

AGRÉGATION D'INFORMATIQUE – session 2000

Admission

Épreuve de Travaux Pratiques

Option : Informatique des systèmes industriels

(préparation : 7 heures, Exposé et discussion : 1 heure dont 20 à 30 min d'exposé)

Mise en œuvre de la commande d'un poste de perçage à l'aide d'un exécutif temps réel

Objet de la manipulation

Il s'agit d'implémenter la commande d'un poste de perçage à partir d'une spécification fournie en utilisant les services d'un exécutif temps réel.

Environnement de travail

- Un PC de type Pentium
- Une carte d'interface entrées/sorties (type 8112 HG) avec sa documentation
- Une carte d'entrées protégées et une carte de sorties avec relais
- Un environnement de développement Turbo C++/ Borland C++
- Un exécutif temps réel NTR installé
- Boîtiers de simulation des entrées/sorties
- La documentation relative aux cartes et à l'exécutif utilisé

Description du poste de perçage

Il s'agit d'une machine de perçage comportant un plateau tournant qui dessert 3 postes de travail (voir figure 1) :

- Poste 1 : Chargement des pièces sur palettes
- Poste 2 : Serrage des palettes et perçage des pièces
- Poste 3 : Contrôle des pièces percées, puis évacuation. Le contrôle de perçage s'effectue par un testeur qui doit descendre en position basse si le trou est correctement percé. Si cela n'est pas réalisé, tout le système se bloque, testeur en position haute, de façon à ce que l'opérateur puisse enlever la pièce défectueuse avant de réarmer manuellement le système.

Un vérin permet la rotation de 120° du plateau extérieur supportant les pièces à usiner. Il permet aussi son indexation, c'est à dire son blocage précis après chaque rotation.

La mise en rotation et l'arrêt de la broche de la perceuse sont gérés automatiquement par l'unité de perçage et ne seront pas pris en compte par le programme de commande.

A l'état initial la machine doit répondre aux conditions suivantes :

- Les tiges des vérins A, B, C, D et E doivent être rentrées (fdca-, fdcb-, fdcc-, fdcd-, fdce-).
- Au moins une pièce doit être présente devant les postes (pp1 ou pp2 ou pp3) les détecteurs sont disposés sous le plateau tournant.

Figure 1 – Poste de perçage

Tableau des entrées-sorties

Description de la maquette

Les capteurs du poste de perçage seront simulés à l'aide de la boîte à boutons, les actionneurs seront simulés par les leds de la carte de sortie.

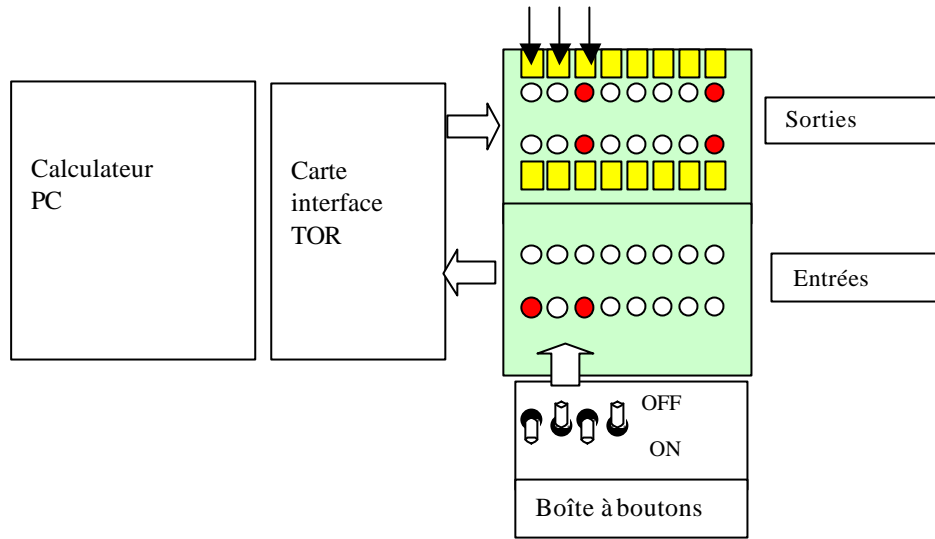


Figure 2 - Structure matérielle

Spécification du système

Un début de spécification du système par SA-RT est fourni. Nous nous intéressons essentiellement à l'implémentation du système.

