

Projet IMA3/4 : Robot Centaure

Année 2019/2020



Théo Evrard, Loïc Ringot, Valérie Salingue, Pierre Sanchez et Thibault Vollerin
05/06/2019

Table des matières

Introduction :	1
I- Présentation du projet et objectifs :	2
II- Réalisation des différentes tâches :	3
A) Modification du PCB et commande à distance :	3
B) Réorganisation du robot :	5
C) Software :	7
Conclusion :	8

Introduction :

Durant notre première année de cycle ingénieur dans le département IMA, nous avons pour but de réaliser un projet au choix. Cela a pour objectif de nous apprendre à gérer un projet d'étude ainsi que d'utiliser les compétences acquises durant notre année d'IMA3. Avec une équipe composée de 5 personnes (qui sont respectivement Théo Evrard, Loïc Lingot, Valériane Salingue, Pierre Sanchez et Thibault Vollerin), nous nous sommes intéressés au projet du Robot autonome Centaure. Celui-ci comportait plusieurs objectifs et consistait à améliorer un robot de grande taille (déjà existant et anciennement amélioré par des étudiants en IMA5 qui devaient, en partie, faire fonctionner le robot afin qu'il puisse se déplacer à l'aide d'une commande filaire).

En effet, il nous est demandé d'améliorer le robot afin qu'il soit utilisé comme guide pour les visiteurs voulant trouver leur chemin dans les différents bâtiments de Polytech. Par exemple, si le visiteur veut se rendre dans le bâtiment A, dans la salle 315, le robot va chercher le chemin le plus court et conduire le visiteur à cette salle, le robot pourra se repérer dans l'espace et éviter les obstacles se trouvant sur son passage. Enfin, il doit posséder un système de reconnaissance faciale et vocale ainsi qu'une synthèse vocale pour dialoguer avec l'utilisateur.

Nous avons donc commencé à prendre en main le robot et comprendre son fonctionnement ainsi qu'étudier les objectifs demandés avant de commencer à travailler sur le projet.

I- Présentation du projet et objectifs :

Le projet « robot Centaure » a débuté en 2008. Ce robot de grande taille (environ 1m52) est constitué d'une structure métallique et de plexiglass permettant de protéger la structure interne de celui-ci. Il utilise deux roues motorisées, utilisées entre autres pour des fauteuils roulants, et une roue libre pour se déplacer. Les deux roues sont chacune reliée à un variateur de vitesse permettant de contrôler la vitesse de rotation des deux roues. Ces variateurs sont à la fois reliés aux batteries ainsi qu'à la carte Arduino du robot (qui lui-même est branché au PC du robot qui sert d'unité centrale contrôlant tous les composants). Il est aussi composé de plusieurs capteurs infrarouges, d'un écran permettant d'afficher une interface sur le robot et d'une Kinect.

Voici un schéma fait par l'ancien groupe travaillant sur le projet afin de voir plus clairement la composition du robot :

