

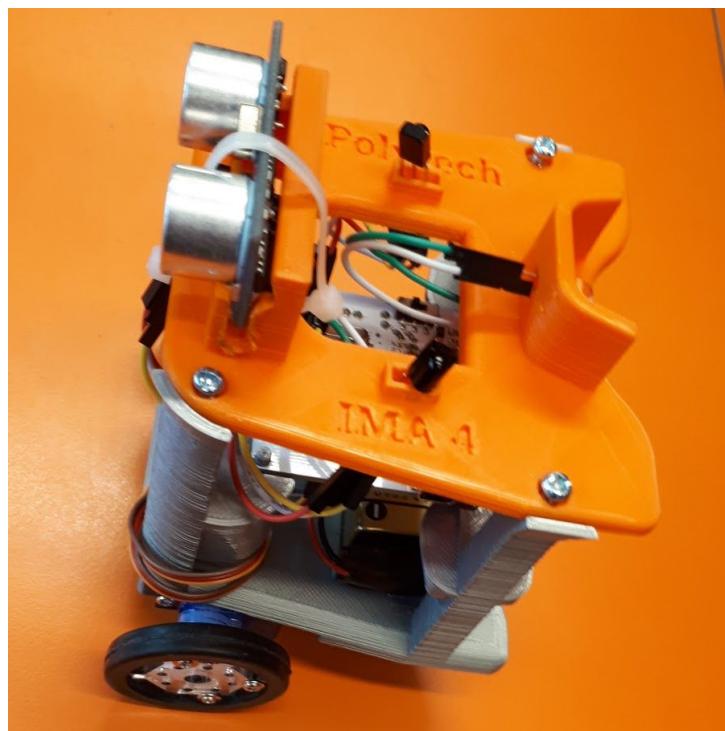
Canu Benjamin

Etcheguibel Ganix

IMA4 - 2016-2017

Rapport de Projet : P25

Essaim de Robot



Responsables :

M. Redon Xavier

M. Boé Alexandre

Remerciements

Nous tenons à remercier différentes personnes qui, par des conseils ou différents moyens matériels, nous ont permis d'avancer sur ce projet :

-M. Xavier Redon pour son suivi en tant que tuteur de projet, nous ayant apporté son aide et ses conseils tout au long du projet.

-M. Thierry Flament qui nous a permis d'avancer en toute sérénité sur le soudage des cartes électroniques au travers du matériel mis à disposition et de son expertise.

-M. Rodolphe Astori ainsi que le Fabricarium de Polytech Lille pour les différents conseils et cours en modélisation 3D et sur l'utilisation des machines, ainsi que pour le matériel mis à disposition.

-M. Fabrice Taingland pour le matériel mis à disposition lors de la finalisation du projet.

Sommaire

Remerciements	1
Introduction	3
Etude préliminaire :	4
Le principe d'essaim	4
L'étude des projets existants	4
Le développement du Robot :	5
Carte Électronique	5
Une initiation épaulée :	5
Une adaptation fonctionnelle :	5
Le soudage :	6
Microprocesseur	7
Composants	7
Les actionneurs :	7
Les capteurs et la détection de l'environnement :	8
Infrarouge	9
Une fréquence d'émission aléatoire	9
De la corruption des messages	9
Un message modulé	9
Déplacements	10
Modélisation 3D	10
Un modèle basé sur la fonctionnalité :	10
Les limites de nos choix	11
Un microprocesseur sous-dimensionné :	11
Un système d'exploitation pas si "mini" que ça :	12
Une détection peu précise :	12
Aller plus loin	13
Une avancée à courts termes	13
La programmation d'un ordonnanceur personnalisé :	13
Une avancée à moyens et longs termes	13
Une interaction avec l'utilisateur :	13
Un test à grande échelle :	14
Un algorithme développé :	14
L'ajout de mémoire externe :	14
Conclusion	15

Introduction

Ce projet s'inscrivant dans la formation de 4eme année de la filière Informatique, Microélectronique et Automatique (I.M.A.), nous avions pour but de réaliser de mini-robots pouvant, une fois en groupe, s'apparenter à un essaim (“swarm”).

