

EPSON

AX Portal 用戶手冊

翻譯版

©Seiko Epson Corporation 2026

Rev. 2

TCM265S8836M

目錄

商標.....	8
使用條款.....	8
1. 簡介.....	9
1.1 程式設計與控制.....	10
1.2 圖形化使用者介面 (GUI).....	10
1.3 軟體模塊.....	10
1.3.1 TCP連線模組.....	10
1.3.2 REST連線模組.....	10
1.3.3 ROS連線模組.....	11
1.3.4 Modbus連線模組.....	11
1.3.5 面板介面模塊.....	11
1.3.6 Cognex攝影機模塊.....	11
2. 管理員面板 (GUI).....	12
2.1 簡介.....	13
2.1.1 介面語言.....	13
2.1.2 儀表板.....	14
2.2 使用者管理.....	14
2.2.1 使用者等級.....	14
2.2.2 登陸頁面.....	14
2.2.3 帳戶管理.....	15
2.2.4 管理其他使用者.....	16
2.3 系統設置.....	17
2.3.1 網絡設置.....	17
2.3.2 時間設置.....	18
2.3.3 SSL 憑證.....	18
2.4 軟體更新.....	19
2.5 說明與支援.....	21
2.6 安全驗證.....	21
2.6.1 匯出 SSL 憑證.....	21
2.6.2 使用者登陸.....	22
2.6.3 連接註冊.....	23
3. USB 匯出與復原更新.....	24
3.1 概述.....	25
3.2 USB 匯出.....	25
3.3 執行復原更新.....	25
4. 桌面介面 (GUI).....	26
4.1 簡介.....	27
4.2 主視窗.....	28
4.2.1 選單概覽.....	28
4.2.2 控制台.....	29
4.2.3 活動控制 (單一控制點).....	30

4.2.3.1 活動控制站.....	30
4.2.3.2 被動觀察者.....	31
4.2.4 程式狀態.....	31
4.2.5 操作模式.....	32
4.2.5.1 自動操作模式.....	33
4.2.5.2 手動操作模式.....	33
4.2.6 精細控制.....	34
4.2.7 運動控制.....	35
4.2.7.1 運動限制.....	35
4.2.7.2 全域參數.....	35
4.2.7.3 全域坐標系.....	36
4.3 模塊與資料庫.....	38
4.4 項目菜單.....	38
4.5 配置菜單.....	40
4.5.1 資料庫(下載與上傳).....	40
4.5.2 I/O 配置.....	41
4.5.3 工具配置.....	43
4.5.4 工具編輯器.....	43
4.5.4.1 單輸出模式.....	45
4.5.4.2 雙輸出模式.....	45
4.5.4.3 分離觸發模式.....	45
4.5.5 機器人上電.....	46
4.6 示教菜單.....	47
4.6.1 動作編輯器.....	48
4.6.2 路徑編輯器.....	50
4.6.3 網格編輯器.....	52
4.7 程式菜單.....	55
4.7.1 程序編輯器.....	56
4.7.2 應用程式程式腳本.....	56
4.7.3 程式設置.....	57
4.7.4 應用程式變數.....	59
4.7.5 共享全域變數.....	60
4.7.6 文件夾編輯器.....	61
4.7.7 塊程式設計.....	62
4.7.7.1 使用程序列表變數與運算式.....	62
4.7.7.2 移動模塊.....	63
4.7.7.3 路徑區塊.....	64
4.7.7.4 拾取區塊.....	64
4.7.7.5 放置區塊.....	64
4.7.7.6 容器模塊.....	64
4.7.7.7 下一個網格位姿模塊.....	64
4.7.7.8 重置網格位姿模塊.....	65
4.7.7.9 設置變數方塊.....	65

4.7.7.10 設置輸出區塊.....	65
4.7.7.11 等待區塊.....	65
4.7.7.12 循環區塊.....	65
4.7.7.13 退出, 當模塊.....	66
4.7.7.14 條件模塊.....	66
4.7.7.15 注釋區塊.....	66
4.7.7.16 退出模塊.....	66
4.7.7.17 對話框模塊.....	66
4.7.7.18 打印區塊.....	66
4.7.7.19 日誌區塊.....	66
4.7.7.20 通知區塊.....	66
4.7.7.21 警告區塊.....	66
4.7.7.22 錯誤區塊.....	66
4.7.7.23 Python 代碼區塊.....	67
4.8 I/O 菜單.....	67
4.9 可視化菜單.....	68
4.10 使能控制.....	69
4.11 系統監視器.....	70
4.11.1 機械臂狀態.....	70
4.11.2 程序輸出.....	70
4.11.3 程序日誌.....	70
4.11.4 歷史訊息.....	70
4.12 幫助菜單.....	71
4.12.1 機器人信息.....	71
4.12.2 快捷鍵.....	71
4.12.3 代碼文檔.....	71
4.12.4 開源許可證.....	71
4.12.5 聯繫方式.....	71
4.13 設置菜單.....	72
4.13.1 介面設置.....	73
4.13.2 自定義導航面板按鈕.....	73
4.13.3 啟動設置.....	73
4.13.4 運動行為.....	73
4.13.4.1 中斷狀態行為.....	74
4.13.4.2 中斷對話框.....	74
4.14 安全設置菜單.....	77
4.14.1 使用者介面.....	77
4.14.2 安全設置.....	79
4.14.2.1 安全限位 (SLP).....	79
4.14.2.2 安全極限關節角度.....	80
4.14.2.3 安全限速 (SLS).....	81
4.14.2.4 安全功率與力限制.....	82
4.14.2.5 安全 I/O 與操作人員限制.....	83

4.14.2.6 驗證資料.....	84
4.14.2.7 安全輸出功能.....	84
4.14.2.8 連接選配裝置.....	84
5. 面板介面 (GUI).....	85
5.1 簡介.....	85
5.2 配置面板介面的程式.....	87
5.3 控制台與設置.....	89
5.4 程式控制.....	90
6. 運動學定義.....	91
6.1 坐標.....	92
6.1.1 航海角.....	93
6.1.2 坐標系.....	93
6.2 配置.....	95
6.3 位姿.....	95
6.4 點.....	95
6.5 路徑.....	95
6.6 路徑段.....	96
6.7 網格.....	96
7. 連接模塊.....	97
7.1 連接、驗證與活動控制.....	98
7.1.1 連接通道.....	98
7.1.2 驗證.....	99
7.1.3 活動控制站.....	99
7.2 TCP連接模塊.....	99
7.2.1 建立連接.....	99
7.2.2 發送指令.....	101
7.2.3 設置.....	102
7.2.3.1 編碼標準.....	102
7.2.3.2 結束字元.....	102
7.2.3.3 發送心跳.....	102
7.2.4 訊息協議 - 接收指令.....	103
7.2.4.1 狀態資訊.....	103
7.2.4.2 執行回傳值.....	104
7.2.4.3 錯誤.....	104
7.2.4.4 歷史訊息.....	104
7.2.4.5 狀態群組定義.....	104
7.2.4.6 狀態群組資料.....	105
7.2.4.7 開啟對話框.....	106
7.2.4.8 腳本發送.....	106
7.2.5 訊息協議 – 發送指令.....	107
7.2.5.1 憑證登陸.....	107
7.2.5.2 請求活動控制.....	107
7.2.5.3 釋放活動控制.....	108

7.2.5.4 請求數據.....	108
7.2.5.5 上電.....	108
7.2.5.6 休眠.....	109
7.2.5.7 暫停.....	109
7.2.5.8 繼續運行.....	109
7.2.5.9 使能手動引導控制 (HGC).....	110
7.2.5.10 保持使能控制 (HGC).....	110
7.2.5.11 停止.....	111
7.2.5.12 變更項目.....	111
7.2.5.13 執行腳本.....	112
7.2.5.14 測試腳本.....	112
7.2.5.15 腳本接收.....	113
7.2.5.16 運動位姿.....	113
7.2.5.17 播放路徑.....	114
7.2.5.18 切換工具.....	114
7.2.5.19 寫入數字與類比輸出.....	115
7.2.5.20 對話框回應.....	116
7.2.5.21 設置狀態變數.....	116
7.2.5.22 設置全域變數.....	117
7.2.5.23 取得全域變數.....	117
7.2.5.24 設置共用變數.....	118
7.2.5.25 取得共用變數.....	118
7.3 REST連接模塊.....	119
7.3.1 建立連接.....	119
7.3.2 發送指令.....	120
7.3.3 設置.....	120
7.4 ROS 處理器模塊.....	121
7.4.1 安裝.....	121
7.4.2 配置.....	122
7.4.3 使用 Cyclone DDS 進行網路探索.....	122
7.4.3.1 ROS 網路接口.....	122
7.4.3.2 探索方法.....	122
7.4.4 建立連接.....	125
7.4.5 透過 ROS 存取機器人數據.....	125
7.4.6 腳本中的 ROS 2 功能.....	126
7.4.7 執行動作.....	126
7.4.8 驗證設置.....	127
7.5 Modbus連接模塊.....	127
7.5.1 概述.....	127
7.5.2 設置.....	127
7.5.3 寫入暫存器(簡單模式描述).....	128
7.5.3.1 命令暫存器.....	128
7.5.3.2 屬性暫存器.....	129

7.5.3.3 使用者寫入暫存器.....	129
7.5.4 寫入暫存器(詳細描述).....	130
7.5.4.1 不執行任何操作.....	130
7.5.4.2 啟動.....	130
7.5.4.3 休眠.....	130
7.5.4.4 停止.....	130
7.5.4.5 暫停.....	130
7.5.4.6 繼續運行.....	131
7.5.4.7 校準.....	131
7.5.4.8 釋放使能控制 (HGC).....	131
7.5.4.9 保持使能控制 (HGC).....	131
7.5.4.10 變更項目.....	131
7.5.4.11 開始腳本.....	132
7.5.4.12 移動到位姿.....	132
7.5.4.13 執行路徑.....	132
7.5.4.14 工具放置.....	132
7.5.4.15 工具拾取.....	132
7.5.4.16 寫入數字輸出.....	133
7.5.4.17 設置腳本事件.....	133
7.5.4.18 確認錯誤.....	133
7.5.4.19 讀取位置.....	133
7.5.4.20 請求活動控制.....	133
7.5.4.21 釋放活動控制.....	134
7.5.4.22 切換到手動模式.....	134
7.5.4.23 切換到自動模式.....	134
7.5.4.24 安全重置.....	134
7.5.4.25 開始運動中斷.....	134
7.5.5 讀取暫存器.....	135
7.5.5.1 狀態暫存器.....	135
7.5.5.2 使用者讀取暫存器.....	138
7.5.5.3 位置暫存器(第一關節至第六關節).....	138
7.5.5.4 座標暫存器 (x, y, z, X軸旋轉, Y軸旋轉, Z軸旋轉).....	139
7.5.6 錯誤暫存器映射.....	139
7.5.6.1 範例.....	140
7.5.7 Python 代碼範例.....	140
7.5.7.1 範例 1.....	140
7.5.7.2 範例 2.....	140
7.5.7.3 範例 3.....	141
7.5.7.4 範例 4.....	141
8. Cognex攝影機模塊.....	143
8.1 概述.....	144
8.2 攝影機設置.....	144
8.3 攝影機程式.....	144

商標

Cognex® 和 In-Sight Explorer® 是 Cognex Corporation 在美國及其他國家的商標或註冊商標。

Google Chrome™ 和 Chrome™ 是 Google LLC 的商標。

Linux® 是 Linus Torvalds 在美國及其他國家的註冊商標。

Apple®, macOS® 和 Safari® 是 Apple Inc. 在美國及其他國家和地區的註冊商標。

Microsoft®, Windows®, Microsoft Edge® 及 Microsoft Visual C++® 是 Microsoft Corporation 及其附屬公司在美國及其他國家的商標 (或註冊商標)。

Mozilla 和 Firefox® 是 Mozilla Foundation 在美國及其他國家的商標。

一般聲明：本文中使用的其他產品名稱僅供識別之用，可能為其各自所有者的商標。Epson 不主張對這些商標擁有任何權利。

所有螢幕截圖僅供說明與培訓用途，並無意亦不暗示與任何商標持有人存在關聯或獲得其認可。

使用條款

未經明確書面許可，不得以任何形式複製或翻印本說明手冊的任何部分。

本文件中的資訊如有變更，恕不另行通知。

若您發現本文件中有任何錯誤，或對其中的資訊有任何疑問，請與我們聯絡。

1. 簡介

1.1 程式設計與控制

軟體框架 AX Portal 為 Epson AX 系列機器人的使用者提供了一種非常直觀的方式來與該機器人進行通訊。透過 AX Portal，使用者不僅能完全操控機器人，還能使用 **Python 3.13** 程式語言編寫應用程式。有關使用 Python 進行程式設計的基本功能，可於網路上查閱。此外，內建的程式文檔中也提供了一些非常基礎的程式碼範例。

除了種類繁多的基本 Python 函式外，AX Portal 程式還支援大量專為控制、維護及與 AX 機器人通訊而設計的函式。AX Portal 所支援的完整指令碼函式列表，請參閱內建的程式碼說明文件。

1.2 圖形化使用者介面 (GUI)

此 GUI 設計為網頁應用程式，可在幾乎任何裝置的常見網頁瀏覽器上運行。官方支援以下瀏覽器：

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

因此，只要機器人與裝置位於同一網路或共享直接通信通道，便無需為您的控制裝置下載特定軟體模塊。除了主要針對桌上型電腦和筆記本電腦程式設計所優化的 AX Portal 默認桌面介面外，AX Portal 還提供面板介面，該介面僅包含操作已預先程式設計機器人所需的必要功能。您將在本文件後續內容中找到更多關於圖形化使用者介面及其所有支援功能的信息。

1.3 軟體模塊

除了本文件中詳細說明的 AX Portal 基礎功能外，AX 機器人還提供一些具備額外控制功能的內建軟體模塊。請注意，以下提及的所有連接模塊皆提供遠程連接至 AX 機器人功能，因此可實現對機器人的控制與監控。

1.3.1 TCP連線模組

TCP連線模組可用作與機器人通信的替代方式。使用者通常透過網頁瀏覽器中的 GUI 來控制機器人。然而，若需透過自訂介面或另一台機器來控制機器人，機器人可直接接受 TCP 指令。透過 TCP連線模組，所有可透過網站實現的功能，皆可透過 TCP 實現。

1.3.2 REST連線模組

REST連線模組可用作與機器人通信的替代方式。機器人通常由使用者透過網頁瀏覽器中的圖形化使用者介面 (GUI) 進行控制。然而，若機器人需由自訂介面或另一台機器控制，它將

直接接受 REST 指令。透過 REST 連線模組，所有可透過瀏覽器實現的功能，亦可透過 REST 達成。

1.3.3 ROS連線模組

ROS連線模組提供將機器人整合至 ROS2 環境的功能。機器人允許 ROS2 網路中的其他客戶端發送指令，用於控制和監控機器人。

1.3.4 Modbus連線模組

Modbus連線模組可作為與機器人通信的替代方案。它採用 Modbus TCP 協議，並允許使用者讀寫標準 Modbus 暫存器。可透過此模塊直接透過 PLC 控制機器人。諸如為機器人供電、執行程式及觸發輸出等主要功能，均可透過 Modbus 模塊實現。

1.3.5 面板介面模塊

面板介面模組提供了一個功能有限的圖形化使用者介面，主要供機器人操作人員而非程式設計師使用。此介面主要用於啟動和監控程式，其內容可由程式設計師進行修改，以符合您的使用情境。面板介面支援觸控螢幕，其設計類似於先進行動裝置中廣為人知的應用程式結構，即使對沒有技術背景的使用者而言，操作起來也直觀易懂。

1.3.6 Cognex攝影機模塊

Cognex相機模塊提供Cognex相機與 AX Portal 之間的通信接口。相機通常會整合至末端執行器中，或以固定方式安裝，用以觀察機器人運作的場景。此模塊非常適合用於可自訂的視覺檢測，並支持Cognex相機的功能範圍。

2. 管理員面板 (GUI)

2.1 簡介

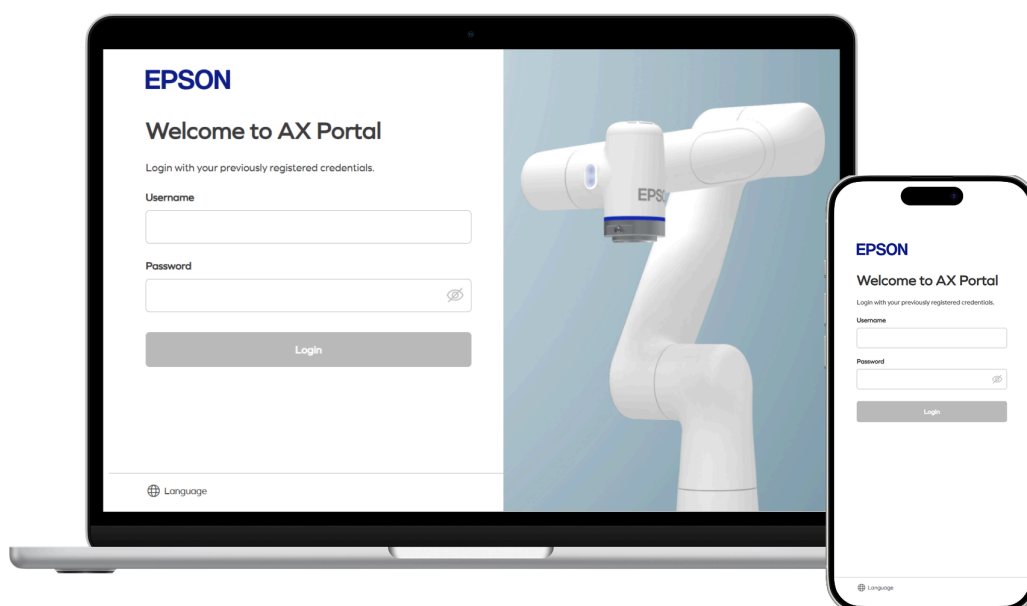


圖 2.1.a: 管理員面板範例截圖

AX Portal 的管理員面板是一個圖形化使用者介面 (GUI)，允許使用者管理使用者帳戶，並更改與機器人所運行作業系統相關的前期設置 (例如網路設置、時間設置等)。這些設置通常在機器人安裝時僅需設置一次，甚至在程式設計師開始編寫機器人控制程式之前便已完成。使用者可透過網頁瀏覽器 (例如 Google Chrome) 存取此介面，方法是在瀏覽器的網址列中輸入機器人的 IP 地址，系統將自動導向至 <https://<IP-ADDRESS>/admin> (正確的 IP 地址請參閱《機器人控制器 RC-A101 手冊》)。

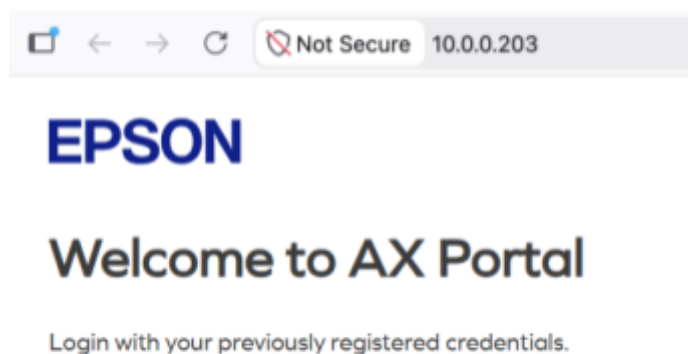


圖 2.1.b: 透過網址連接到管理員面板

本章將詳細說明管理員面板中的所有選單與元素。

2.1.1 介面語言

若要變更介面語言，請點擊介面左下角的「語言」下拉選單，並選擇您偏好的語言。您的瀏覽器會記住您在不同會話和介面中的語言設置，讓您的操作更加簡便。

2.1.2 儀表板

登入後，管理員面板的儀表板可讓您導航至各子選單、切換至 AX Portal 的其他介面，以及登出。請注意，儀表板僅會顯示您目前可存取的選項，具體取決於您的使用者等級（管理員、安全主管、程式設計師、操作人員）。若您無法導航至其他介面，則表示 AX Portal 目前未正常運行，例如在軟體更新中即屬此情況。

2.2 使用者管理

2.2.1 使用者等級

為防止未經授權的使用者對機器人進行配置、程式設計或操作，我們引入了不同的認證層級。所有使用者的使用者等級均由管理員定義。可用的使用者等級如下，依認證層級由高至低排序：

- **管理員**擁有所有機器人控制介面與功能的完整存取權限，可以管理其他使用者，並能存取作業系統中最重要之系統設置。
- **程式設計師**可存取所有機器人控制介面與功能，因此能為操作人員預先準備並編寫日後執行的程式。在管理員面板中，程式設計師基本上僅能存取自身帳戶詳情，以及在各介面間進行導航。
- **安全主管**是具備程式設計師權限（參見上文）且獲准變更機器人安全設置的使用者。其他使用者僅能檢視安全設置。
- **操作人員**僅能存取基礎操作介面，這使他們能夠操作機器人，但無法更改任何設置或程式。在管理員面板中，操作人員基本上僅能存取自己的帳戶詳情。

2.2.2 登陸頁面

登陸頁面包含兩種不同的表單，取決於是否已創建管理員帳戶。

- 若已創建管理員帳戶，則會顯示標準登陸頁面，您需輸入使用者名與密碼進行登陸。輸入密碼時，系統會自動隱藏密碼內容，除非您點擊密碼欄位內的「顯示密碼」按鈕。
- 若尚未建立管理員帳戶，系統將直接要求您建立，且必須先建立管理員帳戶才能繼續操作。詳細信息請參閱《六軸協作機器人 AX6: 快速入門指南 / 安裝手冊》。此處僅能設置絕對必要的數據；若要查看並設置帳戶的所有設置，請前往「帳戶管理」菜單。

每次登出後，系統將自動將您導向登陸頁面。

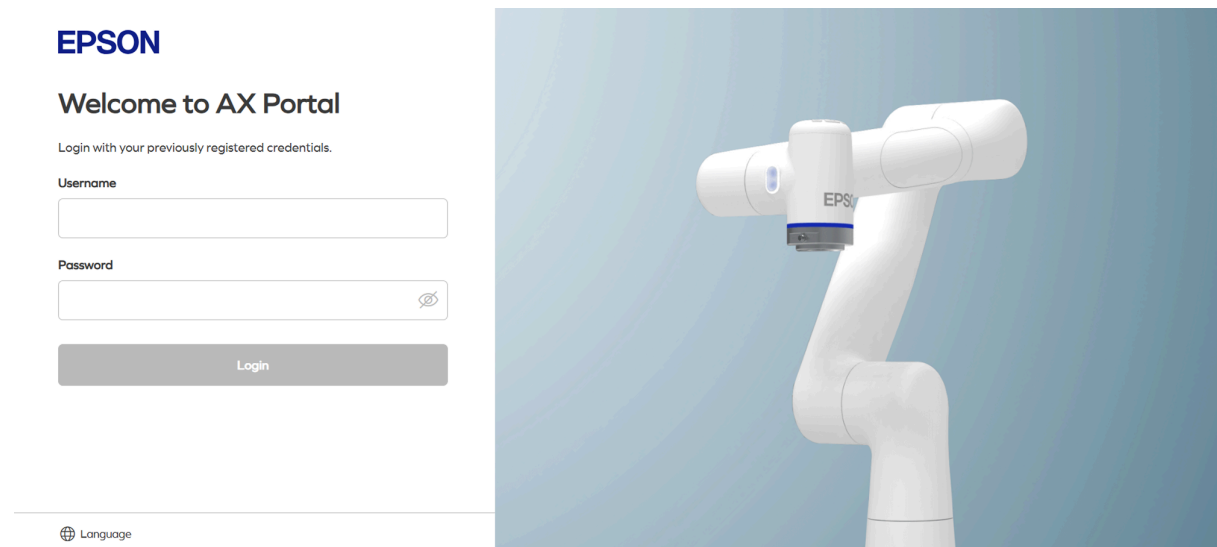


圖 2.2.2：登陸頁面

2.2.3 帳戶管理

在「帳戶管理」菜單中，您可以修改個人資料，具體包括姓名、密碼、檔案照片以及初始介面：

- 「全名」是介面及訊息中用於稱呼您的姓名。此為必填欄位，原意為填寫您的姓名與姓氏，但您亦可自行定義命名規則。
- 「用戶名稱」欄位在創建使用者後無法變更，該名稱由管理員在創建使用者時設置。
- 「初始介面」欄位用於指定登入後要轉至哪個介面。不同使用者等級預設有不同的初始介面，您可在此進行變更。
- 「更改密碼」按鈕可讓您更改帳戶密碼。系統會要求您重新輸入當前密碼，並輸入新密碼兩次，以儘量減少設置密碼時的意外錯誤。若您忘記密碼，任何管理員亦可為您更改。密碼從不以明文儲存，因此無論您或任何其他使用者都無法擷取您的密碼。
- 更換「檔案照片」屬可選操作。若未定義檔案照片，系統將顯示默認使用者圖標。

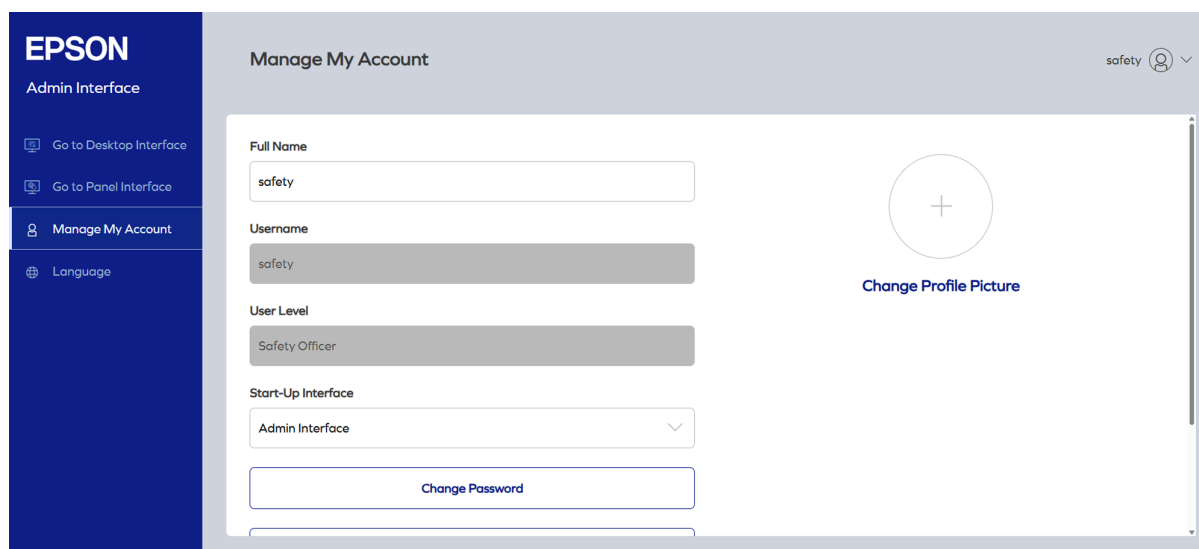


圖 2.2.3：帳戶管理頁面

2.2.4 管理其他使用者

在「管理其他使用者」菜單中，您將看到所有目前可用使用者的概覽。此處提供以下選項：

- 新增使用者可讓您建立具有預設使用者等級及預設初始密碼的使用者。任何使用者在首次登陸後，都應立即變更初始密碼。
- 修改使用者可變更該使用者的多數屬性，唯使用者名稱除外，此欄位完全無法變更。
- 刪除使用者會將其永久從數據庫中刪除。此使用者將無法再觸發機器人控制功能。

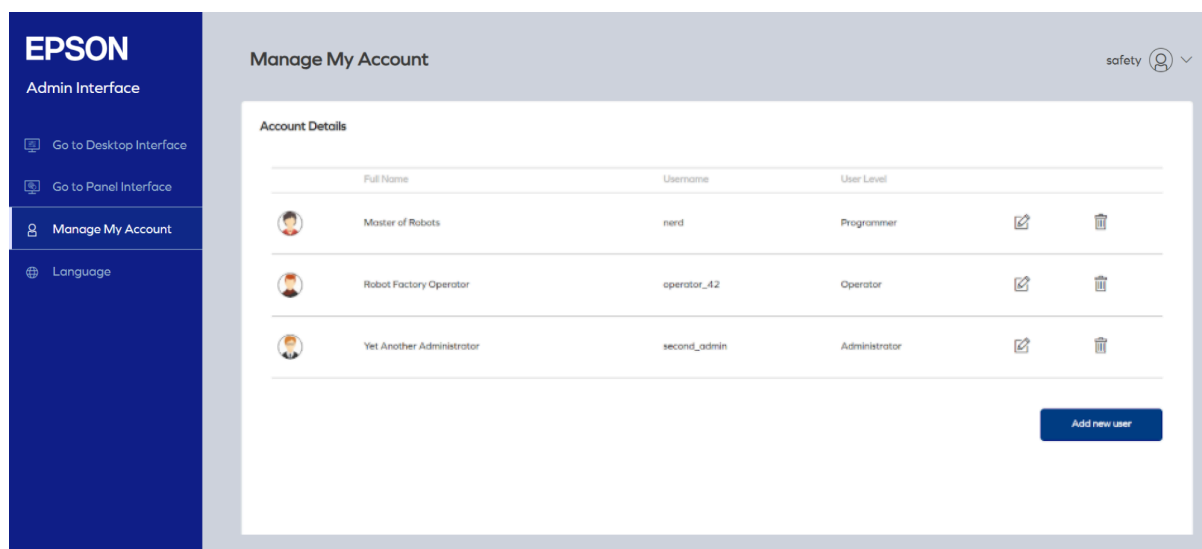


圖 2.2.4：管理其他使用者頁面

2.3 系統設置

2.3.1 網絡設置

「網絡設置」菜單包含所有與定義機器人乙太網路連接相關的設置。請注意，一旦您更改目前有效連接的設置（例如：您是透過乙太網路連接，但更改了乙太網路IP地址），您將失去連接。若您不小心輸入無效設置，之後可能無法再連接到機器人。因此強烈建議您在更改唯一可用的連接設置前，先確認另有兩條不同的連接均可正常運作。

「乙太網路設定」類別會顯示並讓您決定機器人應使用動態 IP 地址或手動 IP 地址。若機器人配置為動態獲取 IP 地址，您亦可在此處看到已選擇的 IP 地址。若機器人配置為手動設置，您需輸入 IP 地址、子網掩碼（必填），並可額外設置網關及最多兩個 DNS 伺服器（選填）。

The screenshot displays the 'Network Settings' page within the EPSON Admin Interface. On the left is a dark blue sidebar with the 'EPSON Admin Interface' logo and a list of navigation links: 'Go to Desktop Interface', 'Go to Panel Interface', 'Manage My Account', 'Manage Other Accounts', 'Network Settings' (highlighted), 'Time Settings', 'SSL Certificate', 'Software Update', 'Help & Support', and 'Language'. The main content area is titled 'Network Settings' and features a 'User' dropdown in the top right. The 'Ethernet IPv4' section contains the following settings: 'IP Address' is set to '192.168.90.200'; 'Dynamic IP Address' is selected with a blue radio button; 'Manual IP Address' is unselected with a white radio button; 'IP Address' (for manual) is '192.168.90.200'; 'Netmask' is '255.255.254.0'; 'Gateway' is '192.168.90.1'; 'DNS Server 1' and 'DNS Server 2' are empty input fields. Below this is a 'Network Information' section with a warning: 'You may use the second Ethernet port to access AX Portal for network maintenance reasons. Its static IP address is 10.0.0.203 and cannot be changed by the administrator. You may connect to "<Robot-Serial>.local" instead of entering the IP address to connect to AX Portal via your browser.' At the bottom are 'Cancel' and 'Save' buttons.

