

Exercices sur les fonctions logiques

La structure à mettre en place est la suivante :

Exercice 1 : OB1 -> FC1.

Exercice 2 : OB1 -> FC1 + FC2.

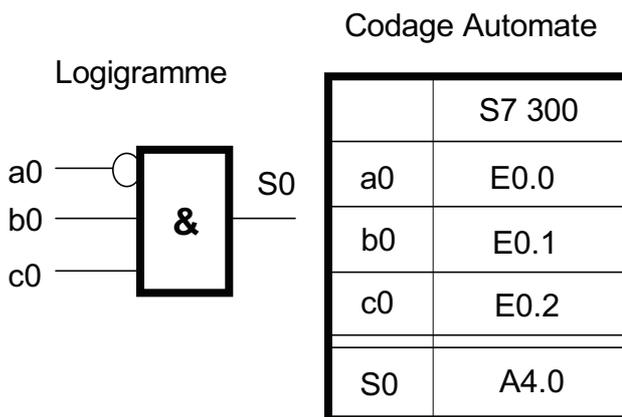
Exercice 3 : OB1 -> FC1+FC2+FC3.

Exercice 4 : OB1 -> FC1+FC2+FC3+FC4.

Exercice 5 : OB1 -> FC1+FC2+FC3+FC4+FC5.

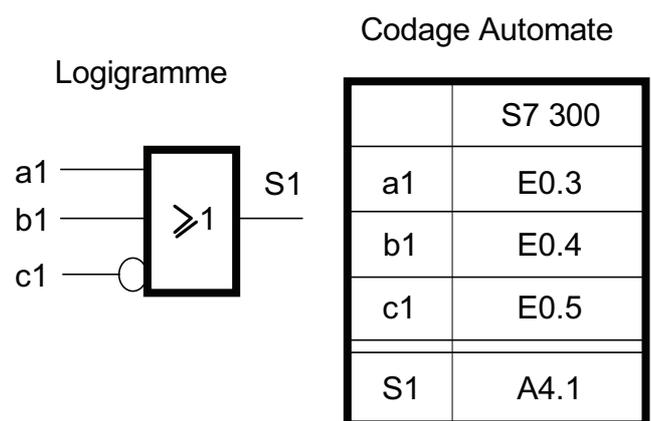
(Structure à imprimer)

Exercice N°1



- Créer le bloc FC1.
- Etablir le programme dans le bloc FC1.
- Appeler inconditionnellement le bloc FC1 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 1**.

Exercice N°2



- Créer le bloc FC2.
- Etablir le programme dans le bloc FC2.
- Ajouter l'appel inconditionnel du bloc FC2 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 2**.

Exercice N°3

Equation

$$S2 = a2 \cdot (b2 \cdot c2 \cdot d2 + e2 (f2 + g2))$$

- Créer le bloc FC3.
- Etablir le programme dans le bloc FC3.
- Ajouter l'appel inconditionnel du bloc FC3 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 3**.

Codage Automate

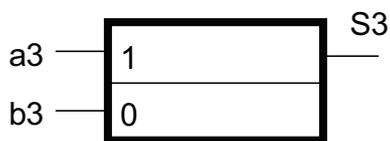
	S7 300
a2	E1.0
b2	E1.1
c2	E1.2
d2	E1.3
e2	E1.4
f2	E1.5
g2	E1.6
S2	A4.2

Exercice 4

On dispose de deux instructions pour réaliser une **fonction mémoire à mise à un prioritaire**.

- Opérateur de mise à un : **S** (Set) s'applique aux opérandes **E, A, M**
- Opérateur de mise à zéro : **R** (Reset) " "

La mise zéro prioritaire ou la mise à un prioritaire est réalisée par l'ordre d'écriture des instructions.



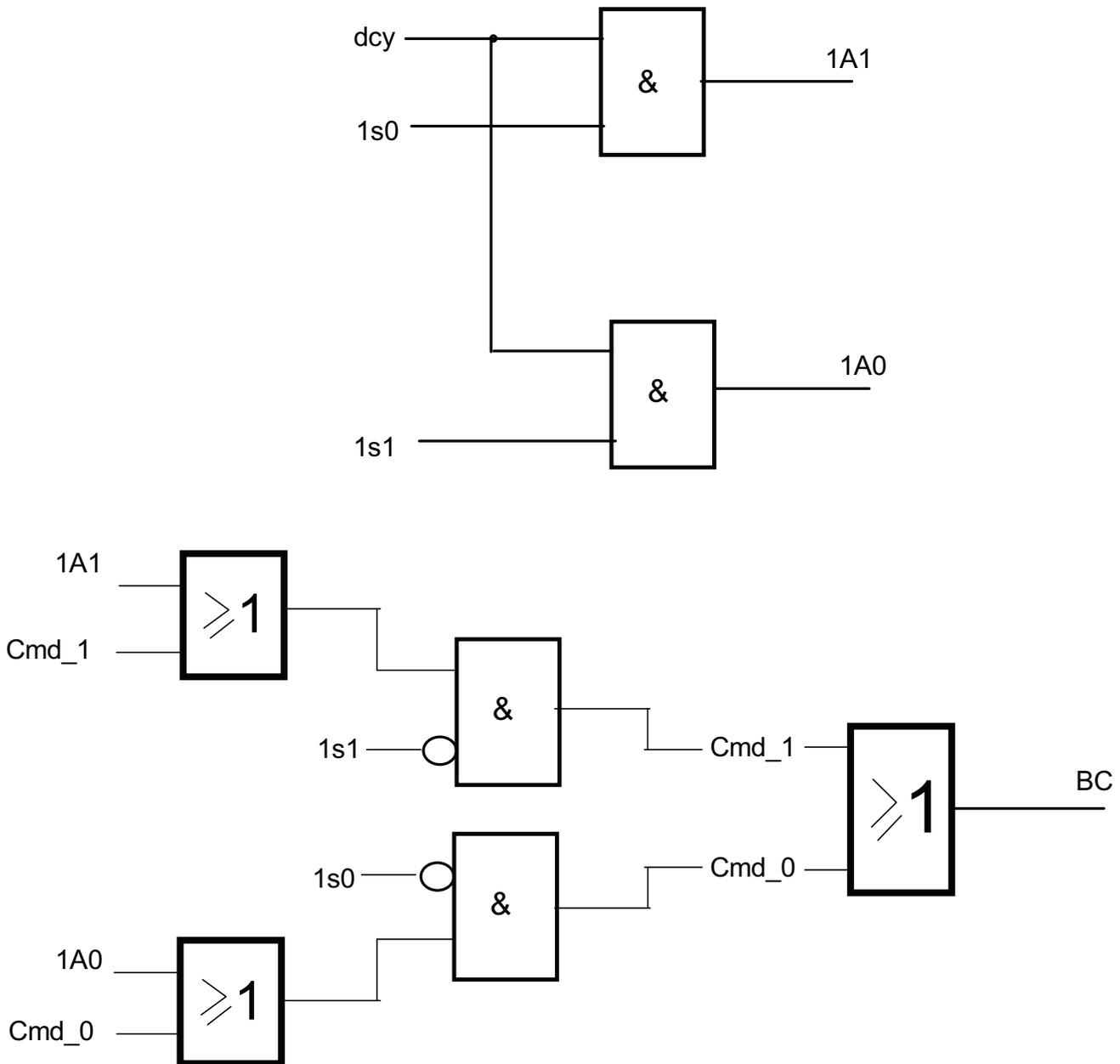
Codage Automate

	S7 300
a3	E2.0
b3	E2.1
S3	A5.0

- Créer le bloc FC4.
- Etablir le programme dans le bloc FC4.
- Ajouter l'appel inconditionnel du bloc FC4 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 4**.

Test Vérin C

Logigramme :



Remarque : Cmd_0 et Cmd_1 sont des variables internes.

- Créer le bloc FC5.
- Etablir le programme dans le bloc FC5.
- Ajouter l'appel inconditionnel du bloc FC5 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 5**.

La structure à mettre en place est la suivante :

Exercice 1 : OB1 -> FC1.

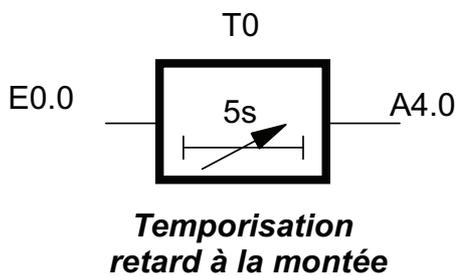
Exercice 2 : OB1 -> FC1 + FC2.

Exercice 3 : OB1 -> FC1+FC2+FC3.

Exercice 4 : OB1 -> FC1+FC2+FC3+FC4.

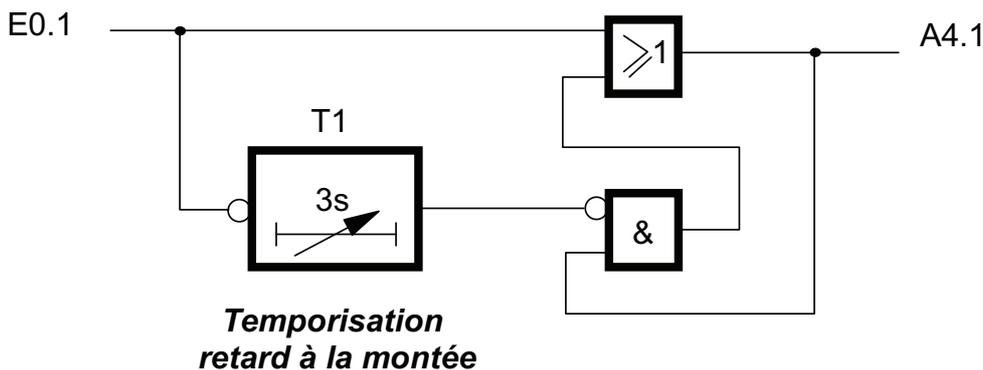
Exercice 5 : OB1 -> FC1+FC2+FC3+FC4+ FC5. (**Structure à imprimer**)

Exercice 1 : Temporisation retard à la montée



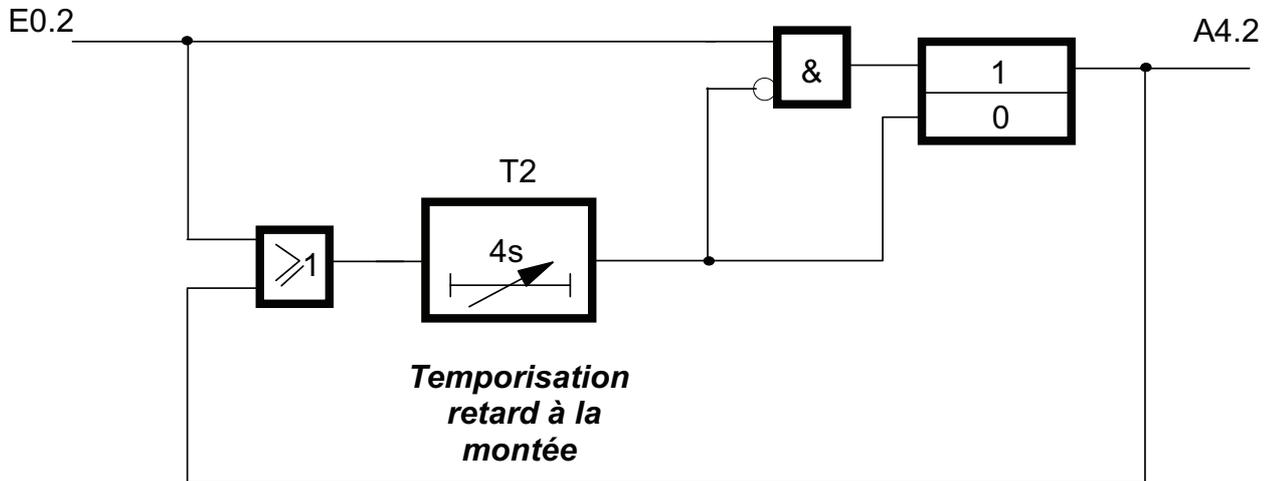
- Créer le bloc FC1.
- Etablir le programme dans le bloc FC1.
- Appeler inconditionnellement le bloc FC1 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 1**.

Exercice 2 : Maintien à l'état 1 d'une sortie pendant trois secondes après disparition de la condition d'entrée.

