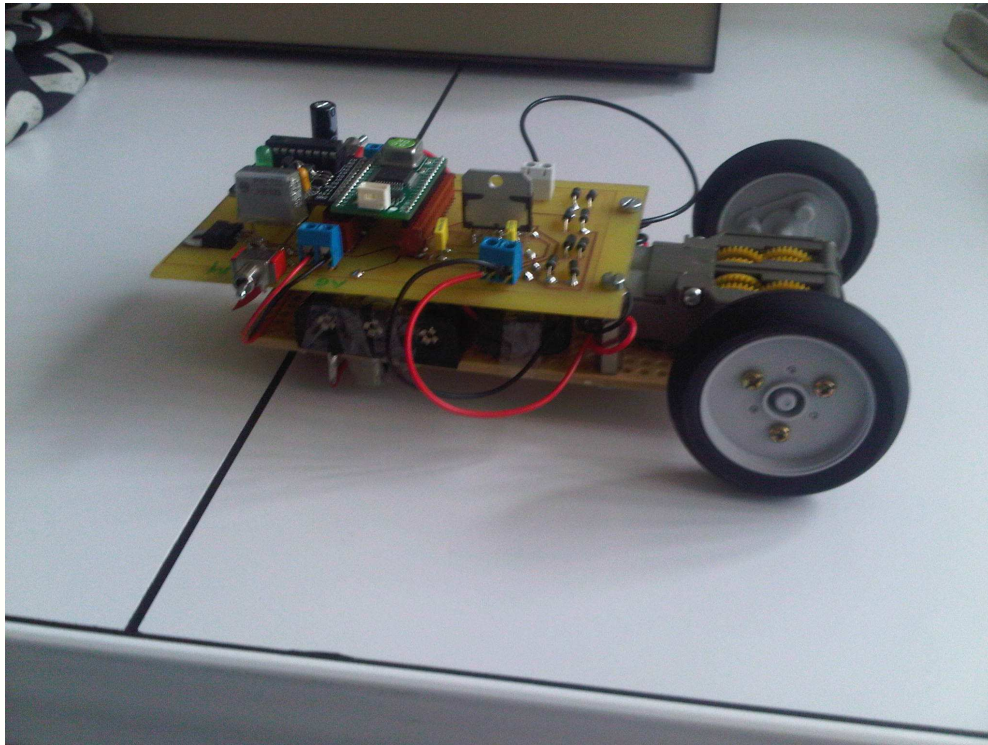


Projet de Physique P6-3
STPI/P6-3/2012 – 6

**Développement de robot à base de module
Lextronic mettant en œuvre la programmation**



Étudiants :

Pierre BERTHIER

Olivier DEBEVEC

Clément ROUSSEAU

Marc ALVAREZ

Julien LEREBOURS

Jonathan LE BLOAS

Enseignant-responsable du projet :

Mounir MESSEDI

Date de remise du rapport : 18/05/2012

Référence du projet : STPI/P6-3/2011 – 6

Intitulé du projet : *Développement de robot à base de module Lextronic mettant en œuvre la programmation.*

Type de projet : *Expérimentation*

Objectifs du projet

L'objectif du projet est le développement d'un robot de la conception à la réalisation. Pour cela, il nous faudra découvrir le monde de la robotique avec le choix des pièces sur internet, le montage du robot, mais aussi la programmation et l'électronique. Pour se faire, nous aurons une période définie de 13 semaines à raison d'1h30 hebdomadaire et devons être en groupe de 6 personnes. Chacun devra travailler pour faire avancer le groupe et le projet.

Mots-clefs du projet :

- *Robotique*
- *Mécanique*
- *Électronique*
- *Programmation*

TABLE DES MATIERES

1. Introduction	1
2. Méthodologie / Organisation du travail	1
3. Travail réalisé et résultats	4
3.1. Le robot et ses composants	4
3.1.1. Partie mécanique	4
3.1.2. Partie électronique	5
3.2. Réalisation du circuit imprimé.....	6
3.3. Programmation.....	7
3.4. Résultats et problèmes rencontrés.....	16
3.4.1. Résultats.....	16
3.4.2. Problèmes rencontrés	19
4. Conclusions et perspectives.....	20
5. Bibliographie	22
6. Annexes.....	23

1. INTRODUCTION

Pendant notre seconde année de cycle préparatoire d'école d'ingénieur à l'insa de Rouen nous réalisons un projet en groupe grâce à la matière P6. Celle-ci nous permet d'appréhender le travail d'équipe pour notre vie professionnelle futur ainsi que de gérer un planning, des délais et surtout de découvrir un aspect plus pratique des sciences.

Une large liste de sujets nous est ainsi proposée et qui recouvre une grande partie des sciences allant des mathématiques appliqués à la chimie en passant par l'informatique ou la physique. Pour notre part le sujet s'intitule : développement de robot à base de module Lextronic mettant en œuvre la programmation. Dans celui-ci nous avons développé un robot de la conception à la réalisation.

Nos à priori sur ce sujet étaient différents selon les personnes. Certains ayant fait SI en Terminale y voyaient un second PPE alors que d'autres venus de SVT voulaient découvrir le monde de l'électronique et de la robotique.

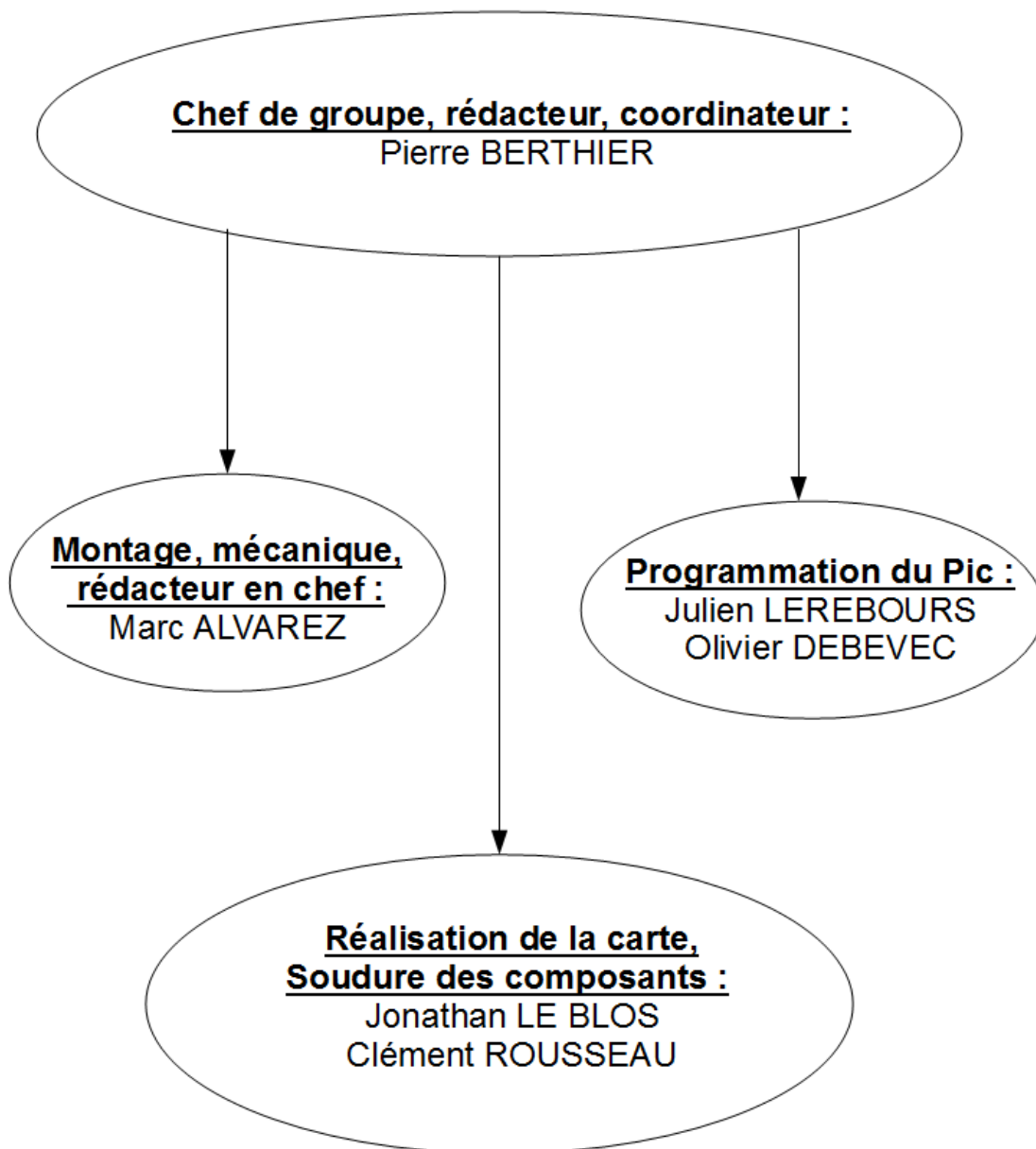
Nous avons ainsi décidé que l'objectif principal serait de faire un robot commandé à distance via une télécommande. Les pièces nous seront fournies par internet sur le site Lextronic, nous aurons à faire le programme, le circuit imprimé et les soudures. Pour mener à bien notre projet nous aurons 1h30 hebdomadaire encadré par notre responsable.