圖 2.3.1：網絡設置頁面

2.3.2 時間設置

「時間設置」菜單顯示機器人作業系統當前日期、時間和時區。請務必輸入正確日期和時間。設置正確的時間後，系統才能用於日誌分析和故障排除，且依賴時間的功能才能正常運行。如果手動更改時間，按下保存按鈕後，新時間將立即生效。若要變更時區，可透過大陸（例如「歐洲」）、國家代碼（例如「GB」）、主要城市（例如「倫敦」），或這些條件的組合（例如「eur lon」→ 歐洲/倫敦）來篩選系統中可用的時區列表。

注意：若機器人具備網路連線能力，您可啟用 NTP（網路時間協議）同步功能。

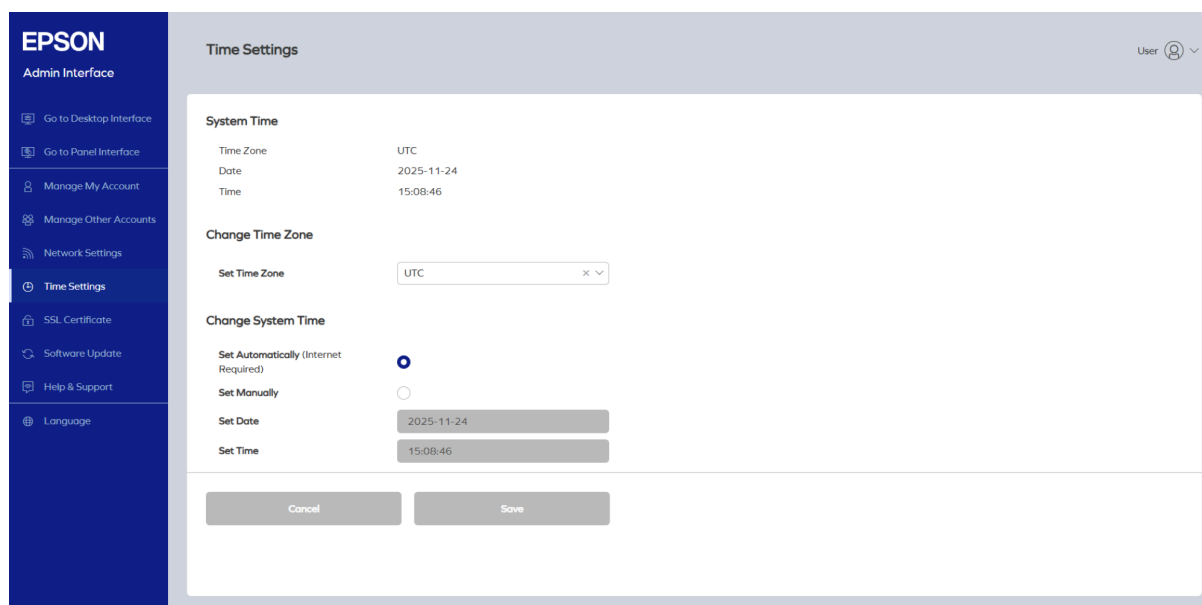


圖 2.3.2：時間設置頁面

2.3.3 SSL 憑證

「SSL 憑證」菜單會顯示目前匯入的 SSL 憑證相關詳細信息，並允許您將此證書替換為您自己的私人或企業證書。SSL 憑證是一種數字證書，用於驗證網站的身份並啟用加密連接。此證書可確保您在網頁瀏覽器中的線上交易和客戶信息保持私密與安全。默認情況下，系統會安裝自簽名證書，建議將其替換為適合使用者環境且由受信任的證書機構簽署的證書。

若發生以下情況，瀏覽器將標記該憑證為不安全：

- 證書中註冊的網域名稱/IP 位址與存取機器人所使用的網域/IP 不符，
- 憑證已過期，或
- 該憑證為自簽名憑證。

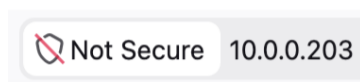


圖 2.3.3.a: 瀏覽器提示 SSL 憑證不安全或缺失的範例

若要更換 SSL 證書，您需要上傳私密金鑰與公開金鑰這對檔案。

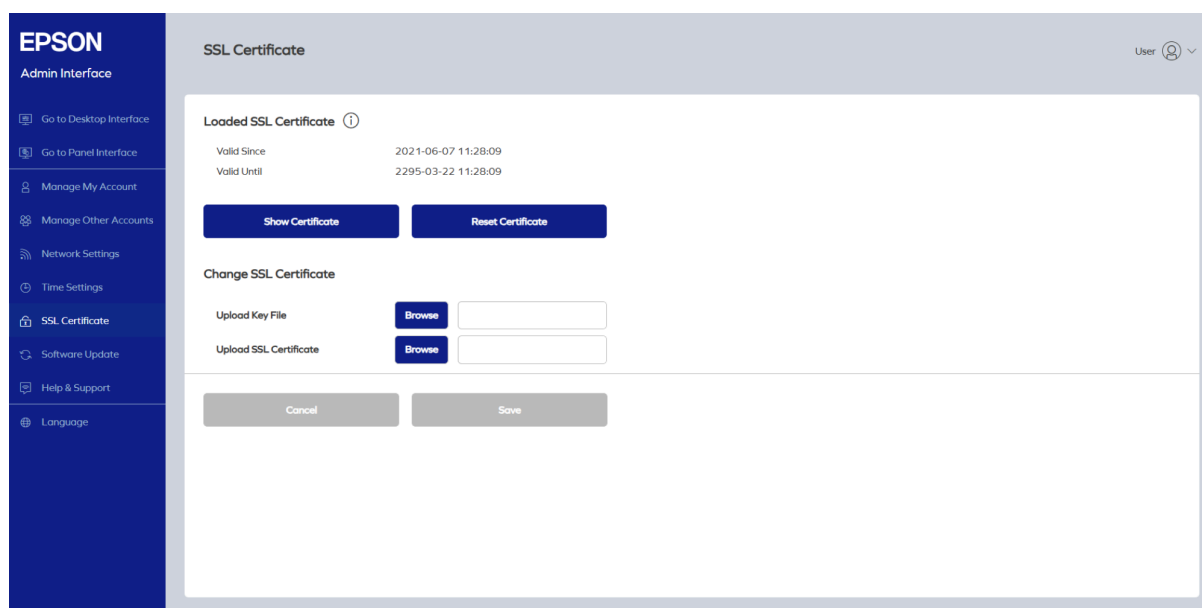


圖 2.3.3.b: SSL 憑證設置頁面

2.4 軟體更新

在「軟體更新」菜單中，您可查看 AX Portal 的當前運行版本。您可在此上傳及安裝新版軟體。進行軟體更新可能是為了修復特定問題，或是為了取得最新功能與錯誤修正。

無論如何，在更新軟體版本之前，請務必先詳閱發行說明，因為其中包含有關更新至特定版本潛在風險的重要資訊。功能改進有時會以淘汰舊版功能或特性為代價。我們建議您在觸發更新之前，先透過桌面備份資料庫，該資料庫包含您所有的機器人程式和設定。

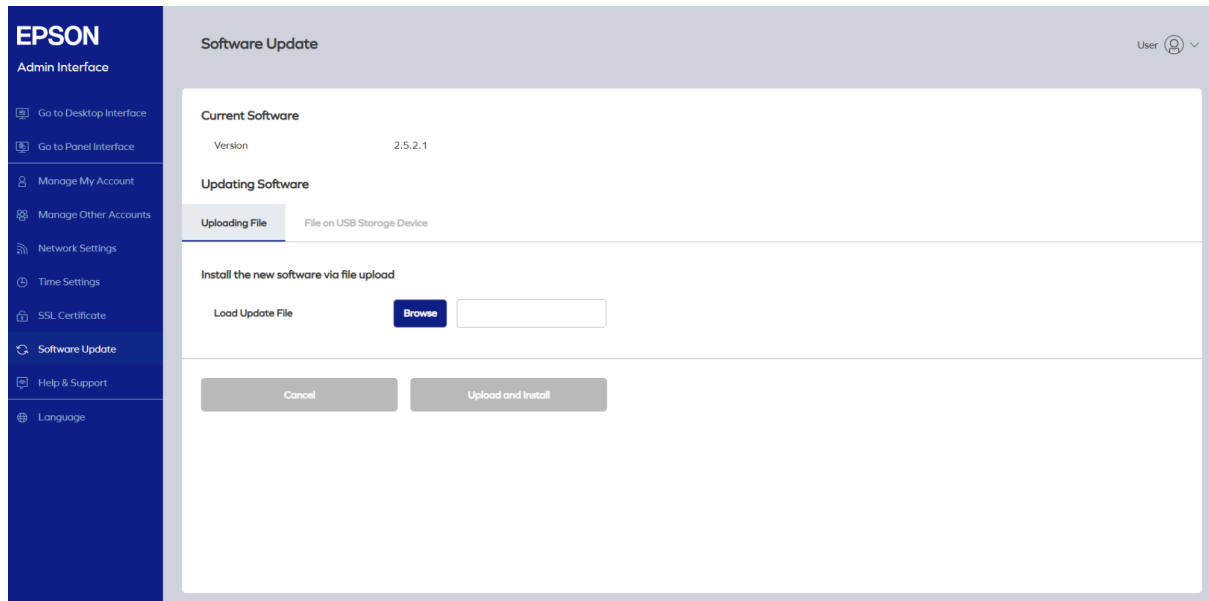


圖 2.4.a:軟體更新頁面

若要安裝軟體更新，您首先需要取得包含欲安裝之 AX Portal 版本更新的 RBIN 檔案。安裝此更新有三種方式：

- 1) 選擇第一張標籤頁「上傳文件」。在此您可以從裝置上傳更新檔。瀏覽檔案系統並選擇更新用的 RBIN 文件，接著繼續安裝更新。
- 2) 選擇第二個標籤頁「文件在 USB 存儲設備」。您可將 RBIN 文件存入 USB 隨身碟，並將其掛載至機器人控制器。從 USB 隨身碟目錄中選擇更新用的 RBIN 文件，接著繼續安裝更新。

注意：目前支援 exFAT 和 FAT32 檔案系統的 USB 隨身碟。

- 3) 若選項 1) 和 2) 均不適用（例如：管理員面板在您的機器人上無法正常運行），您隨時可改為執行「復原更新」。

在更新進行期間，您將看到當前更新的進度狀態、已成功完成的階段，以及若發生錯誤時，錯誤發生的具體階段與錯誤訊息。根據更新的套件不同，更新過程**最多可能需要 15 分鐘**。因此請耐心等待，切勿中斷更新程式（例如切斷機器人的電源）。最壞的情況下，此舉可能導致您之後無法再存取該機器人。

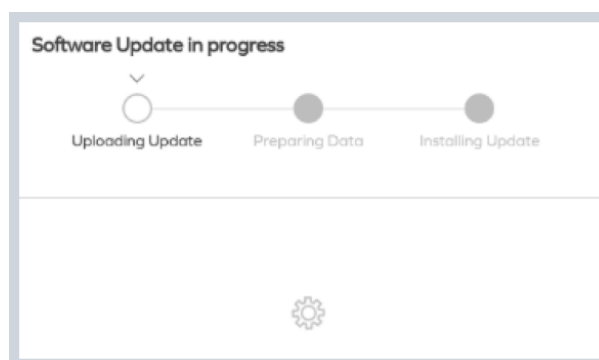


圖 2.4.b:更新進行中視窗

更新成功完成後，必須確認更新。在確認更新之前，系統會顯示紅色訊息欄以示提示。若未確認更新，重新啟動控制器將重置更新。一旦確認，便無法回滾至較早的版本。

2.5 說明與支援

「幫助與支援」菜單顯示您在需要幫助時可聯繫 Epson 的聯繫方式。請務必先閱讀文件說明，再透過電子郵件或電話尋求支持。



圖 2.5：幫助 & 支援頁面

2.6 安全驗證

若您使用連接模塊存取 AX Portal 來控制機器人（無 GUI），在實際控制機器人之前，您需要透過安全的兩步程序進行驗證。

2.6.1 匯出 SSL 憑證

第一步，您需要向內建的驗證伺服器發送登陸請求。若要下載目前加載的 SSL 憑證，可透過瀏覽器將證書匯出。下圖展示如何使用 Google Chrome 執行此操作。其他瀏覽器亦支援類似功能。

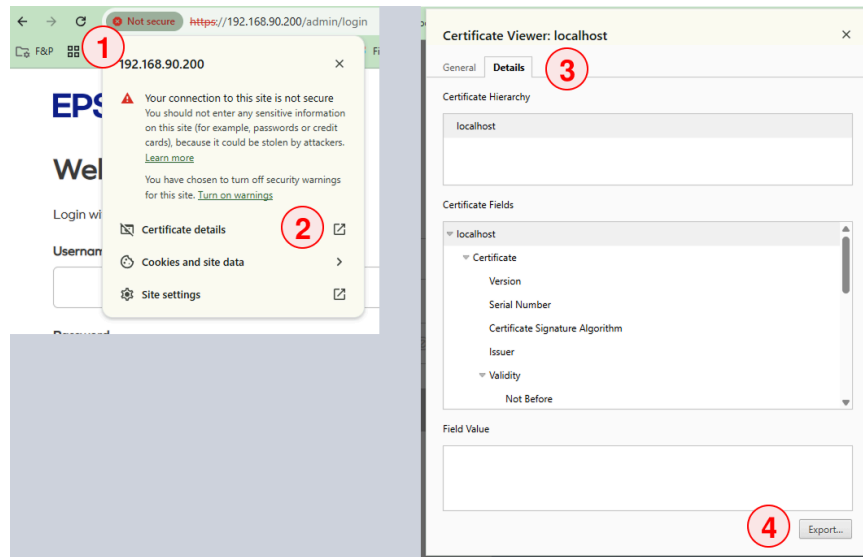


圖 2.6.1：匯出 SSL 憑證

1. 檢視網站信息, 2. 前往「憑證詳細資料」, 3. 前往「憑證檢視器中的詳細資料」, 4. 匯出憑證

2.6.2 使用者登陸

第一步，您需要向內建的認證伺服器發送登陸請求，該伺服器可透過 **TCP 5001** 埠訪問。為此，您需要知道機器人的 **IP** 地址、您的登陸憑證（**用戶名和密碼**），並且必須使用與機器人相同的 **SSL** 憑證。以下是一個透過 TCP 套接字發送登陸請求並返回存取憑證的 Python 範例。您亦可使用其他語言執行此操作。

