INSTITUT d'OPTIQUE GRADUATE SCHOOL | LEnsE Laboratoire d'Enseignement Expérimental

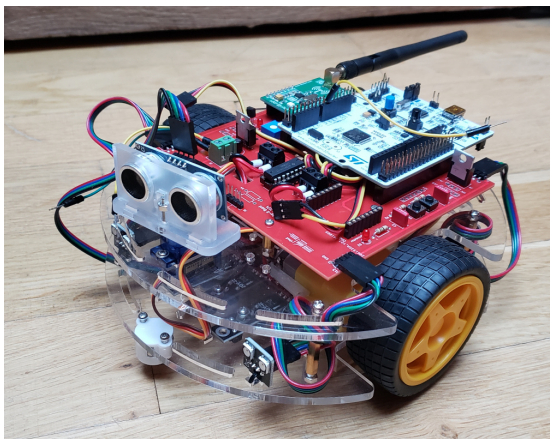
Institut d'Optique Graduate School
INTERFAÇAGE NUMÉRIQUE

Travaux Pratiques

Semestre 6

Robotique et systèmes embarqués

4 séances



Ce sujet est disponible au format électronique sur le site du LEnsE - https ://lense.institutoptique.fr/ dans la rubrique Année / Première Année / Interfaçage Numérique S6 / Bloc 1 Systèmes embarqués.

© 2027 by LEnsE-IOGS

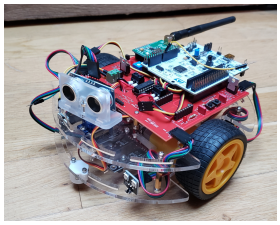
Robotique et systèmes embarqués

À l'issue des séances de TP concernant le **bloc de robotique**, les étudiant·es seront capables de :

— Développer et mettre en œuvre une **solution d'électronique embarquée** pour **mettre en mouvement un robot**
— Concevoir un **programme embarqué** permettant de **rendre autonome les mouvements d'un robot**

Objectifs du mini-projet

L'objectif principal de ce mini-projet est de **développer le code embarqué d'une plateforme robotique** lui permettant de se déplacer de manière autonome le long d'une ligne sans percuter d'obstacle.



Vous aurez à votre disposition une **maquette** basée sur un robot Joy-It Car. Cette maquette est pilotée par une carte Nucléo (contenant un microcontroleur).

Un exemple de base (en version archivée et pré-compilée) est disponible au téléchargement ici :

https://lense.institutoptique.fr/ressources/mbed/hello-L476RG_pre.zip

Afin de faciliter la réutilisation des codes, il pourra être intéressant de définir des fonctions pour le pilotage des différents éléments.

Déroulement du bloc

La liste des étapes à suivre pour la réalisation du programme embarqué de la plate-forme robotique est donnée à titre indicatif. L'ordre et le choix des différentes étapes sont laissés à l'appréciation des différents binômes.

Séance 1 / MBED et Nucléo-STM32 (sans maquette !!)

Le sujet de cette séance est fourni dans un document annexe, disponible aussi sur le site du LEnsE - https ://lense.institutoptique.fr/ dans la rubrique Année / Première Année / Interfaçage Numérique S6 / Bloc 1 Systèmes embarqués / Intro MBED et STM32.

Séance 2 / Maquette et déplacements élémentaires

Etape 7 - 60 min Piloter l'intensité des LEDs de la maquette
Etape 8 - 90 min Piloter les moteurs à courant continu
Etape 9 - 60 min Acquérir des données des capteurs de ligne
Etape 10 - 60 min Suivre une ligne (premiers essais)

Séances 3 et 4 - Asservissement de position et évitement d'obstacles

Ces deux dernières séances seront consacrées au pilotage du robot pour lui permettre de **suivre une ligne** et de **contourner des obstacles** qu'il rencontre sur son chemin (en intégrant un asservissement de position pour contrôler ses mouvements lors du contournement d'un obstacle).

Les étapes possibles sont les suivantes :

Etape 11 - 90 min Suivre une ligne (validation)
Etape 12 - 60 min Acquérir les signaux des codeurs
Etape 13 - 90 min Asservir en position
Etape 14 - 60 min Acquérir les signaux du capteur ultrason
Etape 15 - 90 min Contourner un obstacle
Etape 16 - 120 min Suivre une ligne et éviter des obstacles

DISC / Accumulation de preuves

Nous vous rappelons également que l'un des livrables final de ce module est un **Document Individuel de Suivi de Compétences** qui devra revendiquer votre progression dans l'acquisition des compétences suivantes :

