Projet : mini robot suiveur de ligne :

KONG Alexandre
Professeur : M. Jean Louis SALVAT
Année 2015-2016

SOMMAIRE

I- Introduction

Dans le cadre de notre projet domotique nous avons dû travailler sur des projets industriels proposé par notre professeur M. Jean-Louis Salvat. Le projet décrit dans ce compte rendue est le projet robot suiveur de ligne. Le but de ce projet est la conception et la programmation d'un robot suiveur de ligne avec des fonctionnalité définis dans le cahier des charges ci-apreès.

Cahier des charges :

- Testes des différents éléments
- Fonctionnement du robot suivant direction
- Fonctionnement robot avec commande Bluetooth
- Scan infrarouge (détection d'obstacle)

Matériel disponible :

3 US + support à placer à l'avant
2 servo + support tourelle à placer sur la plateforme
1 boussole GY271
1 arduino nano à tester
1 arduino uno
1 capteur infrarouge 4x4 (température)
1 bouton rotatif + LCD nokia5110

Objectifs atteint :

- Testes de :

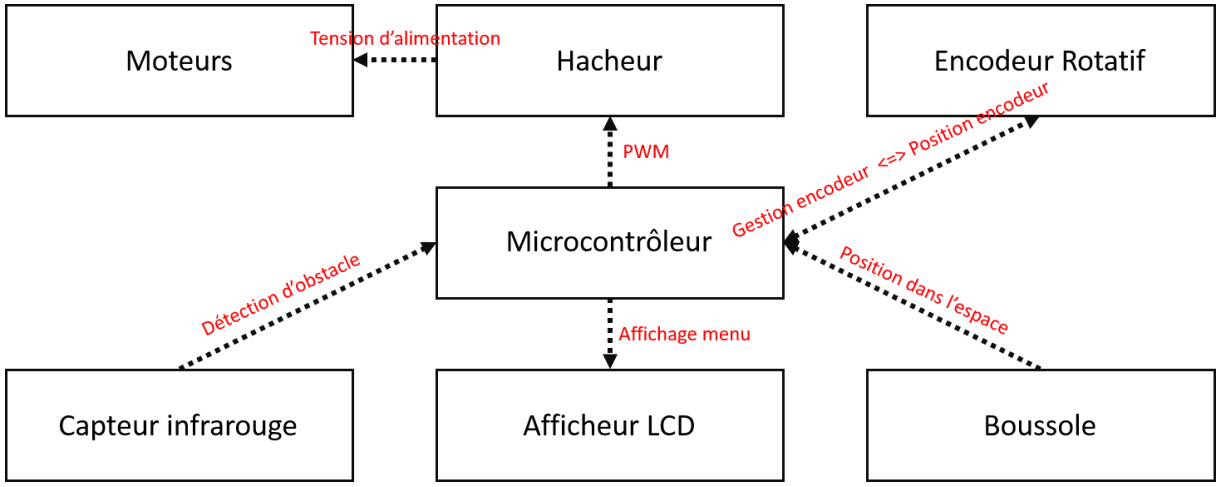
Arduino Uno	=> OK
Capteur IR	=> OK
Hacheur	=> OK
Encodeur rotatif	=> OK
Ecran LCD	=> OK
Capteur ultrason	=> NON
Servo moteur	=> NON
Boussole	=> NON

- Fonctionnement du robot suivant direction :
  Contrôle de la vitesse et le sens du moteur => OK

- Fonctionnement robot avec commande Bluetooth => NON

- Scan infrarouge :
  Détection d'obstacle => OK

Synoptique :



Le microcontrôleur gère la partie traitement.

Le capteur infrarouge envoie au microcontrôleur une valeur analogique en fonction de cette valeur le micro en déduira-s'il y à détection d'obstacle.

L'afficheur LCD affiche le menu afin de configurer la vitesse et le sens de rotation des moteurs.

La boussole indique la position du robot dans l'espace.