2. METHODOLOGIE / ORGANISATION DU TRAVAIL

Suite au choix du type de robot à concevoir, nous nous sommes répartis le travail en fonction des affinités de chacun sur tel ou tel domaine.

Tout d'abord il nous fallait faire des recherches sur Internet pour connaître le fonctionnement d'une telle machine. Pour cela nous avons pris chacun une partie du robot : Pierre et Marc sur la partie moteur, Olivier et Julien sur le pic et la programmation et Jonathan et Clément sur les autres composants. Pour mettre en commun les recherches, nous avons créé un groupe Facebook. Ainsi chacun pouvait avancer le projet chez lui et en faire profiter les autres.

Une fois les recherches terminées nous avons commandé les pièces sur le site Lextronic.com. Pierre et Marc ont commencé à monter le châssis et le moteur tandis que Olivier et Julien commençait la programmation du pic et Clément et Jonathan réfléchissaient au circuit imprimé. Pierre et Marc ont, par la suite, commencé le rapport et aidé les autres à avancer.



Séance 1	<ul style="list-style-type: none"> - Formation des groupes. - Recherche et commande des pièces pour le robot sur Lextronic. - Recherche des fonctionnalités du robot. Création d'une page Facebook pour faciliter la communication et les échanges de données
Séance 2	<ul style="list-style-type: none"> - Formation des binômes de recherche: Pierre/Marc: le hacheur. Clément/Jonathan: la carte électronique et Julien/Olivier: le pic.
Séance 3	<ul style="list-style-type: none"> - Arrivée des pièces du robot, montage du moteur en équipe et installation sur le châssis. - Nous avons choisis de prendre kicad pour dessiner la carte électronique
Séance 4	<ul style="list-style-type: none"> - Montage des roues sur le moteur et mise en commun des recherches personnelles. - Mise au point d'un schéma et d'un dessin du circuit. - Approche de la programmation du pic en PICBASIC. - Analyse descendante de la programmation. - Recherche des composants manquants.
Séance 5	<ul style="list-style-type: none"> - Commencement des tests du pic à l'aide d'une LED. - Création du schéma du circuit électronique.
Séance 6	<ul style="list-style-type: none"> - Création de la carte du circuit imprimé et programmation avec de nombreux test sur LED. - Suite création du schéma du circuit électronique.
Séance 7	<ul style="list-style-type: none"> - Début rédaction rapport (création du plan). - Avancement dans la programmation. - Fin de la création du schéma du circuit électronique. - Début dessin de la carte électronique (PCV).
Séance 8	<ul style="list-style-type: none"> - Suite dessin de la carte électronique (PCV). - Suite programmation. - Suite rédaction rapport.
Séance 9	<ul style="list-style-type: none"> - Redéfinition des connections entre les composants sur la carte. - Fin dessin de la carte électronique (PCV). - Suite programmation. - Suite rédaction rapport.
Séance 10	<ul style="list-style-type: none"> - Réception de la carte imprimée. - Début du soudage des composants sur la carte. - Suite rédaction rapport. - Fin de la structure globale du code.
Séance 11	<ul style="list-style-type: none"> - Fin soudage. - Fin programmation. - Assemblage de la carte et des autres composants sur le châssis.
Séance 12	<ul style="list-style-type: none"> - Premiers tests sur le robot. -Finition du rapport.
Séance 13	<ul style="list-style-type: none"> - Fin tests avec ajouts de différentes fonctions (créneau, demi-tour, changement de sens). -Fin rapport et début organisation oral.

3. TRAVAIL REALISE ET RESULTATS

Le but de notre projet de P6 était de créer un robot. Notre professeur nous a tout de suite dirigés vers un robot « suiveur de ligne » qui aurait eu pour but de se déplacer en suivant une ligne. Mais après réflexion, nous avons plutôt préféré nous diriger vers la création d'une voiture téléguidée, la plus simple possible. Notre robot peut avancer selon des indications données par un utilisateur. Ces indications sont données par le biais d'une télécommande connectée en infrarouge avec le robot. Notre robot, qui est en fait une voiture téléguidée, peut donc avancer, tourner à droite et à gauche et reculer.

3.1. Le robot et ses composants

Notre robot est donc composé d'une partie mécanique (avec moteur roues..) mais aussi d'une partie électronique (PIC, hacheur, circuit imprimé...). En voici une description des éléments les plus pertinents car nous n'avons pas jugé utile d'expliquer le fonctionnement des résistances et condensateurs.

3.1.1. *Partie mécanique*

Tout d'abord les éléments de la partie mécanique.

I. Le châssis

Voulant nous rapprocher le plus possible d'une « voiture téléguidée », nous avons décidé de prendre un châssis rectangulaire. Nous avons donc choisi un châssis « universel » afin qu'il puisse supporter tous les composants que nous souhaitions y fixer.



Figure 1 : Châssis universel

II. Le bloc moteur

Pour que notre robot puisse tourner, il nous était essentiel d'utiliser un double bloc moteur. Le mécanisme est simple, pour tourner à gauche, le moteur gauche s'arrête et le moteur droit fonctionne, et inversement pour tourner à droite.



Figure 2 : Double bloc moteur

III. Les roues

Lors du choix de notre moteur sur le site www.lextronic.fr, il nous était essentiel de choisir des roues adaptables avec notre moteur. Nous avons donc, parmi les choix proposés par le site, choisi les plus grandes afin de ne pas être gêné par la taille des différents composants présents sur le châssis. Comme le montre la figure 2 ci-dessus, le moteur n'entraîne que les deux roues avant. Afin que l'arrière suive donc le mouvement des roues avant, nous y avons installé une roue dite « folle ».



