

Ministère de l'Enseignement Supérieur Et de la Recherche Scientifique

Université Djilali BOUNAAMA Khemis Miliana

Faculté des Sciences Et de la Technologie

Département de Génie Électrique



Mémoire du Projet de Fin d'Études
Pour l'obtention du diplôme de

Master

En

« Génie Électrique »

Option :

« Automatique des systèmes de production »

Titre :

Commande d'un robot mobile

Khepera IV

Réaliser Par :

✓ *BOUR Djilali*

✓ *CHERIEF Makhlouf*

Encadré par :

Dr. HOCINE Abdelfettah

Année Universitaire : 2016/2017

Dédicace

*A cœur vaillant rien d'impossible
A conscience tranquille tout est accessible
Quand il y a la soif d'apprendre
Tout vient à point à qui sait attendre*

Je dédie ce modeste travail à mes très chers parents :

À mon père pour l'amour du travail inculqué, ses encouragements et son soutien permanent. Sa présence nous rappelle le sens du devoir.

À ma mère pour sa présence, sa tendresse, son attention, ses conseils et ses encouragements.

*À mon frère **Hicham** et ma sœur **Nour el houda**, les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour vous.*

*À toute ma famille **BOUR** et **HOUARI***

À tous mes cousins et cousines

*À mes professeurs **Hamlaoui**, **benallou**, **benmoumene***

À tous mes amis et collègues

*À mon ami **Cherief Makhlouf** avec qui j'ai partagé ce modeste travail.*

*À tous les étudiants **Automatisme 2017***

À tous ceux qui, par un mot, m'ont donné la force de continuer ...

Djilali

Dédicace

Je dédie ce modeste travail :

À ma très chère mère aicha qui représente pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

À Ma femme Romaisa

À Mon grand-père Ibrahim « que Dieu la garde pour notre famille ».

À Mes frères Ahmed, Abdelouahab, youcef et Mohamed.

À Mes sœurs Karima, Salima, Fouzia, Fatima, Kheira et Nassira.

À Toute la famille cherief

À tous mes oncles et toutes mes tantes.

À Toute la famille Boutalbi.

À tous mes amis notamment, Nabil, Abderazzak, Mohamed, Boumediene, abdelkader, ahmed farrah, kadari, tmtouissin, ben youcef, hadli, abderraouf, etc.

À Mon ami BOUR Djilali avec qui j'ai partagé ce modeste travail.

À Toute la Promotion Automatismes 2017

Makhlouf



Remerciements

Nous remercions tout d'abord le grand Dieu pour l'achèvement de ce mémoire.

Nous exprimons nos gratitude à Monsieur le président de jury d'avoir accepté examiné ce mémoire.

Nous remercions Messieurs les membres de jury, d'avoir accepté de prendre part à ce jury ainsi que pour l'intérêt qu'ils l'ont portés à ce travail.

*Nous remercions Monsieur **HOCINE Abdelfettah**, notre encadreur, pour ses conseils et suggestions avisés qui nous aidés à mener à bien ce travail, et d'avoir rapporté à ce mémoire ces remarques et conseils.*

*Nous remercions conseils Monsieur **MAOUDJ Abderraouf** qui nous aidés à mener à bien ce travail dans le Domain de programmation.*

*Nous Souhaite particulièrement Mme **BOUNIF Aouda** verser remercions aide de sa précieuse à la relecture et à la correction de mémoire.*

ملخص

تمثل هذه المذكرة عرض للتحكم في الآلي Khepera IV. يتكون هذا العمل من جزأين رئيسيين، الجزء الأول يمثل دراسة نظرية شاملة للآلي Khepera IV والاتصال بينه وبين الحاسوب المحمول. ويخصص الجزء الثاني للتحكم في الآلي، أولاً عن طريق كابل USB باستخدام برنامج Ubuntu_LTS_Khepera4 المثبت على linux، ثانياً يتم التحكم به عبر الاتصال اللاسلكي (Bluetooth)، باستخدام putty المثبت على Windows. من ثم تطوير واجهة باستخدام برنامج Visual Studio C#.

الكلمات المفتاحية: Khepera IV، كابل USB، Ubuntu_LTS_Khepera4، linux، Bluetooth، putty، Windows، Visual Studio C#.

Résumé

Ce mémoire de fin d'étude présente une commande d'un robot mobile Khepera IV. Ce travail se compose de deux parties principales, la première partie présente l'étude théorique globale concernant le robot Khepera IV et la communication entre le pc portable et ce dernier.

La deuxième partie est consacrée à la commande du robot, premièrement par un câble USB en utilisant le programme Ubuntu_LTS_Khepera4 sous linux. Deuxièmement, la commande est réalisée par une liaison sans fil par « Bluetooth », via un programme putty sous Windows. Aussi, une interface graphique a été développée en employant Visual Studio C#.

Mots Clés: Khepera IV, câble USB, Ubuntu_LTS_Khepera4, linux, Bluetooth, putty, Windows, Visual Studio C#.

Summary

This final dissertation presents a command of a mobile robot Khepera IV. This work consists of two main parts, the first part represents the overall theoretical study concerning the robot Khepera IV and the communication between the laptops and the latter.

The second part is devoted to the control of the robot, firstly by a USB cable using the program Ubuntu_LTS_Khepera4 under linux. Secondly, the control is carried out by a wireless connection and we mean by this "Bluetooth", via a putty program under Windows. Also, a graphic interface was developed by using Visual Studio C #.

Key words: Khepera IV, USB cable, Ubuntu_LTS_Khepera4, linux, Bluetooth, putty, Windows, Visual Studio C #.

Table des matières

Liste des figures	ix
Liste des abréviations.....	xi
Introduction générale	1
Chapitre I. Généralités sur les robots	3
I.1. Introduction	3
I.2. Notions de base	3
I.2.1. Domaine robotique.....	3
I.2.2. Automate	3
I.2.3. Robot	4
I.2.4. Intelligence artificielle	4
I.3. Historique	4
I.3.1. Automates (1 ^{ère} génération)	4
I.3.2. Robots (2 ^{ème} génération)	5
I.3.3. Robots dotés d'IA (3 ^{ème} génération)	5
I.4. Disciplines de la robotique	5
I.4.1. Mécanique	5
I.4.2. Electronique	6
I.4.3. Informatique	6
I.4.4. Automatique	6
I.4.5. Traitement du signal.....	6
I.4.6. Mathématiques	6
I.4.7. Sciences cognitives.....	6
I.5. Classification des robots	6
I.5.1. Manipulateurs	6
I.5.2. Télémanipulateurs	8
I.5.3. Robots mobiles	8
I.5.3.1. Classification des robots mobiles	8
I.5.3.2. Les moyens de perception en robotique mobile	9
I.5.3.3. La structure mécanique et motricité des robots mobiles.....	9
I.5.3.4. Traitement des informations et gestion des taches.....	10
I.5.3.5. Les avantages dans l'utilisation des robots mobiles	10
I.6. Caractéristiques d'un robot	11
I.7. Domaines d'applications	11
I.7.1. Domaine industriel	11
I.7.2. Domaine spacial	12
I.7.3. Domaine agriculture.....	13

I.7.4. Domaine de service	13
I.7.5. Domaine militaire	14
I.7.6. Domaine médical	15
I.8. Systèmes multi robots	16
I.8.1. Définition	16
I.8.2. Coopératifs vs compétitifs	16
I.8.3. Problèmes inhérents	17
I.8.4. Coordination : Statique vs dynamique.....	17
I.8.5. Communication	17
I.8.6. Planification	18
I.8.6.1. Planification de mouvements	18
I.8.6.2. Planification de tâches	18
I.9. Conclusion.....	19
Chapitre II. Description technique du robot mobile Khepera IV	20
II.1. Introduction.....	20
II.2. Historique sur les robots Khepera.....	20
II.3. Généralité sur le robot Khepera IV	22
II.3.1. Définition	22
II.3.2. Description physique	22
II.4. Fonctionnalités principales du robot mobile Khepera IV	23
II.5. Caractéristiques techniques du robot Khepera IV	23
II.6. Les différents composants du robot mobile Khepera IV	25
II.6.1. Vue globale du robot	25
II.6.2. Vue détaillé des composantes du robot	28
II.6.2.1. Les roues pivotantes	28
II.6.2.2. Bouton marche-arrêt	28
II.6.2.3. LED d'état	28
II.6.2.4. Connecteur Mini-USB B (mode appareil, sans charge)	28
II.6.2.5. Connecteur USB A (mode hôte, 500 mA)	28
II.6.2.6. Prise d'alimentation (9V, 1.5A, 0.65 mm centre positif)	28
II.6.2.7. Voyant d'état de charge	28
II.6.2.8. Bouton de réinitialisation	29
II.6.2.9. Capteurs infrarouges	29
II.6.2.9.1. Mesure de la lumière ambiante	30
II.6.2.9.2. Mesures de la lumière réfléchie (proximité)	30
II.6.2.10. Capteurs à ultrasons	31
II.6.2.11. Caméra	31
II.6.2.12. Capteurs à infrarouges de fond (bas)	32
II.6.2.13. Contacts pour station d'accueil	32
II.6.2.14. Roues	32

II.6.2.15. Autocollant.....	32
II.6.2.16. Bas écrous M3.....	32
II.6.2.17. Connecteurs d'extension 250 Ko	32
II.6.2.18. Haut écrous M3	32
II.6.2.19. Aimants	33
II.6.2.20. LED RVB.....	33
II.6.3. En profondeur	33
II.6.3.1. Microphones.....	33
II.6.3.2. Haut-parleur	34
II.6.3.3. Accéléromètre	34
II.6.3.4. Gyroscope	35
II.6.3.5. Périphérique USB (mini-USB B connecteur)	35
II.6.3.6. Carte microSD	35
II.6.3.7. Moteurs	35
II.6.3.8. Batterie	36
II.7. Moyenne d'extension	37
II.7.1. La tourelle d'extension	37
II.7.2. Télémètre laser	38
II.7.3. Le préhenseur	38
II.7.4. Système de positionnement global intérieur (Stargazer)	39
II.8. Conclusion	40
Chapitre III. Conception et Implémentation	41
III.1. Introduction	41
III.2. Précautions d'utilisation	41
III.3. Commande du robot mobile et communication via linux	41
III.3.1. Création d'une machine virtuelle avec VMware Workstation	41
III.3.2. Commande du robot via linux.....	43
III.3.2.1. Signification des commandes	46
III.3.2.2. Arrêt du robot	47
III.3.3. Arrêt de la machine virtuelle	48
III.4. Commande du robot mobile et communication via Windows	48
III.4.1. Commande du robot par un programme putty	48
III.5. Application de pilotage du robot	50
III.5.1. La découverte de la programmation orientée objet en C#.....	50
III.5.2. Pilotage et supervision du robot	53
III.5.2.1. Description de l'interface IHMI	54
III.5.2.2. La configuration du port série en C#	55
III.6. La commande du robot sans fil (Bluetooth)	55
III.6.1. Utilisation d'une clé Bluetooth	55
III.6.2. Installation du programme Blue Soleil	56

III.7. Conclusion	57
Conclusion générale	58
Bibliographie	59
Annexes	62

Liste des figures

Figure I.1 : Automate d'Hugo Cabret	4
Figure I.2 : Constitution d'un robot manipulateur	7
Figure I.3 : Parties constituant un RM	8
Figure I.4 : Robots soudeurs et manipulateurs en action	12
Figure I.5 : Robot Rover Martien	13
Figure I.6 : Robot utilisé en agricole	13
Figure I.7 : Robot de service ASIMO	14
Figure I.8 : Robot utilisé dans le domaine militaire	15
Figure I.9 : Endoscopie (Innovations en chirurgie cardiaque)	15
Figure II.1 : Le premier robot Khepera publié en 1996.....	20
Figure II.2 : Khepera II	21
Figure II.3 : Khepera III	21
Figure II.4 : Khepera IV	21
Figure II.5 : Vue du bas, du haut, de l'avant et de la gauche du robot	23
Figure II.6 : Vue de gauche	25
Figure II.7 : Vue arrière	25
Figure II.8 : Vue de droite	26
Figure II.9 : Vue de dessous	26
Figure II.10 : Vue de dessus	27
Figure II.11 : Capteurs à infrarouges vus de fond	29
Figure II.12 : Valeur IR par rapport à la distance	30
Figure II.13 : Capteurs ultrasoniques vus de sommet	31
Figure II.14 : Positions de microphones	33
Figure II.15 : Les directions d'accélération détectables	34
Figure II.16 : Les directions de taux angulaires détectables	35
Figure II.17 : Bloc du moteur avec roue	36
Figure II.18 : La tourelle d'extension	37
Figure II.19 : Extension télémètre laser	38
Figure II.20 : Extension avec le préhenseur	39
Figure II.21 : Le Stargazer	39

Figure III.1 : Interface de la machine virtuelle	42
Figure III.2 : Champs de la machine virtuelle : Nom d'utilisateur et mot de passe	42
Figure III.3 : Emplacement des fichiers de la librairie	43
Figure III.4 : Installation du port minicom	44
Figure III.5 : Paramètres de configuration du port minicom.....	44
Figure III.6 : Configuration des paramètres du port minicom	45
Figure III.7 : L'établissement de la connections entre le robot et linux	45
Figure III.8 : Liste des commandes du robot sous linux	46
Figure III.9 : Arrêt de la machine virtuelle	48
Figure III.10 : Configuration des paramètres du port COM	49
Figure III.11 : L'établissement de la connections entre le robot et Windows	49
Figure III.12 : La liste des commandes du robot sous putty	50
Figure III.13 : Les différentes zones du Visual Studio	51
Figure III.14 : La liste des évènements.....	52
Figure III.15 : Les différentes parties de l'interface de pilotage.....	54
Figure III.16 : Adaptateur Bluetooth	55
Figure III.17 : La détection du robot Khepera IV par le programme Blue Soleil	56

Liste des abréviations

SMR	Système Multi-Robot
IA	Intelligence Artificielle
OT	Organe Terminal (outil)
SMA	Système Mécanique Articulé
RM	Robot Manipulateur
CAO	Conception Assistée par Ordinateur
I2C	Inter Integrated Circuit
IMU	Unité de Mesure Inertielle
CC	Courant Continue
IR	Infra-Rouge
LRF	Télémètre laser (Laser Range Finder)
FPS	Cadre Par Seconde (Framework Per Second)

Introduction Générale

Introduction générale

Depuis les années 70, la robotique est devenue une science extrêmement populaire dans les milieux universitaire. Alliant un grand intérêt pédagogique et industriel, cette nouvelle science demande beaucoup de créativité et des connaissances pluridisciplinaires (Mécanique, électronique, automatique, informatique, mathématique, etc.).

Ainsi, l'utilisation des robots est aujourd'hui couramment envisagée pour l'automatisation de nombreuses tâches. Celles-ci sont particulièrement diversifiées tels que : le nettoyage, l'agriculture, l'assistance aux personnes handicapées, etc.

Dans ce projet de fin d'étude, nous avons choisi la commande d'un robot mobile Khepera IV, qui est disponible au niveau du laboratoire du département de génie électrique à l'université de Djilali BOUNAAMA Khemis Miliana, et qui n'a pas fonctionné depuis son acquisition par l'université. Ce robot est la 4^{ème} génération du robot mobile Khepera pour l'éducation et la recherche, il apporte de nombreuses nouveautés : OS linux, Wi-Fi, caméra, gyroscope, etc. Il a la forme d'un cylindre, avec un diamètre de 14,08 cm et une hauteur du sol de 5,77 cm (y compris les roues) et pèse 566 g.

Les points qui seront traités dans ce mémoire concernent principalement les parties théoriques du robot mobile khepera IV (Mécaniques, capteurs, etc.), celles du microcontrôleur **dsPIC33FJ64** et les moyens d'extension du robot (Les préhenseurs, télémètres laser, etc.).

Le robot sera commandé sans fil, par un programme **Ubuntu_LTS_Khepera4** sous linux et un autre programme **putty** sous Windows via un câble USB.

Nous allons créer une interface graphique sur le micro-ordinateur qui sera implémentée sous Windows en utilisant **Visual Studio C#**. Cette dernière permettra de communiquer avec le robot et envoyer les consignes de l'utilisateur.

Ce travail comporte trois chapitres : Le premier chapitre représentera des généralités sur les robots, leur domaine d'utilisation et leurs différents types tout en détaillant les robots mobiles.

Dans le deuxième chapitre, nous donnerons une description technique du robot mobile Khepera IV, son historique et ses moyens d'extension.

La première partie du dernier chapitre sera consacrée à la communication avec le robot par l'utilisation d'un programme **Ubuntu_LTS_Khepera4** sous linux, alors que la deuxième partie va décrire le programme **putty** sous Windows qui permettra de commander le robot et créer une interface d'utilisateur pour le pilotage du robot sans fil (Communication avec Bluetooth), en utilisant **Visual Studio C#** et une clé de Bluetooth avec une portée de 100 m.

En fin, nous terminerons notre travail par une conclusion générale.

Chapitre I

Chapitre I : Généralités sur les robots

I.1. Introduction

L'utilisation des robots est aujourd'hui couramment envisagée pour l'automatisation de nombreuses tâches. Celles-ci sont particulièrement diversifiées : le nettoyage, le transport dans les ateliers automatisés, l'agriculture, l'exploitation des mines, l'assistance aux personnes handicapées et l'exploration de milieux hostiles en sont quelques exemples.

On va entamer ce chapitre par un aperçu historique sur les robots, leurs caractéristiques, et leurs domaines d'application, ainsi que les robots mobile.

I.2. Notions de base

Dans cette partie nous allons introduire quelque notion concernons notre sujet, on peut citer les suivants :

I.2.1. Domaine robotique

Ensemble des études et des techniques de conception et de mise en œuvre des robots effectuant des tâches déterminées en s'adaptant à leur environnement [1]. La robotique vient de l'anglais robotics, imaginé par le romancier « Isaac Asimov » et popularisé par un livre publié en 1942, (**Runaround**) [2].

I.2.2. Automate

Machine qui par le moyen de dispositifs mécaniques, pneumatiques, hydrauliques, électriques ou électroniques, est capable d'actes imitant ceux des corps animés [3]. Structure physique ou informationnelle qui fonctionne d'après des règles strictes, heuristiques ou probabilistes, sans l'intervention consciente de l'être humain [4]. La figure I.1 illustre un exemple d'automate.



Figure I.1 : Automate d'Hugo Cabret [5]

I.2.3. Robot

Un robot est un automate doté de capteurs et d'effecteurs lui donnant une capacité d'adaptation et d'emplacement proche de l'autonomie. Un robot est un agent physique réalisant des tâches dans l'environnement dans lequel il évolue [6].

I.2.4. Intelligence artificielle

Discipline scientifique relative au traitement des connaissances et au raisonnement, dans le but de permettre à une machine d'exécuter des fonctions normalement associées à l'imitation de la sortie humaine : Compréhension, raisonnement, dialogue, adaptation, apprentissage, etc. [7]

I.3. Historique [8]

On peut schématiquement distinguer trois principales ères en robotique : les automates, les robots n'étant pas dotés d'intelligence artificielle (IA), et ceux disposant d'une IA.

I.3.1. Automates (1^{ère} génération)

Un automate contrairement à un robot (même s'il ne dispose pas d'IA), obéit uniquement à un programme préétabli, que ce soit de manière mécanique ou électronique. De ce fait, il n'y a aucune adaptation possible entre l'automate et son environnement.

On attribue la paternité du tout premier automate de l'humanité à Architos de Tarente (4^{ème} siècle avant J.C). Il s'agissait d'une représentation d'un pigeon capable de voler et étant propulsé par de la vapeur. Malheureusement, aucun vestige ni aucun schéma ou représentation fidèle n'ont été retrouvés.

Les toutes premières traces d'automates remontent à l'antiquité par Héron d'Alexandrie au 1er siècle après J.C., ses réalisations ornèrent les temples et les théâtres de la ville égyptienne.

En 1495 Léonard de Vinci présenta un chevalier humanoïde capable de s'asseoir, de relever sa visière et de bouger ses bras.

L'automate le plus célèbre est le canard mécanique de Jacques De Vaucanson, capable d'ingurgiter de la nourriture et de la digérer tout en se déplaçant. De Vaucanson aurait également présenté en 1738 un second automate représentant un homme jouant d'un instrument à vent.

I.3.2. Robots (2^{ème} génération)

Les robots de 2^{ème} génération disposent d'organe(s) sensoriel(s), autrement dit des capteurs, pouvant influencer sur leurs comportements. Ils sont donc relativement adaptables à leur environnement.

Le chien électrique de Hammond et Miessner (1915) est le premier robot de ce genre. Il se déplace selon la luminosité de l'endroit grâce à son capteur optique.

Le chien Phillipdog de Henri Piraux en 1928, et le renard de Ducrocq (1953) fonctionnent selon le même principe.

Walter Grey équipe en 1950 sa tortue cybernétique de capteurs tactiles et lumineux.

I.3.3. Robots dotés d'IA (3^{ème} génération)

Pour être intégré dans cette famille, un robot doit parvenir à effectuer une tâche par lui-même, sans aucune aide extérieure.

En 1973, l'université de Waseda présente le tout premier humanoïde "intelligent" dénommé Wabot-1. Il est doué de la vision, peut manipuler des objets, effectue un semblant de marche et est même capable de débiter une conversation en japonais.

