

EPSON

Portal AX

Manual do utilizador

Versão traduzida

©Seiko Epson Corporation 2026

Rev. 2

PTM265S8842M

Índice

Marcas.....	8
Condições de utilização.....	8
1. Introdução.....	9
1.1 Programação e controlo.....	10
1.2 Interface Gráfica do Utilizador (GUI).....	10
1.3 Módulos de software.....	10
1.3.1 Módulo de ligação TCP.....	10
1.3.2 Módulo de ligação REST.....	11
1.3.3 Módulo de ligação ROS.....	11
1.3.4 Módulo de ligação Modbus.....	11
1.3.5 Interface do painel.....	11
1.3.6 Módulo de câmara Cognex.....	11
2. Interface de administração (GUI).....	12
2.1 Introdução.....	13
2.1.1 Idioma da interface.....	14
2.1.2 Painel de controlo.....	14
2.2 Gestão de utilizadores.....	14
2.2.1 Níveis de autorização.....	14
2.2.2 Página de início de sessão.....	14
2.2.3 Gerir a sua conta.....	15
2.2.4 Gerir outras contas de utilizador.....	16
2.3 Configurações do sistema.....	17
2.3.1 Configurações de rede.....	17
2.3.2 Configurações de hora.....	18
2.3.3 Certificado SSL.....	18
2.4 Atualização de software.....	20
2.5 Ajuda e Suporte.....	22
2.6 Autenticação segura.....	23
2.6.1 Exportar certificado SSL.....	23
2.6.2 Autenticação do utilizador.....	23
2.6.3 Registo da ligação.....	24
3. Exportação por USB e atualização de recuperação.....	25
3.1 Visão geral.....	26
3.2 Exportação para USB.....	26
3.3 Executar a atualização de recuperação.....	26
4. Interface gráfica (GUI).....	27
4.1 Introdução.....	28
4.2 Janela principal.....	29
4.2.1 Visão geral do menu.....	30
4.2.2 Cockpit.....	30
4.2.3 Controlo ativo (Operação exclusiva a partir de uma estação de comando).....	32
4.2.3.1 Controlo ativo.....	32

4.2.3.2 Observador passivo.....	33
4.2.4 Estado do programa.....	33
4.2.5 Modos de funcionamento.....	35
4.2.5.1 Modo automático.....	35
4.2.5.2 Modo manual.....	35
4.2.6 Controlo de precisão.....	36
4.2.7 Controlo de movimento.....	37
4.2.7.1 Limites de movimento.....	37
4.2.7.2 Parâmetros globais.....	38
4.2.7.3 Sistemas de coordenadas globais.....	39
4.3 Blocos de construção e base de dados.....	40
4.4 Menu de Projetos.....	41
4.5 Menu de configuração.....	42
4.5.1 Base de dados (transferência e carregamento).....	42
4.5.2 Configuração de E/S.....	43
4.5.3 Configuração da ferramenta.....	45
4.5.4 Editor de ferramentas.....	45
4.5.4.1 Modo de saída única.....	47
4.5.4.2 Modo de saída dupla.....	47
4.5.4.3 Modo de disparo separado.....	47
4.5.5 Colocação do robô em funcionamento.....	48
4.6 Menu de aprendizagem.....	49
4.6.1 Editor de poses.....	51
4.6.2 Editor de trajetória.....	53
4.6.3 Editor de grelhas.....	55
4.7 Menu de aplicações.....	58
4.7.1 Editor de aplicações.....	59
4.7.2 Código do programa.....	59
4.7.3 Configurações da aplicação.....	61
4.7.4 Variáveis da aplicação.....	63
4.7.5 Variáveis partilhadas e globais.....	65
4.7.6 Editor de pastas.....	66
4.7.7 Programação por blocos.....	67
4.7.7.1 Utilização de variáveis de aplicação e expressões.....	68
4.7.7.2 Bloco de movimento.....	68
4.7.7.3 Bloco de trajetória.....	69
4.7.7.4 Bloco «Levantar».....	69
4.7.7.5 Bloco Colocar.....	70
4.7.7.6 Bloco «Contentor».....	70
4.7.7.7 Bloco Posição seguinte da grelha.....	70
4.7.7.8 Bloco «Redefinir Posição da Grelha».....	70
4.7.7.9 Bloco «Escrever variável».....	70
4.7.7.10 Bloco Definir Saída.....	70
4.7.7.11 Bloco Esperar.....	71

4.7.7.12 Bloco de loop.....	71
4.7.7.13 Bloco «Sair se».....	71
4.7.7.14 Bloco de condição.....	71
4.7.7.15 Bloco de comentário.....	71
4.7.7.16 Bloco «Terminar».....	71
4.7.7.17 Bloco de diálogo.....	71
4.7.7.18 Bloco de saída.....	72
4.7.7.19 Bloco de registo.....	72
4.7.7.20 Bloco de Notificação.....	72
4.7.7.21 Bloco de aviso.....	72
4.7.7.22 Bloco de erro.....	72
4.7.7.23 Bloco de Código Python.....	72
4.8 Menu I/Os.....	72
4.9 Menu de visualização.....	74
4.10 Controlo manual.....	75
4.11 Monitor do sistema.....	76
4.11.1 Estado do robô.....	76
4.11.2 Saída de texto.....	76
4.11.3 Registo da aplicação.....	76
4.11.4 Histórico de mensagens.....	76
4.12 Menu Ajuda.....	77
4.12.1 Informações sobre o robô.....	77
4.12.2 Atalhos de teclado.....	77
4.12.3 Documentação do código.....	77
4.12.4 Licença de código aberto.....	77
4.12.5 Informações de contacto.....	77
4.13 Menu de configurações.....	78
4.13.1 Configurações da GUI.....	79
4.13.2 Personalizar botões do cockpit.....	79
4.13.3 Configurações de inicialização do programa.....	80
4.13.4 Comportamento de movimento.....	80
4.13.4.1 Comportamento em caso de interrupção.....	80
4.13.4.2 Caixa de diálogo de interrupção.....	81
4.14 Menu de configurações de segurança.....	84
4.14.1 Interface do utilizador.....	84
4.14.2 Configurações de segurança.....	86
4.14.2.1 Posição limitada de segurança (SLP).....	86
4.14.2.2 Ângulos de articulação limitados de forma segura.....	87
4.14.2.3 Velocidade limitada de segurança (SLS).....	88
4.14.2.4 Configurações de segurança para potência e força.....	89
4.14.2.5 E/S de segurança e restrições do operador.....	90
4.14.2.6 Dados de verificação.....	91
4.14.2.7 Funções das saídas de segurança.....	91
4.14.2.8 Ligação de dispositivos opcionais.....	91

5. Interface do painel (GUI).....	92
5.1 Introdução.....	93
5.2 Configurar aplicações para a interface do painel.....	94
5.3 Cockpit e configurações.....	96
5.4 Controlar aplicações.....	97
6. Definições Cinemática.....	98
6.1 Coordenadas.....	99
6.1.1 Ângulos náuticos.....	99
6.1.2 Sistemas de referência.....	100
6.2 Configuração.....	101
6.3 Posição.....	101
6.4 Ponto.....	101
6.5 Trajetória.....	101
6.6 Segmento.....	102
6.7 Grelhas.....	102
7. Módulos de ligação.....	104
7.1 Ligações, autenticação e controlo ativo.....	105
7.1.1 Canais de ligação.....	105
7.1.2 Autenticação.....	105
7.1.3 Controlo ativo.....	106
7.2 Módulo de ligação TCP.....	106
7.2.1 Estabelecer ligação.....	106
7.2.2 Enviar comandos.....	108
7.2.3 Configurações.....	109
7.2.3.1 Padrão de codificação.....	109
7.2.3.2 Caractere de fim.....	109
7.2.3.3 Enviar heartbeat.....	109
7.2.4 Protocolo de mensagens – Receber comandos.....	110
7.2.4.1 Informações de estado.....	110
7.2.4.2 Valor de retorno da ação.....	111
7.2.4.3 Erros.....	111
7.2.4.4 Histórico de mensagens.....	111
7.2.4.5 Definição de um grupo de estados.....	111
7.2.4.6 Dados de um grupo de estado.....	112
7.2.4.7 Abrir caixa de diálogo.....	113
7.2.4.8 Enviar a partir da aplicação.....	113
7.2.5 Registo de mensagens – Enviar comandos.....	114
7.2.5.1 Login com token.....	114
7.2.5.2 Solicitar controlo ativo.....	114
7.2.5.3 Liberar o controlo ativo.....	115
7.2.5.4 Obter dados.....	115
7.2.5.5 Posição de funcionamento.....	116
7.2.5.6 Desligar.....	116
7.2.5.7 Pausar.....	116

7.2.5.8 Continuar.....	117
7.2.5.9 Ativação do controlo manual (HGC).....	117
7.2.5.10 Desativar o controlo manual (HGC).....	118
7.2.5.11 Parar.....	118
7.2.5.12 Mudar de projeto.....	119
7.2.5.13 Executar aplicação.....	119
7.2.5.14 Testar a aplicação.....	119
7.2.5.15 Receber mensagem.....	120
7.2.5.16 Deslocar para a pose.....	120
7.2.5.17 Reproduzir percurso.....	122
7.2.5.18 Alternar ferramenta.....	122
7.2.5.19 Gravação de saídas digitais e analógicas.....	123
7.2.5.20 Responder a uma caixa de diálogo.....	124
7.2.5.21 Definir variável de estado.....	124
7.2.5.22 Definir variável global.....	125
7.2.5.23 Ler variável global.....	125
7.2.5.24 Definir variável partilhada.....	126
7.2.5.25 Ler variável partilhada.....	126
7.3 Módulo de ligação REST.....	127
7.3.1 Estabelecer ligação.....	127
7.3.2 Enviar comandos.....	128
7.3.3 Configurações.....	128
7.4 Módulo de ligação ROS.....	129
7.4.1 Instalação.....	129
7.4.2 Configuração.....	130
7.4.3 Detecção de rede com o Cyclone DDS.....	130
7.4.3.1 Interface de rede.....	130
7.4.3.2 Métodos de deteção.....	131
7.4.4 Estabelecer ligação.....	133
7.4.5 Acesso aos dados do robô através do ROS.....	134
7.4.6 Funcionalidades do ROS 2 em scripts.....	134
7.4.7 Chamar uma ação.....	134
7.4.8 Configurações de autenticação.....	135
7.5 Módulo de ligação Modbus.....	135
7.5.1 Visão geral.....	135
7.5.2 Configurações.....	136
7.5.3 Registos de escrita (descrição simples).....	137
7.5.3.1 Registo de comandos.....	137
7.5.3.2 Registo de propriedades.....	138
7.5.3.3 Registos de escrita definidos pelo utilizador.....	138
7.5.4 Registos de escrita (descrição detalhada).....	139
7.5.4.1 Não fazer nada.....	139
7.5.4.2 Colocação em funcionamento.....	139
7.5.4.3 Desligar.....	139

7.5.4.4 Parar.....	139
7.5.4.5 Pausar.....	139
7.5.4.6 Continuar.....	140
7.5.4.7 Calibração.....	140
7.5.4.8 Ativar o controlo manual (HGC).....	140
7.5.4.9 Desativar o modo de operação manual (HGC).....	140
7.5.4.10 Mudar de projeto.....	140
7.5.4.11 Iniciar a aplicação.....	141
7.5.4.12 Deslocar para a pose.....	141
7.5.4.13 Reproduzir percurso.....	141
7.5.4.14 Colocar ferramenta.....	141
7.5.4.15 Levantar a ferramenta.....	141
7.5.4.16 Gravação de saídas digitais.....	142
7.5.4.17 Definir eventos de aplicação.....	142
7.5.4.18 Confirmar erros.....	142
7.5.4.19 Leitura da posição.....	142
7.5.4.20 Solicitar controlo ativo.....	142
7.5.4.21 Liberar o controlo ativo.....	143
7.5.4.22 Mudar para o modo manual.....	143
7.5.4.23 Mudar para o modo automático.....	143
7.5.4.24 Reinicialização de segurança.....	143
7.5.4.25 Iniciar interrupção de movimento.....	144
7.5.5 Registo de leitura.....	145
7.5.5.1 Registo de estado.....	145
7.5.5.2 Registos de leitura definidos pelo utilizador.....	148
7.5.5.3 Registos de posição (articulações 1 a 6).....	148
7.5.5.4 Registo de coordenadas (x, y, z, roll, pitch, yaw).....	148
7.5.6 Atribuição dos registos de erros.....	149
7.5.6.1 Exemplo.....	150
7.5.7 Exemplos de código Python.....	151
7.5.7.1 Exemplo 1.....	151
7.5.7.2 Exemplo 2.....	151
7.5.7.3 Exemplo 3.....	152
7.5.7.4 Exemplo 4.....	152
8. Módulo de câmara Cognex.....	153
8.1 Visão geral.....	154
8.2 Configurar uma câmara.....	154
8.3 Utilização de uma câmara.....	154

Marcas

Cognex® e In-Sight Explorer® são marcas comerciais ou marcas registradas da Cognex Corporation nos Estados Unidos e noutros países.

Google Chrome™ e Chrome™ são marcas comerciais da Google LLC.

Linux® é uma marca registrada de Linus Torvalds nos EUA e noutros países.

Apple®, macOS® e Safari® são marcas registradas da Apple Inc. nos EUA e noutros países e regiões.

Microsoft®, Windows®, Microsoft Edge® e Microsoft Visual C++® são marcas (ou marcas registradas) da Microsoft Corporation e/ou das suas subsidiárias nos Estados Unidos e noutros países.

Mozilla e Firefox® são marcas da Mozilla Foundation nos EUA e noutros países.

Nota geral: Os nomes de produtos aqui utilizados servem apenas para fins de identificação e podem ser marcas dos respetivos proprietários. A Epson não reivindica quaisquer direitos sobre estas marcas.

Todas as capturas de ecrã servem exclusivamente para fins ilustrativos e de formação. Não se pretende nem se implica qualquer ligação a um titular de marca, nem qualquer aprovação por parte deste.

Condições de utilização

Nenhuma parte deste manual de instruções pode ser reproduzida ou reimpressa sob qualquer forma sem autorização expressa por escrito.

As informações contidas neste documento estão sujeitas a alterações sem aviso prévio.

Entre em contacto connosco caso detete erros neste documento ou tenha dúvidas sobre as informações nele contidas.

1. Introdução

1.1 Programação e controlo

A estrutura de software AX Portal oferece ao utilizador de um robô da série Epson AX uma forma muito intuitiva de comunicar com este robô. Com o AX Portal, o utilizador tem acesso total para manipular e controlar o robô, bem como para escrever aplicações de programa com a linguagem de programação [Python 3.13](#). A funcionalidade básica para a programação com Python está disponível online. No entanto, alguns exemplos de código muito básicos também são fornecidos na documentação de código integrada.

Para além de uma vasta seleção de funções Python básicas, as aplicações do AX Portal suportam uma coleção abrangente de funções desenvolvidas especificamente para o controlo, manutenção e comunicação com robôs AX. A lista completa das funções de script suportadas pelo AX Portal encontra-se na documentação de código integrada.

1.2 Interface Gráfica do Utilizador (GUI)

A GUI foi concebida como uma aplicação web que pode ser executada em navegadores web comuns em praticamente qualquer dispositivo. Os seguintes navegadores são oficialmente suportados:

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

Por conseguinte, não é necessário descarregar um módulo de software específico para o seu controlador, desde que o robô e o dispositivo se encontrem na mesma rede ou partilhem um canal de comunicação direto. Para além da interface de ambiente de trabalho padrão do AX Portal, que está otimizada principalmente para a programação em computadores de secretária e portáteis, o AX Portal oferece uma interface de painel que contém apenas as funções necessárias para operar um robô já pré-programado. Encontrará mais informações sobre as GUIs e todas as funções suportadas mais adiante neste documento.

1.3 Módulos de software

Para além da funcionalidade básica do AX Portal, descrita em pormenor neste documento, estão disponíveis nos robôs AX alguns módulos de software integrados com opções de controlo adicionais. Tenha em atenção que todos os módulos de ligação abaixo referidos oferecem a possibilidade de se ligar remotamente ao robô AX, proporcionando assim oportunidades para controlar e monitorizar o robô.

1.3.1 Módulo de ligação TCP

O módulo de ligação TCP pode ser utilizado como alternativa para a comunicação com o robô. O robô é normalmente controlado pelo utilizador através da GUI no navegador da Web. No entanto, caso o robô deva ser controlado através de uma interface definida pelo utilizador ou de outra máquina, este aceita comandos TCP diretamente. Com o módulo de ligação TCP, tudo o que está disponível através da comunicação do navegador pode também ser acedido via TCP.

1.3.2 Módulo de ligação REST

O módulo de ligação REST pode ser utilizado como alternativa para a comunicação com o robô. Normalmente, o robô é controlado pelo utilizador através da GUI no navegador da Web. No entanto, caso o robô deva ser controlado através de uma interface definida pelo utilizador ou de outra máquina, ele aceita comandos REST diretamente. Com o módulo de ligação REST, tudo o que está disponível através da comunicação do navegador também pode ser acedido através de REST.

1.3.3 Módulo de ligação ROS

O módulo de ligação ROS oferece funcionalidades para a integração do robô num ambiente ROS2. O robô permite que outros clientes na rede ROS2 enviem comandos para controlar e monitorizar o robô.

1.3.4 Módulo de ligação Modbus

O módulo de ligação Modbus pode ser utilizado como alternativa para a comunicação com o robô. Este utiliza o protocolo Modbus-TCP e permite ao utilizador ler e escrever registos Modbus padrão. Utilize este módulo para controlar o robô diretamente através de um PLC. As funções mais importantes, tais como a colocação em funcionamento do robô, a execução de aplicações e o acionamento de saídas, podem ser realizadas através do módulo Modbus.

1.3.5 Interface do painel

A interface do painel oferece uma interface gráfica adicional com funcionalidades limitadas, concebida mais para operadores de robôs do que para programadores. Esta interface destina-se principalmente a iniciar e monitorizar aplicações, e o seu conteúdo pode ser adaptado pelo programador ao seu caso de utilização. A interface do painel está otimizada para dispositivos com ecrã tátil e o seu design assemelha-se à estrutura familiar de aplicações dos dispositivos móveis modernos, tornando-a intuitiva mesmo para utilizadores sem conhecimentos técnicos.

1.3.6 Módulo de câmara Cognex

O módulo de câmara Cognex fornece uma interface de comunicação entre uma câmara Cognex e o AX Portal. As câmaras são normalmente integradas num efector final ou montadas de forma fixa para observar a cena em que o robô opera. Este módulo é ideal para uma inspeção visual personalizável e suporta todo o conjunto de funcionalidades das câmaras Cognex.

2. Interface de administração (GUI)

2.1 Introdução

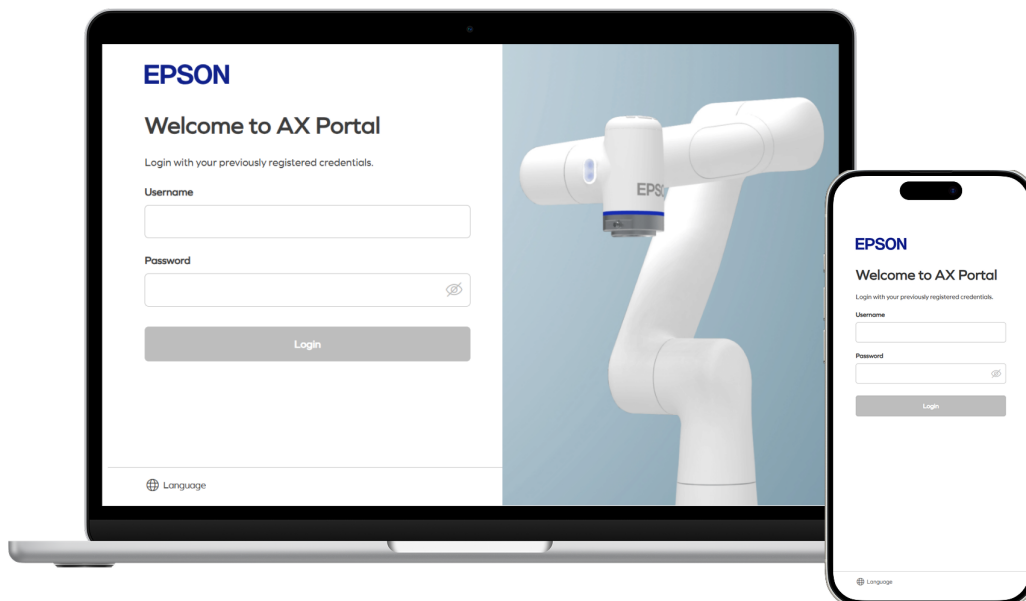


Figura 2.1.a: Exemplos de capturas de ecrã da interface de administração

A interface de administração do AX Portal é uma interface gráfica do utilizador (GUI) através da qual os utilizadores podem gerir contas de utilizador e alterar algumas configurações básicas associadas ao sistema operativo do robô (tais como configurações de rede, configurações de hora, etc.). Estas configurações são normalmente definidas apenas uma vez durante a instalação do robô, antes mesmo de um programador começar a escrever aplicações de controlo do robô. É possível aceder a esta interface através de um navegador da Web (por exemplo, o Google Chrome), introduzindo o endereço IP do robô na barra de endereços do navegador, o que o redirecionará para <https://<IP-ADDRESS>/admin> (consulte o «Manual do Controlo do Robô RC-A101» para obter o endereço IP correto).

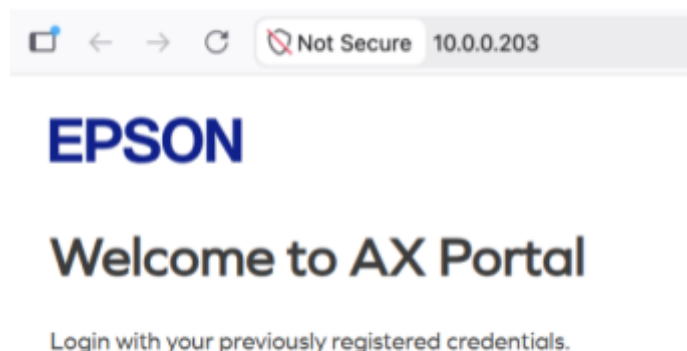


Figura 2.1.b: Ligação à interface de administração através do URL

Neste capítulo, todos os menus e elementos da interface de administração são explicados em detalhe.

2.1.1 Idioma da interface

Para alterar o idioma da interface, clique no menu suspenso de idiomas no canto inferior esquerdo da interface e selecione o seu idioma preferido. O seu navegador facilita-lhe o trabalho, guardando a sua configuração de idioma ao longo das sessões e interfaces.

2.1.2 Painel de controlo

Após o início de sessão, o painel de controlo da interface de administração permite a navegação para os vários submenus, para outras interfaces do AX Portal, bem como o fim da sessão. Tenha em atenção que o painel de controlo apenas apresenta as opções às quais tem acesso atualmente, dependendo do seu nível de utilizador (administrador, responsável pela segurança, programador, operador). Se não conseguir navegar para outras interfaces, o AX Portal não está a funcionar corretamente, o que acontece, por exemplo, durante a execução de uma atualização de software.

2.2 Gestão de utilizadores

2.2.1 Níveis de autorização

Para impedir que o robô seja configurado, programado ou operado por utilizadores não autorizados, são distinguidos vários níveis de autorização. A autorização é definida exclusivamente pelos administradores para cada utilizador. Estão disponíveis os seguintes níveis de autorização, da autorização mais elevada à mais baixa:

- Um **administrador** tem acesso total a todas as interfaces e funções de controlo do robô, tem acesso à gestão de outros utilizadores, bem como às principais configurações do sistema operativo.
- Um **programador** tem acesso a todas as interfaces e funções de controlo do robô e pode preparar e programar aplicações que serão posteriormente executadas pelos operadores. Dentro da interface de administração, o programador tem, essencialmente, apenas acesso aos dados da sua própria conta e à navegação entre as interfaces.
- Um **responsável pela segurança** é um programador (ver acima) que tem permissão para alterar as configurações de segurança do robô. Os outros utilizadores apenas podem visualizar as configurações de segurança.
- Um **operador** tem acesso apenas às interfaces básicas de operação que permitem o manuseamento do robô, mas não pode alterar configurações ou programas. Na interface de administração, o operador tem acesso apenas aos dados da sua própria conta.

2.2.2 Página de início de sessão

A página de início de sessão contém dois formulários diferentes, dependendo de já ter sido criada uma conta de administrador.

- Se já tiver sido criado um administrador, aparece a página de início de sessão habitual, na qual terá de introduzir o seu nome de utilizador e palavra-passe para iniciar a sessão. As palavras-passe são ocultadas durante a introdução, a menos que clique no botão para visualizar a palavra-passe no campo da palavra-passe.
- Se ainda não tiver sido criado nenhum administrador, ser-lhe-á solicitado diretamente que o faça, e só poderá prosseguir depois de ter criado uma conta de administrador. Encontrará mais

detalhes no «Robô colaborativo de 6 eixos AX6: Guia rápido / Manual de instalação». Aqui, só pode definir os dados absolutamente necessários; se pretender visualizar e definir todas as configurações da sua conta, aceda ao menu «Manage My Account».

Após cada saída, será redirecionado para a página de início de sessão.

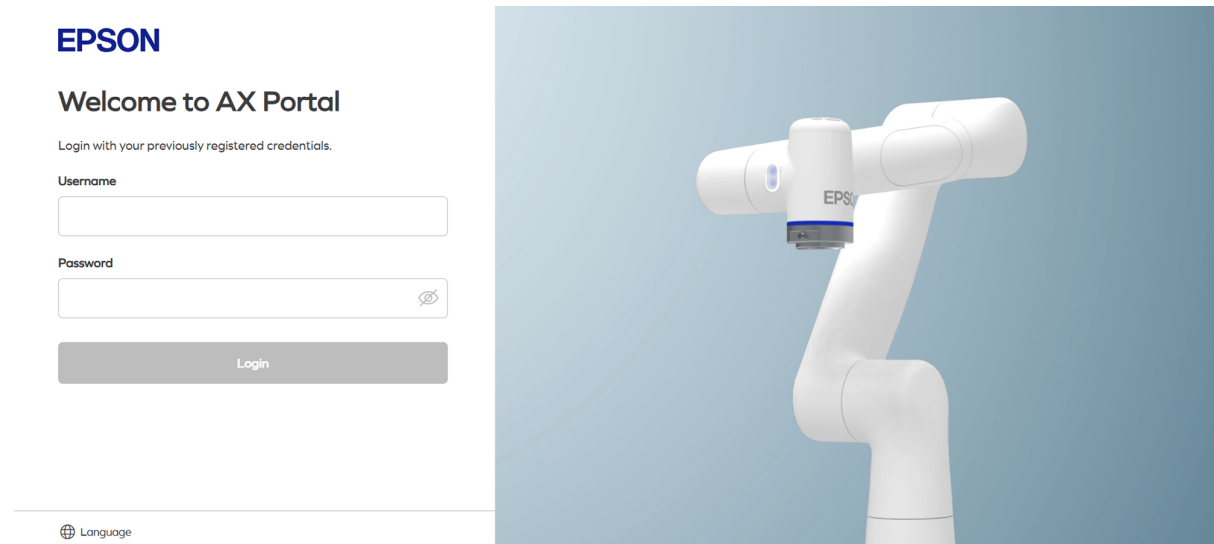


Figura 2.2.2: Página de início de sessão

2.2.3 Gerir a sua conta

No menu «Manage My Account», pode alterar o seu próprio perfil, mais concretamente o seu nome, a sua palavra-passe, a sua imagem de perfil e a página inicial:

- «Nome completo» é o nome utilizado para se dirigir ao utilizador na interface e nas mensagens. Trata-se de um campo obrigatório destinado ao seu próprio nome e apelido; no entanto, também pode definir as suas próprias convenções de nomenclatura.
- O campo «Nome de utilizador» não pode ser alterado após a criação de um utilizador; o administrador define o nome de utilizador durante a criação do utilizador.
- O campo «Interface de Início» determina para qual interface deseja ser redirecionado após o início de sessão. Para diferentes níveis de autorização, existem diferentes interfaces de início padrão predefinidas, que pode alterar aqui.
- Através do botão «Alterar palavra-passe», pode alterar a palavra-passe da sua conta. Ser-lhe-á solicitado que volte a introduzir a sua palavra-passe atual e que introduza a sua nova palavra-passe duas vezes, para minimizar erros de introdução acidentais na definição da palavra-passe. A sua palavra-passe também pode ser alterada por qualquer administrador, caso a esqueça. As senhas nunca são armazenadas em texto simples; por isso, nem o próprio utilizador nem qualquer outro utilizador poderá ler a sua senha.
- A «Imagem de perfil» pode ser alterada opcionalmente. Será exibido um ícone de utilizador padrão, caso não defina uma imagem de perfil.

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

Figura 2.2.3: Página «Gerir a sua conta»

2.2.4 Gerir outras contas de utilizador

No menu «Outras contas de utilizador», poderá ver uma visão geral de todos os utilizadores atualmente disponíveis. Aqui, tem as seguintes opções:

- A adição de um novo utilizador permite-lhe criar um utilizador com um nível de autorização e uma palavra-passe inicial predefinidos. A palavra-passe inicial deve ser alterada por cada utilizador após o primeiro início de sessão.
- A alteração de um utilizador permite-lhe modificar a maioria das propriedades desse utilizador, com exceção do nome de utilizador, que não pode ser alterado.
- A eliminação de um utilizador remove-o permanentemente da base de dados. Este utilizador deixará de poder acionar quaisquer funções de controlo do robô.

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

Figura 2.2.4: Página «Gerir outras contas de utilizador»

2.3 Configurações do sistema

2.3.1 Configurações de rede

O menu «Configurações de rede» contém todas as configurações relevantes para definir a ligação Ethernet do robô. Tenha em atenção que perderá a ligação assim que alterar as configurações da ligação atualmente ativa (por exemplo, se estiver ligado via Ethernet e alterar o endereço IP Ethernet). Caso introduza acidentalmente configurações inválidas, poderá não conseguir estabelecer novamente a ligação com o robô. Por isso, recomenda-se vivamente que se certifique de que duas ligações diferentes estão a funcionar antes de alterar as configurações da única ligação que está a funcionar.

As definições de Ethernet mostram e permitem-lhe decidir se o robô utiliza um endereço IP dinâmico ou manual. Caso o robô esteja configurado para obter um endereço IP dinâmico, poderá também ver aqui o endereço IP selecionado. Caso o robô esteja configurado manualmente, terá de introduzir um endereço IP e a máscara de rede (campo obrigatório); além disso, pode definir o gateway e até dois servidores DNS (opcional).

The screenshot displays the 'EPSON Admin Interface' with a sidebar menu on the left containing options like '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. Under the 'Ethernet IPv4' section, the 'IP Address' is set to '192.168.90.200'. The 'Dynamic IP Address' option is selected with a blue radio button, while the 'Manual IP Address' option is unselected with a grey radio button. Below this, there 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 at the bottom provides a note about using the second Ethernet port for network maintenance. At the very bottom, there are 'Cancel' and 'Save' buttons.

Figura 2.3.1: Página «Configurações de rede»

2.3.2 Configurações de hora

O menu «Configurações de hora» exibe a data atual, a hora atual e o fuso horário do sistema operativo do robô. Certifique-se de que a hora e a data inseridas estão corretas. Ao definir a hora correta, o sistema pode ser utilizado para a análise de registos e a detecção de erros, e as dependências temporais funcionam corretamente. Se alterar a hora manualmente, a hora exata será aplicada imediatamente após pressionar o botão «Guardar». Se pretender alterar o fuso horário, pode filtrar a lista de fusos horários disponíveis no sistema por continente (por exemplo, «Europe»), código de país (por exemplo, «GB»), cidade de grande dimensão (por exemplo, «London») ou uma combinação destes (por exemplo, «eur lon» → Europe/London).

Nota: Se o robô tiver acesso à rede, pode ativar a sincronização NTP (Network Time Protocol).

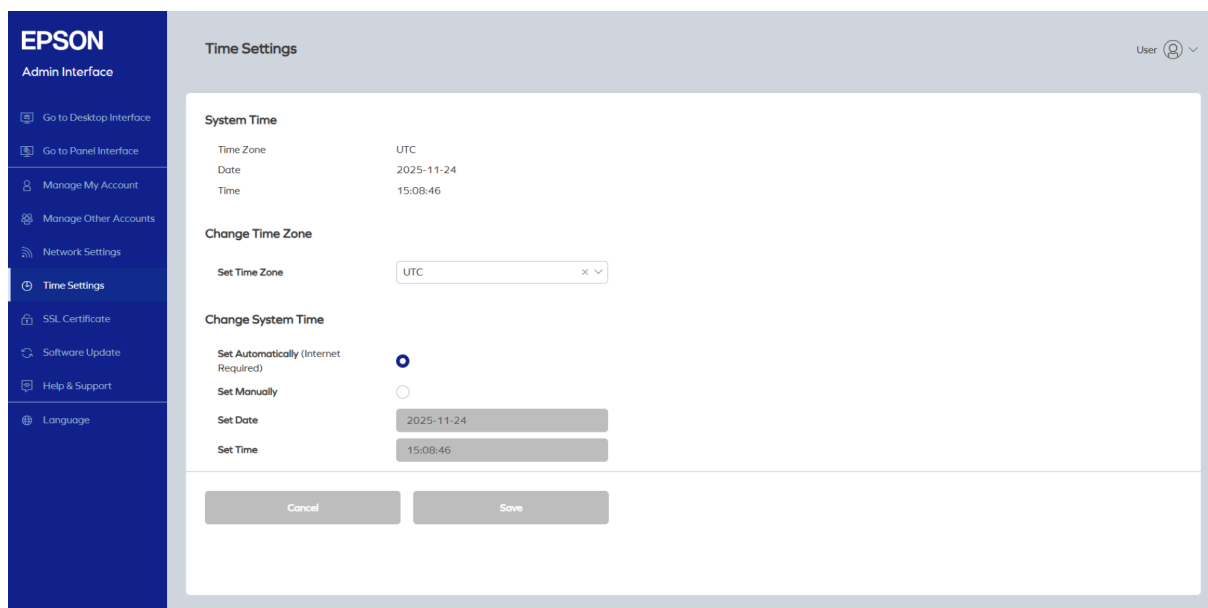


Figura 2.3.2: Página «Definições de hora»

2.3.3 Certificado SSL

O menu «Certificado SSL» apresenta alguns detalhes sobre o certificado SSL atualmente carregado e permite-lhe substituir este certificado pelo seu próprio certificado privado ou empresarial. Um certificado SSL (Secure Sockets Layer) é um certificado digital que autentica a identidade de um site e permite uma ligação encriptada. Este certificado garante que as transações online e as informações dos clientes permaneçam privadas e seguras no seu navegador da Web. Por predefinição, está instalado um certificado autoassinado, que deve ser substituído por um certificado adequado ao ambiente do utilizador e assinado por uma entidade de confiança.

Um certificado é marcado como inseguro pelos navegadores se

- o nome de domínio (ou endereço IP) registado no certificado não corresponder ao domínio (ou endereço IP) utilizado para aceder ao robô,
- o certificado tiver expirado ou
- o certificado for autoassinado.

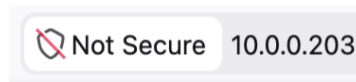


Figura 2.3.3.a: Exemplo de um navegador que sinaliza um certificado SSL inseguro ou ausente

Para substituir o certificado SSL, terá de carregar tanto a chave privada como a chave pública.

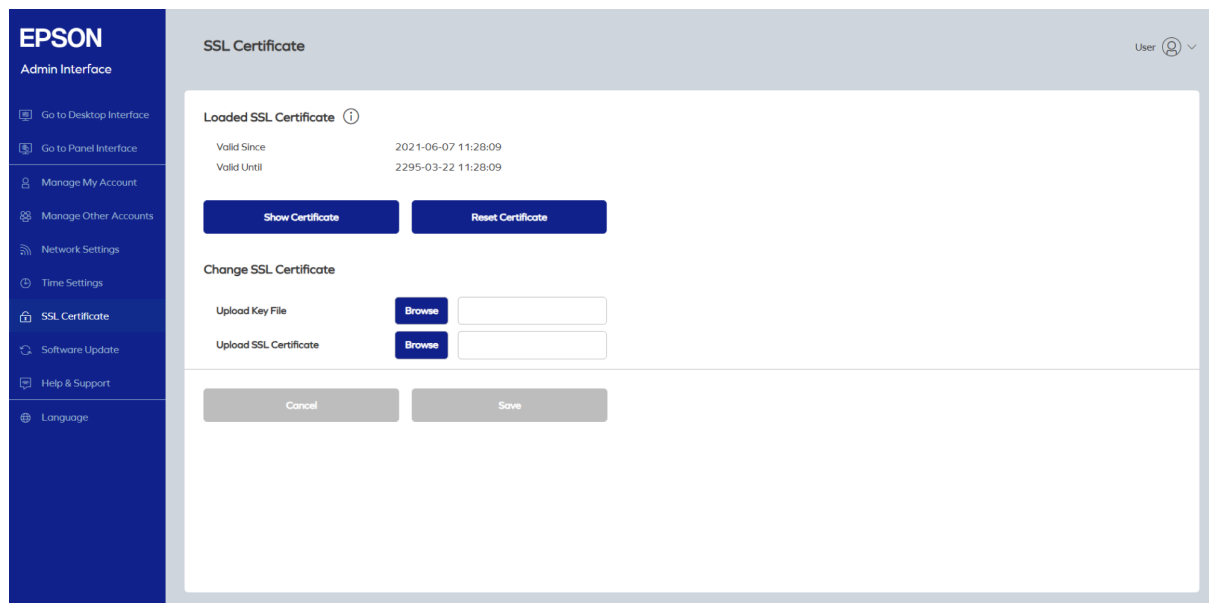


Figura 2.3.3.b: Página «Certificado SSL»

2.4 Atualização de software

No menu «Atualização de software», pode ver a versão atual do AX Portal. Aqui, pode carregar e instalar novas versões de software. Uma atualização de software pode ser necessária para resolver um problema específico ou para obter as funcionalidades mais recentes e correções de erros.