- C4 – Valider une solution technologique
- C7 – Travailler en équipe

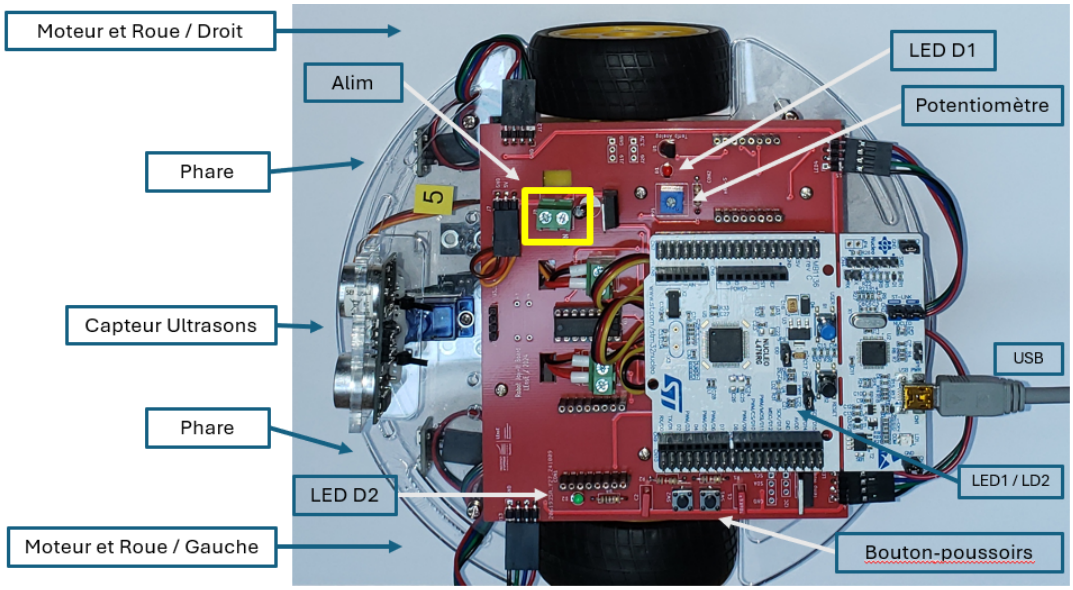
Travail à réaliser

Vous devrez donc tout au long des séances de TP :

- Documenter les tests réalisés pour valider les fonctionnalités mise en oeuvre
- Stocker les preuves pour revendiquer la montée en compétences (résultats, photos, vidéos...)

Séance 2 / Maquette et déplacements élémentaires

Description de la maquette



La maquette est basée sur une plateforme robotique **Joy-It** *Joy-Car* (joy-it.net/en/products/mb-joy-car) constituée :

- de **deux moteurs à courant continu** indépendants, contrôlés par un driver de type L293DN - tension maximale $7\,\mathrm{V}$ associés à deux **encodeurs de position** ;
- de **trois capteurs de ligne** (en dessous du robot) ;
- d'un **capteur à ultrason** (obstacles) monté sur un servomoteur ;
- d'une **carte de contrôle**, composée :
  - de deux bouton-poussoirs ;
  - de deux LEDs standard ;
  - d'un potentiomètre ;
  - d'une carte Nucléo L476RG.

Un schéma de la carte et le brochage des différents éléments sont donnés en annexe de ce document.

Alimentation électrique

La plateforme robotique est utilisable dans deux configurations différentes :

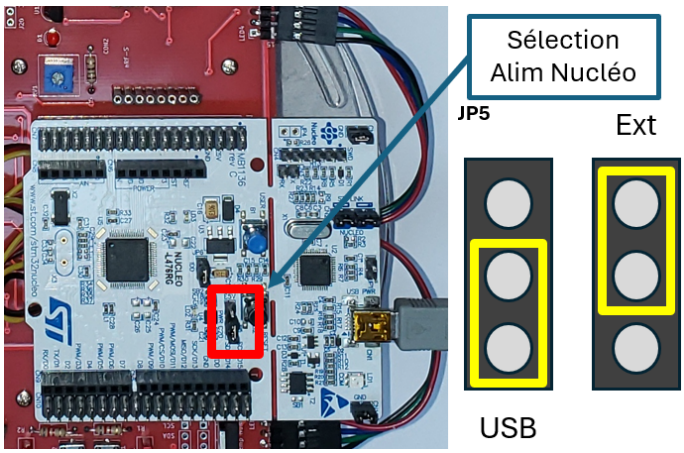
- **non-autonome**, reliée en USB, lors de la phase de programmation de la carte de commande ;
- **autonome**, sur batterie ou alimentation externe.

Carte Nucléo

La carte Nucléo ainsi que la carte de commande (incluant les phares et les capteurs du robot) peut être **alimentée à l'aide du câble USB** (permettant également la programmation du microcontroleur embarqué sur la carte Nucléo).

Attention ! Dans ce régime de fonctionnement, les moteurs et servomoteur ne sont pas utilisables (car non alimentés).

Le cavalier **JP5** de la carte Nucléo doit être positionné du côté **U5V**.



Attention ! Lors de la programmation de la carte Nucléo, il est préconisé de **ne pas utiliser la partie puissance** !

Utilisation de la partie puissance (motorisation)

La tension maximale admissible par les moteurs est de $7\,\mathrm{V}$! (tension minimale de $6\,\mathrm{V}$)

L'alimentation de cette partie se fait à l'aide du connecteur **J8**. La broche **VIN** correspond à la tension positive (comprise entre 5 et 7V) et l'autre broche est reliée à la masse du système.

Attention ! Il est conseillé de tester la motorisation du robot à l'aide d'une alimentation stabilisée - et surtout protégée en courant ! - avant de passer à l'utilisation totalement autonome sur batterie.

Attention ! Pour utiliser la partie puissance, il est préconisé de passer la **carte Nucléo en utilisation autonome** (sur alimentation externe) et ainsi supprimer le lien par le câble USB.

Pour cela, il faut placer le cavalier **JP5** de la carte Nucléo du côté **E5V** (E = externe).

