

École Nationale d'Ingénieurs de Tunis  
École Nationale Supérieure de Techniques  
Avancées



---

**Semaine Système : Rapport de Réalisation**

---

**Elaboré par :** Adnen Saadaoui  
Bilel Yezza  
Léa Musso  
Mohamed Amine Baatout  
Mohamed Chrif Lafram  
Sami Assili  
Sofiene Ouerdiane  
Soumaya Romdhani

**Encadré par :** Joseph Haggege  
Kamel Ben Saad  
Mounir Ayadi

*Année scolaire : 2013-2014*

## Remerciements

Nous tenons à remercier M. Joseph Haggege, M. Kamel Ben Saad et M. Mounir Ayadi pour leur encadrement et leurs conseils précieux tout au long de cette semaine système.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Architecture</b>	<b>6</b>
2.1	Géométrie . . . . .	6
2.2	Composants du Kit Bioloid . . . . .	8
2.2.1	Capteurs . . . . .	9
2.2.2	Actionneurs . . . . .	10
2.2.3	Alimentation . . . . .	11
<b>3</b>	<b>Management du projet</b>	<b>12</b>
3.1	Répartition des tâches . . . . .	12
3.2	Planning de la semaine . . . . .	13
3.2.1	Mercredi . . . . .	13
3.2.2	Jeudi . . . . .	13
3.2.3	Vendredi . . . . .	13
<b>4</b>	<b>Réalisation</b>	<b>14</b>
4.1	Construction . . . . .	14
4.2	Stratégie . . . . .	14
4.3	Programmation . . . . .	14
4.3.1	Test capteur de luminosité . . . . .	16
4.3.2	Test capteur de distance . . . . .	16
4.3.3	Test Moteur . . . . .	17
4.3.4	Fonction Avancer . . . . .	17
4.3.5	Fonction tourner . . . . .	18
4.3.6	Fonction fuir . . . . .	19
4.3.7	Initialisation . . . . .	20
4.3.8	Programme Principal . . . . .	21
<b>5</b>	<b>Conclusion</b>	<b>22</b>

# 1 Introduction

Au cours de cette semaine, nous étions confrontés à un défi, penser un robot sumo de la conception à la réalisation, soit un problème complexe. Les enjeux étaient divers : décrypter un cahier des charges, réussir à rassembler et appliquer dans une réalisation pratique nos connaissances, apprendre à optimiser le travail en groupe, organiser son temps de manière efficace..etc.

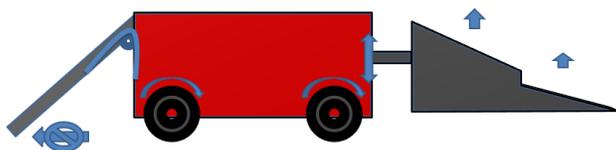
Cela nous a amené à adopter une démarche d'ingénierie système propre au métier d'ingénieur, c'est-à-dire décomposer un problème de manière méthodique en différents sous systèmes, eux même vus comme des systèmes de systèmes, pour pouvoir le résoudre en minimisant les risques et les pertes.

La semaine a été partagée en deux parties, la première pendant laquelle nous avons interprété le cahier des charges, dégagé les exigences et pensé le robot sans savoir le matériel que nous aurions à terme, et la seconde ou nous avons dut adapter nos idées à la réalité après découverte des moyens mis à notre disposition.

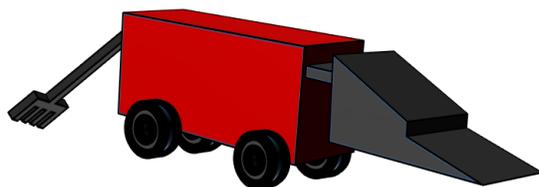
Les exigences étaient :

- Faire sortir l'adversaire.
- Respecter le règlement.
- Obéir à l'opérateur.
- Rester sur le terrain.
- Assurer l'autonomie énergétique.

Au terme de la première étape le robot que nous avons imaginé ressemblait à cela :



**Figure 1:** Schéma 2D initial



**Figure 2:** Schéma 3D final

Avec un système de défense de type « fourchette » destiné à accentuer les frottements et stopper le véhicule si celui-ci était poussé par l'adversaire, et un système d'attaque semblable à une pelle situé à l'avant, permettant de soulever le robot adverse et le déstabiliser.