Afin mener à bien cette mission, nous avons d'abord étudier le principe d'essaim et ce que cela impliquait, ensuite nous avons réfléchis aux besoins et contraintes des robots que nous devrons produire avant de les réaliser.

La réalisation s'est déroulée en plusieurs parties :

D'abord nous devions produire la carte électronique et définir les différents composants nécessaires à sa constitution. Nous devions aussi organiser la programmation au travers du microcontrôleur.

Le robot ne pouvant exister sans une constitution physique, nous avons aussi décidé de réaliser un châssis.

Enfin venait le montage global et la programmation du contrôle et de l'algorithme des robots afin de leur permettre de bouger et de réagir comme voulu.

Le principe de robots roulants ayant déjà été étudié lors des années précédentes, nous avons pu nous appuyer sur les études de ces groupes et reprendre certaines de leur solutions. Cependant leurs problématiques n'ayant pas été les mêmes, nous avons du réétudier la majorité des robots et de leurs fonctionnalités.

Nous avons ainsi pu nous répartir les tâches au cours des quinze semaines afin de mener à bien ce projet et le développer au maximum. Nous avons fait cela avec l'aide de nos enseignants qui nous ont apportés leurs conseils ainsi que leur expertise et le matériel dont nous avions besoin.

Etude préliminaire :

Le principe d'essaim

Nos mini-robots devant s'inscrire dans un essaim, nous avons dû étudier ce principe.

Nous avons ainsi déterminé que les robots, tout en étant de taille réduite, devraient ne disposer que d'une connaissance partielle de leur environnement. En effet, les robots en essaim, tout comme les différents animaux (abeilles, banc de poissons), ne savent presque rien de leur environnement global et ne perçoivent que leurs congénères proches ainsi que les obstacles les environnant.

Pour cela, nous avons décidé de réduire au maximum l'algorithme des robots à des contrôles et des détections de proximités au travers de capteurs. De même, les robots ne communiquent aucune information entre eux mise à part leur identifiant, permettant de les différencier afin d'implémenter, par exemple, une fonction de train de robots.

L'étude des projets existants

Les projets des années précédentes ayant déjà produits des mini-robots, nous avons pu nous en inspirer.

En effet, ces robots étaient fonctionnels, ce qui nous a fourni une base solide de développement. Par exemple nous avons pu récupérer leur modèle de carte électronique afin de la retravailler et de l'adapter à nos fonctionnalités.

L'observation de ces systèmes nous ont aussi permis de choisir une conception physique du robot différente de celles développées durant les années précédentes. En effet, précédemment, le châssis était basé sur la carte électronique et les différents capteurs y étaient soudés directement dessus ou liés grâce à un ensemble de vis, d'entretoises et de plexiglass, ce qui rendait le tout assez fragile, difficilement modulable et donnant un rendu quelque peu brouillon. Nous avons préféré développer un châssis plus complexe, modélisé puis imprimé en 3D. Le fait que l'un de nous ai suivis des cours de Modélisation 3D au travers de la filière Système Autonomes (S.A.) nous confortait dans l'idée de produire une pièce solide et conçue de sorte à servir au maximum les fonctionnalités du système.

Le développement du Robot :

Le travail sur le système que nous devions créer s'est déroulé en plusieurs étapes importantes que nous nous sommes partagés en fonction, soit des préférences, soit des compétences préliminaires ou simplement du contexte.

Quoi qu'il en soit, les étapes se sont révélées découler logiquement autant d'un point de vue chronologique que d'un point de vue fonctionnel. C'est ce dernier que nous mettrons en avant dans ce rapport.

Carte Electronique

Une initiation épaulée :

L'étude de la carte électronique s'est réalisée dans un premier temps péniblement, du fait qu'aucun de nous deux n'était à l'aise avec la conception de circuits électroniques. Cependant avec l'aide de M. Redon, nous avons pu avoir accès à un modèle déjà réalisé pour des projets d'années précédentes que nous avons étudié et adapté au travers du logiciel Fritzing.

Une adaptation fonctionnelle :

Cette carte, contenant alors la plupart des composants nécessaires, n'était pas optimisée pour la fonction d'un robot en essaim. Nous avons donc dû la retravailler et enlevant ou ajoutant certaines parties.

