الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DE TECHNOLOGIE

Département du Génie Électrique



Projet de fin d'études

Présenté pour l'obtention du
Diplôme De Master Automatique

Spécialité : Automatique et Systèmes

THEME

Réalisation d'une interface homme/machine pour le contrôle et la supervision d'un robot mobile

Proposé et dirigé
par,
DR.R. KARA

Préparé et présenté par,
M. OUZZINE Fethi
M. FATAH Nasreddine

Année Universitaire 2017-2018

Dédicace

Je dédie se modeste travail à mes chers parents qui m'ont beaucoup encouragé et soutenu pendant tout leur vie pourvoir le jour-là et pour ces précieux conseils ainsi pour ces recommandations.

A mes très chers frères et ma chère sœur.

A ma future femme

A toute ma famille, grande et petite.

à tous mes collègues Master 2 /2018, à mon collègue FATAH Nasreddine

A tous mes amis sans exception.

Que se mémoire soit l'expression de mon profond respect et mon grand amour.

Fethi

Dédicaces

Je dédie ce modeste travail a tout homme qui croit que la science est la lumière, et l'ignorance est l'obscurité, à l'esprit de ma mère et à mon père Hocine, à tout ma familles frères et sœurs... oncles, tantes et cousins chacun avec son nom, à ma femme et à mon fils Abdellah, à tous mes frères et camarades chacun avec son nom, à tous mes collègues Master 2 /2018, à mon collègue Fethi Ouzzine.
À tout le peuple algérien, à tout le peuple et la terre palestinienne…
Merci et bon courage.

Nasreddine.

Remerciement

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la , la volonté, le courage afin d'accomplir ce travail modeste.

En second lieu, nous tenons à le grand remerciement notre encadreur ***Dr. KARA Réda*** *qui a proposé le thème de ce mémoire, pour ses conseils et ses dirigés du début à la fin de ce travail.*

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Nous tenons à exprimer notre profonde gratitude à nos familles qui nous ont toujours soutenues et à tout ce qui participe de réaliser ce mémoire. Ainsi que l'ensemble des enseignants qui ont contribué à notre formation.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Sommaire

Listes des figures

Listes des tableaux

Notation et acronymes

Bibliographe

Introduction générale 1

Chapitre1 : Les robots mobiles : historique et description

1. Introduction 2
2 .Définition d'un robot mobile 2
3. Aperçu historique 3
4. Classes de robots `a roues 3
4.1. Robot unicycle 3
4.2. Robot tricycle 4
4.3. Robot voiture 5
4.4. Robot omnidirectionnel 5
5. La robotique mobile tout-terrain 6
6. Capteurs utilisés dans les systèmes de robotique mobile 10
6.1. Les Capteurs extéroceptifs 10
a. Les capteurs télémétriques 11
b. Les GPS 11
c. Les caméras 12
7. Les architectures de contrôle 12
7.1. Contrôleurs Hiérarchiques 13
7.2. Contrôleurs réactifs 15
7.3. Contrôleurs hybrides 18
8. Conclusion 19

Chapitre 2 : étude des systèmes embarques utilisées

1. Introduction 20
2. La carte Arduino 21
2.1. Définition 21
2.2. Les Applications 21
2.3. Les atouts majeurs d'un système à carte Arduino 21
2.4. Arduino Uno 23
2.5. Structure de la carte Arduino Uno 23

2.6. Spécifications techniques 25
2.7. Le Logiciel de gestion de la carte Arduino 25
3. La carte Raspberry Pi 27
3.1. La carte Raspberry Pi 3 model B 28
3.2. Les composants standards de Raspberry 29
3.3. Les GPIO 31
3.4. Mise en marche d'un Raspberry Pi 3 31
3.5. Le réseau 33
3.6. Préparation de la carte SD 33
3.7. Système d'exploitation 34
3.8. Programmation 34
3.9. Synthèse 34
4. La caméra Pixy 35
4.1. Aperçu 35
4.2. Bref historique 36
4.3. Simplicité de mise en œuvre du système de vision 36
4.4. Support du contrôleur 37
4.5. La détection d'objets 37
4.6. Définition simple de l'objet à suivre 38
4.7. L'application PixyMon 38
4.8. Spécifications techniques 39
5. Schéma synoptique du robot mobile utilisé 40
6. Conclusion 41

Chapitre3 : les langage de programmation et logiciels d'interfaçage

1. Introduction 42
2. Définition l'interface homme /machine (IHM) 42
3. Les différents types du logiciel de programmation 42
A. java 42
B. java script 43
C. php 43
D. python 43
E. Node –Red 43
4. Pourquoi Node-RED? 44
5. Définition Node-Red 44
6. Historique 45
7. Les avantages et les Inconvénients de Node-Red 45
7.1. Les principaux avantages de Node-Red 46

7.2. Les Inconvénients de Node-Red....46
8. Les Nœuds composant Node Red....47
9. Quelque exemple de sketch node red....48
9.1. Commande moteur dc connecté à un Arduino uno....48
9.2. Capteur de température....51
10. conclusion....54

Chapitre4 : Mise en œuvre de l'interface de commande du robot

1. Introduction....54
2. Installation et configuration du Node-Red....55
2.1. Première méthode....55
2.1.1. Etape 1 – Installation de Node.js et npm....55
2.1.2. Etape 2 – Installer Node-RED....56
2.1.3. Etape 3 – Faire démarrer Node-RED au démarrage du système....58
2.2. Deuxième méthode....59
3. les noueds (nodes) utilises pour implémenté la page web....61
4. Conception de l'interface web sur PC....61
4.1. Démarrage et configuration de la camera pixy....61
4.2. Chois, sélection et lancement des programmes python pour la commande du robot (suive d'un objet)....62
4.3. Affichage et récupération des paramètres ambient du robot....63
5. Conception de l'interface finale sur PC....64
6. Lancement et test la page web sur pc....65
7. Préparation et implémentation de l'interface sur robot (Raspberry)....66
A. Mise à jour et configuration du node red sur robot (Raspberry)....66
B. Mise en service de l'interface web sur robot (Raspberry)....67
C. Lancement et contrôle du robot à partir de la page web....68
8. Conclusion....71

conclusion generale....73

Listes des figures

Chapitre1 : Les robots mobiles : historique et description

Figure 1.1 : Robot de type unicycle. 4
Figure 1.2 : Robot de type tricycle. 4
Figure 1.3 : Robot de type voiture. 5
Figure 1.4 : Robot de type omnidirectionnel Comparaison des différents types. 5
Figure 1.5 : Robots d'exploration planétaires. 7
Figure 1.6 : Chronologie de la robotique mobile. 7
Figure 1.7 : Quelques exemples de robots mobiles tout-terrain. 8
Figure 1.8: exemple d'application militaire pour les robots mobiles. 8
Figure 1.9 : Exemples de robots mobiles rapides. 9

Chapitre 2 : étude des systèmes embarques utilisées

Figure 2.1 : Structure de la carte Arduino Uno. 24
Figure 2.2 : logiciel Arduino 26
Figure 2.3 : Présentation des boutons. 26
Figure 2.4 : Les composants standards d'un raspberry. 29
Figure 2.5 : Pixy CMUcam5 36
Figure 2.6 : structure de la Cam Pixy CMUcam5. 39
Figure 2.7 : Schéma synoptique du robot mobile utilis …… ..40

Chapitre3 : les langage de programmation et logiciels d'interfaçage

Figure 3.1 :Certains des nœuds disponibles dans Node-Red 47
Figure 3.2 : Notre premier exemple moteur dc 49
Figure 3.3 : Configuration du nœud nommé 'START' 50
Figure 3.4 : Configuration du nœud nommé 'STOP' 50
Figure 3.5 : Configuration du nœud nommé 'DC MOTOR' 50
Figure 3.6 : interface de commande moteur DC 51
Figure 3.7 : Notre deuxième exemple lecture de température 51
Figure 3.8 : configuration du nœud nomme 'Arduino in ' 52
Figure 3.9 : Configuration du nœud nommé 'Température gauge' 52
Figure 3.10 : Interface de lecture de température 53

Chapitre4 : Mise en œuvre de l'interface de commande du robot

Figure 4.1 **:** interface de node-red……..……..57
Figure 4.2 : construction de la page web partie camera pixy 62
Figure 4.3 : construction de la page web partie des programmes python 62
Figure 4.4 : construction de la page web partie des paramètres ambient du robot.....63
Figure 4.5: Schéma programme node-red de l'interface de command robot mobile 64
Figure 4.6: l'interface web page camera pixy 65
Figure 4.7: l'interface web (page programmes) 65
Figure 5.8: l'interface web (page paramètres).....66
Figure 4.9:lancement du node red sur robot 67
Figure 4.10 :importé la code source de la page web dans une node red de la Raspberry 67
Figure 4.11 : Schéma programme node-red de l'interface après import sur Raspberry.....68
Figure 4.12 : Lancement de l'interface web localement 69
Figure 4.13 : Lancement de l'interface web à distance 69
Figure 4.14: Lancement de l'interface web localement 70
Figure 4.15: L'affichage les paramètres ambiante du robot 70
Figure 4.16: Choix de programme.....71

Listes des tableaux

Tableau 1.1 : Les avantages et des inconvénients des différents types de robots à roues…….6
Tableau 1.2 : Caractéristiques de quelques robots mobiles rapides……..9
Tableau 1.3 : les avantages et les inconvénients de contrôleurs Hiérarchiques……..14
Tableau 1.4 : les avantages et les inconvénients de contrôleurs réactifs…….17
Tableau 2.1 : caractéristiques de la carte Arduino uno…….25

Liste des acronymes

CCD : Charge Coupled Device.

GPS: Global Positioning System.

Mini-PC: mini personal computer.

LED: light-emitting diode.

USB: Universal Serial Bus.

IDE : Integrated Development Environment.

PWM : Pulse Width modulation (Modulation de la largeur des pics).

ICSP : In-Circuit Serial Programming ou (Programmation Série Dans le circuit).

SRAM : Static Random Access Memory (mémoire vive statique).

GND : Ground (la masse).

TTL : Transistor-Transistor Logic.

SPI : Interface Série Périphérique.

TW : Two Wire Interface ou (Interface "2 fils").

UART : Universal Asynchronous Receiver Transmitter ou (émetteur-récepteur asynchrone universel).

I2C: inter-integrated circuit.

FPGA: field-programmable gate array.

HDMI: High Definition Multimedia Interface.

WIFI: Wireless Internet for Frequent Interface.

CPU: Central processing unit.

RAM: Random Access Memory (mémoire vive).

LCD: liquid crystal display.

GPIO: General Purpose Input/ Output.

MDI: Medium dependent interface.

LX: Réception data.

TX: émission data.

VIN: Voltage Input.

CSI: Camera Serial Interface.

DSI: Desktop Serial Interface.

FFC: Flexible Flat Cable.

LAN: Local Area Network.

JTAG: Joint Test Action Group.

ADC: Analog to Digital Converter.

DAC: Digital to Analog Converter.

DMA*:* Direct Access Memory.

ROM: Read-Only Memory.

SDA: Serial Data Line.

SCL: Serial Clock Line.

MCU: Multipoint Control Unit.

PCB: Printed Circuit Board.

HD: High Definition.

SD: Secure Digital.

SDHC: Secure Digital High-Capacity.

RGB: Red/Green/Blue.

USB OTG: Universal Serial Bus On-The-Go.

AREF : Analogique Référence.

TWI : Two Wire Interface.

AC : Alternating Current.

DC : Direct Current.

SOC : System On a Chip.

IHM :interface homme/machine

JSP: Java Server Pages

Php : Hyper text Preprocessor

MySQL: Structured Query Language

PERL: Practical Extracting and Reporting Language

Unix: operating system

Mac OS: Macintosh operating systems.

MS-DOS: Microsoft Disk Operating System

Mongo DB: data base program

E/S :Entrée/Sortie

NR: Node-Red

API: Automate Programmable Industriel

NPM : node packet manager

IO:input/output

IoT : Internet of Things

HTTP : Hyper Text Transfer Protocol

MQTT : Message Queuing Telemetry Transport

UDP: User Datagram Protocol

TCP: Transmission Control Protocol

DC: Direct Current

UI: User interface

IBM: International Business Machines

ملخص:

يهدف هذا المشروع الى تصميم واجهة ويب تفاعلية بين الإنسان والآلة باستخدام نظام العقد الحمراء. المراد من صفحة الويب هاته هو التحكم في روبوت متحرك ذو عجلتين. يمكن الدخول على هاته الواجهة عن بعد ومن أي حاسوب او هاتف ذكي. تظم هذه الواجهة الأوتوماتيكية ثلاث صفحات. يظم الروبوت المستعمل في هذا التطبيق أنواع مختلفة من الأنظمة الإلكترونية المدمجة (مثل : راسبيري، ارد وينو).

الكلمات المفاتيح : العقد الحمراء ، ارد وينو، راسبيري ،الروبوت ،واجهة ،برمجة.

RESUME:

Ce projet vise à concevoir une interface web homme-machine en utilisant le système de Node-Red. Le but de cette page web est de commander un robot mobile à deux roues. Vous pouvez accéder cette interface à distance et à partir de n'importe quel ordinateur ou téléphone intelligent. Cette interface automatique est de trois pages. Le robot utilisé dans cette application conçoit différents types des systèmes embarquées (tels que: Raspberry, Arduino).

Mots-clés: Node-Red, Arduino, Raspberry, Robot, interface, programmation.

ABSTRACT:

This project aims to design an interactive human-machine web interface using the Node-Red software. The purpose of this web page is to command a two-wheeled robot. You can access this interface remotely and from any computer or smart phone. This automatic interface has three web pages. The robot used in this application has different types of embedded systems (such as: Raspberry, Arduino)

Keywords: Node-Red, Arduino, Raspberry, Robot, interface, programming.

Introduction générale

Introduction générale

L'implémentation et la mise en service d'une interface de communication entre l'homme et la machine revête ces dernières années une importance dans le monde de l'industrie et des robots « autonomes ». Le développent rapide technologie dans les domaines de l'industrie, des machines outils et des robots a poussé les développeur de mettre en œuvre des logiciels pour facilité l'implémentation et la conception des interfaces logiciels et/ou matériel pour automatiser les taches de commande, de contrôle et/ou supervision de ces processus. Ces interfaces ont pour objectif de rendre ces machines le plus accessible a un simple opérateur ou utilisateur.

Pour réaliser les objectifs de notre projet, nous disposons d'une plate-forme robot-vision dotée d'une architecture matérielle bâtie autour de cartes Raspberry, Arduino et d'un système de vision intégré que l'on nome Cam Pixy. De ce fait, l'objectif que nous nous sommes imposé est la conception d'une interface Homme /Machine qui permette la gestion, le contrôle et la supervision de ce robot mobile. Par ailleurs, la conception de cette IHM devra permettre l'accès aux fonctionnalités du robot depuis n'importe quelle machine (PC, Tablette, Smartphone, …) connectée au réseau wifi. Ce prélude permettra de contrôler le robot mobile depuis n'importe quel endroit doté d'une connexion internet.

Ce document est arrangé en quatre chapitres qui maintiennent les aspects suivants :

Le **premier chapitre** présente une introduction à la robotique mobile, des descriptions des différents éléments constituants les robots mobiles ainsi que les différentes structures mécaniques qui existent.

Le **deuxième chapitre** est consacré aux systèmes embarqués utilisés dans notre projet, tel que le système de vision Pixy, la carte Raspberry et la carte Arduino.

Le **troisième chapitre** décrive les différents langages et logiciels d'interfaçage. Ainsi que le choix software Node-red et leur avantages et inconvénients.

Le **quatrième chapitre** contemplera les détaille de l'implémentation de notre interface Homme / Machine. Nous présenterons les différentes étapes nécessaires à l'installation de tous les softwares et les patchs nécessaires sur un PC. Nous décrirons par la suite les phases importantes qui rentrent dans la conception de cette IHM et puis son transfère la plate-forme robot-vision.

Enfin, **la conclusion** résume tous les résultats que nous avons obtenus. Ce mémoire sera achevé par des recommandations et perspectives réalisables dans un futur proche

Chapitre1 : Les robots mobiles : historique et description

1. Introduction

La robotique a atteint son plus grand succès jusqu'à ce jour dans le monde de la fabrication industrielle.

Dans l'industrie de l'électronique, les manipulateurs place des composants montés en surface avec une précision sur humaine, ce qui rend le téléphone portable et ordinateur portable possible.

Cependant, pour tous leurs succès, ces robots commerciaux souffrent d'un inconvénient majeur : le manque de mobilité. Un manipulateur fixe dispose d'une gamme limitée de mouvement qui dépend de l'endroit où elle est boulonnée. En revanche, un robot mobile serait en mesure de voyager à travers l'usine de fabrication, en appliquant avec souplesse ses talents partout où il est le plus efficace.

De manière générale, on regroupe sous l'appellation robots mobiles l'ensemble des robots à base mobile, par opposition notamment aux robots manipulateurs. L'usage veut néanmoins que l'on désigne le plus souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens.

Le but de ce chapitre est de situé le robot mobile objet de notre étude. A partir des descriptions des différents éléments constituants les robots mobiles.

2. Définition d'un robot mobile

- Un robot mobile est un véhicule dote de moyens de locomotion qui lui permettent de se déplacer. Suivant son degré d'autonomie il peut être dote de moyens de perception et de raisonnement.

- Un robot est un automate doté de capteurs et d'effecteurs lui donnant une capacité d'adaptation et de d´emplacement proche de l'autonomie.

3. Aperçu historique

Ce n'est qu'au tout début du XX^{e} siècle que les robots firent leur apparition, suite aux travaux d'ingénieurs qui voulaient tester des hypothèses émises par des biologistes et des psychologues. Le chien électrique conçu par Hammond et Miessner en 1915 était attiré par une lumière, selon le phototropisme animal mis en évidence par Loeb en 1918. Les machines de Russell (1913) et de Stephens (1929), les tortues cybernétiques de Grey Walter (Figure 1.1) (1950), le renard électronique de Ducrocq (1953) ou l'homéostat d'Ashby (1952) étaient, elles, dotées de capacités d'apprentissage directement issues des travaux des psychologues Thorndike (1911), Hull (1943) et du physiologiste Pavlov (1903) sur l'Homme et l'animal.

Ces réalisations sont des robots, car elles ne se comportent plus comme de simples automates dont les organes moteurs - leurs mécanismes - obéissent à un programme préétabli. À la différence des automates, ils ont des organes sensoriels - les capteurs - qui recueillent des informations de l'environnement qui vont, elles, influencer l'activité de leurs organes moteurs - les actionneurs. [1] [3] [7]

4. Classes de robots à roues

Il existe plusieurs classes de robots à roues déterminées, principalement, par la position et le nombre de roues utilisées.

Nous citerons ici les quatre classes principales de robots à roues.

4.1. Robot unicycle

Un robot de type unicycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices.

C'est un robot non-holonome, en effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion.

Sa commande peut être très simple, il est en effet assez facile de le déplacer d'un point

A un autre par une suite de rotations simples et de lignes droites.

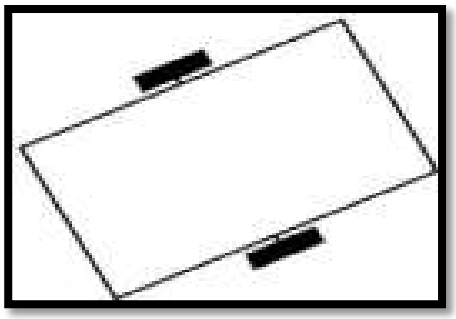


Figure 1.1 : Robot de type unicycle.

4.2. Robot tricycle

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable.

C'est un robot non-holonome. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable. [2] [4]

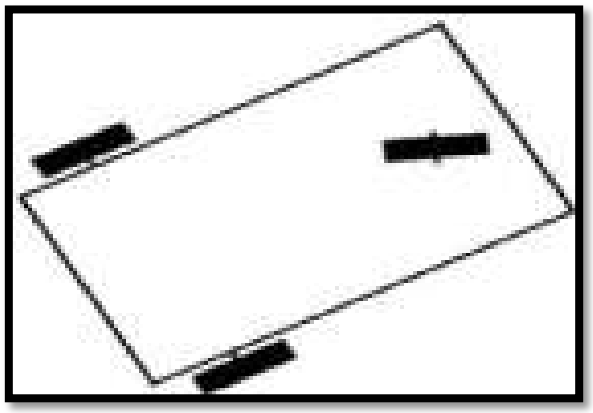


Figure 1.2 : Robot de type tricycle.

4.3. Robot voiture

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe.

Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire.

Toutes les autres propriétés du robot voiture sont identiques au robot tricycle, le deuxième pouvant être ramené au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé. [3]

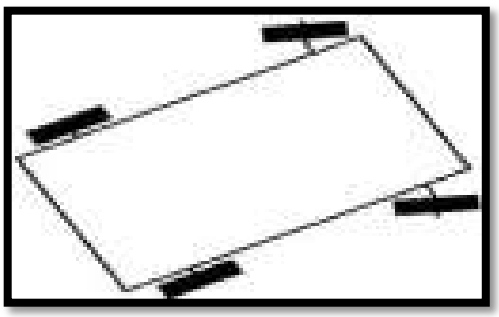


Figure 1.3 : Robot de type voiture.

4.4. Robot omnidirectionnel

Un robot omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral. L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer dans toutes les directions. Mais ceci se fait au dépend d'une complexité mécanique bien plus grande. [3] [4]

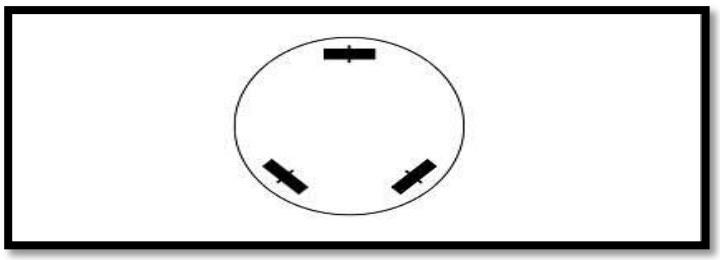


Figure 1.4 : Robot de type omnidirectionnel **Comparaison des différents types.**

Tableau 1.1 : Les avantages et des inconvénients des différents types de robots à roues.

Type de robot	Avantage	Inconvénient
Unicycle	• Stable • Rotation sur soi-même • Complexité mécanique faible	• Non-holonome
Tricycle	• Complexité mécanique modérée	• Non-holonome • Peu stable • Pas de rotation sur soi-même
Voiture	• Stable • Complexité mécanique modérée	• Non-holonome • Pas de rotation sur soi-même
Omnidirectionnel	• Holonome • Stable • Rotation sur soi-même	• Complexité mécanique importante

5. La robotique mobile tout-terrain

Un système de locomotion terrestre a pour finalité le déplacement d'un corps grâce à la génération de forces s'opposant aux phénomènes de résistance au mouvement. Dans le cas des véhicules à roues, le mécanisme de propulsion généré des couples qui mettent ces roues en rotation. La friction ou l'adhérence des roues sur le sol permet de convertir ces couples actionneurs en force motrice du véhicule. Le mouvement résultant dépend de l'environnement (résistance aérodynamique, interaction roue-sol, topologie du terrain et gravité) et du véhicule lui-même (géométrie, dynamique, frottements internes).

En robotique mobile, on cherche à rendre ce véhicule le plus autonome possible dans l'accomplissement de sa tâche. Cette tâche est soit le transport d'une charge utile (instruments de mesure, bras articulé, minerai, etc.), soit l'exploration, soit le travail sur site dans un environnement inconnu, impraticable ou dangereux pour l'être humain. [1] [2]

On peut considérer qu'il s'agit du premier robot mobile dont la liaison perception action n'est plus une simple boucle réflexe mais résulte d'un traitement relativement complexe. Son objectif principal était la navigation dans un environnement très structuré tel que l'intérieur d'un bâtiment.

