

## **PROJET BACCALAUREAT STI 2012**

**AL@DYN**



# Sommaire :

## I – Introduction

1) Mise en situation.....	3
a) Les alarmes et les caméras.....	3
b) Le projet AI@Dyn et le robot mOway.....	3
2) Le système.....	4
a) Le besoin.....	4
b) Solution envisagée.....	4
c) Diagramme sagittal.....	5
d) Diagramme de séquence.....	6
e) Diagramme des exigences.....	7

## II – Présentation

1) Expression du besoin.....	8
a) Analyse fonctionnelle externe.....	8
a.1) Réglementation sur les dispositifs de surveillance.....	8
a.2) La couverture de surveillance.....	9
b) Analyse fonctionnelle interne.....	11
2) Présentation du robot mOway.....	13
a) Capteurs et actionneurs du mOway.....	13
b) Utilisation des capteurs et actionneurs.....	15
3) Présentation des problématiques.....	18

## III – Problématiques communes

1) Etude du servomoteur.....	19
a) Présentation et principe de fonctionnement.....	19
b) Etude du couple mécanique.....	19
c) Utilisation.....	23
2) Etude de la caméra.....	28
3) Transmission des données.....	31

## IV – Problématiques individuelles :

1) La navigation.....	34
a) Introduction.....	34
a.1) Histoire de la navigation.....	35
a.2) Comparatif des moyens de localisation.....	35
b) Solution retenue.....	39
c) Critique et modifications envisagées.....	49
2) Autres problématiques individuelles.....	49
a) Détection d'obstacles.....	49
b) Retour à la base.....	50
c) Rechargement.....	50



***mOway***

# I – Introduction / Mise en situation

## 1) Mise en situation :

### a) Les alarmes et les caméras :

Pour être alerté en cas d'intrusion dans son logement, le particulier dispose de plus en plus de systèmes d'alarme sans fil, faciles à installer et à utiliser. Ces systèmes sont dotés de capteurs, de sirène et d'un transmetteur téléphonique filaire ou GSM. En cas d'intrusion, ces systèmes déclenchent une alarme, et appellent des numéros préprogrammés.



La levée de doute consiste à vérifier le bien fondé du déclenchement d'une alarme, cela peut se faire par visionnage à distance. Il faut donc placer une ou plusieurs caméras fixes ou motorisées, à des endroits stratégiques. Le nombre de caméras est donc lié à la surface à couvrir. Pour éviter la multiplication de ces caméras, une solution consiste à rendre la caméra mobile, en la plaçant sur un robot.

### b) Le projet Al@Dyn et le robot mOway :

Le projet Al@Dyn consiste donc à associer une caméra de surveillance à une plateforme robotique, l'ensemble pouvant être piloté et consulté à distance. Ce système vient donc en complément d'un système d'alarme offrant à la personne alertée la possibilité de procéder à la levée de doute.

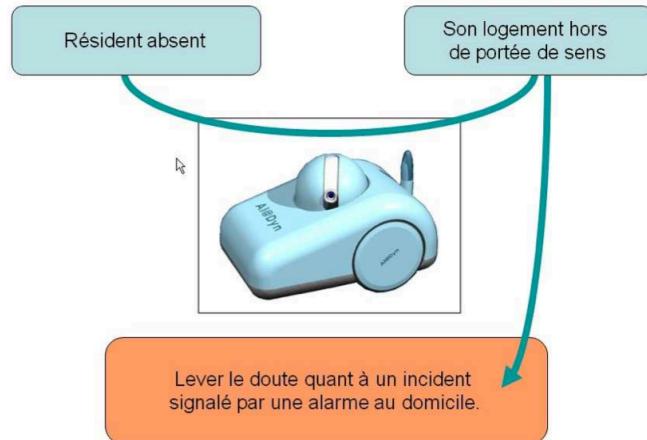




Pour valider la faisabilité d'une maquette prototype, il a été envisagé d'employer un petit robot répondant au nom de mOway développé par la Société Alecop. Il sera utilisé comme support mobile de la caméra. Ce robot est propulsé par deux moteurs à courant continu, dispose de plusieurs capteurs embarqués, d'un port de communication pouvant accueillir une caméra sans fil (Radio Fréquence), un serveur Wifi, une carte de commande Radio Fréquence ou tout autre élément utile développé sur mesure.

## 2) Le système :

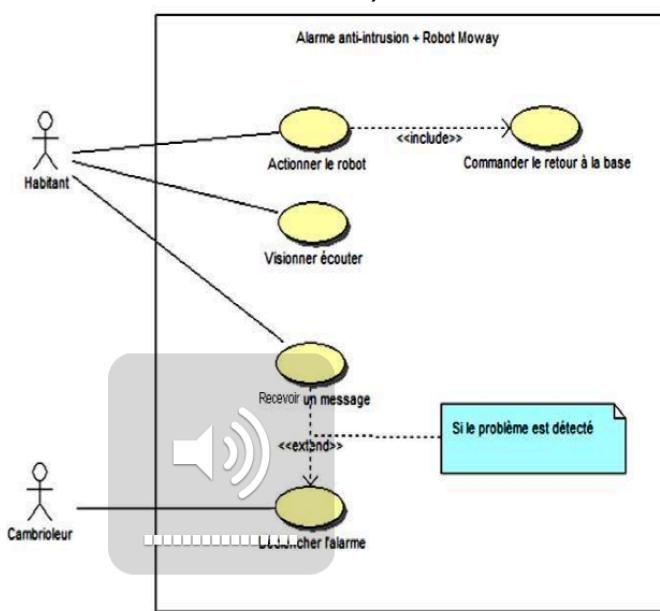
### a) Le besoin :



Ce système rend service à un résident ou habitant lorsqu'il est loin de son logement alors que son système d'alarme lui signale une intrusion.

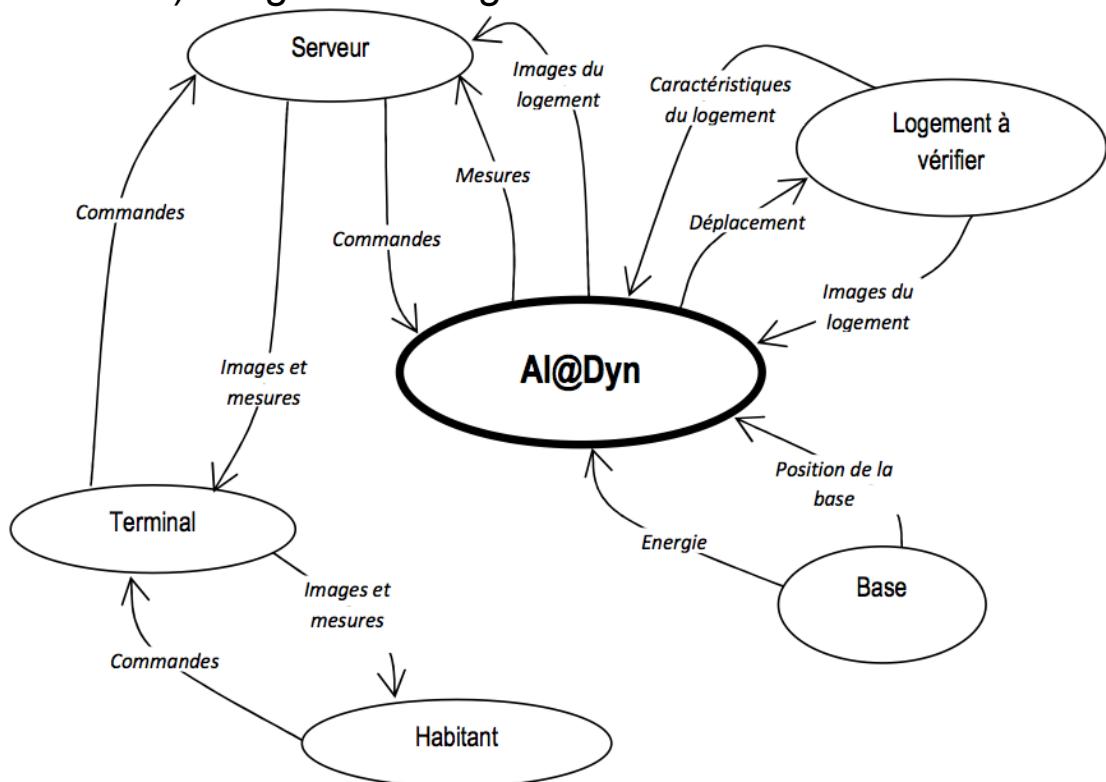
Le système agit donc sur le logement dont il transmet les images tout en s'y déplaçant. Et tout cela dans le but d'effectuer une levée de doute afin d'avoir le droit d'appeler les services de secours.

### b) Solution envisagée :



Le cambrioleur déclenche l'alarme en s'introduisant dans une espace protégé. Le résident reçoit alors un message lui signalant l'incident. Il peut donc vérifier le bien fondé de ce déclenchement en visionnant les images de la caméra de surveillance tout en actionnant son déplacement. Cette vérification pouvant être faite à distance grâce à un ordinateur ou un smartphone connecté sur internet. Il lui appartient de prévenir les forces de l'ordre pour une interpellation du malfaiteur. Lorsque la vérification est terminée, ou si la liaison est mise en défaut, AI@Dyn rentre seul à sa base, où l'attend un poste de mise en charge.

### c) Diagramme sagittal :



**AI@Dyn:** AI@Dyn est un robot équipé d'une caméra de surveillance. L'habitant peut piloter AI@Dyn à distance et visionner les images de son logement. Si l'habitant ne souhaite pas piloter le robot mais vérifier rapidement une ouverture (porte ou fenêtre) ou un objet particulier, il peut lancer une ronde. Dans ce cas AI@Dyn exécute un parcours préprogrammé, en diffusant les images. Lors de cette ronde, il fait une ou plusieurs haltes aux ouvertures ou aux objets à vérifier puis retourne à sa base. Cette dernière opération peut être automatique, si AI@Dyn est assez près de la base pour la détecter et se diriger vers elle.

**Logement:** Le logement à vérifier doit être sur un seul niveau, dans le cas contraire il faut utiliser au moins un robot par niveau. Le logement peut disposer d'une ou plusieurs pièces aux dimensions et de revêtements divers.

**Base:** La base est un système fixe remplissant plusieurs rôles : elle assure la charge des batteries d'AI@Dyn. Lorsque celui-ci n'est pas en patrouille, il est maintenu en veille avec un niveau optimal d'énergie. Elle guide, tel un phare, AI@Dyn lors de la phase d'approche. A terme, la base pourra abriter le serveur.

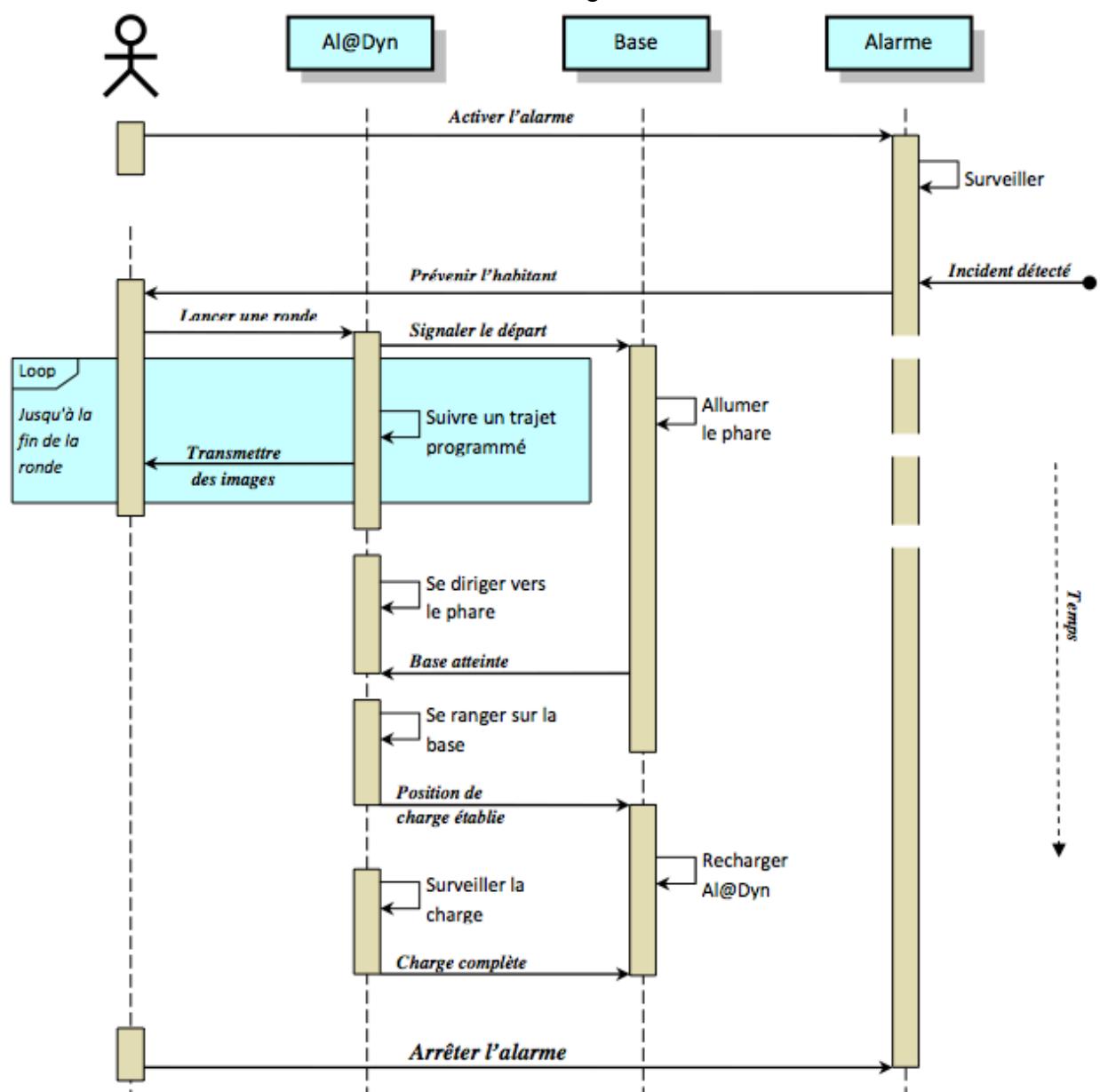
**Serveur:** Le serveur combine et retransmet les images et les caractéristiques que lui fournit AI@Dyn vers le terminal via le réseau. Il doit aussi envoyer les commandes provenant du terminal vers AI@Dyn.

**Terminal:** Le terminal est un ordinateur ou un smartphone, sur lequel l'habitant peut visionner les images et les caractéristiques du logement et du robot. Il permet également d'envoyer des ordres au robot.

**Habitant:** L'habitant est une personne physique qui a reçu un message de son système d'alarme et qui désire faire une « levée de doute » avant d'appeler les services de secours.

#### d) Diagramme de séquences :

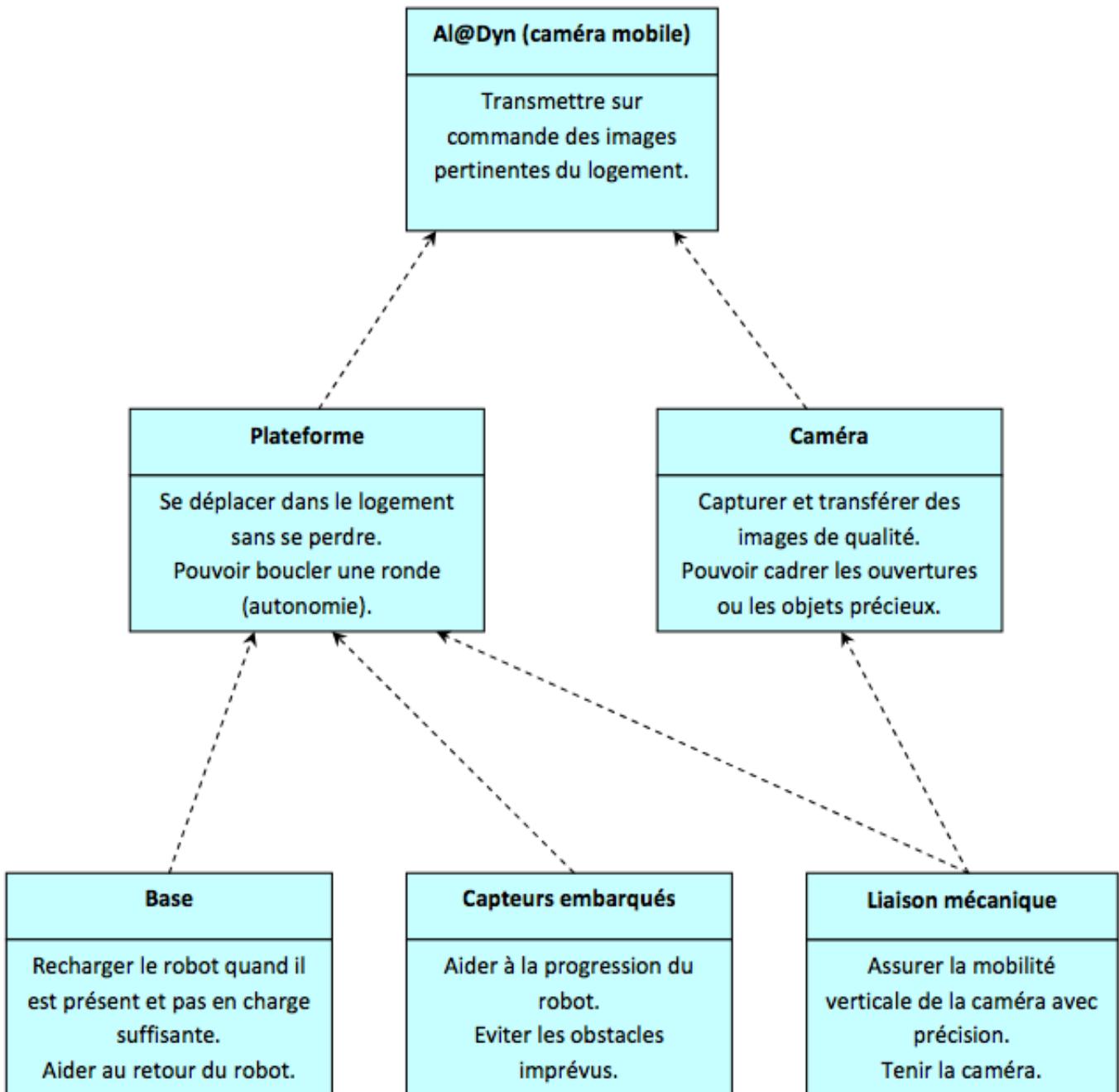
Le diagramme de séquence décrit le comportement dans le temps du système et de ses différents acteurs. Il permet alors d'identifier le flux d'informations nécessaires au fonctionnement global.



### e) Diagramme des exigences :

Le diagramme des exigences fait apparaître dans une analyse descendante (type FAST) les caractéristiques ou performances des composants liés à une donnée du cahier des charges fonctionnel.

On propose, sous une forme plus développée, ce document comme support de synthèse aux études préliminaires au projet.



## II – Présentation

### 1) Expression du besoin :

#### a) Analyse fonctionnel externe :

##### a.1) Réglementation sur les dispositifs de surveillance :

Quelques chiffres :

- Environ **320 173** cambriolages en France, entre juillet 2008 et juillet 2009.
- 80% des cambriolages ont lieu **en ville** et 13% seulement sont élucidés.
- 50% des cambriolages concernent les **résidences principales**, 6% les résidences secondaires et 44% les locaux professionnels.
- 80% des cambrioleurs empruntent **la porte**, les autres passent par le toit ou les fenêtres.
- **5 minutes** : c'est le délai moyen après lequel le monte-en-l'air abandonne son effraction.
- En général, un cambriolage ne dépasse pas une **vingtaine de minutes**.
- La sonnerie d'une alarme ferait fuir près de **95% des intrus**. Un argument majeur pour s'équiper !
- Plus étonnant que tout : **22%** des Français ayant subi un cambriolage n'ont absolument rien fait par la suite pour améliorer leur sécurité !
- **80%** des cambriolages ont lieu en plein jour
- **55%** entre 14 et 17H

Il y a donc également **20%** des cambriolages la nuit pendant le sommeil des propriétaires.

Un cambriolage dans une résidence principale coûte à ses victimes près de **6.500 euros**, se commet parfois en leur présence, la plupart du temps.