D'abord nous avons remarqué la présence de moteurs électriques. Ayant fait le choix (expliqué dans la partie “*Composants*”) d'utiliser des servomoteurs, nous avons dû supprimer cette partie et implémenter le câblage des deux actionneurs.

De plus, le traitement des récepteurs infrarouges se faisait par alternance d'alimentation et par lecture sur une seule pin de l'Atmega. Avec la suppression des moteurs, nous avons alors pu utiliser deux pins de plus afin d'alimenter ces capteurs en continu et donc les lire sur trois pins différentes, permettant d'en utiliser plusieurs en même temps plus simplement.

Le châssis du robot permettant la mise en place des capteurs à distance de la carte, il a aussi fallu revoir les empreintes afin de permettre l'accès au travers de connecteurs externes, autant sur la face supérieures pour les capteurs infrarouges ou ultrasons, que sur la face intérieure pour les moteurs ou la pile.

Il a aussi fallu travailler le routage pour éviter certaines erreurs telles que des câbles trop proches les uns des autres jusqu'à se chevaucher, tout en évitant les angles droits ou de trop nombreuses connections entre les deux faces de la carte.

Enfin, la mise en place de la liaison USB à travers la puce FTDI et le connecteur USB-C n'a pas été à refaire, cependant son étude a été inévitable afin de vérifier les connexions aux différentes pins. L'intégration d'un programmeur AVR ISP a aussi été nécessaire afin de permettre la programmation du robot monté en cas de dysfonctionnement du port USB.

Le soudage :

Une fois la carte réalisée et gravée, il a fallu souder les différents composants. Pour cela, nous avons d'abord placé l'environnement du microprocesseur afin de réaliser la soudure au four.

Une fois ces fonctionnalités vérifiées, nous avons pu souder à la main les composants suivants, toujours fonctions par fonctions afin de limiter les risques de loupés.

Microprocesseur

Le microprocesseur est le composant le plus important de la carte électronique car il contrôle toutes les tâches du robot. Pour le choix du modèle, nous avons d'abord hésité entre une Raspberry Pi et une Atmega 328p (Arduino). Notre projet étant orienté vers un contrôle matériel de bas-niveau (moteurs, LEDs..), l'Arduino s'est révélé plus optimal que la Raspberry, plutôt orientée haut-niveau (flux vidéo, wifi, etc).

Nos robots devant réaliser différentes tâches en simultané, il a fallu implanter un ordonnanceur dans l'Arduino. Pour cela, nous avons choisi FreeRTOS, un mini OS (Système d'exploitation) pour Arduino.

Cependant cela s'est plus tard révélé être un mauvais choix car FreeRTOS prenait une grande partie de la mémoire disponible. Nous aurions dû choisir un microprocesseur plus complet ou programmer notre propre ordonnanceur.

Ce microprocesseur choisi, nous pouvions alors passer à sa programmation en AVR, et à quelques tests avec des composants. Nous avons d'abord décidé de nous attarder sur l'émission de l'identifiant du robot au travers d'une LED infrarouge, puis nous avons tenté de programmer la modulation et la transmission du signal contenant l'identifiant.

Composants

La réalisation de la carte électronique ne peut aller sans le choix des composants à placer afin de réaliser l'ensemble des fonctionnalités du robot.

Ces composants importants se divisent en deux grandes parties : les actionneurs et les capteurs.

Les actionneurs :

Parmis les actionneurs, les moteurs sont indispensables au robot roulant, nous avons donc eu à choisir leur type afin de permettre un contrôle optimal.

Nous avons ainsi choisi des Servomoteurs à rotation continue. Ceux-ci permettant des déplacements précis tout en réduisant la place sur la carte électronique car ils ne nécessitent qu'une seule sortie du microprocesseur pour commander à la fois le sens et la vitesse de rotation.

Les capteurs et la détection de l'environnement :

Les capteurs quant à eux répondent à différents critères techniques.

Nous voulions en effet que le robot puisse se déplacer dans l'espace et donc avoir une connaissance de cet environnement proche. Nous avons alors déterminé quels seraient les différents types d'objets à repérer : Les obstacles et les autres robots.