注意：目前僅支援透過 TCP 進行登入。若您希望透過 TCP 以外的連接模塊（例如 ROS/REST/Modbus）控制機器人，仍需使用 TCP 取得存取憑證。此外，您亦可在 AX Portal 的設置中，針對這些連接模塊關閉驗證功能。請注意：若您禁用驗證機制，任何客戶端皆可透過 ROS/REST 控制機器人，無需事先驗證！

```
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 連接註冊

一旦取得**有效的存取憑證**，您即可在「token_login」函式中使用它，其用法在專門介紹連線模組的章節中有詳細說明。

3. USB 匯出與復原更新

3.1 概述

如果控制器正在運行中，但無法連接到管理員面板，您可以嘗試使用恢復更新方法來恢復控制器。

3.2 USB 匯出

無需存取網站介面，即可透過 USB 隨身碟匯出系統日誌和 AXPortal 資料庫：

1. 將 USB 隨身碟格式化為 exFAT、FAT32 或 NTFS，並確保其不帶有日誌或 RECOVERY 檔案系統標籤
2. 將 USB 隨身碟插入控制器，並等待數分鐘。
3. 拔除 USB 隨身碟。您應能在隨身碟的根目錄中找到一個 ZIP 檔案，其中包含資料庫及相關的系統日誌。

3.3 執行復原更新

要運行復原軟體更新，請依照以下步驟操作：

1. 取得**更新檔案**，該檔案通常為 RBIN 格式。若您沒有更新檔案，請聯繫供應商的聯繫方式。
2. 請確保您手邊有一支可用於更新的 **USB 隨身碟**。該隨身碟應格式化為 exFAT、FAT32 或 NTFS 格式，且第一個分區的檔案系統必須標記為「RECOVERY」。
3. 加載更新文件到 USB 隨身碟的根目錄中。
4. 休眠機器人並重新啟動。請等待**至少 1 分鐘**，讓機器人正常開機。
5. 將準備好的 USB 隨身碟插入控制器上的任一 USB 埠。
6. 等待更新完成。請注意，此過程可能需要**長達 15 分鐘**。嘗試存取管理員面板，並檢查新版本是否已安裝。
7. **重要：**請從控制盒中安全移除 USB 隨身碟。若遺漏此步驟，重啟後系統將再次執行更新，這並非您所期望的結果。

4. 桌面介面 (GUI)

4.1 簡介

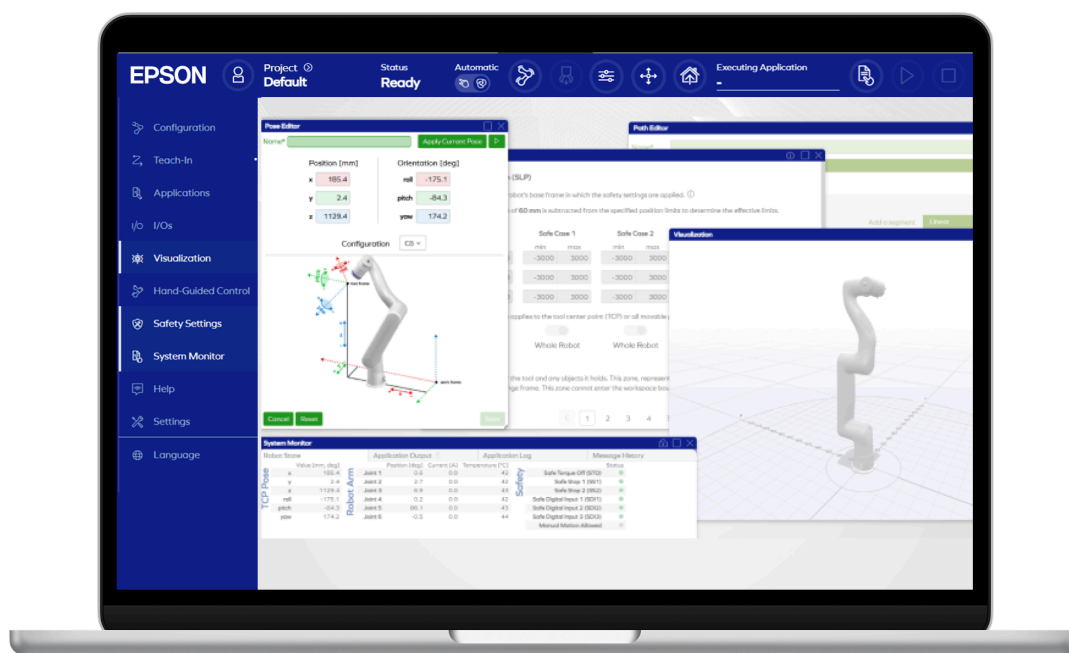


圖 4.1.a: 桌面範例截圖

AX Portal 的桌面介面是主要圖形化使用者介面 (GUI)，讓使用者能夠控制機器人、編寫程式、操作資料庫元素，並視覺化顯示當前機器人狀態。此 GUI 設計直觀且多數功能一目瞭然，即使對機器人控制經驗不多的使用者亦然。您可透過管理員面板登入，並點擊「導航到桌面」按鈕來存取此介面。

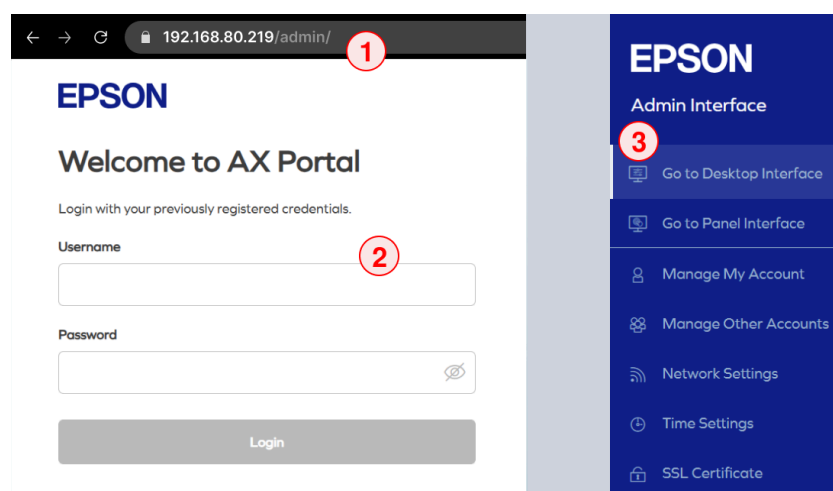


圖 4.1.b: 存取桌面

1. 在瀏覽器網址列輸入 IP 地址，2. 輸入憑證登入，3. 導航至桌面

桌面介面包含機器人所有可用的控制功能，以及用於觀察和修改其狀態的附加功能。若僅需執行預先程式設計的程式，且無需修改任何程式或設置，我們建議改用面板介面。本章將詳細說明桌面介面中的所有視窗、選單及元素。

在某些瀏覽器（特別是在 Windows 系統上），顯示縮放比例默認值可能設定得高（例如 150%），這可能會導致介面部分區域（例如駕駛艙或標誌）出現截斷現象。為獲得最佳使用體驗，建議將顯示縮放比例設定為 100%（或更低）。

4.2 主視窗

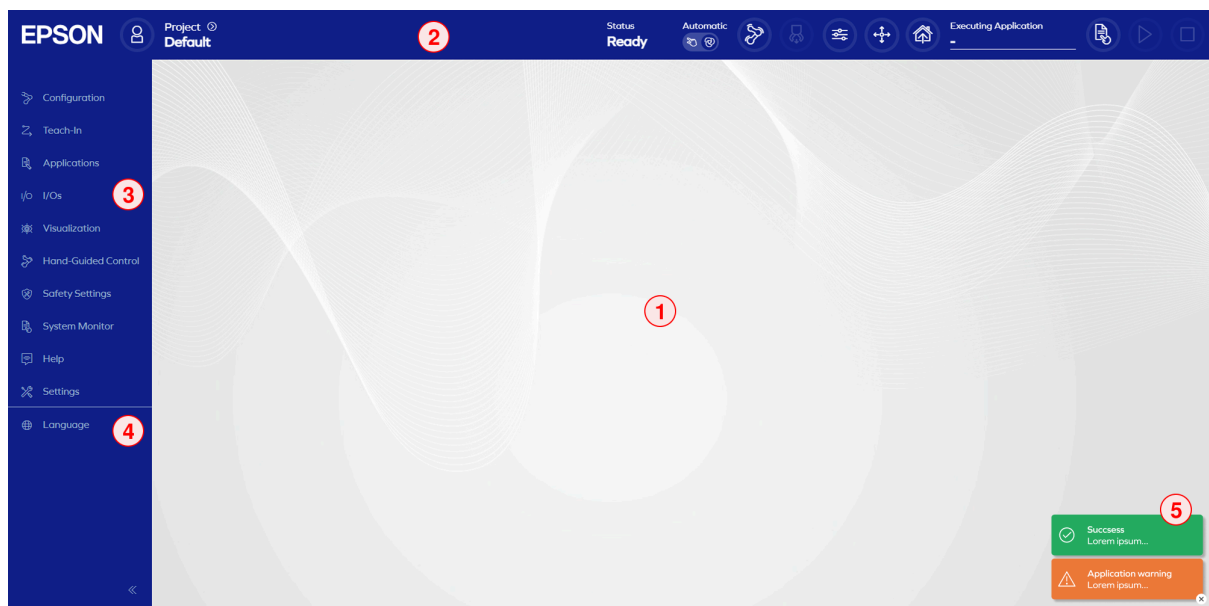


圖 4.2：桌面介面的元素

1. 工作區、2. 儀表板、3. 選單、4. 切換語言、5. 彈出訊息

當您成功將電腦或裝置連接到桌面後，主視窗便會出現在您的網頁瀏覽器中。

此主視窗包含以下項目：

- 位於視窗最上方的控制台，顯示當前程式狀態、訊息，並可快速存取機器人控制功能，
- 視窗左側的菜單按鈕，點擊後可導航到可用菜單（本章後文將詳細說明），
- 螢幕右側的訊息彈出視窗區域，以及
- 主要工作區，使用者可在其中以類似檔案總管的樣式重新排列選單。

4.2.1 選單概覽

在主視窗的左側，您會看到用以進入 AX Portal 各不同選單的按鈕。每個可用選單的具體用途與詳細信息將在後續章節中說明。以下是可用菜單的簡要概述：

	「 配置 」菜單讓使用者能配置並上電啟動機器人。
	「 示教 」菜單讓使用者能向機器人示教位姿、路徑和網格。
	「 應用程式 」菜單讓使用者能夠編寫、修改及測試自訂應用程式。
	「 I/O 」菜單讓使用者監控 I/O 狀態並變更輸出訊號。
	「 可視化模型 」菜單會顯示機器人的當前運動。
	「 使能控制 」菜單讓使用者能手動移動機器人的關節。
	「 安全設置 」菜單讓安全主管能夠修改安全設置。
	「 系統監視器 」菜單可讓使用者檢視系統輸出。
	「 幫助 」菜單包含有關程式和系統的有用信息。
	「 設置 」菜單讓使用者選擇一些基本設置。
	「 語言 」按鈕讓使用者切換介面的顯示語言。

4.2.2 控制台

AX Portal 的儀表板始終在 GUI 最上方顯示，並包含最重要的信息與控制項。讓我們從左至右詳細說明儀表板的所有元素：

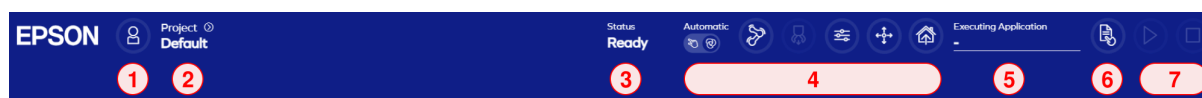


圖 4.2.2：控制台

1. 使用者帳戶選項、2. 項目選擇、3. 機器人狀態、4. 機器人控制、5. 運行中或已選擇的程式、6. 程式選擇、7. 主要控制項



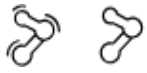
按下「**使用者**」按鈕可更改帳戶設定或登出。這些選項將引導您進入管理員面板。

按下「**項目**」按鈕可進入項目菜單，您可在其中切換項目並查看項目概覽。此按鈕下方會顯示當前活躍項目的名稱。

「**請求控制**」按鈕僅在該介面非機器人當前主控站時顯示。關於此安全功能的詳細說明，請參閱「活動控制」章節。

狀態訊息欄顯示當前的程式狀態。程式狀態基本上決定了指令或按鈕按壓等動作是否可執行。

「**操作模式**」滑塊用於在手動操作模式與自動操作模式之間切換。關於此安全功能的詳細說明，請參閱「操作模式」章節。



「**啟用/停用 HGC**」按鈕可讓使用者切換至手動引導控制模式。啟用時，狀態會切換為「HGC」，使用者即可手動引導機器人。機器人僅能在程式狀態為「就緒」時切換至 HGC 模式。禁用 HGC 將使狀態切換回「就緒」。



「**工具**」按鈕可在已安裝工具的「拾取」與「放置」功能之間切換。在工具編輯器中，使用者可將這些功能鏈接至數字輸入。僅當行為已定義且程式狀態為「就緒」或「HGC」時，方能觸發該工具。



「**精細控制**」按鈕會開啟精細控制控制面板。在此處，使用者可調整機器人的個別坐標或關節角度。



「**運動**」按鈕會開啟運動控制面板。在此，使用者可調整機器人速度、變更全域速度與加速度限制，以及切換坐標模式。



「**歸位**」按鈕可讓使用者將機器人移動到其回零位置。此功能旨在作為快捷方式，例如在運行程式前移動到初始位置，或在斷電中移動到休眠位置。回零位置可在「設置」菜單中進行變更。

「**目前選取**」區域顯示當前與控制按鈕（播放、暫停、繼續、停止）連結的元素（應用程式、路徑、位姿等）。此信息會在您點擊播放按鈕或選取應用程式時隨之變更。



「**應用程式選單**」讓使用者能選擇先前透過「程序編輯器」所創建的應用程式。



「**播放**」按鈕用於執行所選應用。程式狀態必須為「就緒」才能播放應用。「**暫停**」按鈕用於暫停正在運行的應用。若機器人在移動過程中暫停應用，機器人將平穩地停止。「**繼續運行**」按鈕用於恢復已暫停的應用。若應用是在移動過程中被暫停，該移動將平穩地繼續。「**停止/中止**」按鈕可停止運行中或暫停中的程式。若程式在移動過程中被停止，機器人將立即停止並保持當前位置。此按鈕會觸發自動恢復程式，將狀態切換回「就緒」。



4.2.3 活動控制 (單一控制點)

為確保機器人隨時僅能由單一控制站進行控制，介面透過「活動控制」功能受到保護。若您是首位登入機器人的使用者，您將自動獲得控制權。我們將此稱為「活動控制站」。若其他人（或您自己）隨後嘗試使用不同的連線通道（也可能是不同的瀏覽器視窗），該連線將無法取得控制權，僅能監控與觀察機器人（只讀存取）。我們將此稱為「被動觀察者」。

4.2.3.1 活動控制站

主動控制站可選擇釋放其控制權。在介面中，可透過點擊「**使用者**」按鍵並選擇「**釋放控制權限**」來執行此操作。若您習慣在有多名使用者或多個介面的機器人上工作，建議您始終釋放控制權限和/或關閉連接通道。

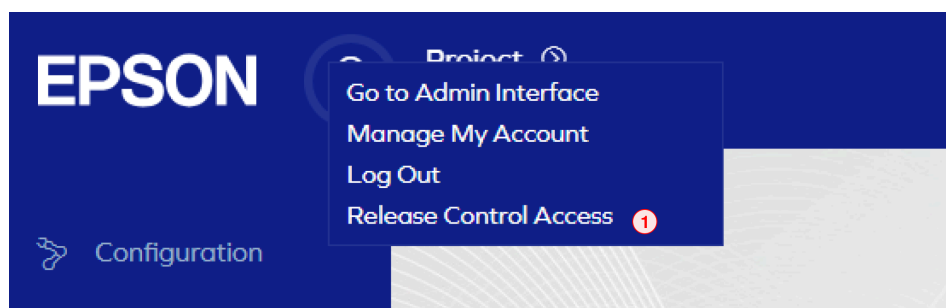


圖 4.2.3.1 釋放控制權限

1. 釋放控制權限

4.2.3.2 被動觀察者

被動觀察者可請求控制權。在介面中，可透過點擊「使用者」按鈕並選擇「申請控制權限」，或直接點擊駕駛艙中出現的「請求控制」大按鈕來執行此操作。請求控制權後，主動控制站與被動觀察者雙方皆會彈出對話框以確認存取權限。被動觀察者只能等待控制權限獲准，而主動控制站則可選擇接受或拒絕該請求。

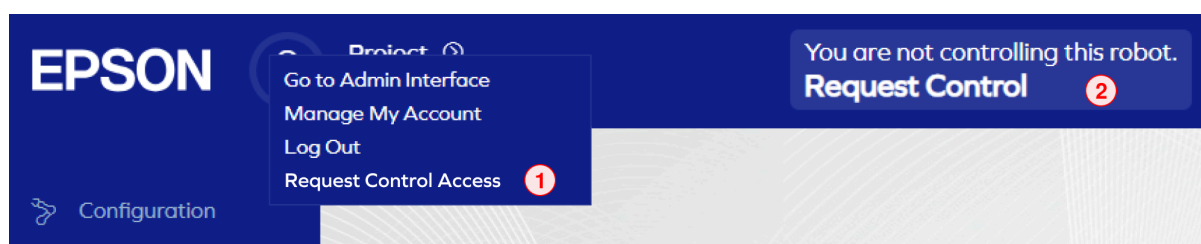


圖 4.2.3.2 申請控制權限

1. 申請控制權限, 2. 申請控制對話框

4.2.4 程式狀態

當前程式和/或機器人的狀態可在駕駛艙中查看。該狀態會提供關於 AX Portal 和/或機器人當前運作狀況的反饋，並基本上決定了當前可執行的動作或指令。每種狀態都有其獨特的屬性，且某些功能僅在特定的程式狀態下才可用。以下是各種狀態及其說明的列表：

未上電	馬達未通電且裝置未連接。此狀態用於在通電前配置機器人。使用上電滑塊會使狀態變更為「上電中」。其他操作可能會將狀態變更為「加載中」，特別是當某項操作不應被中斷且需要一些時間才能完成時。
加載中	機器人目前正在處理某些數據，例如加載資料庫或某些設置。資料加載完成後，狀態會返回“未上電”。
上電中	機器人目前正在上電中，啟動馬達並連接裝置。成功連接後，機器人將開始主動控制馬達，狀態會變更為「就緒」。
就緒	程式已就緒接收指令及/或執行操作，例如執行程式、拾取與放置物體等。此為機器人的閒置狀態。

運行中	目前正在運行的應用程式或動作。可分別透過對應的「暫停」或「停止」按鈕，將此應用程式或動作暫停或停止。當應用程式運行至程式腳本結尾，或動作運行完畢後，狀態將恢復為「就緒」。
暫停	目前應用程式/動作已暫停。按下「繼續運行」按鈕可恢復執行，狀態將恢復為「運行中」；或按下「停止」按鈕停止執行，狀態將恢復為「就緒」。
使能控制	手動引導控制 (HGC) 已針對機器人的單一或多個關節啟用。在此模式下，您可以透過對其關節施加力量，手動引導機器人至任何您想要的姿勢。同時，機器人會透過主動補償重力來維持自身重量。請注意，您需要按下 HGC 按鈕或啟動使能開關才能移動機器人。再次 保持手動引導控制 ，即可將狀態恢復為「就緒」。
處理中	目前正在執行耗時的動作，例如儲存/加載資料或計算路徑。待此操作成功處理完畢後，狀態將恢復為先前值。
停止中	通常，「停止中」狀態會立即自動恢復為「就緒」。若未發生此情況，表示 AX Portal 的某些部分未能正確完成處理。很遺憾，此時只能透過重啟機器人來恢復正常。
急停 正常停止 保護停止 碰撞 安全防護 位置限制	這些狀態對應於不同的中斷。若觸發中斷，系統將顯示對應的狀態，並開啟「中斷視窗」以處理該中斷。一旦中斷解決，狀態通常會變更為「繼續」。
繼續	中斷解決後，狀態會變更為「繼續」。此時需由使用者手動確認，確認繼續執行機器人應用程式或行為是安全的。
錯誤	發生無法自動解決的錯誤，或需要使用者特別注意的錯誤。請查看錯誤訊息，並嘗試在可能的情況下解決錯誤。
斷電中	機器人目前正在斷電中，關停馬達電源、斷開裝置連接，並結束所有運行中的模塊。此後狀態將變更為“未上電”。

4.2.5 操作模式

AX 系列機器人具備兩種操作模式：手動操作模式與自動操作模式。可透過駕駛艙內的專用滑塊切換操作模式。

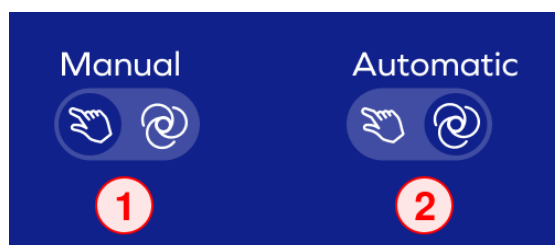


圖 4.2.5 操作模式切換按鈕

**警告**

僅當系統已配備必要的安全功能、經過測試，且其安全性已透過風險評估確認無虞時，方可使用自動操作模式。在其他情況下使用自動操作模式極為危險。如需進一步信息，請參閱《AX6 & RC-A101 安全手冊》。

4.2.5.1 自動操作模式

自動操作模式通常用於正常運作，即當程式已驗證可正常運行時。此模式允許機器人使用全速範圍。

4.2.5.2 手動操作模式

手動操作模式是一項安全功能，通常用於修改機器人程式時，特別是在教授新姿勢或測試程式變更時。在手動操作模式下，機器人以減速狀態移動（最大 250 mm/s），且僅在**使能按鈕**被觸發時才會移動。使能按鈕為三位置安全開關。若使能按鈕未被觸發，系統將透過警告訊息通知使用者，表明不允許進行移動。



圖 4.2.5.2 使能按鈕範例

4.2.6 精細控制

按下操作臺上的精細控制按鈕即可開啟「精細控制」視窗。透過精細控制功能，可以精確定位機器人，精度達到 0.1 [mm]/[deg] 。此視窗包含兩個分頁。第一個分頁允許使用者修改工具中心點 (TCP) 的當前歐幾裡得坐標，此坐標等同於以工件坐標系表示的當前工具坐標系。第二個分頁可讓使用者修改機器人當前的關節角度設定。

這些分頁具備以下功能：

輸入欄位	在輸入欄位中輸入數值可變更單個坐標值或關節值。
滑塊	滑塊同時定義「 增量步長 」（決定短按控制器按鍵時機器人運動的距離）以及「 最大運動速度 」（決定機器人使用這些控制器時運動的速度）。
控制按鈕	短按 控制按鈕會沿按鈕所示方向應用所選擇的增量。另一方面， 長按 控制按鈕則會沿按鈕所示方向進行連續運動。
配置	一個特定的目標坐標點（TCP）最多可透過八組關節角度組合來到達。這些關節角度組合稱為該位姿的配置。透過在下拉菜單中變更“配置”值，使用者可為當前位姿選取特定的配置。此外，按下“下一個配置”按鍵，使用者即可依次瀏覽當前位姿的所有可用配置。請注意，變更配置並非微調步驟，此操作可能會移動機器人的所有關節。
保存為姿勢	按下「 保存為姿勢 」按鈕，此位姿將以新的唯一名稱儲存。所有可用位姿皆列於「示教」菜單中。

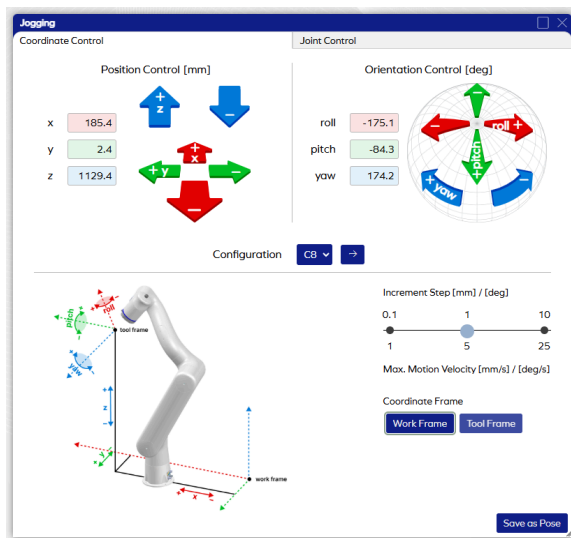


圖 4.2.6.a: 精細控制 – 坐標控制

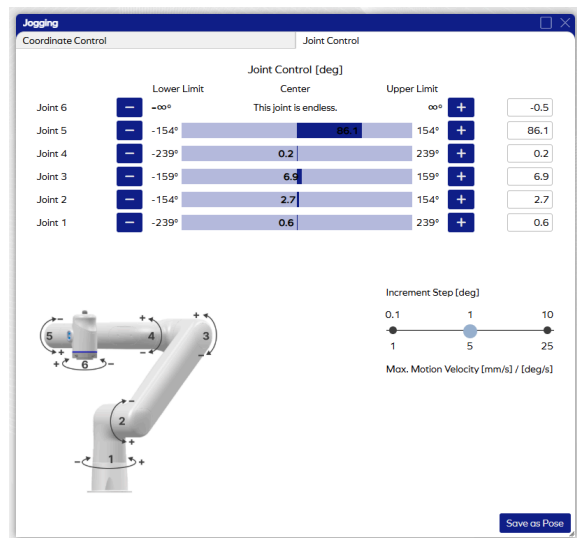


圖 4.2.6.b: 精細控制 – 關節控制

4.2.7 運動控制

按下駕駛艙內的運動控制按鈕，即可開啟「運動控制」視窗。此視窗包含以下三個分頁：

4.2.7.1 運動限制

「運動限制」分頁包含用於限制機器人最大速度，以及透過縮放係數降低所有機器人動作速度的設置。這些控制功能在設置機器人應用時非常有用，例如在驗證新應用或變更現有應用中的動作時，用以限制速度。

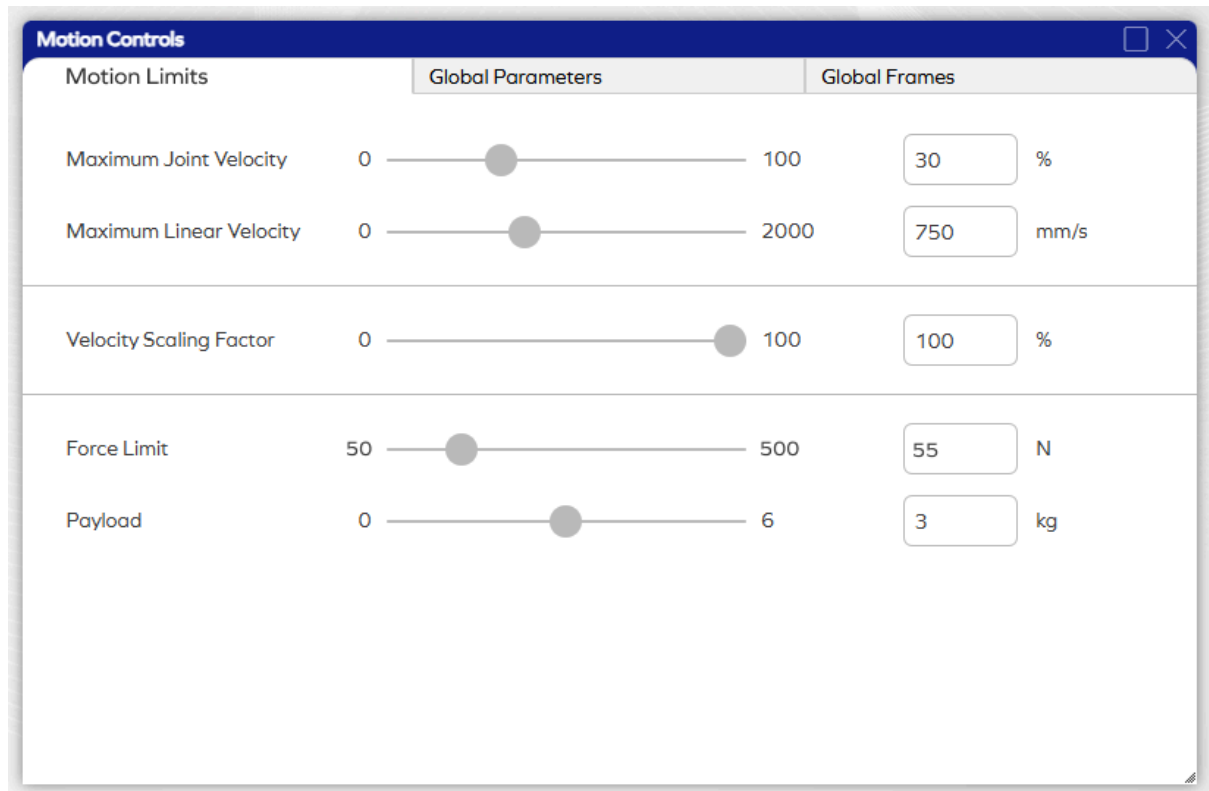


圖 4.2.7.1：運動控制 – 運動限制標籤頁

另一方面，力限制與負載則用於調整機器人的碰撞敏感度。力限制制定了作用於工具上的允許外部力上限，而負載則定義了機器人所能承載的總重量（包含其工具）。

4.2.7.2 全域參數

「全域參數」分頁包含用於定義各種移動指令默認速度與加速度的設置。移動指令分為以下類別：

- 用於以時間最優方式移動機器人的指令使用「joint_velocity」和「joint_acceleration」參數，其默認值在此設置為「**全域關節速度**」和「**全域關節加速度**」。
- 用於讓機器人朝特定方向或形狀（直線、圓弧運動）移動的指令，會使用「linear_velocity」和「linear_acceleration」參數，其默認值在此設置為「**全域線速度**」和「**全域線性加速度**」。
- 用於使工具中心點 (TCP) 保持在同一位置、僅改變其方向的機器人運動指令，使用「angular_velocity」和「angular_acceleration」參數，其默認值在此設置為「**全域角速度**」和「**全域角加速度**」。

- 用於沿包含多個運動區段的路徑移動的指令，會使用「blend_radius」參數，其默認值在此設定為「全域混合半徑」。此參數用於平滑路徑區段之間的插值。

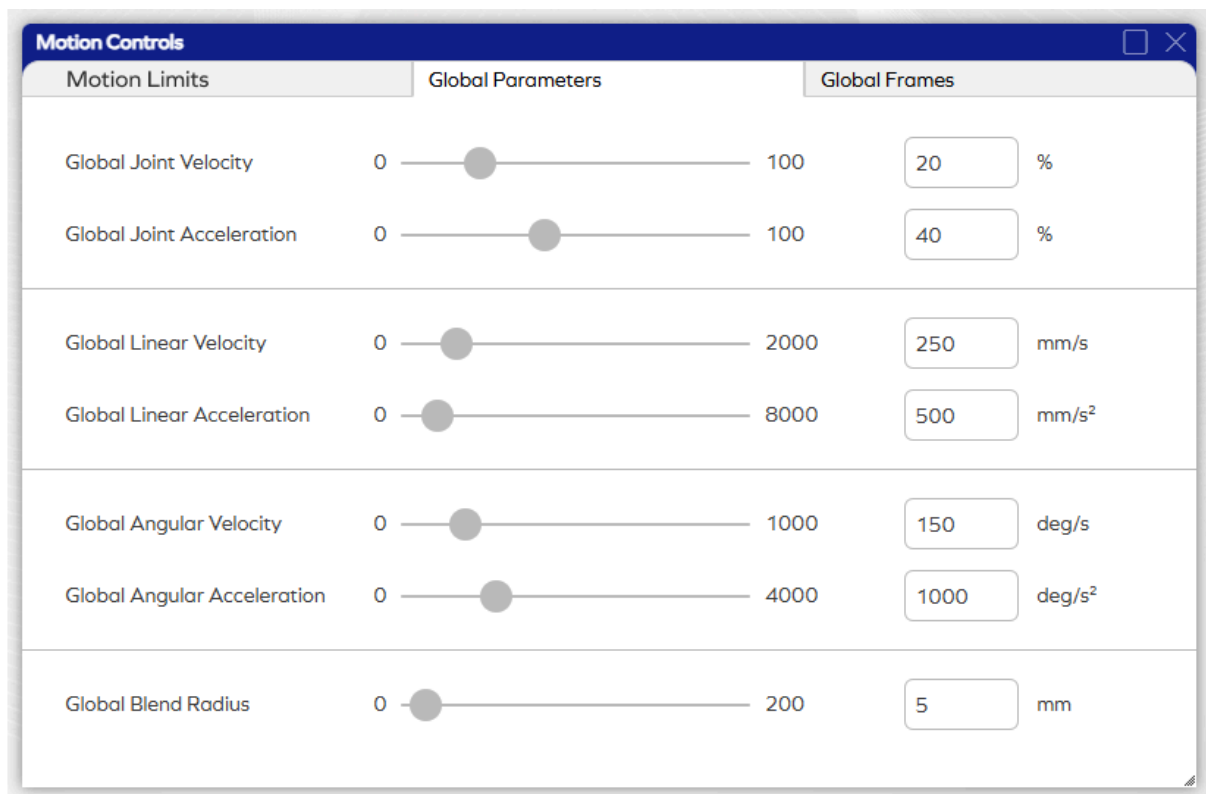


圖 4.2.7.2：運動控制 – 全域參數標籤頁

4.2.7.3 全域坐標系

「全域坐標系」分頁包含用於定義機器人默認坐標模式的設置。雖然「基坐標系」和「法蘭坐標系」是固定連接在機器人上的，但「工件坐標系」和「工具坐標系」可以相對於這些固定坐標系來定義：

- **基坐標系**固定於機器人底部安裝面的中心，使得 z 軸指向機械手臂內部，而 x 軸則指向遠離已連接線纜的方向。
- **法蘭坐標系**固定於機器人頂部安裝板的中心。若機器人所有關節均處於零度位置，其方向將與基坐標系對齊。
- **工作座標系**是使用者在整個程式中用以表示工具中心點 (TCP) 座標的參考座標系。預設情況下，工作座標系與機器人的基坐標系一致。許多移動指令會使用「work_frame」參數，其預設值在此設定為「全域工件坐標系」。
- **工具坐標系**定義了以法蘭坐標系表示的工具中心點 (TCP) 的位置與方向。默認情況下，工具坐標系與機器人的法蘭坐標系一致。許多移動指令會使用「tool_frame」參數，其默認值在此設置為「全域工具坐標系」。

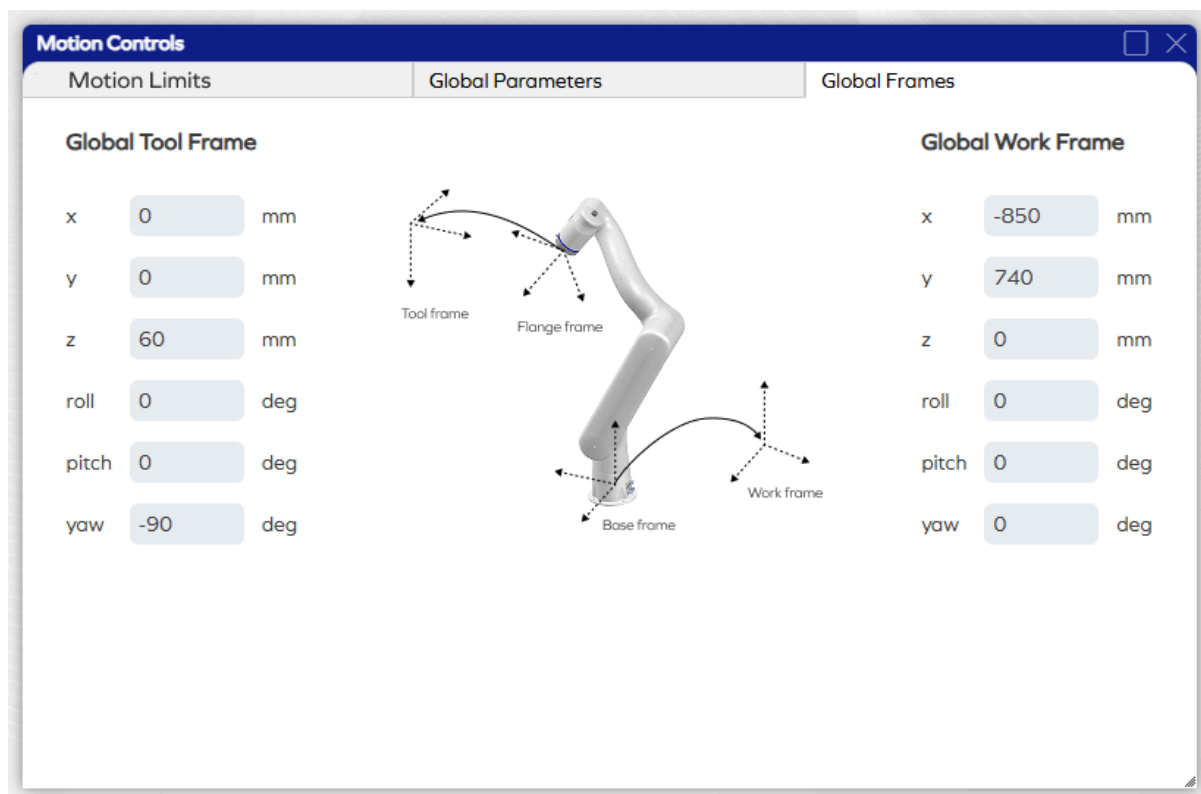


圖 4.2.7.3：運動控制 – 全域坐標系標籤頁

4.3 模塊與資料庫

為了實現簡易且直觀的應用程式編寫，AX Portal 採用了包含各種建構塊的物件導向結構。這些建構塊（以下稱為「元件」）可透過桌面介面存取，各編輯器個別建立。AX Portal 允許使用者智慧地將這些元件相互連結，以物件導向的方式建立應用程式，從而產生簡短、易於閱讀且通常不包含大量數字與坐標的應用程式程式碼。

以下是提供的元件列表：

	程式 主要包含程式代碼（和/或模塊）以及特定於該程式的設置。程式代碼支援 Python 3.13 ，除了基礎的 Python 函數外，還包含許多來自內部機器人控制 API（應用程式接口）的函數。程式是透過「程序編輯器」來創建和修改的。
	位姿是指工具中心點 (TCP) 在特定方向下的位置，由一組關節角度唯一定義。位姿可透過「動作編輯器」進行建立與修改。
	路徑是一系列描述工具中心點 (TCP) 平滑運動的位姿序列，因此也代表了整個機器人的運動軌跡。路徑可透過路徑編輯器進行創建和修改。
	網格 是可自訂的、最多三維的點陣列，用於定義例如預設托盤的位置。網格點可被提取並在程式中用作迭代器。網格可透過網格編輯器進行建立與修改。
	I/O 分為數位與類比訊號。可透過 I/O 配置選單進行配置，且 I/O 的值可在 I/O 選單中查看。I/O 用於連接並與外部裝置通信。

所有這些元素皆儲存於 AX Portal 的資料庫中。您可以隨時透過「配置選單」下載您機器人的資料庫（以備份完整的設定），或上傳新的資料庫（覆寫當前資料庫及其所有元素）。此外，「項目選單」提供資料庫中所有項目與元素的清晰概覽。

4.4 項目菜單

點擊控制台中目前選取的**專案**名稱，即可進入專案選單。此選單包含可用元素（如姿勢、路徑、網格和程式）的概覽。這些元素被分組為多個專案。任何時候僅能選取一個專案，且僅有選取專案中的元素在 AX Portal 的其他選單中可見、可編輯及可用。

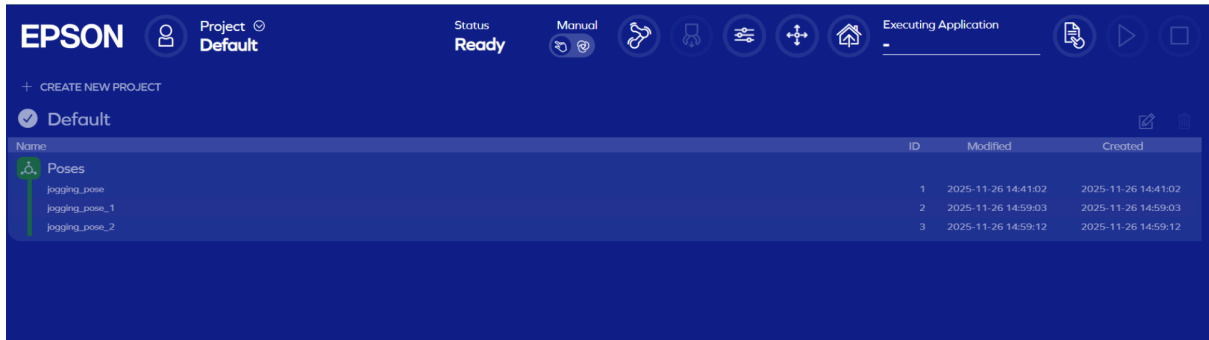


圖 4.4：項目菜單



按下「**創建新項目**」按鈕以建立一個新的、空的項目。



點擊項目名稱左側的圓圈以選擇該項目。



按下項目元素的「**複製**」按鈕，可在同一項目內創建該元素的精確副本。複製後的元素名稱會稍作修改，並以“(副本)”結尾。



按下項目名稱右側的「**編輯**」按鉤，即可變更該項目的名稱。按下項目元素的「**編輯**」按鉤，即可開啟對應編輯器進行查看或修改。



點擊項目名稱右側的「**刪除**」按鑰，即可刪除該項目及其所有元素。點擊項目元素的「**刪除**」按鑰，即可將該特定元素從項目中刪除。

點擊元素圖示（例如：位姿、路徑等）以顯示或隱藏該特定區段中的元素列表。

將元素從一個項目**移動**到另一個項目，請使用鼠標將元素從源項目拖曳並放至目標項目中。拖曳元素時按住 **CTRL** 鍵，可在目標項目中創建該元素的**副本**。

4.5 配置菜單

按下主視窗上的「**配置**」按鈕以開啟配置選單。在此選單中，您可以配置資料庫及硬體設定（I/O 與工具）。若您操作的是用於特定應用的靜態機器人，僅需設定一次配置，之後此視窗僅用於上電啟動機器人。您甚至可透過「設置」菜單中的適當設置，將上電啟動機器人及執行程式的步驟自動化。然而，若您將機器人用於多種用途，則可能更常返回此菜單，並針對不同的項目與目的使用不同的工具，甚至不同的資料庫。

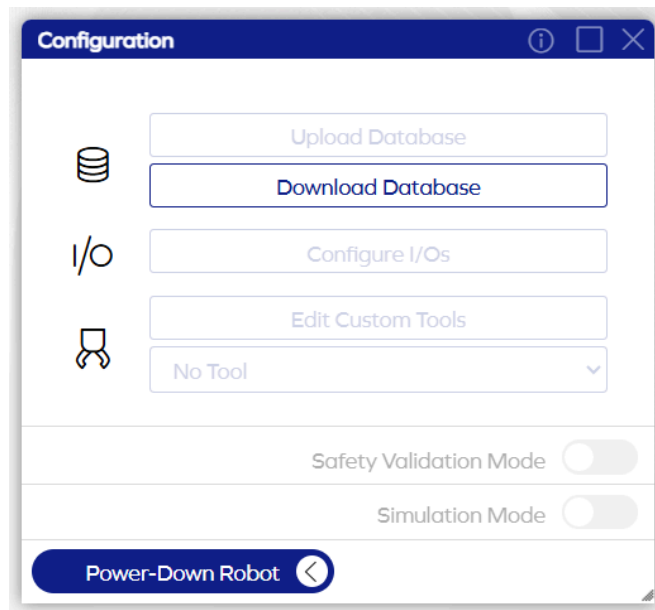


圖 4.5：配置菜單

4.5.1 資料庫（下載與上傳）

