REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
--T L E M C E N--

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
-تلمسان-

Mémoire de fin d'étude

Pour l'obtention du diplôme d'Ingénieur

Filière : Automatique
Spécialité : Automatique

Présenté par : BERREZOUG Lotfi
BENYAGOUB Ali

Thème

Conception et Réalisation d'un Prototype d'un Véhicule Autonome

Soutenu publiquement, le 08 /07 /2021 , devant le jury composé de :

M. BRAHAMI Mostapha Anwar	MCB	ESSA. Tlemcen	Président
M. ABDELLAOUI Ghouti	MCB	ESSA. Tlemcen	Directeur de mémoire
M. ABDI Sidi Mohammed	MCB	ESSA. Tlemcen	Examinateur 1
Mme. NEDJAR Imene	MCB	ESSA. Tlemcen	Examinateur 2

Année universitaire : 2020 /2021


بسم الله الرحمن الرحيم

Remerciements

Nous tenons premier lieu remercier Dieu tout puisant de nous avoir accordé la force et le courage de mener ce travail à terme.

Nous tenons à adresser nos sincères remerciements à notre encadreur de thème le Dr.ABDELLAOUI Ghouti, Enseignant à l'école supérieure en sciences appliquées Tlemcen pour sa disponibilité, pour sa lecture , suggestion et remarques et surtout pour sa confiance sans limite mise en mois tout au long de ce projet de recherche.

Également du président de jury **BRAHAMI Mostapha Anwar** pour l'intérêt qu'il a porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par ses propositions.

Encore, de remercier respectueusement les membres de mon jury d'examen **Monsieur ABDI Sidi Mohammed** et **Madame NEDJAR Imene**, qui ont accepté d'examiner cet humble travail que j'espère à la hauteur de mes ambitions bien modestes.

Nous remercions particulièrement nos famille pour leur soutien moral tout au long de ce travail, merci de nous avoir encouragé, et cru en nous.

Nous remercions également nos amis nos collègues de travail, qui nous ont encouragé tout au long de ce projet et nous ont beaucoup aidé.

ملخص

سيكون مجال دراسة هذه الأطروحة هو تصميم وحدة تنقل ذكية. أولاً ، تحدثنا عن الروبوتات والسيارات المستقلة ومجالات تطبيقها. بعد ذلك ، اقترحنا تحقيق سيارة مستقلة تبحث عن هدف محدد مسبقًا وتحاول السير نحوه من خلال تجنب العقبات، ثم اخترنا المواد اللازمة لضمان حسن سير العمل ، مع تفصيل الإجراءات والنهج المستخدم للوصول إلى هدف المشروع.

الكلمات المفتاحية : Raspberry pi , Pi caméra, Arduino, Ultrason, Moteur CC, Télécommande IR, Servomoteur, Python, IDE Arduino .

Résumé

Le champs d'étude de ce mémoire portera sur la conception d'un module de navigation intelligent. Tout d'abord nous avons parlé sur la robotique, les voitures autonomes et leurs domaines d'applications. Ensuite, nous avons proposé de réaliser une voiture autonome qui cherche une cible prédéfinie et essayé de naviguer vers se dernier en évitant les obstacles, par la suite nous avons choisit le matériel nécessaire pour assurer le bon fonctionnement, en détaillant la procédure et la démarche utilisée pour atteindre l'objectif du projet.

Mots clé : Raspberry pi, Pi caméra, Arduino, Ultrason, Moteur CC, Télécommande IR, Servomoteur, Python, IDE arduino.

Abstract

The field of study of this thesis will be the design of an intelligent navigation module. Firstly, we talked about robotics, autonomous cars and their application areas. Then, we proposed to realize an autonomous car that looks for a predefined target and tries to navigate towards it by avoiding obstacles, then we chose the necessary material to ensure the good functioning, detailing the procedure and the approach used to reach the project objective.

Key words : Raspberry pi, Pi caméra, Arduino, Ultrason, Moteur CC, Télécommande IR, Servomoteur, Python, IDE arduino.

Table des matières

Table des figures

Introduction générale

La robotique est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots.

De cette définition découlent deux interprétations : la première serait de voir le robot comme une machine, qui possède des capteurs, un système logique et des actionneurs. Ceci est matériel. La deuxième laisse penser qu'un robot peut aussi être virtuel.

La robotique actuelle trouve des applications dans différents domaines:

- La robotique industrielle.
- La robotique domestique.
- La robotique médicale.
- La robotique militaire.

La robotique scientifique, par exemple pour l'exploration de l'espace (aérobot), des fonds marins (robots sous-marins autonomes), dans les laboratoires d'analyse (robotique de laboratoire), etc.

La robotique de transport (de personnes et de marchandises), avec par exemple ROPITS (Robot for Personal Intellingent Transport System)5, Robosoft6, RoboCourier7, etc.

Un robot est un système alimenté en énergie qui évolue dans un environnement statique ou dynamique, il est formé d'un microcontrôleur ainsi que d'un ou plusieurs capteurs et actionneurs.

La conception d'un robot se base sur son cahier de charges. Elle comprend l'analyse du comportement souhaité pour le robot et sa synthèse théorique, à l'aide notamment des théories d'asservissements, ainsi que l'implémentation logicielle et matérielle du robot.

La structure d'un robot est contrôlée de manière à effectuer une ou un ensemble de tâches. Ce contrôle inclut trois phases distinctes qui se répètent en boucle : la perception, le traitement et l'action. Un robot fonctionne par l'exécution continue d'un programme informatique constitués d'algorithmes. Ce programme est écrit dans un langage de programmation dont la nature est choisie par le constructeur.

Un véhicule autonome, véhicule automatisé, véhicule à délégation de conduite ou véhicule entièrement automatisé est un véhicule automobile capable de rouler — sur route ouverte — sans intervention d'un conducteur. Le concept désigne un véhicule pouvant circuler sur la voie publique dans les situations prévues de trafic sans intervention humaine. C'est une application typique du domaine de la robotique mobile dans laquelle de nombreux acteurs sont engagés.

Des éléments de solutions techniques, légales, psychologiques et juridiques ont déjà été introduits, mais certaines questions restent non résolues.

La notion de voiture autonome couvre, selon le contexte, un véhicule totalement autonome (niveau 5) ou bien un véhicule « semi-autonome » disposant de différents systèmes d'aide à la conduite ou de conduite semi-automatisée supervisée par le conducteur (niveau 2+), ou bien encore un véhicule à délégation de conduite (niveau 3). Bien que la publicité pour certains véhicules de niveau 2+ évoque « pilotes automatiques » et « conduite entièrement autonome », ces véhicules ne sont considérés par le code pénal français ni comme automatisés, ni comme autonomes, ni comme à délégation de conduite s'ils ne répondent pas à des critères prévus pour le niveau 3.

Certains systèmes de conduite automatisée de niveau 3 sont prévus pour fonctionner dans des conditions spécifiques, par exemple dans les embouteillages sur autoroute ou lors du stationnement automatique.

Notre projet consiste à concevoir un module intelligent de navigation, qui cherche a détecter un objectif (forme géométrique particulière) et a se diriger vers ce dernier en navigant dans un

espace ouvert, en évitant les obstacles.

Pour cela nous avons besoin des microcontrôleurs comme arduino, raspberry, et des capteurs pour faire la détection d'obstacle.

Pour coder il y a beaucoup de logiciel de programmation, on doit choisir les meilleurs et les plus efficaces a utilisé pour notre projet .

Notre mémoire est organisé comme suit :

- Le premier chapitre présente des généralités sur les robots mobiles.
- Le deuxième chapitre est devisée en deux parties :
  - La première concerne le traitement d'image numérique : historique, définitions et applications.
  - La deuxième traite les différentes algorithmes, et faire une petite simulation.
- Le troisième chapitre décrit la réalisation pratique : la partie pratique mécanique, la partie électronique et les programmes développés.
- En fin, nous finissons notre mémoire par une conclusion qui présente le bilan de ce travail.

Chapitre 1

Généralité sur la robotique

Introduction

Dans ce chapitre nous parlons de l'histoire du développement de la robotique et ses différents domaines d'utilisation et l'apparition de l'automobile.

Présenté par certains comme la solution miracle pour la mobilité de demain, le véhicule autonome est porteur de toutes les promesses, notamment celle d'être moins polluant. C'est d'ailleurs le projet mené par l'exécutif français dans le cadre de sa stratégie nationale dévoilée en fin d'année.

1.1 Définition

1.1.1 La robotique

La robotique est l'ensemble des techniques pour automatisée une machine où un robot grâce à un système de commande intelligent a base d'un microprocesseur qui a le rôle de faire tous les calculs et la décision des opérations pour commander les actionneurs (moteur, servo moteur, vérin, vanne...) [1]

1.1.2 Le robot

Un robot est un automate doté de capteurs et d'effecteurs lui donnant une capacité d'adaptation et de déplacement proche de l'autonomie. Un robot est un agent physique réalisant des tâches dans l'environnement dans lequel il évolue [1]. Certain systèmes, exigent un travail collaboratif entre ses agents (robots) et qui doivent configurable dynamiquement selon les différents changement de l'environnement extérieur [2] [3].

1.1.3 Automates

Un automate est un dispositif se comportant de manière automatique, c'est-à-dire sans l'intervention d'un humain. [1]

1.2 Historique

Dei ex Machinis, une encyclopédie très complète en trois volumes décrivant les vies et les œuvres des facteurs d'automates de l'Antiquité jusqu'au début de l'Intelligence artificielle (IA), a été publiée en 2015 par Jean-Arcady Meyer ; elle est encore disponible aux éditions du Net.

Dès l'Antiquité, on signale plusieurs automates bio-inspirés, dont le pigeon volant d'Archytas de Tarente ou les fameuses scènes théâtrales animées de Héron d'Alexandrie.

Au XVIe siècle, Léonard de Vinci, inspiré par les anatomies interdites, aurait construit le premier androïde capable de coordonner les mouvements de ses bras, de ses jambes et même de ses mâchoires.

Au XVIIIe siècle (considéré comme l'âge d'or des automates), le célèbre canard de Jacques de Vaucanson, aujourd'hui perdu, qui pouvait boire, se nourrir, caqueter, s'ébrouer dans l'eau, digérer sa nourriture et même... déféquer, a ébloui par sa complexité les spectateurs de l'époque.

À la même période, les horlogers Jaquet-Droz inventèrent une musicienne, un écrivain et un dessinateur réalisant vraiment les mouvements correspondant à la pratique de leur art.

Au XIXe siècle, l'automate parlant Euphonia, d'Eugène Faber, était supposé dialoguer avec les spectateurs et l'automate turc du baron von Kempelen jouait aux échecs – actionné peut-être par un humain caché dans le dispositif.

L'apparition des premiers robots est qu'au tout début du XXe 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 (1950), le renard électronique de Ducrocq (1953) ou l'homéostat d'Ashby (1952) étaient, eux, dotés 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, ces robots ont des organes sensoriels – les capteurs – recueillant des informations de l'environnement qui vont, elles, influencer l'activité de leurs organes moteurs – les actionneurs. [1]

Histoire du robot industriel

- 1920 : Apparition du mot robot L'origine du mot robot provient de la langue tchèque dans laquelle sont ancêtre "robot" signifie travail forcé. Il a été introduit, en 1920, par l'écrivain tchèque Karel Capek dans la pièce de théâtre Rossâmes Universel Robots

- 1961 : Unimation, le 1er robot industriel Descendant direct des télémanipulateurs développés pour les besoins du nucléaire. Il est vendu à partir de 1961 par la société américaine Unimation (devenu Stäubli Unimation), créée par George Devol et Joseph Engelberger. Il est utilisé pour la première fois sur les lignes d'assemblage de General Motors. Ce robot, grâce à son bras articulé de 1,5 tonnes, était capable de manipuler des pièces de fonderie pesant 150 kg.

- 1972 : Première chaîne de production robotisée Nissan ouvre la première chaîne de production complètement robotisée, Selon une étude de l'IFR, 2142 millions de robots ont été fabriqués entre les années 60 et la fin 2010, Les analystes estiment qu'aujourd'hui, de 1 à 1,3 million de robots travaillent pour nous dans les usines dans le monde.

Les différents types des robots :

- Robots mobiles : Robots capables de se déplacer dans un environnement. Ils sont équipés ou non de manipulateurs suivant leur utilisation.

- Robots domestiques : Robots utilisés pour des tâches ménagères, par exemple en vaisselle, en repassage, en nettoyage.

- Robots collaboratifs : Hommes et robots travaillent ensemble, les robots permettant de diminuer la pénibilité des manipulations manuelles, des efforts ou des mouvements réalisés par l'opérateur.

1.3 Domaine d'application

1.3.1 L'armée

L'apparition de robots destinés à la guerre date de la Seconde Guerre mondiale, avec le Goliath (Figure 1.1), une mine filoguidée pouvant être actionnée à distance. Apres l'apparition des navires de guerre, les appareils volants sans pilotes.[1]

Figure 1.1: Goliath une mine commandé

1.3.2 L'industrie

La robotisation de l'industrie commence dans les années 1960, dans le secteur automobile, puis va se répandre jusqu'à ce que l'on connaît aujourd'hui sous la forme des bras automatisé ou une chaine de production (ensachage de engrais dans des sac).(Figure 1.2)

Figure 1.2: Robot industrielle de type bras manipulatoire

1.3.3 Transport

À la fin du 20e siècle, la robotique de transport (de personnes) fait son apparition avec le métro de Lille Métropole, qui est le premier métro au monde à utiliser la technologie de véhicule automatique léger (VAL), ou la ligne 14 du métro parisien, seule ligne du réseau métropolitain de Paris exploitée de manière complètement automatique dès sa mise en service (15 octobre 1998).(Figure 1.3)

Figure 1.3: Métro de Lille

1.4 L'automobile

Le terme automobile (dérivé de voiture automobile) est un véhicule à roues mû par un moteur et destiné au transport terrestre de personnes et de bien. L'abréviation populaire « voiture » est assez courante, bien que ce terme désigne de nombreux types de véhicules qui ne sont pas tous motorisés.

1.4.1 Naissance de l'automobile

L'histoire de l'automobile rend compte de la naissance et de l'évolution de l'automobile, invention technologique majeure qui a considérablement modifié les sociétés de nombreux pays au cours du xxe siècle. Elle prend naissance au xixe siècle durant la Révolution industrielle.

Le premier véhicule automobile fonctionnel a été inventé en 1769 par Nicolas Joseph Cugnot sous le nom de fardier de Cugnot. Puis une voiture a 3 roues équipé d'un moteur à vapeur destiné à l'armée française et Le premier moteur à essence est inventé par Daimler et Maybach en 1885.

1.4.2 La voiture autonome

Définition

Une voiture dite autonome lorsqu'elle est équipée d'un système de pilotageautomatique qui lui permet de circuler sans intervention humaine dans des conditions de circulation réelles.

Un véhicule autonome, véhicule automatisé, véhicule à délégation de conduite ou véhicule entièrement automatisé est un véhicule automobile capable de rouler — sur route ouverte — sans intervention d'un conducteur. Le concept désigne un véhicule pouvant circuler sur la voie publique dans les situations prévues de trafic sans intervention humaine. C'est une application typique du domaine de la robotique mobile dans laquelle de nombreux acteurs sont engagés.

Des éléments de solutions techniques, légales, psychologiques et juridiques ont déjà été introduits, mais certaines questions restent non résolues.

La notion de voiture autonome couvre, selon le contexte, un véhicule totalement autonome (niveau 5) ou bien un véhicule « semi-autonome » disposant de différents systèmes d'aide à la

conduite ou de conduite semi-automatisée supervisée par le conducteur (niveau 2+), ou bien encore un véhicule à délégation de conduite (niveau 3). Bien que la publicité pour certains véhicules de niveau 2+ évoque « pilotes automatiques » et « conduite entièrement autonome », ces véhicules ne sont considérés par le code pénal français ni comme automatisés, ni comme autonomes, ni comme à délégation de conduite s'ils ne répondent pas à des critères prévus pour le niveau 3.

Certains systèmes de conduite automatisée de niveau 3 sont prévus pour fonctionner dans des conditions spécifiques, par exemple dans les embouteillages sur autoroute ou lors du stationnement automatique. [1]

Historique

L'idée de la voiture autonome apparait en 1939 par le géant générale Motors. En 1950 le premier prototype a été réalisée par Ford et générale Motors suivie des autre constructeurs Japan et européen mais il reste des prototypes qui roule dans des circuits fermés et pas dans une circulation réelle.

Plusieurs tentatives isolées avaient été réalisées depuis les années 1970, mais l'évolution rapide des technologies liées aux capteurs, à la télématique et à la puissance intrinsèque des processeurs numériques, ont finalement abouti à des résultats probants et assez nombreux.

En 1977 le Laboratoire de robotique de Tsukuba au Japon fit fonctionner une automobile automatique sur un circuit dédié. Le suivi de trajectoire était réalisé par reconnaissance du marquage au sol et la vitesse de 30 km/h était atteinte.

En 1984 Mercedes-Benz testa une camionnette automatique équipée de caméras, dont le logiciel de reconnaissance était développé par une équipe de l'université de la Bundeswehr à Munich sous la direction d'Ernst Dickmanns. Le véhicule atteint 100 km/h sur un réseau routier sans trafic.

En 1986 le projet ALV (Autonomous Land Vehicle) financé par l'Agence pour les projets de recherche avancée de défense (DARPA), aboutit à un démonstrateur autonome capable de suivre une route à 30 km/h. Le Laboratoire de Robotique de l'université Carnegie-Mellon de Pittsburgh démarre le développement des véhicules automatiques Navlab.

En 1987 la Commission européenne finança le programme européen Prometheus, à hauteur

de 800 millions d'euros, qui contribua, entre autres, au développement d'outils technologiques dédiés à la conduite automobile automatique. La même année, le laboratoire de recherche Hughes Aircraft (HRL) complète le véhicule ALV en le rendant tout-terrain et capable d'évoluer à 3 km/h dans un environnement complexe (végétation, rochers, ravins).