Antes de atualizar a versão da software, leia sempre primeiro as Notas de Release, pois estas contêm informações importantes sobre possíveis riscos de uma atualização para determinadas versões. As melhorias ocorrem, por vezes, em detrimento de funcionalidades mais antigas, que são classificadas como obsoletas. Recomendamos que crie uma cópia de segurança da sua base de dados, que inclua todas as aplicações e definições dos robôs (através da interface do ambiente de trabalho), antes de iniciar a atualização.

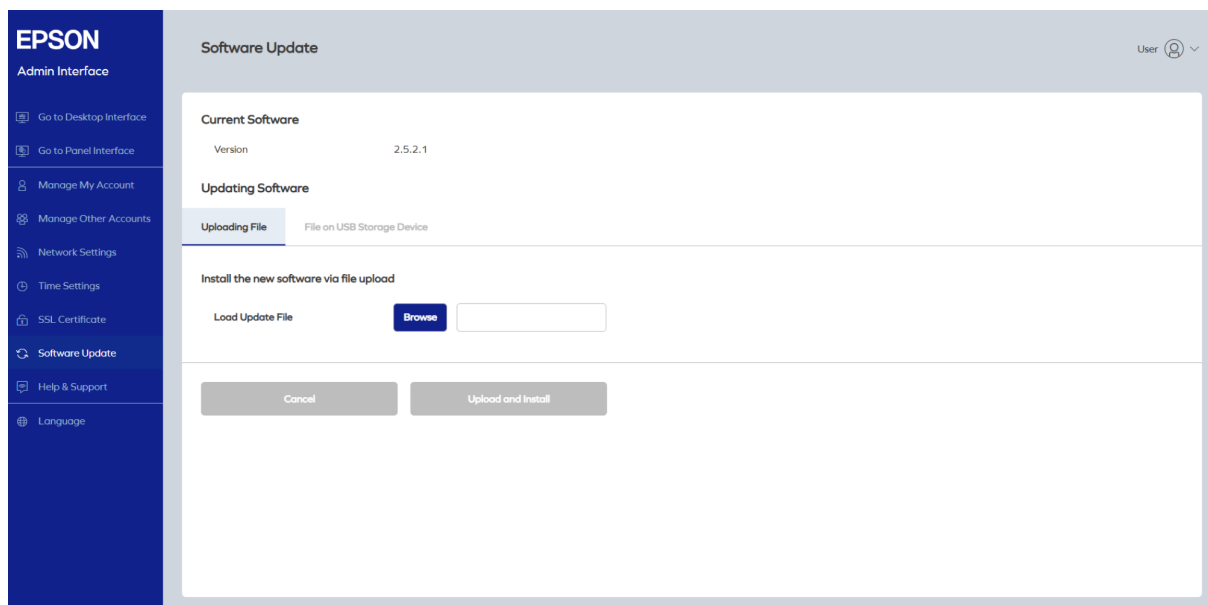


Figura 2.4.a: Página «Atualização de software»

Para instalar uma atualização de software, deve primeiro obter o ficheiro RBIN que contém a atualização para a versão do AX Portal que pretende instalar. Existem três formas de instalar esta atualização:

- 1) Selecione o primeiro separador «Carregar ficheiro». Aqui pode carregar a atualização a partir do seu dispositivo. Navegue pelo sistema de ficheiros e selecione o ficheiro RBIN da atualização; em seguida, prossiga com a instalação da atualização.
- 2) Selecione o segundo separador «Ficheiro do suporte USB». Aqui, pode colocar o ficheiro RBIN numa pen USB e ligá-la ao controlador do robô. Selecione o ficheiro RBIN de atualização no diretório da pen USB e, em seguida, continue com a instalação.
Nota: Atualmente, os sistemas de ficheiros exFAT e FAT32 são suportados para a pen USB.
- 3) Caso as opções 1) e 2) não sejam aplicáveis (por exemplo, se a interface de administração no seu robô não estiver a funcionar corretamente), tem sempre a possibilidade de executar a atualização de recuperação em vez disso.

Enquanto uma atualização está a decorrer, poderá ver qual a etapa que está a ser executada, quais as etapas concluídas com sucesso e – em caso de erro – em que etapa ocorreu o erro, juntamente com a mensagem de erro. Uma atualização pode demorar **até 15 minutos**. Por isso, pedimos a sua paciência e que não interrompa a atualização, por exemplo, desligando a alimentação do robô. Na pior das hipóteses, isso pode fazer com que, posteriormente, já não consiga aceder ao robô.

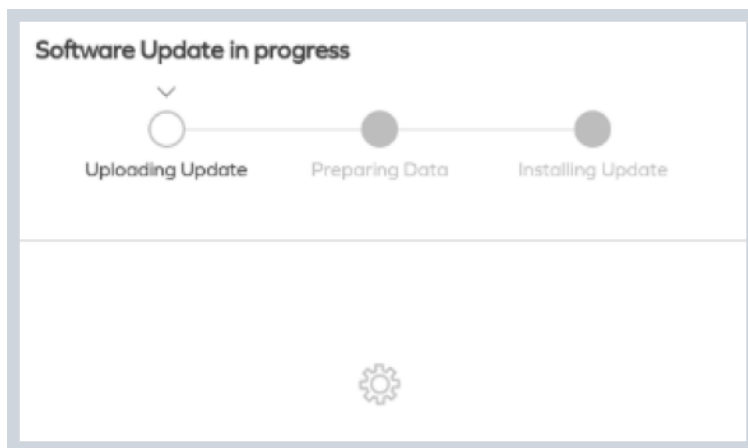


Figura 2.4.b: Janela «Atualização de software em curso»

Após a conclusão bem-sucedida de uma atualização, esta ainda tem de ser confirmada. Até que a atualização seja confirmada, isso é indicado por uma mensagem vermelha. Um reinício do controlador reverte a atualização, caso esta não tenha sido confirmada. Assim que for confirmada, já não é possível reverter para uma versão anterior.

2.5 Ajuda e Suporte

O menu «Ajuda e Suporte» apresenta as informações de contacto que pode utilizar para contactar a Epson, caso necessite de ajuda. Por favor, consulte primeiro a documentação antes de solicitar assistência por e-mail ou telefone.

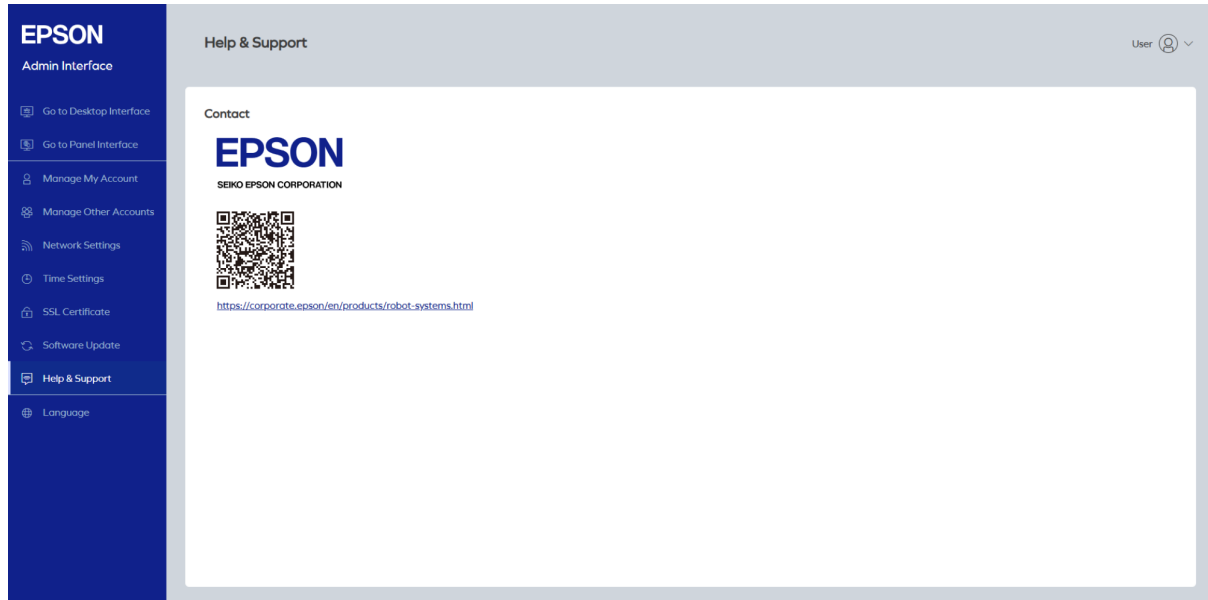


Figura 2.5: Página «Ajuda e Suporte»

2.6 Autenticação segura

Se utilizar módulos de ligação para aceder ao AX Portal e controlar o robô (sem GUI), terá de se autenticar através de um procedimento seguro de duas etapas antes de poder efetivamente controlar o robô.

2.6.1 Exportar certificado SSL

No primeiro passo, deve enviar um pedido de início de sessão ao servidor de autenticação integrado. Se pretender descarregar o certificado SSL atualmente carregado, pode fazê-lo exportando o certificado através do seu navegador. A ilustração seguinte mostra como isto funciona com o Google Chrome. Outros navegadores suportam funcionalidades semelhantes.

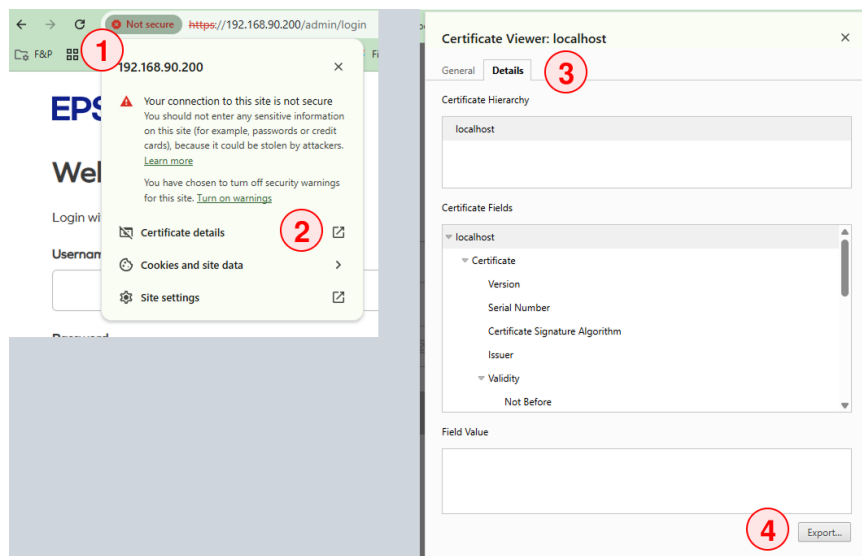


Figura 2.6.1: Exportação do certificado SSL

1. Exibir informações, 2. Aceder aos detalhes do certificado, 3. No visualizador de certificados, aceder a «Detalhes», 4. Exportar o certificado

2.6.2 Autenticação do utilizador

No primeiro passo, deve enviar um pedido de início de sessão para o servidor de autenticação integrado, acessível através de **TCP** na **porta 5001**. Para tal, necessita de conhecer o endereço IP do robô, as suas credenciais de login (nome de utilizador e palavra-passe) e utilizar o mesmo certificado SSL que o robô. Aqui está um exemplo em Python que envia a solicitação de login através de um socket TCP e devolve o token de acesso. Também pode utilizar outras linguagens de programação para o fazer.

