

1. Présentation du projet

Dans ce projet, les étudiants sont répartis en binôme ou trinôme dans l'objectif de réaliser un robot mobile à roues. Les différents robots effectuent chacun une tâche différente (suivre une ligne, se déplacer dans un labyrinthe, trier des objets, arroser ...) ou sont pilotés au moyen d'une technologie distincte (télécommande Bluetooth ou infrarouge, commande par bruits et gestes ...). Tous les robots sont construits à partir d'un châssis identique (à roues ou à chenilles) ; le mode de déplacement est donc le même quel que soit le projet. En conséquence, la réalisation du robot comporte une partie commune à tous les groupes puis une partie spécifique à chaque groupe.

Pour exposer la démarche de réalisation et les matériels et logiciels associés, on présente ci-après l'exemple du robot suiveur de ligne à roues qui doit respecter le cahier des charges suivant :

En mode automatique, le robot, autonome en énergie, doit être capable de suivre une ligne blanche ou noire d'une largeur d'environ 10 mm. Il doit pouvoir communiquer des informations basiques quant à son fonctionnement et être commandable (marche, arrêt ...). En mode de mise au point, on doit pouvoir le piloter directement en marche avant ou arrière en ligne droite.

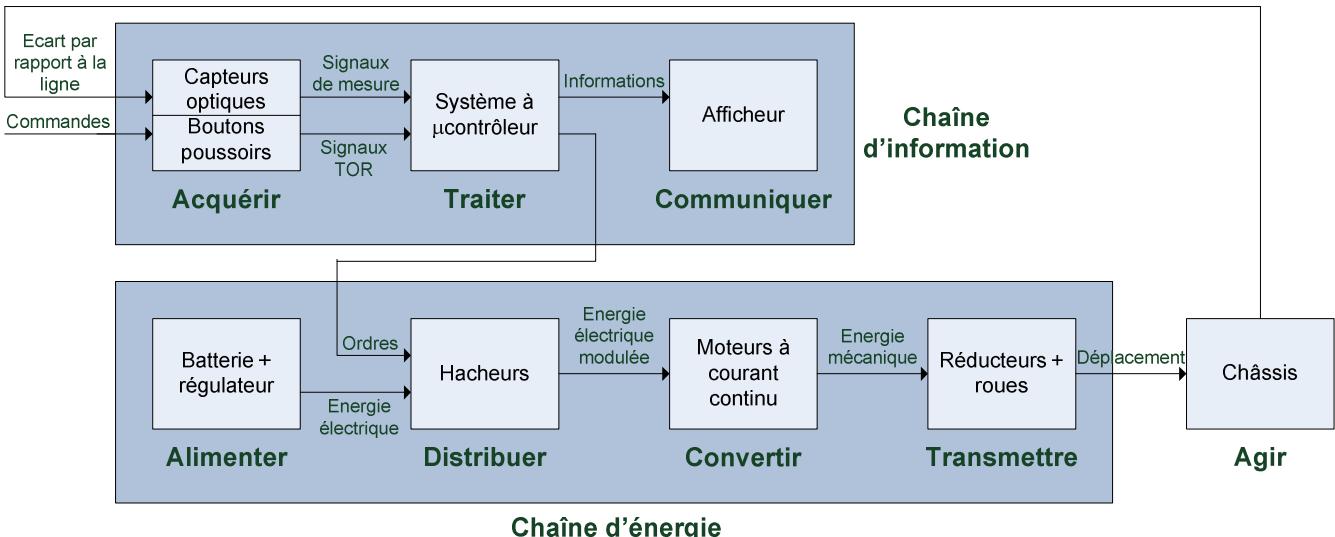
2. Analyse fonctionnelle et schéma structurel simplifié

A partir de l'énoncé ci-dessus, la première étape du projet consiste à réaliser l'analyse fonctionnelle du robot et à en déduire un schéma structurel simplifié qui sera progressivement enrichi.

Considérons tout d'abord la chaîne d'énergie. En tout premier lieu, il faut une source d'énergie autonome donc une pile ou une batterie associée à un régulateur dont l'objectif est de stabiliser la tension d'alimentation des divers composants du robot à une valeur fixe. Le châssis doit pouvoir se déplacer pour suivre une ligne droite ou courbe donc, pour tourner avec des roues fixes, il faut un moteur par roue et ajuster la vitesse de chaque moteur. Du fait de l'alimentation par batterie, il faut utiliser des moteurs à courant continu. Ceci suppose, en amont de chaque moteur, la présence d'un hacheur qui va moduler l'énergie délivrée au moteur pour en faire varier la vitesse et, en aval, d'un réducteur pour diminuer la vitesse et augmenter le couple.