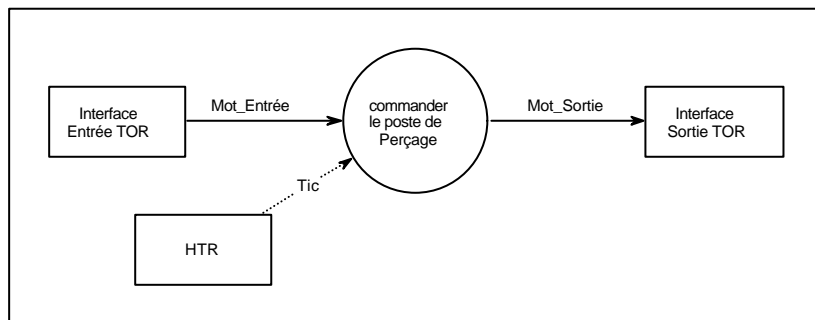


Diagramme de contexte

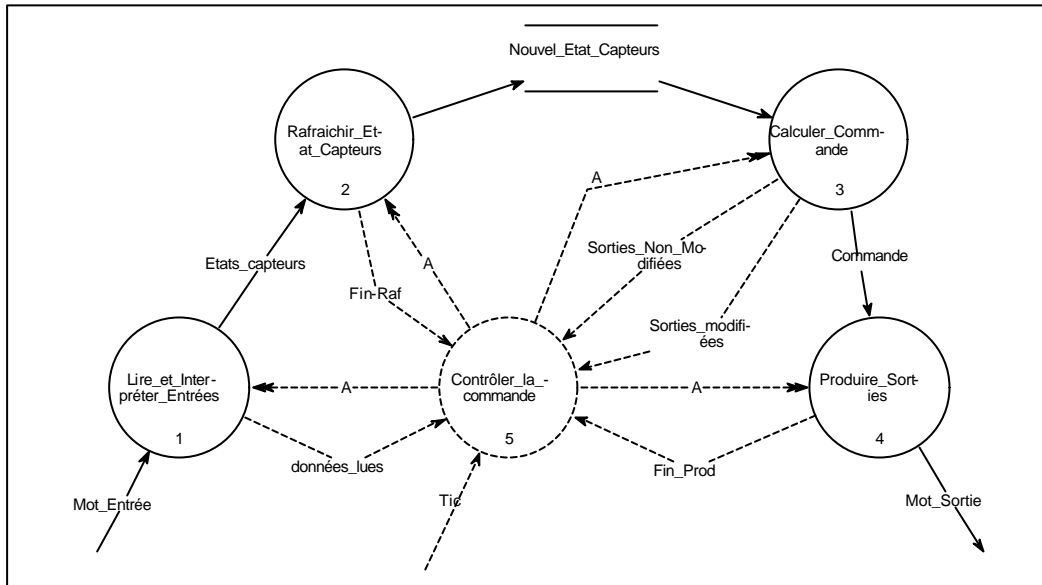


Schéma préliminaire – ST0 – Commander le poste de perçage

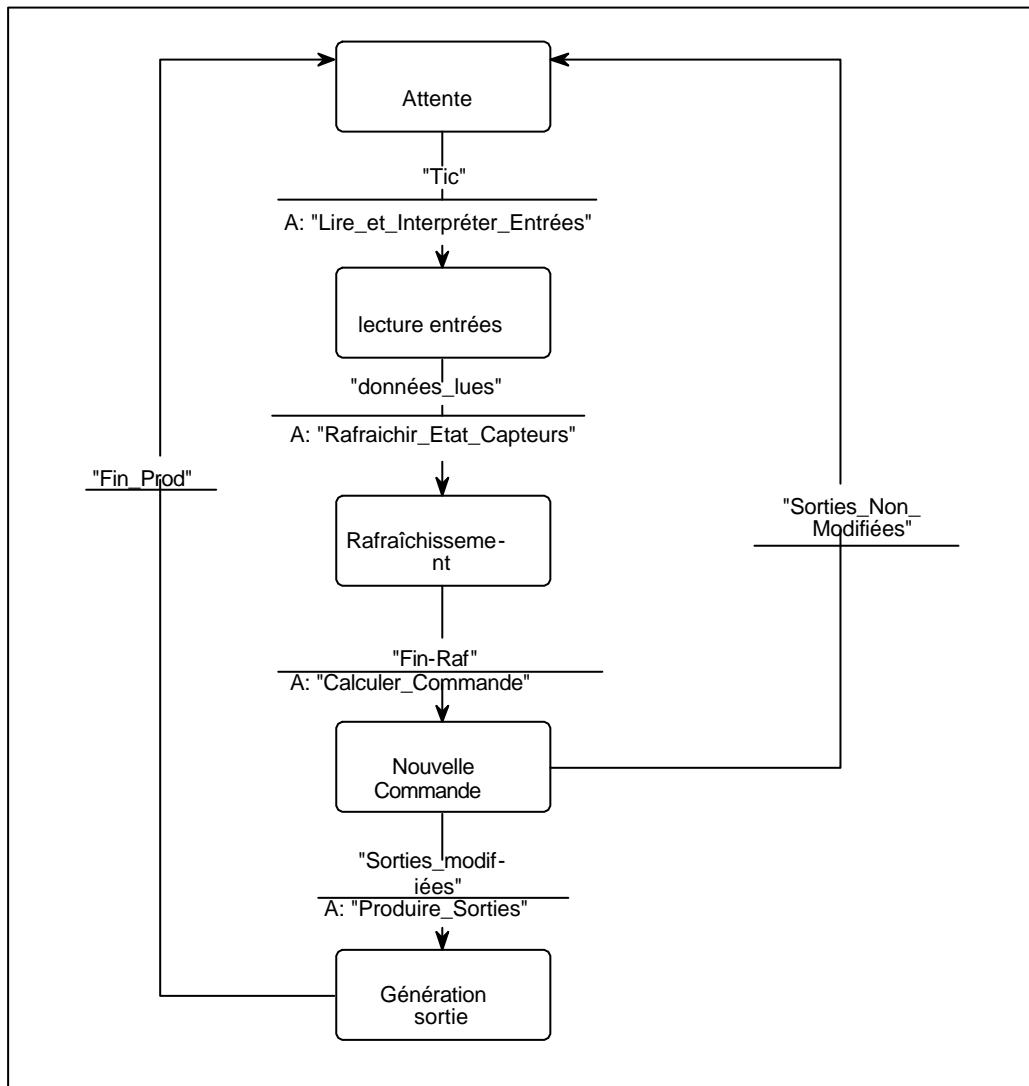


Diagramme Etats –Transition de ST 0.5 "Contrôler_la_commande"

(A = Activer)

Le processus terminal "Calculer_Commande" est spécifié à l'aide du Grafcet suivant :

Dictionnaire de données partiel

Name: Commande

Type: Discrete flow

Bnf: "a+" + "a-" + "b+" + "b-" + "c+" + "c-" + "d+" + "d-" + "e+" + "e-" + "f+"

@booléen=\vrai | faux\

@a+=boolean @a-=boolean @b+=boolean @b-=boolean @c+=boolean @c-=boolean

@d+=boolean @d-=boolean @e+=boolean @e-=boolean @f+=boolean

Name: Etats_Capteurs

Type: Discrete flow

Bnf: "bpm"+"bprearm"+"pp1"+"pp2"+"pp3" + "fdca"+"fdca-" + "fdcb"+"fdcb-" + "fdcc"+"fdcc-" + "fdcd"+"fdcd-" + "fdce"+"fdce-" + "fdcf+"

@booléen=\vrai | faux\

@bpm=boolean @bprearm=boolean @pp1=boolean @pp2=boolean

@pp3=boolean @fdca+=boolean @fdca-=boolean @fdcb+=boolean

@fdcb-=boolean @fdcc+=boolean @fdcc-=boolean @fdcd+=boolean

@fdcd-=boolean @fdce+=boolean @fdce-=boolean @fdcf+=boolean

Name: Fin-Raf

Type: Control flow

Comment: Généré lorsque Etat_Capteurs a changé : dans ce cas le processus "Calculer_Commande" sera activé.

Name: Mot_Entrée

Type: Discrete flow

Bnf: octet_haut + octet_bas

@octet_haut=octet

@octet_bas=octet

@octet=/suite de 8 bits b0 .. b7/

