

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITÉ IBN-KHALDOUN DE TIARET
FACULTÉ DES SCIENCES APPLIQUÉES
DÉPARTEMENT DE GÉNIE ELECTRIQUE



MÉMOIRE DE FIN D'ÉTUDES

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologies

Filière : AUTOMATIQUE

Spécialité : Automatique & Informatique industriels

THÈME

Conception d'un Manipulateur Mobile Connecté Application pour une tâche de saisie-déposer

Préparé par : KASDI Leila

Devant le Jury :

Nom et Prénoms	Grade	Qualité
H. BENABID	MAA	Président
L. BESSOLTANE	MAA	Examineur1
A. TAHRI	MCB	Examineur2
A. ADDA BENATTIA	MCB	Encadreur
H. BOUMEDIENE	MCB	Co-Encadreur

Promotion 2024/2025



Remerciements

Louange à **ALLAH** qui m'a accordé la force et la patience pour mener à bien ce travail.

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui m'ont soutenu(e) tout long de ce parcours.

Mes remerciements vont tout particulièrement à mon encadrant **M. ADDA BENATTIA Abderrahmane**, pour ses conseils précieux, sa disponibilité et son soutien constant.

Je remercie spécial à ma famille, pour leur amour inconditionnel, leur patience et leur prières, sans lesquels je n'aurais pu avancer.

À mes amis et collègues, pour leur encouragement et leur présence confortable, je vous adresse toute ma reconnaissance.

Que ce travail soit un hommage à tous ceux qui ont cru en moi et m'ont accompagné sur ce chemin.

DÉDICACE

Je dédie ce travail

À ma chère mère, pour son amour infini et ses sacrifices.

À mon père, pour sa confiance et son soutien précieux.

À mes sœurs, pour leur tendresse, leur présence et leurs encouragements constants.

À mes amies fidèles, pour leur soutien et leur belle énergie.

À mes enseignants et professeurs, pour leur accompagnement et leurs conseils éclairés.



SOMMAIRE

Remerciements.....	2
<i>DÉDICACE</i>	3
SOMMAIRE.....	4
LISTE DES FIGURES.....	7
LISTE DES TABLEAUX.....	9
LISTE DES ACRONYMES.....	9
Introduction générale.....	10
Chapitre 1 : Manipulateur Mobile dans l'Industrie.....	11
1. Introduction.....	11
2. MMs : Historique et travaux connexes.....	11
2.1. Vers une Usine plus Intelligente.....	13
2.2. Énoncé du problème.....	13
2.3. Motivation et contribution.....	13
2.4. Méthodologie.....	13
3. Intégration et mise en œuvre des MM dans les environnements industriels.....	14
3.1. La recherche à l'industrie.....	14
3.2. Application industrielle.....	15
4. Industrie 4.0 : exigence pour le développement de MM adaptables.....	21
4.1. Connectivité et interopérabilité accrues.....	21
4.2. Collecte et analyse des données.....	22
4.3. Flexibilité améliorée.....	22
4.3.1. Navigation et planification d'itinéraires pour robots mobiles.....	22
4.3.2. Programmation flexible pour systèmes de production hybrides.....	23
4.4. Virtualization.....	23
4.5. Cyber-sécurité.....	24
5. Discussion.....	24

6. Conclusion.....	26
Chapitre 2 : Modélisation et conception, analyse fonctionnelle et technique.....	28
1. Introduction.....	28
2. Architecture logicielle du manipulateur mobile.....	28
3. Description du bras manipulateur mobile.....	28
4. Outils de conception de travail.....	28
4.1. Système de développement Node-Red.....	29
4.2. Python.....	30
4.3. Bitwise SSH client.....	30
4.4. Définition de Advanced IP Scanner.....	30
4.5. Présentation du Raspberry Pi.....	31
4.5.1. Caractéristiques du Raspberry Pi utilisé.....	31
4.5.2. Alimentation.....	32
4.5.3. Choix du Raspberry Pi 3.....	32
4.5.4. Connecteurs et ports disponibles sur Raspberry Pi.....	33
4.6. Présentation du moteur NEMA 17(JK 42HS34 1334AC).....	34
4.6.1. Caractéristiques principales du moteur.....	34
4.6.2. Choix du moteur.....	35
4.7. Driver de moteur à double pont en H L298N.....	35
4.7.1. Caractéristique du driver L298N.....	36
4.7.2. Choix du composant driver de moteur L298N.....	36
4.7.3. Alimentation du module L298N.....	37
4.8. Servomoteur MG996R.....	38
4.8.1. Caractéristiques.....	38
4.8.2. Choix du servomoteur.....	38
4.9. Module PCA9685-Contrôleur PWM / I2C.....	39
4.9.1. Caractéristique.....	39
4.9.2. Choix de Module PCA9685-Contrôleur PWM/I2C.....	39

4.10. Module de détection aux Ultrasons HC-SR04.....	39
4.11. Électroaimant.....	41
4.12. Châssis mobile.....	41
4.12.1. Fonction principale.....	42
4.12.2. Choix de Châssis mobile.....	42
5. Conclusion.....	42
Chapitre 03 : Simulation, réalisation et intégration du manipulateur mobile.....	43
1. Introduction.....	43
2.1. Modèle géométrique direct.....	43
2.2. Modèle géométrique inverse :.....	45
3. Programmation avec Node-Red.....	47
3.1. Interface de programmation de Châssis mobile.....	48
3.3. Interface de programmation de MGD du bras manipulateur.....	50
4. Protocole I2C : Communication entre dispositifs électriques.....	52
5. Protocole PWM : Modulation de largeur d’impulsion.....	52
6. Connexion du module PCA9685 avec Raspberry Pi via I2C.....	53
7. Moteur pas à pas pour le déplacement du châssis mobile.....	54
7.1. Connexions établies.....	54
7.2. Fonctionnement.....	54
8. Intégration du capteur HC-SR04 avec Raspberry Pi.....	55
8.1. Détection de la position de la base mobile par ultrasons.....	55
8.2. Interface Dashboard – Contrôle avancé du châssis mobile et du bras manipulateur.....	57
8.3. Déplacement du châssis mobile.....	58
9. Explication de désigne de bras manipulateur.....	59
Chapitre 04 : Résultats obtenus et Discussions.....	60
1. Introduction.....	60
2. Résultats obtenus concernant le bras du manipulateur.....	60

3.	Détails techniques sur le fonctionnement du bras.....	61
4.	Description détaillée du châssis mobile.....	61
5.	Contrôle du système à l'aide d'une interface Dashboard connecté.....	62
6.	Gestion synchronisée du bras et de sa base mobile.....	62
7.	Gestion de la connectivité Réseau et contrôle à distance.....	63
7.1.	Connexion Wi-fi du Raspberry Pi.....	63
7.2.	Utilisation de Advanced IP Scanner.....	63
7.3.	Accès à distance avec Bitwise SSh client.....	63
7.4.	Contrôle du Bras Manipulateur via le Dashboard.....	64
8.	Connexion et complémentarité entre les éléments du système.....	64
9.	Perspectives d'évolution et applications potentielles.....	64
10.	Conclusion.....	65
	Conclusion général.....	66
	Références bibliographiques.....	67

LISTE DES FIGURES

	Figure 1 : Chronologies des MM existants de 1984 à 2010[3].....	12
	Figure 2 Chronologie des différents projets qui ont contribué au développement de MM.....	14
	Figure 3 . Modèle d'infrastructure adaptée à l'intégration des MM [3].....	15
	Figure 4 . Adéquation de l'utilisation des MM pour chaque classe d'opération industrielle.....	16
16	Figure 5 . Petits assistants de la 1 ^{ère} à la 7 ^{ème} génération (de gauche à droite) adoptés.	
	Figure 6 . Little Helper 3 de l'Univ-Aalborg (à gauche), omniRob de Kuka (à droite).....	17
17	Figure 7 . Système de contrôle intégrant les MM dans les systèmes de production [1].	
	Figure 8 . MOMA d'OMRON adopté de [1].....	18
	Figure 9 . MM pour transporter de grandes pièces avec la coopération humaine [1].....	18
	Figure 10 . Projet SAPHARI, applications industrielles requises chez KUKA (à gauche) et exigence d'assemblage d'avions d'Airbus (à droite) [1].....	19

Figure 11 . Prototype ColRobt pour les industries automobiles [1].	19
Figure 12 . Mesurer la réflectivité du miroir - central énergétique Valle par MainBOT [1]	20
Figure 13 . MM conçu dans le projet THOMAS [1].	23
Figure 14 . Lien sémantique entre les principes clés à prendre en compte pour leur intégration dans les systèmes de fabrication basés sur l'industrie 4.0.	25
Figure 15 . Niveau de maturité technologique	26
Figure 16 . Schéma bloc de fonctionnement du bras manipulateur mobile.	28
Figure 17 . Logo de Node-RED.	29
Figure 18 . Module et Interface de Dashboard	29
Figure 19 . Logo de python	30
Figure 20 . Logo de Bitwise SSH client	30
Figure 21 . Logo de Advanced IP Scanner	31
Figure 22 . Raspberry Pi 3	31
Figure 23 . Emplacement de différents composants de Raspberry Pi 3 modèle B	32
Figure 24 . Ports GPIO	33
Figure 25 . NEMA 17(JK 42HS34 1334AC)	34
Figure 26 . Driver de moteur à double pont en H L298N	36
Figure 27 . Schéma double pont H driver L298N.	37
Figure 28 . Servomoteur MG996R	38
Figure 29 . Module PCA9685-Contrôleur PWM/I2C	39
Figure 30 . HC-SR04	39
Figure 31 . Fonctionnement du capteur HC-SR04	40
Figure 32 . Électroaimant	41
Figure 33 . Châssis mobile	42
Figure 34 . Position simplifiée du robot pour la modélisation	43
Figure 35 . Interface graphique de Node-Red	47
Figure 36 . Interface de programmation de châssis mobile	48
Figure 37 . L'interface de programmation de position de bras manipulateur	49
Figure 38 . Les interfaces de programmation MGD	50
Figure 39 . Les interfaces de programmation MGI	51
Figure 40 . Connexion du module PCA9685 avec Raspberry Pi via I2C	53
Figure 41 . Connexion entre moteur NEMA JK 17 et Raspberry Pi &driver L298N	55
Figure 42 . Capteur Ultrasons de mesure de déplacement	57
Figure 43 . Interface de Dashboard dans la position0	57

Figure 44 . Interface de Dashboard dans la position de saisie.....	58
Figure 45 . Interface de Dashboard dans la position de déposer.....	58
Figure 46 . Bras manipulateur + châssis mobile.....	59
Figure 47 . Éléments du bras manipulateur	60
Figure 48 . Différentes positions de bras.....	61
Figure 49 . Châssis mobile sur rails.....	62
Figure 50 . Dashboard de commande.....	64

LISTE DES TABLEAUX

Tableau 1 Classification des application industriel MM.	15
Tableau 2 . Résumé des applications industrielles des MM [1].	20
Tableau 3 . Paramètres de Denavit-Hartenberg.....	43
Tableau 4 . Connexion PCA9685 avec Raspberry Pi via I2C.....	53

LISTE DES ACRONYMES

- AI : Intelligence Artificiel
- AMR: Robot mobile autonome
- AR: Réalité augmentée
- CIM: Fabrication intégré par ordinateur
- CMfg: Fabrication en cloud
- Cobots: Robot collaborative
- CPS: Système Cyber -Physique
- DOF: Degrée de liberté
- DT: Jumeaux numériques
- IIOT: Internet Industriel des objets
- IOT: Internet des objets
- LH(s): Petit Assistant
- MMs: Manipulateur mobile(s)
- ROS: Système d'exploitation des robot
- TRL: Niveau de Maturité Technologique
- UAV(s): Unmanned Aerial Vehicle
- VR : Virtual Reality



Introduction générale

Ces dernières années, le secteur industriel a connu une transformation importante avec l'introduction du concept de l'industrie 4.0, qui vise à rendre les systèmes de production plus intelligents, connectés et autonomes.

Dans ce contexte, les manipulateurs mobiles prennent une place de plus en plus importante grâce à leur capacité à se déplacer dans l'environnement industriel, à accomplir des tâches de manière autonome, et à être contrôlés à distance. Le présent travail s'inscrit dans cette dynamique et propose la conception et la réalisation d'un manipulateur mobile connecté, capable de se déplacer sur une trajectoire définie (linéaire) et de transmettre des informations en temps réel.

Le programme de contrôle est implémenté sur un mini-ordinateur de type Raspberry Pi 3B+, et l'interface graphique de manipulation est réalisée avec l'outil d'automatisation de flux Node-Red.

Pour atteindre cet objectif, il a été nécessaire d'adopter une approche méthodique, en commençant par une étude théorique approfondie des manipulateurs mobiles et leur rôle dans les systèmes industriels intelligents. Nous avons ensuite identifié les besoins techniques du système à concevoir, choisi avec soin les composants matériels adéquats, et planifié les étapes de réalisation de manière progressive afin d'assurer la fiabilité et la cohérence du projet dans son ensemble.

La suite du travail présente donc les notions de base autour des manipulateurs mobiles, leur rôle dans l'industrie moderne, ainsi que les problématiques rencontrées. Ensuite, nous avons justifié le choix des composants matériels utilisés dans notre système, tels que le Raspberry Pi, le moteur pas à pas, les servomoteurs, le système d'assemblage physique et la programmation du manipulateur mobile.

Enfin, nous cherchons à montrer comment une solution simple, économique et connectée peut s'intégrer efficacement dans un environnement industriel intelligent.

Chapitre 1 : Manipulateur Mobile dans l'Industrie

1. Introduction

Aujourd'hui, la production industrielle repose fortement sur une combinaison d'intelligence artificielle (IA) et de systèmes avancés incluant des capteurs intelligents et des technologies modernes. Ces outils aident les entreprises à répondre aux demandes croissantes des consommateurs et à développer continuellement de nouvelles technologies. Le domaine de la robotique a connu de grands changements au fil du temps, les robots étant utilisés pour réduire les tâches répétitives et ennuyeuses accomplies par les humains dans les usines. Avec le temps, des systèmes robotiques avancés, tels que les bras robotisés, ont été développés pour être utilisés dans des processus de fabrication précis et complexes. Ces robots combinent la capacité de se déplacer et de manipuler des objets, ce qui permet leur utilisation dans de nombreux domaines comme l'assemblage industriel et même la chirurgie. Dans ce chapitre, nous aborderons comment ces robots ont évolué et comment ils ont contribué à améliorer la production industrielle en augmentant la précision et la flexibilité du travail.

2. MMs : Historique et travaux connexes

Fondamentalement, les manipulateurs mobiles sont composés d'un ou plusieurs bras robotiques qui leur permettent de se déplacer en fonction de l'environnement pour accomplir une tâche spécifique, en utilisant l'outil sélectionné pour l'effecteur final du bras. Les MM peuvent être conçus avec des fonctionnalités spéciales en adaptant leur taille, leur forme, leur charge utile et leur mobilité. Les plates-formes mobiles peuvent être équipées de roues ou de pieds [1].

S.Bogh et al.[3]ont fourni un aperçu des MM existants depuis le premier prototype en 1984(MORO) jusqu'en 2010. Cependant, les origines de la création des MM remontent encore plus loin, au 'Stanford Research Institut', qui a développé SHAKEY de 1966 à 1972.



Figure 1 : Chronologies des MM existants de 1984 à 2010[3].

L'évolution des manipulateurs mobiles a traversé plusieurs étapes importantes. Leur utilisation dans les lignes de production a permis d'améliorer la navigation, le contrôle et la perception, notamment dans des compétitions comme '*RoboCup*'. En intégrant des plateformes mobiles avec des bras robotiques, les MM ont gagné en flexibilité, avec plus de neuf degrés de liberté, ce qui a permis de les utiliser dans des secteurs comme les services.

Cependant, l'élargissement de leur espace de travail a posé des défis en termes de contrôle et de conception. Une approche collaborative a vu le jour, avec les MM utilisés comme outils collaboratifs dans l'industrie. Cela a mené à la création du 'Manipulateur

Mobile Industriel Collaboratif', qui combine du matériel complexe et des logiciels intelligents pour contrôler et planifier les tâches.

Une autre tendance est l'intégration des drones (UAV) avec les MM, ouvrant des perspectives pour des opérations de montage et de logistique plus dynamiques. Bien que prometteur, ce domaine reste encore en développement et nécessite plus de recherches.

Malgré les progrès, certains aspects des MM, comme l'optimisation de la collaboration et du contrôle, n'ont pas encore été suffisamment étudiés.

2.1. Vers une Usine plus Intelligente

La révolution industrielle vise à créer des usines intelligentes, efficaces adaptatives et flexibles. En intégrant des technologies avancées, l'intelligence artificielle et les capteurs, elle améliore la sécurité et l'efficacité de la production.

La capacité de l'appareil à communiquer est l'une des conditions essentielles pour comprendre et analyser l'environnement, et cela peut être amélioré grâce à l'intelligence artificielle.

2.2. Énoncé du problème

Avec l'évolution et l'adaptation rapides de l'industrie l'accent est de plus en plus mis sur la flexibilité opérationnelle et l'adaptabilité dans les environnements de fabrication. Les MM sont devenus des atouts essentiels pour répondre à cette demande, offrant des capacités uniques qui transcendent la robotique traditionnelle. Cependant, bien que les MM possèdent intrinsèquement des caractéristiques adaptées au paradigme de l'industrie, la question de leur intégration opérationnelle et structurelle dans un système de production numérisé demeure un défi majeur nécessitant des adaptations technologiques, organisationnelles et humaines approfondies, pour exploiter pleinement le potentiel évolué. Cette étude se penche sur les avancées nécessaires aux MM, grandissant leur déploiement transparent et efficace dans le paysage de l'industrie.

2.3. Motivation et contribution

L'intégration des robots mobiles manipulant dans le cadre de l'industrie, caractérisée par des usines plus intelligentes et interconnectées. L'objectif principal est de fournir un aperçu complet des MM, en explorant leur évolution dans l'industrie et en proposant une feuille de route pour leur développement futur. L'étude vise à analyser en profondeur les connaissances actuelles sur ce sujet et à identifier les directions possibles pour leur utilisation et leur amélioration dans les environnements de production moderne.

2.4. Méthodologie

Pour examiner systématiquement les évolutions des MM dans le contexte de l'industrie 4.0, notre méthodologie a été construite en plusieurs étapes :

- Recherche préliminaire.
- Revue de la littérature séparée.
- Analyse des applications MM dans l'industrie
- Synthèse des recommandations de l'industrie 4.0 pour les MM.
- Exploration des MM.
- Les MM dans le contexte de l'industrie 4.0.

3. Intégration et mise en œuvre des MM dans les environnements industriels

La section décrit l'évolution des MM, de la recherche à l'industrie, avec un focus sur leurs applications Industrielles réelles. Les projets récents, présentés dans la figure 1 et 2, montrent les progrès depuis 2010 et sont analysés dans les sections suivantes.

3.1. La recherche à l'industrie

La recherche scientifique et les besoins industriels sont désormais liés. Les besoins des entreprises peuvent inspirer des projets de recherche, tandis que les progrès scientifiques offrent des solutions innovantes. Cependant, les entreprises demeurent prudentes vis-à-vis des nouvelles solutions en raison des coûts associés, et l'adaptation des systèmes de production existants s'avère essentielle pour assurer leur adoption dans l'industrie.

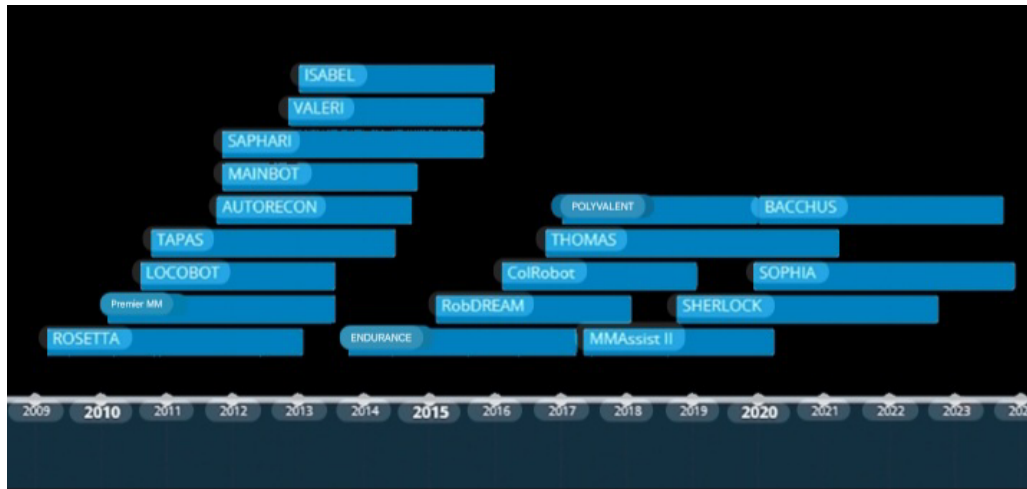


Figure 2 Chronologie des différents projets qui ont contribué au développement de MM.

