

CAPACITE DE SYNTHESE DE NOS ETUDIANTS

par

Pascal VRIGNAT

IUT de Châteauroux

Pascal.vrignat@univ-orleans.fr

PREAMBULE

Les étudiants d'IUT appartenant à la formation **Génie Electrique et Informatique Industrielle** vont pour la première année se rencontrer à travers un **aspect ludique de la technologie** : « la robotique mobile ». Les dates retenues sont : 6,7,8 et 9 juin 2002 à VIERZON. Pendant ces journées, 20 minutes seront consacrées à une présentation orale et publique des « robots » par les équipes . Nous sommes en général , assez critiques quant à la rédaction, à l'esprit de synthèse de nos étudiants.

Voici la présentation d'un robot par des étudiants rédigé à l'occasion d'une participation à un challenge de robotique mobile en 2000. **Ce document n'a pas été retouché par l'équipe enseignante.**

Bonne lecture et encore merci à **Bernard CARON** et aux collègues qui se sont ralliés à lui pour que cette rencontre ait lieu. Un grand merci également à **Michel AUFAUVRE** et à toute son équipe de l'association ASTECH de VIERZON qui assure le financement ainsi que la logistique de ces 4 jours du mois de juin.

PROJET ROBOT IUT DE CHATEAUROUX

Louis LEPINE, Christophe MAUBOIS, Florent MICHALLON, Benoît PASCAUD

Etudiants en 2^{ème} année GEII 2000

IUT de Châteauroux, Département GEII
2, avenue François Mitterrand, 36000 Châteauroux
tel 02.54.08.25.50

Résumé : Ce document présente toute la démarche engagée par le groupe d'étudiants de l'IUT de l'Indre chargés du projet robot pour mener à bien leur mission. Ainsi, après une brève introduction, nous retrouverons un développement pour chaque composante du robot, du châssis au contrôleur. La présentation de ces différents éléments se fait par ordre de réalisation : carte de commande avec le PIC16F877 [1], carte d'adaptation de l'énergie avec le convertisseur L78S05CV , carte de détection de piste avec les capteurs infrarouge référencés EE-SB5/-SB5-B [2], présentation du châssis, de l'alimentation générale, du système de ramassage des boules, et enfin la carte de détection d'obstacles avec les transducteurs ultrasoniques 400ET180/400ER180 ou un simple contacteur et le PIC16F876 [1]. Le résultat est à la hauteur de nos espérances : un robot rapide et simple à adapter à toute situation de par le mode de programmation du microcontrôleur et les technologies embarquées.

Mots clés : microcontrôleur, infrarouge, ultrason, pont en H, simplicité, fonctionnalité.

Principales caractéristiques du robot :

- 75*47*20cm
- poids 5Kg
- Vitesse normale: 3.6 m/s. Vitesse Maximale : dépend de la configuration.
- Microcontrôleur 16f877 et 16F876, ponts en H L6202 [3].
- Programmé en C, liaison série avec l'ordinateur.
- Batterie de 12 volts, autonomie de 30 minutes.
- Capteurs infrarouges et transducteurs ultrasoniques.
- Prix : 570 €.

1. INTRODUCTION

Cette année, comme pour les années précédentes, l'objectif de l'IUT de Châteauroux est de finaliser un "automate" capable de participer au championnat international de robotique mobile dans la catégorie monotype.

Notre groupe est formé de quatre membres qui sont :

- Louis LEPINE,
- Christophe MAUBOIS,
- Benoît PASCAUD,
- Florent MICHALLON.

Nous sommes encadrés par deux enseignants de l'IUT de Châteauroux :

- M. Eric PERONNIN, enseignant en Electronique et Informatique ;
- M. Bruno EMILE, enseignant en Asservissement, Logique, Informatique Industrielle et Mathématiques.

A travers ce dossier nous essayons d'expliquer comment nous avons travaillé ainsi que les choix qui ont été faits. Les grands axes sont définis comme suit : Définition du besoin, Définition de la structure de base, Définition des différents éléments et composants internes à chaque partie.

2. LE ROBOT

Le challenge monotype auquel nous participons pour le compte de l'IUT peut être défini comme suit : il s'agit de développer une plateforme physique et logicielle capable de répondre précisément aux exigences de la compétition.

Il faut pour cela concevoir un robot capable des différentes actions :

- détecter une piste ;
- suivre cette piste ;
- détecter des obstacles ;
- éviter ces obstacles ;
- récupérer des boules de billard ;
- s'arrêter à la fin du parcours.

Tout ceci doit se faire de façon totalement autonome (pas de télécommande ou d'enregistrement du parcours...).

3. LA REALISATION

3.1. Méthode de travail

Quelle que soit la partie à traiter le fonctionnement du groupe a toujours été le même et se déroulait suivant 4 points :

- ❖ premièrement, analyse rapide de l'importance du sujet pour décider

combien de personnes y seront affectées. Dès lors seule les personnes concernées poursuivront la suite de l'analyse;

- ❖ Deuxièmement, recherche des différentes méthodes permettant d'accomplir la tâche et élimination des moins pertinentes.
- ❖ Troisièmement, essais sur platine des différentes solutions et composants, puis validation de la solution retenue.
- ❖ Quatrièmement, réalisation concrète de la partie « hard » et implémentation des éventuels programmes nécessaires.

De plus, nous considérons, qu'il est nécessaire que tous les membres du groupe soient à même d'appréhender l'ensemble du robot et de ses caractéristiques. Cela implique qu'une bonne communication soit établie entre les différents membres.

3.2. Présentation générale du robot

Les bases étant définies, nous pouvons commencer à entrer dans 'le vif du sujet'. Pour plus de clarté dans ce qui va suivre, le robot sera assimilé à un être vivant, et, ainsi, un organe sera affectée à chaque fonction déterminée précédemment :

Le squelette est tout simplement le châssis du robot.

Le cœur est composé de la batterie.

Le cerveau correspond à la carte comportant le microcontrôleur et la conversion de l'alimentation. C'est la carte pensante de l'ensemble, analysant les données et agissant en conséquence.

La partie yeux correspond à la carte de détection de piste avec les différents retours de capteurs.