Voyons maintenant la chaîne d'information. Il faut évidemment prévoir des boutons poussoirs pour commander le robot (marche/arrêt notamment). Pour suivre la ligne, il faut la repérer donc disposer d'un ou plusieurs capteurs optiques. Les informations issues du ou des capteurs doivent être traitées pour piloter correctement les moteurs via les hacheurs. Vu la taille du châssis, un système à microcontrôleur semble adapté pour réaliser ce traitement. Enfin, ce système peut communiquer des informations grâce à un afficheur.

En définitive, le schéma auquel on pourrait aboutir est le suivant :



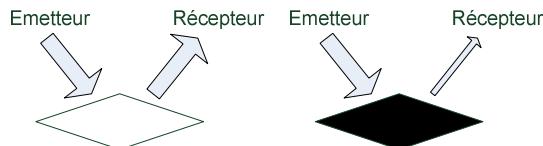
3. Etude de quelques parties du robot

Sans préjuger du choix technologique final, il est possible d'affiner celui de deux parties centrales de la structure du robot : le suivi de ligne et la commande des moteurs. De même, quelques compléments sur les boutons pousoirs peuvent être apportés.

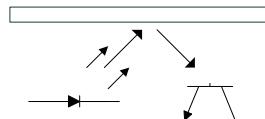
3.1 Suivi de ligne

Le suivi de ligne comporte deux aspects :

- un aspect métrologique : détection d'une ligne blanche sur un fond foncé ou détection d'une ligne noire sur un fond clair¹ ; ceci revient à distinguer le noir du blanc. Pour faire simplement cette distinction, on utilise un capteur composé d'un émetteur et d'un récepteur infrarouges : l'émetteur envoie une onde infrarouge sur la surface à analyser ; cette onde est, pour partie, absorbée par la surface et, pour partie, réfléchie vers le récepteur. C'est là qu'intervient la différence de comportement entre surfaces : la surface noire absorbe beaucoup plus l'onde que la surface blanche :

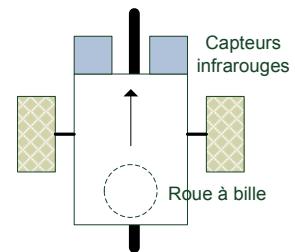


A noter que ce distinguo est aussi vrai pour le couple brillant/mat. Ainsi, les conditions de discrimination sont optimales pour des couleurs foncées et mates opposées à des couleurs claires et brillantes. Il suffit donc, au niveau du récepteur, de quantifier le rayonnement reçu pour discriminer les couleurs. Par ailleurs, l'utilisation d'une onde infrarouge permet aussi de s'affranchir significativement des variations de l'éclairage ambiant. En définitive, on peut utiliser une diode infrarouge comme émetteur et un phototransistor, c.-à-d. un transistor dont la base est sensible au rayonnement lumineux, comme récepteur :



- un aspect de mise en œuvre : combien de capteurs faut-il utiliser ? Où les positionner ? Comment gérer les informations issues des capteurs ? Différentes approches sont possibles et nous nous contenterons de présenter la plus simple :

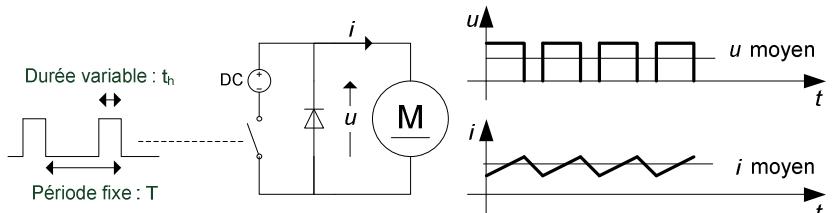
On utilise deux capteurs situés à l'avant du robot, de part et d'autre de l'axe médian, avec un entraxe un peu plus grand que la largeur de la ligne à suivre. Au démarrage du robot, on suppose que son axe médian est situé sur la ligne. Si celle-ci s'incurve par exemple vers la gauche, alors le capteur gauche va la détecter ; alors il faut faire tourner le moteur droit un peu plus vite de telle sorte que la ligne ne soit plus détectée par le capteur gauche mais pas trop vite cependant afin que le capteur droit ne se trouve pas en contact avec la ligne.



3.2 Commande des moteurs

La commande en vitesse des moteurs est réalisée à l'aide de hacheurs. Le hacheur est un convertisseur continu-continu qui permet d'obtenir une tension de valeur moyenne réglable à partir d'une source de tension continue, soit ici la tension de batterie régulée.

Dans sa forme la plus simple, le hacheur utilise un interrupteur commandé et une diode.