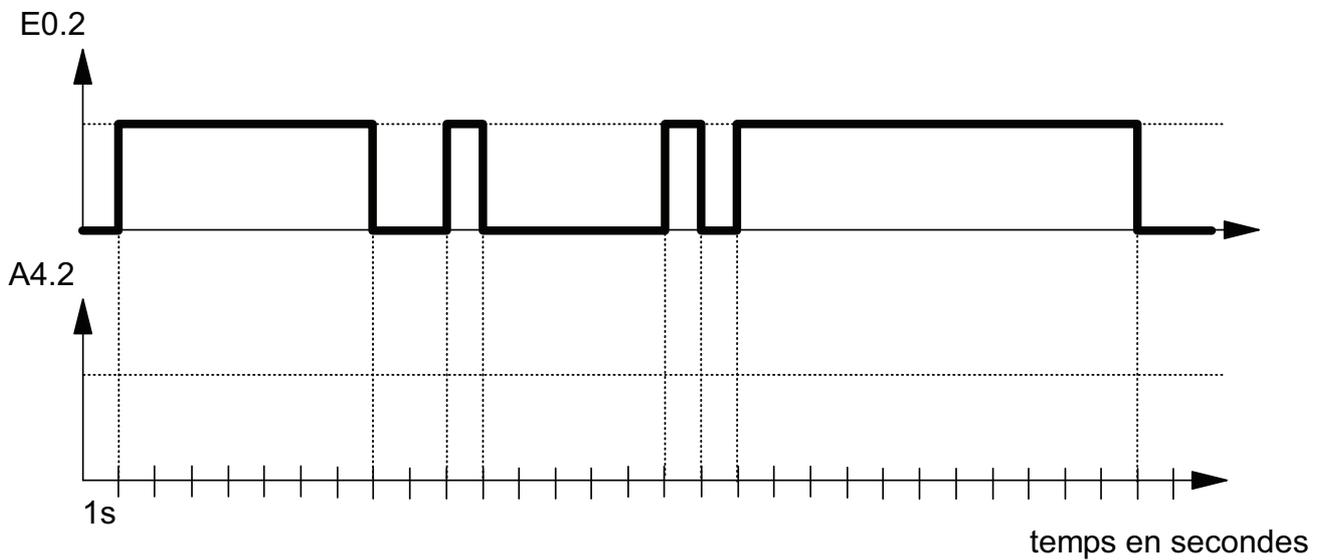


- Créer le bloc FC2.
- Etablir le programme dans le bloc FC2.
- Ajouter l'appel inconditionnel du bloc FC2 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 2**.

Exercice 3 : Ce montage génère une sortie de durée déterminée, quelle que soit la forme ou la durée de l'impulsion d'entrée.

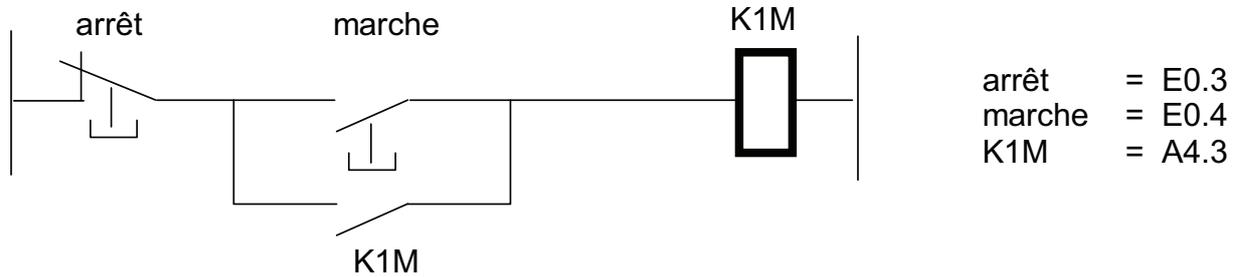


Compléter le chronogramme

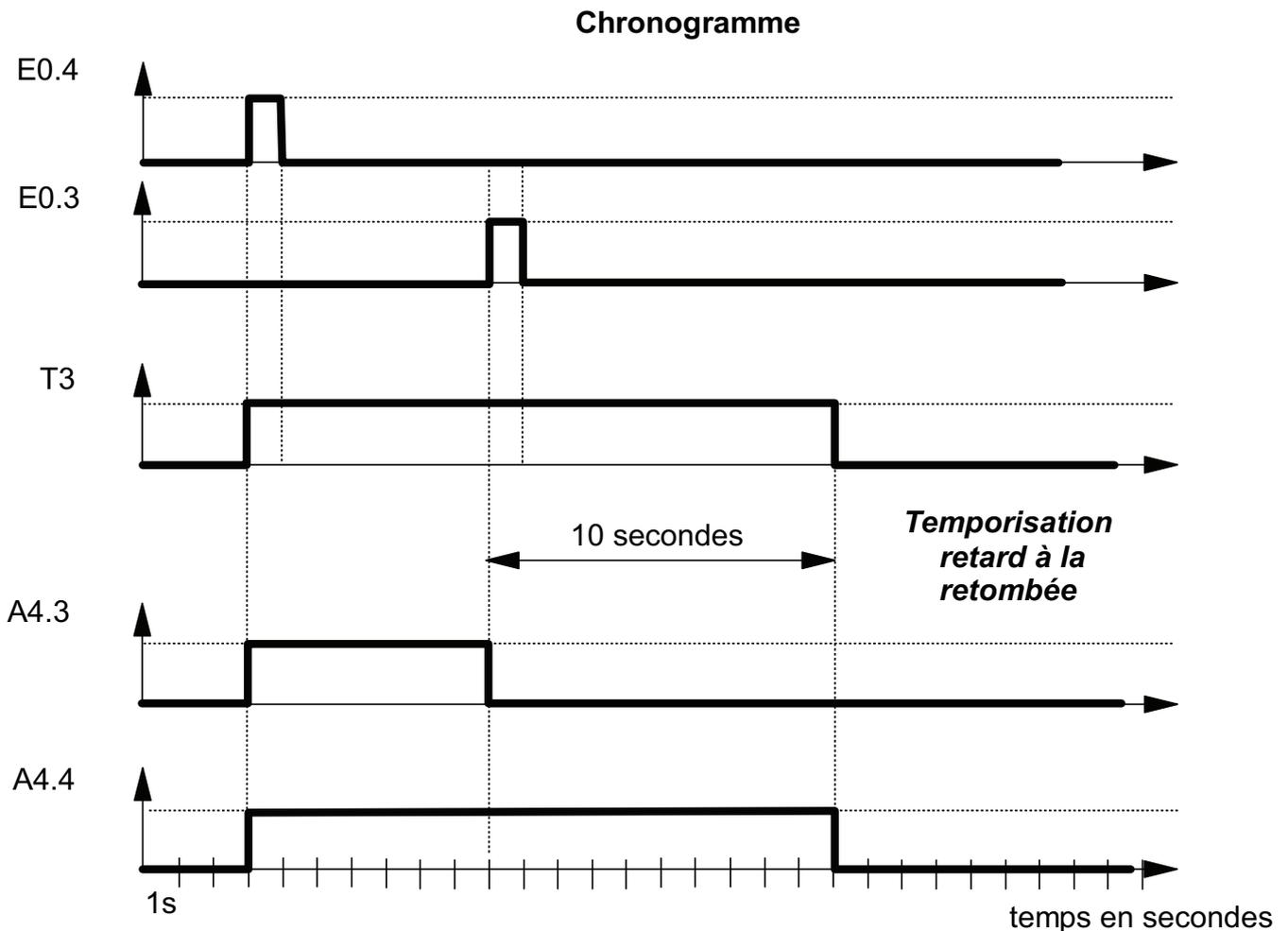


- Créer le bloc le bloc FC3.
- Etablir le programme dans le bloc FC3.
- Ajouter l'appel inconditionnel du bloc FC3 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 3**.

Exercice 4 : 1) Réaliser la fonction Marche - ARRET d'un moteur.



2) Sachant que le moteur est muni d'un voyant de fonctionnement V (A4.4) qui ne s'éteint que 10s après l'arrêt du moteur, réaliser cette fonction en utilisant un temporisateur.

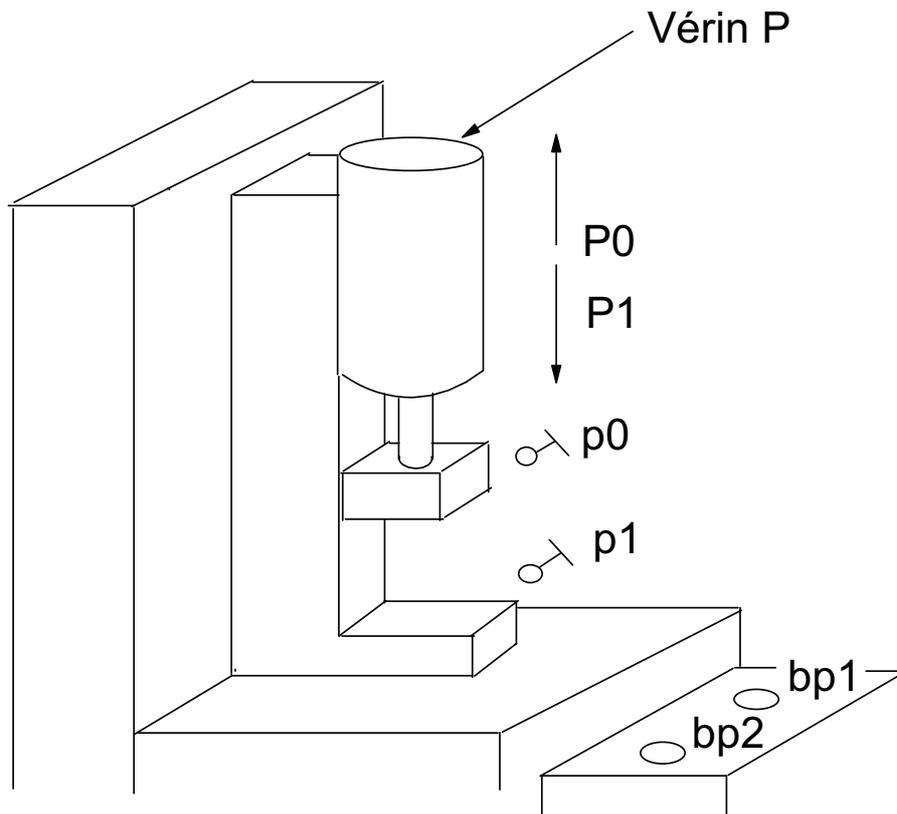


- Créer le bloc FC4.
- Etablir le programme dans le bloc FC4.
- Ajouter l'appel inconditionnel du bloc FC4 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 4**.

Exercice 5

Etude d'une commande bimanuelle

Pour actionner la descente d'une presse, l'opérateur devra appuyer sur deux boutons poussoirs dans un intervalle de temps inférieur à 0,3 secondes.



Remarque :

La presse reste 3 secondes en position basse avant de remonter.

Adressage des variables E/S

Désignation	Nom de la variable	Codage Automate
Départ cycle 1	bp1	E0.5
Départ cycle 2	bp2	E0.6
Presse haute	p0	E0.7
Presse basse	p1	E1.0
Descendre presse	P_1	A4.5
Remontée presse	P_0	A4.6

Exercice 5 (suite)

1) Equations de fonctionnement de la presse

- a) Mémoriser les actions de l'opérateur sur les boutons poussoirs bp1 et bp2

Une mémoire est mise à un lorsque l'opérateur appuie sur bp1 ou bp2 lorsque la presse est en position haute.

$$\text{Mise à un de la mémoire : Memo} = (bp1 + bp2) \cdot p0 \text{ _ _ _ _}$$

La mémoire est mise à zéro lorsque les deux bouton sont relâchés

$$\text{Mise à zéro de la mémoire : Memo} = \overline{bp1} \cdot \overline{bp2} \text{ _ _ _ _}$$

- b) L'état un de la mémoire démarre le temps de référence (0,3 s) à l'aide d'un temporisateur mode TP (impulsion)

$$\text{Tempo}_0 = \text{Memo} \text{ _ _ _ _}$$

- c) La presse ne doit descendre que si les boutons poussoirs bp1 et bp2 sont appuyés et pendant l'écoulement du temps de 0,3s .

$$P_1 = bp1 \cdot bp2 \cdot \text{Tempo}_0 \text{ _ _}$$

- d) Lancer une Temporisation de 3s lorsque la presse est en position basse.

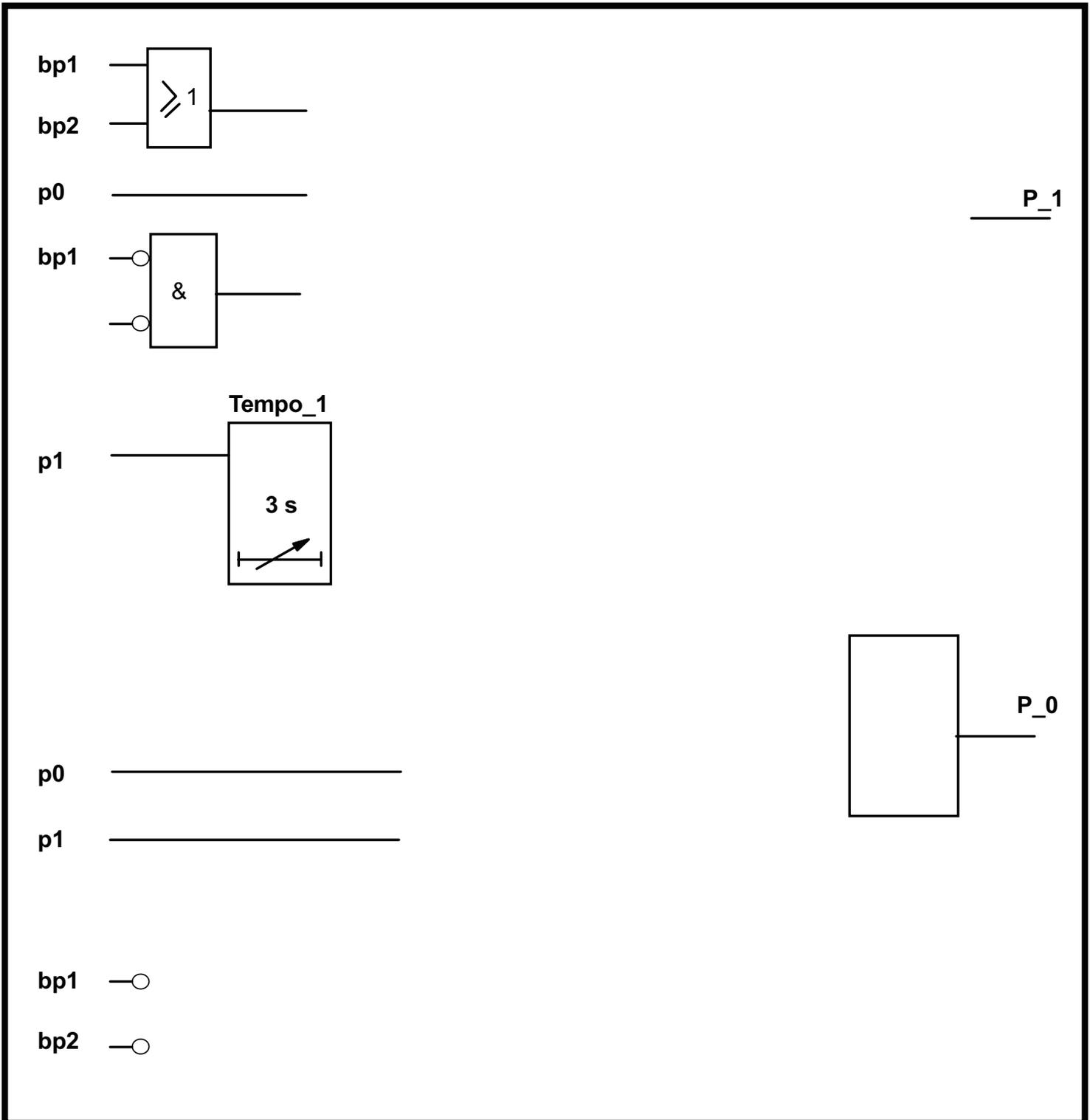
$$\text{Tempo}_1 = p1 \text{ _ _ _ _}$$

- e) Commander la remontée de la presse en fin de temporisation ou si l'opérateur relâche l'un des deux boutons poussoir avant que la presse soit en position basse (sécurité de l'opérateur).

