



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Automatique
Automatique

Réf. : Entrez la référence du document

Présenté et soutenu par :

Nacer Yaakoub

Benaissa Messaoud

Le :

Simulation Des Bras De Rebot (Stage Guedila)

Jury :

M.	Meghrebi Hassina	Pr	Université de Biskra	Président
M.	Hanane Nebbar	MAA	Université de Biskra	Examineur
Dr.	Abada Khaled	MAA	Université de Biskra	Rapporteur

Année universitaire : 2024 / 2025



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Automatique
Automatique

Réf. : Entrez la référence du document

Simulation Des Bras De Rebot (Stage Guedila)

Le :

Présenté par :
Nacer Yaakoub
Benaissa Messoud

Avis favorable de l'encadreur :
Dr. Abada Khaled

Signature Avis favorable du Président du Jury

Cachet et signature

Résumé :

Dans ce projet de fin d'études, l'objectif est de maîtriser le contrôle à distance d'un bras robotique. En créant, concevant et mettant en place un robot géré par un automate programmable (automatiser) et administré au niveau de l'utilisateur via une -interface machine qui interagit avec l'implémentation selon une norme de communication industrielle OPC UA. Pour le développement de la station, le logiciel de simulation FANUC RoboGuide a été employé, incluant les robots de son répertoire pour réaliser les manœuvres. L'assemblage est pilotée par un automate SIEMENS S7-1500, émulée via l'outil PLCSIM et codée Depuis le TIA Portal. Finalement, une interface utilisateur est intégrée pour la gestion et la supervision de l'implantation, réalisée dans TIA Portal et opérée dans WinCC.

Mots clés : Robotique, FANUC, RoboGuide, HandlingPro ,Automation, SIEMENS, TIA Portal, WinCC, PLC, HMI, KEPServerEX, OPC UA.

Abstract:

The project involves the remote operation of a robotic arm. It aims to devise, design and build a robot that is controlled by a Programmable Logic Controller (PLC) and can be managed by the user through a graphical touch screen interface (HMI). The robot will be interfaced with an industrial standard OPC UA.

In order to develop the station, the ROBOTICS FANC FANUC RoboGUIDE simulation software was utilized as well as the robots from its library to perform the required tasks. The assembly process is under the control of a SIEMENS S7-1500 PLC, which is emulated using PLCSIM and is programmed via TIA Portal. Subsequently, a user interface where the control and supervision of the setup is managed integrated at TIA Portal and opened with winCC.

Keywords: Robotics, FANUC, RoboGuide, HandlingPro, Automation, SIEMENS, TIA Portal, and WinCC, PLC, HMI, KEPServerEX, OPC UA companie's keywords.

Remerciements

Je remercie Dieu le tout Puissant qui m'a donné la force et la volonté pour réaliser ce modeste travail.

Tout d'abord je tiens à remercier le Dr. Khaled Abada d'avoir accepté d'être mon encadreur durant de ce travail, et pour la confiance qu'il m'a donnée et ses précieux conseils.

Mes remerciements vont également aux membres de jury d'avoir accepté de juger mon travail. Megherbi Hassina et Hanane Nebbar

Je remercie l'équipe et les travailleurs d'usine Guedila et spécialement Mr Guerri Mohamed Yacine qui m'a beaucoup aidé dans mon projet j'ai bénéficié de lui beaucoup d'information et de compétences.

Enfin, j'exprime mes sincères remerciements à mes parents, tout l'ensemble de ma famille, pour leur soutien et leur encouragement tout au long de cette période.

Dédicaces

Je dédie ce modeste travail

A mes très chers parents

A mes frères et ma sœur

A toute Ma famille

A tous mes amis

A tous mes camarades de promotion 2024/2025

SOMMAIRE

LA LISTE DES FIGURES

LA LISTE DES TABLEAUX

LA LISTE DES ABREVIATIONS

INTRODUCTION GÉNÉRALE..... 1

CHAPITRE I: DESCRIPTION DU PROCESSUS DE L'USINE GUEDILA ET L'AUTOMATE
SIEMENS S7-1500 3

I.1 Introduction : 3

I.2 Présentation de l'usine de Guedila : 3

I.2.1. Définition : 3

I.2.2. Situation géographique : 3

I.2.3. Historique : 3

I.2.4. Principaux Produits Fabriqués : 3

I.2.5 Perspectives et Objectif de Développement de la société : 4

I.2.6 La ligne de production : 4

I.3 Définition : 5

I.3.1 Pourquoi l'automatisation ? 5

I.3.2 Architecture de l'automate programmable : 5

I.4 L'automate siemens s7-1500 : 6

I.4.1 Constitution de l'Automate S7-1500 : 6

I.4.2 Avantages de l'automate S7-1500 : 6

I.4.3 Modularité du S7-1500: 7

I.4.4 Domaine d'utilisation : 7

I.4.5 Normes de Communication: 7

I.5 Conclusion : 8

CHAPITRE II : DESCRIPTION DES BRAS DE ROBOTS DANS LA CHAÎNE DE
PRODUCTION..... 9

II.1 Introduction : 9

II.2 LA SECURITE EN ROBOTIQUE: 9

II.3 Description du robot FANUC : 10

II.3.1 R-2000iC/100P : 10

II.3.2 R-2000iC/125L : 10

II.3.3 R-2000iC/210WE: 11

II.3.4 R-2000iC/165R: 11

II.3.5 R-2000iC/165F: 11

II.3.6 R-2000iC/ 210F: 12

II.3.7 R-2000iC/ 210R:	12
II.3.8 R-2000iC/ R-2000iC/270F:.....	12
II.3.9 R-2000iC/210L :.....	13
II.3.10 R-2000iC/ 270R:	13
II.3.11 R-2000iC/ 220U:	13
II.4 Présentation du robot FANUC R-2000iC /165F :.....	14
II.4.1 Caractéristiques du robot :.....	14
II.5 Présentation de la baie :.....	15
II.5.1 Les fonctions de la baie :	15
II.5.2 Constitution de la baie :	15
II.5.3 Eléments extérieurs :.....	16
II.5.4 Le panneau opérateur :.....	16
II.5.5 Composants internes :.....	17
II.6 Présentation du Teach Pendant (boîtier d'apprentissage) :	17
II.7 Modes de déplacement du Robot :.....	18
II.7.1 En JOINT : axe par axe :.....	19
II.7.2 Dans un repère universel : WORLD	19
II.7.3 Dans un repère outil : TOOL	20
II.7.4. Dans un repère outil : USER.....	20
II.7.4. Dans un repère outil : JOG FRAME.....	21
II.8 La vitesse de déplacement :	21
CHAPITRE III : SIMULATION ET RÉSULTATS.....	23
III .1 Introduction :	23
III .2 Simulaion de Roboguide :	23
III .2.1 Création d'un environnement de travail « Works Cell » :.....	23
III.2.2 Ouverture le fichier :	23
III.2.2.1 Ouvrez le logiciel Roboguide :	23
III.2.2.2 Sélectionnez « Handling PRO6 » :.....	24
III.2.2.3 Sélectionnez teach pendant pour écrire le programme :.....	24
III.2.3 Les groupes Input/Output :	26
III.2.4. Position_Robot :	28
III.2.5 Correspondance des Signaux :	29
III.2.5.1 Les Entrée UOP :	29
III.2.5.2 Les Sorties UOP :	30
III.3.Configuration matérielle du portail TIA :.....	31
III.4 Création des variables de système :.....	34

III.5 HMI :	35
III.6 Les Blocs :	41
III.6.1 Création du programme :	42
III.6.2 Bloc de conversion AI « OB1 » :	42
III.6.3 Lire AI « FC1 » :	42
III.6.4 Position des axes fanuc_1 [FB1] :	45
III.7 Simulation TIA Portal :	47
III.8 Conclusion :	49
CONCLUSION GENERALE	50
BIBLIOGRAPHIE	

LA LISTE DES FIGURES

Figure 1 : La ligne de production de la société Guedila.....	4
Figure 2: Les automates peuvent être de type compact.....	5
Figure 3 : Les automates peuvent être de type Modulaire.....	6
Figure 4 : Automate modulaire SIMATIC S7-1500 de Siemens.	7
Figure 5: Zones sécurité virtuelles en option (DCS).....	9
Figure 6 : La R-2000iC/100P version.	10
Figure 7 : R-2000iC/125L	10
Figure 8 : R-2000iC/210WE.	11
Figure 9 : R-2000iC/165R.....	11
Figure 10 : R-2000iC/ 165F	11
Figure 11 : R-2000iC/ 210F.	12
Figure 12 : R-2000iC/ 210R.....	12
Figure 13: R-2000iC/ 270F.	12
Figure 14 : R-2000iC/ 210L.....	13
Figure 15 : R-2000iC/ 270R.....	13
Figure 16 : R-2000iC/ 220U.....	13
Figure 17 : Dimension du robot FANUC R-2000iC/165F.....	14
Figure 18 : Eléments extérieurs de la baie.	16
Figure 19 : Panneau opérateur.....	16
Figure 20 : Composants internes de la baie.....	17
Figure 21 : Description du Teach Pendant.	18
Figure 22 : Déplacement du Robot En JOINT.	19
Figure 23 : Déplacement du Robot En WORLD.	19
Figure 24 : Déplacement du Robot En TOOL.....	20
Figure 25 : Déplacement du Robot En USER.....	20
Figure 26 : Déplacement du Robot En JOG FRAME.	21
Figure 27 : HMI DE COMMANDES DE ROBOT.	22
Figure 28 : HandlingPRO6.....	23
Figure 29 : Regard intérieur [HandlingPRO6]	24
Figure 30 : Teach Pendant.....	24
Figure 31 : Programme Manuel	25
Figure 32 : G [I/O].	27
Figure 33 : Position_Robot	28
Figure 34 : UOP (I/O).	29
Figure 35 : mise en réseau.....	31
Figure 36 : vue d'ensemble du réseau.	31
Figure 37 : Vue topologie.....	31
Figure 38 : Vue d'ensemble la topologie.	32
Figure 39 : PLC_2.....	32
Figure 40 : Vue d'ensemble des appareils(PLC).....	32
Figure 41 : Carte CP-1616.	33
Figure 42 : Vue d'ensemble des appareilles(CP_1616).	33
Figure 43 : HMI.....	33
Figure 44 : Vue d'ensemble des appareilles (HMI).	34
Figure 45 : Table de variables PLC_2.	34
Figure 46 : Bloc de données_FANUC_1.	35

Figure 47: Variable HMI.....	35
Figure 48 : Page HMI commande robot en manuel.....	36
Figure 49 : Les buttons des axes en HMI.....	39
Figure 50 : Les Coordonnées des axes en HMI	39
Figure 51: Le bouton marche en HMI.....	40
Figure 52 : Écran pour afficher la vitesse du robot.....	40
Figure 53 : Button Auto/Manuel en HMI.....	41
Figure 54 : Bloc_1 [OB1].....	42
Figure 55 : Réseau 1.....	42
Figure 56 : Réseau 2.....	43
Figure 57: Réseau 3.....	44
Figure 58: Bloc paramétré [FB1].....	45
Figure 59 : Posision_ovoaxes_fanuc1_[DB].....	46
Figure 60 : Réseau 4.....	46
Figure 61 : Charger programme avec PLCSim.....	47
Figure 62 : Liaison en ligne.....	47
Figure 63 : visualisation de réseau 1.....	48
Figure 64 : Visualisation de réseau 2.....	48
Figure 65 : Visualisation de réseau 3.....	48
Figure 66 : Visualisation de réseau 4.....	48
Figure 67: Simulation HMI sans erreurs.....	49

LA LISTE DES TABLEAUX

Tableau 1 : Détails du robot FANUC R-2000iC/165F.	14
---	----

LA LISTE DES ABREVIATIONS

API : Automate Programmable industriel.

CPU : Central Processing Unit.

DB : Bloc des données.

E/S : Entrées/Sorties.

FB : Bloc de fonction.

FC : Fonction.

HMI: Interface homme/machine.

MPI : Multi Point Interface.

OB: Bloc d'organisation.

PC : Partie commande.

PLC : Automate programmable (Programmable Logic Controller).

PLCsim: Programmable Logic Controller Simulator.

PO : Partie opérative.

Tia Portal : Totally integrated automation Portal.

TOR : Tout ou rien.

WinCC : Windows Control Center.

INTRODUCTION GÉNÉRALE

INTRODUCTION GÉNÉRALE

L'Automatique est un ensemble de théories, de techniques, d'outils ... utilisés pour rendre les systèmes autonomes, indépendants de l'intervention humaine, afin de réduire la fréquence et la difficulté des tâches humaines [3].

L'automatique est l'art de modéliser, d'analyser puis de commander les systèmes. C'est aussi celui de traiter l'information et de prendre des décisions [3].

L'évolution de la science de l'automatique a vu émerger le domaine de la Robotique.

La robotique est un ensemble de disciplines techniques (mécanique, électronique, automatique et informatique) articule autour d'un objectif et d'un objet commun [1].

Lorsque nous parlons de robotique, tant d'idées nous viennent à l'esprit, nous pensons à des machines qui ressemblent à des humains et qui effectuent n'importe quelle tâche que l'homme ne peut pas faire. Les gens peuvent même penser à des choses au-delà de l'imagination, mais ce n'est pas une mauvaise idée car relier la science à la science-fiction conduit au développement de nouvelles inventions technologiques.

De nos jours, les industries ont explosé les études sur la robotique en raison de leurs grands besoins de leurs avantages pour assurer le succès économique et pratique dans les projets d'automatisation, ils consistent en de nombreuses activités qui assurent la continuité des productions, la puissance économique, le développement.

Le robot dans les industries améliorera non seulement la capacité de production, mais aussi la qualité des produits en ce qui concerne la précision, la vitesse élevée, la difficulté et la répétabilité. Les robots d'aujourd'hui se déplacent à très grande vitesse avec une trajectoire calculée en protégé et structuré environnement, sans aucune interaction avec les opérateurs humains. Ces robots sont conçus en respectant les contraintes de l'industrie, telles que la répétabilité, les tâches à construire joliment et en respectant le schéma ou le mode de production...

En fait, notre vie quotidienne est pratiquement affectés par les robots, l'objectif de la robotique est de créer des robots pratiques et utiles qui facilitent nos tâches quotidiennes en raison de l'indépendance du robot, ils ont une durée de vie plus longue par rapport aux humains et sont certainement utiles dans l'industrie. Le besoin de robots diffère par leur application, en particulier dans les industries, il existe de nombreuses applications qui peuvent nécessiter des

robots même s'ils imitent les comportements humains mais les appliquent ensuite comportements à la compétence qui conduit les robots à accomplir une certaine tâche.

Les robots ne se fatiguent pas non plus s'ils imitent les comportements humains, mais appliquent ensuite ces comportements à la compétence qui leur permet d'accomplir une certaine tâche, ils ne font pas face aux commandes de manière émotionnelle, par exemple le transport d'objets, la soudure, l'assemblage, la peinture, le tri, la découpe, la palettisation...

La palettisation requiert donc l'application d'un robot plus que toute autre tâche, surtout lorsqu'il s'agit de charges lourdes et de répétabilité.

La robotique est le domaine de connaissances que j'ai inscrit en génie automatisé. D'autre part, la programmation des robots manipulateurs s'avère aujourd'hui d'une grande importance dans les environnements industriels ; et c'est pourquoi nous avons décidés de réaliser un projet lié à la robotique industrielle [11].

D'autre part, il a développé un goût pour l'automatisation industrielle La robotique et l'automatisation sont étroitement liées car pour inclure les robots dans les processus industriels, le contrôle automatisé est nécessaire [5].

Notre objectif dans ce mémoire est la Simulation de bras de robot en utilisant le logiciel ROBOGUIDE de FANUC avec Tia portal et HMI.

Pour bien présenter notre travail, le mémoire sera composé de en trois chapitres qui peuvent être résumés comme suit :

- **Le chapitre 1** : Description du processus de l'usine Guedila et l'automate siemens s7-1500 ;
- **Le chapitre 2** : Description des bras de robots dans la chaine de production ;
- **Le chapitre 3**: Simulation et résultats.

CHAPITRE I :
DESCRIPTION DU PROCESSUS
DE L'USINE GUEDEILA ET
L'AUTOMATE SIEMENS S7-150

CHAPITRE I: DESCRIPTION DU PROCESSUS DE L'USINE GUEDILA ET L'AUTOMATE SIEMENS S7-1500

I.1 Introduction :

Notre étude s'appuie sur le cas concret de l'usine Guedila, spécialisée dans la production et l'embouteillage de l'eau minérale. L'automatisation de cette usine repose sur des systèmes avancés permettant de contrôler l'ensemble du processus de production, de la captation de l'eau à l'emballage final des bouteilles. Ce premier chapitre sera consacré à la description du processus de production de l'usine Guedila, ainsi qu'à une présentation détaillée de l'automate Siemens S7-1500, utilisé pour le contrôle et la supervision des opérations automatisées.

I.2 Présentation de l'usine de Guedila :

I.2.1. Définition :

Guedila est une marque algérienne d'eau minérale prélevée de la source Guedila et exploitée par la société Guedila sur la commune de Djemmorah dans la wilaya de Biskra dans le sud-est algérien [10].

I.2.2. Situation géographique :

L'usine de Guedila est située dans le village de Guedila, Cette localité rattachée à la commune de Djemmorah, située à 30 km au nord de Biskra, en contrebas du massif montagneux d'Auras qui constitue son réservoir d'eau [10].

I.2.3. Historique :

Guedila, du nom d'une petite ville, accrochée à 500 m d'altitude, aux pieds du massif des Aurès d'où elle jaillit, est réputée dans tout le pays pour sa pureté et son équilibre en sels minéraux. La source découverte en 1982 est d'abord régie par un monopole d'Etat puis reprise en 2003 par la famille Hoggui.

La société Guedila, à ses débuts, n'avait aucune connaissance approfondie du marché. "C'était une aventure totale", reconnaît Laid Hoggui, Président-Directeur Général. Cependant, elle a depuis acquis l'expertise nécessaire pour garantir la disponibilité de ses produits et répondre aux exigences du marché ainsi qu'aux attentes d'une clientèle de plus en plus exigeante.

C'est cette logique qui a conduit Guedila, conseillée par un leader en la matière, le bureau d'études Sidel, à investir dans deux lignes d'une capacité de 21000 bouteilles/h en 2004 et la seconde de 36000 bouteilles/h en 2010. Les deux lignes donnant un total de 57000 bouteilles/h [11].

I.2.4. Principaux Produits Fabriqués :

Les principaux produits de la société Guédila sont présentés comme suit :

- ~ Bouteille 0.33 litre normale et sport ;
- ~ Bouteille 0.5 litre normale et sport ;
- ~ Bouteille 1 litre normale et sport ;
- ~ Bouteille 1.5 litre ;
- ~ Bouteille 02 litres.

I.2.5 Perspectives et Objectif de Développement de la société :

L'objectif principal de l'entreprise consiste à satisfaire les exigences des clients en matière de qualité à la hauteur de leurs attentes, et les exigences de ce secteur d'activité (agroalimentaire) dont la certification de l'entreprise par plusieurs normes ISO.

I.2.6 La ligne de production :

La ligne de production est constituée de plusieurs stations et des stations satellites (figure 1).

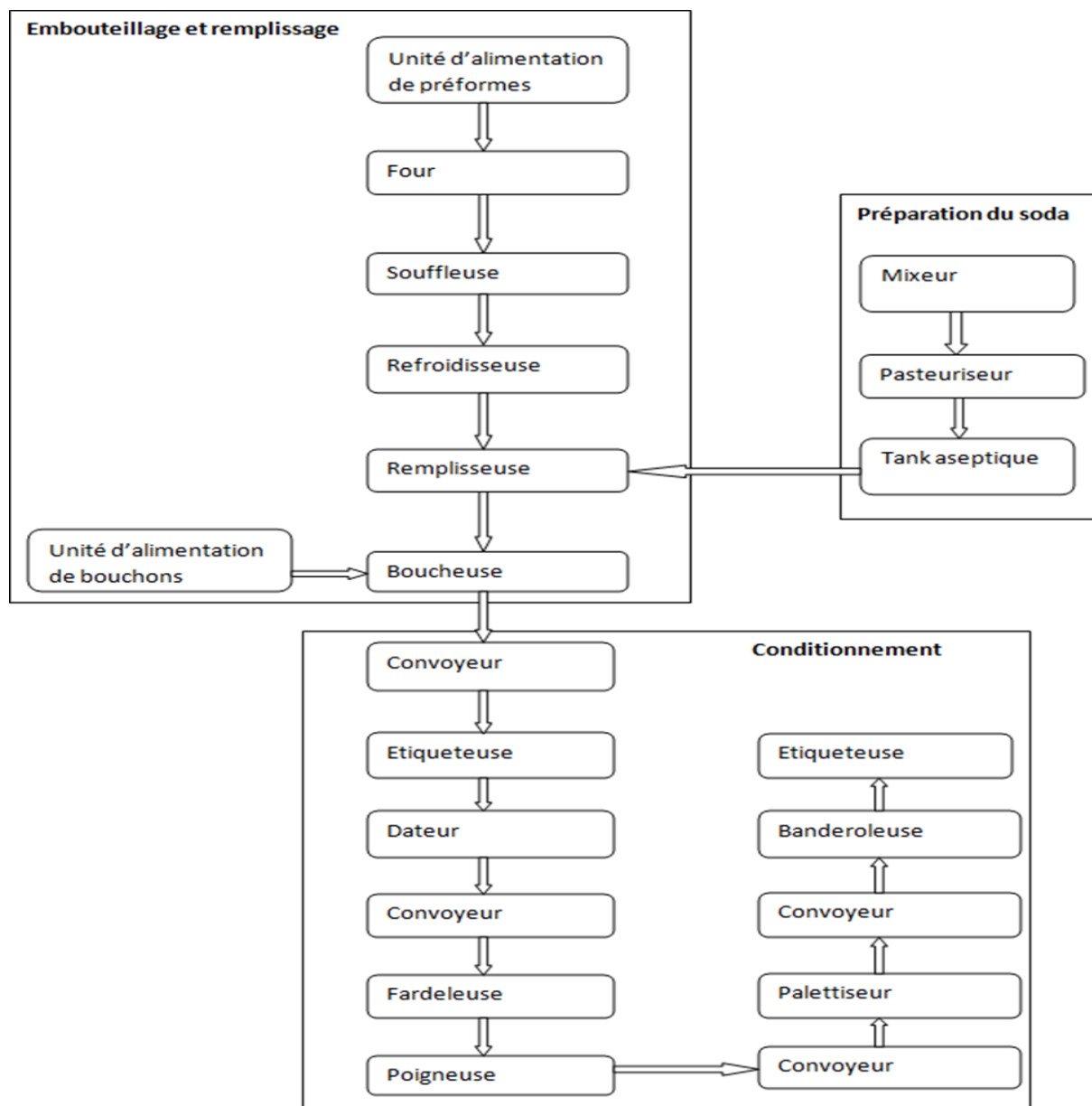


Figure 1 : La ligne de production de la société Guedila.