L'exploration et le transport en environnement non structuré, c'est-`a-dire non créé et non modifié par l'homme, apparaissent comme un enjeu majeur. Les robots mobiles à roues représentent une des solutions possibles. C'est le cas de l'exploration planétaire extraterrestre. Le premier véhicule planétaire automatique télé opéré est déployé en 1970

En 1997, la mission Pathfinder de la NASA envoie Séjourner, le premier robot mobile

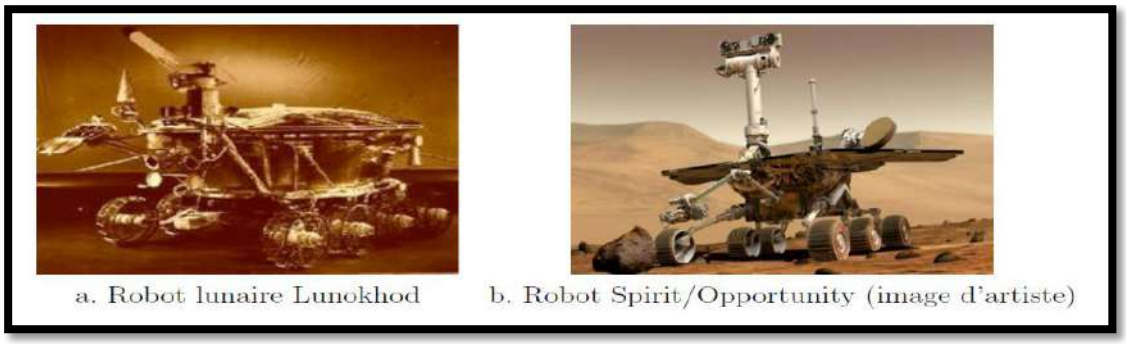


Figure 1.5 : Robots d'exploration planétaires.

Cette chronologie succincte montre les développements des dernières décennies dans le domaine de la robotique mobile. Au cours des premières décennies, l'invention de l'ordinateur (1948) et le développement de l'automatique en tant que science permet la conception de véhicules autonomes d'intérieur.

La conquête du milieu non artificiel par les robots mobiles commence dans les années soixante. Mais l'utilisation de véhicules avec un degré d'autonomie dépassant la télé opération est plus récente grâce, en particulier, à la miniaturisation de la logique de commande (microprocesseur) et des capteurs.

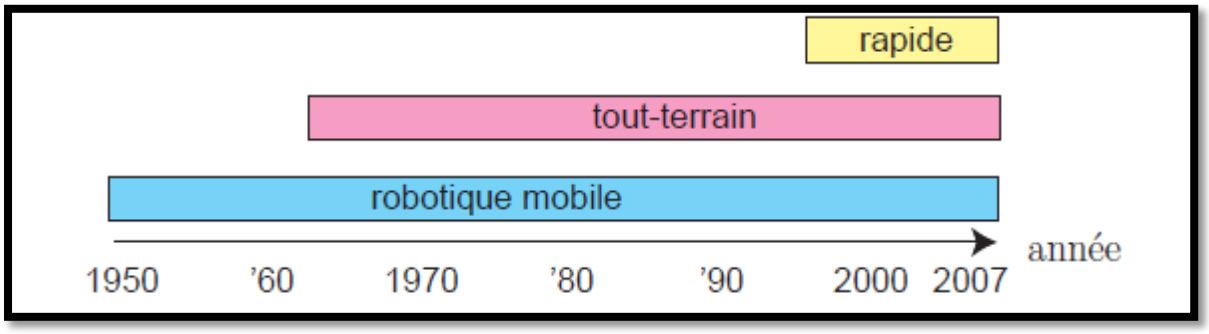


Figure 1.6 : Chronologie de la robotique mobile.

Il existe aujourd'hui de nombreux systèmes robotiques en environnement naturel exploités pour diverses applications.

D’une manière générale, les recherches portent sur l’amélioration des performances des robots en matière de franchissement. L’attention est focalisée sur la navigation et la locomotion sur des terrains non structurés, présentant des difficultés topologiques (pentes, dépressions, obstacles, etc.).

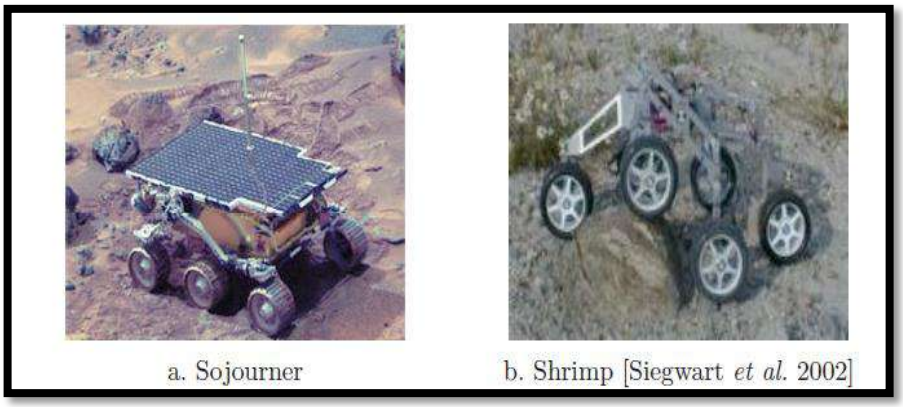


Figure 1.7 : Quelques exemples de robots mobiles tout-terrain.

Le problème des véhicules autonomes à grande vitesse a été peu abordé jusqu’ici. Ce concept de grande vitesse pour un robot mobile dépend du terrain traversé et du véhicule lui-même. On considérera qu’un véhicule autonome se déplace à grande vitesse sur un terrain donné si les effets dynamiques ont une influence importante sur sa trajectoire, au point d’engendrer des comportements spécifiques (retournements, glissement, dérapage, chutes balistiques, etc.)

Il existe plusieurs véhicules télé opérés rapides, principalement utilisés pour des applications militaires. (Figure 1.8)



Figure 1.8: exemple d’application militaire pour les robots mobiles.

Citons le Dragon Runner, développé par l'institut de robotique de l'Université de Carnegie Mellon. La plateforme est réversible et les roues sont directionnelles. Le RobucarTT est un véhicule à quatre roues motrices indépendantes. Toutes les roues sont directionnelles.



Figure 1.9 : Exemples de robots mobiles rapides.

Le tableau 1.2 donne les caractéristiques de quelques systèmes robotiques rapides (d´passant 5 m/s). Il s'agit généralement de systèmes télé opérés avec une faible autonomie décisionnelle.

Tableau 1.2 : Caractéristiques de quelques robots mobiles rapides.

Robot	**Longueur (Cm)**	**Largeur (Cm)**	**Poids (Kg)**	**Vitesse Maximale (m/s)**
Drogon Runner	39,4	28,6	7,4	9
RobucarTT	184	130	350	5
UGCV RST	213	137	450	11
ARTmiS [Shimoda et al. 2005]	89	61	35,4	7

La commande de robots mobiles rapides implique des contraintes spécifiques que nous détaillons dans ce mémoire. Nous verrons en particulier qu'on ne peut plus supposer le roulement sans glissement et donc la non-holonomie du système.

6. Les capteurs utilisés dans les systèmes de robotique mobile

Pour réaliser un système de robotique mobile intelligent, il est nécessaire d'utiliser des capteurs qui fourniront la perception requise de l'environnement pour une prise de décision intelligente. A ce jour, il n'existe pas de capteur qui se démarque des autres par une efficacité vraiment supérieure. Ils ont tous des avantages et des inconvénients avec lesquels il faut pouvoir composer. Par contre, leur importance peut être passablement influencée par l'environnement dans lequel le système est amené à évoluer. Le défi est donc de réaliser des

systèmes qui utilisent le mieux possible les capteurs ou une combinaison de capteurs en tenant compte de leurs conditions d'utilisation, et ceci requiert une bonne connaissance de leurs caractéristiques. Nous pouvons définir deux catégories de capteurs couramment utilisées en robotique mobile : ceux qui délivrent une information caractérisant l'environnement et ceux qui interviennent dans le fonctionnement du système robotique. Les premiers sont appelés capteurs extéroceptifs qui fournissent des informations sur le monde extérieur au système mobile, et les seconds sont les capteurs proprioceptifs qui fournissent des informations propres au comportement interne du système mobile, c'est-à-dire sur son état à un instant donné. Dans ce qui va suivre, nous allons nous contenter de décrire quelques capteurs extéroceptifs. [2],[6].

6.1. Les Capteurs extéroceptifs

Les capteurs extéroceptifs sont employés en robotique mobile pour collecter des informations sur l'environnement d'évolution du système mobile. Ils sont le complément indispensable aux capteurs proprioceptifs présentés précédemment. Des méthodes de fusion de données sont alors utilisées pour conditionner et traiter les informations sensorielles de natures différentes. Ils sont notamment utilisés dans les domaines d'application tels que l'évitement d'obstacle, la localisation, la navigation et la modélisation d'environnements. Les principaux capteurs utilisés en robotique mobile sont : les capteurs télémétriques (les ultrasons, les lasers et les infrarouges), le GPS et les caméras.

a. Les capteurs télémétriques

On appelle télémétrie toute technique de mesure de distance par des procédés acoustiques, optiques ou radioélectriques. L'appareil permettant de mesurer les distances est appelé télémètre. Il existe différentes technologies pour réaliser un télémètre. Nous présentons dans ce qui suit les télémètres les plus utilisés dans les systèmes de robotique mobile, en donnant une idée de leur gamme de mesure et d'application.

b. Les GPS

Un des systèmes les plus connus de ces dernières années est le Système de Positionnement Global (GPS) du département de la défense des Etats-Unis qui exploite une constellation de vingt-quatre satellites en orbite à haute altitude.

Chaque satellite détecte les récepteurs terrestres fixes ou mobiles qui se trouvent dans sa zone de visibilité et transmet à chacun, à période fixe, un signal contenant des informations telles que son identification, l'heure d'émission du signal et les paramètres définissant son orbite donc sa position. Théoriquement, un récepteur doit être visible d'au moins trois

satellites, pour pouvoir calculer sa position dans un repère centré sur la terre, en calculant l'intersection de trois sphères (trois cercles pour un système mobile au sol) ; de fait, pour tenir compte des décalages temporels entre les mesures fournies par les satellites, quatre mesures de distance entre le satellite et le récepteur sont requises pour disposer d'une précision satisfaisante [3].

Ce système est cependant loin de résoudre tous les problèmes de localisation des systèmes mobiles. Il fonctionne en effet difficilement dans des environnements urbains, et n'est pas utilisable à l'intérieur des bâtiments car les ondes employées sont fortement atténuées en traversant les murs. Sa précision est de plus souvent trop faible pour qu'un système mobile terrestre puisse utiliser ces informations seules. En pratique, il est souvent couplé à un système inertiel qui permet de pallier aux pertes du signal GPS et il ne remplace de toute façon pas les capteurs du système mobile qui lui permettent de percevoir son environnement immédiat [7].

Mis à part son utilisation pour la localisation, ce système est aussi utilisé pour l'aide à la navigation d'avions et de bateaux lors de l'approche d'aéroports ou de ports, ainsi que dans la surveillance de camionnettes de livraison et de véhicules d'urgence, en vue de choisir au mieux leurs itinéraires.

c. Les caméras

Depuis les années 70, les caméras CCD (Charge Coupled Device) ont été déterminantes dans l'évolution de la vision : la rapidité d'acquisition, la robustesse et la miniaturisation sont autant d'avantages qui ont facilité leur intégration.

Elles sont très performantes en termes de portée, précision et quantité d'informations exploitables.

Elles sont de plus les seules capables de restituer une image sensorielle de l'environnement la plus proche de celle perçue par l'être humain. C'est sans doute le capteur fournissant la plus grande variété d'informations à partir d'une seule acquisition. En effet, en appliquant divers algorithmes de traitement d'images, il est possible d'extraire des éléments caractéristiques de l'environnement comme des segments de droite, des plans, mais aussi des balises et des couleurs.

En revanche, l'inconvénient majeur de tels capteurs se situe d'abord au niveau de la gestion du flux important de données exploitables (traiter une image demeure une opération

délicate et surtout coûteuse en temps de calcul), ensuite à leur sensibilité aux conditions d'éclairage.

Pour augmenter le champ de vision de ces caméras, on peut les associer soit avec d'autres caméras ou soit à des systèmes de réflexion de type miroir : il s'agit des techniques de vision omnidirectionnelle. Cette association permet d'obtenir une vue sur 360 degrés de l'environnement [5].

7. Les architectures de contrôle

Les architectures de contrôles peuvent être classées en trois grandes catégories que nous détaillerons par la suite :

- Les contrôleurs hiérarchiques
- Les contrôleurs réactifs
- Les contrôleurs hybrides.

Un exemple typique pour un robot mobile est l'arbitrage qui doit être fait entre l'exécution la plus précise possible d'un plan préétabli pour atteindre un but et la prise en compte d'éléments imprévus, tels que les obstacles mobiles. Ces arbitrages, que ce soit au niveau du choix de stratégie, ou au niveau de l'utilisation des capteurs, des effecteurs ou des ressources de calcul, sont réglés par un ensemble logiciel appelé architecture de contrôle du robot. Cette architecture permet donc d'organiser les relations entre les trois grandes fonctions que sont la perception, la décision et l'action. [3]

Selon cette définition une architecture doit donc être conçue pour un robot précis, mais en utilisant des modules génériques. De manière plus générale il existe également des règles de conception relativement générales qui permettent de réaliser ces implémentations. En fonction de ces règles, les architectures de contrôle peuvent être classées en trois grandes catégories que nous détaillerons par la suite : les contrôleurs hiérarchiques, les contrôleurs réactifs et les contrôleurs hybrides . [3]

Comme le précise cette définition, toutes ces architectures ne diffèrent pas forcement par les méthodes élémentaires employées mais plutôt par leur agencement et leurs relations.

7.1. Contrôleurs Hiérarchiques

Historiquement, les premiers robots mobiles dérivés des recherches en intelligence artificielle utilisaient des contrôleurs hiérarchiques dont le fonctionnement repose essentiellement sur la capacité de décision travaillant sur un modèle du monde supposé quasiment-parfait.

Ces architectures fonctionnent selon un cycle rigide de modélisation de l'environnement, planification des actions au sein de cette représentation, puis exécution du plan. La capacité de décision était issue des premiers travaux en intelligence artificielle et reposait essentiellement sur des traitements de données symboliques. Ces architectures ont rapidement montré leurs limites et leur incapacité à fonctionner dans un environnement qui ne soit pas statique et simplifié à l'extrême. [5]

L'essentiel des problèmes de ces architectures provient de l'utilisation d'un modèle interne central qui est le seul pris en compte pour guider le robot. Elles se trouvent confrontées à tous les problèmes des premiers développements de l'intelligence artificielle symbolique.

Ces architectures supposent premièrement qu'un modèle informatique du monde puisse représenter toutes les informations pertinentes pour le déplacement du robot. Or un tel modèle ne peut être suffisant dans un environnement dynamique car au moment de la réalisation de l'action l'environnement peut avoir suffisamment changé pour que la décision ne soit plus valide. Ce problème était particulièrement crucial au début de la robotique mobile ou les capacités de calcul limitées entraînaient des temps de planification de l'ordre de plusieurs dizaines de minutes. Mais l'augmentation des capacités de calcul ne suffit pas à résoudre ce problème qui ne permet pas de gérer un environnement de travail réaliste. [3]

Le tableau suivant montre les avantages et les inconvénients de contrôles hiérarchiques

Tableau I.3 : les avantages et les inconvénients de contrôleurs Hiérarchiques

Les avantages	➢ L'avantage essentiel de ces architectures réside dans la possibilité d'intégrer des raisonnements de haut niveau (niveau mission, planification) qui s'appuient sur des modèles assez complets (cartes par exemple) de l'environnement dans lequel évolue le robot. Cet aspect est, comme nous le verrons plus loin, absent des architectures purement réactives.
les inconvénients	➢ Leur principal inconvénient provient de leur faible vitesse de réaction. Elles sont en effet totalement incapables de prendre en compte des obstacles dynamiques ou non détectés lors de la modélisation. Ceci est dû à la lenteur des phases de modélisation et de planification qui ne peuvent généralement pas être exécutées en temps réel, même lorsqu'elles sont implantées sur de gros calculateurs extérieurs au robot. ➢ La construction du modèle sur lequel se base la planification n'est pas une tache aisée. Il est en pratique souvent difficile de relier les données sensorielles aux objets du monde réel. Ceci est dû en grande partie à l'imprécision de l'information délivrée par les capteurs utilisés en robotique. R. A. Brooks résume ce problème en affirmant qu'il existe une grande différence entre capter et percevoir. ➢ Ces systèmes sont en général peu robustes de par leur modèle centralisé. Une défaillance d'un module peut provoquer le blocage de toute l'architecture. L'intelligence artificielle traditionnelle s'est appuyée sur la volonté de reproduire des systèmes vivants, à travers des concepts plus ou moins valides issus d'observations macroscopiques empiriques, ne correspondant pas aux mécanismes réels régissant les êtres vivants.

7.2. Contrôleurs réactifs

Cette approche, radicalement différente de la précédente se distingue par l'abandon des phases de modélisation et de planification. Elle est née à partir de considérations issues de l'éthologie et plus particulièrement de l'entomologie, dont en particulier les insectes, ne possèdent, en raison de leur taille que très peu de capacités de modélisation et de raisonnement. Ils n'en sont pas moins capables de mener à bien des tâches complexes. Leur comportement semble même du point de vue d'un observateur extérieur résulter d'une intelligence avancée, similaire par certains côtés à la nôtre. Cette constatation induit l'idée suivante : pourquoi ne pas abandonner le concept centré sur la modélisation de l'environnement et le raisonnement, au profit d'une architecture. [3], [5]

Les robots utilisant cette architecture sont donc en général efficaces pour la tâche précise pour laquelle ils ont été conçus, dans l'environnement pour lequel ils ont été prévus, mais sont souvent difficiles à adapter à une tâche différente.

Les réussites de ces architectures sont liées au couplage direct entre la perception et l'action qui permet une prise en compte très rapide des phénomènes dynamiques de l'environnement.

En donc une bonne robustesse dans des environnements complexes.

Comme nous l'avons mentionné, ces architectures sont en général basées sur plusieurs comportements : évitement d'obstacles, déplacement aléatoire, déplacement vers un but, fuite d'un point... Pour guider le robot, il faut donc choisir à chaque instant lequel de ces comportements activer. Ce problème est connu dans la littérature scientifique sous le nom de sélection de l'action.

La solution proposée par Brooks, l'architecture de subsomption [7] est devenue un classique et utilise une hiérarchie des comportements qui se déclenchent donc selon un ordre de priorité en fonction des perceptions du robot.

Une bonne architecture de sélection de l'action doit être capable de choisir un système comportemental possédant un certain nombre de caractéristiques, dont voici les principales :

- Motivé : le système comportemental sélectionné correspond prioritairement à celui de plus forte motivation en fonction des objectifs et des ressources de

l'agent. Par exemple, la faim peut être provoquée par un niveau d'énergie faible ; la faim peut déclencher la recherche de nourriture, à condition qu'une activité prioritaire, telle que la fuite devant un prédateur, ne soit pas en cours.

- Persistant : s'il est pertinent, le système comportemental en cours doit continuer afin d'éviter l'oscillation entre plusieurs systèmes comportementaux éligibles. Par exemple, si l'agent est à la recherche de nourriture, il doit continuer même s'il a subitement autant soif que faim.
- Opportuniste : si l'occasion se présente et si le système comportemental courant le permet, un système comportemental de moins forte motivation peut être sélectionné le temps de satisfaire un objectif secondaire. Par exemple, si l'agent passe à côté d'une source de boisson alors qu'il cherche de la nourriture sans être trop affamé, il pourra faire un détour pour boire puis reprendre sa recherche.
- Direct : le système comportemental doit préférer les actions consommatoires aux actions appétitives, c'est-à-dire favoriser les actions qui mènent directement à la satisfaction d'un objectif (action finale) par rapport aux actions qui préparent une action consommatoire. Par exemple, l'agent a intérêt à manger ce qui est disponible là où il se trouve plutôt que d'aller vers une autre source de nourriture.
- Conciliant : les actions choisies doivent correspondre de préférence à un compromis entre plusieurs solutions, de façon à satisfaire un maximum d'objectifs concurrents plutôt que de n'en satisfaire qu'un seul de façon optimale.

Le tableau suivant montre les avantages et les inconvénients de contrôles réactifs

Tableau I.4 : les avantages et les inconvénients de contrôleurs réactifs

Les avantages	➢ Rapidité de réponse qui lui donne la capacité d'évoluer dans des environnements fortement dynamiques. La simplicité des comportements permet d'accroître encore cette vitesse par une réalisation sur circuit imprimé. ➢ Robustesse qui découle de la parallélisassions et de l'indépendance relative des comportements qui permet au robot de continuer à fonctionner en cas de panne de l'un d'entre eux, ce qui est très important lorsque l'on a à faire à un robot autonome. ➢ L'aspect incrémental en niveau : il suffit de mettre en place un niveau de compétence puis d'ajouter les niveaux supérieurs sans avoir, en théorie, à remettre en question les niveaux inférieurs. Cet aspect est primordial pour la mise au point et la recherche de pannes. Cette approche permet également d'accroître la complexité du comportement global du robot sans pour autant accroître celle du contrôle global puisque celui-ci est complètement distribué. ➢ Sa simplicité implique un faible coût de production et des possibilités de miniaturisation importantes. Ceci ouvre de nouveaux domaines d'application. Brooks s'intéresse notamment à l'utilisation d'un grand nombre de petits robots dans le domaine spatial.
Les inconvénient	➢ L'approche comportementale souffre d'une absence de capacité de modélisation et de raisonnement.

7.3. Contrôleurs hybrides

Les approches hiérarchiques et réactives sont diamétralement opposées.

Cependant, chacune présente des caractéristiques intéressantes. Pour cela, des chercheurs ont essayé de les combiner en mettant au point des architectures hybrides permettant notamment d'allier des capacités de raisonnement et de décision de haut niveau, s'appuyant sur des représentations abstraites des connaissances, avec des comportements réactifs garantissant robustesse et flexibilité.

La plupart des contrôleurs actuellement utilisés choisissent une solution intermédiaire entre ces deux approches sous la forme d'une architecture hybride. Cette notion est à distinguer de celle de système hybride en intelligence artificielle (ou système hybride neuro symbolique) qui désigne un système alliant des composantes utilisant des techniques issues de l'intelligence artificielle symbolique classique et des réseaux de neurones artificiels.

L'architecture hybride se compose de deux niveaux. Le premier est chargé des tâches de navigation de haut niveau, telles que la localisation, la cartographie et la planification. Pour cela, il s'appuie sur un second niveau réactif qui est chargé d'exécuter les commandes avec le plus de précision possible et de gérer les éléments non modélisés de l'environnement tels que les obstacles inconnus ou dynamiques.

L'action conjointe de ces deux niveaux permet de réagir rapidement face aux variations imprévues de l'environnement, tout en permettant la réalisation d'actions planifiées à plus long terme.

Le bas niveau de ces architectures peut être réalisé sous forme de comportements, tels que ceux utilisés dans les architectures réactives. Ces comportements sont des boucles sensorimotrices qui relient les actions aux perceptions avec une phase de décision très courte, qui assure la réactivité. Dans le même temps, les informations sensorielles sont utilisées par le haut niveau dans une boucle sensorimotrice à une échelle de temps beaucoup plus longue. C’est la mise en parallèles de ces deux échelles de temps qui fait la force de ces architectures.

Les exemples d’architectures hybrides foisonnent car de très nombreux laboratoires et organismes travaillant sur la robotique ont développé leur architecture de ce type.

8. Conclusion

La présentation faite dans cette première partie nous a permis de décrire les éléments constituants d'un robot mobile. A partir de ce descriptif, nous avons constaté que pour réaliser la commande de ce type de robot, nous devions répondre à trois impératifs à savoir la perception, la décision et l'action.

La combinaison d'informations issues de capteur proprioceptifs et de capteurs extéroceptifs permet d'envisager la réalisation de tâches robotiques complexes. Dans le cas de la robotique mobile, le traitement de toutes ces informations nécessite une puissance de calcul importante. Du fait de sa mobilité, l'interaction avec ce type de robot nécessite souvent l'élaboration d'une interface homme / machine. Celle-ci permettra à n'importe quel utilisateur de le commander et de récupérer les données des différents capteurs qui équipent ces robots mobiles.

Mais avant d'exposer notre travail de conception de l'interface de commande, nous vous présenterons dans le prochain chapitre les systèmes embarqués qui équipent la plate-forme robot-vision.

Chapitre 2 : étude des systèmes embarques utilisées

1. Introduction

Dans ce chapitre nous allons introduire et présenter les deux cartes électroniques embarquées ainsi que du système de vision que nous utiliserons dans l'élaboration de notre système d'acquisition de données, leur traitement et le calcul des lois de commandes nécessaires à l'asservissement de notre robot mobile. L'architecture matérielle que nous avons développée s'articule essentiellement autour de la carte Arduino de la carte Raspberry Pi et du système de vision Pixy. La caméra embarquée Pixy est considérée comme l'œil humain dans le sens ou grâce à son électronique elle nous fournira directement les caractéristiques visuelles nécessaires à la poursuite d'un objet cible. Le Raspberry Pi sera quant à lui assimilé au cerveau de notre robot tandis que le Arduino aura pour rôle la gestion de l'ensemble Capteurs / Actionneurs du robot.

