Bibliographie projet Arduino

Objectif : construire une voiture contrôlée par une télécommande (joystick) et qui s'arrête automatiquement dès qu'un obstacle se présente à l'avant ou à l'arrière. La voiture est capable également de se garer automatiquement sur le côté gauche (selfparking)

Moteur pour diriger les roues

Il existe plusieurs types de servomoteurs : -
- Positionnements angulaires
- Rotation continue
- Déplacement linéraire

Si on veut diriger les roues, on doit choisir un servomoteur angulaire car il permet de déplacer un objet (en l'occurrence les roues) le plus couramment dans une plage de 0 à 180°. Il existe d'autres servomoteur angulaire permettant de déplacer un objet dans d'autres plages (par exemple de 0 à 360°) mais celui allant jusqu'à 180° suffit puisqu'il est inutile que les roues puissent faire un tour complet.

Pour pouvoir piloter se servomoteur, il faut installer la bibliothèque servo.h. On peut noter que l'utilisation de servos limite l'Arduino (sauf si c'est une Mega). En effet l'instruction « analogWrite() » ne peut plus être utilisé sur les broches 9 et 10 (A VERIFIE).Il n'existe visiblement pas d'autre bibliothèque que servo.h pour contrôler un servomoteur.

Cette bibliothèque possède plusieurs fonctions qui pourraient être utile :

- Attach() qui est obligatoire pour initialisé la bibliothèque. Elle prend un argument unique correspondant au numéro de la broche sur laquelle le servo est branché. Elle peut également prendre 2 paramètres optionnel « min » et « max » correspondant aux durées minimum et maximum de l'impulsion, c'est-à-dire de l'angle de rotation des roues.

- Write() permettant de modifié l'angle du bras du servomoteur en prenant en paramètre un entier compris entre0 et 180 correspondant à l'angle en question.

- Read() permettant d'obtenir la dernière valeur connue de Write()

- -writeMicroseconds() prenant en paramètre la durée d'impulsion a transmettre au servomoteur. Cette fonction est similaire a write() mais peut-être plus précise.

Parmi les différents servomoteurs angulaires, celui qui semble le plus courant est le sg90. Le servomoteur sg90 possède donc une amplitude de 180° avec une tension de fonctionnement entre 4.8 et 6V. Il a l'avantage de ne peser que 9g et ces dimensions sont assez modestes (22mm x 11.5mm x 27mm). Son couple s'élève a 1.2kg/cm (sous 4.8v) ce qui est suffisant pour faire tourner les roues.

Parmi les autres références, on retrouve le FS90R qui possède à peu près les mêmes caractéristiques si ce n'est un couple à peine plus élevé mais le composant est légèrement plus grand.

De même concernant le ms2810mg qui malgré l'avantage de posséder un couple élevé (2,8kg/cm) pèse assez lourd (environ 21g).

D'autres servomoteurs ne disposent pas d'une amplitude suffisante comme c'est le cas du FS0403 (amplitude de 0 à 120°).