Les jambes correspondent à l'ensemble de la commande des moteurs. Ceci se trouve sur la carte comprenant le microcontrôleur et ce pour des raisons de simplicité de connexion.

Les oreilles à la carte de détection d'obstacles, celle-ci étant en cours de réalisation, le choix entre les transducteurs ultrasoniques et le contacteur.

Les bras sont le système de retenue des boules autour du robot.

3.2.1. La tête : centrale de commande et électrique

Elle se trouve sur la carte le "cerveau" du robot, le module de conversion permettant d'obtenir l'alimentation 5V, ainsi que la commande des moteurs. Cette partie de carte doit être la plus simple possible, c'est à dire que les composants qui sont utilisés ne doivent pas nécessiter l'utilisation de structures

complexes conduisant généralement à ce que l'on appelle une usine à gaz. Notre choix est également légèrement dirigé par nos enseignants, qui refusent toute structure trop complexe ou impersonnelle.

Comme cette carte est décomposée en deux sous-parties, nous verrons donc successivement les deux aspects, à savoir Alimentation et Contrôle. Outre la simple justification de choix du matériel utilisé, nous verrons également la partie programmation de la structure de commande.

3.3.1.1. Partie alimentation

3.3.1.1.1. Besoins

Nos besoins sont les suivants : pouvoir fournir une tension d'alimentation à tous les composants et qu'elle soit suffisamment puissante, donc adaptée.

Les différentes tensions nécessaires sont ainsi définies :

- Microcontrôleurs : 5/0 Volts ;
- Amplificateurs : -12/+12 Volts ;
- Moteurs : 3 à 9 Volts ;
- Ponts en H (pour commander les moteurs) : 12/0 Volts ;
- Emetteur à ultrasons : 12/0 Volts ;
- Capteurs Infrarouges : 5/0 Volts ;
- Portes logiques : 5/0 Volts.

3.3.1.1.2. Réponses possibles

En source principale, nous avons peu de possibilité : la batterie s'impose, en tension 6 ou 12 Volts. Dans le premier cas, la valeur est très proche de celle recherchée (5 Volts), mais il apparaît difficile de retrouver la tension utile pour les émetteurs, les ponts en H et les différents amplificateurs. A partir du 12 Volts, cependant, il est aisé de reproduire n'importe quelle tension inférieure. Dans cet objectif, nous pouvons nous diriger vers deux choix :

- un convertisseur, qui en sortie nous donne du 5, -15, +15 Volts ;
- des régulateurs 5 Volts, - 5Volts, 12 Volts et -12 Volts.

3.3.1.1.2. Réponse adoptée

Après essais, il s'est avéré que les convertisseurs ne remplissaient pas leur rôle : c'est donc la deuxième solution, qui consiste à utiliser des régulateurs de tension continue positive ou négative, qui a été retenue.

3.3.1.2. Partie contrôle

3.3.1.2.1. Matériel

3.3.1.2.1.1. Besoins

Pour réaliser la fonction de contrôle, nous devons trouver des composants simples à utiliser, qui nécessitent peu d'éléments annexes. De plus, ils doivent être courant sur le marché (dans

l'hypothèse de panne lors d'éventuels déplacements à l'étranger du projet robot), d'un coût peu élevé, et surtout pouvoir très facilement et rapidement s'adapter à une nouvelle utilisation.

3.3.1.2.1.2. Réponses possibles

Dans le cadre d'actions dépendantes d'entrée de capteurs, nous pouvons sélectionner deux types de composants de contrôle :

- des portes logiques, qui seront agencées suivant des équations combinatoires ;
- des microcontrôleurs, qui peuvent avoir plusieurs fonctions.

On s'aperçoit ici que des deux possibilités la première est déjà éliminée, car le système de contrôle a été défini comme devant être simple et pouvant être facilement modifiable, ce qui n'est pas le cas de simples composants logiques montés les uns à la suite des autres.

Cependant, il existe plusieurs types de microcontrôleurs : il faut encore déterminer le plus adapté à notre besoin. Trois séries s'offrent à nous : PIC, 6809 et 8051.

3.3.1.2.1.2. Réponse adoptée

Les séries 8051 et 6809 sont des composants qui ont fait leurs preuves, mais ils nécessitent du matériel autour, et la programmation n'est pas ni des plus rapides ni des plus aisées. Le PIC semble donc être celui qui correspond le mieux à nos exigences. D'où notre choix : un microcontrôleur de la famille PIC de Microship. Il ne nous reste plus qu'à le définir exactement. La taille mémoire doit être suffisamment importante pour accueillir plusieurs programmes, il doit y avoir possibilité de contrôle par utilisation de PWM (Pulse Width Modulation : on fait varier le rapport cyclique du signal pour effectuer la commande. Ceci sera détaillé plus en détails plus loin), et éventuellement d'avoir des entrées analogiques. Au vue de ces demandes, nous avons sélectionné le PIC 16F877 (appelé par la suite PIC).

3.3.1.2.2. Logiciel

La partie logicielle correspond à l'écriture du programme de commande du robot, implanté sur le microcontrôleur.

3.3.1.2.2.1. Besoins

Nos besoins sont relativement simples : apporter une solution à un problème, qui est de définir la stratégie de conduite de notre automate. Plus clairement, il est défini par les fonctions que doit remplir le robot :

- suivre la piste ;
- détecter des obstacles ;
- contourner les obstacles.

Comme la présentation du robot l'a montré, une carte est spécialement dédiée à la fonction de

détection des obstacles, et sera donc par conséquent doté du programme adéquat. Cependant, il reste les deux autres éléments qui sont propres à ce chapitre.

3.3.1.2.2.2. Réponses possibles

Afin de résoudre le problème, le groupe de travail s'est interrogé sur les différents moyens de commander un moteur (en utilisant bien sûr une interface de puissance). Nous en sommes arrivés à la conclusion qu'il y a seulement deux moyens principaux de pilotage :

- une commande directe de l'interface de puissance. C'est le principe de l'interrupteur : si condition vraie, alors allume, sinon éteint.
- Par une commande plus « intelligente », qui consiste à utiliser des fonctions spécifiques de notre composant : la gestion par Modulation de Largeur d'Impulsion (MLI ou PWM), ainsi que les interruptions, qui permettent d'exécuter un programme sans tenir compte de ce qui se passe autour du composant, et donc des informations qui lui parviennent.

