

**Institut Universitaire de Technologie,  
Aix-Marseille Université**

**ANNEXES**  
**Bachelor Universitaire de Technologie**  
**Spécialité Réseaux et Télécommunications**  
**parcours cybersécurité**

**Gestion de données scientifiques au format  
NetCDF  
Interopérabilité et Programmation Python**

**Killian PAVY**

**OSU Institut Pythéas**

**Responsable entreprise : Maurice LIBES**

**Responsable académique : Djamel MERAD**

**2023**



## Table des matières

1. Les Fonctions et leur utilité.....	4
2. Programme de répartition en dossiers par années .....	5
3. Images complémentaires.....	6

## 1. Les Fonctions et leur utilité

### ➤ `load_config(config_file)` :

- `config_file` : le chemin du fichier de configuration à charger.

Cette fonction charge un fichier de configuration (`config_file`) qui est un fichier texte contenant des paires clé-valeur sous la forme clé=valeur. La fonction retourne un dictionnaire contenant les clés et les valeurs du fichier de configuration.

### ➤ `Write_Metadata_NC(ncFile, metadataFileCSV)` :

- `ncFile` : l'objet NetCDF dans lequel écrire les métadonnées.
- `metadataFileCSV` : le chemin du fichier CSV contenant les métadonnées.

Cette fonction lit les métadonnées à partir d'un fichier CSV et les écrit dans l'objet NetCDF spécifié.

### ➤ `add_zeros_to_date(date, separator)` :

- `date` : la date au format de chaîne de caractères à laquelle ajouter des zéros.
- `separator` : le séparateur utilisé dans la date (par exemple, '/' ou '-').

Cette fonction ajoute des zéros à la date et à l'heure pour formater correctement les valeurs.

### ➤ `convert_date_to_iso(date_str)`:

- `date_str` : la date au format de chaîne de caractères à convertir en format ISO.

Cette fonction convertit une date au format "jour/mois/année heure:minute" en format ISO "année-mois-jourTheure:minute".

### ➤ `read_and_write_csv(csv_filename, OutCSV_DIR)`:

- `csv_filename` : le chemin du fichier CSV à lire.
- `OutCSV_DIR` : le répertoire de sortie où écrire le nouveau fichier CSV.

Cette fonction lit le fichier CSV, stocke les données dans un dictionnaire et écrit les données dans un nouveau fichier CSV.

### ➤ `Write_csv(data_dict, num_records, OutCSV_DIR, csv_filename)`:

- `data_dict` : le dictionnaire contenant les données à écrire dans le fichier CSV.
- `num_records` : le nombre total d'enregistrements de données.
- `OutCSV_DIR` : le répertoire de sortie où écrire le fichier CSV.
- `csv_filename` : le nom du fichier CSV à créer.

Cette fonction écrit les données du dictionnaire dans un nouveau fichier CSV avec un format approprié.

### ➤ `write_nc(nc_filename, data_dict, metadataFileCSV)`:

- `nc_filename` : le chemin du fichier NetCDF à créer.
- `data_dict` : le dictionnaire contenant les données à écrire dans le fichier NetCDF.
- `metadataFileCSV` : le chemin du fichier CSV contenant les métadonnées.

Cette fonction crée un fichier NetCDF, écrit les métadonnées et les données du dictionnaire dans le fichier avec le format approprié.

