Institut d'Optique Graduate School
ProTIS

ProTIS - TP/Projet
Phase 1 - Prise en main technique

Objectifs

Cette première phase (sur 2 séances) est consacrée à la prise en main technique de la **programmation de systèmes embarqués** (ici des cartes **Nucléo** de *STMicroelectronics*, basées sur des microcontroleurs *STM32*) et la prise en main de l'interface de développement **Keil Studio Cloud** (et les bibliothèques MBED).

VOUS DEVREZ, EN BINÔME, TESTER ET DOCUMENTER LES EXEMPLES QUE NOUS VOUS PROPOSONS. VOUS DEVREZ RÉDIGER LES PROTOCOLES DE TEST PERMETTANT DE COMPRENDRE LE FONCTIONNEMENT DES CODES FOURNIS AINSI QU'UNE ANALYSE DES RÉSULTATS OBTENUS.

DANS UN SECOND TEMPS, VOUS DEVREZ ADAPTER CES EXEMPLES POUR METTRE EN OEUVRE LE **CAPTEUR ANALOGIQUE** ET L'**ACTIONNEUR** QUE VOUS AUREZ TIRÉS AU SORT LORS DE CETTE PHASE.

IL EST IMPORTANT DE COMMENTER VOS CODES ET D'ÉCRIRE DES EN-TÊTES QUI PERMETTENT D'IDENTIFIER RAPIDEMENT LE RÔLE DU PROGRAMME QUE LE FICHIER CONTIENT.

Ressources

Pour cette prise en main technique, vous aurez accès à une **carte de prototypage**, nommée **TUNIS**, incluant une carte *Nucléo L476RG*. Une documentation est fournie en annexe de ce document.

Ce **sujet** est disponible au format électronique sur le site du LEnsE - https ://lense.institutoptique.fr/ dans la rubrique Année / Deuxième Année / ProTIS / Phase 1 - Prise en main technique.

Des **bibliothèques de fonctions** pour certains composants sont disponibles à l'adresse suivante :

https://github.com/IOGS-LEnsE-ressources/mbed6-libraries / Sections *progs* et *platforms*

*dépôt GitHub - bibliothèques validées sous **MBED-OS 6.17***

Des **tutoriels** pour la prise en main de KEIL UVISION et de MBED-OS (proposés au semestre 6 classique de la formation d'ingénieur) sont disponibles au format électronique sur le site du LEnsE - https ://lense.institutoptiqu dans la rubrique Année / Première Année / Interfaçage Numérique / Bloc 1 - Systèmes embarqués / Intro MBED et STM32.

Avant de pouvoir programmer le microcontroleur présent sur les cartes Nucléo que nous allons utiliser, il faut au préalable **avoir un projet** pour *Keil uVision*.

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

A - Systèmes embarqués et interruptions

Afin de vous familiariser avec la démarche qualité mise en place dans ce module et de vous remettre à niveau sur la programmation embarquée, nous vous proposons d'étudier les codes fournis dans les fichiers PROTIS_TEMPLATE_NUCLEO_INTERRUPT.CPP et PROTIS_TEMPLATE_NUCLEO_TICKER.CPP, à tester sur la carte TUNIS.

Interruptions externes

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

➜ **M** Récupérer le fichier PROTIS_TEMPLATE_NUCLEO_INTERRUPT.CPP sur le site du LEnsE (ProTIS / Phase 1 - Prise en main technique).

Compétences visées

— Tester une application sur une carte à microcontrôleur (ici Nucleo / MBED)
— Afficher une chaine de caractère sur une console (TeraTerm)
— Mesurer le temps d'exécution d'une instruction à l'aide d'un oscilloscope
— Utiliser des interruptions
— Documenter son code

Travail à réaliser

— Préciser quels sont les éléments mis en oeuvre dans ce programme.
— Visualiser les signaux S1 et S2.
— Décrire le fonctionnement de ce code sous forme d'un organigramme ou/et d'un chronogramme.
— Documenter le code fourni.
— Proposer alors un protocole pour la mesure d'un temps d'exécution et/ou de répétition d'une action (ensemble d'instructions).