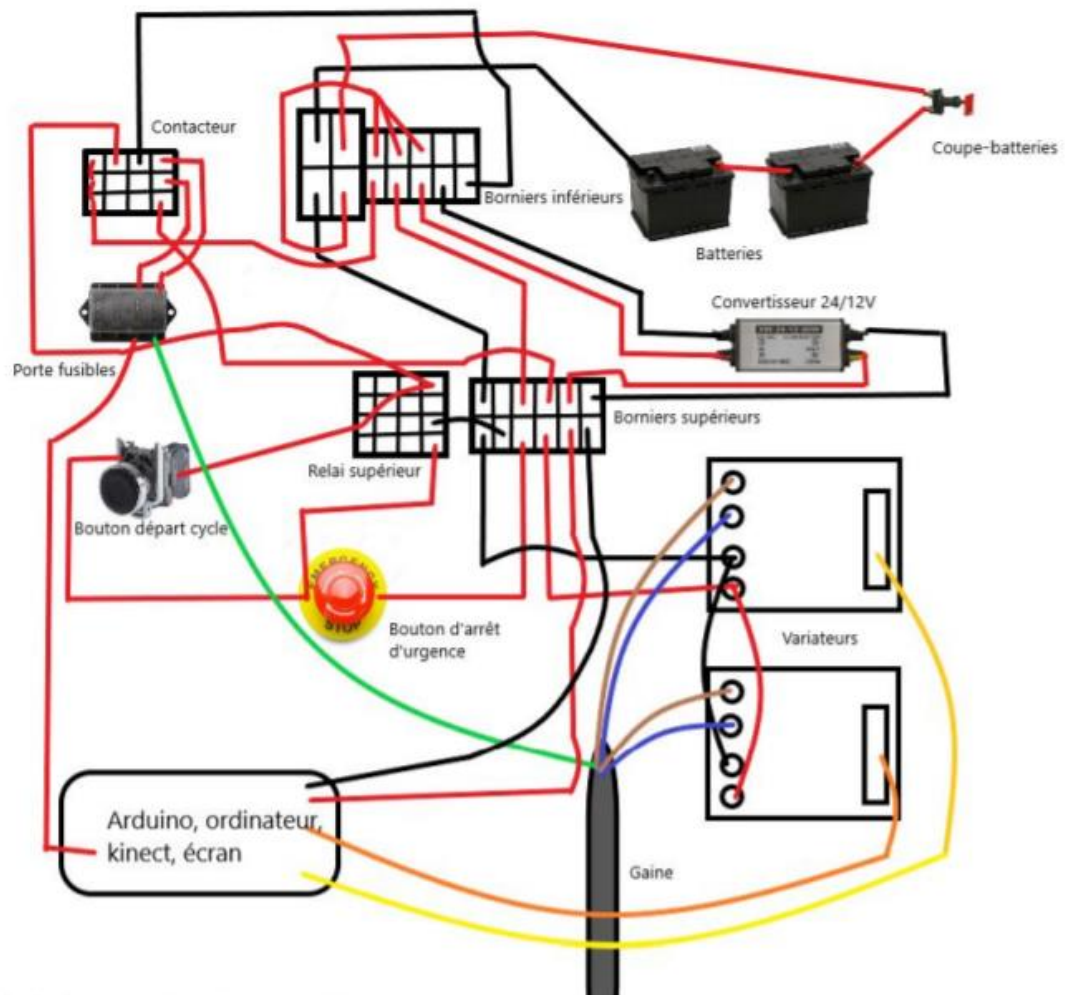


Schéma de câblage général du robot

Nous avons donc commencé par voir comment allumer le robot et comment le contrôler. Après avoir procédé à l'allumage du robot, nous avons pu tester ce qu'avait fait les cinquièmes

années concernant le contrôle et le déplacement du robot. Ensuite, nous avons chacun réfléchi à un objectif à réaliser.

Nous avons donc commencé la reconnaissance vocale, faciale et la synthèse vocale et nous avons commencé à chercher un moyen au robot de se repérer dans les bâtiments de Polytech. Puis, en voulant afficher sur l'écran du robot l'interface du PC, nous nous sommes rendu compte que l'écran s'éteignait sans raison. Nous avons donc essayé de trouver la cause du problème et nous avons remarqué que l'écran n'était pas réellement éteint, mais que le rétroéclairage de l'écran et la luminosité avaient chuté brutalement mais que l'écran était tout de même allumé.

Nous avons donc fait un test de tension pour voir si ce n'était pas un problème d'alimentation. Et en effet, le problème venait du convertisseur DC/DC de 24/12V qui ne fournissait pas exactement 12V, ce qui fait que l'écran était sous-alimenté. L'écran se mettait alors à une luminosité qu'il pouvait assurer en fonction de la tension qu'il recevait. Plus tard, on remarquera que l'écran doit en réalité être alimenté en 13,3V, donc le problème ne venait pas du convertisseur.

Nous avons donc dû changer nos objectifs pour ce semestre car les tâches que nous devions réaliser n'aurait pas pu être faites dans les temps compte tenu des différentes corrections à faire sur le robot.

II- Réalisation des différentes tâches :

A) Modification du PCB et commande à distance :

Nous avons choisi de modifier le PCB réalisé par l'ancien groupe pour y implémenter des borniers à vis. Cela permettra d'avoir des branchements plus solides. Pour ce faire, nous avons dû télécharger le logiciel Altium Designer nous permettant de modifier leur schéma. Cette partie nous aura donc permis d'acquérir des connaissances sur ce logiciel ainsi que sur la création de PCB en général.

