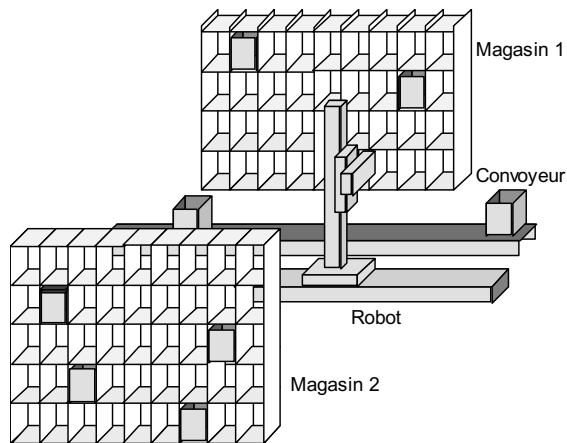


## C5 - GESTION DES AXES X Z



STATION UTILISÉE : TOOL 8

### OBJECTIFS DU TP :

Création d'un FB pour déplacer le bras vers une alvéole dont on précise son numéro.

### CONSIGNES GÉNÉRALES

Chaque étudiant doit faire son application sur son PC. Dans le cas contraire la validation ne sera faite que pour l'étudiant qui l'a faite.

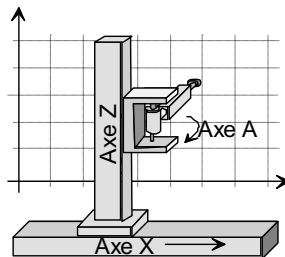
**Vérifiez** la mise en route (bouton 0/I du panneau de commande) du Tool ou des Tools que vous utilisez.  
**Repérez** bien les adresses IP indiquées sur les matériels.

### OBJECTIFS

1. Le programme doit permettre l'initialisation du bras : lorsque l'API passe en RUN (ou bien est mis sous tension), le bras doit se déplacer de gauche à droite et de bas en haut, et s'arrêter lorsqu'il atteint les fins de course "FC Xor" et "FC Zor" sur les deux axes.
2. Le programme doit permettre de déplacer le bras, lorsque "BP START" est appuyé, vers les coordonnées (X=0 à 6 et Z=0 à 5) définies par deux octets : "CmdAxeX" et "CmdAxeZ".
3. Le programme doit permettre de déplacer le bras, à une impulsion sur "BP Start", vers l'alvéole définie par son numéro (de 1 à 54) renseignée sur la variable Input #Alvéole.