Dans l'optique de réaliser un robot capable de réagir vite et bien, on peut d'ores et déjà éliminer la commande directe. En effet, un tel principe de fonctionnement implique un comportement un peu « carré » du robot. En simplifiant, sa démarche sera plus proche du zigzag que de la ligne droite (ou légèrement courbe). En comparaison, la deuxième méthode nécessite un algorithme précis, qui va faire entrer en jeu de nombreuses caractéristiques spécifiques au microcontrôleur. Cependant, le gain n'est pas négligeable : la variation de commande sera moins brusque, car pouvant vraisemblablement être associée à une échelle de grandeur. Plus simplement, le contrôle n'est pas du type tout ou rien, mais plutôt proportionnel à une référence (vitesse maximale du robot).

3.3.1.2.2.3. Réponse adoptée

Au vu des remarques émises dans la partie précédente, nous avons jeté notre dévolu sur la solution intelligente, malgré les potentielles difficultés que nous pourrions être amenés à rencontrer lors du développement du programme. Néanmoins, celui-ci doit être défini le plus clairement possible dans ces fonctions. Les fonctions dont nous parlons sont aussi bien internes au programme (sous-programmes) que générales (propres au robot).

A partir de ce point, on peut commencer à résumer les différentes fonctions que doit réaliser le programme :

- lire l'état des capteurs de piste ;

- les comparer par rapport à une banque de donnée constituée des positions possibles du robot sur la piste ;
- associer un ordre à la position ;
- envoyer cette ordre à l'interface de puissance ;
- se brancher sur la fonction d'évitement d'obstacle et se tenir uniquement à cette fonction pendant toute la durée de la manœuvre ;
- communiquer avec un ordinateur.

Ce schéma bloc nous permet de définir l'algorithme du programme. Celui est développé en langage C en annexe III. Cet algorithme est alors créé.

Fonction : récupère l'information en provenance des capteurs de piste et agit suivant cette valeur.

Variables d'entrée : PortD registre, Piste char.

Variables internes : VitesseMoteurDroit bit, VitesseMoteurGauche bit, DrapeauFinTimer drapeau, Faux booléen, Marche booléen, Arrêt booléen.

Variables de sortie :

contrôle{1,CommandeMoteurGauche, CommandeMoteurDroit, SensGauche1, SensGauche2, SensDroit1,SensDroit2,1} structure char.

Début

Tant que 1 Faire

Début

Lire PortD

Piste ← PortD

Tant que DrapeauFinTimer = Faux Faire

Début

VitesseMoteurDroit ← 100

VitesseMoteurGauche ← 100

Contrôle.SensGauche1 ← Marche

Contrôle{SensGauche2} ← Arrêt

Contrôle{SensDroit1} ← Marche

Contrôle{SensDroit2} ← Arrêt

Fin

Si ((Piste = 0011100) ou (Piste = 1100011))

Alors

Début

VitesseMoteurDroit ← 100

VitesseMoteurGauche ← 100

Contrôle{SensGauche1} ← Marche

Contrôle{SensGauche2} ← Arrêt

Contrôle{SensDroit1} ← Marche

Contrôle{SensDroit2} ← Arrêt

Fin

AutreSi ((Piste = 0011000) ou (Piste = 1100111))

Alors

Début

VitesseMoteurDroit ← 100

VitesseMoteurGauche ← 90

Contrôle{SensGauche1} ← Marche

Contrôle{SensGauche2} ← Arrêt

```

    Contrôle{SensDroit1} ← Marche
    Contrôle{SensDroit2} ← Arrêt
Fin
AutreSi ((Piste = 0001100) ou (Piste =
1110011))
Alors
Début
    VitesseMoteurDroit ← 90
    VitesseMoteurGauche ← 100
    Contrôle{SensGauche1} ← Marche
    Contrôle{SensGauche2} ← Arrêt
    Contrôle{SensDroit1} ← Marche
    Contrôle{SensDroit2} ← Arrêt
Fin
AutreSi ((Piste = 0111000) ou (Piste =
1000111))
Alors
Début
    VitesseMoteurDroit ← 75
    VitesseMoteurGauche ← 100
    Contrôle{SensGauche1} ← Marche
    Contrôle{SensGauche2} ← Arrêt
    Contrôle{SensDroit1} ← Marche
    Contrôle{SensDroit2} ← Arrêt
Fin
AutreSi ((Piste = 0001110) ou (Piste =
1110001))
Alors
Début
    VitesseMoteurDroit ← 100
    VitesseMoteurGauche ← 75
    Contrôle{SensGauche1} ← Marche
    Contrôle{SensGauche2} ← Arrêt
    Contrôle{SensDroit1} ← Marche
    Contrôle{SensDroit2} ← Arrêt
Fin
AutreSi ((Piste = 0000000) ou (Piste =
1111111))
Alors
Début
    VitesseMoteurDroit ← 0
    VitesseMoteurGauche ← 0
    Contrôle{SensGauche1} ← Arrêt
    Contrôle{SensGauche2} ← Arrêt
    Contrôle{SensDroit1} ← Arrêt
    Contrôle{SensDroit2} ← Arrêt
Fin
Sinon
Début
    VitesseMoteurDroit ← 80
    VitesseMoteurGauche ← 40
    Contrôle{SensGauche1} ← Marche
    Contrôle{SensGauche2} ← Arrêt
    Contrôle{SensDroit1} ← Marche
    Contrôle{SensDroit2} ← Arrêt
Fin
Fin
Fin

```

Cet algorithme a été réalisé pour une lecture de piste par seulement trois capteurs, et illustre assez bien la méthode de réflexion du robot pour déterminer sa route. Les constantes utilisées correspondent à une valeur proportionnelle de la commande maximale: 100 = à fond et 0 = arrêt. Le drapeau nous indique qu'un cycle de deux secondes s'est écoulé. Cela se fait par l'intermédiaire d'un Timer du microcontrôleur qui sert d'horloge de base au système de MLI, en étant le support de la période du signal.

De plus, pour que le robot soit autonome, il faut que ces actions se répètent indéfiniment. D'où la présence d'une boucle infinie en début d'algorithme, avec les problèmes de blocages qu'elle peut entraîner.