Attention ne plus commander de remontée de la presse lorsque la position haute est atteinte.

$$P_0 = \overline{\text{tempo}_1} \cdot Q + (bp1 + bp2) \cdot p1 \cdot p0$$

2) Logigramme



- Créer le bloc FC5.
- Etablir le programme dans le bloc FC5.
- Ajouter l'appel inconditionnel du bloc FC5 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 5**.

La structure à mettre en place est la suivante :

Exercice 1 : OB1 -> FC1.

Exercice 2 : OB1 -> FC1 + FC2.

Exercice 3 : OB1 -> FC1 + FC2 + FC3. (**Structure à imprimer**)

Exercice 1 : Programmation d'un compteur Z0

Un compteur Z0 est initialisé à la valeur zéro par l'entrée API E0.0.

Il est incrémenté par l'entrée API E0.1.

La valeur courante du compteur est transférée en permanence dans le mot MW0.

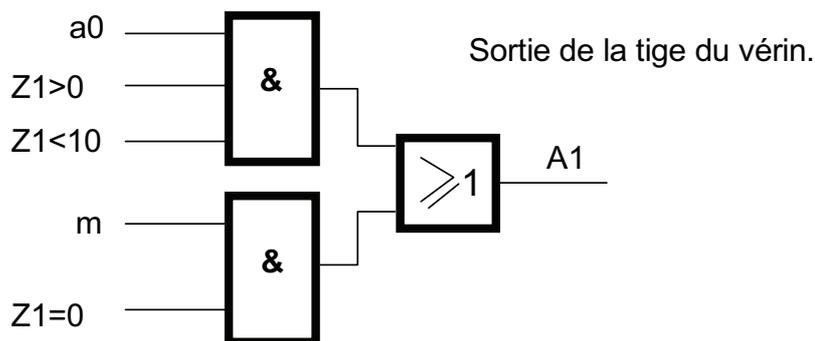
Lorsque la valeur courante du compteur est égale à trois, on commande la sortie A4.0.

- Créer le bloc FC1.
- Etablir le programme dans le bloc FC1.
- Appeler inconditionnellement du bloc FC1 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 1**.

Exercice 2 : Poste de test

Afin d'effectuer les tests d'usure d'un guidage, il est nécessaire de réaliser 10 cycles consécutifs avant l'arrêt.

a) Une impulsion sur un bouton poussoir marche :



Equation de la sortie A1 :

b) La rentrée de tige est temporisée de 2 secondes :

Equation de T0 : $T0 = a1$

Equation de A0 : $A0 = T0$

Exercice 2 (suite)

c) Comptage des cycles

Initialisation du compteur Z1

Le compteur Z1 est initialisé à la valeur zéro si les conditions suivantes sont respectées :

- Vérin tige rentrée (a0)
- Valeur courante du compteur Z1 égale à dix.

Incrémentation du compteur Z1

Le compteur Z1 est incrémenté si la condition suivante est respectée :

- Fin de temporisation (T0)

Transférer en permanence la valeur courante du compteur Z1 dans le mot **MW2**.

Adressage des variables E/S

Point de vue P.O.	Point de vue P.C.	Point de vue réalisateur
Départ de cycle	m	E0.2
Tige rentrée	a0	E0.3
Tige sortie	a1	E0.4
Sortie tige	A1	A4.1
Rentrée tige	A0	A4.2

- Créer le bloc FC2.
- Etablir le programme dans le bloc FC2.
- Ajouter l'appel inconditionnel du bloc FC2 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 2**.

Exercice 3 : Marche de vérification

On désire réaliser une marche de vérification dans l'ordre de trois actionneurs définie par le tableau suivant :

Compteur Z2 = 0	A0	B0	C0
Compteur Z2 = 1	A1	B0	C0
Compteur Z2 = 2	A1	B1	C0
Compteur Z2 = 3	A1	B1	C1

+

-

Incrémentation compteur = $a \cdot b$

Décrémentation compteur = $a \cdot \overline{b}$

Il est possible à tous moments d'effectuer une remise à zéro du compteur Z2.

Adressage des variables E/S

Point de vue P.C.	Point de vue réalisateur	Point de vue P.C.	Point de vue réalisateur
a	E1.0	B0	A5.2
b	E1.1	B1	A5.3
Raz	E1.2	C0	A5.4
A0	A5.0	C1	A5.5
A1	A5.1		

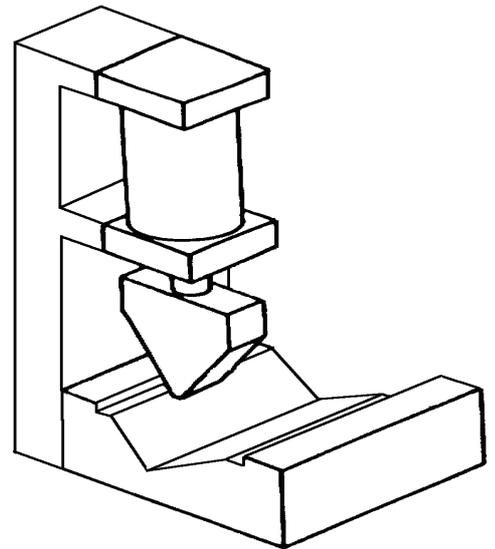
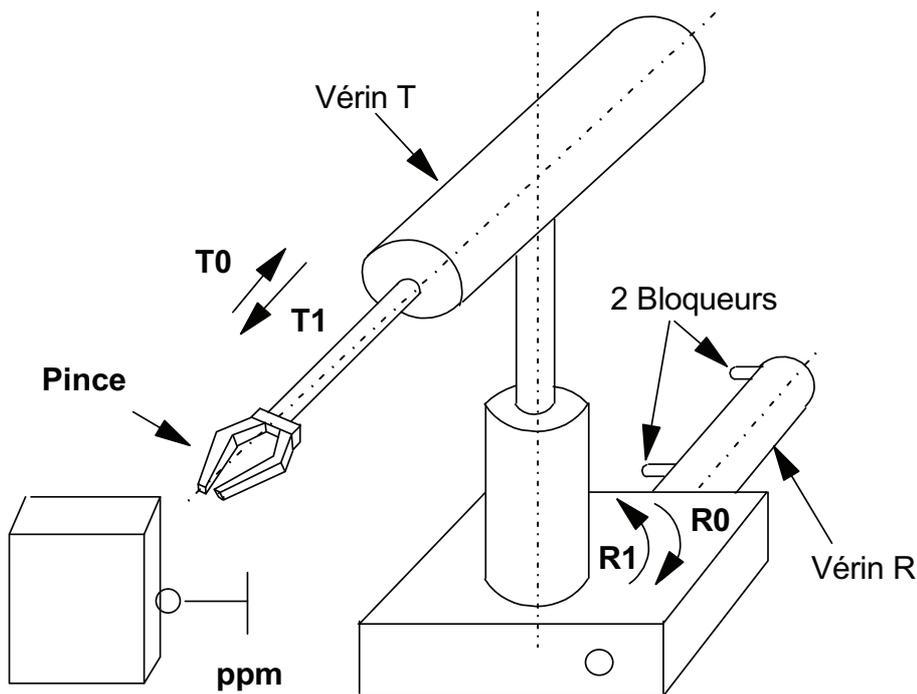
Utiliser le mot **MW4** pour la recopie de la valeur courante du compteur.

- Créer le bloc FC3.
- Etablir le programme dans le bloc FC3.
- Ajouter l'appel inconditionnel du bloc FC3 dans le bloc OB1.
- Tester le fonctionnement dans la table d'animation **Exercice 3**.

1) Description

Un poste de pliage est composé des éléments suivants :

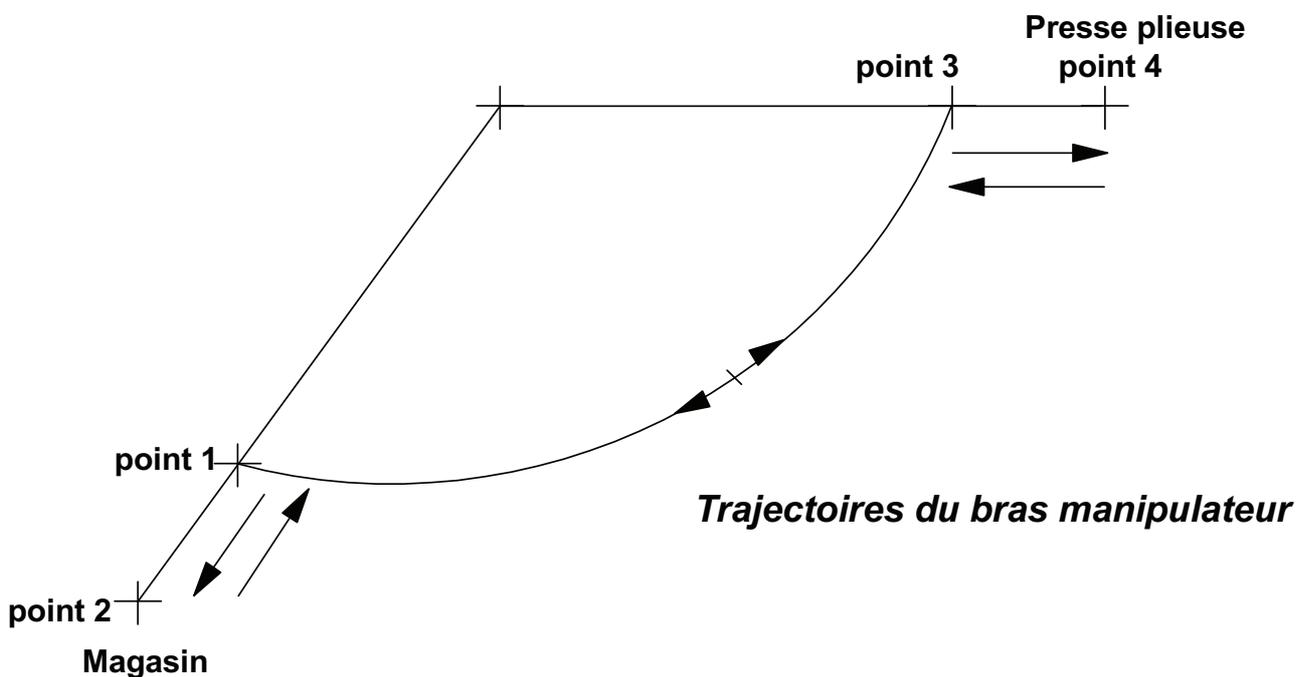
- Un magasin d'alimentation en pièces brutes.
- Un bras manipulateur à deux mouvements (une translation et une rotation), muni d'une pince.
- Une presse plieuse.



