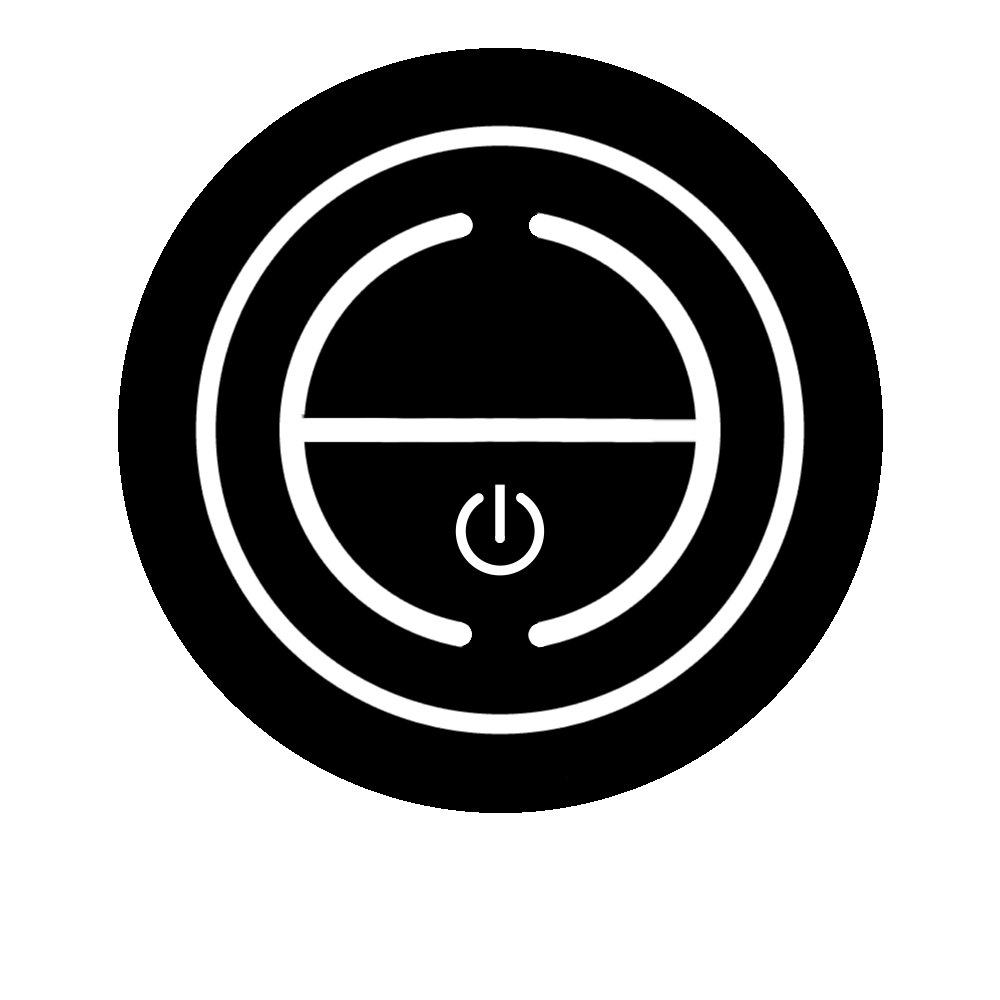
OPEN HOOVER



**Cahier Des Charges**

**Sommaire**

[1. Introduction 3](#_Toc1673882)

[2. Description du document 4](#_Toc1673883)

[a. Info 4](#_Toc1673884)

[b. Révisions 4](#_Toc1673885)

[3. Description du projet 5](#_Toc1673886)

[a. Contexte 5](#_Toc1673887)

[b. Objectifs 6](#_Toc1673888)

[c. Contraintes Fonctionnelles 7](#_Toc1673889)

[d. Design 9](#_Toc1673890)

[4. Réalisation du projet 10](#_Toc1673891)

[a. Description fonctionnelle des besoins 10](#_Toc1673892)

[b. Technologies utilisées 17](#_Toc1673893)

[c. Charge de travail 18](#_Toc1673894)

[d. Disponibilité / Planning 19](#_Toc1673895)

[e. Méthodologie 20](#_Toc1673896)

[5. Contact 21](#_Toc1673897)

1. Description générale du projet

OpenHoover est une solution complète permettant à chacun de construire son propre aspirateur autonome, aussi également appelé [Robovac](https://en.wikipedia.org/wiki/Robotic_vacuum_cleaner).

L’intégralité des plans de construction, de la structure mais également les composants nécessaires à sa fabrication, qu’il soit électronique ou mécanique, et aussi son algorithme de fonctionnement sont en libre disposition et utilisable par tous.

