



# INTERFAÇAGE NUMÉRIQUE

Travaux Pratiques

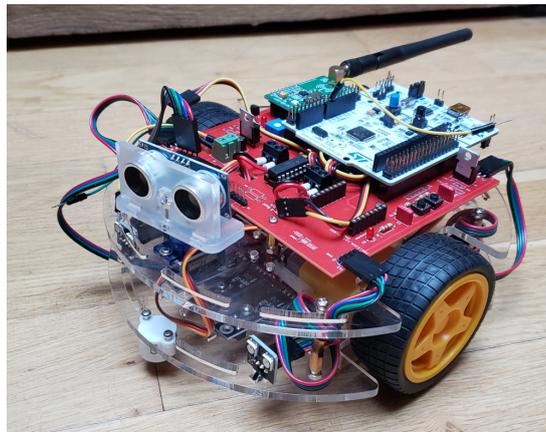
Semestre 6

---

## Robotique et systèmes embarqués

---

4 séances





---

## Robotique et systèmes embarqués

---

À l'issue des séances de TP concernant le **bloc de robotique** (4 séances), les étudiant-es seront capables de ??? .

*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 Robot.*

---

Pour cela, ils-elles seront capables de :  
— ???

---

## Objectifs du mini-projet

L'objectif principal de ce mini-projet est de **développer le code embarqué d'une plateforme robotique** lui permettant :

- soit de se déplacer de manière autonome le long d'une ligne sans percuter d'obstacle.
- soit d'être piloter à distance par une télécommande et d'afficher des informations provenant de capteurs intégrés à la plateforme.

## Ressources

Matériel : robot Joy-It car, télécommande, logiciel Arduino, carte Nucléo L476RG, capteurs, cartes de communication nRF24L01

*Un descriptif des ressources de la plateforme robotique est disponible dans la suite de ce document.*

Tutoriaux Arduino

Codes d'exemple : Communication entre deux nRF24, affichage sur écran LCD...

## Déroulement du bloc

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

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

## Déplacements élémentaires

Les deux premières séances sont communes sur la démarche de conception d'une application embarquée.

## Séance 1 - Prise en main de la maquette et premiers programmes Arduino

**Etape 1 - 30 min** Piloter des sorties numériques - LED

**Etape 2 - 30 min** Acquérir des données numériques - Bouton-poussoirs (interruptions)

**Etape 3 - 60 min** Acquérir des données analogiques - Potentiomètres

**Etape 4 - 30 min** Utiliser des sorties modulées en largeur d'impulsion (PWM) - LEDs

**Etape 5 - 90 min** Piloter les deux moteurs du robot

## Séance 2 - Capteurs et bibliothèques

**Etape 6 - 60 min** Piloter les phares du robot à l'aide de la bibliothèque WS2812 (NeoPixel)

**Etape 7 - 30 min** Acquérir des données du capteur de température analogique

**Etape 8 - 60 min** Acquérir des données des capteurs de ligne

**Etape 9 - 90 min** Acquérir des données de l'accéléromètre (I2C)

## Pilotage de haut niveau

Les deux séances suivantes seront consacrées au pilotage du robot en fonction du cahier des charges choisi (suivi d'une ligne ou pilotage par une télécommande).

### Robot autonome / Suivi Ligne

*Réalisable par un seul binôme*

**Etape A11 - 120 min** Définir et tester une première structure de code permettant de piloter les deux moteurs du robot en fonction de la détection des lignes

**Etape A12 - 90 min** Acquérir les signaux du capteur ultrason

**Etape A13 - 90 min** Piloter le servomoteur associé au capteur ultrason

**Etape A14 - 180 min** Améliorer le programme de contrôle du robot

### Robot télécommandé

*Réalisable par association de deux binômes, un binôme sur la télécommande (t) et un autre sur le robot (r)*

**Etape T11t - 60 min** Prendre en main la maquette de la télécommande

**Etape T12t - 60 min** Acquérir les données provenant du joystick

**Etape T11r - 120 min** Définir et tester une première structure de code permettant de piloter les deux moteurs du robot en fonction d'un ordre de direction

**Etape T13 - 120 min** Utiliser la bibliothèque nRF24 pour mettre en place un échange de données entre la télécommande et le robot

**Etape T14t - 60 min** Transmettre les données de direction depuis la télécommande vers le robot

**Etape T14r - 60 min** Traduire les données de direction en ordre pour les moteurs du robot

**Etape T15 - 180 min** Améliorer le programme de contrôle du robot (utilisation de l'écran LCD de la télécommande pour afficher la direction, les données de température, de l'accéléromètre...)