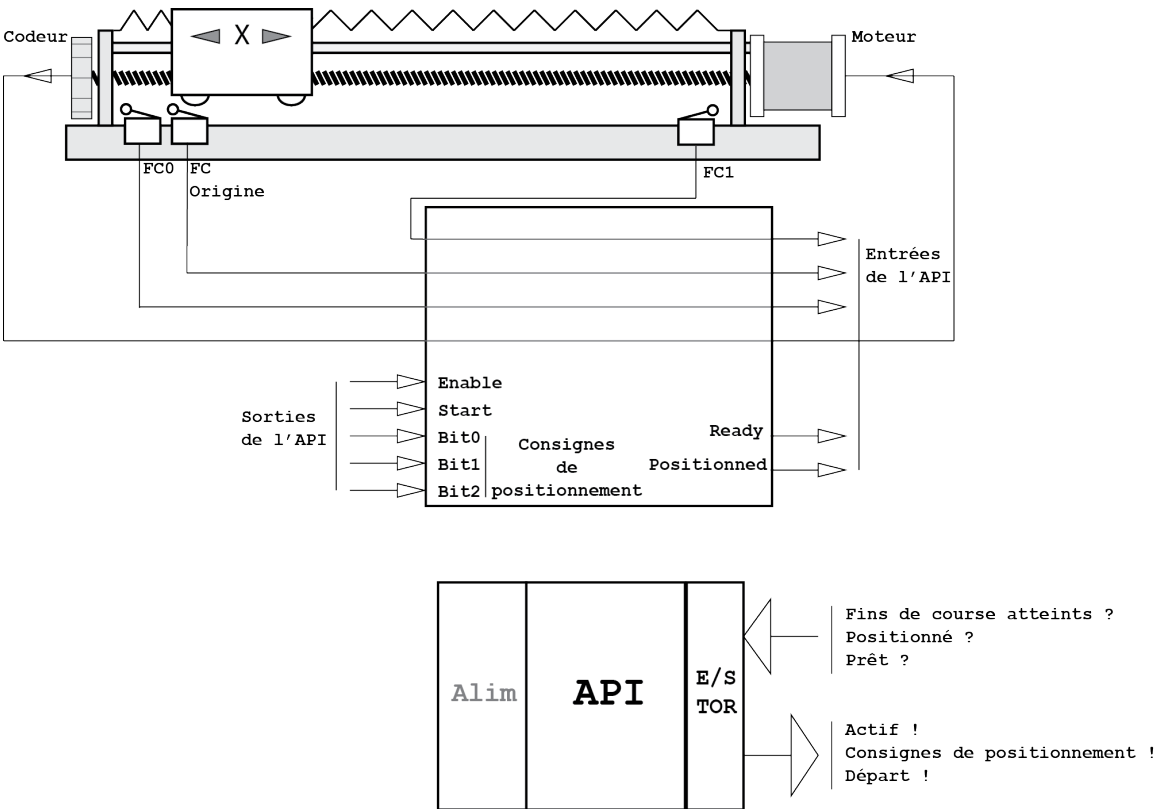
### AVANT-PROPOS

Le robot 2 axes comporte un axe numérique nommé  $x$  pour la translation horizontale et un autre nommé  $z$  pour la translation verticale. L'orientation de la ventouse vers le magasin N°1 ou le magasin N°0 est assurée par un vérin rotatif nommé  $A$ . Ce vérin est commandé par un distributeur monostable en TOR dont la commande est nommée "EV  $A$ ".



### ÉTUDE TECHNOLOGIQUE D'UN AXE X OU Z

Les parties mobiles du robot sont, pour chaque axe  $X$  et  $Z$ , guidées en translation par un dispositif axe et douilles à billes. Le mouvement de translation se fait par l'intermédiaire d'un système vis/écrou à billes. La vis est mise en rotation par un moteur. Un codeur incrémental donne la position angulaire de la vis et, par conséquent, celle de l'ensemble en translation. Un servo-variateur (modèle Ultra 3000i Allen Bradley) asservit l'axe en vitesse et en position.



Commande d'un axe (X ou Z) et liaisons servo-variateur/API

## LE DIALOGUE DE L'API AVEC LES SERVO-VARIATEURS

Le dialogue entre les servo-variateurs et l'API se fait par le biais des entrées et des sorties TOR (cf. schéma page précédente).

Les trois capteurs de fin de course sur les deux axes, ainsi que les sorties des variateurs, sont câblés sur les entrées TOR de l'API (Cf. Figure 1). Des sorties TOR de l'API sont câblées sur des entrées des variateurs.

Le variateur possède un petit API. Un programme, interne au variateur, exécute des mouvements de l'axe pré établis, en fonction de l'état des entrées du variateur.

Une entrée du variateur X est activée par la sortie "EnableX" (et "EnableZ" sur Z). Ceci met sous tension le moteur et provoque le déplacement vers la gauche et vers le haut. Lorsque le moteur n'est pas alimenté, le mouvement est réversible, ceci donne la possibilité de déplacer l'axe à la main. Pensez à retenir le bras lorsque l'alimentation est coupée et à le déplacer légèrement vers la droite.

Trois entrées sur l'axe X (X0, X1 et X2) et sur l'axe Z (Z0, Z1 et Z2) lancent l'exécution d'un programme interne au servo-variateur pour déplacer les axes sur les positions de 0 à 6 (de 000 à 110). La combinaison particulière 111 (7) provoque un déplacement d'une alvéole vers la droite ou vers le haut du bras.

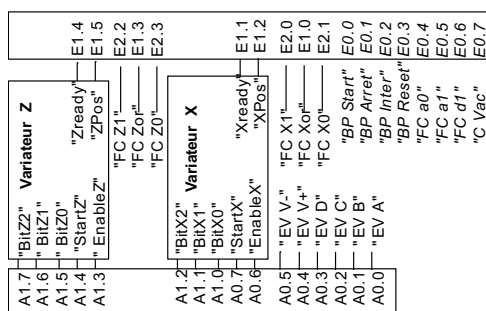


Figure 1 Dialogue entre l'API et les deux servo-variateurs X et Z

**Avertissement :** Les servo-variateurs se mettent en erreur si l'un des bras vient en butée franche (dépassement des fins de course). Au lieu de l'animation habituelle (segments allumés l'un après l'autre), Les afficheurs 7 segments indiquent un code d'erreur (représenté par un chiffre). Il est alors nécessaire de les redémarrer :

- Passez l'API en mode STOP ;
- Maintenez le bouton AU enfoncé (au moins 10 sec.) jusqu'à ce que les servo-variateurs s'éteignent ;
- Remplacez le bras vers la droite (entre les fins de course) ;
- Réarmez l'AU pour redémarrer les servo-variateurs ;
- Vérifiez votre code pour trouver l'erreur qui a provoqué le dépassement des fins de course ;
- Rechargez l'API et repassez en mode RUN.