Toutefois, il ne faut pas oublier le programme de contournement d'obstacle. Celui-ci est simplement basé sur une suite d'ordres dépendants uniquement d'une temporisation entre chaque étape de l'évitement.

Les valeurs de Td1, Tg et Td2 correspondent à une temporisation de la commande.

On en déduit l'algorithme qui lui est adapté.

Fonction : évite un obstacle sur la piste.
 Variables d'entrée : ObstacleDétecté byte
 Variables internes : DrapeauFinVirageDroit1 drapeau,
 DrapeauFinVirageGauche drapeau,
 DrapeauFinVirageDroit2 drapeau, Faux booléen
 Variables de sortie : aucune

```

Début
    Tant que DrapeauFinVirageDroit1 = Faux Faire
    Début
        VitesseMoteurDroit ← 65
        VitesseMoteurGauche ← 100
        Contrôle{SensGauche1} ← Marche
        Contrôle{SensGauche2} ← Arrêt
        Contrôle{SensDroit1} ← Marche
        Contrôle{SensDroit2} ← Arrêt
    Fin
    Tant que DrapeauFinVirageGauche = Faux Faire
    Début
        VitesseMoteurDroit ← 100
        VitesseMoteurGauche ← 65
        Contrôle{SensGauche1} ← Marche
        Contrôle{SensGauche2} ← Arrêt
        Contrôle{SensDroit1} ← Marche
        Contrôle{SensDroit2} ← Arrêt
    Fin
    Tant que DrapeauFinVirageDroit2 = Faux Faire
    Début
        VitesseMoteurDroit ← 85
        VitesseMoteurGauche ← 100
        Contrôle{SensGauche1} ← Marche
        Contrôle{SensGauche2} ← Arrêt
    Fin

```

```

Contrôle{SensDroit1} ← Marche
Contrôle{SensDroit2} ← Arrêt
Fin
Fin

```

La valeur de commande correspondant aux deux variables de vitesse a été définie totalement au hasard, la régulation s'effectuant sur la durée des actions. Celles-ci sont déterminées, très simplement, par des essais sur piste.

Pour les deux programmes ainsi définis, il est intéressant de noter que n'est déterminée ici que la structure des fonctions principales. En effet, pour chacun des moteurs il convient de ne pas oublier qu'une initialisation et la définition de fonctions spéciales (gestion de la MLI, des interruptions, des différents compteurs) sont nécessaires.

3.3.1.2.2.4. Principe

Le principe de fonctionnement est simple : au début de chaque cycle de scrutation du programme, le PIC récupère l'image de la piste sous la forme d'un mot de huit bits. Ce mot est ensuite comparé à une banque d'images de piste possible : le robot sait alors où il se trouve. A partir de là, il lui est facile de se déplacer de telle sorte d'être toujours sur la même trajectoire, car à chaque positionnement possible, un ordre de correction peut être associé et ainsi commander les moteurs en conséquence. Cependant, et ce à n'importe quel moment du programme, le PIC peut recevoir l'information comme quoi un obstacle est présent sur la piste ; dans ce cas une sauvegarde de l'état du microcontrôleur est réalisé et le programme associé à cette interruption est exécuté sans tenir compte des informations qui peuvent parvenir au PIC. Une fois que la (les) tâche(s) d'interruption est (sont) terminée(s), une restitution de l'état du microcontrôleur est faite et le programme reprend son cours à l'endroit où il avait été interrompu.

Remarque : l'état du microcontrôleur correspond à l'ensemble des registres et variables qui lui sont propres, et dépendants de la situation globale dans laquelle se trouve le composant dans son programme.

3.3.2. Les yeux

3.3.2.1. Besoins

Le but est de détecter une piste qui est alternativement blanche sur fond noir et noire sur fond blanc, de 5 cm de large (une description plus précise en est faite dans le règlement fournie en annexe I). Il faut donc renvoyer une information au PIC principal indiquant clairement la position du robot.

3.3.2.2. Les solutions

Pour réaliser cette fonction nous avons le choix entre deux méthodes différentes :

- mise en œuvre d'une caméra,
- utilisation de capteurs infrarouges.

L'emploi d'une caméra aurait obligé à effectuer un traitement de l'image et donc nécessité une technologie très lourde, de même qu'une grosse quantité d'étude. Au contraire, les capteurs infrarouges imposaient beaucoup moins de contraintes, quant à leur mise en œuvre.

Nous avons donc opté pour cette seconde solution et recherché les capteurs qui correspondaient le plus à nos exigences (capacité à différencier aisément le noir du blanc).

3.3.2.3. Le principe

Pour connaître la position du robot, nous disposons une bardée de capteurs en ligne et séparés par des distances judicieusement choisies. Ceci pour voir si l'on se trouve plus ou moins loin de la piste. La fig.1 présente la disposition qui a été adoptée.

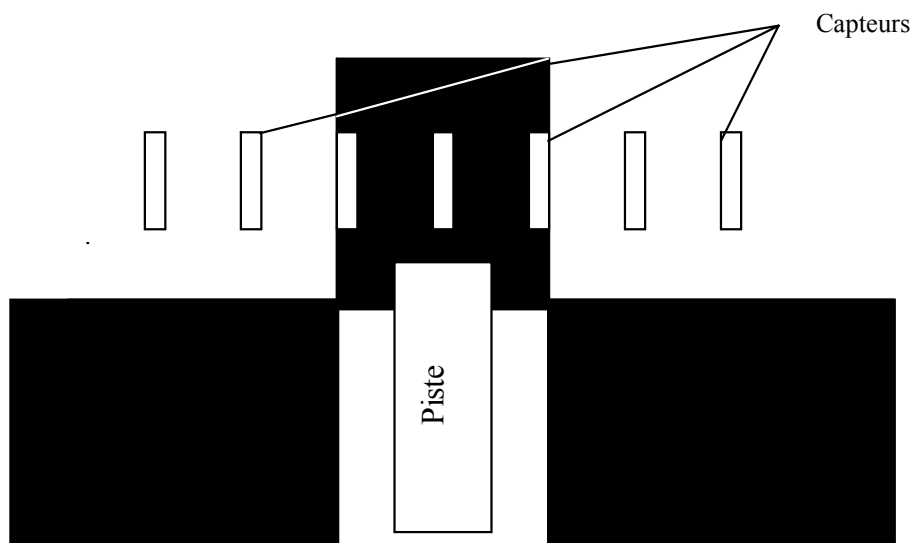


Fig.1 : emplacement des capteurs de piste

Après mûre réflexion, cette solution semble être la plus efficace. L'éventualité d'une disposition en pyramide ou en zigzag a été évoquée, mais réfutée car il devenait alors très difficile d'envisager tous les cas possibles d'information renvoyée par les capteurs (en particulier lors des changement de zones).