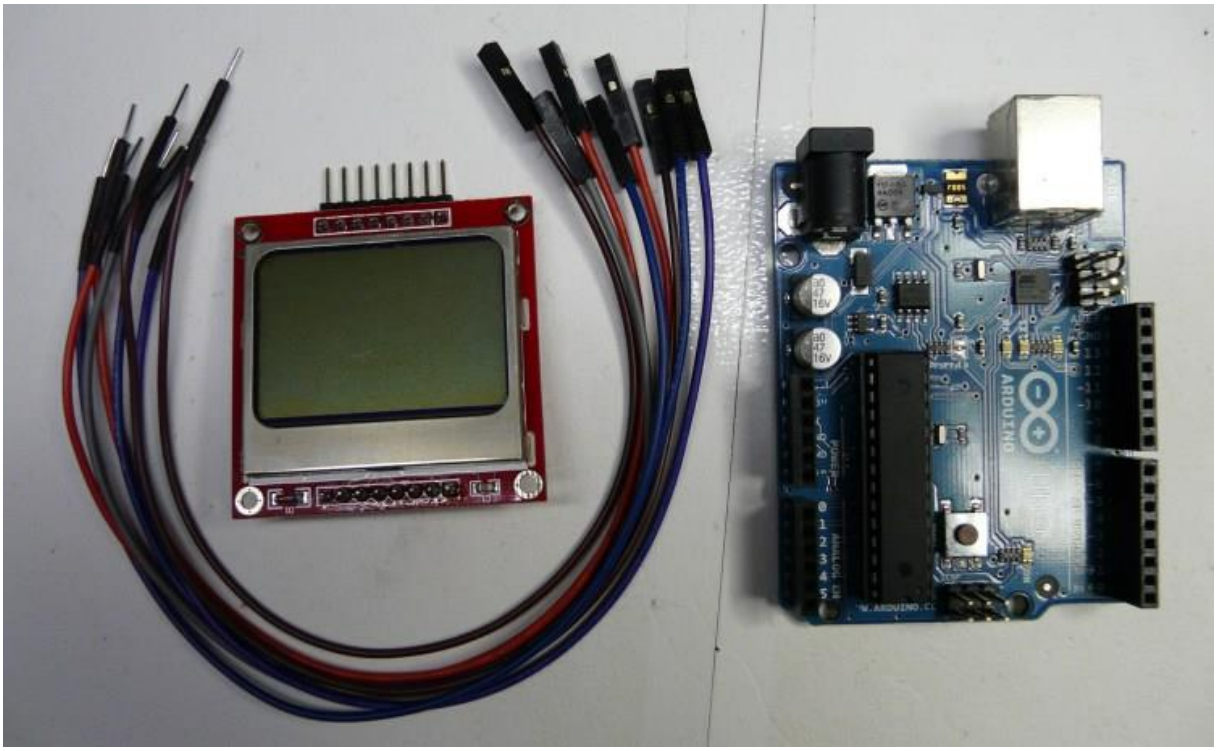
Le hacheur délivre une tension au moteur en fonction du signal pwm envoyé par le micro

L'encodeur permet le réglage manuel de la vitesse et du sens de rotation.

II- Travail réalisé

Dans un premier temps je devais tester les différent composant que l'on m'avais fournis, les tests ci-après sont les tests que j'ai pu effectuer.

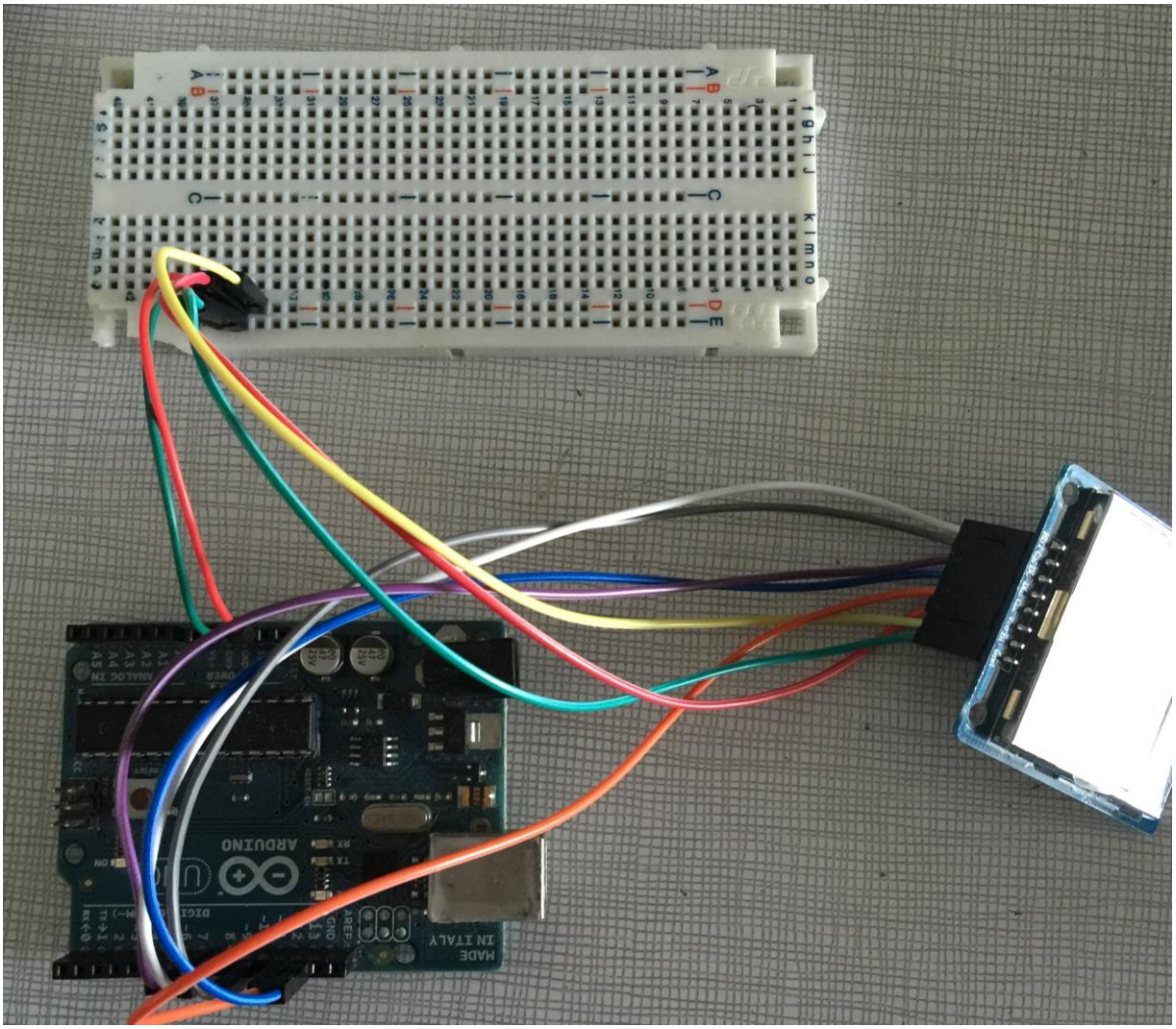
Teste écran LCD Nokia 5110 :



