Notice d’utilisation de keilμ5 et modification du payload

Contents

1. Objectif

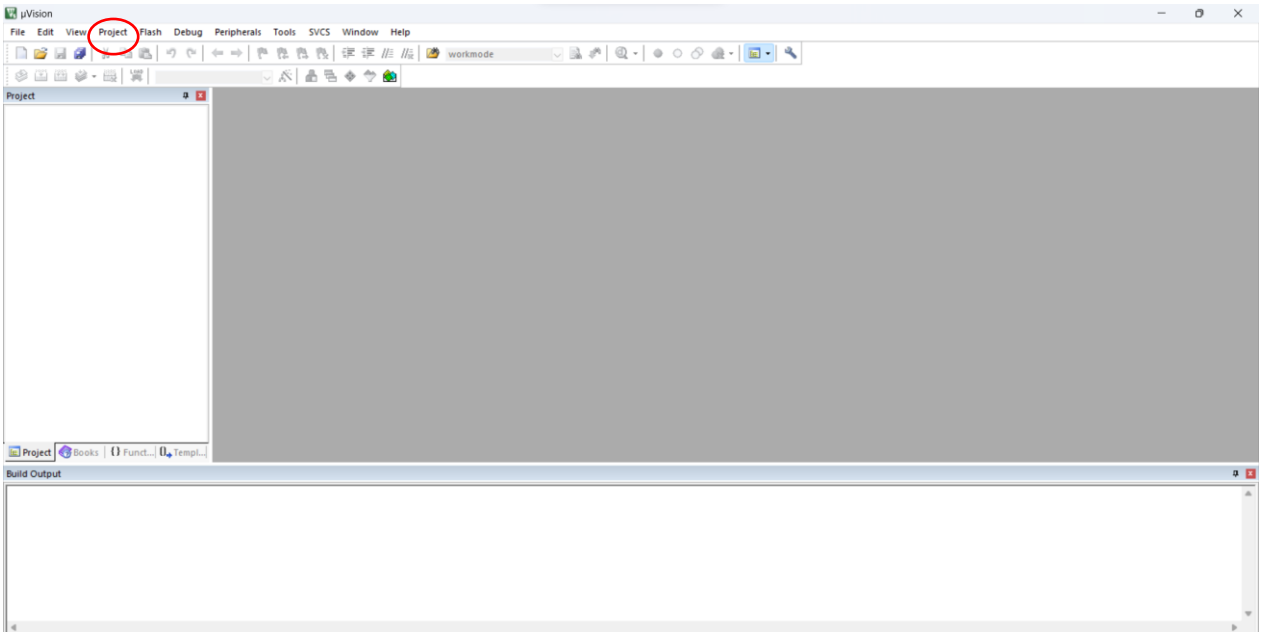
L'objectif de cette notice est de pouvoir prendre en main Keilµ5 de manière à pourvoir utiliser rapidement ce logiciel pour modifier le code source du dragino. A la fin se trouve une explication des modifications faites au payload pour correspondre mieux et avoir une approche plus cohérente.

Keil est le logiciel qu'il est conseillé d'utiliser par ceux qu'ils l'ont conçu. Lorsque nous avons commencé à travailler sur le projet, nous avions rencontré plusieurs erreurs qui ont freiner fortement notre avancée, et les documents mis à notre disposition par les étudiants de l'année précédentes ne nous étaient pas d'une grande utilité. Cette notice permet donc de montrer les étapes à faire pour utiliser le logiciel ainsi que quelques manipulations qui permettent d'empêcher les problèmes lors de la compilation.

NB : Nous conseillons de créer un github qui permettra aux étudiants de se passer le projet facilement et sans trop de problèmes.

2. Ouvrir un projet

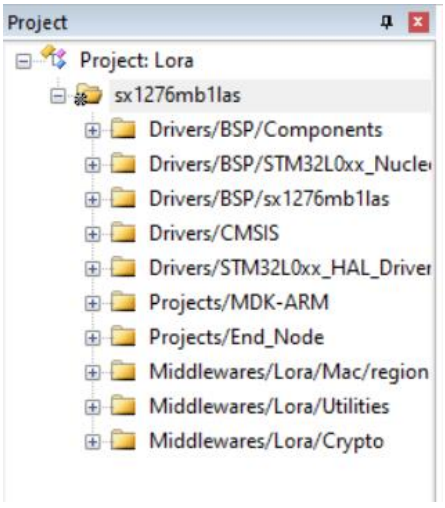
Si vous n'avez pas de projet déjà ouvert sur Keil, le logiciel devrait ressembler à cela :