在資料庫區段中，您可以下載應用程式資料庫的當前版本，或上傳（並覆寫）另一個相容版本的資料庫。**下載數據庫**將在您的裝置上建立數據庫 ZIP 檔案的本地副本。建議您**定期備份數據庫**，特別是當您擁有大量程式，且已投入大量時間編寫和微調程式時。**上傳數據庫**時，瀏覽器會要求您從裝置中選擇 ZIP 檔案。選擇數據庫檔案後，系統將**覆蓋當前數據庫**並以上傳的檔案取代。

透過在一個機器人上下載數據庫並上傳至另一個機器人，您可以複製所有程式列表和元素到不同機器人之間。建議您定期備份數據庫。

另請注意，數據庫具有版本控制，並非所有數據庫都與 AX Portal 每個版本相容。舊版數據庫可匯入至新版軟體中，但新版數據庫無法加載至舊版軟體中。換言之，數據庫不具向下相容性。

4.5.2 I/O 配置

按下「**設置 I/Os**」按鈕以開啟 **I/Os** 配置視窗。在此處，您可以設置所有 I/Os、將預設功能指派給數字輸入，並為您的 I/Os 指定自訂名稱。自訂名稱可用於應用程式腳本中，以提高腳本的可讀性。在腳本中，您只需將整數 I/Os 識別碼 (int) 替換為 I/Os 名稱 (str) 即可。

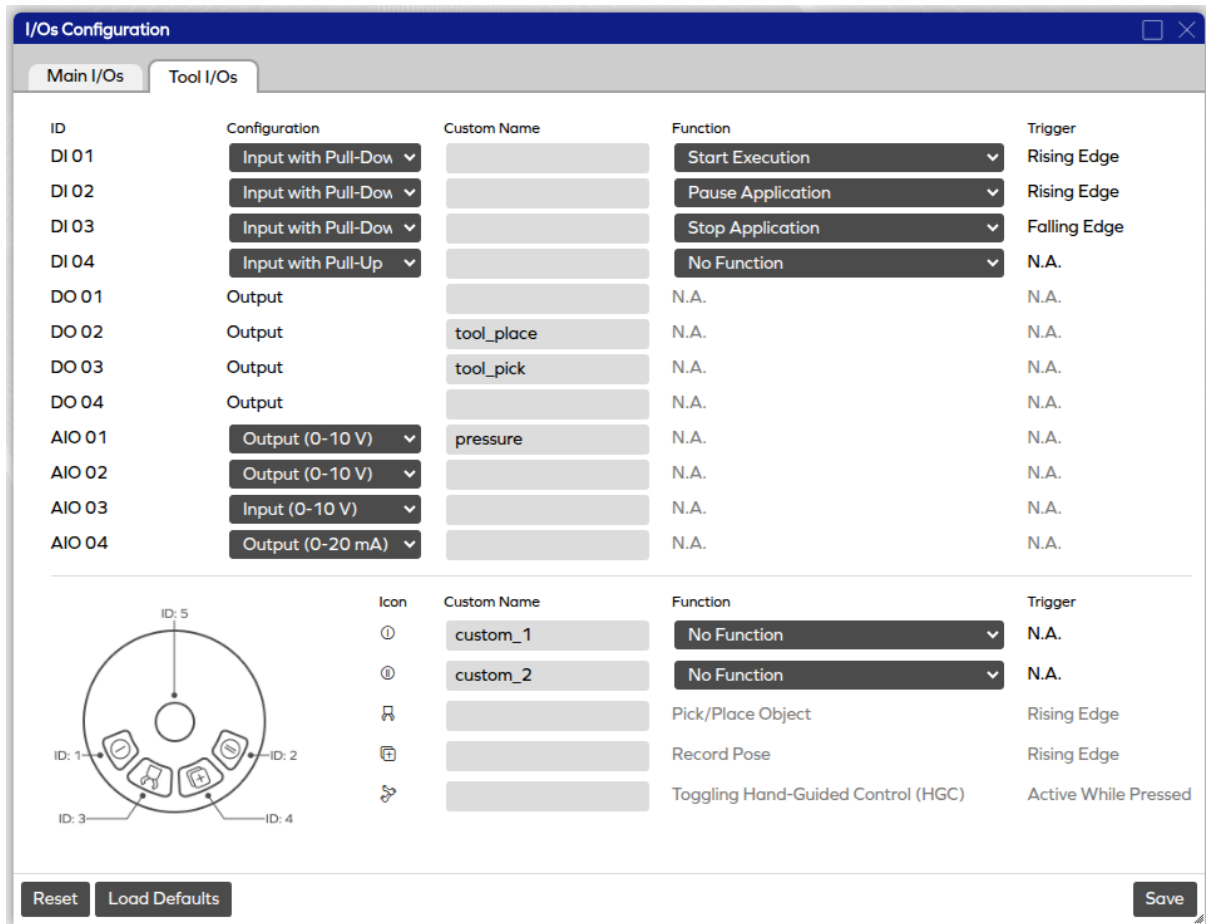


圖 4.5.2.a: I/O 配置菜單

AX 系列在最上方的關節處配備了一個**按鈕介面**。您可以透過此菜單，以與指派其他數字輸入功能完全相同的方式，將功能指派給這兩個自訂按鈕。

請注意，當透過數字輸入觸發任何功能時，當前控制站的活動控制將被撤回。

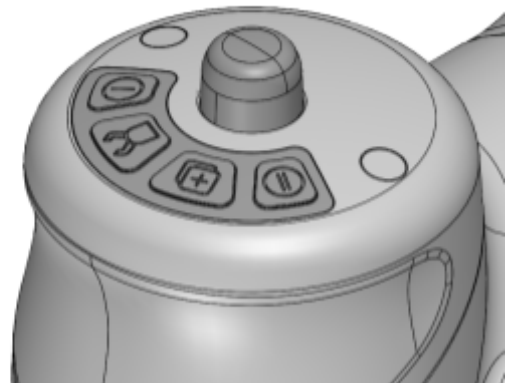


圖 4.5.2.b: AX 按鈕介面

數字輸入提供以下預設功能，您可以設置為上升沿（默認）或下降沿觸發：

無功能	沒有觸發激活任何動作。此為默認設置。
暫停程序	暫停目前正在運行的程式。 此操作類似於按下駕駛艙上的「暫停」按鈕。
中止程序	停止目前運行中或已暫停的應用程式。 這類似於按下駕駛艙上的「停止」按鈕。
開始執行	運行目前選擇的應用。 這類似於按下駕駛艙上的「執行」按鈕。 如果機器人從中斷中恢復，此功能會將機器人返回中斷時的位姿，並繼續之前的運動。 這類似於在中斷發生後，在 GUI 彈出的對話框中按下「繼續」和「繼續運行」按鈕。
重置安全	在中斷後重置安全系統。 這類似於在中斷發生後，透過 GUI 中彈出的對話框重置安全。
啟用 HGC	為所有關節釋放手動引導控制 (HGC)（狀態為就緒時）。
禁用HGC	保持所有關節的手動引導控制 (HGC)（當狀態為手動引導控制時）。
啟用/禁用 HGC	在所有關節的 HGC 啟用與禁用之間切換。觸發的功能取決於當前程式狀態。
抓取物體	觸發工具拾取功能（預設或鏈接至數字輸出）。此操作類似於按下駕駛艙上的「拾取」按鈕。
放置物體	觸發工具放置功能（預設或鏈接至數字輸出）。此操作類似於按下駕駛艙上的「放置」按鈕。
張開/關閉夾具	根據當前工具狀態，切換工具的拾取/放置功能。
已記錄的動作	錄製當前機器人位姿，並儲存為「button_pose」（或類似名稱）。
手動操作模式	將機器人切換到手動操作模式。 這與使用駕駛艙內的開關切換模式類似。
自動操作模式	將機器人切換到自動模式。 這與使用駕駛艙內的開關切換模式類似。

4.5.3 工具配置

在「工具」區段中，您可以配置機器人的工具。如果您即將整合自訂工具（例如來自第三方的工具，或是您為特定應用所開發的專用工具），請使用「**編輯用戶工具**」選項。此操作將開啟已存使用者工具列表，並提供以下選項：

	按下「 創建新工具 」按鈕，即可開啟工具編輯器，用以定義工具的運動學特性和動態性能，以及拾取和放置觸發條件。
	按下「 搜索 」按鈕，可依名稱篩選自訂工具列表。輸入三個或更多字元即可啟用搜索篩選功能。
	按下「 編輯 」按鈕（或雙擊元件上的任意位置），將在獨立窗口中打開工具編輯器，以便查看或修改。
	按下「 刪除 」按鈕，將不可逆地從資料庫中移除該工具。

4.5.4 工具編輯器

工具編輯器分為四個區塊（參見下圖）。要填入這些值，您可能需要研究即將整合至 AX Portal 的工具的資料表，其中可能詳細指定了該工具的物理與電氣特性。

在第一節中，您需要定義**工具中心點（TCP）**，即工具在空間中進行互動時的位姿。對於用於拾取物體的工具而言，這意味著 TCP 定義了工具在拾取過程中被視為所在的位置。TCP 是在附著於頂部連接板正中心的坐標模式中定義的，該坐標模式的方向與基坐標系相同（假設所有關節角度皆為零度）。若未定義 TCP，此坐標模式即為「**默認工具坐標系**」（x、y、z、X軸旋轉、Y軸旋轉、Z軸旋轉及偏航皆為零）。您可從此處平移並旋轉 TCP，使其與自訂工具的交互點相符。

在下一個步驟中，必須調整自訂工具的**動態性能**。重力補償和碰撞檢測等功能的精準度取決於機器人的動態模型，因此值得測量並定義工具的這些性能。您需要定義工具的總質量，以及質量中心位置（與 TCP 相同，該位置定義於機器人的「**默認工具坐標系**」中）。慣性旋轉瞬間可以手動輸入（若已知），或自動估算（AX Portal 無法得知您的工具形狀與質量分佈）。

Tool Editor : pneumatic_gripper

Name* pneumatic_gripper

1. Defining the Global Tool Frame

Translation [mm]

x0y0z36

Rotation [deg]

roll0pitch0yaw90

2. Defining the Dynamic Properties

Mass [kg]

0.662

Center of Mass [mm]

x2y-1.5z14

Rotational Moment of Inertia [kg m²]

AutomaticManual

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

Place

Pick Output

Place Output

T_1

T_1

Cancel

Reset

Save & Close

圖 4.5.4：工具編輯器

最後，若您的工具配備了拾取物體的機構，您可以在此定義哪些數位輸出會由「工具拾取」和「工具放置」功能觸發，這些功能可透過控制台和腳本功能存取。可用的拾取行為如下：

4.5.4.1 單輸出模式

這是所有可用的拾取行為中最簡單的一種。您只需定義一個數位輸出，用於在拾取與放置之間切換，並可設定拾取動作是由 HIGH 訊號或 LOW 訊號觸發。



圖 4.5.4.1：單輸出模式範例，採用 高 觸發拾取（低 觸發放置）

4.5.4.2 雙輸出模式

此拾取行為允許您分別為拾取與放置定義獨立的數位輸出。拾取/放置 切換後，將以（最小）時間延遲 T_1 （單位為毫秒）依序觸發兩個輸出。此外，您可定義有效狀態是指 HIGH 訊號還是 LOW 訊號。

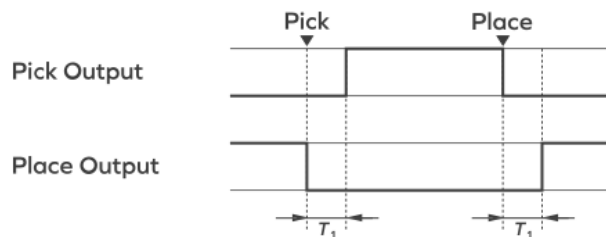


圖 4.5.4.2：高電平作為有效觸發狀態的雙輸出模式範例

4.5.4.3 分離觸發模式

在分離觸發模式下，您可以定義兩個數位輸出：一個用於定義方向（拾取或放置），另一個用於定義觸發（是否觸發所選方向）。對於方向輸出，您可以選擇使用 HIGH 或 LOW 訊號進行拾取（另一種則是放置）。對於觸發輸出，您亦可選擇使用 HIGH 或 LOW 訊號進行觸發。此外，您可定義從選擇方向到觸發所選方向之間的（最小）時間延遲 T_1 （以毫秒為單位），以及定義觸發保持有效時間長度的（最小）時間持續時間 T_2 （以毫秒為單位）。

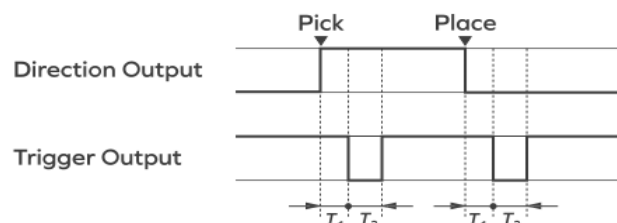


圖 4.5.4.3：分離觸發模式範例，其中 HIGH 代表拾取方向，LOW 代表觸發

4.5.5 機器人上電

載入資料庫並設置 I/Os 及工具後，即可進行機器人上電。通常您無需變更機器人的初始配置，可直接將「**啟動機器人**」滑塊完全拖曳至右側，以觸發初始化例程。

若您不希望啟動硬體，而是僅修改資料庫內容（程式、位姿等）且不使用任何硬體（例如：不移動機器人、不讀寫 I/O），可先啟用「**仿真模式**」開關，然後將「**啟動機器人**」滑塊拖曳至右側。請注意，在仿真模式下，所有動作皆會即時模擬。

當機器人處於上電中時，您將無法更改機器人的任何初始配置。您只能下載數據庫或選擇「**休眠機器人**」；若機器人處於閒置或待機狀態，建議執行此操作。

若您以安全主管身分登入，此處將可存取「**安全驗證模式**」開關。啟用此模式會指示軟體忽略位置、速度及力限制值，藉此可驗證當發生限值違規時，安全裝置是否確實觸發。請注意，在此模式下使用任何「運動控制」功能仍會覆寫這些限制值。

4.6 示教菜單

按下主視窗上的「示教」按鈕即可開啟示教菜單。此菜單包含目前選擇的項目中所有可用動作、路徑及網格的概覽。

姿勢與路徑可進行建立、編輯、複製、刪除，甚至直接執行。另一方面，網格是協助整理重複且等間距坐標的有用工具。請注意，網格並非直接用於移動機器人，而是將一組組坐標值彙整並儲存，以維持程式腳本的結構化，並避免您需手動輸入坐標。網格中的點位列表可被提取以供各種用途，不過通常網格會設定為其點位直接代表 TCP（工具中心點）坐標，這些坐標隨後將作為移動函式的輸入值。

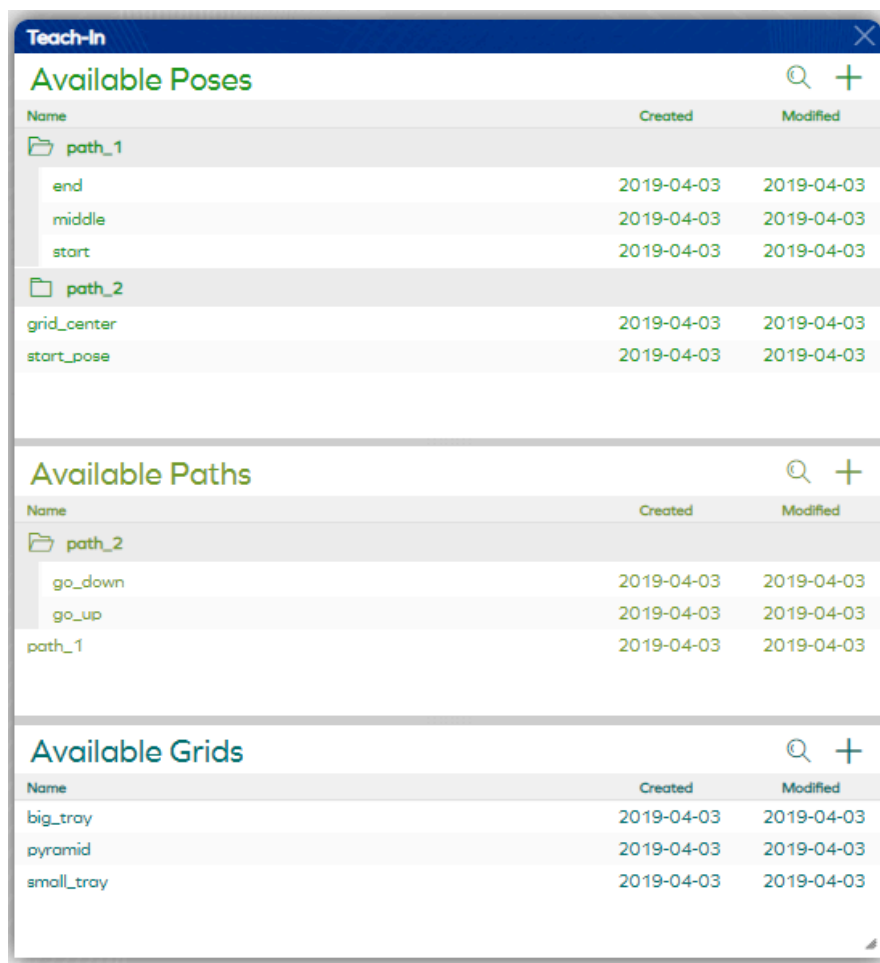


圖 4.6：示教菜單

	按下「 加號/新增 」按鈕，即可在新視窗中開啟位姿、路徑或網格編輯器。本節稍後將詳細說明所有這些編輯器。
	按下「 搜索 」按鈕，可依名稱篩選姿勢、路徑或網格列表。輸入三個或更多字元即可啟用搜尋篩選功能。
	按下「 播放 」按鈕，可將機器人移動至特定位姿或執行路徑。請注意，執行路徑會先將機器人移動至路徑中的起始位姿，然後才開始執行路徑。
	按下「 複製 」按鈕，可建立此位姿或路徑的精確副本。複製後的元素名稱會稍作修改，並以「(副本)」結尾。
	按下「 編輯 」按鈕（或雙擊元素上的任意位置），即可在獨立視窗中開啟此位姿、路徑或網格，以便檢視或修改。
	按下「 刪除 」按鈕，將不可逆地從資料庫中刪除該位姿、路徑或網格。
	點擊一個資料夾以顯示或隱藏該資料夾內的動作、路徑或網格。元素僅在名稱中包含一個或多個「/」字元時，才會顯示在資料夾（及其子資料夾）中，例如「資料夾/子資料夾/名稱」。

4.6.1 動作編輯器

動作編輯器允許使用者創建新動作或修改現有的動作。請注意，變更動作的坐標或配置並不會自動移動機器人。若需移動機器人，應改為使用精細控制。

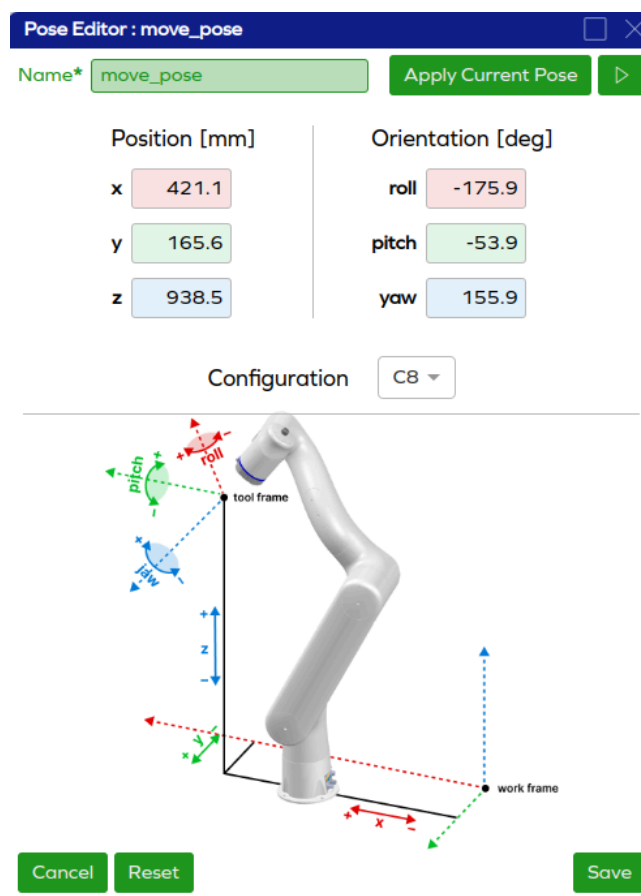


圖 4.6.1：動作編輯器

姓名:

指定位姿在資料庫中儲存的名稱。可在名稱中使用「/」字元，將您的位姿以類似資料夾的結構進行群組。

位置:

表示以毫米為單位的工具中心點 (TCP) 位置，此位置是相對於全域工件坐標系定義的。請注意，可以輸入無效的位置值（例如機械臂無法到達的位置）。然而，這些位置會以紅色標示，且按下播放按鈕時會導致錯誤。之所以允許這樣做，是因為當使用不同的坐標模式和偏移量時，該位姿可能仍然有效。

方向:

表示以度為單位的工具中心點 (TCP) 方向，相對於全域工件坐標系定義。方向採用航海角表示法。與位置值相同，無效的方向值會以紅色標示。

配置:

指定機械臂的配置。使用者可透過下拉式選單從所有可用配置（C1–C8）中進行選擇。請注意，修改配置並不會改變 TCP 的位置或方向，僅會影響用於實現指定位姿的關節角度。

動作編輯器中的其他執行操作

保存	點擊「 保存 」按鈕可將當前位姿以指定名稱儲存。若未輸入有效名稱，則無法儲存位姿。
取消	點擊「 取消 」按鈕將放棄編輯器中所有未儲存的更改，並關閉窗口。
重置	點擊「 重置 」按鈕將撤銷所有未儲存的變更，並將位姿還原至上次儲存的狀態。
 演示	點擊視窗頂端的「 演示 」按鈕，會命令機器人移動到位姿。
應用當前位置	點擊「 應用當前位置 」按鈕，將更新所有欄位以反映機器人的當前位姿值。

4.6.2 路徑編輯器

路徑編輯器允許使用者建立新路徑或修改現有路徑。它分為三個可編輯區段：**名稱**、**初始姿勢**以及**此路徑的區段**。

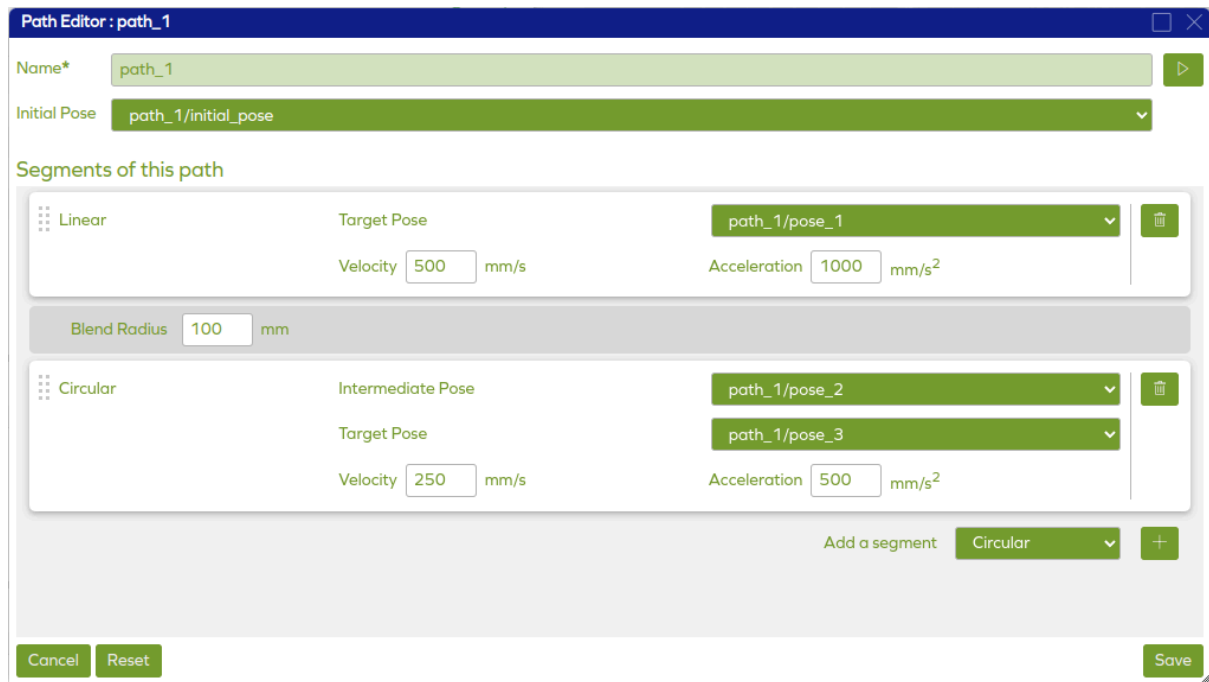


圖 4.6.2：路徑編輯器

姓名:

指定路徑在數據庫中儲存的名稱。可在名稱中使用「/」字元，將路徑以類似資料夾的結構進行分組。

初始姿勢:

定義路徑的起始位姿。機器人先使用默認的速度和加速度設置移動到此位姿，然後才依次執行已定義的路徑區段。

此路徑的各部分：

列出並定義構成路徑的各個區段。可用的區段類型如下：

- 直線運動區段
- 圓弧運動段
- 坐標區段
- 位姿區段
- 方向段

各區段的詳細描述請參閱「區段」章節。

要添加一個段，請從右下角的下拉式菜單中選擇所需的分段類型，然後點擊加號 (+) 按鈕。添加段後，可透過拖放功能自由調整其**順序**。

每個區段都需要指定一個**目標姿勢**，該姿勢定義了該區段的最終位姿。目標姿勢必須預先在「動作編輯器」中定義，之後可透過下拉式菜單進行選擇。圓弧運動區段獨特地需要一個**額外的中間姿勢**，該姿勢也必須從預定義姿勢列表中選擇。

每個區段皆可設定速度與加速度參數。若未指定，執行時將採用全域默認值。

可配置混合半徑以控制區段間過渡的平滑度與效率。此參數定義了當前區段開始位姿相較于前一區段終端位姿的最大允許偏差（單位為毫米）。

路徑編輯器中的其他功能

路徑編輯器還提供其他功能，可有效管理、編輯和整理路徑。

保存	點擊「 保存 」按鈕可將當前路徑以指定名稱儲存。若未輸入有效名稱，則無法儲存路徑。
取消	點擊「 取消 」按鈕將放棄編輯器中所有未保存的更改，並關閉窗口。
重置	點擊「 重置 」按鈕將撤銷所有未保存的變更，並將路徑還原至上次保存的狀態。
 演示	點擊視窗頂端的「 播放 」按鈕，將指令機器人執行已定義的路徑。若機器人當前的位姿與路徑指定的初始位姿不符，它將先移動至該初始位姿，然後才開始執行。
 刪除	點擊某個路徑區段旁的“刪除”按鉤，將從路徑中刪除該區段。

4.6.3 網格編輯器

創建新網格時，網格編輯器會自動開啟。此編輯器將引導使用者逐步設置所有可用的網格選項。請使用「**下一個**」和「**返回**」按鈕導航此精靈，直到定義出符合需求的網格點有序列表為止。若要編輯現有網格，網格編輯器會直接跳至手動操作模式。

第一步，使用者必須選擇「**網格維度**」。通常，網格被視為二維陣列，但它也可以是獨立於第二維度的簡單列表（例如從輸送帶上拾取物體），甚至可以是三維結構（例如將物體堆疊在箱子中）。網格初始化時包含完全對齊的點，但請注意，單個網格點日後隨時可進行修改、新增或刪除。

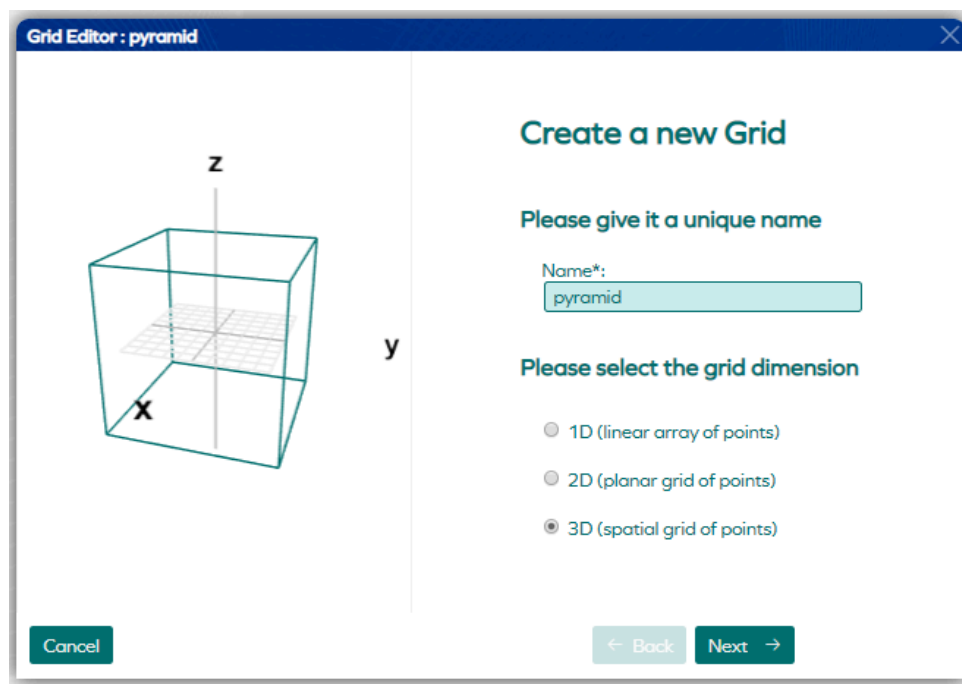


圖 4.6.3.a:網格編輯器（第一步）

在第二步中，使用者必須選擇所選網格的**角點**。不同維度的網格具有不同的角點數量，而角點的標記方式為「A」至例如「D」（針對二維網格）。使用者可直接在輸入欄位中輸入角點座標值（以機器人的基坐標系為準），或直接引導末端執行器的工具中心點(TCP)至網格的角點，並按下相應的「示教」按鍵以儲存坐標。

一旦教導的頂點數量足夠以定義網格（例如二維網格至少需 3 個點），使用者即可按下「**網格校正**」按鈕，系統將評估已給定的頂點，並自動填入缺失的值，使其符合平行四邊形（二維）或長方體（三維）的形狀。若啟用額外的「**自動對齊**」標記，評估程式將考量水平與垂直坐標軸及平面，並嘗試將生成的網格沿著這些軸與平面進行對齊（若距離足夠近）。

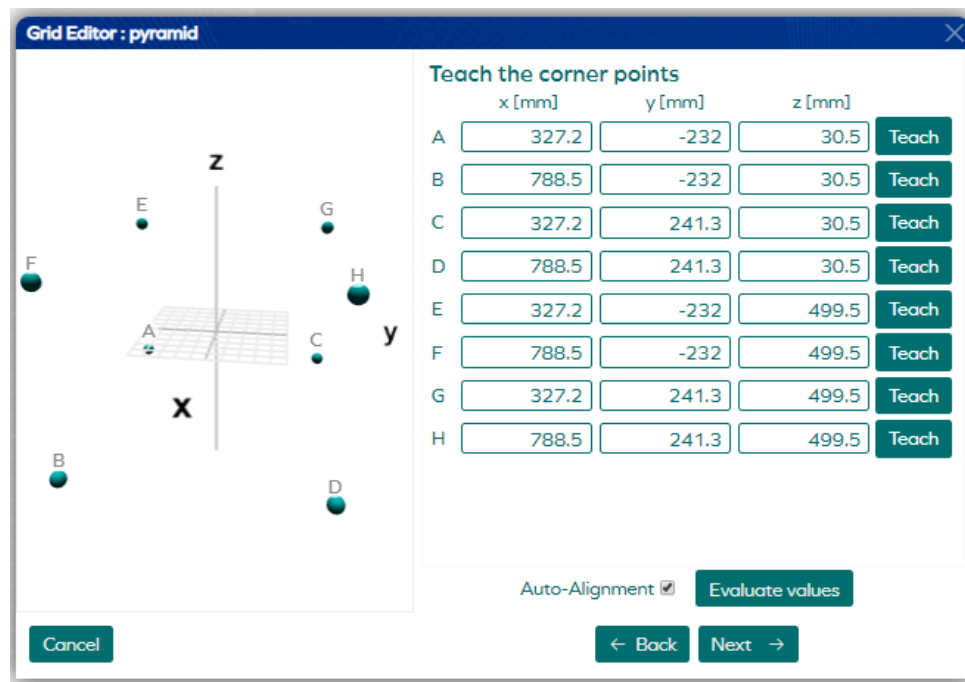


圖 4.6.3.b: 網格編輯器（第二步）

在第三步中，系統會要求使用者選擇點數、特殊延伸選項及迭代方向。讓我們詳細看看這些可用的選項：

- 各維度的「**點數**」可分別設定。若難以區分各維度（例如行與列），請參照先前示教的關鍵點標籤。
- 對於二維網格，存在一種**延伸選項**，可用於在每組四個相鄰點之間添加中間點。對於三維網格，此選項不僅適用於每個網格平面，也適用於空間中每組八個相鄰點。
- 「**迭代方向**」指的是點被返回並用作程式循環中迭代器的方向，其由起始點以及每個方向的向量所決定。最後一個方向總是根據上一次選擇的選項而定，僅供參考及視覺驗證之用。下拉菜單中的字母對應於該網格各角點的標籤。

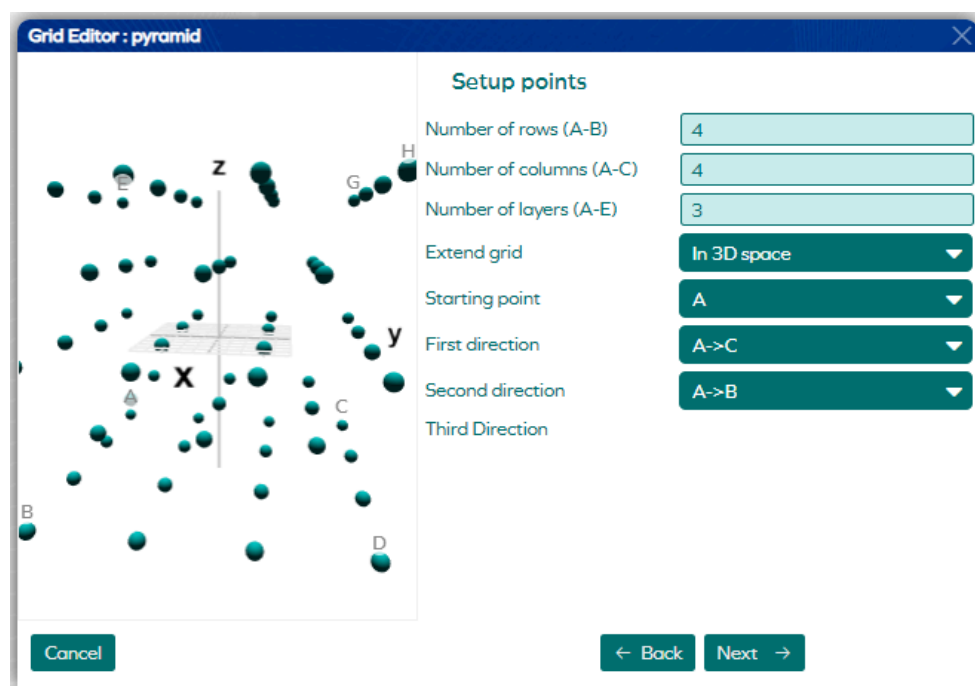


圖 4.6.3.c:網格編輯器（第 3 步）

最後，「網格點列表」已就緒供手動編輯。在此，可複製、刪除或移動單個網格點，並手動更改坐標值。同時，可視化模型會顯示網格點的順序及其相對坐標。

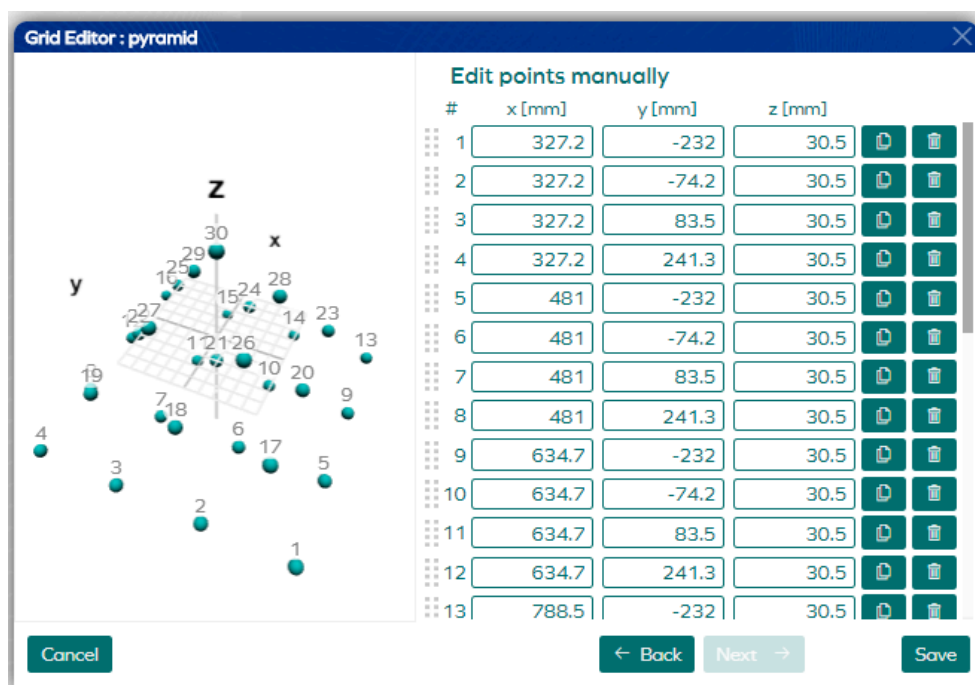


圖 4.6.3.d:網格編輯器（第 4 步）

編輯過程中，您隨時可返回檢視先前設定的選項與值。您亦可變更這些選項，但此操作將重置後續視窗中的所有變更。

4.7 程式菜單

按下主視窗上的「**應用程式**」按鈕即可開啟應用程式菜單。此菜單包含所有已儲存的程式的概覽，以及最後執行的程式的輸出結果。使用者亦可透過此菜單開啟「**程序編輯器**」視窗，以創建新程式或編輯現有程式。

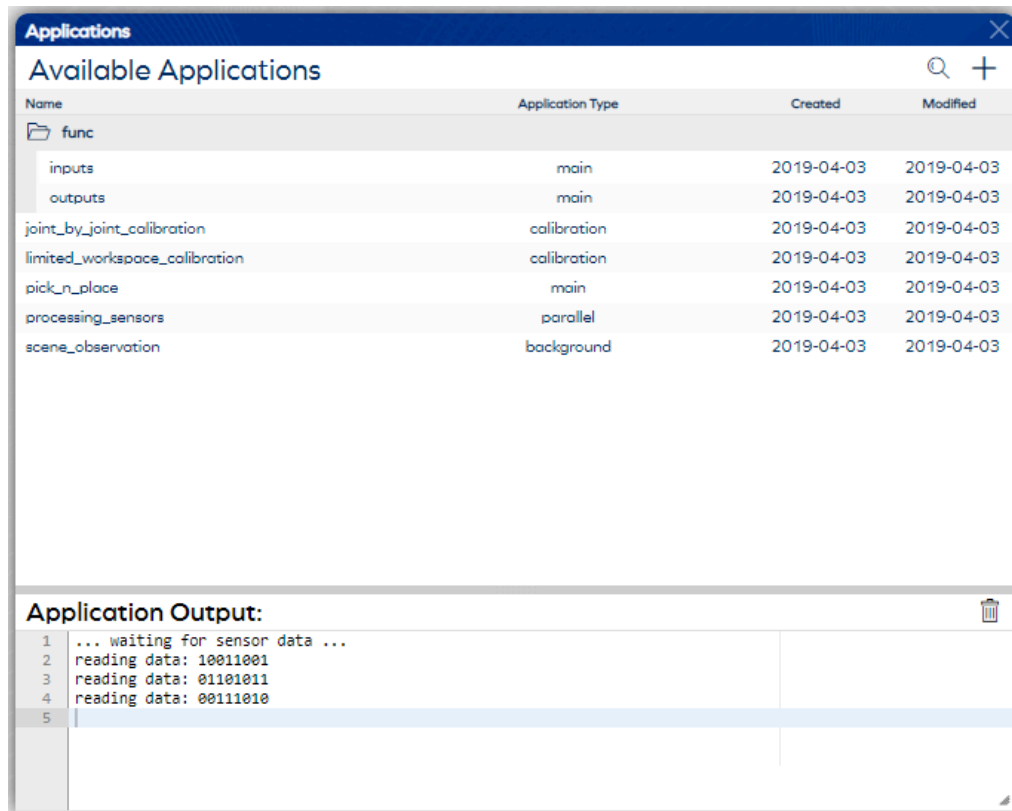


圖 4.7：程式列表菜單

	按下「 加號/新增 」按鈕可創建新程式。此操作將開啟程式編輯器，其功能將於本節後文詳細說明。
	按下「 搜索 」按鈕可依名稱篩選程式列表。輸入三個或更多字元即可啟用搜索篩選功能。
	按下「 執行 」按鈕以執行程式。請注意，僅有「主」程式可透過此方式執行，因為在控制台中僅能選擇「主」程式。
	按下「 複製 」按鈕可創建此程式的完全副本。複製項的名稱會稍作修改，並以「(副本)」結尾。
	按下「 編輯 」按鈕（或雙擊元素上的任意位置），即可在獨立視窗中開啟此程式以進行檢視或修改。
	按下「 刪除 」按鈕，將不可逆地從您的資料庫中移除該應用程式。此操作將移除程式腳本以及該應用程式的特定設置。
	點擊一個資料夾以顯示或隱藏該資料夾內的程式。只有當元素名稱包含一個或多個「/」字元時（例如「資料夾/子資料夾/名稱」），這些元素才會顯示在資料夾（及其子資料夾）中。您可以點擊出現的「 編輯 」按鈕以進入文件夾編輯器。

4.7.1 程序編輯器

「應用程式」菜單中的「已存程序」區段會列出所有先前儲存的程式。您可以點擊「**編輯**」按鈕來修改現有程式，或按下「**創建新程序**」按鈕來創建新程式。無論哪種情況，都會彈出「**程序編輯器**」視窗。其功能在兩種情況下皆相似。在深入探討細節之前，讓我們先快速瀏覽程式編輯器的主要按鈕：

保存	按下「 保存 」按鈕，系統將以指定名稱儲存該程式。若未指定名稱，則無法儲存程式。您可在名稱中使用「/」字元，將程式以類似資料夾的結構進行分組。
取消	按下「 取消 」按鈕，編輯器中所有未儲存的變更將被拒絕，並關閉編輯器。
▶ 測試	按下視窗頂端的「 測試 」按鈕，程式將以測試模式執行。測試模式表示程式腳本會執行，但不會阻止您繼續編輯該程式。在測試程式後所做的變更將不會被執行。

4.7.2 應用程式程式腳本

「**程序腳本**」分頁是程式編輯器三個子視窗中的主要分頁。此區塊會顯示程式腳本，並可選擇性地顯示包含 AX Portal 函數接口與元件的函數庫，以便將其插入程式腳本中。內建的程式語言為 **Python 3.13**，具備其大部分基礎功能，並包含若干預先安裝的 Python 套件。此外，AX Portal 還附帶一個龐大的函數接口，旨在簡化機器人應用程式的編寫（例如：運動機器人、讀取感測器資料、與使用者通信等）。有關這些函數接口的詳細資訊，請參閱《代碼文檔》。

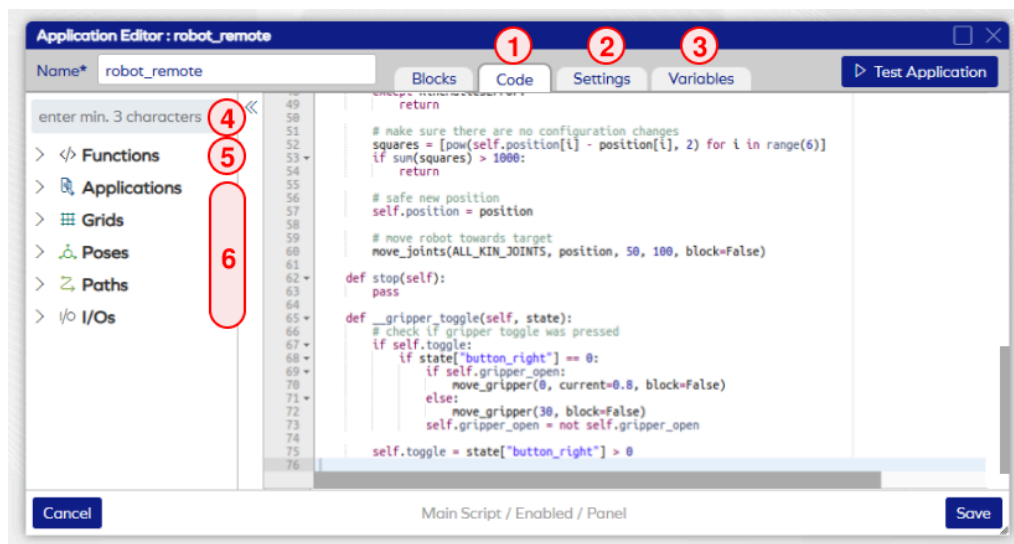


圖 4.7.2：程序編輯器（程序腳本區塊）

1. 在此處編寫您的 Python 程式腳本，2. 修改程式設置，3. 定義和修改程式變數，4. 搜索篩選器，5. 函式庫，6. 可用的元件

點擊編輯器左側的任一**功能圖示**或**元件圖示**，即可開啟元件**面板**。您將在此找到所有可用功能的完整列表。此外，您可透過雙擊所有元件來將其整合至程式腳本中，因此無需記憶元件名稱。

提示：透過移除雙擊函數時插入程式腳本中的可選參數，可提升 Python 代碼的可讀性。例如，您可以寫成

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

，而非

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

4.7.3 程式設置

「**程序設置**」分頁包含一些程式的額外設置，包括

- 一個可禁用的「**可執行**」標記，例如當您的應用程式仍在構建中且不應被執行時，
- 一個標記，用於在項目面板中**顯示**此程式，
- 上傳並替換面板介面中顯示**圖片**的功能，
- 應用程式**標籤**，用於將程式名稱翻譯為面板介面當前選定的介面語言，
- 以及最重要的「**程序類型**」，其說明詳見下一節：

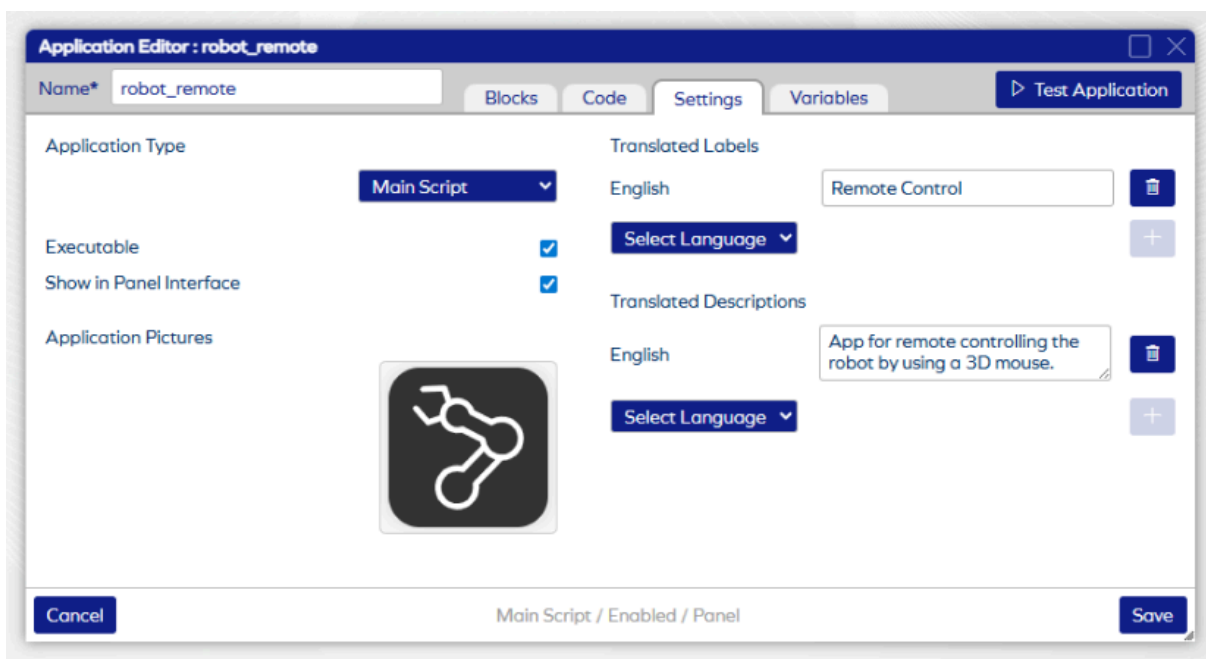


圖 4.7.3.a: 程式編輯器（設置區段）

「**程序類型**」指定了程式應在何種情境下執行，因此決定了程式的執行方式，以及該程式內可使用的功能：