En 1994, à l'occasion d'une conférence scientifique sur le thème, Daimler-Benz réalise une démonstration, en situation réelle de trafic sur l'autoroute A1 à partir de Paris, de deux véhicules autonomes (VaMP and Vita-2) pilotés par logiciels de l'équipe de Dickmanns et capables de réaliser conduite en file, changement de file et dépassement avec une vitesse de pointe de 130 km/h.

En 1995 un de ces véhicules réalisa le trajet Munich-Copenhague et retour (1600 km) avec une vitesse atteinte de 175 km/h. La plus longue section de conduite automatique continue fut 158 km. La même année un véhicule Navlab réalisa l'opération No Hands Across America sur le trajet de Washington, D.C. à San Diego.

En août 1997 a lieu à San Diego une importante démonstration du « consortium américain de l'autoroute automatisée » National Automated Highway System Consortium (NAHSC) où divers véhicules autonomes peuvent être comparés. À cette occasion une infrastructure spécifique avait été préparée par l'insertion de plots magnétiques servant au guidage dans certaines sections d'autoroute. [1]

Rapport du département des transports des États-Unis(Années 1960 à 2015)

Les faits relatés ci-après sont tirés d'un rapport du département des transports des États-Unis. Par conséquent, il est possible que l'exposition de certains faits montre un caractère tendancieux en faveur de la politique américaine.

Les années 1960 marquent l'émergence des systèmes intelligents. Le nombre de véhicule en circulation passe désormais le cap des 75 millions. Pour des raisons de sécurité, des normes commencent à être mises en place par les agences du gouvernement des États-Unis, donnant naissance au développement de nouvelles technologies intelligentes comme les ceintures de sécurité ou les airbags par exemple. General Motors, un constructeur automobile américain, développe DAIR (Driver Aided Information and Routing System), un système connecté intégré dans une

voiture, permettant à la fois de recevoir des informations sur la direction à prendre, d'obtenir des informations sur les conditions routières ou bien d'envoyer des messages d'urgence à une centrale. Cependant, en raison du manque de ressources pour déployer les infrastructures, le projet est abandonné.

Durant la fin des années 1960, vient ensuite le projet du Bureau des voies publiques (Federal Highway Administration), ERGS (Experimental Route Guidance System). Ce système permet la communication entre plusieurs véhicules. Plusieurs constructeurs automobiles américains, comme General Motors bien Philco-Ford se penchent sur le sujet. Il y a des tentatives d'essais avec des prototypes mais sans grand succès. En 1970, le projet est abandonné car les infrastructures qui auraient été nécessaires auraient coûté trop cher.

Dans les années 1970, avec l'avancée technologique et les progrès mathématiques, des algorithmes peuvent être utilisés pour modéliser les routes et les stocker dans des bases de données. Robert L. French développe alors l'ARCS (Automatic Route Control System), le premier système de guidage autonome. Ce système utilise des voix pré-enregistrées pour donner les indications routières au chauffeur, mais comme le système est loin d'être parfait, la deuxième version de ARCS inclut un écran affichant visuellement les informations.

En 1977, une équipe japonaise du laboratoire de Tsukuba fait rouler la première voiture capable de suivre une voie de signalisation grâce à des capteurs optiques.

Dans les années 1980, la sécurité et l'environnement sont au cœur de la politique des transports aux États-Unis. La cause de ce changement est notamment le nombre élevé de dysfonctionnements de véhicules en 1980 et la diminution des ressources pétrolières depuis 1970, au point qu'un mandat a été mis en place pour que les nouveaux véhicules respectent[pas clair] une certaine norme[Laquelle ?]. La technologie devient cependant meilleure et plus accessible, permettant l'émergence de plus d'applications dans le domaine des transports. Pendant cette période, des programmes de recherche concernant le développement technologique des transports font leur apparition, tel que PATH (The California Program On Advanced Technology For The Highway) qui est encore actif et un des leaders concernant les systèmes de transports intelligents aujourd'hui.

En 1986, le camion VaMoRs — équipe de Ernst Dickmanns — est le premier véhicule se déplaçant quasiment sans intervention humaine grâce à des caméras, des capteurs et un ordinateur contrôlant le volant et la vitesse.

Dans les années 1990, peu après la fin de la Guerre froide et de la chute du mur de Berlin, les États-Unis profitent de la paix pour faire des progrès dans le domaine de l'industrie, des transports et de la santé. On assiste aussi à l'émergence d'Internet avant la fin du siècle. L'association ITS America est fondée en 1991 par American Association of State Highway and Transportation Officials (AASHTO), le Transportation Research Board (TRB) et l'Institute of Transportation Engineers. Son principal but est de faciliter la collaboration de compagnies privées ou d'agences publiques pour le développement des systèmes de transports intelligents.

Dans les années 2000, les progrès technologiques, surtout dans le domaine de la communication, permettent aux systèmes intelligents de transport de faire un bond en avant. En effet, les objets connectés, par exemple les smartphones permettent désormais à l'utilisateur de recevoir des informations en temps réel sur les transports et le trafic par le biais d'applications. Mais cela marche aussi dans l'autre sens, c'est-à-dire que le voyageur partage à son tour ses informations (comme sa position par exemple) en temps réel qui peuvent être collectées dans une base de données et être analysées. Les smartphones ont joué un rôle majeur dans le développement des systèmes automatisés de transport, car les utilisateurs peuvent désormais envisager un avenir où les moyens de transports seraient composés en grande partie de véhicules autonomes.

En 2007, des mails dévoilés par le Guardian auraient commencé à être échangés entre Uber et le gouverneur de l'Arizona pour autoriser secrètement les premières voitures autonomes Uber.

Après 2010 et la crise économique, le but était d'adopter une utilisation plus efficace du réseau routier et du parc automobile. Par ailleurs, avec l'évolution rapide des technologies de la communication et de l'information, de nombreuses applications de transport liant des parcs de véhicules localisés géographiquement à des interfaces utilisateurs intuitives sont arrivées sur le marché.

En 2009, le projet Auto-Driving Car de Google a débuté. À l'origine, ce projet équipait des

véhicules existants, comme la Toyota Prius et la Lexus RX 450h. Cependant, Google a aussi conçu son propre prototype qui a notamment comme particularité d'abandonner le volant et les pédales. Ils testent actuellement ce prototype dans plusieurs villes aux États-Unis. Leur technologie peut atteindre une automatisation de niveau.

De 2012 à 2013 à Ann Arbor, au Michigan a eu lieu un test grandeur nature de la technologie des véhicules connectés. Cet événement a rassemblé environ 2 700 véhicules. Chaque véhicule était équipé de la technologie aidant à éviter les accidents lors du parcours de son itinéraire. Les conducteurs reçoivent des alertes telles que le freinage des véhicules, les véhicules en angle mort et les non-respects des feux de signalisation.

En août 2014, la National Highway Traffic Safety Administration (NHTSA) a publié un rapport de recherche sur la technologie de communication entre véhicules V2V. On trouve dans ce rapport les résultats de recherches menées par les ministères de la faisabilité technique, la vie privée et la sécurité. Ces recherches montrent que deux applications de sécurité : Left Turn Assist et Intersection Movement Assist, pourraient empêcher jusqu'à 592 000 accidents et sauver 1 083 vies par an.

Le Département des Transports des États-Unis (USDOT) a sélectionné en septembre 2015 trois sites de déploiement de véhicules connectés afin de réaliser des tests grandeur nature. Premièrement, les technologies de véhicules connectés sont utilisées dans le sud du Wyoming pour rendre le transit des camions plus sûr et plus efficace. Deuxièmement, on utilise la technologie V2V ainsi que la communication des intersections pour fluidifier et sécuriser la circulation sur les grands axes New-Yorkais. Troisièmement, de nombreuses applications de mobilité ont été déployées à Tempa en Floride.

Projet CATS en Europe

CATS est un projet de recherche européen qui a duré cinq ans (de 2010 à 2014) mené dans le cadre du FP7 (en français le septième programme-cadre de l'Union européenne pour la recherche et le développement technologique) et dont l'objectif a été d'étudier la faisabilité d'une mise en place d'un système de transport basé sur des véhicules électriques autonomes.

Les paragraphes suivants se réfèrent à l'article « Pioneering driverless electric vehicles in Europe: the City Automated Transport System (CATS) » écrit par Derek Christie, Anne Koymans, Thierry Chanard, Jean-Marc Lasgouttes, et Vincent Kaufmann.

Le projet commence le 1er janvier 2010 et l'objectif primaire est d'encourager le déploiement de Cristal, le véhicule autonome développé et créé par le groupe privé français Lohr Industrie, spécialisé dans la conception et la commercialisation de systèmes de transports de biens. C'est après avoir effectué des analyses concernant les besoins en matière de mobilité dans trois villes, qu'il a été déduit que Strasbourg était la ville la plus adaptée pour une démonstration publique. L'expérience a pu permettre la collecte de données ; des informations sur les émissions de CO2, et sur l'acceptation par le marché du système Cristal ont pu être relevées.

Changement de véhicule et de propriétaire: Le projet commence alors à rencontrer ses premiers obstacles. Lohr Industrie qui s'occupait de fournir les véhicules autonomes, fait faillite en 2013. Elle cesse alors la production de Cristal et se retire en partie du projet CATS. Afin de poursuivre le projet, un autre véhicule autonome nommé Navia est choisi pour ses nombreux points communs avec Crystal. Il est développé par une autre compagnie française qui s'appelle Induct Technology. La contribution d'Induct Technology en fin d'année 2013 au projet, permet alors l'accomplissement d'une grosse étape qui a lieu début 2014, où trois véhicules Navia circulèrent avec succès dans le parc d'innovation d'Illkirch à Strasbourg pendant plusieurs mois, hélas sans prendre de passagers pour des raisons légales. En mai 2014, Induct Technology fait faillite à son tour, et se fait racheter par un nouveau propriétaire qui est aussi intéressé par le projet CATS et renomme les véhicules en « Navya ».

De Strasbourg à Lausanne: Après une rencontre faite avec plusieurs ministères en France, la Communauté urbaine de Strasbourg reçoit sa première autorisation pour utiliser les véhicules autonomes sur le domaine public, mais pour des raisons de sécurité et légales, il est demandé de mener la suite de l'expérience dans un endroit mieux protégé. Le lieu de la démonstration est alors déplacé en Suisse à l'École polytechnique fédérale de Lausanne (EPFL). En plus d'être un pays sécurisé, la Suisse possède une politique plus flexible qui est en faveur de l'innovation et de

la création45. Cependant, le fait que la suite de l'expérience se déroule à cet endroit n'est pas sans désavantage, car, en effet, l'école regorge majoritairement de personnes intéressées par la technologie et l'innovation, et cela peut avoir une influence sur le choix des personnes à emprunter ou pas les navettes autonomes. En plus de regorger majoritairement des personnes intéressées par la technologie, la majorité des personnes se trouvant dans cette école sont des hommes. En effet, il a été compté lors de l'expérience que 66% des utilisateurs étaient des hommes.

L'expérience sur le campus de l'EPFL est supervisée par Bestmile (une startup née au sein de l'EPFL) du 10 au 31 juillet 2014 les jours de la semaine de 7 h 30 jusqu'à 18 h 00 pour un total de 168 heures sur seize jours. L'expérience suit un schéma précis : un étudiant est présent dans chaque navette pour répondre aux éventuelles questions, distribuer des questionnaires pour que les passagers partagent leur expérience et arrêter le véhicule en cas d'urgence. Des personnes plus qualifiées sont postées sur la route pour surveiller, gérer les étudiants ou bien intervenir en cas d'urgence. Mais malgré tout le dispositif mis en place pour garantir le bon déroulement de l'expérience, cela n'a pas empêché les navettes de rencontrer plusieurs problèmes, notamment liés à des défauts techniques et logiciels.

Du 21 au 31 juillet, lors des mesures, il est enregistré qu'en huit jours, plus de 800 personnes ont emprunté les navettes autonomes. Un total de 181 questionnaires est distribué et complété pendant les deux semaines. Cette collecte de données permet de récolter plusieurs catégories d'informations, comme des informations liées à l'utilisateur du véhicule (l'âge, le sexe, ainsi que le métier par exemple) ou bien des avis sur la qualité et l'aspect de Navya, etc. Même si les sondages révèlent que l'expérience a été très appréciée par la majorité des personnes, les choses sont plus compliquées au niveau législatif. Les voitures autonomes ne sont pas encore là, des lois doivent encore être modifiées et créées et cela peut prendre un temps considérable. Les auteurs de l'article concluent que seuls, les progrès et les innovations technologiques ne sont pas suffisants pour le développement des véhicules autonomes, mais qu'il faut aussi que les puissances politiques agissent pour que des expériences et des tests puissent encore avoir lieu. [1]

Autres Projets

De nombreux acteurs travaillent sur des projets de voiture autonome : constructeurs automobiles Audi, Toyota, Renault, Nissan, Peugeot, General Motors, Mercedes-Benz, ou encore Tesla mais aussi des équipementiers comme Valeo, Continental ou Bosch. Google, un acteur pour qui ce n'est pas le cœur de métier, développe également son système. Malgré le départ d'un des ingénieurs principaux du projet et un accident en 2016, la société reste un acteur emblématique du secteur. Apple semble aussi vouloir se positionner sur le marché, avec son projet Titan.

En octobre 2010, Google annonce avoir conçu son système de pilotage automatique pour automobile, qu'il a installé sur huit véhicules. Anthony Levandowski et Sebastian Thrun sont impliqués dans ce projet.

En août 2013, Nissan annonce vouloir commercialiser ses premières voitures sans conducteur en 2020. Volvo qui travaille depuis sur les années 1970 sur l'accidentologie de ses véhicules souhaite aussi proposer un modèle sans accident avant 2020.

Parmi les constructeurs automobiles français, plusieurs ont un projet de voiture autonome à l'instar du groupe PSA qui fait circuler depuis l'été 2015 sur route ouverte plusieurs véhicules de type Citroën C4 Picasso14, tandis que Ligier a déployé la navette EasyMile EZ.

Fin 2015, la FIA annonce le lancement en 2016-2017 d'un championnat de voitures électriques sans conducteur, Roborace.

À l'image de sa tentative avortée à San Francisco pour un problème réglementaire, Uber entend déployer massivement des voitures sans conducteur, ce que son fondateur Travis Kalanick juge « existentiel » pour sa société : celle-ci y voit en effet un moyen de baisser ses prix. Le 20 octobre 2016, un semi-remorque Otto, entreprise rachetée par Uber en août 2016, a effectué la première livraison mondiale par un camion autonome, sans chauffeur, en mode de pilotage automatique sur une autoroute entre Fort Collins et Colorado Springs soit un trajet de 200 kilomètres.

Une étude de PwC, réalisée en 2016 aux États-Unis, révèle que 66 % de la technologie utilisée par les voitures autonomes est aussi fiable qu'un conducteur moyen. Les principales peurs partagées sont les risques d'accidents et de vol. Seuls 13 % des consommateurs interrogés ne voient

aucun avantage dans ce type de voiture.

Tesla envisage de commercialiser des fonctions de conduite autonome à partir d'aout 2018.

Le 16 août 2018, la start-up américaine Nuro annonce le lancement d'expérimentation de service de livraison autonome sur les routes d'Arizona, en partenariat avec le retailer Kroger.

Les types des voitures autonomes

Il existe deux types des voitures autonomes :

Le premier(self-contained)

Ce type est basé sur les information que la voiture collecte par des capteur, des radar et camera à 360 °.

Le deuxième type(interconnected)

C'est-à-dire que la voiture et connecté à plusieurs réseaux. Pour avoir des informations de météo, des donné GPS etc. Ce type représente bien la voiture automatisée.

Le fonctionnement de la voiture autonome

La voiture autonome est basée sur les informations collectées par les capteurs laser (lidar), les radars, Les cameras qui sont placée autour de la voiture pour modélisée l'environnement en 3 dimensions et identifier les éléments essentielle (l'être humain, les animaux, les bâtiments, les trottoirs, les lignes de la route, les plaque de signalisations. . .),et elle aussi connecte au GPS pour avoir les information de la route(direction, la vitesse de la route, les chemin optimisée, les écoles, les usine, les restaurant . . .).

1.5 Composition de la voiture autonome

1.5.1 Matériel

Les cameras

Le rôle des cameras est très important, c'est grâceà elle la voiture peut identifier les diffèrent élément avec le processus reconnaissance des objet (Object recognition).la détection et la

reconnaissance des objets est fait par des bibliothèquegraphique Prédéfinie dans les langages de programmation comme opencv (open computer vision), c'est la plus utilisée dans ce domaine.

Les radars

Le rôle des radars est la détection des obstacles ensuite calculé la distance pour éviter les collusions et assurée la sécurité du conducteur et la voiture aussi.

Logicielle

Apprêt tout le matérielles que la voiture autonome contient ils ne s'airent à rien San le programme intelligent qui est responsable de tous les calculs .il fait la réception des informations envoyer par le capteur et les traite après il prend des décisions pour les envoyer a les actionneurs.

1.6 Les diffèrent niveau de la voiture autonome:

L'automatisation des voitures a été développé lentement, à cause des exigences de sécurité et assurance de déroulement de processus de la conduite autonome, Jusqu'au aujourd'hui il existe 6 niveaux: [1]

1.6.1 Le niveau 0

C'est ou le conducteur est responsable toute la conduite et surveillé La route en cas des dangers, donc la voiture n'assite rien, elle peut afficher juste les avertissements de danger (manque d'essence, manque d'huile, manque de l'eau de refroidissement, température. . .)

Aucune électronique d'assistance. C'est par exemple une 2CV mais plus près de nous une R5 ainsi qu'une grande majorité de voitures produites avant 2004 (année où l'ABS s'est généralisé en Europe). Tout est régi par de la mécanique, sans intelligence logicielle pour piloter les organes de sécurité.

1.6.2 Le niveau 1(eyes-on hand-on)

Dans le niveau 1 la voiture entre un peut dans le processus de conduite, elle peut contrôlée la vitesse et la distance par apport aux voitures précédente et le contrôle de suivi des lignes blanche.

Des composants électroniques peuvent influer sur la conduite, que ce soit au niveau du

freinage (ABS, ESP, freinage automatique d'urgence), du maintien des distances (ACC), de la direction ou de l'aide au stationnement (Park Assist). Mais, ils ne sont pas reliés entre eux. Le véhicule ne peut gérer qu'une action à la fois.