## OBJECTIFS

Configuration de l'API, Programmation en SCL pour initialiser le bras.

## NIVEAU 1

### OBJECTIFS DU NIVEAU

Configuration de l'API, Programmation en SCL pour initialiser le bras. Le programme doit permettre l'initialisation du bras, c'est-à-dire que lorsque l'API passe en RUN (ou bien est mis sous tension) le bras doit se déplacer de gauche à droite et de bas en haut, et s'arrêter lorsqu'il atteint les fins de course "FC Xor" et "FC Zor" sur les deux axes.

### INSTALLATION DE LA CPU 1512C ET ADRESSAGE DES E/S

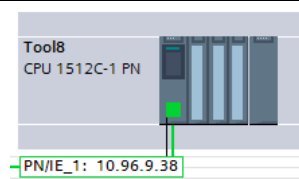
**Ouvrez** un nouveau projet nommé du nom du TP. **Utilisez la notice** pour configurer le matériel si nécessaire.

Dans Appareils & Réseaux, installer une CPU1512c, **insérez** un sous réseau, et **indiquez l'adresse IP (notée sur l'API)**. **Configurez** les adresses E/S :

E/S Analogiques : 1000, 1000

E/S Numériques : 0, 0 (carte 1) et 2, 2 (carte 2).

Cf. notice sur le poste.



Dans la config API, **installez** le module de communication RS232 BA. **Configurez** le module en indiquant les valeurs du protocole (indiquées sur le lecteur).



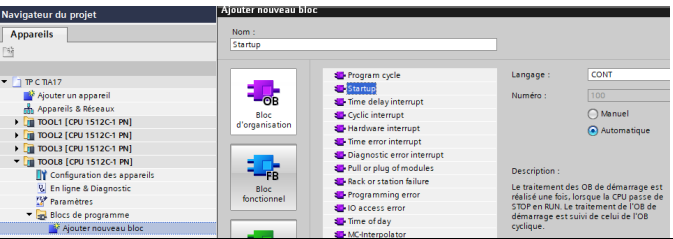
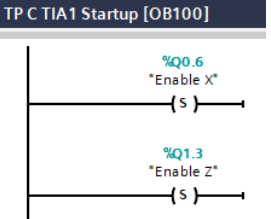
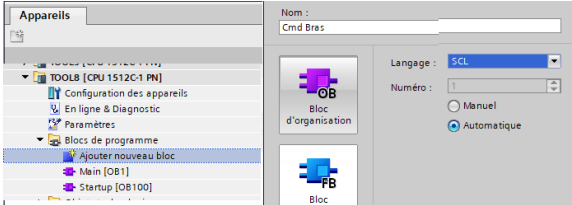
### DÉCLARATION DES VARIABLES GLOBALES

Dans la table des variables, **ajoutez** les variables d'entrées et de sorties indiquées ci-contre.

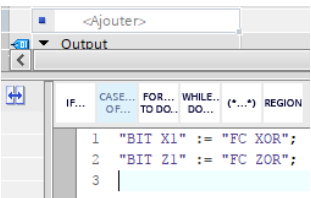
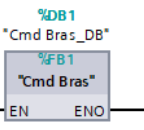
Nota : toutes ces variables ne seront pas utilisées dans ce TP. Elles permettent la gestion complète du stockage-déstockage des boîtes.

Variables API					Variables API				
	Nom	T...	Typ..	Adresse		Nom	T...	Typ..	Adresse
1	EV A	T...	Bool	%Q0.0	22	BP START	T...	Bool	%I0.0
2	EV B	T...	Bool	%Q0.1	23	BP STOP	T...	Bool	%I0.1
3	EV C	T...	Bool	%Q0.2	24	INTER I/II	T...	Bool	%I0.2
4	EV D	T...	Bool	%Q0.3	25	BP RESET	T...	Bool	%I0.3
5	EV V+	T...	Bool	%Q0.4	26	FC a0	T...	Bool	%I0.4
6	EV V-	T...	Bool	%Q0.5	27	FC a1	T...	Bool	%I0.5
7	ENABLE X	T...	Bool	%Q0.6	28	FC d1	T...	Bool	%I0.6
8	START X	T...	Bool	%Q0.7	29	C Vide	T...	Bool	%I0.7
9	BIT X0	T...	Bool	%Q1.0	30	FC XOR	T...	Bool	%I1.0
10	BIT X1	T...	Bool	%Q1.1	31	X READY	T...	Bool	%I1.1
11	BIT X2	T...	Bool	%Q1.2	32	X POS	T...	Bool	%I1.2
12	ENABLE Z	T...	Bool	%Q1.3	33	FC ZOR	T...	Bool	%I1.3
13	START Z	T...	Bool	%Q1.4	34	Z READY	T...	Bool	%I1.4
14	BIT Z0	T...	Bool	%Q1.5	35	Z POS	T...	Bool	%I1.5
15	BIT Z1	T...	Bool	%Q1.6	36	C PRES B...	T...	Bool	%I1.6
16	BIT Z2	T...	Bool	%Q1.7	37	LIBRE	T...	Bool	%I1.7
17	BIP	T...	Bool	%Q2.3	38	FC X0	T...	Bool	%I2.0
18	CONV	T...	Bool	%Q2.4	39	FC X1	T...	Bool	%I2.1
19	L VERTE	T...	Bool	%Q2.5	40	FC Z0	T...	Bool	%I2.2
20	L ORANGE	T...	Bool	%Q2.6	41	FC Z1	T...	Bool	%I2.3
21	L ROUGE	T...	Bool	%Q2.7	42	FC B1	T...	Bool	%I2.4
					43	LIBRE2	T...	Bool	%I2.5
					44	LIBRE3	T...	Bool	%I2.6