Les étapes de fabrication et mise en fonctionnement de l’aspirateur sont simples, détaillés et accessibles à tous, qu’il soit néophyte ou féru d’électronique/informatique.

L’aspirateur possède la majorité des fonctionnalités qu’ont les aspirateurs autonomes commercialisés de nos jours, il est capable de se repérer dans l’espace, et de réaliser un trajet optimal pour nettoyer au mieux l’environnement dans lequel il se trouve.

Il est également peu couteux, la structure principale de l’aspirateur peut être fabriqué à l’aide d’une imprimante 3D ou bien à partir de matériaux de récupération. Pour ce qui est de l’électronique, les composants utilisés sont tous de type Arduino et très commun et donc bon marché.

1. Description du document
   1. Info

|  |  |
| --- | --- |
| Type | Cahier des charges |
| Date | 17/02/2019 |
| Auteur | Loic Robin |
| Projet | Open Hoover |
| Version | 1.0 |
| Pages | 21 |

* 1. Révisions

|  |  |  |
| --- | --- | --- |
| **N° Version** | **Date** | **Description et Commentaires** |
| **V 1.0** | 17/02/2019 | Premier prototype fonctionnel |

1. Description du projet
   1. Contexte

De nos jours, de nombreux nouveaux objets domotiques sont présentés fréquemment.

Ils ont pour objectif de simplifier notre quotidien et de nous assister dans nos tâches ménagères, ou même parfois de les effectuer à notre place.

Un des objets phare de la domotique et des plus démocratisés actuellement est l’aspirateur automatique autrement appelé Robovac.

Ces aspirateurs permettent sur commande d’effectuer le nettoyage du sol de nos surfaces de vie. Ils nous libèrent ainsi de cette contrainte/tâche récurrente et nous offre ainsi plus de temps libre.

Malheureusement le problème avec tous ces objets domotique, c’est qu’ils sont souvent peu abordables, et d’une durée de vie plutôt courte.

En effet il est comme pour la plupart des appareils électroménager difficilement réparable, les pièces sont chères, et pas forcément facile à trouver.

De plus le fonctionnement interne de ces appareils nous est inconnu et nous n’avons donc par forcément la capacité d’effectuer cette tâche par nous-même.

Nous revenons donc à ce problème de coût et d’entretien que ces appareils nous posent.

Un autre problème est la récolte/analyse de données, les aspirateurs mesurent de plus en plus précisément nos intérieurs et sont donc capable d’établir un plan complet de nos habitats.

Certains aspirateurs possèdent même une/des caméra(s), mettant en jeu notre vie privée, quand on ne sait absolument pas ce que les fabricants font de ces données.

* 1. Objectifs

L’objectif principal du projet OpenHoover est de permettre à tous d’accéder à ce type de nouvelle technologie domotique, mais également d’en avoir le contrôle total pour un faible coût via un manuel simple et détaillé.

Ce manuel de fabrication OpenHoover devra donc permettre la construction et l’entretien d’un appareil Robovac de façon accessible en mettant à disposition toutes les informations nécessaires à sa réalisation de manière claire et intelligible.

Les pièces qui seront sélectionnés pour la fabrication du OpenHoover devront être de bonnes qualités et facilement disponible à l’achat pour un bas coût.

L’utilisation de technologie ouverte comme l’Arduino permettra donc une réduction des coûts de fabrication énorme et une réparabilité tout aussi importante.

De plus l’accès libre à l’intégralité du code des composants et du logiciel de contrôle de l’aspirateur permettra une vérification constante par tous les utilisateurs, augmentant considérablement le contrôle qualité et fonctionnel du OpenHoover.

Il devra être quasiment aussi performant que les autres Robovac disponible dans le commerce à ses débuts et par la suite, à l’aide des retours utilisateurs, égaler, ou même dépasser les performances de la « concurrence ».

* 1. Contraintes Fonctionnelles

Afin de rendre le système le plus adapté possible aux besoins de l’utilisateur j’ai cerné différentes contraintes fonctionnelles qui devront être respectés au sein du projet.

**Construction Simple**

Les plans fournis doivent être le plus simplifier possible pour permettre à n’importe qui de facilement construire son propre OpenHoover.

**Mise en fonctionnement Rapide**

Après sa construction le OpenHoover devra pouvoir être facilement mis en fonction par l’utilisateur.

**Système Antiblocage**

Pour éviter l’intervention de l’utilisateur, l’aspirateur devra être capable par lui-même de se sortir d’un endroit s’il s’y retrouve bloqué. Le cas échéant il devra notifier clairement à l’utilisateur qu’il est bloqué.