1.6.3 Le niveau 2(eyes-on hand-off)

Dans le niveau 2 la voiture peut contrôler la vitesse et la distance comme le niveau 1 on ajoutant la surveillance de l'environnement mais il faut que le conducteur soit prêt en cas de nécessité car la voiture n'a pas toutes les informations sur l'environnement.

Les systèmes d'aide à la conduite sont combinés et peuvent gérer automatiquement des manœuvres simultanées, comme par exemple le fait d'avancer et de freiner dans les bouchons tout en agissant sur la direction, ou changer de file sur autoroute pour dépasser quand le régulateur de vitesse adaptatif est activé. Le conducteur peut retirer ses pieds des pédales, mais il n'a pas le droit de lâcher des mains le volant. Une alerte se déclenche d'ailleurs quand les capteurs ne détectent plus une présence humaine. Certaines voitures (comme chez Volkswagen) vont jusqu'à s'arrêter si personne ne reprend le contrôle. Le niveau 2 est aujourd'hui la norme chez tous les constructeurs, y compris Tesla.

1.6.4 Le niveau 3(eyes-off hand-off)

Dans le niveau 3 la voiture est plus autonome que les niveaux précèdent Elle est capable de contrôler la vitesse et la distance et de détecter les paramètres de son entourage mais il faut que le conducteur soit prêt à tout moment de prendre le contrôle, car la voiture est autonome dans des conditions prédéfinies comme les autoroutes.

On passe à un cran au-dessus. Le système de pilotage est plus évolué et permet, lors de certaines phases de conduite, de lâcher les mains pour peu que la réglementation l'autorise. C'est précisément le cas au japon pour le système Traffic Jam Pilot de Honda, qui peut contrôler ce que fait le véhicule en cas d'embouteillage sur autoroute, dans une plage de vitesse de 0 à 50 km/h. Rien de très spectaculaire, mais c'est tout de même une première. Pour délivrer cette autorisation, les autorités japonaises ont fixé des conditions, dont la présence d'un enregistreur de

bord et d'un sticker identifiant le véhicule. Audi avait prévu en 2018 de proposer le niveau 3 sur l'A8 en Allemagne. Mais il n'y était pas arrivé, faute d'une évolution de la législation européenne. Et la marque n'a pas prévu de doter sa remplaçante du même système en 2021. Précisons qu'au niveau 3, le conducteur est tenu de superviser que ce fait le véhicule et doit pouvoir reprendre le volant en quelques secondes.

1.6.5 Le niveau 4

Le niveau 4 est presque le même que le niveau 3 mais plus développé grâce aux capteurs et radars avec une précision plus élevés. Dans des conditions prédéfinies le conducteur n'est plus responsable à la conduite.

On commence à entrer vraiment dans l'ère du véhicule autonome. Encore plus évolué, le système de pilotage permet cette fois au conducteur de faire autre chose. Il peut par exemple lire, consulter ses messages ou regarder un film pendant que le véhicule est conduit par l'intelligence artificielle. Toutefois, ce niveau ne s'applique pas partout. Il peut être activé par exemple sur autoroute, ou dans des zones bien délimitées et pas forcément ouvertes au trafic. Dans les faits, ce sont les navettes autonomes et les robots-taxis qui accèdent à ce niveau. Et encore, il s'agit encore pour la plupart de véhicules utilisés dans le cadre d'expérimentations et pouvant être contrôlés à distance. En Chine et aux USA (en particulier dans le Nevada et en Californie), des constructeurs, équipementiers et start-ups ont des permis pour tester des véhicules de ce type. Après avoir promis le niveau 4 dès 2021 (BMW, Ford), les constructeurs se sont ravisés et se montrent plus prudents.

1.6.6 Le niveau 5

Dans ce niveau la voiture est totalement autonome , elle est capable et responsable à toutes les surveillances et les décisions, la voiture peut faire un trajet complet dans les autoroutes sans aucune intervention humaine.

C'est le stade ultime. Le véhicule peut se conduire tout seul, partout et par tous les temps et sans intervention humaine. C'est super, sauf qu'on n'y arrivera peut-être jamais (et ce sont des personnes qualifiées qui le disent, comme le PDG de Waymo – filiale de Google – et le responsable

du développement de la technologie chez Volkswagen). Après avoir été survendue et généré un buzz incroyable, la voiture autonome fait moins rêver. Il faut dire que c'est bien plus compliqué que prévu, en raison des besoins en matière de cartes, de précision au centimètre de la localisation, d'une communication permanente et d'une sûreté de fonctionnement se rapprochant des standards d'un centrale nucléaire et de l'aéronautique.

Conclusion

Dans ce chapitre donné une brève information sur la robotique et ses domaines d'utilisation ainsi que le développement de la voiture autonome.

En résumé, les constructeurs n'ont donc pas renoncé à la voiture autonome. Mais, l'horizon se situe plutôt vers 2030 ou au-delà. Même dans les pays réputés plus libéraux, on prend le temps de préparer le cadre pour autoriser ces véhicules sur route ouverte. C'est le cas aux Etats-Unis, où l'agence en charge de la sécurité routière, la NHTSA, a engagé un protocole qui doit faire la synthèse entre les préoccupations du public en matière de sécurité et la volonté des industriels de promouvoir l'innovation. Le processus devrait prendre des années.

Chapitre 2

Cahier de charge et conception

Introduction

Dans ce chapitre nous abordons les notions de base nécessaires et des techniques de traitement d'images. Tels que : la définition d'image, les types et caractéristiques d'image, les opérations ou les traitements possibles appliquer sur l'image, et de présenter tous les algorithmes utilisés dans ce projet a fin de bien comprendre les programmes. Et nous allons proposé un cahier de charge a fin de pouvoir le réalisé.

2.1 Généralités sur le traitement d'images

2.1.1 Définition de l'image

Une image est une représentation visuelle de description d'un objet ou personne par dessin, peinture, photographie, films,...etc. C'est aussi derrière l'affichage sur écran c'est un ensemble organisé d'informations, acquise, créée, traitée ou stockée sous forme binaire (suite de 0 et de 1). Ce qui nécessite sa numérisation.(Figure 2.1) [4]

Figure 2.1: Image numérique

2.1.2 Image numérique (numérisée)

L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs.

La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques f(x,y), comme la montre la figure 1.2 où (x,y) coordonnées cartésiennes d'un point de l'image. f(x, y) : niveau d'intensité. La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur(Figure 2.2) [4].

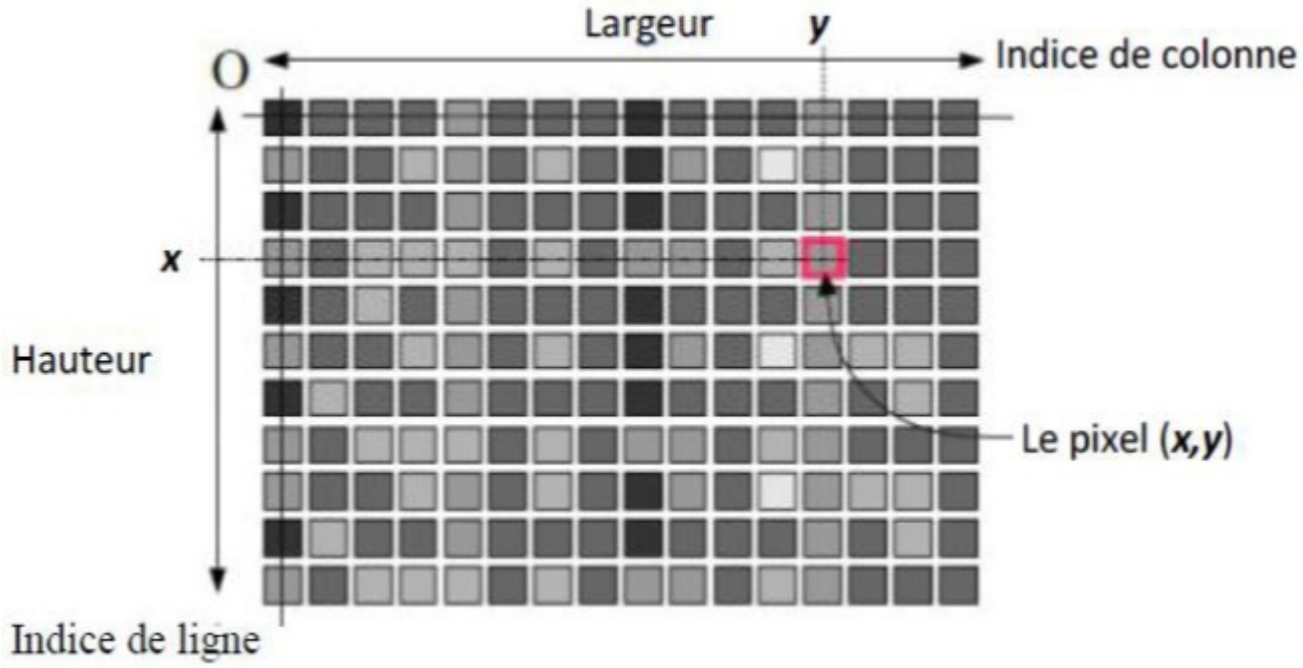


Figure 2.2: Représentation d'image numérique

2.1.3 Caractéristiques d'une image numérique

Comme nous l'avons déjà dit, l'image est un ensemble organisé d'informations qui contient les caractéristiques suivantes :

Pixel

Une image numérique est constituée d'un ensemble de points appelés pixels (abréviation de Picture Elément) pour former une image. Le pixel représente ainsi le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image. Par exemple, peut être affichée comme un groupe de pixels (Figure 2.3) [4].

Figure 2.3: Représentation en groupe de Pixel d'une image

La résolution

La résolution est un terme souvent confondu avec la "définition", détermine par contre le nombre de points ou pixels par unité de surface, exprimé en points par pouce (PPP, en anglais DPI pour Dots Per Inch), un pouce représentant 2.54 cm (Figure 2.4) [4].

Formule de conversion cm, Pixel, Pouce

$$\left.\begin{array}{l} 1\ cm = 37.79527559055\ pixel \\ 100\ pixel = 2.646\ cm \\ 1\ pouce = 2.54\ cm \\ 1\ cm = 0.3937\ pouce \end{array}\right\} \quad \text{alors 1 pouce} = 72 \text{ pixel} \tag{2.1}$$

Figure 2.4: Représentation de Résolution d'une image

Dimension

Nous appellons dimension, le nombre de points (pixel) constituant l'image, c.à.d. sa (dimension informatique). Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image (Figure 2.5) [4].

Figure 2.5: Représentation de dimension d'une image

La texture

Une texture est une région dans une image numérique qui a des caractéristiques homogènes. Ces caractéristiques sont, par exemple, un motif basique qui se répète. La texture est composée de Texel, l'équivalent des pixels.

Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur (Figure 2.6) [4].

(a) Image sans bruit (b) Image avec bruit

Figure 2.6: Image sans et avec bruit

La luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet(Figure 2.7) [4].

Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses[4].
- L'absence de parasites.

Figure 2.7: Image représente l'éffet de luminance

Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image. Il permet de donner un grand nombre d'informations sur la distribution des niveaux de gris (Couleur) et de voir entre quelles bornes est repartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée(Figure 2.8) [4].

Figure 2.8: Image avec histogramme

Le contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L1 et L2 sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 d'une image, le contraste C est défini par le rapport [4]:

$$C = \frac{L_1 - L_2}{L_1 + L_2} \tag{2.2}$$

2.1.4 Types d'images

On distingue deux types d'images :

Images matricielles

Dans la description que nous avons faite jusqu'à présent des images nous avons utilisé une matrice. On dit alors que l'image est matricielle ou en anglais bitmap. Ce type d'image est adapté à l'affichage sur écran mais peu adapté pour l'impression car bien souvent la résolution est faible (couramment de 72 à 150 ppp pour les images sur Internet).

Images vectorielles

Le principe des images vectorielles est de représenter les données de l'image à l'aide de formules mathématiques. Cela permet alors d'agrandir l'image indéfiniment sans perte de qualité et d'obtenir un faible encombrement.

Par exemple, pour décrire un cercle dans une image, il suffit de noter la position de son centre et la valeur de son rayon plutôt que l'ensemble des points de son contour. Ce type est généralement obtenu à partir d'une image de synthèse créée par logiciel (exemple : Autocad) et non pas à partir d'un objet réel. Ce type est donc particulièrement adapté pour le travail de redimensionnement d'images, la cartographie ou l'infographie [4].

2.1.5 Codages des couleurs

Nous avons vu qu'une image apparait comme une matrice où chaque case contient des nombres associés à une couleur. Usuellement on distingue 3 grands types de couleurs pour une image numérique :

- Le noir et blanc.
- Les niveaux de gris.
- La couleur.

Ces types sont généralement à choisir lors d'une numérisation par scanner ou lors de la configuration d'un appareil photographique .

Image noir et blanc

Le noir et blanc est le plus simple. Le contenu de chaque case de la matrice est soit un 0 (noir) soit 1 (blanc). Le nombre de couleurs n'est que de 2 et le rendu de l'image le moins performant mais parfois suffisant dans le cadre par exemple de documents scripturaux(Figure 2.9) [4].

Figure 2.9: Comparaison du volume d'une image pour différents codages

Niveaux de gris

Le codage dit en niveaux de gris permet d'obtenir plus de nuances que le simple noir et blanc. Il offre des possibilités supplémentaires pour coder le niveau de l'intensité lumineuse. La couleur est codée souvent sur un octet soit 8 bits ce qui offre la possibilité d'obtenir 256 niveau de gris (0 pour le noir et 255 pour le blanc). On peut aussi le faire avec 16 niveaux de gris (4 bits)(Figure 2.10).

Figure 2.10: Image en niveaux de gris

Image couleur

Principe

La couleur d'un pixel est obtenue, comme le ferait un peintre, par le mélange de couleurs fondamentales. Il ne s'agit pas ici de décrire toutes les techniques utilisées. Nous allons décrire un des principes les plus couramment utilisé qui est celui de la synthèse additive [4].

Codage RVB

Le principe consiste à mélanger les 3 couleurs : rouge, vert et bleu (noté RVB ou RGB en anglais). A l'aide de ces 3 couleurs, on obtient toute une palette de nuances allant du noir au blanc. A chaque couleur est associé un octet (donc 256 niveaux de luminosité) de chacune des couleurs fondamentales (Figure 2.11) [4].

Rouge	Vert	Bleu	Couleur
0	0	0	Noir
0	0	1	Nuance de noir
255	0	0	Rouge
0	255	0	Vert
0	0	255	Bleu
128	128	128	Gris
255	255	255	Blanc

Figure 2.11: Principe codage de la couleur

Un pixel "couleur" est alors codé avec 3 octets et on a alors la possibilité d'obtenir 224 possibilités de couleurs soit de l'ordre de 16 millions de couleurs différentes(Figure 2.12).

Figure 2.12: Valeurs RVB d'un pixel

Question comment convertit-on une image couleur en une image en niveau de gris?

$$NdG = \frac{R + G + B}{3} \tag{2.3}$$

$$NdG = \frac{\log(R) + \log(G) + \log(B)}{3} \tag{2.4}$$

$$NdG = 0.30R + 0.59G + 0.11B \tag{2.5}$$

2.1.6 Formats d'image

Lors de son enregistrement, une image est stockée suivant un format d'image précis. Ce format doit permettre de stocker l'information de l'image avec un minimum de perte d'informations. Il existe ainsi différents formats qui pourront favoriser soit la conservation de la qualité soit la diminution de la taille du fichier informatique.

Le tableau suivant donne les principales caractéristiques des principaux standards utilisés(Figure 2.13) [4]:

Format	Type	Compression données	Nombre couleurs	Affichage progressif	Usage
BMP	matriciel	non	de 2 à 16 millions	non	Image non dégradée ; Taille fichier importante.
JPG	Matriciel	oui	16 millions	oui	Taux de compression réglable ; Perte de qualité.
GIF	Matriciel	oui	De 2 à 256 couleurs	oui	Pas de perte de qualité ; Usage pour Internet.
TIFF	Matriciel	oui	16 millions	non	Pas d'usage Internet
PNG	Matriciel	oui	de 2 à 16 millions	Oui	Recommandé pour Internet
SVG	Vectoriel	oui	16 millions	non	Usage cartographie, animations

Figure 2.13: Principaux formats d'image

2.1.7 Le système de traitement d'images

Nous pouvons appliquer plusieurs traitements sur l'image numérique, la figure suivante résume l'ensemble de ces traitements(Figure 2.14).

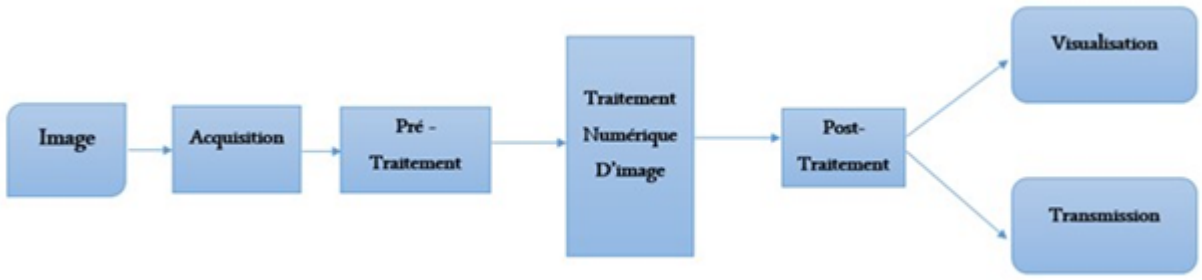


Figure 2.14: Schéma d'un système de traitement d'images

Acquisition

Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation (échantillonnage, quantification). On utilise plus couramment des caméras vidéo, des appareils photos numériques. En médecine, on utilise des

imageurs IRM, TEP, scanner X, écho doppler, échographie, scintigraphie etc... [4]

Filtrage

Est une opération qui consiste à réduire le bruit contenu dans une image au moyen d'algorithmes provenant des mathématiques par l'utilisation de méthodes d'interpolation ou de la morphologie mathématique.