Figure 3 : Roues arrières



Figure 4 : Roue folle

3.1.2. Partie électronique

Voici la liste des principaux composants électroniques de notre robot.

I. Le PICBASIC



Figure 5 : PICBASIC

Le PICBASIC est l'élément principal de notre carte électronique. C'est lui qui contient le programme et qui fait la liaison entre le récepteur et le reste du circuit. Le programme est saisi sur un PC et transmis au pic par le biais d'un câble USB. Une fois le programme installé sur le PICBASIC, il devient totalement autonome et ne nécessite plus de modifications. En fonction du courant venant du récepteur dans telle ou telle entrée, il fournit un courant de sortie dans les sorties correspondantes. L'attribution de ces sorties est dictée par le programme.

II. Le hacheur

Le hacheur est un élément qui permet de modifier la valeur de la tension d'une source de tension continue avec un rendement plus élevé. Ici, il nous permet d'augmenter le rendement fourni par le PICBASIC pour le transmettre au moteur. En fonction du courant reçu via le PICBASIC il fait tourner les moteurs dans un sens ou dans l'autre, de façon indépendante.



Figure 6 : Hacheur L298

III. Récepteur infrarouge

Le récepteur est le module qui permet de capter le signal infrarouge transmis par la télécommande et qui le traduit par un courant électrique dans le circuit. En fonction du signal capté qui dépend lui-même de la touche enfoncée, le récepteur envoie du courant sur la sortie 1 à 8.

3.2. Réalisation du circuit imprimé

Jonathan et Clément se sont occupés de concevoir la carte électronique. Pour cela il fallait tout d'abord choisir un logiciel permettant de dessiner le schéma électrique de notre robot, ainsi que de dessiner la carte avec le routage. Notre choix s'est logiquement porté sur Kicad, puisqu'il répondait à nos besoins et qu'il était totalement gratuit.

Le premier travail était de répertorier tous les composants qui devaient figurer sur notre carte et de les chercher dans la bibliothèque du logiciel. Pour certaines parties nous nous sommes basés sur des modèles relativement proches. En effet un schéma comme celui de l'alimentation avec le régulateur de tension LM7805 est tout à fait classique pour un robot de cette taille. Il fallait aussi définir les liaisons avec le PIC suivant nos besoins et les contraintes liés à la programmation.

Une fois tous les composants téléchargés, nous avons dessiné le schéma électrique de la carte en reliant les différents composants.

Avec tous nos composants sur le schéma, nous avons pu réaliser la net-list qui contient l'ensemble des composants à placer sur la carte électronique ainsi que leurs caractéristiques, par exemple la distance entre chaque broches, puis ceux-ci ont été chargés sur une feuille de dessin du logiciel.

Nous devons placer tous les composants dans un cadre que nous devons définir en fonction de la taille réelle de la future carte. Nous avons ensuite réalisé le routage des composants. Chaque connexion entre composants devait soit être réalisée du côté cuivre(en vert sur le logiciel), soit du côté composant(en rouge sur le logiciel) et nous disposions des via qui permettent de changer de côté. Il fallait privilégier le côté cuivre pour des raisons esthétiques et pratiques avec le soudage.

Lorsque nous avons reçu la carte, imprimée par notre enseignant, nous avons pu commencer le soudage des composants sur la carte. Une fois tous les composants installés

et soudés, nous avons effectué des tests pour vérifier si toutes les connexions étaient bonnes avec un ohmmètre qui sonnait lorsqu'il y avait une liaison électrique voulue ou non. Une fois toutes les vérifications effectuées et corrections apportées si besoin était, la dernière étape était de fixer la carte sur le robot ainsi que de la relier aux piles.

La liste des composants, le tracé des pistes ainsi que le schéma du circuit imprimé sont en annexe.

3.3. Programmation

Description des variables:

Le but de notre robot est de pouvoir se déplacer à l'aide d'une télécommande. Il pourra aller en avant, en arrière et tourner soit à droite soit à gauche. Pour cela nous appuyons sur différents boutons de la télécommande:

- le bouton 2 pour aller en avant
- le bouton 8 pour aller en arrière
- le bouton 4 pour aller à gauche
- le bouton 6 pour aller à droite
- le bouton 5 pour stopper le robot
- le bouton 1 pour changer de direction (avant/arrière)
- le bouton 3 pour faire un demi-tour
- le bouton 7 pour faire un créneau

Le récepteur ensuite envoie ou non du courant vers le pic. Il envoie du courant par la sortie D0 si on appuie sur la touche 1 de la télécommande, D1 si on appuie sur le bouton 2 jusqu'à envoyer du courant par la sortie D7 si on appuie sur la touche 8 de la télécommande. Ainsi les sorties respectives des commandes précédentes sont D1, D7, D3, D5, D4, D0, D2 et D6.

Le récepteur est relié au pic qui les transmet les informations. Les ports sont reliés aux sorties du récepteur de la façon suivante:

- le port 22 est relié à la sortie D1
- le port 20 est relié à la sortie D3
- le port 18 est relié à la sortie D5
- le port 16 est relié à la sortie D7
- le port 19 est relié à la sortie D4
- le port 23 est relié à la sortie D0
- le port 21 est relié à la sortie D2
- le port 17 est relié à la sortie D6

Ensuite nous envoyons des informations aux sorties 1 à 6.

Paramètres de configuration :

DIM : Le premier outil de programmation utilisé est: dim. Il permet de créer une variable avec le type que l'on désire. Par exemple: dim boutonAvant as integer permet de créer la variable boutonAvant qui sera du type entier. Le langage du pic basic ne fait pas la différence entre les mots avec ou sans majuscule (non sensible à la casse), cependant pour une meilleure compréhension nous avons préféré mettre une majuscule au début du deuxième mot.

IN : Le second outil de programmation utilisé est: in. Cet outil permet de récupérer la valeur 0 ou 1 d'un port du pic et de le mettre dans une variable. Prenons l'exemple suivant: boutonAvant = in(22). Ainsi nous regardons si le bouton avant est activé ou non, s'il est activé il met une valeur dans la variable boutonAvant , sinon cette variable reste vide.

OUT : Nous utilisons aussi la commande out qui permet de faire passer du courant ou non dans un port de sortie. Par exemple la commande "out 5,1" fera passer du courant par le port 5 et "out 5,0" ne le fera plus passer.

If ...then : L'outil suivant est if...then. Cela permet d'exécuter ou non la suite du code. Comme nous pouvons le voir dans cette exemple: "if boutonAvant >0 then goto avant", si la variable boutonAvant est non nulle alors le programme exécutera le code situé après le then, sinon il passera à la ligne en dessous.

Goto : De plus nous pouvons voir dans cet exemple l'outil goto. Ceci permet de faire un saut dans le programme et d'aller soit à la ligne de code numéro 10 si nous avons goto 10 soit au début d'une boucle repérée par mot-clé : si on a goto mot-clé. Dans l'exemple précédent nous avons goto avant, ce qui nous permet d'aller à la ligne de code juste après "avant:".

DELAY : Pour finir nous utilisons la commande delay qui permet de faire une pause dans le programme. Cette pause exprimée en milliseconde peut s'étendre de 1 à 65535 ms. Par exemple la commande "Delay 1000" fera une pause de 1000 ms lors de l'exécution du programme.