Ecran LCD / Arduino Uno.

Câblage :

LCD	ARDUINO UNO
RST	Pin 8
CE	Pin 10
DC	Pin 7
Din	Pin 13
Clk	Pin 12
Bl	+ 5V
Vcc	+ 5V
Gnd	Gnd



Le but de ce composant dans notre projet sera d'afficher un menu pour notre robot sur l'écran. Au niveau du programme dans un premier temps, je déclare une table ASCII afin de déclarer tous les caractères affichables sur l'écran. Pour ce faire j'ai créé une fonction 'LcdCaracter'

Ensuite il me faut une fonction permettant de réinitialiser l'écran ('LcdClear')

La fonction 'LcdInitialise' me permet d'initialiser tous les entrées de l'écran c'est également dans cette fonction que je peux modifier le contraste.

```
LcdWrite(LCD_C, 0x21 );  // LCD Extended Commands.
LcdWrite(LCD_C, 0xBf );  // Set LCD Vop (Contrast).
LcdWrite(LCD_C, 0x04 );  // Set Temp coefficent. //0x04
LcdWrite(LCD_C, 0x14 );  // LCD bias mode 1:48. //0x13
LcdWrite(LCD_C, 0x0C );  // LCD in normal mode.
LcdWrite(LCD_C, 0x20 );
LcdWrite(LCD_C, 0x0C );
```

Extrait du code de teste de l'écran LCD.

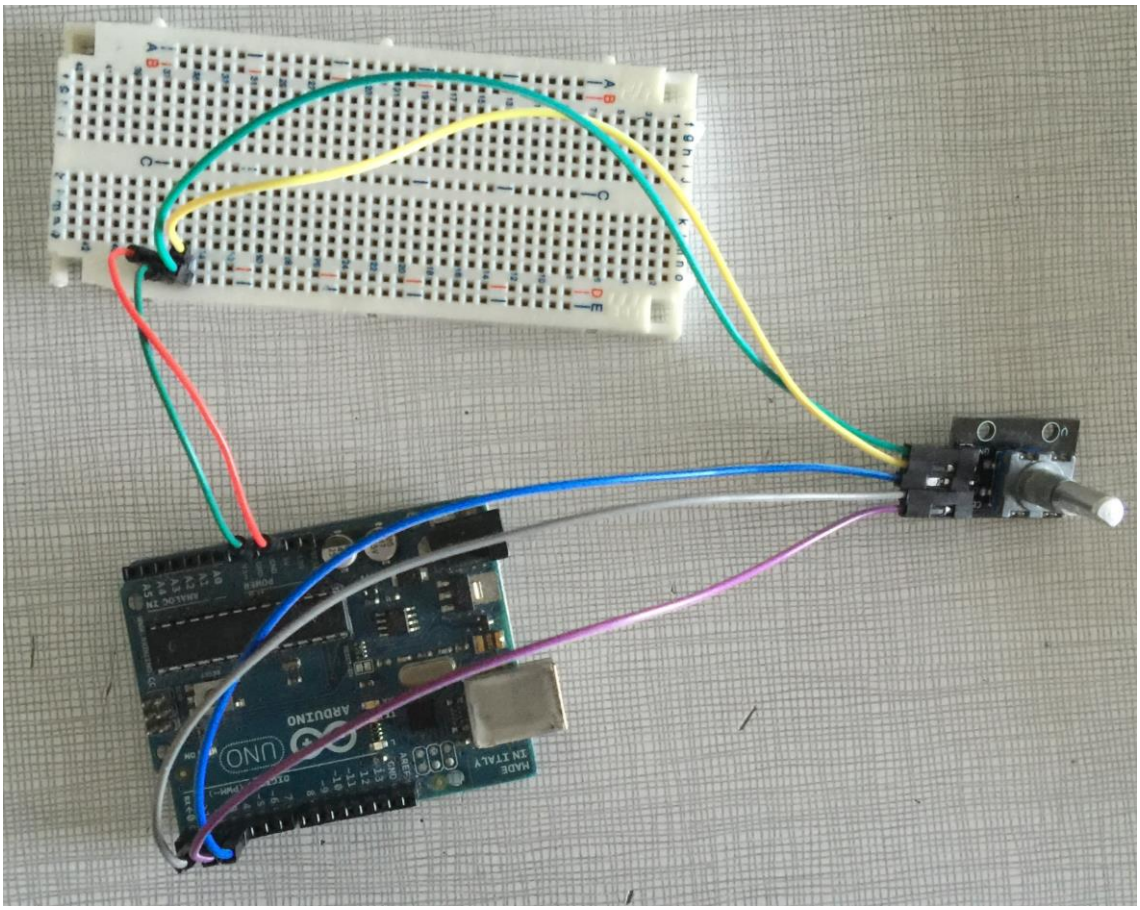
Quant à la fonction ‘LcdString’, elle permet d’écrire des mots elle reprend la fonction ‘LcdCharacter’ elle la mette dans un tableau de ‘char’.

