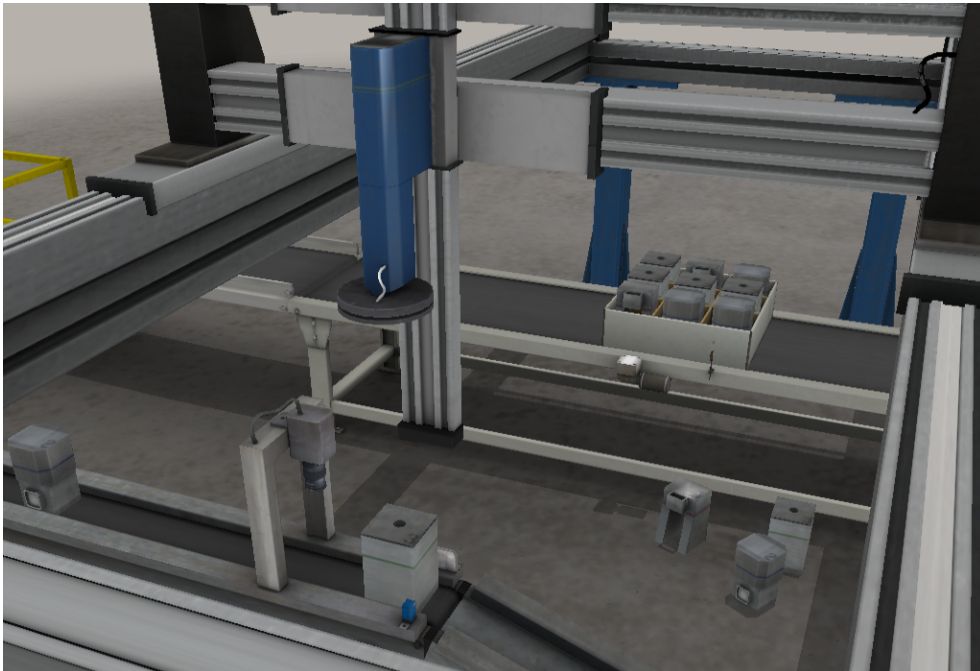




TP capteurs
Commande d'un robot Pick and Place

Valentin DOSIAS, Maxence NEUS



Novembre 2021

Contents

1	Introduction	2
2	Une pièce par boîte	3
3	Encapsulation des tâches de base	4
4	Remplissage d'une boîte	5
4.1	Nouveau programme de déplacement	5
5	Tri par type de pièce	8
5.1	Nouvelle admission de pièces	8
5.2	Ajustement du programme de déplacement	9
6	Conclusion	10

1 Introduction

Le TP consiste à concevoir et implémenter le système de commandes d'un robot Pick and place. La partie commande se fait via l'automate Schneider Modicon M340. La partie opérative est simulée sur l'ordinateur via un logiciel de simulation 3D appelé ITS PLC Professional Edition. Pour la commande du robot, nous utilisons des graficets que nous concevons grâce au logiciel Unity Pro. L'objectif de commande finale est de remplir totalement la caisse en ayant qu'un seul type de pièces par colonne. Lorsqu'une colonne est remplie, les pièces en trop doivent être évacuées. Pour ce faire nous avons d'abord pris en main le logiciel de simulation en activant nous-mêmes les différents actionneurs pour voir quel capteur est déclenché et à quel moment. Ensuite nous avons fait un programme pour que le robot mette une seule pièce par caisse. Une fois que la pièce est mise, la caisse est évacuée. Ensuite, nous avons modifié ce programme pour que la caisse soit remplie en entière et ensuite évacuée. Pour finir, nous avons ajouté la détection de chaque type de pièce pour remplir les colonnes en fonction du type de pièce.

2 Une pièce par boîte

Tout d'abord nous avons fait un grafcet pour prendre une pièce et le mettre dans la case (C0;L1). Une fois que la pièce est posée dans la caisse, la caisse peut-être évacuée. Nous avons donc conçu le grafcet suivant.

