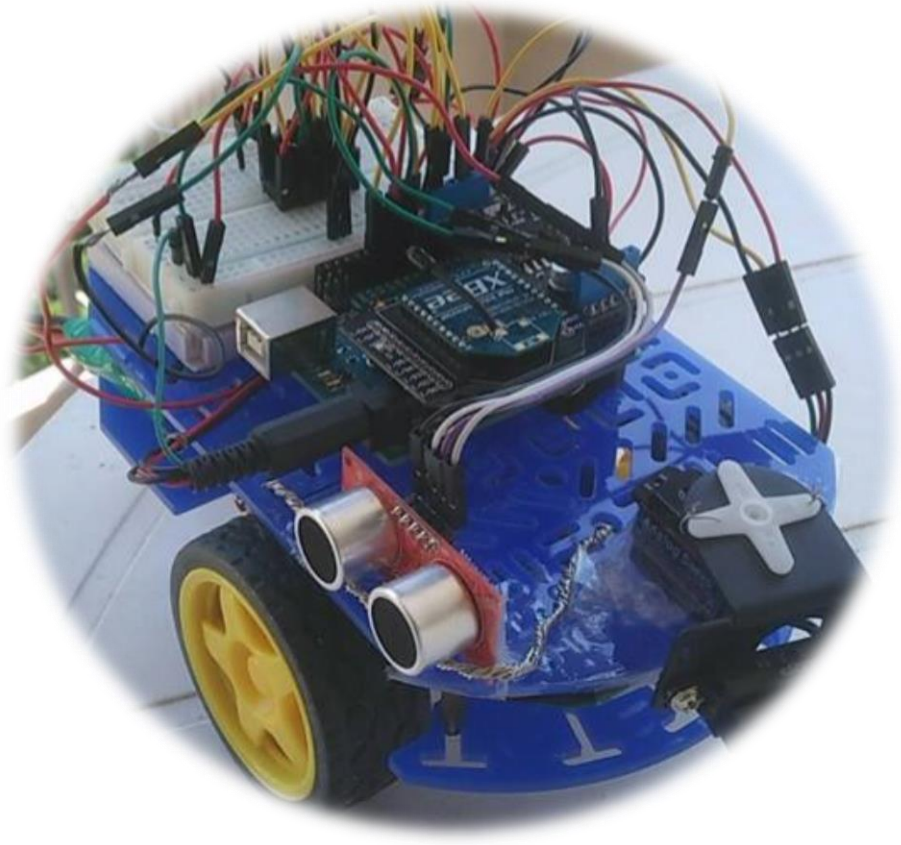
UPSSITECH
ÉCOLE D'INGÉNIEURS INTERNE

Année 2016 - 2017

# Projet Xion
## Rapport Final



PARTIE 3

## CARTOGRAPHIE ROBOTIQUE D'UNE SALLE


Enseignant responsable :
Michaël LAUER

CALESTANI Brice | CORTACERO Kévin | MORANT Thibaut
PECCHIOLI Mathieu | SAUVAGE Thomas

# Table des matières

I. Introduction ..... 3
II. Rappel du cahier des charges ..... 3
III. Analyse du cahier des charges ..... 3
1. Analyse modulaire de fonctionnement ..... 3
2. Communication des composants ..... 3
3. Listes des composants ..... 5
IV. Conception ..... 6
1. Robot mobile ..... 6
2. Communication ..... 7
3. Système décisionnel ..... 7
4. Système de pilotage ..... 7
a) Clavier ..... 7
b) Manette ..... 8
5. Simulation physique ..... 8
6. Interface graphique ..... 8
7. Affichage de l'état du robot ..... 8
8. Affichage de la puissance des moteurs ..... 8
9. Gestion de l'état du robot ..... 8
10. Verrou / sécurité du robot (Homme mort) ..... 8
11. Mapping de l'environnement ..... 8
V. Intégration ..... 8
VI. Validation ..... 8
VII. Organisation ..... 8
1. Temporelle ..... 8
2. Physique ..... 8
3. Outils ..... 9
VIII. Bilans ..... 9

# I. Introduction

Le projet que nous allons vous présenter dans ce dossier a été développé dans le cadre d'un projet de robotique des étudiants de première année de l'UPSSITECH en spécialité Systèmes Robotiques et Interactifs (SRI). Ce projet, rentrant dans les maquettes pédagogiques sous le nom de projet fil rouge portera le nom de Projet Xion conformément aux vœux de l'équipe d'étudiant y ayant travaillé dessus.

