



入門ガイド

v1.7.4

目次

1. ビジョンシステムの概要	1
2. 入門ガイド：ビジョンシステムによるピッキング（Vizティーチング通信）	7
2.1. ビジョンシステムのハードウェア設置	8
2.2. ロボット通信設定	18
2.3. ロボットハンド・アイ・キャリブレーション	34
2.4. ワーク認識	53
2.5. 把持と配置を実行	74
3. 入門ガイド：ビジョンシステムによるピッキング（標準インターフェース通信）	93
3.1. ビジョンシステムのハードウェア設置	94
3.2. ロボット通信設定	104
3.3. ロボットハンド・アイ・キャリブレーション	151
3.4. ワーク認識	176
3.5. 把持と配置を実行	198
4. 入門ガイド：ビジョンシステムによる単載デパレタイズ（Vizティーチング通信）	219
4.1. ビジョンシステムのハードウェア設置	221
4.2. ロボット通信設定	230
4.3. ロボットハンド・アイ・キャリブレーション	246
4.4. 段ボール箱の認識	266
4.5. 把持と配置を実行	272

1. ビジョンシステムの概要

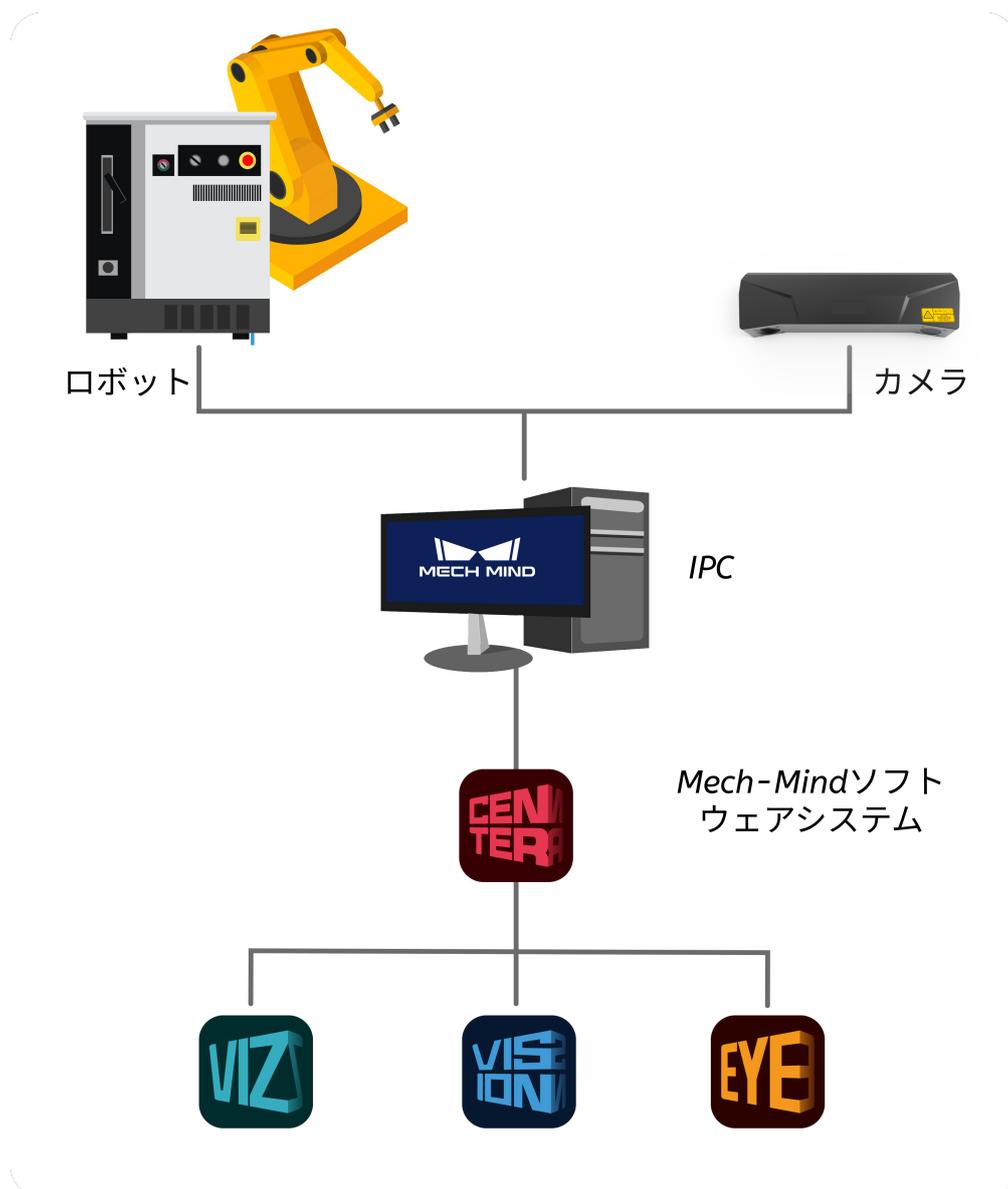
Mech-Mindビジョンシステムの説明

Mech-Mindビジョンシステムは、Mech-Mindが提供する自動車、物流、スーパーマーケット、重工業などの幅広い業界に適用可能な完全な3Dビジョンソリューションです。これは、部品供給、デパレタイジング・デパレタイジング、位置決め・組立、商品仕分け、品質検査などのシーンで広く採用されています。

Mech-Mindビジョンシステムは、一般的に産業用ロボットや協働ロボットと組み合わせて使用されます。

ビジョンシステムの構成

下図に示すように、Mech-Mindビジョンシステムは一般的に、カメラ、Mech-Mindソフトウェアシステム、IPCおよびロボットで構成されています。



カメラ

Mech-Mindが独自に研究開発したMech-Eye産業用3Dカメラを指します。それを使用して対象物の画像や位置情報を取得できます。

Mech-Mindソフトウェアシステム

Mech-Mindソフトウェアシステムは、カメラで取得した画像と位置情報をもとにビジョン処理を行い、対象物の位置と姿勢、計画されたロボットの動作経路を出力し、把持、パレタイジング・デパレタイジング、接着、仕分けなどの自動化作業でロボットをガイドするために使用されます。

IPC

Mech-Mindソフトウェアシステムの動作環境を提供するコンピュータ機器を指します。

ロボット

ロボットは、移動、動作、位置決めなどのタスクを実行するために、ある程度の自律性を持

ってプログラムされた機械装置です。Mech-Mindビジョンシステムにおいて、ロボットはビジョンシステムの出力結果に応じた自動化作業を実行します。



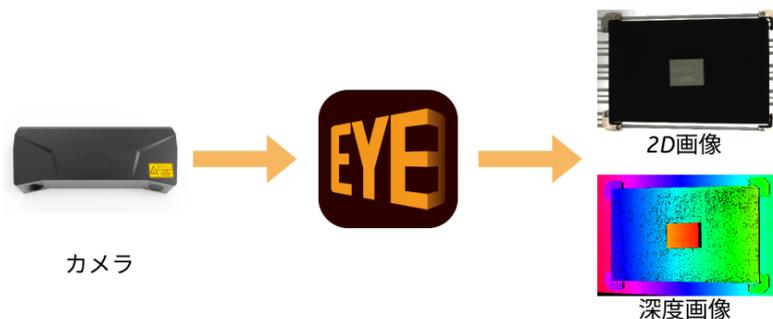
- ロボットセルは、ロボット本体、コントローラおよびティーチペンダントで構成されます。
- 高度な自動化が求められる産業用途では、ロボットの動きや動作を制御するためにPLC（プログラマブルロジックコントローラ）を使用することがあります。より高度な制御とモニタが必要な場合、ロボットは、動作経路計画、タスクスケジューリング、動作制御など、より複雑なプログラミングとロボットの制御のために上位システムを使用することもできます。Mech-Mindビジョンシステムは、PLCや上位システムと組み合わせて使用することをサポートしています。
- 本ガイドでは、ロボット、PLC、上位システムを総称して「ロボット側」、カメラとMech-Mindソフトウェアシステムを総称して「ビジョン側」とします。

Mech-Mindソフトウェアシステムの概要

Mech-Mindソフトウェアシステムは、ロボットとその周辺機器に3Dビジョンソリューションを提供するためにMech-Mindが独自に研究開発したソフトウェア群で、以下のソフトウェアで構成されています。

- Mech-Eye Viewer カメラ調整ソフトウェア

Mech-Eye Viewerは、Mech-Eye産業用3Dカメラのパラメータを対象物の特徴に合わせて調整し、高品質な2D画像、深度画像および点群を簡単かつ迅速に取得することができます。



- Mech-Vision 画像処理ソフトウェア

Mech-Visionは、完全なグラフィカルインターフェースにより、ノーコードでバラ積み対象物のピッキング、高精度位置決め、組立、工業検査/測定、自動経路生成などの高度なマシンビジョンアプリケーションを実現します。

このソフトウェアは、現場で取得された画像データに基づいて一連のビジョン処理を行い、最終的にビジョン結果（対象物の位置や向きなど）を出力します。また、ビジョン結果に対して簡単な経路計画を行い、ロボットの把持経路を出力することも可能です。



● Mech-Viz ロボット制御ソフトウェア

Mech-Vizは、Mech-Visionから得られた点群やワークの位置情報を使用して、把持、取り扱い、配置などのロボット経路を自動的に計画します。

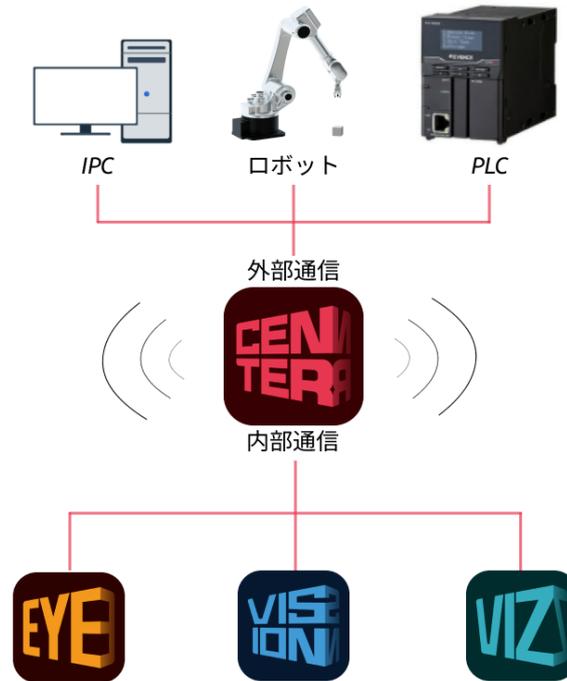
ロボットのワークフローを可視化で構築することに対応し、実際に操作する前に動作確認ができる3Dシミュレーション機能を提供します。数多くの世界中のメーカーのロボットに適用しています。



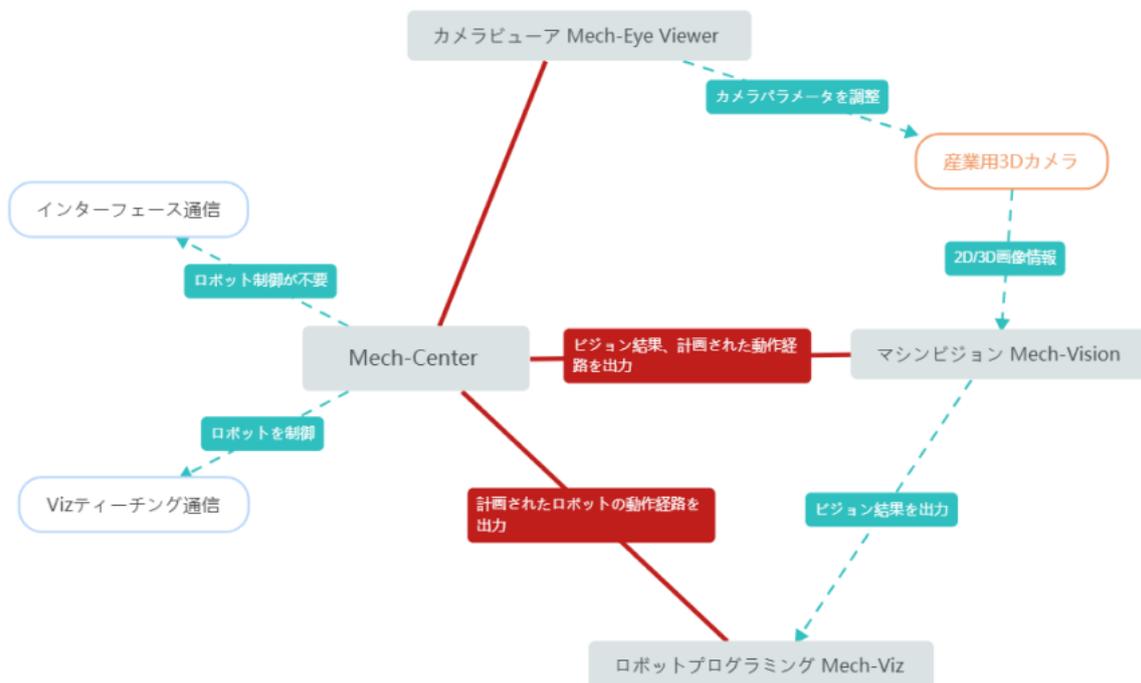
● Mech-Center 統合管理ソフトウェア

Mech-Centerは、標準インターフェースの通信プロトコルを提供するほか、お客様が通信サービスをカスタマイズし、外部機器と通信してビジョンシステムによるロボットの自動化作業のために連携できるようにすることも可能です。

ソフトウェアシステムでは、Mech-Centerがソフトウェア群の仲介役を担っています。例えば、Mech-Visionによって出力されたビジョン結果を送信し、ロボットの把持経路を計画するためにMech-Vizに送信します。外部機器と通信する場合、Mech-CenterはMech-VisionやMech-Vizによって出力されたビジョン結果または計画されたロボットの動作経路をロボット、PLCまたは上位システムに送信します。ソフトウェアシステムと外部機器との通信方式は下図のとおりです。



Mech-Mindソフトウェアシステムでは、各ソフトウェア間の関係を下図に示します。



ロボット側との通信

ビジョン側によって出力されたビジョン結果（Mech-Vision）または計画されたロボットの動作経路（Mech-VisionまたはMech-Viz）をロボット側に送信するために、Mech-Mindソフトウェアシステムがロボット側と正常に通信する必要があります。

Mech-Mindビジョンシステムは、以下の通信方式をサポートしています。

- Vizティーチング通信

Vizティーチング通信では、ビジョン側がロボットを制御する、つまり、ビジョン側がマスター、ロボットがスレーブとして機能します。ビジョン側は、ワークの供給やパレタイジング・デパレタイジングなど、計画されたロボットの動作経路に従って、対応するタスクを実行するようにロボットを制御することになります。

この通信方式を使用する場合、Vizとの通信プログラムをロボットに読み込むか、またはロボットのSDK（ソフトウェア開発キット）を介してロボットの制御を取得する必要があります。ビジョン側とPLCまたは上位システムとの通信に対応していません。

- インターフェース通信

インターフェース通信では、ビジョン側がロボットの制御を取得する必要がなく、ロボット側がマスター、ビジョン側がスレーブとして機能します。ロボット側とビジョン側は同じ標準インターフェースの通信プロトコルで通信を行い、ロボット側からリクエストを送信し、ビジョン側がそれを処理して結果を返します。リクエストに応じて、ビジョン側はビジョン結果か、計画されたロボットの動作経路を返すことになります。ロボットは、ビジョン側から返された結果に基づいて、さらなる判断や適切なタスクを実行します。

この通信方式を使用する場合、ロボットのインターフェースプログラム（ビジョン側と通信するためのプログラム）とロボットの把持プログラム（ビジョン側から返されたデータを受信し、ロボットを制御するためのプログラム）を作成してロボットに読み込む必要があります。ビジョン側がロボット、PLC、上位システムとの通信に対応しています。

通信方式の詳細については、[通信方式の概要](#)をご参照ください。

2. 入門ガイド：ビジョンシステムによるピッキング（Vizティーチング通信）

本ガイドを読むことで、Vizティーチング通信を使用して、3Dビジョンシステムによる小型金属部品の把持アプリケーションを実装する方法を習得できます。

概要

- カメラ：Eye to Hand方式で取り付けられているMech-Eye PRO Mカメラ
- ロボット：ABB_IRB_1300_11_0_9
- ワーク：トラックリンク（小型金属ワーク）



このアプリケーションでは、ワークのCADモデルファイルを事前に用意する必要があります。CADモデルファイルは、点群マッチングモデルの生成に使用されます。

- ロボットハンド：グリッパー（内径把持型）



このアプリケーションでは、経路計画で衝突検出を行うために、グリッパーのOBJ形式のモデルファイルを事前に用意する必要があります。

- 使用されるソフトウェア：Mech-Vision 1.7.2バージョン、Mech-Viz 1.7.2バージョン、Mech-Center 1.7.2バージョン、Mech-Eye Viewer 2.1.0バージョン
- 通信方式：Vizティーチング通信



上記と異なるカメラ型番やロボットブランド、ワークを使用する場合、一部調整を行う必要があります。

ビジョンアプリケーションの実装

ビジョンアプリケーションの実装は、一般的に5つの段階に分けられています。



詳細は下表のとおりです。

番号	階段	説明
1	ビジョンシステムのハードウェア設置	Mech-Mindビジョンシステムのハードウェアの取り付けと接続、ソフトウェアのインストールと接続を完了します。
2	ロボット通信設定	Vizとの通信プログラムと設定ファイルをロボットシステムに読み込み、ビジョン側とロボット側との通信を構築し、Mech-Mindソフトウェアシステムによるロボットの制御を実現します。
3	ロボットハンド・アイ・キャリブレーション	自動キャリブレーション（Eye To Hand）を実行し、カメラ座標系とロボット座標系との対応関係を求めます。
4	ワーク認識	ソリューションライブラリ内の「一般的な部品認識」を使用してワークの位置姿勢を計算し、ビジョン結果を出力します。
5	把持と配置を実行	Mech-Vizでワークフローを構築し、ロボットがワークの把持と配置を繰り返し実行するようにします。

本節の説明は終わりです。次に、アプリケーションの実装を完了させてください。

2.1. ビジョンシステムのハードウェア設置

本ガイドを読むことで、Mech-Mindビジョンシステムの設置方法を習得できます。

Mech-Mindビジョンシステムの設置は、**梱包内容の確認** → **ハードウェアの取り付け** → **ネットワーク接続** → **ソフトウェアのアップグレード（オプション）** → **ビジョンシステムが画像を正常に取得できるか確認**の順で行う必要があります。

▶ <https://www.youtube.com/watch?v=IQ-anVFFEk> (YouTube video)

動画：ビジョンシステムのハードウェア設置

梱包内容の確認

1. お受け取りの際は、梱包に問題がないことを確認してください。
2. 梱包中にある『同梱包一覧』を確認し、商品や付属品に欠品や損傷がないことを確認してください。

下図は、梱包箱に含まれるものおよび付属品の例を示しています。下記の同梱品一覧は参照のみを目的としています。梱包中に入っているものは、実際の梱包の中にある『同梱包一覧』をご参照ください。



番号	種類	名前	機能
1	IPCと付属品	IPC	Mech-Mindソフトウェアシステム搭載
2		IPCの付属品	WIFIアンテナなどの付属品
3		IPCの電源ケーブルとアダプター	IPCの電源を入れる
4	カメラと付属品	Mech-Eye産業用3Dカメラ	画像取得
5		カメラ取扱説明書	Mech-Eye産業用3Dカメラの取扱説明書
6		カメラの付属品	カメラの取り付け

番号	種類	名前	機能
7	その他の付属品	ドングル	ソフトウェアのライセンス供与
8		キャリブレーションボード	カメラキャリブレーション
9		フランジ	キャリブレーションボードの取り付け
10		カメラDC電源ケーブル（標準仕様20メートル）	カメラをレール電源に接続（オプションでより長い電源ケーブルも利用可能）
11		カメラのLANケーブル（標準仕様20メートル）	カメラをIPCに接続（オプションでより長いLANケーブルも利用可能）
12		レール電源（標準仕様）	Mech-Eye産業用3Dカメラの電源を入れます（オプションで電源アダプターも利用可能）
13	同梱品一覧		梱包に含まれるすべてのものと付属品を一覧表示します



万一、商品紛失・破損等が発生した場合は、Mech-Mind株式会社にお問合せください。

その他のものを準備

本ガイドでは、カメラ梱包に入っているものの他に、下表に示すようなものを自分で用意する必要があります。

名前	機能
モニター	IPCの画面表示
HDMIケーブル	IPCとモニターを接続
RJ45 LANケーブル	IPCとコントローラを接続



本ガイドでは、IPCはRJ45 LANケーブルでコントローラと接続し、カメラのLANケーブルでカメラと接続します。ルータ経由でIPCをロボットコントローラに、IPCをカメラに接続することもできます。

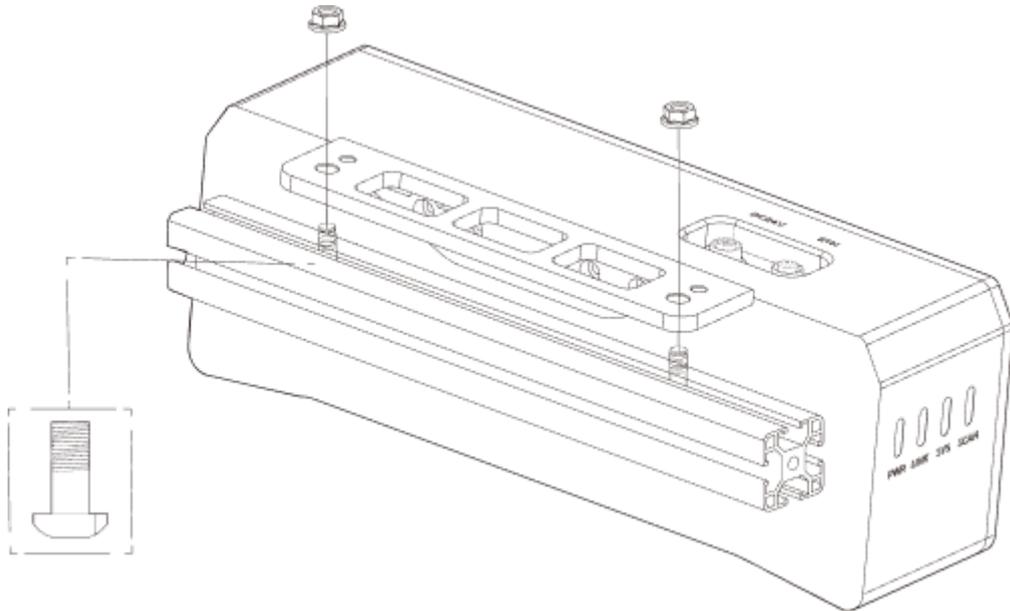
ハードウェアの取り付け

カメラの取り付け

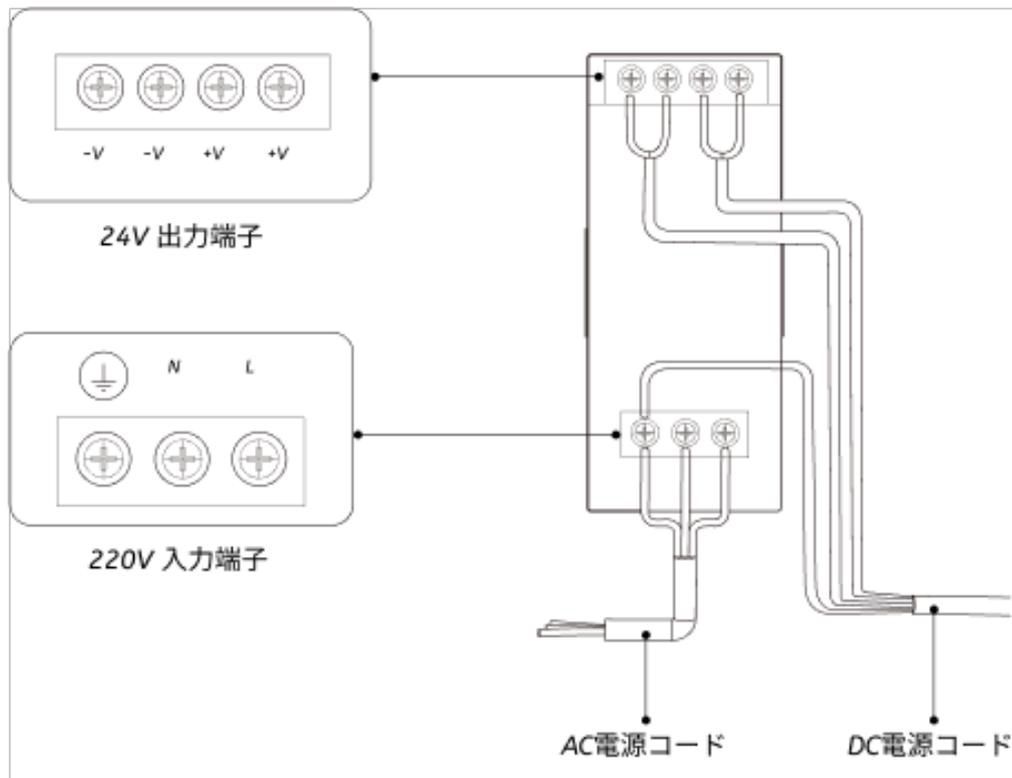


本ガイドでは、カメラはカメラスタンドに設置されています（Eye to Hand方式）。また、カメラをロボット先端に取り付けることも可能です（Eye in Hand方式）。

1. 付属品からカメラを取り付けるためのネジとスパナを探します。
2. 下図に示すように、レンチを使用して2本のネジを締め、カメラを固定します。

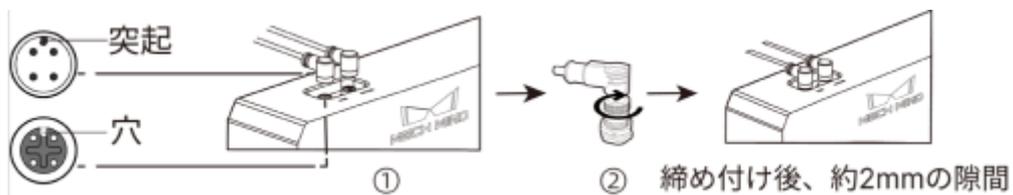


3. カメラ取り付け後、レンズ保護フィルムを剥がします。
4. レール電源でカメラの電源を入れます。
 - DC電源ケーブルの接続：
 - +Vを24V出力端子の+Vに接続します。
 - -Vを24V出力端子の-Vに接続します。
 - PEを220Vの入力端子Ⓧに接続します。



5. カメラのLANケーブルを取り付けます。

カメラのLANケーブルの航空コネクタの突起をETHポートの開口部に差し込み、ナットを締めます。



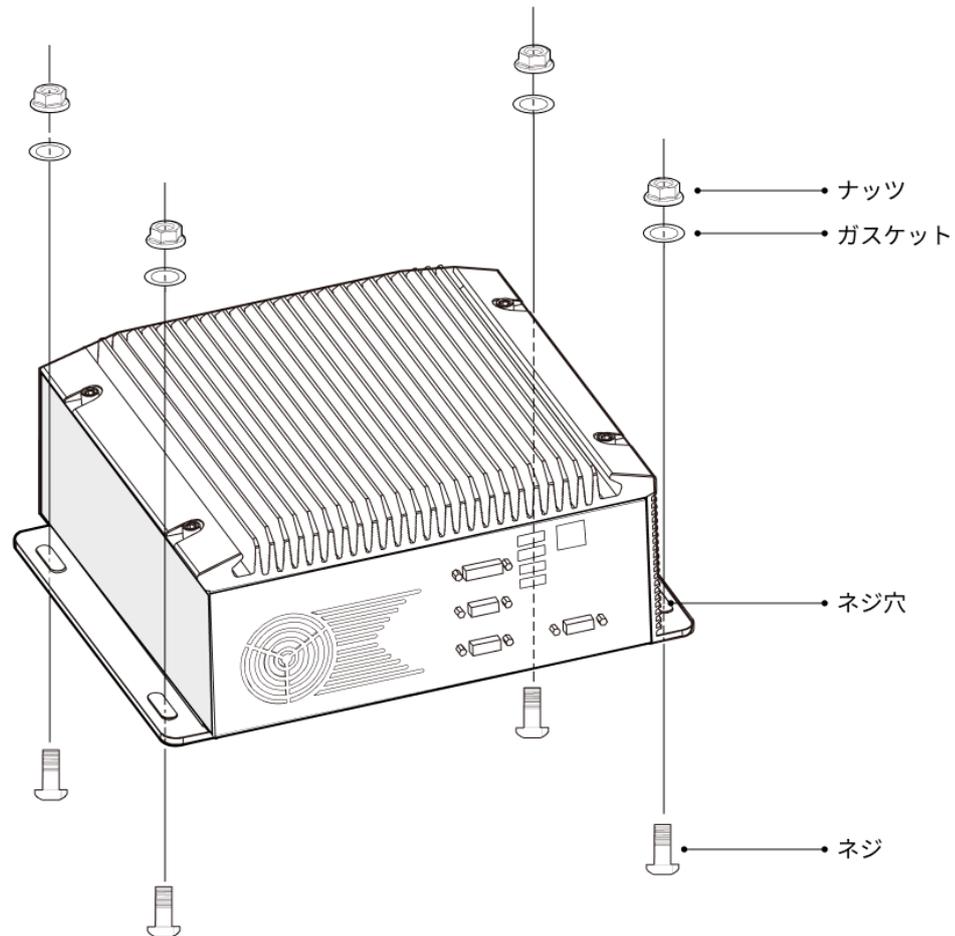
IPCの取り付け



IPCは一般的にコントローラに設置されます。IPCの設置環境は、放熱性、通気性、防塵効果が必要であり、LANケーブル、HDMIケーブル、USBポートの設置やメンテナンスがしやすい場所を選択する必要があります。

IPCを取り付けるには、以下の手順を実行します。

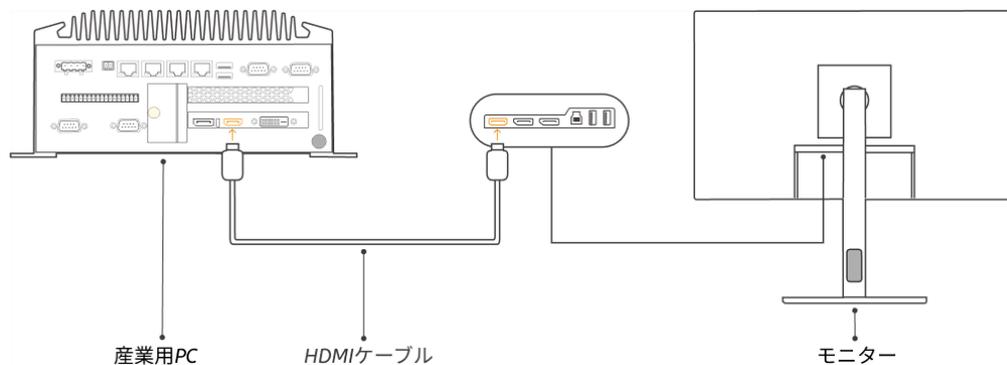
1. スパナ、ネジ、ナット、スペーサーは各自でご用意ください。
2. コントローラにIPCの取り付け穴がある場合は、下図のようにネジ、スペーサー、ナットの順に入れ、スパナでナットを締めて、コントローラにIPCを取り付けます。



コントローラの位置が移動しない場合は、この手順を省略してIPCをコントローラに直接設置することができます。

3. IPCとモニターをHDMIケーブルで接続します。

下図のように、HDMIケーブルの一端をモニターのHDMIポートに、もう一端をIPCのHDMIポートに差し込みます。



4. 電源アダプタでIPCの電源を入れます。

電源アダプタの電源プラグをIPCの電源入力ポートに差し込み、電源アダプタのもう一方の端を電源に接続します。

5. ドングルを差し込みます。

ドングルをIPCのUSBポートに差し込みます。

6. IPCの電源が入った後、IPCを起動します。

- IPCの正常な起動後は、電源インジケータは常に点灯している必要があります。
- IPCが起動しない場合は、Mech-Mindにお問い合わせください。

ネットワーク接続

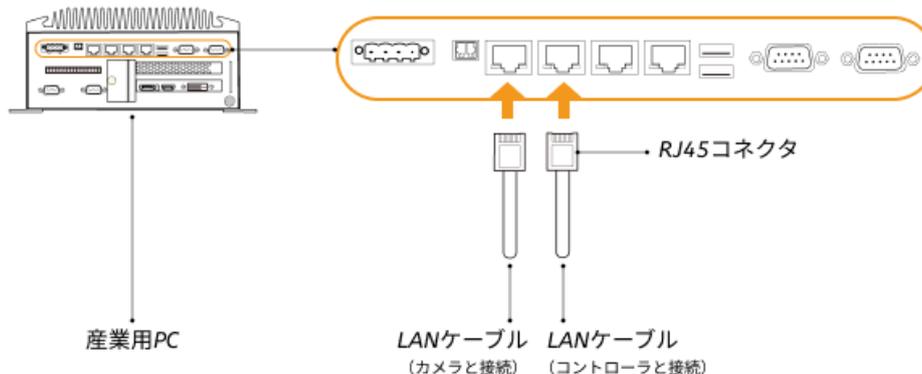
以下では、IPCとカメラとのネットワーク接続、IPCとロボットとのネットワーク接続について説明します。

以下の操作で、次のIPアドレスに従ってネットワークを設定します。実際のネットワーク環境に応じて調整してください。

機器		IPアドレス
IPC	カメラに接続されているLANポート	192.168.100.10
	コントローラに接続されているLANポート	192.168.200.10
カメラ		192.168.100.20
ロボット		192.168.200.20 (ロボットが設定済み)

IPCとカメラ、IPCとロボットコントローラを接続

1. カメラが接続されているLANケーブルのもう一方の端をIPCのLANポートに差し込みます。



2. 両端にRJ45コネクタが付いたLANケーブルを使用して、LANケーブルの一端をIPCのLANポートに、もう一端をロボットコントローラのLANポートに差し込みます。

IPCのIPアドレスを設定

1. IPCで、**コントロールパネル** > **ネットワークとイーサネット** > **ネットワークと共有センター** > **アダプターの設定の変更**を選択すると、**ネットワークの接続**の画面が表示されます。
2. カメラが接続されているLANポートを選択し、右クリックして**名前の変更**を選択し、「To_camera」など、ネットワーク接続を示すポートに名前を変更します。
3. カメラが接続されているLANポートを選択し、右クリックして**プロパティ**を選択し、**イーサネットのプロパティ**の画面に入ります。
4. **イーサネットプロトコルバージョン 4 (TCP/IPv4)** を選択して[**プロパティ**]をクリックし、**Internet プロトコルバージョン 4 (TCP/IPv4) のプロパティ**の画面に入ります。
5. **次のIPアドレスを使う**を選択し、**IPアドレス**を「192.168.100.10」、**サブネットマスク**を「255.255.255.0」、**デフォルトゲートウェイ**を「192.168.100.1」に設定してから、[**OK**]をクリックします。



6. 手順2~5を繰り返し、ロボットコントローラが接続されているLANポートの名前を変更し（例：To_robot）、LANポートのIPアドレスを設定します。例えば、LANポートのIPアドレスは「192.168.200.10」です。



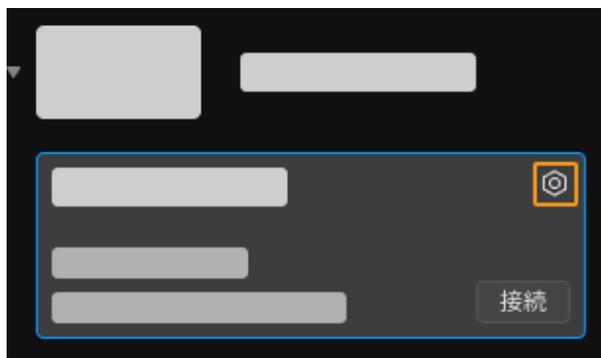
ロボットコントローラに接続するIPCのLANポートのIPアドレスは、ロボットのIPアドレスと同じネットワークセグメント内にある必要があります。

カメラのIPアドレスを設定

1. IPCのデスクトップ上にある  アイコンをダブルクリックし、Mech-Eye Viewerを起動しま

す。

- 検出されたカメラリストからカメラを選択し、マウスをカメラ情報バーに移動すると、が表示されます。クリックすると**IP設定画面**に入ります。



カメラが検出できない、接続できない場合は、[カメラのトラブルシューティング](#)を参照し、解決してください。

- カメラエリアで、**静的IPに設定**を選択し、**IPアドレスクラス**を「クラスC 192.168.x.x」、**IPアドレス**を**192.168.100.20**、**サブネットマスク**を「255.255.255.0」に設定してから、**[適用]**をクリックします。



カメラのIPアドレスは、カメラが接続されているIPCのLANポートのIPアドレスと同じネットワークセグメント内にある必要があります。

ネットワーク接続をテスト

- ショートカットキー **Win + R** で**実行画面**を表示します。
- 名前に**cmd**と入力し、**[OK]**をクリックします。

3. コマンドウィンドウにping XXX.XXX.XX.XXと入力し、[Enter]をクリックしてコマンドを実行します。



XXX.XXX.XX.XXは、実際に設定されたカメラまたはロボットのIPアドレスに変更します。

ネットワークの接続が正常であれば、次のようなメッセージが表示されます。

```
XXX.XXX.XX.XXにPingを送信しています 32バイトのデータ:  
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128  
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128  
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128  
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128
```

ソフトウェアのアップグレード（オプション）

Mech-Mindから購入したIPCには、Mech-Mindソフトウェアシステムがプリインストールされています。

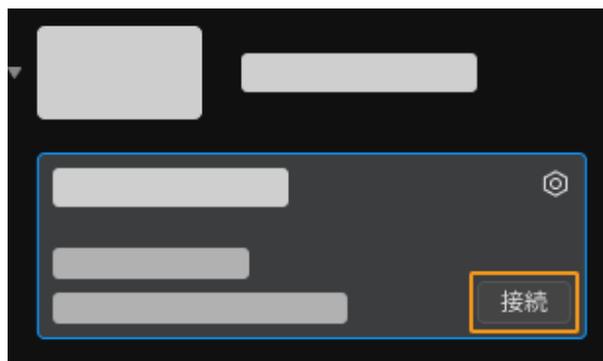
本機のソフトウェアが最新バージョンであるかどうかを確認してください。ソフトウェアがすでに最新バージョンである場合は、以下の内容をスキップしてください。ソフトウェアが最新バージョンでない場合は、以下の内容を参照して、ソフトウェアを最新バージョンにアップグレードしてください。

- [Mech-Eye SDKソフトウェアのダウンロードとインストール](#)
- [Mech-Vision、Mech-Viz、Mech-Centerソフトウェアのダウンロードとインストール](#)

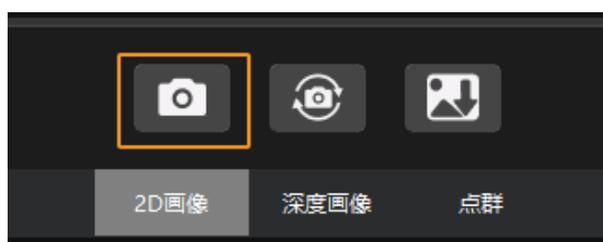
ビジョンシステムが正常に画像を取得できるか確認

IPC、カメラとロボット間のネットワークにアクセスできることを確認した後、以下の手順でビジョンシステムが正常に画像を取得できることを確認します。

1. ワークをカメラの視野中心に置き、エッジと最高層にあるワークが視野に入ることを確認します。
2. IPCのデスクトップにある  アイコンをダブルクリックしてMech-Eye Viewerを起動します。
3. 検出されたカメラからカメラを選択して[接続]をクリックします。

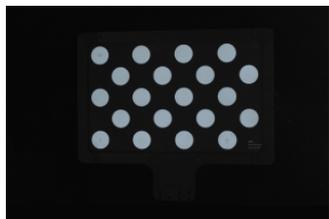
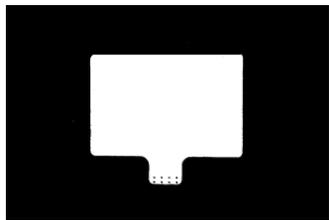
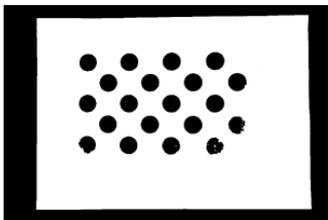
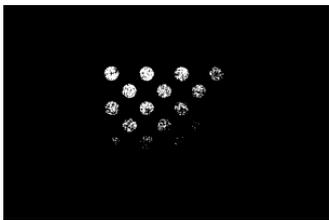


4. カメラ接続後、[一回キャプチャ]をクリックします。



5. 撮影した画像の品質が条件を満たしていることを確認します。

- 2D画像：明らかな露出過度（白すぎて対象物が見えない）、露出不足（暗すぎて対象物のディテールが判別できない）がありません。
- 深度画像と点群：対象物に激しい抜けがありません。

	正常	露出過度	露出不足
2D画像			
点群			



取得した画像が要件を満たしていない場合、[Mech-Eye Viewer](#)でパラメータを調整してください。

以上でビジョンシステムのハードウェアの構築方法の説明は終わりです。

2.2. ロボット通信設定

本ガイドを読むことで、Vizとの通信プログラムをABBロボットに読み込む方法、Vizティーチン

グの通信設定を習得できます。



- Vizとの通信プログラムの読み込みは、Vizとの通信プログラムと設定ファイルをロボットシステムにロードすることで、ビジョン側とロボット側との通信を確立し、Mech-Mindソフトウェアシステムによるロボットの制御を実現します。
- それ以外のロボットを使用している場合は、[Vizティーチング通信](#)を参照してロボットの通信設定を行います。

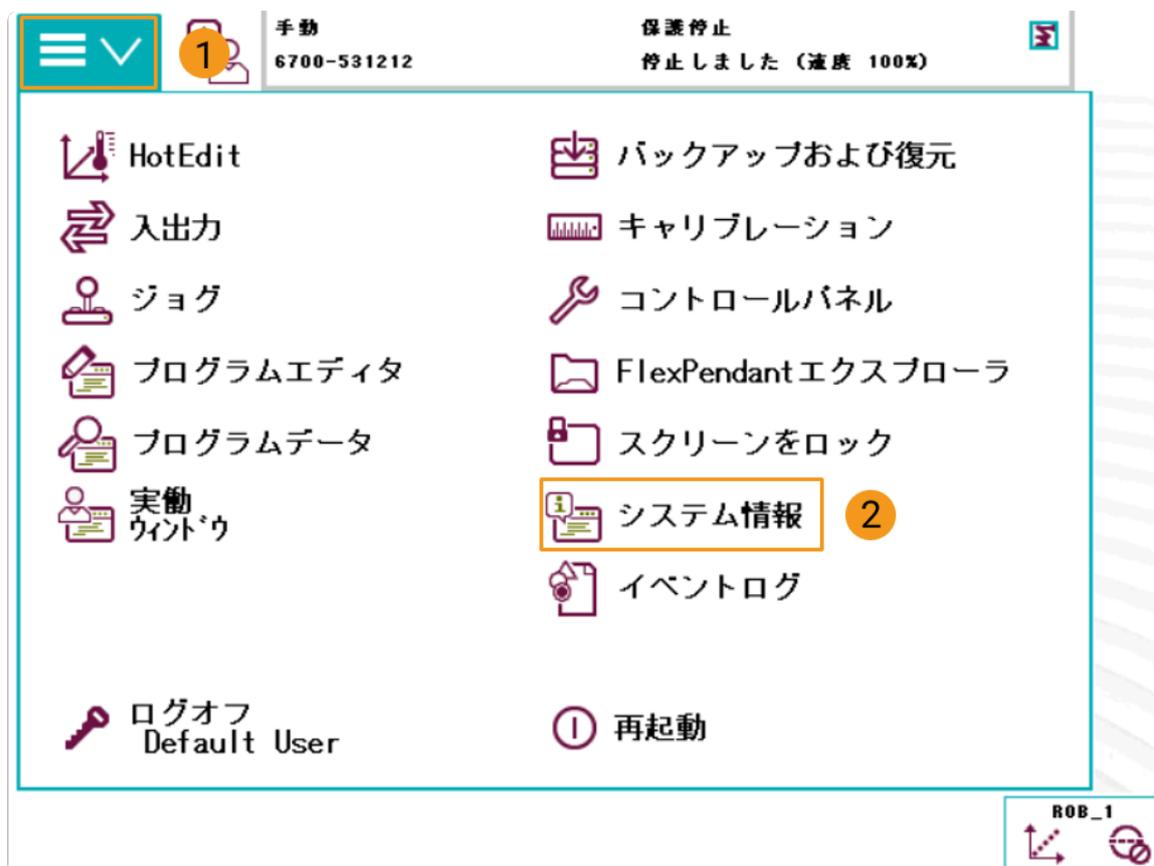
▶ <https://www.youtube.com/watch?v=-rJ2tTYvEJA> (YouTube video)

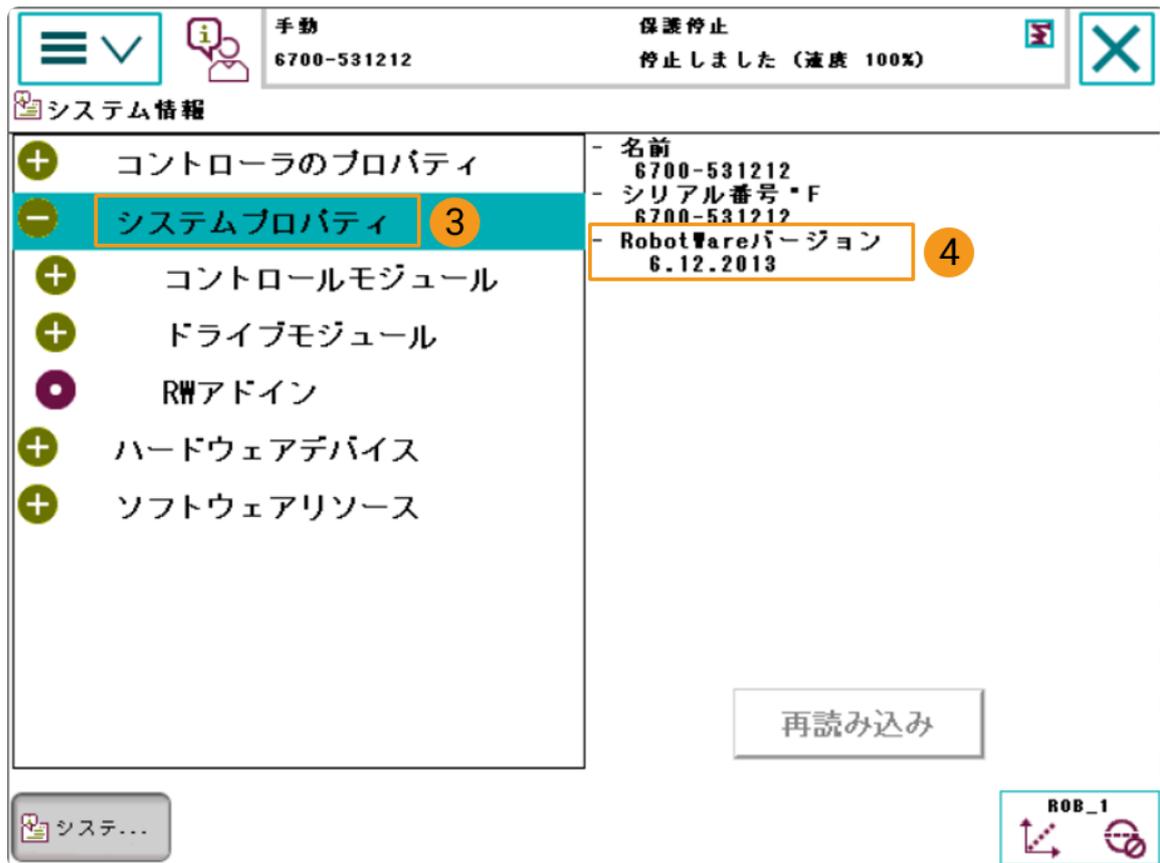
動画：ロボット通信設定 (Vizティーチング)

事前準備

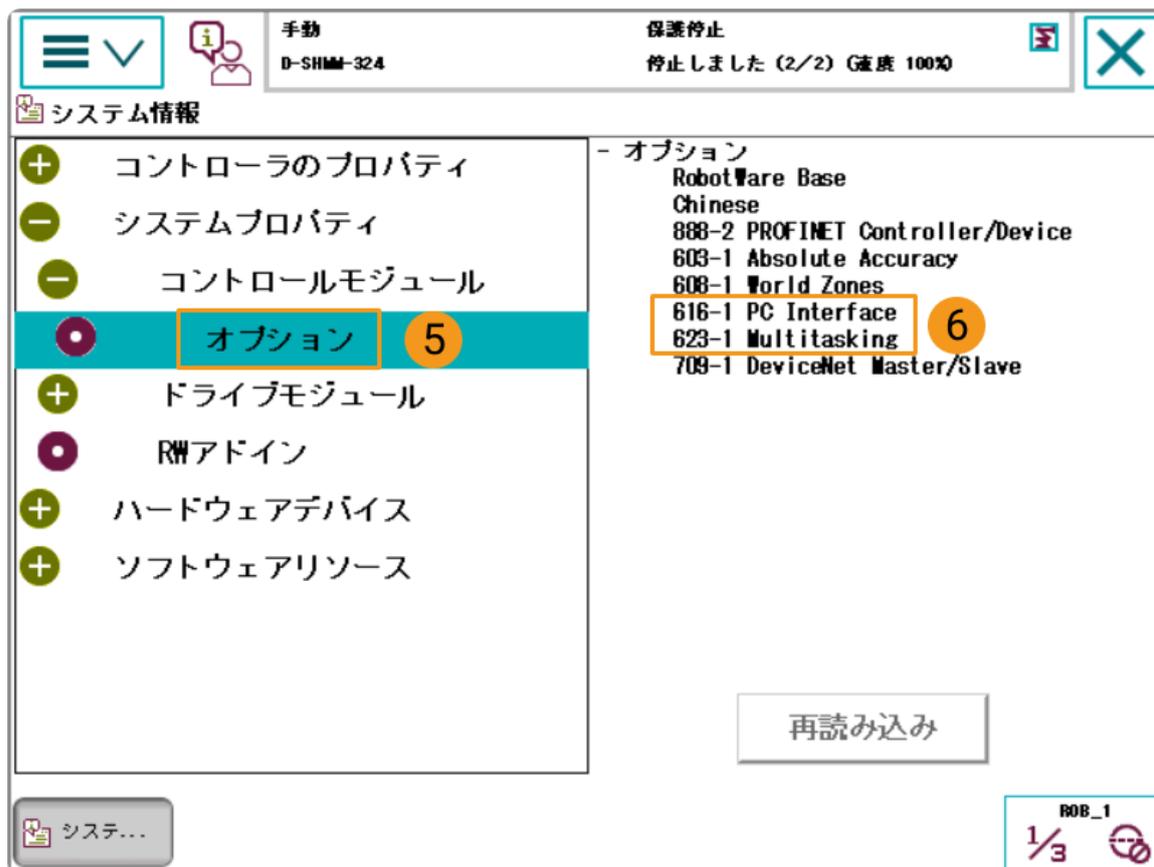
コントローラとRobotWareのバージョンを確認

1. コントローラにD652またはDSQC1030のIOボードが装着されていることを確認します。
2. ティーチペンダントでRobotWareのバージョンが6.0以上であることを確認します。





3. ティーチペンダントで、コントローラに以下のコントロールモジュールがインストールされていることを確認します。
- 623-1 Multitasking
 - 616-1 PCInterface



上記の条件を満たしていない場合は、Vizティーチング通信ができませんので、ロボットメーカーにお問い合わせください。

ロボットシステムのリセット

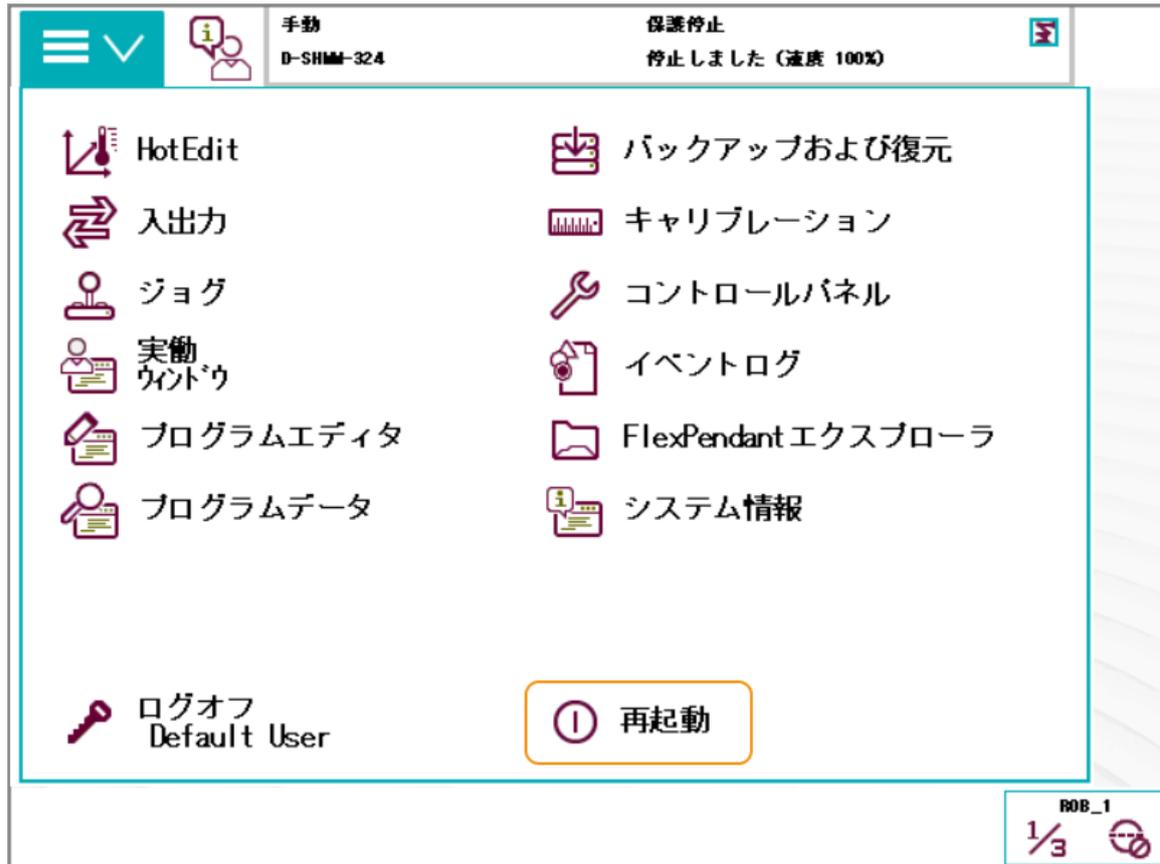
コントローラのハードウェアとソフトウェアの状態を確認した後、ロボットシステムのリセットが必要です。お使いのロボットは工場出荷時の状態である場合、ロボットシステムのリセットは不要です。



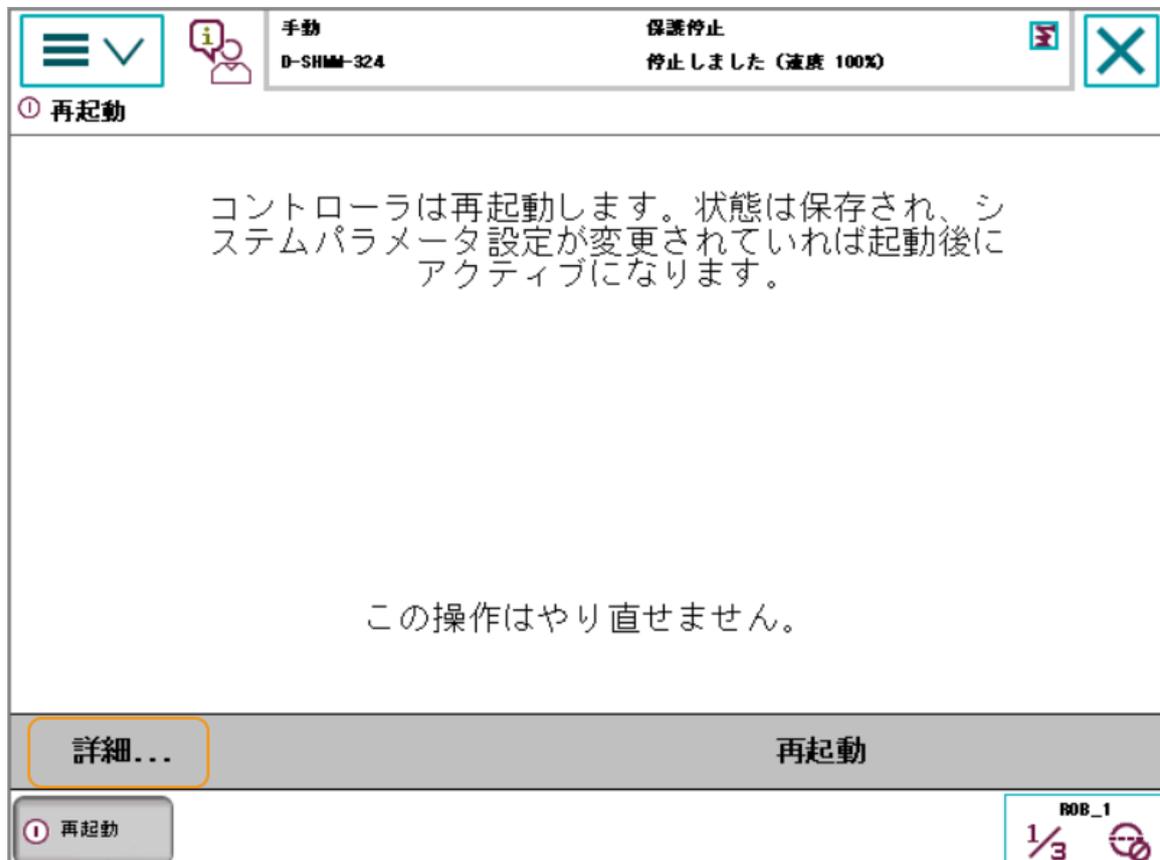
リセットすると工場出荷時の設定に戻りますので、バックアップ操作が完了していることを確認してください。

ロボットシステムをリセットするには、以下の手順を実行します。

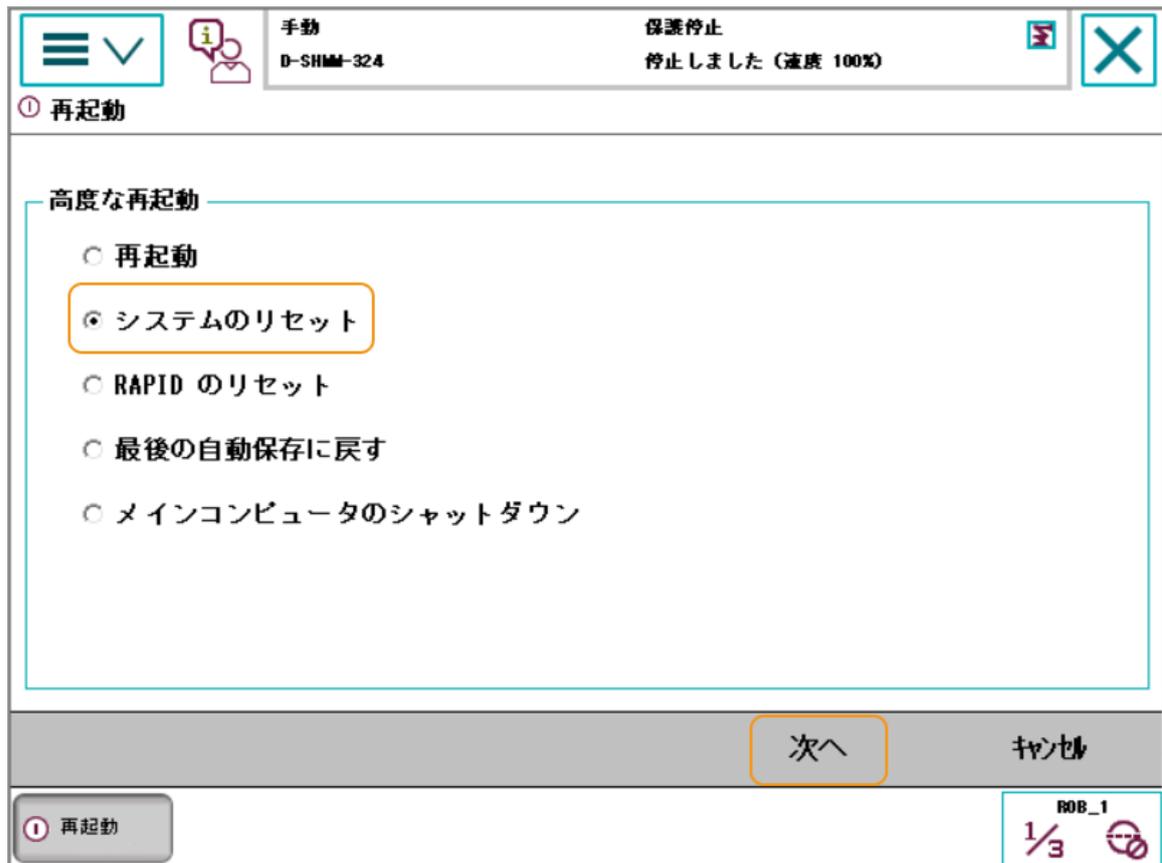
1. ティーチペンダントで、左上隅のメニューバーをクリックしてメイン画面を表示し、[再起動]をクリックします。



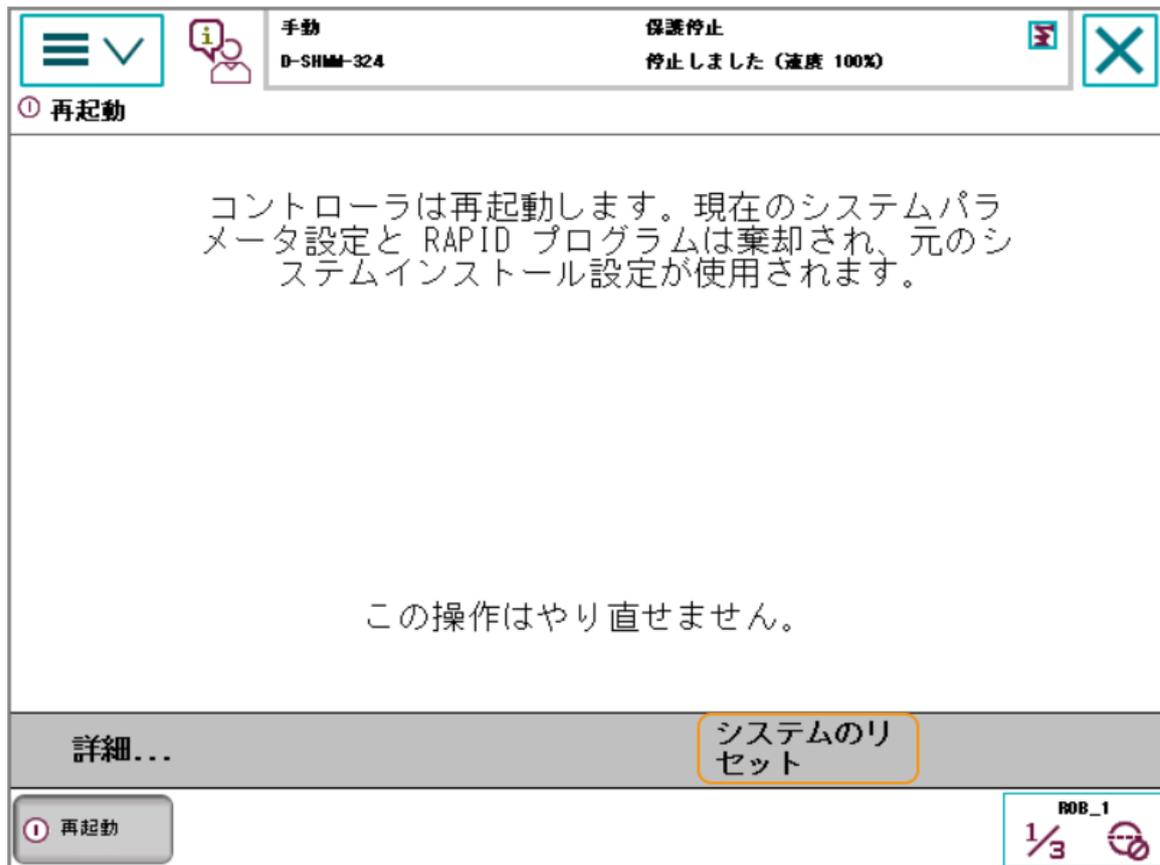
2. [詳細...]をクリックします。



3. システムのリセットを選択し、[次へ]をクリックします。



4. [システムのリセット]をクリックします。



リセットにかかる時間は1~2分程度です。再びメイン画面が表示されると、リセットは完了します。

ネットワーク接続

1. IPCのLANケーブルのもう一端を、下図のようにロボットコントローラの **X6 (WAN)** LANポートに接続します。



2. ABBロボットのIPアドレスが、IPCのIPアドレスと同じネットワークセグメントにあることを確認します。

読み込みファイルの準備

1. IPCで、Mech-Mindソフトウェアシステムのインストールディレクトリに格納されている

Mech-Center/Robot_Server/Robot_FullControl/abb/server on ABB フォルダを開きます。

- このフォルダをコピーしてUSBメモリに貼り付け、USBメモリをRobotStudioソフトウェアがインストールされているPCに差し込みます。



RobotStudioは、ABB製ロボットのシミュレーションおよびオフラインプログラミングソフトウェアです。IPCや他のPCにインストールすることができます。この例では、RobotStudioは別のPCにインストールされています。

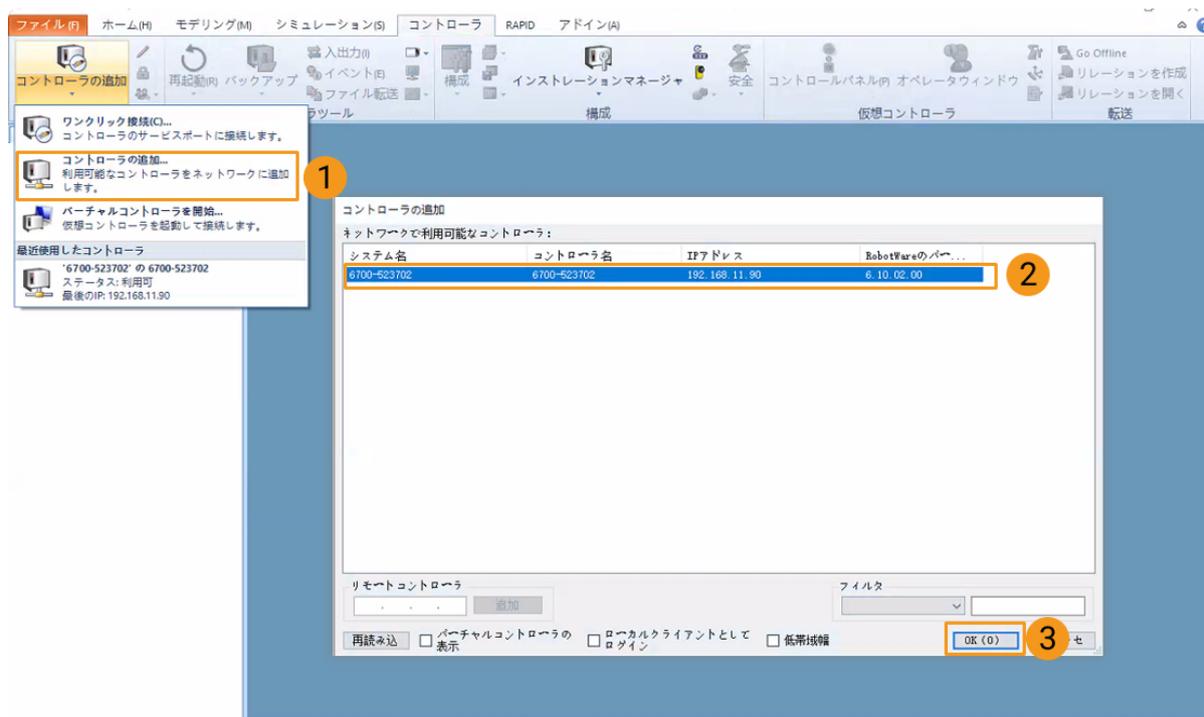
ファイルの説明は以下の通りです：

- フォルダ「MM」：ロボットプログラムモジュール。
- 「config」：ロボットの設定ファイル。
 - D652 IOボードを現場で使用する場合は、D652.cfgとSYS.cfgのファイルが必要です。
 - DSQC1030 IOボードを現場で使用する場合は、DSQC1030.cfgとSYS.cfgのファイルが必要です。
 - D652 IOまたはDSQC1030 IOボードを現場で使用しない場合は、EIO.cfgとSYS.cfgが必要です。

ロボットプログラムの読み込み

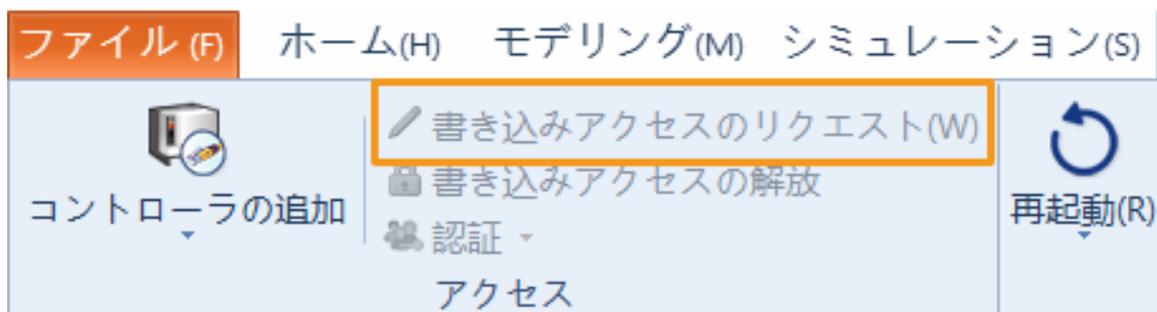
RobotStudioの起動とコントローラの接続

RobotStudioのコントローラメニューをクリックし、ツールバーで**コントローラの追加**、**コントローラの追加**を選択します。表示される**コントローラの追加**画面で、コントローラを選択して[OK]をクリックします。



ロボットの書き込みアクセスの取得

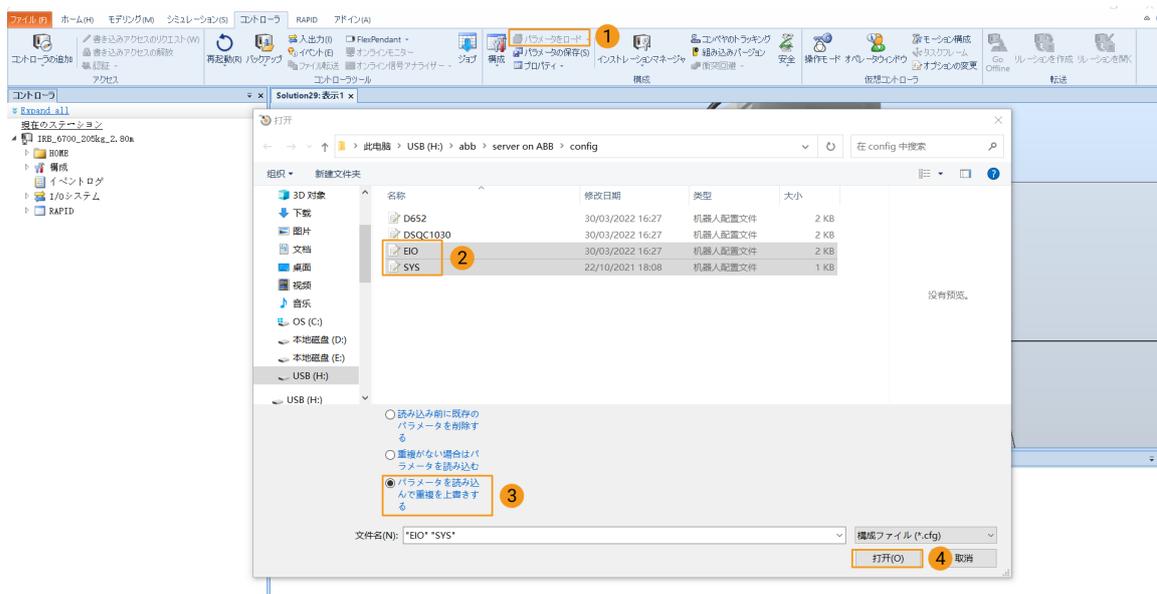
1. RobotStudioのツールバーで書き込みアクセスのリクエストをクリックします。



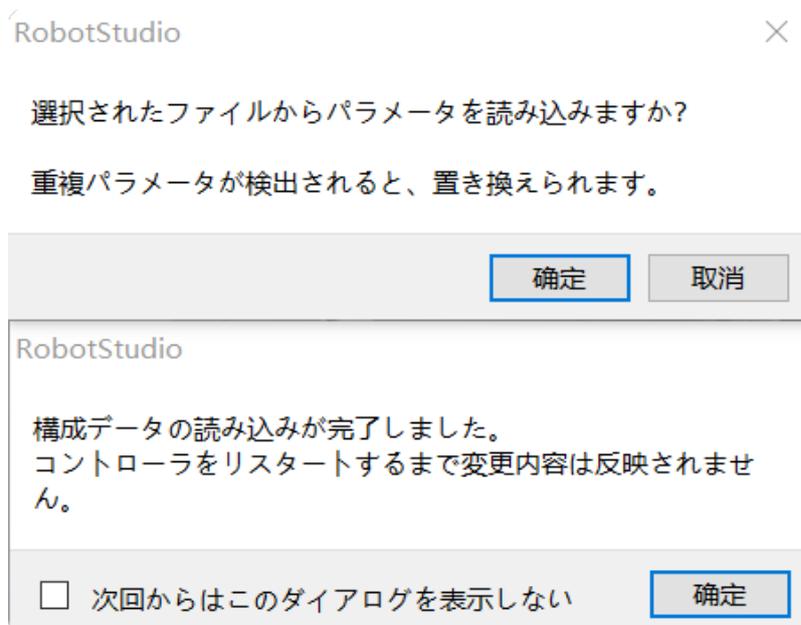
2. ティーチペンダントで表示される書き込みアクセスのリクエスト画面で[同意]をクリックします。

ロボットの設定ファイルの読み込み

1. RobotStudioでコントローラのメニューをクリックし、ツールバーでパラメータをロードを選択し、USBメモリでロードする設定ファイルを選択し、パラメータを読み込んで重複を上書きするを選択してから、[開く]をクリックします。

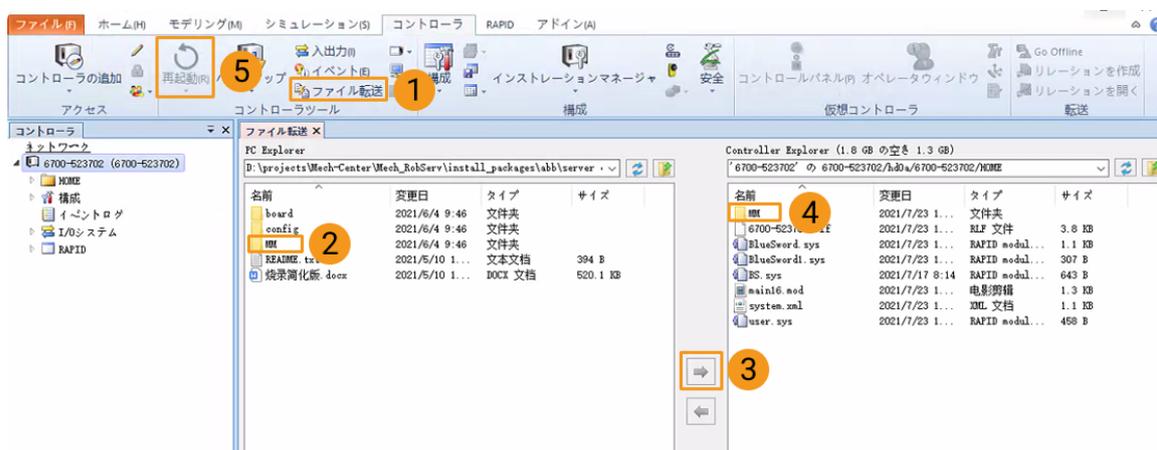


2. ポップアップするダイアログで[OK]ボタンを連続してクリックします。



ロボットプログラムモジュールのロード

1. RobotStudioでコントローラメニューをクリックし、ツールバーでファイル転送をクリックします。ファイル転送画面の左側パネルで「MM」フォルダを選択し、転送ボタンをクリックしてロボットシステムのHOMEパスに転送します。



2. RobotStudioでコントローラメニューをクリックし、ツールバーで再起動をクリックしてロボットシステムを再起動します。

これで、Vizとの通信プログラムとコンフィグファイルはロボットに読み込まれています。

Vizティーチング通信が有効であるかをテスト

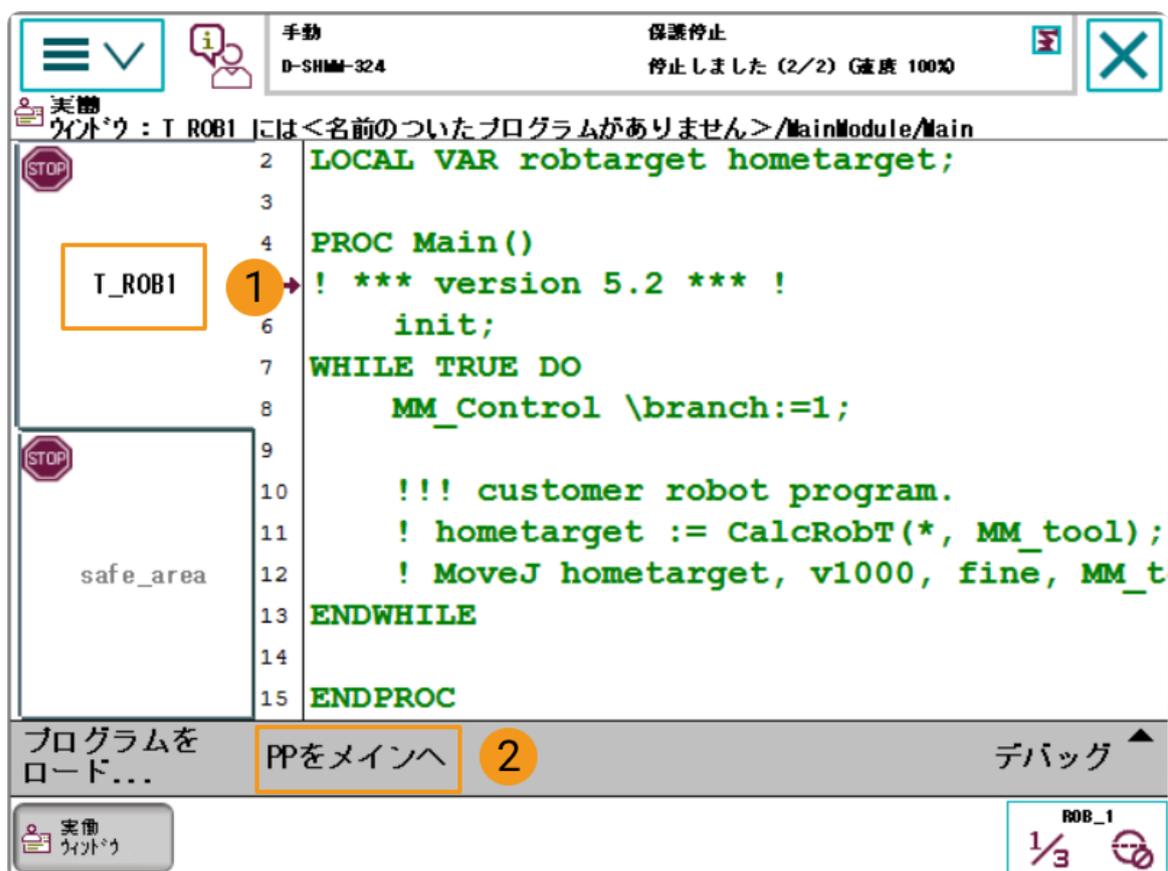
ロボットシステムが再起動後、Vizによりロボットをティーチングするかどうかをテストするには、以下の手順を実行します。

ロボットの自動モードへの切り替え

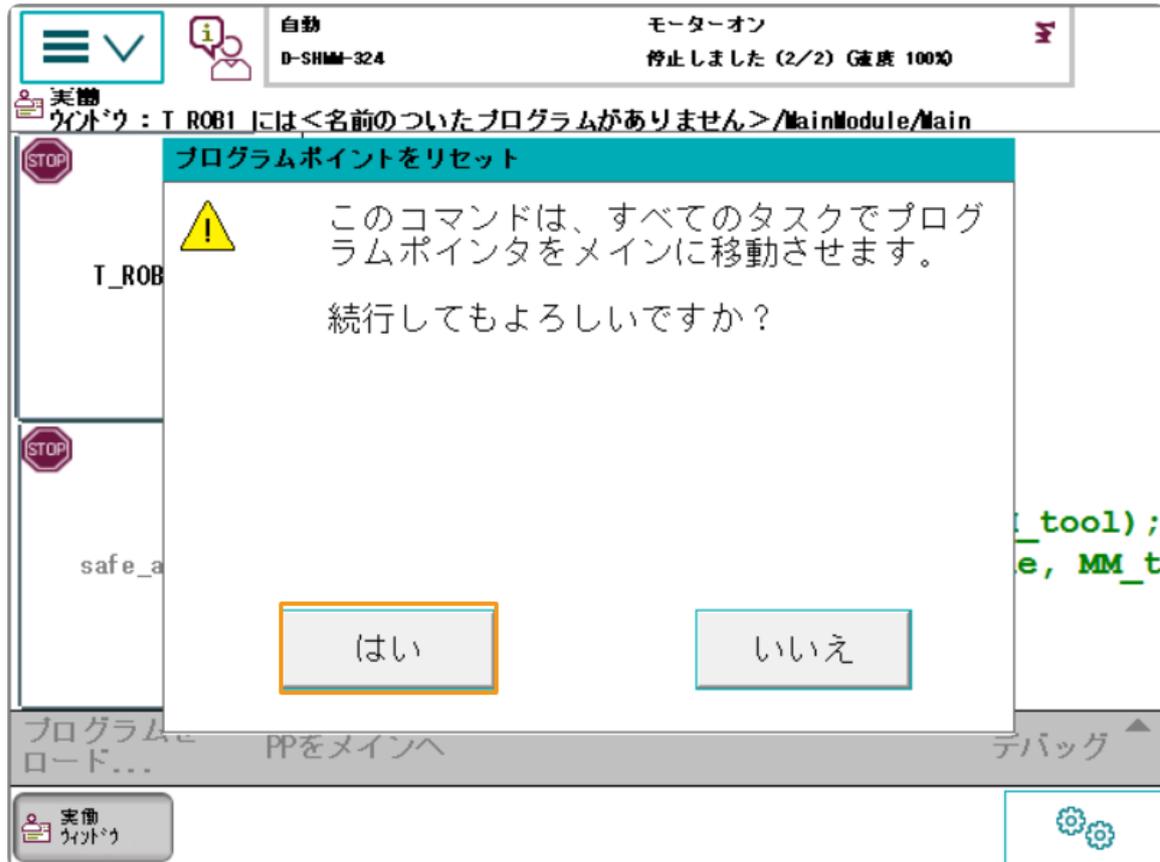
1. ロボットコントローラでは、キースイッチによってロボットを自動モードに切り替えます。
2. ティーチペンダントでポップアップするダイアログで[OK]をクリックします。
3. ロボットコントローラで、パワーオンのボタンを押してロボットの電源を入れます。このボタンは、電源投入に成功すると必ず点灯します。

メインプログラムの実行

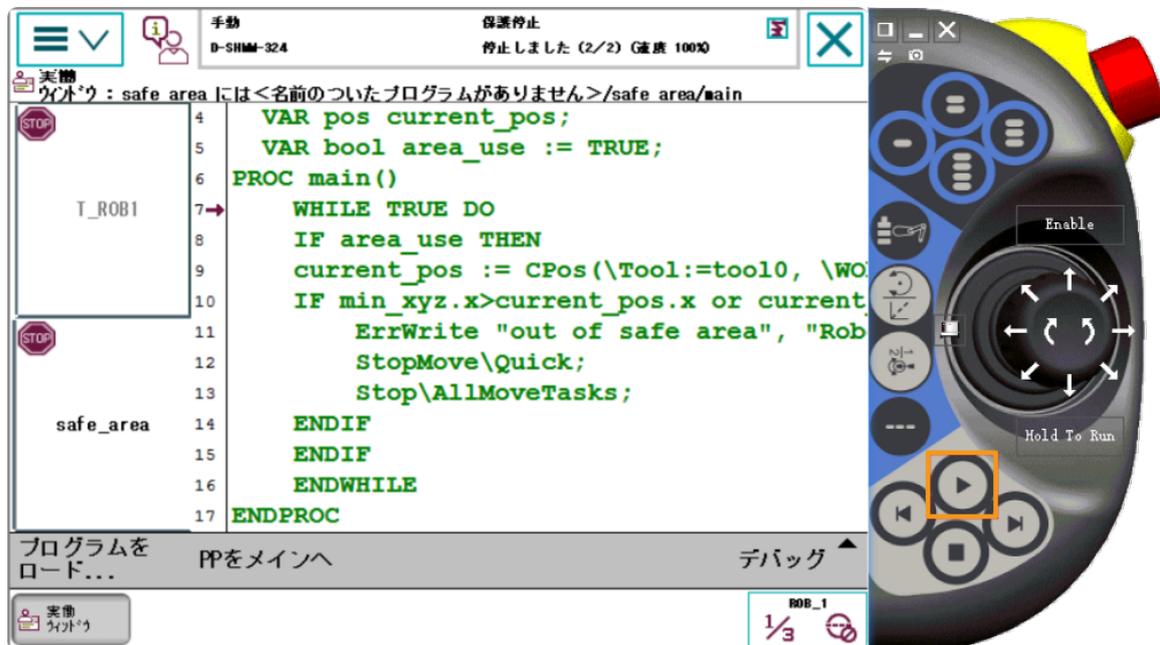
1. ティーチペンダントで、T_ROB1のプログラムポインターをメインプログラムに移動してから、[PPをメインへ]をクリックします。



2. ポップアップするダイアログで[はい]をクリックします。

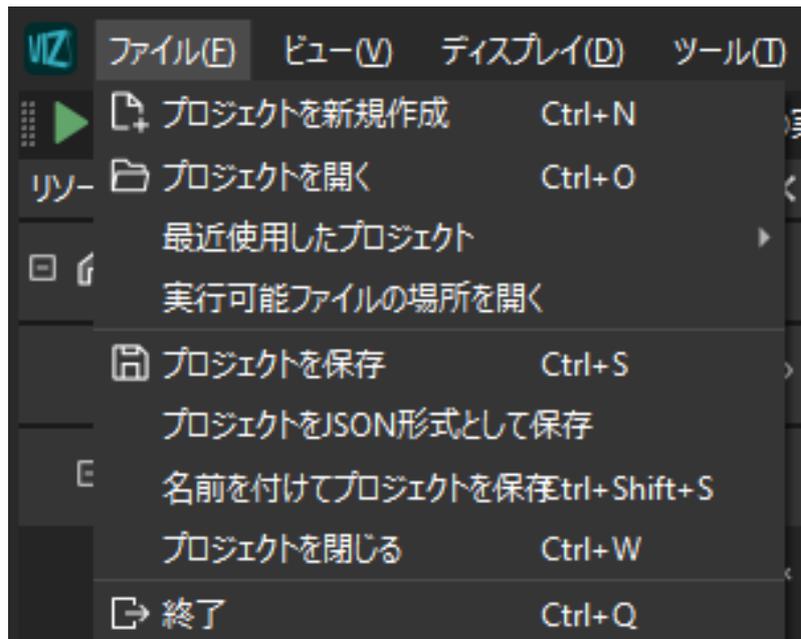


3. ティーチペンダントの右側の実行ボタンを押します。

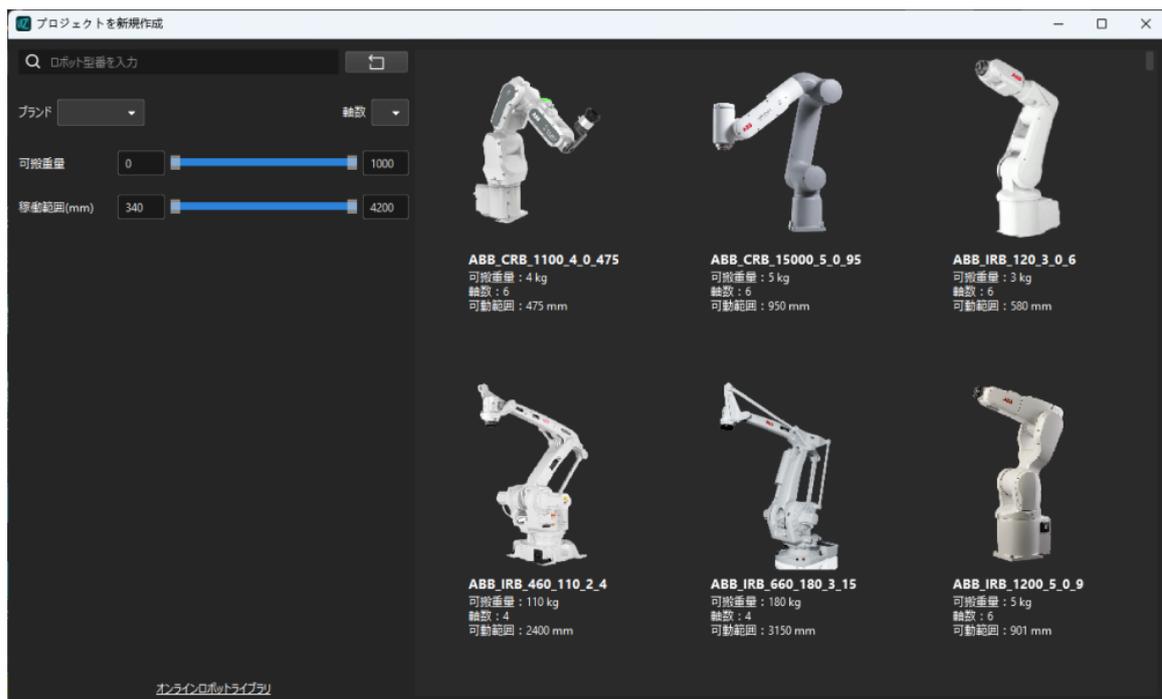


Mech-Vizプロジェクトの作成

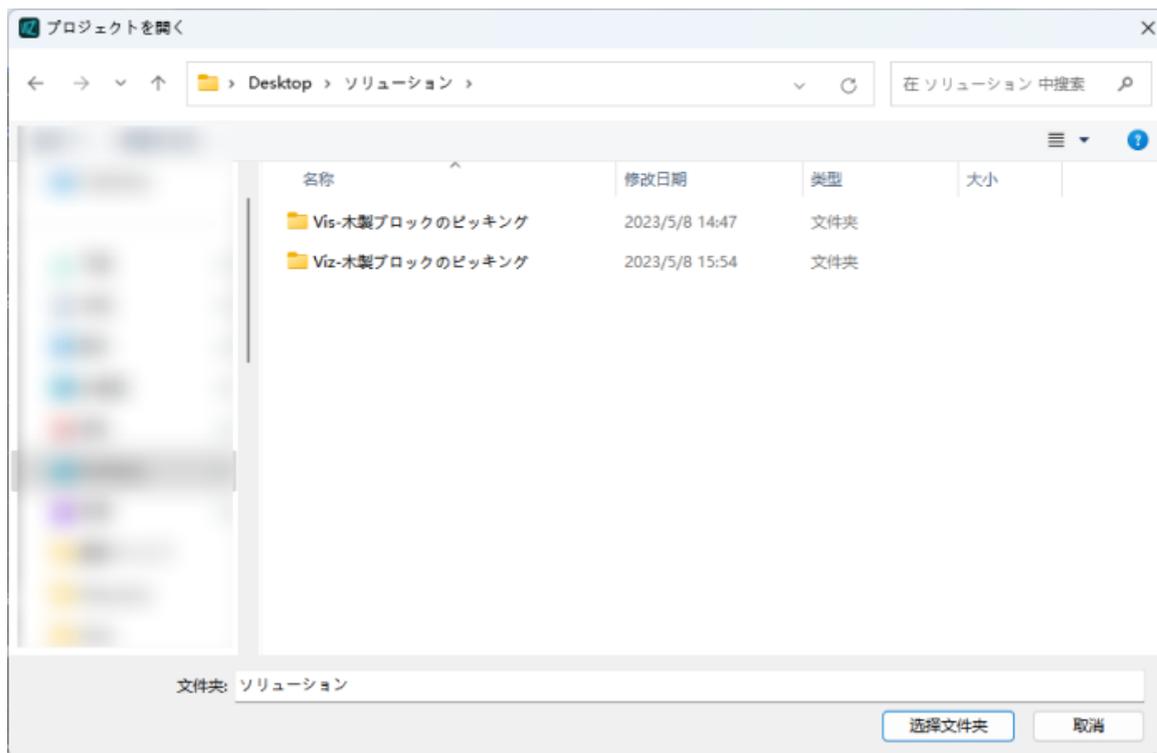
1. Mech-Vizを起動し、**ファイル** > **プロジェクトを新規作成**を選択します。



