

# Cours Systèmes embarqués et robotique

## Donnée des miniprojets

Le but du miniprojet est de partir sur la base des éléments que vous avez vu lors des TPs 1-5 pour créer plusieurs tâches plus complexes à résoudre par le robot e-puck2.

### Consigne

Vous êtes libres de déterminer vous-même les tâches que le robot doit effectuer, et donc la forme de la démonstration de votre programme. Les contraintes sont les suivantes :

- 1) Le projet doit être fait sur la base de la librairie e-puck2\_main-processor vue lors des TPs 4-5.
- 2) Vous devez obligatoirement utiliser les éléments suivants du robot e-puck2 dans votre projet :
  - a. **Les deux moteurs pas-à-pas.** Par exemple Régulation PID, odométrie précise, forme géométrique, etc.
  - b. **Un des capteurs de distance** (Capteurs de proximités infrarouges ou capteur Time-of-Flight). Par exemple détection d'obstacle petite ou grande distance.
  - c. **Un capteur parmi ceux que vous avez investigué durant les TPs 3-5, donc un capteur parmi : la caméra, les micros, l'IMU.** Par exemple avec la caméra : détection d'objet, suivi de lignes. Par exemple avec les micros : détection de sons sur la base de l'amplitude et/ou fréquence. Par exemple avec l'accéléromètre, détection d'un plan incliné, d'un choc. Par exemple avec le gyroscope, détection du mouvement d'un plan incliné.
- 3) Chaque capteur/actuateur doit être utilisé/géré dans une Thread, à l'image de leur utilisation dans les TPs 4-5. La création de threads doit suivre les besoins, mais toujours de manière à respecter les taux de rafraîchissements des capteurs et en faisant un usage intelligent des outils disponibles dans ChibiOS (messagebus, sémaphores, etc.).
- 4) Le code doit être rendu sous la forme d'une librairie (avec divers .c/.h) qui s'intègre avec la librairie e-puck2\_main-processor que vous avez utilisé lors des Travaux pratiques.

**Voici donc un exemple de projet :** Le robot e-puck2 reproduit les mouvements d'une balle sur un plan incliné en utilisant l'accéléromètre comme détecteur de la direction du vecteur gravité. Le robot "rebondit" lorsqu'il détecte un obstacle avec ses capteurs de proximité infrarouges.

## Forme du rendu

Le rendu du miniprojet sera composé de:

- 1) Une présentation (avec slides de préférence, pas obligatoire) de **3 minutes**
- 2) Une démonstration live (pas de vidéos) du programme final, d'une durée de **2 minutes**.
- 3) Un rapport de 3-4 pages (longueur indicative) qui donne un aperçu de la méthode de travail, des analyses, la conception du logiciel et des résultats obtenus. Une section doit donner un retour sur votre méthode de travail de groupe.
- 4) Le code structuré et commenté.
- 5) Une discussion sur le projet lors du rendu/présentation.

**+ demo + présentation = durée totale de passage ~25 minutes**

## Critères d'évaluation

Les critères d'évaluations seront :

- Clarté et propreté du code source (respect de conventions de base du C, des commentaires utiles, des valeurs définies, des fonctions et variables avec des noms clairs, une bonne utilisation du RTOS, etc.)
- Clarté et propreté du rapport (Numéro de section, légendes aux figures, axes aux graphiques, citations et références) **Un point enlevé sur la note finale du projet (note entière) pour des erreurs de ce type!**
- Efficacité du code en termes d'utilisation des ressources (temps, mémoire, tâches).
- Pratique d'utilisation de GitHub.
- Pratique de travail de groupe.
- Originalité de la démonstration.

## GCtronic Award

Un prix sera décerné au projet ayant obtenu la meilleure marque sur la base de ces critères d'évaluation.