TP 2: Programmation Grafcet avec gestion de ressources
Automatismes Industriels

Vincent Laurain

1 Introduction

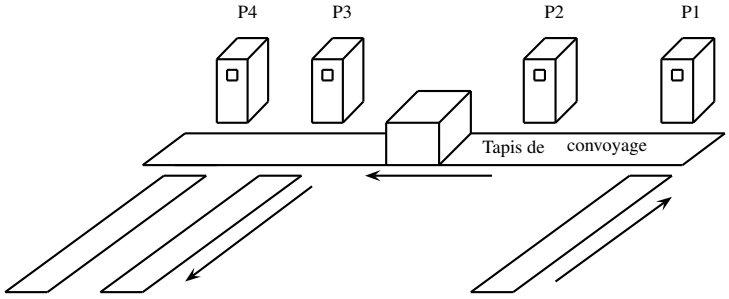
Le but dans ce TP est d'une part de concevoir un GRAFCET simple et de se familiariser avec les Grafcets programmés sous Unity.

La durée de ce TP est de 6 heures. Le compte rendu est à rendre sous ARCHE.

Description du Système

Système

Le système auquel nous nous intéressons est un convoyeur de bande représenté dans la figure suivante. Le but est ici de gérer le tapis de convoyage présent sur la maquette selon le cahier des charges décrit ci dessous. Les capteurs disponibles sont les capteurs P1 à P4. Les actionneurs seront la mise sous tension du variateur puis la commande du moteur à gauche ou à droite.



Cahier des charges

Nous ne nous intéressons ici qu'à la gestion de la bande de convoyage centrale. Les autres tapis sont considérés comme étant gérés par un autre automate (vos mains et votre tête sur la maquette).

Lorsqu'une pièce arrive au poste 1, la pièce est convoyée pour un contrôle en poste 2. Elle devra être acheminée au poste 3 ou 4 selon la disponibilité de ces postes (qui seront donc considérés **comme des ressources**) :

- Si le poste 3 est libre, la pièce attend 3 secondes en position 2 puis se dirige au poste 3. **Le poste 3 est prioritaire dans le cas où les deux postes sont libres.**
- Si le poste 4 est libre, la pièce attend 2 secondes en position 2 et se dirige au poste 4.
- Si aucun des postes n'est libre la pièce attend au poste 2 jusqu'à ce qu'un poste se libère.

Les postes 3 et 4 sont des postes de soudure. Lorsque la pièce arrive au poste concerné, le soudeur libère la pièce du tapis manuellement. Le cycle peut recommencer.

La gestion des ressources : Les postes 3 ou 4 sont considérés comme occupés dès que la pièce arrive devant leur poste respectif. Les postes sont considérés comme libre après que le soudeur appuie sur un bouton d'acquittement (Bp droit et Bp Gauche respectivement). **Une LED verte s'allume lorsque le**

poste est occupé.

NB : En ce qui concerne les temporisations, et par soucis pédagogique, on souhaite ici réaliser la tempo du poste 3 avec un bloc tempo, et une temporisation avec un bloc compare (voir annexe de ce document). Lisez bien l'annexe de ce document qui contient de nombreuses et précieuses informations.

Entrées et Sorties

- Référez vous à l'annexe du précédent TP pour les différentes entrées/sorties.
- **Attention : pensez à alimenter la sortie 15 de la carte 2 sui correspond à la mise sous tension du variateur (le moteur ne tournera pas sinon).**

2 Tache 1

Réalisez le GRAFCET sur papier (faites le valider) et implémentez le sur Unity.