Presse plieuse

Magasin

Bras manipulateur



Trajectoires du bras manipulateur

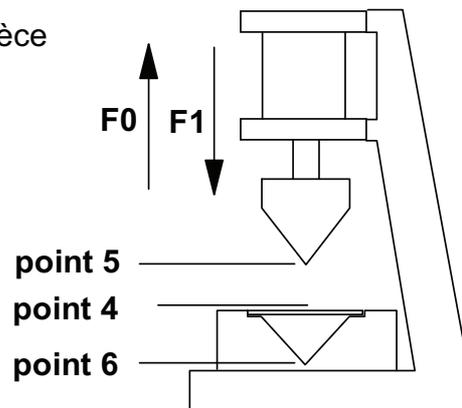
2) Analyse des tâches opératives

Tâche T1 : Pliage

Effectuer un pliage à 90° sur la pièce

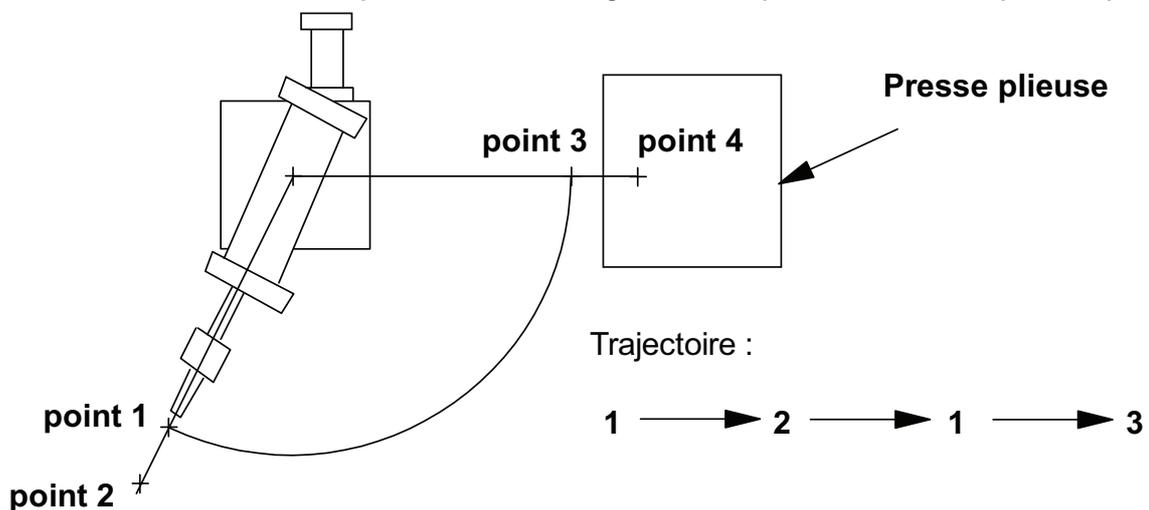
Trajectoire :

5 → 6 → 5



Tâche T2 : Approvisionnement

Prendre une pièce dans le magasin et la placer devant la presse plieuse

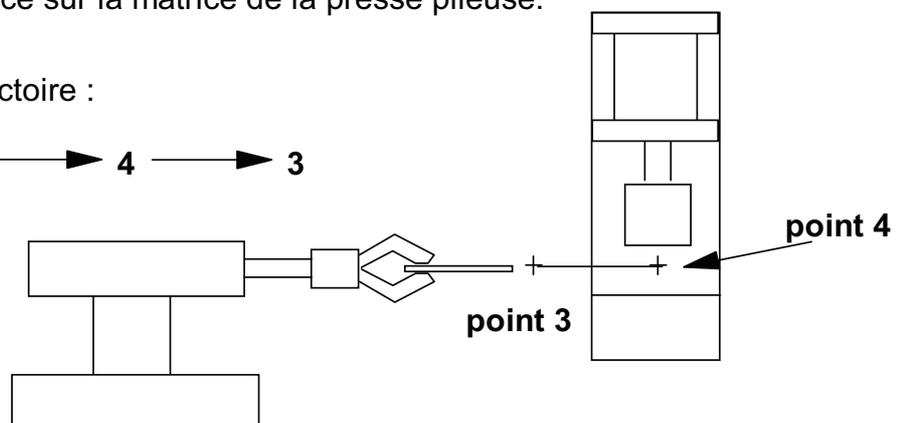


Tâche T3 : Dépose

Placer une pièce sur la matrice de la presse plieuse.

Trajectoire :

3 → 4 → 3



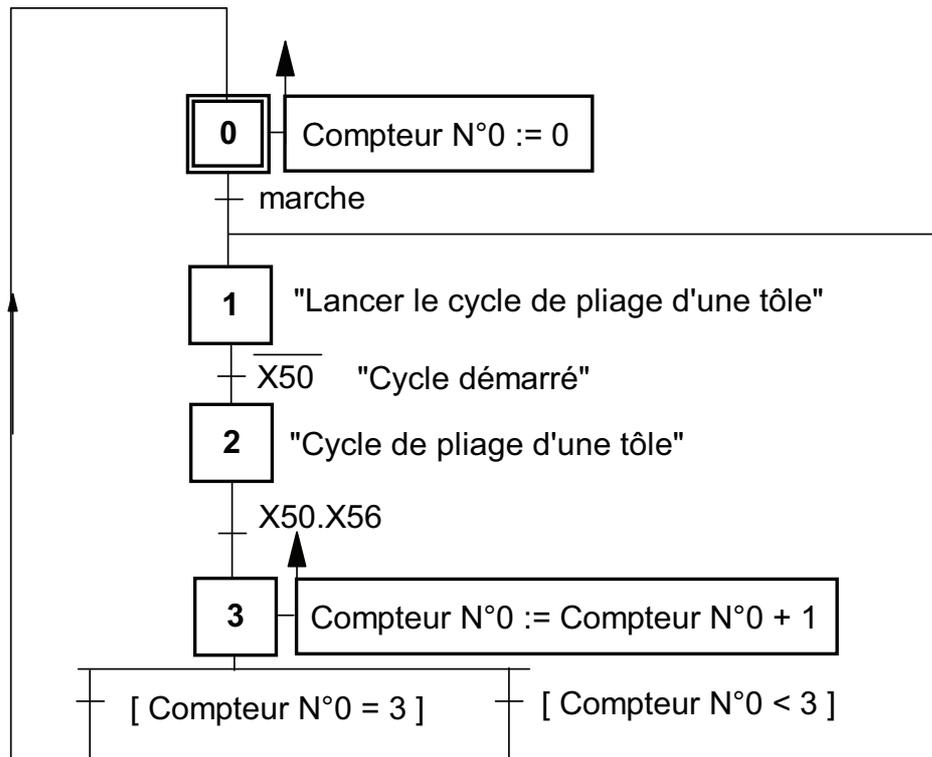
Tâche T4 : Retour du bras manipulateur

Retour du bras manipulateur en face du magasin.

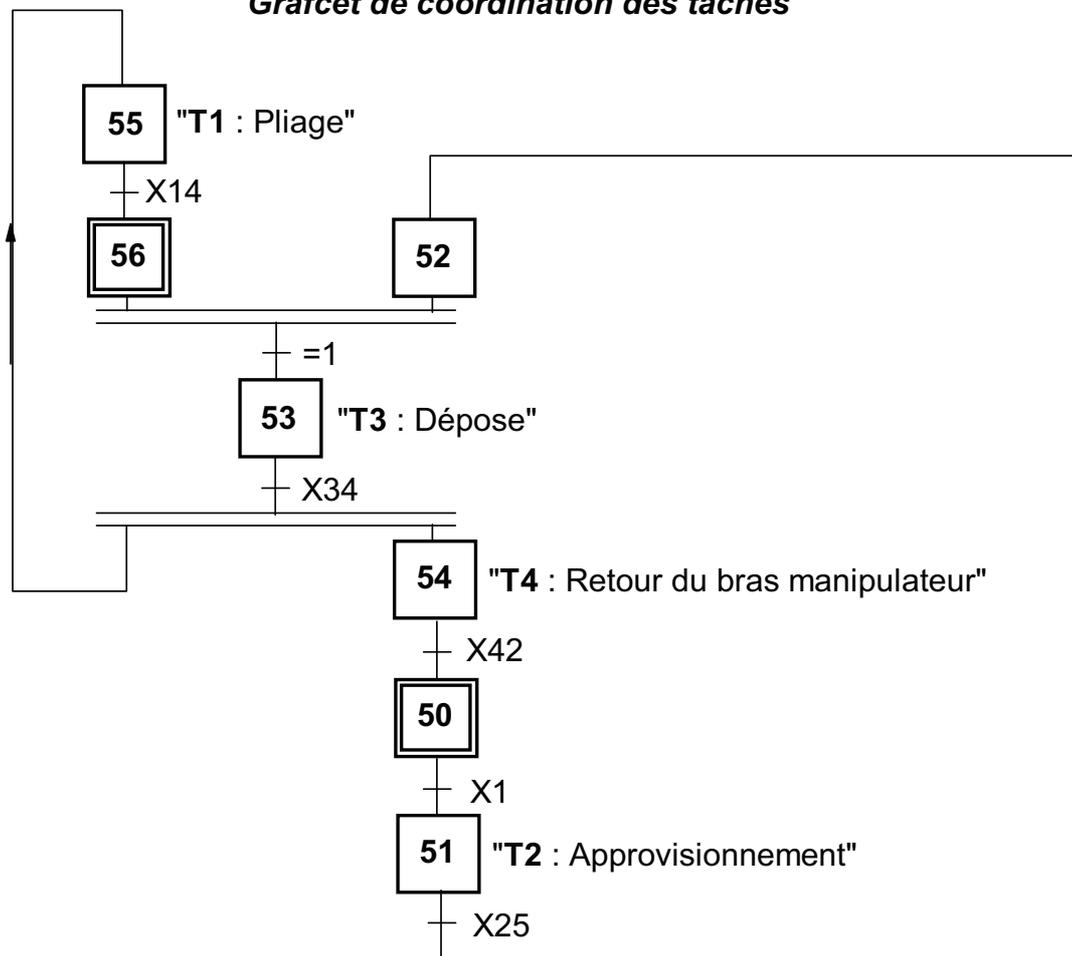
Trajectoire : 3 → 1

3) Graficets point de vue PC

Graficet de conduite

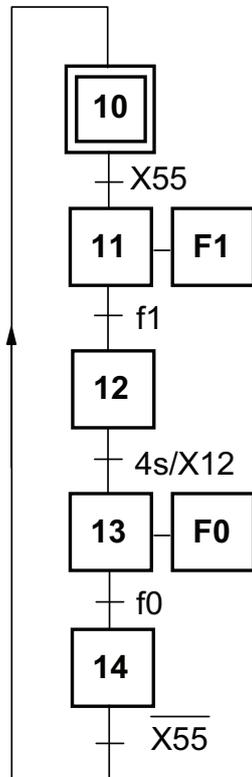


Graficet de coordination des tâches

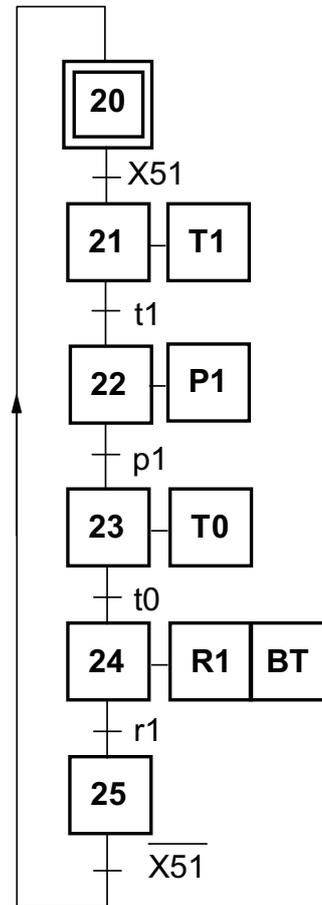


3) Grafquets point de vue PC (Suite)

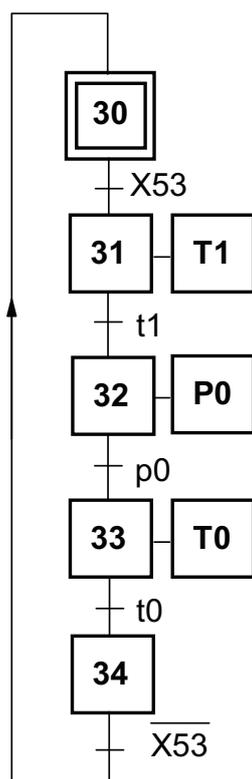
Grafquet tâche T1 : Pliage



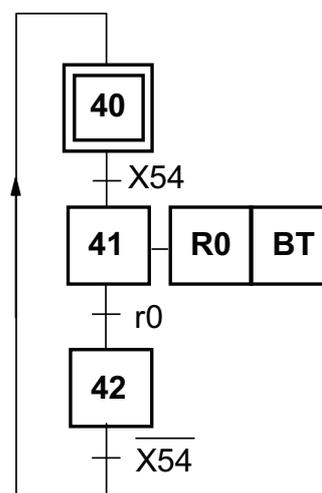
Grafquet tâche T2 : Approvisionnement



Grafquet tâche T3 : Dépose



Grafquet tâche T4 : Retour du bras manipulateur

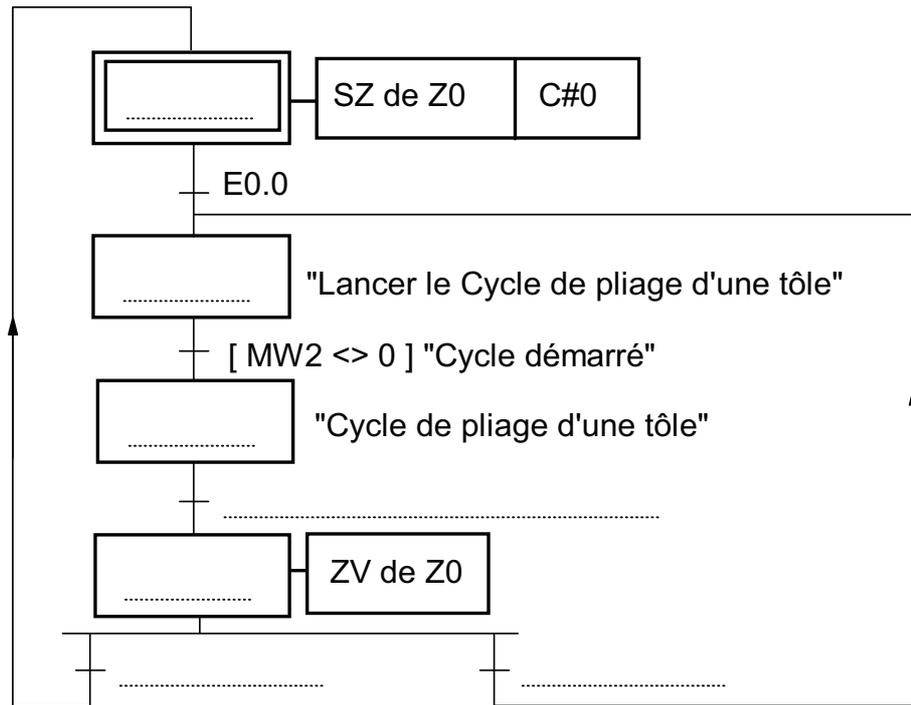


4) Graficets A.P.I.