Teste encodeur rotatif :

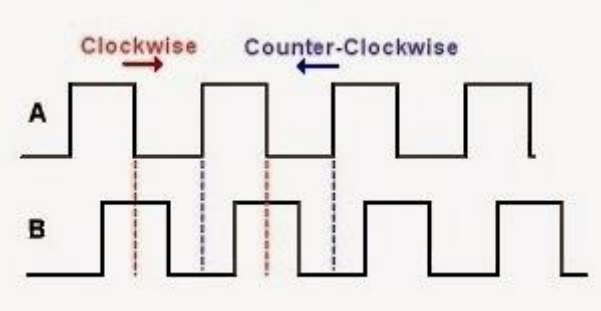
Encodeur rotatif.

Câblage :

Encodeur rotatif	ARDUINO UNO
CLK	Pin 3
DT	Pin 2
SW	Pin 4
Vcc	+ 5V
Gnd	Gnd



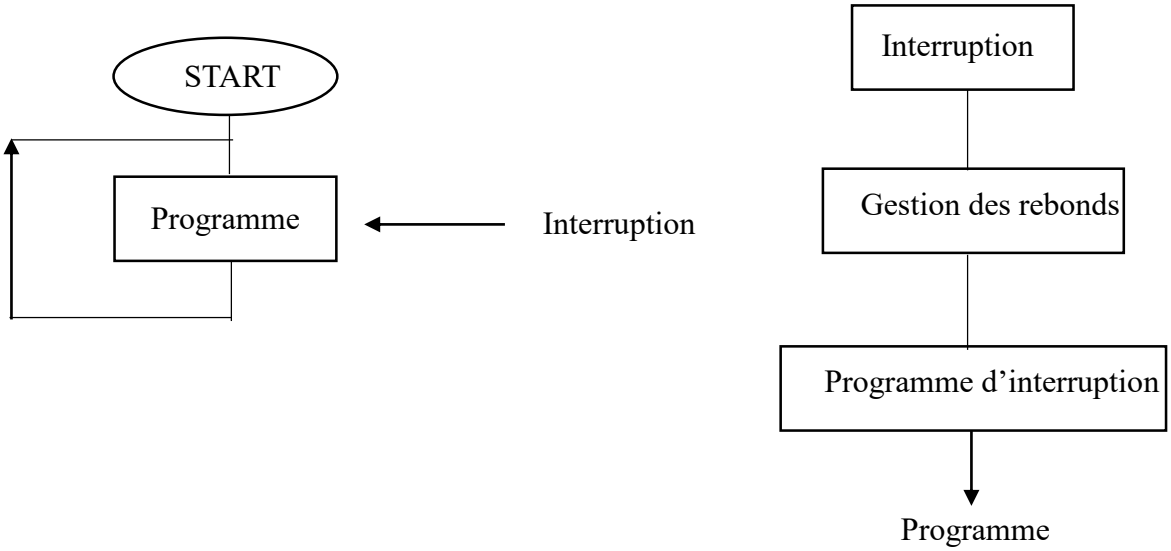
Le but de ce composant est commander le menu, pour cela il faut que l'on puisse récupérer les positions de l'encodeur et détecter l'appui du bouton.



Logigramme de l'encodeur rotatif.

Pour la gestion de la position de l'encodeur j'ai utilisé des interruptions disponibles sur Arduino Uno (interruption 0 broche 2 et interruption 1 broche 3).

