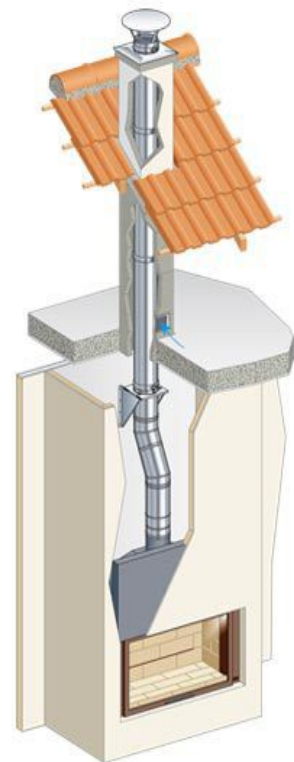
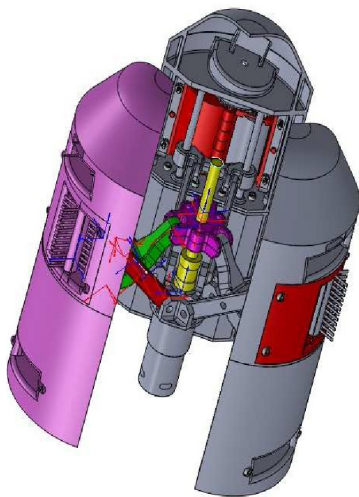


## Projet de Baccalauréat STI2D-SIN :

### Robot RAMON

Équipier N°1 : François MARI

Adaptation, Déplacement et Nettoyage du conduit



# Sommaire

## Introduction :

- Présentation Générale du Projet

- Énoncé du Besoin

## I- Analyse Fonctionnelle

## II- Conception Préliminaire & Exploration des Solutions

- 2.1- Le mécanisme de dépliement/repliement.

- 2.2- Mesurer la force de contact.

- 2.3- Envoi des ordres de déploiement/repliement.

## III- Tests, Validation & Prototypage

- 3.1- Déploiement et Repliement.

- 3.2- Mesure du courant

## IV- Finalisation & Validation du Fonctionnement

# Introduction :

## -Présentation du projet :

De plus en plus en France, les foyers se munissent d'inserts, de poêles, et autres cheminées ; entre 1999 et 2012 le nombre d'appareils en circulation est passé de 5,9 à 7,4 millions et certaines estimations prédisent que cette augmentation va rester constante jusqu'à atteindre d'ici 2020 un total d'environ 9 millions de ménages équipés par ce moyen de chauffage qui est le plus économique du marché.

Cependant toutes ces installations représentent un danger potentiel pour ses utilisateurs du fait des dépôts de suie dans les conduits qui en s'accumulant risquent de provoquer des intoxications au monoxyde de carbone ainsi que des incendies involontaires c'est pourquoi la loi contraint les usagers à effectuer un ramonage de leurs conduits une fois par an pour ceux utilisant un chauffage au gaz et deux fois par an pour ceux utilisant le charbon, le bois ou le fuel.

Jusqu'à ce jour le ramonage des conduits était effectué par des professionnels soit manuellement grâce à des moyens mécaniques comme le hérisson soit par des procédés chimiques grâce à des bûches ou des sachets de ramonage qui sont moins efficaces et pas toujours reconnus par les assurances en cas de sinistre.



## -Énoncé du besoin :

C'est dans une optique d'amélioration de ces procédés de nettoyage que nous avons basé notre réflexion et orienté notre processus de recherches afin de concevoir un système automatisé qui se présentera sous la forme d'un robot capable de se déployer dans le conduit, de se déplacer dans celui-ci selon deux modes (automatique ou manuel) choisis par l'opérateur au préalable ; en mode automatique le robot s'adapte au conduit, démarre son cycle de nettoyage et revient dans sa position initiale une fois le ramonage effectué. En mode manuel l'opérateur effectue toutes les opérations (Déploiement, Montée, Descente, Activation des brosses, Repliement) à l'aide des différents boutons présents sur un pupitre communiquant à distance avec le robot.