Par conséquent, nous commencerons ce chapitre par la description des cartes Arduino puis nous présenterons l'architecture et le fonctionnement du mini-PC Raspberry Pi. Enfin, nous détaillerons les spécifications et le fonctionnement du système de vision Pixy.

2. La carte Arduino

2.1. Définition

Arduino est une plate-forme électronique open-source basée sur un matériel et un logiciel faciles à utiliser. Arduino est né au Ivrea Interaction Design Institute comme un outil facile pour le prototypage rapide, destiné aux étudiants sans formation en électronique et en programmation. [10].

Au fil des années, Arduino a été le cerveau de milliers de projets, des objets de tous les jours aux instruments scientifiques complexes. Une communauté mondiale de décideurs - étudiants, amateurs, artistes, programmeurs et professionnels - s'est réunie autour de cette plate-forme open source, leurs contributions ont contribué à une connaissance incroyable de connaissances accessibles qui peuvent être d'une grande aide pour les novices et les experts.

2.2. Les Applications

Les cartes Arduino peuvent lire des entrées - lumière sur un capteur, un doigt sur un bouton ou un message Twitter - et le transformer en sortie - activer un moteur, allumer une LED, publier quelque chose en ligne.

L'étendue de l'utilisation de l'Arduino est gigantesque. Pour vous donner quelques exemples, vous pouvez :

- Contrôler des systèmes intelligents (smart home)

- Fabriques des robots et des machines (bras robots, cnc, imprimante 3D)
- Communiquer avec des plateformes différents (web, serveur)
- Systèmes de détection (gas, mouvement, visage)

2.3. Les atouts majeurs d'un système à carte Arduino

Grâce à son expérience utilisateur simple et accessible, Arduino a été utilisé dans des milliers de projets et d'applications différents. Le logiciel Arduino est facile à utiliser pour les débutants, mais assez flexible pour les utilisateurs avancés. Il fonctionne sur Mac, Windows et Linux. Les enseignants et les étudiants l'utilisent pour construire des instruments scientifiques à faible coût, pour prouver les principes de chimie et de physique, ou pour commencer avec la programmation et la robotique. [9]

Les concepteurs et les architectes créent des prototypes interactifs, des musiciens et des artistes l'utilisent pour les installations et expérimentent de nouveaux instruments de musique. Les fabricants, bien sûr, l'utilisent pour construire plusieurs des projets exposés chez Maker Faire, par exemple. Arduino est un outil clé pour apprendre de nouvelles choses. Toute personne - les enfants, les amateurs, les artistes, les programmeurs - peut commencer à bricoler juste après les instructions étape par étape d'un kit, ou en partageant des idées en ligne avec d'autres membres de la communauté Arduino. [9]

- **Les cartes Arduino peu coûteuses :** sont relativement peu coûteuses par rapport aux autres plates-formes de microcontrôleurs. La version la moins chère du module Arduino peut être assemblée à la main, et même les modules Arduino préassemblés coûtent moins de 50 $.
- **Cross-Platform** : Le logiciel Arduino (IDE) s'exécute sur les systèmes d'exploitation Windows, Macintosh OSX et Linux. La plupart des systèmes de microcontrôleurs sont limités à Windows.
- **Environnement de programmation simple et clair** : Le logiciel Arduino (IDE) est facile à utiliser pour les débutants, mais suffisamment flexible pour que les utilisateurs avancés puissent en profiter. Pour les enseignants, il est commodément basé sur l'environnement de programmation de traitement, afin que les élèves qui apprennent à programmer dans cet environnement connaîtront la façon dont l'IDE Arduino fonctionne.
- **Logiciel open source et extensible** : Le logiciel Arduino est publié sous la forme d'outils open source, disponibles pour l'extension par des programmeurs expérimentés. La langue peut être étendue à travers les bibliothèques C ++, et les personnes

souhaitant comprendre les détails techniques peuvent faire passer le saut d'Arduino au langage de programmation AVR C sur lequel elles se basent. De même, vous pouvez ajouter le code AVR-C directement dans vos programmes Arduino si vous le souhaitez.

- **Matériel open source et extensible** : Les plans des cartes Arduino sont publiés sous une licence Créative Commons, de sorte que les concepteurs de circuits expérimentés peuvent créer leur propre version du module, l'étendre et l'améliorer. Même les utilisateurs relativement inexpérimentés peuvent créer la version boardboard du module pour comprendre comment cela fonctionne et économiser de l'argent. [9], [10]

2.4. Arduino Uno

Arduino Uno est un panneau de microcontrôleur basé sur l'ATmega328P . Il dispose de 14 broches d'entrée / sortie numériques (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, un cristal à quartz 16 MHz, une connexion USB, une prise de courant, un en-tête ICSP et un bouton de réinitialisation. Il contient tout le nécessaire pour supporter le microcontrôleur ; Il suffit de le connecter à un ordinateur avec un câble USB ou de l'alimenter avec un adaptateur AC-DC ou une batterie pour commencer. Vous pouvez bricoler avec votre UNO sans trop déranger pour faire quelque chose de mal, le pire scénario que vous pouvez remplacer Puce pour quelques dollars et recommencer. [9], [10]

2.5. Structure de la carte Arduino Uno

Dans ce qui va suivre, nous détaillerons la structure de ce système et leurs intérêts.

- **Le microcontrôleur**

Le microcontrôleur correspond le cerveau de notre carte, c'est lui qui va traiter provenant des capteurs dans l'entrée et qui va donner la réponse à la sortie à travers le programme que vous avez créé et qui va le stocker dans sa mémoire.

- **L'alimentation**

Il faut tout d'abord connaître les spécifications de la carte Arduino que l'on possède. Elles utilisent tout d'abord toutes du courant continu.

Si l'on regarde les différentes broches possibles, on va généralement trouver une broche étiquetée 5V, une broche étiquetée 3,3V et une broche étiquetée VIN ou RAW.

- VIN ou RAW correspondent à l'entrée du régulateur de tension présent sur la carte. Pour alimenter l'Arduino via cette broche, il faut une alimentation dont la tension est comprise entre 7 et 12V ;
- 5V n'a pas le même rôle selon que le microcontrôleur qui équipe l'Arduino est alimenté en 5V ou en 3,3V. Dans le premier cas, c'est directement la tension d'alimentation du microcontrôleur. Dans le second cas, le microcontrôleur est alimenté via un régulateur 3,3V dont l'entrée est connectée sur le 5V ;
- De la même manière, le 3,3V va être la tension d'alimentation du microcontrôleur pour ceux alimentés en 3,3V et une tension de service pour connecter des dispositifs externes nécessitant du 3,3V si le microcontrôleur nécessite du 5V.

En peut alimenter la carte Arduino par :

- Alimentation par USB et par connecteur Jack
- Alimentation par 2 broches de l'Arduino : VIN et GND [9]

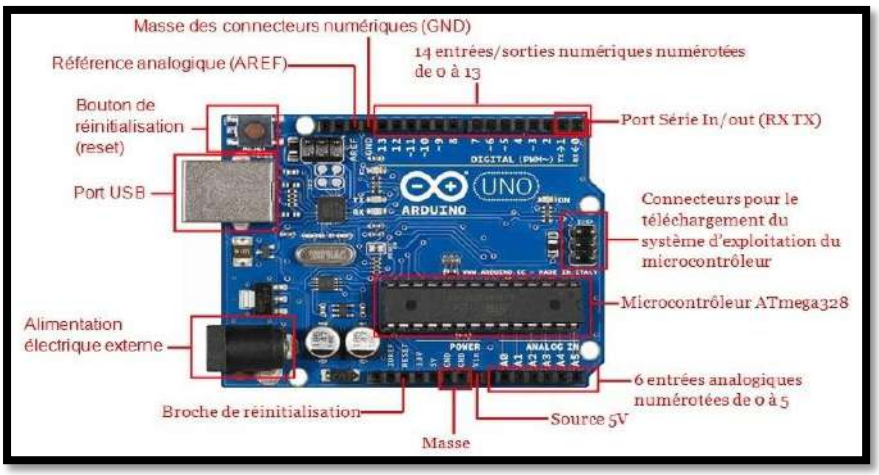
➢ **La visualisation**

On a trois LED qu'on trouve sur la carte donner à deux choses :

❖ **La première (L) :** elle est connectée à une broche du microcontrôleur et va servir pour tester le matériel.

Nota : Quand on branche la carte au PC, elle clignote quelques secondes.

❖ **Les deux LED :** servent à visualiser l'activité sur la voie série (une pour l'émission (TX) et l'autre pour la réception (LX)).



.**Figure 2.1 :** Structure de la carte Arduino Uno.

2.6. Spécifications techniques

Tableau 2.1 : caractéristiques de la carte Arduino Uno.

Microcontrôleur	ATmega328P
Tension de fonctionnement	5V
Tension d'entrée (recommandée)	7-12V
Tension d'entrée (limite)	6-20V
Puces d'E / S numériques	14 (dont 6 fournissent une sortie PWM)
PWM Digital I / O Pins	6
Broches d'entrée analogique	6
Courant continu par broche E / S	20 mA
Courant pour broche de 3.3V	50 mA
Mémoire programme Flash	32Ko(ATmega328P) Dont 0,5 Ko utilisé par boot loader
Mémoire SRAM	2 Ko (ATmega328P)
Mémoire EEPROM	1 KB (ATmega328P)
Vitesse de l'horloge	16 MHz
LED_BUILTIN	13
Longueur	68,6 mm
Largeur	53,4 mm
Poids	25 g

2.7. Le Logiciel de gestion de la carte Arduino

Arduino avoir un environnement spéciale (IDE en anglais) est une application écrite en java inspiré du langage Processing, L'IDE permet d'écrire, de modifier un programme et de convertir en une série d'instructions compréhensibles pour la carte.

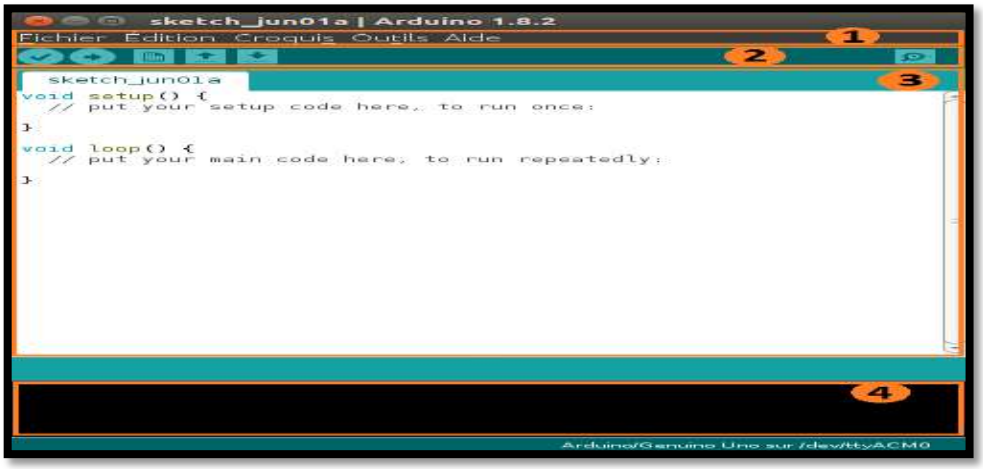


Figure 2.2 : logiciel Arduino.

- Le cadre numéro 1 : options de configuration du logiciel.

- Le cadre numéro 2 : il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes.
- Le cadre numéro 3 : ce bloc va contenir le programme que nous allons créer.
- Le cadre numéro 4 : celui-ci est important, car il va nous aider à corriger les

  Fautes dans notre programme. C'est le débogueur.

➢ **Les boutons**

Voyons à présent à quoi servent les boutons, encadrés en orange et numérotés par le chiffre 2.



Figure 2.3 : Présentation des boutons.

1. Vérifiée le programme.
2. Compile le code et exporte le programme dans le microcontrôleur.
3. Création d'un nouveau fichier.
4. Ouverture d'un fichier.
5. Sauvegarde le fichier.
6. Ouvre le moniteur série.

➢ **Moniteur série**

Le moniteur sériel est utilisé pour afficher l'information qui est envoyée par la carte Arduino vers l'application (habituellement par le câble USB). Il permet aussi d'envoyer de l'information à la carte Arduino. Pour ce faire, il suffit d'écrire du texte dans le champ situé en haut de la fenêtre et d'appuyer sur le bouton « Send ». Bien évidemment, avant de pouvoir faire ceci il vous faut coder un programme approprié, prêt à recevoir cette information. Il est également possible de régler le baud rate du moniteur sériel, qui précise à quelle vitesse le transfert des données s'effectuera. Il est également possible d'établir une communication

sérielle entre la carte Arduino et d'autres périphériques ou logiciels. Nous verrons plus en détails cette fonction dans les projets « Texte brillant » et « Oscilloscope ». [13]

3. La carte Raspberry Pi

Nous présentons dans la suite, le Raspberry pi 3 Model B que l'on a utilisé dans notre projet. Ce dernier est considéré comme le cerveau de notre système parce qu'il va gérer les informations issues de la caméra, les traiter et calculer les lois de commandes nécessaires aux mouvements du robot mobile.

Le Raspberry Pi 3 Model B est un petit ordinateur (Mini-PC) qui fonctionne sous le système d'exploitation Linux sur carte SD destiné à des applications d'informatique embarquée.

Le cœur de l'ordinateur est un FPGA (Broadcom 2836) intégrant un processeur ARMv7 cadencé à 900MHz et de nombreux périphériques.

- **Pourquoi Pi ?**

  • C'est économique.

  • Très facile à utiliser

  • C'est un code libre basé sur Linux avec un accès aux logiciels et téléchargements gratuits pour vous aider avec votre projet.

  • Il existe une grande communauté Pi.

  • Excellente connectivité

  • Flexible pour l'utilisateur. Vous voulez jouer à des jeux ? Diffuser de la musique et/ou des vidéos ? Travailler ? Créer une station météorologique ? Créer une caméra de sécurité domestique ? Aucune limite.

- **Pourquoi Pi 3 B ?**

  • Six fois la puissance de traitement des précédents modèles Pi

  • Démarrage plus rapide

  • Une meilleure compatibilité avec les derniers programmes et logiciels éducatifs

  • Le processeur BCM2837 de Broadcom, cadencé à 1,2 GHz prend en charge toutes les dernières distributions GNU/LINUX ARM, telles que Snappy Ubuntu, et Microsoft Windows 10.

3.1. La carte Raspberry Pi 3 model B

Le Raspberry Pi 3 (modèle B) est un ordinateur miniature de la taille d'une carte de crédit. Il repose sur un processeur à quadruple cœur ARM Cortex-A53, le BCM2837 de Broadcom, cadencé à 1,2 GHz. Cela signifie qu'il est 50 à 60% plus rapide que le Raspberry Pi 2 B. La

communication par Wi-Fi 802.11n et du Bluetooth 4.1 est maintenant intégrée au RPi 3 ; ce nouveau modèle est toujours rétro compatible avec les modèles précédents.[22], les composant de la carte Raspberry pi 3 model B st :

- Processeur BCM2837 de Broadcom, cadencé à 1,2 GHz
- Quadruple cœur ARM Cortex-A53 à 64 bits
- Wi-Fi 802.11 b/g/n intégré
- Bluetooth 4.1 (Classic & Low Energy) intégré
- Coprocesseur multimédia à double cœur Videocore IV
- Mémoire LPDDR2 de 1 Go
- Supporte les distributions GNU/Linux ARM les plus récentes et Windows 10 IoT
- Connecteur micro-USB pour alimentation 2,5 A
- 1x port Ethernet 10/100
- 1x connecteur audio/vidéo HDMI
- 1x connecteur audio/vidéo RCA
- 4x port USB 2.0
- 40 broches d'E/S à usage général
- Antenne sous forme de puce
- Connecteur DSI pour écran
- Lecteur de carte micro-SD
- Dimensions : 85 x 56 x 17 mm

Parce qu'il dispose d'un processeur ARMv7, il peut exécuter la gamme complète de distributions ARM GNU / Linux, y compris Snappy Ubuntu Core, ainsi que Microsoft Windows 10.

Le Raspberry Pi 3 a un facteur de forme identique au Modèle B + antérieur (Pi 1) et a une compatibilité complète avec Raspberry Pi 1. [22]

3.2. Les composants standards de Raspberry

Afin de bien visualiser ce qu'il est possible d'adjoindre à la carte Raspberry Pi nue, la figure 2.5 suivante représente tout à la fois son organisation générale et les différents connecteurs disponible d'un Raspberry standard :

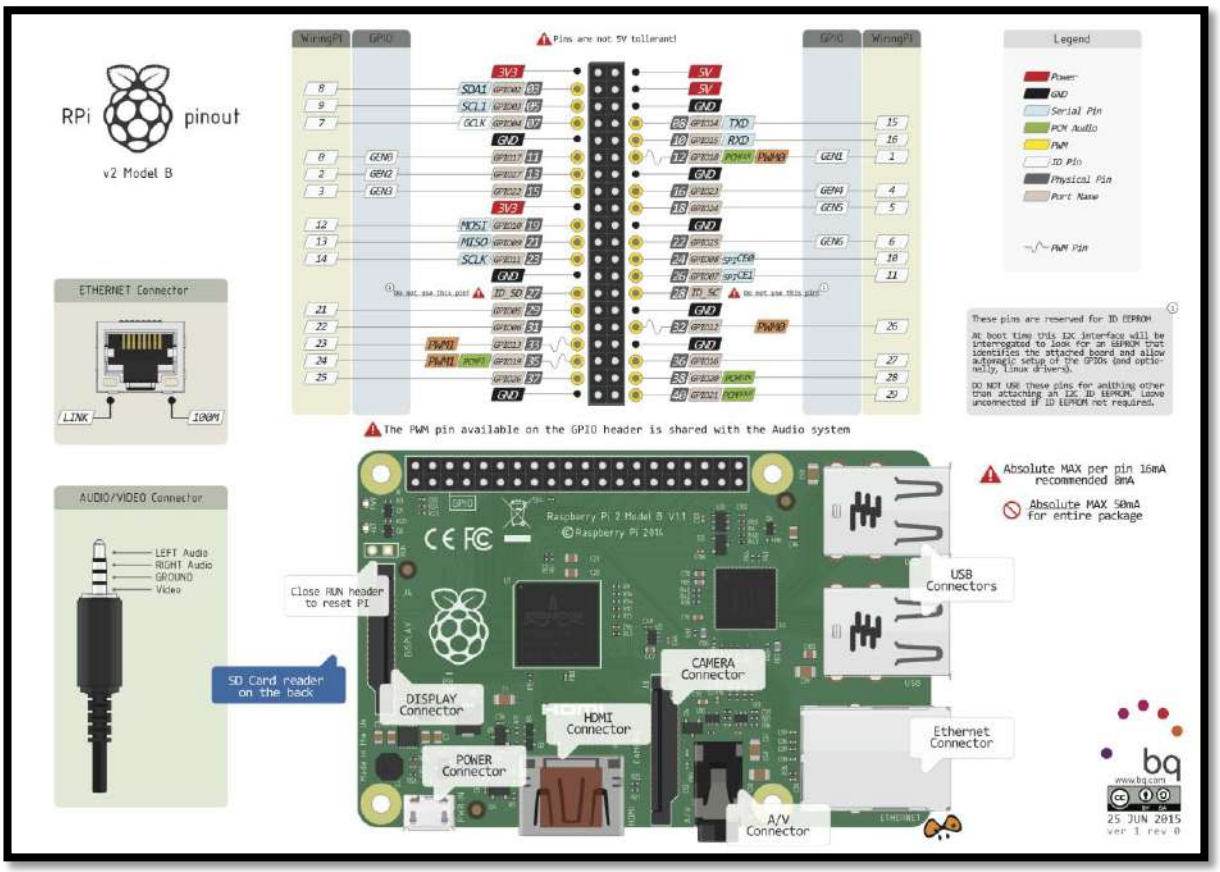


Figure 2.4 : Les composants standards d'un raspberry.

Le centre de la carte est occupé par le processeur ou, plus exactement, par un empilement du processeur et de sa mémoire qui sont littéralement montés l'un sur l'autre. [12]

La connectique qui occupe la majorité de la surface de la carte et non les composants actifs, Dans ce qui va suivre nous détaillerons les différents de la carte Raspberry pi 3 Model B :

- Connecteurs à câble plat flexible (FFC) à la Raspberry Pi 3B :

  - **<u>Connecteur d'affichage</u>,** près du connecteur micro-USB, est l'interface série d'affichage (DSI). Il permet une interface de bas niveau avec les écrans LCD et autres écrans avec Raspberry Pi. Il s'agit d'un connecteur plat flexible monté en surface à 15 broches, offrant deux voies de données, une voie d'horloge, 3.3V et GND.

  - **<u>Connecteur de caméra</u>**, Situé entre le connecteur LAN et HDMI, est le connecteur d'interface MIPI Camera Serial Interface 2 (CSI-2) pour les modules de caméra. Il s'agit d'un connecteur flexible plat à 15 broches

monté en surface, fournissant deux lignes de données, une voie d'horloge, une interface de commande bidirectionnelle compatible avec I2C, 3.3V et GND.

- La prise Ethernet : Raspberry Pi possède une prise Ethernet standard au format RJ 45 et tout réseau Ethernet Classique à 10 Mbits ou 100 Mbits par seconde.
- La prise USB : Utiliser pour connecter n'importe quelle périphérique de type USB comme clavier, souris, carte réseau sans fil...etc .
- HDMI :(Hight Définition Multimédia Interface) permet de relier le Raspberry Pi a un dispositif (écran LCD ou un vidéo projecteur), capable de surcroît de fournir de la vidéo au format full HD de 1080p, et véhicule également le son.
- Le support de carte SD (Secure Digital) :Comme vous pourrez rapidement le constater, le Raspberry Pi ne possède pas de disque dur, tout est stocké sur une carte SD.
- Prise jack audio/vidéo : Prise audio combinée de 3,5 mm et vidéo composite.
- Les broches d'entrées-sorties (GPIO) :Les ports GPIO (General Purpose Input/Output) très utilisés dans le monde des microcontrôleurs et de l'électronique embarquée.
  Ces broches GPIO offre à la Raspberry Pi la possibilité de communiquer avec d'autres circuits électroniques.

3.3. Les GPIO

- **GPIO simples :** entrées /sorties numériques : interrupteurs ON/OFF numériques (manette de jeu, bouton-poussoir, allumer-éteindre lumière ou moteur, etc.)
- **PWM :** « Modulation de la largeur des pics » signaux carres largeur variable : commande moteur pas à pas, variateur lumière LED, variateur vitesse moteur continu, Il y a un seul port GPIO configurable en PWM hardware sur Raspberry Pi, ce dernier est nécessaire pour piloter les servomoteurs (temps de réponse courts pour encodeur en plus régularité précise).
  Pour contourner ce problème : la bibliothèque Wiring PI permet d'ajouter autant de PWM que souhaite par soft (valable pour périphériques autres que servomoteurs).
- **UART :** échange informations série asynchrone (pas d'horloge). Dialogue avec PC Windows(Putty).

- **I2C :** échange informations série synchrone (horloge). Tous les récepteurs au même niveau. → Automobile, matériel hifi.
- **SPI :** échange informations série synchrone (horloge). Récepteur maitre/esclave, maitre tournant possible. → Pilotage écrans LCD.

Nous allons exploiter dans les parties suivantes les caractéristiques décrites précédemment pour relier notre Raspberry au monde extérieur et lui permettre ainsi de commander et de s'interfacer avec la carte Arduino.

Par ailleurs, le Raspberry Pi ne comporte pas de convertisseur analogique/numérique ADC/DAC. Il est impossible de lire directement les signaux des capteurs analogiques. C'est pour cela que l'utilisation de la carte Arduino vient palier cette limitation.

3.4. Mise en marche d'un Raspberry Pi 3

Pour faire fonctionner notre Raspberry Pi, nous avons besoin des accessoire suivants :

- **Une carte mémoire SD**

Cette carte va contenir le système d'exploitation du Raspberry Pi, il ne pourra donc pas démarrer sans cette dernière. La fondation recommande une carte d'au moins 4 Go et de classe 4 minimum. [12]

- **Alimentation micro USB externe**

L'alimentation peut être la cause de nombreux problèmes sur lesquels nous aurons l'occasion de revenir à plusieurs reprises. Le Raspberry Pi ne peut pas être alimenté par le port USB d'un ordinateur ou d'une télévision, il faut par conséquent utiliser une alimentation USB dédiée.

