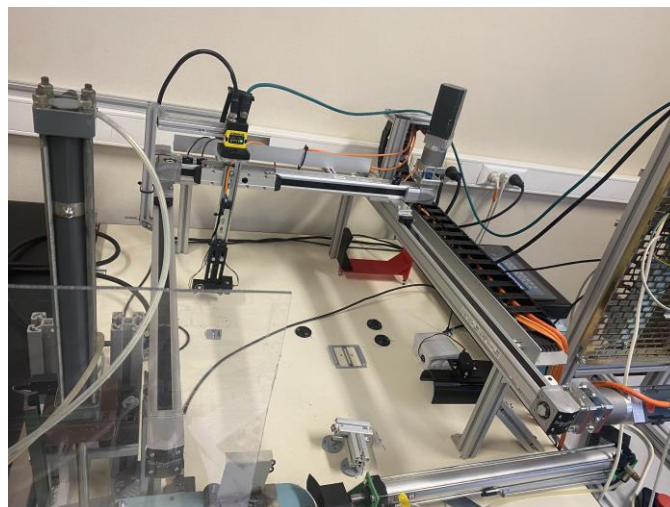
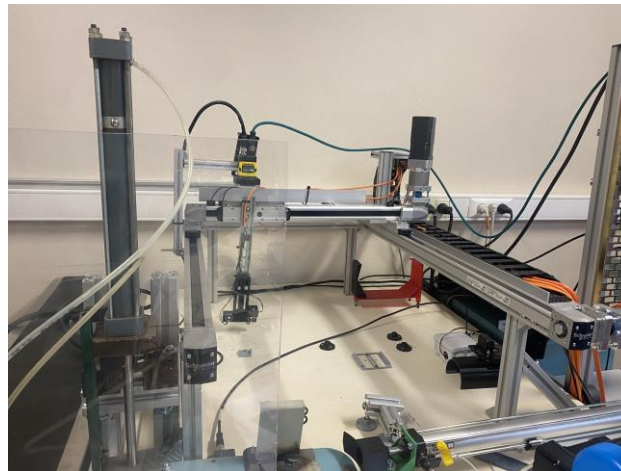