Nota: Atualmente, o início de sessão só é possível através de TCP. Se pretende controlar o seu robô através de um dos outros módulos de ligação que não seja o TCP (por exemplo, ROS/REST/Modbus), terá, mesmo assim, de utilizar o TCP para obter o token de acesso. Em alternativa, pode desativar a autenticação para estes módulos de ligação nas definições do AX Portal. Atenção: Se desativar a autenticação, qualquer cliente poderá controlar o robô através de ROS/REST sem autenticação prévia!

```
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 Registo da ligação

Assim que tiver um token de acesso válido, pode utilizá-lo na função «token_login», cuja aplicação é descrita em pormenor no capítulo dedicado ao respetivo módulo de ligação.

3. Exportação por USB e atualização de recuperação

3.1 Visão geral

Se não for possível aceder à interface de administração, apesar de o controlador estar em funcionamento, pode tentar restaurar o controlador utilizando o método de atualização de recuperação.

3.2 Exportação para USB

O registo do sistema e a base de dados AXPortal podem ser exportados sem acesso à interface web, utilizando uma pen USB:

1. Formatar uma pen USB com exFAT, FAT32 ou NTFS e certifique-se de que NÃO tem a designação de sistema de ficheiros LOG ou RECOVERY
2. Insira-a no controlador e aguarde alguns minutos.
3. Retire a pen USB. Deverá encontrar, no diretório raiz da pen, um ficheiro ZIP que contém a base de dados e os registos de sistema relevantes.

3.3 Executar a atualização de recuperação

Para executar a atualização do software de recuperação, proceda da seguinte forma:

1. Obtenha o ficheiro de atualização, que normalmente está disponível no formato RBIN. Caso não possua um ficheiro de atualização, contacte o fabricante.
2. Certifique-se de que tem à mão uma pen USB que possa utilizar para a atualização. Esta deve estar formatada em exFAT, FAT32 ou NTFS, e o sistema de ficheiros da primeira partição deve estar identificado com «RECOVERY».
3. Carregue o ficheiro de atualização para o diretório raiz da pen USB.
4. Desligue o robô e reinicie-o. Aguarde pelo menos 1 minuto até que o robô tenha arrancado corretamente.
5. Insira a pen USB preparada numa qualquer porta USB do controlador.
6. Aguarde até que a atualização esteja concluída. Tenha em atenção que este processo pode demorar **até 15 minutos**. Tente aceder à interface de administração e verifique se a nova versão foi instalada.
7. **Importante:** Retire a pen USB do controlador. Se se esquecer deste passo, a atualização será executada novamente após o reinício, o que não é desejável.

4. Interface gráfica (GUI)

4.1 Introdução

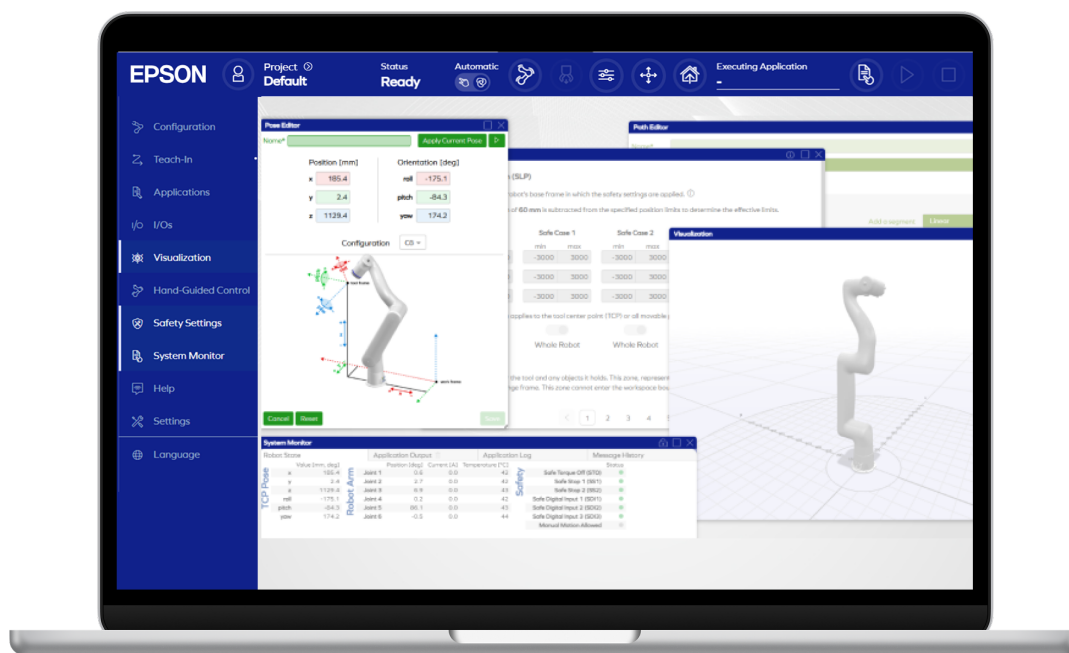


Figura 4.1.a: Exemplo de captura de ecrã da interface de secretária

A interface de desktop do AX Portal é a principal interface gráfica do utilizador (GUI) através da qual o utilizador controla o robô, programa aplicações, edita elementos da base de dados e visualiza o estado atual do robô. Esta GUI foi concebida para ser intuitiva e, em grande medida, autoexplicativa, mesmo para utilizadores sem grande experiência em controlo de robôs. Pode aceder a esta interface através da interface de administração, iniciando sessão e utilizando o botão «Mudar para a interface do ambiente de trabalho».

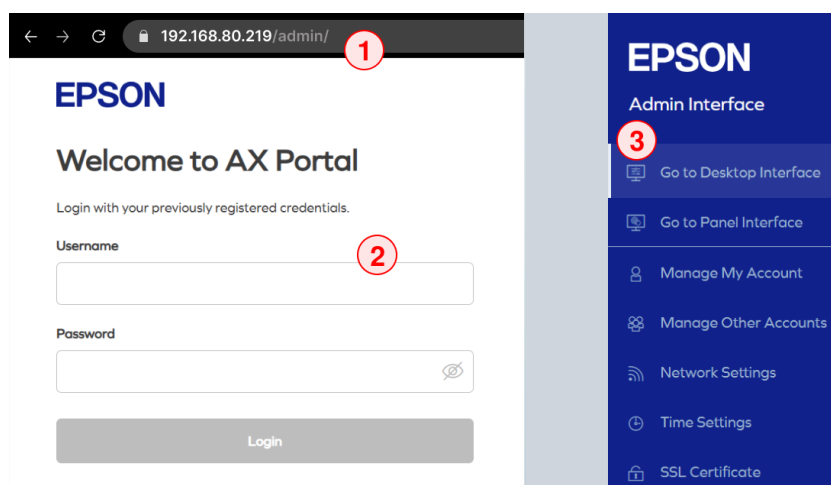


Figura 4.1.b: Acesso à interface de secretária

1. Introduzir o endereço IP na URL, 2. Introduzir os dados de início de sessão, 3. Mudar para a interface de secretária

A interface de desktop contém todas as funções de controlo disponíveis para o robô, bem como funções adicionais para observar e alterar o seu estado. Para executar facilmente aplicações programadas sem ter de alterar aplicações ou configurações, recomendamos a utilização da interface do painel. Neste capítulo, todas as janelas, menus e elementos da interface de desktop são explicados em pormenor.

Em alguns navegadores (especialmente no Windows), a escala/ampliação do ecrã está predefinida para um valor elevado (por exemplo, 150 %), o que pode fazer com que partes da interface (como o Cockpit ou o logótipo) pareçam cortadas. Para uma melhor visualização, recomendamos que defina a escala/ampliação do ecrã para 100 % (ou ainda menos).

4.2 Janela principal

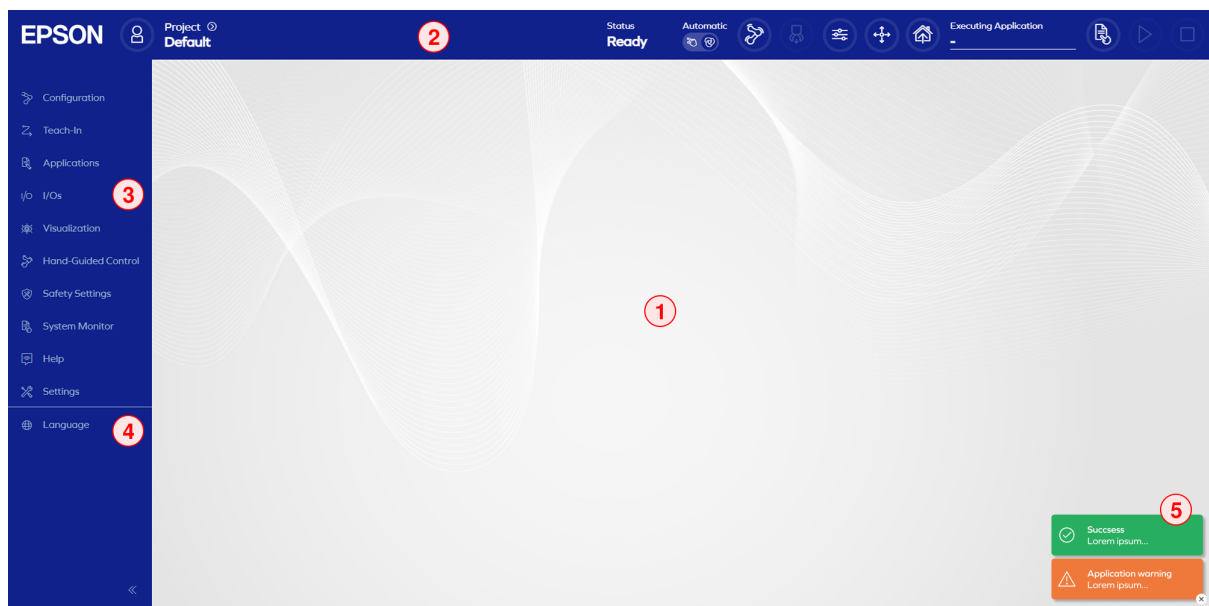


Figura 4.2: Elementos da interface do ambiente de trabalho

1. Área de trabalho, 2. Painel de controlo, 3. Menus, 4. Alterar idioma, 5. Mensagens pop-up

Assim que tiver ligado com sucesso o seu computador ou dispositivo à interface do ambiente de trabalho, a janela principal aparecerá no seu navegador da Web.

Esta janela principal contém os seguintes elementos:

- um painel de controlo na parte superior da janela, que exhibe o estado atual do programa, mensagens e acesso rápido aos controlos do robô,
- os botões de menu no lado esquerdo da janela, nos quais pode clicar para navegar pelos menus disponíveis (todos explicados mais adiante neste capítulo),

- o número da versão no canto inferior direito,
- espaço para pop-ups de mensagens no lado direito do ecrã e
- a área de trabalho principal, na qual o utilizador pode mover e organizar menus.

4.2.1 Visão geral do menu

No lado esquerdo da janela principal, encontrará botões para aceder aos vários menus do AX Portal. A finalidade exata e as informações detalhadas sobre cada menu disponível são descritas nas secções seguintes. Aqui fica uma breve visão geral dos menus disponíveis:

	No menu «Configuração», o utilizador pode configurar e ligar o robô.
	No menu «Programação», o utilizador pode programar poses, percursos e grelhas para o robô.
	No menu «Aplicações», o utilizador pode escrever, alterar e testar aplicações personalizadas.
	No menu «I/Os», o utilizador pode monitorizar as E/S e alterar os sinais de saída.
	O menu «Visualização» mostra o movimento atual do robô.
	No menu «Controlo manual», o utilizador pode mover manualmente as articulações do robô.
	No menu «Configurações de segurança», o responsável pela segurança pode alterar a configuração de segurança.
	No menu «Monitor do sistema», o utilizador pode observar as saídas do sistema.
	O menu «Ajuda» contém informações úteis sobre o programa e o sistema.
	No menu «Definições», o utilizador pode definir algumas configurações básicas.
	Com o botão «Idioma», o utilizador pode alterar o idioma de exibição das interfaces de utilizador.

4.2.2 Cockpit

O Cockpit do AX Portal está sempre visível na parte superior da GUI e contém as informações e os controlos mais importantes. A seguir, todos os elementos do Cockpit são explicados em pormenor, da esquerda para a direita:

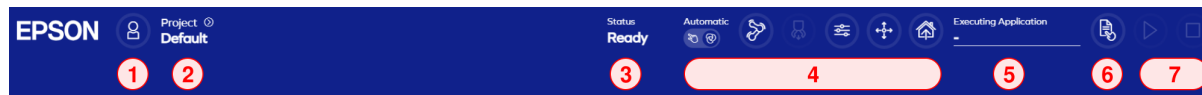


Figura 4.2.2: Cockpit

1. Opções da conta de utilizador, 2. Seleção de projeto, 3. Estado do robô, 4. Controlos do robô, 5. Aplicações atualmente em execução ou selecionadas, 6. Seleção de aplicações, 7. Controlo principal



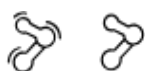
Clique no botão «Utilizador» para alterar as definições da sua conta ou para sair. Estas opções conduzem-no, de uma forma ou de outra, à interface de administração.

Prima o botão «Projeto» para aceder ao menu de projetos, onde pode alternar entre projetos e visualizar a visão geral do projeto. Por baixo deste botão, verá o nome do projeto ativo.

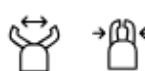
O botão «Solicitar controlo» só é apresentado se esta interface não for o controlo ativo do robô. Para saber mais sobre esta função de segurança, consulte a secção Controlo ativo.

A barra de mensagens de estado indica o estado atual do programa. O estado do programa determina essencialmente se um comando ou uma ação, como carregar um botão, é executável ou não.

Com o controlo deslizante do modo de funcionamento, o robô alterna entre o modo manual e o modo automático. Para saber mais sobre esta função de segurança, consulte a secção Modos de funcionamento.



Com o botão «Ativar/desativar controlo manual», o utilizador pode alternar para o modo manual. Quando a função está ativada, o estado muda para «Controlo manual» e o utilizador pode controlar o robô manualmente. O robô só pode alternar para o modo manual se o estado do programa for «Pronto». Ao desativar, o estado volta a «Pronto».



O botão «Ferramenta» alterna entre as funções «pick» e «place» da ferramenta montada. No editor de ferramentas, o utilizador pode associar estas funções a entradas digitais. A ferramenta só pode ser acionada se o comportamento estiver definido e se o estado do programa for «Pronto» ou «Controlo manual».



O botão «Controlo de precisão» abre uma janela para o controlo de precisão dos ângulos das articulações e das coordenadas do robô. Aqui, o utilizador pode alterar coordenadas individuais do robô ou ângulos das articulações.



O botão «Controlo de movimento» abre uma janela para o controlo de movimento. Aqui, o utilizador pode ajustar a velocidade do robô, alterar os limites globais de velocidade e aceleração, bem como mudar os sistemas de referência.



Com o botão «Homing», o utilizador pode mover o robô para a sua posição de homing. Isto serve como um atalho para, por exemplo, deslocar o robô para uma posição inicial antes de executar uma aplicação ou para uma posição de repouso antes de o desligar. A posição de homing pode ser alterada no menu de configurações.

A área «Atualmente selecionado» indica qual o elemento (aplicação, trajetória, pose, etc.) que está atualmente associado aos botões de controlo (executar, pausar, continuar, parar). Esta informação altera-se assim que clicar num botão de execução ou selecionar uma aplicação.



No menu «Seleção de aplicações», o utilizador pode selecionar uma aplicação que tenha sido criada anteriormente com o Editor de Aplicações.

-
-  O botão «Executar» executa a aplicação seleccionada. O estado do programa deve estar «Pronto» para reproduzir uma aplicação. O botão «Pausar» interrompe uma aplicação em execução. Se uma aplicação for pausada durante um movimento do robô, este irá parar suavemente. O botão «Continuar» retoma uma aplicação em pausa. Se a aplicação tiver sido colocada em pausa durante um movimento, esse movimento é retomado suavemente. O
 -  O botão «Parar» interrompe uma aplicação em execução ou em pausa. Se uma aplicação for interrompida durante um movimento, o robô pára imediatamente e mantém a sua posição. O
 -  estado volta a passar para «Pronto».
 - 
-

4.2.3 Controlo ativo (Operação exclusiva a partir de uma estação de comando)

Para garantir que o robô só possa ser controlado por uma única estação de controlo em cada momento, as interfaces e os painéis de controlo estão protegidos. Se for o primeiro utilizador a iniciar sessão no robô, receberá automaticamente o controlo. Chamamos a isto de controlo ativo. Se, posteriormente, outra pessoa (ou o próprio utilizador) iniciar sessão através de outro canal de ligação (pode ser também outra janela do navegador), essa ligação não terá controlo e terá apenas acesso para monitorizar e observar o robô (acesso de leitura). Chamamos a isto de observador passivo.

4.2.3.1 Controlo ativo

A estação de controlo ativa pode partilhar o seu controlo. Na interface, isto é feito clicando no botão «Utilizador» e seleccionando «Partilhar autorização de controlo». Se costuma trabalhar com vários utilizadores ou interfaces num único robô, recomenda-se que partilhe sempre o acesso de controlo e/ou feche o canal de ligação.

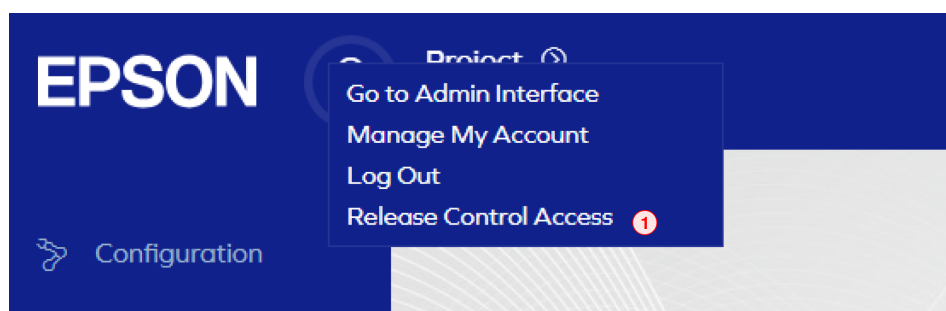


Figura 4.2.3.1 Partilha do acesso ao controlo

1. Partilhar acesso ao controlo

4.2.3.2 Observador passivo

O observador passivo pode solicitar o controlo. Na interface, isto é feito clicando no botão «Utilizador» e selecionando «Solicitar acesso ao controlo» ou simplesmente clicando no botão grande «Solicitar controlo» que aparece no cockpit. Após a solicitação do controlo, surge uma caixa de diálogo a solicitar o acesso, tanto na estação de controlo ativa como no observador passivo. Enquanto o observador passivo apenas pode aguardar até que o acesso ao controlo seja concedido, a estação de controlo ativa pode aceitar ou recusar o pedido.

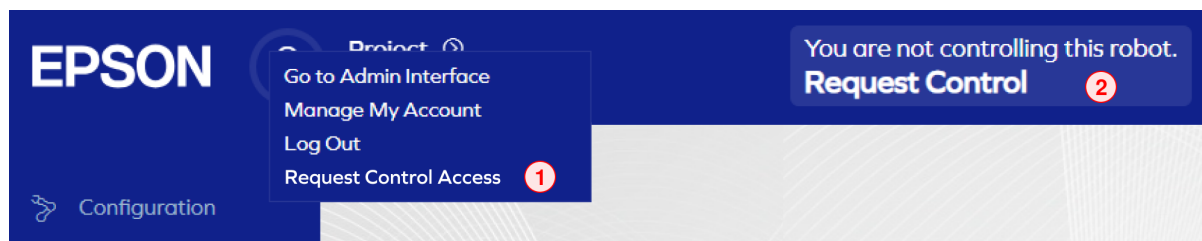


Figura 4.2.3.2 Solicitação de acesso ao controlo
1. Solicitar acesso ao controlo, 2. Botão «Solicitar controlo»

4.2.4 Estado do programa

O estado atual do programa e/ou do robô é visível no cockpit. O estado fornece feedback sobre o que o AX Portal e/ou o robô estão a fazer no momento e determina, essencialmente, quais as ações ou comandos que podem ser executados atualmente. Cada estado tem as suas próprias características e algumas funções só estão disponíveis em determinados estados do programa. Segue-se uma lista dos diferentes estados e as respetivas explicações:

Fora de serviço	Os motores não estão a receber corrente e os dispositivos não estão ligados. Este estado é utilizado para a configuração do robô antes de o ligar. Ao utilizar o controlo deslizante de ligar, o estado muda para «Ligar». Outras ações podem alterar o estado para «A carregar», especialmente se uma ação não deve ser interrompida e requer algum tempo para ser concluída.
A carregar	O robô está a processar dados, por exemplo, a carregar uma base de dados ou definições. Após o carregamento bem-sucedido dos dados, o estado volta a «Fora de serviço».
A ligar	O robô está neste momento a ligar os seus motores e a ligar os seus dispositivos. Após a ligação bem-sucedida, o robô começa a controlar ativamente os seus motores e o estado muda para «Pronto».
Pronto	O programa está pronto para receber comandos e/ou executar ações, tais como a execução de uma aplicação, a pegada e a colocação de objetos, etc. Este é o estado de inatividade do robô.
Em execução	Uma aplicação/movimento está atualmente em execução. Esta aplicação/movimento pode ser pausada ou interrompida com o botão «Pausar» ou «Parar» correspondente. Depois de a aplicação atingir o fim do seu código ou de o movimento em execução terminar, o estado volta a «Pronto».
Em pausa	Uma aplicação/movimento está atualmente em pausa. Prima o botão «Continuar» para prosseguir, o que repõe o estado para «Em execução», ou prima o botão «Parar» para cancelar, o que repõe o estado para «Pronto».

Controlo manual	O Controlo Manual (HGC) está ativado para uma ou mais articulações do robô. Neste modo, pode conduzir manualmente o robô para qualquer pose, exercendo força sobre as suas articulações. Ao mesmo tempo, o robô mantém o seu próprio peso, compensando ativamente a gravidade. Tenha em atenção que terá de premir a tecla HGC ou acionar o botão de confirmação para mover o robô. Desative novamente o modo de controlo manual para restabelecer o estado para «Pronto».
Em processamento	Está a decorrer uma ação demorada, como guardar/carregar dados ou calcular um percurso. Após a conclusão bem-sucedida desta ação, o estado volta ao seu valor anterior.
Parado	Normalmente, o estado «Paragem» recupera automaticamente e em muito pouco tempo para «Pronto». Caso tal não aconteça, uma parte do AX Portal não pôde ser concluída corretamente. Infelizmente, isto só pode ser resolvido reiniciando o robô.
Paragem de emergência Paragem normal Paragem de proteção Colisão Paragem de segurança Limitação de posição limitação	Estes estados correspondem a várias interrupções. Se for desencadeada uma interrupção, o estado correspondente é apresentado e a janela de interrupção abre-se para o tratamento da interrupção. Assim que a interrupção for resolvida, o estado muda normalmente para «A ser continuado».
Em Andamento	Após a resolução de uma interrupção, o estado muda para «Em Andamento». Isto requer que o utilizador confirme manualmente que é seguro prosseguir com a aplicação ou o movimento do robô.
Erro	Ocorreu um erro que não pôde ser resolvido automaticamente ou que requer atenção especial do utilizador. Consulte a mensagem de erro e tente resolver o problema, se possível.
Desligar	O robô está a desligar os seus motores, a desligar os seus dispositivos e a encerrar todos os módulos em execução. Em seguida, o estado muda para «Fora de serviço».

4.2.5 Modos de funcionamento

O robô da série AX dispõe de dois modos de funcionamento: modo manual e modo automático. O modo de funcionamento pode ser alterado através do cursor correspondente no cockpit.

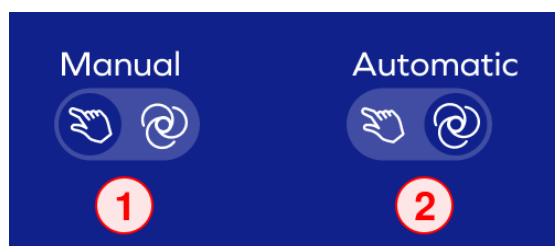


Figura 4.2.5 Interruptor para seleção do modo de funcionamento

 AVISO	O modo automático só deve ser utilizado se o sistema estiver equipado com as funções de segurança necessárias, tiver sido testado e a sua segurança tiver sido confirmada por uma avaliação de riscos. A utilização do modo automático noutras situações é extremamente perigosa. Para mais informações, consulte o «AX6 / RC-A101: Manual de Segurança».
---	---

4.2.5.1 Modo automático

O modo automático é normalmente utilizado em funcionamento normal, ou seja, quando se verificou que as aplicações estão a funcionar corretamente. Este modo de funcionamento permite toda a gama de velocidades do robô.

4.2.5.2 Modo manual

O modo manual é uma função de segurança e é normalmente utilizado ao alterar as aplicações do robô, em particular ao programar novas posições ou ao testar alterações nas aplicações. No modo manual, o robô move-se a velocidade reduzida (máx. 250 mm/s) e apenas quando o interruptor de autorização é acionado. Trata-se de um interruptor de segurança de 3 posições. Se o interruptor de autorização não for acionado, o utilizador é informado por meio de um aviso que indica que os movimentos não são permitidos.



Figura 4.2.5.2 Exemplo de um interruptor de autorização

4.2.6 Controlo de precisão

Prima o botão de controlo de precisão no cockpit para abrir as opções de controlo de precisão. Aqui, o robô pode ser ajustado com uma precisão de até 0,1 [mm]/[°]. Esta janela tem dois separadores. O primeiro separador permite ao utilizador alterar as coordenadas euclidianas atuais do TCP (Tool Center Point), o que corresponde ao sistema de referência atual da ferramenta, expresso no sistema de referência de trabalho. O segundo separador permite ao utilizador alterar os ângulos atuais das articulações do robô.

Estes separadores dispõem das seguintes funções:

Campos de entrada	Ao introduzir um número no campo de entrada, altera-se uma única coordenada ou um valor de articulação.
Controlo deslizante	O controlo deslizante define tanto o intervalo de passo, que determina a distância que o robô percorre com um clique curto num botão de controlo, como a velocidade máxima de movimento, que determina a rapidez com que o robô se move com estes controlos.
Botões	Um toque curto no botão provoca o avanço do passo selecionado na direção indicada pelo botão. Um toque longo num botão, por outro lado, desencadeia um movimento contínuo na direção indicada.
Configuração	Um determinado TCP pode ser alcançado através de até oito combinações de ângulos articulares. Uma combinação de ângulos articulares é designada como configuração dessa pose. Ao alterar o valor «Configuração» na lista suspensa, o utilizador pode selecionar uma configuração específica para a pose atual. E ao clicar no botão «Próxima configuração», o utilizador pode percorrer as configurações disponíveis da pose atual. Note que alterar a configuração não é um passo de ajuste fino; neste caso, todas as articulações se movem.
Guardar como Pose	Ao clicar no botão «Guardar como Pose», esta pose é guardada com um novo nome único. Todas as poses disponíveis são listadas no menu «Aprendizagem».



Figura 4.2.6.a: Controlo de precisão – Controlo de coordenadas

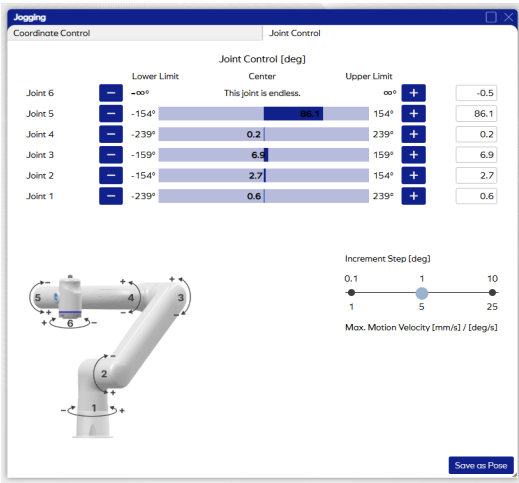


Figura 4.2.6.b: Ajuste fino – Controlo das articulações

4.2.7 Controlo de movimento

Clique no botão «Controlo de movimento» no cockpit para abrir a janela de controlo de movimento. Esta janela dispõe os seguintes três separadores:

4.2.7.1 Limites de movimento

O separador «Limites de movimento» contém definições para limitar a velocidade máxima do robô e para reduzir a velocidade de todos os movimentos do robô através de um fator de escala. Estes controlos são úteis durante a colocação em funcionamento de aplicações robóticas, por exemplo, para limitar a velocidade para validar uma nova aplicação ou para alterar movimentos em aplicações existentes.

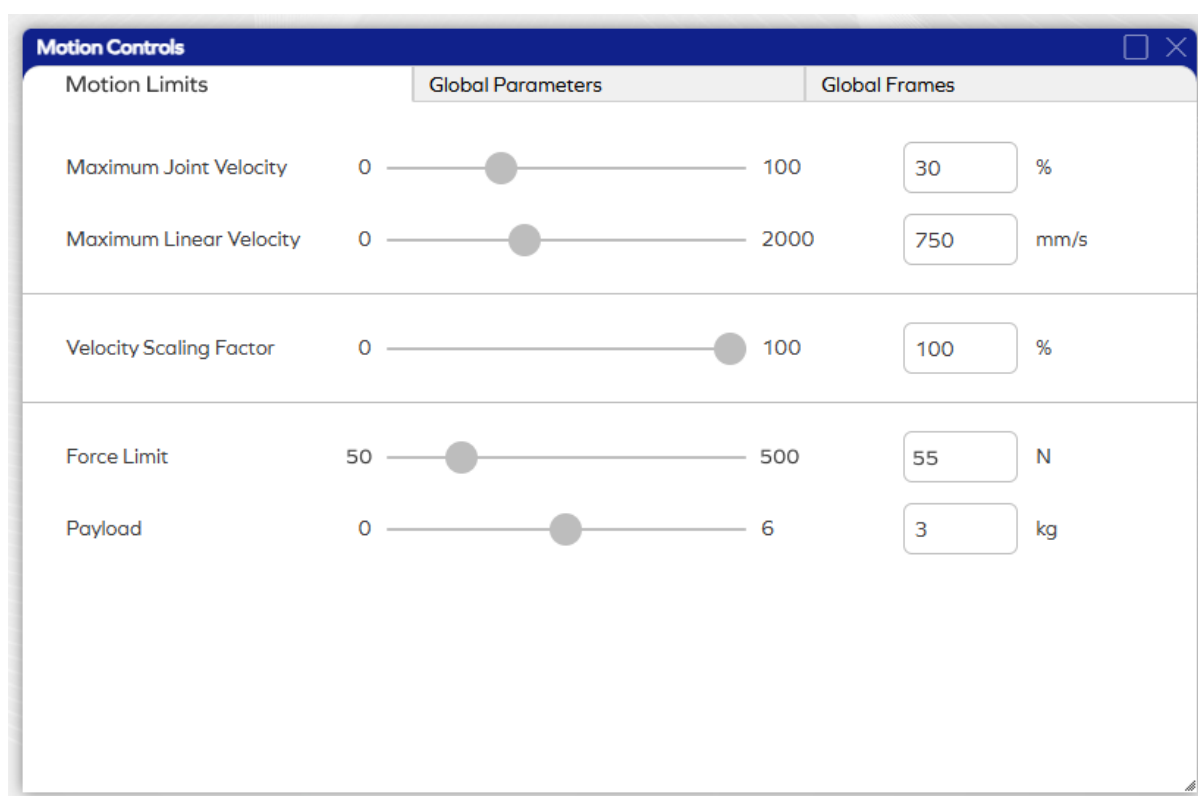


Figura 4.2.7.1: Controlo de movimento – separador «Limites de movimento»

Por outro lado, o limite de força e a carga útil são utilizados para ajustar a sensibilidade do robô a colisões. O limite de força define um limite máximo para as forças externas admissíveis na ferramenta, e a carga útil define o peso total que o robô suporta (incluindo a sua ferramenta).

4.2.7.2 Parâmetros globais

O separador «Parâmetros globais» contém definições para definir a velocidade e a aceleração padrão para vários comandos de movimento. Os comandos de movimento são divididos nas seguintes categorias:

- Os comandos para o movimento do robô com otimização de tempo utilizam os argumentos «joint_velocity» e «joint_acceleration», cujos valores padrão são aqui definidos como **Velocidade Global da Articulação** e **Aceleração Global da Articulação**.
- Os comandos para mover o robô numa determinada direção ou forma (linear, circular) utilizam os argumentos «linear_velocity» e «linear_acceleration», cujos valores padrão são aqui definidos como **Velocidade Linear Global** e **Aceleração Linear Global**.
- Os comandos para mover o robô de forma a que o TCP (Tool Center Point) permaneça na mesma posição e apenas a sua orientação mude utilizam os argumentos «angular_velocity» e «angular_acceleration», cujos valores padrão são aqui definidos como **Velocidade de rotação global** e **Aceleração de rotação global**.
- Os comandos para mover ao longo de um percurso que contém vários segmentos de movimento utilizam o argumento «blend_radius», cujo valor padrão é aqui definido como **Raio de transição global**. Este parâmetro é utilizado para suavizar a interpolação entre segmentos do percurso.

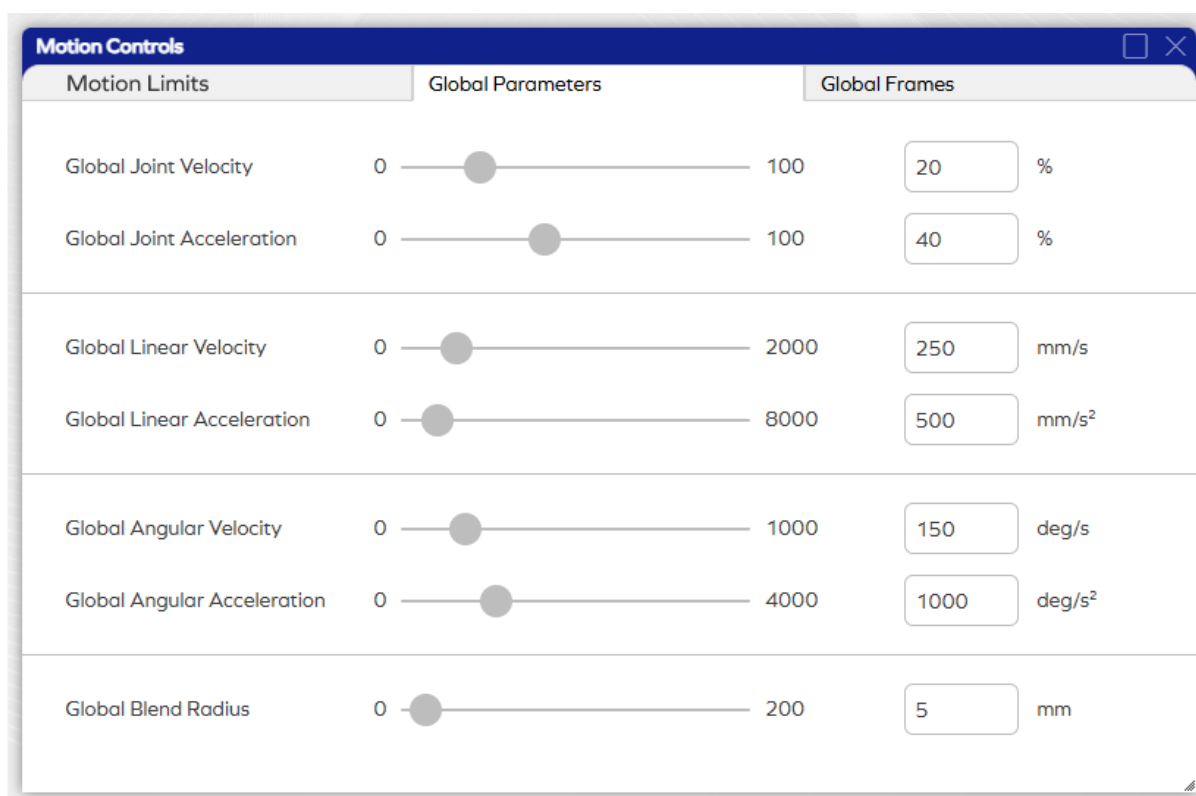


Figura 4.2.7.2: Controlo de movimento – Separador «Parâmetros globais»

4.2.7.3 Sistemas de coordenadas globais

O separador «Sistemas de coordenadas globais» contém definições para definir os sistemas de coordenadas de referência padrão do robô. Enquanto o sistema de base e o sistema de flange estão fixos ao robô, o sistema de trabalho e o sistema de ferramenta podem ser definidos em relação a estes sistemas de referência fixos:

- O **sistema de base** está fixado no centro da superfície de montagem inferior do robô, de modo que o eixo z aponta para o braço do robô e o eixo x aponta para o outro lado dos cabos ligados.
- O **sistema de flange** está fixado no centro da placa de montagem superior do robô. A sua orientação está alinhada com o sistema de referência quando todas as articulações do robô estão na posição de zero graus.
- O **sistema de trabalho** é o sistema de referência no qual o utilizador expressa as coordenadas do TCP (Tool Center Point) nas aplicações. Por predefinição, o sistema de trabalho corresponde ao sistema de base do robô. Muitos comandos de movimento utilizam o argumento «work_frame», cujo valor predefinido é aqui definido como Sistema de Trabalho Global.
- O **sistema de ferramenta** define a posição e a orientação do TCP (Tool Center Point), expressas no sistema de flange. Por padrão, o sistema de trabalho corresponde ao sistema de flange do robô. Muitos comandos de movimento utilizam o argumento «tool_frame», cujo valor padrão é aqui definido como Sistema de Ferramenta Global.

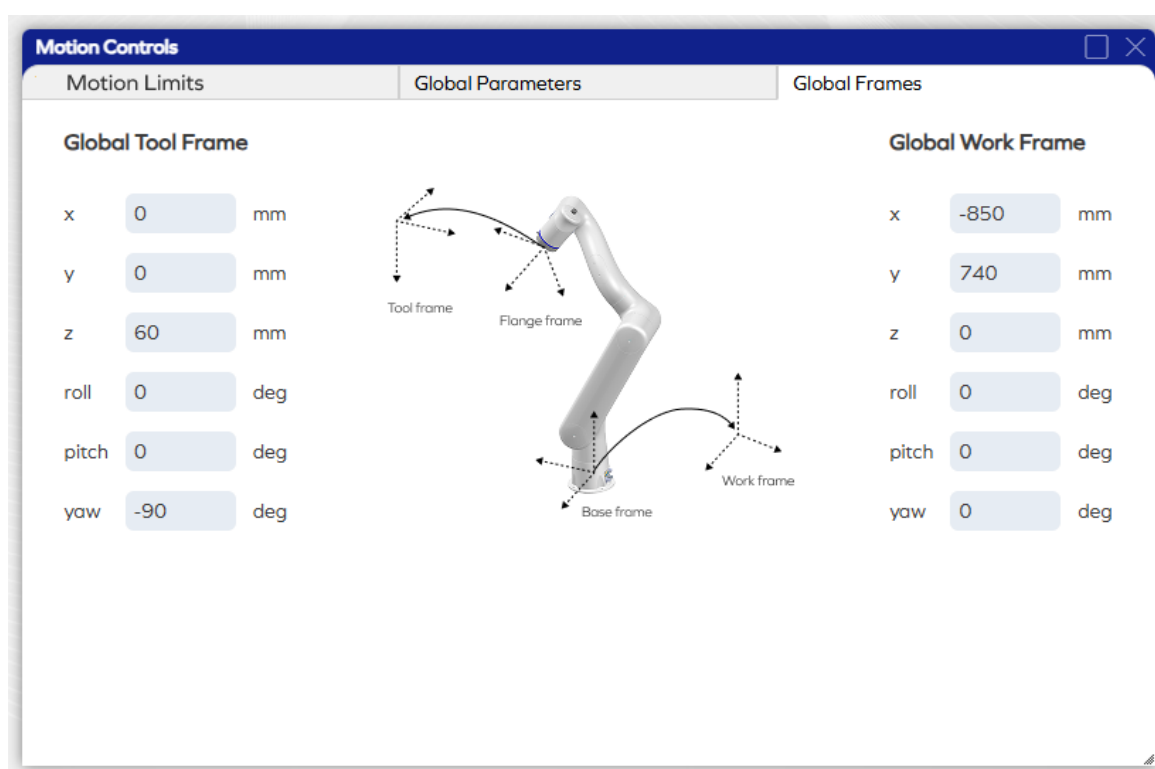


Figura 4.2.7.3: Controlo de movimento – separador «Sistemas de coordenadas globais»

4.3 Blocos de construção e base de dados

Para permitir uma programação de aplicações simples e intuitiva, o AX Portal oferece uma estrutura orientada para objetos com vários blocos de construção. Estes blocos de construção (doravante «elementos») podem ser criados individualmente em vários editores, acessíveis através da interface do ambiente de trabalho. O AX Portal permite então ao utilizador ligar estes elementos entre si de forma inteligente, para criar aplicações orientadas para objetos, o que resulta em código de aplicação conciso, de fácil leitura e que, normalmente, não contém muitos números e coordenadas.

Segue-se uma lista dos elementos disponibilizados:

	As aplicações (ou aplicações) contêm principalmente o código de programa a executar (e/ou blocos) e configurações específicas da aplicação. O código de programa suporta Python 3.13 e, além das funções básicas do Python, contém também muitas funções da API (Interface de Programação de Aplicações) interna de controlo de robôs. As aplicações são criadas e alteradas com o Editor de Aplicações.
	As poses são posições do Ponto Central da Ferramenta (TCP) com orientações específicas, definidas de forma unívoca por uma combinação de ângulos articulares. As poses são criadas e alteradas com o Editor de Poses.
	Os percursos são sequências de poses que descrevem movimentos suaves do TCP e, consequentemente, movimentos de todo o robô. Os percursos são criados e alterados com o Editor de Percursos.
	As grelhas são disposições de pontos personalizáveis, até tridimensionais, que definem, por exemplo, as posições de tabuleiros predefinidos. Os pontos da grelha são extraídos e utilizados como iteradores nas aplicações. As grelhas são criadas e alteradas com o Editor de Grelhas.
	As E/S dividem-se em entradas e saídas digitais e analógicas. Podem ser configuradas através do menu «Configuração de E/S», e o valor das E/S é visível no menu «E/S». As E/S servem para a ligação e comunicação com dispositivos externos.

Todos estes elementos são armazenados na base de dados do AX Portal. Pode descarregar a base de dados do seu robô a qualquer momento (para ter uma cópia de segurança de toda a sua configuração) ou carregar uma nova base de dados (substituindo a base de dados atual e todos os seus elementos) – através do menu de configurações. Além disso, o menu de projetos oferece uma excelente visão geral de todos os projetos e elementos armazenados na sua base de dados.

4.4 Menu de Projetos

Clique no nome do projeto atualmente selecionado no Cockpit para aceder ao menu de projetos. Este menu contém uma visão geral dos elementos disponíveis, tais como poses, percursos, grelhas e aplicações. Estes elementos estão agrupados em vários projetos. Apenas um projeto está selecionado de cada vez, e apenas os elementos do projeto selecionado são visíveis, editáveis e utilizáveis nos outros menus do AX Portal.

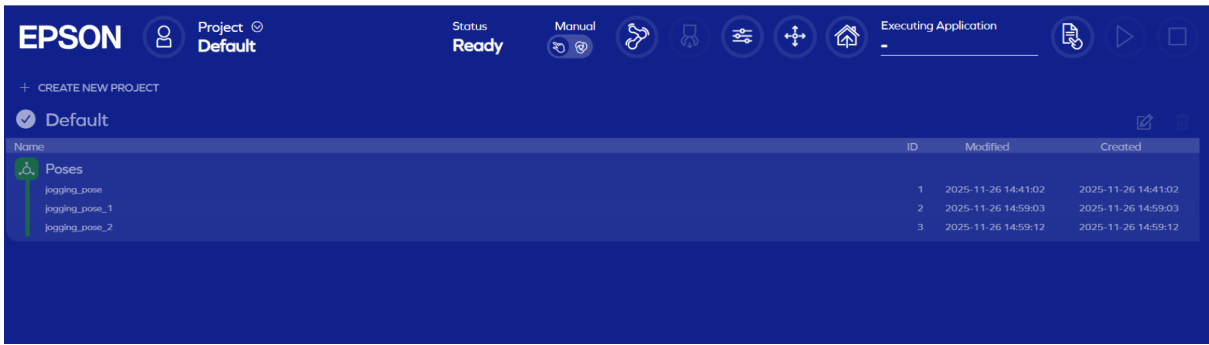


Figura 4.4: Menu do projeto

-
-  Clique no botão «Criar novo projeto» para criar um novo projeto vazio.
 -  Clique no círculo à esquerda do nome do projeto para selecionar esse projeto.
 -  Clique no botão «Copiar» de um elemento do projeto para criar uma cópia exata desse elemento dentro do mesmo projeto. O nome do elemento copiado é ligeiramente alterado e termina com «(cópia)».
 -  Clique no botão «Editar» à direita do nome do projeto para alterar o nome desse projeto. Clique no botão «Editar» de um elemento do projeto para abrir o editor correspondente para visualização ou edição.
 -  Clique no botão «Eliminar» à direita do nome do projeto para remover esse projeto, incluindo todos os seus elementos. Clique no botão «Eliminar» de um elemento do projeto para remover esse elemento específico de um projeto.
-
- Clique num ícone de elemento (por exemplo, poses, percursos, etc.) para mostrar ou ocultar a lista de elementos nessa área específica.
-
- Mova um elemento de um projeto para outro, arrastando-o com o rato a partir do projeto original e soltando-o no projeto de destino. Mantenha a tecla CTRL premida enquanto arrasta o elemento para criar uma cópia desse elemento no projeto de destino.
-

4.5 Menu de configuração

Clique no botão «Configuração» na janela principal para aceder ao menu de configuração. Neste menu, pode configurar a base de dados e a configuração de hardware (E/S e ferramentas). Se estiver a utilizar um robô fixo para uma aplicação específica, só precisa de definir a configuração uma vez e, posteriormente, utilizar esta janela apenas para ligar o robô. Pode até automatizar a inicialização do robô e a execução de uma aplicação através das definições correspondentes no menu de definições. No entanto, se utilizar o seu robô para vários fins, recorrerá a este menu com mais frequência e utilizará diferentes ferramentas ou mesmo diferentes bases de dados para diferentes projetos e finalidades.

