220)

[1.3. Installation/configuration du décodeur 2](#_Toc367782221)

[1.4. Mise en œuvre du décodeur 3](#_Toc367782222)

[1.4.1. Principe de gestion des fichiers SBD 3](#_Toc367782223)

[1.4.2. Programme decode\_provor\_2\_csv 5](#_Toc367782224)

[1.4.3. Programme decode\_provor\_2\_nc 6](#_Toc367782225)

[1.4.4. Programme decode\_provor\_2\_nc\_rt 8](#_Toc367782226)

[1.5. Décodage temps réel des flotteurs Remocéan par le DAC Coriolis 10](#_Toc367782227)

[1.5.1. Déclaration des flotteurs 10](#_Toc367782228)

[1.5.2. Lancement de la première session du décodeur 10](#_Toc367782229)

[1.5.3. Intégration d’un nouveau flotteur 10](#_Toc367782230)

## Description de la livraison

Le décodeur est livré dans un répertoire « decPrv\_*yyyymmdd* » qui contient :

* Un répertoire « matlab » contenant le code matlab de l’application et les fichiers de configuration associés,
* Un répertoire « json » contenant les fichiers json associés au décodeur,
* Un répertoire « prvFloatInfo » contenant le fichier « \_provor\_floats\_information\_co.txt » regroupant les paramètres de décodage des flotteurs déclarés,
* Un répertoire « lists » contenant des listes de numéros WMO de flotteurs,

### Répertoire « matlab » :

Il contient :

* Les 3 programmes permettant de mettre en œuvre le décodeur :
  + **decode\_provor\_2\_csv**,
  + **decode\_provor\_2\_nc**,
  + **decode\_provor\_2\_nc\_rt**,
* Les 2 fichiers de configuration :
  + « \_prv\_decoder.conf » (pour les programmes **decode\_provor\_2\_csv** et **decode\_provor\_2\_nc),**
  + « \_prv\_decoder\_conf.json » (pour le programme **decode\_provor\_2\_nc\_rt**),
* Le répertoire « sub » contenant les sous-programmes développés pour le décodeur,
* Le répertoire « sub\_foreign » contenant les sous-programmes utiles au décodeur et issus d’internet ou du LPO,
* Le répertoire « util » contenant quelques utilitaires externes au décodeur et le sous-répertoire « sub » contenant les sous-programmes associés.

### Répertoire « json » :

Il contient :

* Le répertoire « configParamNames » : contenant le fichier de description des paramètres de configuration des flotteurs V5.9 (**\_config\_param\_name\_105.json**) utilisés dans le fichier meta.nc,
* Le répertoire « techParamNames » : contenant le fichier de description des paramètres techniques des flotteurs V5.9 (**\_tech\_param\_name\_105.json**) utilisés dans le fichier tech.nc,
* Le répertoire « json\_float\_info » : contenant un fichier de paramètres de décodage pour chaque flotteur déclaré (ex : **6901528\_lovbio018c\_info.json**),
* Le répertoire « json\_float\_meta\_remocean » : contenant un fichier de méta-données pour chaque flotteur déclaré (ex : **6901528\_meta.json**).

Rmq : Actuellement, le contenu de ce fichier est généré à partir des données de la base Coriolis à l’exception du contenu des rubriques :

* + CONFIG\_PARAMETER\_NAME,
  + CONFIG\_PARAMETER\_VALUE,
  + CONFIG\_MISSION\_NUMBER,
  + SENSOR\_MOUNTED\_ON\_FLOAT,
  + CALIBRATION\_COEFFICIENT.

## Installation/configuration du décodeur

Une fois le contenu du répertoire « decPrv\_*yyyymmdd* » copié sur la machine hôte :

* L’installation se fait en ajoutant au path matlab le répertoire « matlab » et tout ses sous-répertoires,
* La configuration du décodeur se fait en renseignant les paths des fichiers de configuration « matlab/**\_prv\_decoder.conf** » et « matlab/**\_prv\_decoder\_conf.json** » conformément au répertoire d’installation sur la machine hôte.