Interruptions internes

Le second code à tester et documenter est PROTIS_TEMPLATE_NUCLEO_TICKER.CPP.

Ce code permet de convertir à intervalle régulier un signal analogique sur l'entrée A0 et d'afficher la valeur convertie sur une console série (TeraTerm par exemple).

ATTENTION : les entrées des cartes Nucléo n'acceptent pas de tension supérieure à 3.3V et inférieure à 0V !

Compétences visées

— Faire une conversion analogique-numérique à intervalle constant (i.e. à fréquence d'échantillonnage fixe).
— Mesurer les temps d'exécution de chacune des instructions mises en jeu, à l'aide d'un oscilloscope.
— Mesurer la période d'échantillonnage, à l'aide d'un oscilloscope.

Travail à réaliser

— Décrire le fonctionnement de ce code sous forme d'un organigramme ou/et d'un chronogramme.
— Documenter le code fourni.
— Afficher la tension en V mesurée par la carte sur la console série (au lieu de la valeur brute).
— Décrire la méthode de mesure de la fréquence d'échantillonnage.
— Déterminer la fréquence d'échantillonnage maximale que l'on peut obtenir.
— Quelle est la durée de l'affichage ? Quel est l'impact de cet affichage sur la fréquence maximale ?

B - Traitement du signal

B1 - Signaux "classiques"

A partir du code précédent, réaliser et tester un dispositif embarqué permettant de calculer la moyenne glissante de 10 échantillons d'un signal analogique et de restituer son résultat de manière analogique.

Compétences visées

— Mettre en oeuvre une conversion numérique-analogique à intervalle constant (i.e. à fréquence d'échantillonnage fixe).
— Réaliser un traitement numérique simple sur un signal électronique.
— Mettre en oeuvre des méthodes de caractérisation classique de système (réponse en fréquence, réponse indicielle, réponse impulsionnelle).

Travail à réaliser

— Décrire le fonctionnement de ce code sous forme d'un organigramme ou/et d'un chronogramme.
— Documenter le code fourni.
— Rédiger le protocole de test permettant de valider le fonctionnement de votre système (réponse en fréquence, réponse indicielle et réponse impulsionnelle).
— Rédiger une analyse succincte des résultats obtenus.

B2 - Capteur analogique

On souhaite à présent faire l'acquisition à intervalle régulier (en moyennant sur N points) de la grandeur physique mesurée par le capteur analogique tiré au hasard.

Compétences visées

— Mettre en oeuvre un mesure analogique d'une grandeur physique.
— Réaliser un traitement numérique simple sur un signal électronique.

Travail à réaliser

— Rédiger le protocole de test permettant de valider le fonctionnement de votre système.
— Rédiger une analyse succincte des résultats obtenus.

C - Pilotage d'un composant de puissance

En fonction de l'élément de puissance à commander, choisissez la section C1 ou C2.

C1 - Pilotage d'un moteur à courant continu

On souhaite à présent réaliser un dispositif embarqué permettant de **piloter un moteur à courant continu en vitesse**, à l'aide d'un pont en H (présent sur la carte TUNIS), dans les deux directions. La vitesse sera imposée par une tension analogique (entrée E1 par exemple). Le sens sera choisi en appuyant sur le bouton-poussoir de la carte Nucleo.

Compétences visées

— Piloter la vitesse d'un moteur à courant continu de manière numérique à l'aide de signaux modulés en largeur d'impulsion.
— Utiliser un étage de puissance et en comprendre le fonctionnement et l'intérêt.