**Système d’évitement**

L’appareil devra prévoir à l’avance la collision avec un obstacle pour ne pas arriver à pleine vitesse dans ceux-ci. Prévenant ainsi les damages que ce soit de l’aspirateur ou de l’obstacle concerné.

**Trajet Optimisé**

L’aspirateur saura calculer/optimiser son trajet pour nettoyer au mieux l’environnement qui l’entoure et également en évitant de repasser trop fréquemment aux mêmes endroits.

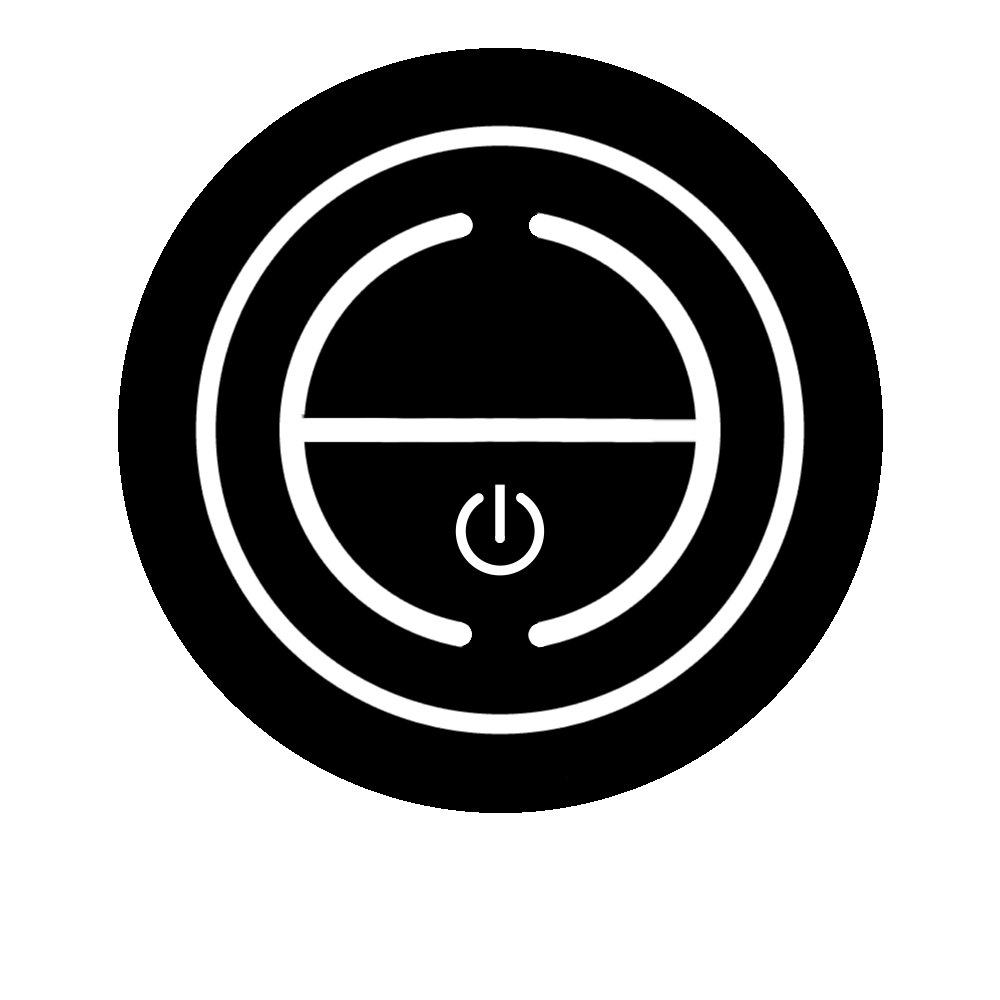
**Contrôle à distance**

Afin de faciliter l’interaction avec l’aspirateur, il devra être capable de communiquer avec un boitier distant qui devra également être bon marché. Ce boitier devra permettre des fonctionnalités minimales, tel la mise en route/arrêt de l’aspirateur et également un contrôle manuel via un simple joystick/pad donnant la possibilité à l’utilisateur de contrôler l’aspirateur s’il le souhaite.

* 1. Design

Le projet ne possède pas de chartre graphique particulière, étant principalement une solution algorithmique et matériel, seul un logo l’accompagne pour lui permettre d’être facilement identifiable.

Le Logo :



Ce logo contient les lettres O et H du nom du projet OpenHoover et représente l’aspirateur en lui-même, de forme circulaire et possédant un bouton de marche/arrêt.