## Mise en œuvre du décodeur

Le décodeur peut être mis en œuvre de trois manières différentes, via les programmes :

* **decode\_provor\_2\_csv** : pour un traitement **temps différé** des données vers un fichier **CSV**,
* **decode\_provor\_2\_nc** : pour un traitement **temps différé** des données vers des fichiers **NetCDF**,
* **decode\_provor\_2\_nc\_rt** : pour un traitement **temps réel** des données vers des fichiers **NetCDF**.

Un traitement **temps différé** correspond à l’exploitation intégrale de l’ensemble des fichiers SBD dont on dispose au moment du décodage.

Un traitement **temps réel** correspond à l’exploitation « au fil de l’eau » d’un flux de données. Dans ce cas l’intégralité des données peut ne pas être exploité (le décodeur étant en attente de données complémentaires).

**Attention :** ces deux modes de fonctionnement étant **incompatibles** il est prudent de ne pas mélanger les données d’entrée et de sortie associées à ces deux fonctionnements.

En particulier, les répertoires définis par les variables de configuration IRIDIUM\_DATA\_DIRECTORY et DIR\_OUTPUT\_NETCDF\_FILE ne doivent pas être les mêmes dans les fichiers « matlab/\_prv\_decoder.conf » (fichier de configuration de **decode\_provor\_2\_csv** et **decode\_provor\_2\_nc**) et « matlab/\_prv\_decoder\_conf.json » (fichier de configuration de **decode\_provor\_2\_nc\_rt**).

### Principe de gestion des fichiers SBD

La gestion des fichiers SBD s’effectue dans le répertoire défini par le paramètre de configuration IRIDIUM\_DATA\_DIRECTORY. Ce répertoire va contenir un sous-répertoire dédié à chaque flotteur traité (par exemple **IRIDIUM\_DATA\_DIRECTORY/lovbio018c/** pour le flotteur 6901528 dont le « login name » est lovbio018c).

Ce répertoire **IRIDIUM\_DATA\_DIRECTORY/lovbio018c/** va contenir les sous-répertoires :

* « archive » : contenant les fichiers traités,
* « spool » : contenant les fichiers à traiter lors de la session du décodeur,
* « buffer » : contenant les fichiers en cours de traitement,
* « tmp » : contenant les informations temporaires stockées entre deux sessions du programme **decode\_provor\_2\_nc\_rt**.

Au cours d’une session, les fichiers SBD sont traités dans l’ordre chronologique défini par la date codée dans leur nom (ex : ***yymmdd*\_*HHMMSS*\_lovbio018c\_00019.b64.sbd**).

L’enchainement des étapes est le suivant :

1. Déplacement du plus vieux fichier SBD du répertoire « spool » vers le répertoire « buffer »,
2. S’il existe, dans le répertoire « buffer » des fichiers datés plus de **2 heures** avant le nouvel entrant, on **force** leur traitement. Une fois exploités, ils sont déplacés vers le répertoire « archive »,
3. Le contenu des fichiers SBD présents dans le répertoire « buffer » est alors **pré-décodé** afin d’analyser le contenu global du buffer :
   1. Si le contenu du buffer est **incomplet**, le traitement reprend à l’étape 1,
   2. Si le contenu du buffer est **complet**, il est traité et les fichiers associés sont déplacés vers le répertoire « archive ». Le traitement reprend alors à l’étape 1.

Le processus s’arrête lorsque le répertoire « spool » est vide.

Un buffer est **complet** si l’on a reçu pour chaque sous-cycle :

* Le message technique du flotteur,
* Le message technique associé à chaque capteur,
* Pour chaque capteur, le nombre de messages de données mentionné dans le message technique associé à ce capteur.