COMPTE RENDU DU TP2 : GESTION D'UN AXE CARTESIEN



Rokhaya TRAORE

Mohamed BAKARY



INTRODUCTION

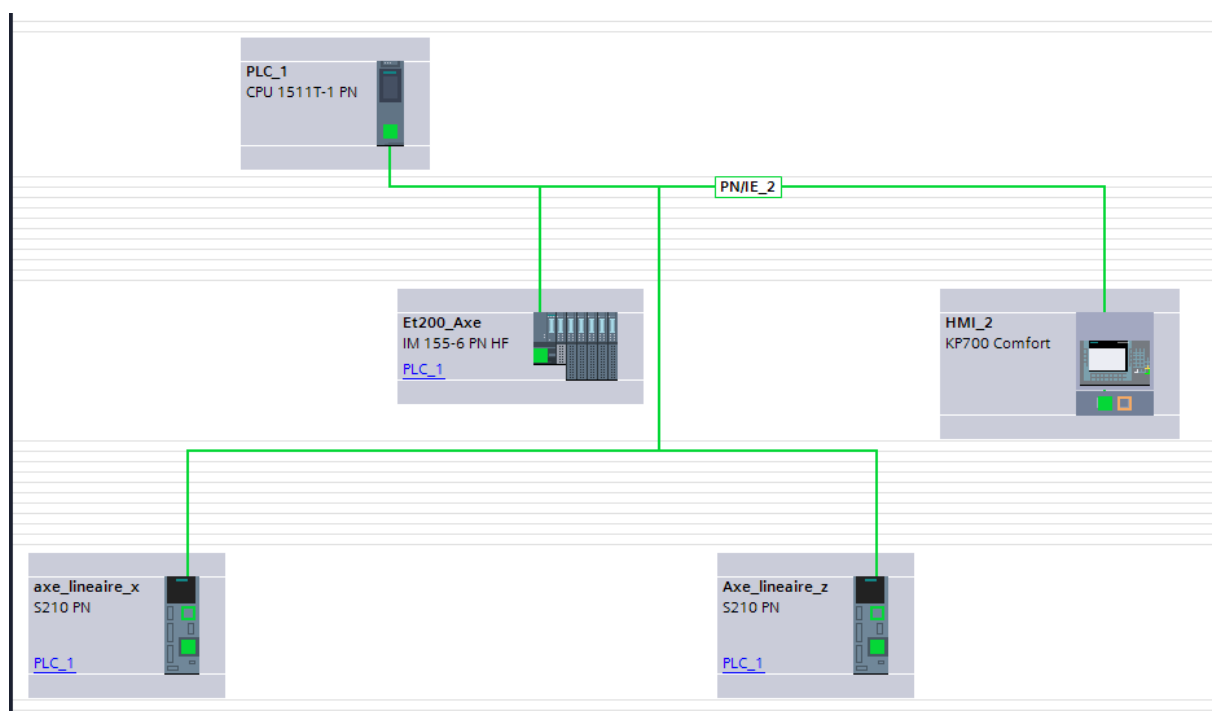
Ce travail pratique (TP) porte sur la gestion d'un axe cartésien au moyen d'un robot équipé de deux axes linéaires contrôlés par des servomoteurs brushless. Le TP vise à nous familiariser avec l'utilisation de nouveaux blocs de fonctions et les systèmes modernes d'asservissement, intégrant des fonctions de commande numérique. Pour cela, il est nécessaire de configurer et de commander le système à l'aide du langage GCODE.

L'objectif principal est de comprendre et de mettre en œuvre les étapes nécessaires pour piloter un système automatisé, tout en respectant les exigences liées à l'automatisation des processus. Cela inclut la gestion de la Prise d'Origine Machine (POM) et la programmation en fonction du cahier des charges.

À travers ce TP, nous serons amenés à gérer les fichiers GCODE et à les intégrer dans une boucle opérationnelle itérative, tout en assurant une sécurité optimale du système.

I. Système de contrôle

Tout d'abord, nous avons configuré une topologie du réseau qui est composée de l'IHM et des composants de l'automate.



Pour mener à bien la programmation au sein de l'automate des variables ont été créés.