Analyse des cas possibles :

Nous avons étudié les cas suivants :

- le robot est à l'arrêt et il le reste.
- le robot est à l'arrêt et il va à gauche/droite.
- le robot est à l'arrêt et il va en avant puis il tourne à droite ou à gauche.
- le robot est à l'arrêt et il va en arrière puis il tourne à droite ou à gauche.
 - le robot change de direction, fait demi-tour ou fait un créneau.

Tout d'abord, étudions le code suivant:

Boucle:

```
boutonAvant = in(22)
boutonGauche = in(20)
boutonDroit = in(18)
boutonArriere = in(16)
boutonArret=in(19)
```

```
if boutonAvant >0 then goto avant
if boutonGauche>0 then goto gauche
if boutonDroit>0 then goto droite
if boutonArriere>0 then goto arriere else goto boucle
```

Ici nous sommes au début de notre programme, le robot ne bouge pas et il attend les instructions pour bouger. En effet tant que nous n'appuyons pas sur un des 4 boutons, le robot reste à l'arrêt.

Étudions dans un premier temps ce qu'il se passe si nous appuyons sur le bouton droit. Si nous appuyons dessus, le port 18 prend une valeur qui la met dans la variable boutonDroit via la commande : "boutonDroit = in(18)". Ensuite comme les variables boutonAvant et boutonGauche sont nulles nous passons à la ligne de code "if boutonDroit>0 then goto droite" . Comme la variable boutonDroit n'est pas nulle, nous sautons à la boucle "droite" :

```
droite :
out 5,1
out 2,1
goto ContinueDroite
```

Afin d'aller à droite nous activons le moteur gauche dans le sens avant puis nous sautons au code de la boucle "ContinueDroite:"

```
ContinueDroite:
boutonDroit = in(18)
if boutonDroit=0 then goto arret else goto ContinueDroite
```

Nous regardons maintenant la valeur de la variable boutonDroit, si elle est égale à 0 nous à la boucle "stop" sinon elle retourne dans ContinueDroit. Cela permet au robot de continuer à aller à droite tant que nous appuyons sur le bouton 6. Des que nous le lâchons, nous allons dans "stop:" qui permet de remettre tout à 0 et de stopper le robot. Après cela, nous retourner dans boucle, afin de recevoir les nouvelles instructions de la télécommande.

Nous pouvons remarquer que cela est similaire si on avait appuyé sur le bouton gauche.

Cependant si nous appuyons sur le bouton avant ou arrière cela est un peu différent . Prenons le cas où nous nous appuyons sur le bouton 2 de la télécommande, alors boutonAvant prend une valeur et nous allons dans:

```
avant:
out 5,1
out 6,1
out 4,1
out 2,1
goto continueAvant
```

Nous voyons que nous activons le moteur de droite et de gauche afin d'aller en avant. En effet le port 5 représente le moteur de gauche et 6 celui de droite. 2 et 4 représente mutuellement le fonctionnement de la marche avant du moteur gauche et droite. Pour l'arrière c'est respectivement 1 et 3. Après cette activation, nous allons dans la boucle "continueAvant":

```
continueAvant:
```

```

boutonGauche = in(20)
boutonDroit = in(18)
boutonArret=in(19)
if boutonArret>0 then goto arret
if boutonGauche>0 then goto AvantGauche
if boutonDroit>0 then goto AvantDroite else goto continueAvant

```

Le robot va alors tout droit et nous avons 3 commandes disponibles pour lui faire changer de direction. Notons que nous pouvons lâcher le bouton 2, si nous appuyons dessus cela n'aura aucune influence et si nous n'appuyons sur aucun bouton, il retournera juste à la ligne continueAvant encore et encore. Nous pouvons appuyer sur le boutons 4 qui lui fera aller à gauche, 6 pour la droite et 5 pour qu'il s'arrête.

Nous restons donc dans cette boucle tant que l'arrêt n'est pas demandé (bouton 5) et le robot continue d'avancer. A noter que le code pour aller à gauche ou à droite depuis cette boucle est similaire au code ayant la même fonction depuis la boucle principale, la différence étant simplement le retour dans la boucle "continueAvant" après l'appui sur les boutons gauche ou droite afin que le robot continue d'avancer. Si on appui sur arrêt les moteurs sont arrêtés et on retourne dans la boucle principale.

Une autre boucle intéressante :

```

continueArret:
boutonAvant = in(22)
boutonGauche = in(20)
boutonDroit = in(18)
boutonArriere = in(16)
if boutonAvant >0 then goto continueArret
if boutonGauche>0 then goto continueArret
if boutonDroit>0 then goto continueArret
if boutonArriere>0 then goto continueArret else goto boucle

```

Toutes ces boucles nous permettent d'éviter les mauvaises manipulations lorsque nous appuyons sur plusieurs boutons à la fois. Par exemple, ici afin de pouvoir à nouveau envoyer une instruction au robot, nous devons tout d'abord lâcher tous les boutons de la télécommande.

La marche arrière est similaire à la marche avant. Nous devons appuyer une seule fois sur le 8 pour aller en arrière puis nous pouvons aller à gauche ou à droite en restant appuyé sur les boutons 4 ou 6. Enfin pour le stopper nous appuyons sur le bouton 5.

Toutes les fonctions décrites ci-dessus étant selon nous des fonctions de base, nous avons choisi d'ajouter des fonctions un peu plus gadget : inverseAvant, InverseArriere, DemiTourAvant, demiTourArriere et creneau. Ainsi avec inverseAvant/InverseArriere, le robot lorsqu'il est dans la boucle avant ou arrière, peut changer directement de marche avant à marche arrière ou inversement. Par exemple s'il va en marche avant, il ira alors en marche arrière. Avec demiTourAvant/demiTourArriere il fera un tour sur lui même de 180° et reprendra sa course. Pour terminer, la fonction creneau permet de faire un créneau lorsque le robot est dans la boucle principale.

Le programme:

```

dim boutonAvant as integer #correspond à la sortie d1
dim boutonGauche as integer #correspond à la sortie d3