**Pourquoi un buffer de 2 heures ?**

Pour assurer le bon fonctionnement du décodeur, il est nécessaire que le buffer puisse contenir **toutes** les données transmises à l’issue d’**une seule** phase en sub-surface (pouvant être constituée d’un ou de plusieurs sous-cycles). L’analyse des données transmises à ce jour par les Remocéans nous a conduits à dimensionner ce buffer à **2 heures**. La difficulté est que, les paramètres de configuration étant multiples, la durée de la phase de sub-surface n’est pas proportionnelle à la quantité de données à transmettre, donc à la durée de la transmission. Il n’est donc pas exclu que la durée du buffer doivent être adaptée à chaque flotteur via un paramètre de configuration de décodage (ce qui sera possible car, par construction, le buffer est spécifique à chaque flotteur).

A l’issue d’une session du décodeur :

* Le répertoire « spool » est vide,
* Le répertoire « buffer » est :
  + Vide avec **decode\_provor\_2\_csv** et **decode\_provor\_2\_nc** (puisque, à la dernière étape de ces programmes, on **force** le traitement du buffer),
  + Non nécessairement vide avec **decode\_provor\_2\_nc\_rt** (si l’on est en attente de l’arrivée d’un nouveau flux de données pour compléter le buffer).
* Le répertoire « tmp » est :
  + Vide avec **decode\_provor\_2\_csv** et **decode\_provor\_2\_nc** (car non utilisé dans ce cas),
  + Non vide avec **decode\_provor\_2\_nc\_rt** (cf. 1.4.4 où son contenu est explicité).

### Programme decode\_provor\_2\_csv

Les fichiers associés à ce programme sont :

* Entrées :
  + Fichier de configuration du décodeur : « matlab/\_prv\_decoder.conf »,
  + Fichier des paramètres de décodage des flotteurs déclarés : « prvFloatInfo/\_provor\_floats\_information\_co.txt »,
  + Fichiers de méta-données des flotteurs déclarés : ceux du répertoire « json/json\_float\_meta\_remocean » (ex : **6901528\_meta.json**).

Seules les rubriques :

* + - CONFIG\_PARAMETER\_NAME,
    - CONFIG\_PARAMETER\_VALUE,
    - CONFIG\_MISSION\_NUMBER,
    - SENSOR\_MOUNTED\_ON\_FLOAT,
    - CALIBRATION\_COEFFICIENT

sont ici exploitées,

* + Fichiers SBD d’entrée : ceux stockés dans le répertoire « archive » d’archivage des fichiers SBD de chaque flotteur (ex : **IRIDIUM\_DATA\_DIRECTORY/lovbio018c/archive/**).
* Sorties :
  + Fichier log créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_LOG\_FILE,
  + Fichier CSV créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_CSV\_FILE.

Paramètres d’entrée du programme :

* En l’absence d’argument, le décodage s’applique à tous les flotteurs de la liste définie par le paramètre de configuration FLOAT\_LIST\_FILE\_NAME,
* Il est cependant possible de préciser les flotteurs à décoder en fournissant en argument leur numéro WMO (ex : decode\_provor\_2\_csv(6901528) ou decode\_provor\_2\_csv(6901472, 6901482, 6901529)).

La première étape du décodage consiste à déplacer les fichiers SBD du répertoire « archive » vers le répertoire « spool ». Seuls les fichiers SBD dont le numéro de cycle répond aux critères mentionnés par le paramètre de configuration EXPECTED\_CYCLE\_LIST sont déplacés.

**Attention :** le décodage de ces flotteurs nécessitant un suivi des messages de configuration reçus, il est impératif que le critère EXPECTED\_CYCLE\_LIST corresponde à « traiter tous les cycles reçus » ou « traiter les N premiers cycles reçus » (i.e. le traitement doit débuter au premier cycle et il ne doit pas y avoir de trous dans les numéros des cycles traités).

Une fois ce déplacement réalisé, le traitement se poursuit comme expliqué dans 1.4.1.

L’utilisation de ce décodeur nécessite de remplir préalablement le répertoire « archive » d’archivage des fichiers SBD de chaque flotteur traité (ex : **IRIDIUM\_DATA\_DIRECTORY/lovbio018c/archive/**).