Ce rapport a pour but de mettre en évidence les modifications que nous avons mises en place pour maîtriser les contraintes matérielles que l'on nous avait imposées et les difficultés que nous avons pu rencontrer.

## 2 Architecture

### 2.1 Géométrie

Au vu du matériel fournit, nous ne pouvions pas mettre en place une « pelle avant » assez massive pour déplacer le centre de gravité vers le bas, et nous n'étions pas plus en mesure de lui donner une forme de bec et d'assurer ainsi sa solidité.

Aussi avons-nous opté pour une nouvelle solution : nous avons placé les deux roues arrières sous la carcasse, alors que les deux roues avant, fixées aux moteurs étaient placées devant la carcasse et au même niveau que celle-ci, créant alors une pente entre l'arrière et l'avant.

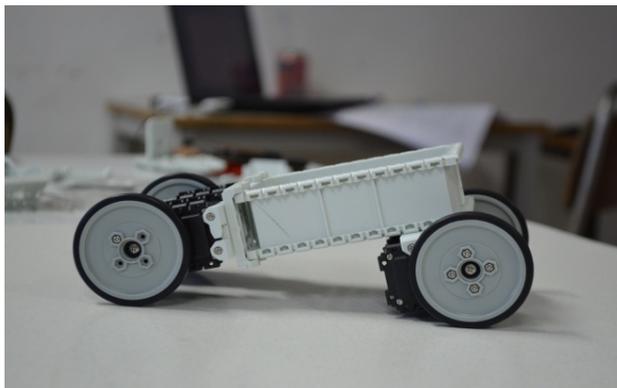


Figure 3: forme inclinée

Notre robot était donc plus stable, avec un centre de gravité bas, et plus difficile à renverser. De plus la disposition des roues avant, permettait d'empêcher un bras articulé adverse de passer sous la carcasse pour nous soulever.

Pour respecter le cahier des charges et pouvoir pousser un éventuel adversaire, nous avons ajouté à "Robo", le robot, deux bras ,dont les extrémités étaient des petites pelles, que nous avons placés à l'avant du robot.

Nous avons rajouté sur le coté avant droit du robot une sorte d'aile permettant de diminuer la surface de contact avec l'ennemi en cas d'attaque latérale. Nous l'avons placé vers l'avant, car si le robot arrive par le coté sur la partie arrière de "Robo", celui-ci ne le détectera pas et donc sera en train de tourner. Or, il ne tourne que dans un sens que la poussée sur la partie

arrière droite favorise. Une telle attaque n'aurait donc pas de conséquence sur notre robot.

Sur l'autre face nous avons rajouté une pièce en U les bras tournés vers l'extérieur. Cette pièce permet de diminuer la surface de contact avec un autre robot nous attaquant latéralement par la gauche, et donc d'en diminuer les conséquences.