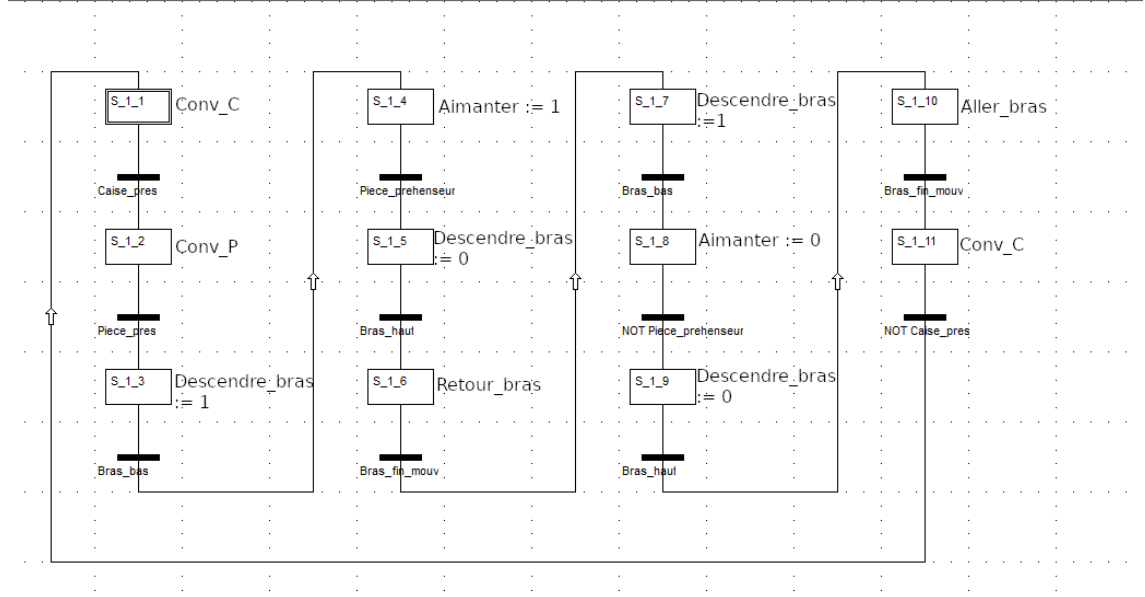


Figure 1: Une pièce par boîte

En premier lieu, nous attendons une caisse en activant le convoyeur des caisses (S_1.1 : Conv_C). Une fois que la caisse est là, le convoyeur est automatiquement désactivé. Nous attendons ensuite une pièce donc nous activons ensuite le convoyeur pour les pièces (S_1.2 : Conv_P). Une fois que la pièce est présente, le convoyeur est désactivé. Nous supposons dans tout le reste du TP que lorsque nous commençons notre programme, le bras est toujours dans sa position de repos soit en (C0;L0). La pièce étant là nous pouvons descendre le bras et le maintenir en bas tant que la pièce n'est pas prise par le préhenseur (Descendre_Bras:=1). Ensuite nous prenons la pièce avec le préhenseur, il faut maintenir l'aimantation tant que le bras n'est pas arrivé à la position au-dessus de la caisse (aimanter := 1). Le bras ne doit pas se déplacer en position basse donc nous le remontons pour qu'il puisse aller jusqu'à la caisse (Descendre_bras := 0). Ensuite nous faisons aller le bras jusqu'à la position (C0;L1) (Retour_bras), nous descendons donc le bras (Descendre_bras := 1) pour poser la pièce sans la lâcher de haut. Nous pouvons maintenant poser la pièce (Aimanter := 0), remonter le bras (Descendre_bras

$:= 0$) et faire revenir le bras à sa position de repos en $(C0;L0)$. Pour finir nous faisons partir la caisse, il faut donc attendre que le capteur de présence de pièce repasse à 0 (NOT Caise_pres). Une fois la pièce évacuée le grafcet peut recommencer.