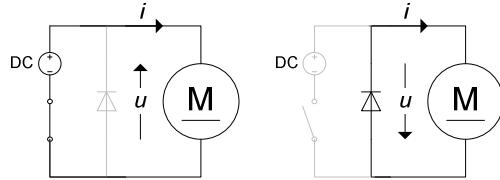
L'interrupteur est piloté avec une commande en créneau. La tension u du moteur est donc soit nulle, soit égale à la tension d'alimentation. Si la période T du créneau est faible, du fait de la nature inductive du moteur, le courant i

¹ Les deux sont possibles mais pas simultanément c'est-à-dire que l'on doit définir au départ la couleur de la ligne que l'on veut suivre.

est "lissé" et évolue autour d'une valeur moyenne. Il suffit donc de faire varier la durée t_h pour faire varier le courant moyen dans le moteur et donc sa vitesse. Cette commande en créneau est nommée "Modulation de Largeur d'Impulsion" (MLI²) ; elle est caractérisée par son rapport cyclique $\alpha = t_h/T$ qui évolue entre 0 et 1.

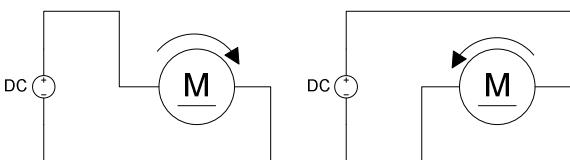
La diode, dite "de roue libre", permet d'éviter les surtensions aux bornes du moteur lorsque l'interrupteur s'ouvre. En effet, en l'absence de diode, le courant dans l'inductance du moteur passe brutalement de sa valeur nominale à 0 (d'où un di/dt élevé). Soit L l'inductance du moteur, on a à ses bornes : $u_L = L \frac{di}{dt}$ d'où un pic de tension néfaste.

A contrario, lorsque la diode est présente, à l'ouverture de l'interrupteur l'inductance va devenir génératrice et rendre la diode passante ; le courant circule alors, en roue libre, dans la boucle ainsi créée et il n'y a pas de surtension :

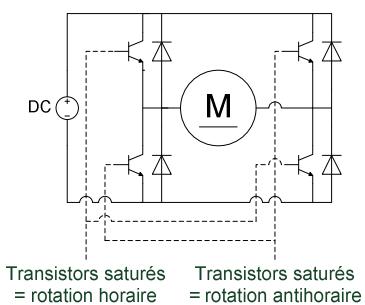


Les moteurs du robot devant fonctionner en marche avant et arrière, le schéma initial du hacheur doit être modifié. Pour la clarté du propos nous allons procéder en plusieurs étapes et temporairement considérer que l'on commande directement le moteur avec l'alimentation en tension, sans se préoccuper de la variation de vitesse.

1. Le changement du sens de rotation d'un moteur revient à permuter les polarités de sa tension d'alimentation, d'où les schémas ci-dessous :

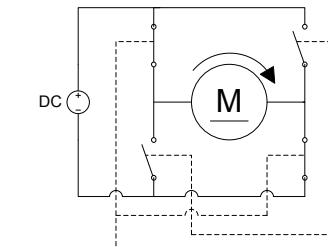


3. Les interrupteurs sont souvent réalisés à l'aide de transistors fonctionnant en commutation et, pour protéger les différents éléments contre les surtensions, chaque transistor est doté d'une diode de roue libre montée en parallèle :



Ce schéma est nommé "pont en H" de par sa géométrie.

Une fois que le sens de rotation est fixé en appliquant un courant approprié sur la base de chaque transistor, on peut faire varier la vitesse du moteur avec une commande MLI sur l'un des deux transistors initialement saturés. Le cheminement du courant est différent selon le niveau de tension de la commande et on retrouve le phénomène de roue libre pour le niveau bas :

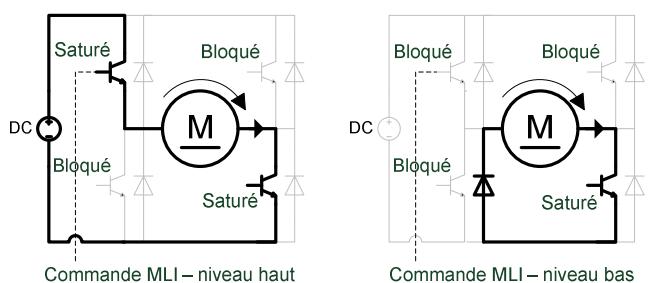
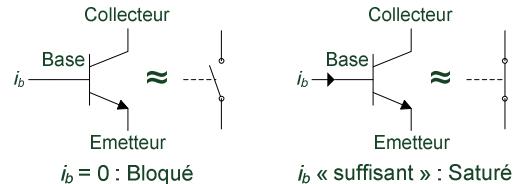