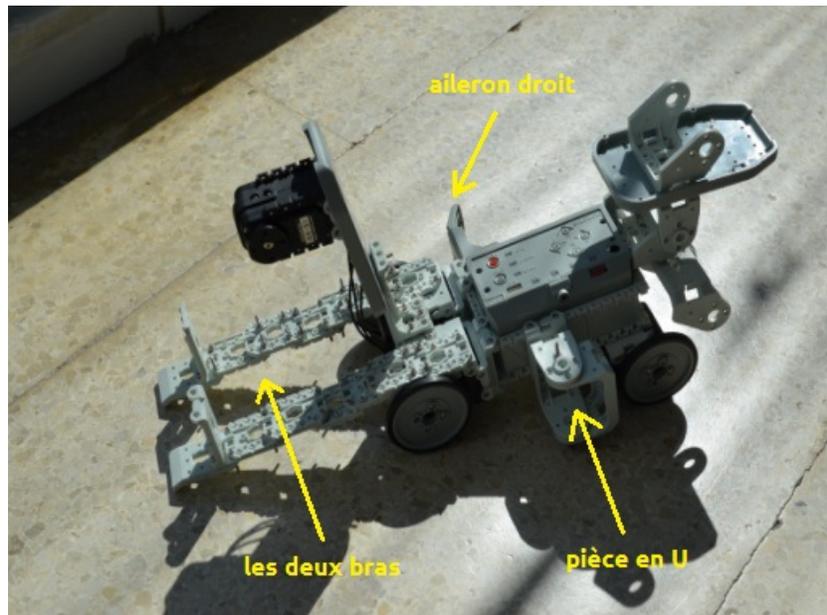


Figure 4: Forme finale

## 2.2 Composants du Kit Bioloid

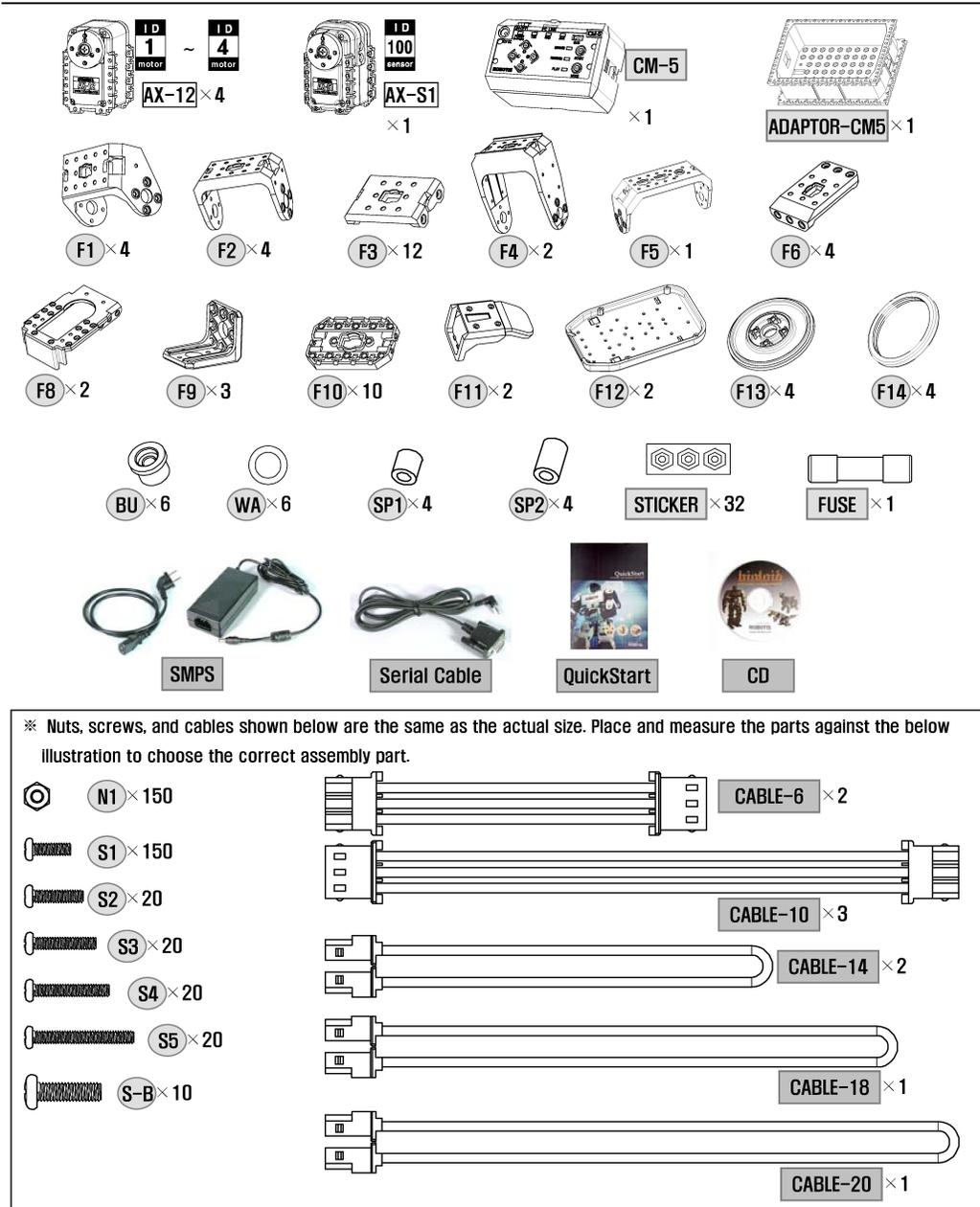


Figure 5

### 2.2.1 Capteurs

A l'origine, nous avons prévu 5 capteurs, 4 pour détecter la ligne blanche, et un pour repérer l'adversaire. Cependant dans le kit que nous avons reçu il n'y avait que trois capteurs capables de détecter dans 3 directions fixes. Il a donc fallu adapter notre stratégie en fonction de cette nouvelle contrainte.