La programmation de grafcet en langage SFC Unity

- **Etape 1 : Créer une nouvelle section pour laquelle le langage de programmation est SFC**. Vous verrez que le dessin du GRAFCET est relativement aisé et facile d'accès. Vous pourrez implémenter toutes les étapes, transitions et divergences ou convergences habituelles.
- **Etape 2 :** Une fois le GRAFCET tracé, vous allez devoir renseigner les différentes réceptivités liées aux transitions et les actions liées aux étapes. Pour cela, un **double clic** sur un des éléments et :
  - **Dans une transition :** Soit vous allez définir une variable qui sert directement de réceptivité (par exemple %I0.2.3), soit vous allez pouvoir définir un schema (Section Transition que vous pouvez par exemple écrire en LADDER). La dernière option vous permettra de décrire des réceptivités plus compliquées ou des équations logiques.
  - **Dans une étape :** De même vous allez pouvoir directement définir une variable ou une section à activer lorsque l'étape devient active. Vous pourrez choisir un ensemble de conditions (N : Normal, R : Raz de l'action précédente, S : Mémorise l'action qui sera désactivée par une étape "R", L : Action temporisée...) qui vont décider du déclenchement de ces actions. **Pour que la variable soit prise en compte, il faut la choisir, puis ensuite cliquer sur "Nouveau".**

Annexes et autres aspects nécessaires

- Dans ce Grafcet, vous aurez besoin de temporisations. De manière générale, toute sorte de fonctions (compteurs, lecture dans un autre automate, tempo...) peuvent être intégrées à un schéma en LADDER. Pour insérer un bloc fonctionnel dans un schéma ladder, vous pouvez faire un simple clic droit "assistant de saisie FBB". Alors s'ouvre une fenêtre vous demandant un nom de bloc à insérer. Vous avez également à coté un symbole "..." vous permettant de chercher dans une liste. En cliquant sur ces points s'ouvrent tous les objets disponibles dans la bibliothèque "V8.0". En général, pour connaître le nom d'un bloc donné, une recherche sur votre moteur internet préféré fera l'affaire. Ici, nous aurons principalement besoin de temporisations (qui s'appellent TON ou TOF) et de compteurs (qui s'appellent CTU). Cependant, il faudra faire attention puisque dans unity pro, **ces blocs FBB ne peuvent pas être intégrés directement dans les transitions ou étapes**. Vous devrez faire preuve d'inventivité.
- En ce qui concerne les étapes (par exemple, l'étape S_1_2), plusieurs grandeurs sont accessibles :
  - S_1_2.x qui est l'état de l'étape : active (S_1_2.x=1) ou non active (S_1_2.x=0) ;
  - S_1_2.t qui est le temps d'activité de l'étape. Cette grandeur est un objet temps. Les objets temps se manipulent de manière particulière dans Unity Pro. Par exemple, la commande S_1_2.t=T#5s attribue a valeur "5 secondes" à S_1_2.t. Notez simplement la notation utilisant le signe # .
- Un automate dispose de mémoire interne. On peut utiliser la mémoire interne de l'automate pour stocker de l'information. Pour simplement créer une variable, il suffit de la définir dans les variables