La surveillance d'un logement est un acte dissuasif dont l'objectif est la protection des biens contre le vol et la dégradation. Elle peut aussi s'étendre aux risques accidentels tels un incendie ou une inondation.

Lorsque le résident n'est pas là, l'alerte donnée aux forces d'intervention (police ou secours) doit être justifiée par une « levée de doute ».

Les conditions de cette procédure (levée de doute) sont codifiées par un texte de loi.

## Loi du 12 juillet 1983

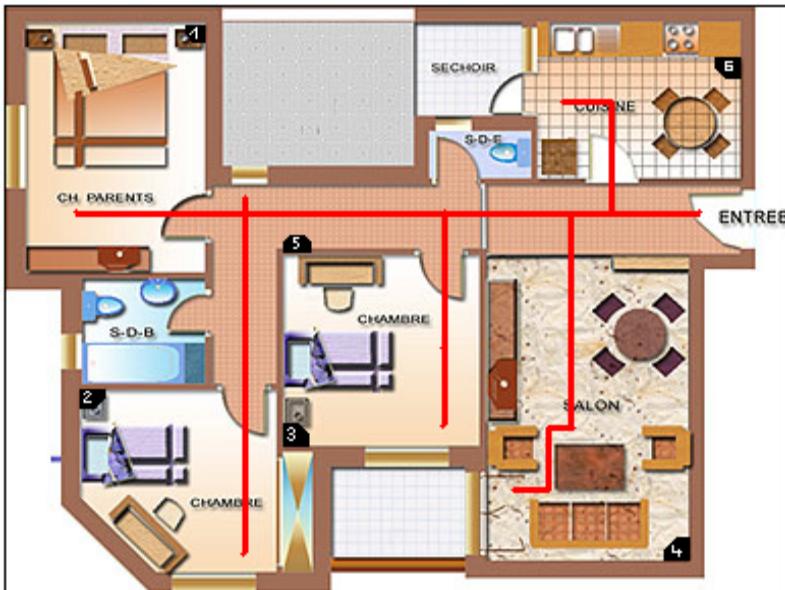
*Article 16-1 de la loi du 12 juillet 1983*

*Modifié par loi 2003-239 2003-08-18 art. 103, 105 JORF 19 mars 2003*

« Est injustifié tout appel des services de la police nationale ou de la gendarmerie nationale par les personnes physiques ou morales exerçant des activités de surveillance à distance des biens meubles ou immeubles qui entraîne l'intervention indue de ces services, faute d'avoir été précédé d'une levée de doute consistant en un ensemble de vérifications, par ces personnes physiques ou morales, de la matérialité et de la concordance des indices laissant présumer la commission d'un crime ou délit flagrant dans les locaux surveillés. »

### a.2) La couverture de surveillance :

Pour justifier l'emploi d'une caméra mobile, nous avons étudié le plan de différents appartements.

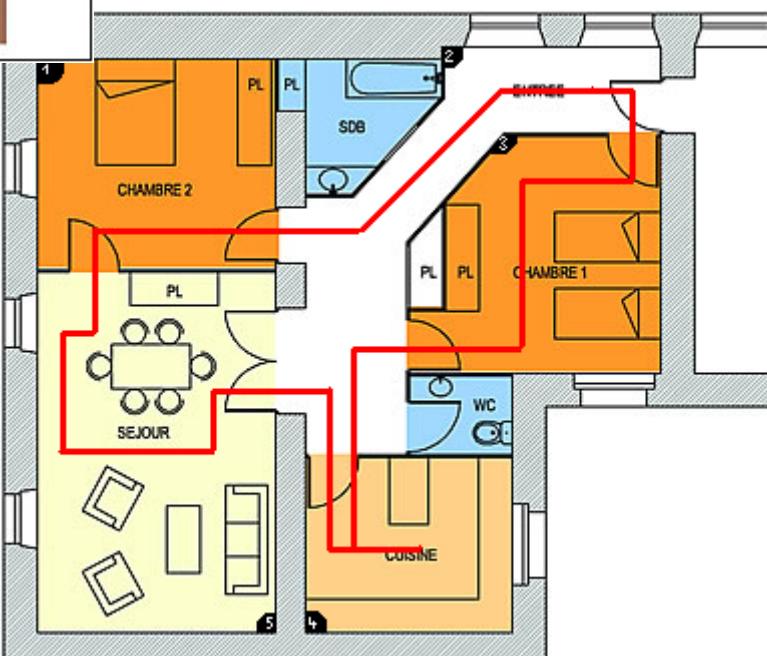


1

Caméra

Trajet d'un robot muni  
d'une caméra

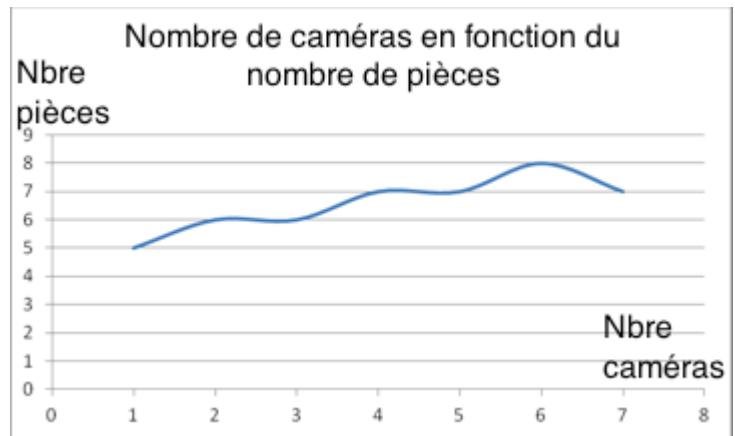
On remarque que pour un appartement de 108 m<sup>2</sup>, nous avons besoin de 6 caméras pour surveiller 5 pièces et l'entrée.



On remarque que pour celui-ci qui ne fait que 85 m<sup>2</sup>, nous avons besoin de 5 caméras pour surveiller 4 pièces et l'entrée.

Suite à l'étude de tous les appartements, nous avons pu en déduire ce tableau et cette courbe lissée.

	Nombres pièces	Caméras	Surface
N°1	10	6	99m <sup>2</sup>
N°2	8	7	105m <sup>2</sup>
N°3	9	8	113m <sup>2</sup>
N°4	8	6	93m <sup>2</sup>
N°5	6	4	104m <sup>2</sup>
N°6	6	5	201m <sup>2</sup>
N°7	6	7	129m <sup>2</sup>
N°8	8	7	108m <sup>2</sup>
N°9	6	5	85m <sup>2</sup>



Nous avons alors fais une étude sur le coût des caméras de surveillance qui nous a amené à ces 2 modèles qui nous servirons de référence.

CAMÉRA DOME INFRAROUGE COULEUR 700 TVL 20M (Référence: DOME540TVL)



PLUS D'INFORMATION

Option alimentation

Sans Alimentation  
 Avec Alimentation

EN STOCK Livré en 24h ou 72h

PRIX 106,44 € TTC

AJOUTER AU PANIER

VOIR TOUTES LES ACCESSOIRES

CAMERA MOTORISÉE INTERIEUR 540 TVL AVEC VISION DE NUIT 20 METRES (Référence: MOTO3X540)



PLUS D'INFORMATION

EN STOCK Livré en 24h ou 72h

PRIX 214,08 € TTC

AJOUTER AU PANIER

VOIR TOUTES LES ACCESSOIRES

Soyez le premier à donner votre avis sur ce produit : [Ajouter votre avis](#)

La mallette mOway contenant 2 robots mOway ainsi qu'un module de réception vidéo, 1CD, un module Wi-Fi, un émetteur radio fréquence, d'une clé USB récepteur radio fréquence et d'un module caméra.

A la vue du nombre de caméras à installer, le Moway est une solution moins onéreuse.

Le Moway est donc une **solution flexible et évolutive**, qui ne nécessite **pas d'installation**.



### b) Analyse fonctionnelle interne :

Nous avons ensuite étudié la caméra pour vérifier qu'elle réponde à nos critères pour assurer une qualité d'image suffisante pour effectuer la levée de doute.

- Pour étudier la **profondeur de champs**, nous avons posé le mOway et un objet sur le sol. Nous avons ensuite déplacé le mOway muni de sa caméra à différentes distances de l'objet.

Nous avons relevé que la netteté de l'image s'atténue à **environ 7 mètres** de l'objet pour atteindre le flou total passé 15 mètres.

- Nous avons ensuite cherché à connaître **l'angle de champs** de la caméra. Pour cela, à l'aide de notre projet Solidworks, nous avons tracé deux lignes de construction et nous avons trouvé un angle de **45°** pour obtenir une vue jusqu'à 1,50 mètres de hauteur (Valeur que nous avons choisi comme hauteur moyenne des poignées de portes et fenêtres).

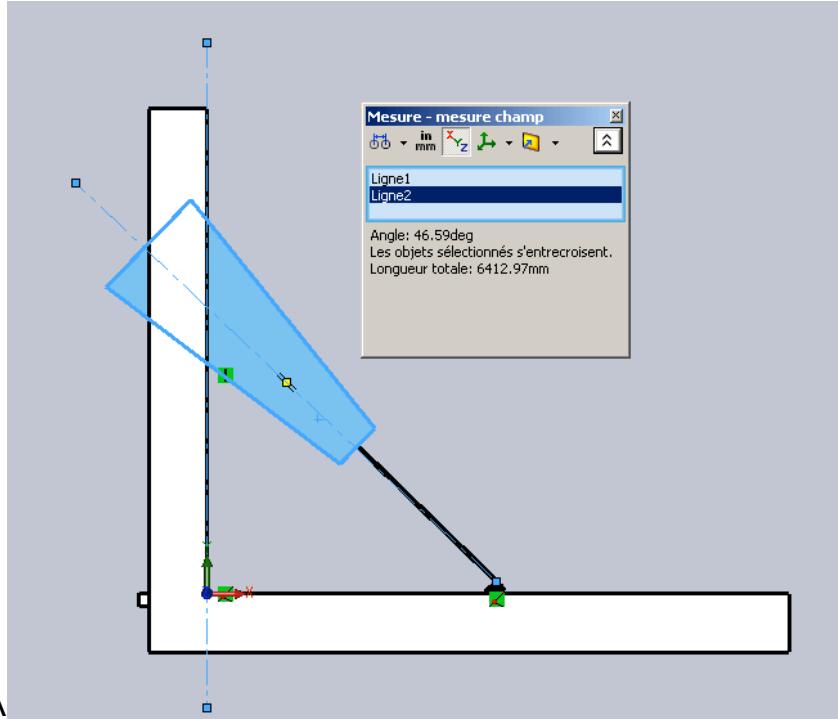
- Pour déterminer la **Résolution** (ou définition), nous avons retrouvé dans la documentation technique du capteur de la caméra, la dimension (en pixel largeur/hauteur) de l'image capturée par la caméra : **628x586**

Nous avons ensuite évalué les distances pour lesquelles les informations transmises sont satisfaisantes. Pour cela, nous avons mis en place le protocole suivant : Un mOway, un mètre et un objet quelconque.

Nous avons relevé que pour distinguer correctement une serrure forcée, il faudrait que le robot soit placé à moins de **3 mètres** de celle-ci.

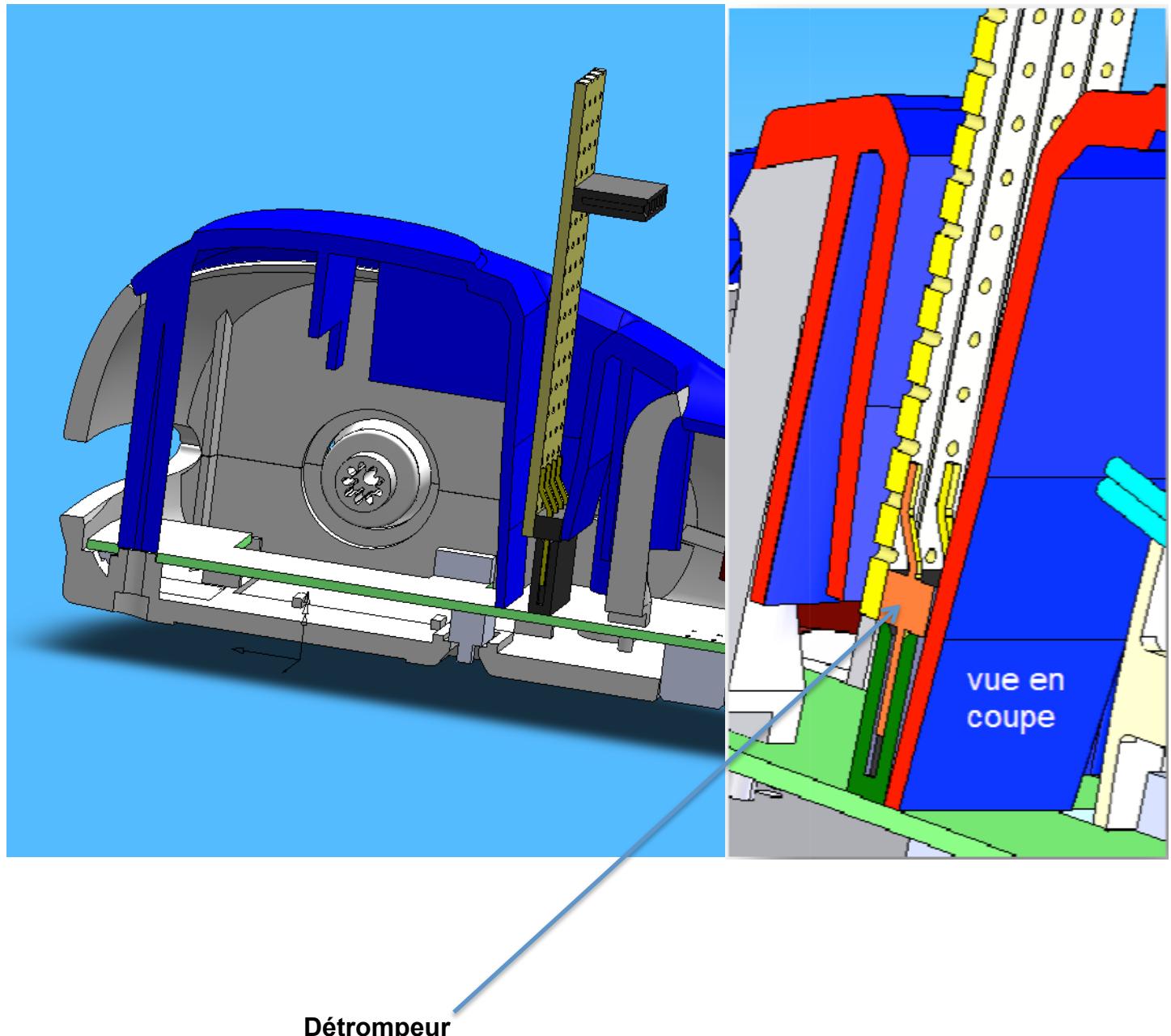
- Nous avons remarqué également que le mOway muni de sa caméra, posé au sol, ne filme qu'à la hauteur des chevilles. Il est donc primordial de pouvoir relever le champs de vision de la caméra pour pouvoir filmer correctement les serrures de portes et fenêtres.

A l'aide de Solidworks, nous avons pu relever un angle nécessaire d'environ **45°** :



Enfin, il ne serait pas convenable de brancher la caméra à l'envers car l'objectif de cette dernière ne filmerait pas ce qui se trouve devant le mOway. Cette action est bien sûr rendue impossible grâce au détrompeur situer sur le support de la caméra.

Vue en coupe



## 2) Présentation du robot mOway :

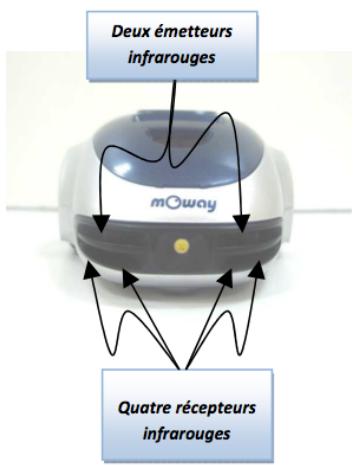
### a) Capteurs et actionneurs du mOway :

Lors du projet AI@Dyn, la plateforme robotique qui nous a été fournie est le robot mOway.



Ce robot est composé de :

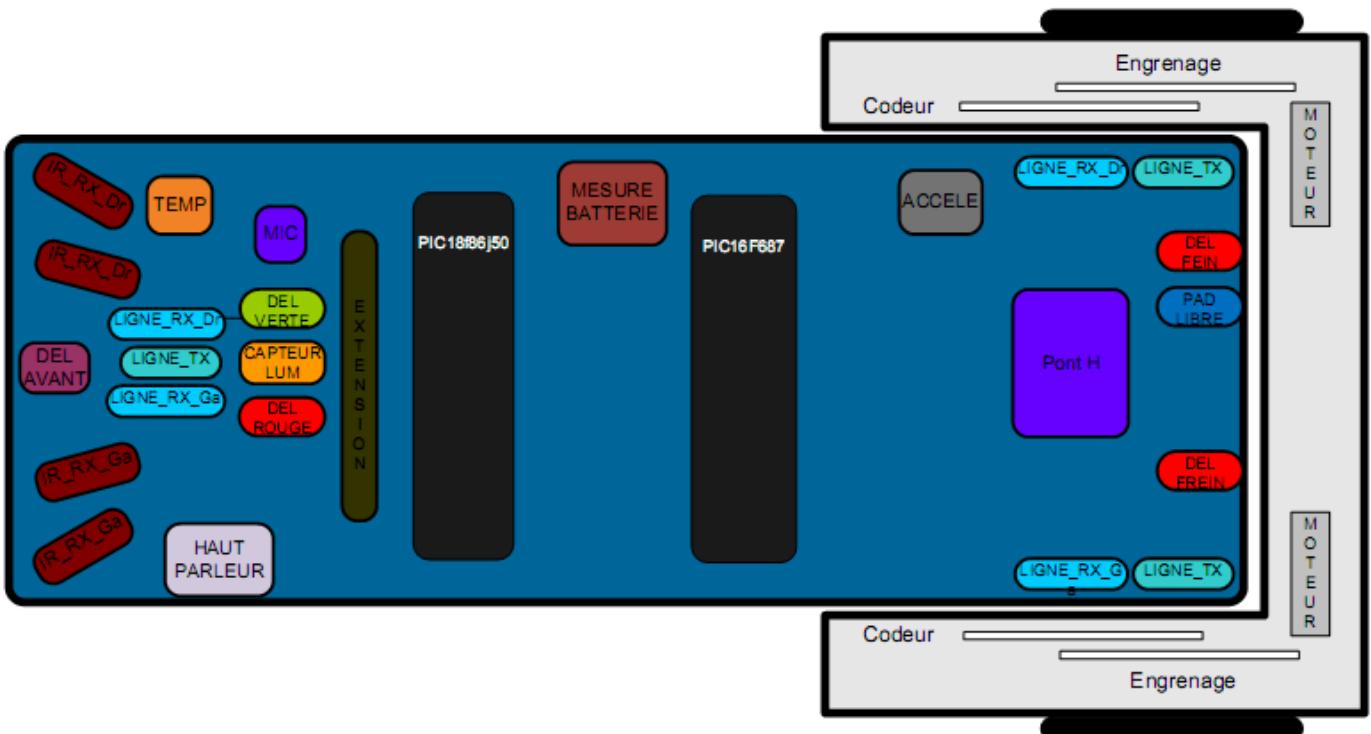
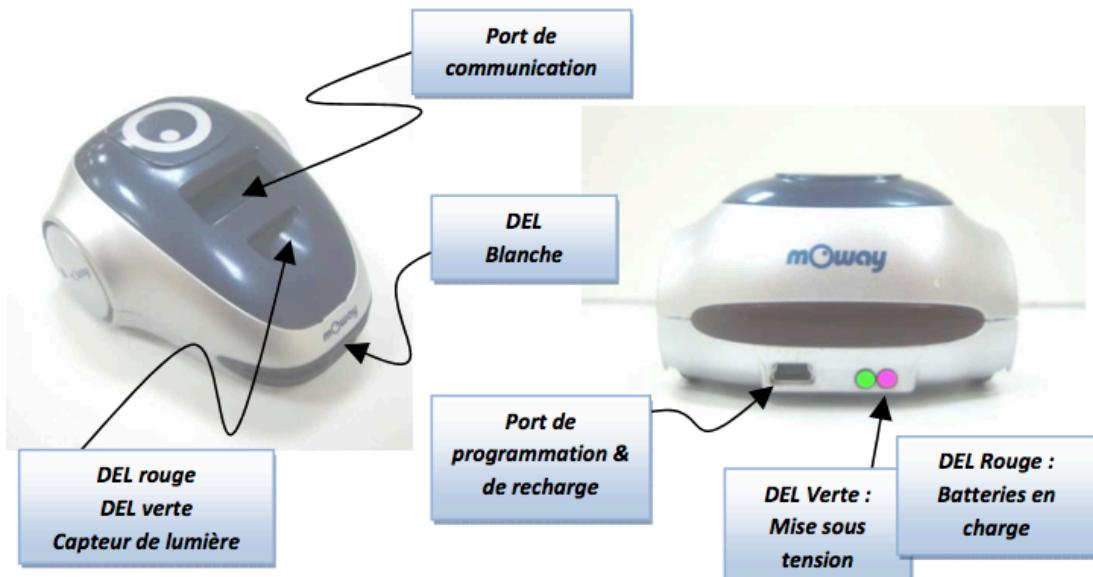
Capteurs : - 2 capteurs de lignes : Ce sont deux capteurs infrarouges situés sur la partie inférieure avant du robot. Ils utilisent la réflexion de la lumière infrarouge pour détecter le ton du sol au point où se situe le robot.