Cliquez alors sur l'onglet « project » puis « open project »

Puis aller cherche le projet dans vos dossiers, à l'intérieur, il se trouve dans : IOT_Fournier_Lacheze\STM32CubeExpansion_LRWAN\Projects\Multi\Applications\LoRa\DRAGINO-LRWAN(AT)\MDK-ARM\STM32L072CZ-Nucleo\Lora.uvprojx"

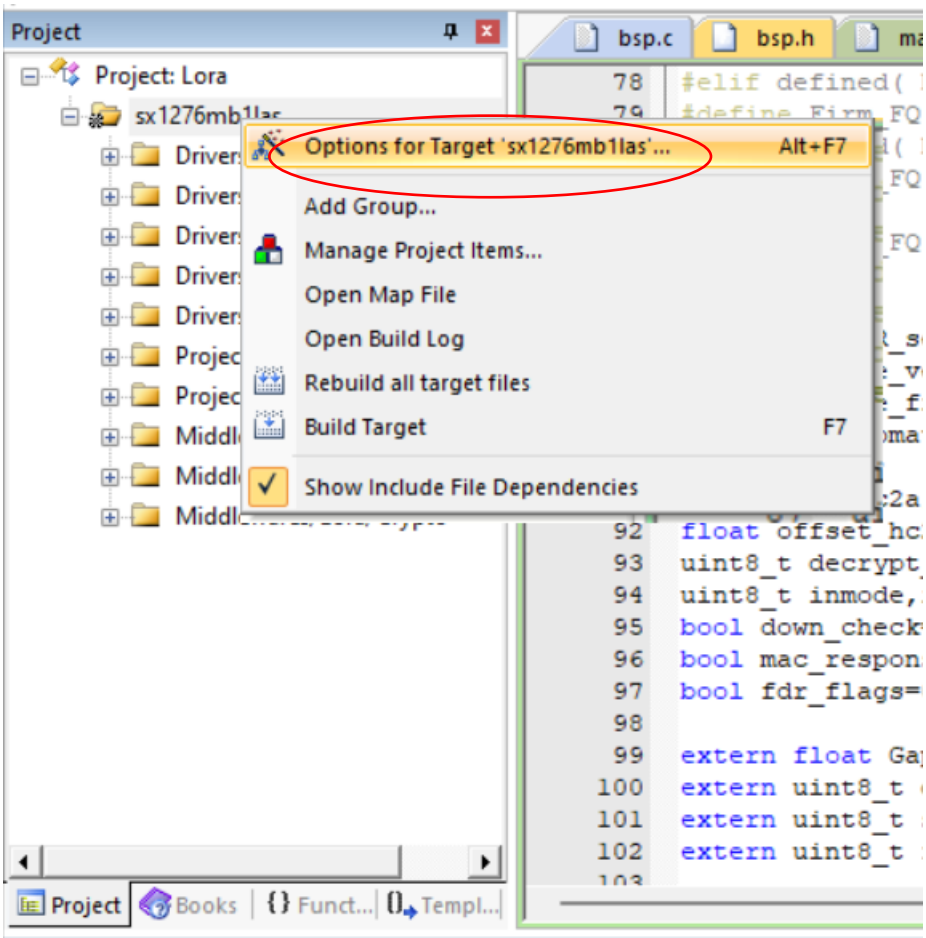
Vous devriez avoir le projet qui apparait à gauche avec les différents dossiers :



3. Générer le fichier exécutable qui sera flashé dans le dragino

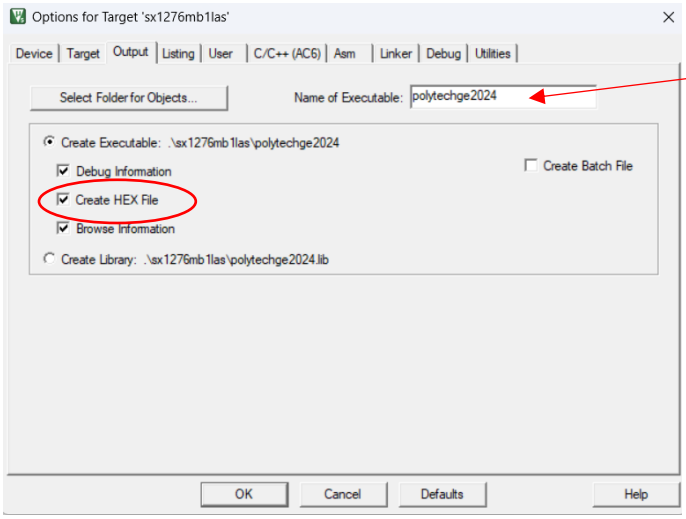
Le fichier .HEX est le fichier qui sera mis dans le dragino via le logiciel stm32cubeprogrammer et qui contiendra le code transformé pour la mesure des sondes.