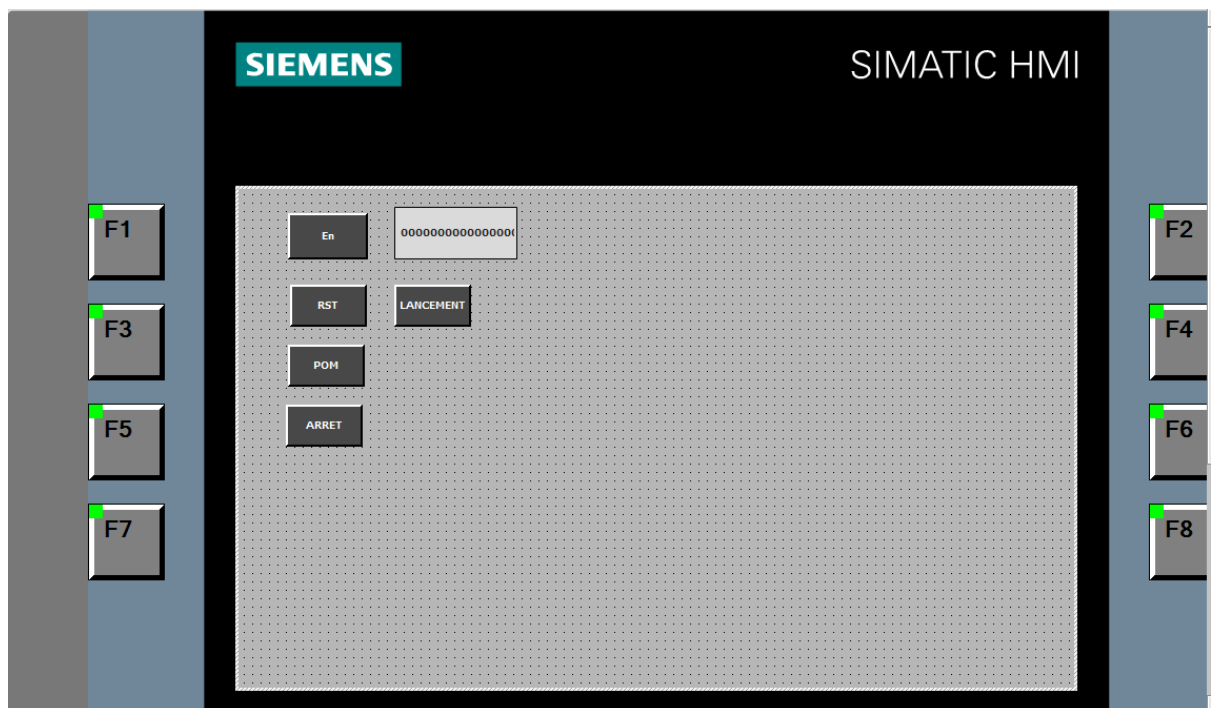
Variables API									
	Nom	Table des variables	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Surveillance
1	Kms	Table de variabl...	Bool	%I4.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	fdn_x	Table de variables s..	Bool	%I4.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	fdp_x	Table de variables s..	Bool	%I4.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	fdp_z	Table de variables s..	Bool	%I4.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	fdn_z	Table de variables s..	Bool	%I4.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Bp_Jaune	Table de variables s..	Bool	%I4.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	V_Bleu	Table de variables s..	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	V_blanc	Table de variables s..	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	V_Vert	Table de variables s..	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	PositioningAxis_1_Actor_Interf...	Table de variables s..	"PD_TEL105_IN"	%I256.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	PositioningAxis_1_Actor_Interf...	Table de variables s..	"PD_TEL105_O..."	%Q256.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	PositioningAxis_1_Actor_Interf...	Table de variables s..	"PD_TEL106_IN"	%I256.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	PositioningAxis_1_Actor_Interf...	Table de variables s..	"PD_TEL106_O..."	%Q256.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	PositioningAxis_2_Actor_Interf...	Table de variables s..	"PD_TEL105_IN"	%I276.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	PositioningAxis_2_Actor_Interf...	Table de variables s..	"PD_TEL105_O..."	%Q276.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	<Ajouter>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

II. Intégration du programme

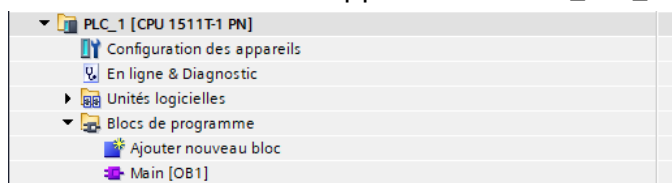
a) Activation et POM

Dans cette partie, nous avons commencé par créer un écran avec 5 boutons pour le bon fonctionnement du système. Ces 5 boutons sont les suivants :

