



AX Portal

Guide de l'utilisateur

Consignes traduites

©Seiko Epson Corporation 2026

Rev. 2

FRM265S8837M

Table des matières

Marques.....	8
Conditions d'utilisation.....	8
1. Introduction.....	9
1.1 Programmation et commande.....	10
1.2 Interface utilisateur graphique (GUI).....	10
1.3 Modules logiciels.....	10
1.3.1 Module de connexion TCP.....	10
1.3.2 Module de connexion REST.....	11
1.3.3 Module de connexion ROS.....	11
1.3.4 Module de connexion Modbus.....	11
1.3.5 Interface du panneau ("Panel Interface").....	11
1.3.6 Module de caméra Cognex.....	11
2. Interface d'administration (GUI).....	12
2.1 Introduction.....	13
2.1.1 Langue de l'interface.....	14
2.1.2 Tableau de bord.....	14
2.2 Gestion des utilisateurs.....	14
2.2.1 Niveaux d'autorisation.....	14
2.2.2 Page de connexion.....	14
2.2.3 Gérer votre compte.....	15
2.2.4 Gérer d'autres comptes utilisateurs.....	16
2.3 Paramètres système.....	17
2.3.1 Paramètres réseau.....	17
2.3.2 Paramètres de l'heure.....	18
2.3.3 Certificat SSL.....	18
2.4 Mise à jour logicielle.....	20
2.5 Aide et assistance.....	22
2.6 Authentification sécurisée.....	23
2.6.1 Exporter le certificat SSL.....	23
2.6.2 Connexion de l'utilisateur.....	23
2.6.3 Enregistrement de la connexion.....	24
3. Exportation USB et mise à jour de restauration.....	25
3.1 Aperçu.....	26
3.2 Exportation USB.....	26
3.3 Exécuter la mise à jour de restauration.....	26
4. Interface de bureau (GUI).....	27
4.1 Introduction.....	28
4.2 Fenêtre principale.....	30
4.2.1 Aperçu des menus.....	30
4.2.2 Cockpit.....	31
4.2.3 Commande active (Commande exclusive depuis un poste de commande).....	33
4.2.3.1 Contrôle actif.....	33

4.2.3.2 Observateur passif.....	34
4.2.4 État du programme.....	34
4.2.5 Modes de fonctionnement.....	36
4.2.5.1 Mode automatique.....	36
4.2.5.2 Mode manuel.....	36
4.2.6 Commandes de précision.....	37
4.2.7 Commande de mouvement.....	39
4.2.7.1 Limites de mouvement.....	39
4.2.7.2 Paramètres globaux.....	40
4.2.7.3 Systèmes de coordonnées globaux.....	41
4.3 Blocs fonctionnels et base de données.....	42
4.4 Menu Projet.....	43
4.5 Menu de configuration.....	44
4.5.1 Base de données (téléchargement et téléversement).....	44
4.5.2 Configuration des E/S.....	45
4.5.3 Configuration de l'outil.....	47
4.5.4 Éditeur d'outils.....	47
4.5.4.1 Mode sortie unique.....	50
4.5.4.2 Mode double sortie.....	50
4.5.4.3 Mode de déclenchement séparé.....	50
4.5.5 Mise en service du robot.....	51
4.6 Menu d'apprentissage.....	52
4.6.1 Éditeur de poses.....	54
4.6.2 Éditeur de trajectoire.....	56
4.6.3 Éditeur de grille.....	58
4.7 Menu Applications.....	62
4.7.1 Éditeur d'applications.....	63
4.7.2 Code du programme.....	63
4.7.3 Paramètres de l'application.....	64
4.7.4 Variables d'application.....	67
4.7.5 Variables partagées et globales.....	68
4.7.6 Éditeur de dossiers.....	69
4.7.7 Programmation par blocs.....	69
4.7.7.1 Utilisation des variables d'application et des expressions.....	70
4.7.7.2 Bloc Mouvement.....	71
4.7.7.3 Bloc de trajectoire.....	72
4.7.7.4 Bloc « Saisir ».....	72
4.7.7.5 Bloc « Placer ».....	72
4.7.7.6 Bloc « Conteneur ».....	73
4.7.7.7 Bloc « Pose de grille suivante ».....	73
4.7.7.8 Bloc « Réinitialiser la pose de la grille ».....	73
4.7.7.9 Bloc « Écriture de variable ».....	73
4.7.7.10 Bloc « Définir la sortie ».....	73
4.7.7.11 Bloc Attendre.....	73

4.7.7.12 Bloc « Boucle ».....	74
4.7.7.13 Bloc « Sortie si ».....	74
4.7.7.14 Bloc « conditionnel ».....	74
4.7.7.15 Bloc de commentaire.....	74
4.7.7.16 Bloc « Quitter ».....	74
4.7.7.17 Bloc « Boîte de dialogue ».....	74
4.7.7.18 Bloc de sortie.....	74
4.7.7.19 Bloc de journalisation.....	75
4.7.7.20 Bloc de notification.....	75
4.7.7.21 Bloc d'avertissement.....	75
4.7.7.22 Bloc d'erreur.....	75
4.7.7.23 Bloc de code Python.....	75
4.8 Menu E/S.....	76
4.9 Menu Visualisation.....	77
4.10 Commande manuelle.....	78
4.11 Moniteur système.....	79
4.11.1 État du robot.....	79
4.11.2 Sortie texte.....	79
4.11.3 Journal des applications.....	79
4.11.4 Historique des messages.....	79
4.12 Menu Aide.....	80
4.12.1 Informations sur le robot.....	80
4.12.2 Raccourcis clavier.....	80
4.12.3 Documentation du code.....	80
4.12.4 Licence open source.....	80
4.12.5 Coordonnées.....	80
4.13 Menu des paramètres.....	81
4.13.1 Paramètres de l'interface graphique.....	82
4.13.2 Personnaliser les boutons du cockpit.....	82
4.13.3 Paramètres de démarrage du programme.....	83
4.13.4 Comportement de mouvement.....	83
4.13.4.1 Comportement en cas d'interruption.....	83
4.13.4.2 Boîte de dialogue d'interruption.....	84
4.14 Menu des paramètres de sécurité.....	87
4.14.1 Interface utilisateur.....	87
4.14.2 Paramètres de sécurité.....	89
4.14.2.1 Position limite de sécurité (SLP).....	89
4.14.2.2 Angles d'articulation limités en toute sécurité.....	90
4.14.2.3 Vitesse limitée en toute sécurité (SLS).....	91
4.14.2.4 Paramètres de sécurité pour la puissance et la force.....	92
4.14.2.5 E/S de sécurité et restrictions d'accès.....	93
4.14.2.6 Données de vérification.....	94
4.14.2.7 Fonctions de sortie de sécurité.....	94
4.14.2.8 Raccordement d'appareils optionnels.....	94

5. Interface panneau (GUI).....	95
5.1 Introduction.....	96
5.2 Configurer les applications pour l'interface du panneau ("Panel Interface").....	97
5.3 Cockpit et paramètres.....	99
5.4 Contrôler les applications.....	100
6. Définitions cinématiques.....	101
6.1 Coordonnées.....	102
6.1.1 Angles nautiques.....	102
6.1.2 Systèmes de référence.....	103
6.2 Configuration.....	104
6.3 Pose.....	104
6.4 Point.....	104
6.5 Trajectoire.....	104
6.6 Segment.....	105
6.7 Grilles.....	105
7. Modules de connexion.....	107
7.1 Connexions, authentification et contrôle actif.....	109
7.1.1 Canaux de connexion.....	109
7.1.2 Authentification.....	109
7.1.3 Commande active.....	109
7.2 Module de connexion TCP.....	111
7.2.1 Établir une connexion.....	111
7.2.2 Envoi de commandes.....	112
7.2.3 Paramètres.....	113
7.2.3.1 Norme de codage.....	113
7.2.3.2 Caractère de fin.....	113
7.2.3.3 Envoyer un heartbeat.....	113
7.2.4 Journal des messages – Réception des commandes.....	113
7.2.4.1 Informations d'état.....	114
7.2.4.2 Valeur de retour de l'action.....	114
7.2.4.3 Erreurs.....	115
7.2.4.4 Historique des messages.....	115
7.2.4.5 Définition d'un groupe d'états.....	115
7.2.4.6 Données d'un groupe d'état.....	116
7.2.4.7 Ouvrir une boîte de dialogue.....	116
7.2.4.8 Envoi depuis une application.....	117
7.2.5 Journal des messages – Envoi de commandes.....	117
7.2.5.1 Connexion par jeton.....	118
7.2.5.2 Demander le contrôle actif.....	118
7.2.5.3 Libérer le contrôle actif.....	118
7.2.5.4 Interroger les données.....	119
7.2.5.5 Mise en service.....	119
7.2.5.6 Mise hors service.....	119
7.2.5.7 Mise en pause.....	120

7.2.5.8 Reprendre.....	120
7.2.5.9 Activation du mode de commande manuelle (HGC).....	120
7.2.5.10 Désactivation du mode de commande manuelle (HGC).....	121
7.2.5.11 Arrêt.....	121
7.2.5.12 Changer de projet.....	121
7.2.5.13 Exécuter une application.....	122
7.2.5.14 Tester l'application.....	122
7.2.5.15 Réception d'un message.....	123
7.2.5.16 Se déplacer vers une pose.....	123
7.2.5.17 Lire le parcours.....	124
7.2.5.18 Changer d'outil.....	124
7.2.5.19 Écriture des sorties numériques et analogiques.....	125
7.2.5.20 Répondre à une boîte de dialogue.....	126
7.2.5.21 Définir une variable d'état.....	126
7.2.5.22 Définir une variable globale.....	127
7.2.5.23 Lire une variable globale.....	127
7.2.5.24 Définir une variable partagée.....	128
7.2.5.25 Lire une variable partagée.....	128
7.3 Module de connexion REST.....	129
7.3.1 Établir une connexion.....	129
7.3.2 Envoyer des commandes.....	130
7.3.3 Paramètres.....	131
7.4 Module de connexion ROS.....	132
7.4.1 Installation.....	132
7.4.2 Configuration.....	133
7.4.3 Détection du réseau avec Cyclone DDS.....	133
7.4.3.1 Interface réseau.....	133
7.4.3.2 Méthodes de détection.....	134
7.4.4 Établir la connexion.....	136
7.4.5 Accès aux données du robot via ROS.....	137
7.4.6 Fonctionnalités ROS 2 dans les scripts.....	137
7.4.7 Appeler une action.....	138
7.4.8 Paramètres d'authentification.....	138
7.5 Module de connexion Modbus.....	139
7.5.1 Aperçu.....	139
7.5.2 Paramètres.....	139
7.5.3 Registres d'écriture (description simple).....	139
7.5.3.1 Registre de commande.....	139
7.5.3.2 Registre de propriétés.....	140
7.5.3.3 Registres d'écriture définis par l'utilisateur.....	141
7.5.4 Registres d'écriture (description détaillée).....	141
7.5.4.1 Ne rien faire.....	141
7.5.4.2 Mise en service.....	141
7.5.4.3 Mise hors service.....	141

7.5.4.4 Arrêt.....	142
7.5.4.5 Mise en pause.....	142
7.5.4.6 Reprendre.....	142
7.5.4.7 Calibrage.....	142
7.5.4.8 Activation de la commande manuelle (HGC).....	142
7.5.4.9 Désactiver le mode manuel (HGC).....	142
7.5.4.10 Changer de projet.....	143
7.5.4.11 Démarrer l'application.....	143
7.5.4.12 Se déplacer vers une pose.....	143
7.5.4.13 Lire le parcours.....	143
7.5.4.14 Placer avec l'outil.....	143
7.5.4.15 Saisir avec l'outil.....	144
7.5.4.16 Écriture sur les sorties numériques.....	144
7.5.4.17 Définition des événements d'application.....	144
7.5.4.18 Acquitter les erreurs.....	144
7.5.4.19 Lecture de la position.....	144
7.5.4.20 Demander le contrôle actif.....	145
7.5.4.21 Libérer le contrôle actif.....	145
7.5.4.22 Passer en mode manuel.....	145
7.5.4.23 Passer en mode automatique.....	145
7.5.4.24 Réinitialisation de la sécurité.....	145
7.5.4.25 Lancer l'interruption de mouvement.....	146
7.5.5 Registre de lecture.....	146
7.5.5.1 Registre d'état.....	146
7.5.5.2 Registres de lecture définis par l'utilisateur.....	149
7.5.5.3 Registres de position (articulations 1 à 6).....	149
7.5.5.4 Registre des coordonnées (x, y, z, roll, pitch, yaw).....	149
7.5.6 Affectation des registres d'erreur.....	151
7.5.6.1 Exemple.....	151
7.5.7 Exemples de code Python.....	152
7.5.7.1 Exemple 1.....	152
7.5.7.2 Exemple 2.....	152
7.5.7.3 Exemple 3.....	153
7.5.7.4 Exemple 4.....	153
8. Module de caméra Cognex.....	154
8.1 Présentation.....	155
8.2 Configuration d'une caméra.....	155
8.3 Utilisation d'une caméra.....	155

Marques

Cognex® et In-Sight Explorer® sont des marques commerciales ou des marques déposées de Cognex Corporation aux États-Unis et dans d'autres pays.

Google Chrome™ et Chrome™ sont des marques commerciales de Google LLC.

Linux® est une marque déposée de Linus Torvalds aux États-Unis et dans d'autres pays.

Apple®, macOS® et Safari® sont des marques déposées d'Apple Inc. aux États-Unis et dans d'autres pays et régions.

Microsoft®, Windows®, Microsoft Edge® et Microsoft Visual C++® sont des marques (ou des marques déposées) de Microsoft Corporation et/ou de ses filiales aux États-Unis et dans d'autres pays.

Mozilla et Firefox® sont des marques de la Fondation Mozilla aux États-Unis et dans d'autres pays.

Remarque générale : les noms de produits utilisés ici sont uniquement destinés à des fins d'identification et peuvent être des marques de leurs propriétaires respectifs. Epson ne revendique aucun droit sur ces marques.

Toutes les captures d'écran sont fournies à titre d'illustration et de formation uniquement. Elles ne visent ni ne sous-entendent aucun lien avec un titulaire de marque ni aucune approbation de sa part.

Conditions d'utilisation

Aucune partie de ce manuel d'utilisation ne peut être reproduite ou réimprimée sous quelque forme que ce soit sans autorisation écrite expresse.

Les informations contenues dans ce document sont susceptibles d'être modifiées sans préavis.

Veuillez nous contacter si vous constatez des erreurs dans ce document ou si vous avez des questions concernant les informations qu'il contient.

1. Introduction

1.1 Programmation et commande

Le framework logiciel AX Portal offre à l'utilisateur d'un robot de la série Epson AX un moyen très intuitif de communiquer avec ce robot. Avec AX Portal, l'utilisateur dispose d'un accès complet pour manipuler et commander le robot, ainsi que pour écrire des applications à l'aide du langage de programmation [Python 3.13](#). Les fonctionnalités de base pour la programmation en Python sont disponibles en ligne. Cependant, quelques exemples de code très élémentaires sont également fournis dans la documentation intégrée relative au code.

Outre un large éventail de fonctions Python de base, les applications AX Portal prennent en charge une vaste collection de fonctions spécialement conçues pour le contrôle, la maintenance et la communication avec les robots AX. Vous trouverez la liste complète des fonctions de script prises en charge par AX Portal dans la documentation intégrée relative au code.

1.2 Interface utilisateur graphique (GUI)

L'interface graphique a été conçue comme une application web pouvant s'exécuter sur les navigateurs web courants de pratiquement n'importe quel appareil. Les navigateurs suivants sont officiellement pris en charge :

- Google Chrome (version ≥ 147)
- Mozilla Firefox (version ≥ 150)
- Apple Safari (version ≥ 26)
- Microsoft Edge (version ≥ 147)

Il n'est donc pas nécessaire de télécharger un module logiciel spécifique pour votre contrôleur, tant que le robot et l'appareil se trouvent sur le même réseau ou partagent un canal de communication direct. En plus de l'interface de bureau standard d'AX Portal, principalement optimisée pour la programmation sur des ordinateurs de bureau et des ordinateurs portables, AX Portal propose une interface de panneau ("Panel Interface") qui ne contient que les fonctions nécessaires à la commande d'un robot déjà préprogrammé. Vous trouverez de plus amples informations sur les interfaces graphiques et toutes les fonctions prises en charge plus loin dans ce document.

1.3 Modules logiciels

Outre les fonctionnalités de base d'AX Portal, décrites en détail dans ce document, les robots AX disposent de plusieurs modules logiciels intégrés offrant des possibilités de commande supplémentaires. Veuillez noter que tous les modules de connexion mentionnés ci-dessous permettent de se connecter à distance au robot AX et offrent ainsi des possibilités de commande et de surveillance du robot.

1.3.1 Module de connexion TCP

Le module de connexion TCP peut être utilisé comme alternative pour communiquer avec le robot. Le robot est généralement commandé par l'utilisateur via l'interface graphique (GUI) dans le navigateur Web. Toutefois, si le robot doit être commandé via une interface personnalisée ou une autre machine, il accepte directement les commandes TCP. Grâce au module de connexion TCP, tout ce qui est disponible via la communication par navigateur peut également être accessible via TCP.

1.3.2 Module de connexion REST

Le module de connexion REST peut être utilisé comme alternative pour communiquer avec le robot. Le robot est généralement commandé par l'utilisateur via l'interface graphique (GUI) dans le navigateur Web. Toutefois, si le robot doit être commandé via une interface personnalisée ou une autre machine, il accepte directement les commandes REST. Grâce au module de connexion REST, tout ce qui est disponible via la communication par navigateur peut également être accessible via REST.

1.3.3 Module de connexion ROS

Le module de connexion ROS offre des fonctionnalités permettant d'intégrer le robot dans un environnement ROS2. Le robot permet à d'autres clients du réseau ROS2 d'envoyer des commandes pour le commander et le surveiller.

1.3.4 Module de connexion Modbus

Le module de connexion Modbus peut être utilisé comme alternative pour communiquer avec le robot. Il utilise le protocole Modbus-TCP et permet à l'utilisateur de lire et d'écrire dans des registres Modbus standard. Utilisez ce module pour commander le robot directement via un API. Les fonctions principales, telles que la mise en service du robot, l'exécution d'applications et l'activation des sorties, peuvent être réalisées via le module Modbus.

1.3.5 Interface du panneau ("Panel Interface")

L'interface du panneau ("Panel Interface") offre une interface utilisateur graphique supplémentaire aux fonctionnalités limitées, destinée davantage aux opérateurs de robots qu'aux programmeurs. Cette interface sert principalement à lancer et à surveiller des applications, et son contenu peut être adapté par le programmeur en fonction de son cas d'utilisation. L'interface du panneau ("Panel Interface") est optimisée pour les appareils à écran tactile et sa conception s'apparente à la structure familière des applications des appareils mobiles modernes, ce qui la rend intuitive même pour les utilisateurs sans connaissances techniques.

1.3.6 Module de caméra Cognex

Le module de caméra Cognex fournit une interface de communication entre une caméra Cognex et AX Portal. Les caméras sont généralement soit intégrées à un effecteur terminal, soit montées de manière fixe afin d'observer la scène dans laquelle le robot opère. Ce module est idéal pour une inspection visuelle personnalisable et prend en charge l'ensemble des fonctionnalités des caméras Cognex.

2. Interface d'administration (GUI)

2.1 Introduction

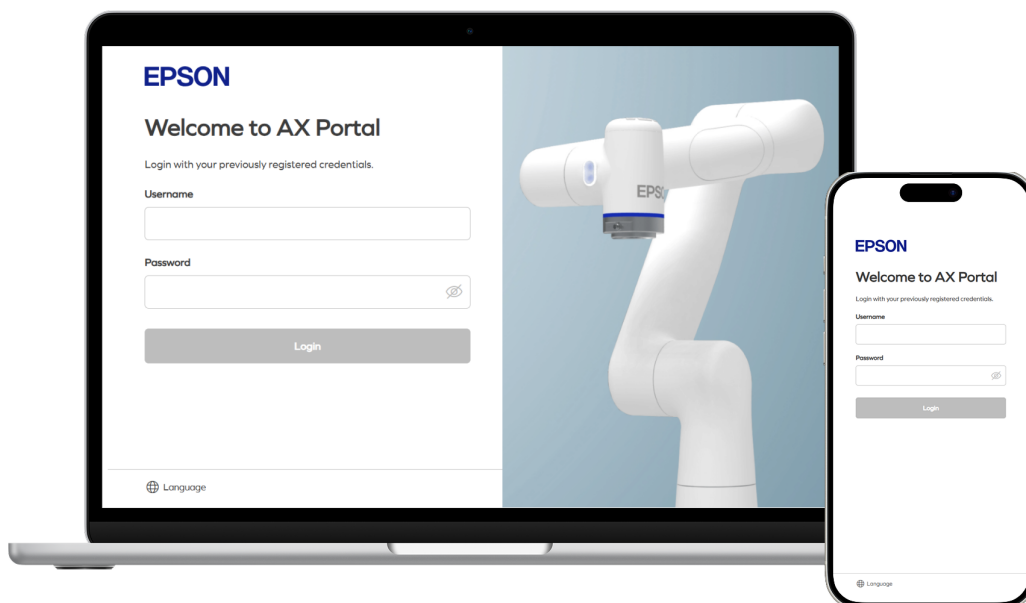


Figure 2.1.a : Exemples de captures d'écran de l'interface d'administration

L'interface d'administration d'AX Portal est une interface utilisateur graphique (GUI) qui permet aux utilisateurs de gérer les comptes utilisateurs et de modifier certains paramètres de base liés au système d'exploitation du robot (tels que les paramètres réseau, les paramètres de date et d'heure, etc.). Ces paramètres ne sont généralement définis qu'une seule fois lors de l'installation du robot, avant même qu'un programmeur ne commence à écrire des applications de commande du robot. Vous pouvez accéder à cette interface via un navigateur Web (par exemple Google Chrome) en saisissant l'adresse IP du robot dans la barre d'adresse du navigateur, ce qui vous redirigera vers <https://<IP-ADDRESS>/admin> (veuillez consulter le « Contrôleur de Robot RC-A101 - Manuel » pour connaître l'adresse IP correcte).

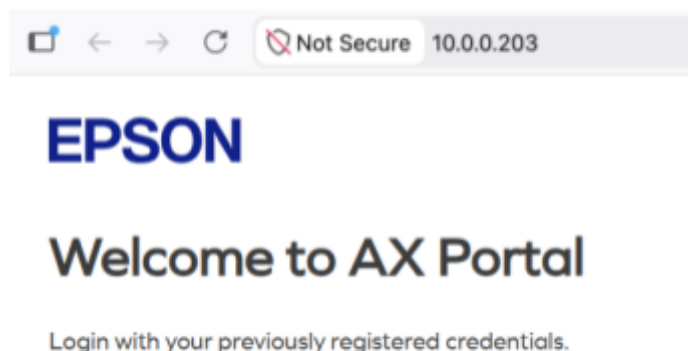


Figure 2.1.b : Connexion à l'interface d'administration via l'URL

Ce chapitre explique en détail tous les menus et éléments de l'interface d'administration.

2.1.1 Langue de l'interface

Pour modifier la langue de l'interface, cliquez sur le menu déroulant des langues situé dans le coin inférieur gauche de l'interface et sélectionnez la langue de votre choix. Votre navigateur vous facilite la tâche en enregistrant votre paramètre de langue d'une session à l'autre et d'une interface à l'autre.

2.1.2 Tableau de bord

Une fois connecté, le tableau de bord de l'interface d'administration vous permet de naviguer vers les différents sous-menus, vers d'autres interfaces d'AX Portal, ainsi que de vous déconnecter. Veuillez noter que le tableau de bord n'affiche que les options auxquelles vous avez actuellement accès, en fonction de votre niveau d'utilisateur (administrateur, responsable de la sécurité, programmeur, opérateur). Si vous ne pouvez pas accéder à d'autres interfaces, cela signifie qu'AX Portal ne fonctionne pas correctement pour le moment, ce qui est par exemple le cas pendant la mise à jour du logiciel.

2.2 Gestion des utilisateurs

2.2.1 Niveaux d'autorisation

Afin d'empêcher que le robot ne soit configuré, programmé ou utilisé par des utilisateurs non autorisés, différents niveaux d'autorisation sont distingués. L'autorisation est définie pour chaque utilisateur exclusivement par les administrateurs. Les niveaux d'autorisation suivants sont disponibles, du plus élevé au plus bas :

- Un **administrateur** dispose d'un accès complet à toutes les interfaces et fonctions de commande du robot, ainsi qu'à la gestion des autres utilisateurs et aux principaux paramètres système du système d'exploitation.
- Un **programmeur** a accès à toutes les interfaces et fonctions de commande du robot et peut donc préparer et programmer des applications qui seront ensuite exécutées par des opérateurs. Au sein de l'interface d'administration, le programmeur n'a essentiellement accès qu'aux données de son propre compte et à la navigation entre les interfaces.
- Un **responsable de la sécurité** est un programmeur (voir ci-dessus) autorisé à modifier les paramètres de sécurité du robot. Les autres utilisateurs ne peuvent que consulter les paramètres de sécurité.
- Un **opérateur** n'a accès qu'aux interfaces utilisateur de base permettant de commander le robot, mais ne peut modifier ni les paramètres ni les programmes. Au sein de l'interface d'administration, l'opérateur n'a pour l'essentiel accès qu'aux données de son propre compte.

2.2.2 Page de connexion

La page de connexion comporte deux formulaires différents, selon qu'un compte administrateur a déjà été créé ou non.

- Si un administrateur a déjà été créé, la page de connexion habituelle s'affiche, sur laquelle vous devez saisir votre nom d'utilisateur et votre mot de passe pour vous connecter. Les mots de passe sont masqués lors de la saisie, sauf si vous cliquez sur le bouton permettant d'afficher le mot de passe dans le champ correspondant.

- Si aucun administrateur n'a encore été créé, vous êtes directement invité à le faire et vous ne pouvez continuer qu'après avoir créé un compte administrateur. Vous trouverez plus de détails dans le « Robot Collaboratif 6 Axes AX6 - Guide de Démarrage Rapide, Manuel d'Installation ». Vous ne pouvez définir ici que les données absolument nécessaires ; si vous souhaitez afficher et définir tous les paramètres de votre compte, accédez au menu « Manage My Account ».

Après chaque déconnexion, vous serez redirigé vers la page de connexion.

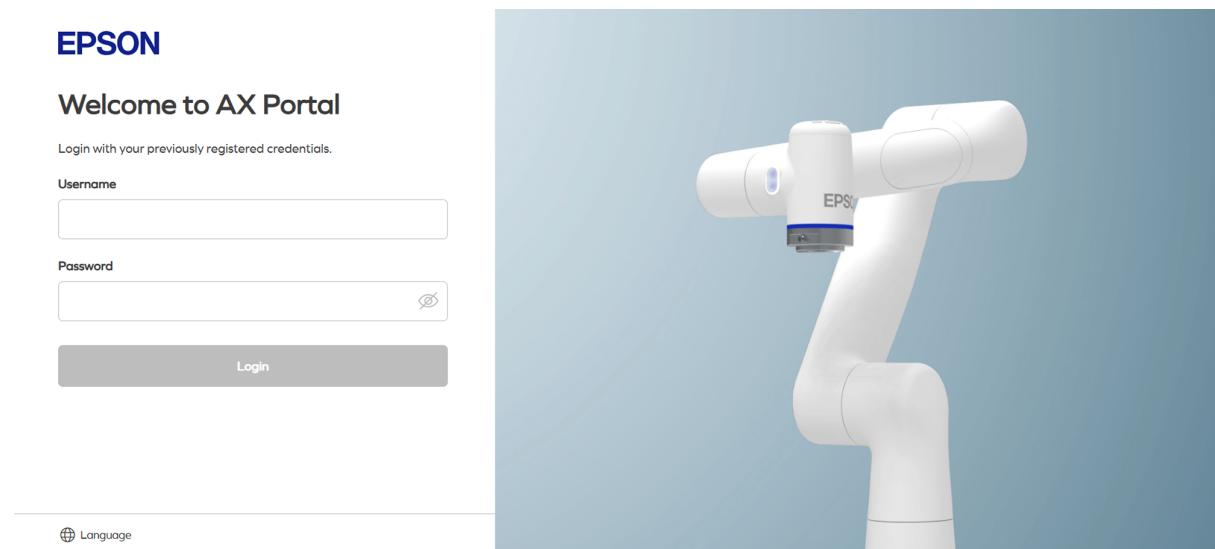


Figure 2.2.2 : Page de connexion

2.2.3 Gérer votre compte

Dans le menu « Gérer mon compte », vous pouvez modifier votre profil, plus précisément votre nom, votre mot de passe, votre photo de profil et l'interface d'accueil :

- Le « Nom complet » est le nom utilisé pour s'adresser à l'utilisateur dans l'interface et dans les messages. Il s'agit d'un champ obligatoire destiné à votre prénom et à votre nom ; vous pouvez toutefois définir vos propres conventions de nommage.
- Le champ « Nom d'utilisateur » ne peut plus être modifié une fois l'utilisateur créé ; l'administrateur définit le nom d'utilisateur lors de la création de l'utilisateur.
- Le champ « Interface de démarrage » permet de définir vers quelle interface vous souhaitez être redirigé après la connexion. Différentes interfaces de démarrage par défaut sont prédéfinies pour différents niveaux d'autorisation ; vous pouvez les modifier ici.
- Le bouton « Modifier le mot de passe » vous permet de modifier le mot de passe de votre compte. Il vous sera demandé de saisir à nouveau votre mot de passe actuel, puis de saisir deux fois votre nouveau mot de passe afin de minimiser les erreurs de saisie accidentelles lors de la définition du mot de passe. Votre mot de passe peut également être modifié par n'importe quel administrateur si vous l'oubliez. Les mots de passe ne sont jamais enregistrés en clair ; par conséquent, ni vous-même ni aucun autre utilisateur ne pouvez les consulter.
- La « photo de profil » peut être modifiée si vous le souhaitez. Une icône d'utilisateur par défaut s'affiche si vous ne définissez pas de photo de profil.

EPSON
Admin Interface

Go to Desktop Interface
Go to Panel Interface
Manage My Account
Language

Manage My Account

safety

Full Name
safety

Username
safety

User Level
Safety Officer

Start-Up Interface
Admin Interface

Change Password

Change Profile Picture

Figure 2.2.3 : Page « Gérer mon compte »

2.2.4 Gérer d'autres comptes utilisateurs

Dans le menu « Autres comptes utilisateurs », vous voyez un aperçu de tous les utilisateurs actuellement disponibles. Vous disposez ici des options suivantes :

- L'ajout d'un nouvel utilisateur vous permet de créer un utilisateur avec un niveau d'autorisation et un mot de passe initial prédéfinis. Le mot de passe initial doit être modifié par chaque utilisateur après sa première connexion.
- La modification d'un utilisateur vous permet de modifier la plupart des propriétés de cet utilisateur, à l'exception du nom d'utilisateur, qui ne peut pas être modifié.
- La suppression d'un utilisateur le supprime définitivement de la base de données. Cet utilisateur ne pourra alors plus déclencher aucune fonction de commande du robot.

EPSON
Admin Interface

Go to Desktop Interface
Go to Panel Interface
Manage My Account
Language

Manage My Account

safety

Account Details

	Full Name	Username	User Level		
	Master of Robots	nerd	Programmer		
	Robot Factory Operator	operator_42	Operator		
	Yet Another Administrator	second_admin	Administrator		

Add new user

Figure 2.2.4 : Page « Gérer d'autres comptes utilisateurs »

2.3 Paramètres système

2.3.1 Paramètres réseau

Le menu « Paramètres réseau » contient tous les paramètres nécessaires à la configuration de la connexion Ethernet du robot. Veuillez noter que vous perdrez la connexion dès que vous modifierez les paramètres de la connexion actuellement active (par exemple, si vous êtes connecté via Ethernet et que vous modifiez l'adresse IP Ethernet). Si vous saisissez par inadvertance des paramètres invalides, vous risquez de ne plus pouvoir vous connecter au robot par la suite. Il est donc fortement recommandé de vous assurer que deux connexions différentes fonctionnent avant de modifier les paramètres de la seule connexion opérationnelle.

Les paramètres Ethernet s'affichent et vous permettent de choisir si le robot utilise une adresse IP dynamique ou manuelle. Si le robot est configuré pour obtenir une adresse IP de manière dynamique, vous verrez également ici l'adresse IP sélectionnée. Si le robot est configuré manuellement, vous devez saisir une adresse IP et le masque de réseau (champ obligatoire) ; vous pouvez en outre définir la passerelle et jusqu'à deux serveurs DNS (facultatif).

The screenshot shows the 'Network Settings' page in the EPSON Admin Interface. The left sidebar contains the EPSON logo and a list of menu items: Admin Interface, Go to Desktop Interface, Go to Panel Interface, Manage My Account, Manage Other Accounts, Network Settings (highlighted), Time Settings, SSL Certificate, Software Update, Help & Support, and Language. The main content area is titled 'Network Settings' and features a 'User' dropdown in the top right. The 'Ethernet IPv4' section includes radio buttons for 'Dynamic IP Address' (selected) and 'Manual IP Address'. Below these are input fields for 'IP Address' (192.168.90.200), 'Netmask' (255.255.254.0), 'Gateway' (192.168.90.1), 'DNS Server 1', and 'DNS Server 2'. A 'Network Information' section contains a note about using the second Ethernet port for maintenance. At the bottom are 'Cancel' and 'Save' buttons.

Ethernet IPv4	
IP Address	192.168.90.200
Dynamic IP Address	☒
Manual IP Address	☐
IP Address	192.168.90.200
Netmask	255.255.254.0
Gateway	192.168.90.1
DNS Server 1	
DNS Server 2	

Network Information

You may use the second Ethernet port to access AX Portal for network maintenance reasons. Its static IP address is 10.0.0.203 and cannot be changed by the administrator. You may connect to '<Robot-Serial>.local' instead of entering the IP address to connect to AX Portal via your browser.

Cancel Save

Figure 2.3.1 : Page « Paramètres réseau »

2.3.2 Paramètres de l'heure

Le menu « Paramètres de l'heure » affiche la date actuelle, l'heure actuelle et le fuseau horaire du système d'exploitation du robot. Veillez à saisir l'heure et la date correctes. Le réglage de l'heure correcte permet d'utiliser le système pour l'analyse des journaux et le dépannage, et garantit le bon fonctionnement des dépendances temporelles. Si vous modifiez l'heure manuellement, l'heure exacte est prise en compte immédiatement après avoir cliqué sur le bouton « Enregistrer ». Si vous souhaitez modifier le fuseau horaire, vous pouvez filtrer la liste des fuseaux horaires disponibles dans le système par continent (par exemple « Europe »), code pays (par exemple « GB »), grande ville (par exemple « London ») ou une combinaison de ces éléments (par exemple « eur lon » → Europe/London).

Remarque : si le robot dispose d'un accès réseau, vous pouvez activer la synchronisation NTP (Network Time Protocol).

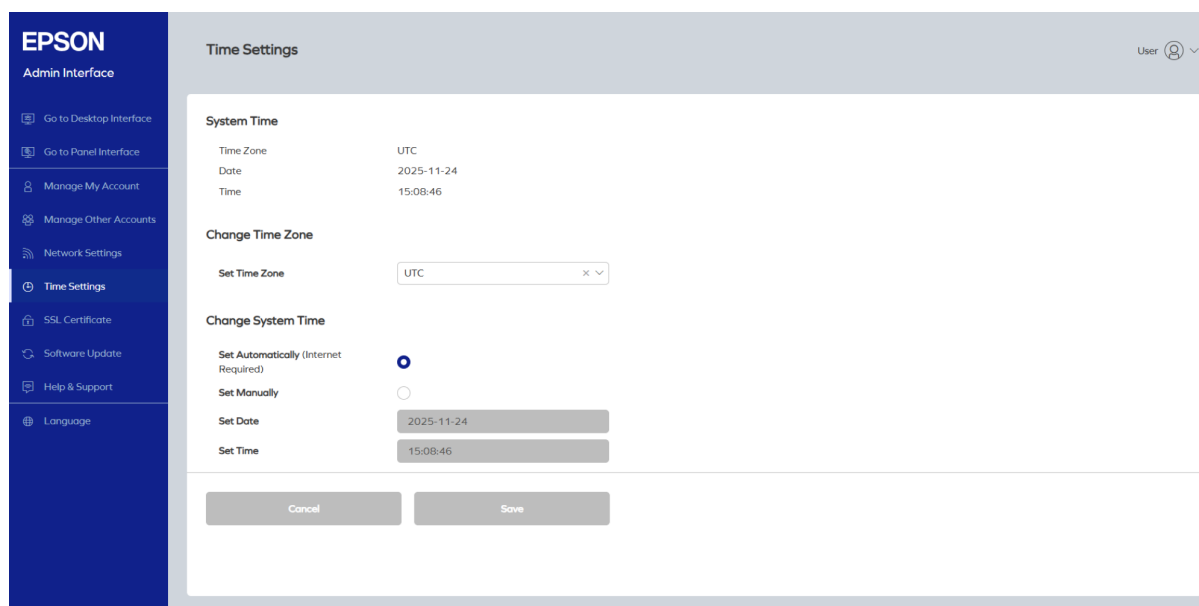


Figure 2.3.2 : Page « Paramètres de l'heure »

2.3.3 Certificat SSL

Le menu « Certificat SSL » affiche quelques détails sur le certificat SSL actuellement chargé et vous permet de remplacer ce certificat par votre propre certificat privé ou professionnel. Un certificat SSL (Secure Sockets Layer) est un certificat numérique qui authentifie l'identité d'un site web et permet une connexion cryptée. Ce certificat garantit que les transactions en ligne et les informations client restent confidentielles et sécurisées dans votre navigateur Web. Par défaut, un certificat auto-signé est installé ; celui-ci doit être remplacé par un certificat adapté à l'environnement de l'utilisateur et signé par une autorité de confiance.

Un certificat est signalé comme non sécurisé par les navigateurs lorsque

- le nom de domaine (ou l'adresse IP) enregistré dans le certificat ne correspond pas au domaine (ou à l'adresse IP) utilisé pour accéder au robot,
- le certificat a expiré ou
- le certificat est auto-signé.

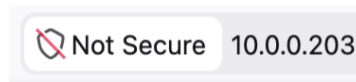


Figure 2.3.3.a : Exemple d'un navigateur signalant un certificat SSL non sécurisé ou manquant

Pour remplacer le certificat SSL, vous devez télécharger à la fois la clé privée et la clé publique.

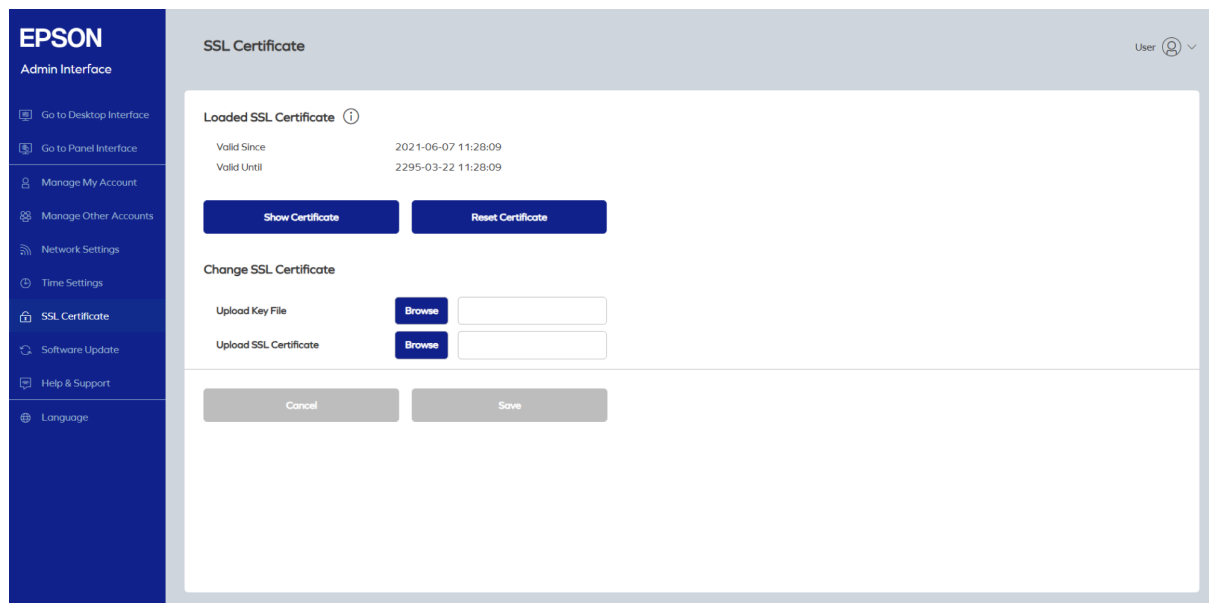


Figure 2.3.3.b : Page « Certificat SSL »

2.4 Mise à jour logicielle

Dans le menu « Mise à jour logicielle », vous pouvez voir la version actuelle d'AX Portal. Vous pouvez y télécharger et installer de nouvelles versions du logiciel. Une mise à jour logicielle peut s'avérer nécessaire pour résoudre un problème spécifique ou pour bénéficier des dernières fonctionnalités et corrections de bugs.

Avant de mettre à jour la version du logiciel, veuillez dans tous les cas lire d'abord les notes de mise à jour, car elles contiennent des informations importantes sur les risques potentiels liés à une mise à jour vers certaines versions. Les améliorations se font parfois au détriment de fonctions plus anciennes, considérées comme obsolètes. Nous vous recommandons de créer une sauvegarde de votre base de données, qui contient toutes les applications et tous les paramètres des robots (via l'interface de bureau), avant de lancer la mise à jour.