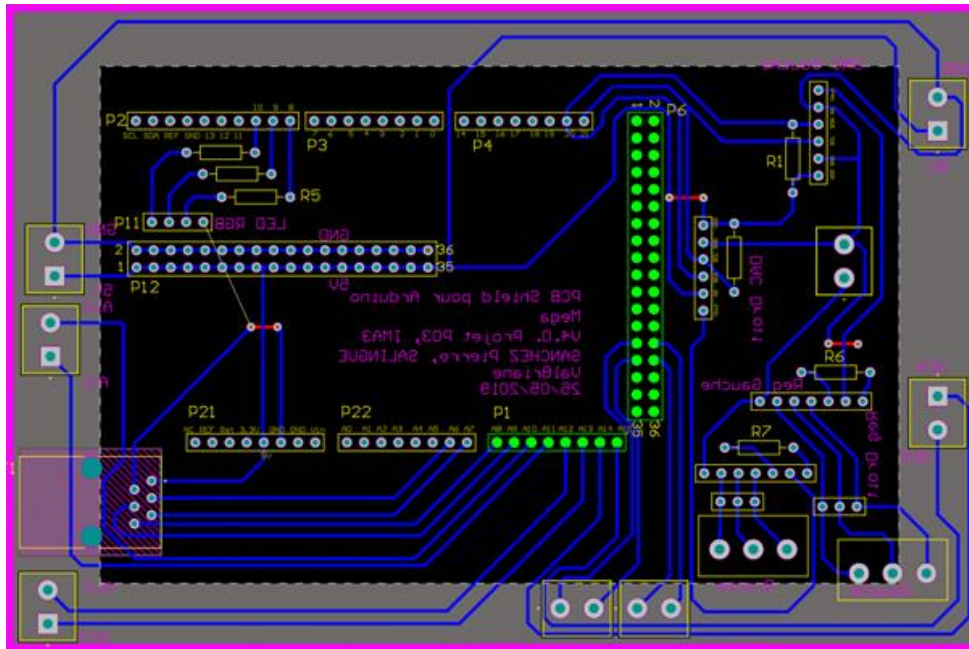


Image du nouveau PCB réalisé sous Altium Designer

Ensuite, nous avons pris connaissance du serveur Web réalisé par les IMA5. Il nous permet de pouvoir contrôler le robot à distance grâce à un joystick disposé sur une page web, celui-ci est réalisé en Node.js, nous avons choisi de le modifier pour comprendre comment celui-ci était fait. Nous avons donc rajouté sur leur page Web deux boutons « Allumer Led » et « Eteindre Led ».

Nous avons modifié le code pour qu'à l'appui d'un de ces boutons, une série de caractères soit envoyée par la liaison série à la carte Arduino. Nous avons également écrit un programme sous Arduino nous permettant d'allumer ou d'éteindre une led lorsqu'il recevait certains caractères sur sa liaison série.

```
#define LED 10

void setup() {
  // put your setup code here, to run once:
  pinMode(LED, OUTPUT);
  Serial.begin(115200);
}

void loop() {
  int i=0;
  char data[10];
  // put your main code here, to run repeatedly:
  while(Serial.available() > 0 && i < 10) {
    data[i] = Serial.read();
    i++;
  }
  i = 0;
  if(data[0] == 'A' && data[1] == 'L') digitalWrite(LED, HIGH);
  if(data[0] == 'E' && data[1] == 'T') digitalWrite(LED, LOW);
}
```