A gauche dans sx1276mb1las, faire clique droit puis « options for target »



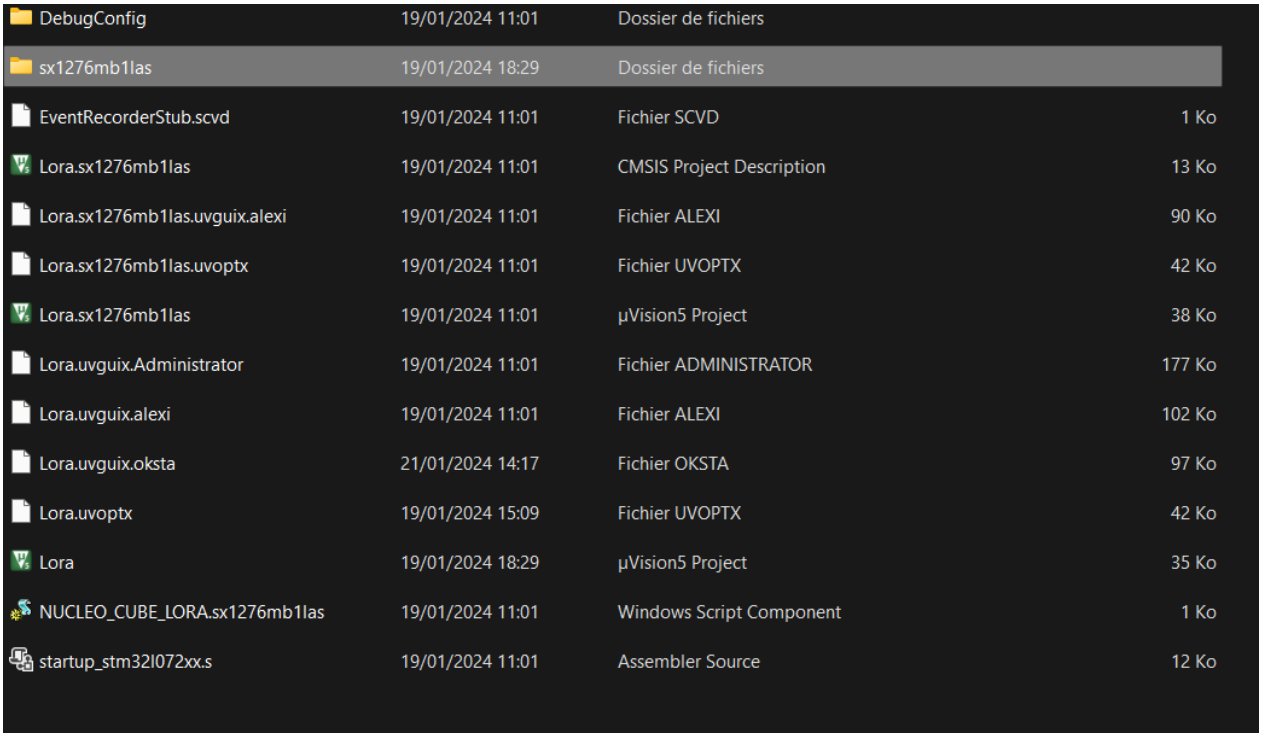
Cela ouvre les options du projet

Dans l'onglet « output », cocher « create HEX file » ce qui va générer automatiquement le fichier .HEX lors de la compilation du projet. Il est aussi possible de modifier le nom de l'exécutable.



Nom du fichier .HEX

Le fichier .HEX se trouve dans à côté de l'emplacement ou se trouve le projet dans ce fichier.