---

## Liste des autres ressources

- **Tutoriaux Arduino :**
  - Projet Arduino et structure d'un code embarqué
  - Entrées/Sorties Numériques
  - Entrées Analogiques
  - Sorties PWM
  - Utilisation d'une bibliothèque
  - Liaison série
  - Liaison SPI / I2C
  - Communication avec nRF24L01
- Pilotage d'un composant de puissance
- Mise en place d'un protocole d'échange de données

## Séance 2 / Capteurs et bibliothèques

### Etape 9 - Acquérir des données de l'accéléromètre (I2C)

*Temps conseillé : 90 min*

Le composant que nous allons étudier est un **accéléromètre et magnétomètre** intégrés sur une même puce de silicium. Sa référence est **FXOS8700CQ**. Ce composant est intégré au module *MikroE DOF6 - IMU Click*.

#### Protocole I2C

DESCRIPTION PROTOCOLE et CONNECTIQUES !

ATTENTION ! Les broches utilisées sur la carte Nucléo pour l'I2C ne sont pas celles par défaut. Il est indispensable de préciser les broches SDA et SCL à l'aide des méthodes suivantes :

```
1 Wire.setSDA( PB9 );  
2 Wire.setSCL( PB8 );
```

→ M Ouvrir le code *09\_accelero.ino* fourni. Compiler ce code et téléverser ce code dans la carte Nucléo.

Ce code contient les fonctions *test\_FXOS()* et *read\_i2c\_buffer()*, ainsi que des définitions des registres internes du composant.

→ M

#### Configuration

#### Récupération des données

*These registers contain the X-axis, Y-axis, and Z-axis 14-bit left-justified sample data expressed as 2's complement numbers. [NXP Doc p.52 of 113]*

#### Traceur Série

```
1 Serial.print(valeur1);  
2 Serial.print(",");  
3 Serial.print(valeur2);  
4 Serial.print(",");  
5 Serial.print(valeur3);  
6 Serial.print(",");  
7 Serial.print(valeur4);  
8 Serial.println();
```

---

## Pilotage / Robot télécommandé

---

### Etape A13 - Piloter le servomoteur associé au capteur ultrason

*Temps conseillé : 90 min*

Utilisation de la sortie modulée PB7

```
1 LL_GPIO_SetAFPin_0_7(GPIOB, GPIO_PIN_7, GPIO_AF1_TIM2);
```

## Pilotage / Robot télécommandé

### Etape T13 - Communication avec des modules nRF24L01

Installation de la bibliothèque **RF24** sur *Arduino*.

#### Protocole SPI

Changer les broches par défaut pour la liaison SPI

Ces lignes doivent être exécutées **AVANT** de démarrer logiquement le module nRF24 (avant la ligne `radio.begin()` ).

```
1 SPI.setMISO(SPI_MISO);  
2 SPI.setMOSI(SPI_MOSI);  
3 SPI.setSCLK(SPI_SCK);
```

Côté robot :

```
1 #define NRF_CE PD2  
2 #define NRF_CSN PA14  
3 #define NRF_INT PA15  
4 #define SPI_SCK PC10  
5 #define SPI_MISO PC11  
6 #define SPI_MOSI PC12
```

Côté télécommande :

```
1 #define NRF_CE PA15  
2 #define NRF_CSN PB7  
3 #define NRF_INT PA14  
4 #define SPI_SCK PA5  
5 #define SPI_MISO PA6  
6 #define SPI_MOSI PA7
```



## Robot Joy-It Car / Présentation du matériel

---

La tension maximale admissible par les moteurs est de 7 V !