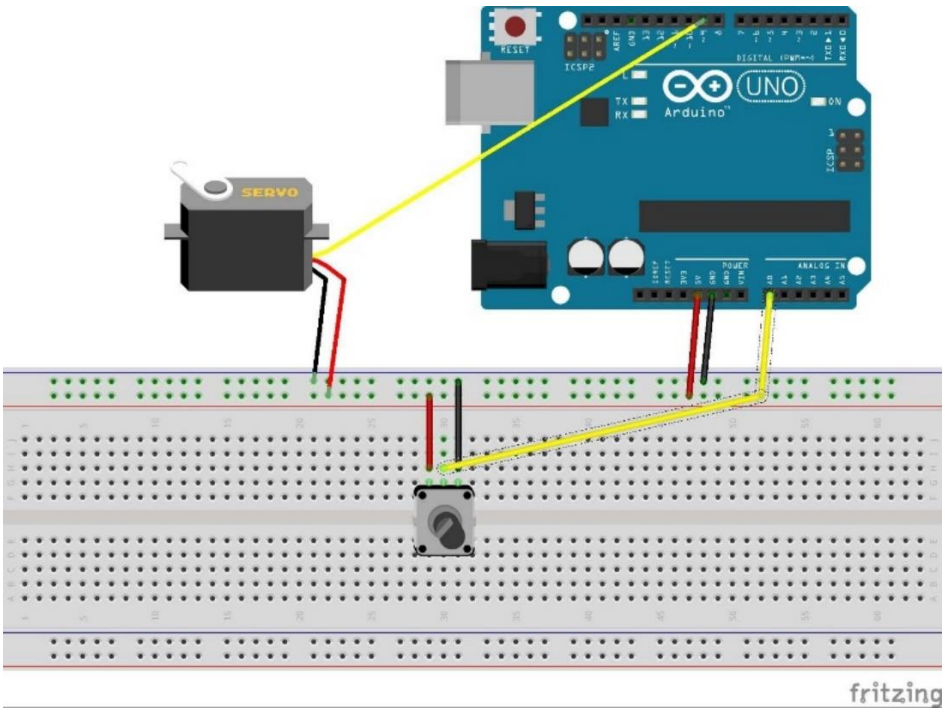
Il existe plusieurs dizaines de référence concernant le servomoteur et notre choix se porte sur le sg90 car il est le parfait compromis entre puissance, poids et dimensions.

Ce servomoteur possède 3 fils (un pour l'alimentation, un pour la masse et l'autre pour lui transmettre des données.

Pour contrôler l'angulation du moteur on va d'abord utiliser un potentiomètre (qui sera transformé en joystick) qui sera branché à la carte et au servomoteur. Ainsi le servomoteur branché sur une sortie recevra l'information du potentiomètre lui-même branché sur une entrée analogique. Nous utiliserons

ensuite un module radiofréquence pour pouvoir contrôler le servomoteur à distance

Le schéma ci-dessous présente les branchements à réaliser.



Moteur pour faire avancer la voiture

Pour faire avancer la voiture, il nous faudra un moteur a courant continue. Comme pour le moteur précédent, il y a plusieurs dizaines de références.

Parmi les types de moteurs à courant continue, on a les moteurs brushless qui sont connues pour atteindre de grandes vitesses comparées aux autres types de moteurs. Il est donc souvent utilisé pour des drones par exemple. Dans le cadre de notre projet on ne cherche pas nécessairement à créer une voiture allant à une très grande vitesse. De plus, ils sont difficiles à contrôler.

Deux moteurs motoréducteurs pourront donc suffire à faire avancer la voiture. Nous avons choisis un moto-réducteur avec une plage de tension élevé (3-9V)

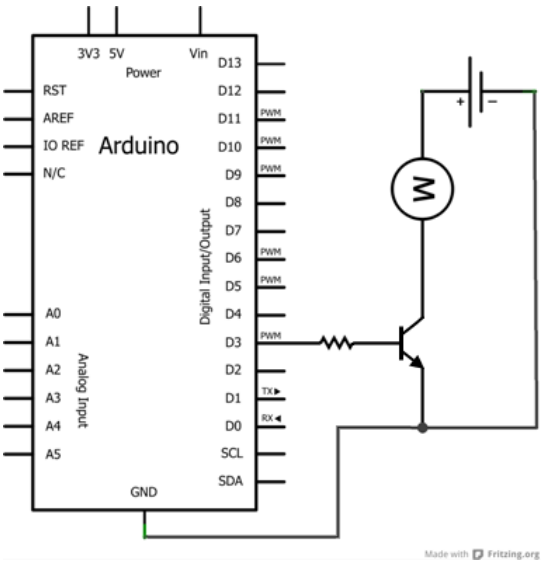
pour une moindre contrainte de cette caractéristique lorsqu'on voudra l'alimenté :