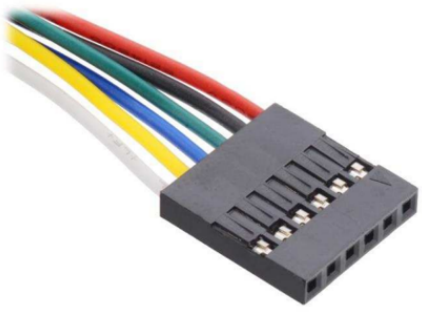
Travail à réaliser

— Câbler le moteur à courant continu.
— Réaliser le programme de pilotage.
— Tester le système et rédiger un rapport de test du fonctionnement.
— Afficher et commenter les courbes d'évolution temporelle des signaux A et B de l'encodeur de position (ces signaux pourront être recopiés sur les sorties BNC - S1 et S2 de la carte TUNIS).
— Réaliser le schéma fonctionnel de ce dispositif.

Matériel à disposition

Des moteurs à courant continu de type `POLOLU 4843`, dont la documentation est fournie à l'adresse suivante : https ://www.pololu.com/file/0J1829/pololu-25d-metal-gearmotors.pdf, sont disponibles.
On donne le schéma de cablage du connecteur :

Lead Color	Function
Red	Motor power
Black	Motor power
Green	Encoder ground
Blue	Encoder Vcc (3.5 V to 20 V)
Yellow	Encoder A output
White	Encoder B output



La tension d'alimentation maximale est de 12V.
Ils sont équipés d'un encodeur dont les caractéristiques sont données ci-dessous :
— Nombre d'encoches : 48
— Rapport de réduction : 20.4 / 1
— Tension d'alimentation maximale de l'encodeur : 5V

C2 - Pilotage d'un bandeau de LEDs RGB

On souhaite à présent réaliser un dispositif embarqué permettant de **piloter la couleur d'un bandeau de LED RGB**, à l'aide de transistors de type IRL540. L'intensité lumineuse de chacune des couleurs du bandeau sera imposée par une tension analogique (entrées E1 et E2 par exemple).

Compétences visées

— Piloter les 3 couleurs de manière indépendante à l'aide de signaux modulés en largeur d'impulsion.
— Utiliser un étage de puissance et en comprendre le fonctionnement et l'intérêt.

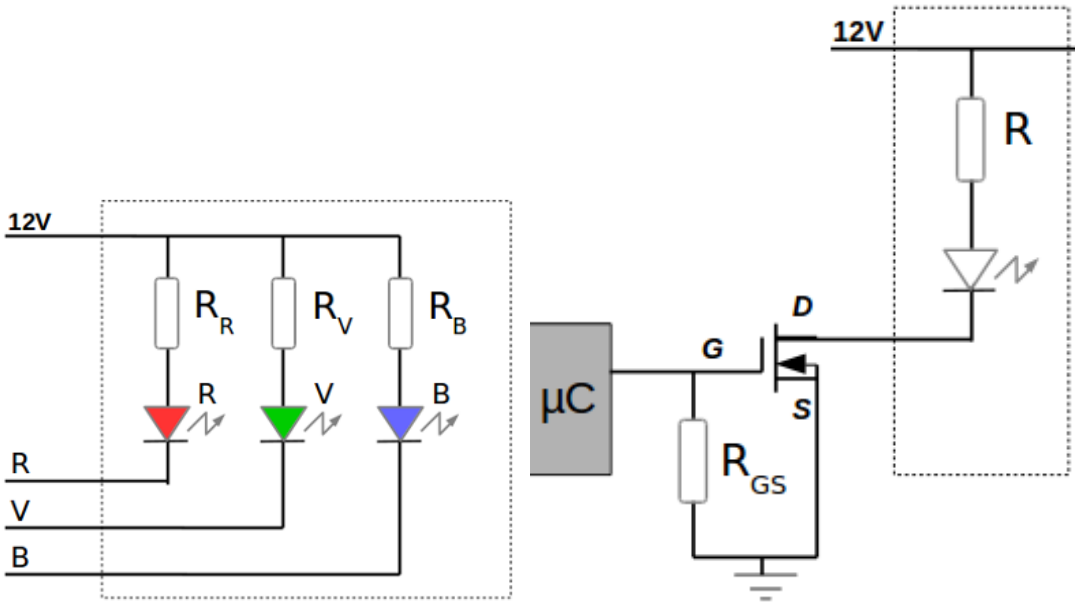
Travail à réaliser

— Câbler le bandeau à l'aide d'un étage de puissance.
— Réaliser le programme de pilotage.
— Tester le système et rédiger un rapport de test du fonctionnement.
— Réaliser le schéma fonctionnel de ce dispositif.