2. ロボットブランドを「ABB」に設定し、右側のパネルで「ABB_IRB_1300_11_0_9」を選択してから[選択]をクリックします。

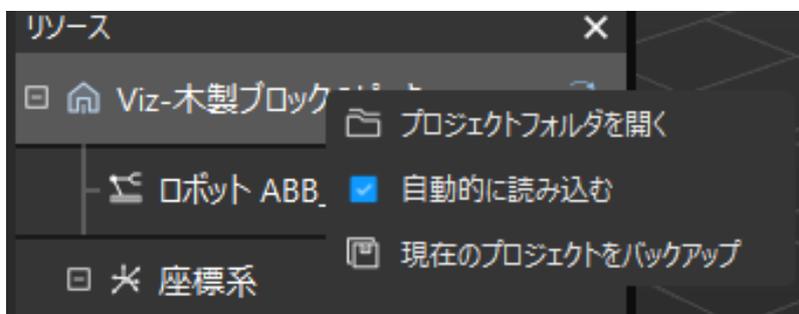


3. キーボードで **Ctrl + S** を押し、「Viz-ワークピッキング」といったフォルダを新規作成して選択し、[フォルダの選択]をクリックします。



Mech-Vizプロジェクトが正常に保存されると、リソースパネルにプロジェクト名が「Viz-ワークピックアップ」と表示されるようになります。

4. プロジェクト名を右クリックし、自動的に読み込むにチェックを入れます。



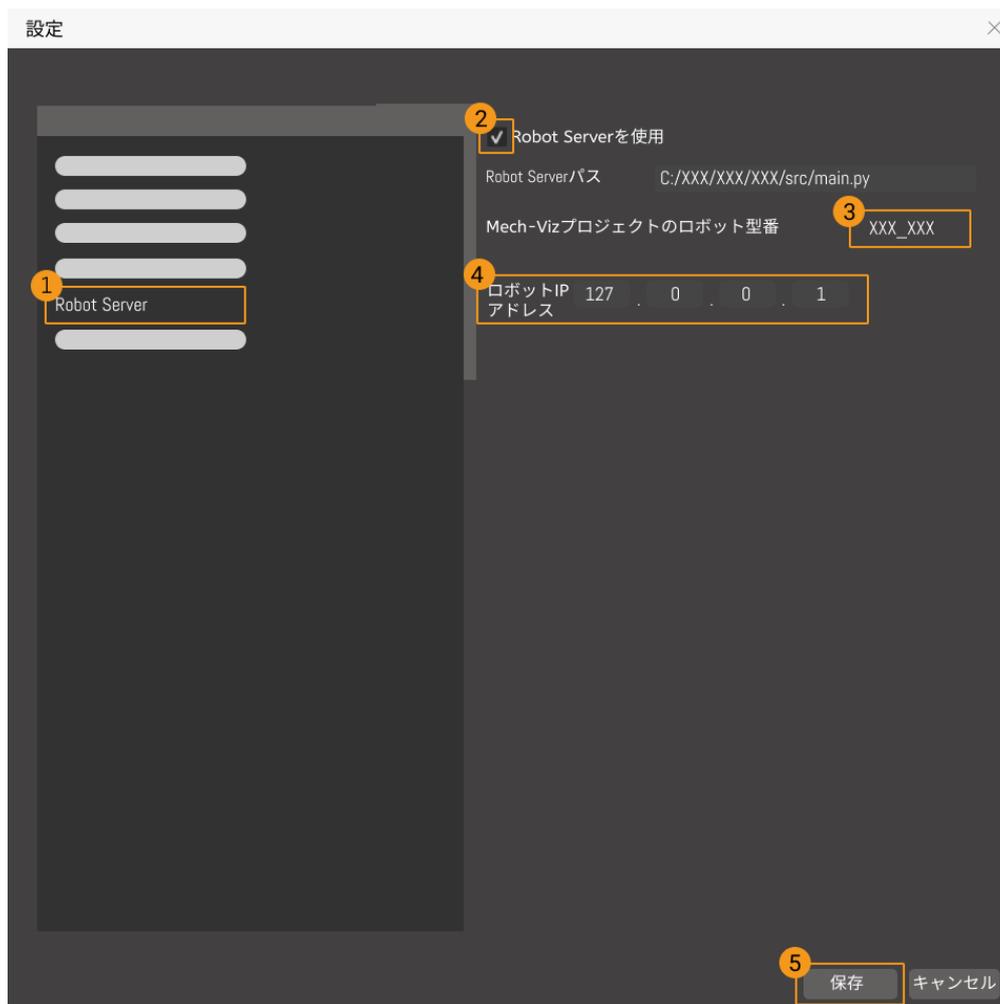
1. ロボットを安全に動かすために、ツールバーのパラメータ **速度** と **加速度** を5%など小さい値に設定します。



2. キーボードで **Ctrl** + **S** を押してプロジェクトを保存します。

Mech-CenterでVizティーチング通信の関連設定を実行

1. Mech-Centerを起動し、**設定 > Robot Server**を選択し、**Robot Serverを使用**にチェックを入れます。
2. ロボット型番がロボット実機の型番と同じであることを確認し、**ロボットIPアドレス**をロボット実機のIPアドレスに設定し、**[保存]**をクリックします。



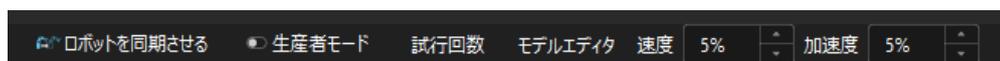
ロボットを接続

Mech-Centerのツールバーで **ロボットを制御**  をクリックします。

- 接続に成功すると、サービスステータスバーにロボットのアイコンと型番が表示され、ログバーにロボット接続成功のメッセージが出力されます。
- 接続に失敗した場合は、前の操作が正しかったかどうか再確認してください。

ロボットを移動

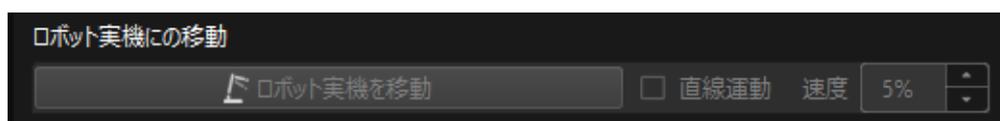
1. Mech-Vizのツールバーで**ロボットを同期させる**をクリックし、仮想ロボットの位置姿勢とロボット実機の位置姿勢を同期させ、再度クリックして**ロボットを同期させる**の選択を解除します。



2. **ロボット**パネルの関節角度のオプションで、J1関節角度を調整（例えば、0°から3°に調整）すると、仮想ロボットが動くようになります。



3. [ロボット実機を移動]をクリックします。



ロボットを操作する際は、人の安全を確保してください。緊急時には、ティーチペダントの緊急停止ボタンを押してください！

ロボット実機が仮想ロボットの位置姿勢に移動することができたら、Vizティーチング通信が有効になります。

2.3. ロボットハンド・アイ・キャリブレーション

本ガイドを読むことで、自動ハンド・アイ・キャリブレーション（Eye to Hand）の実行方法を習得できます。



ハンド・アイ・キャリブレーションとは、カメラ座標系とロボット座標系との対応関係（外部パラメータ）を求めることです。ビジョンシステムにより取得した対象物の位置姿勢をロボット座標系に変換し、ロボットが対象物の把持を完了させるように制御します。

▶ <https://www.youtube.com/watch?v=fmiMzPR0lGU> (YouTube video)

動画：ハンド・アイ・キャリブレーション (Vizティーチング)

事前準備

本節では、キャリブレーションボードの取り付け、カメラパラメータの調整、キャリブレーション前の設定を行う必要があります。

キャリブレーションボードの取り付け



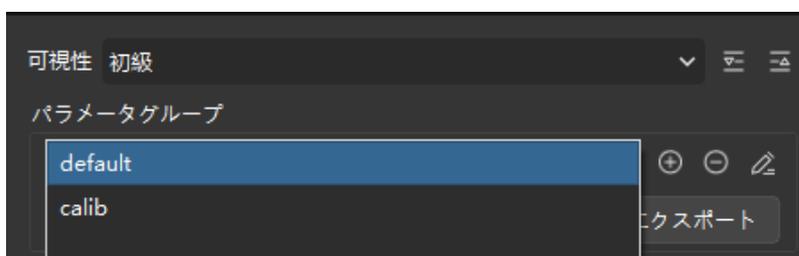
Eye to Handモードでは、キャリブレーションボードはロボット先端のフランジに取り付ける必要があります。

以下の手順を実行します。

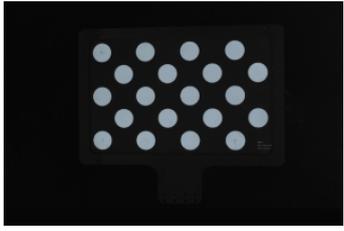
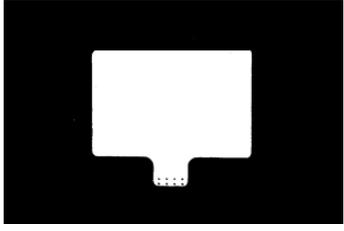
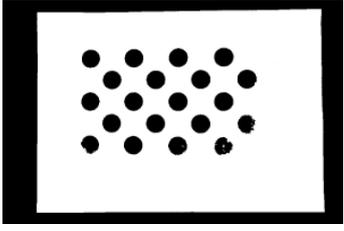
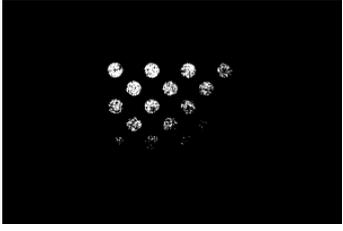
1. カメラ梱包箱からキャリブレーションボードとフランジを取り出します。
2. ネジ、スペーサー、ナットを使って、フランジをロボット先端に固定します。
3. ネジ、スペーサー、ナットを使って、キャリブレーションボードをフランジに固定します。
4. 取り付け後、ロボットを作業領域の最下層のワークの上面、カメラの視野中心に移動させます。

カメラパラメータを調整

1. Mech-Eye Viewerでカメラを接続し、**パラメータグループ**を「calib」に設定します。



2. 2D画像内のキャリブレーションボードがはっきり見え、露光過度や露光不足がないように、2Dパラメータを調整します。
3. キャリブレーションボード上の白い円の点群が完全なものになるように、3Dパラメータを調整します。点群の変動範囲を減らすために、**点群後処理の表面平滑化と外れ値除去**を**Normal**に変更することをお勧めします。

	正常	露光過度	露光不足
2D画像			
点群			

Mech-Visionソリューションの作成と保存

1. Mech-Visionを起動すると、下図のようなようこそ画面が表示され、Mech-Vision が正常に起動されたことを示します。



2. Mech-Visionのようこそ画面で **[ソリューションライブラリから新規作成]** をクリックし、ソリューションライブラリを開きます。



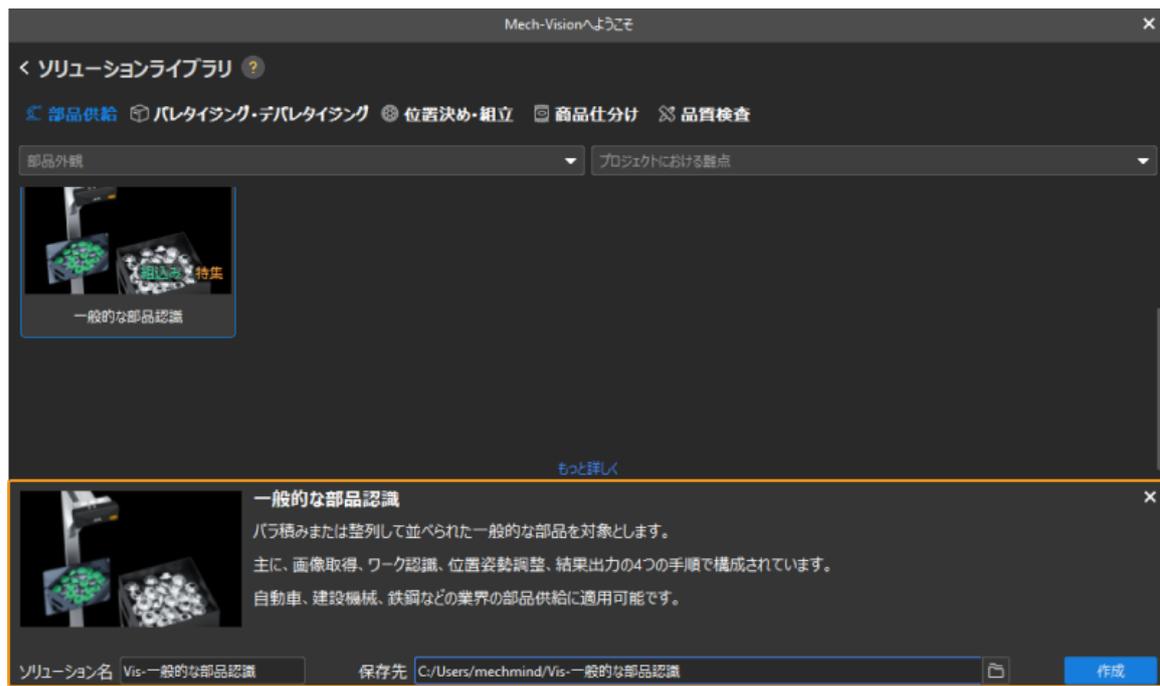
ソリューションライブラリには、異なる業界に適応可能なソリューションやプロジェクトのソースライブラリが含まれています。

3. ソリューションライブラリを開いた画面を下図に示します。一般的な部品認識プロジェクトを選択します。



ソリューションライブラリからこのプロジェクトが見つからない場合、画面下部にある [もっと詳しく] をクリックします。

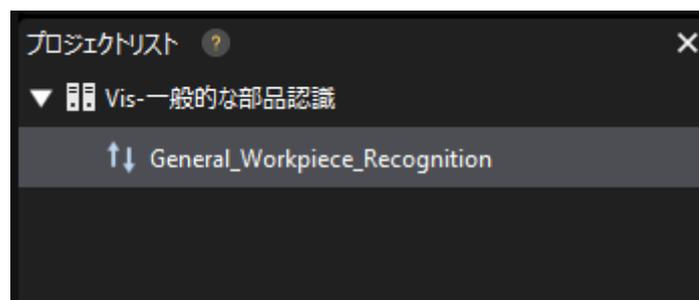
4. このプロジェクトをクリックして選択すると、プロジェクトの関連情報は画面下部に表示されます。ソリューション名と保存先を設定したら、[作成] をクリックします。



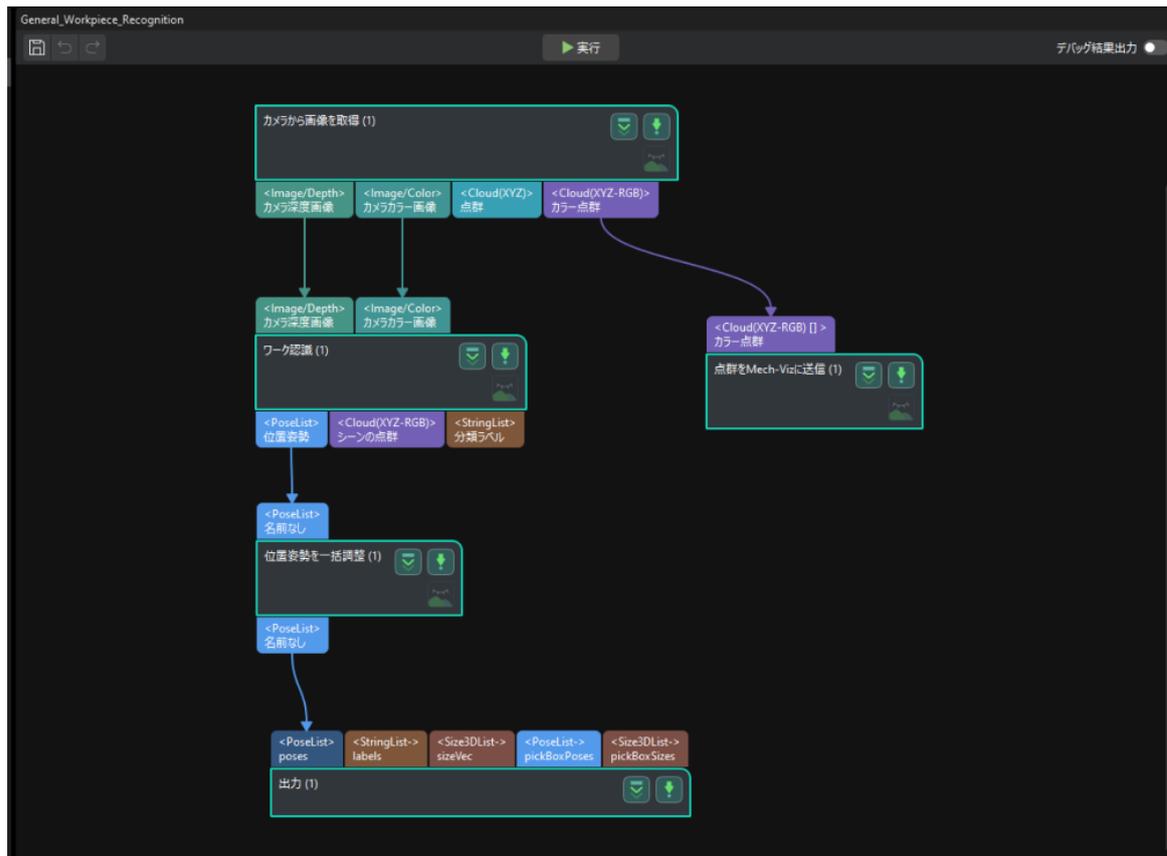
プロジェクトが作成されると、Mech-Visionメイン画面の左上にあるプロジェクトリストに、作成されたソリューションとプロジェクトが表示されます。



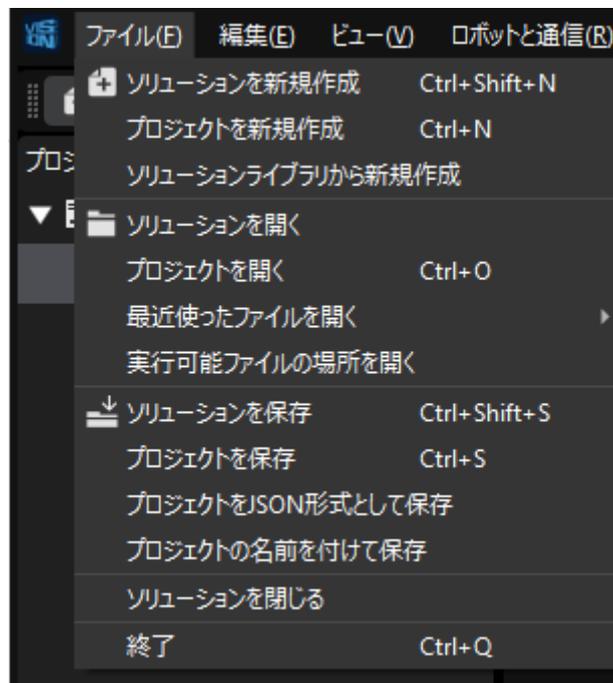
- **ソリューション**とは、ビジョンソリューションを実現するために必要なロボットと通信、ビジョン処理、経路計画などの機能設定とデータの集りのことです。
- **プロジェクト**は、ソリューションにおけるビジョン処理のワークフローです。通常、1つのプロジェクトを含むソリューションで十分ですが、複雑な適用シーンでは複数のプロジェクトが必要になる場合があります。本ガイドに使用されるソリューションでは、1つのプロジェクトしか必要ありません。



メイン画面中央のプロジェクト編集エリアには、「一般的な部品認識」というプロジェクトが表示されます。



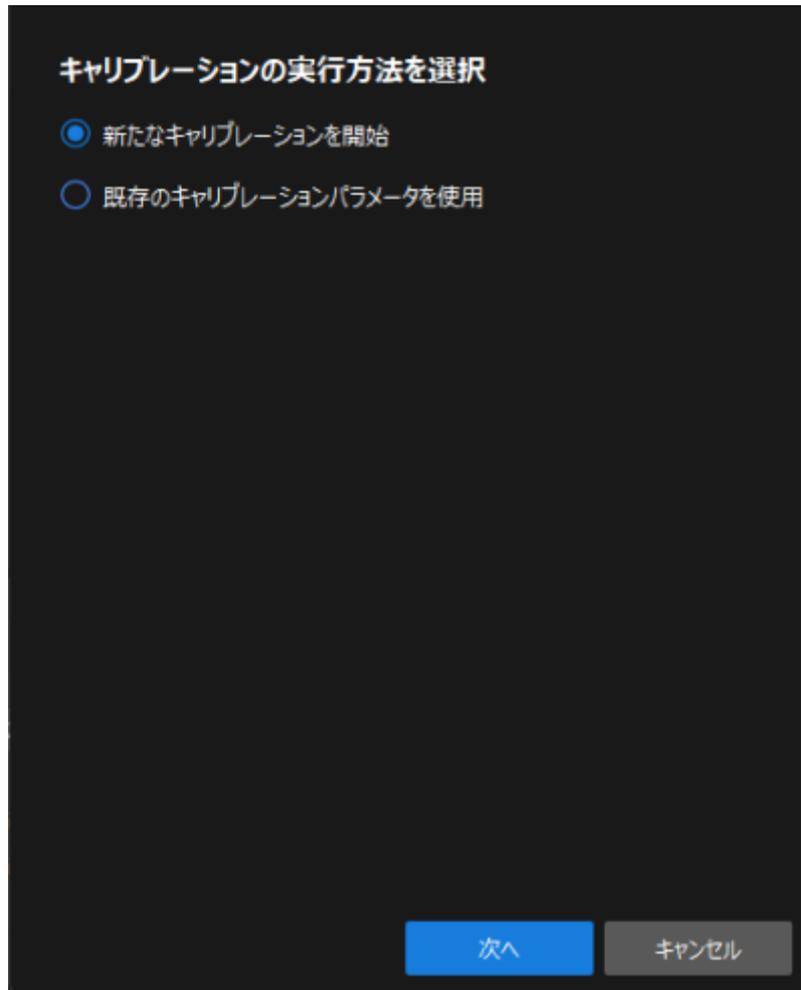
5. メニューバーから**ファイル** > **ソリューションを保存**を選択します。



キャリブレーション前の設定

1. Mech-Visionのツールバーで[**カメラキャリブレーション (標準モード)**]をクリックします。すると、**キャリブレーション前の設定画面**が表示されます。

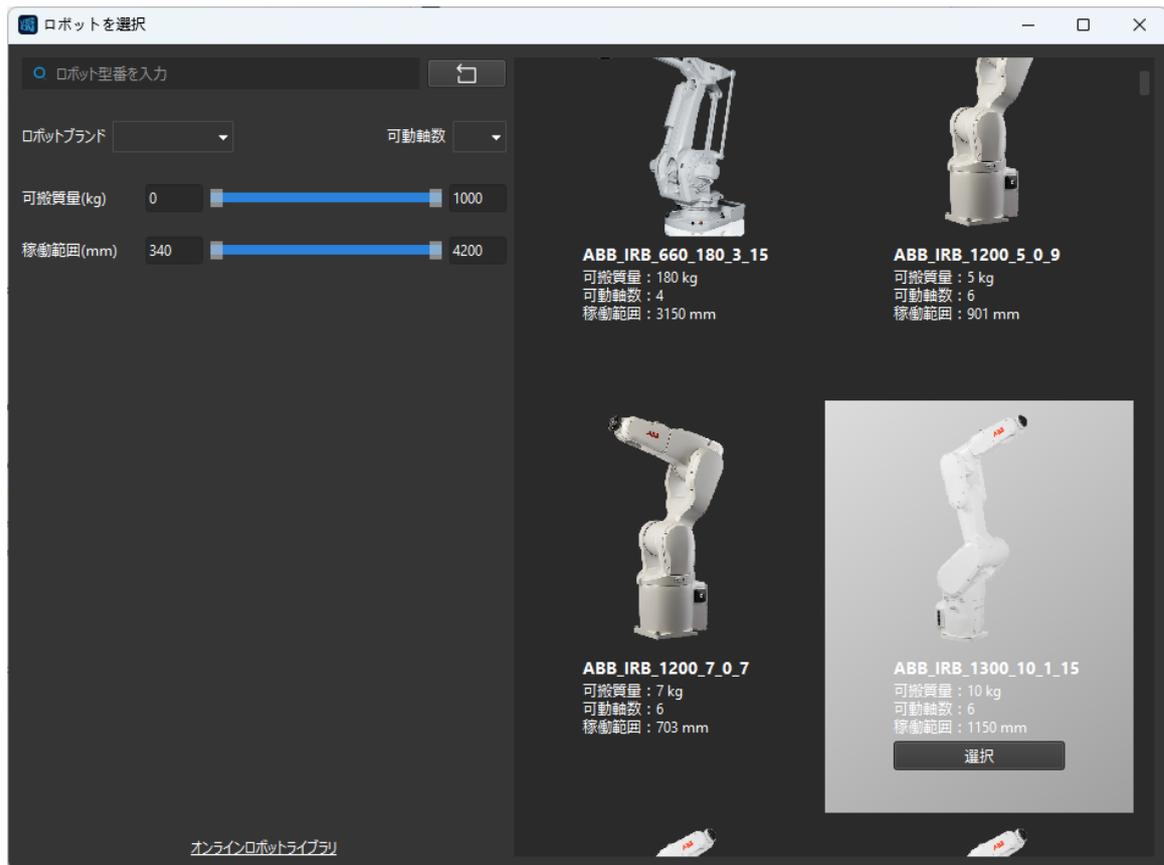
2. キャリブレーションの実行方法を選択画面で、**新たなキャリブレーションを開始**を選択し、[次へ]をクリックします。



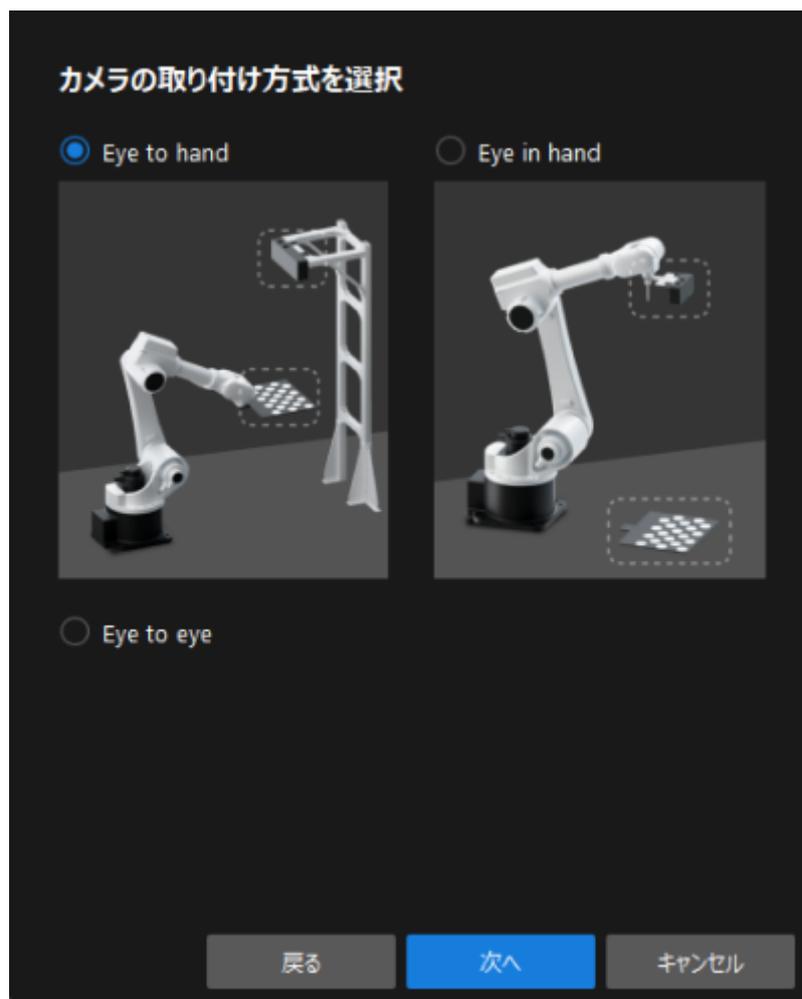
3. キャリブレーションのタスクを選択画面で、ドロップダウンリストボックスから**適応可能なロボットのハンド・アイ・キャリブレーション**を選択し、[ロボット型番を選択]をクリックします。



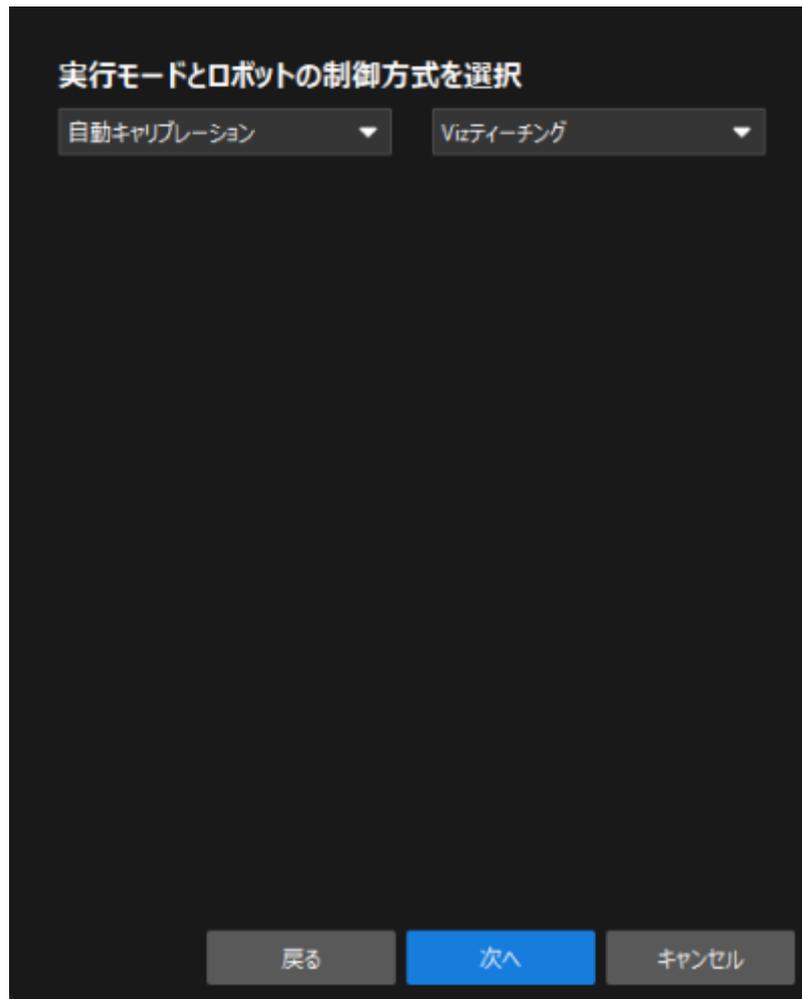
4. ロボットブランドのドロップダウンリストボックスから「ABB」を選択し、右側で「ABB_IRB_1300_11_0_9」を選択して[選択]をクリックし、[次へ]をクリックします。



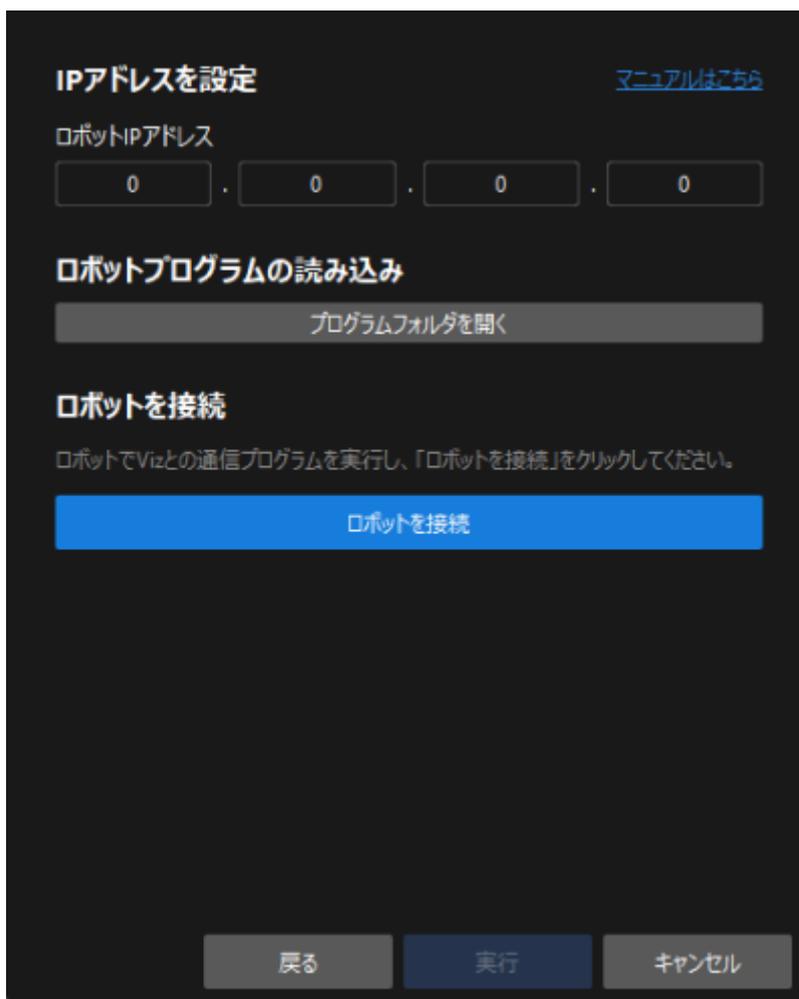
5. カメラの取り付け方式を選択画面で、Eye to Handを選択して[次へ]をクリックします。



6. 実行モードとロボットの制御方式を選択画面で、自動キャリブレーションとVizティーチングを選択して[次へ]をクリックします。



7. 通信設定画面で、ロボットIPアドレスをロボット実機のIPアドレスに設定します。



8. ティーチペンダントで、[MMメインプログラムが起動されていること](#)を確認します。
9. Mech-Visionに戻り、**ロボットを接続**エリアの下に[**ロボットを接続**]をクリックします。このボタンが**ロボットの接続を待機中...**に変わります。
10. **ロボットを接続**の下で「接続済み」が表示されることを確認してから[**実行**]をクリックします。すると、**キャリブレーション (Eye to Hand)** 画面が表示されます。

キャリブレーションの実行手順

カメラの接続

1. **カメラに接続手順**で、**検出されたカメラ**から接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。



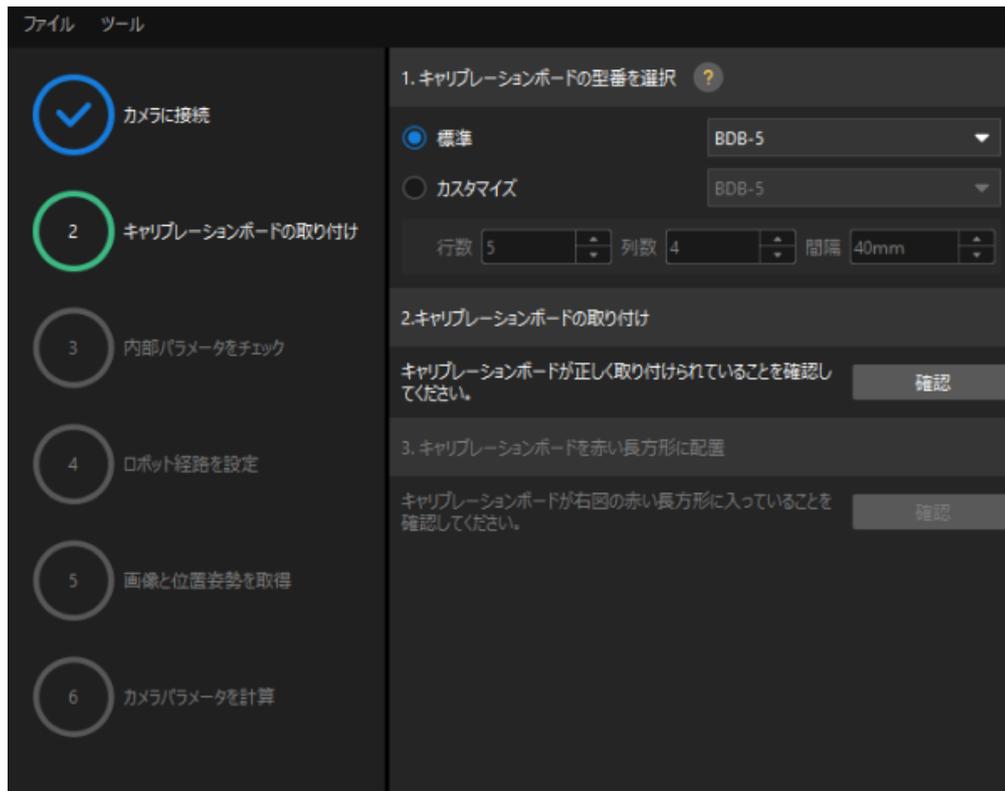
2. カメラ接続後、[一回キャプチャ]または[連続キャプチャ]をクリックします。
3. 画像ビューでは、取得した2D画像と深度画像が要件を満たしていることを確認した上で[次へ]をクリックします。



取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露光パラメータを調整](#)して再取得する必要があります。

キャリブレーションボードの取り付け

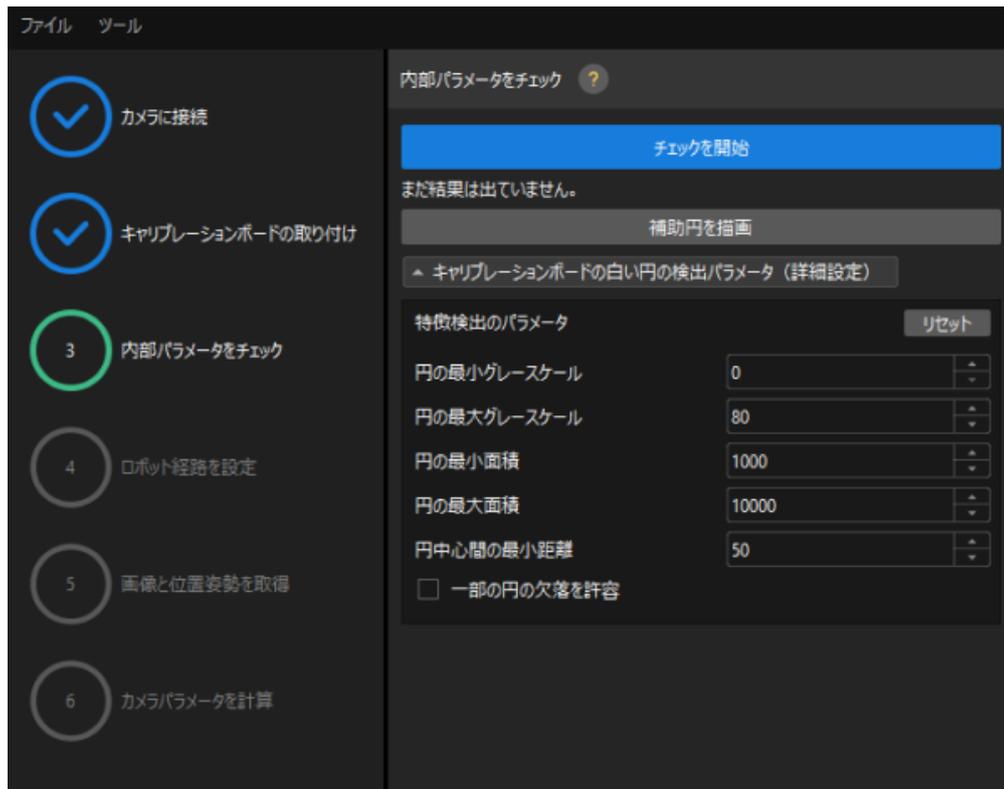
1. キャリブレーションボードの取り付け手順で、**1. キャリブレーションボードの型番を選択で標準**を選択し、実際に使用されるキャリブレーションボードに貼られているラベルに従って型番を選択します。
2. キャリブレーションボードをロボットフランジにしっかりと取り付けたら、**2. キャリブレーションボードの取り付けで[確認]**をクリックします。
3. キャリブレーションボードがカメラの視野中央（赤い長方形内）に配置されることを確認したら、**3. キャリブレーションボードを赤い長方形に配置で[確認]**をクリックします。



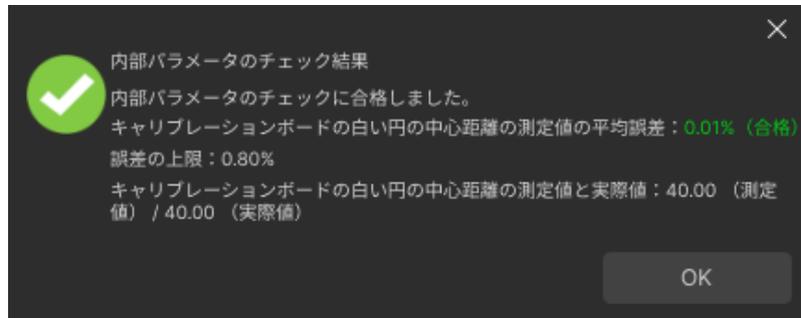
4. キャリブレーションボードに関するすべての操作が完了したら、下部にある[次へ]をクリックします。

カメラの内部パラメータをチェック

1. 内部パラメータをチェック手順で[チェックを開始]をクリックします。



2. カメラの内部パラメータのチェックが合格したことを確認したら、下部にある[次へ]をクリックします。



内部パラメータのチェックが合格しなかった場合、補助円を描画するか、検出パラメータを編集してください。

ロボット経路を設定

1. **ロボット経路を設定**手順で、**高さ範囲**を設定します。このパラメータは、キャリブレーションボードが深度方向における移動可能な範囲に応じて設定する必要があります。



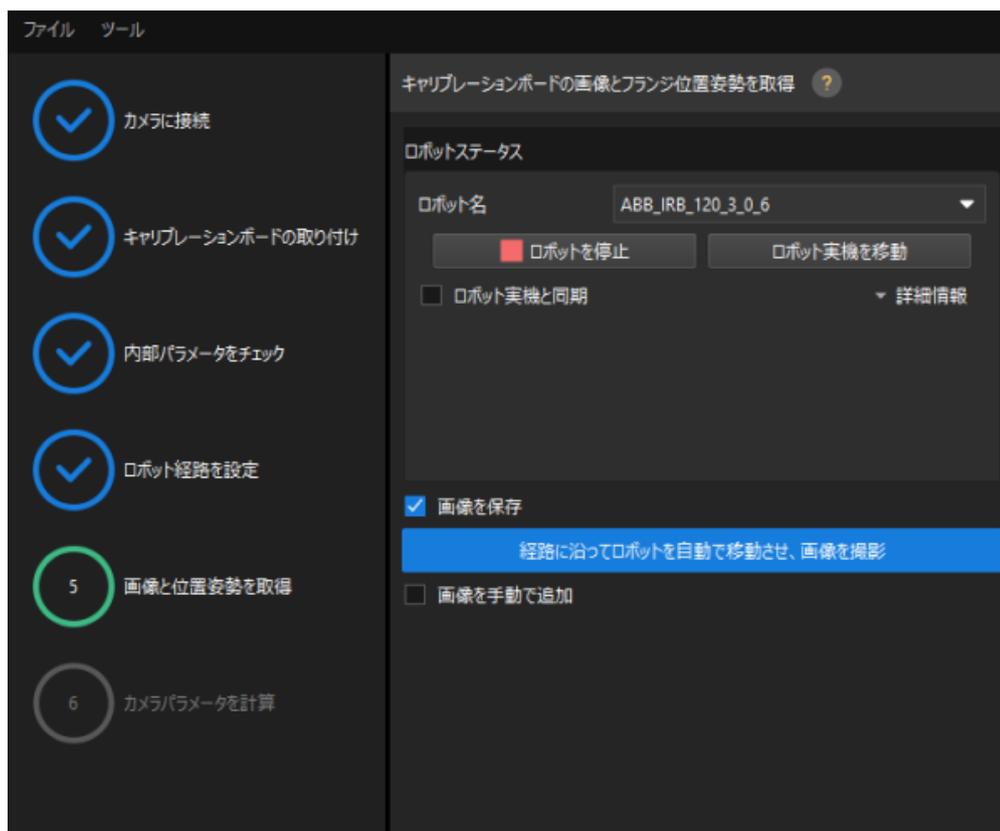
2. 実際の状況に応じて、**経路のタイプ**をToHandに指定し、ピラミッド型の経路パラメータ下の**高さ範囲**、**層数**、**最下層サイズX/Y**、**最上層サイズX/Y**と**層ごとの移動グリッドの列と行数**を設定し、**回転角度**を設定してから、**[確認]**をクリックします。

i | ロボットの経路は、作業領域をカバーするように設定する必要があります。

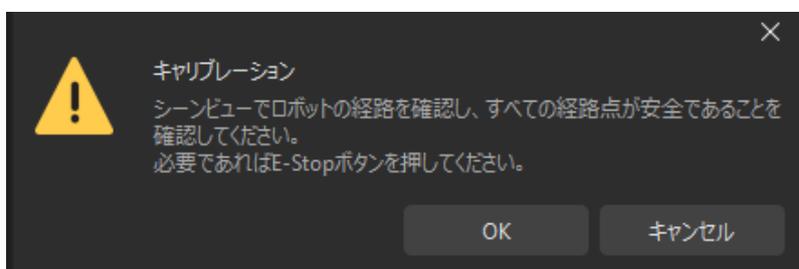
3. 右側の**シーンビュー**では、自動生成された動作経路の各ポイントが周囲と衝突していないことを確認してから、下部にある**[次へ]**をクリックします。

画像と位置姿勢を取得

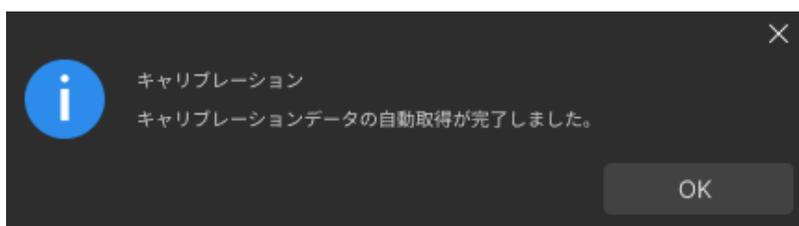
1. **画像と位置姿勢を取得**手順で、**画像を保存**にチェックを入れます。



2. [経路に沿ってロボットを自動で移動させ、画像を撮影]をクリックします。
3. ロボット動作の安全に関する説明をよく読んでから、[OK]をクリックします。

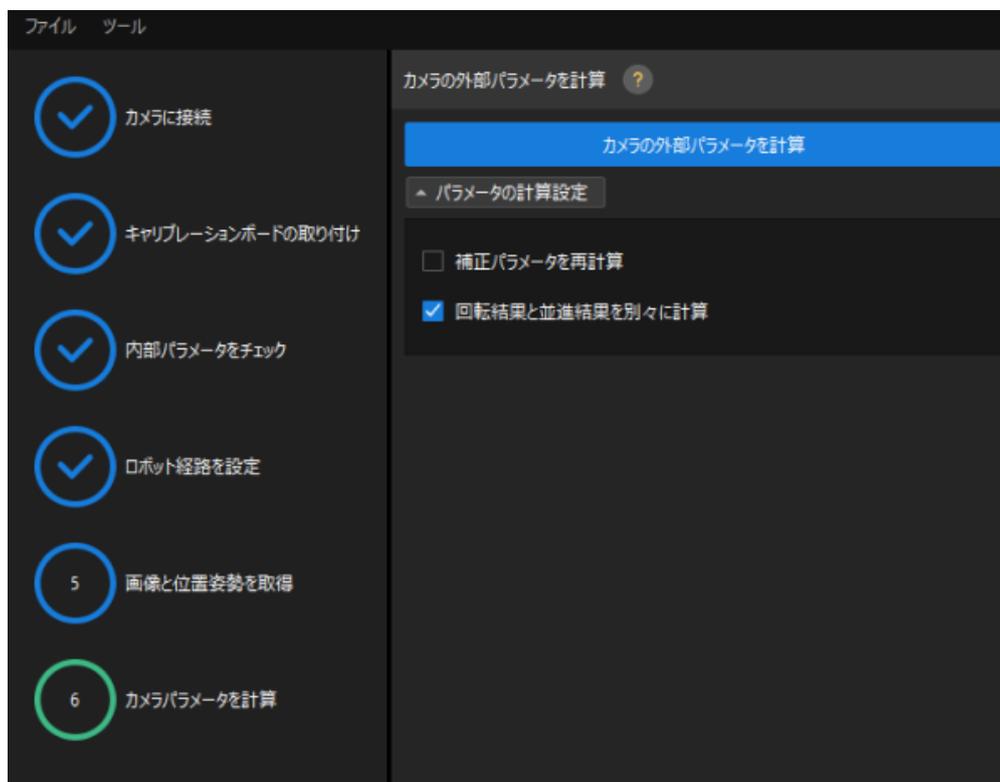


4. ロボットが設定した経路に沿って移動し、カメラが各経路点で画像取得を完了するのを待ちます。右側の画像と位置姿勢一覧には、取得した画像が表示されます。
5. キャリブレーションの自動取得が完了したら、ポップアップウィンドウで[OK]をクリックし、下部にある[次へ]をクリックします。



カメラパラメータを計算

1. カメラパラメータを計算手順で、[カメラの外部パラメータを計算]をクリックします。



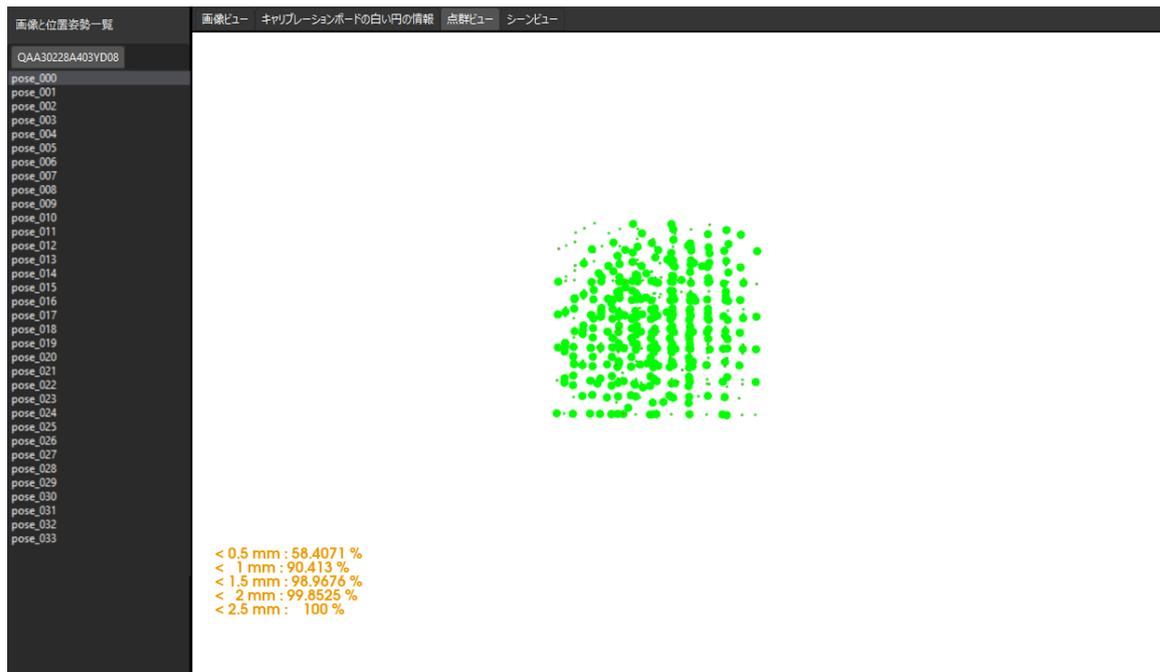
2. キャリブレーションが正常に実行されたことを示すダイアログで、[OK]をクリックします。
3. カメラパラメータを計算の画面でキャリブレーション結果を計算した後、右側の点群ビューパネルでキャリブレーションの誤差点群が表示されます。



誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実際の値と計算値との偏差を示すために使用されます。

4. キャリブレーション精度が要件を満たしていることを確認します。

キャリブレーション精度は、100%を占める誤差の値を求めることで判断できます。例えば、下図は2.5 mm以下の精度を示しています。

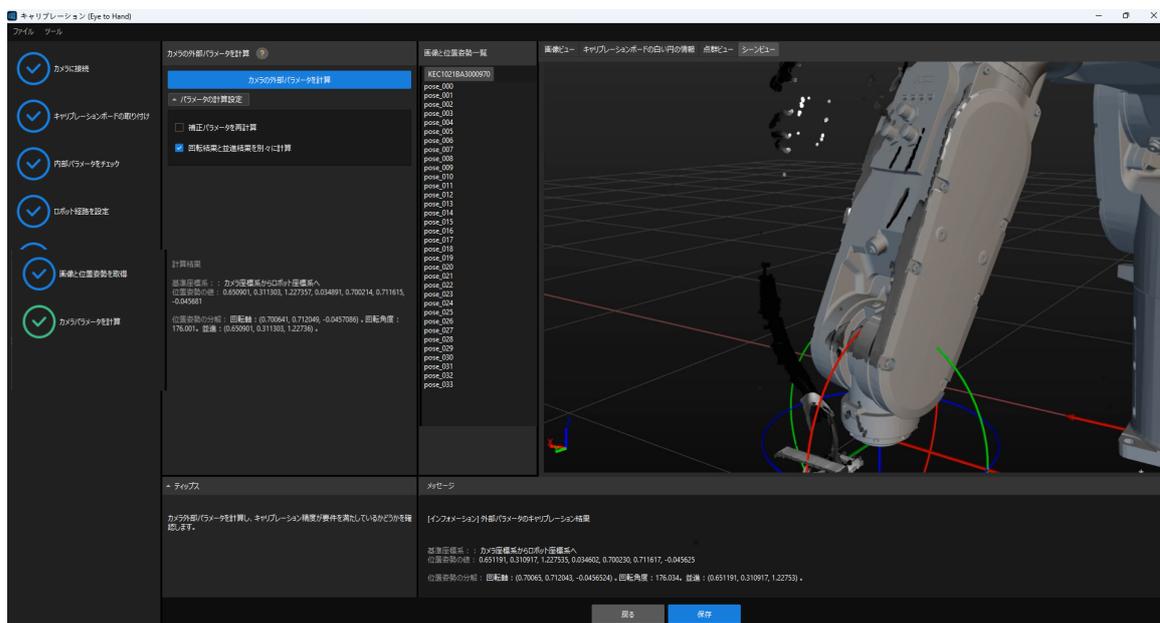


キャリブレーション精度を向上させるには、[キャリブレーション結果の分析](#)をご参照ください。

キャリブレーション結果の確認と保存

1. キャリブレーションが完了したら、ロボットアームをカメラの視野範囲内に移動させます。
2. カメラパラメータを計算手順で[外部パラメータを再計算]をクリックします。クリックすると、カメラが撮影を行います。
3. シーンビューに切り替え、ロボットの点群がロボットモデルの重なりを確認します。

ロボットの点群がロボットモデルとほぼ一致すれば、キャリブレーションは成功です。



- 下部にある[保存]をクリックし、ポップアップする**キャリブレーションファイル**を保存画面で[OK]をクリックします。すると、カメラのキャリブレーション結果は、プロジェクトの「calibration」ディレクトリに自動的に保存されます。

2.4. ワーク認識

本ガイドを読む前に、[ハンド・アイ・キャリブレーション](#)を参照し、「一般的な部品認識」を使用してMech-Visionソリューションを作成する必要があります。

まず、プロジェクトの構築手順を確認し、ステップパラメータの調整とプロジェクト実装により、ワークの位置姿勢を取得してビジョン結果を出力します。



本ガイドでは、ワークのCADモデルを点群マッチングモデルに変換されます。ワークのCADモデルファイルを事前に準備してください。

▶ https://www.youtube.com/watch?v=OL1xnPw_KhA (YouTube video)

動画：ワーク認識

プロジェクトの構築手順

使用するステップと機能は下表のとおりです。

番号	階段	ステップ	サンプル図	説明
1	画像取得	カメラから画像を取得		カメラを接続して画像を撮影します
2	ワーク認識	ワーク認識		3Dマッチングにより、ワークの位置姿勢（把持位置姿勢として使用）を計算します
3	位置姿勢調整	位置姿勢を一括調整		把持位置姿勢をカメラ座標系からロボット座標系に変換します
4	ビジョン結果を出力	出力		ロボット把持のためのワークの位置姿勢を出力します



把持位置姿勢とは、ロボットがワーク表面に把持を実行可能な位置姿勢のことを指します。

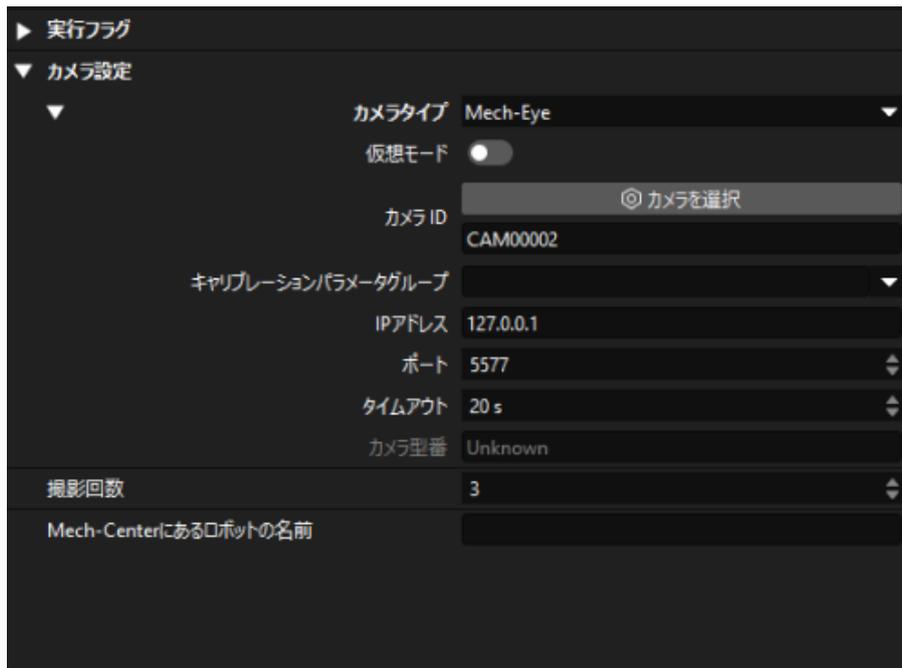
ステップパラメータの調整

以下では、パラメータの調整について説明します。

カメラから画像を取得

「カメラから画像を取得」のステップパラメータを調整し、カメラに接続します。

1. 「カメラから画像を取得」をクリックして選択し、ステップパラメータで **[カメラを選択]** をクリックします。



2. 表示される画面で  をクリックすると、カメラの接続が完了です。カメラが正常に接続された場合、 は  に変わります。



カメラ接続後、パラメータグループを選択します。[パラメータグループを選択]をクリックし、表示されるパラメータグループを選択します。



3. カメラを接続し、パラメータグループを設定すると、キャリブレーションパラメータグループ、IPアドレス、ポートなどのパラメータが自動的に入力され、それ以外のパラメータは初期値のままとなります。

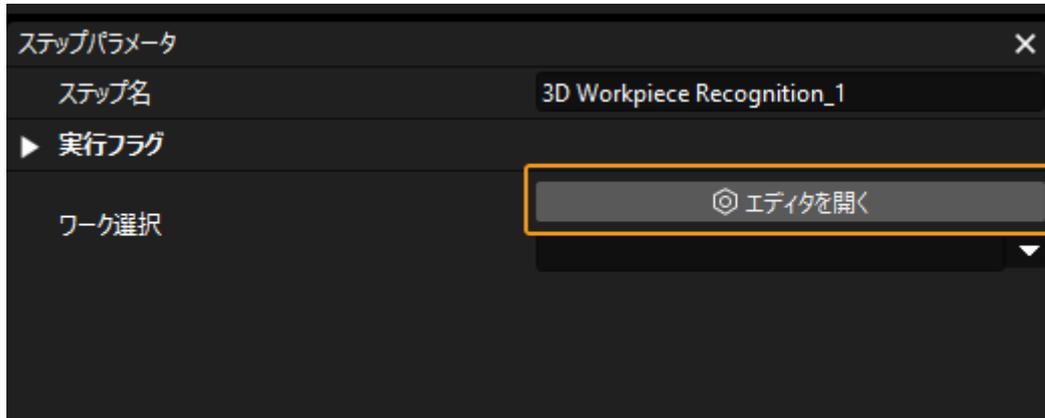


これで、カメラの接続が完了しました。

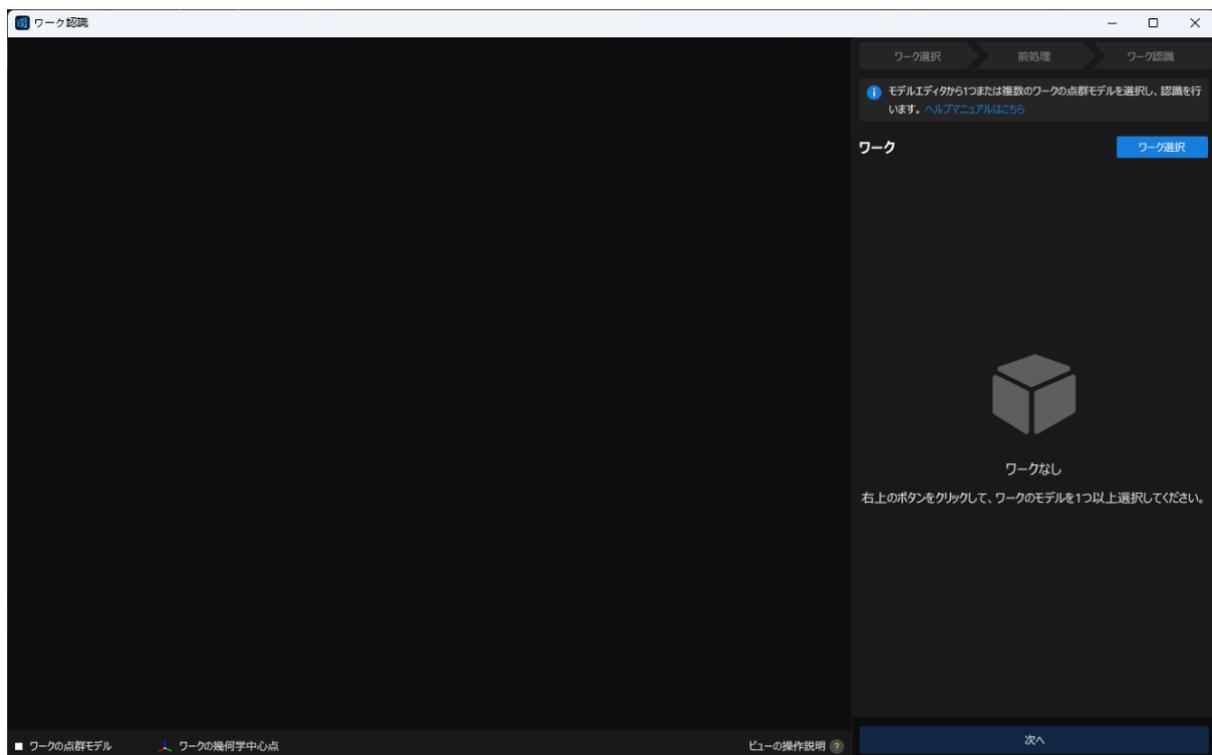
ワーク認識

「ワーク認識」ステップには、可視化設定ツールに組み込まれています。このツールは、点群前処理、モデルマッチング、ワークの位置姿勢（把持位置姿勢）の計算に対応可能です。

「ワーク認識」ステップをクリックして選択し、ステップパラメータで[**エディタを開く**]をクリックします。



「ワーク認識」の可視化設定ツールを下図のように示します。



次に、以下のような手順でワーク認識を行います。



ワーク選択

「ワーク認識」の可視化設定ツールを起動した後、認識するワークの点群モデルを作成する必要があります。

1. モデルエディタを開きます。

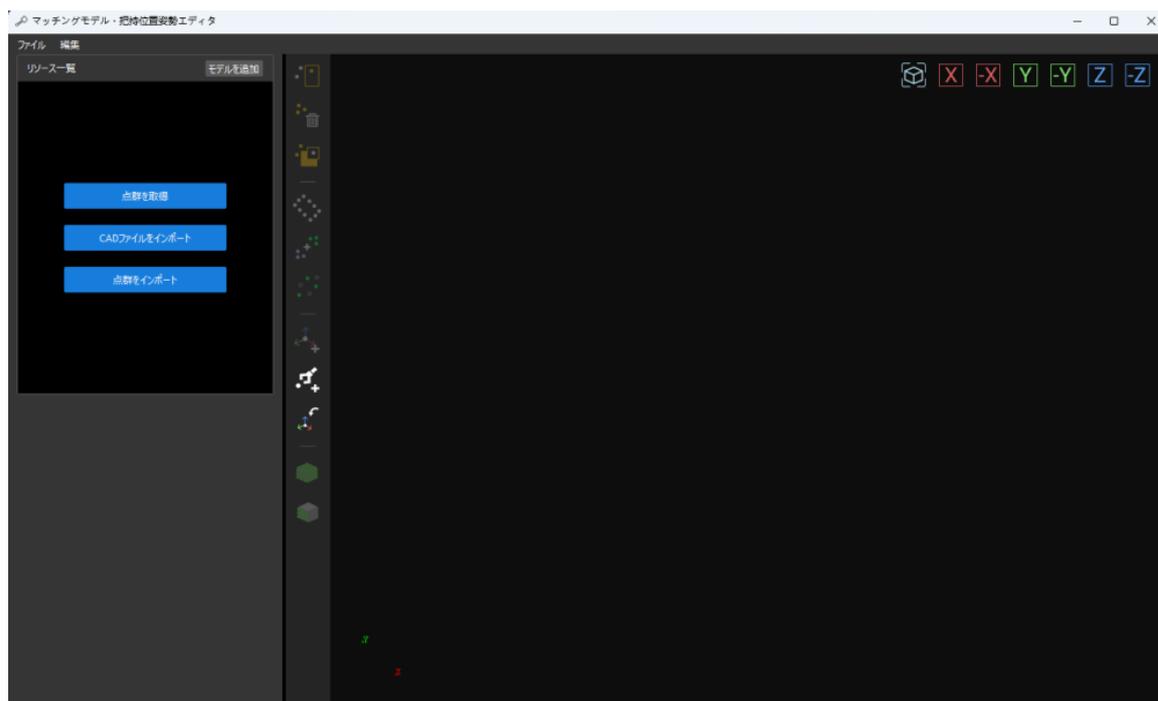
画面の右上隅にある **[ワーク選択]** をクリックします。



表示される画面で **[モデルエディタを開く]** をクリックします。



モデルエディタの画面を下図に示します。

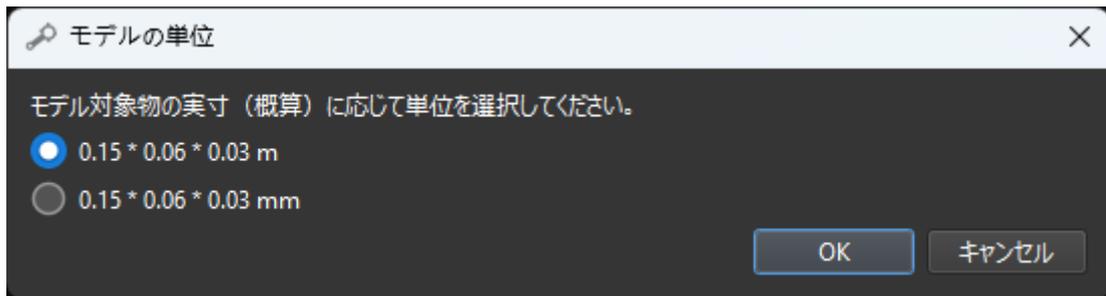


2. CADファイルをインポートします。

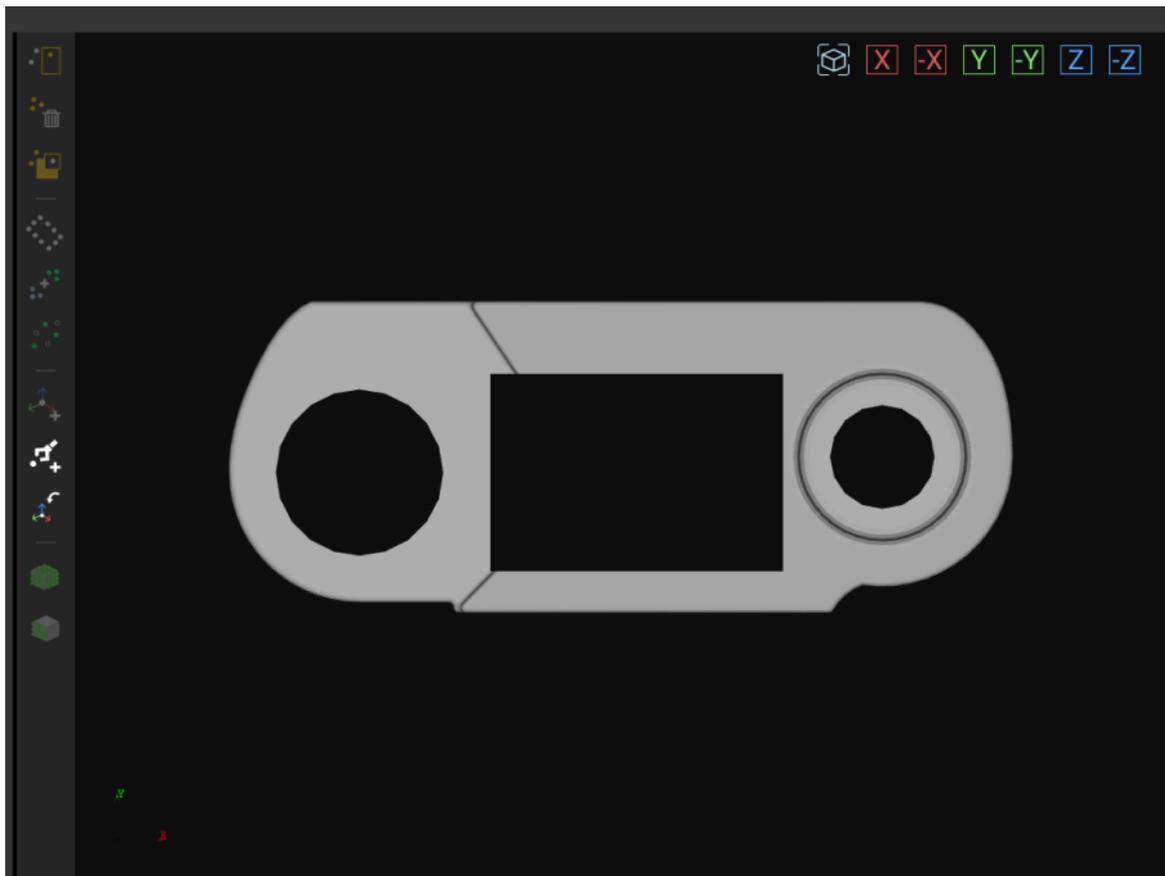
モデルエディタで **[CADファイルをインポート]** をクリックします。



STL形式のワークモデルファイルをインポートし、モデルの寸法単位を選択し、[OK]をクリックします。



CADファイルがインポートされると、モデルエディタ画面中心の可視化エリアに表示されます。



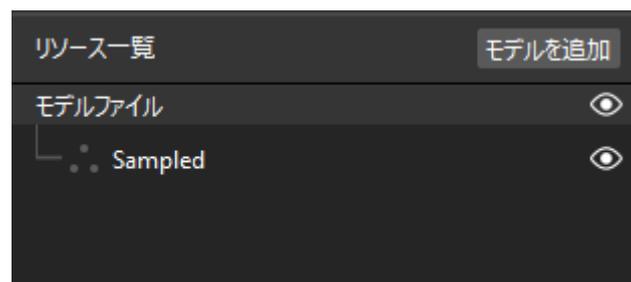
3. CADファイルを使用して点群モデルを作成します。

画面左側のリソース一覧からCADファイルを選択し、ツールバーの  をクリックし、表示されるサンプリング間隔設定画面でサンプリング間隔を設定し、CADモデルの表面点群を生成します。

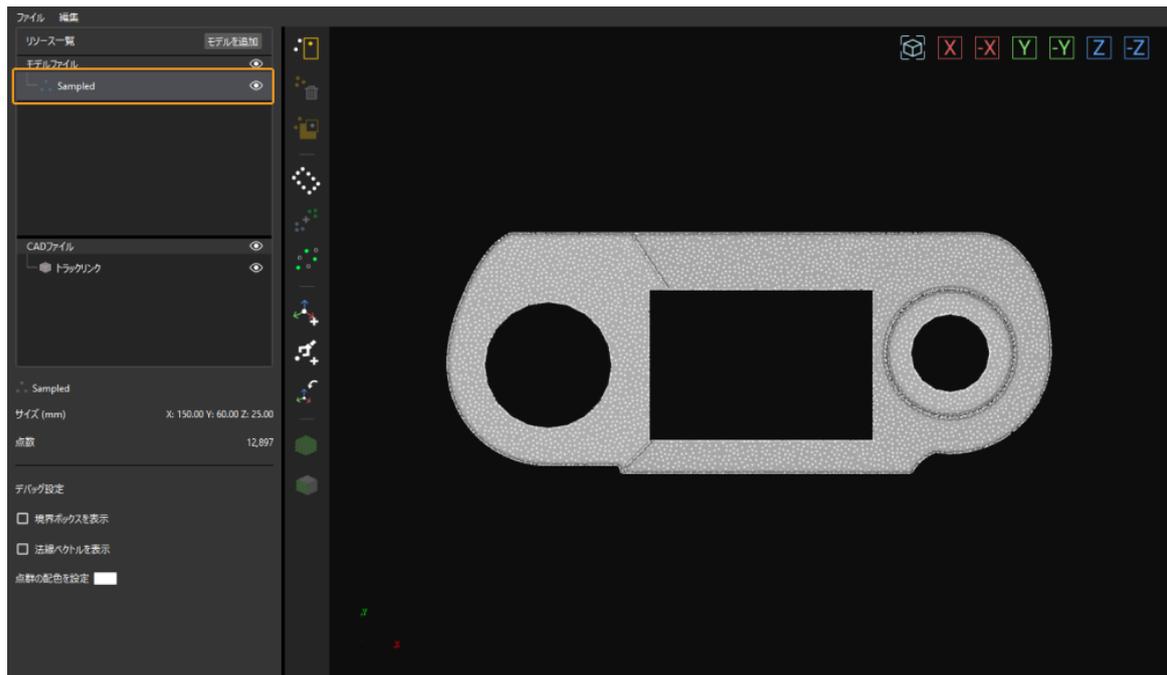


4. 生成された点群モデルを確認します。

CADを使用して作成された点群モデルファイルは、リソース一覧に表示されます。

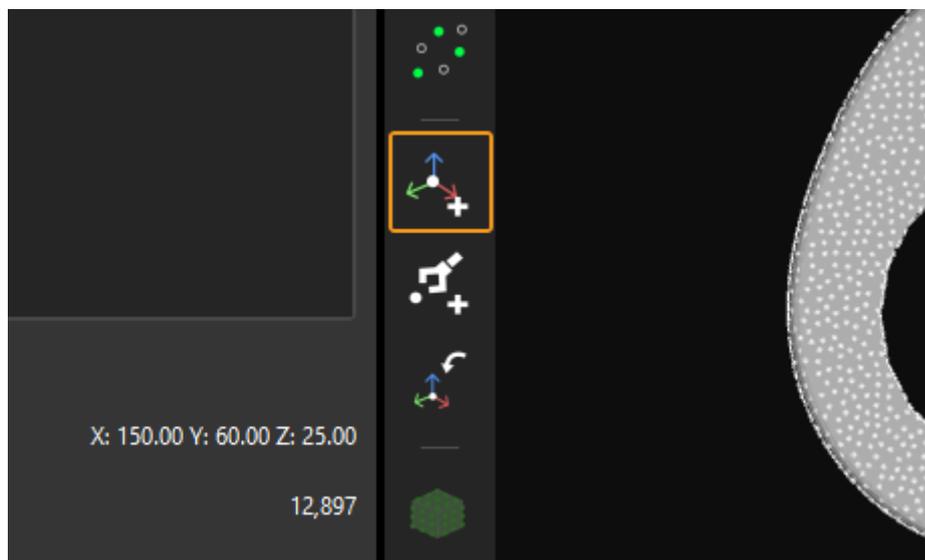


点群モデルファイルをクリックして選択すると、この点群モデルが可視化エリアに表示されます。

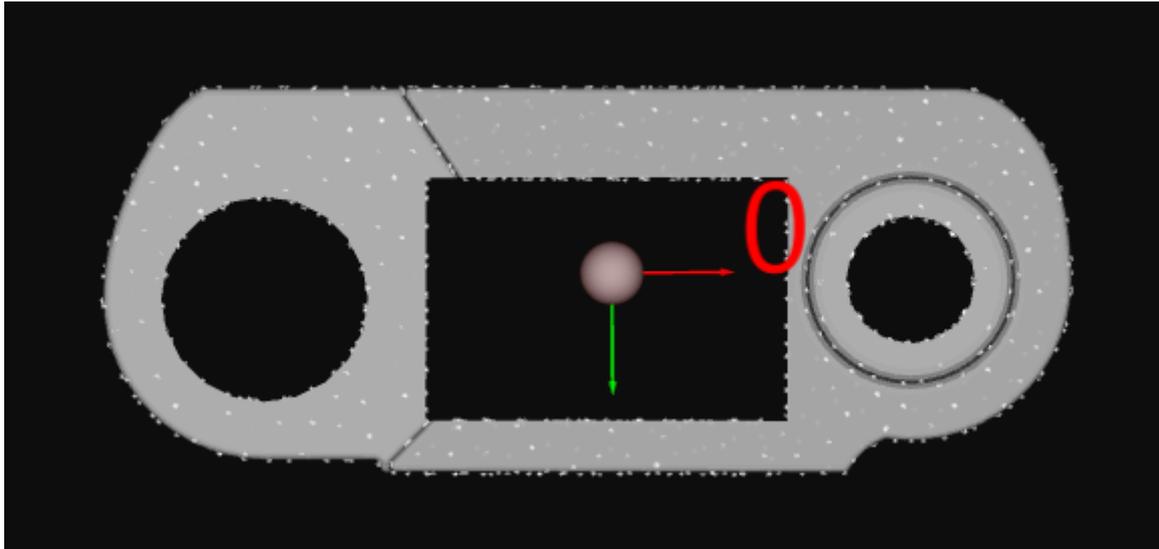


5. 把持位置姿勢を追加します。

ツールバーの  をクリックし、把持位置姿勢として使用する位置姿勢を、ワークの点群モデルに追加します。

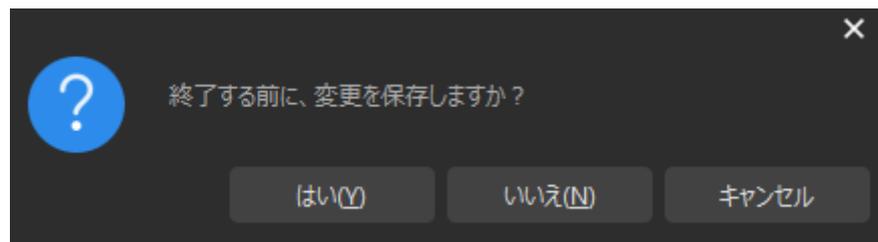


追加された把持位置姿勢を下図に示します。



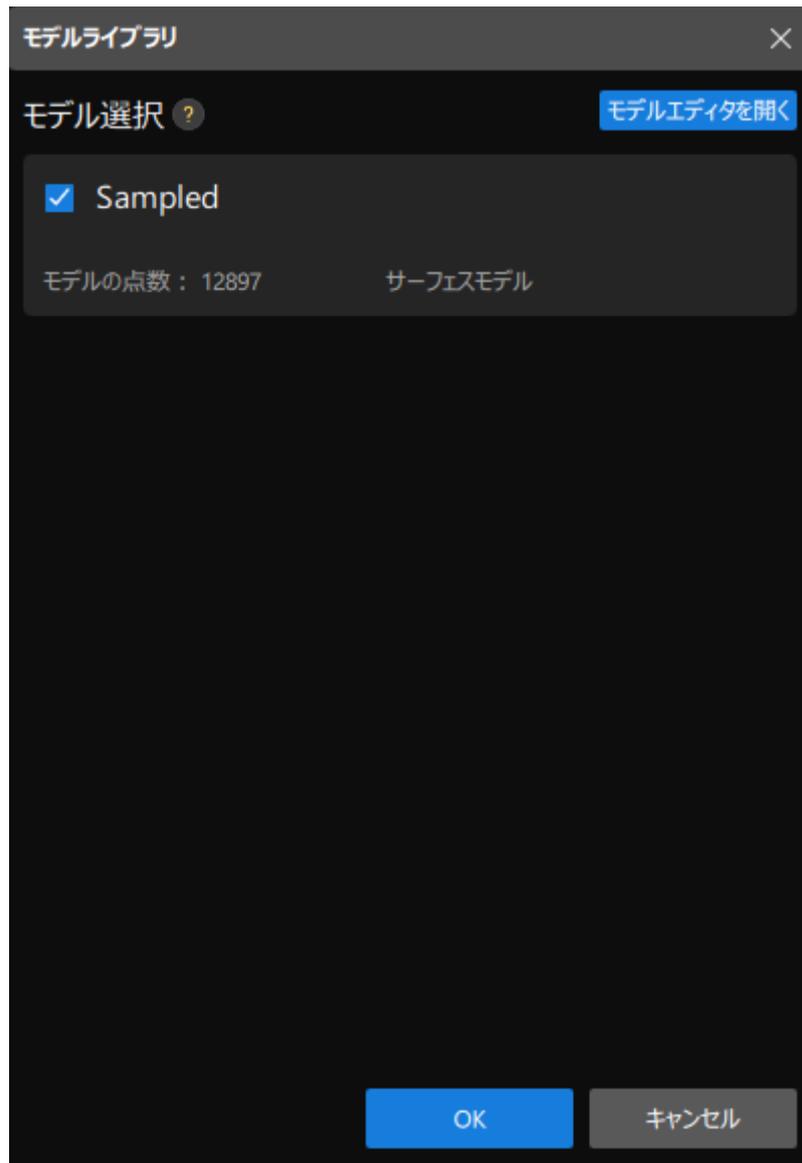
6. モデルと把持位置姿勢を保存します。

モデルエディタの終了ボタンをクリックし、表示される画面で [はい (Y)] をクリックします。

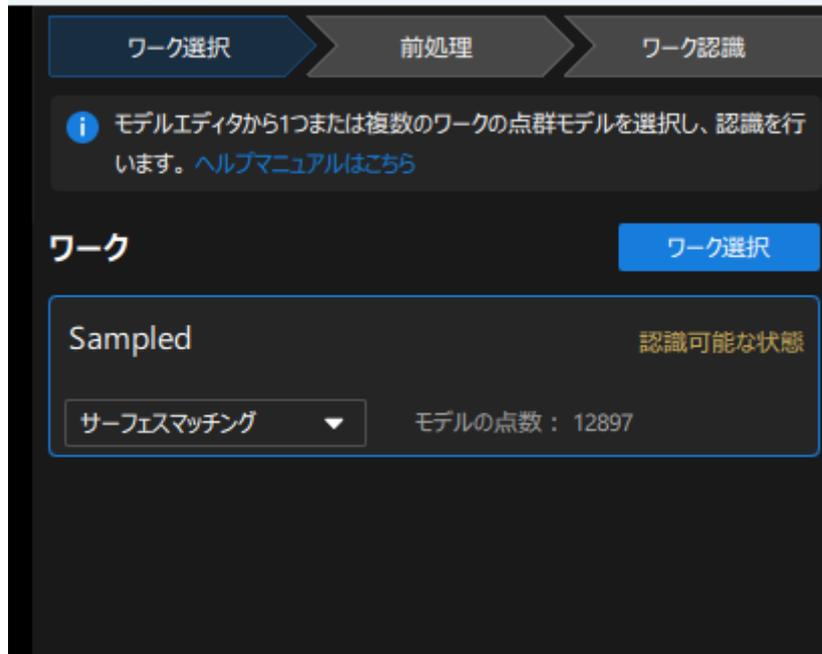


7. モデルライブラリからワークのモデルを選択します。

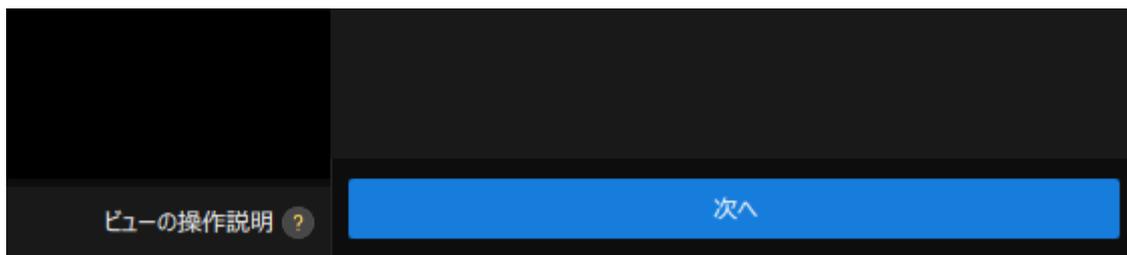
モデルエディタを終了したら、モデルライブラリから保存された点群モデルにチェックを入れ、[OK]をクリックします。



すると、認識するワークは可視化設定ツールの右上隅に表示されます。



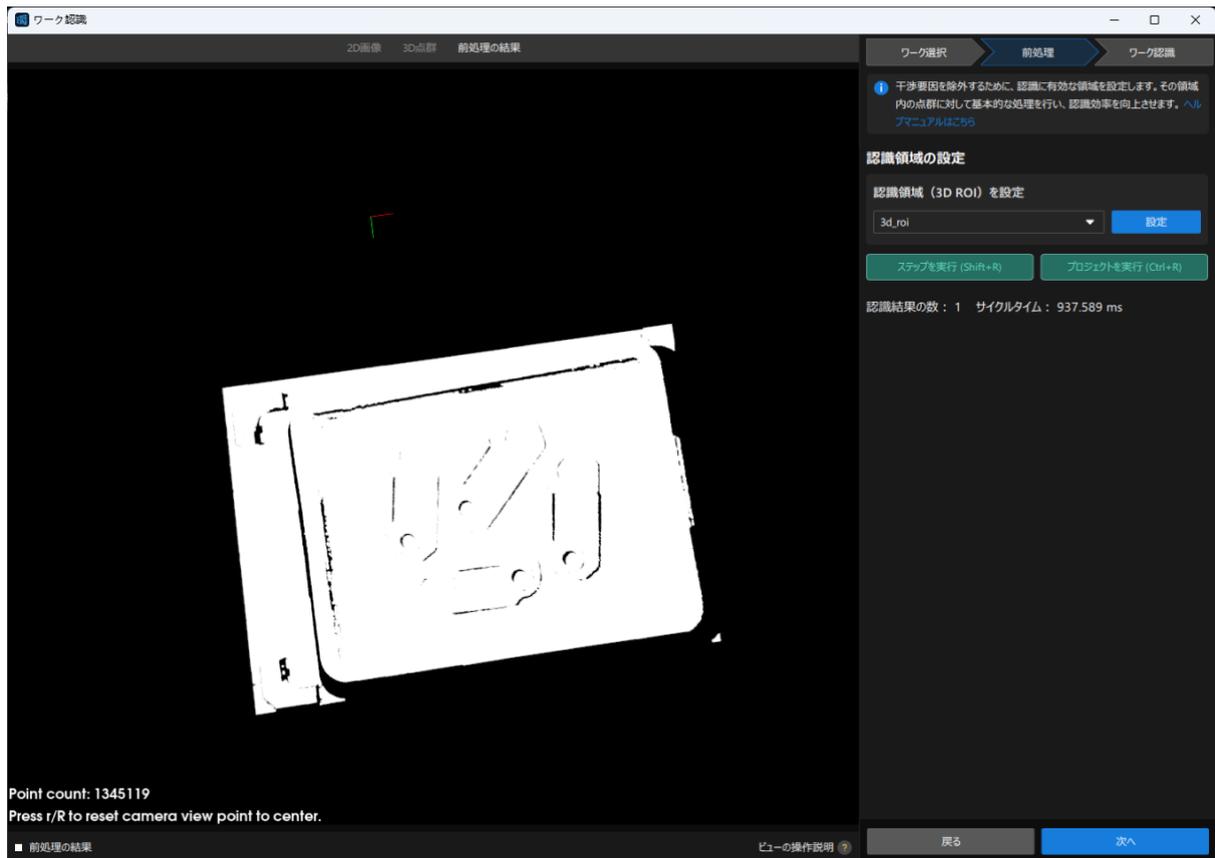
ワークの選択は完了しましたので、画面下部にある[次へ]をクリックして前処理を行います。