- **主要腳本**包含待執行的主要應用程式。它們旨在使用所有可用的腳本函數，並直接控制機器人及其外部裝置。通常，一個簡單的應用程式可以寫成單個腳本，但對於長篇或複雜的應用程式，或是為了提高可讀性，可以將應用程式拆分為多個依次執行的子腳本。不過，多個主要腳本從不並行執行，若需並行處理，則需使用「並行腳本」。您亦可使用「import」語句在主要腳本之間匯入函數。請注意，控制台中僅能選擇主要腳本。
- 另一方面，**並行腳本**在獨立的執行緒中執行，因此專為與主應用程式並行執行而設計。這在某些情況下非常有用，例如持續讀取感測器資料，或是利用機器人移動的空閒時間執行耗費運算資源的計算。並行腳本不得使用所有腳本功能，例如控制機器人，因為若未仔細留意，這可能會嚴重干擾程式流程，或引發不必要的移動行為。當主應用程式暫停時，所有並行腳本也會隨之暫停；一旦主應用程式停止執行，這些腳本便會終止。
- **後臺腳本**是在 AX Portal 後臺執行的腳本，其執行與 AX Portal 的狀態無關，因此適用於持續監控、檢查或記錄資料，或是為項目定義初始全域變數。每個項目僅能包含一個後臺腳本。若已定義此腳本，當對應項目加載時（例如啟動項目或切換項目時），它將自動執行。後臺腳本不受狀態限制的約束，且若包含循環，可能會持續執行直至機器人斷電中為止。每當後臺腳本開始或結束執行時，系統都會顯示相應的通知。您可透過控制台啟動、重啟或停止後臺腳本。



圖 4.7.3.b: 後臺腳本控制器

4.7.4 應用程式變數

「**應用程式變量**」分頁提供拖放介面，可將不同類型的狀態變數新增至此應用程式。狀態變數僅能在主要腳本中使用。它們用於面板介面，在該處非常實用，例如用於監控應用程式的狀態與進度，或為應用程式指定特定輸入值。請分別使用 **set_state_variable(...)** 和 **get_state_variable(...)** 函式來寫入和讀取狀態變數。

以下是變數類型的概述及其在面板介面中的顯示方式：

布爾值型	布爾值以簡單的開關式滑塊表示，狀態為啟用或禁用。其讀寫時採用 Python 的 bool 類型，其中啟用對應 True ，禁用對應 False 。
整數	整數以允許數值輸入的文本欄位呈現，包含在編輯模式下用於增減數值的 +/- 按鈕。其讀寫皆採用 Python 的 int 類型。
浮點數	浮點數以文本欄位呈現，在編輯模式下允許輸入數值。其讀寫皆採用 Python 的 float 類型。
滑塊	百分比變數在編輯模式下以滑塊形式呈現，並設有強制性的最小值與最大值。其讀寫均採用 Python 的浮點數類型。
文本	文本變數在編輯模式下以文本欄位呈現。其讀取與寫入均採用 Python 的 str 類型。
下拉選單	下拉式變數在編輯模式下以下拉式列表呈現。其讀寫皆採用 Python 的 str 類型。

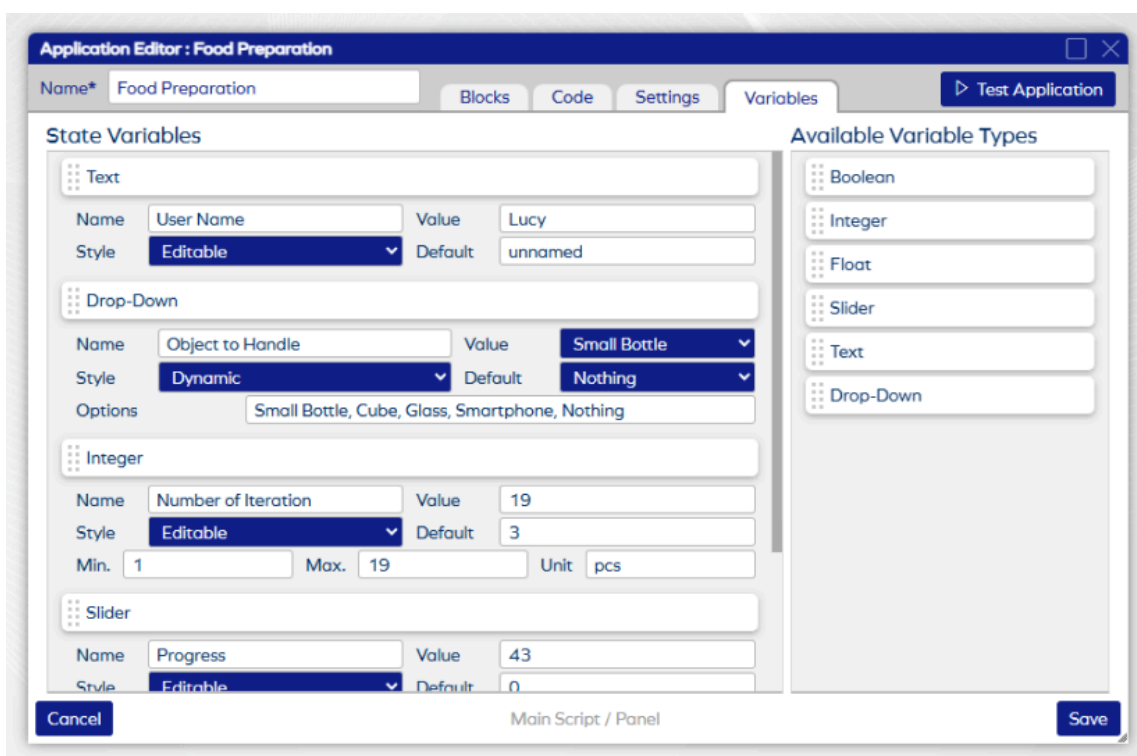


圖 4.7.4：程序編輯器（變數區段）

以下是變數選項的簡要概述，這些選項會因變數類型而異：

姓名	為變數指定一個適當的名稱，日後可透過該名稱以字串形式存取。
數值	變數的值包含該變數當前設定的數值，該數值將以上述說明的 Python 對應類型進行讀取與寫入。
默認	變數的默認值是指在面板介面中使用對應按鍵重置變數時，該變數所採用的值。
樣式	變數的樣式定義了該變數是否、何時以及如何可編輯。
最低值	數值類型允許為該數值設置最小限制。若輸入的數值低於最小值，系統將套用最小值。百分比變數必須設置最小值，而整數/浮點數則為選填。
最大值	數值類型允許為該數值設置最大限制。若輸入的數值高於最大值，系統將套用最大值。百分比變數必須設置上限，而整數/浮點數則為選填。
單位	對於數值類型，您可以為該數值附加一個可選的單位字串。
選項	下拉式變數的選項可輸入為以逗號分隔的字串。

以下是變數樣式簡要概述，這些樣式定義了變數的使用方式：

- **可編輯**變數允許在執行程式之前更改值，但無法在執行過程中更改。
- **受保護**變數可編輯，作為保護層，您必須先按下編輯按鈕才能修改值，並按下儲存按鈕才能接受值變更。
- **動態**變數可編輯，您甚至可以在執行期間動態變更其值。
- **動態和保護**變數兼具受保護變數和動態變數的特性，因此可在執行中透過按下編輯按鈕進行編輯。
- **只讀**變數不可編輯，其值只能由應用程式更改。它們通常用於監控應用程式的狀態或進度。
- **隱藏**變數不會顯示在項目面板介面中，但可像其他狀態變數一樣進行讀寫。

4.7.5 共享全域變數

除了上述特定於應用程式的狀態變數外，亦可定義共享全域變數，以便在並行執行中的多個腳本或同一應用程式的多個工作階段之間共享資料：

- **共用變數**僅在單個程式的執行期間存在。請使用 `set_shared_variable(...)` 和 `get_shared_variable(...)` 函數，在該程式中並行或順序執行的多個腳本之間共享值。
- **全域變數**不僅能在同一應用程式的腳本之間共享值，更能跨所有應用程式共享，且即使重新啟動機器人後仍會保留。因此請使用 `set_global_variable(...)` 和 `get_global_variable(...)` 函式。

4.7.6 文件夾編輯器

若您的應用程式名稱包含字元「/」，該應用程式將自動被歸類至資料夾結構中。您可以透過將遊標懸停在該資料夾上時出現的編輯按鈕來編輯資料夾。這將開啟文件夾編輯器。此編輯器僅包含用於翻譯資料夾名稱的標籤（類似於程式編輯器）。資料夾標籤與程式標籤都會出現在面板介面中，如此一來，面板介面便可輕鬆設定為支援多種語言。

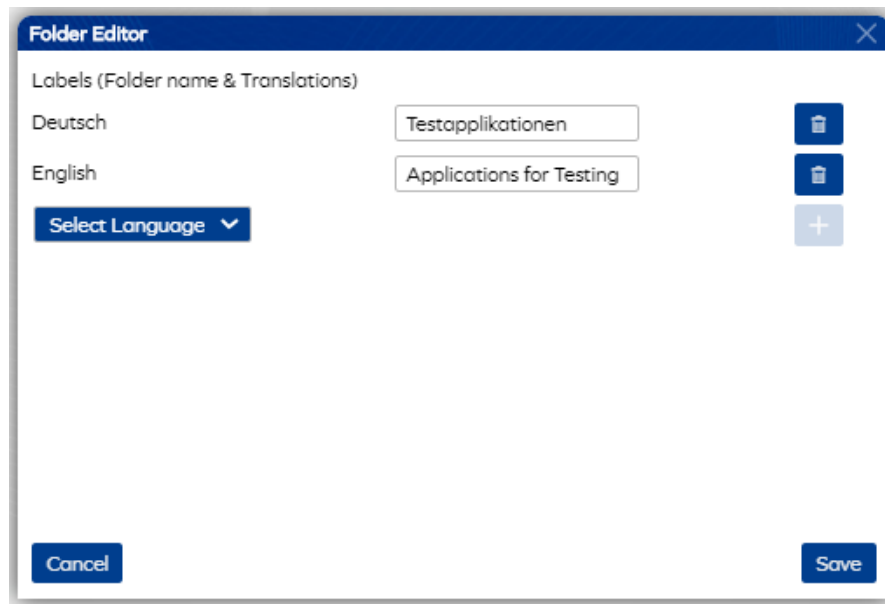


圖 4.7.6：文件夾編輯器

4.7.7 塊程式設計

程式編輯器的「**模塊**」分頁包含一個圖形化程式設計介面。透過此介面，使用者可將程式模塊拖放至程式流程區域，藉此建立簡單直觀的程式流程。

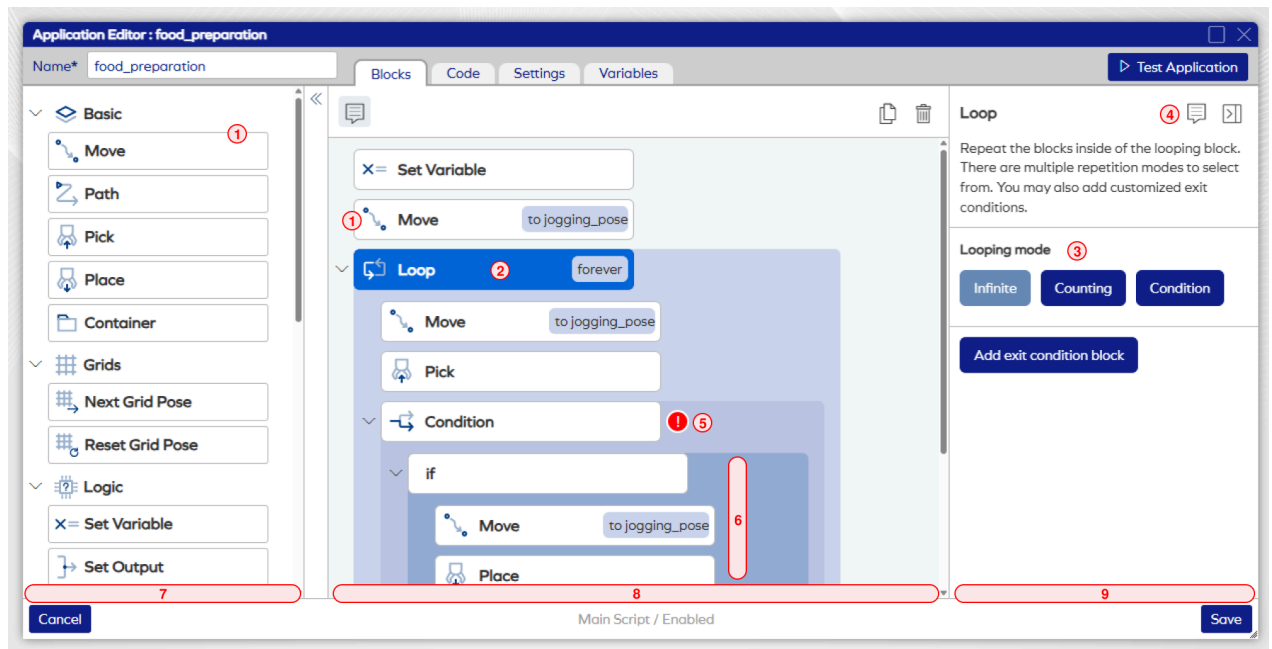


圖 4.7.7：程序編輯器（塊程式設計）

1. 將程式模塊拖曳至程式區域, 2. 選擇要修改的程式模塊, 3. 修改程式模塊詳細信息, 4. 注釋, 5. 錯誤指示器, 6. 程式模塊容器, 7. 可用程式模塊, 8. 程式流程, 9. 程式模塊詳細信息。在程式流程中選擇一個程式塊，即可查看其詳細信息。部分程式塊的詳細信息必須填寫，程式才能執行（例如：對於運動程式塊，您需要提供機器人應移動至何處的信息）。其他程式塊的詳細信息則屬選用（例如：若未提供則使用默認值），或可用於提升程式的可讀性（例如：注釋）。若模塊未正確定義（例如：未定義必填參數，或提供無效數值），程式流程中將顯示錯誤指標；選擇該模塊時，模塊詳細信息中則會顯示錯誤原因。

某些模塊在加入程式流程時會產生彩色容器。這是以視覺化標示這些模塊已分組，且總是依序（由上而下）執行的輔助功能。循環與條件語句即為此類模塊的範例，它們負責執行一組模塊多次，或僅在特定條件成立時才執行該組模塊。

此外，選擇的模塊可被刪除，或複製（並拖曳至程式中的指定位置）。

4.7.7.1 使用程序列表變數與運算式

在詳細說明所有模塊之前，請注意：在模塊詳細信息中的多數文本欄位中，皆可使用數學與邏輯運算式。這些欄位亦支援插入預先定義的程序變量值，以及數字輸入與類比輸入的數值。

這點透過範例說明最為簡明。假設我們已定義以下變數：

State Variables

Boolean

Name: Value: ☐

Style: **Editable** Default: ☐

Integer

Name: Value:

Style: **Editable** Default:

Min.: Max.: Unit:

圖 4.7.7.1.a:變數範例

有效**數值**運算式的範例：

value (number)
42

value (number)
-7 * (8.2 - 3)

value (number)
{counter} + 1

value (number)
{Tool AIO 02}

有效**邏輯**運算式的範例：

value (bool)
FALSE

value (bool)
NOT {flag}

value (bool)
{flag} AND ({counter} > 5)

value (bool)
{Main DIO 09}

圖 4.7.7.1.b:數值與邏輯運算式範例

變數數值與輸入內容亦可直接輸入至一般文本欄位中：

value (text)
The counter is set to {counter}.

圖 4.7.7.1.c:文本運算式範例

4.7.7.2 移動模塊

「**運動**」模塊會執行運動函數，將機器人從當前位置移動至目標位置。使用者必須選擇運動風格……

- **點到點運動** → 以時間最優方式移動至目標，軌跡無限制
- **直線運動** → 沿直線運動路徑移動至目標，方向以線性插值計算
- **圓形** → 沿圓弧運動軌跡移動至目標，方向以直線運動插值計算，且使用者必須在圓弧運動軌跡上指定一個**中間位置**

……以及目標格式。

- **位姿名稱** → 選擇（先前儲存的）目標姿勢名稱
- **網格名稱** → 移動至（先前儲存的）網格點，初始時將移動至該網格的第一個點，然後使用「**下一個網格位姿**」模塊來增加網格點計數器
- **關節角度** → 指定目標姿勢的關節角度
- **坐標** → 指定目標姿勢的坐標

此外，使用者可修改以下選項設置。請注意，若未指定，系統將採用默認值或全域值。

- **速度設置**
 - **關節速度/加速度** → 指定點到點運動的速度
 - **線速度/加速度** → 指定直線運動/圓弧運動的速度
 - **角速度/加速度** → 指定僅改變 TCP 方向（位置保持不變）的移動速度
- **坐標系設置**
 - **工具偏移** → 將 TCP 相對於法蘭坐標系（機械臂頂端）進行偏移
 - **工具坐標系** → 調整 TCP 相對於法蘭坐標系（機械臂頂端）的位置
 - **工件偏移** → 將坐標參考系相對於基坐標系（機械臂底部）進行偏移
 - **工件坐標系** → 針對基坐標系（機械臂底部）調整坐標參考系
- **其他設置**
 - **相對坐標** → 若啟用，坐標數值將加到當前坐標上
 - **阻塞運動** → 若啟用，程式流程僅在運動完成後才會繼續

4.7.7.3 路徑區塊

「**路徑**」區塊會觸發先前定義的路徑。使用者需選擇要執行的路徑。執行此區塊後，機器人會先移動至路徑的初始姿勢，接著以平滑的動作沿著路徑段執行。

4.7.7.4 拾取區塊

「**拾取**」區塊會向連接的工具發送拾取指令。對於角式或平行式夾具，這表示關閉夾具以將物體夾在內部；對於氣動夾具，這表示啟用真空泵以將物體固定在原位。此區塊不附帶參數。

4.7.7.5 放置區塊

放置區塊會向已連接的工具發送放置指令。對於角式或平行式夾具，這表示張開夾具以釋放其中的物體。對於氣動夾具，這表示禁用真空泵以將物體放置到位。此區塊不附帶參數。

4.7.7.6 容器模塊

「**容器**」區塊僅用於將多個區塊群組。它本身並未被賦予任何特殊功能。

4.7.7.7 下一個網格位姿模塊

「**下一個網格位姿**」區塊會增加特定網格的內部位姿計數器。需先選擇受影響的網格。每當使用「**移動**」區塊朝網格移動時，其目標即為該計數器當前指向的位姿。若要實際移動至網格的所有點，可將「**移動**」與「**下一個網格位姿**」區塊置於**循環**結構中，如下所示：

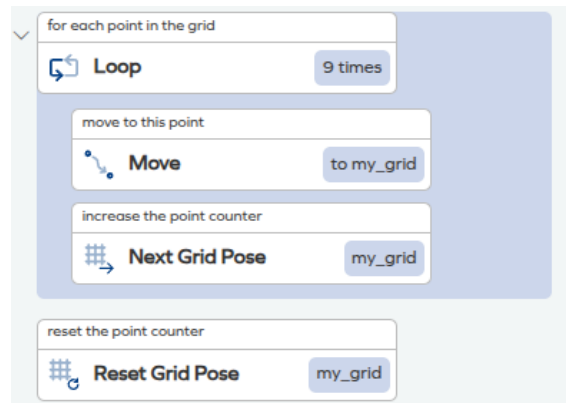


圖 4.7.7.7：使用網格模塊的範例

4.7.7.8 重置網格位姿模塊

「**重置網格位姿**」區塊會將特定網格的內部位姿計數器重置為零（初始值）。需先選擇受影響的網格。

4.7.7.9 設置變數方塊

「**設置變量**」模塊用於設置預先定義的程序變量值。必須先選擇一個變數，再選擇一個數值。允許使用數值和/或邏輯運算式。

4.7.7.10 設置輸出區塊

「**設置輸出**」模塊用於設置數字輸出或類比輸出的數值。使用者需先選擇要設置的輸出，並根據輸出類型（數字輸出或類比輸出）分別設置布林值或數值。

4.7.7.11 等待區塊

「**等待**」區塊會阻斷程式流程，直到發生特定事件為止。有三種不同的等待模式可供選擇：

- **時間** → 等待特定時間過去（允許使用數值表達式）
- **運動** → 等待運行中的移動指令停止
- **輸入** → 等待數字輸入的數值等於特定數值，或類比輸入的數值超過特定閾值

4.7.7.12 循環區塊

「**循環**」模塊會創建一個會被反復呼叫的模塊群組。共有三種不同的循環模式可供選擇：

- **無限循環** → 永遠循環
- **計數** → 循環特定次數（允許使用數值表達式）
- **條件** → 循環直至滿足特定條件（允許使用邏輯運算式）

使用者可將「**退出，當滿足條件時**」模塊加入循環中，詳見下一個節。

4.7.7.13 退出，當模塊

「**退出，當條件成立**」模塊與之對應的「**循環**」模塊綁定，若指定條件成立（允許使用邏輯運算式），則會跳出此循環。

4.7.7.14 條件模塊

「**條件**」區塊會創建一個容器，內部包含兩個群組：「if」和「else」。條件（允許使用邏輯運算式）用於將程式流程分支至「if」群組（若條件成立），或分支至「else」群組（若條件不成立）。

4.7.7.15 注釋區塊

注釋模塊僅是將說明性注釋加入程式流程中。請注意，其他每個模塊也都具備添加注釋的功能，這些注釋會顯示在程式區域中。不過，有時創建一個不與特定模塊綁定的獨立注釋會更有意義。

4.7.7.16 退出模塊

「**退出**」區塊會停止目前正在執行中的程式。剩餘的程式模塊將不會被執行。

4.7.7.17 對話框模塊

「**對話框**」區塊會創建一個包含多個群組的容器。執行時，程式流程會中斷並彈出對話框。使用者必須回應對話框，程式流程才會繼續。區塊內的群組（默認值為「確定」和「取消」）會對應到對話框中的按鈕。根據按下的按鈕，程式流程將分支並繼續執行所選的群組。對話訊息（允許變數注入）、按鈕名稱及按鈕數量皆可在區塊詳細設置中進行修改。

4.7.7.18 打印區塊

「**打印**」模塊會將自訂訊息打印到程式輸出中。

4.7.7.19 日誌區塊

「**日誌**」模塊會將自訂訊息記錄至自訂日誌文件中。日誌文件可透過桌面下載，例如使用快捷鍵 CTRL + SHIFT + L。

4.7.7.20 通知區塊

「**通知**」區塊會將自訂通知訊息發送至 GUI。此通知不會阻斷程式流程。

4.7.7.21 警告區塊

警告區塊會將自訂警告訊息發送至 GUI。此警告不會阻斷程式流程。

4.7.7.22 錯誤區塊

「**錯誤**」區塊會將自訂錯誤訊息發送至 GUI，並停止目前正在執行中的程式。

4.7.7.23 Python 代碼區塊

Python 代碼區塊是一種高級區塊，允許使用者呼叫幾行獨立的 Python 代碼。您可能需要使用此區塊來呼叫那些透過塊程式設計無法直接使用的函數。其中可使用完整的應用程式函數集。

4.8 I/O 菜單

按下主視窗上的 **I/O** 按鈕即可開啟 I/O 選單。此選單會顯示所有數位與類比 I/O，以及先前在 I/O 配置視窗中定義的屬性(包含設定、指派的功能與自訂名稱)，同時也會顯示所有 I/O 的當前值。輸入端為唯讀，因此這些值僅供在此處視覺化檢視。然而，輸出端則可透過此視窗進行設定。

數字 I/O 代表布林值，其中 **True**、**1** 或「綠色」表示「通電」，**False**、**0** 或「灰色」則表示「未通電」。數字輸入會以圓弧形指示器顯示此數值，而數字輸出則可透過按下方形指示器按鈕來變更其狀態。

類比 I/O 以 0.1 為單位的浮點數表示。類比 I/O 的單位由該 I/O 的設定所定義（V 代表電壓，mA 代表電流）。類比輸入以可觀察的數值呈現，而類比輸出則可透過該輸出當前值旁的文字欄位進行調整。

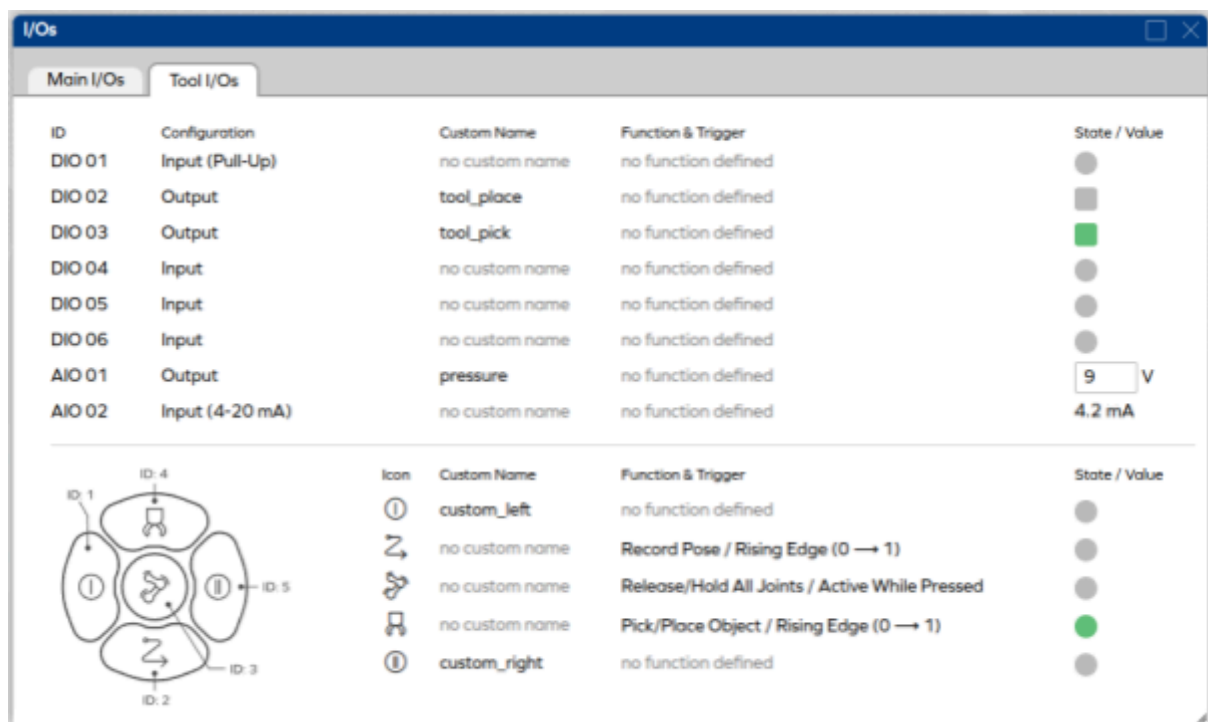


圖 4.8：I/O 菜單

4.9 可視化菜單

按下主視窗上的「**可視化模型**」按鈕即可開啟可視化模型視窗。此視窗僅顯示機器人的當前位姿，這在遠程控制機器人且無直接視覺回饋時特別有用。您還可以在 3D 可視化模型中查看先前儲存的位姿。可視化模型功能具有互動性，並提供以下幾項控制選項：

- 在可視化模型視窗內點擊滑鼠（左鍵）並拖曳，即可**旋轉**影像，並可左右及上下移動影像。
- 分別向上和向下在滑鼠滾輪上轉動，即可放大或縮小**畫面**。
- 使用鍵盤上的方向鍵，或以滑鼠按右鍵並左右拖曳，即可將攝影機左右及上下**移動**。

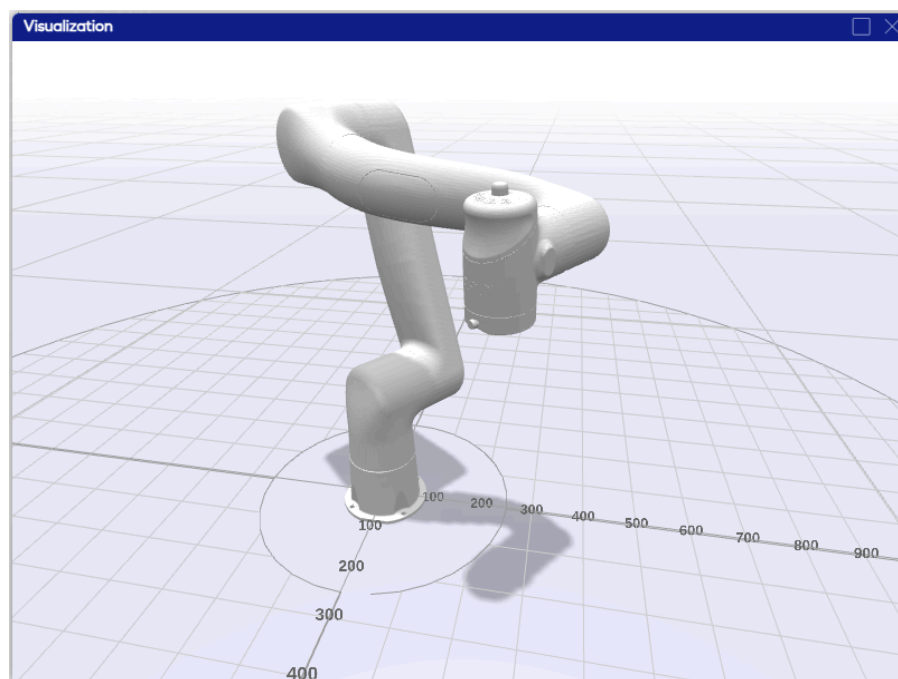


圖 4.9：可視化模型視窗

4.10 使能控制

按下主視窗上的「手導控制按鈕」，即可開啟手導控制菜單。此菜單允許使用者指定在 HGC 模式下，哪些機器人關節可進行移動。默認情況下，按下機器人的 HGC 按鈕（以及駕駛艙中的相關按鈕）將允許使用者移動所有關節。透過點擊此菜單中的個別關節指示器，使用者可啟用或停用特定關節，使其不受 HGC 影響。

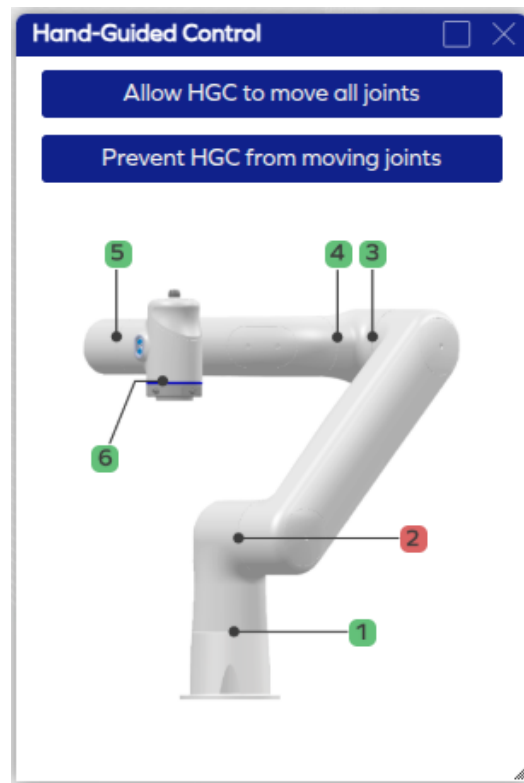


圖 4.10：使能控制菜單

4.11 系統監視器

按下主視窗上的「**系統監視器**」按鈕，即可進入系統監視器菜單。此菜單包含以下供使用者觀察的系統輸出：

4.11.1 機械臂狀態

「**機械臂狀態**」標籤頁顯示機械臂的當前狀態。這包括工具中心點 (TCP) 相對於機械臂基坐標系的坐標，以及所有馬達的位置、電流和溫度值。此外，所有安全輸入的狀態亦可在此處查看。

4.11.2 程序輸出

每當應用程式透過 **print(...)** 函式傳回文本時，該文本便會顯示於此視窗的「**程序輸出**」分頁中。請注意，若有多個應用程式並行執行，所有應用程式都可能將文本輸出至此處。

4.11.3 程序日誌

每當應用程式使用 **log_message(...)** 函式記錄資料時，該資料便會顯示於此視窗的「**程序日誌**」分頁中。此處可對記錄的資料進行篩選與搜尋。若您使用鍵盤快捷鍵 **CTRL + SHIFT + L**，這些自訂日誌將與內建日誌檔案一併下載。

4.11.4 歷史訊息

每當介面中彈出消息時，該訊息也會列於此視窗的「**歷史消息**」分頁中。這些訊息分為四種類型，適用於不同的情境：



信息 訊息會向使用者提供有用的訊息、說明性訊息或提示。



成功 訊息提供回報，讓使用者知悉某個指令或動作已執行成功。



警告 訊息會向使用者提供重要信息，在繼續操作前必須仔細閱讀。警告訊息從不屬關鍵性錯誤，也不會中斷程式流程。



錯誤類型 涵蓋範圍廣泛，從輕微的使用性錯誤（例如錯誤的執行或功能參數）到因機器人損壞等因素所導致的嚴重錯誤。

4.12 幫助菜單

「**幫助**」菜單包含有關目前執行中的系統和模塊、可供使用者使用的快捷鍵，以及在有疑問或問題時可聯繫的聯繫方式等有用的信息。

4.12.1 機器人信息

「幫助」選單中的「**機器人信息**」分頁包含關於 AX Portal 環境及已連接硬體元件（若已連接）的詳細信息。在機器人信息的「**模塊**」區段中，您將看到已安裝模塊的列表，以及它們是否已正確加載。只有當狀態顯示為「OK」時，您才能使用該模塊提供的功能和菜單。若需技術支持，請務必提供 AX Portal 及已連接硬體的詳細機器人信息。您只需在「複製到剪貼板」按鈕上點擊，並將資訊（建議使用等寬字型）貼上至您的支援請求中即可。

4.12.2 快捷鍵

「**快捷鍵**」分頁包含可用快捷鍵的概覽。這些快捷鍵分為兩類：常規鍵盤快捷鍵（例如用於操作控制台中的控制按鈕）以及應用程式編輯器快捷方式（用於編寫 AX Portal 應用程式）。請注意，macOS 的鍵盤布局與其他作業系統（Windows、Linux）有顯著差異，因此快捷鍵列表也會有所不同。

4.12.3 代碼文檔

「**代碼文檔**」分頁包含所有可用於程式設計的函數文件。這些文件分為兩類：始終可用的核心函數，以及與特定已安裝附加模塊相關的新增函數。透過分頁標題欄中的「**下載**」按鈕，可以下載代碼文檔，格式為 ZIP 文件。若要在離線狀態下查看代碼文檔，請在瀏覽器中打開其中的「index.html」文件。

4.12.4 開源許可證

「**開源許可證**」分頁包含有關 AX Portal 所使用之開源函式庫與模塊的所有必要信息。

4.12.5 聯繫方式

「**聯繫方式**」分頁顯示技術支持聯繫方式。在提交技術支持請求前，請務必先參閱手冊。提交技術支持請求時，請務必下載日誌文件並隨請求一起發送。

4.13 設置菜單

在「設置」菜單中，使用者可永久更改 AX Portal 的部分基本設置。按下「保存」以接受所有更改，按下「取消」以拒絕更改，或按下「重置」將所有設置重置為最近儲存的值。以下將說明所有設置類別與選項：

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

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