Cela peut être réalisé avec l’utilitaire **copy\_remocean\_sbd\_files** qui effectue une copie des fichiers SBD du répertoire défini par le paramètre de configuration DIR\_INPUT\_RSYNC\_DATA vers le répertoire défini par le paramètre de configuration IRIDIUM\_DATA\_DIRECTORY. La gestion des paramètres d’entrée de ce programme de copie est identique à celle de **decode\_provor\_2\_csv**.

### Programme decode\_provor\_2\_nc

Les fichiers associés à ce programme sont :

* Entrées :
  + Fichier de configuration du décodeur : « matlab/\_prv\_decoder.conf »,
  + Fichier des paramètres de décodage des flotteurs déclarés : « prvFloatInfo/\_provor\_floats\_information\_co.txt »,
  + Fichier de description des paramètres de configuration des flotteurs V5.9 « json/configParamNames/\_config\_param\_name\_105.json »,
  + Fichier de description des paramètres techniques des flotteurs V5.9 « json/techParamNames/\_tech\_param\_name\_105.json »,
  + Fichiers de méta-données des flotteurs déclarés : ceux du répertoire « json/json\_float\_meta\_remocean » (ex : **6901528\_meta.json**),
  + Fichiers SBD d’entrée : ceux stockés dans le répertoire « archive » d’archivage des fichiers SBD de chaque flotteur (ex : **IRIDIUM\_DATA\_DIRECTORY/lovbio018c/archive/**).
* Sorties :
  + Fichiers log créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_LOG\_FILE,
  + Fichier NetCDF créés dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_NETCDF\_FILE.

Les fichiers NetCDF générés sont ceux demandés par les flags :

* + - GENERATE\_NC\_TRAJ,
    - GENERATE\_NC\_MULTI\_PROF,
    - GENERATE\_NC\_MONO\_PROF,
    - GENERATE\_NC\_TECH,
    - GENERATE\_NC\_META.

du fichier « matlab/**\_prv\_decoder.conf** ».

Paramètres d’entrée du programme :

* En l’absence d’argument, le décodage s’applique à tous les flotteurs de la liste définie par le paramètre de configuration FLOAT\_LIST\_FILE\_NAME,
* Il est cependant possible de préciser les flotteurs à décoder en fournissant en argument leur numéro WMO (ex : decode\_provor\_2\_nc(6901528) ou decode\_provor\_2\_nc(6901472, 6901482, 6901529)).

La première étape du décodage consiste à déplacer les fichiers SBD du répertoire « archive » vers le répertoire « spool ». Seuls les fichiers SBD dont le numéro de cycle répond aux critères mentionnés par le paramètre de configuration EXPECTED\_CYCLE\_LIST sont déplacés.

**Attention :** le décodage de ces flotteurs nécessitant un suivi des messages de configuration reçus, il est impératif que le critère EXPECTED\_CYCLE\_LIST corresponde à « traiter tous les cycles reçus » ou « traiter les N premiers cycles reçus » (i.e. le traitement doit débuter au premier cycle et il ne doit pas y avoir de trous dans les numéros des cycles traités).

Une fois ce déplacement réalisé, le traitement se poursuit comme expliqué dans 1.4.1.

L’utilisation de ce décodeur nécessite de remplir préalablement le répertoire « archive » d’archivage des fichiers SBD de chaque flotteur traité (ex : **IRIDIUM\_DATA\_DIRECTORY/lovbio018c/archive/**).

Cela peut être réalisé avec l’utilitaire **copy\_remocean\_sbd\_files** qui effectue une copie des fichiers SBD du répertoire défini par le paramètre de configuration DIR\_INPUT\_RSYNC\_DATA vers le répertoire défini par le paramètre de configuration IRIDIUM\_DATA\_DIRECTORY. La gestion des paramètres d’entrée de ce programme de copie est identique à celle de **decode\_provor\_2\_nc**.