Matériel à disposition

Des bandeaux de LEDs RGB dont les trois couleurs sont pilotables indépendamment. Ces bandeaux doivent être alimentés sous une tension de **12V** et consomment, à pleine puissance, 7W/m (pour des versions trichromes – il existe aussi des bandeaux monochromes).

Elles sont conçues selon le schéma suivant. Les résistances de limitation du courant sont déjà intégrées au bandeau.



Le pilotage d'un tel bandeau nécessite d'adapter le niveau de tension mais également d'amplifier le courant entre la carte Nucléo (3.3V) et le bandeau. On utilisera des transistors de type MOSFET - IRL540.

Ce type de transistor (appelé MOSFET) est commandé en tension. Lorsqu'une tension est appliquée entre les broches G (grille) et S (source), il y a passage de courant possible entre les broches D (Drain) et S (Source). Lorsque cette tension est nulle, il n'y a pas de courant qui passe entre ces broches. On parle alors de commutation du transistor qui se comporte alors comme un interrupteur commandé en tension.

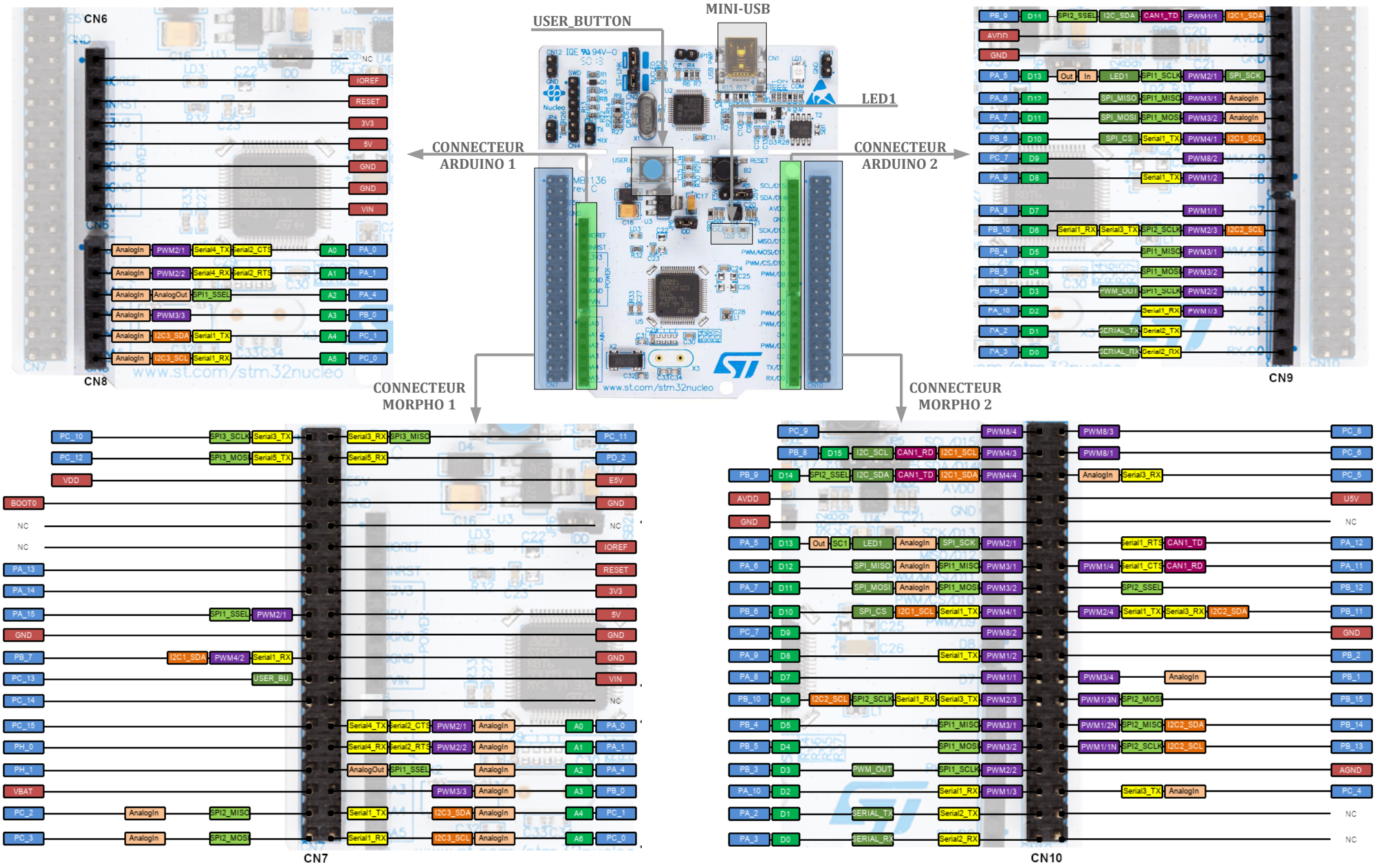
