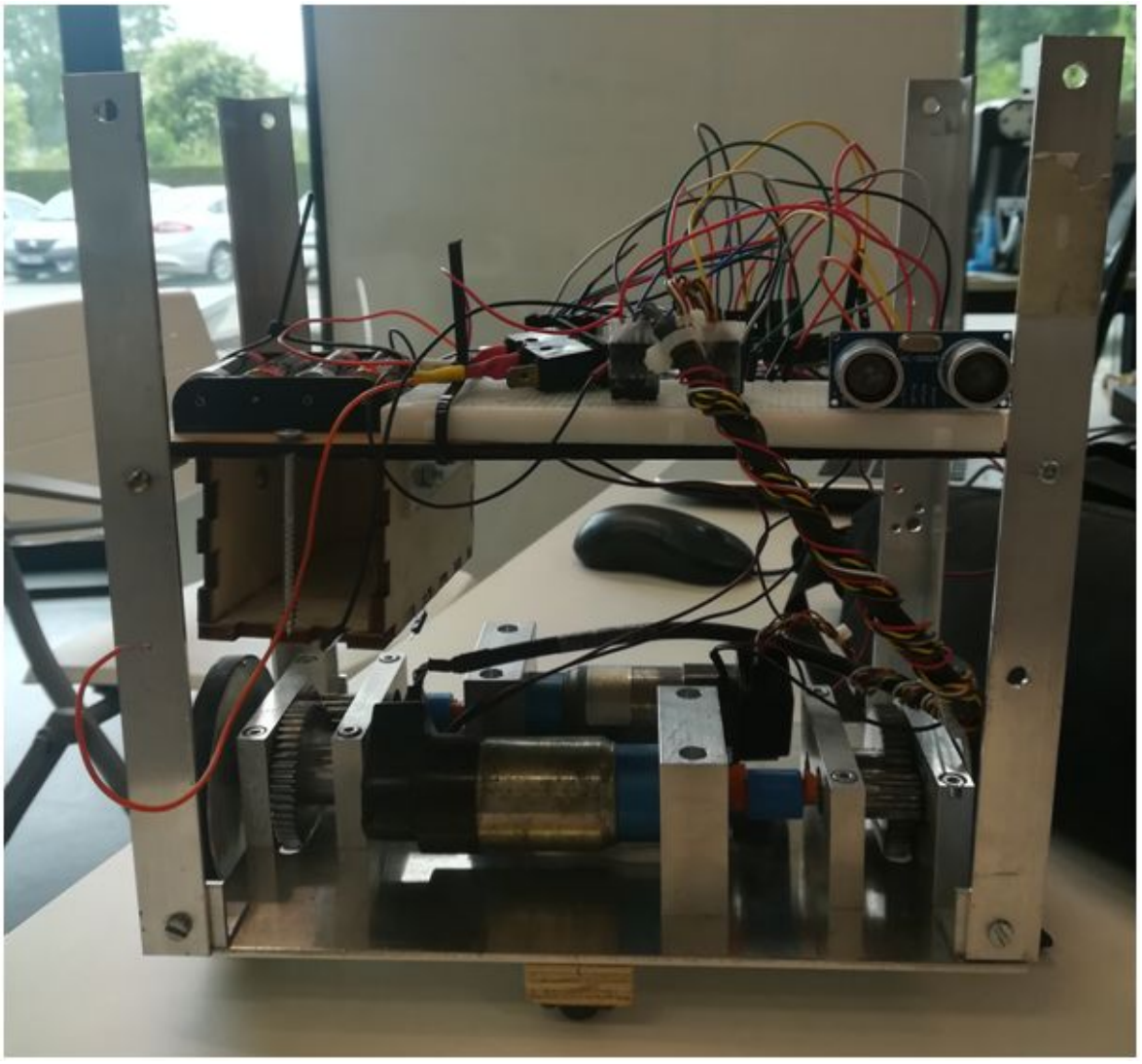
Raphaël Bonvalet
Maxime Fontaine
Pierre Gautreau
Jérémie Hassenforder

Projet IMA3-IMA4
Coupe de France de Robotique



POLYTECH LILLE

Université de Lille

Introduction

Chaque année se déroule la coupe de France de robotique, une compétition nationale, où plus d'une centaine d'équipes s'affrontent avec leurs robots autonomes capables de réaliser diverses tâches imposées. Le but est presque similaire chaque année: accomplir des objectifs comme récolter des objets sur le terrain de jeu pour marquer des points.