Hi-T-Hand d'Hitachi en 1974 manipule des aiguilles à travers des trous grâce à la détection de force.

Vers la fin des années 70 Hans Morava présente les premiers robots capables d'évoluer à l'extérieur, de façon autonome.

I.4. Disciplines de la robotique

I.4.1. Mécanique

Conception, réalisation, modélisation des robots.

I.4.2. Electronique

Mise en place de composants pour les robots, télécommunications, communications, commande, acquisition des données.

I.4.3. Informatique

Gestion des fonctionnalités des robots et de transfert d'information.

I.4.4. Automatique

Commande, calibrage des capteurs, des effecteurs, identification des paramètres.

I.4.5. Traitement du signal

Analyse des informations enregistrées par les capteurs du robot.

I.4.6. Mathématiques

Modèles mathématiques pour la prise de décision ou/et l'apprentissage, calcul de trajectoires, localisation, planification.

I.4.7. Sciences cognitives

Interactions homme-machine, machine-machine, prise de décision.

I.5. Classification des robots

On peut les classés en trois catégories :

I.5.1. Manipulateurs [9]

Ce sont des robots industriels destinés à effectuer des tâches bien déterminées et de manière répétitive. A noter que pour ce type de robot :

- Les trajectoires sont bien déterminées dans l'espace ;
- Les positions sont discrètes avec 2 ou 3 valeurs par axe ;
- La commande est séquentielle.

La figure suivante illustre les différents éléments qui constituent un robot manipulateur :

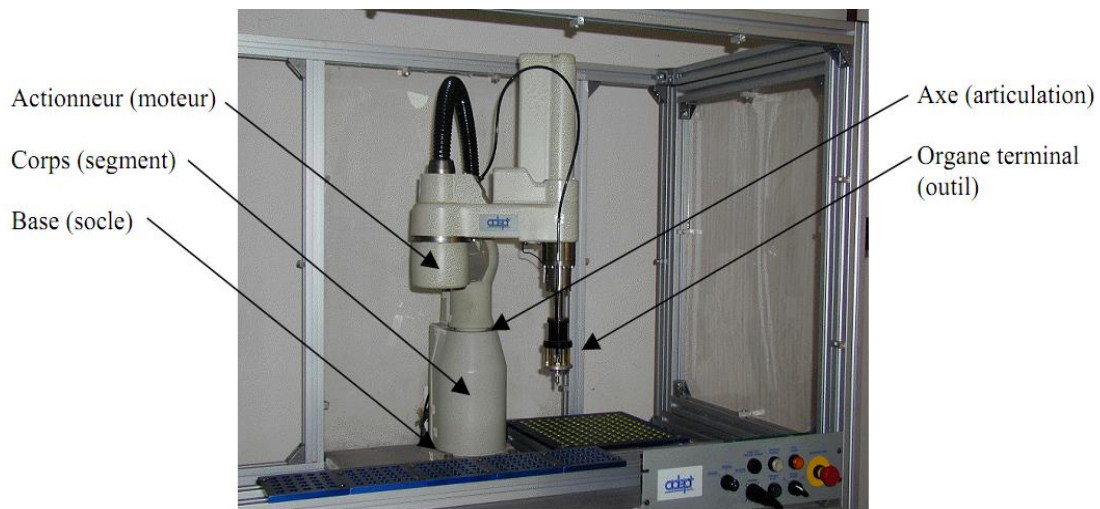


Figure I.2 : Constitution d'un robot manipulateur

Sous le terme organe terminal (OT), on regroupe tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, etc.), ou à les transformer (outils, torche de soudage, pistolet de peinture, etc.). En d'autres termes, il s'agit d'une interface permettant au robot d'interagir avec son environnement.

Un OT peut être multifonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable. Un robot, enfin, peut-être multi-bras, chacun des bras portant un OT différent.

Le système mécanique articulé (SMA) est un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Il permet de remplacer, ou de prolonger, son action. Son rôle est d'amener OT dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données.

Précisons la notion d'articulation : Une articulation lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre.

Ces différents éléments du robot manipulateur (RM) peuvent être regroupés en quatre parties principales comme indiqué sur la (Figure I.3).

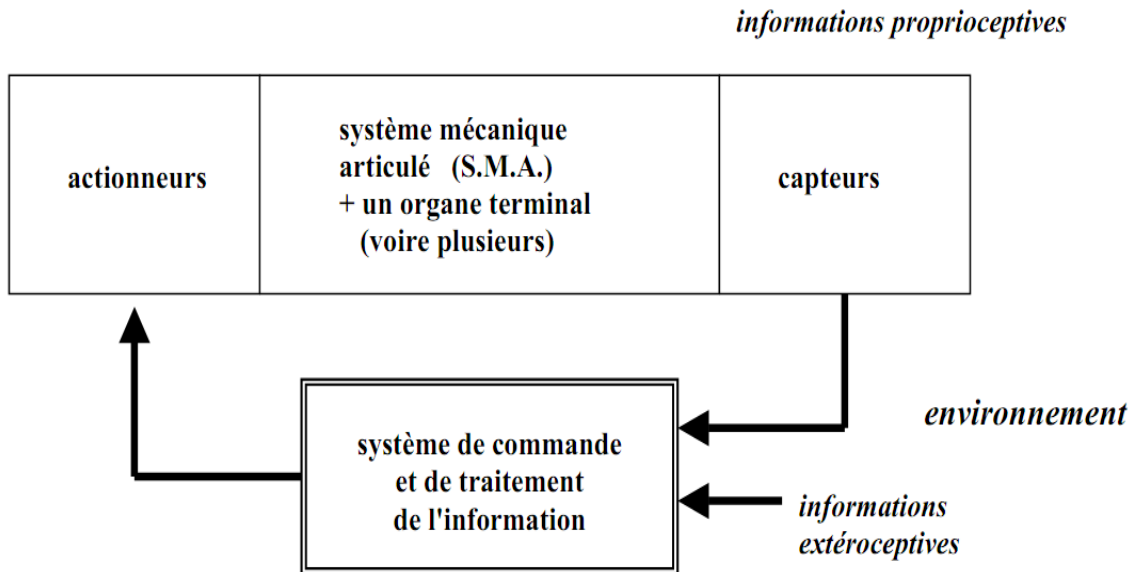


Figure I.3 : Parties constituant un RM

I.5.2. Télémanipulateurs [10]

Appareils de manipulation à distance (pelle mécanique, pont roulant, etc.), apparus vers 1945 aux USA. Dans ce cas :

- Les trajectoires peuvent être quelconques dans l'espace ;
- Les trajectoires sont définies de manière instantanée par l'opérateur.

I.5.3. Robots mobiles

D'une manière générale, on regroupe sous l'appellation robots mobiles l'ensemble des robots à base mobile, par opposition notamment aux robots fixes. L'usage veut néanmoins que l'on désigne le plus souvent par ce terme les robots mobiles à roues.

Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens. L'architecture mécanique d'un robot mobile dépend très largement des spécificités de la mission à effectuer et de l'environnement de travail. Ces données conditionnent entre autres le choix d'un système de locomotion approprié.

I.5.3.1. Classification des robots mobiles

La classification des robots mobiles se fait suivant plusieurs critères (degré d'autonomie, système de locomotion, énergie utilisée, etc.). La classification la plus intéressante, et la plus utilisée est selon leur degré d'autonomie. Un robot mobile autonome est un système automoteur doté de capacités décisionnelles et de moyens d'acquisition et de traitement de l'information qui lui permettent d'accomplir sous contrôle humain réduit,

un certain nombre de tâches, dans un environnement non complètement connu. On peut citer quelques types :

- Véhicule télécommandé par un opérateur qui lui impose chaque tâche élémentaire à réaliser ;
- Véhicule télécommandé au sens de la tâche à réaliser. Le véhicule contrôle automatiquement ses actions ;
- Véhicule semi-autonome réalisant sans l'aide de l'opérateur des tâches prédéfinies ;
- Véhicule autonome qui réalise des tâches semi- définies.

Les principaux problèmes particuliers liés à la conception de tels robots sont :

1. La conception mécanique liée à la mobilité ;
2. La détermination de la position et de l'orientation ;
3. La détermination du chemin optimal.

I.5.3.2. Les moyens de perception en robotique mobile

La perception est un domaine crucial de la robotique. C'est autour de ce concept qui est bâtie la structure d'un robot apte à exécuter des tâches complexes ou à évoluer dans un univers inconnu ou mal connu. L'élément de base du système de perception est le capteur qui a pour objet de traduire en une information exploitable des données représentant des caractéristiques de l'environnement. Les moyens utilisés pour la perception de l'environnement sont nombreux. Nous citons :

- Les systèmes de vision globale ;
- Les télémètres laser et ultrasonores ;
- Les capteurs optiques et infrarouges ;
- Les capteurs tactiles.

I.5.3.3. La structure mécanique et motricité des robots mobiles

Il existe quatre types de structures mécaniques assurant la motricité :

1. Les robots à roues ;
2. Les robots à chenilles ;
3. Les robots marcheurs ;
4. Les robots rampants.

• Les robots mobiles à roues

La mobilité par roues est la structure mécanique la plus communément appliquée. Cette technique assure selon l'agencement et les dimensions des roues un déplacement dans

toutes les directions avec une accélération et une vitesse importantes. Le franchissement d'obstacles ou l'escalade de marches d'escaliers est possible.

- **Les robots à chenilles**

L'utilisation des chenilles présente l'avantage d'une bonne adhérence au sol et d'une faculté de franchissement d'obstacles. L'utilisation est orientée vers l'emploi sur sol accidenté ou de mauvaise qualité au niveau de l'adhérence.

- **Les robots marcheurs**

Les robots marcheurs sont destinés à réaliser des tâches variées dont l'accès au site est difficile. Leur anatomie à nombreux degrés de liberté permet un rapprochement avec les robots manipulateurs. La locomotion est commandée en termes de coordonnées articulaires.

Les différentes techniques étudiées se rapprochent de la marche des animaux et notamment de celle des insectes. Destinés à réaliser des tâches variées dont l'accès au site est difficile. Leur anatomie à nombreux degrés de liberté permet un rapprochement avec RMs. La locomotion est commandée en termes de coordonnées articulaires. Les différentes techniques étudiées se rapprochent de la marche des animaux et notamment de celle des insectes.

I.5.3.4. Traitement des informations et gestion des tâches

L'ensemble de traitement des informations et gestion des tâches constitue le noyau du module d'informatique central qui établit les commandes permettant au robot mobile de réaliser un déplacement et d'activer les divers organes en accord avec l'objectif. A ce niveau, le problème qui se pose est le problème de génération de plan qui consiste à établir la manière dont le robot se déplace par rapport à des connaissances aprioriques «Statiques» ou obtenues en cours d'évolution « dynamiques ». La génération des plans repose sur trois concepts :

1. La stratégie de navigation ;
2. La modélisation de l'espace ;
3. La planification.

I.5.3.5. Les avantages dans l'utilisation des robots mobiles

Les divers avantages des robots mobiles se résument ainsi :

- Accroissement de la capacité de production ;
- Remplacement de l'homme dans l'exécution des tâches pénibles ou dangereuses ;
- Manutentions.

I.6. Caractéristiques d'un robot

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Voici quelques paramètres à prendre en compte [11] :

- La charge maximale transportable (de quelques kilos à quelques tonnes), à déterminer dans les conditions les plus défavorables ;
- L'architecture du système mécanique articulé, le choix est guidé par la tâche à réaliser ;
- Le volume de travail, défini comme l'ensemble des points atteignables par OT. Tous les mouvements ne sont pas possibles en tout point du volume de travail ;
- Le type du robot et la mission à effectuer ;
- La vitesse de déplacement ;
- La masse du robot ;
- Le coût du robot.

I.7. Domaines d'applications [12]

La robotique est un domaine en plein essor depuis quelques années, utilisée dans des domaines extrêmement rigoureux et exigeants. Nous allons explorer ces différents domaines.

I.7.1. Domaine industriel

Depuis leur apparition sur les chaînes de production dans les années 1970, les robots industriels ont pris une place de plus en plus importante dans la plupart des secteurs où les opérations d'assemblage et les manipulations de produits sont récurrentes. Durant les trois dernières décennies, les progrès technologiques ont participé à rendre les robots plus flexibles, plus rapides et plus précis.

Aujourd'hui, ils jouent un rôle prépondérant pour :

- les manipulations de produits à haute cadence ;
- les opérations de packaging avec des manipulations de produits (manipulations primaires) et d'emballage (manipulations secondaires) ;
- l'assemblage de produits (secteurs automobile, pharmaceutique, cosmétique, etc.) ci-dessous (figure I.9) ;
- les opérations de fin de ligne (palettisation).

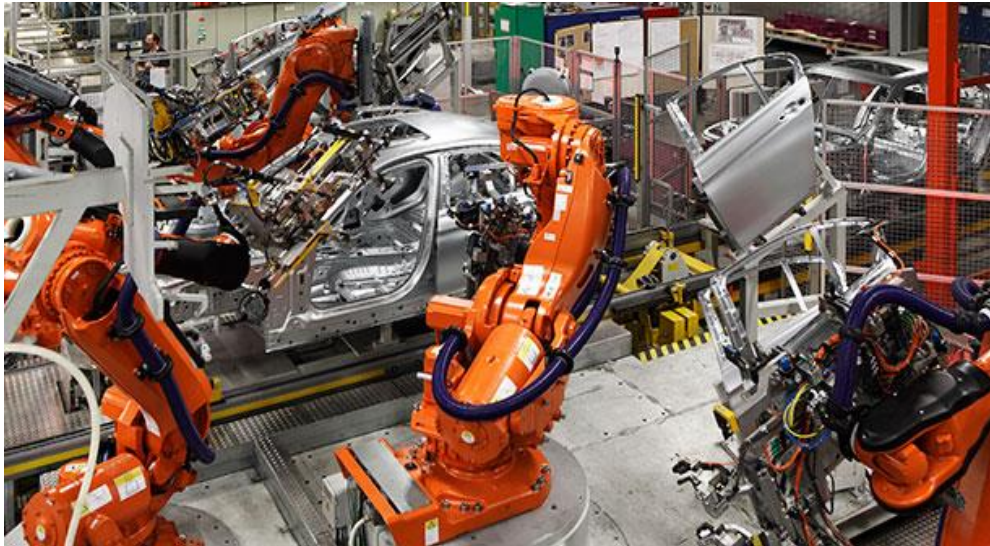


Figure I.4 : Robots soudeurs et manipulateurs en action

D'après les domaines d'applications vus précédemment on constate bien qu'un robot puisse avoir un rendement élevé, et des missions bénéfiques fournis pour son environnement par l'évolution des technologies robotiques. Mais certaines tâches sont complexes, voire impossibles pour être effectuées comme des tâches séparées dans l'espace comme le cas d'une zone industrielle robotisée qui planifie ses travaux et renforce sa production à l'aide d'une flotte de robots ou bien dit un SMR qui sera présenté par la suite. Donc en robotique le fait d'utiliser une équipe de robots de leur inculquer certains concepts sociaux et non seulement d'utiliser un seul robot individuel peut avoir un impact majeur sur les capacités d'offres par les systèmes robotisés.

I.7.2. Domaine spatial

Les manipulateurs mobiles spatiaux sont destinés à explorer des environnements où l'homme ne peut pas se rendre. C'est-à-dire des environnements souvent mortels pour l'homme.

Aujourd'hui, l'histoire de la conquête spatiale est devenue indissociable de celle de la robotique et à ce moment même plusieurs robots sont en activité, aussi bien sur la station spatiale internationale que sur la planète Mars. Les futures missions d'exploration feront elles aussi appel aux robots et à leur IA (figure I.4), que ce soit pour relever de nouveaux défis ou pour mieux connaître notre système solaire. Mais ces machines s'avèreront être de plus en plus variées puisque toutes les tailles, toutes les fonctions et tous les modes de déplacements sont à l'étude afin de développer des robots capables d'assurer des tâches très distinctes, de façon autonome ou en parfaite synergie avec les humains.



Figure I.5 : Robot Rover Martien

I.7.3. Domaine agriculture

Après des décennies d'expérimentation et de tâtonnements, les robots ont enfin fait leur entrée à la ferme. Cette machine totalement autonome fonctionne grâce à l'énergie solaire et circule dans les rangées de plantations pour surveiller et analyser les plants (voir figure I.5). Ce robot a déjà passé avec succès de nombreux tests réalisés dans des champs de légumes mais se contente de surveiller la « bonne santé » et plantations. Grâce à ces nombreux capteurs, senseurs et caméras, il détecte rapidement d'éventuelles anomalies (présence de mauvaises herbes, animaux nuisibles, croissance trop faible) et avertit l'exploitant agricole qui peut ainsi prendre immédiatement les mesures appropriées.



Figure I.6 : Robot utilisé en agricole

I.7.4. Domaine de service

La révolution de la robotique a conduit ces dernières années à voir de nombreux robots s'installer chez les particuliers pour effectuer des tâches à la place de leur possesseur,

la figure I.6 illustre le robot de service ASIMO. En effet, ceux-ci sont capables de faire le ménage, tondre la pelouse, nettoyer la piscine etc. Ce qui conduit certains clients (aisés) à se procurer ces domestiques contemporains.

Enfin, la robotique autrefois réservée à des applications précises ou coûteuses, est aujourd'hui de plus en plus utilisée à titre ludique. En effet, les robots compagnons par exemple sont des objets de plus en plus convoités : les applications « basiques » de jouet pour enfant, jusqu'à l'humanoïde destiné à remplacer une présence humaine.



Figure I.7 : Robot de service ASIMO

I.7.5. Domaine militaire

Les robots sont de plus en plus utilisés dans le domaine militaire. En effet, la miniaturisation permet aujourd'hui de créer des robots discrets mais dotés de nombreux capteurs, ce qui est idéal pour des missions d'espionnage ou d'éclairage, comme le montre la figure I.7 suivante.

De plus, certains robots sont équipés d'un armement pour évoluer en milieu hostile, dans le but de remplacer les soldats pour limiter les pertes humaines.



Figure I.8 : Robot utilisé dans le domaine militaire

I.7.6. Domaine médical

Les robots commencent à être de plus en plus dans le domaine médical, qu'il s'agisse de « simples » échographies ou d'opérations chirurgicales plus délicates. En fait ces robots ne sont pas complètement autonomes mais ils assistent les médecins ou chirurgiens, jusqu'à permettre des opérations médicales à distance (télémédecine). On parle de chirurgie assistée (mot né de l'anglais « surgery » : chirurgie) c'est-à-dire tout ce qui consiste à introduire les derniers outils des technologies informatiques et robotiques dans la pratique médico-chirurgicale.

Cette pratique de « chirurgie assistée » est émergente donc bien que peu répandue, elle est en phase de devenir la chirurgie du futur. La figure I.8, montre une opération à l'aide d'un robot.



Figure I.9 : Endoscopie (Innovations en chirurgie cardiaque)

I.8. Systèmes multi robots

La notion des systèmes multi-agent débute dans les années 1990, notamment dans des travaux regroupant des robots mobiles rassembleurs d'objets [13], des colonies des robots marcheurs [14, 15].

Dans cette section on va faire un tour d'horizon sur ce type de système, en présentant dans la suite quelques notions liées à ce dernier.

I.8.1. Définition

C'est une flotte des robots autonomes constituée de plusieurs robots capables de partager des données et d'effectuer ensemble une ou plusieurs tâches. Ces systèmes sont de plus en plus envisagés en connexion et en coopération avec d'autres objets/capteurs fixes et mobiles dans l'environnement. [16]

I.8.2. Coopératifs vs compétitifs

La notion de comportement collectif dans les SMRs se définit par similitude avec le principe de travail collectif dans notre société. Popenoe [17] a défini un comportement collectif comme un comportement dans lequel il apparaît une réponse à une influence ou à un stimulus commun dans une situation spontanée, imprédictible, non-structurée et instable. Le comportement collectif comprend deux sous-catégories de comportements associées aux deux notions de coopération (appelé comportement coopératif) et de compétition (appelé comportement compétitif). Les environnements multi-robots sont ainsi soient coopératifs, soient compétitifs [18].

Une première définition présente la coopération par l'identification de situations dans lesquelles il apparaît un besoin d'interaction entre plusieurs robots. Cette coopération devient utile quand le résultat de l'exécution de la tâche est estimé meilleur avec plusieurs robots. En effet, la coopération est l'interaction entre des robots qui travaillent dans un intérêt commun [19].

La notion de comportements compétitifs présente les situations et les problèmes à résoudre sous la condition que les robots soient en compétition les uns contre les autres pour satisfaire leur propre intérêt. Chaque robot possède dans ce cas, un ensemble d'intérêts partiellement ou totalement en concurrence avec les intérêts d'autres robots. Cette notion d'intérêt est définie par des fonctions d'utilité en accord ou en contradiction avec les autres robots. [18]

I.8.3. Problèmes inhérents

Ce type de système ou bien dit une flotte de robots, peut engendrer plusieurs problèmes au niveau de l'environnement du travail commun partagé par les différents robots en coopération et en compétition cités ci-dessous :

- Si plusieurs demandes visant la même ressource apparaissent simultanément, il se produit un conflit de ressources. Ce problème a été étudié sous de nombreuses formes : un cas bien connu de problème de ce type est le problème de l'exclusion mutuelle [20] ;
- Les problèmes de partage de média de communication sont souvent associés à des limitations de bande passante [21] ;
- La question de l'identification des sous-buts et de leur répartition dans un SMR ;
- Le problème du partage de l'espace a été étudié principalement dans les problèmes de planification de mouvements multi-robots et dans les problèmes d'évitement de collision, de congestion et d'inter blocage qui vont cerner par la suite la problématique de ce travail.