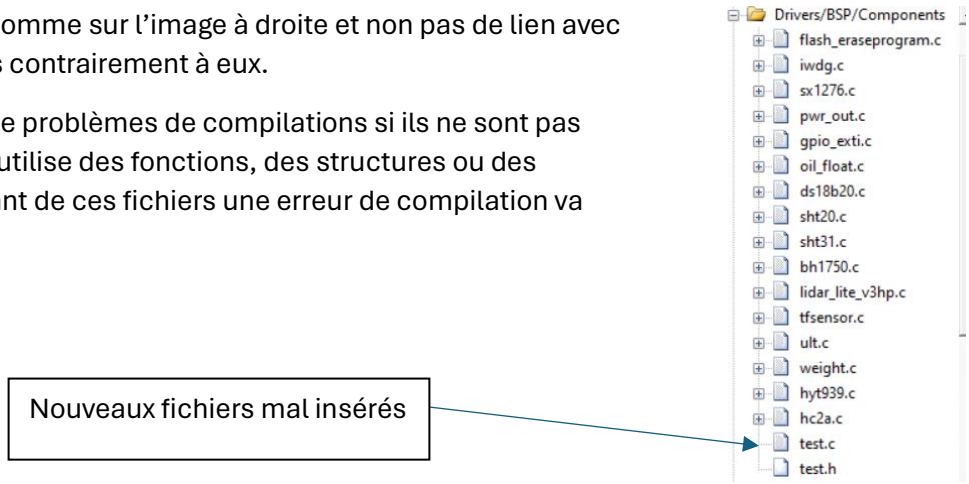
Ce sera là ou il faudra aller chercher le fichier pour le flasher dans le dragino.

4. Problème d'insertion de fichiers

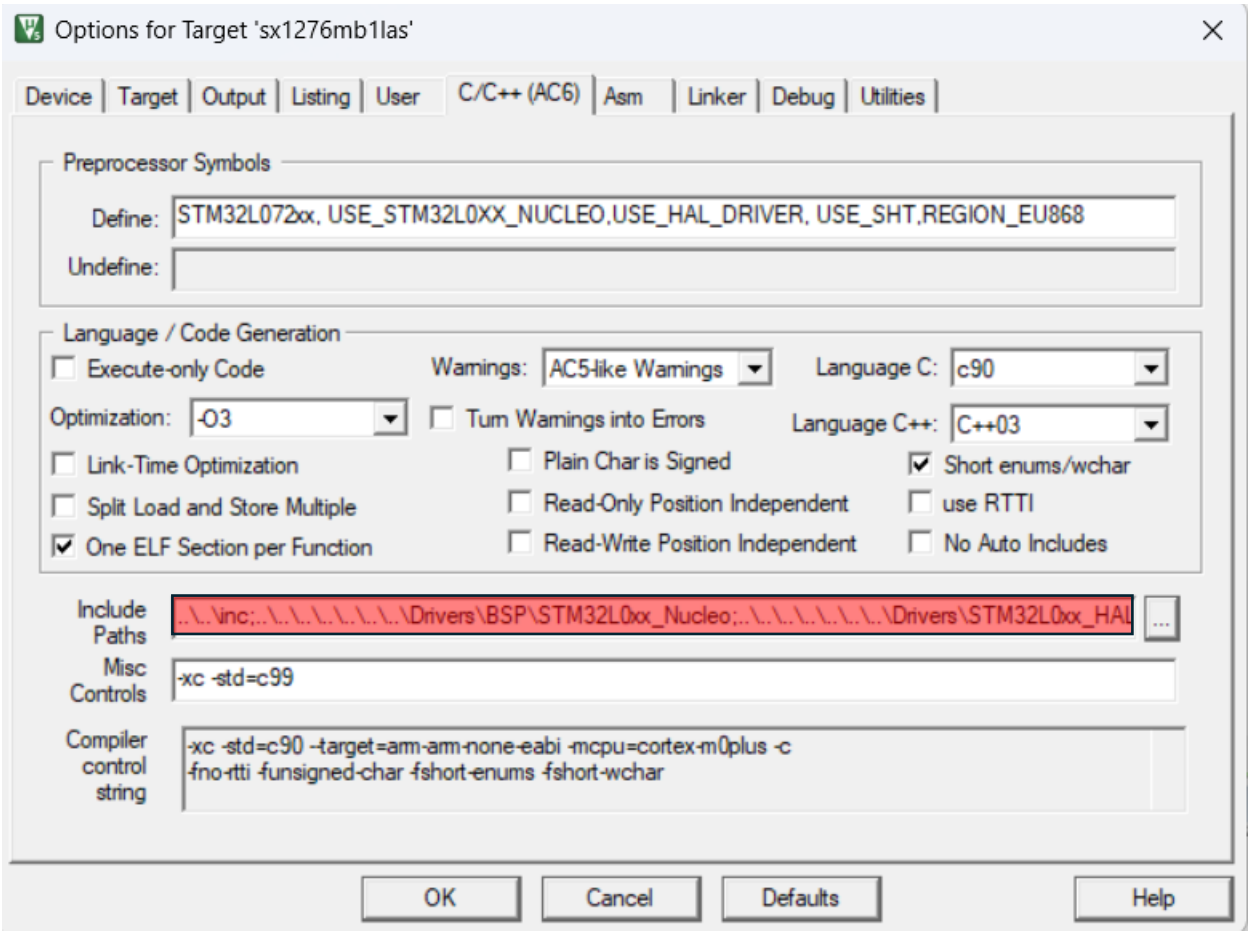
Il arrive parfois qu'un fichier que l'on insère dans le projet ne soit pas réellement rattaché au projet même si on l'a inséré . Généralement, ce sera un driver pour un nouveau type de sonde.