Nous pouvons diviser les filtres en deux grandes catégories:

1- Filtres linéaires

Le filtre est dit linéaire si la valeur du nouveau pixel est une combinaison linéaire des valeurs des pixels du voisinage.

Il y a plusieurs types de filtrage linéaire nous distinguons :

Filtre passe-bas (lissage)

Ce filtre n'affecte pas les composantes de basse fréquence dans les données d'une image, mais doit atténuer les composantes de haute fréquence. L'opération de lissage est souvent utilisée pour atténuer le bruit et les irrégularités de l'image.

Filtre passe-haut (accentuation)

Le renfoncement des contours et leur extraction s'obtiennent dans le domaine fréquentiel par l'application d'un filtre passe-haut. Le filtre digital passe-haut a des caractéristiques inverses du filtre passe-bas. Ce filtre n'affecte pas les composantes de haute fréquence d'un signal, mais doit atténuer les composantes de basse fréquence.

Filtre passe-bande (différentiation)

Cette opération est une dérivée du filtre passe-bas. Elle consiste à éliminer la redondance d'information entre l'image originale et l'image obtenue par filtrage passe-bas. Seule la différence entre l'image source et l'image traitée est conservée.

Filtre directionnel

Dans certains cas, nous cherchons à faire apparaître des détails de l'image dans une direction bien déterminée. Pour cela, nous utilisons des filtres qui opèrent suivant des directions

(horizontales, verticales et diagonales).

* Quelque méthode de filtrage linéaire :

- Filtre moyenneur.
- Filtre gaussien.

Le principale inconvénient des filtres linéaire est que la réduction de bruit s'accompagné d'un étalement des transitions entre régions. Ce problème peut être surmonté par l'utilisation des filtres non linéaires [4].

2- Filtres non linéaire

Le filtrage non-linéaire est une opération qui remplace la valeur de chaque pixel par une combinaison non-linéaire des valeurs de ses pixels voisins, ce type de filtre pallie les inconvénient majeur des filtres linéaires dont la présence des valeurs aberrantes même après filtrage et la mauvaise conservation des transitions. On trouve aussi dans les filtres non linéaire les deux types de filtrages : le Filtre passe-bas et le Filtre passe-haut [4].

* Quelque méthode de filtrage non linéaire :

- Filtre médian.
- Le filtre de Canny.
- Le filtre de laplacien.
- Le filtre de gradient.
- Les filtres de PREWITT, SOBEL, FREEMAN, ET KIRSCH.
- Symmetric Nearest Neighbor .
- Nagao.

Segmentation

A partir d'une image numérique, il convient de d'extraire les informations pertinentes en regard de l'application concernée, les traiter puis les interpréter. Le terme générique d'analyse d'images désigne l'ensemble de ces opérations.

En analyse, d'images on distingue les traitements de bas-niveau et ceux de hautniveau. Cette distinction est liée au continu sémantique des entités traitées et extraites de l'image. Les traitements de bas-niveau opèrent, en général, sur les grandeurs calculées à partir des valeurs attachées à chaque point de l'image sans faire nécessairement la liaison avec la réalité qu'elles représentent. A l'opposé, les traitements de haut-niveau s'appliquent à des entités de nature symboliques associées à une représentation de la réalité extraite de l'image; ils sont relatifs à l'interprétation et à la compréhension de l'image. La segmentation d'image est un traitement de bas-niveau qui consiste à créer une partition d'une image A en sous-ensembles Ri appelés régions, tel que [4]:

$$\forall i,\ \mathcal{R}_i \neq \emptyset \tag{2.6}$$

$$\forall i, j\ i \neq j,\ \mathcal{R}_i \cap \mathcal{R}_j = \emptyset \tag{2.7}$$

$$\mathcal{A} = \cup \mathcal{R}_i \tag{2.8}$$

Une région est un ensemble connexe de pixels ayant des caractéristiques communes (intensité, texture,...) qui les différencient des pixels des régions voisines. Dans le cas des images couleur ces caractéristiques sont déterminées à partir des composantes colorimétriques des pixels. Les connaissances utilisées sont les plus souvent du domaine de l'image numérique et du traitement du signal, donc sémantiquement assez pauvres.

Les différentes méthodes de segmentation

La segmentation définie par l'ensemble des pixels representent en deux méthodes, soit par : approche région de la segmentation ou bien par les contours de la région approche contour de la segmentation. Ces deux approches sont duales du fait que chaque région possède un contour et qu'un contour délimite forcément une région(Figure 2.15) [4].

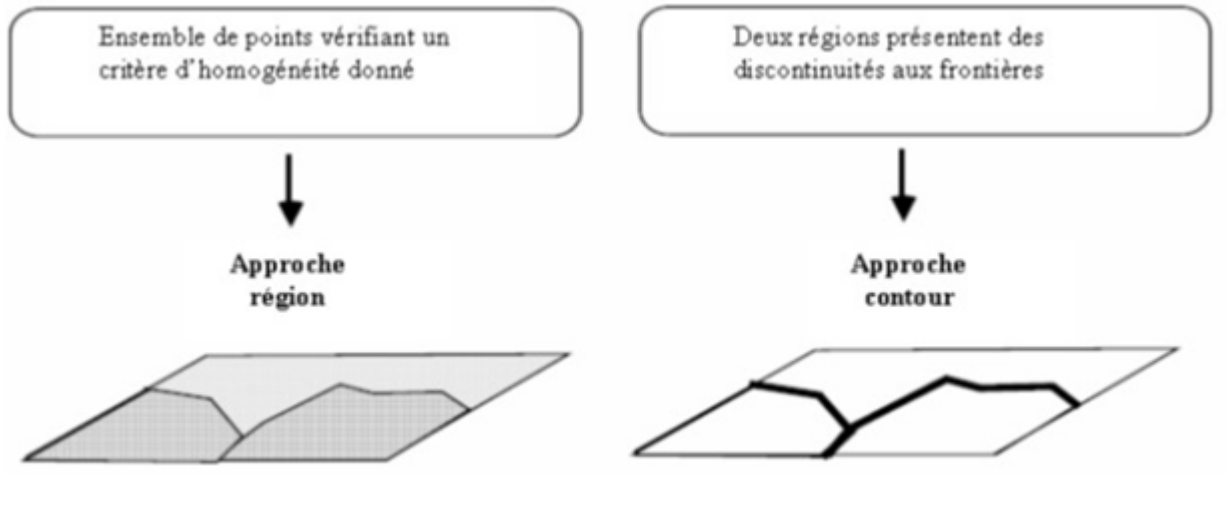


Figure 2.15: Approche région et approche contour

a - Approche "régions"

En approche région, l'affinité des points connexes est favorisée. Cela peut être vu comme une technique contextuelle. Les points connexes ayant des propriétés semblables (attributs) : intensité de gris, couleur, texture, vont être réunis dans le même ensemble.

b - Approche "contours"

Cette approche, est une technique non contextuelle qui ignore les rapports pouvant exister entre les régions de l'image. On regroupe les pixels suivant un attribut global. Elle comprend les techniques de détection de contours, mais les contours obtenus ne conduisent pas toujours directement à la partition recherchée. En effet, les pixels contours mis en évidence pour une forme, généralement ne sont pas connexes. Il faut alors appliquer des algorithmes de fermeture de contours. Ce n'est qu'après fermeture que les régions apparaissent, déterminées par l'intérieur des

contours [4].

2.2 L'outil de développement OpenCv

2.2.1 Définition

OpenCV (Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage et la société ItSeez se sont succédé au support de cette bibliothèque. Depuis 2016 et le rachat de ItSeez par Intel, le support est de nouveau assuré par Intel.

Cette bibliothèque est distribuée sous licence BSD.

NVidia a annoncé en septembre 2010 qu'il développerait des fonctions utilisant CUDA pour OpenCV [5].

2.2.2 Fonctionnalités

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes en partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques [5].

Traitement d'images

Elle propose la plupart des opérations classiques en traitement bas niveau des images [5]:

- Lecture, écriture et affichage d'une image.
- Calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs.
- Lissage, filtrage.
- Seuillage d'image (méthode d'Otsu, seuillage adaptatif).
- segmentation (composantes connexes, GrabCut).
- morphologie mathématique.

Traitement vidéos

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que [5]:

- Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra).
- Détection de droites, de segment et de cercles par Transformée de Hough.
- Détection de visages par la méthode de Viola et Jones.
- Cascade de classifieurs boostés.
- Détection de mouvement, historique du mouvement.
- Poursuite d'objets par mean-shift ou Camshift.
- Détection de points d'intérêts.
- Estimation de flux optique (Méthode de Lucas–Kanade).
- Triangulation de Delaunay.
- Diagramme de Voronoi.
- Enveloppe convexe.
- Ajustement d'une ellipse à un ensemble de points par la méthode des moindres carrés.

Algorithmes d'apprentissages

Certains algorithmes classiques dans le domaine de l'apprentissage artificiel sont aussi disponibles :

- K-means.

- AdaBoost et divers algorithmes de boosting.
- Réseau de neurones artificiels.
- Séparateur à vaste marge.
- Estimateur (statistique).
- Les arbres de décision et les forêts aléatoires.

Calculs Matriciels

Depuis la version 2.1 d'OpenCV l'accent a été mis sur les matrices et les opérations sur celles-ci. En effet, la structure de base est la matrice. Une image peut être considérée comme une matrice de pixel. Ainsi, toutes les opérations de bases des matrices sont disponibles, notamment [6]:

- La transposée.
- Calcul du déterminant.
- Inversion.
- Multiplication (par une matrice ou un scalaire).
- Calcul des valeurs propres.

Autres fonctionnalités

Elle met également à disposition quelques fonctions d'interfaces graphiques, comme les curseurs à glissière, les contrôles associés aux événements souris, ou bien l'incrustation de texte dans une image [6].

2.3 La structure d'arbitrage(Cahier de charge)

La structure d'arbitrage définit une direction qui satisfait deux objectifs, C'est un compromis entre se diriger directement vers l'objectif et se déplacer de manière sur en évitant les obstacles, Pour appliquer cette structure il faut la schématiser avec un organigramme qui résume tous les cas possibles(Figure 2.16).

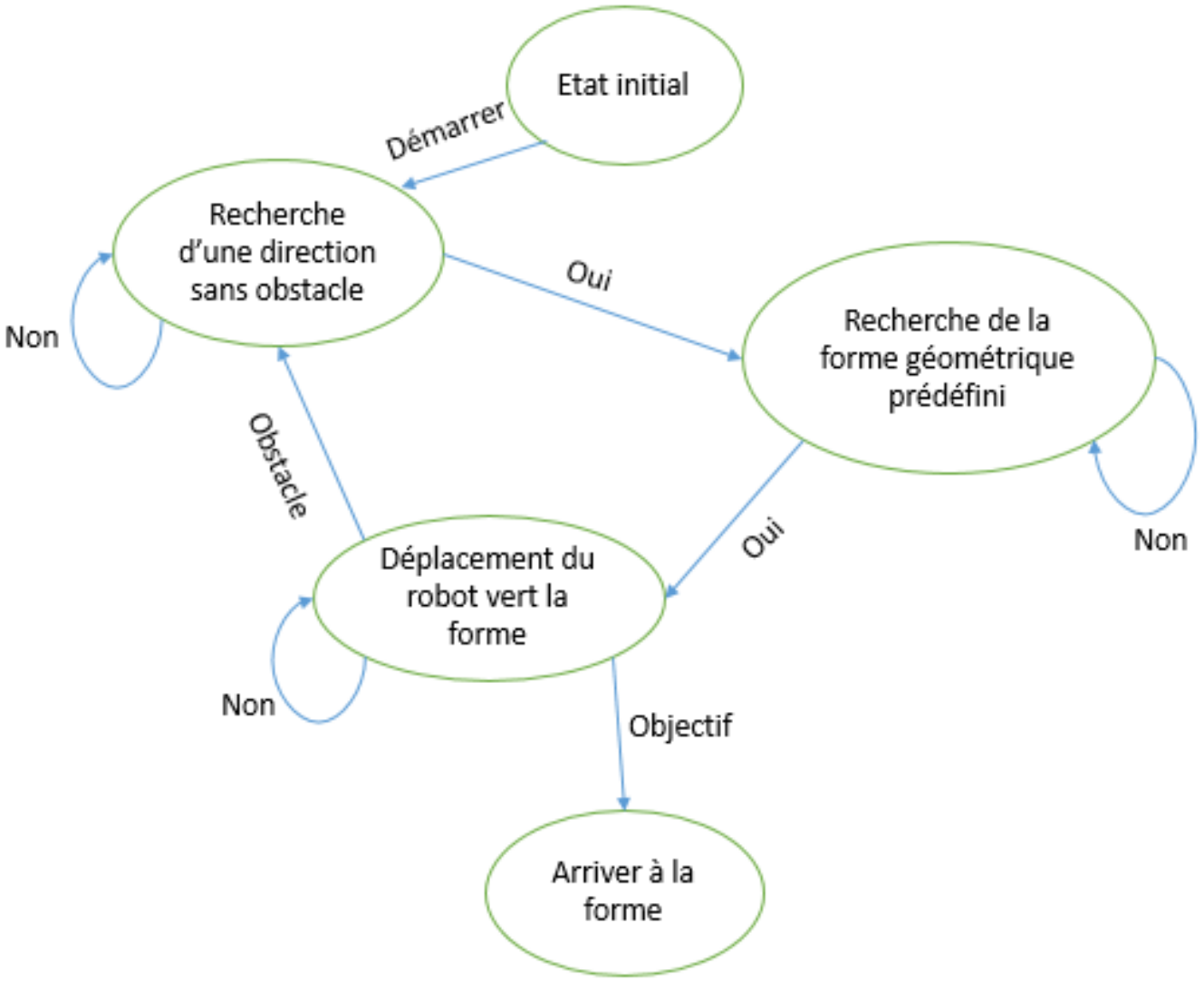


Figure 2.16: Organigramme globale

Déroulement d'organigrame

Selon la structure d'arbitrage on constate qu'on a deux traitements nécessaires, Le premier est d'évite les obstacles et le deuxième est de déterminée le chemin vers l'objectif (forme géométrique prédéfinie), a condition que le premier est prioritaire, donc il faut toujours assurée que le schemin est libre aucun obstacle present , apres on cherche la cible.

2.4 Etude des algorithmes

2.4.1 La recherche d'une direction sans obstacle

Des méthodes spécifiques ont été développées pour certains types d'objets, par exemple pour la détection de visage ou la détection de personne. Ces méthodes peuvent prendre en compte des caractéristiques spécifiques de l'objet comme le rapport largeur/hauteur, la présence des yeux et de la bouche dans le cas des visages, etc.

Dans cette partie, la détection des objets ce fait par deux capteurs ultrasons qui détectent tout type d'objet (un objet quelconque).

Ses deux capteur de distance HC-sr04 installé à l'avant du voiture avec un positionnement prédéfinie de tel sort que la zone de détection est un ongle maximale, aide à La recherche de la bonne direction.

Cette procédure consiste à choisissez la direction Approprié qui assure la sécurité de la voiture, Pour réussir ce processus, il faut suivre les étapes suivantes(Figure 2.17):

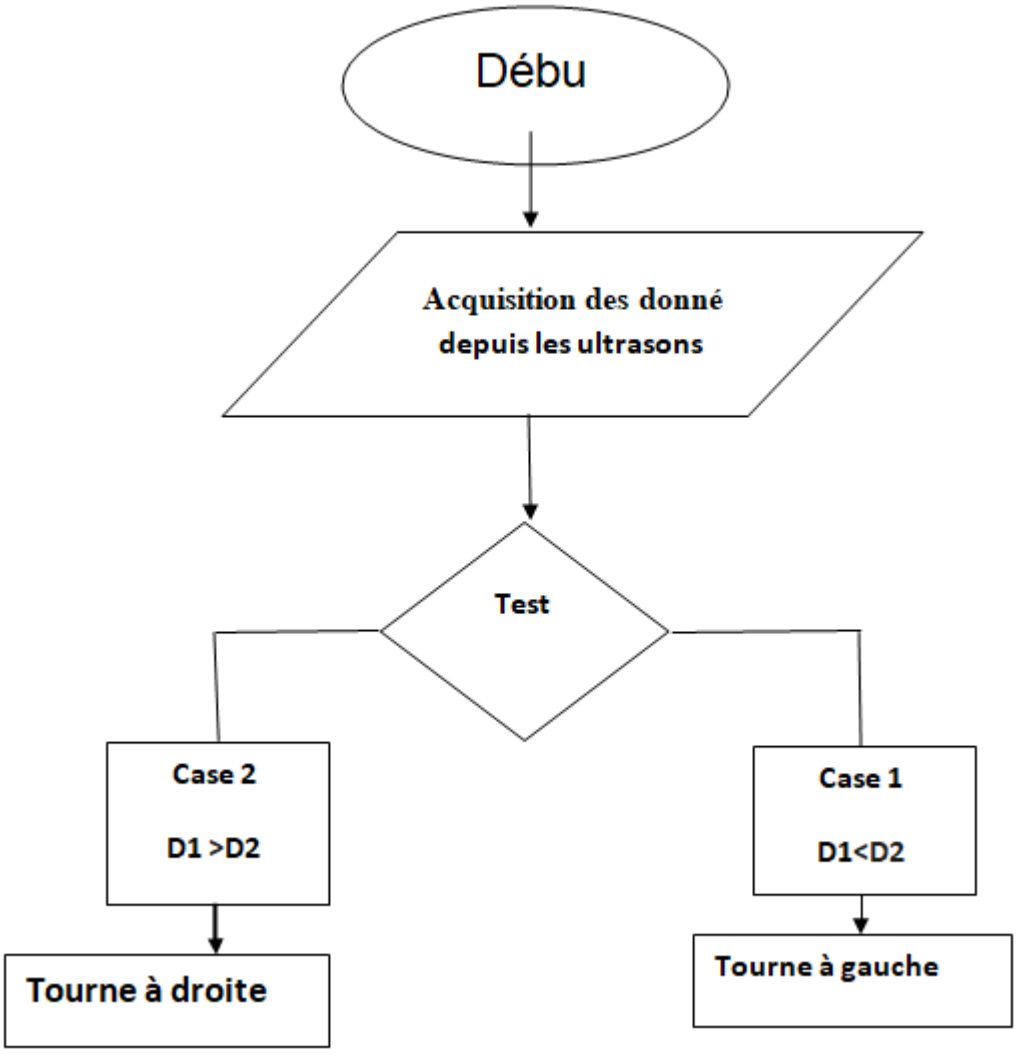


Figure 2.17: Organigramme de la recherche du cible

D1:distance mesuré par l'ultrason de la droite.