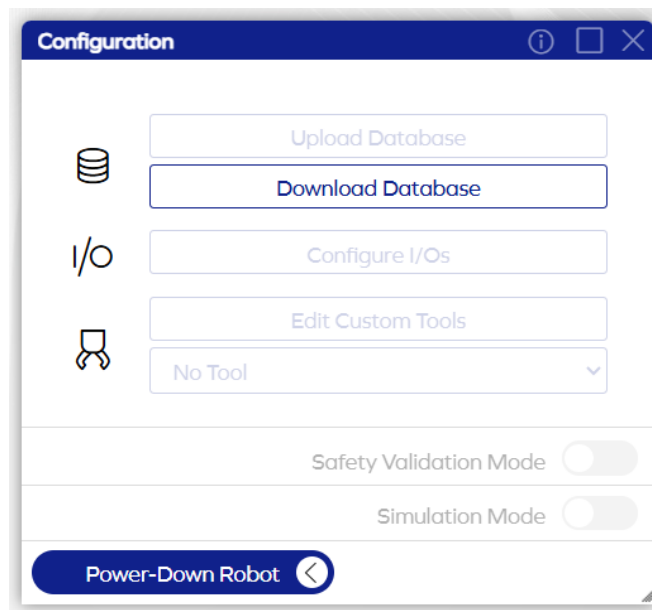


Figura 4.5: Menu de configuração

4.5.1 Base de dados (transferência e carregamento)

Na secção «Base de dados», pode descarregar a versão atual da base de dados da sua aplicação ou subir (e substituir) a base de dados por outra versão compatível. Ao descarregar a base de dados, é criada uma cópia local do ficheiro ZIP da base de dados no seu dispositivo. Recomenda-se que faça regularmente uma cópia de segurança da sua base de dados, especialmente se tiver muitas aplicações e tiver investido muito tempo na criação e aperfeiçoamento das mesmas. Ao carregar uma base de dados, será-lhe pedido que selecione um ficheiro ZIP do seu dispositivo. Ao selecionar o ficheiro da base de dados, a base de dados atual será substituída pela que foi carregada.

Ao descarregar uma base de dados para um robô e carregá-la para outro robô, pode transferir todas as aplicações e elementos entre robôs. Recomenda-se que faça cópias de segurança regulares da base de dados.

Tenha também em atenção que as bases de dados são versionadas e que nem todas as bases de dados são compatíveis com todas as versões do AX Portal. As bases de dados mais antigas podem ser importadas para versões mais recentes de software, mas as versões mais recentes da base de dados não podem ser carregadas em versões mais antigas de software. Por outras palavras: as bases de dados não são retrocompatíveis.

4.5.2 Configuração de E/S

Clique no botão «Configurar E/S» para abrir a janela «Configuração de E/S». Aqui pode configurar todas as E/S, atribuir funções predefinidas às entradas digitais e definir nomes personalizados para as suas E/S. Os nomes definidos pelo utilizador podem ser utilizados em aplicações para aumentar a legibilidade do seu código de programa. Aí, pode simplesmente substituir os identificadores inteiros (int) pelos nomes de E/S definidos (str).

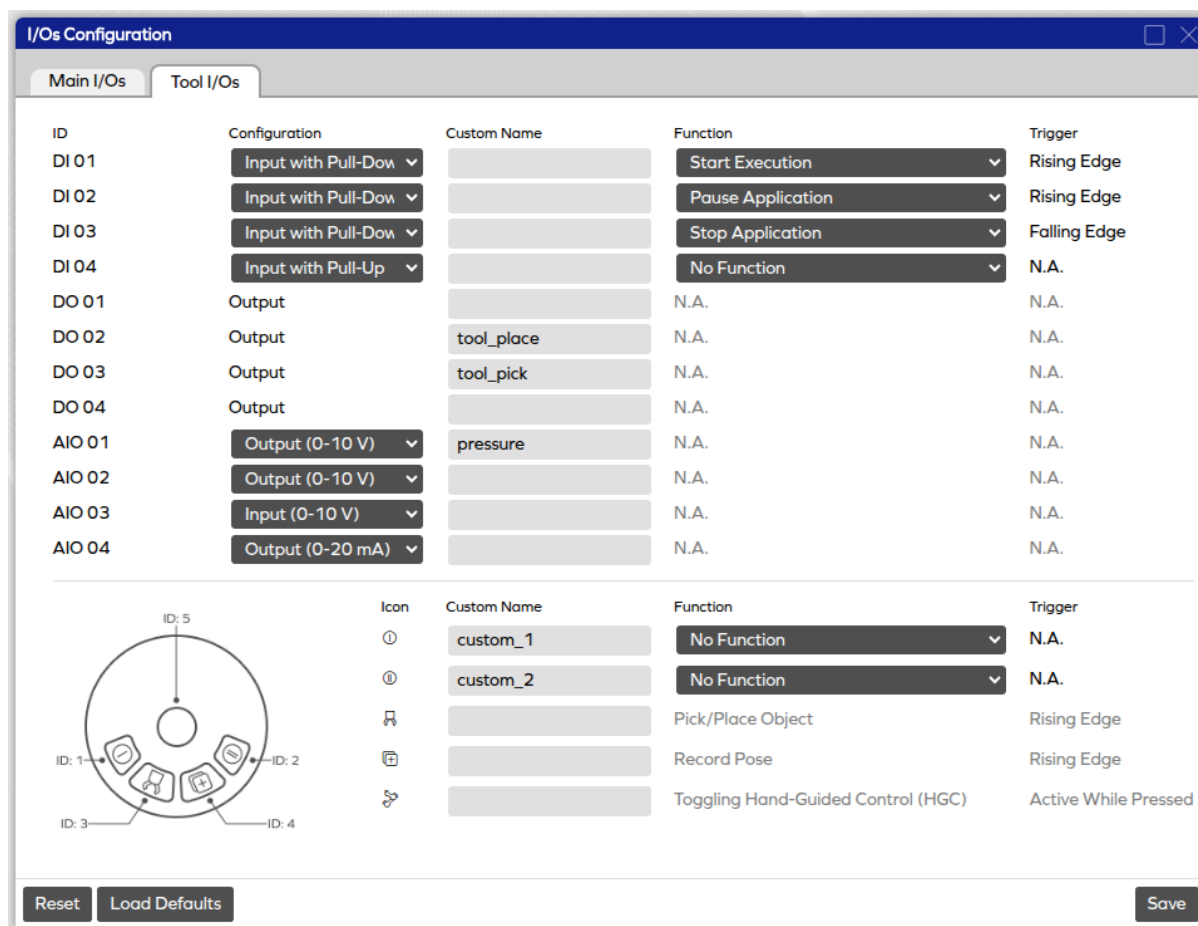


Figura 4.5.2.a: Menu «Configuração de E/S»

A série AX é fornecida com uma «interface de botões» montada no eixo superior. É possível atribuir funções aos dois botões personalizados exatamente da mesma forma que as funções são atribuídas a outras entradas digitais neste menu.

Tenha em atenção que, ao acionar uma função através de uma entrada digital, o controlo ativo é retirado da estação do controlo atual.

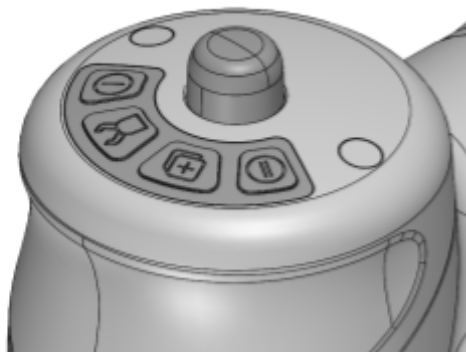


Figura 4.5.2.b: Interface de botões AX

Estão disponíveis as seguintes funções predefinidas para entradas digitais, que pode configurar para serem acionadas na frente ascendente (padrão) ou na frente descendente:

Sem função	Nada é acionado. Esta é a configuração padrão.
Pausar a aplicação	Pausar a aplicação atualmente em execução. Isto corresponde a premir o botão «Pausar» no cockpit.
Cancelar aplicação	Interrompe a aplicação atualmente em execução ou em pausa. Isto corresponde a premir o botão «Parar» no Cockpit.
Iniciar execução	Executar a aplicação atualmente selecionada. Isto equivale a premir o botão «Executar» no Cockpit. Quando o robô recupera após uma interrupção, esta função leva o robô de volta à posição em que foi interrompido e continua o movimento anterior. Isto corresponde a premir o botão «Continuar» na caixa de diálogo que é apresentada na GUI após uma interrupção.
Reinicialização de segurança	Reiniciar a segurança após uma interrupção. Isto corresponde à redefinição da segurança através da caixa de diálogo que é apresentada na GUI após uma interrupção.
Ativar o controlo manual	Ativar o modo de operação manual (HGC) para todos os eixos (enquanto o estado for «pronto»).
Desativar o controlo manual	Desativar o modo de comando manual (HGC) para todos os eixos (enquanto o estado for «comando manual»).
Alternar o controlo manual	Alternar entre ativar e desativar o HGC para todos os eixos. A função acionada depende do estado atual.
Operar a ferramenta (cancelar)	Ativar a função de recolha da ferramenta (predefinida ou associada a saídas digitais). Isto corresponde a premir o botão «retirar» no cockpit.
Operar a ferramenta (colocar)	Ativação da função de colocação da ferramenta (predefinida ou associada a saídas digitais). Isto corresponde a premir o botão «colocar» no cockpit.
Operar a ferramenta (retirar/colocar)	Alternar entre pegar e colocar um objeto com a ferramenta. Depende do estado atual da ferramenta.
Guardar pose	Registe a pose atual do robô e guarde-a como «button_pose» (ou similar).
Mudar para o modo manual	Mudança do robô para o modo manual. Isto corresponde à mudança do modo de funcionamento através do interruptor no cockpit.
Mudar para o modo automático	Mude o robô para o modo automático. Isto corresponde à mudança do modo de funcionamento através do interruptor no cockpit.

4.5.3 Configuração da ferramenta

Na área de ferramentas, pode configurar a ferramenta do robô. Se pretender integrar uma ferramenta personalizada, por exemplo, uma ferramenta de um fornecedor externo ou a sua própria ferramenta especial para uma aplicação específica, pode utilizar a opção «Editar ferramentas personalizadas». Isto abre a lista de ferramentas disponíveis com as seguintes opções:

	Clique no botão «Criar nova ferramenta» para aceder ao editor de ferramentas, onde poderá definir as propriedades cinemáticas e dinâmicas, bem como definir o gatilho de recolha e colocação da sua ferramenta.
	Clique no botão «Filtrar/Pesquisar» para filtrar a lista de ferramentas personalizadas por nome. A introdução de três ou mais caracteres ativa o filtro de pesquisa.
	Clique no botão «Editar» (ou clique duas vezes em qualquer ponto do elemento) para abrir o editor de ferramentas numa janela separada para visualização ou edição.
	Clique no botão «Eliminar» para remover irrevogavelmente uma ferramenta da sua base de dados.

4.5.4 Editor de ferramentas

O Editor de Ferramentas está dividido em quatro secções (ver Figura 4.5.4 abaixo). Para preencher estes valores, poderá ser necessário consultar a ficha técnica da ferramenta que pretende integrar no AX Portal. As características físicas e elétricas da ferramenta devem estar aí descritas.

Na primeira etapa, deve definir o TCP (Tool Center Point), ou seja, a posição no espaço em que a ferramenta interage. No caso de uma ferramenta que agarra objetos, isto significa que o TCP define o ponto em que a ferramenta se encontra durante o processo de agarra. O TCP é definido no sistema de coordenadas de referência, que está posicionado exatamente no centro da placa de fixação superior e que está alinhado da mesma forma que o sistema de base (desde que todos os eixos estejam a zero graus). Este sistema de referência é o sistema de ferramenta padrão, caso não esteja definido nenhum TCP (x, y, z, roll, pitch e yaw todos a zero). A partir daí, pode deslocar e rodar o TCP de forma a que corresponda ao ponto de interação da sua ferramenta.

No passo seguinte, é necessário ajustar as propriedades dinâmicas da sua ferramenta. A qualidade de funções como a compensação da gravidade e a deteção de colisões depende de o modelo dinâmico do robô ser o mais preciso possível; por isso, vale a pena medir e definir estas propriedades da sua ferramenta. Deve definir a massa total da ferramenta, bem como o centro de massa, que (tal como o TCP) é definido no sistema de referência padrão da ferramenta do robô. O [momento de inércia da massa](#) pode ser introduzido manualmente (caso seja conhecido) ou aproximado automaticamente (o AX Portal não conhece a forma e a distribuição de massa da sua ferramenta).

Tool Editor : pneumatic_gripper

Name* pneumatic_gripper

1. Defining the Global Tool Frame

Translation [mm]

x 0

y 0

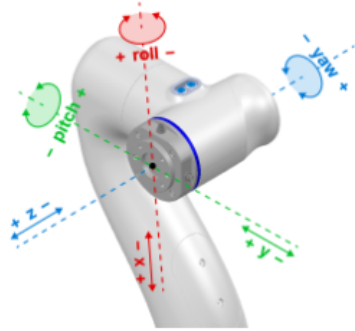
z 36

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

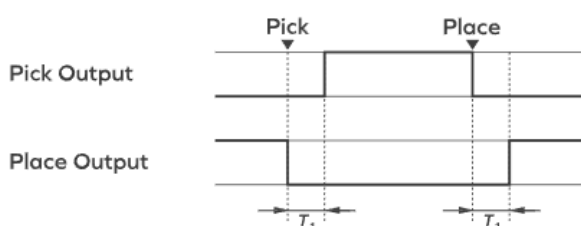
Mode Dual Output Mode

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

Pick

Place

T_1

Cancel Reset Save & Close

Figura 4.5.4: Editor de ferramentas

Se a sua ferramenta tiver um mecanismo para agarrar um objeto, pode definir aqui quais as saídas digitais que serão acionadas pelas funções «levantar» e «colocar», acessíveis tanto através do cockpit como das funções de programação. Estão disponíveis os seguintes comportamentos de prensão:

4.5.4.1 Modo de saída única

Este é o mais simples dos comportamentos de prensão disponíveis. Basta definir uma saída digital que será utilizada para alternar entre prensão e colocação, e pode especificar se a prensão é acionada por um sinal HIGH ou LOW.



Figura 4.5.4.1: Modo de saída única com HIGH para pegar (e LOW para largar)

4.5.4.2 Modo de saída dupla

Este comportamento de prensão permite-lhe definir saídas digitais separadas para pegar e colocar. Ao alternar o estado da ferramenta, ambas as saídas são então adicionadas sucessivamente com um breve atraso T_1 (em milissegundos). Além disso, pode definir se o estado ativo se refere a um sinal HIGH ou LOW.

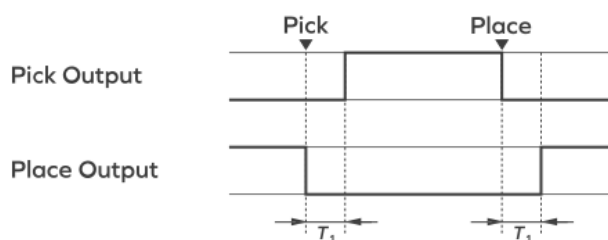


Figura 4.5.4.2: Modo de saída dupla com HIGH como estado de ativação ativo

4.5.4.3 Modo de disparo separado

No modo de disparo separado, pode definir duas saídas digitais: uma que determina a direção (se deve ser cancelada ou colocada) e outra que determina o disparo (se a direção selecionada deve ser acionada). Para a saída de direção, pode escolher se um sinal HIGH ou LOW será utilizado para cancelar (o outro é para colocar). Para a saída de acionamento, pode igualmente escolher se será utilizado um sinal HIGH ou LOW para o acionamento. Além disso, pode definir o breve intervalo de tempo T_1 (em milissegundos) entre a seleção da direção e o acionamento da direção selecionada, bem como o breve intervalo de tempo T_2 (em milissegundos) que determina a duração do acionamento.

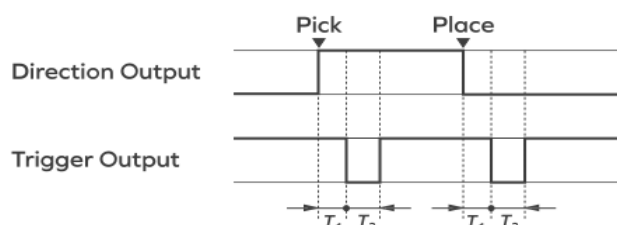


Figura 4.5.4.3: Modo de acionamento separado com HIGH para agarrar e LOW como acionador

4.5.5 Colocação do robô em funcionamento

Depois de carregar a base de dados e de configurar as E/S e a ferramenta, é altura de ligar o robô. Para tal, desloque o controlo deslizante «Colocar o robô em funcionamento» totalmente para a direita, o que inicia a colocação em funcionamento.

Se não pretender ligar o hardware, mas apenas alterar o conteúdo da base de dados (aplicações, poses, etc.) sem utilizar o hardware (por exemplo, sem mover o robô, sem ler ou escrever E/S), pode ativar previamente o botão «Modo de simulação» e, em seguida, deslizar o controlo deslizante «Colocar o robô em funcionamento» para a direita. Tenha em atenção que, no modo de simulação, todos os movimentos são simulados em tempo real.

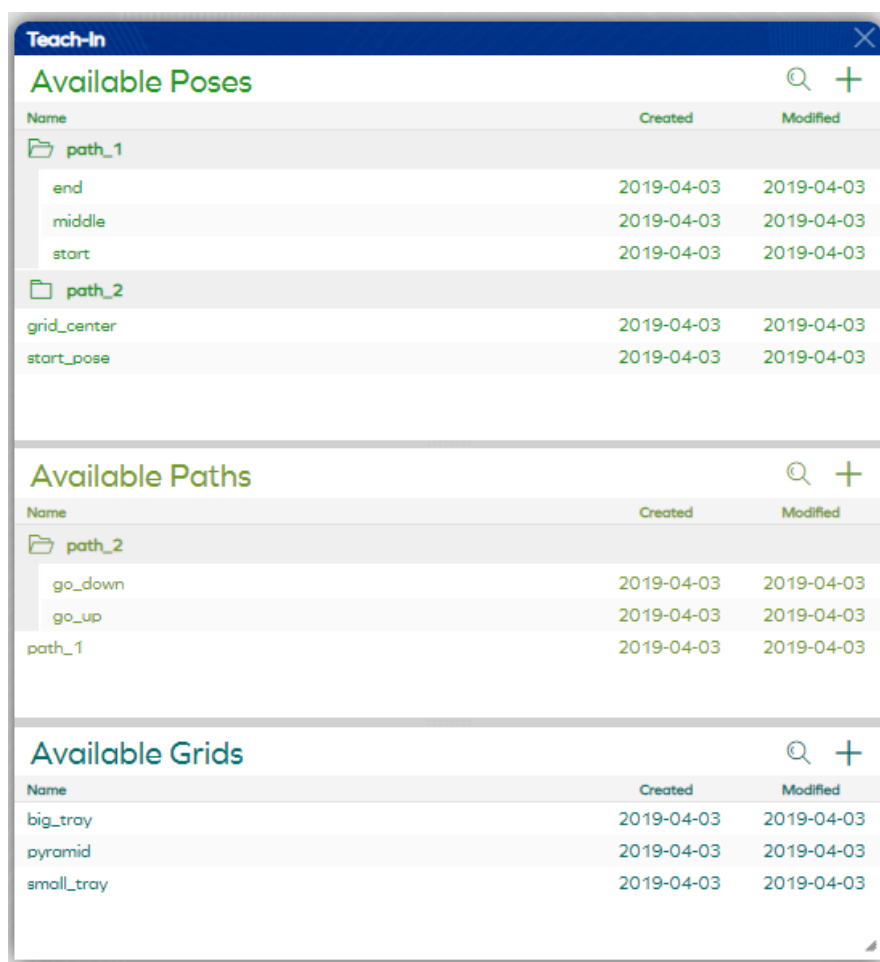
Enquanto o robô estiver ligado, não poderá alterar nenhuma das configurações de hardware do robô. Pode apenas descarregar a base de dados ou arrastar o controlo deslizante «Colocar o robô fora de serviço» para a esquerda, o que é recomendado quando o robô está inativo ou em modo de espera.

Se estiver registado como responsável pela segurança, terá aqui acesso ao botão «Modo de verificação de segurança». A ativação deste modo instrui a software a ignorar os limites de posição, velocidade e força, o que pode ser utilizado para verificar se o dispositivo de segurança é efetivamente acionado em caso de violação dos limites. Tenha em atenção que a utilização de qualquer um dos controlos de movimento neste modo continua a sobrepor-se aos limites.

4.6 Menu de aprendizagem

Clique no botão «Aprendizagem» na janela principal para aceder ao menu de aprendizagem. Este menu contém uma visão geral de todas as poses, percursos e grelhas disponíveis do projeto atualmente selecionado.

As poses e os percursos podem ser criados, editados, copiados, eliminados e até mesmo executados diretamente. As grelhas, por outro lado, são ferramentas úteis para manter organizadas coordenadas recorrentes e uniformemente espaçadas. Tenha em atenção que as grelhas não são utilizadas diretamente para mover um robô, mas sim para agrupar e guardar conjuntos de valores de coordenadas, de modo a manter o seu código de aplicação estruturado e evitar que tenha de introduzir coordenadas. É possível extrair uma lista de pontos de grelha de uma grelha para diversos fins; no entanto, normalmente as grelhas são configuradas de forma a que os seus pontos representem diretamente coordenadas TCP (Tool Center Point), que são depois utilizadas como valores de entrada para funções de movimento.



The screenshot shows the 'Teach-In' window with the 'Available Poses' menu. It contains three sections: 'Available Poses', 'Available Paths', and 'Available Grids'. Each section has a table with columns for Name, Created, and Modified.

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

Figura 4.6: Menu de aprendizagem

	Clique no botão «Novo/Adicionar» para abrir um editor de pose, trajetória ou grelha numa nova janela. Mais adiante nesta secção, todos estes editores serão explicados em pormenor.
	Clique no botão «Filtrar/Pesquisar» para filtrar a lista de poses, percursos ou grelhas por nome. A introdução de três ou mais caracteres ativa o filtro de pesquisa.
	Clique no botão «Executar» para mover o robô para uma pose ou reproduzir um percurso. Tenha em atenção que, ao reproduzir um percurso, o robô se desloca primeiro para a pose inicial desse percurso, caso ainda não se encontre nessa pose específica.
	Clique no botão «Copiar» para criar uma cópia exata dessa pose ou trajetória. O nome do elemento copiado é ligeiramente alterado e termina com «(copy)».
	Clique no botão «Editar» (ou clique duas vezes em qualquer ponto do elemento) para abrir essa pose, esse percurso ou essa grelha numa janela separada para visualização ou edição.
	Clique no botão «Apagar» para remover irrevogavelmente uma pose, um caminho ou uma grelha da sua base de dados.
	Clique numa pasta para mostrar ou ocultar as poses, os percursos ou as grelhas dentro dessa pasta. Os elementos só são apresentados em pastas (e subpastas) se os seus nomes contiverem um ou mais caracteres «/», por exemplo, «pasta/subpasta/nome».

4.6.1 Editor de poses

O Editor de Posições permite aos utilizadores criar uma nova posição ou alterar uma já existente. Tenha em atenção que a alteração das coordenadas ou da configuração de uma posição não move automaticamente o robô. Caso tal seja desejado, deve utilizar-se, em vez disso, o Controlo de Precisão.

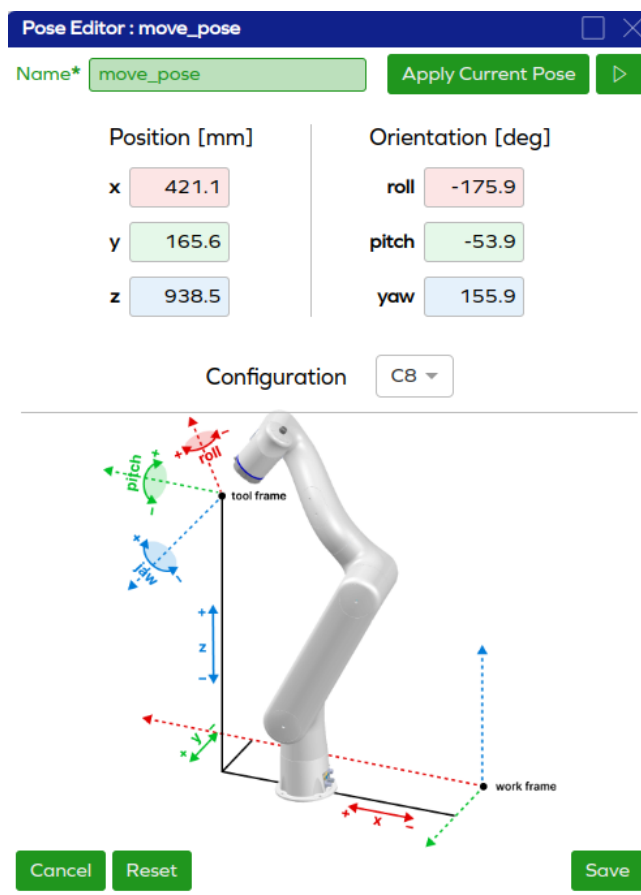


Figura 4.6.1: Editor de Posições

Nome:

Indica o nome sob o qual a pose é guardada na base de dados. Utilize o caractere «/» no nome para agrupar as suas poses numa estrutura semelhante a pastas.

Posição:

Representa a posição TCP (Tool Center Point) em milímetros, definida em relação ao sistema de trabalho global. Tenha em atenção que podem ser introduzidos valores de posição inválidos, por exemplo, aqueles que não podem ser alcançados pelo braço robótico. No entanto, estes serão destacados a vermelho, uma vez que premir o botão de reprodução resultará num erro. Isto é permitido porque a pose pode ser válida quando utilizada com diferentes sistemas de referência de coordenadas e desvios.

Orientação:

Representa a orientação do TCP em graus, definida em relação ao sistema de trabalho global. A

orientação é expressa por meio de ângulos náuticos. Tal como acontece com os valores de posição, os valores de orientação inválidos são destacados a vermelho.

Configuração:

Especifica a configuração do braço robótico. O utilizador pode seleccionar uma das configurações disponíveis (C1–C8) através do menu suspenso. Tenha em atenção que alterar a configuração não altera a posição ou orientação do TCP; afeta apenas os ângulos das articulações utilizados para alcançar a pose especificada.

Ações adicionais no Editor de Posições

Guardar	Ao clicar no botão «Guardar», a pose atual é guardada com o nome especificado. Não é possível guardar uma pose sem um nome válido.
Cancelar	Ao clicar no botão «Cancelar», todas as alterações não guardadas efetuadas no editor serão descartadas e a janela será fechada.
Redefinir	Ao clicar no botão «Redefinir», todas as alterações não guardadas são anuladas e a pose é reposto no seu estado mais recente guardado.
 Reproduzir	Ao clicar no botão «Reproduzir» na parte superior da janela, o robô recebe a instrução para assumir a pose definida.
Aplicar pose atual	Ao clicar no botão «Aplicar pose atual», todos os campos são atualizados de modo a refletirem os valores atuais da pose do robô.

4.6.2 Editor de trajetória

O Editor de percursos permite aos utilizadores criar um novo percurso ou alterar um já existente. Está dividido em três secções editáveis: Nome, Posição inicial e Segmentos deste percurso.

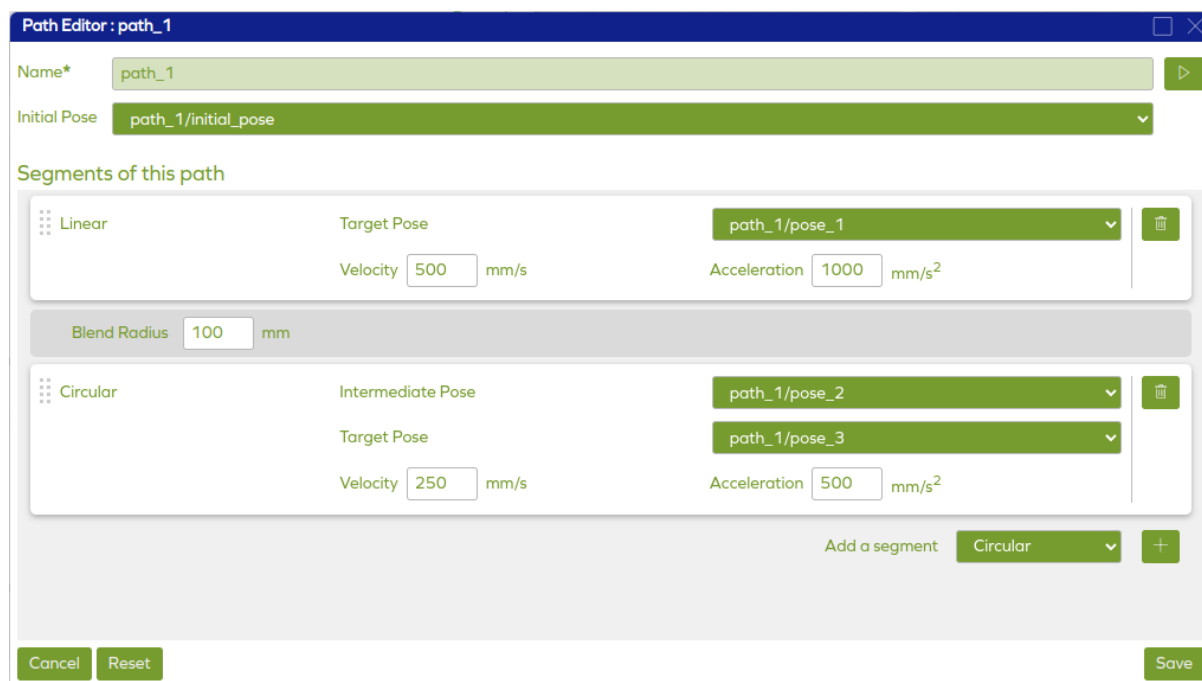


Figura 4.6.2: Editor de percursos

Nome:

Indica o nome sob o qual o percurso é guardado na base de dados. Utilize o caractere «/» no nome para agrupar os seus percursos numa estrutura semelhante a pastas.

Posição inicial:

Define a pose inicial do percurso. O robô move-se inicialmente para esta pose com as configurações padrão de velocidade e aceleração, antes de executar os segmentos definidos sucessivamente.

Segmentos deste percurso:

Enumera e define os segmentos individuais que compõem o percurso. Estão disponíveis os seguintes tipos de segmentos:

- Segmento linear
- Segmento circular
- Segmento de coordenadas
- Segmento de pose
- Segmento de orientação

Encontrará uma descrição detalhada de cada segmento na secção Segmento.

Para adicionar um novo segmento, selecione o tipo de segmento pretendido no menu suspenso no canto inferior direito e clique no botão «+». Assim que os segmentos forem adicionados, a sua ordem pode ser ajustada livremente através da função «arrastar e largar».

Cada segmento requer a especificação de uma pose de destino, que define a pose final para esse segmento. As poses de destino devem ter sido previamente definidas no Editor de Poses e podem então ser selecionadas através do menu suspenso. O segmento circular requer, de forma exclusiva, uma pose intermédia adicional, que também deve ser selecionada a partir da lista de poses previamente definidas.

Para cada segmento, podem ser definidos parâmetros de velocidade e aceleração. Se estes não forem especificados, serão utilizados os valores padrão globais durante a execução.

O raio de transição pode ser configurado para controlar a suavidade e a eficiência das transições entre segmentos. Define o desvio máximo permitido, em mm, da pose final do segmento anterior, que corresponde à pose inicial do segmento atual.

Ações adicionais no Editor de Percursos

O Editor de percursos oferece ainda funcionalidades adicionais para gerir, editar e organizar o percurso de forma eficaz.

Guardar	Ao clicar no botão «Guardar», o caminho atual é guardado com o nome especificado. Não é possível guardar um caminho sem um nome válido.
Cancelar	Ao clicar no botão «Cancelar», todas as alterações não guardadas que foram feitas no editor são descartadas e a janela é fechada.
Redefinir	Ao clicar no botão «Redefinir», todas as alterações não guardadas são anuladas e o caminho é reposto no seu estado mais recente guardado.
 Reproduzir	Ao clicar no botão «Reproduzir» na parte superior da janela, o robô recebe a instrução para executar o percurso definido. Se a pose atual do robô não corresponder à pose inicial especificada para o percurso, ele deslocar-se-á primeiro para essa pose inicial antes de iniciar a execução.
 Apagar	Ao clicar no botão «Apagar» ao lado de um segmento, esse segmento é removido do percurso.

4.6.3 Editor de grelhas

Ao criar uma nova grelha, abre-se o Editor de Grelhas. Este editor orienta o utilizador, passo a passo, através de todas as opções de grelha disponíveis. Utilize os botões «Seguinte» e «Voltar» para navegar neste assistente até que a lista ordenada de pontos da grelha esteja definida conforme desejado. Ao editar grelhas já existentes, o Editor de Grelhas salta diretamente para a edição manual de pontos.

Num primeiro passo, o utilizador deve selecionar a dimensão da grelha. Normalmente, uma grelha é interpretada como uma matriz bidimensional, mas também pode ser uma simples lista de pontos sem dependência de uma segunda dimensão (por exemplo, para selecionar objetos de uma correia transportadora) ou mesmo uma estrutura tridimensional (por exemplo, para empilhar objetos em caixas). A grelha é inicializada com pontos perfeitamente alinhados, devendo ter-se em conta que os pontos individuais da grelha podem ser alterados, adicionados ou removidos posteriormente a qualquer momento.

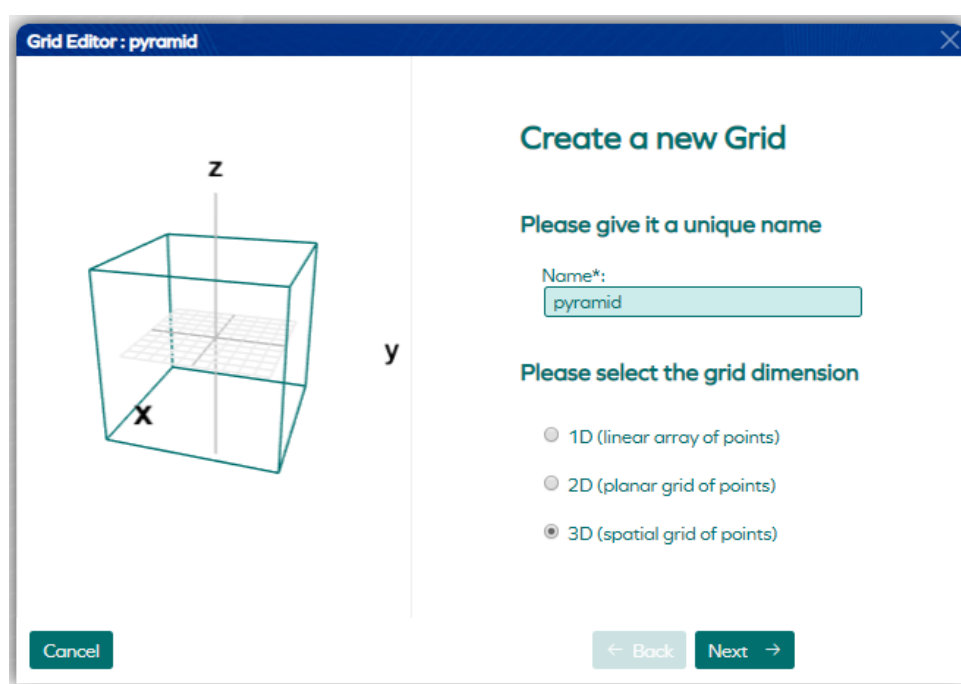


Figura 4.6.3.a: Editor de grelha (1.º passo)

Numa segunda etapa, o utilizador deve determinar os vértices da grelha selecionada. Grelhas de diferentes dimensões têm um número diferente de vértices, e os vértices são designados com «A» até, por exemplo, «D» para uma grelha bidimensional. O utilizador pode introduzir estes valores de coordenadas (especificados no sistema de referência do robô) diretamente nos campos de introdução ou simplesmente conduzir o TCP do efector final até aos vértices da grelha e premir os botões «Registar» correspondentes para ler as coordenadas atuais.

Assim que forem apreendidos os pontos de canto suficientes para definir a grelha (por exemplo, pelo menos 3 pontos para a grelha bidimensional), o utilizador pode carregar o botão «Avaliar valores» para avaliar os pontos de canto indicados e preencher automaticamente os valores em falta, de modo a que se encaixem num paralelogramo (bidimensional) ou num paralelepípedo (tridimensional). Se, além disso, a opção «Alinhamento automático» estiver ativada, o procedimento « » (Alinhar a malha)

tem em conta os eixos e planos de coordenadas horizontais e verticais e tenta alinhar a malha resultante ao longo desses eixos e planos (desde que suficientemente próximos).

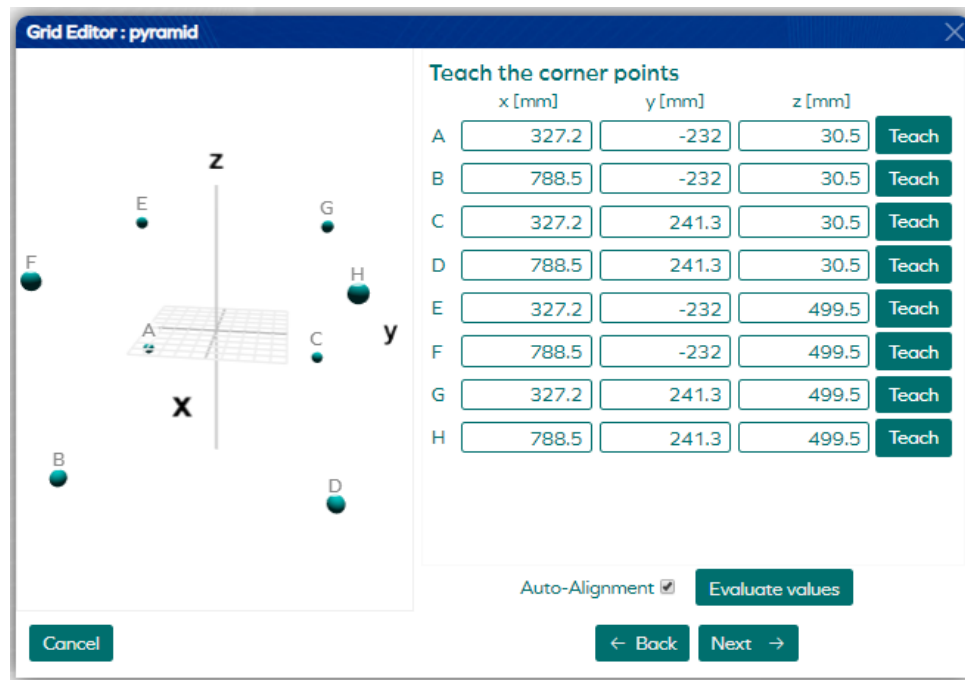


Figura 4.6.3.b: Editor de grelha (2.^a etapa)

Numa terceira etapa, o utilizador é solicitado a seleccionar o número de pontos, opções de expansão específicas e direcções de iteração. Vejamos estas opções disponíveis em detalhe:

- O número de pontos pode ser definido individualmente para cada dimensão. Caso tenha dificuldade em distinguir as dimensões (por exemplo, linhas e colunas), consulte as legendas dos vértices previamente definidos.
- Para grelhas bidimensionais, existe uma opção de extensão para adicionar pontos intermédios entre cada grupo de quatro pontos adjacentes. Para grelhas tridimensionais, esta opção existe tanto para cada plano da grelha como para o grupo de oito pontos adjacentes no espaço.
- A direcção de iteração, ou seja, a direcção na qual os pontos são devolvidos e utilizados como iteradores em loops de programação, é especificada por um ponto inicial e um vetor em cada direcção. A última direcção é sempre determinada pelas opções seleccionadas anteriormente e é exibida apenas para referência e verificação visual. As letras nos menus suspensos correspondem às legendas dos vértices desta grelha.

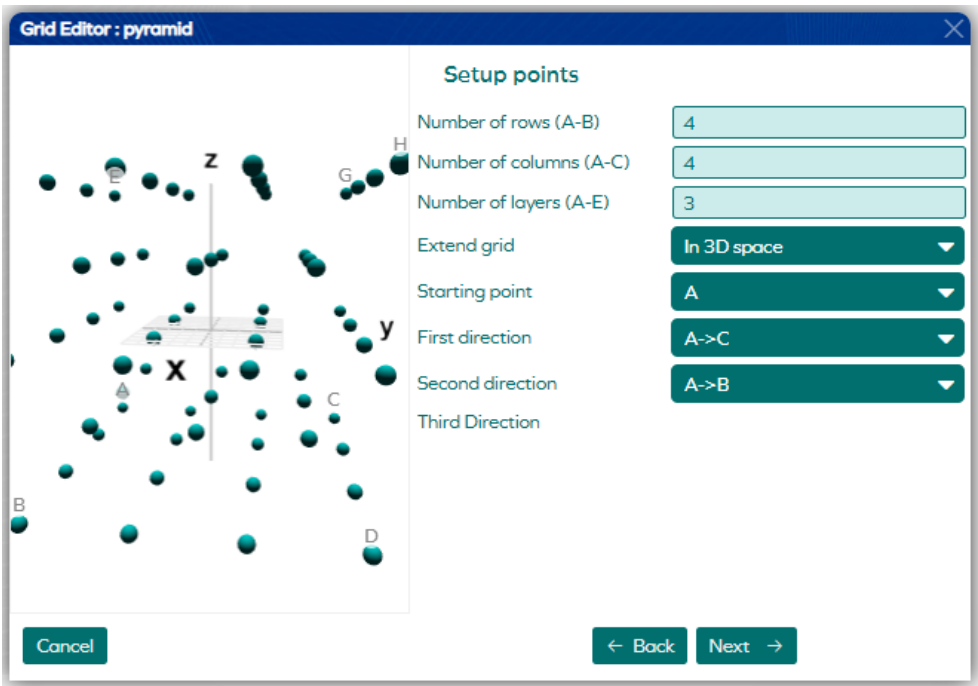


Figura 4.6.3.c: Editor de grelha (3.º passo)

Por fim, a lista de pontos da grelha fica disponível para edição manual. É possível copiar, eliminar ou deslocar pontos individuais, bem como alterar manualmente os valores das coordenadas. Simultaneamente, a visualização apresenta a ordem dos pontos e as coordenadas relativas dos pontos da grelha.

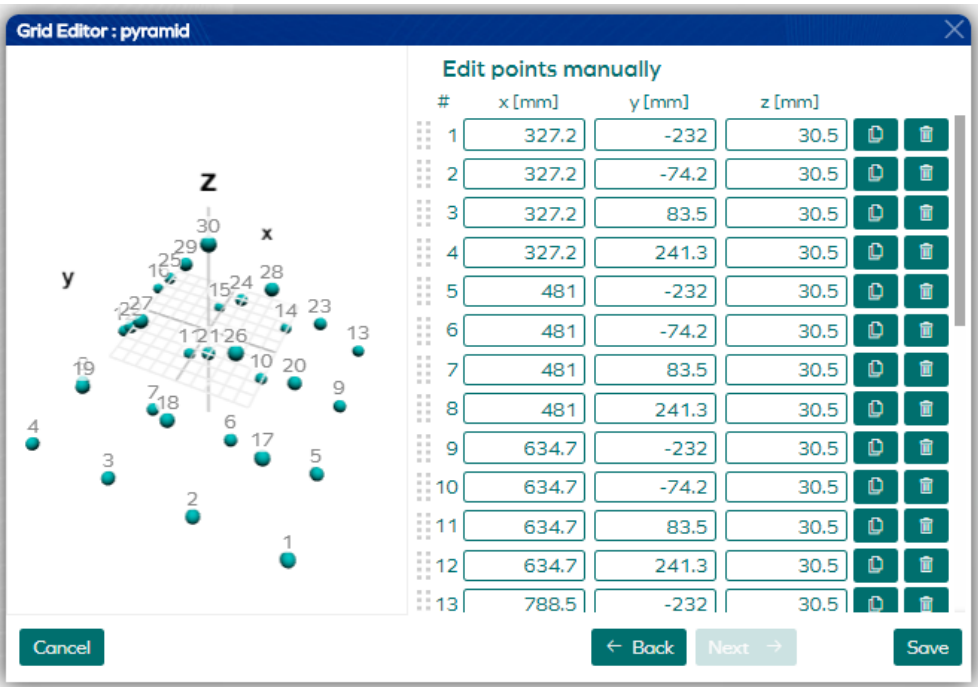


Figura 4.6.3.d: Editor de grelha (4.º passo)

Durante a edição, pode voltar a qualquer momento para verificar as opções e valores definidos anteriormente. Também pode alterar estas opções, mas isso irá repor todas as alterações nas janelas seguintes.

4.7 Menu de aplicações

Clique no botão «Aplicações» na janela principal para aceder ao menu de aplicações. Este menu contém uma visão geral de todas as aplicações guardadas, bem como a saída da última aplicação executada. Permite ainda ao utilizador aceder à janela «Editor de aplicações» para criar novas aplicações ou editar as existentes.

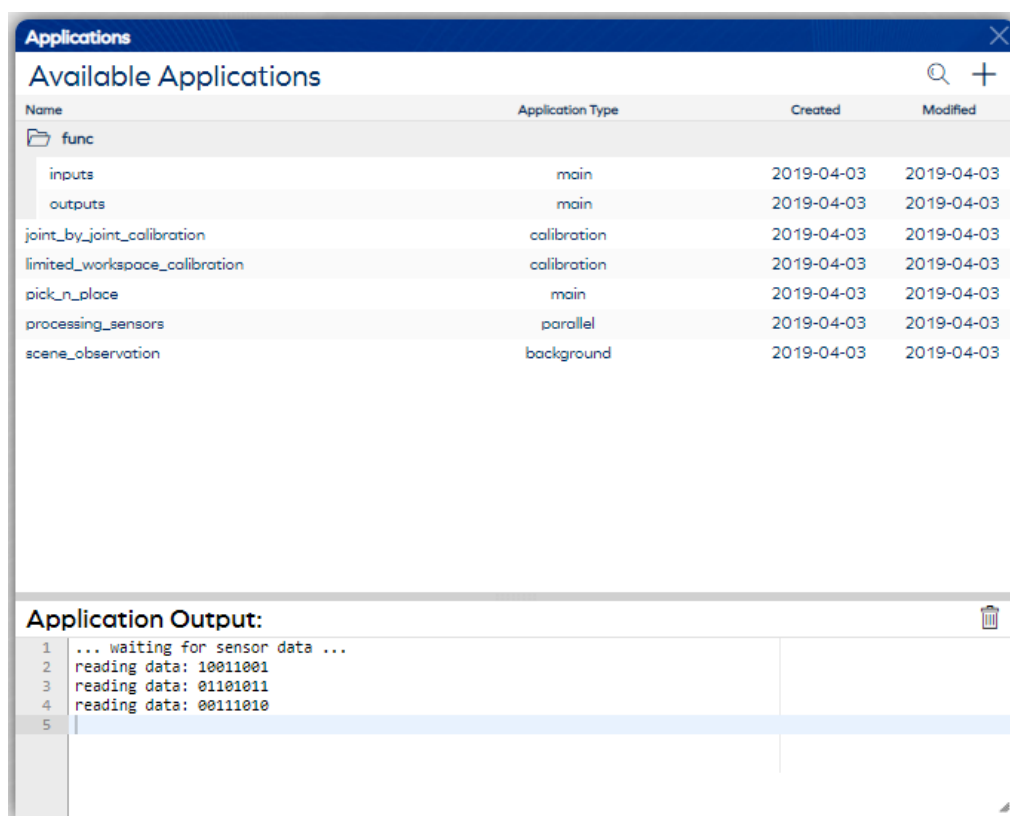


Figura 4.7: Menu de aplicações

	Clique no botão «Novo/Adicionar» para criar uma nova aplicação. Isto abre o Editor de Aplicações, cujas funções são explicadas em pormenor mais adiante nesta secção.
	Clique no botão «Filtrar/Pesquisar» para filtrar a lista de aplicações por nome. A introdução de três ou mais caracteres ativa o filtro de pesquisa.
	Clique no botão «Executar» para executar uma aplicação. Tenha em atenção que, desta forma, apenas as aplicações principais podem ser executadas, uma vez que no Cockpit apenas é possível selecionar aplicações principais.
	Clique no botão «Copiar» para criar uma cópia exata desta aplicação. O nome do elemento copiado é ligeiramente alterado e termina com «(cópia)».
	Clique no botão «Editar» (ou clique duas vezes em qualquer ponto do elemento) para abrir esta aplicação numa janela separada para visualização ou edição.
	Clique no botão «Eliminar» para remover irrevogavelmente uma aplicação da sua base de dados. Isto remove tanto o código do programa como as definições específicas da aplicação.



Clique numa pasta para mostrar ou ocultar as aplicações dentro dessa pasta. Os elementos só são apresentados em pastas (e subpastas) se os seus nomes contiverem um ou mais caracteres «/», por exemplo, «folder/subfolder/name». Pode clicar no botão «Editar» que aparece para abrir o editor de pastas.

4.7.1 Editor de aplicações

Na secção «Aplicações disponíveis» do menu de aplicações, são listadas todas as aplicações previamente guardadas. Pode alterar uma aplicação existente clicando no botão «Editar» ou criar uma nova aplicação clicando no botão «Criar nova aplicação». Em ambos os casos, abre-se a janela «Editor de aplicações». O seu funcionamento é semelhante em ambos os casos. Vamos dar uma breve olhadela aos botões mais importantes do editor de aplicações antes de entrarmos em pormenores:

Guardar	Ao clicar no botão «Guardar», a aplicação é guardada com o nome indicado. Não é possível guardar uma aplicação sem nome. Utilize o sinal «/» no nome para agrupar as suas aplicações numa estrutura semelhante a pastas.
Cancelar	Ao clicar no botão «Cancelar», todas as alterações não guardadas efetuadas neste editor serão descartadas e o editor será fechado.
 Testar	Ao clicar no botão «Testar aplicação» na parte superior da janela, a aplicação é executada no modo de teste. O modo de teste significa que o código do programa é executado, mas não o impede de continuar a editar a aplicação. As alterações à aplicação adicionadas após o teste não são executadas.

4.7.2 Código do programa

O separador «Código» da aplicação é o mais importante das sub-janelas do editor de aplicações. Esta área apresenta o código do programa e, opcionalmente, a biblioteca com funções e elementos do AX Portal que podem ser inseridos no código do programa. A linguagem de programação integrada é o Python 3.13, com a maior parte das suas funcionalidades básicas, incluindo alguns pacotes Python pré-instalados. Além disso, o AX Portal oferece uma vasta biblioteca de funções que simplificam a escrita de uma aplicação robótica (por exemplo, movimentação do robô, leitura de dados de sensores, comunicação com o utilizador, etc.). Encontrará detalhes sobre estas funções na documentação do código.

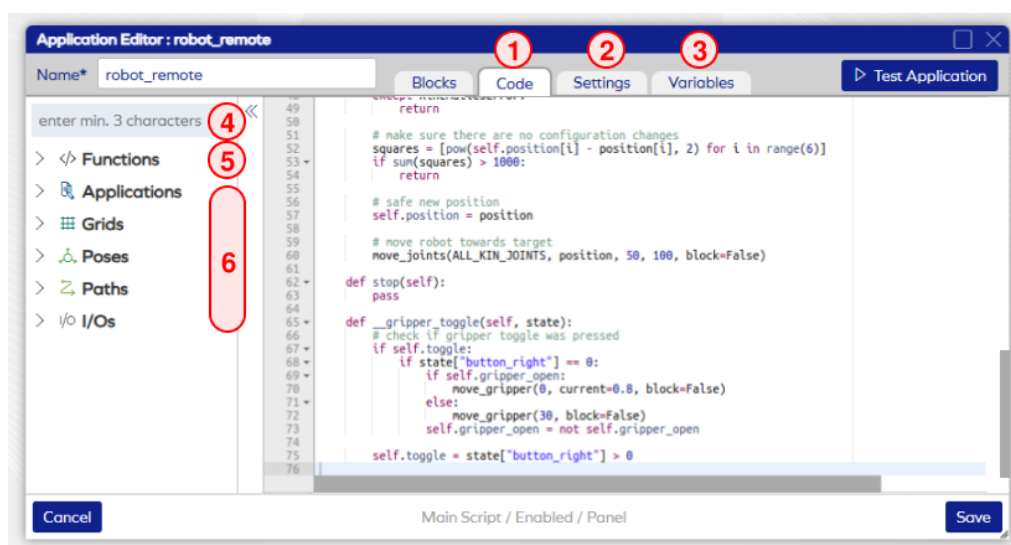


Figura 4.7.2: Editor de aplicações (área de código)

1. Escreva aqui o seu código de programação Python,
2. Alterar definições específicas da aplicação,
3. Definir e alterar variáveis para a aplicação,
4. Filtro de pesquisa,
5. Biblioteca de funções,
6. Elementos disponíveis para utilização no programa

Clique num dos ícones de função ou num dos ícones de elemento no lado esquerdo do editor para abrir a paleta de elementos. Aqui encontrará uma lista completa das funções disponíveis. Além disso, tem acesso a todos os elementos que pode inserir no seu código com um duplo clique, pelo que não precisa de se lembrar dos nomes dos seus elementos.