Le développement des manipulateurs mobiles (MM) dans les systèmes de fabrication. Selon S.Bogh et al.[3], une des clés pour réussir cette intégration est de trouver un 'langage commun' entre le monde académique et l'industrie. Cela signifie qu'il est essentiel de bien communiquer entre ces deux domaines pour mieux travailler ensemble.

Ils ont proposé un modèle appelé 'TPMT' (Technologie Poussée par la Technologie), qui permet d'identifier et d'analyser les opportunités pour utiliser les nouvelles technologie, comme les MM. Une fois qu'on a trouvé les bonnes applications, il est important de définir des modules et des compétences spécifiques pour les MM, car un système modulaire rend les robots plus flexibles.

Au lieu de faire une grande démonstration, ils recommandent de faire plusieurs petites démonstrations pour mieux montrer les capacités des technologies. Cela aide les gens à comprendre plus facilement comment ces technologies peuvent être utiles dans différent contextes.

3.2. Application industrielle

Les manipulateurs mobiles sont utilisés dans des applications industrielles répétitives, dangereuses ou salissantes, similaires aux robots traditionnels, mais avec plus de flexibilité, une plus grande surface de travail et une meilleure efficacité. L'étude de S.Bogh et al, explore leur rôle en tant qu'assistants dans l'industrie.

Tableau 1 Classification des application industriel MM.

Application des manipulateurs mobiles		
Tâches d'assistance	Tâches logistiques	Tâches de service
Surveillance des machines	Transport	Maintenance, réparation et révision (MIRO)
Assemblage/pré-assemblage	Alimentation en pièces (multiple)	Nettoyage
Inspection	Alimentation en pièces (unique)	
Exécution de processus		

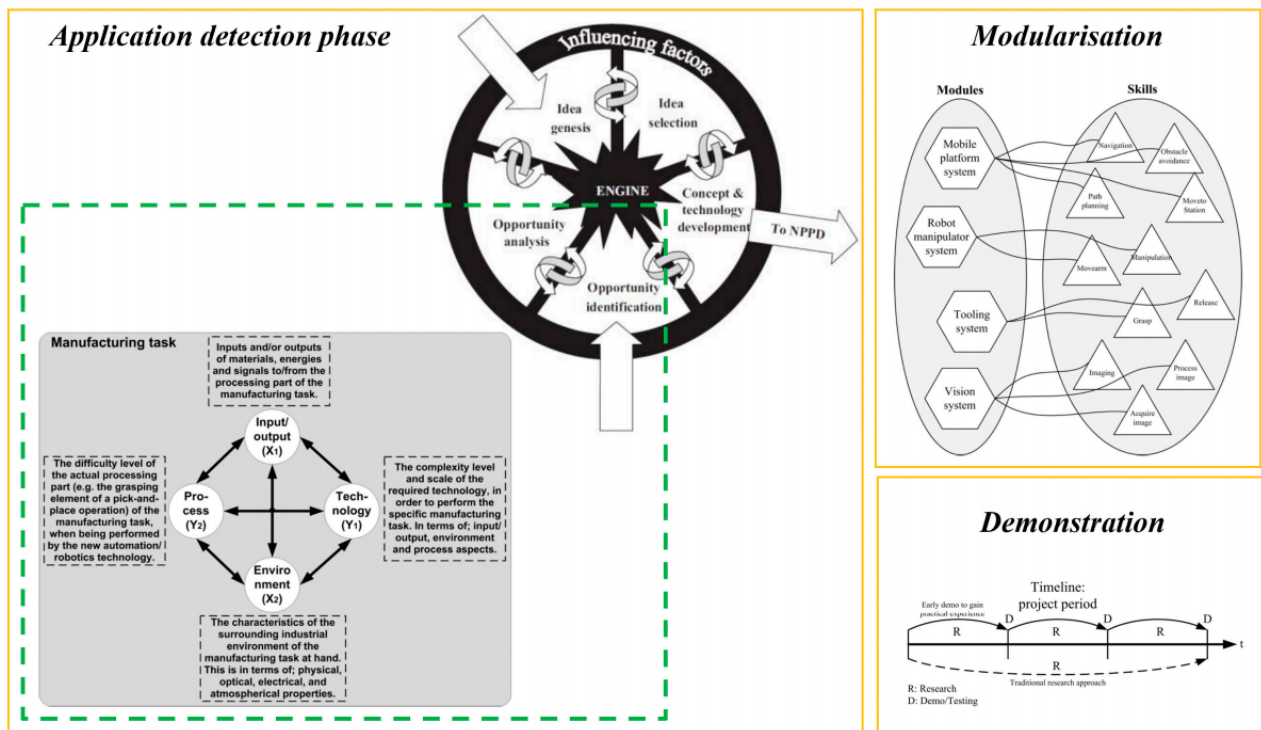


Figure 3. Modèle d'infrastructure adaptée à l'intégration des MM [3]

L'analyse de 556 tâches dans cinq usines montre que les manipulateurs mobiles sont particulièrement adaptés à la logistique. Le transport et l'alimentation des pièces sont les domaines les plus prêts pour l'adoption immédiate de cette technologie.

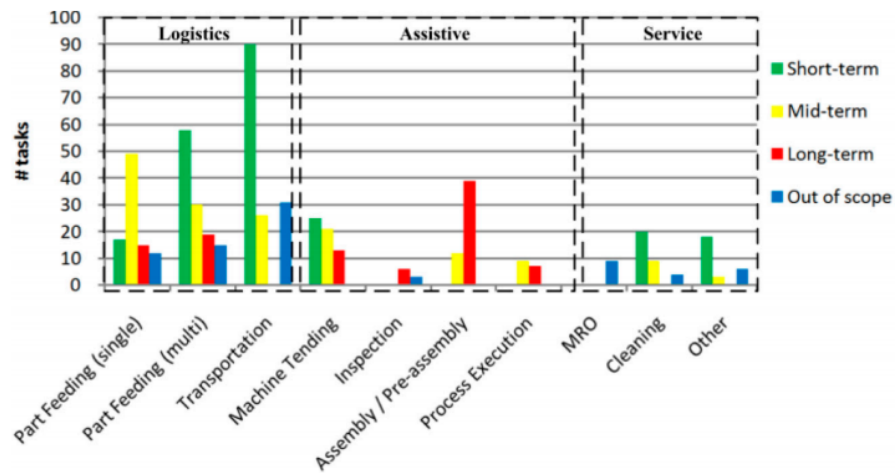


Figure 4. Adéquation de l'utilisation des MM pour chaque classe d'opération industrielle

Dans des environnements industriels, un manipulateur mobile appelé Little Helper (LH), avec sept versions, a été développé à l'Université d'Aalborg. Le projet TAPAS a testé des MM comme LH et omniRob dans des conditions réelles pour évaluer leur préparation industrielle. Deux types de MM, LH et omniRob, ont collaboré : LH pour l'assemblage des rotors et omniRob pour la logistique chez Grundfos. Un contrôleur central (MPC) gérait leur planification. Les défis principaux concernaient la sécurité, l'intégration avec le système de production, et les problèmes de communication, souvent liés aux pertes de connexion.

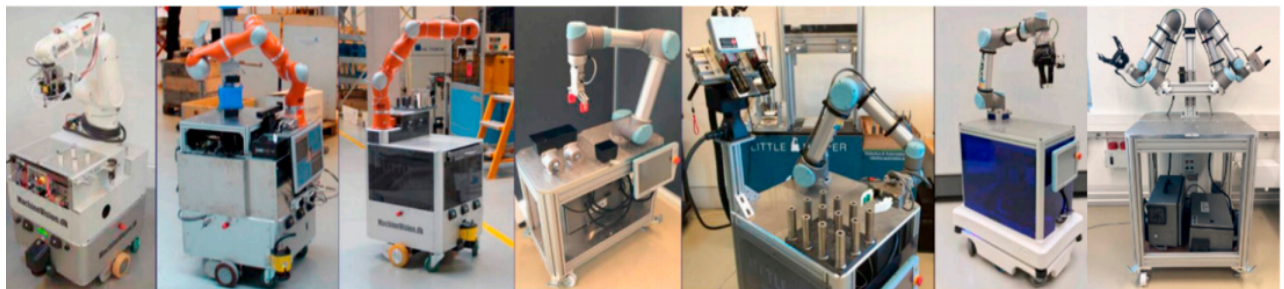


Figure 5. Petits assistants de la 1^{ère} à la 7^{ème} génération (de gauche à droite) adoptés.

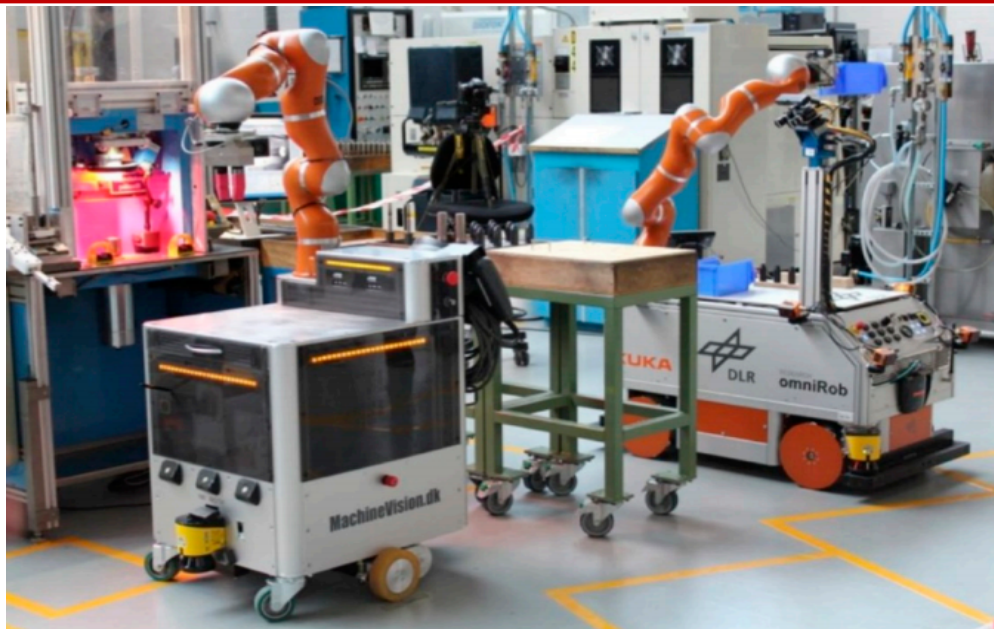


Figure 6. Little Helper 3 de l'Univ-Aalborg (à gauche), omniRob de Kuka (à droite).

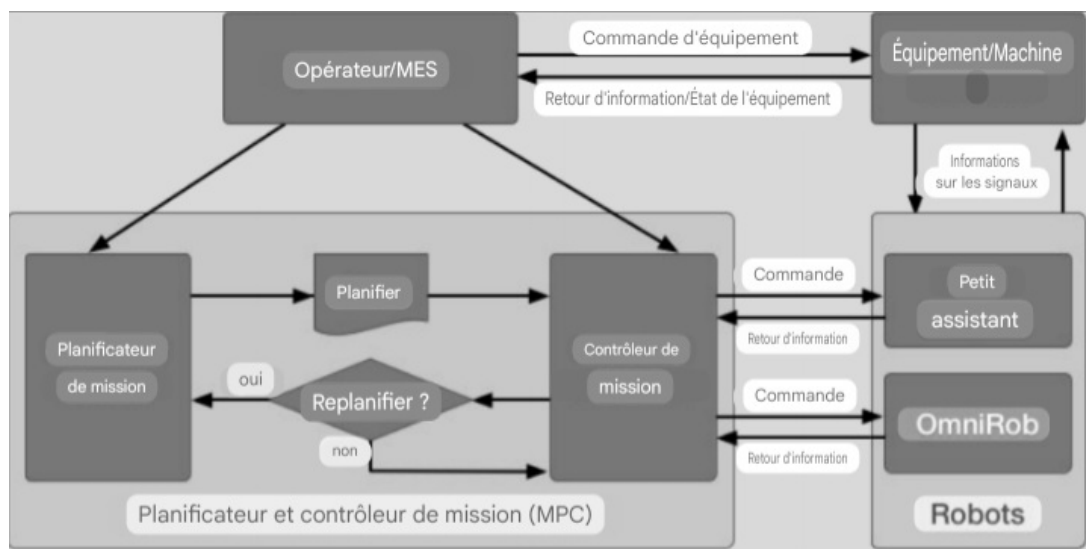


Figure 7. Système de contrôle intégrant les MM dans les systèmes de production [1].

Les auteurs cités ont proposé diverses approches pour l'intégration et mise en œuvre

de manipulateurs mobiles dans des environnements industriels. O.Madsen et al., ont expérimenté l'utilisation de deux MM pour la production de pompes, mais ont rencontré des problèmes de performance et d'efficacité, notamment des temps de cycle élevés et des erreurs de navigation. Ils ont identifié les défis liés à la sécurité, cela robustesse et la flexibilité des MM

Nielsen et al., ont développé une méthodologie pour la mise en œuvre de MM dans des environnements flexibilités, en se basant sur la modularité, l'autonomie, et la standardisation. Ils ont proposé une application d'alimentation en plusieurs parties et ont testé leur approche dans un site de production réel.

Ghodsian et al. Ont expérimenté intégration des MM dans les industries 4.0 avec un manipulateur mobile appelé «MOMA», supervisé par un système MES, pour automatiser un processus de production. Ils ont rencontré des erreurs liées à la communication et la navigation.

R Eckhooldt Andersen et al. ont implémenté un système basé sur les compétences dans une usine basé sur les calibrations du bras robotiques pour compenser les erreurs de la plate-forme mobile. Leur étude a révélé que la communication entre les modules était le principal obstacle à l'efficacité du système.

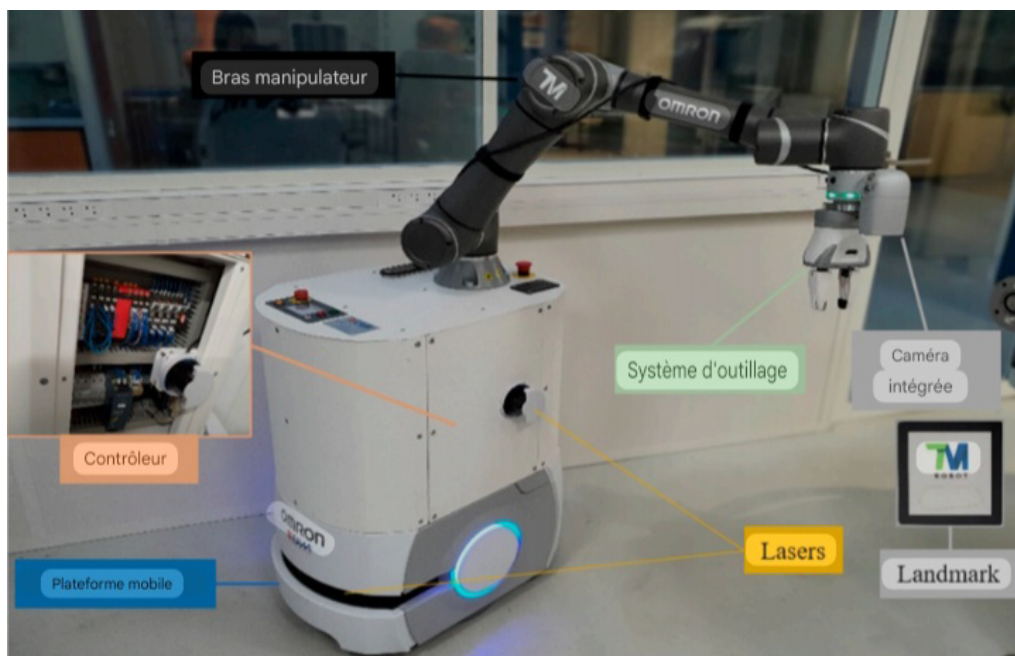


Figure 8. MOMA d'OMRON adopté de [1].

Le projet SOFITEC du programme Sherlock a exploré l'utilisation de manipulateurs mobiles à double bras pour co-manipuler et positionner de grandes pièces composites en fibre de carbone. Ces pièces, bien que légères, sont trop grandes pour être manipulées par une seule personne. Les MM visent à automatiser cette tâche, traditionnellement effectuée par plusieurs opérateurs, afin de rendre le processus plus efficace et précis

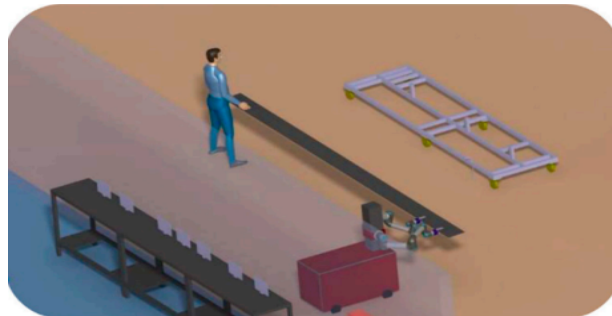


Figure 9. MM pour transporter de grandes pièces avec la coopération humaine [1]

Ce scénario du projet Sherlock porte sur la chaîne de montage de capteurs solaires thermiques chez CALPAK, où l'opérateur doit assembler des capteurs de différentes tailles et poids, ce qui entraîne des risques de fatigue et de douleurs chroniques. Pour alléger cette charge, Sherlock propose l'utilisation d'un manipulateur mobile collaboratif à charge utile élevée. Ce robot aide à soulever et déplacer les capteurs lourds, tandis que l'opérateur se concentre sur des tâches plus petites. Des capteurs IA surveillent les mouvements de l'opérateur pour garantir le bon déroulement du processus, et des outils de réalité augmentée (AR) guident l'opérateur sur les étapes à suivre.

D'autres applications du MM rob@work incluent l'assemblage de pompes hydrauliques et l'assistance au soudage, ainsi que l'assemblage collaboratif chez Airbus dans le projet SAPHARI.



Figure 10. Projet SAPHARI, applications industrielles requises chez KUKA (à gauche) et exigence d'assemblage d'avions d'Airbus (à droite) [1]

Le projet ColRobot intégré des technologies robotiques avancées pour assister les opérateurs dans les processus d'assemblage. Un manipulateur mobile est utilisée comme «troisième main » pour livrer des kits, outils et pièces, tout en maintenant les composants pour faciliter le travail de l'opérateur. Ce système a été validé dans des conditions réelles dans les industries automobiles et aérospatiales. Un prototype ColRobot (KMR IIWA) a été testé à l'ENSAM pour des opérations de kitting chez THALES ALENIA SPACE France. La Figure 11 illustre son utilisation pour le serrage et le remplissage automatique dans l'industrie automobile.



Figure 11. Prototype ColRobot pour les industries automobiles [1].

Le projet BAZAR, développé en 2016 au LIRMM, est un manipulateur mobile à deux bras, équipé de deux bras KUKA LWR4+et d'une plate-forme mobile Neobotix MPO 700. Le projet européen VERSATILE a étudié l'utilisation de MM dans trois secteurs : l'assemblage de tableaux de bord chez PSA, de pièces d'ailes chez AIRBUS, et la manutention de manches de rasoir chez BIC.

Enfin, un cas d'étude mené à la centrale énergétique VALLE a utilisé des capteurs à ultrasons pour réguler la trajectoire d'un outil lors de l'inspection de miroirs.



Figure 12. Mesurer la réflectivité du miroir - central énergétique Valle par MainBOT [1]

LIAAM (Institut d'automatisation et de mécatronique appliquées) développe des projets multidisciplinaires comme FiberRadar, qui améliore la qualité des matériaux composites à fibres en utilisant le MM OMNIVIL pour visualiser les structures de fibres en 3D avant la production. Un système de mesure transportable est également en cours de développement pour la fabrication d'éoliennes. Le projet STAMINA vise à concevoir une flotte de robots pour la manutention de pièces dans l'industrie automobile, avec une cadence de production rapide. En aérospatial et construction navale, le projet VALERI valide l'utilisation de MM aux cotée d'humains, proposant un robot léger KUKA monté sur une plate-forme mobile omniRob, conçu des taches comme l'application et l'inspection de produits d'étanchéité. VALERI aborde des aspects de sécurité et d'interaction homme-robot, en étudiant les Technologies de capteurs, les techniques de fusion de capteurs, et les stratégies de contrôle pour garantir efficacité et sécurité .L'application industrielle des MM dans des situations réelles est résumée dans le tableau 2.

Tableau 2. Résumé des applications industrielles des MM [1].