### Programme decode\_provor\_2\_nc\_rt

Les fichiers associés à ce programme sont :

* Entrées :
  + Fichier de configuration du décodeur : « matlab/**\_prv\_decoder\_conf.json**»,
  + Fichiers des paramètres de décodage de chaque flotteur déclaré : ceux du répertoire « json/json\_float\_info » (ex : **6901528\_lovbio018c\_info.json**),
  + Fichier de description des paramètres de configuration des flotteurs V5.9 « json/configParamNames/\_config\_param\_name\_105.json »,
  + Fichier de description des paramètres techniques des flotteurs V5.9 « json/techParamNames/\_tech\_param\_name\_105.json »,
  + Fichiers de méta-données des flotteurs déclarés : ceux du répertoire « json/json\_float\_meta\_remocean » (ex : **6901528\_meta.json**),
  + Fichiers SBD d’entrée : ceux stockés dans le répertoire défini par le paramètre de configuration DIR\_INPUT\_RSYNC\_DATA,
  + Fichiers log de collecte rsync : ceux stockés dans le répertoire défini par le paramètre de configuration DIR\_INPUT\_RSYNC\_LOG,
* Sorties :
  + Fichiers log créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_LOG\_FILE,
  + Fichier NetCDF créés dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_NETCDF\_FILE.

Les fichiers NetCDF générés sont ceux demandés par les flags :

* + - GENERATE\_NC\_TRAJ,
    - GENERATE\_NC\_MULTI\_PROF,
    - GENERATE\_NC\_MONO\_PROF,
    - GENERATE\_NC\_TECH,
    - GENERATE\_NC\_META.

du fichier « matlab/**\_prv\_decoder\_conf.json** ».

* + Fichier xml de rapport d’exécution créé dans le répertoire défini par le paramètre de configuration DIR\_OUTPUT\_XML\_FILE,
  + Fichiers de stockage **d’informations temporaires**.

Les **informations temporaires** sont stockées dans 2 fichiers situés dans le répertoire « tmp » de chaque flotteur traité (ex : **IRIDIUM\_DATA\_DIRECTORY/lovbio018c/tmp/**) :

* Le fichier « processed\_data\_*WMO*\_yyyymmddTHHMMSS.mat » contient toutes les informations présentes en mémoire à l’issue du traitement d’un flotteur donné et indispensables à la reprise d’une nouvelle session à venir (pour ne pas retraiter à chaque session toutes les données Iridium transmises par le flotteur depuis sa mise à l’eau),
* Le fichier « processed\_rsync\_log\_*WMO*.txt » contient la liste des logs rsync utiles à ce flotteur et déjà traités.

Paramètres d’entrée du programme :

* Le décodage s’applique par défaut à tous les flotteurs de la liste définie par le paramètre de configuration FLOAT\_LIST\_FILE\_NAME,
* Paramètre **obligatoire** :
  + Le paramètre **‘rsynclog’** doit **nécessairement** être fourni au programme. Il peut :
    - Spécifier de traiter tous les logs rsync du répertoire défini par le paramètre de configuration DIR\_INPUT\_RSYNC\_LOG :

decode\_provor\_2\_nc\_rt(‘rsynclog’, ‘all’)

* + - Spécifier de traiter un seul log rsync du répertoire défini par le paramètre de configuration DIR\_INPUT\_RSYNC\_LOG :

decode\_provor\_2\_nc\_rt(‘rsynclog’, ‘rsync\_20130908191704.txt’) par exemple

* Paramètres **optionnels** :
  + Le paramètre optionnel **‘floatwmo’** permet de restreindre le décodage à un flotteur en particulier :

decode\_provor\_2\_nc\_rt(‘rsynclog’, ‘rsync\_20130908191704.txt’, ‘floatwmo’, ‘6901528’) par exemple

* + Le paramètre optionnel **‘floatwmolist’** permet de restreindre le décodage à une liste de flotteurs en particulier :

decode\_provor\_2\_nc\_rt(‘rsynclog’, ‘rsync\_20130908191704.txt’, ‘floatwmolist’, ‘/home/coriolis\_dev/val/binlx/co04/co0414/co041402/exe/lists/new\_remocean.txt’) par exemple