I.8.4. Coordination : Statique vs dynamique

La coordination est une tâche essentielle dans les SMR. La performance globale du système est directement influencée par la qualité de la coordination et la qualité du contrôle de l'exécution de chaque mouvement de chaque robot impliqué. Cette coordination peut être statique ou dynamique. La coordination statique (également connue sous le nom de coordination délibérative [22], ou coordination hors-ligne [23]), se réfère à l'adoption d'une convention entre plusieurs robots avant de s'engager dans une tâche. Dans certaines règles comme « garder la droite », « s'arrêter aux intersections » et « garder suffisamment d'espace devant soi » sont utilisés pour résoudre les problèmes de contrôle de trafic. La coordination dynamique (également connue sous le nom de coordination réactive [22], ou coordination en ligne [23]) est faite lors de l'exécution d'une tâche et basée sur l'analyse et sur la synthèse de l'information. Les informations sont obtenues par des moyens de communication inter-robots [24].

I.8.5. Communication

La communication comme moyen de coordination apparaît souvent comme un comportement rationnel dans les environnements multi-robots. Dans les faits, la communication est un mode d'interaction entre les robots. Par cette interaction, les robots

peuvent d'une part partager les informations de position, l'état de l'environnement et les données des capteurs avec les autres dans le système.

D'autre part, chaque robot peut individuellement obtenir des informations sur les intentions, les objectifs et les actions de chacun des autres robots.

Cao et al [25] ont classé la structure de communication en trois types selon le mode d'interaction qui comprend : l'interaction via l'environnement, l'interaction via les perceptions et l'interaction via la communication.

I.8.6. Planification

Le principe de prévoir une séquence d'actions permettant d'atteindre un objectif est appelé planification [18]. D'un point de vue opérationnel, il s'agit d'exécuter ce plan définissant une succession d'actions pour atteindre l'objectif recherché.

Dans un SMR, la planification est utilisée pour coordonner des robots leur permettant ainsi d'accomplir une mission. Dans le cadre des SMR, la planification optimale est un problème NP-difficile [26].

I.8.6.1. Planification de mouvements

Le problème de la planification de mouvements correspond à la question de la génération d'une série de mouvements continus d'une configuration initiale à une configuration finale dans l'espace des configurations (i.e. l'espace des états du robot défini par l'emplacement, l'orientation et les angles des articulations [18]), tout en évitant les collisions avec les obstacles. Le mouvement est représenté comme un trajet dans l'espace de configuration. La planification de mouvements est éminemment nécessaire pour un robot, puisque par définition, un robot accomplit des tâches en se déplaçant dans le monde réel [27].

La planification de mouvements multi-robots tient compte non seulement des obstacles (statiques ou dynamiques) de l'environnement, mais aussi des interférences possibles entre les robots. Répartis en équipe et destinés à effectuer des tâches indépendantes dans un espace de travail partagé, les robots deviennent mutuellement des obstacles mobiles pour chacun. Ainsi, chaque robot doit prendre en compte le mouvement des autres robots pour mener à bien sa mission [28].

I.8.6.2. Planification de tâches

La planification de tâches multi-robots comprend deux principaux problèmes : la décomposition en tâches et la répartition de ces tâches vers un ensemble de robots. Les résultats de la répartition des tâches sont directement dépendants de la première étape de décomposition.

La décomposition en tâches multi-robots se réfère à la décomposition de la globalité de la mission ou à la décomposition d'une tâche de cette mission. Cette mission ou cette tâche est décomposée en plusieurs sous-tâches simples qui peuvent être réalisées indépendamment par un ou plusieurs robots, selon la caractéristique de la mission, l'exigence recherchée et la répartition des ressources de la mission [14].

I.9. Conclusion

La robotique est un domaine pluridisciplinaire par excellence effectivement il implique de nombreuses thématiques telles que la mécanique, la mécatronique, l'électronique, l'automatique, l'informatique et l'intelligence artificielle.

L'expansion des robots dans les différents domaines est une excellente chose, car elle permet d'accomplir des tâches dangereuses et monotones pour l'homme et leur rendement est beaucoup plus avantageux et stimulant pour l'économie.

Dans ce chapitre, on a abordé quelques notions de base sur la robotique, et notamment sur le domaine démutisation des robots. Le chapitre suivant on présente la description technique du robot mobile khepera IV.

Chapitre II

Chapitre II :

Description technique du robot mobile Khepera IV

II.1. Introduction

Khepera a d'abord été conçu comme un outil de recherche et d'enseignement dans le cadre d'un programme suisse de recherche prioritaire. Il permet la confrontation au monde réel des algorithmes développés dans la simulation pour l'exécution de la trajectoire, l'évitement des obstacles, le prétraitement de l'information sensorielle, l'hypothèse sur le traitement des comportements [29].

On va commencer ce chapitre par un petit aperçu historique sur les différentes versions de robot Khepera, et mettre en évidence la quatrième génération du robot Khepera et les composants existant dans cette génération et leur caractéristique.

II.2. Historique sur les robots Khepera

Khepera est un petit robot mobile pour recherche et enseignement. Des modules additionnels peuvent être très simplement ajoutés (caméras, gripper, entrée sortie (IO) interface). (Plus de 700 unités vendues à la fin 1998).

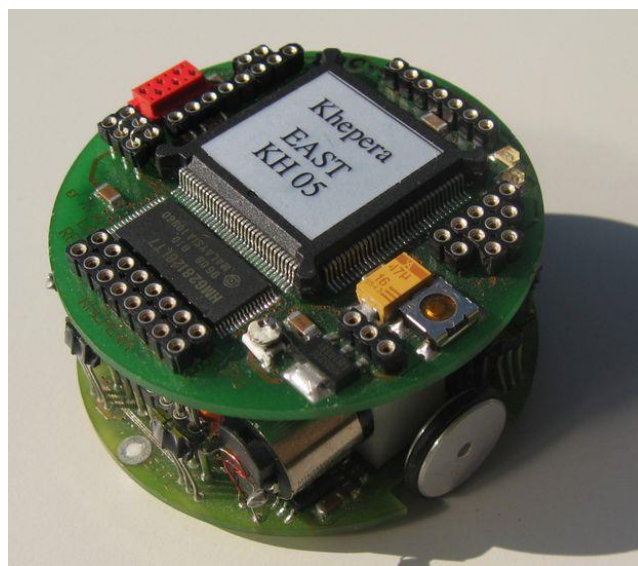


Figure II.1 : Le premier robot Khepera publié en 1996

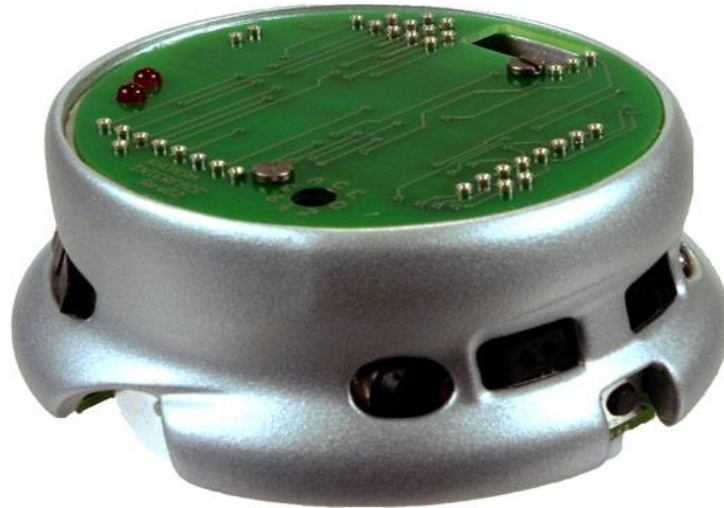


Figure II.2 : Khepera II



Figure II.3 : Khepera III



Figure II.4 : Khepera IV

II.3. Généralité sur le robot Khepera IV

II.3.1. Définition [30]

Le robot Khepera IV est la 4^{ème} version du robot mobile Khepera pour l'éducation et la recherche. Ce robot mobile compact a été conçu pour des surfaces planes (sol du laboratoire, table etc.). Il embarque ce qui se fait de mieux en termes de capteurs et moteurs de précision.

Cette nouvelle version du robot Khepera apporte de nombreuses nouveautés : OS Linux, caméra couleur, Wi-Fi, Bluetooth, Host USB, accéléromètre, gyroscope et une odométrie améliorée et plus précise.

Le robot mobile Khepera IV peut être utilisé pour la recherche et l'éducation en navigation, intelligence artificielle, les systèmes multi-agents, le contrôle, les comportements collectifs, la programmation temps - réelle etc.

II.3.2. Description physique

Le robot mobile Khepera IV est un petit robot différentiel à roues conçu pour l'utilisation intérieure. Il a la forme d'un cylindre, avec un diamètre de 14,08 cm et une hauteur au sol de 5,77 cm (y compris les roues). Sa coquille extérieure est composée de deux pièces en plastique avec fentes pour les capteurs et actionneurs. La figure II.5 montre les dessins techniques pour le robot. Le robot complet pèse 566 g.

Les deux roues actionnées ont un diamètre de 42 mm (comprenant les joints toriques qui servent de pneu) et sont centrés sur chaque côté du robot espacés de 10,54 cm, transfert de deux boules unités, à l'avant et à l'arrière, fournir le reste points de contact, cette solution aboutit à 0.5-1 mm de dégagement moulu, rendant le robot très stable mais empêchant son utilisation sur une surface qui n'est pas à plat et lisse.

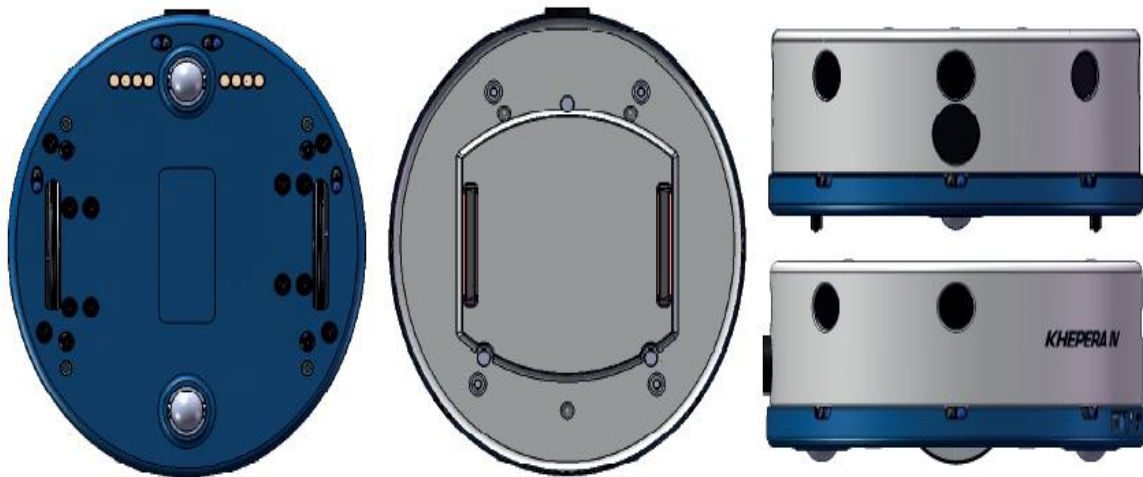


Figure II.5 : Vue du bas, du haut, de l'avant et de la gauche du robot

II.4. Fonctionnalités principales du robot mobile Khepera IV [30]

Le robot khepera IV est :

- Compacte ;
- Hautement modulaire ;
- Forte capacité de calcul embarqué ;
- Nombreux capteurs embarqués ;
- Moteur courant continue (CC) précis et de haute qualité ;
- Nombreux points de fixation pour adapter vos extensions.

II.5. Caractéristiques techniques du robot Khepera IV [30]

Les caractéristiques techniques du robot Khepera IV sont les suivantes :

- Processeur : Linux embarqué fonctionnant sous ARM Cortex-A8 @ 800 MHz + DSP Core (C64x) ;
- RAM : 512 Mo ;
- Mémoire : 512 Mo NAND flash + 4 Go on micro SD card ;
- Motorisation : 2 moteurs CC à balais avec encodeurs incrémentaux (environ 147 pulses par mm de déplacement du robot) ;
- Vitesse maximale : 1 m/s directement et 0,9 m/s en utilisant le PID par défaut ;
- Contrôleur de vitesse : Min : 0.003 m/s ;
- Capteurs :
 - ❖ 8 capteurs de proximité infrarouge pour la navigation ;

- ❖ 4 capteurs infrarouges pour le suivi de ligne et la détection de rebord ;
- ❖ 5 sonars à ultrasons pour la détection d'obstacle à longue distance (de 25 cm à 250 cm) ;
- ❖ Accéléromètre 3 axes et gyroscope 3 axes.
- Audio : 2 microphones embarqués ;
- 1 haut-parleur : 0.7 W 400 à 20000 Hz ;
- Vidéo : Caméra couleur intégrée (752x480 pixels, 30 FPS) ;
- Pack batterie à 7.4 V / 3400 mAh ;
- Autonomie : entre 4 et 7 heures ;
- Communication : 1x port USB 2.0 500 mA (maître) et 1x port USB 2.0 500 mA (esclave) ;
- Connectivité Bluetooth ;
- Wifi 802.11 b/g ;
- Taille : Diamètre de 140 mm, hauteur de 58 mm ;
- Garde au sol : 4 mm. Le robot Khepera IV s'utilise sur une surface plane et solide ;
- Rayon de virage : 0 cm ;
- Indication : 3 LED RVB sur le dessus ;
- Poids : 540 g ;
- Charge utile : environ 2 kg ;
- Plage de température opérationnelle : entre 0°C et 40°C ;
- Environnement de développement pour comportement autonome GNU C/C++ compilé.