Le capteur dont nous disposons est le AX-S1. Ce capteur est capable de détecter les distances, le contraste et la chaleur. Il a la forme suivante :



Figure 6: Capteur AX-S1

Il pourra donc à la fois détecter l'adversaire et la ligne blanche qui délimite le terrain du combat. Nous l'avons donc placé de sorte à avoir un capteur dirigé vers le bas pour détecter la ligne blanche, un vers l'avant pour repérer l'ennemi et le troisième ne sera pas utilisé puisque pointé vers le haut.

Après quelques difficultés, nous avons décidé d'utiliser les capteurs infra rouges tant pour détecter l'adversaire que la ligne blanche en modifiant la valeur de référence que l'on utilisait.

Nous nous sommes par la suite rendu compte que le capteur était capable de repérer la ligne blanche qu'à partir d'une distance minimale, aussi nous avons surélevé la position de notre capteur beaucoup plus que prévu, modifiant une fois de plus la géométrie.

### 2.2.2 Actionneurs

Pour ce qui est des moteurs, nous nous sommes confrontés au problème de n'avoir que 4 moteurs plutôt que les 5 prévus.

Nous avons préféré conserver les 4 moteurs disponibles pour les roues, car la puissance nous paraissait, si ce n'est une arme essentielle, tout du moins un énorme handicap si le robot adverse en avait plus que nous.

En effet si nous avons privilégiés la conception d'un bras articulé dans le but de mettre en place une arme performante, nous l'aurions par la même occasion rendue inefficace car, par son manque de puissance et de vitesse, notre robot de serait fait sortir avant d'avoir eu le temps de se servir de son système d'attaque.

Ainsi nous avons abandonné l'idée de munir notre bras de robot d'un moteur, nous nous contenterons de deux bras fixes. Les moteurs des roues sont des moteurs AX-12.



Figure 7: Capteur AX-12

### **2.2.3 Alimentation**

Initialement on a prévu que le robot soit énergétiquement autonome. Toutefois, lorsque nous avons reçu le Kit Bioloid, on a découvert que la batterie ne fonctionnait pas. On était donc obligé d'utiliser l'alimentation par secteur.

Par ailleurs, on a gardé la batterie dans la boîte CM-5 afin de gagner le plus de poids possible et donc d'assurer la stabilité du robot.

## **3 Management du projet**

### **3.1 Répartition des tâches**

Afin de gagner en temps et en qualité de travail, on a divisé notre groupe en deux équipes :

#### **Equipe Conception**

Sa tâche était de concevoir et d'assembler le robot. Elle est formée par :

- Adnen Saadaoui
- Léa Musso
- Soumaya Romdhani
- Sami Assili

#### **Equipe programmation**

Son rôle principal est de programmer le robot afin qu'il applique les stratégies initialement mises au point . Ces membres sont :

- Bilel Yezza
- Mohamed Amine Baatout
- Mohamed Cherif Lafram
- Sofiene Ouerdiane

## 3.2 Planning de la semaine

### 3.2.1 Mercredi

Réception du kit BIOLOID

- Equipe conception :
  - 14h->16h : prise en main du matériel.
  - 16h->18h : premier essai d'architecture.
  - 21h->minuit : nouvelle architecture et 1ers essais avec une différence de niveau entre l'arrière et l'avant du robot.
- Equipe programmation
  - 14h->18h : prise en main du logiciel, vérification de l'état du matériel.
  - 21h->minuit : test des capteurs, premières ébauches du programme.

### 3.2.2 Jeudi

- Equipe conception :
  - 14h->16h : élaboration de la nouvelle architecture.
  - 16h30->18h : fin de la nouvelle architecture et écriture du rapport.
  - le Soir : mise en place des bras + ajout de quelques modifications au niveau de la forme du robot.
- Tout le monde : 16h->16h30 : mise en place de la stratégie du combat.
- Equipe programmation
  - 14h->16h et 16h30->18h : test des capteurs, mise en place des programmes.
  - Soir : programme d'attaque, essais sur les capteurs.

### 3.2.3 Vendredi

- Equipe conception :
  - 9h->11h : finalisation du robot, rectifications.
  - 11h->12h : rectifications.
- Equipe programmation
  - 9h->12h : capteur lignes blanches.