Les obstacles ne nécessitant pas de détection approfondie de leur type (matériau, forme..), nous avons décidé de placer un émetteur-récepteur à ultrasons à l'avant du robot, permettant un déplacement au sonar et donc d'éviter les objets encombrants.

La détection des robots était plus complexe. En effet, un véhicule devant repérer ses congénères afin de se comporter comme un membre d'un essaim, nous avons dû différencier cela de la détection d'obstacle. De plus, un exemple d'utilisation de notre système était la formation d'un train de véhicules : Les robots devraient se suivre un à un en file. Nous avions donc pour obligation de différencier non seulement les robots des objets, mais aussi les robots entre eux.

Nous avons ainsi choisi une détection par infrarouge que nous modulons afin d'émettre un identifiant unique pour chaque robot. La disposition se faisant de telle sorte que l'émission soit vers l'arrière grâce à une LED infrarouge placée sur le châssis, et la réception puisse se faire à l'avant ainsi que sur les côtés du robot au travers de capteurs infrarouges.

Nous avons aussi pensé réaliser une détection de faible distance au travers d'un capteur à effet Hall. Cette solution permet, grâce à un aimant implanté dans chaque robot de détecter la présence d'un robot à proximité par la modification du champ magnétique.

Infrarouge

Une fréquence d'émission aléatoire

Pour la transmission en infrarouge, chaque robot envoi son identifiant à intervalle aléatoire. En effet, la synchronisation des émissions par plusieurs robots peut entraîner des complications de lecture de leurs identifiants et nous devions donc empêcher toutes superpositions durables des signaux.

De la corruption des messages

Au fur et à mesure de la programmation de cette communication, nous nous sommes rendus compte que l'on allait se confronter à plusieurs problèmes, tel que la corruption du signal infrarouge par l'émission simultanée de messages par différents robots.

Pour cela, nous avons décidé d'utiliser un bit de parité en fin de message ainsi que des bits de stuffing: Tous les n bits similaires nous ajoutons un bit contraire afin de vérifier la non-corruption du message. Ainsi, si il y a $n+1$ bits à l'état haut à la suite, cela signifie que l'on reçoit deux signaux en même temps et on arrête de lire le message en cours, considéré comme corrompu.

Autre problème, comme l'émission et la réception ne sont pas synchronisées, il faut que l'on soit sûrs d'être au début de l'émission. Pour cela, on envoie $n+1$ bits de start.

Un message modulé

Les capteurs choisis étant restreints à la détection de signaux modulés a 38kHz, nous avons aussi dû respecter cette contrainte à l'émission.

Pour cela, nous avons créé notre propre trame :

- Pour savoir si le message en cours est corrompu, nous ajoutons des bits de stuffing tous les “ n ” bits identiques qui se suivent. Ainsi si on est à “ $n+1$ ” bits identiques qui se suivent, soit le message est corrompu, soit ce sont des bits de start.
- Etant donné que les phases d'émission et de réception ne sont pas synchronisées, Lors du commencement de la lecture, on peut se trouver à la moitié de la trame envoyée. Si on lit cette trame elle sera donc fausse. Pour éviter cela, on a également implémenté “ $n+1$ ” bits de start (ce qui est impossible en cours de trame)
- Pour s'assurer de la non corruption et de la bonne terminaison de la trame reçue, on a rajouté un bit de parité après l'identifiant et 2 bits de stop.

Déplacements

Pour se déplacer, le robot dispose de deux servo-moteurs disposés sur chaque côté latéral et d'une roue libre à l'arrière. Pour contrôler les servomoteurs, on doit utiliser une PWM (Pulse Width Modulation). Ceci s'exprime par un échelon haut durant “ t ” millisecondes, puis un niveau bas pendant “ $x-t$ ” millisecondes.

Dans notre cas, les moteurs sont pilotés par un échelon haut de 1 à 2 millisecondes toutes les 20 millisecondes. Ainsi, si l'échelon ne dure que 1 ms, alors le moteur tourne dans le sens négatif (le robot recule). Si il est de 2 ms, il tourne dans le sens positif, et est à l'arrêt si il est de 1.5ms.