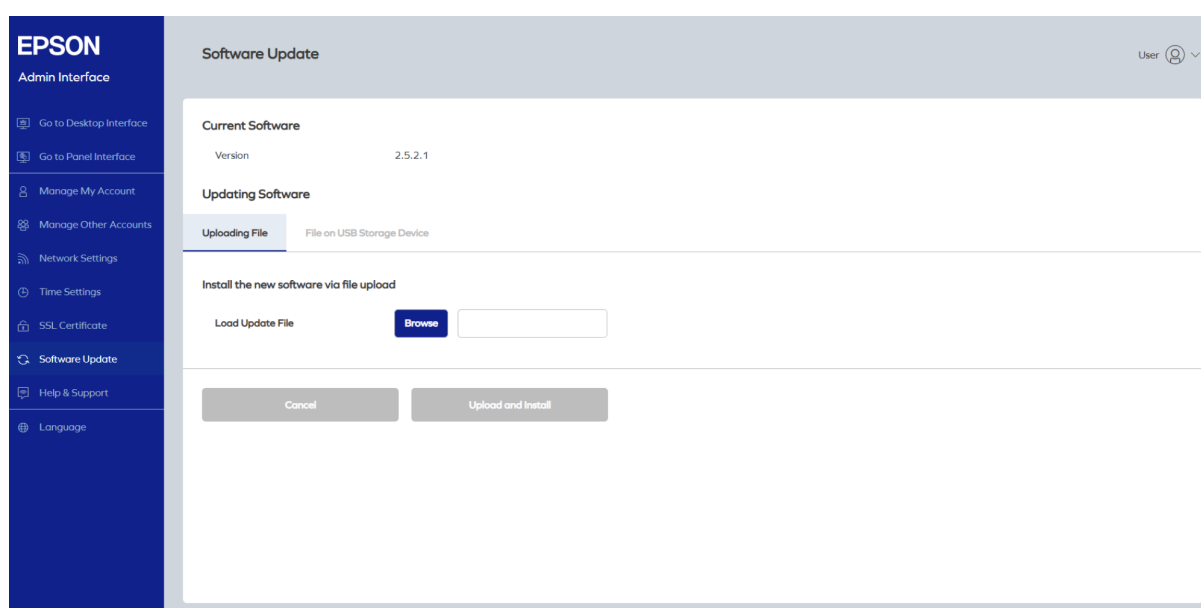


Figure 2.4.a : Page « Mise à jour logicielle »

Pour installer une mise à jour logicielle, vous devez d'abord vous procurer le fichier RBIN contenant la mise à jour pour la version d'AX Portal que vous souhaitez installer. Il existe trois façons d'installer cette mise à jour :

- 1) Sélectionnez le premier onglet « Télécharger un fichier ». Vous pouvez y télécharger la mise à jour depuis votre appareil. Parcourez le système de fichiers et sélectionnez le fichier RBIN de mise à jour, puis poursuivez l'installation de la mise à jour.
- 2) Sélectionnez le deuxième onglet « Fichier depuis un support USB ». Vous pouvez y copier le fichier RBIN sur une clé USB, puis brancher celle-ci sur le contrôleur du robot. Sélectionnez le fichier RBIN de mise à jour dans le répertoire de la clé USB, puis poursuivez l'installation.
Remarque : actuellement, les systèmes de fichiers exFAT et FAT32 sont pris en charge pour la clé USB.
- 3) Si les options 1) et 2) ne sont pas applicables (par exemple, si l'interface d'administration de votre robot ne fonctionne pas correctement), vous avez toujours la possibilité d'exécuter la mise à jour de restauration à la place.

Pendant qu'une mise à jour est en cours, vous pouvez voir quelle étape est en cours d'exécution, quelles étapes ont été menées à bien et, en cas d'erreur, à quelle étape l'erreur s'est produite, accompagnée du message d'erreur. Une mise à jour peut prendre **jusqu'à 15 minutes**. Veuillez donc faire preuve de patience et ne pas interrompre la mise à jour, par exemple en débranchant l'alimentation électrique du robot. Dans le pire des cas, cela pourrait vous empêcher d'accéder au robot par la suite.

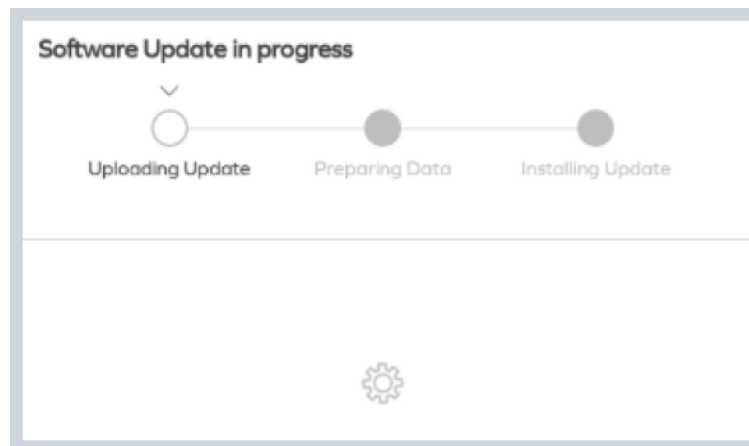


Illustration 2.4.b : Fenêtre « Mise à jour logicielle en cours »

Une fois la mise à jour terminée, celle-ci doit encore être confirmée. Tant que la mise à jour n'est pas confirmée, cela est signalé par un message en rouge. Un redémarrage du contrôleur réinitialise la mise à jour si celle-ci n'a pas été confirmée. Une fois la mise à jour confirmée, il n'est plus possible de revenir à une version antérieure.

2.5 Aide et assistance

Le menu « Aide et assistance » affiche les coordonnées que vous pouvez utiliser pour contacter Epson si vous avez besoin d'aide. Veuillez d'abord consulter la documentation avant de demander de l'aide par e-mail ou par téléphone.

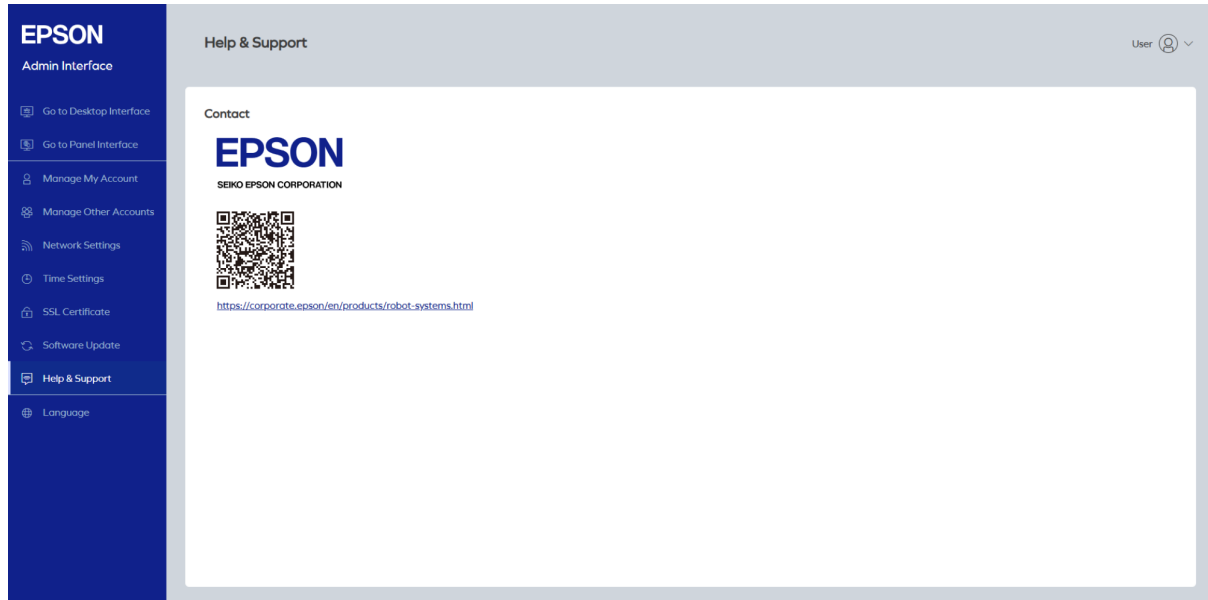


Figure 2.5 : Page « Aide et assistance »

2.6 Authentification sécurisée

Si vous utilisez des modules de connexion pour accéder à AX Portal et contrôler le robot (sans interface graphique), vous devez vous authentifier via une procédure sécurisée en deux étapes avant de pouvoir contrôler le robot.

2.6.1 Exporter le certificat SSL

Dans un premier temps, vous devez envoyer une demande de connexion au serveur d'authentification intégré. Si vous souhaitez télécharger le certificat SSL actuellement chargé, vous pouvez le faire en exportant le certificat via votre navigateur. L'illustration suivante montre comment procéder avec Google Chrome. D'autres navigateurs proposent des options similaires.

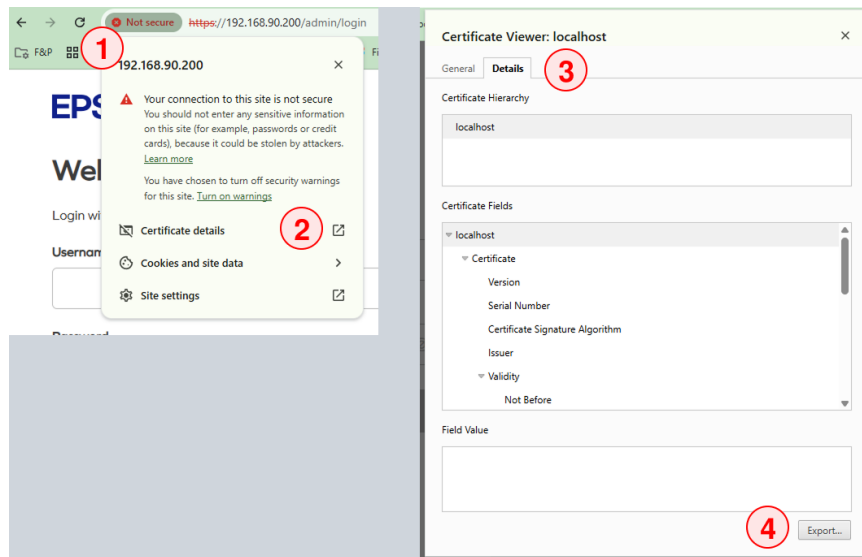


Illustration 2.6.1 : Exportation du certificat SSL

1. Afficher les informations, 2. Accéder aux détails du certificat, 3. Dans la visionneuse de certificats, passer à « Détails », 4. Exporter le certificat

2.6.2 Connexion de l'utilisateur

Dans un premier temps, vous devez envoyer une demande de connexion au serveur d'authentification intégré, accessible via **TCP** sur le **port 5001**. Pour ce faire, vous devez connaître l'**adresse IP** du robot, vos identifiants (**nom d'utilisateur** et **mot de passe**) et utiliser le même **certificat SSL** que le robot. Voici un exemple en Python qui envoie la requête de connexion via un socket TCP et renvoie le jeton d'accès. Vous pouvez également utiliser d'autres langages de programmation pour effectuer cette opération.

Remarque : actuellement, la connexion n'est possible que via TCP. Si vous souhaitez contrôler votre robot via l'un des modules de connexion autres que TCP (par exemple ROS/REST/Modbus), vous devez tout de même utiliser TCP pour obtenir le jeton d'accès. Vous pouvez également désactiver l'authentification pour ces modules de connexion dans les paramètres d'AX Portal. Attention : si vous désactivez l'authentification, n'importe quel client pourra contrôler le robot via ROS/REST sans authentification préalable !

```
import json, socket, ssl

# define the robot ip address, login credentials, and path to SSL certificate
ip_address = "10.0.0.203"
username = "user_xyz"
password = "my_fancy_password"
session_id = "my_session_id"
ssl_cert = "/home/user/robot-ssl/cert.pem"

# create request to send
request = {"request": "authentication", "action": "login", "data":
          {"username": username, "password": password, "session_id": session_id}}

# prepare SSL context to use with the connection
ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLS_CLIENT)
ssl_context.load_verify_locations(ssl_cert)
ssl_context.check_hostname = False
ssl_context.verify_mode = ssl.CERT_NONE

with socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0) as sock:
    with ssl_context.wrap_socket(sock, server_hostname=ip_address) as ssock:
        # connect to the authentication socket
        ssock.connect((ip_address, 5001))
        # send request (and end character)
        message = json.dumps(request) + "\x03"
        ssock.send(message.encode("UTF-8"))
        # receive answer (before end character)
        message = ssock.recv(1024).decode("UTF-8")
        answer = json.loads(message.split("\x03")[0])

token = answer["response"]["data"]["token"]
session_id = answer["response"]["data"]["session_id"]
```

2.6.3 Enregistrement de la connexion

Dès que vous disposez d'un **jeton d'accès valide**, vous pouvez l'utiliser dans la fonction « token_login », dont l'utilisation est décrite en détail dans le chapitre consacré au module de connexion correspondant.

3. Exportation USB et mise à jour de restauration

3.1 Aperçu

Si l'interface d'administration n'est pas accessible alors que le contrôleur fonctionne, vous pouvez essayer de restaurer le contrôleur à l'aide de la méthode de mise à jour de restauration.

3.2 Exportation USB

Le journal système et la base de données AX Portal peuvent être exportés à l'aide d'une clé USB sans avoir accès à l'interface Web :

1. Formatez une clé USB en exFAT, FAT32 ou NTFS et assurez-vous qu'elle ne porte PAS le nom de système de fichiers LOG ou RECOVERY
2. Insérez-la dans le contrôleur et patientez quelques minutes.
3. Retirez la clé USB. Vous devriez trouver dans le répertoire racine de la clé un fichier ZIP contenant la base de données et les journaux système pertinents.

3.3 Exécuter la mise à jour de restauration

Pour exécuter la mise à jour logicielle de restauration, veuillez procéder comme suit :

1. Procurez-vous le **fichier de mise à jour**, qui est généralement au format RBIN. Si vous ne disposez pas de fichier de mise à jour, veuillez contacter le fabricant.
2. Assurez-vous d'avoir à portée de main une **clé USB** que vous pouvez utiliser pour la mise à jour. Elle doit être formatée en exFAT, FAT32 ou NTFS, et le système de fichiers de la première partition doit être nommé « RECOVERY ».
3. Copiez le fichier de mise à jour dans le répertoire racine de la clé USB.
4. Éteignez le robot et redémarrez-le. Attendez **au moins 1 minute** jusqu'à ce que le robot ait correctement démarré.
5. Insérez la clé USB préparée dans n'importe quel port USB du contrôleur.
6. Attendez que la mise à jour soit terminée. Notez que cela peut prendre **jusqu'à 15 minutes**. Essayez d'accéder à l'interface d'administration et vérifiez si la nouvelle version a été installée.
7. **Important :** retirez la clé USB du contrôleur. Si vous oubliez cette étape, la mise à jour sera réexécutée après le redémarrage, ce que vous ne souhaitez pas.

4. Interface de bureau (GUI)

4.1 Introduction

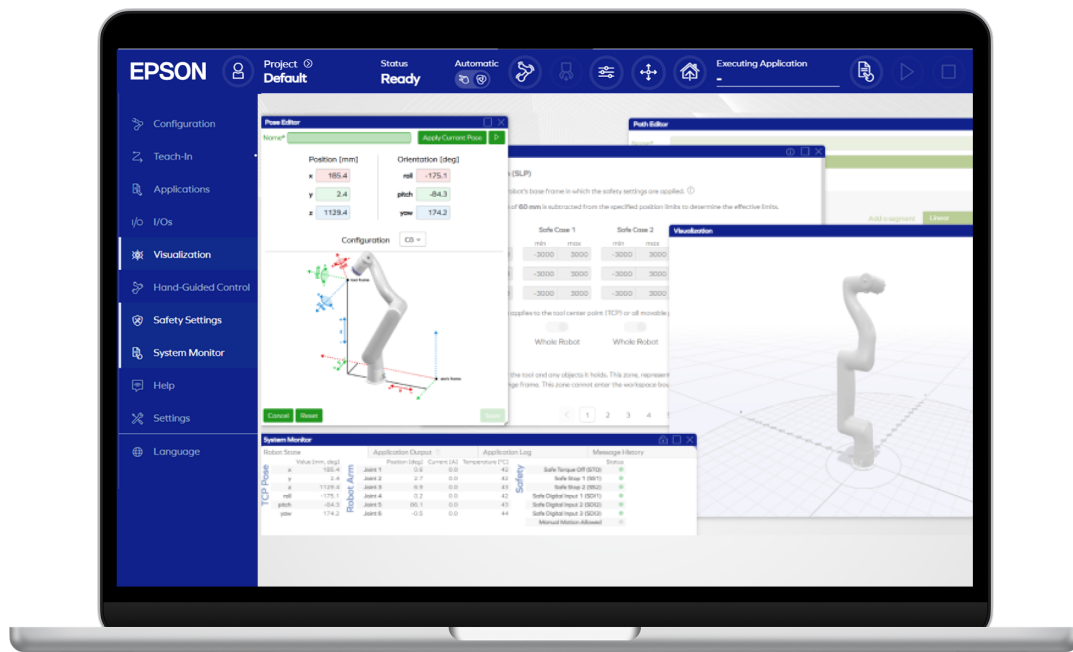


Figure 4.1.a : Exemple de capture d'écran de l'interface de bureau

L'interface de bureau d'AX Portal est l'interface utilisateur graphique (GUI) principale par laquelle l'utilisateur commande le robot, programme des applications, modifie des éléments de la base de données et visualise l'état actuel du robot. Cette GUI a été conçue pour être intuitive et largement intuitive, même pour les utilisateurs n'ayant pas une grande expérience en matière de commande de robots. Vous pouvez accéder à cette interface via l'interface d'administration en vous connectant et en cliquant sur le bouton « Passer à l'interface de bureau ».

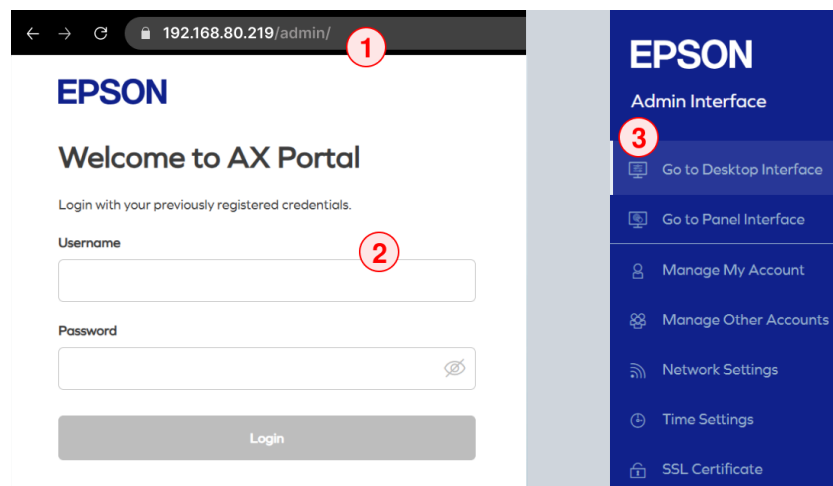


Figure 4.1.b : Accès à l'interface de bureau

1. Saisir l'adresse IP dans l'URL, 2. Saisir les identifiants de connexion, 3. Passer à l'interface de bureau

L'interface de bureau contient toutes les fonctions de commande disponibles pour le robot ainsi que des fonctions supplémentaires permettant d'observer et de modifier son état. Pour exécuter facilement des applications programmées sans avoir à modifier les applications ou les paramètres, nous vous recommandons d'utiliser l'interface du panneau ("Panel Interface") . Ce chapitre explique en détail toutes les fenêtres, tous les menus et tous les éléments de l'interface de bureau.

Dans certains navigateurs (notamment sous Windows), le zoom d'affichage est réglé par défaut sur une valeur élevée (par exemple 150 %), ce qui peut entraîner l'apparition de parties de l'interface (telles que le cockpit ou le logo) coupées. Pour un affichage optimal, nous vous recommandons de régler le zoom d'affichage sur 100 % (ou moins).

4.2 Fenêtre principale

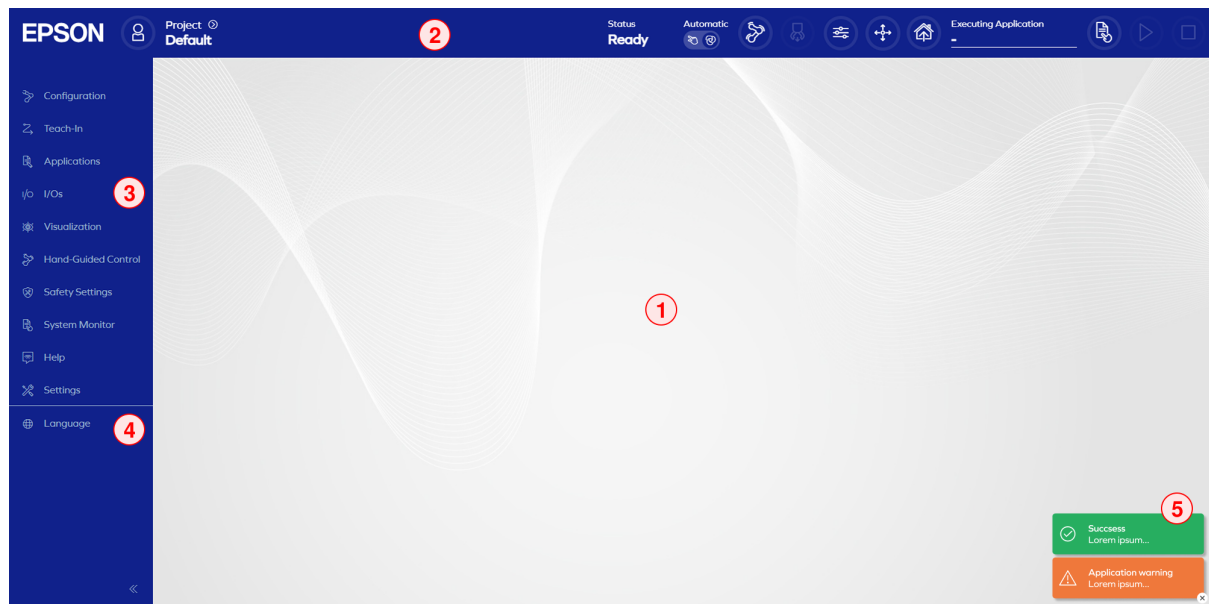


Figure 4.2 : Éléments de l'interface du bureau

1. Espace de travail, 2. Cockpit, 3. Menus, 4. Changer de langue, 5. Messages contextuels

Dès que vous avez connecté votre ordinateur ou votre appareil à l'interface de bureau, la fenêtre principale s'affiche dans votre navigateur Web.

Cette fenêtre principale contient les éléments suivants :

- un cockpit tout en haut de la fenêtre, qui affiche l'état actuel du programme, les messages et l'accès rapide aux commandes du robot,
- les boutons de menu situés à gauche de la fenêtre, sur lesquels vous pouvez cliquer pour naviguer parmi les menus disponibles (tous expliqués plus loin dans ce chapitre),
- le numéro de version dans le coin inférieur droit,
- un espace réservé aux fenêtres contextuelles de messages sur le côté droit de l'écran et
- la zone de travail principale, dans laquelle l'utilisateur peut déplacer et organiser les menus.

4.2.1 Aperçu des menus

Sur le côté gauche de la fenêtre principale, vous trouverez des boutons permettant d'accéder aux différents menus d'AX Portal. La fonction précise et les informations détaillées relatives à chaque menu disponible sont décrites dans les sections suivantes. Voici un bref aperçu des menus disponibles :

	Dans le menu « Configuration », l'utilisateur peut configurer et mettre en marche le robot.
	Dans le menu « Apprentissage », l'utilisateur peut programmer des poses, des trajectoires et des grilles pour le robot.



Dans le menu « **Applications** », l'utilisateur peut écrire, modifier et tester des applications personnalisées.



Dans le menu « **E/S** », l'utilisateur peut surveiller les E/S et modifier les signaux de sortie.



Le menu « **Visualisation** » affiche le mouvement actuel du robot.



Dans le menu « **Commande manuelle** », l'utilisateur peut déplacer manuellement les articulations du robot.



Dans le menu « **Paramètres de sécurité** », le responsable de la sécurité peut modifier la configuration de sécurité.



Dans le menu « **Moniteur système** », l'utilisateur peut surveiller les sorties du système.



Le menu « **Aide** » contient des informations utiles sur le programme et le système.



Dans le menu « **Paramètres** », l'utilisateur peut définir certains paramètres de base.



Le bouton « **Langue** » permet à l'utilisateur de changer la langue d'affichage des interfaces utilisateur.

4.2.2 Cockpit

Le Cockpit d'AX Portal est toujours visible en haut de l'interface graphique et contient les informations et les commandes les plus importantes. Tous les éléments du Cockpit sont expliqués en détail ci-dessous, de gauche à droite :

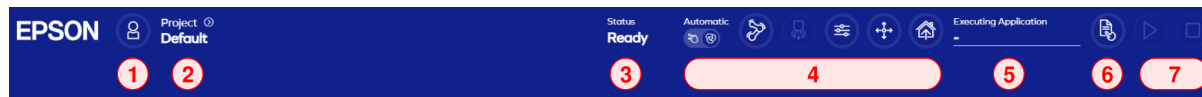


Figure 4.2.2 : Cockpit

1. Options du compte utilisateur, 2. Sélection de projet, 3. État du robot, 4. Commandes du robot, 5. Applications en cours d'exécution ou sélectionnées, 6. Sélection d'application, 7. Commande principale



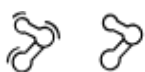
Appuyez sur le bouton « **Utilisateur** » pour modifier les paramètres de votre compte ou vous déconnecter. Ces options vous mènent d'une manière ou d'une autre à l'interface d'administration.

Appuyez sur le bouton « **Projet** » pour accéder au menu des projets, dans lequel vous pouvez changer de projet et consulter l'aperçu des projets. Sous ce bouton, vous voyez le nom du projet actif.

Le bouton « **Demander le contrôle** » ne s'affiche que si cette interface n'est pas le contrôle actif du robot. Pour en savoir plus sur cette fonction de sécurité, consultez la section Contrôle actif.

La barre de messages d'état affiche l'état actuel du programme. L'état du programme détermine essentiellement si une commande ou une action, telle qu'une pression sur une touche, est exécutable ou non.

Le curseur de **mode de fonctionnement** permet au robot de basculer entre le mode manuel et le mode automatique. Pour en savoir plus sur cette fonction de sécurité, consultez la section Modes de fonctionnement.



Le bouton « **Activer/désactiver la commande manuelle** » permet à l'utilisateur de passer en mode manuel. Lorsque cette fonction est activée, l'état passe à « Commande manuelle » et l'utilisateur peut diriger le robot manuellement. Le robot ne peut passer en mode manuel que si l'état du programme est « Prêt ». Lors de la désactivation, l'état repasse à « Prêt ».



Le bouton « **Outil** » permet de basculer entre les fonctions « pick » et « place » de l'outil monté. Dans l'éditeur d'outils, l'utilisateur peut associer ces fonctions à des entrées numériques. L'outil ne peut être déclenché que si le comportement est défini et si l'état du programme est « Prêt » ou « Commande manuelle ».



Le bouton « **Commande fine** » ouvre une fenêtre permettant de régler avec précision les angles d'articulation et les coordonnées du robot. L'utilisateur peut y modifier les coordonnées individuelles du robot ou les angles d'articulation.



Le bouton « **Commande de mouvement** » ouvre une fenêtre permettant de contrôler les mouvements. L'utilisateur peut y régler la vitesse du robot, modifier les limites globales de vitesse et d'accélération, ainsi que changer de système de référence.



Le bouton « **Retour à la position d'origine** » permet à l'utilisateur de déplacer le robot vers sa position d'origine. Cela sert de raccourci, par exemple pour se placer en position de départ avant d'exécuter une application ou en position de repos avant de mettre le robot hors tension. La position d'origine peut être modifiée dans le menu des paramètres.

La zone « **Sélectionné actuellement** » indique quel élément (application, trajectoire, pose, etc.) est actuellement associé aux boutons de commande (exécuter, pause, reprendre, arrêter). Ces informations changent dès que vous cliquez sur un bouton d'exécution ou que vous sélectionnez une application.



Dans le menu « **Sélection d'application** », l'utilisateur peut sélectionner une application préalablement créée à l'aide de l'éditeur d'applications.



Le bouton « **Exécuter** » exécute l'application sélectionnée. L'état du programme doit être « Prêt » pour pouvoir lancer une application. Le bouton « **Pause** » met en pause une application en cours d'exécution. Si une application est mise en pause pendant un mouvement du robot, celui-ci s'arrête en douceur. Le bouton « **Reprendre** » permet de reprendre une application mise en pause. Si l'application a été mise en pause pendant un mouvement, ce mouvement reprend en douceur. Le bouton « **Arrêter** » permet d'arrêter une application en cours d'exécution ou en pause. Si une application est arrêtée pendant un mouvement, le robot s'arrête immédiatement et conserve sa position. Le statut repasse alors à « Prêt ».

4.2.3 Commande active (Commande exclusive depuis un poste de commande)

Afin de garantir que le robot ne puisse être commandé qu'à partir d'un seul poste de commande à tout moment, les interfaces sont protégées. Lorsque vous vous connectez en tant que premier utilisateur, vous obtenez automatiquement le contrôle. C'est ce que nous appelons le contrôle actif. Si, par la suite, une autre personne (ou vous-même) se connecte via un autre canal de connexion (il peut s'agir d'une autre fenêtre de navigateur), cette connexion ne dispose d'aucun contrôle et n'a accès qu'à la surveillance et à l'observation du robot (accès en lecture seule). C'est ce que nous appelons l'observateur passif.

4.2.3.1 Contrôle actif

Le poste de contrôle actif peut partager son contrôle. Dans l'interface, cela s'effectue en cliquant sur le bouton « Utilisateurs » et en sélectionnant « Partager les droits de contrôle ». Si vous avez tendance à travailler avec plusieurs utilisateurs ou interfaces sur un même robot, il est recommandé de toujours partager l'accès au contrôle et/ou de fermer le canal de connexion.

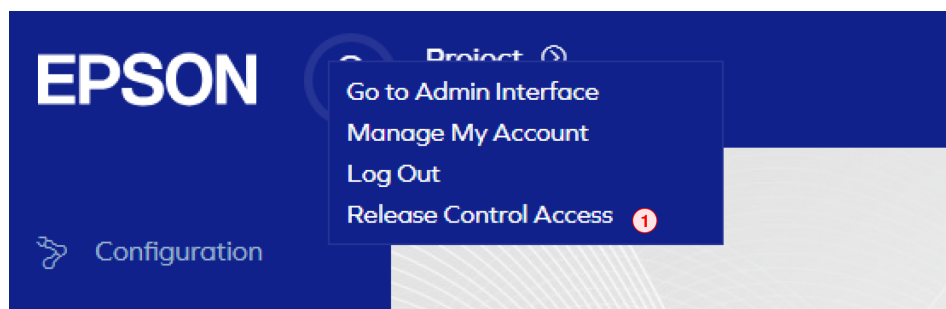


Figure 4.2.3.1 Partage de l'accès au contrôle
1. Partager l'accès au contrôle

4.2.3.2 Observateur passif

L'observateur passif peut demander le contrôle. Dans l'interface, cela s'effectue en cliquant sur le bouton « Utilisateur » puis en sélectionnant « Demander l'accès au contrôle », ou simplement en cliquant sur le grand bouton « Demander le contrôle » qui apparaît dans le cockpit. Une fois le contrôle demandé, une boîte de dialogue demandant l'accès s'affiche à la fois sur le poste de contrôle actif et sur l'observateur passif. Alors que l'observateur passif ne peut qu'attendre que l'accès au contrôle lui soit accordé, le poste de contrôle actif peut soit accepter, soit refuser la demande.

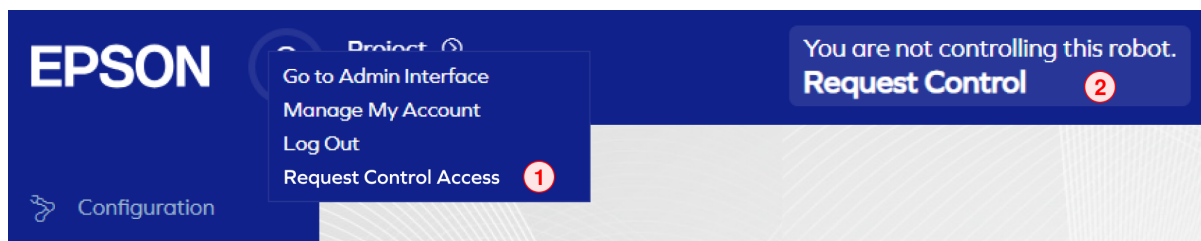


Figure 4.2.3.2 Demande d'accès au contrôle

1. Demander l'accès au contrôle, 2. Bouton « Demander le contrôle »

4.2.4 État du programme

L'état actuel du programme et/ou du robot est visible dans le cockpit. L'état fournit des informations sur ce que font actuellement AX Portal et/ou le robot, et détermine essentiellement quelles actions ou commandes peuvent être exécutées à ce moment-là. Chaque état possède ses propres caractéristiques, et certaines fonctions ne sont disponibles que dans certains états du programme. Voici une liste des différents états et leurs explications :

Hors service	Les moteurs ne sont pas alimentés et les appareils ne sont pas connectés. Cet état est utilisé pour configurer le robot avant sa mise sous tension. En utilisant le curseur de mise sous tension, l'état passe à « Connecter ». D'autres actions peuvent faire passer l'état à « Chargement », notamment lorsqu'une action ne doit pas être interrompue et nécessite un certain temps pour être menée à bien.
Chargement	Le robot traite actuellement des données, par exemple il charge une base de données ou des paramètres. Une fois les données chargées, l'état repasse à « Hors service ».
Connexion	Le robot met actuellement ses moteurs sous tension et connecte ses appareils. Une fois la connexion établie, le robot commence à contrôler activement ses moteurs et l'état passe à « Prêt ».
Prêt	Le programme est prêt à recevoir des commandes et/ou à exécuter des actions, telles que l'exécution d'une application, la préhension et le dépôt d'objets, etc. Il s'agit de l'état de veille du robot.
En cours	Une application/un mouvement est en cours d'exécution. Cette application/ce mouvement peut être mis en pause ou interrompu à l'aide du bouton « Pause » ou « Arrêt » correspondant. Une fois que l'application a atteint la fin de son code ou que le mouvement en cours est terminé, l'état revient à « Prêt ».
En pause	Une application/un mouvement est actuellement en pause. Appuyez sur le bouton « Continuer » pour reprendre, ce qui ramène l'état à « En cours d'exécution », ou appuyez sur le bouton « Arrêter » pour annuler, ce qui ramène l'état à « Prêt ».

Commande manuelle	La commande manuelle (HGC) est activée pour une ou plusieurs articulations du robot. Dans ce mode de fonctionnement, vous pouvez guider manuellement le robot dans n'importe quelle position en exerçant une force sur ses articulations. Simultanément, le robot maintient son propre poids en compensant activement la gravité. Notez que vous devez appuyer sur le bouton HGC ou actionner le bouton de validation pour déplacer le robot. Désactivez à nouveau le mode de commande manuelle pour rétablir l'état « Prêt ».
En cours	Une action longue, telle que l'enregistrement/le chargement de données ou le calcul d'un chemin, est en cours. Une fois cette action traitée avec succès, l'état revient à sa valeur précédente.
Arrêté	Normalement, le statut « Arrêt » revient automatiquement et très rapidement à « Prêt ». Si ce n'est pas le cas, une partie d'AX Portal n'a pas pu être exécutée correctement. Malheureusement, cela ne peut être résolu qu'en redémarrant le robot.
Arrêt d'urgence Arrêt normal Arrêt de sécurité Collision Arrêt de sécurité Limitation de position Limite de position	Ces états correspondent à différentes interruptions. Lorsqu'une interruption est déclenchée, l'état correspondant s'affiche et la fenêtre d'interruption s'ouvre pour permettre de traiter l'interruption. Dès que l'interruption est résolue, l'état passe généralement à « En cours de reprise ».
Reprise	Une fois l'interruption résolue, le statut passe à « En cours de reprise ». Cela nécessite que l'utilisateur confirme manuellement qu'il est sûr de poursuivre l'application ou le mouvement du robot.
Erreur	Une erreur s'est produite qui n'a pas pu être résolue automatiquement ou qui nécessite une attention particulière de la part de l'utilisateur. Consultez le message d'erreur et essayez de résoudre l'erreur si possible.
Déconnexion	Le robot est en train d'arrêter ses moteurs, de déconnecter ses périphériques et de fermer tous les modules en cours d'exécution. Le statut passe ensuite à « Hors service ».

4.2.5 Modes de fonctionnement

Le robot de la série AX dispose de deux modes de fonctionnement : le mode manuel et le mode automatique. Le mode de fonctionnement peut être sélectionné à l'aide du curseur correspondant dans le cockpit.

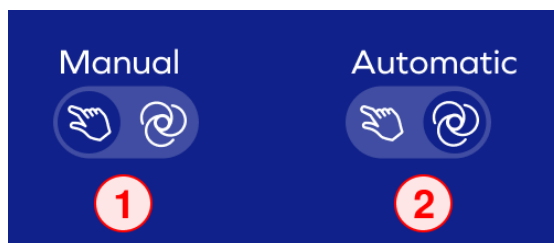


Figure 4.2.5 Commutateur de sélection du mode de fonctionnement

 AVERTISSEMENT	Le mode automatique ne doit être utilisé que si le système est équipé des fonctions de sécurité requises, a été contrôlé et si sa sécurité a été confirmée par une évaluation des risques. L'utilisation du mode automatique dans d'autres situations est très dangereuse. Vous trouverez de plus amples informations dans le « AX6 & RC-A101 - Manuel de Sécurité ».
---	---

4.2.5.1 Mode automatique

Le mode automatique est généralement utilisé en fonctionnement normal, c'est-à-dire lorsqu'il a été vérifié que les applications fonctionnent correctement. Ce mode de fonctionnement permet d'utiliser toute la plage de vitesse du robot.

4.2.5.2 Mode manuel

Le mode manuel est une fonction de sécurité et est généralement utilisé lors de la modification des applications du robot, en particulier lors de l'apprentissage de nouvelles positions ou lors du test de modifications apportées aux applications. En mode manuel, le robot se déplace à vitesse réduite (max. 250 mm/s) et uniquement lorsque **l'interrupteur d'autorisation** est actionné. Il s'agit d'un interrupteur de sécurité à 3 positions. Si l'interrupteur d'autorisation n'est pas actionné, l'utilisateur est informé par un avertissement indiquant que les mouvements ne sont pas autorisés.



Figure 4.2.5.2 Exemple d'un interrupteur d'autorisation

4.2.6 Commandes de précision

Appuyez sur le bouton de réglage fin dans le cockpit pour ouvrir les commandes de réglage fin. Ici, le robot peut être réglé avec une précision allant jusqu'à 0,1 [mm]/[°]. Cette fenêtre comporte deux onglets. Le premier onglet permet à l'utilisateur de modifier les coordonnées euclidiennes actuelles du TCP (Tool Center Point), qui correspondent au système de référence de l'outil actuel exprimé dans le système de référence de travail. Le deuxième onglet permet à l'utilisateur de modifier les angles d'articulation actuels du robot.

Ces onglets disposent des fonctions suivantes :

Champs de saisie	La saisie d'un nombre dans le champ de saisie permet de modifier une coordonnée ou une valeur d'articulation.
Curseur	Le curseur définit à la fois le pas , qui détermine la distance parcourue par le robot lors d'un clic court sur un bouton de commande, et la vitesse de déplacement maximale , qui détermine la rapidité avec laquelle le robot se déplace à l'aide de ces commandes.
Boutons	Une pression brève sur un bouton entraîne un déplacement de la valeur de pas sélectionnée dans la direction indiquée par le bouton. Une pression longue sur un bouton, en revanche, déclenche un mouvement continu dans la direction indiquée.
Configuration	Une TCP donnée peut être atteinte par jusqu'à huit combinaisons d'angles d'articulation. Une combinaison d'angles d'articulation est appelée « configuration » de cette pose. En modifiant la valeur « Configuration » dans la liste déroulante, l'utilisateur peut sélectionner une configuration spécifique pour la pose actuelle. Et en appuyant sur le bouton « Configuration suivante », l'utilisateur peut parcourir les configurations disponibles de la pose actuelle. Notez que la modification de la configuration n'est pas un ajustement fin ; elle entraîne le déplacement de toutes les articulations.
Enregistrer comme pose	En appuyant sur le bouton « Enregistrer comme pose », cette pose est enregistrée sous un nouveau nom unique. Toutes les poses disponibles sont répertoriées dans le menu « Apprentissage ».