---

## INTERFAÇAGE NUMÉRIQUE

### Travaux Pratiques

Semestre 6

---

## Ressources

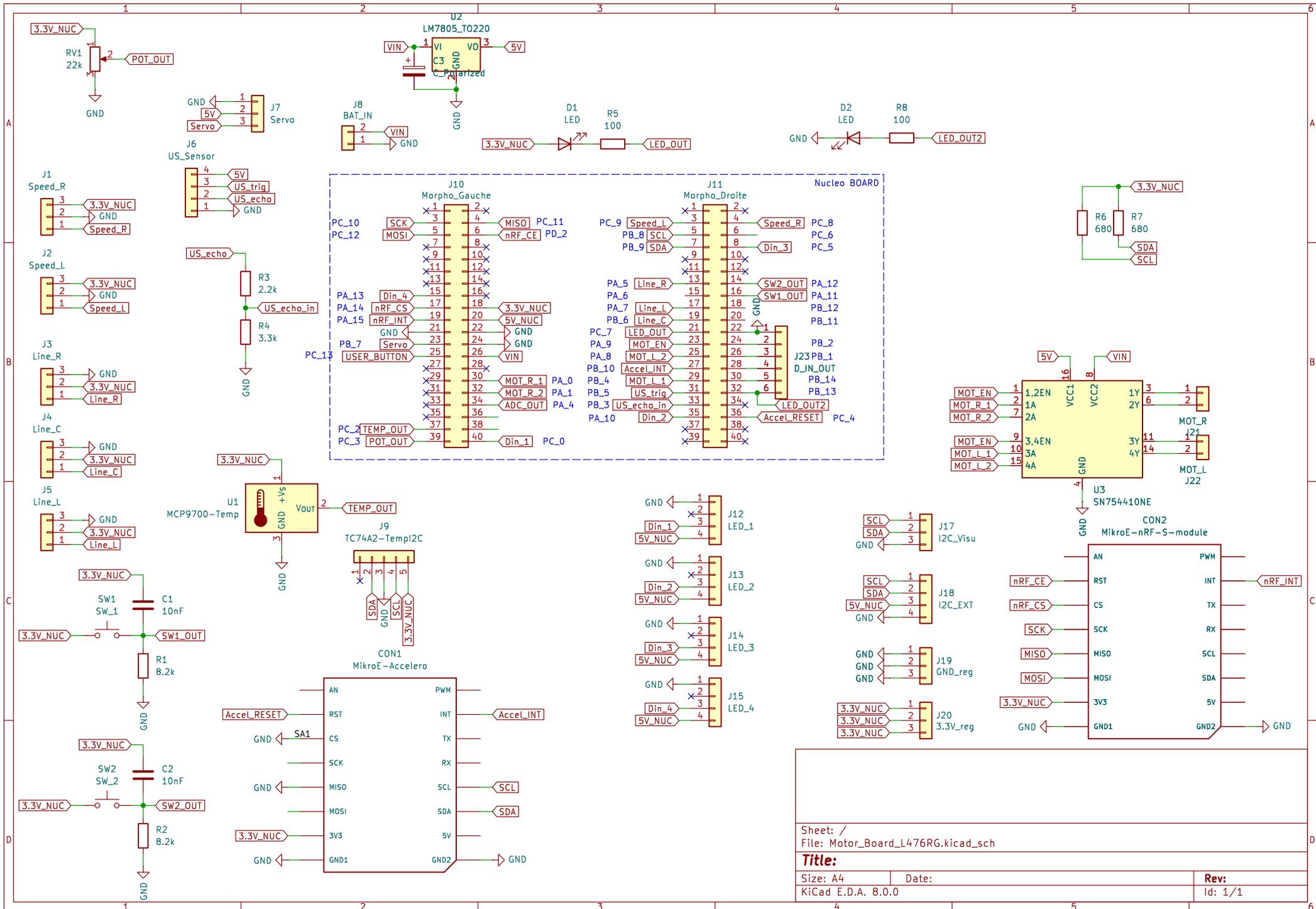
---

Bloc Robot

### Liste des ressources

- [Schéma de la carte du robot Joy-It Car](#)
- [PCB de la carte du robot Joy-It Car](#)





Sheet: /  
 File: Motor\_Board\_L476RG.kicad\_sch  
**Title:**  
 Size: A4 Date:  
 KiCad E.D.A. 8.0.0 Rev:  
 Id: 1/1