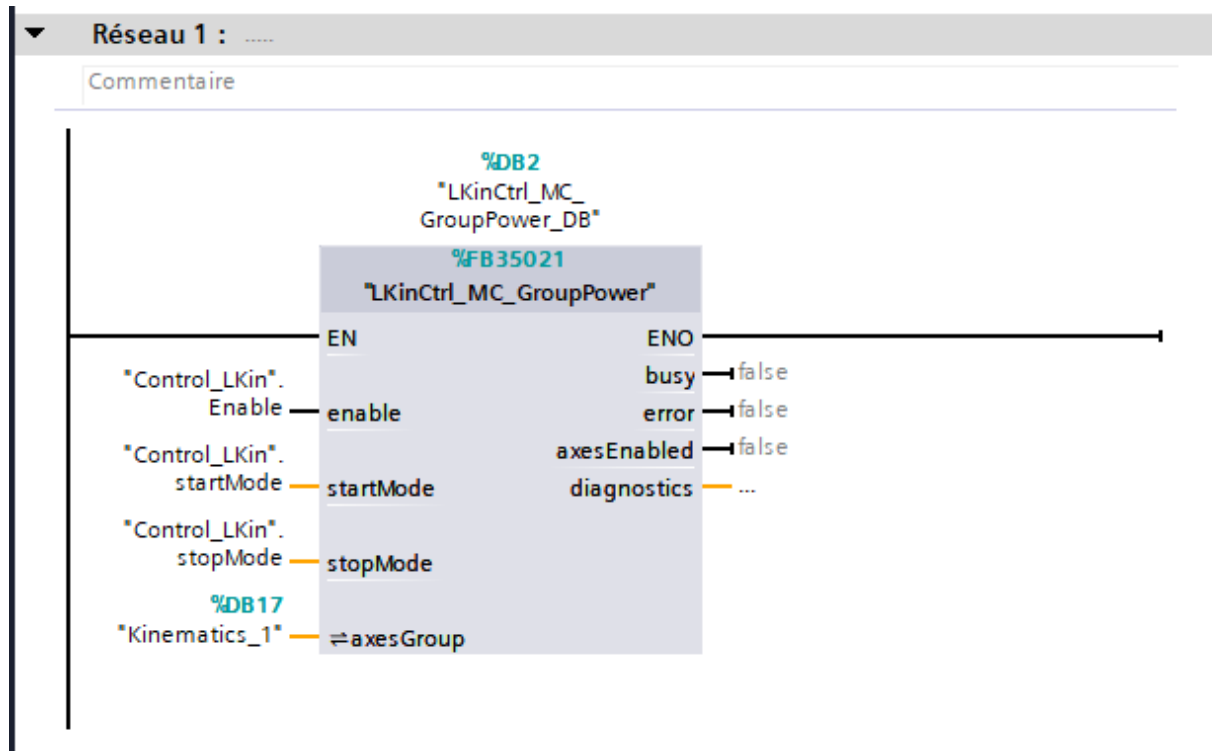
- Activation : Ce bouton est comme un enable. Il donne le feu vert du fonctionnement du système
- Reset : Il permet de “reseter” c’est-à-dire de remettre à zéro ou de supprimer le programme en cours
- POM (Point Origine Machine) : Cela nous permet de demander au système de revenir à sa position initiale ou de départ
- Arrêt : Comme son nom l’indique, ce bouton permet d’arrêter le système
- Lancement : Il permet d’exécuter le GCODE qui a été mis dans Simatic_File_Transfert
- Et enfin nous avons un afficheur pour écrire et afficher le nom du GCODE que l’on veut exécuter



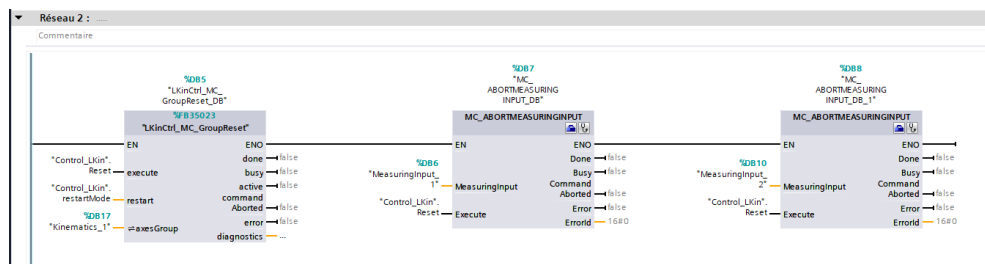
Ensuite, puisque les variateurs ont une puissance coupée, on a pu intégrer dans le “main” une fonction appelée “LKinCtrl_MC_GroupPower”.