Application	Industrie	Projet	MM	Année
Multiple-part feeding	Grundfos		Little Helper	2011
Assembly, machine tending, quality control	Grundfos	TAPAS	LH3 and omni Rob	2014
Assembly, quality control, logistics	Grundfos	TAPAS	Two LHS	2015
Multiple-part feeding	CR 1-2-3 impeller line		LH	2017
Part feeding	Festo cyber-physical factory		LH6	2017
Placement of common object with human	Aerospace	SHERLOCK		2014
High payload assembly	CALPAK	SHERLOCK		2023
Assembly of hydraulic pumps and the assistance of manual gas metal are welding	Hydraulic Pumps	MORPHA	rob@work	2002

Logistics and assembly	Automotive aerospace	SAPHARI	omni Rob	2012
Kitting	TAS-F	ColRobot	KMR IIWA	2020
Fastener tightening and automatic refill	Automotive industries	ColRobot	ColRobot prototype	2020
Smart logistics, assembly of vehicle dashboards, assembly of aircraft wing parts, handling & packaging of shaver handles	PSA, AIRBUS, BIC	VERSATILE	BAZAR	2019
Fetch-and-carry	Clean room	ISABEL	ISABEL	2023
Quality control	Production of wind turbines	FiberRadar	OMNIVIL	2021
Bin-picking, kitting	PSA	STAMINA		2014
Applying sealant and visual inspection	Aerospace, shipbuilding	VALERI	VALERI	2014
Industrial maintenance	VALLLE energy plant	MAINBOT	MAINBOT	2014, 2023

4. Industrie 4.0 : exigence pour le développement de MM adaptables

Les travailleurs de l'atelier doivent être en mesure d'utiliser et de modifier instantanément les MM pour de nouvelles tâches de traitement, et ils doivent être gérables par le système de gestion de l'entreprise.

4.1. Connectivité et interopérabilité accrues

Consiste à établir une architecture décentralisée entre les composants du système. Les technologies de communication IoT et M2M ont avancé, améliorant la connectivité de MM dans les contextes industriels.

Grâce à cette connexion améliorée, les MM peuvent désormais travailler avec d'autres appareils plus efficacement et effectuer des tâches plus compliquées. Afin de promouvoir cette interopérabilité et cette connectivité, un certain nombre de normes et de protocoles se sont développés dans le 'IoT'.

4.2. Collecte et analyse des données

La collecte et l'analyse des données en temps réel sont essentielles dans le cadre de l'industrie 4.0 permettant aux machines intelligentes (MM) de s'équiper de capteurs variés pour rassembler des données nécessaires à la gestion. Des analyses avancées et des algorithmes de machine learning optimisent les performances des MM et leur adaptation à un environnement dynamique. L'analyse de grandes quantités de données aide à améliorer l'efficacité et la productivité, avec des méthodologies d'IA capables de prévoir des problèmes grâce à la maintenance prédictive. L'autonomie des systèmes augmentera avec

la gestion efficace des transferts de données, favorisant l'auto-optimisation et des outils décisionnels améliorés.

4.3. Flexibilité améliorée

La flexibilité améliorée se définit comme la capacité des systèmes à s'adapter rapidement aux variations des conditions et des exigences. Dans le cadre de l'industrie 4.0, cette flexibilité est accentuée par l'utilisation de technologies avancées qui permettent aux systèmes de fabrication d'être plus adaptables. Il est essentiel que les modules de production (MM) fonctionnent de manière flexible et soient intégrables au système de production, assurent ainsi une réponse dynamique à tout changement détecté dans l'ensemble du système. Des recherches sont en cours pour traiter ces défis de flexibilité.

4.3.1. Navigation et planification d'itinéraires pour robots mobiles

La navigation flexible et la planification des itinéraires pour les robots mobiles reposent sur deux types de programmation : préprogrammée, pour des environnements connus, et en temps réel, pour des environnements inconnus. Des techniques comme SLAM (localisation et cartographie simultanées) et le suivi de ligne avec des indices visuels sont cruciales pour améliorer l'efficacité des systèmes de navigation. Les défis incluent la coordination entre la base mobile et le manipulateur, ainsi que l'application d'approches d'apprentissage automatique pour optimiser la planification des mouvements. La planification de chemin est divisée en planification globale dans un environnement connu et planification local pour des obstacles imprévus. Divers algorithmes, tels que le recuit simulé et les algorithmes génétiques, sont utilisés pour résoudre des problèmes d'optimisation dans la planification de trajectoires, en tenant compte des contraintes mécaniques et dynamiques. Plusieurs études précisent des méthodologies variées, incluant la rétroaction et des approches stochastiques pour garantir des mouvements efficaces et éviter les collisions.

4.3.2. Programmation flexible pour systèmes de production hybrides

La programmation flexible dans les systèmes de production hybrides est abordée à travers un cadre de prise de décision utilisant le Robot Operating System (ROS¹) pour optimiser la planification des tâches et des mouvements des ressources mobiles. Ce cadre permet d'intégrer facilement différents types de robots et d'optimiser les opérations d'assemblage en fonction de critères tels que le temps et la distance de déplacement. Il offre des avantages comme la reconfiguration des systèmes de production en temps réel et

¹ ROS, <https://www.ros.org/>

la génération de plans de mouvement efficaces. Une étude sur un problème de planification pour un robot mobile a mis en évidence la nécessité d'une séquence d'alimentation optimale pour un temps de déplacement réduit, en utilisant un modèle de programmation linéaire. La complexité augmente avec le nombre d'alimentateurs, ce qui nécessite des approches méta-heuristiques. Des améliorations futures viseront à intégrer ce système dans des usines existantes, rendant la prise de décision plus pertinente et pratique.



Figure 13. MM conçu dans le projet THOMAS [1].

4.4. Virtualization

L'industrie 4.0 utilise la virtualisation pour permettre un contrôle à distance des manipulateurs mobiles grâce à des technologies comme la réalité virtuelle (RV) et augmentée (RA). Ces technologies créent des environnements virtuels en temps réel pour optimiser la conception des produits, améliorer la production et réduire les coûts. Les MM offrent une meilleure visualisation du travail, permettant de manipuler avec plus précision.

L'intégration des MM avec l'Internet des objets (IoT) et les systèmes cyber - physiques (CPS) permet de collecter des données pour améliorer la productivité. Un exemple est l'utilisation des jumeaux numériques (DT) pour simuler et surveiller les processus, incluant la détection de personnes via des capteurs 2D et 3D.

Les défis à relever incluent l'amélioration des simulations et des flux de données pour créer des modèles plus précis. La 'fabrication en nuage' (CMfg) propose d'utiliser ces technologies via Internet pour transformer les ressources en services de fabrication. Cela permet une gestion centralisée des MM et une meilleure flexibilité dans la production, ce qui est particulièrement bénéfique pour la production de petits lots et des cycles de production

longs.

4.5. Cyber-sécurité

La cyber-sécurité est essentielle pour l'industrie 4.0, et les manipulateurs mobiles (MM) doivent respecter des normes pour protéger les processus de production. Ils doivent sécuriser les canaux de communication pour éviter les fuites de données et les attaques, que ce soit via le réseau ou des connexions physiques comme les ports USB. Les MM doivent interagir avec d'autres systèmes et toute sécurité et être protégés contre les manipulations non autorisées lors de leurs utilisations.

Une architecture de sécurité a été créée pour le robot CHIMERA, basée sur l'isolation des sous-systèmes pour limiter les attaques. Des outils comme AppArmor sont utilisés pour restreindre l'accès aux composants critiques, assurant ainsi une sécurité renforcée.

5. Discussion

Diverses technologies et outils ont été développés pour faciliter l'intégration de manipulateur mobile dans les systèmes de production intelligents de l'Industrie 4.0. La sécurité est un aspect clé de cette intégration, aussi bien pour le MM que pour les humains et machines qui l'entourent. Des outils permettent aux MM de collaborer en toute sécurité avec les personnes, notamment grâce à des systèmes de guidage par mains humaines. Pour assurer cette sécurité, les MM doivent être équipés de capteurs capables d'interpréter leur environnement et de communiquer avec d'autres machines et systèmes de contrôle.

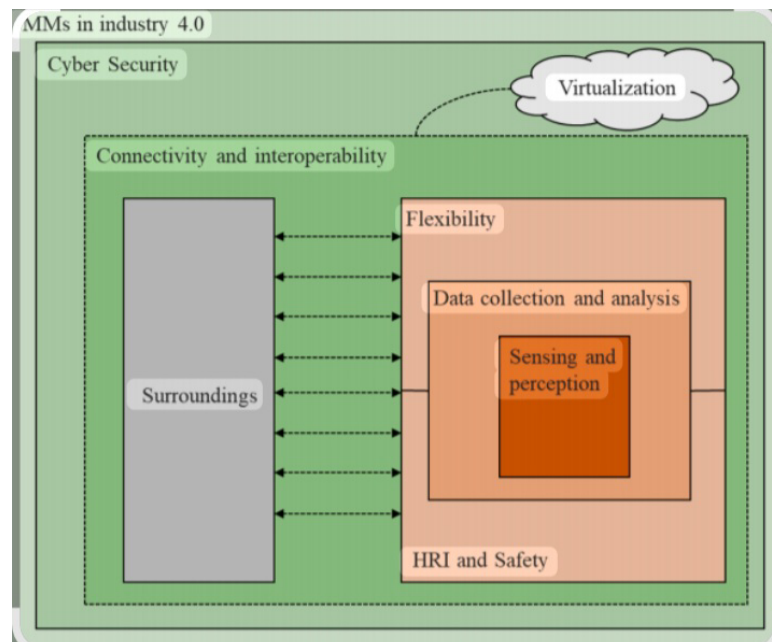


Figure 14. Lien sémantique entre les principes clés à prendre en compte pour leur intégration dans les systèmes de fabrication basés sur l'industrie 4.0

La méthodologie du niveau de maturité technologique (TRL) est utilisée par des organisations comme la NASA pour évaluer la maturité d'une technologie sur une échelle de 1 à 9, où 9 représente une technologie entièrement développée et testée en conditions réelles. Les phases 1 à 3 concernent l'idée et le concept, les Phases 4 à 5 impliquent le développement de prototype, et les phases 6 à 9 se concentrent sur la validation et les Tests en environnement opérationnel. Bien que de nombreuses technologies soient testées en laboratoire, il est plus difficile de simuler un environnement industriel réel, où les conditions sont plus complexes. Tester les manipulateurs mobiles dans des situations réelles permettrait d'obtenir des informations plus pertinentes pour la recherche. Les MM ont fait des progrès dans des domaines comme la logistique, l'automatisation et l'intégration des systèmes de perception. Avec les avancées en IA et en apprentissage automatique, leurs performances devraient s'améliorer considérablement à l'avenir.

TRL 9	Système réel « éprouvé en vol » grâce à des opérations de mission réussies
TRL 8	Système réel terminé et « qualifié en vol » par des tests et une démonstration
TRL 7	Démonstration d'un prototype de système dans un environnement spatial
TRL 6	Démonstration d'un modèle ou d'un prototype de système/sous-système dans un environnement pertinent
TRL 5	Validation des composants et/ou de la maquette dans un environnement pertinent
TRL 4	Validation des composants et/ou de la maquette en laboratoire
TRL 3	Fonction critique analytique et expérimentale et/ou preuve de concept caractéristique
TRL 2	Concept technologique et/ou application formulés
TRL 1	Principes de base observés et rapportés

Figure 15. Niveau de maturité technologique

Cette étude a principalement porté sur les manipulateurs mobiles (MM), mais elle n'a pas examiné de manière exhaustive les avancées des robots mobiles et des bras robotisés, qui sont des domaines connexes. Certaines innovations dans les robots mobiles pourraient également être pertinentes explorées ici. Les résultats de cette étude présentent des applications potentielles pour les industries cherchant à automatiser ou à améliorer leurs processus en utilisant des technologies liées aux MM, favorisant une adoption plus alignée sur les principes de l'industrie 4.0. Les lacunes identifiées et les tendances émergentes dans les MM offrent de perspectives aux chercheurs pour adapter leurs futurs projets aux besoins de cette nouvelle ère industrielle.

6. Conclusion

Les conclusions de cette partie mettent en évidence l'importance des manipulateurs mobiles (MM) dans l'amélioration de la flexibilité et de l'efficacité des systèmes de fabrication dans le contexte de l'industrie 4.0. Les technologies émergentes comme le Cloud, l'intelligence artificielle (IA), et l'internet des objets (IoT) jouent un rôle crucial dans la création de systèmes de production autonomes et adaptables. Les MM, en combinant les bras robotisés avec des robots mobiles, sont particulièrement utiles dans des tâches intra-logistiques et autres opérations en atelier, telles que la maintenance, la soudure et la peinture.

L'étude souligne également le potentiel de l'IA pour améliorer la navigation autonome des robots mobiles et la flexibilité des processus industriels. Les avancées en matière de robots mobiles autonomes (AMR) facilitent l'intégration des MM, qui peuvent collaborer avec les humaines pour augmenter la productivité et la sécurité.

Cependant, les défis liés à la sécurité, la communication et l'alimentation des MM doivent encore être résolus. Bien que la technologie des MM soit prometteuse, elle reste à un stade de maturité intermédiaire (TRL 6 ou 7). Pour une adoption à grande échelle, il est judicieux de se concentrer sur la fiabilité des systèmes, la gestion de l'énergie et d'effectuer des tests expérimentaux approfondis en conditions réelles. Ces améliorations permettront aux MM de jouer un rôle encore plus central dans l'industrie 4.0, en créant des processus de productions plus durables et des environnements de travail collaboratifs et inclusifs.

Chapitre 2 : Modélisation et conception, analyse fonctionnelle et technique

1. Introduction

Dans ce chapitre, nous allons présenter une analyse détaillée des composants matériels sélectionnés pour réaliser notre bras manipulateur mobile. Chaque élément sera étudié en mettant en avant ses caractéristiques techniques, son rôle au sein du système, ainsi que les raisons ayant motivé son choix. Cette étude nous permettra de mieux comprendre l'importance de chaque composant dans le fonctionnement global du projet et d'assurer une intégration cohérente et efficace de l'ensemble du matériel.

2. Architecture logicielle du manipulateur mobile

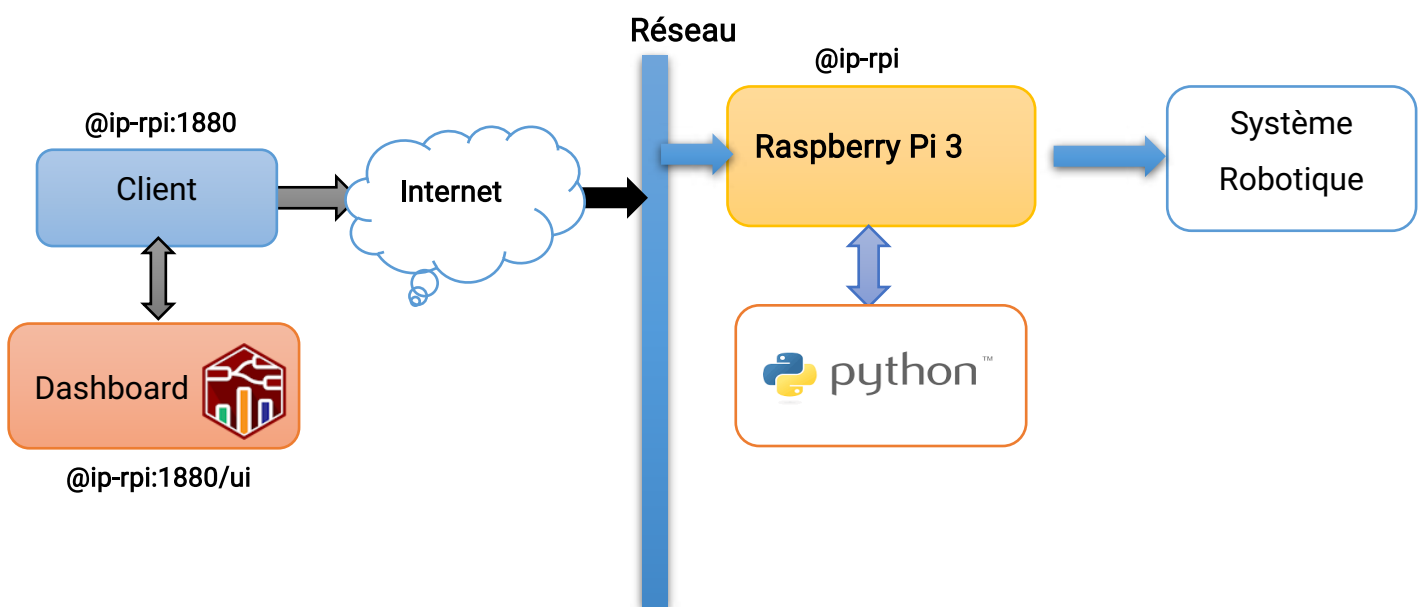


Figure 16. Schéma bloc de fonctionnement du bras manipulateur mobile.

3. Description du bras manipulateur mobile

Un bras manipulateur est bras d'un robot généralement programmable, avec des fonctions similaires à un bras humain. Les liens de ce manipulateur sont reliés par des axes permettant, soit de mouvement de rotation ou de translation de déplacement.

La mobilité, Contrairement au robot qui généralement fixé, le robot mobile est doté de moyens qui lui permettent de se déplacer dans son espace de travail.

4. Outils de conception de travail

Pour réaliser notre conception, qui consiste utiliser un bras robotique avec un châssis mobile, nous avons utilisé plusieurs outils d'assistance, que nous allons expliquer simplement comme suit :

4.1. Système de développement Node-Red²

Node-Red est un outil puissant pour construire des applications de l'Internet des Objets en mettant l'accent sur la simplification de programmation qui se fait grâce à des blocs de code prédéfinis appelés '**nodes**' pour effectuer des tâches. Il utilise une approche de programmation visuelle qui permet aux développeurs de connecter les blocs de code ensemble pour former un flux.

Les nœuds connectés, généralement une combinaison de nœuds de traitement et de nœuds de sortie, lorsqu'ils sont câblés ensemble, constituent un flow.

Node-Red est construit sur **Node.js**, tirant pleinement parti de son modèle non bloquant piloté par les événements. Cela le rend idéal pour fonctionner en périphérie du réseau sur un serveur d'application qui peut être un matériel à faible coût tel que Raspberry Pi ou un serveur dans le Cloud.

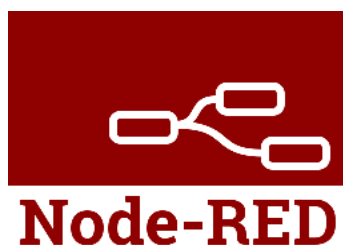


Figure 17. Logo de Node-RED

Le Dashboard est un module de nœuds complémentaires dans Node-RED qui permet de créer facilement des interfaces utilisateur graphiques (UI) interactives et en temps réel. Il s'intègre directement dans l'environnement Node-RED et permet de visualiser, contrôler et interagir avec les données des flux, en utilisant les éléments d'entrée et éléments de sortie du module Node-Red-Dashboard.

² <https://nodered.org/>

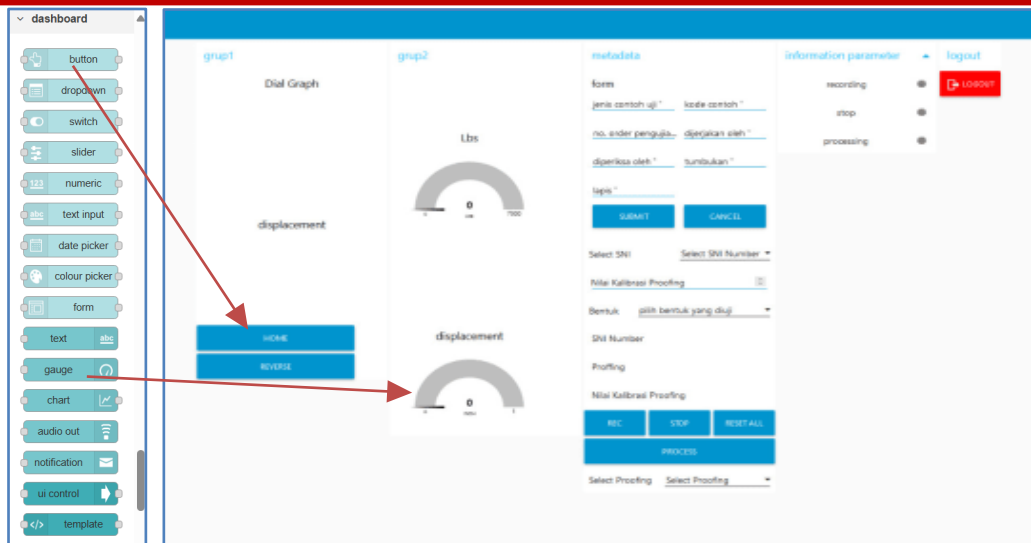


Figure 18. Module et Interface de Dashboard

4.2. Python³

Python est un langage de script de haut niveau, portable, dynamique, extensible, gratuit, structuré et open source conçu pour être orienté objet. Il est multi- paradigme et multi-usage. Développé à l'origine par Guido van Rossum en 1993, il est, comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde.



Figure 19. Logo de python

4.3. Bitwise SSH client⁴

Bitwise SSH est un logiciel sécurisé qui permet d'établir une connexion distance via le protocole SSH (Secure Shell). Il est principalement utilisé pour accéder à distance à des

³ [https:// www.python.org](https://www.python.org)

⁴ Bitwise. (s. d.). Bitwise SSH client. Consulté le 22 avril 2025, depuis <https://www.bitwise.com/ssh-client>

systèmes Windows ou Linux, transférer des fichiers de manière sécurisée (SFTP), et rediriger des ports. Grâce à son interface graphique conviviale, il facilite la gestion des connexions SSH sans nécessiter l'utilisation de lignes de commande complexes.