D2:distance mesuré par l'ultrason de la gauche.

2.4.2 La recherche de la cible (forme géométrique prédéfinie)

La recherche de la forme géométrique c'est la procédure la plus importante dans notre projet, cette procédure Compter sur la camera comme matérielle, avec la technique de la detection du visage qui est basé sur la bibliothèque open cv plus spécialement la caractéristique haar cascade.

Un Haar Cascade est fondamentalement un classificateur utilisé pour détecter l'objet pour lequel il a été formé.

La détection d'objets à l'aide de classificateurs en cascade basés sur les fonctionnalités Haar est une méthode efficace de détection d'objets proposée par Paul Viola et Michael Jones dans leur article intitulé «Détection rapide d'objets à l'aide d'une cascade améliorée de fonctionnalités simples» en 2001. Il s'agit d'une approche basée sur l'apprentissage automatique où la fonction de cascade est formée à partir d'un grand nombre d'images positives et négatives. Il est ensuite utilisé pour détecter des objets dans d'autres images.

Ici, nous allons travailler avec la détection de visage. Au départ, l'algorithme a besoin de beaucoup d'images positives (images de visages) et d'images négatives (images sans visages) pour entraîner le classificateur. Ensuite, nous devons en extraire des fonctionnalités. Pour cela, les fonctionnalités de haar présentées dans l'image ci-dessous (Figure 2.18) sont utilisées. Ils sont comme notre noyau convolutif. Chaque caractéristique est une valeur unique obtenue en soustrayant la somme des pixels sous le rectangle blanc de la somme des pixels sous le rectangle noir.

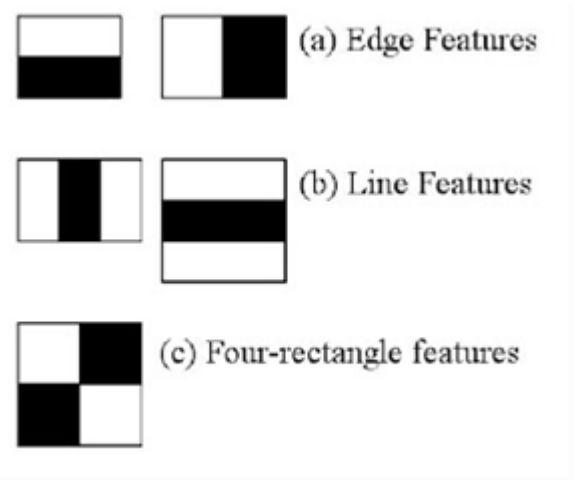


Figure 2.18: Représentation des pixels noire et blanc

Et maitenant nous appliquons un algorithme pour contrôlé le servomoteur qui porte la camera pour faire le suivi de l'objet(Figure 2.19).

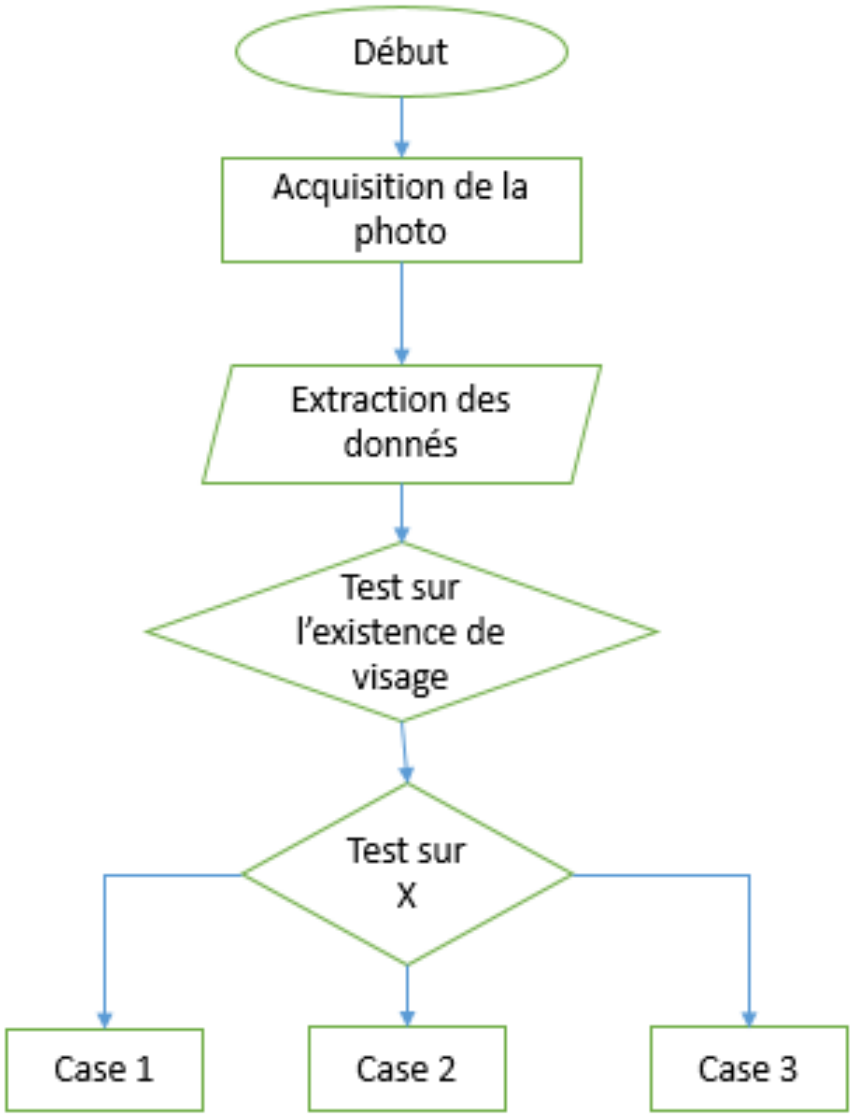


Figure 2.19: Organigrame de recherche du visage

X c'est la position de centre de visage dans l'image

Case 1 : Si x est supérieur que 200 le servomoteur va tourner la camera vers la droite

Case 2 : Si x est inferieur que 100 le servomoteur va tourner la camera vers la gauche

Case 3 : Si x est entre 100 et 200 le servomoteur ne tourne pas parce que notre objectif c'est de mettre l'objet au centre de l'image donc on va suivi la cible avec la camera et les servomoteurs.

2.5 Conception et Simulation

2.5.1 ISIS Proteus

La CAO électronique Proteus est une suite logicielle, éditée par la société Labcenter Electronics et revendue en France exclusivement par Multipower. Proteus est actuellement la seule CAO électronique qui permet la conception d'un système électronique complet et de le simuler, y compris avec le code des microcontrôleurs. Pour ce faire, elle inclut un éditeur de schéma (ISIS), un outil de placement-routage (ARES), un simulateur analogique-numérique, un environnement de développement intégré pour microcontrôleurs, un module de programmation par algorigrammes ainsi qu'un éditeur d'interface pour smartphone afin de piloter à distance des cartes Arduino ou Raspberry [7].

Proteus est un logiciel propriétaire payant. Une version de démonstration gratuite et limitée est disponible.

Quelque exemples d'utilisation de ISIS [7]:

- Filtre numérique.
- Clignotant à LED avec un microcontrôleur PIC 16F876A.
- Clignotant à LED avec un microcontrôleur PIC 18F2550.
- Chronomètre avec un microcontrôleur PIC 16F876A.
- Carte d'acquisition avec un microcontrôleur PIC 16F88.

2.5.2 Simulation de l'ARDUINO avec les autre composant

Tout d'abord il faux téléchargerez la librairie sur le site : http://www.instructables.com/id/How-to-add-Arduino-Library-in-to-Proteus-7-8/ [7].

La décompresser et copier les fichiers dans : C:/ ProgramData/ Labcenter Electronics/ Proteus 8 Professional/ LIBRARY [8].

Relancez Proteus et recherchez dans la bibliothèque de composants [8].

Importer les composantes nécessaires , et les reliés entre eux (Figure 2.20) [8].

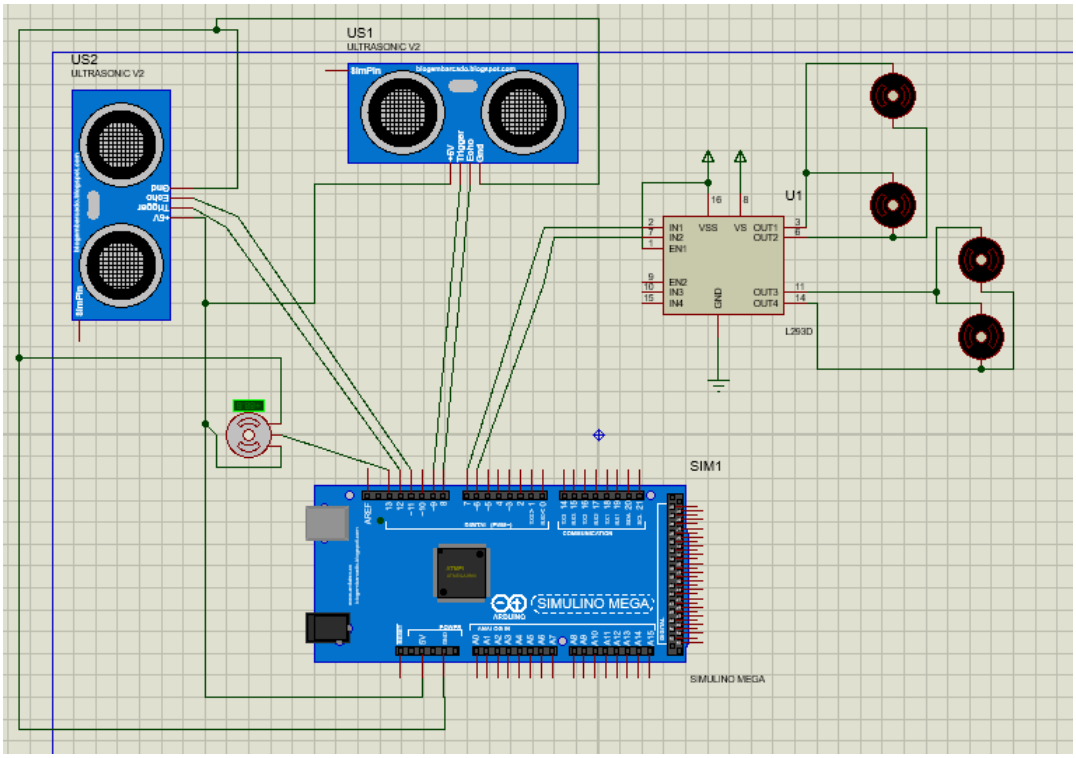


Figure 2.20: Simulation du Arduino sous isis proteus

2.5.3 Simulation du raspberry pi avec la pi camera

Visual designer pour raspberry pi est un produit révolutionnaire permet de concevoir, de simuler et de déboguer des systèmes complets de raspberry pi [8].

La bibliothèque proteus pour raspberry pi est une information importante accompagnée de photos et d'images HD provenant de tous les sites Web du monde. Téléchargez gratuitement cette image en résolution Haute Définition au choix "bouton de téléchargement" ci-dessous. Si vous ne trouvez pas la résolution exacte que vous recherchez, optez pour une résolution native ou supérieure [8].

Le hardware dans Proteus Visual Designer commence avec la galerie de périphériques. Elle contient des douzaines de blocs de circuits prêts à l'emploi qui correspondent à des vrais 'shields' Arduino ou des 'hats' Raspberry Pi. Elle inclut également de nombreux modules Grove, des

capteurs et des cartes d'expérimentation (breakout boards). L'utilisateur choisit son périphérique dans la galerie et ensuite le Visual Designer place automatiquement le circuit sur le schéma et ajoute les méthodes à glisser-déposer dans l'algorigramme pour contrôler le matériel [8]. Lorsqu'un shield/hat est ajouté, il sera automatiquement connecté à la carte processeur. Si un capteur Grove est ajouté, il pourra être édité afin de modifier son ID pour correspondre au hardware réel. Dans tous les cas, le Visual Designer contrôlera que les broches ne sont pas utilisées par plus d'un shield ou d'un capteur(Figure 2.21), (Figure 2.22) [8].

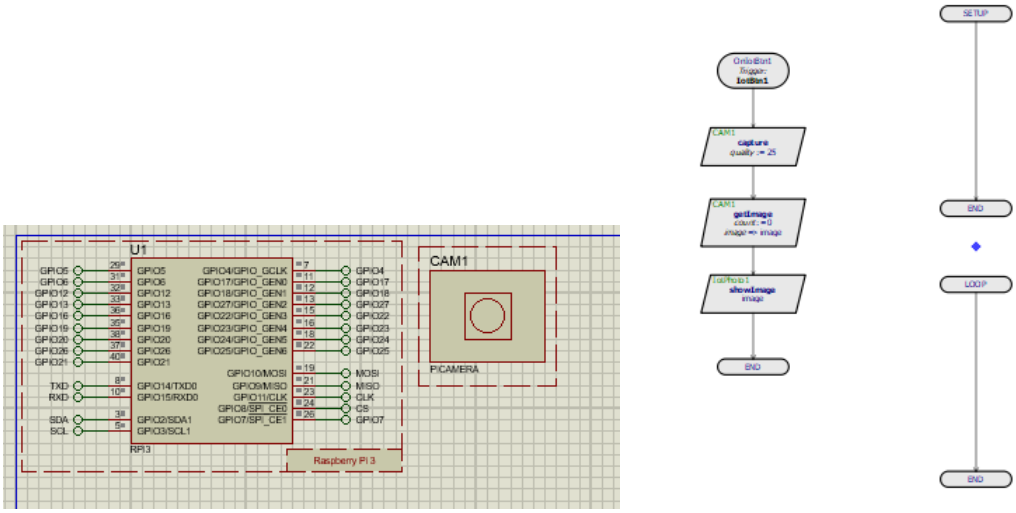


Figure 2.21: Simulation du Raspberry Pi avec la camera sous isis proteus

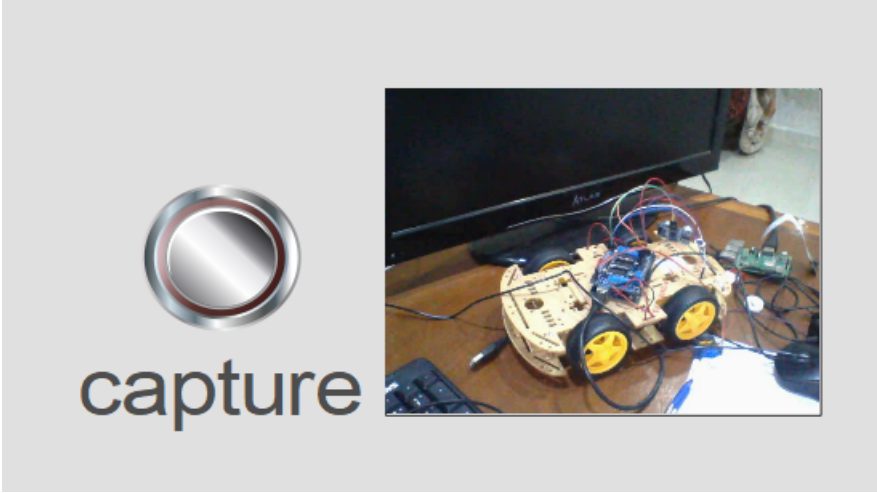


Figure 2.22: Test de la simulation

Conclusion

Dans ce chapitre nous avons présenté une masse importante d'informations concernant les méthodes utilisées dans le domaine de traitement d'imagenous, et nous avons vu les différentes éxigences a réalisé dans ce projet, et les démarches pour réalisé notre programme.

Chapitre 3

Réalisation et implémentation du programme

Introduction

Dans certains projets, il peut être intéressant d'établir une communication série entre Raspberry Pi et Arduino. Il est ainsi possible de coupler la puissance de calcul et les interface sans fil du Raspberry Pi avec les entrées-sorties et la collection de modules Arduino. Le premier exemple qui vient en tête est l'utilisation de ce système pour de la domotique dans lequel le Raspberry Pi va héberger l'interface de contrôle et l'intelligence et les Arduino vont servir d'automate programmable agissant sur les composants en bout de chaîne (lumière, radiateur, ventilateur, capteurs, etc.).

L'objectif de ce chapitre est de présenter tout le matériel utilisé dans le projet avec le principe de fonctionnement de chaque composant, et l'implémentation du programme et faire des test.

3.1 Présentation du matérièl

3.1.1 Liste de matériel utilisé dans le projet

- Le Raspberry pi 3 model b+.
- Arduino méga
- Pi caméra (camera de 5 pmx du Raspberry pi)
- 2 servomoteur
- 2 moteurs à courant continu
- 2 ultrasonshc-sr04
- Shield des moteur l293d
- Télécommande IR
- VoitureRC

3.1.2 Présentation et fonctionnement de chaque composant

Raspberry pi

Le Raspberry pi est une carte logique a la taille d'une carte de crédit fonctionne sous linux comme système d'exploitation, Il a toutes les fonctionnalités d'un ordinateur limité eaux performances A cause de sa taille, mais la chose que lui rendre très intéressent c'est le prix, à 39 euros on peut réaliser beaucoup projet, avec une puissance acceptable, dans notre projet on va utiliser la version 3 b+(Figure 3.1) [9].