- 4 capteurs détecteurs d'obstacles : Ils utilisent également la lumière infrarouge pour détecter les objets situés à l'avant du mOway. Le capteur est composé d'une source de lumière infrarouge et de quatre récepteurs positionnés sur les deux parties extrêmes avant du robot.
- 1 capteur de luminosité : Ce capteur permet de connaître l'intensité lumineuse.
- 1 capteur de température : Ce capteur est un semi-conducteur dont la résistance diminue à mesure que la température augmente.
- 1 microphone : Il permet au robot de détecter des sons compris entre 100 Hz et 20 kHz.
- 1 accéléromètre : Il permet de savoir si le robot est en position correcte ou s'il s'est retourné ou couché sur un côté. Il permet également de savoir si le robot est entré en collision ou s'il est tombé.
- 1 capteur de niveau de charge de la batterie : La lecture de ce capteur permet de connaître le niveau de charge de la batterie restant.

Actionneurs : - 2 moteurs pour faire avancer et tourner le robot.

- 4 types de DEL pour éclairer ou donner des indications sur la charge et autre).
- 1 haut parleur pour émettre des signaux sonores.



Les éléments qui peuvent nous être utiles au projet Al@Dyn sont :

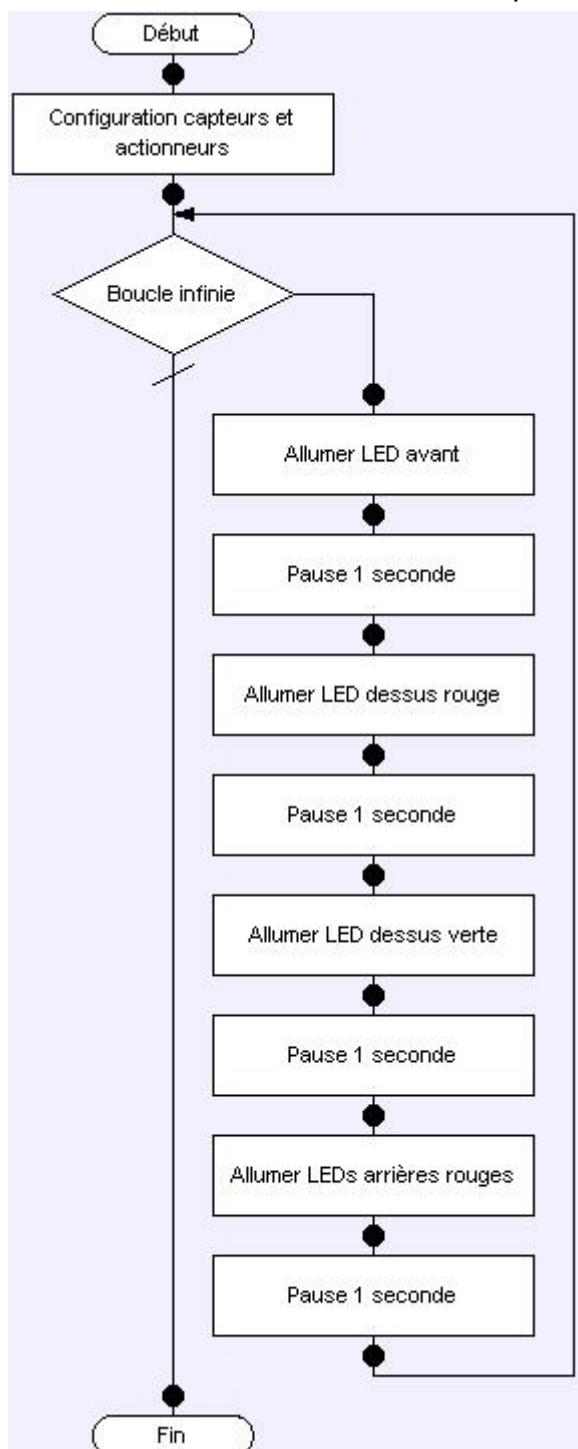
- Les capteurs de lignes (pour détecter la présence d'un sol ou non) ;
- Les capteurs de détection d'obstacles (pour détecter des obstacles sur notre parcours) ;
- Le capteur de luminosité (pour activer des DEL s'il fait trop sombre) ;
- L'accéléromètre (pour détecter une position correcte et stable du robot) ;
- Le capteur de niveau de charge de la batterie (pour permettre un retour à la base de chargement en cas de batterie faible) ;
- Les deux moteurs (pour permettre au robot de se déplacer) ;
- Les DEL (pour s'éclairer et détecter des objets).

### b) Utilisation des capteurs et actionneurs :

Pour étudier le fonctionnement du robot ainsi que de ses capteurs et actionneurs, nous avons réalisé plusieurs programmes en langage C que nous avons chargés sur le robot.

En voici deux présentés sous la forme d'algorigrammes et en langage C :

- 1) Allumer successivement les DEL du mOway avec un intervalle de temps de 1 seconde entre chaque DEL puis les éteindre toutes et répéter constamment cette opération.



Algorigramme du premier programme

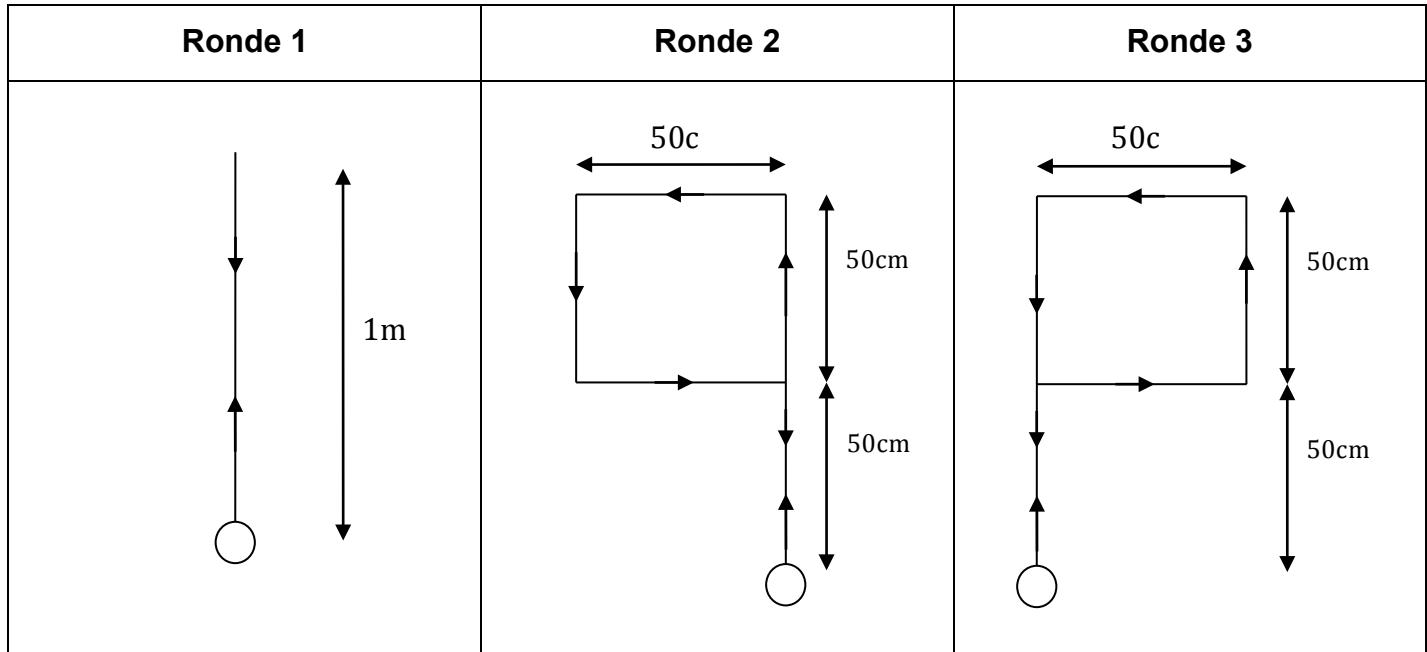
Premier programme en langage C :

```

void main()
{
    //*****SENSOR CONFIGURATION*****
    SEN_CONFIG();
    //*****ENGINE CONFIGURATION*****
    MOT_CONFIG();

    while(1)          // Sert à exécuter le programme en boucle
    {
        LED_FRONT_ON(); // Allumer la LED avant
        PAUSE_SECONDE(1); // Pause d'une seconde
        LED_TOP_RED_ON(); // Allumer la LED du dessus rouge
        PAUSE_SECONDE(1); // Pause d'une seconde
        LED_TOP_GREEN_ON(); // Allumer la LED du dessus verte
        PAUSE_SECONDE(1); // Pause d'une seconde
        LED_BRAKE_ON(); // Allumer les LED arrières rouges
        PAUSE_SECONDE(1); // Pause d'une seconde
        LED_FRONT_OFF(); // Eteindre la LED avant
        LED_TOP_RED_OFF(); // Eteindre la LED du dessus rouge
        LED_TOP_GREEN_OFF(); // Eteindre la LED du dessus verte
        LED_BRAKE_OFF(); // Eteindre les LED arrières rouges
        PAUSE_SECONDE(1); // Pause d'une seconde
    }
}
  
```

2) Le robot doit faire successivement les trois rondes ci-dessous. A la mise sous tension et entre chaque ronde, il doit faire une pause de 3s et attendre une consigne sonore (ex : clappement des mains). Les lignes droites doivent être parcourues sans à-coups.



```

void main()
{
    //*****SENSOR CONFIGURATION*****
    SEN_CONFIG();
    //*****ENGINE CONFIGURATION*****
    MOT_CONFIG();
    PAUSE_SECONDE(3);

    if (SEN_MIC_DIG())
        MOT_LONGUE_DISTANCE(100, FWD, DISTANCE, 1000);
        MOT_ROT(100, FWD, CENTER, RIGHT, ANGLE, 100);
        MOT_LONGUE_DISTANCE(100, FWD, DISTANCE, 1000);
    }
    else MOT_STOP();

    PAUSE_SECONDE(3);

    if (SEN_MIC_DIG());
        MOT_LONGUE_DISTANCE(100, FWD, DISTANCE, 1000);
        MOT_ROT(100, FWD, CENTER, LEFT, ANGLE, 50);
        MOT_LONGUE_DISTANCE(100, FWD, DISTANCE, 500);
        MOT_ROT(100, FWD, CENTER, LEFT, ANGLE, 50);
        MOT_LONGUE_DISTANCE(100, FWD, DISTANCE, 500);
        MOT_ROT(100, FWD, CENTER, LEFT, ANGLE, 50);
        MOT_LONGUE_DISTANCE(100, FWD, DISTANCE, 500);
        MOT_ROT(100, FWD, CENTER, RIGHT, ANGLE, 50);
    }
    else MOT_STOP();

    PAUSE_SECONDE(3);

    if (SEN_MIC_DIG());
        MOT_LONGUE_DISTANCE(100, FWD, DISTANCE, 1000);
        MOT_ROT(100, FWD, CENTER, RIGHT, ANGLE, 50);
        MOT_LONGUE_DISTANCE(100, FWD, DISTANCE, 500);
        MOT_ROT(100, FWD, CENTER, RIGHT, ANGLE, 50);
        MOT_LONGUE_DISTANCE(100, FWD, DISTANCE, 500);
        MOT_ROT(100, FWD, CENTER, LEFT, ANGLE, 50);
    }
    else MOT_STOP();
}

// Pause de 3 secondes pour poser le Moway au sol avant le début du programme

// Début du programme SI le micro détecte un son
// Faire avancer le Moway à vitesse maximum, tout droit, en avant, sur 1 mètre
// Faire tourner le robot à 180° sur la droite à vitesse maximum
// Faire avancer le Moway à vitesse maximum, tout droit, en avant, sur 1 mètre

// Autrement, si le micro ne détecte pas de son, le Moway ne fait rien

// Pause de 3 secondes

// Début du programme suivant SI le micro détecte un son
// Faire avancer le Moway à vitesse maximum, tout droit, en avant, sur 1 mètre
// Faire tourner le robot à 90° sur la gauche à vitesse maximum
// Faire avancer le Moway à vitesse maximum, tout droit, en avant, sur 50 cm
// Faire tourner le robot à 90° sur la gauche à vitesse maximum
// Faire avancer le Moway à vitesse maximum, tout droit, en avant, sur 50 cm
// Faire tourner le robot à 90° sur la gauche à vitesse maximum
// Faire avancer le Moway à vitesse maximum, tout droit, en avant, sur 50 cm
// Faire tourner le robot à 90° sur la droite à vitesse maximum

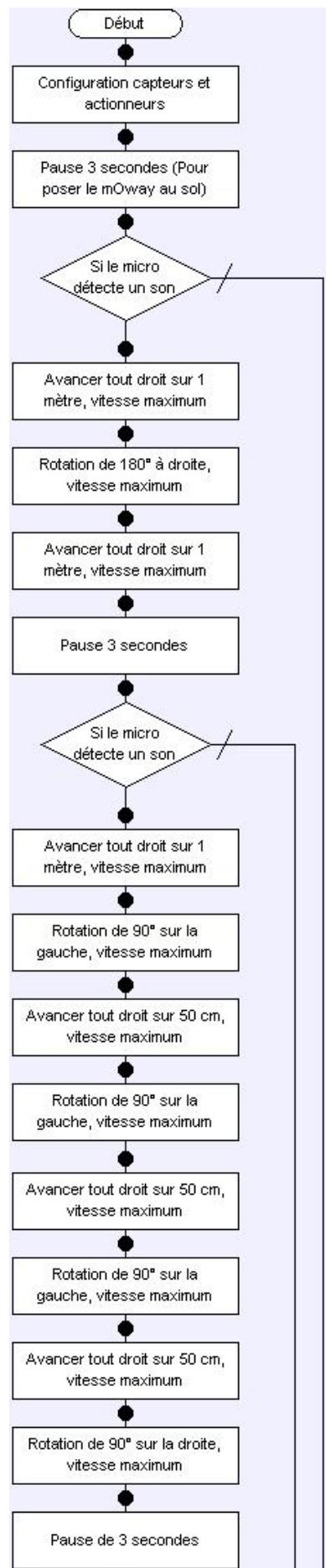
// Autrement, si le micro ne détecte pas de son, le Moway ne fait rien

// Pause de 3 secondes

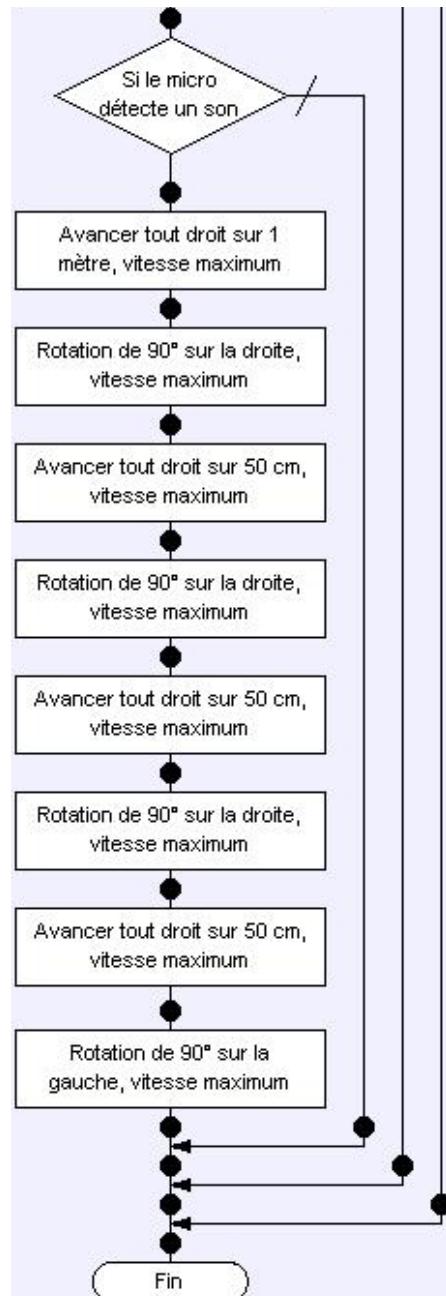
// Début du dernier programme SI le micro détecte un son
// Faire avancer le Moway à vitesse maximum, tout droit, en avant, sur 1 mètre
// Faire tourner le robot à 90° sur la droite à vitesse maximum
// Faire avancer le Moway à vitesse maximum, tout droit, en avant, sur 50 cm
// Faire tourner le robot à 90° sur la droite à vitesse maximum
// Faire avancer le Moway à vitesse maximum, tout droit, en avant, sur 50 cm
// Faire tourner le robot à 90° sur la droite à vitesse maximum
// Faire avancer le Moway à vitesse maximum, tout droit, en avant, sur 50 cm
// Faire tourner le robot à 90° sur la gauche à vitesse maximum

// Autrement, si le micro ne détecte pas de son, le Moway ne fait rien

```



### Algorigramme du second programme :



## - Présentation des problématiques :

Sur commande de l'utilisateur, le robot AI@Dyn doit :

- ✓ **se déplacer** et **transmettre** les images du logement à vérifier
- ✓ **localiser** et **se déplacer** en direction de la base afin de **récupérer** et **stocker** de l'énergie nécessaire à son fonctionnement.

Il doit aussi **mesurer** et **transmettre** le niveau de sa batterie ainsi que les mesures effectuées sur son environnement comme la luminance, la luminosité et la présence d'obstacle ou d'escalier.

Dans le cadre de l'épreuve de STI, notre projet se concentre sur la ronde d'AI@Dyn à travers un logement d'habitation.

Seules la caméra et la plateforme robotique sont fournies.

Nous devons mettre en œuvre l'association de ces deux éléments, leur connexion à un réseau de communication, compléter le dispositif afin de lui conférer toute l'autonomie nécessaire à l'exécution d'une ronde, depuis le départ piloté à distance, jusqu'à son retour automatique vers la base et sa mise en charge.

Problématique commune : « Caméra » : Comment capturer des images pertinentes et de qualités ?

Pour répondre à cette problématique, nous devons mettre en œuvre la plateforme robotique (mOway), fabriquer le module de commande du servomoteur assurant l'élévation du champ de la caméra et configurer le dispositif pour la transmission des images jusqu'au terminal, afin que l'habitant puisse procéder à la levée de doute.

Problématiques individuelles :

« Navigation » : Comment garder le cap lors d'une ronde ?

Concevoir, monter, câbler et mettre en œuvre un dispositif assurant le suivi d'un parcours le plus précisément possible.

« Obstacle » : Comment détecter un obstacle et en rendre compte ?

Concevoir, monter, câbler et mettre en œuvre un capteur d'obstacles et envisager une procédure de contournement.

« Base » : Comment se diriger et se placer sur la base ?

Concevoir, monter, câbler, configurer et mettre en œuvre un phare sur la base.

« Recharge » : Comment recharger le mOway sans contact galvanique ?

Concevoir, monter, câbler et mettre en œuvre un système de recharge par induction.

### III – Problématiques communes

#### 1) Etude du servomoteur :

Pour répondre au problème du niveau de champs de vision de la caméra, nous avons du mettre en place un servomoteur capable de faire pivoter la caméra selon un axe afin d'élever son champs de vision.

##### a) Présentation et principe de fonctionnement :

Pour notre projet, il nous a été fourni un servomoteur Blue Bird BMS-303.