Image du code Arduino qui communique avec le serveur

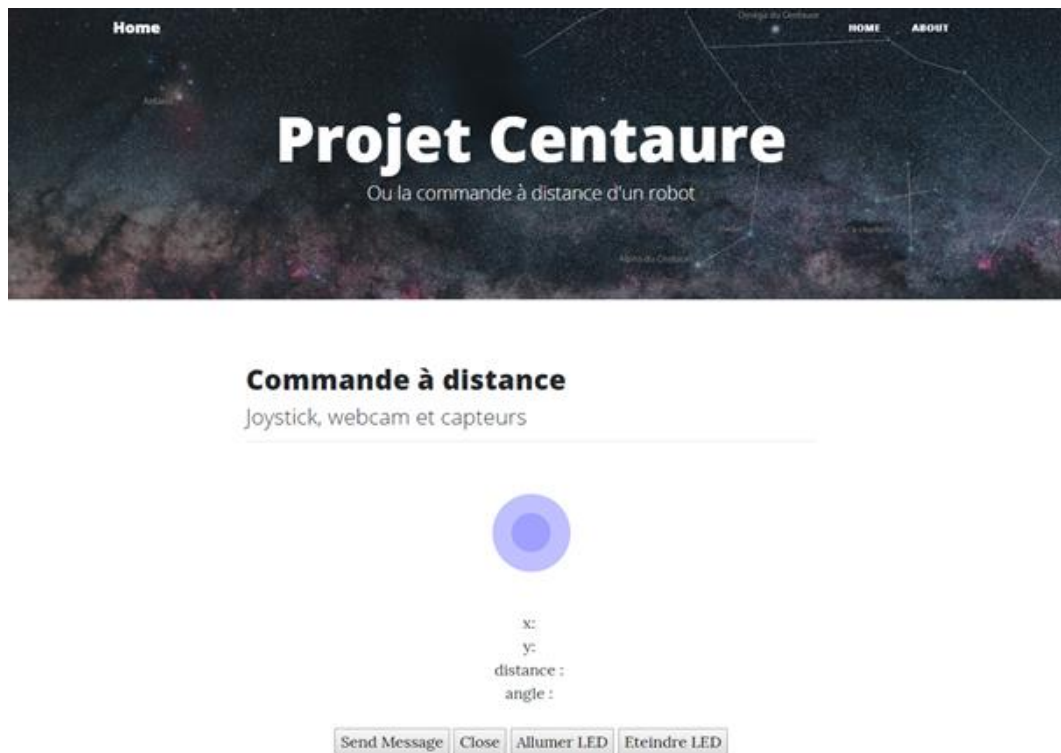


Image du site Web modifié

Malheureusement nous n'avons pas encore pu tester ce programme, car nous n'avons pas réussi pour le moment à lancer le serveur et nous sommes dans l'attente d'une réponse des anciens étudiants qui l'on mis en place.

B) Réorganisation du robot :

Comme vu précédemment, nous avons eu divers problèmes concernant la partie hardware du robot. En effet, le premier problème était celui de l'alimentation de l'écran. Nous avons donc dû commander un autre convertisseur 19V~36V/11V~16V qui permettait d'ajuster la tension de sortie afin de faire fonctionner l'écran. Car nous nous sommes rendu compte que l'écran était sous-alimenté, ce qui explique le manque de rétroéclairage qui faisait croire que l'écran était éteint.

Nous avons donc voulu alimenter l'écran, le pc et la kinect avec ce convertisseur, or, si nous augmentions la tension de sortie afin qu'elle atteigne la tension nécessaire pour que l'écran ne s'éteigne pas, les autres composants seraient suralimentés pour certains ce qui aurait été problématique, comme par exemple pour la Kinect ne supportant pas une tension supérieure à 12V. Nous avons donc dû réorganiser le câblage du robot et refaire un schéma de câblage.