Dica: Aumente a legibilidade do código Python removendo os argumentos opcionais que são inseridos no seu código ao clicar duas vezes numa função. Pode, por exemplo, escrever

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

em vez de

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

4.7.3 Configurações da aplicação

O separador «Definições» contém algumas definições adicionais para a aplicação, incluindo

- um botão «Executável», que pode ser desativado, por exemplo, se a sua aplicação ainda estiver em desenvolvimento e não deva ser executada,
- um botão para exibir esta aplicação na interface do painel,
- funcionalidade para carregar e substituir a imagem exibida na interface do painel,
- títulos personalizados como forma de exibir o nome de exibição na interface do painel no idioma da interface atualmente selecionado,
- e, acima de tudo, o tipo de aplicação, que é explicado na secção seguinte:

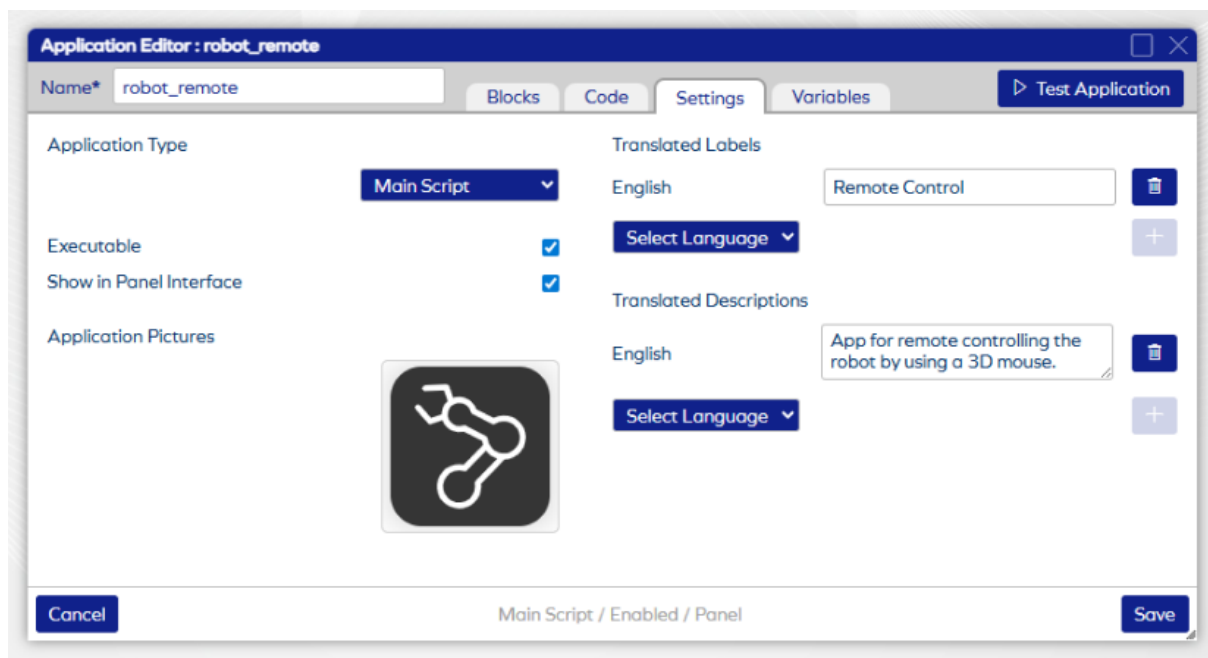


Figura 4.7.3.a: Editor de aplicações (área de configurações)

O tipo de aplicação define em que contexto a aplicação deve ser executada e, por conseguinte, determina como uma aplicação pode ser executada, bem como quais as funções que podem ser utilizadas dentro dessa aplicação:

- As aplicações principais são as aplicações normais que devem ser executadas diretamente. Estão concebidas para suportar todas as funções disponíveis e controlar diretamente o robô, bem como os seus dispositivos externos. Normalmente, uma aplicação simples pode ser escrita num único programa; no entanto, para aplicações longas ou complexas, ou para melhorar a legibilidade, pode fazer sentido dividir uma aplicação em vários programas que são executados sucessivamente. No entanto, várias aplicações principais nunca são executadas em paralelo; para esse efeito, são utilizadas «aplicações paralelas». Também pode utilizar a instrução «import» para importar funções entre aplicações principais. Tenha em atenção que apenas as aplicações principais podem ser selecionadas através do Cockpit.
- **As aplicações paralelas**, por outro lado, são executadas em threads separadas e, por isso, foram concebidas para serem executadas em paralelo com uma aplicação principal. Isto pode ser muito útil, por exemplo, na leitura contínua de dados de sensores ou para aproveitar o tempo de inatividade do robô para cálculos que exigem grande capacidade de processamento. As aplicações paralelas não suportam todas as funções, como, por exemplo, o movimento do robô, uma vez que isso complicaria significativamente o fluxo do programa ou poderia provocar comportamentos de movimento indesejados ou mesmo perigosos em caso de falta de atenção. Todas as aplicações paralelas são pausadas enquanto a aplicação principal está em pausa e são encerradas assim que a aplicação principal é interrompida.
- **As aplicações em segundo plano** são aplicações que funcionam no segundo plano do AX Portal, independentemente do estado do robô, e são, por isso, úteis, por exemplo, para monitorizar, verificar ou registar dados de forma contínua, ou para definir variáveis globais iniciais para um projeto. Cada projeto suporta apenas uma aplicação em segundo plano. Assim que esta for definida, é executada automaticamente quando o projeto correspondente é carregado (por exemplo, durante a inicialização do robô ou ao mudar de projeto). A aplicação em segundo plano não está sujeita às restrições de estado e pode – caso contenha um ciclo – continuar a funcionar até que o robô seja desativado. Sempre que uma aplicação em segundo plano é iniciada e terminada, é apresentada uma notificação correspondente. Pode iniciar, reiniciar ou parar a aplicação em segundo plano definida a partir do Cockpit.



Figura 4.7.3.b: Elementos de controlo para a aplicação em segundo plano

4.7.4 Variáveis da aplicação

O separador «Variáveis» da aplicação contém uma interface de arrastar e largar para adicionar vários tipos de variáveis de estado a esta aplicação. As variáveis de estado só podem ser utilizadas em aplicações principais. São utilizadas na interface do painel, onde são, por exemplo, muito úteis para monitorizar o estado e o progresso de uma aplicação ou para especificar determinados valores de entrada para uma aplicação. Utilize as funções `set_state_variable(...)` e `get_state_variable(...)` para escrever e ler variáveis de estado.

Segue-se uma visão geral dos tipos de variáveis e da forma como são apresentadas na interface do painel:

boolean	Os booleanos são apresentados como simples controles deslizantes, que estão ativados ou desativados. São lidos e gravados como o tipo Python bool , sendo que ativado corresponde a True e desativado a False .
integer	Os valores inteiros são apresentados como campos de texto que permitem entradas numéricas, incluindo botões de mais e menos para aumentar e diminuir os seus valores no modo de edição. São lidos e gravados como o tipo Python int .
float	Os números de ponto flutuante são apresentados como campos de texto que permitem a introdução de valores numéricos no modo de edição. São lidos e gravados como o tipo float da Python.
slider	Os números de ponto flutuante são apresentados no modo de edição como controles deslizantes com um valor mínimo e máximo obrigatórios. São lidos e gravados como o tipo float do Python.
text	Os textos são apresentados no modo de edição como campos de texto. São lidos e gravados como o tipo Python str .
dropdown	As variáveis com seleção específica são apresentadas no modo de edição como listas suspensas. São lidas e gravadas como o tipo Python str .

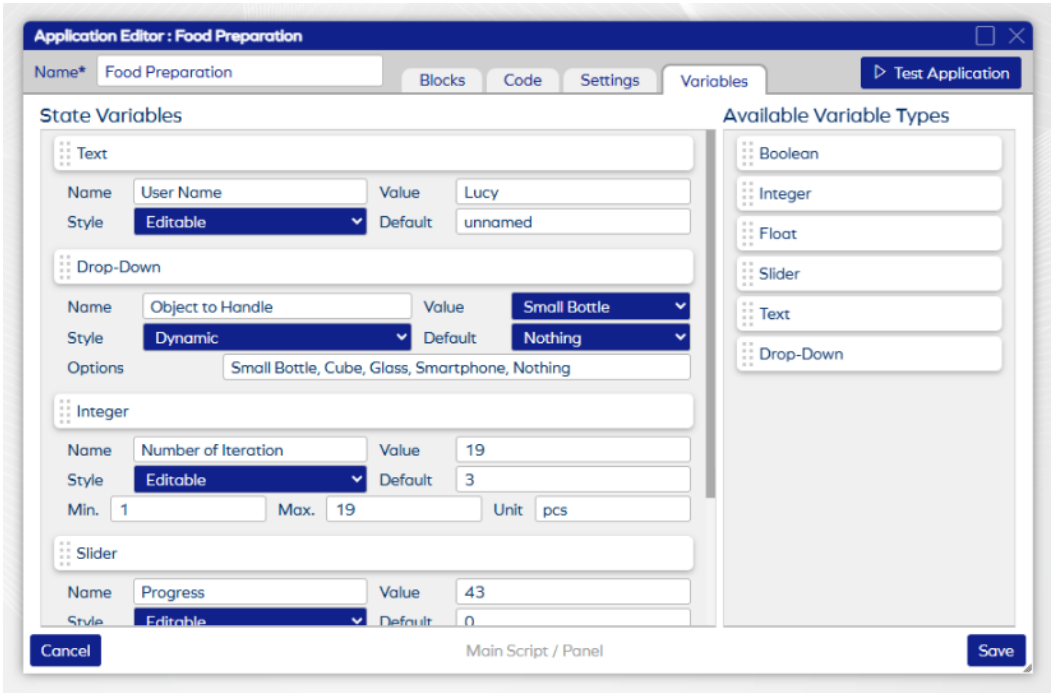


Figura 4.7.4: Editor de aplicações (área de variáveis)

Aqui está uma visão geral das opções, que variam consoante o tipo de variável:

Nome	Atribua um nome adequado à sua variável; esta ficará acessível posteriormente sob esse nome, na forma de uma cadeia de caracteres.
Valor	O valor de uma variável contém o valor atualmente definido para essa variável, que é lido e gravado no tipo Python correspondente, conforme explicado acima.
Padrão	O valor padrão de uma variável é o valor que esta assume quando as variáveis são reiniciadas através do botão correspondente na interface do painel.
Estilo	O estilo de uma variável define se, quando e como essa variável pode ser editada.
Mín.	As variáveis numéricas permitem definir um limite inferior para esse número. Se introduzir valores inferiores ao mínimo, o valor mínimo será aplicado. Os limites são obrigatórios para os controlos deslizantes e opcionais para outras variáveis.
Máx.	As variáveis numéricas permitem definir um limite superior para esse número. Se introduzir valores acima do máximo, será aplicado o valor máximo. Os limites são obrigatórios para os controlos deslizantes e opcionais para outras variáveis.
Unidade	No caso de variáveis numéricas, pode, opcionalmente, adicionar uma unidade ao número sob a forma de uma cadeia de caracteres.
Opções	As opções para variáveis de lista suspensa podem ser introduzidas como uma cadeia de caracteres separada por vírgulas.

Aqui está uma visão geral dos estilos de variáveis, que definem como uma variável é utilizada:

- **As variáveis editáveis** permitem alterar o valor antes da execução da aplicação, mas não durante a execução.
- **As variáveis protegidas** são editáveis, mas é necessário premir uma tecla de edição adicional para alterar o valor e uma tecla de gravação adicional para aplicar a alteração do valor.
- **As variáveis dinâmicas** são editáveis e pode mesmo alterar os seus valores dinamicamente durante a execução.
- **As variáveis dinâmicas protegidas** combinam as características das variáveis protegidas e dinâmicas e, por isso, são editáveis durante a execução premindo uma tecla de edição.
- **As variáveis de leitura apenas** não são editáveis; o seu valor só pode ser alterado pela aplicação. São normalmente utilizadas para monitorizar o estado ou o progresso de uma aplicação.
- **As variáveis ocultas** não são exibidas na interface do painel, mas podem ser gravadas e lidas da mesma forma que outras variáveis de estado.

4.7.5 Variáveis partilhadas e globais

Para além das variáveis de estado específicas da aplicação acima descritas, podem ser definidas variáveis partilhadas e globais para partilhar dados entre vários scripts em execução paralela ou entre várias sessões da mesma aplicação:

- **As variáveis partilhadas** existem apenas durante o tempo de execução de uma única aplicação. Utilize as funções `set_shared_variable(...)` e `get_shared_variable(...)` para partilhar valores entre várias aplicações principais e/ou aplicações paralelas.
- **As variáveis globais** não só partilham valores entre aplicações em execução, como também permanecem após o reinício do robô. Para tal, utilize as funções `set_global_variable(...)` e `get_global_variable(...)`.

4.7.6 Editor de pastas

Se o nome da aplicação contiver o caractere «/», a aplicação será automaticamente classificada numa estrutura de pastas. Pode editar uma pasta clicando no botão de edição que aparece ao passar o cursor sobre essa pasta. Isto abre o Editor de Pastas. Este editor oferece a possibilidade de exibir o nome da pasta no idioma da interface selecionada (semelhante ao Editor de Aplicações). Os títulos das pastas, bem como os títulos das aplicações, aparecem na interface do painel; desta forma, a interface do painel pode ser facilmente configurada de modo a ser compreensível em vários idiomas.

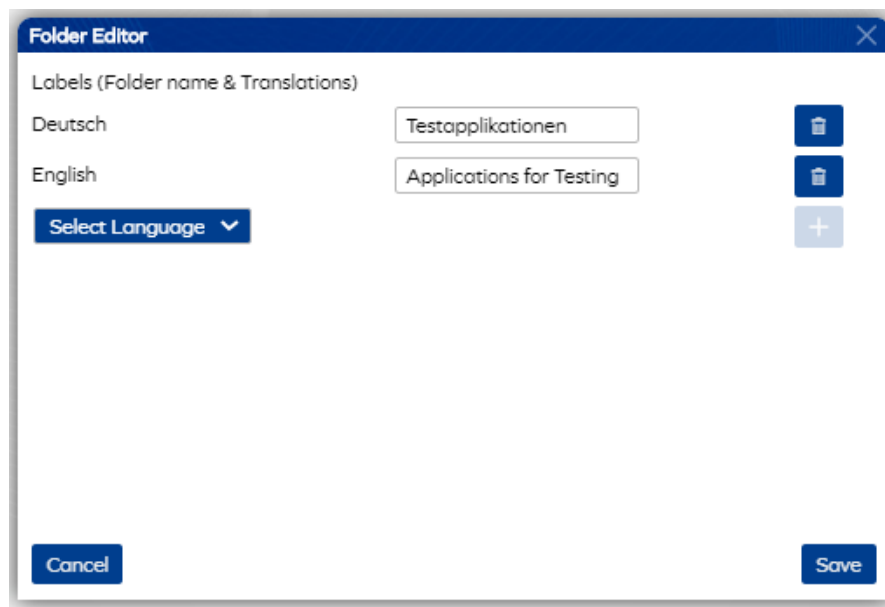


Figura 4.7.6: Editor de pastas

4.7.7 Programação por blocos

O separador «Blocos» do Editor de Aplicações contém uma interface de programação gráfica. Esta interface permite ao utilizador criar sequências de programação simples, arrastando e soltando blocos de programação na área de programação.

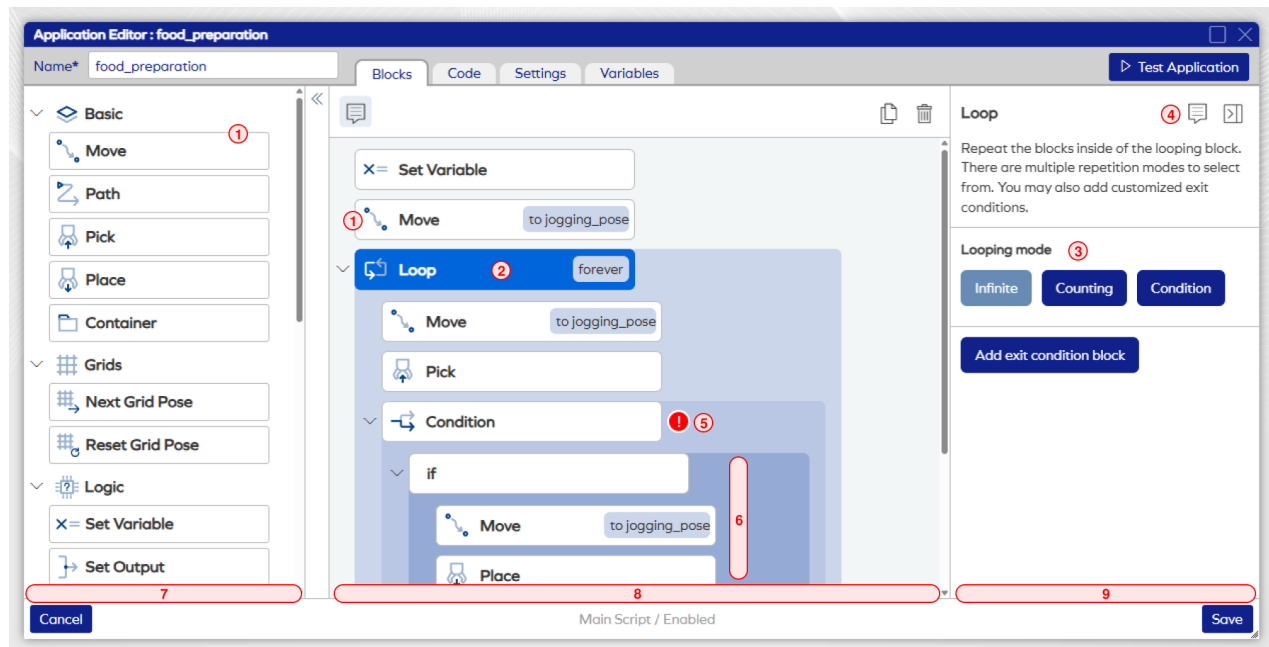


Figura 4.7.7: Editor de aplicações (programação por blocos)

1. Arrastar blocos para a área de programação, 2. Selecionar o bloco para edição, 3. Alterar parâmetros, 4. Comentário, 5. Indicação de erros, 6. Recipiente de blocos, 7. Blocos disponíveis, 8. Área de programação, 9. Detalhes do bloco selecionado

Selecione um bloco no fluxo do programa para ver os seus detalhes. Alguns parâmetros são obrigatórios para que o programa seja executável (por exemplo, num bloco de movimento, deve fornecer informações sobre para onde o robô deve se mexer). Outros parâmetros, por sua vez, são opcionais (por exemplo, são utilizados valores padrão caso não sejam especificados) ou podem ser utilizados para melhorar a legibilidade do programa (por exemplo, comentários). Se um bloco não estiver definido corretamente (por exemplo, se faltar um parâmetro obrigatório ou se tiver sido especificado um valor inválido), ele exibirá uma mensagem de erro no fluxo do programa e, ao ser selecionado, o motivo do erro será exibido nos detalhes.

Alguns blocos, ao serem adicionados, criam contêntores colorido no fluxo do programa. Trata-se de uma ajuda gráfica para indicar que esses blocos estão agrupados e são sempre executados na ordem (de cima para baixo). Os loops e as condições são exemplos desses blocos, responsáveis, por exemplo, por executar um grupo de blocos várias vezes ou por executar um grupo apenas quando uma determinada condição se verifica.

Além disso, um bloco selecionado pode ser eliminado ou copiado (e arrastado para o local desejado no programa).

4.7.7.1 Utilização de variáveis de aplicação e expressões

Antes de explicar em detalhe todos os blocos de programação, tenha em atenção que a maioria dos campos de texto dos parâmetros dos blocos suporta expressões matemáticas e/ou lógicas. Estes campos também suportam a inserção de valores de variáveis de aplicação predefinidas, bem como de valores de entradas digitais e analógicas.

A forma mais simples de explicar isto é através de um exemplo. Vamos assumir que definimos as seguintes variáveis:

The screenshot shows the 'State Variables' configuration interface. It contains two sections: 'Boolean' and 'Integer'.
 In the 'Boolean' section, there is a variable named 'flag' with a style of 'Editable' and a default value of 0.
 In the 'Integer' section, there is a variable named 'counter' with a style of 'Editable', a minimum value of 0, a maximum value of 1000, and no unit. Its default value is also 0.

Figura 4.7.7.1.a: Variáveis de exemplo

Exemplos de **expressões numéricas** válidas:

Four examples of valid numeric expressions are shown in a text field:
 1. 42
 2. -7 * (8.2 - 3)
 3. {counter} + 1
 4. {Tool AIO 02}

Exemplos de **expressões lógicas** válidas:

Four examples of valid logical expressions are shown in a text field:
 1. FALSE
 2. NOT {flag}
 3. {flag} AND ({counter} > 5)
 4. {Main DIO 09}

Figura 4.7.7.1.b: Exemplo de expressões numéricas e lógicas

Os valores das variáveis e entradas também podem ser inseridos em campos de texto normais:

An example of a text expression is shown in a text field: "The counter is set to {counter}."

Figura 4.7.7.1.c: Exemplo de uma expressão de texto

4.7.7.2 Bloco de movimento

O **bloco de movimento** executa uma função de movimento que desloca o robô da sua posição atual para um destino. O utilizador deve seleccionar o estilo de movimento ...

- **Ponto a ponto** → Deslocar-se para o destino da forma mais rápida possível, sem restrições quanto ao percurso
- **Linear** → Deslocar-se para o destino ao longo de um percurso linear, sendo a orientação interpolada linearmente

- **Circular** → Deslocar-se para o destino ao longo de um percurso circular, sendo a orientação interpolada linearmente, e o utilizador deve especificar **pose intermédia** no percurso circular

... e o formato do destino.

- **Nome da pose** → Selecione o nome da pose de destino (previamente guardada)
- **Nome da grelha** → Deslocar-se para os pontos de uma grelha (previamente guardada); inicialmente, deslocar-se-á para o primeiro ponto desta grelha; em seguida, pode o utilizar-se **bloco «Próxima pose da grelha»** para aumentar o contador de pontos da grelha
- **Ângulo da articulação** → Especifique os ângulos das articulações da pose de destino
- **Coordenadas** → Indique as coordenadas da pose de destino

Além disso, o utilizador pode alterar as seguintes definições opcionais. Tenha em atenção que, caso não sejam especificadas, serão utilizados valores padrão/globais.

- **Definições de velocidade**
 - **Velocidade/aceleração da articulação** → Indique a velocidade para movimentos ponto a ponto
 - **Velocidade linear / aceleração** → Indique a velocidade para movimentos lineares/circulares
 - **Velocidade/aceleração de rotação** → Indique a velocidade para movimentos que alteram apenas a orientação do TCP (a posição permanece a mesma)
- **Definições do sistema de coordenadas de referência**
 - **Desvio da ferramenta** → Desloca o TCP em relação ao sistema de flange (parte superior do braço do robô)
 - **Sistema de coordenadas da ferramenta** → Altera o TCP em relação ao sistema de flange (parte superior do braço do robô)
 - **Desvio de trabalho** → Desloca o sistema de referência das coordenadas em relação ao sistema de base (parte inferior do braço do robô)
 - **Sistema de coordenadas de trabalho** → Altera o sistema de referência das coordenadas em relação ao sistema de base (parte inferior do braço do robô)
- **Outras configurações**
 - **Coordenadas relativas** → Quando esta opção está ativada, os valores das coordenadas são somados às coordenadas atuais
 - **Fluxo do programa bloqueado** → Quando esta opção está ativada, o fluxo do programa só continua após a conclusão do movimento

4.7.7.3 Bloco de trajetória

O **bloco de trajetória** reproduz uma trajetória previamente definida. O utilizador seleciona a trajetória a executar. Ao executar este bloco, o robô desloca-se primeiro para a posição inicial da trajetória e percorre depois os segmentos da trajetória num movimento contínuo.

4.7.7.4 Bloco «Levantar»

O **bloco «Suspend»** envia um comando de suspensão à ferramenta conectada. No caso de pinças angulares ou paralelas, isto significa fechar a pinça para manter o objeto no seu interior. No caso de pinças pneumáticas, isto significa ativar a bomba de vácuo para manter o objeto no seu lugar. Este bloco não possui parâmetros.

4.7.7.5 Bloco Colocar

O **bloco Colocar** envia um comando de colocação à ferramenta conectada. No caso de garras angulares ou paralelas, isso significa abrir a garra para liberar o objeto que se encontra no seu interior. No caso de garras pneumáticas, isso significa desativar a bomba de vácuo para depositar o objeto no seu lugar. Este bloco não possui parâmetros.

4.7.7.6 Bloco «Contentor»

O **bloco «Container»** oferece uma forma simples de agrupar vários blocos. Não lhe está atribuída nenhuma função específica.

4.7.7.7 Bloco Posição seguinte da grelha

O **bloco «Próxima Posição da Grelha»** aumenta o contador interno de posições para uma determinada grelha. A grelha em questão deve ser selecionada. Sempre que um **bloco de movimento** é utilizado para se deslocar para uma grelha, é visada a pose para a qual este contador aponta atualmente. Para percorrer efetivamente todos os pontos de uma grelha, pode utilizar tanto o **bloco de movimento** como o **bloco de pose da grelha seguinte** dentro de uma estrutura de loop da seguinte forma:

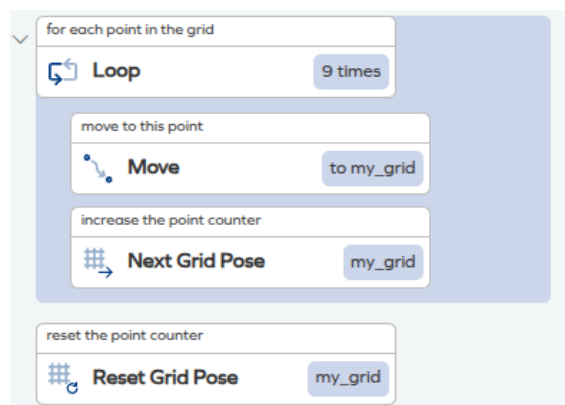


Figura 4.7.7.7: Exemplo de utilização de blocos de grelha

4.7.7.8 Bloco «Redefinir Posição da Grelha»

O **bloco «Redefinir Posição da Grelha»** redefine o contador interno de posições de uma determinada grelha para zero (valor inicial). A grelha em questão deve ser selecionada.

4.7.7.9 Bloco «Escrever variável»

O **bloco «Escrever variável»** define o valor de uma variável de aplicação predefinida. É obrigatório selecionar primeiro uma variável e, em seguida, um valor. É permitido o uso de expressões numéricas e/ou lógicas.

4.7.7.10 Bloco Definir Saída

O **bloco Definir Saída** define o valor de uma saída digital ou analógica. O utilizador deve primeiro selecionar a saída a ser definida; dependendo de se tratar de uma saída digital ou analógica, o valor a ser definido é um valor boolean ou um número.

4.7.7.11 Bloco Esperar

O **bloco de espera** bloqueia o fluxo do programa até que ocorra um determinado evento. Estão disponíveis três modos diferentes:

- **Tempo** → Esperar até que um determinado período de tempo tenha decorrido (expressão numérica permitida)
- **Movimento** → Aguardar até que um comando de movimento em execução pare
- **Entrada** → Esperar até que o valor de uma entrada digital seja definido para um valor específico ou até que o valor de uma entrada analógica ultrapasse um determinado valor limite

4.7.7.12 Bloco de loop

O **bloco de loop** cria um grupo de blocos que é chamado repetidamente. Estão disponíveis três modos diferentes:

- **Infinito** → Loja que continua até que a aplicação seja interrompida
- **Contador** → Loja para um determinado número de repetições (expressão numérica permitida)
- **Condição** → Loop até que uma determinada condição seja satisfeita (expressão lógica permitida)

O utilizador pode adicionar um **bloco «Sair se»** ao ciclo; consulte a secção seguinte.

4.7.7.13 Bloco «Sair se»

O **bloco «Sair se»** está ligado ao seu **bloco de loop** e sai desse loop quando a condição especificada é verdadeira (expressão lógica permitida).

4.7.7.14 Bloco de condição

O **bloco de condição** cria um contêiner com dois grupos, “if” e “else”, no seu interior. A condição (expressão lógica permitida) é utilizada para direcionar o fluxo do programa para o grupo “if” (caso a condição seja verdadeira) ou para o grupo “else” (caso a condição não seja verdadeira).

4.7.7.15 Bloco de comentário

O **bloco de comentário** insere simplesmente um comentário informativo no fluxo do programa. Tenha em atenção que qualquer outro bloco também oferece a possibilidade de adicionar um comentário, que fica então visível na área do programa. Por vezes, porém, faz sentido criar um comentário separado, que não esteja vinculado a um bloco específico.

4.7.7.16 Bloco «Terminar»

O **bloco «Terminar»** interrompe o programa atualmente em execução. Os blocos de programação restantes não serão mais executados.

4.7.7.17 Bloco de diálogo

O **bloco de diálogo** cria um contentor com vários grupos no seu interior. Durante a execução, o fluxo do programa é interrompido e surge uma caixa de diálogo. O utilizador tem de responder à caixa de diálogo para que o fluxo do programa continue. Os grupos no interior (por predefinição, «OK» e

«Cancelar») correspondem aos botões na caixa de diálogo. Dependendo do botão pressionado, o fluxo do programa ramifica-se e prossegue com o grupo selecionado. A mensagem da caixa de diálogo (injeção de variáveis permitida), os nomes dos botões e o número de botões podem ser alterados nos parâmetros.

4.7.7.18 Bloco de saída

O **bloco de saída** apresenta uma mensagem definida pelo utilizador na saída de texto.

4.7.7.19 Bloco de registo

O **bloco de registo** regista uma mensagem definida pelo utilizador num ficheiro de registo definido pelo utilizador. Os ficheiros de registo podem ser descarregados através da interface do ambiente de trabalho, por exemplo, com a combinação de teclas CTRL + SHIFT + L.

4.7.7.20 Bloco de Notificação

O **bloco de notificação** envia uma notificação personalizada para a GUI. A notificação não bloqueia o fluxo do programa.

4.7.7.21 Bloco de aviso

O **bloco de aviso** envia uma mensagem de aviso definida pelo utilizador para a GUI. O aviso não bloqueia a execução do programa.

4.7.7.22 Bloco de erro

O **bloco de erro** envia uma mensagem de erro definida pelo utilizador para a GUI e interrompe o programa atualmente em execução.

4.7.7.23 Bloco de Código Python

O **bloco de código Python** é um bloco avançado que permite ao utilizador chamar algumas linhas de código Python independente. Pode-se utilizar este bloco para chamar funções que, de outra forma, não estariam disponíveis através da programação por blocos. No seu interior, podem ser utilizadas todas as funções.

4.8 Menu I/Os

Clique no botão «E/Ss» na janela principal para aceder ao menu I/Os. Este menu apresenta todas as entradas e saídas digitais e analógicas, bem como as suas propriedades, que foram previamente definidas na janela Configuração de I/O (a sua configuração, funções atribuídas e nomes definidos pelo utilizador), bem como os valores de todas as entradas e saídas. As entradas são de leitura apenas (); por isso, estes valores podem ser facilmente monitorizados aqui. As saídas, por outro lado, também podem ser definidas através desta janela.

As **E/S digitais** representam valores booleanos, em que **True ou 1 ou «verde»** significa «**sob tensão**» e **False ou 0 ou «cinzento»** significa «**sem tensão**». As entradas digitais mostram este valor num indicador circular, enquanto as saídas digitais podem ser alteradas premindo o botão do indicador quadrado para mudar o seu estado.

As E/S analógicas são apresentadas como **valores de vírgula flutuante** em incrementos de 0,1. A unidade de uma E/S analógica é definida pela configuração dessa E/S (V para tensão, mA para corrente). Enquanto as entradas analógicas são apresentadas como um número visível, as saídas analógicas podem ser alteradas através dos campos de texto ao lado do valor atual dessa saída.

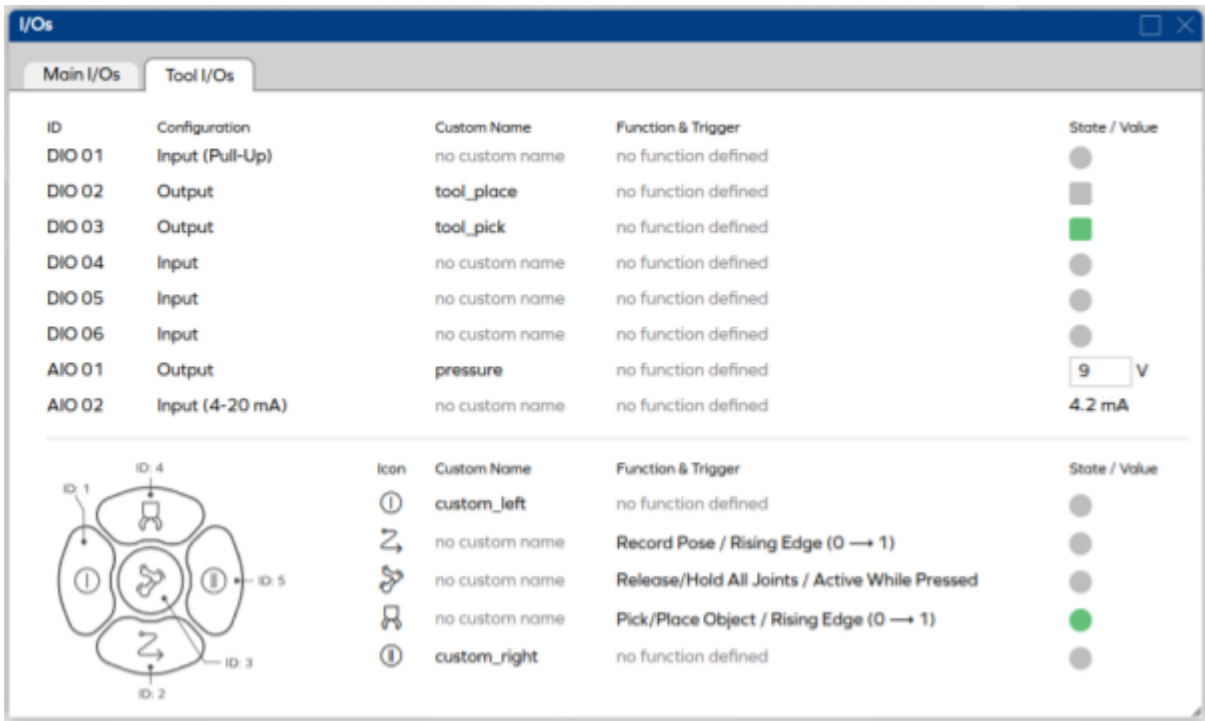


Figura 4.8: Menu E/S

4.9 Menu de visualização

Clique no botão «Visualização» na janela principal para aceder ao menu de visualização. Esta janela mostra a posição e a orientação atuais das articulações, o que é útil, por exemplo, quando o robô é operado à distância sem feedback visual direto. Também pode ver as poses previamente guardadas na visualização 3D. A visualização é interativa e permite alguns controlos:

- Gire a imagem clicando com o rato dentro da visualização (botão esquerdo do rato) e arrastando a imagem para os lados, bem como para cima e para baixo.
- Aumente e diminua o zoom girando a roda do rato para cima ou para baixo.
- Desloque a câmara lateralmente, bem como para cima e para baixo, utilizando as setas do teclado, ou clique com o botão direito do rato e arraste para a esquerda e para a direita.

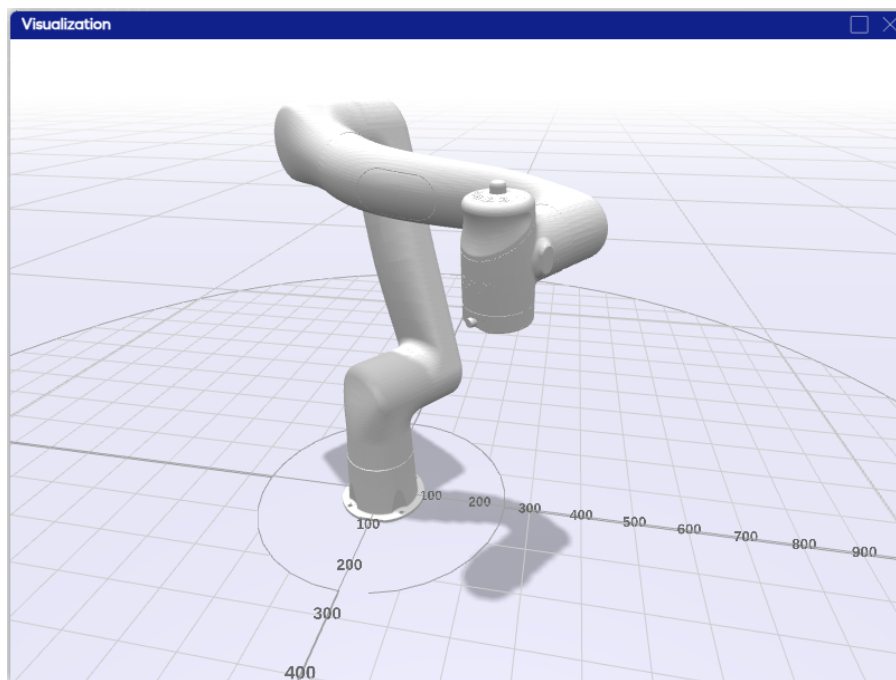


Figura 4.9: Janela de visualização

4.10 Controlo manual

Prima o botão «Controlo manual» na janela principal para aceder ao menu de controlo manual. Neste menu, o utilizador pode definir quais as articulações do robô que podem mover-se durante o controlo manual. Por predefinição, premir o botão HGC no robô (e o botão correspondente no cockpit) permite ao utilizador mover todas as articulações. Ao clicar nos indicadores de cada articulação neste menu, o utilizador pode ativar ou desativar articulações individuais para o controlo manual.

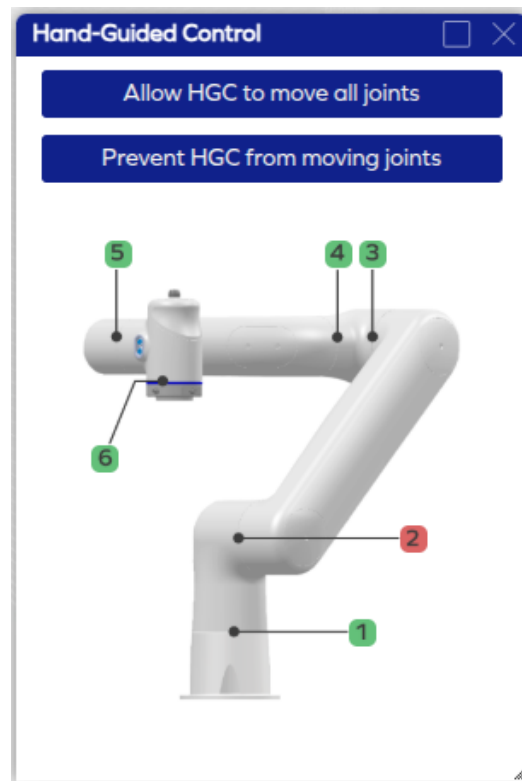


Figura 4.10: Menu para o modo manual

4.11 Monitor do sistema

Clique no botão «Monitor do sistema» na janela principal para aceder ao monitor do sistema. Este menu contém as seguintes saídas do sistema que o utilizador pode monitorizar:

4.11.1 Estado do robô

O separador «Estado do robô» apresenta o estado atual do robô. Isto inclui as coordenadas do Ponto Central da Ferramenta (TCP) em relação ao sistema de base do robô, bem como as posições, correntes e valores de temperatura de todos os motores. Além disso, os estados de todas as entradas de segurança são aqui visíveis.

4.11.2 Saída de texto

Assim que uma aplicação devolve texto através da função **print(...)**, esse texto aparece no separador «Saída de texto» desta janela. Tenha em atenção que, no caso de aplicações em execução paralela, todas as aplicações podem escrever texto na saída de texto.

4.11.3 Registo da aplicação

Assim que uma aplicação regista dados através da função `log_message(...)`, esses dados aparecem no separador «Registo da aplicação» desta janela. Aqui, os dados registados podem ser filtrados e pesquisados. Estes registos definidos pelo utilizador são descarregados juntamente com os ficheiros de registo integrados quando utiliza a combinação de teclas CTRL + SHIFT + L.

4.11.4 Histórico de mensagens

Assim que uma mensagem é exibida na interface do utilizador, esta aparece também no separador «Histórico de mensagens» desta janela. As mensagens estão divididas em quatro tipos, utilizados em diferentes contextos:



As mensagens informativas fornecem ao utilizador indicações úteis, comunicações informativas ou dicas.



As mensagens de sucesso informam o utilizador de que um comando ou uma ação foi executado com sucesso.



Os avisos fornecem ao utilizador informações importantes que devem ser lidas com atenção antes de prosseguir. Os avisos nunca são críticos e não interrompem o fluxo do programa.



Existe um vasto leque de **erros**, desde pequenos erros de operação (por exemplo, ações incorretas ou argumentos de função errados) até erros críticos, causados, por exemplo, por um robô danificado.

4.12 Menu Ajuda

O menu Ajuda contém informações úteis sobre o sistema atualmente em execução e os módulos instalados, sobre as combinações de teclas à disposição do utilizador, bem como sobre formas de contacto em caso de dúvidas ou problemas.

4.12.1 Informações sobre o robô

O separador «Informações do robô» do menu de Ajuda contém informações detalhadas sobre o ambiente AX Portal e os componentes de hardware ligados (caso estejam ligados). Na secção «Módulos» das informações do robô, encontrará uma lista dos módulos instalados, bem como informações sobre se estes foram carregados corretamente. Só se for apresentado o estado «OK» é que poderá utilizar as funções e menus disponibilizados por este módulo. Em caso de pedidos de assistência, indique sempre as informações detalhadas do robô do AX Portal e do hardware ligado. Basta clicar no botão «Copiar para a área de transferência» e colar o conteúdo (de preferência num tipo de [letra de largura fixa](#)) juntamente com o seu pedido de assistência.

4.12.2 Atalhos de teclado

O separador «Atajos de teclado» contém uma visão geral dos atalhos de teclado disponíveis. Os atalhos de teclado estão divididos em atalhos gerais, utilizados, por exemplo, para operar os botões de controlo no cockpit, e atalhos para o Editor de Aplicações, úteis para a programação de aplicações do AX Portal. Tenha em atenção que o layout do teclado no macOS difere significativamente do de outros sistemas operativos (Windows, Linux) e, por isso, a lista de combinações de teclas também é diferente.