## CRÉATION DES BLOCS DE PROGRAMME

<p><b>Ajoutez</b> un nouveau bloc de type Startup. Ce bloc est nommé OB100 par TIA. Il n'est exécuté qu'une fois à la mise en mode RUN.</p>	
	<p>1. Dans l'OB100, <b>installez et paramétrez</b> les actions SET pour forcer à 1 Enable X et Enable Z. A la mise en RUN de l'API ces deux sorties passent à 1 et y restent (bloc OB100).</p> <p><b>Ne pas essayer de charger le programme avant de faire la suite, sinon les deux axes arriveront en butée mécanique et les variateurs se mettront en défaut.</b></p>
<p><b>Insérez</b> un nouveau bloc FB, en langage <b>SCL</b>, que vous nommerez <i>Cmd Bras</i>.</p>	

L'arrêt de l'axe X est provoqué lorsque l'entrée "BIT X1" passe à 1, ce qui se produit lorsque le capteur de fin de course "FC XOR" passe lui aussi à 1 (même chose pour "BIT Z1" avec "FC ZOR").

<p><b>Saisissez</b> les deux lignes pour forcer "Bit X1" lorsque "FC XOR" passe à 1, puis pour forcer "Bit Z1" lorsque "FC ZOR" passe à 1 (copie d'écran ci-contre).</p>	
<p><b>Insérez</b> le FB dans l'OB1 (un DB d'instance est créé avec le même numéro que le FB : DB1 pour FB1).</p>	

## CHARGEMENT DANS L'API

**Sélectionnez** l'API et chargez l'application. **Vérifiez** que tous les voyants de l'API sont verts. Le voyant RUN doit être orange pendant le chargement et vert le chargement terminé. Dans le cas contraire vérifiez la position de l'interrupteur (sous le couvercle de la CPU) il doit être sur la position RUN. Un voyant rouge au niveau de la CPU, indique un défaut de configuration.

## SUR VOTRE COMPTE RENDU :

**Ecrivez** votre programme (SCL et OB100). **Faites valider.**

## VÉRIFICATION NIVEAU 1 (À FAIRE VALIDER PAR L'ENSEIGNANT)

À la mise en RUN de l'API, le bras doit se déplacer vers la gauche et vers le haut et s'arrêter (sans taper sur les butées).

## NIVEAU 2

### OBJECTIFS DU NIVEAU 2

Le programme doit permettre de déplacer le bras, lorsque "BP START" est appuyé, vers les coordonnées (X=0 à 6 et Z=0 à 5) définies par deux octets : "CmdAxeX" et "CmdAxeZ".

### TRAVAIL DEMANDÉ

Dans le FB :

**Ajoutez** la variable Dcy en Input ;

**Saisissez** les deux instructions SCL pour que les sorties "Start X" et "Start Z" aient l'état de l'entrée Dcy ;

Dans l'OB1, **faites un clic droit** sur le FB pour l'actualiser, et **assignez l'entrée TOR** %I0.0 ("BP START") à Dcy.

Dans la table de variables, **déclarez** les deux octets %MB81 et %MB83 en **Byte**. Vous **nommerez** ces 2 octets "CmdAxeX" et "CmdAxeZ".

Ces deux mots sont composés des bits %M81.0 à %M81.7 et %M83.0 à %M83.7. Quand vous donnerez une valeur entre 0 et 7 à "CmdAxeX" (%MB81), les bits %M81.0, %M81.1 et %M81.2 prendront les valeurs 000 à 111. Ceci vaut pour l'axe Z.