```

```
dim boutonDroite as integer #correspond à la sortie d5
dim boutonArriere as integer #correspond à la sortie d7
dim boutonArret as integer #correspond à la sortie d4
dim Boutoninverse as integer #correspond à la sortie d0
dim BoutonDemiTour as integer #correspond à la sortie d3
dim BoutonCreneau as integer #correspond à la sortie d7
```

```
#la boucle principale
```

```
boucle:
```

```
#nous regardons s'il y a des modifications aux entrés
```

```
boutonAvant = in(22)
```

```
boutonGauche = in(20)
```

```
boutonDroit = in(18)
```

```
boutonArriere = in(16)
```

```
boutonArret=in(19)
```

```
boutonCreneau=in(17)
```

```
#si les variables changent, alors nous allons dans la fonction dont la variable correspond
```

```
if boutonAvant >0 then goto avant #si le bouton 2 est appuyé, le programme va à la fonction avant
```

```
if boutonGauche>0 then goto gauche #si le bouton 4 est appuyé, le programme va à la fonction gauche
```

```
if boutonDroit>0 then goto droite #si le bouton 6 est appuyé, le programme va à la fonction droite
```

```
if boutonCreneau>0 then goto creneau #si le bouton 7 est appuyé, le programme va à la fonction creneau
```

```
if boutonArriere>0 then goto arriéré else goto boucle #si le bouton 8 est appuyé, le programme va à la fonction arriéré
```

```
#sinon le programme va à boucle et recommence
```

```
#la fonction arrêt, nous mettons toutes les variables de sorties sur 0
```

```
arret:
```

```
out 5,0
```

```
out 6,0
```

```
out 4,0
```

```
out 2,0
```

```
out 1,0
```

```
out 3,0
```

```
goto continueArret #le programme va à la boucle continueArret
```

```
#la boucle continueArret, nous regardons si un bouton est encore appuyé
```

```
continueArret:
```

```
boutonAvant = in(22)
```

```
boutonGauche = in(20)
```

```
boutonDroit = in(18)
```

```
boutonArriere = in(16)
```

```
boutonArret=in(19)
```

```
if boutonArret>0 then goto continueArret
```

```
if boutonAvant >0 then goto continueArret
```

```
if boutonGauche>0 then goto continueArret
```

```
if boutonDroit>0 then goto continueArret
```

```
if boutonArriere>0 then goto continueArret else goto boucle #si nous relâchons tous les boutons nous retournons dans la boucle principale, sinon nous retournons dans la boucle continueArret
```

```
#la fonction gauche
```

```
gauche:
```

```
#nous allumons le moteur droit
```

```
out 6,1
```

```
out 4,1
```

```
goto ContinueGauche #on va à la boucle ContinueGauche
```

```
#la boucle ContinueGauche , nous regardons si nous appuyons toujours sur 4
```

```
ContinueGauche:
```

```
boutonGauche = in(20)
```

```
if boutonGauche=0 then goto stop else goto ContinueGauche #si on le relâche, nous allons dans la fonction stop
```

```
#la fonction droite
```

```
droite:
```

```
#nous allumons le moteur gauche
```

```
out 5,1
```

```
out 2,1
```

```
goto ContinueDroite # on va à la boucle ContinueDroite
```

```
#la boucle ContinueDroite, nous regardons si nous appuyons toujours sur 6
```

```
ContinueDroite:
```

```
boutonDroit = in(18)
```

```
if boutonDroit=0 then goto stop else goto ContinueDroite #si on le relâche, nous allons dans la fonction stop
```

```
#la fonction stop
```

```
stop:
```

```
out 5,0
```

```
out 6,0
```

```
out 4,0
```

```
out 2,0
```

```
goto boucle #nous retournons dans la boucle principale
```

```
#la fonction arriere
```

```
arriere:
```

```
out 5,1
```

```
out 6,1
```

```
out 1,1
```

```
out 3,1
```

```
goto continueArriere # on va à la boucle continueArriere
```

```
#la boucle continueArriere,nous regardons si on appuie sur les boutons 4, 6 ou 5
```

```
continueArriere:
```

```
boutonGauche = in(20)
```

```
boutonDroit = in(18)
```

```
boutonArret=in(19)
```

```
boutonInverse=in(23)
```

```
boutonDemiTour=in(21)
```



```

if boutonArret>0 then goto arret # si on appuie sur 5, on va à la fonction arret
if boutonInverse>0 then goto inverseAvant # si on appuie sur 1, on va à la fonction
inverseAvant
if boutonDemiTour>0 then goto demiTourArriere # si on appuie sur 3, on va à la fonction
demiTourArriere
if boutonGauche>0 then goto ArriereGauche# si on appuie sur 4, on va à la fonction
ArriereGauche
if boutonDroit>0 then goto ArriereDroite else goto continueArriere # si on appuie sur 6, on va
à la fonction ArriereDroite
#sinon si on appuie sur aucun bouton, nous restons dans cette boucle

```

#la fonction ArriereGauche

ArriereGauche:

#nous coupons le moteur de gauche

out 6,1

out 3,1

out 5,0

out 1,0

goto ContinueArriereGauche # on va dans la boucle ContinueArriereGauche

#la boucle ContinueArriereGauche, on regarde si on appuie toujours sur le bouton 4

ContinueArriereGauche:

boutonGauche = in(20)

if boutonGauche=0 then goto arriere else goto ContinueArriereGauche # si on relâche le
bouton 4, nous allons dans la fonction arriere

#la fonction ArriereDroite

ArriereDroite:

#nous coupons le moteur de droite

out 6,0

out 3,0

out 5,1

out 1,1

goto ContinueArriereDroite # on va dans la boucle ContinueArriereDroite

#la boucle ContinueArriereDroite, on regarde si on appuie toujours sur le bouton 6

ContinueArriereDroite:

boutonDroit = in(18)

if boutonDroit=0 then goto arriere else goto ContinueArriereDroite # si on relâche le bouton
6, nous allons dans la fonction arriere

#la fonction avant

avant:

out 5,1

out 6,1

out 4,1

out 2,1

goto continueAvant # on va dans la boucle continueAvant

la boucle continueAvant,nous regardons si on appuie sur les boutons 4, 6 ou 5

continueAvant:

boutonGauche = in(20)

```

boutonDroit = in(18)
boutonArret=in(19)
boutonInverse=in(23)
boutonDemiTour=in(21)
if boutonArret>0 then goto arret # si on appuie sur 5, on va à la fonction arret
if boutonInverse>0 then goto inverseArriere # si on appuie sur 1, on va à la fonction
inverseArriere
if boutonDemiTour>0 then goto demiTourAvant # si on appuie sur 3, on va à la fonction
demiTourAvant
if boutonGauche>0 then goto AvantGauche # si on appuie sur 4, on va à la fonction
AvantGauche
if boutonDroit>0 then goto AvantDroite else goto continueAvant # si on appuie sur 6, on va à
la fonction AvantDroite
#sinon si on appuie sur aucun bouton, nous restons dans cette boucle

#la fonction AvantGauche
AvantGauche:
#nous coupons le moteur de gauche
out 6,1
out 4,1
out 5,0
out 2,0
goto ContinueAvantGauche # on va dans la boucle ContinueAvantGauche

#la boucle ContinueAvantGauche, on regarde si on appuie toujours sur le bouton 4
ContinueAvantGauche:
boutonGauche = in(20)
if boutonGauche=0 then goto avant else goto ContinueAvantGauche # si on relâche le
bouton 4, nous allons dans la fonction avant

#la fonction AvantDroite
AvantDroite:
#nous coupons le moteur de droite
out 6,0
out 4,0
out 5,1
out 2,1
goto ContinueAvantDroite # on va dans la boucle ContinueAvantDroite

#la boucle ContinueAvantDroite, on regarde si on appuie toujours sur le bouton 6
ContinueAvantDroite:
boutonDroit = in(18)
if boutonDroit =0 then goto avant else goto ContinueAvantDroite # si on relâche le bouton 6,
nous allons dans la fonction avant

# la fonction inverseAvant, comme le robot va en arriere, on coupe tous les moteurs et on les
allume en marche avant
inverseAvant:
out 5,0
out 6,0
out 4,0
out 2,0
out 1,0

