

Travail Pratique #1

Robotique et Vision (420-KH5-LG)

Objectifs

Ce travail pratique a pour but :

- de vous familiariser davantage avec les commandes de contrôle du robot,
- d'implanter et de respecter le protocole de communication entre le contrôleur et l'ordinateur,
- de programmer une interface graphique pour le contrôle du robot à distance,
- d'implanter les méthodes et structures adéquates pour le contrôle d'un périphérique par le port série.

Outils utilisés

- Visual C++/C# pour la production de votre interface graphique et la programmation de l'application.
- Les commandes RAPL du robot pour les mouvements à effectuer.

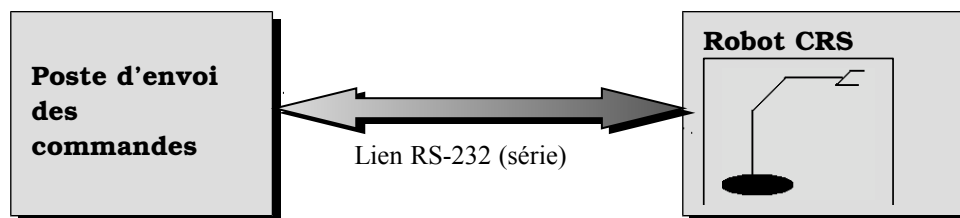
Travail à réaliser

On vous demande de développer une application qui permettra de contrôler le robot à distance à partir d'un poste de travail reliée au contrôleur du robot par lien série.

Dans un premier temps,

Vous devez donc contrôler le robot (mouvements semblables à ceux qu'offre le « Teach Pendant ») à distance, à partir d'un poste de travail (un ordinateur) par le biais d'un lien RS-232.

Voici la cellule de travail dans un contexte de contrôle par le biais du port série :



Vous devez simuler le poste d'envoi des commandes par le biais d'une interface graphique qui devra permettre à un opérateur de contrôler le robot à distance pour pouvoir effectuer des mouvements, enregistrer des points et reproduire des mouvements.

Partie robotique

Les objets de votre interface graphique devront permettre les fonctionnalités suivantes:

Mode manuel

Dans ce mode, l'utilisateur peut envoyer des commandes au robot comme si on les envoyait à partir du « teach pendant ». On veut donc que votre application dans ce mode puisse, entre autre :

- Permettre à l'opérateur d'envoyer directement une commande au robot.
- Bouger les 5 axes du robot indépendamment les uns des autres.
- Permettre l'arrêt d'urgence. Ceci arrête immédiatement tout mouvement du robot.. (commande « HALT »).
- Permettre d'effectuer un retour à la position « maison » (READY)
- Permettre le contrôle de la vitesse.
- Permettre d'enregistrer un point (une position) dans le contrôleur.
- Permettre l'ouverture et la fermeture des pinces.
- Faire afficher la liste des points dans le contrôleur. Ensuite faire déplacer le robot à ce point.(LLOC)
- Faire afficher la liste des tâches (programmes) présents dans le contrôleur, permettre de choisir un programme dans cette liste et faire exécuter ce dernier par le robot. (DIR, RUN)

Mode automatique

Dans ce mode, toutes les commandes effectuées par l'utilisateur à partir de l'interface graphique sont enregistrées (suggestion : dans une liste déroulante « listbox »). L'application permettra ensuite de refaire la séquence automatiquement en refaisant les commandes enregistrées.

BONUS

Un bonus de l'ordre de 5 % sera accordé si, en plus des fonctionnalités demandées, vous implantez la fonctionnalité suivante **au complet** :

- Planter un petit éditeur de texte (très minimal) qui permettra à l'utilisateur de :
 - **Récupérer ou sauvegarder un fichier de commandes.**
 - **Modifier un fichier de commandes (insérer ou détruire des commandes).**
 - **Exécuter ce fichier de commandes.**

Vous comprendrez que cette section n'est pas nécessairement vue en classe non plus. Il y a donc un peu plus de recherche à faire à ce niveau.

Barème de correction

Vous serez évalué en fonction des éléments fonctionnels que vous aurez implantés dans votre programme. Ainsi, le barème suivant sera employé :

Toutes les fonctionnalités implantées :	70 %
Chaque élément non fonctionnel à 100% :	-5 %
Structure du programme :	10%
Qualité de l'interface graphique	10%
Commentaires et documentation :	5%
Respect de l'échéancier :	5%
Bonus possible :	5%
	<hr/>
	105 %

Ce qui est à remettre et quand?

Vous envoyez dans la boîte Col.Net prévue à cet effet :

- Le **code source et l'exécutable** de votre projet en C++ ou C# (selon le cas) dans un format compressé (ZIP / RAR) dans la boîte de remise Col.NET qui sera prévue à cet effet.

Remarques :

- Assurez-vous que votre exécutable fonctionne et que le code source que vous m'envoyez compile adéquatement. Je ne veux pas être obligé de recompiler à chaque fois vos projets. Si je dois recompiler, j'enlève automatiquement 5% des points total.

Échéancier

Vous devrez, tout au long de ce travail, me montrer certains éléments de fonctionnement de votre programme en respectant l'échéancier suivant :

Interface graphique complété (version beta)	Jeudi 2 février 2017
Lien série avec le robot fonctionnel (donc quelques commandes devraient déjà fonctionner (READY, OPEN, CLOSE...)).	Jeudi 9 février 2017
Programme complet et remise finale	Vendredi 17 février 2017 avant 23:59