Buttons at the bottom: Cancel, Reset, Save.

圖 4.13：設置菜單

4.13.1 介面設置

此類別包含數項與使用者介面相關的設置。

實時自動補全	啟用此選項可以在應用程式腳本編輯器中鍵入代碼時實時完成函數的自動補全；如果禁用，您仍然可以使用快捷鍵 CTRL+SPACE 進行自動補全
顯示可用函數參數	在代碼中插入函數片段時，選擇是否顯示或者隱藏函數參數
訊息顯示級別	選擇訊息顯示的種類若希望所有訊息都消失，請盡可能選擇高的級別；若希望所有訊息都消失，請盡可能選擇低的級別。
訊息顯示時間	指定訊息的顯示時間（01 - 60 秒）

4.13.2 自定義導航面板按鈕

這些設置僅允許使用者啟用或禁用圖形化使用者介面中駕駛艙所顯示的控制按鈕，適用於桌面界面及面板介面。這些按鈕的詳細功能說明請參閱「控制台」章節。若使用者不願或無法使用某個按鈕（例如：若未連接互動式工具，則無法使用夾具按鈕），或是使用者不希望操作人員透過面板介面使用該按鈕，此時禁用該按鈕會比較方便。

顯示 * 按鈕	啟用以顯示此按鈕，禁用以隱藏它。
回零位置	指定回零位置的角度，此位置與駕駛艙中可用的「回零按鈕」相關聯。請為每個關節角度輸入一個數值。

4.13.3 啟動設置

在此類別中，所有設定皆與 AX Portal 的啟動程式相關。修改這些設定有助於避免啟動時重複執行相同操作。針對某些啟動設定，請選擇列表項目「**上一次選擇的項目**」（默認選項），以記憶並重新載入其上一次的會話。

啟動項目	選擇啟動時要加載的項目。所有已定義的項目都會出現在列表中。
啟動程序	若已選取項目，請選擇啟動時要加載的程式。該項目中所有已定義的主程式都會顯示在列表中。
要觸發的事件	若希望機器人在開機時自動上電，甚至執行特定程式，請在此進行選擇。默認情況下，此選項為禁用狀態。為防止使用者誤觸發非預期之程式，在選擇觸發激活條件前，必須先選取上述部分開機設定。

4.13.4 運動行為

在此類別中，所有設定皆涉及機器人移動時的行為。

中斷狀態行為	選擇中斷時的運動行為。有關不同中斷狀態行為及如何解決中斷的詳細信息，請參閱以下子章節「中斷狀態行為」和「中斷對話框」。
碰撞後的恢復嘗試次數與超時	設置發生碰撞後可嘗試繼續運行的最大次數。若在設置的超時期間內發生過多碰撞，機器人將停止繼續運行。
溫度速度降低	選擇在驅動器溫度過高時是否自動降低機器人速度。此設置默認啟用，當溫度超過 70 °C 時，動作會自動減速以防止過熱。若檢測到溫度

超過 90 °C，系統將觸發嚴重錯誤，以避免因過熱導致的損毀。有關此功能的更多詳細資訊，請參閱《六軸協作機器人 AX6: 手冊》中的「最大性能限制」章節。

4.13.4.1 中斷狀態行為

中斷可能因多種原因發生，例如：

- 觸發了 SS1 停止事件
- 觸發 SS2 停止事件
- 機器人與人員、物體或環境發生碰撞
- 觸發了安全防護中斷

當機器人關節的電流偏離機器人的動態模型，且超出定義的閾值時，即會檢測到碰撞。碰撞檢測是一項安全功能，提供額外的保護層；切勿刻意將其作為應用程式的組成部分，因為反覆碰撞會對機器人關節造成額外的壓力，並加速磨損。

請注意，為確保碰撞偵測正常運作，程式設計師在拾取或放置物體時，必須將機器人的負載調整至正確值。

除了物理碰撞外，亦可將光柵等外部安全裝置或類似安全防護設備連接至機器人。這些觸發機制在發生物理接觸前即發揮預防作用，但系統會以與碰撞事件相同的方式處理。中斷狀態行為定義了系統如何處理由碰撞及安全防護觸發的中斷。

目前，中斷狀態行為的建議流程如下：

暫停&等待指令

會使正在運行的程式暫停，並等待使用者透過對話框批准是否繼續或停止程式。

4.13.4.2 中斷對話框

當機器人運作期間發生中斷狀況時，系統會顯示中斷對話框。典型的中斷來源包括偵測到的碰撞、急停啟動、安全防護觸發，或其他與安全相關的事件。此對話框將引導使用者以安全且受控的方式解決該條件。

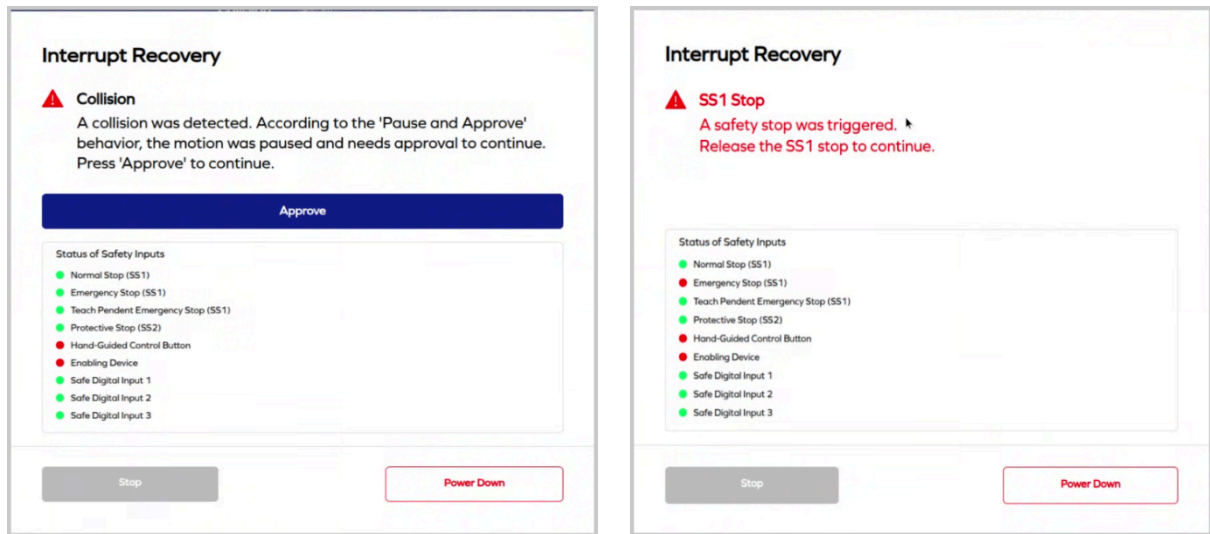


圖 4.13.4.2：中斷恢復

此對話框分為兩個主要區塊：

上部區域 – 中斷信息與解決方法：此區域說明當前偵測到的中斷條件，並提供明確的解決步驟。範例包括：

- 碰撞後清除機器人的路徑。
- 解除急停或其他安全裝置。
- 在必要時重置安全條件。

根據中斷類型，畫面會顯示「**批准**」、「**繼續**」或「**重置**」等操作按鈕。使用者必須依照指示，按下指定的按鈕才能繼續操作。

下半部 – 安全輸入狀態：對話框下半部顯示所有相關安全輸入的實時狀態。這讓使用者能快速辨識目前哪個安全裝置處於活躍狀態，例如：

- 急停（包含示教器緊急停止）
- 保護性停止
- 使能按鈕
- 手導控制按鈕
- 安全數字輸入

綠色指示燈表示正常狀態，紅色指示燈則表示處於活動或阻塞條件（或按鈕未被按下）。

停止選項

當使用者需要在繼續執行前一項程式或停止之間做出選擇時，即可使用「**停止**」按鈕。按下此按鈕將停止機器人並使其返回「就緒」狀態，此時機器人已準備好執行新動作。

休眠選項

「休眠」按鈕在「中斷對話框」中始終可用。它允許隨時關閉機器人電源。請注意，休眠**並不會**自動解除中斷條件；在重新上電後繼續運行之前，可能仍需排除根本原因（例如：急停按鈕被按下）。

典型工作流程

1. 發生中斷時，對話框會自動顯示。
2. 請閱讀上方的偵測條件描述。
3. 檢查下方的安全輸入狀態，以識別當前的觸發原因。
4. 實際排除該條件（例如：釋放急停按鈕、移除障礙物）。
5. 按下指示的操作按鈕（**批准**、**繼續**或**重置**）以繼續。
6. 如有需要，請重複此動作，直到機器人繼續運行為止。

SLP 違規

中斷恢復的一個特殊情況是 SLP（安全限位）違規，當機器人嘗試移動到安全設置中指定的位置限制之外時，就會發生這種情況。如果發生這種情況，中斷對話框會引導使用者切換到手動操作模式，在該模式下，使用者將能夠透過精細控制方式移出位置限制。除了精細控制之外，當機器人處於 SLP 違規狀態時，不允許進行任何移動。若機器人已休眠，或啟動時存在有效的 SLP 違規狀態，請執行以下操作：

1. 切換到手動模式
2. 重新啟動機器人
3. 進入精細控制菜單
4. 以精細控制方式將機器人移出其位置限制範圍

4.14 安全設置菜單

「安全設置菜單」是 AX 系列機器人安全系統的前端。此畫面專為**安全主管**設計。其他使用者僅能讀取及查看安全參數，但不得進行變更。

4.14.1 使用者介面

進入**安全設置菜單**後，此菜單將展示機器人安全系統中目前設置的所有安全參數。所有使用者皆可查看這些設置，但僅有**安全主管**可進行更改。此菜單分為多個子菜單，您可透過視窗底部的頁碼或前後導航按鈕進行導航。

出口	按下「 匯出 」按鈕，安全設置將以 *.yaml 檔案格式匯出。此檔案可用於備份安全設置，亦可匯入至其他機器人以複製安全設置。
編輯	安全主管 可點擊「 編輯 」按鈕（並輸入 用戶密碼 ）進入編輯模式。

進入編輯模式後，所有子選單的設置皆可編輯。安全主管現在可根據機器人的安裝配置及使用情境，更改安全限制。為便於辨識哪些安全設置已更改，更改過的欄位會以藍色標示。

進口	按下「 匯入 」按鈕，即可載入先前匯出的安全設定檔。此操作將覆寫所有子選單中手動變更的值。
申請	按下「 套用 」按鈕（並輸入 使用者密碼 ）後，所有安全設置將發送至安全系統，由其驗證標記與限制值。若驗證失敗，菜單中將以紅色標示無效項目，以便您輕鬆辨識。若驗證成功，菜單將切換至驗證模式。請注意，此時設置值雖已儲存，但尚未經過驗證。驗證設置是啟動機器人的必要步驟。
取消	按下「 取消 」按鈕後，安全設置將不予套用，所有變更均被拒絕。菜單隨後將切換回（默認）觀察模式。

當安全主管完成安全設置的編輯，且所有值均已成功套用後，此菜單即進入驗證模式。先前已套用但尚未驗證的設置會以黃色標示。

驗證	按下「 驗證頁面 」按鈕，安全主管將正式將當前可見子菜單的設置標記為已接受，並選擇下一個子菜單。按下最後一個子菜單的「 完成驗證 」按鈕，即是儲存設置的最後一步。
-----------	---

驗證新數值後，必須驗證安全系統的運作行為。可透過在「安全驗證模式」下上電來執行此操作，此模式允許軟體在驗證過程中忽略所有安全限制，以測試安全系統是否如預期觸發。

Safety Settings

Safely Limited Position (SLP)

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

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

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

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

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

Tool Safety Zone

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

Tool Safety Offset mm

Tool Safety Radius mm

Cancel

< 1 2 3 4 5 6 >

Import

Apply

圖 4.14.1：編輯設置時的安全設置菜單

4.14.2 安全設置

大部分的安全設置（橫跨各子選單）分為 4 欄。這些欄位定義了機器人的**安全案例**。當對應的**安全數字輸入**（SDI 1-3）被觸發時，即會觸發該安全案例。無論安全數字輸入的狀態如何，**默認**參數始終處於有效狀態。

安全數字輸出（SDO）無法進行配置，其電氣規格詳見《機器人控制器 RC-A101 手冊》。

部分設置具有硬編碼的安全裕度。此值指定了軟體可在安全系統觸發之前提前觸發的可接受範圍，以防止發生安全違規。

4.14.2.1 安全限位 (SLP)

安全設置菜單的第一個子菜單包含「安全限制位置」。在此處，可針對機器人的 x、y 和 z 方向進行位置限制，並以機器人的基坐標系表示。若坐標模式不明確，點擊資訊標示將顯示帶有坐標的圖像。您可以限制機器人所有部件（包括底座和連桿）的位置，或者僅限制機器人工具中心點 (TCP) 的位置。

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

圖 4.14.2.1：SLP 設置範例

4.14.2.2 安全極限關節角度

安全設置菜單的第二個子菜單包含「安全極限關節角度」。默認狀態下，「安全極限關節角度」功能禁用。啟用安全模式後，即可監控關節的角位置。

然而，在安全模式下，關節角度的監控範圍僅限於 $[-180^\circ, 180^\circ]$ 之間。因此，相較於機器人的實際物理極限，其運作範圍可能會受到限制。為避免此限制，可分別禁用第一關節、第四關節及第六關節的監控功能。

「安全極限關節角度」功能僅在同時定義上下限時才生效。若兩者皆未定義，則該關節的監控功能將被禁用。

Safety Settings

Safely Limited Joint Angles

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

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

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

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

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

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

圖 4.14.2.2：安全極限關節角度設置的範例

4.14.2.3 安全限速 (SLS)

安全設置菜單的第三個子菜單包含安全限速功能。在此處，可分別限制機器人各關節的速度，以及機器人所有部件（包括工具、底座和連桿）的運動速度。

	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

圖 4.14.2.3：SLS 設置範例

4.14.2.4 安全功率與力限制

安全設置菜單的第四個子菜單包含與功率和力量相關設置：

- 針對每種安全情境，使用者可定義不同的力限制，用以決定機器人對外部力量（例如與人員或物體碰撞）的敏感程度。力限制將應用於工具中心點（TCP）。
- 設置平均負載（以及運作期間的適當負載值）對於提升力限制對特定應用場景的敏感度至關重要，此處的負載定義為工具重量加上抓取的物品重量。
- 機器人的重力矢量必須與其安裝方式相符。若機器人安裝於地面，重力矢量為 $[0, 0, -9.81]$ 。然而，若機器人安裝於牆面、傾斜表面或天花板，則必須先計算重力矢量，並在此處正確輸入。

The screenshot shows the 'Safety Settings' window with the 'Safe Power and Force Limit (PFL)' section selected. The window has a title bar with a question mark icon, a maximize icon, and a close icon. The 'Safe Power and Force Limit (PFL)' section includes a descriptive text: '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.' Below this text is a table with four columns: 'Default', 'Safe Case 1', 'Safe Case 2', and 'Safe Case 3'. Each column has a 'Force Limit' input field set to '250' and a unit 'N'. The 'Average Payload' section includes a descriptive text: '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.' Below this text is an 'Average Payload' input field set to '0' with a unit 'kg'. The 'Gravity Vector' section includes a descriptive text: 'Defining the gravity vector in the robot's base frame based on its mounting orientation.' Below this text is a table with three columns: 'x', 'y', and 'z'. Each column has a 'Gravity Vector' input field. The 'x' and 'y' fields are set to '0', and the 'z' field is set to '-9.81'. The unit is 'm/s²'. At the bottom of the window, there is an 'Export' button, a navigation bar with arrows and numbers 1 through 6 (with 4 selected), an 'Edit' button, and a 'Start Verification' button.

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

Average Payload: 0 kg

	x	y	z	
Gravity Vector	0	0	-9.81	m/s²

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

圖 4.14.2.4：安全功率與力限制設置

4.14.2.5 安全 I/O 與操作人員限制

安全設置菜單的第五個子菜單用於配置與安全相關 I/O 行為及操作人員權限。使用者可定義保護停止裝置的處理方式、在手動模式下是否監測輸入信號、重置條件、模式切換權限，以及變更操作模式所需的密碼要求。

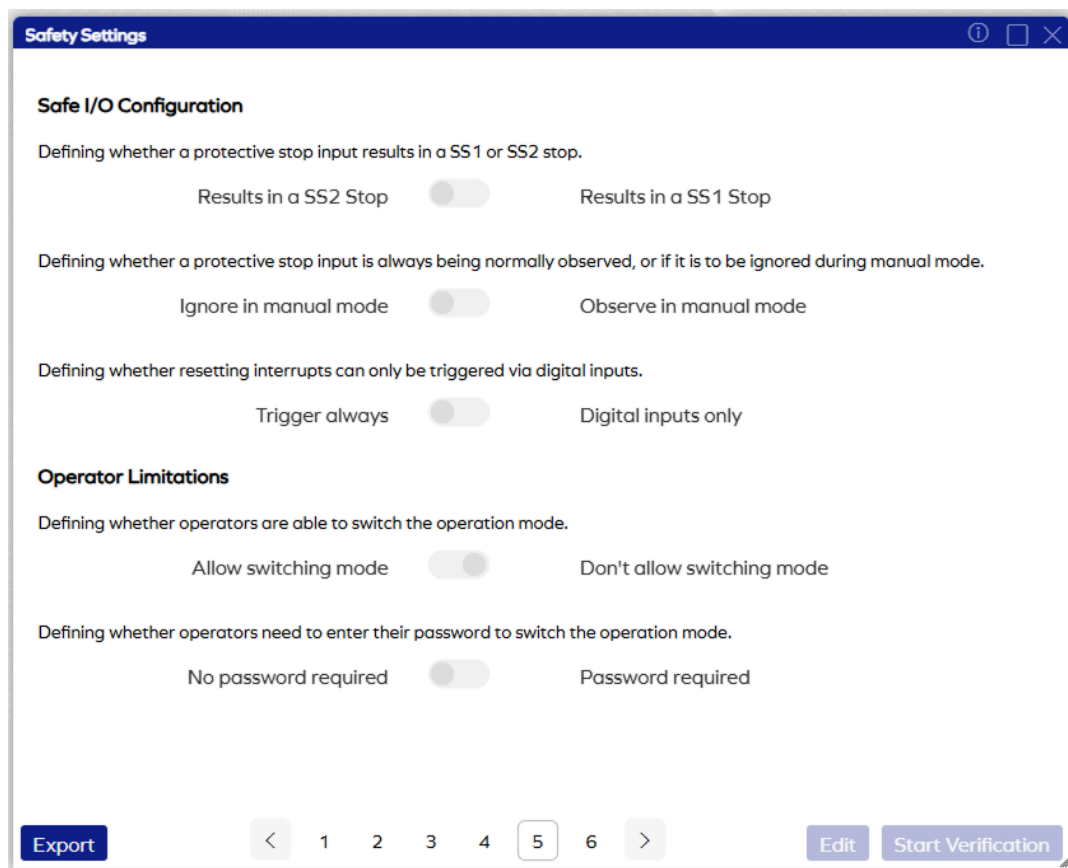


圖 4.14.2.5：安全 I/Os 配置設定

4.14.2.6 驗證資料

安全設置的最後一個子菜單包含有關安全裝置的資訊：

- 從安全裝置讀取的校驗和（用於故障排除）
- 傳輸計數器（記錄安全參數已被寫入的次數）
- 安全參數上次更新的時間
- 安全參數是否已透過驗證
- 安全裝置的序列號與固件版本

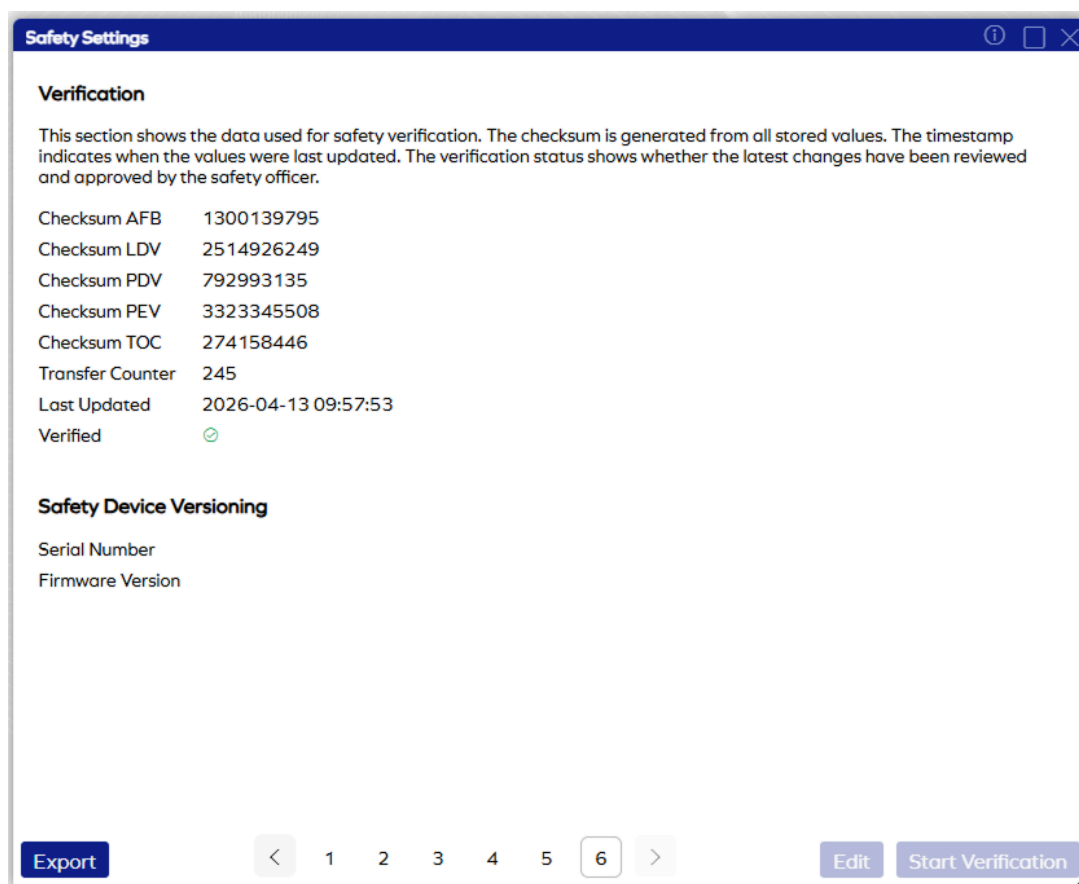


圖 4.14.2.6：驗證設置

4.14.2.7 安全輸出功能

特定安全功能的邏輯狀態與機器人控制器的安全輸出相連。詳細信息請參閱《機器人控制器 RC-A101 手冊》。

4.14.2.8 連接選配裝置

若要連接選配裝置（例如平板支架或使能按鈕），請參閱《機器人控制器 RC-A101 手冊》。

5. 面板介面 (GUI)

5.1 簡介

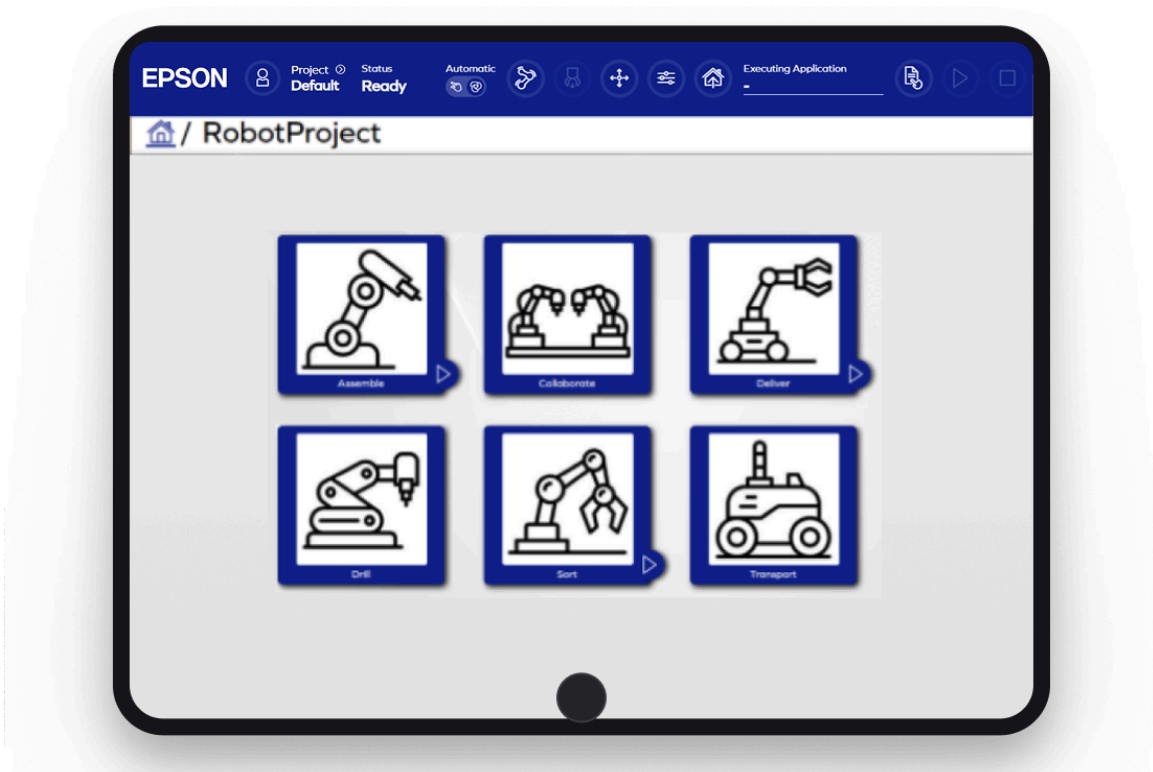


圖 5.1.a: 面板介面範例螢幕截圖

面板介面是替代桌面界面的一種方式，可讓您快速存取主要程式。此介面設計完全可適應客戶需求，特別是僅顯示程式（及其狀態資訊）。面板介面支援觸控螢幕，其設計類似於先進行動裝置中廣為人知的應用程式結構，即使對沒有技術背景的使用者而言也直觀易用。此外，使用面板介面時，使用者無需關心程式設計或應用程式背後的程式腳本。只需點擊相應的應用程式，該應用程式即會開始運行。可透過管理員面板登入後，點擊「導航到控制面板」按鈕來存取此介面。

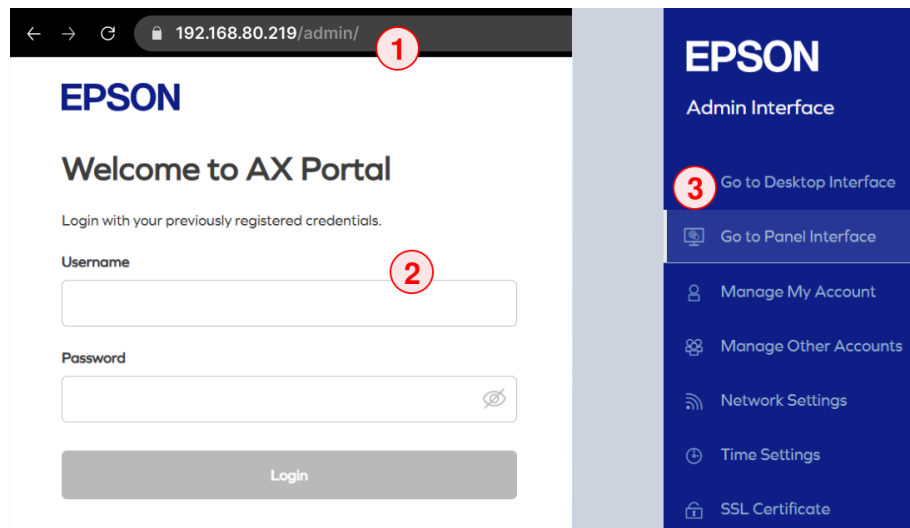


圖 5.1.b:存取面板介面

在某些瀏覽器（特別是在 Windows 系統上），顯示縮放比例默認值通常較高（例如 150%），這可能會導致介面部分內容（例如儀錶板或標誌）出現截斷現象。為獲得最佳使用體驗，我們建議將顯示縮放比例設置為 100%（或更低）。

5.2 配置面板介面的程式

默認情況下，面板介面不會顯示任何程式。使用者需先指定要在此處顯示的程式，以便操作人員快速執行。若要配置面板程式，請依照以下步驟操作：



圖 5.2.a: 程序編輯器中的面板介面相關設置（桌面）
與面板介面相關設置：2、3、4。

1. 導航到桌面，編輯程式，前往「程序設置」，
2. 啟用「**在項目面板中顯示**」標記，
3. 可選擇上傳適當**程式圖片**（最佳尺寸比例為 5:4），該圖片稍後將與程式名稱一同顯示於面板介面的主菜單中，
4. 可選擇導航至「應用程序變數」分頁以**新增狀態變量**，這可讓您透過面板介面監控和/或修改變數，
5. 最後**儲存**應用程式。

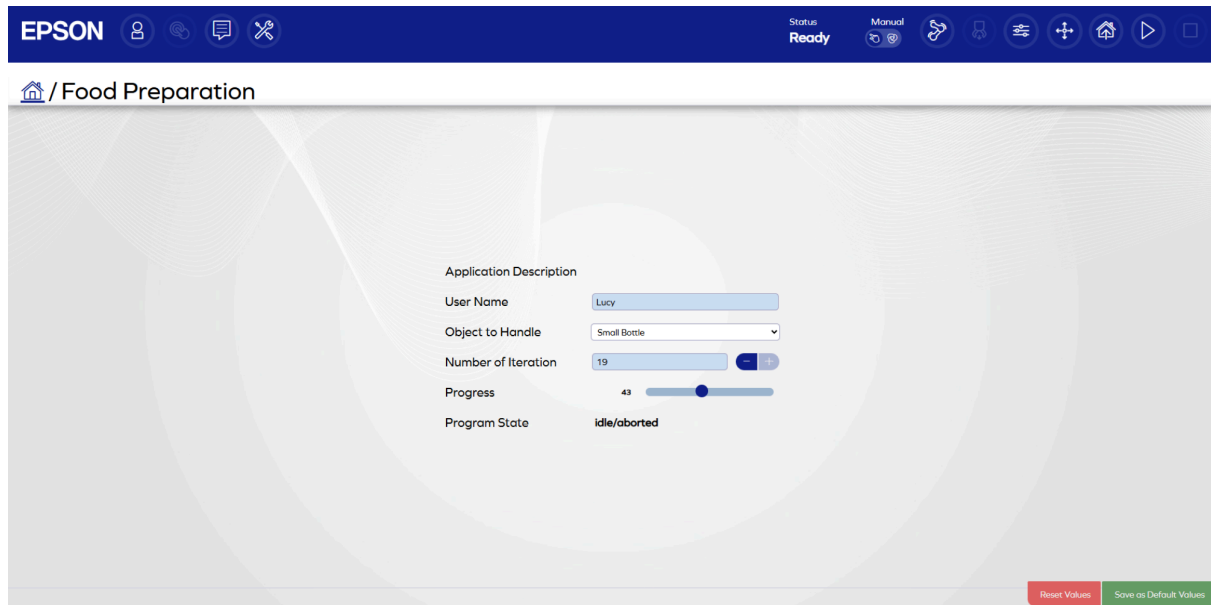


圖 5.2.b: 帶有狀態變數的面板程式

5.3 控制台與設置

面板介面的控制台顯示的按鈕與桌面界面的控制台大致相同。有關所有元素的詳細說明，請參閱「控制台」一節。除了這些元素之外，您會在左側看到一些不同的按鈕：



「**面板接口**」按鈕會開啟主面板頁面，顯示可供選取與運行的程式列表。



「**消息**」按鈕會開啟歷史訊息，所有過往訊息（包括警告與錯誤）皆會按時間順序顯示。



「**設置**」按鈕將引導您進入設置窗口，該窗口僅包含最關鍵的快速訪問設置，即

- 選擇已連接的工具、
- 上電與斷電，啟動與關閉機器人，
- 下載日誌文件。

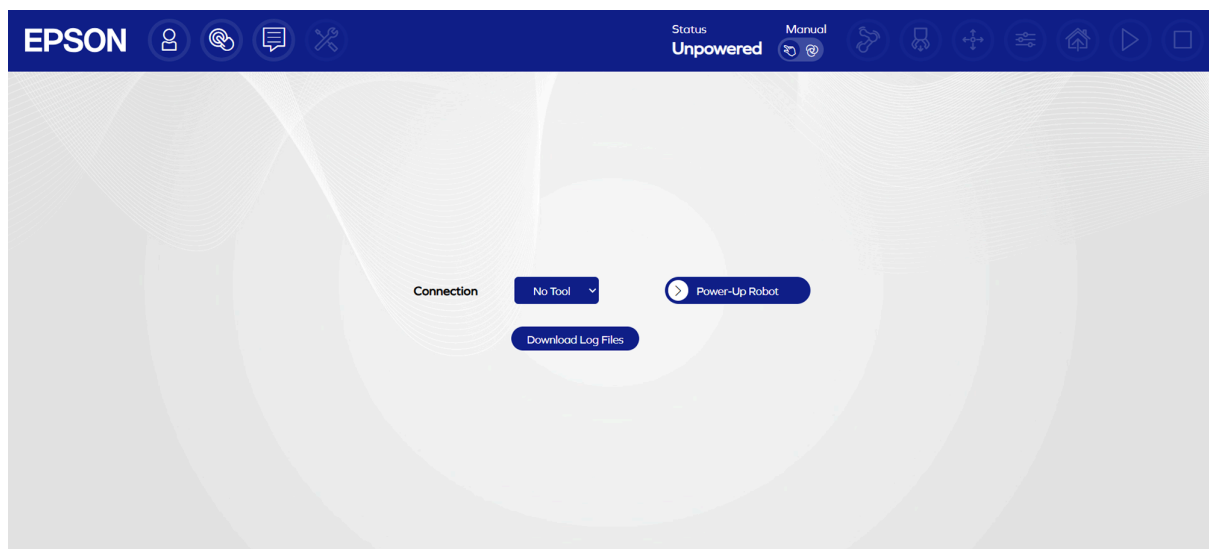


圖 5.3：面板設置視窗

5.4 程式控制

在面板介面的主視窗中，所有程式皆以類似程式列表的結構呈現，顯示其名稱與程式圖片。面板將程式區分為兩種類型：

- **不包含狀態變數**的程式會顯示一個播放按鍵。按下列式按鍵即可直接運行該程式。
- **包含狀態變數**的程式則不顯示播放按鍵。按下列式按鍵可進入程式詳細資訊視圖，顯示所有狀態變數。在該視窗中，您可以修改可編輯的狀態變數，並按下播放按鍵來執行該程式。按下返回按鍵可回到主面板視窗。

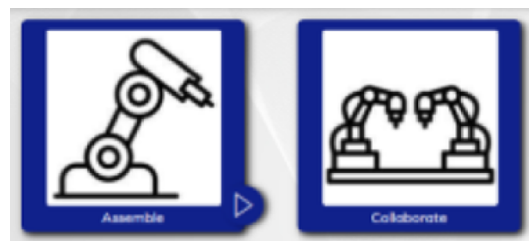


圖 5.4：程式按鈕

6. 運動學定義

機器人運動學描述了機器人的運動方式及其動作的表示方法。要開始使用運動學，使用者必須瞭解可用的運動學參數與物體的不同定義：

6.1 坐標

坐標定義了機器人的絕對位置與方向。位置是一個包含三個值的向量 `[x, y, z]`（單位為毫米），而方向同樣是一個包含三個值的向量 `[roll, pitch, yaw]`（航海角度，單位為°）。

初始化可依下列方式進行：