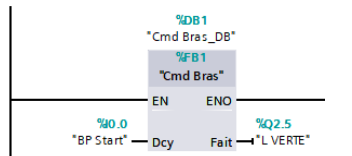
Dans le code SCL, il est possible de noter le bit %M81.b de l'octet "CmdAxeX" "CmdAxeX".%Xb". Ainsi, CmdAxeX".%X1 correspond au bit %M81.1 de l'octet "CmdAxeX". Ceci vaut pour l'axe Z et les autres bits.

**Saisissez\*** les six lignes d'instructions pour que les sorties TOR BIT X0, X1, X2, Z0, Z1 et Z2 prennent l'état des bits %M81.0, %M81.1, %M81.2, M83.0, M83.1 et M83.2. Les lignes seront saisies dans l'ordre des numéros de bits.

**\*On rappelle qu'un bit ne peut être forcé que par une et une seule équation**, Les bits "BIT X1" et "BIT Z1" doivent donc intégrer dans leur définition la première affectation des fins de course FCXOR et FCZOR.

**Saisissez** la ligne pour que la variable output du FB, que vous nommerez #Fait, soit à True lorsque les deux entrées définies par les mnémoniques "X POS" **ET** "Z POS" sont à 1.

Le FB dans l'OB1 peut être affiché en rouge, cela vient du fait du changement des variables Input et Output. Un clic de droite et l'action 'Actualiser' permet de supprimer l'erreur. Renseignez la lampe verte à la sortie #Fait.



**Transférez** dans l'API.

**Insérez** une table de visualisation pour visualisez et forcer les octets MB81 et MB83 (par les icones lunettes et éclair)

Vous pouvez demander l'affichage des trois premiers bits de ces deux octets.

	Nom	Adresse	Form...	Valeur vis...	Valeur de forçage
1	"CmdAxeX"	%MB81	Hexa	16#02	16#02
2		%M81.0	BOOL	<input type="checkbox"/> FALSE	
3		%M81.1	BOOL	<input checked="" type="checkbox"/> TRUE	
4		%M81.2	BOOL	<input type="checkbox"/> FALSE	
5	"CmdAxeZ"	%MB83	Hexa	16#03	16#03
6		%M83.0	B...	<input checked="" type="checkbox"/> TRUE	
7		%M83.1	BOOL	<input checked="" type="checkbox"/> TRUE	
8		%M83.2	BOOL	<input type="checkbox"/> FALSE	

## SUR VOTRE COMPTE RENDU

Votre programme SCL.

## VÉRIFICATION NIVEAU 2 (À FAIRE VALIDER PAR L'ENSEIGNANT)

**Vérifiez** que, lorsque vous appuyer sur le bouton "BP start", le bras se déplace bien vers les positions définies par les octets "CmdAxeX" de 0 à 7, et "CmdAxeZ" de 0 à 5.

On remarquera que lorsque "CmdAxeX" est forcée à la valeur 7 (ses 3 premiers bits sont tous les trois à 1), le bras, depuis sa position en X, se déplace, vers la droite, d'une distance constante égale au pas entre deux alvéoles. **Vérifiez** si la lampe verte s'allume en fin de mouvement.

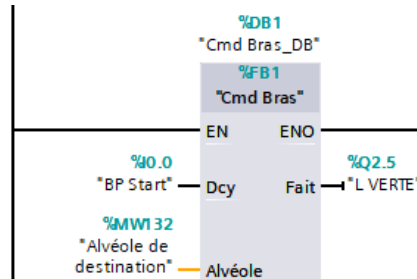
## NIVEAU 3

### OBJECTIFS DU NIVEAU 3

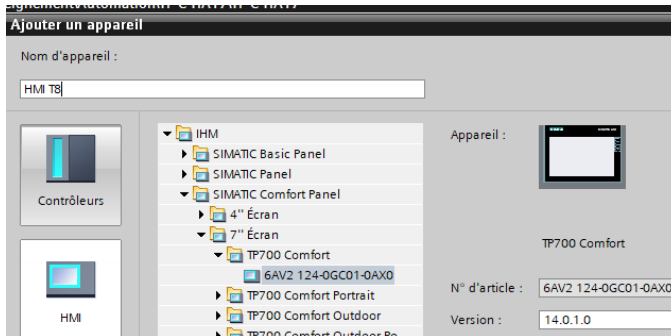
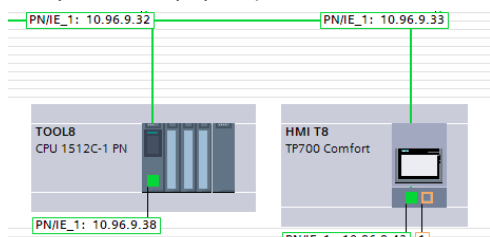
Le programme doit permettre de déplacer le bras, à une impulsion sur "BP Start", vers l'alvéole définie par son numéro (de 1 à 54) renseignée sur la variable Input #Alvéole.