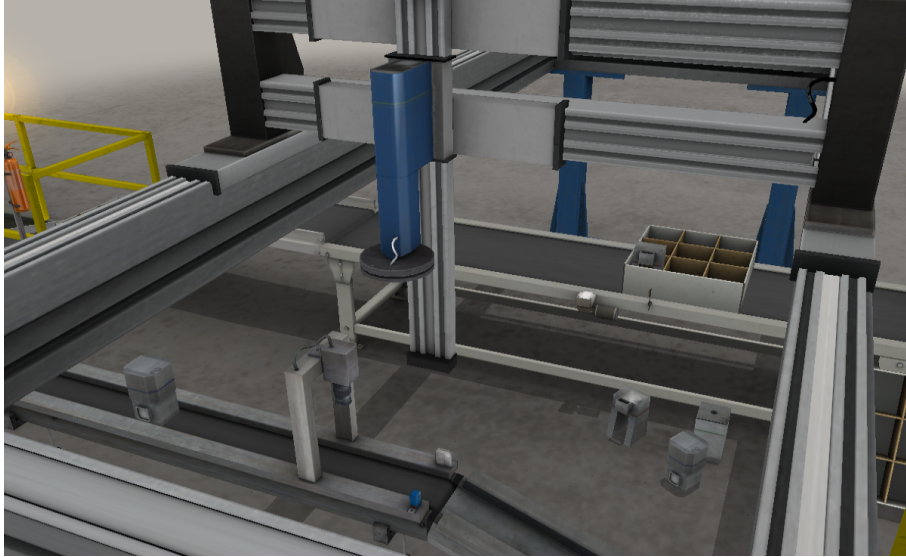


Figure 2: Une pièce par boîte

3 Encapsulation des tâches de base

Pour ne pas avoir un grafcet général trop lourd, nous avons fait des macro-étapes pour les tâches de base pouvant être regroupées en une seule fonctionnalité.

Préhension Nous avons réuni la tâche du préhenseur qui descend le bras saisit la pièce et remonte le bras dans la macro-étape Pick-Up.

Dépose de pièce La macro-étape Place va descendre le bras, poser la pièce et remonter le bras.

Parallelisation Nous avons trouvé un défaut à ce grafcet. Le défaut est qu'il faut attendre que la caisse soit bien présente pour envoyer une pièce, ce qui dans cette utilisation parait assez long. Nous avons donc choisi de gérer la présence des pièces avec un autre grafcet.

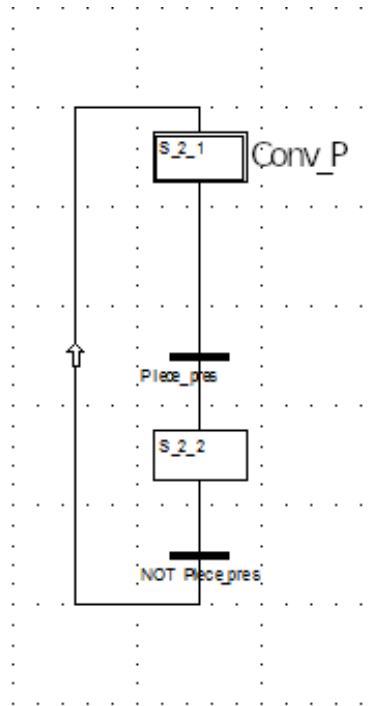


Figure 3: Admission de pièces en parallèle

Le grafcet est assez simple, nous activons le convoyeur, une fois que la pièce est arrivée, le convoyeur s'arrête et attend que la pièce ne soit plus présente, ce qui équivaut au préhenseur qui saisit la pièce.

4 Remplissage d'une boîte

4.1 Nouveau programme de déplacement

Logique de gestion des pièces Pour éviter de faire 8 macro-étapes quasi identiques avec les différents cas de positionnement du bras à la position de la case ou de la position initiale, nous utilisons un compteur prenant des valeurs de 0 à 8. Pour chaque valeur est défini des valeurs pour le déplacement sur les lignes et sur les colonnes. Pour se rendre à la case, nous utilisons un certain nombre de fois les actionneurs Retour_bras et Droite_bras et pour le retour à la position initiale nous utilisons ce même nombre de fois les actionneurs Aller_bras et Gauche_bras.

Algorithme de positionnement Pour déplacer la pièce au bon endroit, nous utilisons les variables décrites dans la partie précédente. Nous les util-

isons avec le grafcet suivant.