Pour le faire fonctionner, ce dernier est alimenté grâce à la nappe du servomoteur qui est composée de trois fils.

On a deux fils pour alimenter le moteur entre 5V (Fil rouge) et la masse (Fil marron), ainsi qu'un fil pour commander la position de l'axe de sortie (Fil orange).

Pour faire bouger l'axe de sortie, il faut lui envoyer un signal sur le fil de commande.

Ce signal est en créneau et c'est la largeur d'impulsion qui détermine l'angle de rotation de l'axe de sortie du servomoteur.



Le servomoteur se compose d'un système réducteur de vitesse (avec des engrenages), d'un moteur à courant continu, d'un potentiomètre et d'un circuit électronique. L'axe du potentiomètre est solidaire de l'axe de sortie.

Donc, la résistance aux bornes du potentiomètre varie en fonction de la position de l'axe : cela donne une information de position au circuit électronique. Le circuit électronique cherche en permanence un équilibre entre le signal de commande et l'information donnée par le potentiomètre. Lorsque le circuit électronique a trouvé cet équilibre, il commande l'arrêt du moteur à courant continu.

L'asservissement de position correspond au fait que lorsque l'on envoie une impulsion avec le monostable, celui-ci permet une rotation de l'axe suivant un certain angle.

##### b) Etude du couple mécanique :

Tout d'abord, nous avons relevé le courant consommé par le servomoteur.

Lorsque l'on place un ampèremètre en série, entre l'alimentation et le servomoteur, on relève un courant  $I=6,85\text{ mA}$ , absorbé par celui-ci.

Rappel :

Le couple désigne l'effort en rotation applique a un axe. Il s'exprime en Newton-mètre (Nm).

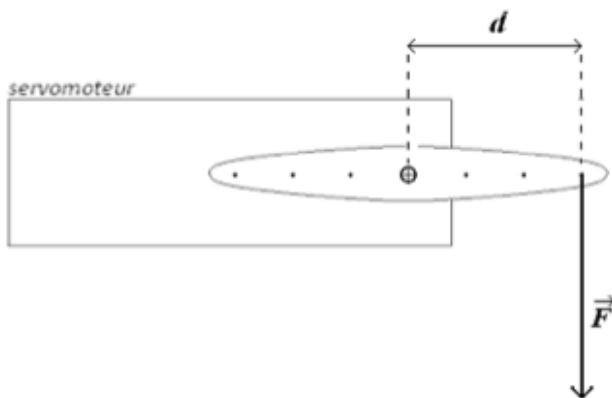
On le calcule à l'aide de la relation suivante :

$$C = F * d$$

C : couple (Nm)

F : force (N)

d : longueur du bras de levier (m)



Le servomoteur déplace la caméra de masse m, la force F correspond également au poids de cette caméra. On peut donc la calculer avec la formule :  $F = C/d$  (qu'on déduit de la formule ci-dessus.)

Le couple C est donc maximum lorsque que la distance d est maximale.

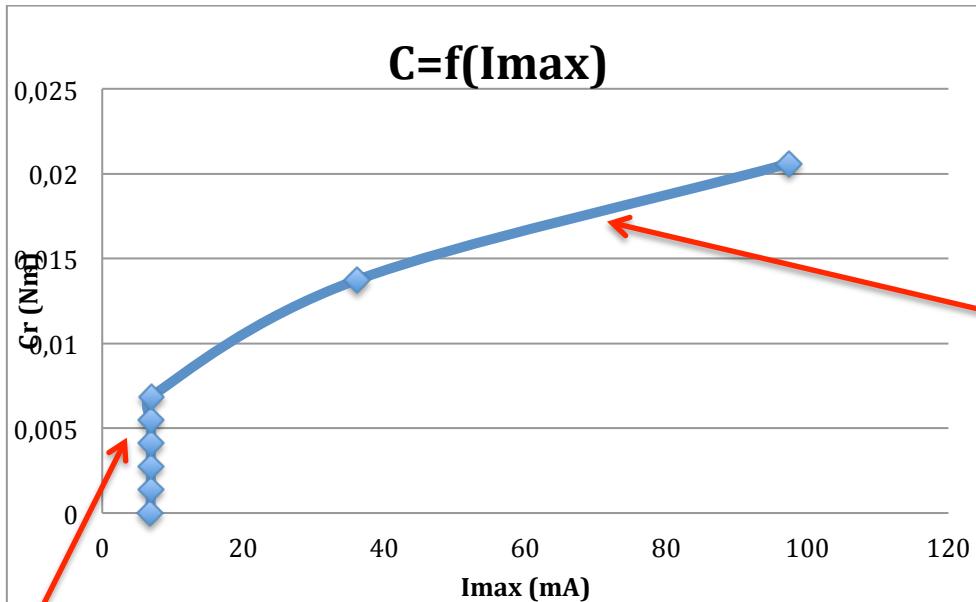
Après mesure, la distance d maximale séparant le trou le plus éloigné de l'axe de rotation est de 1,4 cm.

Nous testons alors le servomoteur avec plusieurs poids de différentes masses.

Suite à nos relevés expérimentaux, nous obtenons le tableau suivant :

Masse m (kg)	0 (Servomoteur à vide)	0,01	0,02	0,03	0,04	0,05	0,1	0,15
Poids P=m.g (N)	0	0,0981	0,1962	0,2943	0,3924	0,4905	0,981	1,4715
I <sub>max</sub> (mA)	6,85	6,89	6,9	6,91	6,94	6,95	36,09	97,42
U (V)	5	5	5	5	5	5	5	5
Couple résistant Cr=P*d (Nm)	0	0,00137	0,00274	0,00412	0,00549	0,00687	0,01373	0,0206
Puissance absorbée Pa (mW)	34,25	34,45	34,5	34,55	34,7	34,75	180,45	487,1

On en déduit cette courbe du couple résistant en fonction de l'intensité  $I_{max}$  :



Partie 1 : Le couple résistant est égale au couple utile. On obtient donc une droite verticale à l'axe des ordonnées.

On peut en déduire une relation liant le couple utile  $C_u$  développé par le servomoteur à l'intensité  $I$  du courant qui traverse le moteur :

$$C_u = a \cdot I + b \quad (\text{avec } a = 0.15 \text{ et } b = 0.0055)$$

Plus le couple résistant augmente plus la puissance absorbée par le servomoteur augmente à son tour. Idéalement, il faut que la caméra soit positionnée au plus près de l'axe de rotation afin que la puissance absorbée par le servomoteur soit minimale.

La caméra ayant une masse inférieure à 150g, ce servomoteur répond à notre besoin.

Nous étudions ensuite la résistance du mOway hors tension :

On mesure les distances séparant les 3 trous du palonnier de l'axe de rotation du palonnier :  
 -d1=0,8cm  
 -d2=1,1cm  
 -d3=1,4cm

La masse maximale que le palonnier peut porter tout en restant horizontal est 80g.

Calcul du couple résistant maximal :

$$\begin{aligned} C_{Rmax1} &= 1.275 \cdot 0.8 \cdot 10^{-3} = 1.02 \cdot 10^{-3} \\ C_{Rmax2} &= 0.981 \cdot 1.1 \cdot 10^{-3} = 1.07 \cdot 10^{-3} \\ C_{Rmax3} &= 0.784 \cdot 1.4 \cdot 10^{-3} = 1.098 \cdot 10^{-3} \end{aligned}$$

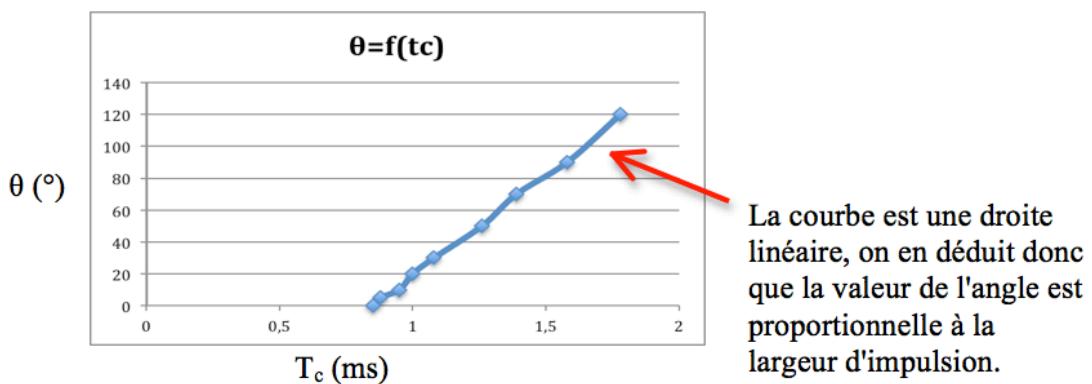
Partie 2 :  
 Le couple résistant est supérieur au couple utile. On obtient donc une droite linéaire.  
 Le servomoteur ne peut plus retenir sa charge, il y a donc une rotation du palonnier.

Avec différents poids nous testons la résistance du servomoteur et obtenons ce tableau :

$m$ (g)	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180
$d$ (cm)																	X	X	X
$d_1=0,8$ cm																			
$d_2=1,1$ cm											X	X	X	X	X	X	X	X	X
$d_3=1,4$ cm									X	X	X	X	X	X	X	X	X	X	X

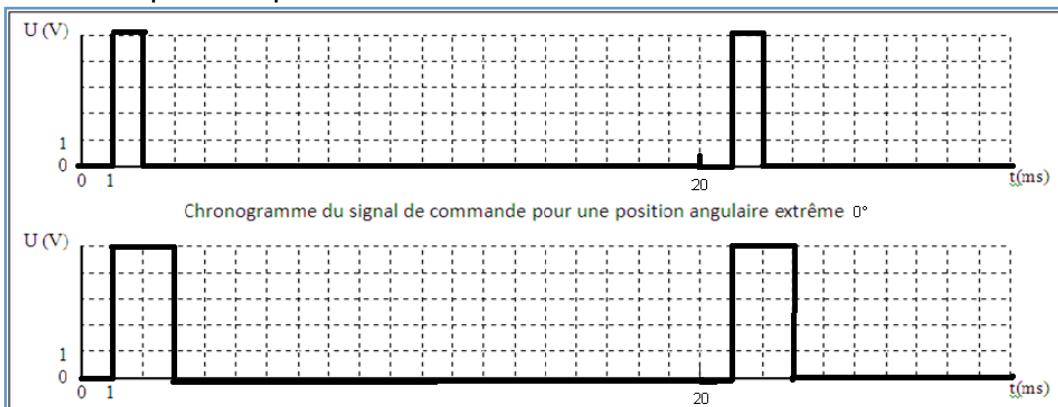
La position du palonnier est définie par la largeur d'impulsion du signal de commande. La largeur d'impulsion minimum est  $T_{cmin}=100\mu s$ .

Suite à différents relevés, nous obtenons cette courbe :



L'intérêt d'une commande linéaire est que l'on peut déterminer un angle précis selon la durée d'impulsion.

On peut donc donner par exemple les chronogrammes du signal de commande pour les positions extrêmes :



La valeur maximale de l'angle balayé par le servomoteur est de  $120^\circ$ . Cet angle permettra donc à la caméra choisie d'assurer une bonne inspection de la pièce.

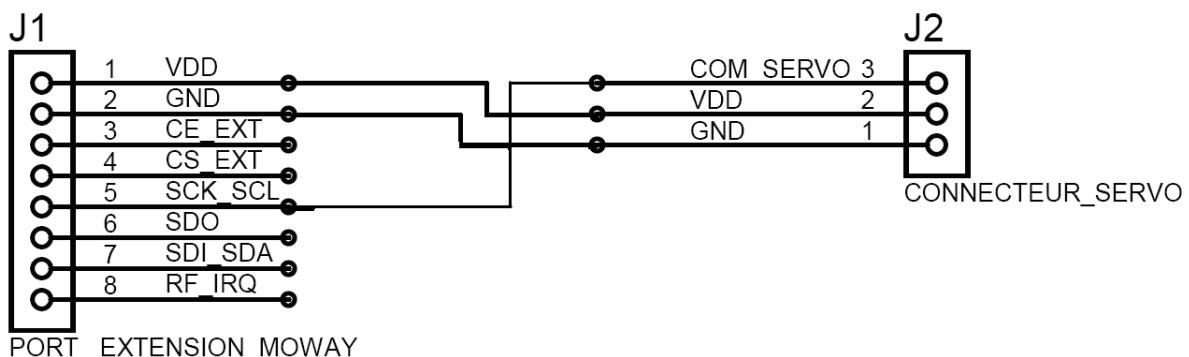
Ce servomoteur répond donc à nos besoins si la caméra est positionnée sur l'axe de rotation. Le signal de commande nous permet d'avoir un angle prédéfini, permettant de garder un angle constant.

Cependant, nous serons limité à un angle de  $120^\circ$ , au delà, le servomoteur ne fonctionnera plus correctement.

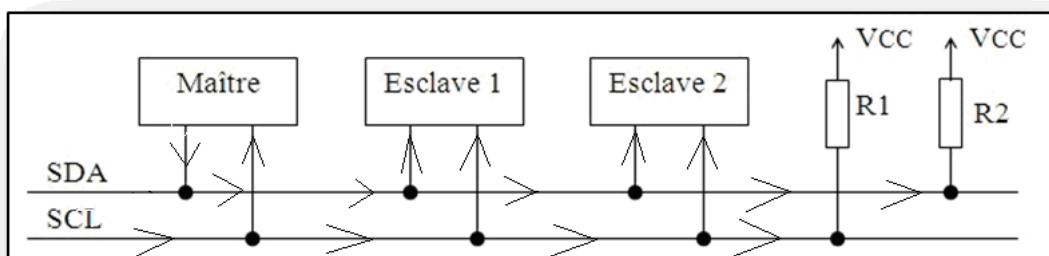
### c) Utilisation:

Le module servomoteur 23rête23e la liaison I2C qui est une liaison série synchrone fonctionnant en half duplex, le signal SDA ne peut aller dans un seul sens à la fois.

Le signal SDA est utilisé pour transmettre les données, et le signal SCL est utilisé pour transmettre un signal d'horloge synchrone.



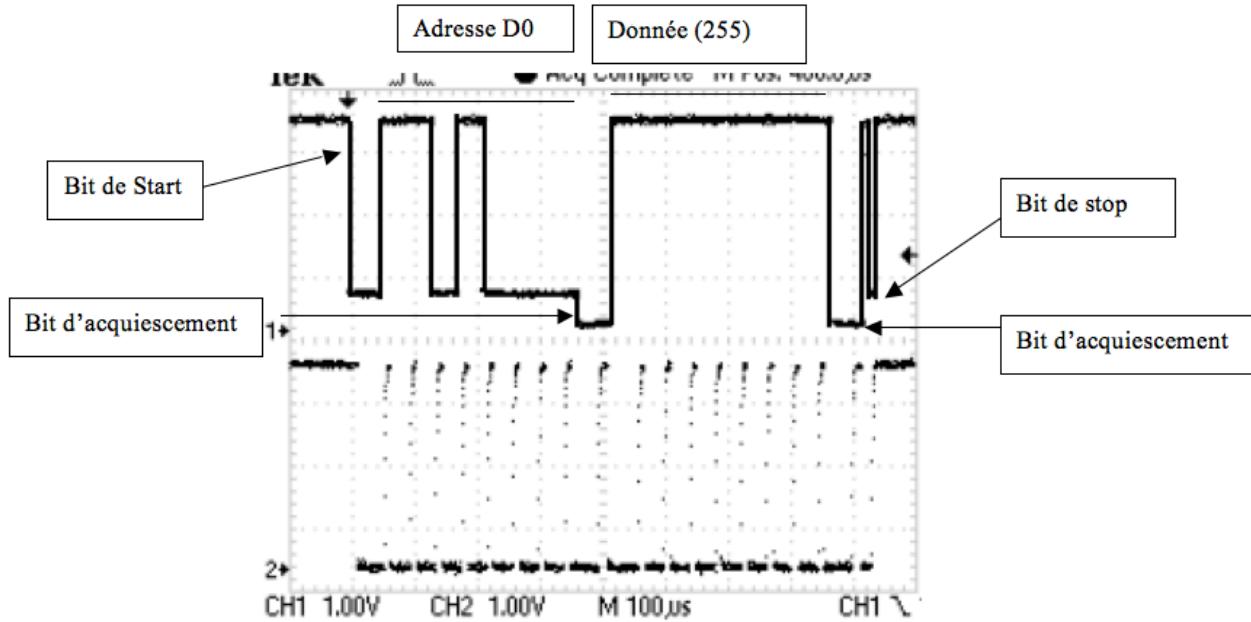
Le module servomoteur se 23rête ainsi sur le mOway.



L'esclave est sélectionné grâce au signal SDA.

Le rôle des résistances R1 et R2 est appelé « résistance de tirage ». Elles sont situées entre l'alimentation et un signal.

Elles ont pour rôle de garder un niveau logique haut sur ce même signal. Si aucune information n'est envoyée par les esclaves ou par les maîtres sur la liaison I2C, celle-ci sera au niveau logique haut grâce aux résistances de tirage.



Pour activer la sortie COM\_SERVO, la donnée à envoyer au PIC est « 255 ».

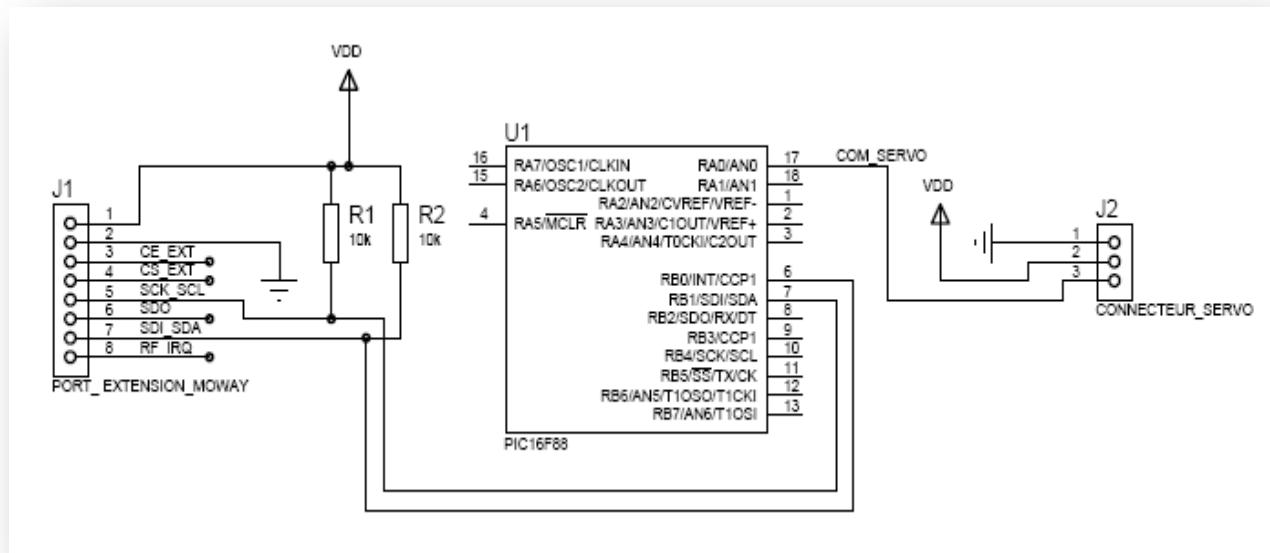
#### Commande directe du servomoteur par le mOway :

Nous programmons le mOway pour qu'il fixe le servomoteur dans une position extrême de 90° :

```
//-----//  
//-----//  
//  
//-----//  
//-----//  
//-----//  
  
//      P R O G R A M M E      P R I N C I P A L  
//  
//-----//  
//-----//  
//-----//  
  
void main(void)  
{  
while(1)  
{  
    SCK_SCL=1;           // Envoi d'un impulsion à SCK pour le faire passer à 1  
    Delay10TCYx (200);  //Envoi de l'impulsion de 2 ms (soit un angle de 90° pour le  
                        //servomoteur)  
    SCK_SCL=0;           //Arrêt de l'envoi d'impulsion à SCK  
    Delay10KTCYx(1);   //Ajout d'un delai de 10 ms pour éviter un redéclenchement immédiat  
                        //du programme  
}
```

Nous n'utiliserons pas le PIC18F87J50 du mOway car si on l'utilise, il ne pourra rien faire d'autre lorsque le servomoteur sera en fonctionnement. Par exemple, nous ne pourrons pas capturer d'images.