Figure 20. Logo de Bitvise SSH client

4.4. Définition de Advanced IP Scanner⁵

Advanced IP Scanner est un logiciel gratuit pour Windows qui permet d'analyser un réseau local (LAN) afin d'identifier tous les appareils connectés. Il peut scanner des centaines d'adresses IP en quelques secondes, afficher les adresses MAC, noms d'hôtes, fabricants des cartes réseau, et permet aussi des actions à distance comme l'arrêt ou le démarrage des machines via RDP et Wake -on-LAN.



Figure 21. Logo de Advanced IP Scanner

4.5. Présentation du Raspberry Pi⁶

Raspberry est une carte mère d'un mini-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier...). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation

⁵ Famatech. (2024).Advanced IP Scanner. Disponible sur : <https://www.Advanced-ip-scanner.com/fr/>

⁶ www.raspberrypi.com

informatique en plusieurs langages (python, scratch...). Elle est aussi capable de lire les vidéos à haute définition et même à installer des jeux vidéo. L'intérêt d'utiliser le Raspberry Pi est sa capacité d'interaction avec le monde extérieur et d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian Debian ...) et des autres logiciels compatibles.⁷

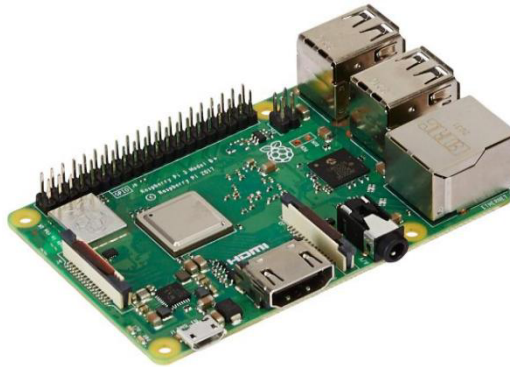


Figure 22. Raspberry Pi 3

4.5.1. Caractéristiques du Raspberry Pi utilisé

Le Raspberry Pi est un mini-ordinateur de la taille d'une carte de crédit, avec les caractéristiques suivantes :

- Processeur ARM cortex A-53 quatre cœurs de 64-bit cadencé à 1,2Ghz
- 1Go de RAM (SDRAM)
- Un contrôleur vidéo Broadcom VideoCore IV
- 4 ports USB 2.0
- Un port HDMI
- Un connecteur Jack 3,5 mm/sortie composite
- Une prise Ethernet 10/100Mbps
- WIFI 802.11n
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 40 pins GPIO
- Une interface de caméra (CSI)
- Une interface d'affichage (DSI)

- Un emplacement de carte micro SD
- Alimentation 5V par micro USB

La figure suivante illustre les différentes parties d'un Raspberry Pi 3 modèle B.

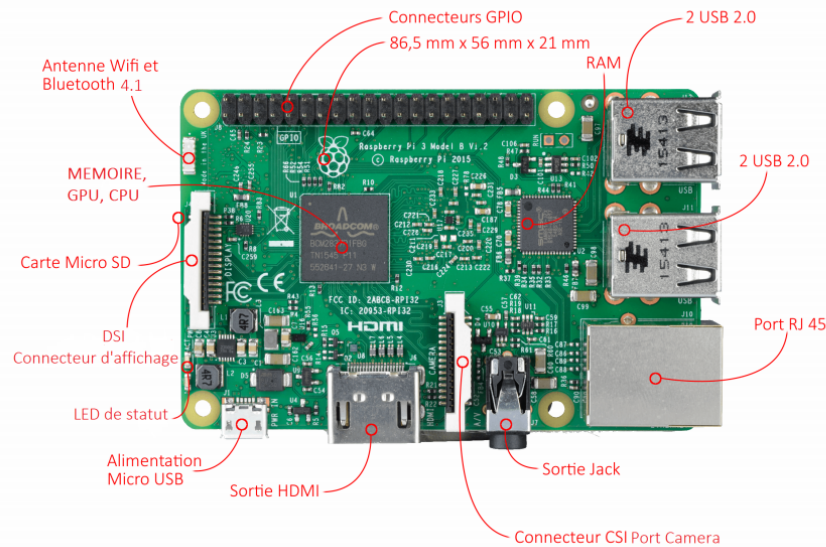


Figure 23. Emplacement de différents composants de Raspberry Pi 3 modèle B

4.5.2. Alimentation

Le Raspberry Pi s'alimente sous tension unique de 5 volts, tension sur laquelle il peut consommer jusqu'à 1.8 A selon les tâches qu'il exécute. Cette alimentation doit être normalement fournie via le connecteur micro USB placé dans un angle de la carte.

En effet, l'utilisation d'un chargeur pour smartphone équipé d'un micro USB qui délivre une tension de 5 volts avec minimum 1.8A est suffisante pour alimenter notre Raspberry Pi.

4.5.3. Choix du Raspberry Pi 3

Dans le cadre de notre projet, nous avons choisi d'utiliser le Raspberry Pi 3 comme unité principale de traitement et de commande. Ce choix est motivé par sa polyvalence, sa puissance de calcul suffisante pour nos besoins, ainsi que sa connectivité intégrée (WIFI, Bluetooth, port RJ45).

Le Raspberry Pi 3 nous permet de lancer des scripts en Python, d'utiliser des outils comme

Node-Red pour l'interface graphique de contrôle, et de communiquer facilement avec

d'autres composants via ses ports GPIO. Il offre aussi la possibilité de contrôler des modules comme le L298N pour les moteurs ou PCA9685 pour les servomoteurs.

4.5.4. Connecteurs et ports disponibles sur Raspberry Pi

1-HDMI : «High Definition multimedia Interface» permet de relier le Raspberry Pi à un dispositif compatible : écran LCD ou un vidéo-projecteur.

2-GPIO : est l'acronyme anglais «General Purpose Input Output» ce que l'on traduit par Entrée/Sortie pour usage général. Le connecteur GPIO dispose actuellement de différents types de connexions :

- Des broches GPIO avec lesquels il est possible d'allumer/éteindre des LED, etc.
- Des broches pour une interface I2C qui permet de se connecter sur du matériel en utilisant uniquement 2 broches/pins de contrôle.
- Une interface bus SPI pour les périphériques SPI, similaire à I2C mais avec un standard différent.
- Les broches Rx et Tx pour la communication avec les périphériques séries.

De plus, certaines broches peuvent être utilisées en :

- PWM (Pulse Width Modulation : pour modulation de largeur d'impulsion) permettant le contrôle de puissance.
- PPM (Pulse Position Modulation : pour modulation positionnelle d'impulsion) permettant de contrôler des servomoteurs.

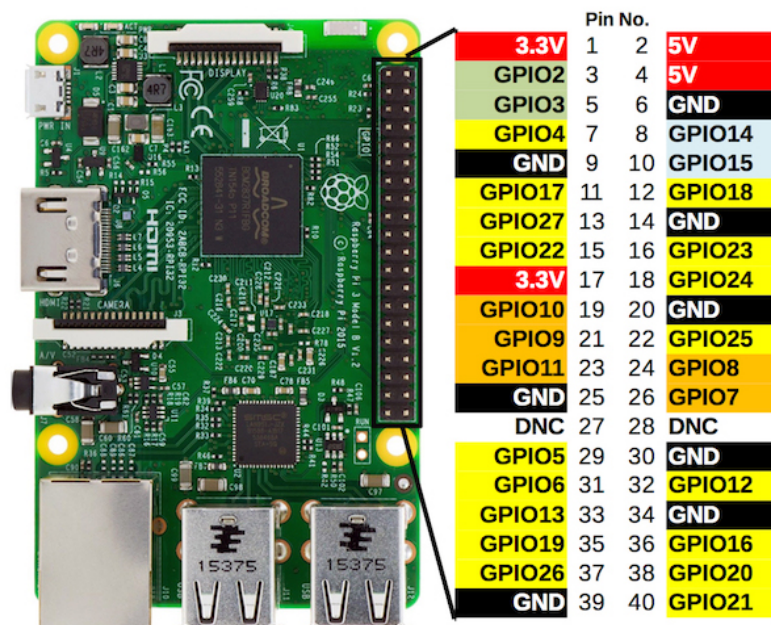


Figure 24. Ports GPIO

3-Ports USB 2.0 : Le port «Universal Serial Bus» est un port série qui sert à connecter le Raspberry aux autres périphérique, et il y a 4 ports USB. Le Raspberry Pi fonctionne avec pratiquement n'importe quels claviers et souris USB. Vous pouvez aussi utiliser la plupart des claviers et des souris sans fils (ceux qui marchent avec un d'ongle que l'on branche un port USB

4-Port Ethernet : C'est un port qui correspond au protocole Ethernet de réseau local à communication de paquets.

5-Un slot cartes micro-SD : Le Raspberry a besoin d'une mémoire externe pour fonctionner. Ce slot permet de connecter la carte mémoire SD, qui être préparée avec une image disque d'un système d'exploitation propre à raspberry pi.

6-Une prise jack : connectique audio-vidéo sa dimension est égale à 3,5mm.

4.6. Présentation du moteur NEMA 17(JK 42HS34 1334AC)⁸

Dans le cadre de notre projet, nous avons opté pour l'utilisation d'un moteur pas à pas de type NEMA17, plus précisément le modèle JK 42HS34 1334AC. Ce moteur est largement utilisé dans les systèmes de positionnement tels que les imprimantes 3D et les machines CNC, grâce à sa capacité à effectuer des mouvements précis et contrôlés.

Le terme 'NEMA 17' se réfère à un standard de dimensions (National Electrical Manufacturer Association), indiquant que la face du moteur mesure 1.7 pouces, soit environ 42mm. Ce format est devenu une référence dans les projets d'automatisation de petites et moyenne envergure.

⁸ www.handsontec.com

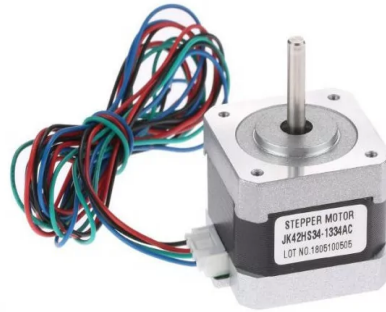


Figure 25. NEMA 17(JK 42HS34 1334AC)

4.6.1. Caractéristiques principales du moteur

Le moteur JK 42HS34 1334AC présente plusieurs caractéristiques techniques qui répondent bien aux exigences de notre système :

- ◆ **Type** : Moteur pas à pas bipolaire, ce qui signifie qu'il dispose de deux enroulements indépendants permettant un contrôle plus précis des mouvements.
- ◆ **Nombre de pas par tour** : 200 pas, soit une résolution de 1.8° par pas, ce qui offre une grande précision sans nécessaire de système de rétroaction.
- ◆ **Courant nominal** : Le courant nominal du moteur est de 1.33 A. Il s'agit du courant recommandé par le constructeur pour garantir un fonctionnement optimal sans risque de surchauffe. Ce courant permet d'exploiter efficacement le couple du moteur tout en assurant sa longévité.
- ◆ **Couple (Torque)** : Environ 3.2 à 4.5 Kg.cm, suffisamment puissant pour les charges légères à moyennes.
- ◆ **Résistances par phase** : Environ 2.1 ohms, ce qui détermine sa tension de fonctionnement en lien avec le courant.
- ◆ **Longueur du boîtier** : 34 mm, d'où le code "HS34" dans sa référence.
- ◆ **Axe de sortie** : Diamètre de 5 mm, standard pour l'intégration avec des poulies ou des engrenages.

4.6.2. Choix du moteur

Le choix de ce moteur n'est pas anodin. Sa précision, sa fiabilité, et sa compatibilité avec des drivers courants continus comme le L298N en font une solution idéale pour la partie mobile de notre projet. Il nous offre un excellent compromis entre performance et

simplicité de mise en œuvre.

Le moteur NEMA 17 (JK 42HS34) a été choisi après une analyse approfondie des besoins du châssis mobile, notamment en termes de masse, de couple nécessaire et de conditions de déplacement. Son courant nominal de 1.33 A et son couple de 3.2 à 4.5 kg .cm s'avèrent suffisants pour assurer un mouvement fluide et précis du système.

4.7. Driver de moteur à double pont en H L298N9

Ce contrôleur de moteur bidirectionnel double est basé sur le très populaire circuit intégré de commande de moteur à double pont en H L298. Ce circuit permet de contrôler facilement et indépendamment deux moteurs jusqu'à 2A chacun dans les deux sens. Idéal pour les applications robotiques, il est parfaitement adapté à la connexion à un microcontrôleur ne nécessitant que quelques lignes de commande par moteur. Il peut également être interfacé avec des interrupteurs manuels simples, des portes logiques TTL, des relais, etc. Cette carte est équipée de voyants d'alimentation LED, d'un régulateur +5 V intégré et de diodes de protection.

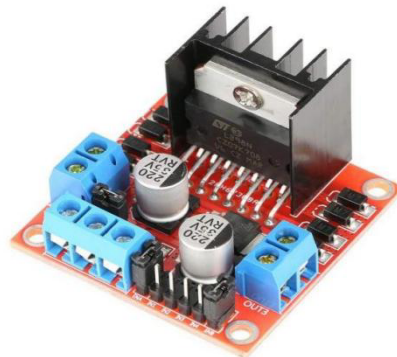


Figure 26. Driver de moteur à double pont en H L298N

4.7.1. Caractéristique du driver L298N

- ✓ Tension d'entrée : 3.2 V - 40 Vcc
- ✓ Pilote : Pilote de moteur CC double pont en H L298N
- ✓ Alimentation : 5 V CC-35 V
- ✓ Courant de crête : 2 A
- ✓ Plage de courant de fonctionnement : 0-36 mA
- ✓ Plage de tension d'entrée du signal de commande :

⁹ www.handsontec.com/L298N

- ◆ Bas : $-0.3 \leq V_{in} \leq 1.5$ V
- ◆ Haut : 2.3 V $V_{in} \leq V_{ss}$
- ✓ Plage de tension d'entrée du signal d'activation :
 - ◆ Bas : -0.3 V $\leq V_{in} \leq 1.5$ V (signal de commande invalide)
 - ◆ Haut : 2.3 V $V_{in} \leq V_{ss}$ (signal de commande actif)
- ✓ Consommation électrique maximale : 20 W (lorsque la température $T=75^{\circ}\text{C}$)
- ✓ Température de stockage : -25°C à $+130^{\circ}\text{C}$
- ✓ Alimentation de sortie régulée +5V intégrée (alimentation de la carte contrôleur, par exemple Arduino)
- ✓ Dimensions : 3.4 cm \times 4.3cm \times 2.7cm

4.7.2. Choix du composant driver de moteur L298N

Dans notre projet de manipulateur mobile connecté, nous avons opté pour le module L298N pour le contrôle des moteur à pas à pas. Ce choix s'explique par plusieurs raisons techniques et pratiques qui répondent bien aux besoins de notre système embarqué.

- ◆ Double pont en H intégré

Le L298N contient deux ponts en H, ce qui permet de contrôler facilement deux moteurs DC, aussi bien en vitesse qu'en sens de rotation. Ceci est essentiel pour assurer les mouvements avant/arrière et gauche/droit du manipulateur mobile.

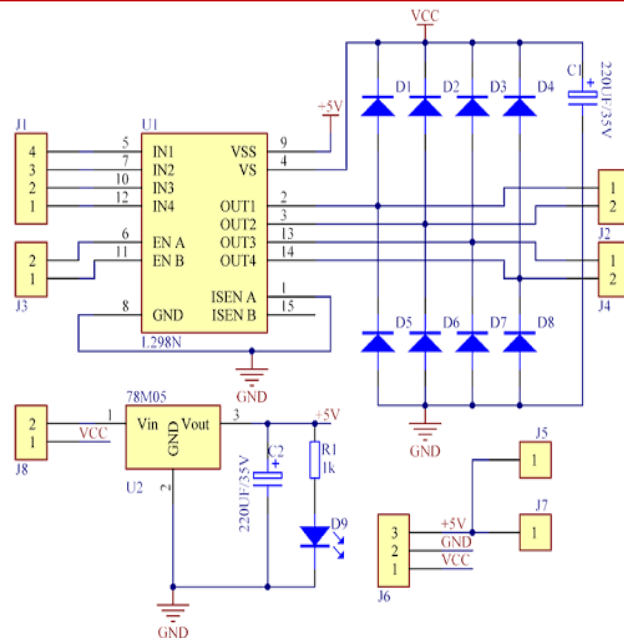


Figure 27. Schéma double pont H driver L298N.

✓ Compatibilité avec les microcontrôleurs

Ce module peut être piloté directement via des signaux logiques provenant de plateformes comme Arduino, ESP32 ou encore un système Node-Red connecté à des GPIO via MQTT ou python. Il accepte les signaux PWM pour ajuster la vitesse des moteurs.

✓ Plage de tension et courant adéquate

Le L298N peut fonctionner avec une tension d'alimentation allant de 5V à 35V, ce qui le rend compatible avec différents types de moteurs. Il peut fournir jusqu'à 2A par I, ce qui est suffisant pour les petits moteurs utilisés dans notre projet.

✓ Protection contre les retours de courant

Le module intègre des diodes de protection contre les retours de courant (back EMF), ce qui protège le microcontrôleur et les autres composants du système.

✓ Facilité d'intégration et disponibilité

Enfin, le L298N est facile à connecter, abordable, et largement disponible sur le marché. Cela en fait un choix idéal pour un prototype ou une application éducative comme la nôtre.

- ✓ Vitesse : 0.17 s/60° à 6 V
- ✓ Contrôlé par signal PWM
- ✓ 3 Fils : VCC, GND, et signal

4.8.2. Choix du servomoteur

Nous avons choisi le MG996R pour sa capacité à fournir un couple suffisant afin de faire bouger les articulations du bras manipulateur de manière stable et précise. C'est un composant fiable, largement documenté, facile à intégrer, et compatible avec nos autres modules.

4.9. Module PCA9685-Contrôleur PWM / I2C

Le PCA9685 est un module qui permet de générer des signaux PWM précis sur 16 canaux, contrôlables via le protocole I2C. Il est utilisé pour piloter plusieurs servomoteurs simultanément sans surcharger la carte du contrôleur [5].

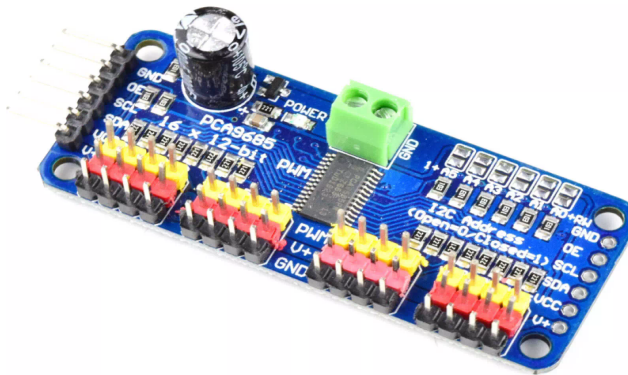


Figure 29. Module PCA9685-Contrôleur PWM/I2C

4.9.1. Caractéristique

- 16 Canaux PWM 12 bits
- Interface I2C
- Fréquences PWM : jusqu'à 1.6 kHz
- Alimentation : 3.3 V/5V pour la logique, alimentation séparée pour les servomoteurs

4.9.2. Choix de Module PCA9685-Contrôleur PWM/I2C

Le module PCA9685 nous permet de contrôler facilement plusieurs servomoteurs MG996R avec un seul microcontrôleur (comme Arduino ou ESP32), tout en évitant les limitations en sorties PWM. Son fonctionnement avec le protocole I2C le rend simple à

câbler et à programmer.

4.10. Module de détection aux Ultrasons HC-SR04

Le capteur HC-SR04 utilise les Ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter.



Figure 30. HC-SR04

4.10.1.

Caractéristiques

- Dimensions : 45mm×20mm×15mm ;
- Plage de mesure : 2cm à 400 cm ;
- Résolution de la mesure : 0.3cm ;
- Angle de mesure efficace : 15° ;
- Largeur d'impulsion sur l'entrée de déclenchement : 10µs (Trigger Input Pulse Width).

4.10.2.

Broches de connexion

- VCC : Alimentation +5V DC
- Trig : Entrée de déclenchement de la mesure (Trigger input)
- Echo : Sortie de mesure donnée en écho (Echo potput)
- GND : Masse de l'alimentation

4.10.3.

Fonctionnement

Pour déclencher une mesure, il faut présenter une impulsion «high» (5V) d'au moins 10 µs sur l'entrée «Trig ». Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 KHZ, puis il attend le signal réfléchi. Lorsque celui-ci est détecté, il envoie un signal «high» sur la sortie «Echo», dont la durée est proportionnelle à la distance mesurée.