Ce dernier s'alimente sous une tension unique de 5 volts ; tension sur laquelle il peut consommer jusqu'à 750 mA selon les taches qu'il exécute. Cette alimentation doit être normalement fournie via le connecteur micro USB placé dans un angle de la carte. [12]

- **Un clavier et une souris USB**

En cas général, pour faire fonctionner notre Raspberry Pi nous avons besoin d'un clavier et une souris, rien de bien particulier n'est à signaler en ce qui les concerne. N'importe quel clavier USB et n'importe quelle souris USB conviennent au Raspberry Pi et se

connectent directement dans les deux ports USB prévus à cet effet sur le modèle que nous allons utiliser le modèle B.

Mais attention car certains claviers ou certaines souris peuvent être de gros consommateurs d'énergie, il sera donc nécessaire dans ce cas de les brancher sur un HUB USB alimenté par une alimentation externe. C'est souvent le cas par exemple des claviers et des souris sans fils. La souris n'est pas indispensable mais elle devient très utile si on passe du mode texte au mode graphique. [12]

- **Un câble HDMI**

Par défaut, le Raspberry Pi délivre des signaux à la norme HDMI 1.4 comportant vidéo et son. Il peut donc être raccordé directement à tout moniteur informatique ou à tout téléviseur à écran plat disposant d'une telle entrée.

Si l'écran choisi est équipé de haut-parleurs, ces derniers reproduiront le son fourni par le Raspberry Pi puisque celui-ci est véhiculé par la liaison HDMI. C'est bien sur les cas des téléviseurs à écran plat mais pas nécessairement celui des moniteurs informatiques.

Dans une telle situation, nous pourrons alors récupérer le son depuis la prise jack stéréo de 3,5 mm que nous avons vu.

En fonction du type d'écran que nous allons utiliser, nous devons nous procurer le câble correspondant. L'usage de la sortie HDMI est bien entendu recommandé, l'image sera de meilleure qualité et le son sera intégré. [12]

3.5. Le réseau

En effet, tous les logiciels susceptibles d'être utilisés avec le Raspberry Pi sont téléchargeables sur internet ou doivent être mis à jour par ce biais, ce qui rend bien vite cet accès indispensable.

Fort heureusement, si vous disposez d'un réseau filaire, sa connexion au Raspberry Pi est immédiate puisqu'il suffit de faire appel à n'importe quel câble RJ45 standard. L'interface réseau du Raspberry Pi est évidemment de type auto MDI et accepte donc indifféremment les câbles réseau droits ou croisés.

Le Raspberry Pi n'intègre pas en standard un module WIFI donc il faudra faire l'acquisition d'une clé USB-WIFI si nous voulons l'utiliser. [12]

3.6. Préparation de la carte SD

Le Raspberry Pi ne dispose en standard d'aucune mémoire de masse comme nous avons déjà indiquée. Ce rôle est donc dévolu à une carte mémoire de type SD qu'il faut enficher dans le connecteur prévu dans la Figure 2.5 Cette carte mémoire, de type SD ou SD HC, peut avoir toute taille comprise entre 2 Go et 32 Go. Comme elle supporte le système d'exploitation ainsi que les éventuels logiciels supplémentaires que nous allons utiliser avec le Raspberry Pi comme la librairie Pixy.

En effet, pour des raisons inexplicables, même par son concepteur, et alors que l'interface des cartes SD et SD HC est parfaitement normalisée, toutes ne fonctionnent pas nécessairement avec le Raspberry Pi.

Tout comme le disque dur d'un PC doit être préparé avec le système d'exploitation que l'on souhaite utiliser, la carte mémoire destinée au Raspberry Pi doit également être initialisée avec le système d'exploitation choisi.

Malgré que nous pouvons utiliser Windows 10 dans le Raspberry Pi 2B, mais on a gardé Linux comme monde du logiciel libre. Plusieurs versions (on dit habituellement « distribution ») ont été recompilées pour le Raspberry Pi et c'est donc avec ce système d'exploitation que vous allez devoir l'utiliser. [12]

3.7. Système d'exploitation

La plupart des systèmes qui fonctionnent sur Raspberry Pi sont des versions du système d'exploitation Linux. Parce que Linux est open source, les développeurs peuvent l'adopter pour des buts spécifiques. Dans le cas du Raspberry Pi, le matériel léger nécessite un système réduit et privé d'applications inutiles.

La distribution est recommandée Raspbian. C'est un système d'exploitation libre basé sur Debian optimisé pour le matériel Raspberry Pi. Cependant Raspbian fournit plus d'un simple système d'exploitation : il est livré avec plus de 35.000 paquets, des logiciels précompilés qui nous facilitent le développement soft [12].

Tous langages qui peuvent être compilés sur ARMv7 sont installés par défaut sur le Raspberry Pi : Python, C, C++, Java, Scratch et Ruby. La fondation Raspberry Pi recommande Python comme langage pour les étudiants.

3.8. Programmation

Il existe plusieurs langages de programmation qui peut être compilé sur le Raspberry Pi, telle que le langage C, le Java, le JavaScript...etc. Le langage utilisant dans notre projet et le Python.

Le Python est un langage de programmation inventé par Guido van Rossum, la première version de python est sortie en 1991.

C'est un langage de programmation interprété, c'est à dire qu'il n'est pas nécessaire de le compiler avant de l'exécuter.

Python étant un langage très riche, surtout si l'on prend en compte les nombreuses Bibliothèques complémentaires disponibles, Ce dernier est à la fois simple et puissant, c'est un langage facile à apprendre et son code est plus lisible, il est donc plus facile à maintenir. Il est parfois jusqu'à 5 fois plus concis que le langage Java par exemple, ce qui augmente la productivité du développeur et réduit mécaniquement le nombre de bugs. [12]

3.9. Synthèse

La haute performance de l'Arduino due et de Raspberry Pi 3B Insuffisantes pour atteindre le bon performance à notre Robot, Le Raspberry va se charger de la partie acquisition, traitement d'images et calcul des lois de commandes qu'il va transmettre à la carte Arduino, tous ces processus font fonctionner lentement et cela affecte de manière significative à la vitesse de traitement.

Pour résoudre ce problème et réduire la pression sur le Raspberry Ce qui est causée par le traitement d'images. parmi les solutions l'utilisation du Pixy Cam cette caméra possède un processeur graphique qui sera fait le traitement d'image au lieu de faire ça dans le Raspberry et on a gardé comme cerveau pour calcul des lois des commandes .

Dans la partie suivante nous allons présenter la Cam Pixy qui nous allons utiliser dans notre projet.

4. La caméra Pixy

4.1. Aperçu

- Système de vision petit, rapide, facile à utiliser, peu coûteux et facilement accessible.
- Apprends à détecter des objets que vous l'enseignez.
- Indique ce qu'il détecte 50 fois par seconde.

- Se connecte à Arduino avec un câble inclus. Fonctionne également avec Raspberry Pi, BeagleBone et des contrôleurs similaires.
- Toutes les bibliothèques pour Arduino, Raspberry Pi, etc. sont fournies.
- C / C ++ et Python sont pris en charge.
- Communique via l'une des nombreuses interfaces : SPI, I2C, UART, USB ou sortie analogique / numérique.
- L'utilitaire de configuration s'exécute sous Windows, MacOs et Linux.
- Tout logiciel / firmware est open-source GNU-licence.
- Toute la documentation sur le matériel, y compris les schémas, la liste de matériaux, la mise en page des PCB, etc. [15]



Figure 2.5 : Pixy CMUcam5

4.2. Bref historique

Pixy (CMUcam5) est un partenariat entre Carnegie Mellon Robotics Institute et Charmed Labs. Pixy vient d'une longue ligne de CMUcams, mais Pixy a vraiment démarré comme une campagne de Kick Starter. Il a d'abord commencé à être livré en mars 2014, mais c'est déjà devenu le système de vision le plus populaire de l'histoire. [15]

4.3. Simplicité de mise en œuvre du système de vision

Si vous souhaitez que votre robot effectue une tâche, comme la collecte d'un objet, la poursuite d'une balle, la localisation d'une station de chargement, etc., et que vous souhaitez

qu'un seul capteur aide à accomplir toutes ces tâches, la vision est votre capteur. Les capteurs Vision (image) sont utiles car ils sont si flexibles. Avec le bon algorithme, un capteur d'image peut détecter ou détecter pratiquement n'importe quoi. Mais il existe deux inconvénients avec les capteurs d'image : 1) ils sortent beaucoup de données, des dizaines de mégaoctets par seconde, et 2) le traitement de cette quantité de données peut submerger de nombreux processeurs. Et si le processeur peut suivre les données, une grande partie de sa puissance de traitement ne sera pas disponible pour d'autres tâches.

Pixy résout ces problèmes en associant un puissant processeur dédié au capteur d'image. Pixy traite les images du capteur d'image et envoie uniquement les informations utiles (par exemple, le dinosaure violet détecté à x = 54, y = 103) à votre microcontrôleur. Et cela se fait à cadence (50 Hz). L'information est disponible via l'une des nombreuses interfaces : UART série, SPI, I2C, USB ou sortie numérique / analogique. Ainsi, votre Arduino ou un autre microcontrôleur peut facilement parler avec Pixy et dispose de nombreuses CPU disponibles
pour d'autres tâches.

Il est possible de brancher plusieurs Pixys sur votre microcontrôleur - par exemple, un robot avec 4 Pixys et 360 degrés de détection. Ou utiliser Pixy sans microcontrôleur et utiliser les sorties numériques ou analogiques pour déclencher des événements, des commutateurs, des servos, etc. [15]

4.4. Support du contrôleur

Pixy peut facilement se connecter à de nombreux contrôleurs différents car il prend en charge plusieurs options d'interface (UART série, SPI, I2C, USB ou sortie numérique / analogique), mais Pixy a commencé sa vie à parler à Arduino. Au cours des derniers mois, nous avons ajouté un support pour Arduino Due, Raspberry Pi et BeagleBone Black. Des bibliothèques de logiciels sont fournies pour toutes ces plates-formes afin que vous puissiez vous lancer rapidement. En outre, nous avons ajouté une API Python si vous utilisez un contrôleur basé sur Linux (par exemple, Raspberry Pi, BeagleBone). [15]

4.5. La détection d'objets

Pixy utilise un algorithme de filtrage basé sur les couleurs pour détecter les objets. Les méthodes de filtrage basées sur les couleurs sont populaires car elles sont rapides, efficaces et

relativement robustes. La plupart d'entre nous connaissent le RGB (rouge, vert et bleu) pour représenter les couleurs. Pixy calcule la couleur (teinte) et la saturation de chaque pixel RGB du capteur d'image et les utilise comme paramètres de filtrage primaires. La teinte d'un objet reste en grande partie inchangée avec les changements d'éclairage et d'exposition. Les changements d'éclairage et d'exposition peuvent avoir un effet frustrant sur les algorithmes de filtrage des couleurs, ce qui leur permet de se casser. L'algorithme de filtrage de Pixy est robuste en matière d'éclairage et de changements d'exposition.

Pixy peut trouver littéralement des centaines d'objets à la fois. Il utilise un algorithme de composants connectés pour déterminer où un objet commence et un autre se termine. Pixy compile les tailles et les emplacements de chaque objet et les signale à travers l'une de ses interfaces (par exemple SPI).

Dans la documentation technique de ce système de vision, nous retrouvons souvent le terme "50 images par seconde" ou "50 frames par seconde". En bref, cela signifie que Pixy est rapide. Pixy traite toute une image 640x400 toutes les 1 / 50ème de seconde (20 millisecondes). Cela signifie que vous obtenez une mise à jour complète de toutes les positions des objets détectés toutes les 20 ms. À ce rythme, il est possible de suivre le parcours de la balle tombante / rebondissant. (Une balle qui se déplace à 30 mph se déplace moins d'un pied en 20 ms.) [15]

4.6. Définition simple de l’objet à suivre

Pixy est unique du fait que l’on peut rapidement et simplement définir l’objet à suivre. Par exemple si cet objet est un dinosaure violet placez le dinosaure devant l’objectif de la caméra Pixy et appuyez sur le seul bouton. Boule orange ? Placez le ballon devant Pixy et appuyez sur le bouton. C'est simple, et c'est rapide.

Plus précisément, vous enseignez à Pixy en tenant l'objet devant son objectif tout en maintenant le bouton situé en haut. En faisant cela, la LED RGB sous l'objectif fournit des commentaires concernant l'objet qu'il regarde directement. Par exemple, la LED devient orange lorsqu'une boule orange est placée directement devant Pixy. Relâchez le bouton et Pixy génère un modèle statistique des couleurs contenues dans l'objet et les stocke en flash. Il utilisera ensuite ce modèle statistique pour trouver des objets avec des signatures de couleurs similaires dans leur cadre à partir de là.

Pixy peut apprendre sept signatures couleur, numérotées 1-7. La signature couleur 1 est la signature par défaut. Pour enseigner à Pixy, les autres signatures (2-7) nécessitent un simple bouton qui appuie sur la séquence. [15]

4.7. L'application PixyMon

PixyMon est une application qui fonctionne sous Windows, MacOs et Linux. Il vous permet de voir ce que Pixy voit, en tant que vidéo brute ou transformée. Il vous permet également de configurer votre Pixy, de définir le port de sortie et de gérer les signatures des couleurs. PixyMon communique avec Pixy sur un câble mini USB standard.

PixyMon est idéal pour le débogage de votre application. Vous pouvez brancher un câble USB à l'arrière de Pixy et exécuter PixyMon, puis voir ce que Pixy voit lorsqu'il est accroché à votre Arduino ou à un autre microcontrôleur - pas besoin de débrancher quoi que ce soit. PixyMon est open source, comme tout le reste. [15]

4.8. Spécifications techniques

La puissance du système de vision Pixy est l'intégration sur une seule carte à la fois du système optique de vision est des composants électroniques (processeur, RAM, …) nécessaires à l'acquisition et aux traitements des images. [15]

-Processeur: NXP LPC4330, 204 MHz, dual core
-Capteur d'image: Omnivision OV9715, 1/4 ", 1280x800
-Lens field-of-view: 75 degrés horizontal, 47 degrés vertical
-Type d'objectif: standard M12 (plusieurs types différents disponibles)
-Consommation électrique: 140 mA typique
-Entrée d'alimentation: entrée USB (5V) ou entrée non régulée (6V à 10V)
-RAM: 264K octets
-Flash: 1M octets
-Sorties de données disponibles: UART série, SPI, I2C, USB, numérique, analogique
-Dimensions: 2.1 "x 2.0" x 1.4
-Poids: 27 Grammes

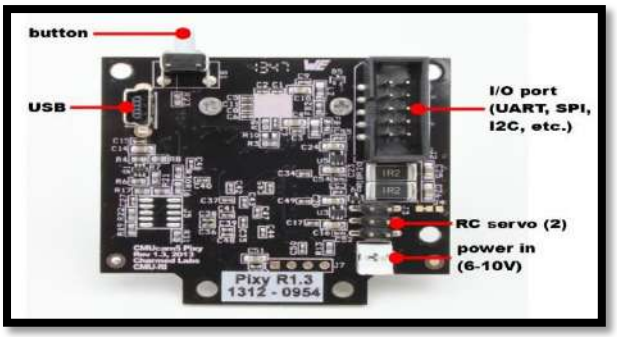


Figure 2.6 : structure de la Cam Pixy CMUcam5.

5. Schéma synoptique du robot mobile utilisé

Le robot utilisé dans notre application est constitué principalement d'un Raspberry de troisième génération (Pi 3). Ce dernier aide comme le cerveau de notre robot, il est chargé d'exécuter et affecter les programmes pour les différents organes. Il assure également le contrôle et l'interaction entre eux. Ce robot regroupe aussi, deux Arduino de type (uno), l'un est conçu pour commander les deux moteurs à travers un shield moteur et l'autre pour récupérer les donnes des paramètres des capteurs (température, humidité et lumière) qui sont attachés au robot à travers cet Arduino. La camera pixy qui est installé sur cet robot est pour sélectionné l'objet qui doit être suivi par le robot (voir figure : 2.7)

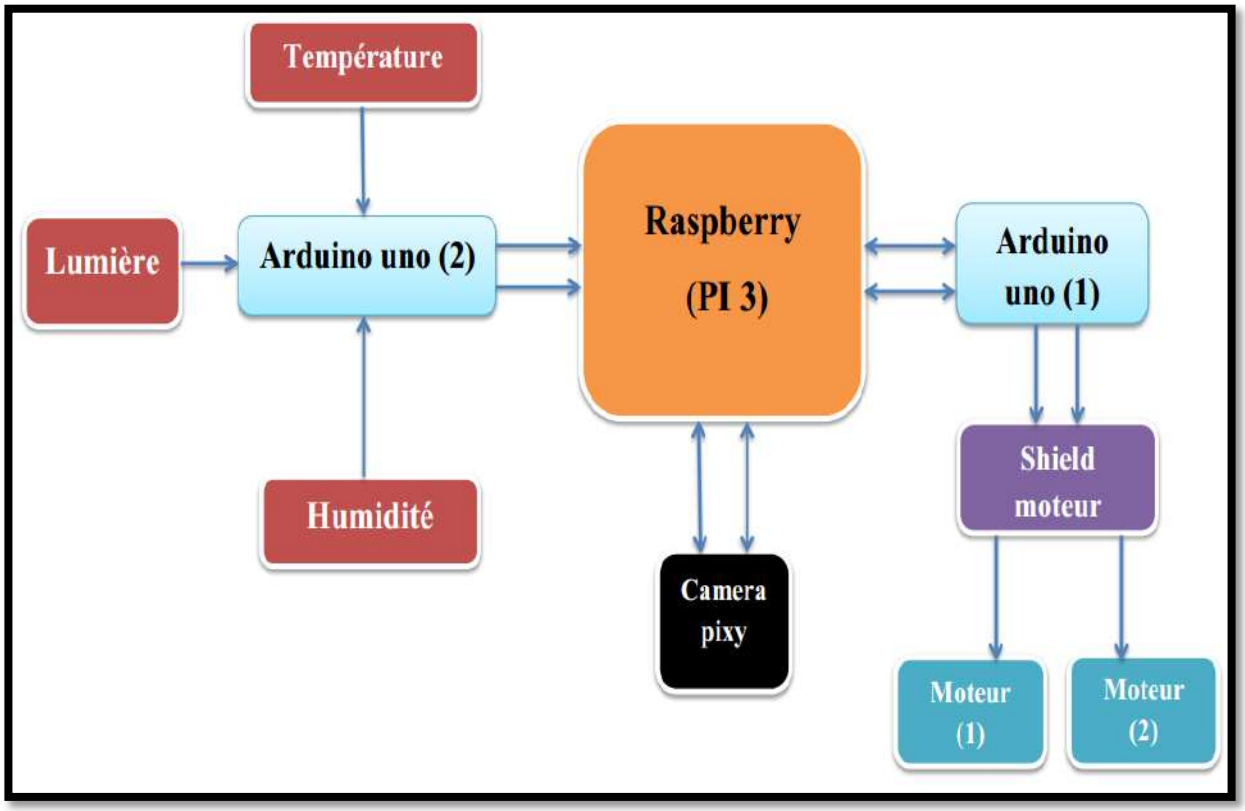


Figure 2.7 : Schéma synoptique du robot mobile utilis

6. Conclusion

Ce chapitre a été pour nous l'occasion de présenter les deux cartes Raspberry PI et Arduino ainsi que le système de vision Pixy que nous avons utilisé pour contrôle et supervision du robot mobile.

La supervision de ce robot basé sur l'utilisation des données visuelles, nécessite d'une part une puissance de calcul importante et d'autre part des calculateurs peu volumineux et léger pouvant être embarqués sur la structure mécanique du robot mobile. De plus, la gestion de cette plate-forme nécessite le développement d'un interface Homme / Machine pour faciliter et simplifier le contrôle et la gestion de cette plate-forme.

Mais avant de rentrer dans les détails de conception de cette interface, il serait important d'introduire et de présenter, dans le prochain chapitre, les différents langages de développement d'interfaces Homme / Machine.

Chapitre3 : les langage de programmation et logiciels d'interfaçage

1. Introduction

L'interface homme /machine (IHM) est le principal outil au travers duquel les opérateurs et les superviseurs peuvent coordonner et contrôler les processus industriels dans les usines. Les interfaces Homme-Machine servent à transformer les variables complexes des processus en informations utilisables et exploitables. Les graphiques visuels des performances des processus donnent un sens et un contexte aux informations recueillis.

Dans ce présent chapitre nous illustrerons les différents types de logiciel les plus connus dans le domaine du développement d'interface web. Cette présentation nous permettra de présenter et de justifier l'utilisation de la plate-forme de développement Node-Red pour mettre en œuvre l'interface de contrôle et de supervision du robot mobile.

2. Définition l'interface homme /machine (IHM)

IHM est un acronyme utilisé pour désigner les interfaces homme/ machine, c'est à dire l'ensemble des moyens utilisés par l'homme pour communiquer avec une machine. L'IHM s'applique essentiellement à l'informatique et comprend essentiellement les périphériques, logiciels et autres applications permettant d'utiliser les ordinateurs.[17]

3. Les différents types du logiciel de programmation

a. java

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp).[18]

b. java script

Le Java script est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web.[18]

Ainsi le langage Java script est fortement dépendant du navigateur appelant la page web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage Java, avec lequel il a longtemps été confondu.

c. Php

Ce langage est principalement utilisé pour produire un site web dynamique. Il est courant que ce langage soit associé à une base de données, tel que MySQL.
Exécuté du côté serveur (l'endroit où est hébergé le site) il n'y a pas besoin aux visiteurs d'avoir des logiciels ou plugins particulier. Néanmoins, les webmasters qui souhaitent développer un site en PHP doivent s'assurer que l'hébergeur prend en compte ce langage.[19]

d. Python

Python est un langage de programmation interprété, orienté objet, similaire à PERL, qui a gagné en popularité grâce à sa syntaxe et à sa lisibilité. Python est relativement facile à apprendre et portable, ce qui signifie que ses instructions peuvent être interprétées dans un certain nombre de systèmes d'exploitation, y compris les systèmes UNIX, Mac OS, MS-DOS, OS / 2 et diverses versions de Microsoft Windows 98. Python a été créé par Guido van Rossum, un ancien résident des Pays-Bas, dont le groupe de comédie préféré à l'époque était Flying Circus de Monty Python. Le code source est librement disponible et ouvert pour modification et réutilisation. Python a un nombre important d'utilisateurs.[20]

e. Node –Red

Est un langage de programmation graphique par assemblage de blocs fonctionnels qui permet de développer des objets connectés (et beaucoup d'autres choses). Node-Red est un projet Open Source (gratuit) soutenu par IBM. On programme avec **Node-RED** en liant des fonctions présentées sous la forme de briques. Le flux de données passe d'un traitement à l'autre (d'une fonction à l'autre). Certaines fonctions proposent des paramètres qu'il suffit de définir à l'aide d'une liste de choix ou d'un champ à remplir (par exemple une clé d'API pour un service météo). Il existe plusieurs centaines de plugins dans tous les domaines : enregistrer les données sur une base de données (MySQL, Mongo DB…), piloter les E/S d'un Arduino ou d'un Raspberry (avec le plugin Johny Five par exemple), ajouter une interface graphique (UI) pour tracer des graphiques, afficher des jauges, commander un relai à l'aide d'un bouton…[16]

4. Pourquoi Node-RED?

- Internet n'a pas de solution unique pour tous

- Chaque nouvelle "chose" a une nouvelle API qui doit être comprise
- Les solutions nécessitent souvent de rassembler plusieurs dispositifs différents API et services en ligne de nouvelles façons intéressantes
- Temps passé à réfléchir sur la façon d'accéder à un port série, ou à compléter un flux sur Twitter n'est pas le temps passé à créer le réel temps d'une solution .[21]

5. Définition Node-Red

Node-Red (NR en English) est un outil de programmation et un cadre pour le câblage de périphériques matériels, d'API et de services en ligne d'une manière nouvelle et intéressante. Il fournit un éditeur basé sur un navigateur qui facilite le câblage des flux en utilisant le large éventail de nœuds de la palette qui peut être déployé en un seul clic. Il est écrit dans Node.JS et utilise beaucoup de modules dans la bibliothèque npm.[16]

6. Historique

Comme mentionné ci-dessus, Node-RED est un outil puissant pour construire des applications et des services Internet. L'effet est omniprésent dans la suite d'IBM. Où ils ont initialement créé Node-RED comme un outil pour eux-mêmes et ils sont professionnels sur les projets IOT.

La version initiale de Node-RED a été lancée en tant que projet ouvert fin 2013 et la communauté des utilisateurs et des développeurs a grandi en 2014.