```

```

out 3,0
out 5,1
out 6,1
out 4,1
out 2,1
goto continueInverseAvant #on va dans la boucle continueInverseAvant

#la boucle continueInverseAvant, nous regardons si on appuie toujours sur le bouton 1
continueInverseAvant:
boutonInverse=in(23)
if boutonInverse =0 then goto avant else goto continueInverseAvant #si le bouton 1 est
relâché, nous allons dans la fonction avant

goto avant #on va à la fonction avant

# la fonction inverseArriere, comme le robot va en arrière, on coupe tous les moteurs et on
les allume en marche arrière
inverseArriere:
out 5,0
out 6,0
out 4,0
out 2,0
out 1,0
out 3,0
out 5,1
out 6,1
out 1,1
out 3,1
goto continueInverseArriere #on va dans la boucle continueInverseArriere

#la boucle continueInverseArriere, nous regardons si on appuie toujours sur le bouton 1
continueInverseArriere:
boutonInverse=in(23)
if boutonInverse =0 then goto arriere else goto continueInverseArriere si le bouton 1 est
relâché, nous allons dans la fonction arriere

#la fonction demiTourArriere, nous coupons le moteur de gauche afin de faire un demi-tour
demiTourArriere:
out 6,1
out 3,1
out 5,0
out 1,0
delay 2400 # nous attendons que le demi tour soit finis
goto arriere #nous allons à la fonction arriere

#la fonction demiTourAvant, nous coupons le moteur de gauche afin de faire un demi-tour
demiTourAvant:
out 6,1
out 4,1
out 5,0
out 2,0
delay 2400 # nous attendons que le demi tour soit finis
goto avant #nous allons à la fonction avant

```

```

# la fonction creneau
creneau:
#nous allumons le moteur de droite afin d aller à gauche
out 6,1
out 4,1
#nous attendons afin d être dans la bonne position
delay 500
# nous coupons le moteur de droite et nous allumons celui de gauche et de droite en mode
marche arrière
out 6,0
out 4,0
delay 300
out 6,1
out 5,1
out 1,1
out 3,1
# nous attendons que le robot soit dans la bonne position
delay 2000
# nous coupons les moteurs et mettons celui de gauche en mode avant
out 6,0
out 5,0
out 1,0
out 3,0
delay 300
out 5,1
out 2,1
# nous attendons que le robot soit dans la bonne position
delay 500
#nous coupons le moteur gauche
out 5,0
out 2,0
delay 300
# nous allumons celui de gauche et de droite en mode marche avant
out 5,1
out 6,1
out 4,1
out 2,1
delay 200
out 5,0
out 6,0
out 4,0
out 2,0

goto boucle # nous retournons dans la boucle principale

```

3.4. Résultats et problèmes rencontrés

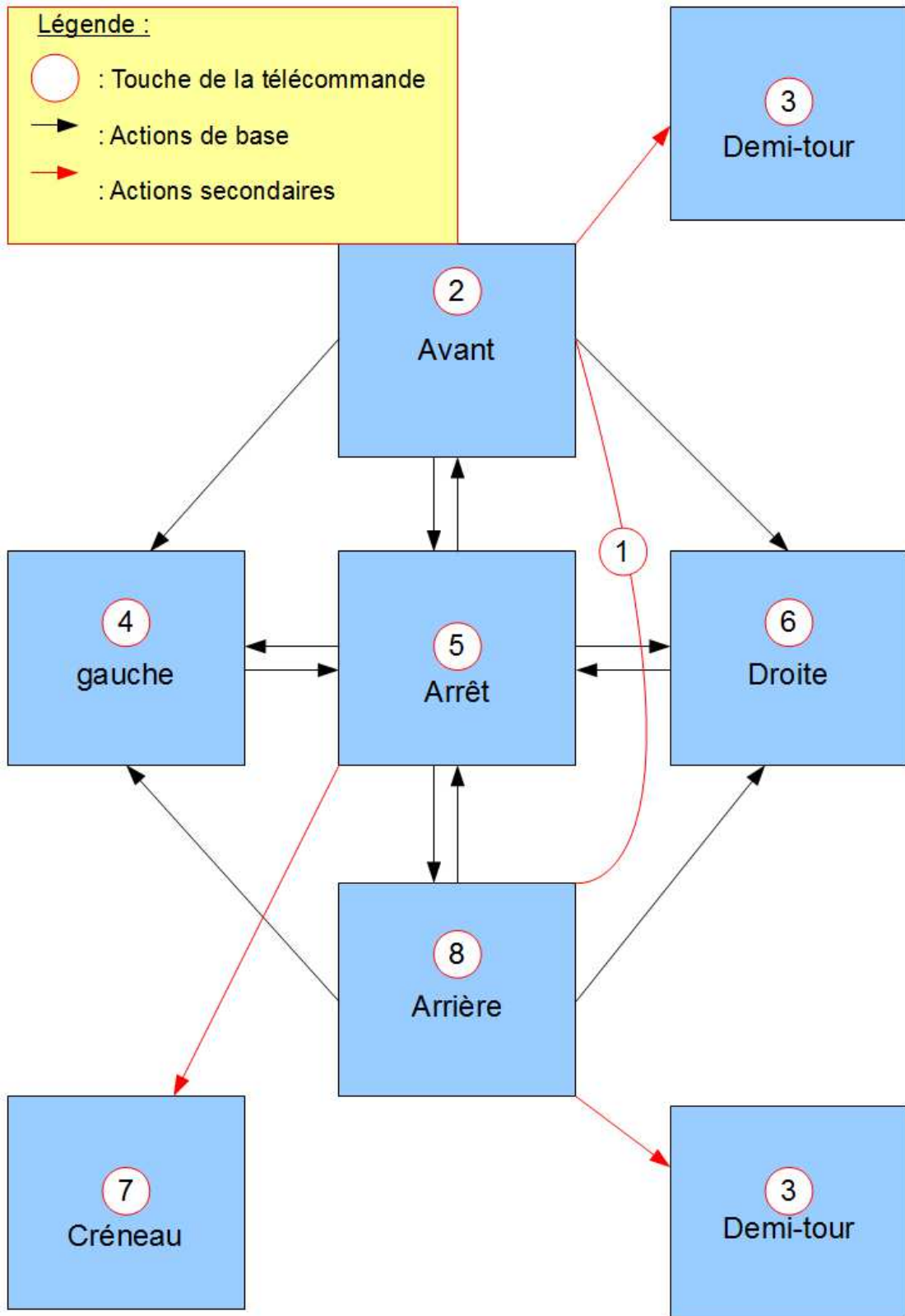
3.4.1. Résultats

Tout d'abord nous avons fait des tests pour le programme sur une plateforme de LED pour vérifier le bon fonctionnement de celui-ci. Ces tests se sont avérés positifs et de bonne

augure pour la suite du projet. De plus, les tests sur le circuit imprimé n'ont pas révélé de court circuit ce qui nous a permis d'avancer vite de ce point de vue.

Ensuite, une fois le robot prêt et le programme mis dans le PICBASIC, nous avons fait un test global au robot : marche avant, marche arrière, gauche, droite, stop. Toutes les commandes ont bien répondu. Le seul petit problème était la détection du signal qui n'était pas optimale.

Cette réussite nous a poussé à ajouter d'autres commandes tels que le changement de sens avant-arrière immédiat où on était avant obligé de passer par stop, un demi-tour et un créneau. Nous avons aussi prévu de faire un parcours autour de cône mais nous n'avons pas eu le temps nécessaire.



3.4.2. Problèmes rencontrés

Lors des séances nous avons dû faire face à des problèmes de natures différentes et les résoudre pour permettre au projet d'avancer.

Tout d'abord, le travail en équipe implique un bon esprit de groupe et une entre-aide entre ses membres ce que nous avons réussi à instaurer dès le début du projet. Malgré des points de vue différents, nous nous sommes écoutés et avons toujours essayé de faire au mieux pour satisfaire tout le monde. De plus, la présence d'un « chef » aurait pu créer des conflits dans le groupe mais il a su rester humble et servir de coordinateur et d'interlocuteur avec le professeur.