II.6. Les différents composants du robot mobile Khepera IV

II.6.1. Vue globale du robot

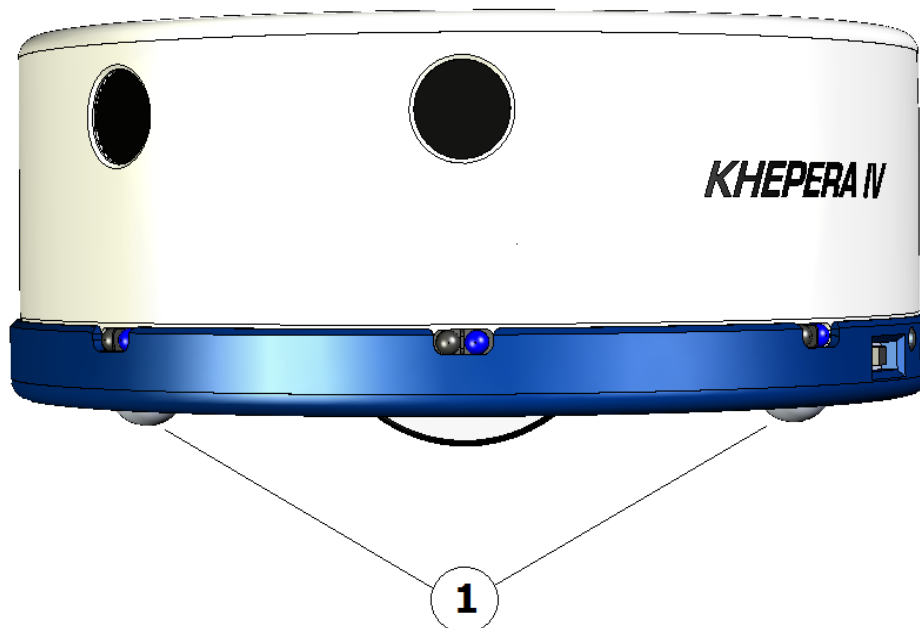


Figure II.6 : Vue de gauche

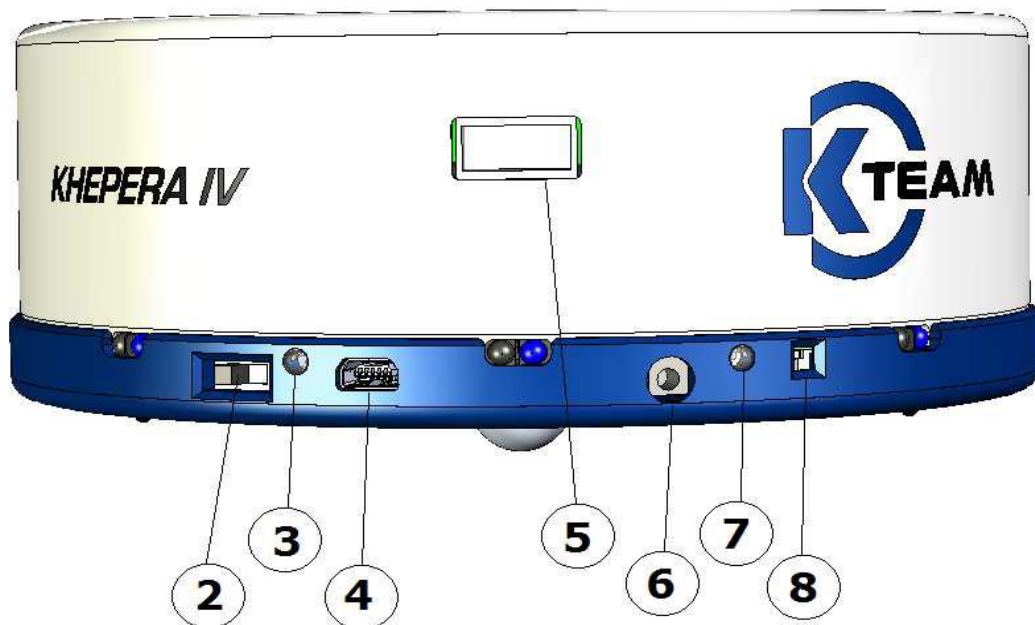


Figure II.7 : Vue arrière

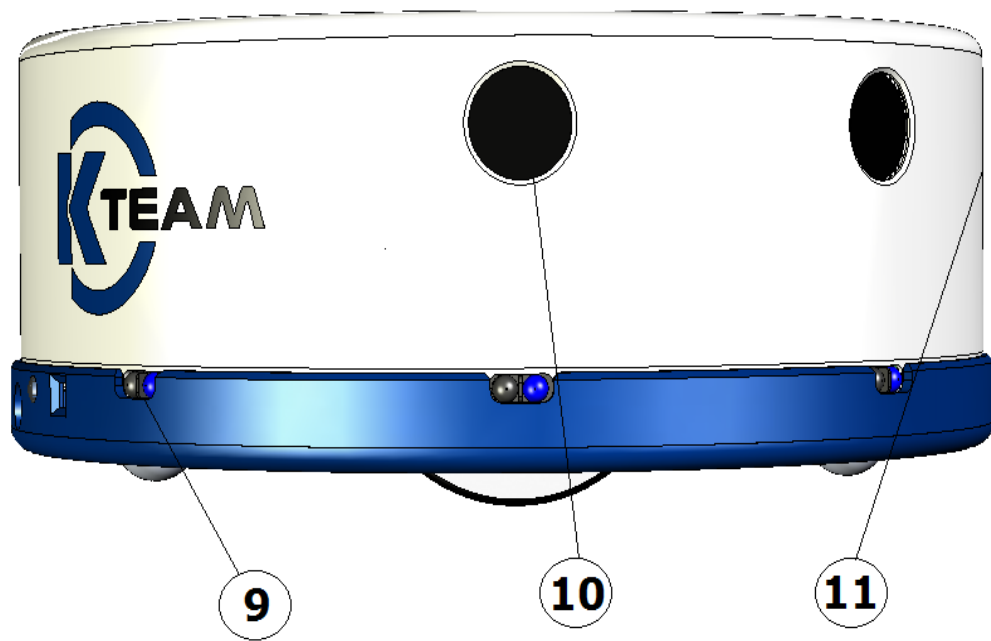


Figure II.8 : Vue de droite

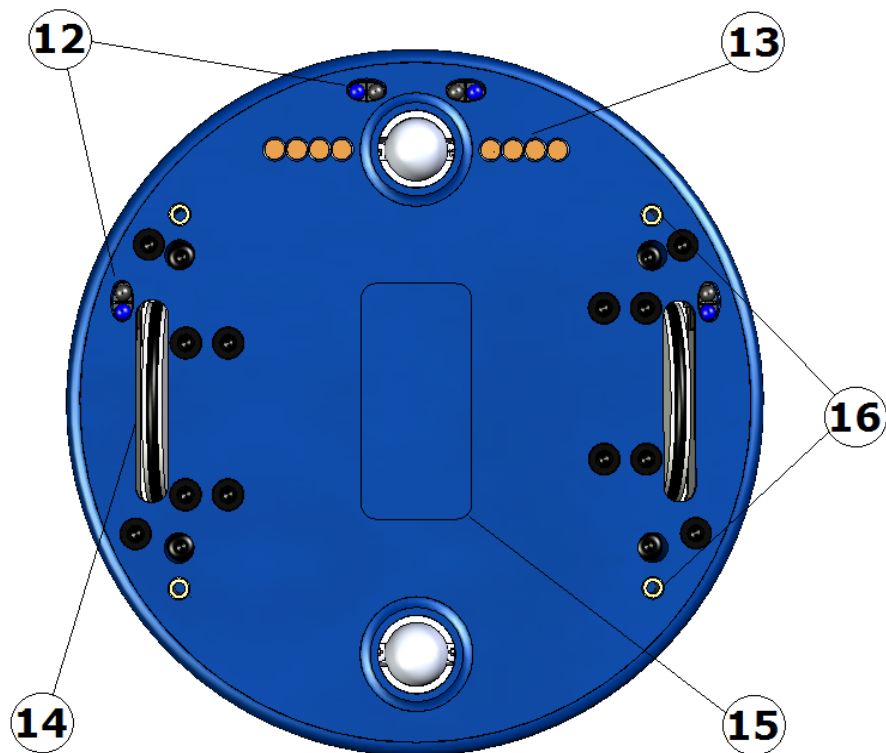


Figure II.9 : Vue de dessous

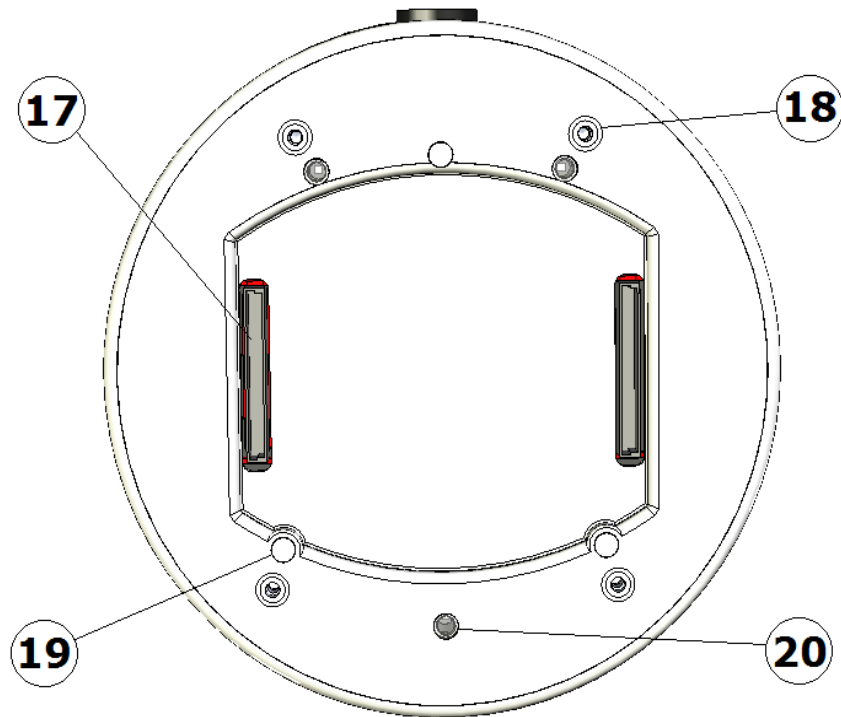


Figure II.10 : Vue de dessus

- 1 Les roues pivotantes
- 2 Bouton marche-arrêt
- 3 LED d'état
- 4 Connecteur Mini-USB B (mode appareil, sans charge)
- 5 Connecteur USB A (mode hôte, 500 mA)
- 6 Prise d'alimentation (9 V, 1.5 A, 0.65 mm centre positif)
- 7 Voyant d'état de charge
- 8 Bouton de réinitialisation
- 9 Capteurs infrarouges (8x)
- 10 Capteurs à ultrasons (5x)
- 11 Caméra
- 12 Capteurs à infrarouges de fond (bas) (4x)
- 13 Contacts pour station d'accueil
- 14 Roues
- 15 Autocollant
- 16 Bas écrous M3
- 17 Connecteurs d'extension 250 Ko
- 18 Haut écrous M3

19 Aimants

20 LED RVB

II.6.2. Vue détaillé des composantes du robot

II.6.2.1. Les roues pivotantes

Il y a deux roues pivotantes en dessous du robot. Ces deux roues permettent au robot d'être très stable, même à forte charge utile ou avec les modules d'extension d'en porte-à-faux long. En retour, le robot n'est pas en mesure de passer par les seuils de port.

II.6.2.2. Bouton marche-arrêt

Ce commutateur aura une action sur la réglementation interne du robot et non pas sur la batterie, ce qui signifie que vous pouvez le charger même s'il est éteint. Pour activer le robot, mettre le commutateur sur l'intérieur. Pour le désactiver, mettre le commutateur sur l'extérieur.

II.6.2.3. LED d'état

Par défaut, ce voyant bicolore vous permet de connaître l'état du robot. Lorsque vous le mettez sous tension, le voyant vert reste allumé et le voyant rouge clignote jusqu'à ce que le système soit prêt. Cette LED est contrôlable par l'utilisateur.

II.6.2.4. Connecteur Mini-USB B (mode appareil, sans charge)

Ce connecteur vous permet d'ouvrir un lien de communication entre le robot et un ordinateur. Il n'est pas possible de charger le robot de cette manière.

II.6.2.5. Connecteur USB A (mode hôte, 500 mA)

C'est un hôte USB connecteur compatible. Vous pouvez brancher sur ce périphérique USB que vous voulez, aussi longtemps qu'il ne pioche pas plus que 500 mA de courant. Vous pouvez par exemple connecter un module GPS ou une clé de mémoire.

NB : Ce connecteur n'a pas ses signaux partagés avec le KO-250. Les connecteurs d'extension, ce qui signifie que vous ne pouvez pas utiliser un port USB sur les postes en même temps que sur ce connecteur.

II.6.2.6. Prise d'alimentation (9 V, 1.5 A, 0.65 mm centre positif)

Il s'agit de la prise centrale positive de 0,65 mm utilisée pour charger la batterie interne du robot. Utilisez uniquement l'adaptateur fourni. La tension d'entrée est de 9 V. Courant consommé par le robot est 1A, de sorte qu'un 1,5 + un adaptateur est nécessaire.

II.6.2.7. Voyant d'état de charge

Il s'agit d'une LED d'indication bicolore indiquant l'état de la charge. Là sont deux modes, selon si le robot est sous tension ou non.

Si le robot est arrêté et que l'adaptateur secteur est branché, le voyant rouge est allumé aussi longtemps que la batterie est en charge. Une fois la charge terminée, le voyant s'éteint.

Si le robot est allumé et que l'adaptateur secteur est branché, la LED rouge s'allume pendant la charge de la batterie. Une fois la charge terminée, la LED rouge s'éteint et la LED verte s'allume.

II.6.2.8. Bouton de réinitialisation

Ce bouton permet de réinitialiser l'ensemble de robot, y compris les modules d'extension.

II.6.2.9. Capteurs infrarouges

Khepera IV a 8 capteurs infrarouges placés tout autour du robot et 4 placés sur le fond. Ces derniers permettent aux expériences comme ligne suivant ou l'évitement de l'obstacle. Ils sont placés et numérotés comme indiqué dans la figure ci-dessous :

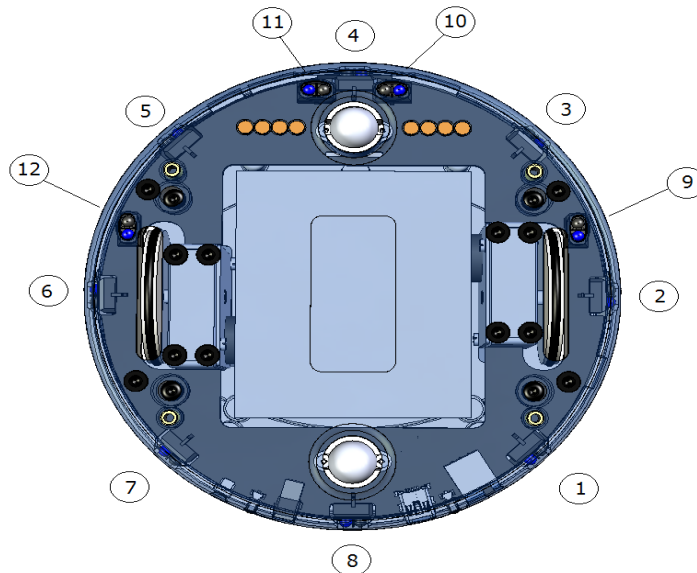


Figure II.11 : Capteurs à infrarouges vus de fond

Ces capteurs intègrent une lumière infrarouge émettrice et le récepteur. Pour une description détaillée, veuillez-vous reporter à la fiche technique du fabricant.

Les douze capteurs sont TCRT5000 capteurs optiques réfléchissantes de Vishay Telefunken. Plage de mesure est de 2 à 250 mm et une longueur d'onde de 950 nm. Chaque capteur est séparé de son voisin d'un angle de 45°. La fréquence d'échantillonnage des capteurs infrarouges est de 200 Hz, quel que soit le mode de fonctionnement.

Ce type de détecteurs permet deux mesures :

- La lumière ambiante normale : Cette mesure est réalisée en utilisant uniquement la partie de l'appareil récepteur, sans émettre de la lumière avec l'émetteur.

Une nouvelle mesure est effectuée tous les 5 ms. La valeur renvoyée à un moment donné est le résultat de la dernière mesure effectuée ;

- La lumière réfléchi par les obstacles : Cette mesure est réalisée en émettant de la lumière à l'aide de l'émetteur de l'appareil. La valeur retournée est la différence entre la mesure effectuée en émettant de la lumière, et la lumière mesurée sans émission de lumière (lumière ambiante). Une nouvelle mesure est effectuée tous les 5 ms. La valeur renvoyée à un moment donné est le résultat de la dernière mesure effectuée.

II.6.2.9.1. Mesure de la lumière ambiante

Mesure de la lumière ambiante est fortement influencé par l'environnement du robot. Selon le type de source de lumière, la couleur, et de la distance de mesure de la lumière ambiante, le profil peut varier. Il n'est pas recommandé d'utiliser avec une grande source de lumière dans l'émission infrarouge, car cela pourrait perturber les capteurs IR. Valeurs 0 à 1023, 0 signifie aucune lumière et 1023 pour la pleine lumière.

II.6.2.9.2. Mesure de la lumière réfléchi (proximité)

Les capteurs sont principalement destinés à détecter des obstacles autour du Khepera. Les mesures pour la lumière réfléchi dépendent de la réflectivité des objets et de la lumière ambiante. Les couleurs de l'objet, les matériaux et les surfaces n'ont une influence sur la réponse du capteur. En outre, comme tout capteur, capteurs IR sont soumis au bruit dans l'environnement. Pour toutes ces raisons, les graphiques ci-dessous sont donnés à titre indicatif et ne doivent pas être considérés comme des références. Valeurs 0 à 1023 bit, 0 signifie Pas d'obstacle, 1023 bit pour de très près de l'obstacle :

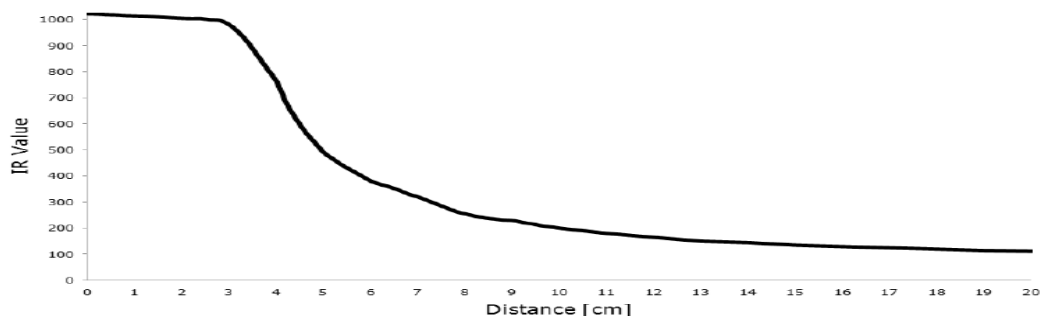


Figure II.12 : Valeur IR par rapport à la distance

La valeur IR ne tombe jamais à 0, puisque, même sans obstacle, l'IR se reflète sur le sol et ajoute une valeur statique. Comme tous les capteurs ne sont pas exactement les mêmes, la solution est d'effectuer un étalonnage de l'IR sans obstacle devant. Avec cet

étalonnage, l'utilisateur sera en mesure d'améliorer la détection d'obstacle à une distance supérieure à 20 cm.

II.6.2.10. Capteurs à ultrasons

Cinq capteurs sont placés autour du robot et sont positionnés et numérotés comme indiqué dans la figure ci-dessous. Ces capteurs sont des émetteurs-récepteurs, ce qui signifie qu'ils peuvent émettre et recevoir les impulsions.

Les capteurs à ultrasons sont entraînés par un Vpp 85 source. La fréquence nominale de ces capteurs est de 40 kHz +/- 1 kHz. La valeur retournée est la distance à l'objet en centimètres, avec une tolérance de +/-2 cm. Plage de mesure est de 25 à 200 cm. Chaque transducteur est séparé de son voisin d'un angle de 45°.

Chaque capteur peut être désactivée afin d'obtenir des taux de rafraîchissement pour un particulier (ou un groupe). Un capteur mesure prend 20 ms. Tous les cinq capteurs besoin de 100 ms à être lu.

Le taux d'échantillonnage effectif dépend du nombre des capteurs activés, allant de 50 Hz pour un seul capteur à 10 Hz si l'ensemble est en utilisation.

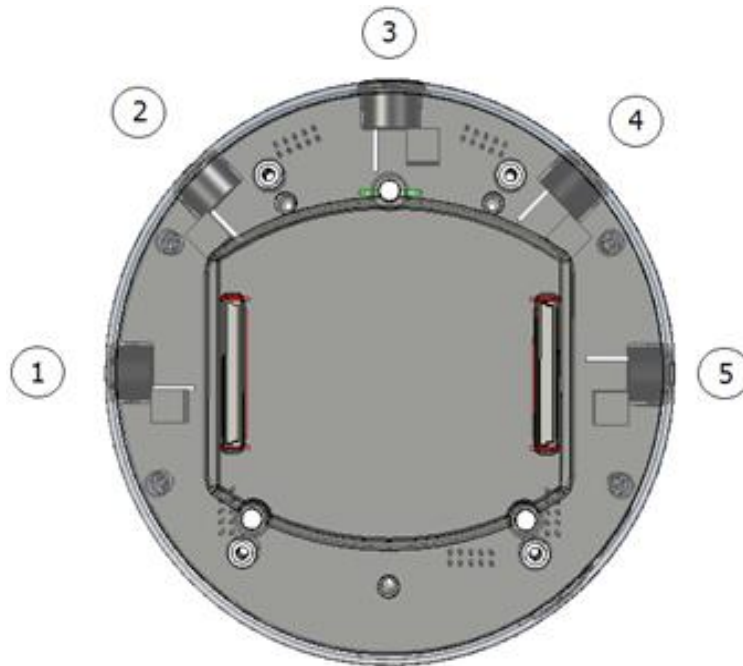


Figure II.13 : Capteurs ultrasoniques vus de sommet

II.6.2.11. Caméra

Dans l'avant du robot, il y a une caméra couleur avec objectif interchangeable par l'utilisateur. Il peut être utilisé pour prendre des photos ou des films qui peuvent être traitées à bord. Le Khepera IV est équipé d'un appareil photo couleur, éliminés sous l'avant du

capteur à ultrasons. Le capteur est un MT9V034C12ST de Aptina. C'est un capteur CMOS 1/3" WVGA.

- Taille d'imageur active : 4.51 x 2.88 mm
- Pixels actifs : 752 x 480

Le réglage par défaut de l'objectif a une focale de 2,1 mm, avec filtre anti-IR et mise au point fixe. La fixation filetage M12x0.5. C'est champ de vision diagonal de 150°, horizontal est 131° et vertical est 101°.

II.6.2.12. Capteurs à infrarouges de fond (bas)

Quatre capteurs infrarouges sont disposés sur la partie inférieure du robot. Ils sont utilisés pour éviter le robot de tomber mais peut également être utilisé pour suivre une ligne.

II.6.2.13. Contacts pour station d'accueil

Le Khepera IV a montré quelques contacts ci-dessous son corps qui signifie que vous pourriez utiliser une station d'accueil pour charger sa batterie ou de communiquer avec elle. Il est également possible d'imaginer de développer une extension qui serait en dessous du robot, par exemple une base mécanique avec les voies. Les signaux qui sont offerts sont : Batterie (de contrôlé par un relais de roseau), 9V dans l'I2C.

II.6.2.14. Roues

Le robot est conduit du différentiel, avec 2 roues équipé d'un joint torique. Les roues sont entraînées par des moteurs à courant continu (CC) avec encodeur et boîte de vitesses.

II.6.2.15. Autocollant

Ici vous trouverez le numéro de série de votre robot.

II.6.2.16. Bas écrous M3

Il y a la possibilité de fixer une extension vers le robot à partir de ci-dessous.

II.6.2.17. Connecteurs d'extension 250 Ko

Ces deux connecteurs sont utilisés pour connecter les modules d'extension pour le robot. La broche de sortie est en conséquence à Ko-250, avec quelques améliorations mineures. Vous pouvez réutiliser Khepera III ou les modules d'extension KoreBot (mais pas le Grappin Kh3).

NB : Ces connecteurs ont leurs signaux d'hôte USB partagé avec l'hôte USB Connecteurs, ce qui signifie que vous ne pouvez pas utiliser un port USB sur les postes en même temps que sur le connecteur. Périphérique USB est utilisable sur les deux en même temps que le robot est équipé d'un moyeu.

II.6.2.18. Haut écrous M3

Il y a la possibilité de fixer une extension vers le robot à partir de ci-dessus.

II.6.2.19. Aimants

Il y a trois aimants utilisés principalement pour fixer le grappin, mais ils peuvent être utilisés pour n'importe quel autre module.

II.6.2.20. LED RVB

Trois LED RVB (19-337/GHBHC6R-A01/2T d'Everlight) sont montés sur le haut de la carte principale. Chacune de ces LED est dotée d'un guide de lumière sur elle.

Les LED sont entraînées par un pilote de LED dédiés (LTC3219 de linéaire) qui offre une résolution de 6 bits (0-63) pour chaque couleur.

Ces LED peut être utilisée pour localiser le Khepera IV avec un appareil photo et de différencier chaque robot (en application de l'essaim). Comme les LED sont placées sur un triangle isocèle, la direction du robot peut également être détectée.

II.6.3. En profondeur

II.6.3.1. Microphones

Le Khepera IV est équipé de deux microphones amplifiés PU0414HR5H-SB de Knowles. Ils sont directement connectés à l'Overo l'entrée micro. Le droit microphone est connecté à l'entrée principale et le micro à gauche sur l'entrée MIC SOUS.

- Gain 20 dB
- Sensibilité (type) -22 dbv/Pa
- Directivité Omnidirectionnelle
- Tension d'alimentation 2.5 V

Leurs positions sur l'sont Khepera indiqué sur la photo ci-dessous :

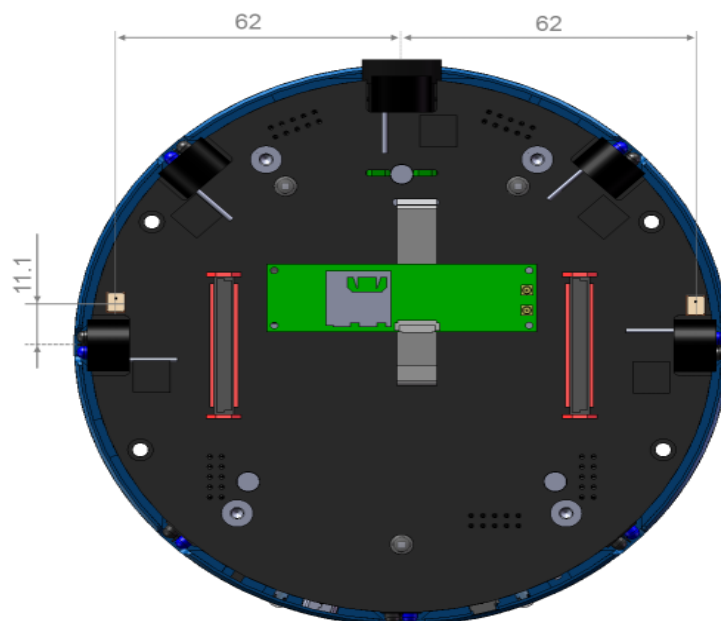


Figure II.14 : Positions de microphones

II.6.3.2. Haut-parleur

Un SMS-1308MS-2-R haut-parleur de PUI audio est monté sur le Khepera IV. Ce haut-parleur est entraîné par un 1W basse distorsion amplificateur de puissance. Le haut-parleur est connecté sur la sortie audio de l'HSOLF OVERO. L'OVERO peut aussi couper l'amplificateur avec GPIO64 (0 = muet, 1 = ampli).