L'industrie a été largement adoptée par les fabricants, les expérimentateurs et un certain nombre de grands et petits partenaires qui les ont utilisés pour répondre à leurs besoins.[16]

Quand IBM a mis en place le Node-RED, il s'est surtout concentré sur les objets Internet, c'est-à-dire l'accès matériel aux processeurs ou vice versa. Node-RED est un outil puissant et flexible pour développer des applications Internet rapides.

7. les avantages et les inconvénients de Node-Red

7.1. Les principaux avantages de Node-Red

Ces dernières années l'approche REST devient l'architecture incontournable des web en utilisant la puissance du protocole HTTP. Parallèlement on assiste à une reconfiguration du paysage côté serveur notamment avec la percée de Node.js

Node s'est imposé comme une des principales techniques car il permet de créer facilement un petit serveur HTTP en une ligne de commande [17],[21].

- **Une technologie performante pour le Web :**

- Si Node.js a su percer dans le domaine des applications web, c'est en partie parce que celui-ci a introduit un changement de paradigme par rapport aux stacks classiques, « révolution » à l'origine de gains de performances notables.

- Vecteur de performance : L'approche événementielle couplée à des IO non bloquant font qu'il n'y a plus besoin de plusieurs threads pour traiter des requêtes de manière concurrente. Ceci permet d'économiser en changements de contexte, et over Head qu'entraîne la gestion de plusieurs threads.[17].

❖ **Une technologie productive et industrialisée**

- La simplicité au cœur, Philosophie et Architecture : Au cœur de la philosophie de node.js, on retrouve la simplicité,. Les applications sont structurées en modules de taille modeste et réutilisable, se concentrant sur une fonctionnalité dans la lignée de la philosophie Unix venant enrichir le cœur de node[17].

- JavaScript au cœur du succès de la plateforme : Certes le langage est loin d'être parfait. De plus node.js est né de la volonté de faire un serveur à boucle d'événement, et non de l'envie de faire du JavaScript coté serveur.

❖ **Une communauté active et qui mûrit vite**

Node en tant que projet open source bénéficie aussi à la fois d'un fort soutien communautaire, mais aussi d'un bon support industriel [17]
Il est possible de créer un nœud et programmer vos propres fonctionnalités puis rendre ce nœud téléchargeable et installable sur Node-Red.

7.2. Les inconvénient de Node-Red

❖ Il y a plusieurs fonctionnalités de programmation web qui ne sont pas disponible sur Node-Red (exp : des limitations pour varier la couleur, l'image et la taille d'un bouton)

❖ Pour construire un nouveau nœud et rendre le installable il est demandé d'avoir une bonne connaissance des langages de programmation de base (Java script et l'HTML)

8. Les Nœuds composant Node Red

Les nœuds de NR sont composés d'un fichier HTML et d'un fichier JavaScript. Le fichier HTML indique à quoi le Nœud doit ressembler, combien d'entrées / sorties doit utiliser. Le fichier JavaScript est responsable de la façon dont le nœud devrait fonctionner. NR offre une variété des nœuds préinstallé qui peuvent être utilisés. Aussi il est possible de créer des nœuds de votre choix, spécifications et fonctionnalité. Vous pouvez aussi partager vos propres nœuds avec la communauté. [17].

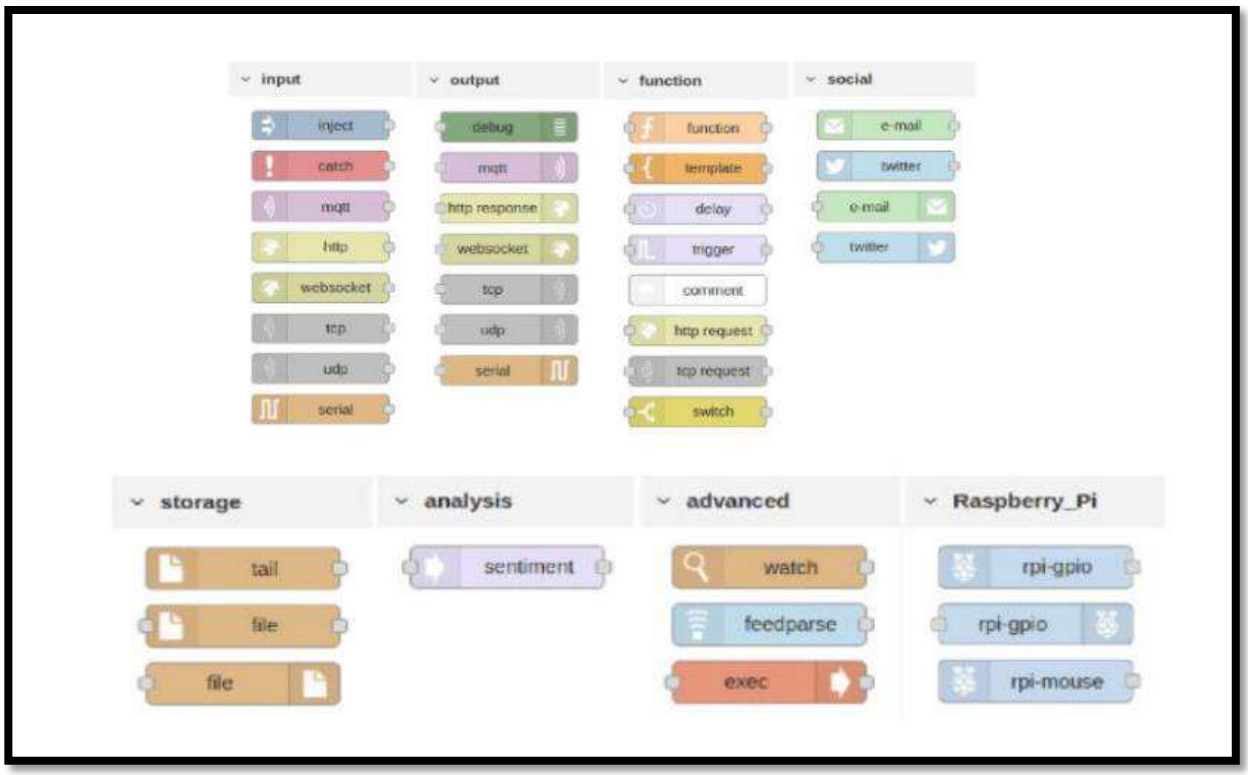


Figure 3.1: Certains des nœuds disponibles dans Node-Red.

- **Il y a 7 nœuds d'entrée de base dans l'installation par défaut. Ces nœuds couvrent les mécanismes de communication de base utilisée principalement par les applications IoT. Cela s'étale des protocoles Internet fondamentaux tels que UDP et TCP jusqu'à un niveau plus élevé tels que HTTP et les mécanismes MQTT de publication / souscription (publish/subscribe).[17]**

- (Output nodes) Les nœuds de sortie : Les nœuds de sorties (*output* en anglais) sont les pendants des nœuds d'entrée. Les nœuds de sortie offrent des moyens d'envoyer des données sur le même ensemble de protocoles (ex: mqtt, http, etc.).

- Les nœuds de fonction : La catégorie "fonction" contient une série de nœuds supportants des fonctions de traitement. Cela va d'un simple nœud délai (*delay*), à un

nœud d'aiguillage (*switch*), à nœud de fonction programmable qui peut être adapté à presque tous les besoins de programmation.

- (Social node) Les nœuds sociaux : Les nœuds "media sociaux" supporte les interactions avec l'email et avec Twitter. Ces nœuds permettent d'envoyer ou recevoir des emails, ou d'envoyer et recevoir des tweets.

- **(Storage node) Nœud de stockage : Etant donné que nous utilisons une version pour Raspberry-Pi, le contenu de la section "stockage" est plus restreint que sur une installation normale.**

- **(analysais node) Nœud d'analyse : Les nœuds d'analyse effectue des analyses standard sur les messages entrants. Seul le nœud "*sentiment*" est disponible dans l'installation par défaut. Le nœud "sentiment" peut être utilisé pour essayer de déterminer le "sentiment" en fonction des mots contenu dans le message entrant, par exemple, un email ou tweet**

- (advanced node) Nœud avancés: Il existe également des nœuds offrant des fonctionnalités avancées.

- (Dashboard) Tableau de bord: Le tableau de bord est une palette se compose de plusieurs nœuds, avec un maximum de 12. Chaque nœud peut lui-même être personnalisé. Cette palette peut être utilisée comme tableau de bord d'administration qui affiche l'état de plusieurs services. Autre exemple : le tableau de bord peut afficher les différents types d'état des tâches ainsi que le nombre de tâche figurant actuellement sous chaque type d'état dans une application de suivi des tâches.[17]

9. Quelque exemple de sketch node red

9.1. Commande d'un moteur DC connecté à un Arduino uno

Dans cet exemple, nous allons créer un flux (sketch) simple qui envoie un ordre qui démarrer ou arrêter un moteur DC. Les nœuds utilisés dans cet exemple appartiennent aux palettes Dashboard et Arduino du Node-red.

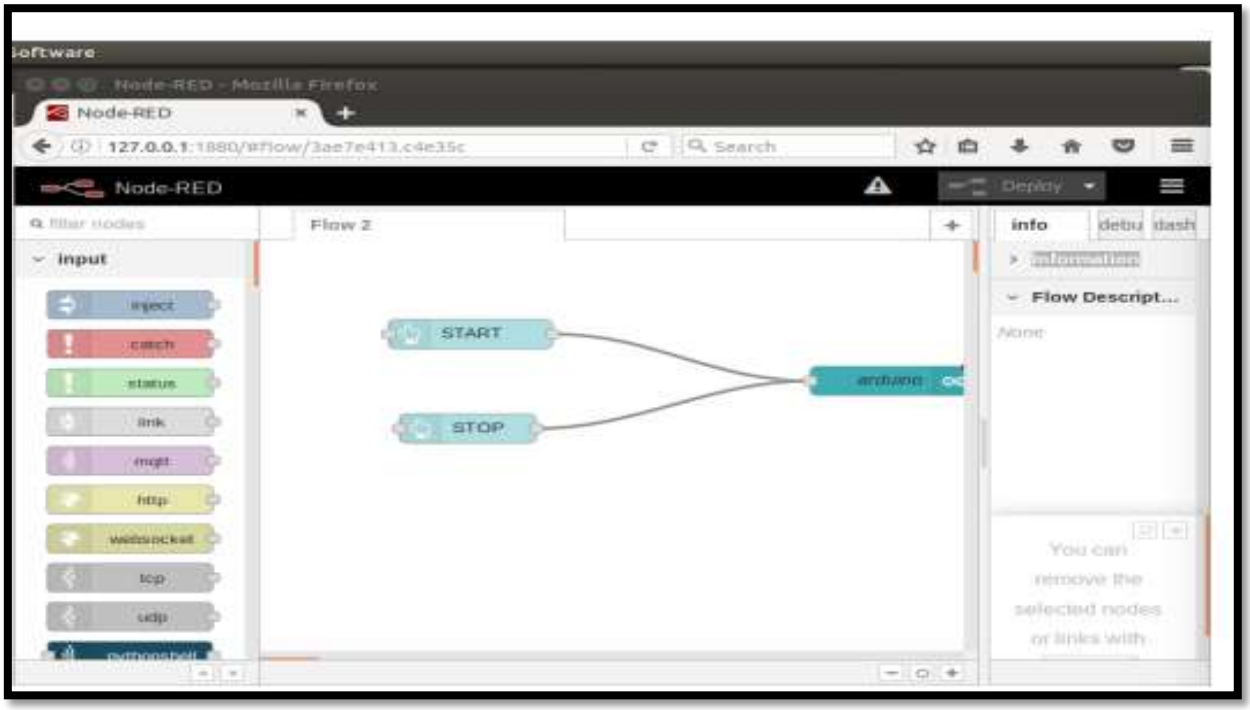


Figure 3.2 : Exemple commande moteur dc

- **<u>Configuration du flux :</u>**

Le nœud Botton changera le nom en START, le nœud Botton changera le nom en "STOP". Connectez les nœuds en cliquant sur la petite boîte ronde pour chaque nœud pour obtenir le même résultat Comme dans la figure ci-dessus.

Avant de voir comment fonctionne le sketch au-dessus, le moteur doit être connecté à l'Arduino. Le placement des câbles de moteur avec l'Arduino peut être comme suite : l'entrée positive du moteur se connecter sur le pin digital 13 de l'Arduino et l'entrée négative doit connecter sur un pin des GND de l'Arduino.

Double-cliquez sur le nœud nommé START du sketch au-dessus pour entrer dans les options de configuration. Définissez les options comme dans la Figure (3.3). Faire la même chose pour configurer les deux autres nœuds nommés STOP et DC MOTOR, voire respectivement les figures suivantes (3.4) et (3.5):

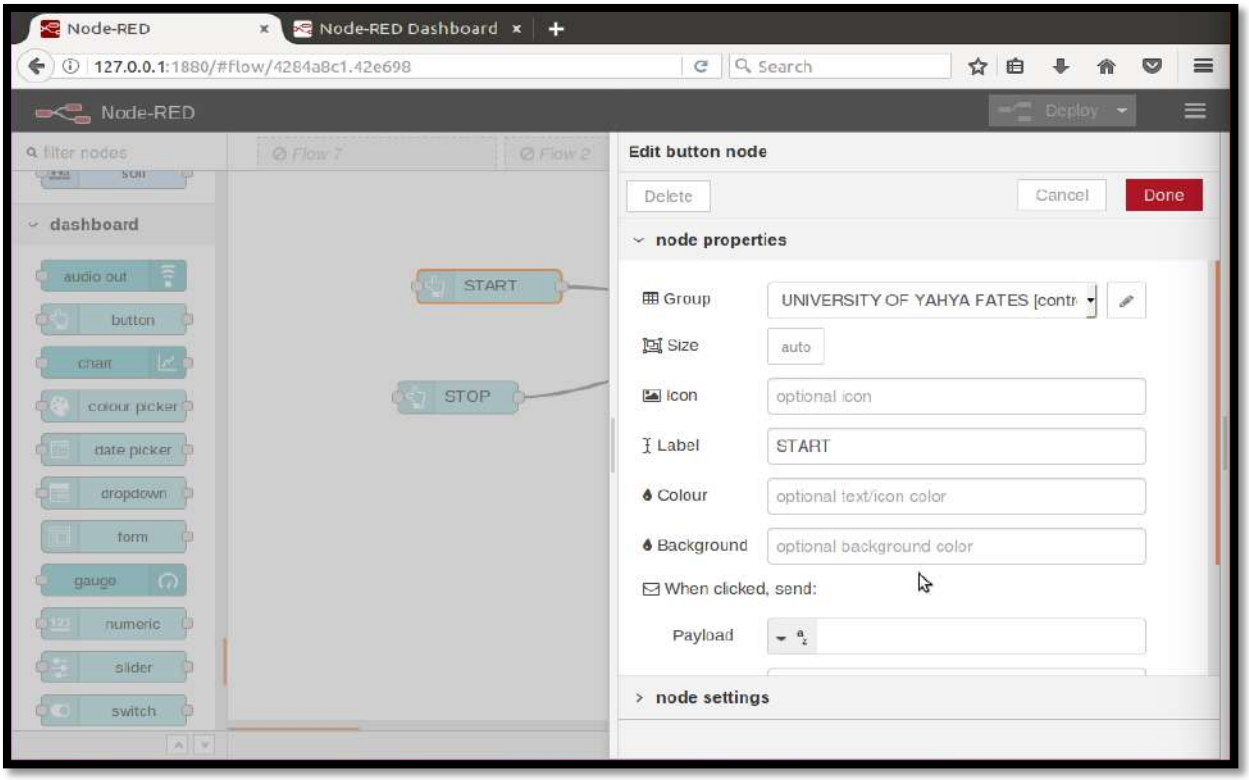


Figure 3.3 : Configuration du nœud nommé ‘START’

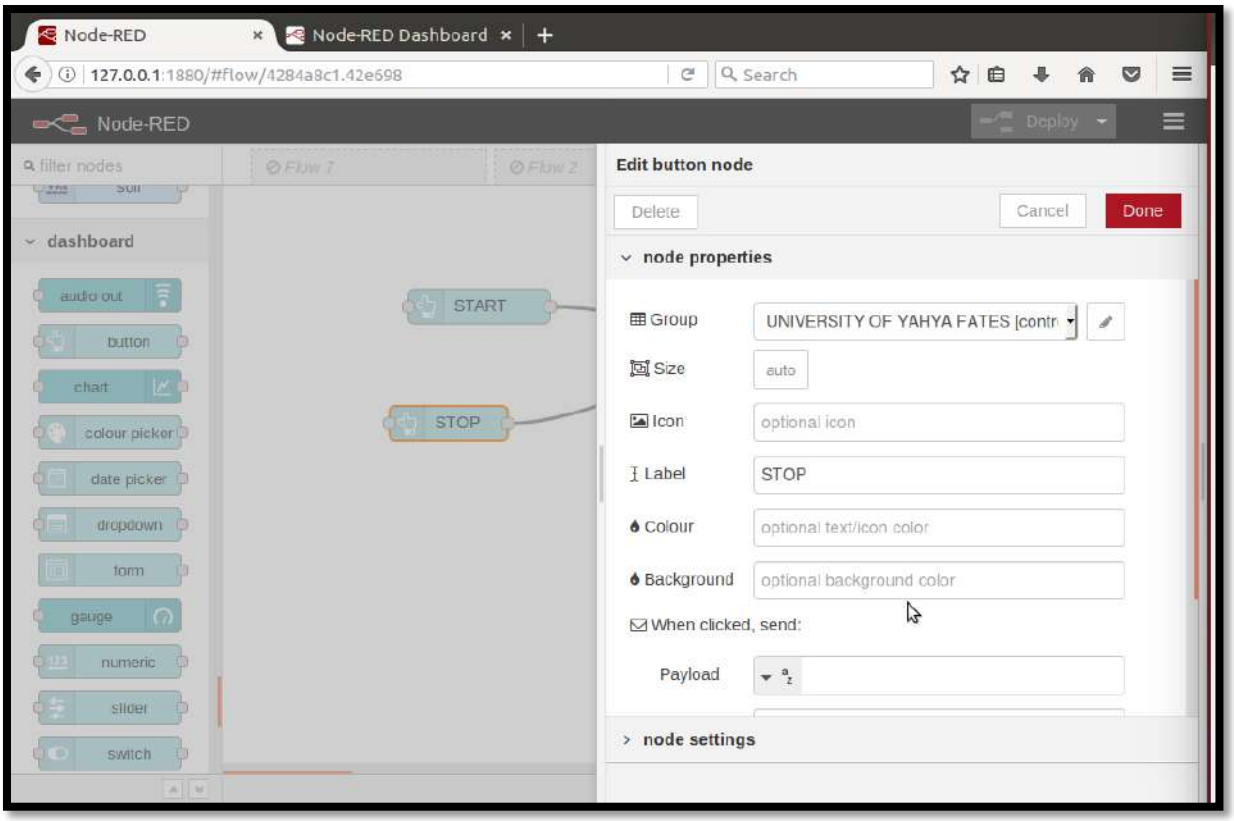


Figure 3.4 : Configuration du nœud nommé ‘STOP’

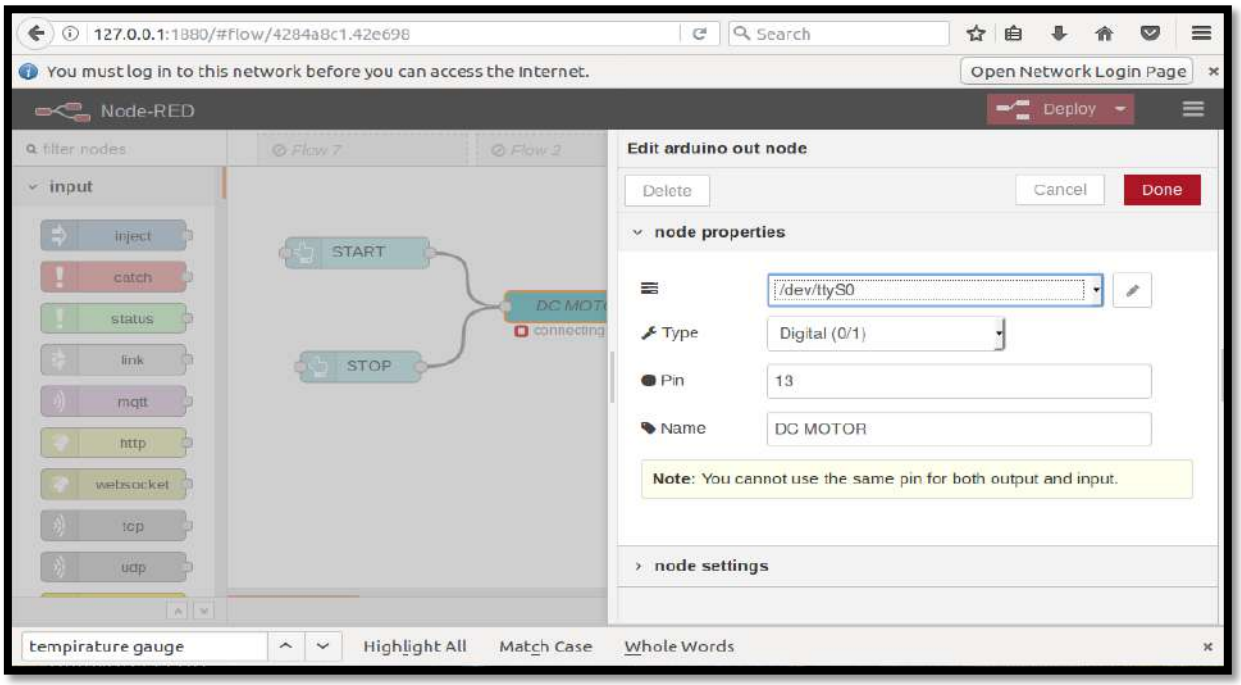


Figure 3.5 : Configuration du nœud nommé 'DC MOTOR'

Enfin, en tapant l'adresse http://127.0.0.1:1880/ui/#/0 pour démarrer l'interface node – red pour le control de moteur DC (figure 3.6). Donc, pour démarrer ou arrêter ce moteur, il suffit de cliquer respectivement sur le bouton START ou STOP.

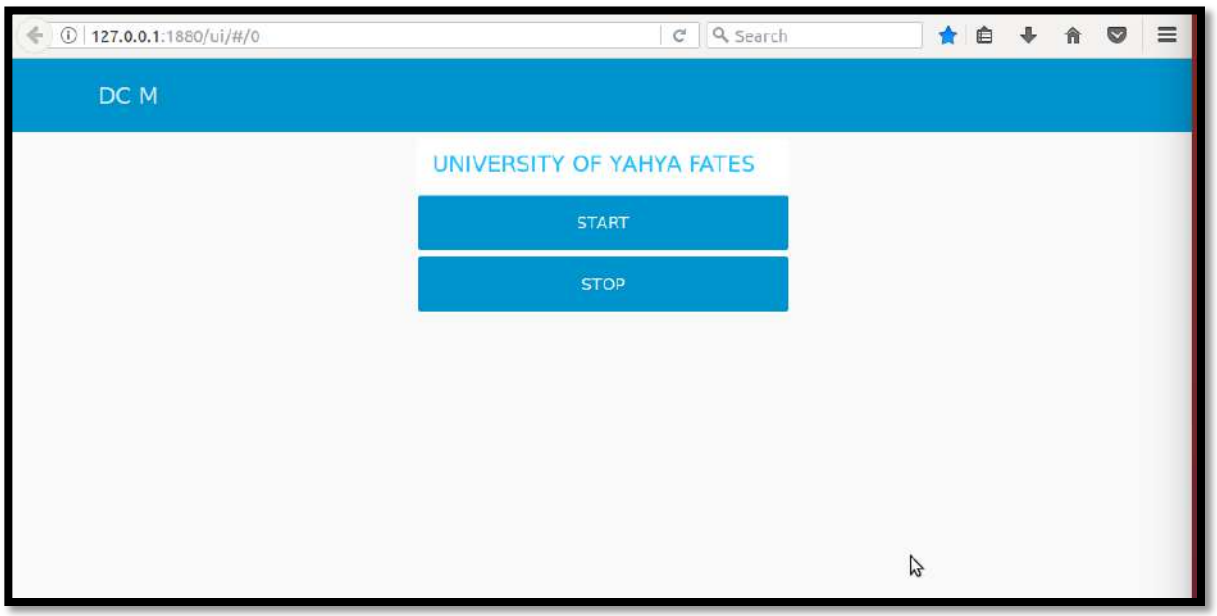


Figure 3.6 : Interface de commande moteur DC

9.2. Capteur de température

Dans cet exemple on vous montrer comment récupérer les lire la température en utilisant le node-red. On est besoin d'un Arduino pré-chargé par le Protocol Standard firmata et un capteur de température. Faire glisser les deux nœuds suivant Arduino in et gauge, puis raccorder les deux comme illustre la figure (3.7).