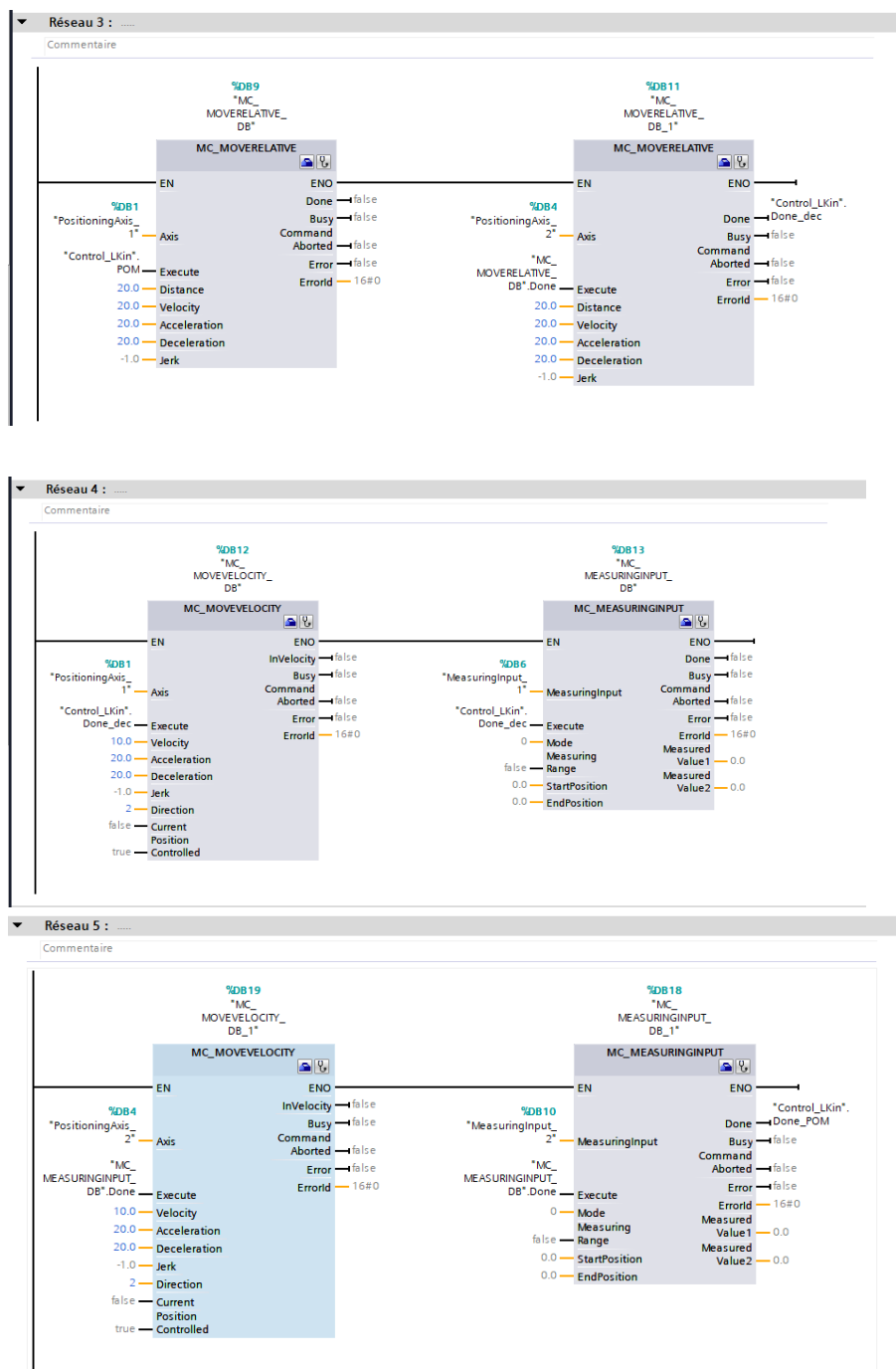
Cette fonction a pour rôle de donner un ordre de validation aux variateurs.