Notre projet consiste donc à réaliser un robot capable de se présenter à la coupe de France de Robotique 2020. Nous devons faire en sorte que celui-ci soit autonome pour une durée de trois minutes minimum, soit la durée de deux matchs. Dans un premier temps, nous devons réaliser le robot, puis commander ses déplacements de façon précise et enfin, quand le règlement sera sorti, réfléchir à une stratégie et aux actionneurs nécessaires pour marquer le plus de points possible.

Pour la fin de ce semestre nous voulons un robot construit proprement et qui peut atteindre une position donnée dans l'espace en un temps le plus court possible.

I. Cahier des Charges

1. Le robot doit être autonome en énergie
2. Le robot doit pouvoir faire une ligne parfaitement droite
3. Le robot doit pouvoir se rendre à un point de l'espace pré-programmé.
4. Il doit pouvoir se repérer dans l'espace en gardant en mémoire sa position par rapport à repère fixe
5. Il doit éviter tout contact frontal avec son environnement
6. Il doit être capable d'éviter un obstacle tel qu'un autre robot et puis continuer son chemin vers son objectif

II. Réalisation de la structure du robot

Il y a quelques contraintes pour créer le robot, principalement des contraintes de tailles comme on peut le voir ci dessous :

Dimensions du robot principal et du robot secondaire :
On mesure le périmètre d'un robot en l'entourant comme le montrent les illustrations ci-dessous :

Dimensions du robot principal :		Dimensions du robot secondaire :	
	Non déployé <= 1200mm Déployé <= 1500mm		Non déployé <= 850mm Déployé <= 1050mmm

Le périmètre du robot principal ne doit pas excéder 1200 mm au moment du départ. Le périmètre de ce robot principal totalement déployé ne doit pas excéder 1500 mm au cours du match.

Le périmètre du robot secondaire est indépendant de celui du robot principal. Il ne doit pas dépasser 850 mm au moment du départ et 1050 mm lorsqu'il est totalement déployé au cours du match.

À tout instant au cours du match, la hauteur du robot principal et du robot secondaire ne doit pas dépasser 350 mm. Cependant, il sera toléré que le bouton d'arrêt d'urgence dépasse de cette hauteur limite pour atteindre 375 mm.