## 2. Programme de répartition en dossiers par années

```
#!/bin/bash
## repartition des fichiers EMSO dans de srepertoires mensuels
## une fois par mois on deplace les fichiers journaliers dans
## des repertoire mensuels pour classer les fichiers par cron

## idee: deplacer les 30 fichiers journaliers dans un repertoire mensuel
## ou aggreger les fichiers netcdf avec ncrctat en UN seul fichier mensuel

deplace_fic()
{
rac=$1
ext=$2
echo "traite fichiers $rac extension $ext"
am=$(ls *.${ext} | cut -d_ -f2 | cut -d. -f1 )
listeam=$(for i in $am; do echo ${i:0:6}; done | uniq)
for am in $listeam
do
aa=${am:0:4}
mm=${am:4:6}
echo "$aa-$mm"
if [ ! -d "$aa-$mm" ]; then
echo "mkdir $aa-$mm"
mkdir "$aa-$mm"
fi
# Modification pour prendre "CTAR" si rac est "CSTAR"
if [ "$rac" == "CSTAR" ]; then
rac="CTAR"
fi
# on deplace les fichier mensuels dans le repertoire mensuel
echo "mv ${rac}_${aa}${mm}*.${ext} $aa-$mm/"
mv ${rac}_${aa}${mm}*.${ext} $aa-$mm/
done
}

#===== prog principal =====

racine="/mnt/EMSO/EMSO_Data/DATA_MII/"

directories=(
"EMSO_OutNetCDF-V2/Emso_Cstar"
"EMSO_OutNetCDF-V2/Emso_Microcat"
"EMSO_OutNetCDF-V2/Emso_Oxygen"
"EMSO_OutNetCDF-V2/Emso_Aquadopp"
"EMSO_OutCSV-V2/Emso_Cstar"
"EMSO_OutCSV-V2/Emso_Microcat"
"EMSO_OutCSV-V2/Emso_Oxygen"
"EMSO_OutCSV-V2/Emso_Aquadopp"
)
```

```

for directory in "${directories[@]}"
do
echo "cd \"${racine}$directory\"""
cd "${racine}$directory"

rac=$(basename "$directory")
filetype_folder=${directory%%/*}
emso_sensor=${directory##*/}
uppercase=${emso_sensor^^}
sensor=${uppercase:5}
echo $filetype_folder

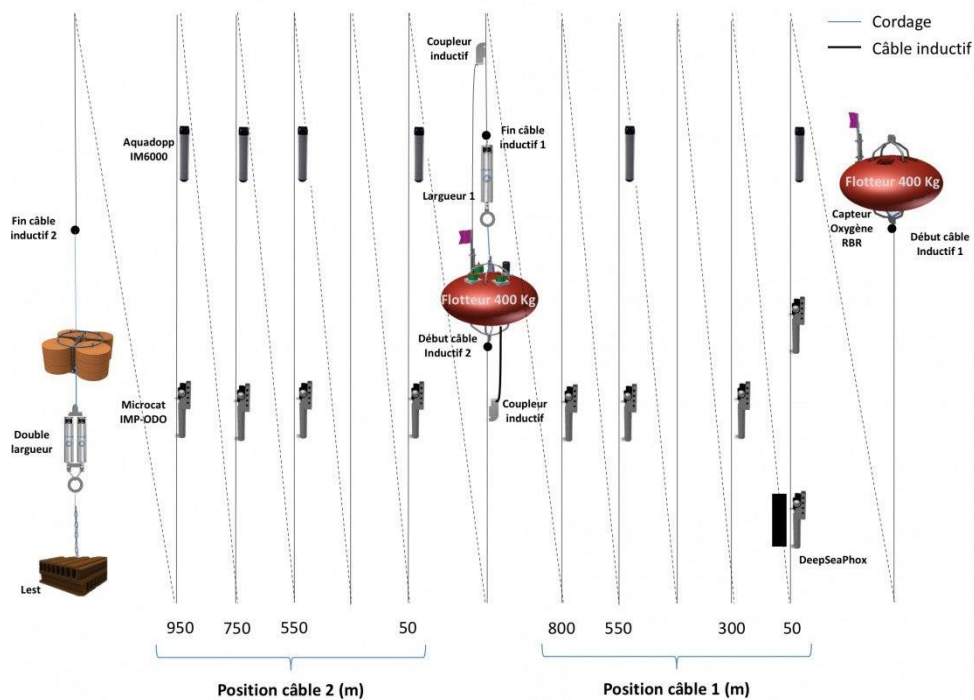
if [[ $filetype_folder == "EMSO_OutNetCDF"* ]]; then
deplace_fic $sensor "nc"
elif [[ $filetype_folder == "EMSO_OutCSV"* ]]; then
deplace_fic $sensor "csv"
fi
done