Nous prendrons un PIC16F88 que nous connecterons à l'aide d'une carte externe.



Avant de commencer la programmation, on s'informe sur le fonctionnement à l'aide du fichier « servo.c ».

L'adresse du module du servomoteur est « 0xD0 ».

On retrouve cette adresse dans la ligne :

`« #define ADRESSE_ESCLAVE 0xD0 »`

Le signal envoyé par le servomoteur se relève sur la pin « COM\_SERVO ».

On transfert maintenant le programme dans le PIC16F88 :

Le programme, une fois transféré, on retrouve sur la patte COM\_SERVO une de 3.5V, soit VDD.

Pour mettre au niveau logique 1 les signaux SDA et SCL, on utilise les commandes suivantes en langage C :

Pour le SDA : On tape SDI\_SDA\_TRIS=1 ;

Pour le SCL : On tape SCK\_SCL\_TRIS=1 ;

Pour activer la sortie COM\_SERVO, on envoie la donnée « 255 ».

Pour la désactiver cette sortie, on envoie « 254 ».

On nous donne :

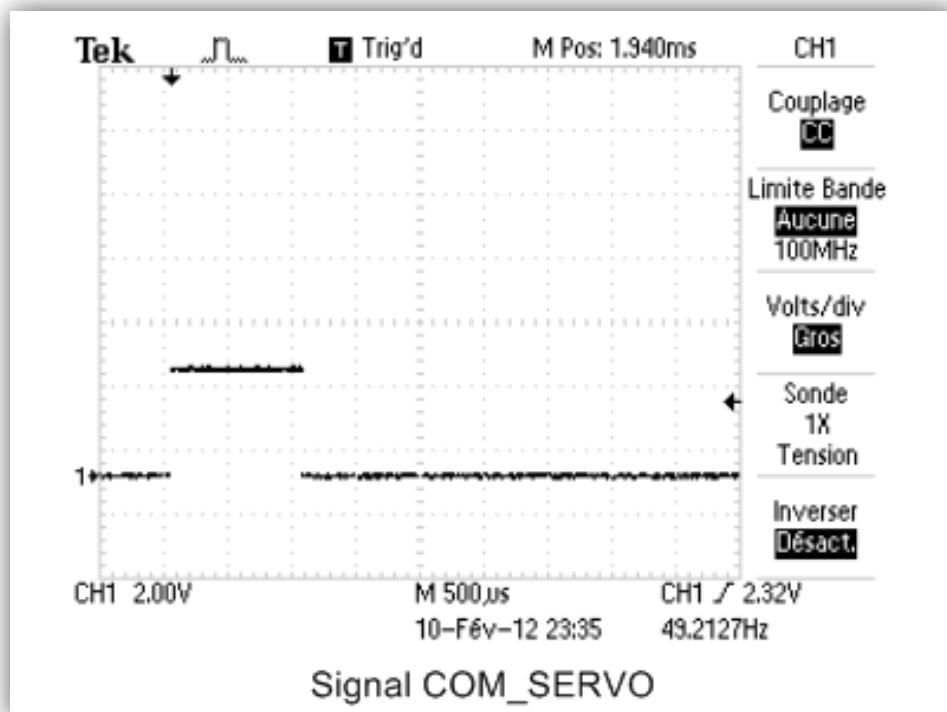
Si  $0 \leq \text{donnée} \leq 250$ , alors le réglage du Ton :  $\text{TON} = 4\mu\text{s} * \text{donnée} + 1\text{ms}$

Pour fixer TON à 1.25ms, il faut envoyer 63 en donnée :

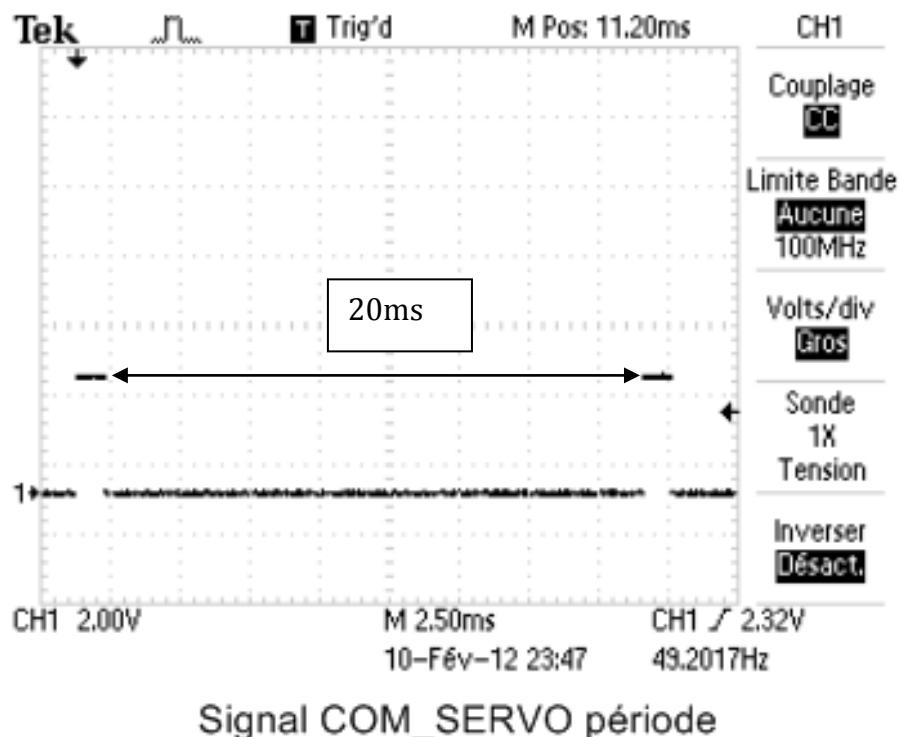
$$63 \times 4\mu\text{s} = 252\mu\text{s} + 1\text{ms} \approx 1.25\text{ms}$$

On obtiendra ainsi la position de 1.25 ms pour le servomoteur.

On crée maintenant un programme afin d'activer la sortie COM\_SERVO, on relève ensuite ce signal à l'aide d'un oscilloscope :



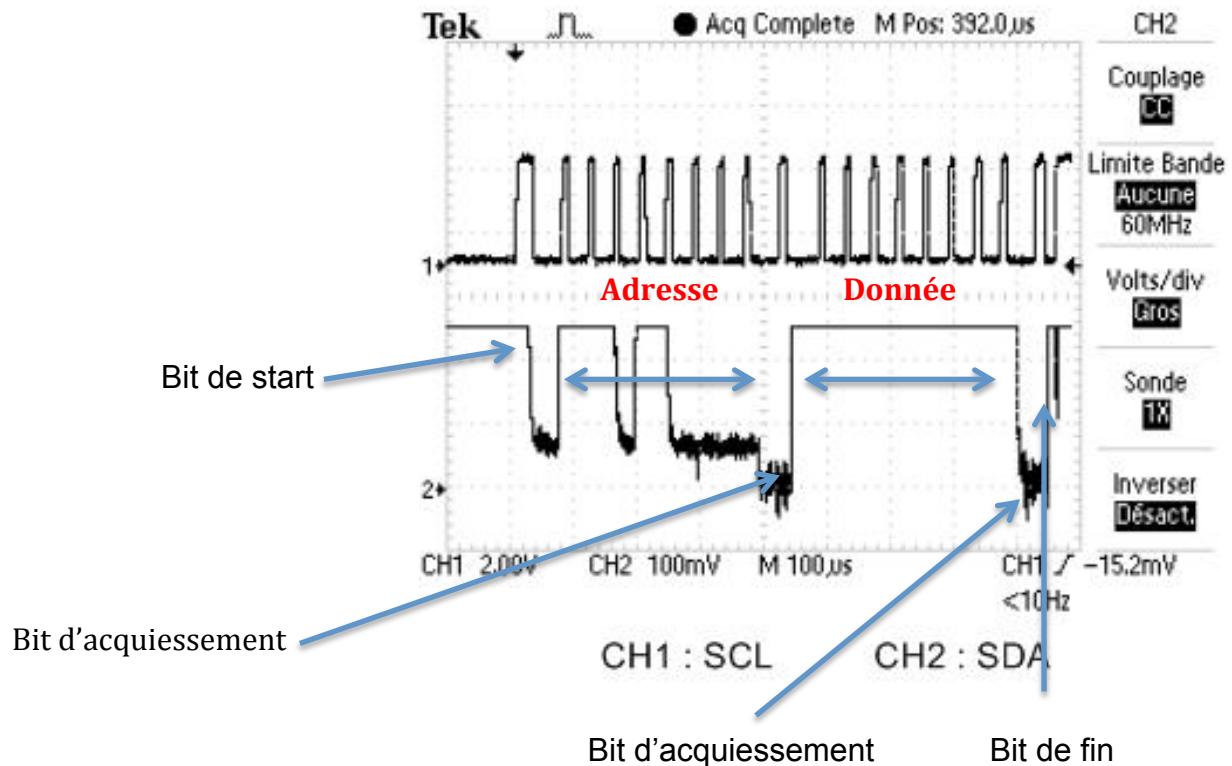
Le robot envoie la donnée 255 au PIC16F88 c'est alors que celui-ci active COM\_SERVO avec  $T_{on} = 1ms$  et une période par défaut de 20 ms.



Pour obtenir ce résultat, nous avons programmé le robot avec le programme ci-dessous :

```
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
  
PROGRAMME PRINCIPAL  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
//-----//  
  
void main(void)  
{  
PAUSE_SECONDE(5);  
Config_I2C1(); // Configuration I2C  
Start_I2C1(); // Démarrage I2C  
Write_I2C1(0xD0); // Envoi de l'adresse à l'esclave correspondant  
IN_ACK_NACK1(); // Lecture de l'acquiescement  
Write_I2C1(255); // Activation de la sortie COM_SERVO  
IN_ACK_NACK1();  
Stop_I2C1(); // Arrêt de l'I2C  
while(1); // Boucle infinie  
}
```

On relève les signaux SCL et SDA en concordance :



## 2) Etude de la caméra :

Comme nous l'avons vu précédemment :

- La profondeur de champs est d'environ 7 mètres ;
- La résolution (ou définition) est la dimension de l'image capturée par la caméra (en pixel largeur/hauteur) : 628x586

Nous avons ensuite évalué les distances pour lesquelles les informations transmises sont satisfaisantes. Nous avons relevé que pour distinguer correctement une serrure forcée, il faudrait que le robot soit placé à moins de 3 mètres de celle-ci.

Ensuite nous avons vérifié le fonctionnement du récepteur vidéo pour transmettre les images filmées sur un ordinateur. Le visionnement des images se fait par le biais du logiciel Yawcam qui nous permettra également par la suite d'envoyer les images sur une page web.

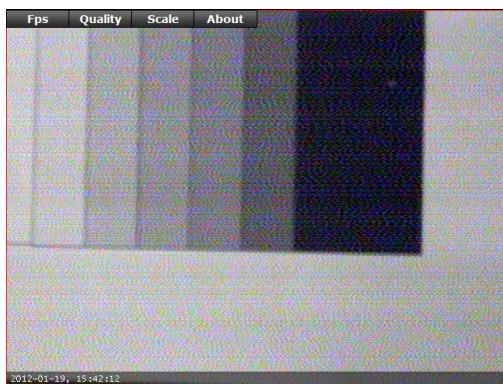
Les premières images obtenues n'étaient pas assez nettes pour pouvoir visionner quoi que ce soit. Après quoi, nous avons réglé l'objectif, nous permettant d'obtenir une image nette.

D'après les données du constructeur, nous savons que la caméra fonctionne avec un capteur CCD, ce principe transforme les photons lumineux en paires « électron-trou » par effet photoélectrique dans un substrat semi-conducteur, puis collecte les électrons dans un puits de potentiel maintenu à chaque photosite.

Son avantage est la meilleure qualité d'image, contrairement aux capteurs CMOS qui bénéficient d'un gain de place, et d'un besoin énergétique moindre.

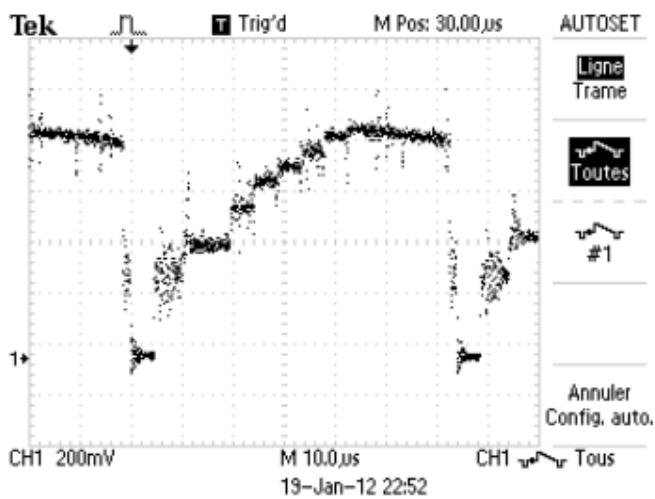
Pour comprendre en profondeur son fonctionnement, nous avons relevé les signaux de ligne et de trame de l'image en plaçant un mire de luminance devant le robot afin d'observer les transformations du signal.

Au préalable nous avons recherché les fréquences de ces signaux pour vérifier la cohérence des résultats que nous obtiendrons. La fréquence image est de 25Hz, celle de ligne est de 15,625kHz, et la fréquence de trame est de 50Hz.



Mire de luminance filmée par le mOway à l'aide de Yawcam

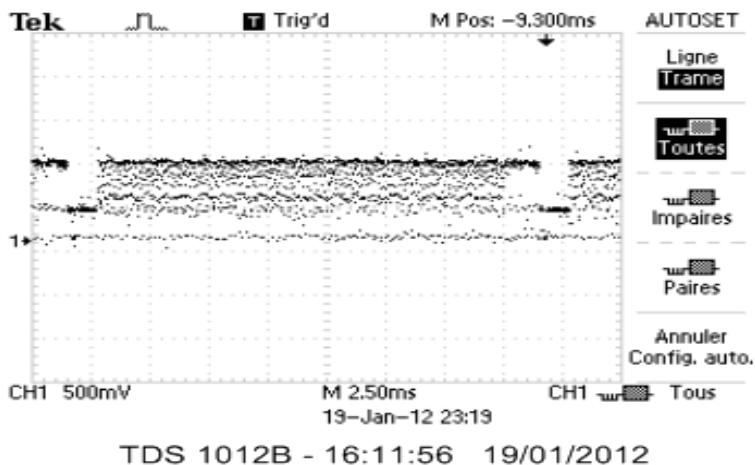
L'oscilloscope est réglé sur  $10\mu\text{s}/\text{div}$  pour visualiser le signal de ligne :



TDS 1012B - 15:44:54 19/01/2012

Chaque « marches » de l'escalier du signal ligne correspond aux différentes teintes que filme la camera (du noir au blanc). La première marche est une ligne de synchronisation.

L'oscilloscope est réglé sur  $2.5\mu\text{s}/\text{div}$  pour visualiser le signal de trame :



TDS 1012B - 16:11:56 19/01/2012

De plus on remarque qu'en plaçant la mire de luminescence à l'horizontale, le signal trame devenait celui de ligne, et vice versa.

Le module de réception vidéo convertit le signal vidéo analogique reçu en fichier numérique compressé. L'objectif de cette activité est de découvrir l'intérêt et le principe de compression des images et des vidéos. En sachant que l'espace mémoire nécessaire d'une image de 1200X960 pixels dont chaque pixel est codé sur 24 bits et de 21.184Mo. La compression est vivement conseillée car l'espace mémoire utilisé est très important, surtout pour une image.

Pour compresser une image on doit utiliser un dispositif appelé codec, Un codec est un procédé capable de compresser et/ou de décompresser un signal numérique. Ce procédé peut être effectué par un circuit imprimé ou un logiciel. Dans l'image, on en distingue deux principaux, le JPEG et le MPEG. Le codec JPEG étant surtout utilisé pour les photos et le MPEG pour la vidéo.

Cependant la compression JPEG apporte un avantage que l'on ne peut ignorer. L'image fournie est généralement de bonne qualité, et la taille reste faible. Nous nous sommes donc intéressé au codec M-JPEG qui fonctionne de la même manière mais pour la vidéo.

Voici les étapes de la compression JPEG :

- Découpage en blocs de 64 (8x8), ou 256 (16x16) pixels ;
- Transformation de couleur ;
- Sous-échantillonnage ;
- DCT, transformation numérique appliquée à chaque blocs et pour chaque couleur ;
- Quantification, étape de compression (diminue l'amplitude des hautes fréquences non-perceptible par l'œil) ;
- Codage RLE et Huffman.

On remarque très nettement une différence de taille lorsque l'on convertit des images .BMP en .JPG :

Nom	Taille	Type	Date de modification
Ligne.bmp	30 Ko	Image bitmap	14/09/2011 14:47
Neuf points éloignés.bmp	30 Ko	Image bitmap	14/09/2011 13:51
Neuf points rapprochés bis.bmp	30 Ko	Image bitmap	14/09/2011 13:54
Neuf points rapprochés.bmp	30 Ko	Image bitmap	14/09/2011 13:53
Point.bmp	30 Ko	Image bitmap	14/09/2011 13:20
Ligne.JPG	2 Ko	Image JPEG	20/01/2012 12:36
Neuf points éloignés.JPG	2 Ko	Image JPEG	20/01/2012 12:36
Neuf points rapprochés bis.JPG	1 Ko	Image JPEG	20/01/2012 12:36
Neuf points rapprochés.JPG	1 Ko	Image JPEG	20/01/2012 12:37
Point.JPG	1 Ko	Image JPEG	20/01/2012 12:37

On a d'ailleurs effectué des tests sur de simple pixel, pour voir la différence avant et après compression.



L'image .BMP est plus nette

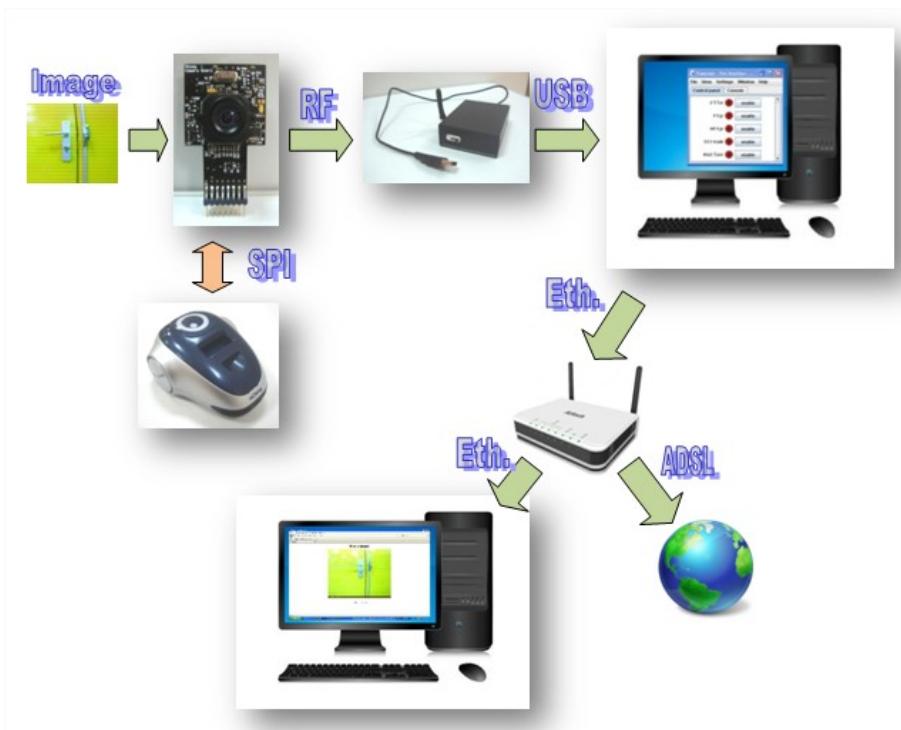
Sur l'image .JPG, on remarque en zoomant un léger nuage de pixels

### 3) Transmission des données :

Pour diffuser les images de la caméra sur le web, on utilise le logiciel Yawcam en mode streaming.

Le streaming est la diffusion d'information sous forme de flux, c'est-à-dire de manière continue, en direct ou en léger différé. Cependant la rapidité de la transmission dépend de la carte réseau de l'ordinateur.

En utilisant *U Driver* on note un débit d'émission de 3,9 Mbps.