1. Réalisation du projet
   1. Description fonctionnelle des besoins

|  |  |  |
| --- | --- | --- |
| Besoin | Fonctionnalité | Estimation temps de travail |
| Locomotion de l’aspirateur et « scanner » d’environnement, ainsi que communication avec son boitier de contrôle | Choix de composants adaptés aux différentes fonctionnalités. | 6 jours/homme |

Le choix des composants est une étape très importante dans la réalisation du OpenHoover. En effet ceux-ci, devront être performant pour une faible consommation de ressources (Calcul, et donc énergétique) mais également peu cher à l’achat.

Ils devront permettre d’établir une solution répondant aux problématiques exposés, soit brièvement, être capable de se mouvoir, de scanner/mapper son environnement et également de communiquer avec un boitier distant.

Les composants devront être comparer sur leurs différents rapports performances/coût, et la solution finale ne devrait pas excéder un prix total d’environ 60€ maximum.

Le choix des composants s’effectuera en 2 étapes :

* Choix des composants minimum

(Quel Senseur ? Quel Moteur(s) ? Quel carte Arduino ?)

~= 3/4 jours/homme

* Comparaison des différents composants sélectionnés

(Consommation électrique, Complexité d’utilisation, Fiabilité, Coût)

~= 2/3 jours/homme

|  |  |  |
| --- | --- | --- |
| Besoin | Fonctionnalité | Estimation temps de travail |
| Structure pour l’aspirateur | Plan détaillé de construction de la structure en 3D | 7 jours/homme |

La réalisation de la structure de l’aspirateur s’effectuera à l’aide d’un logiciel de conception 3D permettant l’impression 3D du modèle.

La structure devra prendre en compte les différentes nécessités fonctionnelles de l’aspirateur. Celle-ci sera de forme circulaire, lui évitant ainsi de se bloquer dans son environnement.

Elle devra être prête à recevoir les composants nécessaires à son fonctionnement ayant été sélectionné précédemment, tel que les moteurs, senseurs, batterie, et autres.

La structure sera réalisée selon les étapes suivantes :

* Conception/Esquisse (dessin, réflexion)

~= 2/3 jours/homme

* Réalisation du modèle 3D via un logiciel de conception

~= 1/2 jours/homme

* Prototypage (En matériaux de récupération)

~= 2/3 jours/homme

|  |  |  |
| --- | --- | --- |
| Besoin | Fonctionnalité | Estimation temps de travail |
| Boitier de contrôle | Création d’un protocole de communication permettant l’échange entre le boitier de contrôle et l’aspirateur | 5 jours/homme |

Le boitier de contrôle devra être capable de communiquer avec l’aspirateur selon un protocole défini, ce boitier est essentiel car il permet aussi le débogage de l’aspirateur.

En effet, étant ici sur un système embarqué possédant très peu de RAM et de ROM, il n’est pas possible de garder ou même de gérer trop de Log, ceux-ci doivent donc être transmis à un boitier distant.

De plus ce boitier permettra la mise en route/arrêt de l’aspirateur, mais également son contrôle, il alloue aussi une plus grande amplitude d’améliorations, ce boitier pouvant par la suite devenir la station d’accueil de l’aspirateur lors de future amélioration.

Il sera mis en œuvre de la manière suivante :

* Idéalisation et création du protocole de communication Boiter/Aspirateur

~= 2/3 jours/homme

* Implémentation du protocole et Test avancé

~= 2/3 jours/homme

|  |  |  |
| --- | --- | --- |
| Besoin | Fonctionnalité | Estimation temps de travail |
| Produit Fini | Assemblage des différentes composantes | 1 jour/homme |

Toutes ces étapes réalisées, la structure et les composants devront être assemblé, et le concepteur devra vérifier l’intégrité de la structure et de son ensemble pour s’assurer qu’aucun composant ne pose problème.

De même pour le boitier de contrôle.

|  |  |  |
| --- | --- | --- |
| Besoin | Fonctionnalité | Estimation temps de travail |
| Déplacement | Algorithme de calibrage des vitesses de déplacements de l’aspirateur (Angulaire et Linéaire) et mise en place de l’interface de contrôle des mouvements | 4 jours/homme |

L’aspirateur étant assemblé par l’utilisateur, il peut comporter de légères variantes d’un assemblage à l’autre, de plus les composants utilisés peuvent aussi varier légèrement, provoquant ainsi un déplacement différent d’un aspirateur à l’autre.

De plus le niveau de charge restant des batteries peu lui aussi faire varier les performances du robot.