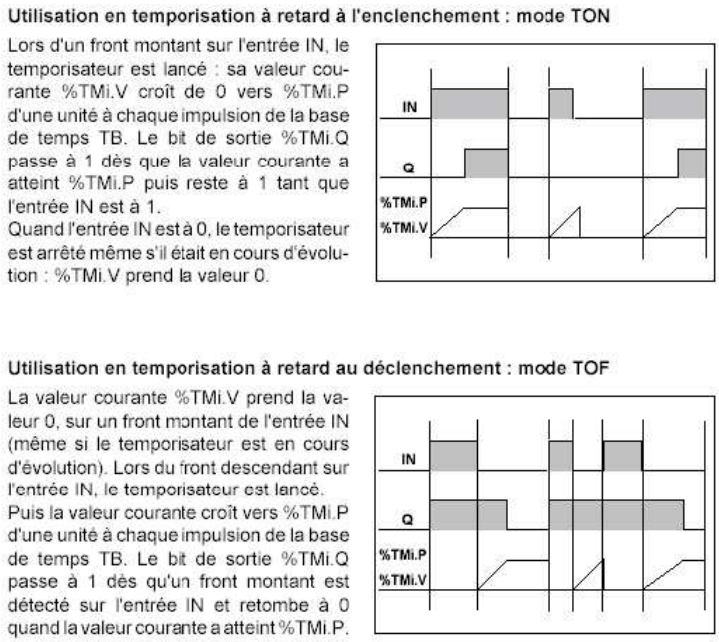


FIGURE 1 – Fonctionnement des temporisations

élémentaires et préciser son type (vous aurez besoin, ici par exemple, de stocker l'information si un poste est occupé ou non). Vous pourrez déclarer par exemple une variable booléenne. De plus, **si vous le souhaitez,** vous pouvez spécifier exactement l'adresse à laquelle cet objet doit se trouver dans la mémoire (ça peut être pratique si vous voulez faire communiquer deux automates ensemble). Pour cela, vous pouvez spécifier une adresse du type :
  - %M0 : bit mémoire à l'adresse 0
  - %MW0 : mot mémoire (pour stocker des entiers ou des réels) à l'adresse 0 (%MW0.5 veut dire le 5eme bit du mot 0).
- Vous aurez également besoin de deux blocs :
  - Le bloc Compare pour comparer deux valeurs : par exemple, vous pouvez noter $C < 5$ et alors la valeur de C sera comparée à 5 et la sortie du Comparateur passera à 1 tant que C est inférieur à 5.
  - Le bloc Operate : ce bloc permet d'effectuer de simples opérations. Faites simplement attention à la nomenclature qui recquiert l'utilisation de " :". Par exemple $C := C + 5$ incrémentera la variable C de 5.

Tache 2

Pour pousser un peu les choses, on vous demande ici de stocker les informations relatives aux performances de chaque ouvrier du poste P3 et P4. On vous demande de stocker (dans un entier), le temps total passé à travailler par chaque opérateur. Le nombre total de pièce traité par chaque opérateur. Le temps passé à traiter la pièce courante. Stockez ces variables à des adresses particulières :
- %MW3 : nombre de pièces traitées OP3
- %MW4 : temps total passé OP3
- %MW5 : temps moyen par pièce OP3
- %MW6 : temps passé sur la dernière pièce OP3
- %MW7 : nombre de pièces traitées OP4
- %MW8 : temps total passé OP4
- %MW9 : temps moyen par pièce OP4
- %MW10 : temps passé sur la dernière pièce OP4

Modifiez votre GRAFCET de sorte à ce que lorsque les deux postes sont libres, la pièce aille au poste qui a traité le moins de pièce (vive l'égalité au travail !). Enfin allumez la LED rouge à l'operateur si son temps de traitement est supérieur à 5 secondes

Tache 3

Nous allons ici étudier les aspects de communication entre automates. Pour ceci, nous allons essayer d'aller capturer les données opérateur des automates voisins en utilisant le réseau Ethernet. Pour cela, vous aurez besoin du bloc fonction FBB s'appelant READ_VAR. Vous allez devoir lire la documentation technique pour comprendre un peu comment il fonctionne. Ensuite pour vous éviter trop de déboires en bugs et autres erreurs, voici quelques détails qui pourraient vous aider :

- Pour entrer l'adresse, vous aurez besoin d'utilise la fonction ADDM (google est votre ami).
- Le type d'objet à lire est %MW
- Pour GEST et RECP, vous aurez besoin **d'un tableau d'entier**. Vous pouvez directement déclarer un tableau dans les variables élémentaires. Par exemple, pour déclarer un tableau de booléen, vous pouvez déclarer `ARRAY[1..4] OF BOOL`.
- Pour résoudre l'erreur 1208 (oui vous l'aurez), allez dans outils → options de projet → variables et cochez "Autoriser les tableaux dynamiques" et "variables de tableau représentées directement".
- Enfin, la lecture de ces mots, se déclenchera au front montant de l'appui sur marche.
- Il faudra mettre le 3eme mot du tableau GEST à 20 (correspondant au timeout).