Pour gérer ces moteurs, on va donc utiliser un *timer* de l'ATMega.

Ceux-ci étant assemblés en miroir afin de conserver l'alignement des roues, ils devront tourner en sens inverses afin de ne pas s'opposer et permettre au robot d'avancer ou de reculer.

Pour faire tourner le robot autour d'un cercle, les moteurs devront avoir des vitesses globales différentes. Si ces vitesses sont opposées, le robot tournera sur lui même sur place.

Modélisation 3D

L'étude préliminaire nous ayant permis de décider de la conception d'un châssis complexe pour nos robots, nous avons décidé de le modéliser et de le rendre imprimable au travers d'une imprimante 3D.

Un modèle basé sur la fonctionnalité :

Afin de réaliser le corps de notre système, nous avons utilisé le logiciel en ligne Onshape, très utilisé dans ce domaine.

La mise en place des points principaux a permis le développement global du modèle. Notre système devrait être composé de deux moteurs et d'une roue libre, d'un emplacement pour la pile et l'aimant, d'un support pour la carte électronique sur laquelle viendraient de nombreux câbles, ainsi que des fixations afin de maintenir les capteurs en place.

Nous souhaitions d'abord, par un soucis d'équilibre physique, placer la pile à l'opposé des moteurs. Cela empêcherait la renverse du robot lors d'accélérations brèves. Nous avons alors défini que les moteurs seraient à l'avant, et la roue folle ainsi que la pile à l'arrière pour permettre la compensation du couple réalisé par l'avance ou le recul du robot.

De plus, nous souhaitions réaliser un robot pratique, autant pour l'impression, le montage ou l'utilisation. Nous avons donc choisi de réaliser une compartimentation au travers d'étages fonctionnels offrant un confort de lisibilité, ainsi qu'un espace libre aux capteurs ou à la carte électronique, autant pour l'activité réalisée que pour les tests.

De ce fait, le chassis serait composé d'un premier étage accueillant les moteurs et la roue libre par vissage sur le dessous ainsi que la pile et l'aimant au dessus. Un deuxième étage offrant des plots de clipsage permettrait à la carte électronique de se fixer tout en laissant un libre accès aux différents câbles.

Enfin un troisième étage servant de support aux capteurs disposés de telle sorte qu'il n'y ait aucune gêne autant d'un capteur sur l'autre que par le châssis.

Par un soucis de montage, ce dernier étage serait séparé du reste de la pièce afin de donner la possibilité d'assembler d'une part les capteurs, et d'autre part la carte électronique avant venir visser le toit sur les poteaux prévus à cet effet.

Les limites de nos choix

Lors de l'avancée du projet, nous nous sommes rendus compte que certains de nos choix avaient développé certains problèmes amenant pour certains au blocage du projet.

Un microprocesseur sous-dimensionné :

Nous avons, comme dit plus tôt, choisi pour ce projet un microprocesseur ATMega328p. Ce composant étant la partie programmable de notre robot, il est nécessaire de le choisir en fonction de l'application à programmer.

Cependant, nous avons été confronté au manque de Timers par ce composant. N'en disposant de seulement 3, nous avons dû répartir plusieurs tâches sur le même timer.

Si nous avions eu plus de timer, nous aurions pu en utiliser un pour gérer la fréquence du modulant et ainsi augmenter la vitesse d'exécution.

Un système d'exploitation pas si “mini” que ça :

L'implémentation d'un OS déjà fait et complet tel que FreeRTOS a permis l'intégration d'un ordonnanceur fonctionnel rapidement. Cependant celui-ci utilise beaucoup de variables statiques si bien que la mémoire prise s'est révélée disproportionnée, allant même jusqu'à utiliser près de 85% de la mémoire vive du microprocesseur ce qui nous a empêchés de pouvoir stocker des données pour nos tâches. Après réflexion, nous aurions dû programmer notre propre ordonnanceur.

Une détection peu précise :

La mise en place d'un capteur à effet Hall permet de détecter la présence de robots dans un environnement proche, mais cela ne permet en aucun cas de déterminer la position, le nombre ou la distance de ceux-ci.