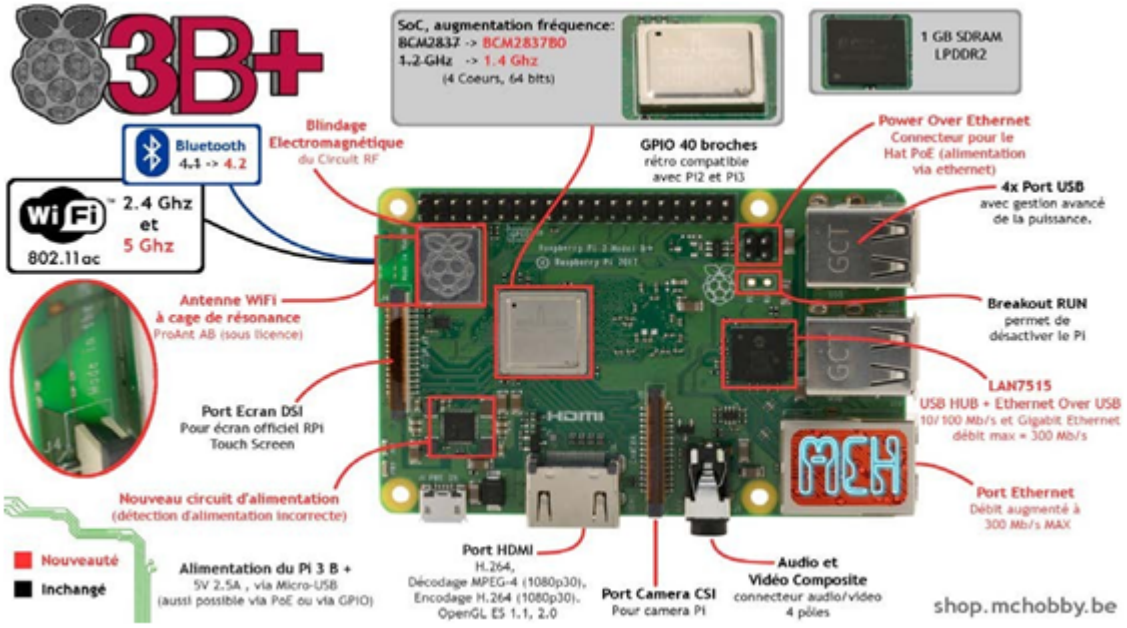


Figure 3.1: Raspberry pi et ses composant

fiche technique du Raspberry pi

- Alimentation : 5 Vcc/ maxi 2,5 A via prise micro-USB (intensité maxi si toutes les fonctions sontutilisées)
- CPU : ARM Cortex-A53 quatre cœurs 1,4 GHz 64bits
- Wifi: Dual-band 2,4 et 5 GHz, 802.11b/g/n/AC (BroadcomBCM43438)
- Bluetooth 4.2 (BroadcomBCM43438)

- Mémoire : 1 GBLPDDR2
- Ethernet 10/100/1000 : jusqu'à 300Mbps
- 4 ports USB2.0
- Port Ethernet 10/100 base T :RJ45
- Bus : SPI, I2C,série
- Support pour cartesmicro-SD
- Sorties audios:- Jack 3,5 mm en stéréo
- Sorties vidéo :HDMI
- Dimensions : 86 x 54 x 17 mm
- Poids : 50 g

Arduino

Définition Un module Arduino est généralement construit autour d'un microcontrôleur Atmel AVR (ATmega328, ATmega32u4 ou ATmega2560 pour les versions récentes, ATmega168, ATmega1280 ou ATmega8 pour les plus anciennes), et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles) [10].

Le microcontrôleur est préprogrammé avec un bootloader de façon qu'un programmateur dédié ne soit pas nécessaire.

Les modules sont programmés avec une connexion série TTL, mais les connexions permettant cette programmation diffèrent selon les modèles. Les premiers Arduino possédaient un port série RS-232, puis l'USB est apparu sur les modèles Diecimila, tandis que certains modules destinés à une utilisation portable comme le Lillypad ou le Pro-mini se sont affranchis de l'interface

de programmation, relocalisée sur un module USB-série dédié (sous forme de carte ou de câble), cela permettait aussi de réduire leur coût, le convertisseur USB-Série TTL (un FTDI232RL de FTDI) coûtant assez cher [10].

L'Arduino utilise la plupart des entrées/sorties du microcontrôleur pour l'interfaçage avec les autres circuits. Le modèle Diecimila par exemple, possède quatorze entrées/sorties numériques, dont six peuvent produire des signaux PWM, et 6 entrées analogiques. Les connexions sont établies au travers de connecteurs femelles HE14 situés sur le dessus de la carte, les modules d'extension venant s'empiler sur l'Arduino. Plusieurs sortes d'extensions sont disponibles dans le commerce [10].

D'autres cartes comme l'Arduino Nano ou le Pro micro utilisent des connecteurs mâles, permettant de les disposer sur une platine d'expérimentation.

La société STMicroelectronics a également travaillé avec Arduino, sur des cartes compatibles. Les cartes STM32 Nucleo, basées sur les processeurs STM32, utilisant l'architecture ARM plutôt que l'architecture Harvard des Atmel AVR. Ces cartes comportent un processeur plus puissant, ARM Cortex-M 32 bits, de M0+ à 32 MHz ou M0 à 48 MHz jusqu'au M4 à 100 MHz, comportant des instructions DSP et un processeur graphique Chrom-ART de STMicroelectronics [10].

Arduino mega Arduino Méga 2560 est une carte à microcontrôleur. Il a 54 numérique broches d'entrée / sortie, 16 entrées analogiques, une céramique de 16 MHz résonateur, une connexion USB, une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation(Figure 3.2) [11].

Figure 3.2: Arduino méga

Nous avons utiliser arduino pour commander le servomoteur à la place du raspberry pi pour plus de précision du signal PWM et les 4 moteurs à courant continu et aussi les 2 ultrason [11].

Picamera

C'est une caméra de 5 méga pixels spécialement pour le Raspberry pi elle se branche au port avec une nappe. Elle serre à prendre des photos et des vidéos jusqu'au 1080p a 30fps et 720p a 60fps et 480p a 60 et 90fps(Figure 3.3) [9].

Figure 3.3: Camera du raspberry pi

servomoteur

Le servo moteur est un moteur commandé qui peut assurer un angle ou placement, il contient un moteur et l'engrenage pour lui donner une force (couple), il a un système d'asservissement(Figure 3.4) [10].

Les caractéristiques du servo moteur sg90

- Dimensions : 22 x 11.5 x 27MM.
- Poids : 9gr.
- Tension d'alimentation : 4.8v à 6v.
- Vitesse : 0.12 s / 60° sous 4.8v.
- Couple : 1.2 Kg / cm sous 4.8v.
- Amplitude : de 0 à 180°.

Figure 3.4: Servomoteur sg90

Fonctionnement du servomoteur

Le signal envoyé au servomoteur correspond à une impulsion comprise entre 1 et 2ms toutes les 20 ms. La durée du signal correspond à un angle entre 0 et 180°. Ainsi 1ms correspondra à un angle de 0°, 1,5ms à 90° et 2ms à 180°(Figure 3.5) [10].

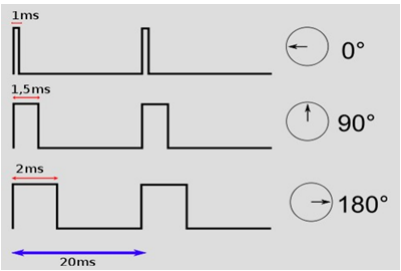


Figure 3.5: PWM générer par arduino méga

Moteur a courant continue

Les moteurs courant continu sont des convertisseurs de puissance, Soit ils convertissent l'énergie électrique absorbée en énergie mécanique lorsqu'ils sont capables de fournir une puissance mécanique suffisante pour démarrer puis entraîner une charge en mouvement. On dit alors qu'ils ont un fonctionnement en moteur. Soit ils convertissent l'énergie mécanique reçue en énergie électrique lorsqu'ils subissent l'action d'une charge entraînante. On dit alors qu'ils ont un fonctionnement engénérateur [10].

Dans notre projet nous avons utiliser deux moteur a courant continu(Figure 3.6).

Figure 3.6: Moteur a courant continue

Les capteur hc-sr04

C'est un capteur utilise les ultrasons pour calculer la distance avec une très grande pression, il est stable et efficace pour les petits projets(Figure 3.7) [10].

Figure 3.7: HC-SR0

Caractéristique

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

broche du Hc-sr04

- Vcc = Alimentation +5 V DC
- Trig = Entrée de déclenchement de la mesure (Triggerinput)
- Echo = Sortie de mesure donnée en écho (Echooutput)
- GND = Masse del'alimentation

Télécommande IR

Le signal est émis par une diode, petit composant électronique qui, au passage d'un courant électrique (environ 20 mA), produit une lumière infrarouge (bande du spectre invisible à l'œil humain et située au-dessous du rouge, de longueur d'onde (800-1 000 nm), de longueur d'onde d'environ 940 nm (0,94 μm)(Figure 3.8) [10].

Figure 3.8: Télécommande IR

Nous l'avons utiliser dans notre projet pour fair un marche/arret du robot.

3.2 Réalisation

3.2.1 La configuration du Raspberry pi

La configuration du Raspberry pi consiste à installer le système d'exploitation officiel RASBIAN et les bibliothèques nécessaires (open cv, numpy) [6].

RASBIAN : est un système d'exploitation très optimisé du Debian qui marche sur le Raspberry pi. On l'Installe sur une carte micro-SD, et après on la met au Raspberry pi [6].

Open cv (open computer vision) est une bibliothèque libre, initialement développé par Intel, spécialisé dans le traitement d'image en temps réel [6].

Numpy est une extension du langage de programmation python elle est spécialisé au calcule matricielle et les tableaux [6].

3.2.2 Liaison du Raspberry pi avec les autres composants

liaison avec la camera

La camera officielle du Raspberry pi est une caméra de 5 mégapixels brancher directement au port(Figure 3.9).

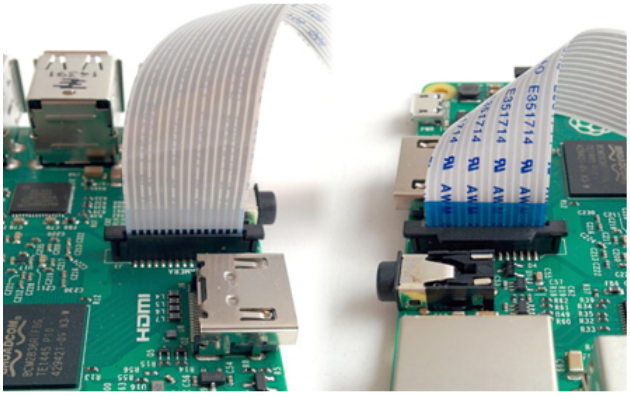


Figure 3.9: Emplacement de la camera

Après le branchement de la camera il faut l'activer depuis le bouton démarrer(Figure 3.10).

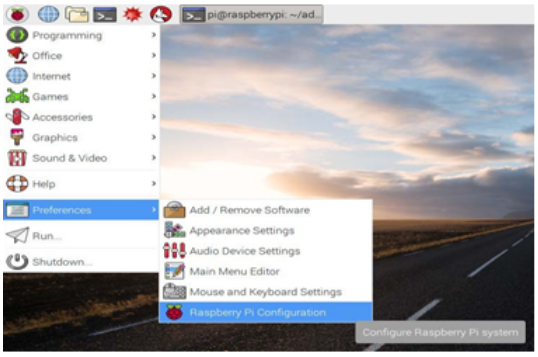


Figure 3.10: Bureau du raspberry pi

- Préférences
- Configuration de Raspberry pi
- Interfaces (Figure 3.11)

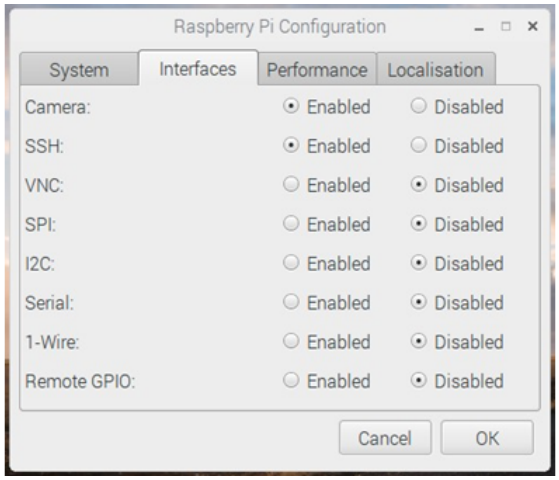


Figure 3.11: Configuration

Après l'activation nous activons SSH pour la comunucation a distanse, ensuite cliquons sur Enable et validé. le Raspberry pi va redémarrer. Finalement la camera et la comunication sont bien configuré .

liaison avec arduino

Dans certains projets, il peut être intéressant d'établir une communication série entre Raspberry Pi et Arduino. Il est ainsi possible de coupler la puissance de calcul et les interface sans fil du Raspberry Pi avec les entrées-sorties et la collection de modules Arduino. Le premier exemple qui vient en tête est l'utilisation de ce système pour de la domotique dans lequel le Raspberry Pi va héberger l'interface de contrôle et l'intelligence et les Arduino vont servir d'automate programmable agissant sur les composants en bout de chaîne (lumière, radiateur, ventilateur, capteurs, etc.) [12].

Maintenant nous allons voir comment mettre en place une communication série entre Raspberry Pi et Arduino via le port USB.

Pour établir la communication série entre Raspberry Pi et Arduino, il suffit des les relier grâce à un câble USB adapté. Dans notre cas, nous utilisons un Raspberry Pi 3B+ et un Arduino MEGA. Il nous faut donc un cable USBA Mâle vers USB B Mâle(Figure 3.12) [12].

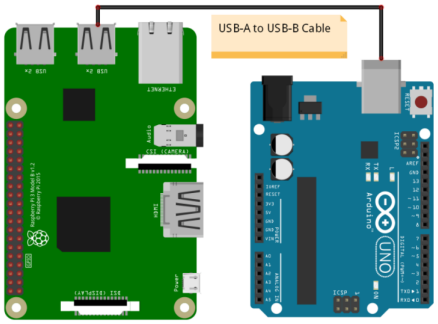


Figure 3.12: Communocation raspberry pi et arduino

Configuration du Raspberry Pi : Pour utiliser l'interface série du Raspberry Pi, celle-ci doit être activée dans le menu de configuration. Pour cela entrez la commande suivante dans un terminal [12]:

```
$sudo apt-get update
```

Dans le menu, sélectionnez "5 – Interfacing Options" puis "P6 Serial" et validez.

Un fois le branchement fait, vous pouvez vérifier les appareils branchés sur le port série en tapant dans le terminal la commande:

```
$lsusb
```

Le Raspberry Pi retourne la liste des appareils branchés sur les ports USB(Figure 3.13).

```
pi@raspberrypi:~ $ lsusb
Bus 001 Device 006: ID 062a:4102 MosArt Semiconductor Corp.
Bus 001 Device 004: ID 2a7a:938f
Bus 001 Device 005: ID 2341:0042 Arduino SA Mega 2560 R3 (CDC ACM)
Bus 001 Device 007: ID 0424:7800 Standard Microsystems Corp.
Bus 001 Device 003: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Figure 3.13: Liste des appareils branchés sur les ports USB

Pour trouver le nom du port sur lequel est branché l'Arduino, nous utilisons la commande [13]:

```
$dmesg | grep "tty"
```

Cette commande retourne les message système relatifs aux port séries. Vous deviez trouver le nom du port dans les derniers message. Dans notre cas le nom du port est ttyACM0(Figure 3.14).

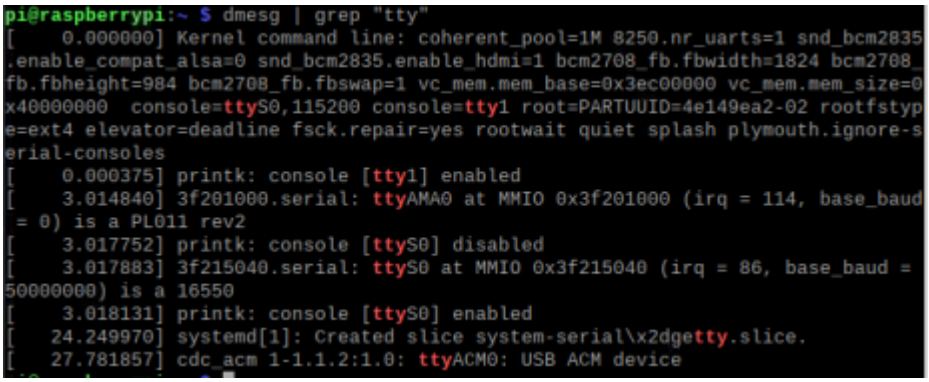


Figure 3.14: Les message système relatifs aux port séries

Installation de l'IDE Arduino sur Raspberry Pi Pour installer l'IDE Arduino sur Raspberry Pi, le mieux est de passer par le terminal. Il vous suffit de rentrer les lignes de code suivantes [13]:

```
$mkdir ~/Applications
$cd ~/Applications
$wget https://downloads.arduino.cc/arduino-1.8.9-linuxarm.tar.xz
$tar xvJf arduino-1.8.9-linuxarm.tar.xz
$cd arduino-1.8.9/
  ./install.sh
$rm ../arduino-1.8.9-linuxarm.tar.xz
```

Ceci vous permettra, par la suite, de programmer l'Arduino directement à partir du Raspberry Pi.

Exemple

Code Arduino La librairie utilisée pour la communication série côté Arduino est la même que pour communiquer avec le moniteur série, la librairie Serial.h que nous connaissons bien. Vérifiez bien que la vitesse de communication est la même pour les deux appareils (baudrate=9600) sinon la communication ne fonctionnera pas [13].

```
String nom = "Arduino";
String msg;

void setup() {
  Serial.begin(9600);
}

void loop() {
  readSerialPort();

  if (msg != "") {
    sendData();
  }
  delay(500);
}

void readSerialPort() {
  msg = "";
  if (Serial.available()) {
    delay(10);
    while (Serial.available() > 0) {
```

```
      msg += (char)Serial.read();
    }
    Serial.flush();
  }
}

void sendData() {
  //write data
  Serial.print(nom);
```

Code Python Dans ce tutoriel, nous allons utiliser le langage Python côté Raspberry Pi. La librairie utilisée pour la gestion de la communication série est la librairie serial.