前処理

前処理の目的は、認識領域を設定することで不要な点群を除去し、ワークの点群だけを残し、プロジェクトの実行速度を向上させることです。

前処理の画面を下図に示します。

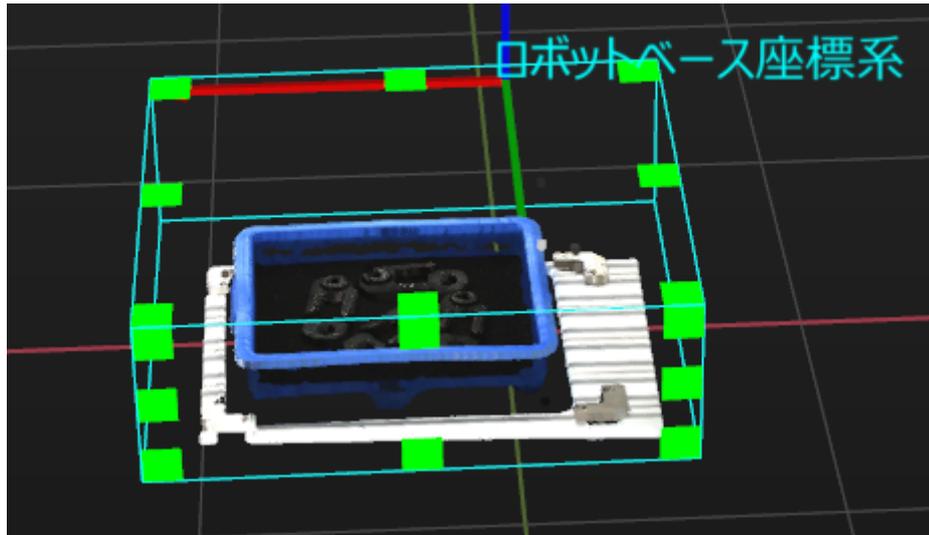


1. 認識領域を設定します。

[設定]をクリックします。



3D関心領域を設定します。**Ctrl** キーを押しながらマウスの左ボタンで3D ROI選択フレームの頂点を長押ししてドラッグし、それを適切なサイズに変更します。変更後の効果を下図に示します。

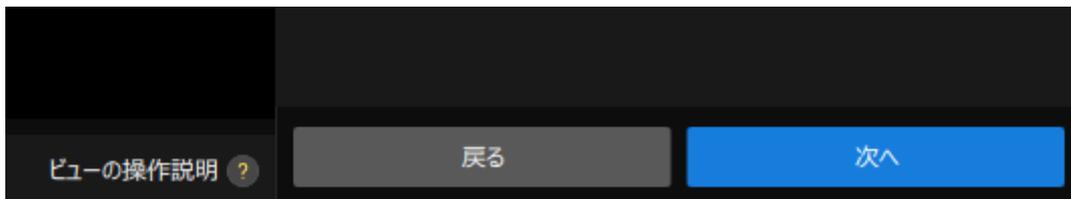


2. 認識領域の設定を保存します。

[保存して適用] をクリックして設定を保存します。



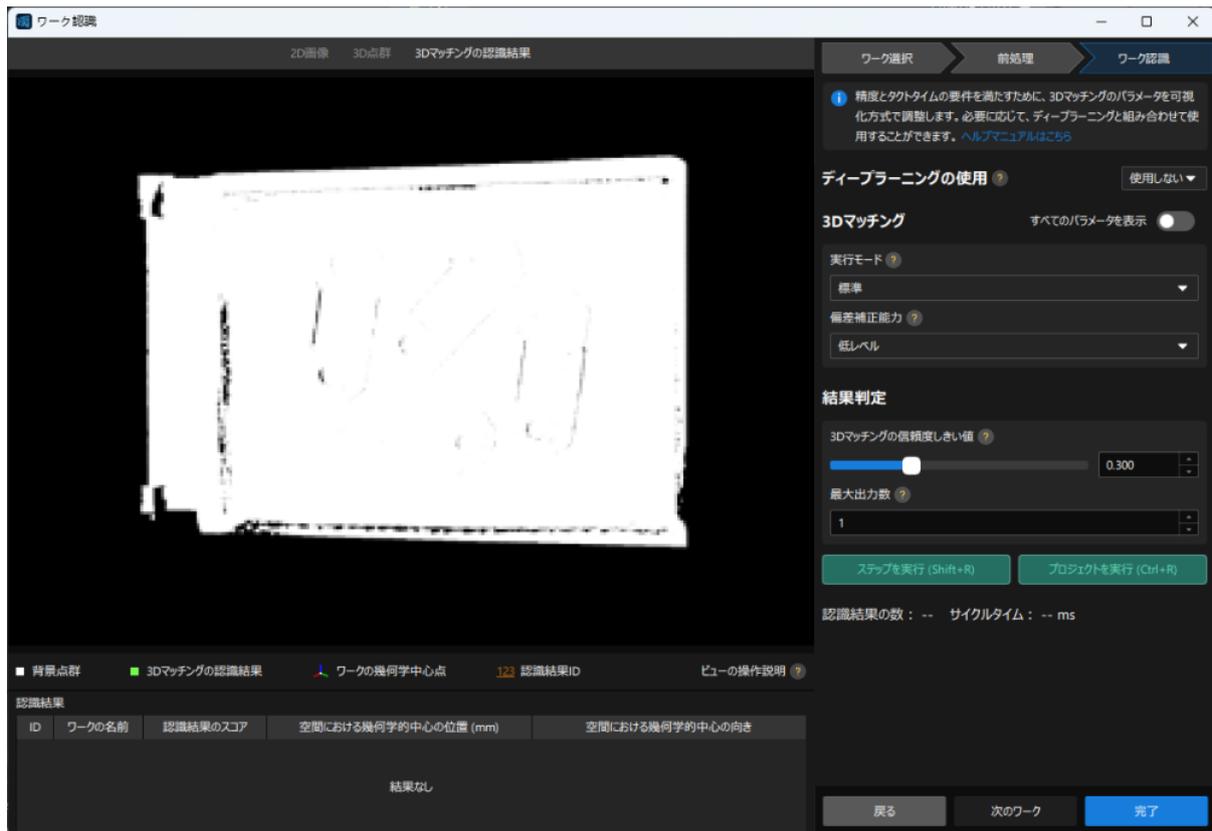
前処理は完了しましたので、画面下部にある[次へ]をクリックしてワーク認識を行います。



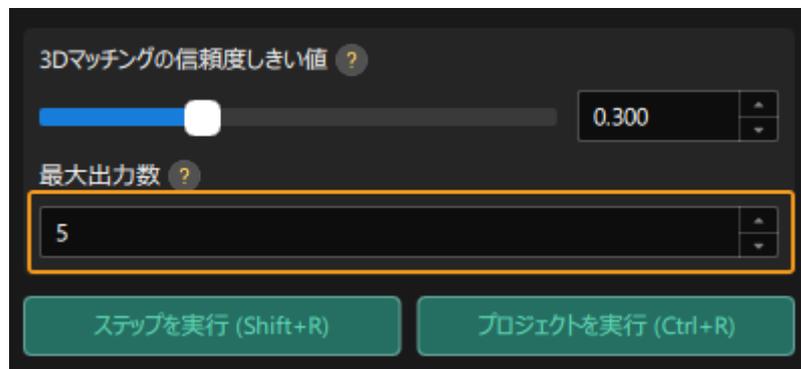
ワーク認識

可視化方式で3Dマッチングの関連パラメータを調整することで、ワークの位置姿勢を出力できます。

認識画面を下図に示します。



1. 本プロジェクトでは最大5つのワークを認識できるため、**最大出力数** を5に設定します。



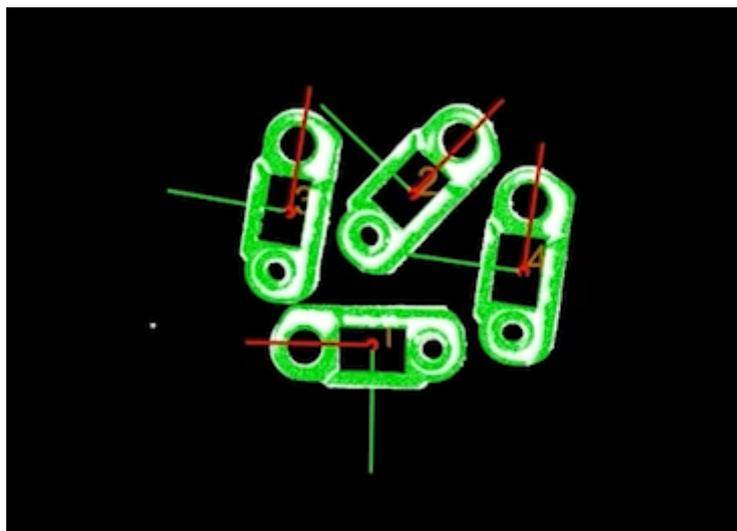
2. 可視化出力結果を確認します。

[ステップを実行] をクリックします。



すると、可視化出力結果が可視化エリアに表示されます。下図に示すように、4つのワーク

の位置姿勢が出力されます。

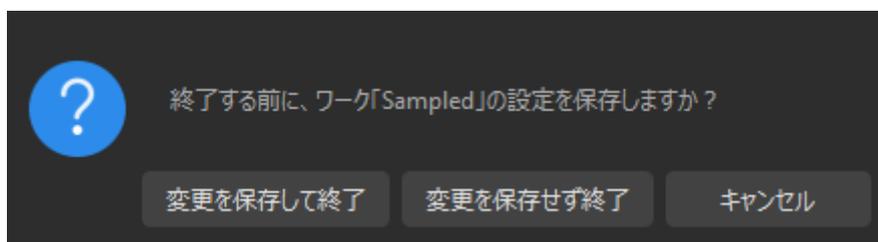


3. 設定を保存します。

可視化設定ツールの画面下部にある **[完了]** をクリックします。



表示される画面で **[変更を保存して終了]** をクリックします。



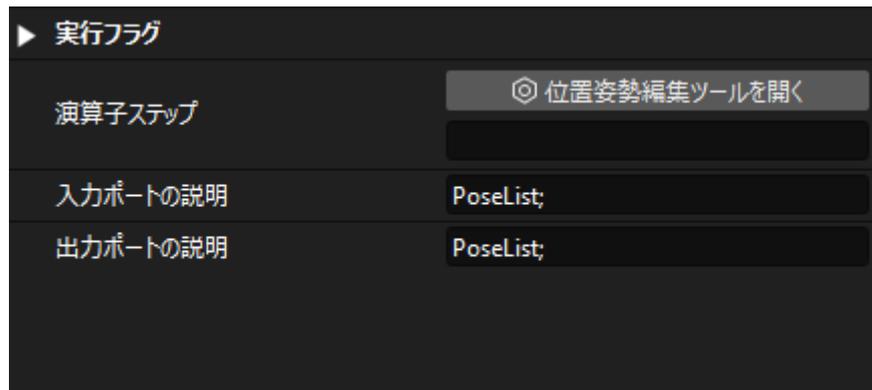
これで、ワークの認識と把持位置姿勢の計算が完了しました。

位置姿勢を一括調整

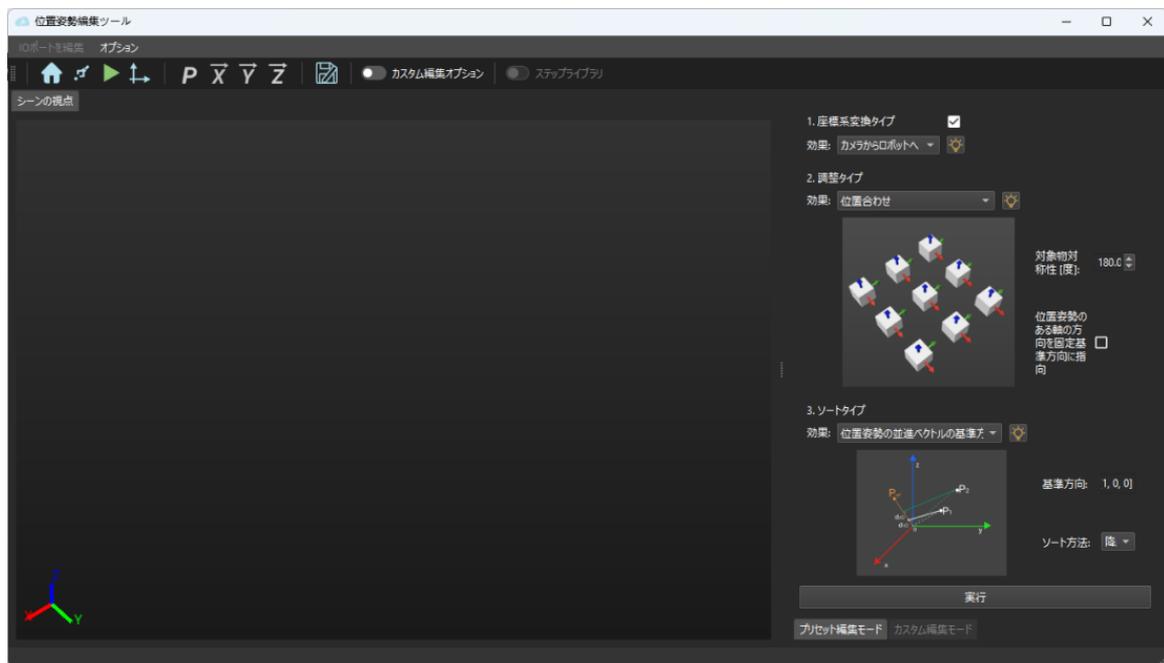
「ワーク認識」ステップによって出力された把持位置姿勢はカメラ座標系にあり、ロボットが把持しやすいようにワークの位置姿勢を調整する必要があります。位置姿勢をカメラ座標系からロボット座標系に変換します。

1. 位置姿勢編集ツールを開きます。

「位置姿勢を一括調整」ステップをクリックして選択し、ステップパラメータで **[位置姿勢編集ツールを開く]** をクリックします。

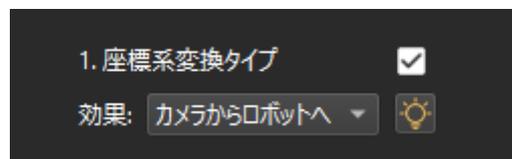


位置姿勢編集ツールの画面を下図に示します。



2. 座標系の変換タイプを調整します。

座標系変換タイプを **カメラからロボットへ** に設定します。

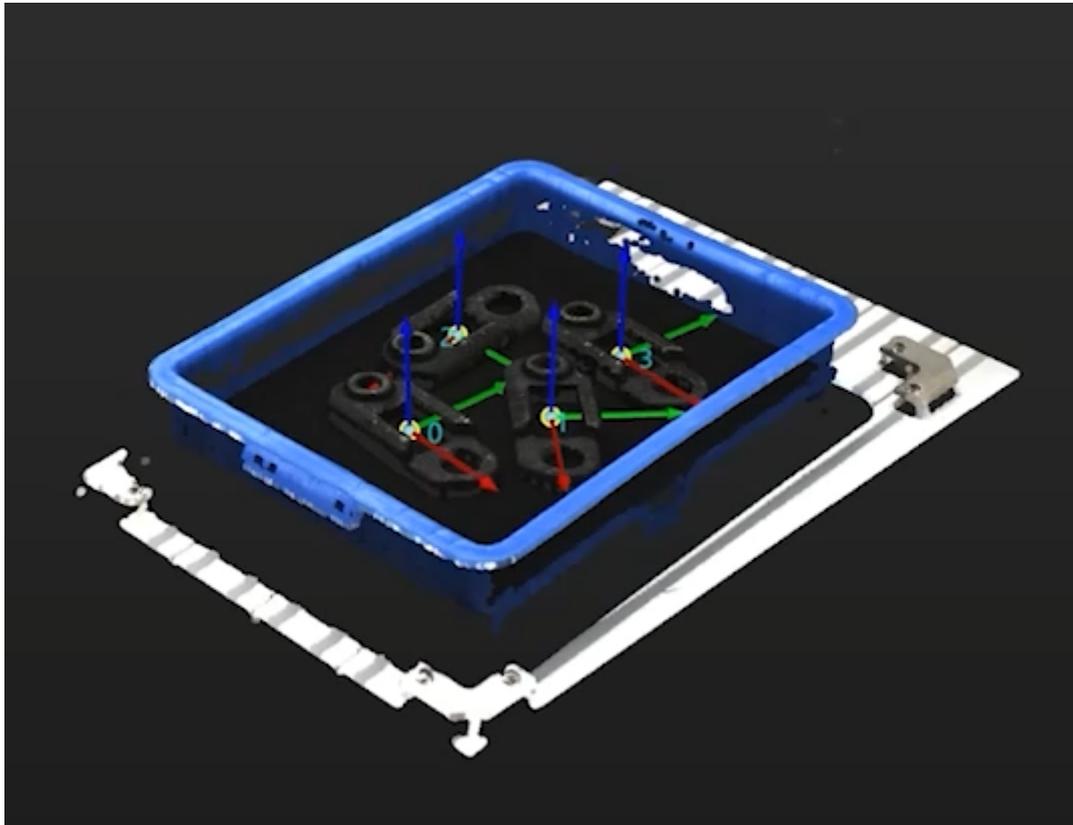


3. 座標系の変換効果を確認します。

画面下部にある[**実行**]をクリックします。

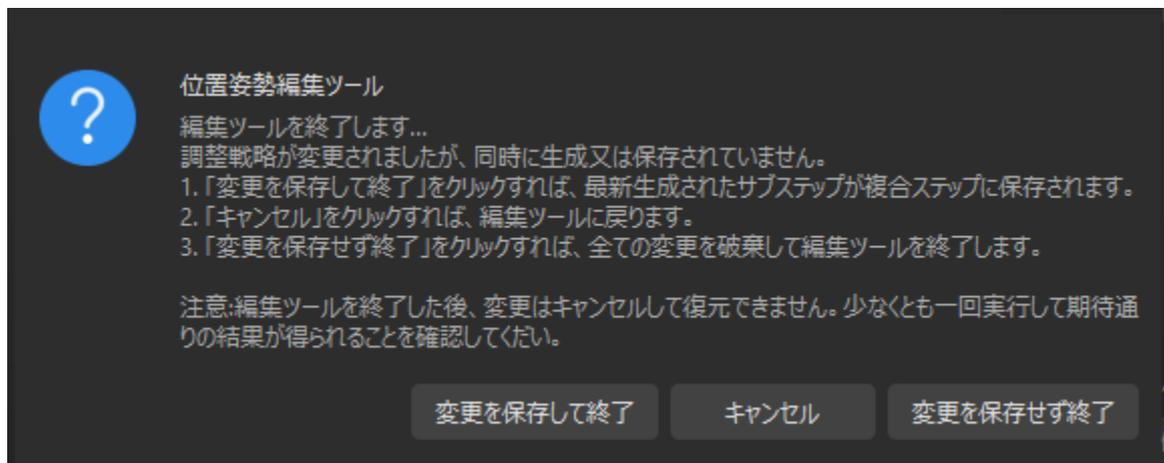


すると、画面中央のシーンの視角から変換後の把持位置姿勢を確認できます。



4. 設定を保存します。

位置姿勢編集ツールの終了ボタンをクリックし、表示される画面で[変更を保存して終了]をクリックします。



これで、把持位置姿勢の座標系変換が完了しました。

出力

「出力」ステップを使用して、現在のプロジェクトの結果をバックエンドサービスに送信します。

これで、Mech-Visionプロジェクトの設定は完了しました。

2.5. 把持と配置を実行

Mech-Visionソリューションでワークの位置姿勢を取得した後、ロボットがワークの把持・配置を繰り返し実行するためにMech-Vizプロジェクトの構築が必要です。



本ガイドでは、衝突検出のためにロボットハンドのOBJ形式のモデルファイルが必要です。ロボットハンドのOBJ形式のモデルファイルを事前に準備してください。

▶ <https://www.youtube.com/watch?v=FPfopHTXqjA> (YouTube video)

動画：最初の把持を実行

今回作成したプロジェクトは下図のような流れで処理を行います。



ロボットとシーンの設定

ロボットの把持・配置時に周囲の対象物との衝突を避けるため、衝突検出のためのロボットハンドモデル、シーンモデル、箱モデルをプロジェクトに追加する必要があります。

ロボットハンドモデルのインポートと設定



ロボットハンドとは、ロボット先端に取り付けられ、対象物を扱う/把持するための装置を指します。

ロボットハンドのインポートと設定を行う目的は、ロボットハンドのモデルを仮想空間に表示し、衝突検出に利用できるようにするためです。

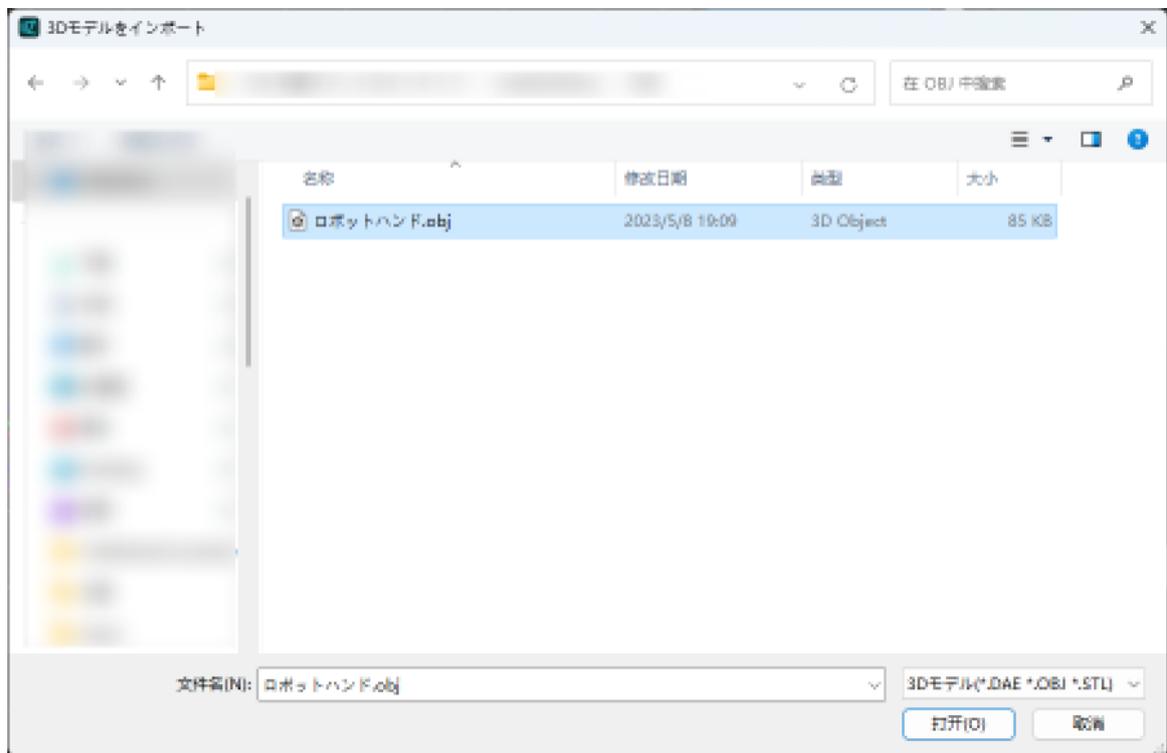
ロボットハンドのモデルのインポート

1. リソース > モデルライブラリの[+]をクリックします。

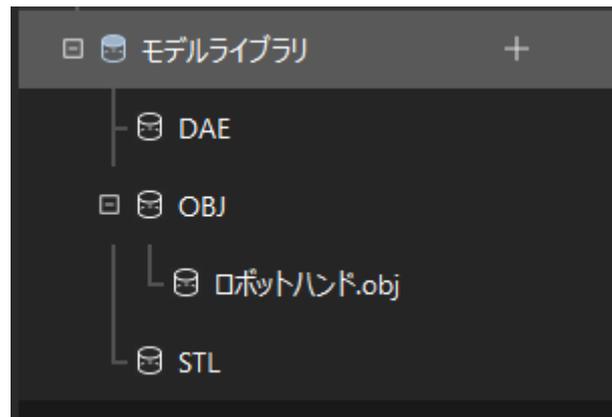


リソースとは、ロボットやロボットハンド、対象物、シーンの物体など、プロジェクトを構築するための要素を指します。

- 表示される画面でOBJ形式の衝突モデルのファイルを選択して[開く]をクリックします。

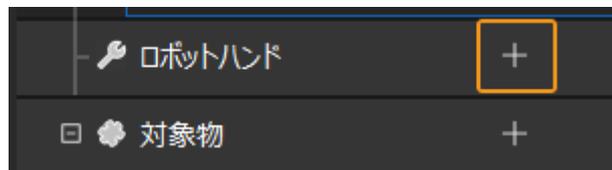


- モデルのインポートが完了すると、モデルライブラリでインポートしたモデルが表示できるようになります。

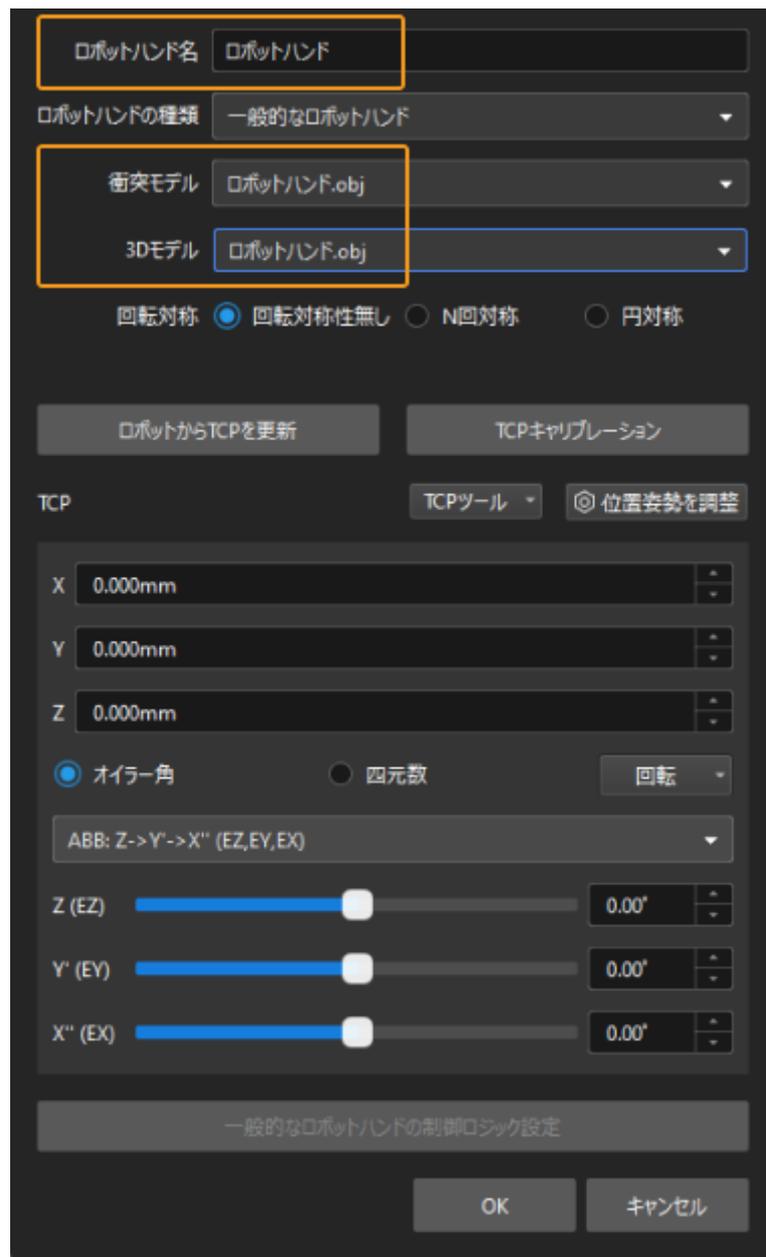


ロボットハンドモデルの設定

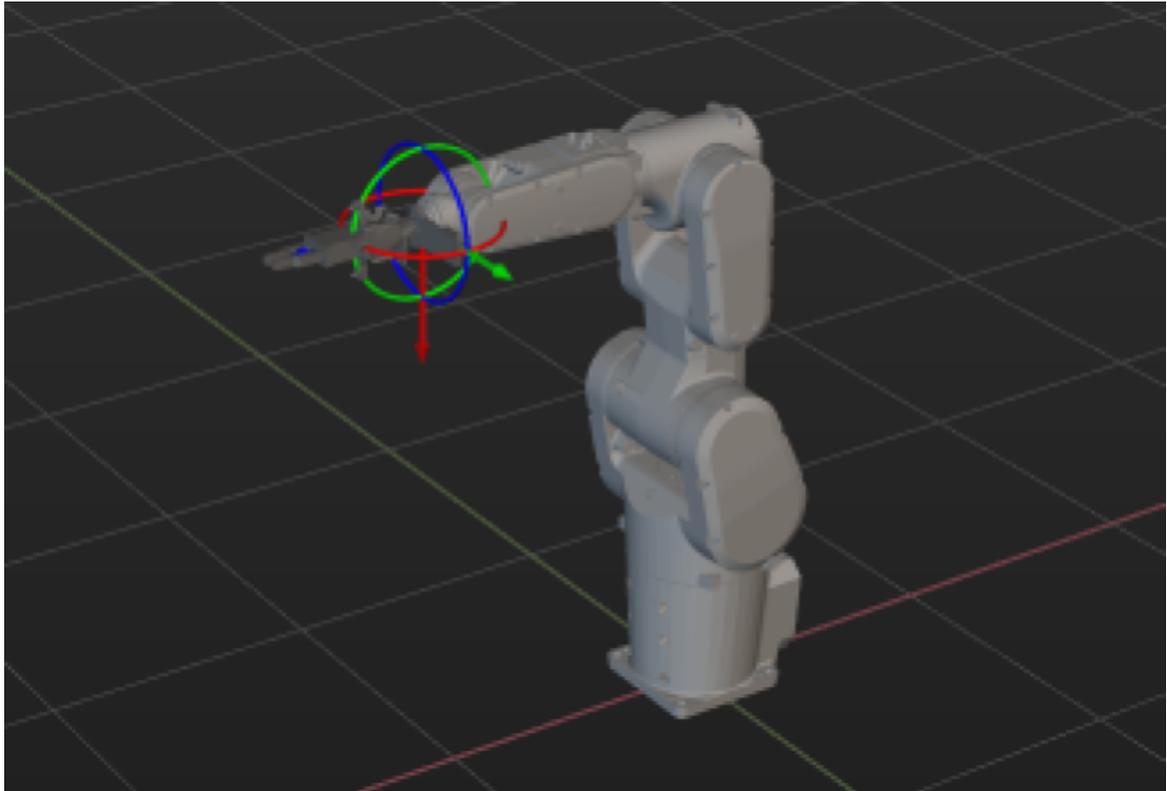
1. リソース > ロボットハンドの[+]をクリックします。



2. 表示される画面にロボットハンドの名前を入力し、インポートしたロボットハンドのモデルファイルを衝突モデルと3Dモデルとして使用し、[OK]をクリックします。



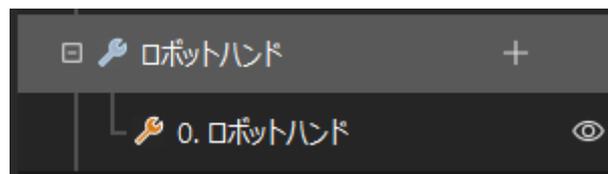
3. ロボットハンドが設定されると、下図のように、設定されたロボットハンドが仮想空間に表示できるようになります。



ロボットハンドの調整

上図では、ロボットに対するロボットハンドの位置が間違っていますので、ロボットハンドの位置調整が必要です。

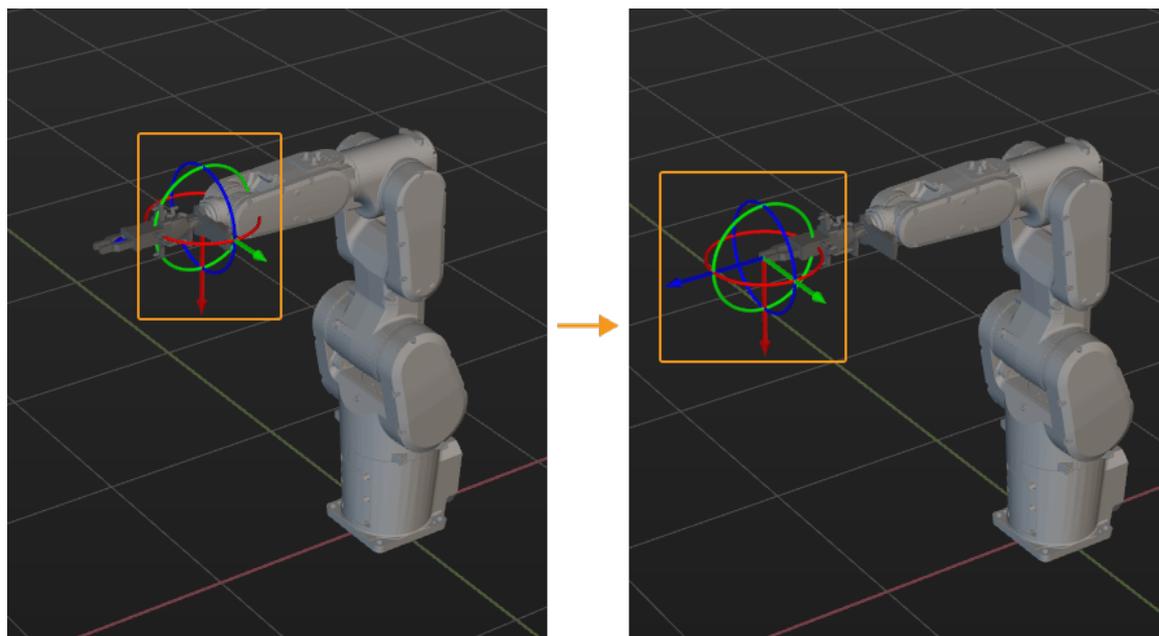
1. モデルライブラリ内のロボットハンドモデルをダブルクリックします。



2. 表示される画面で以下の設定を行います。



3. 調整前と調整後のロボットハンドの位置を下図に示します。



シーンモデルのインポートと設定

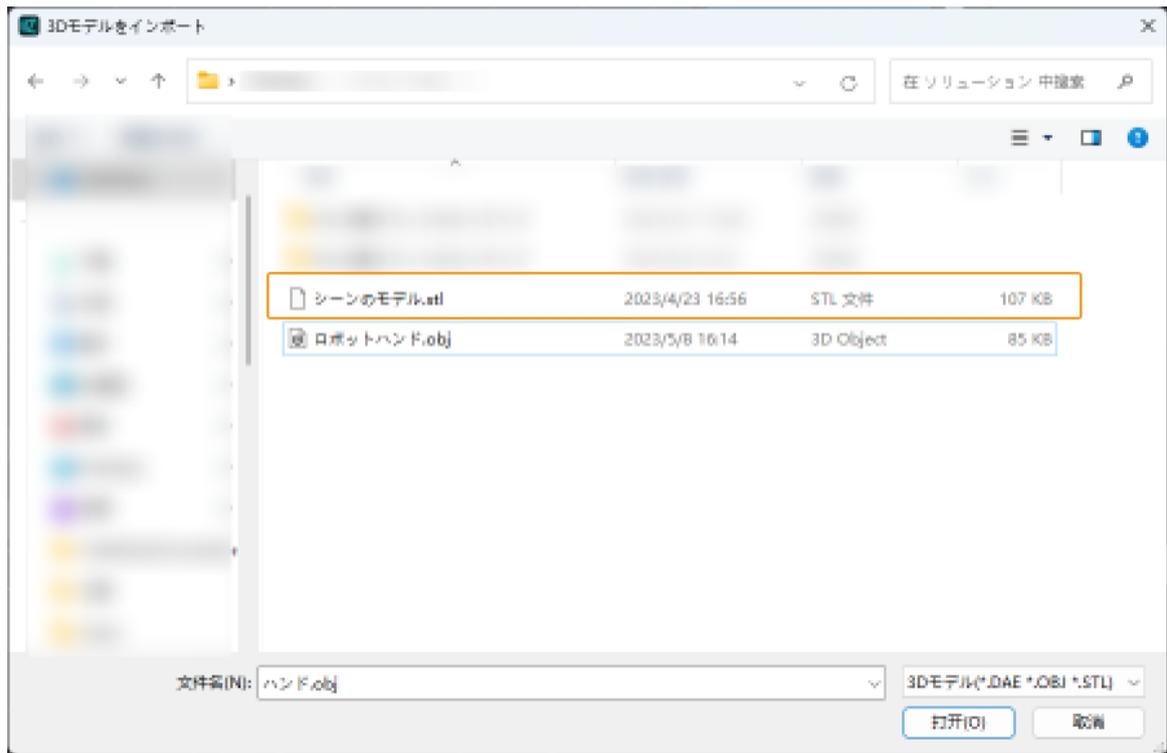
シーンモデルをインポートして設定する目的は、実際の作業現場を再現することで、ロボットの動作経路を計画できるようにすることです。

シーンモデルのインポート

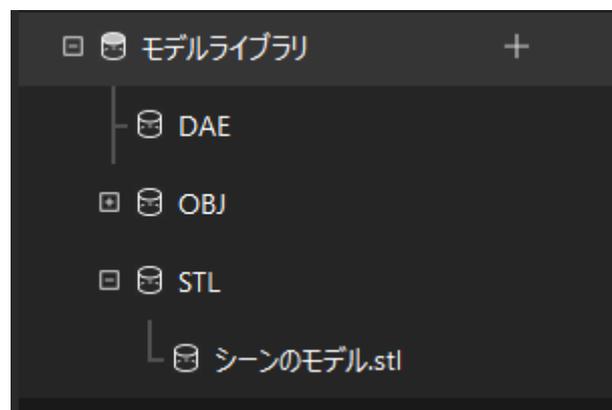
1. リソース > モデルライブラリの[+]をクリックします。



2. 表示される画面でシーンの物体のモデルファイルを選択して[開く]をクリックします。

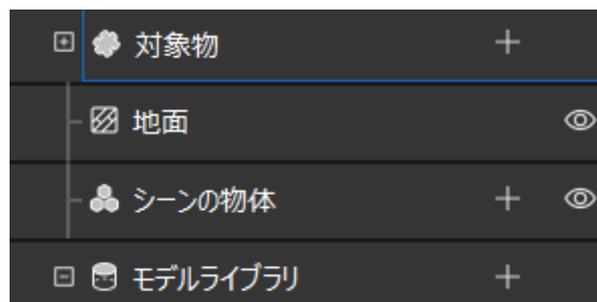


3. モデルのインポートが完了すると、モデルライブラリでインポートしたモデルが表示できるようになります。



シーンモデルの設定

1. リソース > シーンの物体の[+]をクリックします。



2. 表示される画面に物体の名前を入力し、「外部モデル」を「シーンのモデル」とし、インポ

ートしたシーンの物体のモデルファイルを衝突モデルと3Dモデルとして使用し、[OK]をクリックします。

物体設定 物体の位置姿勢

物体名
シーンのモデル

シーンモデル
外部モデル

衝突モデル
シーンのモデル.stl

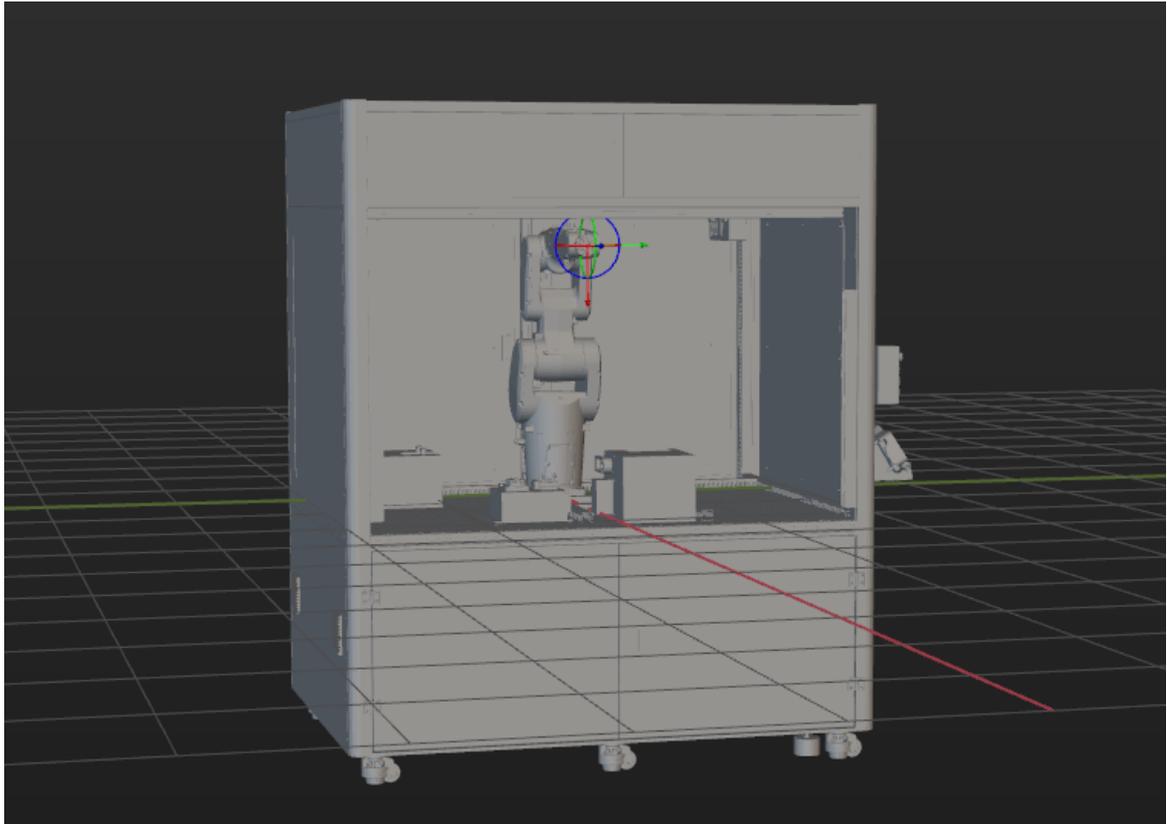
3Dモデル
シーンのモデル.stl

衝突検出に使用する

モデル選択可能

OK キャンセル

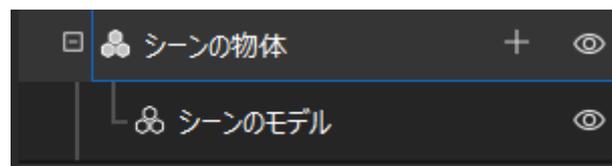
3. 設定されると、シーンの物体が仮想空間に表示できるようになります。



箱モデルの追加と設定

1. 「箱モデル」を追加して設定します。

リソース > シーンの物体の[+]をクリックします。



表示される画面で、「把持ボックス」を物体名と入力し、「箱」をシーンモデルとして選択します。測定された実際の箱の寸法に従って「物体の寸法」を設定してから、[OK]をクリックします。

物体設定 物体の位置姿勢 箱の設定

物体名
把持ボックス

シーンモデル
箱

物体の寸法

X 430.000mm

Y 330.000mm

Z 150.000mm

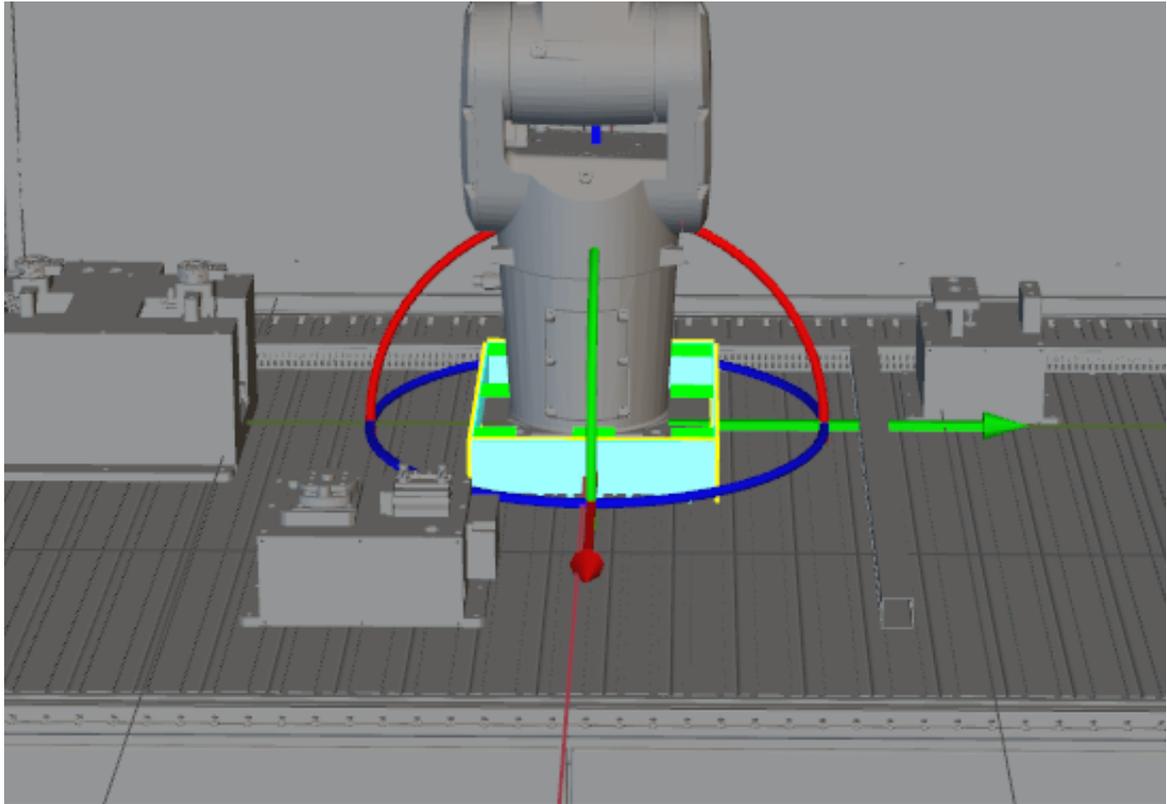
厚さ 2.000mm

衝突検出に使用する

モデル選択可能

OK キャンセル

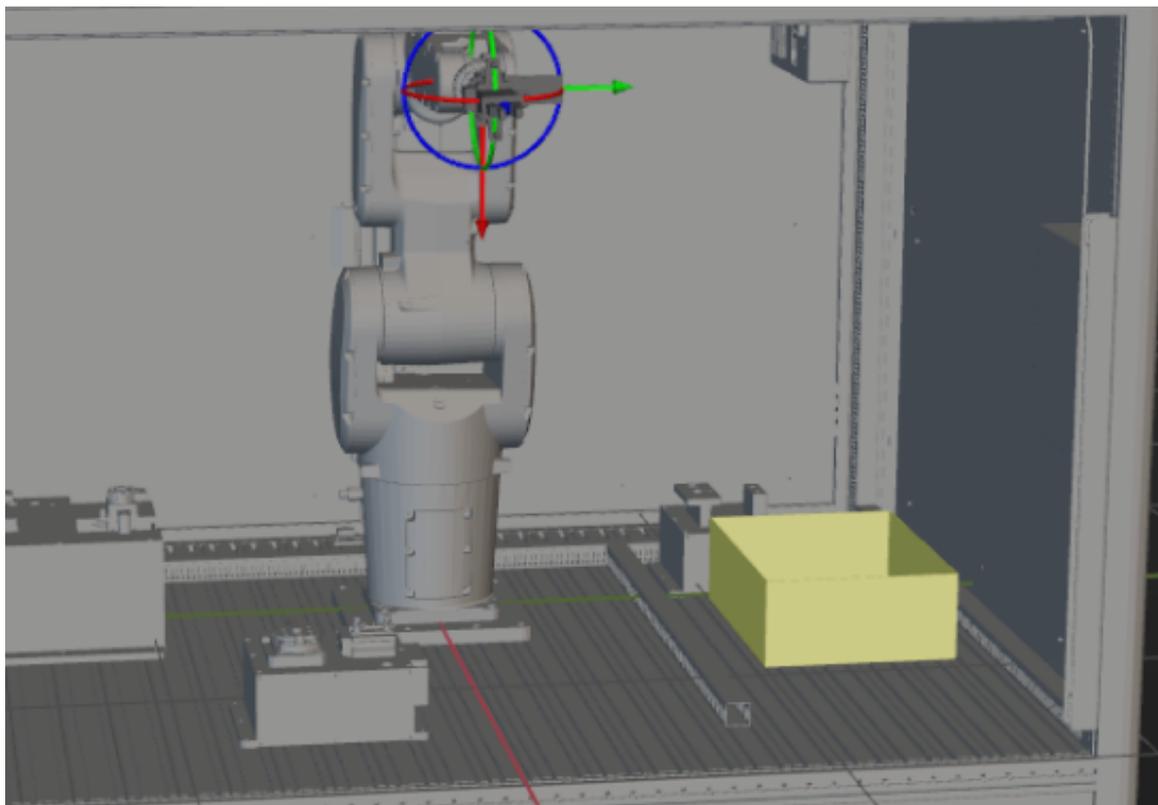
設定されると、「把持ボックス」が仮想空間に表示できるようになります。



2. 「把持ボックス」モデルを調整します。

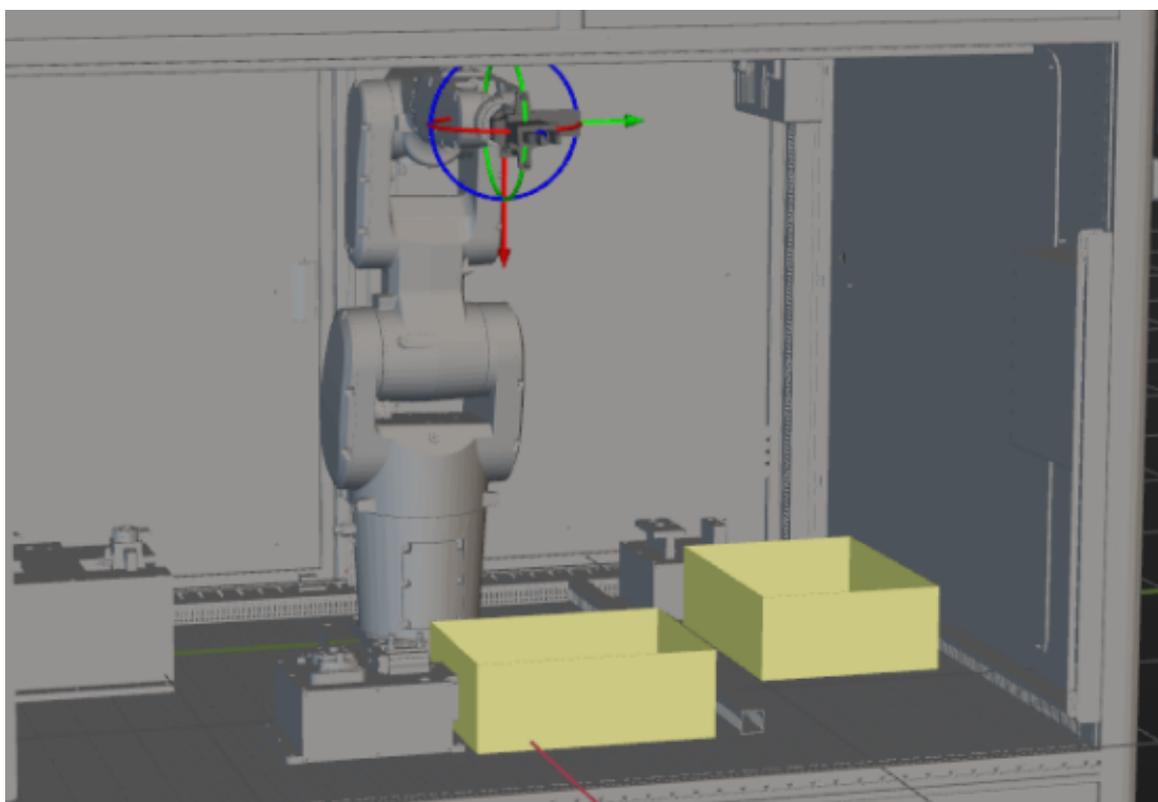
実際の箱の位置を取得するために、Mech-Visionプロジェクトを実行して実際の箱の点群をMech-Vizに送信する必要があります。その後、実際の箱の点群に応じて「把持ボックス」モデルの位置を調整します。

マウスの左ボタンを押しながら **Ctrl** を押し、「把持ボックス」モデルの座標軸をドラッグして位置を調整します。調整完了後、「把持ボックス」モデルは以下のような位置になります。



3. 「配置ボックス」モデルを追加して設定します。

下図に示すように、上記と同様な方法で「配置ボックス」モデルを追加して設定します。



これで、Mech-Vizプロジェクトに必要なリソースの追加と設定が完了しました。

ワークフローの構築

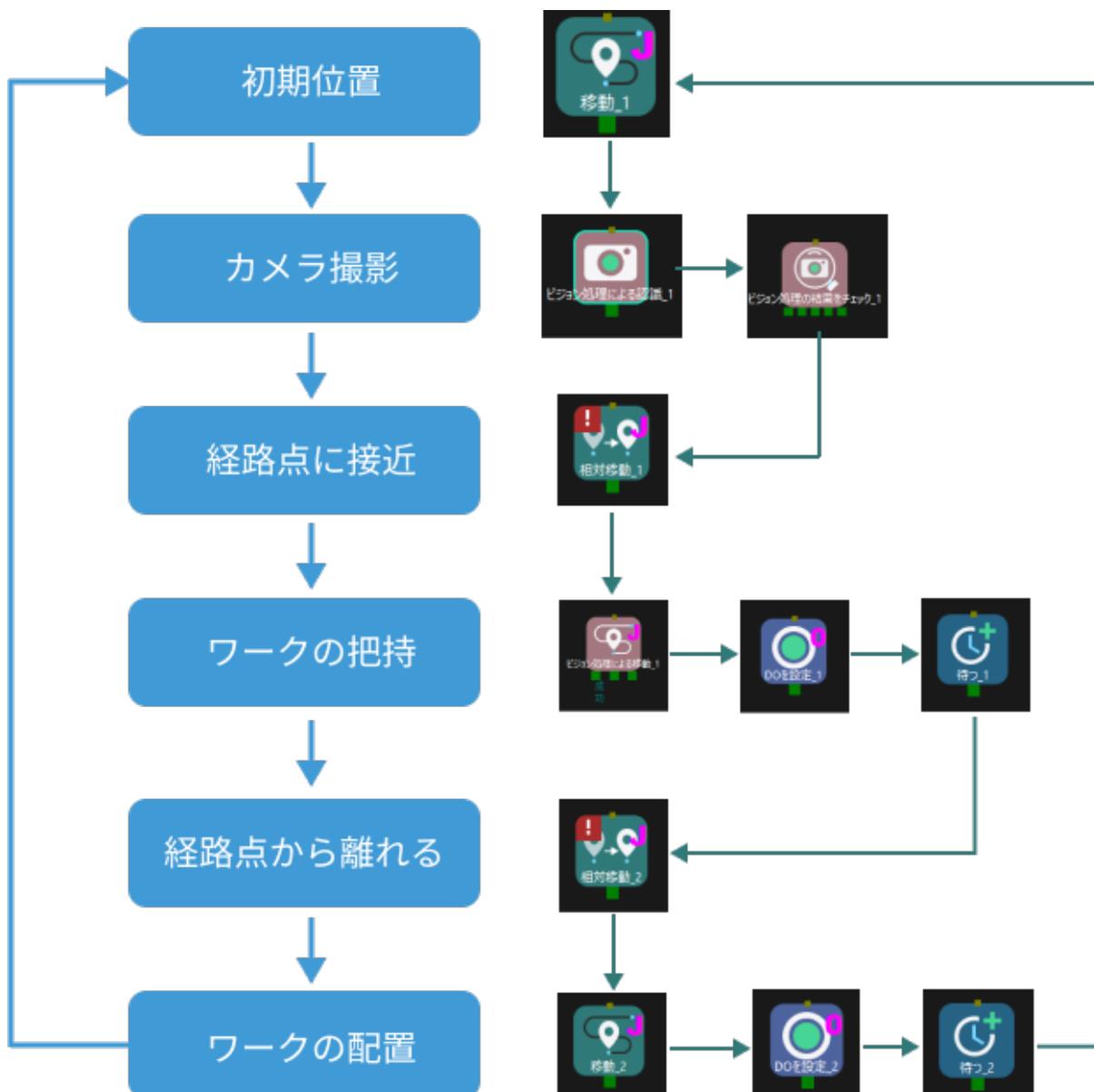
モデルの設定が完了したら、ワークフローの構築を行います。プリセットプログラムの機能は、ステップライブラリからプロジェクト編集エリアにステップをドラッグし、ステップパラメータを設定し、ステップをつなぐことで実現します。



- ワークフローとは、Mech-Vizでフローチャートの形式で作成されたロボット動作制御プログラムです。
- ステップとは、ロボットをプログラミングするための機能モジュールです。

ワークフローの構築手順

下図のような手順でワークフローの構築を行います。



ワークフローが構築されると、以下のような効果が実現できます。



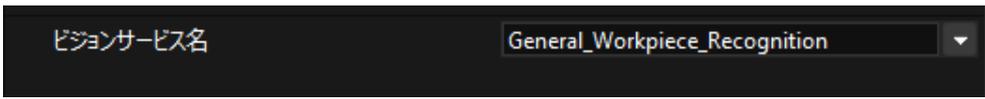
「初期位置」を定義

初期位置はロボット動作の開始位置だけでなく、安全な位置でもあります。この位置で、ロボットが把持する対象物や周囲の機器から離れ、カメラの視野を遮らないようにする必要があります。

ロボットをカスタイズされた初期位置に移動させた後、ステップライブラリから**移動**を選択し、プロジェクト編集エリアにドラッグして「初期位置」という名前を付けます。ツールバーの[**ロボットを同期させる**]をクリックし、現在ロボットにいる位置姿勢を記録します。



カメラ撮影

ステップ	ビジョン処理による認識
説明	Mech-Visionプロジェクトを実行し、ビジョン結果を取得します。
操作	ステップライブラリから「ビジョン処理による移動」を選択し、プロジェクト編集エリアにドラッグします。
パラメータ設定	ビジョンサービス名のドロップダウンリストで General Workpiece Recognition を選択します。
サンプル図	

ビジョン結果があるかどうかを確認するために、「ビジョン処理による移動」の後に**ビジョン処理の結果をチェック**を追加します。

ステップ	ビジョン処理の結果をチェック
------	----------------

説明	ビジョン結果があるかどうかを確認します。
操作	ステップライブラリから「ビジョン処理の結果をチェック」を選択し、プロジェクト編集エリアにドラッグします。
パラメータ設定	パラメータの調整は不要です。

経路点に近接する位置に移動

ビジョン結果を取得したら、**相対移動**ステップを使用してロボットを経路点に近接する位置に移動させます。

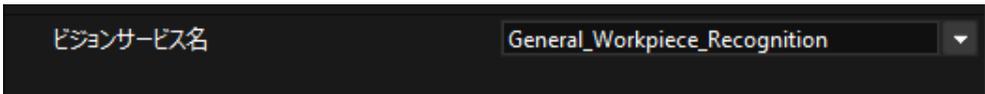
ステップ	相対移動
説明	ビジョン結果に従って移動します。
操作	ステップライブラリから「相対移動」を選択し、プロジェクト編集エリアにドラッグして「経路点に近接」という名前を付けます。
パラメータ設定	移動基準 を次の経路点に対してに設定し、 目標点タイプ をロボットハンドに設定し、 Z軸 の座標を -200mm に設定します。
サンプル図	

ワークの把持

ロボットが経路点に近接する位置に移動したら、次の2つの手順で把持するように制御することができます。

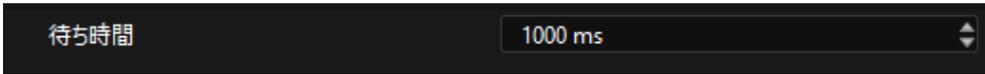
- 手順1：**ビジョン処理による移動**ステップを使用して、ロボットをワークの位置に移動させます。
- 手順2：**DOを設定**ステップを使用して、ロボットハンドを開いてワークを把持します。

詳細は以下の通りです。

ステップ	ビジョン処理による移動
説明	ビジョン結果に従って移動します。
操作	ステップライブラリから「ビジョン処理による移動」を選択し、プロジェクト編集エリアにドラッグします。
パラメータ設定	ビジョンサービス名 のドロップダウンリストで General Workpiece Recognition を選択します。
サンプル図	

ステップ	DOを設定
説明	ロボットハンドを開いてワークを把持します。
操作	ステップライブラリから「DOを設定」を選択し、プロジェクト編集エリアにドラッグして「ハンド開」という名前を付けます。
パラメータ設定	DO値 と DOポート を 1 に設定します。
サンプル図	

ロボットハンドが開くのに時間がかかるので、ロボットが把持しても、何も把持しないように**待つ**のステップを追加する必要があります。

ステップ	待つ
説明	ロボットが把持しても、何も把持しないことを防ぎます。
操作	ステップライブラリから「待つ」を選択し、プロジェクト編集エリアにドラッグして「把持待ち」という名前を付けます。
パラメータ設定	待ち時間 を 1000ms に設定します。
サンプル図	

経路点から離れる位置に移動

ロボットがワークを把持したら、**相対移動**ステップを使用してロボットを経路点から離れる位置に移動させます。

ステップ	相対移動
説明	ビジョン結果に従って移動します。
操作	ステップライブラリから「相対移動」を選択し、プロジェクト編集エリアにドラッグして「経路点から離れる」という名前を付けます。
パラメータ設定	移動基準 を次の経路点に対してに設定し、 目標点タイプ をロボットハンドに設定し、 Z軸 の座標を -200mm に設定します。
サンプル図	

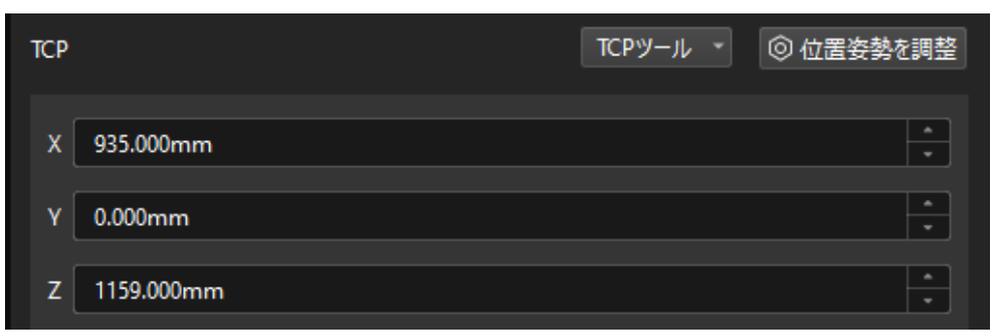
ワークの配置

ロボットがワークを把持したら、次の2つの手順でワークを配置ボックスに配置します。

1. 手順1：**移動**ステップを使用して、ロボットを配置ボックスの位置に移動させます。
2. 手順2：**DOを設定**ステップを使用して、ロボットハンドを閉じてワークを配置します。

詳細は以下の通りです。

ステップ	移動
説明	ロボットを配置ボックスの位置に移動させます。
操作	ステップライブラリから「移動」を選択し、プロジェクト編集エリアにドラッグして「配置位置」という名前を付けます。
パラメータ設定	TCP位置姿勢 を配置位置とするために、それを正しく設定する必要があります。

サンプル図	
-------	------------------------------------------------------------------------------------

ステップ	DOを設定
説明	ロボットハンドを閉じてワークを配置します。
操作	ステップライブラリから「DOを設定」を選択し、プロジェクト編集エリアにドラッグして「ハンド閉」という名前を付けます。
パラメータ設定	DO値を0に、DOポートを1に設定します。
サンプル図	

ロボットハンドが閉じるのに時間がかかるので、ロボットがワークの配置に失敗するのを防ぐために**待つ**ステップを追加する必要があります。

ステップ	待つ
説明	ロボットがワークの配置に失敗するのを防ぎます。
操作	ステップライブラリから「待つ」を選択し、プロジェクト編集エリアにドラッグして「配置待ち」という名前を付けます。
パラメータ設定	待ち時間を1000msに設定します。
サンプル図	

ステップ接続

上記のステップが追加されたら、順番につながります。ワークの把持と配置を繰り返し実行するために、「配置待ち」の出口を「初期位置」の入口につながります。

シミュレーションと実行

1. ツールバーの[シミュレート]をクリックすると、作成したMech-Vizプロジェクトがシミュ

レーションで実行されます。



2. シミュレーション効果が期待通りにある場合、Mech-Vizツールバーの[実行]ボタンをクリックして、ロボット実機を動作させます。



ロボットを低い速度で動作させることをお勧めします。緊急時には緊急停止ボタンを押してください。

3. 入門ガイド：ビジョンシステムによるピッキング（標準インターフェース通信）

本ガイドを読むことで、標準インターフェース通信を使用し、3Dビジョンシステムによる小型金属部品の把持アプリケーションを実装する方法を習得できます。

概要

- カメラ：Eye to Hand方式で取り付けられているMech-Eye PRO Mカメラ
- ロボット：YASKAWA_GP8
- ワーク：トラックリンク（小型金属ワーク）
- ロボットハンド：グリッパー（内径把持型）
- 使用されるソフトウェア：Mech-Vision 1.7.2バージョン、Mech-Viz 1.7.2バージョン、Mech-Center 1.7.2バージョン、Mech-Eye Viewer 2.1.0バージョン
- 通信方式：標準インターフェース通信



上記と異なるカメラ型番やロボットブランド、ワークを使用する場合、一部調整を行う必要があります。

ビジョンアプリケーションの実装

ビジョンアプリケーションの実装は、一般的に5つの段階に分けられています。



詳細は下表のとおりです。

番号	階段	説明
1	ビジョンシステムのハードウェア設置	Mech-Mindビジョンシステムのハードウェアの取り付けと接続、ソフトウェアのインストールと接続を完了します。
2	ロボット通信設定	標準インターフェースプログラムと設定ファイルをロボットシステムに読み込み、ビジョン側とロボット側との通信を構築します。

番号	階段	説明
3	ロボットハンド ・アイ・キャリ ブレーション	自動キャリブレーション（Eye to Hand）を実行し、カメラ座標系とロボット座標系との対応関係を求めます。
4	ワーク認識	ソリューションライブラリの「一般的な部品認識」を使用してワークの位置姿勢を計算し、ビジョン結果を出力します。
5	把持と配置を実 行	ロボットの把持・配置アプリケーションを作成し、取得されたビジョン結果に従ってワークの把持・配置を行います。

本節の説明は終わりです。次に、アプリケーションの実装を完了させてください。

3.1. ビジョンシステムのハードウェア設置

本ガイドを読むことで、Mech-Mindビジョンシステムの設置方法を習得できます。

Mech-Mindビジョンシステムの設置は、**梱包内容の確認** → **ハードウェアの取り付け** → **ネットワーク接続** → **ソフトウェアのアップグレード（オプション）** → **ビジョンシステムが画像を正常に取得できるか確認**の順で行う必要があります。

▶ <https://www.youtube.com/watch?v=IQ-anVFFEk> (YouTube video)

動画：ビジョンシステムのハードウェア設置

梱包内容の確認

1. お受け取りの際は、梱包に問題がないことを確認してください。
2. 梱包中にある『同梱包一覧』を確認し、商品や付属品に欠品や損傷がないことを確認してください。

下図は、梱包箱に含まれるものおよび付属品の例を示しています。下記の同梱品一覧は参照のみを目的としています。梱包中に入っているものは、実際の梱包の中にある『同梱包一覧』をご参照ください。



番号	種類	名前	機能
1	IPCと付属品	IPC	Mech-Mindソフトウェアシステム搭載
2		IPCの付属品	WIFIアンテナなどの付属品
3		IPCの電源ケーブルとアダプター	IPCの電源を入れる
4	カメラと付属品	Mech-Eye産業用3Dカメラ	画像取得
5		カメラ取扱説明書	Mech-Eye産業用3Dカメラの取扱説明書
6		カメラの付属品	カメラの取り付け

番号	種類	名前	機能
7	その他の付属品	ドングル	ソフトウェアのライセンス供与
8		キャリブレーションボード	カメラキャリブレーション
9		フランジ	キャリブレーションボードの取り付け
10		カメラDC電源ケーブル（標準仕様20メートル）	カメラをレール電源に接続（オプションでより長い電源ケーブルも利用可能）
11		カメラのLANケーブル（標準仕様20メートル）	カメラをIPCに接続（オプションでより長いLANケーブルも利用可能）
12		レール電源（標準仕様）	Mech-Eye産業用3Dカメラの電源を入れます（オプションで電源アダプターも利用可能）
13	同梱品一覧		梱包に含まれるすべてのものと付属品を一覧表示します



万一、商品紛失・破損等が発生した場合は、Mech-Mind株式会社にお問合せください。

その他のものを準備

本ガイドでは、カメラ梱包に入っているものの他に、下表に示すようなものを自分で用意する必要があります。

名前	機能
モニター	IPCの画面表示
HDMIケーブル	IPCとモニターを接続
RJ45 LANケーブル	IPCとコントローラを接続



本ガイドでは、IPCはRJ45 LANケーブルでコントローラと接続し、カメラのLANケーブルでカメラと接続します。ルータ経由でIPCをロボットコントローラに、IPCをカメラに接続することもできます。

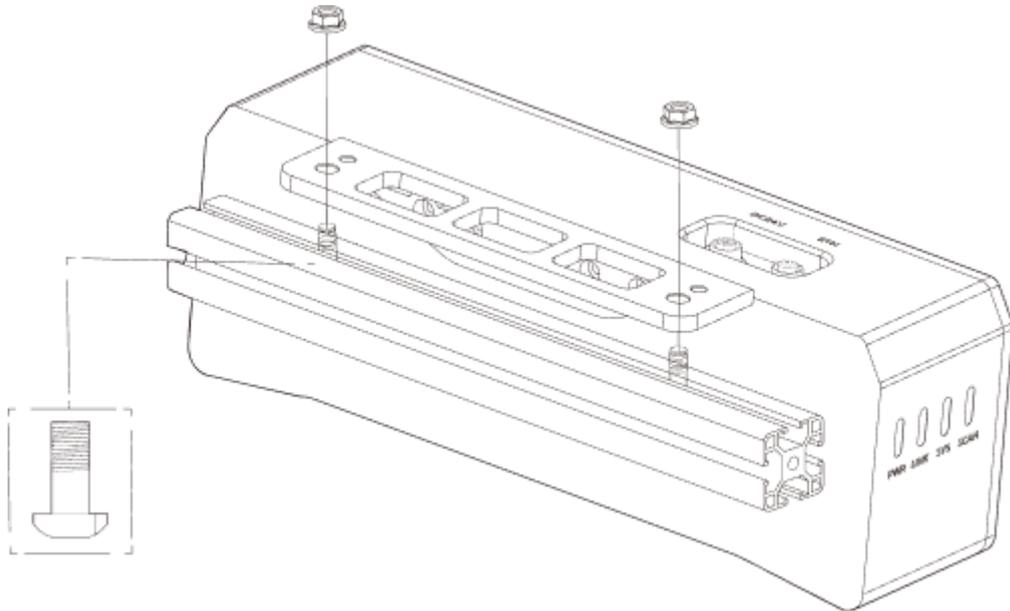
ハードウェアの取り付け

カメラの取り付け

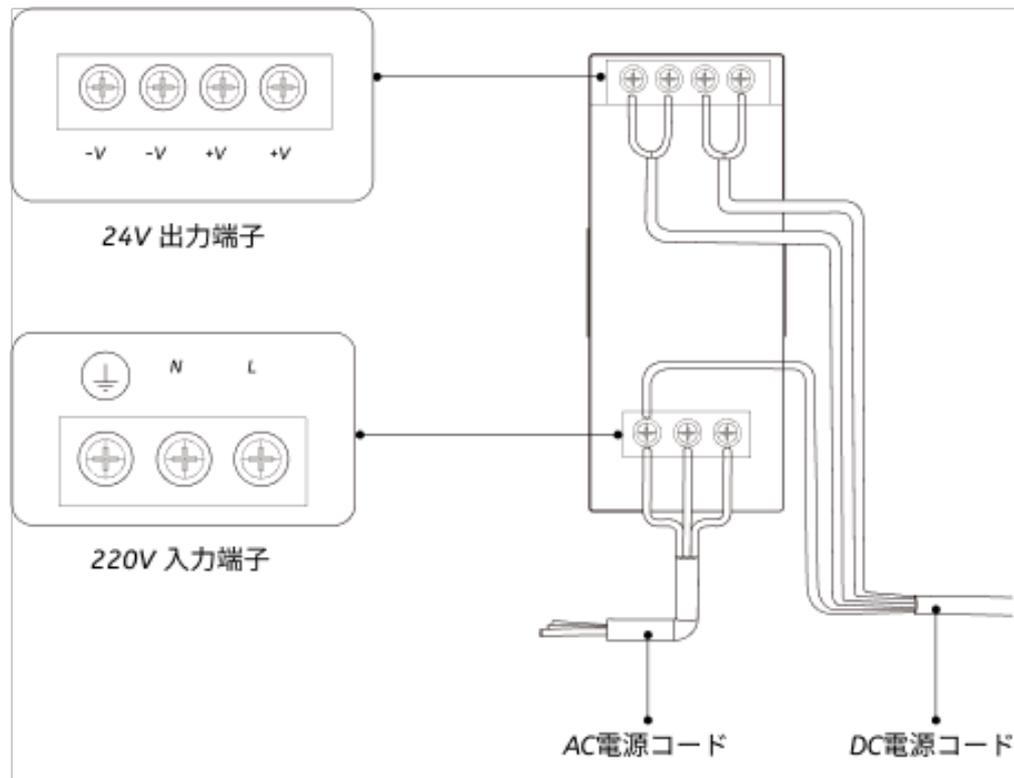


本ガイドでは、カメラはカメラスタンドに設置されています（Eye to Hand方式）。また、カメラをロボット先端に取り付けることも可能です（Eye in Hand方式）。

1. 付属品からカメラを取り付けるためのネジとスパナを探します。
2. 下図に示すように、レンチを使用して2本のネジを締め、カメラを固定します。

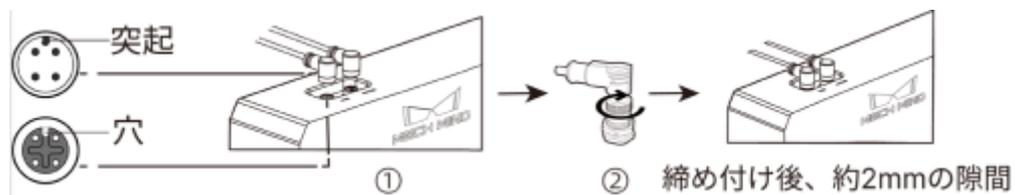


3. カメラ取り付け後、レンズ保護フィルムを剥がします。
4. レール電源でカメラの電源を入れます。
 - DC電源ケーブルの接続：
 - +Vを24V出力端子の+Vに接続します。
 - -Vを24V出力端子の-Vに接続します。
 - PEを220Vの入力端子Ⓧに接続します。



5. カメラのLANケーブルを取り付けます。

カメラのLANケーブルの航空コネクタの突起をETHポートの開口部に差し込み、ナットを締めます。



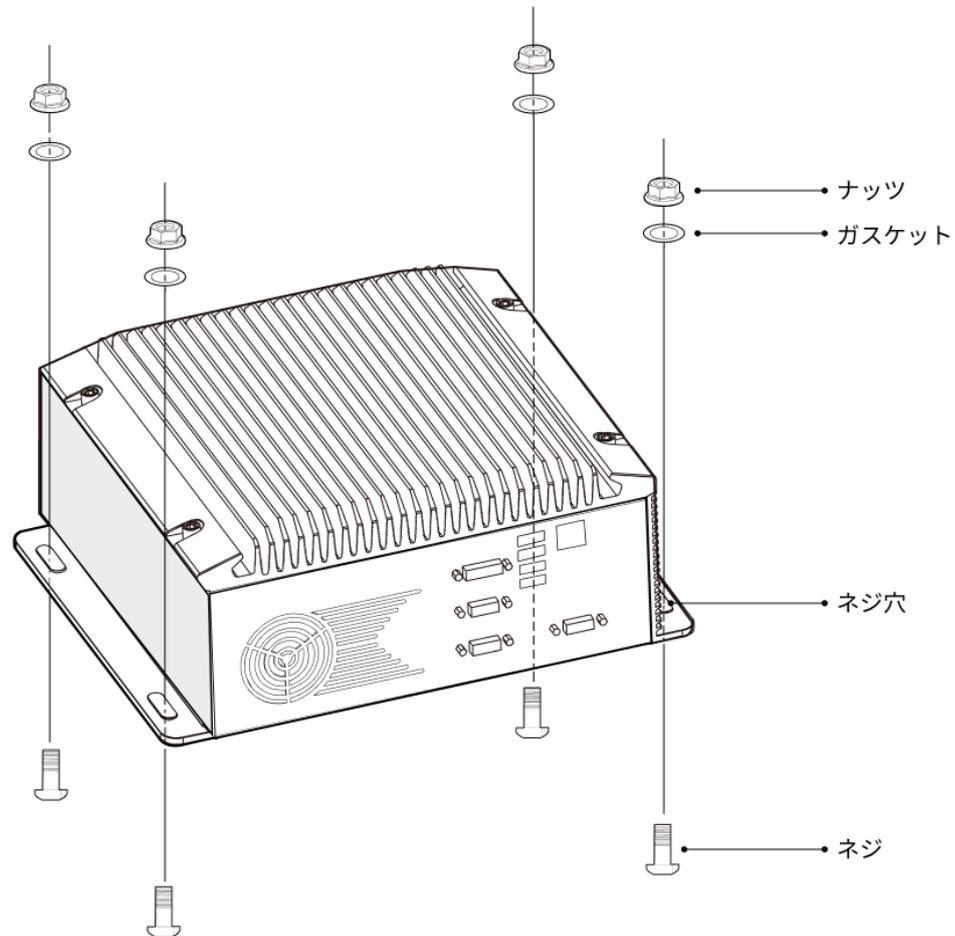
IPCの取り付け



IPCは一般的にコントローラに設置されます。IPCの設置環境は、放熱性、通気性、防塵効果が必要であり、LANケーブル、HDMIケーブル、USBポートの設置やメンテナンスがしやすい場所を選択する必要があります。

IPCを取り付けるには、以下の手順を実行します。

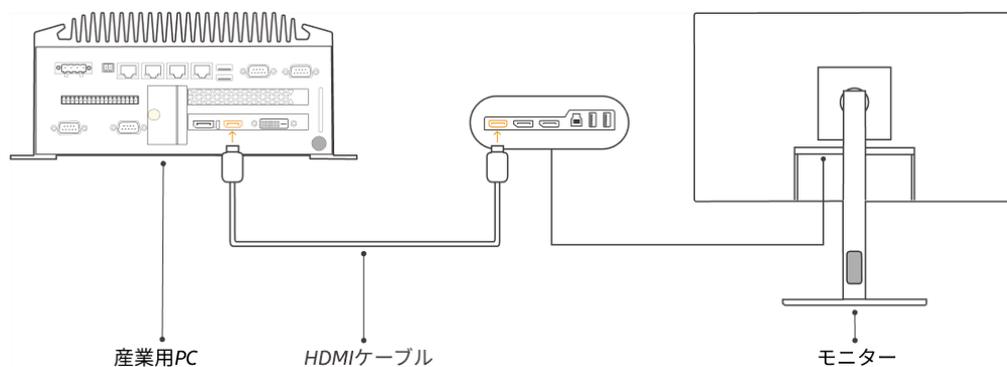
1. スパナ、ネジ、ナット、スペーサーは各自でご用意ください。
2. コントローラにIPCの取り付け穴がある場合は、下図のようにネジ、スペーサー、ナットの順に入れ、スパナでナットを締めて、コントローラにIPCを取り付けます。



コントローラの位置が移動しない場合は、この手順を省略してIPCをコントローラに直接設置することができます。

3. IPCとモニターをHDMIケーブルで接続します。

下図のように、HDMIケーブルの一端をモニターのHDMIポートに、もう一端をIPCのHDMIポートに差し込みます。



4. 電源アダプタでIPCの電源を入れます。

電源アダプタの電源プラグをIPCの電源入力ポートに差し込み、電源アダプタのもう一方の端を電源に接続します。

5. ドングルを差し込みます。

ドングルをIPCのUSBポートに差し込みます。

6. IPCの電源が入った後、IPCを起動します。

- IPCの正常な起動後は、電源インジケータは常に点灯している必要があります。
- IPCが起動しない場合は、Mech-Mindにお問い合わせください。

ネットワーク接続

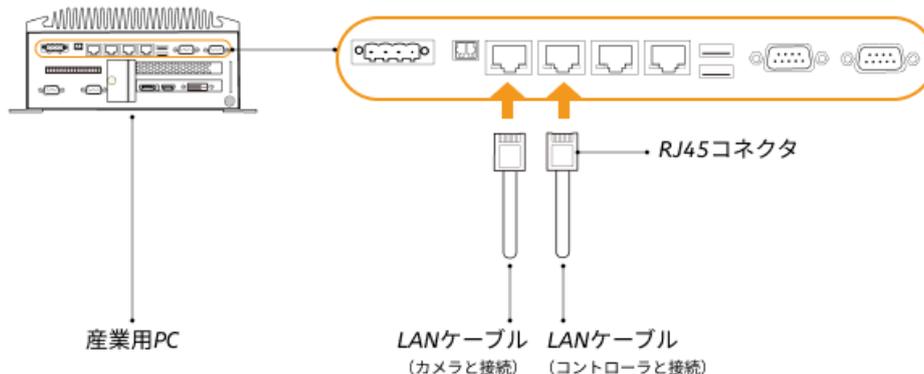
以下では、IPCとカメラとのネットワーク接続、IPCとロボットとのネットワーク接続について説明します。

以下の操作で、次のIPアドレスに従ってネットワークを設定します。実際のネットワーク環境に応じて調整してください。

機器		IPアドレス
IPC	カメラに接続されているLANポート	192.168.100.10
	コントローラに接続されているLANポート	192.168.200.10
カメラ		192.168.100.20
ロボット		192.168.200.20 (ロボットが設定済み)

IPCとカメラ、IPCとロボットコントローラを接続

1. カメラが接続されているLANケーブルのもう一方の端をIPCのLANポートに差し込みます。



2. 両端にRJ45コネクタが付いたLANケーブルを使用して、LANケーブルの一端をIPCのLANポートに、もう一端をロボットコントローラのLANポートに差し込みます。

IPCのIPアドレスを設定

1. IPCで、**コントロールパネル** > **ネットワークとイーサネット** > **ネットワークと共有センター** > **アダプターの設定の変更**を選択すると、**ネットワークの接続**の画面が表示されます。
2. カメラが接続されているLANポートを選択し、右クリックして**名前の変更**を選択し、「To_camera」など、ネットワーク接続を示すポートに名前を変更します。
3. カメラが接続されているLANポートを選択し、右クリックして**プロパティ**を選択し、**イーサネットのプロパティ**の画面に入ります。
4. **イーサネットプロトコルバージョン 4 (TCP/IPv4)** を選択して[**プロパティ**]をクリックし、**Internet プロトコルバージョン 4 (TCP/IPv4) のプロパティ**の画面に入ります。
5. **次のIPアドレスを使う**を選択し、**IPアドレス**を「192.168.100.10」、**サブネットマスク**を「255.255.255.0」、**デフォルトゲートウェイ**を「192.168.100.1」に設定してから、[**OK**]をクリックします。



6. 手順2~5を繰り返し、ロボットコントローラが接続されているLANポートの名前を変更し（例：To_robot）、LANポートのIPアドレスを設定します。例えば、LANポートのIPアドレスは「192.168.200.10」です。



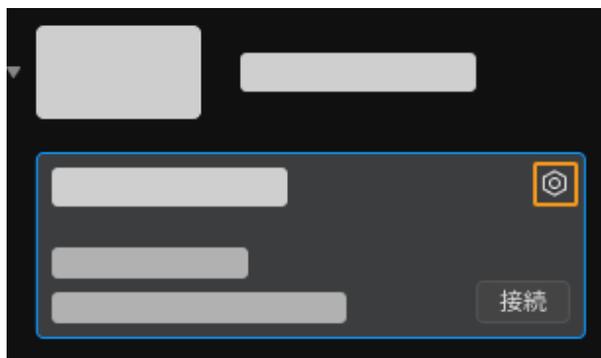
ロボットコントローラに接続するIPCのLANポートのIPアドレスは、ロボットのIPアドレスと同じネットワークセグメント内にある必要があります。

カメラのIPアドレスを設定

1. IPCのデスクトップ上にある  アイコンをダブルクリックし、Mech-Eye Viewerを起動しま

す。

- 検出されたカメラリストからカメラを選択し、マウスをカメラ情報バーに移動すると、が表示されます。クリックすると**IP設定画面**に入ります。



カメラが検出できない、接続できない場合は、[カメラのトラブルシューティング](#)を参照し、解決してください。

- カメラエリアで、**静的IPに設定**を選択し、**IPアドレスクラス**を「クラスC 192.168.x.x」、**IPアドレス**を**192.168.100.20**、**サブネットマスク**を「255.255.255.0」に設定してから、**[適用]**をクリックします。



カメラのIPアドレスは、カメラが接続されているIPCのLANポートのIPアドレスと同じネットワークセグメント内にある必要があります。

ネットワーク接続をテスト

- ショートカットキー **Win + R** で**実行画面**を表示します。
- 名前に**cmd**と入力し、**[OK]**をクリックします。

3. コマンドウィンドウにping XXX.XXX.XX.XXと入力し、[Enter]をクリックしてコマンドを実行します。



XXX.XXX.XX.XXは、実際に設定されたカメラまたはロボットのIPアドレスに変更します。

ネットワークの接続が正常であれば、次のようなメッセージが表示されます。

```
XXX.XXX.XX.XXにPingを送信しています 32バイトのデータ:
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128
```

ソフトウェアのアップグレード（オプション）

Mech-Mindから購入したIPCには、Mech-Mindソフトウェアシステムがプリインストールされています。

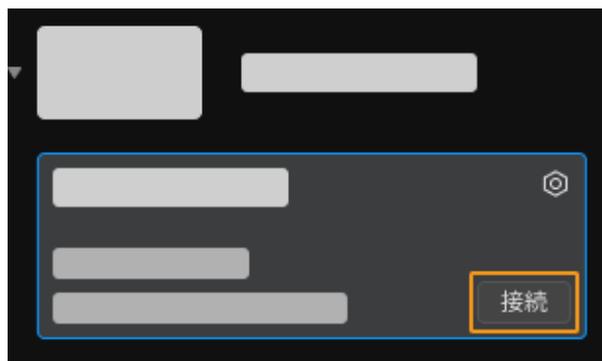
本機のソフトウェアが最新バージョンであるかどうかを確認してください。ソフトウェアがすでに最新バージョンである場合は、以下の内容をスキップしてください。ソフトウェアが最新バージョンでない場合は、以下の内容を参照して、ソフトウェアを最新バージョンにアップグレードしてください。

- [Mech-Eye SDKソフトウェアのダウンロードとインストール](#)
- [Mech-Vision、Mech-Viz、Mech-Centerソフトウェアのダウンロードとインストール](#)

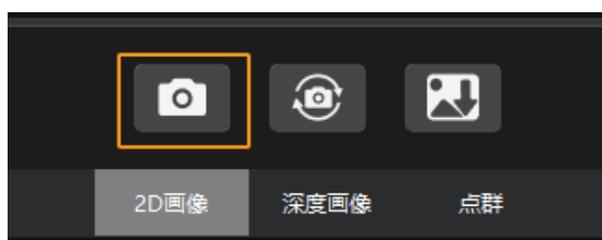
ビジョンシステムが正常に画像を取得できるか確認

IPC、カメラとロボット間のネットワークにアクセスできることを確認した後、以下の手順でビジョンシステムが正常に画像を取得できることを確認します。

1. ワークをカメラの視野中心に置き、エッジと最高層にあるワークが視野に入ることを確認します。
2. IPCのデスクトップにある  アイコンをダブルクリックしてMech-Eye Viewerを起動します。
3. 検出されたカメラからカメラを選択して[接続]をクリックします。

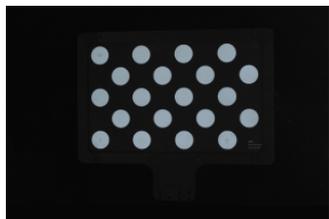
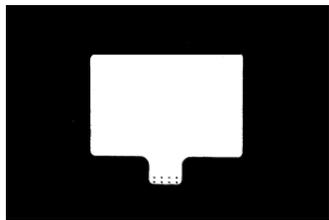
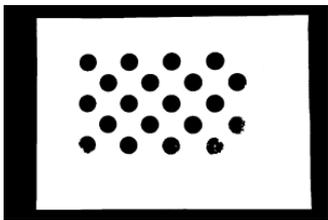
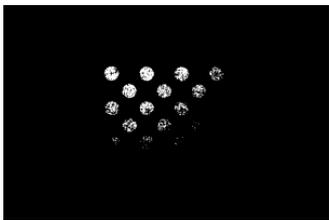


4. カメラ接続後、[一回キャプチャ]をクリックします。



5. 撮影した画像の品質が条件を満たしていることを確認します。

- 2D画像：明らかな露出過度（白すぎて対象物が見えない）、露出不足（暗すぎて対象物のディテールが判別できない）がありません。
- 深度画像と点群：対象物に激しい抜けがありません。

	正常	露出過度	露出不足
2D画像			
点群			



取得した画像が要件を満たしていない場合、[Mech-Eye Viewer](#)でパラメータを調整してください。

以上でビジョンシステムのハードウェアの構築方法の説明は終わりです。

3.2. ロボット通信設定

本ガイドでは、標準インターフェースプログラムをYaskawaロボットに読み込む方法を習得

し、Mech-Mindビジョンシステムとロボット間の標準インターフェース通信設定を行います。



- 標準インターフェースプログラムの読み込みは、ロボット標準インターフェースプログラムと設定ファイルをロボットシステムにロードし、ビジョンシステムとロボット間の標準インターフェース通信を確立します。
- それ以外のロボットを使用している場合は、[標準インターフェース通信](#) を参照してロボットの標準インターフェース通信設定を行います。

▶ <https://www.youtube.com/watch?v=2ERGMtWkZV8/PLVcMd7cW2rXVtrAejMyVQni2dUDv8bxje> (YouTube video)