I.3 Définition :

Un automate programmable industriel (ou API) est un dispositif électronique programmable destiné à automatiser des processus tels que la commande de machines au sein d'une usine et à piloter des robots industriels.

I.3.1 Pourquoi l'automatisation ?

L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée par le système. Ces éléments sont exprimables en termes d'objectifs par :

- ~ Accroître la productivité (rentabilité, compétitivité) du système ;
- ~ Améliorer la flexibilité de production ;
- ~ Améliorer la qualité du produit ;
- ~ Adaptation à des contextes particuliers tel que les environnements hostiles pour l'homme (milieu toxique, dangereux.. nucléaire...), adaptation à des tâches physiques ou intellectuelles pénibles pour l'homme (manipulation de lourdes charges, tâches répétitives parallélisées...)
- ~ Augmenter la sécurité, etc... [2]

I.3.2 Architecture de l'automate programmable :

- **Aspect extérieur :** Les automates peuvent être de type compact ou modulaire.
- **Compact :** On distinguera les modules de programmation (LOGO de SIEMENS, ZELIO de SCHNEIDER, MILLENIUM de CROUZET ...) des micros automates. Il intègre le processeur, l'alimentation, les entrées et les sorties.



Figure 2: Les automates peuvent être de type compact.

- **Modulaire :** Le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur plusieurs racks. Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.

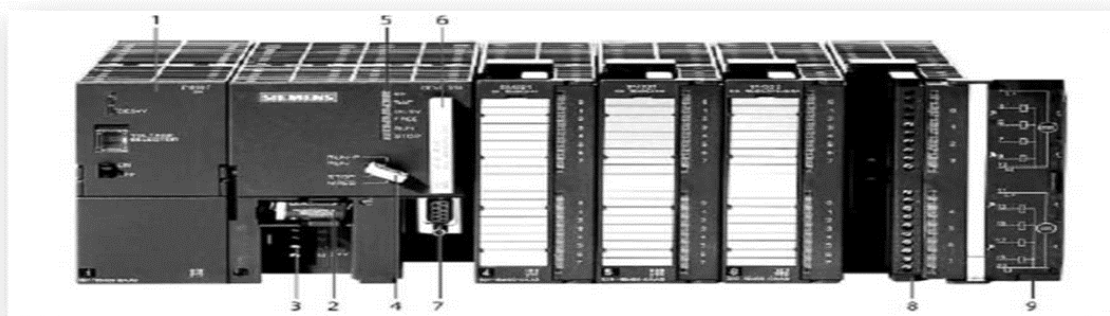


Figure 3 : Les automates peuvent être de type Modulaire.

I.4 L'automate siemens s7-1500 :

Le système d'automatisation SIMATIC S7-1500 est un perfectionnement des systèmes d'automatisation SIMATIC S7-300 et S7-400. Avec l'intégration de nombreuses caractéristiques de performance nouvelles, le système d'automatisation S7-1500 allie convivialité et meilleures performances [7].

I.4.1 Constitution de l'Automate S7-1500 :

Le système d'automatisation SIMATIC S7-1500 se compose des éléments suivants :

- CPU ;
- Modules de périphérie TOR et analogiques ;
- Modules de communication (PROFINET/Ethernet, PROFIBUS, point à point) ;
- Modules technologiques (comptage, détection de position) ;
- Alimentation système.

Vous montez le système d'automatisation S7-1500 sur un profilé support. Il peut se composer de 32 modules maximum. Vous reliez les modules entre eux par des connecteurs en U.

I.4.2 Avantages de l'automate S7-1500 :

Les avantages de l'automate S7-1500 de Siemens sont nombreux et en font un choix privilégié dans le domaine de l'automatisation industrielle. Ces avantages incluent Fiabilité : Le S7-1500 est réputé pour sa fiabilité et sa robustesse :

- **Flexibilité :** La modularité permet d'adapter le contrôleur à diverses applications ;
- **Intégration Facile :** Il s'intègre facilement aux systèmes existants grâce à ses protocoles de communication standard.
- **Maintenance Simplifiée :** La structure modulaire facilite la maintenance et le dépannage [8].

I.4.3 Modularité du S7-1500:

La modularité de l'automate S7-1500 est illustrée dans la figure suivante :

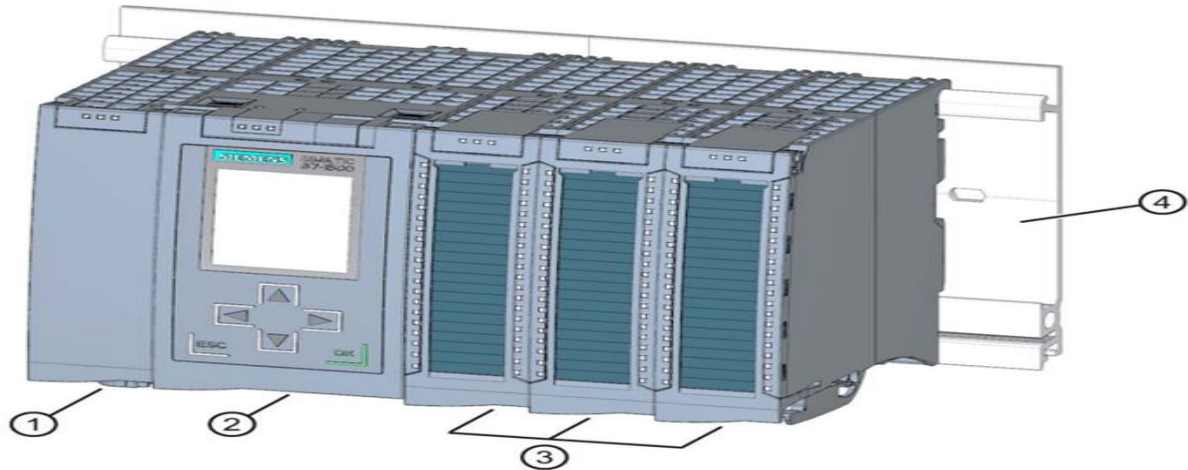


Figure 4 : Automate modulaire SIMATIC S7-1500 de Siemens.

Les composants d'automate modulaire SIMATIC S7-1500 de Siemens sont présentés ci-après conformément à la figure précédente :

1. Alimentation ;
2. CPU ;
3. Modules d'Entrée/Sortie (Input/Output) ;
4. Rack.

I.4.4 Domaine d'utilisation :

Le système d'automatisation S7-1500 offre la flexibilité et la puissance nécessaires à un large éventail d'applications d'automatisation dans la construction de machines et d'installations. Sa structure modulaire vous permet d'adapter votre automate aux conditions sur site. Vous pouvez réaliser des applications pour la technique de sécurité grâce à l'utilisation de CPU S7-1500 de sécurité. Vous procédez à la configuration et à la programmation de votre programme de sécurité, tout comme la configuration et la programmation de vos CPU standard, dans TIA Portal. Le système d'automatisation S7-1500 est homologué pour l'indice de protection IP20 et conçu pour être installé dans une armoire électrique [7].

I.4.5 Normes de Communication:

Les normes de communication d'automate modulaire SIMATIC S7-1500 sont énumérées ci-après :

- ❖ PROFINET IO ;
- ❖ PROFIBUS DP ;
- ❖ Point à point (RS232, RS485) ;

- ❖ Communication IHM ;
- ❖ Serveur Web intégré ;
- ❖ Technologie intégrée ;
- ❖ Diagnostic système intégré ;
- ❖ Fonctions de protection intégrées ;
- ❖ Mode de sécurité (en cas d'utilisation de CPU de sécurité) ;

I.5 Conclusion :

Dans ce chapitre, nous avons présenté une description globale de l'usine de Guédila, en mettant en lumière son fonctionnement, ses principaux processus industriels et les technologies utilisées. Une attention particulière a été accordée aux Automates Programmables Industriels (API), qui jouent un rôle central dans l'automatisation et le contrôle des opérations. Parmi ces automates, le Siemens S7-1500.

CHAPITRE II :
DESCRIPTION DES BRAS DE
ROBOTS DANS LA CHAÎNE DE
PRODUCTION

CHAPITRE II : DESCRIPTION DES BRAS DE ROBOTS DANS LA CHAÎNE DE PRODUCTION

II.1 Introduction :

FANUC, fondée en 1956 par le Dr. Seiueemon Inaba, est pionnière dans l'automatisation industrielle grâce à l'introduction des commandes numériques. De l'automatisation d'une simple machine à celle de lignes entières de production, l'entreprise a révolutionné l'industrie avec des technologies avancées comme le moteur électrique à impulsion et les robots industriels. Ses innovations, telles que ROBOCUT, ROBODRILL et ROBOSHOT, ont optimisé la productivité des usines dès les années 70. Aujourd'hui, avec plus de 4,9 millions de commandes numériques et 810 000 robots installés, FANUC est un leader mondial, poursuivant son engagement envers l'innovation et l'amélioration des processus de production.

Bras Fanuc est une entreprise spécialisée en robotique industrielle et en automatisation. La société a été fondée en 1982 en tant que joint-venture entre la société française Groupe Bull et la société japonaise Fanuc Ltd. Aujourd'hui, Bras Fanuc est un leader mondial dans le domaine de la robotique et de l'automatisation, avec une large gamme de produits et services utilisés dans les industries du monde entier.

II.2 LA SECURITE EN ROBOTIQUE:

Les dernières évolutions des normes (EN 10218-1, EN 10218-2, EN autorisent des 13849) solutions nouvelles de sécurité.

La solution logicielle intelligente intégrée en option par FANUC, DCS (Dual Check Safety), est conçue pour garantir la sécurité totale des opérateurs, des machines et des outils. Elle évite, d'investir dans des équipements de sécurité coûteux et encombrants et réduit au minimum l'espace requis pour les cellules robotisées [4]..

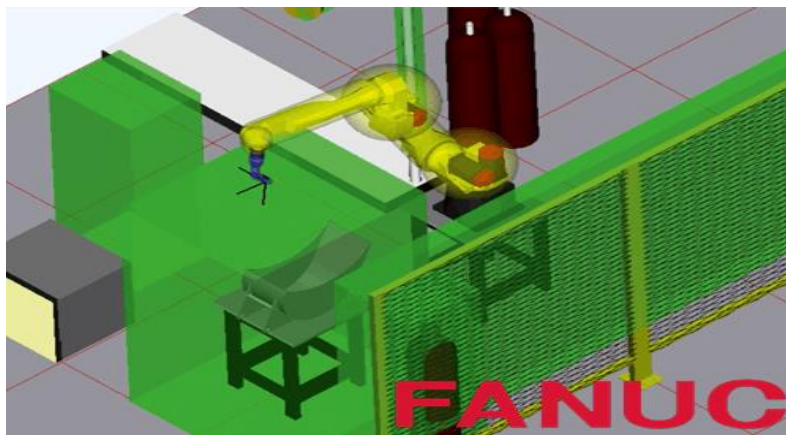


Figure 5: Zones sécurité virtuelles en option (DCS).

II.3 Description du robot FANUC :

La série R-2000 est le robot FANUC qui rencontre le plus de succès dans le monde. Doté de fonctionnalités de haute performance pour des capacités de charge allant de 100 à 250 kg, c'est la solution idéale pour des applications multiples impliquant des charges élevées. Compact et ultra rapide avec son poignet et son bras fins, le R-2000 est une solution complète conçue pour toutes les applications de manutention et de soudage par points.

La série de robots FANUC R-2000iC est la dernière génération de robots 6 axes, fortes charges et hautes performances. Sa large gamme de charges embarquées (de 100 à 210 kg) et ses hautes performances en font la solution parfaite pour de nombreuses applications [9].

 **Le R-2000iC est disponible en 11 versions.**

II.3.1 R-2000iC/100P :

RAYON D'ACTION : 3540 mm **CHARGE UTILE :** 100 kg bras très long. Présenté dans la figure suivante :



Figure 6 : La R-2000iC/100P version.

II.3.2 R-2000iC/125L :

RAYON D'ACTION : 3100 mm **CHARGE ADMISSIBLE :** 125 kg bras long. Présenté dans la figure suivante :



Figure 7 : R-2000iC/125L

II.3.3 R-2000iC/210WE:

RAYON D'ACTION : 2450 mm **CHARGE UTILE :** 210 kg.



Figure 8 : R-2000iC/210WE.

II.3.4 R-2000iC/165R:

PORTÉE : 3095 mm **CHARGE ADMISSIBLE :** 165 kg.



Figure 9 : R-2000iC/165R.

II.3.5 R-2000iC/165F:

PORTÉE : 2655 mm **CHARGE ADMISSIBLE :** 165 kg



Figure 10 : R-2000iC/ 165F

II.3.6 R-2000iC/ 210F:

PORTÉE : 2 655 mm charge admissible : 210 kg.



Figure 11 : R-2000iC/ 210F.

II.3.7 R-2000iC/ 210R:

PORTÉE : 3095 mm charge admissible : 210 kg.



Figure 12 : R-2000iC/ 210R.

II.3.8 R-2000iC/ R-2000iC/270F:

RAYON D'ACTION : 2655 mm charge admissible : 270 kg.



Figure 13: R-2000iC/ 270F.

II.3.9 R-2000iC/210L :

RAYON D'ACTION : 3100 mm charge admissible : 210 kg.



Figure 14 : R-2000iC/ 210L.

II.3.10 R-2000iC/ 270R:

REACH: 3095 mm **LOAD CAPACITY:** 270 kg.



Figure 15 : R-2000iC/ 270R.

II.3.11 R-2000iC/ 220U:

RAYON D'ACTION: 2518 mm **CHARGE UTILE:** 220 kg.



Figure 16 : R-2000iC/ 220U.

II.4 Présentation du robot FANUC R-2000iC /165F :

Modèle polyvalent de la gamme R-2000, la flexibilité et la capacité de charge élevée de ce modèle 6 axes définit de nouvelles normes dans presque toutes les industries. Équipé d'un poignet fin et capable d'atteindre d'excellents temps de cycles, ce robot de soudage par points agile convient aux applications automobiles et similaires. Il a également été conçu pour optimiser l'utilisation de l'espace, ce qui lui permet de travailler à proximité d'autres robots.

II.4.1 Caractéristiques du robot :

Le tableau suivant détaillé les caractéristiques du robot FANUC R-2000iC/165F :

Axe	Répétabilité (mm)	Masse unité mécanique (kg)	Rayon (°)						Vitesse de mouvement (°/s)						J4 Moment/ Inertie (Nm/kgm ²)	J5 Moment/ Inertie (Nm/kgm ²)	J6 Moment/ Inertie (Nm/kgm ²)
			j1	j2	j3	j4	j5	j6	j1	j2	j3	j4	j5	j6			
6	± 0.05*	1090	370	136	312	720	250	720	130	115	125	180	180	260	940/120	940/120	940/100

Tableau 1 : Détails du robot FANUC R-2000iC/165F.

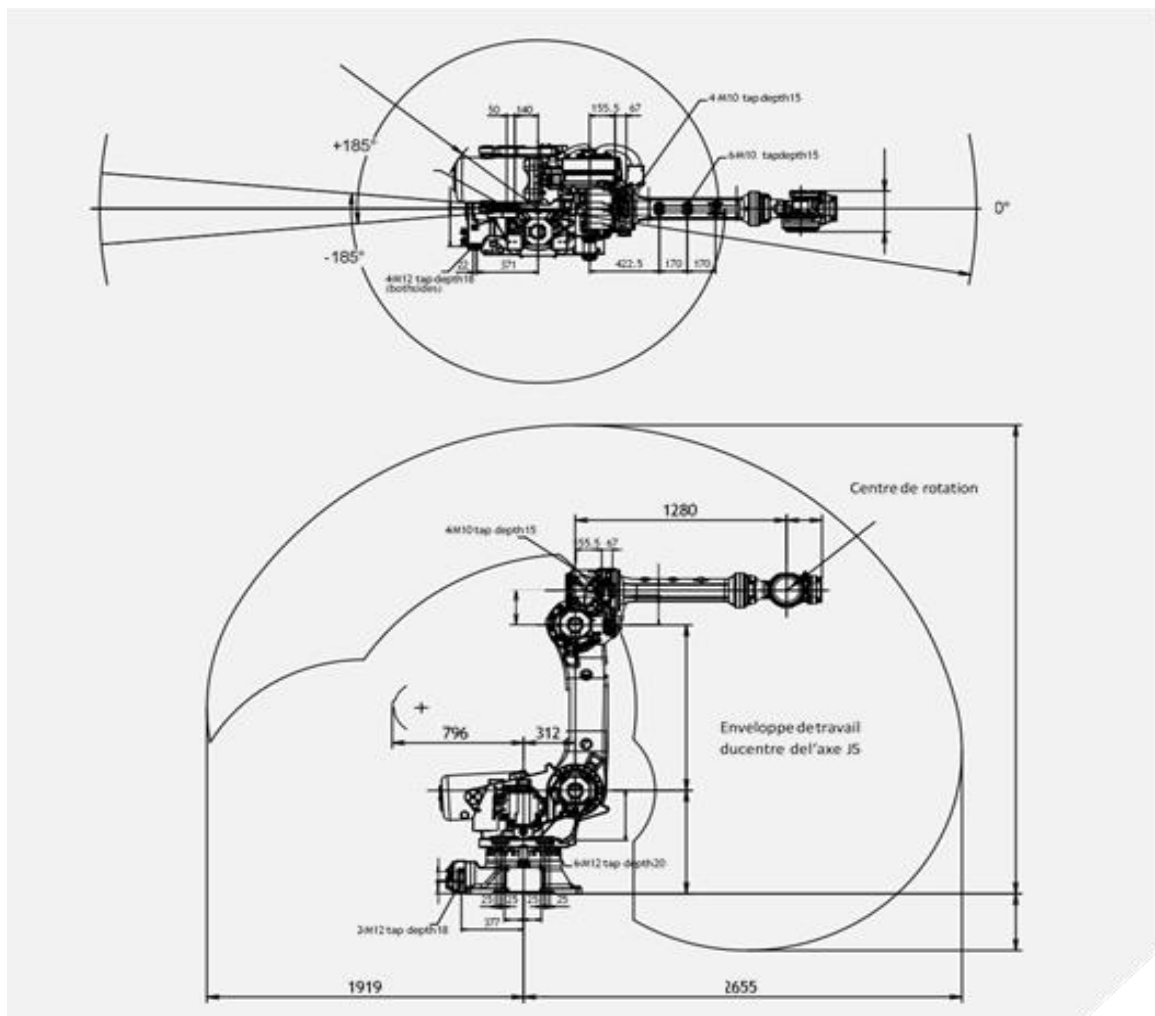


Figure 17 : Dimension du robot FANUC R-2000iC/165F.

II.5 Présentation de la baie :

Une **baie** (ou "rack" en anglais) est une structure métallique utilisée pour installer, organiser et sécuriser des équipements électroniques, informatiques ou de télécommunication dans un espace optimisé. Dans cette section, nous présenterons principalement leurs fonctions ainsi que leur constitution.

II.5.1 Les fonctions de la baie :

Les fonctions principales de la baie sont énumérées ci-après :

- ✓ Elle regroupe un ensemble d'équipements nécessaires pour faire fonctionner un robot ;
- ✓ C'est à partir de la baie que le robot est commandé, que ce soit en MANUEL ou en AUTOMATIQUE. Le robot peut être contrôlé par l'intermédiaire du Teach Pendant ou par le panneau opérateur situé en face avant de la baie ;
- ✓ La pince électrique est commandée avec un variateur (7ème axe) situé à l'intérieur de la baie. Elle recueille donc les informations en provenance de la pince (défauts, position de la pince...) ;
- ✓ Elle dialogue avec tous les périphériques (automate, armoire d'encollage, générateur soudure ASA, séquence ARO....) par réseau ou par les entrées/sorties ;
- ✓ Les chaînes de sécurités sont également gérées au niveau de la baie ;
- ✓ Les différentes alimentations, nécessaires au bon fonctionnement du robot.

II.5.2 Constitution de la baie :

Elle est constituée d'une armoire dans laquelle on retrouve les éléments principaux tels :

1. Un sectionneur ;
2. La carte CPU ;
3. Les alimentations ;
4. Le variateur de commande des 6 axes du robot ;
5. Le variateur pince électrique (le cas échéant) ;
6. La carte de sécurité ;
7. Le module d'entrées/sorties ;
8. Le module changeur de pince (si changeur).

II.5.3 Éléments extérieurs :

Les éléments extérieurs d'une baie (rack) sont les composants visibles et accessibles qui forment sa structure et facilitent son utilisation. Voici une liste des principaux éléments extérieurs présenté dans la figure suivante :

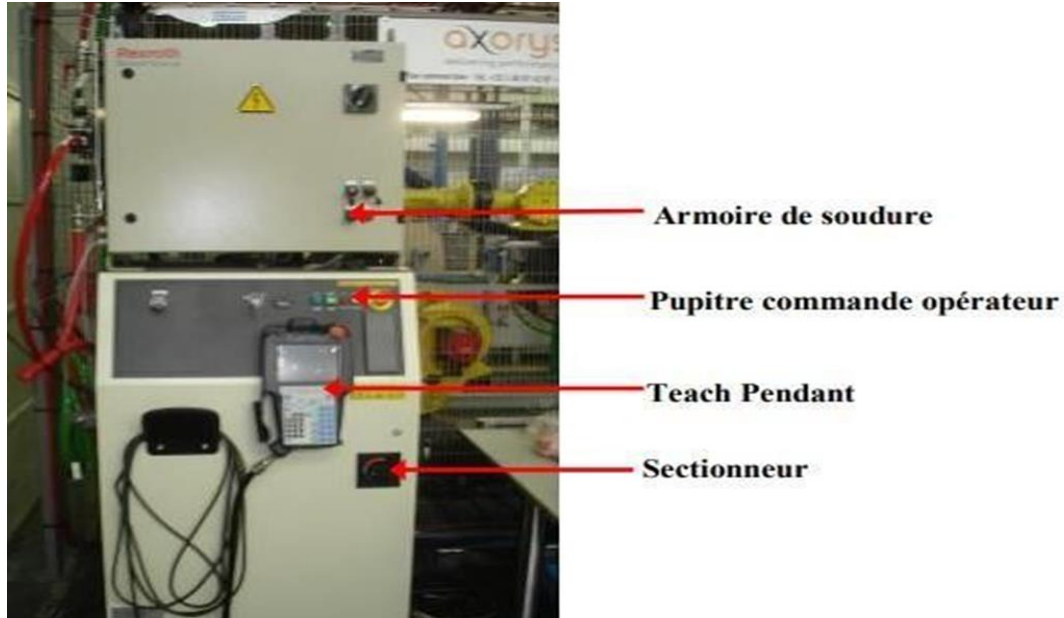


Figure 18 : Eléments extérieurs de la baie.