Interrupteurs fermés = rotation horaire
Interrupteurs fermés = rotation antihoraire

2. En combinant les deux schémas on peut, avec des interrupteurs commandés, choisir le sens de rotation. Les interrupteurs sont commandés par paires :

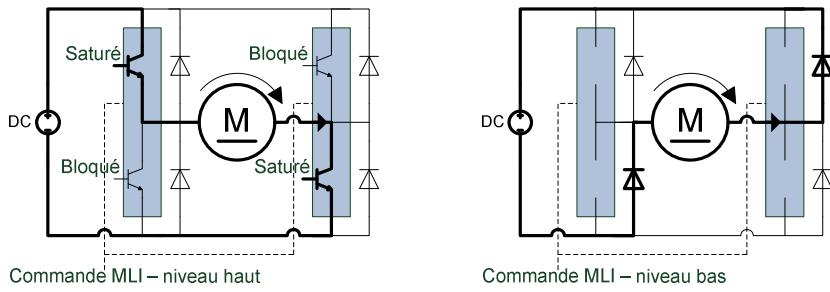
4. Rappel sur le transistor en commutation

Un transistor bipolaire fonctionnant en commutation a deux états selon la valeur du courant de base :



Lorsque les moteurs sont de faible puissance, il est possible d'utiliser des circuits intégrés qui contiennent les transistors et les diodes. Dans ces circuits, les éléments sont généralement conditionnés en demi-ponts afin d'être exploitables pour diverses applications. Les transistors de ces demi-ponts peuvent être activés ou inhibés à partir d'une logique de pilotage. Dans ce cas, on peut appliquer la commande MLI sur celle-ci comme présenté ci-après.

² En anglais : Pulse Width Modulation, soit PWM.

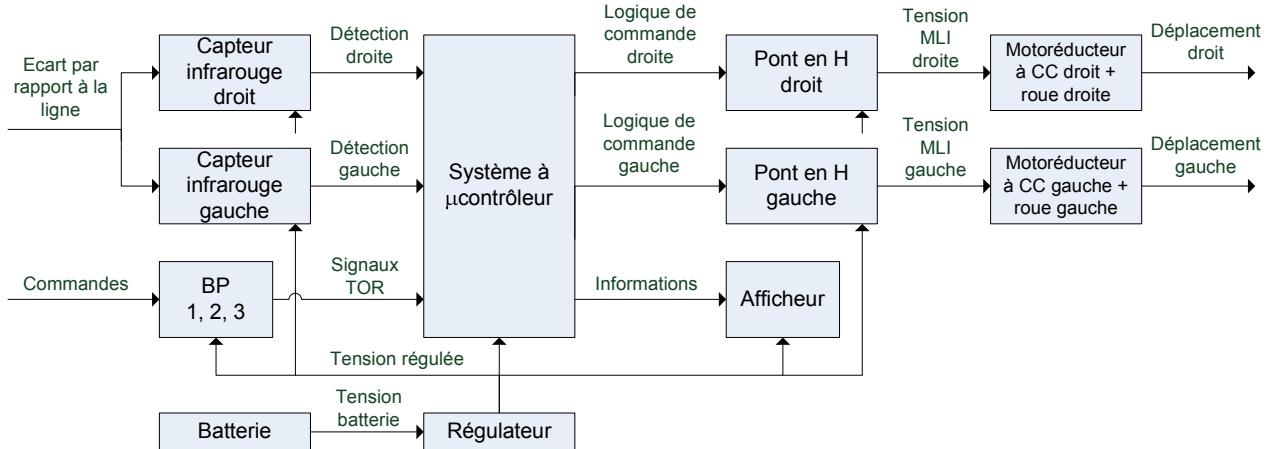


3.3 Boutons poussoirs

Comme déjà indiqué, la commande du robot se fera à l'aide de boutons poussoirs dont la fonction pourra varier selon le programme implanté dans le système à microcontrôleur (mise au point, pilotage automatique ...). On décide de fixer leur nombre à trois, ce qui semble suffisant pour traiter l'ensemble des scénarii envisagés.