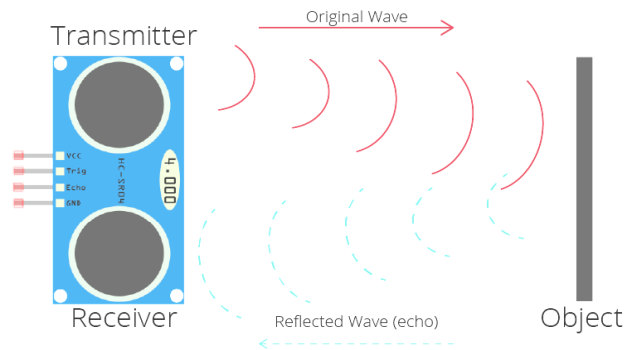


Figure 31. Fonctionnement du capteur HC-SR04

4.10.4. Distance de l'objet

La distance parcourue par un son se calcule en multipliant la vitesse du son, environ 340 m/s(ou 34000 cm/1000000 μ s) par le temps de propagation, soit $d=v*t$ (distance =vitesse*temps).

Le HC-SR04 donne une durée d'impulsion en dizaines de μ s. Il faut donc multiplier la valeur obtenue par 10 μ s pour obtenir le temps t. Sachant aussi que le son fait un aller-retour. La distance vaut donc la moitié.

$$d = 34000\text{cm}/1000000\mu\text{s} \times 10\mu\text{s} \times \text{valeur}/2$$

$$\text{En simplifiant : } d=170000/1000000 \text{ cm} \times \text{valeur}$$

$$\text{Finalement, } d=17/100 \text{ cm} \times \text{valeur}$$

4.11. Électroaimant¹²

Un électroaimant est dispositif électromagnétique qui génère un champ magnétique lorsqu'un courant électrique le traverse. Il est composé d'une bobine de fil conducteur enroulée autour d'un noyau métallique. Lorsqu'il est alimenté, il agit comme un aimant capable d'attirer des objets métalliques ferromagnétiques, et lorsqu'il est désactivé, il relâche ces objets.

Dans notre projet, l'électroaimant est utilisé comme système de préhension pour attraper et relâcher des pièces métalliques automatiquement.

¹² <https://www.mecalectro.com/les-electro-aimants-modes-de-fonctionnement/>



Figure 32. Electroaimant

Choix de l'électroaimant¹³

Nous avons choisi un module électroaimant 5V pour les raisons suivantes :

- ▶ **Simplicité d'utilisation** : il peut être facilement commandé par un relais ou un transistor à partir d'un Raspberry pi.
- ▶ **Compatibilité** : adapté pour les tensions standards utilisées dans les montages embarqués.
- ▶ **Efficacité** : il fournit une force suffisante pour soulever de petites pièces métalliques, ce qui correspond aux besoins du projet.
- ▶ **Contrôle numérique** : il fonctionne en tout-ou-rien (ON/OFF), ce qui permet une intégration simple avec Node-Red.

4.12. Châssis mobile

Le châssis mobile représente la base roulante du manipulateur. Il est généralement constitué d'une plate-forme munie de roues et de moteurs permettant le déplacement du système sur une surface plane.



Figure 33. Châssis mobile

¹³ <https://www.dzduino.com/capteur-de-levage-daimant-electrique-dc-5v-10n/>

4.12.1. Fonction principale

L'intégration d'un Châssis mobile permet d'augmenter la portée du bras et de travailler sur une zone plus large. Ainsi, cela permet d'assurer la mobilité du bras manipulateur en se déplaçant de manière contrôlée sur un rail ou une piste.

4.12.2. Choix de Châssis mobile

Nous avons choisis un mode de déplacement sur rails, entraîné par une courroie dentée et un moteur pas à pas. Ce mode de déplacement à faible degrés de liberté offre plus de flexibilité du système, en rendant le manipulateur capable d'atteindre plus de points différents de l'espace de travail.

5. Conclusion

Dans ce chapitre, nous avons présenté l'ensemble des composants matériels utilisés pour la réalisation du bras manipulateur mobile. Chaque élément a été choisi avec soin en fonction de ses caractéristiques techniques, de sa compatibilité avec le système, et de son rôle dans le fonctionnement globale du projet. Cette sélection nous permet de garantir une structure fiable, modulable et adapté aux besoins de notre application.

Chapitre 03 : Simulation, réalisation et intégration du manipulateur mobile

1. Introduction

Dans ce chapitre, nous présentons la partie expérimentale de notre projet. Nous commençons par la simulation du manipulateur mobile à l'aide de la plate-forme Node-Red pour tester son fonctionnement avant de passer à la réalisation physique.

Ensuite, nous allons procéder à l'assemblage des différents composants matériels, comme le moteur, le driver, le châssis mobile et la carte Raspberry Pi. Dans cette étape, nous allons procéder à tester le fonctionnement global du système et de préparer la phase finale des tests.

2. Modélisation géométrique

Dans ce travail, nous allons étudier le modèle géométrique direct et inverse du bras. Cette analyse nous permettra d'obtenir les formules nécessaires à la programmation du

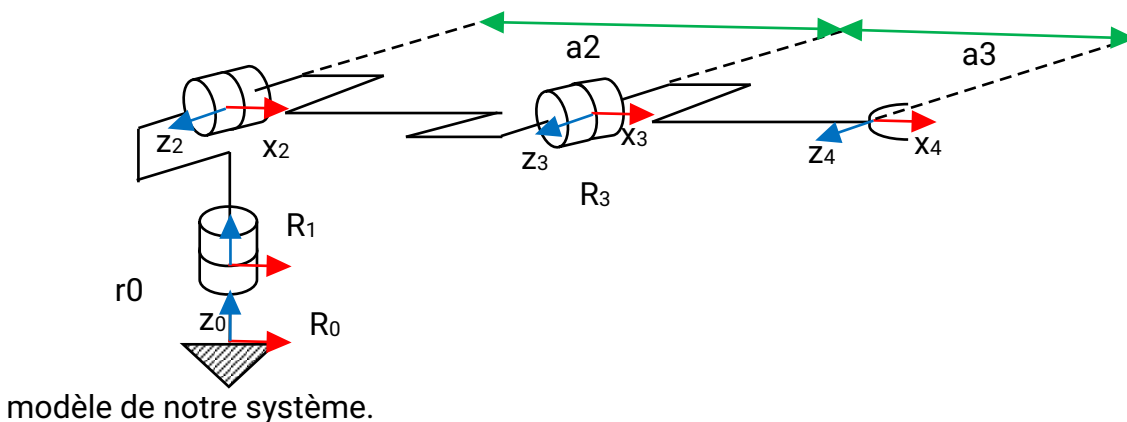


Figure 34. Position simplifiée du robot pour la modélisation

2.1. Modèle géométrique direct

Le tableau des paramètres de Denavit-Hartenberg (DH) est une méthode standardisée pour représenter les transformations entre les articulations successives d'un robot manipulateur.

Tableau 3. Paramètres de Denavit-Hartenberg

i	${}^0T^1$	${}^1T^2$	${}^2T^3$	${}^3T^4$
α_i	0	$\pi/2$	0	0
$(d_i) a_i$	0	0	a_2	a_3
θ_i	θ_1	θ_2	θ_3	0

r_i	r_0	0	0	0
-------	-------	---	---	---

1. Calculer les matrices de transformation homogènes ${}^0T_1; {}^1T_2; {}^2T_3; {}^3T_4$

$$2. {}^{i-1}T_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 & d_i \\ \cos \alpha_i \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i & -r_i \sin \alpha_i \\ \sin \alpha_i \sin \theta_i & \sin \alpha_i \cos \theta_i & \cos \alpha_i & r_i \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$r_0=11.5, \quad a_2=10, \quad a_3=10.8$$

$${}^0T^1 = \begin{bmatrix} c1 & -s1 & 0 & 0 \\ s1 & c1 & 0 & 0 \\ 0 & 0 & 1 & r_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad {}^1T^2 = \begin{bmatrix} c2 & -s2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s2 & c2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$${}^2T^3 = \begin{bmatrix} c3 & -s3 & 0 & a_2 \\ s3 & c3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad {}^3T^4 = \begin{bmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

3. Calcul du produit matriciel :

$${}^2T^4 = {}^2T^3 * {}^3T^4 = \begin{bmatrix} c3 & -s3 & 0 & a_3 c3 + a_2 \\ s3 & c3 & 0 & a_3 s3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T^4 = {}^1T^2 * {}^2T^4 = \begin{bmatrix} c23 & -s23 & 0 & a_3 c23 + a_2 c2 \\ 0 & 0 & -1 & 0 \\ s23 & c23 & 1 & a_3 s23 + a_2 s2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T^4 = {}^0T^1 * {}^1T^4 = \begin{bmatrix} c1 & -s1 & 0 & 0 \\ s1 & c1 & 0 & 0 \\ 0 & 0 & 1 & r_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} c23 & -s23 & 0 & a_3 c23 + a_2 c2 \\ 0 & 0 & -1 & 0 \\ s23 & c23 & 1 & a_3 s23 + a_2 s2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T^4 = \begin{bmatrix} c1 c23 & -c1 s23 & s1 & c1(a_3 c23 + a_2 c2) \\ s1 c23 & -s1 s23 & -c1 & s1(a_3 c23 + a_2 c2) \\ s23 & c23 & 1 & r_0 + a_3 s23 + a_2 s2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Matrice de passage 3x3 = $\begin{bmatrix} c1 c23 & -c1 s23 & s1 \\ s1 c23 & -s1 s23 & -c1 \\ s23 & c23 & 1 \end{bmatrix}$ qui exprime l'orientation de

l'effecteur par rapport au repère R_0

- Vecteur de position 3x1 = $\begin{bmatrix} Px = c1(10.8 c23 + 10 c2) \\ Py = s1(10.8 c23 + 10 c2) \\ Pz = 11.5 + 10.8 s23 + 10 s2 \end{bmatrix}$ qui exprime la position

de l'effecteur par rapport au repère R_0

Remarque:

$$C1 = \text{Cos}(\theta_1); \quad C23 = \text{Cos}(\theta_2 + \theta_3) \quad C23 = c2c3 - s2s3$$

$$S1 = \text{Sin}(\theta_1); \quad S23 = \text{Sin}(\theta_2 + \theta_3) \quad S23 = c2s3 + s2c3$$

4. Soit les coordonnées articulaires

$$\theta_1 = \pi/2; \quad \theta_2 = \pi/4; \quad \theta_3 = -\pi/4$$

$$a_2 = 2; \quad a_3 = 2$$

Le vecteur de position correspondant est donné par :

$$P_x = C(\pi/2)(\dots) = 0$$

$$P_y = S(\pi/2)(a_3 * c(\pi/4 - \pi/4) + a_2 * c(-\pi/4)) = 1 * (a_3 * 1 + a_2 * 0,71) = 3,42$$

$$P_z = r_0 + a_3 * s(\pi/4 - \pi/4) + a_2 * s(\pi/4) = r_0 + a_3 * 0 + a_2 * 0,71 = r_0 + 1,42$$

2.2. Modèle géométrique inverse :

Déterminer les valeurs des variables articulaires : $\begin{cases} \theta_1 = ? \\ \theta_2 = ? \\ \theta_3 = ? \end{cases}$

- Calculons θ_1 :

$$\left. \begin{matrix} Px = C1(a_3 C23 + a_2 C2) \\ Py = S1(a_3 C23 + a_2 C2) \end{matrix} \right\} \Rightarrow \frac{S1}{C1} = \frac{Py}{Px} \Rightarrow \theta_1 = \text{ATAN2}(Py, Px)$$

- Calcul de θ_3

Posons $r_0 = 0$

$$\left. \begin{aligned} \frac{P_x}{C_1} &= a_3 C_2 C_3 + a_2 C_2 \\ \frac{P_z}{C_1} &= a_3 S_2 C_3 + a_2 S_2 \end{aligned} \right\}$$

Équation de type 8 avec :

$$Z_1 = \frac{P_x}{C_1}; \quad Z_2 = P_z; \quad X = a_2; \quad Y = a_3$$

$$\text{Elevé au } \wedge^2, \Sigma \Rightarrow a_2^2 + a_3^2 + 2a_2 a_3 C_3 = \left(\frac{P_x}{C_1}\right)^2 + P_z^2 \Rightarrow C_3 = \frac{\left(\frac{P_x}{C_1}\right)^2 + P_z^2 - a_2^2 - a_3^2}{2a_2 a_3}$$

$$\theta_3 = \text{ATAN2}(\pm \sqrt{1 - C_3^2}, C_3)$$

- Calcul de θ_2

Faisons le développement du système d'équation (1)

$$\left\{ \begin{aligned} \frac{P_x}{C_1} &= a_2 C_2 + a_3 C_2 C_3 - a_3 S_2 S_3 \\ \frac{P_z}{C_1} &= a_2 S_2 + a_3 C_2 S_3 + a_3 S_2 C_3 \end{aligned} \right. \Rightarrow \left\{ \begin{aligned} \frac{P_x}{C_1} &= (a_2 + a_3 C_3) C_2 - a_3 S_3 S_2 \\ \frac{P_z}{C_1} &= (a_2 + a_3 C_3) S_2 + a_3 S_3 C_2 \end{aligned} \right.$$

Nous avons un système à deux équations à deux inconnus C_2 et S_2

Faisons un changement de variables : $B_1 = a_2 + a_3 C_3$, $B_2 = a_3 S_3$

$$\text{Le système devient : } \begin{cases} Z_1 = B_1 C_2 - B_2 S_2 \\ Z_2 = B_1 S_2 + B_2 C_2 \end{cases} \dots\dots \text{Equation de type 02}$$

$$S_2 = \frac{B_1 Z_2 - B_2 Z_1}{B_1^2 + B_2^2} \qquad \Rightarrow \qquad S_2 = \frac{(a_2 + a_3 C_3) * P_z - a_3 S_3 * \frac{P_x}{C_1}}{(a_2 + a_3 C_3)^2 + (a_3 S_3)^2}$$

$$C_2 = \frac{B_1 Z_1 - B_2 Z_2}{B_1^2 + B_2^2} \qquad \Rightarrow \qquad C_2 = \frac{(a_2 + a_3 C_3) * \frac{P_x}{C_1} + a_3 S_3 * P_z}{(a_2 + a_3 C_3)^2 + (a_3 S_3)^2}$$

$$\Rightarrow \theta_2 = \text{ATAN2}(S2, C2)$$



3. Programmation avec Node-Red

Dans cette section, nous allons explorer l'interface de Node-Red et définir les nœuds utilisés pour contrôler le manipulateur mobile. Chaque nœud sera expliqué en détail, en précisant son rôle et son paramétrage dans le système.

L'accès à l'interface de développement se fait en utilisant une page web à l'adresse url et le port suivants : `@ip-serveur-node-red :1880`, si le serveur est lancé sur un raspberry d'adresse 192.168.54.250, alors l'adresse d'accès sera :

`http://192.168.54.250:1880`

Sur raspberry, la commande de lancement du serveur Node-red est :

`$node-red`

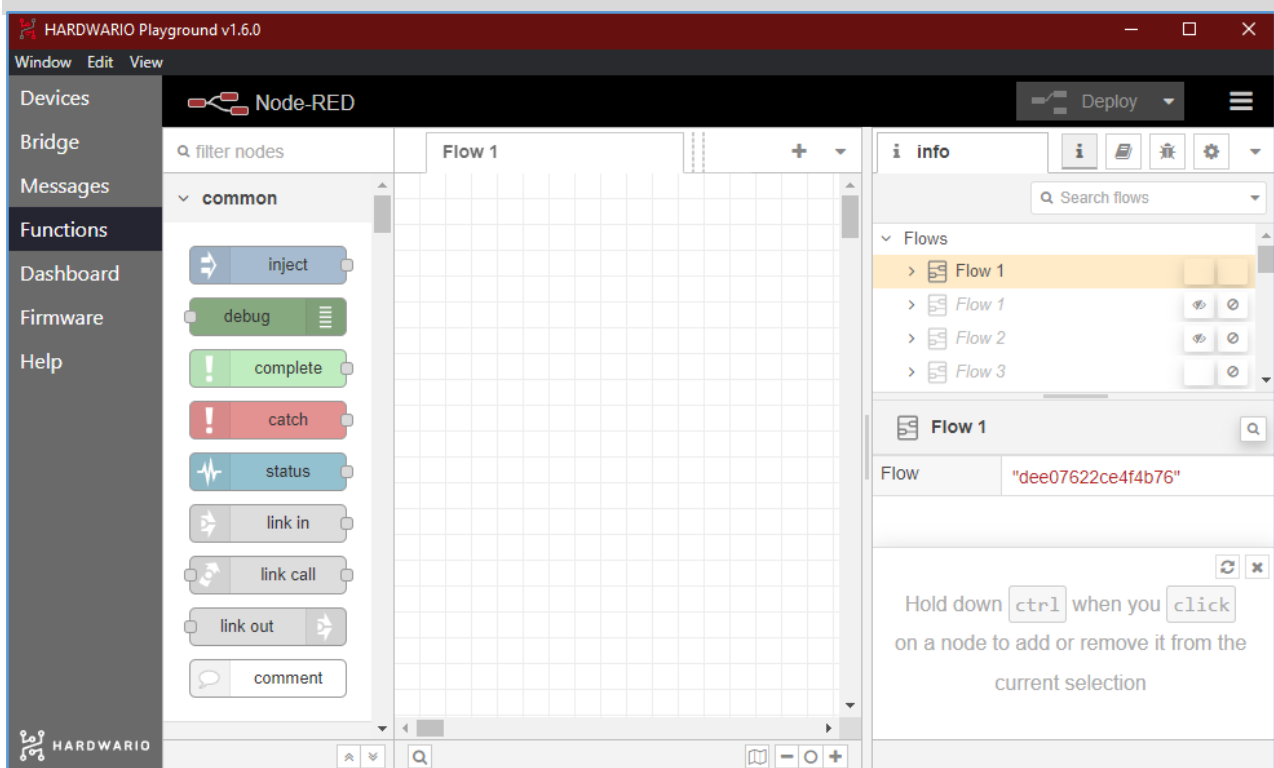


Figure 35. Interface graphique de Node-Red

- (1). **Inject**¹⁴ : Ce nœud permet d'injecter un message manuellement ou à intervalle régulier dans un flow Node-Red. Déclencher des actions automatiquement ou manuellement pour les tests.
- (2). **Fonction** : Permet d'écrire du code JavaScript personnalisé pour traiter les données. Manipuler les messages complexes.
- (3). **Debug** : Affiche le contenu des messages dans l'onglet Debug de l'interface. Vérification du contenu de msg.payload pour le débogage.
- (4). **Switch** : Route le message selon des conditions définies (ex : la valeur de msg.payload). Logique conditionnelle dans les flows.
- (5). **Change** : Permet de modifier des propriétés du message sans utiliser de code. Ajouter ou changer la valeur d'un champ dans msg.
- (6). **Template** : Crée un message à partir d'un modèle texte (HTML ou brut). Générer des réponses personnalisées ou des vues HTML.
- (7). **Delay**¹⁵ : Retarde ou limite la fréquence d'envoi des messages. Ajouter une temporisation ou éviter les flux excessifs.
- (8). **Dashbord(optionnels)**¹⁶ : Fournissent des composants graphiques pour créer des interfaces utilisateur dans Node-Red. Créer un tableau de bord interactif.

¹⁴ Node-RED Docs. «Fonction node», <https://nodered.org/docs/user-guide/nodes#fonction>

¹⁵ Node-RED Docs. «delay node», <https://nodered.org/user-guide/nodes#delay>

¹⁶ Node-RED Dashboard GitHub. «node-red-dashbord», <https://github.com/node-red/node-red-dashbord>

3.1. Interface de programmation de Châssis mobile

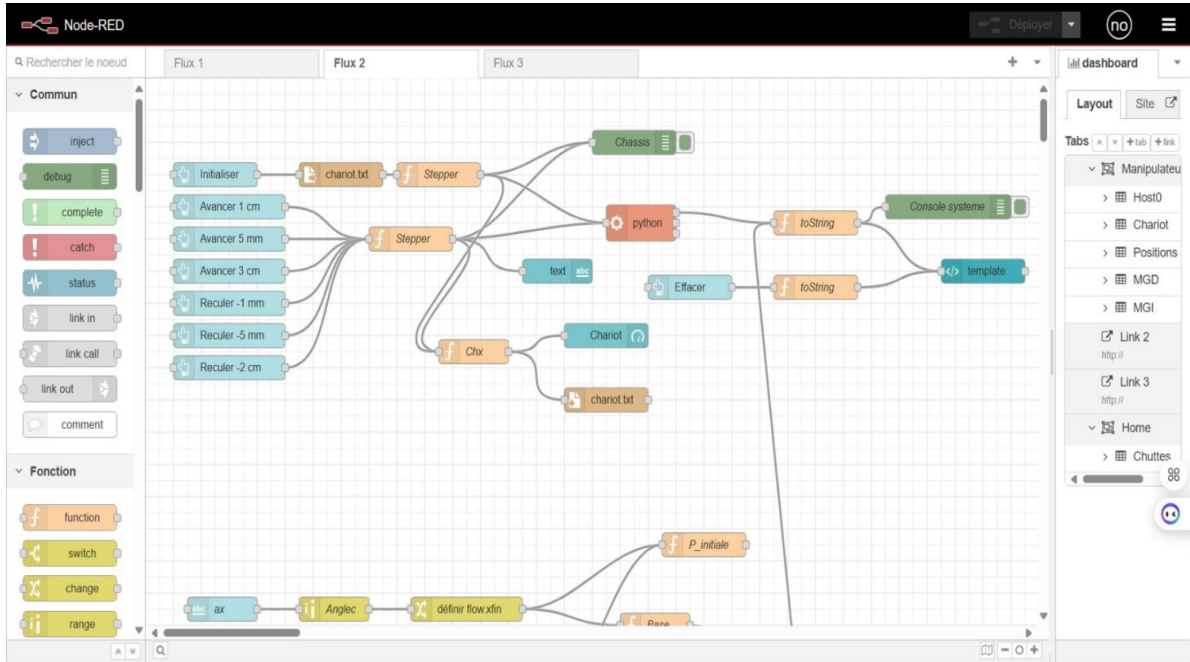


Figure 36. Interface de programmation de châssis mobile

Sur le plan fonctionnel, chaque commande envoyée depuis l'interface (par exemple Avancer 1cm) est transmise sous forme de message à une fonction qui traduit cette instruction en nombre de pas nécessaires pour le moteur pas à pas. Cette fonction applique un calcul basé sur la résolution du moteur et la mécanique du chariot. Ensuite, le message est redirigé vers un nœud Python, où un script exécute les ordres en activant les broches GPIO selon la séquence adéquate pour produire le déplacement voulu. Une fois l'action complétée, le résultat (comme la position actuelle ou un message de confirmation) est converti en texte puis affiché dans une console intégrée à l'interface. Ce processus permet un contrôle temps réel fiable, avec une traçabilité grâce à l'enregistrement des commandes dans un fichier texte.