On a tout de même remarqué certains changements parfaitement clairs.  
 Plus on fait bouger le mOway (donc, plus il capture d'images), plus ce débit est faible.  
 Ce phénomène est totalement normal car le nombre d'informations à traiter à la seconde est beaucoup trop important.

Sur le pc client on retrouve trois paramètres vidéo modifiables :  
 -FPS (Frame Per Seconde) ;  
 -Quality ;  
 -Scale.

(En français IPS (Image Par Seconde), qualité et échelle)

L'IPS correspond donc au nombre d'image par seconde en vidéo il est généralement de 25 images par seconde, et pour notre caméra également car sa fréquence d'image est de 25Hz.

La qualité sert à définir le nombre de pixel qui sera affiché à l'écran.  
 L'échelle, quant à elle, détermine l'échelle de la fenêtre vidéo.

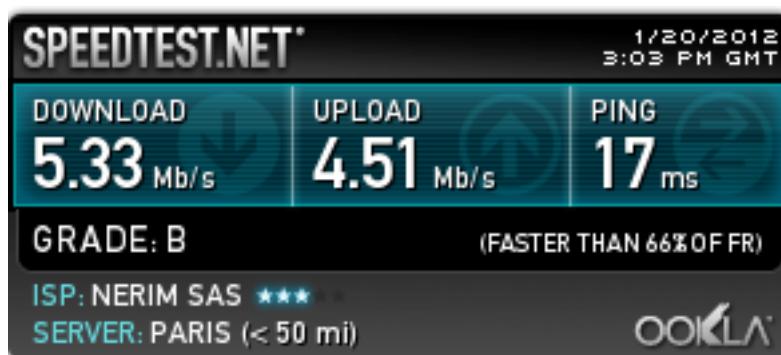
Il existe aussi d'autres types de transmissions réseaux tel que l'ADSL.

L'ADSL est une technique de communication numérique de la famille xDSL. Elle permet d'utiliser une ligne téléphonique, une ligne spécialisée, ou encore une ligne RNIS, pour transmettre et recevoir des données numériques de manière indépendante du service téléphonique conventionnel.

Il y a aussi l'Universal Mobile Télécommunications System (UMTS) qui est l'une des technologies de téléphonie mobile de troisième génération (3G) européenne. C'est l'un des principaux réseaux utilisé par les smartphones.  
 (La vitesse de transmission offerte par les réseaux UMTS est 2 Mb/s).

Nous avons voulu comparer les vitesses d'émissions ADSL et streaming dont nous disposons.

On remarque que la transmission ADSL est plus rapide (4,5 contre 3,9 Mb/s) :



Pour finir, nous avons mis en place une page web où les images filmées par le mOway sont retransmises en direct avec un taux de rafraîchissement réglé par nos soins.

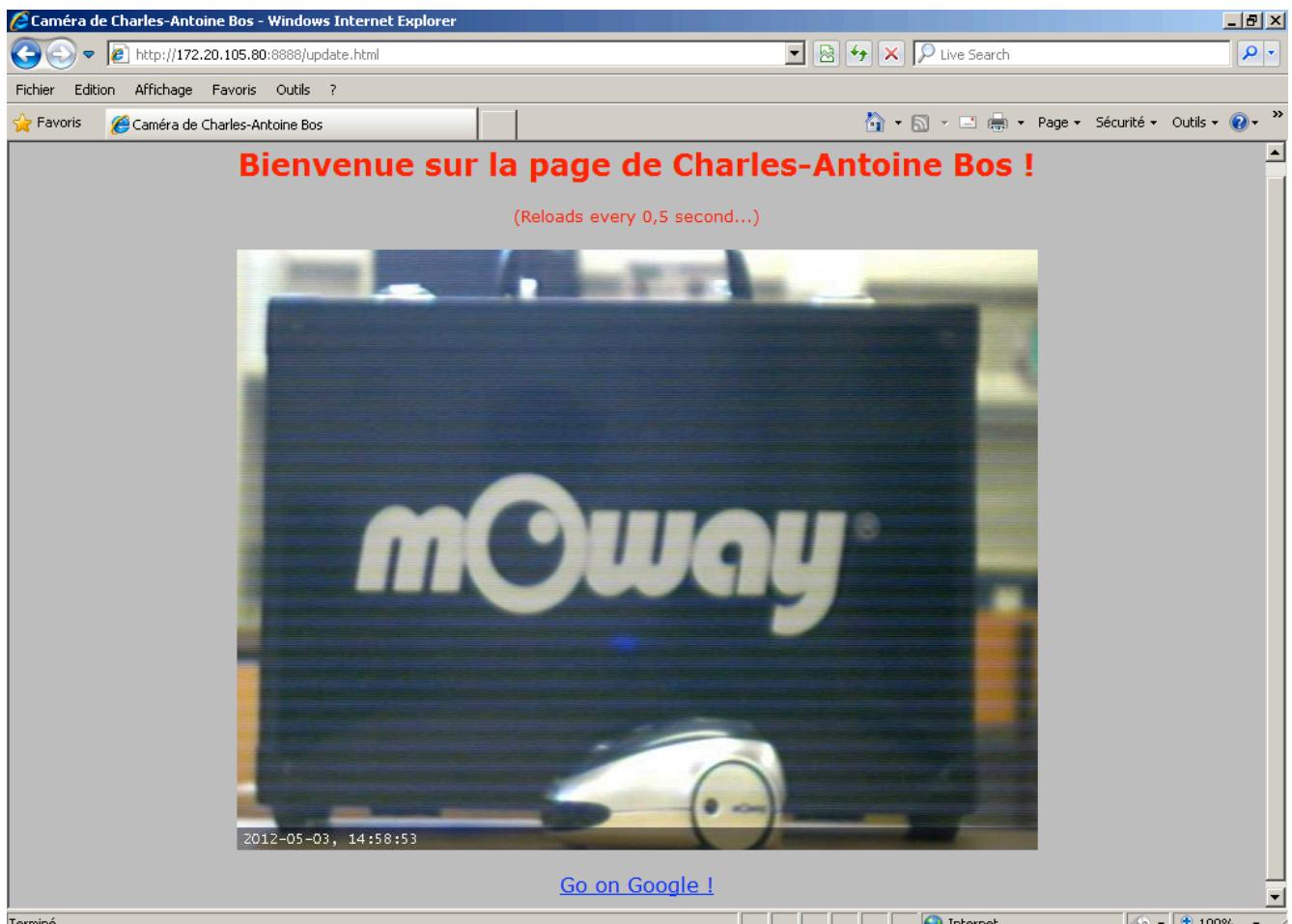
Nous avons également personnalisé cette page web à l'aide du fichier HTML le permettant.

Grâce à cela, nous avons pu configurer le nom de la page web, la mise en page, la couleur du fond et des polices, ainsi que d'ajouter d'autres liens.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <BODY BGCOLOR=silver>
      <Font Color="#FF0000">
        <title>Caméra de Charles-Antoine Bos</title>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <meta http-equiv=Refresh content="0,5">
      </head>