4.12.3 Documentação do código

O separador «Documentação do código» contém a documentação de todas as funções disponíveis para a programação das aplicações. A documentação está dividida em funções principais, que estão sempre disponíveis, e funções adicionais, que se referem a módulos instalados adicionalmente. Através do botão «Descarregar» na parte superior do separador, a documentação do código pode ser descarregada como ficheiro ZIP. Para visualizar a documentação do código offline, abra o ficheiro «index.html» no seu navegador.

4.12.4 Licença de código aberto

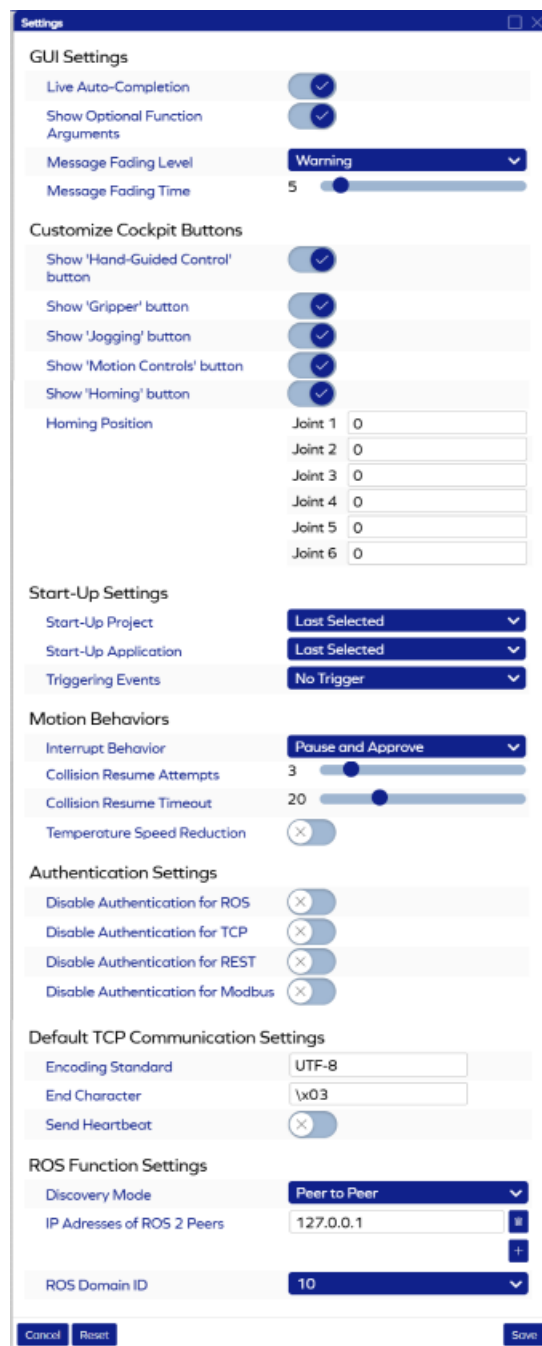
O separador «Licença de código aberto» contém todas as informações necessárias sobre as bibliotecas e módulos de código aberto utilizados pelo AX Portal.

4.12.5 Informações de contacto

O separador «Contacto» apresenta as informações de contacto do serviço de assistência. Por favor, consulte sempre os manuais antes de enviar pedidos de assistência. Ao enviar pedidos de assistência, certifique-se de que descarrega os ficheiros de registo e os envia juntamente com o seu pedido.

4.13 Menu de configurações

No menu de configurações, o utilizador pode alterar permanentemente algumas das configurações básicas do AX Portal. Clique em «Guardar» para aplicar todas as alterações, em «Cancelar» para as rejeitar ou em «Redefinir» para repor todas as configurações para os valores guardados mais recentemente. A seguir, são explicadas todas as categorias e opções:



The Settings window is organized into the following sections:

- 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
 - End Character: \x03
 - Send Heartbeat: ☒
- ROS Function Settings**
 - Discovery Mode: Peer to Peer (dropdown)
 - IP Addresses of ROS 2 Peers: 127.0.0.1 (text input with add/remove buttons)
 - ROS Domain ID: 10 (dropdown)

Buttons at the bottom: Cancel, Reset, Save.

Figura 4.13: Menu de configurações

4.13.1 Configurações da GUI

Esta categoria contém várias configurações relacionadas com a interface do utilizador.

Preenchimento automático de texto	Ative esta opção para ver sugestões de autocompletar texto em tempo real ao introduzir código de programa. Se estiver desativada, ainda poderá completar automaticamente a palavra atual através da combinação de teclas CTRL+SPACE.
Mostrar argumentos opcionais	Selecione se os argumentos opcionais das funções devem ou não ser inseridos ao inserir funções no código do programa. Para manter o código livre de argumentos de função desnecessários, recomendamos que desative esta opção. No entanto, se ainda não estiver familiarizado com as funções integradas, recomendamos que ative esta opção.
Nível de ocultação de mensagens	Selecione até que nível as mensagens exibidas serão ocultadas. Para que todas as mensagens sejam ocultadas, selecione o nível mais alto possível. Para que todas as mensagens permaneçam visíveis permanentemente, selecione o nível mais baixo possível.
Tempo de ocultação das mensagens	Indique o tempo de ocultação das mensagens a ocultar em segundos (entre 1 e 60).

4.13.2 Personalizar botões do cockpit

Com estas definições, o utilizador pode ativar e desativar facilmente os botões de controlo que aparecem no cockpit da interface gráfica do utilizador, tanto para a interface de ambiente de trabalho como para a interface do painel. O funcionamento detalhado destes botões é explicado na secção «Cockpit». É aconselhável desativar um botão se o utilizador não pretender utilizá-lo ou não puder utilizá-lo (por exemplo, não é possível utilizar o botão «Garra» se não estiver ligada nenhuma ferramenta interativa), ou se o utilizador não pretender que um operador o utilize através da interface do painel.

Mostrar botão *	Ative para exibir este botão e desative para ocultá-lo.
Posição de referência	Especifique os ângulos da posição de referência, ou seja, a posição associada ao botão «Homing» acessível no cockpit. Introduza um valor para cada ângulo de articulação.

4.13.3 Configurações de inicialização do programa

Nesta categoria, todas as definições dizem respeito ao processo de arranque do AX Portal. Ao ajustar estas definições, é possível evitar ações repetitivas durante o arranque. Para algumas definições de arranque do programa, selecione a entrada da lista «Selecione recentemente» (entrada padrão) para guardar e carregar novamente a definição utilizada pela última vez.

Projeto selecionado	Selecione o projeto que será carregado no arranque do programa. Todos os projetos definidos aparecem nesta lista.
Aplicação selecionada	Quando um projeto estiver selecionado, selecione a aplicação que será carregada no arranque do programa. Todas as aplicações principais definidas para este projeto aparecem nesta lista.
Evento desencadeador	Selecione se o robô deve ser colocado automaticamente em funcionamento ao iniciar o programa ou se deve mesmo executar uma aplicação. Por predefinição, esta opção está desativada. Para selecionar um gatilho, é necessário primeiro selecionar algumas das configurações de arranque acima indicadas, a fim de evitar que o utilizador execute acidentalmente aplicações indesejadas.

4.13.4 Comportamento de movimento

Nesta categoria, todas as configurações referem-se ao comportamento durante os movimentos do robô.

Comportamento de interrupção	Selecione o comportamento de movimento em caso de interrupção. Para mais detalhes sobre os diferentes comportamentos em caso de interrupções e sobre como estas são resolvidas, consulte as subsecções seguintes «Comportamento em caso de interrupção» e «Diálogo de interrupção».
Configurações Defina	Defina o número máximo de tentativas para retomar o movimento após uma colisão. Se ocorrerem demasiadas colisões dentro do tempo limite definido, o robô interrompe a tentativa de retomada.
Redução em temperatura	Selecione se a velocidade do robô deve ser reduzida automaticamente em caso de temperaturas elevadas do acionamento. Esta configuração está ativada por predefinição, o que faz com que os movimentos acima dos 70 °C abrandem automaticamente para evitar o sobreaquecimento. Se forem detetadas temperaturas superiores a 90 °C, é acionado um erro crítico para evitar danos causados pelo sobreaquecimento. Mais detalhes sobre esta função estão descritos na secção «Limites de desempenho» do «Manual do robô colaborativo de 6 eixos AX6».

4.13.4.1 Comportamento em caso de interrupção

As interrupções podem ocorrer por diversos motivos, tais como:

- foi acionado um evento de paragem SS1
- foi acionado um evento de paragem SS2
- ocorreu uma colisão do robô com uma pessoa, um objeto ou o ambiente
- foi acionada uma interrupção do dispositivo de proteção

Uma colisão é detetada assim que as correntes dos eixos do robô se desviam do modelo dinâmico do robô para além de um valor-limite definido. A deteção de colisões é uma função de segurança que oferece um nível adicional de proteção; não deve ser utilizada intencionalmente como parte de uma aplicação, uma vez que colisões repetidas sobrecarregam os eixos do robô e aceleram o desgaste.

Tenha em atenção que, para o correto funcionamento da deteção de colisões, o programador deve ajustar a carga útil do robô para o valor correto assim que um objeto for recolhido ou depositado.

Para além das colisões físicas, é possível ligar ao robô dispositivos de segurança externos, como barreiras de luz ou dispositivos de proteção semelhantes. Estes disparadores atuam de forma preventiva, antes de ocorrer qualquer contacto físico, mas são tratados da mesma forma que os eventos de colisão. O comportamento de interrupção define a forma como o sistema trata as interrupções desencadeadas tanto por colisões como por dispositivos de proteção.

Atualmente, este é o procedimento recomendado para o comportamento de interrupção:

Pausar e Confirmar	Esta configuração faz com que a aplicação em execução entre em pausa em caso de interrupção e aguarde a confirmação do utilizador (janela de diálogo) para continuar ou parar a aplicação.
---------------------------	--

4.13.4.2 Caixa de diálogo de interrupção

A caixa de diálogo de interrupção é exibida assim que ocorre uma condição de interrupção durante o funcionamento do robô. Os fatores que normalmente desencadeiam interrupções são colisões detetadas, o acionamento de uma paragem de emergência, o disparo de um dispositivo de proteção ou outros eventos relevantes para a segurança. A caixa de diálogo orienta o utilizador de forma segura e controlada na resolução da situação.

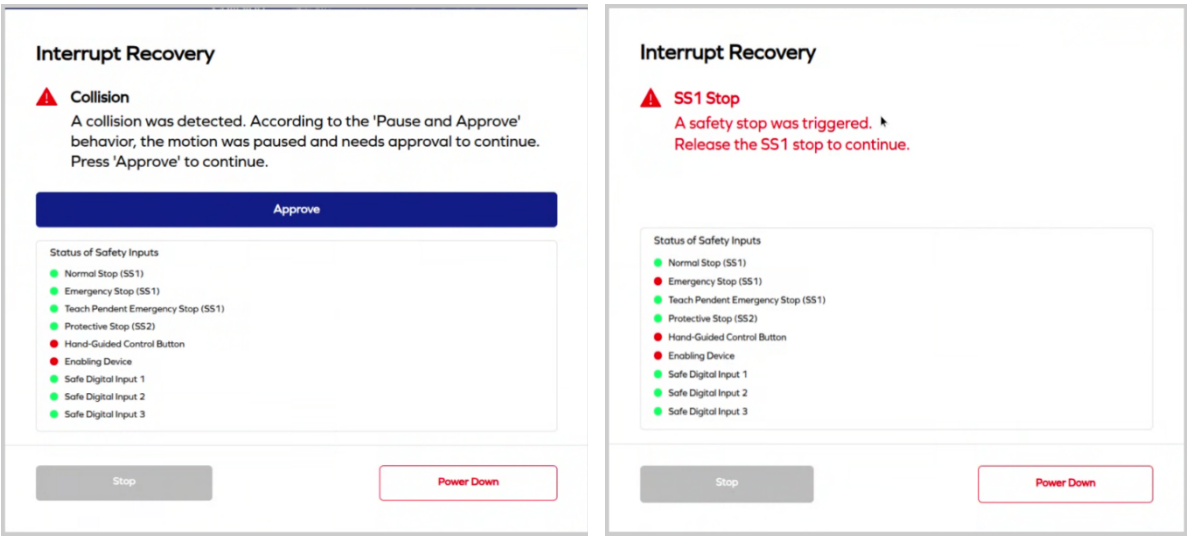


Figura 4.13.4.2: Recuperação após interrupção

A caixa de diálogo está dividida em duas áreas principais:

Área superior – Informações sobre a interrupção e a resolução: Esta área descreve a condição de interrupção atualmente detetada e fornece instruções claras para a sua resolução. Exemplos disso são:

- Garantir que o robô possa continuar a mover-se após uma colisão.
- Desativar uma paragem de emergência ou outros dispositivos de segurança.
- Reiniciar uma condição de segurança, se necessário.

Dependendo do tipo de interrupção, é apresentado um botão de ação como «Confirmar», «Continuar» ou «Reiniciar». O utilizador deve seguir as instruções e premir o botão indicado para prosseguir com o funcionamento.

Área inferior – Estado das entradas de segurança: A metade inferior da caixa de diálogo apresenta o estado em tempo real de todas as entradas de segurança relevantes. Isto permite ao utilizador identificar rapidamente qual o dispositivo de segurança atualmente ativo, por exemplo:

- Paragem de emergência (incluindo a paragem de emergência no comando manual)
- Paragem de proteção
- Dispositivo de autorização
- Botão de comando manual
- Entradas digitais seguras

Os indicadores verdes sinalizam um estado normal, enquanto os indicadores vermelhos indicam um estado ativo ou de bloqueio (ou que o botão não está pressionado).

Opção de paragem

O botão «Parar» está sempre disponível quando o utilizador tem de escolher entre continuar a aplicação/movimento anterior ou parar. Ao premir o botão, o robô é parado e regressa ao estado «Pronto», no qual está preparado para novas ações.

Opção «Desativar»

O botão «Desligar» está sempre disponível na caixa de diálogo de interrupção. Permite desligar o robô a qualquer momento. Tenha em atenção que o desligamento não resolve automaticamente a condição de interrupção; a causa subjacente (por exemplo, um botão de paragem de emergência premido) poderá ter de ser eliminada antes de se poder retomar o funcionamento normal após o reinício.

Fluxo de trabalho típico

1. Ocorre uma interrupção e a caixa de diálogo é apresentada automaticamente.
2. Leia a descrição do estado detetado na área superior.
3. Verifique o estado das entradas de segurança na parte inferior para identificar a causa ativa.
4. Resolva a situação manualmente (por exemplo, desbloqueie a paragem de emergência, remova o obstáculo).
5. Prima o botão de ação apresentado («Confirmar», «Continuar» ou «Reiniciar») para prosseguir.
6. Repita a ação, se necessário, até que o robô retome o funcionamento normal.

Violação da SLP

Um caso especial de recuperação de interrupção é a violação de SLP (Safe Limited Position), que ocorre quando o robô tenta mover-se para fora dos limites de posição definidos nas configurações de segurança. Neste caso, a caixa de diálogo de interrupção orienta o utilizador a mudar para o modo manual, permitindo que o utilizador saia do limite de posição (jogging). Com exceção do

deslocamento, não são permitidos movimentos enquanto o robô se encontrar numa violação de SLP. Se o robô for desligado ou arrancar com uma violação de SLP ativa, proceda da seguinte forma:

1. Mude para o modo manual
2. Coloque o robô em funcionamento
3. Navegue até ao controlo de precisão
4. Mova o robô para fora dos limites de posição através do controlo de precisão

4.14 Menu de configurações de segurança

O menu de configurações de segurança é o acesso ao sistema de segurança do robô da série AX. Esta vista foi concebida para **responsáveis pela segurança**. Outros utilizadores podem apenas ler e exportar os parâmetros de segurança, mas não os podem alterar.

4.14.1 Interface do utilizador

Ao aceder às definições de segurança, este menu apresenta todos os parâmetros de segurança atualmente definidos no sistema de segurança do robô. Qualquer utilizador pode visualizar estas definições, mas apenas um responsável pela segurança pode alterá-las. Este menu está dividido em submenus, pelos quais pode navegar utilizando os números de página ou os botões «Avançar» e «Voltar» na parte inferior da janela.

Exportar	Ao clicar no botão «Exportar», as definições de segurança são exportadas como um ficheiro *.yaml. Este ficheiro pode ser guardado para fazer uma cópia de segurança das definições de segurança e também importado para outros robôs, ao fim de duplicar a configuração de segurança.
Editar	O responsável pela segurança pode clicar no botão «Editar» (e introduzir a sua palavra-passe de utilizador) para aceder ao modo de edição.

Assim que o menu estiver no modo de edição, as definições de todos os submenus passam a ser editáveis. O responsável pela segurança pode agora ajustar os limites de segurança de acordo com a disposição de montagem e a aplicação do robô. Para facilitar a visualização das definições de segurança que foram alteradas, os campos alterados são assinalados a azul.

Importar	Ao clicar no botão «Importar», é possível carregar um ficheiro previamente exportado com as definições de segurança. Isto substituirá os valores alterados manualmente em todos os submenus.
Aplicar	Ao clicar no botão «Aplicar» (e introduzir a palavra-passe do utilizador), todas as configurações de segurança são enviadas para o sistema de segurança, que verifica as opções e os valores-limite. Caso a verificação não seja bem-sucedida, as entradas inválidas no menu são apresentadas a vermelho, para que possam ser facilmente identificadas. Se a verificação for bem-sucedida, o menu passa para o modo de verificação. Tenha em atenção que, nesta fase, os valores são guardados, mas ainda não são verificados. A verificação das definições é necessária para ligar o robô.
Cancelar	Ao pressionar o botão «Cancelar», as configurações de segurança não são aplicadas e todas as alterações são descartadas. O menu volta então ao modo de observação (padrão).

Assim que o responsável pela segurança concluir a edição das configurações de segurança e todos os valores tiverem sido aplicados com sucesso, este menu entra no modo de verificação. As configurações aplicadas anteriormente, mas ainda não verificadas, são assinaladas em amarelo.

Verificar	Ao clicar no botão «Verificar página», o responsável pela segurança marca oficialmente as configurações do submenu atualmente visível como aceites, e o próximo submenu é selecionado. Clicar no botão «Concluir verificação» do último submenu é o último passo para guardar as configurações.
------------------	---

Após a verificação dos novos valores, é obrigatório validar o comportamento do sistema de segurança.

Isto pode ser feito através do arranque no modo de validação de segurança, o que permite ao software ignorar todos os limites de segurança durante o processo de validação, a fim de verificar se o sistema de segurança aciona conforme o esperado.

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	
	minmax	minmax	minmax	minmax	
x	-5003000	-30003000	-30003000	-30003000	mm
y	-800800	-30003000	-30003000	-30003000	mm
z	-30003000	100-100	-30003000	-30003000	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 robotLimit 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 Offset0 mmTool Safety Radius0 mm

Cancel

<123456>

ImportApply

Figura 4.14.1: Menu de configurações de segurança durante a edição das configurações

4.14.2 Configurações de segurança

A maioria das configurações de segurança (em todos os submenus) está dividida em 4 colunas. Estas colunas definem os casos de segurança do robô. Um caso de segurança é acionado quando a respetiva entrada digital segura (SDI 1-3) é acionada. Os parâmetros padrão estão sempre ativos, independentemente das entradas digitais seguras.

As saídas digitais seguras (SDO) não são configuráveis e as suas especificações elétricas estão descritas no «Manual do Controlador do Robô RC-A101».

Em algumas configurações, é definida uma reserva de segurança. Este valor indica o intervalo permitido em que o software pode acionar mais cedo do que o sistema de segurança, a fim de evitar uma violação de segurança.

4.14.2.1 Posição limitada de segurança (SLP)

O primeiro submenu das configurações de segurança contém a posição limitada de segurança. Aqui, é possível limitar a posição do robô nas direções x, y e z, em relação ao sistema de referência do robô. Se os sistemas de referência não forem claros, clicar no ícone de informação exibe uma imagem com as coordenadas. Pode limitar a posição de todas as partes do robô (incluindo a base e os membros) ou apenas a posição do Ponto Central da Ferramenta (TCP) do robô.

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**

Figura 4.14.2.1: Exemplos de configurações para SLP

4.14.2.2 Ângulos de articulação limitados de forma segura

O segundo submenu das configurações de segurança contém os ângulos das articulações com limitação de segurança. Por predefinição, os ângulos das articulações com limitação de segurança estão desativados. Ao ativar um caso de segurança, é possível monitorizar a posição angular das articulações.

No entanto, dentro de um caso de segurança, a área de monitorização dos ângulos das articulações está limitada ao intervalo $[-180^\circ, 180^\circ]$. Isto permite restringir a área de trabalho do robô em relação aos limites físicos reais do robô. Para evitar esta restrição, a monitorização dos eixos 1, 4 e 6 pode ser desativada individualmente.

Os ângulos das articulações com limites de segurança só estão ativos se tanto o limite inferior como o superior estiverem definidos. Se ambos os limites forem deixados em branco, a monitorização para a articulação correspondente é desativada. Tenha em atenção que os limites inferior e superior devem ser ambos deixados em branco ou ambos definidos.

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

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

Figura 4.14.2.2: Exemplos de configurações para ângulos de articulação limitados de forma segura

4.14.2.3 Velocidade limitada de segurança (SLS)

O terceiro submenu das configurações de segurança contém a velocidade limitada de segurança. Aqui, a velocidade de cada eixo do robô pode ser limitada individualmente, assim como a velocidade de movimento de todas as partes do robô, incluindo a ferramenta, a base e os membros.

The screenshot shows the 'Safety Settings' window with the 'Safely Limited Speed (SLS)' tab selected. The window title is 'Safety Settings'. Below the title bar, there is a sub-header 'Safely Limited Speed (SLS)' and a description: 'Defining the speed limits of each joint and all movable parts of the robot.' A note states: '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

At the bottom of the window, there is an 'Export' button on the left, a navigation bar with buttons '<', '1', '2', '3' (selected), '4', '5', '6', and '>', and 'Edit' and 'Start Verification' buttons on the right.

Figura 4.14.2.3: Exemplos de configurações para SLS

4.14.2.4 Configurações de segurança para potência e força

O quarto submenu das definições de segurança contém definições relativas à potência e à força:

- Para cada caso de segurança, o utilizador pode definir um valor limite de força diferente, que determina a sensibilidade com que o robô reage a forças externas, como colisões com pessoas ou objetos. O valor limite de força é aplicado no Ponto Central da Ferramenta (TCP).
- A definição da carga útil média (e dos valores corretos de carga útil durante o funcionamento), definida como o peso da ferramenta mais os objetos transportados, é importante para ajustar os limites de força de forma mais sensível à sua aplicação específica.
- O vetor de gravidade do robô deve corresponder à forma de montagem do robô. Se o robô estiver montado no chão, o vetor de gravidade é $[0, 0, -9,81]$. Se, pelo contrário, o robô estiver montado numa parede, numa superfície inclinada ou no teto, o vetor de gravidade deve ser calculado previamente e introduzido corretamente aqui.

The screenshot shows the 'Safety Settings' window with the following sections:

- 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.
- Force Limit**
A table with four columns: Default, Safe Case 1, Safe Case 2, and Safe Case 3. Each column has a text input field containing the value '250'. To the right of the table is a unit label '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**
A text input field containing '0' followed by the unit 'kg'.
- Gravity Vector**
Defining the gravity vector in the robot's base frame based on its mounting orientation.
- Gravity Vector**
Three text input fields for x, y, and z components. The x and y fields contain '0', and the z field contains '-9.81'. To the right is a unit label 'm/s²'.

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

Figura 4.14.2.4: Configurações seguras para limites de potência e força

4.14.2.5 E/S de segurança e restrições do operador

O quinto submenu do menu de configurações de segurança permite definir o comportamento das E/S relevantes para a segurança e as autorizações do operador. Os utilizadores podem definir como as paragens de segurança são tratadas, se as entradas são monitorizadas no modo manual, as condições de reinicialização, as autorizações para a mudança de modo de funcionamento, bem como os requisitos de palavra-passe para alterações no funcionamento.

The screenshot shows a window titled "Safety Settings" with a blue header bar. It contains 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, there is a navigation bar with buttons: "Export", "<", "1", "2", "3", "4", "5" (highlighted), "6", ">", "Edit", and "Start Verification".

Figura 4.14.2.5: Definições de configuração para E/S de segurança

4.14.2.6 Dados de verificação

O último submenu das definições de segurança contém informações sobre a configuração de segurança:

- As somas de verificação lidas da configuração de segurança (para resolução de erros)
- O contador de transferências (indica quantas vezes os parâmetros de segurança foram gravados)
- Quando os parâmetros de segurança foram atualizados pela última vez
- Se os parâmetros de segurança foram verificados
- O número de série e a versão do firmware do dispositivo de segurança

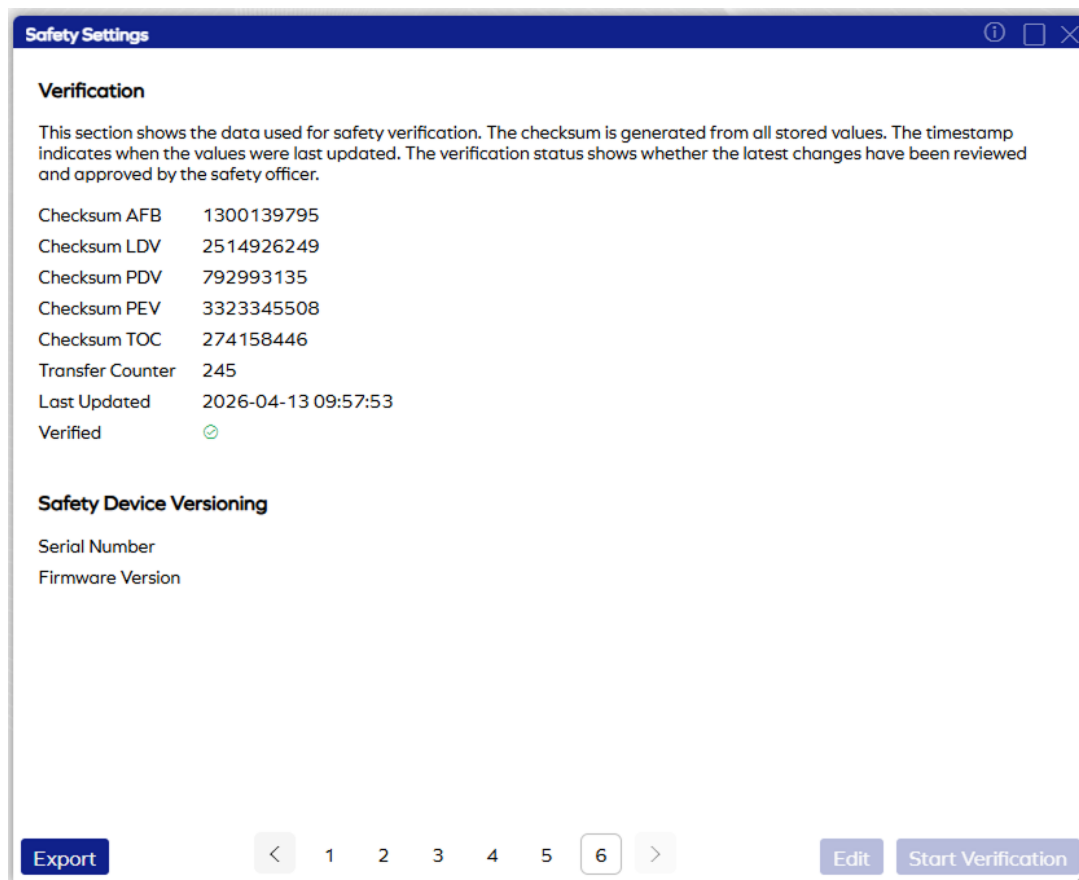


Figura 4.14.2.6: Configurações de verificação

4.14.2.7 Funções das saídas de segurança

Os estados lógicos de determinadas funções de segurança estão ligados às saídas de segurança do controlador do robô. Encontrará informações detalhadas no «Controlador do robô RC-A101: Manual».

4.14.2.8 Ligação de dispositivos opcionais

Para a ligação de dispositivos opcionais, tais como suportes para tablets ou interruptores de autorização, consulte o «Manual do controlador do robô RC-A101».

5. Interface do painel (GUI)

5.1 Introdução

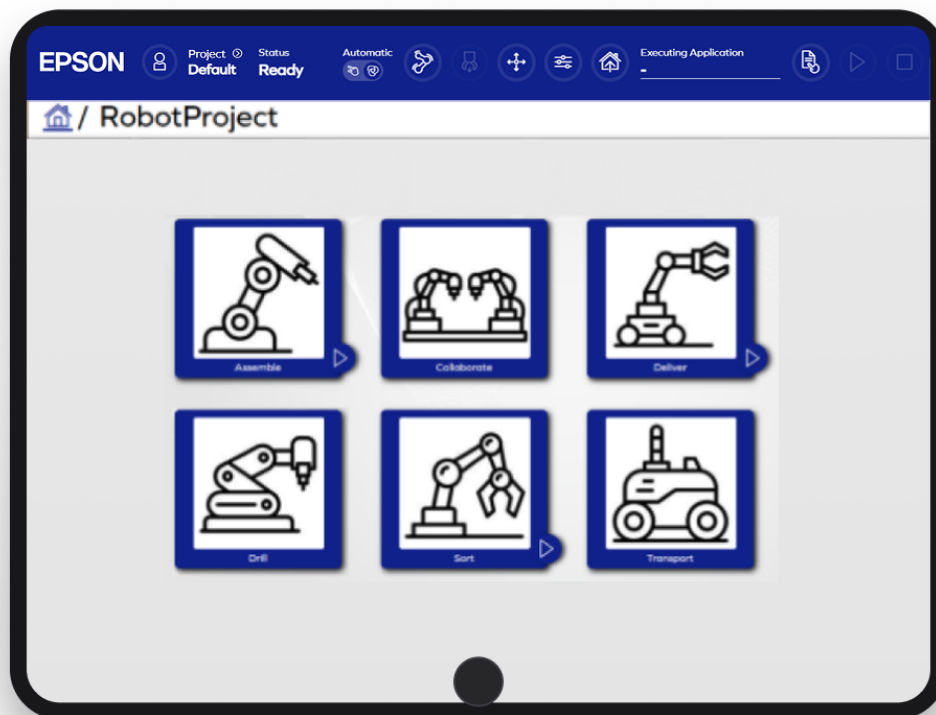


Figura 5.1.a: Exemplo de captura de ecrã da interface do painel

A interface do painel é uma alternativa à interface do ambiente de trabalho para um acesso rápido às suas principais aplicações. Esta interface foi concebida para ser totalmente personalizável de acordo com os requisitos do cliente e, em particular, para apresentar exclusivamente as aplicações (e informações sobre o seu estado). A interface do painel é compatível com ecrãs touch screen e o seu design assemelha-se à conhecida estrutura tipo «app» dos dispositivos móveis modernos, tornando-a intuitiva mesmo para utilizadores sem conhecimentos técnicos. Na interface do painel, o utilizador também não precisa de se preocupar com a programação ou com o código subjacente à aplicação. Basta clicar na aplicação correspondente e a aplicação é iniciada. Esta interface pode ser acedida através da interface de administração, iniciando a sessão e utilizando o botão «Go to Panel Interface».

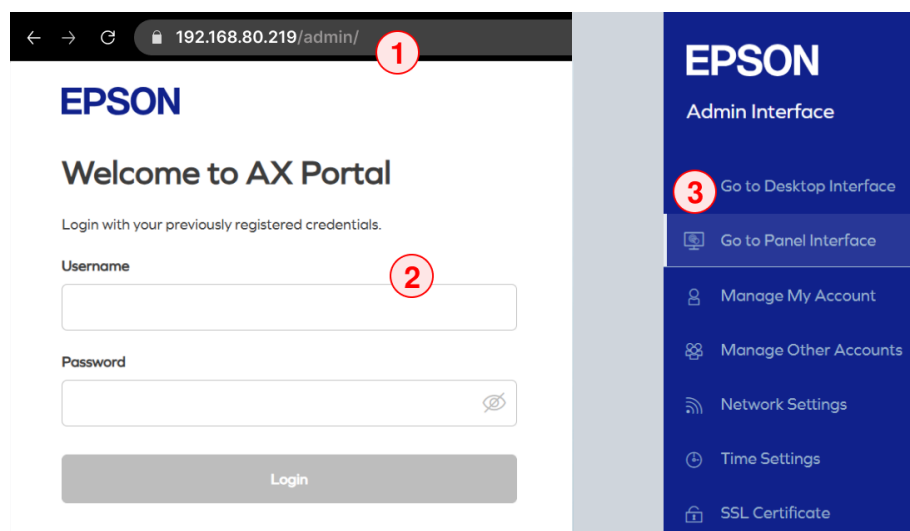


Figura 5.1.b: Acesso à interface do painel

Em alguns navegadores (especialmente no Windows), a escala/ampliação da visualização está predefinida para um valor elevado (por exemplo, 150 %), o que pode fazer com que partes da interface (como o Cockpit ou o logótipo) pareçam cortadas. Para uma melhor visualização, recomendamos que defina a escala/ampliação da exibição para 100 % (ou ainda menos).

5.2 Configurar aplicações para a interface do painel

Por predefinição, a interface do painel não apresenta quaisquer aplicações. O utilizador deve primeiro definir qual a aplicação que deve ser apresentada para execução rápida pelo operador. Para configurar uma aplicação do painel, proceda da seguinte forma:

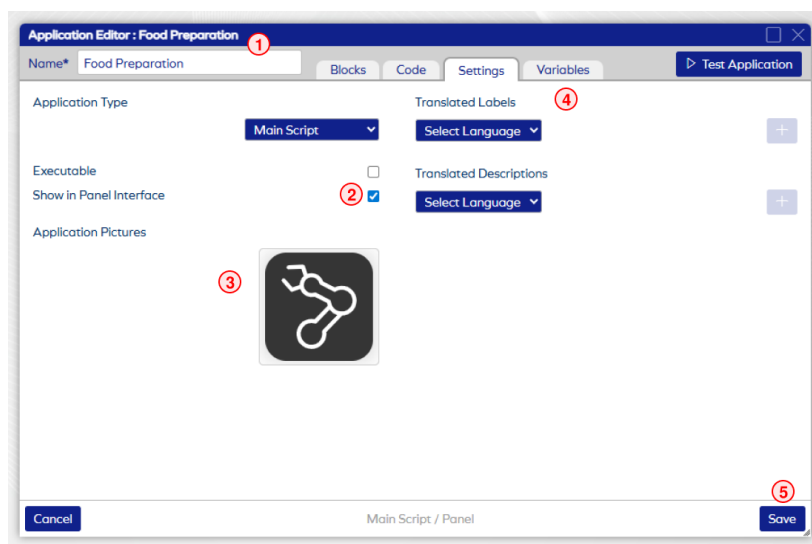


Figura 5.2.a: Definições relevantes para a interface do painel no Editor de Aplicações (interface do ambiente de trabalho)

Definições relevantes para a interface do painel: 2., 3., 4.

1. Navegue até à interface do ambiente de trabalho, edite a aplicação e abra as definições da aplicação,
2. Ative a opção «Mostrar na interface do painel»,
3. Opcionalmente, carregue uma imagem adequada da aplicação (a proporção ideal é 5:4), que será posteriormente exibida juntamente com o nome da aplicação no menu principal da interface do painel,
4. Opcionalmente, aceda ao separador Variáveis da aplicação para adicionar variáveis de estado, com as quais poderá monitorizar e/ou alterar variáveis através da interface do painel,
5. E, por fim, guarde a aplicação.

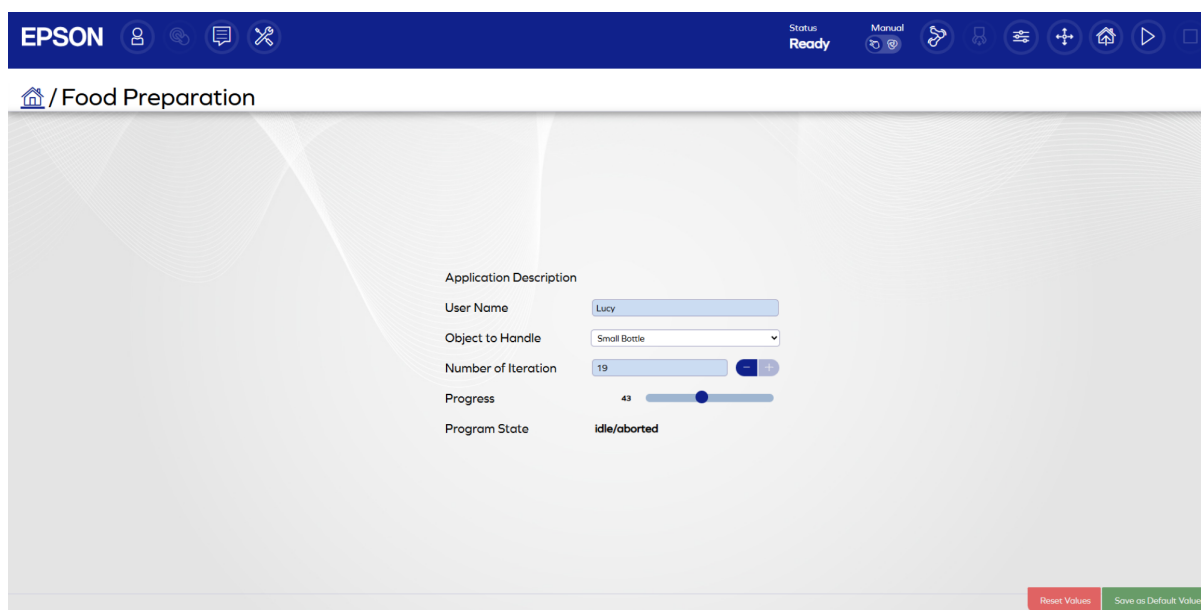


Figura 5.2.b: Aplicação do painel com variáveis de estado

5.3 Cockpit e configurações

O Cockpit da interface do painel apresenta, em grande parte, os mesmos botões que o Cockpit da interface do ambiente de trabalho. Encontrará uma explicação detalhada de todos os elementos na secção Cockpit. Para além destes elementos, verá, no lado esquerdo, alguns botões diferentes:



O botão «Painel» abre a página principal do painel, onde são apresentadas as aplicações a seleccionar e a executar.



O botão «Mensagens» abre o histórico de mensagens, onde todas as mensagens anteriores, incluindo avisos e erros, são apresentadas por ordem cronológica.



O botão «Definições» leva-o à janela de definições, que contém apenas as definições mais importantes para acesso rápido, nomeadamente

- a seleção da ferramenta conectada,
- Ativação e desativação do robô,
- Download dos ficheiros de registo.

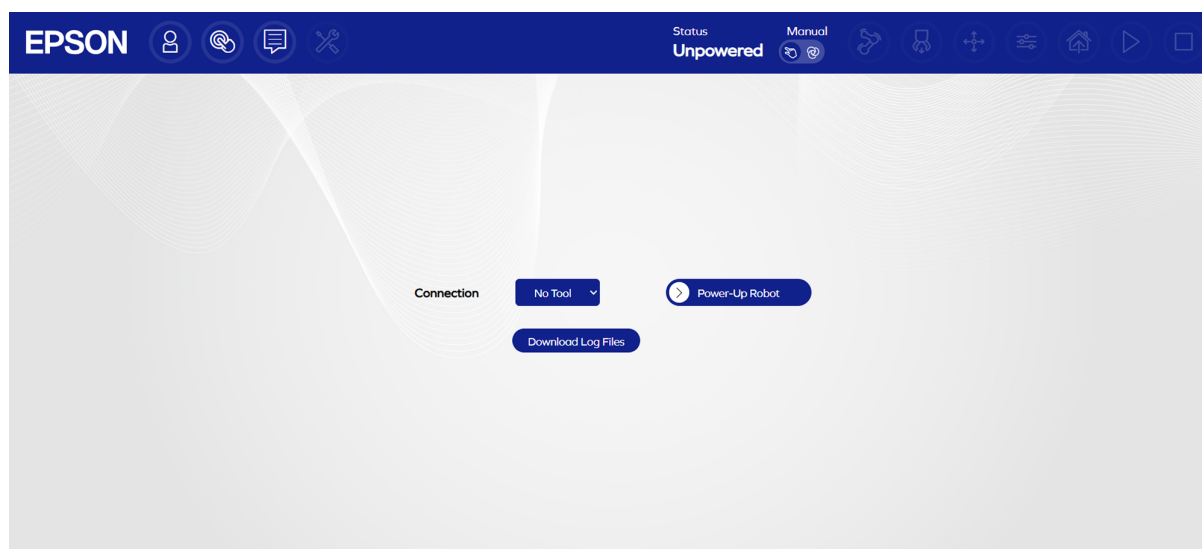


Figura 5.3: Configurações da interface do painel

5.4 Controlar aplicações

Na janela principal da interface do painel, todas as aplicações são listadas numa estrutura semelhante a uma aplicação, sendo exibidos o seu nome e a sua imagem. O painel distingue entre dois tipos de aplicações:

- As aplicações sem variáveis de estado apresentam um botão «Play». Clique na aplicação para a executar diretamente.
- As aplicações com variáveis de estado são apresentadas sem o botão «Play». Clique nas aplicações para aceder à vista detalhada com todas as variáveis de estado. Nesta janela, pode alterar as variáveis de estado editáveis e clicar no botão «Executar» para executar essas aplicações. Clique no botão «Voltar» para regressar à janela principal do painel.

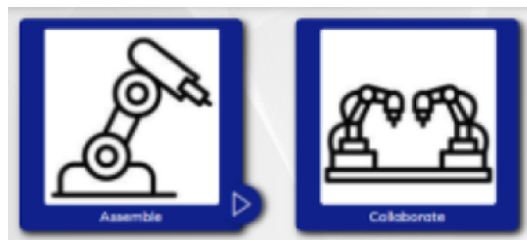


Figura 5.4: Botões para seleção de aplicações

6. Definições Cinemática

A cinemática do robô descreve como o robô se move e como os seus movimentos podem ser representados. Para trabalhar com a cinemática, o utilizador deve compreender as várias definições dos parâmetros cinemáticos e dos objetos disponíveis:

6.1 Coordenadas

As coordenadas definem uma posição e orientação absolutas do robô. A posição é um vetor com três valores $[x, y, z]$ em mm, e a orientação é igualmente um vetor com três valores $[\text{roll}, \text{pitch}, \text{yaw}]$ (ângulos náuticos) em graus.

A inicialização pode ser efetuada da seguinte forma:

```
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})
```

O acesso aos elementos pode ser feito da seguinte forma:

```
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 Ângulos náuticos

Os ângulos náuticos são uma das muitas formas de definir a componente de orientação (roll, pitch, yaw) das coordenadas. Os ângulos náuticos são normalmente utilizados para a orientação de aeronaves. Nos ângulos náuticos, uma rotação tridimensional é definida como

1. uma rotação em torno do eixo z negativo na direção da guinada (yaw, ψ),
2. uma rotação em torno do eixo y negativo na direção do pitch (inclinação, θ) e
3. uma rotação em torno do eixo x positivo na direção do rolamento (roll, ϕ).

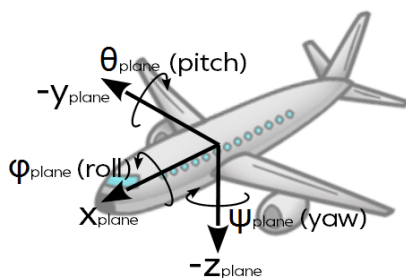


Figura 6.1.1.a: Sistema de referência náutico

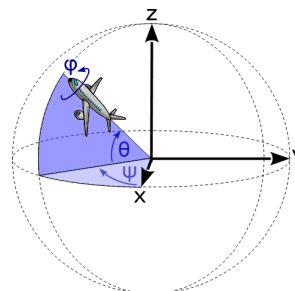


Figura 6.1.1.b: Ordem das rotações

Nota: Assim que uma função exigir que trabalhe com orientações tridimensionais do efector final, é útil imaginar um plano fixado no efector final, de modo a que os sistemas de referência do efector final e da aeronave coincidam. Desta forma, os ângulos (roll, pitch e yaw) que conduzem precisamente a esta orientação podem ser deduzidos com muito mais facilidade.

6.1.2 Sistemas de referência

Para a manipulação de robôs da série AX, os seguintes sistemas de referência são relevantes:

- O **sistema de referência** de base de um robô é definido de forma a que o seu eixo z aponte para fora da placa de base e o eixo x seja paralelo à placa de base e aponte para fora das ligações na base. A conhecida regra da mão direita define então a direção do eixo y.
- O **sistema de trabalho** de um robô é um sistema de referência definido pelo utilizador em relação ao sistema de referência de base e corresponde a este, salvo indicação em contrário. As coordenadas são especificadas neste sistema de referência. O utilizador pode utilizar este sistema de referência para definir um espaço de trabalho próprio para o robô.
- O **sistema de flange** de um robô difere consoante o tipo de ferramenta:
 - Se não estiver montada nenhuma ferramenta, a origem deste sistema de referência situa-se no centro da flange (placa de ligação) na parte superior do robô. A orientação corresponde à do sistema de base, quando todos os ângulos das articulações são nulos.
 - Se estiver montada uma ferramenta definida pelo utilizador (e definida através do Editor de Ferramentas), o sistema de referência é deslocado para a posição e orientação especificadas nessa ferramenta.
- O **sistema de ferramentas** de um robô é um sistema de referência definido pelo utilizador em relação ao sistema de referência da flange e corresponde a este, salvo indicação em contrário. As coordenadas são obtidas na origem deste sistema de referência. O utilizador pode utilizar este sistema de referência para definir o ponto central da ferramenta.

Encontra-se um exemplo destes sistemas de referência na figura seguinte:

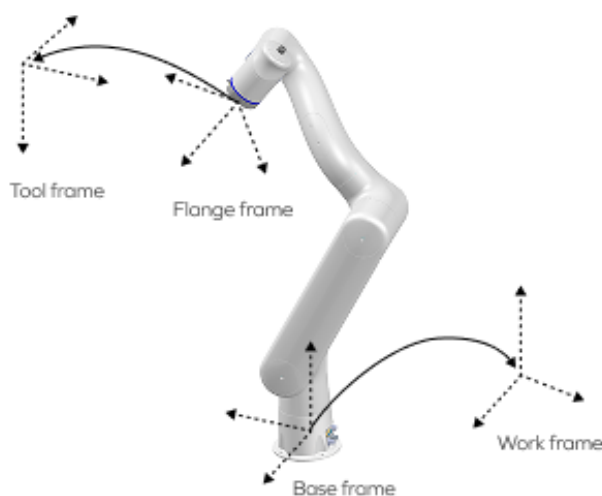


Figura 6.1.2: Sistemas de referência do robô

6.2 Configuração

Para uma determinada posição e orientação do robô, a cinemática inversa pode fornecer várias configurações de articulações diferentes. Estas configurações diferem na forma como as articulações são dobradas ou rodadas. Através da definição de configuração, é selecionada uma configuração específica entre estas opções possíveis. Num robô típico de seis eixos, diferentes combinações das orientações do ombro, cotovelo e punho podem resultar em até oito configurações diferentes, mesmo que a posição e a orientação do efector final sejam idênticas. Para efeitos de distinção, estas configurações são designadas por «c1» a «c8».

Tenha em atenção que as configurações não são adequadas para comparar duas poses ou para verificar se um percurso é contínuo. Uma alteração na configuração não torna necessariamente um percurso não contínuo. Da mesma forma, um percurso não contínuo não tem necessariamente de incluir mudanças de configuração.

6.3 Posição

Uma pose define as coordenadas e a configuração do robô. Pode ser definida através do ID ou do nome de uma pose predefinida na base de dados. Em alternativa, pode ser criada através da indicação de coordenadas e configuração com a função «Criar nova pose» ou através da indicação dos ângulos das articulações com a função de cinemática direta.

A inicialização pode ser efetuada da seguinte forma:

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

6.4 Ponto

Um ponto é um elemento discreto de um percurso. Além da pose do robô, indica também a velocidade do robô, bem como o intervalo de tempo decorrido desde o ponto anterior. Isto é designado por PVT (Posição, Velocidade, Tempo) e é normalmente guardado como um vetor de ângulos de articulação, um vetor de velocidades de articulação e um intervalo de tempo (Float). Note que uma versão alargada de um ponto pode incluir adicionalmente a aceleração do robô.

6.5 Trajetória

Uma trajetória define a forma como o robô se move entre duas ou mais posições. Uma trajetória contínua é definida exclusivamente no espaço da ferramenta por vários segmentos, que podem ser lineares ou circulares. A trajetória é posteriormente discretizada e transferida para o espaço das articulações. Uma trajetória é designada como contínua quando as soluções discretas da trajetória não exigem o reposicionamento do robô (mudança de configuração). Uma trajetória é designada como contínua quando a aceleração ao longo da trajetória é constante.

Uma trajetória começa numa pose especificada (em oposição a coordenadas). Os segmentos da trajetória definem normalmente coordenadas, uma vez que a configuração da trajetória é determinada pela pose inicial. Ao utilizar a solução mais próxima da cinemática inversa em relação ao ponto

discreto anterior, a trajetória torna-se, sempre que possível, automaticamente contínua. Note que utilizamos os termos «trajeto» e «trajetória» como sinónimos, embora, tecnicamente, um trajeto não contenha informações de velocidade.

A inicialização pode ser efetuada da seguinte forma:

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

6.6 Segmento

Um segmento representa uma parte individual de um caminho que define como o TCP se move em direção ao destino.

Os tipos de segmento disponíveis são:

- **Linear:** O TCP move-se em linha reta até às coordenadas de destino definidas pelo segmento.
- **Circular:** O TCP segue uma trajetória circular a partir da sua posição atual, passando por uma posição intermédia definida, até às coordenadas de destino do segmento. Note que a orientação do TCP durante este movimento é determinada automaticamente pelo algoritmo.
- **Coordenadas:** O TCP move-se pelo caminho mais eficiente até às coordenadas de destino. Isto corresponde a um movimento linear no espaço articular.
- **Posição:** O TCP move-se pelo caminho mais eficiente até à posição de destino. Isto corresponde a um movimento linear no espaço articular.
- **Orientação:** O TCP executa um movimento puramente rotacional para atingir a orientação de destino especificada, mantendo a sua posição atual.

Para cada segmento, podem ser definidos parâmetros de velocidade e aceleração:

- Para segmentos de orientação, estes correspondem à velocidade angular em [deg/s] e à aceleração angular em [deg/s²]
- Para todos os outros tipos de segmentos, correspondem à velocidade linear em [mm/s] e à aceleração linear em [mm/s²]

Além disso, pode ser definido um raio de transição (Blend Radius), que define o desvio máximo permitido em mm em relação à pose final do segmento anterior — que corresponde à pose inicial do segmento atual — durante a transição. Um raio de transição de 0 mm obriga o TCP a percorrer exatamente a pose definida, enquanto um valor mais elevado permite transições mais suaves e naturais, permitindo que o percurso entre os segmentos seja curvo.

6.7 Grelhas

As grelhas são matrizes de pontos personalizáveis, até tridimensionais, que definem, por exemplo, as posições de tabuleiros predefinidos e podem ser utilizadas para agrupar e guardar conjuntos de valores de coordenadas. Normalmente, uma grelha é interpretada como uma matriz bidimensional; no entanto, pode também ser uma simples lista de pontos sem a segunda dimensão (por exemplo, para recolher objetos de uma correia transportadora) ou mesmo uma estrutura tridimensional (por exemplo, para empilhar objetos em caixas).

7. Módulos de ligação

Para além das interfaces de utilizador intuitivas e fáceis de utilizar para o controlo e monitorização do robô, o AX Portal inclui também um conjunto de módulos de ligação que permitem controlar e monitorizar o robô através de vários métodos de comunicação, tais como TCP, REST, ROS e Modbus. Cada um destes métodos de comunicação é adequado para diferentes casos de utilização:

- **O TCP** é ideal para comunicação em tempo real, em que é necessária uma troca de dados rápida e contínua. É frequentemente utilizado para aplicações de controlo direto, implementações de clientes personalizadas ou para integração em sistemas que requerem ligações de socket persistentes.
- **O REST** é mais adequado para integração em serviços ou aplicações baseados na web. É normalmente utilizado quando a facilidade de utilização, a escalabilidade e a compatibilidade com sistemas padronizados baseados em HTTP são mais importantes do que o desempenho em tempo real.
- **O ROS** (Robot Operating System) foi concebido para a investigação e desenvolvimento na área da robótica. É particularmente útil para a integração do robô em sistemas robóticos complexos e permite funções como a fusão de sensores, o comportamento autónomo e a interação com outros dispositivos compatíveis com o ROS.
- **O Modbus** é frequentemente utilizado em ambientes de automação industrial. É adequado para a comunicação com PLC (controladores lógicos programáveis) e outros dispositivos industriais, nos quais a fiabilidade e a padronização são fundamentais.

A utilização destes módulos é explicada neste capítulo.

7.1 Ligações, autenticação e controlo ativo

Antes de utilizar um módulo de ligação, é importante compreender como estes módulos se ligam aos clientes, como funciona a autenticação e como funciona o princípio do controlo ativo no AX Portal.

7.1.1 Canais de ligação

A ligação entre um dispositivo e o robô é sempre estabelecida através de um canal de ligação específico. Estes canais de ligação são implementados de forma ligeiramente diferente para os vários módulos de ligação.

O **módulo TCP** suporta **vários canais**. Isto significa que cada dispositivo que pretenda ligar-se ao robô cria o seu próprio canal de ligação e inicia a sessão separadamente. O encerramento de um destes canais interrompe a ligação e, além disso, liberta o controlo ativo (ver abaixo).

 AVISO	Os módulos REST, ROS e Modbus dispõem, cada um, de apenas um único canal. Isto significa que, assim que a ligação a um destes módulos for estabelecida e autenticada, qualquer dispositivo pode enviar comandos ao robô. Estes canais são criados no arranque do AX Portal e não podem ser encerrados através da ligação de dispositivos.
---	--

7.1.2 Autenticação

O AX Portal utiliza uma autenticação segura em duas etapas. Siga primeiro o procedimento de início de sessão remoto descrito na secção Autenticação segura para obter um token de acesso válido. Este

passo deve ser sempre realizado através de uma **ligação TCP separada**. Depois de iniciar sessão com sucesso e de ter gerado um token de acesso, pode registar este token através do canal de ligação seleccionado com o comando **token_login** para autenticar o seu utilizador.

 ATENÇÃO	A etapa de autenticação pode ser ignorada se desativar a autenticação via TCP no menu de configurações. Tenha em atenção que o módulo Modbus não oferece a possibilidade de enviar um comando <code>token_login</code>; por isso, a autenticação deste módulo deve ser desativada no menu de configurações para que possa ser utilizado.
---	--

7.1.3 Controlo ativo

O AX Portal dispõe de uma proteção da estação de controlo ativa, para que outros utilizadores ou ligações não possam controlar o robô simultaneamente. Sem esta proteção, poderiam surgir situações potencialmente perigosas.

O comando **request_active_control** é utilizado por um canal de ligação para se tornar a estação de controlo ativa. Assim que este comando for executado com sucesso, essa estação de controlo detém o controlo permanente até que o canal seja encerrado ou o robô fique sem alimentação. Se outra estação de controlo detiver o controlo, este comando fica bloqueado até que essa estação de controlo aceite o pedido de controlo.

O comando **release_active_control** é utilizado pela estação de controlo ativa para libertar o controlo ativo. Recomenda-se libertar sempre o controlo assim que o canal deixar de ser utilizado ativamente. Caso contrário, este canal bloqueia o controlo ativo de forma permanente.

7.2 Módulo de ligação TCP

7.2.1 Estabelecer ligação

O lado do servidor do módulo de ligação TCP é baseado em [Python](#) e foi implementado utilizando a [biblioteca de sockets](#) integrada. Por predefinição, o servidor está acessível através da **porta 38152**. A mensagem é codificada com o padrão de codificação **UTF-8**. As mensagens são separadas pelo caractere de fim de texto 0x03 (em Python «\x03»). Utilize o seguinte script de exemplo para se ligar ao seu robô e apresentar todas as respostas recebidas. O significado das mensagens recebidas é explicado mais adiante na secção «Receber comandos».

```
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 Enviar comandos

Antes de enviar comandos ao robô, certifique-se de que compreendeu os princípios explicados na secção Ligações, autenticação e controlo ativo. O exemplo seguinte envia os comandos iniciais **token_login** e **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)

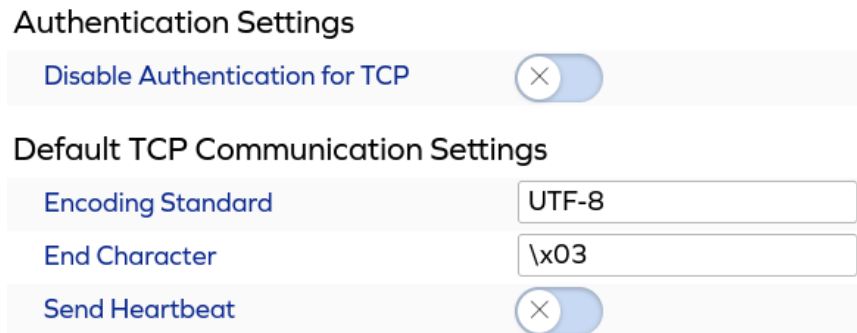
    # make sure to acquire active control, which makes the TCP connection the active
    # control station of the robot (only one control station can be active at any time)
    request = {"action": "request_active_control", "bridge": "core",
               "arguments": {}}
    message = json.dumps(request)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

    time.sleep(0.5)

# add more robot control commands here
```

7.2.3 Configurações

Os exemplos acima utilizam as seguintes configurações TCP padrão, que podem ser alteradas no menu de configurações da interface do ambiente de trabalho. Aqui pode desativar permanentemente a autenticação para TCP, bem como alterar a codificação e o caractere de fim das mensagens TCP. Além disso, pode ativar um heartbeat para verificar continuamente se a ligação continua ativa.



The image shows a configuration interface for TCP settings. It is divided into two sections: 'Authentication Settings' and 'Default TCP Communication Settings'. In the 'Authentication Settings' section, there is a toggle switch for 'Disable Authentication for TCP' which is currently turned on (indicated by a blue bar and a white 'X' icon). In the 'Default TCP Communication Settings' section, there are three items: 'Encoding Standard' set to 'UTF-8', 'End Character' set to '\x03', and 'Send Heartbeat' which is a toggle switch currently turned on (indicated by a blue bar and a white 'X' icon).

Figura 7.2.3: Configurações do módulo TCP

7.2.3.1 Padrão de codificação

Aqui pode definir a codificação das mensagens enviadas através da ligação TCP. São aceites os [padrões de codificação Python](#) habituais. Os padrões utilizados regularmente são UTF-8 (valor padrão) ou ASCII.

7.2.3.2 Caractere de fim

Aqui pode alterar o caractere de fim das mensagens enviadas através da ligação TCP. O caractere de fim sinaliza o fim da mensagem. Ao receber uma mensagem, o caractere de fim é automaticamente removido da mensagem.

7.2.3.3 Enviar heartbeat

Aqui pode definir se o servidor TCP envia automaticamente um heartbeat. Um heartbeat é enviado a cada 50 ms e consiste numa mensagem vazia (é enviado o caractere de fim).

7.2.4 Protocolo de mensagens – Receber comandos

Todos os comandos enviados e recebidos são cadeias de caracteres [JSON](#). Conforme indicado acima, as mensagens são codificadas com o padrão de codificação UTF-8 e separadas pelo caractere de fim de texto 0x03 (em Python «\x03»). Um comando recebido é um dicionário codificado em JSON que contém uma chave (key) e um valor. A chave define o tipo de mensagem. O valor contém as informações da mensagem. Encontrará um exemplo de código que recebe estas mensagens do robô na secção Estabelecer ligação.

Note que esta lista não é exaustiva, mas contém os comandos mais importantes.

7.2.4.1 Informações de estado

O estado do robô é enviado assim que este se altera.

Chave	Valor	Tipo
status_info	O estado atual do robô, de acordo com a lista seguinte: ● 0 → Fora de serviço ● 1 → A ligar ● 2 → A carregar ● 3 → Pronto ● 4 → Em processamento ● 5 → Calibração ● 6 → Em execução ● 7 → Modo autónomo ● 8 → Em pausa ● 9 → Será retomado ● 10 → Paragem de segurança ● 11 → Paragem de segurança ● 12 → Limitação de posição ● 13 → Controlo manual ● 14 → Colisão ● 15 → Paragem de emergência ● 16 → Paragem normal ● 17 → Erro ● 18 → Desligar ● 19 → Parado	Inteiro

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

7.2.4.2 Valor de retorno da ação

Após a execução de uma ação, o valor de retorno é enviado juntamente com o comando da ação. Se a ação não tiver um valor de retorno, este é enviado como zero. A ação só é bem-sucedida se não tiver ocorrido qualquer erro durante a execução da função. Por fim, o número de sequência é um parâmetro opcional, que só é adicionado se o utilizador tiver especificado uma sequência ao enviar o comando.

Chave	Valor	Tipo
action_return_value	O nome da ação de retorno, o valor de retorno e se a ação foi bem-sucedida. Em caso de erro, a ação não foi bem-sucedida.	Dicionário

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

7.2.4.3 Erros

Caso ocorra um erro, é enviada uma mensagem de erro com informações mais detalhadas.

Chave	Valor	Tipo
error	As informações sobre o erro.	Dicionário

```
{"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 Histórico de mensagens

Lista das mensagens anteriores enviadas para a interface do utilizador.

Chave	Valor	Tipo
message_history	As mensagens contêm texto, título, código e carimbo de data/hora.	Lista

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

7.2.4.5 Definição de um grupo de estados

Define os dados de posição do robô. Um grupo de estados contém um ID que indica a que se referem os dados incluídos. Com base nas definições, os dados podem ser identificados.

Chave	Valor	Tipo
state_group_define	Definição do grupo de estados.	Dicionário

```
{
  "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 Dados de um grupo de estado

Os dados de um grupo de estado contêm dados TCP ou do motor. Utilize «request_data» para receber estes dados.

Chave	Valor	Tipo
state_group_data	O valor contém o TCP do robô (id = «pose»), os dados do motor (id = «robot») ou os estados de segurança (id = «safety»).	Dicionário

```
{
  "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 Abrir caixa de diálogo

Quando uma caixa de diálogo é aberta na interface do utilizador, é recebido o comando «open_dialog». A caixa de diálogo pode, em seguida, ser respondida com a ação «answer_dialog».

Chave	Valor	Tipo
open_dialog	A caixa de diálogo assume os seguintes valores: • id → ID única da caixa de diálogo (para responder à caixa de diálogo) • kind ○ 0 → caixa de diálogo de confirmação (botões para confirmar/recusar) ○ 1 → caixa de diálogo de entrada (introdução de texto) ○ 2 → Diálogo de ficheiros (carregamento de ficheiros) ○ 3 → Diálogo suspenso (requer lista de opções) ○ 4 → Diálogo de seleção (requer lista de opções) • padrão → Valor de retorno, caso não seja recebida qualquer resposta • cor ○ 0 → Informação ○ 1 → Sucesso ○ 2 → Aviso ○ 3 → Erro • title → título a exibir • text → texto a exibir • options → Lista de opções para diálogos suspensos/de seleção	Dicionário

```
{"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 Enviar a partir da aplicação

Recebe uma mensagem enviada por uma aplicação.

Chave	Valor	Tipo
script_send	Valor enviado pela aplicação	Vários

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

7.2.5 Registo de mensagens – Enviar comandos

Todos os comandos enviados e recebidos são cadeias de texto [JSON](#). Conforme indicado acima, as mensagens são codificadas com o padrão **UTF-8** e separadas pelo **caractere de fim de texto** 0x03 (em Python, «\x03»). Um comando enviado deve conter um dicionário codificado em JSON com três chaves obrigatórias (e uma opcional). Encontrará um exemplo de código que envia mensagens para o robô na secção Enviar comandos.

action (função)	define qual o comando a ser executado	str
bridge (módulo)	define o módulo que contém o comando, normalmente «core»	str
arguments (argumentos)	definem os argumentos do comando; consulte abaixo as descrições detalhadas dos argumentos	dict
sequence (identificador)	<opcional> Identificador do comando, que é devolvido no final em action_result_value	int

Note que a lista seguinte não é exaustiva, mas contém os comandos mais importantes.

7.2.5.1 Login com token

O primeiro comando a enviar deve ser sempre uma chamada de login para utilizar a ligação ativa. Siga o procedimento de autenticação remota descrito na secção Autenticação segura para obter um token de acesso válido.

Função	token_login	
Módulo	core	
Argumentos	token	session_id
Valor do argumento	o token de início de sessão personalizado fornecido pelo comando de início de sessão	o ID de sessão único fornecido pelo comando de início de sessão
Tipo de argumento	String	String

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

7.2.5.2 Solicitar controlo ativo

Obter o controlo ativo, de modo a que a ligação TCP passe a ser a estação de controlo ativa do robô. Tenha em atenção que, neste caso, é enviada uma solicitação à estação de controlo ativa atual, a qual pode ser aceite ou recusada.

Função	request_active_control
Módulo	core

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

7.2.5.3 Liberar o controle ativo

Liberar o controle ativo, de modo a que a ligação TCP deixe de ser a estação de controle ativa do robô.

Função	release_active_control
Módulo	core

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

7.2.5.4 Obter dados

Este comando pode ser utilizado como Heartbeat. Recebe vários tipos de dados.

Função	request_data
Módulo	core

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

7.2.5.5 Posição de funcionamento

Faz com que o robô ligue os seus motores.

Função	power_up
Módulo	core
Argumentos	tool_id
Valor do argumento	ID de uma ferramenta definida pelo utilizador ou 0 para a deteção automática de ferramentas
Tipo de argumento	Inteiro ou cadeia de caracteres

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

7.2.5.6 Desligar

Encerra a ligação e faz com que o robô desligue os seus motores.

Função	power_down
Módulo	core

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

7.2.5.7 Pausar

Pausa todas as funções de movimento e aplicações em execução.

Função	pause
Módulo	core

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

7.2.5.8 Continuar

Retoma uma aplicação ou movimento, caso este esteja em pausa.

Função	resume
Módulo	core

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

7.2.5.9 Ativação do controlo manual (HGC)

Liberta as articulações do robô para que estas fiquem no modo de compensação gravitacional e possam ser movidas manualmente. O estado deve ser «Pronto» e muda para «Controlo manual».

Função	release
Módulo	core
Argumentos	joint_ids
Valor do argumento	<opcional> Lista dos IDs dos eixos para os quais o modo manual deve ser ativado. Por predefinição, o modo manual é ativado para todos os eixos.
Tipo de argumento	Lista

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

7.2.5.10 Desativar o controlo manual (HGC)

Mantém as articulações do robô de modo a que estas se encontrem no modo de controlo de posição e não possam ser mexidas manualmente. O estado deve ser «Controlo manual» e muda para «Pronto».

Função	hold
Módulo	core
Argumentos	joint_ids
Valor do argumento	<opcional> Lista dos IDs dos eixos a manter. Por predefinição, todos os eixos são mantidos.
Tipo de argumento	Lista

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

7.2.5.11 Parar

Interrompe todas as funções de movimento e aplicações em execução.

Função	stop
Módulo	core

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

7.2.5.12 Mudar de projeto

Altera o projeto de base de dados atual para o projeto com um determinado ID. O estado deve ser «Pronto».

Função	change_project
Módulo	core
Argumentos	project_id
Valor do argumento	ID do projeto para o qual se pretende mudar. O projeto padrão tem o ID 1.
Tipo de argumento	Inteiro

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

7.2.5.13 Executar aplicação

Executa uma aplicação especificada pelo seu nome ou ID. O estado deve ser «Pronto» e muda para «Em execução». Enquanto uma aplicação está em execução, o robô pode ser pausado. Assim que terminar, o estado volta a «Pronto».

Função	execute_script	
Módulo	core	
Argumentos	script_name	script_id
Valor do argumento	O nome da aplicação a ser executada.	O ID da aplicação a ser executada.
Tipo de argumento	String	Inteiro

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

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

7.2.5.14 Testar a aplicação

Executa o código de programação de uma aplicação no modo de teste. O estado deve estar em «Pronto» e muda para «Em execução». Enquanto uma aplicação está em execução, o robô pode ser pausado. Assim que terminar, o estado volta a «Pronto».

Função	test_script		
Módulo	core		
Argumentos	script_code	script_type	script_id
Valor do argumento	O código Python da aplicação a ser testada.	O tipo do script: • «main» • «parallel» • «background»	O ID do script a ser testado. Caso o script não exista na base de dados, o ID é 0.
Tipo de argumento	String	String	Inteiro

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

7.2.5.15 Receber mensagem

Utilize esta função para enviar uma mensagem do seu canal de ligação para uma aplicação. A mensagem pode ser recebida através da função **receive_message** na aplicação.

Função	script_receive
Módulo	core
Argumentos	message
Valor do argumento	Mensagem recebida pela aplicação.
Tipo de argumento	Vários

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

7.2.5.16 Deslocar para a pose

Mova o robô para uma pose com um ID ou nome específico. O estado deve ser «Pronto» e muda para «Em execução». Enquanto o robô se move, o movimento pode ser pausado. Assim que o movimento estiver concluído, o estado volta a «Pronto».

Função	move_pose
Módulo	core
Argumentos	pose_id

Valor do argumento	ID ou nome da pose para a qual se pretende deslocar.
Tipo de argumento	Inteiro ou cadeia de caracteres

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

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

7.2.5.17 Reproduzir percurso

Reproduz o percurso com um ID ou nome específico. O estado deve estar em «Pronto» e muda para «Em execução». Enquanto o robô se move, o movimento pode ser pausado. Assim que o movimento estiver concluído, o estado volta a «Pronto».

Função	play_path
Módulo	core
Argumentos	path_id
Valor do argumento	ID ou nome do caminho a ser executado.
Tipo de argumento	Inteiro ou cadeia de caracteres

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

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

7.2.5.18 Alternar ferramenta

Altera a função da ferramenta, normalmente entre «Desfazer» e «Colocar». O estado deve estar em «Pronto» e muda para «Em execução». Assim que a função estiver concluída, o estado volta a «Pronto».

Função	toggle_tool
Módulo	core
Argumentos	pick
Valor do argumento	true para retirar um objeto, false para colocar um objeto. Isto é determinado pelo comportamento de prensão.
Tipo de argumento	Booleano

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

7.2.5.19 Gravação de saídas digitais e analógicas

Descreve uma ou mais saídas (digitais ou analógicas) do controlador principal ou do controlador da ferramenta (linhas de sinal que são passadas do controlador principal para a ferramenta). Para cada uma destas combinações, é utilizada uma função específica (com os mesmos argumentos). Encontrará uma lista de todas as E/S no menu de configuração de E/S ou no menu de E/S.