3.2. L'interface de programmation de position de bras manipulateur

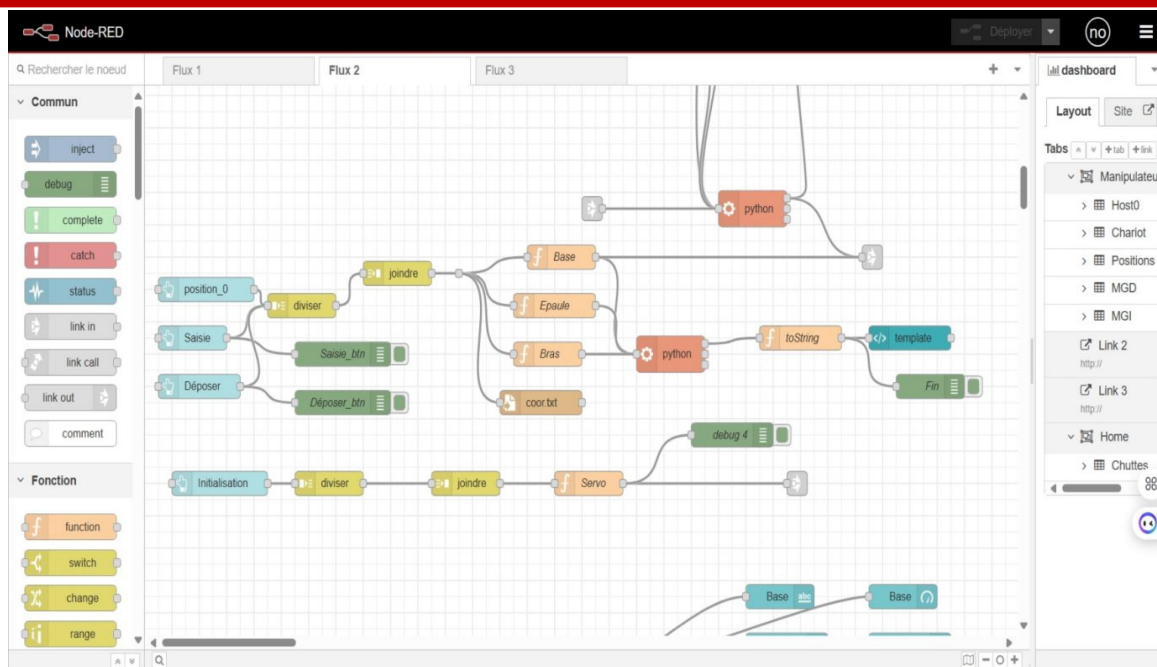
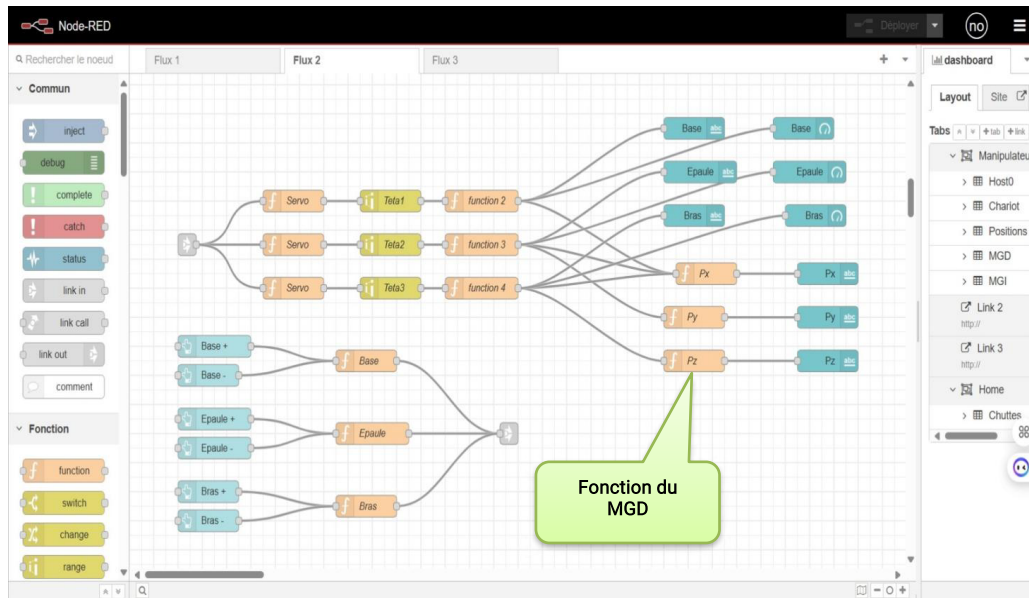


Figure 37. L'interface de programmation de position de bras manipulateur

Dans cette section, l'architecture du programme développé sous Node-Red permet d'assurer le traitement des commandes liées au mouvement du manipulateur. Le flux commence par des nœuds d'entrée tels que Saisie et Déposer, représentant les actions de l'utilisateur. Ces entrées sont ensuite traitées par des fonctions personnalisées comme diviser et joindre, servant à organiser les données en coordonnées exploitables. Trois fonctions principales, Base, Épaule et Bras, réalisent le calcul des angles nécessaires pour chaque articulation du bras. Les résultats sont ensuite transférés à un script Python via un nœud dédié, permettant l'envoi des commandes vers le matériel physique. Une sauvegarde des coordonnées est également effectuée dans un fichier texte (coor.txt) pour un usage ultérieur. Enfin, certaines données sont converties en chaîne de caractères à l'aide du nœud fonction (toString), puis préparées pour un affichage structuré. Cette organisation modulaire permet une gestion efficace, claire et évolutive du système de contrôle.

3.3. Interface de programmation de MGD du bras manipulateur



```

1 var a1=flow.get('a1');
2 var a2=flow.get('a2');
3 msg.payload = Math.cos(flow.get('teta1')) * (a1 * Math.cos(flow.get('teta2')) + a2 * Math.cos(flow.get('teta2') + flow.get('teta3')))
4 msg.payload=msg.payload.toFixed(2);
5 flow.set('px',msg.payload);
6 return msg;

```

Figure 38. Les interfaces de programmation MGD

Dans cette partie, nous avons implémenté le modèle géométrique direct (MGD) du bras manipulateur à trois articulations rotoïde. L'objectif est de déterminer la position de l'extrémité du bras (end-effector) dans l'espace tridimensionnel à partir des angles articulaires, et des longueurs des segments [$r_0=11.5$, $a_2=10$, $a_3=10.8$].

La programmation a été réalisée dans l'environnement Node-Red à l'aide de fonctions

JavaScript. Chaque coordonnée, a été calculée séparément selon les équations trigonométriques du MGD :

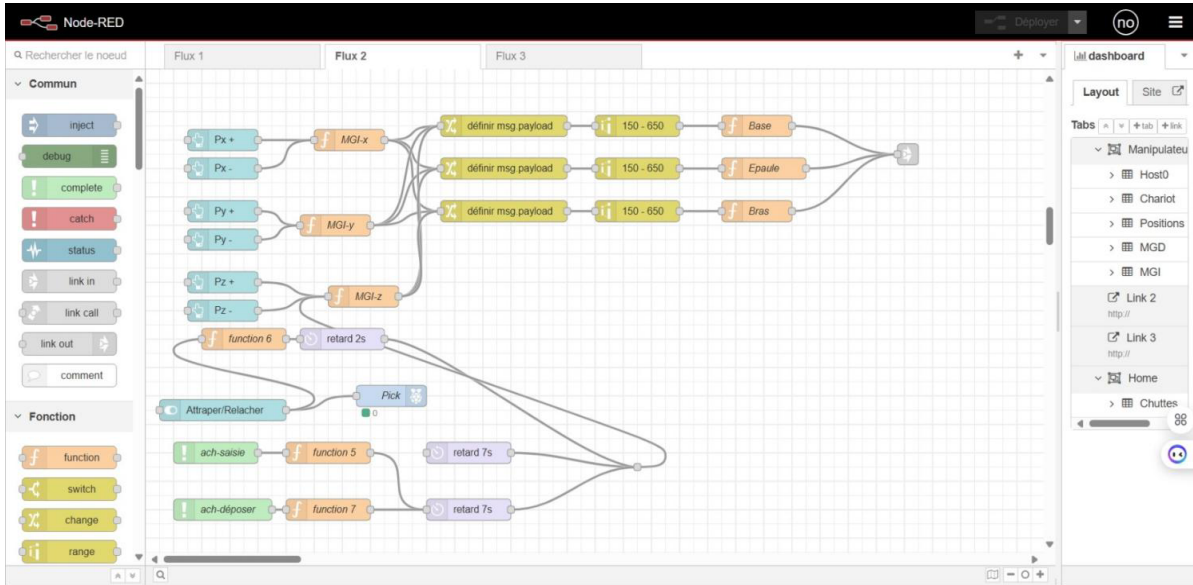
$$PX = \cos(\text{teta1}) * (a3 * \cos(\text{teta23}) + a2 * \cos(\text{teta2}))$$

$$PY = \sin(\text{teta1}) * (a3 * \cos(\text{teta23}) + a2 * \cos(\text{teta2}))$$

$$PZ = r_0 + a3 * \sin(\text{teta23}) + a2 * \sin(\text{teta2})$$

Les résultats sont ensuite stockés dans des variables de type float pour être utilisés dans les étapes suivantes du traitement.

3.4. Interface de programmation de MGI du bras manipulateur



Modifier le noeud function

Supprimer Annuler Terminer

Propriétés

Nom: MGI-x

Configurations: Au démarrage Message reçu À l'arrêt

```

1 var px = Number(flow.get('px'))+msg.payload;
2 var py = Number(flow.get('py'));
3 var pz = Number(flow.get('pz'))-11.5;
4
5 var a1 = flow.get('a1');
6 var a2 = flow.get('a2');
7 var teta1=Math.atan2(py,px);
8 var c3=(( px/Math.cos(teta1) )**2+(pz)**2-a1**2-a2**2 ) / (2*a1*a2);
9 var s3=-Math.sqrt(1-c3**2);
10 var teta3=Math.atan2(s3,c3);
11
12 var s2=((a1+a2*c3)*pz-a2*s3*px/Math.cos(teta1) )/((a1+a2*c3)**2+(a2*s3)**2);
13 var c2=((a1+a2*c3)*px/Math.cos(teta1)+a2*s3*pz )/((a1+a2*c3)**2+(a2*s3)**2);
14 var teta2=Math.atan2(s2,c2);
15
16 teta1=(teta1*180/Math.PI).toFixed(2);
17 teta2=(teta2*180/Math.PI).toFixed(2);
18 teta3=(teta3*180/Math.PI).toFixed(2);
19
20 if (!Array.isArray(msg.payload)) {
21     msg.payload = [];

```

Figure 39. Les interfaces de programmation MGI

Dans cette étape, nous avons implémenté le Modèle Géométrique Inverse (MGI) du manipulateur anthropomorphe à trois articulations rotoïde. L'objectif du MGI est de calculer les angles articulaires nécessaires ($\theta_1, \theta_2, \theta_3$) afin que l'extrémité du bras atteigne une position cible définie par les coordonnées cartésiennes (P_x, P_y, P_z).

La programmation dans Node-Red repose sur des nœuds de fonction qui contiennent les équations trigonométriques (*Math.cos()*) permettant de résoudre le MGI. Ces équations ont été dérivées à partir du modèle géométrique du bras robotisé, en tenant compte des longueurs des segments et de la configuration spatiale du manipulateur.

Les valeurs des angles calculés sont ensuite transmises aux servomoteurs correspondants, permettant ainsi au bras de se positionner correctement dans l'espace. Ce processus est essentiel pour le contrôle précis du manipulateur dans un environnement industriel ou expérimental.

Dans le flux MGI, après chaque commande de type Saisir ou Déposer, une élévation automatique du bras est effectuée. Concrètement, une fois que l'actionneur électromécanique exécute l'ordre '*Rattraper*' ou '*Relâcher*', le système envoie une nouvelle position en P_z , augmentée d'une valeur fixe (par exemple 2 cm), afin de relever le bras. Cette étape permet d'éviter tout contact involontaire avec l'objet ou la surface après la prise ou la dépose, elle est essentielle pour assurer un mouvement fluide et sécurisé du manipulateur.

4. Protocole I2C : Communication entre dispositifs électriques

Le protocole I2C (Inter-Integrated Circuit) est un protocole de communication série qui permet à plusieurs dispositifs électroniques de communiquer entre eux en utilisant seulement deux fils : un pour les données (SDA) et un autre pour l'horloge (SCL). Ce protocole est largement utilisé dans les microcontrôleurs et autres composants électroniques pour simplifier le câblage et permettre une communication rapide efficace [5].

Le bus I2C fonctionne selon une architecture maître-esclave, où le maître contrôle l'horloge et initie la communication, tandis que l'esclave répond aux requêtes envoyées. Chaque dispositif connecté sur le bus I2C possède une adresse unique, permettant au maître de communiquer avec un esclave spécifique.

Ce protocole est particulièrement avantageux pour des projets de robotique ou d'automatisation, où plusieurs composants doivent interagir entre eux. Par exemple, un Raspberry Pi peut contrôler un module PCA9685 pour gérer plusieurs servomoteurs via un seul bus I2C, simplifiant ainsi le câblage.

5. Protocole PWM : Modulation de largeur d'impulsion¹⁷

Le PWM (Pulse Width Modulation) est une technique utilisée pour générer un signal analogique à partir d'une source numérique. Elle consiste à faire varier la largeur des impulsions dans un signal périodique à fréquence constante. L'information est transmise non pas par la tension du signal, mais par la durée pendant laquelle le signal reste à l'état haut (duty cycle).

Dans le contexte du contrôle de servomoteurs, le PWM permet de définir la position de l'arbre du moteur. Typiquement, une impulsion de 1ms correspond à 0°, une impulsion de 1.5 ms à 90°, et une impulsion de 2ms à 180°, avec une fréquence de 50 Hz (soit une période de 20 ms).

Cette méthode est très utilisée en électronique embarquée pour/

- _ Contrôler la vitesse des moteurs à courant continu
- _ Réguler l'intensité lumineuse des LEDs
- _ Commander des servomoteurs de manière précise et économique

6. Connexion du module PCA9685 avec Raspberry Pi via I2C¹⁸

Pour commander plusieurs servomoteurs de manière efficace, nous avons utilisé un module PCA968, un contrôleur PWM 16 canaux, relié à un Raspberry Pi à travers le bus I2C. Ce choix permet de libérer les broches GPIO du Raspberry Pi tout en garantissant un contrôle précis des impulsions PWM nécessaires à l'impulsion PWM nécessaire au positionnement des servomoteurs.

Tableau 4. Connexion PCA9685 avec Raspberry Pi via I2C

Composant Raspberry Pi	Module PCA9685	Description
5v (pin 2 ou 4)	VCC	Alimentation du module
GND (pin 6)	GND	Masse commune
GPIO (Pin 3-SDA)	SDA	Données I2C
GPIO (Pin5-SCL)	SCL	Horloge I2C

Les servomoteurs sont ensuite connectés aux sorties du module :

- Fil signal sur PWMx

¹⁷ <https://www.geeksforsgeeks.org/pulse-width-modulation-pwm/>

¹⁸ <https://learn.adafruit.com/16-Channel-pwm-servo-driver>

- VCC (5V)
- GND
- Principe de fonctionnement

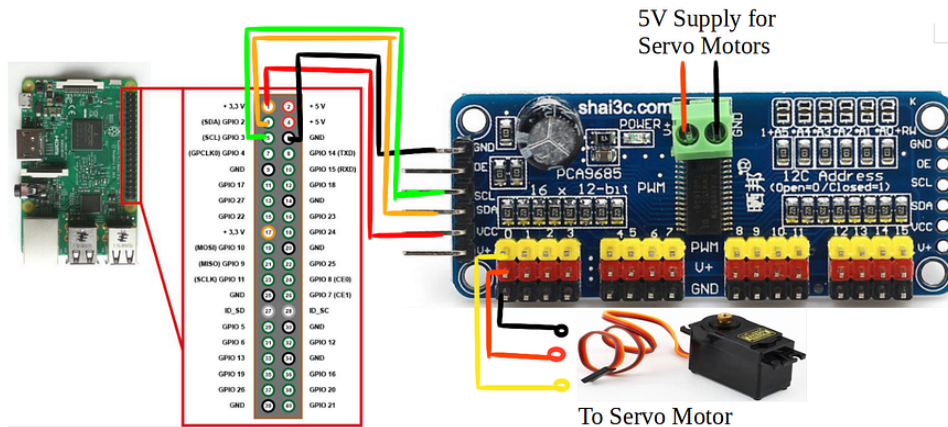


Figure 40. Connexion du module PCA9685 avec Raspberry Pi via I2C

Le Raspberry Pi agit comme maître I2C et envoie les instructions au module PCA9685, qui génère à son tour les signaux PWM appropriés pour chaque servomoteur. L'adresse par défaut du module est généralement 0x40, mais elle peut être modifiée selon les besoins via des cavaliers d'adressage.

7. Moteur pas à pas pour le déplacement du châssis mobile

Dans notre réalisation, le moteur NEMA 17 (JK42HS34-1334AC) a été utilisé pour faire déplacer le châssis mobile. Contrairement aux moteurs DC, ce moteur pas à pas permet un contrôle plus précis du mouvement grâce à son fonctionnement par pas. Cette précision est utile dans certaines applications où la régularité et le contrôle sont nécessaires.

Pour le commander, nous avons utilisé le driver L298N, qui permet de générer les séquences de base pour le faire fonctionner à basse vitesse.

7.1. Connexions établies

- GPIO 17(pins 11) → IN1(L298N)
- GPIO 27(pin 13) → IN2
- GPIO 22(pin 15) → IN3

- GPIO 23(pin 16) → GND(L298N)

Le moteur est connecté comme suite :

- Bobine A → OUT1 et OUT 2
- Bobine B → OUT3 et OUT4

L'alimentation du moteur est assurée par une source externe (12V) connectée à la borne VCC du L298N.

7.2. Fonctionnement

Le Raspberry Pi envoie une séquence de signaux aux entrées IN1 à IN4 pour créer une rotation progressive du moteur. Ce dernier étant monté sur le châssis, sa rotation provoque directement le mouvement du châssis mobile, ce qui nous a permis de le faire tourner dans les deux sens selon le besoin.

Même si cette méthode est expérimentale, elle est suffisante pour tester et démontrer la faisabilité du déplacement du châssis à l'aide d'un moteur pas-à-pas.