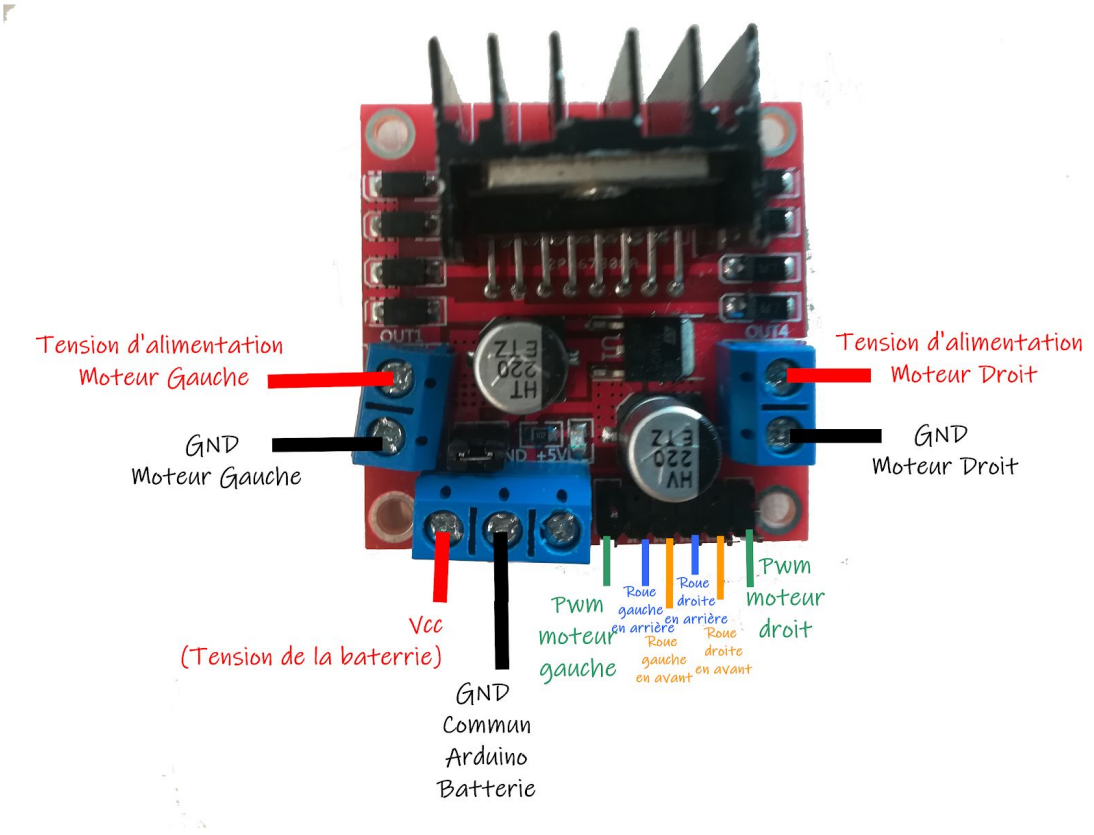
Extrait du règlement de la coupe de France 2019

Pour réaliser nos robots, nous devrons donc prendre en compte ces paramètres. Nous avons récupérer la base roues + moteurs d'un ancien robot de Robotech. Nous devons à présent construire le reste.
Nous avons fait le choix de fixer des équerres sur les 4 coins du robot. Puis à ces équerres, nous choisissons d'y fixer une plaque de bois où l'on viendra attacher tous les composants par des liens colson dans l'attente de la réalisation d'un PCB. Nous avons laisser un maximum d'espace pour pouvoir greffer les modules qui serviront à réaliser les objectifs fixés par le règlement de la coupe de France 2020.
Concernant le robot secondaire, nous avons fait le choix de réaliser complètement la structure. Afin de réaliser cette structure nous avons utiliser du bois que nous avons découpé à l'aide de la découpeuse laser du Fabricarium. Pour la forme du robot nous avons décidé de suivre le même format que le principal. Afin de les fixer au mieux nous avons utilisé le système de "boîtes à encoches" pour une surface de contact optimale.

III. Contrôle des Moteurs

1. Le controlleur moteur

On a choisit un controlleur moteur L298 dual Full-Bridge Driver car il permet de supporter une tension d'alimentation de 46V pour un courant de 4A max et de contrôler les deux moteur avec une seule carte.



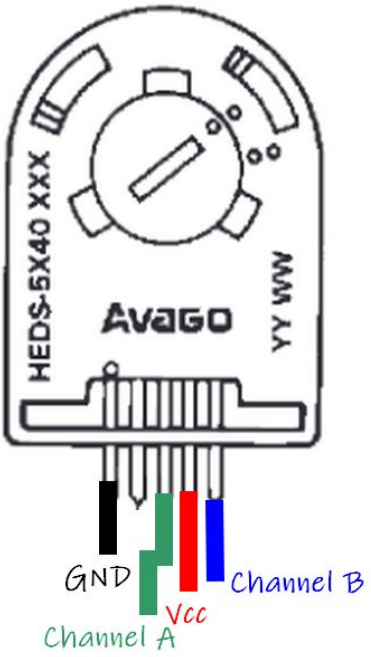
Controlleur Moteur L298

La batterie pleine charge délivre une tension de 16V,qui est connecté sur la broche Vcc du controlleur moteur qui se charge de la délivrer aux moteurs en fonction des entrées PWM (sorties de l'arduino).On connecte donc les deux moteur sur les broches de côté. Les pins en bleu et en orange permettent de gérer la marche avant ou la marche arrière du robot à l'aide d'un DigitalWrite(pin,High/low) sur arduino.
En vert, on envoie une valeur comprise entre 0 et 255 (soit 0V->5V) qui correspond à la pwm des moteurs c'est à dire le pourcentage de tension de la batterie envoyé au moteur. On utilise pour cela la fonction AnalogWrite(pin,valeur).