Ensuite, pour le montage du moteur, nous avons deux choix, roues qui tournent vite mais moteurs dépendants ou roues qui tournent moins vite mais indépendantes. Finalement, pour pouvoir tourner, nous avons choisis la deuxième possibilité.

En outre, lors du montage, nous n'avions pas les outils les plus adaptés et hélas, Marc a détruit avec fougue un tournevis lors du serrage des vis. De même au niveau matériel informatique, nous n'arrivions pas à connecter l'ordinateur au pic basic. En effet nous n'arrivions pas à installer pic basic studio sur cet ordinateur or nous ne pouvions pas faire sans. Finalement les restrictions de l'INSA ont été levées et nous avons pu le faire.

Pour le circuit imprimé, nous avons dû faire face à plusieurs difficultés lors de la réalisation de la carte. Tout d'abord, ce genre de démarche était plutôt nouveau pour nous, bien que nous ayons tous les deux suivi une formation de sciences de l'ingénieur (SI) au lycée; il a donc fallu découvrir petit à petit la démarche à suivre en nous informant sur internet. Cela nous a pris du temps et nous a conduits à réaliser une partie du travail chez nous.

Ensuite, les possibilités de Kicad étant limitées, plusieurs composants ne figuraient pas dans la bibliothèque du logiciel. Heureusement, le logiciel permet de créer soit même certains composants et nous avons réussi à trouver sur internet les réalisations d'utilisateurs qui avaient déjà créé les composants dont nous avons besoin comme le PIC PB2H et le récepteur infra rouge, que nous avons achetés.

Lors du routage, nous nous sommes rendu compte que les connexions entre le PIC et le hacheur étaient entremêlées et qu'il serait difficile de faire le routage. Cela s'est confirmé par le fait que le routage automatique proposé par le logiciel ne fonctionnait pas; nous avons donc dû faire le routage à la main en essayant de résoudre les problèmes inhérents à ces entremêlements.

Nous ne savions pas lors du routage qu'il fallait privilégier les liaisons sur le côté cuivre, ainsi plusieurs connexions ont été dessinées sur le côté composant alors qu'il aurait été possible de les faire du côté cuivre. Il fallait ainsi faire attention à ne pas endommager la carte ainsi que les composants lors des soudures. Nous avons donc fait une seconde carte au cas où les soudures rendues difficiles par cette erreur auraient empêché le robot de fonctionner.

4. CONCLUSIONS ET PERSPECTIVES

Pour conclure nous pouvons dire que nous sommes tous très satisfait du travail que nous avons mené ce semestre dans l'EC P6. En particulier nous sommes fiers d'avoir conçu un robot télécommandé alors qu'aucun de nous n'en aurait été capable avant. Au vu des résultats finaux nous sommes satisfaits de notre robot même si plein d'autres idées d'améliorations n'ont pu voir le jour par manque de temps. Par exemple nous aurions voulu concevoir une coque avec des jeux de lumières qui auraient rendu notre robot plus design, ou encore des programmes complets de circuits suivant des cônes.

Jonathan :

Pour ma part ce projet ma beaucoup apporté car j'ai pu apprendre à concevoir une carte électronique du début jusqu'à la fin bout excepté l'impression de celle-ci, de plus ce fut l'occasion d'apprendre à utiliser kicad un logiciel gratuit de CAO. Cette expérience me permettra également de ne pas commettre les même erreurs la prochaine fois, par exemple avec les pistes du mauvais côté. De plus je pense peut être m'investir dans l'association robotique de l'INSA, l'ARI, l'année prochaine ce qui fait de ce projet l'occasion parfaite pour me perfectionner dans ce domaine. Enfin ce projet m'a également permis d'encore mieux aborder la notion de répartition des tâches que l'on avait déjà vécu avec les autres projets notamment avec le projet informatique.

Clément :

Ce projet m'a permis premièrement de découvrir un nouveau logiciel: Kicad, qui n'est pas utilisé dans le monde professionnel mais qui répondait à nos besoins. J'ai surtout découvert le monde de la robotique qui m'a toujours attiré mais je n'avais jamais suivi les étapes de la conception d'un robot de la définition de son utilité aux tests pour vérifier qu'il répond à nos besoins en passant par la programmation, la conception de la carte électronique et l'assemblage du châssis et du moteur. Enfin, au fil de ce travail, j'ai réalisé que l'on peut travailler à plusieurs, indépendamment les uns des autres sur une partie d'un projet pour aboutir à un résultat dans lequel les différentes tâches sont indispensables et se recourent.

Pierre :

Tout d'abord, ce projet m'a permis de découvrir le monde de la robotique qui m'était jusqu'alors complètement inconnu du point de vu technique car j'avais déjà pensé un jour à essayer de construire un robot mais sans réelle motivation. C'est pourquoi, dès que j'ai vu l'intitulé du projet je me suis jeté sur l'occasion. J'ai ainsi pu découvrir chacune des facettes de ce domaine : de la conception sur ordinateur à la programmation en passant par le choix des éléments et le soudage. En t'en que chef de groupe désigné lors de la première séance, j'ai pris plaisir à tenir mon rôle et mes obligations envers notre professeur référent. Je pense m'en être bien sorti et que mes camarades ne m'ont pas trop maudits. Ainsi, cela m'a permis de comprendre le rôle de manager pour mon futur métier.

Julien :

Pour conclure ce projet je dirais tout d'abord que nous commençons à être habitués aux projets de groupe ce qui nous a permis d'être plutôt efficaces dans la gestion des tâches désignation d'un chef dès la première séance qui a pris les choses en main au niveau de la distribution du travail : la répartition a été assez simple car ce projet pouvait aisément être découpé en plusieurs parts de travail : parties mécanique, électronique et informatique. Le projet a donc suivi son cours sans obstacle particulier bien que toutes les parties citées ci-dessus nécessitaient beaucoup de travail car la plupart des notions nous étaient inconnues

Pour ma part j'ai fait équipe avec Olivier sur la programmation du PICBASIC. Cette part du travail m'intéressait fortement car j'ai pour objectif le département ASI en fin de semestre. Ainsi j'ai pu récolter quelques notions supplémentaires en relation directe avec ce domaine que ce soit au niveau de la programmation en BASIC ou encore du fonctionnement du PIC. J'aimerais également souligner le fait que ce projet était particulièrement intéressant car nous avons jusque là mené à bien des projets tous théoriques. Or ici nous avons eu la satisfaction de pouvoir utiliser notre robot et réellement voir le fruit de notre travail.

Olivier :

Construire un robot de A à Z m'a tout d'abord apporté beaucoup de nouvelles connaissances. En effet, étant chargé de la programmation du pic avec Julien, j'ai obtenu des bases en PicBasic. Ce paradigme de programmation est très différent du pascal ou du C qu'on a l'habitude de voir, il ne fait pas appelle à des fonctions mais fait des sauts dans le programme, ce qu'on n'a pas l'habitude de voir. De plus la programmation d'un robot est une chose très différente que de concevoir un programme qui calcule une aire par exemple. Nous pouvons interagir avec lui, ce qui est plus simple pour voir les anomalies. Pour conclure ce projet n'est qu'un plus pour notre parcours d'ingénieur, car il nous a appris à concevoir un projet sans avoir de pré-connaissance ou un professeur qui nous tient la main. Ainsi nous avons été obligés de faire des recherches par nous même et faire un réel travail d'équipe.