## 4 Réalisation

### 4.1 Construction

Nous avons commencé par construire le robot de la manière la plus proche que ce que nous avions prévu. Cependant nous nous sommes confrontés à la réalité des pièces et à quelques imprévus, comme le fait d’avoir des pièces manquantes (notamment deux grandes pièces grises qui pouvaient être des pelles plus efficaces que celles que nous avons construites).

Nous avons donc essayé de nous débrouiller avec ce qu’on avait, surtout qu’on s’est aperçu de l’absence de ces pièces qu’au jour de la compétition puisqu’on n’avait pas le droit de voir les kits des groupes concurrents.

### 4.2 Stratégie

Dès que notre robot sera sur le ring, s’il détecte le robot adverse, il lui fonce dessus, sinon il tourne sur lui-même jusqu’à repérer son adversaire. A ce moment, il s’arrête pour ensuite avancer en ligne droite dans la direction où il a détecté l’autre robot et continue tant qu’il ne détecte pas de ligne blanche et que son adversaire est toujours dans son champs de vision.

### 4.3 Programmation

Après avoir découvert le kit Bioloïd de construction du robot, le groupe de programmation a commencé à lire le manuel d’utilisation du logiciel de programmation et la description des différents composants. On a aussi découvert tout ce que le robot est capable de faire.

Au départ nous voulions que Robo joue de la musique tout au long du combat mais nous nous sommes rendus compte que ce n’était pas possible simultanément avec son déplacement. C’est avec grande tristesse que nous avons choisi de privilégier le combat à la musique.

Le logiciel de programmation fourni “behavior control program” était de « très haut niveau » si on peut dire. Tout se faisait avec la souris. C’était une toute nouvelle expérience pour les membres du groupe programmation, mais elle s’est avérée pas très agréable à cause de l’interface mal organisée. La programmation classique haut niveau est mieux.

## Detection de l'adversaire

Pour la détection du robot adverse on a utilisé le module AX-S1 précisément le capteur infrarouge. Le principe du fonctionnement est d'envoyer une onde infrarouge d'une intensité donnée et recevoir l'onde réfléchi. s'il ne reçoit rien, c'est qu'il n'y a aucun signe du robot adverse et donc Robo continue à tourner.

## Detection des lignes blanches

Pour la détection de la ligne blanche on a utilisé le module AX-S1. Beaucoup pensaient que la détection de la ligne blanche se fait avec le capteur de luminosité. ça n'a rien à voir avec ça ! Il s'agit d'utiliser le capteur infrarouge de même nature que celui utilisé pour la détection des obstacles. Comme on l'a précisé dans le paragraphe précédent le principe du fonctionnement est d'envoyer une onde infrarouge d'une intensité donnée et recevoir l'onde réfléchi. cette onde ne se reflète pas de la même façon sur une surface blanche ou une surface obscure.

Mais pour percevoir cette différence il faut changer un paramètre du capteur dédié pour décider si on veut privilégier la distance ou la précision.

Evidemment pour la détection de la ligne blanche on doit privilégier la précision, ainsi une faible valeur pour ce paramètre qui peut avoir une valeur de 0 à 255. Pour notre cas on a choisit 15.

Une série de tests a permis de déterminer une distance optimale entre le capteur et le sol pour que la détection soit efficace. On observe qu'il faut un saut d'au moins 10 points pour détecter une ligne blanche étant donné la hauteur à laquelle on a positionné notre capteur. Ainsi au moment où ROBO s'installe, elle prend comme valeur de référence la valeur retournée par le capteur IR, et si on dépasse cette valeur de 30 points (par exemple du 140 pour le vert à 170 pour le blanc), alors on avance, car cela signifie que l'on a passé une ligne blanche et donc que l'on se fait sortir du terrain. Dans ce cas le robot effectue un demi tour et avance pour 2 secondes.

Finalement, notre plus importante astuce est le MAX\_TORQUE. Il s'agit du couple maximal que peut fournir le moteur. Cette variable système peut avoir une valeur entre 0 et 1023. La valeur par défaut était environ 890, donc on l'a changé à 1023. Et ce, après être sûr que cela ne peut pas endommager le robot, car le CM5 a un système de fermeture immédiate en cas de danger pour n'importe quel composant (ex : élévation de température).

