**Propositions de designs pour le projet RobLip6**

1. **Design Général :**

HIGH LEVEL (Carte Firefly avec ROS)

Rôle :

* Décision en fonction des acquisitions de la caméra, du lidar et des capteurs du LOW LEVEL
* Envoi des ordres de haut niveau au LOW LEVEL

LOW LEVEL (Microcontrôleur MBED ou FPGA)

Rôle :

* Transformation des ordres de haut niveau en un ordre de bas niveau compréhensible pour les actionneurs.
* Envoi des données des capteurs au HIGH LEVEL

Valeurs brutes des capteurs

Ordres de haut niveau

Données brutes

Images

Valeurs brutes

Ordres de bas niveau

Lidar

Actionneurs

Capteurs

Caméra

Figure 1 : Design haut niveau du projet

1. **Designs possibles de l’application java pour la carte HIGH LEVEL :**

Classes de gestion de capteurs reliés directement à la carte HIGH LEVEL (caméra, lidar)

Ordres

Centre de décision

Classes de gestion de communication vers la carte LOW LEVEL

(ETHERNET ou UART)

Données brutes des capteurs internes

Données brutes des capteurs externes

Classes de génération de la carte des obstacles

(OPENCV)

Carte des obstacles

API

Figure 2 : Design haut niveau de l’application JAVA pour la carte HIGH LEVEL

1. Communication vers la carte LOW LEVEL
2. TCP/IP

* Avantages : fiabilité, débit important
* Inconvénients : lourdeur

write()

sendData()

TCPInterface

ConnexionTCPClient

Représente la connexion avec le client

read()

getData()

Connexion

TCPServer

Accepte la connexion

Création de la connexion

HIGH LEVEL

LOW LEVEL via ETHERNET

Figure 3 : Design de l’interface TCP côté serveur (HIGH LEVEL). En rouge l’initialisation de la connexion et en bleu la lecture et l’écriture.

1. UDP/IP

* Avantages : débit très important, légèreté
* Inconvénients : fiabilité

UDPInterface

sendData()

write()

UDPConnexion

Représente la connexion UDP avec le LOW LEVEL

getData()

read()

HIGH LEVEL

LOW LEVEL via ETHERNET

Figure 4 : Design de l’interface UDP

1. UART

* Avantages : fiabilité, simplicité
* Inconvénients : débit plus faible que pour UDP et TCP

sendData()

UARTInterface

write()

UARTCommunication

Représente la communication UART avec le LOW LEVEL

getData()

read()

HIGH LEVEL

LOW LEVEL via USB

Figure 5 : Design de l’interface UART

1. Gestion des capteurs « internes » (reliés directement à la carte HIGH LEVEL)
2. Génération de la carte des obstacles
3. Centre de décision
4. **Designs possibles pour la carte LOW LEVEL**