Marc :

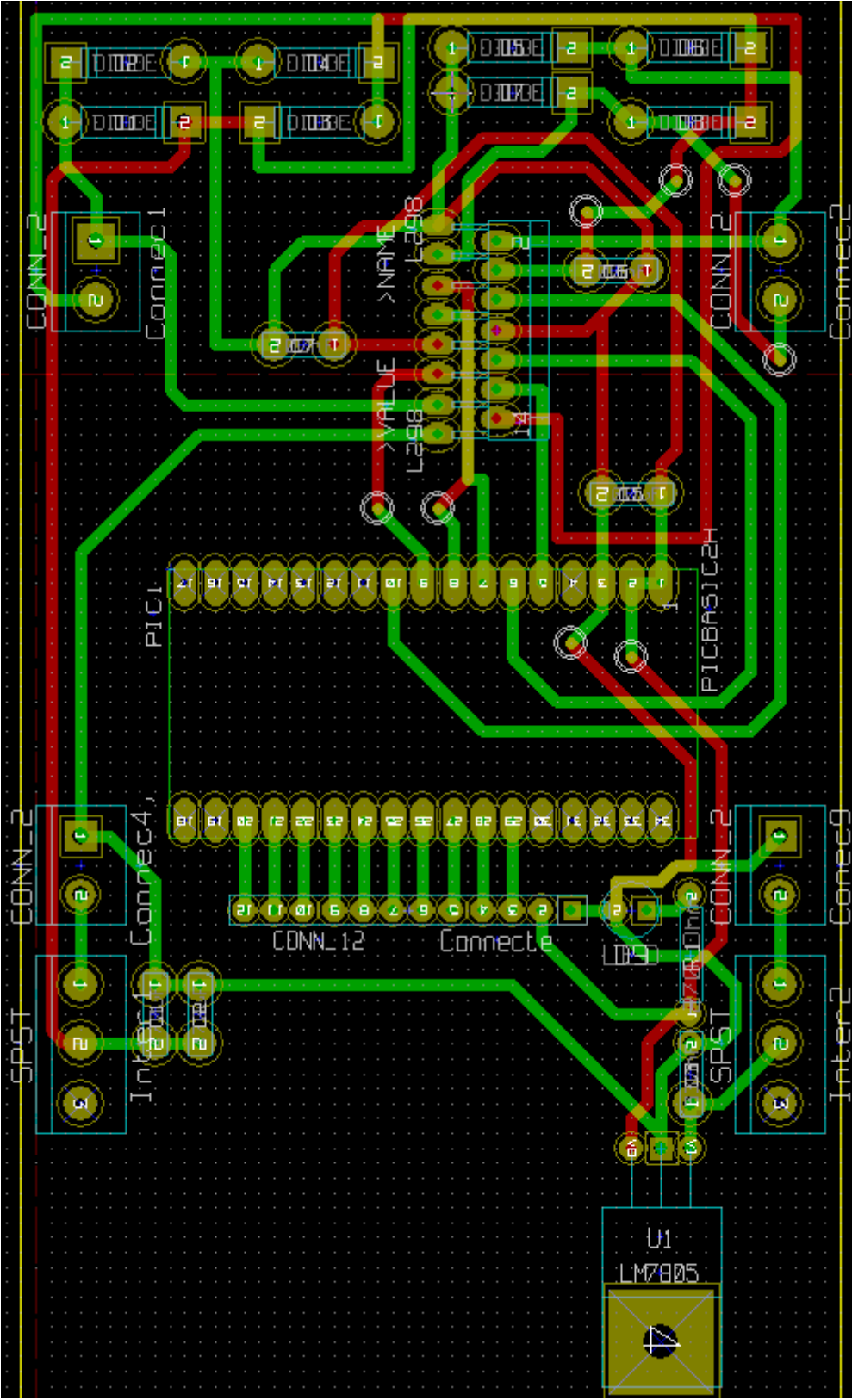
Ce projet m'a donc permis de découvrir plusieurs choses. La première est que la réalisation d'un robot n'est pas aussi impossible que je le pensais. Je ne voyais pas, avant ce projet, comment il était possible de réaliser un robot en 13 séances de 1h30, sans aucune connaissance préalables sur le sujet, et pourtant nous avons réussi. Deuxièmement, ce projet m'a appris à travailler en groupe, à bien répartir les tâches de chacun. Les anciens projets de groupe effectués précédemment lors de mon cursus à l'INSA, n'avait jamais abouti à la réalisation d'un objet matériel, seulement à des dossiers ou des programmes informatiques. Le fait de voir notre robot avancer et effectuer les tâches que nous lui demandons est une vraie satisfaction pour moi et surtout pour l'ensemble de notre groupe. Même si je suis conscient que ces capacités sont limités, car il ne peut que se déplacer, je ressors satisfait de notre travail effectué lors de ce projet de P6. Ma seule petite déception réside dans le fait que, afin de finir le robot à temps, je n'ai pas pu m'intéresser de près à toutes les parties du projet. En effet, afin d'être le plus efficace possible nous avons réparti les tâches, et je n'ai pas pu aider et m'impliquer dans la réalisation du circuit et de la carte électronique.

Pour des idées de projets similaires on peut envisager de faire des robots qui répondraient à une tâche précise plus complexe que le simple fait d'avancer ou de tourner. Ainsi dans le même domaine on pourrait concevoir des robots qui déplacent des objets ou autres fonctions simples. Un robot articulé serait envisageable.

5. BIBLIOGRAPHIE

- picbasic : <http://www.lextronic.fr/P753-module-picbasic-2s.html>
- hacheur : <http://ralphy.chez-alice.fr/Formulaire/PasaPas.htm#L298>
- télécommande : <http://arn.qui.free.fr/modules.htm>
- bloc moteur : <http://www.lextronic.fr/P1928-double-bloc-moteur-70097.html>
- roues : <http://www.lextronic.fr/P4235-paire-de-roue-diametre-32-mm.html>
- roues libre : <http://www.lextronic.fr/P4868-ball-caster-avec-bille-metal-38.html>
- châssis : <http://www.lextronic.fr/P19594-base-robotique-universel.html>

6. ANNEXES



Vu Kicad du circuit imprimé.

1	C1 -	470uF : C2
2	C2 -	470uF : C2
3	C3 -	330nF : C2
4	C5 -	100uF : C2
5	C6 -	100nF : C2
6	C7 -	100nF : C2
7	Conec9 -	CONN_2 : bornier2
8	Connec1 -	CONN_2 : bornier2
9	Connec2 -	CONN_2 : bornier2
10	Connec4, -	CONN_2 : bornier2
11	Connecte -	CONN_12 : SIL-12
12	D1 -	DIODE : D4
13	D2 -	DIODE : D4
14	D3 -	DIODE : D4
15	D4 -	DIODE : D4
16	D5 -	DIODE : D4
17	D6 -	DIODE : D4
18	D7 -	DIODE : D4
19	D8 -	DIODE : D4
20	D9 -	LED : LEDV
21	Inter1 -	SPST : bornier3
22	Inter2 -	SPST : bornier3
23	L298 -	L298 : sgs-thom-MW-15
24	PIC1 -	PICBASIC2H : comfile-PICBASIC2H
25	R1 -	270_Ohm : R4
26	U1 -	LM7805 : LM78XX

Liste des composants