Ils apparaissent comme sur l'image à droite et non pas de lien avec les autres fichiers contrairement à eux.

Il ne posent pas de problèmes de compilations si ils ne sont pas utilisé mais si on utilise des fonctions, des structures ou des variables provenant de ces fichiers une erreur de compilation va apparaitre



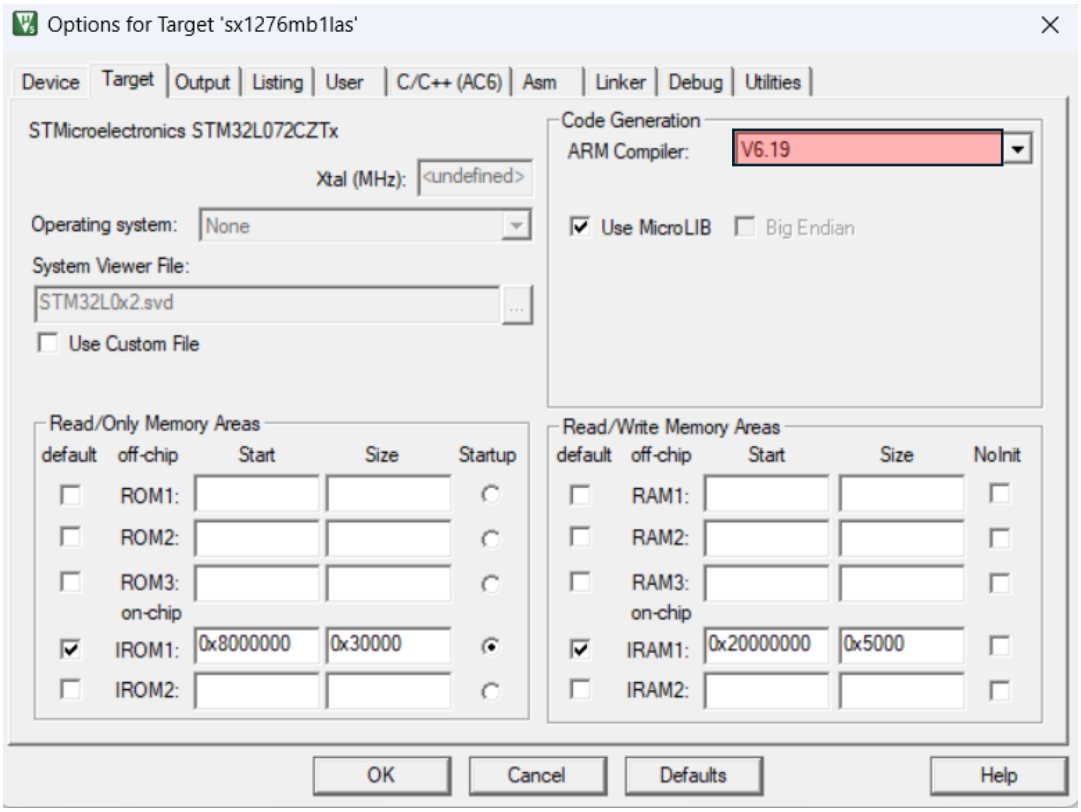
Il faut donc les insérer manuellement dans le projet, cela se fait dans les options du projet dans l'onglet c/c++ ou il faut rajouter le chemin du fichier dans « include path »



Lors de la compilation, il ne devrait puis y avoir d'erreur et le nouveau fichier devrait ressembler aux autres.