Brochage / Robot Joy-It Car

Maquette	**Nucléo**	Type	Description
LED1	**PC_7**	Sortie / PWM	Led active à l'**état bas**
LED2	**PB_13**	Sortie / PWM	Led active à l'**état haut**
SW1	**PA_11**	Entrée	Bouton-poussoir, par défaut état bas
SW2	**PA_12**	Entrée	Bouton-poussoir, par défaut état bas
USERBUTTON	**PC_13**	Entrée	Bouton-poussoir, par défaut état haut
POT_OUT	**PC_3**	Entrée analogique	Potentiomètre

Moteurs

Maquette	**Nucléo**	Type	Description
MOT_EN	**PA_9**	Sortie	Validation des moteurs
MOT_L_1	**PB_4**	Sortie / PWM	Moteur Gauche direction 1
MOT_L_2	**PA_8**	Sortie / PWM	Moteur Gauche direction 2
MOT_R_1	**PA_0**	Sortie / PWM	Moteur Droit direction 1
MOT_R_2	**PA_1**	Sortie / PWM	Moteur Droit direction 2

Capteurs

Maquette	**Nucléo**	Type	Description
LINE_L	**PA_7**	Entrée	Capteur de ligne Gauche
LINE_C	**PB_6**	Entrée	Capteur de ligne Centre
LINE_R	**PB_12**	Entrée	Capteur de ligne Droit
SPEED_L	**PC_9**	Entrée	Vitesse moteur Gauche
SPEED_R	**PC_8**	Entrée	Vitesse moteur Droit
US_TRIG	**PB_5**	Sortie	Capteur ultrason - Trig
US_ECHO	**PB_3**	Entrée	Capteur ultrason - Echo
SERVO	**PB_7**	Sortie / PWM	Servomoteur du capteur Ultrason

Attention : PA_12 et PB_12 ne peuvent pas être utilisées en même temps en entrée d'interruption.

La tension d'alimentation externe doit être comprise entre $6\,\mathrm{V}$ et $7\,\mathrm{V}$

Phares NeoPixel / SW2812

Maquette	**Nucléo**	Type	Description
DIN_1	PC_0	Sortie	Phare avant droit
DIN_2	PA_10	Sortie	Phare avant gauche
DIN_3	PC_5	Sortie	Phare arrière gauche
DIN_4	PA_13	Sortie	Phare arrière droit
TEMP_OUT	PC_2	Entrée analogique	Capteur de température MCP9700

Un **exemple de base** (en version archivée et pré-compilée - pour Keil uVision) est disponible au téléchargement ici :

https://lense.institutoptique.fr/ressources/mbed/hello-L476RG_pre.zip

Etape 7 / Piloter l'intensité des LEDs (maquette)

➜ M A partir de cet exemple, réaliser un programme qui permet de **modifier la luminosité des diodes LED1 et LED2** de la maquette à l'aide des deux bouton-poussoirs SW1 et SW2 (un pour augmenter, l'autre pour diminuer la luminosité). **Votre programme devra utiliser uniquement le principe d'interruption.**

Vous pourrez vous inspirer des programmes que vous avez réalisés lors de la séance 1.

Etape 8 / Piloter les moteurs à courant continu

Un **moteur à courant continu** est un système permettant de **transformer une énergie électrique en mouvement de rotation** continue. Ce type de moteur doit être parcouru par un courant continu.

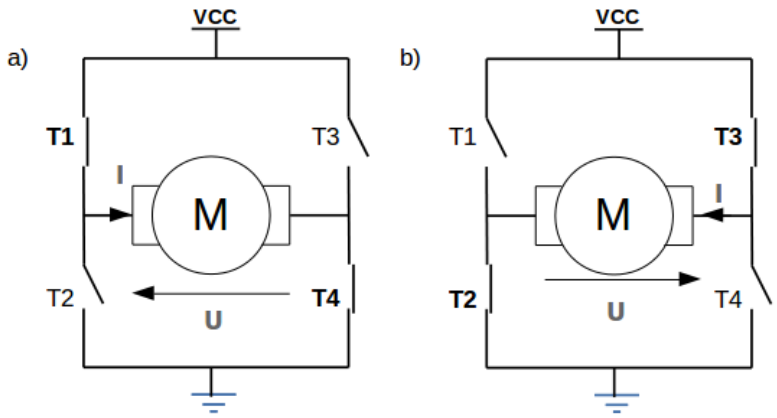
La **vitesse de rotation** d'un tel moteur est liée à la **tension d'alimentation** de ce dernier.

Conversion de puissance

Ces composants de puissance nécessitent souvent des alimentations externes pour fonctionner et ne peuvent pas être directement alimentés par des microcontroleurs (tel que celui présent sur la carte Nucléo). Il est donc indispensable d'ajouter des **étages de conversion de puissance** permettant de passer des signaux de commande (basse puissance) sortant des organes de controle (microcontroleur par exemple) à des signaux de puissance.

Pont en H

Il existe une structure (souvent intégrée dans ce qu'on appelle des drivers) permettant le pilotage d'un moteur dans les deux directions. Cette structure se nomme un **pont en H** et utilise 4 transistors.



Il faut alors piloter en même temps deux des quatre transistors pour laisser passer du courant dans une direction particulière dans le moteur :

- piloter T1 et T4 (figure a) permet de faire tourner le moteur dans un sens ;
- piloter T2 et T3 (figure b) permet de faire tourner le moteur dans l'autre sens.

L293D ou SN754410NE

Le composant **L293D** (ou son cousin **SN754410NE**) intègre deux structures de ce type pour pouvoir piloter 2 moteurs indépendamment dans deux directions (ou 4 moteurs dans une direction).