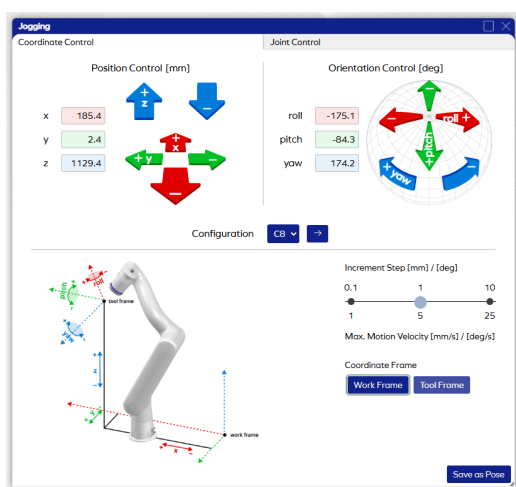


Figure 4.2.6.a : Commande de précision –
Commande par coordonnées

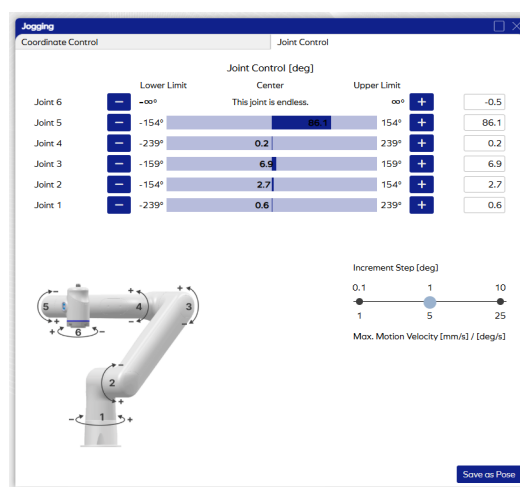


Figure 4.2.6.b : Commande de précision –
Commande des articulations

4.2.7 Commande de mouvement

Appuyez sur le bouton « Commande de mouvement » dans le cockpit pour ouvrir la fenêtre de commande de mouvement. Cette fenêtre comporte les trois onglets suivants :

4.2.7.1 Limites de mouvement

L'onglet « Limites de mouvement » contient des paramètres permettant de limiter la vitesse maximale du robot et de réduire la vitesse de tous les mouvements du robot à l'aide d'un facteur d'échelle. Ces commandes sont utiles lors de la mise en service d'applications robotiques, par exemple pour limiter la vitesse lors de la validation d'une nouvelle application ou lors de la modification de mouvements dans des applications existantes.

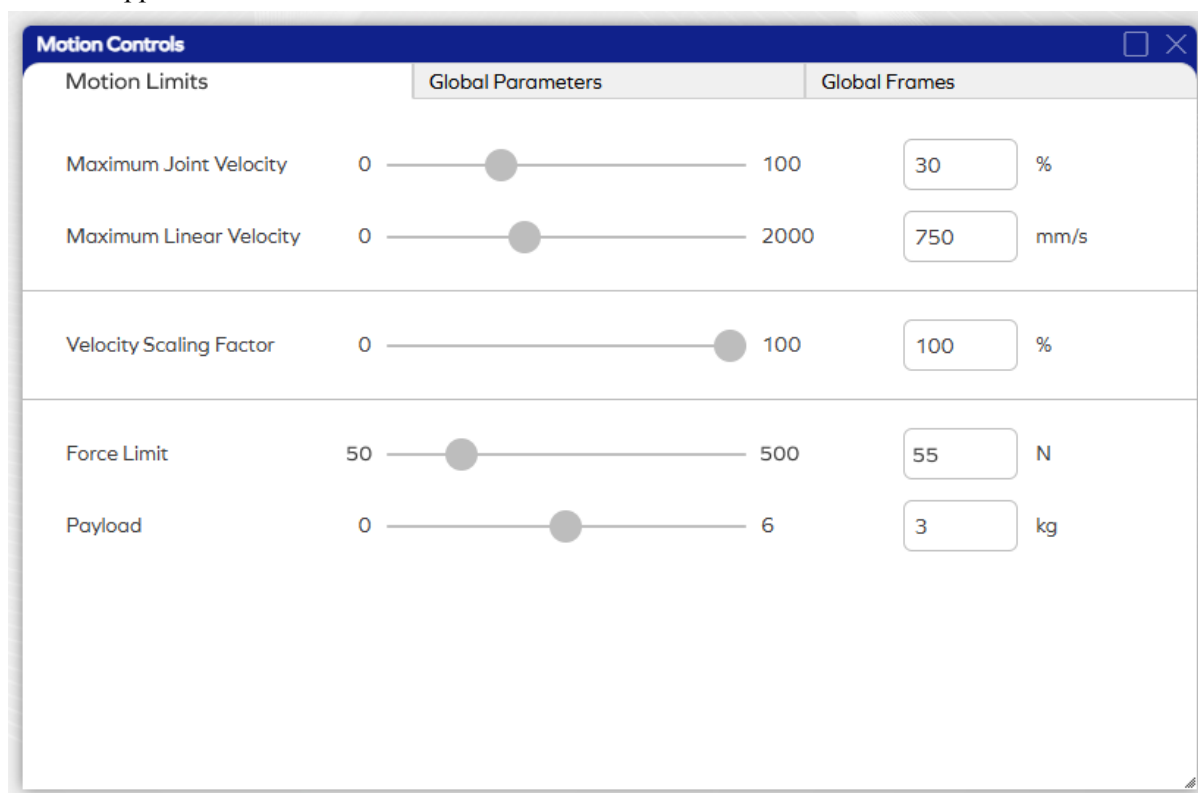


Figure 4.2.7.1 : Contrôle des mouvements – Onglet « Limites de mouvement »

La limite de force et la charge utile, quant à elles, servent à régler la sensibilité aux collisions du robot. La limite de force définit une limite supérieure pour les forces externes admissibles sur l'outil, et la charge utile définit le poids total supporté par le robot (y compris son outil).

4.2.7.2 Paramètres globaux

L'onglet « Paramètres globaux » contient des paramètres permettant de définir la vitesse et l'accélération par défaut pour différentes commandes de mouvement. Les commandes de mouvement sont classées dans les catégories suivantes :

- Les commandes permettant de déplacer le robot de manière optimale en termes de temps utilisent les arguments « joint_velocity » et « joint_acceleration », dont les valeurs par défaut sont définies ici comme **vitesse globale des articulations** et **accélération globale des articulations**.
- Les commandes permettant de déplacer le robot dans une direction ou selon une forme spécifique (linéaire, circulaire) utilisent les arguments « linear_velocity » et « linear_acceleration », dont les valeurs par défaut sont définies ici comme **vitesse linéaire globale** et **accélération linéaire globale**.
- Les commandes permettant de déplacer le robot de manière à ce que le TCP (Tool Center Point) reste à la même position et ne change que d'orientation utilisent les arguments « angular_velocity » et « angular_acceleration », dont les valeurs par défaut sont ici définies comme **vitesse de rotation globale** et **accélération de rotation globale**.
- Les commandes permettant de se déplacer le long d'un chemin comportant plusieurs segments de mouvement utilisent l'argument « blend_radius », dont la valeur par défaut est ici définie comme « **rayon de transition global** ». Ce paramètre sert à lisser l'interpolation entre les segments du chemin.

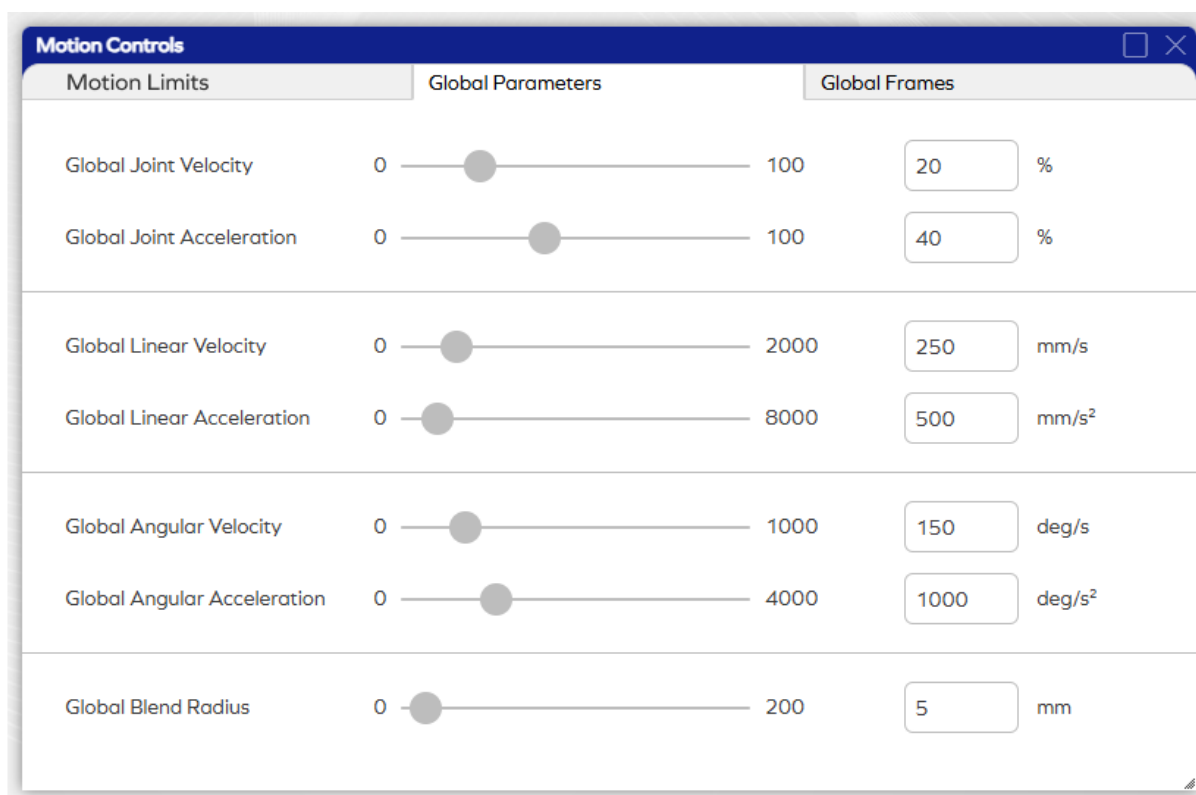


Figure 4.2.7.2 : Contrôle du mouvement – Onglet « Paramètres globaux »

4.2.7.3 Systèmes de coordonnées globaux

L'onglet « Systèmes de coordonnées globaux » contient des paramètres permettant de définir les systèmes de coordonnées de référence par défaut du robot. Alors que le système de base et le système de bride sont fixés au robot, le système de travail et le système d'outil peuvent être définis par rapport à ces systèmes de référence fixes :

- Le **système de base** est fixé au centre de la surface de montage inférieure du robot, de sorte que l'axe z soit orienté vers le bras du robot et que l'axe x soit orienté à l'opposé des câbles raccordés.
- Le **système de bride** est solidement fixé au centre de la plaque de montage supérieure du robot. Son orientation est alignée sur le système de base lorsque toutes les articulations du robot sont à zéro degré.
- Le **système de travail** est le système de référence dans lequel l'utilisateur exprime les coordonnées du TCP (Tool Center Point) dans les applications. Par défaut, le système de travail correspond au système de base du robot. De nombreuses commandes de mouvement utilisent l'argument « work_frame », dont la valeur par défaut est définie ici comme **système de travail global**.
- Le **système d'outil** définit la position et l'orientation du TCP (Tool Center Point), exprimées dans le système de bride. Par défaut, le système de travail correspond au système de bride du robot. De nombreuses commandes de mouvement utilisent l'argument « tool_frame », dont la valeur par défaut est définie ici comme **système d'outil global**.

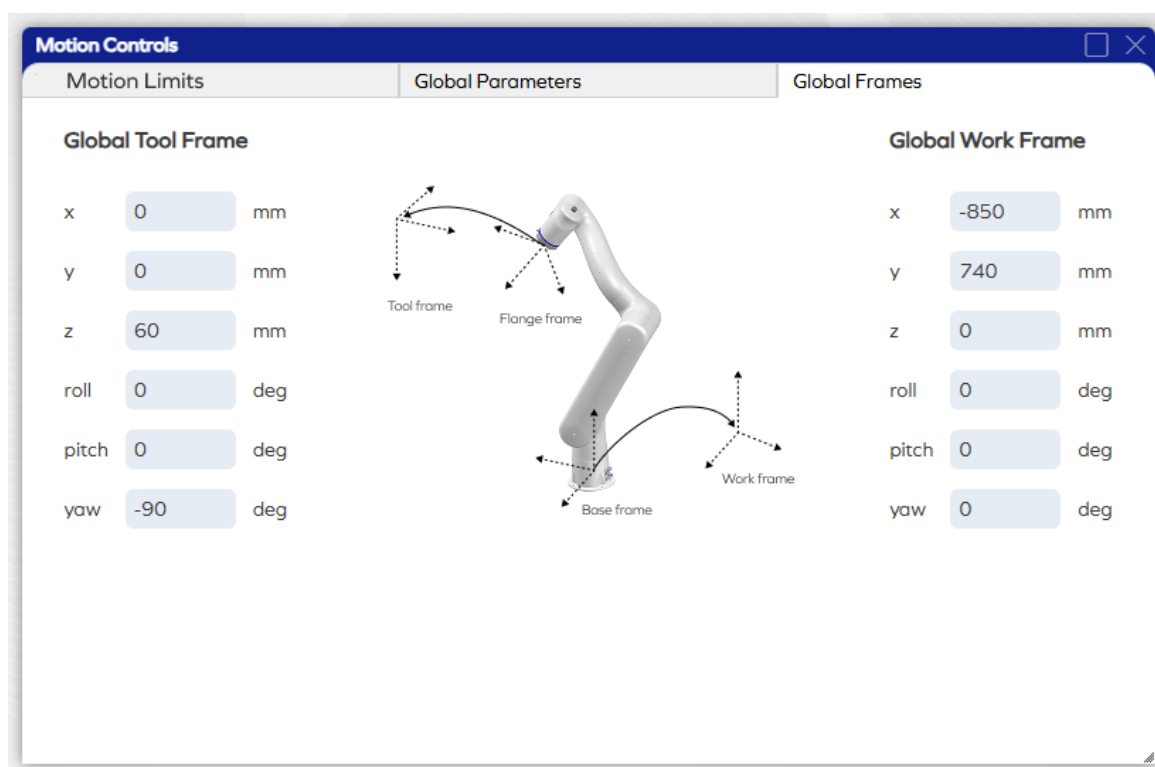


Figure 4.2.7.3 : Commande de mouvement – Onglet « Systèmes de coordonnées globaux »

4.3 Blocs fonctionnels et base de données

Afin de permettre une programmation d'applications simple et intuitive, AX Portal propose une structure orientée objet composée de différents blocs. Ces blocs (ci-après dénommés « éléments ») peuvent être créés individuellement dans divers éditeurs accessibles via l'interface de bureau. AX Portal permet ensuite à l'utilisateur de relier intelligemment ces éléments entre eux afin de créer des applications orientées objet, ce qui aboutit à un code d'application concis, lisible et ne contenant généralement pas beaucoup de chiffres ni de coordonnées.

Voici une liste des éléments fournis :

	Les applications contiennent principalement le code de programme (et/ou les blocs) à exécuter ainsi que des paramètres spécifiques à l'application. Le code de programme prend en charge Python 3.13 et comprend, outre les fonctions Python de base, de nombreuses fonctions issues de l'API (interface de programmation d'application) interne de commande des robots. Les applications sont créées et modifiées à l'aide de l'éditeur d'applications.
	Les poses sont des positions du point central de l'outil (TCP) avec des orientations spécifiques, clairement définies par une combinaison d'angles d'articulation. Les poses sont créées et modifiées à l'aide de l'éditeur de poses.
	Les trajectoires sont des séquences de poses qui décrivent les mouvements fluides du TCP et, par conséquent, les mouvements de l'ensemble du robot. Les trajectoires sont créées et modifiées à l'aide de l'éditeur de trajectoires.
	Les grilles sont des arrangements de points personnalisables, pouvant aller jusqu'à trois dimensions, qui définissent par exemple les positions de plateaux prédéfinis. Les points de grille sont extraits et utilisés comme itérateurs dans les applications. Les grilles sont créées et modifiées à l'aide de l'éditeur de grilles.
	Les E/S sont divisées en entrées et sorties numériques et analogiques. Elles peuvent être configurées via le menu Configuration des E/S, et la valeur des E/S est visible dans le menu E/S. Les E/S servent à la connexion et à la communication avec des appareils externes.

Tous ces éléments sont enregistrés dans la base de données d'AX Portal. Vous pouvez à tout moment télécharger la base de données de votre robot (pour disposer d'une sauvegarde de l'ensemble de votre configuration) ou en téléverser une nouvelle (ce qui écrasera la base de données actuelle et tous ses éléments) via le menu « Configuration ». De plus, le menu « Projet » offre un aperçu clair de tous les projets et éléments enregistrés dans votre base de données.

4.4 Menu Projet

Cliquez sur le nom du projet actuellement sélectionné dans le Cockpit pour accéder au menu Projet. Ce menu contient un aperçu des éléments disponibles tels que les poses, les trajectoires, les grilles et les applications. Ces éléments sont regroupés en plusieurs projets. Un seul projet est sélectionné à la fois, et seuls les éléments du projet sélectionné sont visibles, modifiables et utilisables dans les autres menus d'AX Portal.

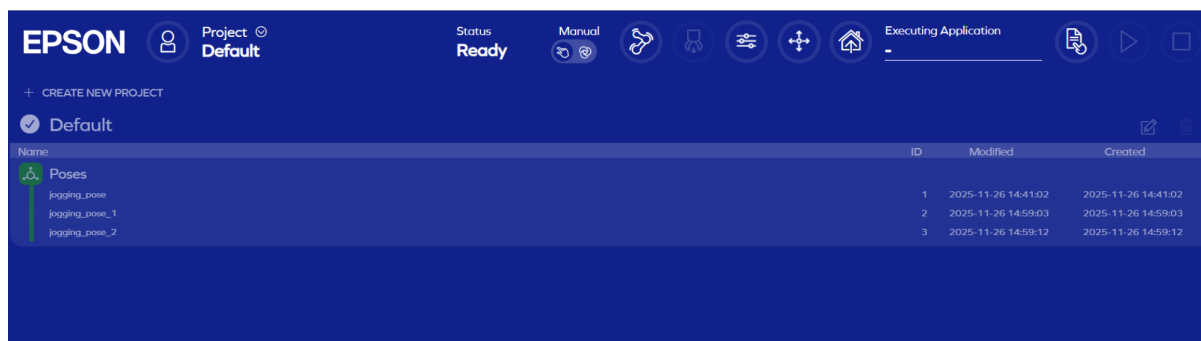


Figure 4.4 : Menu Projet



Cliquez sur le bouton « **Créer un nouveau projet** » pour créer un nouveau projet vide.



Cliquez sur le cercle situé à gauche du nom du projet pour sélectionner ce projet.



Cliquez sur le bouton « **Copier** » d'un élément de projet pour créer une copie exacte de cet élément au sein du même projet. Le nom de l'élément copié est légèrement modifié et se termine par « (copie) ».



Cliquez sur le bouton « **Modifier** » situé à droite du nom du projet pour modifier le nom de ce projet. Cliquez sur le bouton « **Modifier** » d'un élément de projet pour ouvrir l'éditeur correspondant afin de le consulter ou de le modifier.



Cliquez sur le bouton « **Supprimer** » situé à droite du nom du projet pour supprimer ce projet, y compris tous ses éléments. Cliquez sur le bouton « **Supprimer** » d'un élément de projet pour supprimer cet élément spécifique d'un projet.

Cliquez sur l'icône d'un élément (par exemple, poses, tracés, etc.) pour afficher ou masquer la liste des éléments de cette catégorie spécifique.

Déplacez un élément d'un projet vers un autre en le faisant glisser avec la souris depuis le projet d'origine et en le déposant dans le projet de destination. Maintenez la touche **CTRL** enfoncée pendant que vous faites glisser l'élément pour créer une **copie** de cet élément dans le projet de destination.

4.5 Menu de configuration

Cliquez sur le bouton « **Configuration** » dans la fenêtre principale pour accéder au menu de configuration. Dans ce menu, vous pouvez configurer la base de données et la configuration matérielle (E/S et outils). Si vous utilisez un robot fixe pour une application spécifique, vous n'avez besoin de configurer le robot qu'une seule fois et n'utiliserez ensuite cette fenêtre que pour mettre le robot sous tension. Vous pouvez même automatiser la mise en service du robot et l'exécution d'une application via les paramètres correspondants du menu des paramètres. Toutefois, si vous utilisez votre robot à des fins multiples, vous aurez plus souvent recours à ce menu et utiliserez différents outils, voire différentes bases de données, pour différents projets et objectifs.

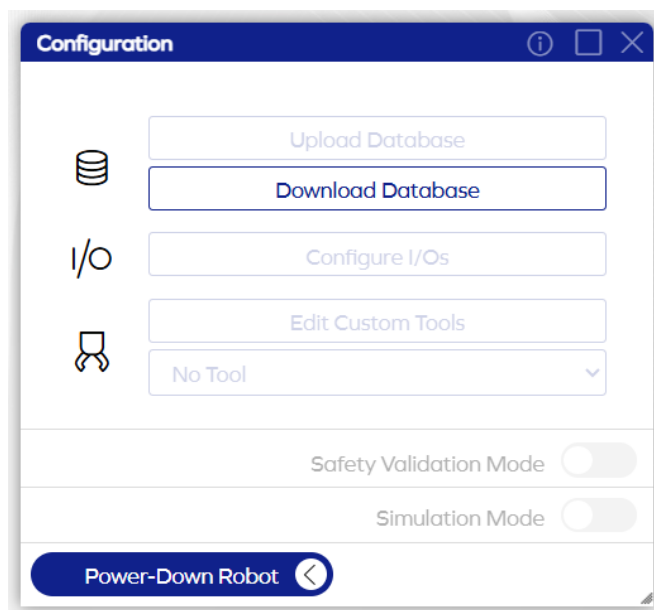


Figure 4.5 : Menu de configuration

4.5.1 Base de données (téléchargement et téléversement)

Dans la section « Base de données », vous pouvez télécharger la version actuelle de la base de données de votre application ou remplacer (écraser) la base de données par une autre version compatible. Lors du **téléchargement de la base de données**, une copie locale du fichier ZIP de la base de données est créée sur votre appareil. Il est recommandé de **sauvegarder régulièrement votre base de données**, en particulier si vous disposez de nombreuses applications et si vous avez consacré beaucoup de temps à leur écriture et à leur perfectionnement. Lors du **téléversement d'une base de données**, il vous est demandé de sélectionner un fichier ZIP sur votre appareil. En sélectionnant le fichier de base de données, **la base de données actuelle est écrasée** et remplacée par celle qui a été téléversée.

En téléchargeant une base de données sur un robot et en la téléversant sur un autre robot, vous pouvez transférer toutes les applications et tous les éléments d'un robot à l'autre. Il est recommandé de sauvegarder régulièrement la base de données.

Notez également que les bases de données sont versionnées et que toutes les bases de données ne sont pas compatibles avec toutes les versions d'AX Portal. Les bases de données plus anciennes peuvent être importées dans les versions plus récentes du logiciel, mais les versions plus récentes de la base de

données ne peuvent pas être chargées dans les versions plus anciennes du logiciel. En d'autres termes, les bases de données ne sont pas rétrocompatibles.

4.5.2 Configuration des E/S

Cliquez sur le bouton « **Configurer les E/S** » pour ouvrir la fenêtre « **Configuration des E/S** ». Vous pouvez y configurer toutes les E/S, attribuer des fonctions prédéfinies aux entrées numériques et définir des noms personnalisés pour vos E/S. Les noms personnalisés peuvent être utilisés dans les applications afin d'améliorer la lisibilité de votre code de programme. Vous pouvez alors simplement remplacer les identifiants entiers (int) par les noms d'E/S définis (str).

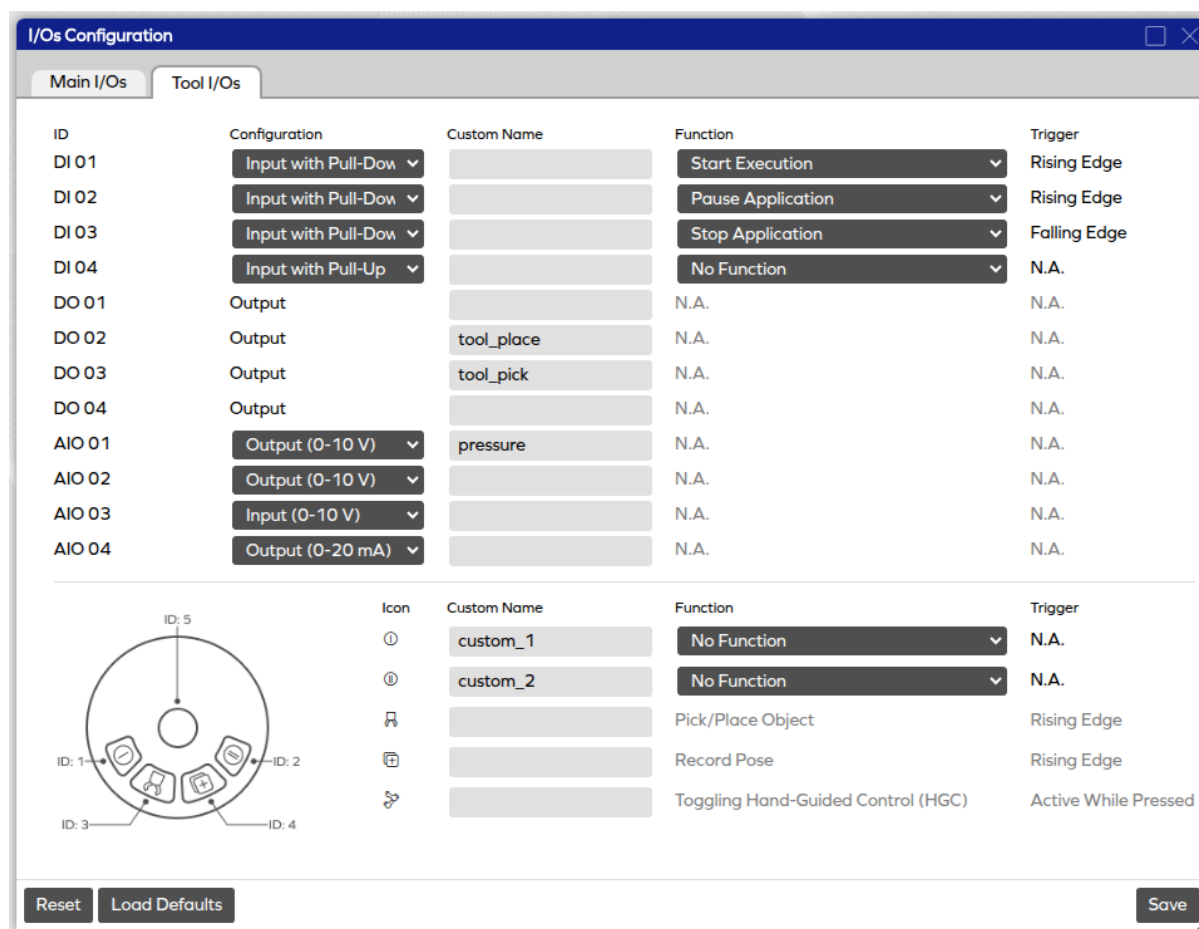


Figure 4.5.2.a : Menu « Configuration des E/S »

La série AX est livrée avec une « **interface à boutons** » fixée sur l'axe supérieur. Des fonctions peuvent être attribuées aux deux boutons personnalisés exactement de la même manière que des fonctions sont attribuées à d'autres entrées numériques dans ce menu.

Veuillez noter que lorsqu'une fonction est déclenchée via une entrée numérique, le contrôle actif est retiré de la station de commande actuelle.

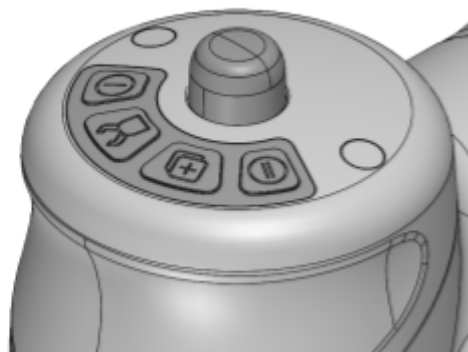


Figure 4.5.2.b : Interface à boutons AX

Les fonctions prédéfinies suivantes pour les entrées numériques sont disponibles ; vous pouvez les configurer pour qu'elles se déclenchent sur le front montant (par défaut) ou sur le front descendant :

Aucune fonction	Rien n'est déclenché. Il s'agit du réglage par défaut.
Mettre l'application en pause	Mise en pause de l'application en cours d'exécution. Cela correspond à l'action de cliquer sur le bouton « Pause » du cockpit.
Interrompre l'application	Arrêt de l'application en cours d'exécution ou en pause. Cela correspond à l'action d'appuyer sur le bouton « Arrêter » du Cockpit.
Lancer l'exécution	Exécuter l'application actuellement sélectionnée. Cela correspond à l'action d'appuyer sur le bouton « Exécuter » du Cockpit. Lorsque le robot se rétablit après une interruption, cette fonction le ramène à la position où il a été interrompu et reprend le mouvement précédent. Cela correspond à l'action d'appuyer sur le bouton « Continuer » dans la boîte de dialogue qui s'affiche dans l'interface graphique après une interruption.
Réinitialisation de la sécurité	Réinitialisation de la sécurité après une interruption. Cela correspond à la réinitialisation de la sécurité via la boîte de dialogue qui s'affiche dans l'interface graphique après une interruption.
Activer la commande manuelle	Activer le mode manuel (HGC) pour tous les axes (lorsque l'état est « prêt »).
Désactiver la commande manuelle	Désactiver le mode manuel (HGC) pour tous les axes (lorsque l'état est « Commande manuelle »).
Basculer la commande manuelle	Basculer entre l'activation et la désactivation du HGC pour tous les axes. La fonction déclenchée dépend de l'état actuel.

Actionner l'outil (annuler)	Déclenchement de la fonction de prise d'outil (prédéfinie ou associée à des sorties numériques). Cela correspond à l'actionnement du bouton « Saisir » du cockpit.
Manipuler l'outil (placer)	Déclenchement de la fonction de dépose de l'outil (prédéfinie ou associée à des sorties numériques). Cela correspond à l'actionnement du bouton « Placer » du cockpit.
Utiliser l'outil (prendre/poser)	Basculer entre la prise et la dépose d'un objet à l'aide de l'outil. En fonction de l'état actuel de l'outil.
Enregistrer la pose	Enregistrez la position actuelle du robot et sauvegardez-la sous le nom « button_pose » (ou similaire).
Passer en mode manuel	Passer le robot en mode manuel. Cela correspond au changement de mode de fonctionnement via le commutateur dans le cockpit.
Passer en mode automatique	Faites passer le robot en mode automatique. Cela correspond au changement de mode de fonctionnement via le commutateur dans le cockpit.

4.5.3 Configuration de l'outil

Dans la zone des outils, vous pouvez configurer l'outil du robot. Si vous souhaitez intégrer un outil personnalisé, par exemple un outil tiers ou votre propre outil spécialisé pour une application spécifique, vous pouvez utiliser l'option « **Modifier les outils personnalisés** ». Cela ouvre la liste des outils disponibles avec les options suivantes :

	Cliquez sur le bouton « Créer un nouvel outil » pour ouvrir l'éditeur d'outils afin de définir les propriétés cinématiques et dynamiques ainsi que le déclencheur de préhension et de placement de votre outil.
	Cliquez sur le bouton « Filtrer / Rechercher » pour filtrer la liste des outils personnalisés par nom. La saisie de trois caractères ou plus active le filtre de recherche.
	Cliquez sur le bouton « Modifier » (ou double-cliquez n'importe où sur l'élément) pour ouvrir l'éditeur d'outils dans une fenêtre séparée afin de le consulter ou de le modifier.
	Cliquez sur le bouton « Supprimer » pour supprimer définitivement un outil de votre base de données.

4.5.4 Éditeur d'outils

L'éditeur d'outils est divisé en quatre sections (voir l'illustration ci-dessous). Pour remplir ces champs, vous devrez peut-être consulter la fiche technique de l'outil que vous souhaitez intégrer dans AX Portal. Les propriétés physiques et électriques de l'outil y sont décrites et doivent être reportées dans l'éditeur d'outils.

Dans la première section, vous devez définir le TCP (Tool Center Point), c'est-à-dire la position dans l'espace à laquelle l'outil interagit. Pour un outil destiné à saisir des objets, cela signifie que le TCP définit l'emplacement où se trouve l'outil pendant la saisie. Le TCP est défini dans le système de coordonnées de référence situé exactement au centre de la plaque de raccordement supérieure et orienté de la même manière que le système de base (à condition que tous les axes soient à zéro degré). Ce système de référence est le **système d'outil par défaut** si aucun TCP n'est défini (x, y, z, roll, pitch).

et yaw tous à zéro). À partir de là, vous pouvez déplacer et faire pivoter le TCP de manière à ce qu'il corresponde au point d'interaction de votre outil.

À l'étape suivante, il est nécessaire d'ajuster les **propriétés dynamiques** de votre outil. La qualité de fonctions telles que la compensation de la gravité et la détection des collisions dépend de la précision du modèle dynamique du robot ; il est donc utile de mesurer et de définir ces propriétés de votre outil. Vous devez définir la masse totale de l'outil ainsi que son centre de masse, qui (tout comme le TCP) est défini dans le **système de référence standard de l'outil** du robot. Le [moment d'inertie](#) peut être saisi manuellement (s'il est connu) ou estimé automatiquement (AX Portal ne connaît pas la forme ni la répartition de la masse de votre outil).

Tool Editor : pneumatic_gripper

Name* pneumatic_gripper

1. Defining the Global Tool Frame

Translation [mm]

x 0

y 0

z 36

Rotation [deg]

roll 0

pitch 0

yaw 90

2. Defining the Dynamic Properties

Mass [kg] 0.662

Center of Mass [mm]

x 2

y -1.5

z 14

Rotational Moment of Inertia [kg m²]

Automatic ☒ Manual

0.0004649	0	0
0	0.0004649	0
0	0	0.0005880

3. Defining the Picking Behaviour

Mode Dual Output Mode

Select Digital Output for Picking tool DO 03 (tool_pick)

Select Digital Output for Placing tool DO 02 (tool_place)

Output State for Picking & Placing High

Delay Between Output Triggers T_1 [ms] 20

Pick Output

Place Output

Cancel
Reset
Save & Close

Figure 4.5.4 : Éditeur d'outils

Si votre outil dispose d'un mécanisme permettant de saisir un objet, vous pouvez définir ici quelles sorties numériques sont déclenchées par les fonctions « **Saisir** » et « **Placer** », accessibles aussi bien via le cockpit que via les fonctions de programmation. Les **comportements de préhension** suivants sont disponibles :

4.5.4.1 Mode sortie unique

Il s'agit du comportement de préhension le plus simple parmi ceux disponibles. Il vous suffit de définir une sortie numérique utilisée pour basculer entre la préhension et la dépose, et vous pouvez déterminer si la préhension est déclenchée par un signal HIGH ou LOW.



Figure 4.5.4.1 : Mode à sortie unique avec HIGH pour la prise (et LOW pour la dépose)

4.5.4.2 Mode double sortie

Ce mode de préhension vous permet de définir des sorties numériques distinctes pour la prise et la dépose. Lors de la commutation de l'état de l'outil, les deux sorties sont alors déclenchées l'une après l'autre avec un court délai T_1 (en millisecondes). De plus, vous pouvez déterminer si l'état actif correspond à un signal HIGH ou LOW.

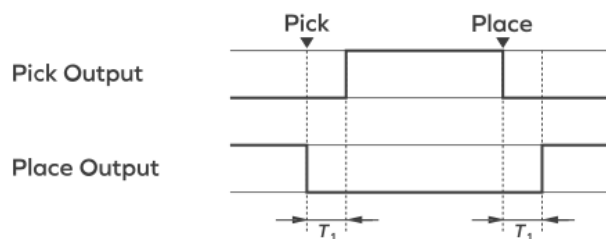


Figure 4.5.4.2 : Mode double sortie avec HIGH comme état de déclenchement actif

4.5.4.3 Mode de déclenchement séparé

En mode de déclenchement séparé, vous pouvez définir deux sorties numériques : l'une qui détermine la direction (à retirer ou à placer) et l'autre qui détermine le déclenchement (si la direction choisie doit être déclenchée). Pour la sortie de direction, vous pouvez choisir d'utiliser un signal HIGH ou LOW pour le retrait (l'autre étant réservé au placement). Pour la sortie de déclenchement, vous pouvez également choisir si un signal HIGH ou LOW est utilisé pour le déclenchement. De plus, vous pouvez définir le court délai T_1 (en millisecondes) entre la sélection de la direction et le déclenchement de la direction choisie, ainsi que la courte durée T_2 (en millisecondes) qui détermine la durée du déclenchement.

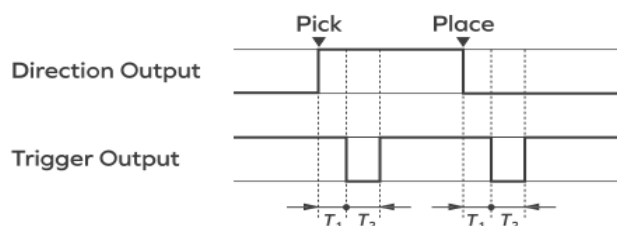


Figure 4.5.4.3 : Mode de déclenchement séparé avec HIGH pour la préhension et LOW comme déclencheur

4.5.5 Mise en service du robot

Une fois la base de données chargée et les E/S ainsi que l'outil configurés, il est temps de mettre le robot sous tension. Pour ce faire, faites glisser le curseur « Mise en service du robot » complètement vers la droite, ce qui lance la mise en service.

Si vous ne souhaitez pas mettre le matériel sous tension, mais seulement modifier le contenu de la base de données (applications, poses, etc.) sans utiliser le matériel (par exemple, sans déplacer le robot, sans lire ni écrire d'E/S), vous pouvez d'abord activer le commutateur « Mode simulation », puis faire glisser le curseur « Mettre le robot en service » vers la droite. Notez qu'en mode simulation, tous les mouvements sont simulés en temps réel.

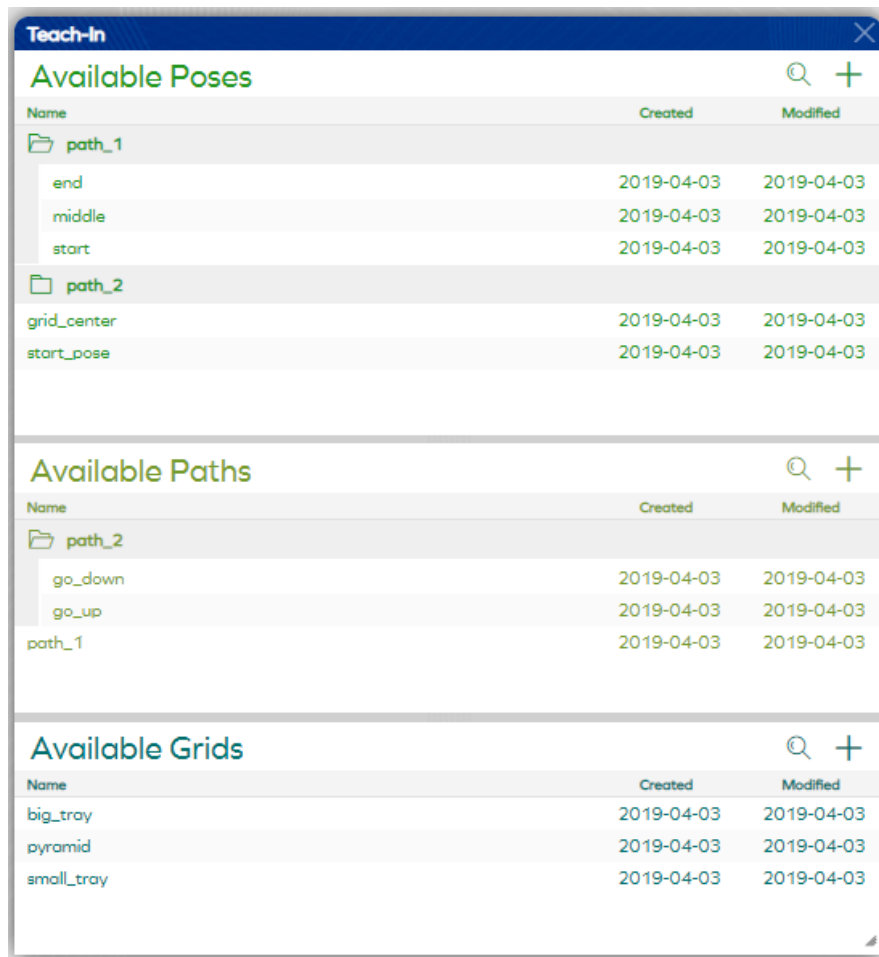
Tant que le robot est sous tension, vous ne pouvez modifier aucun des paramètres matériels du robot. Vous pouvez uniquement télécharger la base de données ou faire glisser le curseur « Mettre le robot hors service » vers la gauche, ce qui est recommandé lorsque le robot est au repos ou en veille.

Si vous êtes connecté en tant que responsable de la sécurité, vous avez ici accès au bouton « Mode de contrôle de sécurité ». L'activation de ce mode indique au logiciel d'ignorer les limites de position, de vitesse et de force, ce qui peut être utilisé pour vérifier si le dispositif de sécurité se déclenche effectivement en cas de dépassement des limites. Veuillez noter que l'utilisation de l'une des commandes de mouvement dans ce mode continue de passer outre les limites.

4.6 Menu d'apprentissage

Appuyez sur le bouton « **Apprentissage** » dans la fenêtre principale pour accéder au menu d'apprentissage. Ce menu contient un aperçu de toutes les poses, trajectoires et grilles disponibles pour le projet actuellement sélectionné.

Les poses et les trajectoires peuvent être créées, modifiées, copiées, supprimées et même exécutées directement. Les grilles, quant à elles, sont des outils utiles pour organiser les coordonnées récurrentes et régulièrement espacées. Veuillez noter que les grilles ne sont pas utilisées directement pour déplacer un robot, mais regroupent et enregistrent des ensembles de valeurs de coordonnées afin de structurer votre code d'application et de vous éviter d'avoir à saisir des coordonnées. Une liste de points de grille peut être extraite d'une grille à des fins diverses ; cependant, les grilles sont généralement configurées de manière à ce que leurs points représentent directement des coordonnées TCP (Tool Center Point), qui sont ensuite utilisées comme valeurs d'entrée pour les fonctions de mouvement.