De plus, dans une optique de visibilité par rapport au client et de respects des lois en vigueur sur le ramonage l'opérateur doit pouvoir fournir une preuve de l'état de propreté du conduit c'est pourquoi nous avons également muni notre appareil d'une lampe et d'une mini caméra sans fil.

Ce projet s'inscrit dans une démarche d'éco-conception et de partage des compétences avec l'étroite collaboration entre les deux groupes de travail SIN et ITEC tout au long de l'année.

# I- Analyse Fonctionnelle

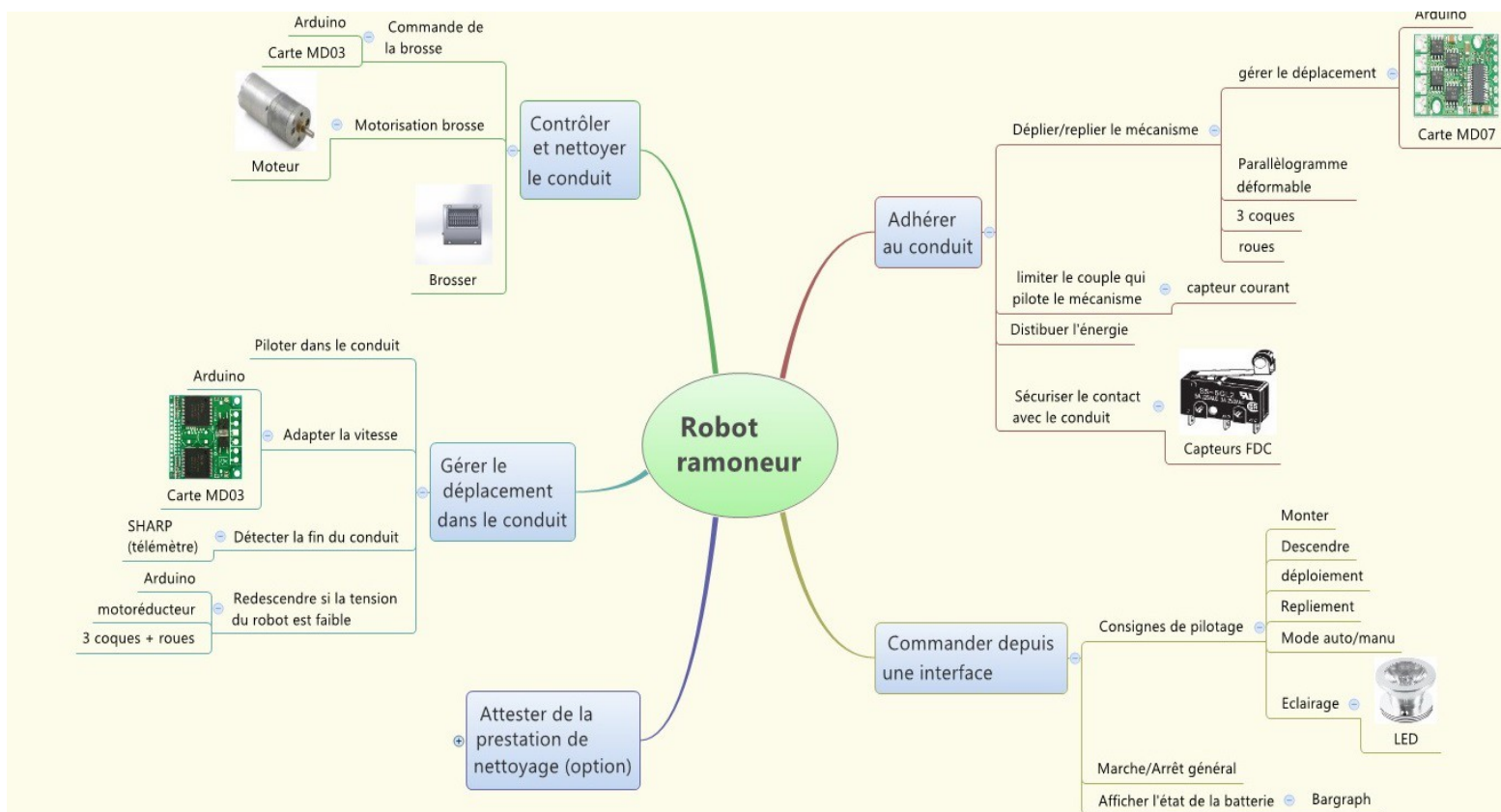
Avant de débiter la conception du projet à proprement parler nous sommes passé par une phase d'étude et d'appropriation du cahier des charges pour bien nous imprégner de l'ampleur du travail à réaliser.