L'étape suivante consiste à la programmation du RESET pour la réinitialisation des variables.



Après que les procédures de préparation sont faites, nous avons intégré des blocs de programme qui gèrent la POM du système.

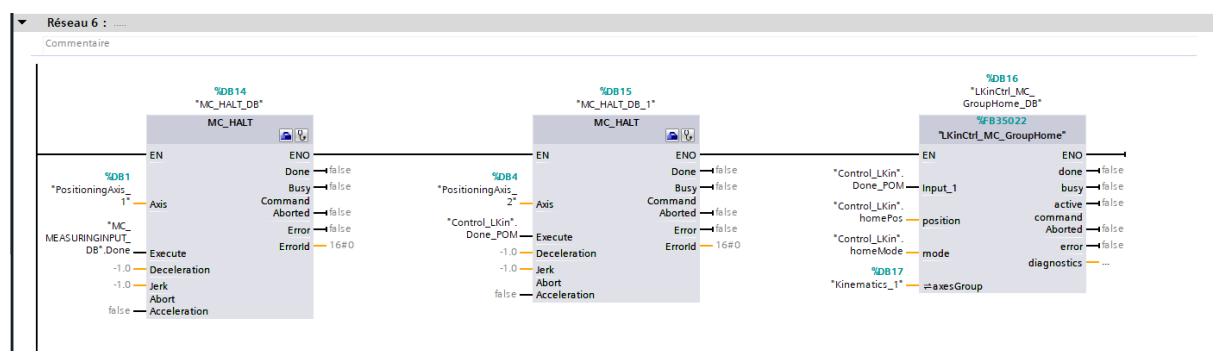


Ces blocs ont pour rôle de permettre aux variateurs d'effectuer une recherche d'origine machine active puisqu'ils peuvent le faire seulement en passive.

- Les deux premiers blocs consistent à faire un décalage positif pour extraire le mobile de la POM.

- Après le décalage, on effectue un retour séquentiellement sur chaque axe. Ce qui est le rôle des deux seconds blocs.
- Et enfin avec les deux derniers blocs, une détection de capteur POM est lancée en parallèle via le détecteur inséré.

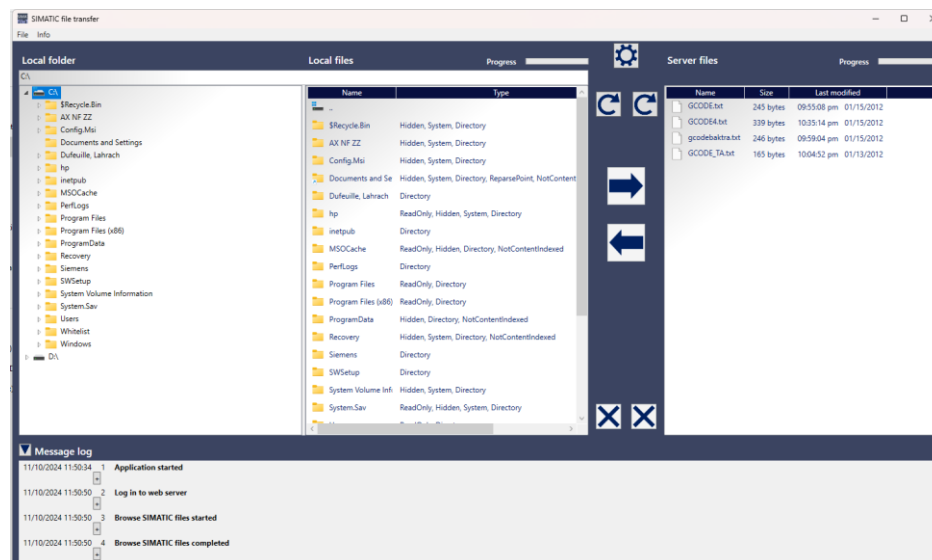
Enfin, nous avons intégré des blocs qui assurent la fin de la POM du système. Ces blocs ont pour rôle de demander un offset lorsque les axes sont arrêtés et qu'ils sont sur leur POM.