- Le président Power 0.7 W (max 1W)
- Impédance 8 Ohms
- SPL de sortie 88 dB
- Distorsion (max) 5 %
- Fréquence de résonance 850 Hz $\pm 20\%$
- Gamme de fréquence 400 ~ 20000 Hz

II.6.3.3. Accéléromètre

L'accéléromètre monté sur le Khepera IV est une LSM330DLC de ST. Ce dispositif inclut dans un paquet un accéléromètre 3D et un gyroscope 3D.

L'appareil est exactement au centre du robot (placé sur le centre de rotation). L'appareil est situé sur le haut de la carte principale.

L'accéléromètre est orienté vers l'axe 1 à l'avant droit, et cela renvoie une valeur positive pour l'axe X pour aller de l'avant. L'axe Y est positif sur la gauche, et enfin de l'axe Z est négatif avec la gravité.

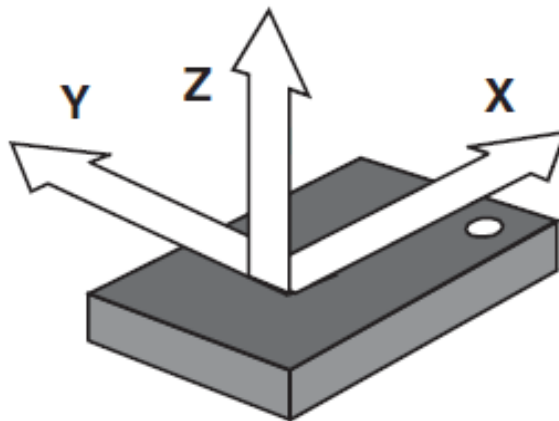


Figure II.15 : Les directions d'accélération détectables

L'accéléromètre renvoie 12 bits de données (complément à 2) avec une gamme de ± 2 g. Cela signifie qu'une valeur de 1g sera de retour d'une valeur de 16384. Le débit de données est configuré à 100 Hz, comme le dsPIC des actualisations Khepera 10 valeurs à la fois, l'utilisateur doit lire toutes les 100 ms (10 Hz) pour obtenir de nouvelles données.

II.6.3.4. Gyroscope

Le gyroscope de Khepera IV est inclus dans le même package que l'accéléromètre. Les directions des taux angulaire détectables sont définies autour de l'axe de l'accéléromètre.

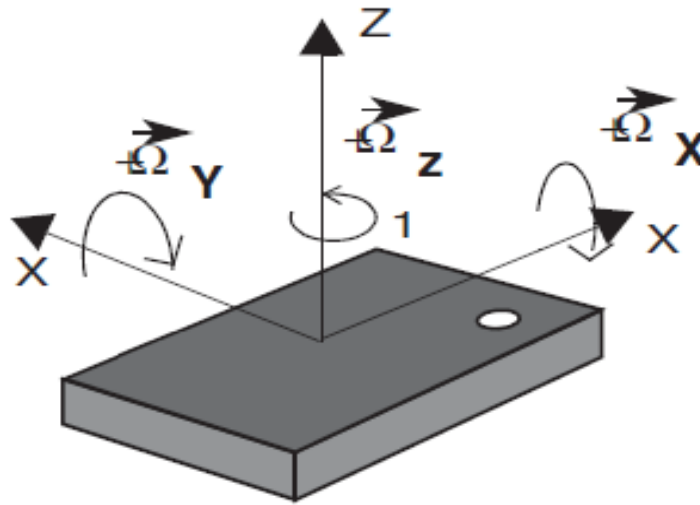


Figure II.16 : Les directions de taux angulaires détectables

Le format de données est sur 12 bits trop, la gamme complète est configuré à +/- 2000 dps (360 dps = 5898) et le débit de données est configurée à 95 Hz.

Le gyroscope, les données sont lues par des paquets de 10 valeurs à la fois, ce qui signifie que l'utilisateur peut lire les nouvelles données chaque 105 ms pour obtenir de nouvelles valeurs.

La sortie doit être multipliée par 0,066 d'avoir [deg/s] unités.

II.6.3.5. Périphérique USB (mini-USB B connecteur)

Un mini-USB B vous donne accès à un port USB/série adaptateur (FT234XD de FTDI) pour accéder directement à le ttyS2 de la Gumstix. En utilisant un terminal fournit l'accès à l'amorçage du système.

Lorsque vous connectez un ordinateur à ce connecteur pour la première fois, le système vous demande pour pilote. FT234XD pilote peut être trouvé à <http://www.ftdichip.com/Products/ICs/FT234XD.html>.

II.6.3.6. Carte MicroSD

Une carte MicroSD de 4 Go est fournie à l'intérieur du Khepera IV. Le robot démarre sur elle et utiliser cette carte. Il contient déjà l'OS, ainsi que les fichiers d'amorçage du noyau.

II.6.3.7. Moteurs

Le Khepera IV a eu 2 moteurs à courant continu afin d'entraîner ses deux roues. Les moteurs ont 1.96 W puissance nominale. La boîte de vitesses intégrée a un rapport de réduction de 19:1 et d'une efficacité de 78 %. Il y a une autre boîte de vitesses dans

le carter du bloc moteur, avec un ratio de 2:1 et d'une efficacité de 85 %. Ratio total est alors de 38:1 et l'efficacité est de 66,3 %, ce qui signifie qu'il est utilisable de 15 puissances mécaniques par roue.

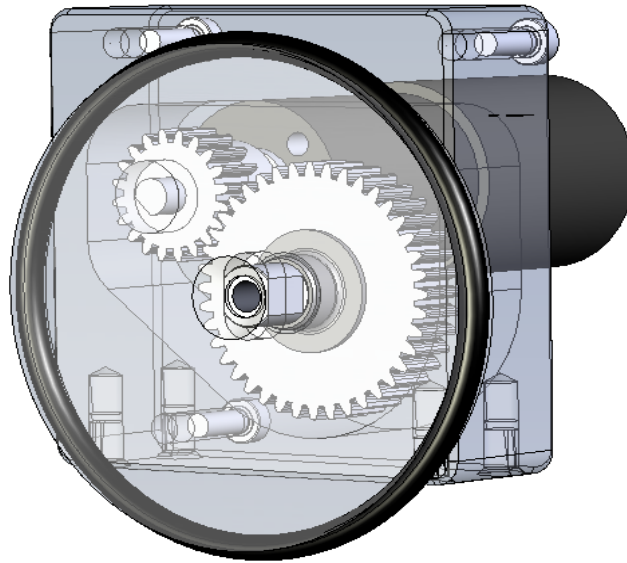


Figure II.17 : Bloc du moteur avec roue

II.6.3.8. Batterie

Le Khepera IV est équipé d'un processus interne non-amovible batterie lithium polymère. Il est construit dans une configuration 2S1P 7.4 V, 8.4 V, nominale Tension de charge et une capacité de 3400 mAh.

- Tension nominale 7.4 V
- La tension de coupure 6.0 V
- La tension de charge 8.4 V
- Capacité nominale 3400 mAh
- Courant de décharge max 3400 mA (1C)
- Courant de charge 1100 mA
- Temps pour une charge complète Environ 4 à 5 heures

Il n'y a pas de système de gestion de l'alimentation spécifique sur le Khepera. Lorsque la tension de la batterie tombe sous 6V, la batterie s'ouvre lui-même le circuit pour éviter une décharge profonde des cellules. Les utilisateurs peuvent mettre en œuvre leur propre système de gestion de l'alimentation du logiciel pour gérer les extensions à arrêter correctement avant de ce cas se produit.

La batterie peut être chargée à partir de 3 endroits :

- À partir de la prise ;

- Des contacts situés sous le robot ;
- Des connecteurs d'extensions 250 Ko.

II.7. Moyenne d'extension

La fonctionnalité native du robot peut être étendue Par l'utilisation de périphériques USB ou Bluetooth génériques, ou en concevant des cartes personnalisées enfichables dans le bus Ko-250. Cette connexion à 100 broches fournit une alimentation, I2C, SPI, Bus USB, ainsi que des lignes plus spécifiques pour, par exemple, Interface dsPIC. K-Team commercialise plusieurs planches, y compris un préhenseur, un télémètre laser etc.

II.7.1. La tourelle d'extension

La tourelle d'extension peut être branchée sur le haut du Khepera IV afin d'améliorer les capacités du Khepera IV. Cette tourelle offrira de nombreuses options différentes pour le robot :

- Télémètre laser (LRF) capacité de connexion (Hokuyo URG-04LX-UG01) ;
- StarGazer (global position intérieure et le roulement de la fonction de connexion) ;
- Batterie supplémentaire pour une plus grande autonomie avec la K4 pack chargeur externe ;
- L'accessibilité de l'IO.



Figure II.18 : La tourelle d'extension

II.7.2. Télémètre laser

La tourelle d'extension peut recevoir en option un module télémètre laser (LRF). Il est utilisé pour faire de la navigation, spécialement pour la cartographie et la localisation. Un laser à l'intérieur du capteur est balayage de l'environnement du robot, de mesurer chaque 0,36 degrés la distance à partir de n'importe quel objet à sa hauteur.



Figure II.19 : Extension télémètre laser

II.7.3. Le préhenseur

Le préhenseur est une tourelle qui peut être branché sur le robot. Il donne à Khepera IV la capacité de saisir et déplacer des objets.

Deux degrés de liberté pour la manipulation et le transport précis et robuste d'objets de 5 cm de largeur et 50 g de charge.

Fait à partir de pièces mécaniques de haute précision.

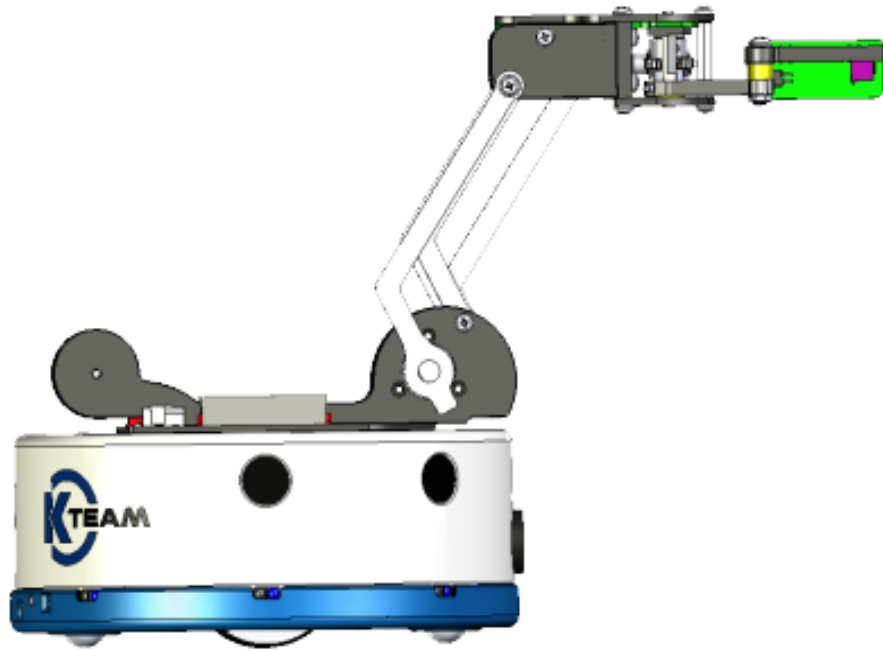


Figure II.20 : Extension avec le préhenseur

II.7.4. Système de positionnement global intérieur (Stargazer)

Le Stargazer est un module permettant au robot Khepera IV d'avoir une position globale à l'intérieur et porter sur des repères passifs, Il est utilisé pour faire de la navigation, spécialement pour la cartographie et la localisation.

Le module a un IR LED's pour l'éclairage des points de repère et une caméra pour capturer l'image, qui est traité par l'électronique embarquée. Il calcule la position et le roulement liés aux repères collés au plafond, Il utilise le module Stargazer.

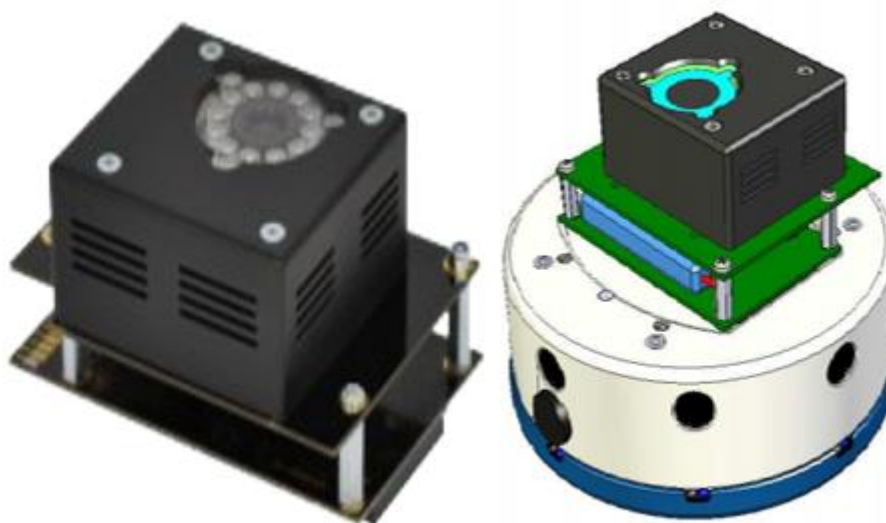


Figure II.21 : Le Stargazer

II.8. Conclusion

Dans ce chapitre on découvre les différents composants du robot mobile Khepera IV embarqué de nombreux capteurs et une puissance de calcul appréciable pour un robot mobile autonome, Et découvre aussi les fonctionnalités du robot (caméra couleur, Wi-Fi, Bluetooth, hôte USB, accéléromètre, gyroscope, microphone, haut-parleur, 3 LED RVB supérieures, odométrie améliorée et précision).

Ainsi que à la fin de ce chapitre, nous avons vu les différents moyens d'extension pour le robot mobile Khepera IV.

Chapitre III

Chapitre III :

Conception et implémentation

III.1. Introduction

Ce chapitre est consacré à la présentation des différentes étapes suivies pour se connecter au robot Khepera IV par la machine virtuelle (`Ubuntu_LTS_Khepera4_light_tools`).

Nous présenterons la programmation du robot Khepera IV par Windows avec le programme **putty** [31] et le pilotage du robot sans fil avec une interface en **Visual Studio C#**. La planification d'une trajectoire sera ensuite abordée. Nous finirons par une conclusion qui résume les tâches exécutées.

III.2. Précautions d'utilisation

Avant d'utiliser le robot pour la première fois, il faut respecter les règles ou conseils exigés par le mode d'emploi convenable spécifié par le constructeur et qui est le suivant :

- 1- Laisser la batterie se charger complètement pour au moins 5 heures. Pour ce faire, une fois que le robot est retiré de son emballage, on s'assure qu'il est éteint, on branche le chargeur dans la prise du robot et ensuite dans le secteur.
- 2- Ajuster soigneusement l'objectif de la caméra à sa place, en avant du robot. effectuer cette opération dès que le robot est retiré de son emballage pour éviter toute poussière sur le capteur. Il est recommandé de lancer une application de test pour ajuster correctement la mise au point. On s'assure d'enlever le couvercle de l'objectif pour l'utiliser. Si le robot a été livré avec l'objectif déjà installé, il faut s'assurer qu'il est propre. Sinon, on utilise une serviette en microfibre sec pour le nettoyer ;
- 3- On peut avoir un accès à la console de différentes façons : USB, Bluetooth et Wi-Fi.

III.3. Commande du robot mobile et communication via linux

III.3.1. Création d'une machine virtuelle avec VMware Workstation

Cette procédure nécessite les étapes suivantes :

1. Télécharger et installer la machine virtuelle, **VMware Workstation** en utilisant le lien de téléchargement suivant :

https://my.vmware.com/fr/web/vmware/info/slug/.../vmware_workstation/10_0 ;

2. Importer le fichier image « **Ubuntu_LTS_Khepera4_light_tools** » avec le menu Fichier → Import Appliance → **VMware Workstation** ; ce qui donne la fenêtre de dialogue figure III.1 ci-dessous :

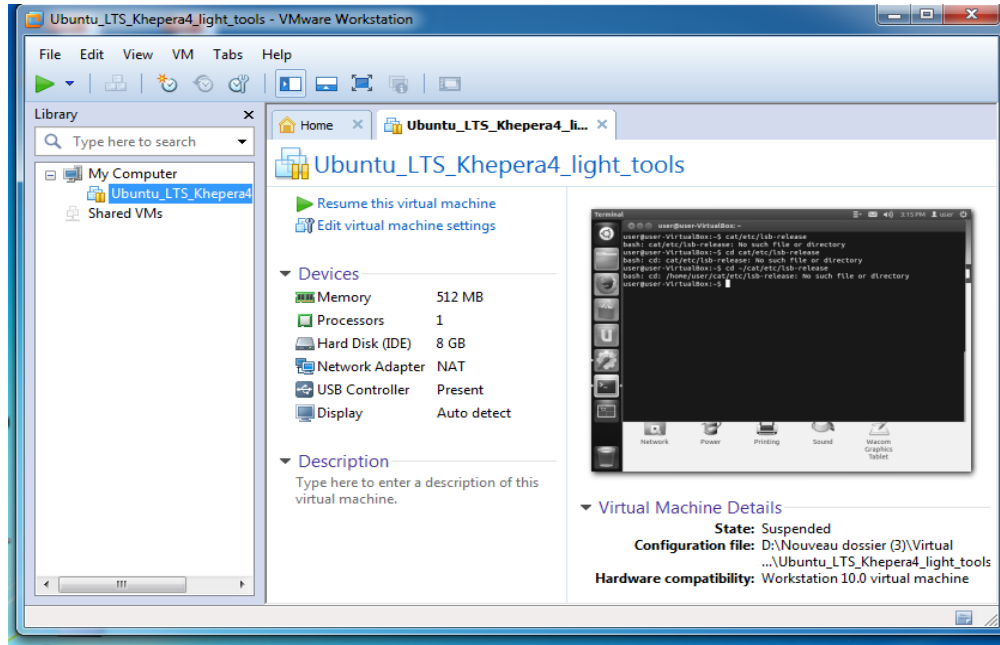


Figure III.1 : Interface de la machine virtuelle

3. Créer le répertoire **C : \virtual_machine_shared** sur votre ordinateur. Ça sera le répertoire partagé pour le transfert des données entre la machine virtuelle et notre ordinateur.
4. Démarrer la machine virtuelle avec l'image importée : Sa connexion se fait à l'aide de :

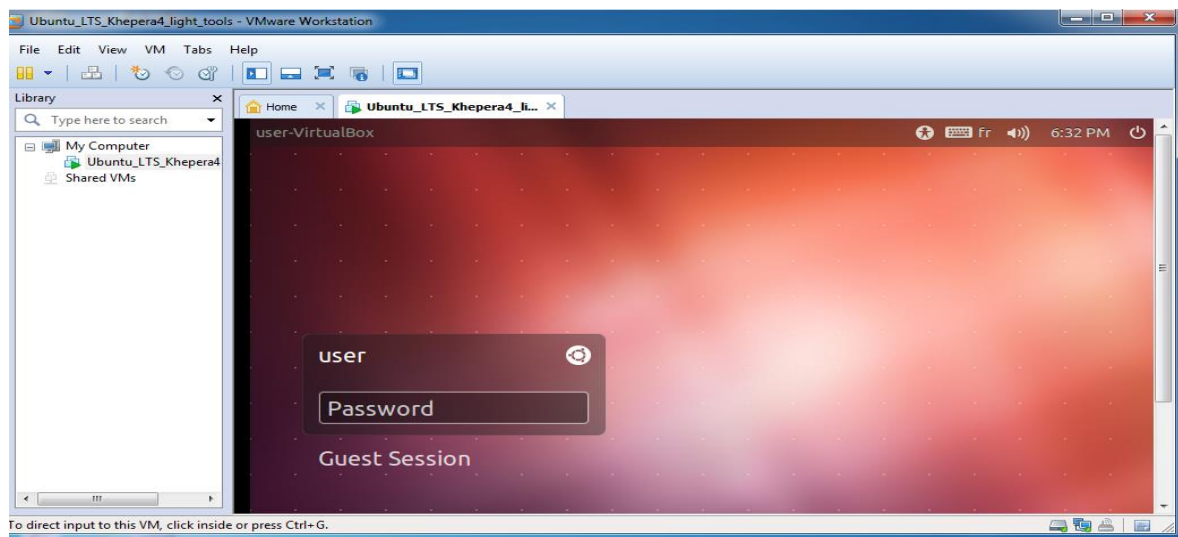


Figure III.2 : Champs de la machine virtuelle : Nom d'utilisateur et mot de passe

Après avoir obtenu la fenêtre de dialogue donnée par la figure III.2, il faut remplir les champs :

Nom d'utilisateur : **user**

Mot de passe : **root2014**

5. Les outils de développement sont déjà installés, le dossier de développement s'obtient par :

home → **user** → **Khepera4_development**

Cette succession de commande est illustrée dans la fenêtre suivante :