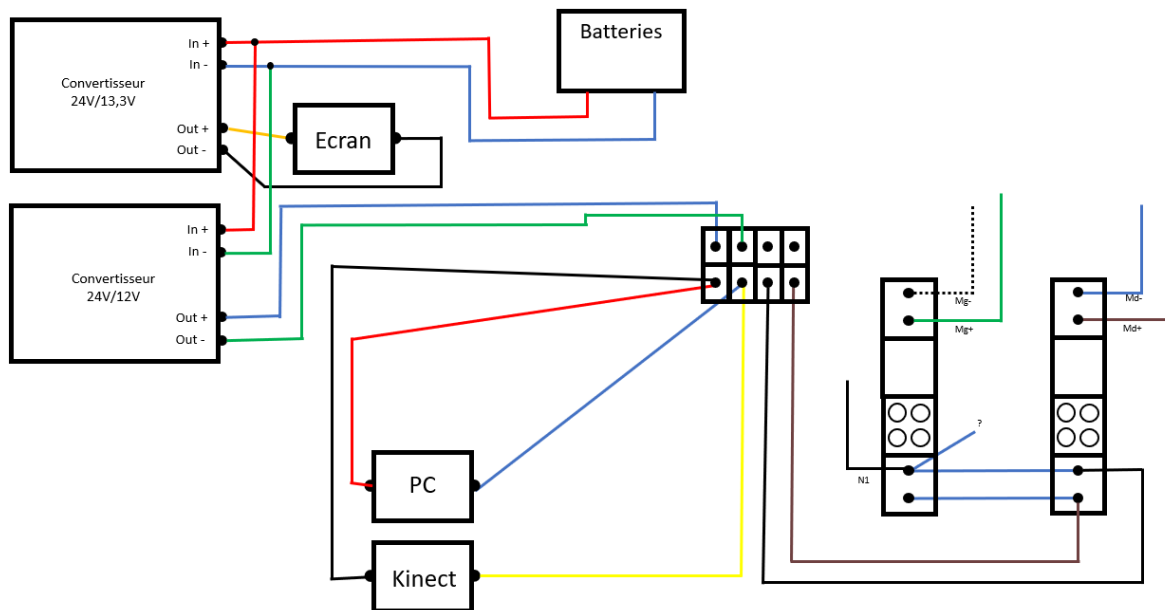
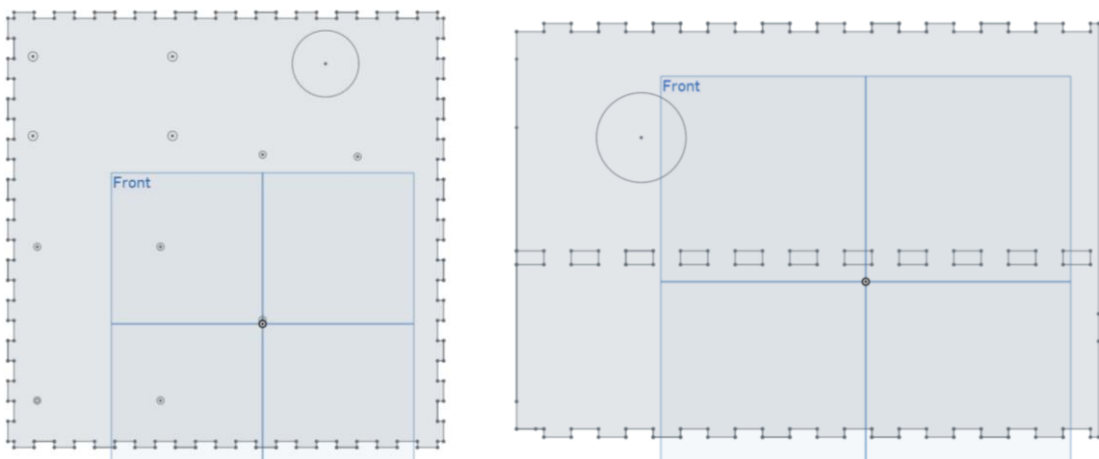


Schéma de câblage de la partie supérieure du robot

A noter que sur le schéma de câblage ci-dessus, la couleur des câbles ne se réfère pas aux couleurs utilisées habituellement en électricité comme par exemple le rouge pour une phase et noire ou bleu pour la masse. Un changement de câbles va être fait dans les prochains semestres afin de reconnaître plus facilement la fonction de chaque câble.

Pour contrer le problème d'alimentation de tous les composants, nous avons donc branché en parallèle les deux convertisseurs pour en avoir un avec une sortie à 13,3 V et le deuxième à 12V. Nous avons aussi essayé d'organiser l'intérieur du robot pour pouvoir manipuler les câbles sans contraintes particulières et espacer au maximum les différentes parties de l'intérieur du robot. Nous avons aussi conçu une boîte pour l'ordinateur et l'arduino pour avoir un rendu visuel plus propre.



Images de la modélisation de deux parties de la boîte sur le logiciel On shape

C) Software :

Nous avons, en parallèle de l'enveloppe physique de notre robot, commencé à "donner vie" à notre robot.

Ayant pour objectif d'accueillir et de guider les visiteurs au sein de Polytech, nous avons décidé de permettre à notre robot d'identifier les visages humains ainsi que de pouvoir communiquer oralement avec eux.

Reconnaissance faciale :

Nous parvenons actuellement à récupérer et analyser un flux vidéo issu d'une webcam. Nous utilisons pour cela des bibliothèques nommées `opencv_python` et `numpy`. `Open_cv` prend en charge l'acquisition vidéo. Après le lancement de la webcam, notre programme analyse chaque image successivement.

Pour analyser ces images, nous les convertissons en un ensemble de matrices à plusieurs dimensions (représentant les valeurs de Red, Green et Blue).