D - Premier système

A partir du capteur analogique, que vous avez tiré au hasard, vous devez **piloter le moteur à courant continu** précédent en fonction de l'une des **grandeurs physiques résultant de ce capteur** (position angulaire, température, distance...).

Travail à réaliser

- Décrire le fonctionnement de votre système par l'intermédiaire :
  - d'une description fonctionnelle
  - d'un cas d'utilisation
- Documenter le code fourni.
- Rédiger le protocole de test permettant de valider le fonctionnement de votre système.
- Rédiger une analyse succincte des résultats obtenus.

Carte Nucléo-64 / STM32L476 / Broches d'entrées-sorties

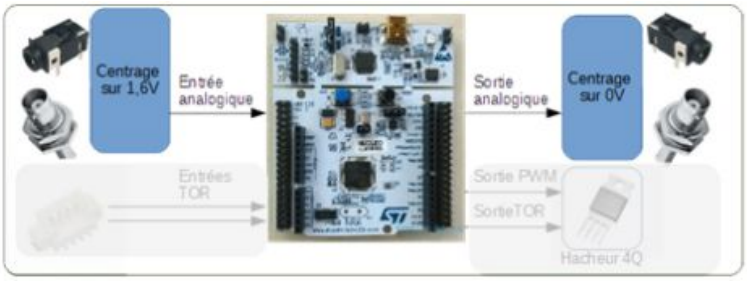


Carte d'extension TUNIS

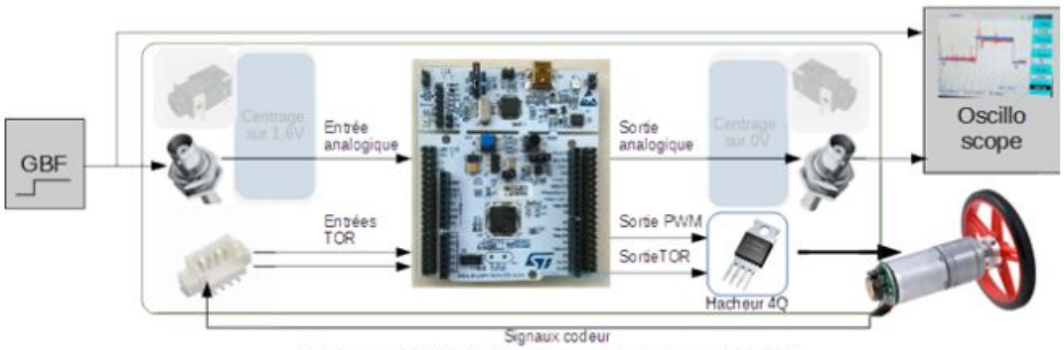
Traitement du signal / asservissement de position

Cette **maquette d'interface** avec une carte Nucleo de type L476RG facilite la mise en œuvre de deux types d'applications embarquées :

- du traitement du signal (filtrage numérique par exemple) ;
- de l'asservissement de vitesse ou de position d'un moteur à courant continu.