The screenshot shows the 'Teach-In' window with three sections: 'Available Poses', 'Available Paths', and 'Available Grids'. Each section has a search icon and a plus icon in the top right corner. The 'Available Poses' section lists poses under two folders: 'path_1' (end, middle, start) and 'path_2' (grid_center, start_pose). The 'Available Paths' section lists paths under 'path_2' (go_down, go_up) and 'path_1'. The 'Available Grids' section lists grids: big_tray, pyramid, and small_tray. All items have 'Created' and 'Modified' dates of 2019-04-03.

Available Poses		
Name	Created	Modified
path_1		
end	2019-04-03	2019-04-03
middle	2019-04-03	2019-04-03
start	2019-04-03	2019-04-03
path_2		
grid_center	2019-04-03	2019-04-03
start_pose	2019-04-03	2019-04-03

Available Paths		
Name	Created	Modified
path_2		
go_down	2019-04-03	2019-04-03
go_up	2019-04-03	2019-04-03
path_1	2019-04-03	2019-04-03

Available Grids		
Name	Created	Modified
big_tray	2019-04-03	2019-04-03
pyramid	2019-04-03	2019-04-03
small_tray	2019-04-03	2019-04-03

Figure 4.6 : Menu d'apprentissage

	Cliquez sur le bouton « Nouveau/Ajouter » pour ouvrir un éditeur de pose, de trajectoire ou de grille dans une nouvelle fenêtre. Tous ces éditeurs sont expliqués en détail plus loin dans cette section.
	Cliquez sur le bouton « Filtrer / Rechercher » pour filtrer la liste des poses, des trajectoires ou des grilles par nom. La saisie de trois caractères ou plus active le filtre de recherche.
	Appuyez sur le bouton « Exécuter » pour déplacer le robot vers une pose ou pour lire un chemin. Notez que lors de la lecture d'un chemin, le robot se positionne d'abord sur la pose de départ de ce chemin, à moins qu'il ne se trouve déjà dans cette pose spécifique.
	Cliquez sur le bouton « Copier » pour créer une copie exacte de cette pose ou de ce chemin. Le nom de l'élément copié est légèrement modifié et se termine par « (copie) ».
	Cliquez sur le bouton « Modifier » (ou double-cliquez n'importe où sur l'élément) pour ouvrir cette pose, ce parcours ou cette grille dans une fenêtre séparée afin de les visualiser ou de les modifier.
	Cliquez sur le bouton « Supprimer » pour supprimer définitivement une pose, un chemin ou une grille de votre base de données.
	Cliquez sur un dossier pour afficher ou masquer les poses, les chemins ou les grilles qu'il contient. Les éléments ne s'affichent dans les dossiers (et sous-dossiers) que si leur nom contient un ou plusieurs caractères « / », par exemple « dossier/sous-dossier/nom ».

4.6.1 Éditeur de poses

L'éditeur de pose permet aux utilisateurs de créer une nouvelle pose ou de modifier une pose existante. Veuillez noter que la modification des coordonnées ou de la configuration d'une pose ne déplace pas automatiquement le robot. Si vous le souhaitez, il convient d'utiliser plutôt la commande fine.

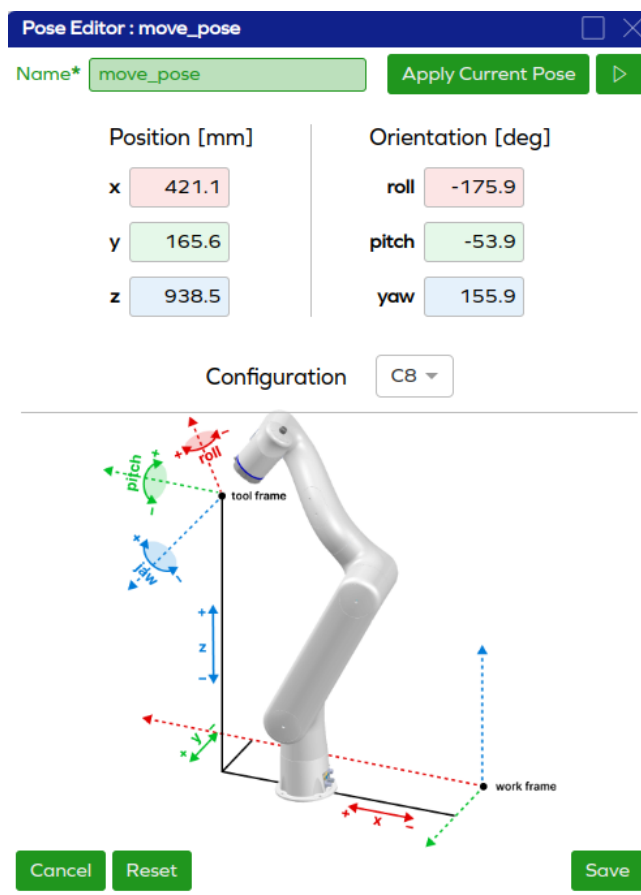


Figure 4.6.1 : Éditeur de pose

Nom :

Indique le nom sous lequel la pose est enregistrée dans la base de données. Utilisez le caractère « / » dans le nom pour regrouper vos poses dans une structure de type arborescence.

Position :

Représente la position TCP (Tool Center Point) en millimètres, définie par rapport au système de coordonnées global. Notez que des valeurs de position non valides, par exemple celles qui ne peuvent pas être atteintes par le bras du robot, peuvent être saisies. Elles sont toutefois surlignées en rouge, car appuyer sur le bouton de lecture entraîne une erreur. Cela est autorisé car la pose peut être valide lorsqu'elle est utilisée avec différents systèmes de coordonnées et décalages.

Orientation :

Représente l'orientation du TCP en degrés, définie par rapport au système de travail global. L'orientation est exprimée en angles nautiques. Comme pour les valeurs de position, les valeurs d'orientation non valides sont mises en évidence en rouge.

Configuration :

Indique la configuration du bras robotique. L'utilisateur peut choisir parmi toutes les configurations disponibles (C1–C8) via le menu déroulant. Veuillez noter que la modification de la configuration ne change pas la position ou l'orientation du TCP ; elle influence uniquement les angles d'articulation utilisés pour atteindre la pose spécifiée.

Actions supplémentaires dans l'éditeur de pose

Enregistrer	En cliquant sur le bouton « Enregistrer », la pose actuelle est enregistrée sous le nom spécifié. Une pose ne peut pas être enregistrée sans nom valide.
Annuler	En cliquant sur le bouton « Annuler », toutes les modifications non enregistrées effectuées dans l'éditeur sont rejetées et la fenêtre se ferme.
Réinitialiser	En cliquant sur le bouton « Réinitialiser », toutes les modifications non enregistrées sont annulées et la pose est rétablie dans son dernier état enregistré.
 Lecture	En cliquant sur le bouton « Lire » en haut de la fenêtre, vous demandez au robot de se mettre dans la pose définie.
Appliquer la pose actuelle	En cliquant sur le bouton « Appliquer la pose actuelle », tous les champs sont mis à jour afin de refléter les valeurs de pose actuelles du robot.

4.6.2 Éditeur de trajectoire

L'éditeur de trajectoire permet aux utilisateurs de créer une nouvelle trajectoire ou de modifier une trajectoire existante. Il est divisé en trois sections modifiables : **Nom**, **Position initiale** et **Segments de cette trajectoire**.

The screenshot shows the 'Path Editor : path_1' window. At the top, there is a 'Name*' field with 'path_1' and a play button. Below it is the 'Initial Pose' dropdown set to 'path_1/initial_pose'. The main section is titled 'Segments of this path'. It contains two segment cards. The first card is for a 'Linear' segment with a 'Target Pose' dropdown set to 'path_1/pose_1', a 'Velocity' input of 500 mm/s, and an 'Acceleration' input of 1000 mm/s². The second card is for a 'Circular' segment with a 'Blend Radius' input of 100 mm, an 'Intermediate Pose' dropdown set to 'path_1/pose_2', a 'Target Pose' dropdown set to 'path_1/pose_3', a 'Velocity' input of 250 mm/s, and an 'Acceleration' input of 500 mm/s². At the bottom right of the segments list is an 'Add a segment' button with a 'Circular' dropdown and a '+' icon. At the very bottom are 'Cancel', 'Reset', and 'Save' buttons.

Figure 4.6.2 : Éditeur de trajectoire

Nom :

Indique le nom sous lequel le chemin est enregistré dans la base de données. Utilisez le caractère « / » dans le nom pour regrouper vos chemins dans une structure de type arborescence.

Position initiale :

Définit la pose de départ de la trajectoire. Le robot se déplace d'abord vers cette pose en utilisant les paramètres par défaut de vitesse et d'accélération, avant d'exécuter les segments définis les uns après les autres.

Segments de ce chemin :

Répertorie et définit les différents segments qui composent le chemin. Les types de segments suivants sont disponibles :

- Segment linéaire
- Segment circulaire
- Segment de coordonnées
- Segment de position
- Segment d'orientation

Vous trouverez une description détaillée de chaque segment dans la section Segment.

Pour **ajouter un nouveau segment**, sélectionnez le type de segment souhaité dans le menu déroulant situé dans le coin inférieur droit, puis cliquez sur le bouton « + ». Une fois les segments ajoutés, leur **ordre** peut être librement modifié par glisser-déposer.

Chaque segment nécessite la spécification d'une **pose cible**, qui définit la pose finale de ce segment. Les poses cibles doivent avoir été préalablement définies dans l'éditeur de poses et peuvent ensuite être sélectionnées via le menu déroulant. Le segment circulaire nécessite spécifiquement une **pose intermédiaire** supplémentaire, qui doit également être sélectionnée dans la liste des poses préalablement définies.

Des paramètres de **vitesse** et **d'accélération** peuvent être définis pour chaque segment. Si ceux-ci ne sont pas spécifiés, les valeurs par défaut globales sont utilisées lors de l'exécution.

Le rayon de transition peut être configuré pour contrôler la fluidité et l'efficacité des transitions entre les segments. Il définit l'écart maximal autorisé en mm par rapport à la pose finale du segment précédent, qui correspond à la pose de départ du segment actuel.

Actions supplémentaires dans l'éditeur de trajectoire

L'éditeur de trajectoire offre également des fonctions supplémentaires permettant de gérer, modifier et organiser efficacement la trajectoire.

Enregistrer	En cliquant sur le bouton « Enregistrer », le chemin actuel est enregistré sous le nom indiqué. Un chemin ne peut pas être enregistré sans nom valide.
Annuler	En cliquant sur le bouton « Annuler », toutes les modifications non enregistrées effectuées dans l'éditeur sont rejetées et la fenêtre se ferme.
Réinitialiser	En cliquant sur le bouton « Réinitialiser », toutes les modifications non enregistrées sont annulées et le chemin d'accès est rétabli dans son dernier état enregistré.
 Lecture	En cliquant sur le bouton « Lecture » en haut de la fenêtre, vous demandez au robot d'exécuter le chemin défini. Si la position actuelle du robot ne correspond pas à la position de départ indiquée pour le chemin, il se déplace d'abord vers cette position de départ avant de commencer l'exécution.
 Supprimer	En cliquant sur le bouton « Supprimer » à côté d'un segment, ce segment est supprimé du chemin.

4.6.3 Éditeur de grille

Lors de la création d'une nouvelle grille, l'éditeur de grille s'ouvre. Cet éditeur guide l'utilisateur pas à pas à travers toutes les options de grille disponibles. Utilisez les boutons « **Suivant** » et « **Précédent** » pour naviguer dans cet assistant jusqu'à ce que la liste ordonnée des points de grille soit définie comme souhaitée. Lors de la modification de grilles existantes, l'éditeur de grille passe directement à l'édition manuelle des points.

Dans un premier temps, l'utilisateur doit sélectionner la **dimension de la grille**. En général, une grille est interprétée comme un tableau bidimensionnel, mais il peut également s'agir d'une simple liste de points sans dépendance à une deuxième dimension (par exemple pour prélever des objets sur un convoyeur) ou même d'une structure tridimensionnelle (par exemple pour empiler des objets dans des caisses). La grille est initialisée avec des points parfaitement alignés, sachant que les points individuels de la grille peuvent être modifiés, ajoutés ou supprimés à tout moment par la suite.

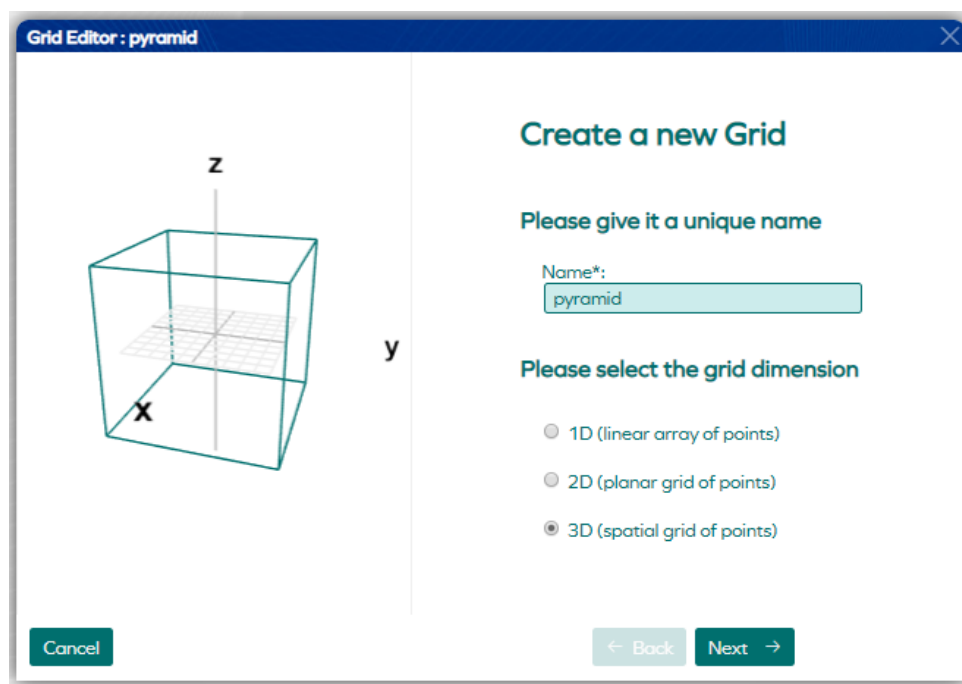


Figure 4.6.3.a : Éditeur de grille (1re étape)

Dans un deuxième temps, l'utilisateur doit déterminer les **sommets de la grille** sélectionnée. Les grilles de dimensions différentes ont un nombre différent de sommets, et les sommets sont désignés par « A » à « D », par exemple, pour une grille bidimensionnelle. L'utilisateur peut saisir ces valeurs de coordonnées (indiquées dans le système de référence du robot) directement dans les champs de saisie ou simplement diriger le TCP de l'effecteur terminal vers les sommets de la grille et appuyer sur les boutons « Enregistrer » correspondants pour lire les coordonnées actuelles.

Dès qu'un nombre suffisant de coins a été appris pour définir la grille (par exemple, au moins 3 points pour la grille bidimensionnelle), l'utilisateur peut cliquer sur le bouton « **Évaluer les valeurs** » pour analyser les coins indiqués et compléter automatiquement les valeurs manquantes de manière à ce qu'elles s'inscrivent dans un parallélogramme (bidimensionnel) ou un parallélépipède (tridimensionnel). Si l'option « **Alignement automatique** » est également activée, la méthode prend

en compte les axes et plans de coordonnées horizontaux et verticaux et tente d'aligner la grille résultante le long de ces axes et plans (dans la mesure où ils sont suffisamment proches).

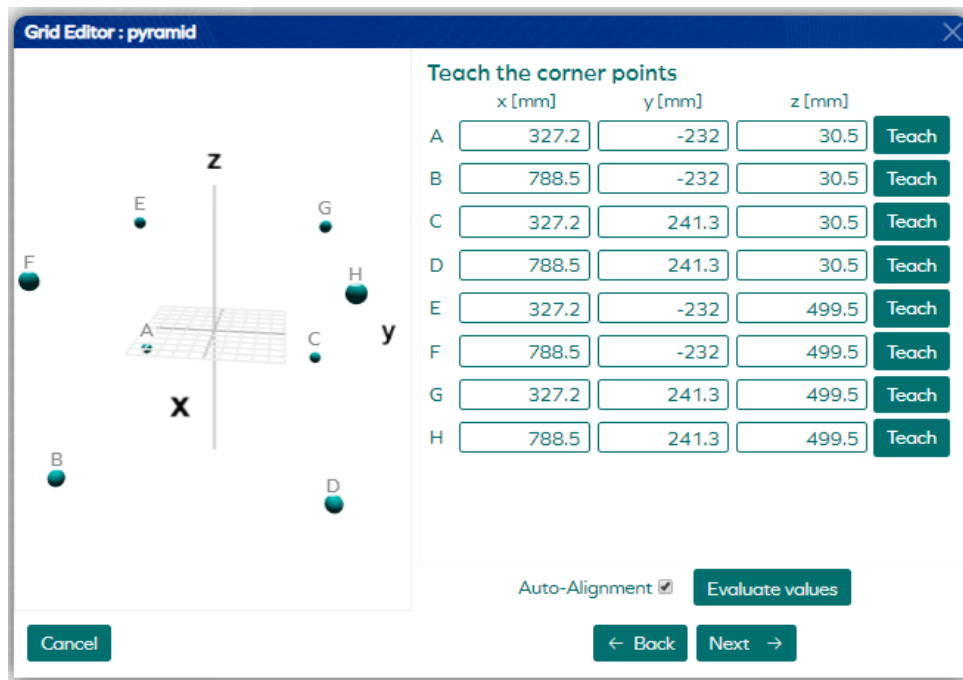


Figure 4.6.3.b : Éditeur de grille (2e étape)

Dans une troisième étape, l'utilisateur est invité à choisir le nombre de points, des options d'extension spécifiques et les directions d'itération. Examinons ces options disponibles en détail :

- Le **nombre de points** peut être défini individuellement pour chaque dimension. Si vous avez du mal à distinguer les dimensions (par exemple, les lignes et les colonnes), référez-vous aux libellés des points d'angle définis précédemment.
- Pour les grilles bidimensionnelles, il existe une **option d'extension** permettant d'ajouter des points intermédiaires entre chaque groupe de quatre points adjacents. Pour les grilles tridimensionnelles, cette option est disponible à la fois pour chaque plan de la grille et pour le groupe de huit points adjacents dans l'espace.
- La **direction d'itération**, c'est-à-dire la direction dans laquelle les points sont renvoyés et utilisés comme itérateurs dans les boucles de programmation, est spécifiée par un point de départ et un vecteur dans chaque direction. La dernière direction est toujours déterminée par les options sélectionnées précédemment et n'est affichée qu'à titre de référence et de vérification visuelle. Les lettres dans les menus déroulants correspondent aux étiquettes des sommets de cette grille.

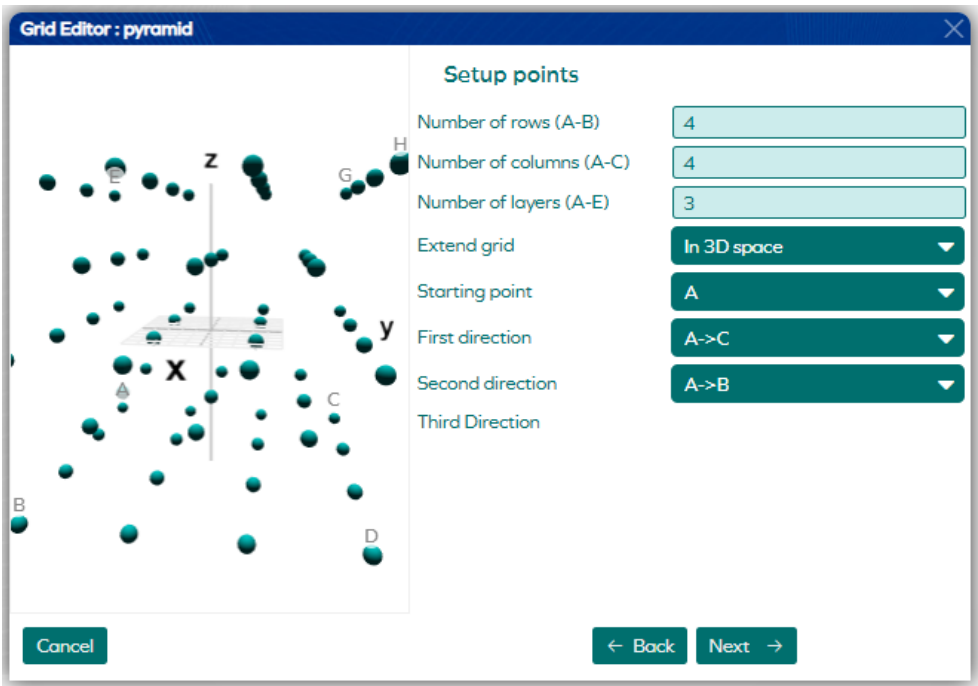


Figure 4.6.3.c : Éditeur de grille (3e étape)

Enfin, la **liste des points** de la grille est disponible pour une édition manuelle. Il est ainsi possible de copier, supprimer ou déplacer des points individuels, et de modifier manuellement les valeurs des coordonnées. Parallèlement, la visualisation affiche l'ordre des points et les coordonnées relatives des points de la grille.

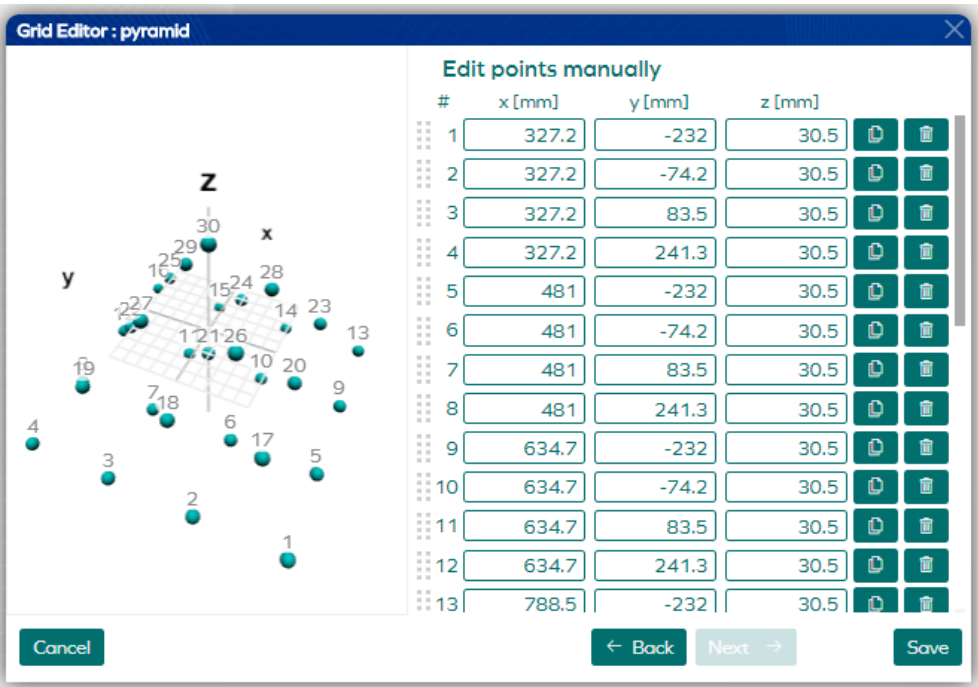


Figure 4.6.3.d : Éditeur de grille (4e étape)

Pendant l'édition, vous pouvez à tout moment revenir en arrière pour vérifier les options et les valeurs précédemment définies. Vous pouvez également modifier ces options, mais cela réinitialisera toutes les modifications effectuées dans les fenêtres suivantes.

4.7 Menu Applications

Cliquez sur le bouton « **Applications** » dans la fenêtre principale pour accéder au menu Applications. Ce menu contient un aperçu de toutes les applications enregistrées ainsi que la sortie de la dernière application exécutée. Il permet également à l'utilisateur d'accéder à la fenêtre « **Éditeur d'applications** » pour créer de nouvelles applications ou modifier celles existantes.

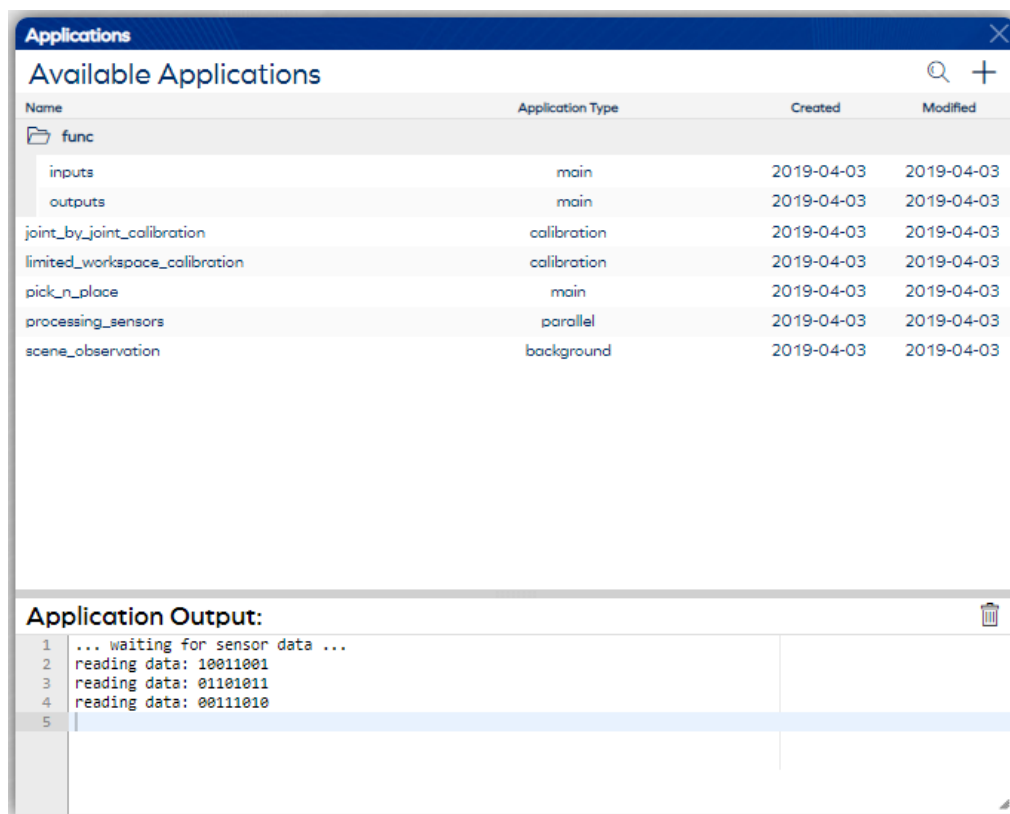


Figure 4.7 : Menu Applications



Appuyez sur le bouton « **Nouveau/Ajouter** » pour créer une nouvelle application. Cela ouvre l'éditeur d'applications, dont les fonctions sont expliquées en détail plus loin dans cette section.



Appuyez sur le bouton « **Filtrer / Rechercher** » pour filtrer la liste des applications par nom. La saisie de trois caractères ou plus active le filtre de recherche.



Appuyez sur le bouton « **Exécuter** » pour exécuter une application. Notez que seules les applications principales peuvent être exécutées de cette manière, car seules les applications principales peuvent être sélectionnées dans le Cockpit.



Cliquez sur le bouton « **Copier** » pour créer une copie exacte de cette application. Le nom de l'élément copié est légèrement modifié et se termine par « (copie) ».



Cliquez sur le bouton « **Modifier** » (ou double-cliquez n'importe où sur l'élément) pour ouvrir cette application dans une fenêtre séparée afin de la consulter ou de la modifier.



Cliquez sur le bouton « **Supprimer** » pour supprimer définitivement une application de votre base de données. Cela supprime à la fois le code du programme et les paramètres spécifiques à l'application.



Cliquez sur un **dossier** pour afficher ou masquer les applications qu'il contient. Les éléments ne s'affichent dans les dossiers (et sous-dossiers) que si leur nom contient un ou plusieurs caractères « / », par exemple « dossier/sous-dossier/nom ». Vous pouvez cliquer sur le bouton « **Modifier** » qui apparaît pour ouvrir l'éditeur de dossiers.

4.7.1 Éditeur d'applications

La section « **Applications disponibles** » du menu **Applications** répertorie toutes les applications précédemment enregistrées. Vous pouvez modifier une application existante en cliquant sur le bouton « **Modifier** » ou créer une nouvelle application en appuyant sur le bouton « **Créer une nouvelle application** ». Dans les deux cas, la fenêtre « **Éditeur d'applications** » s'ouvre. Son fonctionnement est similaire dans les deux cas. Jetons un bref coup d'œil aux boutons les plus importants de l'éditeur d'applications avant d'entrer dans les détails :

Enregistrer	En cliquant sur le bouton « Enregistrer », l'application est enregistrée sous le nom indiqué. Une application ne peut pas être enregistrée sans nom. Utilisez le caractère « / » dans le nom pour regrouper vos applications dans une structure de type dossier.
Annuler	En cliquant sur le bouton « Annuler », toutes les modifications non enregistrées effectuées dans cet éditeur sont rejetées et l'éditeur se ferme.
 Tester	En cliquant sur le bouton « Tester l'application » en haut de la fenêtre, l'application est exécutée en mode test. Le mode test signifie que le code du programme est exécuté, mais que vous pouvez continuer à modifier l'application. Les modifications apportées à l'application après le test ne sont pas exécutées.

4.7.2 Code du programme

L'onglet « **Code** » de l'application est le plus important des sous-fenêtres de l'éditeur d'application. Cette zone affiche le code du programme et, en option, la bibliothèque contenant les fonctions et les éléments AX Portal pouvant être insérés dans le code du programme. Le langage de programmation intégré est **Python 3.13**, avec la plupart de ses fonctionnalités de base, y compris certains paquets Python préinstallés. De plus, AX Portal propose une vaste bibliothèque de fonctions qui simplifient l'écriture d'une application robotique (par exemple, déplacer le robot, lire les données des capteurs, communiquer avec l'utilisateur, etc.). Vous trouverez des détails sur ces fonctions dans la documentation relative au code.

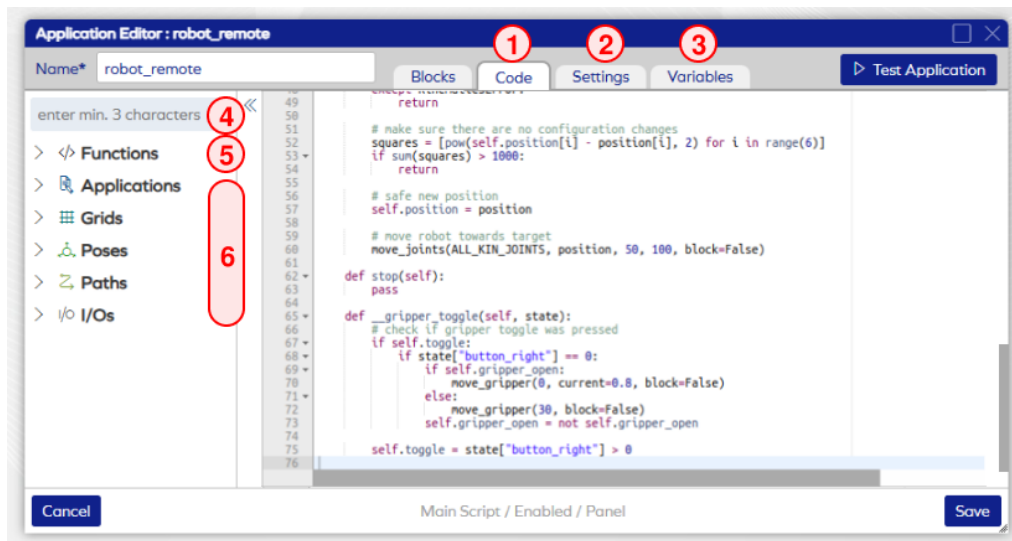


Figure 4.7.2 : Éditeur d'application (zone de code)

1. Écrivez ici votre code de programme Python, 2. Modifier les paramètres spécifiques à l'application,
3. Définir et modifier les variables de l'application, 4. Filtre de recherche, 5. Bibliothèque de fonctions, 6. Éléments disponibles à utiliser dans le programme

Cliquez sur l'une des **icônes de fonction** ou l'une des **icônes d'élément** situées à gauche de l'éditeur pour ouvrir la **palette** d'éléments. Vous y trouverez une liste complète des fonctions disponibles. Vous avez également accès à tous les éléments que vous pouvez intégrer à votre code, par un double-clic, ce qui vous évite d'avoir à mémoriser les noms de vos éléments.

Astuce : améliorez la lisibilité du code Python en supprimant les arguments facultatifs qui sont insérés dans votre code lorsque vous double-cliquez sur une fonction. Vous pouvez par exemple écrire

```
move_joints(joints, joint_position)
```

au lieu de

```
move_joints(joints, joint_position, joint_velocity=None, joint_acceleration=None,
relative=False, block=True)
```

4.7.3 Paramètres de l'application

L'onglet « **Paramètres** » contient quelques paramètres supplémentaires pour l'application, notamment

- un bouton « **Exécutable** », qui peut par exemple être désactivé si votre application est encore en cours de développement et ne doit pas être exécutée,
- un bouton permettant **d'afficher** cette application dans l'interface du panneau ("Panel Interface") ,
- une fonctionnalité permettant de télécharger et de remplacer **l'image** affichée dans l'interface du panneau ("Panel Interface") ,
- des **titres personnalisés** permettant d'afficher le nom d'affichage dans l'interface du panneau ("Panel Interface") dans la langue actuellement sélectionnée,
- et surtout le **type d'application**, qui est expliqué dans la section suivante :

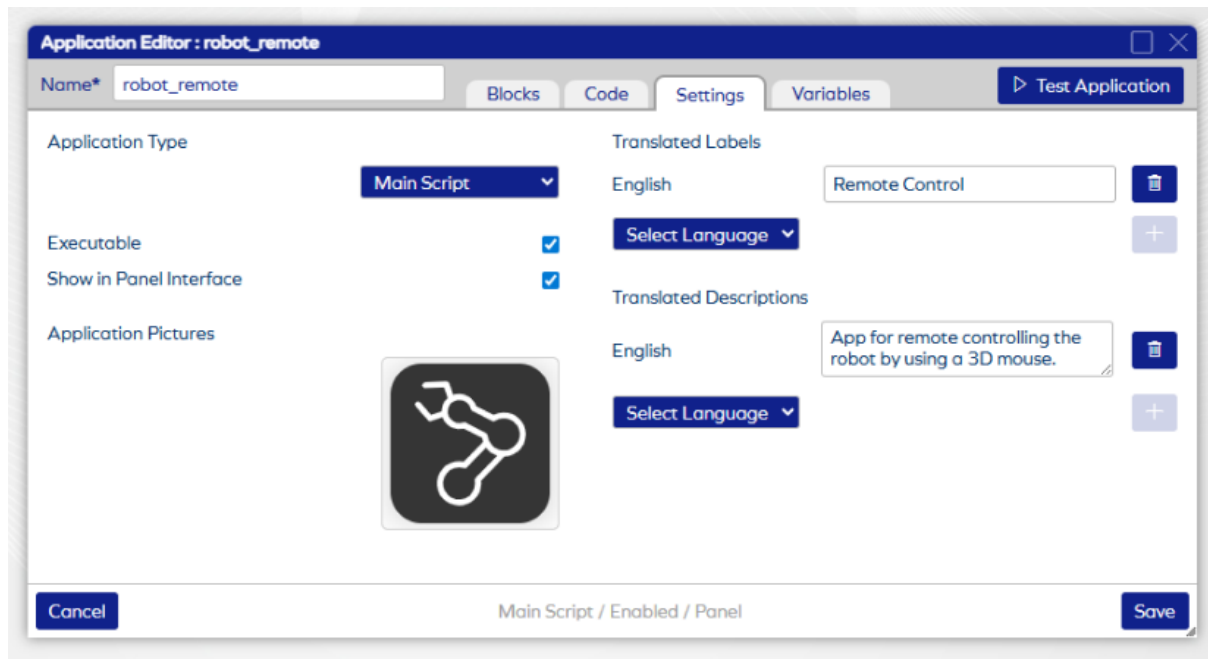


Figure 4.7.3.a : Éditeur d'application (zone des paramètres)

Le **type d'application** définit le contexte dans lequel l'application doit être exécutée et détermine donc comment une application peut être exécutée ainsi que les fonctions pouvant être utilisées au sein de cette application :

- Les **applications principales** sont les applications normales destinées à être exécutées directement. Elles sont conçues pour prendre en charge toutes les fonctions disponibles et pour contrôler directement le robot ainsi que ses périphériques externes. En général, une application simple peut être écrite dans un seul programme ; toutefois, pour les applications longues ou complexes, ou pour améliorer la lisibilité, il peut être judicieux de diviser une application en plusieurs programmes qui s'exécutent les uns après les autres. Cependant, plusieurs applications principales ne s'exécutent jamais en parallèle ; c'est à cette fin que l'on utilise des « applications parallèles ». Vous pouvez également utiliser l'instruction « import » pour importer des fonctions d'une application principale à l'autre. Notez que seules les applications principales peuvent être sélectionnées via le Cockpit.
- Les **applications parallèles**, en revanche, s'exécutent dans des threads distincts et sont donc conçues pour fonctionner en parallèle d'une application principale. Cela peut s'avérer très utile, par exemple pour la lecture continue de données de capteurs ou pour exploiter le temps d'inactivité du robot afin d'effectuer des calculs intensifs. Les applications parallèles ne prennent pas en charge toutes les fonctions, telles que le déplacement du robot, car cela compliquerait considérablement le déroulement du programme ou pourrait entraîner des mouvements indésirables, voire dangereux, en cas de manque d'attention. Toutes les applications parallèles sont mises en pause lorsque l'application principale est en pause, et elles sont arrêtées dès que l'application principale s'arrête.
- Les **applications d'arrière-plan** sont des applications qui s'exécutent en arrière-plan d'AX Portal, indépendamment de l'état du robot, et qui sont donc utiles, par exemple, pour surveiller, vérifier ou consigner en permanence des données, ou pour définir des variables

globales initiales pour un projet. Chaque projet ne prend en charge qu'une seule application d'arrière-plan. Une fois celle-ci définie, elle s'exécute automatiquement lorsque le projet correspondant est chargé (par exemple lors de la mise en service du robot ou lors du changement de projet). L'application d'arrière-plan ne tient pas compte des restrictions d'état et peut, si elle contient une boucle, s'exécuter jusqu'à ce que le robot soit mis hors service. Chaque fois qu'une application d'arrière-plan démarre et s'arrête, une notification correspondante s'affiche. Vous pouvez démarrer, redémarrer ou arrêter l'application d'arrière-plan définie depuis le Cockpit.



Figure 4.7.3.b : Commandes pour l'application d'arrière-plan

4.7.4 Variables d'application

L'onglet « **Variables** » de l'application comporte une interface de type glisser-déposer permettant d'ajouter différents types de variables d'état à cette application. Les variables d'état ne peuvent être utilisées que dans les applications principales. Elles sont utilisées dans l'interface du panneau ("Panel Interface"), où elles sont par exemple très utiles pour surveiller l'état et la progression d'une application ou pour spécifier certaines valeurs d'entrée pour une application. Utilisez les fonctions **set_state_variable(...)** et **get_state_variable(...)** pour écrire et lire des variables d'état.

Voici un aperçu des types de variables et de la manière dont elles sont affichées dans l'interface du panneau ("Panel Interface") :

booléen	Les booléens sont représentés sous forme de simples curseurs qui sont soit activés, soit désactivés. Ils sont lus et écrits en tant que type Python bool , où activé correspond à True et désactivé à False .
entier	Les valeurs entières sont représentées sous forme de champs de texte permettant la saisie de chiffres, avec des boutons « + » et « - » pour augmenter ou diminuer leur valeur en mode édition. Elles sont lues et écrites en tant que type Python int .
float	Les nombres à virgule flottante sont représentés sous forme de champs de texte qui permettent la saisie de valeurs numériques en mode édition. Ils sont lus et écrits en tant que type float en Python.
slider	Les nombres à virgule flottante sont représentés en mode édition sous forme de curseurs avec des valeurs minimale et maximale obligatoires. Ils sont lus et écrits en tant que type Python float .
text	Les textes sont affichés sous forme de champs de texte en mode édition. Ils sont lus et écrits en tant que type Python str .
dropdown	Les variables avec une sélection spécifique sont représentées en mode édition sous forme de listes déroulantes. Elles sont lues et écrites en tant que type Python str .