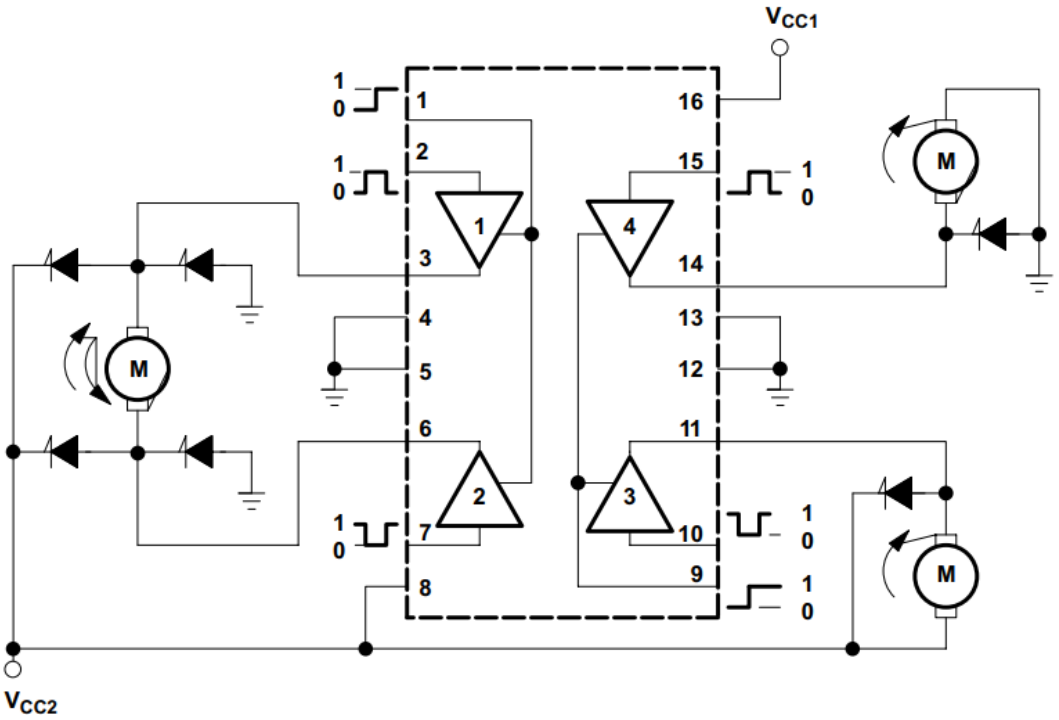


Schéma de cablage du L293D / Documentation Texas Instrument

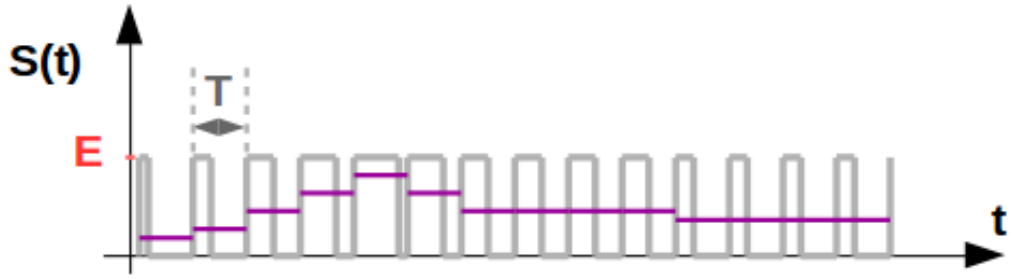
Les **entrées 2 et 7** du composant permettent de faire **tourner le moteur dans un sens ou dans l'autre**. L'**entrée 1** est un **interrupteur général** des modules 1 et 2.

Les signaux des entrées 1, 2 et 7 sont des signaux de faible puissance provenant de la carte Nucléo. Les sorties 3 et 6 sont des sorties de puissance, qui tirent leur énergie de l'alimentation V_{CC2}. Une alimentation de faible puissance (typiquement 5V) - qui doit être indépendante de la source de puissance - doit également être fournie au composant sur V_{CC1}.

Utilisation de signaux modulés en largeur d'impulsion

Les **moteurs à courant continu** se pilotent grâce à des **tensions continues**. Or dans les deux structures vues précédemment, la tension appliquée au moteur ne peut prendre que 3 valeurs : $-V_{CC}$, $0V$ ou V_{CC}, ne laissant alors que 3 réactions possibles du moteur : tourner à une vitesse fixe Ω (dépendant de la valeur de V_{CC}) dans une direction ou dans une autre, ou être à l'arrêt.

En utilisant le **principe de la modulation en largeur d'impulsions** (*PWM* en anglais) de signaux numériques "rapides", il est toutefois possible de contrôler la vitesse de rotation des moteurs à courant continu.



Signal PWM et sa valeur moyenne

En effet, les systèmes mécaniques, que sont les moteurs, ont des **constantes de temps de réaction assez élevées** (de l'ordre de la centaine de millisecondes pour les moteurs de ce robot). En leur appliquant un signal rectangulaire suffisamment rapide, le moteur n'aura pas le temps de réagir aux fluctuations rapides du signal. Seul le **signal moyen** sera alors traduit par le moteur en consigne (système passe-bas).

Travail à réaliser

➜ **M** Télécharger le projet de base et décompresser l'archive dans un répertoire de votre ordinateur.

https://lense.institutoptique.fr/ressources/mbed/hello-L476RG_pre.zip

➜ **M** Paramétrer les broches de commande du moteur de gauche en sortie modulée (MOT_L_1 et MOT_L_2).

➜ **M** Paramétrer la broche MOT_EN en sortie standard. Forcer la sortie à 1 pour activer les sorties de puissance sur les moteurs.

➜ **M** Ecrire une fonction qui permet de faire tourner le moteur gauche dans une direction à un rapport cyclique donné en argument (entre 0 et 100%).

➜ **M** Ecrire une fonction qui permet de stopper le moteur.