### 4.3.1 Test capteur de luminosité

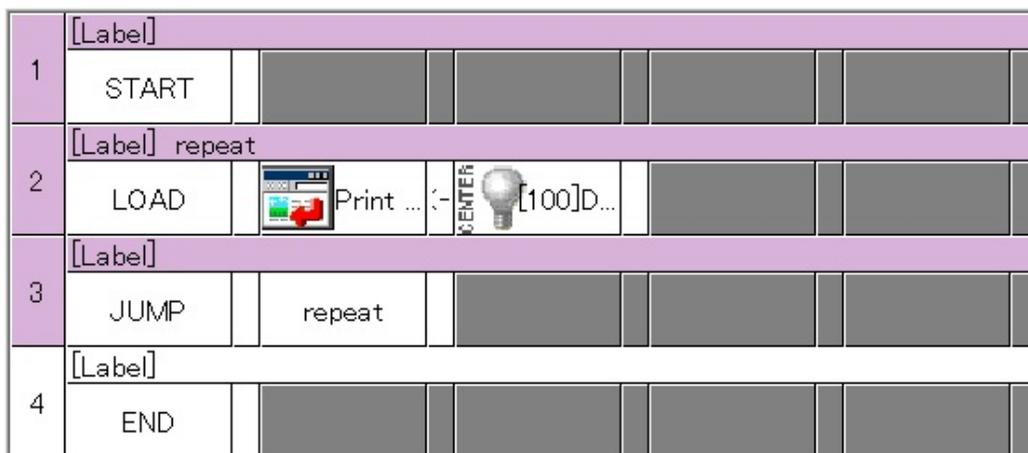


Figure 8

Ce code permet d'afficher la luminosité mesurée par le capteur à tout instant.

### 4.3.2 Test capteur de distance

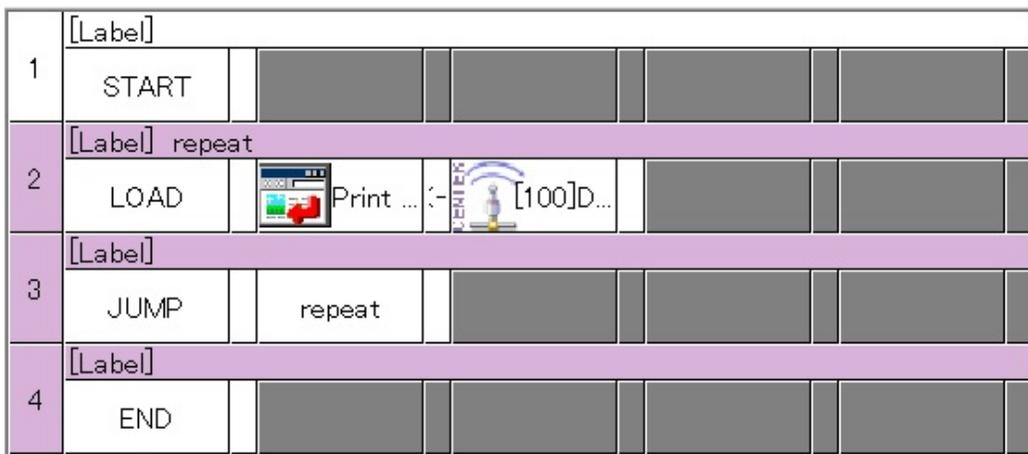


Figure 9

Ce code permet de détecter la présence d'un objet et d'afficher la distance mesurée entre l'objet et le capteur à tout instant.

### 4.3.3 Test Moteur

1	[Label]						
	START						
2	[Label]						
	LOAD	1.8	←	0			
3	[Label]						
	LOAD	 [1]Dyn...	←	1023			
4	[Label]						
	END						

Figure 10

Ce code met le moteur en rotation continue.

### 4.3.4 Fonction Avancer

[Label] avancer							
18	[Label]						
	LOAD	 [1]Dyn...	←	1023			
19	[Label]						
	LOAD	 [4]Dyn...	←	1023			
20	[Label]						
	LOAD	 [3]Dyn...	←	2047			
21	[Label]						
	LOAD	 [2]Dyn...	←	2047			
22	[Label]						
	RETURN						

Figure 11

Permet de faire avancer le robot.

### 4.3.5 Fonction tourner

	[Label] tourner						
23	LOAD		[1]Dyn...	<-	1023		
	[Label]						
24	LOAD		[4]Dyn...	<-	523		
	[Label]						
25	LOAD		[3]Dyn...	<-	1023		
	[Label]						
26	LOAD		[2]Dyn...	<-	523		
	[Label]						
27	RETURN						

Figure 12

Permet de faire tourner le robot pour capter le robot adverse.