Il est donc essentiel qu’il possède des algorithmes lui permettant de se (re)-calibrer en fonction de ces différents paramètres. Il utilisera les senseurs à disposition pour définir sa vitesse angulaire ainsi que linéaire.

|  |  |  |
| --- | --- | --- |
| Besoin | Fonctionnalité | Estimation temps de travail |
| Mémoire de l’environnement | Algorithme de mapping de l’environnement et optimisation de la donnée de structure pour un stockage optimale et léger de ces informations | 9 jours/homme |

Les systèmes embarqués n’ayant que très peu de puissance de calcul et peu de stockage. Les données récoltées par l’aspirateur devront donc être stocker de sorte à utiliser le moins d’espace possible tout en conservant le maximum d’information sur son environnement.

La réalisation de l’algorithme de mapping d’environnement devra donc être réaliser selon la procédure suivante :

* Choix du type de donnée en fonction des senseurs (Image, Distance, …) de caractérisation de celle-ci.

~= 2/3 jours/homme

* Choix d’un type de structure de donnée optimisé (Arbre, Liste, …)

~= 2/3 jours/homme

* Mise en œuvre de l’algorithme et test en simulation ainsi que test de sauvegarde/chargement en condition réel sur l’aspirateur

~= 3/4 jours/homme

|  |  |  |
| --- | --- | --- |
| Besoin | Fonctionnalité | Estimation temps de travail |
| Navigation | Algorithme de recherche de trajet dans l’environnement mappé précédemment | 7 jours/homme |

Une des fonctions les plus important de l’aspirateur est la navigation, celui-ci doit pouvoir calculer un trajet optimal en fonction de son emplacement.

Pour pouvoir nettoyer efficacement son environnement il doit ainsi couvrir un maximum d’endroit tout en évitant de passer plusieurs fois au même endroit.

Lors de sa mise en œuvre l’algorithme devrait suivre ces étapes :

* Réalisation d’un Algorithme de « Path Finding » global, pour le trajet principal de l’aspirateur couvrant la plus grande zone possible

~= 3/4 jours/homme

* Test et gestion des cas particuliers tel que les plaintes, meubles, obstacles, ou même le cas d’objets externe mobiles dans l’environnement.

~= 3/4 jours/homme

* 1. Technologies utilisées

**C/C++ :**

Utilisation avancée des langages C et C++, optimisation de la mémoire et de la répartition des tâches sur un environnement mono-cœur.

Utilisé pour les algorithmes de calibrage, de localisation, mapping d’environnement et optimisation de trajet ainsi que correction d’erreur, mais également pour le protocole de communication Boitier/Aspirateur.

**Arduino :**

Exploitation de Librairies multiples Open-Source pour les différents modules/composants utilisés, tels que les moteurs, et différents capteurs/senseurs.

**IDE :**

* **Visual Studio 2017**
* **Arduino IDE**
  1. Charge de travail

Comme décrit précédemment ce projet représente une charge de travail considérable, tant sur la partie ingénierie de la conception de la structure de l’aspirateur et de ses composants, que sur les différentes solutions algorithmiques permettant son bon fonctionnement.

En effet le choix de l’Arduino ayant une puissance de calcul très réduites, pour la réduction des coûts de fabrication de l’aspirateur, implique donc la nécessité d’algorithme extrêmement optimisé.

Tout cet environnement devra donc être complétement maitrisé pour permettre le meilleur de ce qu’il autorise.

La totalité du projet comptabilise environ une 40ène de jour/homme.

* 1. Disponibilité / Planning

|  |  |  |
| --- | --- | --- |
| **Développeur(s)** | **Disponibilité** | **Commentaire** |
| **Loic Robin** | Temps plein  =~50-55h/semaine | Etant actuellement disponible à temps plein et tenant ce projet très à cœur, j’y consacrerais la quasi-intégralité de mon temps libre.  Je tacherais ainsi de respecter au mieux les différents délais pour les tâches qui me sont assignés conformément à ce document. |

En conséquent, et aux vues des disponibilités le projet devrait pouvoir être réalisé au terme d’environ 1 mois ½ ou tout au plus 2 mois.

* 1. Méthodologie

Pour ce projet, les différentes tâches devront être réalisé dans l’ordre énuméré précédemment dans le tableau des fonctionnalités.

On parle donc ici de méthodologie en cascade.

Chaque tâche nécessitant la réalisation de la précédente, tout retard dans la réalisation d’une de celle-ci provoquera le retard de l’intégralité du projet. Les délais devront donc ainsi être aux mieux respectés.

1. Contact

Pour tout renseignements supplémentaires ou autres informations concernant ce projet, veuillez nous contacter via les dispositifs suivants :

Email : [loic.robin@epitech.eu](mailto:loic.robin@epitech.eu) / loic.robin@hotmail.com