➜ **M** Ecrire une boucle infinie qui permette de tester vos différentes fonctions :

- lancer le moteur à 80% pendant 2 secondes
- stopper le moteur pendant 3 secondes
- lancer le moteur à 50% pendant 4 secondes
- stopper le moteur pendant 1 seconde

➜ **M** Connecter l'alimentation externe sur l'entrée d'alimentation du robot (connecteur J8 - voir partie **Alimentation électrique** de ce document).

➜ **M** Tester votre programme.

Attention ! Couper l'alimentation externe lorsque vous n'utilisez pas le robot (en particulier lorsque vous modifier et téléverser votre programme dans la carte).

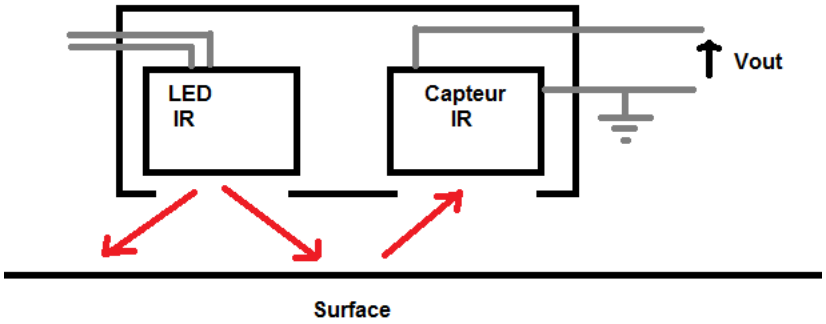
➜ **M** Faire le même travail pour le moteur droit.

➜ **M** Ajouter des fonctions permettant de faire avancer et reculer votre robot. Tester ces fonctions.

On pourra utiliser le signe de la vitesse pour déterminer le sens de rotation de la roue...

Etape 9 / Acquérir des données des capteurs de ligne

Les **capteurs de ligne** utilisés sur cette plateforme sont des **capteurs optiques**. Une **LED infrarouge** émet un signal vers le bas du robot qui est ensuite réfléchie plus ou moins fortement par le matériau présent au sol. Cette réflexion est alors captée par une **photodiode** ou un phototransistor.



Principe du capteur de ligne / gedgeyblog.wordpress.com

Un comparateur est ensuite ajouté en sortie pour transmettre uniquement un signal numérique : '0' si ligne détectée, '1' sinon (ou inversement selon le type de compareteur utilisé). Un potentiomètre de réglage du seuil de détection est présent sur les capteurs. Une LED visible est également présente permettant de visualiser l'état du capteur (cela permet notamment de régler le seuil plus facilement).

Travail à réaliser

➜ **M** A partir du programme de base, modifier le code principal pour paramétrer les broches des **capteurs de ligne** en entrée d'interruption (LINE_L, LINE_C et LINE_R).

➜ **M** Ajouter une fonction d'interruption, exécutée lors de l'activation d'un des capteurs de ligne, permettant de stocker l'état des 3 capteurs dans 3 variables indépendantes (booléennes par exemple) et d'afficher leur valeur sur la console.

➜ **M** Tester votre programme à l'aide d'une feuille blanche contenant une ligne noire de largeur 2 à 3 cm (ou d'un morceau de piste).

Etape 10 / Suivre une ligne

➜ M A partir des étapes précédentes (8 et 9), **écrire**, **tester** et **valider une application embarquée sur le robot** qui permette de répondre à ce cahier des charges : **suivre une ligne le plus longtemps possible** (sur piste).

Il vous est conseillé de modifier la fonction d'interruption réalisée dans l'étape 9 pour qu'elle modifie la vitesse des moteurs en fonction de l'endroit où se trouve le robot par rapport à la ligne. Vous pourrez vous aider d'une table de vérité de ce type.

LINE_L	LINE_C	LINE_L	Roue Gauche	Roue Droite
NON	NON	NON	0	0
NON	OUI	NON	70%	70%
...	...	...	...	...

Jettez un coup d'oeil au cahier des charges requis pour l'étape 11...

Faire valider votre application par un·e encadrant·e.

Séances 3 et 4 / Asservissement de position et évitement d'obstacle

Etape 11 / Suivre une ligne (validation)

Cahier des charges

Vous avez **60 min** pour mettre au point l'application embarquée permettant de répondre au cahier des charges suivant.

- Le bouton bleu (PC_13) devra permettre l'**arrêt d'urgence** de votre robot
- Le bouton SW1 (PA_11) devra permettre le **démarrage du robot**. Le robot sera posé au début de la piste à l'arrêt et sera mis en mouvement après appui sur ce bouton.
- La prise en compte des capteurs de ligne devra se faire sur interruption dans votre programme.
- Le robot devra être autonome dans son pilotage.

Condition de validation

Puis vous aurez un créneau de 10 min sur l'une des pistes pour valider votre application.

Niveau 1 Votre robot fait un tour de piste sans sortir de sa ligne

Niveau 2 Votre robot fait un tour de piste sans sortir de sa ligne en moins de 2 minutes

Niveau 3 Votre robot fait deux tours de piste sans sortir de sa ligne en moins de 3 minutes

Etape 12 / Acquérir les signaux des codeurs

➜ **M** Faire un programme qui copie la sortie du codeur de la roue gauche sur une sortie numérique visualisable à l'oscilloscope.

Prenez une sortie non utilisée par d'autres composants de la maquette ou une broche déjà utilisée par une LED (par exemple).

➜ **M** Visualiser ce signau lorsque vous faites tourner la roue gauche à la main.

➜ **Q** Que représentent-t-ils ? Comment peut-on s'en servir pour détecter la vitesse de rotation de la roue ? Sa position ?

➜ **Q** Peut-on connaître le sens de rotation simplement avec ce signal ?