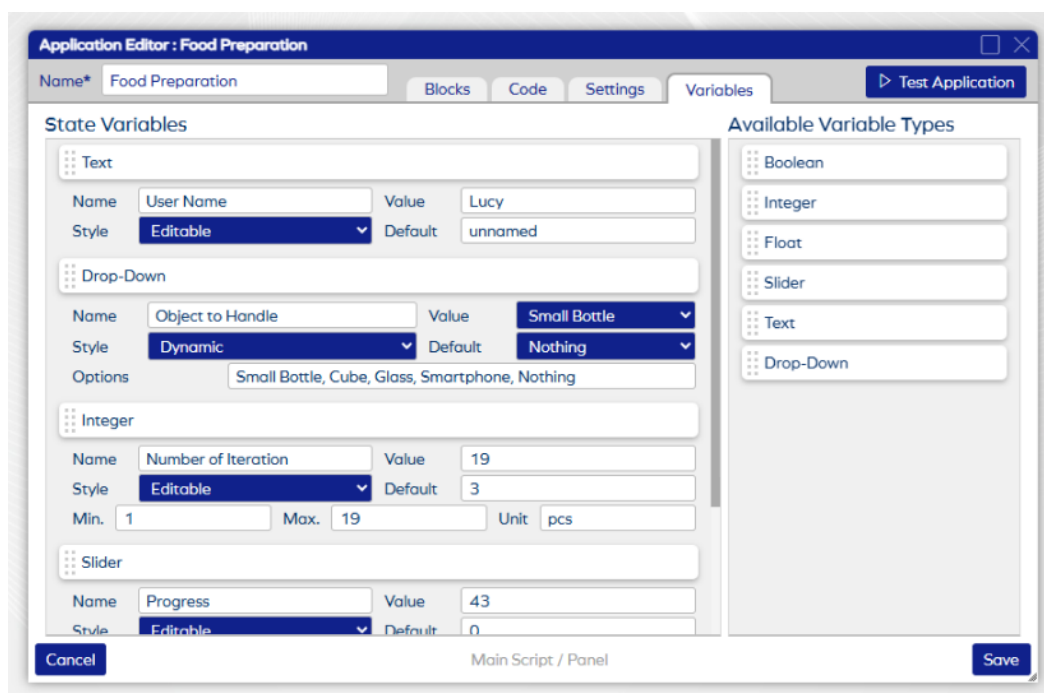


Figure 4.7.4 : Éditeur d'application (zone des variables)

Voici un aperçu des options, qui varient en fonction du type de variable :

Nom	Donnez un nom approprié à votre variable ; elle sera ensuite accessible sous ce nom sous la forme d'une chaîne de caractères.
Valeur	La valeur d'une variable contient la valeur actuellement définie pour cette variable, qui est lue et écrite dans le type Python correspondant expliqué ci-dessus.
Par défaut	La valeur par défaut d'une variable est la valeur qu'elle prend lorsque les variables sont réinitialisées à l'aide du bouton correspondant dans l'interface du panneau ("Panel Interface") .
Style	Le style d'une variable définit si, quand et comment cette variable est modifiable.
Min.	Les variables numériques permettent de définir une limite inférieure pour ce nombre. Si vous saisissez des valeurs inférieures au minimum, la valeur minimale est prise en compte. Les limites sont obligatoires pour les curseurs et facultatives pour les autres variables.
Max.	Les variables numériques permettent de définir une limite supérieure pour ce nombre. Si vous saisissez des valeurs supérieures au maximum, la valeur maximale est prise en compte. Les limites sont obligatoires pour les curseurs et facultatives pour les autres variables.
Unité	Pour les variables numériques, vous pouvez, si vous le souhaitez, ajouter une unité sous forme de chaîne de caractères au nombre.
Options	Les options pour les variables de liste déroulante peuvent être saisies sous forme de chaîne de caractères séparée par des virgules.

Voici un aperçu des styles de variables qui déterminent la manière dont une variable est utilisée :

- **Les variables modifiables** permettent de modifier la valeur avant l'exécution de l'application, mais pas pendant l'exécution.
- **Les variables protégées** sont modifiables, mais il faut appuyer sur une touche de modification supplémentaire pour modifier la valeur, puis sur une touche d'enregistrement supplémentaire pour valider la modification.
- **Les variables dynamiques** sont modifiables, et vous pouvez même modifier leurs valeurs de manière dynamique pendant l'exécution.
- **Les variables dynamiques protégées** combinent les propriétés des variables protégées et dynamiques ; elles sont donc modifiables en cours d'exécution en appuyant sur une touche d'édition.
- **Les variables en lecture seule** ne sont pas modifiables ; leur valeur ne peut être modifiée que par l'application. Elles sont généralement utilisées pour surveiller l'état ou la progression d'une application.
- **Les variables masquées** ne s'affichent pas dans l'interface du panneau ("Panel Interface") , mais peuvent être lues et écrites de la même manière que les autres variables d'état.

4.7.5 Variables partagées et globales

En plus des variables d'état spécifiques à l'application décrites ci-dessus, il est possible de définir des variables partagées et globales afin de partager des données entre plusieurs scripts s'exécutant en parallèle ou entre plusieurs sessions d'une même application :

- **Les variables partagées** n'existent que pendant la durée d'exécution d'une seule application. Utilisez les fonctions `set_shared_variable(...)` et `get_shared_variable(...)` pour partager des valeurs entre plusieurs applications principales et/ou applications parallèles.

- **Les variables globales** partagent non seulement des valeurs entre les applications en cours d'exécution, mais elles persistent également après le redémarrage du robot. Pour cela, utilisez les fonctions `set_global_variable(...)` et `get_global_variable(...)`.

4.7.6 Éditeur de dossiers

Si le nom de l'application contient le caractère « / », celle-ci est automatiquement classée dans une structure de dossiers. Vous pouvez modifier un dossier en cliquant sur le bouton d'édition qui apparaît lorsque vous survolez ce dossier. Cela ouvre l'éditeur de dossiers. Cet éditeur offre la possibilité d'afficher le nom du dossier dans la langue d'interface sélectionnée (à l'instar de l'éditeur d'applications). Les titres des dossiers ainsi que ceux des applications apparaissent dans l'interface du panneau ("Panel Interface") ; de cette manière, l'interface du panneau ("Panel Interface") peut être facilement configurée pour être compréhensible en plusieurs langues.

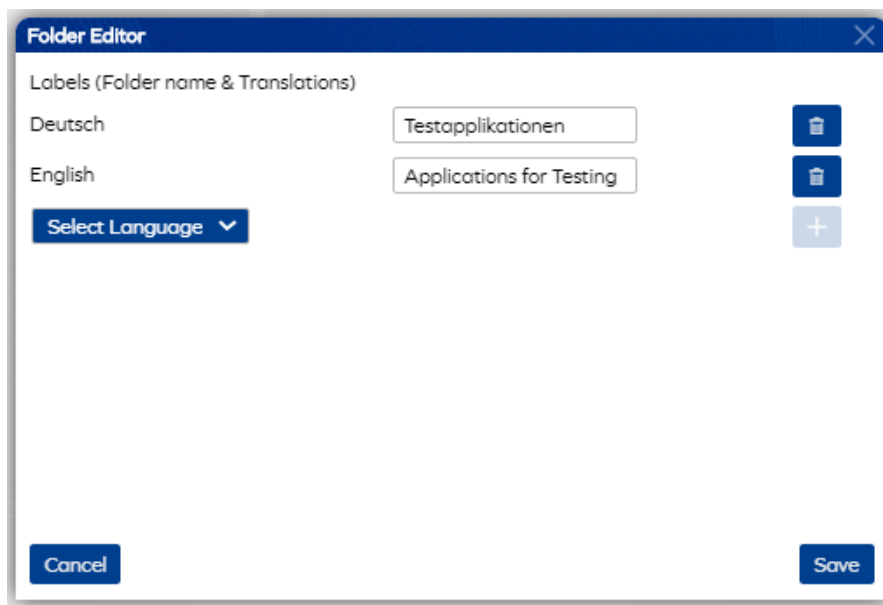


Figure 4.7.6 : Éditeur de dossiers

4.7.7 Programmation par blocs

L'onglet « **Blocs** » de l'éditeur d'application contient une interface de programmation graphique. Cette interface permet à l'utilisateur de créer des séquences de programme simples en glissant-déposant des blocs de programmation dans la zone de programme.

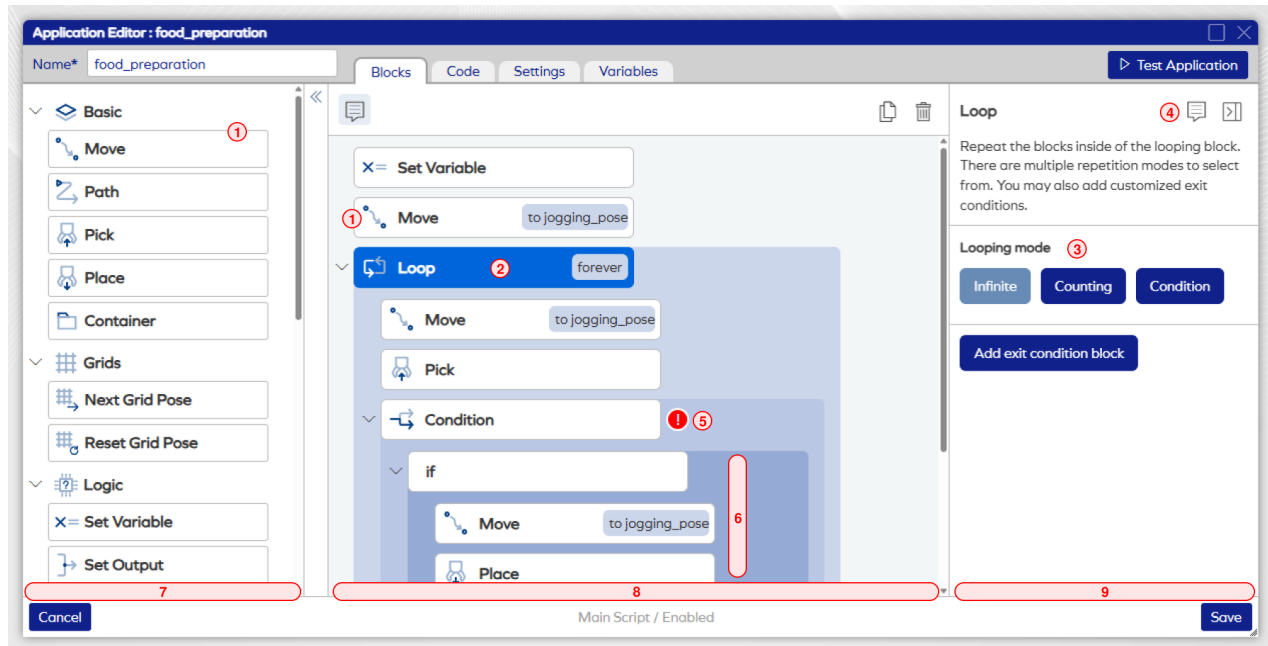


Figure 4.7.7 : Éditeur d'applications (programmation par blocs)

1. Glisser des blocs dans la zone de programme, 2. Sélectionner un bloc à modifier, 3. Modifier les paramètres, 4. Commentaire, 5. Affichage des erreurs, 6. Conteneur de blocs, 7. Blocs disponibles, 8. Zone de programme, 9. Détails du bloc sélectionné

Sélectionnez un bloc dans le déroulement du programme pour en afficher les détails. Certains paramètres doivent obligatoirement être renseignés pour que le programme soit exécutable (par exemple, pour un bloc de mouvement, vous devez indiquer où le robot doit se déplacer). D'autres paramètres sont quant à eux facultatifs (par exemple, des valeurs par défaut sont utilisées si aucune indication n'est fournie) ou peuvent être utilisés pour améliorer la lisibilité du programme (par exemple, les commentaires). Si un bloc n'est pas correctement défini (par exemple, s'il manque un paramètre obligatoire ou si une valeur non valide a été indiquée), il affiche un message d'erreur dans le déroulement du programme et, lorsqu'il est sélectionné, la cause de l'erreur s'affiche dans les détails.

Certains blocs créent, lors de leur ajout, une structure imbriquée sur fond coloré dans le déroulement du programme. Il s'agit d'une aide graphique indiquant que ces blocs sont regroupés et s'exécutent toujours dans l'ordre (de haut en bas). Les boucles et les conditions sont des exemples de tels blocs, qui sont chargés, par exemple, d'exécuter plusieurs fois un groupe de blocs ou de n'exécuter un groupe que si une condition particulière est remplie.

De plus, un bloc sélectionné peut être supprimé ou copié (et déplacé à l'emplacement souhaité dans le programme).

4.7.7.1 Utilisation des variables d'application et des expressions

Avant d'expliquer en détail tous les blocs de programmation, veuillez noter que la plupart des champs de texte des paramètres de bloc prennent en charge les expressions mathématiques et/ou logiques. Ces champs permettent également d'insérer les valeurs de variables d'application prédéfinies ainsi que les valeurs d'entrées numériques et analogiques.

Le plus simple est d'illustrer cela à l'aide d'un exemple. Supposons que nous ayons défini les variables suivantes :

The screenshot shows the 'State Variables' configuration window. It contains two sections: 'Boolean' and 'Integer'.
 In the 'Boolean' section, there is a variable named 'flag' with a value of 0 and a 'Default' checkbox that is unchecked. The 'Style' is set to 'Editable'.
 In the 'Integer' section, there is a variable named 'counter' with a value of 0 and a 'Default' of 0. The 'Min.' is 0 and the 'Max.' is 1000. The 'Unit' is empty. The 'Style' is also set to 'Editable'.

Figure 4.7.7.1.a : Exemples de variables

Exemples d'**expressions numériques** valides :

Four examples of valid numerical expressions are shown in light blue boxes, each preceded by 'value (number)':
 1. 42
 2. -7 * (8.2 - 3)
 3. {counter} + 1
 4. (Tool AIO 02)

Exemples d'**expressions logiques** valides :

Four examples of valid logical expressions are shown in light blue boxes, each preceded by 'value (bool)':
 1. FALSE
 2. NOT {flag}
 3. {flag} AND ({counter} > 5)
 4. {Main DIO 09}

Figure 4.7.7.1.b : Exemple d'expressions numériques et logiques

Les valeurs des variables et des entrées peuvent également être saisies dans des champs de texte normaux :

An example of a text expression is shown in a light blue box, preceded by 'value (text)':
 The counter is set to {counter}.

Figure 4.7.7.1.c : Exemple d'expression textuelle

4.7.7.2 Bloc Mouvement

Le **bloc de mouvement** exécute une fonction de déplacement qui déplace le robot de sa position actuelle vers une destination. L'utilisateur doit sélectionner le type de déplacement...

- **Point à point** → Se déplacer vers la cible de la manière la plus rapide possible, sans restriction de trajectoire
- **Linéaire** → Se déplacer vers la cible le long d'un chemin linéaire, l'orientation est interpolée linéairement
- **Circulaire** → Se déplacer vers la cible le long d'un chemin circulaire, l'orientation est interpolée linéairement, et l'utilisateur doit spécifier **pose intermédiaire** sur le chemin circulaire

... ainsi que le format de la cible.

- **Nom de la pose** → Sélectionnez le nom de la pose cible (précédemment enregistrée)

- **Nom de la grille** → Déplacement vers les points d'une grille (préalablement enregistrée) ; le déplacement s'effectue d'abord vers le premier point de cette grille, puis le bloc « **Pose de grille suivante** » peut être utilisé pour incrémenter le compteur de points de la grille
- **Angles articulaires** → Indiquez les angles articulaires de la pose cible
- **Coordonnées** → Indiquez les coordonnées de la pose cible

De plus, l'utilisateur peut modifier les paramètres optionnels suivants. Veuillez noter que si aucun paramètre n'est spécifié, les valeurs par défaut/globales sont utilisées.

- **Paramètres de vitesse**
 - **Vitesse / accélération des articulations** → Indique la vitesse pour les mouvements point à point
 - **Vitesse / accélération linéaire** → Indique la vitesse pour les mouvements linéaires/circulaires
 - **Vitesse / accélération de rotation** → Indique la vitesse pour les mouvements qui modifient uniquement l'orientation du TCP (la position reste inchangée)
- **Paramètres du système de coordonnées de référence**
 - **Décalage de l'outil** → Décale le TCP par rapport au système de bride (partie supérieure du bras du robot)
 - **Système de coordonnées de l'outil** → Modifie la position du TCP par rapport au système de bride (partie supérieure du bras du robot)
 - **Décalage de travail** → Décale le système de référence des coordonnées par rapport au système de base (face inférieure du bras du robot)
 - **Système de coordonnées de travail** → Modifie le système de référence des coordonnées par rapport au système de base (face inférieure du bras du robot)
- **Autres paramètres**
 - **Coordonnées relatives** → Lorsque cette option est activée, les valeurs de coordonnées sont ajoutées aux coordonnées actuelles
 - **Flux de programme bloqué** → Lorsque cette option est activée, l'exécution du programme ne reprend qu'une fois le mouvement terminé

4.7.7.3 Bloc de trajectoire

Le **bloc de trajectoire** exécute une trajectoire préalablement définie. L'utilisateur sélectionne la trajectoire à exécuter. Lors de l'exécution de ce bloc, le robot se déplace d'abord vers la position initiale de la trajectoire, puis parcourt les segments de la trajectoire en un seul mouvement continu.

4.7.7.4 Bloc « Saisir »

Le bloc « **Saisir** » envoie une commande de maintien à l'outil connecté. Dans le cas de préhenseurs angulaires ou parallèles, cela signifie fermer le préhenseur afin de maintenir l'objet à l'intérieur. Dans le cas de préhenseurs pneumatiques, cela signifie activer la pompe à vide afin de maintenir l'objet en place. Ce bloc ne comporte aucun paramètre.

4.7.7.5 Bloc « Placer »

Le bloc « **Placer** » envoie une commande de placement à l'outil connecté. Pour les préhenseurs angulaires ou parallèles, cela signifie ouvrir le préhenseur afin de libérer l'objet qui s'y trouve. Pour les préhenseurs pneumatiques, cela signifie désactiver la pompe à vide afin de déposer l'objet à sa place. Ce bloc ne comporte aucun paramètre.

4.7.7.6 Bloc « Conteneur »

Le bloc « **Conteneur** » offre un moyen simple de regrouper plusieurs blocs. Aucune fonction particulière ne lui est attribuée.

4.7.7.7 Bloc « Pose de grille suivante »

Le bloc « **Pose de grille suivante** » incrémente le compteur de pose interne pour une grille donnée. La grille concernée doit être sélectionnée. Chaque fois qu'un **bloc de mouvement** est utilisé pour se déplacer vers une grille, la pose visée est celle à laquelle ce compteur fait actuellement référence. Pour atteindre effectivement tous les points d'une grille, vous pouvez utiliser à la fois le **bloc de mouvement** et le bloc « **Pose de grille suivante** » au sein d'une structure de boucle comme suit :

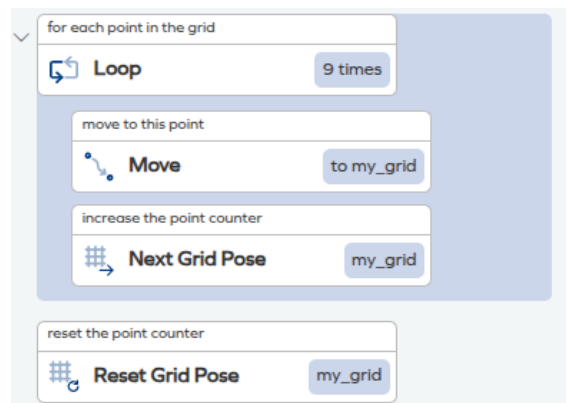


Figure 4.7.7.7 : Exemple d'utilisation des blocs de grille

4.7.7.8 Bloc « Réinitialiser la pose de la grille »

Le bloc « **Réinitialiser la pose de la grille** » remet à zéro (valeur initiale) le compteur de pose interne d'une grille donnée. La grille concernée doit être sélectionnée.

4.7.7.9 Bloc « Écriture de variable »

Le bloc « **Écriture de variable** » définit la valeur d'une variable d'application prédéfinie. Il est impératif de sélectionner d'abord une variable, puis une valeur. L'utilisation d'expressions numériques et/ou logiques est autorisée.

4.7.7.10 Bloc « Définir la sortie »

Le bloc « **Définir la sortie** » définit la valeur d'une sortie numérique ou analogique. L'utilisateur doit d'abord sélectionner la sortie à définir ; selon qu'il s'agit d'une sortie numérique ou analogique, la valeur à définir est une valeur booléenne ou un nombre.

4.7.7.11 Bloc Attendre

Le **bloc d'attente** bloque l'exécution du programme jusqu'à ce qu'un événement spécifique se produise. Trois modes différents sont disponibles :

- **Temps** → Attendre qu'un certain laps de temps se soit écoulé (expression numérique autorisée)
- **Mouvement** → Attendre qu'une commande de mouvement en cours s'arrête

- **Entrée** → Attendre que la valeur d'une entrée numérique soit définie sur une valeur spécifique ou que la valeur d'une entrée analogique dépasse un seuil défini

4.7.7.12 Bloc « Boucle »

Le bloc « **Boucle** » crée un groupe de blocs qui est appelé de manière répétée. Trois modes différents sont disponibles :

- **Infini** → Boucle qui s'exécute jusqu'à ce que l'application soit interrompue
- **Compteur** → Boucle pour un nombre défini de répétitions (expression numérique autorisée)
- **Condition** → Boucle jusqu'à ce qu'une condition spécifique soit remplie (expression logique autorisée)

L'utilisateur peut ajouter un bloc « **Sortie si** » à la boucle, voir la section suivante.

4.7.7.13 Bloc « Sortie si »

Le bloc « **Sortie si** » est lié à son **bloc de boucle** et sort de cette boucle lorsque la condition spécifiée est remplie (expression logique autorisée).

4.7.7.14 Bloc « conditionnel »

Le bloc « **conditionnel** » crée un conteneur comprenant deux groupes « if » et « else ». La condition (expression logique autorisée) est utilisée pour diriger l'exécution du programme soit vers le groupe « if » (si la condition est vraie), soit vers le groupe « else » (si la condition n'est pas vraie).

4.7.7.15 Bloc de commentaire

Le **bloc de commentaire** permet simplement d'insérer un commentaire informatif dans le déroulement du programme. Notez que chaque autre bloc offre également la possibilité d'ajouter un commentaire, qui est alors visible dans la zone de programme. Il est toutefois parfois utile de créer un commentaire distinct qui n'est pas lié à un bloc particulier.

4.7.7.16 Bloc « Quitter »

Le bloc « **Quitter** » interrompt le programme en cours d'exécution. Les blocs de programmation restants ne sont plus exécutés.

4.7.7.17 Bloc « Boîte de dialogue »

Le **bloc de dialogue** crée un conteneur renfermant plusieurs groupes. Lors de l'exécution, le déroulement du programme est interrompu et une boîte de dialogue s'affiche. L'utilisateur doit répondre à la boîte de dialogue pour que le programme puisse se poursuivre. Les groupes qu'il contient (par défaut « OK » et « Annuler ») correspondent aux boutons de la boîte de dialogue. En fonction du bouton sur lequel l'utilisateur a cliqué, le déroulement du programme bifurque et se poursuit avec le groupe sélectionné. Le message de la boîte de dialogue (injection de variables autorisée), les noms des boutons et le nombre de boutons peuvent être modifiés dans les paramètres.

4.7.7.18 Bloc de sortie

Le **bloc de sortie** affiche un message défini par l'utilisateur dans la sortie texte.

4.7.7.19 Bloc de journalisation

Le **bloc de journalisation** enregistre un message défini par l'utilisateur dans un fichier journal personnalisé. Les fichiers journaux peuvent être téléchargés via l'interface du bureau, par exemple à l'aide de la combinaison de touches CTRL + MAJ + L.

4.7.7.20 Bloc de notification

Le **bloc de notification** envoie une notification personnalisée à l'interface graphique. La notification ne bloque pas l'exécution du programme.

4.7.7.21 Bloc d'avertissement

Le **bloc d'avertissement** envoie un message d'avertissement défini par l'utilisateur à l'interface graphique. L'avertissement ne bloque pas l'exécution du programme.

4.7.7.22 Bloc d'erreur

Le **bloc d'erreur** envoie un message d'erreur défini par l'utilisateur à l'interface graphique et interrompt le programme en cours d'exécution.

4.7.7.23 Bloc de code Python

Le **bloc de code Python** est un bloc avancé qui permet à l'utilisateur d'appeler quelques lignes de code Python autonomes. Vous pouvez utiliser ce bloc pour appeler des fonctions qui ne sont pas disponibles via la programmation par blocs. Toutes les fonctions peuvent y être utilisées.

4.8 Menu E/S

Appuyez sur le bouton « **I/O** » dans la fenêtre principale pour accéder au menu I/O. Ce menu affiche toutes les entrées et sorties numériques et analogiques ainsi que leurs propriétés, qui ont été préalablement définies dans la fenêtre de configuration des E/S (leur configuration, les fonctions attribuées et les noms personnalisés), ainsi que les valeurs de toutes les entrées et sorties. Les entrées sont en lecture seule ; ces valeurs peuvent donc simplement être surveillées ici. Les sorties, en revanche, peuvent également être définies via cette fenêtre.

Les E/S numériques représentent des valeurs booléennes, où **True ou 1 ou « vert »** signifie « **sous tension** » et **False ou 0 ou « gris »** signifie « **hors tension** ». Les entrées numériques affichent cette valeur dans un indicateur circulaire, tandis que les sorties numériques peuvent être modifiées en appuyant sur le bouton d'affichage carré pour changer leur état.

Les E/S analogiques sont représentées sous forme de **valeurs à virgule flottante** par incréments de 0,1. L'unité d'une E/S analogique est définie par la configuration de cette E/S (V pour la tension, mA pour le courant). Alors que les entrées analogiques sont représentées sous forme de nombre observable, les sorties analogiques peuvent être modifiées via les champs de texte situés à côté de la valeur actuelle de cette sortie.

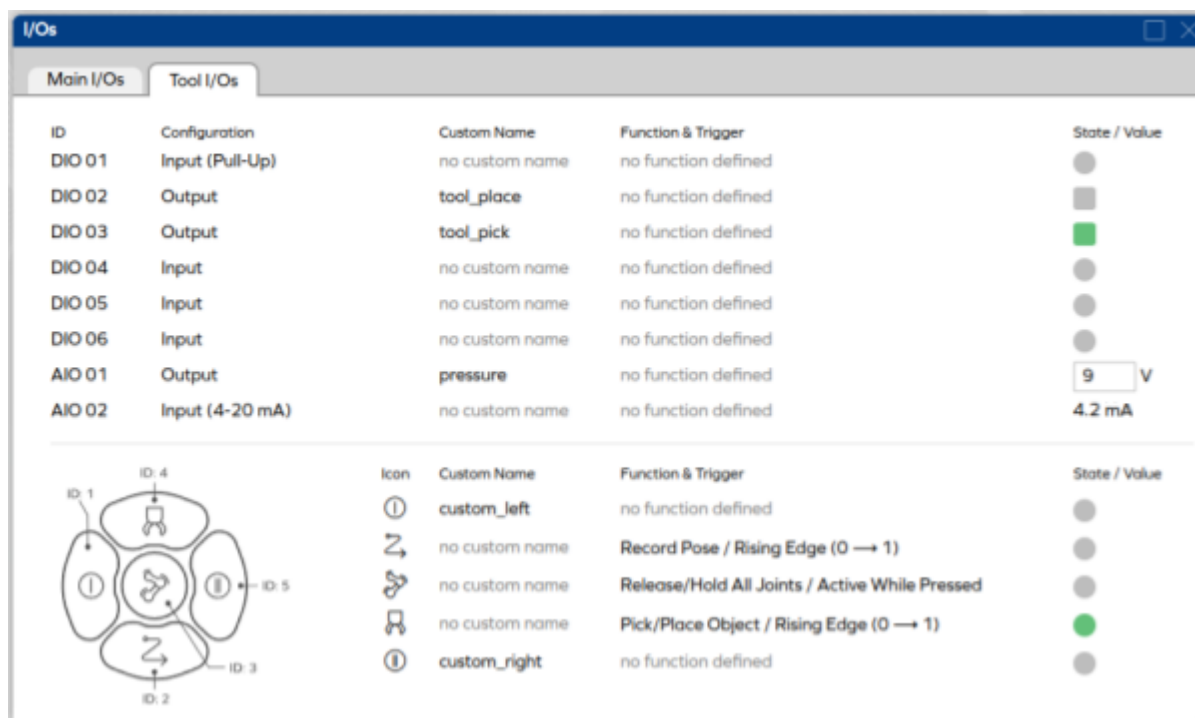


Figure 4.8 : Menu E/S

4.9 Menu Visualisation

Appuyez sur le bouton « **Visualisation** » dans la fenêtre principale pour accéder au menu Visualisation. Cette fenêtre affiche la position et l'orientation actuelles des articulations, ce qui est utile, par exemple, lorsque le robot est commandé à distance sans retour visuel direct. Vous pouvez également voir les poses précédemment enregistrées dans la visualisation 3D. La visualisation est interactive et permet certaines commandes :

- Faites **pivoter** l'image en cliquant avec votre souris à l'intérieur de la visualisation (bouton gauche) et en faisant glisser l'image latéralement ainsi que vers le haut et vers le bas.
- **Zoomez** et **dézoomez** en tournant la molette de votre souris vers le haut ou vers le bas.
- **Déplacez** la caméra latéralement ainsi que vers le haut et vers le bas à l'aide des touches fléchées de votre clavier, ou cliquez avec le bouton droit de la souris et faites glisser vers la gauche et vers la droite.

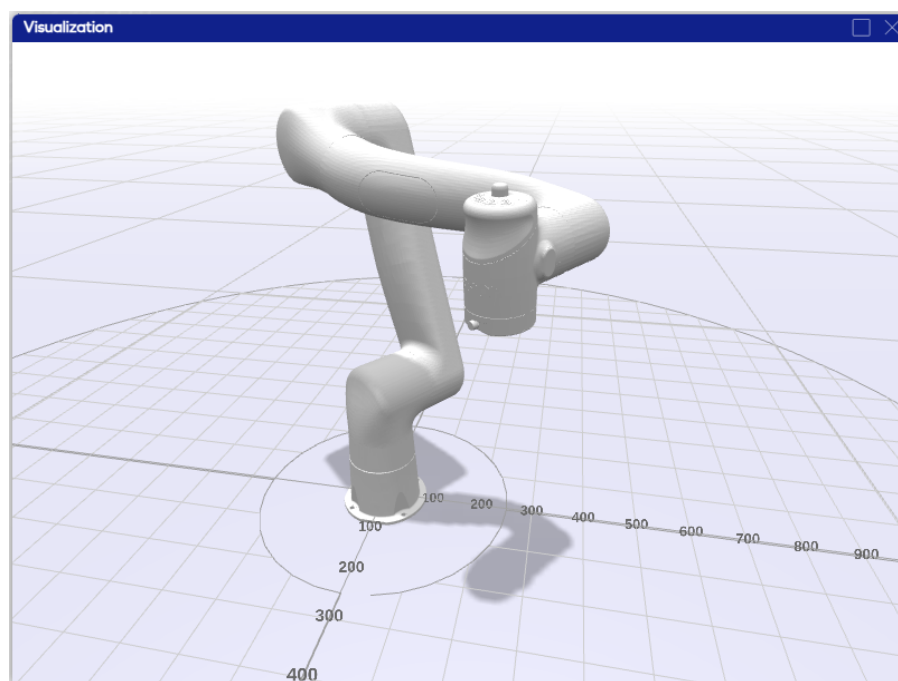


Figure 4.9 : Fenêtre d'affichage

4.10 Commande manuelle

Appuyez sur le bouton « **Commande manuelle** » dans la fenêtre principale pour afficher le menu de commande manuelle. Dans ce menu, l'utilisateur peut définir quelles articulations du robot sont autorisées à bouger pendant la commande manuelle. Par défaut, le fait d'appuyer sur le bouton HGC du robot (et sur le bouton correspondant dans le cockpit) permet à l'utilisateur de déplacer toutes les articulations. En cliquant sur les indicateurs de chaque articulation dans ce menu, l'utilisateur peut activer ou désactiver des articulations individuelles pour le mode manuel.

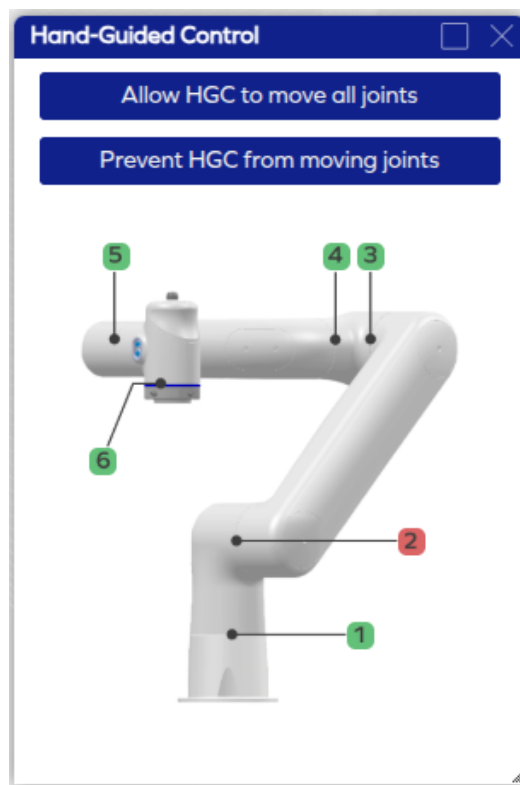


Figure 4.10 : Menu pour le mode manuel

4.11 Moniteur système

Dans la fenêtre principale, cliquez sur le bouton « **Moniteur système** » pour afficher le moniteur système. Ce menu contient les informations système suivantes que l'utilisateur peut surveiller :

4.11.1 État du robot

L'onglet « **État du robot** » affiche l'état actuel du robot. Cela comprend les coordonnées du point central de l'outil (TCP) par rapport au système de base du robot, ainsi que les positions, les courants et les valeurs de température de tous les moteurs. De plus, les états de toutes les entrées de sécurité y sont visibles.

4.11.2 Sortie texte

Dès qu'une application renvoie du texte à l'aide de la fonction **print(...)**, ce texte apparaît dans l'onglet « **Sortie texte** » de cette fenêtre. Veuillez noter que lorsque plusieurs applications s'exécutent en parallèle, toutes les applications peuvent écrire du texte dans la sortie texte.

4.11.3 Journal des applications

Dès qu'une application enregistre des données à l'aide de la fonction **log_message(...)**, ces données s'affichent dans l'onglet « **Journal d'application** » de cette fenêtre. Vous pouvez y filtrer et rechercher les données enregistrées. Ces journaux personnalisés sont téléchargés avec les fichiers journaux intégrés lorsque vous utilisez la combinaison de touches CTRL + MAJ + L.

4.11.4 Historique des messages

Dès qu'un message s'affiche dans l'interface utilisateur, il apparaît également dans l'onglet « **Historique des messages** » de cette fenêtre. Les messages sont classés en quatre types, utilisés dans différents contextes :



Les messages d'information fournissent à l'utilisateur des indications utiles, des communications informatives ou des conseils.



Les messages de réussite indiquent à l'utilisateur qu'une commande ou une action a été exécutée avec succès.



Les avertissements fournissent à l'utilisateur des informations importantes qui doivent être lues avec attention avant de poursuivre. Les avertissements ne sont jamais critiques et n'interrompent pas le déroulement du programme.



Il existe un large éventail **d'erreurs**, allant de simples erreurs de manipulation (par exemple, des actions ou des arguments de fonction incorrects) à des erreurs critiques causées, par exemple, par un robot endommagé.

4.12 Menu Aide

Le **menu Aide** contient des informations utiles sur le système actuellement en cours d'exécution et les modules installés, sur les raccourcis clavier à la disposition de l'utilisateur, ainsi que sur les moyens de contact en cas de questions ou de problèmes.

4.12.1 Informations sur le robot

L'onglet « **Informations sur le robot** » du menu Aide contient des informations détaillées sur l'environnement AX Portal et les composants matériels connectés (le cas échéant). Dans la section « **Modules** » des informations sur le robot, vous trouverez une liste des modules installés ainsi que des informations indiquant s'ils sont correctement chargés. Ce n'est que lorsque le statut « OK » s'affiche que vous pouvez utiliser les fonctions et les menus fournis par ce module. Pour toute demande d'assistance, veuillez toujours indiquer les informations détaillées sur le robot provenant d'AX Portal et du matériel connecté. Il vous suffit d'appuyer sur le bouton « Copier dans le presse-papiers » et de coller le contenu (de préférence dans une [police à espacement](#) fixe) avec votre demande d'assistance.

4.12.2 Raccourcis clavier

L'onglet « **Raccourcis clavier** » contient un aperçu des raccourcis clavier disponibles. Les raccourcis clavier sont divisés en raccourcis généraux, utilisés par exemple pour actionner les boutons de commande dans le cockpit, et en raccourcis pour l'éditeur d'applications, utiles pour la programmation d'applications AX Portal. Veuillez noter que la disposition du clavier sous macOS diffère considérablement de celle des autres systèmes d'exploitation (Windows, Linux) et que la liste des raccourcis clavier est donc également différente.

4.12.3 Documentation du code

L'onglet « **Documentation du code** » contient la documentation de toutes les fonctions disponibles pour la programmation des applications. La documentation est divisée en fonctions de base, qui sont toujours disponibles, et en fonctions supplémentaires, qui se rapportent à des modules installés en plus. Le bouton « **Télécharger** » situé en haut de l'onglet permet de télécharger la documentation du code sous forme de fichier ZIP. Pour consulter la documentation du code hors ligne, ouvrez le fichier « index.html » dans votre navigateur.

4.12.4 Licence open source

L'onglet « **Licence open source** » contient toutes les informations nécessaires concernant les bibliothèques et modules open source utilisés par AX Portal.

4.12.5 Coordonnées

L'onglet « **Contact** » affiche les coordonnées du service d'assistance. Veuillez toujours consulter les manuels avant de soumettre une demande d'assistance. Lors de vos demandes d'assistance, assurez-vous de télécharger les fichiers journaux et de les joindre à votre demande.

4.13 Menu des paramètres

Dans le **menu des paramètres**, l'utilisateur peut modifier de manière permanente certains paramètres de base d'AX Portal. Cliquez sur « **Enregistrer** » pour appliquer toutes les modifications, sur « **Annuler** » pour les rejeter ou sur « **Réinitialiser** » pour rétablir tous les paramètres aux dernières valeurs enregistrées. Toutes les catégories et options sont expliquées ci-dessous :

The screenshot shows the 'Settings' window with the following sections and options:

- GUI Settings**
 - Live Auto-Completion: ☒
 - Show Optional Function Arguments: ☒
 - Message Fading Level: Warning (dropdown)
 - Message Fading Time: 5 (slider)
- Customize Cockpit Buttons**
 - Show 'Hand-Guided Control' button: ☒
 - Show 'Gripper' button: ☒
 - Show 'Jogging' button: ☒
 - Show 'Motion Controls' button: ☒
 - Show 'Homing' button: ☒
 - Homing Position
 - Joint 1: 0
 - Joint 2: 0
 - Joint 3: 0
 - Joint 4: 0
 - Joint 5: 0
 - Joint 6: 0
- Start-Up Settings**
 - Start-Up Project: Last Selected (dropdown)
 - Start-Up Application: Last Selected (dropdown)
 - Triggering Events: No Trigger (dropdown)
- Motion Behaviors**
 - Interrupt Behavior: Pause and Approve (dropdown)
 - Collision Resume Attempts: 3 (slider)
 - Collision Resume Timeout: 20 (slider)
 - Temperature Speed Reduction: ☒
- Authentication Settings**
 - Disable Authentication for ROS: ☒
 - Disable Authentication for TCP: ☒
 - Disable Authentication for REST: ☒
 - Disable Authentication for Modbus: ☒
- Default TCP Communication Settings**
 - Encoding Standard: UTF-8 (text)
 - End Character: \x03 (text)
 - Send Heartbeat: ☒
- ROS Function Settings**
 - Discovery Mode: Peer to Peer (dropdown)
 - IP Addresses of ROS 2 Peers: 127.0.0.1 (text, with add/remove buttons)
 - ROS Domain ID: 10 (dropdown)

Buttons at the bottom: Cancel, Reset, Save.

Figure 4.13 : Menu des paramètres

4.13.1 Paramètres de l'interface graphique

Cette catégorie contient plusieurs paramètres relatifs à l'interface utilisateur.

Saisie semi-automatique	Activez cette option pour voir s'afficher en temps réel des suggestions de complétion automatique du texte lors de la saisie de code. Si cette option est désactivée, vous pouvez néanmoins compléter automatiquement le mot en cours à l'aide du raccourci clavier CTRL+ESPACE.
Afficher les arguments facultatifs	Choisissez si les arguments facultatifs des fonctions doivent être insérés ou non lors de l'insertion de fonctions dans le code. Afin d'éviter d'encombrer le code avec des arguments de fonction inutiles, nous vous recommandons de désactiver cette option. Toutefois, si vous n'êtes pas encore familiarisé avec les fonctions intégrées, nous vous recommandons de l'activer.
Niveau de masquage des messages	Choisissez jusqu'à quel niveau les messages qui s'affichent doivent être masqués. Pour que tous les messages soient masqués, sélectionnez le niveau le plus élevé possible. Pour que tous les messages restent affichés en permanence, sélectionnez le niveau le plus bas possible.
Durée de masquage des messages	Indiquez la durée de masquage des messages à masquer en secondes (entre 1 et 60).

4.13.2 Personnaliser les boutons du cockpit

Ces paramètres permettent à l'utilisateur d'activer ou de désactiver très facilement les boutons de commande apparaissant dans le cockpit de l'interface utilisateur graphique, tant pour l'interface de bureau que pour l'interface de panneau ("Panel Interface"). Le fonctionnement détaillé de ces boutons est expliqué dans la section « Cockpit ». Il est judicieux de désactiver un bouton si l'utilisateur ne souhaite pas l'utiliser ou ne peut pas l'utiliser (par exemple, vous ne pouvez pas utiliser le bouton de préhension si aucun outil interactif n'est connecté), ou si l'utilisateur ne souhaite pas qu'un opérateur l'utilise via l'interface du panneau ("Panel Interface") .