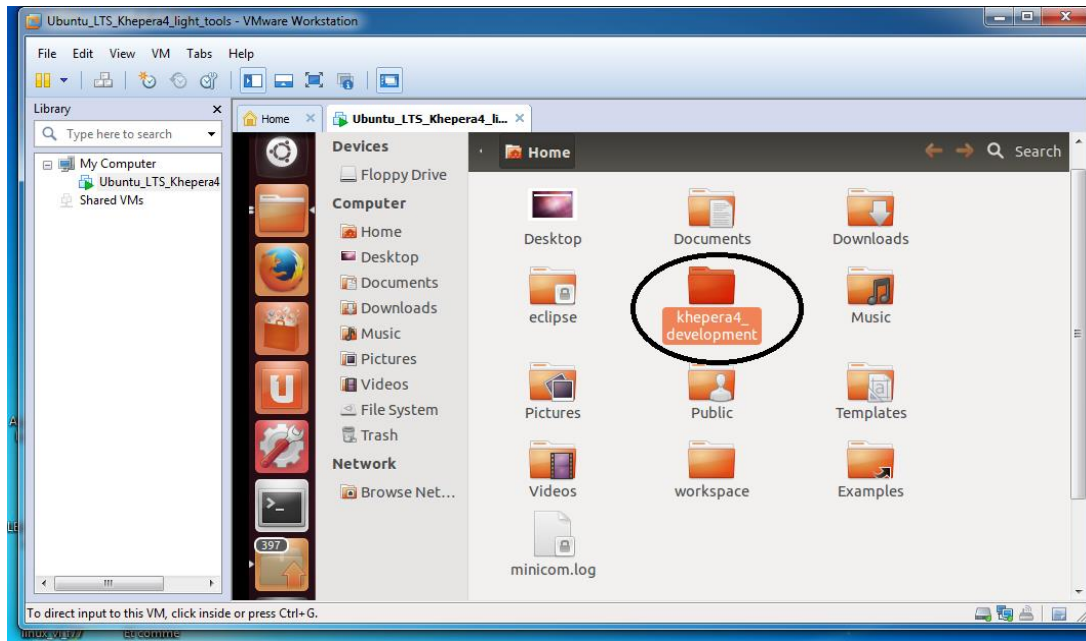


Figure III.3 : Emplacement des fichiers de la librairie

III.3.2. Commande du robot via linux

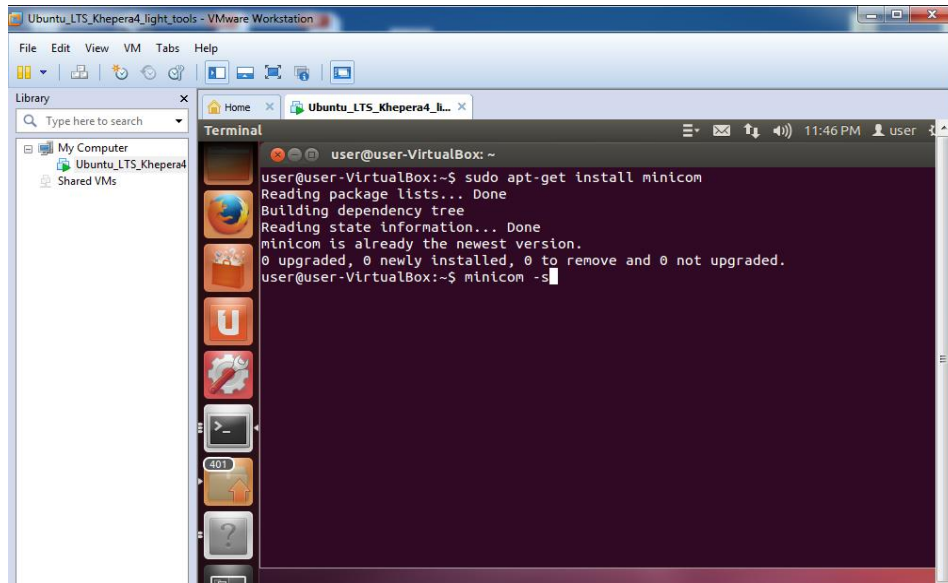
Pour commander le robot, nous devons suivre les étapes suivantes :

1. installer sur Windows les drivers **FTDI** avant de brancher l'USB du robot Khepera IV, en utilisant le lien d'installation suivant :

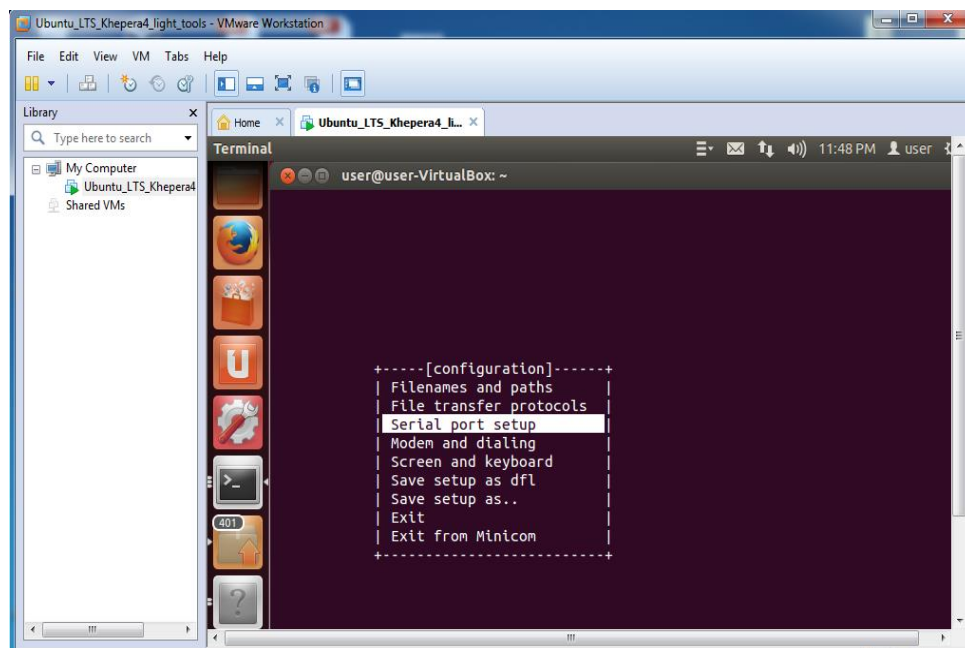
<http://www.ftdichip.com/Products/ICs/FT234XD.html> ;

2. Brancher le câble USB à l'ordinateur, et le côté mini-USB au robot ;
3. Allumer le robot ;
4. Sous linux, le pilote **FTDI** est installé par défaut, utiliser **minicom** pour l'installer sur Ubuntu. Nous écrivons dans le terminal du linux la commande suivante :

sudo apt-get install minicom

Figure III.4 : Installation du port **minicom**

5. Pour configurer **minicom**, nous avons utilisé la commande suivante : **minicom -s** ce qui a affiché la fenêtre ci-dessous :

Figure III.5 : Paramètres de configuration du port **minicom**

- ❖ En sélectionnant le menu « **Serial Port Setup** » la configuration choisie est décrite comme le montre la figure III.6 ;

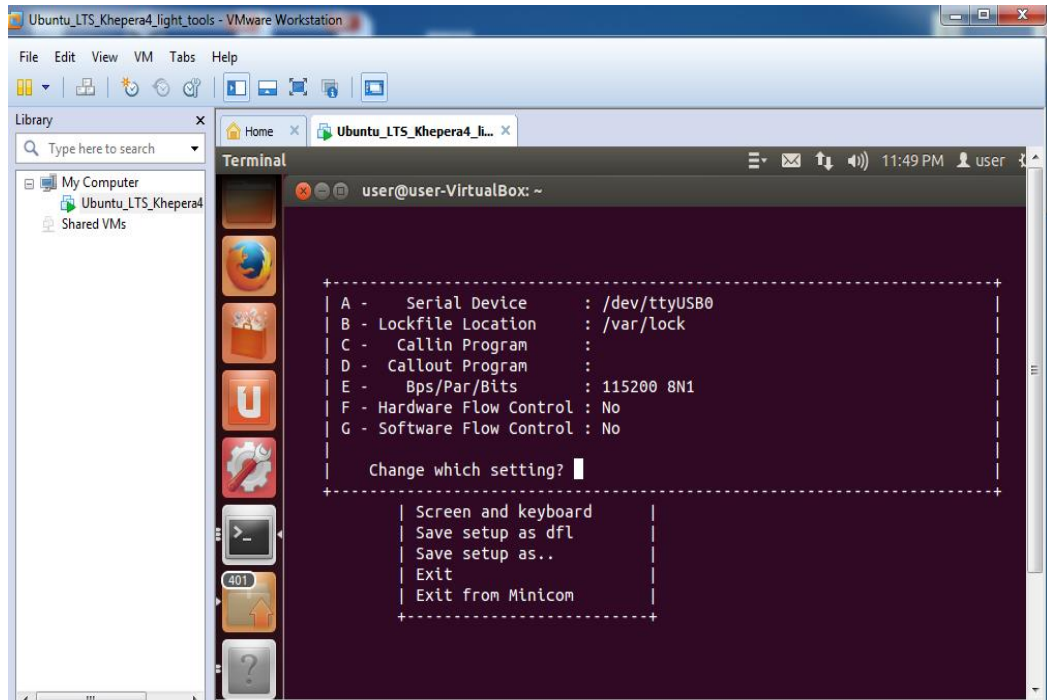


Figure III.6 : Configuration des paramètres du port minicom

- ❖ L'enregistrement des paramètres se fait avec la commande « **Save setup as dfl** » du menu (paramètres de configuration), puis **Exit** ;
- ❖ L'attend de la connexion est illustrée dans la figure III.7 ;

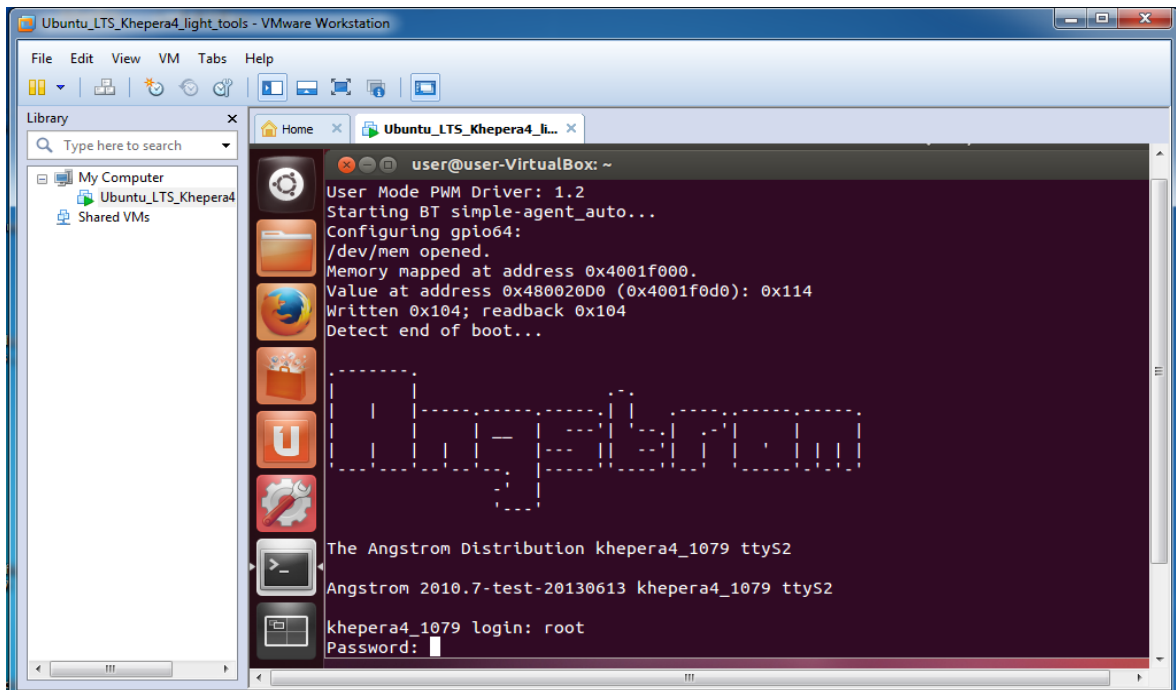


Figure III.7 : L'établissement de la connections entre le robot et linux

❖ Nous entrons :

Le nom d'utilisateur : **root**

Mot de passe : **bour**

6. Nous sommes dans le répertoire **/home/root** par défaut après la connexion. Nous pouvons exécuter un programme en précisant son nom avec **./ : ./khepera4_test**

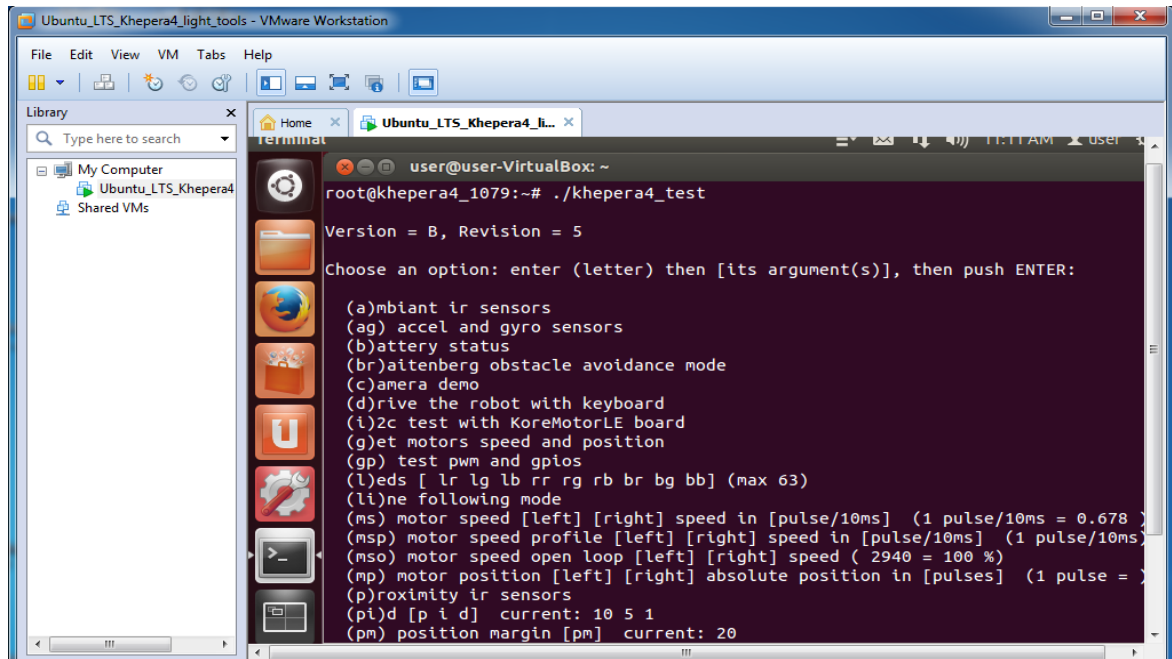


Figure III.8 : Liste des commandes du robot sous linux

III.3.2.1. Signification des commandes

- (a) Capteurs infrarouges ambiantes
- (ag) Capteur d'accélération et gyroscopique
- (b) L'état du batterie
- (br) Mode d'évitement des obstacles braitenberg
- (c) Démonstration de caméra
- (d) Conduire le robot avec le clavier
- (i) Test i2c avec carte koremotorLE
- (g) Obtenir la vitesse et la position des moteurs
- (gp) Test pwm et gpios
- (l) LEDs [lr lg lb rr rg rb br bg bb]
- (li) Mode suivant ligne
- (ms) Vitesse de la motrice [gauche] [droite], vitesse dans [impulsion / 10 ms]
(1 impulsion / 10 ms = 0,678)

- (msp) Profil de vitesse de la motrice [gauche] [droite], vitesse dans [impulsion / 10 ms]
(1 impulsion / 10 ms)
- (mso) Vitesse de la motrice de boucle ouverte [gauche] [droite], vitesse (2940 = 100%)
- (mp) Position de la motrice [gauche] [droite], position absolue dans [impulsions]
- (p) Capteurs infrarouges de proximité
- (pi) [p i d] courant : 10 5 1
- (pm) Marge de position [pm] courant : 20
- (s) Arrête le moteur
- (sp) Profil de vitesse
- (so) Démonstration du son
- (st) État des LEDs (rouge allumé = 1, vert allumé = 16, tous deux activé = 0, tous deux désactivés = 17)
- (re) Réinitialiser les encodeurs
- (rm) Réinitialiser le microcontrôleur
- (q) Quitter le programme
- (u) Capteurs à ultrasons
- (ua) Capteurs à ultrasons activés (1 : à gauche, 2 : avant gauche, 4 : avant, 8 : avant à droite)

✓ Appuyez sur **q** et sur la touche **entrée** pour redémarrer.

NB : Vous pouvez ajouter un mot de passe à la connexion de votre robot avec la commande : **passwd**

III.3.2.2. Arrêt du robot

Lorsqu'il est connecté à la console, il est préférable de faire un bon arrêt que d'éteindre simplement le robot. Cela garantira l'ouverture des fichiers qui sont correctement fermés et avec les paramètres enregistrés. Ceci est fait comme suit :

1. Exécuter la commande suivante dans la console : **poweroff**
2. Attendre 15 s (si vous êtes connecté au câble USB, vous attendez jusqu'à ce que la mise hors tension apparaisse). Ensuite, vous pouvez éteindre le robot.

NB : Si nous souhaitons redémarrer le robot, nous pouvons utiliser la commande suivante : **reboot**

III.3.3. Arrêt de la machine virtuelle

En fermant la machine, nous aurons les trois choix visualisés sur la fenêtre de la Figure III.9 et qui sont :

- 01- « Enregistrer l'état de la machine » : Enregistre l'état actuel, qui sera rechargé la prochaine fois que vous le redémarrez ;
- 02- « Envoyer le signal d'arrêt », ou appuyez sur le bouton d'arrêt dans le machine : Éteignez la machine et enregistrez uniquement le travail effectué ;
- 03- « Allumez la machine » : Avertissement : Il ne sauvegarde rien ; tout le travail effectué sera rejeté !

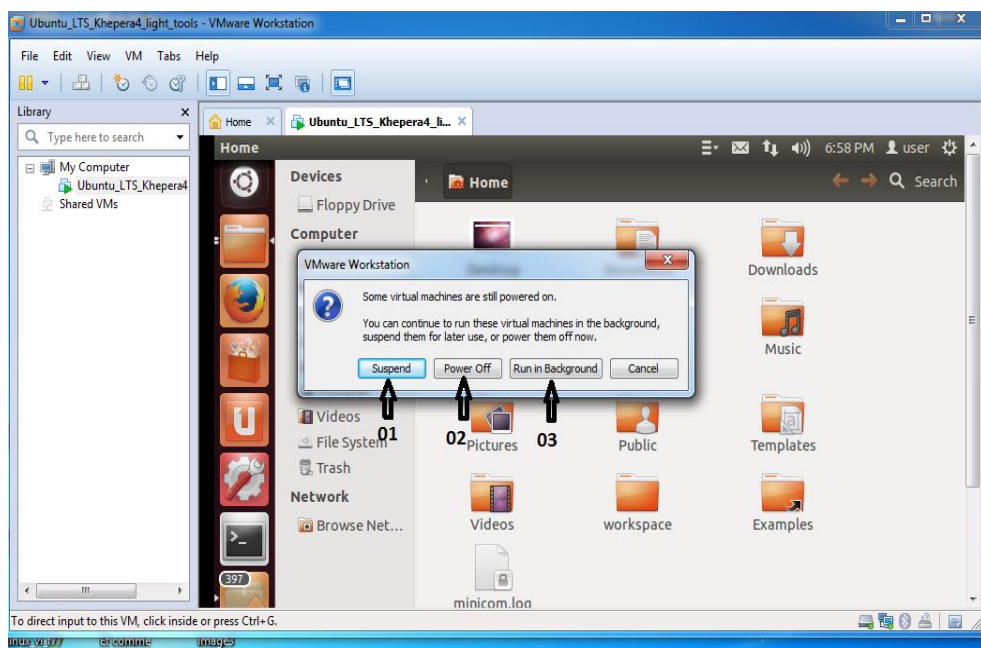


Figure III.9 : Arrêt de la machine virtuelle

III.4. Commande du robot mobile et communication via Windows

III.4.1. Commande du robot par un programme « putty »

Cette commande nécessite les étapes suivantes :

- Installer sur Windows les drivers **FTDI** ;
- Installer sur Windows le programme **putty** ;
- Brancher le câble USB à l'ordinateur, et le côté mini-USB au robot ;
- Allumer le robot ;
- Configurer le port série.

