

TD n° 1

Système temps-réel

I) Etude d'un poste d'assemblage

L'atelier d'assemblage comprend les éléments suivants :

- 2 tapis roulants A et B commandés au pas à pas ;
- 1 tapis roulants A+B avançant en continue ;
- 1 poste d'usinage pouvant travailler 5 pièces simultanément ;
- 1 poste d'assemblage d'une pièce A avec une pièce B ;
- 1 plateau tournant à 4 bacs numérotés de 1 à 4 et d'un repère (chaque bac ne peut contenir qu'une pièce) ;
- 2 robots *Robot_A* et *Robot_B* dont les zones d'interventions sont les disques fins.

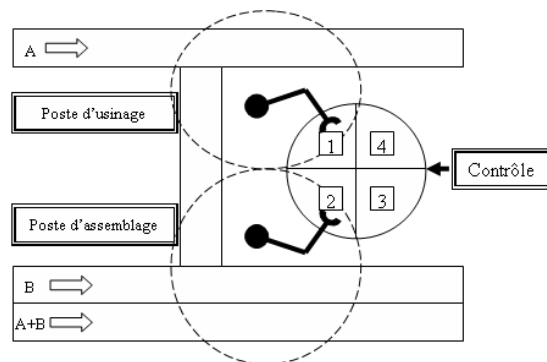


FIGURE 1 – schéma de l'atelier d'assemblage

Pour la suite du problème on ne s'intéresse qu'au fonctionnement des deux robots et plus particulièrement aux synchronisations d'enlèvement et de dépose d'éléments sur le plateau tournant, respectivement `Prendre()` et `Deposer()`. Chaque robot n'a accès qu'à un seul bac du plateau tournant.

Question 1. A quel type de problème appartient le sous-système composé du *Robot_A*, *Robot_B* et *Plateau tournant* ?

Question 2. Décrire de façon simple ce sous-système en synchronisant les deux robots à l'aide des primitives $P(.)$ et $V(.)$, dans le cas où : le *Robot_A* appelle uniquement la fonction `Deposer_Ra()` et le *Robot_B* appelle uniquement la fonction `Prendre_Rb()` ?

On va maintenant s'intéresser aux synchronisations entre les fonctions `Deposer_Ra()` et `Prendre_Rb()`.

Contexte :

- On dispose de la fonction `Poser_la_piece()` pour le *Robot_A* et `Enlever_la_piece()` pour le *Robot_B*;
- On dispose de la fonction `Examiner()` qui rend `LIBRE` ou `OCCUPE` à l'appelant (contenu du bac accessible);
- On dispose de la fonction `Tourner()` pour faire tourner le plateau suivant ce qui a été retourné par la fonction `Examiner()`.

Question 3. En utilisant des sémaphores/mutex pour la synchronisation, décrire **uniquement** les squelettes de programme des fonctions `Deposer_Ra()` pour *Robot_A* et `Prendre_Rb()` pour *Robot_B*. Avant de commencer, pensez à exprimer les contraintes sur les actions qui l'est possible (ou non) d'effectuer en parallèle.

Les squelettes de base de chaque fonction vous sont fournis, vous n'aurez donc qu'à rajouter l'utilisation adéquat (primitive $P(.)$ et $V(.)$) des sémaphores/mutex pour la synchronisation.

<pre>Deposer_Ra() { while(examiner() != LIBRE) { Tourner(); } Poser_la_piece(); }</pre>	<pre>Prendre_Rb() { while(examiner() != OCCUPE) { Tourner(); } Enlever_la_piece(); }</pre>
--	---