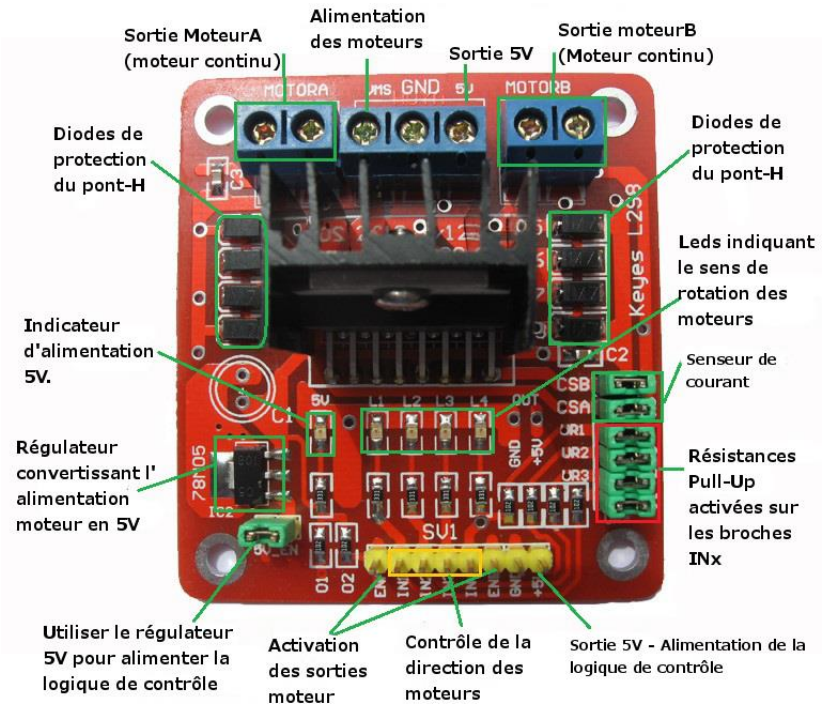
Lorsqu'une interruption survienne soit sur la broche 2 soit sur la broche 3 (sens horaire ou antihoraire) le programme renvoie à une fonction de gestion des rebond cette fonction prend en compte la première interruption et déclenche une temporisation qui me permet d'ignorer tous les autres interruptions durant ce court laps de temps.



Organigramme du programme d'interruption.

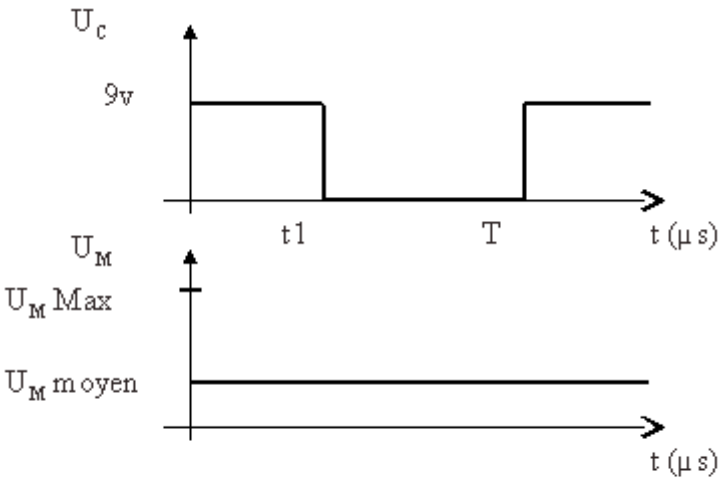
Pour la détection de l'appui je commence dans un premier temps à lire l'état du bouton ensuite je compare cette état a sont état précédent si un changement a eu lieu cela veut dire qu'il y a eu un front (un appui).

Teste Hacheur L298N :



Hacheur.

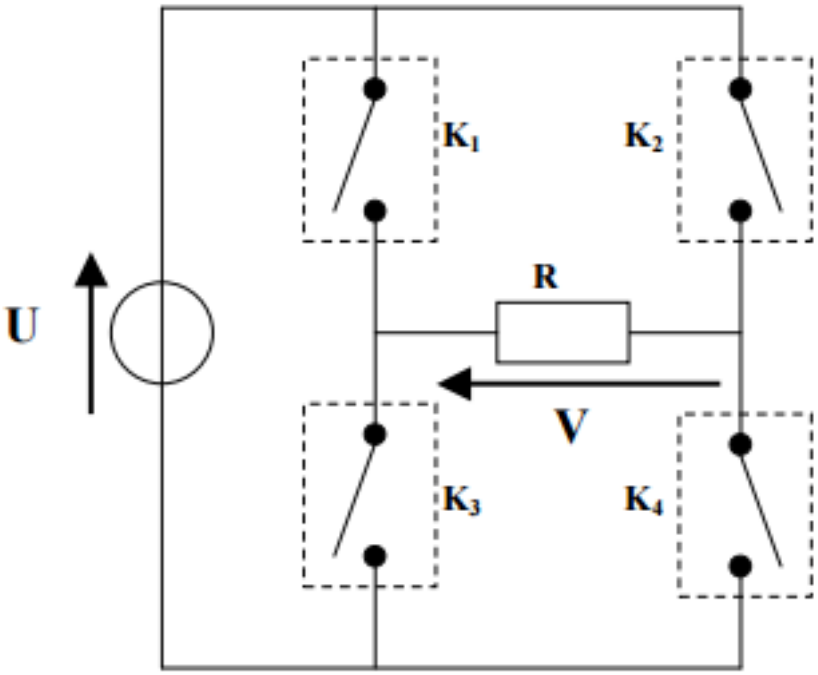
Calcule de la tension moyenne avec le rapport cyclique.