➜ **M** Faire un programme qui détecte les fronts montants et descendants sur le codeur de la roue gauche et qui incrémente un compteur. Le programme devra se baser sur un système d'interruption. L'affichage des valeurs se fera sur la console série, depuis la fonction principale.

➜ **Q** Quel protocole de test proposeriez-vous pour valider votre programme ?

➜ **Q** Le résultat est-il différent selon que la roue tourne dans l'une ou l'autre des directions ?

➜ **M** Faire un programme qui affiche la valeur des compteurs des deux roues du robot. Ajouter les fonctions de commande de base du robot pour faire tourner les moteurs dans un sens ou dans un autre à une vitesse donnée.

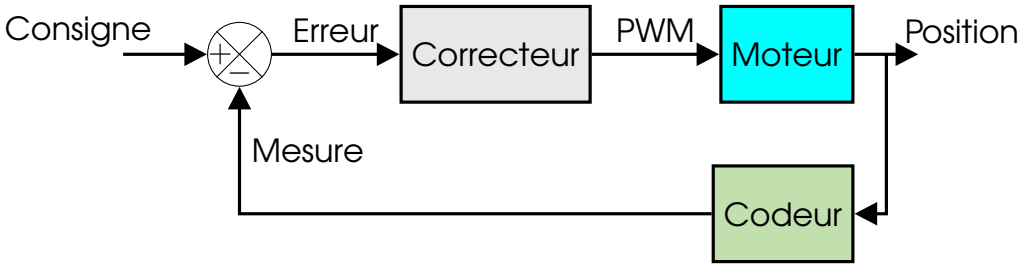
➜ **M** Tester votre programme en faisant tourner les moteurs (dans le vide) pendant 1 à 10 secondes et en relevant la valeur des deux compteurs de position.

➜ **Q** Que pouvez-vous conclure de ce test ?

Faire valider votre application par un·e encadrant·e.

Etape 13 / Asservir la position

Le principe de l'**asservissement de position d'un moteur à courant continu** est fourni par le schéma suivant :



*Le correcteur n'est pas indispensable, mais permet de meilleures performances de l'asservissement. Un **étage de puissance** est nécessaire entre la sortie PWM et le moteur.*

Le **codeur incrémental**, relié à une entrée numérique du microcontroleur, fournit un front montant (signal logique) à chaque changement de position de la roue (selon 20 encoches par tour sur ce modèle) qui va incrémenter la valeur d'un compteur.

Ce compteur indique le nombre d'encoches qui se sont écoulées depuis l'origine (donc le nombre de portion de tours que la roue à effectuer)

Ce compteur peut-être ensuite comparé à une consigne de position et produit un **signal d'erreur** : écart entre consigne et mesure.

Ce signal d'erreur va permettre ensuite de générer la **consigne de tension** à appliquer au moteur (ici par l'intermédiaire d'un signal PWM) afin qu'il atteigne sa position.

Travail à réaliser

On propose le code...

➜ **M** Remplacer le contenu du fichier *main.cpp* de l'exemple de base par le code contenu dans le fichier précédent.

➜ **M** Tester le code sur la plateforme robot.

➜ **Q** Décrire le fonctionnement de ce script sous forme d'un chronogramme, à partir du moment où une nouvelle consigne de position est appliquée.

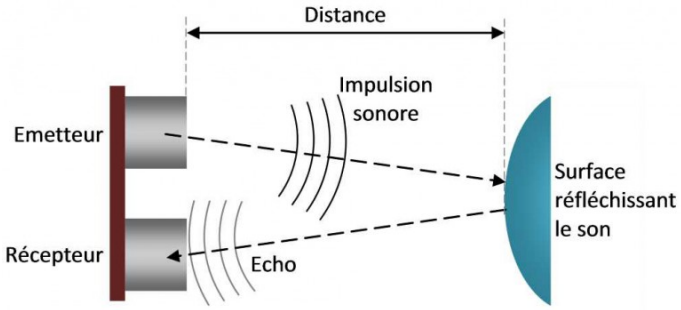
➜ **Q** Proposer une méthode de calibration (validation) de ce procédé de positionnement des roues.

➜ **M** Mettre en œuvre ce protocole.

Faire valider votre protocole de test ainsi que l'analyse de vos résultats par un·e encadrant·e.

Etape 14 / Acquérir les signaux du capteur ultra-son

La maquette est équipée d'un capteur de distance à ultrasons. Le principe est décrit dans le schéma suivant :



Principe du capteur de distance à ultrasons /
https ://arduino.blaisepascal.fr/capteur-de-distance-a-ultrasons/

Un **émetteur** génère une **impulsion sonore brève** (à une fréquence non audible). Cette onde est **réfléchie** par la surface à détecter. L'onde retour, appelée *écho* est reçue par un **récepteur**. On mesure alors le **temps** entre le moment où l'onde est émise et le moment où l'écho revient - souvent nommé temps de vol. Ces ondes se propageant à la vitesse du son (environ 340m/s), il est alors possible de revenir sur la distance parcourue par cette onde.

Travail à réaliser

On se propose de tester le code *dist_us.cpp*. Ce fichier est disponible sur le site du LEnsE dans la rubrique *Année / Première Année / Interfaçage Numérique S6 / Bloc 1 Systèmes embarqués / Robotique / Exemples MBED pour STM32*.

Attention : le robot doit **être alimenté par un générateur extérieur** pour que le capteur à ultrasons soit opérationnel.

➜ **M** Tester ce code et afficher sur la console Série la durée obtenue en plaçant un objet "plat" devant le robot. Recommencer l'expérience en faisant varier la distance entre l'objet et le robot.