II.5.4 Le panneau opérateur :

Un **panneau opérateur** est une interface homme-machine (HMI, Human-Machine Interface) utilisée pour permettre aux opérateurs d'interagir avec des machines, des systèmes de contrôle ou des équipements industriels. Il joue un rôle clé dans les environnements de production en facilitant la supervision, le contrôle, et le diagnostic des systèmes. Ce dernier est présenté d'une manière détaillé dans la figure ci-après :



Figure 19 : Panneau opérateur.

II.5.5 Composants internes :

Les composants internes d'une baie sont dépendants de son utilisation spécifique informatique, télécommunications, automation industrielle, etc. Voici dans la figure suivante les composants internes souvent présentés dans une baie industrielle :

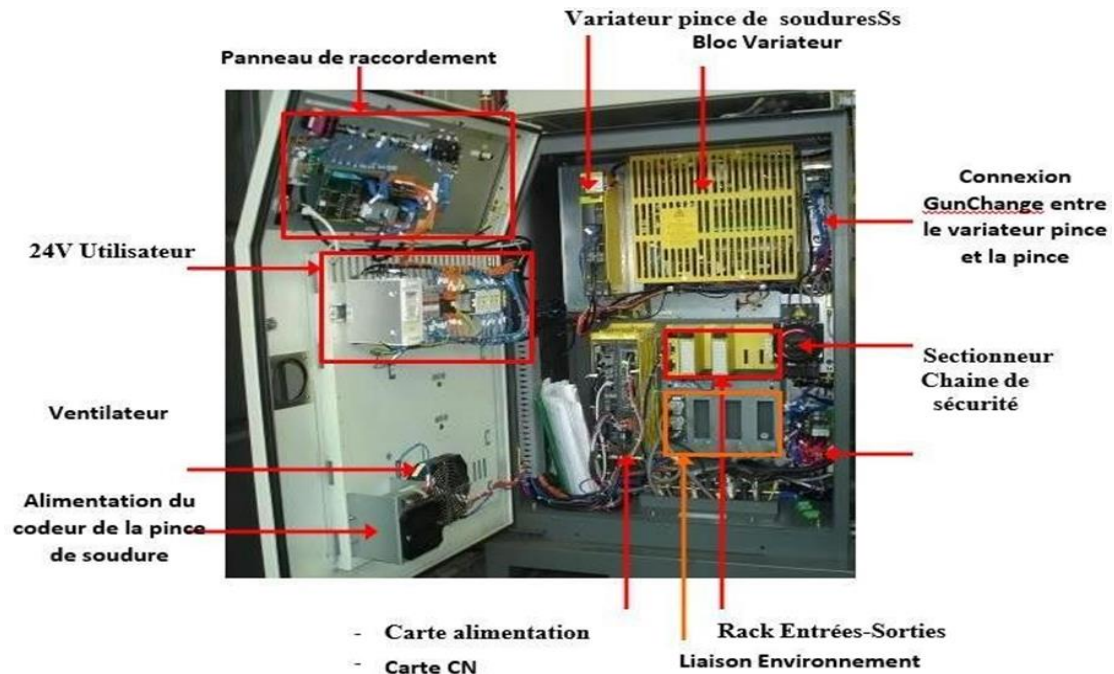


Figure 20 : Composants internes de la baie.

II.6 Présentation du Teach Pendant (boîtier d'apprentissage) :

Le Teach Pendant assure l'interface entre l'armoire et l'opérateur. Il est directement connecté à l'unité centrale. Il est utilisé pour :

- ❖ Déplacer le robot ;
- ❖ Stopper le robot ;
- ❖ Régler la vitesse du robot ;
- ❖ Sélectionner les différents repères (TOOL, USER, WORLD...) ;
- ❖ Sélectionner les différents modes de déplacements ;
- ❖ Faire les mouvements en manuel (ferm. Préh., ouv. Préh.) ;
- ❖ Visualiser les défauts, attentes, les inhibitions... ;
- ❖ Créer les trajectoires ;
- ❖ Tester des programmes ;
- ❖ Visualiser les différentes données (E/S, registres.....) ;
- ❖ Faire les sauvegardes et les restitutions des programmes ;
- ❖ Calibrer le robot ;
- ❖ Calibrer la pince électrique. Il comprend :

- ✓ Un B.P. d'Arrêt d'Urgence.
- ✓ Un commutateur ON / OFF (T.P. actif ou non actif).
- ✓ « L'homme mort » (Deadman).
- ✓ Un écran 16 lignes et 40 caractères.
- ✓ Onze leds, témoins d'état.
- ✓ Toutes les touches (déplacements, vitesse, coordonnées, menus...).
- ✓ Des touches configurées spécialement pour faciliter l'utilisation d'ARCTOOL

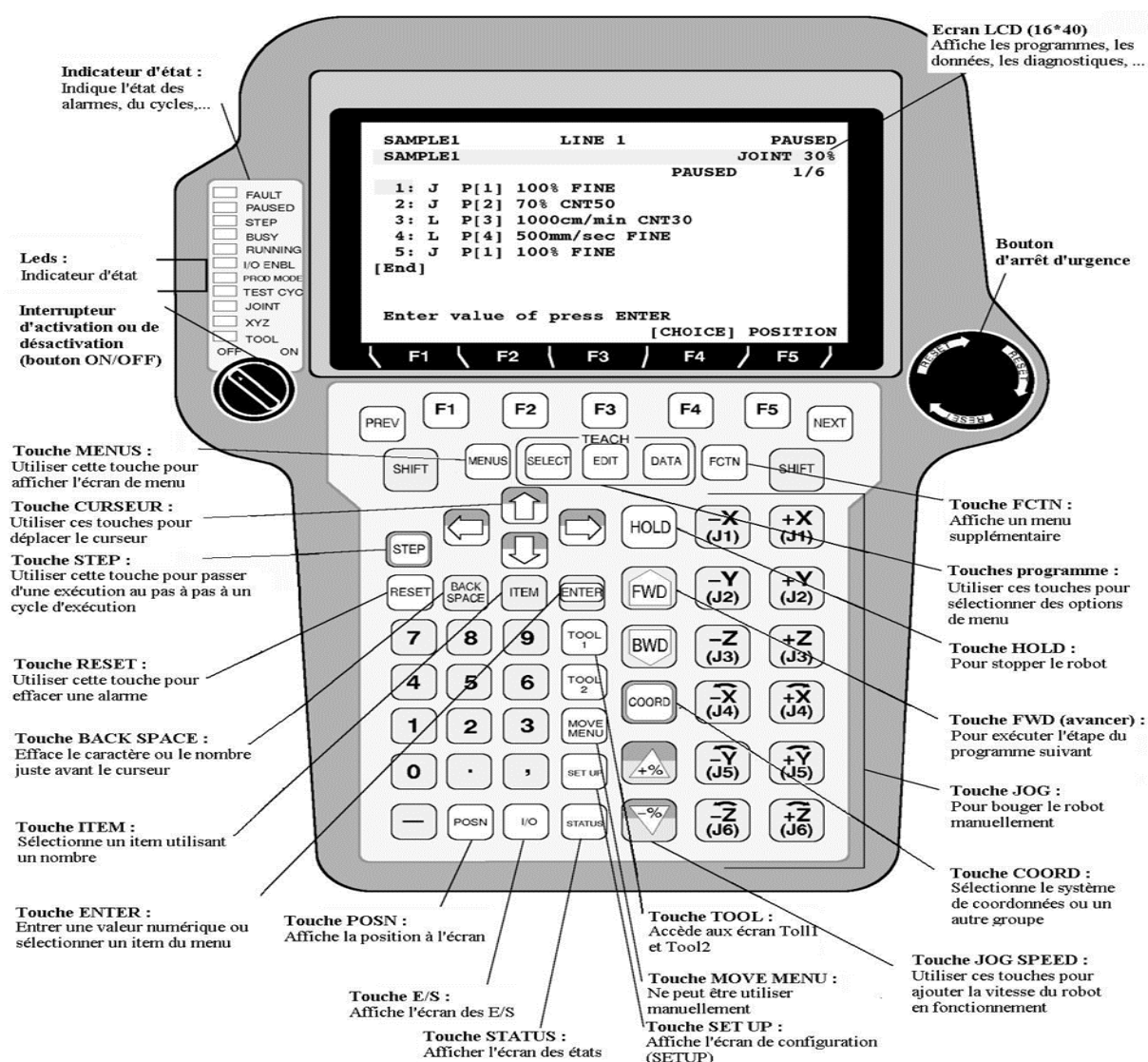


Figure 21 : Description du Teach Pendant.

II.7 Modes de déplacement du Robot :

Les robots industriels, en particulier les bras manipulateurs, disposent de plusieurs modes de déplacement qui permettent de contrôler leurs mouvements avec précision en fonction des tâches à effectuer. Voici une description des modes de déplacement du robot de Guedila présenté comme suits :

II.7.1 En JOINT : axe par axe :

Le mode **JOINT**, également appelé **mode axe par axe**, est un mode de déplacement des robots industriels dans lequel chaque axe ou articulation du robot se déplace indépendamment des autres pour atteindre une position finale [4].

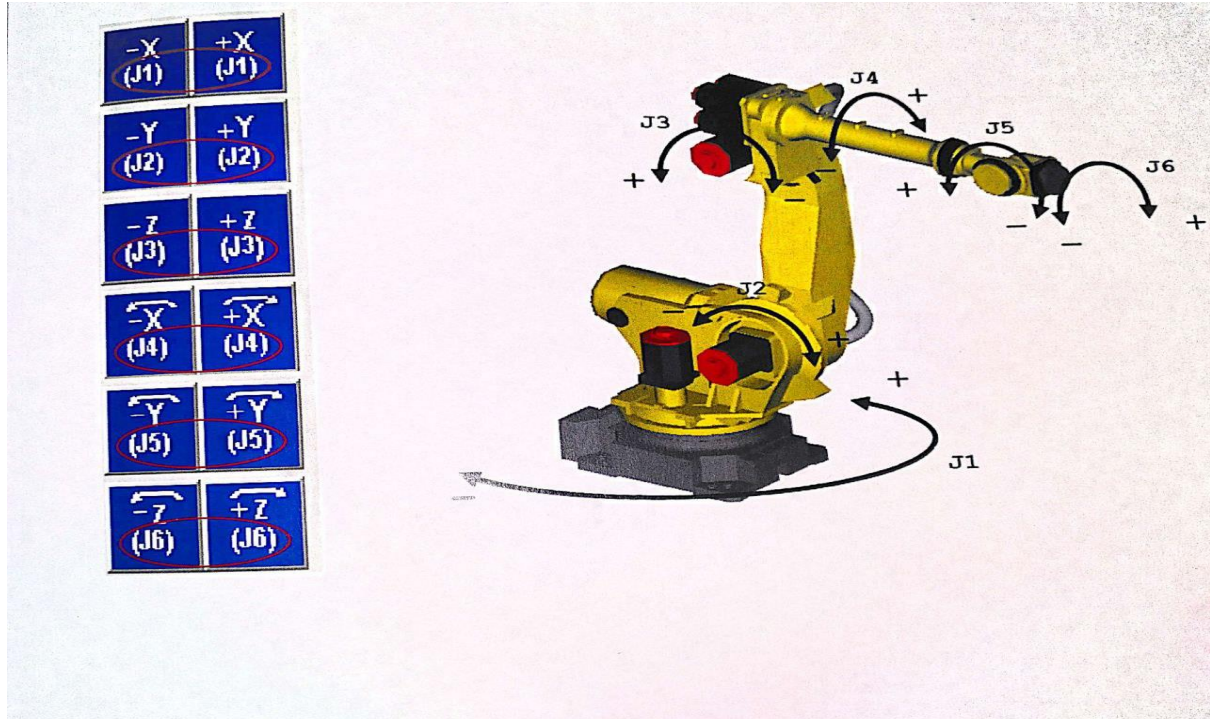


Figure 22 : Déplacement du Robot En JOINT.

II.7.2 Dans un repère universel : WORLD

Le mode **WORLD** est un mode de déplacement des robots industriels dans lequel les mouvements sont définis par rapport à un repère global fixe appelé **repère universel** ou **repère mondial** (World Frame). Ce repère est généralement défini par les axes X, Y, et Z d'un système de coordonnées cartésiennes fixé dans l'espace de travail global [4].

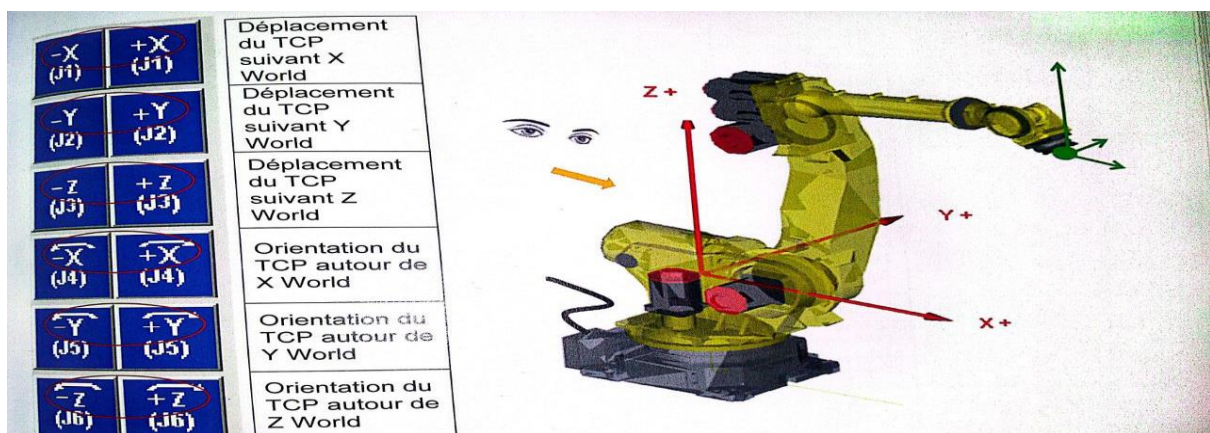


Figure 23 : Déplacement du Robot En WORLD.

II.7.3 Dans un repère outil : TOOL

Le **repère outil** (souvent appelé **Tool Frame** ou repère de l'outil) est un système de coordonnées défini par rapport à l'outil installé à l'extrémité d'un bras robotique. Ce repère est essentiel pour contrôler les mouvements et l'orientation de l'outil dans l'espace, qu'il s'agisse d'une pince, d'une buse de soudure, ou d'une caméra [4].

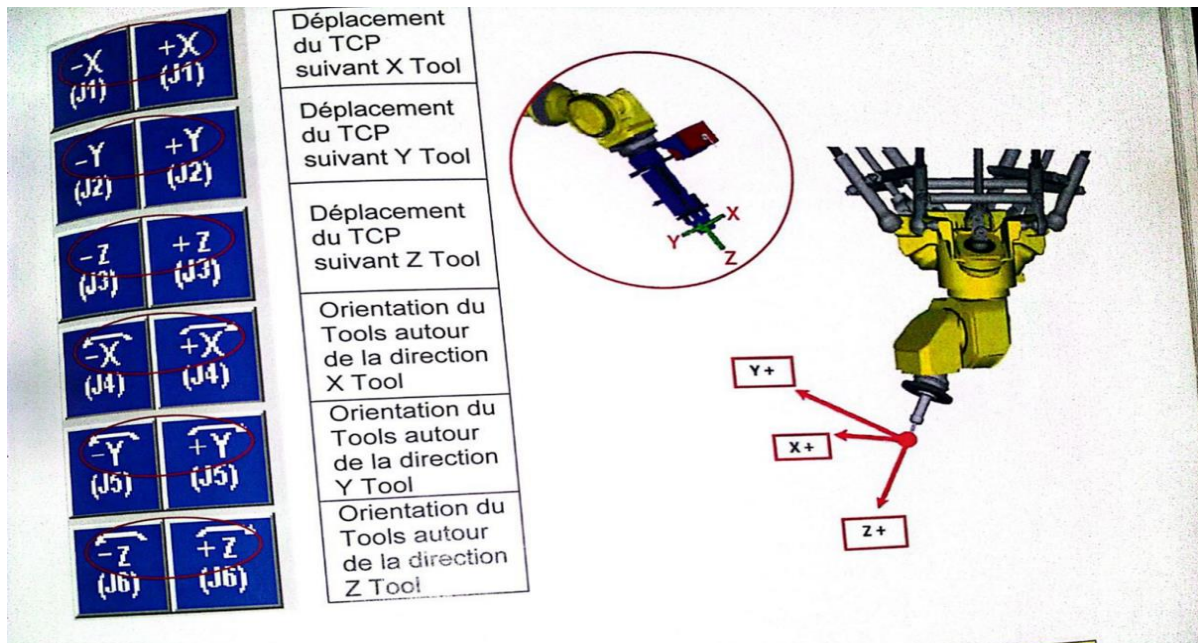


Figure 24 : Déplacement du Robot En TOOL.

II.7.4. Dans un repère outil : USER

User est un repère défini par l'utilisateur en fonction des besoins spécifiques d'une application ou d'un environnement de travail. Ce repère est souvent utilisé pour simplifier la programmation et permettre de travailler plus facilement avec des coordonnées relatives à un point ou un objet spécifique dans l'espace de travail [4].

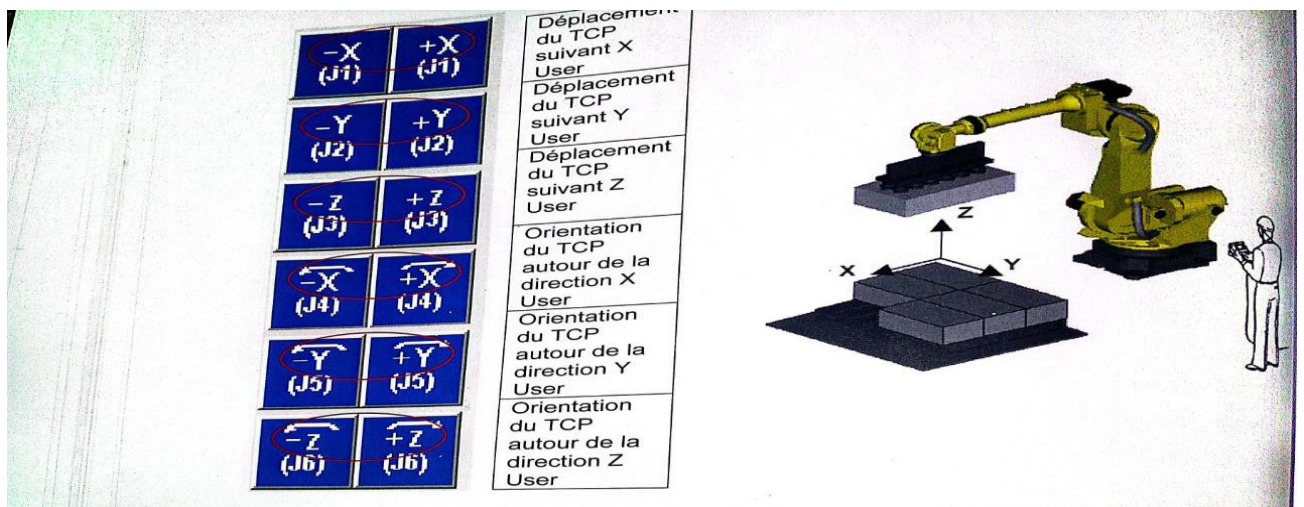


Figure 25 : Déplacement du Robot En USER.

II.7.4. Dans un repère outil : JOG FRAME

Le **Jog Frame** est un mode ou un repère utilisé dans le contrôle de robots industriels pour permettre un déplacement manuel (jogging) intuitif du robot en fonction d'un système de coordonnées spécifique. Ce repère est généralement défini pour simplifier la manipulation du robot lors des réglages ou des interventions manuelles [4].

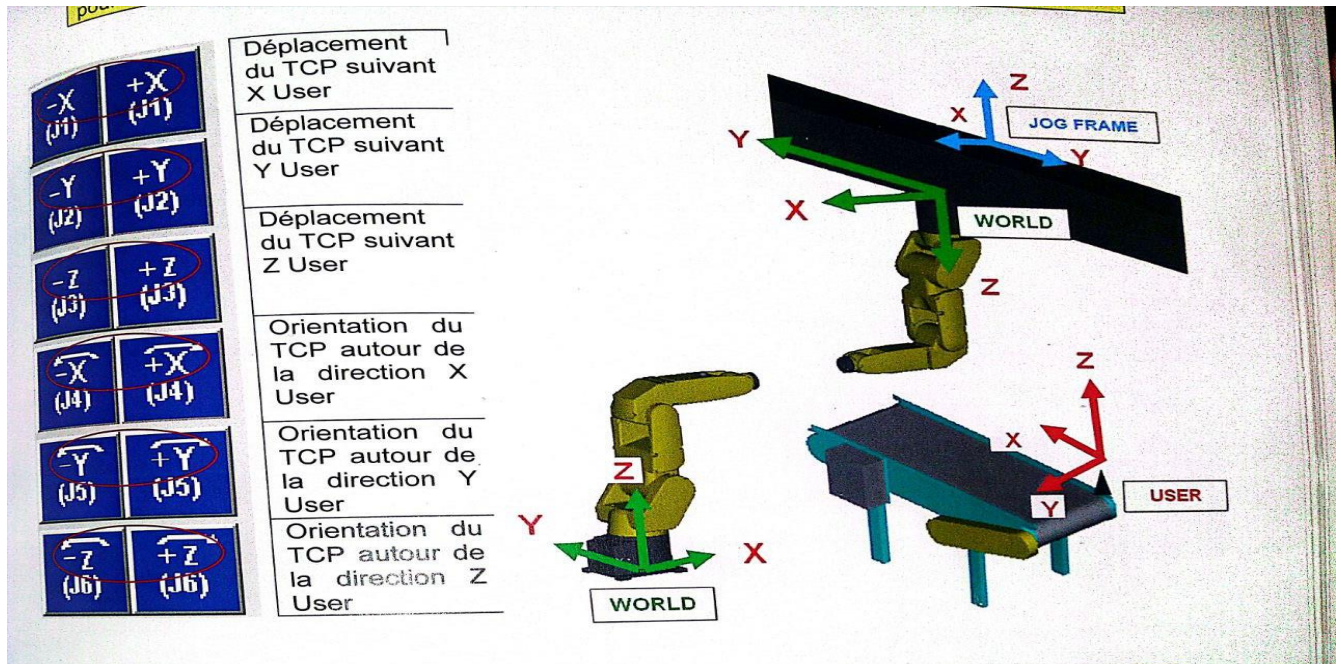
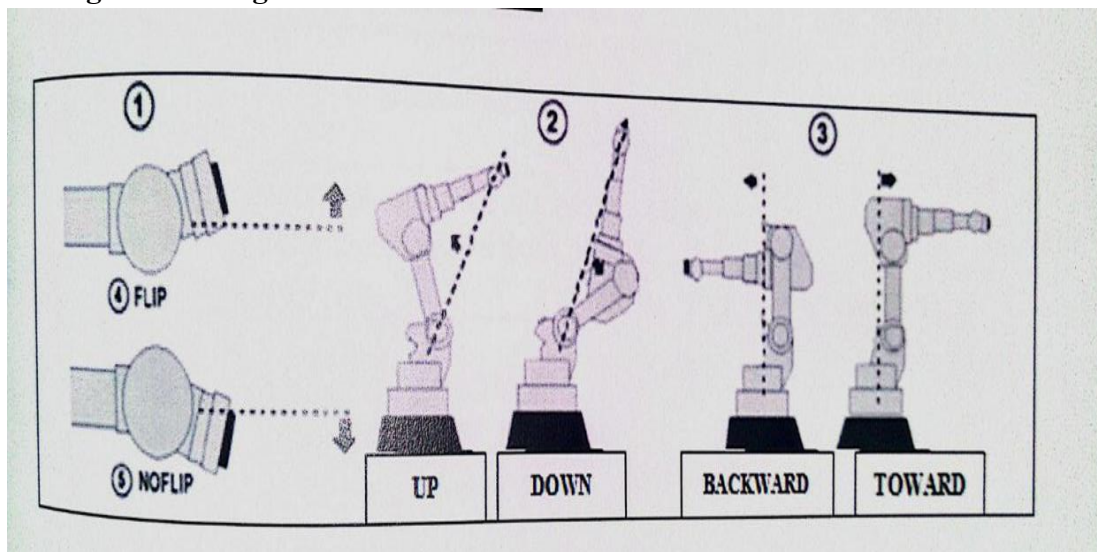


Figure 26 : Déplacement du Robot En JOG FRAME.