* + Chaque paramètre du fichier « matlab/**\_prv\_decoder\_conf.json**» peut être remplacé par un paramètre de même nom passé en argument du programme :

decode\_provor\_2\_nc\_rt(‘rsynclog’, ‘rsync\_20130908191704.txt’, ‘GENERATE\_NC\_MULTI\_PROF’, ‘0’) par exemple

La première étape du décodage consiste à :

* Analyser le contenu du(des) log(s) rsync passé(s) en argument et présent(s) dans le répertoire défini par le paramètre de configuration DIR\_INPUT\_RSYNC\_LOG afin d’établir la liste des SBD concernés par la session du décodeur. Lors de cette étape, on exclut du traitement les logs rsync déjà traités (présents dans le fichier « tmp/processed\_rsync\_log\_*WMO*.txt »).
* Copier les fichiers SBD concernés du répertoire défini par le paramètre de configuration DIR\_INPUT\_RSYNC\_DATA vers le répertoire « spool » de chaque flotteur.

Une fois cette copie effectuée, le traitement se poursuit comme expliqué dans 1.4.1.

## Décodage temps réel des flotteurs Remocéan par le DAC Coriolis

### Déclaration des flotteurs

Le flotteur 6901528 (login name : lovbio018c) est **déclaré** lorsque l’on a généré ses fichiers :

* « json/json\_float\_info/6901528\_lovbio018c\_info.json »,
* « json/json\_float\_meta\_remocean/6901528\_meta.json».

### Lancement de la première session du décodeur

Démarche à suivre pour initier le décodage temps réel des flotteurs Remocéans à Coriolis :

1. Etablir la liste des flotteurs Remocéans **déclarés** et la mettre dans la liste FLOAT\_LIST\_FILE\_NAME du fichier « matlab/**\_prv\_decoder\_conf.json** »,
2. Suspendre l’éventuelle collecte via rsync en crontab,
3. Supprimer le contenu des répertoires définis par les variables de configuration IRIDIUM\_DATA\_DIRECTORY et DIR\_OUTPUT\_NETCDF\_FILE,
4. Lancer le décodeur avec la commande :

decode\_provor\_2\_nc\_rt(‘rsynclog’, ‘all’)

1. Attendre la fin de cette session (d’une durée actuelle d’environ 6 heures) puis mettre en crontab la collecte rsync suivie du lancement du décodeur avec la commande :

decode\_provor\_2\_nc\_rt(‘rsynclog’, ‘rsync\_*yyyymmddHHMMSS*.txt’)

où rsync\_*yyyymmddHHMMSS*.txt désigne le nom du log rsync généré par la collecte concernée.

**Attention** : le cadencement, via la crontab, de la collecte rsync suivie du décodage doit être effectué en évitant que deux sessions de décodage ne soient lancées simultanément.

### Intégration d’un nouveau flotteur

Dans l’hypothèse où le nouveau flotteur a déjà transmis des données, il est nécessaire de les décoder avant d’intégrer ce flotteur aux opérations de collecte et décodage communes à tous les autres flotteurs. Il est en particulier impératif de ne « manquer » aucun log rsync issu des collectes.

Démarche à suivre pour ajouter un nouveau flotteur au décodage temps réel des flotteurs Remocéans à Coriolis :

1. **Déclarer** le nouveau flotteur (ex : 6901529),
2. Effectuer le premier décodage du nouveau flotteur avec la commande

decode\_provor\_2\_nc\_rt(‘rsynclog’, ‘all’, ‘floatwmo’, ‘6901529’)

1. Des collectes rsync ayant pu avoir lieu pendant ce décodage, il peut être nécessaire de relancer cette commande plusieurs fois jusqu’à ce que la session lancée n’entraine plus aucun décodage de données,
2. S’assurer qu’il n’y a pas de collecte rsync en cours avant d’ajouter le nouveau flotteur à la liste FLOAT\_LIST\_FILE\_NAME du fichier « matlab/**\_prv\_decoder\_conf.json** ».