➜ **M** Visualiser les signaux des broches D4 et D3 à l'aide de l'oscilloscope.

➜ **Q** Quelle est la limite de détection de ce capteur ?

Etape 15 / Contourner un obstacle

Etape 16 / Suivre une ligne et éviter des obstacles

INTERFAÇAGE NUMÉRIQUE

Travaux Pratiques

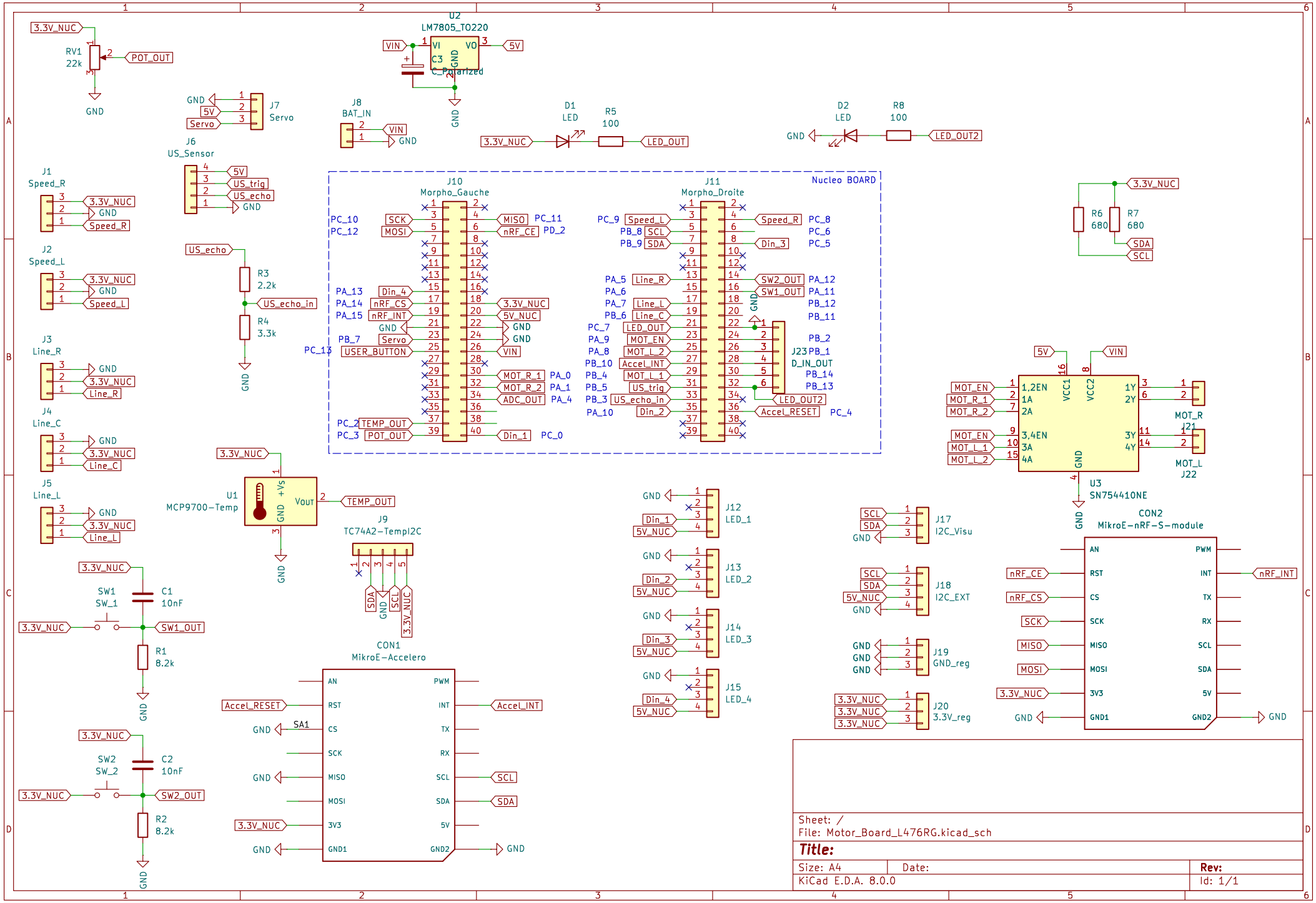
Semestre 6

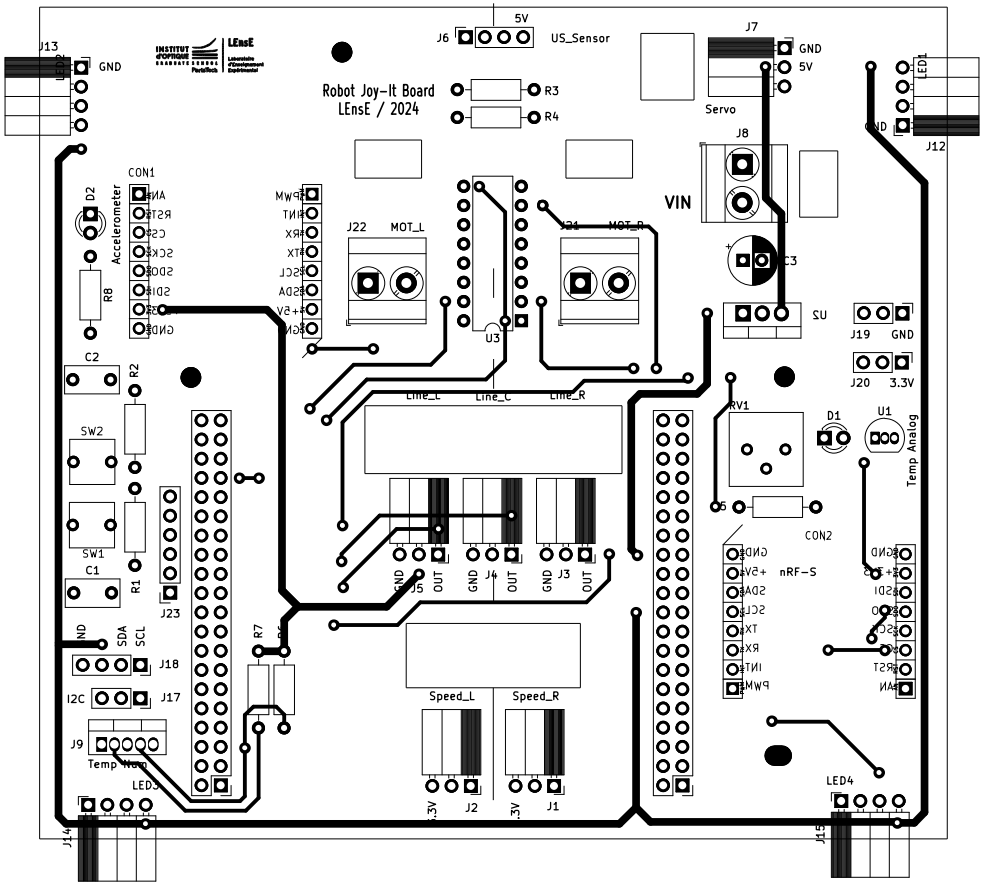
Ressources

Bloc Robot

Liste des ressources

- Schéma de la carte du robot Joy-It Car
- PCB de la carte du robot Joy-It Car


3.3V_NUC
RV1
22k
POT_OUT
GND
U2
LM7805_TO220
VIN
VI
VO
5V
C3
C_Polarized
GND
J7
Servo
GND
5V
Servo
J8
BAT_IN
VIN
GND
D1
LED
R5
100
LED_OUT
D2
LED
R8
100
LED_OUT2
J6
US_Sensor
5V
US_trig
US_echo
GND
J1
Speed_R
3.3V_NUC
GND
Speed_R
J2
Speed_L
3.3V_NUC
GND
Speed_L
J3
Line_R
GND
3.3V_NUC
Line_R
J4
Line_C
GND
3.3V_NUC
Line_C
J5
Line_L
GND
3.3V_NUC
Line_L
US_echo
R3
2.2k
US_echo_in
R4
3.3k
GND
Nucleo BOARD
J10
Morpho_Gauche
PC_10
PC_12
SCK
MOSI
MISO
PC_11
nRF_CE
PD_2
PA_13
PA_14
PA_15
Din_4
nRF_CS
nRF_INT
3.3V_NUC
5V_NUC
GND
PB_7
Servo
USER_BUTTON
VIN
PC_13
MOT_R_1
MOT_R_2
ADC_OUT
PA_0
PA_1
PA_4
PC_2
PC_3
TEMP_OUT
POT_OUT
Din_1
PC_0
J11
Morpho_Droite
PC_9
Speed_L
PB_8
SCL
PB_9
SDA
Speed_R
Din_3