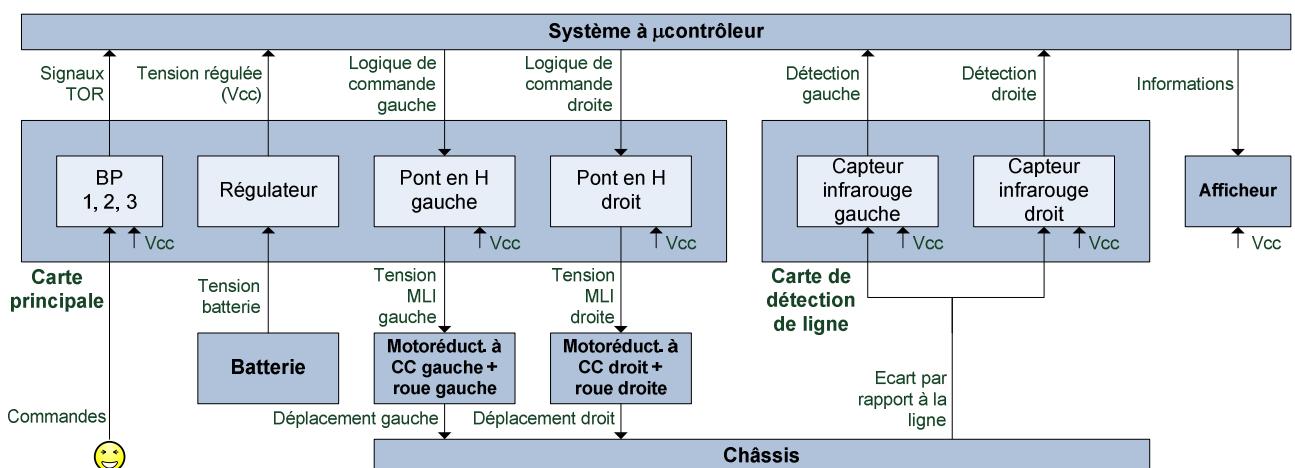
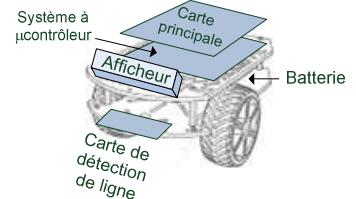
4. Schéma structurel

Sur la base de ce qui précède, voici un schéma plus précis du robot :



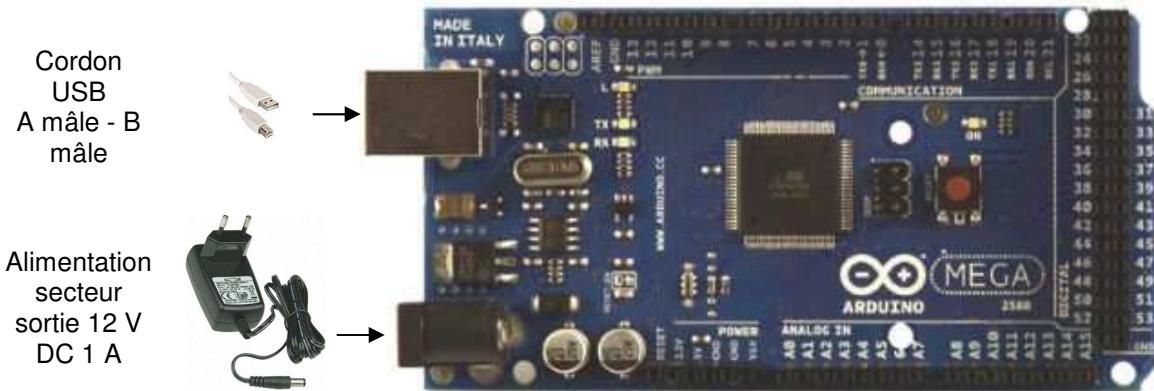
Ce schéma présente les éléments principaux sans préjuger de leur réalisation. Pour avancer sur ce dernier point, on peut préciser que le système à microcontrôleur, l'afficheur, les motoréducteurs et la batterie sont des éléments intégrés avec lesquels on peut interagir, via des fils de liaison, mais pour lesquels il n'y a pas de développement matériel à prévoir. Par contre, les autres éléments doivent être implantés sur un circuit imprimé avec, le cas échéant, l'usage de composants complémentaires (résistance, condensateur, fusible ...).

D'un point de vue pratique, on décide de réaliser deux circuits imprimés : une carte de détection de ligne sur laquelle les capteurs infrarouges seront positionnés et qui sera située à l'avant du robot, et une carte principale, placée sur la partie supérieure du robot, au dessus du système à microcontrôleur, et supportant l'ensemble des autres composants. Les schémas ci-contre et ci-après précisent cette organisation. Les liaisons entre le châssis et les éléments se font avec des entretoises.



5. Choix du matériel

En définitive, on décide de piloter le robot à l'aide d'une carte de développement Arduino MEGA 2560 :



Cette plate-forme est à code source libre et elle utilise un microcontrôleur AVR ATmega 2560 dans son noyau. Pour communiquer avec l'environnement extérieur, elle dispose de :

- 54 broches d'entrées/sorties numériques dont :
 - 14 peuvent être utilisées comme sorties MLI,
 - 8 permettent de réaliser 4 ports séries (Universal Asynchronous Receiver Transmitter - UART),
 - 2 sont paramétrables pour mettre en œuvre un bus I²C (Inter-Integrated Circuit),
- 16 entrées analogiques.