    <body>
      <div align="center">
        <p><strong><font size="5" face="Verdana, Arial, Helvetica, sans-serif">Bienvenue sur la page de Charles-Antoine Bos !</font></strong></p>
        <p><font size="2" face="Verdana, Arial, Helvetica, sans-serif">(Reloads every 0,5 second...)</font></p>
        <p></p>
        <p><a href="http://www.google.com" target="_blank"> <font face="Verdana, Arial, Helvetica, sans-serif">Go on Google !</font></a></p>
      </div>
    </body>
  </html>
```

Voici le résultat obtenu :



# IV – Problématiques individuelles

## 1) La navigation :

### a) Introduction :

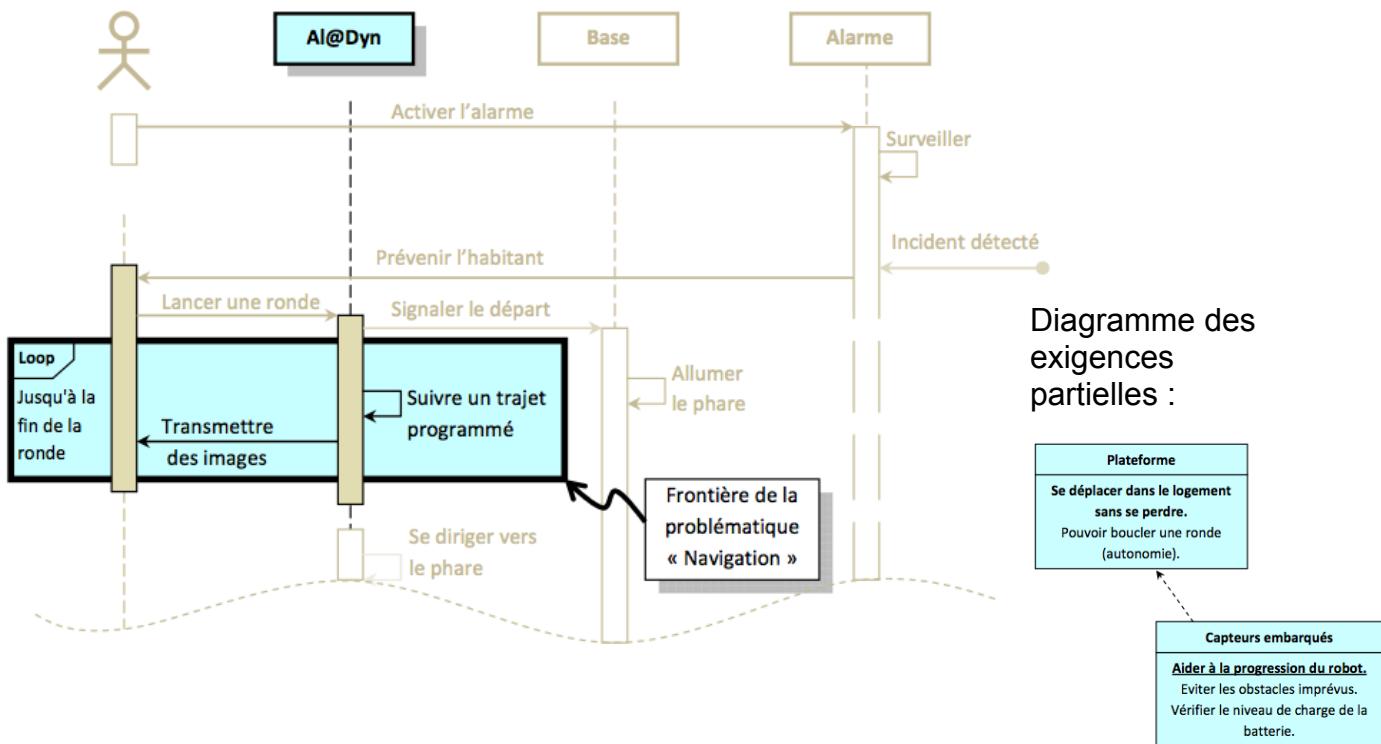
AI@Dyn est un robot équipé d'une caméra de surveillance. L'habitant peut piloter AI@Dyn à distance tout en visionnant les images de son logement. Si l'habitant ne souhaite pas piloter le robot mais vérifier rapidement une ouverture (porte ou fenêtre) ou un objet particulier, il peut lancer une ronde. Dans ce cas AI@Dyn exécute un parcours préprogrammé tout en diffusant les images, fait une halte en face de l'ouverture ou de l'objet à vérifier puis retourne à sa base. Cette dernière opération peut être automatique, si AI@Dyn est assez près de la base pour la détecter.

Plutôt que d'offrir à l'utilisateur une ronde très longue qui couvre tout le logement, il est préférable de lui proposer plusieurs rondes plus courtes mais plus ciblées.

Chaque ronde est ponctuée d'une ou deux haltes pour observation de l'ouverture ou de l'objet particulier à vérifier.

Cette solution apporte un confort à l'utilisateur qui peut choisir la ronde qu'il désire et permet aussi de réduire la dispersion du robot, puisque ce dernier se positionne sur sa base après chaque ronde.

### Diagramme de séquence :



**Cahier des charges :**

Le robot doit partir de sa base exécuter une ronde type et revenir à sa base. L'erreur admissible est un couloir virtuel de 80cm de large (ce qui correspond aux pas de porte) dont l'axe est la ronde idéal. En fonction des moyens dont vous disposez en classe, proposez une ronde type. Vous devez y faire apparaître clairement la base, la ronde idéale, le couloir toléré et les deux haltes numérotées pour observation d'une ouverture ou d'un objet.

Le robot doit exécuter sa ronde le plus rapidement possible pour atteindre les haltes sur lesquels il doit s'arrêter 5 secondes. La halte intermédiaire peut être réalisée à l'aller ou au retour au choix.

**a.1) Histoire de la navigation :**

Les premiers navigateurs utilisaient les étoiles pour se repérer. Les hommes ont ensuite inventé toutes sortes d'instruments pour s'orienter mais la boussole est apparue quatre siècles avant J-C en Chine. Elle a permis aux hommes de s'orienter et d'explorer le monde.

Il y a eu aussi le compas qui est un instrument de navigation qui donne une référence de direction (le Nord) sur le plan horizontal et permet ainsi la mesure d'angles horizontaux par rapport à cette direction. Le compas est gradué de  $0^\circ$  (nord) à  $359^\circ$  dans le sens des aiguilles d'une montre.

La navigation est le domaine principal où l'utilisation du compas est essentielle.

A l'heure actuelle, on utilise le GPS (Global Positioning System) pour se localiser à l'aide de satellites. On utilise 4 satellites pour effectuer une géolocalisation :

- 3 pour la triangulation (Indique 2 positions possibles du récepteur)
- 1 pour le repère de temps (Position exacte)

**a.2) Comparatif des moyens de localisation :****La boussole :**

La boussole est un appareil qui permet la localisation du Nord magnétique terrestre. Elle est constituée d'un dispositif utilisant une aiguille aimantée tournant librement sur un pivot.

Un boussole est donc un aimant : La Terre se comporte comme un gros aimant, et un plus petit aimant, comme celui qu'est la boussole, a tendance à s'aligner sur les lignes de champ magnétique créé par l'aimant le plus important. Il se trouve que ces lignes de champ sont orientées sur un axe Nord-Sud. La boussole indique donc la direction du Nord et du Sud.

En pratique, une boussole n'indique pas exactement le Nord. Elle pointe en fait vers ce qu'on appelle le « Nord magnétique ». C'est un point assez proche du pôle Nord.

La déclinaison « magnétique terrestre » correspond à la différence entre le Nord magnétique et le Nord géographique. Sa valeur est de l'ordre de 2 degrés à Paris.

Le champ magnétique d'un aimant se mesure en Tesla. Par exemple, la valeur moyenne de l'intensité du champ magnétique terrestre à Paris est de  $47 \mu T$ .

La boussole numérique, quant à elle, fonctionne à l'aide de 2 capteurs (magnétorésistance ou capteurs à effet Hall).

La magnétorésistance est une résistance qui varie en fonction du champ magnétique. Les capteurs à effet Hall, eux, délivrent une tension  $V_h$  lorsqu'un courant  $I_0$  traverse un matériau conducteur ou semi-conducteur, et si un champ magnétique d'induction  $B$  est appliqué perpendiculairement au sens de passage du courant.

L'inconvénient majeur est que la boussole numérique pourrait être perturbée lorsqu'elle passerait près d'un fil, dû au champ magnétique créé par ce dernier.

On en trouve aux alentours de 30€ sur internet :



### Boussole miniature compensée CMPS10

Code 24499

Le module CMPS10 est capable de détecter le nord a été conçu pour le guidage de robots. Il est compensé par rapport à son inclinaison.

Commander
Ajouter au panier

Article en stock

Quantité :

30,52 € HT ~~36,50 € TTC~~

### Le gyroscope :



Le gyroscope fut inventé en 1852 par Léon Foucault. Il indique une direction fixe dans l'espace. Un gyroscope vient de « gyro » (signifie rotation) et « scope » (signifie visualisation), il permet donc de visualiser la rotation terrestre par exemple.

La propriété d'un gyroscope est la fixité dans l'espace, c'est à dire que le rotor, en raison de son inertie (c'est la résistance d'un corps à une variation de vitesse), est fixe dans l'espace si l'on prend soin de le suspendre convenablement. En effet, le plan de rotation ne varie pas.

L'effet gyroscopique est donc la force exercée par un objet en rotation en réponse à un changement d'axe de rotation. De ce fait, un objet qui tourne a tendance à rester dans la même position.

On trouve également des gyroscopes électroniques. Ils sont beaucoup utilisés en aéromodélisme.

Cette solution peut être envisagée pour le projet AI@Dyn.  
On en trouve à tout prix :

- Du bas de gamme :



**Gyroscope Nano Head Lock (servo digital) modelcraft GR302**

★★★★★

Soyez le premier à écrire un avis

**MODEL CRAFT**

■ Idéal pour les petits hélicoptères électriques  
■ Menu spécial pour servos numériques

Petit par ses dimensions, mais grand par ses performances. Une qualité de vol parfaite est assurée par le système électrique piezo avec équilibrage automatique...

[Voir les détails du produit](#)

**-37%**

**79.90€**  
**49.95€**

Eco-part : 0.00 €

1

**Ajouter au panier**

**En stock** **Modes de paiement**

- Au haut de gamme :



**Gyroscope Robbe HC3-SX**

★★★★★

Soyez le premier à écrire un avis

■ Avec dispositif de sécurité

Le HC3-SX est un article à la pointe de la technologie de la marque Robbe. Il permet de commander tous les types d'hélicoptère standards. Le HC3-Xtreme...

[Voir les détails du produit](#)

**479.00€ \***

Eco-part : 0.00 €

1

**Ajouter au panier**

**En stock** **Modes de paiement**

**La localisation Wi-Fi :**

La localisation Wi-Fi s'inspire du GPS (qui signifie Global Positioning System). Le système GPS est composé de 24 satellites en orbite autour de la Terre à 20200 Km d'altitude.

Les satellites envoient des ondes magnétiques contenant la position du satellite, l'heure d'envoi, ... Il faut 4 satellites pour connaître la position, les 3 premiers pour la triangulation (Il existe mathématiquement 2 positions avec 3 repère : une dans l'espace, l'autre sur Terre. On exclut donc logiquement la position dans l'espace) et le 4<sup>ème</sup> car le récepteur GPS ne peut calculer la durée de l'envoi. Ce système nécessite la précision de l'horloge atomique car les ondes magnétiques se déplacent à la vitesse de la lumière et une erreur d'un millième de seconde se traduirait par une erreur de 300 Km.

Pour ce qui concerne le Wi-Fi, il faut savoir que le signal Wi-Fi est un ensemble de protocoles de communication sans fil. Un réseau Wi-Fi permet de relier sans fils plusieurs appareils informatiques au sein d'un réseau d'informatique afin de permettre la transmission de données entre eux.

Il suffit de 3 bornes Wi-Fi pour se localiser, une 4<sup>ème</sup> n'étant pas nécessaire car nous n'avons plus besoin de la référence de temps. En effet, les bornes Wi-Fi ne sont pas assez puissantes pour provoquer une erreur de 300 Km.

Il faudra donc 3 émetteurs pour localiser un récepteur. Il faut également que ce dernier soit également émetteur car il pourrait envoyer des informations aux bornes Wi-Fi, lui permettant une localisation par celle-ci.

**Synthèse :**

En conclusion, nous avons pu dresser un tableau des avantages et des inconvénients de chaque système :

	Boussole numérique	Gyroscope numérique	Localisation wifi
Avantages	Peu onéreux, Permet de trouver le nord	Permet de garder n'importe quel cap, Prix accessible	Très précis
Inconvénients	Moins précis, Facilement dérégliable par les champs magnétiques	Peut se dérégler, Nécessite une reprogrammation pour tout changement de direction	Cher, Installation complexe, De nombreuses contraintes

### b) Solution retenue :

Malgré l'incontestable précision du GPS, le moyen le plus simple pour déterminer la direction à prendre reste encore la boussole magnétique. Nous prendrons une boussole dont la référence est CMP03. Dans cette réalisation, le montage ne fait pas appel à une aiguille aimantée mais à un équivalent électronique reposant sur l'emploi de deux circuits à effet de Hall sensibles au champ magnétique terrestre.

Ce petit module est capable de détecter le Nord magnétique terrestre grâce à l'emploi de 2 capteurs spécialisés montés en angle à 90° et, par déduction, de nous indiquer son orientation par le biais d'un échange d'information via un bus I2C™ ou d'un signal "PWM" (fournissant une impulsion comprise entre 1 ms (0°) et 36,99 ms (359,9°) avec une résolution d'environ 100 µs / °).

Idéalement conçue pour les applications liées à la robotique ludique cette boussole pourra par exemple être très facilement interfacée avec la plupart des microcontrôleurs.

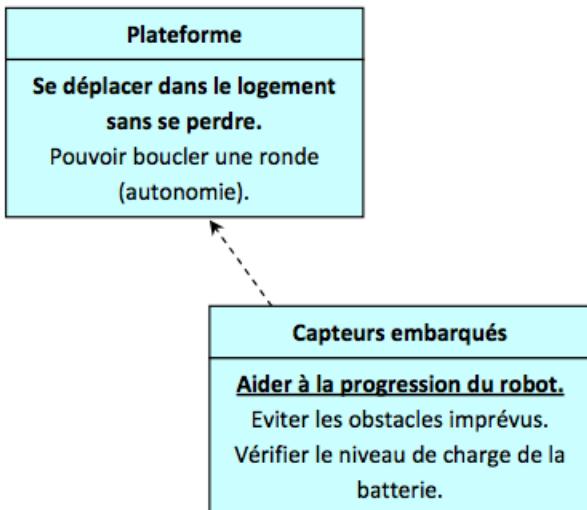
Pour une plus grande facilité, le bus "I2C™" pourra également être utilisé pour récupérer les informations sous forme d'un mot binaire compris entre 0 et 255 ou 0 et 3599 (correspondant ici à 0 et 359,9 °).

Pour ce faire, il suffira d'adresser le module à la manière d'une mémoire série qui vous "répondra" en vous renvoyant la valeur mesurée. Afin d'augmenter la précision de la mesure, le module dispose également d'une procédure de calibrage et d'une intégration de l'influence des perturbations liées aux réseaux 50/60Hz.

#### **Caractéristiques techniques du CMP03:**

- Alimentation: + 5 Vcc.
- Consommation: 20 mA.
- Résolution: 0.1 °.
- Précision: 3 à 4 ° env.
- Dimensions: 32 x 35 mm.





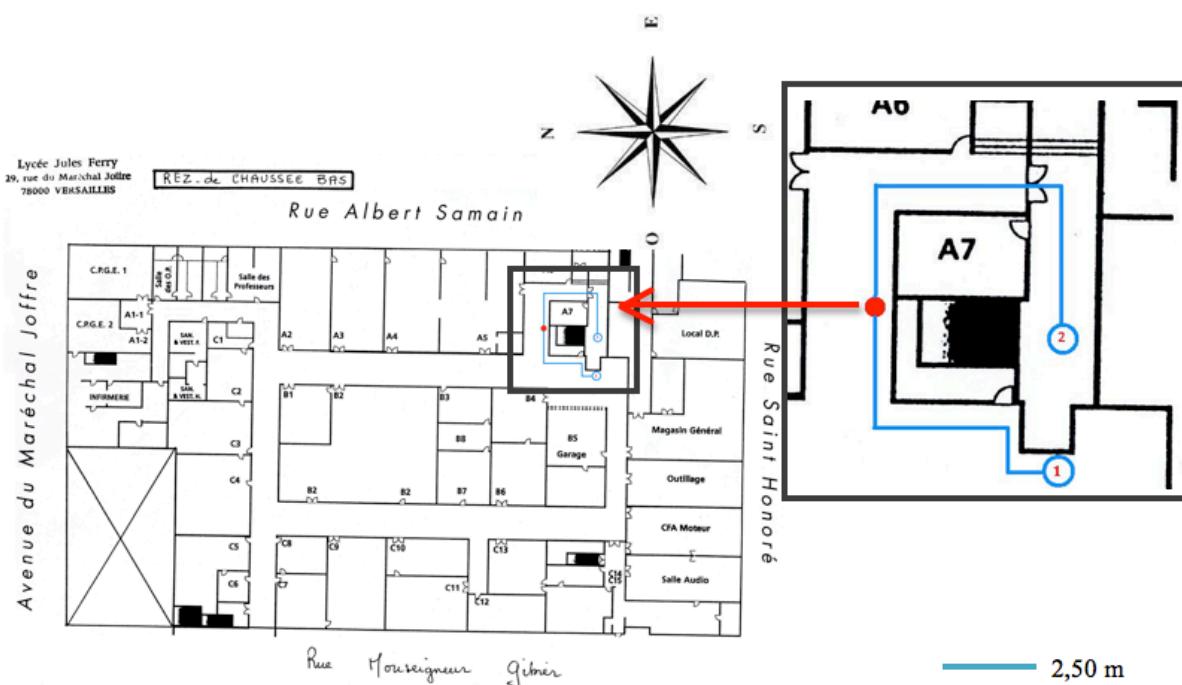
Nous devons donc permettre au robot de se déplacer dans le logement sans se perdre mais également de corriger sa trajectoire en cas de décalage par rapport au trajet initial.



Nous avons mesuré les distances parcourues par le mOway lors des différentes rondes types qui nous étaient proposées ci-contre :

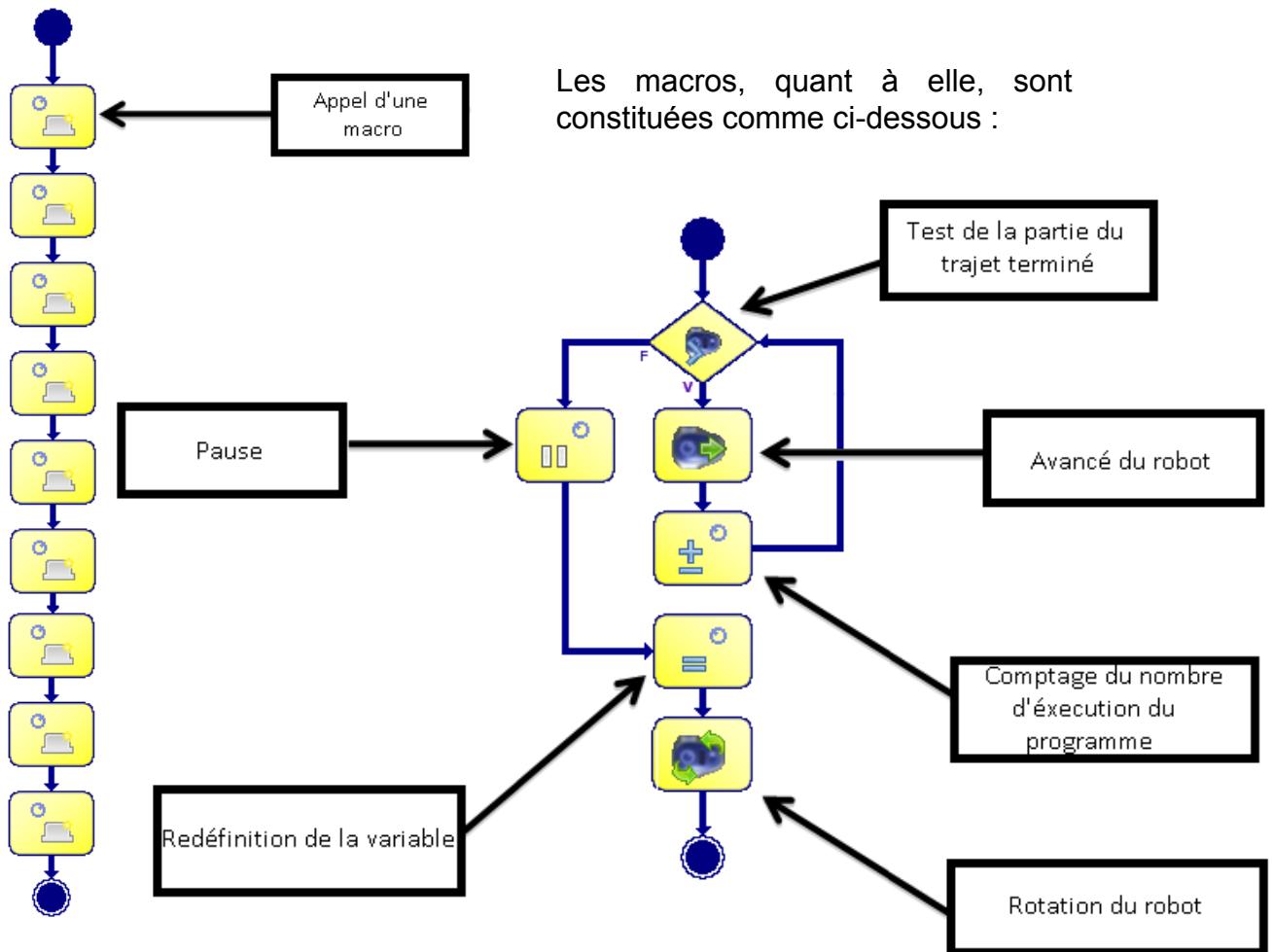
- Ronde Bleu : 17,5 m (10 virages)
- Ronde Jaune : 28 m (15 virages)
- Ronde Rouge : 25 m (14 virages)
- Ronde Verte : 24 m (12 virages)

Nous avons donc imaginé une ronde type dans notre lycée dont la distance est d'environ 40 mètres (Aller-Retour) et contenant 12 virages.

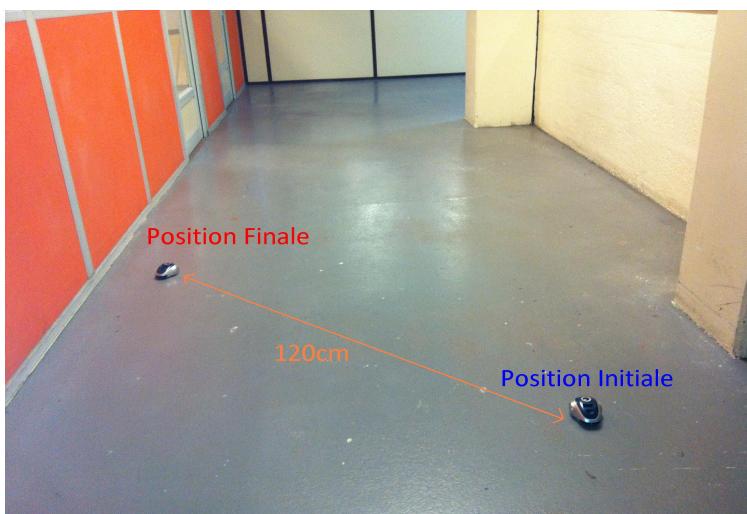


Nous avons réalisé un programme de ronde pour le robot mOway, afin qu'il effectue la ronde de surveillance. Ce programme est composé de 9 macros dont 2 composées de plusieurs étapes du programme à la fois, mOwayGuy ne permettant pas de créer plus de 9 macros.

Le programme général est celui-ci :



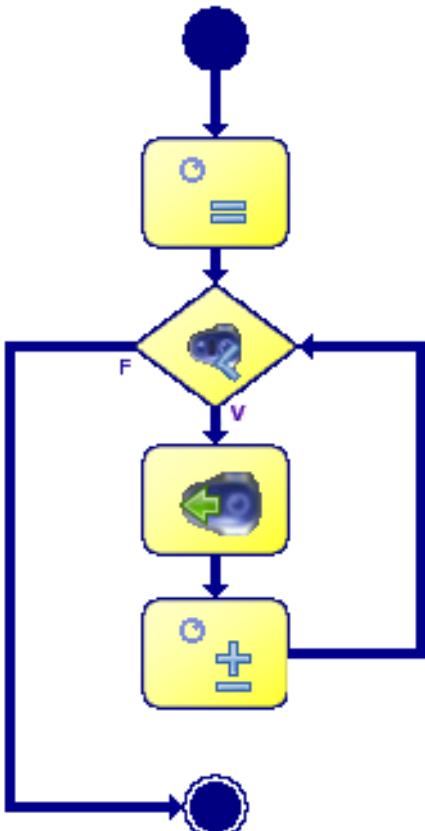
Une fois la ronde exécutée, un constat très net apparaît alors :



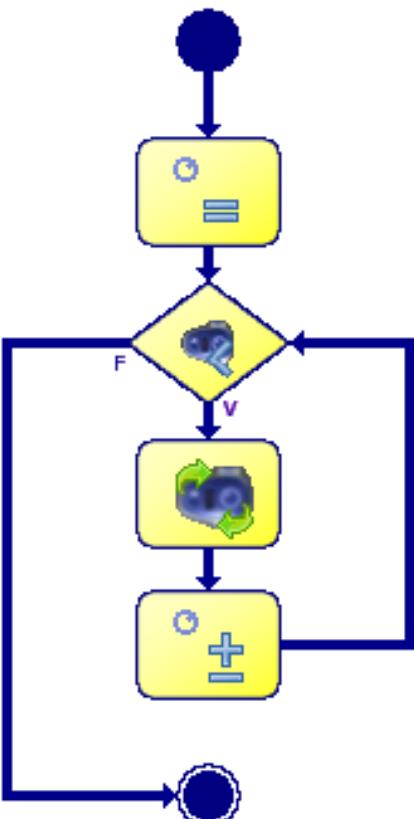
Le décalage entre la position initiale et la position finale est bien au dessus des 80 cm imposés par le cahier des charges (ici, 120 cm ont été relevé).

Pour comprendre d'où ce problème provenait, nous avons décidé d'effectuer deux tests pour essayer de trouver l'origine de ce décalage très important.

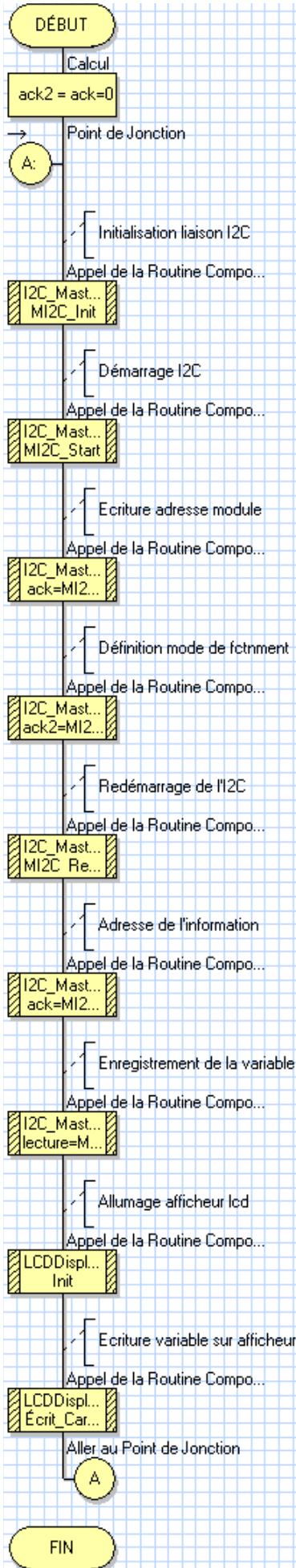
Nous avons pour cela tout d'abord testé le mOway en ligne droite :



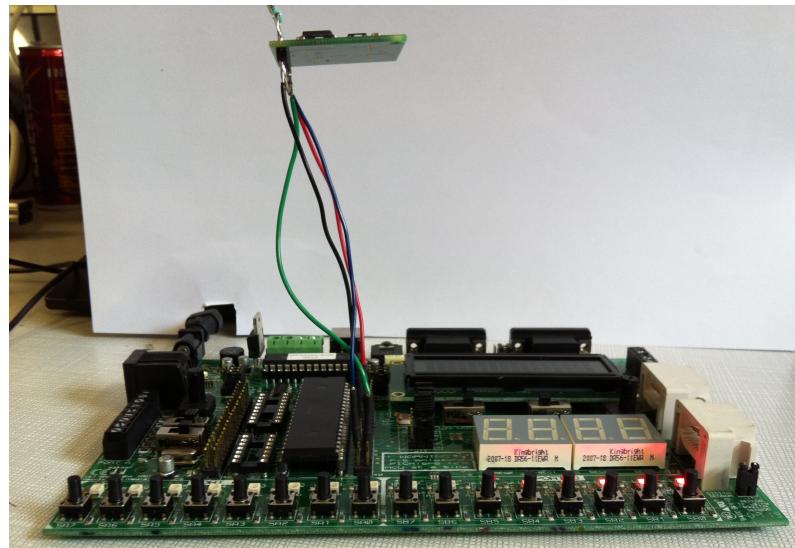
Ce programme nous permet de faire avancer le robot tout droit en permanence. Nous avons pu remarqué que le mOway se déporté légèrement d'un côté, cela étant dû à la nature et la qualité du sol mais également à l'adhérence des pneus. En y regardant de plus près, nous avons pu noté qu'il y avait un léger décalage entre les deux roues.



Ce programme, quant à lui, nous permet de faire pivoter le robot en permanence. Nous avons pu remarqué un très net décalage entre les 90° que nous lui avons demandé de faire et la rotation réellement exécuté par le mOway, cela étant dû à l'imprécision des virages, mais encore une fois de la nature et qualité du sol, de l'adhérence des pneus et de la différence de vitesse entre les deux roues

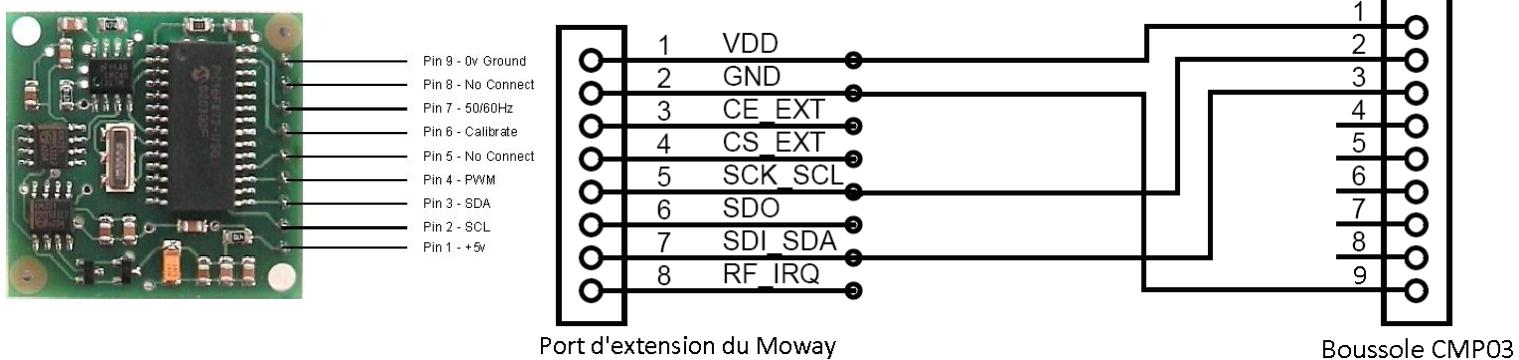


Nous avons ensuite testé la boussole numérique CMP03 sur une plaque évolutive possédant un écran LCD à l'aide du programme ci-contre réalisé à l'aide de FlowCode :



Ce programme nous permet de lire la valeur de l'angle par rapport au Nord magnétique terrestre grâce à une valeur comprise entre 0 et 255 en 8 bits et entre 0 et 3599 en 16 bits.

Nous avons ensuite branché le module boussole sur le mOway.

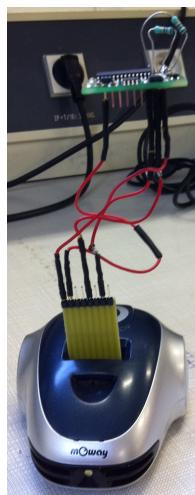


On a tout d'abord programmé le mOway pour qu'il allume une LED s'il détecte une réponse du module boussole lorsque l'on communique avec celui-ci :

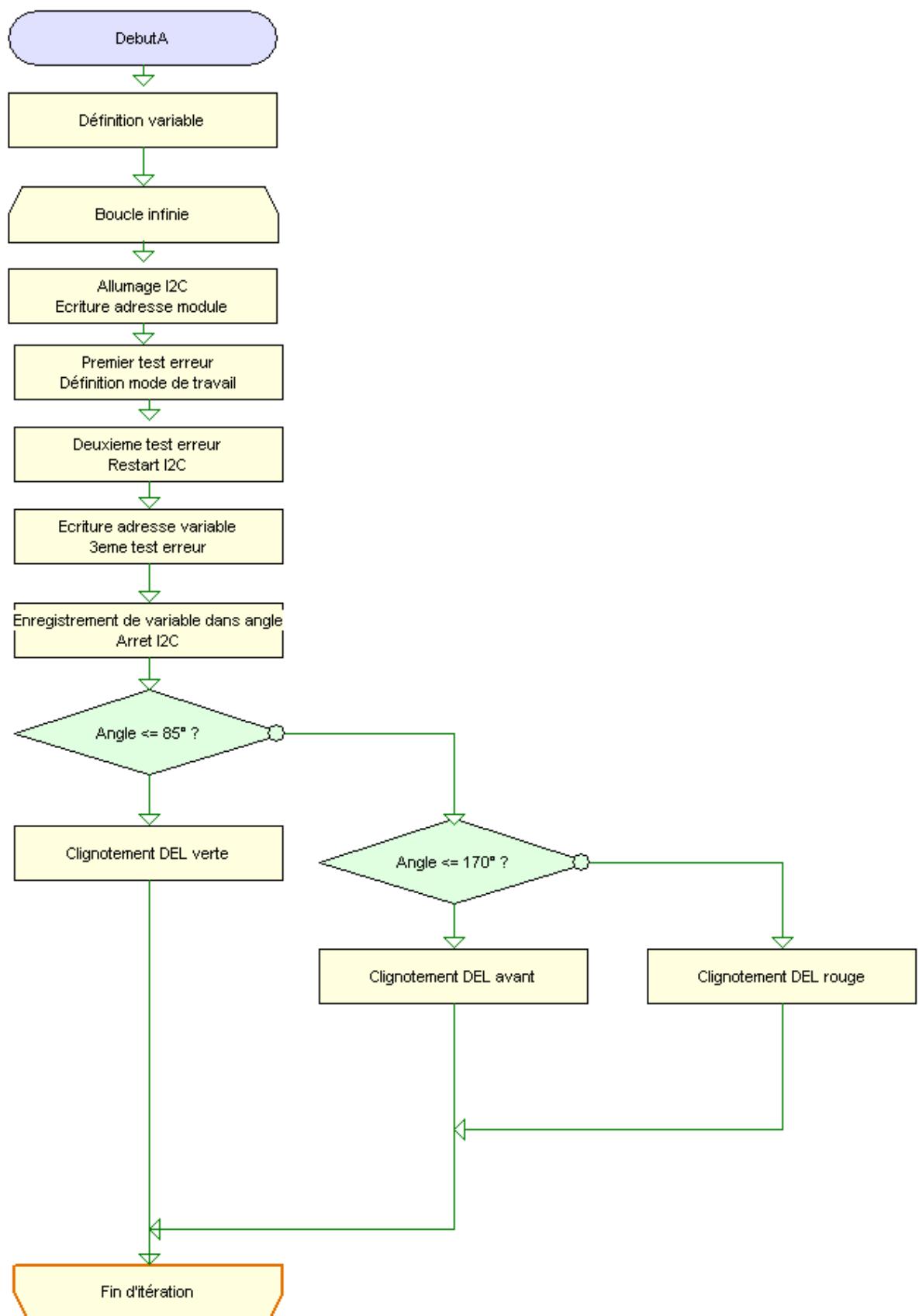
```