Le ground du moteur doit être commun avec celui de l'arduino pour que le signal de commande fonctionne.

2. Les encodeurs

On choisit des encodeurs HEDS-5500 qui on le grand avantage de pouvoir être fixés sur les moteurs que nous avons choisi. De plus ils fonctionnent par interruption, c'est à dire qu'ils incrémentent ou décrémentent dès qu'ils détectent une variation.



Le ground est connecté en commun avec ceux de l'arduino et le la batterie, le Vcc est le 5V de l'arduino. Le channel A permet de compter le nombre de tics effectués par le moteur. Nous avons mesurés le nombre de tics effectués par 100 tours de roue et on en a déduit qu'un tour représente 1485 tics.

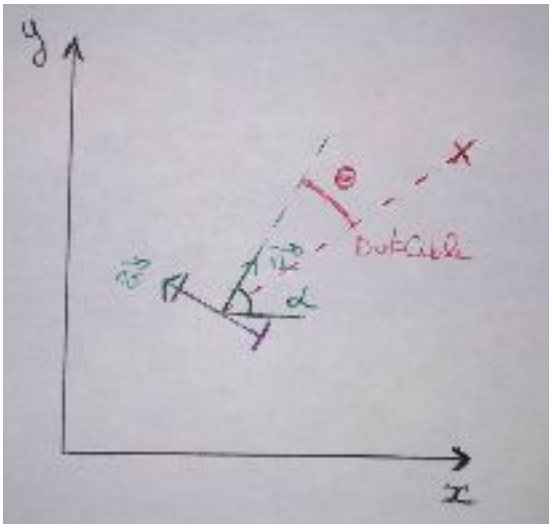
Le channel B permet de connaître le sens de rotation pour pouvoir incrémenter ou décrémenter le nombre de tics et ainsi savoir quelle est la distance parcourue par le robot.

Grâce au nombre de tics par tour de roue et au diamètre de la roue (82 mm), on peut en déduire que:

$$1\ tic\ = \frac{82*}{1485} = 0.17\ mm$$

IV. Asservissement

A terme le robot doit se déplacer d'objectifs en objectifs. Ces objectifs ont pour la plupart une localisation connue à l'avance. Etant donné que nous disposons d'encodeurs, nous utilisons la technique de l'odométrie. L'encodeur présent sur chaque roue, nous permet de connaître la position et la vitesse de chacune. Cette technique permet au robot de connaître sa position dans l'espace en temps réel. Ceci nous offrira par la suite une grande liberté dans l'exploitation de notre automate.



En partant du principe que nous connaissons la position de l'automate à un instant *t*, il faut désormais déterminer sa position à l'instant *t+1.* Pour cela nous avons déterminé la mobilité de l'automate en fonction de la contribution de chaque roue. Ce qui nous donne des équations horaires.

$$x(t) = \left[\cos\left(\alpha(t)\right) \cdot \frac{\dot{\theta}_G + \dot{\theta}_D}{2} \cdot r_{roue}\right] \Delta t + x(t-1)$$

$$y(t) = \left[\sin\left(\alpha(t)\right) \cdot \frac{\dot{\theta}_G + \dot{\theta}_D}{2} \cdot r_{roue}\right] \Delta t + y(t-1)$$

$$\alpha(t) = \left[\frac{\dot{\theta}_D - \dot{\theta}_G}{d}\right] \cdot \Delta t + \alpha(t-1)$$