Afficher le bouton *	Activez cette option pour afficher ce bouton et désactivez-la pour le masquer.
Position de référence	Indiquez les angles de la position de retour à l'origine, c'est-à-dire la position associée au bouton « Retour à l'origine » accessible dans le cockpit. Saisissez une valeur pour chaque angle d'articulation.

4.13.3 Paramètres de démarrage du programme

Dans cette catégorie, tous les paramètres concernent le processus de démarrage d'AX Portal. En ajustant ces paramètres, vous pouvez éviter des actions répétitives lors du démarrage. Pour certains paramètres de démarrage du programme, sélectionnez l'entrée de liste « **Dernière sélection** » (entrée par défaut) afin d'enregistrer et de recharger le dernier paramètre utilisé.

Projet sélectionné	Sélectionnez le projet qui sera chargé au démarrage du programme. Tous les projets définis apparaissent dans cette liste.
Application sélectionnée	Une fois un projet sélectionné, choisissez l'application qui sera chargée au démarrage du programme. Toutes les applications principales définies pour ce projet apparaissent dans cette liste.
Événement déclencheur	Choisissez si le robot doit être mis en service automatiquement ou s'il doit même exécuter une application. Par défaut, cette option est désactivée. Pour sélectionner un déclencheur, il faut d'abord choisir certains des paramètres de démarrage mentionnés ci-dessus afin d'éviter que l'utilisateur n'exécute par inadvertance des applications non souhaitées.

4.13.4 Comportement de mouvement

Dans cette catégorie, tous les paramètres concernent le comportement pendant les mouvements du robot.

Comportement en cas d'interruption	Sélectionnez le comportement de déplacement en cas d'interruption. Vous trouverez plus de détails sur les différents comportements en cas d'interruption et sur la manière dont celles-ci sont résolues dans les sous-sections suivantes « Comportement en cas d'interruption » et « Boîte de dialogue d'interruption ».
Reprise après collision, nombre de tentatives et délai d'attente	Définissez le nombre maximal de tentatives pour reprendre le mouvement après une collision. Si trop de collisions se produisent pendant le délai défini, le robot interrompt la reprise.
Réduction en cas température	Choisissez si la vitesse du robot doit être automatiquement réduite en cas de températures élevées des moteurs. Ce paramètre est activé par défaut, ce qui entraîne un ralentissement automatique des mouvements au-delà de 70 °C afin d'éviter toute surchauffe. Si des températures supérieures à 90 °C sont détectées, une erreur critique est déclenchée afin d'empêcher tout dommage dû à la surchauffe. Vous trouverez plus de détails sur cette fonction dans la section « Limites de performance » du « Robot Collaboratif 6 Axes AX6 - Manuel ».

4.13.4.1 Comportement en cas d'interruption

Des interruptions peuvent survenir pour diverses raisons, par exemple :

- un événement d'arrêt SS1 a été déclenché
- un événement d'arrêt SS2 a été déclenché
- une collision du robot avec une personne, un objet ou l'environnement s'est produite
- une interruption du dispositif de protection a été déclenchée

Une collision est détectée dès que les courants des axes du robot s'écartent du modèle dynamique du robot au-delà d'un seuil défini. La détection de collision est une fonction de sécurité qui offre un niveau de protection supplémentaire ; elle ne doit pas être utilisée intentionnellement comme élément

d'une application, car des collisions répétées sollicitent davantage les axes du robot et accélèrent leur usure.

Veuillez noter que pour un fonctionnement correct de la détection de collision, le programmeur doit ajuster la charge utile du robot à la valeur correcte dès qu'un objet est saisi ou déposé.

Outre les collisions physiques, des dispositifs de sécurité externes tels que des barrières lumineuses ou des dispositifs de protection similaires peuvent être connectés au robot. Ces déclencheurs agissent de manière préventive avant qu'un contact physique ne se produise, mais sont traités de la même manière que les événements de collision. Le comportement d'interruption définit la manière dont le système traite les interruptions déclenchées tant par des collisions que par des dispositifs de protection.

Actuellement, voici la procédure recommandée pour le comportement en cas d'interruption :

Mettre en pause et confirmer	Ce paramètre provoque la mise en pause de l'application en cours en cas d'interruption et attend la confirmation de l'utilisateur (fenêtre de dialogue) pour poursuivre ou arrêter l'application.
-------------------------------------	---

4.13.4.2 Boîte de dialogue d'interruption

La boîte de dialogue d'interruption s'affiche dès qu'une condition d'interruption survient pendant le fonctionnement du robot. Les déclencheurs typiques d'interruptions sont les collisions détectées, le déclenchement d'un arrêt d'urgence, le déclenchement d'un dispositif de protection ou d'autres événements liés à la sécurité. La boîte de dialogue guide l'utilisateur de manière sûre et contrôlée pour remédier à la situation.

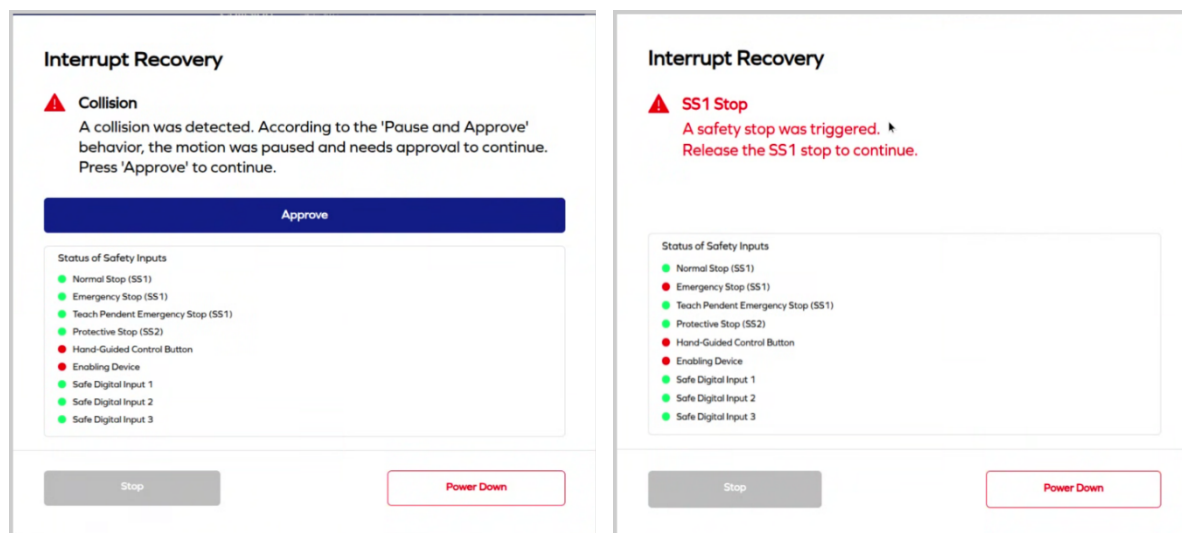


Figure 4.13.4.2 : Rétablissement après une interruption

La boîte de dialogue est divisée en deux zones principales :

Zone supérieure – Informations sur l'interruption et la résolution : cette zone décrit la condition d'interruption actuellement détectée et fournit des instructions claires pour y remédier. En voici quelques exemples :

- S'assurer que le robot peut continuer à se déplacer après une collision.
- Débloquer un arrêt d'urgence ou d'autres dispositifs de sécurité.
- Réinitialiser une condition de sécurité, si nécessaire.

En fonction du type d'interruption, un bouton d'action tel que « **Confirmer** », « **Continuer** » ou « **Réinitialiser** » s'affiche. L'utilisateur doit suivre les instructions et appuyer sur le bouton indiqué pour reprendre le fonctionnement.

Zone inférieure – État des entrées de sécurité : la moitié inférieure de la boîte de dialogue affiche l'état en temps réel de toutes les entrées de sécurité pertinentes. Cela permet à l'utilisateur d'identifier rapidement quel dispositif de sécurité est actuellement actif, par exemple :

- Arrêt d'urgence (y compris l'arrêt d'urgence sur le boîtier de commande manuelle)
- Arrêt de protection
- Dispositif d'autorisation
- Bouton de guidage manuel
- Entrées numériques de sécurité

Les voyants verts indiquent un état normal, tandis que les voyants rouges signalent un état actif ou bloqué (ou que le bouton n'est pas enfoncé).

Option d'arrêt

Le bouton « **Arrêt** » est toujours disponible lorsque l'utilisateur doit choisir entre poursuivre l'application/le mouvement précédent et s'arrêter. En appuyant sur ce bouton, le robot s'arrête et revient à l'état « Prêt », dans lequel il est prêt pour de nouvelles actions.

Option « Mettre hors service »

Le bouton « **Mettre hors service** » est toujours disponible dans la boîte de dialogue d'interruption. Il permet d'arrêter le robot à tout moment. Veuillez noter que l'arrêt ne résout pas automatiquement la condition d'interruption ; la cause sous-jacente (par exemple, un arrêt d'urgence activé) doit éventuellement encore être éliminée avant que le fonctionnement normal puisse reprendre après le redémarrage.

Déroulement typique

1. Une interruption se produit et la boîte de dialogue s'affiche automatiquement.
2. Lisez la description de l'état détecté dans la partie supérieure.
3. Vérifiez l'état des entrées de sécurité dans la partie inférieure afin d'identifier la cause active.
4. Réglez manuellement l'état (par exemple, déverrouiller l'arrêt d'urgence, retirer l'obstacle).
5. Appuyez sur le bouton d'action affiché (« **Confirmer** », « **Continuer** » ou « **Réinitialiser** ») pour poursuivre.
6. Répétez l'action si nécessaire jusqu'à ce que le robot reprenne son fonctionnement normal.

Violation de la SLP

La violation SLP (Safe Limited Position) constitue un cas particulier de reprise après interruption ; elle se produit lorsque le robot tente de se déplacer en dehors des limites de position définies dans les

paramètres de sécurité. Dans ce cas, la boîte de dialogue d'interruption invite l'utilisateur à passer en mode manuel, ce qui lui permet de sortir de la limite de position par des mouvements de jogging. À l'exception du déplacement par petits pas, aucun mouvement n'est autorisé tant que le robot se trouve en situation de violation SLP. Si le robot est arrêté ou démarre avec une violation SLP active, procédez comme suit :

1. Passez en mode manuel
2. Mettez le robot en service
3. Accédez au contrôle de précision
4. Déplacez le robot hors de ses limites de position à l'aide de la commande fine

4.14 Menu des paramètres de sécurité

Le menu des paramètres de sécurité permet d'accéder au système de sécurité des robots de la série AX. Cette interface est destinée aux **responsables de la sécurité**. Les autres utilisateurs sont autorisés à consulter et à exporter les paramètres de sécurité, mais ne peuvent pas les modifier.

4.14.1 Interface utilisateur

Lorsque vous accédez aux **paramètres de sécurité**, ce menu affiche tous les paramètres de sécurité actuellement définis dans le système de sécurité du robot. Tout utilisateur peut consulter ces paramètres, mais seul un **responsable de la sécurité** est autorisé à les modifier. Ce menu est divisé en sous-menus, dans lesquels vous pouvez naviguer à l'aide des numéros de page ou des boutons Précédent et Suivant situés au bas de la fenêtre.

Exporter	En cliquant sur le bouton « Exporter », les paramètres de sécurité sont exportés sous forme de fichier *.yaml. Ce fichier peut être enregistré pour sauvegarder les paramètres de sécurité et également importé sur d'autres robots afin de dupliquer la configuration de sécurité.
Modifier	Le responsable de la sécurité peut cliquer sur le bouton « Modifier » (et saisir son mot de passe utilisateur) pour accéder au mode d'édition.

Dès que le menu est en mode édition, les paramètres de tous les sous-menus deviennent modifiables. Le responsable de la sécurité peut alors adapter les limites de sécurité en fonction de la configuration de montage et de l'application du robot. Pour permettre de voir plus clairement quels paramètres de sécurité ont été modifiés, les champs modifiés sont surlignés en bleu.

Importer	En cliquant sur le bouton « Importer », vous pouvez charger un fichier précédemment exporté contenant les paramètres de sécurité. Cela écrasera les valeurs modifiées manuellement dans tous les sous-menus.
Appliquer	En cliquant sur le bouton « Appliquer » (et en saisissant le mot de passe utilisateur), tous les paramètres de sécurité sont envoyés au système de sécurité, qui vérifie les options et les valeurs limites. Si la vérification échoue, les entrées non valides s'affichent en rouge dans le menu, ce qui permet de les repérer facilement. Si la vérification aboutit, le menu passe en mode de vérification. Veuillez noter qu'à ce stade, les valeurs sont enregistrées, mais pas encore vérifiées. La vérification des paramètres est nécessaire pour mettre le robot sous tension.
Annuler	En appuyant sur le bouton « Annuler », les paramètres de sécurité ne sont pas enregistrés et toutes les modifications sont rejetées. Le menu revient ensuite en mode d'observation (par défaut).

Dès que le responsable de la sécurité a terminé la modification des paramètres de sécurité et que toutes les valeurs ont été enregistrées avec succès, ce menu se trouve en mode de vérification. Les paramètres précédemment enregistrés, mais non encore vérifiés, sont surlignés en jaune.

Vérifier	En appuyant sur le bouton « Vérifier la page », le responsable de la sécurité marque officiellement les paramètres du sous-menu actuellement visible comme acceptés, et le sous-menu suivant est sélectionné. Appuyer sur le bouton « Terminer la vérification » du dernier sous-menu constitue la dernière étape pour enregistrer les paramètres.
-----------------	--

Après avoir vérifié les nouvelles valeurs, il est impératif de valider le comportement du système de

sécurité. Cela peut être effectué en démarrant en mode de validation de sécurité, ce qui permet au logiciel d'ignorer toutes les limites de sécurité pendant le processus de validation afin de vérifier si le système de sécurité se déclenche comme prévu.

Safety Settings

Safely Limited Position (SLP)

Defining the areas in the robot's base frame in which the safety settings are applied. ⓘ

Note that a safety margin of **60 mm** is subtracted from the specified position limits to determine the effective limits.

	Default		Safe Case 1		Safe Case 2		Safe Case 3		
	min	max	min	max	min	max	min	max	
x	-500	3000	-3000	3000	-3000	3000	-3000	3000	mm
y	-800	800	-3000	3000	-3000	3000	-3000	3000	mm
z	-3000	3000	100	-100	-3000	3000	-3000	3000	mm

Defines whether the limiting area applies to the tool center point (TCP) or all movable parts of the robot.

Limit only the TCP of the robot ☒ Limit all parts of the robot

Tool Safety Zone

Defining a safety zone for the tool and any objects it holds. This zone, represented as a sphere with a specified radius, is offset from the center of the flange frame. This zone cannot enter the workspace boundaries. ⓘ

Tool Safety Offset mm Tool Safety Radius mm

Cancel < 1 2 3 4 5 6 > Import Apply

Figure 4.14.1 : Menu des paramètres de sécurité pendant la modification des paramètres

4.14.2 Paramètres de sécurité

La plupart des paramètres de sécurité (dans l'ensemble des sous-menus) sont répartis en 4 colonnes. Ces colonnes définissent les **cas de sécurité** du robot. Un cas de sécurité est déclenché lorsque l'**entrée numérique de sécurité** correspondante (SDI 1-3) est activée. Les **paramètres par défaut** sont toujours actifs, indépendamment des entrées numériques de sécurité.

Les sorties numériques de sécurité (SDO) ne sont pas configurables et leurs spécifications électriques sont décrites dans le « Contrôleur de Robot RC-A101 - Manuel ».

Pour certains réglages, une marge de sécurité est définie. Cette valeur indique la plage autorisée dans laquelle le logiciel peut se déclencher avant le système de sécurité afin d'éviter une violation de la sécurité.

4.14.2.1 Position limite de sécurité (SLP)

Le premier sous-menu des paramètres de sécurité contient la position limitée en toute sécurité. Vous pouvez y limiter la position du robot dans les directions x, y et z, par rapport au système de référence du robot. Si les systèmes de référence ne sont pas clairs, un clic sur l'icône d'information affiche une image avec les coordonnées. Vous pouvez soit limiter la position de toutes les parties du robot (y compris la base et les bras), soit uniquement la position du point central de l'outil (TCP) du robot.

Safety Settings

Safely Limited Position (SLP)

Defining the areas in the robot's base frame in which the safety settings are applied. ⓘ

Note that a safety margin of 60 mm is subtracted from the specified position limits to determine the effective limits.

	Default		Safe Case 1		Safe Case 2		Safe Case 3		
	min	max	min	max	min	max	min	max	
x	-3000	3000	-3000	3000	-3000	3000	-3000	3000	mm
y	-3000	3000	-3000	3000	-3000	3000	-3000	3000	mm
z	-3000	3000	-3000	3000	-3000	3000	-3000	3000	mm

Defines whether the limiting area applies to the tool center point (TCP) or all movable parts of the robot.

Limit only the TCP of the robot ☐ Limit all parts of the robot ☒

Tool Safety Zone

Defining a safety zone for the tool and any objects it holds. This zone, represented as a sphere with a specified radius, is offset from the center of the flange frame. This zone cannot enter the workspace boundaries. ⓘ

Tool Safety Offset mm Tool Safety Radius mm

Export < 1 2 3 4 5 6 > **Edit** **Start Verification**

Figure 4.14.2.1 : Exemples de paramètres pour SLP

4.14.2.2 Angles d'articulation limités en toute sécurité

Le deuxième sous-menu des paramètres de sécurité contient les angles d'articulation limités en toute sécurité. Par défaut, les angles d'articulation limités en toute sécurité sont désactivés. L'activation d'un cas de sécurité permet de surveiller la position angulaire des articulations.

Dans le cadre d'un cas de sécurité, la plage de surveillance des angles d'articulation est toutefois limitée à la plage $[-180^\circ, 180^\circ]$. Cela permet de restreindre la zone de travail du robot par rapport à ses limites physiques réelles. Pour éviter cette restriction, la surveillance peut être désactivée individuellement pour les axes 1, 4 et 6.

Les angles d'articulation limités en toute sécurité ne sont actifs que si les limites inférieure et supérieure sont toutes deux définies. Si les deux limites sont laissées vides, la surveillance de l'articulation correspondante est désactivée. Veuillez noter que les limites inférieure et supérieure doivent soit être toutes deux laissées vides, soit être toutes deux définies.

Safety Settings

Safely Limited Joint Angles

Defining the position limits of each joint of the robot. ⓘ

The default position limits are determined by the physical constraints of the joints and cannot be modified.

Note that some joints can move beyond **180 deg**, but safe cases do not support these ranges. Disabling the limits allows the full joint range to be used.

Note that a safety margin of **5 deg** is subtracted from the specified position limits to determine the effective limits.

	Default		Safe Case 1		Safe Case 2		Safe Case 3		deg
	min	max	min	max	min	max	min	max	
Joint 6	Disabled		Dis.	Dis.	Dis.	Dis.	Dis.	Dis.	
Joint 5	Disabled		-160	160	-160	160	-160	160	
Joint 4	Disabled		Dis.	Dis.	-100	100	Dis.	Dis.	
Joint 3	Disabled		-165	165	-165	165	-165	165	
Joint 2	Disabled		-160	160	-160	160	-160	160	
Joint 1	Disabled		Dis.	Dis.	Dis.	Dis.	-179.9	179.9	

Export < 1 2 3 4 5 6 > Edit Start Verification

Figure 4.14.2.2 : Exemples de réglages pour les angles d'articulation limités en toute sécurité

4.14.2.3 Vitesse limitée en toute sécurité (SLS)

Le troisième sous-menu des paramètres de sécurité concerne la vitesse limitée en toute sécurité. Il permet de limiter individuellement la vitesse de chaque axe du robot, ainsi que la vitesse de déplacement de toutes les parties du robot, y compris l'outil, la base et les bras.

Safety Settings

Safely Limited Speed (SLS)

Defining the speed limits of each joint and all movable parts of the robot.

Note that a safety margin of **7 deg/s** or **100 mm/s** is subtracted from the specified speed limits to determine the effective limits.

	Default	Safe Case 1	Safe Case 2	Safe Case 3	
Joint Velocity 6	247	247	247	247	deg/s
Joint Velocity 5	247	247	247	247	deg/s
Joint Velocity 4	247	247	247	247	deg/s
Joint Velocity 3	187	187	187	187	deg/s
Joint Velocity 2	187	187	187	187	deg/s
Joint Velocity 1	187	187	187	187	deg/s
Speed	2000	2000	2000	2000	mm/s

Export < 1 2 3 4 5 6 > Edit Start Verification

Figure 4.14.2.3 : Exemples de réglages pour la SLS

4.14.2.4 Paramètres de sécurité pour la puissance et la force

Le quatrième sous-menu des paramètres de sécurité contient les réglages relatifs à la puissance et à la force :

- Pour chaque scénario de sécurité, l'utilisateur peut définir une limite de force différente, qui détermine la sensibilité du robot aux forces externes telles que les collisions avec des personnes ou des objets. La limite de force s'applique au point central de l'outil (TCP).
- La définition de la charge utile moyenne (et des valeurs de charge utile correctes pendant le fonctionnement), définie comme le poids de l'outil plus les objets transportés, est importante pour adapter les limites de force de manière plus précise à votre application spécifique.
- Le vecteur de gravité du robot doit correspondre à son mode de montage. Si le robot est monté au sol, le vecteur de gravité est $[0, 0, -9,81]$. En revanche, si le robot est monté sur un mur, sur une surface inclinée ou au plafond, le vecteur de gravité doit d'abord être calculé, puis saisi correctement ici.

Safety Settings

Safe Power and Force Limit (PFL)

Defining the maximum allowable force at the tool center point (TCP). If this limit is exceeded, a collision interrupt is triggered. Raising the limit reduces collision detection sensitivity. When payload variations are large, the force limit may need to be increased to account for calculation differences.

	Default	Safe Case 1	Safe Case 2	Safe Case 3	
Force Limit	250	250	250	250	N

Average Payload

Defining the robot's average payload — the combined weight of its tool and any picked items. During operation, the payload changes as the robot picks up and places items. The safety system uses the average payload to assess whether the applied forces remain within acceptable limits.

Average Payload: 0 kg

Gravity Vector

Defining the gravity vector in the robot's base frame based on its mounting orientation.

	x	y	z	
Gravity Vector	0	0	-9.81	m/s ²

Export < 1 2 3 4 5 6 > Edit Start Verification

Figure 4.14.2.4 : Réglages sûrs pour les limites de puissance et de force

4.14.2.5 E/S de sécurité et restrictions d'accès

Le cinquième sous-menu du menu des paramètres de sécurité permet de configurer le comportement des E/S de sécurité et les autorisations des opérateurs. Les utilisateurs peuvent définir la manière dont les arrêts de sécurité sont gérés, si les entrées sont surveillées en mode manuel, les conditions de réinitialisation, les autorisations pour le changement de mode de fonctionnement ainsi que les exigences en matière de mot de passe pour les modifications du mode de fonctionnement.

The screenshot shows a 'Safety Settings' window with a blue header bar containing an information icon, a maximize icon, and a close icon. The window is divided into two main sections: 'Safe I/O Configuration' and 'Operator Limitations'.

Safe I/O Configuration

Defining whether a protective stop input results in a SS1 or SS2 stop.

Results in a SS2 Stop ☐ Results in a SS1 Stop ☐

Defining whether a protective stop input is always being normally observed, or if it is to be ignored during manual mode.

Ignore in manual mode ☐ Observe in manual mode ☐

Defining whether resetting interrupts can only be triggered via digital inputs.

Trigger always ☐ Digital inputs only ☐

Operator Limitations

Defining whether operators are able to switch the operation mode.

Allow switching mode ☐ Don't allow switching mode ☐

Defining whether operators need to enter their password to switch the operation mode.

No password required ☐ Password required ☐

At the bottom of the window, there is a navigation bar with an 'Export' button, a set of numbered tabs (1, 2, 3, 4, 5, 6) where tab 5 is selected, and 'Edit' and 'Start Verification' buttons.

Figure 4.14.2.5 : Paramètres de configuration pour les E/S de sécurité

4.14.2.6 Données de vérification

Le dernier sous-menu des paramètres de sécurité contient des informations sur la configuration de sécurité :

- Les sommes de contrôle lues à partir de la configuration de sécurité (à des fins de dépannage)
- Le compteur de transferts (indique le nombre de fois où les paramètres de sécurité ont été écrits)
- La date de la dernière mise à jour des paramètres de sécurité
- Si les paramètres de sécurité ont été vérifiés
- Le numéro de série et la version du micrologiciel du dispositif de sécurité

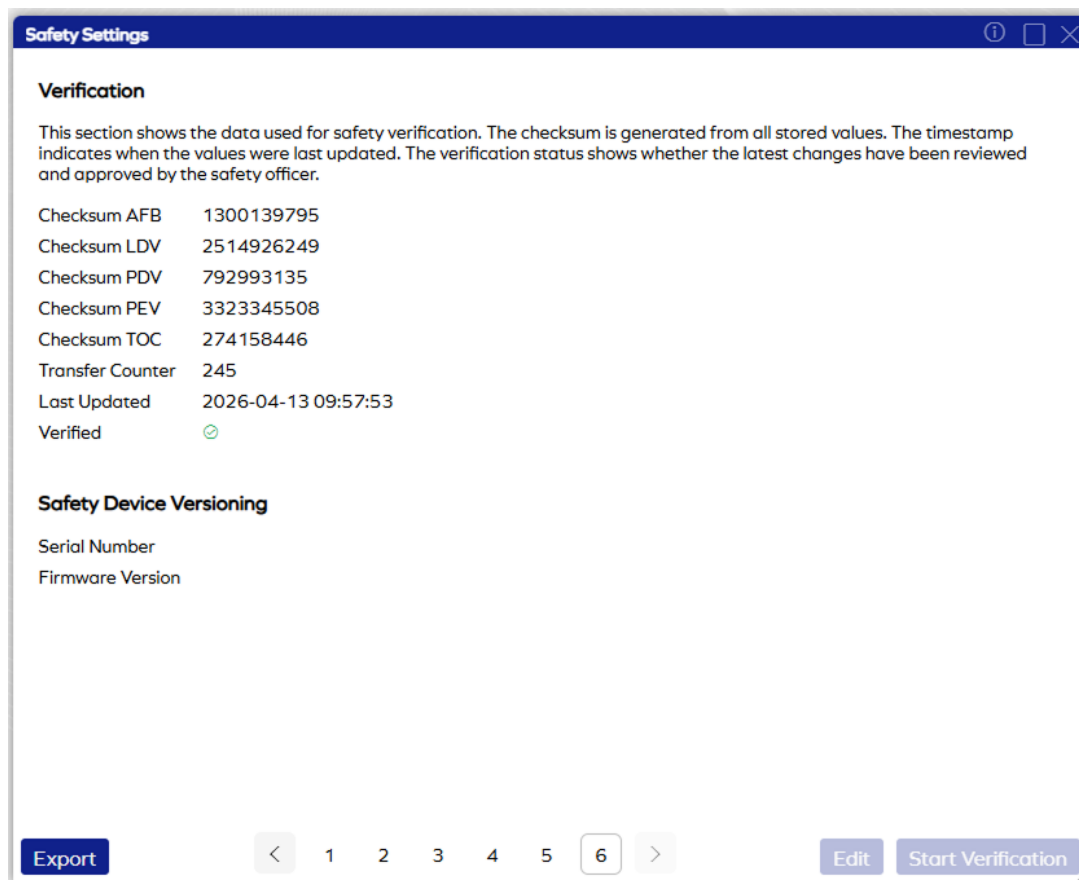


Figure 4.14.2.6 : Paramètres de vérification

4.14.2.7 Fonctions de sortie de sécurité

Les états logiques de certaines fonctions de sécurité sont reliés aux sorties de sécurité du contrôleur du robot. Vous trouverez des informations détaillées dans le « Contrôleur de Robot RC-A101 - Manuel ».

4.14.2.8 Raccordement d'appareils optionnels

Pour le raccordement d'appareils optionnels tels qu'un support pour tablette ou un interrupteur d'assentiment, veuillez consulter le « Contrôleur de Robot RC-A101 - Manuel ».

5. Interface panneau (GUI)

5.1 Introduction

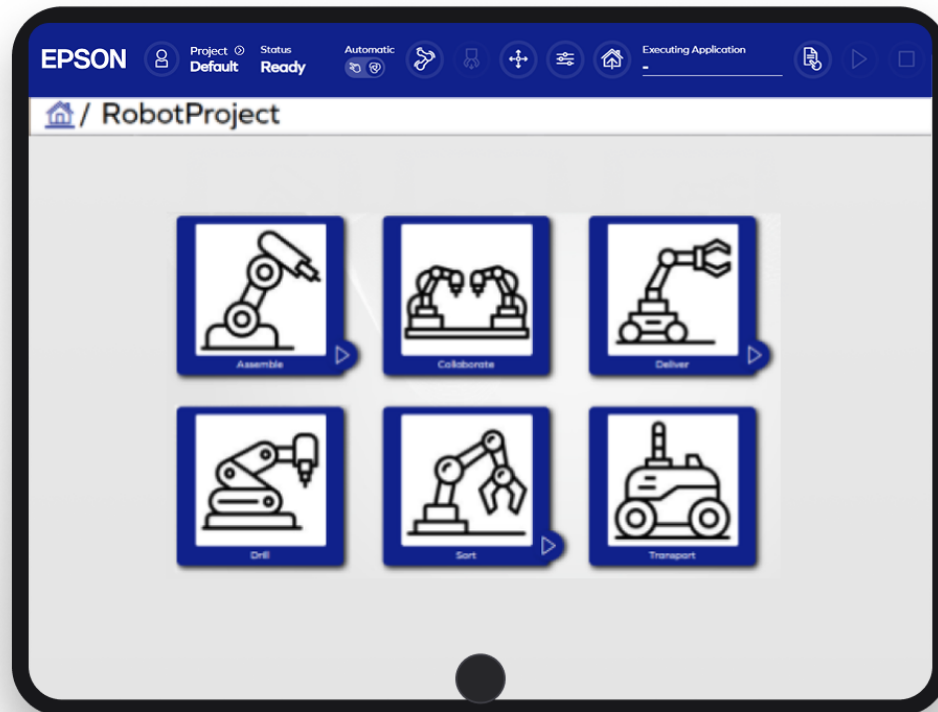


Figure 5.1.a : Exemple de capture d'écran de l'interface du panneau ("Panel Interface")

L'interface du panneau ("Panel Interface") constitue une alternative à l'interface de bureau pour un accès rapide à vos applications principales. Cette interface a été conçue pour être entièrement personnalisable en fonction des besoins des clients et, en particulier, pour afficher exclusivement les applications (et les informations relatives à leur état). L'interface du panneau ("Panel Interface") est compatible avec les écrans tactiles et sa conception s'apparente à la structure de type application bien connue des appareils mobiles modernes, ce qui la rend intuitive même pour les utilisateurs sans connaissances techniques. Avec l'interface du panneau ("Panel Interface"), l'utilisateur n'a en outre pas à se soucier de la programmation ou du code sous-jacent de l'application. Il suffit de cliquer sur l'application correspondante pour que celle-ci se lance. Cette interface est accessible depuis l'interface d'administration : connectez-vous et cliquez sur le bouton « Aller à la Panel Interface ».

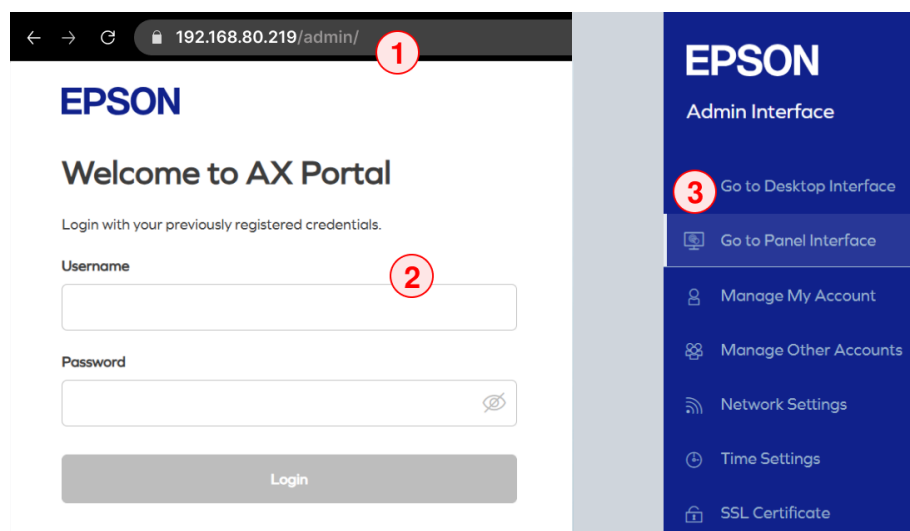


Figure 5.1.b : Accès à l'interface du panneau ("Panel Interface")

Dans certains navigateurs (notamment sous Windows), le zoom d'affichage est réglé par défaut sur une valeur élevée (par exemple 150 %), ce qui peut entraîner l'apparition de parties de l'interface (telles que le cockpit ou le logo) coupées. Pour un affichage optimal, nous vous recommandons de régler le zoom d'affichage sur 100 % (ou moins).

5.2 Configurer les applications pour l'interface du panneau ("Panel Interface")

Par défaut, l'interface du panneau ("Panel Interface") n'affiche aucune application. L'utilisateur doit d'abord définir quelle application doit y être affichée pour une exécution rapide par l'opérateur. Pour configurer une application du panneau, procédez comme suit :

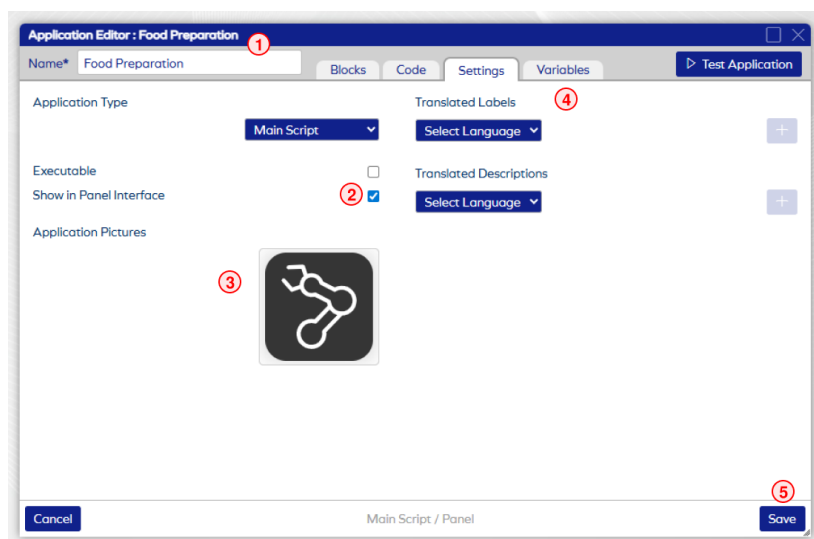


Figure 5.2.a : Paramètres pertinents pour l'interface du panneau ("Panel Interface") dans l'éditeur d'applications (interface de bureau)

Paramètres pertinents pour l'interface du panneau ("Panel Interface") : 2., 3., 4.

1. Accédez à l'interface de bureau, modifiez l'application et ouvrez les paramètres de l'application,
2. activez l'option « **Afficher dans l'interface du panneau (“Panel Interface”)** »,
3. téléchargez éventuellement une **image d'application** appropriée (le format optimal est 5:4), qui s'affichera ultérieurement avec le nom de l'application dans le menu principal de l'interface du panneau (“Panel Interface”) ,
4. accédez éventuellement à l'onglet **Variables d'application** pour ajouter des variables d'état qui vous permettront de surveiller et/ou de modifier des variables via l'interface du panneau (“Panel Interface”) ,
5. et enfin, **enregistrez** l'application.

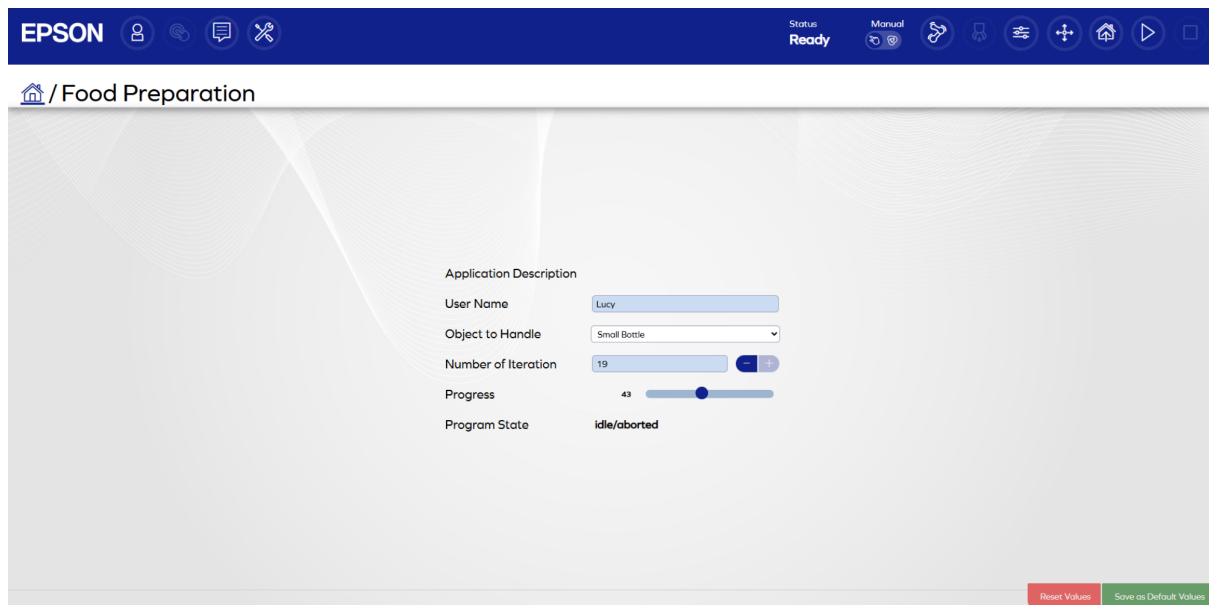


Figure 5.2.b : Application de panneau avec variables d'état

5.3 Cockpit et paramètres

Le Cockpit de l'interface du panneau ("Panel Interface") affiche en grande partie les mêmes boutons que le Cockpit de l'interface du bureau. Vous trouverez une explication détaillée de tous les éléments dans la section Cockpit. En plus de ces éléments, vous verrez sur le côté gauche quelques boutons différents :



Le bouton « **Panel** » ouvre la page principale du panneau de commande, sur laquelle s'affichent les applications à sélectionner et à exécuter.



Le bouton « **Messages** » ouvre l'historique des messages, dans lequel tous les messages précédents, y compris les avertissements et les erreurs, sont affichés par ordre chronologique.



Le bouton « **Paramètres** » vous permet d'accéder à la fenêtre des paramètres, qui ne contient que les réglages les plus importants pour un accès rapide, à savoir

- la sélection de l'outil connecté,
- la mise en service et la mise hors service du robot,
- le téléchargement des fichiers journaux.

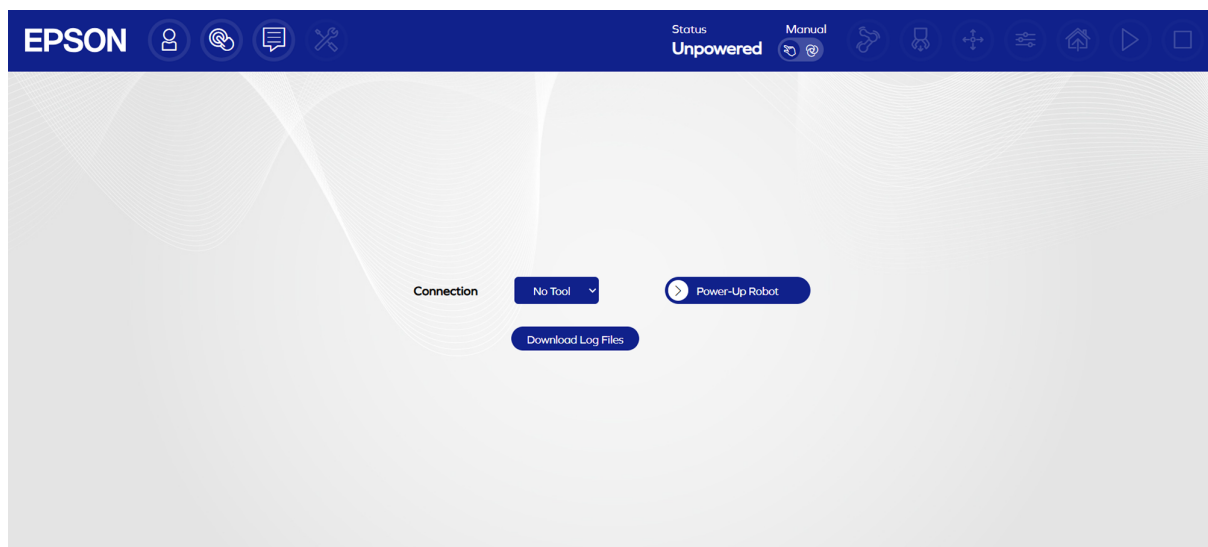


Figure 5.3 : Paramètres de l'interface du panneau ("Panel Interface")

5.4 Contrôler les applications

Dans la fenêtre principale de l'interface du panneau ("Panel Interface"), toutes les applications sont répertoriées dans une structure de type « app », avec affichage de leur nom et de leur icône. Le panneau distingue deux types d'applications :

- Les applications **sans variables d'état** affichent un bouton « Play ». Appuyez sur l'application pour l'exécuter directement.
- Les applications **avec variables d'état** s'affichent sans bouton « Play ». Appuyez sur ces applications pour ouvrir la vue détaillée contenant toutes les variables d'état. Dans cette fenêtre, vous pouvez modifier les variables d'état modifiables et appuyer sur le bouton « Exécuter » pour lancer ces applications. Appuyez sur le bouton « Retour » pour revenir à la fenêtre principale du panneau.