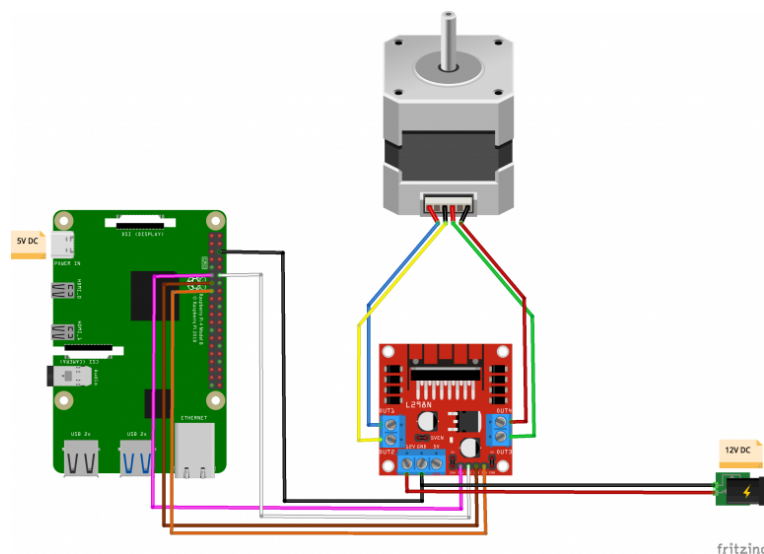


Figure 41. Connexion entre moteur NEMA JK 17 et Raspberry Pi & driver L298N¹⁹

8. Intégration du capteur HC-SR04 avec Raspberry Pi

Le capteur ultrasonique HC-SR04 a été choisi pour sa simplicité d'utilisation et sa capacité à mesurer des distances avec une précision satisfaisante pour le besoin de notre

¹⁹ <https://ben.akrin.com/driving-a-bipolar-stepper-motor-with-an-l298n-and-a-raspberry-pi/>

application. Pour l'intégrer au système embarqué, il a été connecté directement au Raspberry Pi via les ports GPIO [6] .

Le capteur comporte quatre broches : VCC, GND, TRIG et ECHO. Le branchement s'effectue de la manière suivante :

- VCC est connecté au 5V du Raspberry Pi.
- GND est relié à la masse.
- TRIG est connecté à une broche GPIO configurée comme entrée, avec une résistance de protection ou un diviseur de tension ($R1=1k\Omega$, $R2=2k\Omega$) pour abaisser le signal 5V à 3.3V compatible avec Raspberry Pi.

Le fonctionnement est assuré par un script en python utilisant des bibliothèques telles que

`RPI.GPIO` ou `GPIOZERO`, qui permettent de gérer le déclenchement du signal et la mesure du temps de retour. La distance est ensuite calculée à partir du temps de vol de l'onde ultrasonore.

8.1. Détection de la position de la base mobile par ultrasons

Dans notre réalisation, la base mobile (support du bras robotique) est susceptible de se déplacer par rapport au châssis mobile selon une trajectoire linéaire. Afin de connaître en temps réel sa position relative, un capteur à ultrasons HC-SR04 a été utilisé, fixé sur la face inférieure du châssis, orienté vers le bas.

Le principe consiste à mesurer la distance entre le capteur et une surface réfléchissante noire solidaire de la base mobile. En se déplaçant, la base modifie la distance qui sépare du capteur. Ainsi, chaque valeur mesurée par le HC-SR04 peut être associée à une position spécifique sur la trajectoire de la base. Cette méthode permet un suivi simple et sans contact de la position, tout en maintenant un coût réduit et une intégration facile.

L'acquisition périodique des données de distance permet au microcontrôleur de déduire la position actuelle de la base mobile via un système de correspondance entre les valeurs de distance et la position calibrée. Cette approche nécessite une phase initiale de calibrage, dans laquelle les distances sont mesurées pour différentes positions connues de la base, afin d'établir une table correspondance fiable.

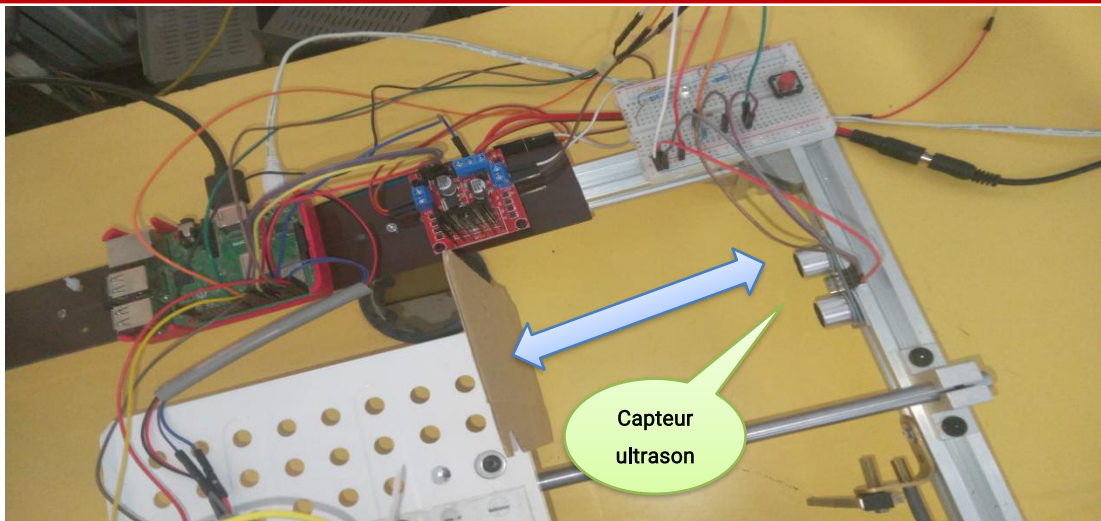


Figure 42. Capteur Ultrasons de mesure de déplacement

Vue réelle de l'implémentation du système afin de valider la faisabilité de notre système, plusieurs tests ont été réalisés en laboratoire. Les images suivantes illustrent le montage physique des composants ainsi que leur intégration avec l'interface Node-Red.

8.2. Interface Dashboard – Contrôle avancé du châssis mobile et du bras manipulateur

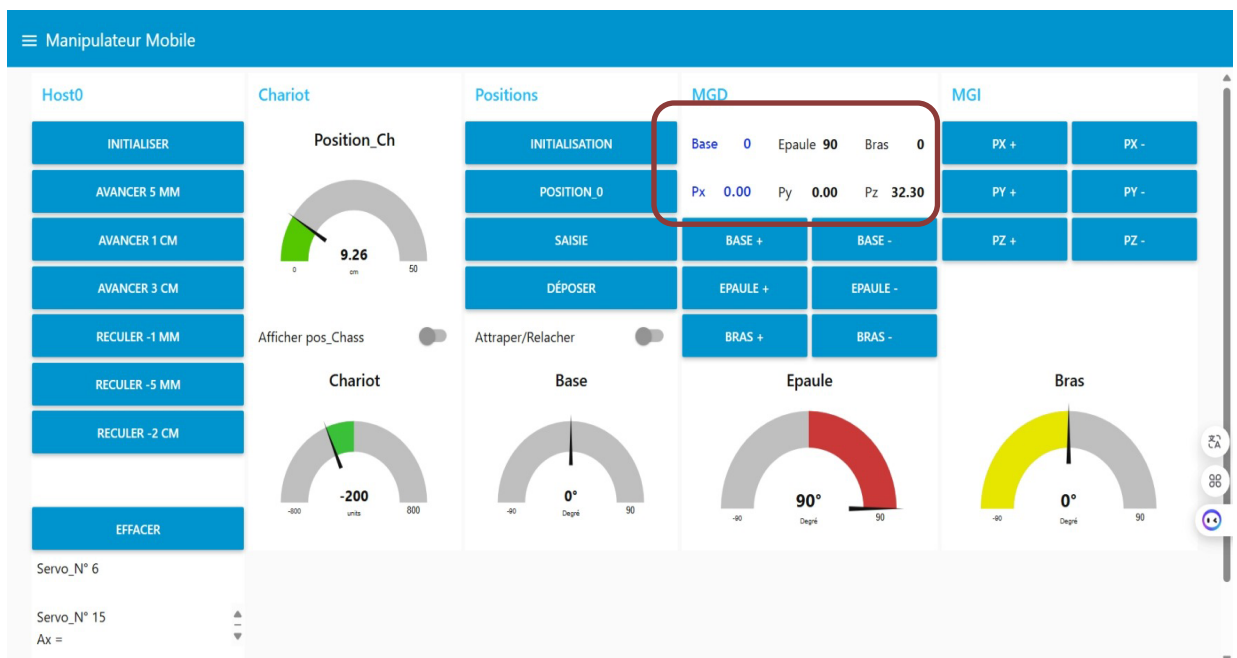


Figure 43. Interface de Dashboard dans la position0

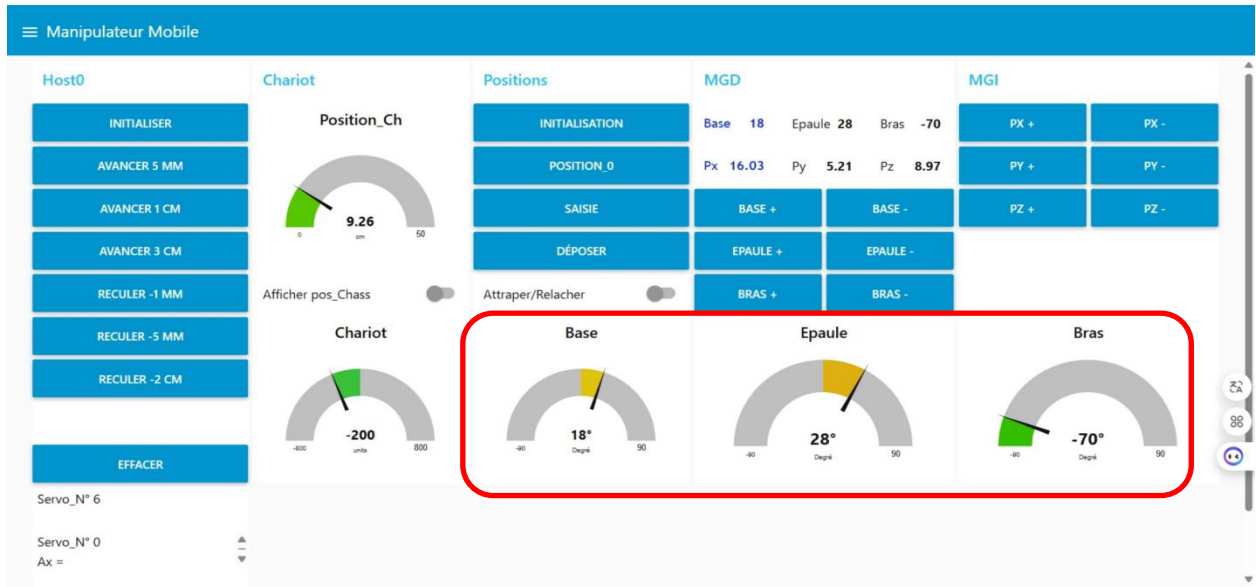


Figure 44. Interface de Dashboard dans la position de saisie

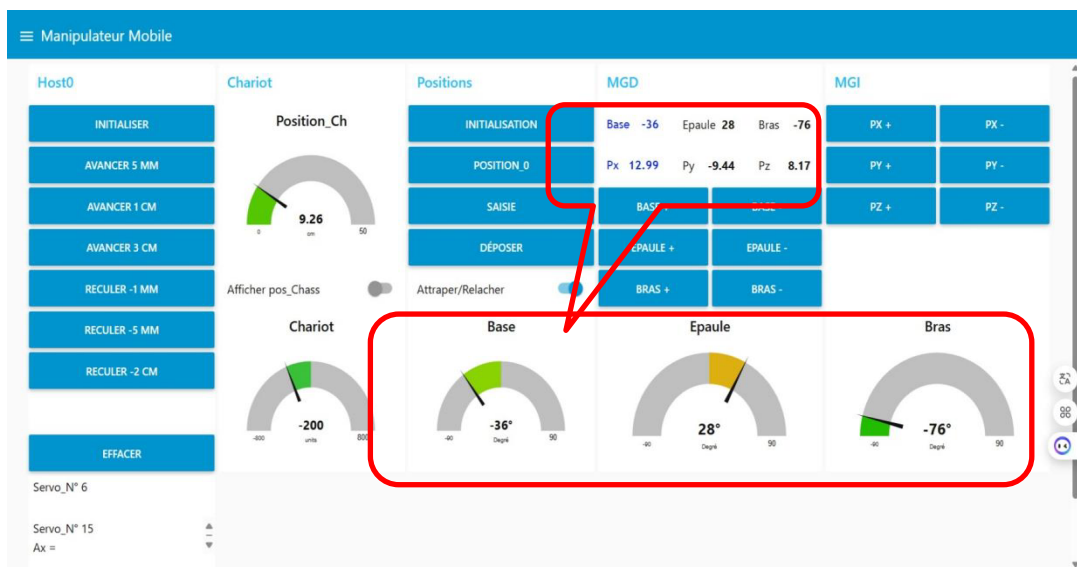


Figure 45. Interface de Dashboard dans la position de déposer

L'interface Dashboard développée avec Node-Red assure un pilotage complet du châssis mobile et du bras manipulateur à trois degrés de liberté, via des boutons interactifs accessibles à l'utilisateur.

8.3. Déplacement du châssis mobile

Le système de déplacement est basé sur un module de puissance L298N, connecté directement au Raspberry Pi, sans passer par des entrées externes comme un joystick ou des capteurs.

Toutes les commandes sont envoyées via l'interface utilisateur de Node-Red, ce qui simplifie le contrôle de la plateforme matérielle.

Boutons de déplacement :



Avancer de 1 cm, 3 cm, ou plus selon les besoins,



Reculer sur commande, et d'effectuer d'autres

mouvements définis.

Les signaux sont générés (function) qui font appel à un nœud (Exec) pour lancer un script python qui définit l'état des broches GPIO



par des nœuds inject associés à des script js (function) qui font appel à un nœud (Exec) pour lancer un script python qui définit l'état des broches GPIO pilotant les entrées IN1, IN2, IN3, IN4 du L298N.



Le bras est contrôlé selon deux modes :

Par positions prédéfinies (Position0, Saisir, Déposer),

Par coordonnées cartésiennes (MGI) grâce aux boutons PX±, PY±, PZ±, qui recalculent automatiquement les angles nécessaires.



Un autre mode de contrôle permet d'agir directement sur les articulations via des boutons Base±, Épaule±, Bras±, basés sur le modèle géométrique direct



Des jauges intégrées dans le Dashboard indiquent en temps réel :

La position du bras (X, Y, Z), l'état angulaire de chaque articulation, et les commandes de déplacement exécutées par le châssis.

Cette interface logicielle intuitive, permet un contrôle complet du robot en ligne à travers une page web en utilisant l'adresse IP du Node-red et le port : 1880.

Exemple : <http://192.168.16.145:1880/ui>

9. Explication de désigne de bras manipulateur

La figure suivante illustre l'enchaînement des articulations rotoïde formant la base (theta1), l'épaule (theta2) et le bras (theta3).

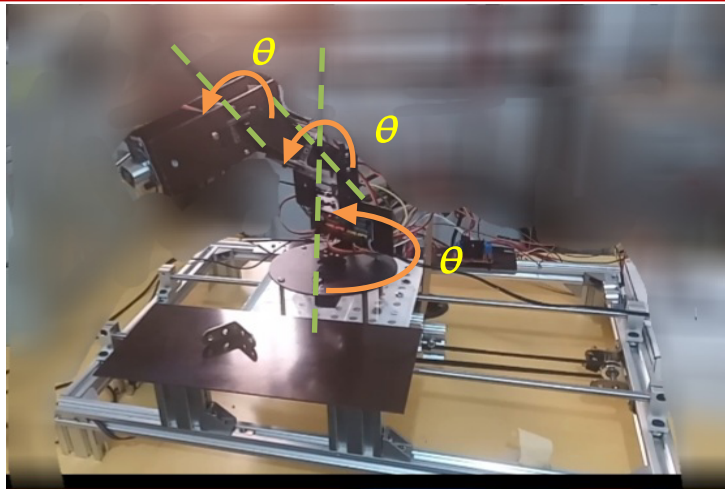


Figure 46. Bras manipulateur + châssis mobile

Chapitre 04 : Résultats obtenus et Discussions

1. Introduction

Dans ce chapitre, nous allons présenter les résultats obtenus à travers les tests du manipulateur mobile connecté. Nous expliquerons le type de bras utilisé et comment il se déplace avec le châssis mobile. Nous aborderons également le contrôle des mouvements via l'interface de commande et discuterons des diverses applications possibles du système.

Enfin, nous explorerons les améliorations et les extensions que nous pourrions apporter à ce système, ainsi que les différentes possibilités d'utilisation dans divers secteurs industriels et technologiques.

2. Résultats obtenus concernant le bras du manipulateur

Le bras obtenu est composé de plusieurs éléments essentiels. On y trouve notamment des moteurs qui assurent les mouvements articulés, une structure légère facilitant la manipulation, et des connexions électriques permettant de transmettre les commandes. Ces composants sont assemblés de manière à garantir un bon équilibre entre performance et simplicité.

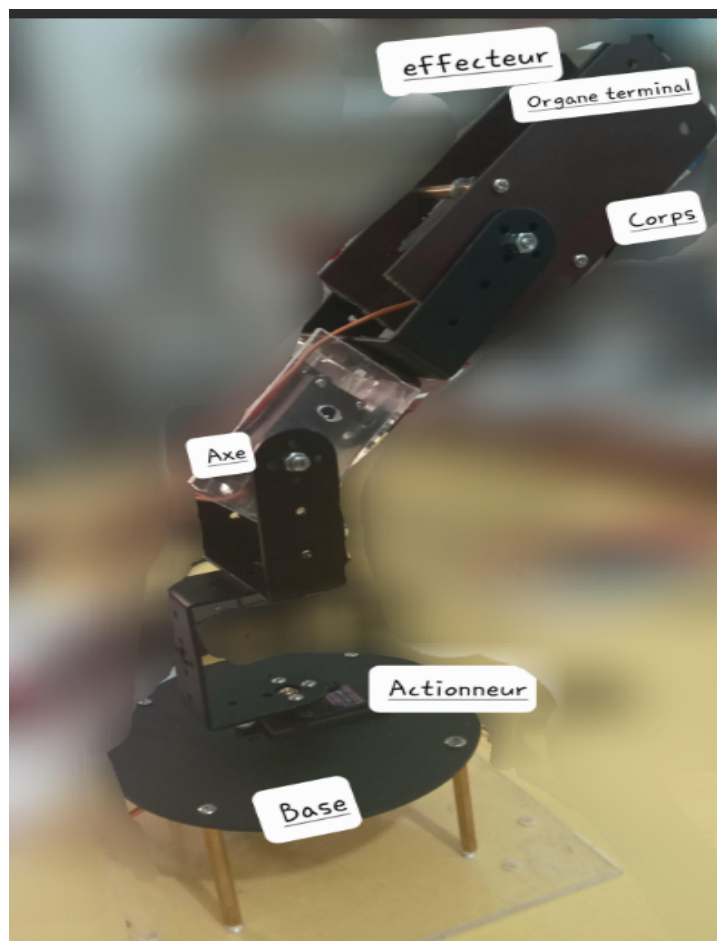


Figure 47. Éléments du bras manipulateur

Le bras manipulateur obtenu est de type anthropomorphe est un bras robotisé dont la structure et les mouvements s'inspirent de l'anatomie humaine, généralement composé de segments et d'articulations simulant ceux du bras humain (épaule, coude, poignet). Ce type de robot permet une interaction plus naturelle avec l'environnement, notamment dans les applications industrielles, médicales ou collaboratives [7].

3. Détails techniques sur le fonctionnement du bras

Le bras fonctionne grâce à l'activation coordonnée de plusieurs servomoteurs, chacun étant associé à une articulation spécifique (épaule, coude, poignet, etc.). Chaque servomoteur est programmé pour effectuer une rotation de 0 à 180 degrés, ce qui permet d'assurer des mouvements fluides et précis à chaque niveau du bras.

Chaque moteur est contrôlé indépendamment via l'interface Node-red (Dashboard), recevant des instructions calculées à l'avance en fonction des besoins. Cela garantit que chaque angle de rotation est défini avec précision et reproductible de manière fiable.

La structure du bras a été conçue avec soin, en utilisant des matériaux légers afin de ne pas surcharger les servomoteurs. Ce choix permet de maintenir un bon équilibre entre performance et durabilité, tout en assurant une réactivité optimale des moteurs sans échauffement excessif.

L'ensemble du système est calibré pour éviter les interférences entre les mouvements, assurant une synchronisation parfaite. Grâce à cette précision, le bras peut accomplir des tâches comme la saisie, la levée ou la rotation d'objets avec exactitude et stabilité.