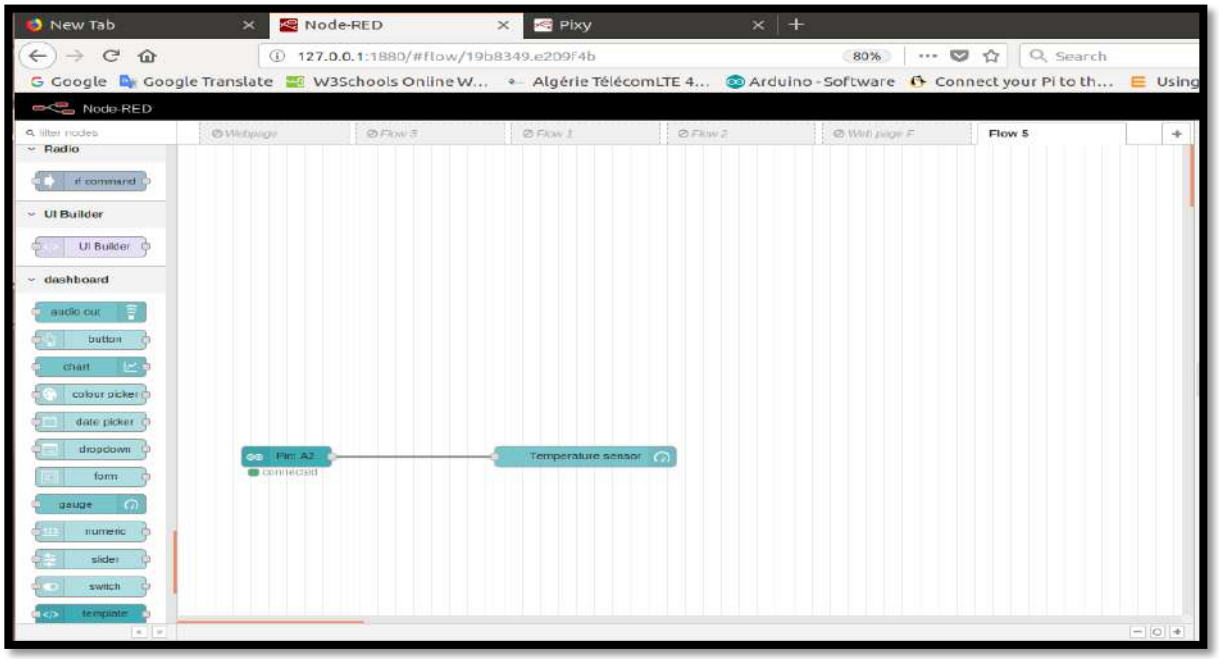


Figure 3.7 : Exemple lecture de température

- **<u>Configuration du flux :</u>**

Configurer les deux utilisés ('Arduino in' et Température gauge) conformément au figures suivants figure 3.8, figure 3.9

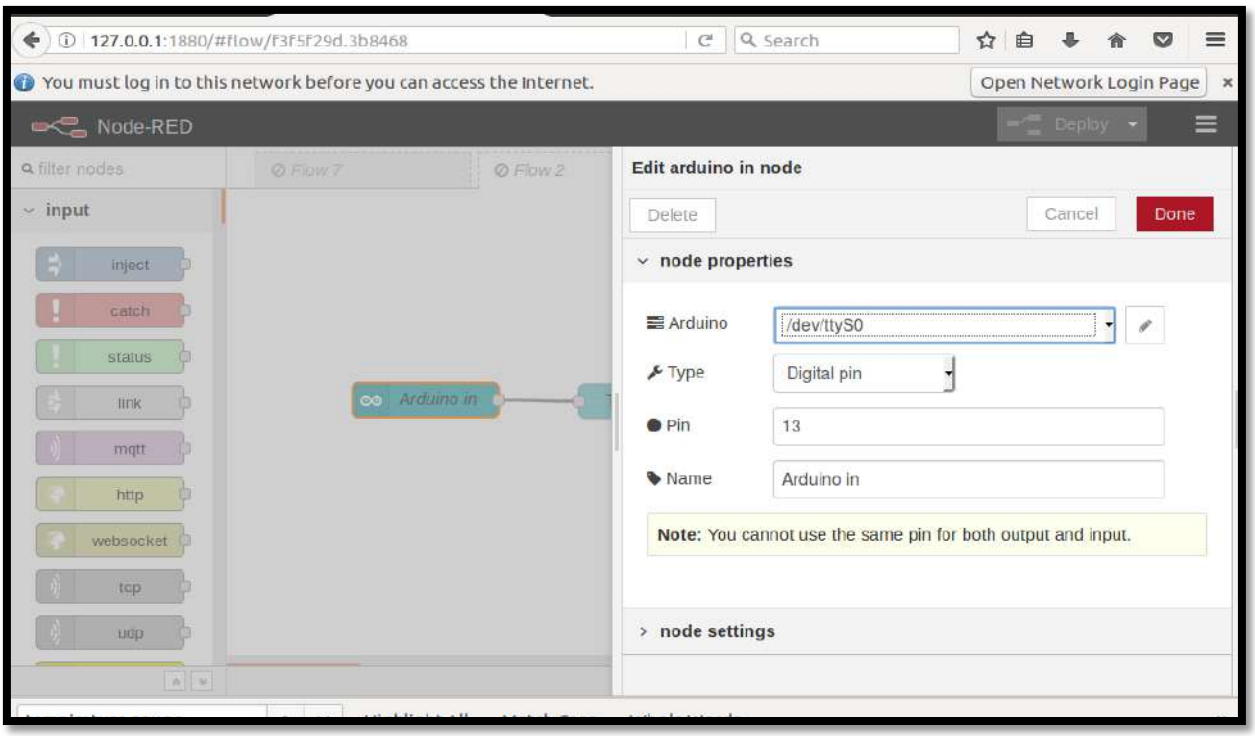


Figure 3.8 : Configuration de la nœud nommée (‘Arduino in’)

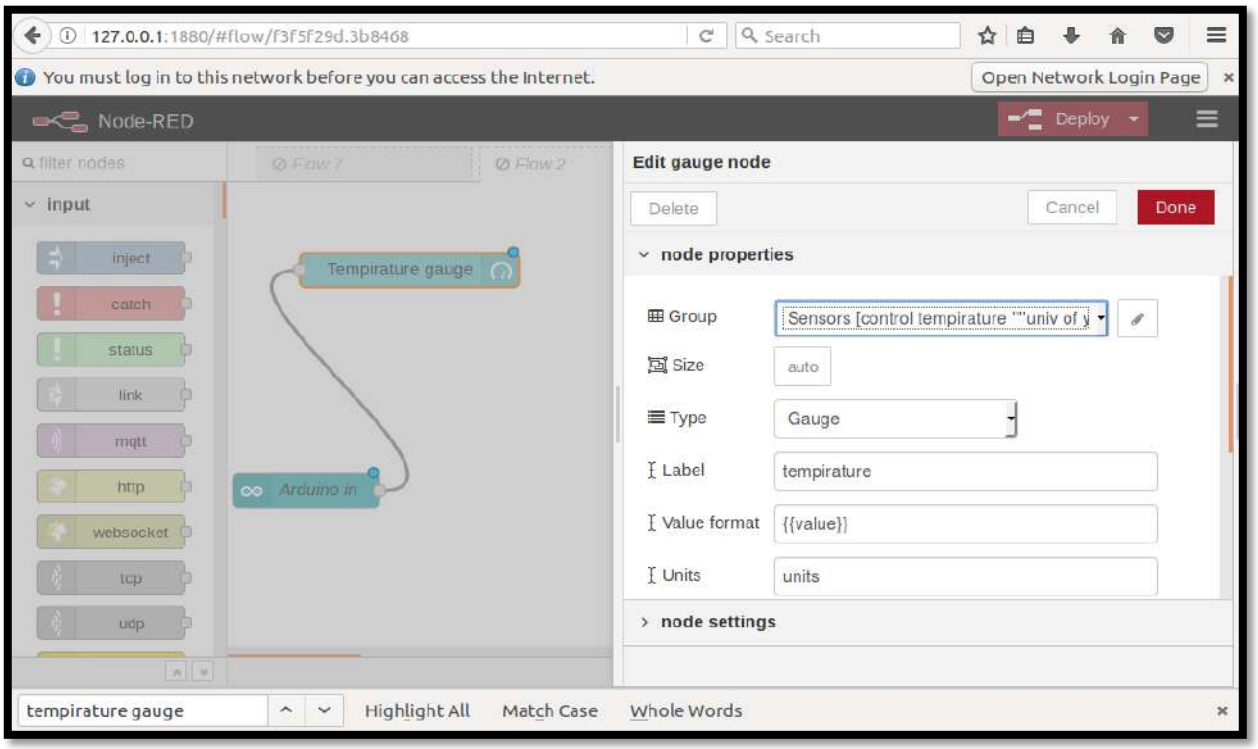


Figure 3.9 : Configuration du nœud nommée ‘Température gauge‘

Taper l'addresse http://127.0.0.1:1880/ui/#/0 pour démarrer l'interface de contrôle température (figure 3.11)

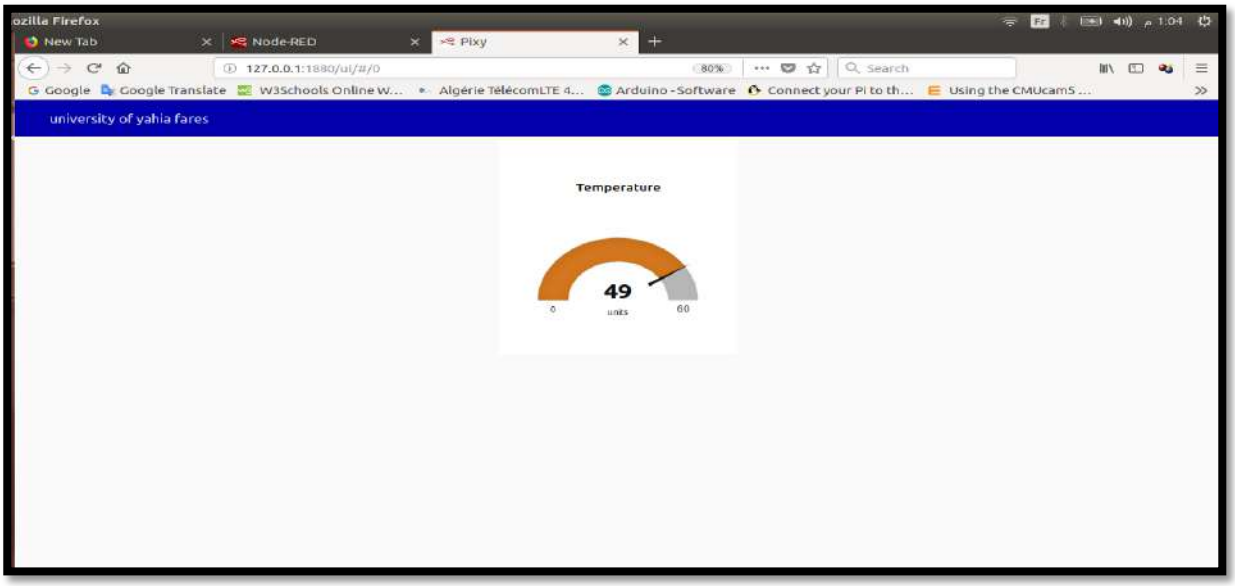


Figure 3.10 : Interface de lecture de température

10. conclusion

Les interfaces Homme / Machine sont depuis quelques décennies devenues des outils indispensables dans le contrôle et la gestion de processus industriel en général et les robots en particulier. Celle-ci procure une facilité et une aisance dans la commande de robots mobiles. Par conséquent ce chapitre a été pour nous l'occasion décrire les différentes plate-forme qui permettent la réalisation d'IHM. Cette présentation a permis de situer et de démontrer l'utilité et l'importance d'avoir eu recours à la plate-forme de conception Node-Red. En effet, l'électronique embarquée de notre robot mobile est constituée de cartes Arduino et Raspberry, les interfacées en utilisant une palte-forme de conception classique aurait nécessité de grandes connaissances en interface web et un temps de développement important. A contrario, le fait que Node-red dispose dans son cœur des nœuds dédiés à la gestion des cartes Arduino et Raspberry a facilité grandement la conception de notre IHM. Et c'est donc logiquement que dans le prochain chapitre, nous détaillerons la conception et l'implémentation du serveur et de l'IHM qui permettra de contrôler et de gérer notre robot mobile.

Chapitre4 : Mise en œuvre de l'interface de commande du robot

1. Introduction

Pour notre application, nous avons utilisé un robot mobile dont la stratégie de commande est basée sur l'asservissement visuel. En effet le suivi de cibles mobiles doit être réalisé uniquement à partir d'information visuelles fournie par la caméra Pixy. L'architecture matérielle qui compose notre robot mobile est composée d'un Raspberry et d'un ensemble Arduino /shield. Pour manipuler cette plate-forme, nous avions besoin forcement d'une interface Homme / Machine qui faciliterai l'accès aux ressources de cette plate-forme et qui permettrai de la gérer de manière aisée.

Et donc ce fut logiquement que notre choix s'est orienté vers l'utilisation du nouvel logiciel Node-Red. En effet, ce dernier s'y prête parfaitement du fait qu'il dispose dans son cœur des nœuds dédiés à la gestion des cartes Arduino et Raspberry.

Dans ce présent chapitre, nous décrirons toutes les étapes nécessaires pour la réalisation et le déploiement de l'interface web de gestion du robot mobile.

- **Objectif du chapitre**

Ce chapitre est conçu pour clarifier toutes les étapes à suivre pour mettre en œuvre une interface web qui permette de commander et de superviser notre robot.

2. Installation et configuration du Node-Red

Dans ce qui suit, nous allons installer Node.js et Node-RED sur une platform de system d'exploitation (Ubuntu 16.04 LTS).

Node-RED a une interface beaucoup plus puissante et flexible, et une grande communauté open source créant des nœuds pour interagir avec une grande variété d'applications et de services

Les conditions initiales pour implémenté l'installation est :

- System d'exploitation Windows, Linux, Unix ou un autre system (dans notre cas on a choisi l'Ubuntu 16.04 LTS c'est un type de system d'exploitation Linux)
- L'Ubuntu 16.04 est gratuitement installé -freeware-

2.1. Première méthode

2.1.1. Etape 1 – Installation de Node.js et npm

Pour avoir installé le Node-red, il est obligatoire de fonder la platform Node.js. L'installation de Node.js est très simple sur Ubuntu 16.04 LTS car il est inclus dans les dépôts par défaut. Dans un Terminal, exécutez la commande suivante : -vous devez être connecter sur internet pendant toute la période d'installation-

```
sudo apt-get install nodejs-legacy
```

Note: un terminal c'est l'invite de commande de Lunix (équivalent de MS-DOS pour windows-

Vérifiez que l'installation est réussie en vérifiant la version. Node.js vous affichera sa version. Exemple:

```
node -v
v4.2.6
```

Node.js utilise un gestionnaire de paquet appelé npm (Node Package Manager). Il permet d'installer et de gérer les packages. Comme Node-RED est une "sur-couche graphique" de Node.js, il est donc nécessaire de l'installer.

```
sudo apt-get install npm
```

Une fois l'installation terminée, vérifiez la version. Si la version est retournée sans erreur, vous pouvez poursuivre l'installation de Node-RED. Pour vérifier que l'installation a réussi, demandez à npm d'imprimer ses informations de version par :

```
npm -v
3.5.2
```

2.1.2. Etape 2 – Installer Node-RED

Node-RED s'installe à l'aide du gestionnaire de paquets npm. Nous allons en profiter pour installer le module node-red-admin qui ajoute quelques outils d'administration pour Node-RED.

```
sudo npm install -g --unsafe-perm node-red node-red-admin
```

Par défaut, npm installe les paquets dans le répertoire courant. L'option -g permet d'installer les paquets de manière globale, c'est à dire qu'ils seront disponibles pour tous les projets Node.js. L'option --unsafe-perm permet de passer outre les erreurs qui peuvent se produire durant l'installation. Par exemple lorsque certains modules doivent être compilés au moment de l'installation. Node-RED utilise le port 1880 par défaut. Il faut donc l'autoriser

```
sudo ufw allow 1880
```

L'interface d'administration principale de Node-RED va se charger come la figure suivant :

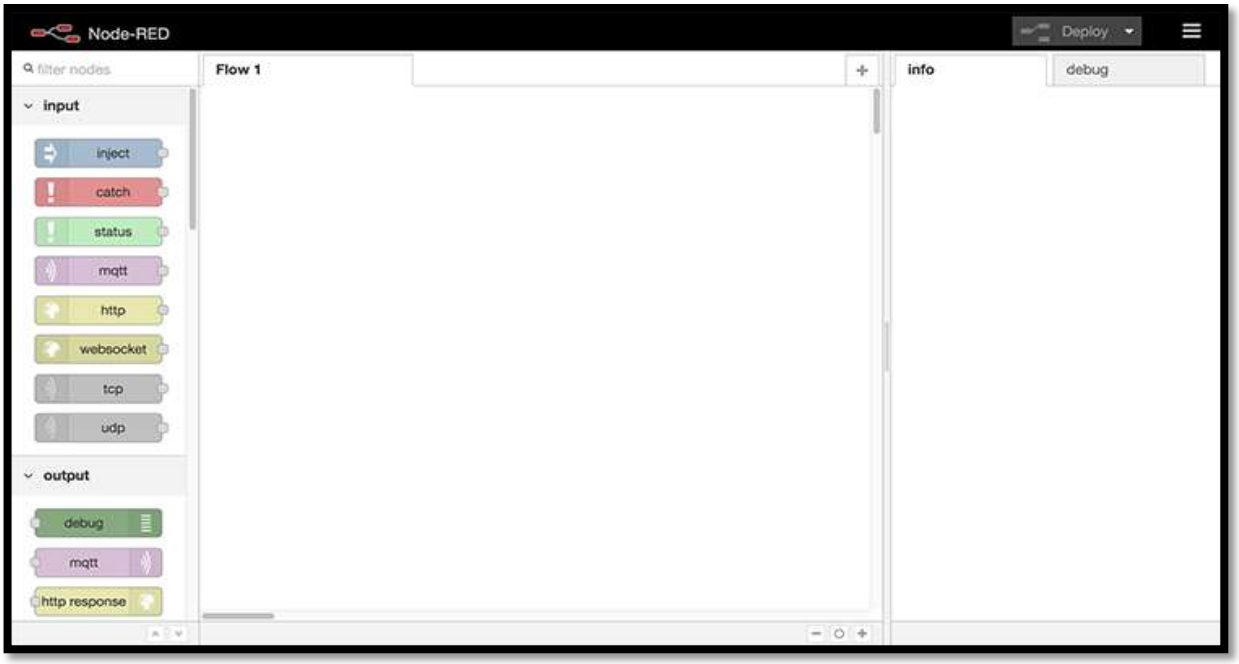


Figure 4.1 : interface de node-red

Si cela a fonctionné vous pouvez taper CRTL+C dans votre terminal pour fermer Node-RED et retourner à l'invite de commande. Vous avez installé avec succès Node-RED et vous l'avez testé, alors maintenant nous allons le configurer pour se lancer en même temps que le système.

2.1.3. Etape 3 – Faire démarrer Node-RED au démarrage du système

Afin de démarrer Node-RED automatiquement au démarrage, nous aurons besoin d'installer un fichier node-red.service au lieu du script init plus traditionnel. Pour pouvoir démarrer automatiquement Node-RED au démarrage d'Ubuntu, nous allons créer un script qui sera lancé par le system.

```
Description=Node-RED
After=syslog.target network.target
[Service]
ExecStart=/usr/local/bin/node-red-pi --max-old-space-size=128 -v
Restart=on-failure
KillSignal=SIGINT
# log output to syslog as 'node-red'
SyslogIdentifier=node-red
StandardOutput=syslog
# non-root user to run as
WorkingDirectory=/home/sammy/
User=sammy
Group=sammy
[Install]
WantedBy=multi-user.target
```

Ouvrez un fichier vide nommé node-red.service par cette commande

```
sudo nano /etc/systemd/system/node-red.service
```

Collez le contenu du script au-dessus puis remplacez la clé user par votre nom d'utilisateur. Enregistrez le fichier avec CTRL+X puis O. Maintenant que le fichier de configuration est prêt, on indique au system qu'il doit l'exécuter au démarrage.

```
sudo systemctl enable node-red
```

Une explication complète des fichiers de service systemd est hors du sujet de ce projet , mais vous pouvez en apprendre plus en lisant : Systemd Essentials: Working with Services, Units, and the Journal.

2.2. Deuxième méthode

Alternativement, nous pourrons installer et mètre a jour le node-red par une seul commande script suivant:

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
```

```
Running for user pi at /home/pi
 This can take 20-30 minutes on a Pi 1 - please wait.
  Stop Node-RED                    ✔
  Remove old version of Node-RED   ✔
  Remove old version of node.js    ✔
  Install node.js for Armv6        ✔  Node v6.11.0   Npm 3.10.10
  Clean npm cache                  ✔
  Install Node-RED core            ✔  0.16.2
  Install extra nodes              ✔
  Install serialport node          ✔
  Npm rebuild existing nodes       ✔
  Add menu shortcut                ✔
  Update systemd script            ✔
  Update update script             ✔
 Any errors will be logged to   /var/log/nodered-install.log
 All done.
  You can now start Node-RED with the command  node-red-start
  or using the icon under   Menu / Programming / Node-RED
  Then point your browser to localhost:1880 or http://{your_pi_ip-address}:1880
 Started  Thu 02 Jun 14:42:28 UTC 2018  -  Finished  Thu 02 Jun 15:00:14 UTC 2018
```

Maintenant que notre service est installé et compris, nous avons besoin de l'activer. Ceci lui permettra de s'exécuter au démarrage.