Technical Specifications

Model	Gear Motor including Wheel			
Article No.	COM-Motor01			
Shaft	3.6mm two-sided with opening 1.9mm			
Power Supply	3 - 9V DC (recommended: 4.5V)			
Dimensions (Wheel)	Diameter: 65mm Width: 27mm			
Dimensions (Motor)	37.6 x 64.2 x 22.5mm			
Scope of Delivery	Motor, Wheel			
EAN	4250236815503			
Voltage DC (V)	4.5	6	7.2	9
Current in Idle (mA)	190	160	180	200
Revs per min. in idle (± 10%)	90	190	230	300
Torque (gf/cm)	800	800	1000	1200

Branché le moteur directement à la carte arduino n'est pas envisageable car il risque d'abîmer celle-ci car elle ne délivrera pas un courant assez important par rapport à ce que « demande » le moteur.

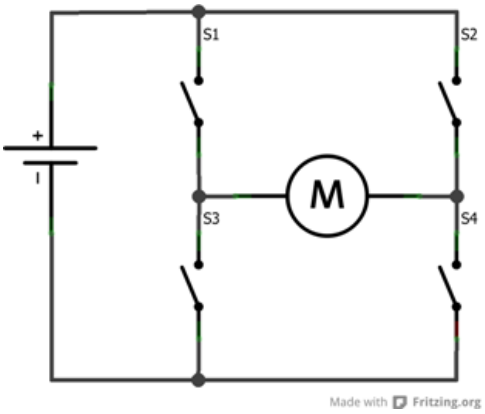
Un transistor permet de régler ce problème car il joue le rôle d'un amplificateur de courant. Il possède 3 fils parmi lesquels ont connecte une sorti de la carte arduino à celui du milieu. Sur les 2 autres fils on connecte le moteur et la masse. Voici donc le schéma de ces branchements :



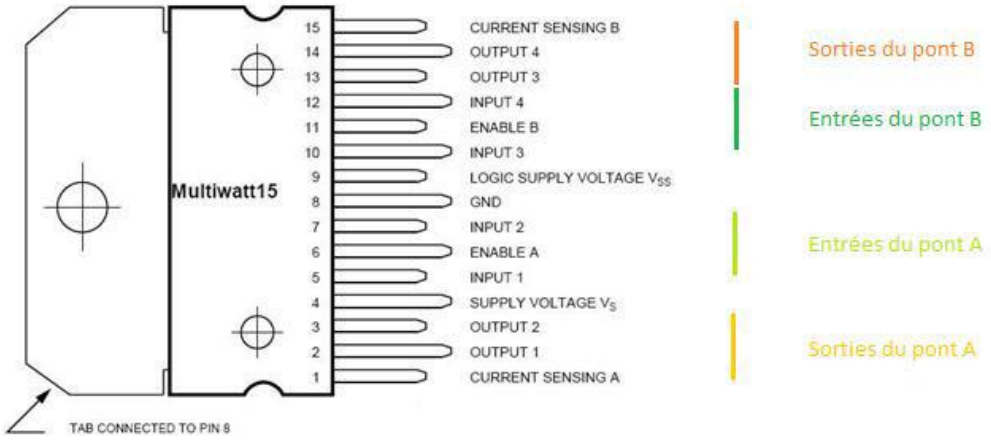
Lorsqu'on coupe l'alimentation, a cause de son inertie, le moteur continue de tourner est créée un champ induit, c'est-à-dire que le moteur créer une tension qui pourrai endommager le transistor. C'est pour cela qu'on installera une diode en parallèle du moteur car celle-ci aura pour effet de garder la tension

induite à proximité du moteur. On ajoute a cela un condensateur qui permettra d'éliminé les courants parasite autour du moteur, il joue en effet un rôle de filtrage.

Pour pouvoir faire reculer la voiture, il faudra utiliser un pont en H, c'est-à-dire placé le moteur au milieu de 4 interrupteur qui permettront d'inversé le sens de rotation du moteur suivant quel interrupteur et ouvert ou fermé. Le schéma ci-dessous représente un point en H.