PC_8
PC_6
PC_5
PA_5
Line_R
PA_6
PA_7
Line_L
PB_6
Line_C
PC_7
LED_OUT
PA_9
MOT_EN
PA_8
MOT_L_2
PB_10
Accel_INT
PB_4
MOT_L_1
PB_5
US_trig
PB_3
US_echo_in
PA_10
Din_2
SW2_OUT
SW1_OUT
PA_12
PA_11
PB_12
PB_11
PB_2
PB_1
D_IN_OUT
PB_14
PB_13
LED_OUT2
Accel_RESET
PC_4
J23
R6
680
R7
680
SDA
SCL
U1
MCP9700-Temp
+Vs
VOUT
TEMP_OUT
J9
TC74A2-TempI2C
SDA
GND
SCL
3.3V_NUC
U3
SN754410NE
5V
VIN
MOT_EN
MOT_R_1
MOT_R_2
1,2EN
1A
2A
VCC1
VCC2
1Y
2Y
MOT_R
J21
3,4EN
3A
4A
3Y
4Y
MOT_L
J22
CON2
MikroE-nRF-S-module
AN
RST
CS
SCK
MISO
MOSI
3V3
GND1
PWM
INT
TX
RX
SCL
SDA
5V
GND2
nRF_INT
SW1
SW_1
C1
10nF
SW1_OUT
R1
8.2k
SW2
SW_2
C2
10nF
SW2_OUT
R2
8.2k
CON1
MikroE-Accelero
Accel_RESET
SA1
Accel_INT
J12
LED_1
J13
LED_2
J14
LED_3
J15
LED_4
J17
I2C_Visu
J18
I2C_EXT
J19
GND_reg
J20
3.3V_reg
Sheet: /
File: Motor_Board_L476RG.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. 8.0.0
Id: 1/1


Robot Joy-It Board
LEnsE / 2024
VIN
Servo
US_Sensor
Accelerometer
Temp Analog
Temp Num
nRF-S
Line_L
Line_C
Line_R
Speed_L
Speed_R
MOT_L
MOT_R
J13
J12
J8
J6
J7
J22
J21
U3
C3
R3
R4
R8
C2
R2
SW2
SW1
C1
R1
J23
J18
J17
J9
J14
J15
J5
J4
J3
J2
J1
J19
J20
RV1
D1
U1
D2
LED3
LED4
LED1
CON1
CON2
R7
GND
5V
3.3V
SDA
SCL
I2C