動画：ロボット通信設定（標準インターフェース通信）

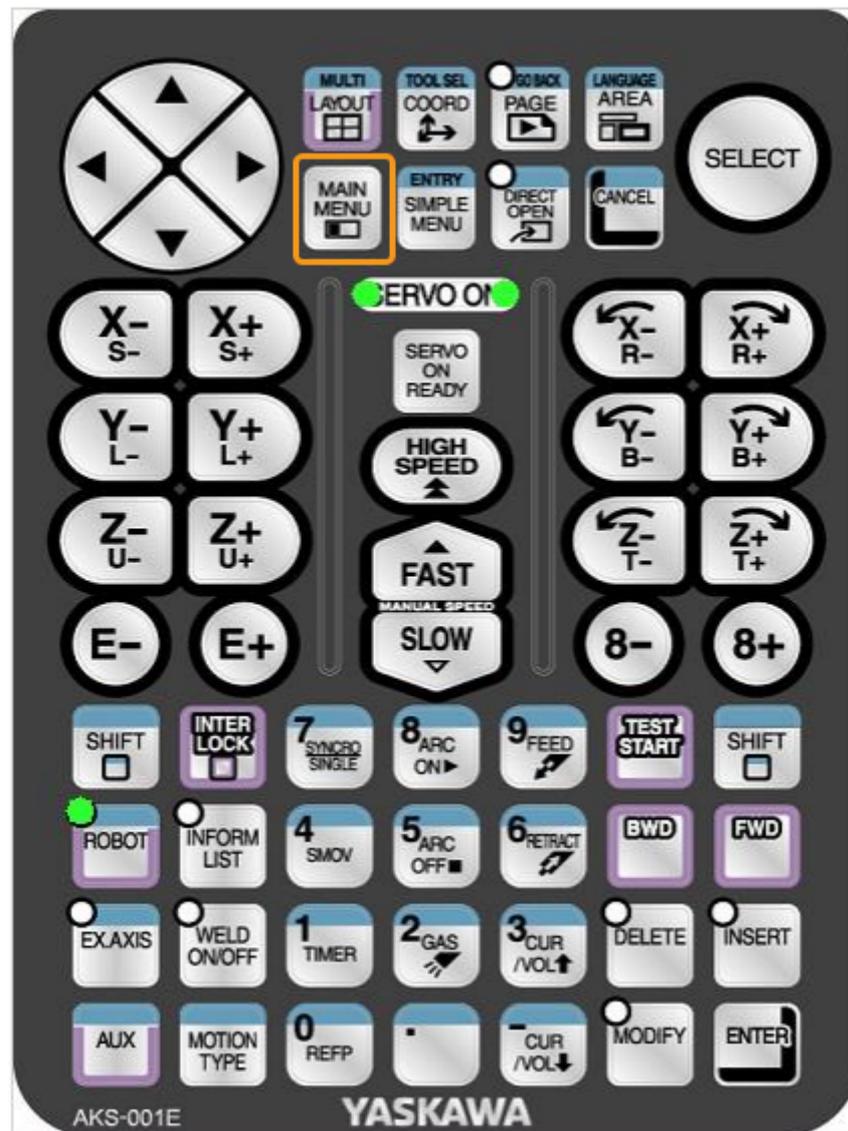
事前準備

ハードウェアとソフトウェアバージョンを確認

- Yaskawaの6軸ロボットを使用していることを確認します。本ガイドでは、YASKAWA_GP8ロボットを使用しています。
- ロボットコントローラはYRC1000、YRC1000microまたはDX200であることを確認します。本ガイドでは、YRC1000のコントローラを使用しています。
- Yaskawaロボットのイーサネット機能が有効になっていることを確認します。

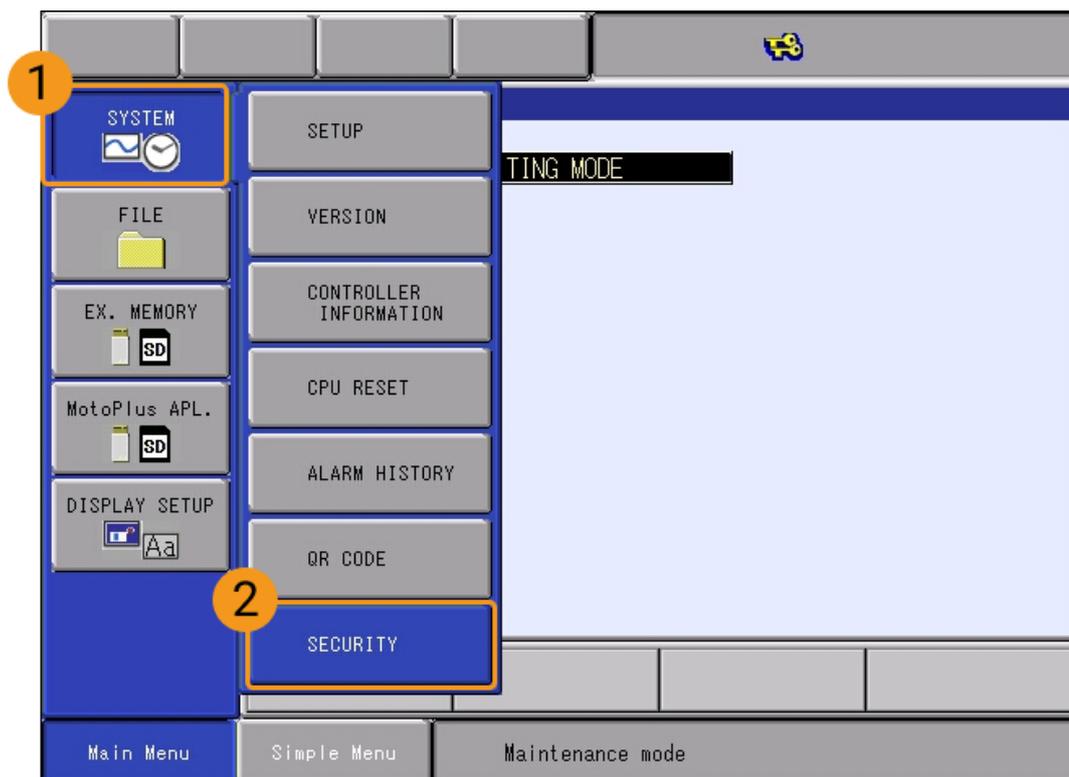
▼ (クリックして展開) 詳細方法

- a. ティーチペンダントの **MAIN MENU** キーを押しながらロボットの電源を入れると、**メンテナンスモード** になります。

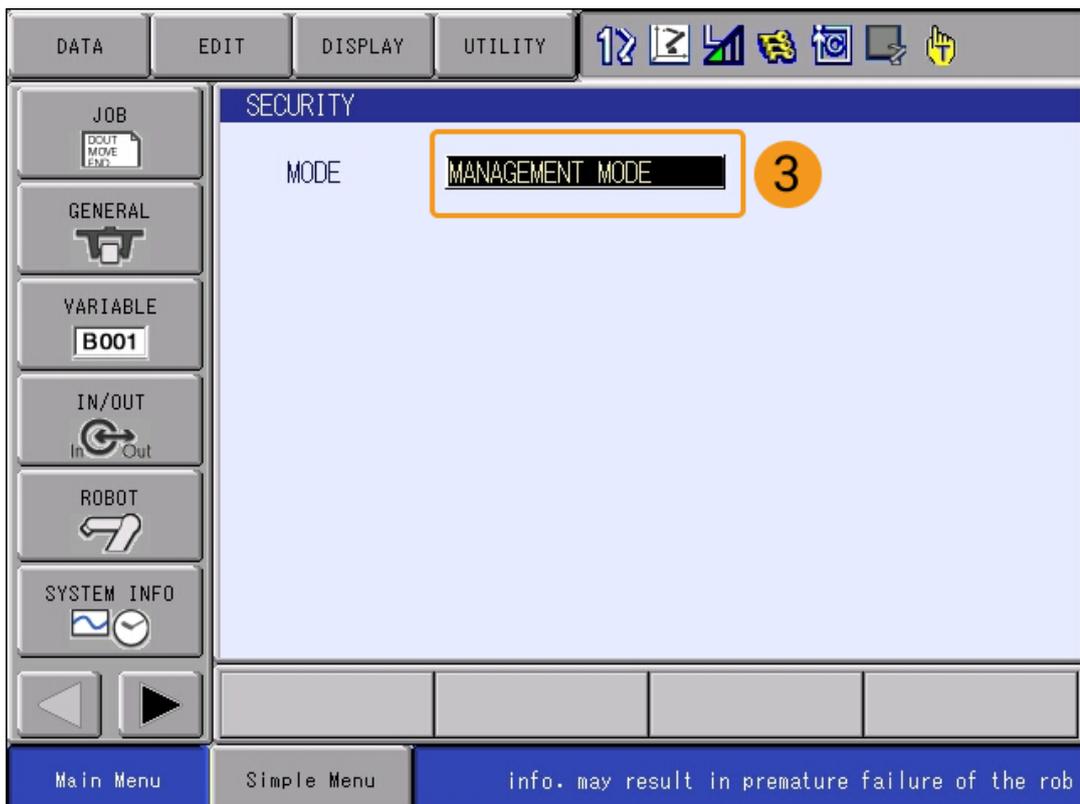
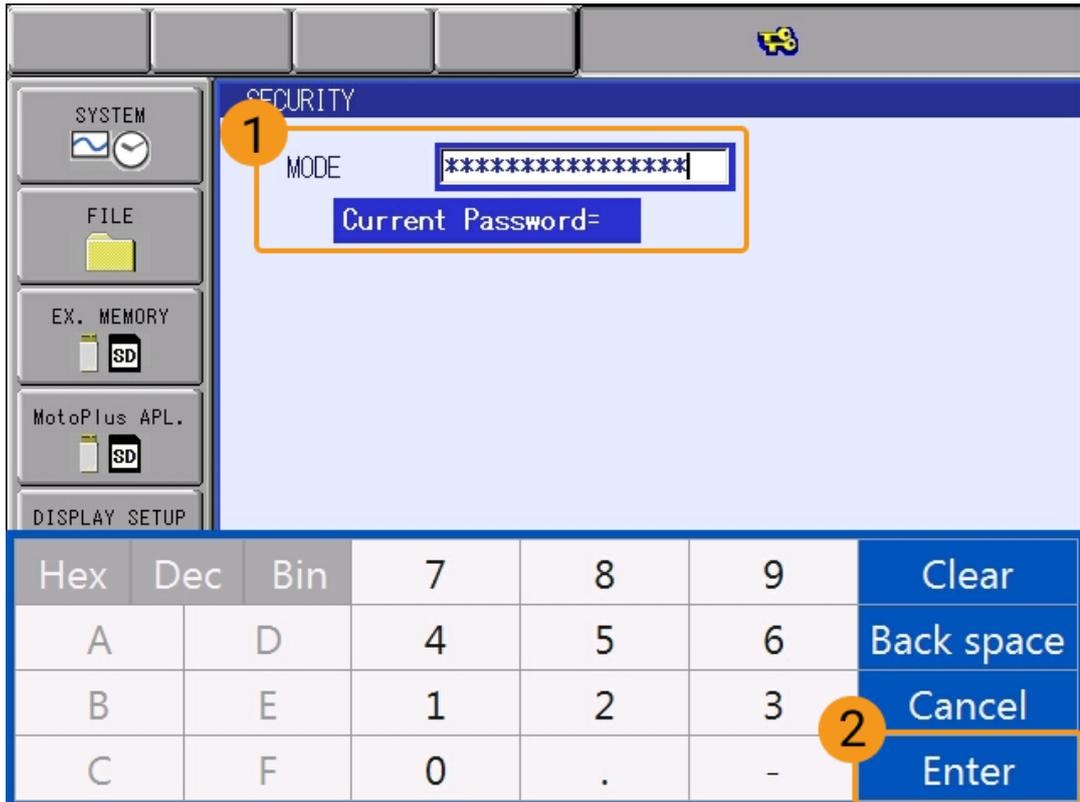


ロボットがすでに起動している場合は、**MAIN MENU** キーを押しながらロボットを再起動します。

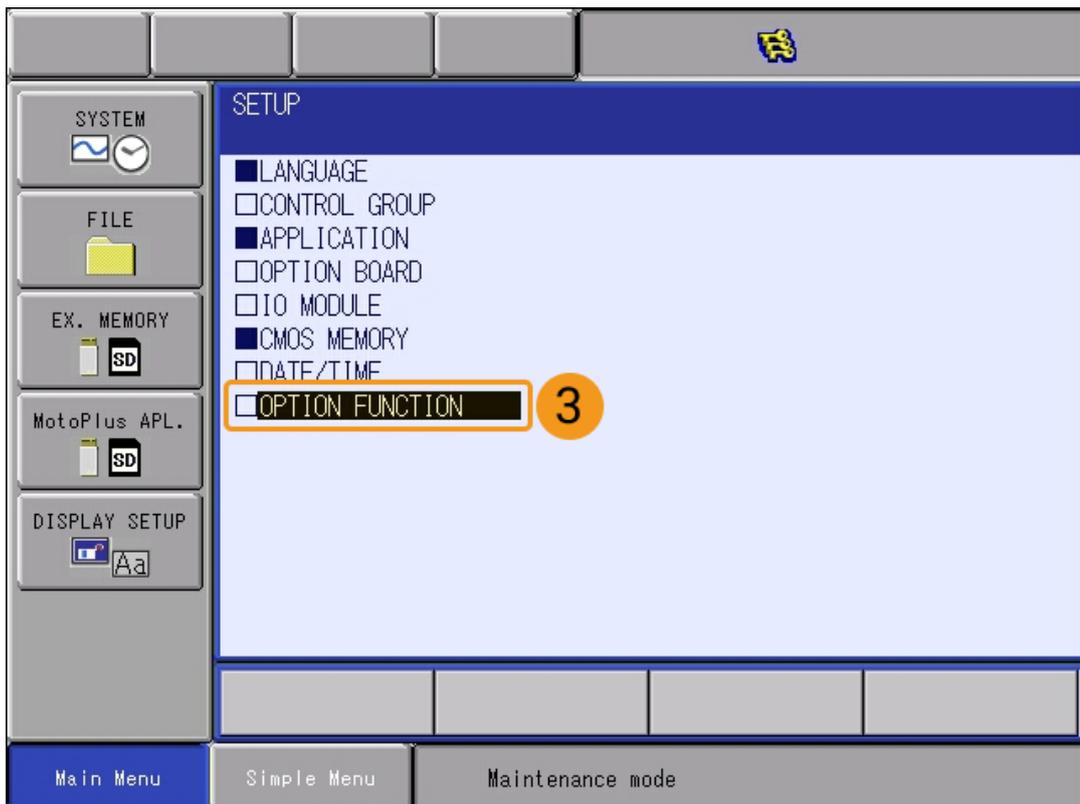
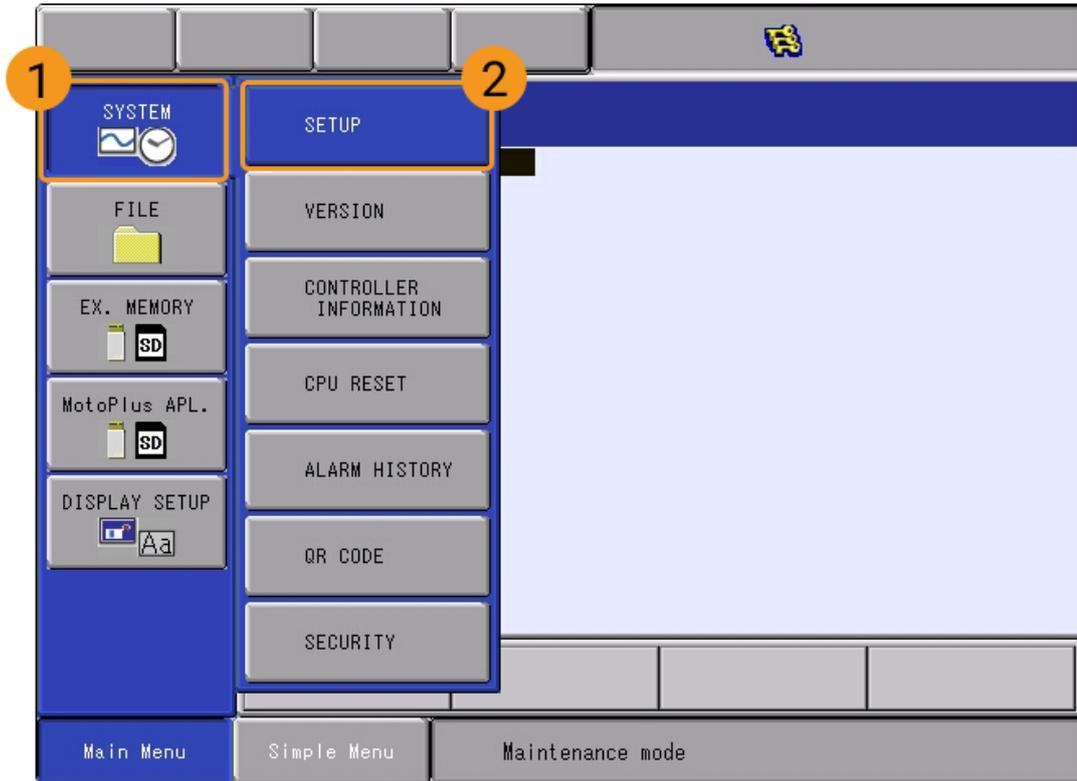
- b. メンテナンスモードでは、**SYSTEM > SECURITY > MANAGEMENT MODE**を選択します。

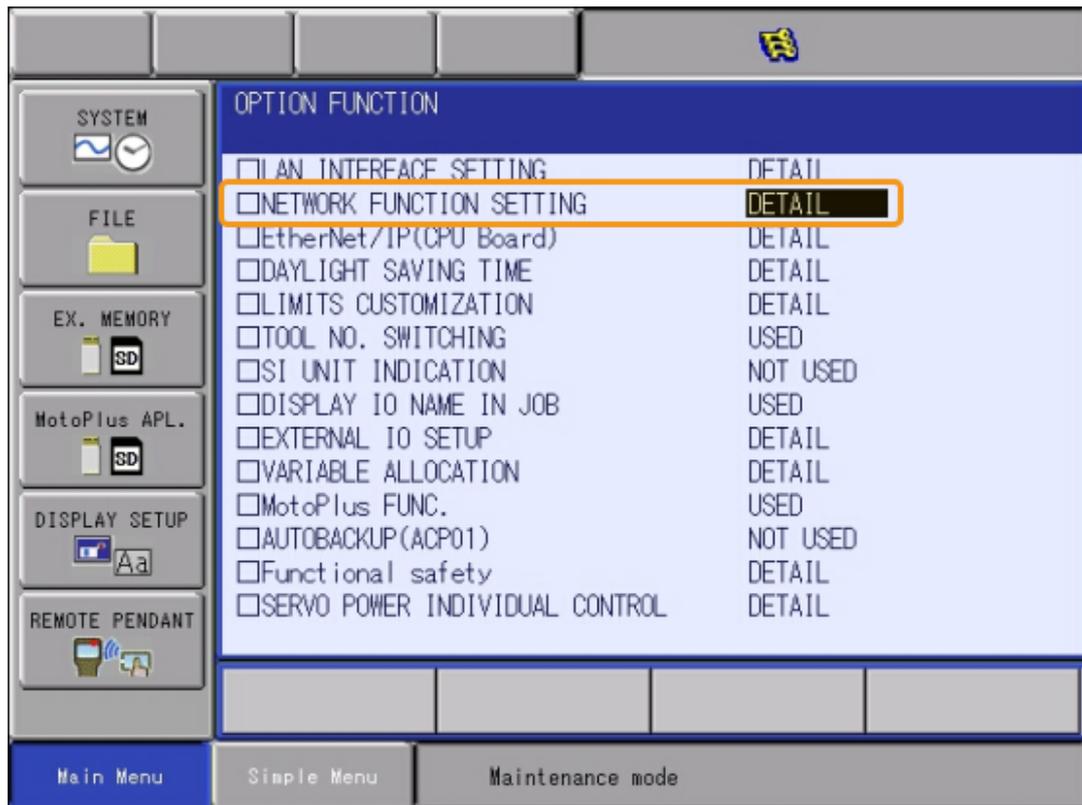


- c. パスワードを入力し、[ENTER]をクリックして **MANAGEMENT MODE** になります。デフォルトのパスワードは16個の「9」です。

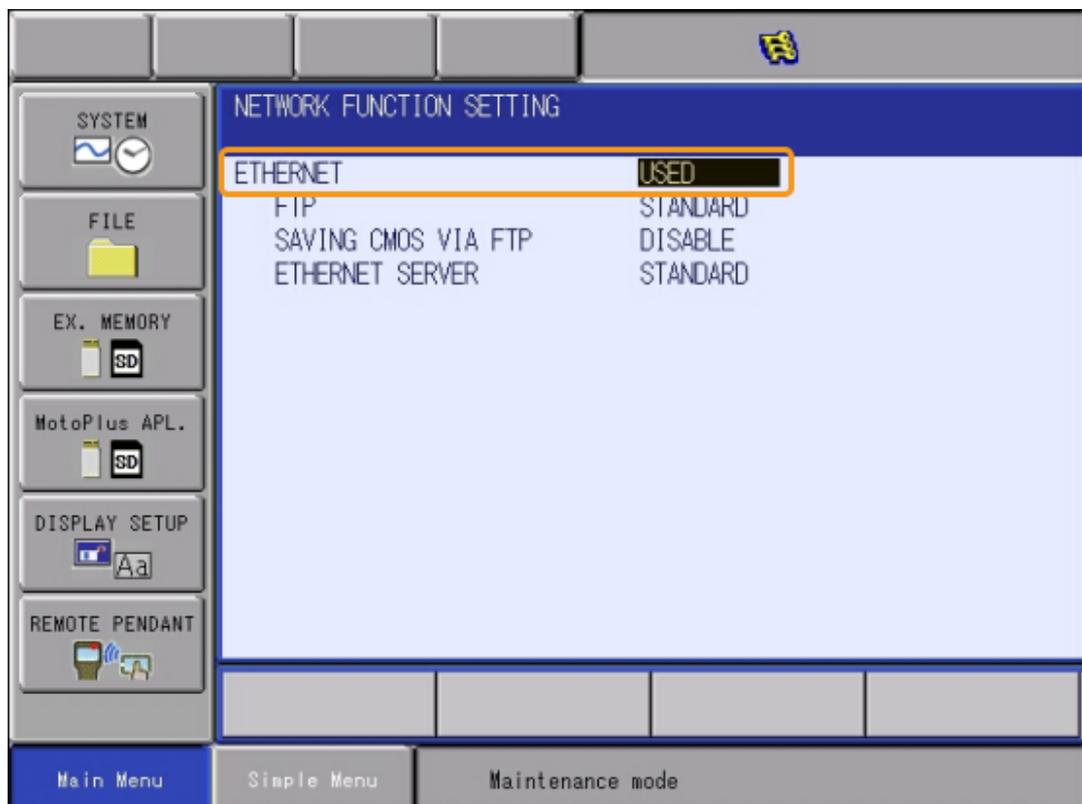


- d. Main Menu から **SYSTEM** > **SETUP** > **OPTION FUNCTION** をクリックし、オプション機能の画面に入ります。次に、**NETWORK FUNCTION SETTING** の **DETAIL** をクリックし、ティーチペンダントの **SELECT** キーを押してネットワーク機能設定の画面に入ります。



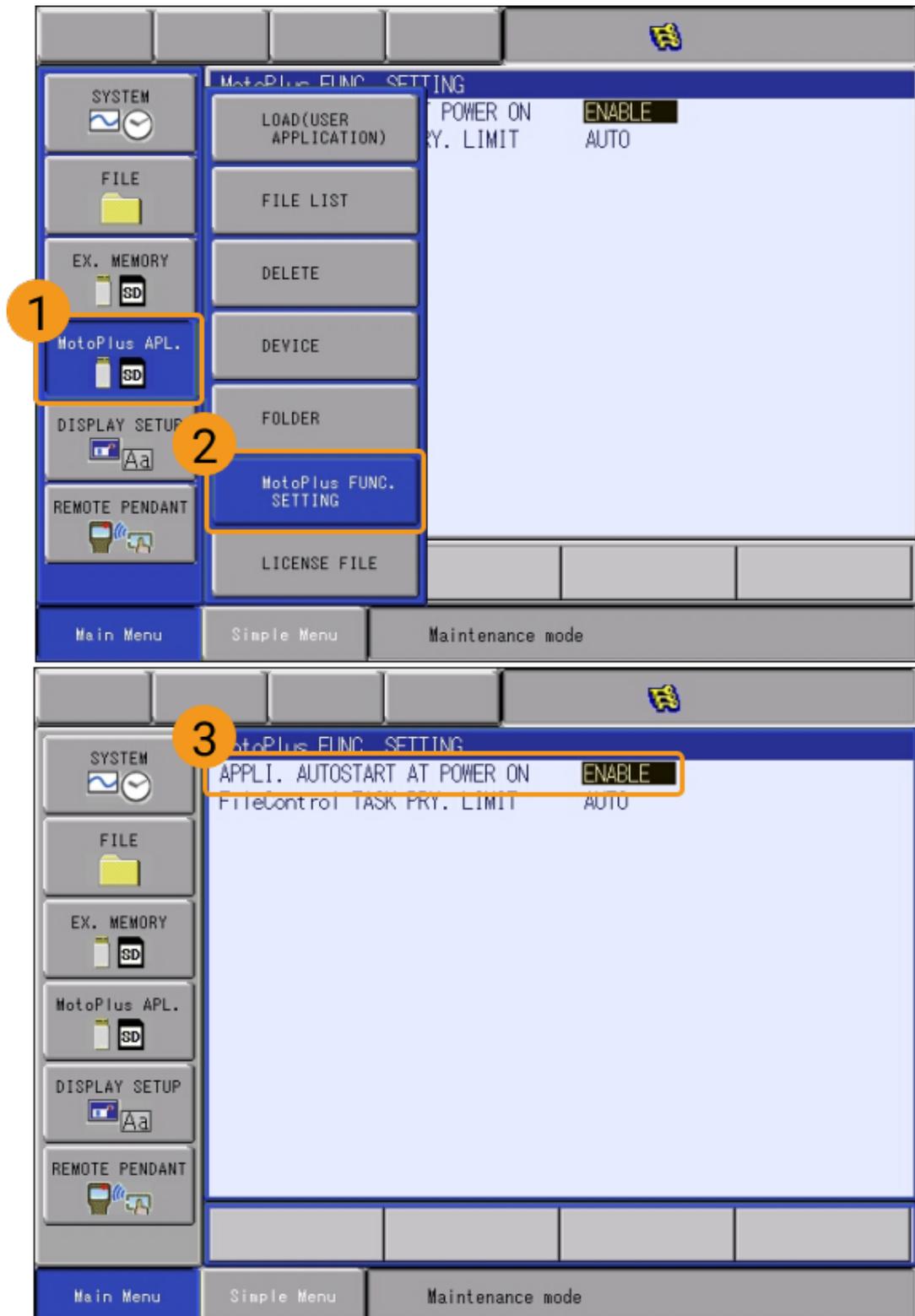


e. イーサネットが **USED** と表示されているか確認します。

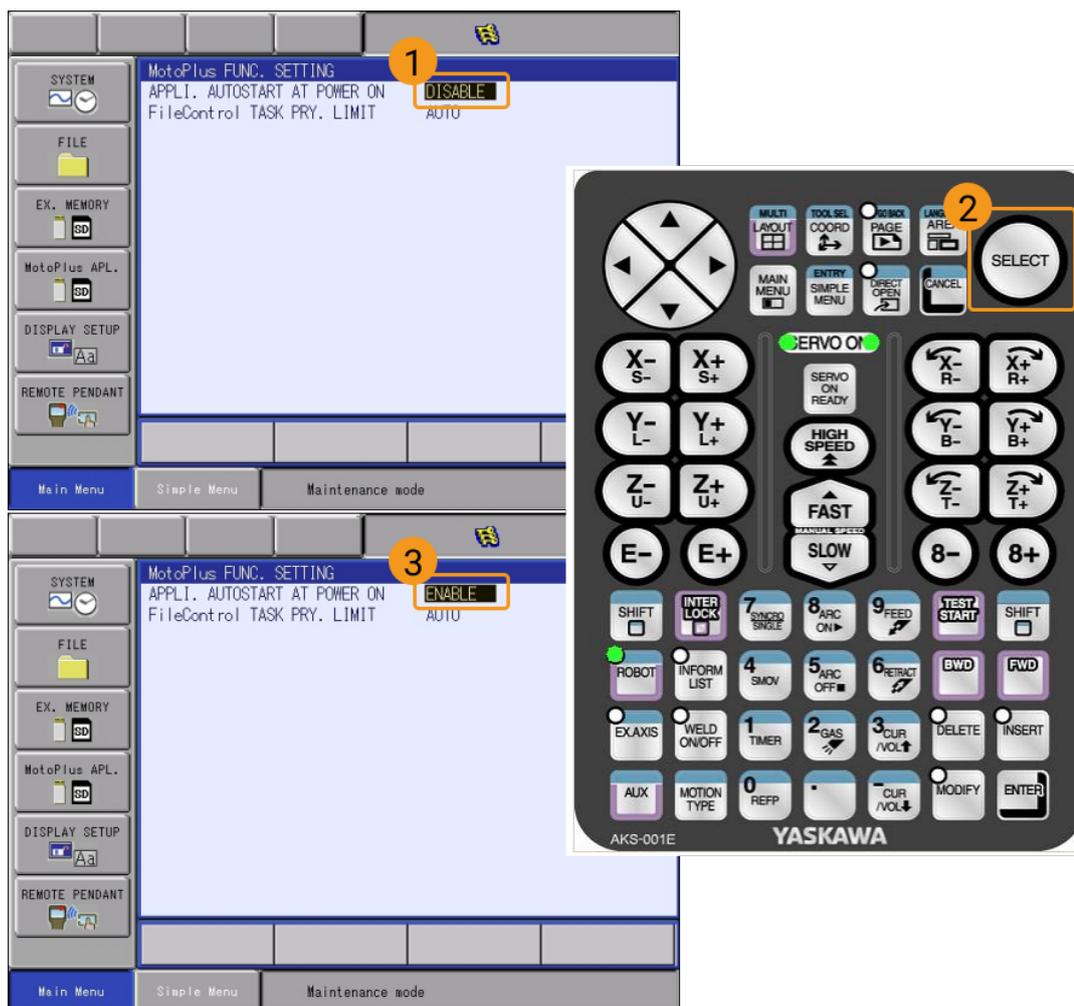


ステータスが「UNUSED」の場合は、ロボットメーカーにお問い合わせください。

- YaskawaロボットのMotoPlus機能が有効になっていることを確認します。
 - ▼ (クリックして展開) 詳細方法
 - a. メンテナンスモードの**管理モード**では、**Main Menu**から**MotoPlus APL.** > **MotoPlus FUNC.SETTING**を選択します。
 - b. **MotoPlus FUNC.SETTING**の画面で、**APPLI. AUTOSTART AT POWER ON** ステータスが **ENABLE** と表示されているか確認します。



- c. 「DISABLE」である場合、**DISABLE**を選択し、ティーチペンダントの **SELECT** キーを押すと、「ENABLE」になります。



上記の条件を満たしていない場合は、標準インターフェース通信ができませんので、ロボットメーカーにお問い合わせください。

ネットワーク接続

1. IPCのLANケーブルのもう一端をコントローラのCPUボードのLAN2（CN106）ポートに接続します。

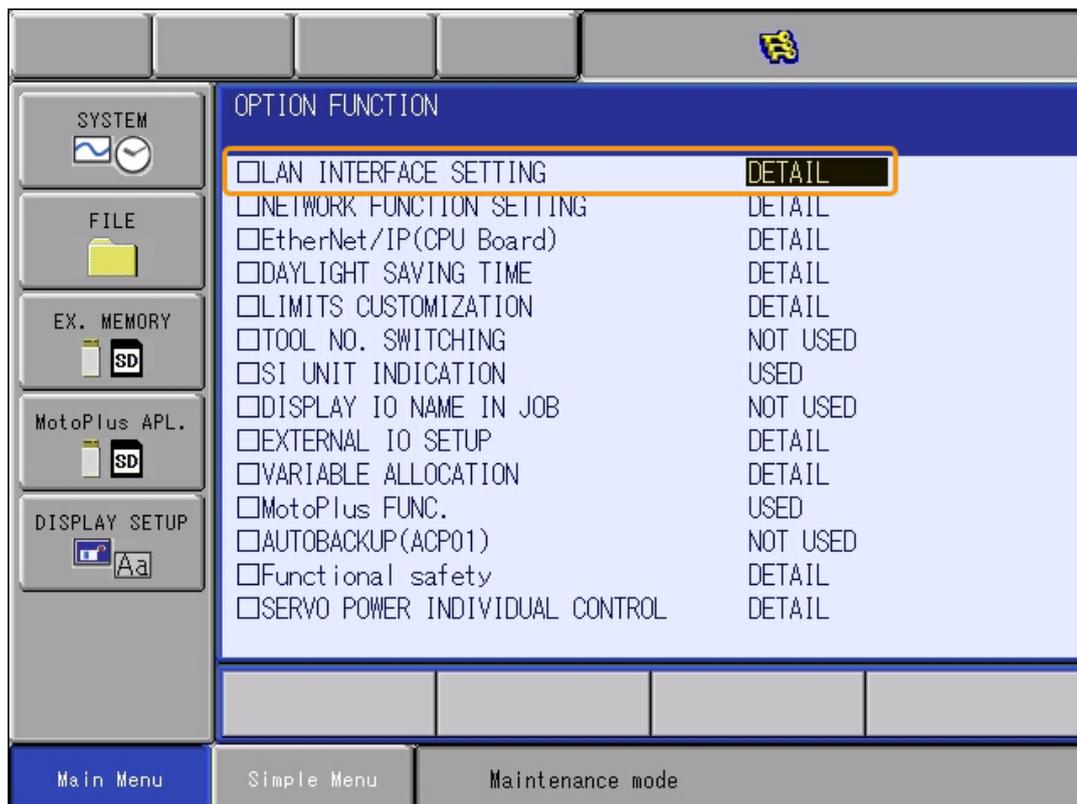


DX200シリーズのコントローラを使用している場合は、IPCのLANケーブルのもう一端をコントローラのCPUボードのCN104ポートに接続します。

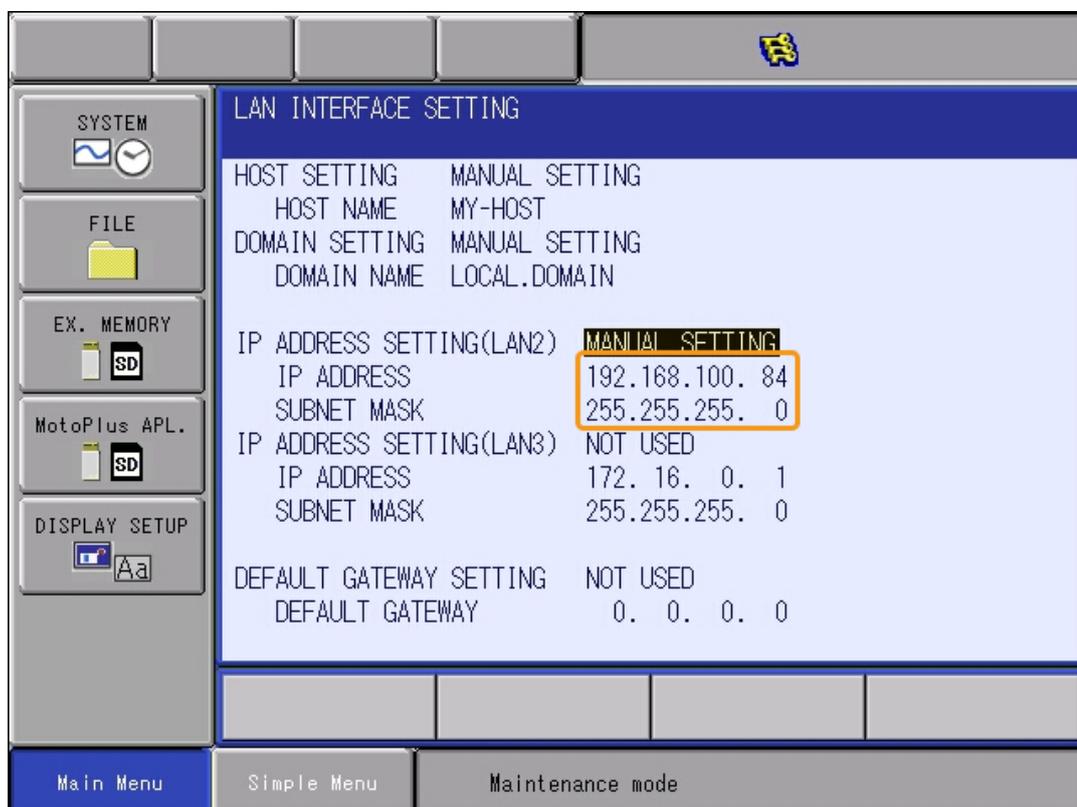
2. YaskawaロボットのIPアドレスが、IPCのIPアドレスと同じネットワークセグメントにあることを確認します。

▼ (クリックして展開) 詳細方法

- a. メンテナンスモードの管理モードでは、Main Menuから**SYSTEM** > **SETUP** > **OPTION FUNCTION**をクリックし、オプション機能の画面に入ります。次に、**LAN INTERFACE SETTING**の**DETAIL**をクリックし、ティーチペンダントの**SELECT**を押すと、**LANインターフェース設定**の画面が表示されます。



- b. ロボットのIPアドレス（LAN2ポートのIPアドレス）を確認します。



- c. ロボットのIPアドレスが、IPCのIPアドレスと同じネットワークセグメントにあることを確認します。同じネットワークセグメントにない場合、IPCのIPアドレスを設定

を参照してIPCのIPアドレスを設定します。

Mech-Visionソリューションの作成と保存

1. Mech-Visionを起動すると、下図のようなようこそ画面が表示され、Mech-Visionが正常に起動されたことを示します。



2. Mech-Visionのようこそ画面で [ソリューションライブラリから新規作成] をクリックし、ソリューションライブラリを開きます。



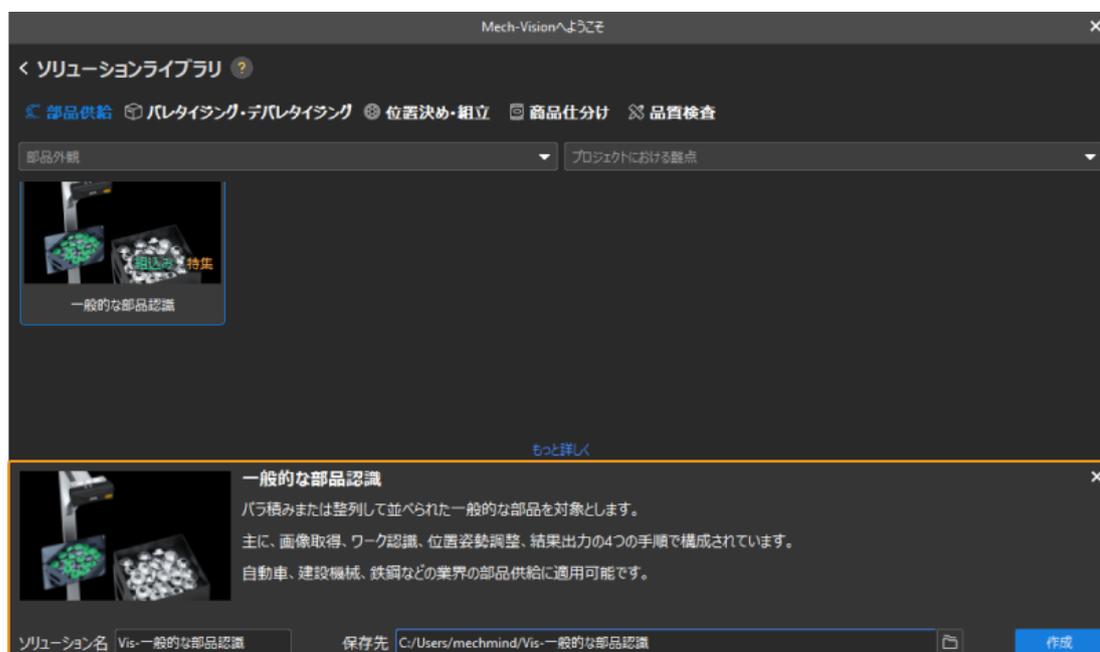
ソリューションライブラリには、異なる業界に適用可能なソリューションやプロジェクトのソースライブラリが含まれています。

3. ソリューションライブラリを開いた画面を下図に示します。一般的な部品認識プロジェクトを選択します。



ソリューションライブラリからこのプロジェクトが見つからない場合、画面下部にある [もっと詳しく] をクリックします。

4. このプロジェクトをクリックして選択すると、プロジェクトの関連情報は画面下部に表示されます。ソリューション名と保存先を設定したら、[作成] をクリックします。



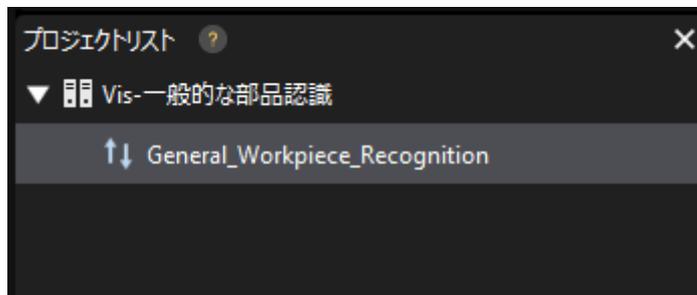
プロジェクトが作成されると、Mech-Visionメイン画面の左上にあるプロジェクトリストに、作成されたソリューションとプロジェクトが表示されます。



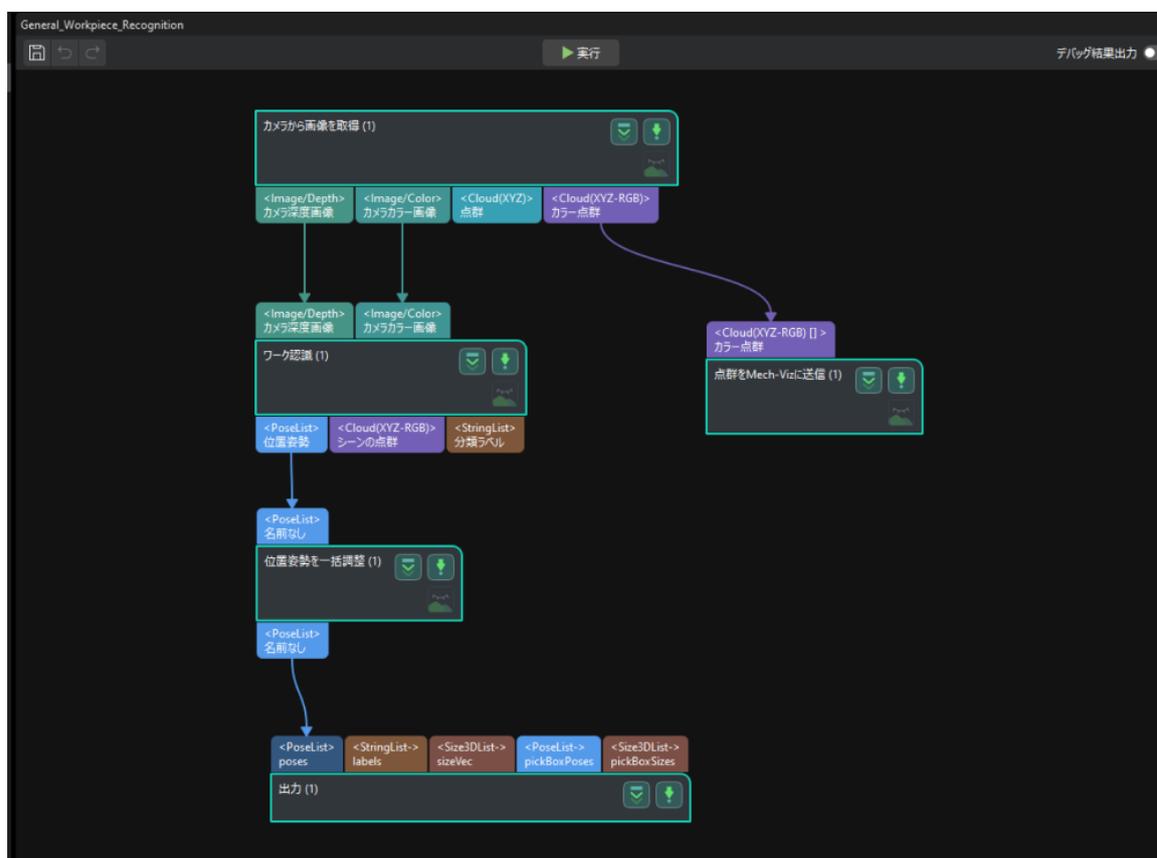
- **ソリューション**とは、ビジョンソリューションを実現するために必要なロボット

と通信、ビジョン処理、経路計画などの機能設定とデータの集りのことです。

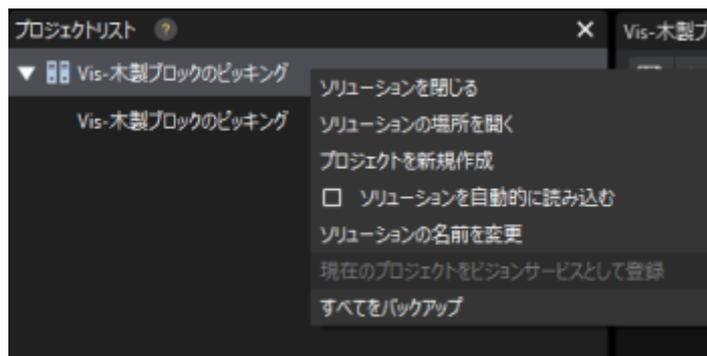
- **プロジェクト**とは、ソリューションにおけるビジョン処理のワークフローです。通常、1つのプロジェクトを含むソリューションで十分ですが、複雑な適用シーンでは複数のプロジェクトが必要になる場合があります。本ガイドに使用されるソリューションでは、1つのプロジェクトしか必要ありません。



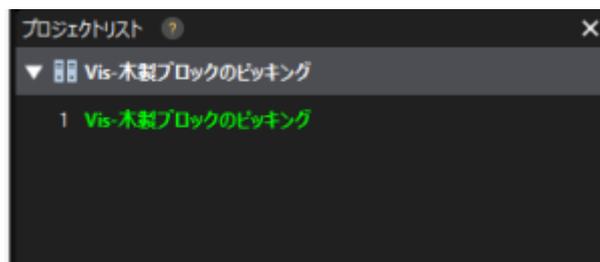
メイン画面中央のプロジェクト編集エリアには、「一般的な部品認識」というプロジェクトが表示されます。



5. プロジェクトリストでこのソリューションを右クリックし、[ソリューションを自動的に読み込む]にチェックを入れます。

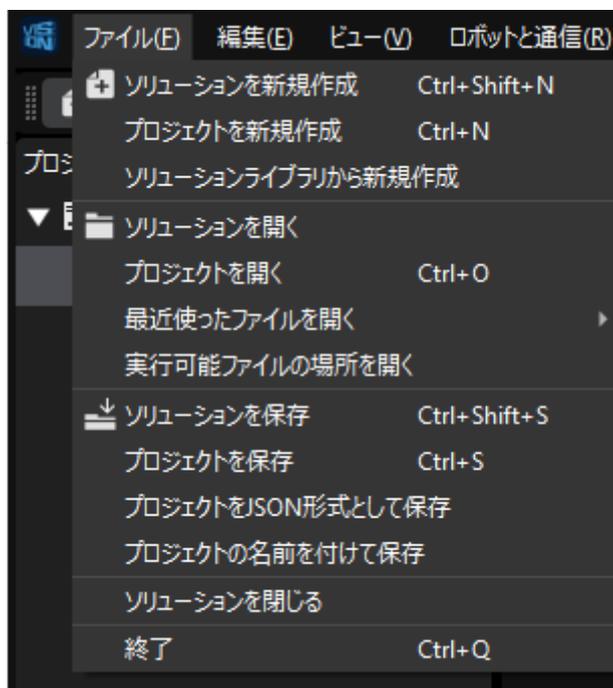


ソリューションが自動読み込みに設定されている場合、プロジェクト名が緑色で表示され、プロジェクト名の左側にプロジェクト番号が表示されます。



プロジェクト番号は、ロボットの把持・配置アプリケーションでMech-Visionプロジェクトの実行をトリガーするために使用されます。

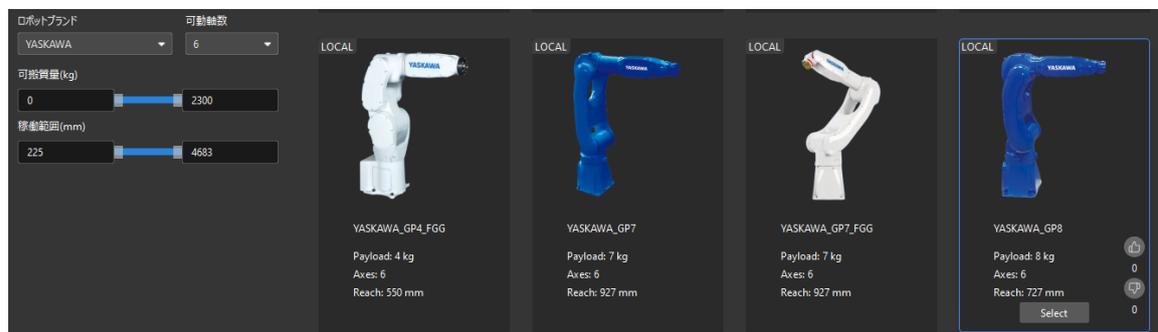
6. メニューバーから**ファイル** > **ソリューションを保存**を選択します。



「ロボット通信設定」を実行

1. Mech-Visionのツールバーで [**ロボット通信設定**] をクリックします。

2. **ロボットを選択** のドロップダウンボックスをクリックし、**適応可能なロボット** を選択して **[ロボット型番を選択]** をクリックします。表示される画面で「YASKAWA_GP8」を選択して **[次へ]** をクリックします。



3. **通信方式** の画面で、インターフェースタイプを **標準インターフェース** に、通信プロトコルを **TCP Server** に、プロトコル形式を **ASCII** に選択してから、**[適用]** をクリックします。

通信方式 [ロボット通信設定のマニュアル](#)

インターフェースタイプ

標準インターフェース

通信プロトコル

TCP Server

ASCII

詳細設定 ▾

IPアドレス

ホストIPアドレス

0

.

0

.

0

.

0

:

50000

ロボット読み込みの説明

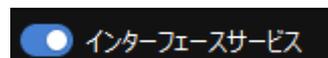
プログラムフォルダを開く

ソリューションを開くとインターフェースサービスを自動的に起動

戻る

適用

4. Mech-Visionのメイン画面で、ツールバーのインターフェースサービスが起動されていることを確認します。



読み込みファイルの準備

1. IPCで、Mech-Mindビジョンシステムのインストールディレクトリにある**Mech-Center/Robot_Interface/YASKAWA**フォルダを開きます。
2. JBIフォルダ、**sample**フォルダおよび**mm_module_ycrc1000.out**ファイルをコピーしてUSBメモリのルートディレクトリに貼り付けます。
 - **mm_module_ycrc1000.out**ファイル：バックグラウンドプログラムファイルとなります。

- **JBI**フォルダ：フォアグラウンドプログラムファイルを格納しています。
- **sample**フォルダ：簡単な把持・配置のサンプルプログラムを格納しています。



DX200型番のコントローラを使用している場合は、**mm_module_dx200.out**というバックグラウンドプログラムファイルをコピーする必要があります。

3. USBメモリをティーチペンダントの背面にあるUSBポートに挿入します。

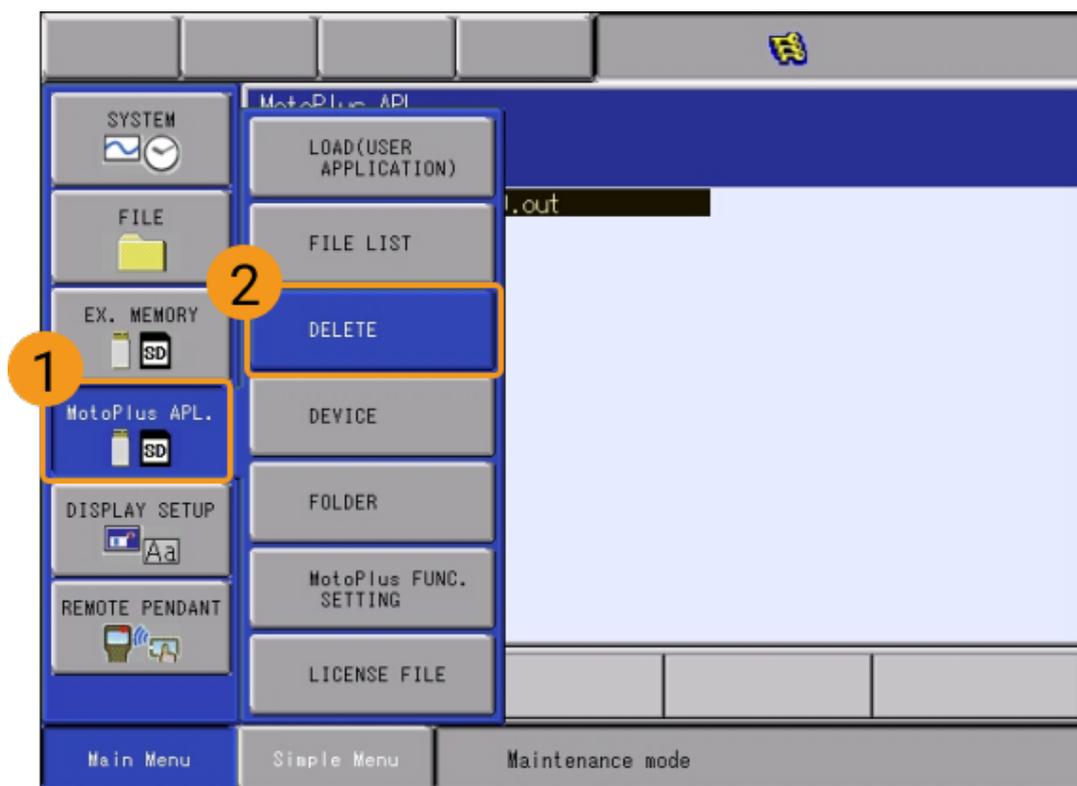
MotoPlusで他のプログラムが実行されていないことを確認

ロボットプログラムをロボットに読み込む前に、他のMotoPlusプログラムが実行されていないことを確認してください。

- 実行中のプログラムがない場合は、この手順をスキップしてください。
- 実行中のプログラムがある場合は、それを削除してください。

▼ (クリックして展開) 詳細方法

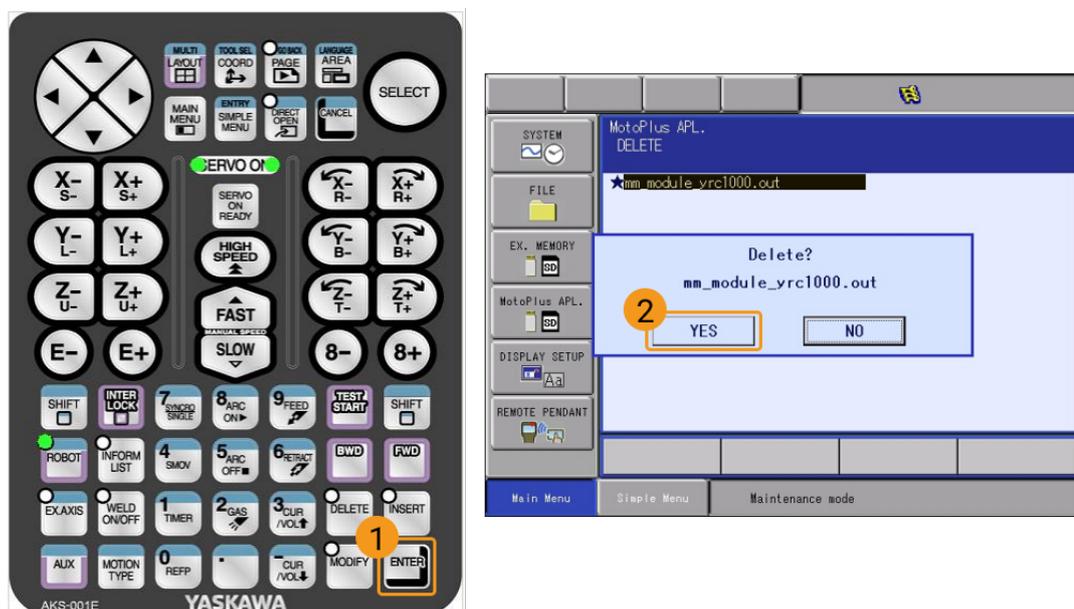
1. メンテナンスモードの**管理モード**では、**Main Menu**から**MotoPlus APL. > DELETE**を選択します。



2. 削除画面では、削除するプログラムファイルを選択し、ティーチペンダントの**SELECT**を押してそのファイルを選択します。



3. ティーチペンダントの**ENTER**キーを押し、ポップアップダイアログボックスの[**YES**]ボタンをクリックしてプログラムを削除します。

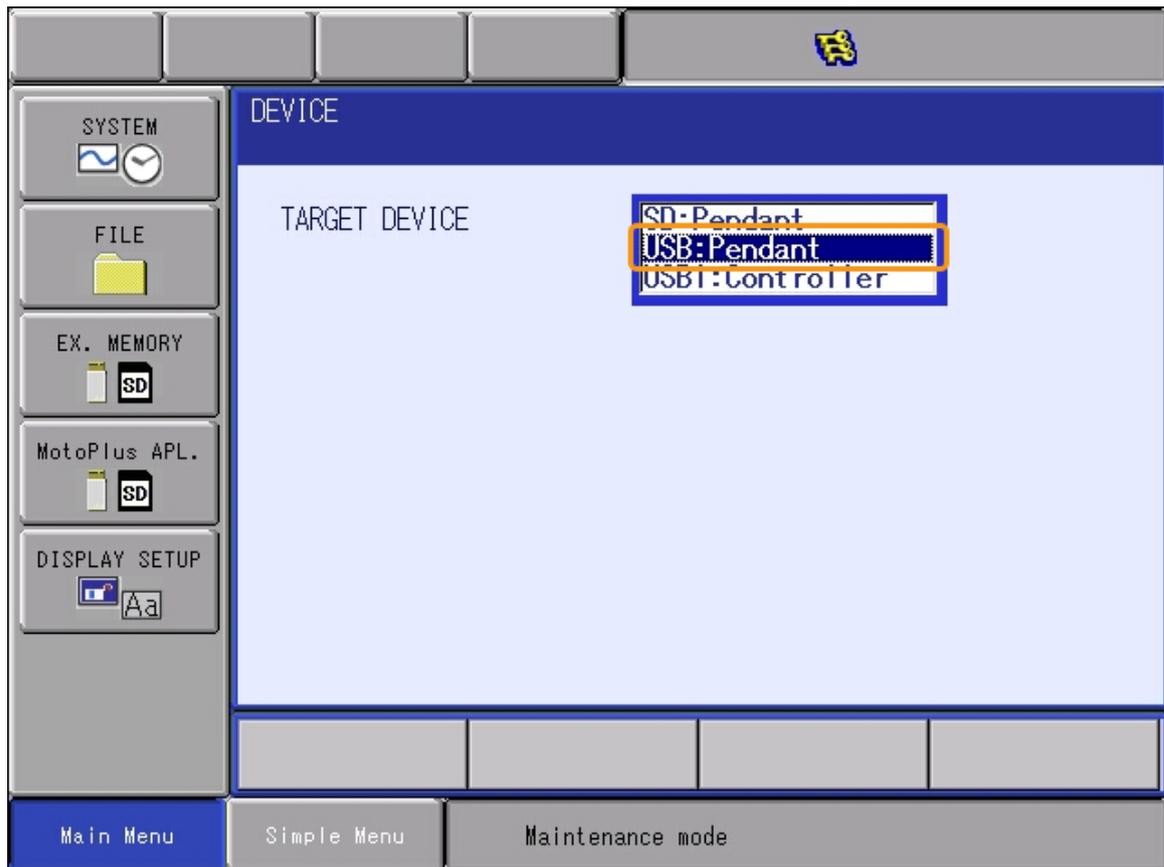


ファイルの読み込み

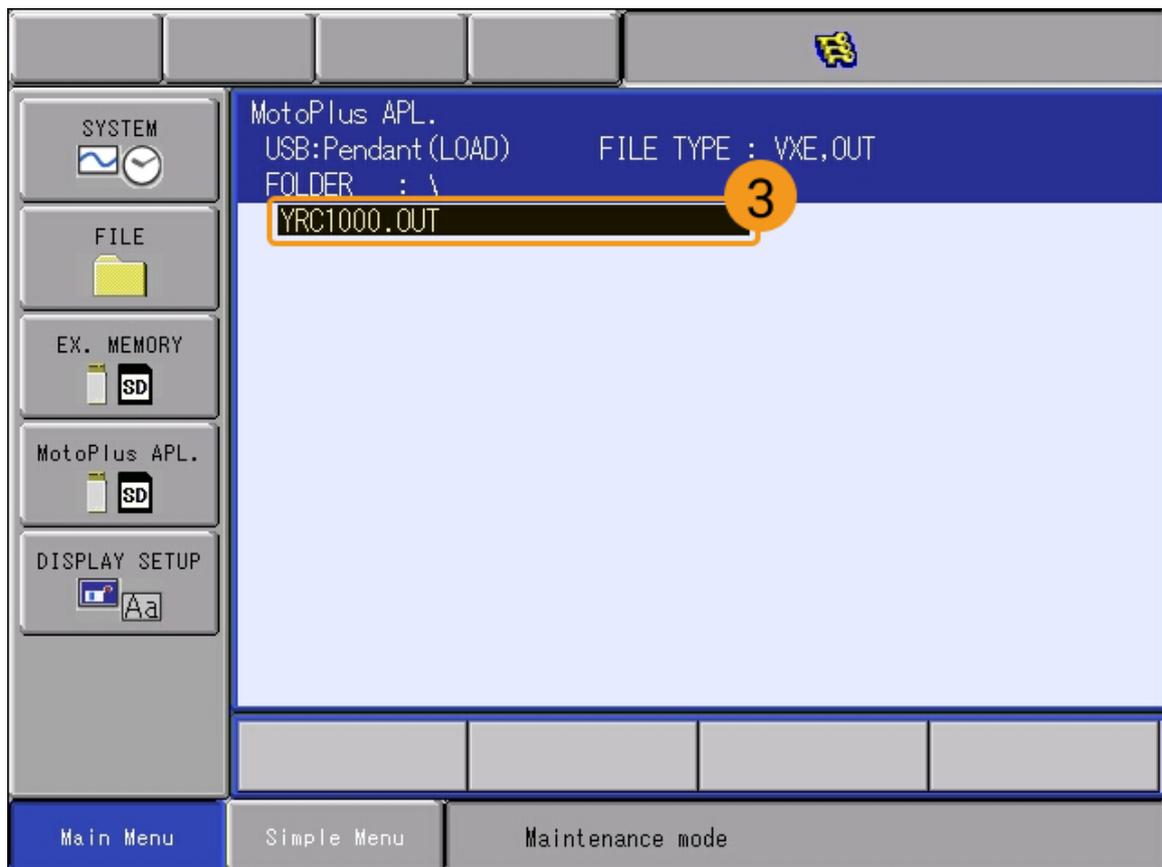
バックグラウンドプログラムファイルの読み込み

1. メンテナンスモードでは、Main Menuから **MotoPlus APL. > DEVICE > USB : Pendent** を選択します。





2. **MotoPlus APL. > LOAD (USER APPLICATION)** を選択し、**mm_module_ycr1000.out** を選択します。次に、ティーチペンダントの **ENTER** キーを押して **[YES]** を選択し、インストールを実行します。





DX200型番のコントローラを使用している場合は、ここでは`mm_module_dx200.out`を選択する必要があります。

- インストール後、**MotoPlus APL.** ▶ **FILE LIST**を選択します。インストールされたファイルにバックグラウンドプログラム（YRC1000.OUT）が表示されていれば、バックグラウンドプログラムのインストールは成功です。



DX200型番のコントローラを使用している場合は、表示されたプログラムはDX200.OUTとなります。

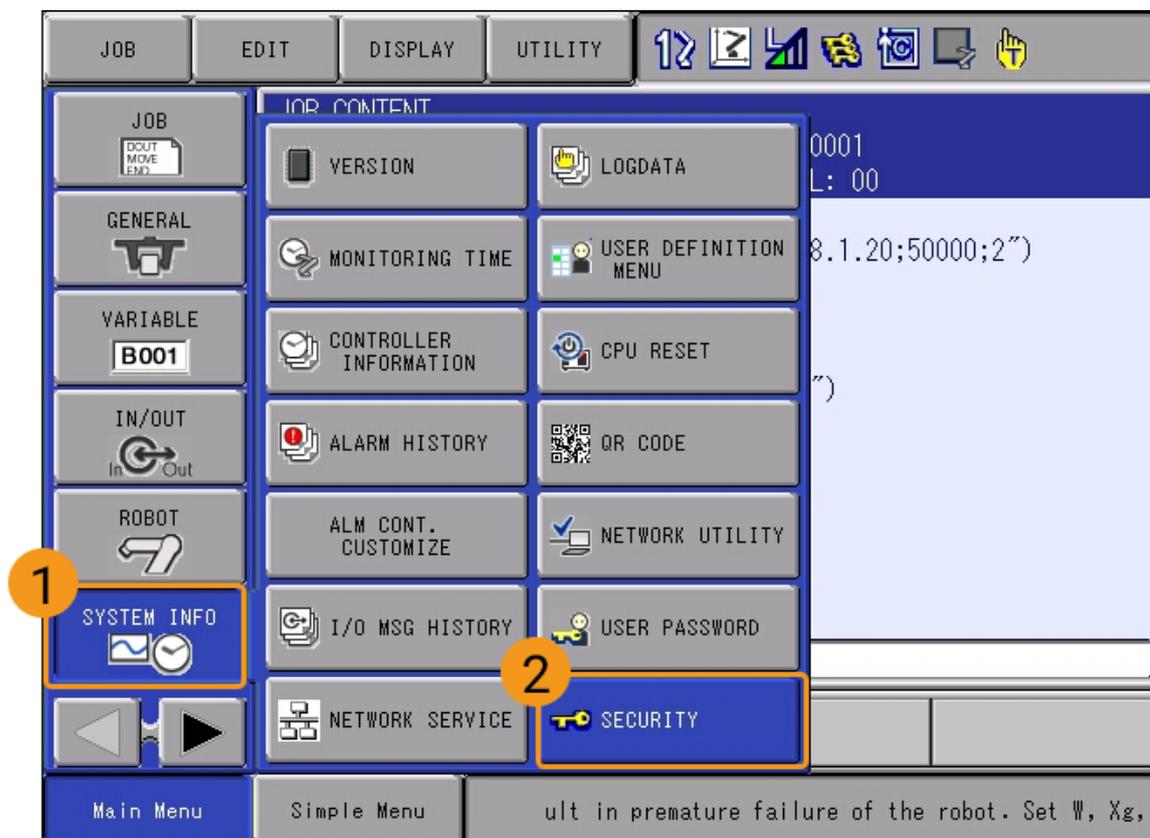
4. バックグラウンドプログラムファイルが読み込まれた後、ロボットを再起動するとオンラインモードになります。

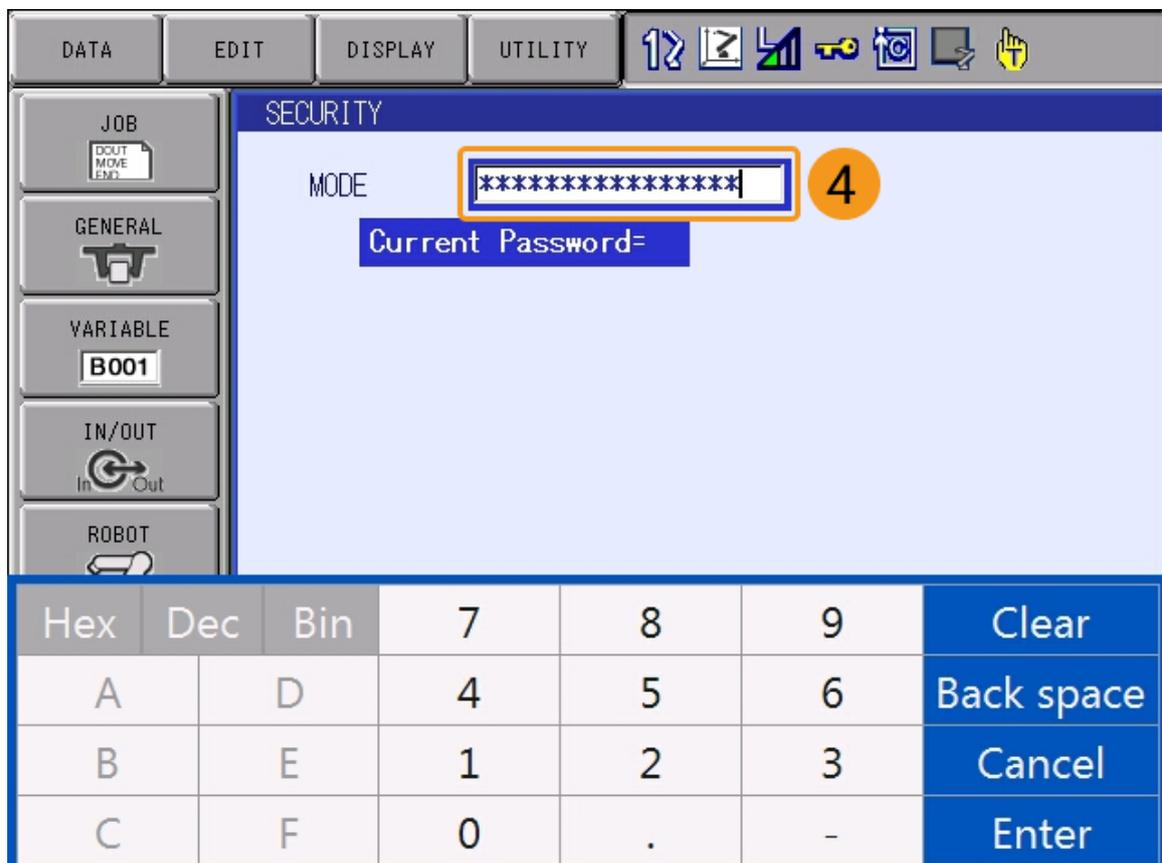
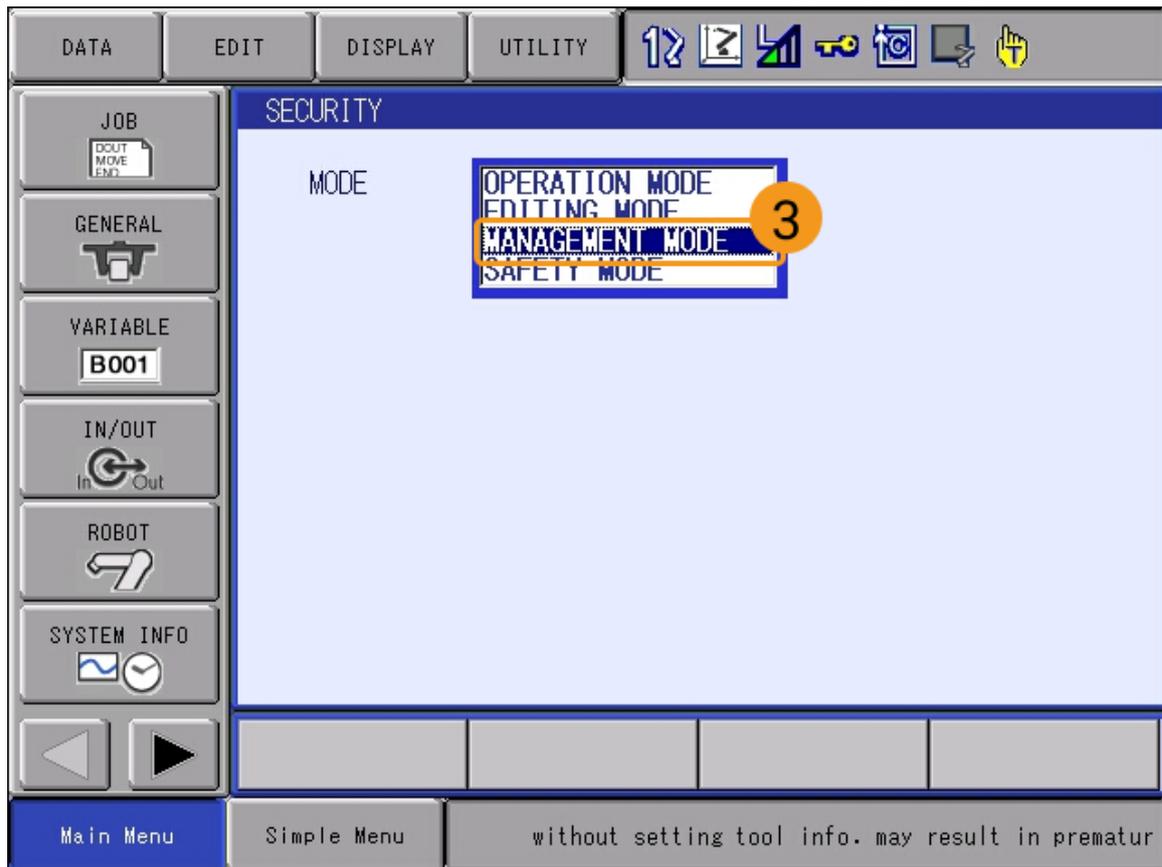


フォアグラウンドプログラムファイルとサンプルプログラムファイルを読み込むには、オンラインモードで操作する必要があるため、この操作はバックグラウンドプログラムの読み込みが完了してから行ってください。

フォアグラウンドプログラムファイルの読み込み

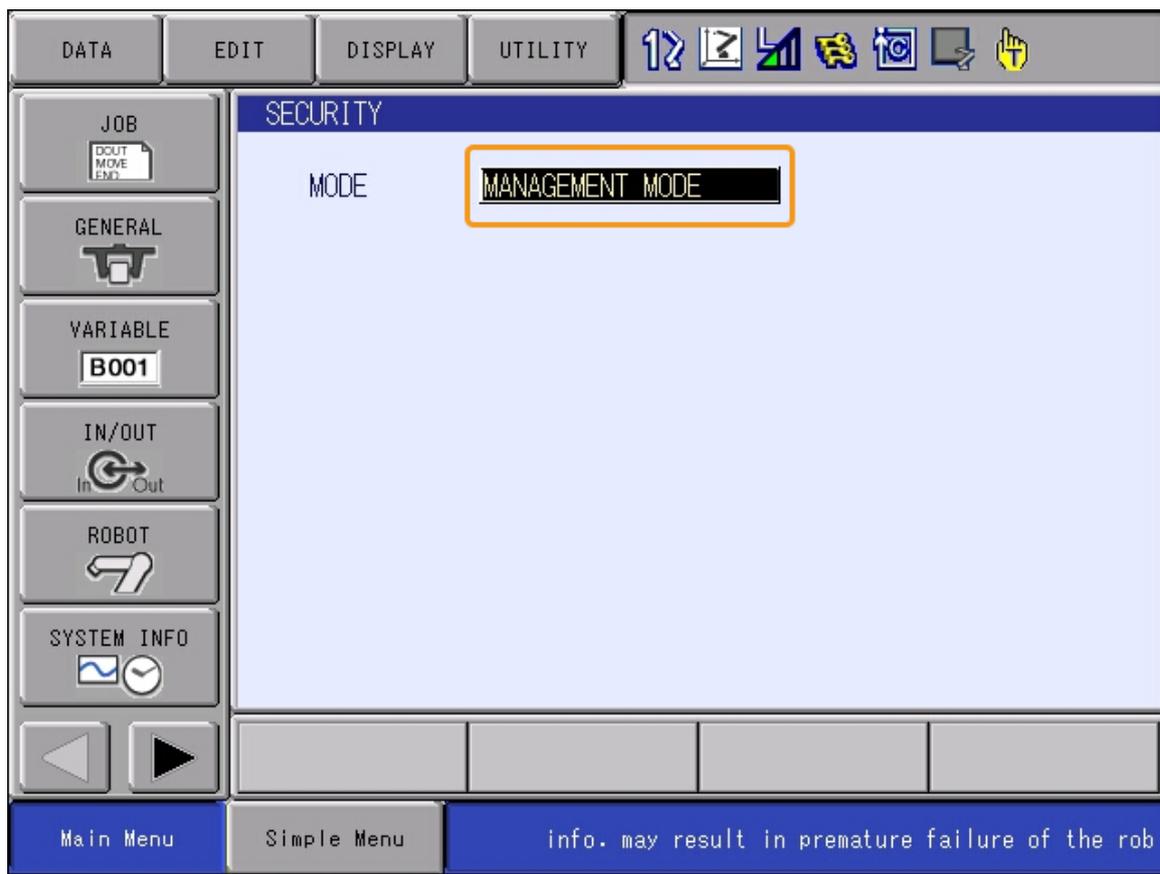
1. オンラインモードでは、Main Menuから **SYSTEM INFO** > **SECURITY**をクリックし、**MANAGEMENT MODE**を選択します。パスワード（デフォルトのパスワードは16個の9）を入力します。



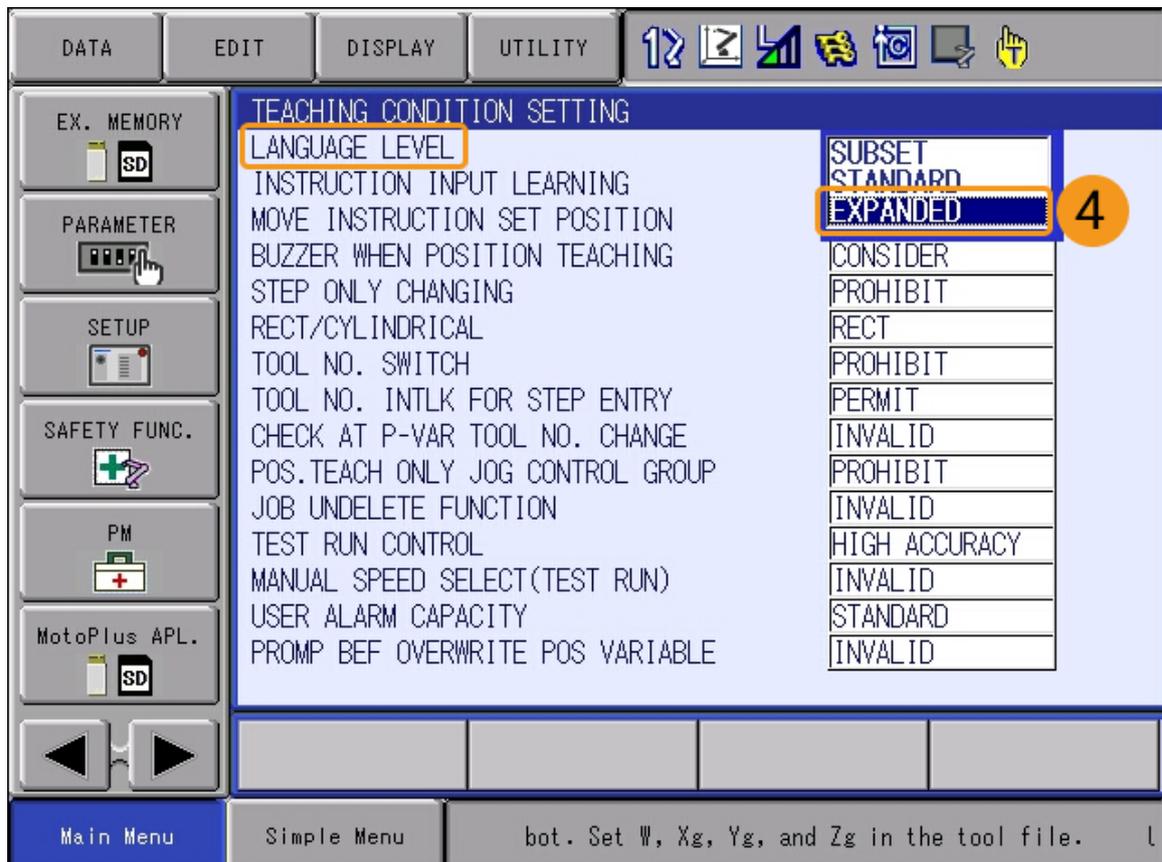
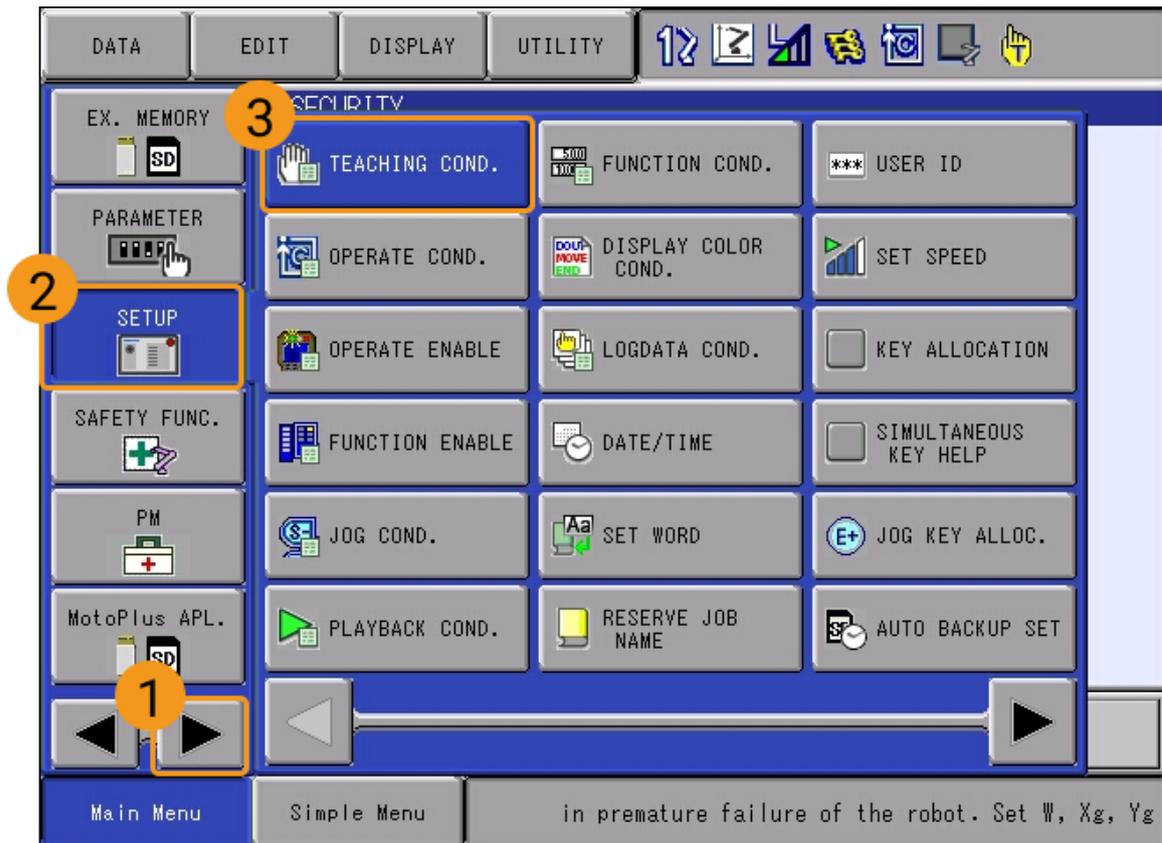


パラメータを入力した後、ティーチペンダントの右下隅にある **ENTER** キーを押して管理モー

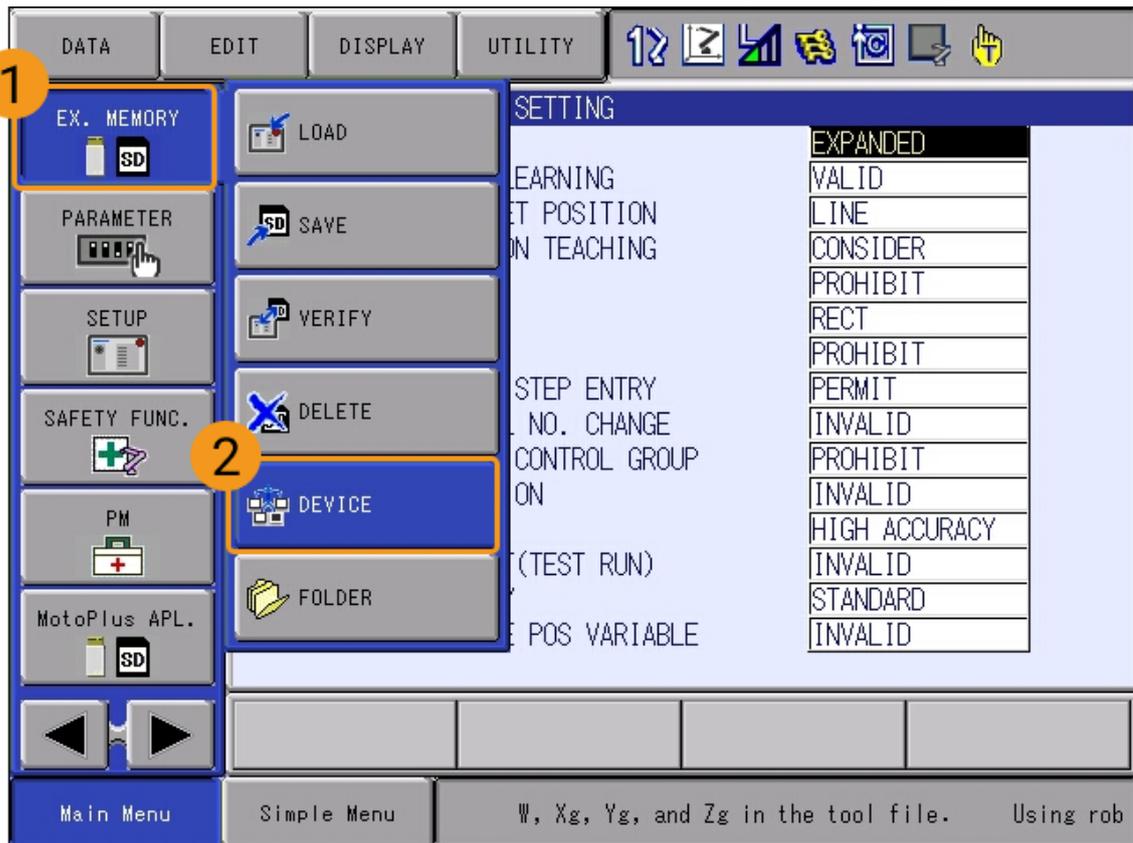
ドに切り替えます。



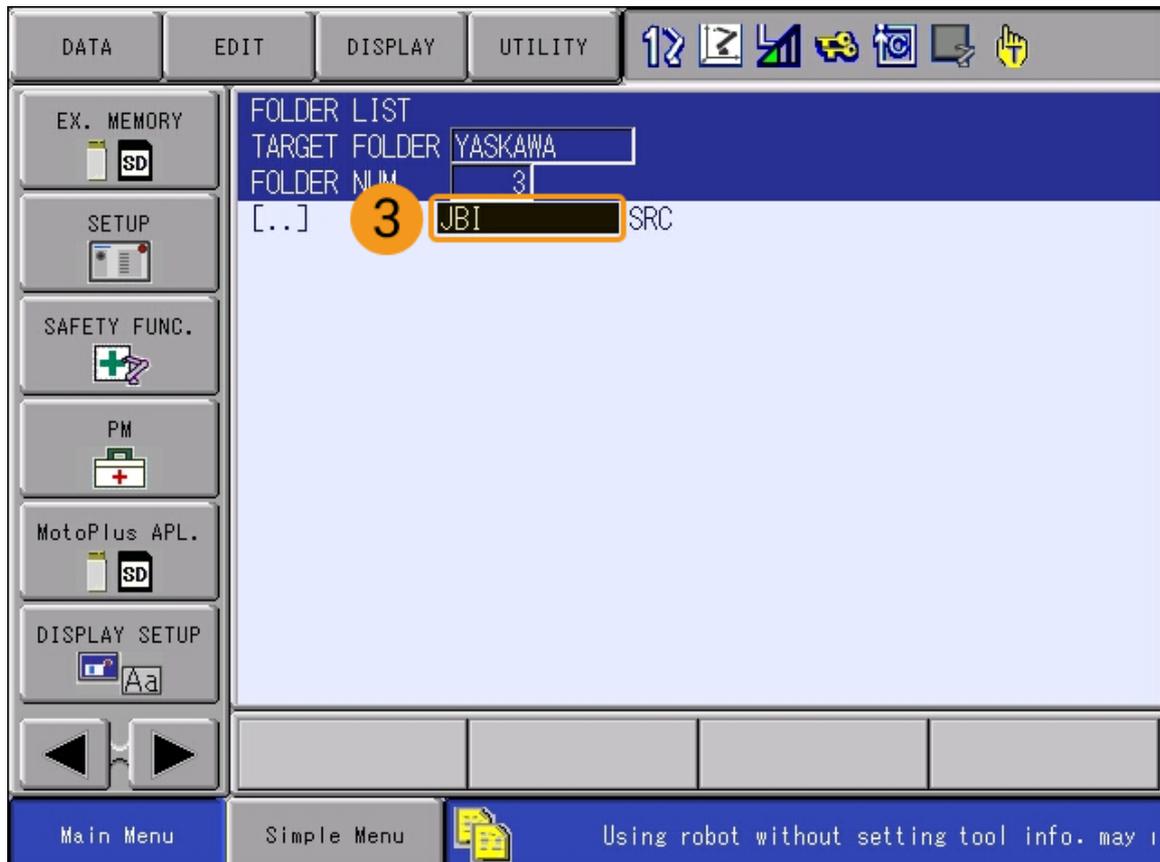
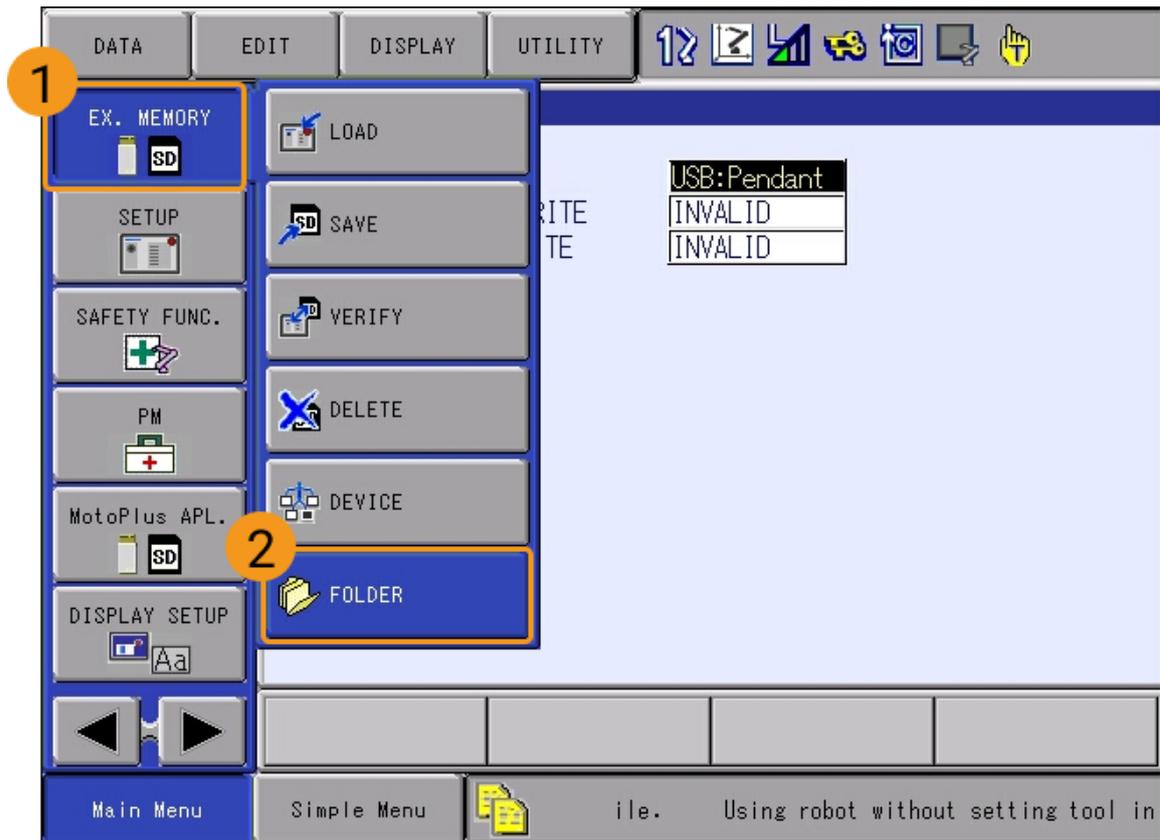
2. ティーチペンダントパネルの左下隅にあるページ切替  ボタンをクリックし、**SETUP** > **TEACHING COND**を選択します。LANGUAGE LEVELのドロップダウンメニューで**EXPANDED**を選択します。