```

### 3. Images complémentaires

#### Ligne de mouillage instrumentée inductive ALBATROSS

Configuration 09/2020



ALBATROSS : Autonomous Line with a Broad Acoustic Transmission for Research in Oceanography and Sea Sciences

Figure 1 : Ligne de mouillage instrumentée inductive Albatross



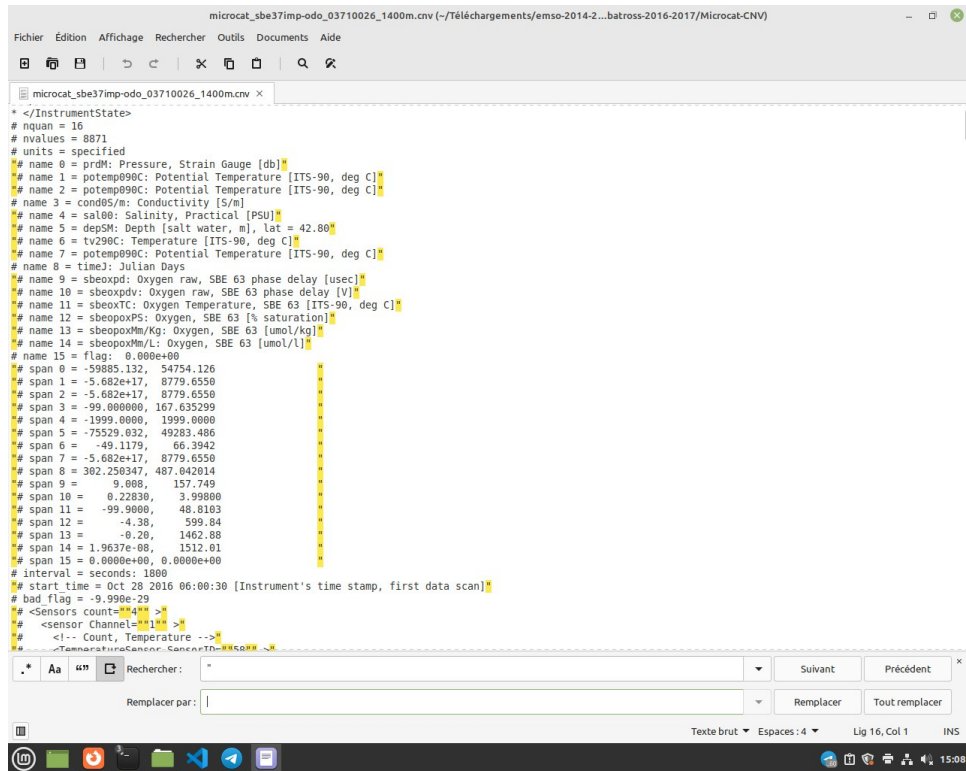


Figure 27 : Exemple de fichier .cnv brut corrompu

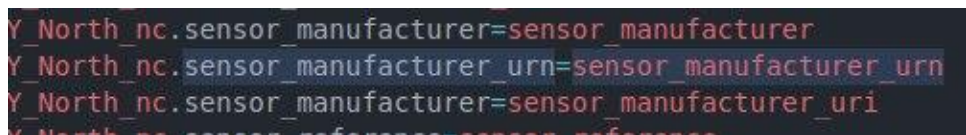


Figure 32 : Erreur d'inattention (double sensor\_manufacturer)

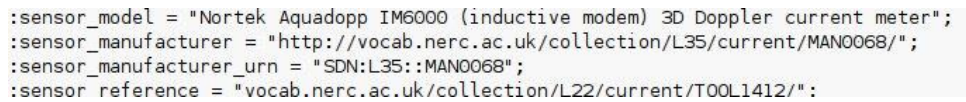


Figure 33 : Attribut sensor\_manufacturer\_uri manquant