

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 d'une commande de processus à l'aide d'un exécutif temps réel : Commande des feux d'une automobile

Objet de la manipulation

A partir de la spécification de l'application fournie il s'agit d'implémenter la commande du processus 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
- Un exécutif temps réel NTR installé
- Un boîtier de simulation des entrées sorties
- La documentation relative aux cartes et à l'exécutif utilisé

Description de l'application

On dispose sur une automobile de 4 commandes tout ou rien indépendantes symbolisées : v, c, r et a qui permettent, sous le contrôle d'un système que nous allons étudier, les allumages respectifs:

- des veilleuses V
- des feux de croisement C (2 phares)
- des feux de route R (2 phares)
- des antibrouillards A (2 phares)

Les règles de fonctionnement sont les suivantes :

- 1) Les veilleuses ne sont pas considérées comme des phares.
- 2) Quatre phares ne peuvent être allumés simultanément (exemple route et croisement). En cas de commande simultanée, priorité est donnée aux feux de croisement puis aux antibrouillards.
- 3) Les veilleuses peuvent être allumées seules, mais les allumages respectifs de C, A ou R entraînent obligatoirement l'allumage de V.

Nous nous proposons de réaliser une maquette du système avec un calculateur de type PC muni d'une carte d'entrées/sorties TOR (type 8112 HG).

Description du matériel

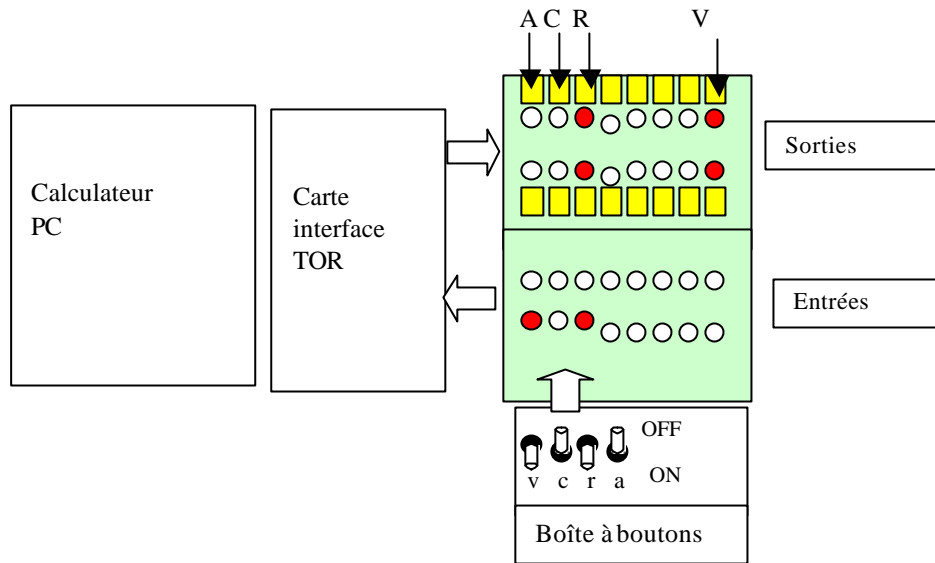


Figure 1- Structure matérielle

Les entrées (v, c, r, a) et les sorties (V, C, R, A) sont connectées selon le schéma figure 1.

Un début de spécification du système par SA-RT ainsi que le modèle d'architecture logicielle sont fournis. Nous nous intéressons essentiellement à l'implémentation du système.