```
sudo systemctl enable node-red.service
```

3. les nœuds (nodes) utilisés pour mettre en œuvre notre interface web

- **les Noeuds de tableau de bord (Dashboard) :** Ce module fournir une ensemble des nœuds du node red pour crée un en-ligne data tableau-de-bord
- **les Noeuds de sortie(output) :** déboguer (debug) Offre une façon simple de voir des message qui sont affichés dans le volet "debug" (de débogage). Peut être configuré pour afficher le seul msg.payload ou l'objet msg en entier.
- **les noueds de fonction(fonctions) :** Fonction (function) Noeud générique pour programmation de fonction. En utilisant du JavaScript standard, le noeud peut être personnalisé pour réaliser des tâches de traitement complexe sur les messages d'entrées et générant un ou plusieurs messages en sortie.
- **les nœuds de stockage(Storage) :** fichier (file) Ecrit le contenu de msg.payload dans le fichier spécifié, ex: créer un fichier de log. Le nom du fichier peut être configuré dans le nœud. Si laissé blanc, msg.filename sera fixé par le message entrant. Le contenu est ajouté en fin de fichier par défaut, il est possible de modifier cette fonctionnalité pour écraser le fichier à chaque fois; par exemple, pour générer une page web "statique"ou un rapport
- **les Noeuds de Arduino :** l'Arduino doit être chargé par le Protocol Standard firmata, ce programme est figurer sur liste des exemples Arduino. Vous pouvez choisir une entrée analogique ou digitale. Ces entrées sont sensible au changement des capteurs. Mais, on a une limitation des valeurs sur les pins digitaux. Par contre sur l'analogues, la marge des valeurs plus vaste ce qui nous offre plus de choix[16],[21].

4. Conception de l'interface web sur PC

Cette page ressemble trois parties essentiels

4.1. Démarrage et configuration de la camera pixy

Ouvrir le node-red local sur PC, en tapant l'adresse suivant: **http://127.0.0.1:1880** sur un des navigateurs (Firefox, chromium,..etc). Après avoir figuré la page principale de l'interface node-red, commencer à insérer, configurer, liai et compiler (deploy) les boutons ce qui suit dans la figure suivants :

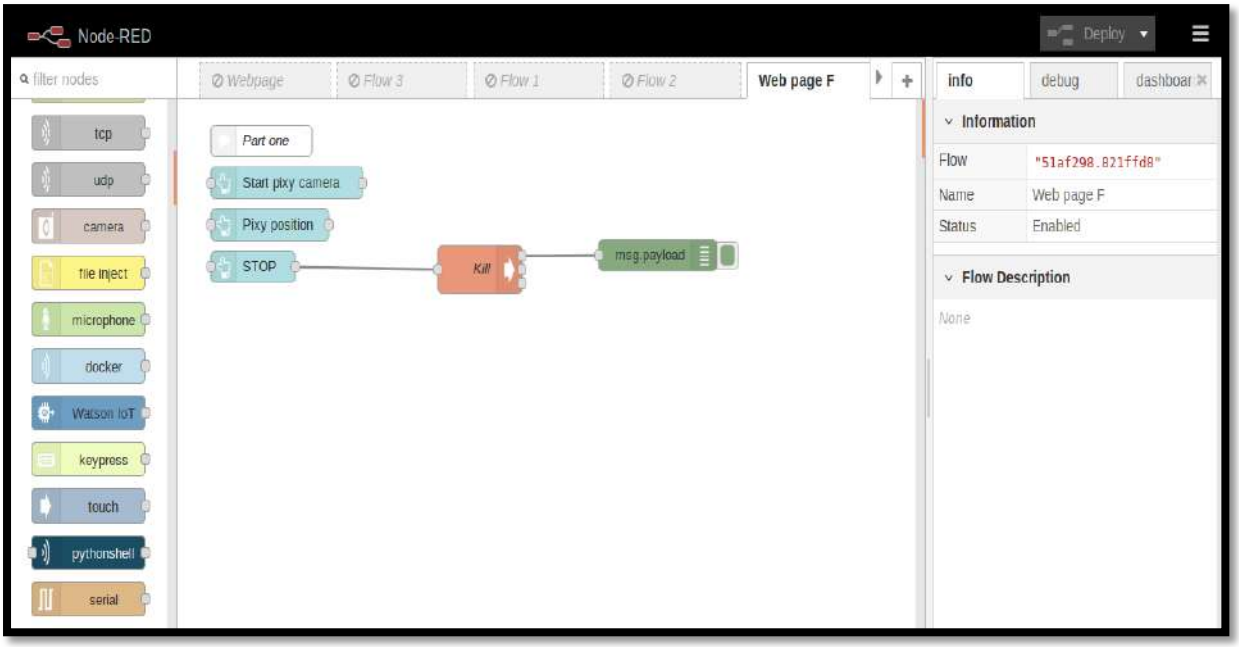


Figure 4.2 : Construction de la page web partie camera pixy

4.2. Chois, sélection et lancement des programmes pour la commande du robot

Continuer à glisser, insérer et configurer les boutons comme ceci dans le schéma suivant :

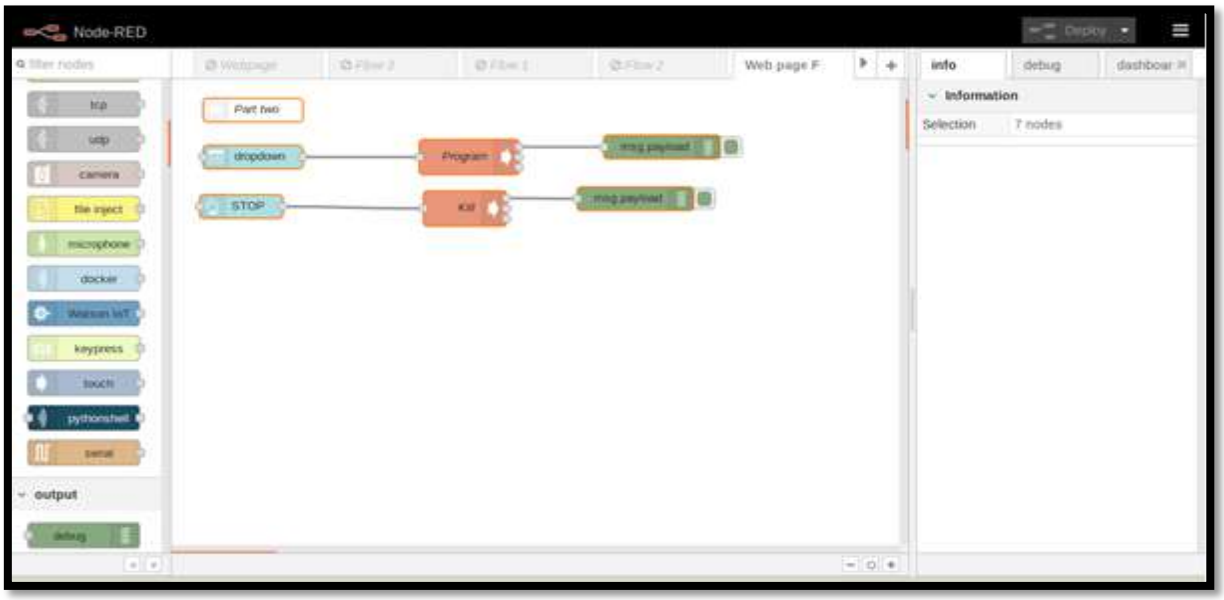


Figure 4.3 : Construction de la page web partie des programmes.

4.3. Affichage et récupération des paramètres ambients du robot

Schématiser et configurer la dernière partie comme suite :

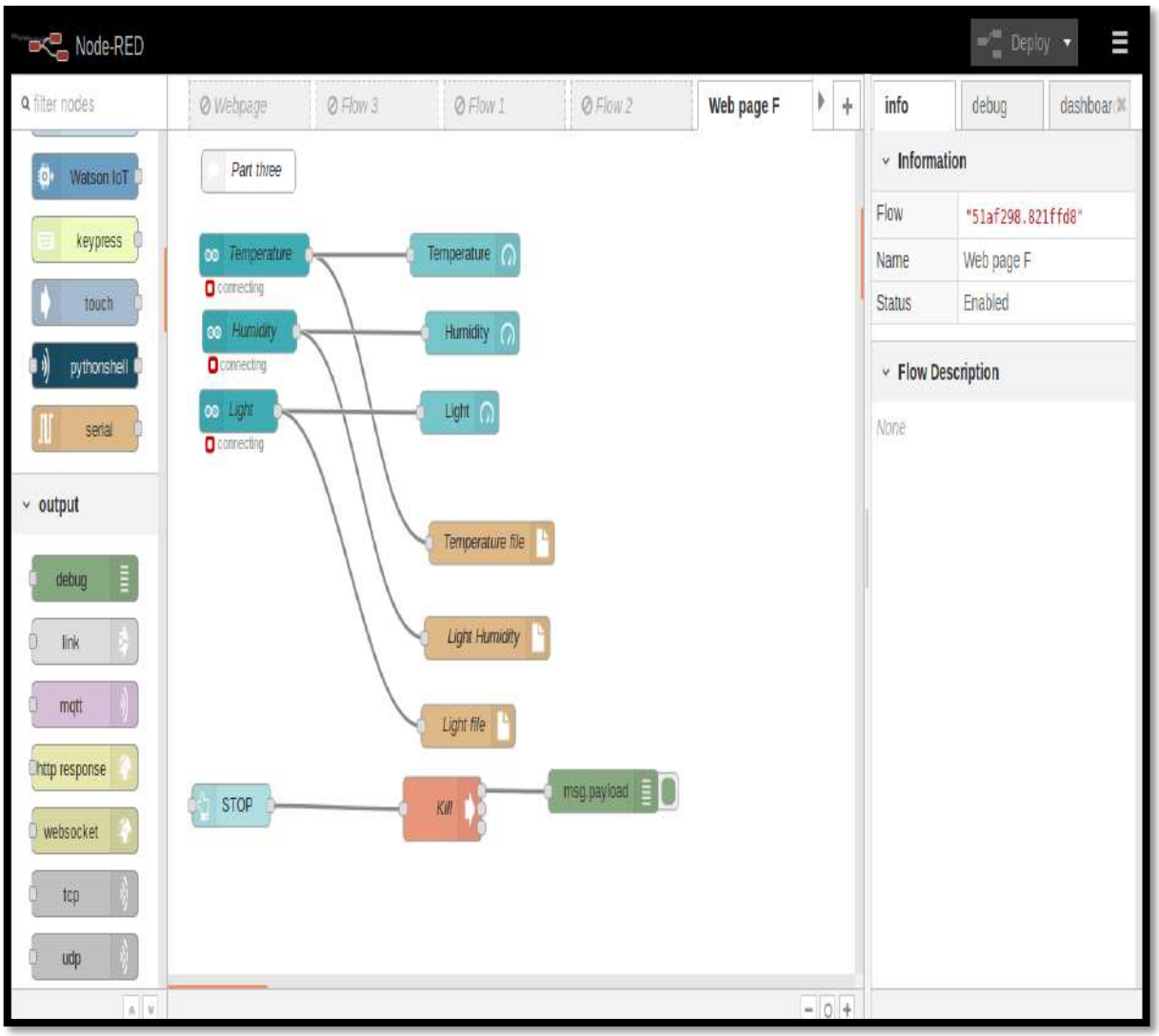


Figure 4.4 : Construction de la page web partie **des** paramètres

5. Conception de l'interface finale sur PC

A la fin on a le schéma complet suivant :

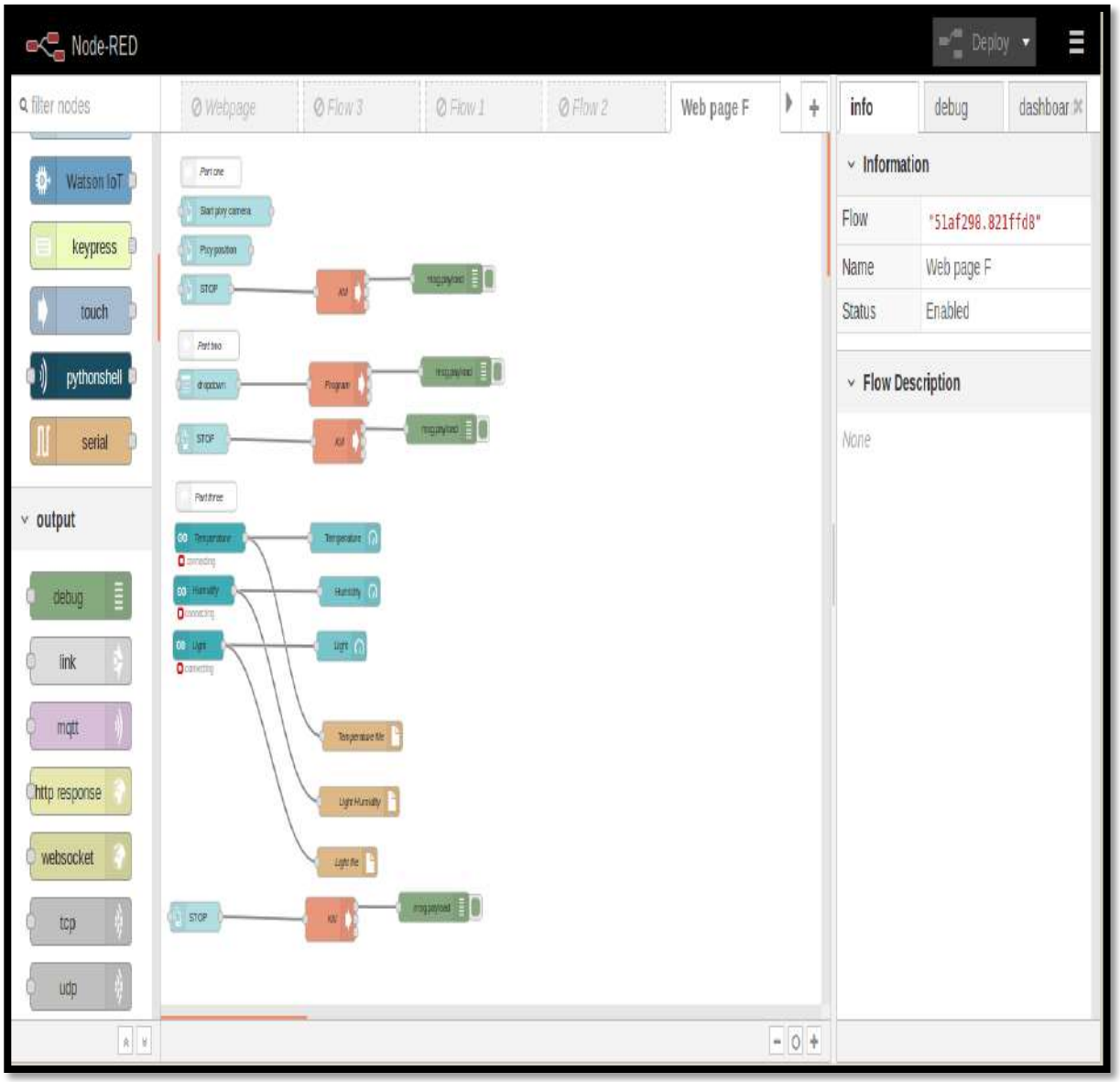


Figure 4.5: Schéma programme node-red de l'interface de command robot mobile

Le code source de la page est attaché à l'annexe A, vous pouvez l'importer et coller sur n'importe quel machine -à partir du node red- et vous aurez le même programme comme cela dans la figure précédente.

6. Lancement et test la page web sur PC

En raison de temps d'exécution, on a préféré de construire l'interface web sur PC, puis on va la transmettre sur robot. Pour lancer cette interface, simplement en tapant sur un navigateur l'adresse suivant : **http://127.0.0.1:1880/ui/#/0** cette adresse est attribué automatiquement par le node red au cours de la conception de la page web, elle contient deux parties : une partie peut être variable (**127.0.0.1**) et le reste est fixe (**:1880/ui/#/0**). Et voilà la page:

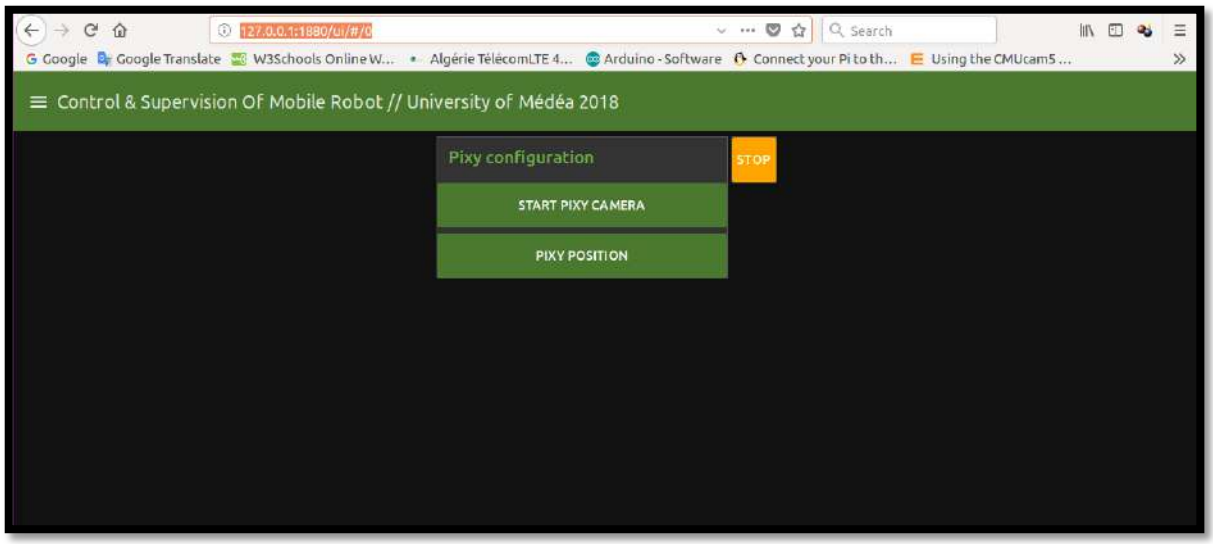


Figure 4.6: l'interface web (page camera pixy)

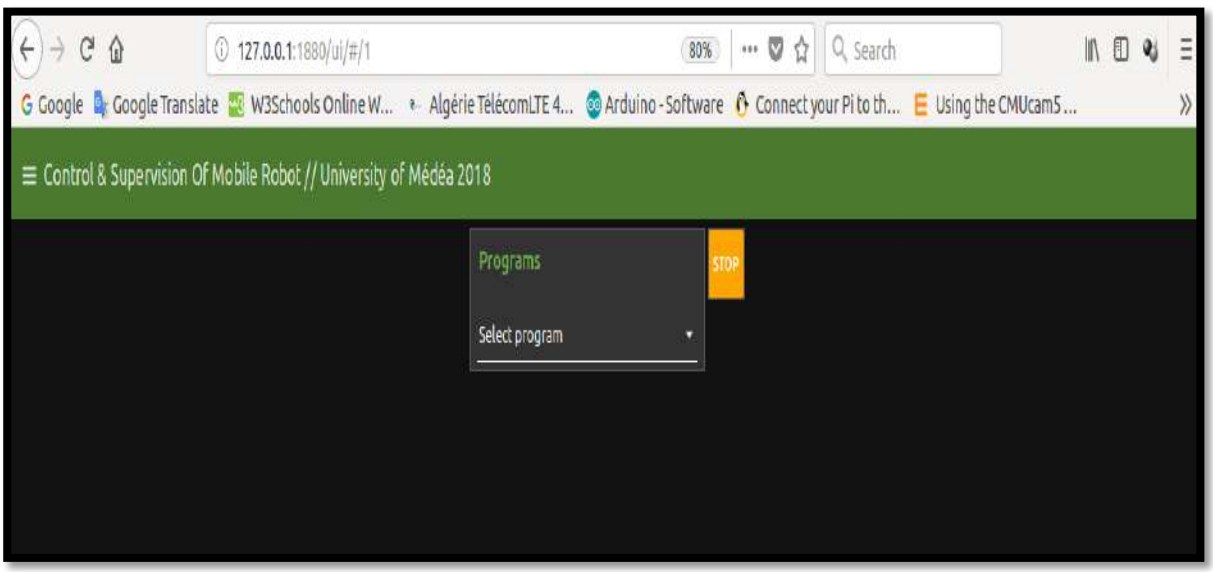


Figure 4.7: l'interface web (page programmes)

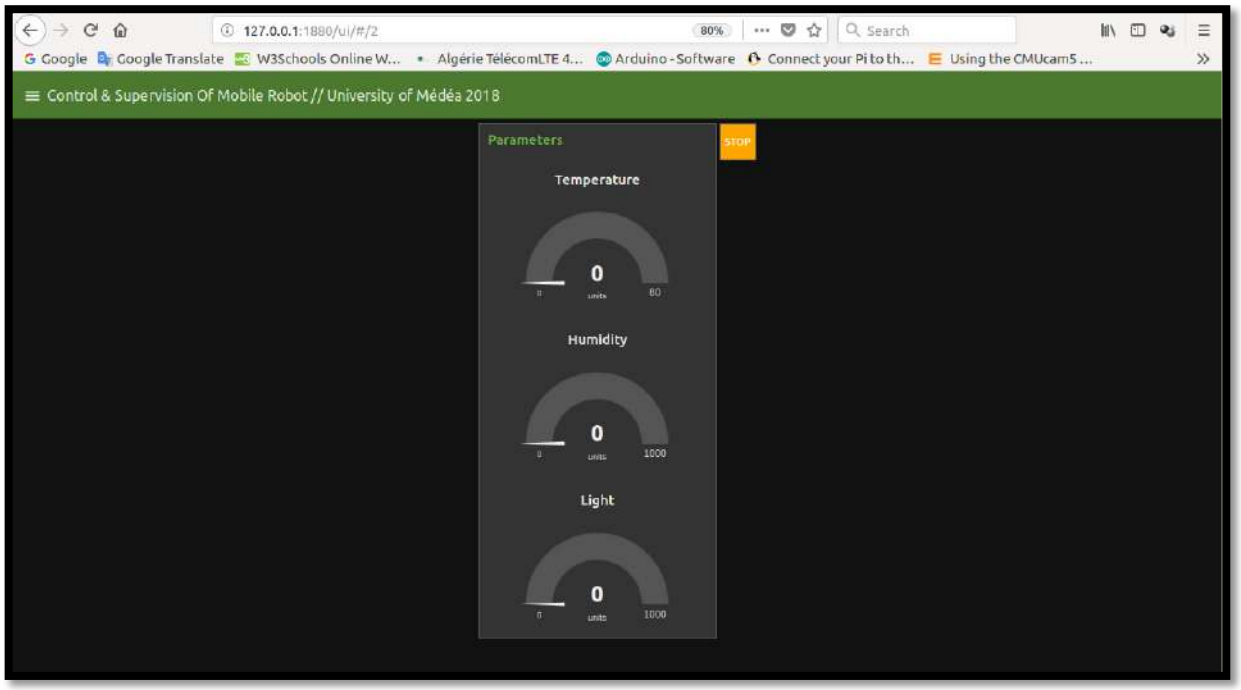


Figure 4.8: l'interface web (page paramètres)

7. Préparation et implémentation de l'interface sur robot

A. Mise à jour et configuration du node red sur robot (Raspberry)

Mètre à jour le node-red par la commande script suivant:

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
```

Après, mètre le node red exécutable au démarrage par la commande suivant :

```
sudo systemctl enable node-red
```

B. Mise en service de l'interface web sur robot (Raspberry)

Maintenant, importer le code source de l'interface web du PC et l'installer sur le robot. (Voir figure 4.9)

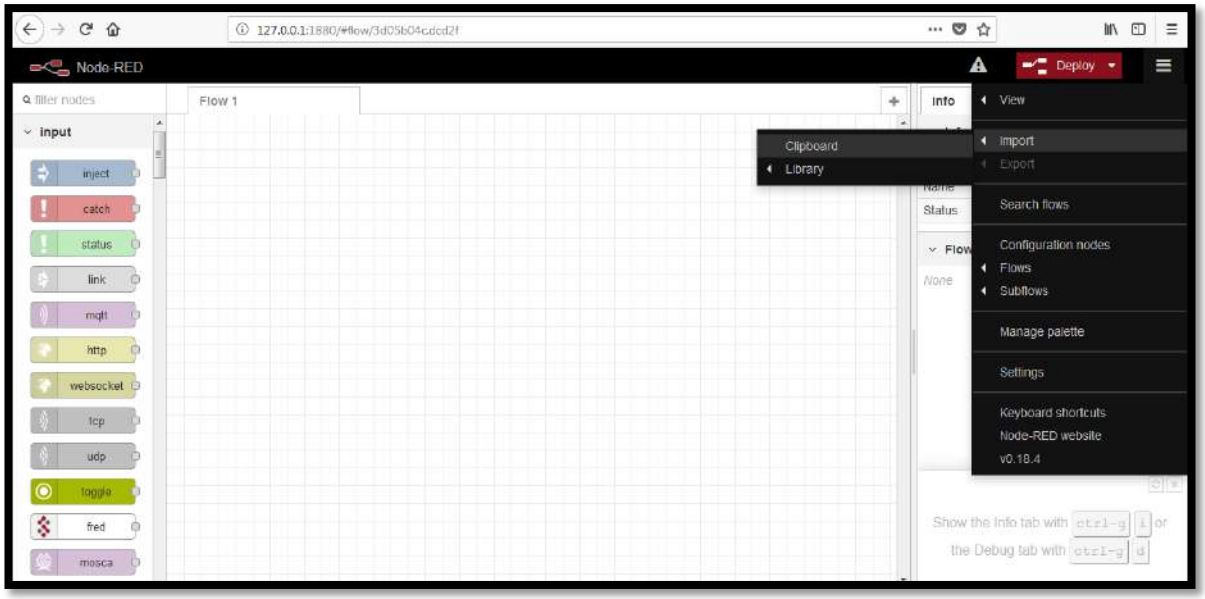


Figure 4.9: Implémentation de l'interface web sur le robot

Après avoir copié le code source de la page, coller le sur la nouvelle fenêtre puis on cliquant sur 'Import'. (Voir figure 4.10)