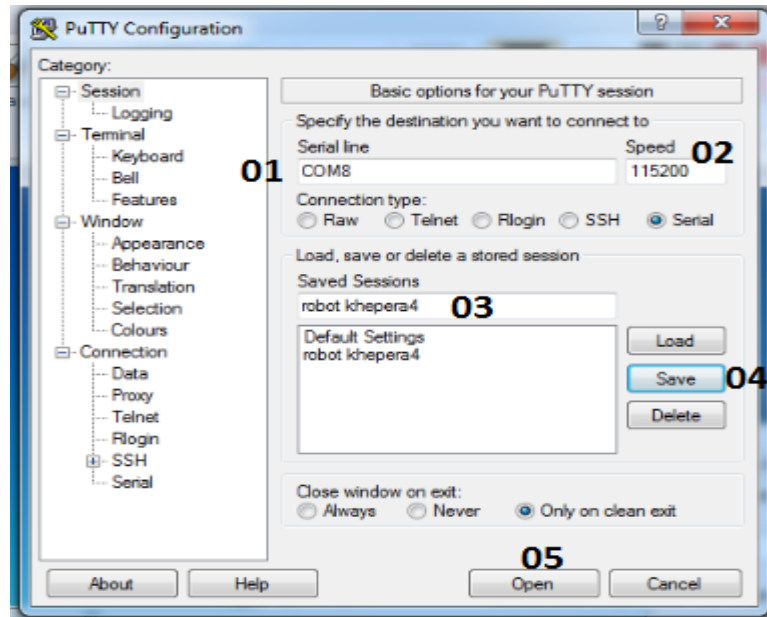


Figure III.10 : Configuration des paramètres du port COM

La figure III.10 montre les différents champs de configuration du programme « **putty** » dont la signification est la suivante :

- 01- Le nom du port COM ;
- 02- Sélection de la vitesse : 115200 ;
- 03- Nomination du projet ;
- 04- Enregistrement des paramètres avec le bouton : **Save** ;
- 05- Ouverture du terminal avec le bouton **Open** et taper **entrée**.

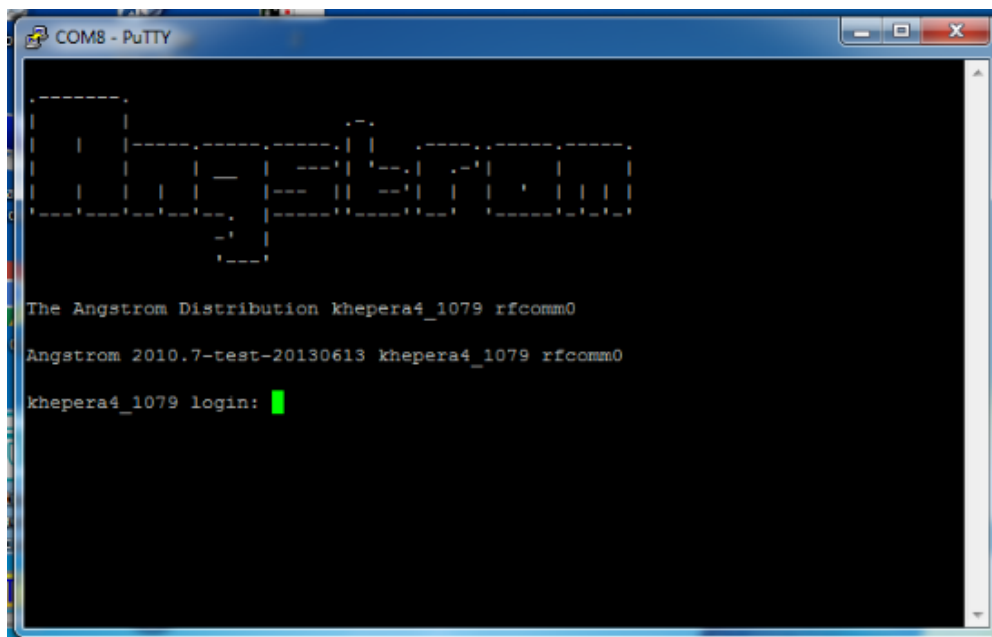
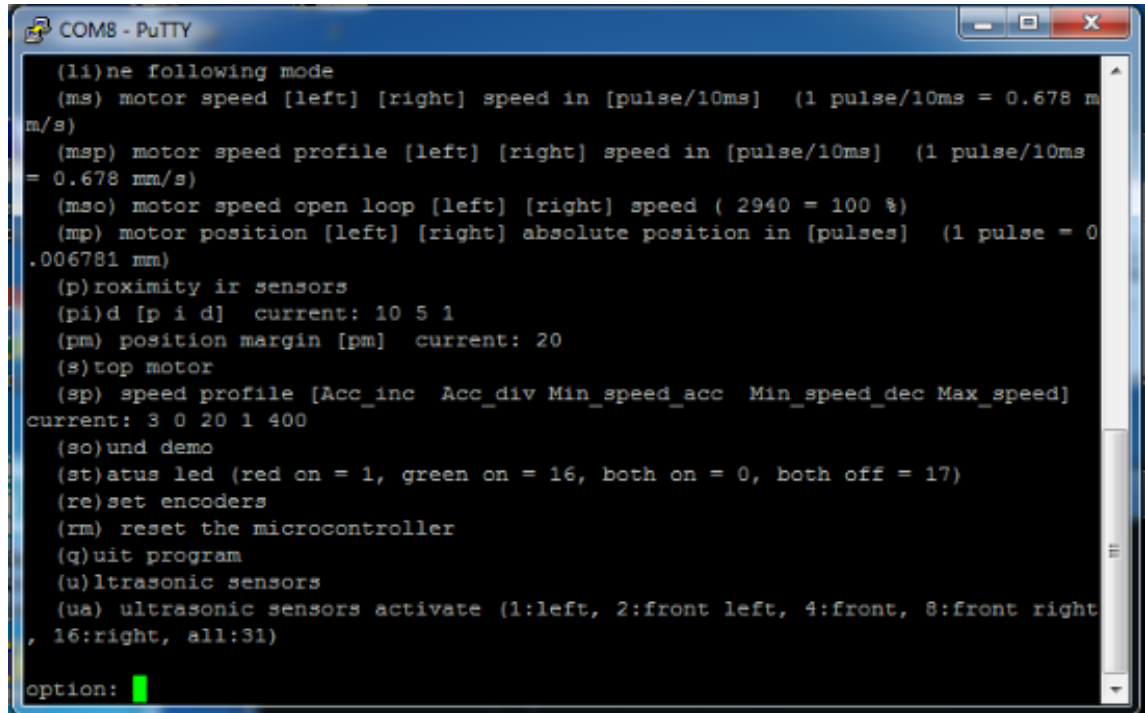


Figure III.11 : L'établissement de la connections entre le robot et Windows

La connexion se fait en entrant : le nom d'utilisateur : **root** et le mot de passe : **bour**. Ceci nous mène à :

- Nous sommes dans le répertoire de « **putty** », donc nous pouvons exécuter un programme en précisant son nom avec ./ : **./khepera4_test**



```

COM8 - PuTTY
(line) following mode
(ms) motor speed [left] [right] speed in [pulse/10ms] (1 pulse/10ms = 0.678 m
m/s)
(msp) motor speed profile [left] [right] speed in [pulse/10ms] (1 pulse/10ms
= 0.678 mm/s)
(mso) motor speed open loop [left] [right] speed ( 2940 = 100 %)
(mp) motor position [left] [right] absolute position in [pulses] (1 pulse = 0
.006781 mm)
(p)roximity ir sensors
(pi)d [p i d] current: 10 5 1
(pm) position margin [pm] current: 20
(s)top motor
(sp) speed profile [Acc_inc Acc_div Min_speed_acc Min_speed_dec Max_speed]
current: 3 0 20 1 400
(so)und demo
(st)atus led (red on = 1, green on = 16, both on = 0, both off = 17)
(re)set encoders
(rm) reset the microcontroller
(q)uit program
(u)ltrasonic sensors
(ua) ultrasonic sensors activate (1:left, 2:front left, 4:front, 8:front right
, 16:right, all:31)
option: █

```

Figure III.12 : La liste des commandes du robot sous « **putty** »

- Nous tapons les commandes pour contrôler le robot.

III.5. Application de pilotage du robot

Notre application est l'interface utilisateur, elle est implémentée sous Windows en utilisant **Visual Studio C#**, elle permet de communiquer via le port série et la clé **Bluetooth** pour envoyer les consignes d'utilisateur.

III.5.1. La découverte de la programmation orientée objet en C#

Dans cette partie, nous allons apprendre à programmer en **C#** avec des interfaces graphiques. Le but est de réaliser un terminal permettant l'envoi et la réception de messages sur le port série du PC. Ce terminal servira dans un premier temps de messagerie instantanée entre le PC et le robot relié par un câble série, puis il servira ensuite à piloter un robot mobile tout en observant son comportement interne.

Pour commencer, nous apprendrons à travailler avec les objets de base des interfaces graphiques. En particulier nous apprendrons à gérer les propriétés de ces objets et les événements qui leurs sont associés.

Pour avoir une prise en main de Visual Studio, nous allons créer une application Windows **Form** comme suit :

- Commençons par lancer **Visual Studio** et créer un projet Windows **Form**.

La fenêtre que nous obtenons est la suivante :

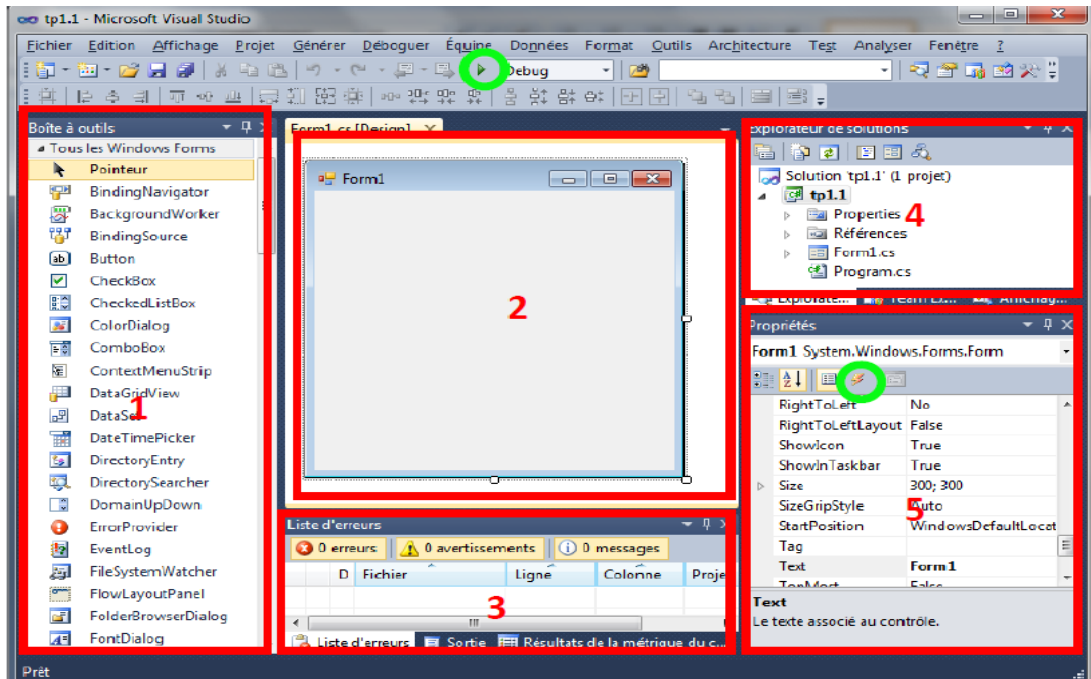


Figure III.13 : Les différentes zones du Visual Studio

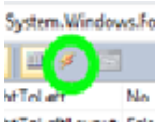
L'interface est composée de cinq zones principales qui sont réparties comme suit :

- 1- L'ensemble des composants disponibles que nous pouvons ajouter à notre projet. Il peut s'agir des composants visuels (un bouton) ;
- 2- La zone de conception : Où nous construisons l'interface graphique et éditons le code source ;
- 3- La zone de débogage : Tous les messages du compilateur sont indiqués ici ;
- 4- L'explorateur de solution : Où tous les fichiers (codes, configurations, ressources, etc.) qui composent votre solution sont indiqués ;
- 5- Liste des propriétés (nom, texte, taille, couleur, etc.) du composant sélectionné.

D'après la figure III.13 :



La flèche verte, au milieu en haut, permet de construire et d'exécuter le projet.



Le petit éclair dans la **zone 5** permet d'accéder à tous les événements (MouseDown, MouseMove, KeyPressed, Paint, etc.) gérés par le composant sélectionné et de définir de nouveaux comportements.

En plus des raccourcis habituels des applications Windows (Copier-Coller, etc.) Visual Studio propose de nombreux raccourcis dont les deux plus importants sont certainement : **Ctrl + Espace** pour la compilation automatique, et **Ctrl + .** pour l'ajout automatique des directives **using**.

- Modifions les propriétés de l'objet **form1** (la fenêtre principale), donnons-lui un nouveau nom, une nouvelle couleur de fond, et un nouveau curseur par défaut ;
- Ajoutons maintenant un bouton à notre fenêtre. Modifions son nom et appelons le **HelloButton**, modifions également le texte affiché dans le bouton ;
- Ajoutons un événement **Click** à notre bouton (faisons un double clic sur le bouton ou cherchons l'évènement dans la liste des évènements gérés par l'objet) ;

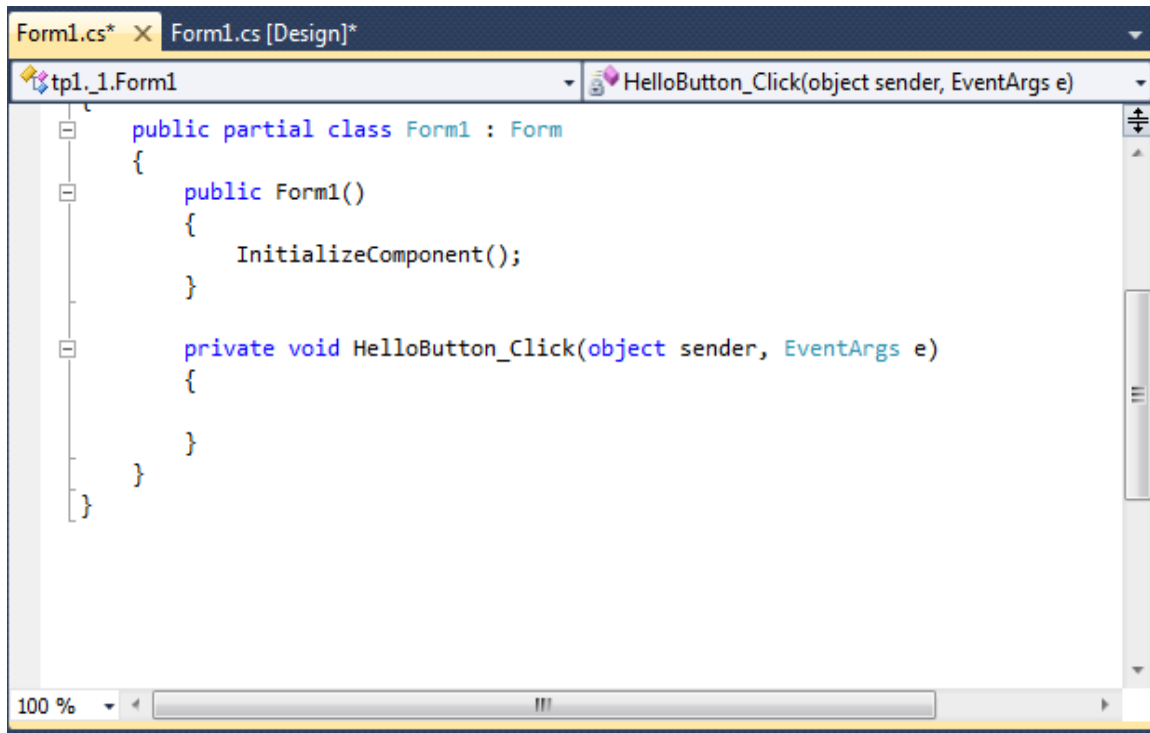


Figure III.14 : La liste des évènements

- Ajoutons du code dans la fonction `<nom_du_composant>_Click` pour afficher le message « **Hello World !** » (Utilisons la méthode statique **Show** de la classe **MessageBox**) ;
- Vérifions que tout fonctionne bien !

III.5.2. Pilotage et supervision du robot

Nous allons utiliser un système de messagerie pour piloter le robot et le superviser. La supervision permet de rendre observable des variables internes au robot tels que l'état des LEDs, les vitesses moteurs ou la position du robot etc.

Le pilotage et la supervision sont basés sur un ensemble de messages définis à l'avance et qui forment une bibliothèque devant être connue par le robot et de la plateforme de supervision.

L'opérateur communique et interagit (introduire les données, superviser le système, lancer le système, etc.) avec l'architecture de contrôle à travers cette interface qui est répartie comme suit :

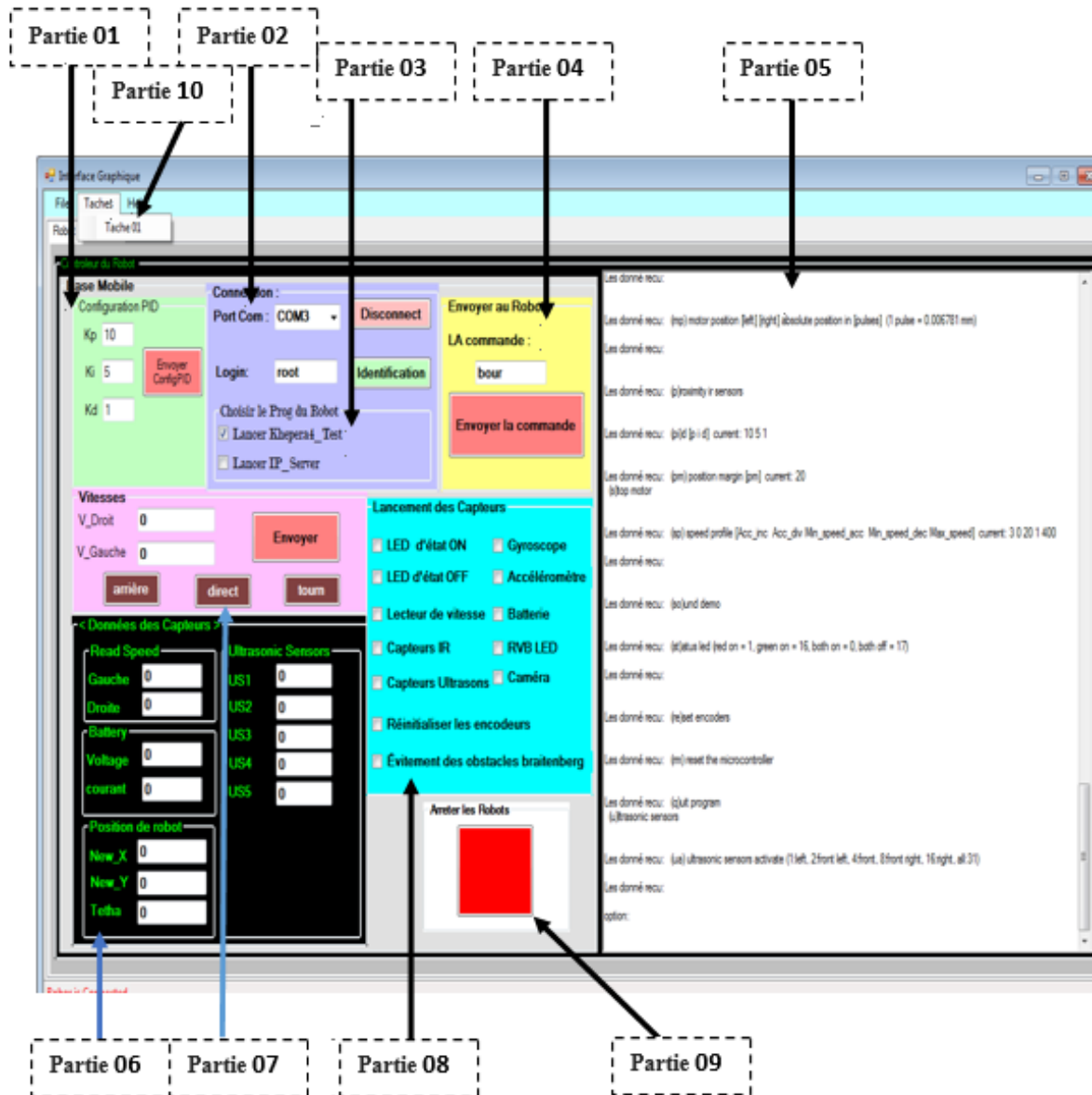


Figure III.15 : Les différentes parties de l'interface de pilotage

III.5.2.1. Description de l'interface IHMI

Comme cette interface est constituée de plusieurs parties, nous allons décrire chaque partie individuellement comme suit :

Partie 01 : Elle permet la configuration **PID** des moteurs avec les paramètres suivants :

$$Kp = 10 \quad Ki = 5 \quad Kd = 1$$

Elle contient aussi un Bouton voyant pour l'affichage.

Partie 02 : Contient une case de sélection du port **COM** et une deuxième case de définition du robot (login) avec deux boutons d'envoi des données ;

Partie 03 : Pour choisir le programme de fonctionnement du robot ;

Partie 04 : Contient une case d'écriture d'une commande ou de mot de passe du robot avec un bouton d'envoi pour l'exaction de la commande ;

Partie 05 : Un champ pour afficher les consignes du robot ;

Partie 06 : Les cases de lecture détectent les données des capteurs ;

Partie 07 : Pour commander le moteurs avec deux méthodes :

- 1- Envoyer des vitesses par les champs de la vitesse droite et la vitesse gouache ;
- 2- Trois Boutons pour la commande automatique des moteurs à une vitesse déterminée.

