```
/*
 * Fonctions spécifiques à la gestion globale de la barrière
 */

#ifndef _BARRIERE_H
#define _BARRIERE_H

#include <Socket.h>
#include <string.h>

// Cette méthode permet d'activer la montée de la barrière.
// Si la barrière est déjà en haut, aucune action n'est réalisée.
inline
void setCommandeSensMontee(Socket &barriere)
{
  barriere.envoyer("Barriere:setCommandeSensMontee");
  barriere.recevoir();
}

// Cette méthode permet d'activer la descente de la barrière.
// Si la barrière est déjà en bas, aucune action n'est réalisée.
inline
void setCommandeSensDescente(Socket &barriere)
{
  barriere.envoyer("Barriere:setCommandeSensDescente");
  barriere.recevoir();
}

// Indique l'état du capteur "Fin De Course Haut"
// permettant de savoir si la barrière est en position haute.
inline
bool isInFDCH(Socket &barriere)
{
  barriere.envoyer("Barriere:isInFDCH");
  return strcmp(barriere.recevoir(), "haut") == 0;
}


// Indique l'état du capteur "Fin De Course Bas (Low)"
// permettant de savoir si la barrière est en position basse.
inline
bool isInFDCL(Socket &barriere)
{
  barriere.envoyer("Barriere:isInFDCL");
  return strcmp(barriere.recevoir(), "bas") == 0;
}

// Indique l'état du capteur de présence de la voiture en entrée.
inline
bool isBoucleAmont(Socket &barriere)
{
  barriere.envoyer("Barriere:isBoucleAmont");
  return strcmp(barriere.recevoir(), "entree") == 0;
}

// Indique l'état du capteur de présence de la voiture en sortie.
inline
bool isBoucleAval(Socket &barriere)
{
  barriere.envoyer("Barriere:isBoucleAval");
  return strcmp(barriere.recevoir(), "sortie") == 0;
```

```
}

#endif
```