

Documentation Utilisateur

Tout le code se fait dans ~/Projet_Long/catkin_ws/src/planificateur/src/planificateur.cpp
Les fonctions accessibles sont incluses dans les objets "capteurState.h" , "switchState.h" et "stopState.h". Les 3 objets sont déjà déclarés et initialisés:

```
«
    sensorState SensorSt;
    SensorSt.init(nh);
    switchState switchSt;
    switchSt.init(nh);
    stopState stopSt;
    stopSt.init(nh);
»
```

Cette initialisation permet l'accès aux fonctions de l'utilisateur, qui sont :

- Dans SensorSt les états des capteurs (uniquement en lecture) :

```
SensorSt.CPI[1..8]
SensorSt.CP[1..10]
SensorSt.PS[1..16]
SensorSt.DG[1..12]
SensorSt.DD[1..12]
```

Chacune de ces valeurs sont à 0 ou à 1 selon si le capteur est activé (1) ou non (0)

- Dans switchSt les fonctions qui contrôlent les aiguillages:

```
switchSt.unlockSwitch(i), qui déverrouille l'aiguillage i
switchSt.lockSwitch(i), qui verrouille l'aiguillage i
switchSt.turnLeft(i), qui tourne l'aiguillage i vers la gauche
switchSt.turnRight(i), qui tourne l'aiguillage i vers la droite
```

i allant de 1 à 12

Un aiguillage ne peut pas tourner tant qu'il n'est pas déverrouillé (un message d'erreur s'affiche alors). Attention, pensez également à reverrouiller l'aiguillage une fois tourné.

- Dans stopSt les fonctions qui contrôlent les points d'arrêts:

```
stopSt.stop(i), qui met le point d'arrêt i en mode stop
stopSt.go(i), qui met le point d'arrêt i en mode continu
```

i allant de 1 à 24

Les fonctions qui contrôlent les stops et les aiguillages n'ont juste qu'à être appelées, les publications ROS sont faites au sein des fonctions.

Dans le code, il y a la partie déclaration des variables qui permet de déclarer en amont les variables, et la partie code qui se situe dans le « while(ros::ok) ». Ce while est l'équivalent d'un while(1), qui tourne donc en boucle (à une fréquence de 25Hz), et s'arrête uniquement lorsque vous quittez la simulation (par un Ctrl+C).