```
import serial,time


if __name__ == '__main__':

    print('Running. Press CTRL-C to exit.')
    with serial.Serial("/dev/ttyACM0", 9600, timeout=1) as arduino:
        time.sleep(0.1) #wait for serial to open
        if arduino.isOpen():
            print("{} connected!".format(arduino.port))
            try:
                while True:
                    cmd=input("Enter command : ")
                    arduino.write(cmd.encode())
```

```
                    #time.sleep(0.1) #wait for arduino to answer
                    while arduino.inWaiting()==0: pass
                    if  arduino.inWaiting()>0:
                        answer=arduino.readline()
                        print(answer)
                        arduino.flushInput() #remove data after
                           reading
            except KeyboardInterrupt:
                print("KeyboardInterrupt has been caught.")
```

Résultat Le Raspberry Pi envoie la commande « Hello Arduino » à l'Arduino, et l'Arduino répond avec son nom et la commande reçu(Figure 3.15).

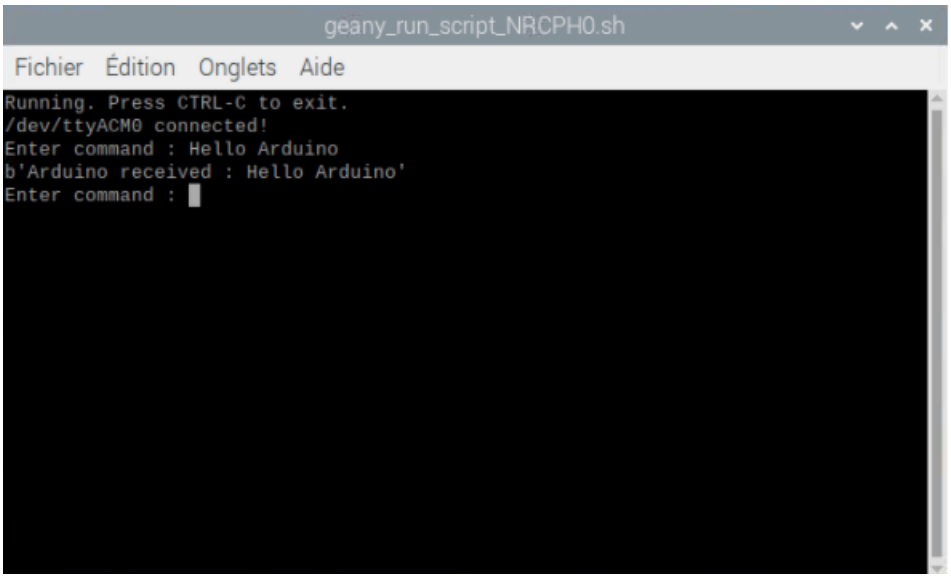


Figure 3.15: Résultat du l'exécution dans raspberry pi

Exemple pratique

Code Arduino

```
const int ledPin=13;
String nom = "Arduino";
String msg;

void setup() {
  Serial.begin(9600);
}

void loop() {
  readSerialPort();

  if (msg == "data") {
    sendData();
  }else if(msg=="led0"){
    digitalWrite(ledPin,LOW);
    Serial.print(" Arduino set led to LOW");
  }else if(msg=="led1"){
    digitalWrite(ledPin,HIGH);
    Serial.print(" Arduino set led to HIGH");
  }
  delay(500);
}

void readSerialPort() {
```

```
  msg = "";
  if (Serial.available()) {
    delay(10);
    while (Serial.available() > 0) {
      msg += (char)Serial.read();
    }
```

Code Python

```
import serial,time



if __name__ == '__main__':
    print('Running. Press CTRL-C to exit.')
    with serial.Serial("/dev/ttyACM0", 9600, timeout=1) as arduino:
        time.sleep(0.1) #wait for serial to open
        if arduino.isOpen():
            print("{} connected!".format(arduino.port))
            try:
                while True:
                    cmd=input("Enter command (data,led0 or led1): "
                       )
                    arduino.write(cmd.encode())
                    #time.sleep(0.1) #wait for arduino to answer

                    while arduino.inWaiting()==0: pass
                    if  arduino.inWaiting()>0:
```

```
                    answer=str(arduino.readline())
                    print("---> {}".format(answer))
                    if cmd=="data":
                        dataList=answer.split("x")
                        print("led state : {}".format(dataList
                            [0]))
                        print("Analog input A0 : {}".format(
                            dataList[1]))
                        print("Analog input A1: {}".format(
                            dataList[2]))

                        arduino.flushInput() #remove data after
                             reading

            except KeyboardInterrupt:
                print("KeyboardInterrupt has been caught.")
```

Résultat Une fois les deux codes téléversés et lancés, on observe que lorsqu'on entre la commande « data » dans le terminal, Arduino renvoie bien des bytes contenant les valeurs de capteurs. Il est possible de séparer cette réponse en liste à l'aide de la fonction split() et du caractère « x » et, ainsi, récupérer les valeurs des capteurs et l'état de la led. Grâce à ce code, il nous est possible de piloter l'état de la LED de la broche 13(Figure 3.16) [13].

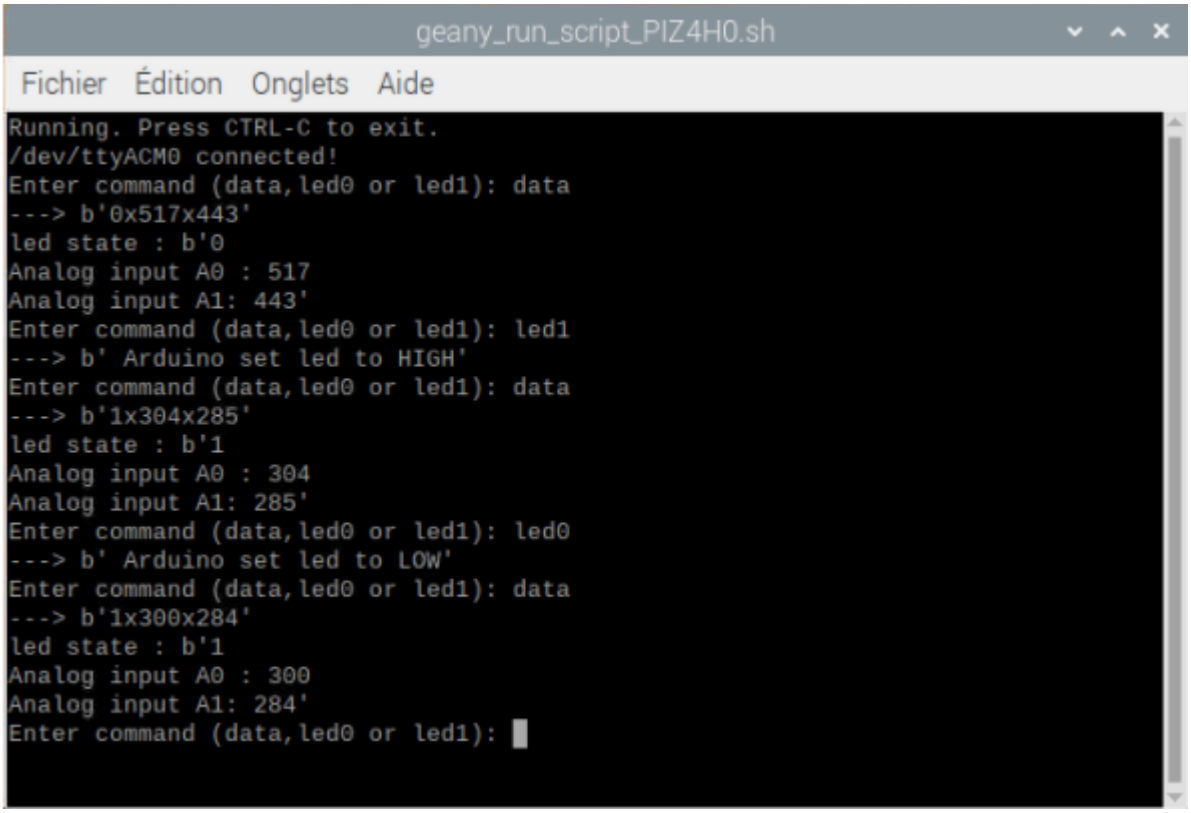


Figure 3.16: Résultat de la communication série entre le raspberry pi et l'arduino

3.2.3 Liaison du Arduino avec les autres composants

liaison avec les moteur DC et le servomoteur Nous allons connecter les moteurs et le servomoteur avec l'ARDUINO à l'aide un shield moteur L293D(Figure 3.17) [10].

Caractéristiques : Il posède plusieurs avantages par rapport au autres Shield [10]:

- Deux interfaces pour servomoteurs 5V, connectés au timer haute résolution de l'Arduino.
- Peut piloter 4 moteurs à courant continu DC, ou 2 moteurs pas à pas, ou 2 servo à la fois.
- Jusqu'à 4 moteurs DC bi-directionnels avec sélection de la vitesse individuelle (sur 8 bit).
- Jusqu'à 2 moteurs pas à pas (unipolaire ou bipolaire) avec une seule bobine, double bobine, ou demi pas.
- 4 pont en H (H-Bridges).

- Fournit 0,6 A par pont (1.2A en courant de crête) avec protection thermique.
- Pilotage des moteurs à courant continu de 4.5V à 36V.
- Des résistances pull down désactivent les moteurs au cours de la mise sous tension.
- Bouton de réinitialisation (Reset).
- 2 interfaces d'alimentation pour séparer la partie logique de la partie puissance (moteurs).
- Compatible avec les cartes Arduino Mega, Diecimila et Duemilanove .

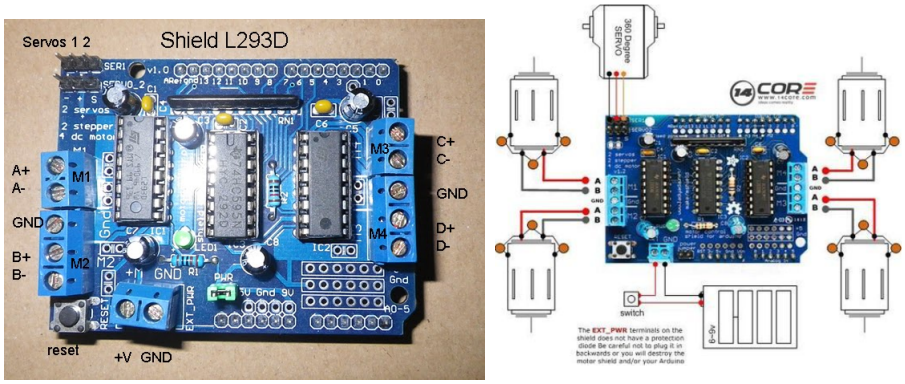


Figure 3.17: Shield L293D

Maitenant nous allons connecter l293d avec notre ARDUINO comme ce ci (Figure 3.18):

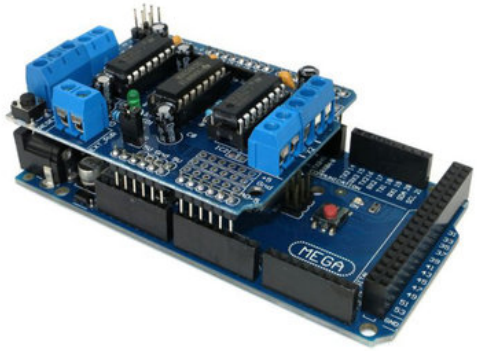


Figure 3.18: Connexion du Shield L293D avec Arduino

Liaison avec HC-SR04 : Les ultrasons de type hc-sr04 contient 4 ports[10]:

- Vcc
- Trigo
- Echo
- Gnd

Le câblage avec l'ARDUINO est comme suit (Figure 3.19):

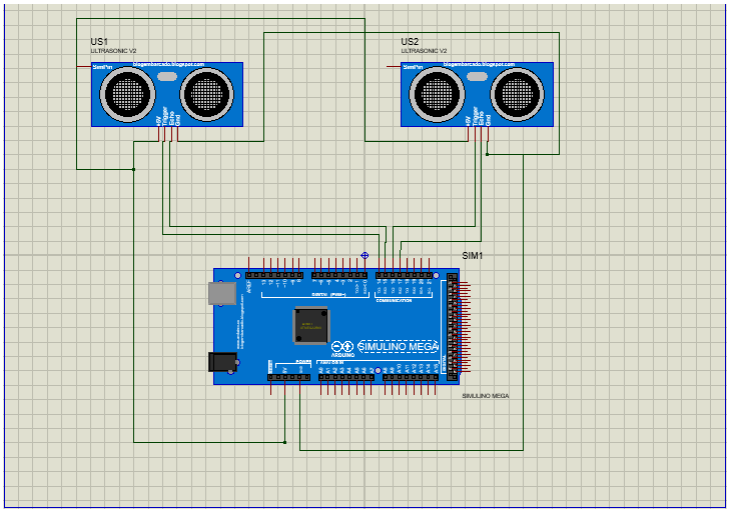


Figure 3.19: Cablage d'ultrason avec l'Arduino

3.3 Programme

Pour programmer notre robot nous allons utiliser IDE Arduino pour la carte arduino, et Python pour le raspberry pi.

3.3.1 IDE Arduino:

Les créateurs de Arduino ont développé un logiciel pour que la programmation des cartes arduino soit visuelle, simple et complète à la fois.

C'est ce que l'on appelle une IDE, qui signifie Integrated Development Environment ou Environnement de Développement "Intégré" en français (donc EDI).

L'IDE Arduino est le logiciel qui permet de programmer les cartes Arduino [11].

L'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation. Vous pouvez donc saisir votre programme, l'enregistrer, le compiler, le vérifier, le transférer sur une carte arduino... [11]

3.3.2 Python:

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages [14].

3.3.3 programme arduino

Dans cette partie nous allons présenter les codes des composantes utilisés avec arduino.

Tout d'abord nous allons commencer par la télécommande IR, voici son programme:

```
#include <IRremote.h>

const int receiver = 14;
IRrecv irrecv(receiver);
decode_results results;

int s;
void translateIR(){
  switch(results.value){
    case 0xFD00FF : s = 1; break;
    case 0xFD807F : s = 2; break;
    }
  }

void setup() {
  irrecv.enableIRIn();
}

void loop() {

  if (irrecv.decode(&results))
  {
  translateIR();
```

```
    delay(200);
    irrecv.resume();
    }
}
```

ensuite le code pour pouvoire détecté les obstacles par les deux capteurs utlrason est:

```
#include "LiquidCrystal.h"

const int trigPinG = 15; //trig pin connection
const int echoPinG = 16;  //echopin connection

const int trigPinD = 18; //trig pin connection
const int echoPinD = 17;  //echopin connection

long durationG;
long distanceG;
long g;

long durationD;
long distanceD;
long d;

void setup() {
pinMode(trigPinG, OUTPUT);
pinMode(echoPinG, INPUT);

pinMode(trigPinD, OUTPUT);
```

```
pinMode(echoPinD, INPUT);
}

void loop() {
  digitalWrite(trigPinG, LOW);
delayMicroseconds(2);
digitalWrite(trigPinG, HIGH);
delayMicroseconds(10);
digitalWrite(trigPinG, LOW);

durationG = pulseIn(echoPinG, HIGH);
distanceG = (durationG*0.034)/2;
g = abs(distanceG);

digitalWrite(trigPinD, LOW);
delayMicroseconds(2);
digitalWrite(trigPinD, HIGH);
delayMicroseconds(10);
digitalWrite(trigPinD, LOW);

durationD = pulseIn(echoPinD, HIGH);
distanceD = (durationD*0.034)/2;
d = abs(distanceD);
}
```

Maintenant il y a un tas d'instruction pour voir comment les servos moteurs li les instruction fourni par le raspberry pi, et compare la position de la camera par la position du visage, et faire le suivi du visage :

```
#include<Servo.h>


Servo x;
Servo y;
int width = 640, height = 480;  // total resolution of the video
int xpos = 90;  // initial positions of both Servos
int ypos = 90;

const int angle = 1;   // degree of increment or decrement

void setup() {

Serial.begin(4000000);
  x.attach(9);
  x.write(xpos);

  y.attach(10);
  y.write(ypos);

}

void loop() {

if (Serial.available() > 0)
```

```
{
  int x_mid, y_mid;
  if (Serial.read() == 'X')
  {
    x_mid = Serial.parseInt();  // read center x-coordinate
    if (Serial.read() == 'Y')
      y_mid = Serial.parseInt();
    }
    if (x_mid > width / 2 + 30)
    xpos += angle;
    if (x_mid < width / 2 - 30)
    xpos -= angle;

  if (y_mid < height / 2 + 30)
    ypos -= angle;
  if (y_mid > height / 2 - 30)
    ypos += angle;

    // if the servo degree is outside its range
  if (xpos >= 180)
    xpos = 180;
  else if (xpos <= 0)
    xpos = 0;

   if (ypos >= 180)
    ypos = 180;
  else if (ypos <= 0)
```

```
        ypos = 0;


        x.write(xpos);
        y.write(ypos);
    }


}
```

Maintenant ils nous restes comment faire bouger le robot pour suivre le visage,donc il faut faire ça a l'aide des moteurs a courant continu.

Tous d'abord il faut inclure la bibliothèque AFMotor.h pour commandé le Shield L293D, et déclaré les deux moteurs .

Pour facilité et bien organisé notre code, il est nécessaire de déclaré des fonction pour définir les direction (gauche,droite,avant,arrière):

```
#include <AFMotor.h>


AF_DCMotor motor1(3);
AF_DCMotor motor2(4);


void avant(){


   motor1.run(FORWARD);
   motor2.run(FORWARD);
              }


void arriere(){
```

```
    motor1.run(BACKWARD);
    motor2.run(BACKWARD);
                }

void turnleft(){

    motor1.run(RELEASE);
    motor2.run(FORWARD);
                }

void turnright(){

    motor1.run(FORWARD);
    motor2.run(RELEASE);
                }

void allstop(){

    motor1.run(RELEASE);
    motor2.run(RELEASE);
            }

void setup() {

    motor1.setSpeed(255);
    motor2.setSpeed(255);
}
```

Enfin toute est pré pour pouvoir marché notre robot de tel sorte qu'il va suivre le visage, et évité tout les obstacles sur son chemin :

```
void loop() {


if(s == 1){
  if(g>10 && d>10){


  if ((xpos >= 80 ) & (xpos <= 100)) {
   avant();
  }
  else if ((xpos >= 100 ) & (xpos <= 180)) {
    turnleft();
    delay(10);
    allstop();
  }
  else if ((xpos >= 1 ) & (xpos <= 80)) {
    turnright();
    delay(10);
    allstop();
  }
  }


else{
  if(g<10 && d>10){
 turnright();
 delay(2000);
 turnleft();
```

```
 delay(2000);
  }


  if(g>10 && d<10){
 turnleft();
 delay(2000);
 turnright();
 delay(2000);
  }
    }
    }
    else{if(s == 2){
      allstop();
    }
    }

}
```

3.3.4 programme python

Dans cette partie nous allons voir comment détecté le visage, et comment envoyer à l'arduino la position de la caméra et la coordonnée du visage qu'il va le suivre.

Tout d'abord il faut importer les librairies nécessaires :

```
import cv2

import numpy as np
```

```
import serial,time
```

Maintenant nous allons dire à python que nous souhaitons utiliser la caméra du raspberry pi :

```


cap = cv2.VideoCapture(0)
```

Pour détecter les visage nous allons utiliser du machine learning, pour ce la nous allons utiliser un fichier xml de la détection le visage:

```
face_cascade = cv2.CascadeClassifier("
   haarcascade_frontalface_default.xml")
```

La fonction CascadeClassifier permet de lire notre fichier machine learning xml.

Après cette étape nous commençont à envoyé les donnés à l'arduino par la commande suivante :

```
ArduinoSerial=serial.Serial('/dev/ttyACM0',4000000,timeout=0.1)
```

On effectue un traitement d'images avant de passer à la détection des visages.

En OpenCV nous utilisons une boucle while pour les flux vidéo afin de lire une image après l'autre. Après tout, une vidéo n'est qu'une suite d'images.