```
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})
```

存取元素可透過以下方式進行：

```
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 航海角

航海角是定義坐標中方向部分（X軸旋轉、Y軸旋轉、Z軸旋轉）的眾多方法之一。航海角常被用於飛機的方向表示。使用航海角時，三維旋轉定義為

- 1st 以負 z 軸為中心，沿 Z 軸旋轉 (ψ) 方向的旋轉，
- 2nd 以俯仰 (θ) 方向繞負 Y 軸旋轉，以及
- 3rd 繞正 X 軸在 X 軸旋轉 (ϕ) 方向上的旋轉。

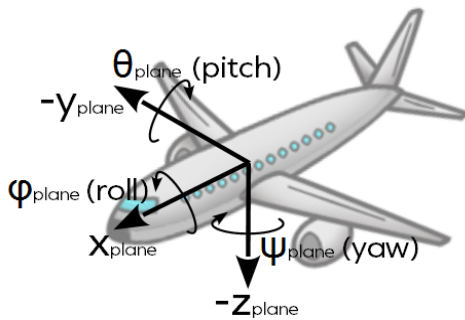


圖 6.1.1.a: 航海坐標系

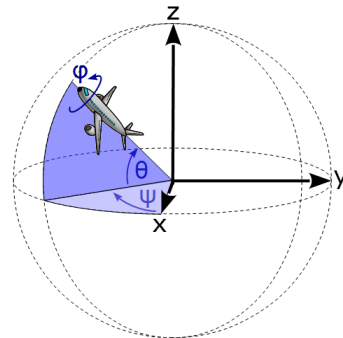


圖 6.1.1.b: 航海旋轉順序

提示：每當某個函數要求您處理末端執行器的三維方向時，不妨想像有一平面附著於末端執行器上，使得末端執行器與該平面的坐標系重合。如此一來，便能更輕鬆地推導出導致此特定方向的角度（X軸旋轉、Y軸旋轉與Z軸旋轉）。

6.1.2 坐標系

對於操作 AX 系列機器人而言，以下不同坐標系對使用者至關重要：

- 機器人的**基坐標系**定義如下：其 z 軸指向遠離底板的方向，x 軸則與底板平行，並指向遠離底板後面板接口的連接器方向。根據廣為人知的右手定則，即可確定 y 軸的方向。
- 機器人工件**坐標系**是相對於基坐標系由使用者定義的坐標系，除非另有說明，否則其與基坐標系相同。坐標將以此坐標系表示。使用者可利用此坐標系為機器人定義自訂工作空間。
- 機器人**法蘭坐標系**會因連接至機器人的工具類型而異：
 - 若無任何工具安裝，此座標系的原點位於機器人頂部法蘭（連接板）的中心。當所有關節角度皆為零時，其方向與基坐標系的方向一致。
 - 若安裝了自訂工具（並透過工具編輯器定義），則工具坐標系會根據該工具指定的位置與方向進行偏移。
- 機器人的**工具坐標系**是相對於法蘭坐標系的使用者定義坐標系，除非另有指定，否則等同於法蘭坐標系。坐標值將以該坐標系的原點為基準。使用者可利用此坐標系來定義工具中心點。

關於這些坐標系的範例，請參閱下圖：

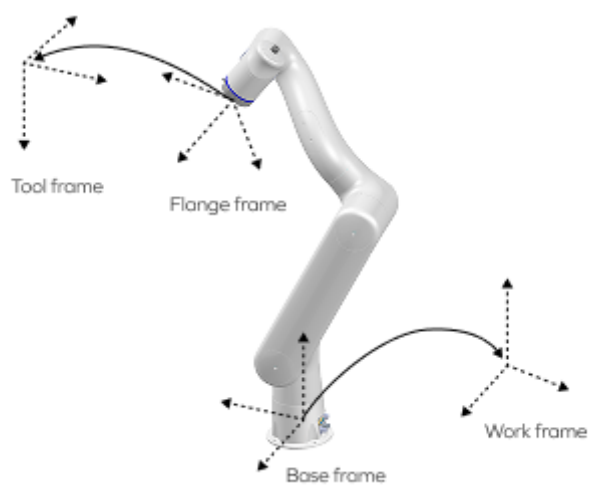


圖 6.1.2：機器人坐標模式

6.2 配置

針對給定的機器人位置與方向，逆運動學計算可能產生多種不同的關節配置。這些配置的差異在於關節的彎曲或旋轉方式。配置設置是用來從這些可能的選項中選擇一種特定的關節配置。對於典型的六軸機器人，即使末端執行器的位置和方向相同，肩部、肘部和腕部的不同方向組合仍可能產生多達八種不同的配置。為識別這些配置，它們被命名為「c1」至「c8」。

請注意，配置不適用於比較兩個位姿，或檢查路徑是否無縫。配置更改未必會導致路徑變得不無縫。此外，不無縫的路徑也未必包含配置更改。

6.3 位姿

位姿定義了機器人的坐標與配置。可透過資料庫中預定義位姿的 ID 或名稱來定義。此外，亦可透過指定坐標與配置並使用創建位姿函數，或透過指定關節角度並使用正向運動學函數來創建位姿。

初始化可依下列方式進行：

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

6.4 點

點是路徑的離散元素。除了機器人位姿外，它還指定機器人的速度以及自上一個點以來經過的時間間隔。這稱為 PVT（位置、速度、時間），通常儲存為關節角度向量、關節速度向量以及時間間隔浮點數。請注意，點的擴展版本也可能詳細說明機器人的加速度。

6.5 路徑

路徑定義了機器人如何在兩個或多個位姿之間移動。連續路徑僅在工具座標系中由多個線性或圓弧運動段定義。隨後，該路徑會被離散化並轉換至關節座標系。若路徑的離散解無需機器人重新定位（配置更改），則稱為無縫路徑；若沿路徑的加速度是連續的，則稱為連續路徑。

路徑始於指定的位姿（而非坐標）。路徑的各部分通常定義坐標，因為路徑的配置是由初始位姿決定的。透過使用逆運動學函數對每個前一離散點的最接近解，路徑將在可能的情況下自動成為無縫路徑。請注意，儘管從技術上講路徑並不包含速度資訊，但我們仍將「路徑」與「軌跡」互換使用。

初始化可依下列方式進行：

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

6.6 路徑段

線段代表路徑中的單一區段，用以定義傳輸控制點 (TCP) 如何朝目標運動。

可用的區段類型包括：

- **線性**：TCP 沿直線運動至該區段所定義的目標坐標。
- **圓弧運動**：TCP 從其當前位姿出發，沿著圓弧路徑經過定義的中間位置，最終抵達該區段的目標坐標。請注意，此運動過程中的 TCP 方向將由演算法自動決定。
- **坐標**：TCP 將使用最有效率的路徑移動至目標坐標。這相當於在關節空間中的直線運動。
- **位姿**：TCP 將沿最有效路徑移動至目標姿勢。這相當於關節空間中的直線運動。
- **方向**：TCP 執行純旋轉運動，在維持當前位置的同時達到指定的目標方向。

對於每個區段，可定義速度與加速度參數：

- 對於「方向」段，這些參數分別對應於角速度 [deg/s] 和角加速度 [deg/s²]
- 對於所有其他段落類型，則分別對應於 [mm/s] 單位的線速度與 [mm/s²] 單位的線性加速度

此外，可指定混合半徑，該參數定義了在過渡期間，當前節段起始位姿相對於前一節段終止位姿的最大允許偏差（單位為 mm）。若混合半徑設定為 0 mm，將強制 TCP 精確透過定義的位姿；而較高的值則允許路徑在節段之間彎曲，從而實現更平滑、更自然的過渡。

6.7 網格

網格是可自訂的、最多三維的點陣列，用於定義例如預設託盤的位置，並可用於彙整及儲存一組組的坐標值。通常，網格被視為二維陣列，但它也可以是獨立於第二維度的簡單列表（例如從輸送帶上拾取物體），甚至可以是三維結構（例如將物體堆疊在箱子中）。

7. 連接模塊

除了用於控制和監控機器人的直觀且易於使用的圖形化使用者介面（GUI）外，AX Portal 還提供了一組連接模組，可透過 TCP、REST、ROS 和 Modbus 等不同通訊方式來控制和監控機器人。這些通訊方式各自適用於不同的使用情境：

- **TCP** 非常適合需要快速且持續資料交換的即時通信。它通常用於直接控制程式、自訂客戶端實作，或與需要持久套接字連線的系統整合時。
- **REST** 最適合用於與基於網站的服務或程式進行整合。當易用性、可擴展性以及與標準 HTTP 系統的相容性比即時性能更為重要時，通常會採用此通訊方式。
- **ROS**（機器人作業系統）專為機器人研發設計。當需將機器人整合至複雜的機器人系統時，此協議特別實用，可實現感測器融合、自主模式，以及與其他支援 ROS 的裝置進行互動等功能。
- **Modbus** 常用於工業自動化環境。它非常適合與 PLC（可程式邏輯控制器）及其他工業設備進行通信，在這些情境中，可靠性與標準化至關重要。

本章將說明這些模塊的使用方法。

7.1 連接、驗證與活動控制

在使用任何連接模塊之前，您必須瞭解這些模塊如何與客戶端建立連接、驗證機制如何運作，以及 AX Portal 中活動控制的運作原理。

7.1.1 連接通道

裝置與機器人之間的連接總是透過特定的連接通道進行。不同連接模塊對這些連接通道的實現方式略有差異。

TCP 模組支援多重通道。這意味著每個欲連線至機器人的裝置都會建立專屬的連線通道，並分別進行登入。關閉其中任一通道將終止連線，並釋放活動控制權（詳見下文）。



警告

REST、ROS 和 Modbus 模組僅提供單一通道。這意味著一旦與其中一個模組建立並驗證連線，任何裝置皆可向機器人發送指令。這些通道是在 AX Portal 啟動時創建的，且無法由已連接的裝置關閉。

7.1.2 驗證

AX Portal 採用安全的兩步驟驗證機制。首先，請依照「安全驗證」章節所述的遠端登入程式，取得有效的存取憑證。此步驟必須透過**獨立的 TCP 連線**進行。成功登入並創建存取憑證後，即可透過選定的連線通道，使用 **token_login** 指令註冊此憑證以驗證使用者身分。



注意

若您在設置菜單中禁用透過 TCP 的驗證功能，則可跳過此驗證步驟。

請注意，Modbus 模塊不具備發送 **token_login** 指令的功能，因此必須在「設置菜單」中禁用此模塊的驗證功能，方能使用該模塊。

7.1.3 活動控制站

AX Portal 具備活動控制站保護功能，可防止其他通道同時控制機器人。若未啟用此保護功能，可能會導致危險情況。

連接通道會使用 **request_active_control** 指令來成為主動控制站。一旦此指令運行成功，該控制站將擁有永久控制權，直到通道關閉或機器人斷電為止。若另一控制站正在控制，此指令將被阻塞，直到該控制站接受控制請求為止。

指令 **'release_active_control'** 由當前控制站使用，用於釋放活動控制權。建議在通道不再被主動使用時，務必釋放控制權。否則，該通道將永久佔用控制權。

7.2 TCP連接模塊

7.2.1 建立連接

TCP連接模塊的伺服器端基於 Python 開發，並透過內建的 socket 函式庫實現。默認情況下，可透過 **38152** 號埠連接至伺服器。訊息採用 **UTF-8** 編碼標準進行編碼，並以**文本結束字元 0x03**（Python 中為“\x03”）進行分隔。請使用以下範例腳本連接至您的機器人，並列印所有收到的回應。關於傳入訊息的含義，將在後面的「接收指令」章節中進一步說明。

```
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 發送指令

在向機器人發送指令之前，請確保您已理解「連接、驗證與活動控制」章節中闡述的原則。以下範例將發送初始的 **token_login** 和 **request_active_control** 指令。

```
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)

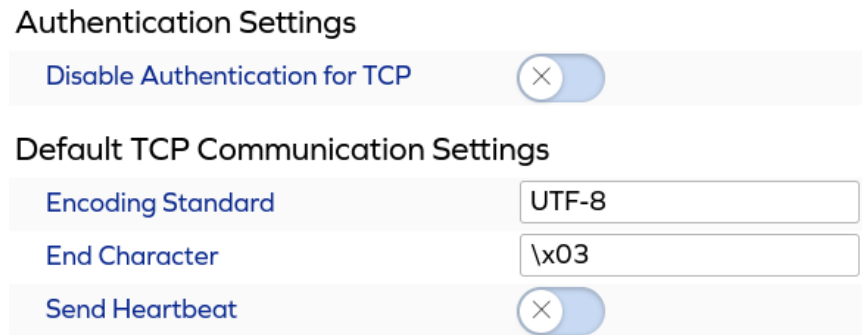
    # make sure to acquire active control, which makes the TCP connection the active
    # control station of the robot (only one control station can be active at any time)
    request = {"action": "request_active_control", "bridge": "core",
               "arguments": {}}
    message = json.dumps(request)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

    time.sleep(0.5)

# add more robot control commands here
```

7.2.3 設置

上述範例使用以下默認 TCP 設置，這些設置可在桌面介面的「設置」菜單中進行修改。在此處，您可以永久關閉 TCP 認證，並更改 TCP 訊息的編碼和結束字元。此外，您亦可啟用心跳功能，以持續檢查連接是否仍有效。



Authentication Settings

Disable Authentication for TCP ☒

Default TCP Communication Settings

Encoding Standard

End Character

Send Heartbeat ☒

圖 7.2.3：TCP 模塊設置

7.2.3.1 編碼標準

在此可定義透過 TCP 連接發送訊息的編碼標準。系統接受常見的 Python 編碼標準，常用標準包括默認的 UTF-8 或 ASCII。

7.2.3.2 結束字元

在此您可以變更透過 TCP 連線發送訊息的結束字元。結束字元標示訊息的結束。當收到訊息時，系統會自動從訊息中截取該結束字元。

7.2.3.3 發送心跳

在此您可以選擇是否讓 TCP 伺服器自動發送心跳訊號。心跳訊號將每 50 毫秒發送一次，內容為一則空訊息（僅發送結束字元）。

7.2.4 訊息協議 - 接收指令

所有發送和接收的指令皆為 JSON 字串。如前所述，訊息採用 **UTF-8** 編碼標準，並以**文本結束字元 0x03**（Python 中為“\x03”）進行分隔。接收到的指令是一個以 JSON 編碼的字典，其中包含一個鍵和一個值。該鍵定義訊息的類型，而值則包含訊息的詳細資訊。您可以在「建立連線」章節中找到從機器人接收這些訊息的程式腳本範例。

請注意，此清單並非完整列表，但包含最重要的指令。

7.2.4.1 狀態資訊

每當機器人狀態發生變更時，系統便會發送其狀態。

鍵	數值	類型
status_info	根據以下列表的當前機器人狀態： ● 0 → 未上電 ● 1 → 上電中 ● 2 → 加載中 ● 3 → 就緒 ● 4 → 處理中 ● 5 → 校準中 ● 6 → 運行中 ● 7 → 自主模式 ● 8 → 暫停 ● 9 → 繼續 ● 10 → 保護性停止 ● 11 → 安全防護 ● 12 → 位置限制 ● 13 → 使能控制 ● 14 → 碰撞 ● 15 → 急停 ● 16 → 正常停止 ● 17 → 錯誤 ● 18 → 斷電中 ● 19 → 停止中	整數

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

7.2.4.2 執行回傳值

執行動作後，系統會發送包含該動作指令的回傳值。若該動作沒有回傳值，則發送 `null`。只有在函數執行過程中未發生錯誤時，才視為成功。最後，序列號是一個可選參數，僅在使用者發送指令時指定了序列號時才會加入。

Key	數值	類型
<code>action_return_value</code>	回傳動作的名稱、回傳值以及動作是否成功。若發生錯誤，則表示動作未成功。	字典

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

7.2.4.3 錯誤

若發生錯誤，系統會發送包含更詳細資訊的錯誤訊息。

鍵	數值	類型
錯誤	錯誤資訊。	字典

```
{"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 歷史訊息

發送至使用者介面的過往訊息列表。

鍵	數值	類型
<code>message_history</code>	訊息包含文本、標題、代碼和時間戳記。	列表

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

7.2.4.5 狀態群組定義

定義機器人的位置資料。狀態群組包含一個 ID，用以指定所包含資料的內容。這些定義可用於識別狀態群組的資料。

鍵	數值	類型
state_group_define	狀態群組定義。	字典

```
{"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 狀態群組資料

狀態群組資料包含 TCP 或馬達數據。請使用「request_data」來接收此數據。

鍵	數值	類型
state_group_data	該值包含機器人的 TCP 數據（id = “pose”）、馬達數據（id = “robot”），或安全標誌（id = “safety”）。	字典

```
{"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 開啟對話框

當使用者介面中的對話框開啟時，系統會接收到「open_dialog」指令。隨後可透過「answer_dialog」動作對對話框進行回應。

鍵	數值	類型
open_dialog	此對話框可取以下數值： ● id → 唯一的對話框 ID（用於回應該對話框） ● kind ○ 0 → 確認對話框（包含確認/拒絕按鈕） ○ 1 → 輸入對話框（文本輸入） ○ 2 → 檔案對話框（檔案上傳） ○ 3 → 下拉式對話框（需提供選項列表） ○ 4 → 選項對話框（需提供選項列表） ● 默認 → 若未收到回覆時返回的數值 ● 顏色 ○ 0 → 資訊 ○ 1 → 成功 ○ 2 → 警告 ○ 3 → 錯誤 ● 標題 → 顯示的標題 ● 文本 → 顯示文本 ● 選項 → 下拉式選單/選擇對話框的選項列表	字典

```
{
  "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 腳本發送

接收由腳本發送的訊息。

鍵	數值	類型
script_send	由腳本發送的數值	多種

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

7.2.5 訊息協議 – 發送指令

所有傳送與接收的指令皆為 JSON 字串。如前所述，訊息採用 **UTF-8** 編碼標準，並以**文本結束字元** 0x03（Python 中為“\x03”）進行分隔。傳送的指令必須包含一個 JSON 編碼的字典，其中包含三個必填（及一個可選）鍵。您可在「傳送指令」章節中找到向機器人傳送訊息的程式腳本。

執行	定義要執行的指令	str
bridge	定義包含該指令的模組，通常為「core」	str
參數	定義命令參數，請在下方查看參數的詳細描述	字典
序列	<可選> 命令識別碼，將於結尾時在 action_result_value 中返回	整數

請注意，此列表並非完整，但包含了最重要的命令。

7.2.5.1 憑證登陸

為了使用有效的連接，第一個發送的命令必須始終是登陸呼叫。請遵循「安全驗證」章節中所述的遠程登陸程式，以取得有效的訪問憑證。

執行	token_login	
Bridge	核心	
參數金鑰	token	session_id
參數值	由登陸指令提供的個人化登陸憑證	由登陸指令提供的唯一會話 ID
參數類型	字串	字串

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

7.2.5.2 請求活動控制

取得活動控制權，使 TCP 連線成為機器人的活動控制站。請注意，此操作會向活動控制站發送請求，該請求可能被接受或拒絕。

執行	request_active_control
橋接器	core

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

7.2.5.3 釋放活動控制

釋放活動控制權，使該 TCP 連線不再是機器人的活動控制站。

執行	release_active_control
橋接器	core

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

7.2.5.4 請求數據

此命令可用作心跳檢測。可接收不同類型的數據。

執行	request_data
橋接器	core

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

7.2.5.5 上電

讓機器人啟動其馬達。

執行	power_up
橋接器	core
參數鍵	tool_id
參數值	ID of a customized tool , or 0 to auto-detect the tool
參數類型	Integer or String

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

7.2.5.6 休眠

終止連接，讓機器人關閉其馬達電源。

執行	power_down
橋接器	core

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

7.2.5.7 暫停

暫停所有正在執行的移動函式與腳本。

執行	pause
橋接器	core

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

7.2.5.8 繼續運行

若機器人處於暫停狀態，則繼續運行。

執行	resume
橋接器	core

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

7.2.5.9 使能手動引導控制 (HGC)

釋放機器人關節，使其進入重力補償模式並可手動運動。狀態需為“就緒”，並將切換至使能控制模式。

執行	release
橋接器	Core
參數鍵	joint_ids
參數值	<optional> The list of joint IDs for which to enable hand-guided control. Enable hand-guided control for all joints by default.
參數類型	List

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

7.2.5.10 保持使能控制 (HGC)

將機器人關節固定在位置控制模式下，使其無法再透過手動操作運動。狀態需設置為「使能控制」，並切換為「就緒」。

執行	hold
橋接器	core
參數鍵	joint_ids
參數值	The list of joint IDs to hold. Hold all joints by default.
參數類型	List

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

7.2.5.11 停止

中斷所有正在執行的移動函式和腳本。

執行	stop
橋接器	core

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

7.2.5.12 變更項目

將當前數據庫項目變更為具有特定 ID 的項目。該項目的狀態必須為“就緒”。

執行	change_project
橋接器	core
參數鍵	project_id
參數值	ID of the project to change to. The default project has ID 1.
參數類型	Integer

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

7.2.5.13 執行腳本

根據名稱或 ID 執行程式。狀態必須為「就緒」，並將切換為「運行中」。在腳本運行期間，機器人可以暫停。一旦腳本執行完畢，狀態將切換回「就緒」。

執行	execute_script	
橋接器	core	
參數鍵	script_name	script_id
參數值	要運行的腳本名稱。	要運行的腳本的 ID。
參數類型	String	Integer

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

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

7.2.5.14 測試腳本

執行程式的 Python 腳本。狀態需為「就緒」，並將切換為「運行中」。腳本執行期間，機器人可暫停。腳本執行完畢後，狀態將切換回「就緒」。

執行	test_script		
橋接器	core		
參數鍵	script_code	script_type	script_id
參數值	待測試應用程式的 Python 代碼。	腳本的類型： • “main” • “parallel” • “background”	待測試腳本的 ID。若該腳本不存在於資料庫中，ID 為 0。
參數類型	String	String	Integer

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

7.2.5.15 腳本接收

使用此功能可從您的連接通道向應用程式腳本發送訊息。該訊息可透過腳本的 **receive_message** 函數接收。

執行	script_receive
橋接器	core
參數鍵	訊息
參數值	腳本將接收的訊息。
參數類型	多種類型

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

7.2.5.16 運動位姿

將機器人移動至具有特定 ID 或特定名稱的位姿。狀態需為「就緒」，並將切換為「運行中」。機器人運動期間，其運動可被暫停。一旦運動完成，狀態將切換回「就緒」。

執行	move_pose
橋接器	core
參數鍵	pose_id
參數值	要移動至的位姿的 ID 或名稱。
參數類型	整數或字串

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

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

7.2.5.17 播放路徑

執行具有特定 ID 或特定名稱的路徑。狀態需為「就緒」，並將切換為「運行中」。機器人運動期間，其運動可被暫停。一旦運動完成，狀態將切換回「就緒」。

執行	播放路徑
橋接器	core
參數金鑰	路徑 ID
參數值	要執行的路徑的 ID 或名稱。
參數類型	整數或字串

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

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

7.2.5.18 切換工具

切換工具功能，通常為拾取與放置。狀態需為「就緒」，並將切換為「運行中」。一旦功能完成，狀態將切換回「就緒」。

執行	toggle_tool
橋接器	core
參數鍵	拾取
參數值	true 表示選取物體, false 表示放置物體。此行為由拾取行為所定義。
參數類型	布林值

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

7.2.5.19 寫入數字與類比輸出

寫入主控制器或工具控制器（從主控制器傳輸至工具的訊號線）的一個或多個輸出（**數位或類比**）。針對每種組合，會使用不同的函式（但參數相同）。所有 I/O 的列表可於 I/O 配置選單或 I/O 選單中找到。

執行	write_digital_main_output write_analog_main_output write_digital_tool_output 寫入_類比_工具_I/O	
橋接器	core	
參數鍵	識別碼	值
參數值	識別單一 I/O 的方法有很多，例如透過 ● 數字 (整數) ● 客戶名 (字串) 若要同時寫入多個 I/O，您可以選擇將編號或客戶名組合成一個列表。	根據您選擇的「識別碼」類型，數值可以是 ● 數位 I/O 的 true/false (布爾值) ● 數值 (浮點數) 適用於類比 I/O 或若需同時執行多個 I/O 操作，則為這些數值的列表。
參數類型	整數 / 字串 / 列表	布爾值 / 浮點數 / 列表

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

此範例會寫入數位工具輸出 3 並將其設為 true。

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

此範例寫入名為「my_analog_output」的類比工具輸出，並將其設定為 9.5V。

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

此範例寫入數字輸出 1 和 2，並將 1 設為 true，2 設為 false。

7.2.5.20 對話框回應

回應已開啟的對話框。對話框的回應取決於對話框的類型。

執行	answer_dialog	
橋接器	core	
參數鍵	對話框 ID	對話框_回答
參數值	要發送回覆的對話框之唯一識別碼。此 ID 屬傳入的 open_dialog 回應的一部分。	要發送給該對話框的回覆。
參數類型	字串	布爾值 / 整數 / 字串

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

7.2.5.21 設置狀態變數

將狀態變數「variable_name」的值設定為「variable_value」。該變數必須已存在於 ID 為「script_id」的腳本中，且該腳本需已儲存於資料庫中。此指令可在腳本運行中執行。

執行	set_state_variable		
橋接器	core		
參數鍵	script_id	變數名稱	變數值
參數值	包含待修改狀態變數的腳本 ID。	要修改的狀態變數名稱。	新值。
參數類型	整數	字串	各種

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

7.2.5.22 設置全域變數

將全域變數「name」的值設定為「value」。全域變數的儲存範圍超越腳本的執行時間，即使機器人重啟後，其值仍會保留。

執行	set_global_variable	
橋接器	core	
參數鍵	姓名	值
參數值	全域變數名稱	全域變數的值
參數類型	字串	多種

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

7.2.5.23 取得全域變數

取得全域變數「name」的值。全域變數會儲存至腳本執行週期結束之後，即使機器人重啟後仍會保留。

執行	get_global_variable	
橋接器	core	
參數鍵	名稱	
參數值	全域變數名稱	
參數類型	字串	

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

7.2.5.24 設置共用變數

將共用變數「name」的值設定為「value」。共用變數僅在腳本執行期間可用。腳本執行完畢或機器人停止運作後，這些變數將會被刪除。

執行	set_shared_variable	
橋接器	core	
參數鍵	名稱	值
參數值	共用變數的名稱	共用變數的值
參數類型	字串	各種

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

7.2.5.25 取得共用變數

取得共用變數「name」的值。共用變數僅在腳本執行期間可用。當腳本結束或機器人停止運作後，這些變數將會被刪除。

執行	get_shared_variable	
橋接器	core	
參數鍵	姓名	
參數值	共用變數的名稱	
參數類型	字串	

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

7.3 REST連接模塊

7.3.1 建立連接

REST連接模塊提供資源 **https://<IP地址>/rest/**網關，用於接收來自連接客戶端的指令。請注意，僅有一個連接通道。若有多名使用者透過 REST 連接，他們將共用此通道。

請使用以下範例腳本透過 REST 連線控制機器人，並開始向機器人發送指令。請注意，在發送指令之前，您必須已登入，且必須擁有活動控制權。因此，建議在不再使用該連線時，釋放控制權並登出。