L'objectif principal de ce développement est la « Réalisation d'un robot mobile autonome de cartographie ».

# II. Rappel du cahier des charges

Suite à la réception du cahier des charges par notre client, nous avons réalisé un tableau récapitulatif des fonctions contraintes que nous devions respecter. Nous avons également pris la décision d'enrichir les demandes initiales avec des compléments utiles à l'utilisateur :

| Fonction | Description | Flexibilité |
|---|---|---|
| FP1 | Le robot doit suivre les contours de la pièce | 0 |
| FP2 | Le logiciel doit disposer d'une interface graphique cartographiant la pièce en temps réel | 0 |
| FP3 | Le logiciel doit disposer d'une partie décisionnelle autonome (non embarqué sur le robot) | 0 |
| FP4 | Le logiciel doit disposer d'un mode de commande manuel | 0 |
| FP5 | Le logiciel doit permettre d'afficher l'état des capteurs du robot pour l'utilisateur | 0 |
| FP6 | La communication entre le robot et le logiciel doit être distante | 0 |
| FP7 | Le robot doit embarquer un système anticollisions | 0 |
| FS1 | Le logiciel peut disposer d'un mode de simulation autonome | 2 |

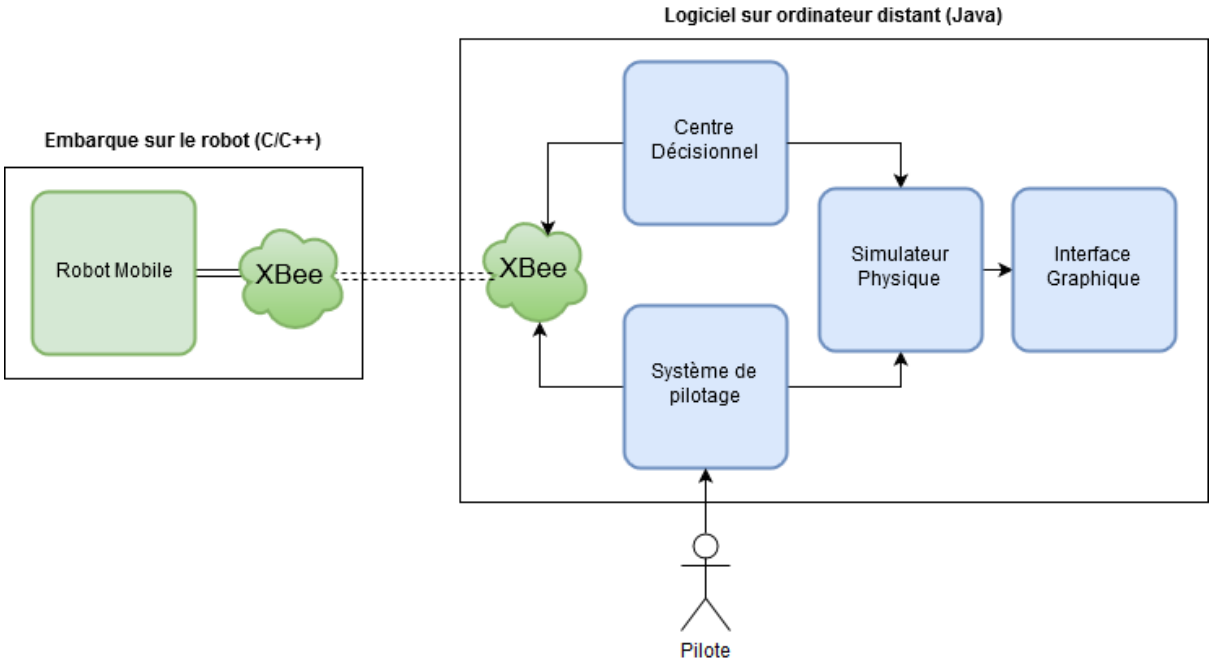
# III. Analyse du cahier des charges

## 1. Analyse modulaire de fonctionnement

Style de corps de texte. Les styles sont à copier avec l'outil de reproduction de mise en forme et non à modifier.