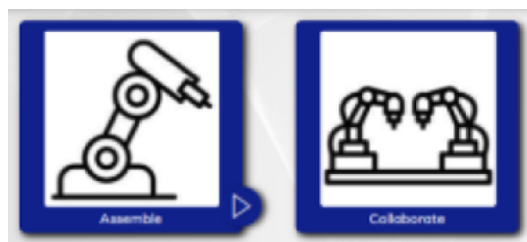


Figure 5.4 : Boutons de sélection des applications

6. Définitions cinématiques

La cinématique du robot décrit la manière dont le robot se déplace et comment ses mouvements peuvent être représentés. Pour travailler avec la cinématique, l'utilisateur doit comprendre les différentes définitions des paramètres cinématiques et des objets disponibles :

6.1 Coordonnées

Les coordonnées définissent la position et l'orientation absolues du robot. La position est un vecteur à trois valeurs $[x, y, z]$ en mm, et l'orientation est également un vecteur à trois valeurs $[\text{roll}, \text{pitch}, \text{yaw}]$ (angles nautiques) en degrés.

L'initialisation peut s'effectuer comme suit :

```
coord = Coordinates([600, 0, 300, 180, -90, 0])
# ... this is similar to ...
coord = Coordinates({"x": 600, "y": 0, "z": 300, "roll": 180, "pitch": -90, "yaw": 0})
```

L'accès aux éléments peut s'effectuer comme suit :

```
coord = Coordinates([600, 0, 300, 180, -90, 0])

# accessing individual elements (x, y, z, roll, pitch, yaw)
x = coord.x
x = coord["x"]
x = coord[0]

# accessing position or orientation
position = coord.position
orientation = coord.orientation
```

6.1.1 Angles nautiques

Les angles nautiques constituent l'une des nombreuses possibilités permettant de définir la composante d'orientation (roulis, tangage, lacet) des coordonnées. Les angles nautiques sont généralement utilisés pour l'orientation des aéronefs. Dans le cas des angles nautiques, une rotation tridimensionnelle est définie comme suit

1. une rotation autour de l'axe z négatif dans le sens du lacet (yaw, ψ),
2. une rotation autour de l'axe y négatif dans la direction du tangage (pitch, θ) et
3. une rotation autour de l'axe x positif dans le sens du roulis (ϕ).

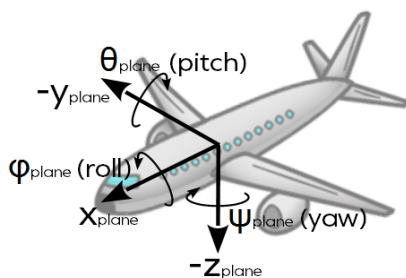


Figure 6.1.1.a : Système de référence nautique

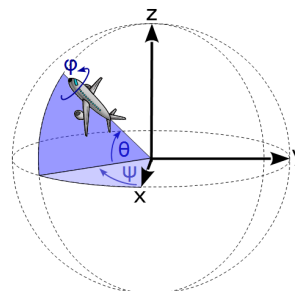


Figure 6.1.1.b : Ordre des rotations

Remarque : dès qu'une fonction vous oblige à travailler avec des orientations tridimensionnelles de l'effecteur terminal, il est utile d'imaginer un plan fixé à l'effecteur terminal, de sorte que les systèmes de référence de l'effecteur terminal et de l'avion coïncident. Cela permet de déduire beaucoup plus facilement les angles (roulis, tangage et lacet) qui conduisent précisément à cette orientation.

6.1.2 Systèmes de référence

Pour la manipulation des robots de la série AX, les systèmes de référence suivants sont importants :

- Le **système de base** d'un robot est défini de telle sorte que son axe z soit orienté à l'opposé de la plaque de base et que l'axe x soit parallèle à la plaque de base et orienté à l'opposé des raccords situés sur la base. La règle bien connue de la main droite définit alors la direction de l'axe y.
- Le **système de travail** d'un robot est un système de référence défini par l'utilisateur par rapport au système de référence de base, et correspond à celui-ci sauf indication contraire. Les coordonnées sont indiquées dans ce système de référence. L'utilisateur peut utiliser ce système de référence pour définir son propre espace de travail pour le robot.
- Le **système de bride** d'un robot varie en fonction du type d'outil :
 - o Si aucun outil n'est monté, l'origine de ce système de référence se trouve au centre de la bride (plaque de raccordement) sur la face supérieure du robot. L'orientation correspond à celle du système de base lorsque tous les angles d'articulation sont à zéro.
 - o Lorsqu'un outil personnalisé est monté (et défini via l'éditeur d'outils), le système de référence est décalé de la position et de l'orientation spécifiées pour cet outil.
- Le **système d'outils** d'un robot est un système de référence défini par l'utilisateur par rapport au système de référence de la bride et correspond à celui-ci, sauf indication contraire. Les coordonnées sont obtenues à l'origine de ce système de référence. L'utilisateur peut utiliser ce système de référence pour définir le point central de l'outil.

Vous trouverez un exemple de ces systèmes de référence dans l'illustration suivante :

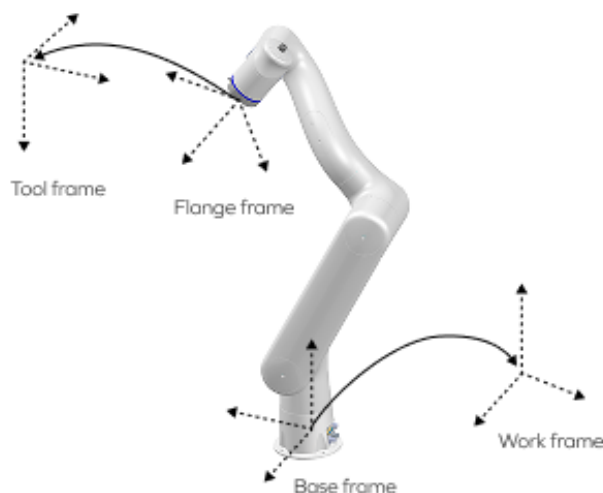


Figure 6.1.2 : Systèmes de référence du robot

6.2 Configuration

Pour une position et une orientation données du robot, la cinématique inverse peut fournir plusieurs configurations d'articulations différentes. Ces configurations se distinguent par la manière dont les articulations sont inclinées ou pivotées. Le paramètre de configuration permet de sélectionner une configuration spécifique parmi ces options possibles. Sur un robot à six axes typique, différentes combinaisons des orientations de l'épaule, du coude et du poignet peuvent donner lieu à jusqu'à huit configurations différentes, même si la position et l'orientation de l'effecteur terminal sont identiques. Pour les distinguer, ces configurations sont désignées par « c1 » à « c8 ».

Veuillez noter que les configurations ne sont pas adaptées pour comparer deux poses ou vérifier si un chemin est continu. Un changement de configuration ne rend pas nécessairement un chemin discontinu. De même, un chemin discontinu ne contient pas nécessairement de changement de configuration.

6.3 Pose

Une pose définit les coordonnées et la configuration du robot. Elle peut être définie par l'ID ou le nom d'une pose prédéfinie dans la base de données. Elle peut également être créée en spécifiant les coordonnées et la configuration à l'aide de la fonction « Créer une nouvelle pose » ou en spécifiant les angles articulaires à l'aide de la fonction de cinématique directe.

L'initialisation peut s'effectuer comme suit :

```
# read predefined pose from database by index
pose = Pose(1)
# read predefined pose from database by name
pose = Pose("fancy_pose")
```

6.4 Point

Un point est un élément discret d'un chemin. Outre la pose du robot, il indique également la vitesse du robot ainsi que l'intervalle de temps écoulé depuis le point précédent. Ces informations sont désignées par l'acronyme PVT (Position, Velocity, Time) et sont généralement stockées sous la forme d'un vecteur d'angles d'articulation, d'un vecteur de vitesses d'articulation et d'un intervalle de temps (nombre à virgule flottante). Veuillez noter qu'une version étendue d'un point peut également contenir l'accélération du robot.

6.5 Trajectoire

Une trajectoire définit la manière dont le robot se déplace entre deux ou plusieurs positions. Une trajectoire continue est définie exclusivement dans l'espace des outils par plusieurs segments, qui peuvent être linéaires ou circulaires. La trajectoire est ensuite discrétisée et transposée dans l'espace des articulations. Une trajectoire est dite sans discontinuité lorsque les solutions discrètes de la trajectoire ne nécessitent pas de repositionnement du robot (changement de configuration). Une trajectoire est dite continue lorsque l'accélération le long de celle-ci est constante.

Une trajectoire commence à une pose spécifiée (par opposition à des coordonnées). Les segments de la trajectoire définissent généralement des coordonnées, car la configuration de la trajectoire est

déterminée par la pose de départ. En utilisant la solution la plus proche de la cinématique inverse par rapport au point discret précédent, la trajectoire devient automatiquement sans discontinuité dans la mesure du possible. Veuillez noter que nous utilisons les termes « chemin » et « trajectoire » comme synonymes, même si, d'un point de vue technique, un chemin ne contient pas d'informations sur la vitesse.

L'initialisation peut s'effectuer comme suit :

```
# read predefined path from database by name
path = Path("fancy_path")
```

6.6 Segment

Un segment représente une partie distincte d'un chemin qui définit la manière dont le TCP se déplace vers la destination.

Les types de segments disponibles sont les suivants :

- **Linéaire** : le TCP se déplace en ligne droite vers les coordonnées de destination définies par le segment.
- **Circulaire** : le TCP suit une trajectoire circulaire depuis sa position actuelle, en passant par une position intermédiaire définie, jusqu'aux coordonnées cibles du segment. Veuillez noter que l'orientation du TCP est automatiquement déterminée par l'algorithme pendant ce mouvement.
- **Coordonnées** : le TCP se déplace selon le chemin le plus efficace vers les coordonnées cibles. Cela correspond à un mouvement linéaire dans l'espace articulaire.
- **Position** : le TCP se déplace selon le chemin le plus efficace vers la position cible. Cela correspond à un mouvement linéaire dans l'espace articulaire.
- **Orientation** : le TCP effectue un mouvement de rotation pur pour atteindre l'orientation cible spécifiée, tout en conservant sa position actuelle.

Pour chaque segment, des paramètres de vitesse et d'accélération peuvent être définis :

- Pour les segments d'orientation, ceux-ci correspondent à la vitesse angulaire en [deg/s] et à l'accélération angulaire en [deg/s²]
- Pour tous les autres types de segments, ils correspondent à la vitesse linéaire en [mm/s] et à l'accélération linéaire en [mm/s²]

De plus, il est possible de définir un rayon de transition (Blend Radius) qui détermine l'écart maximal autorisé en mm par rapport à la pose finale du segment précédent – qui correspond à la pose de départ du segment actuel – pendant la transition. Un rayon de transition de 0 mm oblige le TCP à passer exactement par la pose définie, tandis qu'une valeur plus élevée permet des transitions plus douces et plus naturelles en autorisant une courbure de la trajectoire entre les segments.

6.7 Grilles

Les grilles sont des tableaux de points personnalisables, pouvant aller jusqu'à trois dimensions, qui définissent par exemple les positions de plateaux prédéfinis et peuvent être utilisées pour regrouper et

enregistrer des ensembles de valeurs de coordonnées. En règle générale, une grille est interprétée comme un tableau bidimensionnel ; elle peut toutefois également être une simple liste de points sans deuxième dimension (par exemple pour prélever des objets sur un convoyeur) ou même une structure tridimensionnelle (par exemple pour empiler des objets dans des caisses).

7. Modules de connexion

Outre les interfaces utilisateur conviviales et intuitives permettant de commander et de surveiller le robot, AX Portal comprend également une série de modules de connexion grâce auxquels le robot peut être commandé et surveillé via différentes méthodes de communication telles que TCP, REST, ROS et Modbus. Chacune de ces méthodes de communication est adaptée à différents cas d'utilisation :

- **TCP** est idéal pour la communication en temps réel, qui nécessite un échange de données rapide et continu. Il est souvent utilisé pour des applications de commande directe, des implémentations client spécifiques ou pour l'intégration dans des systèmes nécessitant des connexions socket persistantes.
- **REST** est particulièrement adapté à l'intégration dans des services ou des applications web. Il est généralement utilisé lorsque la convivialité, l'évolutivité et la compatibilité avec des systèmes standardisés basés sur HTTP priment sur les performances en temps réel.
- **ROS** (Robot Operating System) est conçu pour la recherche et le développement dans le domaine de la robotique. Il est particulièrement utile pour l'intégration du robot dans des systèmes robotiques complexes et permet des fonctionnalités telles que la fusion de capteurs, le comportement autonome et l'interaction avec d'autres appareils compatibles ROS.
- **Modbus** est fréquemment utilisé dans les environnements d'automatisation industrielle. Il est particulièrement adapté à la communication avec les automates programmables (API) et d'autres appareils industriels pour lesquels la fiabilité et la normalisation sont essentielles.

L'utilisation de ces modules est expliquée dans ce chapitre.

7.1 Connexions, authentification et contrôle actif

Avant d'utiliser un module de connexion, il est important de comprendre comment ces modules se connectent aux clients, comment fonctionne l'authentification et comment s'applique le principe de la commande active dans AX Portal.

7.1.1 Canaux de connexion

La connexion entre un appareil et le robot s'effectue toujours via un canal de connexion spécifique. Ces canaux de connexion sont implémentés de manière légèrement différente pour les différents modules de connexion.

Le **module** TCP prend en charge **plusieurs canaux**. Cela signifie que chaque appareil souhaitant se connecter au robot crée son propre canal de connexion et s'authentifie séparément. La fermeture de l'un de ces canaux met fin à la connexion et libère également le contrôle actif (voir ci-dessous).

 AVERTISSEMENT	Les modules REST, ROS et Modbus ne disposent chacun que d'un seul canal. Cela signifie que dès que la connexion à l'un de ces modules est établie et authentifiée, n'importe quel appareil peut envoyer des commandes au robot. Ces canaux sont créés au démarrage d'AX Portal et ne peuvent pas être fermés par la connexion d'appareils.
---	---

7.1.2 Authentification

AX Portal utilise une authentification sécurisée en deux étapes. Commencez par suivre la procédure de connexion à distance décrite dans la section Authentification sécurisée afin d'obtenir un jeton d'accès valide. Cette étape doit toujours être effectuée via une **connexion TCP distincte**. Une fois que vous vous êtes connecté avec succès et que vous avez généré un jeton d'accès, vous pouvez enregistrer ce jeton via le canal de connexion choisi à l'aide de la commande **token_login** afin d'authentifier votre utilisateur.

 ATTENTION	L'étape d'authentification peut être ignorée si vous désactivez l'authentification via TCP dans le menu des paramètres. Veuillez noter que le module Modbus ne permet pas d'envoyer une commande <code>token_login</code> ; l'authentification de ce module doit donc être désactivée dans le menu des paramètres pour pouvoir l'utiliser.
---	--

7.1.3 Commande active

AX Portal dispose d'une protection de la station de contrôle active, afin qu'aucun autre utilisateur ou connexion ne puisse contrôler le robot simultanément. Sans cette protection, des situations potentiellement dangereuses pourraient survenir.

La commande **request_active_control** est utilisée par un canal de connexion pour devenir la station de commande active. Dès que cette commande a été exécutée avec succès, cette station de commande dispose du contrôle permanent jusqu'à ce que le canal soit fermé ou que le robot soit mis hors tension. Si une autre station de commande détient le contrôle, cette commande est bloquée jusqu'à ce que cette station de commande accepte la demande de contrôle.

La commande **release_active_control** est utilisée par la station de contrôle active pour céder le contrôle actif. Il est recommandé de toujours libérer le contrôle dès que le canal n'est plus utilisé activement. Sinon, ce canal bloque le contrôle actif de manière permanente.

7.2 Module de connexion TCP

7.2.1 Établir une connexion

Le côté serveur du module de connexion TCP est basé sur [Python](#) et a été implémenté à l'aide de la [bibliothèque de sockets](#) intégrée . Par défaut, le serveur est accessible via le **port 38152**. Le message est encodé selon la norme **UTF-8**. Les messages sont séparés par le caractère de **fin de texte** 0x03 (en Python « \x03 »). Utilisez le script d'exemple suivant pour vous connecter à votre robot et afficher toutes les réponses entrantes. La signification des messages entrants est expliquée plus bas dans la section Réception des commandes.

```
import socket
import time

# define the robot ip address
ip_address = "10.0.0.203"

# create the tcp socket
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as client_socket:
    # connect to the server
    client_socket.connect((ip_address, 38152))
    # prepare message buffer
    message_buffer = ""
    while True:
        # read data and add it to message buffer
        try:
            message_buffer += client_socket.recv(1024).decode("UTF-8")
        except socket.timeout:
            time.sleep(0.1)
            continue
        # check for complete messages in buffer
        if "\x03" in message_buffer:
            # extract complete messages
            message_split = message_buffer.split("\x03")
            for message in message_split[:-1]:
                print(message)
            # incomplete message (if any) remains in buffer
            message_buffer = message_split[-1]
```

7.2.2 Envoi de commandes

Avant d'envoyer des commandes au robot, assurez-vous d'avoir bien compris les principes expliqués dans la section Connexions, authentification et contrôle actif. L'exemple suivant envoie les commandes initiales **token_login** et **request_active_control**.

```
import json
import socket
import time

# define the robot ip address, access token, and session identifier
ip_address = "10.0.0.203"
token = "MY_ACCESS_TOKEN"
session_id = "MY_SESSION_ID"

# create the tcp socket
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as client_socket:
    # connect to the server
    client_socket.connect((ip_address, 38152))

    # use the authentication token to log in (needed if authentication is not disabled)
    request = {"action": "token_login", "bridge": "core",
               "arguments": {"token": token, "session_id": session_id}}
    message = json.dumps(request) + "\x03"
    client_socket.sendall(message.encode("UTF-8"))

    time.sleep(0.5)

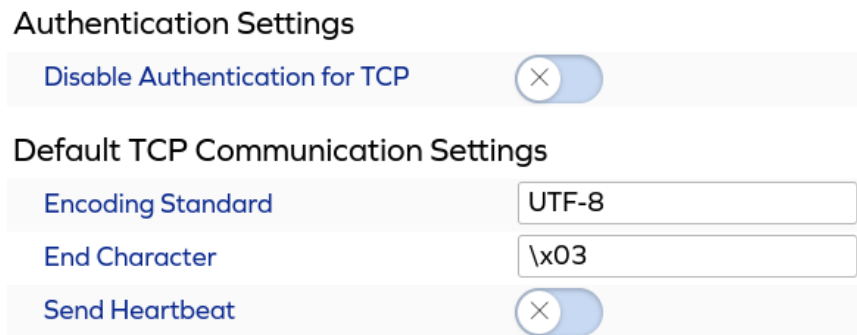
    # make sure to acquire active control, which makes the TCP connection the active
    # control station of the robot (only one control station can be active at any time)
    request = {"action": "request_active_control", "bridge": "core",
               "arguments": {}}
    message = json.dumps(request) + "\x03"
    client_socket.sendall(message.encode("UTF-8"))

    time.sleep(0.5)

# add more robot control commands here
```

7.2.3 Paramètres

Les exemples ci-dessus utilisent les paramètres TCP par défaut suivants, qui peuvent être modifiés dans le menu des paramètres de l'interface de bureau. Vous pouvez y désactiver définitivement l'authentification pour TCP, ainsi que modifier le codage et le caractère de fin des messages TCP. Vous pouvez également activer un signal de pulsation afin de vérifier en permanence si la connexion est toujours établie.



Authentication Settings

Disable Authentication for TCP ☐

Default TCP Communication Settings

Encoding Standard UTF-8

End Character \x03

Send Heartbeat ☐

Figure 7.2.3 : Paramètres du module TCP

7.2.3.1 Norme de codage

Vous pouvez ici définir le codage des messages envoyés via la connexion TCP. Les [normes de codage Python](#) habituelles sont acceptées. Les normes couramment utilisées sont UTF-8 (valeur par défaut) ou ASCII.

7.2.3.2 Caractère de fin

Vous pouvez ici modifier le caractère de fin des messages envoyés via la connexion TCP. Le caractère de fin signale la fin du message. À la réception d'un message, le caractère de fin est automatiquement supprimé du message.

7.2.3.3 Envoyer un heartbeat

Vous pouvez ici définir si le serveur TCP envoie automatiquement un heartbeat. Un heartbeat est envoyé toutes les 50 ms et consiste en un message vide (le caractère de fin est envoyé).

7.2.4 Journal des messages – Réception des commandes

Toutes les commandes envoyées et reçues sont des chaînes [JSON](#). Comme indiqué ci-dessus, les messages sont encodés selon la norme **UTF-8** et séparés par le caractère de **fin de texte** 0x03 (en Python « \x03 »). Une commande reçue est un dictionnaire codé en JSON contenant une clé (identifiant) et une valeur. La clé définit le type de message. La valeur contient les informations du message. Vous trouverez un exemple de code recevant ces messages du robot dans la section Établir la connexion.

Veuillez noter que cette liste n'est pas exhaustive, mais qu'elle contient les commandes les plus importantes.

7.2.4.1 Informations d'état

L'état du robot est envoyé dès qu'il change.

Clé	Valeur	Type
status_info	L'état actuel du robot selon la liste suivante : ● 0 → Hors service ● 1 → Connexion ● 2 → En cours de chargement ● 3 → Prêt ● 4 → En cours ● 5 → Calibrage ● 6 → En cours ● 7 → Mode autonome ● 8 → En pause ● 9 → Reprise ● 10 → Arrêt de sécurité ● 11 → Arrêt de sécurité ● 12 → Limitation de position ● 13 → Commande manuelle ● 14 → Collision ● 15 → Arrêt d'urgence ● 16 → Arrêt normal ● 17 → Erreur ● 18 → Déconnexion ● 19 → Arrêté	Entier

```
{"status_info": 0}
```

7.2.4.2 Valeur de retour de l'action

Une fois l'action exécutée, la valeur de retour est envoyée avec la commande de l'action. Si l'action n'a pas de valeur de retour, celle-ci est envoyée comme étant nulle. L'action n'est réussie que si aucune erreur ne s'est produite lors de l'exécution de la fonction. Enfin, le numéro de séquence est un paramètre facultatif qui n'est ajouté que si l'utilisateur a spécifié une séquence lors de l'envoi de la commande.

Key	Valeur	Type
action_return_value	Le nom de l'action renvoyée, la valeur de retour et si l'action a abouti. En cas d'erreur, l'action n'a pas abouti.	Dictionnaire

```
{"action_return_value": {"action": "release", "value": null,
                        "sequence": 100, "success": true}}
```

7.2.4.3 Erreurs

En cas d'erreur, un message d'erreur contenant des informations détaillées est envoyé.

Clé	Valeur	Type
error	Les informations relatives à l'erreur.	Dictionnaire

```
{"error": {"text": "The argument 'variable_value' is missing for the  
              'set_state_variable' action.",  
          "title": "Minor Error",  
          "color": 3,  
          "code": "e0010035",  
          "time": 1554907874.0436678}}
```

7.2.4.4 Historique des messages

Liste des messages précédents envoyés à l'interface utilisateur.

Clé	Valeur	Type
message_history	Les messages contiennent du texte, un titre, un code et un horodatage.	Liste

```
{"message_history": [{"text": "The specified bridge does not exist.",  
                          "title": "Minor Error", "color": 3,  
                          "code": "e0010019",  
                          "time": 1554897027.5446029}]}
```

7.2.4.5 Définition d'un groupe d'états

Définit les données de position du robot. Un groupe d'états contient un identifiant qui précise à quoi se rapportent les données qu'il contient. Les données peuvent être identifiées à l'aide de ces définitions.

Clé	Valeur	Type
state_group_define	Définition du groupe d'états.	Dictionnaire

```
{"state_group_define": {"id": "pose",  
                       "label": "tcp_pose",  
                       "members": [{"id": "x", "label": "x"},  
                                   {"id": "y", "label": "y"},  
                                   {"id": "z", "label": "z"},  
                                   {"id": "roll", "label": "roll"},  
                                   {"id": "pitch", "label": "pitch"},  
                                   {"id": "yaw", "label": "yaw"}],  
                       "cols": [{"id": "value", "unit": "mm, deg"}]}}
```

7.2.4.6 Données d'un groupe d'état

Les données d'un groupe d'état contiennent des données TCP ou des données moteur. Utilisez « request_data » pour recevoir ces données.

Clé	Valeur	Type
state_group_data	La valeur contient soit le TCP du robot (id = « pose »), soit les données moteur (id = « robot »), soit les états de sécurité (id = « safety »).	Dictionnaire

```
{
  "state_group_data": {
    "id": "pose",
    "data": {
      "x": { "value": "0.0" },
      "y": { "value": "0.0" },
      "z": { "value": "1378.5" },
      "roll": { "value": "0.0" },
      "pitch": { "value": "90.0" },
      "yaw": { "value": "180.0" }
    }
  }
}
```

7.2.4.7 Ouvrir une boîte de dialogue

Lorsqu'une boîte de dialogue s'ouvre dans l'interface utilisateur, la commande « open_dialog » est reçue. Il est ensuite possible de répondre à la boîte de dialogue à l'aide de l'action « answer_dialog ».

Clé	Valeur	Type
open_dialog	La boîte de dialogue prend les valeurs suivantes : • id → identifiant unique de la boîte de dialogue (pour y répondre) • kind ○ 0 → boîte de dialogue de confirmation (boutons pour confirmer/refuser) ○ 1 → boîte de dialogue de saisie (saisie de texte) ○ 2 → boîte de dialogue de fichier (téléchargement de fichier) ○ 3 → Boîte de dialogue déroulante (nécessite une liste d'options) ○ 4 → Boîte de dialogue de sélection (nécessite une liste d'options) • default → Valeur de retour si aucune réponse n'est reçue • couleur ○ 0 → Information ○ 1 → Succès ○ 2 → Avertissement ○ 3 → Erreur • title → titre à afficher • text → texte à afficher • options → Liste d'options pour les menus déroulants/boîtes de dialogue de sélection	Dictionnaire

```
{
  "open_dialog": {
    "id": "5cee85a8-9ab2-4de8-b1ea-52ded8cc68c0",
    "kind": 0, "default": true, "color": 2,
  }
}
```

```
"title": "Movement Warning",
"text": "In order to Calibrate, the robot will move.
        Do you want to proceed?"}}
```

7.2.4.8 Envoi depuis une application

Reçoit un message envoyé par une application.

Clé	Valeur	Type
script_send	Valeur envoyée par l'application	Divers

```
{"script_send": 123}
```

7.2.5 Journal des messages – Envoi de commandes

Toutes les commandes envoyées et reçues sont des chaînes [JSON](#) . Comme indiqué ci-dessus, les messages sont encodés selon la norme **UTF-8** et séparés par le caractère de **fin de ligne** 0x03 (en Python « \x03 »). Une commande envoyée doit contenir un dictionnaire encodé en JSON comportant trois clés obligatoires (et une clé facultative). Vous trouverez un exemple de code permettant d'envoyer des messages au robot dans la section Envoyer des commandes.

action (fonction)	définit la commande à exécuter	str
bridge (module)	définit le module contenant la commande, généralement « core »	str
arguments (arguments)	définissent les arguments de la commande ; voir ci-dessous les descriptions détaillées des arguments	dict
sequence (identifiant)	<facultatif> Identifiant de la commande, qui est renvoyé à la fin dans action_result_value	int

Veuillez noter que la liste suivante n'est pas exhaustive, mais qu'elle contient les commandes les plus importantes.

7.2.5.1 Connexion par jeton

La première commande à envoyer doit toujours être un appel de connexion pour utiliser la connexion active. Suivez la procédure de connexion à distance décrite dans la section Authentification sécurisée pour obtenir un jeton d'accès valide.

Fonction	token_login	
Module	core	
Arguments	token	session_id
Valeur de l'argument	le jeton de connexion personnalisé fourni par la commande de connexion	l'identifiant de session unique fourni par la commande de connexion
Type d'argument	Chaîne	Chaîne

```
{"action": "token_login",  
  "bridge": "core",  
  "arguments": {"token": "MY_ACCESS_TOKEN",  
                "session_id": "MY_SESSION_ID"}}
```

7.2.5.2 Demander le contrôle actif

Obtention du contrôle actif, de sorte que la connexion TCP devienne la station de contrôle active du robot. Veuillez noter qu'une requête est envoyée à la station de contrôle actuellement active, qui peut l'accepter ou la refuser.

Fonction	request_active_control
Module	core

```
{"action": "request_active_control",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.3 Libérer le contrôle actif

Céder le contrôle actif afin que la connexion TCP ne soit plus la station de contrôle active du robot.

Fonction	release_active_control
Module	core

```
{"action": "release_active_control",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.4 Interroger les données

Cette commande peut être utilisée comme signal de présence. Elle reçoit différents types de données.

Fonction	request_data
Module	core

```
{"action": "request_data",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.5 Mise en service

Demande au robot de démarrer ses moteurs.

Fonction	power_up
Module	core
Arguments	tool_id
Valeur de l'argument	ID d'un outil défini par l'utilisateur ou 0 pour la détection automatique de l'outil
Type d'argument	Entier ou chaîne de caractères

```
{"action": "power_up",  
  "bridge": "core",  
  "arguments": {"tool_id": 0}}
```

7.2.5.6 Mise hors service

Met fin à la connexion et ordonne au robot d'arrêter ses moteurs.

Fonction	power_down
Module	core

```
{"action": "power_down",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.7 Mise en pause

Met en pause toutes les fonctions de mouvement et applications en cours d'exécution.

Fonction	pause
Module	core

```
{"action": "pause",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.8 Reprendre

Reprend une application ou un mouvement s'il est en pause.

Fonction	resume
Module	core

```
{"action": "resume",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.9 Activation du mode de commande manuelle (HGC)

Libère les articulations du robot afin qu'elles se trouvent en mode de compensation gravitationnelle et puissent être déplacées manuellement. L'état doit être « Prêt » et passe à « Commande manuelle ».

Fonction	release
Module	core
Arguments	joint_ids
Valeur de l'argument	<facultatif> Liste des identifiants d'axes pour lesquels le mode manuel doit être activé. Par défaut, le mode manuel est activé pour tous les axes.
Type d'argument	Liste

```
{"action": "release",  
  "bridge": "core",  
  "arguments": {"joint_ids": ["1", "2", "5", "6"]}}
```

7.2.5.10 Désactivation du mode de commande manuelle (HGC)

Verrouille les articulations du robot afin qu'elles soient en mode de contrôle de position et ne puissent plus être déplacées manuellement. L'état doit être « Commande manuelle » et passe à « Prêt ».

Fonction	hold
Module	core
Arguments	joint_ids
Valeur de l'argument	<facultatif> Liste des identifiants d'axes à conserver. Par défaut, tous les axes sont conservés.
Type d'argument	Liste

```
{"action": "hold",  
  "bridge": "core",  
  "arguments": {"joint_ids": ["1", "2", "5", "6"]}}
```

7.2.5.11 Arrêt

Interrompt toutes les fonctions de mouvement et applications en cours d'exécution.

Fonction	stop
Module	core

```
{"action": "stop",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.12 Changer de projet

Permet de passer du projet de base de données actuel au projet ayant un identifiant spécifique. Le statut doit être « Prêt ».

Fonction	change_project
Module	core
Arguments	project_id
Valeur de l'argument	ID du projet vers lequel il faut basculer. Le projet par défaut a l'ID 1.
Type d'argument	Entier

```
{"action": "change_project",  
  "bridge": "core",  
  "arguments": {"project_id": 1}}
```

7.2.5.13 Exécuter une application

Exécute une application spécifiée par son nom ou son ID. Le statut doit être « Prêt » et passe à « En cours d'exécution ». Pendant qu'une application est en cours d'exécution, le robot peut être mis en pause. Dès qu'elle est terminée, le statut repasse à « Prêt ».

Fonction	execute_script	
Module	core	
Arguments	script_name	script_id
Valeur de l'argument	Le nom de l'application à exécuter.	L'identifiant de l'application à exécuter.
Type d'argument	Chaîne	Entier

```
{"action": "execute_script",  
  "bridge": "core",  
  "arguments": {"script_id": 3}}
```

```
{"action": "execute_script",  
  "bridge": "core",  
  "arguments": {"script_name": "my_application"}}
```

7.2.5.14 Tester l'application

Exécute le code d'une application en mode test. Le statut doit être « Prêt » et passe à « En cours d'exécution ». Pendant qu'une application est en cours d'exécution, le robot peut être mis en pause. Dès qu'elle est terminée, le statut repasse à « Prêt ».

Fonction	test_script		
Module	core		
Arguments	code_script	script_type	script_id
Valeur de l'argument	Le code Python de l'application à tester.	Le type du script : • « main » • « parallel » • « background »	L'ID du script à tester. Si le script n'existe pas dans la base de données, l'ID est 0.
Type d'argument	Chaîne	Chaîne	Entier

```
{"action": "test_script",  
  "bridge": "core",  
  "arguments": {"script_code": "print('Hello World')",  
               "script_type": "main",  
               "script_id": 0}}
```

7.2.5.15 Réception d'un message

Utilisez cette fonction pour envoyer un message depuis votre canal de connexion vers une application. Le message peut être reçu à l'aide de la fonction **receive_message** dans l'application.

Fonction	script_receive
Module	core
Arguments	message
Valeur de l'argument	Message reçu par l'application.
Type d'argument	Divers

```
{"action": "script_receive",  
  "bridge": "core",  
  "arguments": {"message": 123}}
```

7.2.5.16 Se déplacer vers une pose

Déplace le robot vers une pose ayant un identifiant ou un nom spécifique. L'état doit être « Prêt » et passe à « En cours ». Pendant que le robot se déplace, le mouvement peut être mis en pause. Dès que le mouvement est terminé, l'état repasse à « Prêt ».

Fonction	move_pose
Module	core
Arguments	pose_id
Valeur de l'argument	ID ou nom de la pose vers laquelle il faut se déplacer.
Type d'argument	Entier ou chaîne de caractères

```
{"action": "move_pose",  
  "bridge": "core",  
  "arguments": {"pose_id": 2}}
```

```
{"action": "move_pose",  
  "bridge": "core",  
  "arguments": {"pose_id": "start_pose"}}
```

7.2.5.17 Lire le parcours

Lecture du chemin avec un identifiant ou un nom spécifique. L'état doit être « Prêt » et passe à « En cours ». Pendant que le robot se déplace, le mouvement peut être mis en pause. Dès que le mouvement est terminé, l'état repasse à « Prêt ».

Fonction	play_path
Module	core
Arguments	path_id
Valeur de l'argument	ID ou nom du chemin à exécuter.
Type d'argument	Entier ou chaîne de caractères

```
{"action": "play_path",  
  "bridge": "core",  
  "arguments": {"path_id": 2}}
```

```
{"action": "play_path",  
  "bridge": "core",  
  "arguments": {"path_id": "p1"}}
```

7.2.5.18 Changer d'outil

Permet de basculer la fonction de l'outil, généralement « Supprimer » et « Placer ». Le statut doit être « Prêt » et passe à « En cours ». Dès que la fonction est terminée, le statut repasse à « Prêt ».

Fonction	toggle_tool
Module	core
Arguments	pick
Valeur de l'argument	true pour soulever un objet, false pour le poser. Ceci est déterminé par le comportement de préhension.
Type d'argument	Booléen

```
{"action": "toggle_tool",  
  "bridge": "core",  
  "arguments": {"pick": false}}
```

7.2.5.19 Écriture des sorties numériques et analogiques

Décrit une ou plusieurs sorties (**numériques** ou **analogiques**) du contrôleur **principal** ou du contrôleur **d'outil** (lignes de signal transmises du contrôleur principal à l'outil). Une fonction distincte (avec les mêmes arguments) est utilisée pour chacune de ces combinaisons. Vous trouverez une liste de toutes les E/S dans le menu de configuration des E/S ou dans le menu E/S.

Fonction	write_digital_main_output write_analog_main_output write_digital_tool_output write_analog_tool_io	
Module	core	
Arguments	identificateurs	valeurs
Valeur de l'argument	Il existe plusieurs façons d'identifier une E/S unique, soit par • numéro (entier) • un nom défini par l'utilisateur (chaîne de caractères) Lorsque plusieurs E/S sont écrites simultanément, vous pouvez combiner des numéros ou des noms définis par l'utilisateur dans une liste.	En fonction du type d'« identifiants » choisi, les valeurs peuvent être : • vrai/faux (booléen) pour les E/S numériques • des nombres (flottants) pour les E/S analogiques ou une liste de ces valeurs lorsque plusieurs E/S sont écrites simultanément.
Type d'argument	Entier / Chaîne / Liste	Booléen / Flottant / Liste

```
{
  "action": "write_digital_tool_io",
  "bridge": "core",
  "arguments": {"identifiers": 3, "values": true}}
```

Cet exemple décrit la sortie numérique 3 de l'outil et la définit sur true.

```
{
  "action": "write_analog_tool_io",
  "bridge": "core",
  "arguments": {"identifiers": "my_analog_output", "values": 9.5}}
```

Cet exemple décrit la sortie analogique de l'outil, à laquelle le nom personnalisé « my_analog_output » a été attribué, et la règle sur 9,5 V.

```
{
  "action": "write_digital_main_io",
  "bridge": "core",
  "arguments": {"identifiers": [1, 2], "values": [true, false]}}
```

Cet exemple décrit les sorties numériques principales 1 et 2 et définit 1 sur true et 2 sur false.

7.2.5.20 Répondre à une boîte de dialogue

Répond à une fenêtre de dialogue ouverte. La réponse dépend du type de dialogue.

Fonction	answer_dialog	
Module	core	
Arguments	dialog_id	dialog_answer
Valeur de l'argument	L'identifiant unique de la boîte de dialogue à laquelle la réponse est envoyée. Cet identifiant fait partie de la réponse « open_dialog » reçue.	La réponse à envoyer à ce dialogue.
Type d'argument	Chaîne	Booléen / Entier / Chaîne

```
{
  "action": "answer_dialog",
  "bridge": "core",
  "arguments": {
    "dialog_id": "5cee85a8-9ab2-4de8-b1ea-52ded8cc68c0",
    "dialog_answer": true
  }
}
```

7.2.5.21 Définir une variable d'état

Définit la valeur d'une variable d'état « variable_name » sur « variable_value ». La variable doit déjà exister dans l'application avec l'ID « script_id ». De plus, l'application doit être enregistrée dans la base de données. Cette commande peut être exécutée pendant qu'une application est en cours d'exécution.

Fonction	set_state_variable		
Module	core		
Arguments	script_id	nom_de_la_variable	valeur_de_la_variable
Valeur de l'argument	ID de l'application contenant la variable d'état à modifier.	Nom de la variable d'état à modifier.	Nouvelle valeur.
Type d'argument	Entier	Chaîne	Divers

```
{
  "action": "set_state_variable",
  "bridge": "core",
  "arguments": {
    "script_id": 2,
    "variable_name": "integer_1",
    "variable_value": 42
  }
}
```

7.2.5.22 Définir une variable globale

Définit la valeur d'une variable globale « name » sur « value ». Les variables globales sont conservées au-delà de la durée d'exécution d'une application et sont préservées même après le redémarrage du robot.

Fonction	set_global_variable	
Module	core	
Arguments	name	value
Valeur de l'argument	Nom de la variable globale	Valeur de la variable globale
Type d'argument	Chaîne	Divers

```
{"action": "set_global_variable",  
  "bridge": "core",  
  "arguments": {"name": "my_variable", "value": 3}}
```

7.2.5.23 Lire une variable globale

Lit la valeur d'une variable globale « name ». Les variables globales sont conservées au-delà de la durée d'exécution d'une application et sont préservées même après le redémarrage du robot.

Fonction	get_global_variable	
Module	core	
Arguments	name	
Valeur de l'argument	Nom de la variable globale	
Type d'argument	Chaîne	

```
{"action": "get_global_variable",  
  "bridge": "core",  
  "arguments": {"name": "my_variable"}}
```

7.2.5.24 Définir une variable partagée

Définit la valeur d'une variable partagée « name » sur « value ». Les variables partagées ne sont disponibles que pendant l'exécution d'une application. Elles sont supprimées une fois l'application terminée ou le robot arrêté.

Fonction	set_shared_variable	
Module	core	
Arguments	name	value
Valeur de l'argument	Nom de la variable partagée	Valeur de la variable partagée
Type d'argument	Chaîne	Divers

```
{"action": "set_shared_variable",  
  "bridge": "core",  
  "arguments": {"name": "my_variable", "value": 3}}
```

7.2.5.25 Lire une variable partagée

Lit la valeur d'une variable partagée « name ». Les variables partagées ne sont disponibles que pendant la durée d'exécution d'une application. Elles sont supprimées une fois l'application terminée ou le robot arrêté.

Fonction	get_shared_variable	
Module	core	
Arguments	name	
Valeur de l'argument	Nom de la variable partagée	
Type d'argument	Chaîne	

```
{"action": "get_shared_variable",  
  "bridge": "core",  
  "arguments": {"name": "my_variable"}}
```

7.3 Module de connexion REST

7.3.1 Établir une connexion

Le module de connexion REST met à disposition la ressource **https://<IP-ADDRESS>/rest/gateway** afin de recevoir les commandes des clients connectés. Veuillez noter qu'il n'existe qu'un seul canal de connexion. Si plusieurs utilisateurs se connectent via REST, ils partagent ce canal.

Utilisez le script d'exemple suivant pour prendre le contrôle du robot via la connexion REST, puis envoyer des commandes au robot. Notez que vous devez être connecté et détenir le contrôle actif avant d'envoyer des commandes. Il est donc recommandé de partager le contrôle et de vous déconnecter dès que la connexion n'est plus nécessaire.