### 4.3.6 Fonction fuir

1	[Label] fuir	[Comment]
	LOAD  Timer <- 16	
2	[Label] reculer	[Comment]
	LOAD  [1]Dyn... <- 2047	
3	[Label]	[Comment]
	LOAD  [4]Dyn... <- 2047	
4	[Label]	[Comment]
	LOAD  [3]Dyn... <- 1023	
5	[Label]	[Comment]
	LOAD  [2]Dyn... <- 1023	
6	[Label] hold3	[Comment]
	IF (  Timer > 0 ) THEN JUMP hold3	
7	[Label]	[Comment]
	RETURN	

Figure 13

Elle permet au robot de s'éloigner de la ligne blanche.

### 4.3.7 Initialisation

1	[Label] Initialisation							
	LOAD		[100]D...	<-	20			
2	[Label]							
	LOAD		TORQUE [1]Dyn...	<-	1023			
3	[Label]							
	LOAD		TORQUE [2]Dyn...	<-	1023			
4	[Label]							
	LOAD		TORQUE [3]Dyn...	<-	1023			
5	[Label]							
	LOAD		TORQUE [4]Dyn...	<-	1023			
6	[Label]							
	LOAD		TORQUE [4]Dyn...	<-	1023			
7	[Label]							
	LOAD		1:8	<-	0			
8	[Label]							
	LOAD		2:8	<-	0			
9	[Label]							
	LOAD		3:8	<-	0			
10	[Label]							
	LOAD		4:8	<-	0			

Figure 14

Permet d'initialiser le robot. La valeur de la variable MAX\_TORQUE qui représente le couple maximal du robot est changé à 1023 (valeur max).

### 4.3.8 Programme Principal

1	[Label]	START								[Comment]
2	[Label]	LOAD	 Timer	<-	24					[Comment]
3	[Label] hold	IF	(  Timer	>	0	)	THEN	JUMP	hold	[Comment]
4	[Label]	LOAD	vert	<-	 [100]D...					[Comment]
5	[Label]	LOAD	 Timer	<-	28					[Comment]
6	[Label]	CALL	avance...							[Comment]
7	[Label] hold2	IF	(  Timer	>	0	)	THEN	JUMP	hold2	[Comment]
8	[Label] repeat	COMPUTE	diff	=	 [100]D...	-	vert			[Comment]
9	[Label]	IF	( diff	>	30	)	THEN	CALL	fuir	[Comment]
10	[Label]	ELSE IF	(  [100]D...	>	10	)	THEN	CALL	avance...	[Comment]
11	[Label]	ELSE	CALL	tourne...						[Comment]
12	[Label]	JUMP	repeat							[Comment]
13	[Label]	END								[Comment]

Figure 15

## 5 Conclusion

Au terme de la semaine système, notre robot était potentiellement capable de répondre aux exigences fixées. Il peut repérer l'adversaire et la ligne blanche, et est donc capable, grâce à sa géométrie optimisée dans la limite du possible, de sortir l'adversaire et de rester sur le terrain. Il démarre bien après 3 secondes et possède deux bras articulés le rendant apte à suivre les règles fixées. Sa carte programmable et sa batterie lui permettent de se plier aux deux dernières exigences.

Les différences entre ce que nous avons prévu et ce que nous avons fait en réalité s'explique principalement par deux faits, d'une part le logiciel de programmation était beaucoup plus compliqué à prendre en main que ce que nous pensions, ce qui a entraîné un décalage entre les 2 groupes et généré du retard car le groupe de conception avait besoin d'informations sur les performances des capteurs entre autres. Et ces informations n'étaient atteignables qu'une fois le logiciel maîtrisé. D'autre part le groupe conception a beaucoup tâtonné avant de se fixer sur une architecture définitive et nous avons essayé au moins trois géométries différentes. Notre travail manquait donc d'organisation, et les deux groupes travaillaient de manière trop indépendante, en oubliant de communiquer, ce qui entraînait des quiproquos et des retards.

Cette semaine système nous a permis d'expérimenter différentes facettes du métier d'ingénieur : le travail en groupe, l'organisation du temps et du travail, la méthodologie d'analyse, les vas et vient entre expérience et théorie, la persévérance, et la remise en question...