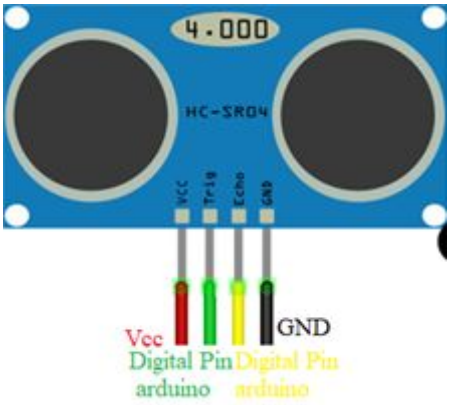
Nous commandons le robot en lui donnant les coordonnées de sa prochaine cible. Nous connaissons la distance à parcourir ainsi que l'angle entre le chemin le plus court et l'axe de déplacement "*u"* du robot.
Nous asservissons la commande de rotation du robot afin que ses temps de trajet soient le plus court possible. Pour cela nous avons recours à un PID numérique dont voici le calcul de la correction :

*deltaCommande = kP*Erralpha + kI*Salpha + kD * dErr*

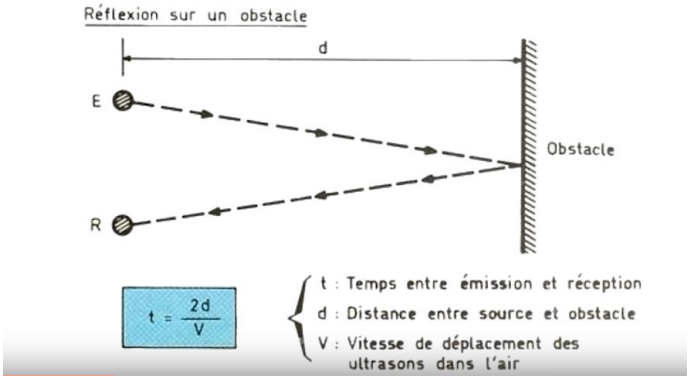
Erralpha est l'erreur sur l'angle : la différence entre la consigne et l'angle actuelle du robot.
Salpha est la somme des Erralpha.
dErr est la différence entre l'erreur sur l'angle actuelle et l'erreur précédente.

V. Repérage et évitement d'obstacle

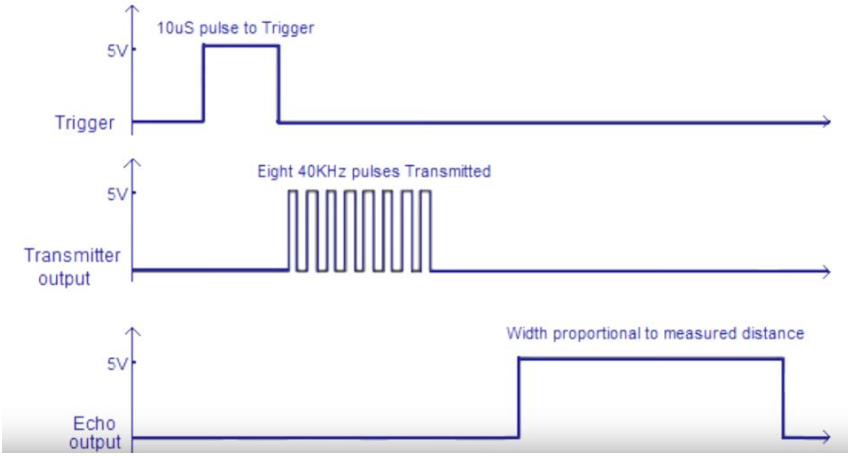
Pour le repérage d'obstacle, nous avons tout d'abord choisi la solution du capteur ultrasons. Cette solution nous paraissait la plus évidente car une des moins chère tout en restant assez précise. Le capteur ultrasons se câble de la manière suivante :



Le capteur de distance comporte un émetteur et un récepteur d'ondes qui sont représenté sur le schéma ci-dessus par les deux ronds noirs. L'émetteur envoie une onde, celle-ci se réfléchit sur l'obstacle et le récepteur reçoit donc un écho de l'onde émise par l'émetteur.