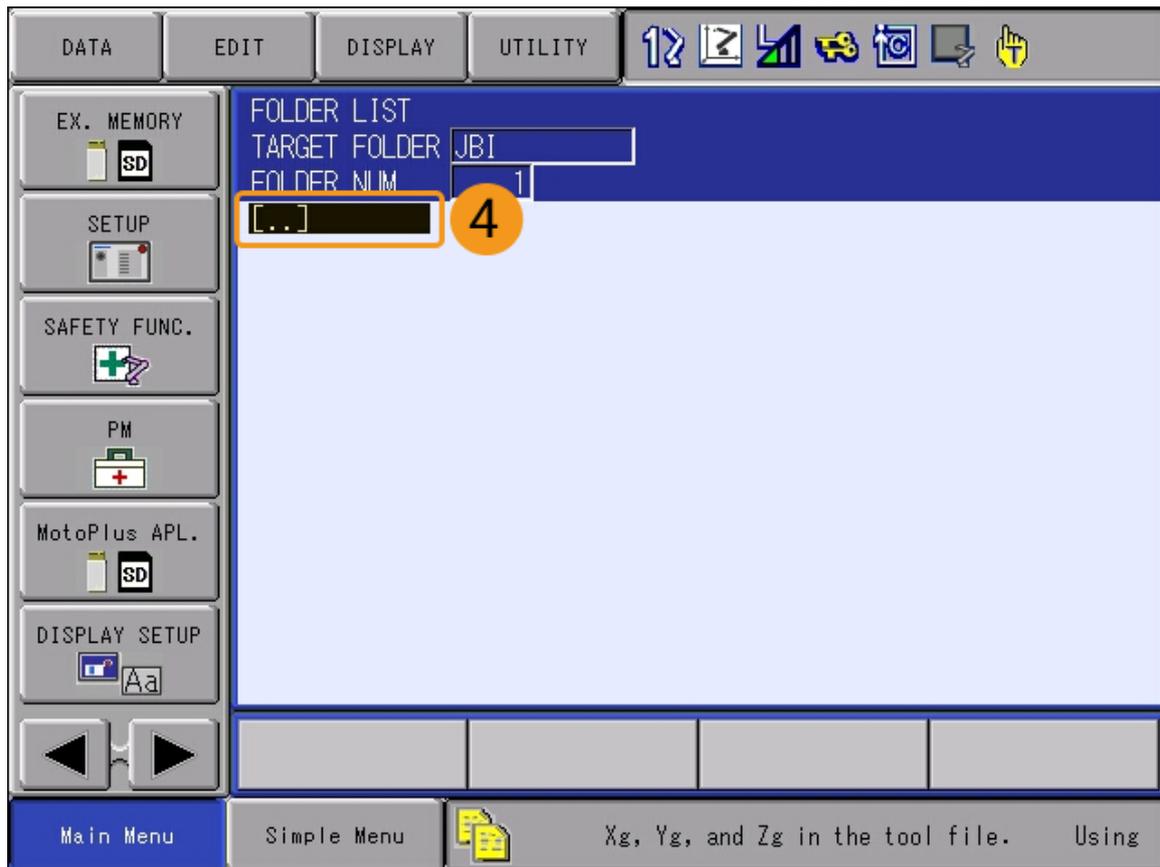


3. EX. MEMORY ▶ DEVICE をクリックし、USB : Pendant を選択します。

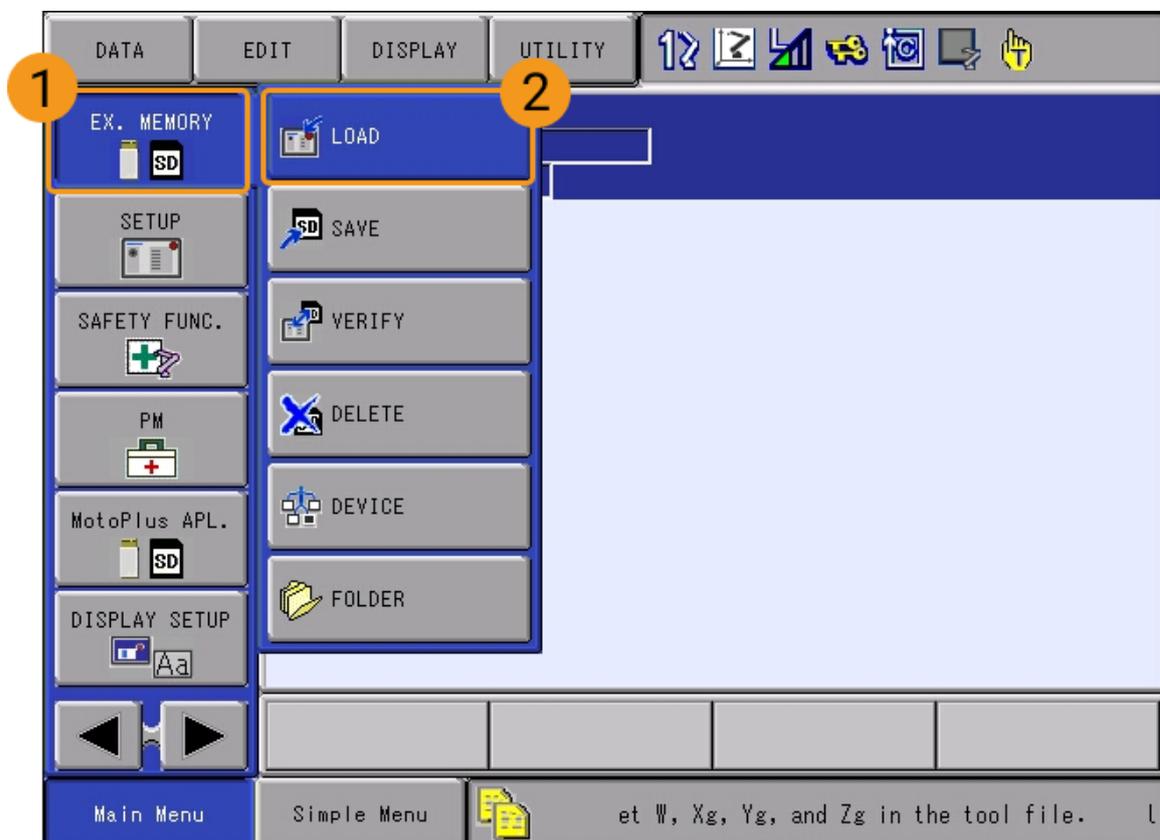


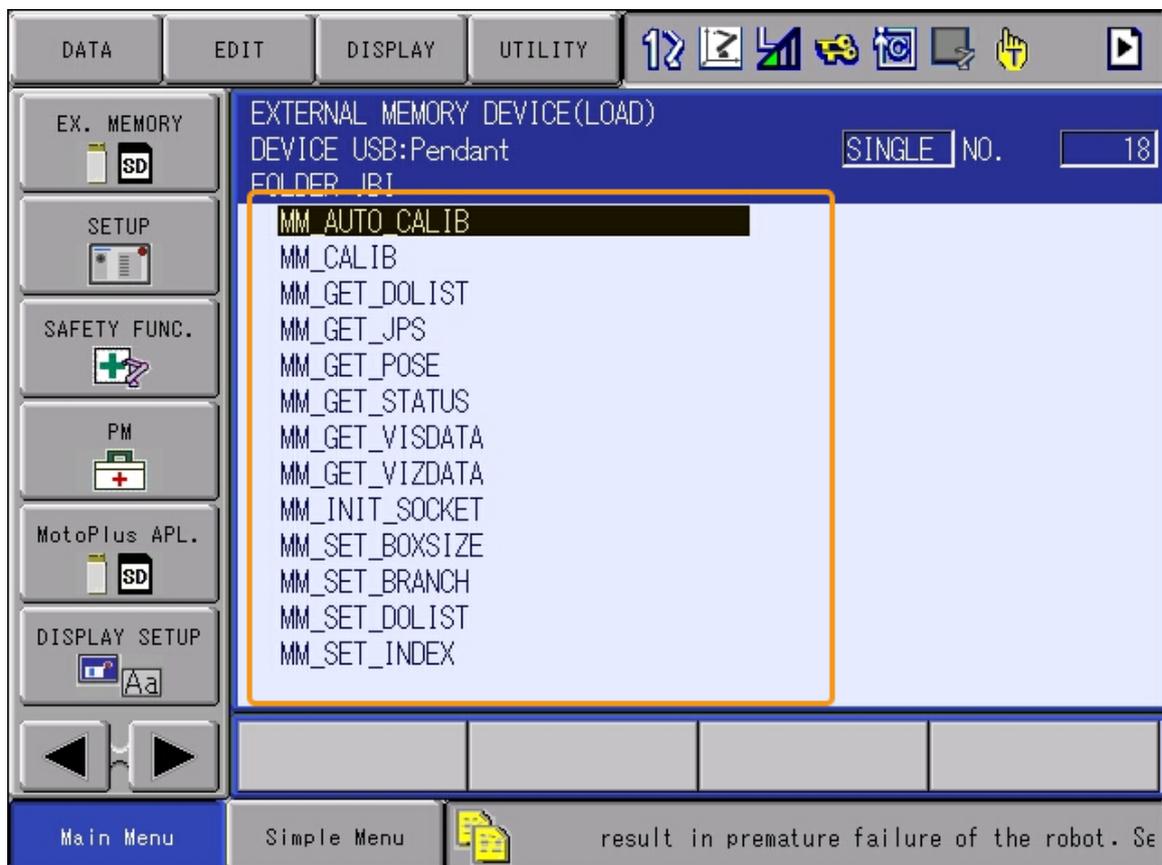
4. **EX. MEMORY** > **FOLDER** を選択します。ファイルリストから **JB1** を選択します。



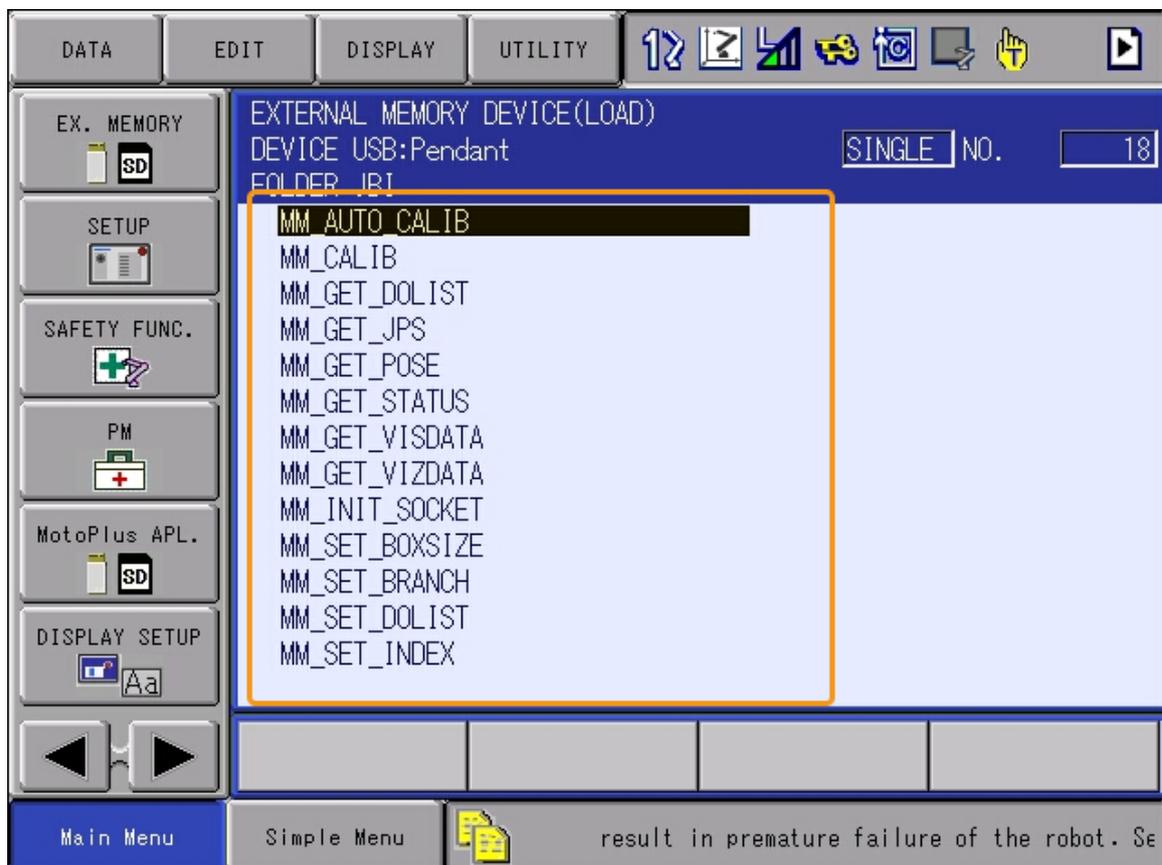
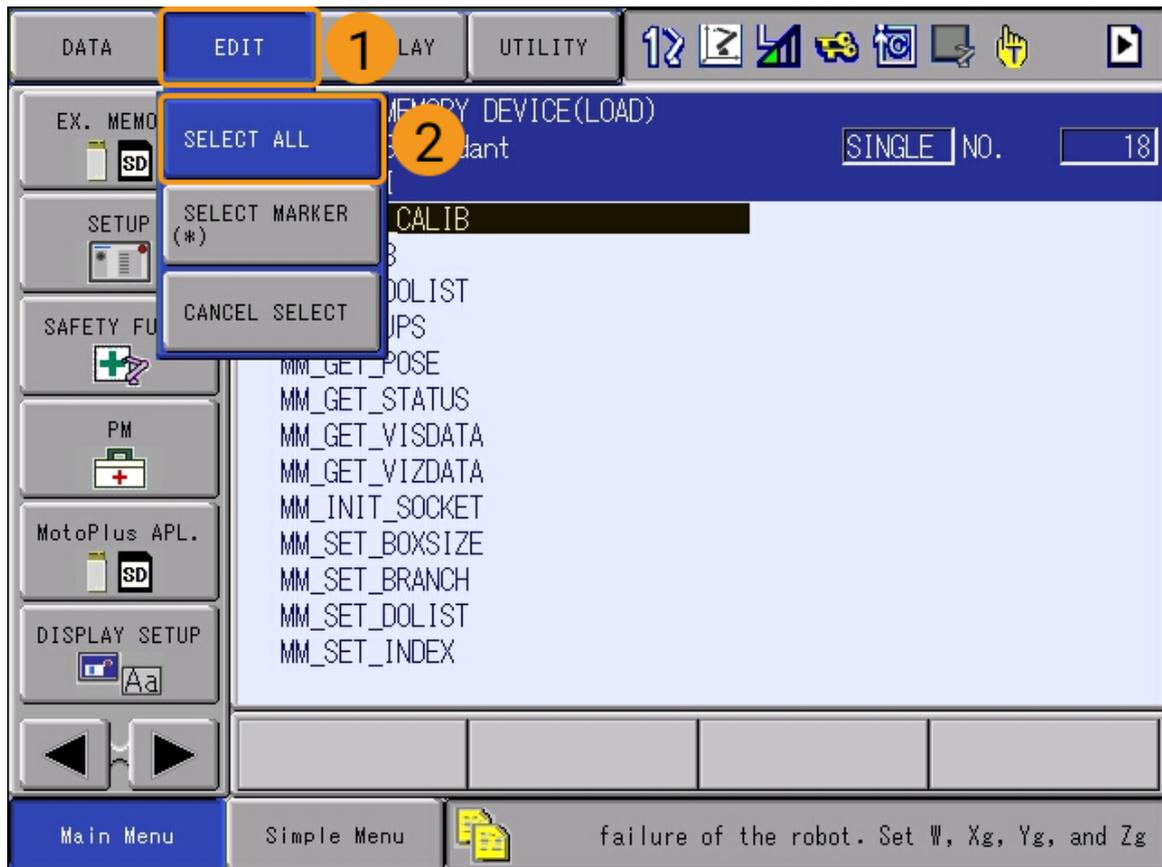


5. **EX. MEMORY** ▶ **LOAD** を選択し、**JOB** を選択します。インストールするプログラム一覧が画面に表示されます。



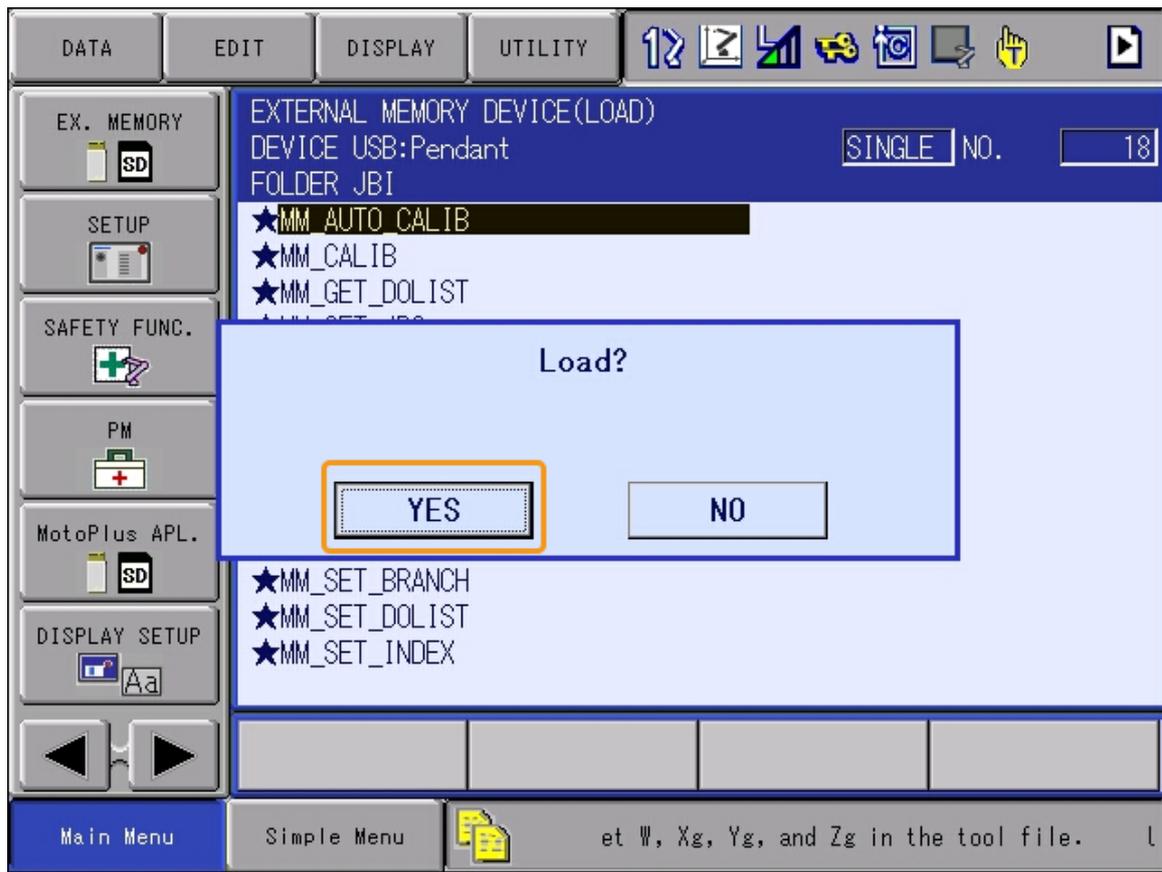


6. **EDIT** > **SELECT ALL** を選択し、すべてのプログラムを選択します。

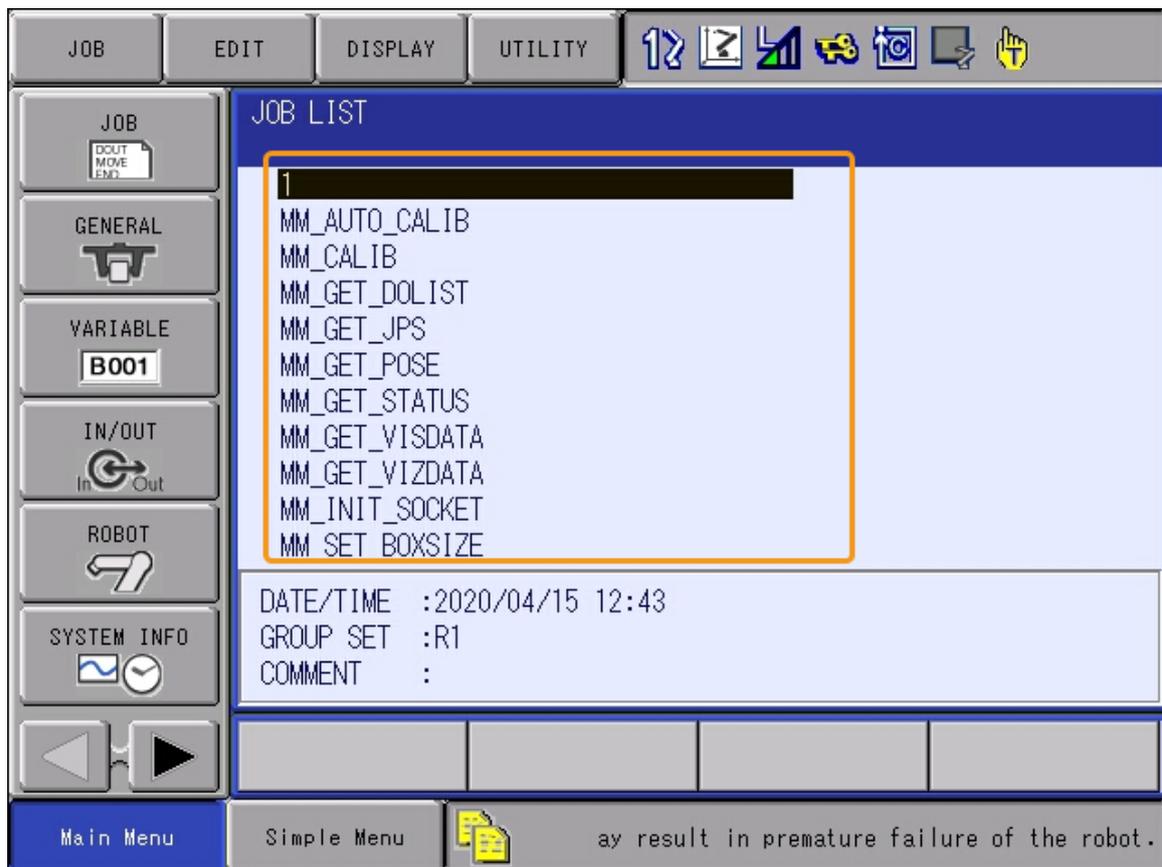
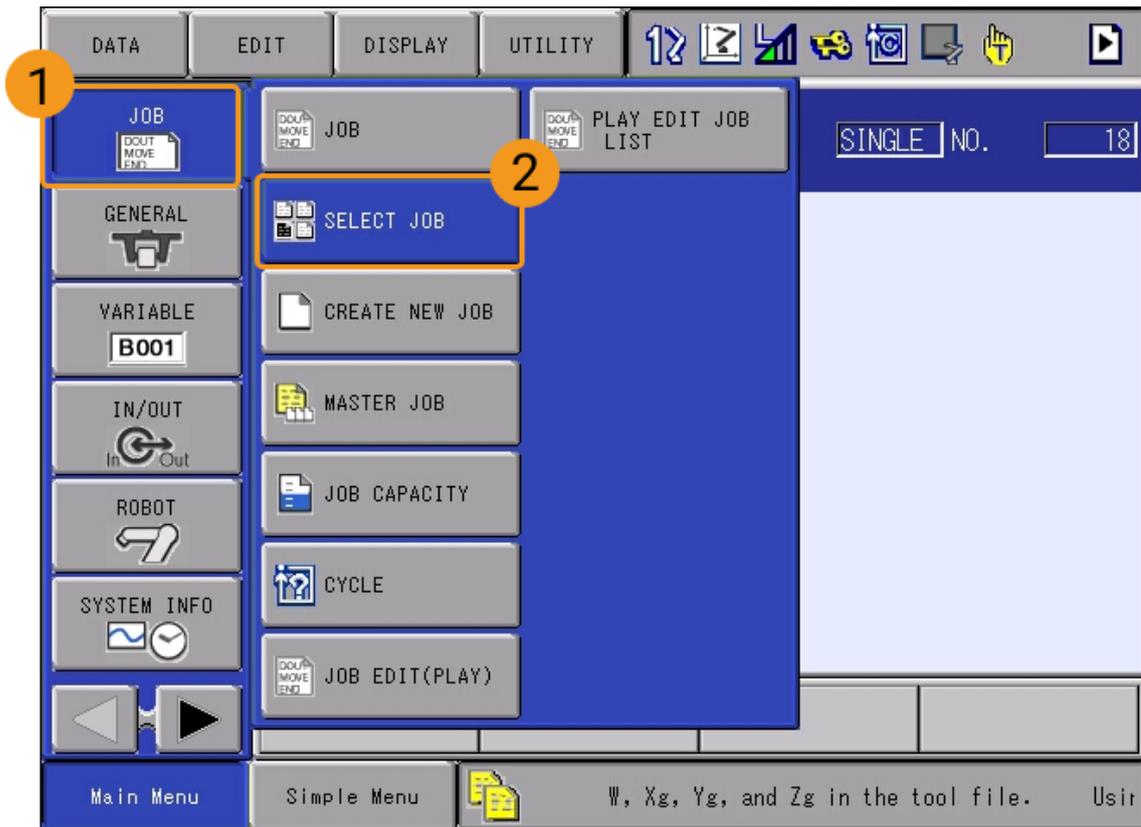


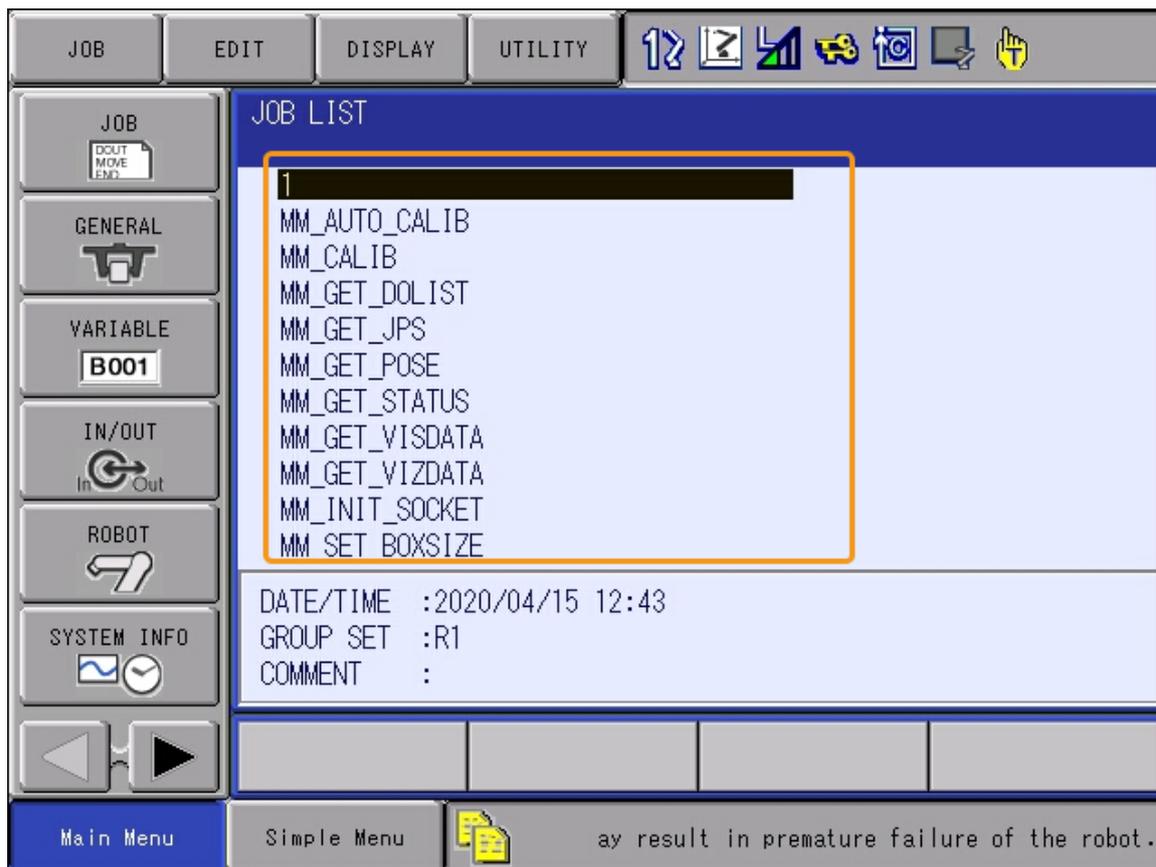
7. ティーチペンダントのENTERキーを押し、ポップアップウィンドウでYESを選択してフォア

グラウンドプログラムのインストールを開始します。

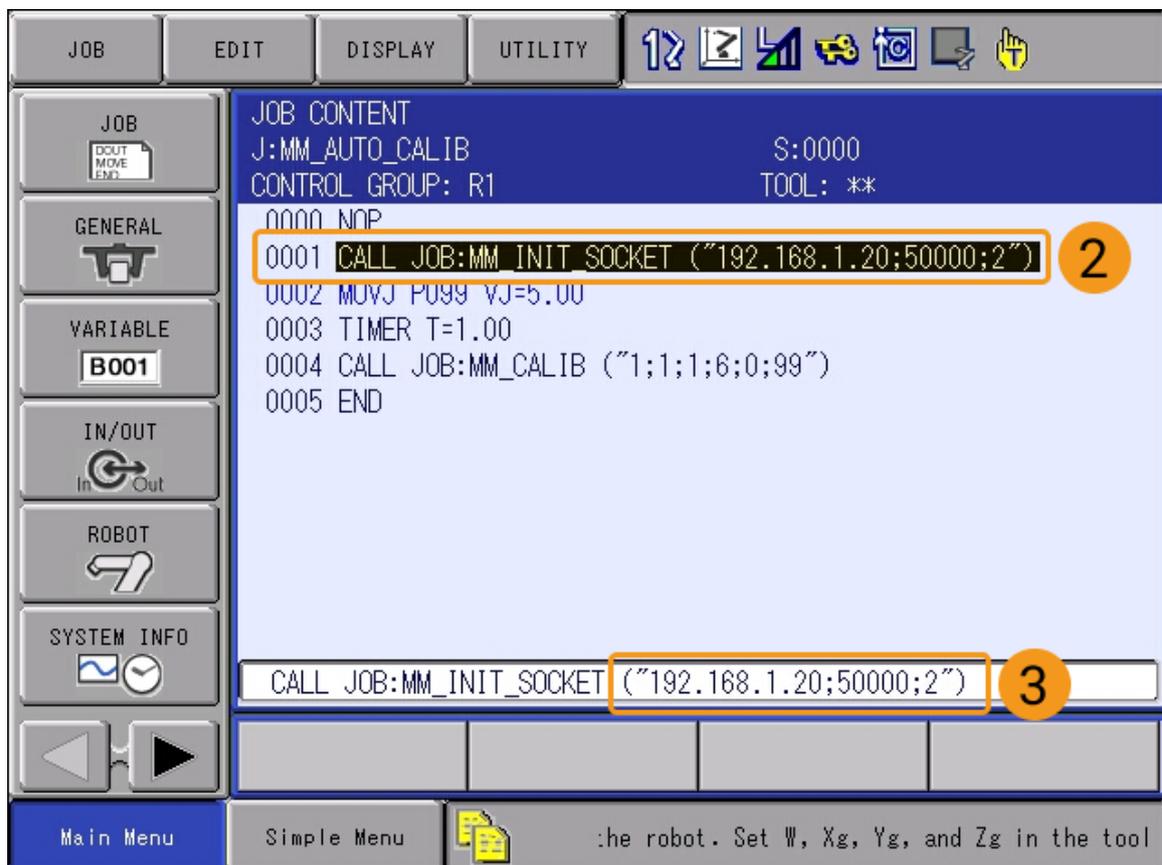
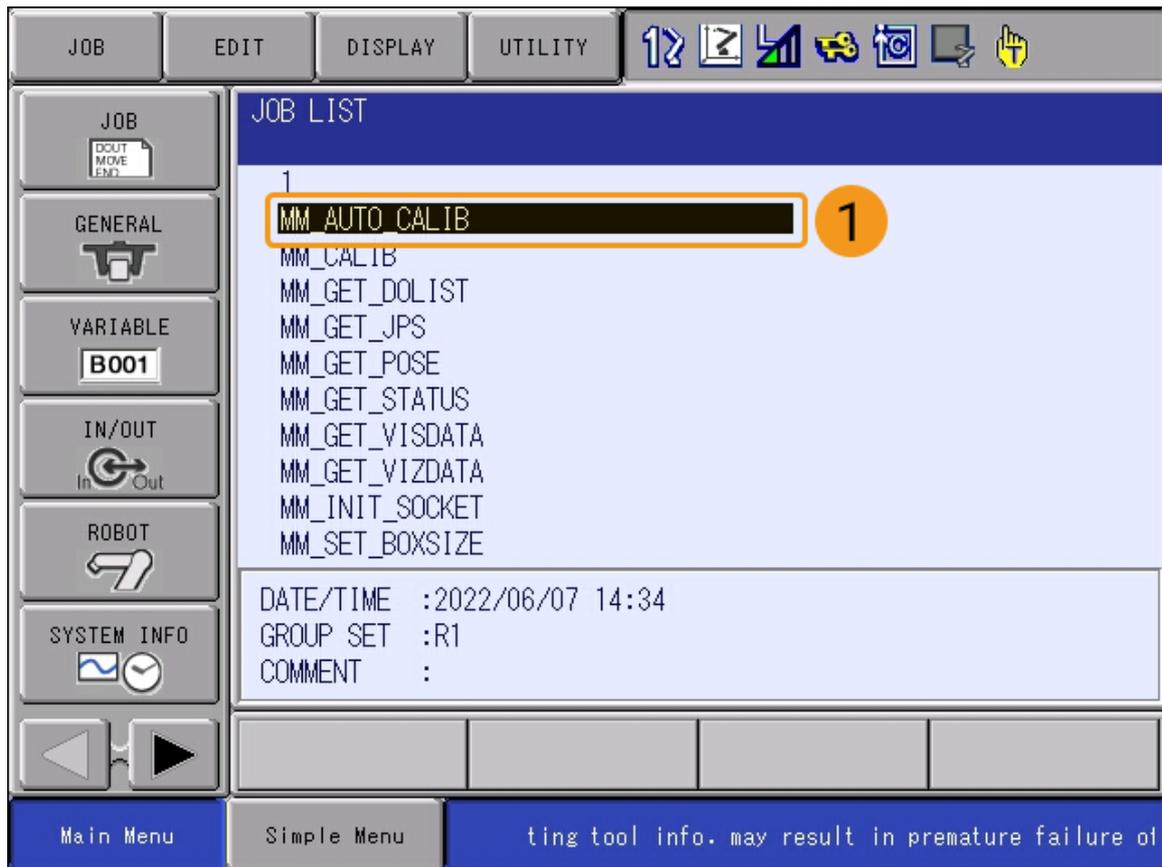


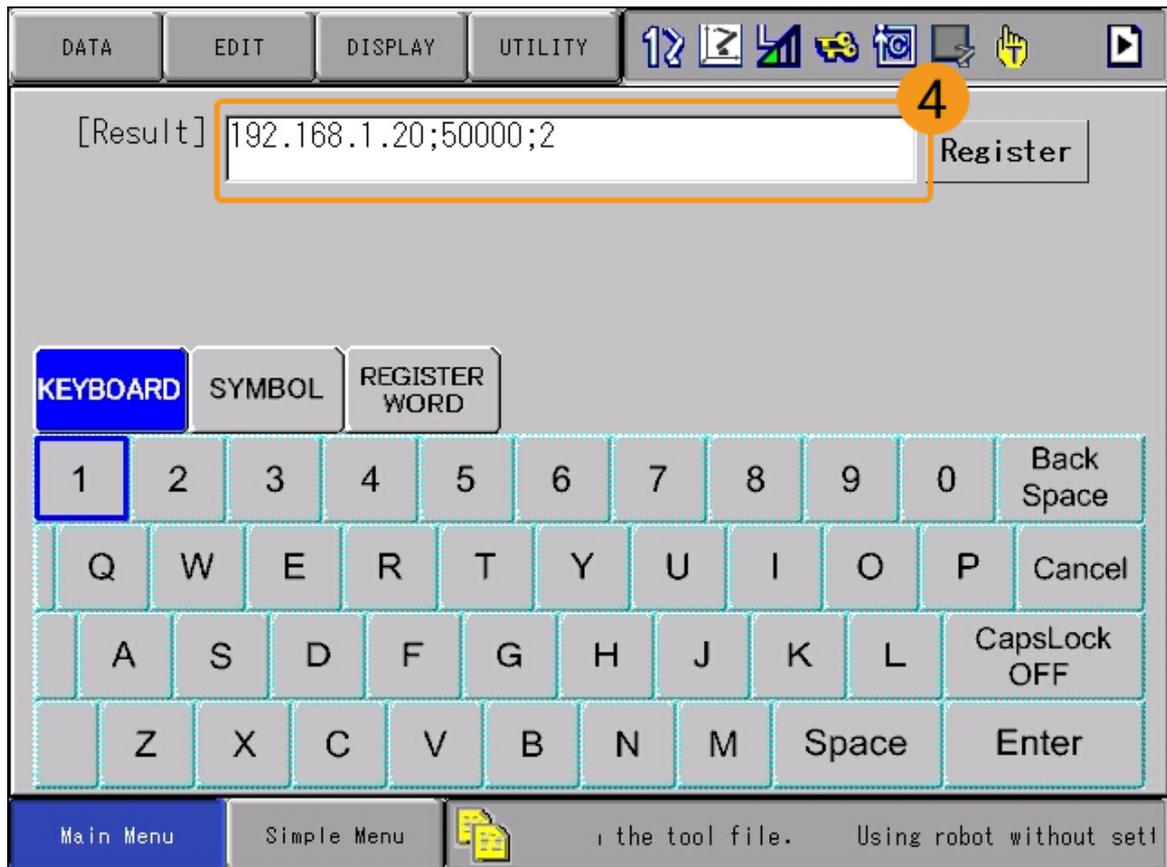
8. フォアグラウンドプログラムがインストールされた後、**JOB > SELECT JOB**を選択します。インストールされたプログラムリストにすべてのフォアグラウンドプログラムが表示されていれば、フォアグラウンドプログラムは正常にインストールされています。



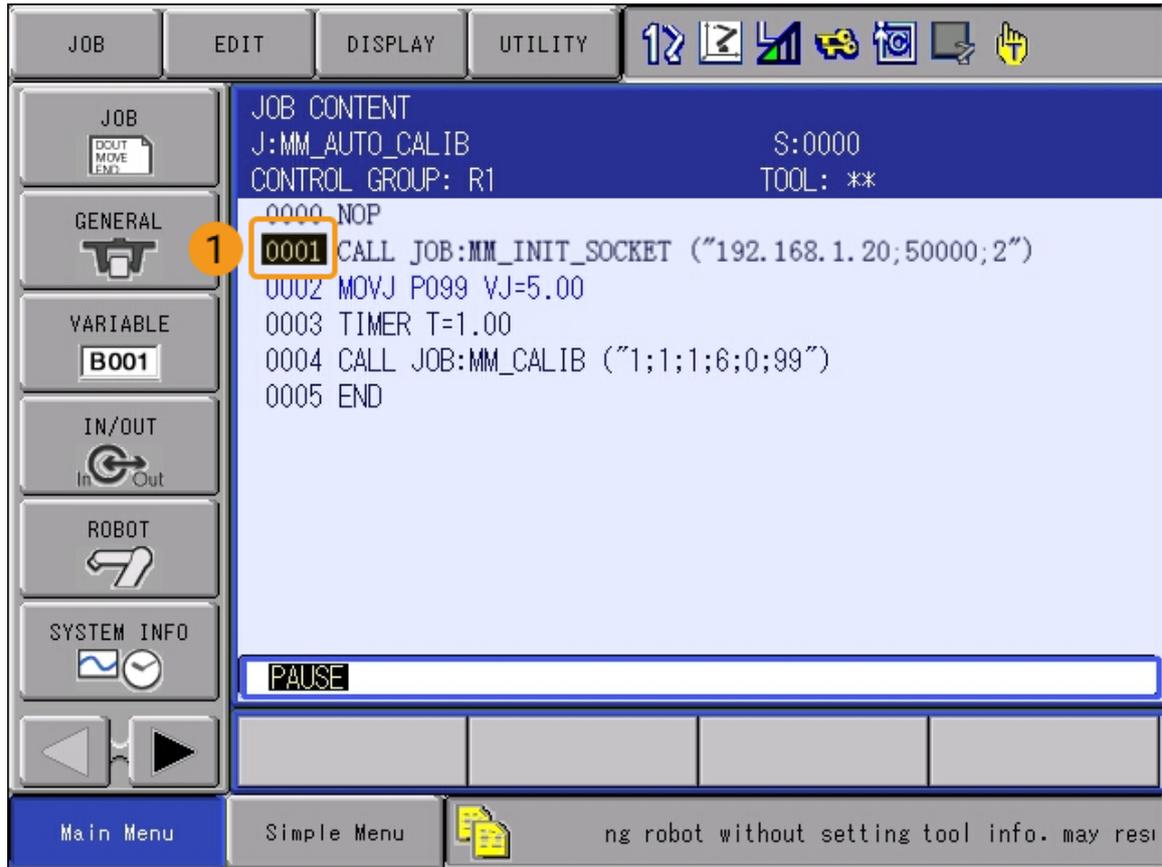


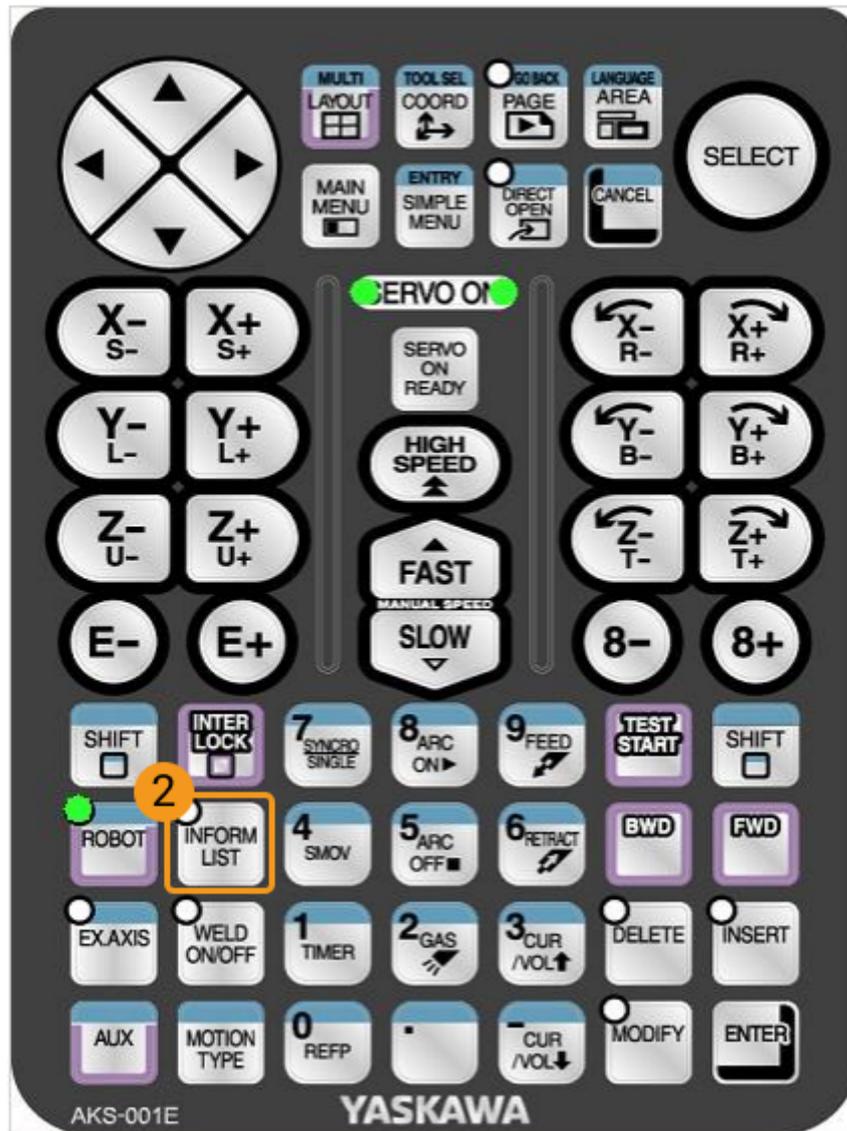
2. MM_AUTO_CALIBプログラムを選択し、**SELECT** をキーを押して開きます。0001行目を選択して右下隅にあるボックスでIPアドレスおよびポート番号を選択します。次に、**ENTER** を押して変更画面に表示された後、IPアドレスをIPCのIPアドレスおよびポート番号に変更します。

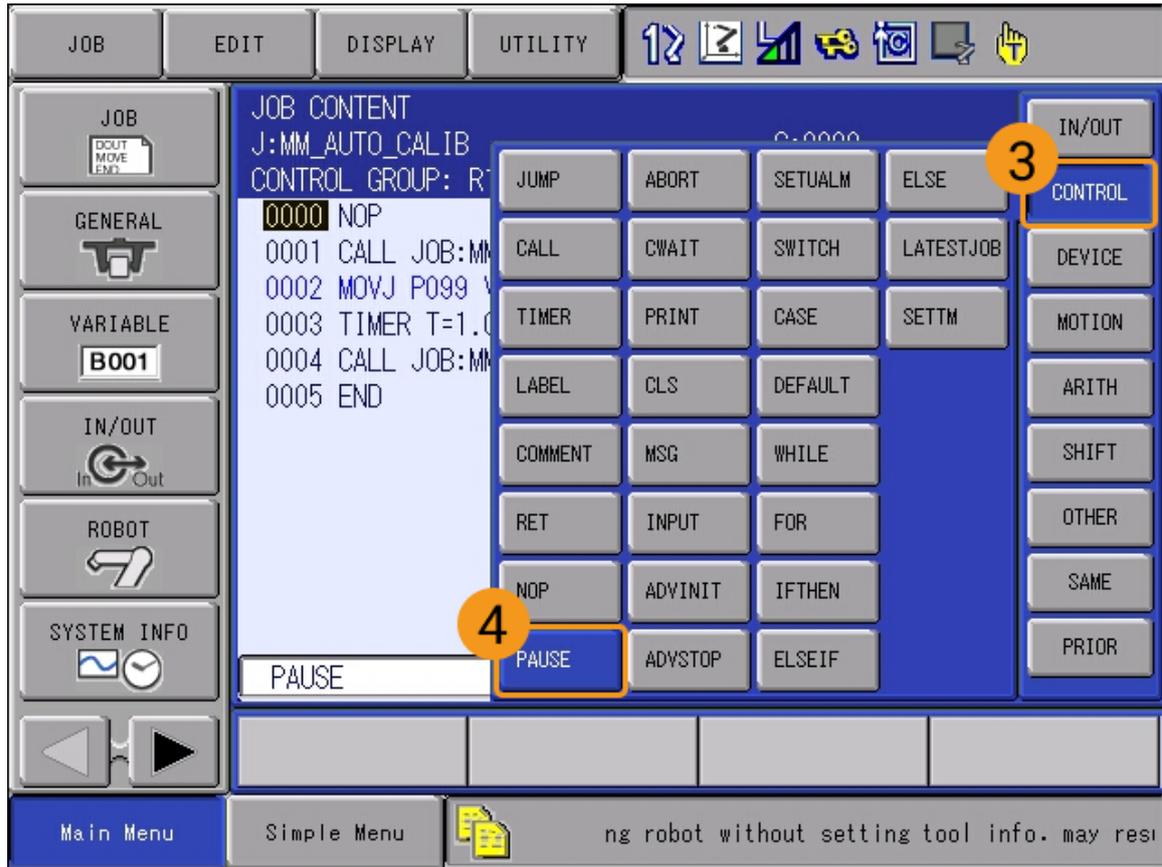


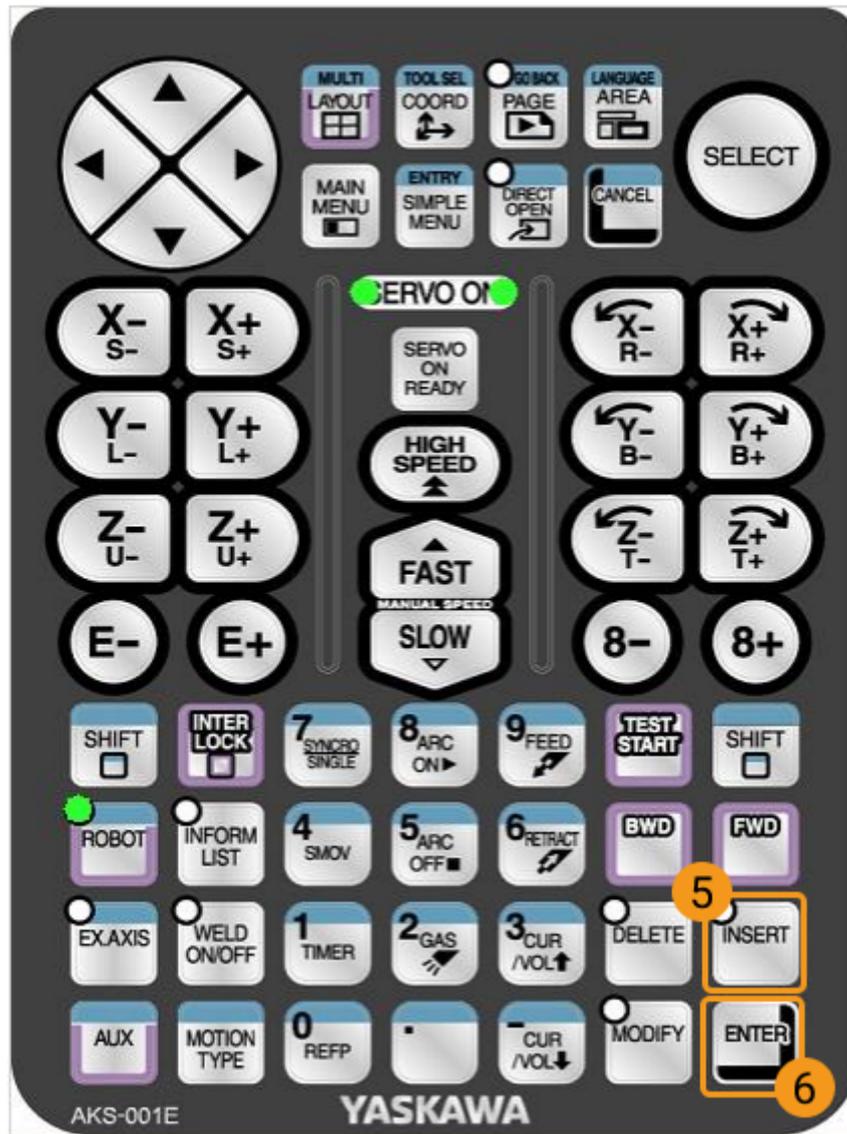


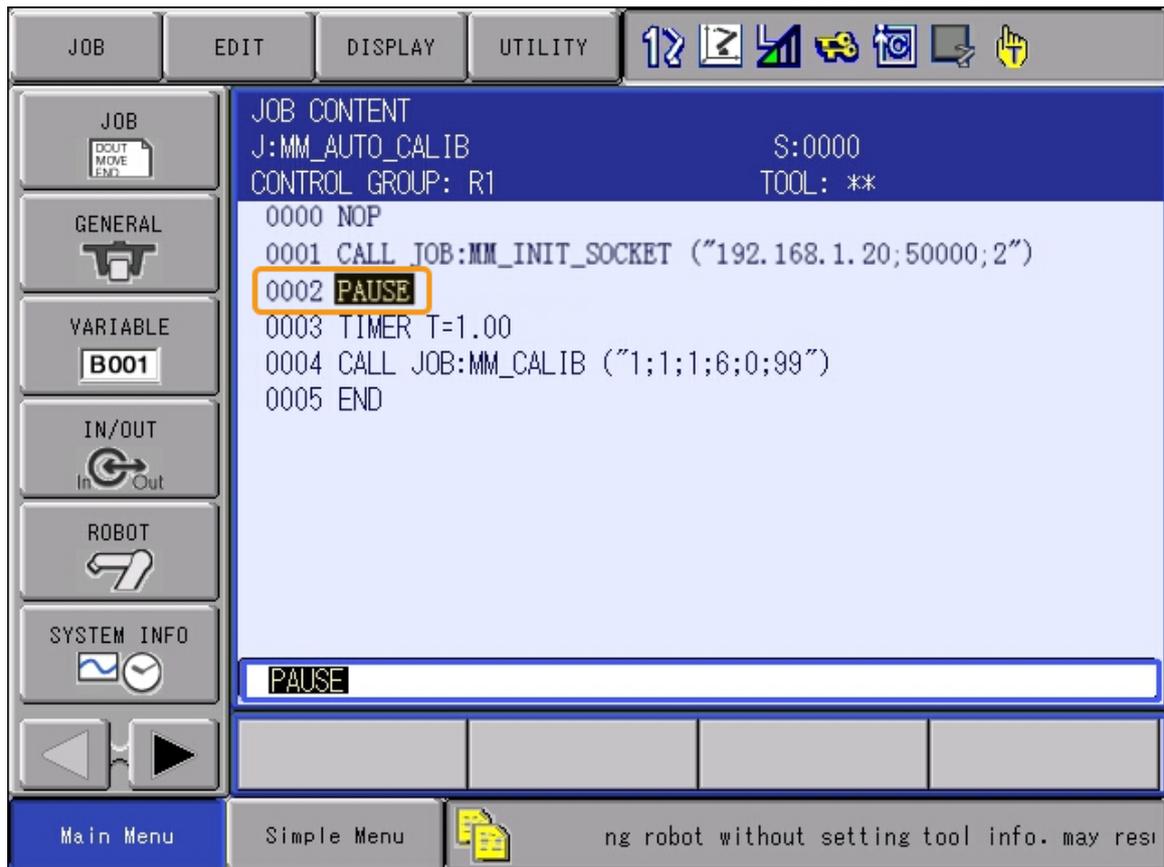
3. カーソルが0001行目に合わせ、ティーチペンダントの**INFORM LIST**キーを押し、右側のメニューから**CONTROL > PAUSE**を選択し、**INSERT**を選択して**ENTER**キーを押します。すると、0001行目の後に**PAUSE**コマンドが挿入されます。



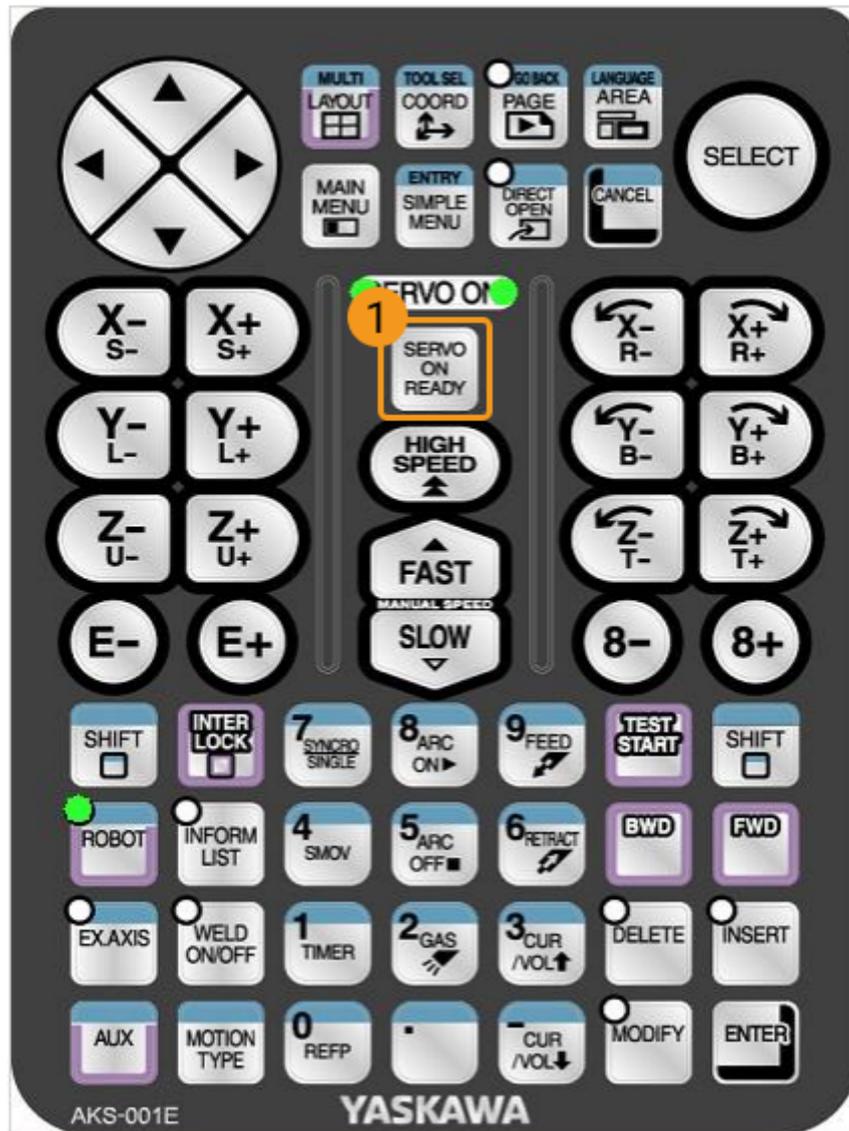


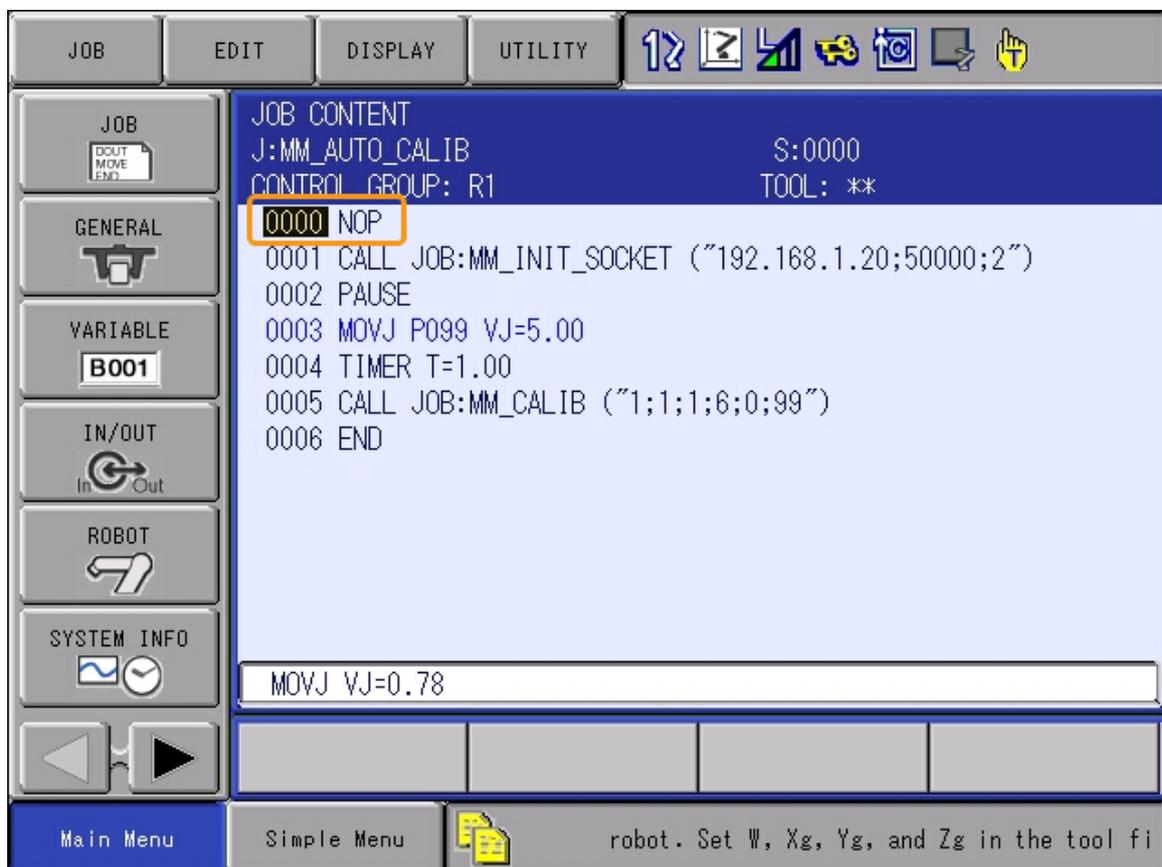
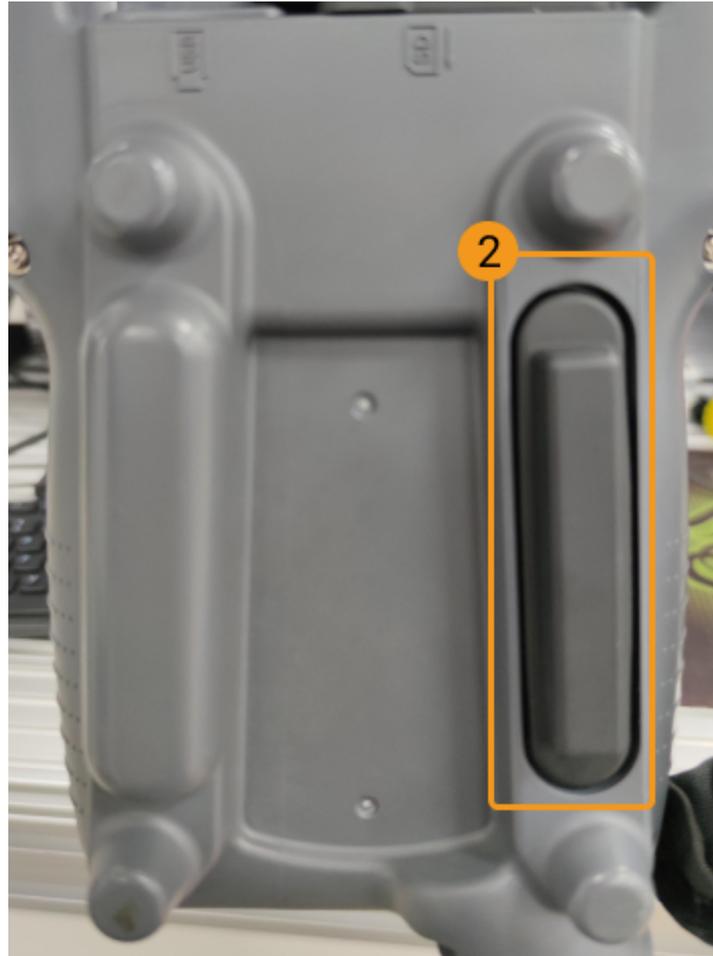






4. ティーチペンダントのキーを**TEACH**に合わせると**ティーチモード**になります。
5. ティーチペンダントの**SERVO ON READY**を押してから、背面の**イネーブルスイッチ**を押しながらカーソルを最初の行（0000）に移動します。

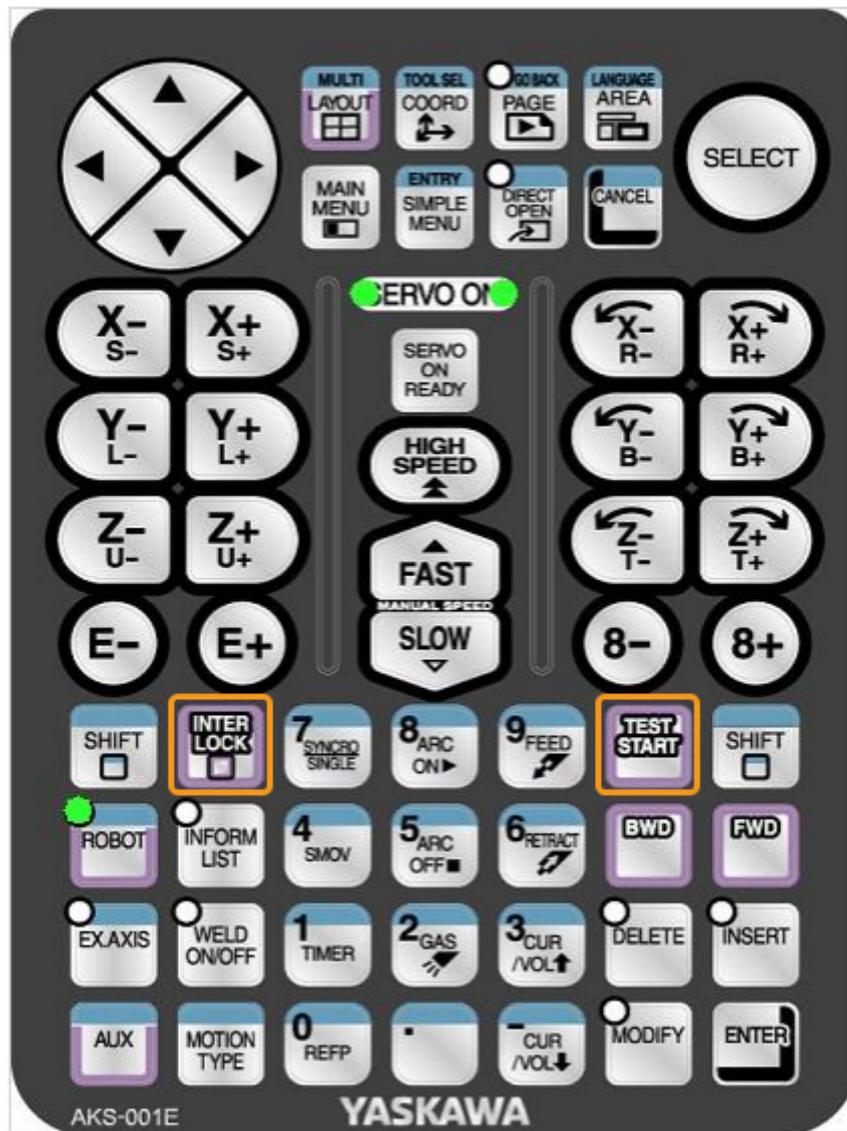


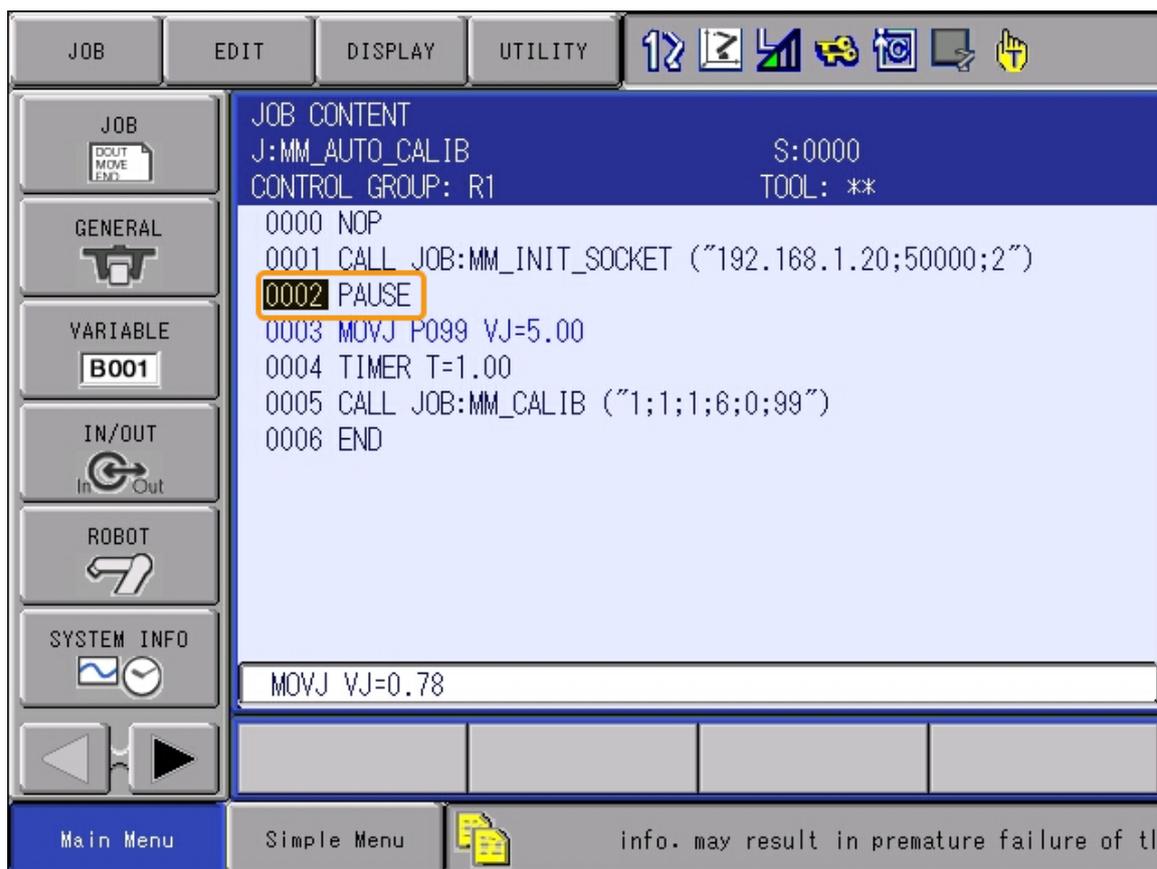




これは、プログラムが最初の行から実行されるようにすることです。

6. その後、ティーチパダントの**INTER LOCK**と**TEST START**を同時に押します。ロボットが0001行目の実行を終了すると、黒カーソルは自動的に**PAUSE（一時停止）**コマンドのある0002行目で停止します。





7. Mech-Visionのログバーの「Console」タブに通信の関連ログが表示されたかどうかを確認します。以下のログが表示された場合、Yaskawaロボットとビジョンシステムは標準インターフェースで正常に通信できるようになります。

ログ			
時刻	レベル	出力先	メッセージ
11:44:55.775	i	Console	TCP Serverインタフェースサービスを起動します。
11:44:58.306	i	Interface	クライアント 192.168.25.96:47752 が接続されています。



- テスト完了後、ロボットキャリブレーションに影響を与えないように **PAUSE** コマンドを削除する必要があります。**PAUSE** コマンドを削除するには、**PAUSE** コマンドのある0002行目にカーソルを移動し、ティーチペンダントの**DELETE**キーを押して**ENTER**キーを押します。

3.3. ロボットハンド・アイ・キャリブレーション

本ガイドを読むことで、自動ハンド・アイ・キャリブレーション（Eye to Hand）の実行方法を習得できます。



ハンド・アイ・キャリブレーションとは、カメラ座標系とロボット座標系との対応関係（外部パラメータ）を求めることです。ビジョンシステムにより取得した対象物の位置姿勢をロボット座標系に変換し、ロボットが対象物の把持を完了させるように制御します。

▶ <https://www.youtube.com/watch?v=w2vN5qXXJ8M/PLVcMd7cW2rXVtrAejMyVQni2dUDv8bxje> (YouTube video)

動画：ハンド・アイ・キャリブレーション（標準インターフェース通信）

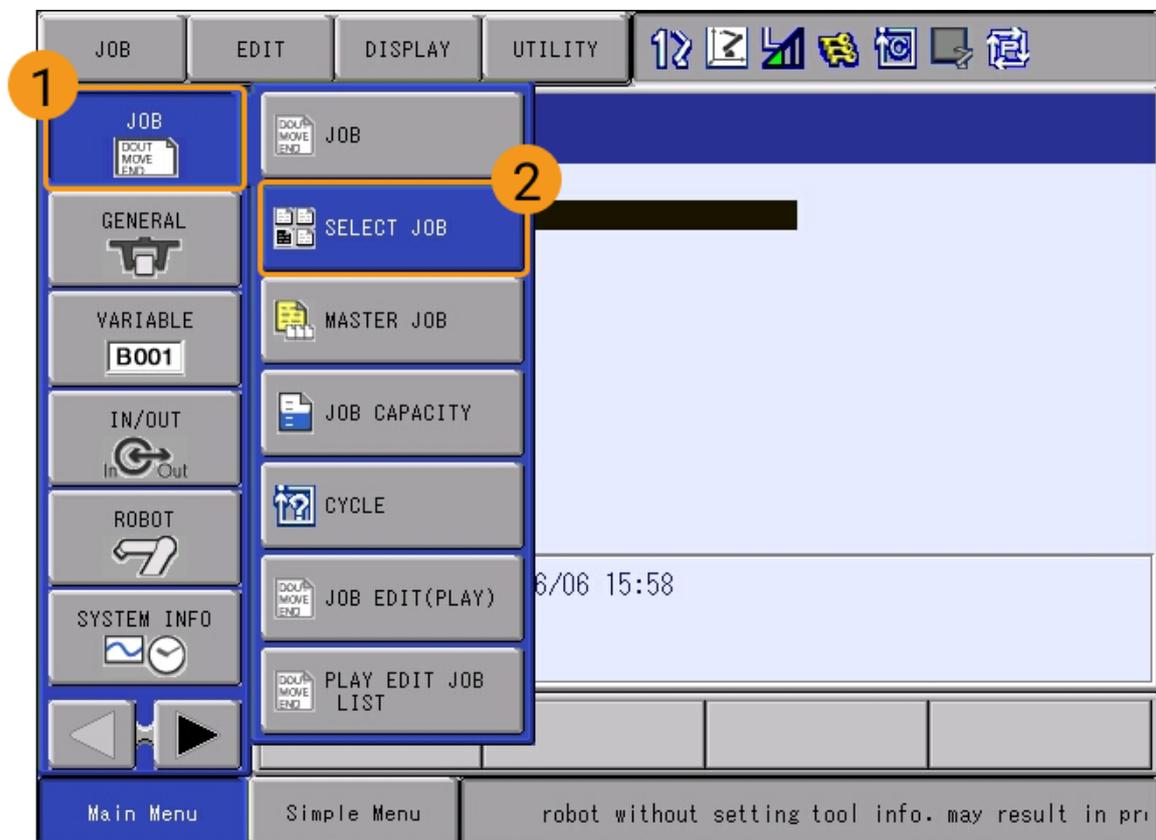
事前準備

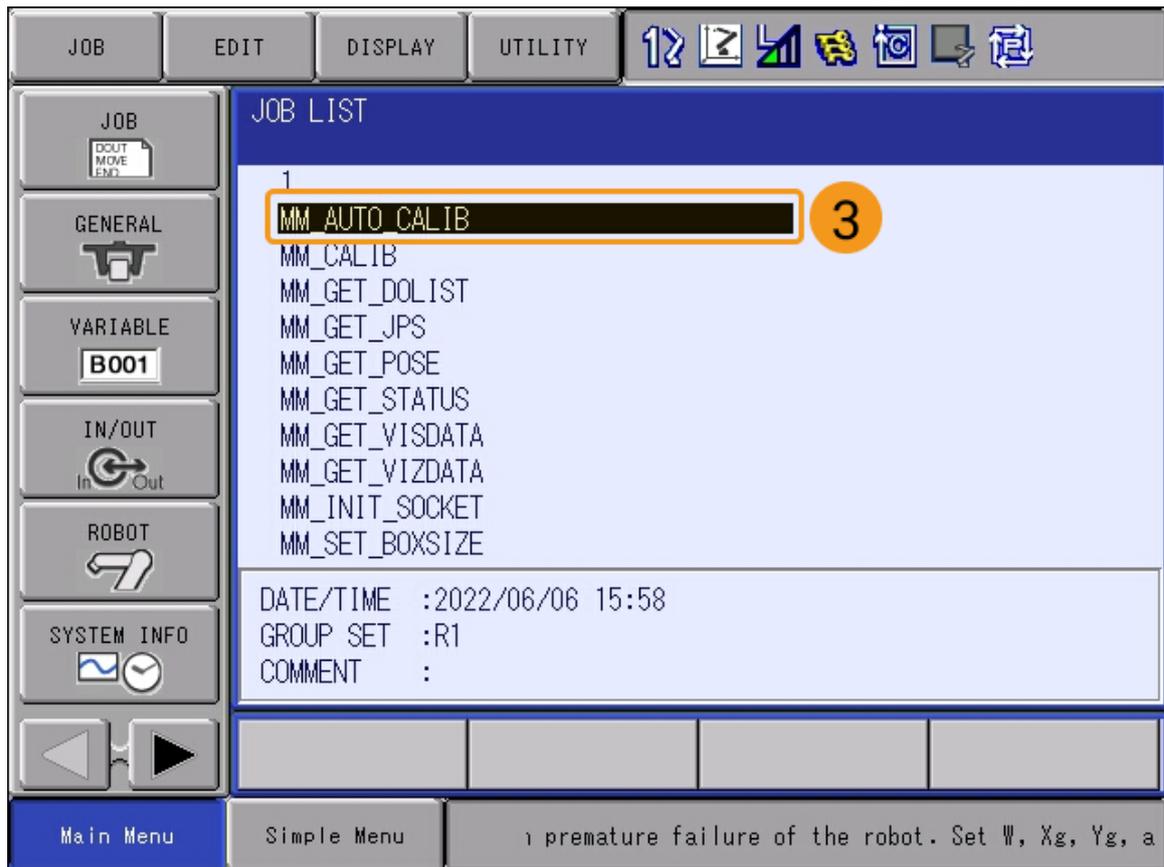
本節では、ロボットキャリブレーションプログラムの修正、キャリブレーションボードの取り付け、カメラパラメータの調整、キャリブレーション前の設定を行う必要があります。

ロボットキャリブレーションプログラム（MM_AUTO_CALIB）の修正

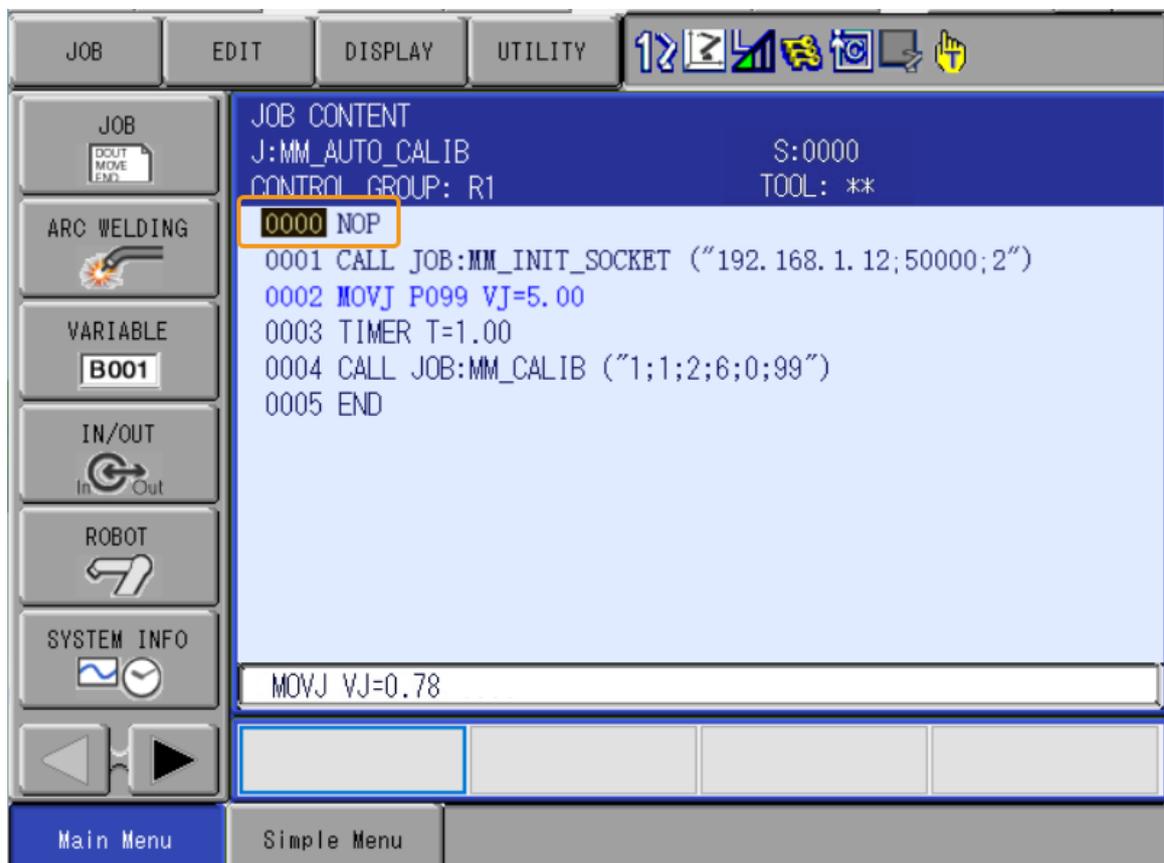
キャリブレーションするためのプログラムを選択

1. ティーチモードでは、ティーチペンダントで **JOB** ▶ **SELECT JOB** をクリックします。カーソルを **MM_AUTO_CALIB** に移動して、ティーチペンダントの **SELECT** を押します。





プログラムを選択した後、以下のような画面が表示されます。





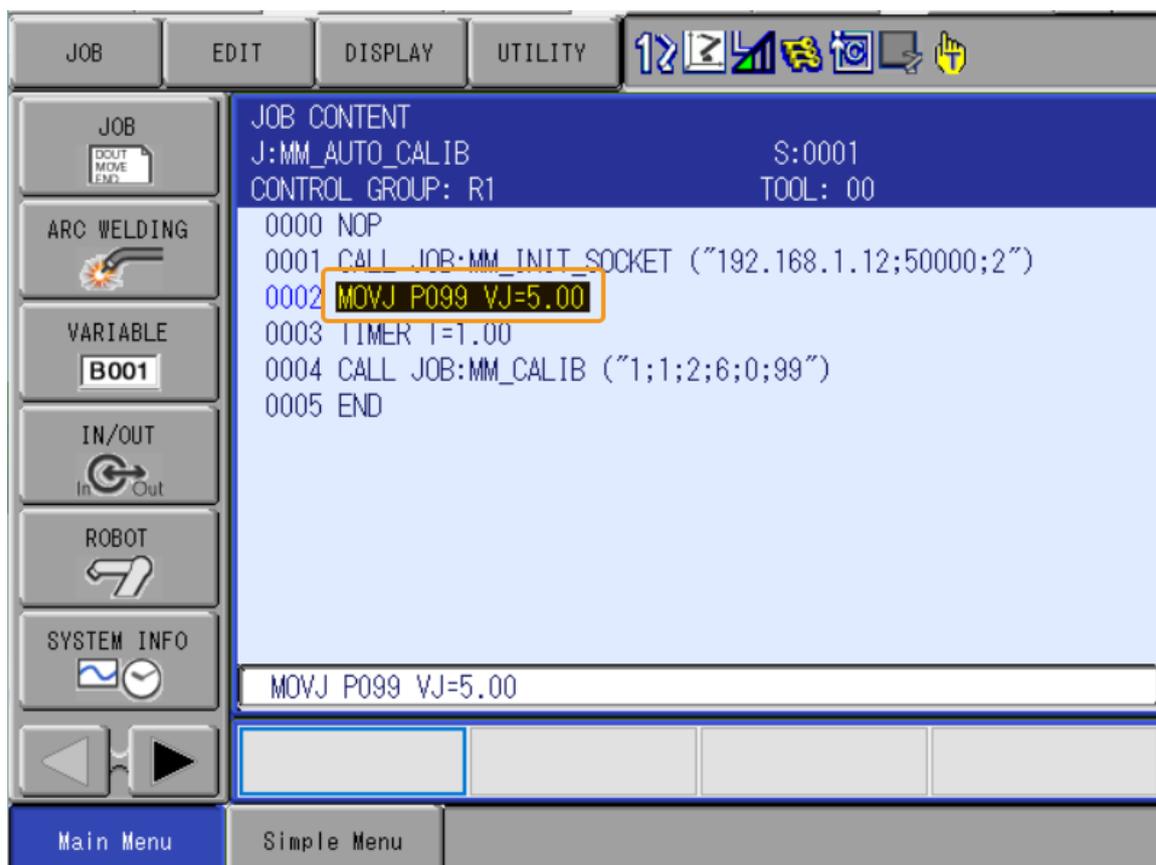
標準インターフェース通信をテストの節でIPCのIPアドレスとポートを変更しました。したがって、ここではキャリブレーションの初期位置を設定する必要があります。

キャリブレーションの初期位置を設定

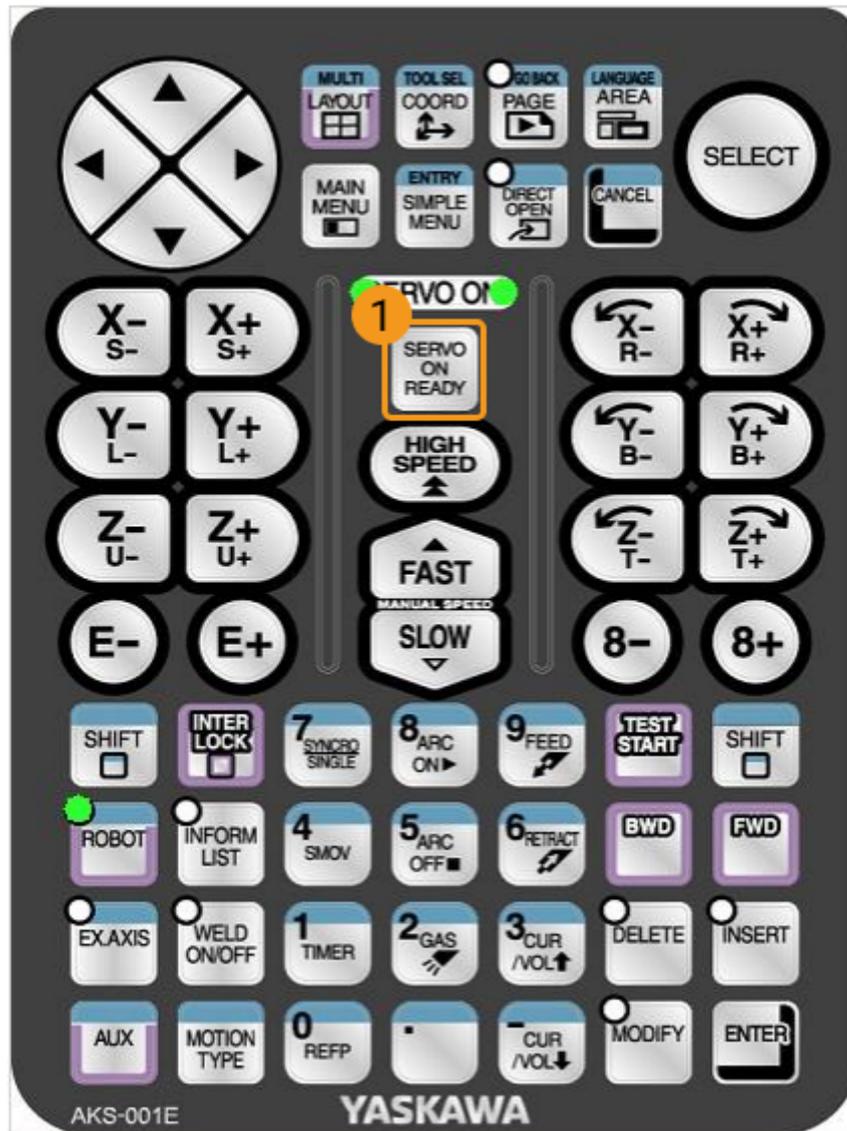


キャリブレーション中、ロボットがMech-Visionに送信された位置姿勢はフランジ位置姿勢である必要があります。したがって、キャリブレーションの初期位置を設定する前に、ツールIDをすべてのツールのオフセット値 (ToolOffset) が0のツールID (通常は00) に切り替える必要があります。ツール00のTCPはロボットのフランジにあります。