## 2. Communication des composants

Durant la partie précédente, nous avons réalisé l'analyse de la composition de notre produit. Dans la logique des choses, nous nous sommes donc ensuite intéressé à la réalisation d'une architecture modulaire nous permettant de remplir l'ensemble des tâches demandées. Cependant, Nous avons souhaité en plus de ce cahier des charges rendre notre produit le plus modulaire possible. De ce fait, l'architecture que nous avons développée s'agence tel que présenté ci-dessous :




Les différentes parties sont :

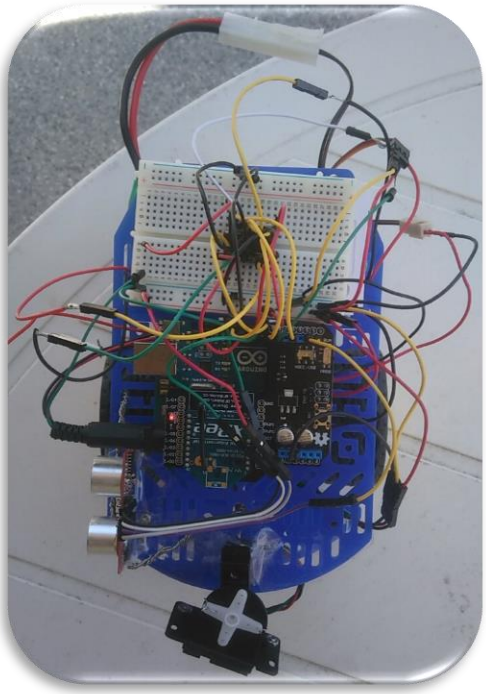
- *Robot mobile :* partie aussi bien matérielle que logicielle, elle comprend le montage du robot mobile ainsi que le développement du contrôle des actionneurs et des capteurs.
- *Xbee :* composé d'un point de vue matériel de deux xbee montés soit du cotés robot mobile sur un Shield Arduino soit sur un adaptateur USB cotés ordinateur, cette partie correspond à la liaison dématérialisée entre les deux blocs que sont le robot mobile et l'ordinateur.
- *Centre décisionnel :* ce module permet lorsque l'on est en mode automatique de définir le comportement du robot mobile à partir des données des capteurs
- *Simulateur physique :* véritable antichambre de l'interface graphique, le simulateur physique permet de générer la cartographie de la pièce ainsi que la position du robot. Il effectue la conversion entre le monde réel et les données virtuelles perçues.
- *Interface Graphique :* lien avec l'utilisateur, l'interface graphique permet d'informer l'opérateur et de lui fournir l'ensemble des données dont il a besoin pour commander en mode manuel par exemple
- *Système de pilotage :* utilisé en mode manuel, le système de pilotage permet de faire de lien entre les commandes entrée par l'utilisateur et les informations que le robot doit appliquer.

Les flèches de notre schéma représentent l'ensemble des communications entre les modules. Ces données transmises permettent à chacun de réaliser son travail de manière indépendant des autres.