α : Rapport cyclique

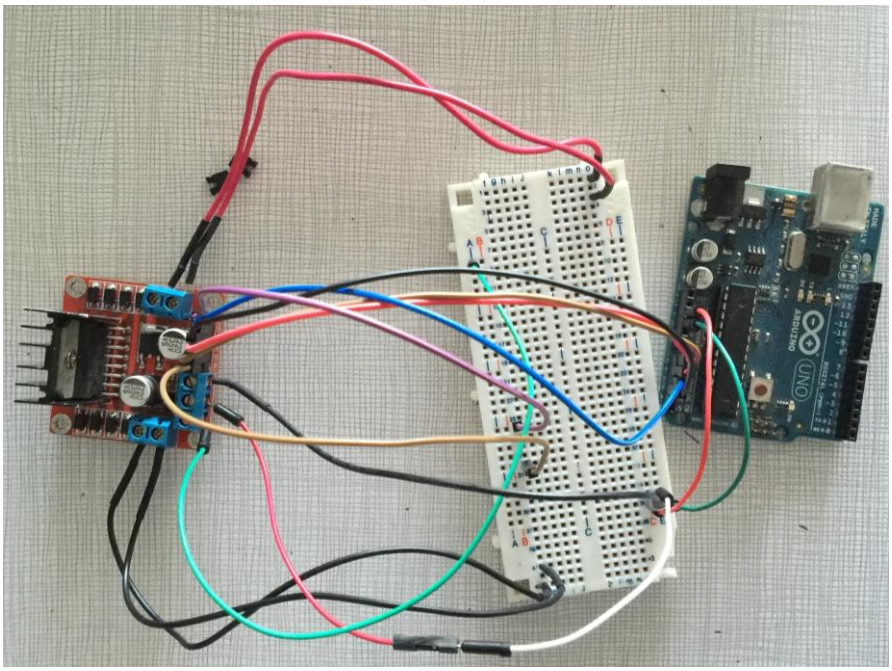
$$U_M \text{ moyen} = U_M \text{ Max} \times \alpha$$

Schéma d'un hacheur



Câblage :

Hacheur	ARDUINO UNO
ENA(signal pwm)	Pin 5
ENB(signal pwm)	Pin 6
IN1	A0
IN2	A1
IN3	A3
IN4	A2
OUT 1	MOTEUR DC 1 +
OUT 2	MOTEUR DC 1 -
OUT 3	MOTEUR DC 2 +
OUT 4	MOTEUR DC 2 -
+ 5V	+ 5V
+ 12V	+12V
Gnd	Gnd



Le but de ce composant est de générer un signal PWM afin de commander la vitesse et le sens de 2 moteurs DC.

Le hacheur possède 2 broches qui peux recevoir un signale PWM (Pulse Width Modulation) et deux broches permettant de configurer le sens de rotation d'un moteur.
Voici la configuration afin de configurer les sens de marche possible des moteurs :

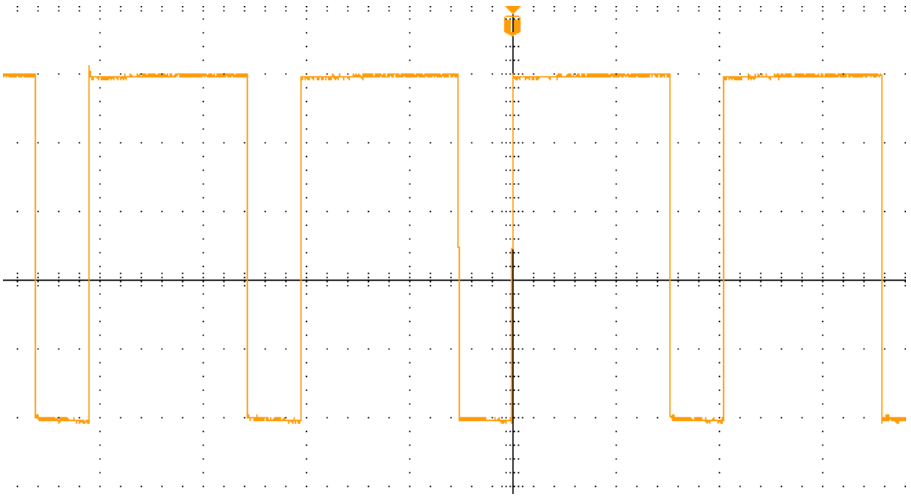
IN1(IN3) = HIGH et IN2(IN4) LOW sens horaire.
IN1(IN3) = LOW et IN2(IN4) HIGH sens horaire.

Pour la partie programmation on utilisera les fonction 'analogRead()' ou 'digitalRead()'
Voici les relevées à l'oscilloscope du programme de teste pour un PWM à 25%, à 50% et à 100% dans un sens.