40
41 //*****[MAIN]*****
42
43 void PAUSE_SECONDE(unsigned char nb_secondes)
44 {
45     unsigned char i;
46     for (i=0;i<nb_secondes;i++)
47     {
48         Delay10KTCYx (100);      // Pause de 1 seconde //
49     }
50 }
51
52 unsigned char VALEUR_BOUSSOLE()
53 {
54     unsigned int angle=0;
55     unsigned int ack1=0;
56     unsigned int ack2=0;
57     unsigned int ack3=0;
58
59     angle=0;
60     Start_I2C1 ();
61     Write_I2C1 (0xc0);
62     ack1=IN_ACK_NACK1 ();
63     if (ack1) LED_BRAKE_ON;
64     Write_I2C1 (1);
65     ack2=IN_ACK_NACK1 ();
66     if (ack2) LED_BRAKE_ON;
67     Stop_I2C1 ();
68     Delay10KTCYx (10);
69     Start_I2C1 ();
70     Write_I2C1 (0xcl);
71     ack3=IN_ACK_NACK1 ();
72     if (ack3) LED_BRAKE_ON;
73     angle=Read_I2C1 ();
74     OUT_NACK1 ();
75     return angle;
76 }
77

```



A l'aide d'un programme, le mOway nous indiqua à l'aide des LED l'angle par rapport au Nord magnétique terrestre dans lequel il était :



```

79 //-----//  

80 //-----//  

81 //-----//  

82 // P R O G R A M M E           P R I N C I P A L //  

83 //-----//  

84 //-----//  

85 //-----//  

86 void main()  

87 {  

88  

89     unsigned char mesure=0;  

90     unsigned int angle=0;  

91     unsigned int ack1=0;  

92     unsigned int ack2=0;  

93     unsigned int ack3=0;  

94  

95     //*****SENSOR CONFIGURATION*****//  

96         SEN_CONFIG();  

97     //*****ENGINE CONFIGURATION*****//  

98         MOT_CONFIG();  

99     //*****I2C CONFIGURATION*****//  

100        Config_I2C1();  

101  

102     while(1)  

103  

104     //*****Récupération d'information*****//  

105  

106     {  

107         angle=0;  

108         Start_I2C1 ();  

109         Write_I2C1 (0xc0);           // Adresse du module //  

110         ack1=IN_ACK_NACK1 ();  

111         if (ack1) LED_BREAK_ON;  

112         Write_I2C1 (1);           // Définition du mode de travail 8 bits //  

113         ack2=IN_ACK_NACK1 ();  

114         if (ack2) LED_BREAK_ON;  

115         Stop_I2C1 ();  

116         Delay1OKTCYx (10);        // Délai nécessaire pour que le stop soit pris en compte //  

117         Start_I2C1 ();  

118         Write_I2C1 (0xcl);        // Adresse de l'information recherchée //  

119         ack3=IN_ACK_NACK1 ();  

120         if (ack3) LED_BREAK_ON;  

121         angle=Read_I2C1 ();       // Lecture de l'information //  

122         OUT_NACK1 ();  

123         mesure=angle;  

124  

125     //*****Phase de test*****//  

126  

127     if(mesure<=85)  

128     {  

129         LED_TOP_GREEN_ON();  

130         Delay1OKTCYx (10);  

131         LED_TOP_GREEN_OFF();  

132         Delay1OKTCYx (10);  

133     }  

134     else  

135     {  

136         if(mesure<=170)  

137         {  

138             LED_FRONT_ON();  

139             Delay1OKTCYx (10);  

140             LED_FRONT_OFF();  

141             Delay1OKTCYx (10);  

142         }  

143         else  

144         {  

145             LED_TOP_RED_ON();  

146             Delay1OKTCYx (10);  

147             LED_TOP_RED_OFF();  

148             Delay1OKTCYx (10);  

149         }  

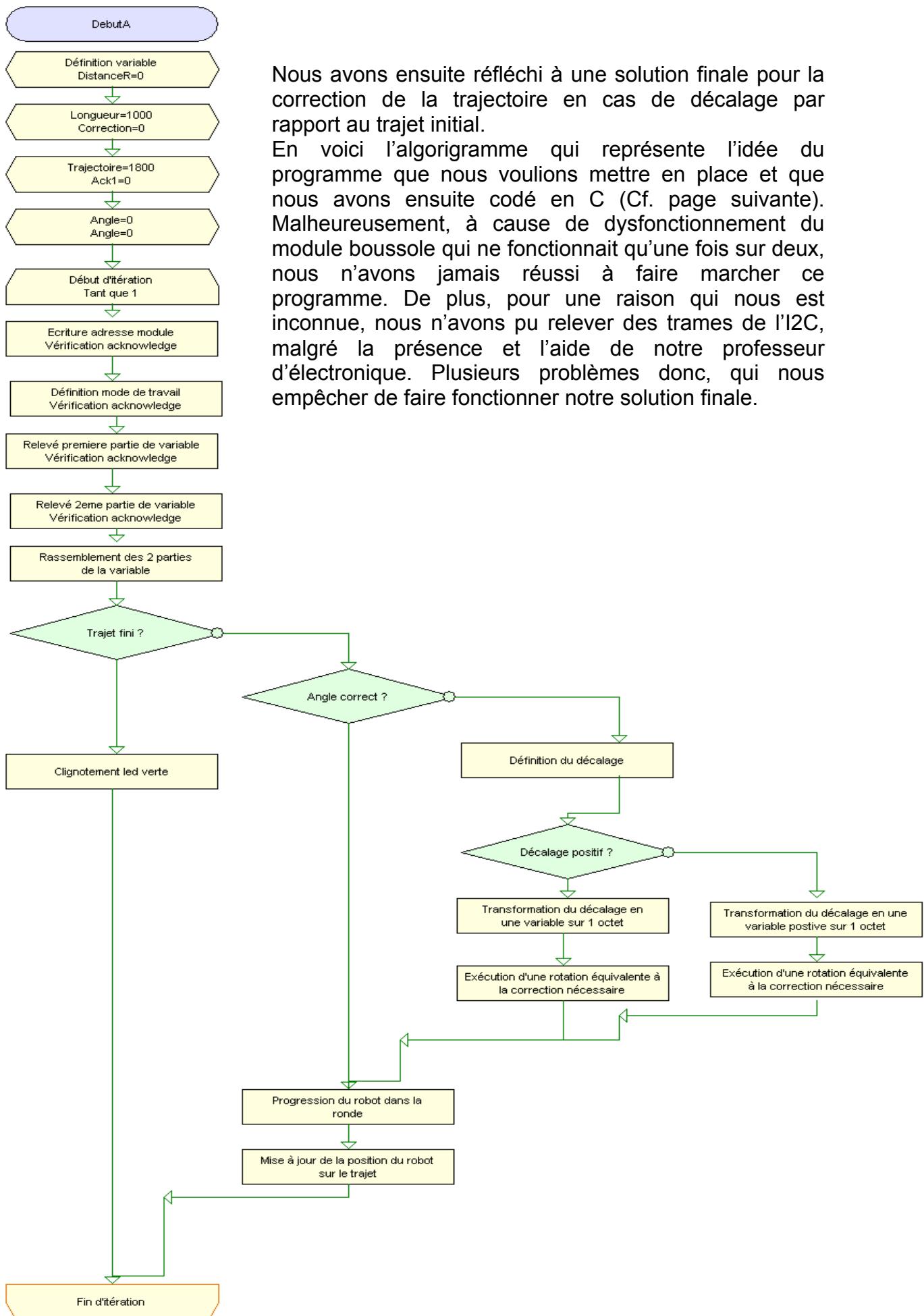
150     }  

151 }  

152 }  

153 }  

154 }
```



```

53 //-----//  

54 //-----//  

55 //-----//  

56 // P R O G R A M M E          P R I N C I P A L      //  

57 //-----//  

58 //-----//  

59 //-----//  

60 void main()  

61 {  

62     // définition variable //  

63     unsigned int distancer=0;  

64     unsigned int longeur=1000;  

65     signed int correction=0;  

66     unsigned int trajetoire=1800;  

67     unsigned int angle=0;  

68     unsigned int angles=0;  

69     unsigned int ack1=0;  

70     unsigned int ack2=0;  

71     unsigned int ack3=0;  

72     unsigned int ack4=0;  

73  

74  

75 //*****SENSOR CONFIGURATION*****//  

76     SEN_CONFIG();  

77 //*****ENGINE CONFIGURATION*****//  

78     MOT_CONFIG();  

79 //*****I2C CONFIGURATION*****//  

80     Config_I2C1();  

81  

82 while(1)  

83 {  

84     angle=0;  

85     Start_I2C1 ();  

86     Write_I2C1 (0xc0);  

87     ack1=IN_ACK_NACK1 ();  

88     if (ack1) LED_BRAKE_ON;  

89     Write_I2C1 (2);  

90     ack2=IN_ACK_NACK1 ();  

91     if (ack2) LED_BRAKE_ON;  

92     Stop_I2C1 ();  

93     Delay10KTCYx (10);  

94     Start_I2C1 ();  

95     Write_I2C1 (0xc1);  

96     ack3=IN_ACK_NACK1 ();  

97     if (ack3) LED_BRAKE_ON;  

98     angle=Read_I2C1 ();  

99     OUT_NACK1 ();  

100  

101     Stop_I2C1 ();  

102     Delay10KTCYx (10);  

103     Start_I2C1 ();  

104     Write_I2C1 (0xc2);  

105     ack4=IN_ACK_NACK1 ();  

106     if (ack4) LED_BRAKE_ON;  

107     angles=Read_I2C1 ();  

108     OUT_NACK1 ();  

109     Stop_I2C1 ();  

110     angle=angle << 8;  

111     angle=angle + angles;  

112     if(distance>distancer)  

113     {  

114         if(angle!=trajetoire)  

115         {  

116             correction=trajetoire-angle;  

117             if(correction<0)  

118             {  

119                 correction=correction-correction-correction;  

120                 correction=correction/10;  

121                 MOT_ROT(100,FWD,CENTER,RIGHT,angle,correction);  

122             }  

123             else  

124             {  

125                 correction=correction/10;  

126                 MOT_ROT(100,FWD,CENTER,LEFT,angle,correction);  

127             }  

128             MOT_STR(100,FWD,DISTANCE, 250);  

129             distancer=distancer+250;  

130         }  

131     }  

132 }

```

### c) Critique et modifications envisagées :

Pour améliorer la correction de la trajectoire, nous pourrions vérifier la position du mOway plus souvent.

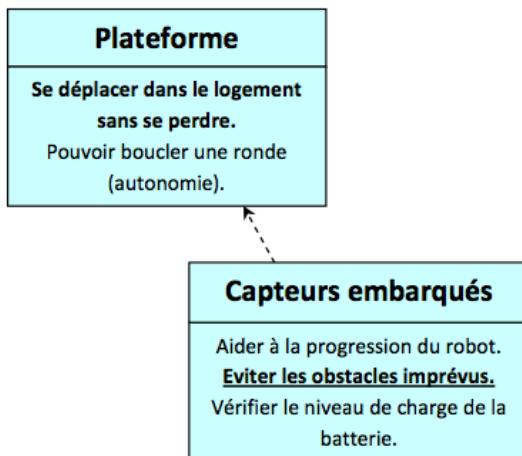
Nous pourrions également vérifier la position du robot avant et après les rotations, permettant ainsi au robot de ne pas partir avec une position de référence mauvaise.

## 2) Autres problématiques :

### a) Détection d'obstacles :

Un obstacle imprévu sur le parcours programmé doit interrompre la marche. Cette situation anormale doit alors déclencher une procédure de tentative de contournement.

Les obstacles à prendre en compte sont ceux qui peuvent perturber AI@Dyn dans sa progression. Contrairement au robot mOway, le robot AI@Dyn devra être équipé de roues plus grandes (pas de bille à l'avant) et d'une garde au sol plus importante. Cet équipement est indispensable pour se déplacer facilement sur un sol souple par exemple (moquettes à poils longs) ou franchir les barres de seuil, les tapis ou les joints de carrelage, etc. Lors de la ronde, AI@Dyn ne doit pas percuter les obstacles qui peuvent se trouver sur son passage. Car cela peut le faire dévier de sa trajectoire ou encore le faire patiner et de ce fait lui faire perdre son parcours.



Pour cela, les élèves chargés de cette problématique ont choisi d'utiliser des capteurs à ultrasons car le problème avec les capteurs infrarouges du mOway, c'est que le noir absorbe la lumière infrarouge, et l'obstacle n'aurait pu être évité.

Voici la solution finale montée sur le mOway :

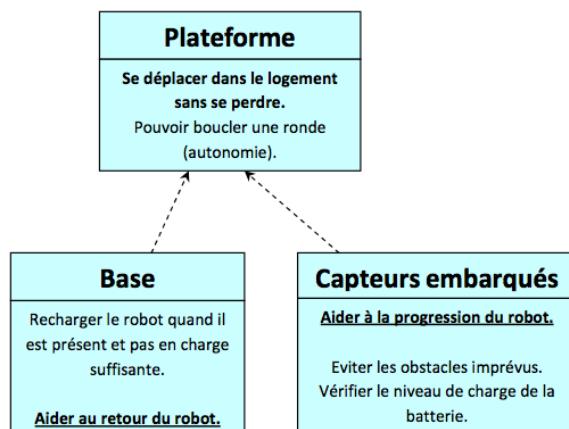


### b) Retour à la base :

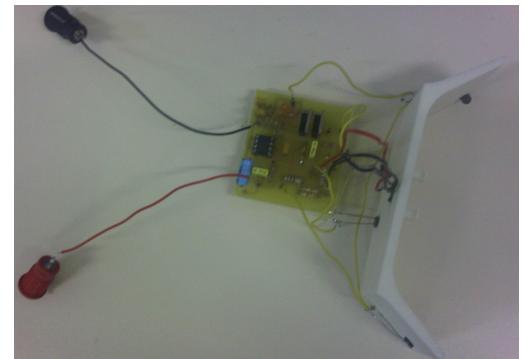
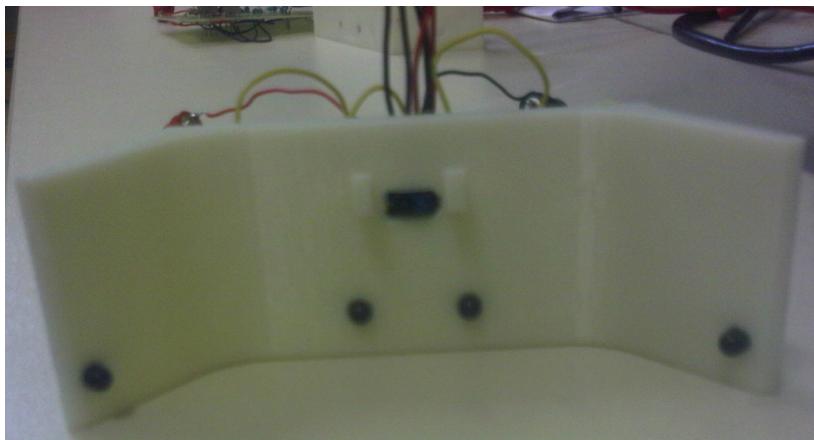
Lors de la ronde, le mOway fait une ou plusieurs haltes en face de l'ouverture ou de l'objet à vérifier puis retourne à sa base. Cette dernière opération peut être automatique, si Al@Dyn est assez près de la base pour la détecter.

La base est un système fixe remplissant plusieurs rôles : elle doit tel un phare guider, Al@Dyn lors de la phase d'approche (fin de la ronde). Elle assure également la recharge de la batterie.

Pour cela, les élèves chargés de cette problématique ont choisi de réaliser une base muni de diodes infrarouges. Le mOway, lorsqu'il détecte les rayons infrarouges émis par la base, se dirige vers celle-ci et y effectue un demi-tour pour venir se positionner près de la solution de recharge.



Voici la base équipée de ses diodes infrarouges :



### c) Rechargement :

Le robot mOway doit être capable d'effectuer des rondes pour procéder à la levée de doutes en l'absence de l'habitant. Lorsqu'il est sur sa base, sa batterie doit donc être maintenue en charge. On opte ici pour une recharge sans contact galvanique.

L'alimentation devra être capable de fournir au moins 100mA au robot mOway. La base sera alimentée par un bloc d'alimentation 15V DC capable de fournir jusqu'à 500mA.

Pour cela, les élèves chargés de cette problématique devaient utiliser une transmission d'énergie par induction entre la base et le robot.

La solution finale devait ressembler à cela :

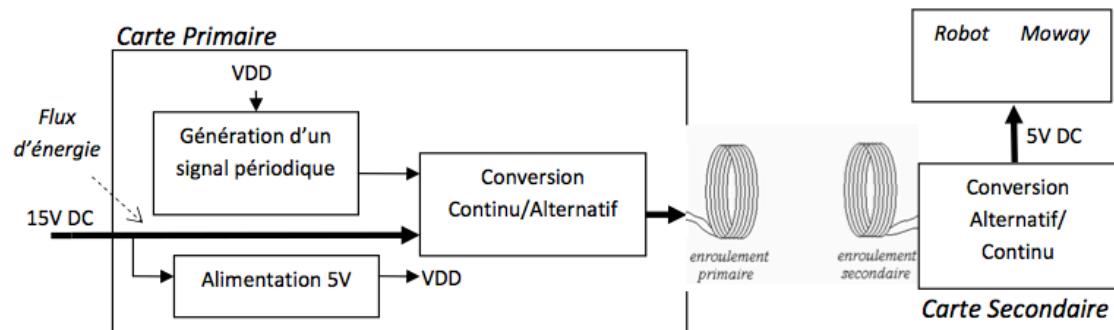


Figure 1 : Représentation fonctionnelle du circuit de charge du robot Moway

Voici la solution finale conçue par les élèves :