Les signaux des capteurs sont mis en forme pour pouvoir être considérés comme des informations tout ou rien par le PIC principal.

3.3.2.4. La solution concrète

Une carte a finalement été dédiée à cette fonction. Elle comporte donc sept capteurs disposés comme indiqué précédemment ; elle est capable de renvoyer au PIC principal l'état des capteurs, de manière tout ou rien, via un connecteur.

3.3.2.5. Remarques

Cette carte comprend aussi une partie permettant l'adaptation des signaux du retour tachymétrique, fourni avec la maquette, pour pouvoir être utilisé par le pic.

Mais comme la stratégie que nous avons décidé d'adopter ne nécessite pas l'utilisation de cette information, il n'en sera pas fait description dans cet ouvrage. Rappelons simplement que cette fonction est câblée sur le pic et qu'elle pourra être mise en œuvre le moment venu.

3.4. Les jambes : courir le plus vite possible

Il nous fallait un système permettant de commander les moteurs de notre plate-forme à partir d'instructions fournies par le PIC. Une carte a donc été mise au point afin de remplir cette tâche.

3.4.1. Définition des besoins

Il nous fallait un module :

- Capable d'accepter une commande d'entrée en 0-5V ;
- Pouvant donner une tension de sortie compatible avec celle des moteurs : à savoir 3-12V ;
- Capable d'accepter des variations de la tension moyenne à son entrée afin de pouvoir faire varier la vitesse des moteurs.

3.4.2. Les solutions disponibles

Nous pouvons réduire ces solutions à trois cas distincts.

3.4.2.1. Solution industrielle

Nous aurions pu reprendre les cartes industrielles présentes sur la plate-forme qui était celle que nous devions utiliser au départ. Il s'agissait de variateurs MS608. Ces cartes permettaient en effet de répondre à une grande partie de nos besoins ; cependant elles n'étaient pas compatibles avec la plage de fonctionnement de nos moteurs et la puissance qu'elles fournissaient était beaucoup trop élevée.

3.4.2.2. Solution autonome

Nous aurions aussi pu réaliser nous même un système de conversion permettant de commander les moteurs à partir des signaux du fournis par la carte PIC. Cependant une telle réalisation est assez lourde à mener et surtout très gourmande en temps : elle nécessite une étude complète des besoins, la mise au point d'un montage (qui implique elle même beaucoup de recherche documentaire sur tous les composants utilisés), le test et les modifications de ce montage pour qu'il 'colle' parfaitement à ce que l'on attend de lui.

3.4.2.3. Solution commerciale

La troisième solution était de faire appel à une structure toute faite et qui nous permettait d'obtenir les

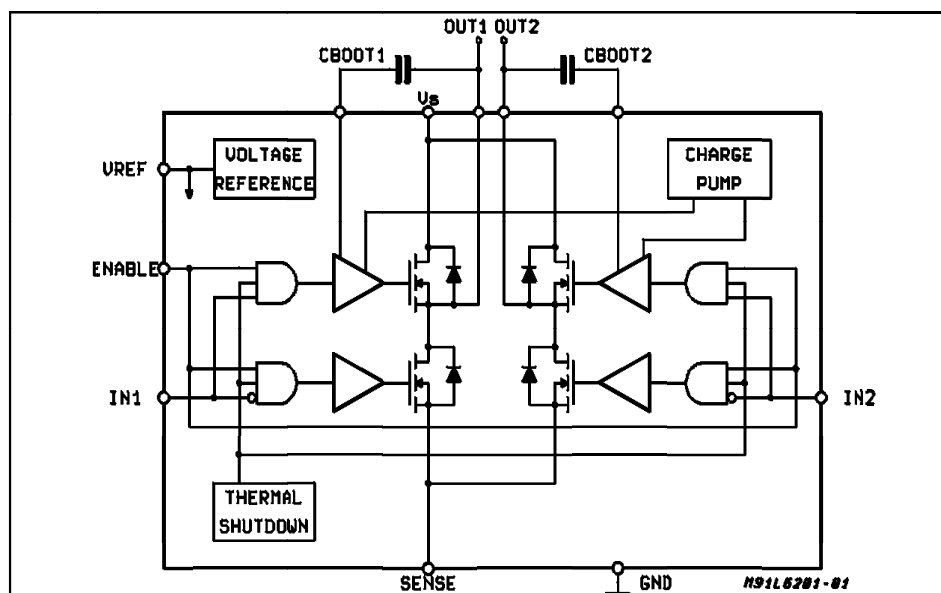


Fig.2 : Structure du pont en H

résultats escomptés avec un minimum de câblage supplémentaire. Une telle structure est vulgairement appelée un 'pont en H' du fait de l'agencement des composants qui le constituent. Le schéma donné en Fig.2 permet de voir pourquoi le montage est appelé 'un pont en H'.

La forme du H vient du fait que l'on retrouve quasiment la même structure de part et d'autre de l'axe vertical médian du montage.

Un module de ce type permet généralement de contrôler une rotation de moteur dans les deux sens (certains ne permettent qu'un seul sens de rotation). De plus, en choisissant bien le modèle de composant qui contient cette structure, on peut obtenir la plage de tension que l'on souhaite aussi bien en entrée qu'en sortie.

3.4.2.4. Solution retenue

Nous avons choisi la dernière des possibilités énoncées ci-dessus. En effet, il apparaît que c'est la plus simple à mettre en œuvre. Elle implique tout de même de concevoir une carte pour la recevoir.