Função	write_digital_main_output write_analog_main_output write_digital_tool_output write_analog_tool_io	
Módulo	core	
Argumentos	identificadores	values
Valor do argumento	Existem várias formas de identificar uma única E/S, seja por • número (inteiro) • nome definido pelo utilizador (cadeia de caracteres) Se forem gravadas várias E/S simultaneamente, pode combinar números ou nomes definidos pelo utilizador numa lista.	Dependendo do tipo de «identificadores» selecionado, os valores podem ser: • true/false (booleano) para E/S digitais • Números (Float) para E/S analógicas ou uma lista destes valores, quando várias E/S são gravadas simultaneamente.
Tipo de argumento	Inteiro / String / Lista	Booleano / Flutuante / Lista

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

Este exemplo descreve a saída digital da ferramenta 3 e define-a como true.

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

Este exemplo descreve a saída analógica da ferramenta, à qual foi atribuído o nome definido pelo utilizador «my_analog_output», e define-a para 9,5 V.

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

Este exemplo descreve as saídas digitais principais 1 e 2 e define 1 como verdadeiro e 2 como falso.

7.2.5.20 Responder a uma caixa de diálogo

Responde a uma janela de diálogo aberta. A resposta depende do tipo de diálogo.

Função	answer_dialog	
Módulo	core	
Argumentos	dialog_id	dialog_answer
Valor do argumento	O identificador único do diálogo para o qual a resposta é enviada. O ID faz parte da resposta «open_dialog» recebida.	A resposta a enviar para este diálogo.
Tipo de argumento	String	Booleano / Inteiro / String

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

7.2.5.21 Definir variável de estado

Define o valor de uma variável de estado «variable_name» como «variable_value». A variável já deve existir na aplicação com o ID «script_id». Além disso, a aplicação deve estar guardada na base de dados. Este comando pode ser executado enquanto uma aplicação está em execução.

Função	set_state_variable		
Módulo	core		
Argumentos	script_id	nome_da_variável	valor_da_variável
valor do argumento	ID da aplicação que contém a variável de estado a ser alterada.	Nome da variável de estado a ser alterada.	Novo valor.
Tipo de argumento	Inteiro	String	Diversos

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

7.2.5.22 Definir variável global

Define o valor de uma variável global «name» como «value». As variáveis globais são guardadas para além do tempo de execução de uma aplicação e permanecem mesmo após o reinício do robô.

Função	set_global_variable	
Módulo	core	
Argumentos	nome	valor
Valor do argumento	Nome da variável global	Valor da variável global
Tipo do argumento	String	Vários

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

7.2.5.23 Ler variável global

Lê o valor de uma variável global «name». As variáveis globais são guardadas para além do tempo de execução de uma aplicação e permanecem mesmo após o reinício do robô.

Função	get_global_variable	
Módulo	core	
Argumentos	nome	
Valor do argumento	Nome da variável global	
Tipo do argumento	String	

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

7.2.5.24 Definir variável partilhada

Define o valor de uma variável partilhada «name» como «value». As variáveis partilhadas só estão disponíveis durante o tempo de execução de uma aplicação. São eliminadas após o encerramento da aplicação ou a paragem do robô.

Função	set_shared_variable	
Módulo	core	
Argumentos	nome	valor
Valor do argumento	Nome da variável partilhada	Valor da variável partilhada
Tipo de argumento	String	Vários

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

7.2.5.25 Ler variável partilhada

Lê o valor de uma variável partilhada «name». As variáveis partilhadas só estão disponíveis durante o tempo de execução de uma aplicação. São eliminadas após o encerramento da aplicação ou a paragem do robô.

Função	get_shared_variable	
Módulo	core	
Argumentos	nome	
Valor do argumento	Nome da variável partilhada	
Tipo de argumento	String	

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

7.3 Módulo de ligação REST

7.3.1 Estabelecer ligação

O módulo de ligação REST disponibiliza o recurso **https://<IP-ADDRESS>/rest/gateway** para receber comandos de clientes ligados. Tenha em atenção que existe apenas um único canal de ligação. Se vários utilizadores se ligarem via REST, partilham este canal.

Utilize o seguinte script de exemplo para assumir o controlo do robô através da ligação REST e, em seguida, enviar comandos ao robô. Tenha em atenção que, antes de enviar comandos, tem de estar autenticado e ter o controlo ativo. Por isso, recomenda-se que partilhe o controlo e termine a sessão assim que a ligação já não for necessária.

```
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 Enviar comandos

Para além dos comandos mencionados no exemplo acima, estão disponíveis muitos outros comandos. Consulte a lista de comandos no módulo de ligação TCP, uma vez que os mesmos comandos também são utilizados para a ligação REST.

Para além de enviar um parâmetro «bridge», «action», «arguments» e um parâmetro opcional «sequence», também pode enviar um parâmetro opcional «timeout». Se enviar um timeout, a solicitação REST, caso seja enviada com sucesso, não fornecerá a resposta padrão (200, {}), mas sim o comando será executado de forma bloqueante, e receberá a resposta como (200, action_return_value), em que o valor de retorno da ação contém a informação sobre qual a função API que foi chamada, se a chamada foi bem-sucedida e qual o valor de retorno devolvido pela função chamada.

7.3.3 Configurações

No menu de configurações da interface de desktop, encontrará uma opção para desativar a autenticação do módulo de ligação REST. Ao desativar a autenticação, deixa de ser necessário iniciar e terminar a sessão.

Authentication Settings

Disable Authentication for REST



Figura 7.3.3: Configurações do módulo REST

7.4 Módulo de ligação ROS

O módulo de ligação ROS permite-lhe integrar o seu robô num ambiente ROS 2 (Robot Operating System) e comunicar com o AX Portal através de comandos ROS 2. As instruções deste capítulo são compatíveis com o ROS Jazzy e [e](#) [o](#).

 AVISO	O módulo ROS oferece acesso a determinadas funções de baixo nível. Por exemplo, é possível enviar comandos de controlo diretos aos motores do robô, que são executados imediatamente. Na interface do utilizador, algumas funções requerem verificações adicionais ou confirmações por caixa de diálogo, que podem não estar disponíveis aqui. Por isso, tenha especial cuidado ao controlar o seu robô diretamente através do ROS.
---	--

Nota:

- Esta é a primeira versão do AX Portal com o ROS 2; estão previstas funcionalidades adicionais no futuro.
- Atualmente, o módulo ROS não está disponível com o simulador do AX Portal, mas apenas com robôs reais.

7.4.1 Instalação

Para comunicar com o módulo ROS do AX Portal, necessita de ter o ROS instalado no(s) seu(s) computador(es). Para a configuração, a [disponibiliza](#) nado ROS várias opções, por exemplo, pacote deb, Docker image, instalação a partir do código-fonte, etc. Neste exemplo, utilizamos a imagem Docker oficial do ROS para comunicar com o robô. O seguinte ficheiro docker-compose cria um ambiente ROS simples no seu computador:

```
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"
```

Note que, ao iniciar, instalamos o `ros-jazzy-rmw-cyclonedds-cpp` e definimos `RMW_IMPLEMENTATION` como `rmw_cyclonedds_cpp` para instruir o ROS a utilizar o CycloneDDS – o mesmo RMW que também está a ser executado no robô. Além disso, partilhamos um ficheiro `cyclonedds_config.xml` e definimos `CYCLONEDDS_URI` para a sua localização, a fim de utilizar uma configuração personalizada do CycloneDDS.

7.4.2 Configuração

A comunicação ROS entre o robô e o seu computador requer ajustes em ambos os dispositivos. No seu portátil, terá de definir uma nova variável de ambiente e disponibilizar um ficheiro de configuração de rede. No robô, não tem acesso direto a estes elementos, mas pode editá-los na interface do utilizador, na secção «ROS» do menu de configurações:

The image shows a user interface titled "ROS Function Settings". It contains several configuration options:

- Network Interface:** A dropdown menu currently showing "Wifi".
- Multicast:** A toggle switch that is currently turned off (indicated by a grey circle with an 'X').
- IP Addresses of ROS 2 Peers:** A list of input fields. The first two contain "192.168.0.1" and "192.168.0.2". Each field has a trash icon to its right. Below the list is a blue button with a "+" sign to add more addresses.
- ROS Domain ID:** A dropdown menu currently showing "15".

Figura 7.4.2: Configurações do módulo ROS

Nas secções seguintes, irá conhecer o significado de cada uma das definições. Além disso, irá saber quais os ajustes correspondentes necessários no seu computador.

7.4.3 Detecção de rede com o Cyclone DDS

O Cyclone DDS já foi mencionado várias vezes anteriormente. Do que se trata? Como utilizador do AX Portal, deve saber o seguinte:

Quando o Cyclone DDS é selecionado como middleware ROS, ele assume a tarefa de distribuir pacotes de dados relacionados com o ROS na sua rede. É responsável por estabelecer uma ligação de rede entre os seus dispositivos, para que o ROS 2 possa enviar dados através dessas ligações.

7.4.3.1 Interface de rede

Em primeiro lugar, o Cyclone DDS precisa de saber qual a interface de rede utilizada pelo ROS. A predefinição é Ethernet.

Utilize a configuração «Interface de rede» em «Configurações para funções ROS» para selecionar a interface pretendida.

Nota: Ao iniciar o módulo ROS através do AX Portal, o Cyclone DDS analisa a interface de rede em busca de endereços IP de computadores pares. Se esta interface não for acessível, o módulo ROS não será carregado. Por isso, se tiver definido a interface de rede como Ethernet, certifique-se de que o cabo está ligado antes de iniciar o robô.

7.4.3.2 Métodos de detecção

O Cyclone DDS oferece dois métodos para a detecção de outros dispositivos na rede: Multicast e Peer-to-Peer. No multicast, cada pacote de rede é enviado a todos os clientes acessíveis na área de rede. A vantagem reside na simplicidade: não é necessário definir quais os computadores que podem comunicar entre si. Nesta configuração, a variável de ambiente ROS_DOMAIN_ID (ver abaixo) continua disponível para impedir que dois robôs misturem mensagens.

A desvantagem é a sobrecarga da rede: todos os pacotes são processados por todos os clientes da rede. Mesmo a definição de diferentes ROS_DOMAIN_IDs não consegue impedir isso. Mesmo que não esteja instalado o ROS num computador: uma vez que o Cyclone DDS funciona com base em UDP, todos os computadores na rede recebem e processam, na mesma, estes pacotes UDP. Para redes mais pequenas e volumes de dados reduzidos, isso não cria, normalmente, um problema. No entanto, se um robô enviar, por exemplo, um fluxo de câmara para todos os clientes numa rede de grandes dimensões, isso pode tornar toda a rede visivelmente mais lenta. Nesse cenário, é mais seguro definir explicitamente na configuração do Cyclone DDS um conjunto de endereços IP que possam comunicar entre si através de ligações ponto a ponto.

Nas duas subsecções seguintes, irá aprender como configurar a sua rede para comunicação multicast ou ponto a ponto. Nota: Na maioria das distribuições Linux, o comando `ip addr show` é útil para identificar os seus endereços IP e os nomes das interfaces de rede associadas.

7.4.3.2.1 Peer-to-Peer

Nas instruções de instalação acima, incluímos um ficheiro de configuração do Cyclone DDS no ficheiro `docker-compose`. Aqui está um exemplo de configuração para o Cyclone DDS, para configurar o seu computador para comunicação ponto a ponto:

```
<?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" />
      </Peers>
    </Discovery>
  </Domain>
</CycloneDDS>
```

```
<Peer address="192.168.1.2" />
</Peers>
</Discovery>
</Domain>
</CycloneDDS>
```

Primeiro, substitua `INTERFACE_NAME` pelo nome real da interface de rede do seu computador que é utilizada para a comunicação ROS. Exemplos: `enp1s0`, `wlp2s0`, etc.

Em seguida, adapte a lista em «Peers» de modo a que esta reflita os endereços IP dos computadores na sua rede que pretende ligar entre si através do ROS.

À semelhança do ficheiro acima, existe um ficheiro semelhante guardado no seu robô. Não pode aceder diretamente a este, mas pode editar o seu conteúdo através do AX Portal, na secção «Configurações para funções ROS» do menu de configurações. Aí, a configuração «Endereços IP de pares ROS 2» disponibiliza uma lista de campos de texto que pode preencher com endereços IP.

Tenha em atenção:

- É necessário reiniciar o robô para que a alteração entre em vigor.
- Se um endereço IP na lista estiver incorreto, o AX Portal exibirá um erro e a nova lista não será guardada.
- A lista deve conter o endereço IP do próprio robô. Atualmente, este não é lido automaticamente.
- Se um endereço IP for removido da lista e a nova lista for guardada, o endereço IP será igualmente removido da configuração do Cyclone DDS.
- Devido às configurações da firewall, pode ser necessário que todos os participantes do ROS se encontrem na mesma sub-rede.

7.4.3.2.2 Multicast

Aqui está outra configuração de exemplo para o Cyclone DDS, para configurar o seu computador para multicast:

```
<?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>
```

Substitua também aqui `INTERFACE_NAME` pelo nome real da interface de rede do seu computador, utilizada para a comunicação ROS. Exemplos: `enp1s0`, `wlp2s0`, etc.

Para configurar o seu robô para multicast, basta ativar o botão de configuração «Multicast». Também neste caso é necessário reiniciar o robô para que a alteração entre em vigor.

7.4.4 Estabelecer ligação

Para iniciar agora o ambiente ROS no seu computador, navegue até ao seu ficheiro `docker-compose` e execute:

```
docker compose up -d
```

Verifique se o contentor Docker está em execução com

```
docker ps -a
```

o que gera uma saída semelhante à seguinte:

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

Mude para o modo interativo do contentor com o seguinte comando:

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

Não se esqueça de carregar o ficheiro de configuração do ROS sempre que entrar no contentor:

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

Para poder operar vários robôs na mesma rede, utilizamos um `ROS_DOMAIN_ID` diferente em cada robô. No AX Portal, abra as «Definições para funções ROS» e defina o ROS Domain ID pretendido. O intervalo permitido situa-se entre 6 e 41. Tenha também em atenção que, após esta ação, o robô terá de ser reiniciado. Regresse agora ao terminal onde o seu contentor Docker ROS está aberto e defina o mesmo valor para `ROS_DOMAIN_ID`:

```
export ROS_DOMAIN_ID=SELECTED_NUMBER
```

Deve agora conseguir ver um nó AX Portal-ROS chamado «robot» com o seguinte comando:

```
ros2 node list
```

7.4.5 Acesso aos dados do robô através do ROS

Assim que vir o nó, pode visualizar a lista de tópicos disponíveis executando o seguinte:

```
ros2 topic list
```

Através dos tópicos ROS 2, pode transmitir dados em tempo real do seu robô. Por exemplo, para subscrever posições de eixos, velocidades, etc., execute o seguinte:

```
ros2 topic echo /joint_states
```

7.4.6 Funcionalidades do ROS 2 em scripts

Pode importar o rclpy diretamente para os scripts. Isto permite-lhe criar objetos ROS 2, tais como editores, assinantes, etc.

O módulo ROS cria o seu próprio nó ROS. Utilize a função `ros_handler.ros_node()` para aceder ao mesmo. Atenção: não execute o `rclpy.spin` neste nó, uma vez que isso já ocorre dentro do módulo ROS.

7.4.7 Chamar uma ação

O nó ROS do AX Portal cria um servidor de ação denominado «/robot/execute_function» para a execução de funções de ponte. A ação tem uma definição personalizada `ExecuteFunction` do pacote `generic_msgs` (<https://github.com/Epson-Robots/epson-ax-ros2>), que deve ser instalado para obter a definição da ação. Clone o pacote para o seu espaço de trabalho e compile-o com o `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
```

Para enviar ações ROS2 ao robô, o seu cliente deve assumir o controlo ativo. Para tal, execute o comando abaixo no seu terminal e confirme a caixa de diálogo na GUI do AX Portal:

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

Agora pode executar uma função de ponte a partir de um terminal com os seguintes comandos:

```
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 Configurações de autenticação

O módulo ROS funciona atualmente apenas sem autenticação. No menu de configurações da interface de secretária, terá de desativar manualmente a autenticação para este módulo de ligação específico.

Authentication Settings

Disable Authentication for ROS



Figura 7.4.8: Configurações de autenticação do ROS

7.5 Módulo de ligação Modbus

7.5.1 Visão geral

O protocolo utilizado pelo módulo Modbus é [o Modbus TCP](#) . Do lado do servidor, o módulo baseia-se em Python e foi implementado com a ajuda da [biblioteca PyModbus](#) . Encontrará mais detalhes na [documentação PyModbus](#). O módulo está dividido em registos de leitura (Analog Input Registers, 30000-39999) e registos de escrita (Analog Output Holding Registers, 40000-49999). Para comunicar com o servidor, pode ser utilizado qualquer cliente TCP Modbus. A porta a utilizar é a 5020. Na última secção deste capítulo, encontrará vários exemplos de código Python, alguns dos quais explicam como estabelecer uma ligação ao servidor utilizando um script Python.

7.5.2 Configurações

Atualmente, a ligação Modbus funciona apenas sem autenticação. No menu de configurações da interface de secretária, terá de desativar manualmente a autenticação para este módulo de ligação específico.

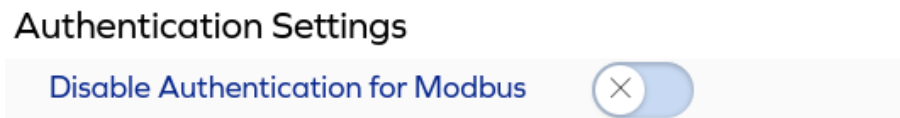


Figura 7.5.2: Configurações do módulo Modbus

7.5.3 Registos de escrita (descrição simples)

O módulo Modbus reage a qualquer alteração de informação, tanto no registo de comandos como no registo de propriedades. A eliminação dos dados dos registos de escrita é da responsabilidade do utilizador.

7.5.3.1 Registo de comandos

Escreva um único bit no registo de comandos (30000) para executar um comando. Alguns comandos requerem um argumento no registo de dados de comando (30001). Caso não consiga escrever em ambos os registos simultaneamente, certifique-se de que escreve primeiro no registo de dados de comando.

Endereço absoluto: 30000 - 30001 Endereço relativo: 0x000 - 0x001		
Bit	Registo de comandos (30000)	Dados (30001)
0	Colocação em funcionamento	ID da ferramenta
1	Colocar fora de serviço	
2	Parar	
3	Pausar	
4	Continuar	
5	Calibrar	ID da aplicação
6	Ativar o controlo manual (HGC)	
7	Desativar o controlo manual (HGC)	
8	Mudar de projeto	ID do projeto
9	Executar aplicação	ID da aplicação
10	Deslocar-se para a pose	ID da pose
11	Reproduzir trajetória	ID do caminho
12	Colocar ferramenta	
13	Retirar ferramenta	
14 ... 15	<i>reservado</i>	

7.5.3.2 Registo de propriedades

Escreva um único bit no registo de propriedades (30002) para executar um comando. Alguns comandos requerem um argumento no registo de dados de propriedades (30003). Caso não consiga escrever em ambos os registos simultaneamente, certifique-se de escrever primeiro no registo de dados de propriedades.

Endereço absoluto: 30002 - 30003 Endereço relativo: 0x002 - 0x003		
Bit	Registo de propriedades (30002)	Dados (30003)
0	Gravar saídas digitais	Estado de E/S
1	Definir eventos de aplicação	ID do evento
2	Confirmar erro	
3	Ler posição	
4	Solicitar controlo ativo	
5	Autorizar controlo ativo	
6	Mudar para o modo manual	
7	Mudar para o modo automático	
8	Reiniciar a função de segurança	
9	Continuar o movimento	
10 ... 15	<i>reservado</i>	

7.5.3.3 Registos de escrita definidos pelo utilizador

Endereço absoluto: 30016 Endereço relativo: 0x010		...	Endereço absoluto: 30255 Endereço relativo: 0x0FF	
0 ... 15	Dados definidos pelo utilizador		0 ... 15	Dados definidos pelo utilizador

Utilize os registos de escrita do utilizador para enviar informações para a sua aplicação. A função **modbus.write_user_register** na aplicação permite a leitura de dados definidos pelo utilizador.

7.5.4 Registos de escrita (descrição detalhada)

7.5.4.1 Não fazer nada

A limpeza dos dados dos registos de escrita é da responsabilidade do utilizador. Para limpar o registo de comandos (30000) ou o registo de propriedades (30002), basta escrever o valor inteiro 0.

0x000	0x001	0x002	0x003
0		0	

7.5.4.2 Colocação em funcionamento

Antes de começar a trabalhar com o robô, este deve ligar os seus motores. A identificação da ferramenta personalizada a ligar é indicada pelo número n. O robô altera o seu estado de «**Fora de serviço**» (1, bit 0) para «**Pronto**» (8, bit 3).

0x000	0x001	0x002	0x003
1	n		

7.5.4.3 Desligar

Quando o dia de trabalho terminar, o robô pode desligar os seus motores. O robô pode ser desligado em qualquer estado. Assim que o robô for desligado, o estado muda para **Fora de serviço** (1, bit 0).

0x000	0x001	0x002	0x003
2			

7.5.4.4 Parar

Interrompa todos os movimentos do robô e altere o estado de **Em execução** (64, bit 6) de volta para **Pronto** (8, bit 3).

0x000	0x001	0x002	0x003
4			

7.5.4.5 Pausar

Pare o robô enquanto este se encontra no estado **Em funcionamento** (64, bit 6). O estado muda para **Em pausa** (128, bit 7).

0x000	0x001	0x002	0x003
8			

7.5.4.6 Continuar

Retome o funcionamento do robô se este se encontrar no estado «**Em pausa**» (128, bit 7). O estado muda para «**Em funcionamento**» (64, bit 6).

0x000	0x001	0x002	0x003
16			

7.5.4.7 Calibração

A calibração não tem qualquer efeito neste momento. Este bit está reservado.

0x000	0x001	0x002	0x003
32			

7.5.4.8 Ativar o controlo manual (HGC)

Ative o controlo manual para todas as articulações do robô, de modo a que estas se encontrem no modo de compensação da gravidade e possam ser movidas manualmente. O estado deve ser **Pronto** (8, bit 3) e muda para **Controlo manual** (2048, bit 11).

0x000	0x001	0x002	0x003
64			

7.5.4.9 Desativar o modo de operação manual (HGC)

Mantenha todas as articulações do robô de modo a que se encontrem no modo de regulação de posição e não possam ser movidas manualmente. O estado deve ser **Controlo Manual** (2048, Bit 11) e muda para **Pronto** (8, Bit 3).

0x000	0x001	0x002	0x003
128			

7.5.4.10 Mudar de projeto

Mude o projeto de base de dados atual para o projeto com o ID n. O estado deve ser **Pronto** (8, bit 3).

0x000	0x001	0x002	0x003
256	n		

7.5.4.11 Iniciar a aplicação

Execute a aplicação com o ID n. O estado deve estar em **Pronto** (8, bit 3) e muda para **Em execução** (64, bit 6). Enquanto uma aplicação está em execução, o robô pode ser colocado em pausa. Assim que terminar, o estado volta a ficar em **Pronto** (8, bit 3).

0x000	0x001	0x002	0x003
512	n		

7.5.4.12 Deslocar para a pose

Mova o robô para a pose com o ID n. O estado deve estar em **Pronto** (8, bit 3) e muda para **Em execução** (64, bit 6). Enquanto o robô se move, o movimento pode ser pausado. Assim que o movimento terminar, o estado volta a ficar em **Pronto** (8, bit 3).

0x000	0x001	0x002	0x003
1024	n		

7.5.4.13 Reproduzir percurso

Reproduza o percurso com o ID n. O estado deve estar em **Pronto** (8, bit 3) e muda para **Em execução** (64, bit 6). Enquanto o robô se move, o movimento pode ser pausado. Assim que o movimento terminar, o estado volta a ficar em **Pronto** (8, bit 3).

0x000	0x001	0x002	0x003
2048	n		

7.5.4.14 Colocar ferramenta

Posicione um objeto com a ajuda da ferramenta (por exemplo, abrindo a pinça). O estado deve estar em **Pronto** (8, bit 3) e muda para **Em execução** (64, bit 6). Enquanto a pinça se move, o movimento pode ser pausado. Assim que o movimento terminar, o estado volta a **Pronto** (8, bit 3).

0x000	0x001	0x002	0x003
4096			

7.5.4.15 Levantar a ferramenta

Levante um objeto com a ajuda da ferramenta (por exemplo, fechando a pinça). O estado deve estar em **Pronto** (8, bit 3) e muda para **Em execução** (64, bit 6). Enquanto a pinça se move, o movimento pode ser pausado. Assim que o movimento terminar, o estado volta para **Pronto** (8, bit 3).

0x000	0x001	0x002	0x003
8192			

7.5.4.16 Gravação de saídas digitais

Função reservada para a gravação de saídas digitais.

0x000	0x001	0x002	0x003
		1	d

7.5.4.17 Definir eventos de aplicação

Se uma aplicação do utilizador tiver de aguardar um evento enviado pelo dispositivo Modbus, podem ser utilizados os eventos de aplicação. Estão disponíveis, no total, 16 eventos. Dependendo do valor do dado d, os eventos são definidos ou eliminados. A função de **aplicação modbus.wait_for_event** aguarda até que um determinado evento seja definido. Se d for igual a 0, todos os eventos são eliminados. Se d for igual a 65535, todos os eventos são definidos. Se d for igual a 9, o primeiro e o quarto eventos são definidos, sendo os restantes eliminados.

0x000	0x001	0x002	0x003
		2	d

7.5.4.18 Confirmar erros

Os erros têm de ser confirmados antes de serem apagados. No entanto, isto só é necessário para ler os erros. O estado do robô não é afetado pela confirmação. Se tanto o código de erro principal como o secundário (40004-40005) forem iguais a 0, não é necessário confirmar mais nenhum erro.

0x000	0x001	0x002	0x003
		4	

7.5.4.19 Leitura da posição

Atualize os dados de posição do robô. Os ângulos atuais das articulações e as coordenadas do robô são gravados nos registos de leitura correspondentes (40256-40267) assim que os dados de posição forem solicitados.

0x000	0x001	0x002	0x003
		8	

7.5.4.20 Solicitar controlo ativo

Solicite o controlo ativo para poder enviar comandos ao robô. É possível aceder ao registo de leitura de dados sem controlo ativo. A interface que detém atualmente o controlo ativo é notificada e deve

conceder o controlo ativo. O registo de leitura correspondente (40008) é definido assim que o controlo for concedido.

0x000	0x001	0x002	0x003
		16	

7.5.4.21 Liberar o controlo ativo

Liberte o controlo ativo para que outra interface possa enviar comandos ao robô. É possível aceder aos registos de leitura de dados e aos registos de leitura sem controlo ativo. O registo de leitura correspondente (40008) muda para 0.

0x000	0x001	0x002	0x003
		32	

7.5.4.22 Mudar para o modo manual

Mude para o modo manual. Neste modo de funcionamento, o robô move-se apenas enquanto o utilizador mantiver premido o botão de confirmação. Tenha em atenção que é necessária uma reinicialização de segurança para a mudança de modo.

0x000	0x001	0x002	0x003
		64	

7.5.4.23 Mudar para o modo automático

Mude para o modo automático. Tenha em atenção que é necessária uma reinicialização de segurança para a mudança de modo.

0x000	0x001	0x002	0x003
		128	

7.5.4.24 Reinicialização de segurança

Reinicie o sistema de segurança após uma interrupção. Para determinadas funções de segurança, o robô requer uma ação clara e intencional do utilizador para continuar. Para verificar se tal ação é necessária, utilize o registo de leitura correspondente (40009).

0x000	0x001	0x002	0x003
		256	

7.5.4.25 Iniciar interrupção de movimento

Após uma interrupção por motivos de segurança, o robô pode retomar o movimento e continuar a sua execução. Com esta funcionalidade, o robô pode ser colocado novamente em movimento. Caso contrário, pode ser enviado um comando de paragem para interromper a aplicação em execução. Para verificar se tal ação é necessária, utilize o registo de leitura correspondente (40009).

0x000	0x001	0x002	0x003
		512	

7.5.5 Registo de leitura

7.5.5.1 Registo de estado

Utilize o registo de estado para ler o estado atual do robô. O estado é fornecido como um número de bit.

7.5.5.1.1 1.º registo de estado

Endereço absoluto: 40000 Endereço relativo: 0x000	
0	Fora de serviço
1	Ligar
2	A carregar
3	Pronto
4	Em processamento
5	Calibração
6	Em execução
7	Modo autónomo
8	Em pausa
9	Continua
10	Paragem de segurança
11	Paragem de segurança
12	Limitação de posição
13	Comando manual (HGC)
14	Colisão
15	Paragem de emergência

7.5.5.1.2 2. Registo de estado

Endereço absoluto: 40001 Endereço relativo: 0x001	
0	Paragem normal
1	Erro
2	Desligar
3	Parado
4.. 15	<i>Reservado</i>

7.5.5.1.3 3. Registo de estado

Endereço absoluto: 40002 Endereço relativo: 0x002	
0 ... 15	<i>reservado</i>

7.5.5.1.4 4. Registo de estado

Endereço absoluto: 40003 Endereço relativo: 0x003	
0 ... 15	<i>reservado</i>

7.5.5.1.5 Registo de erros graves

Endereço absoluto: 40004 Endereço relativo: 0x004	
0 ... 15	Informações sobre a categoria de erro.

O registo de erros graves lê os primeiros 2 bytes do identificador do erro atual. Confirme o erro através do registo de propriedades para ler o erro seguinte.
Encontrará detalhes sobre o significado dos valores do registo de erros na tabela de correspondência dos registos de erros.

7.5.5.1.6 Registo de erros menores

Endereço absoluto: 40005 Endereço relativo: 0x005	
0 ... 15	Informações sobre o erro dentro de uma categoria.

O registo de erros menores lê os últimos 2 bytes do identificador do erro atual. Confirme o erro através do registo de propriedades para ler o erro seguinte.

Encontrará detalhes sobre o significado dos valores do registo de erros na tabela de correspondência dos registos de erros.

7.5.5.1.7 Registo de aplicação

Endereço absoluto: 40006 Endereço relativo: 0x006	
0 ... 15	ID da aplicação atual

O registo de aplicação lê o ID da aplicação atualmente em execução.

7.5.5.1.8 Registo de calibração

Endereço absoluto: 40007 Endereço relativo: 0x007	
0	Se o robô está calibrado

O registo de calibração indica se o robô está calibrado. 0 significa que o robô não está calibrado. 1 significa que o robô está calibrado. Tenha em atenção que o robô tem de estar ligado ao seu motor para fornecer uma resposta significativa sobre a calibração.

7.5.5.1.9 Registo de controlo ativo

Endereço absoluto: 40008 Endereço relativo: 0x008	
0	Se o canal Modbus tem o controlo ativo

O registo de controlo ativo indica se o canal Modbus tem o controlo ativo. 0 significa que o Modbus não tem o controlo ativo. 1 significa que o Modbus tem o controlo ativo. O canal Modbus pode solicitar o controlo ativo através de 0x30002 0x04 Controlo ativo.

7.5.5.1.10 Registo de interrupções

O registo de interrupções apresenta as opções de diálogo de interrupção ativas.

Endereço absoluto: 40009 Endereço relativo: 0x009	
0	Reinicialização de segurança necessária
1	É necessário continuar o movimento

2	É necessário mudar para o modo manual
3.. 15	<i>reservado</i>

7.5.5.1.11 Registo de modos de funcionamento

O registo de modos de funcionamento indica se o robô se encontra no modo manual (1) ou no modo automático (0).

Endereço absoluto: 40010 Endereço relativo: 0x00A	
0	Se o robô está no modo manual

7.5.5.2 Registos de leitura definidos pelo utilizador

Endereço absoluto: 40016 Endereço relativo: 0x010		...	Endereço absoluto: 40255 Endereço relativo: 0x0FF	
0 ... 15	Dados definidos pelo utilizador		0 ... 15	Dados definidos pelo utilizador

Utilize os registos de leitura do utilizador para receber informações da aplicação. A função **modbus.read_user_register** da aplicação lê os dados.

7.5.5.3 Registos de posição (articulações 1 a 6)

Endereço absoluto: 40256 Endereço relativo: 0x100		...	Endereço absoluto: 40261 Endereço relativo: 0x105	
0 ... 15	Ângulo da articulação 1		0 ... 15	Ângulo da articulação 6

Leia o valor do ângulo de uma articulação específica em décimos de grau. Se for lido o valor 101, o ângulo do eixo é de 10,1°. O valor só é atualizado se o sinalizador de leitura de posição for gravado no registo de propriedades. Se forem lidos valores superiores a 32768 (=215), estes correspondem a valores negativos. Subtraia 32768 para obter o valor absoluto. Um valor de 32869 corresponde, por exemplo, a um ângulo de articulação de -10,1°.

7.5.5.4 Registo de coordenadas (x, y, z, roll, pitch, yaw)

Endereço absoluto: 40262 Endereço relativo: 0x106		...	Endereço absoluto: 40267 Endereço relativo: 0x10B	
0 ... 15	Coordenada x do TCP		0 ... 15	Coordenada de guinada do TCP

Leia as coordenadas ou a orientação do centro da ferramenta na ordem x, y, z, roll, pitch, yaw. As coordenadas espaciais são lidas em décimos de milímetro. Se for lida uma coordenada de 314 para x, o valor é de 31,4 milímetros. As coordenadas de rotação são lidas em décimos de grau. Se for lida uma

coordenada de 271, o valor é de 27,1 (mm ou °). O valor só é atualizado quando o sinalizador Read-Position é gravado no registo de propriedades. Se forem lidos valores superiores a 32768 (=215), estes correspondem a valores negativos. Subtraia 32768 para obter o valor absoluto. Um valor de 33039 corresponde, por exemplo, a um valor de coordenada de -27,1 (mm ou °).

7.5.6 Atribuição dos registos de erros

As informações de erro aparecem sempre em dois registos. O registo de erros menores contém o número de erro propriamente dito. O registo de erros graves contém um dos seguintes valores, que são atribuídos às diferentes categorias de erros da seguinte forma:

101	ADDONS
102	CONFIG
103	BASE DE DADOS
104	CONTROLADOR
105	EDITOR
106	BRIDGE
107	CONEXÃO
108	DIALOG
109	DISTRIBUIDOR
110	EVENTO
111	CINEMATICA
112	SOCKET
113	CORE
114	SAFETY
115	CONFIGURAÇÕES
116	SINAL
117	STATUS
118	SCRIPT
119	THREAD

201	BLOCK
202	COGNEX
203	CONTROLO
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	SO
312	PPM
313	PYAPI
314	SIG
315	SYS
316	TLC
317	UTIL

7.5.6.1 Exemplo

Se lerem os registos de erros [113, 4], isso corresponde ao código de erro CORE_004 gerado pelo robô.

7.5.7 Exemplos de código Python

Estes exemplos destinam-se a ser utilizados num sistema externo que se liga remotamente ao robô.

7.5.7.1 Exemplo 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()
```

Ligar-se ao servidor. Ler o estado do robô.

7.5.7.2 Exemplo 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()
```

Ligar-se ao servidor. Ligar o robô aos seus motores.

Nota: O comando << é utilizado para o deslocamento de bits.

Bit	Comando	Inteiro
0	1 << 0	1
5	1 << 5	32

7.5.7.3 Exemplo 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()
```

Ligar-se ao servidor. Executar a aplicação com o ID 5.

7.5.7.4 Exemplo 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()
```

Ligar-se ao servidor. Ler a mensagem de erro atual. Confirmar a mensagem de erro.

8. Módulo de câmara Cognex

8.1 Visão geral

O módulo de câmara Cognex permite-lhe ligar uma câmara Cognex pré-configurada ao AX Portal. Após a ligação, pode acionar as tarefas pré-configuradas na câmara para capturar e processar uma imagem. Os dados processados são posteriormente transferidos para o AX Portal, onde pode ler os dados na sua aplicação.

8.2 Configurar uma câmara

Para configurar uma câmara Cognex para ser compatível com o AX Portal, siga os seguintes passos:

1. Descarregue e instale o software [In-Sight Explorer](#) da Cognex.
2. Ligue a câmara à alimentação e conecte-a à sua rede. Também pode conectá-la diretamente ao seu computador, desde que a configuração de rede esteja definida para um endereço IP estático.
3. Inicie o In-Sight Explorer, autorize o acesso, caso lhe seja solicitado, e instale os componentes necessários (por exemplo, Microsoft Visual C++, Microsoft .NET Framework).
4. Em seguida, a vista Easybuilder do Explorer abre-se automaticamente.
5. Navegue até «Sistema» → «Adicionar Sensor/Dispositivo à Rede». Procure a sua câmara e adicione-a à rede.
6. Certifique-se de que as configurações de rede da câmara são compatíveis com as configurações de rede do robô da série AX. Certifique-se também de que a porta Telnet está definida para 3000.
7. Selecione a sua câmara e clique no botão «Connect» para a ligar ao In-Sight Explorer.
8. Agora está pronto para criar tarefas (por exemplo, para calibrar a câmara e para reconhecer objetos que o robô deve agarrar) e guardá-las na câmara. Para mais informações, consulte a documentação da Cognex.

8.3 Utilização de uma câmara

Para utilizar a câmara da Cognex numa aplicação do AX Portal, é necessário o seguinte:

- O robô e a câmara têm de estar na mesma rede,
- a porta Telnet da câmara deve estar definida para 3000,
- e a câmara deve estar definida como online.

Aqui está um exemplo de como uma tarefa pré-configurada pode ser utilizada para reconhecer as coordenadas do círculo.

```
# 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)
```