Carte en configuration traitement du signal



Carte en configuration asservissement de position

Entrées / Sorties - Interfaçage avec Nucléo

Le tableau suivant référence les différentes entrées-sorties de la maquette d'interface et les broches associées de la carte Nucléo.

Fonction	Repère	Direction	Nucléo
Entrée analogique 1	E1 / EA0	Entrée	PA_0 / A0
Entrée analogique 2	E2 / EA1	Entrée	PA_1 / A1
Sortie analogique 1	S1 / DAC1	Sortie	PA_4 / A2
Sortie analogique 2	S2 / DAC2	Sortie	PA_5 / D13*
PWM1	PWM1	Sortie	PB_6 / D10
PWM2	PWM2	Sortie	PC_7 / D9
Erreur moteur	EF	Entrée	PA_7 / D11
Codeur voie A	SA	Entrée	PB_8 / D15
Codeur voie B	SB	Entrée	PB_9 / D14

** **ATTENTION** : lorsque la sortie analogique S2 est utilisée, vous ne pouvez plus accéder à LED1 sur la carte Nucléo (qui est également reliée sur D13 ou PA_5)*

Toutes les autres broches de la carte Nucléo sont utilisables dans votre application et peuvent être connectées directement sur les connecteurs Arduino ou Morpho.