II.8 La vitesse de déplacement :

La vitesse de déplacement correspond à un pourcentage de la vitesse maximum à laquelle il est permis de faire avancer le robot. Cette valeur est une valeur entière comprise entre 1 et 100%.

II.9 Configuration Angulaire d'un robot :



F : FLIP	Poignet « cassé » (positionné vers le haut)
N : NOFLIP	Poignet « non cassé » (positionné vers le bas)
U: UP	Avant-bras positionné vers le haut
D: DOWN	Avant-bras positionné vers le bas
T:TOWARD	Bras positionné vers l'avant
B: BACKWARD	Bras positionné vers l'arrière

Où 0, 0,0 représente respectivement le nombre de tour sur les axes J4, J5 et J6 :

-1 : -539° □ -180°

0 : -179° □ 179°

1 : 180° □ 539°

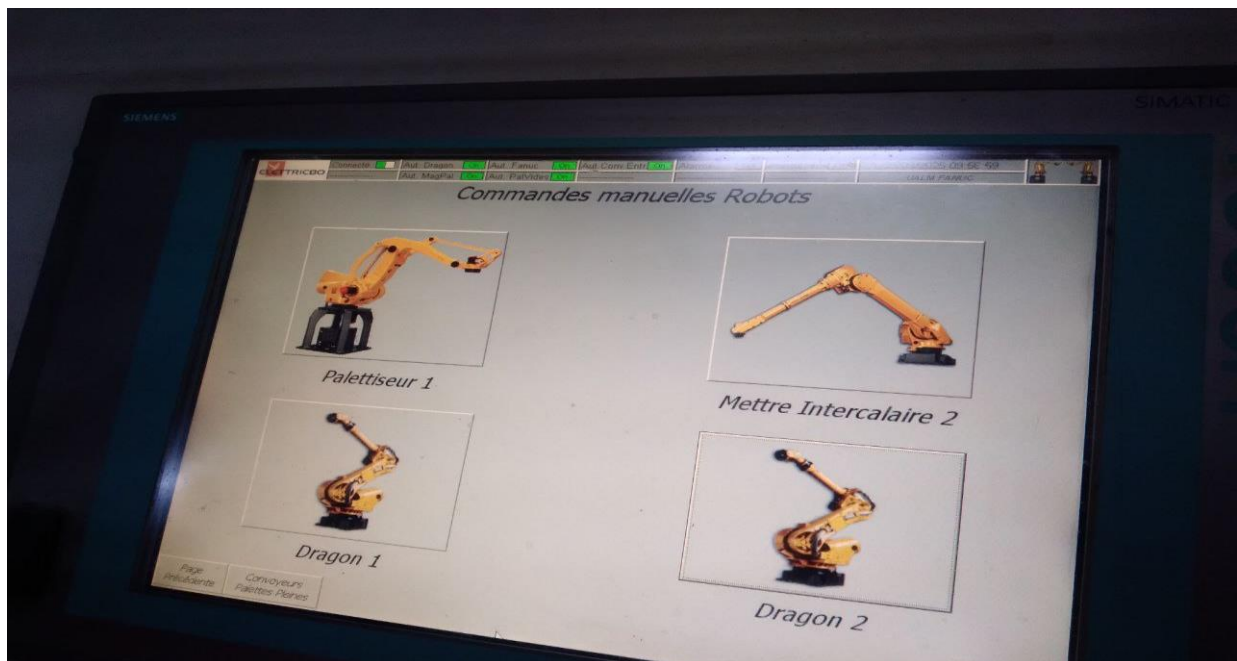


Figure 27 : HMI DE COMMANDES DE ROBOT.

II.10 Conclusion :

Dans ce chapitre, nous avons présenté une vue d'ensemble des robots FANUC, en abordant leur conception, leurs principales caractéristiques techniques, ainsi que les systèmes de sécurité intégrés, notamment le dispositif DCS (Dual Check Safety). Nous avons également détaillé le modèle spécifique FANUC R-2000iC/165F utilisé dans l'usine, en décrivant ses spécifications, ses capacités et son rôle dans les opérations automatisées.

CHAPITRE III : SIMULATION ET RÉSULTATS

CHAPITRE III : SIMULATION ET RÉSULTATS

III .1 Introduction :

Nous avons vu dans le chapitre précédent des généralités sur le domaine de la robotique et plus particulièrement des bras manipulateurs qui sont l'objet de notre mémoire, dans ce dernier chapitre, nous allons vous présenter comment nous utilisons le logiciel ROBOGUIDE avec TIA Portal pour programmer le bras manipulateur R-2000iC/165F.

III .2 Simulaion de Roboguide :

III .2.1 Création d'un environnement de travail « Works Cell » :

Nous vous présenterons le travail sous la forme d'un tutoriel qui vous guidera pas à pas pour créer un environnement virtuel qui simule le contrôle à distance d'un robot R- 2000iC/165F.

III.2.2 Ouverture le fichier :

Pour commencer la simulation dans RoboGuide, procédez à l'ouverture du fichier de projet correspondant au robot R-2000iC/165F :

III.2.2.1 Ouvrez le logiciel Roboguide :

Pour débiter, lancez le logiciel RoboGuide. Une fois ouvert, vous accédez à l'interface principale qui vous permettra de gérer vos projets de simulation robotique comme présenté dans la figure suivante :

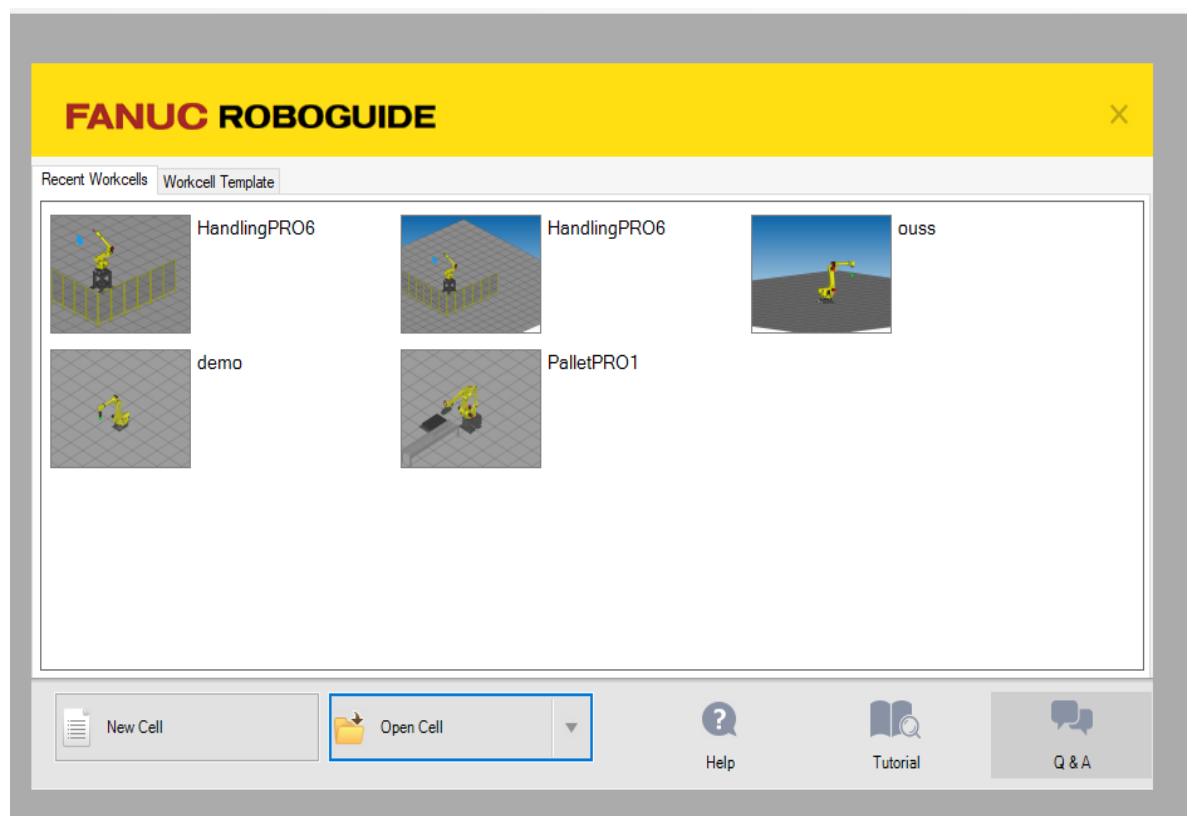


Figure 28 : HandlingPRO6.

III.2.2.2 Sélectionnez « Handling PRO6 » :

Dans l'interface principale de RoboGuide, choisissez le projet ou la cellule de travail intitulée « Handling PRO6 ».

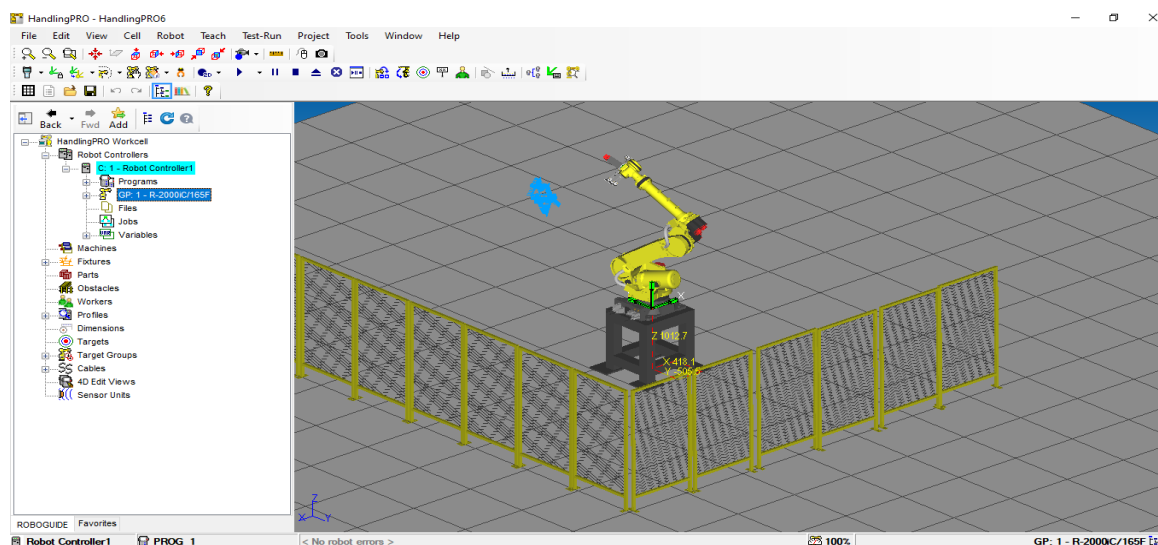


Figure 29 : Regard intérieur [HandlingPRO6]

III.2.2.3 Sélectionnez teach pendant pour écrire le programme :

Accédez à l'interface **Teach Pendant**, l'outil virtuel de programmation, afin de créer, modifier ou visualiser le programme de contrôle du robot.

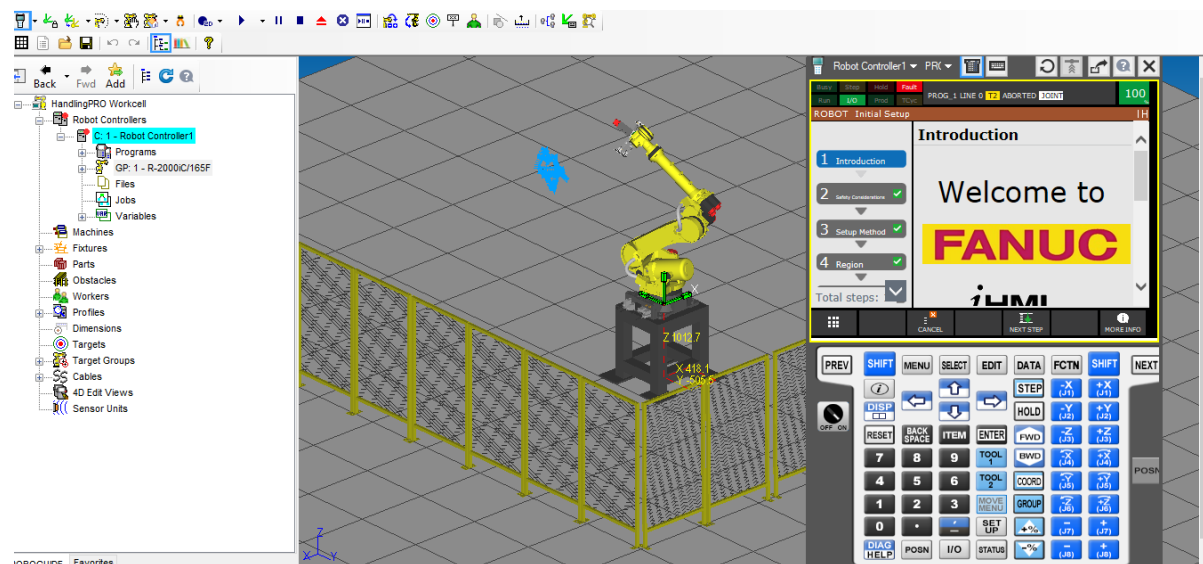


Figure 30 : Teach Pendant.

III.2.2.4 Afficher le programme :

Visualisez le programme en cours d'exécution ou en édition sur l'interface du Teach Pendant afin de pouvoir l'analyser, le modifier ou le valider.



Figure 31 : Programme Manuel

Programme Manuel :

- ✓ Commande Manuel.
- ✓ PLC SIEMENS.
- ✓ Ligne [5] : R [32]= GI [6]>>> :(Sélection du mode manuel).
- ✓ Ligne [7] : LBL [1].
- ✓ Ligne [8.9] : R [32]= GI [7]>>> :(Déterminer la vitesse de déplacement).
- ✓ Ligne [10] : PR [1]= JPOS>>> :(Déterminer les coordonnées en JOINT).
- ✓ Ligne [11.....22] : SELECT R [32]=1, JMP LBL [101]
=12, JMP LBL [112].
- ✓ Ligne [25] : JMP LBL [99].
- ✓ Ligne [26...61] :
- ✓ Exemple ligne [26..28] : LBL [101].
PR [1,1]= PR [1,1] + 40.
JMP LBL [99]. >>> :(Déterminez le premier axe et
Ordonnez-lui de se déplacer avec la valeur 40 dans le sens positif).
- ✓ . Exemple ligne [47..49] : LBL [108]. PR [1,4]= PR [1,4] -3. JMP LBL [99]. >>> :(
Déterminez le quatrième axe et ordonnez-lui de se déplacer avec la valeur -3 dans le sens
négatif)
- ✓ Ligne [70] : LBL [99].
- ✓ Ligne [71..73] : SKIP CONDITION DI [41 :OFF]= OFF. L PR [1] 600mm/sec CNT0 .
Skip, LBL [50]>>> :(Si cette condition est remplie, le paramètre du robot passe de Joint à
Linéaire et change avec lui position registre du robot Et dirigé vers LBL [50]).
- ✓ Ligne [74.76] : LBL [50].
IF DI [41 : OFF] =ON, JMP LBL [1]>>>:(Si cette condition est remplie, sautez
LBL[1]).
- ✓ Ligne [76] : END.

III.2.3 Les groupes Input/Output :

Les entrées-sorties groupées permettent d'accéder aux données de plusieurs signaux d'entrées digitales ou de sorties digitales à la fois. Les instructions d'entrées-sorties groupées permettent alors de contrôler ces signaux sous forme de valeur binaire codée en décimal ou en hexadécimal.

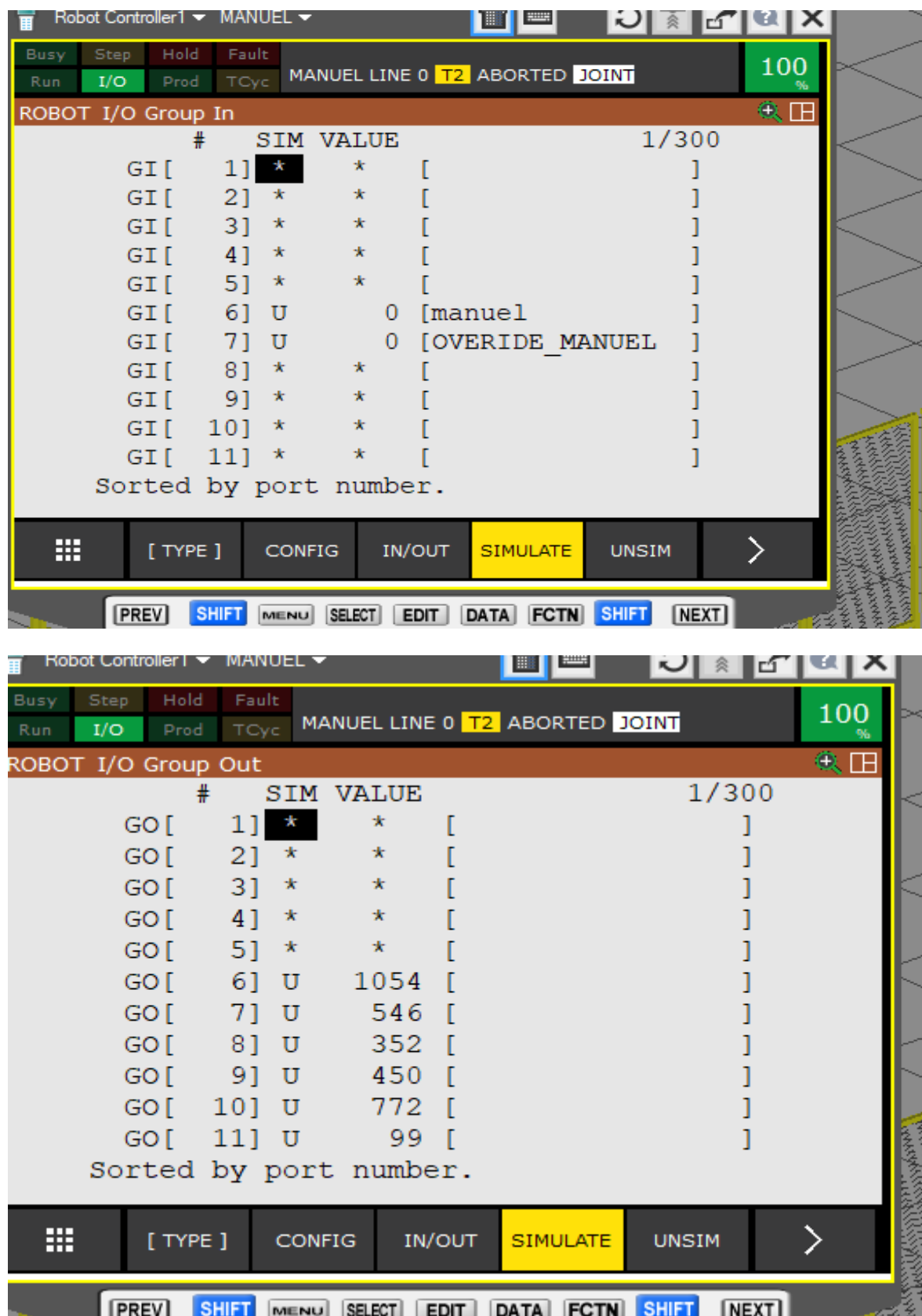


Figure 32 : G [I/O].