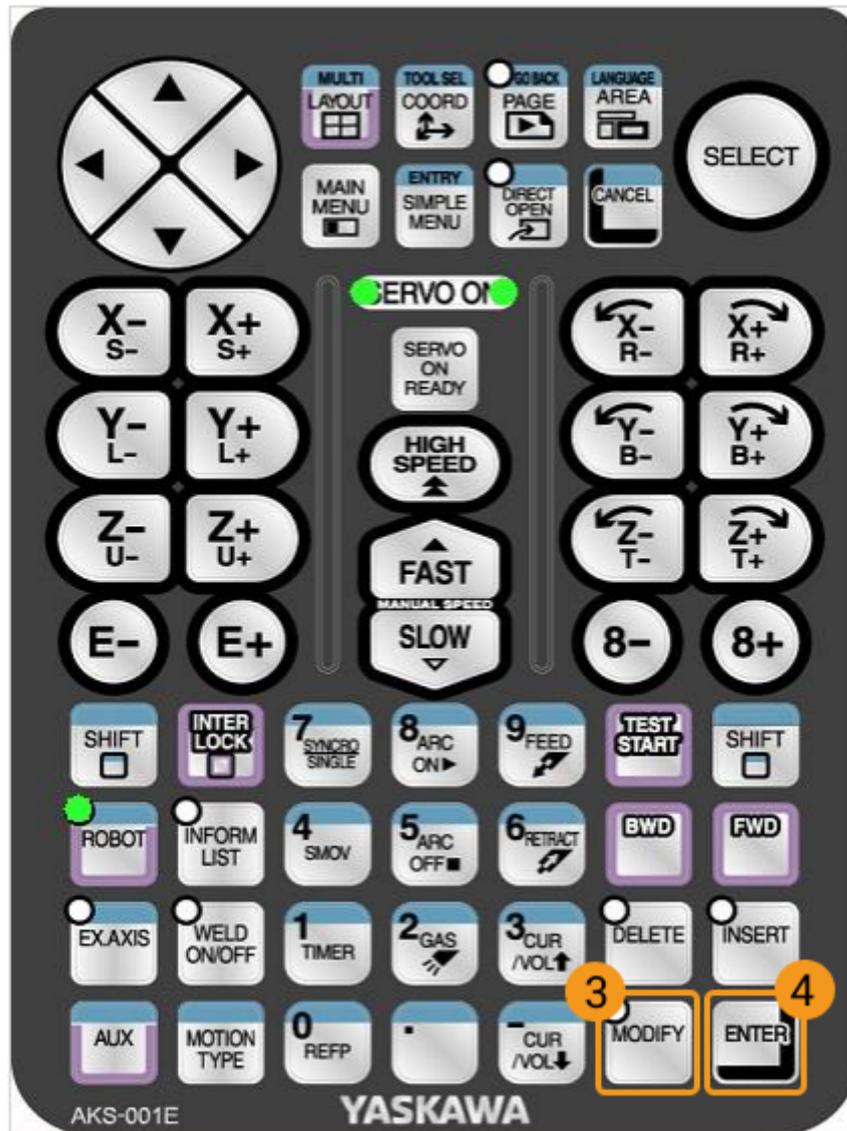
1. ロボットを手動でキャリブレーションの初期位置に移動させます。カーソルを **MOVJ P099 VJ=5** に移動し、ティーチペンダントの **DIRECT OPEN** キーを押します。

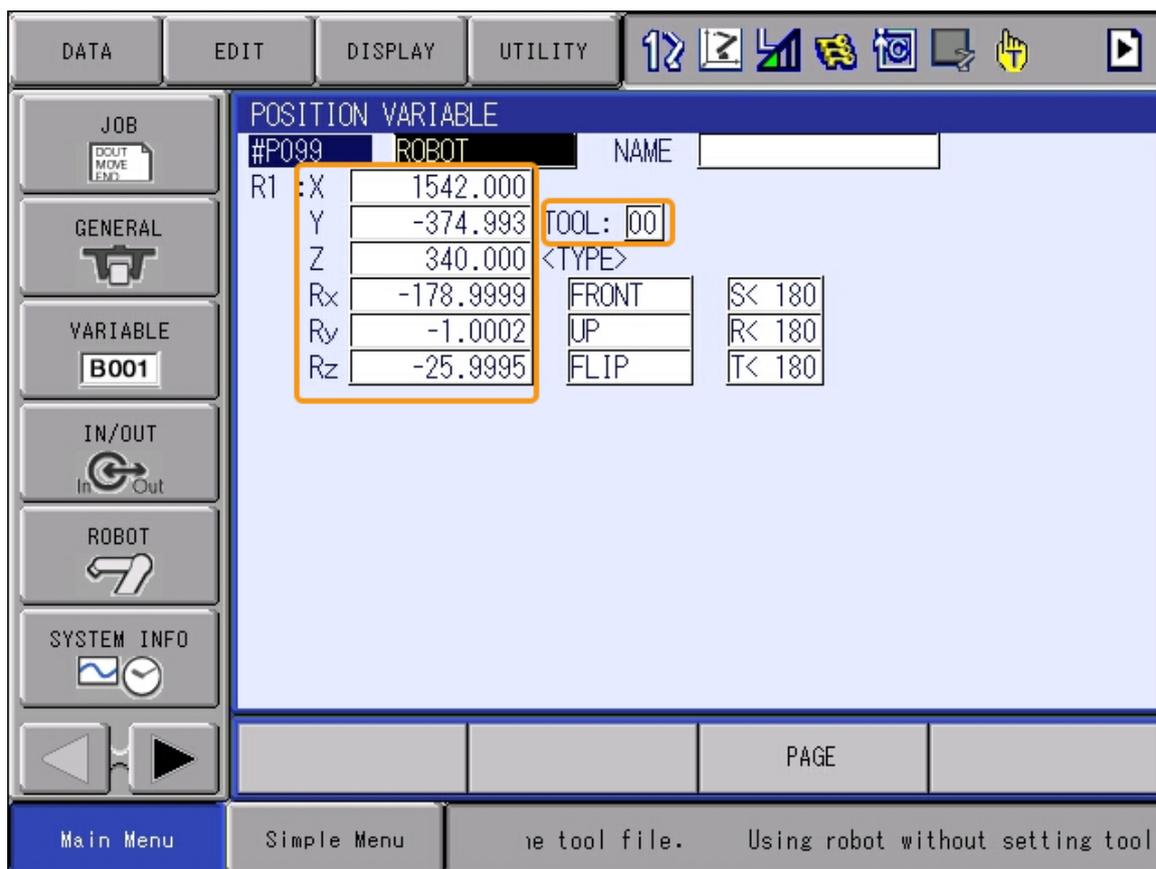


2. ティーチペンダントの **SERVO ON READY** を押した後、ティーチペンダントの背面の **イネーブ** **ルスイッチ** を押しながら、**MODIFY** を押して **ENTER** を押します。位置変数 **P099** を現在の位置に変更し、「TOOL」が **00** であることを確認します。









3. ティーチペンダントの **DIRECT OPEN** キーをもう一度押すと、**JOB** の画面に戻ります。

これで、Yaskawaロボットの自動キャリブレーションを実行するためのプログラムの修正が完了しました。

キャリブレーションボードの取り付け



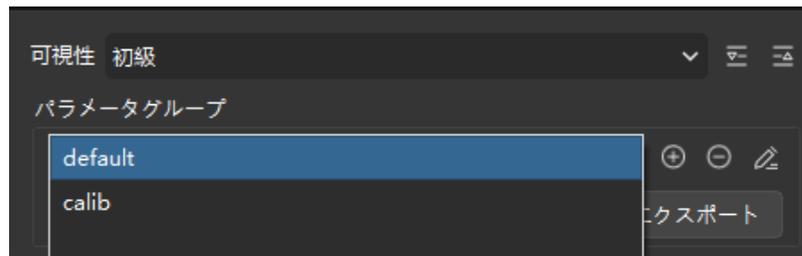
Eye to Handモードでは、キャリブレーションボードはロボット先端のフランジに取り付ける必要があります。

取り付け手順は以下の通りです。

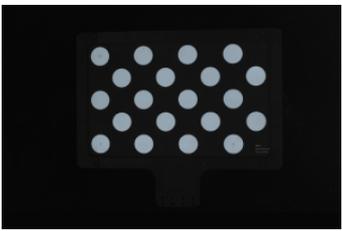
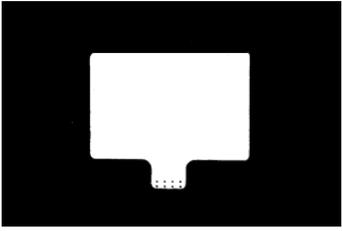
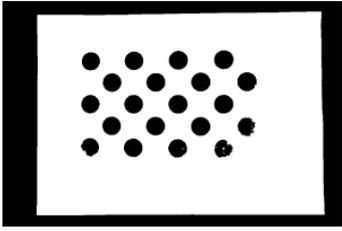
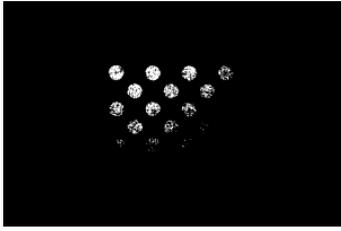
1. カメラ梱包箱からキャリブレーションボードとフランジを取り出します。
2. ネジ、スペーサー、ナットを使って、フランジをロボット先端に固定します。
3. ネジ、スペーサー、ナットを使って、キャリブレーションボードをフランジに固定します。
4. 取り付け後、ロボットを作業領域の最下層のワークの上面、カメラの視野中心に移動させます。

カメラパラメータを調整

1. Mech-Eye Viewerでカメラを接続し、**パラメータグループ**を「calib」に設定します。

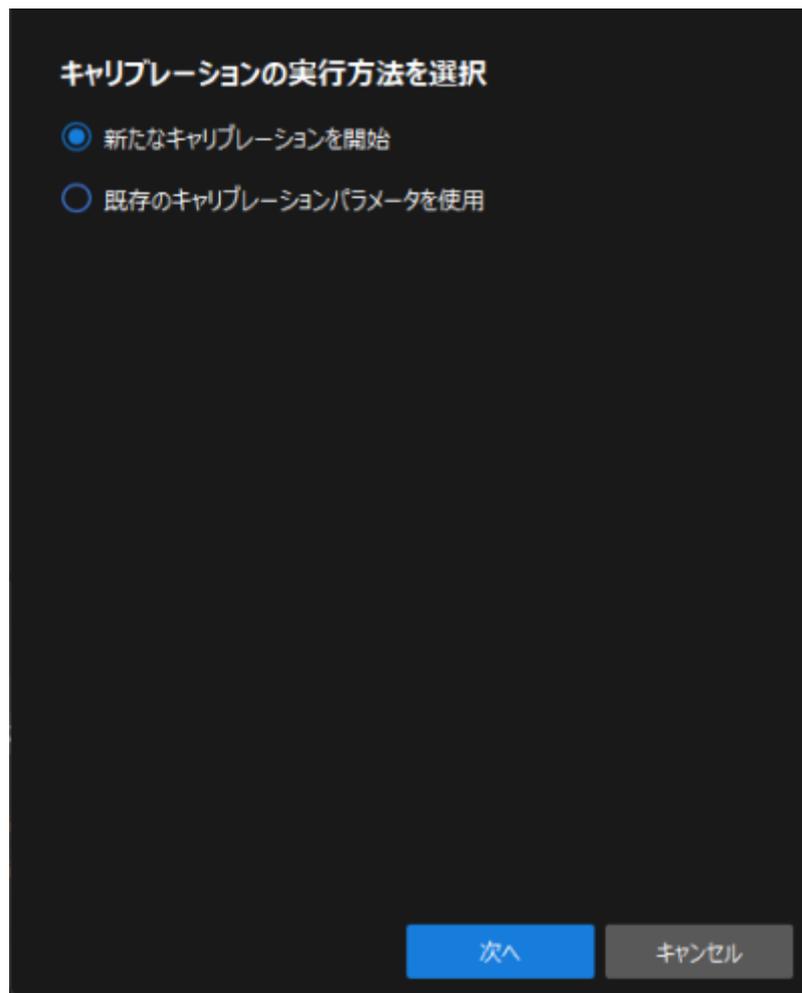


2. 2D画像内のキャリブレーションボードがはっきり見え、露出過度や露出不足がないように、2Dパラメータを調整します。
3. キャリブレーションボード上の白い円の点群が完全なものになるように、3Dパラメータを調整します。点群の変動範囲を減らすために、**点群後処理の表面平滑化と外れ値除去**を**Normal**に変更することをお勧めします。

	正常	露出過度	露出不足
2D画像			
点群			

キャリブレーション前の設定

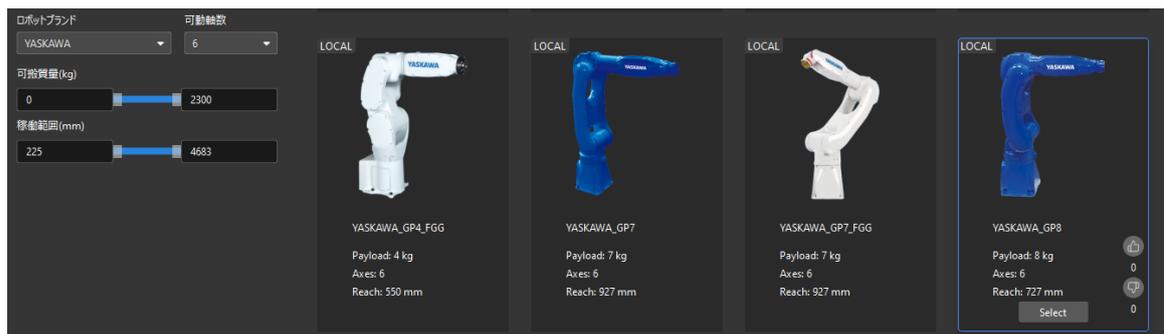
1. Mech-Visionのツールバーで[**カメラキャリブレーション (標準モード)**]をクリックします。すると、**キャリブレーション前の設定画面**が表示されます。
2. **キャリブレーションの実行方法**を選択画面で、**新たなキャリブレーションを開始**を選択し、[**次へ**]をクリックします。



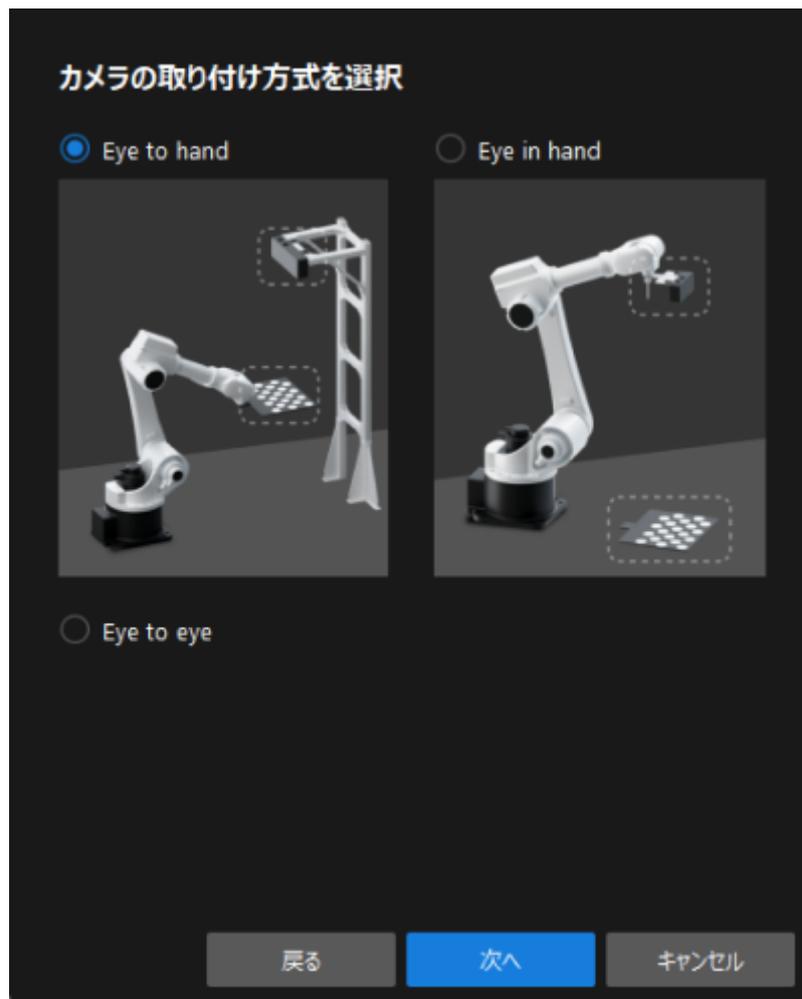
3. キャリブレーションのタスクを選択画面で、ドロップダウンリストボックスから適応可能なロボットのハンド・アイ・キャリブレーションを選択し、[ロボット型番を選択]をクリックします。



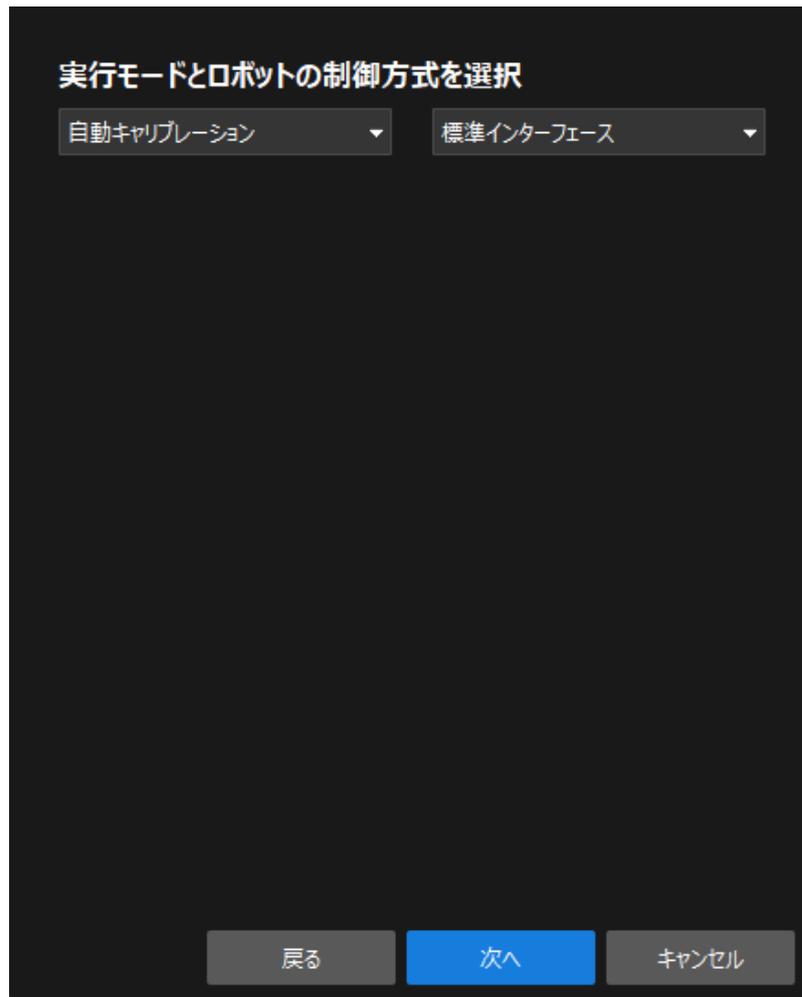
4. ロボットブランドのドロップダウンリストボックスから「YASKAWA」を選択し、右側で「YASKAWA_GP8」を選択して[選択]をクリックし、[次へ]をクリックします。



5. カメラの取り付け方式を選択画面で、Eye to handを選択して[次へ]をクリックします。



6. 実行モードとロボットの制御方式を選択 画面で、自動キャリブレーションと標準インターフェースを選択してから、[次へ]をクリックします。



7. 通信設定画面では、ホストIPアドレスのポートを50000に設定します。

通信設定 [マニュアルはこちら](#)

通信プロトコル

TCP Server

HEX (リトルエンディアン)

IPアドレスを設定

ホストIPアドレス

0

0

0

0

:

50000

ロボットプログラムの読み込み

プログラムフォルダを開く

ロボットを接続

接続されていません。

インターフェースサービスを起動

戻る

実行

キャンセル



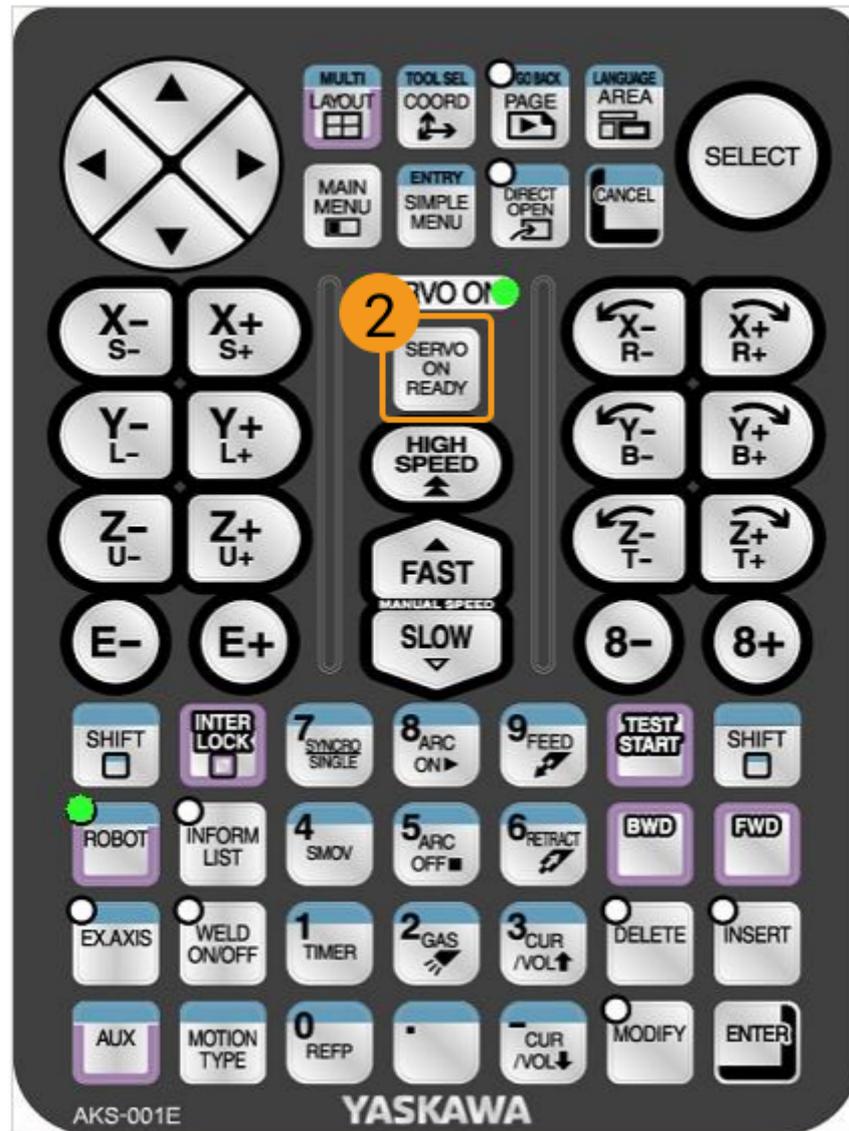
- ° 本ガイドでは、ビジョンシステムはTCPサーバーとし、**ホストIPアドレス**をIPCのIPアドレスとポートに設定する必要があります。IPCのIPアドレスはローカルIPアドレスで、設定する必要はありません。すべてのソフトウェアは「0.0.0.0」を使用します。
- ° IPCは、デフォルトでポート50000を使用してインターフェースサービスを提供します。必要に応じて変更することができます。このポートを変更した場合、ロボット側とビジョンシステムとの通信を確立する際に正しいポート番号を使用してください。

8. **ロボットを接続**エリアの下に[**ロボットを接続**]をクリックします。このボタンが**ロボットの接続を待機中...**に変わります。
9. ティーチペンダントでキャリブレーションプログラムを実行します。

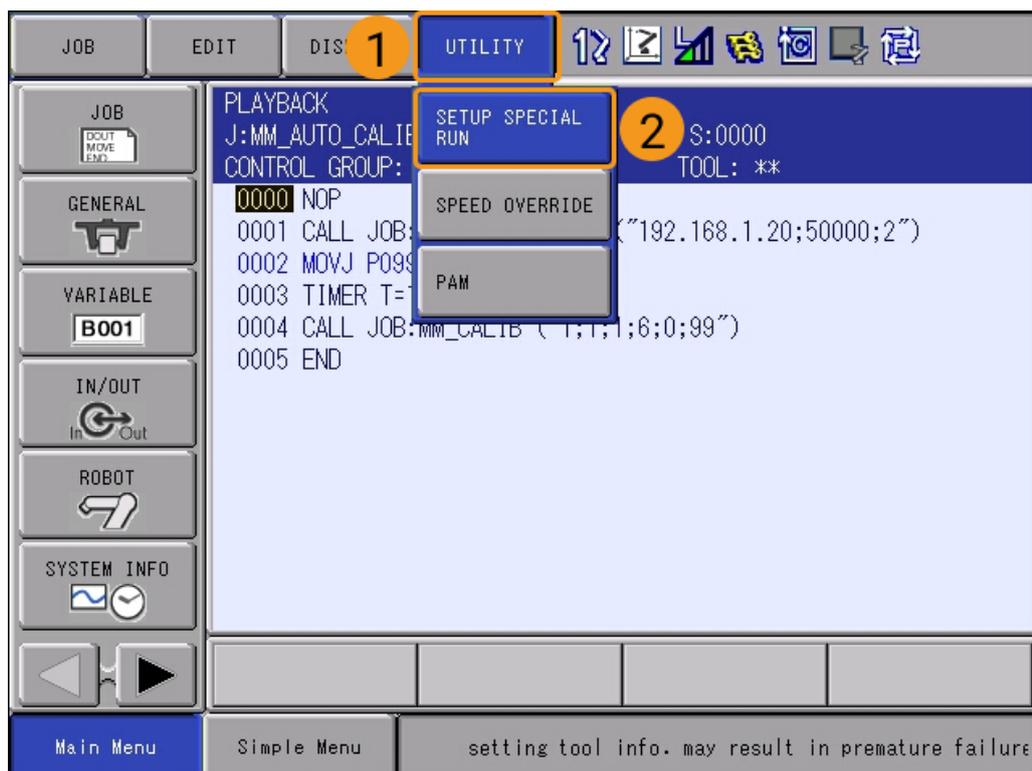
▼ (クリックして展開) 詳細方法

- a. まずはカーソルをキャリブレーションプログラムの1行目に移動し、ティーチペンダントのキーを**PLAY (中央位置)**に合わせると**再現モード**になります。その後、ティーチペンダントの**SERVO ON READY**キーを押します。





1. 低い速度で動作する必要がある場合、ティーチペンダントパネルの **UTILITY** > **SETUP SPECIAL RUN** をクリックして、**SPEED LIMIT** を **VALID** に変更します。



- b. ティーチペンダント上部の緑色の[スタート]ボタンを押します。スタートボタンのランプが点灯すると、ロボットが動作を開始しています。



キャリブレーションプログラムが正常に実行すると、Mech-Visionのロボットを接続エリアに「接続済」が表示されます。

- a. ロボット接続後、[実行]をクリックします。すると、キャリブレーション (Eye to Hand) 画面が表示されます。

キャリブレーションの実行手順

カメラの接続

1. カメラに接続手順で、検出されたカメラから接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。



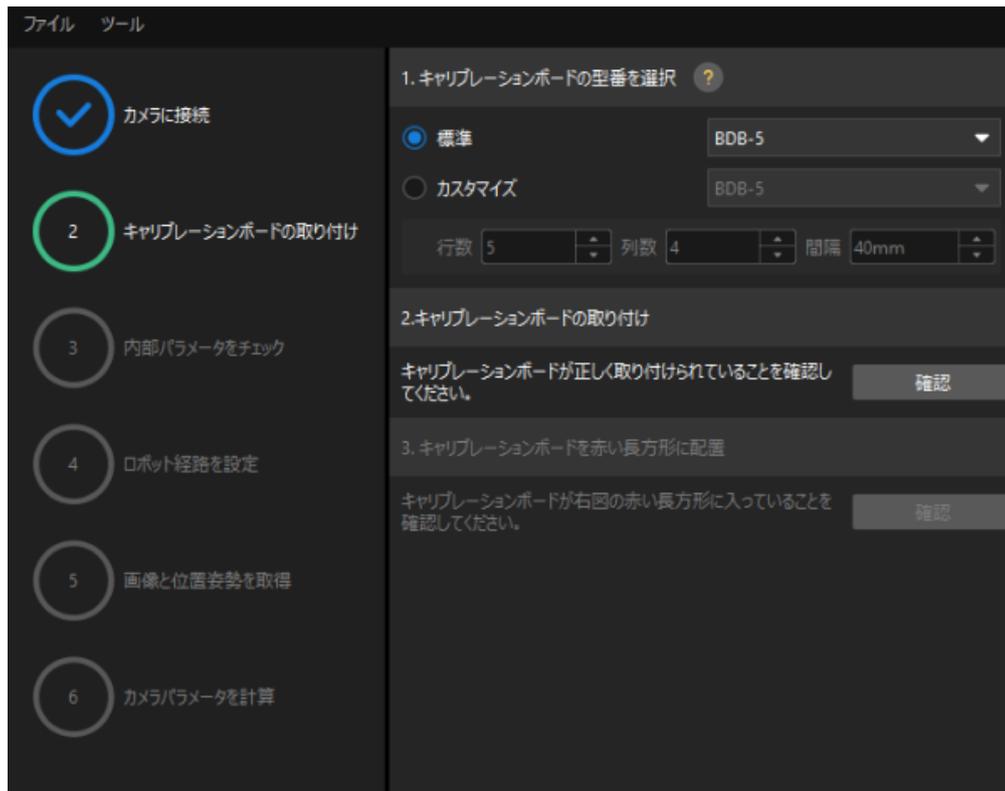
2. カメラ接続後、[**一回キャプチャ**]または[**連続キャプチャ**]をクリックします。
3. **画像ビュー**では、取得した2D画像と深度画像が要件を満たしていることを確認した上で[**次へ**]をクリックします。



取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露出パラメータを調整](#)して再取得する必要があります。

キャリブレーションボードの取り付け

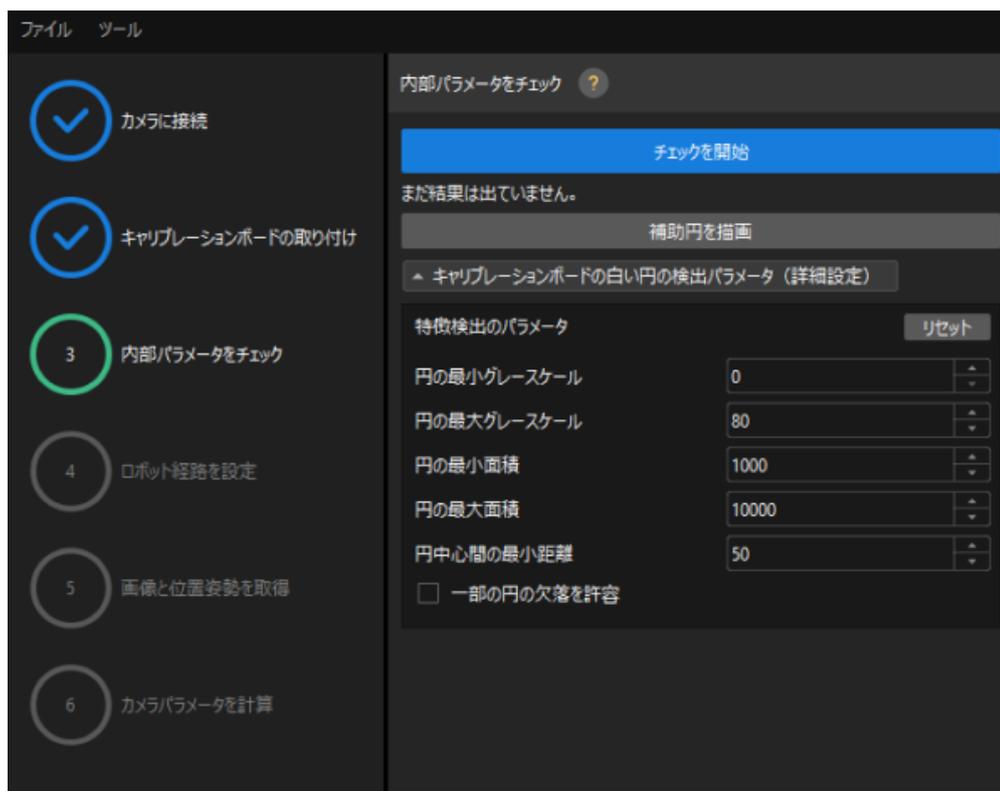
1. キャリブレーションボードの取り付け手順で、**1.キャリブレーションボードの型番を選択で標準**を選択し、実際に使用されるキャリブレーションボードに貼られているラベルに従って型番を選択します。
2. キャリブレーションボードをロボット先端のフランジにしっかりと取り付けたら、**2.キャリブレーションボードの取り付けで[確認]**をクリックします。
3. キャリブレーションボードがカメラの視野中央（赤い長方形内）に配置されることを確認したら、**3.キャリブレーションボードを赤い長方形に配置で[確認]**をクリックします。



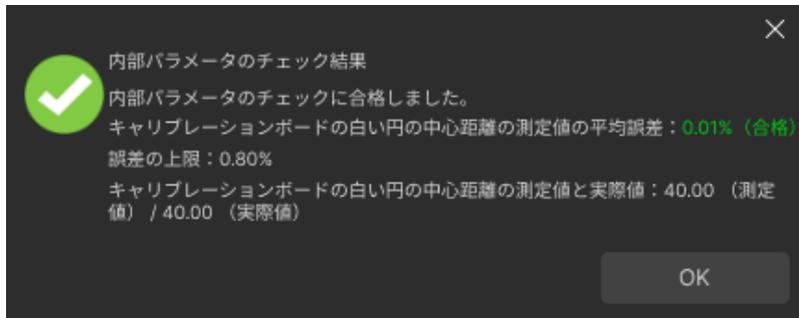
4. キャリブレーションボードに関するすべての操作が完了したら、下部にある[次へ]をクリックします。

カメラの内部パラメータをチェック

1. 内部パラメータをチェック手順で[チェックを開始]をクリックします。



2. カメラの内部パラメータのチェックが合格したことを確認したら、下部にある[次へ]をクリックします。



内部パラメータのチェックが合格しなかった場合、[補助円を描画するか](#)、[検出パラメータを編集](#)してください。

ロボット経路を設定

1. **ロボット経路を設定**手順で、**高さ範囲**を設定します。このパラメータは、キャリブレーションボードが深度方向における移動可能な範囲に応じて設定する必要があります。



2. 実際の状況に応じて、**経路のタイプ**をToHandに指定し、ピラミッド型の経路パラメータ下の**高さ範囲**、**層数**、**最下層サイズX/Y**、**最上層サイズX/Y**と**層ごとの移動グリッドの列と行数**を設定し、**回転角度**を設定してから、**[確認]**をクリックします。

i | ロボットの経路は、作業領域をカバーするように設定する必要があります。

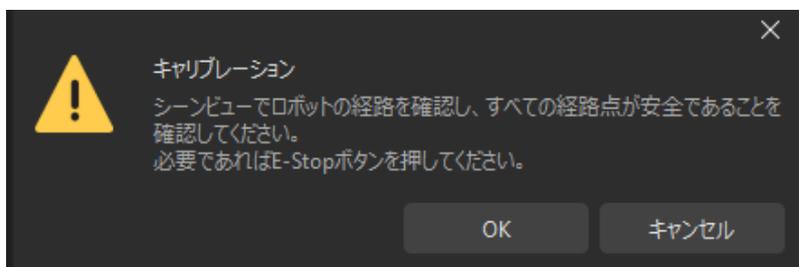
3. 右側の**シーンビュー**では、自動生成された動作経路の各ポイントが周囲と衝突していないことを確認してから、下部にある**[次へ]**をクリックします。

画像と位置姿勢を取得

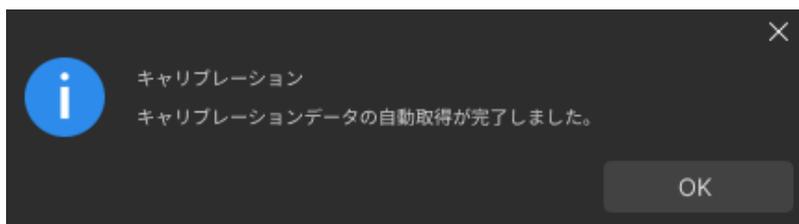
1. **画像と位置姿勢を取得手順**で、**画像を保存**にチェックを入れます。



2. [経路に沿ってロボットを自動で移動させ、画像を撮影]をクリックします。
3. ロボット動作の安全に関する説明をよく読んでから、[OK]をクリックします。

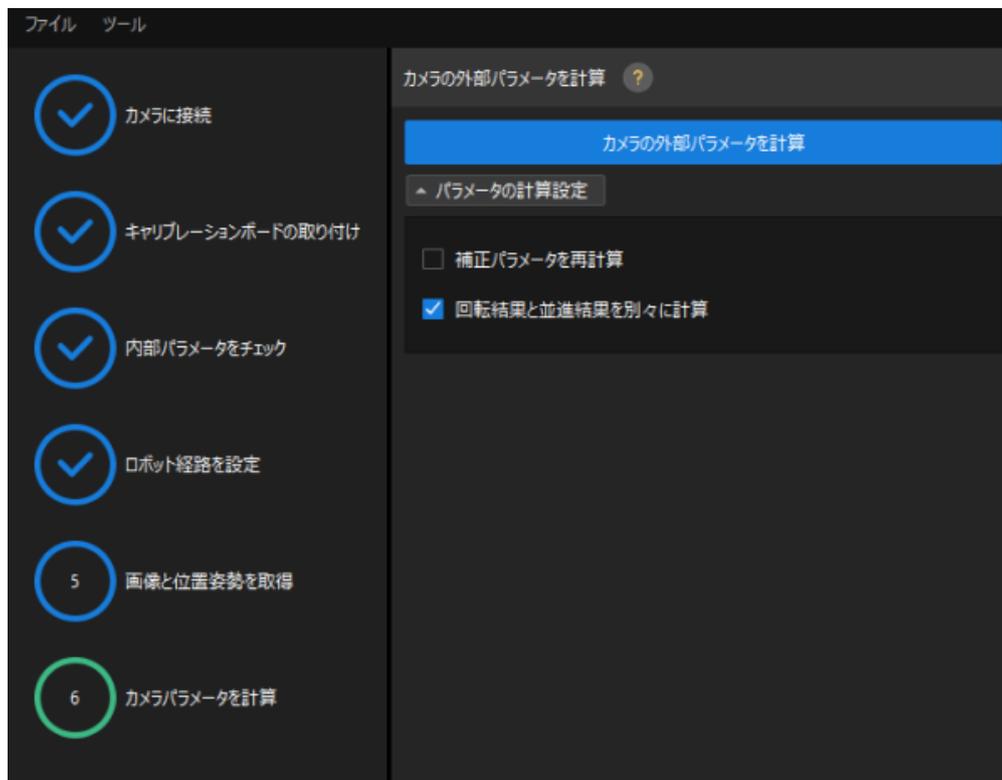


4. ロボットが設定した経路に沿って移動し、カメラが各経路点で画像取得を完了するのを待ちます。右側の画像と位置姿勢一覧には、取得した画像が表示されます。
5. キャリブレーションデータの自動取得が完了したら、ポップアップウィンドウで[OK]をクリックし、下部にある[次へ]をクリックします。



カメラパラメータを計算

1. カメラパラメータを計算手順で、[カメラの外部パラメータを計算]をクリックします。



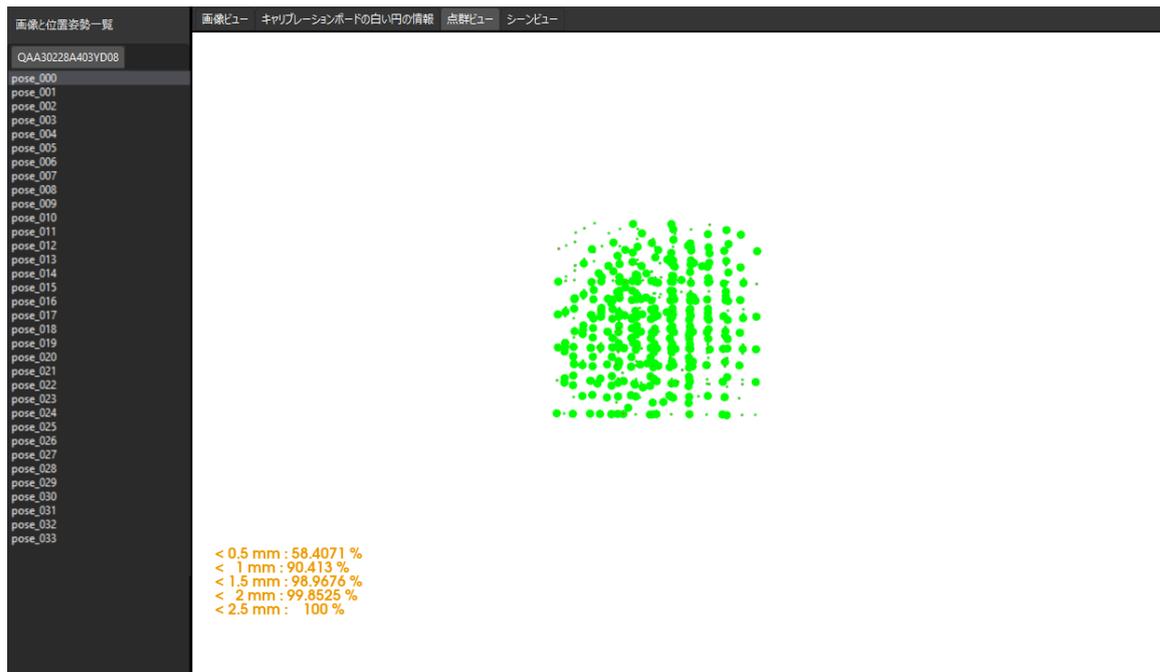
2. キャリブレーションが正常に実行されたことを示すダイアログで、[OK]をクリックします。
3. 右側の点群ビューパネルでキャリブレーションの誤差点群が表示されます。



誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実際の値と計算値との偏差を示すために使用されます。

4. キャリブレーション精度が要件を満たしていることを確認します。

キャリブレーション精度は、100%を占める誤差の値を求めることで判断できます。例えば、下図は2.5mm以下の精度を示しています。

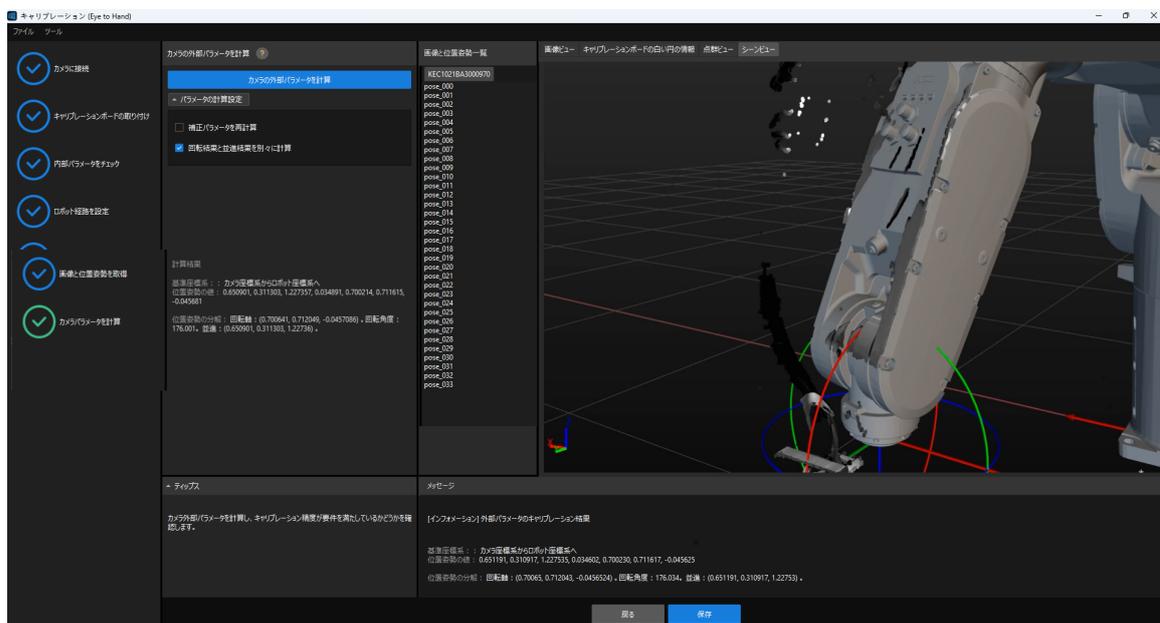


キャリブレーション精度を向上させるには、[キャリブレーション結果の分析](#)をご参照ください。

キャリブレーション結果の確認と保存

1. キャリブレーションが完了したら、ロボットアームをカメラの視野範囲内に移動させます。
2. カメラパラメータを計算手順で[外部パラメータを再計算]をクリックします。クリックすると、カメラが撮影を行います。
3. シーンビューに切り替え、ロボットの点群がロボットモデルの重なりを確認します。

ロボットの点群がロボットモデルとほぼ一致すれば、キャリブレーションは成功です。



- 下部にある[保存]をクリックし、ポップアップする**キャリブレーションファイル**を保存画面で[OK]をクリックします。すると、カメラのキャリブレーション結果は、プロジェクトの「calibration」ディレクトリに自動的に保存されます。

3.4. ワーク認識

本ガイドを読む前に、[Mech-Visionソリューションの作成と保存](#)を参照し、「一般的な部品認識」を使用してMech-Visionソリューションを作成する必要があります。

まず、プロジェクトの構築手順を確認し、ステップパラメータの調整とプロジェクト実装により、ワークの位置姿勢を取得してビジョン結果を出力します。

▶ <https://www.youtube.com/watch?v=CFNR4zeyR0k/PLVcMd7cW2rXVtrAejMyVQni2dUDv8bxje> (YouTube video)

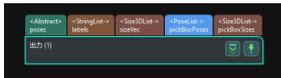
動画：ワーク認識



本ガイドでは、Mech-Visionプロジェクトを一回実行すると1つのビジョンポイントを出力できます。

プロジェクトの構築手順

使用するステップと機能は下表のとおりです。

番号	階段	ステップ	サンプル図	説明
1	画像取得	カメラから画像を取得		カメラを接続して画像を撮影します
2	ワーク認識	ワーク認識		3Dマッチングにより、ワークの位置姿勢（把持位置姿勢として使用）を計算します
3	位置姿勢調整	位置姿勢を一括調整		把持位置姿勢をカメラ座標系からロボット座標系に変換します
4	ビジョン結果を出力	出力		把持するワークの位置姿勢を出力します



把持位置姿勢とは、ロボットがワークに対してピッキング可能な位置のことを指します。

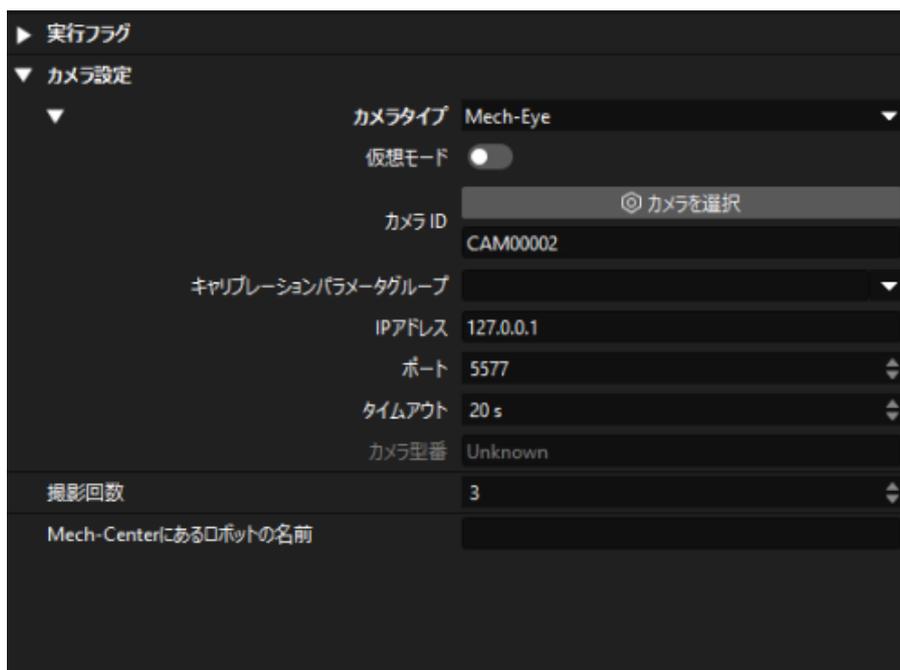
ステップパラメータの調整

以下では、パラメータの調整について説明します。

カメラから画像を取得

「カメラから画像を取得」のステップパラメータを調整し、カメラに接続します。

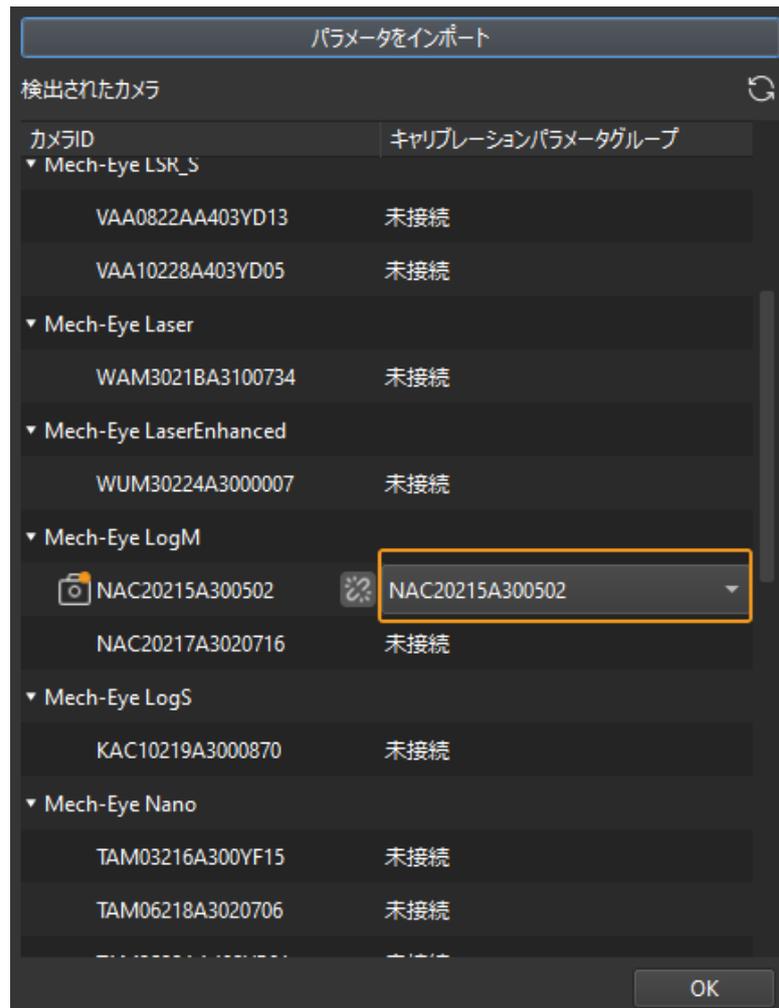
1. 「カメラから画像を取得」をクリックして選択し、ステップパラメータで **[カメラを選択]** をクリックします。



2. 表示される画面で  をクリックすると、カメラの接続が完了です。カメラが正常に接続された場合、 は  に変わります。



カメラ接続後、パラメータグループを選択します。[パラメータグループを選択]をクリックし、表示されるパラメータグループを選択します。



3. カメラを接続し、パラメータグループを設定すると、キャリブレーションパラメータグループ、IPアドレス、ポートなどのパラメータが自動的に入力されます。それ以外のパラメータ設定は不要です。

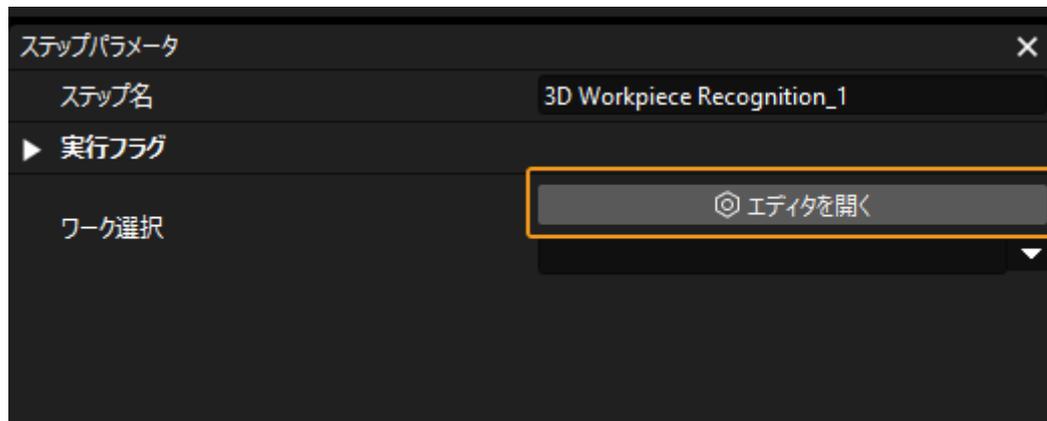


これで、カメラの接続が完了しました。

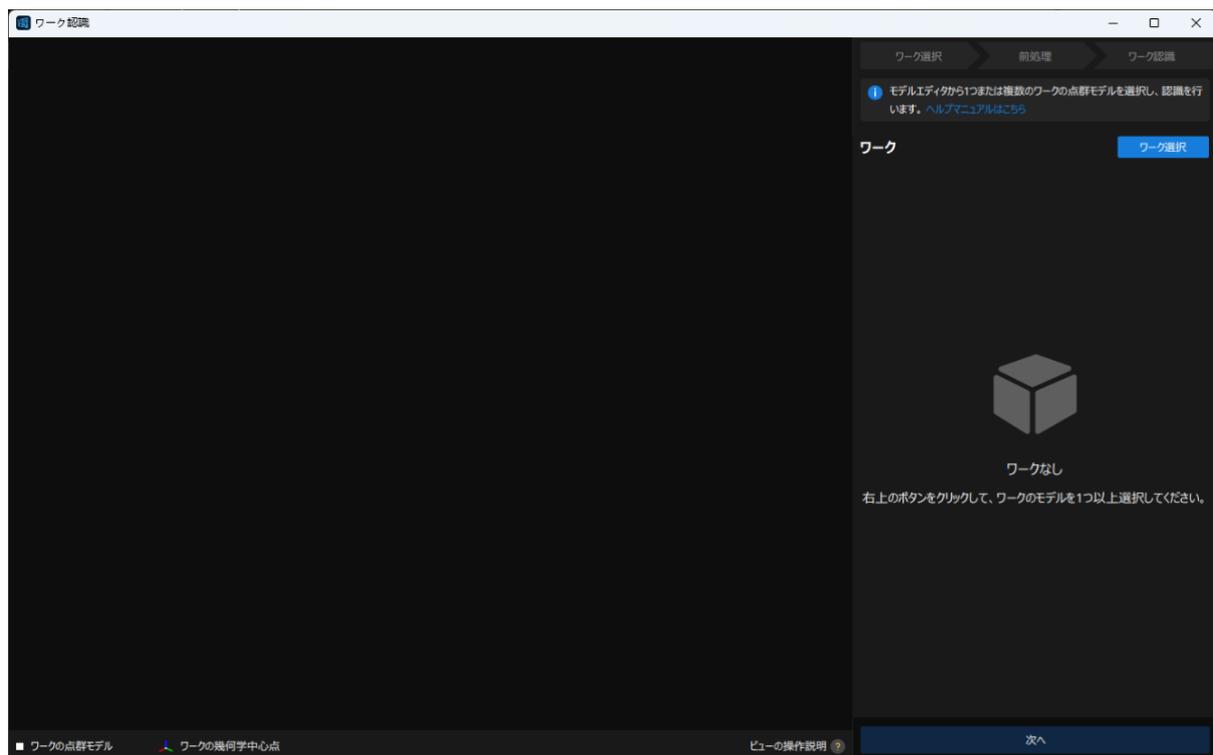
ワーク認識

「ワーク認識」ステップには、可視化設定ツールに組み込まれています。このツールは、点群前処理、モデルマッチング、ワークの位置姿勢（把持位置姿勢）の計算に対応できます。

「ワーク認識」ステップをクリックして選択し、ステップパラメータで[**エディタを開く**]をクリックします。



「ワーク認識」の可視化設定ツールを下図のように示します。



次に、下図のような手順でワーク認識を行います。

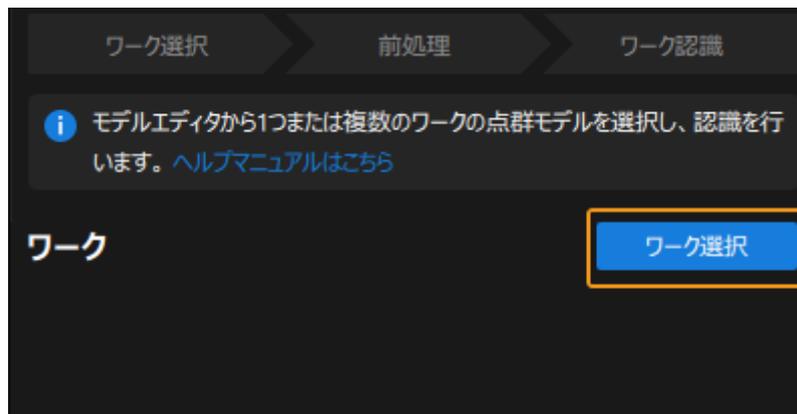


ワーク選択

「ワーク認識」の可視化設定ツールを起動した後、認識するワークの点群モデルを作成する必要があります。

1. モデルエディタを開きます。

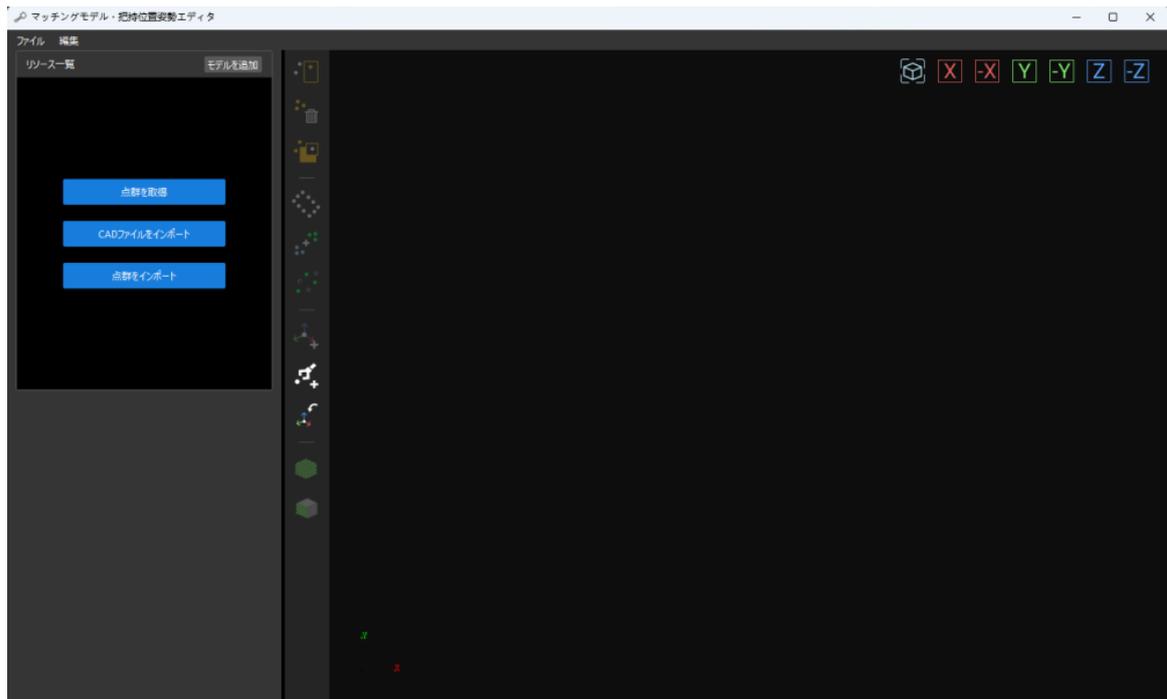
画面の右上隅にある [ワーク選択] をクリックします。



表示される画面で [モデルエディタを開く] をクリックします。



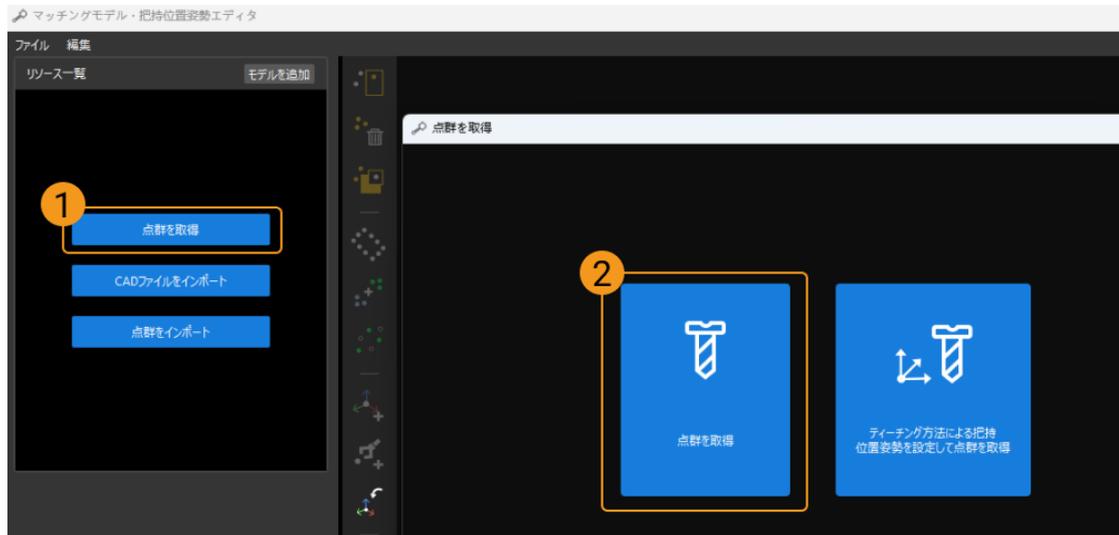
モデルエディタの画面を下図に示します。



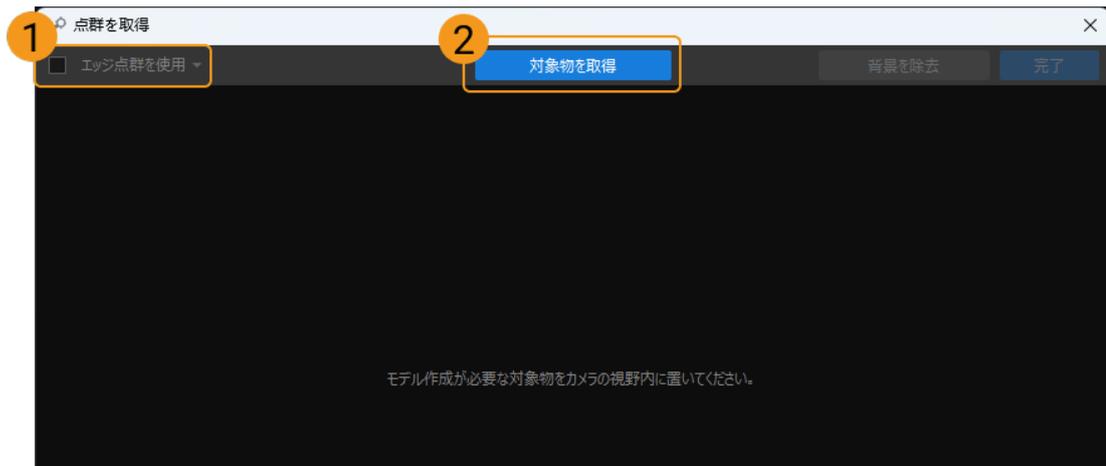
2. 点群取得により点群モデルを生成します。

a. 対象物の深度画像を取得します。

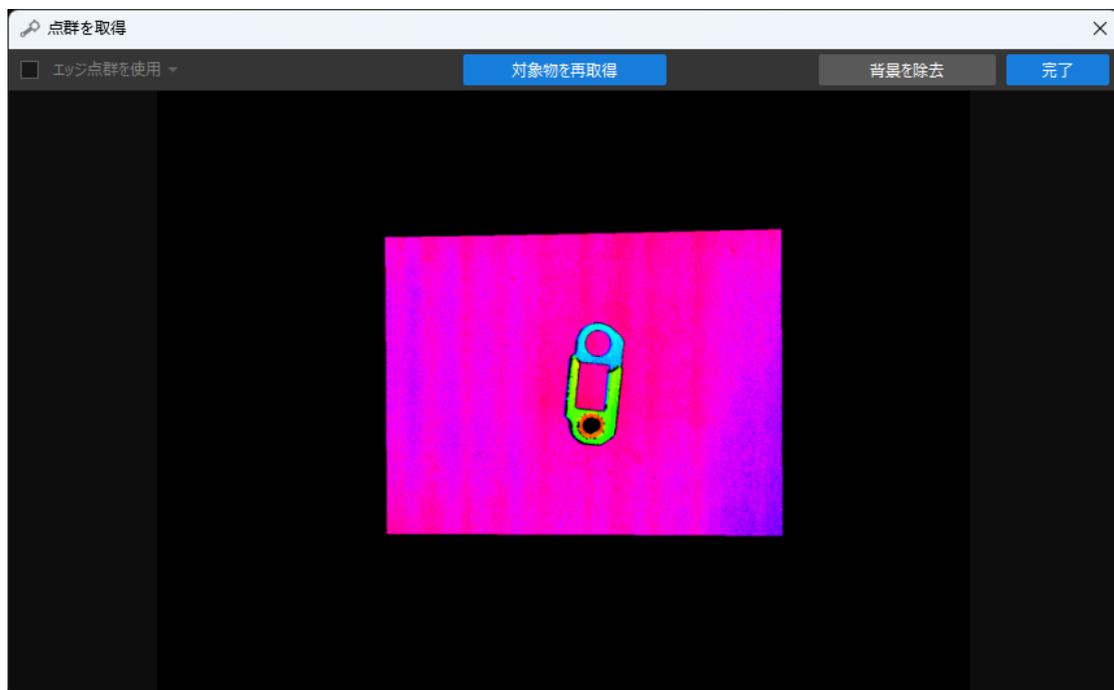
[点群を取得] をクリックして、表示される画面で [点群を取得] をクリックします。



トラックリンクの表面特徴が多いため、サーフェスモデルを作成することを推奨します。ここでは、**エッジ点群を使用**のチェックを外します。その後、[対象物を取得] をクリックして対象物の深度画像を取得します。

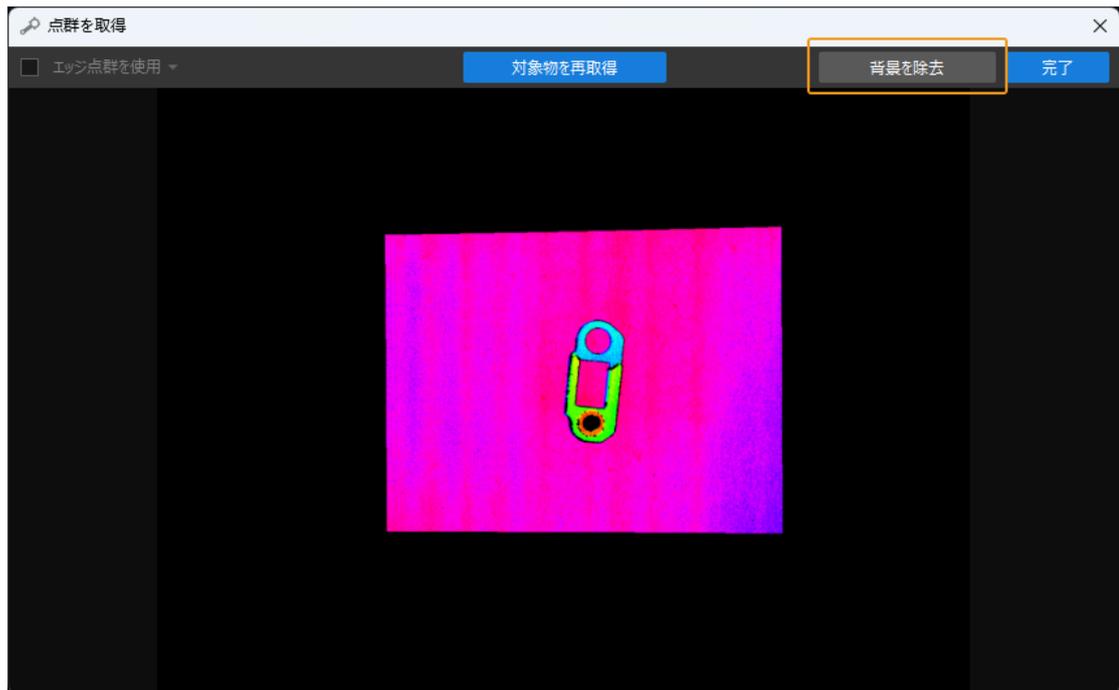


取得後、対象物と背景の深度画像は下図のようになります。



b. 背景画像を取得します。

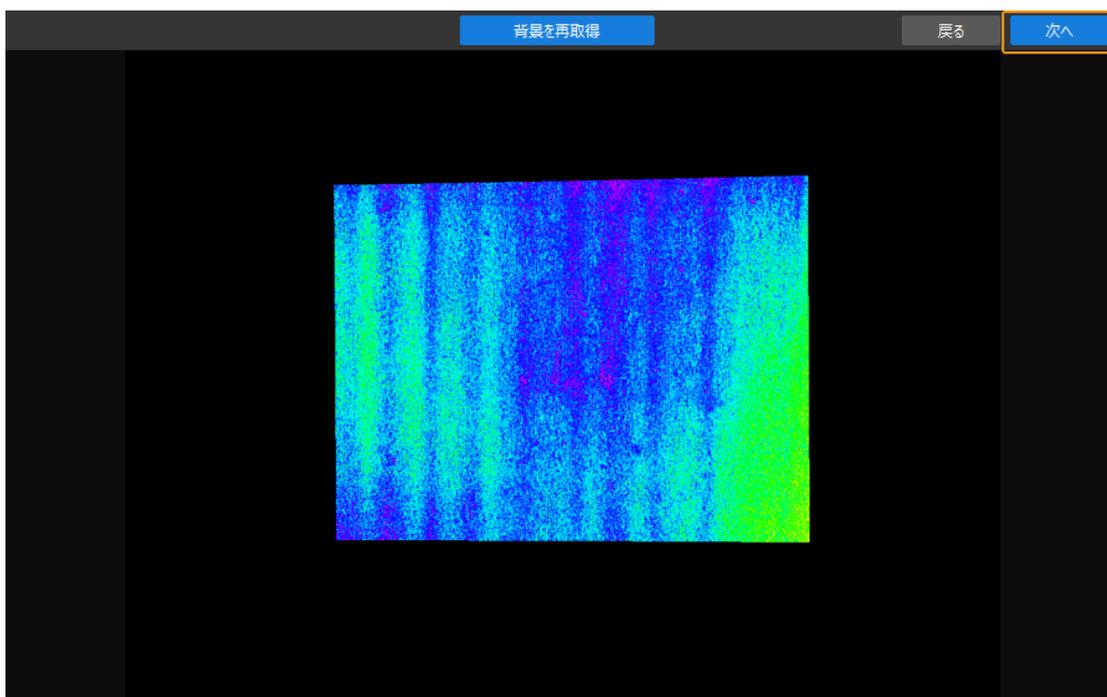
右上隅にある [背景を除去] をクリックします。



次に、対象物をカメラの視野外に移動させます。[背景を取得]をクリックして背景を取得します。



背景取得後の画像は下図のようになります。右上隅にある[次へ]をクリックします。



c. 背景を除去します。

この画面で [背景を除去] をクリックすると、背景除去後の対象物は下図のようになります。



右下隅にある [完了] をクリックすると、背景が除去された対象物が「マッチングモデル・把持位置姿勢エディタ」にインポートされます。



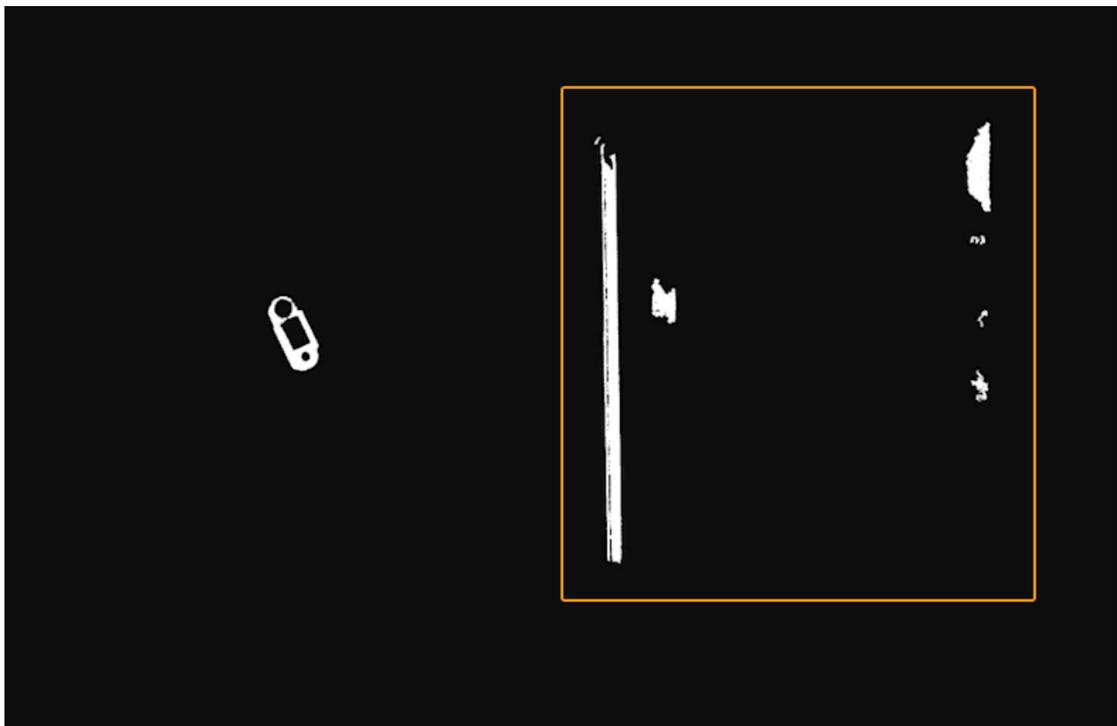
3. 点群モデルを編集します。

生成された点群モデルは実際の応用シーンに適さない可能性があるため、ノイズ除去や点群ダウンサンプリングなどの編集が必要となります。

a. ノイズを除去します。

 をクリックした後、除去するノイズを選択して  をクリックします。すると、選択した領域内のノイズが除去されます。

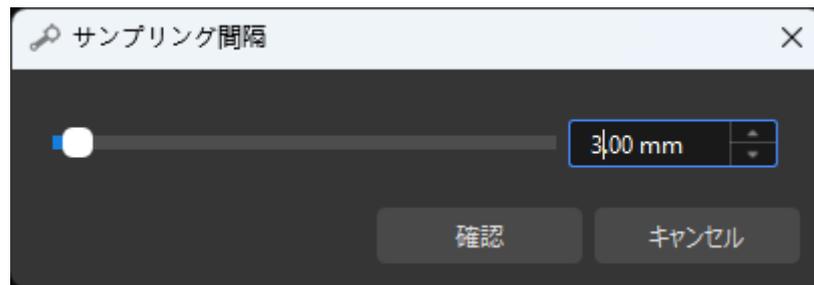
下図に示すように、選択した点はノイズであり、上記の方法で除去することが可能です。



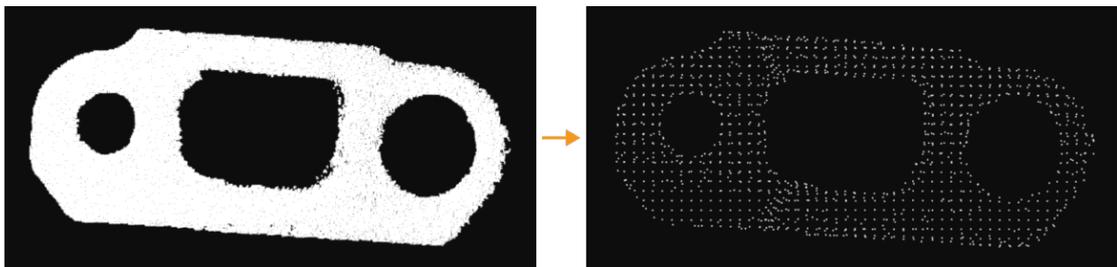
b. 点群ダウンサンプリングを行います。

点群ダウンサンプリングの目的は、モデル点群の点数を減らし、モデルマッチングの効率を向上させることです。

 をクリックし、表示される画面でサンプリング間隔を設定します。



例えば、サンプリング間隔を3mmに設定すると、ダウンサンプリング前後の点群モデルを下図に示します。

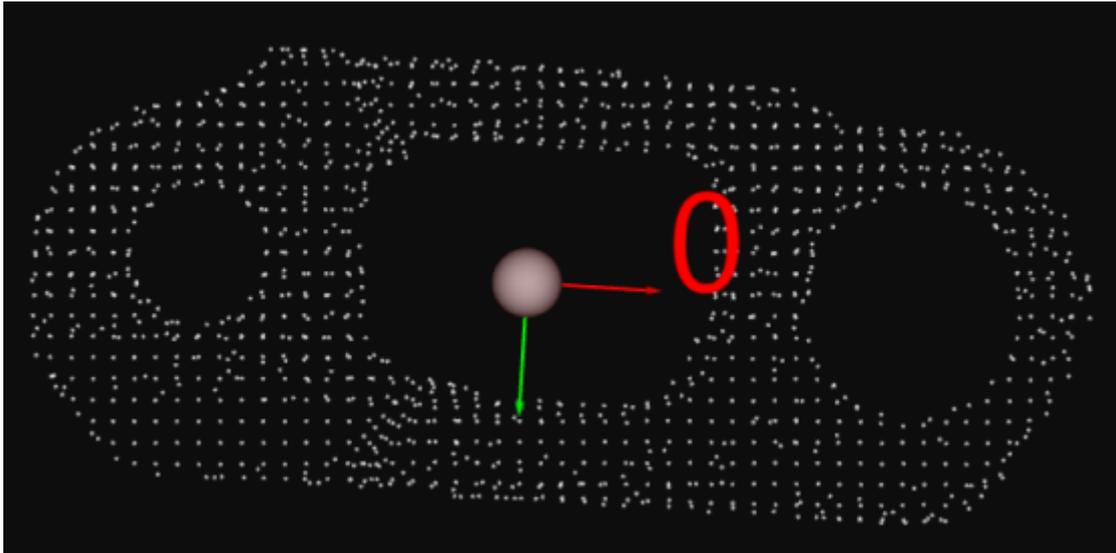


4. 把持位置姿勢を追加します。

ツールバーの  をクリックし、把持位置姿勢として使用する位置姿勢を、ワークの点群モデルに追加します。

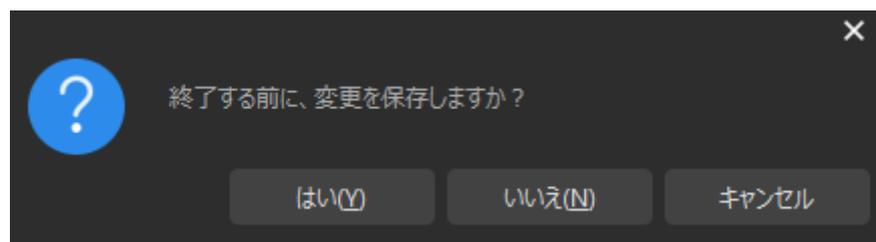


追加された把持位置姿勢を下図に示します。



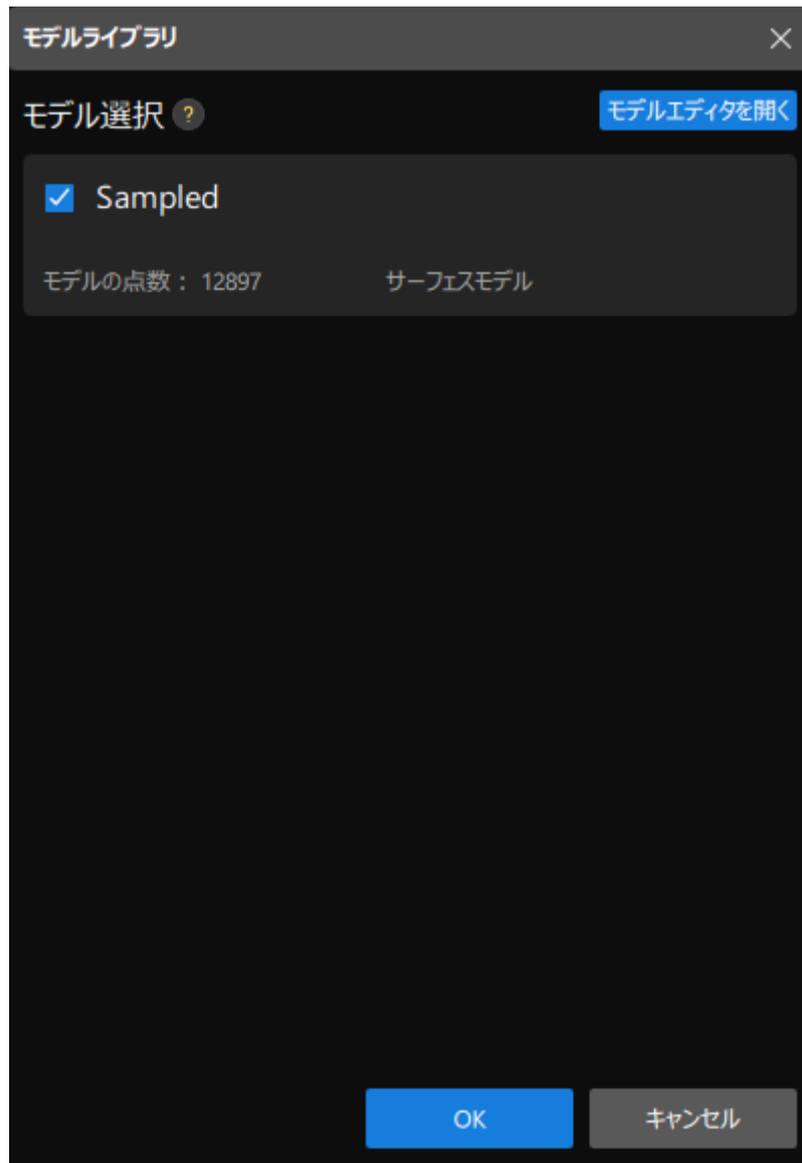
5. モデルと把持位置姿勢を保存します。

モデルエディタの終了ボタンをクリックし、表示される画面で[はい (Y)]をクリックします。

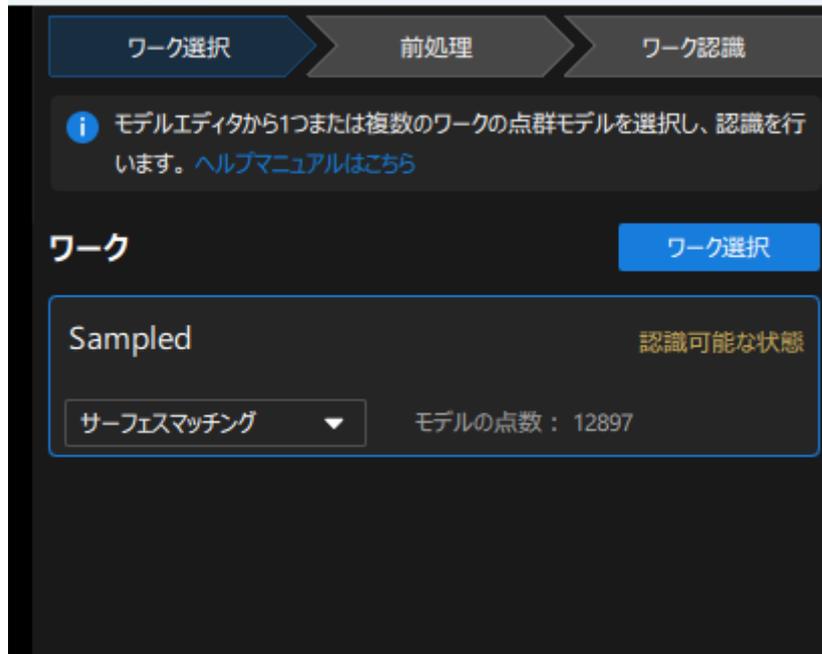


6. モデルライブラリからワークのモデルを選択します。

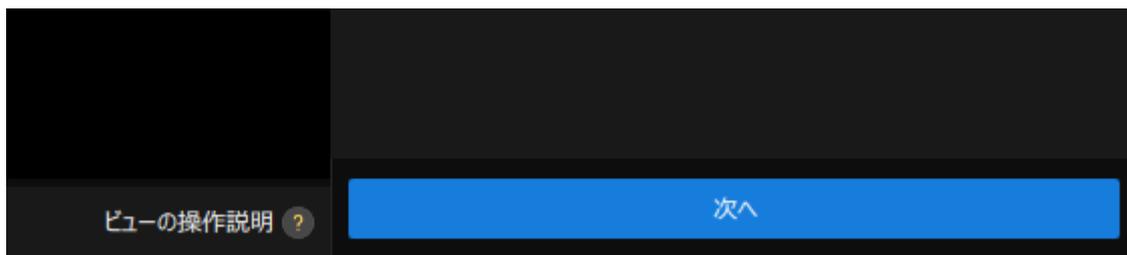
モデルエディタを終了したら、モデルライブラリから保存された点群モデルにチェックを入れ、[OK]をクリックします。



すると、認識するワークは可視化設定ツールの右上隅に表示されます。



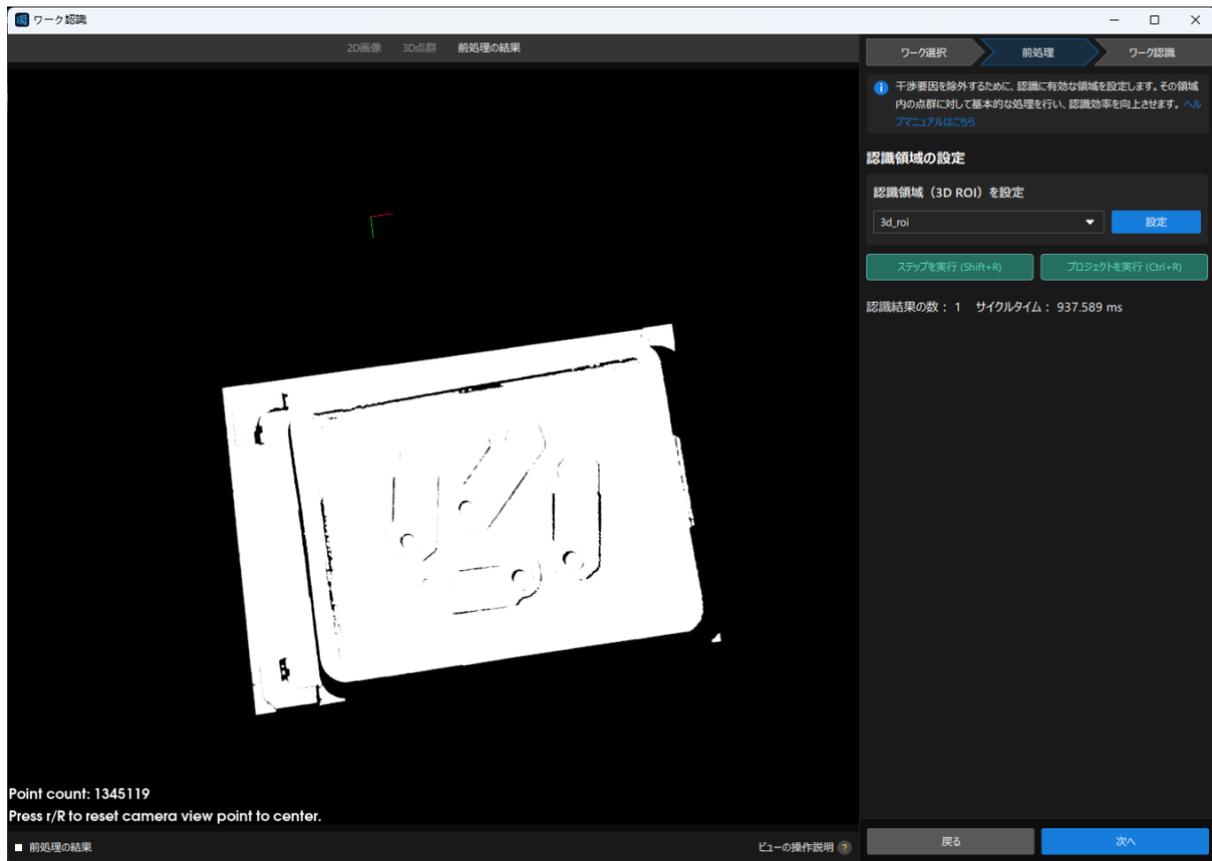
ワークの選択は完了しましたので、画面下部にある [次へ] をクリックして前処理を行います。



前処理

前処理の目的は、認識領域を設定することで不要な点群を除去し、ワークの点群のみを残し、プロジェクトの実行速度を向上させることです。

前処理の画面を下図に示します。

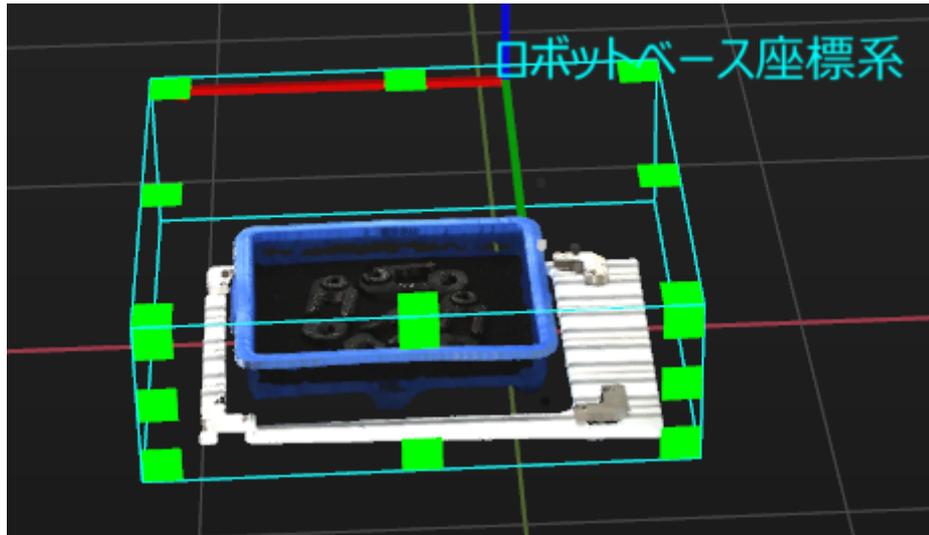


1. 認識領域を設定します。

[設定] をクリックします。



3D関心領域を設定します。**Ctrl** キーを押しながらマウスの左ボタンで3D ROI選択フレームの頂点を長押ししてドラッグし、それを適切なサイズに変更します。変更後の効果を下図に示します。