Spécification du système

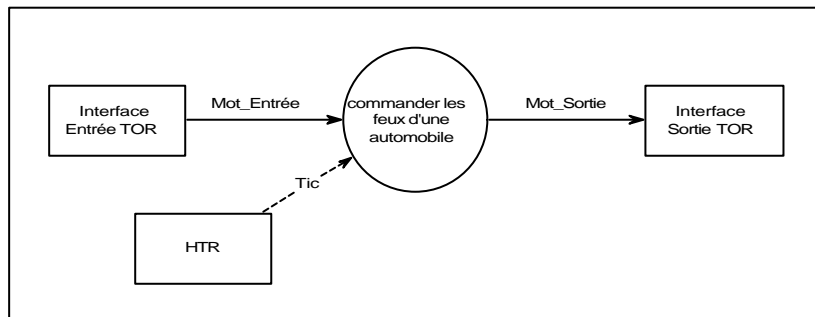


Diagramme de contexte

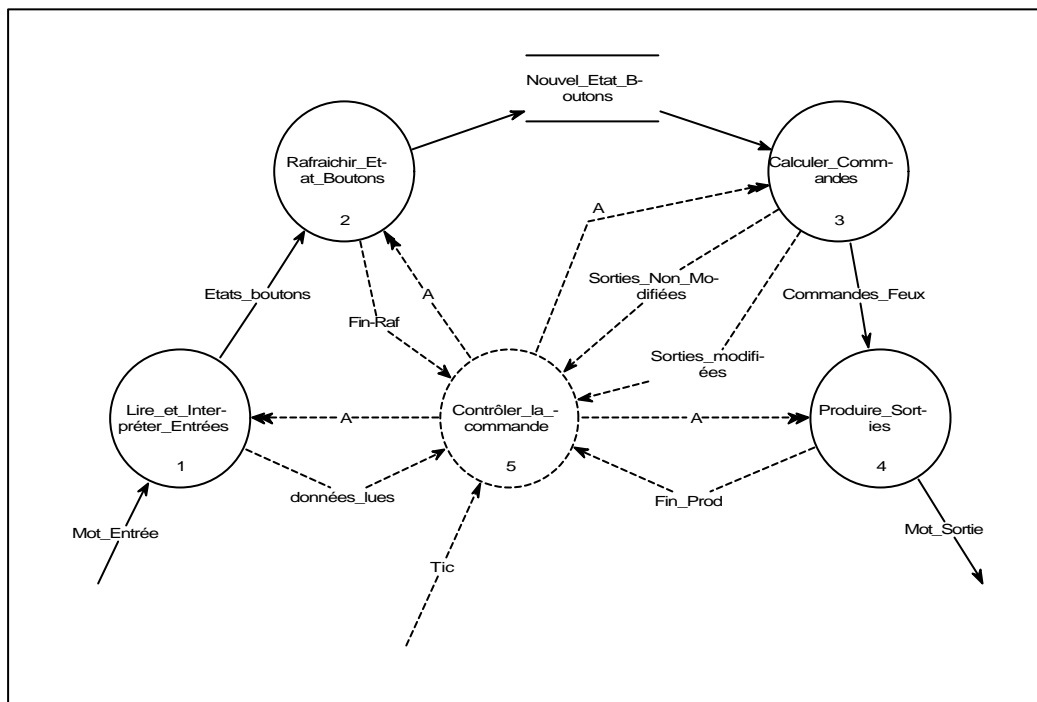


Schéma préliminaire – ST0 – Commander les feux d'une automobile

Dictionnaire de données partiel

Name: Commandes_Feux

Type: Discrete flow

Bnf: v+c+r+a

@booléen=\vrai | faux\

@V=booléen

@C=booléen

@R=booléen

@A=booléen

Name: données_lues

Type: Control flow

Name: Etats_boutons

Type: Discrete flow

Bnf: v+c+r+a

@booléen=\vrai | faux\

@v=booléen

@c=booléen

@r=booléen

@a=booléen

Name: Etat_Courant

Type: Discrete flow

Bnf: Nouvel_Etat_Boutons

Name: Fin-Raf

Type: Control flow

Comment: Généré lorsque Etat_Boutons a changé : dans ce cas le processus "Calculer_Commandes" sera activé.

Name: Fin_Calcul
Type: Control flow

Name: Fin_Prod
Type: Control flow

Name: Mot_Entrée
Type: Discrete flow
Bnf: octet
 @octet=/suite de 8 bits b0 .. b7/
Comment: b0=a
 b1=c
 b2=r
 b7=v

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
 b0=A
 b1=C
 b2=R
 b7=V
 Octet_bas
 b0=A
 b1=C
 b2=R
 b7=V

Name: Nouvel_Etat_Boutons
Type: Store
Bnf: /etat actuel des boutons/
Comment: Modifié par "Rafraichir_Etat_Boutons " lorsque Etat
 bouton a changé