```
import json
import requests

IP_ADDRESS = "10.0.0.203"
TOKEN = "MY_ACCESS_TOKEN"
SESSION_ID = "MY_SESSION_ID"

REST_URL = f"https://{IP_ADDRESS}/rest/gateway"
HEADERS = {"Accept": "application/json", "Content-Type": "application/json"}

def request(method, url, headers, data, timeout=None):
    try:
        response = method(url=url, headers=headers, data=data, timeout=timeout,
                           verify=False)
        response_code = response.status_code

        # this is equal to 'response.status_code < 400' (errors have codes 400+)
        if response.ok:
            # try to parse json payload
            try:
                response_payload = response.json()
            except json.decoder.JSONDecodeError:
                # usually this happens because the response payload is empty
                # this happens for example when a request returns "", 200
                response_payload = dict()
                print("response payload json was empty")
            else:
                print("response.ok returned False")

        except requests.Timeout:
            response_payload = dict()
            response_code = 408 # timeout
        except Exception as e:
            # usually connection problems...
            print(f"rest client could not connect to specified endpoint: {e}")
            response_payload = dict()
            response_code = 404 # not found

    return response_code, response_payload

def post_request(data, timeout=None):
    return request(method=requests.post, url=REST_URL, headers=HEADERS, data=data,
```

```
timeout=timeout)

# login
data = {"bridge": "core", "action": "token_login", "arguments": {"token": "<TOKEN>",
"session_id": "<SESSION_ID>"}}
response = post_request(data, timeout=10.0)
# acquire the active control before start sending commands to the robot
data = {"bridge": "core", "action": "request_active_control", "arguments": {}}
response = post_request(data, timeout=1.0)

# add your commands here

# when you're done using this connection, give away the active control
data = {"bridge": "core", "action": "release_active_control", "arguments": {}}
response = post_request(data, timeout=1.0)
# when you're done working with this connection, you should logout
data = {"bridge": "core", "action": "logout", "arguments": {}}
response = post_request(data, timeout=1.0)
```

7.3.2 Envoyer des commandes

Outre les commandes mentionnées dans l'exemple ci-dessus, de nombreuses autres commandes sont disponibles. Veuillez consulter la liste des commandes du module de connexion TCP, car ces mêmes commandes sont également utilisées pour la connexion REST.

En plus d'envoyer un paramètre « bridge », « action », « arguments » et un paramètre « séquence » facultatif, vous pouvez également envoyer un paramètre « timeout » facultatif. Si vous envoyez un délai d'expiration, la requête REST ne renvoie pas la réponse standard (200, {}) en cas de succès, mais la commande est exécutée de manière bloquante, et vous recevez la réponse sous la forme (200, action_return_value), la valeur de retour de l'action contenant les informations suivantes : quelle fonction API a été appelée, si l'appel a abouti et quelle valeur de retour a été renvoyée par la fonction appelée.

7.3.3 Paramètres

Dans le menu des paramètres de l'interface de bureau, vous trouverez un paramètre permettant de désactiver l'authentification pour le module de connexion REST. La désactivation de l'authentification supprime la nécessité de se connecter et de se déconnecter.



Figure 7.3.3 : Paramètres du module REST

7.4 Module de connexion ROS

Le module de connexion ROS vous permet d'intégrer votre robot dans un environnement ROS 2 (Robot Operating System) et de communiquer avec AX Portal à l'aide de commandes ROS 2. Les instructions fournies dans ce chapitre sont compatibles avec ROS Jazzy et [CycloneDDS](#) ainsi qu'avec [ROS Middleware \(RMW\)](#).

 AVERTISSEMENT	Le module ROS donne accès à certaines fonctions de bas niveau. Par exemple, il est possible d'envoyer des commandes de contrôle directes aux moteurs du robot, qui sont exécutées immédiatement. Dans l'interface utilisateur, certaines fonctions nécessitent des vérifications supplémentaires ou des confirmations par boîte de dialogue qui peuvent ne pas être disponibles ici. Soyez donc particulièrement prudent lorsque vous contrôlez votre robot directement via ROS.
---	---

Remarque :

- Il s'agit de la première version d'AX Portal avec ROS 2 ; des fonctionnalités supplémentaires sont attendues à l'avenir.
- Actuellement, le module ROS n'est pas disponible avec le simulateur AX Portal, mais uniquement avec des robots réels.

7.4.1 Installation

Pour communiquer avec le module ROS d'AX Portal, vous devez disposer d'une installation ROS sur votre ou vos ordinateurs. Pour la configuration, la documentation officielle de ROS proposent différentes options, par exemple un paquet deb, une image Docker, une installation à partir du code source, etc. Dans cet exemple, nous utilisons l'image Docker officielle de ROS pour communiquer avec le robot. Le fichier docker-compose suivant crée un environnement ROS simple sur votre ordinateur :

```
services:
  ros2_development:
    image: osrf/ros:jazzy-desktop
    container_name: "ros2_development"
    command: /bin/bash -c
      "apt update &&
      apt install -y ros-jazzy-rmw-cyclonedds-cpp &&
      sleep infinity"
    environment:
      - RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
      - CYCLONEDDS_URI=/ros/cyclonedds_config.xml

    network_mode: "host"

volumes:
  - "./cyclonedds_config.xml:/ros/cyclonedds_config.xml"
```

Notez que lors du démarrage, nous installons `ros-jazzy-rmw-cyclonedds-cpp` et définissons `RMW_IMPLEMENTATION` sur `rmw_cyclonedds_cpp` afin d'indiquer à ROS d'utiliser

CycloneDDS – le même RMW qui s'exécute également sur le robot. De plus, nous partageons un fichier `cyclonedds_config.xml` et définissons `CYCLONEDDS_URI` sur son emplacement afin d'utiliser une configuration CycloneDDS personnalisée.

7.4.2 Configuration

La communication ROS entre le robot et votre ordinateur nécessite des ajustements sur les deux appareils. Sur votre ordinateur portable, vous devez définir une nouvelle variable d'environnement et fournir un fichier de configuration réseau. Sur le robot, vous n'avez pas d'accès direct à ces éléments, mais vous pouvez les modifier dans l'interface utilisateur, dans la section « ROS » du menu des paramètres :

The screenshot displays the 'ROS Function Settings' window. It contains the following elements:

- Network Interface:** A blue dropdown menu currently showing 'Wifi'.
- Multicast:** A toggle switch with a blue circle and a white 'X' icon, currently in the 'off' position.
- IP Addresses of ROS 2 Peers:** A list of two text input fields. The first contains '192.168.0.1' and the second contains '192.168.0.2'. To the right of each field is a blue trash icon. Below the list is a blue '+' button to add more addresses.
- ROS Domain ID:** A blue dropdown menu currently showing '15'.

Figure 7.4.2 : Paramètres du module ROS

Dans les sections suivantes, vous découvrirez la signification de chaque paramètre. Vous apprendrez également quels ajustements correspondants sont nécessaires sur votre ordinateur.

7.4.3 Détection du réseau avec Cyclone DDS

Cyclone DDS a déjà été mentionné à plusieurs reprises. De quoi s'agit-il ? En tant qu'utilisateur d'AX Portal, voici ce que vous devez savoir :

Lorsque Cyclone DDS est sélectionné comme middleware ROS, il se charge de distribuer les paquets de données liés à ROS sur votre réseau. Il est chargé d'établir une connexion réseau entre vos appareils afin que ROS 2 puisse envoyer des données via ces connexions.

7.4.3.1 Interface réseau

Tout d'abord, Cyclone DDS doit savoir quelle interface réseau est utilisée par ROS. Le paramètre par défaut est Ethernet.

Utilisez le paramètre « Interface réseau » sous « Paramètres des fonctions ROS » pour sélectionner l'interface souhaitée.

Remarque : lors du démarrage du module ROS via AX Portal, Cyclone DDS analyse l'interface réseau à la recherche des adresses IP des ordinateurs pairs. Si cette interface est inaccessible, le module ROS

ne se charge pas. Par conséquent, si vous avez défini l'interface réseau sur Ethernet, assurez-vous que le câble est branché *avant de démarrer le robot*.

7.4.3.2 Méthodes de détection

Cyclone DDS propose deux méthodes pour détecter d'autres appareils sur le réseau : la multidiffusion et le peer-to-peer. Avec la multidiffusion, chaque paquet réseau est envoyé à tous les clients joignables dans la zone réseau. L'avantage réside dans la simplicité : vous n'avez pas besoin de définir quels ordinateurs sont autorisés à communiquer entre eux. Dans cette configuration, la variable d'environnement `ROS_DOMAIN_ID` (voir ci-dessous) reste disponible pour empêcher que deux robots ne mélangent leurs messages.

L'inconvénient réside dans la surcharge réseau : tous les paquets sont traités par chaque client du réseau. Même la définition de différents `ROS_DOMAIN_ID` ne peut empêcher cela. Même si ROS n'est pas installé sur un ordinateur ; comme Cyclone DDS fonctionne sur la base du protocole UDP, tous les ordinateurs du réseau reçoivent et traitent néanmoins ces paquets UDP. Pour les petits réseaux et les faibles volumes de données, cela ne pose généralement pas de problème. Toutefois, si un robot envoyait par exemple un flux de caméra à chaque client d'un grand réseau, cela pourrait ralentir sensiblement l'ensemble du réseau. Dans un tel scénario, il est plus sûr de définir explicitement, dans la configuration de Cyclone DDS, une série d'adresses IP autorisées à communiquer entre elles via des connexions peer-to-peer.

Les deux sous-sections suivantes vous expliquent comment configurer votre réseau pour la communication multicast ou peer-to-peer. Remarque : sur la plupart des distributions Linux, la commande ``ip addr show`` est utile pour déterminer vos adresses IP et les noms d'interface réseau associés.

7.4.3.2.1 Peer-to-Peer

Dans les instructions d'installation ci-dessus, nous avons intégré un fichier de configuration Cyclone DDS dans le fichier `docker-compose`. Voici un exemple de configuration pour Cyclone DDS afin de configurer votre ordinateur pour la communication peer-to-peer :

```
<?xml version='1.0' encoding='utf-8'?>
<CycloneDDS xmlns="https://cdds.io/config"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://cdds.io/config
https://raw.githubusercontent.com/eclipse-cyclonedds/cyclonedds/master/etc/cyclonedds.xsd"
">
  <Domain id="any">
    <General>
      <Interfaces>
        <NetworkInterface name="INTERFACE_NAME" multicast="false" />
      </Interfaces>
      <AllowMulticast>false</AllowMulticast>
      <EnableMulticastLoopback>false</EnableMulticastLoopback>
    </General>
    <Discovery>
      <MaxAutoParticipantIndex>119</MaxAutoParticipantIndex>
      <ParticipantIndex>auto</ParticipantIndex>
      <Peers>
        <Peer address="192.168.1.1" />
        <Peer address="192.168.1.2" />
      </Peers>
    </Discovery>
  </Domain>
</CycloneDDS>
```

Commencez par remplacer INTERFACE_NAME par le nom réel de l'interface réseau de votre ordinateur utilisée pour la communication ROS. Exemples : enp1s0, wlp2s0, etc.

Ensuite, adaptez la liste sous « Peers » afin qu'elle reflète les adresses IP des ordinateurs de votre réseau que vous souhaitez connecter entre eux via ROS.

À l'instar du fichier ci-dessus, un fichier similaire est enregistré sur votre robot. Vous ne pouvez pas y accéder directement, mais vous pouvez modifier son contenu via AX Portal, dans la section « Paramètres des fonctions ROS » du menu des paramètres. Le paramètre « Adresses IP des pairs ROS 2 » y propose une liste de champs de texte que vous pouvez remplir avec des adresses IP.

Remarque :

- Un redémarrage du robot est nécessaire pour que la modification prenne effet.
- Si une adresse IP de la liste est erronée, AX Portal affiche un message d'erreur et la nouvelle liste n'est pas enregistrée.
- La liste doit contenir l'adresse IP du robot lui-même. Actuellement, celle-ci n'est pas lue automatiquement.
- Si une adresse IP est supprimée de la liste et que la nouvelle liste est enregistrée, l'adresse IP est également supprimée de la configuration Cyclone DDS.
- En raison des paramètres du pare-feu, il peut être nécessaire que tous les participants ROS se trouvent dans le même sous-réseau.

7.4.3.2.2 Multicast

Voici un autre exemple de configuration pour Cyclone DDS afin de configurer votre ordinateur pour la multidiffusion :

```
<?xml version="1.0" encoding="utf-8"?>
<CycloneDDS
  xmlns="https://cdds.io/config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://cdds.io/config
https://raw.githubusercontent.com/eclipse-cyclonedds/cyclonedds/master/etc/cyclonedds.xsd
"
>
  <Domain Id="any">
    <General>
      <Interfaces>
        <NetworkInterface name="INTERFACE_NAME" multicast="true" />
      </Interfaces>
      <AllowMulticast>true</AllowMulticast>
      <EnableMulticastLoopback>true</EnableMulticastLoopback>
    </General>
  </Domain>
</CycloneDDS>
```

Ici aussi, remplacez `INTERFACE_NAME` par le nom réel de l'interface réseau de votre ordinateur utilisée pour la communication ROS. Exemples : `enp1s0`, `wlp2s0`, etc.

Pour faire passer votre robot en mode Multicast, il vous suffit d'activer le bouton de configuration « Multicast ». Là encore, un redémarrage du robot est nécessaire pour que la modification prenne effet.

7.4.4 Établir la connexion

Pour démarrer l'environnement ROS sur votre ordinateur, accédez à votre fichier `docker-compose` et exécutez :

```
docker compose up -d
```

Vérifiez si le conteneur Docker est en cours d'exécution à l'aide de

```
docker ps -a
```

ce qui génère un résultat similaire à celui-ci :

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
132bdd6fa4de	osrf/ros:jazzy-desktop	"/ros_..."	...	...	...	...

Passez en mode interactif du conteneur à l'aide de la commande suivante :

```
docker exec -it ros2_development bash
```

N'oubliez pas de charger le fichier de configuration ROS à chaque fois que vous entrez dans le conteneur :

```
source /opt/ros/jazzy/setup.bash
```

Pour pouvoir faire fonctionner plusieurs robots sur le même réseau, nous utilisons un ROS_DOMAIN_ID différent sur chaque robot. Dans AX Portal, ouvrez les « Paramètres des fonctions ROS » et définissez l'identifiant de domaine ROS souhaité. La plage autorisée est comprise entre 6-41. Notez également ici que le robot doit être redémarré après cette action. Revenez maintenant au terminal dans lequel votre conteneur Docker ROS est ouvert et définissez la même valeur pour ROS_DOMAIN_ID :

```
export ROS_DOMAIN_ID=SELECTED_NUMBER
```

Vous devriez désormais pouvoir voir un nœud AX Portal-ROS nommé « robot » à l'aide de la commande suivante :

```
ros2 node list
```

7.4.5 Accès aux données du robot via ROS

Dès que vous voyez le nœud, vous pouvez afficher la liste des sujets disponibles en exécutant la commande suivante :

```
ros2 topic list
```

Les sujets ROS 2 vous permettent de diffuser en direct les données de votre robot. Pour vous abonner, par exemple, aux positions des axes, aux vitesses, etc., exécutez la commande suivante :

```
ros2 topic echo /joint_states
```

7.4.6 Fonctionnalités ROS 2 dans les scripts

Vous pouvez importer relpy directement dans des scripts. Cela vous permet de créer des objets ROS 2 tels que des éditeurs, des abonnés, etc.

Le module ROS crée son propre nœud ROS. Utilisez la fonction `ros_handler.ros_node()` pour y accéder. Attention : n'exécutez pas `relpy.spin` sur ce nœud, car cela se fait déjà au sein du module ROS.

7.4.7 Appeler une action

Le nœud ROS AX Portal crée un serveur d'actions nommé « /robot/execute_function » pour l'exécution des fonctions Bridge. L'action dispose d'une définition personnalisée ExecuteFunction issue du paquet generic_msgs (<https://github.com/Epson-Robots/epson-ax-ros2>), qui doit être installé pour récupérer la définition de l'action . Clonez le paquet dans votre espace de travail et compilez-le avec colcon :

```
mkdir -p /colcon_ws/src
cd /colcon_ws/src
git clone https://github.com/Epson-Robots/epson-ax-ros2
cd ..
colcon build --packages-up-to generic_msgs
source install/setup.bash
```

Pour envoyer des actions ROS2 au robot, votre client doit prendre le contrôle actif. Pour ce faire, exécutez la commande suivante dans votre terminal et confirmez la boîte de dialogue dans l'interface graphique AX Portal :

```
ros2 action send_goal --feedback /execute_function generic_msgs/action/ExecuteFunction
"{action: request_active_control, bridge: core, arguments: '{}'}"
```

Vous pouvez désormais exécuter une fonction de pont à partir d'un terminal à l'aide des commandes suivantes :

```
ros2 action send_goal --feedback /execute_function generic_msgs/action/ExecuteFunction
"{action: read_actuator_position, bridge: core, arguments: '{}\"actuator_ids\": 6}'}"

ros2 action send_goal /execute_function generic_msgs/action/ExecuteFunction "{action:
move_joints, bridge: core, arguments: '{}\"joints\": 6, \"joint_position\": 100.0}'}"
```

7.4.8 Paramètres d'authentification

Le module ROS ne fonctionne actuellement qu'en mode sans authentification. Dans le menu des paramètres de l'interface de bureau, vous devez désactiver manuellement l'authentification pour ce module de connexion spécifique.

Authentication Settings

Disable Authentication for ROS



Figure 7.4.8 : Paramètres d'authentification ROS

7.5 Module de connexion Modbus

7.5.1 Aperçu

Le protocole utilisé pour le module Modbus est [TCP Modbus](#). Côté serveur, le module est basé sur Python et a été mis en œuvre à l'aide de la [bibliothèque PyModbus](#). Vous trouverez plus de détails dans la [documentation PyModbus](#). Le module est divisé en registres de lecture (Analog Input Registers, 30000-39999) et en registres d'écriture (Analog Output Holding Registers, 40000-49999). N'importe quel client TCP Modbus peut être utilisé pour communiquer avec le serveur. Le port à utiliser est le **5020**. Dans la dernière section de ce chapitre, vous trouverez plusieurs exemples de code Python, dont certains expliquent comment établir une connexion avec le serveur à l'aide d'un script Python.

7.5.2 Paramètres

La connexion Modbus ne fonctionne actuellement qu'en mode sans authentification. Dans le menu des paramètres de l'interface de bureau, vous devez désactiver manuellement l'authentification pour ce module de connexion spécifique.

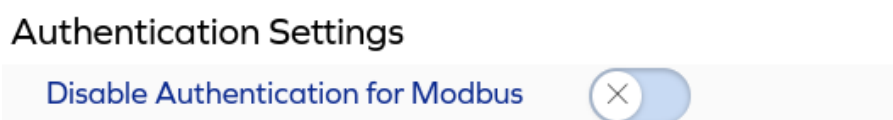


Figure 7.5.2 : Paramètres du module Modbus

7.5.3 Registres d'écriture (description simple)

Le module Modbus réagit à toute modification des informations tant dans le registre de commande que dans le registre de propriétés. L'effacement des données des registres d'écriture relève de la responsabilité de l'utilisateur.

7.5.3.1 Registre de commande

Écrivez un bit unique dans le registre de commande (30000) pour exécuter une commande. Certaines commandes nécessitent un argument dans le registre de données de commande (30001). Si vous ne pouvez pas écrire dans les deux registres simultanément, veuillez à écrire d'abord dans le registre de données de commande.

Adresse absolue : 30000 - 30001 Adresse relative : 0x000 - 0x001		
Bit	Registre de commande (30000)	Données (30001)
0	Mise en service	ID d'outil
1	Mise hors service	
2	Arrêt	

3	Mise en pause	
4	Reprendre	
5	Calibrer	ID de l'application
6	Activer la commande manuelle (HGC)	
7	Désactiver la commande manuelle (HGC)	
8	Changer de projet	ID du projet
9	Exécuter l'application	ID de l'application
10	Se déplacer vers la pose	ID de la pose
11	Lire le chemin	ID du chemin
12	Placer avec l'outil	
13	Saisir avec l'outil	
14 ... 15	<i>Réservé</i>	

7.5.3.2 Registre de propriétés

Écrivez un bit unique dans le registre de propriétés (30002) pour exécuter une commande. Certaines commandes nécessitent un argument dans le registre de données de propriétés (30003). Si vous ne pouvez pas écrire dans les deux registres simultanément, veuillez à écrire d'abord dans le registre de données de propriétés.

Adresse absolue : 30002 - 30003 Adresse relative : 0x002 - 0x003		
Bit	Registre de propriétés (30002)	Données (30003)
0	Écriture des sorties numériques	État des E/S
1	Définir les événements d'application	ID d'événement
2	Acquitter l'erreur	
3	Lire la position	
4	Demander la commande active	
5	Relâcher la commande active	
6	Passer en mode manuel	
7	Passer en mode automatique	

8	Réinitialiser la fonction de sécurité	
9	Reprendre le mouvement	
10 ... 15	<i>Réservé</i>	

7.5.3.3 Registres d'écriture définis par l'utilisateur

Adresse absolue : 30016 Adresse relative : 0x010		...	Adresse absolue : 30255 Adresse relative : 0x0FF	
0 ... 15	Données personnalisées		0 ... 15	Données personnalisées

Utilisez les registres d'écriture utilisateur pour envoyer des informations à votre application. La fonction **modbus.write_user_register** de l'application permet de lire les données définies par l'utilisateur.

7.5.4 Registres d'écriture (description détaillée)

7.5.4.1 Ne rien faire

La suppression des données des registres d'écriture relève de la responsabilité de l'utilisateur. Pour effacer soit le registre de commande (30000), soit le registre de propriétés (30002), il suffit d'écrire la valeur entière 0.

0x000	0x001	0x002	0x003
0		0	

7.5.4.2 Mise en service

Avant de pouvoir travailler avec le robot, celui-ci doit mettre ses moteurs sous tension. L'identifiant de l'outil personnalisé à connecter est indiqué par le chiffre n. Le robot passe de l'état « **Hors service** » (1, bit 0) à l'état « **Prêt** » (8, bit 3).

0x000	0x001	0x002	0x003
1	n		

7.5.4.3 Mise hors service

À la fin de la journée de travail, le robot peut couper ses moteurs. Le robot peut être mis hors tension quel que soit son état. Dès que le robot est mis hors tension, son état passe à « **Hors service** » (1, bit 0).

0x000	0x001	0x002	0x003
2			

7.5.4.4 Arrêt

Interrompez tous les mouvements du robot et ramenez l'état de « **En cours** » (64, bit 6) à « **Prêt** » (8, bit 3).

0x000	0x001	0x002	0x003
4			

7.5.4.5 Mise en pause

Arrêtez le robot lorsqu'il est en état « **En marche** » (64, bit 6). L'état passe à « **En pause** » (128, bit 7).

0x000	0x001	0x002	0x003
8			

7.5.4.6 Reprendre

Faites reprendre le robot s'il se trouve à l'état **En pause** (128, bit 7). L'état passe à **En cours d'exécution** (64, bit 6).

0x000	0x001	0x002	0x003
16			

7.5.4.7 Calibrage

Le calibrage n'a actuellement aucun effet. Ce bit est réservé.

0x000	0x001	0x002	0x003
32			

7.5.4.8 Activation de la commande manuelle (HGC)

Activez la commande manuelle pour toutes les articulations du robot afin qu'elles soient en mode de compensation de la gravité et puissent être déplacées manuellement. L'état doit être **Prêt** (8, bit 3) et passe à **Commande manuelle** (2048, bit 11).

0x000	0x001	0x002	0x003
64			

7.5.4.9 Désactiver le mode manuel (HGC)

Maintenez toutes les articulations du robot de manière à ce qu'elles se trouvent en mode de régulation de position et ne puissent plus être déplacées manuellement. L'état doit être « **Commande manuelle** » (2048, bit 11) et passe à « **Prêt** » (8, bit 3).

0x000	0x001	0x002	0x003
128			

7.5.4.10 Changer de projet

Passez du projet de base de données actuel au projet portant l'ID n. L'état doit être **Prêt** (8, bit 3).

0x000	0x001	0x002	0x003
256	n		

7.5.4.11 Démarrer l'application

Exécutez l'application portant l'ID n. L'état doit être **Prêt** (8, bit 3) et passe à **En cours d'exécution** (64, bit 6). Pendant qu'une application est en cours d'exécution, le robot peut être mis en pause. Dès qu'elle est terminée, l'état repasse à **Prêt** (8, bit 3).

0x000	0x001	0x002	0x003
512	n		

7.5.4.12 Se déplacer vers une pose

Déplacez le robot vers la pose dont l'ID est n. L'état doit être **Prêt** (8, bit 3) et passe à **En cours** (64, bit 6). Pendant que le robot se déplace, le mouvement peut être mis en pause. Dès que le mouvement est terminé, l'état repasse à **Prêt** (8, bit 3).

0x000	0x001	0x002	0x003
1024	n		

7.5.4.13 Lire le parcours

Lisez le chemin portant l'ID n. L'état doit être **Prêt** (8, bit 3) et passe à **En cours** (64, bit 6). Pendant que le robot se déplace, le mouvement peut être mis en pause. Dès que le mouvement est terminé, l'état repasse à **Prêt** (8, bit 3).

0x000	0x001	0x002	0x003
2048	n		

7.5.4.14 Placer avec l'outil

Placez un objet à l'aide de l'outil (par exemple en ouvrant le préhensuer). L'état doit être **Prêt** (8, bit 3) et passe à **En cours** (64, bit 6). Pendant que le préhenseur se déplace, le mouvement peut être mis en pause. Dès que le mouvement est terminé, l'état repasse à **Prêt** (8, bit 3).

0x000	0x001	0x002	0x003
4096			

7.5.4.15 Saisir avec l'outil

Soulevez un objet à l'aide de l'outil (par exemple en fermant le préhenseur). L'état doit être **Prêt** (8, bit 3) et passe à **En cours** (64, bit 6). Pendant que le préhenseur se déplace, le mouvement peut être mis en pause. Dès que le mouvement est terminé, l'état repasse à **Prêt** (8, bit 3).

0x000	0x001	0x002	0x003
8192			

7.5.4.16 Écriture sur les sorties numériques

Fonction réservée à l'écriture des sorties numériques.

0x000	0x001	0x002	0x003
		1	d

7.5.4.17 Définition des événements d'application

Si une application utilisateur doit attendre un événement envoyé par le dispositif Modbus, les événements d'application peuvent être utilisés. Au total, 16 événements sont disponibles. En fonction de la valeur de la donnée d, les événements sont activés ou désactivés. La fonction **d'application modbus.wait_for_event** attend qu'un événement spécifique soit activé. Si d est égal à 0, tous les événements sont désactivés. Si d est égal à 65535, tous les événements sont activés. Si d est égal à 9, le premier et le quatrième événement sont activés, les autres sont désactivés.

0x000	0x001	0x002	0x003
		2	d

7.5.4.18 Acquitter les erreurs

Les erreurs doivent être acquittées avant d'être effacées. Cela n'est toutefois nécessaire que pour lire les erreurs. L'acquiescement n'a aucune incidence sur l'état du robot. Si les codes d'erreur majeur et mineur (40004-40005) sont tous deux égaux à 0, aucune autre erreur ne doit être acquittée.

0x000	0x001	0x002	0x003
		4	

7.5.4.19 Lecture de la position

Actualisez les données de position du robot. Les angles d'articulation et les coordonnées actuels du robot sont écrits dans les registres de lecture correspondants (40256-40267) dès que les données de position sont demandées.

0x000	0x001	0x002	0x003
		8	

7.5.4.20 Demander le contrôle actif

Demandez le contrôle actif afin de pouvoir envoyer des commandes au robot. Les registres de lecture de données sont accessibles sans contrôle actif. L'interface qui détient actuellement le contrôle actif est notifiée et doit céder activement le contrôle. Le registre de lecture correspondant (40008) est activé dès que le contrôle a été accordé.

0x000	0x001	0x002	0x003
		16	

7.5.4.21 Libérer le contrôle actif

Libérez le contrôle actif afin qu'une autre interface puisse envoyer des commandes au robot. Les registres de lecture de données et les registres de lecture sont accessibles sans contrôle actif. Le registre de lecture correspondant (40008) passe à 0.

0x000	0x001	0x002	0x003
		32	

7.5.4.22 Passer en mode manuel

Passez en mode manuel. Dans ce mode, le robot ne se déplace que lorsque l'utilisateur appuie sur le bouton de validation. Veuillez noter qu'une réinitialisation de sécurité est nécessaire pour changer de mode.

0x000	0x001	0x002	0x003
		64	

7.5.4.23 Passer en mode automatique

Passez en mode automatique. Veuillez noter qu'une réinitialisation de sécurité est nécessaire pour changer de mode.

0x000	0x001	0x002	0x003
		128	

7.5.4.24 Réinitialisation de la sécurité

Réinitialisez le système de sécurité après une interruption. Pour certaines fonctions de sécurité, le robot nécessite une action claire et intentionnelle de la part de l'utilisateur pour continuer. Pour vérifier si une telle action est requise, utilisez le registre de lecture correspondant (40009).

0x000	0x001	0x002	0x003
		256	

7.5.4.25 Lancer l'interruption de mouvement

Après une interruption pour des raisons de sécurité, le robot peut redémarrer et poursuivre son mouvement. Cette fonction permet de remettre le robot en mouvement. Sinon, une commande d'arrêt peut être envoyée pour interrompre l'application en cours. Pour vérifier si une telle action est nécessaire, utilisez le registre de lecture correspondant (40009).

0x000	0x001	0x002	0x003
		512	

7.5.5 Registre de lecture

7.5.5.1 Registre d'état

Utilisez le registre d'état pour lire l'état actuel du robot. L'état est fourni sous forme de numéro de bit.

7.5.5.1.1 1er registre d'état

Adresse absolue : 40000 Adresse relative : 0x000	
0	Hors service
1	Connecter
2	Chargement
3	Prêt
4	En cours de traitement
5	Calibrage
6	En cours
7	Mode autonome
8	En pause
9	Reprise
10	Arrêt de protection
11	Dispositif de protection

12	Limitation de position
13	Commande manuelle (HGC)
14	Collision
15	Arrêt d'urgence

7.5.5.1.2 2ème Registre d'état

Adresse absolue : 40001 Adresse relative : 0x001	
0	Arrêt normal
1	Erreur
2	Mise hors service
3	En cours d'arrêt
4.. 15	<i>réservé</i>

7.5.5.1.3 3ème Registre d'état

Adresse absolue : 40002 Adresse relative : 0x002	
0 ... 15	<i>réservé</i>

7.5.5.1.4 4ème Registre d'état

Adresse absolue : 40003 Adresse relative : 0x003	
0 ... 15	<i>réservé</i>

7.5.5.1.5 Registre des erreurs majeures

Adresse absolue : 40004 Adresse relative : 0x004	
0 ... 15	Informations sur la catégorie d'erreur.

Le registre des erreurs majeures lit les 2 premiers octets de l'identifiant de l'erreur actuelle. Acquitez l'erreur via le registre des propriétés pour lire l'erreur suivante.

Vous trouverez des détails sur la signification des valeurs du registre d'erreurs dans le tableau de correspondance des registres d'erreurs.

7.5.5.1.6 Registre des erreurs mineures

Adresse absolue : 40005 Adresse relative : 0x005	
0 ... 15	Informations sur l'erreur au sein d'une catégorie.

Le registre d'erreurs mineures lit les 2 derniers octets de l'identifiant de l'erreur actuelle. Acquitez l'erreur via le registre de propriétés afin de lire l'erreur suivante.

Vous trouverez des détails sur la signification des valeurs du registre d'erreurs dans le tableau de correspondance des registres d'erreurs.

7.5.5.1.7 Registre d'application

Adresse absolue : 40006 Adresse relative : 0x006	
0 ... 15	ID d'application actuel

Le registre d'application lit l'ID de l'application en cours d'exécution.

7.5.5.1.8 Registre de calibrage

Adresse absolue : 40007 Adresse relative : 0x007	
0	Si le robot est calibré

Le registre de calibrage indique si le robot est calibré. 0 signifie que le robot n'est pas calibré. 1 signifie que le robot est calibré. Veuillez noter que le robot doit être connecté à son moteur pour fournir un retour d'information significatif concernant le calibrage.

7.5.5.1.9 Registre de commande active

Adresse absolue : 40008 Adresse relative : 0x008	
0	Indique si le canal Modbus a le contrôle actif

Le registre de contrôle actif indique si le canal Modbus détient le contrôle actif. 0 signifie que Modbus ne détient pas le contrôle actif. 1 signifie que Modbus détient le contrôle actif. Le canal Modbus peut demander le contrôle actif à l'aide de 0x30002 0x04 Contrôle actif.

7.5.5.1.10 Registre d'interruption

Le registre d'interruption affiche les options de dialogue d'interruption actives.

Adresse absolue : 40009 Adresse relative : 0x009	
0	Réinitialisation de sécurité requise
1	Poursuite du mouvement requise
2	Passage en mode manuel requis
3.. 15	<i>Réservé</i>

7.5.5.1.11 Registre des modes de fonctionnement

Le registre des modes de fonctionnement indique si le robot est en mode manuel (1) ou en mode automatique (0).

Adresse absolue : 40010 Adresse relative : 0x00A	
0	Indique si le robot est en mode manuel

7.5.5.2 Registres de lecture définis par l'utilisateur

Adresse absolue : 40016 Adresse relative : 0x010		...	Adresse absolue : 40255 Adresse relative : 0x0FF	
0 ... 15	Données personnalisées		0 ... 15	Données personnalisées

Utilisez les registres de lecture utilisateur pour recevoir des informations de l'application. La fonction **modbus.read_user_register** de l'application lit les données.

7.5.5.3 Registres de position (articulations 1 à 6)

Adresse absolue : 40256 Adresse relative : 0x100		...	Adresse absolue : 40261 Adresse relative : 0x105	
0 ... 15	Angle de l'articulation 1		0 ... 15	Angle de l'articulation 6

Lisez la valeur angulaire d'une articulation donnée en dixièmes de degré. Si la valeur lue est 101, l'angle de l'axe est de 10,1°. La valeur n'est mise à jour que si l'indicateur de lecture de position est écrit dans le registre des propriétés. Si des valeurs supérieures à 32768 (=215) sont lues, celles-ci correspondent à des valeurs négatives. Soustrayez 32768 pour obtenir la valeur absolue. Une valeur de 32869 correspond par exemple à un angle d'articulation de -10,1°.

7.5.5.4 Registre des coordonnées (x, y, z, roll, pitch, yaw)

Adresse absolue : 40262	...	Adresse absolue : 40267
-------------------------	-----	-------------------------

Adresse relative : 0x106	
0 ... 15	Coordonnée x du TCP

Adresse relative : 0x10B	
0 ... 15	Coordonnée de lacet du TCP

Lisez les coordonnées ou l'orientation du centre de l'outil dans l'ordre x, y, z, roll, pitch, yaw. Les coordonnées spatiales sont lues en dixièmes de millimètre. Si une coordonnée de 314 est lue, la valeur est de 31,4 millimètres. Les coordonnées de rotation sont lues en dixièmes de degré. Si une coordonnée de 271 est lue, la valeur est de 27,1 (mm ou °). La valeur n'est mise à jour que lorsque le drapeau Read-Position est écrit dans le registre des propriétés. Si des valeurs supérieures à 32768 (=215) sont lues, celles-ci correspondent à des valeurs négatives. Soustrayez 32768 pour obtenir la valeur absolue. Une valeur de 33039 correspond par exemple à une valeur de coordonnée de -27,1 (mm ou °).

7.5.6 Affectation des registres d'erreur

Les informations d'erreur apparaissent toujours dans deux registres. Le registre d'erreurs mineures contient le numéro d'erreur proprement dit. Le registre d'erreurs majeures contient l'une des valeurs suivantes, qui sont attribuées aux différentes catégories d'erreurs comme suit :

101	ADDONS
102	CONFIG
103	DATABASE
104	CONTRÔLEUR
105	EDITOR
106	BRIDGE
107	CONNECTION
108	DIALOG
109	DISTRIBUTEUR
110	ÉVÉNEMENT
111	CINÉMATIQUE
112	SOCKET
113	CŒUR
114	SAFETY
115	PARAMÈTRES
116	SIGNAL
117	STATUT
118	SCRIPT
119	THREAD

201	BLOCK
202	COGNEX
203	CONTROL
204	ROS
205	MODBUS
206	REST
207	TCP

301	AX
302	CFG
303	CMK
304	CMD
305	CTL
306	ECAT
307	LIC
308	MATH
309	MCM
310	MISC
311	OS
312	PPM
313	PYAPI
314	SIG
315	SYS
316	TLC
317	UTIL

7.5.6.1 Exemple

Si l'on consulte les registres d'erreurs [113, 4], cela correspond au code d'erreur **CORE_004** généré par le robot.

7.5.7 Exemples de code Python

Ces exemples sont destinés à être utilisés sur un système externe qui se connecte à distance au robot.

7.5.7.1 Exemple 1

```
from pymodbus.client.sync import ModbusTcpClient

def main():
    # connect to client
    client = ModbusTcpClient("192.168.80.203", port=5020)
    client.connect()

    # read all status
    address = 0x000
    count = 4
    status = client.read_input_registers(address, count)
    print("STATUS", status.registers[0], status.registers[1],
          status.registers[2], status.registers[3])

if __name__ == "__main__":
    main()
```

Se connecter au serveur. Lire l'état du robot.

7.5.7.2 Exemple 2

```
from pymodbus.client.sync import ModbusTcpClient

def main():
    # connect to client
    client = ModbusTcpClient("192.168.80.203", port=5020)
    client.connect()

    # command to connect to motors
    address = 0x000
    values = [1 << 0]
    client.write_registers(address, values, unit=1)

if __name__ == "__main__":
    main()
```

Se connecter au serveur. Connecter le robot à ses moteurs.

Remarque : la commande << est utilisée pour le décalage de bits.

Bit	Commande	Entier
0	1 << 0	1
5	1 << 5	32

7.5.7.3 Exemple 3

```
from pymodbus.client.sync import ModbusTcpClient

def main():
    # connect to client
    client = ModbusTcpClient("192.168.80.203", port=5020)
    client.connect()

    # command to run script with id 5
    address = 0x000
    values = [1 << 9, 5]
    client.write_registers(address, values, unit=1)

if __name__ == "__main__":
    main()
```

Se connecter au serveur. Exécuter l'application portant l'ID 5.

7.5.7.4 Exemple 4

```
from pymodbus.client.sync import ModbusTcpClient

def main():
    # connect to client
    client = ModbusTcpClient("192.168.80.203", port=5020)
    client.connect()

    # read all error
    address = 0x004
    count = 2
    error = client.read_input_registers(address, count)
    print("ERROR", error.registers[0], error.registers[1])

    # acknowledge error
    address = 0x002
    values = [1 << 2]
    client.write_registers(address, values, unit=1)

if __name__ == "__main__":
    main()
```

Se connecter au serveur. Lire le message d'erreur actuel. Acquitter le message d'erreur.

8. Module de caméra Cognex

8.1 Présentation

Le module de caméra Cognex vous permet de connecter une caméra Cognex préconfigurée à AX Portal. Une fois la connexion établie, vous pouvez déclencher les tâches préconfigurées sur la caméra afin de capturer et de traiter une image. Les données traitées sont ensuite transférées vers AX Portal, où vous pouvez les lire dans votre application.

8.2 Configuration d'une caméra

Pour configurer une caméra Cognex afin qu'elle soit compatible avec AX Portal, veuillez procéder comme suit :

1. Téléchargez et installez le logiciel [In-Sight Explorer](#) de Cognex.
2. Alimentez la caméra et connectez-la à votre réseau. Vous pouvez également la connecter directement à votre ordinateur, à condition que sa configuration réseau soit réglée sur une adresse IP statique.
3. Lancez In-Sight Explorer, autorisez l'accès si vous y êtes invité et installez les composants requis (par exemple, Microsoft Visual C++, Microsoft .NET Framework).
4. La vue Easybuilder de l'Explorer s'ouvre alors automatiquement.
5. Accédez à « Système » → « Ajouter un capteur/périphérique au réseau ». Recherchez votre caméra et ajoutez-la au réseau.
6. Assurez-vous que les paramètres réseau de la caméra sont compatibles avec ceux du robot de la série AX. Veillez également à ce que le port Telnet soit réglé sur 3000.
7. Sélectionnez votre caméra et cliquez sur le bouton « Connect » pour la connecter à In-Sight Explorer.
8. Vous êtes désormais prêt à créer des tâches (par exemple, pour calibrer la caméra et détecter les objets que le robot doit saisir) et à les enregistrer sur la caméra. Pour plus de détails, veuillez consulter la documentation Cognex.

8.3 Utilisation d'une caméra

Pour utiliser la caméra Cognex dans une application AX Portal, les conditions suivantes doivent être remplies :

- Le robot et la caméra doivent se trouver sur le même réseau,
- le port Telnet de la caméra doit être réglé sur 3000,
- et la caméra doit être mise en ligne.

Voici un exemple illustrant comment utiliser une tâche préconfigurée pour détecter les coordonnées du cercle.

```
# modify these settings to fit to your application
camera_ip = "10.0.0.210"
job_name = "CircleDetection"
event_id = 1
keyword = "coordinates"

# connect the camera to the application
cognex.connect_to_camera(camera_ip)
```

```
# select the job to trigger
cognex.change_job(camera_ip, job_name)
# trigger the event and wait for a response
response = cognex.get_camera_message(camera_ip, event_id, keyword)
print(response)
# close the connection
cognex.disconnect_camera(camera_ip)
```