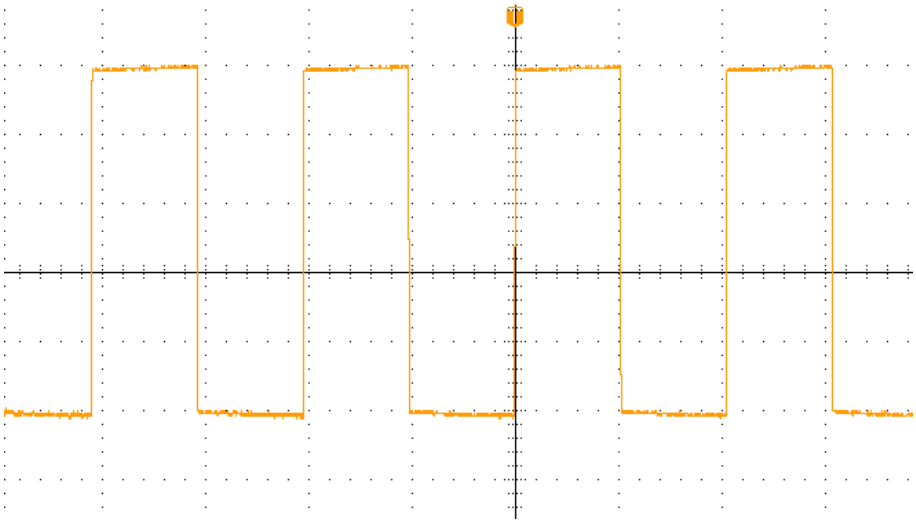
Relevé :

Teste avec des moteur en mode recule.

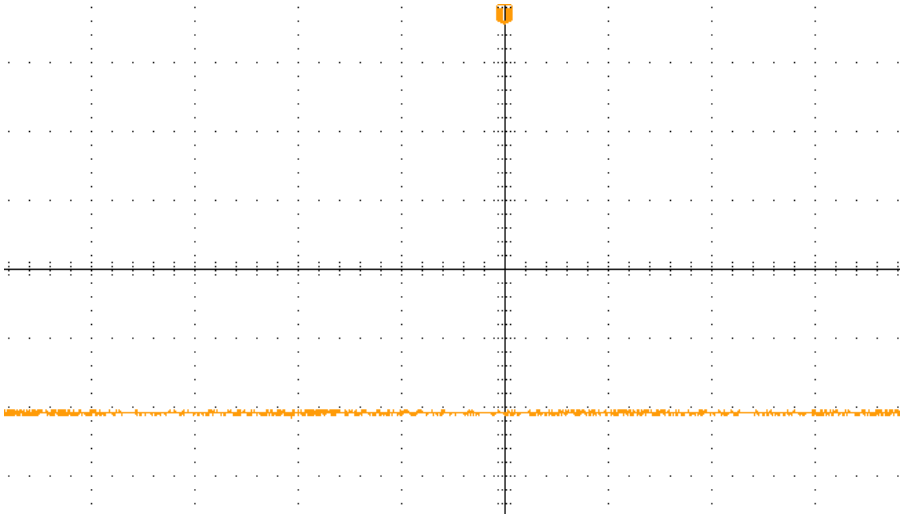
Pwm 25% :



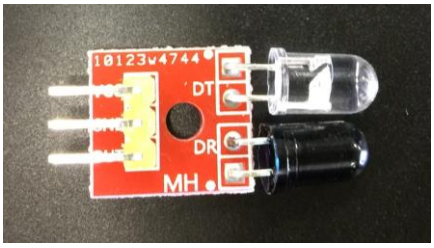
Pwm 50% :



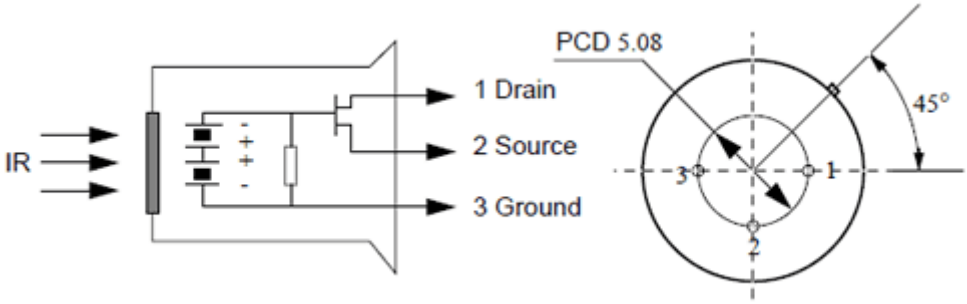
Pwm 100% :



Teste Capteur infrarouge :



Capteur IR.



Câblage :

LCD	ARDUINO UNO
OUT	A5
+ 5V	+ 5V
Gnd	Gnd

Le but de ce composant est de détecté la présence d'un obstacle.

Le capteur délivre un signal analogique. Pour récupérer la valeur que le capteur envoie j'utilise la fonction 'analogRead' et je l'affiche dans un premier temps sur la console du logiciel Arduino. Lors de mes teste j'ai relevé que le capteur envoie une valeur inferieur a 100 quand elle détecte un obstacle. De ce fait dans mon programme teste j'affiche simplement « obstacle » quand le capteur délivre un signale inferieur a la valeur 100.

