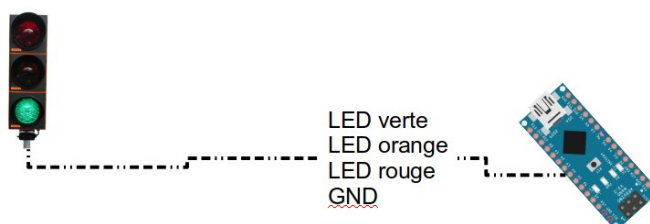


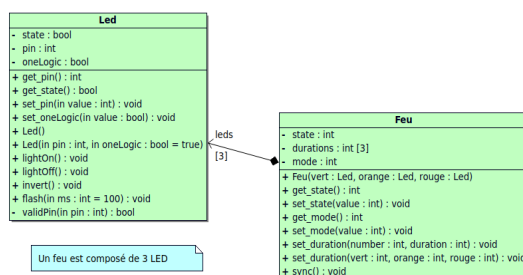
## Un Feu basé sur trois LED

En repartant de la classe Led développée au TP1, nous vous proposons de simuler le fonctionnement d'un feu de signalisation routière, "feu rouge" basé sur trois LED directement connectées aux ports de la carte.



### Fonctionnalités de base :

- 3 états : vert, orange et rouge, (*state*)
- 3 modes de fonctionnement (0=manuel, 1=automatique, 2=orange clignotant).
- 3 durées d'allumage (*durations*)
- pouvoir régler les durées d'allumage (du rouge, de l'orange, du vert)
- pouvoir changer de mode : *automatique* ou *clignotant*.
- fonctionnement non-bloquant. Nous utiliserons une méthode **sync**, appelée dans la boucle principale.



Représentation UML

### On demande :

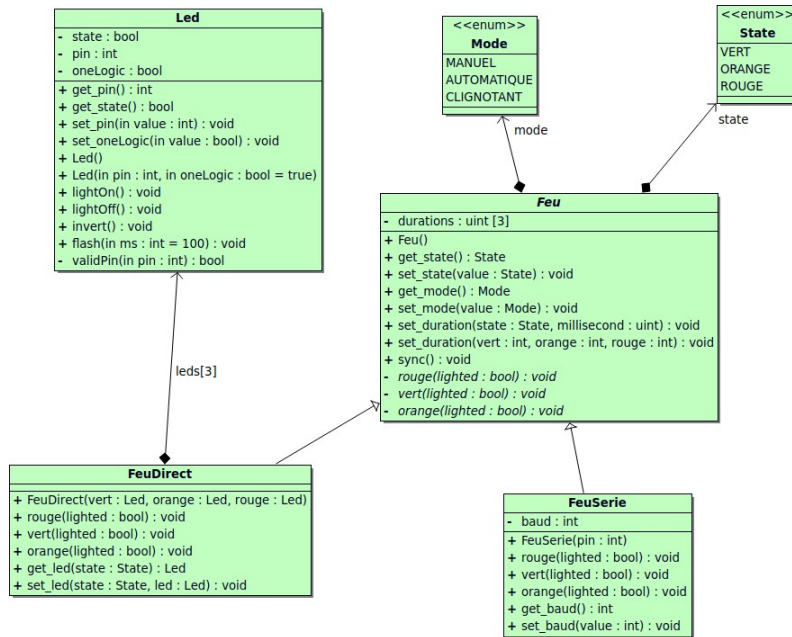
- Créer et implémenter la classe Feu (fichiers feu.h et feu.cpp)
- Tester le fonctionnement en permettant de changer le mode *via* la liaison série ; ex : '1' → automatique, '2' → clignotant.

## Plusieurs type de Feu

Nous voulons maintenant commander deux types de feu différents. Le premier utilise une connexion tout ou rien directe aux ports comme précédemment, le deuxième communique via une liaison série (Feu intelligent) ; on peut imaginer d'autres types de feu, ex : liaison I2C, SPI, .etc.

Pour la liaison série, nous allons supposer que l'envoi des chaînes de caractères "R0" éteint de rouge, "R1" allume le rouge, "O0" et "O1" éteint/allume l'orange et "V0" et "V1" éteint/allume le vert. Les paramètres de la liaison sont : 115200 bauds, parité paire, 1 stop bit.

Nous pourrions recommencer tout le travail fait précédemment sur le feu directement connecté aux ports pour le feu communicant en série. Hors, si on réfléchit bien, on voit bien que quelque soit le mode de communication utilisé, le fonctionnement de tous les feux est identique. L'héritage va permettre de créer une classe abstraite, **Feu**, ayant le comportement commun à tous les feux puis deux classes héritées **FeuDirect** et **FeuSerie** ayant uniquement la partie propre à chacun d'eux. (un **FeuDirect** est un **Feu**, un **FeuSerie** est un **Feu**, .etc.).



Représentation UML

Les particularités propres à chaque **Feu** sont représentées par les méthodes *rouge*, *vert* et *orange* qui permettent de changer de couleur. Elles ont été définies *abstraite* dans la classe mère pour être sûr qu'elle seront bien redéfinies dans les classes filles.

#### Marche à suivre :

Déclarer et implémenter les méthodes de la classe mère puis vérifier qu'il est impossible de l'instancier (classe abstraite).

Déclarer et implémenter la classe fille **FeuDirect** puis tester le fonctionnement au moyen des trois LED fournies.

Déclarer et implémenter la classe fille **FeuSerie** puis tester le fonctionnement au moyen du programme de simulation d'un feu série donné lors de la séance.