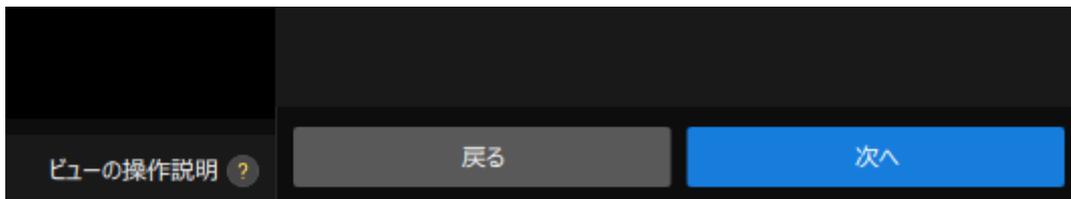


2. 認識領域の設定を保存します。

[保存して適用] をクリックして設定を保存します。



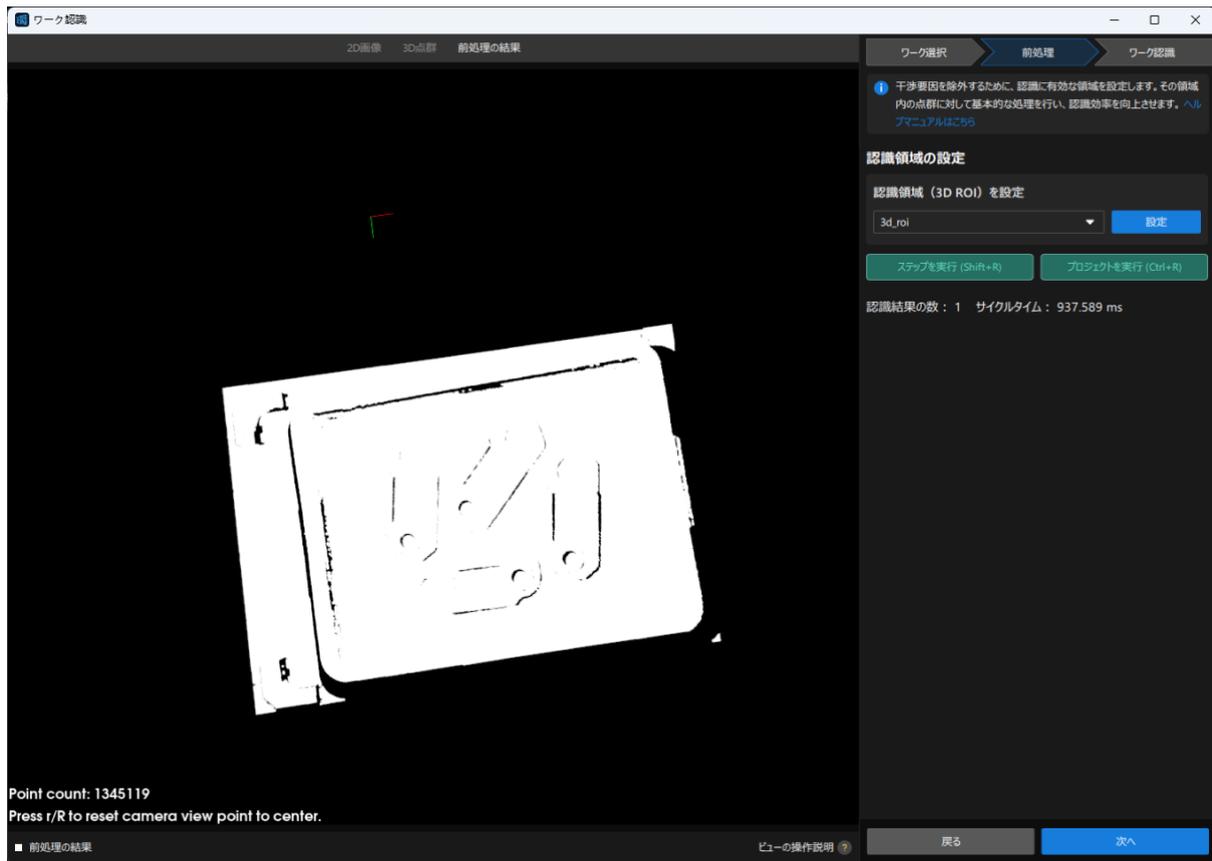
前処理は完了しましたので、画面下部にある [次へ] をクリックしてワーク認識を行います。



ワーク認識

可視化方式で3Dマッチングの関連パラメータを調整して、ワークの位置姿勢を出力できます。

認識画面を下図に示します。

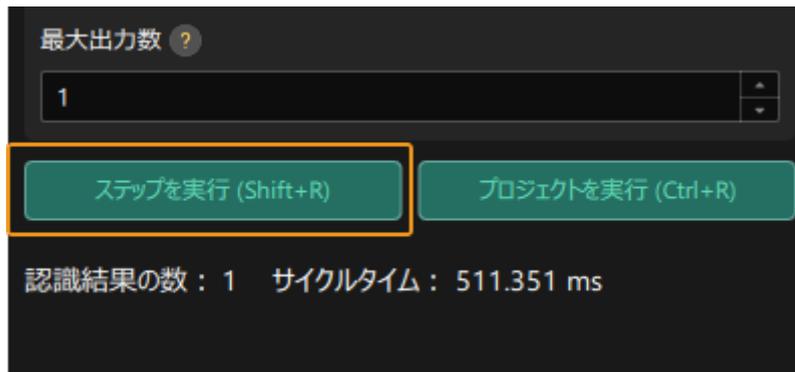


1. ロボットがビジョン結果に基づいてワークの把持と配置を実行する必要があるため、このプロジェクトでは、**最大出力数** を1に設定します。

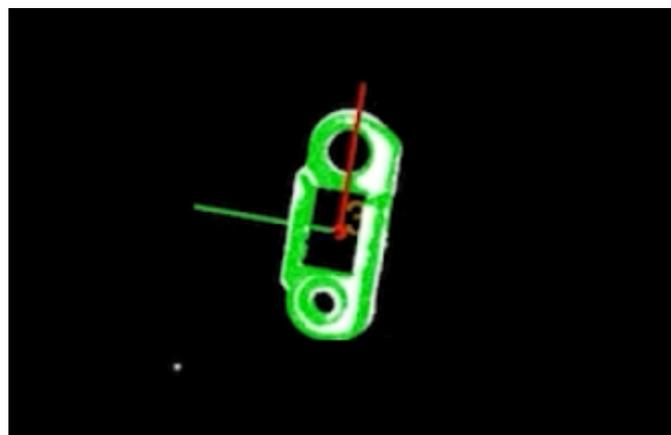


2. 可視化出力結果を確認します。

[**ステップを実行**] をクリックします。



すると、可視化出力結果が可視化エリアに表示されます。下図に示すように、1つのワークの位置姿勢が出力されます。

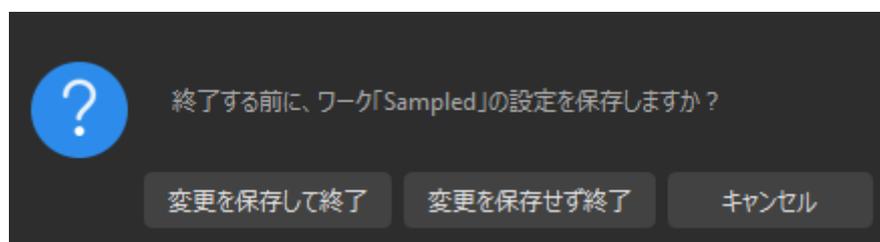


3. 設定を保存します。

可視化設定ツールの画面下部にある [完了] をクリックします。



表示される画面で [変更を保存して終了] をクリックします。



これで、ワークの認識と把持位置姿勢の計算が完了しました。

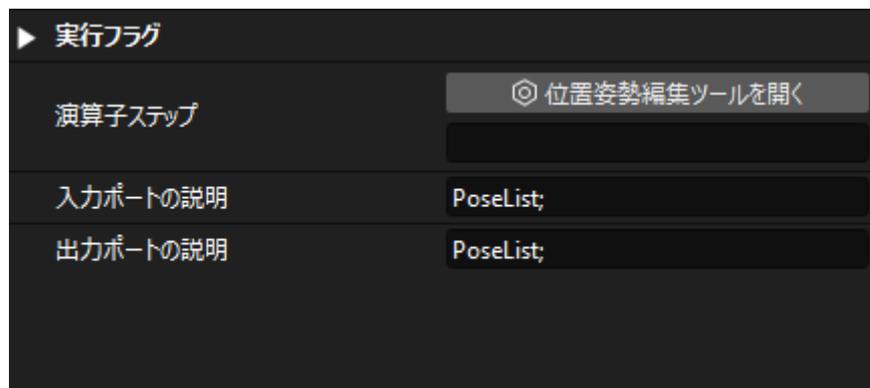
位置姿勢を一括調整

「ワーク認識」ステップによって出力された把持位置姿勢はカメラ座標系にあり、ロボットが

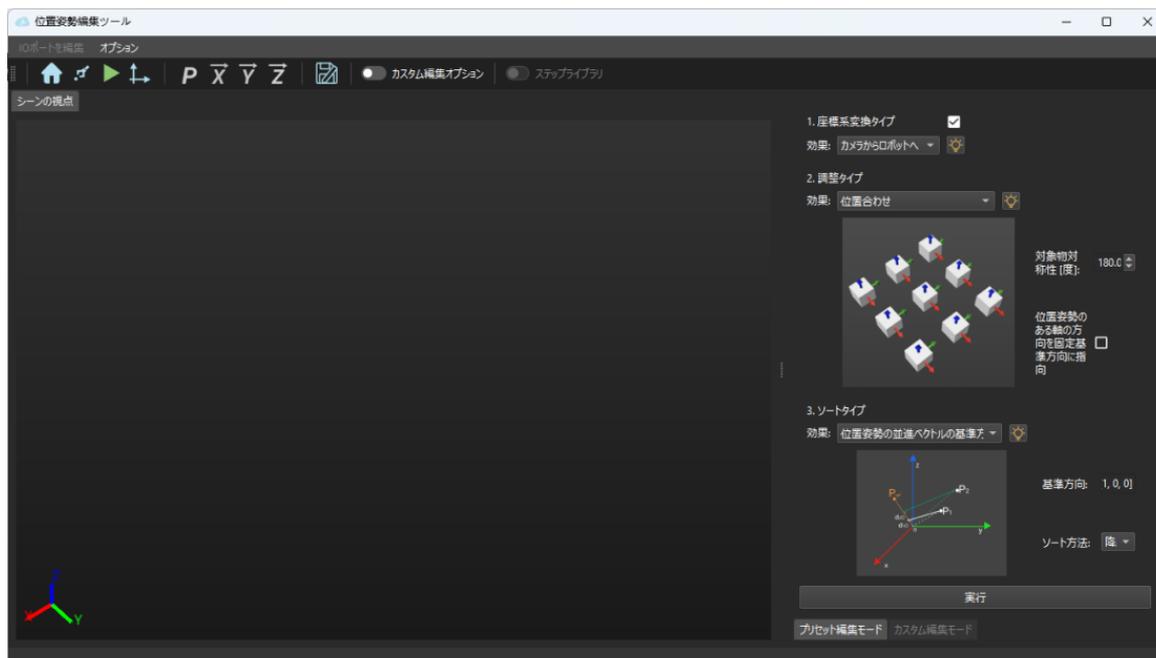
把持しやすいようにワークの位置姿勢を調整する必要があります。位置姿勢をカメラ座標系からロボット座標系に変換します。

1. 位置姿勢編集ツールを開きます。

「位置姿勢を一括調整」ステップをクリックして選択し、ステップパラメータで [位置姿勢編集ツールを開く] をクリックします。

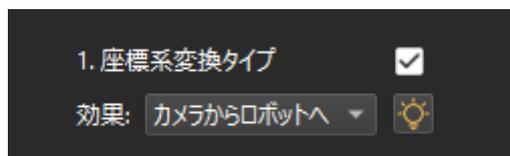


位置姿勢編集ツールの画面を下図に示します。



2. 座標系の変換タイプを調整します。

座標系変換タイプを **カメラからロボットへ** に設定します。



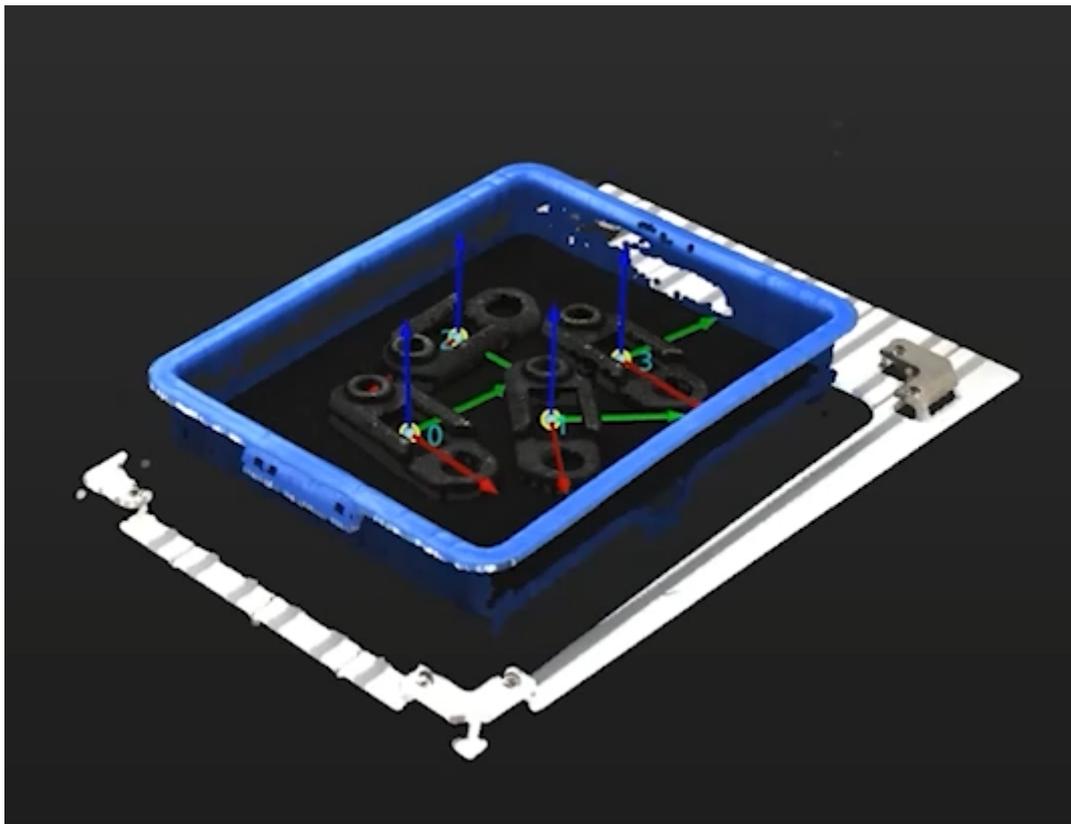
3.

座標系の変換効果を確認します。

画面下部にある **[実行]** をクリックします。

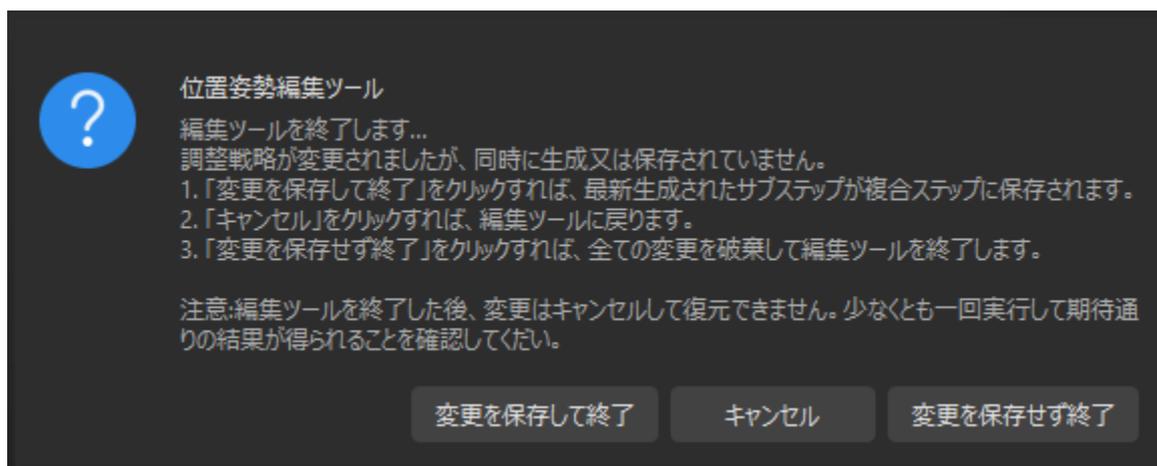


すると、画面中央のシーンの視点から変換後の把持位置姿勢を確認できます。



4. 設定を保存します。

位置姿勢編集ツールの終了ボタンをクリックし、表示される画面で **[変更を保存して終了]** をクリックします。



これで、把持位置姿勢の座標系変換は完了しました。

出力

「出力」ステップを使用して、現在のプロジェクトの結果をバックエンドサービスに送信します。

これで、Mech-Visionプロジェクトの設定は完了しました。

3.5. 把持と配置を実行

本ガイドでは、簡単な把持・配置ロジックを構築するためのロボットプログラムの書き方を習得できます。

本ガイドでは、ロボットアプリケーションは以下の機能を実装しています。

- プログラムを実行するたびに、Mech-Visionプロジェクトは1回実行させます。
- Mech-Visionプロジェクトを1回実行すると、1つのビジョンポイント（TCP位置姿勢）が返されます。
- ロボットは、返されたビジョン結果と事前定義済みの把持・配置経路に基づいて、ワークの把持と配置を完了します。

▶ <https://www.youtube.com/watch?v=ZzYij6FueVg/PLVcMd7cW2rXVtrAejMyVQni2dUDv8bxje>
(YouTube video)

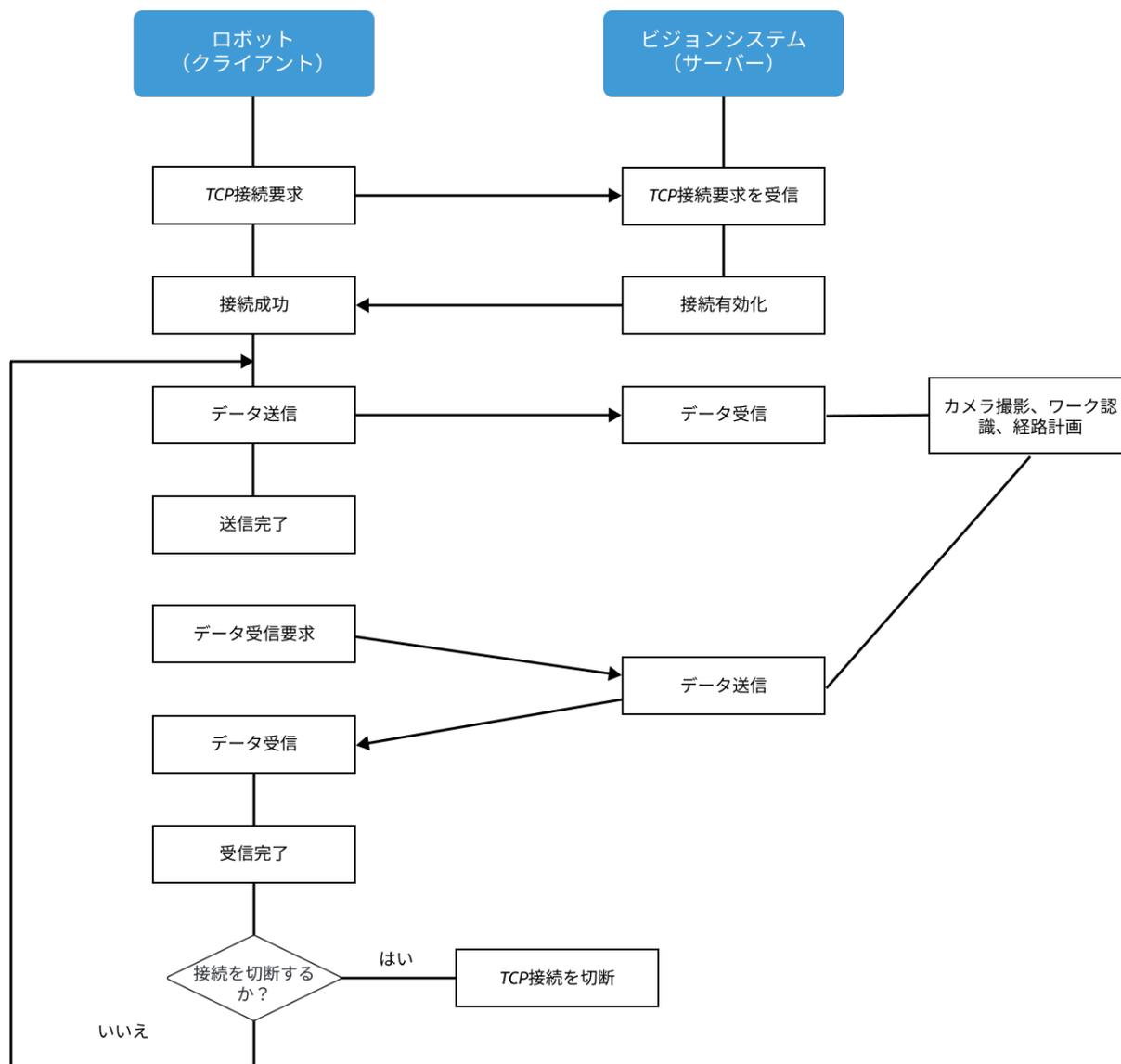
動画：プログラムのデバッグと実行

通信概要

以下では、YaskawaロボットとMech-Mindビジョンシステム間の通信の基本原則について説明します。

YaskawaロボットとMech-Mindビジョンシステムは、TCP Socketプロトコル（データ形式はASCII文字列）に基づく標準インターフェースを使用して相互に通信します。その中、ロボッ

トはTCPクライアントとし、ビジョンシステムはTCPサーバーとします。ロボットとビジョンシステム間の通信フローを下図に示します。



プログラミング説明

以下では、Yaskawaロボットの把持・配置アプリケーションを作成するための基礎知識について説明します。

把持・配置アプリケーションを作成する目的は、ロボットとビジョンシステム間の標準インターフェース通信を行うことと、ビジョンシステムによりロボットの把持・配置作業を可能にすることです。このアプリケーションには、TCP接続の確立/切断、データ転送、ロボット動作とDO制御など、様々な機能が含まれています。把持・配置アプリケーションを作成するためのコマンドには、Yaskawaロボットのコマンドと標準インターフェースコマンドの2種類があります。

Yaskawaロボットのコマンド

Yaskawaロボットのコマンドとは、YaskawaロボットのINFORMロボット言語でサポートされているコマンドで、入出力コマンド、制御コマンド、演算コマンド、移動コマンド、シフトコマンド、およびその他の追加コマンドなどが含まれます。簡単な把持・配置アプリケーションを作成するためのコマンド説明については、[Yaskawaロボットのよく使うコマンド説明](#)をご参照ください。

標準インターフェースコマンド

標準インターフェースコマンドとは、標準インターフェースを使用したYaskawaロボットでサポートされているコマンドのことで、ビジョンシステムとの標準インターフェース通信に使用されます。これにはTCP Socket通信やデータ交換などの機能に関するコマンドが含まれます。Yaskawaロボットがサポートしている標準インターフェースコマンドの詳細については、[Yaskawaロボットの標準インターフェースコマンド説明](#)をご参照ください。

Yaskawaロボットがサポートしている標準インターフェースコマンドは、ロボットに読み込まれたフォアグラウンドプログラム（Job）に組み込まれています。[フォアグラウンドプログラムがロボットに読み込まれた後](#)、Yaskawaロボットのコマンド **CALL** を使用してフォアグラウンドプログラムを呼び出すことで、標準インターフェースコマンドの呼び出しを実現します。例えば、**CALL JOB:MM_OPEN_SOCKET** のようなコマンドを使用します。

把持・配置アプリケーションの作成方法

把持・配置アプリケーションを作成する最も簡単な方法は、既存のサンプルプログラムを基に修正し、デバッグすることです。

Mech-Mindビジョンシステムと併用可能な把持・配置アプリケーションを迅速に作成するために、Mech-Mindは3つのサンプルプログラムを提供しています。本ガイドでは、最初のサンプルプログラムを基に修正とデバッグを行います。このサンプルは簡単な把持と配置を実現しています。一度にビジョンシステム（Mech-Visionのみ）から1つのみのビジョンポイントを取得し、把持と配置を行います。

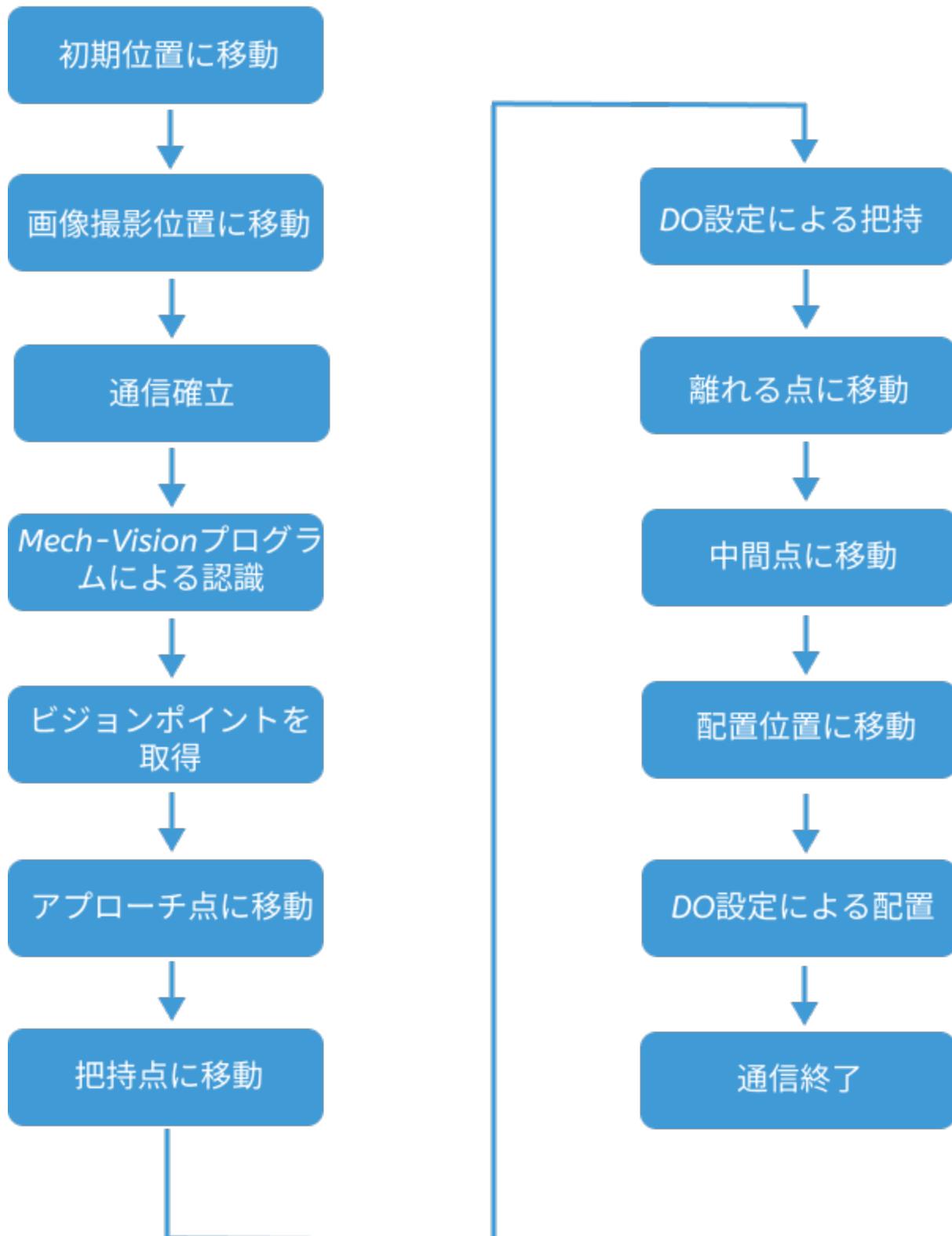
サンプルコードは以下の通りです。

```
NOP
*****
'FUNCTION:simple pick and place
'with Mech-Vision
'mechmind, 2022-5-27
*****
'clear I50 to I69
CLEAR I050 20
'Initialize p variables
SUB P070 P070
SUB P071 P071
```

```

'set 200mm to Z
SETE P070 (3) 200000
'move to the home position
MOVJ VJ=80.00
'move to the camera position
MOVJ VJ=80.00 PL=0
'set ip address of IPC
CALL JOB:MM_INIT_SOCKET ("192.168.170.22;50000;1")
TIMER T=0.20
CALL JOB:MM_OPEN_SOCKET
'set vision recipe
CALL JOB:MM_SET_MODEL ("1;1")
TIMER T=0.20
'Run vision project
CALL JOB:MM_START_VIS ("1;0;2")
TIMER T=1.00
'get result from Vis
CALL JOB:MM_GET_VISDATA ("1;50;51;52")
TIMER T=0.20
PAUSE IF I052<=>1100
'set the first pos to P071;
'set labels to I61;
'set speed to I62;
CALL JOB:MM_GET_POSE ("1;71;61;62")
SFTON P070
MOVJ P071 VJ=50.00
SFTOF
MOVL P071 V=80.0 PL=0
'enable gripper
DOUT OT#(1) ON
SFTON P070
MOVJ P071 VJ=50.00
SFTOF
MOVJ VJ=80.00
'drop point
MOVJ VJ=80.00 PL=0
'release gripper
DOUT OT#(1) OFF
CALL JOB:MM_CLOSE_SOCKET
END
  
```

上記のサンプルプログラムによって作成された把持・配置ロジックを下図に示します。



下表は、把持・配置ロジックの各ステップと修正が必要な部分を示します。

ステップ	コード	詳細
変数初期化	<pre>NOP 'clear I50 to I69 CLEAR I050 20 'Initialize p variables SUB P070 P070 SUB P071 P071 'set 200mm to Z SETE P070 (3) 200000</pre>	<p>プログラムに使用される整数型変数と位置型変数を初期化します。通常の場合、対応するコードの修正は必要ありません。</p>
初期位置に移動	<pre>'move to the home position MOVJ VJ=80.00</pre>	<p>ロボットTCPを設定された初期位置に移動させます。初期位置はロボット動作の開始位置だけでなく、安全な位置でもあります。この位置で、ロボットが把持する対象物や周囲の機器から離れ、カメラの視野を遮らないようにする必要があります。</p> <p>このステップでは、初期位置の設定が必要です。</p>
画像撮影位置に移動	<pre>'move to the camera position MOVJ VJ=80.00 PL=0</pre>	<p>ロボットTCPを設定された画像撮影位置に移動させます。画像撮影位置とは、カメラが画像を撮影したときにロボットがいる位置を指します。この位置では、ロボットハンドがカメラ視野を遮らないようにする必要があります。</p> <p>このステップでは、画像撮影位置を設定する必要があります。</p>
通信確立	<pre>'set ip address of IPC CALL JOB:MM_INIT_SOCKET ("192.168.170.22;500 00;1") TIMER T=0.20 CALL JOB:MM_OPEN_SOCKET</pre>	<p>通信パラメータを初期化し、ビジョンシステムとのTCP Socket通信を確立します。</p> <p>このステップでは、「通信初期化」コマンド (MM_INIT_SOCKET) のパラメータ (IPCのIPアドレスとポート) を調整する必要があります。</p>

ステップ	コード	詳細
Mech-Visionプロジェクトによるビジョン認識	<pre>'set vision recipe CALL JOB:MM_SET_MODEL ("1;1") 'Run vision project CALL JOB:MM_START_VIS ("1;0;2")</pre>	<p>まずMech-Visionパラメータレシピ（複数種類のワークの場合）を切り替え、Mech-Visionプロジェクトの実行をトリガーしてワークの把持位置姿勢を取得します。本ガイドでは、1種類のワークのみを処理するため、パラメータレシピの切り替えは必要ありません。</p> <p>このステップでは、「Mech-Visionパラメータレシピの切り替え」コマンド（MM_SET_MODEL）をコメントアウトし、「Mech-Visionプロジェクトを実行」コマンドのパラメータを調整する必要があります。</p>
ビジョンポイントを取得	<pre>'get result from Vis CALL JOB:MM_GET_VISDATA ("1;50;51;52") TIMER T=0.20 PAUSE IF I052<=1100 'set the first pos to P071; 'set labels to I61; 'set speed to I62; CALL JOB:MM_GET_POSE ("1;71;61;62")</pre>	<p>Mech-Visionからビジョンポイントを取得し、ビジョンポイントの位置姿勢、位置姿勢が対応するラベルと速度値を変数に保存します。</p> <p>本ガイドでは、ロボットは一度にMech-Visionから1つのビジョンポイントを取得します。サンプルコードは、変更せずにそのまま使用できます。</p>
アプローチ点に移動	<pre>SFTON P070 MOVJ P071 VJ=50.00 SFTOF</pre>	<p>ロボットTCPをアプローチ点に移動させ、画像撮影位置から把持点までの移動中に、ロボットがシーンの物体（コンテナなど）と衝突するのを防ぎます。</p> <p>サンプルプログラムでは、把持点（P071）より200mm上方（アプローチ点）にロボットTCPを移動させます。アプローチ点から把持点の距離が小さすぎる場合、変数P070のZ軸値を変更することができます。</p> <pre>'set 200mm to Z SETE P070 (3) 200000</pre>

ステップ	コード	詳細
把持点に移動	<pre>MOVL P071 V=80.0 PL=0</pre>	<p>ロボットTCPを把持点（P071）に移動させます。</p> <p>サンプルプログラムでは、ロボットTCPはアプローチ点から把持点まで直線移動させます。サンプルコードは、変更せずにそのまま使用できます。</p>
DO設定による把持	<pre>'enable gripper DOUT OT#(1) ON</pre>	<p>ロボットハンドを開いてワークを把持します。</p> <p>本ガイドで使用されるロボットハンドは、汎用出力信号2がオンになると開いた状態に保持されます。したがって、ここでは汎用出力信号を2に設定する必要があります。汎用出力信号は実際の状況に応じて設定する必要があります。</p>
離れる点に移動	<pre>SFTON P070 MOVJ P071 VJ=50.00 SFTOF</pre>	<p>ロボットTCPを離れる点に移動させ、ワーク把持後にロボットがシーンの物体（コンテナなど）と衝突するのを防ぎます。</p> <p>サンプルプログラムでは、把持点（P071）より200mm上方（離れる点）にロボットTCPを移動させます。離れる点から把持点の距離が小さすぎる場合、変数P070のZ軸値を変更することができます。</p> <pre>'set 200mm to Z SETE P070 (3) 200000</pre> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> 把持位置姿勢のZ軸のオフセットとして、アプローチ点と離れる点の両方に共通の変数P070のZ軸値を使用します。もし変数P070のZ軸値を変更する場合、その2つの点に影響が及びます。</p> </div>

ステップ	コード	詳細
中間点に移動	<pre>MOVJ VJ=80.00 PL=0</pre>	<p>ロボットTCPを設定された中間点に移動させ、ロボットがシーンの物体と衝突するのを防ぎます。</p> <p>このステップでは、中間点の設定が必要です。実際の状況に応じて中間点をさらに追加することができます。</p>
配置位置に移動	<pre>'drop point MOVJ VJ=80.00 PL=0</pre>	<p>ロボットTCPを設定された配置位置に移動させます。</p> <p>このステップでは、配置位置の設定が必要です。</p>
DO設定による配置	<pre>'enable gripper DOUT OT#(1) OFF</pre>	<p>ロボットハンドを閉じてワークを配置します。</p> <p>本ガイドで使用されているロボットハンドは、汎用出力信号1がオンになると閉じられ、ワークの配置を行います。したがって、ここでは汎用出力信号1のステータスをONに設定する必要があります。汎用出力信号の番号とステータスは、実際の状況に応じて設定する必要があります。</p>
通信終了	<pre>CALL JOB:MM_CLOSE_SOCKET END</pre>	<p>TCP Socket通信を切断します。サンプルコードは、変更せずにそのまま使用できます。</p>



サンプルプログラムの詳細については、[サンプルプログラムの解説](#)をご参照ください。

サンプルプログラムの修正とデバッグ



このサンプルプログラムで使用されているロボット位置姿勢は、TCP位置姿勢となります。サンプルプログラムの修正とデバッグを行う前に、ツールID（01など）を把持のためのロボットハンドの番号に切り替え、このツールのTCP設定を完了する必要があります。

サンプルプログラムを開く

把持・配置のサンプルプログラムの修正とデバッグを行うには、まずはサンプルプログラムを開きます。

1. **ONLINE MODE** では、**Main Menu** から **JOB > SELECT JOB** を選択し、インストールされて

いるプログラム一覧が表示されます。

2. **mm_sample** プログラムをクリックして選択し、**SELECT** を押して開きます。



サンプルプログラム冒頭のコードは変数の初期化に使用され、通常は変更する必要はありません。

初期位置に移動

ロボットTCPを設定された初期位置に移動させます。ここではロボットの初期位置を設定する必要があります。

下記のコードの修正が必要です。

```
'move to the home position
MOVJ VJ=80.00
```

ロボットの初期位置を設定するには、以下の操作を実行します。

1. ティーチペンダントのキーを **TEACH** に合わせると **ティーチモード** になります。
2. 手動でロボットをプログラムによって設定された初期位置に移動させます。
3. ティーチペンダントの ▼ キーを押して「MOVJ VJ=80.00」行を選択し、**MODIFY** キーを押します。その後、**ENTER** キーを押し、初期位置の位置姿勢をロボットの現在のTCP位置姿勢に変更します。



「VJ=80.00」は関節運動の速度が80%であることを示します。その他のパラメータの調整は必要ありません。

画像撮影位置に移動

ロボットTCPを設定された画像撮影位置に移動させます。ここでは、ロボットの画像撮影位置を設定する必要があります。

下記のコードの修正が必要です。

```
'move to the camera position
MOVJ VJ=80.00 PL=0
```

画像撮影位置を設定するには、以下の操作を実行します。

1. **ティーチモード** では、手動でロボットを画像撮影位置に移動させます。
2. ティーチペンダントの ▼ キーを押して「MOVJ VJ=80.00 PL=0」行を選択し、**MODIFY** を押します。その後、**ENTER** キーを押し、画像撮影位置をロボットの現在のTCP位置姿勢に変更します。



「VJ=80.00」は関節運動の速度が80%で、「PL=0」は位置レベルが0であることを示

します。この2つのパラメータを調整する必要はありません。位置レベルは、ロボットをティーチ位置に設定した際のその位置への接近度を調整するために使用されます。値の範囲は0から8で、値が小さいほど位置はより近くなります。

通信確立

ビジョンシステムとのTCP接続を確立します。このステップでは、「通信初期化」コマンド (MM_INIT_SOCKET) のパラメータ (IPCのIPアドレスとポート) を調整する必要があります。

以下のサンプルコードの修正が必要です。

```
'set ip address of IPC
CALL JOB:MM_INIT_SOCKET ("192.168.170.22;50000;1")
TIMER T=0.20
CALL JOB:MM_OPEN_SOCKET
```

「通信初期化」コマンドのパラメータを設定するには、以下の手順を実行します。

1. **ティーチモード**では、ティーチペンダントの ▼ キーを押して「CALL JOB:MM_INIT_SOCKET ("192.168.170.22;50000;1")」行を選択し、**SELECT**キーを押します。
2. ティーチペンダントパネルのソフトキーボードを使用して「192.168.170.22」と「50000」をそれぞれIPCのIPアドレスとポートに変更し、パネルの**OK**を押して変更を保存します。

Mech-Visionプロジェクトによるビジョン認識

Mech-Visionプロジェクトを実行してビジョン処理による認識を行い、ビジョン結果を出力します。

ここでは、「Mech-Visionプロジェクトを実行」コマンドのパラメータを変更する必要があります。本ガイドは1種類のワークのみを処理するため、「Mech-Visionパラメータレシピの切り替え」コマンドをコメントアウトする必要もあります。

以下のサンプルコードの修正が必要です。

```
CALL JOB:MM_SET_MODEL ("1;1")
'Run vision project
CALL JOB:MM_START_VIS ("1;0;2")
```

「Mech-Visionパラメータレシピの切り替え」コマンドをコメントアウトし、「Mech-Visionプロジェクトを実行」コマンドのパラメータを変更するには、以下の手順を実行します。

1. **ティーチモード**では、ティーチペンダントの ▼ キーを押して「CALL JOB:MM_SET_MODEL ("1;1")」行を選択し、**EDIT**タブをクリックします。その後、[**COMMENT OUT**]を選択してティーチペンダントの**SELECT**キーを押します。



複数種類のワークが含まれるプロジェクトでは、パラメータレシピを使用する必要があります。この場合、2番目のパラメータを現在のワークに対応するパラメータレシピの番号に変更する必要があります。

2. ティーチペンダントの ▼ キーを押して「CALL JOB:MM_START_VIS ("1;0;2")」行を選択し、**SELECT**キーを押します。
3. ティーチペンダントパネルのソフトキーボードを使用して「MM_START_VIS」コマンドのパラメータ（例えば、1;1;0）を変更し、パネルの**OK**をクリックして変更を保存します。
 - 1：Mech-Visionプロジェクト番号です。本ガイドで使用されているプロジェクトの番号が1であるため、変更する必要はありません。
 - 0：このサンプルプログラムでは、一度にMech-Visionから1つのビジョンポイントを取得するため、この値を1に変更する必要があります。
 - 2：このサンプルプログラムでは、ロボットの位置姿勢をMech-Visionに送信する必要がないため、この値を0に変更する必要があります。

ビジョンポイントを取得

Mech-Visionからビジョン結果を取得し、ビジョン結果の取得が成功したかどうかを判断します。取得に成功すると、ビジョンポイントの位置姿勢は位置型変数P071に格納され、この位置姿勢が対応するラベルデータは整数型変数I061に格納され、この位置姿勢が対応する移動速度が整数型変数I062に格納されます。

下記のコードは、修正せずにそのまま使用できます。

```
'get result from Vis
CALL JOB:MM_GET_VISDATA ("1;50;51;52")
PAUSE IF I052<=1100
'set the first pos to P071;
'set labels to I61;
'set speed to I62;
CALL JOB:MM_GET_POSE ("1;71;61;62")
```

アプローチ点に移動

ロボットTCPをアプローチ点に移動させます。



- 把持点に到達する過程で、ロボットハンドとコンテナや他のワークとの衝突を避けるため、ロボットTCPは通常、把持点の真上のある距離（アプローチ点と呼ばれる）まで移動させ、その位置から把持点まで直線運動させます。
- 把持位置姿勢のZ軸の値をオフセットし、アプローチ点の位置姿勢を取得できます。

下記のコードは、修正せずにそのまま使用できます。本ガイドでは、Z軸のオフセット値を200mmに設定します。実際の状況に応じて設定する必要があります。

```
'set 200mm to Z
SETE P070 (3) 200000
...
SFTON P070
MOVJ P071 VJ=50.00
SFTOF
```

把持点に移動

ロボットTCPはアプローチ点から把持点まで直線移動させます。

下記のコードは、修正せずにそのまま使用できます。

```
MOVL P071 V=80.0 PL=0
```

DO設定による把持



本ガイドでは、開閉式のロボットハンドが使用されており、出力信号による制御します。出力信号2をオンにするとロボットハンドが開き、出力信号1をオンにするとロボットハンドが閉じます。

出力信号2をオンにすることでロボットハンドを開き、ワークの把持を実現します。したがって、出力信号の番号を2に設定する必要があります。

以下のサンプルコードの修正が必要です。

```
'enable gripper
DOUT OT#(1) ON
```

出力信号の番号を2に変更するには、以下の手順を実行します。

1. ティーチモードでは、ティーチペンダントの ▼ キーを押して「DOUT OT#(1) ON」行を選択し、SELECTキーを押します。
2. 出力信号「1」を「2」に変更し、パネルのOKをクリックして確定します。

離れる点に移動

ロボットTCPは把持点から、離れる点まで直線移動させます。



- ロボットハンドが把持後に離れる際、コンテナや他のワークとの衝突を避けるため、ロボットTCPを把持点の真上の一定距離（離れる点と呼ばれる）まで直線移動させ、その位置から離れて配置するのが一般的です。
- 把持位置姿勢のZ軸の値をオフセットし、離れる点の位置姿勢を取得できます。
- 通常、アプローチ点と離れる点は同じ場所にあります。このサンプルでは、アプロ

一子点と離れる点は、ともに把持点の真上200mmの位置にあります。

下記のコードは、修正せずにそのまま使用できます。

```
'set 200mm to Z
SETE P070 (3) 200000
...

SFTON P070
MOVJ P071 VJ=50.00
SFTOF
```

中間点に移動

ロボットTCPを設定された配置位置に移動させます。



ロボットとシーンの衝突を避けるため、離れる点と配置位置の間に中間点を追加します。ここでは、1つの中間点のみを追加します。実際の応用シーンでは、配置経路を最適化するためにさらに中間点を追加することができます。

下記のコードの修正が必要です。

```
MOVJ VJ=80.00
```

中間点を設定するには、以下の手順を実行します。

1. **ティーチモード**では、手動でロボットを適切な中間点に移動させます。
2. ティーチペンダントの ▼ キーを押して「MOVJ VJ=80.00」行を選択し、**MODIFY**を押します。その後、**ENTER**キーを押し、中間点をロボットの現在のTCP位置姿勢に変更します。

配置位置に移動

ロボットTCPを設定された配置位置に移動させます。

下記のコードの修正が必要です。

```
'drop point
MOVJ VJ=80.00 PL=0
```

ロボットの配置位置を設定するには、以下の手順を実行します。

1. **ティーチモード**では、手動でロボットを適切な配置位置に移動させます。
2. ティーチペンダントの ▼ キーを押して「MOVJ VJ=80.00 PL=0」行を選択し、**MODIFY**を押します。その後、**ENTER**キーを押し、配置位置をロボットの現在位置姿勢に変更します。

DO設定による配置

出力信号1をオンにすることでロボットハンドを閉じ、ワークの配置を実現します。

対応するコートは以下の通りです。出力信号1をオンにするために、「OFF」を「ON」に変更する必要があります。

```
'release gripper
DOUT OT#(1) OFF
```

通信終了

このサンプルでは、一回の把持と配置のみを行います。したがって、一回の把持・配置が完了した後、「MM_CLOSE_SOCKET」コマンドを呼び出して通信接続を切断する必要があります。

```
CALL JOB:MM_CLOSE_SOCKET
```

これで、このサンプルプログラムの修正が完了しました。次にプログラムの実行をテストします。

プログラムの実行をテスト

1. ティーチペンダントのキーを **PLAY** に合わせると、**再生モード**になります。
2. ティーチペンダントの **SERVO ON READY** を押し、青い **[RUN]** ボタンを押します。

ロボットは把持・配置アプリケーションを自動的に実行し、ワークの把持と配置を完了します。



ロボットの動作経路にご注意ください。緊急時には緊急停止ボタンを押してください。

これで、簡単な把持・配置アプリケーションの作成が完了しました。

参考情報

Yaskawaロボットのよく使うコマンド

より詳しい情報は、Yaskawaロボットの《YRC1000/YRC1000micro/DX200 INFORM 取扱説明書》をご参照ください。

コマンド	詳細	プログラミング文法	例
NOP	プログラムの開始コマンドであり、何の操作も実行しません。	NOP	NOP

コマンド	詳細	プログラミング文法	例
,	コードのコメントアウトを実行します。	'コメントアウトする文字列	'Run vision project
END	プログラムは終了します。	END	END
CLEAR	データ2で指定された数に従って、データ1で指定された番号以降の変数の内容をゼロにします。	CLEAR <データ1> <データ2>	CLEAR I050 20 変数I50からI69（合計20個）をゼロにします。
SUB	データ1からデータ2を引き、結果をデータ1に保存します。	SUB <データ1> <データ2>	SUB P070 P070 位置型変数P070から位置型変数P070を引き、結果を変数P070に保存します。
SETE	データ2をデータ1に設定する位置型変数の要素となります。	SETE <データ1> <データ2>	SETE P070 (3) 200000 変数P070のZ座標値に、オフセット200000（ミクロン単位、200mmに相当）を増加させます。
MOVJ	関節補間（関節運動方式）によりティーチ位置に移動します。対象位置に移動する際、ロボットの経路を制限しない場合に使用されます。	MOVJ ロボットティーチ位置のファイル番号 VJ=関節運動の速度(%)	MOVJ VJ=80.00 ロボットは関節運動でティーチ位置に移動します。
MOVL	直線補間（直線運動方式）によりティーチ位置に移動します。直線経路に従って、ティーチング後の位置に移動します。	MOVL ロボットティーチ位置のファイル番号 V=TCP速度（mm/秒） PL=位置ラベル（0~8）	MOVL P071 V=80.0 PL=0 ロボットは直線運動でP071によって指定された位置に移動します。
CALL	指定したプログラムを呼び出します。	CALL JOB：呼び出されるプログラム	CALL JOB:MM_OPEN_SOCKET 標準インターフェースコマンド「MM_OPEN_SOCKET」を呼び出してTCP接続を構築します。
TIMER	指定した時間まで待機します。	TIMER T=時間（秒単位）	TIMER T=0.20 ロボットは0.2秒待機します。

コマンド	詳細	プログラミング文法	例
PAUSE	プログラムの実行を一時停止します。	PAUSE IF	PAUSE IF I052<>1100 変数I052の値が1100でない場合、プログラムの実行は一時停止されます。
IF	アクション（動作）を実行する際に、ある条件が成立しているかどうかを判断して、適切な処理を行います。これは、他の処理コマンドの後に追加して使用されます。	IF <対象1>演算子 (=、<>など) <対象1>	PAUSE IF I052<>1100 変数I052の値が1100でない場合、プログラムの実行は一時停止されます。
SFTON	並進移動を開始します。並進移動の量は、座標系内のX、Y、Zの増分値であり、位置型変数に設定されます。	SFTON 位置型変数の番号	<pre> SETI P070 (3) 200000 ... SFTON P070 MOVJ P071 VJ=50.00 SFTOF </pre> ロボットが関節運動で把持点（P071）の200mm上（アプローチ点）まで移動します。
SFTOF	並進移動を終了します。通常、SFTONとペアで使用されます。	SFTOF	SFTOF
DOUT	汎用出力信号のオン／オフを切り替えます。	DOUT OT#(出力信号) ON/OFF	DOUT OT#(1) ON 汎用出力信号1がオンになります。 DOUT OT#(1) OFF 汎用出力信号1がオフになります。

サンプルプログラムの解説



```
NOP
```

意味：プログラムの実行を開始します。



```
'clear I50 to I69
CLEAR I050 20
```

意味：整数型変数I50からI69の値をゼロにします。本ガイドでは、I50からI69の変数はステータス、数量、ラベル値を保存するために使用されます。

- ```
'Initialize p variables
SUB P070 P070
SUB P071 P071
```

意味：位置型変数P070とP071の値をゼロにします。本ガイドでは、変数P071は把持位置姿勢を保存するために使用されます。把持点に対するアプローチ点と離れる点のZ軸のオフセットは、変数P070のZ軸値に保存されます。

- ```
'set 200mm to Z
SETE P070 (3) 200000
```

意味：把持点に対するアプローチ点と離れる点のZ軸のオフセットをP070変数のZ軸値として保存します。単位はミクロンです。本ガイドでは、オフセットの初期値は200000ミクロン、即ち200ミリメートルとなります。

- ```
'move to the home position
MOVJ VJ=80.00
```

意味：ロボットTCPを設定された初期位置に移動させます。「VJ=80.00」は関節運動の速度が80%であることを示します。ここでは初期位置の設定が必要です。

- ```
'move to the camera position
MOVJ VJ=80.00 PL=0
```

意味：ロボットTCPを設定された画像撮影位置に移動させます。「VJ=80.00」は関節運動の速度が80%で、「PL=0」は位置レベルが0であることを示します。位置レベルは、ロボットをティーチ位置に設定した際のその位置への接近度を調整するために使用されます。値の範囲は0から8で、値が小さいほど位置はより近くなります。

- ```
'set ip address of IPC
CALL JOB:MM_INIT_SOCKET ("192.168.170.22;50000;1")
```

意味：通信パラメータ（IPCのIPアドレスとポート）を初期化します。

- ```
TIMER T=0.20
```

意味：ロボットは0.2秒待機します。

- ```
CALL JOB:MM_OPEN_SOCKET
```

意味：TCP Socket通信接続を確立します。

- ```
'set vision recipe  
CALL JOB:MM_SET_MODEL ("1;1")
```

意味：プロジェクトのパラメータレシピを切り替えます。プロジェクトは複数種類のワークを処理する場合、パラメータレシピの使用が必要になります。

- ```
'Run vision project
CALL JOB:MM_START_VIS ("1;0;2")
```

意味：Mech-Visionプロジェクトの実行をトリガーし、ビジョン処理による認識を行います。「MM\_START\_VIS」コマンドのパラメータを正しく設定する必要があります。

- ```
'get result from Vis  
CALL JOB:MM_GET_VISDATA ("1;50;51;52")
```

意味：ビジョンポイントを取得し、ビジョンポイントの送信が完了したかどうかのステータス、返された位置姿勢の数、コマンドのステータスコードをそれぞれ変数I050~I052に保存します。

- ```
PAUSE IF I052<>1100
```

意味：コマンドのステータスコード（I052変数値）が1100（正常実行完了のコード）でない場合、プログラムの実行は中止されます。

- ```
'set the first pos to P071;  
'set labels to I61;  
'set speed to I62;  
CALL JOB:MM_GET_POSE ("1;71;61;62")
```

意味：取得されたビジョン結果から、最初の位置姿勢とそれに対応するラベルおよび移動速度データをP071、I061、I062に保存します。後続の呼び出しに使用されます。

- ```
SFTON P070
MOVJ P071 VJ=50.00
SFTOF
```

意味：並進移動コマンドにより、ロボットTCPを関節運動で画像撮影位置からアプローチ点に移動させます。把持点に対するアプローチ点のZ軸のオフセットは、変数P070のZ軸値によって決まります。

```
● MOVL P071 V=80.0 PL=0
```

意味：ロボットTCPは直線運動で把持点（変数P071によって指定された位置）に移動させます。TCP移動速度は80mm/sであり、位置ラベルは0です。

```
● 'enable gripper
 DOUT OT#(1) ON
```

意味：出力信号1をオンにすることで、ロボットハンドを制御してワークを把持します。汎用出力信号は実際の状況に応じて設定する必要があります。

```
● SFTON P070
 MOVJ P071 VJ=50.00
 SFTOF
```

意味：並進移動コマンドにより、ロボットTCPは直線運動で離れる点に移動させます。把持点に対する離れる点のZ軸のオフセットは、変数P070のZ軸値によって決まります。

```
● MOVJ VJ=80.00
```

ロボットTCPを設定された中間点に移動させ、ロボットがシーンの物体と衝突するのを防ぎます。ここでは中間点を設定する必要があります。

```
● 'drop point
 MOVJ VJ=80.00 PL=0
```

ロボットTCPを設定された中間点に移動させ、ロボットがシーンの物体と衝突するのを防ぎます。ここでは中間点を設定する必要があります。

```
● 'release gripper
 DOUT OT#(1) OFF
```

意味：出力信号1をオフにし、ロボットハンドを制御してワークを配置します。実際の応用シーンでは、実際の状況に応じて出力信号の番号やステータスを変更する必要があります。

```
● CALL JOB:MM_CLOSE_SOCKET
```

意味：TCP Socket通信接続を切断します。



END

意味：プログラムは終了します。

## 4. 入門ガイド：ビジョンシステムによる単載デパレタイズ（Vizティーチング通信）

本ガイドを読むことで、Vizティーチング通信を使用し、3Dビジョンシステムによる段ボール箱のデパレタイズアプリケーションを実装する方法を習得できます。

### 概要

- カメラ：Eye to Hand方式で取り付けられているMech-Eye DEEPカメラ
- ロボット：ABB\_IRB\_1300\_11\_0\_9
- ワーク：1種類（寸法が同じ）の段ボール箱
- 使用されるソフトウェア：Mech-Vision 1.7.2バージョン、Mech-Viz 1.7.2バージョン、Mech-Center 1.7.2バージョン、Mech-Eye Viewer 2.1.0バージョン
- 通信方式：Vizティーチング通信



上記と異なるカメラ型番やロボットブランド、ワークを使用する場合、一部調整を行う必要があります。

▶ <https://www.youtube.com/watch?v=v9QrLGzeLI3Y/PLVcMd7cW2rXWbHhTQX8m2R1r6PhX6vrCl> (YouTube video)

動画：ビジョンシステムによるデパレタイズ

### 用語説明

|          |                           |
|----------|---------------------------|
| パレット     | 荷物（段ボール箱）を載せる台            |
| デパレタイズ   | パレットに積まれた荷物を降ろす作業         |
| パレタイズ    | パレットに荷物を積んでいく作業           |
| 単載デパレタイズ | 1種類の（寸法が同じ）段ボール箱のデパレタイズ   |
| 混載デパレタイズ | 複数種類の（寸法が異なる）段ボール箱のデパレタイズ |



単載デパレタイズ



混載デパレタイズ

## ビジョンアプリケーションの実装

ビジョンアプリケーションの実装は、一般的に5つの段階に分けられています。



詳細は下表のとおりです。

| 番号 | 階段                   | 説明                                                                                         |
|----|----------------------|--------------------------------------------------------------------------------------------|
| 1  | ビジョンシステムのハードウェア設置    | Mech-Mindビジョンシステムのハードウェアの取り付けと接続、ソフトウェアのインストールと接続を完了します。                                   |
| 2  | ロボット通信設定             | Vizとの通信プログラムと設定ファイルをロボットシステムに読み込み、ビジョン側とロボット側との通信を構築し、Mech-Mindソフトウェアシステムによるロボットの制御を実現します。 |
| 3  | ロボットハンド・アイ・キャリブレーション | 自動キャリブレーション（Eye To Hand）を実行し、カメラ座標系とロボット座標系との対応関係を求めます。                                    |
| 4  | 段ボール箱の認識             | ソリューションライブラリの「同じ種類の段ボール箱」プロジェクトを使用し、段ボール箱を認識し、ロボットがそれを正確に把持できるようにガイドします。                   |
| 5  | 把持と配置を実行             | Mech-Vizプロジェクトを作成し、ロボットが計画された衝突のない経路に従って段ボール箱を把持・配置します。                                    |

本節の説明は終わりです。次に、アプリケーションの実装を完了させてください。

## 4.1. ビジョンシステムのハードウェア設置

本ガイドを読むことで、Mech-Mindビジョンシステムの設置方法を習得できます。

Mech-Mindビジョンシステムの設置は、**梱包内容の確認** → **ハードウェアの取り付け** → **ネットワーク接続** → **ソフトウェアのアップグレード（オプション）** → **ビジョンシステムが画像を正常に取得できるか確認**の順で行う必要があります。

▶ <https://www.youtube.com/watch?v=N8iwcoSjEdU/PLVcMd7cW2rXWbHhTQX8m2R1r6PhX6vrCl> (YouTube video)

動画：ビジョンシステムのハードウェア設置

### 梱包内容の確認

1. お受け取りの際は、梱包に問題がないことを確認してください。
2. 梱包中にある『同梱包一覧』を確認し、商品や付属品に欠品や損傷がないことを確認してください。

下図は、梱包箱に含まれるものおよび付属品の例を示しています。下記の同梱品一覧は参照のみを目的としています。梱包中に入っているものは、実際の梱包の中にある『同梱包一覧』をご参照ください。



| 番号 | 種類      | 名前                      | 機能                                           |
|----|---------|-------------------------|----------------------------------------------|
| 1  | IPCと付属品 | IPC                     | Mech-Mindソフトウェアシステム搭載                        |
| 2  |         | IPCの付属品                 | WIFIアンテナなどの付属品                               |
| 3  |         | IPCの電源ケーブルとアダプター        | IPCの電源を入れる                                   |
| 4  | カメラと付属品 | Mech-Eye産業用3Dカメラ        | 画像取得                                         |
| 5  |         | カメラ取扱説明書                | Mech-Eye産業用3Dカメラの取扱説明書                       |
| 6  |         | カメラの付属品                 | カメラの取り付け                                     |
| 7  | その他の付属品 | dongle                  | ソフトウェアのライセンス供与                               |
| 8  |         | キャリブレーションボード            | カメラキャリブレーション                                 |
| 9  |         | フランジ                    | キャリブレーションボードの取り付け                            |
| 10 |         | カメラDC電源ケーブル（標準仕様20メートル） | カメラをレール電源に接続（オプションでより長い電源ケーブルも利用可能）          |
| 11 |         | カメラのLANケーブル（標準仕様20メートル） | カメラをIPCに接続（オプションでより長いLANケーブルも利用可能）           |
| 12 |         | レール電源（標準仕様）             | Mech-Eye産業用3Dカメラの電源を入れます（オプションで電源アダプターも利用可能） |
| 13 | 同梱品一覧   |                         | 梱包に含まれるすべてのものと付属品を一覧表示します                    |



万一、商品紛失・破損等が発生した場合は、Mech-Mind株式会社にお問合せください。

## その他のものを準備

本ガイドでは、カメラ梱包に入っているものの他に、下表に示すようなものを自分で用意する必要があります。

| 名前       | 機能          |
|----------|-------------|
| モニター     | IPCの画面表示    |
| HDMIケーブル | IPCとモニターを接続 |

| 名前           | 機能            |
|--------------|---------------|
| RJ45 LANケーブル | IPCとコントローラを接続 |



本ガイドでは、IPCはRJ45 LANケーブルでコントローラと接続し、カメラのLANケーブルでカメラと接続します。ルータ経由でIPCをロボットコントローラに、IPCをカメラに接続することもできます。

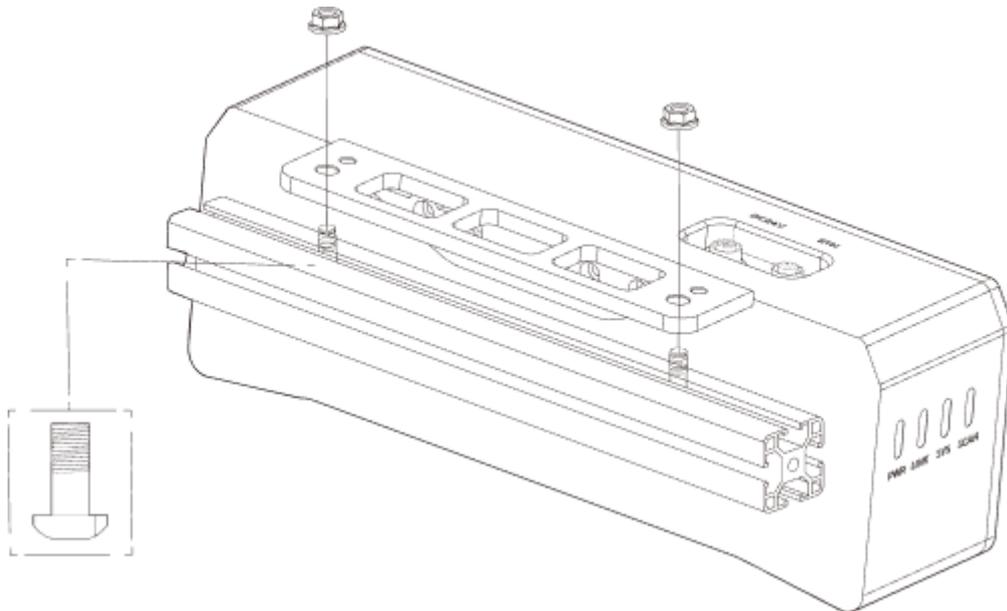
## ハードウェアの取り付け

### カメラの取り付け

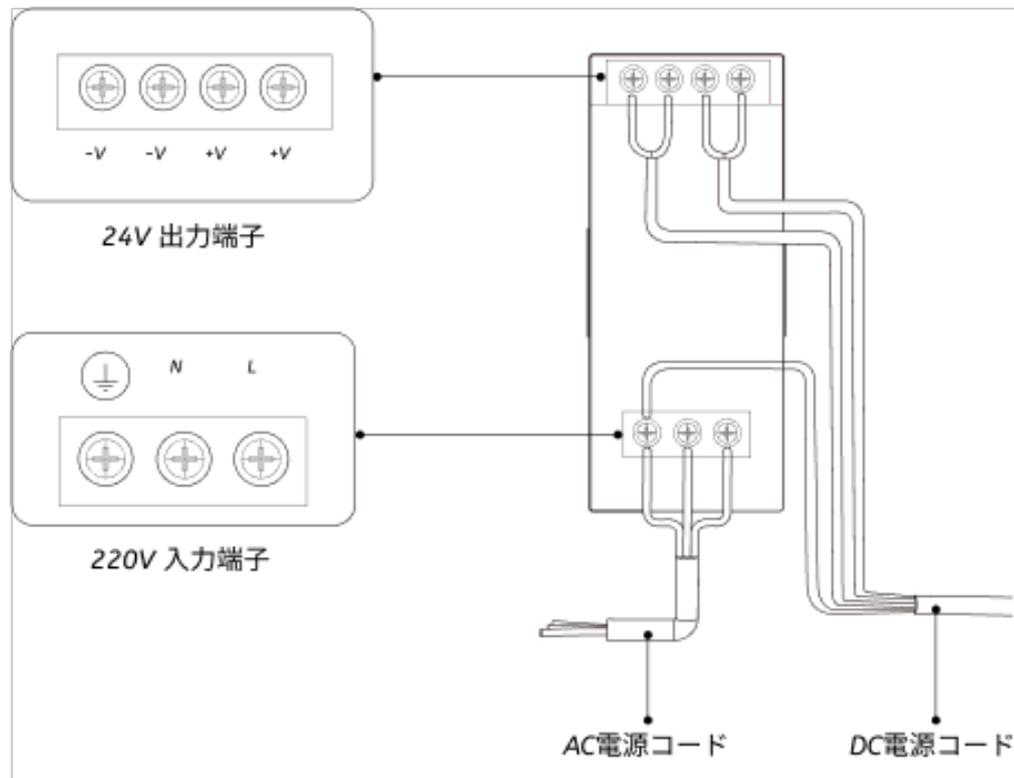


本ガイドでは、カメラはカメラスタンドに設置されています（Eye to Hand方式）。また、カメラをロボット先端に取り付けることも可能です（Eye in Hand方式）。

1. 付属品からカメラを取り付けるためのネジとスパナを探します。
2. 下図に示すように、レンチを使用して2本のネジを締め、カメラを固定します。

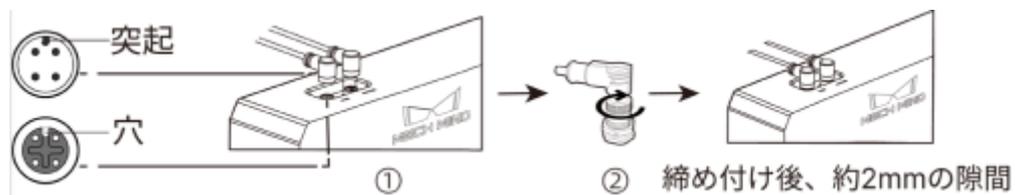


3. カメラ取り付け後、レンズ保護フィルムを剥がします。
4. レール電源でカメラの電源を入れます。
  - DC電源ケーブルの接続：
    - +Vを24V出力端子の+Vに接続します。
    - -Vを24V出力端子の-Vに接続します。
    - PEを220Vの入力端子⊕に接続します。



#### 5. カメラのLANケーブルを取り付けます。

カメラのLANケーブルの航空コネクタの突起をETHポートの開口部に差し込み、ナットを締めます。



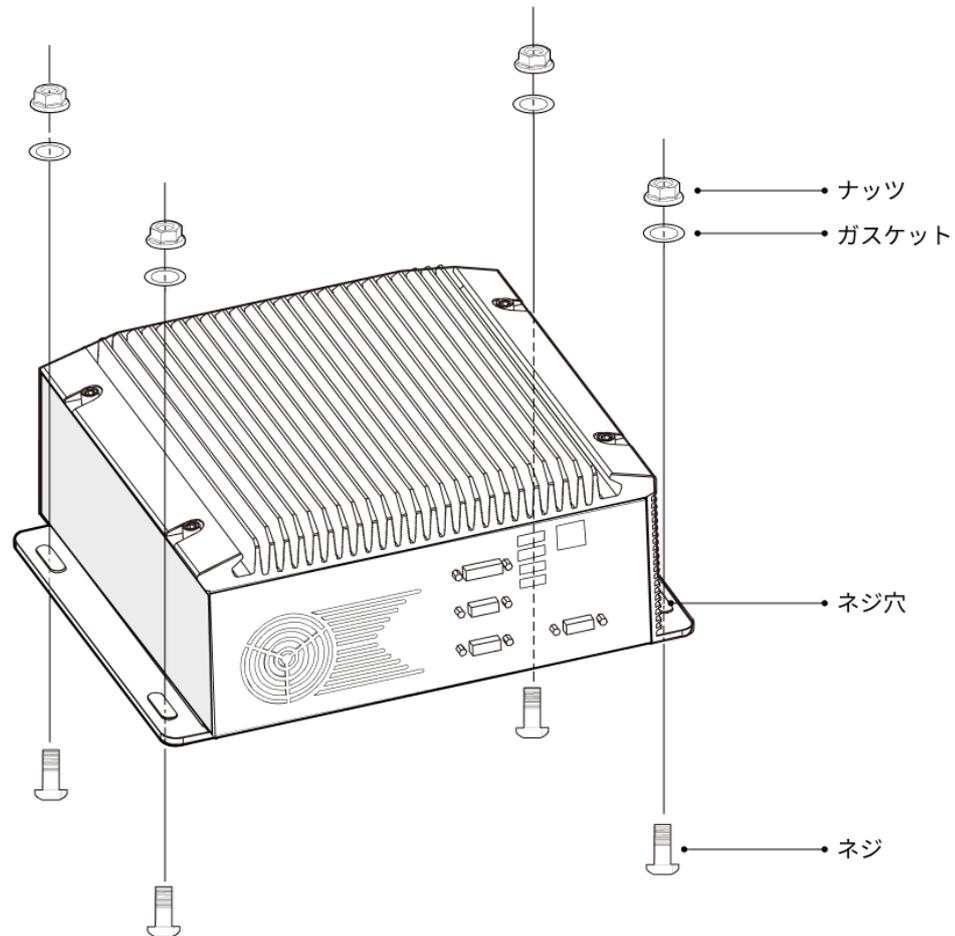
#### IPCの取り付け



IPCは一般的にコントローラに設置されます。IPCの設置環境は、放熱性、通気性、防塵効果が必要であり、LANケーブル、HDMIケーブル、USBポートの設置やメンテナンスがしやすい場所を選択する必要があります。

IPCを取り付けるには、以下の手順を実行します。

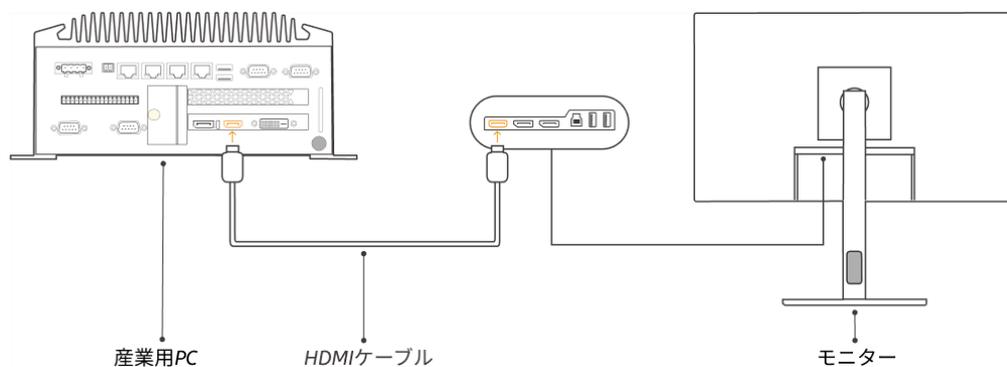
1. スパナ、ネジ、ナット、スペーサーは各自でご用意ください。
2. コントローラにIPCの取り付け穴がある場合は、下図のようにネジ、スペーサー、ナットの順に入れ、スパナでナットを締めて、コントローラにIPCを取り付けます。



コントローラの位置が移動しない場合は、この手順を省略してIPCをコントローラに直接設置することができます。

### 3. IPCとモニターをHDMIケーブルで接続します。

下図のように、HDMIケーブルの一端をモニターのHDMIポートに、もう一端をIPCのHDMIポートに差し込みます。



### 4. 電源アダプタでIPCの電源を入れます。

電源アダプタの電源プラグをIPCの電源入力ポートに差し込み、電源アダプタのもう一方の端を電源に接続します。

5. ドングルを差し込みます。

ドングルをIPCのUSBポートに差し込みます。

6. IPCの電源が入った後、IPCを起動します。

- IPCの正常な起動後は、電源インジケータは常に点灯している必要があります。
- IPCが起動しない場合は、Mech-Mindにお問い合わせください。

## ネットワーク接続

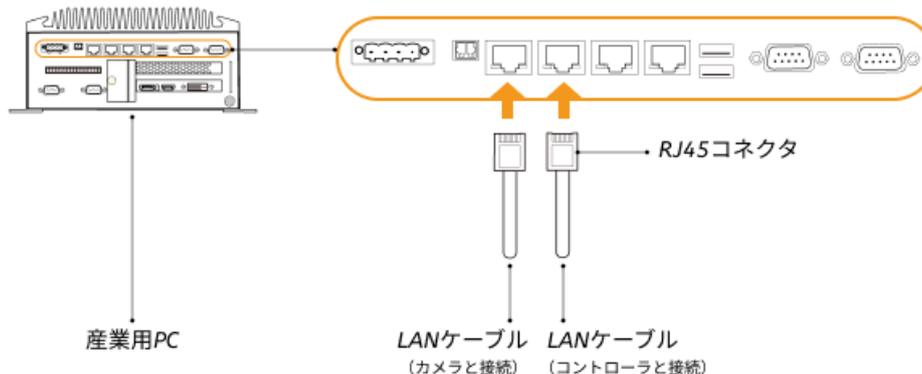
以下では、IPCとカメラとのネットワーク接続、IPCとロボットとのネットワーク接続について説明します。

以下の操作で、次のIPアドレスに従ってネットワークを設定します。実際のネットワーク環境に応じて調整してください。

| 機器   |                      | IPアドレス                     |
|------|----------------------|----------------------------|
| IPC  | カメラに接続されているLANポート    | 192.168.100.10             |
|      | コントローラに接続されているLANポート | 192.168.200.10             |
| カメラ  |                      | 192.168.100.20             |
| ロボット |                      | 192.168.200.20 (ロボットが設定済み) |

### IPCとカメラ、IPCとロボットコントローラを接続

1. カメラが接続されているLANケーブルのもう一方の端をIPCのLANポートに差し込みます。



2. 両端にRJ45コネクタが付いたLANケーブルを使用して、LANケーブルの一端をIPCのLANポートに、もう一端をロボットコントローラのLANポートに差し込みます。

### IPCのIPアドレスを設定

1. IPCで、**コントロールパネル** > **ネットワークとイーサネット** > **ネットワークと共有センター** > **アダプターの設定の変更**を選択すると、**ネットワークの接続**の画面が表示されます。
2. カメラが接続されているLANポートを選択し、右クリックして**名前の変更**を選択し、「To\_camera」など、ネットワーク接続を示すポートに名前を変更します。
3. カメラが接続されているLANポートを選択し、右クリックして**プロパティ**を選択し、**イーサネットのプロパティ**の画面に入ります。
4. **イーサネットプロトコルバージョン 4 (TCP/IPv4)** を選択して[ **プロパティ** ]をクリックし、**Internet プロトコルバージョン 4 (TCP/IPv4) のプロパティ**の画面に入ります。
5. **次のIPアドレスを使う**を選択し、**IPアドレス**を「192.168.100.10」、**サブネットマスク**を「255.255.255.0」、**デフォルトゲートウェイ**を「192.168.100.1」に設定してから、[ **OK** ]をクリックします。



6. 手順2~5を繰り返し、ロボットコントローラが接続されているLANポートの名前を変更し（例：To\_robot）、LANポートのIPアドレスを設定します。例えば、LANポートのIPアドレスは「192.168.200.10」です。



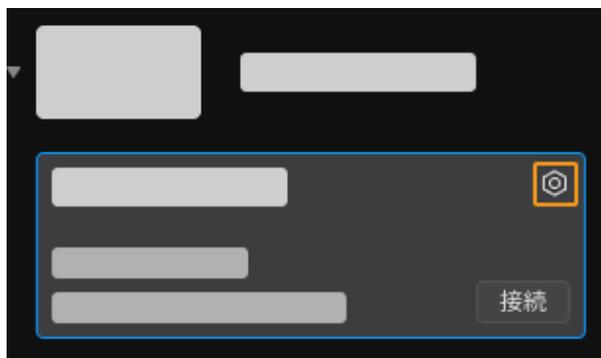
ロボットコントローラに接続するIPCのLANポートのIPアドレスは、ロボットのIPアドレスと同じネットワークセグメント内にある必要があります。

### カメラのIPアドレスを設定

1. IPCのデスクトップ上にある  アイコンをダブルクリックし、Mech-Eye Viewerを起動しま

す。

- 検出されたカメラリストからカメラを選択し、マウスをカメラ情報バーに移動すると、が表示されます。クリックすると**IP設定画面**に入ります。



カメラが検出できない、接続できない場合は、[カメラのトラブルシューティング](#)を参照し、解決してください。

- カメラエリアで、**静的IPに設定**を選択し、**IPアドレスクラス**を「クラスC 192.168.x.x」、**IPアドレス**を**192.168.100.20**、**サブネットマスク**を「255.255.255.0」に設定してから、**[適用]**をクリックします。



カメラのIPアドレスは、カメラが接続されているIPCのLANポートのIPアドレスと同じネットワークセグメント内にある必要があります。

## ネットワーク接続をテスト

- ショートカットキー **Win + R** で**実行画面**を表示します。
- 名前に**cmd**と入力し、**[OK]**をクリックします。

3. コマンドウィンドウにping XXX.XXX.XX.XXと入力し、[Enter]をクリックしてコマンドを実行します。



XXX.XXX.XX.XXは、実際に設定されたカメラまたはロボットのIPアドレスに変更します。

ネットワークの接続が正常であれば、次のようなメッセージが表示されます。

```
XXX.XXX.XX.XXにPingを送信しています 32バイトのデータ:
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128
XXX.XXX.XX.XXからの応答: バイト数=32 時間<1ms TTL=128
```

## ソフトウェアのアップグレード（オプション）

Mech-Mindから購入したIPCには、Mech-Mindソフトウェアシステムがプリインストールされています。

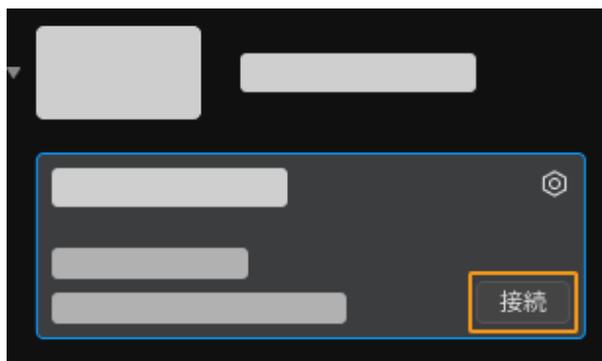
本機のソフトウェアが最新バージョンであるかどうかを確認してください。ソフトウェアがすでに最新バージョンである場合は、以下の内容をスキップしてください。ソフトウェアが最新バージョンでない場合は、以下の内容を参照して、ソフトウェアを最新バージョンにアップグレードしてください。

- [Mech-Eye SDKソフトウェアのダウンロードとインストール](#)
- [Mech-Vision、Mech-Viz、Mech-Centerソフトウェアのダウンロードとインストール](#)

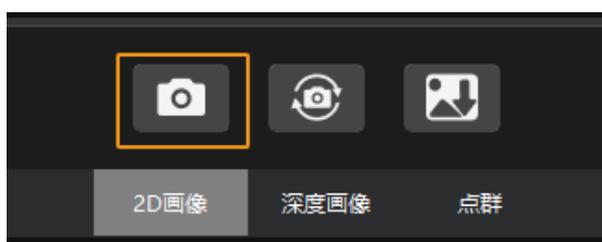
## ビジョンシステムが正常に画像を取得できるか確認

IPC、カメラとロボット間のネットワークにアクセスできることを確認した後、以下の手順でビジョンシステムが正常に画像を取得できることを確認します。

1. ワークをカメラの視野中心に置き、エッジと最高層にあるワークが視野に入ることを確認します。
2. IPCのデスクトップにある  アイコンをダブルクリックしてMech-Eye Viewerを起動します。
3. 検出されたカメラからカメラを選択して[接続]をクリックします。

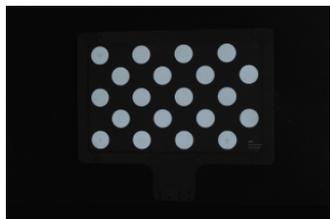
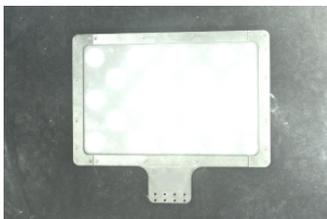
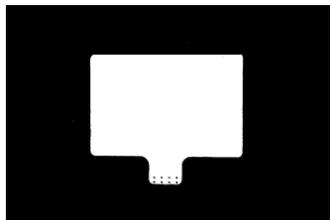
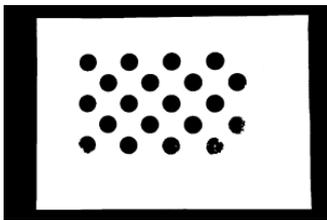
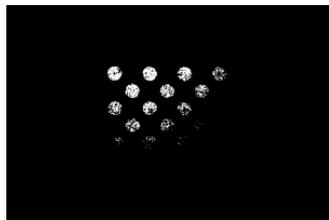


4. カメラ接続後、[一回キャプチャ]をクリックします。



5. 撮影した画像の品質が条件を満たしていることを確認します。

- 2D画像：明らかな露出過度（白すぎて対象物が見えない）、露出不足（暗すぎて対象物のディテールが判別できない）がありません。
- 深度画像と点群：対象物に激しい抜けがありません。

|      | 正常                                                                                  | 露出過度                                                                                 | 露出不足                                                                                  |
|------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 2D画像 |  |  |  |
| 点群   |  |  |  |



取得した画像が要件を満たしていない場合、[Mech-Eye Viewer](#)でパラメータを調整してください。

以上でビジョンシステムのハードウェアの構築方法の説明は終わりです。

## 4.2. ロボット通信設定

本ガイドを読むことで、Vizとの通信プログラムをABBロボットに読み込む方法、Vizティーチン

グの通信設定を習得できます。



- Vizとの通信プログラムの読み込みは、Vizとの通信プログラムと設定ファイルをロボットシステムにロードすることで、ビジョン側とロボット側との通信を確立し、Mech-Mindソフトウェアシステムによるロボットの制御を実現します。
- それ以外のロボットを使用している場合は、[Vizティーチング通信](#)を参照してロボットの通信設定を行います。

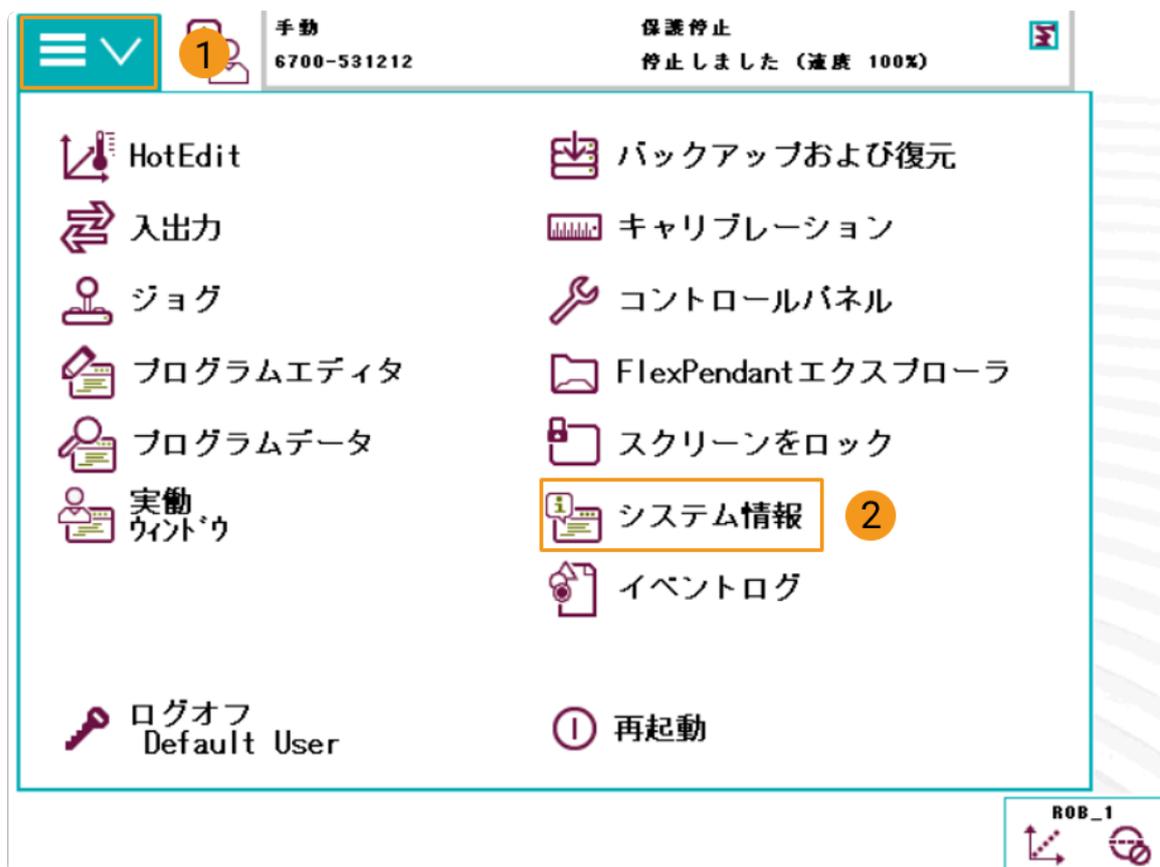
▶ <https://www.youtube.com/watch?v=kPFvHWFKIYA/PLVcMd7cW2rXWbHhTQX8m2R1r6PhX6vrCl> (YouTube video)

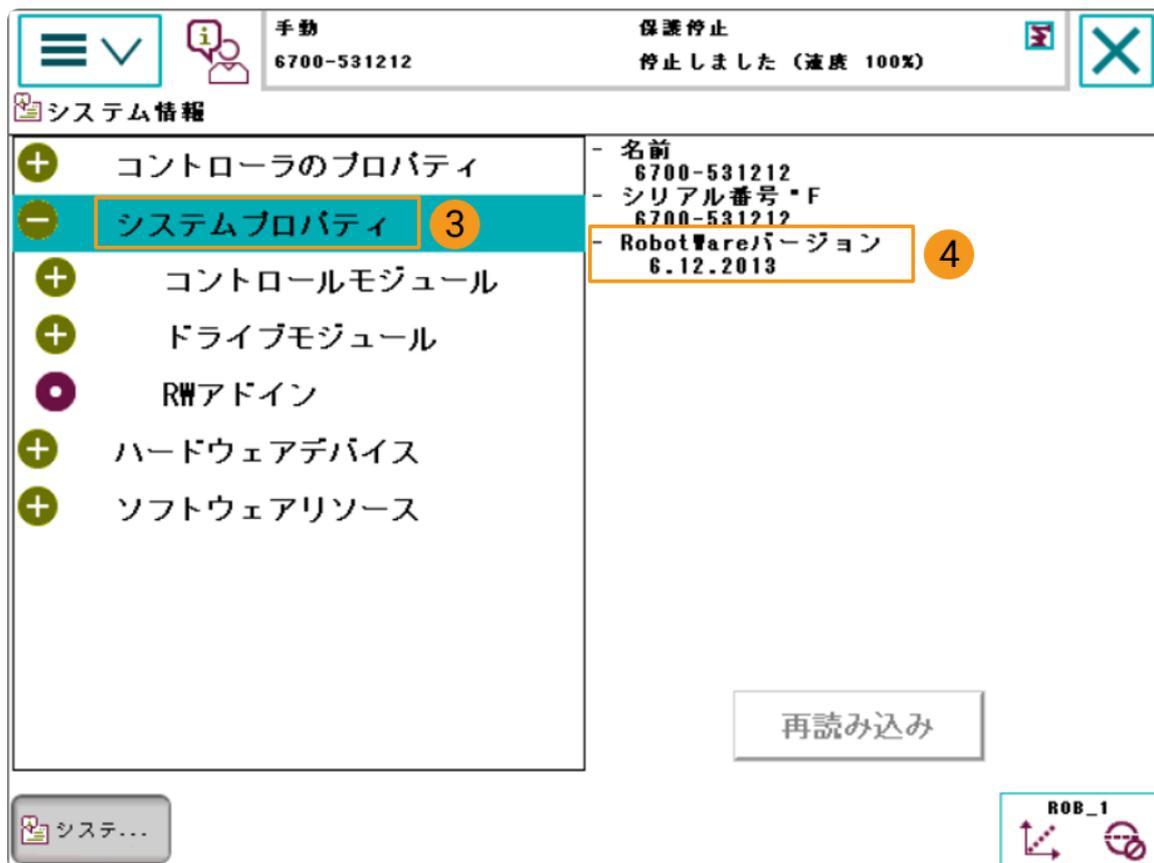
動画：ロボット通信設定

## 事前準備

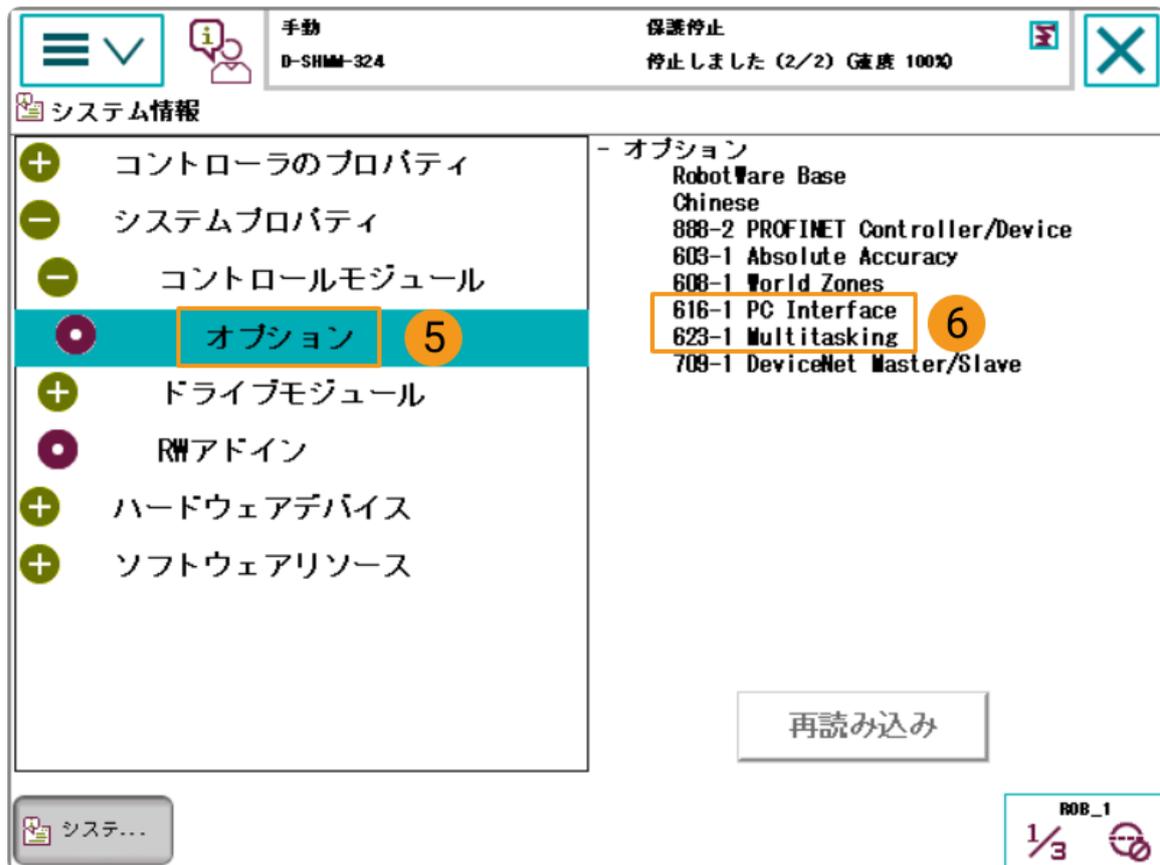
### コントローラとRobotWareのバージョンを確認

1. コントローラにD652またはDSQC1030のIOボードが装着されていることを確認します。
2. ティーチペンダントでRobotWareのバージョンが6.0以上であることを確認します。