III.2.4. Position_Robot :

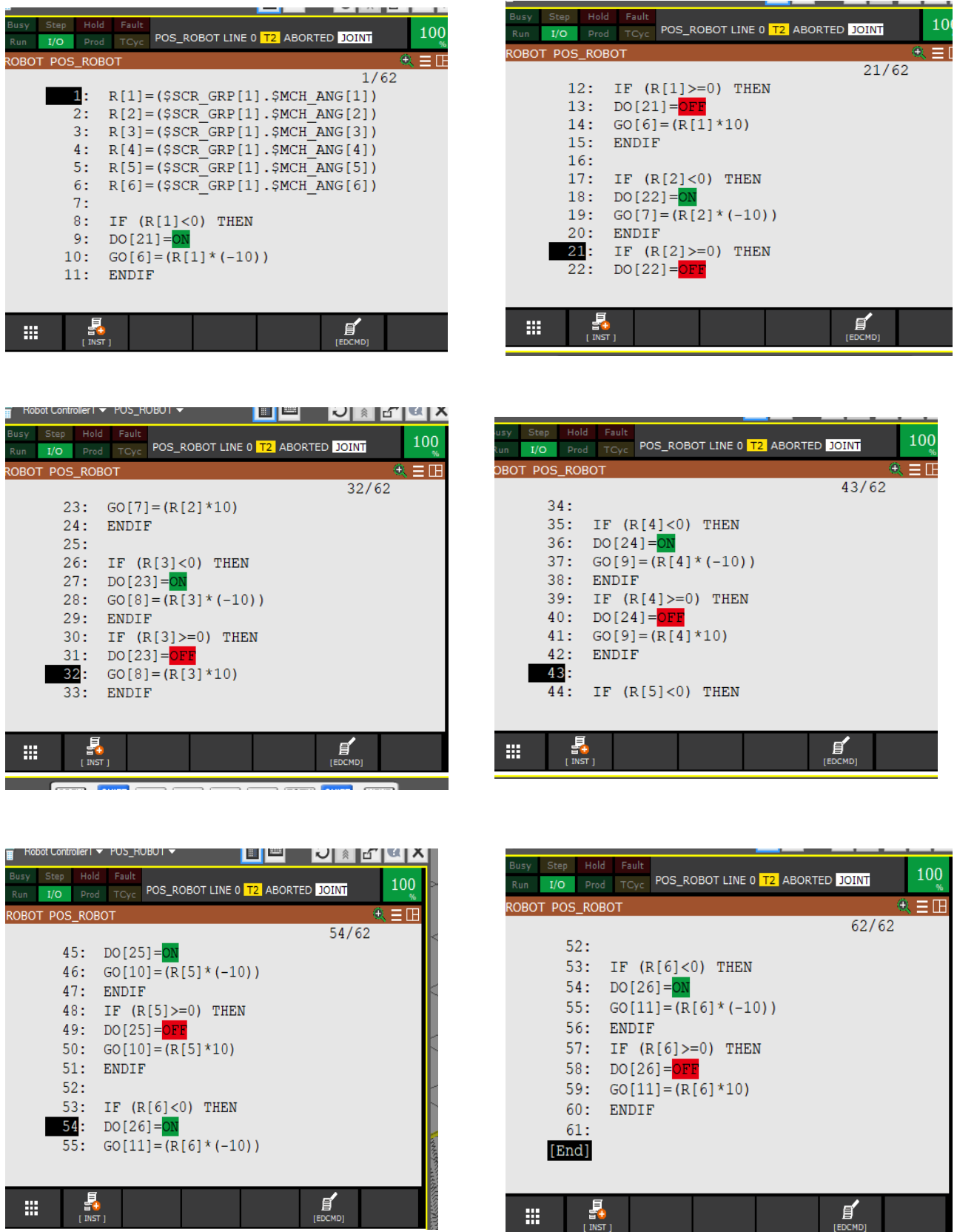


Figure 33 : Position_Robot

III.2.5 Correspondance des Signaux :



Figure 34 : UOP (I/O).

III.2.5.1 Les Entrée UOP :

~ Entrée "IMSTP UI [1] (toujours activée) :

Le signal d'arrêt immédiat spécifie un arrêt d'urgence logiciel En fonctionnement UOP, le signal "IMSTP doit être à ON. Lorsque le signal passe à OFF, les évènements suivant se produisent:

- Les opérations robot s'arrêtent immédiatement ;
- Une alarme est générée et la puissance moteur est coupée ;
- L'exécution du programme est stoppée.

~ **Entrée "HOLD UI [2] (toujours activée) :**

Le signal d'arrêt temporaire spécifie un arrêt temporaire depuis un équipement externe. En fonctionnement UOP, l'entrée "HOLD doit être à ON. Lorsque le signal passe à OFF, les actions suivantes se produisent:

- Le robot décélère jusqu'à l'arrêt, puis l'exécution du programme s'interrompt ;
- Si la fonction Brake on Hold est ENABLE (voir écran de paramétrage des items généraux), le robot s'arrête, une alarme est générée et la puissance retombe.

~ **Entrée "SFSPD UI [3] (toujours activée) :**

Le signal de vitesse de sécurité stoppe temporairement le robot lorsqu'une sécurité porte est ouverte. En fonctionnement UOP, l'entrée "SFSPD doit être à ON Lorsque le signal passe à OFF, les actions suivantes se produisent

- Le robot en cours de mouvement décélère, s'arrête et l'exécution du programme est arrêtée également ;
- La vitesse est réduite à la valeur spécifiée dans \$SCR.SFENCEOVRD.

~ **Signal d'entrée Enable, ENBL :**

Lorsque le signal ENBL est à OFF, le système inhibe les mouvements manuels et l'activation d'un programme incluant des instructions de mouvement. Un programme en cours d'exécution est arrêté si ce signal passe à OFF.

III.2.5.2 Les Sorties UOP :

~ **Sortie SYSRDY**

SYSRDY est présent lorsque la puissance moteur est à ON. Ce signal place le robot en état de mouvement. Dans cet état, les déplacements manuels du robot peuvent être exécutés et un programme incluant des opérations de mouvement peut être démarré. Le robot passe dans cet état lorsque les conditions de validation des opérations suivantes sont satisfaites.

~ **Sortie TPENBL**

TPENBL est présent lorsque l'interrupteur d'activation sur le Teach Pendant est mis à ON.

III.3. Configuration matérielle du portail TIA :

Pour regarder les appareils avec les adresses :

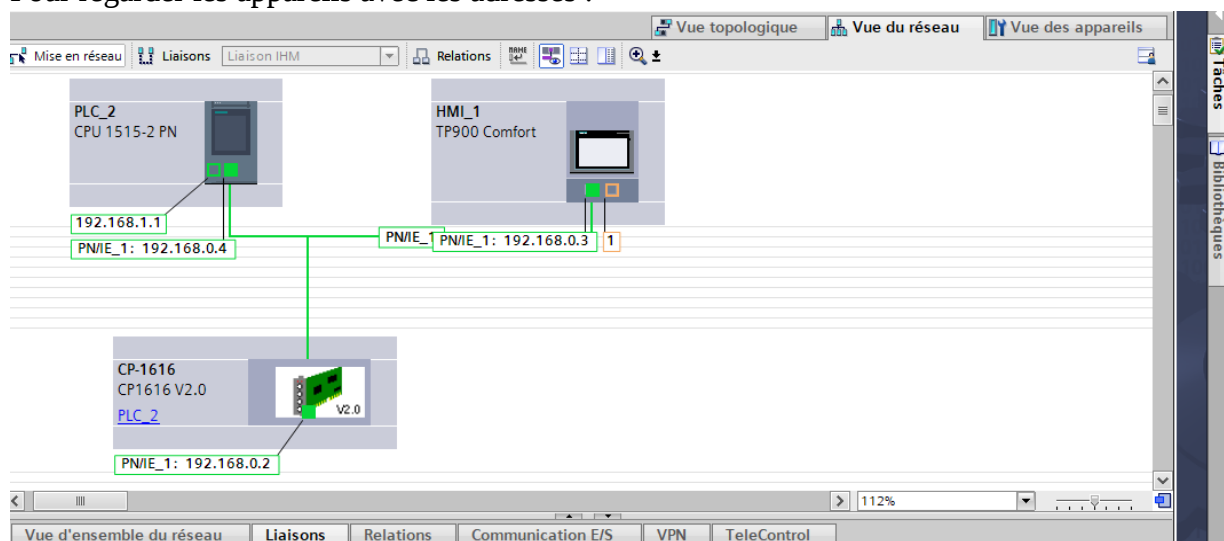


Figure 35 : mise en réseau.

Vue d'ensemble du réseau	Liaisons	Relations	Communication E/S	VPN	TeleControl	
Appareil	Type	Adresse dans le ...	Sous-réseau	Réseau maître IO	Numéro d'appareil	Commentaire
▼ HMI_1	TP900 Comfort					
HMI_RT_1	TP900 Comfort					
▶ HMI_1.IE_CP_1	Interface PROFINET					
HMI_1.MPI/DP_CP_1	Interface MPI/DP	1	non connecté			
▼ Appareil GSD_1	Appareil GSD					
▶ CP-1616	CP1616 V2.0					
▼ Station S71500/ET200MP_1	Station S71500/ET200...					
▼ PLC_2	CPU 1515-2 PN					
▶ Interface PROFINET_1	Interface PROFINET	192.168.0.4	PN/IE_1	PROFINET IO-System		
▶ Interface PROFINET_2	Interface PROFINET	192.168.1.1	non connecté			

Figure 36 : vue d'ensemble du réseau.

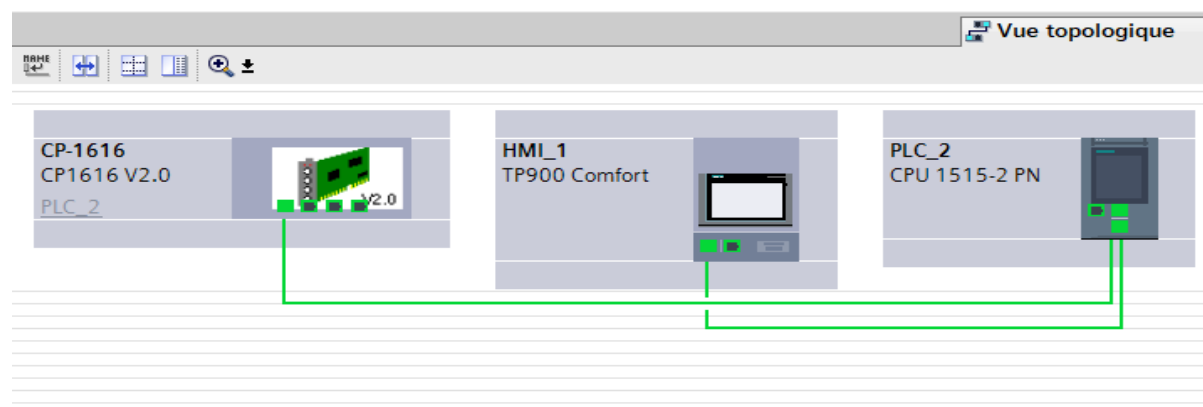


Figure 37 : Vue topologie.

Vue d'ensemble de la topologie		Comparaison topologique					
Appareil / Port	Empla...	Station partenai...	Appareil parten...	Interface parten...	Port partenaire	Données de câ...	
▼ HMI_1							
HMI_RT_1	1						
▼ HMI_1.IE_CP_1	5						
▼ Interface PROFINET_1	5 X1						
Port_1	5 X1 P...	Station S71500...	PLC_2	Interface PROFI...	Port_2	< 100 m (0,6 µs)	
Port_2	5 X1 P...				Partenaire quel...	---	
▼ Appareil GSD_1							
▼ CP-1616	0						
▼ PN-IO	0 X1						
Port 1 - RJ45	0 X1 P1	Station S71500...	PLC_2	Interface PROFI...	Port_1	< 100 m (0,6 µs)	
Port 2 - RJ45	0 X1 P2				Partenaire quel...	---	
Port 3 - RJ45	0 X1 P3				Partenaire quel...	---	
Port 4 - RJ45	0 X1 P4				Partenaire quel...	---	
▼ Station S71500/ET200MP_1							
▼ PLC_2	1						
▼ Interface PROFINET_1	1 X1						
Port_1	1 X1 P1	Appareil GSD_1	CP-1616	PN-IO	Port 1 - RJ45	< 100 m (0,6 µs)	
Port_2	1 X1 P2	HMI_1	HMI_1.IE_CP_1	Interface PROFI...	Port_1	< 100 m (0,6 µs)	
▼ Interface PROFINET_2	1 X2						
Port_1	1 X2 P1				Partenaire quel...	---	

Figure 38 : Vue d'ensemble la topologie.

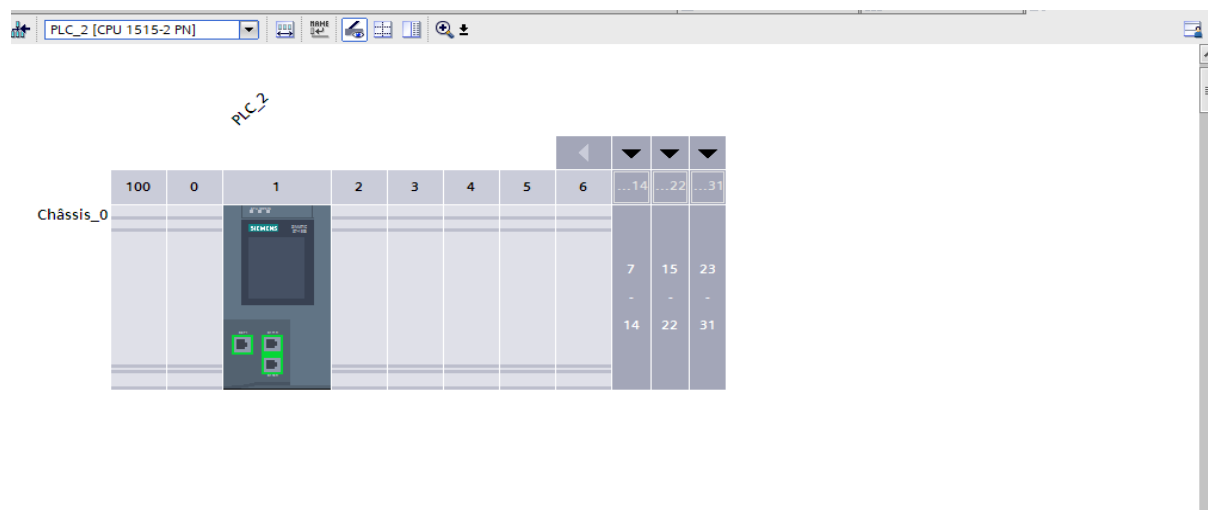


Figure 39 : PLC_2.

Vue d'ensemble des appareils									
Module	Châssis	Empla...	Adresse I	Adresse...	Type	N° d'article	Firmware	Commentaire	
▼ PLC_2	0	100							
Interface PROFINET_1	0	0							
Interface PROFINET_2	0	1			CPU 1515-2 PN	6ES7 515-2AM01-0AB0	V2.6		
	0	2			Interface PROFINET				
	0	3							
	0	4							
	0	5							
	0	6							
	0	7							
	0	8							
	0	9							
	0	10							
	0	11							
	0	12							
	0	13							
	0	14							
	0	15							
	0	16							
	0	17							
	0	18							
	0	19							
	0	20							
	0	21							
	0	22							
	0	23							
	0	24							

Figure 40 : Vue d'ensemble des appareils(PLC).

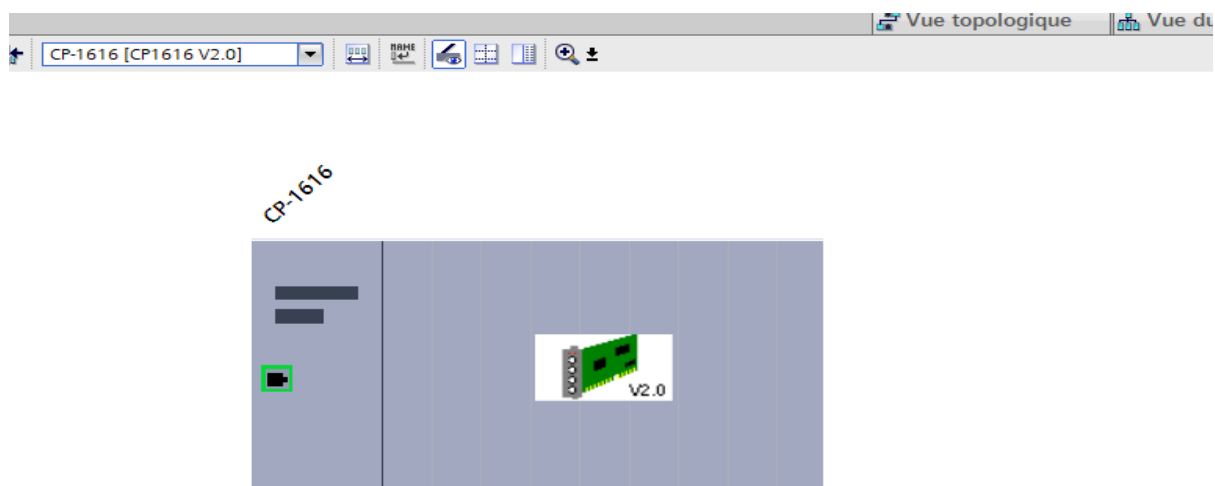


Figure 41 : Carte CP-1616.

Vue d'ensemble des appareils									
Module	Châssis	Empla...	Adresse I	Adresse...	Type	N° d'article	Firmware	Commentaire	
▼ CP-1616	0	0			CP1616 V2.0	6GK1 161-6AA00	V2.0		
▶ PN-IO	0	0 X1			CP-1616				
64 octets_1	0	1	0...63	0...63	64 octets		1.0		
	0	2							
	0	3							
	0	4							
	0	5							
	0	6							
	0	7							
	0	8							
	0	9							
	0	10							
	0	11							
	0	12							
	0	13							
	0	14							
	0	15							

Figure 42 : Vue d'ensemble des appareils(CP_1616).

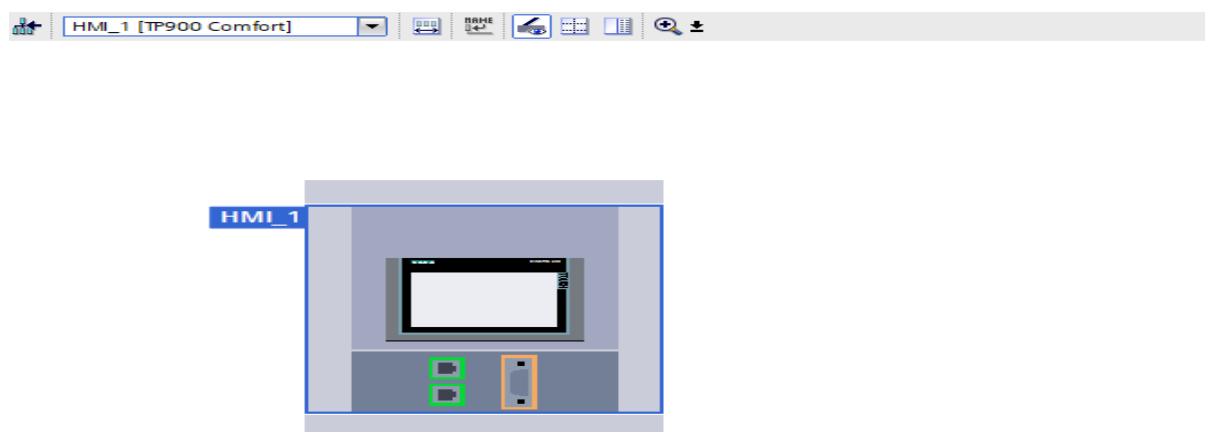


Figure 43 : HMI.

Vue d'ensemble des appareils						
Module	Index	Type	N° d'article	Version de f..	Commentaire	
HMI_RT_1	1	TP900 Comfort	6AV2 124-0JC01-0AX0	15.1.0.0		
	2					
	3					
	4					
▼ HMI_1.IE_CP_1	5	Interface PROFINET		15.0.1.0		
▶ Interface PROFINET_1	5 X1	Interface PROFINET				
	6					
▼ HMI_1.MPI/DP_CP_1	7 X2	Interface MPI/DP		15.0.1.0		
	7 1					
	8					
	9					
	10					
	11					
	12					
	13					

Figure 44 : Vue d'ensemble des appareils (HMI).

III.4 Création des variables de système :

Dans tout programme il faut définir la liste des variables qui vont être utilisées lors de la Programmation pour cela le tableau des variables est créé pour l'insérer des variables du système. L'utilisation des noms appropriés rend le programme plus compréhensible et plus facile à manipuler.

Les figures suivantes représentent le tableau des variables utilisés dans notre programme :

Table de variables standard									
	Nom	Type de données	Adresse	Réma...	Acces...	Ecritu...	Visibl...	Surveilla...	Commentaire
1	M100.0	Bool	%M100.0		☒	☒	☒		MANUEL CYCLE
2	M100.1	Bool	%M100.1		☒	☒	☒		ENABLE_COMMANDE_MANUEL
3	MB45	Byte	%MB45		☒	☒	☒		GI_6
4	DI41	Bool	%M141.0		☒	☒	☒		ACTIVE_MACRO_FANUC
5	Tag_1	Bool	%M100.3		☒	☒	☒		
6	MB46	Byte	%MB46		☒	☒	☒		SPEED_MANUEL
7	Tag_2	DWord	%MD74		☒	☒	☒		
8	Go_J1	Word	%MW48		☒	☒	☒		
9	Go_J2	Word	%MW50		☒	☒	☒		
10	Go_J3	Word	%MW52		☒	☒	☒		
11	Go_J4	Word	%MW54		☒	☒	☒		
12	Go_J5	Word	%MW56		☒	☒	☒		
13	Go_J6	Word	%MW58		☒	☒	☒		
14	DO_FANUC_BIT_NEGATIF_J2	Bool	%M26.1		☒	☒	☒		
15	DO_FANUC_BIT_NEGATIF_J3	Bool	%M26.2		☒	☒	☒		
16	DO_FANUC_BIT_NEGATIF_J4	Bool	%M26.3		☒	☒	☒		
17	DO_FANUC_BIT_NEGATIF_J5	Bool	%M26.4		☒	☒	☒		
18	DO_FANUC_BIT_NEGATIF_J6	Bool	%M26.5		☒	☒	☒		
19	DO_FANUC_BIT_NEGATIF_J1	Bool	%M26.0		☒	☒	☒		
20	<Ajouter>				☒	☒	☒		

Figure 45 : Table de variables PLC_2.

Bloc de données_FANUC_1									
	Nom	Décalage	Valeur de départ	Rémanence	Accessible ...	Écritu...	Visible da...	Valeur de ...	Surveilla...
1	Static								
2	Page_IHM_Manuel	0.0	16#0	☒	☒	☒	☒		
3	CMDMAN_FANUC	2.0	16#0	☒	☒	☒	☒		GI6_FANUC01
4	SPEED_IHM	3.0	16#0A	☒	☒	☒	☒		Speed input HMI
5	POS_J1	4.0	0.0	☒	☒	☒	☒		POSITION AXE " J1 "
6	POS_J2	8.0	0.0	☒	☒	☒	☒		POSITION AXE " J2 "
7	POS_J3	12.0	0.0	☒	☒	☒	☒		POSITION AXE " J3 "
8	POS_J4	16.0	0.0	☒	☒	☒	☒		POSITION AXE " J4 "
9	POS_J5	20.0	0.0	☒	☒	☒	☒		POSITION AXE " J5 "
10	POS_J6	24.0	0.0	☒	☒	☒	☒		POSITION AXE " J6 "
11	Bit_valeur_Negative_J1	28.0	false	☒	☒	☒	☒		
12	Bit_valeur_Negative_J2	28.1	false	☒	☒	☒	☒		
13	Bit_valeur_Negative_J3	28.2	false	☒	☒	☒	☒		
14	Bit_valeur_Negative_J4	28.3	false	☒	☒	☒	☒		
15	Bit_valeur_Negative_J5	28.4	false	☒	☒	☒	☒		
16	Bit_valeur_Negative_J6	28.5	false	☒	☒	☒	☒		

Figure 46 : Bloc de données_FANUC_1.