La bibliothèque `opencv` prend en charge ces matrices et nous permet de les manipuler assez simplement. Nous utilisons ensuite des fonctions et fichiers spéciaux (des `haarcascades`) issus de la méthode de Viola et Jones pour détecter d'éventuels visages dans le champ de la caméra. De la même façon, nous analysons chaque visage repéré dans le but d'y détecter ses yeux. En effet, la détection des yeux nous indique que la personne se tient face au robot et à une distance assez proche. Ces deux éléments nous indiquent que cette personne veut probablement interagir avec le robot.

Identification faciale :

Le module `OpenCV` comprend également des fonctions nous permettant "d'apprendre" des visages à notre robot.

Afin d'y parvenir nous implémentons un algorithme d'apprentissage de visage à partir de photos rangées dans une base de données nommée "faces". De plus, nous créons un algorithme d'entraînement pour permettre de tester et de peupler plus efficacement notre base de données.

Ainsi, lorsqu'un visage et ses deux yeux sont détectés par notre robot, celui-ci peut tenter d'identifier cette personne pour la saluer personnellement et d'engager le contact.

Reconnaissance vocale :

Afin d'effectuer notre reconnaissance vocale, nous utilisons la bibliothèque `speech_recognition`. Notre robot écoute ce que la personne dit. Tant que cette dernière parle, il l'écoute. Nous avons mis une limite d'écoute de 3sec dans le cas où la personne ne parle pas pendant 3 sec, le cas échéant notre robot le signale à la personne. Cette phrase est ensuite stockée dans une variable puis découpée mot par mot et stockée dans une liste.

Nous avons également mis en place un fichier texte qui lie un mot clé et un fichier .mp3 de la manière suivant : <mot clé> *<n° procédure><fichier .mp3>

Notre liste de mot est analysée mot par mot et comparée au mot clé dans une de nos fonctions dans notre programme principal. Le fichier .mp3 est ensuite joué et le numéro de la procédure est enregistrée.

Ce numéro de procédure est ensuite utilisé en fonction de la demande de l'utilisateur, si le robot vient de dire au revoir ou vient de faire une blague il n'attend rien d'autre alors que s'il vient de lui indiquer un lieu spécifique à l'utilisateur il peut afficher une page web à l'écran (tel qu'un plan de l'école).

Synthèse vocale :

Concernant notre synthèse vocale, nous utilisons la librairie mise à disposition par Google : gTTS (Google Text To Speech). Cette librairie comporte pour autant un désavantage, elle nécessite une connexion Internet.

Mais aussi la librairie PlaySound qui permet de jouer des sons sur un ordinateur (tel que des .mp3)

Nous utilisons principalement 2 programmes, l'un créant tous les fichiers .mp3 avec des phrases pré faites, tel que la phrase pour dire au revoir, dire à l'utilisateur qu'il ne l'a pas reconnu ou encore la phrase indiquant à l'utilisateur qu'il n'a pas compris sa requête. Le deuxième fichier crée des fichiers .mp3 nominatifs afin de dire bonjour à la personne qu'il reconnaît grâce à la reconnaissance faciale. Pour ce faire il faut que la bibliothèque de visage soit préalablement créée, comme expliqué dans la partie reconnaissance faciale. Pour chaque personne, un fichier .mp3 sera créé.

Ces deux programmes ne sont lancés qu'une seule fois avant le lancement du robot. Si une personne est ajoutée à la base de données de visage il suffit simplement de relancer la fonction qui crée l'ensemble des .mp3 nominatifs.

Conclusion :

Ainsi, nous avons pu réaliser différentes tâches sur le robot. En effet, nous pouvons constater que nous avons, pour le moment, réussi à faire un système de synthèse vocale et un système de reconnaissance faciale ainsi que de reconnaissance vocale. Nous avons aussi corrigé certains problèmes tel que la sous-alimentation de l'écran du robot ou encore l'optimisation de la place à l'intérieur du robot. Enfin, nous avons amélioré le PCB et conçu une boîte afin de ranger le PC dans le robot, toujours dans une optique d'optimiser l'espace dans le robot. Cependant, il nous reste beaucoup de tâches à faire si nous voulons atteindre nos objectifs pour l'année prochaine. En effet, nous devons implémenter la partie software sur le PC du robot, gérer la reconnaissance de QR Code par le robot, le déplacement autonome du robot, en utilisant les capteurs mis à disposition et la Kinect pour éviter les différents

obstacles pouvant se dresser sur son chemin et enfin, la création d'un système d'itinéraire afin d'optimiser le trajet du robot pour atteindre une salle.