Le module que nous avons utilisé est désigné sous le nom de L6203. il est construit par SGS-THOMSON Microelectronics. Ce boîtier constitue un pont complet (permettant la rotation dans les deux sens du moteur qui lui est connecté) réalisé à partir d'une technologie dite BCD-Multipuissance qui permet d'utiliser des transistors de puissance DMOS avec des circuits CMOS et Bipolaire ; le tout étant mis dans le même composant.

Un tel circuit peut fonctionner de 9 à plus de 48 V dans ces conditions classiques d'utilisation. Cependant, il est possible de le faire fonctionner dans une plage de tension allant de 9 à 18 V en réalisant un montage adapté.

De plus, il possède des vitesses de commutation au niveau de sa composition interne

très rapide et toutes ses entrées sont compatibles microcontrôleurs ce qui permet de le contrôler à partir d'un signal transmis par un PIC avec une fréquence de rafraîchissement assez élevée ce qui n'est pas négligeable si l'on veut que le système réagisse vite.

Chacune des parties (demi-pont affecté chacun à un sens de rotation) du composant possède sa propre entrée : cela permet de contrôler le sens de rotation du moteur. Nous avons choisi la version appelée multiwatt car il est possible d'y fixer un dissipateur de chaleur car un tel composant fournissant de la puissance a tendance à chauffer.

La carte que nous avons réalisée autour de ce composant est présentée en annexe IV. Elle permet de contrôler les deux moteurs de la plate-forme, c'est pour cela qu'elle comporte deux boîtiers L6203.

3.5. Le cœur : partie vitale par excellence

Le cœur sert à fournir de l'énergie à tout le reste du robot. De plus, cette énergie se doit d'être indépendante : le seul choix possible comme source d'alimentation du Robot est donc une Batterie. La valeur de la tension qu'elle doit délivrer est définie par rapport aux exigences des différents composants. Ainsi, à cause de ceux utilisés pour la commande des moteurs, on choisit une batterie de 12 V. L'alimentation sera ensuite répartie sur les différentes cartes par l'intermédiaire de câbles dédiés à cette fonction. En quelque sorte ce sont les veines et artères de notre entité.

Une représentation du circuit d'alimentation à réaliser est donnée en Annexe II.

3.6. Le squelette : un châssis solide et fonctionnel

C'est une plate-forme circulaire avec un étage et des pattes de fixation pour les moteurs et les capteurs optoélectronique de retour d'information pour la vitesse du robot.

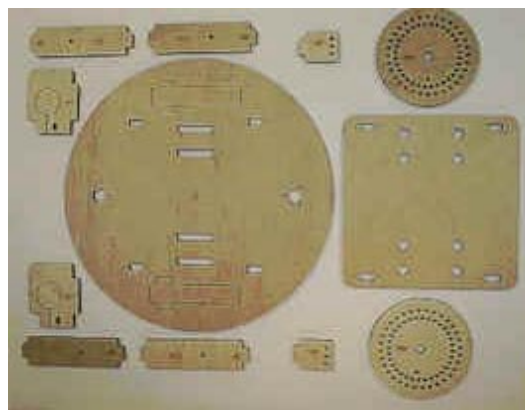


Fig.3 : Eclaté du châssis

Un schéma plus éclaté nous donne la représentation de la fig.3.

Notre but est de motoriser et de commander cette structure par une méthode appropriée. Nous devons tenir compte également que ce squelette aura une peau, symbolisée éventuellement par une coque, mais de manière plus sûre par la jupe qui servira de système de retenue des boules.

3.7. Les bras : boules, je vous tiens !

Nous allons nous attacher ici à décrire, outre le système de retenue des billes de billard, l'aspect extérieure de notre robot.

3.7.1. Besoins

Nous avons besoin, dans le but de répondre aux exigences du règlement, de concevoir un système capable de ramasser et garder des boules. Comme le terme « ramassage » est quelque peu vague, nous allons donc maintenant préciser la signification de ce mot.

Réglementairement, il s'agit simplement de faire en sorte que les billes disposées sur la piste soient à la fin du parcours avec le robot, dans le

robot, sur le robot, bref, elles doivent faire partie de ce qui va passer la ligne d'arrivée.

Dans ce cadre ainsi défini, nous pouvons déterminer les exigences fondamentales :

- concevoir un système très léger ;
- concevoir un système simple à fixer ;
- le système ne doit en aucun cas entraver la progression du robot ;
- cela doit apporter une certaine protection à l'ensemble de la structure.

Connaissant ces quatre conditions principales, nous pouvons alors envisager les différentes structures possibles.

3.7.2. Réponses possibles

3.7.2.1. Première solution

La porte est un système ne permettant le passage des boules que dans un seul sens. Ce choix est le seul point commun (avec le châssis) pour toutes les solutions. Il est réalisé, tout comme les parois, en aluminium. Une représentation est donnée en fig.3.

Ce schéma montre bien la simplicité du système, mais qui cependant rencontre des limites : la boule sera poussée, et donc elle peut exercer des forces sur l'avant du robot et dans ce cas le perturber dans sa trajectoire.

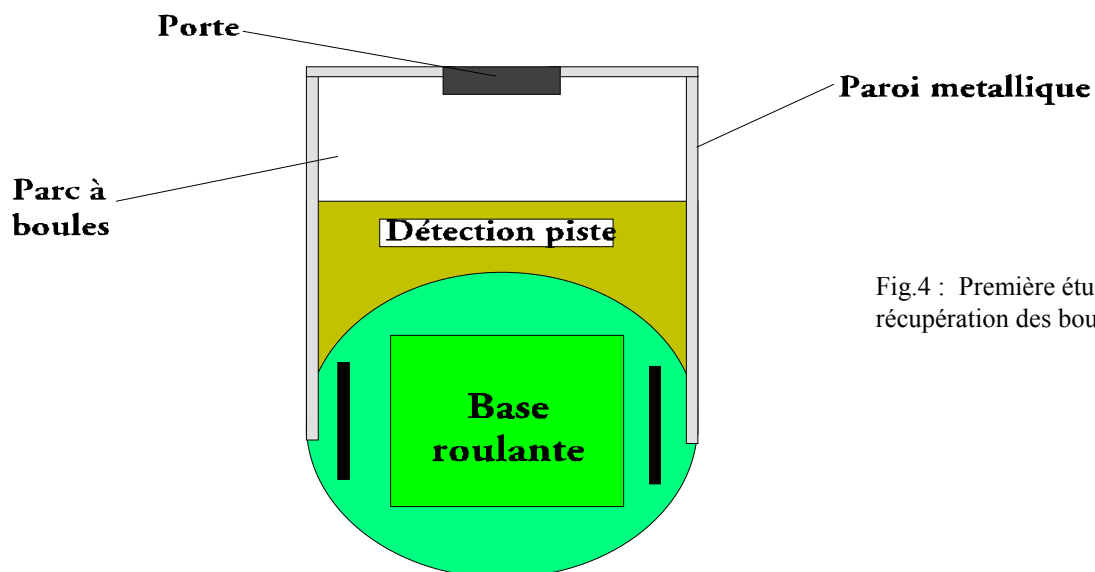


Fig.4 : Première étude de récupération des boules.