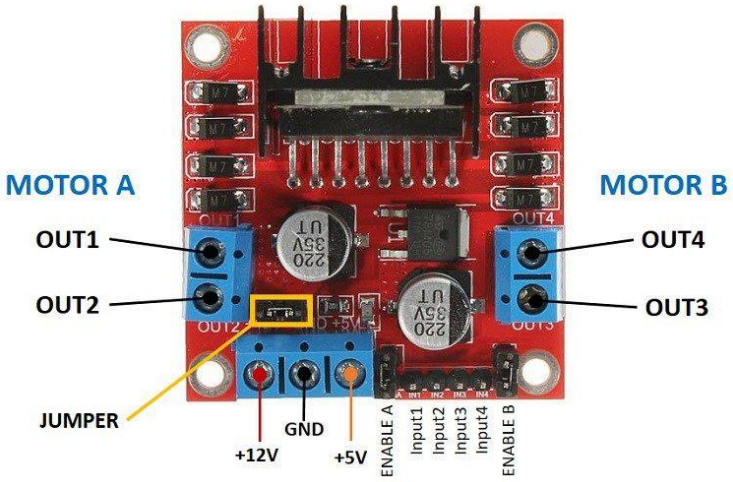
Mais ce point en H peut être réuni en un seul composant appelé L298. Ce composant est capable de délivré un courant de 2A par ponts (il en contient deux) fonctionnant jusqu'à 46V. Le schéma ci-dessous indique les différente entrées et sorties de ce composant.



Notons qu'il existe un autre composant, le L293 possédant un quadri-pont en H (dans notre cas deux suffisent). La principale différence avec le L298 réside dans le fait que le L293 débite nettement moins de courant donc nous

préférons assurer en choisissant le L298 qui est tout aussi abordable au niveau du prix.

Pour simplifier tous ces branchements qui sont assez fastidieux on peut utiliser un circuit intégré L298N contenant tout ce qui a été dit précédemment concernant le moteur a courant continue (dont les diodes permettant de protéger le point en H et le moteur.



Comme le montre le schéma, ce module est capable de contrôler deux moteurs.

Pour alimenter ce moteur, il nous faudra une source d'énergie à savoir piles ou batterie. Les batteries ont l'avantage non-négligeable de pouvoir être rechargeable (sauf pour les batteries Alcaline). Les piles sont pour la plupart à usage unique mais il en existe des rechargeables.

Parmi le type de batteries que l'ont peu utilisé on a :

- Les batteries au plomb. Elles présentent l'avantage d'avoir une excellente durée de vie. Cependant elles ont de nombreux défaut dont leur encombrement important, poids élevé ou encore une autonomie limité.

- Les batteries Ni-mh ont pour principal défaut leur effet mémoire qui consiste en une perte de leur capacité au fil du temps. De plus, la détection de faible charge est plus complexe. Autre défaut, elles se déchargent rapidement si elles ne sont pas utilisées.

- Les batteries Ni-Cad sont proches des Ni-mh. Elles sont cependant moins onéreuses mais sont sujette au phénomène d'auto-décharge.

- Enfin les batteries Lithium sont légères, possèdent une bonne autonomie et n'ont pas d'effet mémoire mais s'usent dans le temps et sont plus onéreuses.

-Pour le fonctionnement des composants à l'intérieur de la voiture, si l'on devait choisir une batterie, ce serait donc la technologie **lithium**. Mais reste maintenant savoir quel est la source d'alimentation la plus adaptée entre des piles rechargeables et une batterie Lithium…