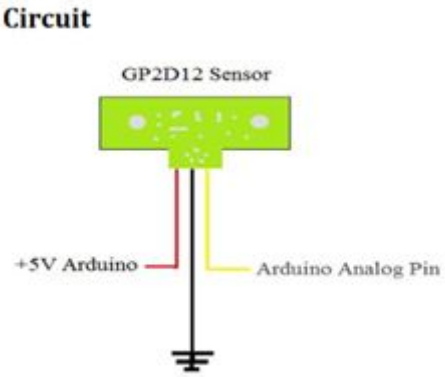


Pour programmer ce capteur, on doit envoyer du 5V sur le pin Trig (PIn vert au dessus) pendant une durée de 10µs. Après cela, le capteur va émettre 8 impulsions de 40kHz. Et de ce fait l'Echo (Pin jaune au dessus) va envoyer du 5V à l'arduino pendant une durée proportionnelle à la distance mesurée. Il suffit donc ensuite d'appliquer une formule pour avoir la distance qui sépare le capteur à l'obstacle.



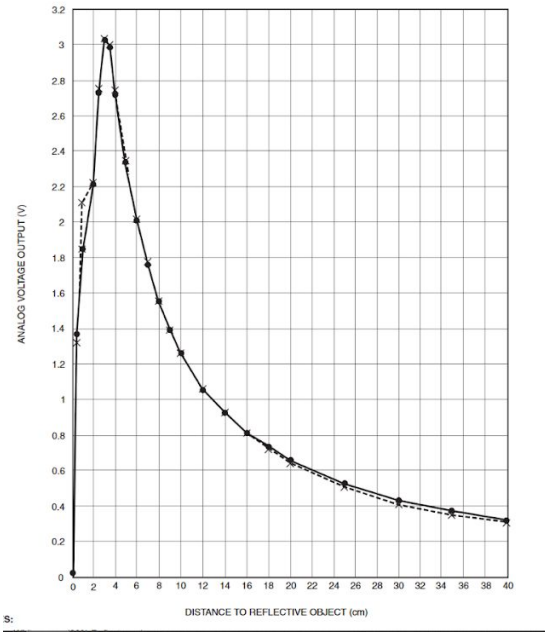
Le problème que l'on pourrait rencontrer en utilisant un capteur ultrasons est que nos adversaires utilisent eux aussi cet ultrasons est que de ce fait le robot esquive alors qu'il n'y avait pas d'obstacle.

Nous nous sommes ensuite intéressés au capteur de distance infrarouge car, comme pour le capteur ultrasons, il n'est pas cher et est assez précis. Le capteur infrarouge se câble de la manière suivante :



Comme on peut le voir sur le pin jaune, le capteur infrarouge n'utilise pas de pin digital comme le capteur ultrasons mais il utilise un pin analogique. Ce qui peut être plutôt pratique comme beaucoup de nos composants utilise les pins digitals, je pense notamment au contrôleur moteur qui en utilise 6 par exemple. Par contre, tout comme le capteur ultrasons, il est composé d'un émetteur infrarouge (LED infrarouge) qui va éclairer l'obstacle et d'un récepteur (photodiode ou phototransistor) qui va mesurer la lumière réfléchie et donc la renvoyer de manière analogique à l'arduino.

Le problème de ce capteur est qu'il réagit de manière non proportionnelle à la distance comme on peut le voir sur ce graphique trouvé sur la datasheet du composant :



Par contre, grâce à celui-ci nous pouvons donc en déduire l'équation qui est la suivante : distance = (2914 / (analogInput - 3)) - 4 et donc en déduire la distance à un obstacle à chaque instant.
Le deuxième problème rencontré avec ce capteur est qu'il peut être facilement déjoué rien que par la lumière du soleil ou d'autres lumières qui peuvent être présentes lors du concours. Ce capteur peut aussi être déjoué à cause d'une matière non réfléchissante, en effet si le robot adverse est équipé d'une matière non réfléchissante noir la réflexion de la lumière envoyé sera très faible. De ce fait l'interprétation de la distance sera mauvaise et le risque de collision augmente.