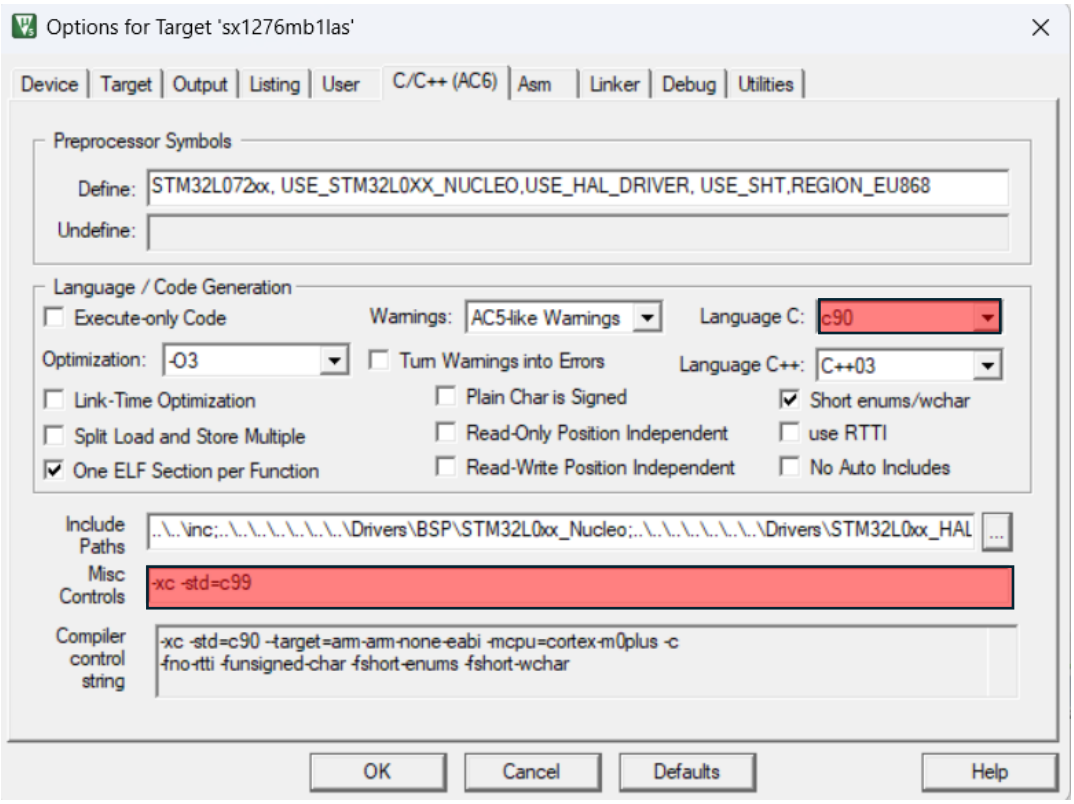
5. Autres erreurs de compilation que nous avons trouvés

La version du compilateur peut ne pas être bonne. Dans ce cas aller dans les options du projet onglet « target » et choisir la bonne version du ARM compiler, ici la 6.19. Si besoin la télécharger sur internet.



Autres problèmes :

Il peut arriver qu'il y ait d'autres problèmes lors de la compilation, dans ce cas vérifier que ces paramètres soit mis dans les options :



6. Ecriture du payload

Le payload est écris en javascript et est mis sur TTN pour décoder les trames. Le payload d'origine était celui fourni pour le dragino et à été modifié tout d'abord par les étudiants de l'année dernière puis pas nous-même. Dans le payload, l'octet contenant le mode du capteur est sur l'octet 6 de la trame envoyé. On y fait un masque et un décalage car pour les autres modes, l'octet 6 permet de fournir d'autres informations. Ce décalage et le masque font qu'il n'y a qu'en réalité 32 modes différents possible (31 à cause du mode alarme). Or les étudiants avaient arbitrairement choisis le mode 40 pour se trouver loin dans le cas ou le dragino auraient eu des mises à jour de mode.

Donc en réalité le mode 40 du dragino correspond au mode 30 du payload sur TTN. Nous avons donc modifié le mode sur le projet afin que les modes utilisé par les étudiants correspondent.

Le mode des sondes hyt939 est désormais le mode 20 et le mode hc2a est le mode 21. Cela laissera de la place pour d'autres modes de fonctionnement pour d'autres sondes.