FC1 : Graficet de conduite

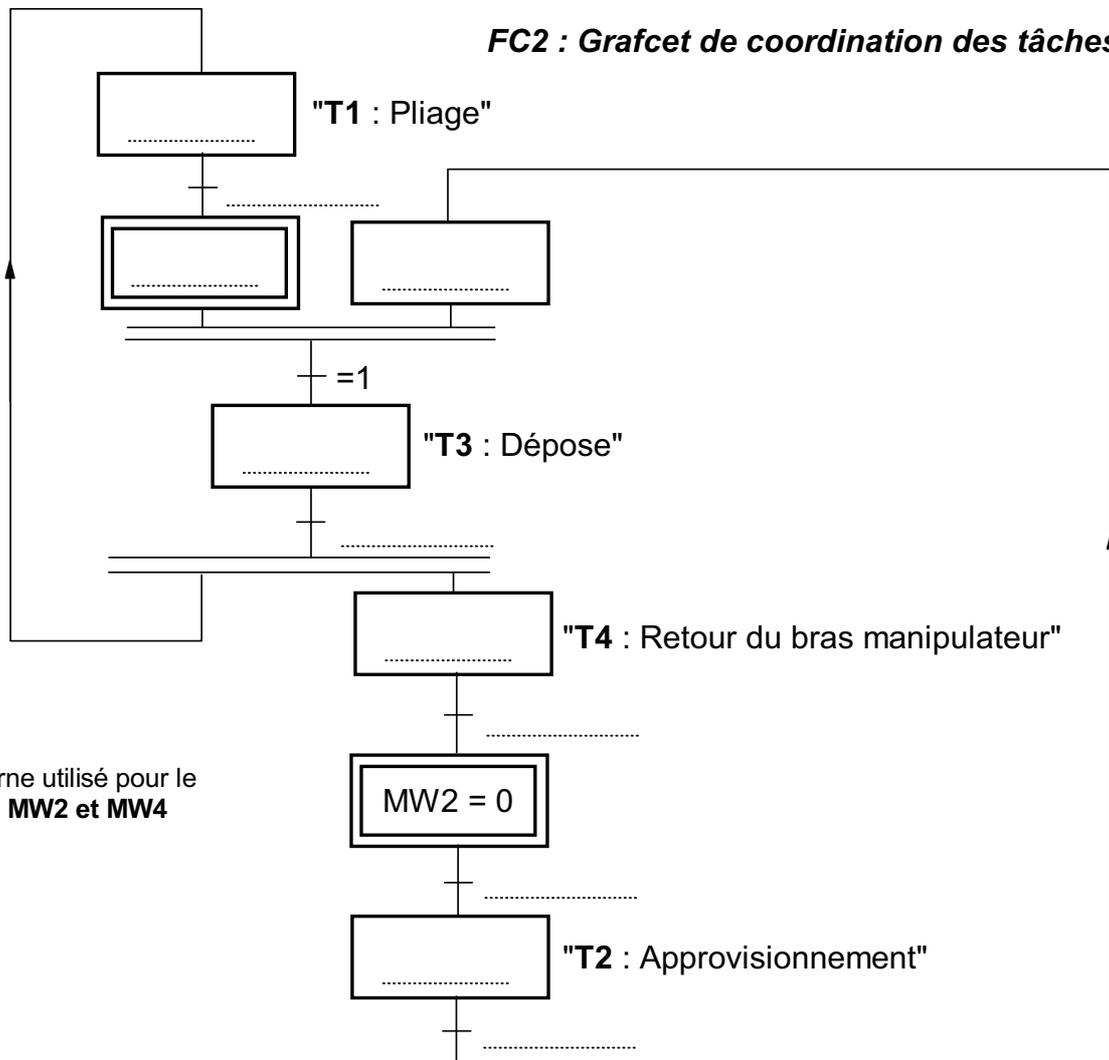
Mot interne utilisé pour le graficet : **MW0**

Recopie de la valeur courante du compteur : **MW20**



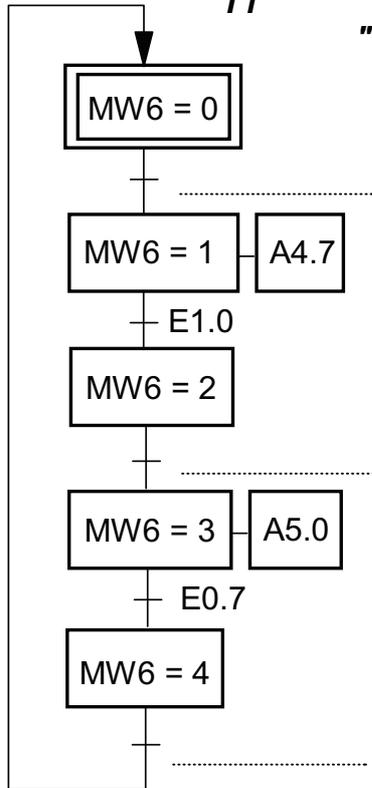
FC2 : Graficet de coordination des tâches

Mot interne utilisé pour le graficet : **MW2 et MW4**

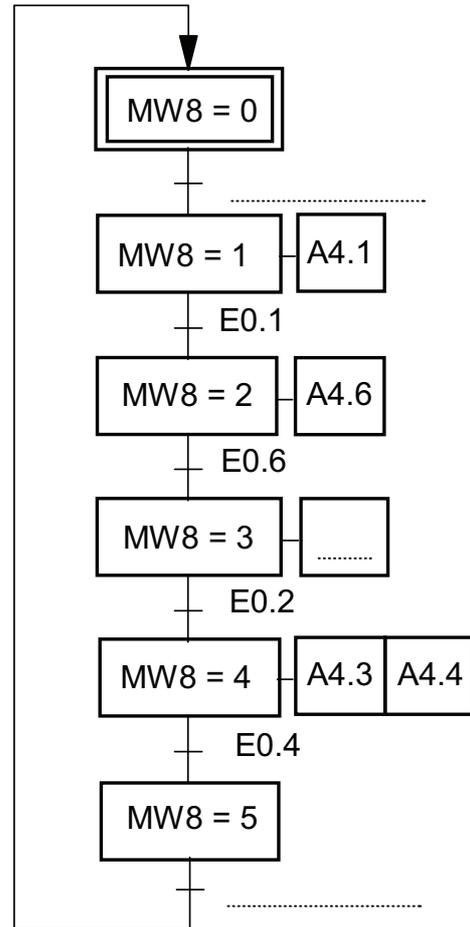


4) Graficets A.P.I. (Suite)

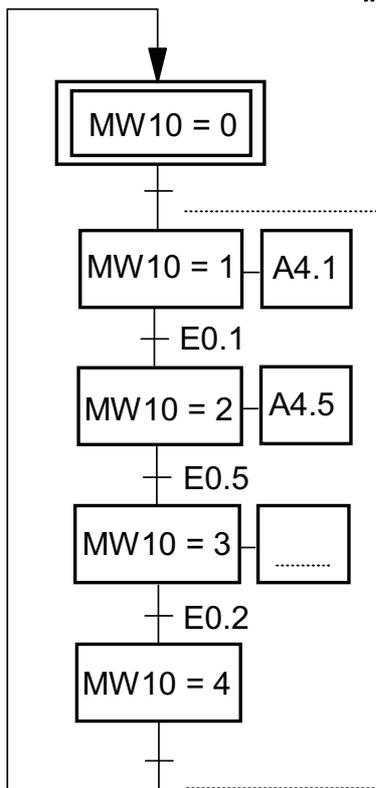
FC3 : Graficet tâche T1
" Pliage "



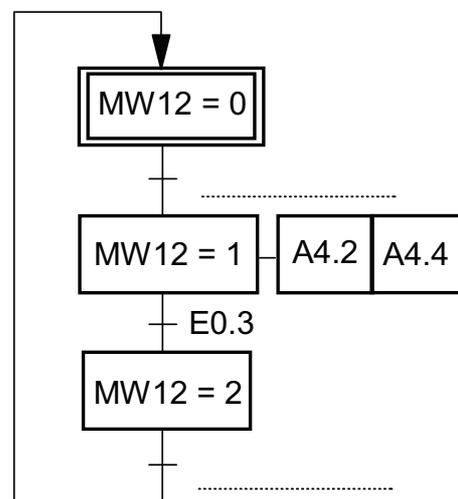
FC4 : Graficet tâche T2
" Approvisionnement "



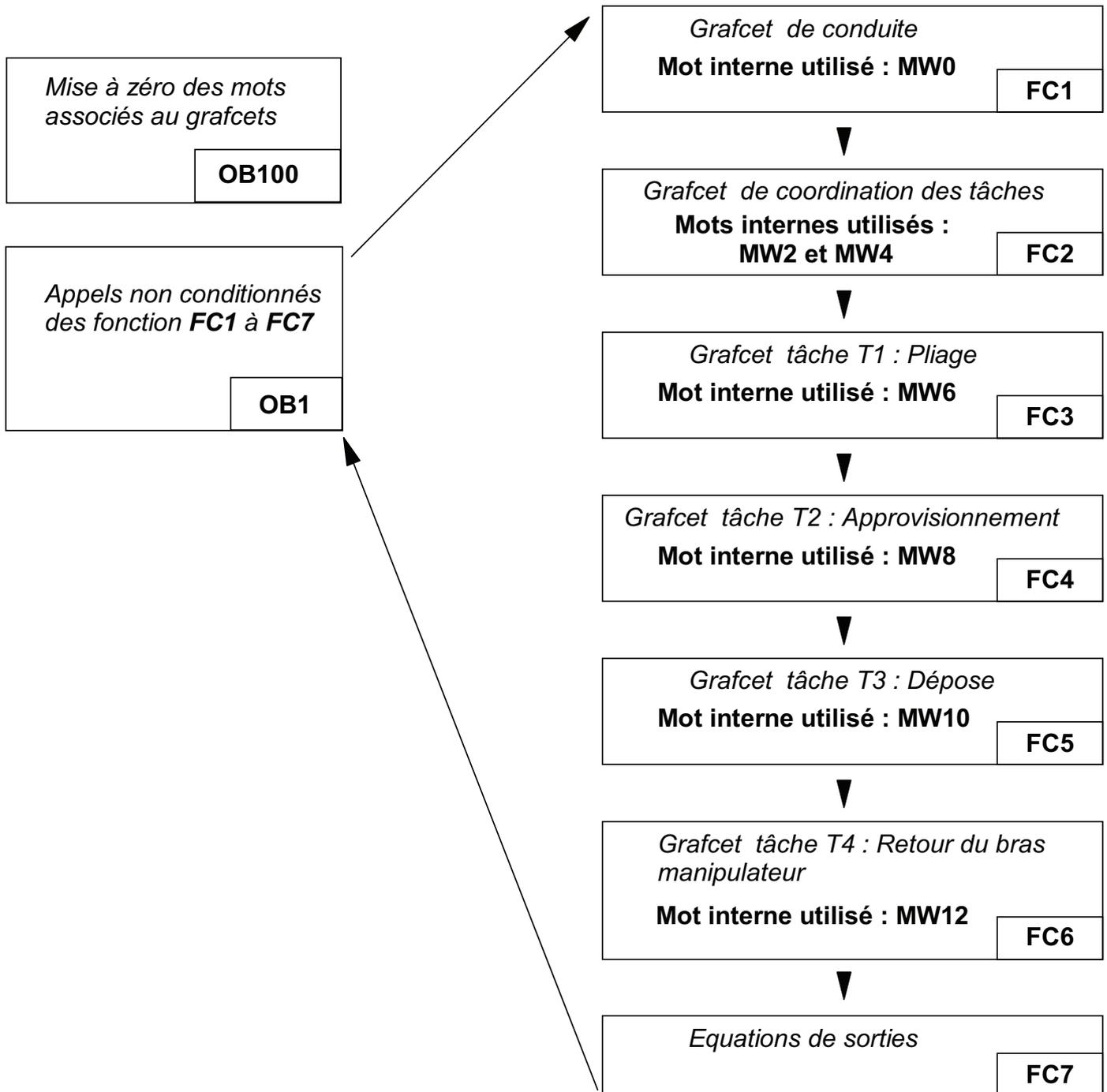
FC5 : Graficet tâche T3
" Dépose "