Partie 08 : Un champ pour le lancement des différent capteurs du robot ;

Partie 09 : Bouton pour l'arrêter du robot ;

Partie 10 : La tache de trajectoire.

III.5.2.2. La configuration du port série en C#

A présent, nous allons faire communiquer le robot avec le PC, relié par un câble USB-série. L'envoi et la réception des données se fera par l'intermédiaire d'un objet de type **Serial Port** en C#. Pour cela nous allons procéder comme suit :

- Nous ajoutons un objet de type **Serial Port** à notre projet, puis nous allons dans ses propriétés pour régler la vitesse de transmission à 115200 bauds et le nom du port à "COMX" où X est le numéro du port correspondant à l'adaptateur **USB/Série** que nous trouverons dans le gestionnaire de périphériques de Windows ;
- Dans la fonction d'envoi codée précédemment, nous rajoutons un envoi des données vers le port série en utilisant la méthode **Write Line** de l'objet **Serial Port**.

III.6. La commande du robot sans fil (Bluetooth)

III.6.1. Utilisation d'une clé Bluetooth

Il s'agit d'un adaptateur USB qui permet à l'ordinateur de communiquer en réseau avec le robot qui est muni d'un système Bluetooth. La Figure III.16 montre un adaptateur Bluetooth.



Figure III.16 : Adaptateur Bluetooth

III.6.2. Installation du programme Blue Soleil

Le **Blue Soleil** est un programme complet conçu pour nous donner accès à ces appareils Bluetooth que nous avons et que nous ne pouvons pas les utiliser en raison d'un problème de connexion avec notre PC. Grâce à cette application, nous serons capables de détecter et de connecter n'importe quel appareil Bluetooth autour de notre ordinateur (nous devons bien sûr avoir un adaptateur Bluetooth branché à notre PC). Le lien du téléchargement est le suivant :

<https://blusoleil.fr.uptodown.com/windows>

A l'aide du programme **Blue soleil**, nous avons pu détecter le Bluetooth du robot dont le nom est : **khepera4_1079**. La figure III.17 montre la procédure de cette détection.

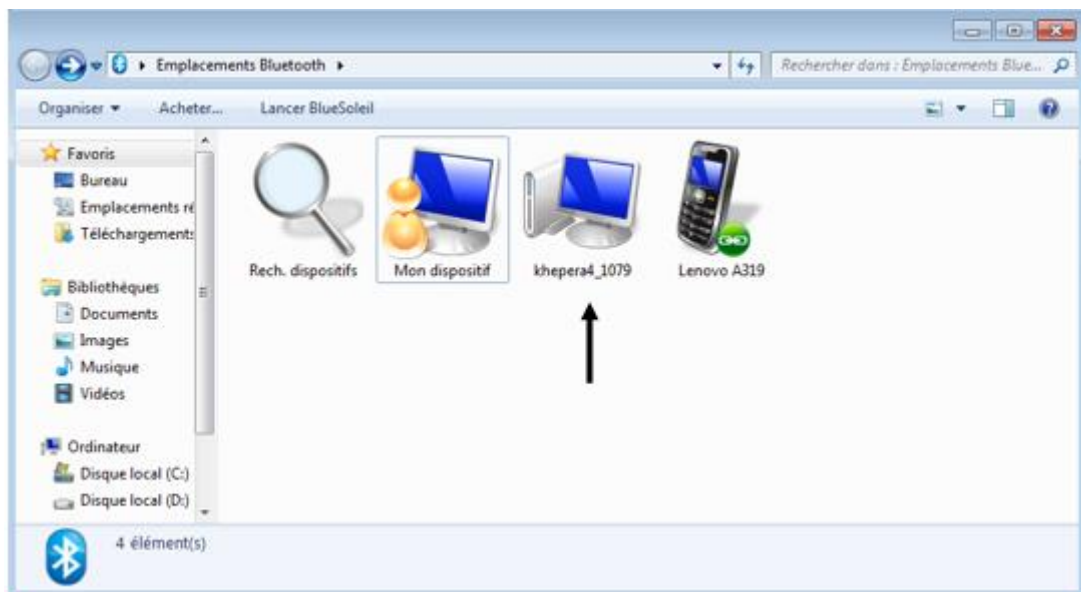


Figure III.17 : La détection du robot Khepera IV par le programme Blue Soleil

Pour commander le robot sans fil, nous utilisons le **port COM** de la clé Bluetooth dans l'interface d'utilisateur en C#.

III.7. Conclusion

Dans ce chapitre nous avons présenté les deux méthodes de contrôle d'un robot mobile :

La première technique est implémentée sous **linux** (elle se charge de communiquer directement via le port USB) en envoyant des commandes pour faire mouvoir le robot ou lire ses capteurs.

La deuxième méthode est implémentée sous Windows (programmes **putty**). Ainsi, nous avons créé une application représentant l'interface utilisateur, en utilisant **Visual Studio C#**. Elle nous a permis de communiquer et d'envoyer des consignes de l'utilisateur vers le robot avec une clé USB et sans fil.

Conclusion Générale

Conclusion générale

Finalement, le robot mobile Khepera IV est en état de marche. Ceci est réalisé au prix d'immense sacrifice, tant sur le plan intellectuel que matériels. Ainsi, pratiquement nous avons consacré notre temps aux tâches suivantes :

- Programmation du robot pour la commande des moteurs à courant continu utilisés, et pour l'exécution des commandes transmises par le **PC** ;
- Réalisation d'une interface graphique en **C#** permettant de commander le robot soit en mode de commande manuelle, soit en mode automatique.

En effet, ce projet de réalisation nous a été une expérience très fructueuse, bénéfique et enrichissante. Il nous a permis d'exploiter les différentes notions théoriques vues durant notre période de formation. Il nous a permis également de mettre le premier pas dans le monde pratique de la robotique. Ce projet nous a effectivement permis de communiquer sans fil avec le robot mobile Khepera IV et de proposer des méthode de commande.

Sur le plan software, ce projet nous a mis directement en contact avec des langages de programmation puissants tel que le langage **C#**, ainsi que différents logiciels spécialisés dans ce domaine, à savoir : **putty**, **Visual Studio** et **linux**.

En fin, nous espérons que notre travail va être amélioré au futur par d'autres étudiants et nous proposons comme perspectives ce qui suit :

- Création d'une trajectoire via la programmation des capteurs.
- Commander le robot par d'autre langage de programmation.
- La commande du robot à distance via Wi-Fi.
- La commande en 3D par une caméra.
- Utilisation des moyens d'extensions.

Bibliographie

Bibliographie

- [1] Trucs & Astuces sous Windows, gualbert adohoueto, page 428.
- [2] http://www.futura-sciences.com/definitions_robotique
- [3] Le français dans le monde, numéros 3 à 4 ; numéro 322 à 324, cle international 2002, l'université du Michigan, 24 Mars 2008, page 1.
- [4] http://www.futura-sciences.com/definitions_automate
- [5] <http://www.cinematheque.fr/objet/347.html>
- [6] <http://eduscol.education.fr/Robots>, page 4.
- [7] http://www.futura-sciences.com/definitions_informatique_intelligence_artificielle
- [8] <https://perso.univ-lyon2.fr/~mollon/EO-web/12-13/Virgo/definition%20et%20historique%20de%20la%20robotique/definition%20et%20historique%20de%20la%20robotique.html>
- [9] https://www.ensta-bretagne.fr/jaulin/mastersds_cours_robot_boimond.pdf, pages 3-4-5-6.
- [10] https://www.ensta-bretagne.fr/jaulin/mastersds_cours_robot_boimond.pdf, page 6.
- [11] https://www.ensta-bretagne.fr/jaulin/mastersds_cours_robot_boimond.pdf, page 7.
- [12] <http://jf.duchet.pagesperso-orange.fr/robotique/Pages/applications.htm>
- [13] Jean-Louis Deneubourg Ralph Beckers, Owen E. Holland. From local actions to global tasks : Stigmergy and collective robotics. In Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, pages 181–189, Cambridge, MA, USA, July 1994. 1
- [14] Jianping Chen, Yimin Yang, and Liang Wei. Research on the approach of task decomposition in soccer robot system. In Proceedings of the 1st International Conference on Digital Manufacturing and Automation (ICDMA'10), pages 284–289, Changsha, China, December 2010. 16
- [15] Arab Ali Cherif. Collective behavior for a micro-colony of robots. In Proceedings of the Second European Conference on Artificial Life (ECAL'93), pages 1–10, Brussels, Belgium, May 1993. 1
- [16] Intergiciels pour systèmes multi-robots : état de l'art Stefan-Gabriel Chitic, Julien Ponge, Olivier Simonin, Université de Lyon.
- [17] David Popenoe. Sociology (11th Edition). Prentice Hall, 1999. 7

- [18] Stuart J. Russell and Peter Norvig. *Artificial Intelligence : A Modern Approach* (2nd Edition). Prentice Hall, 2002. 7, 8, 15, 20, 21
- [19] D. P. Barnes and J. O. Gray. Behaviour synthesis for co-operant mobile robot control. In *Proceedings of the 1991 International Conference on Control (Control'91)*, pages 1135 – 1140, Edinburgh, UK, March 1991. 7
- [20] Edsger W. Dijkstra. Cooperating sequential processes. In F. Genuys, editor, *Programming Languages : NATO Advanced Study Institute*, pages 43–112. Academic Press, 1968. 9
- [21] Wei Ye, Richard T. Vaughan, Gaurav S. Sukhatme, John Heidemann, Deborah Estrin, and Maja J. Mataric. Evaluating control strategies for wireless- ´ networked robots using an integrated robot and network simulation. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA'01)*, pages 2941–2947, Seoul, Korea, May 2001. 9
- [22] Luca Iocchi, Daniele Nardi, and Massimiliano Salerno. Reactivity and deliberation : a survey on multi-robot systems. *Lecture Notes in Computer Science*, 2103 :9–32, 2001. 12
- [23] Eduardo Todt, Gustavo Raush, and Raúl Suárez. Analysis and classification of multiple robot coordination methods. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA'00)*, pages 3158–3163, San Francisco, CA, USA, April 2000. 12
- [24] Shin Kato, Sakae Nishiyama, and Jun'ichi Takeno. Coordinating mobile robots by applying traffic rules. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'92)*, pages 1535–1541, Raleigh, NC, USA, July 1992. 12, 58
- [25] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng. Cooperative mobile robotics : Antecedents and directions. *Autonomous Robots*, 4(1) :7–27, 1997. 2, 13
- [26] M. Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination : A survey and analysis. Technical Report CMU-RI-TR-05-13, Carnegie Mellon University, Pittsburgh, PA, USA, April 2005. 15
- [27] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991. 20, 33, 47, 60, 67, 72, 80.
- [28] Yi Guo and Lynne E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA'02)*, pages 2612–2619, Washington, DC, USA, May 2002. 21.

[29] <http://ftp.k-team.com/khepera/documentation/KheperaUserManual.pdf>, page 1.

[30] <https://www.generationrobots.com/fr/402241-robot-mobile-khepera-iv.html>.

[31] **Putty** : est un programme qui établit la connexion directe par liaison série. À l'origine disponible uniquement pour Windows, il est à présent porté sur diverses plates-formes linux

Annexes

Instructions de l'interface de pilotage du robot

Ces instructions sont divisées en différentes classes comme suit :

- **Instructions des bibliothèques**

```
using System ;  
using System.Collections.Generic ;  
using System.ComponentModel ;  
using System.Data ;  
using System.Drawing ;  
using System.Linq ;  
using System.Text ;  
using System.Windows.Forms ;  
using System.Net ;  
using System.Net.Sockets ;  
using System.Threading ;  
using System.IO.Ports ;  
using System.Drawing.Drawing2D ;  
using System.Collections ;  
using System.IO ;  
using System.Runtime.InteropServices.
```

- **Instruction de connexion par le port COM**



```
private void button5_Click(object sender, EventArgs e)  
{  
    Thread readThread = new Thread(Read) ;  
    if (button5.Text == "Disconnect")  
    {  
        //serialPort1.PortName = "COM1" ;  
        if (mySerialPort.IsOpen)  
        {  
            //treadport.Abort() ;  
            //readThread.Join() ;  
            mySerialPort.Close() ;  
            button5.Text = "Connect" ;  
            // serialPort1.Close() ;  
        }  
    }  
    else  
    {  
        button5.Text = "Disconnect" ;  
        mySerialPort = new SerialPort(comboBox1.Text) ;  
        mySerialPort.BaudRate = 115200 ;  
    }  
}
```

```

mySerialPort.Parity = Parity.None ;
//mySerialPort.StopBits = StopBits.One ;
mySerialPort.DataBits = 8 ;
mySerialPort.Handshake = Handshake.None ;
//mySerialPort.RtsEnable = true ;
mySerialPort.Open() ;
_continue = true ;
mySerialPort.DataReceived += new
SerialDataReceivedEventHandler(DataReceivedHandler) ;
richTextBox1.Text += "Port USB of the robot is Connected " + "\n" ;
toolStripStatusLabel2.Text = "Robor is Connected" ;
// toolStripStatusLabel1.Text = "ROBOT IS Connected" ;
}
}

```

- **Instruction d'identification du robot**

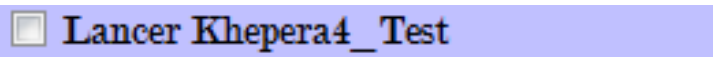


```

private void login_Clic(object sender, EventArgs e)
{
    mySerialPort.WriteLine("root") ;
    richTextBox1.Text += "Identification : Root" + "\n" ;
}

```

- **Instruction de lancement du programmeur du robot**

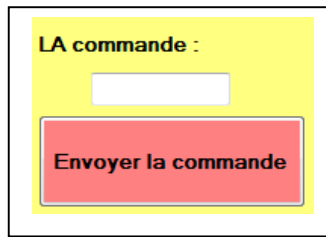


```

private void Lancer_CheckedChanged(object sender, EventArgs e)
{
    mySerialPort.WriteLine("./khepera4_test") ;
    richTextBox1.Text += "Lancement du programme : khepera4_test" + "\n" ;
}

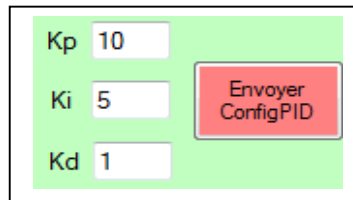
```

- **Instruction d'envoi des commandes**



```
private void psw6_Click(object sender, EventArgs e)
{
    mySerialPort.WriteLine(textBox14.Text);
    richTextBox1.Text += "Identification : Root" + "\n";
}
```

- **Instruction de la configuration PID**



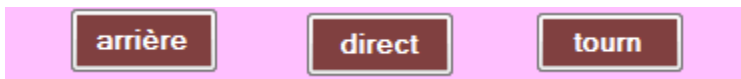
```
private void button4_Click(object sender, EventArgs e)
{
    mySerialPort.WriteLine("H," + textBox5.Text + "," + textBox4.Text + "," +
    textBox3.Text);
    richTextBox1.Text += "Send to the robot: " + "PID : " + textBox5.Text + " | " +
    textBox4.Text + " | " + textBox3.Text + "\n";
}
```

- **Instruction d'envoi des vitesses au robot**



```
public void button1_Click(object sender, EventArgs e)
{
    string bytetosent = "ms "+textBox1.Text+" "+textBox2.Text;
    mySerialPort.WriteLine(bytetosent);
    richTextBox1.Text += "Envoey to the robot : Set speed " + textBox1.Text + " " +
    textBox2.Text + "\n";
}
```

- **Instruction des directions des moteurs**



- **Tourne**

```
private void button9_Click(object sender, EventArgs e)
{
    sendUSBTToRobot("ms 200 -200");
    richTextBox1.Text += " tourn mouve\n";
}
```

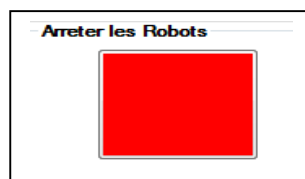
- **Arrière**

```
private void button7_Click(object sender, EventArgs e)
{
    sendUSBTToRobot("ms -200 -200");
    richTextBox1.Text += " arriere mouve\n";
}
```

- **Direct**

```
private void button8_Click_1(object sender, EventArgs e)
{
    sendUSBTToRobot("ms 200 200");
    richTextBox1.Text += " direct mouve\n";
}
```

- **Instruction d'arrêt du robot**



```
private void button3_Click(object sender, EventArgs e)
{
    mySerialPort.WriteLine("s");
    richTextBox1.Text += "Send to the robot: " + "Stop" + "\n";
}
```

- **Instruction de lancement des capteurs**



- **RVB LED marche**

```
private void checkBox3_CheckedChanged(object sender, EventArgs e)
{
    sendUSBTToRobot("st 16");
}
```

- **RVB LED arrêt**

```
private void checkBox20_CheckedChanged(object sender, EventArgs e)
{
    sendUSBTToRobot("st 17");
}
```

- **RVB LED**

```
private void checkBox10_CheckedChanged(object sender, EventArgs e)
{
    sendUSBTToRobot("1 lr lg lb");
}
```

- **Lecteur de vitesse**

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    mySerialPort.WriteLine("g");
    richTextBox1.Text += "Send to the robot: " + "E" + "\n";
}
```

- **Accéléromètre**

```
private void checkBox7_CheckedChanged(object sender, EventArgs e)
{
    mySerialPort.WriteLine("ag");
    richTextBox1.Text += "Read gyroscope values" + "\n";
}
```

- **Réinitialiser les encodeurs**

```
private void checkBox8_CheckedChanged(object sender, EventArgs e)
{
    mySerialPort.WriteLine("I");
    richTextBox1.Text += "Reset the values of the motors encoders" + "\n";
}
```

- **Démonstration d'appareil photo**

```
private void checkBox9_1_CheckedChanged(object sender, EventArgs e)
{
    mySerialPort.WriteLine("c");
    richTextBox1.Text += "Camera demo" + "\n";
}
```

- **Gyroscope**

```
private void checkBox5_CheckedChanged(object sender, EventArgs e)
{
    mySerialPort.WriteLine("ag");
    richTextBox1.Text += "Read gyroscope values" + "\n";
}
```

- **Capteurs ultrasons**

```
private void checkBox2_CheckedChanged(object sender, EventArgs e)
{
    mySerialPort.WriteLine("u"); //u
    richTextBox1.Text += "Send to the robot: " + "Get US values" + "\n";
}
```

- **Capteurs infrarouges**

```
private void checkBox4_CheckedChanged(object sender, EventArgs e)
{
    sendUSBTToRobot("p");
}
```

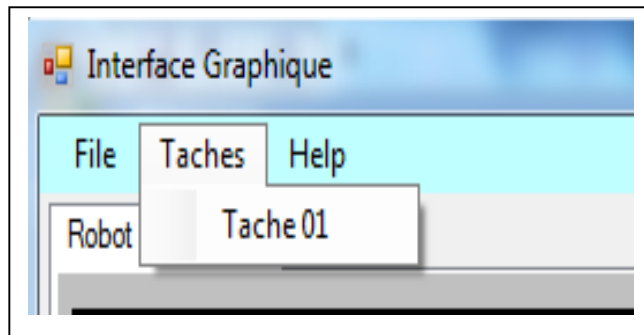
- **État de batterie**

```
private void checkBox6_CheckedChanged(object sender, EventArgs e)
{
    mySerialPort.WriteLine("b");
    richTextBox1.Text += "Send to the robot: " + "Read robot Speed" + "\n";
}
```

- **Évitement des obstacles braitenberg**

```
private void checkBox9_CheckedChanged_1(object sender, EventArgs e)
{
    sendUSBTToRobot("br");
}
```

- **Trajectoire avec des times**



```
private void tache01ToolStripMenuItem_Click(object sender, EventArgs e)
{
    //DisplayReceivedMessag(" Go 100 100 \n");
    // status = 0 ;
    // timer1.Start();

    sendUSBTToRobot("ms 300 300"); Thread.Sleep(2500);
    sendUSBTToRobot("ms 120 -120"); Thread.Sleep(1000);
    sendUSBTToRobot("ms 200 200"); Thread.Sleep(2000);
    sendUSBTToRobot("ms -120 120"); Thread.Sleep(1000);
    sendUSBTToRobot("ms 200 200"); Thread.Sleep(2000);
    sendUSBTToRobot("ms -120 120"); Thread.Sleep(1000);
}
```

```
sendUSBTToRobot("ms 200 200"); Thread.Sleep(2000);  
sendUSBTToRobot("ms 120 -120"); Thread.Sleep(1000);  
sendUSBTToRobot("ms 350 350"); Thread.Sleep(2000);  
sendUSBTToRobot("ms -120 120"); Thread.Sleep(1000);  
sendUSBTToRobot("ms 200 200"); Thread.Sleep(1500);  
sendUSBTToRobot("ms -120 120"); Thread.Sleep(1000);  
sendUSBTToRobot("ms 400 400"); Thread.Sleep(2000);  
sendUSBTToRobot("ms -120 120"); Thread.Sleep(1000);  
sendUSBTToRobot("ms 200 200"); Thread.Sleep(2000);  
sendUSBTToRobot("ms 120 -120"); Thread.Sleep(1000);  
sendUSBTToRobot("ms 400 400"); Thread.Sleep(3000);  
sendUSBTToRobot("s");  
  
}
```