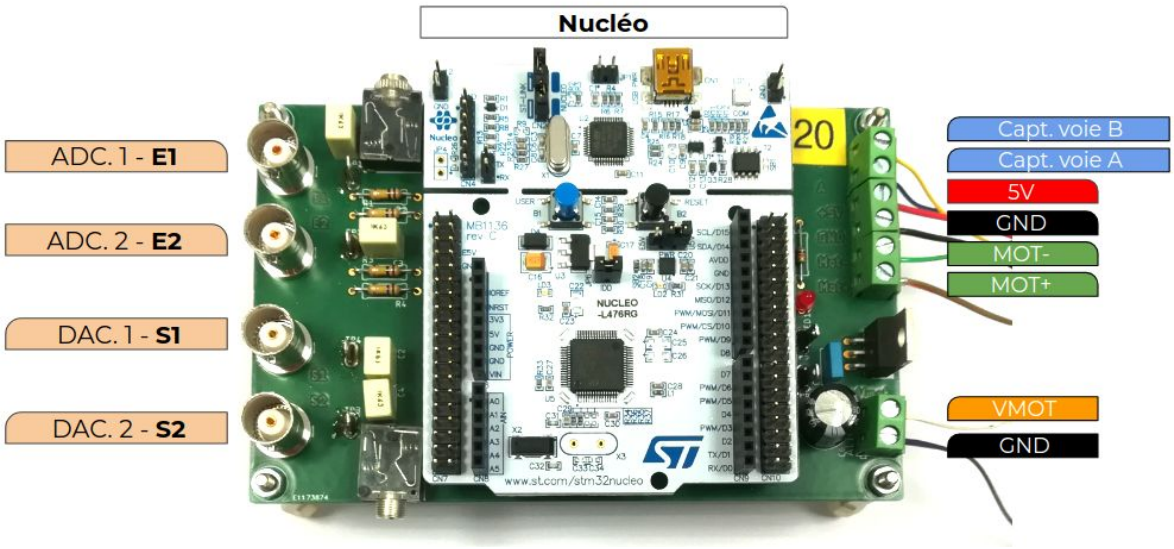
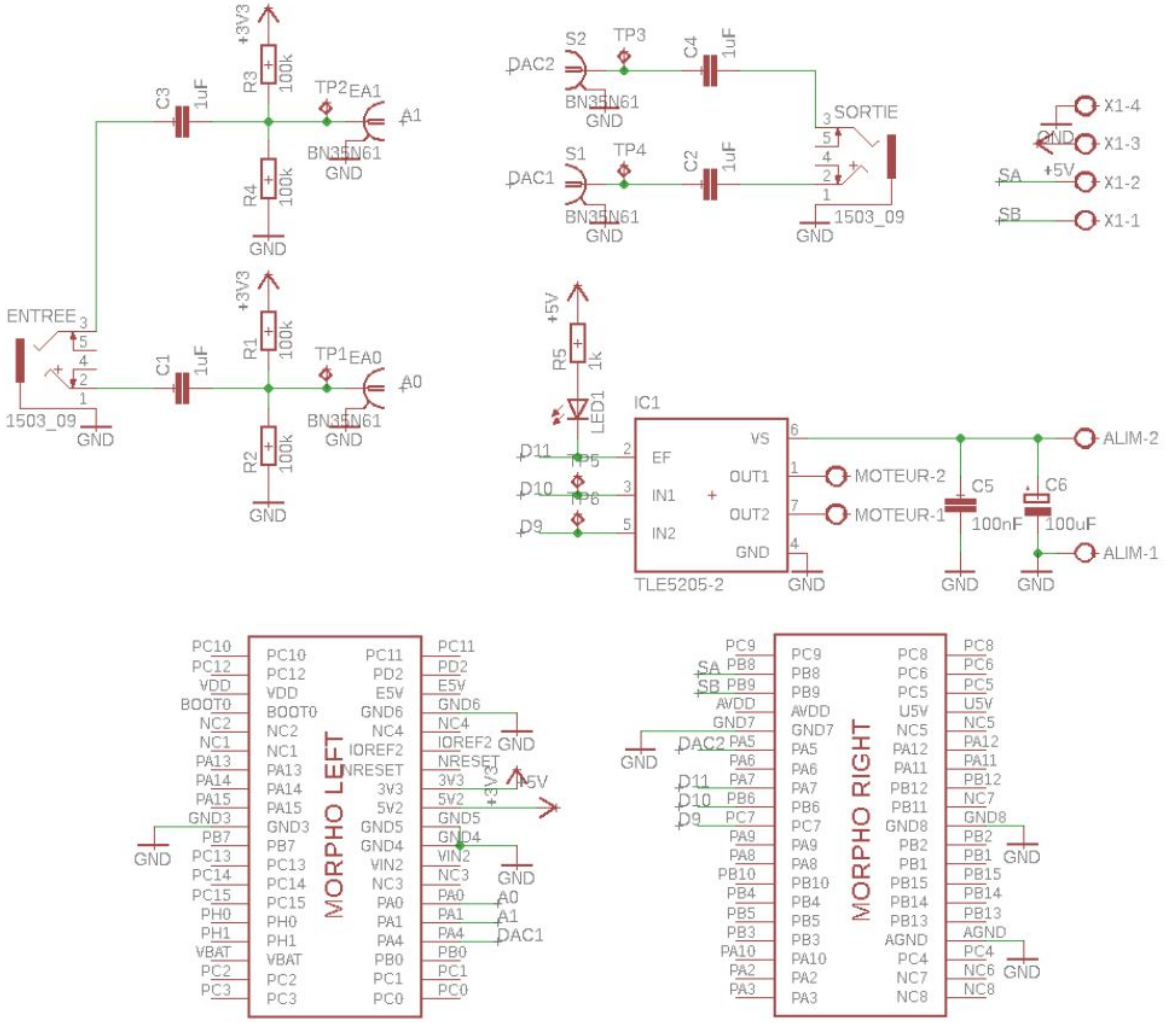



Schéma de la carte d’extension