La carte est reliée à un PC via une connexion USB qui peut servir d'alimentation (5 V et 500 mA maximum). Il est également possible d'utiliser une alimentation externe, dans une plage recommandée comprise entre 7 et 12 V : soit un adaptateur secteur avec un connecteur jack 2.1 mm, soit une batterie. A noter que la carte dispose d'un régulateur de tension ; dans le cas de l'utilisation d'une batterie celle-ci peut donc directement alimenter la carte.

Comme alimentation autonome, on choisit une batterie de 9 V avec une capacité de 500 mAh ; les autres éléments sont présentés ci-dessous. Des informations plus détaillées quant à leurs fonctionnement et câblage se trouvent dans les notices accessibles à partir du site du département.

Libellé	Châssis	Motoréducteur à CC	Convertisseur DC/DC (régulateur)	Porte-fusible	Cartouche fusible	Bouton-poussoir
Référence	DG007	DG01D	Tracopower	5 x 20mm OGN 10 A	5x20mm,1A, 250 V, type F	D6C
Fournisseur	DAGU	DAGU	TSR-1-2450	Schurter	Schurter	C & K Components
Visuel						

Capteur infrarouge	Quadruple 1/2 pont en H	Support de circuit intégré	Afficheur 4 digits, 7 segm.	Connecteur mâle sécable	Embase femelle sécable	Bornier fil-à carte	Câbles de connexion M/F
OPB608A	L293D	A 16-LC-TT	COM-11629	STB40L	BL1-036	1729128	BBJ3
Optek	Texas Instruments	Assmann WSW	Sparkfun	Gotronic	E-TEC	Phoenix Contact	Gotronic

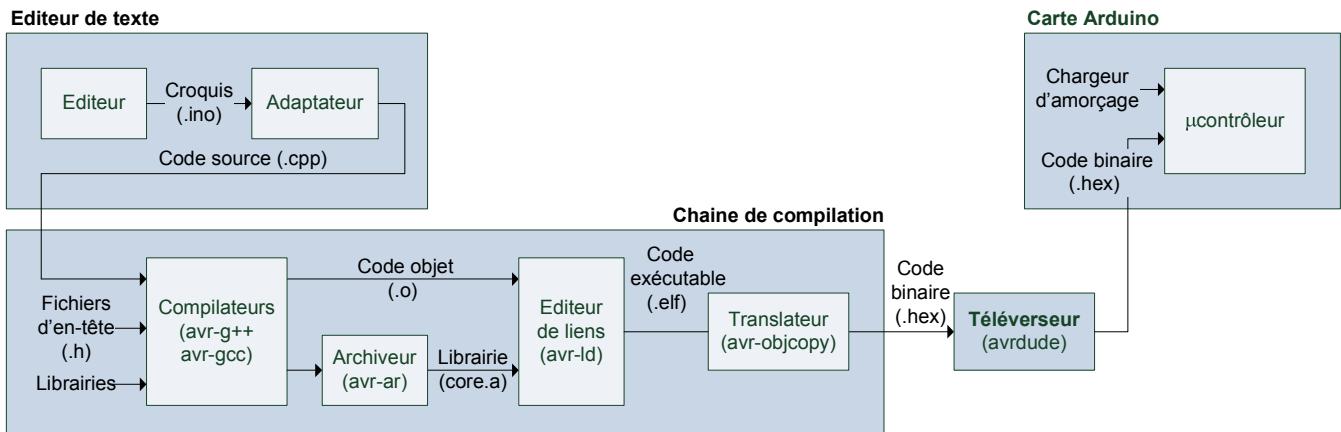
Les connecteurs, embases et borniers à vis sont soudés sur les circuits imprimés et permettent de les relier, pour les premiers, à la carte Arduino (via les câbles de connexion) et pour les derniers, aux motoréducteurs. D'autres éléments (résistance, condensateur, transistor) sont susceptibles d'être utilisés pour mettre en œuvre certains composants électroniques.