### TRAVAIL DEMANDÉ

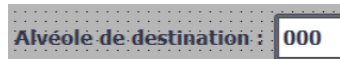
1. **Ajoutez** la variable #Alvéole en Int dans les Input du FB. Ouvrez l'OB1 et actualisez le bloc (clic de droite sur le FB). Renseignez cette Input par %MW132, que l'on renommera "Alvéole de destination".
2. **Ajoutez** un appareil HMI, nommé HMI T8, dont les références sont identiques à celles indiquées sur le pupitre au Tool8.



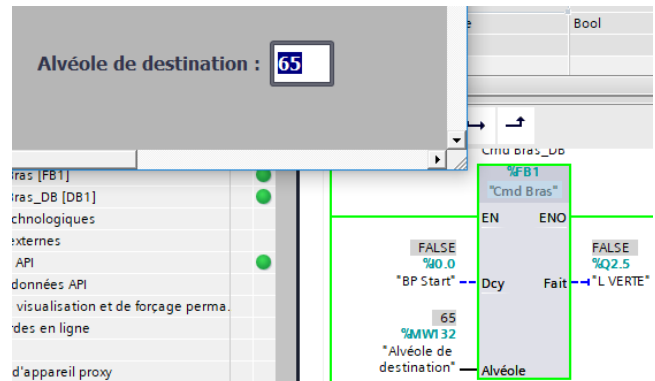
Demandez la connexion avec le Tool8. Indiquez l'adresse IP de l'HMI (voir adresse indiquée sur le pupitre).



Effacez tous les éléments se trouvant dans la vue existante du HMI Tool8, et placez un champ d'E/S pour la variable : "Alvéole de destination"



**Transférez** dans le HMI et dans l'API, **vérifiez** que lorsque vous introduisez une valeur dans le champ d'E/S, la variable "Alvéole de destination" (%MW132) prend bien cette valeur. Cela permet de vérifier que le dialogue entre le HMI et l'API est correctement établi.



Dans les variables Static du FB, **déclarez** les variables :

- #Magasin (Bool) pour sélectionner le magasin avant ou arrière.
- #CmdX, CmdZ (INT) pour y placer les coordonnées en X et Z à atteindre (qu'il faudra transférer, par la suite, vers les deux octets de commande : "CmdAxeX" (0 à 7), et "CmdAxeZ" (0 à 5).
- Enfin la variable #NDI qui correspond au nombre d'incrémentations sur Z pour atteindre les positions de 55 à 81 (pour niveau suivant).

Cmd Bras		
	Nom	Type ..
8	<Ajouter>	
9	Static	
10	Magasin	Bool
11	CmdX	Int
12	CmdZ	Int
13	NDI	Int

La position, numérotée 00, n'est pas une alvéole du magasin, mais la position sur le convoyeur de prise et pose de la boîte.

Celles situées sur les trois colonnes de droite obligent un déplacement à CmdX=6 puis suivi de 1 à 3 déplacements incrémentaux. Les alvéoles de la colonne accessibles par la commande #CmdX=0, mise à part celle qui correspond à la prise de la boîte en butée, ne sont pas utilisées elles sont repérées par un X dans le tableau.

Par exemple, pour se déplacer vers l'alvéole numéro 38, la variable #CmdX doit prendre la valeur 5 et #CmdZ, la valeur 3 et #Magasin doit être forcé à l'état FALSE.

CmdZ

Magasin = TRUE

						1	2	3
						NDI	NDI	NDI
5	X	09	18	27	36	45	54	63
4	X	08	17	26	35	44	53	62
3	X	07	16	25	34	43	52	61
2	X	06	15	24	33	42	51	60
1	X	05	14	23	32	41	50	59

Magasin = FALSE

5	X	04	13	22	31	40	49	58
4	X	03	12	21	30	39	48	57
3	X	02	11	20	29	38	47	56
2	X	01	10	19	28	37	46	55
1								

Positions possibles non utilisées

00

0 1 2 3 4 5 6 6+ 6+ 6+

CmdX

Tapez les lignes de programme en langage SCL pour calculer, en fonction du numéro de l'alvéole à atteindre, les variables de consigne #CmdX et le nombre de déplacements incrémentaux #NDI à effectuer, puis la variable #CmdZ, et la variable #Magasin. La structure CASE permet facilement ces affectations.