Name: Sorties_modifiées
Type: Control flow

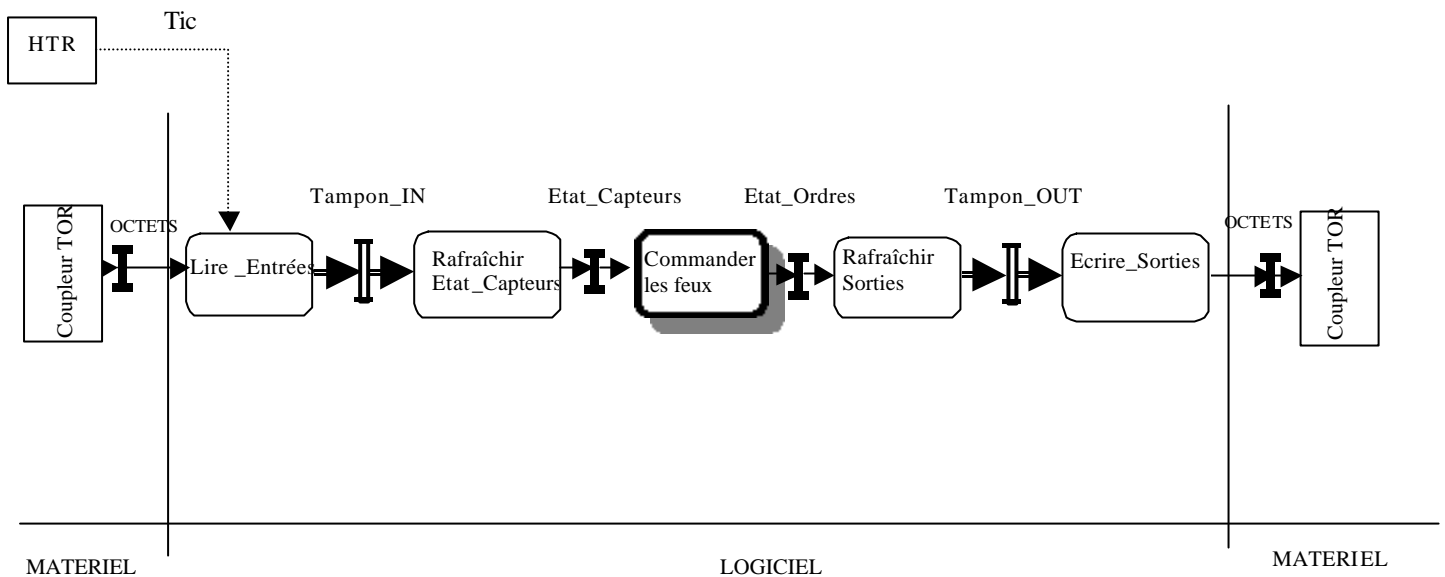
Name: Sorties_Non_Modifiées
Type: Control flow

Name: Tic
Type: Control flow

Le processus terminal de contrôle/commande des feux sera spécifié par une table de vérité et par les relations algébriques simplifiées qui en découlent. On notera que le système est purement combinatoire (*Revoir les notions de systèmes combinatoires et séquentiels*).

Conception et réalisation

Architecture logicielle proposée



Légende :

— I —> Données échangées entre tâches ou entre environnement et tâches

— II —> Passage de données entre tâches par boîte à lettre

□ Tâche

- - - - -> Interruption matérielle

Figure 2 - Architecture logicielle

Travail demandé

Première partie

- 1) Etudiez le modèle de spécification fourni et explicitez par un diagramme états-transitions ou un grafcet le processus de contrôle "contrôler_la_commande" défini dans le schéma préliminaire.
- 2) Après avoir déterminé les relations algébriques entre les entrées et les sortie (calcul combinatoire des sorties en fonction des entrées) de la transformation de données "Calculer_Commandes", proposez une description algorithmique de ce processus terminal.
- 3) Réaliser les modules logiciels qui implémentent l'architecture logicielle proposée. On veillera à réaliser des modules indépendants que l'on pourra tester progressivement et individuellement.
- 4) On désire intégrer au système une commande supplémentaire warning (un bouton supplémentaire) noté "w" qui commandera le clignotement des feux de changement de direction (à rajouter au système). Chaque fois que le bouton w sera activé, tous les feux devront être désactivé et les feux de changement de direction devront commencer à clignoter.

- 3.a introduire les modifications nécessaires au modèle de spécification et au modèle d'architecture logicielle.
- 3.b introduire les modifications nécessaires au modèle de commande proposé.
- 3.c effectuer les modifications nécessaires sur les modules logiciels réalisés pour prendre en compte ce nouveau fonctionnement.

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 prérequis supposés ainsi que les composants matériels et logiciels à mettre à leur disposition.