Pour pouvoir alimenter les 2 moteurs qui supportent 9V, nous allons utiliser une tension au-delà des 9V . Il faudra tenir compte de la chute de tension (1 à 2v) sur la carte .Chaque Lithium-ion cellule délivrant 3,7V , nous choisissons d'alimenter le moteur driver L298N par 3 piles rechargeables en série , qui pourront dont fournir 11,1 V . Le moteur driver délivrera le 9V aux moteurs et également la carte Arduino (qui contient un régulateur permettant de délivrer le 5V pour tous les composants internes) . A noter, que la carte Arduino doit être alimenter par du 7v Min et 12v Max sur l'entré VIN. Chaque moteur consommant 300mA (600mA au total), et la carte arduino 50 mA et chaque capteur 15mA (60 mA au total) , le servomoteur 200mA . La consommation totale est donc de l'ordre de 1A , nous recherchons donc 1,5Ah pour avoir une marge . Nous retiendrons celle-ci qui se rapproche le plus :

- Marque : Samsung
- Technologie : Lithium-ion
- Tension : 3,6V
- Capacité minimum : 2,6Ah
- Dimension de l'unité : 65mm (h) - 18,4mm (Ø)

-Pour la partie télécommande , une **pile de 9V classique** suffit

Contrôler les moteurs à distance

Pour pouvoir piloter la voiture a distance, on va devoir utiliser un signal. Pour cela, on va utiliser les radiofréquences a l'aide d'un module radiofréquence et de la bibliothèque VirtualWire.

Plusieurs types de connexion radiofréquence s'offrent alors à nous. En effet nous avons les module Bluetooth, RF 433MHz, wifi , Lora , ou encore ZigBee.

Comme nous souhaitons piloter notre voiture à l'aide d'un joystick les modules Bluetooth et wifi ne sont pas adapté (ils le sont surtout si l'on veut contrôler la voiture grâce aux smartphones). De plus le point faible du Bluetooth est sa courte distance de portée.

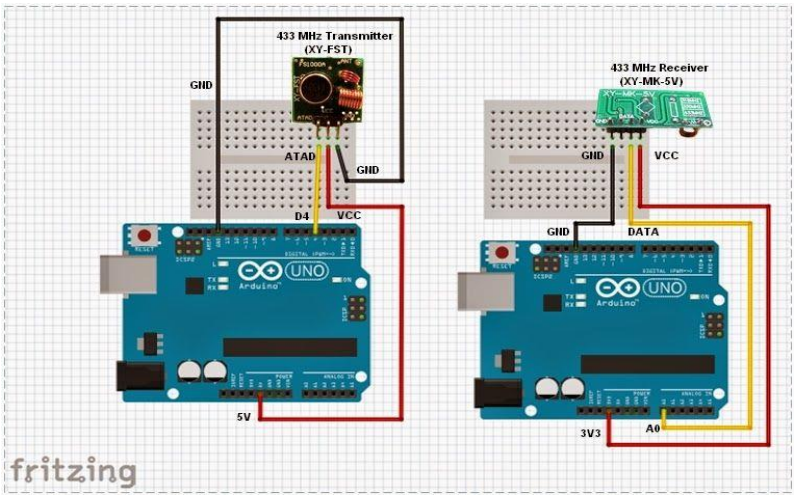
Le module Zigbee quant à lui est peut-être trop performant pour notre modeste voiture radio-commandé. En effet il possède une portée pouvant aller jusqu'à 1000m et une fréquence de 2.4GHz. Il n'est pas adapté à la voiture radiotélécomandé car initialement, ce module permet de créer un réseaux d'objet connecté chez sois par exemple. Il est donc d'une grande utilité dans le domaine de la domotique.

LoRa est un protocole de télécommunication permettant une communication a bas débit avec des objets a faible consommation électrique. Or le moteur de notre voiture aura, à l'inverse une consommation électrique relativement conséquente. C'est pourquoi nous ne choisissons pas ce type de module radiofréquence.

Il nous reste donc le module radiofrequence 433MHz que nous choisirons. L'emetteur possède 3 entrées (Vcc,GND et les données a envoyé). Avec ce module, nous avons la possibilité d'augmenter la distance d'envois en soudant une antenne plus ou moins longue. La tension de fonctionnement de ce module est comprise entre 3 et 12V. La puissance émise par cet émetteur peut aller jusqu'à 25mW et le débit est au maximum de 10kbps.

