

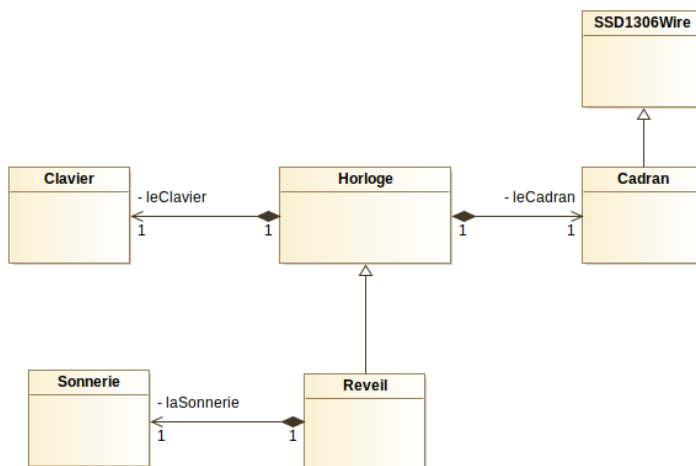
Prérequis : La librairie pour piloter un écran SSD1306 OLED 0,96" (128x64) avec un ESP32 doit être installée. L'écran est connecté via le bus I2C à l'ESP32.

Le projet **Horloge** est fonctionnel en ressource les fichiers sources sont disponibles.

1- Création du projet

Avec l'outil **PlatformIO**, initialisez un nouveau projet nommé **Reveil** et ouvrez-le avec votre environnement de développement.

Dans un premier temps, vous allez recopier dans le répertoire **src** de votre nouveau projet, les différents fichiers du projet **Horloge** réalisés lors du **TD3**. Compilez et téléversez le projet sur la cible, le projet doit être à nouveau opérationnel.

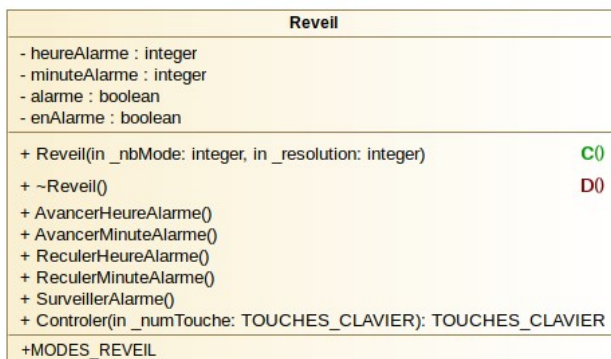


L'objectif final est de réaliser l'application en respectant le diagramme de classes ci-contre :

- Pour cela, ajoutez une nouvelle classe nommée **Reveil** à votre projet. Cette classe hérite de la classe **Horloge**.
- Ajoutez également une nouvelle classe nommée **Sonnerie** qui sera étudiée par la suite.

2- Mise au point de la classe Reveil

À partir des éléments écrits dans le cours d'introduction à la programmation-objet, compléter la classe **Reveil** pour être conforme à la figure suivante :



Complétez le code des méthodes permettant le réglage de l'heure et de la minute de l'alarme.

Dans un premier temps, réalisez le squelette de la méthode **SurveillerAlarme()**.

À partir du diagramme état transition de la classe **Reveil** et les éléments du cours, complétez le codage de la méthode **Controler()**

À ce stade, vous pouvez vérifier les différents modes du réveil en remplaçant dans le programme principal l'instance de la classe **Horloge** par une instance de la classe **Reveil**.

Surchargez la méthode **Afficher()** de la classe **Cadran** pour qu'elle puisse recevoir deux objets **String** du genre « AL » et « ON » ou « AL » et « OFF »

3- Réalisation de la classe Sonnerie

Notre carte ne disposant pas physiquement de buzzer, nous allons remplacer son fonctionnement en allumant ou en éteignant une LED.

Sonnerie
- broche : uint8_t
+ Sonnerie(in _broche: uint8_t) C0
+ Declencher()
+ Arrêter()

À partir de la représentation graphique de la classe **Sonnerie**, complétez les fichiers contenant cette classe.

Par défaut, le constructeur reçoit le numéro de la broche correspondant à la LED1 de la carte (voir dans le cours) et initialise cette broche pour la piloter en sortie.

La méthode **Declencher()** place la broche au niveau HAUT et la méthode **Arrêter()** la place au niveau BAS.

4- Intégration avec la classe Clavier

Réalisez en C++ la relation de composition entre la classe **Clavier** et la classe **Sonnerie**.

Complétez la méthode **Surveiller()** de la classe **Clavier**. Pouvez-vous utiliser les attributs de la classe **Horloge** ?

Solution : modifiez la section privée de la classe **Horloge** pour qu'elle devienne une section protégée afin de pouvoir comparer **heureAlarme** et **heures** entre autres...

Vérifiez le fonctionnement de votre réveil.