J'ai donc établis un tableau regroupant les différentes phases du projet et y ai détaillé les tâches à effectuer.

( Voir Annexe 1 )

Cette appropriation des taches passe par l'analyse fonctionnelle du projet qui regroupe plusieurs diagrammes ayant tous un rôle bien spécifique.

Tout d'abord la Carte Mentale résultante d'un brainstorming fait pour connaître tous les éléments à prendre en compte lors de la réalisation du projet.



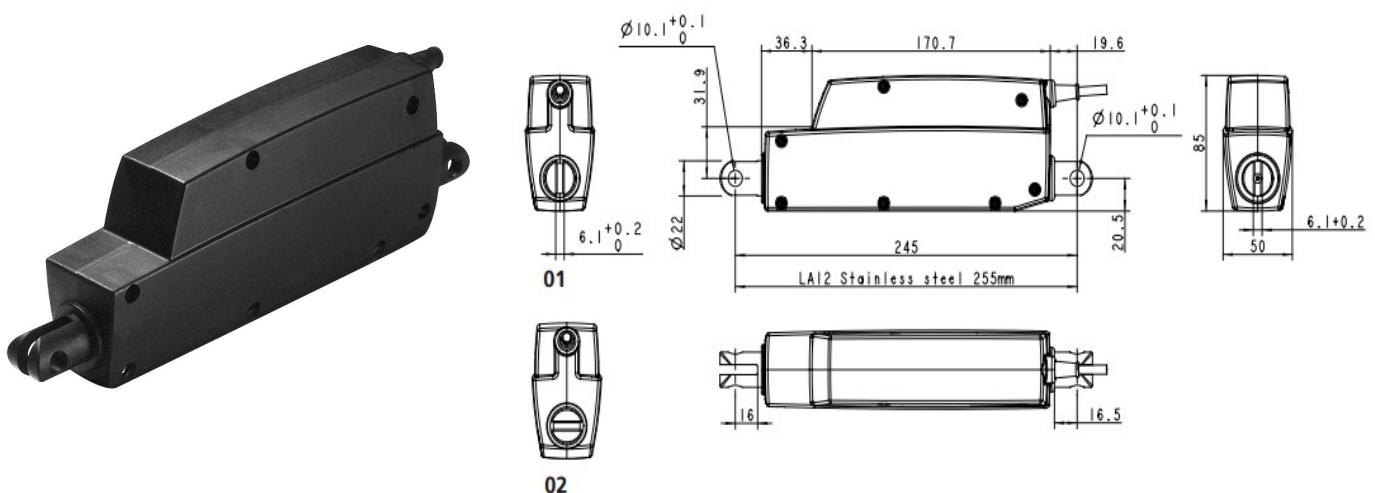
Le reste de l'analyse fonctionnelle complète et détaillée est disponible dans le dossier de groupe.