3.7.2.2. Deuxième solution

Dans ce cas, la boule est conduite à travers un demi-tube en plastique tout autour du robot. L'intérêt est que les boules vont s'équilibrer toutes seules à l'arrière, et le comportement du robot sera moins perturbé. Cependant, il reste toujours un problème : lors du choc de la boule sur le conduit, la structure risque d'être déséquilibrée, et donc de quitter la trajectoire idéale ou bien, ce qui peut être pire, la boule peut rebondir et ressortir de ce conduit. Un aperçu est en fig.4.

Ici, la seule difficulté est de faire une jupe qui ne puisse en aucun cas toucher le sol. Ceci dépend donc principalement des moyens de fixation de cette jupe sur le châssis.

3.7.3. Bilan sur le ramassage de boules

Nous voyons ici un nouvel aspect de la conception : l'improvisation, car notre spécialité n'est pas du tout la mécanique. Cependant, grâce à des matériaux adaptés et une étude précise, nous pensons que la dernière solution envisagée est de

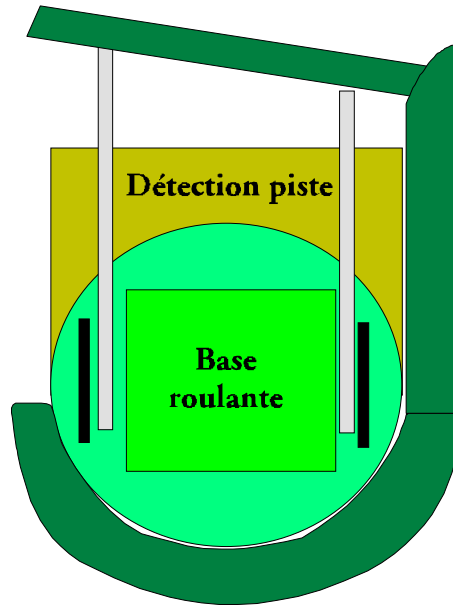


Fig.5 : Deuxième solution pour le ramassage des boules

3.7.2.3. Troisième solution

Cette dernière solution consiste à réaliser un conjugué des deux premières : une jupe tout autour du robot avec le système de clapet anti-retour, avec d'autres portes de ce type plus bas pour faire en sorte que les boules se stabilisent bien à l'arrière du robot. Un croquis de cette possibilité est montré en Fig.6.

loin la plus fiable, mis à part le fait que cela implique un très bon centrage du robot sur la piste. Néanmoins, ceci peut être comblé par une augmentation de la taille de la porte de capture de la boule. Comme quoi, tout problème a sa solution.

En définitive, on peut apporter une modification à ce qui vient d'être dit : en réalité, aucune des trois solutions n'a été retenue. Après essais et une longue

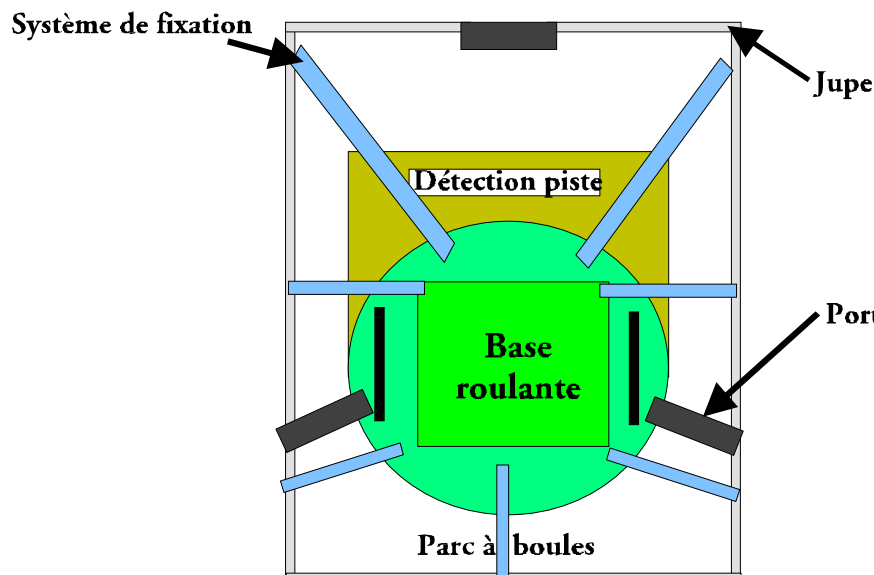


Fig.6 : Dernière idée pour la conception du système garde-boule

réflexion, il s'est avéré que le meilleur compromis est de mixer les solutions deux et trois : le garde-boule du deux, associé à la porte du trois. C'est ce système qui a été réalisé

3.8. Les oreilles : obstacle, je t'entends !

3.8.1. Besoins

Il est nécessaire de détecter des obstacles qui se trouveront disposés sur la piste. Le but est de renvoyer une information au PIC principal concernant la présence ou non d'un obstacle et si tel est le cas sa distance approximative.

3.8.2. Les solutions

Ces quelques contraintes nous ont poussés à créer une carte basée sur le principe du télémètre à ultrasons. En effet la nécessité de renvoyer une information concernant la distance impose le choix d'une solution ultrasonique. La même opération réalisée avec un laser serait possible mais non réalisable dans le cadre de nos contraintes matérielles et financières.