```
while True:

    cap.set(cv2.CAP_PROP_FRAME_WIDTH,640);

    cap.set(cv2.CAP_PROP_FRAME_HEIGHT,480);

    ret, frame = cap.read()

    frame = cv2.flip(frame, 1)

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Avec notre fichier haar cascade nous allons détecter les visages:

```
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
```

La fonction detectMultiScale renvoie une liste de rectangles où un visage a été détecté.

Le paramètre gray correspond à notre image du raspberry et en niveaux de gris.

Le paramètre 1.3 correspond au facteur d'échelle. Il spécifie dans quelle mesure la taille de l'image est réduite à chaque échelle d'image. Ici, on lui donne la valeur 1.3 mais vous pouvez modifier pour voir ce que ça donne.

Enfin, le dernier argument donne le nombre minimum de voisins que doit avoir un rectangle pour être accepté. Ici, on lui donne la valeur 5 mais encore une fois vous pouvez modifier ce paramètre pour voir ce que ça donne.

Ici je renote les instructions précédentes de la boucle while, mais il s'agit bien de la même boucle. Donc inutile d'en mettre une nouvelle.

```
  for x,y,w,h in faces:
```

Envoi de coordonnées à Arduino :

```
    string='X{0:d}Y{1:d}'.format((x+w//2),(y+h//2))

    print(string)

        ArduinoSerial.write(string.encode('utf-8'))
```

Tracer le centre du visage :

```
    cv2.circle(frame,(x+w//2,y+h//2),2,(0,75,0),2)
```

Tracer la région au carré au centre de l'écran :

```
    cv2.rectangle(frame,(640//2-30,480//2-30),(640//2+30,480//2+30)
       ,(75,75,75),3)
```

Pour chaque rectangle de la liste visage, nous tracons le rectangle sur l'image de la camera (celle en couleur).

```
    cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,75),3)
```

La fonction rectangle trace un rectangle sur image en coordonnées (x,y) et le point opposé en coordonnées (x+w, y+h).

```
    cv2.imshow("Frame", frame)
```

cv2.imshow pour afficher le résultat final.

```
    key = cv2.waitKey(1)
```

On utilise key pour fermer le programme.

```
cap.release()
cv2.destroyAllWindows()
```

La fonction destroyAllWindow permet de fermer toutes les fenêtres.

Conclusion

Dans ce chapitre nous avons présenter le matériel utilisé (arduino, raspberry pi ...) dans notre projet et définir chaque outil, et utilisé une communication série entre l'arduino et le raspberry pi à l'aide d'un cable USB, et après plusieurs test sur le robot nous avons donner le résultat obtenu après la réalisation pratique du projet de notre thème.

Conclusion générale

De plus en plus, les technologies connectées et la conduite autonome permettront aux usagers de faire un choix entre conduire et être conduits. Ce constat ouvre la voie à de nouveaux scénarios de mobilité.

Ce projet couvre un grand nombre de domaines :

- La partie hardware : nous avons utiliser une carte de commande Arduino méga, qui se caractérise essentiellement par la programmation directe par un PC , pour la commande de deux moteurs DC 12V, ces composants sont monté sur le châssis du robot.
- La partie software : nous avons utiliser le langage Python, et la librairie open CV pour la réalisation de notre programme après la configuration de ces deux logiciels et des autres modules annexes.

Ce travail exige la maîtrise de plusieurs notions techniques aussi bien le domaine électronique et celui de l'informatique, ce que fait que nous avons eu l'occasion à découvrir, la programmation par La carte Arduino méga, Langage PYTHON, et la librairie Open CV.

Notre projet consiste a la conception d'un module de navigation intelligent, après le phase pratique et tout les essai nous avons pu améliorer la performance de notre application et l'objectif atteint.

Comme perspectives, nous pouvons améliorer le coté software (programme) en ajoutant des autres conditions à notre voiture comme l'optimisation des chemins en implémentant des

algorithmes du recherche du chemin plus cours, développer des AGV (Auto Guided Vehicule) à base de notre modèle du robot.

Annexe

Code arduino

```
#include <AFMotor.h>
#include <Servo.h>
#include <IRremote.h>
#include "LiquidCrystal.h"

AF_DCMotor motor1(3);
AF_DCMotor motor2(4);

Servo x;
Servo y;
int width = 640, height = 480;  // total resolution of the video
int xpos = 90;  // initial positions of both Servos
int ypos = 90;
const int angle = 1;   // degree of increment or decrement

const int trigPinG = 15; //trig pin connection
const int echoPinG = 16;  //echopin connection
```

```
const int trigPinD = 18; //trig pin connection
const int echoPinD = 17;  //echopin connection

long durationG;
long distanceG;
long g;

long durationD;
long distanceD;
long d;

const int receiver = 14;
IRrecv irrecv(receiver);
decode_results results;
int s;
void translateIR(){
  switch(results.value){
    case 0xFD00FF : s = 1; break;
    case 0xFD807F : s = 2; break;
    }
  }

void avant(){

   motor1.run(FORWARD);
   motor2.run(FORWARD);
```

```
            }

void arriere(){

   motor1.run(BACKWARD);
   motor2.run(BACKWARD);
            }

void turnleft(){

   motor1.run(RELEASE);
   motor2.run(FORWARD);
             }

void turnright(){

   motor1.run(FORWARD);
   motor2.run(RELEASE);
              }

void allstop(){

   motor1.run(RELEASE);
   motor2.run(RELEASE);
          }

void setup() {
```

```
   motor1.setSpeed(255);
   motor2.setSpeed(255);


pinMode(trigPinG, OUTPUT);
pinMode(echoPinG, INPUT);

pinMode(trigPinD, OUTPUT);
pinMode(echoPinD, INPUT);

  Serial.begin(4000000);

  x.attach(9);
  x.write(xpos);

  y.attach(10);
  y.write(ypos);

 irrecv.enableIRIn();
}

void loop() {

if (Serial.available() > 0)
  {
    int x_mid, y_mid;
```

```
if (Serial.read() == 'X')
{
  x_mid = Serial.parseInt();  // read center x-coordinate
  if (Serial.read() == 'Y')
    y_mid = Serial.parseInt();
  }
  if (x_mid > width / 2 + 30)
  xpos += angle;
  if (x_mid < width / 2 - 30)
  xpos -= angle;

if (y_mid < height / 2 + 30)
  ypos -= angle;
if (y_mid > height / 2 - 30)
  ypos += angle;

  // if the servo degree is outside its range
if (xpos >= 180)
  xpos = 180;
else if (xpos <= 0)
  xpos = 0;

 if (ypos >= 180)
  ypos = 180;
else if (ypos <= 0)
  ypos = 0;
```

```
      x.write(xpos);
      y.write(ypos);
  }

digitalWrite(trigPinG, LOW);
delayMicroseconds(2);
digitalWrite(trigPinG, HIGH);
delayMicroseconds(10);
digitalWrite(trigPinG, LOW);

durationG = pulseIn(echoPinG, HIGH);
distanceG = (durationG*0.034)/2;
g = abs(distanceG);

digitalWrite(trigPinD, LOW);
delayMicroseconds(2);
digitalWrite(trigPinD, HIGH);
delayMicroseconds(10);
digitalWrite(trigPinD, LOW);

durationD = pulseIn(echoPinD, HIGH);
distanceD = (durationD*0.034)/2;
d = abs(distanceD);

 if (irrecv.decode(&results))
  {
  translateIR();
```

```
    delay(200);
    irrecv.resume();
    }

  if(s == 1){
    if(g>10 && d>10){

    if ((xpos >= 80 ) & (xpos <= 100)) {
     avant();
    }
    else if ((xpos >= 100 ) & (xpos <= 180)) {
      turnleft();
      delay(10);
      allstop();
    }
    else if ((xpos >= 1 ) & (xpos <= 80)) {
      turnright();
      delay(10);
      allstop();
    }
    }

  else{
    if(g<10 && d>10){
   turnright();
   delay(2000);
   turnleft();
```

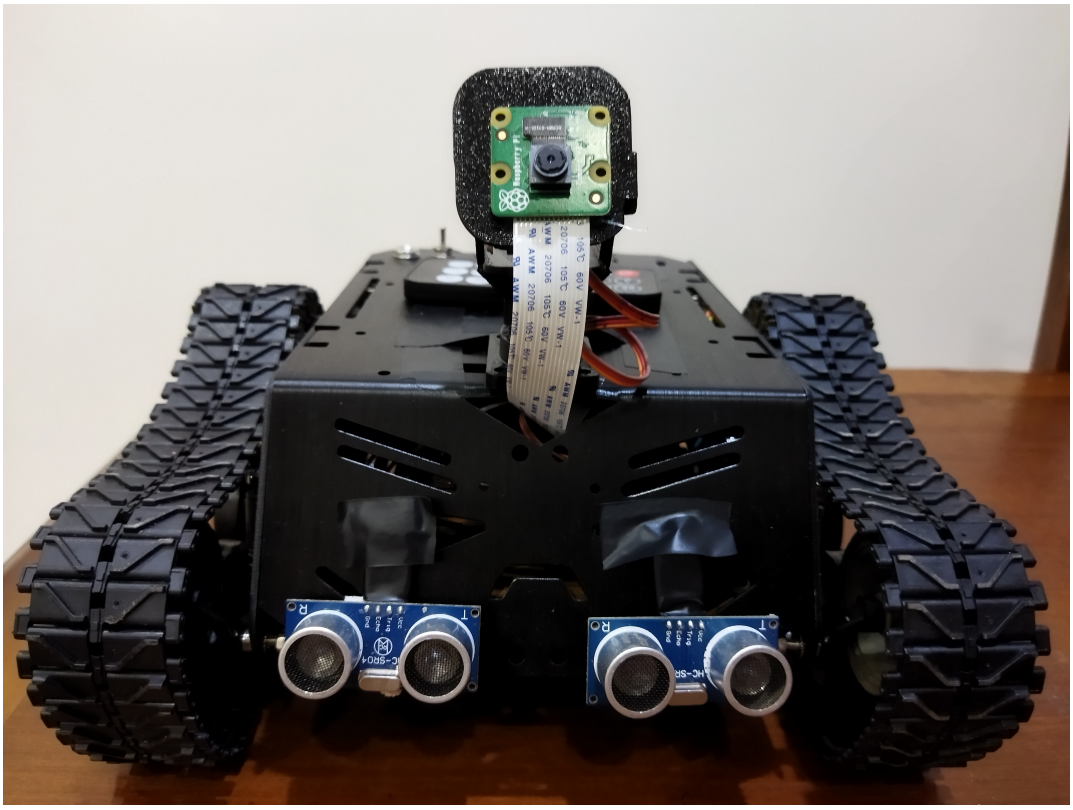
```
 delay(2000);
  }


  if(g>10 && d<10){
 turnleft();
 delay(2000);
 turnright();
 delay(2000);
  }
    }
    }
    else{if(s == 2){
      allstop();
    }
    }
}
```

Code python

```
import cv2
import numpy as np
import serial,time

def nothing(x):
    pass

cap = cv2.VideoCapture(0)

face_cascade = cv2.CascadeClassifier("
   haarcascade_frontalface_default.xml")

cv2.namedWindow("Frame")
cv2.createTrackbar("Neighbours", "Frame", 5, 20, nothing)
ArduinoSerial=serial.Serial('/dev/ttyACM0',4000000,timeout=0.1)
time.sleep(1)

while True:
    cap.set(cv2.CAP_PROP_FRAME_WIDTH,640);
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT,480);
    ret, frame = cap.read()

    frame = cv2.flip(frame, 1)  # mirror the image
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
        faces = face_cascade.detectMultiScale(gray, 1.3)
        for x,y,w,h in faces:
            #sending coordinates to Arduino
            string='X{0:d}Y{1:d}'.format((x+w//2),(y+h//2))
            print(string)
            ArduinoSerial.write(string.encode('utf-8'))
            #plot the center of the face
            cv2.circle(frame,(x+w//2,y+h//2),2,(0,75,0),2)
            #plot the roi
            cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,75),3)
            #plot the squared region in the center of the screen
            cv2.rectangle(frame,(640//2-30,480//2-30),
                          (640//2+30,480//2+30),
                           (75,75,75),3)

        cv2.imshow("Frame", frame)

        key = cv2.waitKey(1)
        if key == 27:
            break
    cap.release()
    cv2.destroyAllWindows()
```

Le robot



Reférences

[1] D. Le JURY, "Intitulé: Conception d'un module de navigation intelligent," Ph.D. dissertation, UNIVERSITE BADJI MOKHTAR ANNABA, 2019.

[2] G. Abdellaoui and F. T. Bendimerad, "Dynamic reconfiguration of lpwans pervasive system using multi-agent approach," *Int. J. Adv. Comput. Sci. Appl*, vol. 9, pp. 300–305, 2018.

[3] G. ABDELLAOUI, H. MEGNAFI, and F. T. BENDIMERAD, "A novel model using reo for iot self-configuration systems," in *2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP)*. IEEE, 2020, pp. 1–5.

[4] D. Boukhlouf, "Résolution de problèmes par écosystèmes: Application au traitement d'images," Ph.D. dissertation, Université Mohamed Khider-Biskra, 2005.

[5] G. Bradski and A. Kaehler, "Opencv," *Dr. Dobb's journal of software tools*, vol. 3, 2000.

[6] ——, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[7] O. Akinwole, "Design and simulation of a 1kva arduino microcontroller based modified sine wave inverter using proteus," *Journal of Electrical & Electronics Systems*, vol. 7, no. 4, p. 1, 2018.

[8] S. Motahhir, A. Chalh, A. Ghzizal, S. Sebti, and A. Derouich, "Modeling of photovoltaic panel by using proteus," *Journal of Engineering Science and Technology Review*, vol. 10, pp. 8–13, 2017.

[9] R. Org, "Raspberry pi 3 model b," *Feb-2016.[Online]. Available: https://www. raspberrypi. org/products/raspberry-pi-3-model-b/.[Accessed: 02-Jun-2017]*, 2016.

[10] P. Kumar and U. C. Pati, "Arduino and raspberry pi based smart communication and control of home appliance system," in *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*. IEEE, 2016, pp. 1–6.

[11] Y. Bailly, *Initiation à la programmation avec Python et C++*. Pearson Education France, 2008.

[12] N. Agrawal and S. Singhal, "Smart drip irrigation system using raspberry pi and arduino," in *International Conference on Computing, Communication & Automation*. IEEE, 2015, pp. 928–932.

[13] S. Ferdoush and X. Li, "Wireless sensor network system design using raspberry pi and arduino for environmental monitoring applications," *Procedia Computer Science*, vol. 34, pp. 103–110, 2014.

[14] G. Van Rossum, B. Warsaw, and N. Coghlan, "Pep 8: style guide for python code," *Python. org*, vol. 1565, 2001.

[15] S. Cox, "Steps to make raspberry pi supercomputer," *University of Southampton*, 2013.

[16] A. A. Goloborodko, L. I. Levitsky, M. V. Ivanov, and M. V. Gorshkov, "Pyteomics—a python framework for exploratory data analysis and rapid software prototyping in proteomics," *Journal of The American Society for Mass Spectrometry*, vol. 24, no. 2, pp. 301–304, 2013.

[17] G. Lindstrom, "Programming with python," *IT Professional Magazine*, vol. 7, no. 5, p. 10, 2005.

[18] D. Kushner, "The making of arduino," *IEEE spectrum*, vol. 26, 2011.

[19] L. Louis, "working principle of arduino and u sing it," *International Journal of Control, Automation, Communication and Systems (IJCACS)*, vol. 1, no. 2, pp. 21–29, 2016.

[20] A. Holovatyy, V. Teslyuk, M. Lobur, S. Pobereyko, and Y. Sokolovsky, "Development of arduino-based embedded system for detection of toxic gases in air," in *2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, vol. 1. IEEE, 2018, pp. 139–142.

[21] P. Anzhelika, G. Olga, E. Ivanov, A. Sokolyanskii, and S. Kurson, "Development and application of remote laboratory for embedded systems design," in *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*. IEEE, 2015, pp. 69–73.

[22] M. Micheloud and M. Rieder, *Programmation orientée objets en C++: une approche évolutive*. PPUR presses polytechniques, 2003.