Pour palier à cela, nous aurions pu placer 4 capteurs du même type de chaque côté du robot afin de déterminer la position d'un possible robot. Ceux-ci étant de type *switch*, c'est à dire qu'ils délivrent une sortie binaire selon la détection ou non de détection d'un champ magnétique à proximité, et disposant d'un rayon peu important, la présence d'au moins un robot sur chaque côté serait assurée.

Aller plus loin

Lors de ce projet, nous avons ainsi réalisé un robot roulant qui pourrait de détecter les obstacles afin de les éviter ainsi que de repérer ses congénères. Cela donne l'accès au comportement d'un essaim de robots. Cependant si notre étude disposait de plus de temps, nous aurions pu mettre à bien la suite du cahier des charges.

Une avancée à courts termes

La programmation d'un ordonnanceur personnalisé :

Pour programmer notre ordonnanceur, il faut:

- Créer un pointeur de données SP, qui va pointer l'adresse de la pile de la tâche en cours d'exécution.
- Subdiviser la taille de la RAM de l'ATMega selon le nombre de tâches à exécuter.

Ensuite avec un timer, toutes les x ms, on change de tâche, on stocke les 32 registres de l'Arduino dans la pile de la tâche en cours. Puis on se place sur la pile de la tâche à exécuter.

On insère les 32 dernières valeurs de la pile dans les registres de l'ATMega. Puis on effectue la tâche à partir de là où on s'était arrêté, grâce au pointeur de programme, stocké dans la pile.

Cela aurait permis de libérer près de 70% de la RAM utilisée inutilement dans ce cas par FreeRTOS.

Une avancée à moyens et longs termes

Une interaction avec l'utilisateur :

Le système actuel permet la programmation des robots, cependant rien ne lie l'utilisateur au robot directement lors de son activation. Une interface utilisateur permettant le contrôle d'un leader ou de l'ensemble de l'essaim par des ordres diffus (le principe de l'essaim ne permet pas le contrôle de l'ensemble mais seulement une suite de buts simples que l'essaim traitera en fonction de leur environnement proche) pourrait être développée au travers, par exemple, d'une application Android ou d'une télécommande infrarouge.

Un test à grande échelle :

Le cahier des charges stipulant un “essaim de robots”, le tests sur deux robots n'est que le lancement. En effet, il faudrait produire plusieurs robots (au moins trois) afin de tester les limites du système et la coopération.

Un algorithme développé :

Un essaim étant un groupe d'individus réagissants simplement, nous pourrions aussi développer un algorithme d'intelligence artificielle simple permettant la réalisation d'actions basiques telles que se déplacer dans une pièce, tout en restant groupés, ou la mise en scène du projet : la cartographie d'une pièce. La présence des capteurs permet ce type d'actions, et un développement approfondi de l'algorithme de déplacement serait une suite logique à notre projet.

Cependant la cartographie implique un nouveau problème matériel : la mémoire.

L'ajout de mémoire externe :

Afin de conserver les données enregistrées pendant la cartographie, il est nécessaire d'implémenter un composant capable de les accueillir. Pour cela, il est possible de rajouter un support pour carte mémoire, simple d'utilisation et modulable, permettant à la fois de conserver les données sans surplus de poids et de les extraire facilement.

Conclusion

Ce projet s'est révélé ambitieux et complet, au cours de cette 4^{eme} année à Polytech Lille, nous avons pu apprendre et développer nos compétences sur de nombreuses disciplines en rapport avec notre formation ce qui est à la fois extrêmement bénéfique pour nos recherches de stages et futurs emplois, et agréable de part la réalisation intégrale d'un système.

Nous nous sommes aussi rendu compte de la difficulté de mettre en application nos connaissances sur un projet aussi complet, en particulier lors du lien entre les différentes parties que nous réalisions souvent séparément.

Le développement de la vision du travail que nous avons envie de fournir lors des stages et emplois dans notre future vie professionnelle a aussi été un point essentiel de ce projet, nous montrant l'importance de la communication au sein du groupe, de l'organisation des différentes parties du projet, ainsi nous donnant un idée d'un pendant du métier d'ingénieur : la pluridisciplinarité et la mise en oeuvre d'un projet, du cahier des charges aux tests en passant par la conception.