3.8.3. Le principe

Le principe est le suivant : un PIC, que nous appelleront par la suite PIC obstacle, envoie un train d'ondes à un transducteur (émetteur) ultrasonique par le biais d'un commutateur de puissance. La fréquence de ce train d'ondes est judicieusement choisie pour que le transducteur entre en résonance, et envoie ainsi un message qui pourra être réceptionné par un capteur ultrason spécifiquement adapté à cette fréquence. Le message ira donc, après avoir été envoyé par le transducteur, 'rebondir' contre l'obstacle et reviendra vers le robot où il sera perçu grâce au récepteur. Après filtrage et amplification le signal sera analysé par le PIC obstacle. Il suffira alors à celui-ci de compter le temps mis par le message pour faire l'aller et retour et de le diviser par la vitesse du son pour connaître la distance qui sépare

le robot de l'obstacle.

Un schéma sur le principe de fonctionnement du système est présenté en Fig.7.

Il ne lui restera plus alors qu'à renvoyer l'information sur la distance, codée sur quatre bits, via un connecteur, au Pic principal qui agira en conséquence. Notons que les quatre bits à zéro signifient l'absence d'obstacle.

3.8.4. Les pièges

Le fait que nous travaillons avec une fréquence proche de l'audible (20 KHz) nous impose l'utilisation d'un filtre assez sélectif pour ne pas devenir sensible aux bruits d'ambiance qui se propageront autour du robot.

Autre point, lorsque le transducteur entrera en oscillation, c'est toute la carte, y compris le récepteur, qui vibrera à 40 KHz. Il ne faudra donc pas tenir compte des évolutions de celui-ci tout le temps où le train d'ondes sera en cours d'élaboration.

De plus il y a un risque, si l'obstacle est relativement éloigné, qu'un premier train d'ondes revienne juste après qu'un nouveau train d'ondes soit parti, laissant ainsi croire que l'obstacle est très proche. Il conviendra donc de laisser un laps de temps suffisamment long entre l'émission de deux trains d'ondes pour prévenir ce genre de problème.

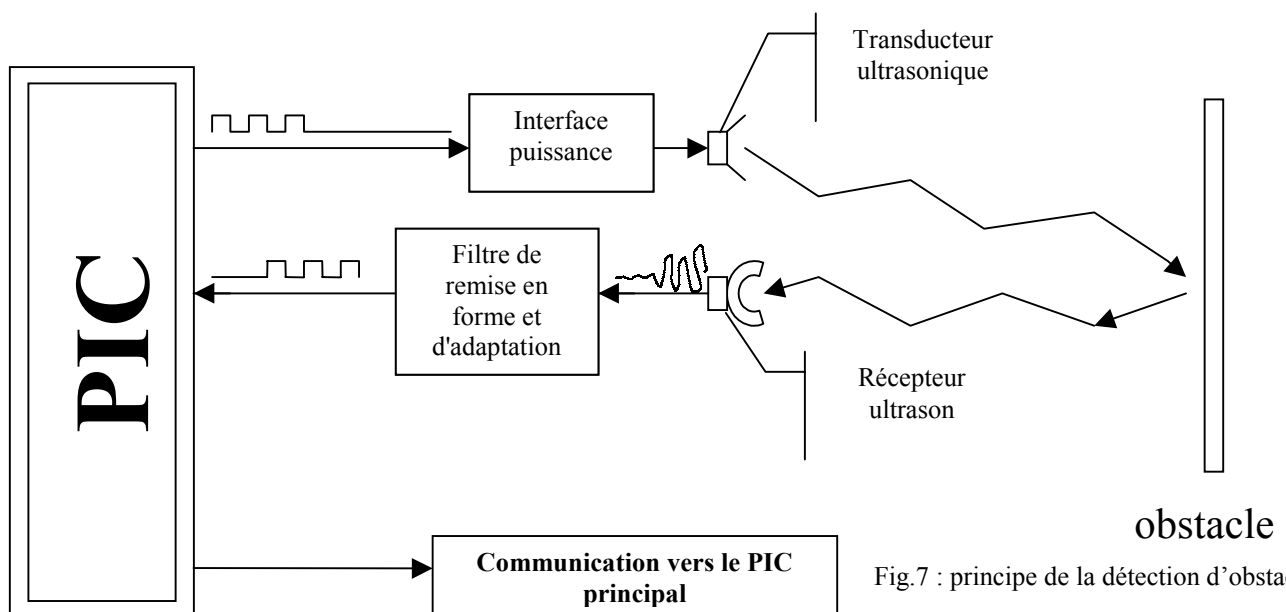


Fig.7 : principe de la détection d'obstacle

4. CONCLUSION

Globalement, nous n'avons pas encore atteint l'objectif que nous nous étions fixé lors du début de ce projet, mais nous avons tout de même avancé.

Au moment où nous rendons ce dossier, nous ne sommes pas arrivés à bout de toutes les tâches qui nous incombent. Ainsi, sur l'ensemble des cartes que nous avons à réaliser, seules la carte PIC principale, la carte de détection de piste et celle de commande des moteurs sont finalisées d'un point de vue conception. La carte de détection d'obstacle est en phase finale de développement, c'est à dire que toute la partie théorique la concernant est faite.

Parallèlement à ce travail sur les cartes, des algorithmes ont été écrits pour le suivi de piste

et la commande des moteurs ainsi que pour la détection d'obstacles. Le programme relatif au premier algorithme est en phase d'adaptation avec la liaison pour la carte de détection d'obstacle.

Il est clair que, d'un point de vue temps, nous avons du retard. Ce retard peut être imputé au fait que nous avons dû remettre en cause la totalité des éléments définissant le projet. En effet, comme cela a déjà été dit, plusieurs semaines après que nous ayons commencé le projet, nous avons subi un changement de règlement qui nous a conduit à choisir une nouvelle catégorie de concours techniquement plus abordable que celle dans laquelle nous étions inscrit avec la précédente plate-forme.

5. REFERENCES

1. Documentation technique du PIC16F87X, Microchip, <http://www.microchip.com/10/lit/PICmicro/16F87X/index.2.htm>
2. Documentation technique du EE-SB5/-SB5-B, Omron, <http://www.omron.com>
3. Documentation technique du L6202, ST Microelectronics, <http://www.st.com/stonline/books/pdf/docs/1373.pdf>