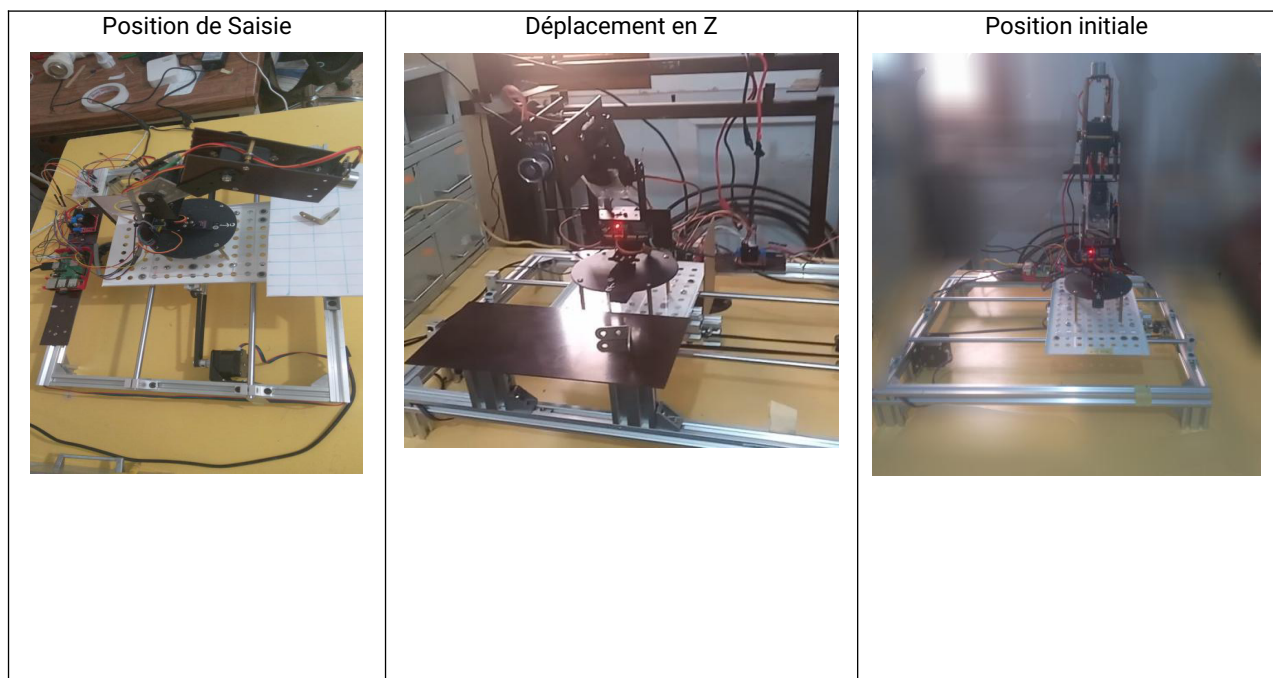


Figure 48. Différentes positions de bras

4. Description détaillée du châssis mobile

Le châssis mobile est conçu comme une structure mécanique assurant le déplacement précis d'un bras manipulateur le long d'un rail guidé. Il est équipé d'un seul moteur central, qui transmet le mouvement aux roues latérales à l'aide d'une courroie d'entraînement, permettant ainsi un déplacement fluide et équilibré.

Aux extrémités du rail, des capteurs de fin de course sont installés afin de détecter l'arrivée du châssis aux limites, ce qui permet l'arrêt automatique ou le changement de direction.

Le bras est fixé sur une base mobile soutenue par deux rails. Assurant un guidage précis et stable en utilisant des roulements. Le manipulateur ainsi que sa base ont été conçus pour être légers, ce qui facilite la tâche du moteur en réduisant la charge à déplacer et en assurant un mouvement sans effort excessif.

Sous la base, une surface noire est placée en face du capteur HC-SR04 monté en bas du châssis. Ce capteur à ultrasons permet de déterminer avec précision la position de la base le long du rail, garantissant un suivi en temps réel du mouvement.

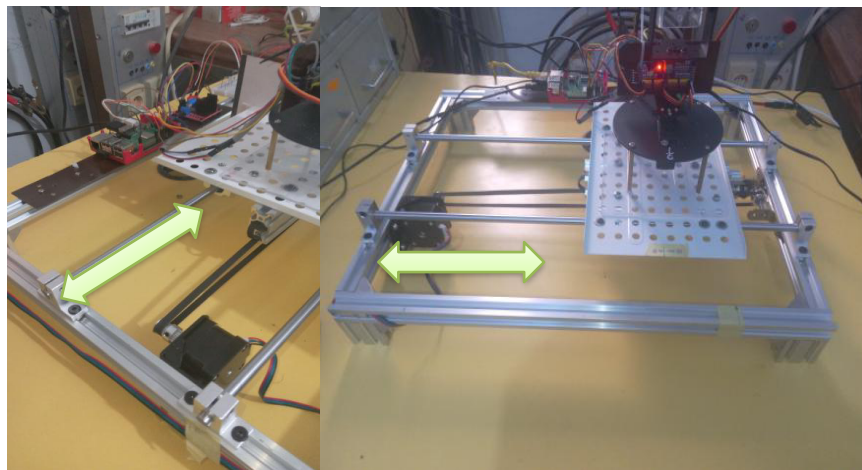


Figure 49. Châssis mobile sur rails

5. Contrôle du système à l'aide d'une interface Dashboard connecté

Dans le cadre de ce projet, le système a été piloté de façon simple et intuitive grâce à une interface appelée Dashboard, intégrée dans la plat forme Node-Red.

L'interface Dashboard développée dans ce projet comprend plusieurs boutons permettant de contrôler les mouvements du bras manipulateur et du châssis mobile. Chaque bouton a été conçu pour remplir une tâche précise, facilitant ainsi l'interaction avec le système.

6. Gestion synchronisée du bras et de sa base mobile

Dans le cadre de la validation du système, une attention particulière a été accordée à la coordination entre les différents éléments principaux du projet. Le bras manipulateur, monté base mobile légère, est conçu de manière à garantir un équilibre optimal pour ne pas surcharger le moteur responsable du déplacement châssis ; ce dernier assure le positionnement précis du bras sur la zone de travail.

L'interface de commande, réalisée via le Dashboard de Node-Red, permet à l'utilisateur de contrôler séparément ou successivement les actions du châssis et du bras. Le système répond fidèlement aux instructions envoyées : le châssis s'arrête avant toute manipulation du bras, et chaque mouvement de celui est exécuté selon des angles prédéfinis. L'utilisation de capteurs de fin de course garantit des limites physiques de sécurité, renforçant la précision et la fiabilité du fonctionnement global.

7. Gestion de la connectivité Réseau et contrôle à distance

Dans notre projet, la connexion stable et sécurisée entre les différents composants du système, notamment le Raspberry Pi, est essentielle pour garantir un contrôle à distance efficace. Pour ce faire, nous avons utilisé plusieurs outils et techniques pour assurer une communication fiable et sans interruption, en exploitant la technologie Wi-fi pour le transfert de données.

7.1. Connexion Wi-fi du Raspberry Pi

Le Raspberry Pi est connecté au réseau via wi-fi, ce qui permet de le relier à d'autres composants et de permettre un contrôle à distance via la plateforme Node-Red.

Cependant, une difficulté majeure réside dans le fait que l'adresse IP du Raspberry Pi est généralement attribuée dynamiquement par le routeur. Cela signifie que l'IP locale peut changer à chaque redémarrage du réseau, compliquant ainsi l'accès constant à l'interface de contrôle.

7.2. Utilisation de Advanced IP Scanner

Pour résoudre ce problème, nous avons intégré l'utilisation du logiciel Advanced IP Scanner. Cet outil permet de scanner rapidement le réseau local et de détecter tous les appareils connectés, y compris leur adresse IP. Ainsi, chaque fois que le Raspberry Pi

redémarre, nous avons pouvons facilement retrouver son adresse IP et nous assurer que l'accès à l'interface Node-Red reste possible. Cela permet un contrôle continu et gestion efficace du système sans avoir à configurer un IP statique.

7.3. Accès à distance avec Bitvise SSh client

Une fois l'adresse IP identifiée, nous avons utilisée Bitvise SSH client pour établir une connexion SSH sécurisée entre notre ordinateur et le Raspberry Pi. Bitvise est un client SSH qui permet d'accéder au terminal du Raspberry Pi à distance via un réseau local. Cette connexion nous permet d'exécuter des commandes, de surveiller les processus, de mettre à jour le système et de transférer des fichiers, le tout sans besoin d'un écran, d'un clavier ou d'une souris connectés directement au Raspberry Pi.

Bitvise offre également une fonctionnalité de transfert de fichiers via SFTP (Secure file transfert protocole), ce qui permet de transférer facilement des scripts ou des fichiers de configuration entre l'ordinateur et le Raspberry Pi. Cela simplifie la gestion du système et le prototypage rapide, car nous pouvons ajuster les paramètres et tester de nouvelles fonctionnalités à distance, directement depuis notre Pc.

7.4. Contrôle du Bras Manipulateur via le Dashbord

L'accès à Node-Red et la gestion du bras manipulateur se fait principalement via le Dashbord accessible depuis n'importe quel appareil connecté au même réseau Wi-fi. Grâce à cette connexion, nous pouvons envoyer des commandes en temps réel pour contrôler la position et les mouvements du bras manipulateur, tout en assurant une surveillance continue du système.

Toutefois, il est important de noter que le contrôle à distance est limité à la portée du réseau Wi-fi local. Si nous souhaitons étendre cette portée et permettre un contrôle à distance depuis l'extérieur du réseau local, des solutions avancées comme le VPN ou le port forwarding devraient être mises en place.

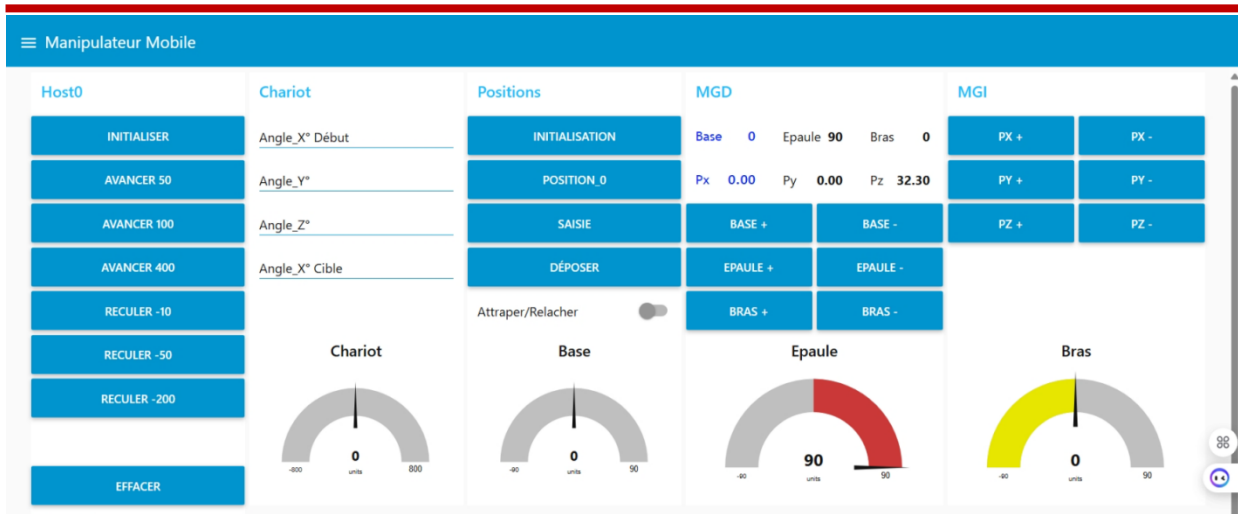


Figure 50. Dashboard de commande

8. Connexion et complémentarité entre les éléments du système

Ce projet réunit trois composants clés : un bras manipulateur léger, un châssis mobile précis, et une interface de commande intuitive. Ces éléments interagissent de manière fluide pour accomplir tâches ciblées. Le bras est monté sur une base mobile bien adaptée à son poids, évitant toute surcharge. Le châssis se déplace sur des rails grâce à un moteur central, assisté par des capteurs de position. L'ensemble est piloté via une interface Dashboard, assurant une communication claire entre les commandes de l'utilisateur et les réponses mécaniques du système.

9. Perspectives d'évolution et applications potentielles

Le système développé dans ce projet constitue une base modulaire et évolutive, capable d'être adaptée à une variété d'applications industrielles ou de services. Grâce à sa structure mobile et à son bras manipulateur, il peut être envisagé comme un outil pour des tâches de surveillance en temps réel, notamment à travers l'intégration de caméras motorisées utilisées dans les plages, les zones industrielles ou les environnements sensibles. Le bras peut alors servir à orienter la caméra ou à déclencher des actions ciblées à distance.

En milieu industriel, le système pourrait être transformé en une plate-forme de tri automatisé, capable de distinguer et de manipuler différents objets à l'aide de capteurs visuels (comme des caméras ou capteurs de couleurs), pour trier les produits finis, les

déchets ou les matériaux recyclables. En adaptant la puissance des moteurs et la robustesse des matériaux, il pourrait également être utilisé dans des applications plus lourdes telles que le déplacement d'objets dans un entrepôt ou l'assistance dans des tâches logistiques.

D'autres perspectives d'évolution incluent l'ajout d'une intelligence artificielle embarquée, permettant au système de prendre des décisions de manière autonome selon les données des capteurs. L'utilisation de la connectivité (Wi-fi, Bluetooth, 4G) ouvrirait aussi la voie à un contrôle à distance complet, transformant ce système en un agent robotique mobile, interactif, et intelligent.

Ainsi, ce prototype n'est qu'une première étape vers des solutions plus avancées, et sa modularité permet de l'adapter à des usages très variés, qu'il s'agisse de l'éducation, de la recherche, de l'industrie ou même de l'assistance personnelle.

10. Conclusion

Dans ce chapitre, nous avons présenté et analysé les résultats obtenus suite à la réalisation complète du projet. Les essais effectués ont démontré l'efficacité du système proposé, aussi bien au niveau du déplacement du châssis mobile que de la réactivité du bras robotique face aux commandes émises via l'interface de contrôle. L'utilisation de plateforme Node-Red s'est révélée être un choix judicieux, offrant à la fois souplesse de programmation et simplicité d'interaction. Ces résultats confirment la pertinence des choix techniques effectués et ouvrent la voie à de potentielles améliorations et extensions fonctionnelles dans des contextes industriels variés.

Conclusion général

En conclusion, ce travail de mémoire a permis de concrétiser un projet ambitieux et novateur qui vise à intégrer les technologies de la robotique mobile dans un cadre industriel. Le projet a débuté par une étude approfondie des systèmes existants et des défis actuels liés à la robotique mobile dans l'industrie. Grâce à une analyse rigoureuse et à la sélection minutieuse des composants matériels et logiciels, nous avons réussi à concevoir un manipulateur mobile contrôlé à distance via une interface intuitive, réalisée avec la plateforme Node-Red.

Le développement du système a suivi une approche méthodique qui a combiné la théorie et la pratique. À travers la mise en place de chaque étape, du choix du matériel jusqu'à la programmation et la validation des résultats, nous avons pu démontrer la faisabilité et l'efficacité du projet.

L'utilisation de Node-Red s'est avérée être un choix stratégique, car elle a permis d'assurer une flexibilité et une interactivité optimales dans le contrôle du robot. La réalisation du manipulateur mobile a également offert un aperçu concret de la manière dont des technologies modernes, telles que l'internet des objets (IoT) et l'automatisation industrielle, peuvent transformer les processus de production et de gestion dans les usines 4.0.

Les résultats obtenus à travers les tests pratiques ont confirmé que le système répondait aux attentes définies au début du projet. Toutefois, bien que le prototype actuel soit fonctionnel et offre des performances satisfaisantes, ce projet ouvre la voie à de nombreuses améliorations possibles, tant sur le plan de l'optimisation de la précision des mouvements du robot que sur l'enrichissement des fonctionnalités de l'interface de contrôle.

Enfin, ce travail n'a pas seulement permis de développer un prototype fonctionnel, mais il a également constitué une source d'apprentissage et d'expérimentation précieuse. Il a mis en lumière les défis techniques associés à la robotique mobile, tout en soulignant l'importance de l'intégration entre matériel et logiciel pour obtenir des systèmes efficaces et performants. Nous espérons que cette étude contribuera à la compréhension et à l'évolution des systèmes robotiques dans l'industrie, et qu'elle servira de base solide pour des travaux futurs qui exploiteront les technologies émergentes pour répondre aux besoins croissants de l'industrie 4.0.

Références bibliographiques

- [1]. Staal, A.S.,Salvatierra, C.G.,Albertson, D.D.,Mahendram, M.,Ravichandran,R.(2020). Towards a collaborative omnidirectional mobile robot in a smart cyber-physical environment. Sensors, 20(1) ,193. [https : //doi.org/10.3390/s20010193](https://doi.org/10.3390/s20010193)
- [2]. Lauar, A. & Korichi,K.(2016). Un système de recommandation pour l'assistance à la navigation sur l'Internet (Mémoire de master académique, Université Kasdi Merbah-ouargla,Algérie). Faculté des nouvelles Technologies de l'information et de la Communication.
- [3]. Bouakline, N&Benchaba,A(2018).conception et réalisation d'une Maison intelligente via Rasperry Pi 3,Electronique industrielle, Mouloud Mammeri de Tizi Ouzou.
- [4]. Ait Mohammeds, W. (2018). Conception et Réalisation d'un Radar de Recul Automobile à base d'Arduino et Processing, Mémoire de Master professionnel, Université Mouloud Mammeri de Tizi-Ouzou.
- [5]. UM10204, I²C-bus specification and user manual, Rev. 7.0 – 1 October 2021
- [6]. M.Rashid, Introduction to Mechatronics and Meaurement Systems, McGraw-Hill, 2011
- [7]. Siciliano, B.,&Khatib, o.(2016).Springer Handbook of Robotics. Springer. (chapitre4-robot Arms and Manipulators)

ملخص

يقدم هذا البحث تصميم وتنفيذ ذراع روبوتي متحرك متصل، يُمكن التحكم فيه عن بُعد من خلال بنية تعتمد على جهاز Raspberry Pi وواجهة برمجة رسومية Node-RED. يتكون النظام المطور من هيكل متحرك على قضبان مزود بمحرك يتيح حركة دقيقة وموثوقة، بالإضافة إلى ذراع روبوتي قابل للبرمجة قادر على تنفيذ مهام متنوعة في التعامل مع الأشياء. لضمان تفاعل فعال مع البيئة المحيطة، تم تجهيز الجهاز بعدة حساسات، منها حساسات للمسافة والموقع، التي توفر بيانات في الوقت الحقيقي. تعالج هذه المعلومات بواسطة Raspberry Pi، الذي يلعب دور المتحكم المركزي، حيث ينسق بين المحركات والذراع الروبوتي والحساسات. يُدار كامل النظام عبر واجهة ويب (لوحة تحكم) تم تطويرها باستخدام Node-RED، مما يسمح بالتحكم عن بُعد بطريقة بسيطة ومرنة، ومتاحة من أي جهاز متصل بالإنترنت. تساعد مرونة النظام وقدرته على الاتصال في تكييفه مع سيناريوهات تطبيق مختلفة، بدءاً من الصيانة وصولاً إلى التعامل مع الأجسام في مناطق يصعب الوصول إليها. كما يسلط هذا العمل الضوء على التحديات المتعلقة بالتكامل بين المكونات المادية والبرمجية، والحلول المعتمدة لضمان موثوقية وسرعة استجابة النظام. وأخيراً، تفتح هذه الدراسة آفاقاً لتطوير روبوتات متنقلة ذكية تساهم في التحول الرقمي للصناعات.

Résumé

Ce mémoire présente la conception et la réalisation d'un manipulateur mobile connecté, destiné à être contrôlé à distance grâce à une architecture intégrant un Raspberry Pi et une interface de programmation graphique Node-RED. Le système développé se compose d'un châssis mobile sur rails équipé d'un moteur permettant une mobilité fiable et précise, ainsi que d'un bras robotisé programmable capable d'exécuter diverses tâches manipulatoires. Pour assurer une interaction efficace avec son environnement, le dispositif est doté de plusieurs capteurs, notamment des capteurs de distance et de position, qui fournissent des données en temps réel. Ces informations sont traitées par le Raspberry Pi, le rôle de contrôleur central, assurant la coordination entre les moteurs, le bras robotisé et les capteurs. L'ensemble du système est piloté via une interface web (Dashboard), développée avec Node-RED, permettant un contrôle à distance simple et flexible, accessible depuis n'importe quel terminal connecté. La modularité et la connectivité de la plateforme facilitent son adaptation à divers scénarios d'application, allant de la maintenance à la manipulation d'objets dans des zones difficiles d'accès. Ce travail met également en lumière les défis liés à l'intégration matérielle et logicielle, ainsi que les solutions adoptées pour garantir la fiabilité et la réactivité du système. Enfin, cette réalisation ouvre des perspectives pour le développement de robots mobiles intelligents, contribuant à la transformation numérique des industries.

Summary

This thesis presents the design and implementation of a connected mobile manipulator, remotely controlled through an architecture integrating a Raspberry Pi and a graphical programming interface, Node-RED. The developed system consists of a mobile chassis on rails equipped with a motor that enables reliable and precise mobility, as well as a programmable robotic arm capable of performing various manipulation tasks. To ensure effective interaction with its environment, the device is equipped with several sensors, including distance and position sensors, which provide real-time data. The Raspberry Pi process information and serves as the central controller, coordinating the motors, the robotic arm, and the sensors. The entire system is operated via a web interface (dashboard) developed with Node-RED, allowing simple and flexible remote control accessible from any connected terminal. The platform's modularity and connectivity facilitate its adaptation to various application scenarios, ranging from maintenance to handling objects in hard-to-reach areas. This work also highlights the challenges related to hardware and software integration, as well as the solutions adopted to ensure system reliability and responsiveness. Finally, this achievement opens up prospects for the development of intelligent mobile robots, contributing to the digital transformation of

industries.