FC6 : Graficet tâche T4
"Retour du bras manipulateur "



5) Structure du programme



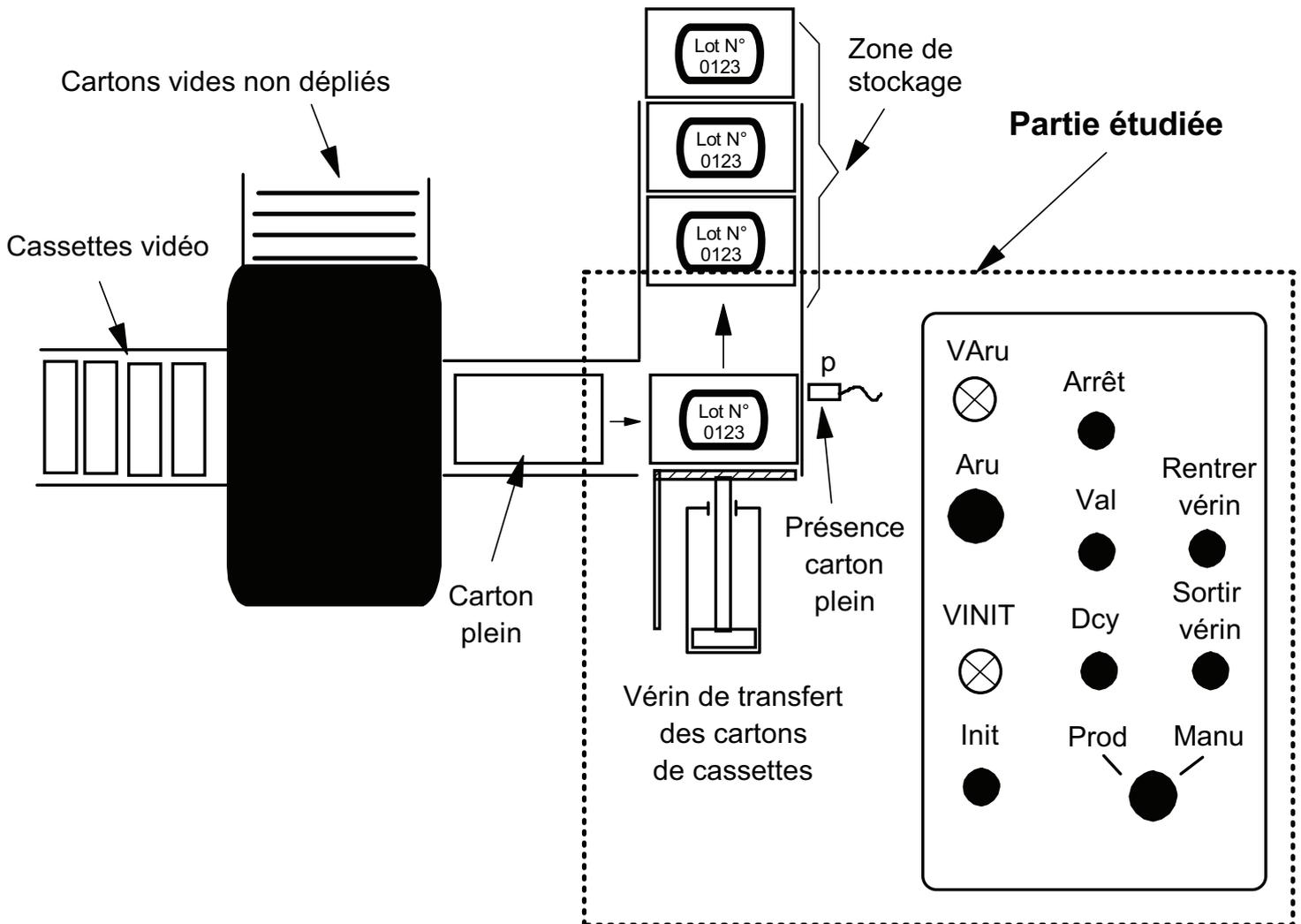
Travail demandé :

- 1) - Créer l'application TP4 dans SIMATIC Manager.
- 2) - Compléter les documents page 5/8 et page 6/8. (Aide : TD4 et application TP4)
- 3) - Créer et programmer le bloc OB100 de démarrage.
- 4) - Créer et programmer les blocs FC1 à FC6 respectant la méthode de programmation sur mot. (Aide : TD4)
- 5) - Programmer la commande des sorties dans le bloc fonction FC7.
- 6) - Créer et programmer le bloc OB1.
- 7) - Câbler, transférer le programme dans l'API et tester le fonctionnement.
- 8) - Sauvegarder puis imprimer le programme.

6) Adressage des entrées et des sorties

<i>Point de vue PO</i>	<i>Point de vue PC</i>	<i>Réalisateur</i>
mise en fonctionnement	marche	E0.0
bras avancé	t1	E0.1
bras reculé	t0	E0.2
bras devant le magasin	r0	E0.3
bras devant la presse	r1	E0.4
pince ouverte	p0	E0.5
pince fermée	p1	E0.6
presse remontée	f0	E0.7
presse descendue	f1	E1.0
Reculer le bras	T0	A4.0
Avancer le bras	T1	A4.1
Rotation vers le magasin	R0	A4.2
Rotation vers la presse	R1	A4.3
Libérer la rotation	BT	A4.4
Ouvrir la pince	P0	A4.5
Fermer la pince	P1	A4.6
Descendre la presse	F1	A4.7
Remonter la presse	F0	A5.0

1) Description



L'étude se limite au transfert des cartons pleins vers la zone de stockage

2) Cahier des charges

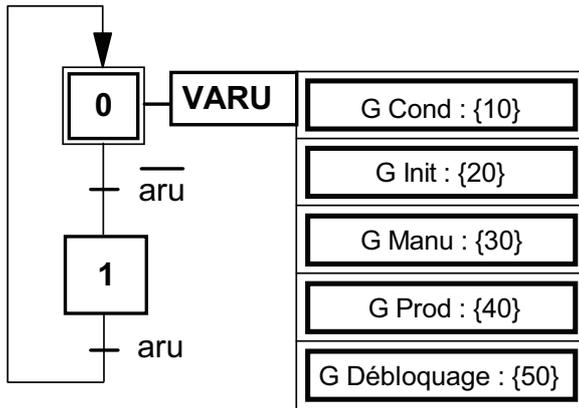
A l'arrivée d'un carton plein devant le vérin de transfert des cartons, l'opérateur colle manuellement une étiquette d'identification, puis appuie sur le bouton **Val** afin de transférer le carton dans la zone de stockage.

Modes de marche :

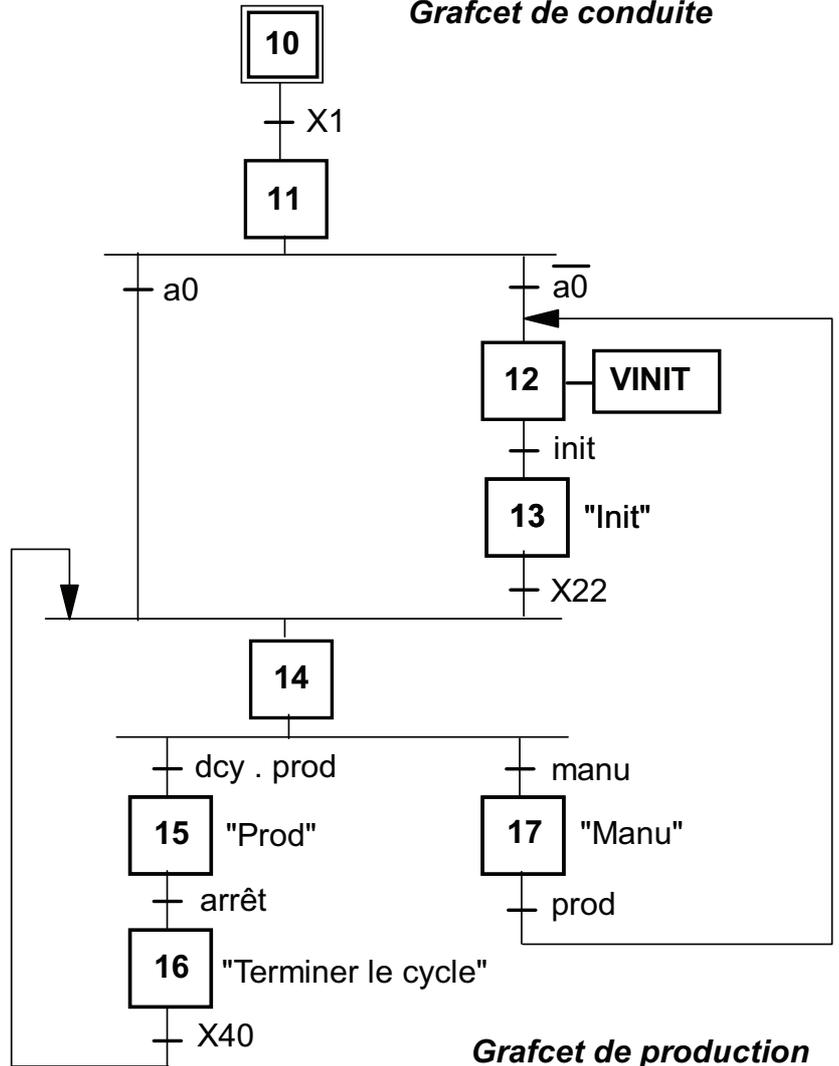
- Production normale. (**Départ cycle** et **Arrêt normal**)
- Initialisation si les conditions initiales ne sont pas présentes lors de la mise sous tension ou après acquittement de l'arrêt d'urgence.
- Commandes manuelles au pupitre pour réglages.
- **Arrêt d'urgence** : Arrêt en position de la partie opérative.