```
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 發送指令

除了上述範例中提到的指令外，還有許多其他指令可用。請參閱 TCP 連接模塊中列出的指令列表，因為這些指令同樣適用於 REST 連接。

除了發送「bridge」、「執行」、「arguments」以及可選的「sequence」參數外，您也可以發送可選的「超時」參數。若發送超時參數，REST 請求即使成功發送，也不會產生默認回應 (200, {}), 而是該指令會阻塞，您將收到 (200, action_return_value) 的回應，其中「Action Return Value」包含以下資訊：呼叫了哪個 API 函數、呼叫的函數是否成功，以及該函數的實際回傳值。

7.3.3 設置

在桌面介面的「設置」菜單中，您可以找到一項用於關閉 REST 連接模塊認證的設置。關閉認證後，便無需進行登陸和登出操作。

Authentication Settings

Disable Authentication for REST



圖 7.3.3：REST 模塊設置

7.4 ROS 處理器模塊

ROS 處理器模塊可讓您將機器人整合至 ROS 2 環境(機器人作業系統), 並使用 ROS 2 指令與 AX Portal 進行通信。本章的說明適用於 ROS Jazzy 及 CycloneDDS 作為 ROS 中介軟件(RMW)。



警告

ROS 處理器提供對某些低階功能的存取權限。例如，您可以直接向機器人的馬達發送控制指令，這些指令將立即執行。在圖形化使用者介面中，某些功能可能需要額外的檢查或對話框確認，但此處可能無法提供這些機制。因此，當您透過 ROS 直接控制機器人時，請務必格外謹慎。

注意：

- 這是首個支援 ROS 2 的 AX Portal 版本，未來預計將推出更多功能。
- 目前，ROS 處理器尚不支援 AX Portal 模擬器，僅適用於實際機器人。

7.4.1 安裝

若要與 AX Portal 的 ROS 處理器進行通信，您的電腦上必須安裝 ROS。官方 ROS 文件提供了多種設定選項，例如 deb 套件、Docker 映像檔、原始碼安裝等。在此範例中，我們將使用官方 ROS Docker 映像檔來與機器人進行通信。以下 Docker Compose 檔案可在您的電腦上建立一個簡單的 ROS 環境：

```
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

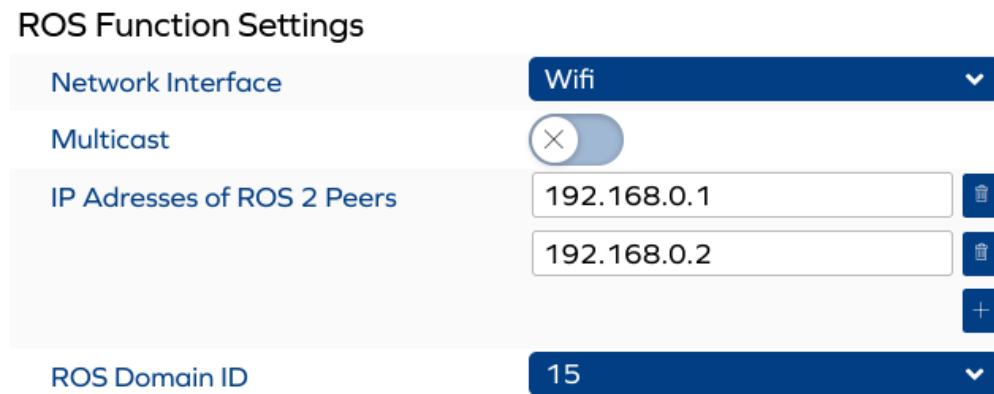
  network_mode: "host"

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

請注意，我們在啟動時安裝 `ros-jazzy-rmw-cyclonedds-cpp`，並將 `RMW_IMPLEMENTATION` 設定為 `rmw_cyclonedds_cpp`，以告知 ROS 使用 CycloneDDS，這與機器人上使用的 RMW 相同。此外，我們共用一個 `cyclonedds_config.xml` 檔案，並將 `CYCLONEDDS_URI` 設定為其位置，以建立自訂的 CycloneDDS 設定。

7.4.2 配置

機器人與您的電腦之間的 ROS 通信需要在兩台機器上進行調整。在您的筆記型電腦上，您需要設定一個新的環境變數並提供一個網路配置檔案。在機器人上，您無法直接存取這些元素，但您可以在設定選單的 ROS 區段中透過使用者介面進行編輯：



ROS Function Settings

Network Interface: Wifi

Multicast: ☐

IP Addresses of ROS 2 Peers: 192.168.0.1, 192.168.0.2

ROS Domain ID: 15

圖 7.4.2：ROS 處理器設置

在接下來的章節中，您將瞭解這些設置項目的具體含義，並學習電腦端需要進行哪些相應的調整。

7.4.3 使用 Cyclone DDS 進行網路探索

我們先前已多次提及 Cyclone DDS。它究竟是什麼？身為 AX Portal 使用者，您需要瞭解以下內容：

當選擇 Cyclone DDS 作為 ROS 中介軟體時，它將負責透過您的網路分發 ROS 相關數據包。其主要任務是建立裝置之間的網路連接，以便 ROS 2 能透過這些連接發送數據。

7.4.3.1 ROS 網路接口

首先，Cyclone DDS 需要知道 ROS 將使用哪個 ROS 網路接口。默認值為乙太網路。

請透過「ROS 功能設置」下的「ROS 網路接口」設置，選擇您所需的介面。

注意：當 AX Portal 啟動 ROS 處理程式時，Cyclone DDS 會掃描 ROS 網路接口以搜尋對等機器的 IP 地址。若無法存取此接口，ROS 處理程式將無法載入。因此，若您已將 ROS 網路接口設定為乙太網路，請務必在啟動機器人前確認網路線已連接妥當。

7.4.3.2 探索方法

Cyclone DDS 提供兩種方法來發現網路中的其他裝置：多播與點對點。對於多播，它會將每個網路封包發送給網路範圍內所有可達的客戶端。其優勢在於簡單：您無需費心定義哪些電

腦被允許相互通訊。在此設定中，`ROS_DOMAIN_ID` 環境變數（見下文）仍然可用，以防止兩台機器人混淆訊息。

缺點在於網路開銷：每個網路客戶端都會處理所有封包。即使設定不同的 `ROS_DOMAIN_ID` 也無法避免此情況。即使某台電腦未安裝 ROS：由於 Cyclone DDS 基於 UDP，網路中的所有電腦仍會接收並處理這些 UDP 封包。對於小型網路和少量數據而言，這通常不是問題。但若機器人需將攝影機串流發送至大型網路中的每個客戶端，這將顯著拖慢整個網路速度。在此情境下，更安全的做法是在 Cyclone DDS 配置中明確指定若干 IP 地址，允許這些地址透過點對點連接相互通信。

在接下來的兩個子章節中，您將分別學習如何設定網路以支援多播或點對點通信。提示：在大多數 Linux 發行版中，使用 `ip addr show` 指令可方便地查詢您的 IP 地址及對應的 ROS 網路接口名稱。

7.4.3.2.1 點對點

在上述安裝說明中，我們已在 Docker Compose 文件中掛載了一個 Cyclone DDS 配置文件。以下是一個 Cyclone DDS 配置範例，用於設置您的機器進行點對點通信：

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

首先，將 `INTERFACE_NAME` 替換為您電腦上實際用於 ROS 通信的 ROS 網路接口名稱。範例：`enp1s0`、`wlp2s0` 等。

下一個，調整「Peers」下的列表，使其反映您網路中那些您希望透過 ROS 相互連接的機器的 IP 地址。

與上述檔案類似，您的機器人上也儲存著一個相似的檔案。您無法直接存取該檔案，但可透過 AX Portal 設定選單中的「ROS 功能設置」區段來編輯其內容。該處的「ROS 2 對等節點的 IP 地址」設定提供了一張列表，您可在此填入 IP 地址。

請注意：

- 需要重啟機器人，變更設置才會生效。
- 若列表中的任何 IP 地址格式不正確，AX Portal 將觸發錯誤，且新列表將無法儲存。
- 該列表必須包含機器人自身的 IP 地址。目前系統不會自動讀取此地址。
- 當從列表中移除 IP 地址並儲存新列表時，該 IP 地址也會從 Cyclone DDS 配置中移除。
- 基於防火牆設置，所有 ROS 參與者可能必須位於同一網路子網段內。

7.4.3.2.2 多播

以下是另一個 Cyclone DDS 配置範例，用於設定您的機器以進行多播：

```
<?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>
```

再次提醒，請將 INTERFACE_NAME 替換為您電腦上實際用於 ROS 通信的 ROS 網絡接口名稱。範例：enp1s0、wlp2s0 等。

若要將機器人切換至多播模式，只需切換「多播」設定按鈕即可。同樣地，需要重啟機器人，更改設置才會生效。

7.4.4 建立連接

現在要啟動電腦上的 ROS 環境，請導航至您的 Docker Compose 檔案並執行：

```
docker compose up -d
```

使用以下指令確認 Docker 容器是否正在運行中：

```
docker ps -a
```

這會列印類似以下內容：

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

使用以下指令以互動模式進入容器

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

每次進入容器時，請務必載入 ROS 設定檔：

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

為了讓多個機器人能在同一網路中運作，我們會在每個機器人上使用不同的 ROS_DOMAIN_ID。請前往 AX Portal 中的「ROS 功能設置」，並設定所需的 ROS 域 ID。允許的範圍為 6 至 41。此外，請注意執行此操作後必須重啟機器人。現在，請返回開啟 ROS Docker 容器的終端機，並將 ROS_DOMAIN_ID 設定為相同值：

```
export ROS_DOMAIN_ID=SELECTED_NUMBER
```

此時您應能透過以下指令看到名為「機器人」的 AX Portal ROS 節點：

```
ros2 node list
```

7.4.5 透過 ROS 存取機器人數據

一旦看到該節點，即可運行以下指令查看可用主題列表：

```
ros2 topic list
```

透過 ROS 2 主題，您可以串流機器人的即時數據。例如，若要訂閱關節位置、速度等數據，請執行：

```
ros2 topic echo /joint_states
```

7.4.6 腳本中的 ROS 2 功能

您可以在腳本中直接匯入 `rcipy`。這讓您能夠建立 ROS 2 物體，例如發佈者、訂閱者等。

ROS 處理器會建立其專屬的 ROS 節點。請使用 `ros_handler.ros_node()` 函式來存取該節點。注意：請勿在此節點上執行 `rcipy.spin`，因為此操作已在 ROS 處理器內部完成。

7.4.7 執行動作

AX Portal ROS 節點會建立一個名為「/robot/execute_function」的動作伺服器，用於執行橋接函式。此動作採用來自 `generic_msgs` 套件 (<https://github.com/Epson-Robots/epson-ax-ros2>) 的自訂定義 `ExecuteFunction` 動作，必須安裝該套件才能取得動作定義。請將套件複製到您的工作區，並使用 `colcon` 進行建置：

```
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
```

為了發送 ROS2 動作到機器人，您的客戶端需要取得活動控制權。要執行此操作，請在終端機中執行以下指令，並在 AX Portal GUI 中確認對話框：

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

現在您可以透過終端機執行橋接功能，使用以下指令：

```
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 驗證設置

目前 ROS 處理器僅支援無驗證模式。您需在桌面介面的「設定」菜單中，手動停用此特定連線模組的驗證功能。

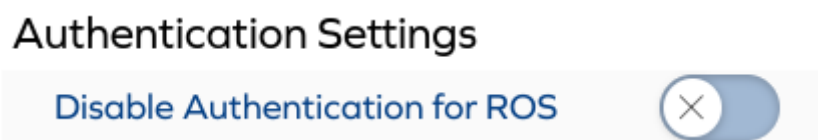


圖 7.4.8：ROS 驗證設置

7.5 Modbus連接模塊

7.5.1 概述

Modbus 模組所使用的協議為 TCP Modbus。在伺服器端，此模組基於 Python 開發，並透過 PyModbus 函式庫實現。詳細資訊請參閱 PyModbus 文件。此模塊分為讀取暫存器（類比輸入暫存器，30000-39999）與寫入暫存器（類比輸出暫存器，40000-49999）。任何 TCP Modbus 客戶端皆可透過與伺服器進行通信。所使用的埠號為 **5020**。本章最後一節提供了數個 Python 代碼範例，其中部分範例說明瞭如何使用 Python 程式腳本建立與伺服器的連接。

7.5.2 設置

目前 Modbus 連線僅支援無驗證模式。您需在桌面介面的「設置」菜單中，手動關閉此特定連接模塊的驗證功能。

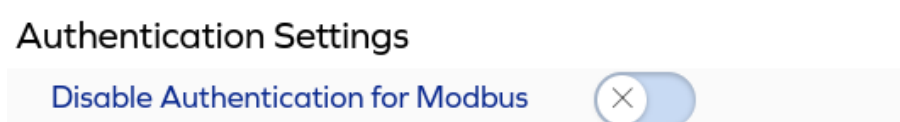


圖 7.5.2：Modbus 模塊設置

7.5.3 寫入暫存器(簡單模式描述)

Modbus 附加元件會對命令暫存器與屬性暫存器的任何資訊變更做出反應。清除寫入暫存器的數據屬於使用者的責任。

7.5.3.1 命令暫存器

向命令暫存器 (30000) 寫入單一位元即可執行命令。部分命令需要在命令數據暫存器 (30001) 中提供參數。若無法同時寫入兩個暫存器，請務必先寫入命令數據暫存器。

絕對地址：30000 - 30001 相對地址：0x000 - 0x001		
位元	指令暫存器 (30000)	數據 (30001)
0	上電	工具 ID
1	休眠	
2	停止	
3	暫停	
4	繼續運行	
5	校準	腳本 ID
6	釋放	
7	保持	
8	變更項目	項目 ID
9	開始腳本	腳本 ID
10	移動到位姿	位姿 ID
11	執行路徑	路徑 ID
12	工具放置	
13	工具選擇	
14 ... 15	保留	

7.5.3.2 屬性暫存器

向屬性暫存器 (30002) 寫入單一位元即可執行指令。部分指令需要在屬性數據暫存器 (30003) 中提供參數。若無法同時寫入兩個暫存器，請務必先寫入屬性數據暫存器。

絕對地址：30002 - 30003 相對地址：0x002 - 0x003		
位元	屬性暫存器 (30002)	數據 (30003)
0	寫入數字輸出	I/O 狀態
1	設置腳本事件	事件 ID
2	確認錯誤	
3	讀取位置	
4	請求活動控制	
5	釋放活動控制	
6	切換到手動模式	
7	切換到自動模式	
8	重置安全設定	
9	繼續運動	
10 ... 15	保留	

7.5.3.3 使用者寫入暫存器

絕對地址：30016 相對地址：0x010		...	絕對地址：30255 相對地址：0x0FF	
0 ... 15	自訂數據		0 ... 15	自訂數據

使用「使用者寫入寄存器」將信息發送至您的腳本應用程式。腳本應用程式的 **modbus.write_user_register** 函數可用於讀取自訂數據。

7.5.4 寫入暫存器(詳細描述)

7.5.4.1 不執行任何操作

清除寫入暫存器的數據是使用者的責任。若要清除指令暫存器 (30000) 或屬性暫存器 (30002)，只需寫入整數 0 即可。

0x000	0x001	0x002	0x003
0		0	

7.5.4.2 啟動

在操作機器人之前，必須先為其馬達供電。要連接的自訂工具 ID 由數字 n 指定。機器人的狀態將從「未上電」（1，位元 0）變更為「就緒」（8，位元 3）。

0x000	0x001	0x002	0x003
1	n		

7.5.4.3 休眠

當工作日結束時，機器人可關閉其馬達電源。機器人可在任何狀態下進行休眠。一旦機器人休眠，狀態將變更為“未上電(1, Bit 0)”。

0x000	0x001	0x002	0x003
2			

7.5.4.4 停止

中斷所有機器人動作，並返回機器人狀態到「運行中」（64，第 6 位元），然後返回「就緒」（8，第 3 位元）。

0x000	0x001	0x002	0x003
4			

7.5.4.5 暫停

在機器人運行中暫停（64，第 6 位）。狀態將切換為暫停（128，第 7 位）。

0x000	0x001	0x002	0x003
8			

7.5.4.6 繼續運行

若機器人處於「暫停」狀態（128，第 7 位元），則繼續運行。狀態將切換為「運行中」（64，第 6 位元）。

0x000	0x001	0x002	0x003
16			

7.5.4.7 校準

目前校準無效。此位元為保留位元。

0x000	0x001	0x002	0x003
32			

7.5.4.8 釋放使能控制 (HGC)

啟用所有機器人關節的手動引導控制，使其處於重力補償模式並可手動運動。狀態需為「就緒」（8，第 3 位元），並將切換至「使能控制」（2048，第 11 位元）。

0x000	0x001	0x002	0x003
64			

7.5.4.9 保持使能控制 (HGC)

將所有機器人關節固定，使其處於位置控制模式且無法再以手動方式移動。狀態需為使能控制 (2048, 第 11 位元)，並將切換至就緒 (8, 第 3 位元)。

0x000	0x001	0x002	0x003
128			

7.5.4.10 變更項目

將當前資料庫項目變更為 ID 為 n 的項目。狀態必須為「就緒」（8，第 3 位元）。

0x000	0x001	0x002	0x003
256	n		

7.5.4.11 開始腳本

執行 ID 為 n 的腳本。狀態需為「就緒」（8，第 3 位元），並將切換為「運行中」（64，第 6 位元）。腳本執行期間，機器人可暫停。腳本執行完畢後，狀態將切換回「就緒」（8，第 3 位元）。

0x000	0x001	0x002	0x003
512	n		

7.5.4.12 移動到位姿

將機器人移動至 ID 為 n 的位姿。狀態需為「就緒」（8，位元 3），並將切換為「運行中」（64，位元 6）。機器人運動期間，其運動可暫停。一旦運動完成，狀態將切換回「就緒」（8，位元 3）。

0x000	0x001	0x002	0x003
1024	n		

7.5.4.13 執行路徑

執行 ID 為 n 的路徑。狀態需為「就緒」（8，位元 3），並將切換為「運行中」（64，位元 6）。機器人運動期間，其運動可暫停。一旦運動完成，狀態將切換回「就緒」（8，位元 3）。

0x000	0x001	0x002	0x003
2048	n		

7.5.4.14 工具放置

使用工具放置物體（例如：張開夾具）。狀態需為「就緒」（8，位元 3），並將切換為「運行中」（64，位元 6）。當夾具運動時，其運動可暫停。一旦運動完成，狀態將切換回「就緒」（8，位元 3）。

0x000	0x001	0x002	0x003
4096			

7.5.4.15 工具拾取

使用工具拾取物體（例如：關閉夾具）。狀態需為「就緒」（8，位元 3），並將切換為「運行中」（64，位元 6）。當夾具運動時，其運動可暫停。一旦運動完成，狀態將切換回「就緒」（8，位元 3）。

0x000	0x001	0x002	0x003

8192			
------	--	--	--

7.5.4.16 寫入數字輸出

此功能保留用於寫入數字輸出。

0x000	0x001	0x002	0x003
		1	d

7.5.4.17 設置腳本事件

若使用者應用程式需等待 Modbus 裝置發送的事件，可使用腳本事件。共有 16 個事件。根據數據 d 的值，事件會被設定或清除。**modbus.wait_for_event** 指令碼函式會等待直到特定事件被設定為真。若 d 為 0，則所有事件皆被清除；若 d 為 65535，則所有事件皆被設定為真；若 d 為 9，則第一和第四個事件被設定為真，其餘事件則被清除。

0x000	0x001	0x002	0x003
		2	d

7.5.4.18 確認錯誤

錯誤必須先被確認，才能被清除。然而，此步驟僅需用於讀取錯誤信息。機械手臂狀態不會受到確認操作的影響。若主要與次要錯誤代碼（40004-40005）皆為 0，則無需再確認任何錯誤。

0x000	0x001	0x002	0x003
		4	

7.5.4.19 讀取位置

更新機器人的位置數據。每當請求位置數據時，當前的機器人關節角度和機器人坐標便會寫入對應的讀取暫存器（40256-40267）。

0x000	0x001	0x002	0x003
		8	

7.5.4.20 請求活動控制

請求活動控制以向機器人發送指令。使用者資料暫存器與讀取暫存器可在無需主動控制的情況下存取。目前擁有主動控制權的監控者將收到通知，並需主動釋放控制權。一旦控制權獲准，相應的讀取暫存器（40008）即被設定。

0x000	0x001	0x002	0x003
-------	-------	-------	-------

		16	
--	--	----	--

7.5.4.21 釋放活動控制

釋放活動控制權限，以便其他監控者能向機器人發送指令。即使未持有活動控制權限，仍可存取使用者數據暫存器與讀取暫存器。對應的讀取暫存器 (40008) 將切換為 0。

0x000	0x001	0x002	0x003
		32	

7.5.4.22 切換到手動模式

切換到手動模式。在此模式下，機器人僅會在使用者按下啟用按鈕時運動。請注意，機器人需執行安全重置才能切換模式。

0x000	0x001	0x002	0x003
		64	

7.5.4.23 切換到自動模式

切換到自動模式。請注意，機器人切換模式時必須執行安全重置。

0x000	0x001	0x002	0x003
		128	

7.5.4.24 安全重置

啟動/重啟互鎖重置。針對某些安全功能，機器人需要使用者執行明確且有意的操作才能繼續運行。若要檢查是否需要此類操作，請使用對應的讀取暫存器 (40009)。

0x000	0x001	0x002	0x003
		256	

7.5.4.25 開始運動中斷

安全中斷發生後，機器人可繼續執行並恢復其運動。此屬性可用於使機器人再次運動。否則，可發送停止指令來停止正在運行的程式。若要檢查是否需要此類動作，請使用對應的讀取暫存器 (40009)。

0x000	0x001	0x002	0x003
		512	

7.5.5 讀取暫存器

7.5.5.1 狀態暫存器

使用狀態暫存器讀取機器人的當前狀態。狀態以位元數的形式提供。

7.5.5.1.1 第一個狀態暫存器

絕對地址：40000 相對地址：0x000	
0	未上電
1	上電中
2	加載中
3	就緒
4	處理中
5	校準中
6	運行中
7	自主模式
8	暫停
9	繼續
10	保護停止
11	安全防護
12	位置限制
13	使能控制 (HGC)
14	碰撞
15	急停

7.5.5.1.2 第二狀態暫存器

絕對地址：40001 相對地址：0x001	
0	正常停止
1	錯誤
2	斷電中
3	停止中
4.. 15	保留

7.5.5.1.3 第三狀態暫存器

絕對地址：40002 相對地址：0x002	
0 ... 15	保留

7.5.5.1.4 第四狀態暫存器

絕對地址：40003 相對地址：0x003	
0 ... 15	保留

7.5.5.1.5 主要錯誤暫存器

絕對地址：40004 相對地址：0x004	
0 ... 15	有關錯誤類別的信息

主要錯誤暫存器讀取當前錯誤識別碼的前 2 位元組。請使用屬性暫存器確認錯誤，以便讀取下一個錯誤。

有關錯誤暫存器值的詳細含義，請參閱錯誤寄存器映射。

7.5.5.1.6 次要錯誤暫存器

絕對地址：40005 相對地址：0x005	
0 ... 15	關於錯誤類別中確切錯誤的信息

輕微錯誤暫存器讀取當前錯誤識別碼的最後 2 位元組。請使用屬性暫存器確認錯誤，以便讀取下一個錯誤。

有關錯誤暫存器值的詳細含義，請參閱錯誤寄存器映射。

7.5.5.1.7 程式列表暫存器

絕對地址：40006 相對地址：0x006	
0 ... 15	當前腳本 ID

程式列表讀取目前正在運行中的腳本 ID。

7.5.5.1.8 校準暫存器

絕對地址：40007 相對地址：0x007	
0	機器人是否已校正

校準暫存器用於讀取機器人是否已校準。0 表示機器人尚未標定，1 表示機器人已校準。請注意，機器人必須連接其馬達，才能提供有意義的校準回報。

7.5.5.1.9 活動控制暫存器

絕對地址：40008 相對地址：0x008	
0	Modbus 監控器是否具有活動控制權

「活動控制寄存器」用於讀取 Modbus 監看器是否具有活動控制權。0 表示 Modbus 沒有活動控制權。1 表示 Modbus 具有活動控制權。Modbus 監看器可使用 0x30002 0x04「請求活動控制」來請求活動控制權。

7.5.5.1.10 中斷暫存器

中斷暫存器指示有效的中斷對話選項。

絕對地址：40009 相對地址：0x009	
0	需重置安全
1	需繼續執行運動
2	切換到手動模式
3.. 15	保留

7.5.5.1.11 操作模式暫存器

操作模式暫存器指示機器人處於手動操作模式 (1) 或自動操作模式 (0)。

絕對地址：40010 相對地址：0x00A	
0	機器人是否處於手動操作模式

7.5.5.2 使用者讀取暫存器

絕對地址：40016 相對地址：0x010		...	絕對地址：40255 相對地址：0x0FF	
0 ... 15	自訂數據		0 ... 15	自訂數據

使用使用者讀取暫存器從腳本程式接收信息。程式的 **modbus.read_user_register** 函式將讀取數據。

7.5.5.3 位置暫存器(第一關節至第六關節)

絕對地址：40256 相對地址：0x100		...	絕對地址：40261 相對地址：0x105	
0 ... 15	第一關節的關節角度		0 ... 15	第六關節的關節角度

讀取特定關節的角度數值，單位為十分之一度。若讀取數值為 101，則該關節的角度為 10.1°。該值僅在屬性暫存器中寫入「讀取位置標誌」時才會更新。若讀取的值大於 32768 ($=2^{15}$)，則對應負值。請減去 32768 以讀取絕對值。例如，值 32869 對應關節角度 -10.1°。

7.5.5.4 座標暫存器 (x, y, z, X軸旋轉, Y軸旋轉, Z軸旋轉)

絕對地址：40262 相對地址：0x106		...	絕對地址：40267 相對地址：0x10B	
0 ... 15	TCP 的 x 坐標		0 ... 15	TCP 的 Z軸旋轉坐標

讀取工具中心點的坐標或方向，順序為 x、y、z、X軸旋轉、俯仰、Z軸旋轉。空間坐標以十分之一毫米為單位讀取。若某坐標讀取數值為 314，則該數值為 31.4 毫米。旋轉坐標以十分之一度為單位讀取。若某坐標讀取數值為 271，則該數值為 27.1（毫米或度）。僅當在屬性暫存器中寫入讀取位置標誌時，該值才會更新。若讀取的值大於 32768（ 2^{15} ），則對應為負值。請減去 32768 以讀取絕對值。例如，值 33039 對應的坐標值為 -27.1（mm 或 °）。

7.5.6 錯誤暫存器映射

錯誤信息總是出現在兩個暫存器中。次要錯誤寄存器包含實際的錯誤編號。主要錯誤寄存器包含以下其中一個值，這些值對應於不同的錯誤類別，對應關係如下：

101	附加模組	201	模塊	301	AX
102	配置	202	Cognex	302	CFG
103	資料庫	203	控制	303	CMK
104	控制器	204	ROS	304	CMD
105	編輯	205	MODBUS	305	CTL
106	橋接器	206	REST	306	ECAT
107	連接	207	TCP	307	LIC
108	DIALOG			308	數學
109	經銷商			309	MCM
110	活動			310	其他
111	運動學			311	作業系統
112	SOCKET			312	PPM
113	核心			313	PYAPI
114	安全			314	SIG
115	設置			315	系統
116	訊號			316	TLC

117	狀態
118	腳本
119	THREAD

317	UTIL
-----	------

7.5.6.1 範例

若錯誤暫存器讀取值為 **[113, 4]**，則對應于機器人觸發的錯誤代碼 **CORE_004**。

7.5.7 Python 代碼範例

這些範例是用於遠端連接至機器人的外部系統。

7.5.7.1 範例 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()
```

連接至伺服器。讀取機器人狀態。

7.5.7.2 範例 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()
```

連接至伺服器。將機器人連接至其馬達。

提示： << 指令用於位元移位。

位元	指令	整數
0	1 << 0	1
5	1 << 5	32

7.5.7.3 範例 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()
```

連接至伺服器。運行 ID 為 5 的應用程式。

7.5.7.4 範例 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()
```

連接至伺服器。讀取當前錯誤訊息。確認錯誤訊息。

8. Cognex攝影機模塊

8.1 概述

Cognex 攝像頭模塊可讓您將預先配置好的 Cognex 攝影機連接到 AX Portal。連接完成後，您可以觸發攝影機上的預設工作，以拍攝並處理影像。處理後的數據隨後會傳輸到 AX Portal，您可在程式中讀取這些數據。

8.2 攝影機設置

若要設置 Cognex 攝影機以確保與 AX Portal 相容，請依照以下步驟操作：

1. 下載並安裝 Cognex 的 In-Sight Explorer 軟體。
2. 為攝影機供電並將其連接到您的網路。若攝影機的網路設置為靜態 IP 地址，您亦可直接將其連接到計算機。
3. 啟動 In-Sight Explorer，若系統要求，請允許訪問權限，並安裝所需組件（例如 Microsoft Visual C++、Microsoft .NET Framework）。
4. 完成後，Explorer 的 Easybuilder 視圖將自動開啟。
5. 導航至「系統」→「將傳感器/裝置新增至網路」。尋找您的攝影機並將其新增至網路。
6. 請確認攝影機的網路設置與 AX 系列機器人的網路設置相容。同時請確認 Telnet 埠已設置為 3000。
7. 選擇您的攝影機並按下「連接」按鈕，將其與 In-Sight Explorer 連接。
8. 現在您已就緒建立工作（例如用於校準攝影機，以及識別供機器人拾取的物體），並儲存它們到攝影機中。更多詳細信息請參閱 Cognex 文件。

8.3 攝影機程式

要在 AX Portal 程式中使用 Cognex 攝影機，需確保

- 機器人與攝影機必須位於同一網路中，
- 攝像機的 Telnet 埠號須設置為 3000，
- 且攝影機已設置為線上狀態。

以下是一個範例，說明如何使用預先配置的工作來偵測圓形的坐標。

```
# 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)
```