Comment: Octet_haut Octet_bas

b0=bpm b0= fdcb+

b1=bprearm b1= fdcb-

b2=pp1 b2= fdcc+

b3=pp2 b3= fdcc-

b4=pp3 b4= fdcd+

b5=fdca+ b5= fdcd-

b6=fdca- b6= fdce+

b7=fdcf+ b7= fdce-

Name: Mot_Sortie

Type: Discrete flow

Bnf: octet_haut + octet_bas

@octet_haut=octet

@octet_bas=octet

@octet=/suite de 8 bits b0 .. b7/

Comment: Octet_haut Octet_bas

b0= b0= b+

b1= b1= b-

b2= b2= c+

b3= b3= c-

b4= b4= d+

b5=a+ b5= d-

b6=a- b6= e+

b7=f+ b7= e-

Name: Nouvel_Etat_Capteurs

Type: Store

Bnf: /etat actuel des capteurs/

Comment: Modifié par "Rafraichir_Etat_Capteurs " lorsque Etat capteurs a changé

Travail demandé

Première partie

- 1) En se basant sur la spécification fournie, identifier les tâches qui permettent d'implémenter le système décrit ainsi que les communications entre les tâches.
- 2) Réaliser les modules logiciels qui implémentent l'architecture logicielle identifiée précédemment. L'implémentation sera faite en utilisant les services de l'exécutif temps réel NTR. On veillera à réaliser des modules indépendants que l'on pourra tester progressivement et individuellement.

Deuxième partie

En vous inspirant des développements réalisés dans cette application, élaborer un sujet de travaux pratiques destinés à des étudiants de deuxième année de formation technologique. Proposer le texte à fournir, les pré requis supposés ainsi que les composants matériels et logiciels à mettre à leur disposition.