Programme à compléter pour tous les cas définis dans le tableau ci-dessus

<pre> 10 CASE #Alvéole OF // pour CmdX et NDI 11 0: 12     #CmdX := 1; #NDI := 0; 13     // ... 14 28..36: 15     #CmdX := 4; #NDI := 0; 16     // ... 17 73..81: 18     #CmdX := 6; #NDI := 3; 19 END_CASE; </pre>	<pre> 20 CASE #Alvéole OF // Pour CmZ et Magasin 21 03, 12, 21, 30, 39, 48, 57, 66, 75: 22     #CmdZ := 4; 23     #Magasin := false; 24 08, 17, 26, 35, 44, 53, 62, 71, 80: 25     #CmdZ := 4; 26     #Magasin := true; 27 END_CASE; </pre>
---	---

Les octets "CmdAxeX" et "CmdAxeZ" doivent prendre la valeur des variables #CmdX et #CmdZ convertis de type INT en type BYTE. La variable "EV A" doit passer à 1 si #Dcy est à 1 et selon la valeur de la variable BOOL #Magasin. **On vérifiera**, en forçant sur le HMI, la valeur de %MW132 renseignée sur l'entrée Alvéole, si le bras prend bien la position de cette alvéole. La commande "EV A", provoquant la rotation du bras, doit être conditionnée par la variable Magasin.

## VÉRIFICATION À FAIRE VALIDER PAR L'ENSEIGNANT

Le bras doit se doit se déplacer vers l'alvéole définie de 0 à 54, par le mot %MW132 forcé sur le HMI du Tool8, lorsque "BP Start" passe à l'état 1.

## SUR VOTRE COMPTE RENDU

Votre programme SCL.

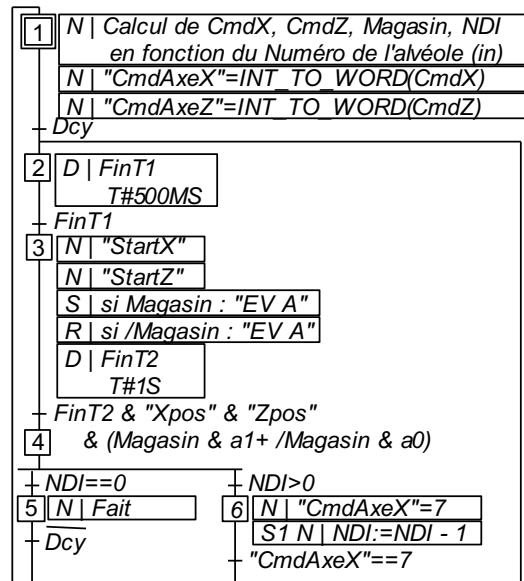
## BONUS : DÉPLACEMENT VERS UNE ALVÉOLE DE 1 A 89

Dans le cas où la position, en X à atteindre, est supérieure à 6, il faut se déplacer vers la position 6 et effectuer un nombre de déplacements incrémentaux représentés par #NDI (Déjà calculés précédemment).

L'étape 1 du grafcet exécute les calculs fait précédemment pour les consigne de déplacement, c'est-à-dire l'affectation des variables : "CmdAxeX" et "CmdAxeZ"

#NDI, #Magasin

- 1 Déclarez, pour les bits d'étape, les variables Static : #X1, #X2... #X6, en Bool. Puis les deux bits de fins de tempo, #FinT1 et FinT2.



### 2 Programmez (en tête de programme SCL) les équations pour les étapes.

```
#X1 := NOT (#X1) AND NOT (#X2) AND NOT (#X3) AND NOT (#X4) AND NOT (#X5) AND NOT (#X6)
OR #X5 AND NOT (#Dcy)
OR NOT (#X2) AND #X1;
#X2 := #X1 AND #Dcy OR #X6 AND "CmdAxeX" = 7
OR NOT (#X3) AND #X2;
#X3 := #X2 AND #FinT1
OR NOT (#X4) AND #X3;
#X4 := #X3 AND #FinT2 AND ("X Pos" AND "Z Pos" AND (#Magasin AND "FC a1" OR NOT (#Magasin)
AND "FC a0"))
OR NOT (#X6 OR #X5) AND #X4;
#X5 := #X4 AND (#NDI = 0)
OR NOT (#X1) AND #X5;
#X6 := #X4 AND (#NDI > 0)
OR NOT (#X2) AND #X6;
```