## II- Conception Préliminaire & Exploration des Solutions

### 2.1- Le mécanisme de dépliement/repliement.

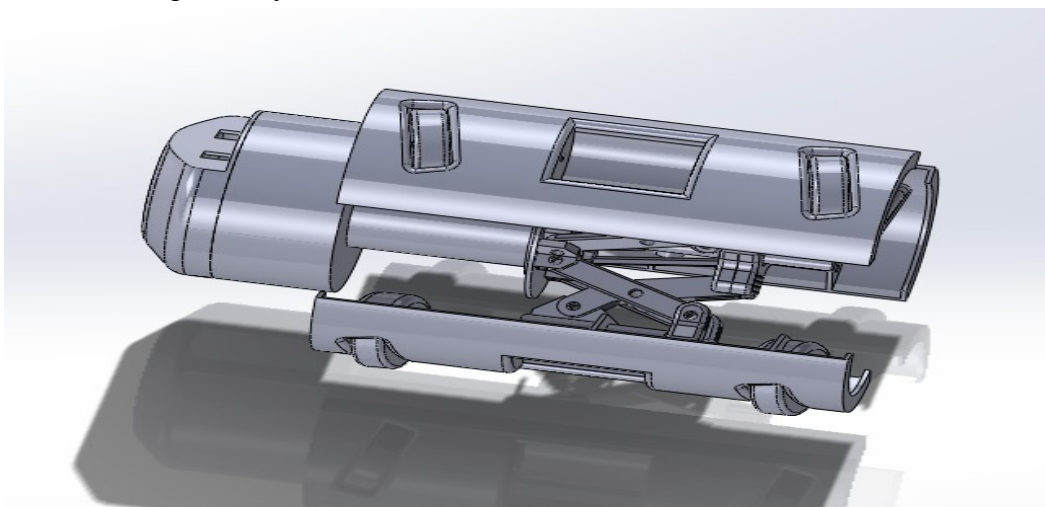
Pour faire fonctionner le mécanisme de déploiement et de repliement du robot j'ai réfléchi à plusieurs solutions qui aurait pu remplir cette fonction.

En premier lieu, j'avais songé à utiliser un vérin linéaire électrique alimenté en 6 Volts de la marque LINAK ; lors de mes recherches, j'ai sélectionné le modèle LA12



Ce modèle permet de respecter le cahier des charges aussi bien pour la longueur de déploiement du robot que pour la force de contact sur le conduit.

Cependant la collaboration avec les ITEC a révélé que cette option ne pouvait être viable car le design de la coque ne pouvait pas accueillir une telle technologie. En effet, la structure en trois parties est activée par un système de vis-écrou.

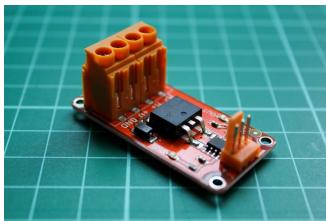


C'est pourquoi il m'a fallu trouver un moteur capable de délivrer assez de puissance pour faire fonctionner l'ensemble et exercer une force suffisante sur les parois du conduit pour empêcher le robot de tomber.

Je me suis donc penché donc penché sur le moteur 940D Series 32mm de chez MFA.



Les calculs ont montrés que pour accomplir ses tâches, le moteur devait avoir un couple minimum de 0,16 Nm or ce moteur avec un rapport de réduction de 100:1 délivre un couple de 0,46Nm avec une alimentation 12V.



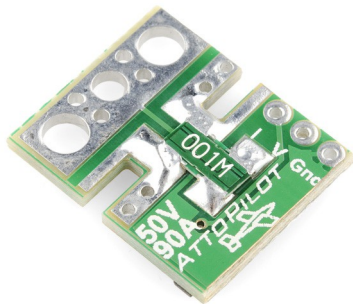
Pour pouvoir contrôler ce moteur, j'ai choisis un module MOSFET Tinkerkit.

Ce module qui se branche sur un Sensor Shield Tinkerkit permet de contrôler aisément un moteur fonctionnant en 24 Volts maximum.

## 2.2- Mesurer la force de contact.

Pour mesurer la force de contact exercée par le robot sur le conduit j'ai choisis la solution du capteur de courant piloté grâce à une carte Arduino UNO

La carte SEN-09028 de chez SparkFun utilisée en 90 Ampères permet d'avoir une précision suffisante pour mesurer efficacement le courant qui passe dans le moteur.



La mesure de tension se fait par un convertisseur Analogique/Numérique avec une plage de précision de 3,3 Volts. Elle est aussi auto-alimentée.