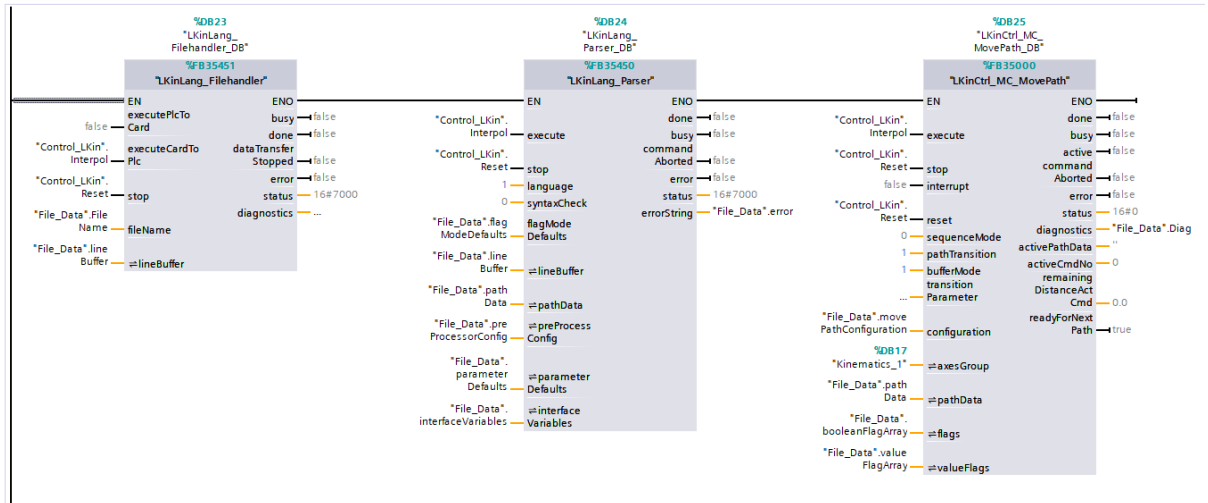
III. GCODE

Le GCODE est un langage de programmation dédié aux commandes numériques. C'est un fichier texte où chaque ligne représente un mouvement.

Et pour utiliser ce fichier dans notre TP, nous avons à notre disposition un logiciel appelé Simatic_File_Transfert pour télécharger un fichier GCODE donné comme exemple dans le webserver de l'automate.

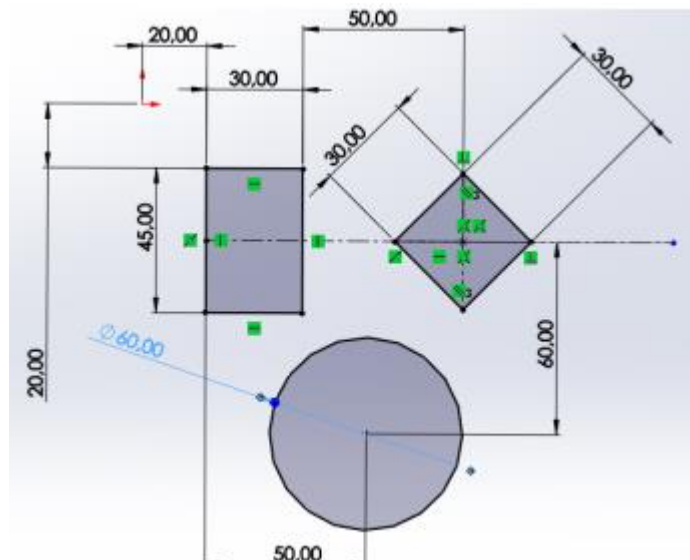


Après avoir chargé le fichier dans l'automate, nous avons créé le lien entre le fichier et l'exécution de mouvement avec des blocs qui sont les suivants:



IV. Création de GCODE

Nous avons à la suite créé et testé des programmes GCODE. Les deux GCODE représentent une suite de formes géométriques que le système exécutera.