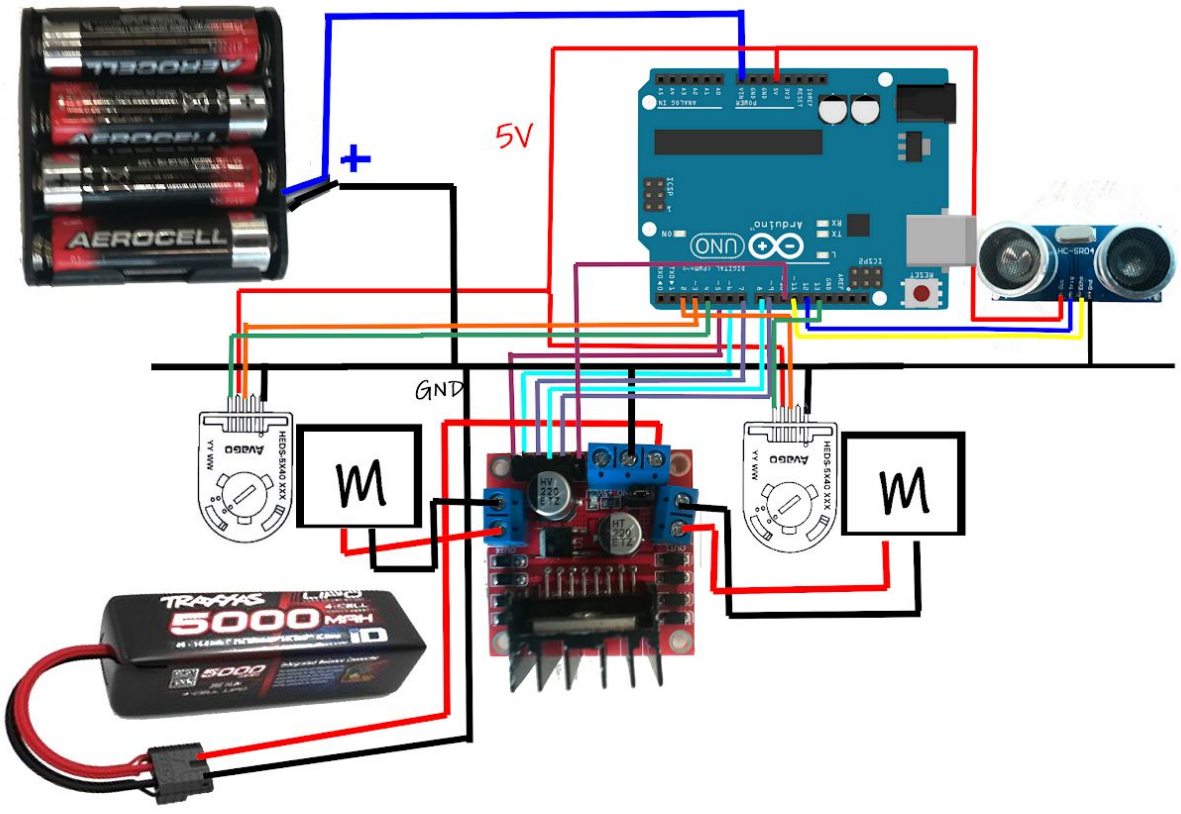
Par conséquent nous allons placer le capteur ultrasons sur le robot principal et le capteur infrarouge sur le robot secondaire maintenant qu'il est construit pour voir comment ils vont réagir lors de leur rencontre. Pour ainsi par la suite les faire communiquer entre eux.

VI. Conclusion: ce qu'il reste à faire

Nous avons réussi à concevoir un robot qui arrive à se déplacer d'un point A à un point B. Il sera donc capable d'atteindre sa cible, il ne lui restera ensuite qu'à marquer des points. Pour réfléchir à cela, nous devrons attendre la sortie du règlement de la coupe de France 2020, d'ici octobre prochain. Mais si l'on se réfère aux années précédentes notre robot devra certainement ramasser des cibles et les mettre dans dans zones prédéterminés. A nous maintenant de concevoir ces actionneurs qui nous permettront de marquer le plus de points possibles. N'y

Nous devons aussi programmer le robot de telle sorte qu'il arrive à éviter les obstacles qui se dressent devant lui, ce que nous n'avons pas eu le temps de faire. Il devra les contournés et poursuivre son chemin vers sa cible.

Annexes :



Câblage du robot