Variables IHM										
	Nom	Table des variables	Type de don...	Connexion	Nom API	Variable API	Adresse	Cycle d'a...	Archivé	Commentaire source
	Bloc de données_2_CMDMAN...	Table de variables stan.	Byte	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8B2	1 s	☐	GI6_FANUC01
	Bloc de données_FANUC_1_Bit...	Table de variables stan.	Bool	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8X28.0	1 s	☐	
	Bloc de données_FANUC_1_Bit...	Table de variables stan.	Bool	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8X28.1	1 s	☐	
	Bloc de données_FANUC_1_Bit...	Table de variables stan.	Bool	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8X28.2	1 s	☐	
	Bloc de données_FANUC_1_Bit...	Table de variables stan.	Bool	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8X28.3	1 s	☐	
	Bloc de données_FANUC_1_Bit...	Table de variables stan.	Bool	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8X28.4	1 s	☐	
	Bloc de données_FANUC_1_Bit...	Table de variables stan.	Bool	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8X28.5	1 s	☐	
	Bloc de données_FANUC_1_PO...	Table de variables s...	Real	HMI_Liaison...	PLC_2	"Bloc de données_FA...	%D82.D8D4	1 s	☐	POSITION AXE " J1 "
	Bloc de données_FANUC_1_PO...	Table de variables stan.	Real	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8D8	1 s	☐	POSITION AXE " J2 "
	Bloc de données_FANUC_1_PO...	Table de variables stan.	Real	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8D12	1 s	☐	POSITION AXE " J3 "
	Bloc de données_FANUC_1_PO...	Table de variables stan.	Real	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8D16	1 s	☐	POSITION AXE " J4 "
	Bloc de données_FANUC_1_PO...	Table de variables stan.	Real	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8D20	1 s	☐	POSITION AXE " J5 "
	Bloc de données_FANUC_1_PO...	Table de variables stan.	Real	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8D24	1 s	☐	POSITION AXE " J6 "
	Bloc de données_FANUC_1_SP...	Table de variables stan.	Byte	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8B3	1 s	☐	Speed input HMI
	M100.0	Table de variables stan.	Bool	HMI_Liaison_2	PLC_2	"M100.0"	%M100.0	1 s	☐	MANUEL CYCLE
	M100.1	Table de variables stan.	Bool	HMI_Liaison_2	PLC_2	"M100.1"	%M100.1	1 s	☐	ENABLE_COMMANDE_MANUEL
	MB46	Table de variables stan.	Byte	HMI_Liaison_2	PLC_2	MB46	%MB46	1 s	☐	SPEED_MANUEL
	Numéro_vue_variable	Table de variables stan.	Word	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8WD	1 s	☐	
	SPEED_HMI	Table de variables stan.	Byte	HMI_Liaison_2	PLC_2	"Bloc de données_FANUC...	%D82.D8B3	1 s	☐	Speed input HMI
	<ajouter>									

Figure 47: Variable HMI.

III.5 HMI :

Les HMI sont des dispositifs électroniques permettant d'interagir avec des machines, de contrôler un processus ou de saisir des instructions (opérations telles que démarrage, arrêt, changement de mode de fonctionnement...) au niveau de l'utilisateur. Celles-ci doivent être préalablement programmées, en précisant les fonctionnalités dont disposeront les utilisateurs. Il communique ensuite avec l'automate via le port série.

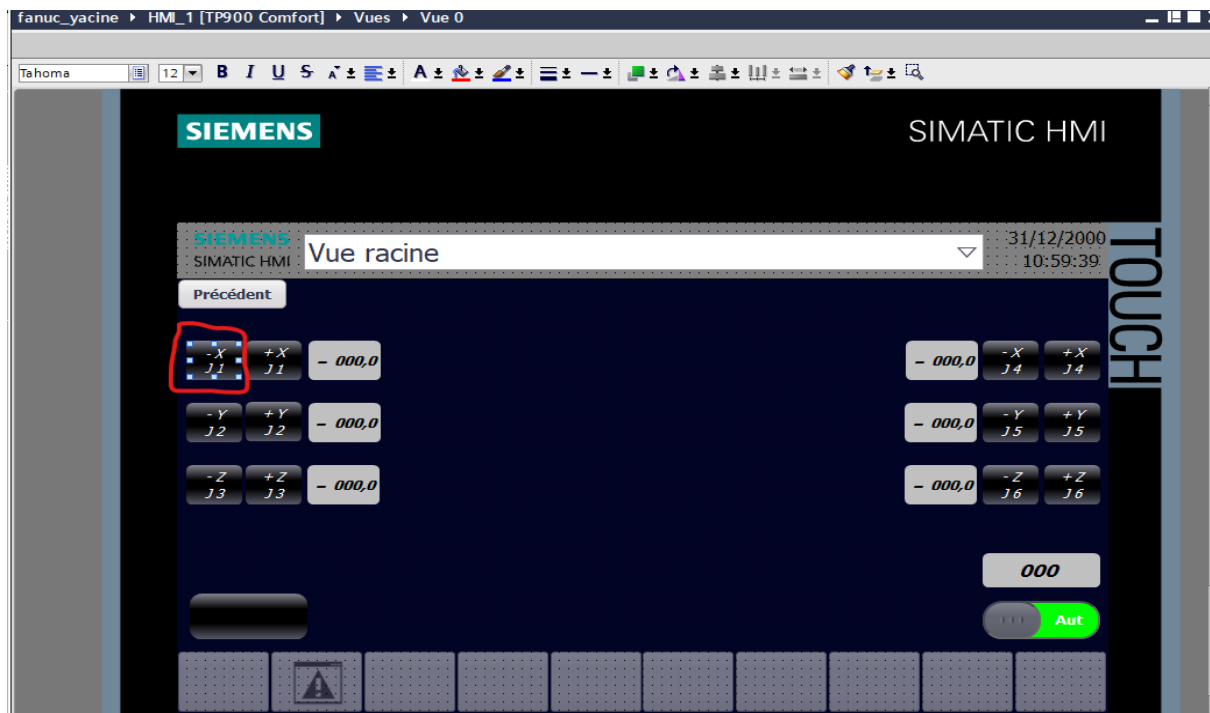
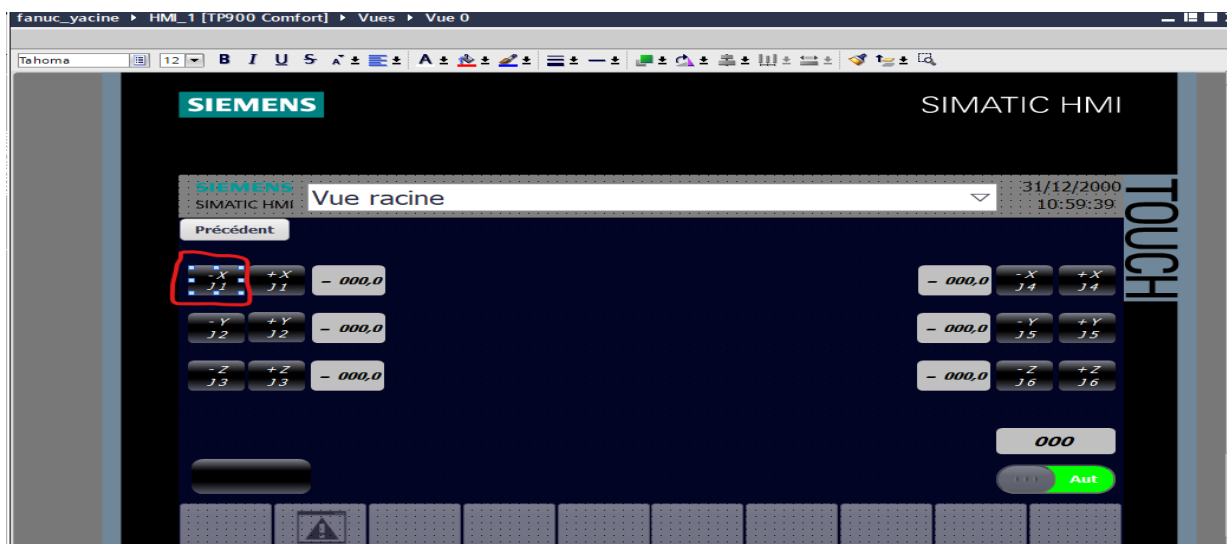
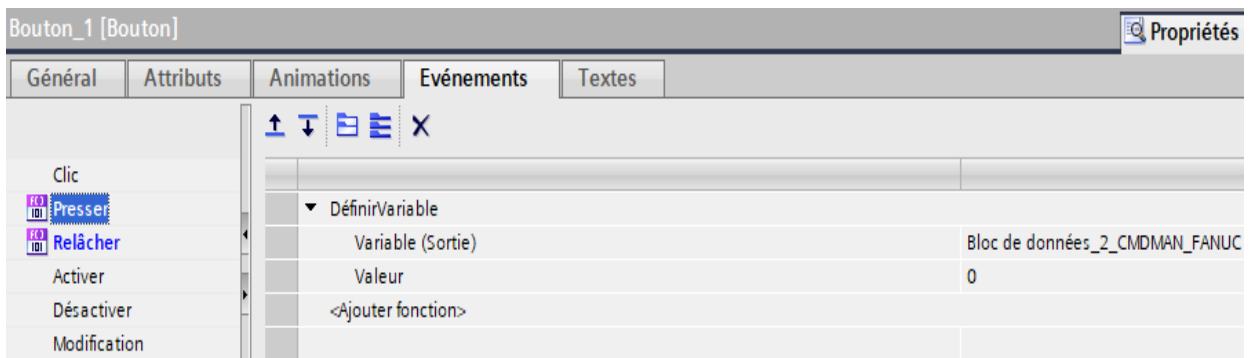
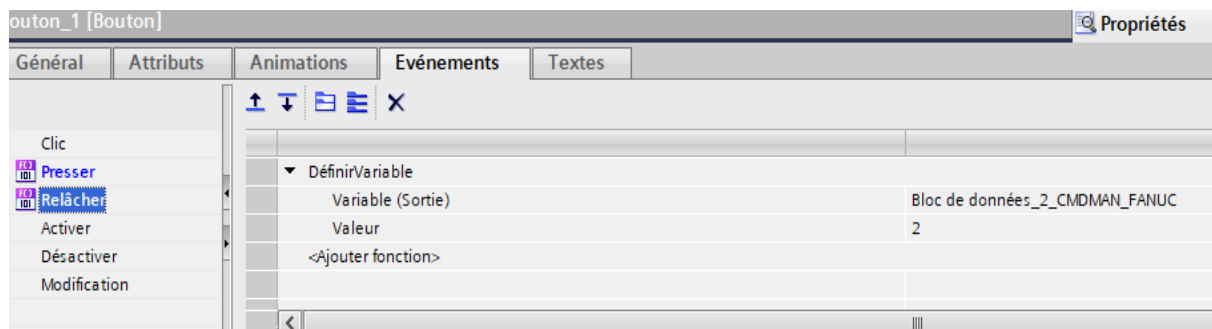


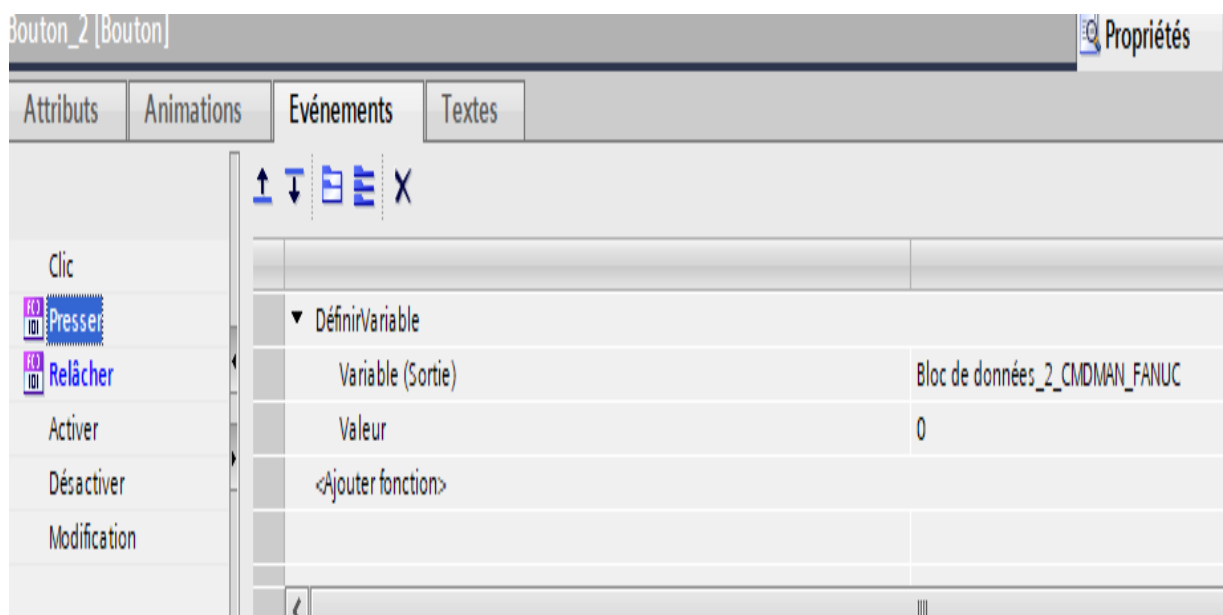
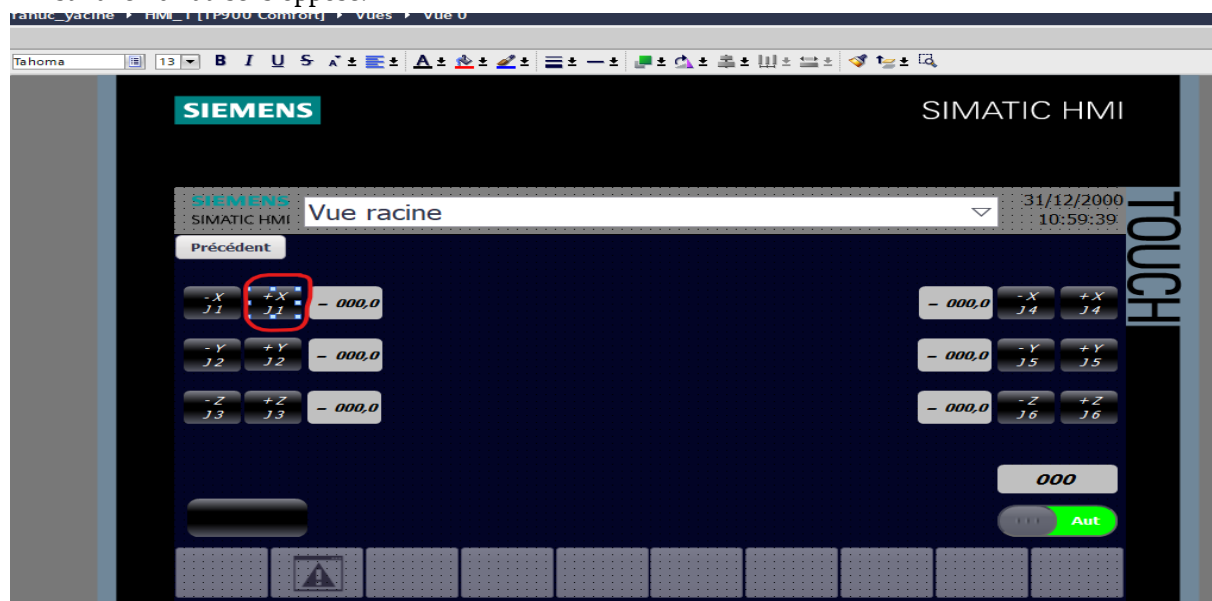
Figure 48 : Page HMI commande robot en manuel

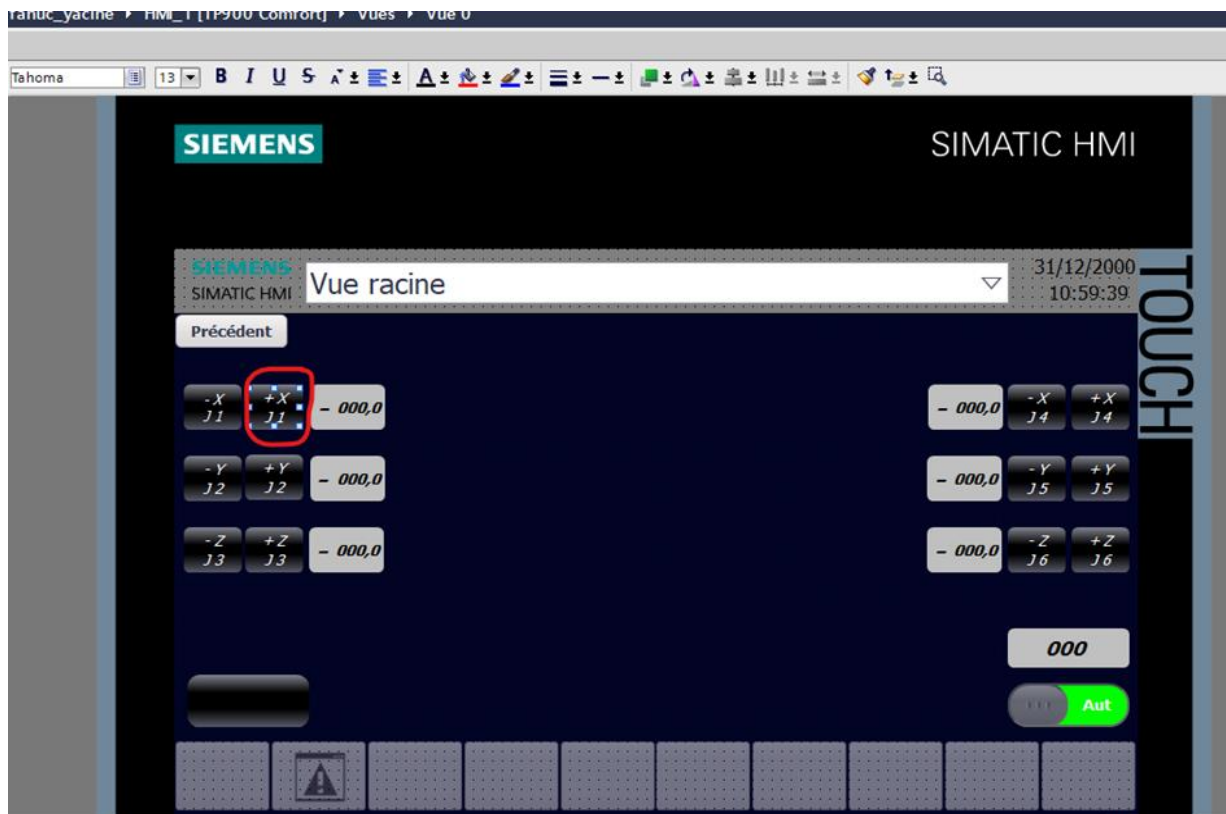
~ **Presser** : Lorsque le bouton -X est enfoncé, il envoie variable à Bloc de données_2_CMDMAN_FANUC=0 ;



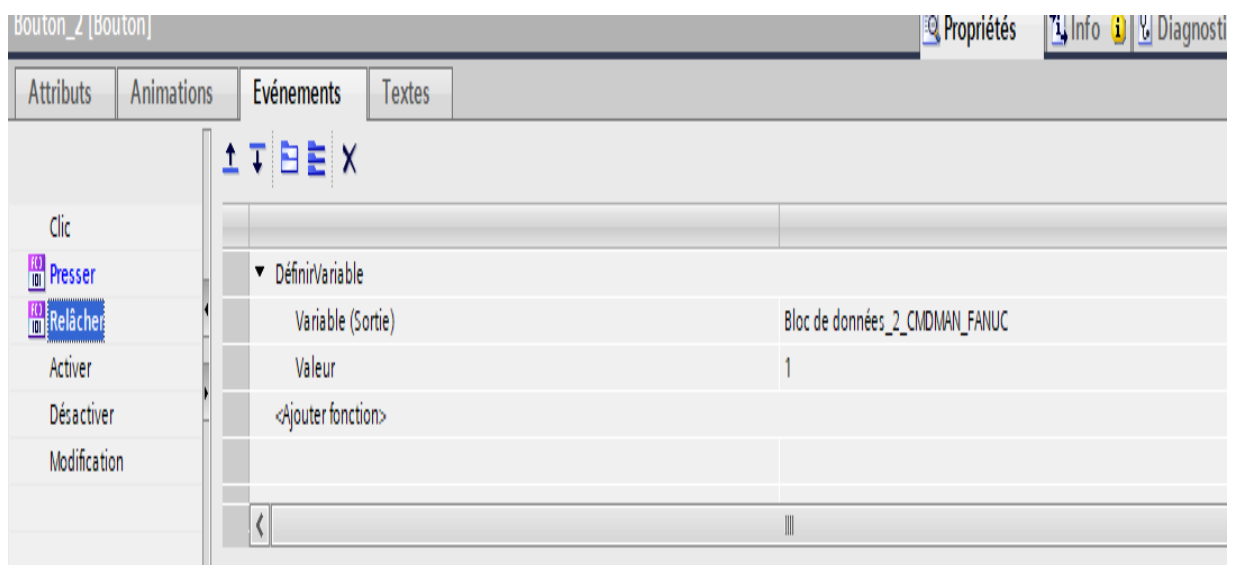


~ **Relâcher** : Lorsque le bouton -X est au repos, nous savons que le bouton marche est enfoncé simultanément, il envoie variable à Bloc de données_2_CMDMAN_FANUC=2, Mouvement du robot sur axe n01 au sens opposé.





~ **Presser** : Lorsque le bouton +X est enfoncé, il envoie variable à Bloc de données_2_CMDMAN_FANUC=0.



~ **Relâcher** : Lorsque le bouton +X est au repos, nous savons que le bouton marche est enfoncé simultanément, il envoie variable à Bloc de données_2_CMDMAN_FANUC=1. Mouvement du robot sur axe n01 au sens opposé.

~ Il en va de même pour le fonctionnement avec le reste des boutons pour déplacer les axes du robot.