Voici les fichiers textes que nous avons créés pour qu'ils correspondent au mouvement décrypté sur l'image ci-dessus :

- Premier GCODE



```
I
F100
G18
G641 BR2
G91
G1 X20 Z0
G1 X0 Z40
F25
G1 X0 Z45
G1 X30 Z0
G1 X0 Z-45
G1 X-30 Z0
F100
G1 X80 Z0
F25
G1 X30 Z30
G1 X-30 Z30
G1 X-30 Z-30
G1 X30 Z-30
F100
G1 X0 Z90
F25
G2 X-60 Z0 I-30 K0
G2 X60 Z0 I30 K0
M30
```

- Deuxième GCODE

```
F100  
G18  
G641 BR2  
G91  
G1 X10 Z0  
G1 X0 Z25  
F25  
G1 X8 Z0  
G2 X2 Z2 I0 K2  
G1 X0 Z6  
G2 X-2 Z2 I-2 K0  
G1 X-8 Z0  
G1 X0 Z-10  
F100  
G1 X17 Z0  
F25  
G1 X2 Z0  
G2 X4 Z4 I0 K4  
G1 X0 Z12  
G2 X-4 Z4 I-4 K0  
G1 X-2 Z0  
G2 X-4 Z-4 I0 K-4  
G1 X0 Z-12  
G2 X4 Z-4 I4 K0
```

kn

V. Les problèmes rencontrés au cours de ce TP

Au cours de ce travail pratique, nous n'avons pas rencontré beaucoup de difficultés. Tout d'abord, la mise en place de la Prise d'Origine Machine (POM) a posé des problèmes de calibration, ce qui a entraîné des erreurs de positionnement lors des premiers tests. Il a fallu ajuster manuellement les paramètres et recalibrer le système pour assurer un bon alignement des axes X et Z.

Et enfin, l'interaction entre le programme GCODE et l'asservissement a parfois généré des comportements imprévus. Ces erreurs étaient souvent liées au fait qu'il mémorisait toujours le premier fichier GCODE. Des ajustements dans les fichiers GCODE et la reconfiguration de certaines commandes ont été nécessaires pour corriger ces dysfonctionnements.



Ces problèmes, bien que contraignants, nous ont permis de mieux comprendre les étapes critiques de la configuration et de la gestion des systèmes automatisés, et d'apprendre à diagnostiquer et résoudre efficacement les erreurs techniques.

CONCLUSION :

Ce travail pratique nous a permis de comprendre et de mettre en œuvre les principes fondamentaux de la gestion d'un axe cartésien au travers un système à deux axes linéaires, contrôlé par des servomoteurs brushless. Grâce à l'utilisation du GCODE, nous avons pu automatiser et piloter les déplacements du robot en respectant les contraintes imposées par le cahier des charges, notamment la gestion de la Prise d'Origine Machine (POM).

Ce TP a également mis en évidence l'importance de la programmation dans l'automatisation des systèmes industriels, en intégrant des étapes de contrôle précises et en assurant un cycle de production fluide, tout en minimisant les risques d'erreurs. Nous avons ainsi renforcé nos compétences dans la configuration et la gestion des systèmes d'asservissement modernes. Enfin, ce TP souligne le rôle clé des interfaces homme-machine (IHM) dans la gestion efficace des automatismes, offrant un contrôle optimal du système et une réponse rapide aux dysfonctionnements.

En conclusion, cet exercice nous a non seulement permis de mieux appréhender le fonctionnement des axes cartésiens, mais aussi de comprendre l'importance de l'automatisation pour la performance et la sécurité dans les environnements industriels.