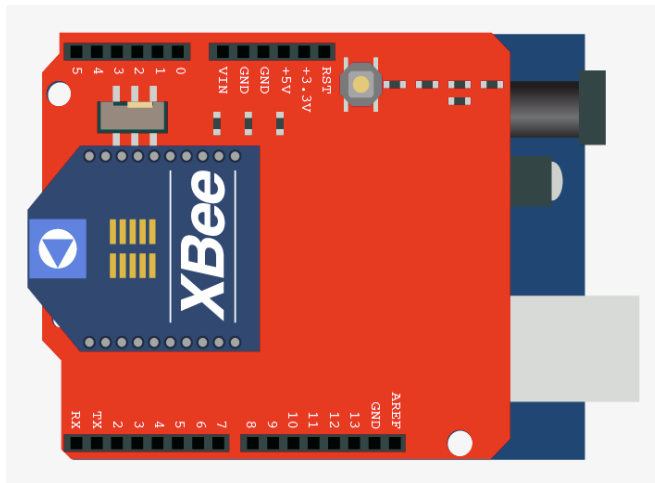
De plus sa taille très réduite (4 x 15 x 19mm) convient parfaitement pour notre robot qui ne dispose que de peu de place pour l'électronique.

## 2.3- Envoi des ordres de déploiement/repliement.

Le robot étant commandé par le biais d'un pupitre externe communiquant sans liaison filaire avec le robot il nous a fallut réfléchir à un dispositif pour envoyer les ordres émanant du pupitre sus-mentionné.

Pour se faire nous avons choisis la solution du module X-Bee que nous avons déjà étudiés au cours des deux dernières années le protocole ZigBee étant bien adapté au projet qui ne nécessite ni une grande portée ni une rapidité de transmission excessive et dont la consommation électrique est relativement faible.

Les modules X-Bee fonctionnent par paires ; nous en avons placé un dans le robot et un dans le pupitre, tous deux fichés sur un shield Arduino comme suit.



La communication entre les deux cartes se fait par le biais d'un programme réalisé par un autre équipier.



# III- Tests, Validation & Prototypage

## 3.1- Déploiement et Repliage.

Pour déployer le robot dans le conduit et le replier à la fin du cycle de ramonage il m'a fallu réaliser un programme de test à l'aide d'une carte de prototypage Arduino Uno et d'un module MOSFET.

Le programme est celui présenté en annexe 2.

Il est composé de plusieurs parties :

Tout d'abord les parties inhérentes à tous les programmes sous Arduino c'est à dire :

- L'Affectation des Entrées et Sorties.
- La Déclaration des Variables Globales.

Ensuite dans le cœur du programme j'ai utilisé la fonction Switch Case pour définir les ordres à envoyer. Par exemple, lorsque dans le moniteur série du logiciel Arduino je tape la consigne "D", le moteur tourne dans le sens de déploiement pendant 5 secondes puis se stoppe ; pour la consigne "R" le moteur tourne dans le sens de repliement pendant 5 secondes puis se stoppe et enfin, la consigne "A" le moteur se stoppe uniquement.

Exemples :

```
switch(consigne) J'ai aussi créé les boucles correspondantes pour pouvoir appliquer les actions choisies
{
case 'A':          void deployer()                                void replier()
arreter();|        {
attendre();        STANDBY=1;
break;            SENS=1;
                  VITESSE=255;

case 'D':          digitalWrite(CDE_RST,STANDBY);                digitalWrite(CDE_RST,STANDBY);
deployer();        digitalWrite(CDE_DIR,SENS);                    digitalWrite(CDE_DIR,SENS);
attendre();        analogWrite(CDE_PWM,VITESSE);                analogWrite(CDE_PWM,VITESSE);
arreter();        }
attendre();        }
break;

case 'R':          void attendre()                                void arreter()
replier();        {
attendre();        delay (5000);
arreter();        }
attendre();        }
break;
```



### 3.2- Mesure du courant.

Pour tester la mesure de courant j'ai fait un programme qui mesure la moyenne du courant, la moyenne  $I_{ref}$ , la moyenne  $I_{mes}$  calcule l'écart de courant et les affiche sur le moniteur série.

Voir Annexe n°3.

## IV- Finalisation & Validation du Fonctionnement.

Pour terminer, nous avons monté le robot intégré les cartes dans le pupitre et validé le fonctionnement de l'ensemble.

Les schémas de câblage ainsi que les tableaux des E/S ont été dressés :

Voir annexes 4, 5 et 6.



Je remercie les professeurs pour l'aide et le soutien qu'ils nous ont apporté toute l'année ainsi que le lycée pour le matériel et la logistique.