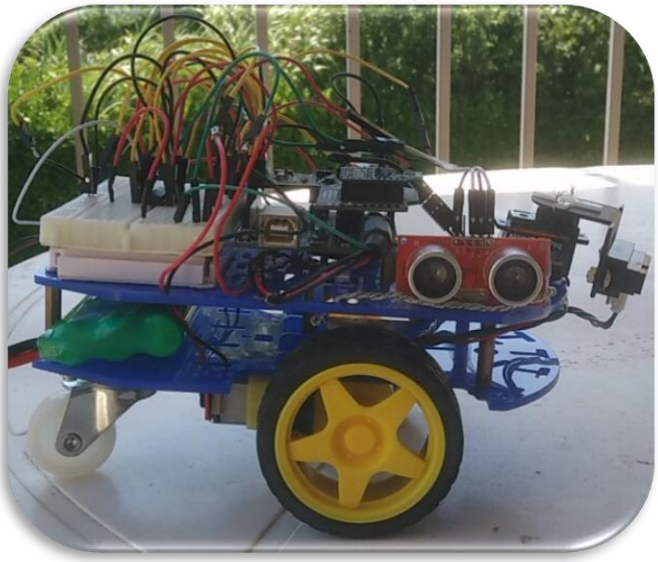
## 3. Listes des composants

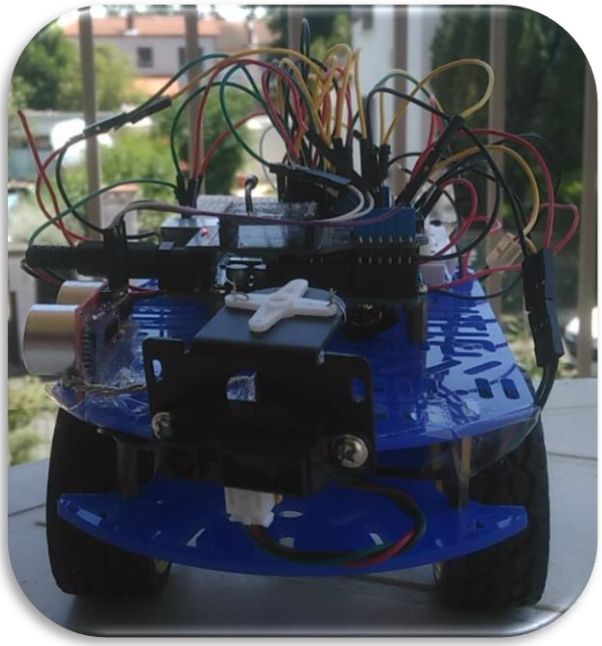
Pour mener à terme le projet Xion, nous avons eu besoin d'une partie matérielle. Voici donc la liste de l'ensemble des composants utilisés :

- ✓ 1 châssis robotique composer de deux plateformes espacées avec des entretoises, deux roues fixes montées sur des moteurs à courant continu et une roues folles pour obtenir une meilleure stabilité
- ✓ 1 batterie 7V – 2A
- ✓ 1 planche de prototypage type breadboard
- ✓ 1 servomoteur de petite puissance
- ✓ 1 capteur de distance ultrasons
- ✓ 1 capteur de distance infra-rouge
- ✓ 1 support en équerre pour le capteur infra-rouge et sa visserie
- ✓ 1 pont en H de type L293
- ✓ 1 Arduino Uno
- ✓ 1 Shield Xbee pour Arduino Uno
- ✓ 1 adaptateur USB pour Xbee
- ✓ 2 puces Xbee
- ✓ Des cables jumpers de prototypage rapide



L'ensemble de ses composants monté sur le robot mobile nous donnent un rendu final comme présenté sur les deux photos suivantes :





## 4. Choix technologiques matériel

Afin de réaliser les fonctionnalités voulues et nécessaire au projet Xion, nous avons dû faire des choix technologiques. Chaque choix a été réfléchis de manière à optimiser soit la fiabilité du produit final, soit la facilité de mise en œuvre.