3) Grafsets point de vue PC

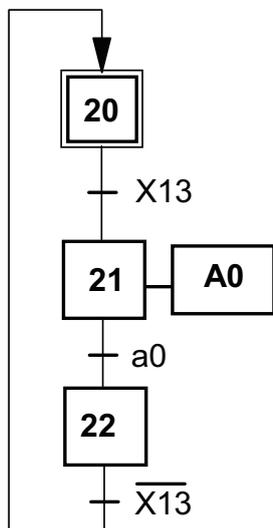
Grafset de sûreté



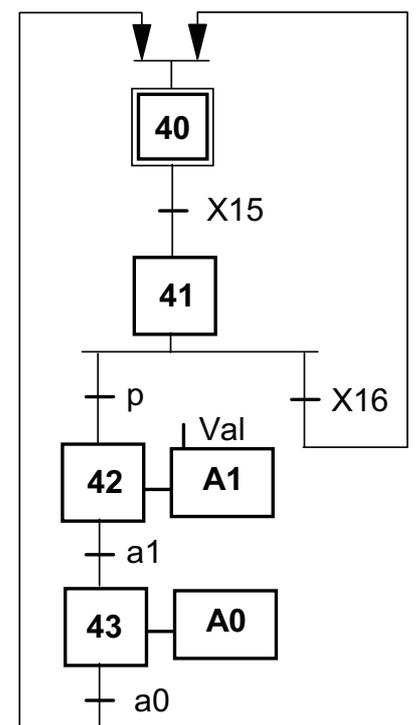
Grafset de conduite



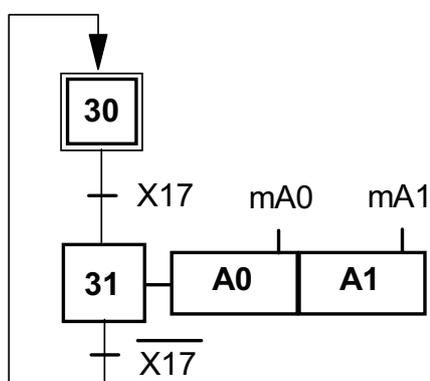
Grafset d'initialisation



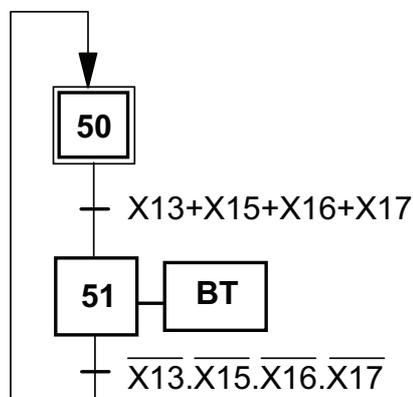
Grafset de production



Grafset de mode manuel

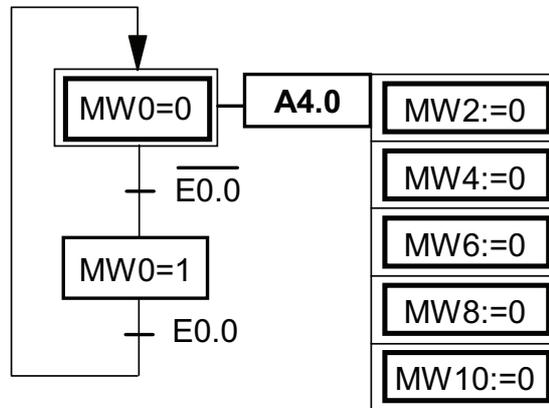


Grafset déblocage transfert

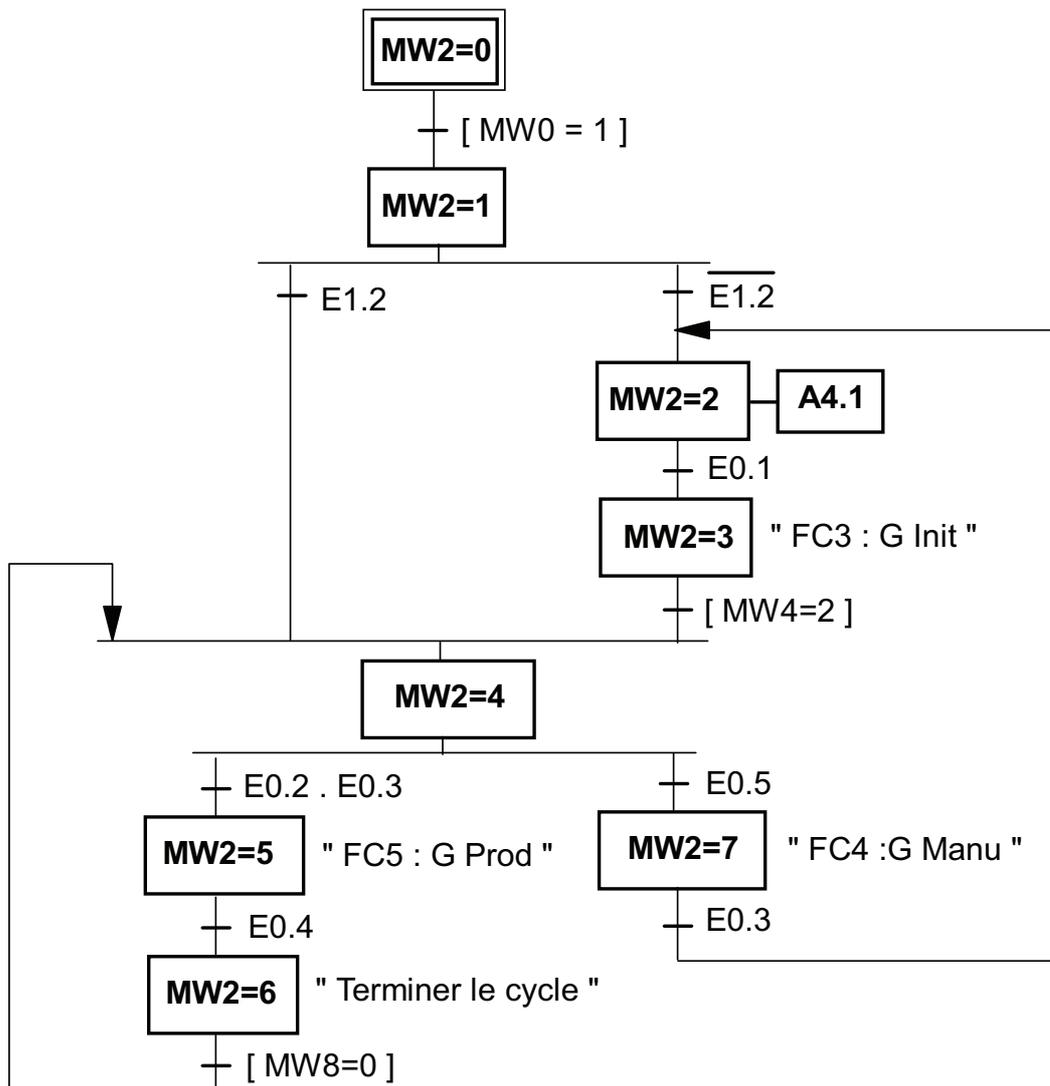


4) Grafquets A.P.I.

FC1 : Grafcet de sûreté

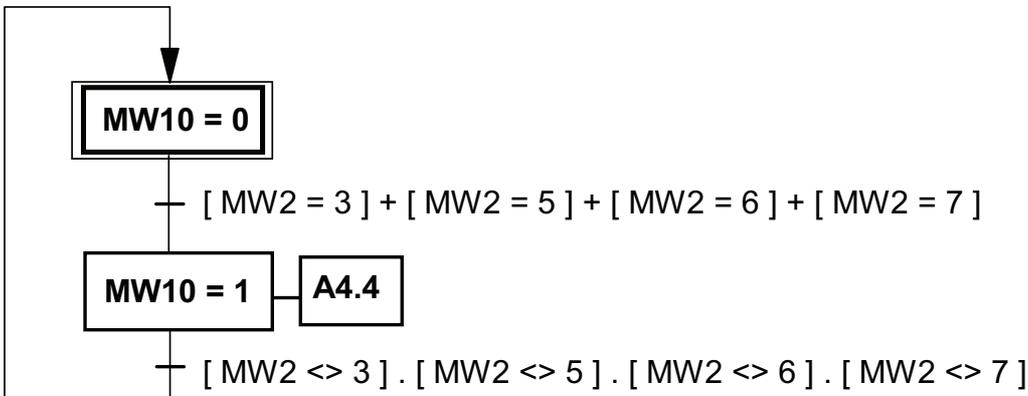


FC2 : Grafcet de conduite

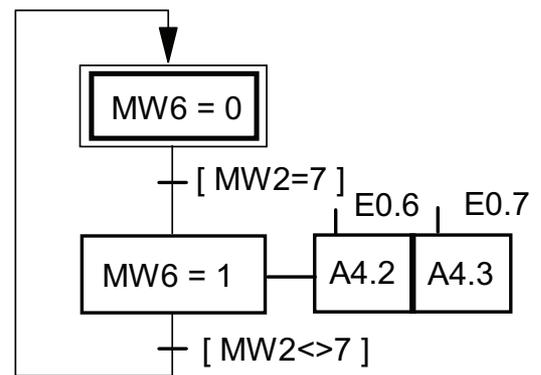


4) Graficets A.P.I. (Suite)

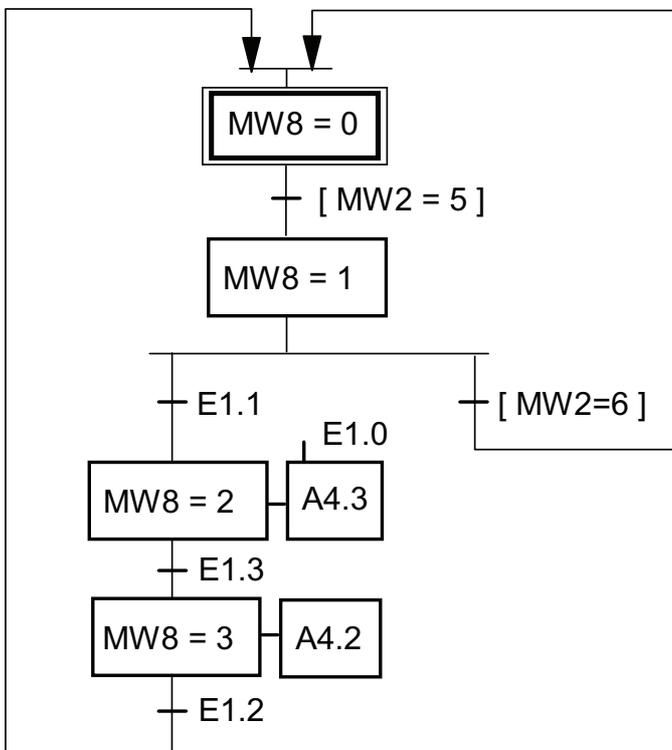
FC6 : Graficet déblocage transfert



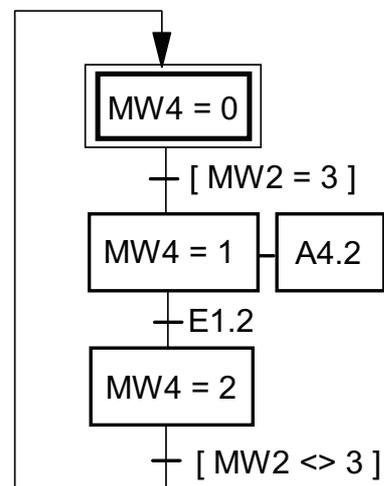
FC4 : Graficet de mode manuel



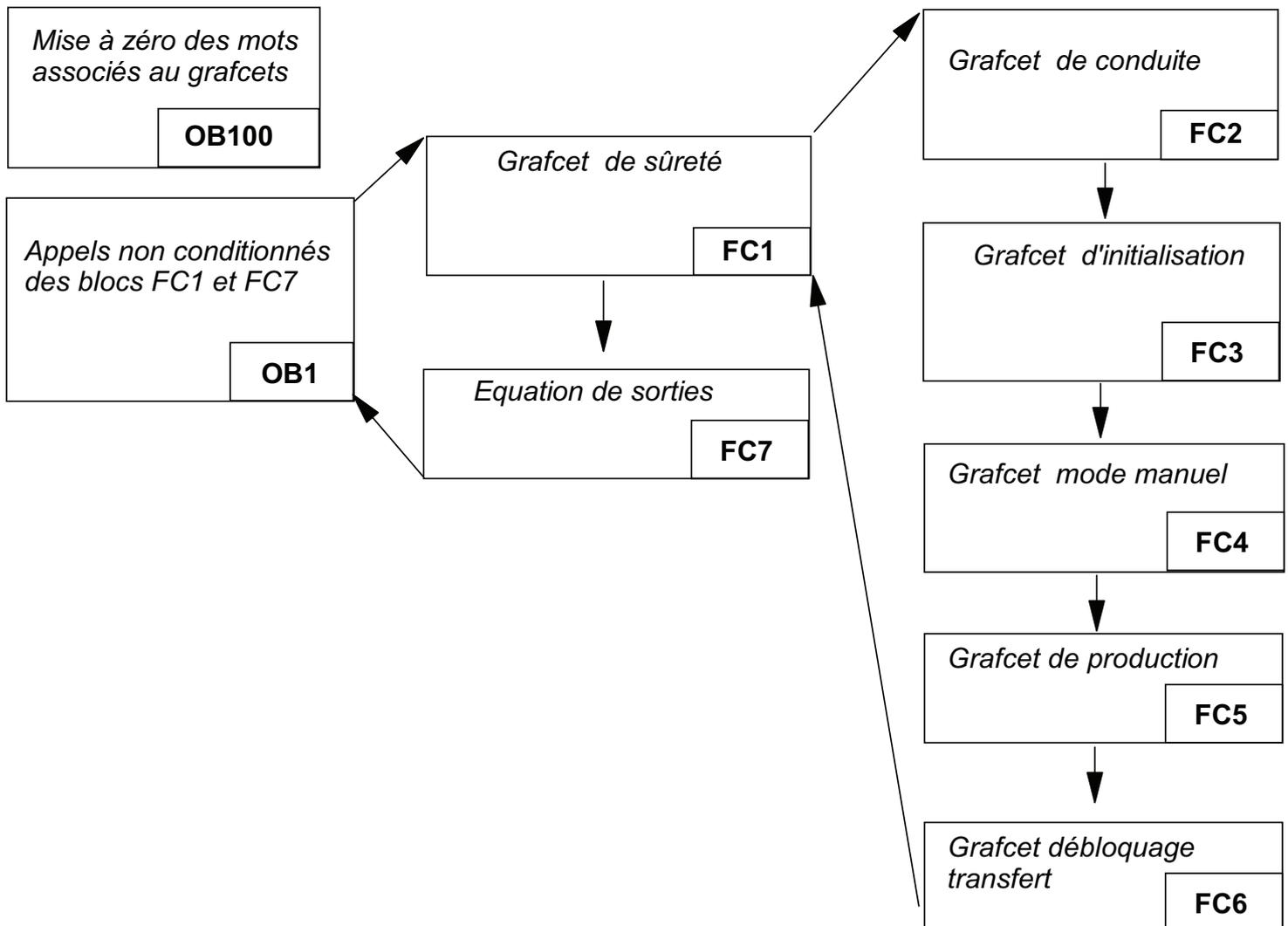
FC5 : Graficet de production"



FC3 : Graficet d'initialisation



5) Structure du programme



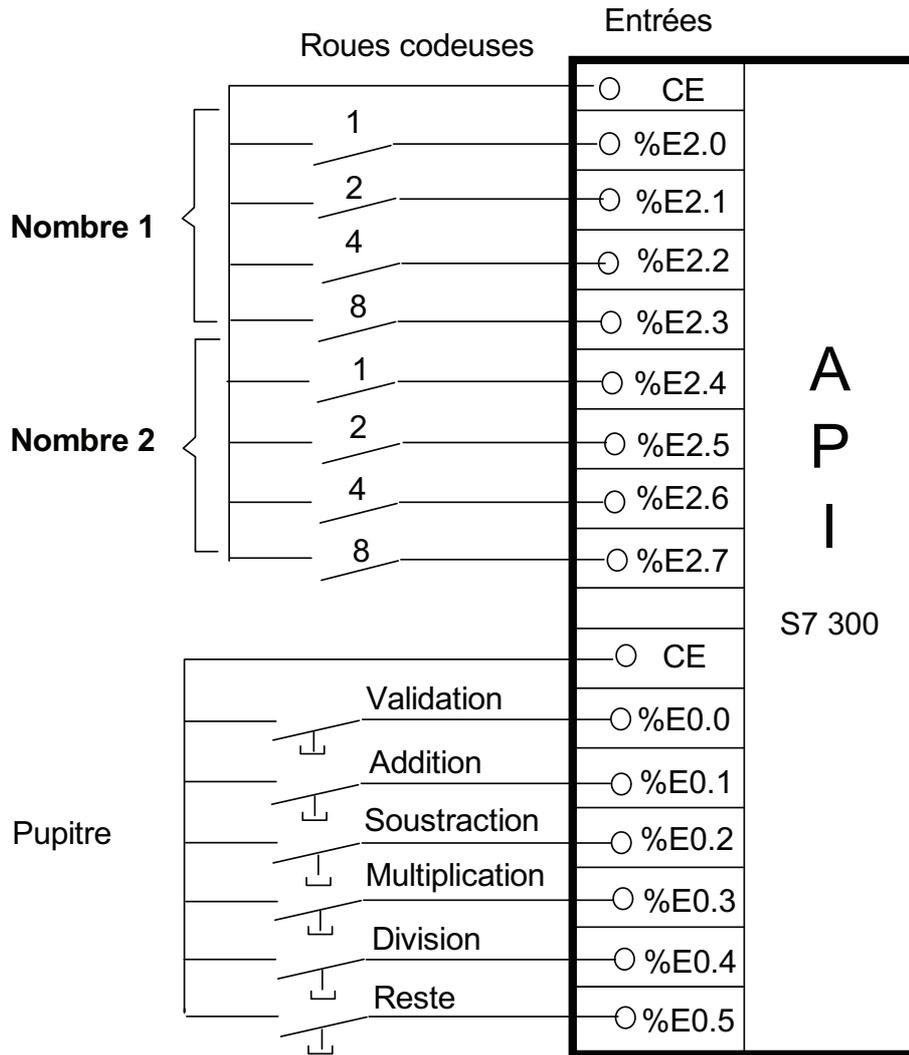
Travail demandé :