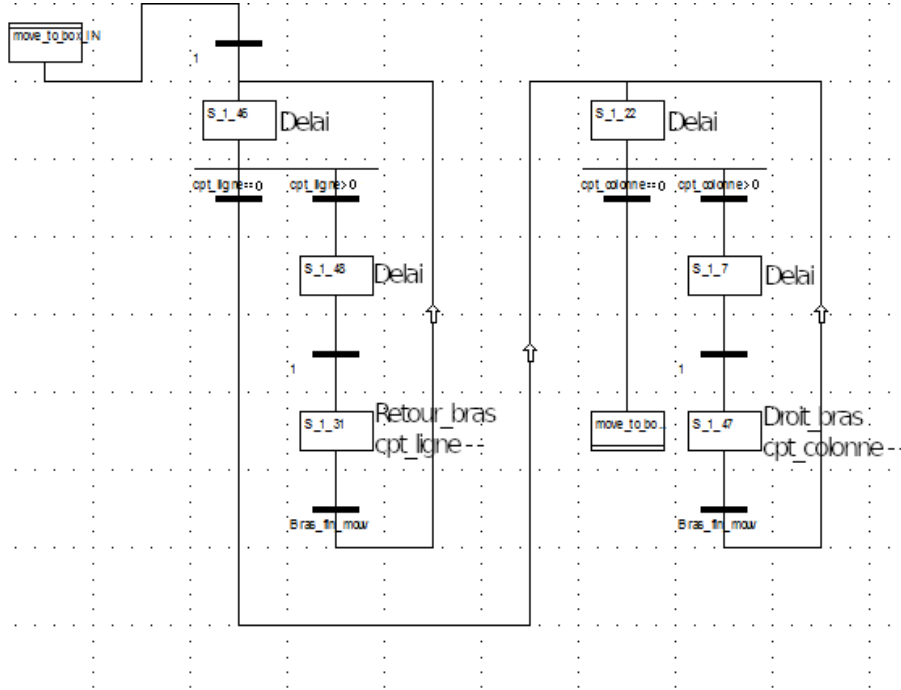


Figure 4: Algorithme de positionnement

Tout d'abord nous gérons les déplacements sur les lignes. Si les compteurs ont atteint la bonne valeur nous pouvons passer au déplacement sur les colonnes sinon nous faisons un déplacement sur la ligne en incrémentant le compteur. Nous faisons une pause de 3 secondes après le déplacement pour s'assurer que le capteur de fin de mouvement soit bien réinitialisé, ensuite nous vérifions que les valeurs du compteur et recommençons tant qu'il n'est pas arrivé à la bonne ligne. Pour les déplacements sur les colonnes le principe est le même. La macro-étape pour le retour à la position initiale est la même, seuls les actionneurs changent. Pour se rendre à la case, nous utilisons un certain nombre de fois les actionneurs Retour_bras et Droite_bras et pour le retour à la position initiale nous utilisons ce même nombre de fois les actionneurs Aller_bras et Gauche_bras.



Ensuite la pièce est amenée au-dessus de la case correspondante grâce à la macro-étape `move_to_box`, puis posée dans la case avec la macro-étape `Place`. Le bras est ensuite amené à sa position initiale avec la macro-étape `...`. Ensuite grâce au compteur on peut vérifier si la caisse est pleine ou non. Si elle est pleine, elle va être évacuée sinon une nouvelle pièce va être placée dans la caisse.


```

case cpt of
  0 : cpt_ligne := 1; cpt_colonne := 0;
  1 : cpt_ligne := 2; cpt_colonne := 0;
  2 : cpt_ligne := 3; cpt_colonne := 0;
  3 : cpt_ligne := 1; cpt_colonne := 1;
  4 : cpt_ligne := 2; cpt_colonne := 1;
  5 : cpt_ligne := 3; cpt_colonne := 1;
  6 : cpt_ligne := 1; cpt_colonne := 2;
  7 : cpt_ligne := 2; cpt_colonne := 2;
  8 : cpt_ligne := 3; cpt_colonne := 2;
end_case;

```

Figure 6: Init_cpts.st

5 Tri par type de pièce

5.1 Nouvelle admission de pièces

Une fois la boîte remplie sans ordre particulier des pièces, il faut maintenant trier les pièces par type. Chaque type doit correspondre à une colonne particulière. Nous avons choisi de mettre les pièces de type 1 dans la colonne C0, les pièces de type 2 dans la colonne C1 et les pièces de type 3 dans la colonne C2.

Nous reprenons la logique du compteur précédent ou cette fois pour le compteur de 0 à 2 ce sont les pièces de type 1, pour le compteur de 3 à 5 ce sont les pièces de type 2 et pour des valeurs de 6 à 8 ce sont les pièces de type 3. Chaque valeur de compteur à un nombre de déplacement défini pour que les pièces soit placés au bon endroit. Nous utilisons ces valeurs pour faire aller le bras vers la bonne case et aussi le faire revenir à la position initiale.