6. Aspects logiciels

La carte Arduino est programmée en langage compilé C++. Pour réaliser cette programmation, on utilise un environnement de développement intégré (EDI) écrit avec le langage de programmation orienté objet Java et inspiré du langage Processing. L'EDI est composé de trois éléments principaux :

- un éditeur de texte qui permet de créer, modifier, sauvegarder ... le code source. Un programme "Arduino" créé par l'éditeur est nommé Croquis (*Sketch*) ; il a pour extension ".ino" (les 3 dernières lettres d'Arduino). Ce fichier est transformé via un adaptateur en un fichier source C++ avec l'extension .cpp (*C plus plus*) en ajoutant le fichier d'en-tête correspondant à la carte Arduino (instruction : #include <Arduino.h>).
- une chaine de compilation adaptée aux microcontrôleurs AVR qui réalise une analyse lexicale, syntaxique et sémantique du code source puis signale les éventuelles erreurs et transforme le code source en code binaire exécutable par le microcontrôleur de la carte Arduino. Il s'agit d'un compilateur croisé, c.-à-d. que le code objet peut être exécuté sur un processeur différent de celui du compilateur. Le fichier source est compilé avec les compilateurs C++ (avr-g++) et C (avr-gcc ; gcc = *GNU Compiler Collection*). On obtient alors un fichier objet avec l'extension .o (*object*) et un fichier de dépendance, qui recense les fichiers utiles pour la construction du fichier objet, avec l'extension .d (*dependancy*). Outre le code source, le compilateur peut être amené à intégrer des informations issues de fichiers d'en-tête (*header*) ou de diverses librairies. Les fichiers sources correspondants sont compilés et les fichiers objets résultants sont rassemblés dans une librairie core.a via un archiveur avr-ar (*archiver*). L'éditeur de lien avr-ld permet alors de lier le code objet à la librairie et de créer un code exécutable au format elf (*executable and linkable format*). Enfin, le translateur avr-objcopy transforme le code précédent en code binaire au format hex (*hexadecimal*) d'Intel.
- un téléverseur du code objet sur la carte Arduino (*uploader*) via la liaison série USB : avrdude. Une fois le code objet transféré, la carte peut être dissociée du PC, le chargeur d'amorçage (*bootloader*) chargeant le code objet dans le microcontrôleur au démarrage.

Le schéma ci-dessous précise l'interaction entre ces différents éléments :



L'interface graphique de l'EDI est, de haut en bas, composée :

- d'une série de menus,
- d'une barre d'outils avec, à gauche, des boutons correspondant aux fonctions les plus courantes (vérifier/compiler, téléverser, nouveau, ouvrir, enregistrer) et, à droite, un bouton d'accès à un moniteur série permettant de communiquer avec la carte. Sur ce moniteur, la carte peut afficher une chaîne de caractères ou la valeur d'une variable du programme ; elle peut aussi récupérer des caractères tapés au clavier.
- de la zone de l'éditeur de texte,
- d'une zone de message,
- d'une zone de texte donnant des informations plus détaillées que la précédente sur les erreurs, la compilation et le téléchargement.



Avant la 1^{ère} utilisation, il faut configurer l'EDI (menu Outils), ce qui revient à préciser le type de carte, voire de processeur, le port de communication utilisé et le type de programmateur.

Un croquis est structuré en trois parties :

- une partie déclaration des variables (optionnelle),
- une partie initialisation et configuration des entrées/sorties : la fonction setup () ; c'est dans cette partie que l'on va notamment retrouver les fichiers d'en-tête,
- une partie principale qui s'exécute en boucle : la fonction loop ().

A noter qu'il n'est pas nécessaire que la carte Arduino soit connectée pour utiliser l'éditeur de texte et compiler. Par ailleurs, dans le menu "Fichier + Préférences", on peut spécifier les numéros de lignes ainsi que le niveau de détail des informations (déroulement, erreurs) de compilation et de téléversement.

7. Etapes du projet

A ce stade de connaissance de la structure et du fonctionnement du robot, on peut résumer le projet ainsi : à partir d'un ensemble de composants et à l'aide des notices, il s'agit de réaliser un robot selon un cahier des charges prédefini, avec un appui matériel et humain du département. C'est un projet pluridisciplinaire qui associe électronique, électrotechnique, microinformatique, communication et gestion de projets.

Le projet vise à acquérir des connaissances nouvelles dans les domaines techniques mais aussi à se construire une méthode de travail, à développer l'autonomie et l'initiative, la capacité à travailler efficacement en groupe, toutes qualités indispensables dans une vie professionnelle. Il convient donc de ne pas l'aborder, dans un suivisme quiet, comme une succession de séances de travaux pratiques. Les modalités pédagogiques sont d'ailleurs différentes entre TP et projet : le premier est encadré, le second tutoré ; dans le premier les consignes sont précises, dans le second on se limite à des objectifs généraux.

Toutes les phases de mise en œuvre du robot ne pouvant être exécutées dans le volume horaire imparti, certaines étapes significatives ont été sélectionnées et devront être réalisées ; elles sont présentées ci-après. Cependant, si les étapes indiquées ne peuvent qu'être accomplies dans l'ordre indiquée, il est recommandé, au sein d'une tâche, de paralléliser au maximum les activités voire d'anticiper certains aspects préparatoires de tâches ultérieures.