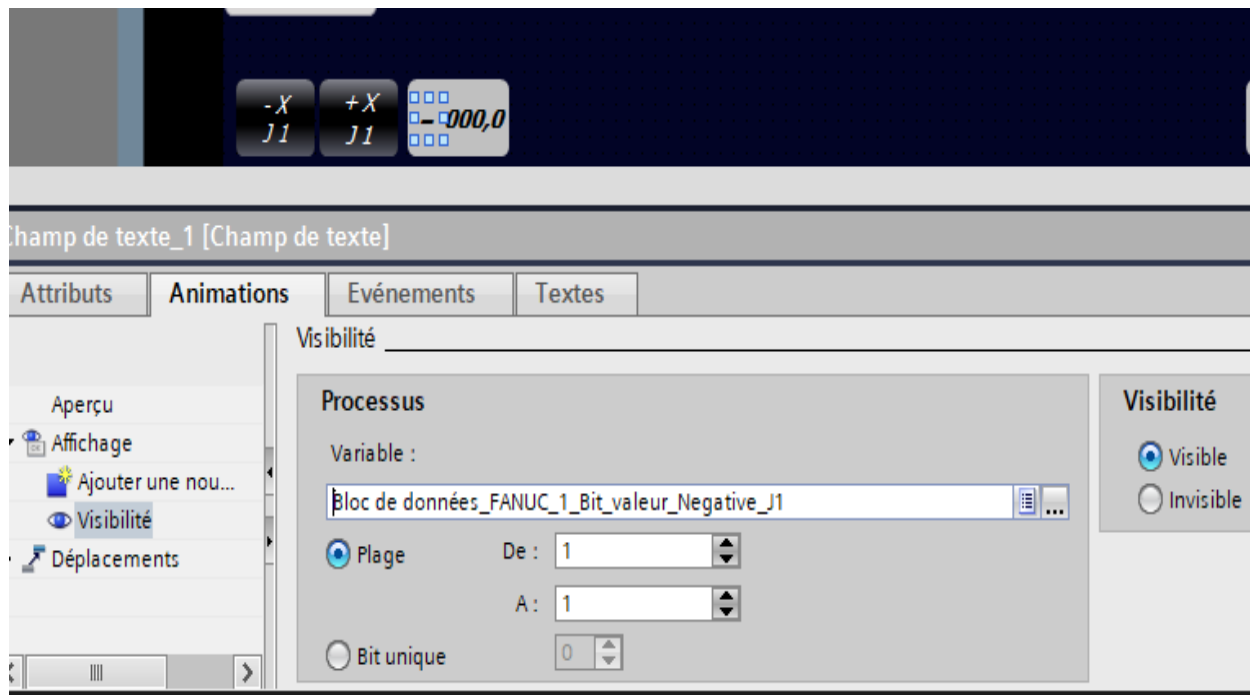


Figure 49 : Les buttons des axes en HMI.

- ~ **Signal (-)** : Il disparaît et apparaît selon les coordonnées dans lesquelles les bras du robot se déplacent, positivement ou négativement, afin de :

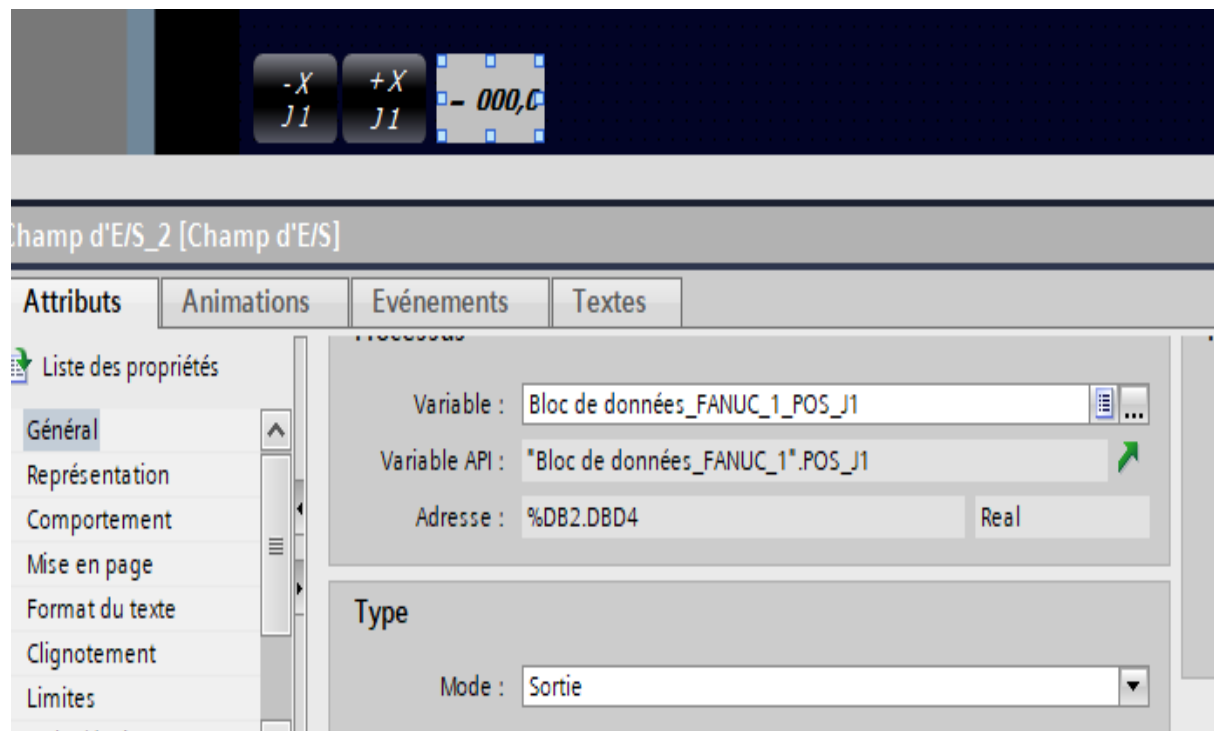


Figure 50 : Les Coordonnées des axes en HMI

- ~ **000.0** : Ces nombres représentent les coordonnées avec lesquelles les bras du robot se déplacent dans l'espace.

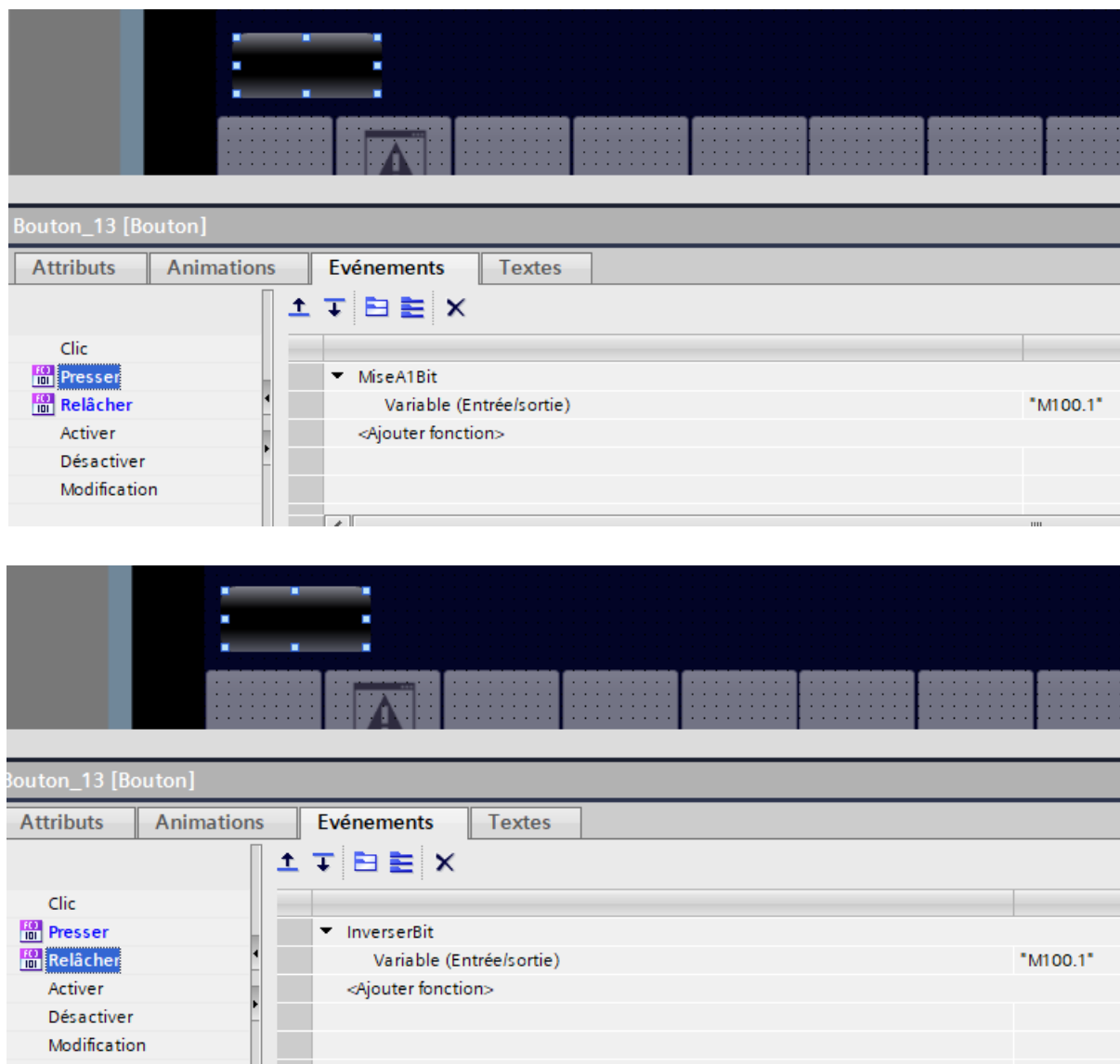


Figure 51: Le button marche en HMI.

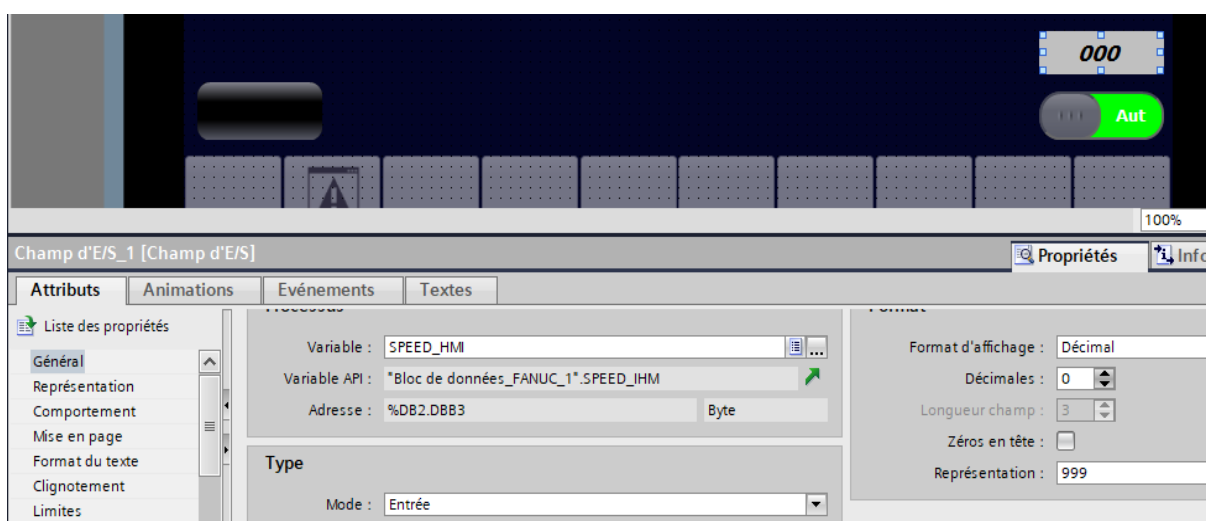


Figure 52 : Écran pour afficher la vitesse du robot.

- ~ Les chiffres dans le cadre rouge nous indiquent la vitesse de déplacement des bras du robot.

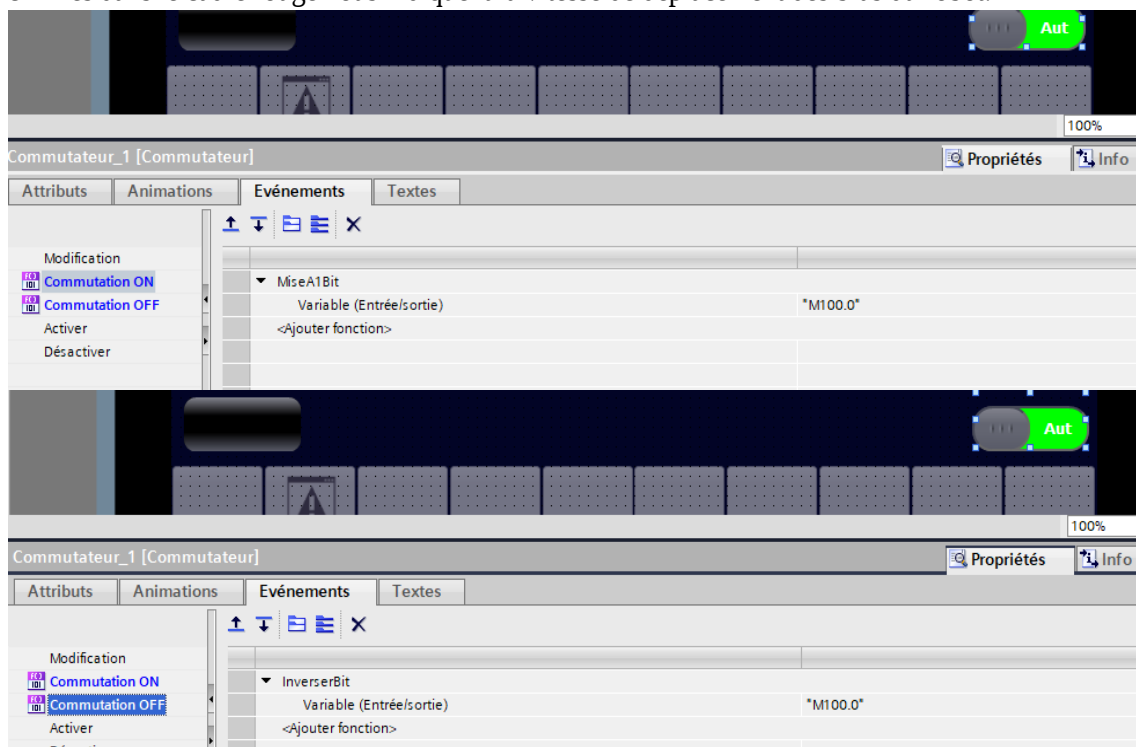


Figure 53 : Button Auto/Manuel en HMI.

- ~ **Communication On** : AN100.0>>Valeur=1 = RLE [1]. Transfer MB"45" Vers DI"41".
- ~ **Communication Off** : Inverse Bit.

III.6 Les Blocs :

- ~ **OB** : Les blocs d'organisation permettent de structurer votre programme. Ils servent d'interface entre le système d'exploitation et le programme utilisateur. Les OB sont déclenchés sur événement. La CPU exécute un OB à la suite d'un événement, tel qu'une alarme de diagnostic ou un intervalle de temps. Certains OB ont des événements déclencheurs et un comportement prédéfini. A l'état MARCHE, les OB de cycle de programme s'exécutent au niveau de priorité le plus bas et peuvent être interrompus par tous les autres types d'événements ;
- ~ **FB** : Les blocs fonctionnels sont des blocs de code qui sauvegardent en permanence leurs valeurs dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement du bloc ;
- ~ **FC** : une fonction ne possède pas une zone de mémoire propre. Les données locales d'une fonction sont perdues après l'exécution de la fonction. Il est également possible d'appeler d'autre FB et FC dans une fonction via des instructions d'appel de blocs ;
- ~ **DB** : (Blocs de données) : Ces blocs de données servent uniquement à stocker des informations et des données mais pas d'instructions, ces données seront utilisées par d'autres blocs.

III.6.1 Création du programme :

III.6.2 Bloc de conversion AI « OB1 » :

CALL		
▼ Titre du bloc "Main Program Sweep (Cycle)"		
Commentaire		
▼ Réseau 1 :		
Commentaire		
1	CALL "Bloc_1"	%FC1

Figure 54 : Bloc_1 [OB1]

III.6.3 Lire AI « FC1 » :

Les Figures suivantes représentent un bloc de fonction (FC1) :

Réseau 2 : CYCLE MANUEL FANUC		
Commentaire		
1	AN "M100.0"	%M100.0
2	JCN MAN3	
3	L 0	0
4	T "Bloc de données_FANUC_1".CMDMAN_FANUC	%DB2.DBB2
5	MAN3: NOP 0	
6		
7	A(	
8	L "Bloc de données_FANUC_1".Page_IHM_Manuel	%DB2.DBW0
9	L 2	2
10	<>I	
11	)	
12	ON "M100.0" //MANUEL CYCLE	%M100.0
13	JC M001	
14		
15	A(	
16	L "Bloc de données_FANUC_1".CMDMAN_FANUC	%DB2.DBB2
17	L 1	1
18	==I	
19	)	
20	L 1 //AVANCE AXE j1 +	1
21	JC M002	
22		
▼ Réseau 1 : ACTIVE_MACRO_FANUC		
ACTIVE_MACRO_FANUC_DI41		
1	A "M100.0" //MANUEL CYCLE	%M100.0
2	A "M100.1" //ENABLE_COMMANDE_MANUEL	%M100.1
3	A(	
4	L "MB45" //GI_6_FANUC	%MB45
5	L 0	0
6	<>I	
7	)	
8	= "DI41" //MACRO_FANUC	%M141.0

Figure 55 : Réseau 1.

❖ Réseau 1 :

Condition si :

- ~ Bouton manuel = 1 ;
- ~ Bouton marche = 1 ;
- ~ Mb45 différent de zéro ;
- ~ Active le bit M141.0 (MACRO).

23	A(					
24	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
25	L	2				2
26	==I					
27	)					
28	L	2	//AVANCE AXE j1 -			2
29	JC	M002				
30						
31	A(					
32	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
33	L	3				3
34	==I					
35	)					
36	L	3	//AVANCE AXE j2 +			3
37	JC	M002				
38						
39						
40	A(					
41	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
42	L	4				4
43	==I					
44	)					
45	L	4	//AVANCE AXE j2 -			4
46	JC	M002				
47						
48	A(					
49	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
50	L	5				5
51	==I					
52	)					
53	L	5	//AVANCE AXE j3 +			5
54	JC	M002				
55						
56	A(					
57	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
58	L	6				6
59	==I					
60	)					
61	L	6	//AVANCE AXE j3 -			6
62	JC	M002				
63						
64	A(					
65	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
66	L	7				7
67	==I					
68	)					
69	L	7	//AVANCE AXE j4 +			7
70	JC	M002				
71						
70	JC	M002				
71						
72	A(					
73	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
74	L	8				8
75	==I					
76	)					
77	L	8	//AVANCE AXE j4 -			8
78	JC	M002				
79						
80	A(					
81	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
82	L	9				9
83	==I					
84	)					
85	L	9	//AVANCE AXE j5 +			9
86	JC	M002				
87						
88	A(					
89	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
90	L	10				10
91	==I					
92	)					
93	L	10	//AVANCE AXE j5 -			10
94	JC	M002				
95	JC	M002				
96	A(					
97	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
98	L	11				11
99	==I					
100	)					
101	L	11	//AVANCE AXE j6 +			11
102	JC	M002				
103	A(					
104	L	"Bloc de données_FANUC_1".CMDMAN_FANUC				%DB2.DBB2
105	L	12				12
106	==I					
107	)					
108	L	12	//AVANCE AXE j6 -			12
109	JC	M002				
110						
111	L	0				0
112	M002: T	"MB45"				%MB45
113	M001: NOP	0				

Figure 56 : Réseau 2.

❖ Réseau 2 :

Condition Si :

- ~ Si M100.0 = 0 ;
- ~ Saut mylabel MAN3 ;
- ~ Si M100.0 = 1 Le programme continue à la ligne 8 ;
- ~ DB2.DBW0 différent deux ;
- ~ bouton marche =1 ;
- ~ saut M001 ;
- ~ Si M001=1 Le programme continue à la ligne 15 ;
- ~ Exemple axe J1(+) ;
- ~ DB2.DBB2 Égal 1 ;
- ~ Saut M002 ;
- ~ M002 Transfert vers " MB45" Et l'axe J 1 se déplace à la direction positive ;
- ~ Exemple axe J1(-) ;
- ~ DB2.DBB2 Égal 2 ;
- ~ Saut M002 ;
- ~ M002 Transfert vers " MB45" Et l'axe J 1 se déplace à la direction négative ;
- ~ Ainsi, de suite pour toutes les touches de la HMI.

Réseau 3 : OVERRIDE IN MANUEL			
Commentaire			
1	L	"Bloc de données _FANUC_1".SPEED_IHM	%DB2.DBB3
2	T	"MB46"	%MB46

Figure 57: Réseau 3.

❖ Réseau 3 :

Condition Si :

- ~ Charger DB2.DBB3 depuis HMI ;
- ~ Transfert vers "MB46" Pour contrôler la vitesse de déplacement du bras du robot.

III.6.4 Posision des axes fanuc_1 [FB1] :

Posision_of_axes_Fanuc_1

Nom	Type de données	Décalage	Valeur par déf.	Accessible ...	Ecritu...	Visible da...	Valeur de ..	Surveilla...	Commentaire
Input									
Go(n)_Fanuc	Word	0.0	16#0	☒	☒	☒	☐		Word output fanuc for Posision
DO_FANUC_BIT_NEG...	Bool	2.0	false	☒	☒	☒	☐		
Output									
Real_posi_axe_HMI	Real	4.0	0.0	☒	☒	☒	☐		DB POSISION HMI
Bloc de données_FAN...	Bool	8.0	false	☒	☒	☒	☐		
InOut									

Commentaire

Réseau 1 :

Commentaire

1	L	#"Go(n)_Fanuc"	
2	L	10	10
3	/R		
4	T	#Real_posi_axe_HMI	

Réseau 2 :

Commentaire

1	A	#DO_FANUC_BIT_NEGATIF	
2	=	#"Bloc de données_FANUC_1.Bit_valeur_Negative_J(N) "	

Figure 58: Bloc paramétré [FB1].