- 1) Ouvrir l'application TP5 dans SIMATIC Manager.
- 2) Créer et programmer les blocs **FC2 à FC6**.
- 3) Créer et programmer le bloc **FC1** en respectant la méthode de programmation sur mot.
(Aide : **TD4** et **TD5**)
- 4) Créer et programmer le bloc **OB1**.
- 5) Programmer la commande des sorties dans le bloc fonction **FC7**.
- 6) Créer et programmer le bloc **OB100**.
- 7) Câbler, transférer le programme dans l'API et tester le fonctionnement.
- 8) Sauvegarder puis imprimer le programme.

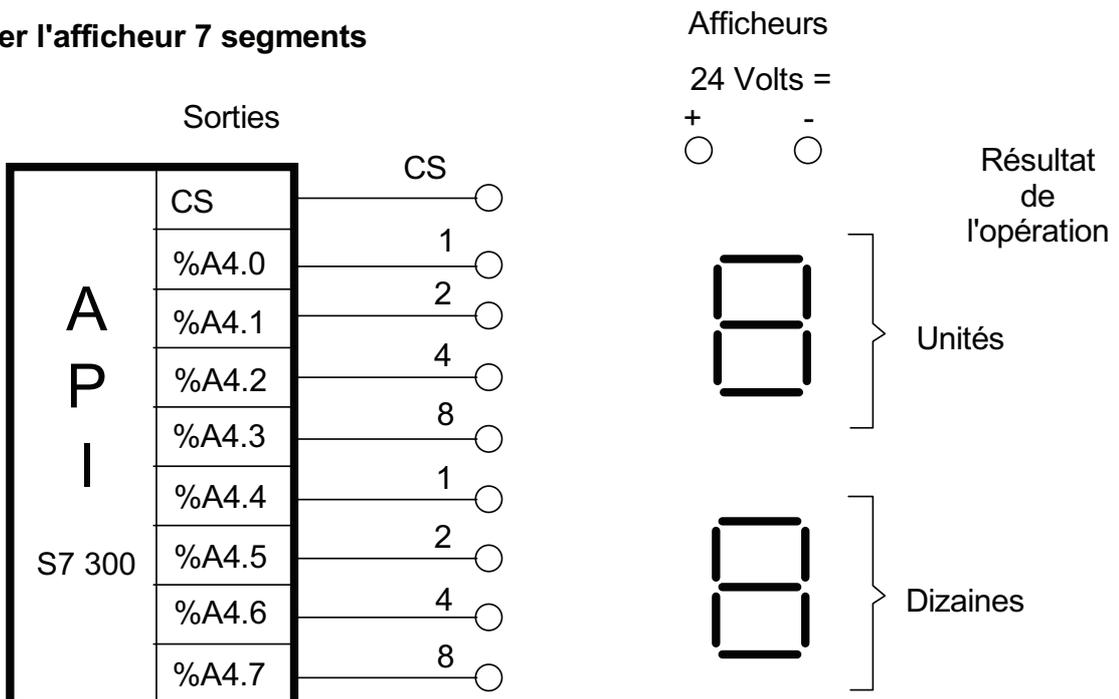
6) Adressage des entrées et des sorties

<i>Point de vue PO</i>	<i>Point de vue PC</i>	<i>Réalisateur</i>
arrêt d'urgence	aru	E0.0
initialisation	init	E0.1
départ cycle	dcy	E0.2
production	prod	E0.3
arrêt fin de production	arrêt	E0.4
manuel	manu	E0.5
rentrer vérin en manuel	mA0	E0.6
sortir le vérin en manuel	mA1	E0.7
validation	val	E1.0
présence carton plein	p	E1.1
vérin tige rentrée	a0	E1.2
vérin tige sortie	a1	E1.3
Voyant d'arrêt d'urgence	VARU	A4.0
Voyant demande d'initialisation	VINIT	A4.1
Rentrer la tige du vérin	A0	A4.2
Sortir la tige du vérin	A1	A4.3
Débloquer la tige du vérin	BT	A4.4

1) Câbler les entrées



2) Câbler l'afficheur 7 segments



3) Etablir le programme

L'objectif de l'exercice est de réaliser un programme effectuant les opérations de bases :

Addition, Soustraction, Multiplication, Division entière, reste de la division entière.

Etablir le programme correspondant à l'algorithme suivant :

DEBUT

Si Validation Alors

 Début

 Lire les roues codeuses et transférer dans la variable **Val_Roues**

 Isoler par masquage le **premier nombre** dans la variable **Nombre1**.

 Isoler par décalage le **deuxième nombre** dans la variable **Nombre2**.

 Fin

Si (**Nombre1** > 0) et (**Nombre2** > 0) Alors

 Début

 Si **Addition** Alors **resultat** := **Nombre1** + **Nombre2**

 Si **Soustraction** Alors

 Début

 Si (**Nombre1** > **Nombre2**) Alors **resultat** := **Nombre1**

-Nombre2.

 Fin

 Si **Multiplication** Alors **resultat** := **Nombre1** * **Nombre2**.

 Si **Division** Alors **resultat** := **Nombre1** / **Nombre2**.

 Si **Reste** Alors **resultat** := **Nombre1** - **Nombre2*** (**Nombre1** / **Nombre2**).

 Fin

Convertir le **resultat en DCB** et transférer le résultat dans la variable **ResultatBCD**.

Transférer **ResultatBCD** sur l'afficheur 7 segments.

FC10

FIN.

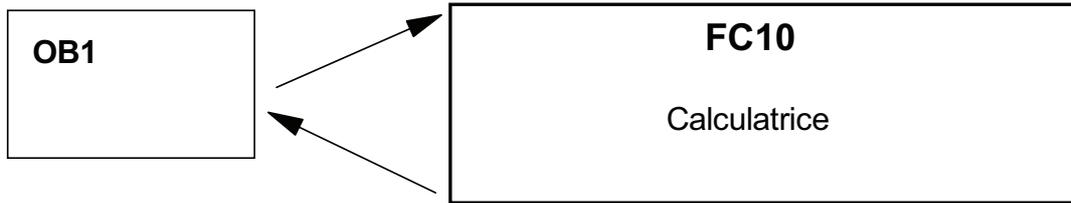
4) Adressage des entrées et des sorties

Désignation	Codage	Automate	
Entrée validation du traitement	Validation	E0.0	
Entrée addition	Addition	E0.1	
Entrée soustraction	Soustraction	E0.2	
Entrée multiplication	Multiplication	E0.3	
Entrée division	Division	E0.4	
Entrée reste	Reste	E0.5	
Nombre 1 bit 1	N1b1	E2.0	EB2
Nombre 1 bit 2	N1b2	E2.1	
Nombre 1 bit 4	N1b4	E2.2	
Nombre 1 bit 8	N1b8	E2.3	
Nombre 2 bit 1	N2b1	E2.4	
Nombre 2 bit 2	N2b2	E2.5	
Nombre 2 bit 4	N2b4	E2.6	
Nombre 2 bit 8	N2b8	E2.7	
Afficheur 1 bit 1	Af1b1	A4.0	AB4
Afficheur 1 bit 2	Af1b2	A4.1	
Afficheur 1 bit 4	Af1b4	A4.2	
Afficheur 1 bit 8	Af1b8	A4.3	
Afficheur 2 bit 1	Af2b1	A4.4	
Afficheur 2 bit 2	Af2b2	A4.5	
Afficheur 2 bit 4	Af2b4	A4.6	
Afficheur 2 bit 8	Af2b8	A4.7	

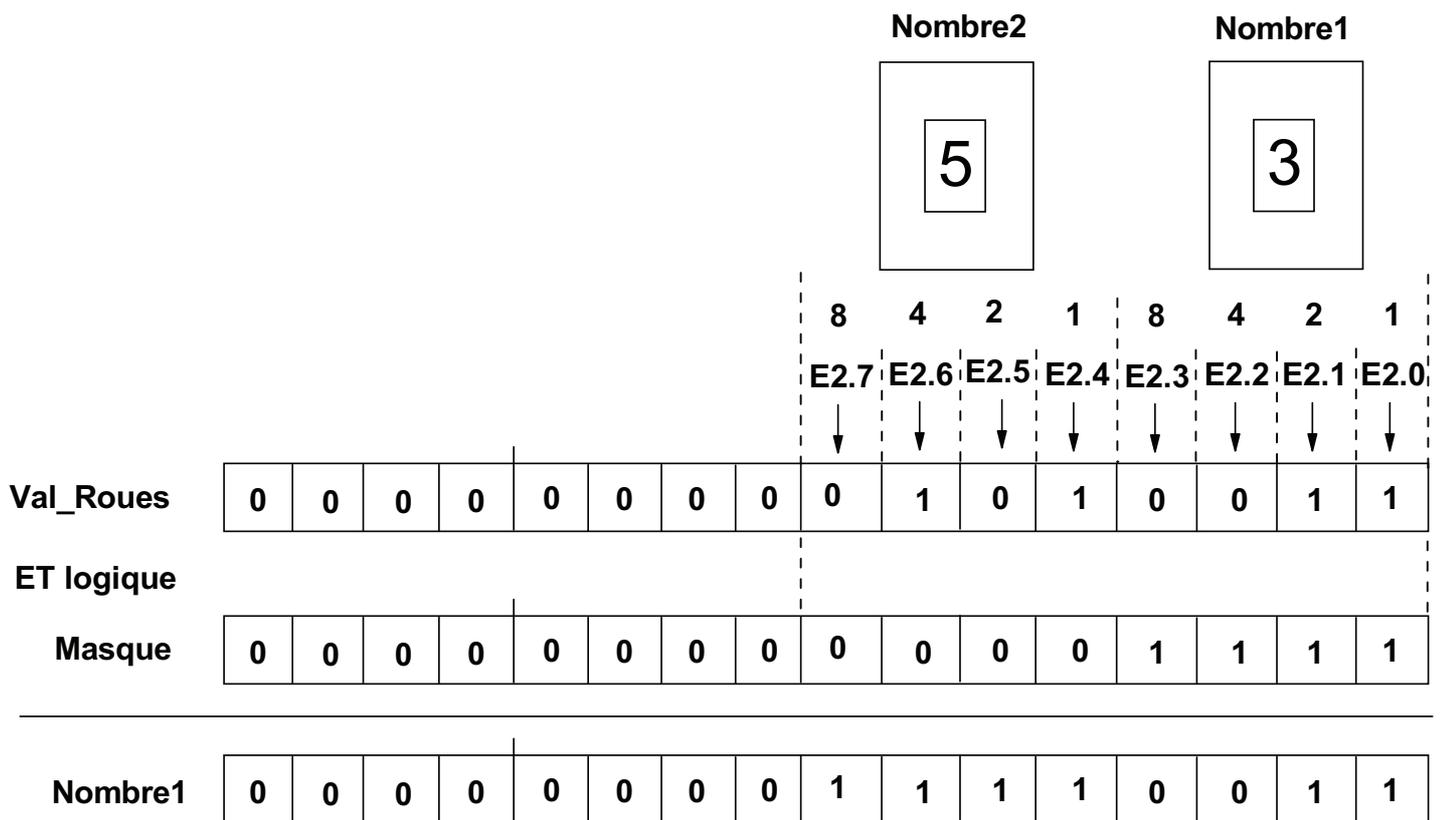
5) Adressage des mots

Désignation	Codage	Automate
valeur en décimal codé binaire sur les roues codeuses	Val_Roues	MW0
Premier nombre obtenu par masquage	Nombre1	MW2
Deuxième nombre obtenu par décalage	Nombre2	MW4
Résultat d'une opération en binaire pur (décimal)	Resultat	MW6
Résultat d'une opération en décimal codé binaire	ResultatBCD	MW8
Calcul intermédiaire du reste de la division entiere	Reste	MW10

6) Structure du programme



7) Mode opératoire pour isoler par masquage le nombre 1



8) Mode opératoire pour isoler par decalage le nombre 2