Nous avons aussi des compteurs pour les colonnes nous permettant de savoir si elles sont pleines ou non. Ces compteurs nous permettent aussi d'évacuer les pièces lorsque la colonne correspondante est pleine. Cela nous a donné le grafcet suivant :

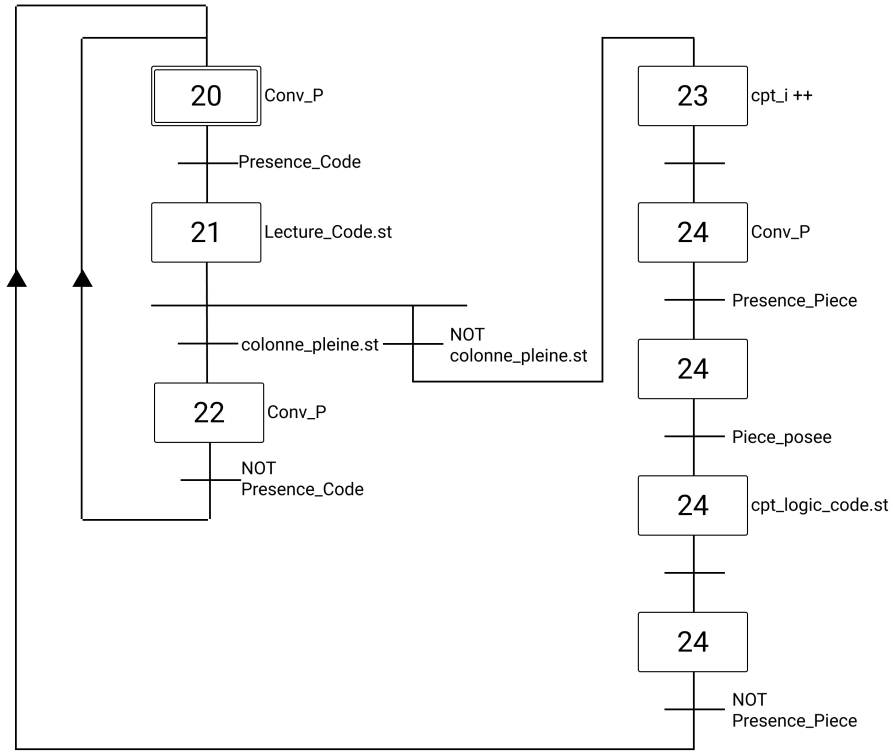


Figure 7: Admission finale

Tout d'abord nous activons le convoyeur jusqu'à ce que le détecteur de type de pièce (presence_code) nous envoie un signal ensuite selon le type si la colonne est pleine, nous réactivons le convoyeur pour évacuer celle-ci et le grafset revient à l'étape initiale et nous attendons la détection d'un code. Si la colonne correspondante n'est pas pleine, nous activons le convoyeur jusqu'à ce que la pièce soit bien positionnée (Piece_pres). Nous incrémentons aussi les compteurs de colonne. Une fois la pièce saisie par le préhenseur, nous attendons la nouvelle détection d'un code de pièce.

5.2 Ajustement du programme de déplacement

Le grafset est ici sensiblement identique à celui de la partie précédente à la différence près que l'on initialise 2 compteurs pour lignes et colonnes. On réalise alors le case sur le code_piece une seule fois au lieu de 2 ce qui permet d'éviter que le code ou les compteurs de pièces ne changent pendant le mouvement du bras ce qui, avec le grafset précédent, rendrait impossible le retour en position initiale.



Figure 8: Remplissage d'une boîte triée

6 Conclusion

Pour conclure, notre grafacet final répond à la problématique posée. La caisse est bien remplie avec une colonne correspondant à un type de pièce. Les pièces sont bien évacuées lorsque la colonne correspondante est pleine. L'inconvénient de notre programme est qu'il est long à l'exécution, en effet pour chaque déplacement nous avons une pause de 3 secondes le temps que le capteur de position (`bras_fin_mouv`) soit réinitialisé. Nous avons préféré nous concentrer sur la bonne exécution de toutes les tâches du programme plutôt que sur son efficacité. Nous aurions dû travailler sur front montant plutôt que d'attendre la réinitialisation du capteur mais l'option front montant n'étant pas disponible sur le logiciel, nous n'avons pas essayé de la reproduire. Ce qui nous a posé le plus de problème a été la gestion des délais pour la mise en place des pièces. A certain moment, les mouvements étaient pris en compte dans le programme mais pas effectués dans la simulation.