### 3 Conditionnez par l'étape 1 les calculs déjà programmés

```
IF #X1 = true AND #Alvéole >= 0 AND #Alvéole <= 90 THEN
CASE #Alvéole OF // Calcul de CmdX et Nombre de déplacements
0: #CmdX := 0; #NDI := 0;
1..9: #CmdX := 1; #NDI := 0;
10..18: #CmdX := 2; #NDI := 0;
19..27: #CmdX := 3; #NDI := 0;
28..36: #CmdX := 4; #NDI := 0;
37..45: #CmdX := 5; #NDI := 0;
46..54: #CmdX := 6; #NDI := 0;
55..63: #CmdX := 6; #NDI := 1;
64..72: #CmdX := 6; #NDI := 2;
73..81: #CmdX := 6; #NDI := 3;
END_CASE;
CASE #Alvéole OF
0: IF "C Vide" THEN #CmdZ := 1;
ELSE #CmdZ := 0;
END_IF;
#Magasin := false;
01, 10, 19, 28, 37, 46, 55, 64, 73:
#CmdZ := 2; #Magasin := false;
02, 11, 20, 29, 38, 47, 56, 65, 74:
#CmdZ := 3; #Magasin := false;
03, 12, 21, 30, 39, 48, 57, 66, 75:
#CmdZ := 4; #Magasin := false;
```

```

04, 13, 22, 31, 40, 49, 58, 67, 76:
    #CmdZ := 5; #Magasin := false;
05, 14, 23, 32, 41, 50, 59, 68, 77:
    #CmdZ := 1; #Magasin := true;
06, 15, 24, 33, 42, 51, 60, 69, 78:
    #CmdZ := 2; #Magasin := true;
07, 16, 25, 34, 43, 52, 61, 70, 79:
    #CmdZ := 3; #Magasin := true;
08, 17, 26, 35, 44, 53, 62, 71, 80:
    #CmdZ := 4; #Magasin := true;
09, 18, 27, 36, 45, 54, 63, 72, 81:
    #CmdZ := 5; #Magasin := true;
END_CASE;
"CmdAxeX" := INT_TO_BYTE(#CmdX);
"CmdAxeZ" := INT_TO_BYTE(#CmdZ);
END_IF; // de X1 = true
IF "FC X0" AND NOT #FrontFCX0 THEN
    "CmdAxeX" := 0; "CmdAxeZ" := 0;
    #FrontFCX0 := true;
END_IF;
IF NOT "FC X0" THEN
    #FrontFCX0 := false;
END_IF;
"Bit X0" := "CmdAxeX".%X0;
"Bit X1" := "CmdAxeX".%X1 OR "FC Xor";
"Bit X2" := "CmdAxeX".%X2;
"Bit Z0" := "CmdAxeZ".%X0;
"Bit Z1" := "CmdAxeZ".%X1 OR "FC Zor";
"Bit Z2" := "CmdAxeZ".%X2;

```

### 3. Tapez les lignes pour les autres actions

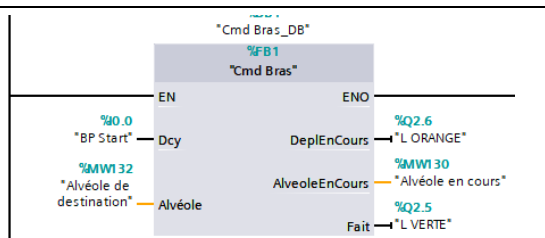
```

// -----Equations des actions placées sur les étapes-----
-----
IF #X6 THEN
    "CmdAxeX" := 7; #NDI := #NDI - 1;
END_IF;
IF #X3 AND #Magasin THEN
    "EV A" := true;
END_IF;
IF #X3 AND NOT #Magasin THEN
    "EV A" := false;
END_IF;
"Start X" := #X3;
"Start Z" := #X3;
#Fait := #X5;
IF #X5 AND NOT #FrontX5 THEN
    #AlveoleEnCours := #Alvéole;
    #FrontX5 := true;
END_IF;
IF NOT #X5 THEN #FrontX5 := false; END_IF;
#ValT1enBCD := S_ODT(T_NO := 1, S := #X2, TV := S5T#500mS, R := "BP Reset", BI => #T1Bin, Q
=> #FinT1);
#ValT2enBCD := S_ODT(T_NO := 2, S := #X3, TV := S5T#1S, R := "BP Reset", BI => #T2Bin, Q =>
#FinT2);
#DeplEnCours := NOT #X1 AND "GBF3"; // Si une des étapes est à 1 (sauf la X1) : cligotement

```

On déclarera toutes les variables qui ne le sont pas.  
Placer #DeplEnCours et #AlveoleEnCours dans les variables Output du FB.

Actualisez le bloc FB dans l'OB1 et renseignez-le



VÉRIFICATION À FAIRE VALIDER PAR L'ENSEIGNANT

Le bras doit se doit se déplacer vers l'alvéole définie de 0 à 89, par le mot %MW132 forcé sur le HMI du Tool8, lorsque "BP Start" passe à l'état 1. La lampe orange doit clignoter pendant le déplacement. La position du bras est visible dans le mot %MW130.