Les étapes 1 à 3 et 5 sont communes à tous les groupes, l'étape 4 est spécifique à chaque groupe.

1. *Réalisation du schéma de principe de la carte principale*

A partir des notices, il s'agit de comprendre le fonctionnement des composants de la carte principale, leur câblage (avec éventuellement l'ajout d'éléments complémentaires) et l'interfaçage avec la carte Arduino, puis de réaliser le schéma de principe de cette carte. Sur ce schéma, on doit retrouver tous les composants ainsi que les connecteurs et borniers, à l'exception de l'afficheur et des capteurs infrarouges. Le schéma peut être réalisé à la main, en utilisant un logiciel de dessin ou avec l'éditeur de schémas du logiciel KiCad utilisé à l'IUT.

Les trois boutons-poussoirs vert, rouge et blanc ont respectivement comme fonction : commande de la marche avant, commande de la marche arrière, reset du microcontrôleur. A noter que la fonction des boutons vert et rouge est purement indicative : selon la programmation, l'appui sur un bouton peut provoquer une action différente ; par contre le schéma de câblage des boutons reste inchangé.

- Livrable : schéma de la carte principale

2. *Implantation des composants sur la carte principale et liaison de celle-ci avec les autres éléments*

Cette tâche consiste en l'implantation des composants sur la carte principale fournie puis en la réalisation des liaisons filaires entre la carte et la batterie, les motoréducteurs, la carte Arduino. Ce montage permettra de tester la carte. **Attention** : la batterie et l'alimentation secteur ne doivent pas être branchés simultanément.

- Livrable : carte principale opérationnelle

3. *Programmation du microcontrôleur : commande des moteurs*

L'objectif de cette tâche est, d'une part, de s'approprier le fonctionnement de la carte Arduino et, d'autre part, de vérifier le bon fonctionnement de la carte principale. A cette fin, on réalisera un programme qui fait tourner les moteurs en marche avant ou arrière selon l'appui sur les boutons vert ou rouge. On vérifiera également que la vitesse des moteurs peut être modifiée (prendre une fréquence de MLI supérieure à 1 kHz).

- Livrables :
 - motorisation opérationnelle
 - programme de commande des moteurs en marche avant/arrière à vitesse variable

4. *Réalisation de la partie spécifique du cahier des charges*

Cette étape concerne l'analyse du cahier des charges spécifique puis, selon le projet, la réalisation complète d'une carte ou l'assemblage de cartes et composants fournis, suivi de la programmation associée.

- Livrables :
 - robot opérationnel
 - schéma électrique et, le cas échéant, PCB
 - programme commenté

5. Présentation et démonstration

Lors de cette étape terminale, les étudiants présentent leur projet avec comme support visuel deux posters, l'un présentant le déroulement du projet, notamment son phasage, et l'autre détaillant les aspects techniques. Après une présentation orale de 2 à 3 minutes, une démonstration du fonctionnement du robot est réalisée.

- Livrable : rapport de projet et posters

8. Organisation pédagogique

Le projet tutoré est planifié en deux phases. La 1^{ère} phase, qui se déroulera de mi-février à fin mai, concerne la réalisation des étapes 1 à 3. Pendant celle-ci, une plage horaire de quatre heures par quinzaine sera bloquée spécifiquement à l'emploi du temps et deux enseignants "ressources" pourront être sollicités. Bien évidemment, la réussite du projet suppose une implication allant au-delà de ces heures ainsi qu'une bonne répartition des tâches entre les membres du groupe. La 2^{ème} phase (étape 4) sera réalisée courant juin : après tirage au sort de leur partie spécifique et échange avec l'enseignant référent, les groupes disposeront de quatre jours en autonomie pour finaliser leur projet.

A noter que le technicien du département pourra être mis à contribution, pour du prêt de matériel et pour la réalisation des circuits imprimés notamment.

9. Evaluation

Le projet sera évalué sur la base :

- d'un QCM validant la compréhension des méthodes et outils du projet, synthétisés dans ce document (février)
- du schéma de la carte principale (mars)
- d'un QCM technique validant la compréhension du fonctionnement de la maquette (avril)
- d'une démonstration du fonctionnement de la motorisation du robot, avec programme associé (mai)
- d'une présentation et démonstration du robot complet avec rapport de projet et posters (juin)

Le cas échéant, s'il s'avère qu'une distorsion d'implication entre les membres d'un groupe est constatée, les notes pourront être individualisées.

Annexe : brochage de la carte MEGA