Le récepteur possède également 3 entrées (Vdd, GND et les données à recevoir) et tout comme l'émetteur il y a possibilité de soudé une antenne. En revanche, l'alimentation doit être de 5V.

Voici les branchements a réaliser dans le cas d'un émetteur (FS100A) et d'un récepteur (XY-MK-5v) :



Comme nous l'avons dit précédemment nous allons utiliser la bibliothèque VirtualWire qui comporte quelques fonctions qui peuvent nous être utile :

- vw_setup() est une fonction essentiel pour initialiser la bibliothèque. Elle prend en argument la vitesse de transmission en bit par seconde. On remarquera que plus la vitesse est lente plus la portée sera élevé et inversement.

- vw_set_tx_pin() qui prend en argument l'entrée de la broche utilisé pour transettre les données. Par défaut, il s'agit de la broche numéro 12.

- vw_set_rx_pin() qui prend en argument l'entrée de la broche utilisé pour recevoir les données. Par défaut, il s'agit de la broche numéro 11.

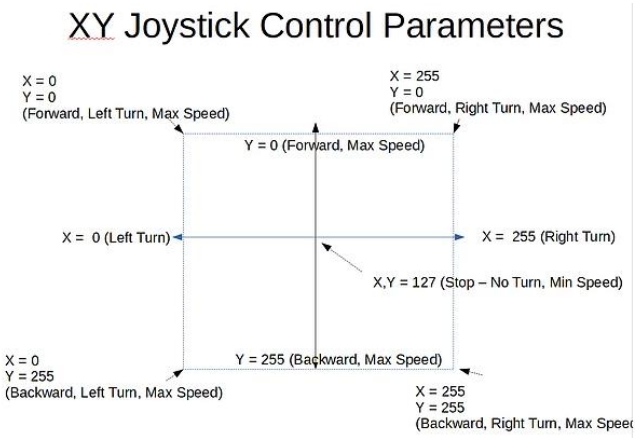
- vw_set_ptt_pin() qui prend en argument la broche utilisé comme broche PTT(« Press to talk »), c'est-à-dire un système similaire au talkie-walkie avec un émission de signaux seulement lorsqu'un bouton est appuyer. La broche PTT simule donc l'appui de ce bouton avant de transmettre des données.

- vw_rx_start() qui déclenche le processus de réception du signal.

- vw_rx_stop() qui arrête le processus de réception du signal. On ne recevra donc plus de message.

Utilisation du Joystick :

La position Y contrôle la direction (vers l'avant ou vers l'arrière) et la vitesse du mouvement de la petite voiture. La position X contrôle la rotation de la petite voiture pour ne pas tourner, tourner à gauche ou à droite.



Empêcher les collisions et « self parking »

Pour empêcher les collisions et faire en sorte que la voiture se gare seule du côté gauche, nous allons avoir recours à l'utilisation de 4 capteurs HC-SR04 :

1. À l'avant
2. À l'arrière
3. 2 sur le côté gauche (parking à gauche)

Les caractéristiques techniques du module sont les suivantes :

– Alimentation : 5v.

– Consommation en utilisation : 15 mA.

– Gamme de distance : 2 cm à 5 m.

– Résolution : 0.3 cm.

– Angle de mesure : < 15°.

Il existe d'autres référence comme le SRF05 ressemblant beaucoup au HC-SR04 mais qui paraît cependant plus cher.

Système anticollision : Dés qu'un obstacle se situe à moins de 3 cm (à définir) de l'avant ou de l'arrière , la voiture s'arrête automatiquement (La télécommande est désactivée)

Système self parking : Uniquement sur le **côté gauche**

Lorsque la voiture recherche une place de parking, elle recherche en fait une place de parking de plus de 10 cm supérieur à la longueur de la voiture par exemple (à définir). S'il voit un espace inférieur à 10 cm, elle continuera à rechercher un espace de stationnement qui convient.

Enfin, 2 cartes Arduino Uno (une pour la voiture , une pour la télécommande) seront suffisantes.