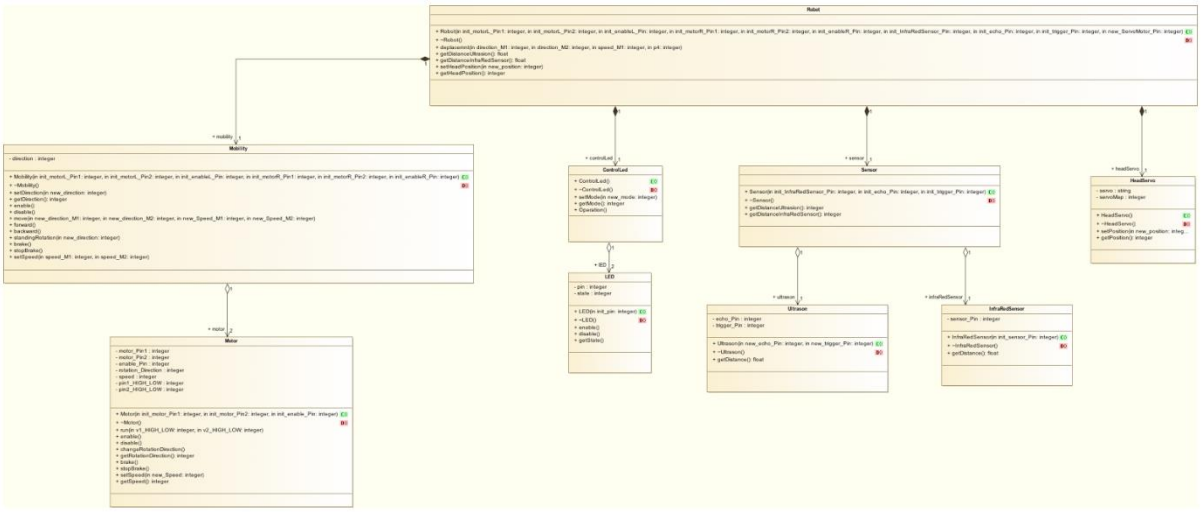
Prenons tout d'abord le cas des capteurs de distance. Nous aurions pu utiliser deux capteurs de distance de même type mais nous avons décidé de tirer parti de deux technologies que sont le capteur infra-rouge et le capteur ultrason de manière à utiliser le premier en qualité de parechoc avant et le second pour du mapping latéral. Afin de renforcer l'efficacité de notre parechoc avant et pour supprimer les angles morts, nous avons monté le capteur avant (capteur IR) sur un servomoteur vertical afin de réaliser un balayage. La fiabilité de la distance obtenue avec un capteur ultrason latéral a également été choisie.

Concernant l'utilisation d'une planche de prototypage et de jumper, cette solution a permis à l'équipe de développement de tester plusieurs configurations de matériel avec un temps de montage/démontage négligeable.

Pour l'autonomie, une batterie 7V – 2A permet un rechargement plus accessible comparé à un rack de piles. De plus, une fois chargé, l'autonomie globale du robot mobile s'en est vu grandement augmenté permettant la réalisation de cession de test bien plus longues.

# IV. Conception

## 1. Robot mobile




Pour la réalisation du programme contenu dans l'arduino, il était plus pertinent de concevoir directement une bibliothèque spécifique au robot permettant ainsi d'obtenir un code propre et structuré, mais aussi de partir à la découverte du C++.

Par ailleurs, il existe un Sketch qui est utilisé comme main afin de permettre une communication entre le robot et le centre décisionnel.

Pour la structure de la bibliothèque, il était intéressant de partir sur une classe, Robot, qui regroupe toutes les fonctionnalités du robot. Ainsi on obtient trois autres classes, Mobility, ControlLed, Sensor et HeadServo.

La classe Mobility est utilisé pour contrôler le déplacement du robot grâce à deux instance de la classe Motor, elle offre la possibilité de se déplacer en marche avant, marche arrière, tourner à droite ou à gauche en modifiant uniquement la vitesse ou en inversant le sens de rotation des moteurs. Il est aussi possible de s'arrêter en roue libre mais aussi en frein magnétique.

La classe ControlLed, permet de gérer l'affichage des différents modes d'utilisation du robot, c'est à dire, le mode manuel ou le mode automatique grâce à deux instances de la classe LED. La fonctionnalité est implémentée mais comme le robot n'est pas encore équipé de LEDs, elle n'est pas utilisée.