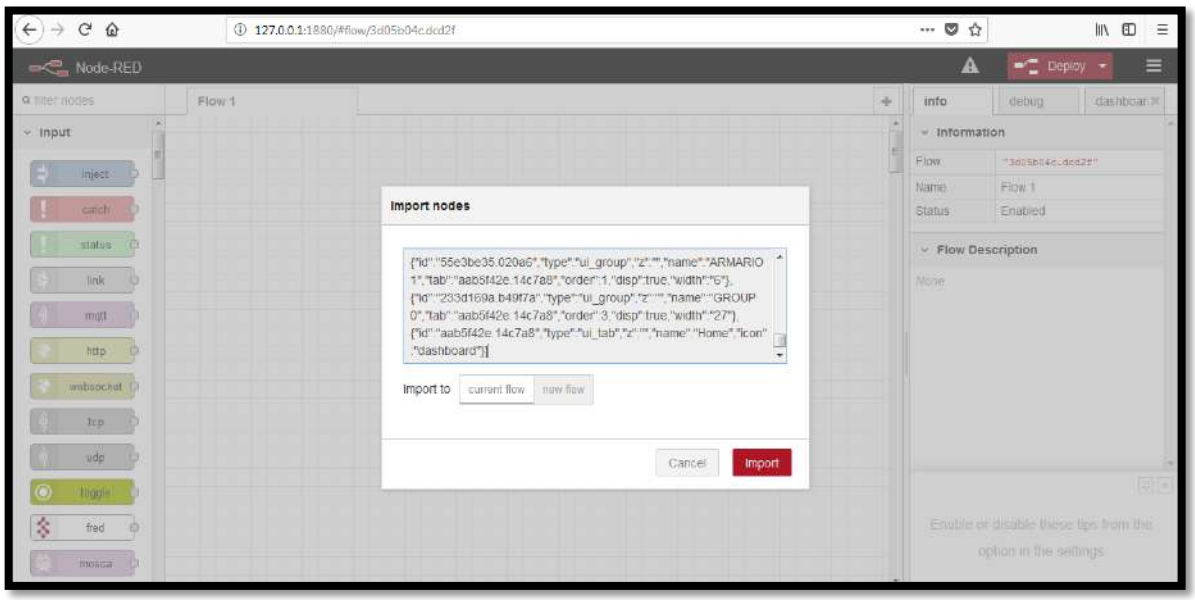


Figure 4.10 : Implémentation de l'interface web sur le robot

A la fin vous aurez le flux node-red suivant:

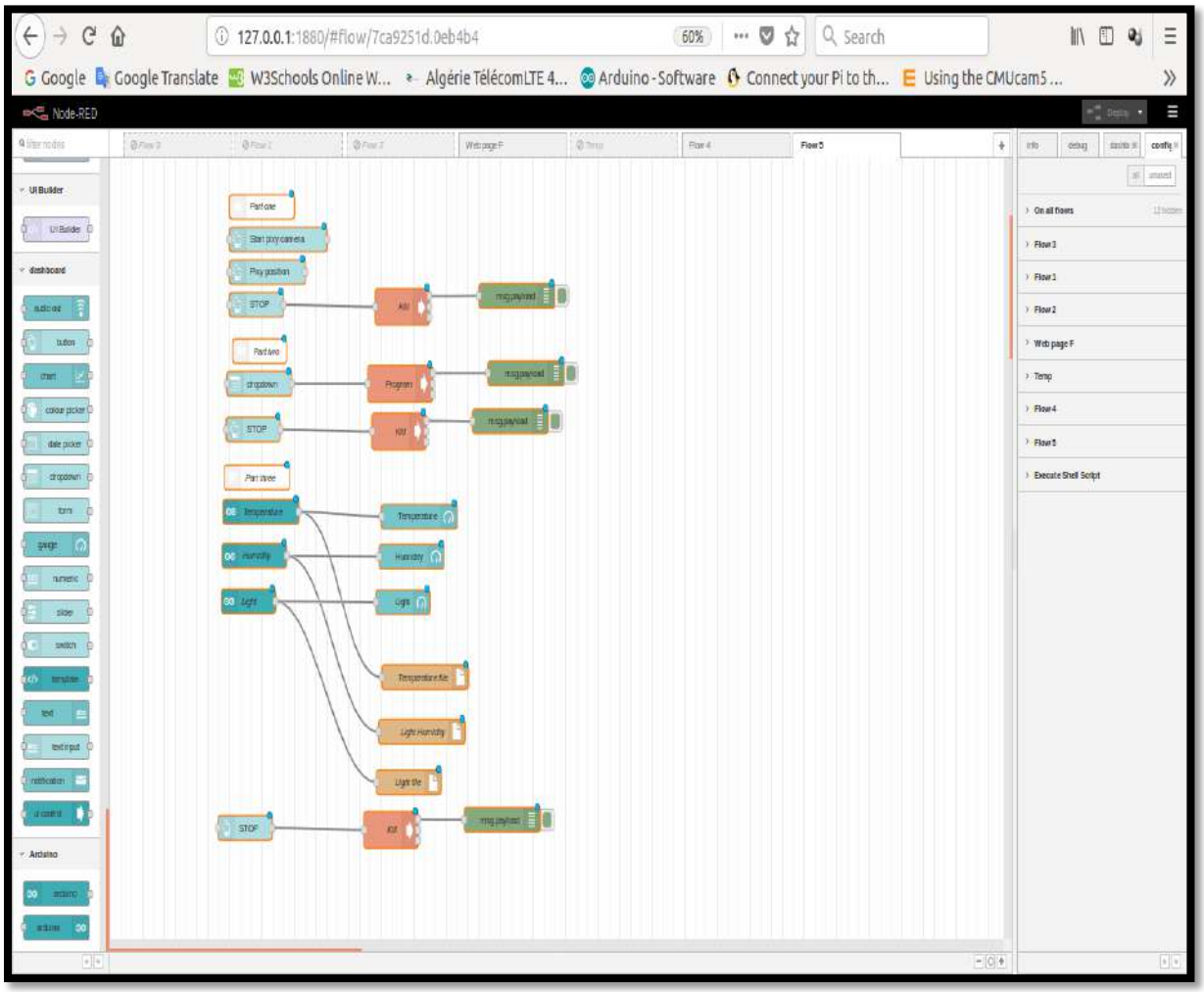


Figure 4.11 : Schéma programme node-red de l'interface après importé sur Raspberry

C. Lancement et contrôle du robot à partir de la page web

On peut lancer l'interface web localement en accédant au Raspberry puis en tapant l'adresse **http://127.0.0.1:1880/ui/#/0** (voir figure:4.12). Ou à partir d'une autre machine qui ne contient même pas le node red et sans avoir accéder le bureau du robot (Raspberry), en tapant sur n'importe quel machine -a condition doit être connecté au robot par WiFi/câble réseau - l'adresse suivant : **http://192.168.8.100:1880/ui/ #/0** . Cette adresses est subdivise en deux portions : la première partie c'est l'adresse IP du robot **(192.168.8.100)** -peut être variable selon la configuration du WiFi- et la deuxième **(:1880/ui/#/0)** est l'adresse de la page web par rapport au serveur local du node red (Figure 4.13) , (Figure 4.14).

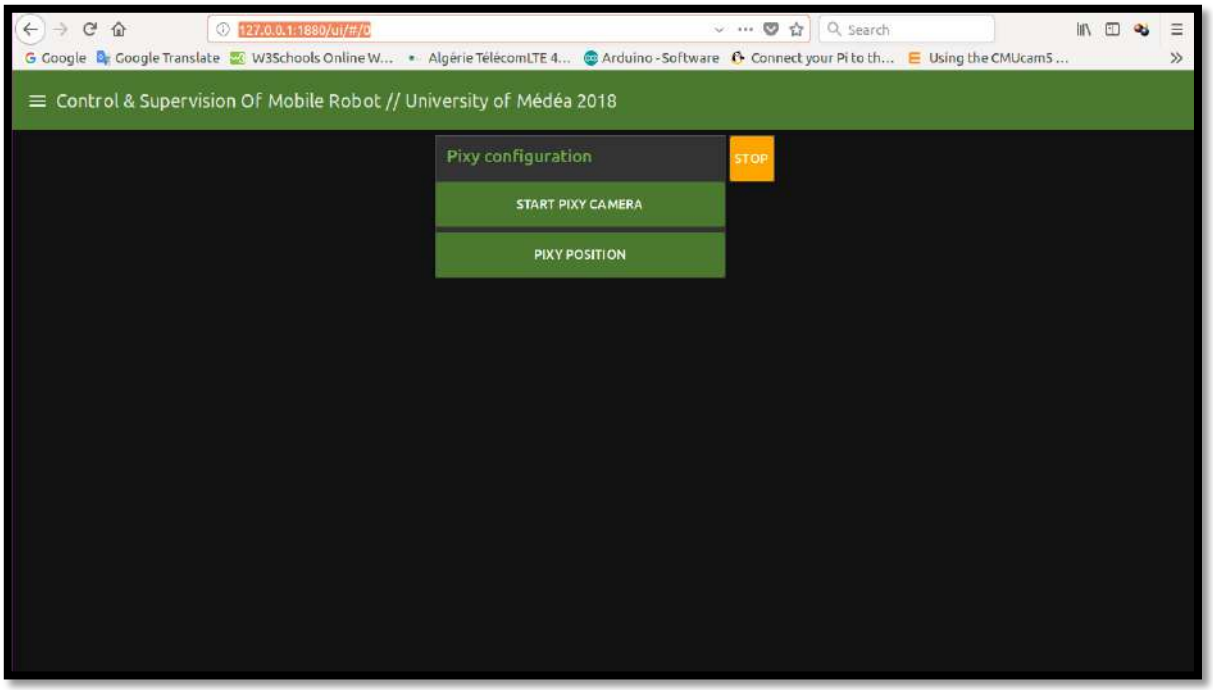


Figure 4.12 : lancement de l'interface web localement

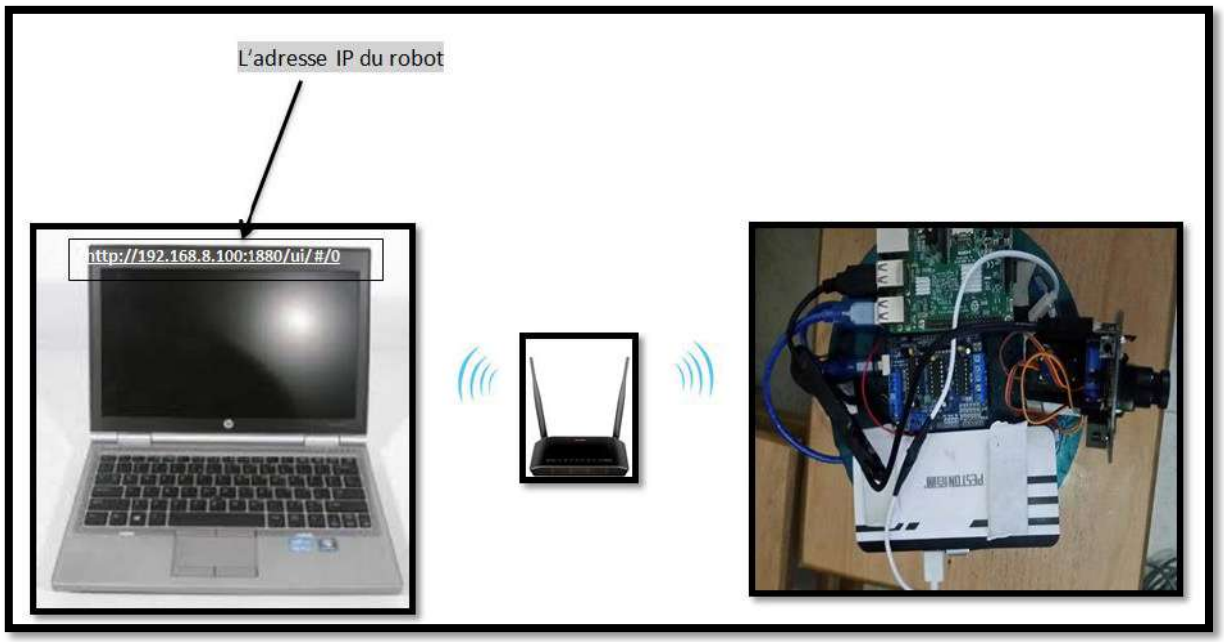


Figure 4.13 : lancement de l'interface web à distance

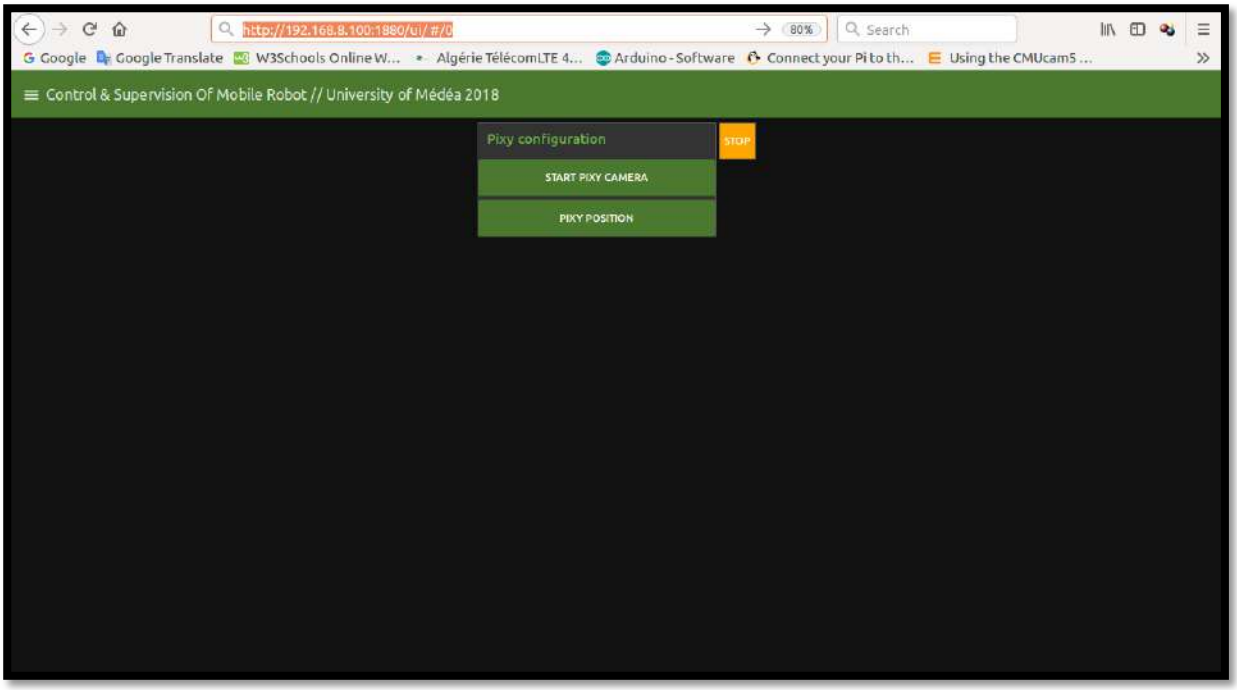


Figure 4.14 : lancement de l'interface web à distance

Apres avoir lancé la page, on peut lire en direct et enregistrer les paramètres ambiante du robot (température, humidité et la lumière) – (Figure 4.15).

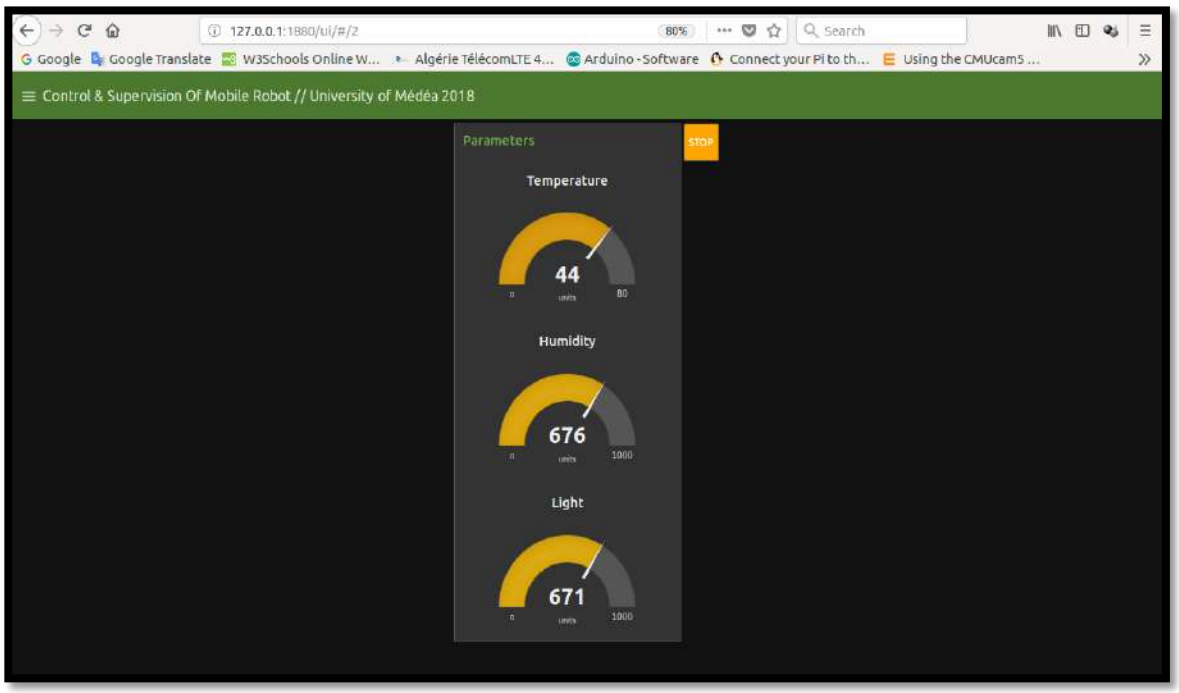


Figure 4.15 : Affichage des paramètres ambiants du robot

Il est possible aussi d'avoir une liste des programmes sur la page parmi eux vous pouvez sélectionner et exécuter. (Voir figure 4.16)



Figure 4.16: Choix de programme

En plus vous pouvez arrêter le robot à n'importe quel moment en cas de nécessité à l'aide du bouton STOP qui est figuré sur tous les pages de l'interface (Figure : 4.14).

8. Conclusion

Ce chapitre nous a permis de présenter dans un premier temps, le détail des étapes nécessaires à l'installation et au déploiement de la plate-forme Node-Red. Ces étapes ont été illustrées par des lignes de commande Linux et par des figures pour fournir un guide de référence. Par ailleurs, nous avons détaillé notre conception de l'interface Homme / Machine pour la gestion et la supervision de robot mobile. Ces phases de mise en œuvre ont été décrites et illustrées par des figures qui ont montré comment manipuler les différents nœuds présents dans le logiciel Node-Red.

Enfin, nous avons démontré la portabilité de notre interface en la transférant rapidement sur la carte Raspberry. Cette étape a été réalisée simplement en important sur la carte Raspberry le code source de l'interface conçue et validée sur PC. Cette méthode de conception permettra l'évolution de cette IHM sans perturber le fonctionnement du robot mobile. Il suffit de rajouter des capteurs ou des fonctionnalités au robot, les interfacés avec Node-Red, tester cette nouvelle version puis l'implémenter rapidement sur le robot mobile. D'ailleurs, le bouton d'arrêt d'urgence a été conçu, tester et valider sur un PC avant d'être implémenter sur le robot. La conception de cette IHM permettra dans le futur de gérer ce robot depuis n'importe quelle machine (PC, Tablette, Smartphone, …) connectée au réseau wifi du robot et par extension depuis n'importe quel endroit doté d'une connexion internet.

conclusion générale

Conclusion générale et perspectives

Au conclusion de ce travail, on a généré une interface web à trois pages à l'aide du software Node Red. Ces pages sont capables de démarrer, contrôler et arrêter le robot mobile. L'utilisation de cette interface n'est pas exclusivement seulement pour le robot utiliser dans cette application, mais tout robot mobile a carte Raspberry peut être configuré et manipuler a travers ces pages web.

Ce qui on peut faire à travers cet interface :

- ✓ Il est possible de connecter et utiliser l'interface web à un travers un câble réseau ou Wifi, en tapant seulement l'adresse IP de la page web.
- ✓ La page est exécutable sur une large liste des navigateurs (fire fox, google chrome, chromium …etc.)
- ✓ Cette interface est accessible à partir de n'importe qu'elle machine à microprocesseur (PC, Smartphone, Tablet...etc.)
- ✓ Il est facile de sélectionner sur une liste des programmes à partir de la page web pour lancer directement le robot.
- ✓ Aussi, il est faisable d'arrêter le robot dans n'importe quel moment à travers un bouton qui est tout le temps figuré sur la page web.
- ✓ Encore, on peut lire et afficher sur la page les valeurs de température, humidité et lumière ambient du robot à travers les capteurs placés sur le robot.
- ✓ En plus il est possible d'enregistrer ces données des capteurs dans un fichier puis les exploiter.

À la fin de ce travail, en recommande au future étudiants et /ou chercheurs de prendre les points suivants :

- Continuer le développement de cette interface web pour être capable de lancer et configurer la camera pixy à partir de cette bord
- Mettre ces pages web accessibles à travers le réseau public d'internet
- Sécurisé cette interface par un mot de passe
- Pour aller à une automatisation réel de ce robot, il est recommandé de mettre à niveau la camera pixy pour devient une œil artificiel.
- La création et le développent d'un langage gestuel pour surmonter les limitations d'une interface web. l'implémentation de ce langage et l'utilisation de l'œil artificiel a la place d'une camera pixy peut rendre ce robot plus libre et autonome.

Bibliographie

Bibliographie

[1] BENAISSA el mahdi, BENYAIHIA Dalal, " *Asservissement visuel d'un robot mobile"*, Université Yahia Fares de Médéa, année 2016
[2] ZAIBAK Serhane, BOUAISSA Mouadh, " *Asservissement visuel d'un robot mobile avec évitement d'obstacle*", Projet de fin d'étude, Université Yahia Fares de Médéa, année 2017

[3] BOUKLI HACENE Lotfi Fazil, *" Commande et supervision d'un ensemble de robots via internet",* thèse de magister*,* université des sciences et de la technologie d'Oran – Mohamed Boudiaf, 26 Avril 2012

[4] BELKACEM ANES ,DIDOUH MOHAMED, " Commande à distance d'un robot mobile via un smartphone Android", Projet de fin d'étude, Université Aboubakr Belkaïd –Tlemcen – , 12/06/2017

[4] Alain COURCELLE, "*Localisation d'un robot mobile : Application à l'aide à la mobilité des personnes handicapées moteur*", Thèse de doctorat, l'Université de Metz, France, janvier 2000.

[5] BOULKROUNE Ramzi, *Les systèmes embarqués*, Université de Annaba-Ingénieur d'état en électronique option télécommunication 2009.
[6] COLLECTIF, Saidi Hamza, Koudri Filali Hamza, A. K. Kheddaoui Med Amine, " *Etude Et Réalisation D'un Robot Mobile Commandé a Distance* ", Editions universitaires europeennes EUE, 2012

[7] Oualid Djekoune. " *Localisation Et Guidage Du Robot Mobile Atrv2 Dans Un Environnement Naturel Robotique*", Thèse doctorat, Université des Sciences et de la Technologie Houari Boumediene, 23 Feb 2012

Webographie

[8] CMUcam5 pixy , "*documentation globale sur CMUcam5 pixy*",
https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:start
[9] Arduino, "*Introduction*",
https://www.arduino.cc/en/Guide/HomePage

[10] locoduino, " *Alimentation d'Arduino* ",
http://www.locoduino.org/spip.php?article16

[11] Raspberry pi, "*Raspberry pi 2 model B,* raspberry pi foundation UK registred charity 112909
https://www.raspberrypi.org/products/raspberry-pi-2-model-b/

[12] RASPBERRY PI 3 (MODÈLE B) *" Raspberry Pi Les autres produits de Raspberry Pi"*
https://www.elektor.fr/raspberry-pi-3-model-b
[13] Arduino, "*Arduino Board Uno*",
https://www.arduino.cc/en/main/arduinoBoardUno

[14] Arduino, "*Arduino IDE* ",
https://github.com/arduino/arduino/wiki/arduino-ide-1.5-3rd-party-hardware-specification

[15] CMUcam5 Pixy, Open Source Programmable Embedded Color Vision Sensors,
"*Aperçu sur CMUcam5 Pixy* ",
http://www.cmucam.org/projects/cmucam5/wiki/Installing_PixyMon_on_Linux?version=8

[16] Node-Red ,"*Node-Red documentation*",
http:// https://nodered.org/docs/hardware/arduino.htmlwire Serial Interface
[17] Rasp-Node-Red "*MCHobby – Wiki*",
https://wiki.mchobby.be/index.php?title=Rasp-Node-Red-Install-Node.html

[18] Accueil/Tech/Définitions, '*Lettres d'information'*
https://www.futura-sciences.com/tech/definitions/internet-java-485/.html

[19] langage de php , "*documentation globale php*",
http://www.jobintree.com/dictionnaire/definition-php-309.html

[20] Programmation Python/Introduction, "*documentation sur python*",
https://fr.wikibooks.org/wiki/Programmation_Python/Introduction.html

[21] Pourquoi Node-red, " *Programmation puis Node -Red.* ",
https://blog.octo.com/node-for -api

[22] RASPBERRY PI 3 (MODÈLE B) *" Raspberry Pi Les autres produits de Raspberry Pi"*
https://www.elektor.fr/raspberry-pi-3-model-b