Relever :

COM3 (Arduino/Genuino Uno)

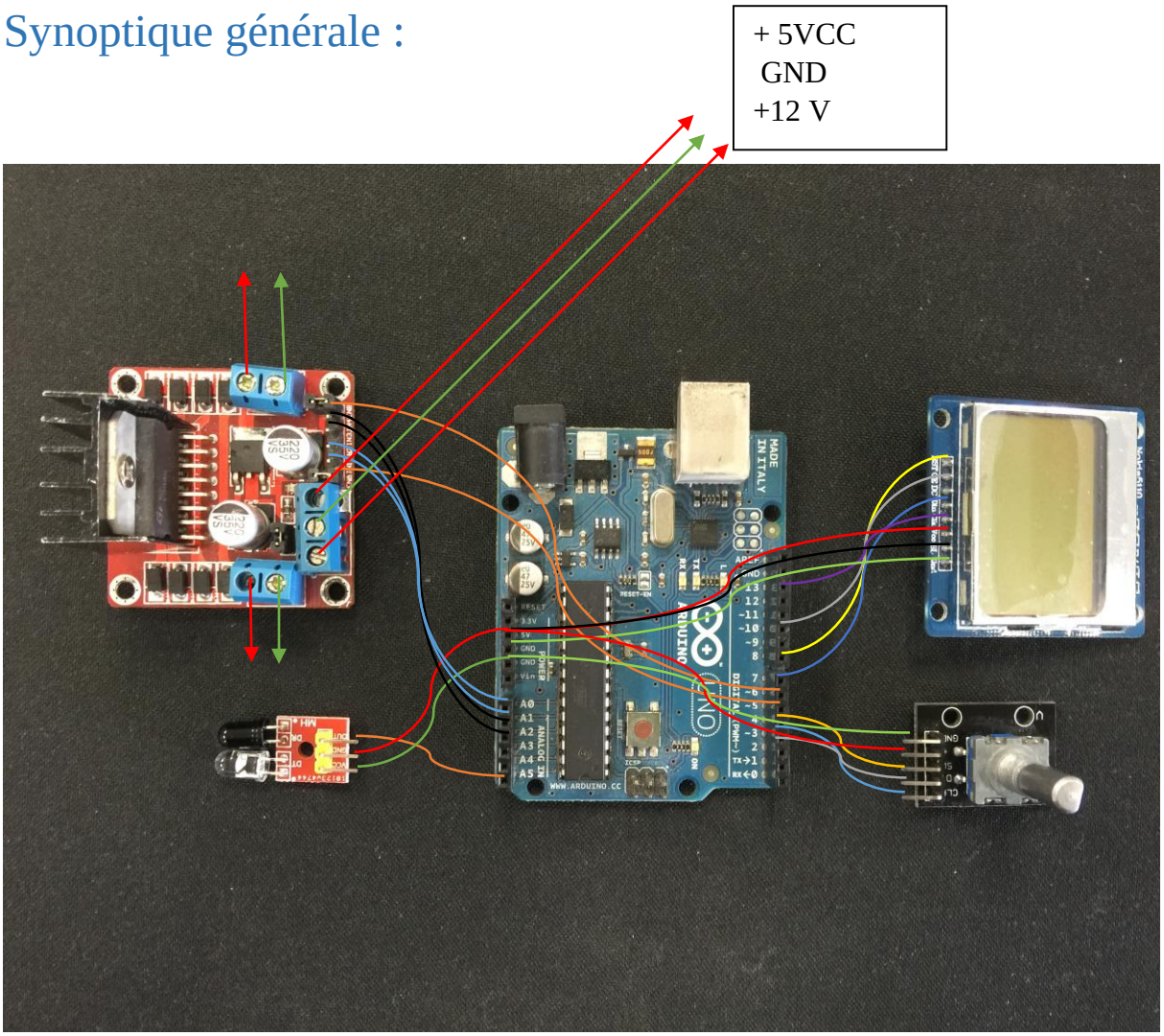
```
423
Pas Obstacle
463
Pas Obstacle
466
Pas Obstacle
465
Pas Obstacle
478
Pas Obstacle
477
Pas Obstacle
475
Pas Obstacle
466
Pas Obstacle
461
Pas Obstacle
469
Pas Obstacle
470
Pas Obstacle
449
Pas Obstacle
5
Obstacle
4
Obstacle
```

Teste d’application :

Nous allons maintenant mettre en application tous les composant testée précédemment, nous allons afficher sur l’écran un menu qui permet de configurer la vitesse et le sens de rotation des deux moteurs après sélection des paramètres. Les informations saisies seront traitées par notre microcontrôleur et envoyé au hacheur afin de permettre la commande des moteurs. Lorsque le capteur IR détecte un obstacle les moteurs dois cesser de fonctionner jusqu’à la prochaine consigne.

Affichage menu sur écran.

Synoptique générale :



III- Conclusion

Durant ce projet j'ai pu tester différent composant que je ne connaissais pas, j'ai apprécier le fait de programmer sur Arduino, je trouve qu'il est très accessible et simple d'utilisation. Cependant je n'ai malheureusement pas eu le temps de tester tous les composant et donc de terminer mon projet. J'ai également pu apprendre à utiliser GitHub un outils que je ne connaissais pas.

Source :

Tous les programmes se trouve dans la branche programme de mon GitHub

https://github.com/alexandrekong/LPAII-Domotique/tree/Programe

Vidéo teste de l'application :

https://www.youtube.com/watch?v=6NvYZ5g_OBw