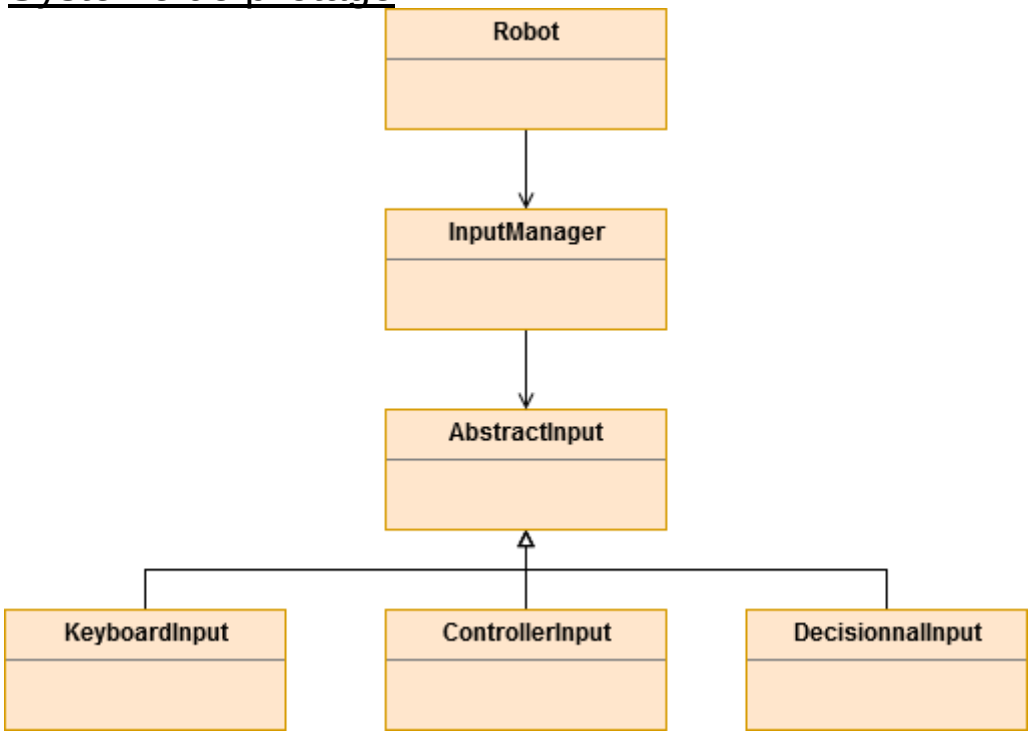
La classe Sensor, permet de gérer le retour des capteurs ultrason et infrarouge grâce au deux classes qu'ils leurs sont associés. L'ultrason est utilisé pour détecter des objets survenant du côté droits mais aussi de suivre un mur. L'infrarouge quant à lui, monté sur un servomoteur va servir uniquement de capteur tout ou rien afin de détecter les objets venant de face.

La classe HeadServo, est utilisé afin d'obtenir un champ de détection plus large pour les objets venant de face. Pour cela l'utilisation d'une instance de la classe servo est utilisé.

## 2. Communication

## 3. Système décisionnel

## 4. Système de pilotage




Le système de pilotage doit être une solution parallèle à la prise de décision du robot, nous avons donc généralisé la gestion des actionneurs du robot.

### a) Clavier

Nous avons donc choisi le pilotage par clavier car cela permet de s'abstenir de regarder l'écran pour piloter le robot dans la pièce. Cette solution est d'ailleurs plus confortable qu'un contrôle à la souris.

b) Manette

D'autre part, nous avons voulu ajouté la possibilité de diriger le robot par une manette quelconque (Xbox, PS4, PS3…) car les joysticks permettent de jauger la vitesse de déplacement et de rotation de manière plus précise. C'est d'ailleurs une solution couramment utilisée dans le pilotage de robots comme les drones.

5. Simulation physique

6. Interface graphique

7. Affichage de l'état du robot

8. Affichage de la puissance des moteurs

9. Gestion de l'état du robot

10. Verrou / sécurité du robot (Homme mort)

11. Mapping de l'environnement

# V. Intégration

# VI. Validation

# VII. Organisation

1. Temporelle

2. Physique

3. Outils

## VIII. Bilans