Posision_of_axes_Fanuc_1_DB

Nom	Type de données	Décalage	Valeur de départ	Rémanence	Accessible ...	Ecritu...	Visible da...	Valeur de ..	Surveilla...	Commentaire
Input										
Go(n)_Fanuc	Word	0.0	16#0	☒	☒	☒	☒	☐		Word output fanuc for Posision
DO_FANUC_BIT_NEG...	Bool	2.0	false	☒	☒	☒	☒	☐		
Output										
Real_posi_axe_HMI	Real	4.0	0.0	☒	☒	☒	☒	☐		DB POSISION HMI
Bloc de données_FAN...	Bool	8.0	false	☒	☒	☒	☒	☐		
InOut										
Static										

Posision_of_axes_Fanuc_1_DB_1

Nom	Type de données	Décalage	Valeur de départ	Rémanence	Accessible ...	Ecritu...	Visible da...	Valeur de ..	Surveilla...	Commentaire
Input										
Go(n)_Fanuc	Word	0.0	16#0	☒	☒	☒	☒	☐		Word output fanuc for Posision
DO_FANUC_BIT_NEG...	Bool	2.0	false	☒	☒	☒	☒	☐		
Output										
Real_posi_axe_HMI	Real	4.0	0.0	☒	☒	☒	☒	☐		DB POSISION HMI
Bloc de données_FAN...	Bool	8.0	false	☒	☒	☒	☒	☐		
InOut										
Static										

Posision_of_axes_Fanuc_1_DB_2

Nom	Type de données	Décalage	Valeur de départ	Rémanence	Accessible ...	Ecritu...	Visible da...	Valeur de ..	Surveilla...	Commentaire
Input										
Go(n)_Fanuc	Word	0.0	16#0	☒	☒	☒	☒	☐		Word output fanuc for Posision
DO_FANUC_BIT_NEG...	Bool	2.0	false	☒	☒	☒	☒	☐		
Output										
Real_posi_axe_HMI	Real	4.0	0.0	☒	☒	☒	☒	☐		DB POSISION HMI
Bloc de données_FAN...	Bool	8.0	false	☒	☒	☒	☒	☐		
InOut										
Static										

Posision_of_axes_Fanuc_1_DB_3

Nom	Type de données	Décalage	Valeur de départ	Rémanence	Accessible ...	Ecritu...	Visible da...	Valeur de ..	Surveilla...	Commentaire
Input										
Go(n)_Fanuc	Word	0.0	16#0	☒	☒	☒	☒	☐		Word output fanuc for Posision
DO_FANUC_BIT_NEG...	Bool	2.0	false	☒	☒	☒	☒	☐		
Output										
Real_posi_axe_HMI	Real	4.0	0.0	☒	☒	☒	☒	☐		DB POSISION HMI
Bloc de données_FAN...	Bool	8.0	false	☒	☒	☒	☒	☐		
InOut										
Static										

Position_of_axes_Fanuc_1_DB_4										
	Nom	Type de données	Décalage	Valeur de départ	Rémanence	Accessible ...	Ecritu...	Visible da...	Valeur de ..	Surveilla...
	▼ Input									
	Go(n)_Fanuc	Word	0.0	16#0	☒	☒	☒	☒		Word output fanuc for Posision
	DO_FANUC_BIT_NEG...	Bool	2.0	false	☒	☒	☒	☒		
	▼ Output									
	Real_posi_axe_HMI	Real	4.0	0.0	☒	☒	☒	☒		DB POSISION HMI
	Bloc de données_FAN...	Bool	8.0	false	☒	☒	☒	☒		
	InOut				☐	☐	☐	☐		
	Static				☐	☐	☐	☐		

Position_of_axes_Fanuc_1_DB_5										
	Nom	Type de données	Décalage	Valeur de départ	Rémanence	Accessible ...	Ecritu...	Visible da...	Valeur de ..	Surveilla...
1	▼ Input									
2	Go(n)_Fanuc	Word	0.0	16#0	☒	☒	☒	☒		Word output fanuc for Posision
3	DO_FANUC_BIT_NEG...	Bool	2.0	false	☒	☒	☒	☒		
4	▼ Output									
5	Real_posi_axe_HMI	Real	4.0	0.0	☒	☒	☒	☒		DB POSISION HMI
6	Bloc de données_FAN...	Bool	8.0	false	☒	☒	☒	☒		
7	InOut				☐	☐	☐	☐		
8	Static				☐	☐	☐	☐		

Figure 59 : Posision_foaxes_fanuc1_[DB].

Réseau 4 :										
AXE " J1J6 EN JOINT "										
1	// AXE_J1									
2	CALL "Posision_of_axes_Fanuc_1", "Posision_of_axes_Fanuc_1_DB"									\$FB1, \$DB1
3	Go(n)_Fanuc := "Go_J1"									\$MW48
4	DO_FANUC_BIT_NEGATIF := "DO_FANUC_BIT_NEGATIF_J1"									\$M26.0
5	Real_posi_axe_HMI := "Bloc de données_FANUC_1".POS_J1									\$DB2.DBD4
6	Bloc de données_FANUC_1.Bit_valeur_Negative_J(N) := "Bloc de données_FANUC_1".Bit_valeur_Negative_J1									\$DB2.DBX28.0
7	// AXE_J2									
8	CALL "Posision_of_axes_Fanuc_1", "Posision_of_axes_Fanuc_1_DB_1"									\$FB1, \$DB3
9	Go(n)_Fanuc := "Go_J2"									\$MW50
10	DO_FANUC_BIT_NEGATIF := "DO_FANUC_BIT_NEGATIF_J2"									\$M26.1
11	Real_posi_axe_HMI := "Bloc de données_FANUC_1".POS_J2									\$DB2.DBD8
12	Bloc de données_FANUC_1.Bit_valeur_Negative_J(N) := "Bloc de données_FANUC_1".Bit_valeur_Negative_J2									\$DB2.DBX28.1
13	// AXE_J3									
14	CALL "Posision_of_axes_Fanuc_1", "Posision_of_axes_Fanuc_1_DB_2"									\$FB1, \$DB4
15	Go(n)_Fanuc := "Go_J3"									\$MW52
16	DO_FANUC_BIT_NEGATIF := "DO_FANUC_BIT_NEGATIF_J3"									\$M26.2
17	Real_posi_axe_HMI := "Bloc de données_FANUC_1".POS_J3									\$DB2.DBD12
18	Bloc de données_FANUC_1.Bit_valeur_Negative_J(N) := "Bloc de données_FANUC_1".Bit_valeur_Negative_J3									\$DB2.DBX28.2
19	// AXE_J4									
20	CALL "Posision_of_axes_Fanuc_1", "Posision_of_axes_Fanuc_1_DB_3"									\$FB1, \$DB5
21	Go(n)_Fanuc := "Go_J4"									\$MW54
22	DO_FANUC_BIT_NEGATIF := "DO_FANUC_BIT_NEGATIF_J4"									\$M26.3
23	Real_posi_axe_HMI := "Bloc de données_FANUC_1".POS_J4									\$DB2.DBD16
24	Bloc de données_FANUC_1.Bit_valeur_Negative_J(N) := "Bloc de données_FANUC_1".Bit_valeur_Negative_J4									\$DB2.DBX28.3
25	// AXE_J5									
26	CALL "Posision_of_axes_Fanuc_1", "Posision_of_axes_Fanuc_1_DB_4"									\$FB1, \$DB6
27	Go(n)_Fanuc := "Go_J5"									\$MW56
28	DO_FANUC_BIT_NEGATIF := "DO_FANUC_BIT_NEGATIF_J5"									\$M26.4
29	Real_posi_axe_HMI := "Bloc de données_FANUC_1".POS_J5									\$DB2.DBD20
30	Bloc de données_FANUC_1.Bit_valeur_Negative_J(N) := "Bloc de données_FANUC_1".Bit_valeur_Negative_J5									\$DB2.DBX28.4
31	// AXE_J6									
32	CALL "Posision_of_axes_Fanuc_1", "Posision_of_axes_Fanuc_1_DB_5"									\$FB1, \$DB7
33	Go(n)_Fanuc := "Go_J6"									\$MW58
34	DO_FANUC_BIT_NEGATIF := "DO_FANUC_BIT_NEGATIF_J6"									\$M26.5
35	Real_posi_axe_HMI := "Bloc de données_FANUC_1".POS_J6									\$DB2.DBD24
36	Bloc de données_FANUC_1.Bit_valeur_Negative_J(N) := "Bloc de données_FANUC_1".Bit_valeur_Negative_J6									\$DB2.DBX28.5

Figure 60 : Réseau 4.

❖ Réseau 4 :

Condition Si :

- ~ CALL bloc paramétré [FB1].
- ~ Visibilité du symbole (-) sur HMI.

III.7 Simulation TIA Portal :

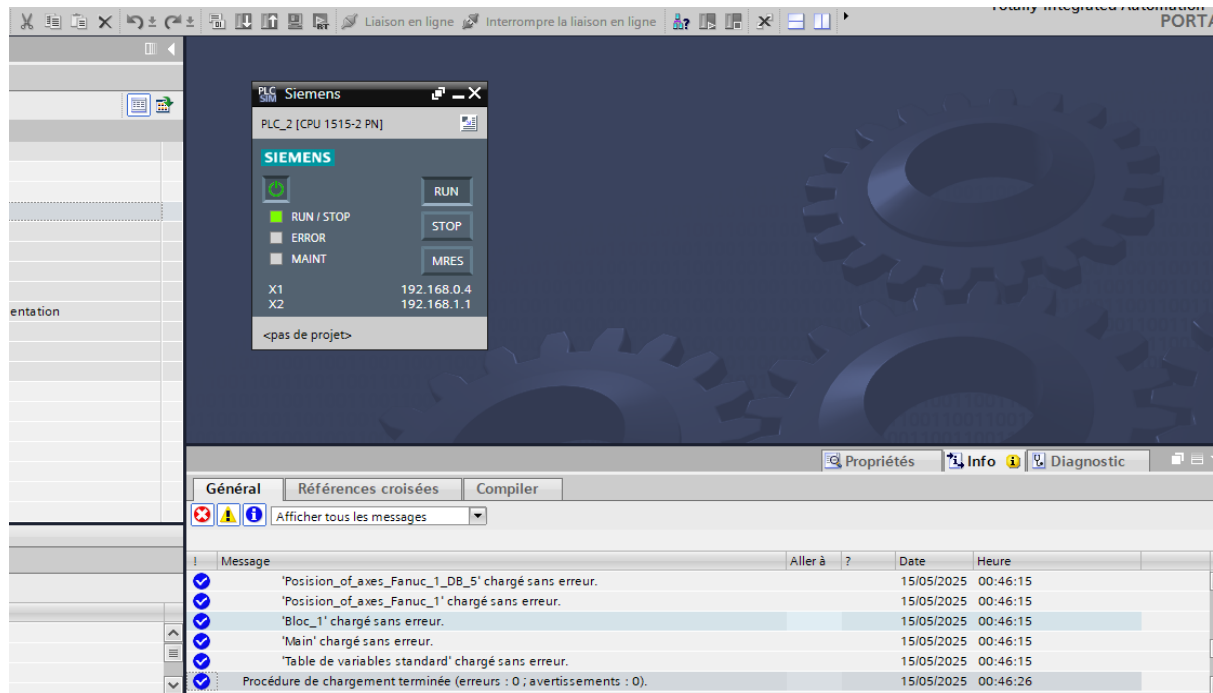


Figure 61 : Charger programme avec PLCSim.

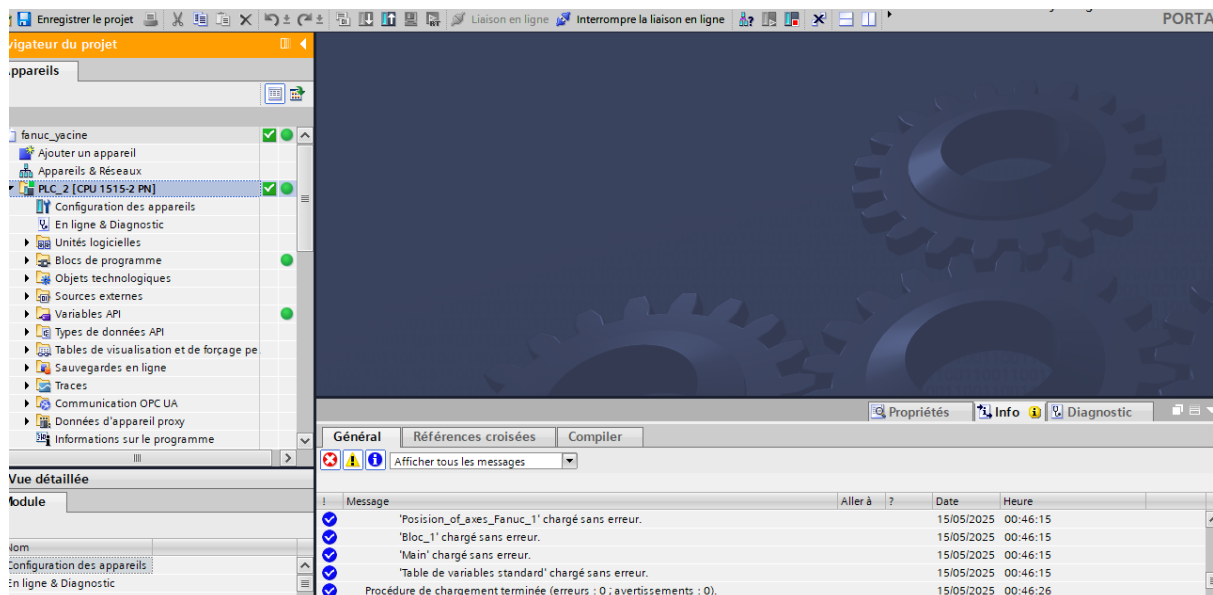


Figure 62 : Liaison en ligne.

▼ Réseau 1 : ACTIVE_MACRO_FANUC

ACTIVE_MACRO_FANUC_DI41

			Adresse	RLO	Valeur	Supplém
1	A	"M100.0"	%M100.0	0	0	
2	A	"M100.1"	%M100.1	0	0	
3	A	(		0		
4	L	"MB45"	%MB45	0	16#00	
5	L	0	0	0	0	
6	<>I			0		
7	)			0		
8	=	"DI41"	%M141.0	0	0	

Figure 63 : visualisation de réseau 1.

▼ Réseau 2 : CYCLE MANUEL FANUC

Commentaire

			Adresse	RLO	Valeur	Supplém
1	AN	"M100.0"	%M100.0	1	0	
2	JCN	MAN3		1		
3	L	0	0	1	0	
4	T	"Bloc de données_FANUC_1".CMEMAN_FANUC	%DB2.DBB2	1	16#00	
5	MAN3:	NOP 0				
6						
7	A	(		1		
8	L	"Bloc de données_FANUC_1".Page_IHM_Manuel	%DB2.DBW0	1	16#0000	
9	L	2	2	1	2	
10	<>I			1		
11	)			1		
12	ON	"M100.0"	%M100.0	1	0	
13	JC	M001		1		
14						
15	A	(				
16	L	"Bloc de données_FANUC_1".CMEMAN_FANUC	%DB2.DBB2			
17	L	1				
18	==I					
19	)					

Figure 64 : Visualisation de réseau 2.

▼ Réseau 3 : OVERRIDE IN MANUEL

Commentaire

			Adresse	RLO	Valeur	Supplém
1	L	"Bloc de données_FANUC_1".SPEED_IRM	%DB2.DBB3	1	16#0A	
2	T	"MB46"	%MB46	1	16#0A	

Figure 65 : Visualisation de réseau 3.

▼ Réseau 4 : ...

AXE "J1J6 EN JOINT"

			Adresse	RLO	Valeur	Supplém
1	//	AXE_J1				
2	CALL	"Position_of_axes_Fanuc_1", "Position_of_axes_Fanuc_1_DB"	%FB1, %DB1		In	Out
3	Go(n)_Fanuc	:= "Go_J1"	%MW48		16#0000	
4	DO_FANUC_BIT_NEGATIF	:= "DO_FANUC_BIT_NEGATIF_J1"	%M26.0		0	
5	Real_posi_axe_HMI	:= "Bloc de données_FANUC_1".POS_J1	%DB2.DBD4			16#7FC0_C
6	Bloc de données_FANUC_1.Bit_valeur_Negative_J(N)	:= "Bloc de données_FANUC_1".Bit_va	%DB2.DBX28.0			0
7	//	AXE_J2				
8	CALL	"Position_of_axes_Fanuc_1", "Position_of_axes_Fanuc_1_DB_1"	%FB1, %DB3		In	Out
9	Go(n)_Fanuc	:= "Go_J2"	%MW50		16#0000	
10	DO_FANUC_BIT_NEGATIF	:= "DO_FANUC_BIT_NEGATIF_J2"	%M26.1		0	
11	Real_posi_axe_HMI	:= "Bloc de données_FANUC_1".POS_J2	%DB2.DBD8			16#7FC0_C
12	Bloc de données_FANUC_1.Bit_valeur_Negative_J(N)	:= "Bloc de données_FANUC_1".Bit_va	%DB2.DBX28.1			0

Figure 66 : Visualisation de réseau 4.

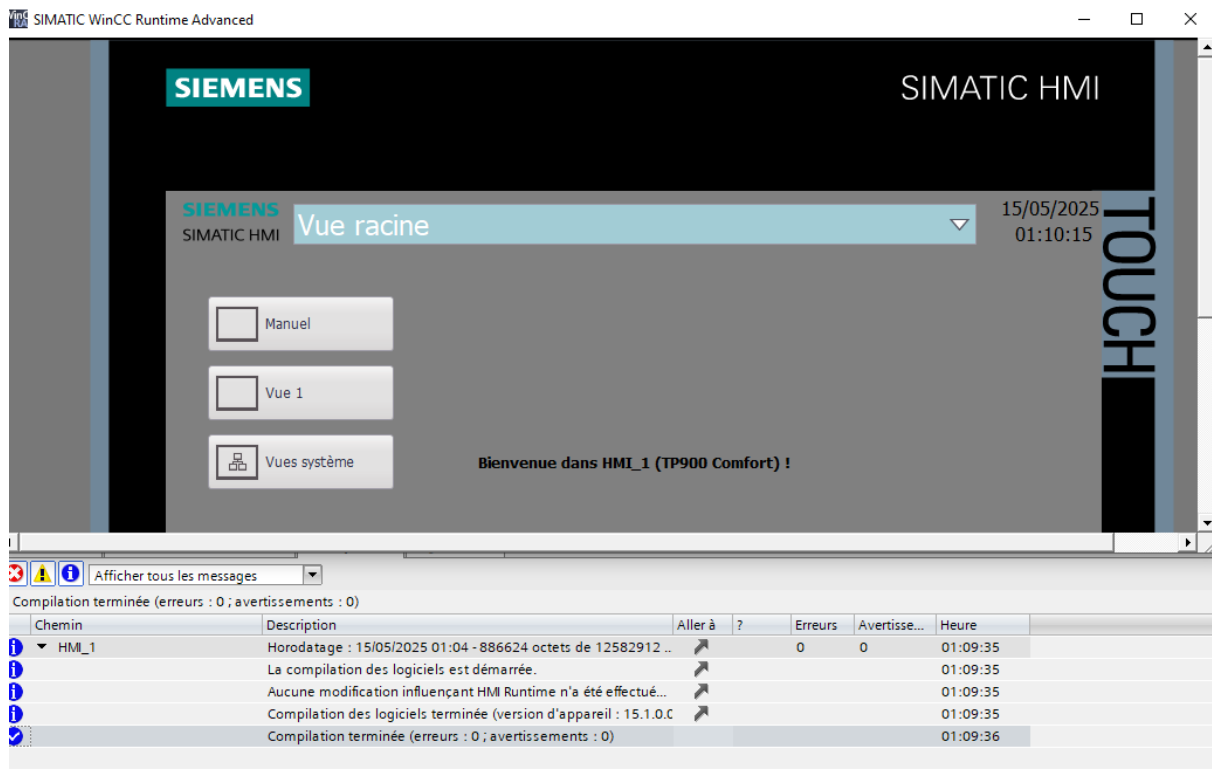


Figure 67: Simulation HMI sans erreurs.

III.8 Conclusion :

Dans ce chapitre, nous avons expliqué comment contrôler le robot, en utilisant l'application robot et Tia Portal. Grâce à l'intégration de systèmes automatisés et de robots industriels, l'usine Guedila parvient à assurer une production en grande quantité, avec un haut niveau de qualité et une réduction des coûts opérationnels.

CONCLUSION GÉNÉRALE

CONCLUSION GENERALE

Les robots ont été initialement conçus comme une source d'amusement et divertissement. Au moment où le domaine de la robotique progresse, il a changé la face de son objectif initial. Maintenant, les robots sont partie essentielle des industries automatisées. Ils deviennent plus d'un besoin qu'un luxe pour la croissance industrielle. En outre, les robots sont utilisés pour la rééducation, l'assistance, réalité virtuelle, centrales nucléaires et ainsi de suite.

Actuellement la robotique est l'un des sujets les plus intéressants pour les scientifiques 13.

- La robotique est la science qui s'intéresse aux robots. En fait, il s'agit d'un domaine multidisciplinaire : on y trouve des aspects concernant la mécanique, l'informatique, l'électronique, ...
- Un automatisme est un appareillage assurant le fonctionnement automatique d'une machine ou d'une installation 12 .
- Un robot est un système mécanique poly articulé mû par des actionneurs et commandé par un ordinateur qui est voué à effectuer une large pluralité de tâches. Il est autant défini comme étant une machine apte d'exercer des tâches et de manier des objets selon un programme de manière automatique. Ils sont habituellement utilisés pour succéder les humains dans des situations où ces derniers sont impuissants de pratiquer le travail, des circonstances plus délicates, d'une importante exactitude ou répétitif.
- Ce stage nous a été une expérience enrichissante et bénéfique parce qu'il nous a permis de consolider nos connaissances théoriques et pratiques, de nous confronter à la réalité professionnelle pour notre intégration future, et de développer le sens de la responsabilité en s'engageant dans un projet portant un intérêt pour l'entreprise

L'objectif de cette mémoire de fin de cycle est de :

- L'un des principaux objectifs est de mettre en pratique les compétences et les connaissances acquises au cours du diplôme et des stages en entreprise en ce qui concerne les différentes branches de l'ingénierie telles que la robotique, le contrôle, l'automatisation ou la vision artificielle ;
- Conception et développement d'un projet de simulation regroupant différentes branches de l'ingénierie ;
- Appliquer les notions de programmation robotique apprises lors des stages en entreprise.
- Servir d'outil d'étude pour les cellules robotiques dans les environnements marque FANUC simulée. Qui devient de plus en plus présent dans les industries proches

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [1] IAMRACHE MASSINISSA, B,N . Etude et réalisation d'un bras robotique. thème de master Université Mouloud Mammeri De Tizi-Ouzou2017
- [2] https://elearning.univmsila.dz/moodle/pluginfile.php/465379/mod_folder/content/0/chapitre%201-20211104.zip?forcedownload=1
- [3] [Automatique et Informatique Industrielle – Ecole Nationale Supérieure des Technologies Avancées \(ensta.edu.dz\)](#)
- [4] www.facuc.eu FANUC ACADEMY
- [5] Villa González, J. (2023). Simulación de una instalación robotizada de soldadura de piezas, controlada por PLC, gestionada por una Interfaz Hombre-Máquina y comunicada mediante OPC UA
- [6] BOUDIBA, N. I. Etude de la commande d'automatique de la machine banderoleuse de palette par S7-300. thème de master université de Mohamed Khider Biskra 2019.
- [7] https://cache.industry.siemens.com/dl/files/792/59191792/att_856651/v1/s71500_et200mp_system_manual_fr-FR_fr-FR.pdf
- [8] <https://electutor.com/les-automates-programmables-simatic-s7-1500-de-siemens/>
- [9] [FANUC | The Factory Automation Company](#)
- [10] <https://www.guedila.com/Presentation>
- [11] Telmami karim et Naili Amine memoir master command des bras de robot
- [12] <http://archives.univ-biskra.dz/bitstream/123456789/11176/3/Bensizerara-Saliha.pdf>
- [13] http://archives.univbiskra.dz/bitstream/123456789/22328/1/KHIREDDINE_Salah.pdf