3. ティーチペンダントで、コントローラに以下のコントロールモジュールがインストールされていることを確認します。
- 623-1 Multitasking
  - 616-1 PCInterface



上記の条件を満たしていない場合は、Vizティーチング通信ができませんので、ロボットメーカーにお問い合わせください。

## ロボットシステムのリセット

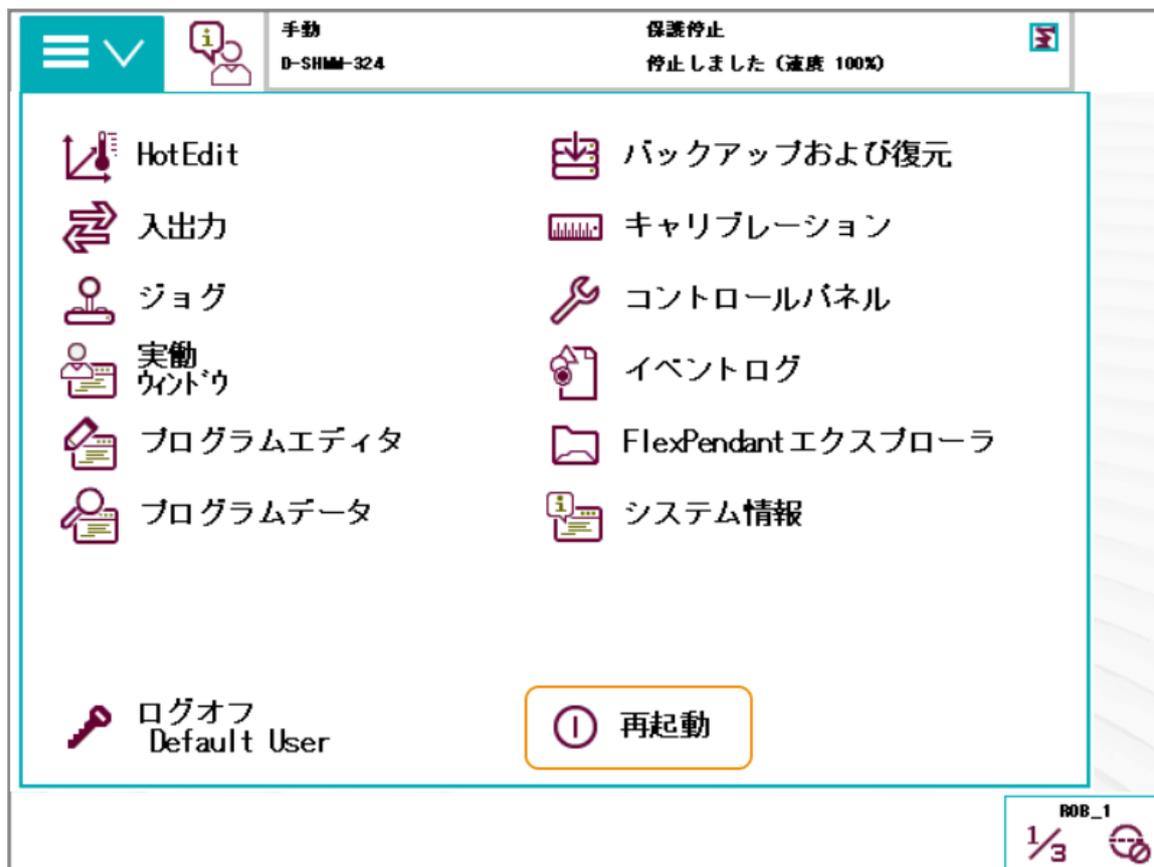
コントローラのハードウェアとソフトウェアの状態を確認した後、ロボットシステムのリセットが必要です。お使いのロボットは工場出荷時の状態である場合、ロボットシステムのリセットは不要です。



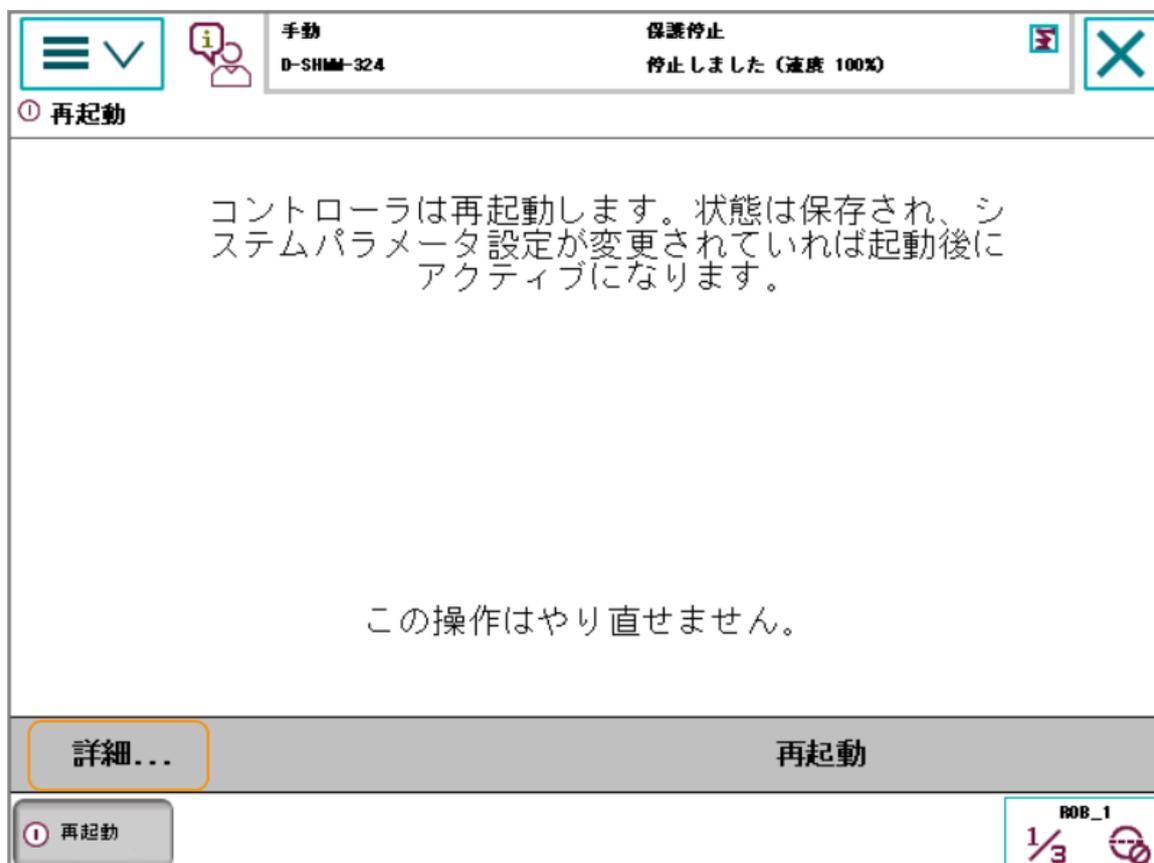
リセットすると工場出荷時の設定に戻りますので、バックアップ操作が完了していることを確認してください。

ロボットシステムをリセットするには、以下の手順を実行します。

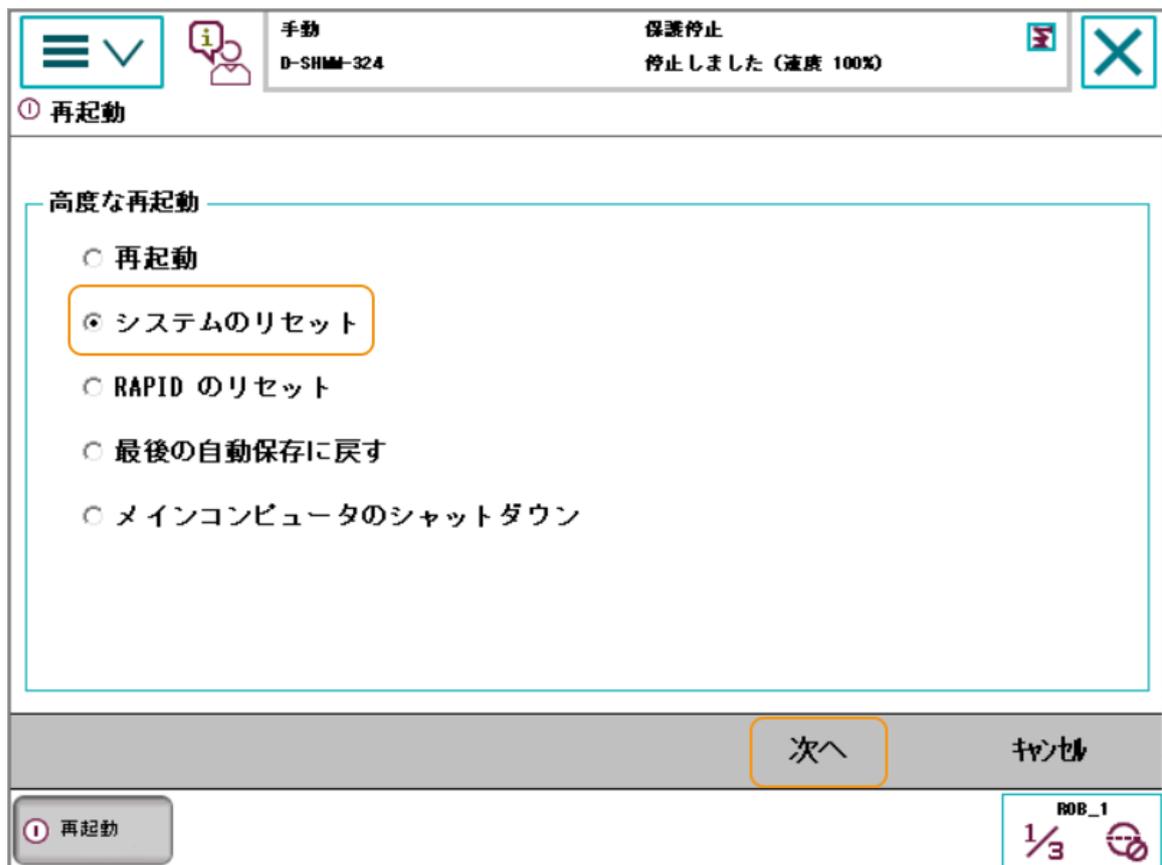
1. ティーチペンダントで、左上隅のメニューバーをクリックしてメイン画面を表示し、[再起動]をクリックします。



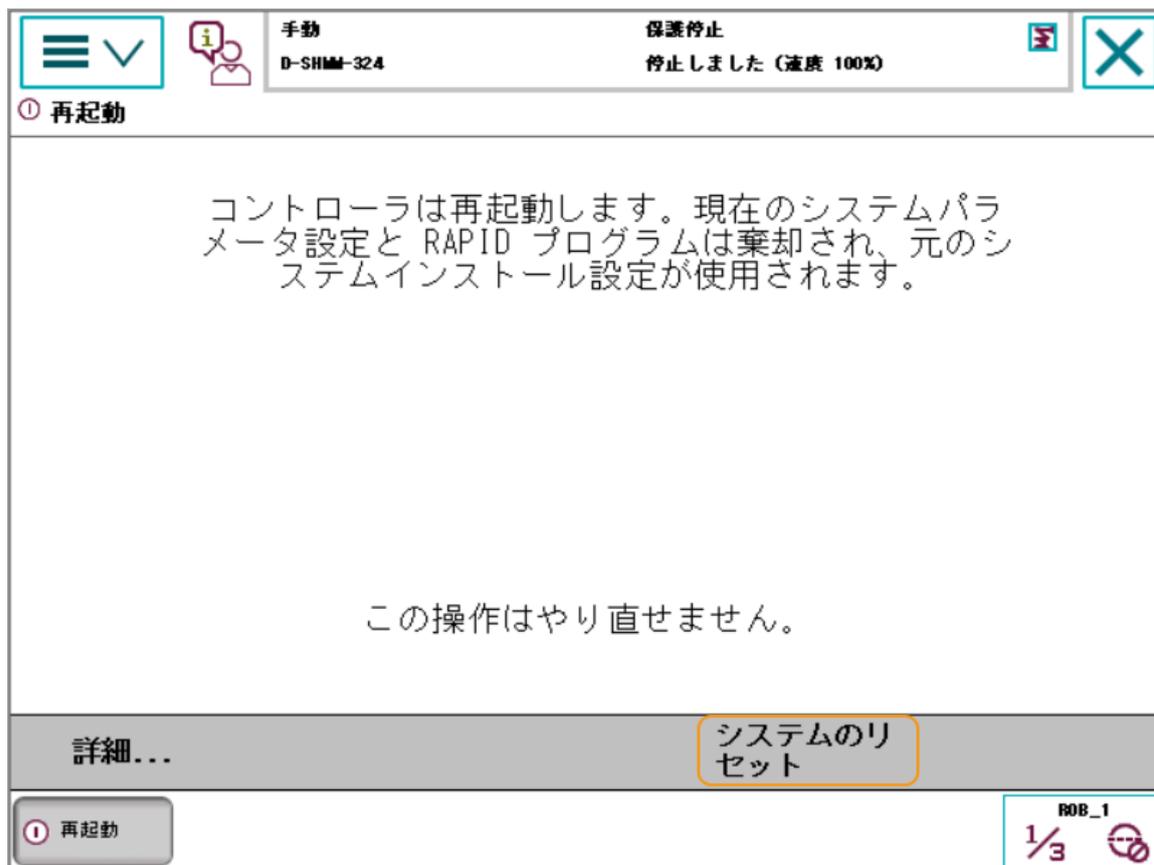
2. [詳細...]をクリックします。



3. システムのリセットを選択し、[次へ]をクリックします。



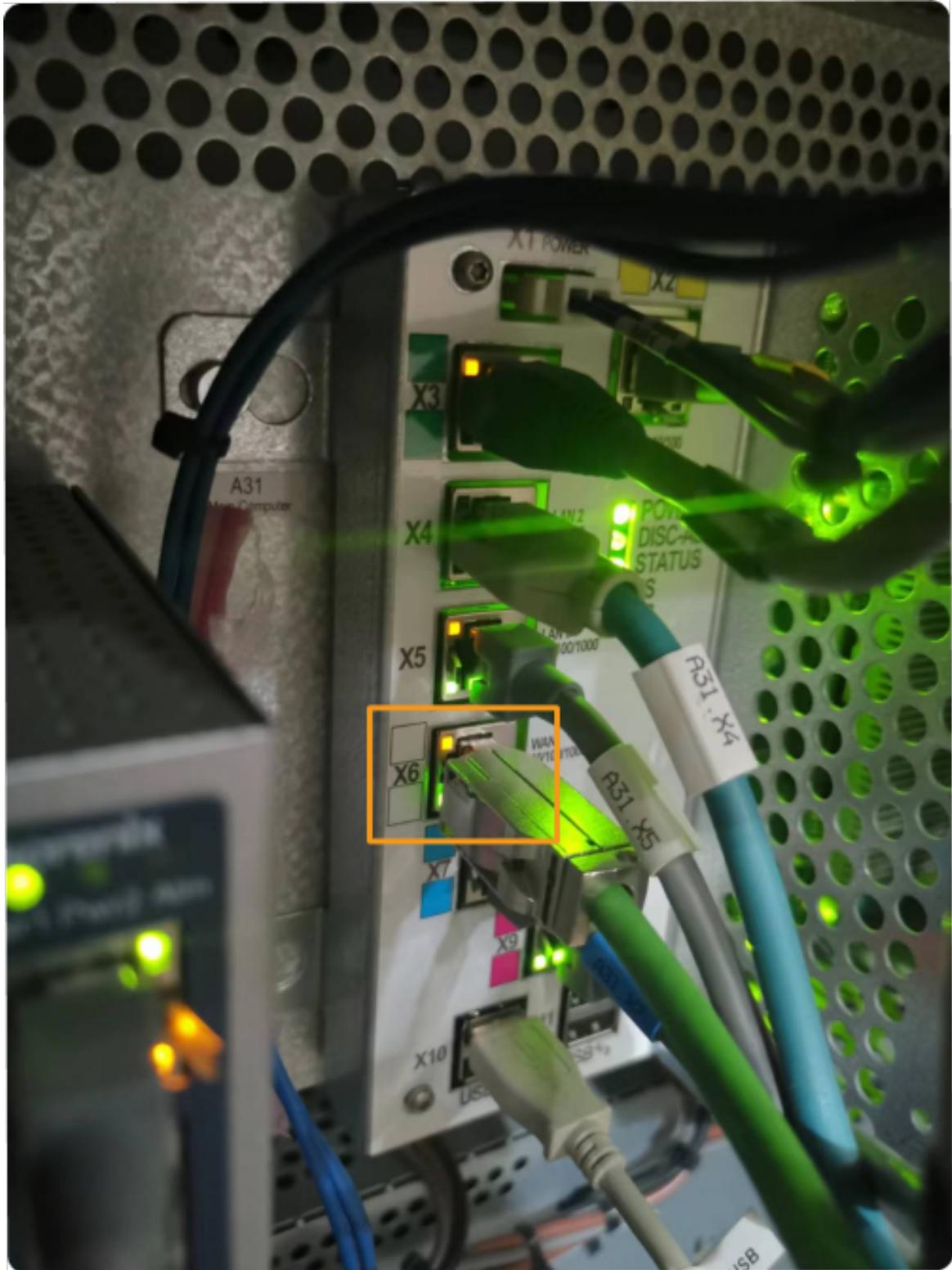
4. [システムのリセット]をクリックします。



リセットにかかる時間は1~2分程度です。再びメイン画面が表示されると、リセットは完了します。

## ネットワーク接続

1. IPCのLANケーブルのもう一端を、下図のようにロボットコントローラの **X6 (WAN)** LANポートに接続します。



2. ABBロボットのIPアドレスが、IPCのIPアドレスと同じネットワークセグメントにあることを確認します。

### 読み込みファイルの準備

1. IPCで、Mech-Mindソフトウェアシステムのインストールディレクトリに格納されている

Mech-Center/Robot\_Server/Robot\_FullControl/abb/server on ABB フォルダを開きます。

- このフォルダをコピーしてUSBメモリに貼り付け、USBメモリをRobotStudioソフトウェアがインストールされているPCに差し込みます。



RobotStudioは、ABB製ロボットのシミュレーションおよびオフラインプログラミングソフトウェアです。IPCや他のPCにインストールすることができます。この例では、RobotStudioは別のPCにインストールされています。

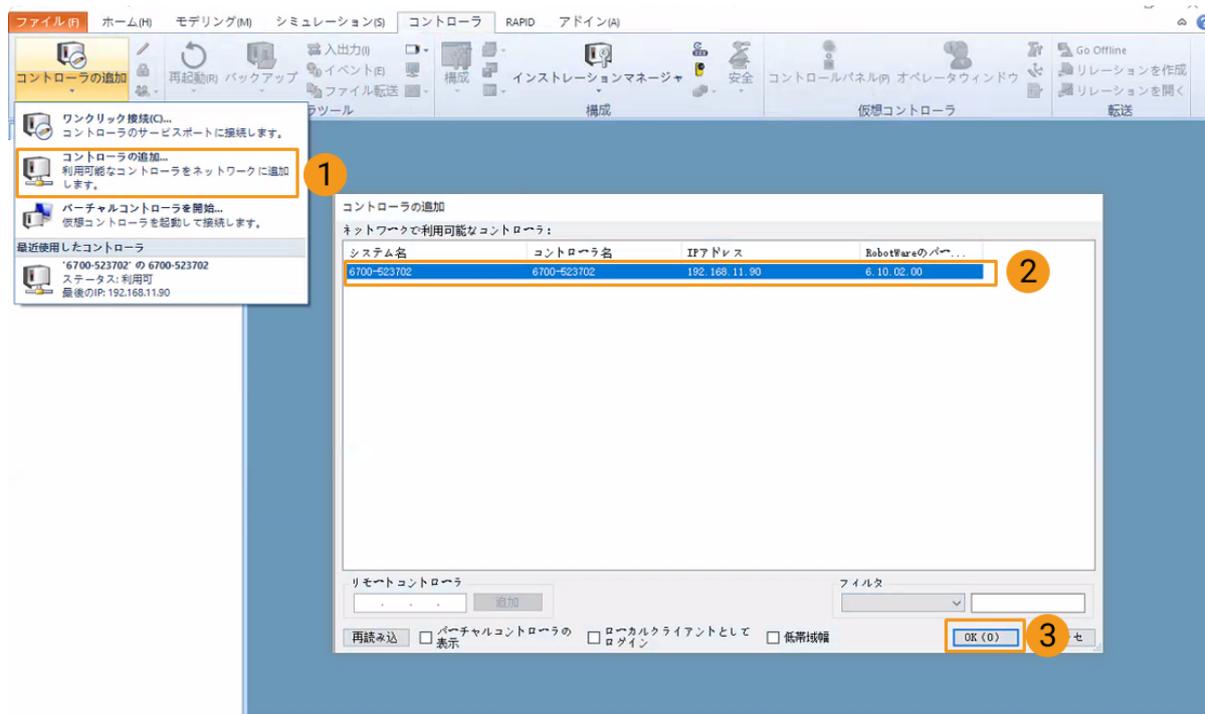
ファイルの説明は以下の通りです：

- フォルダ「MM」：ロボットプログラムモジュール。
- 「config」：ロボットの設定ファイル。
  - D652 IOボードを現場で使用する場合は、D652.cfgとSYS.cfgのファイルが必要です。
  - DSQC1030 IOボードを現場で使用する場合は、DSQC1030.cfgとSYS.cfgのファイルが必要です。
  - D652 IOまたはDSQC1030 IOボードを現場で使わない場合は、EIO.cfgとSYS.cfgが必要です。

## ロボットプログラムの読み込み

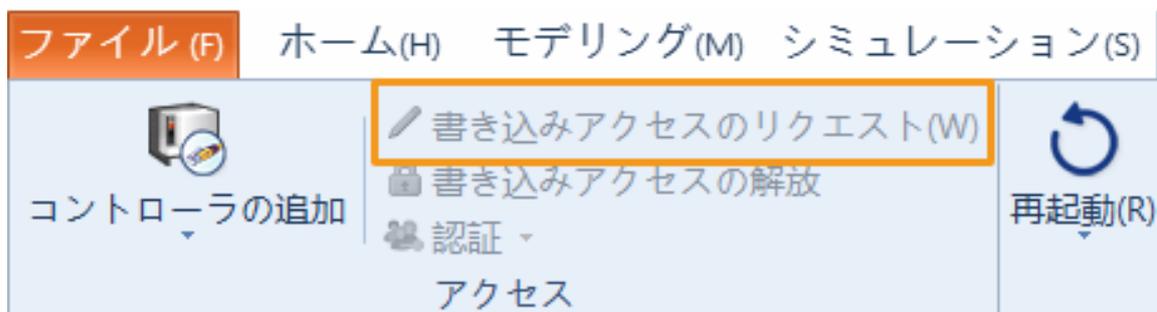
### RobotStudioの起動とコントローラの接続

RobotStudioのコントローラメニューをクリックし、ツールバーで**コントローラの追加**、**コントローラの追加**を選択します。表示される**コントローラの追加**画面で、コントローラを選択して[OK]をクリックします。



## ロボットの書き込みアクセスの取得

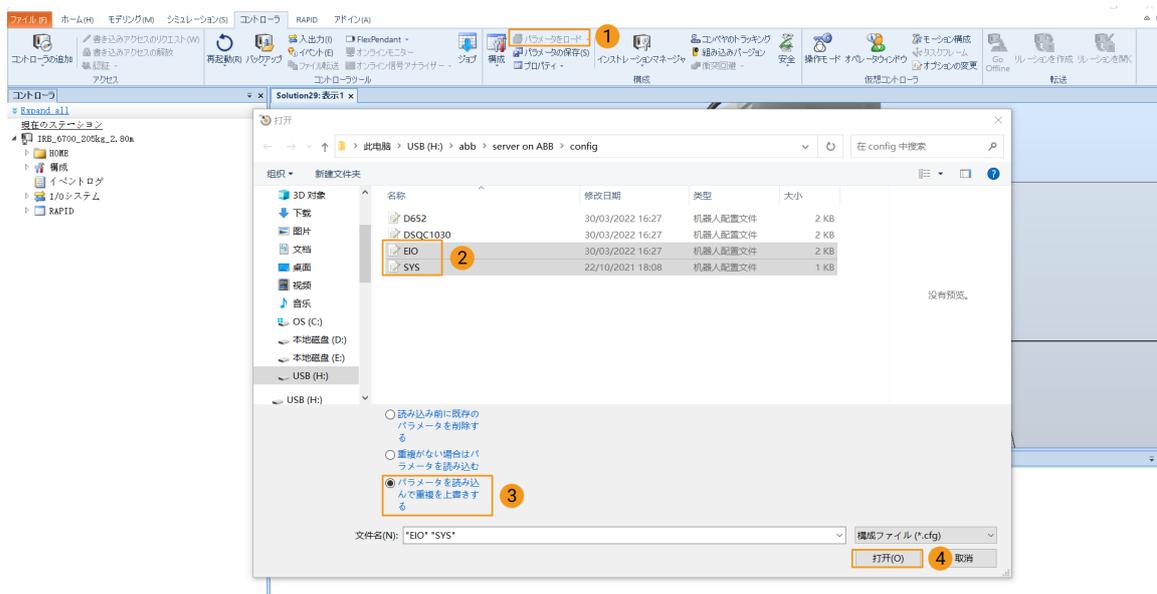
1. RobotStudioのツールバーで書き込みアクセスのリクエストをクリックします。



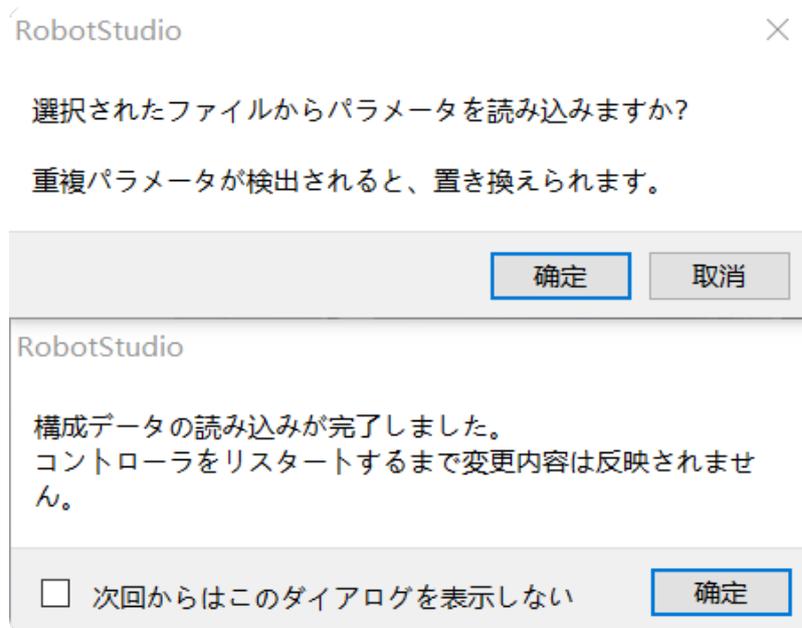
2. ティーチペンダントで表示される書き込みアクセスのリクエスト画面で[同意]をクリックします。

## ロボットの設定ファイルの読み込み

1. RobotStudioでコントローラのメニューをクリックし、ツールバーでパラメータをロードを選択し、USBメモリでロードする設定ファイルを選択し、パラメータを読み込んで重複を上書きするを選択してから、[開く]をクリックします。

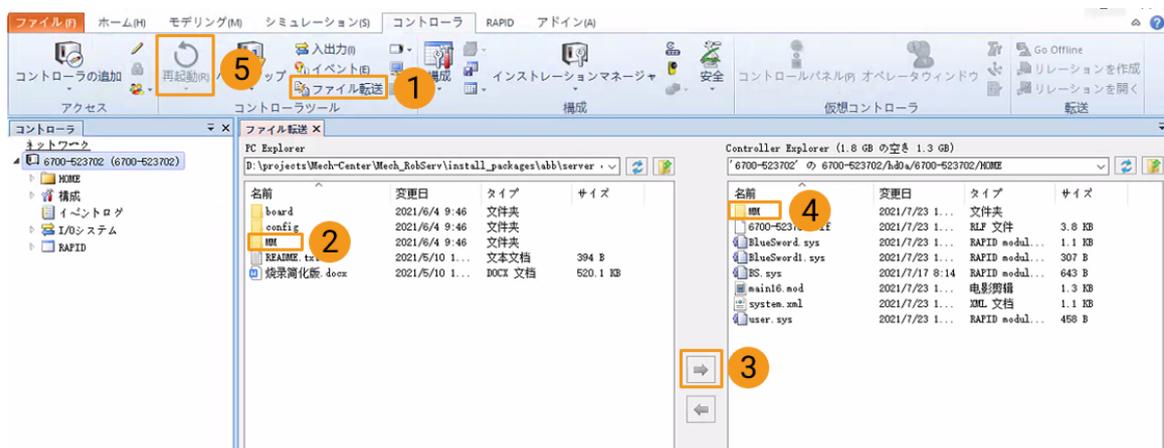


2. ポップアップするダイアログで[OK]ボタンを連続してクリックします。



## ロボットプログラムモジュールのロード

1. RobotStudioでコントローラメニューをクリックし、ツールバーでファイル転送をクリックします。ファイル転送画面の左側パネルで「MM」フォルダを選択し、転送ボタンをクリックしてロボットシステムのHOMEパスに転送します。



2. RobotStudioでコントローラメニューをクリックし、ツールバーで再起動をクリックしてロボットシステムを再起動します。

これで、Vizとの通信プログラムとコンフィグファイルはロボットに読み込まれています。

## Vizティーチング通信が有効であるかをテスト

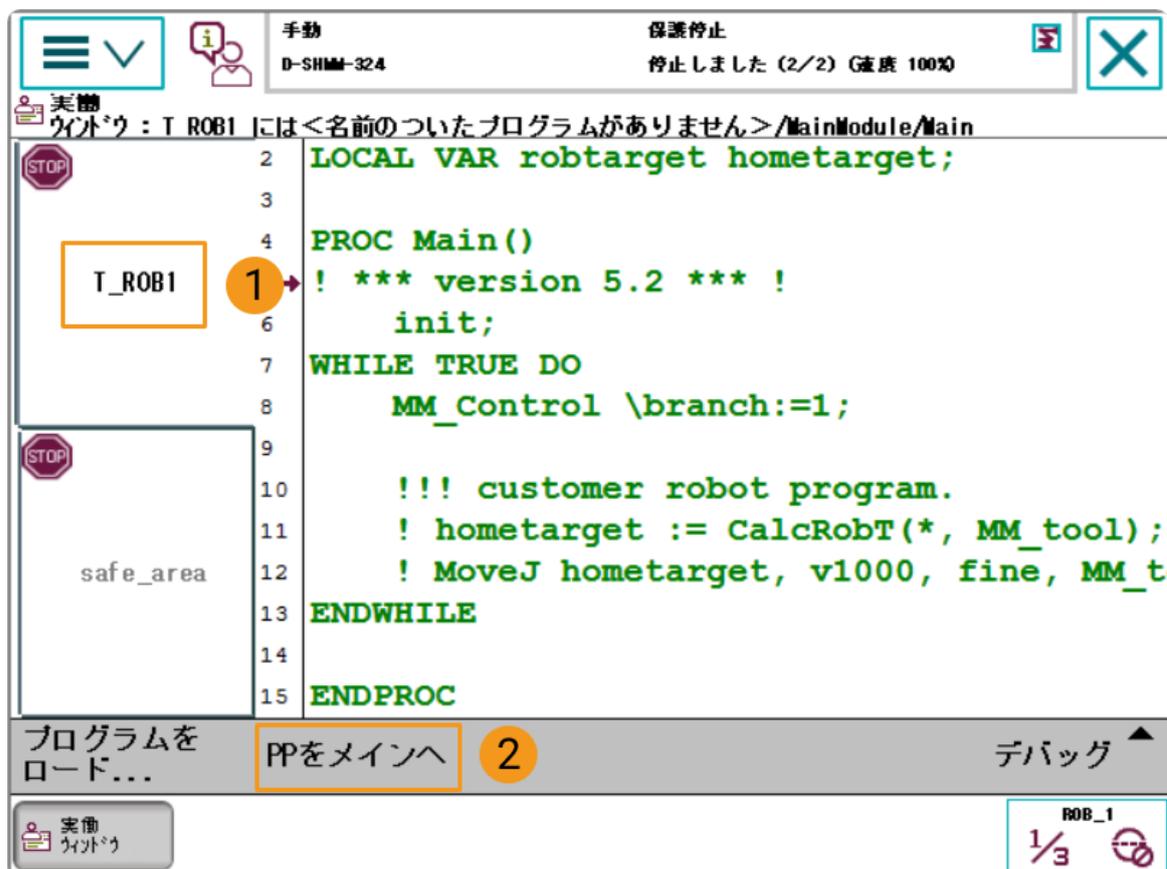
ロボットシステムが再起動後、Vizによりロボットをティーチングするかどうかをテストするには、以下の手順を実行します。

## ロボットの自動モードへの切り替え

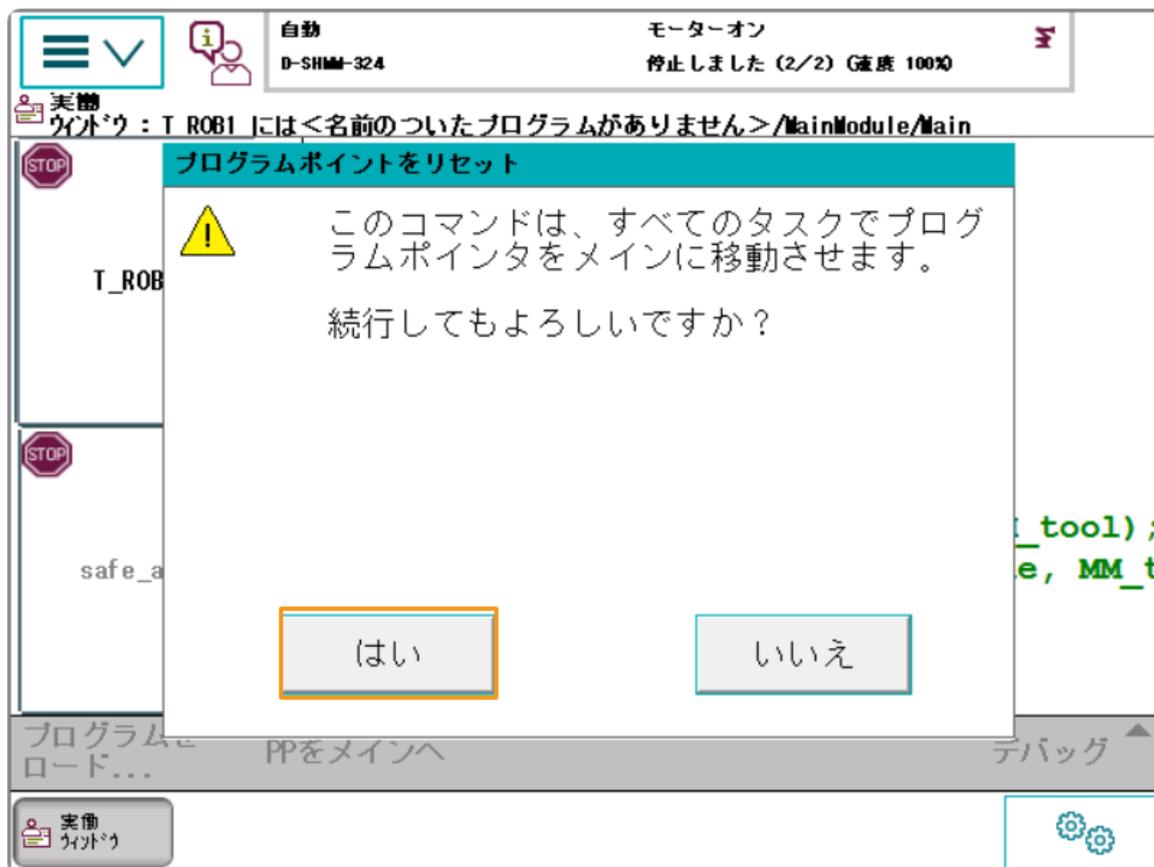
1. ロボットコントローラでは、キースイッチによってロボットを自動モードに切り替えます。
2. ティーチペンダントでポップアップするダイアログで[OK]をクリックします。
3. ロボットコントローラで、パワーオンのボタンを押してロボットの電源を入れます。このボタンは、電源投入に成功すると必ず点灯します。

## メインプログラムの実行

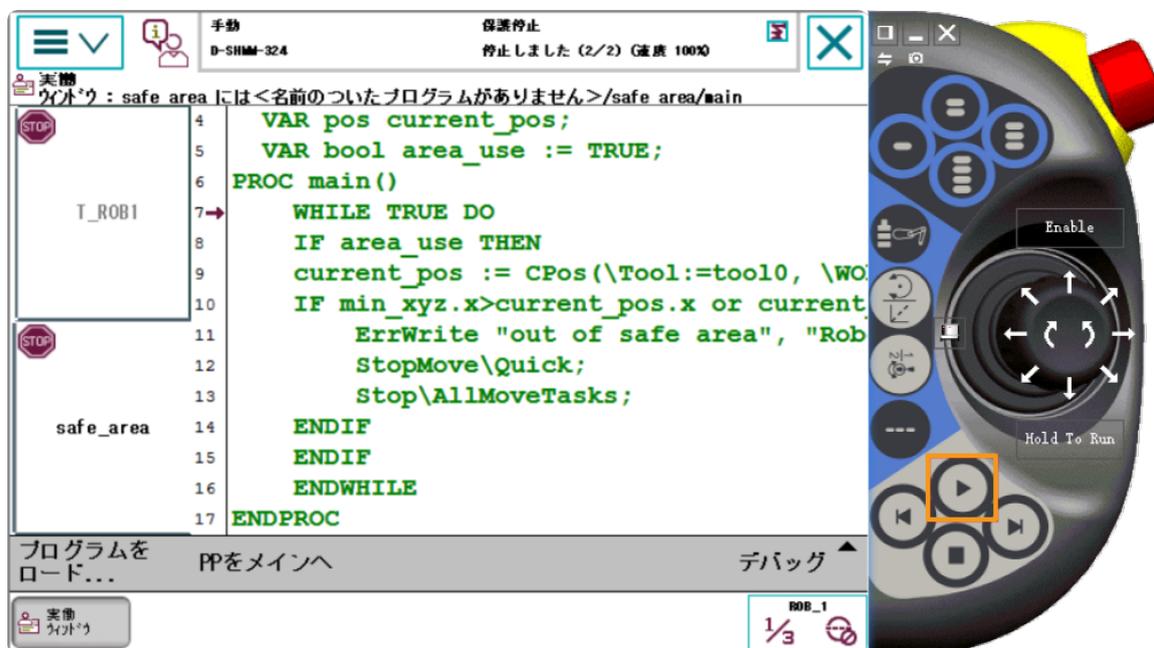
1. ティーチペンダントで、T\_ROB1のプログラムポインターをメインプログラムに移動してから、[PPをメインへ]をクリックします。



2. ポップアップするダイアログで[はい]をクリックします。

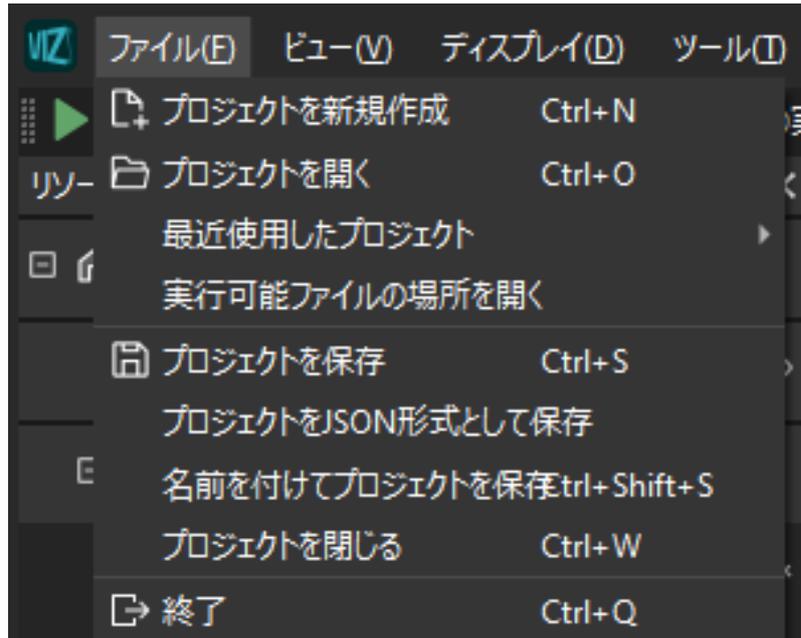


3. ティーチペンダントの右側の実行ボタンを押します。

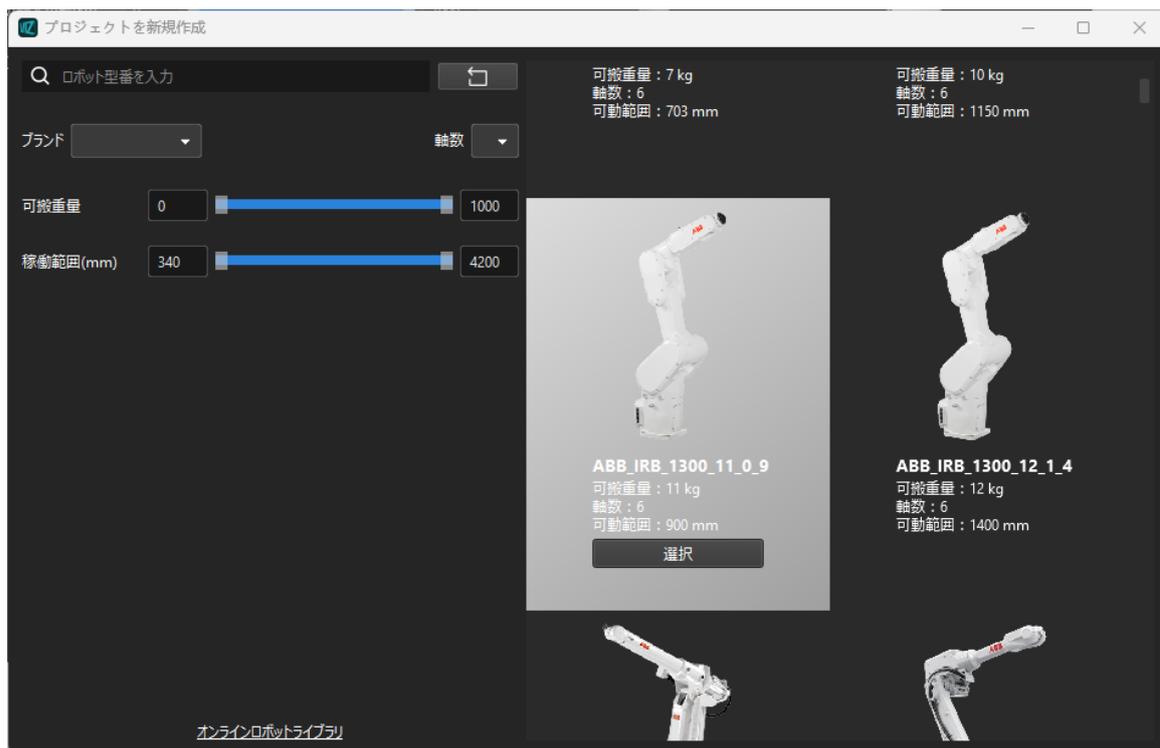


## Mech-Vizプロジェクトの作成

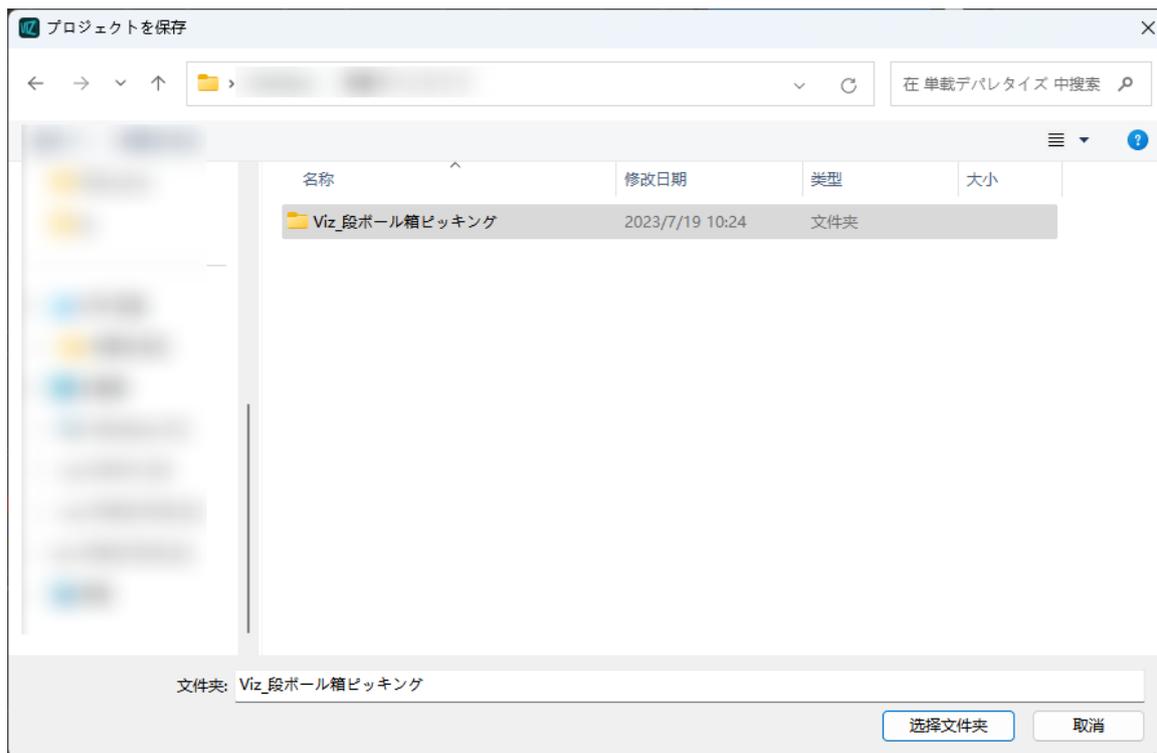
1. Mech-Vizを起動し、**ファイル** > **プロジェクトを新規作成**を選択します。



2. ロボットブランドを「ABB」に設定し、右側のパネルで「ABB\_IRB\_1300\_11\_0\_9」を選択してから[選択]をクリックします。



3. キーボードで **Ctrl** + **S** を押し、「Viz\_段ボール箱ピッキング」といったフォルダを新規作成して選択し、[フォルダの選択]をクリックします。



Mech-Vizプロジェクトが正常に保存されると、**リソースパネル**にプロジェクト名が「Viz\_段ボール箱ピッキング」と表示されるようになります。

- プロジェクト名を右クリックし、**自動的に読み込む**にチェックを入れます。



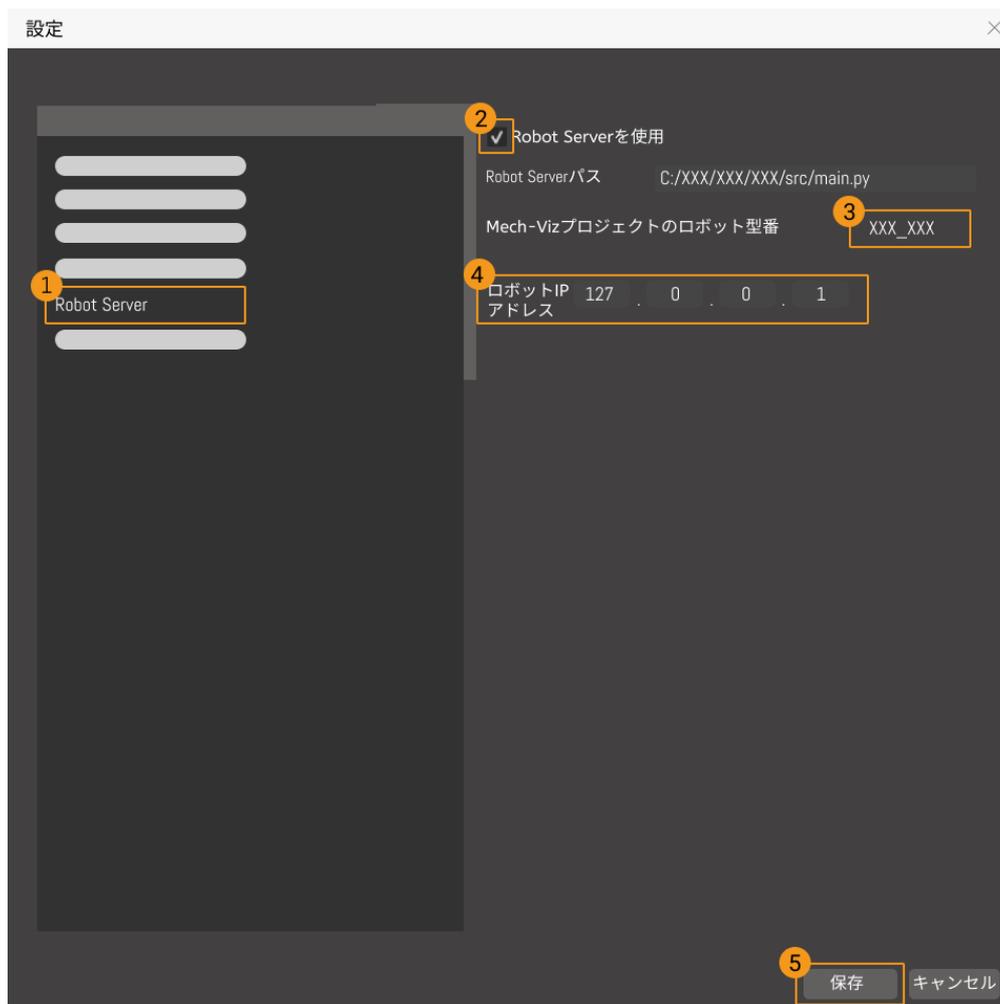
- ロボットを安全に動かすために、ツールバーのパラメータ **速度** と **加速度** を5%など小さい値に設定します。



- キーボードで **Ctrl + S** を押してプロジェクトを保存します。

### Mech-CenterでVizティーチング通信の関連設定を実行

- Mech-Centerを起動し、**設定** > **Robot Server** を選択し、**Robot Serverを使用**にチェックを入れます。
- ロボット型番がロボット実機の型番と同じであることを確認し、**ロボットIPアドレス**をロボット実機のIPアドレスに設定し、**[保存]**をクリックします。



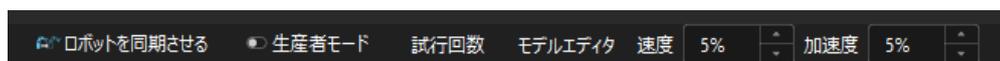
## ロボットを接続

Mech-Centerのツールバーで **ロボットを制御**  をクリックします。

- 接続に成功すると、サービスステータスバーにロボットのアイコンと型番が表示され、ログバーにロボット接続成功のメッセージが出力されます。
- 接続に失敗した場合は、前の操作が正しかったかどうか再確認してください。

## ロボットを移動

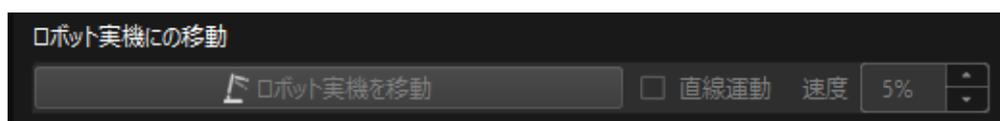
1. Mech-Vizのツールバーで**ロボットを同期させる**をクリックし、仮想ロボットの位置姿勢とロボット実機の位置姿勢を同期させ、再度クリックして**ロボットを同期させる**の選択を解除します。



2. **ロボット**パネルの関節角度のオプションで、J1関節角度を調整（例えば、0°から3°に調整）すると、仮想ロボットが動くようになります。



3. [ロボット実機を移動]をクリックします。



ロボットを操作する際は、人の安全を確保してください。緊急時には、ティーチペダントの緊急停止ボタンを押してください！

ロボット実機が仮想ロボットの位置姿勢に移動することができたら、Vizティーチング通信が有効になります。

## 4.3. ロボットハンド・アイ・キャリブレーション

本ガイドを読むことで、自動ハンド・アイ・キャリブレーション（Eye to Hand）の実行方法を習得できます。



ハンド・アイ・キャリブレーションとは、カメラ座標系とロボット座標系との対応関係（外部パラメータ）を求めることです。ビジョンシステムにより取得した対象物の位置姿勢をロボット座標系に変換し、ロボットが対象物の把持を完了させるように制御します。

▶ <https://www.youtube.com/watch?v=u4m1Jsklfk8/PLVcMd7cW2rXWbHhTQX8m2R1r6PhX6vrCl> (YouTube video)

動画：ハンド・アイ・キャリブレーション (Vizティーチング)

## 事前準備

本節では、キャリブレーションボードの取り付け、カメラパラメータの調整、キャリブレーション前の設定を行う必要があります。

### キャリブレーションボードの取り付け



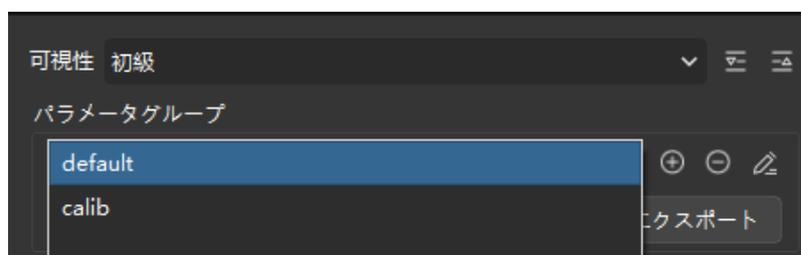
Eye to Handモードでは、キャリブレーションボードはロボット先端のフランジに取り付ける必要があります。

以下の手順を実行します。

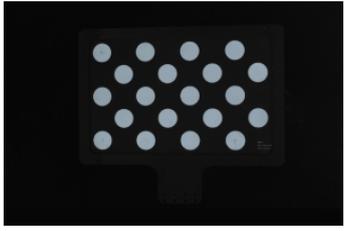
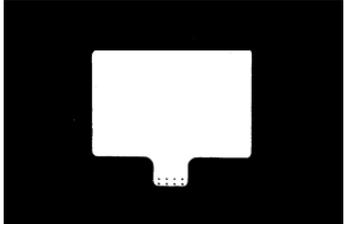
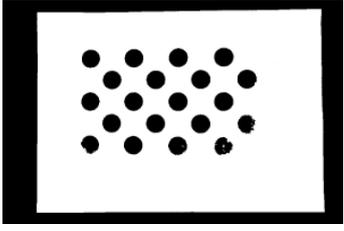
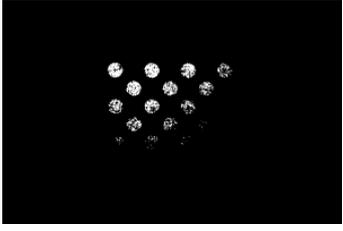
1. カメラ梱包箱からキャリブレーションボードとフランジを取り出します。
2. ネジ、スペーサー、ナットを使って、フランジをロボット先端に固定します。
3. ネジ、スペーサー、ナットを使って、キャリブレーションボードをフランジに固定します。
4. 取り付け後、ロボットを作業領域の最下層のワークの上面、カメラの視野中心に移動させます。

### カメラパラメータを調整

1. Mech-Eye Viewerでカメラを接続し、**パラメータグループ**を「calib」に設定します。

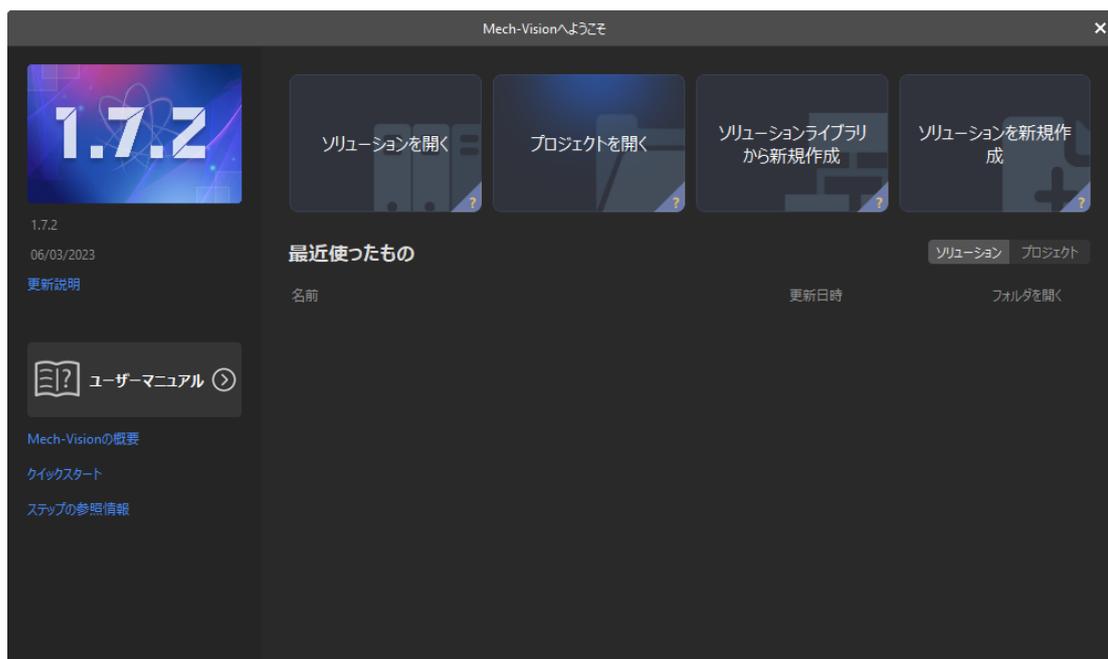


2. 2D画像内のキャリブレーションボードがはっきり見え、露光過度や露光不足がないように、2Dパラメータを調整します。
3. キャリブレーションボード上の白い円の点群が完全なものになるように、3Dパラメータを調整します。点群の変動範囲を減らすために、**点群後処理の表面平滑化と外れ値除去**を**Normal**に変更することをお勧めします。

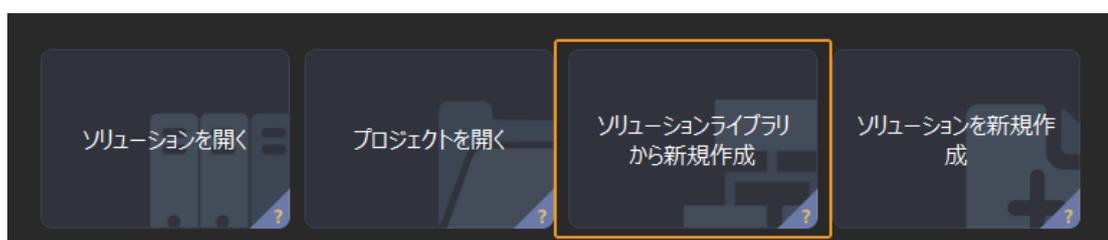
|      | 正常                                                                                | 露光過度                                                                               | 露光不足                                                                                |
|------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| 2D画像 |  |  |  |
| 点群   |  |  |  |

## Mech-Visionソリューションの作成と保存

1. Mech-Visionを起動すると、下図のようなようこそ画面が表示され、Mech-Vision が正常に起動されたことを示します。



2. Mech-Visionのようこそ画面で [ソリューションライブラリから新規作成] をクリックし、ソリューションライブラリを開きます。





ソリューションライブラリには、異なる業界に適応可能なソリューションやプロジェクトのソースライブラリが含まれています。

3. ソリューションライブラリを開いた画面を下図に示します。同じ種類の段ボール箱 プロジェクトを選択します。



ソリューションライブラリからこのプロジェクトが見つからない場合、画面下部にある [もっと詳しく] をクリックします。

4. このプロジェクトをクリックして選択すると、プロジェクトの関連情報は画面下部に表示されます。ソリューション名と保存先を設定したら、[作成] をクリックします。

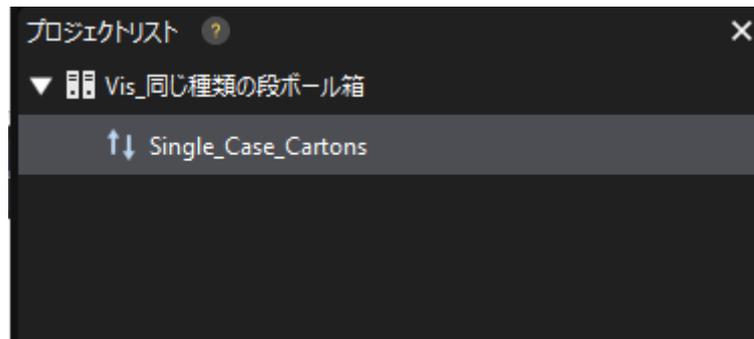


プロジェクトが作成されると、Mech-Visionメイン画面の左上にあるプロジェクトリスト

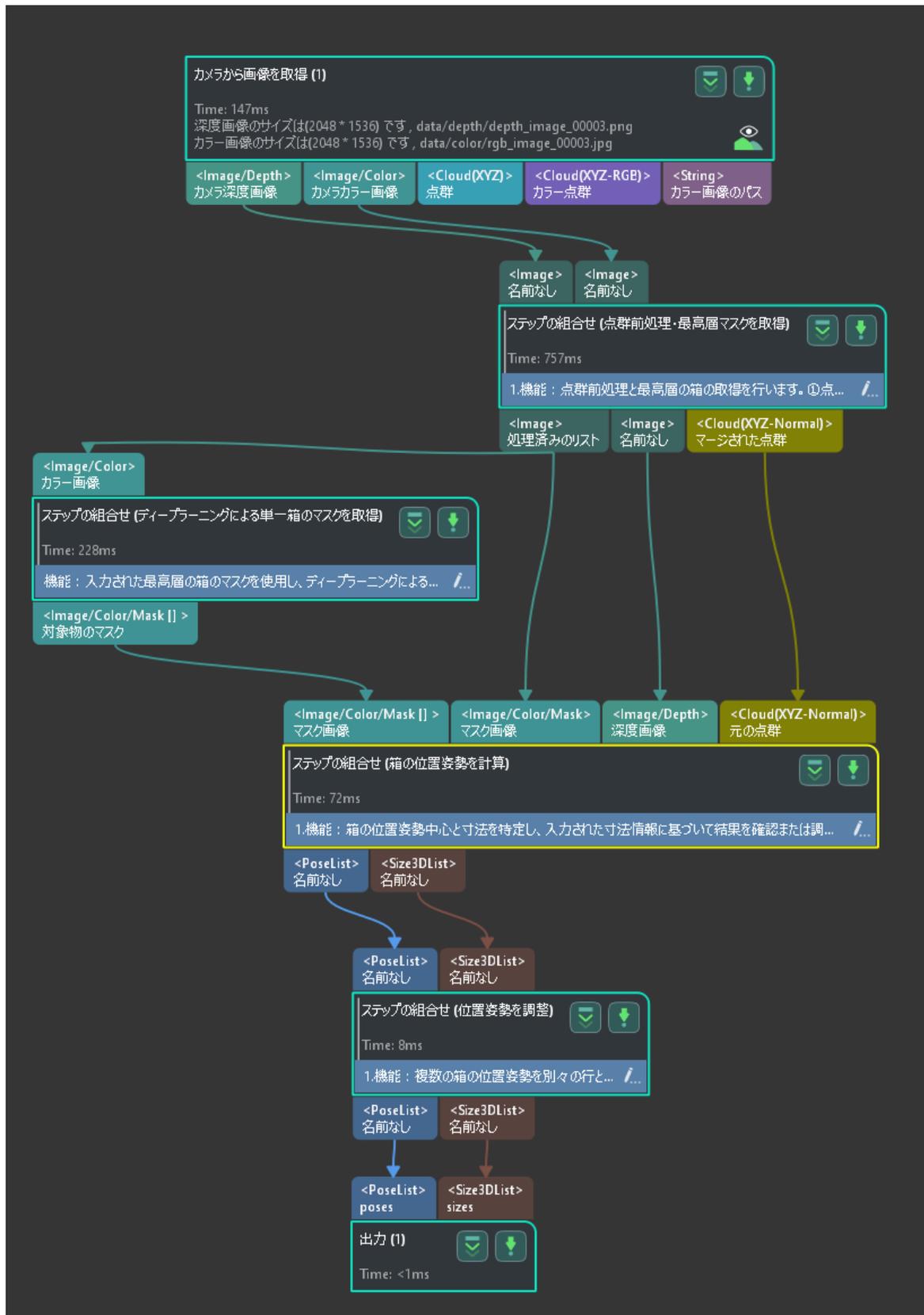
に、作成されたソリューションとプロジェクトが表示されます。



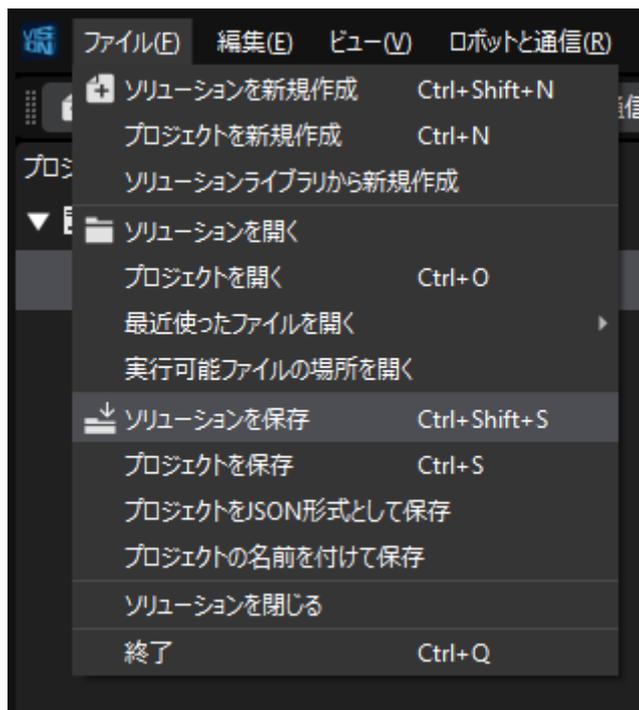
- **ソリューション**とは、ビジョンソリューションを実現するために必要なロボットと通信、ビジョン処理、経路計画などの機能設定とデータの集りのことです。
- **プロジェクト**は、ソリューションにおけるビジョン処理のワークフローです。通常、1つのプロジェクトを含むソリューションで十分ですが、複雑な適用シーンでは複数のプロジェクトが必要になる場合があります。本ガイドに使用されるソリューションでは、1つのプロジェクトしか必要ありません。



メイン画面中央のプロジェクト編集エリアには、「同じ種類の段ボール箱」というプロジェクトが表示されます。

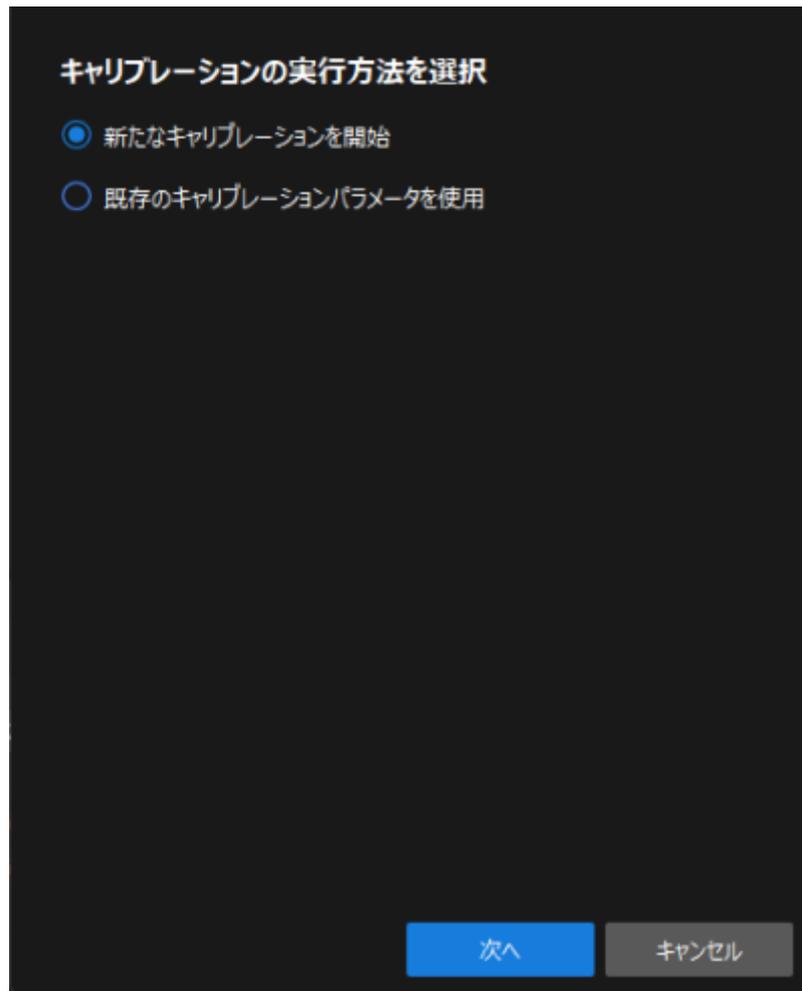


5. メニューバーから **ファイル**、**ソリューションを保存** を選択します。



### キャリブレーション前の設定

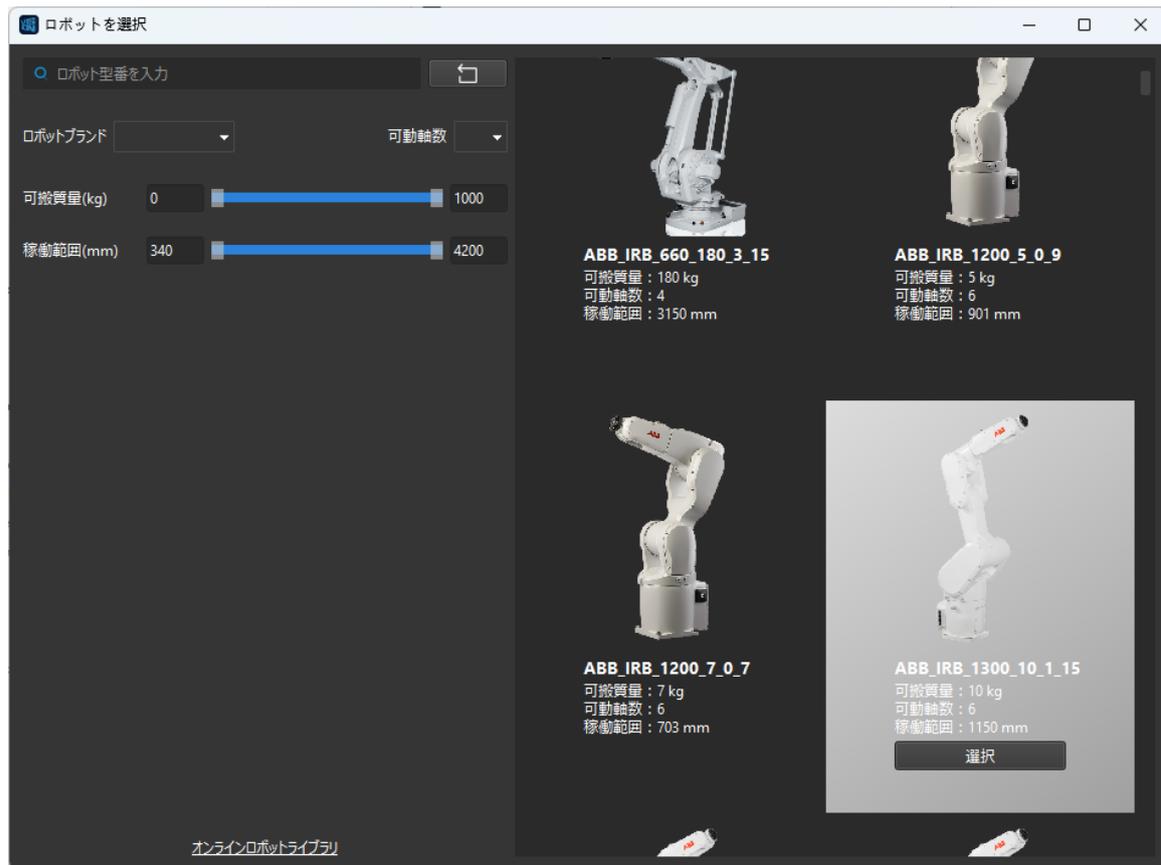
1. Mech-Visionのツールバーで[ **カメラキャリブレーション (標準モード)** ]をクリックします。すると、**キャリブレーション前の設定**画面が表示されます。
2. **キャリブレーションの実行方法**を選択画面で、**新たなキャリブレーションを開始**を選択し、[ **次へ** ]をクリックします。



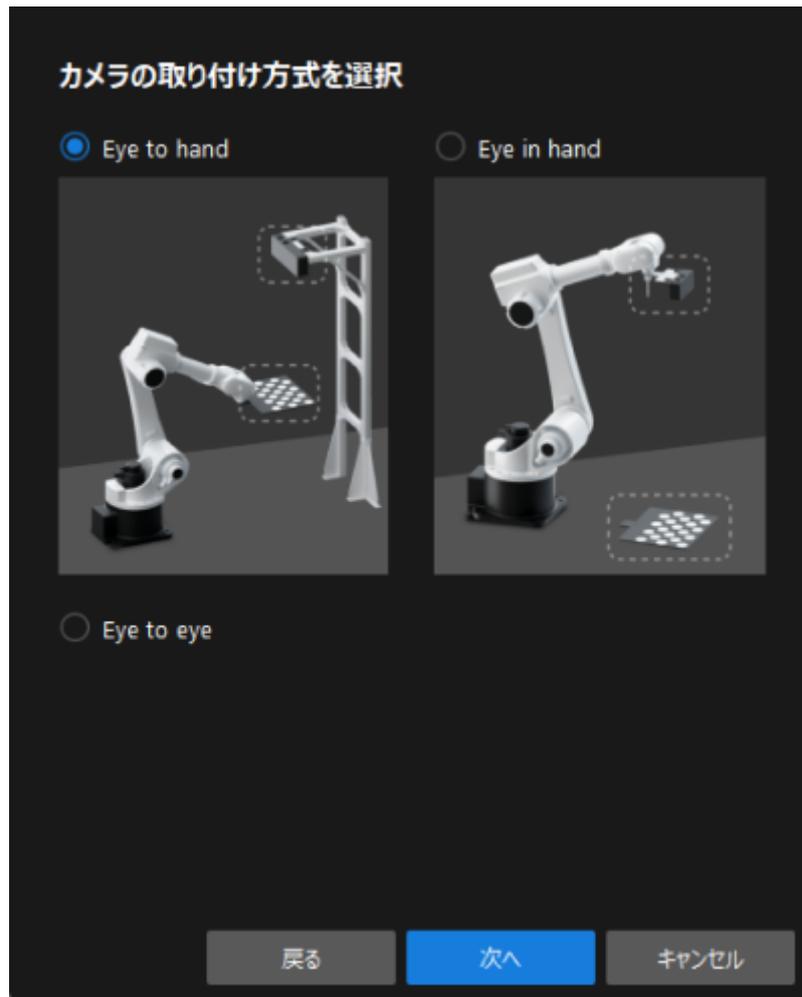
3. キャリブレーションのタスクを選択画面で、ドロップダウンリストボックスから適応可能なロボットのハンド・アイ・キャリブレーションを選択し、[ロボット型番を選択]をクリックします。



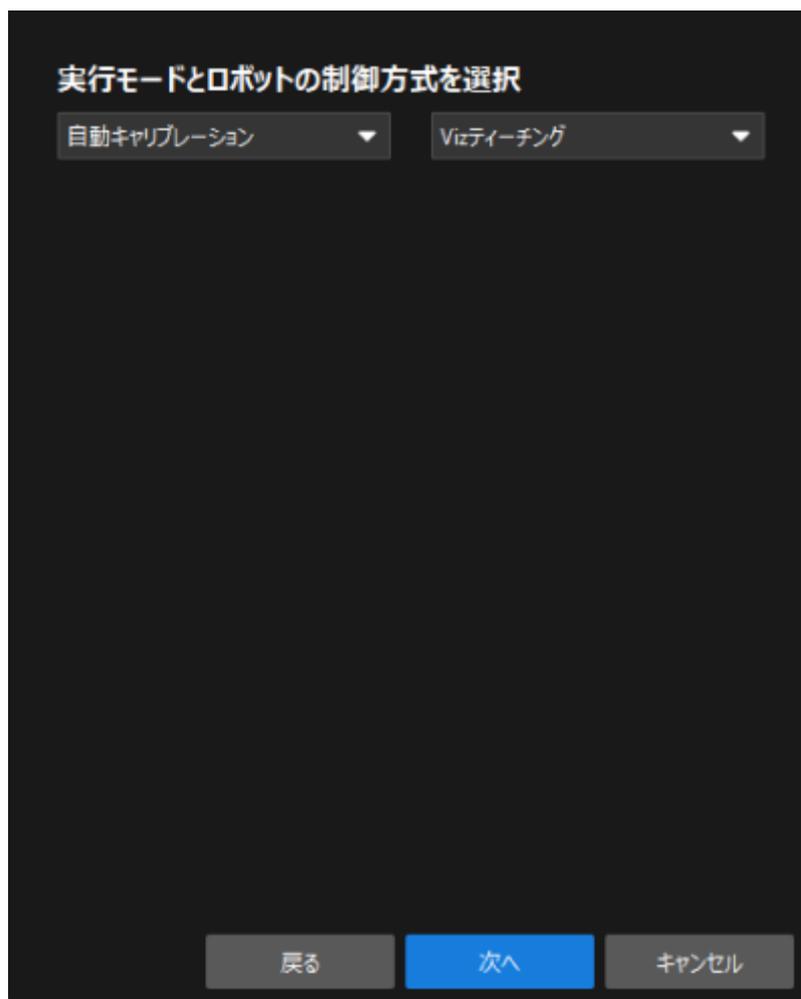
4. ロボットブランドのドロップダウンリストボックスから「ABB」を選択し、右側で「ABB\_IRB\_1300\_11\_0\_9」を選択して[選択]をクリックし、[次へ]をクリックします。



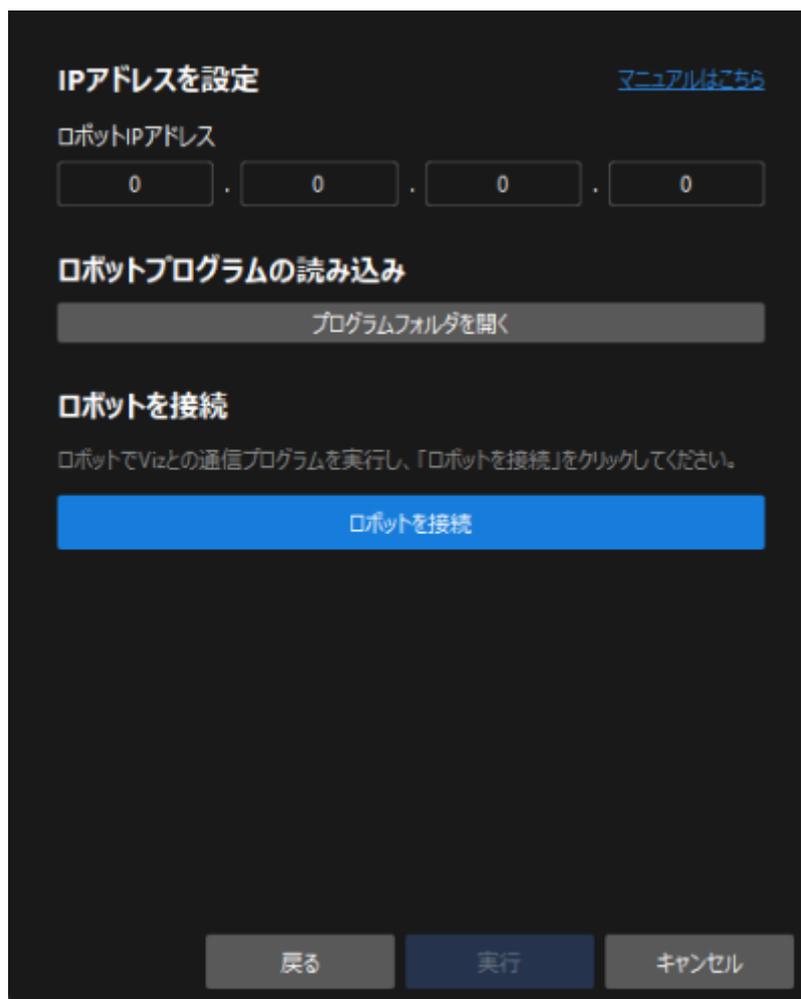
5. カメラの取り付け方式を選択画面で、Eye to Handを選択して[次へ]をクリックします。



6. 実行モードとロボットの制御方式を選択画面で、自動キャリブレーションとVizティーチングを選択して[次へ]をクリックします。



7. 通信設定画面で、ロボットIPアドレスをロボット実機のIPアドレスに設定します。



8. ティーチペンダントで、[MMメインプログラムが起動されていること](#)を確認します。
9. Mech-Visionに戻り、**ロボットを接続**エリアの下に[**ロボットを接続**]をクリックします。このボタンが**ロボットの接続を待機中...**に変わります。
10. **ロボットを接続**の下で「接続済み」が表示されることを確認してから[**実行**]をクリックします。すると、**キャリブレーション (Eye to Hand)** 画面が表示されます。

## キャリブレーションの実行手順

### カメラの接続

1. **カメラに接続手順**で、**検出されたカメラ**から接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。



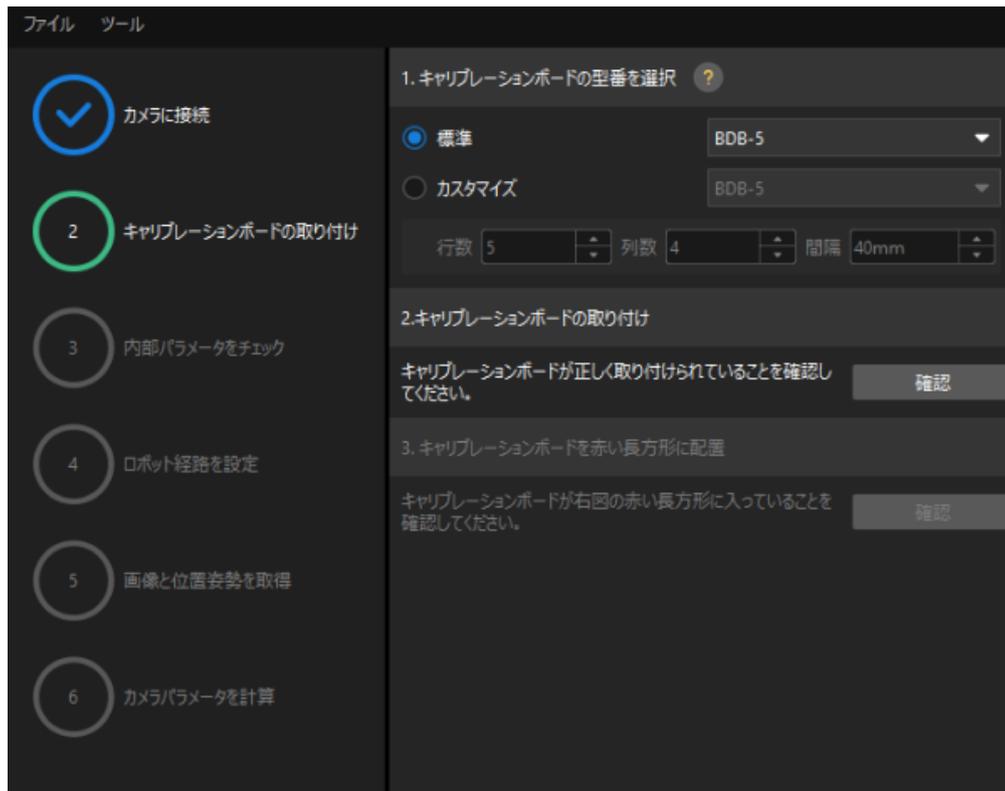
2. カメラ接続後、[一回キャプチャ]または[連続キャプチャ]をクリックします。
3. 画像ビューでは、取得した2D画像と深度画像が要件を満たしていることを確認した上で[次へ]をクリックします。



取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露光パラメータを調整](#)して再取得する必要があります。

### キャリブレーションボードの取り付け

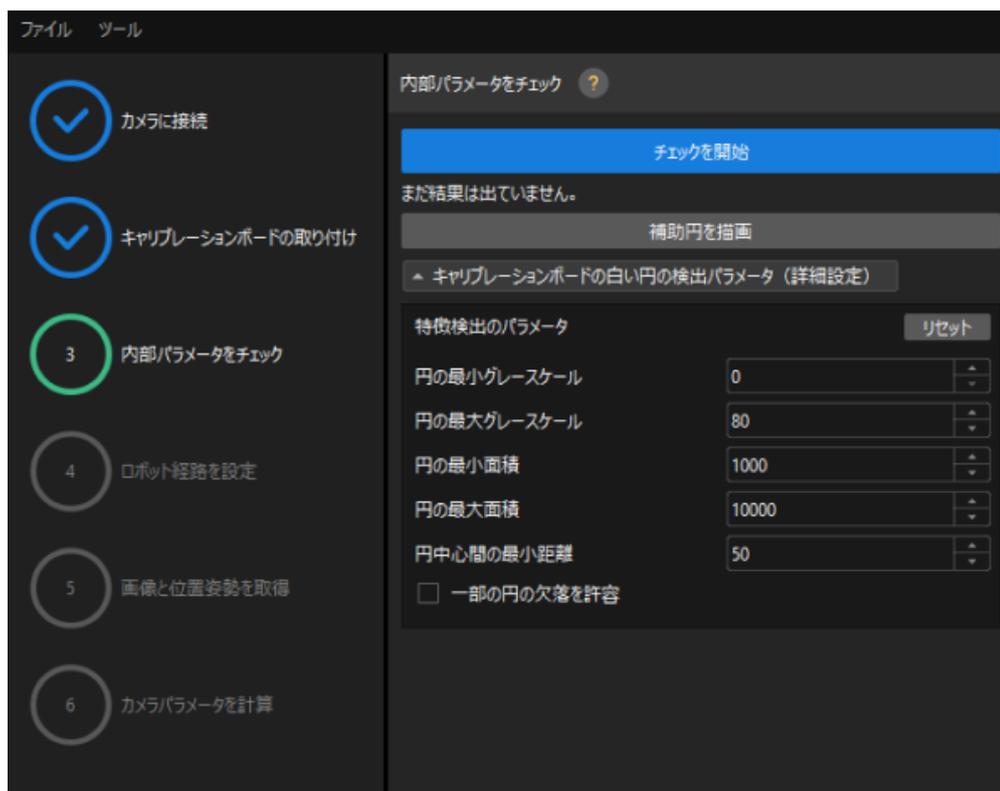
1. キャリブレーションボードの取り付け手順で、**1. キャリブレーションボードの型番を選択で標準**を選択し、実際に使用されるキャリブレーションボードに貼られているラベルに従って型番を選択します。
2. キャリブレーションボードをロボットフランジにしっかりと取り付けたら、**2. キャリブレーションボードの取り付けで[確認]**をクリックします。
3. キャリブレーションボードがカメラの視野中央（赤い長方形内）に配置されることを確認したら、**3. キャリブレーションボードを赤い長方形に配置で[確認]**をクリックします。



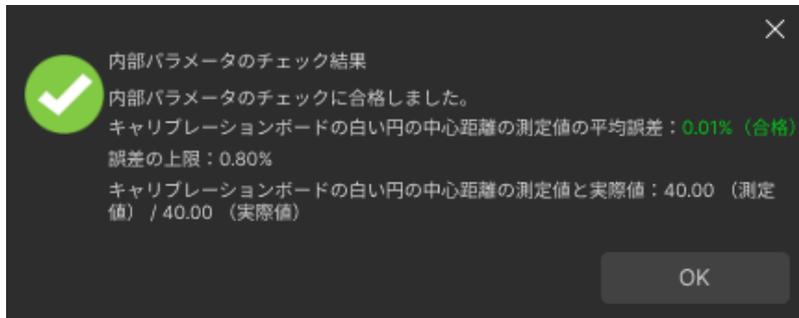
4. キャリブレーションボードに関するすべての操作が完了したら、下部にある[次へ]をクリックします。

### カメラの内部パラメータをチェック

1. 内部パラメータをチェック手順で[チェックを開始]をクリックします。



2. カメラの内部パラメータのチェックが合格したことを確認したら、下部にある[次へ]をクリックします。



内部パラメータのチェックが合格しなかった場合、[補助円を描画するか](#)、[検出パラメータを編集](#)してください。

## ロボット経路を設定

1. **ロボット経路を設定**手順で、**高さ範囲**を設定します。このパラメータは、キャリブレーションボードが深度方向における移動可能な範囲に応じて設定する必要があります。



2. 実際の状況に応じて、**経路のタイプ**をToHandに指定し、ピラミッド型の経路パラメータ下の**高さ範囲**、**層数**、**最下層サイズX/Y**、**最上層サイズX/Y**と**層ごとの移動グリッドの列と行数**を設定し、**回転角度**を設定してから、**[確認]**をクリックします。

**i** | ロボットの経路は、作業領域をカバーするように設定する必要があります。

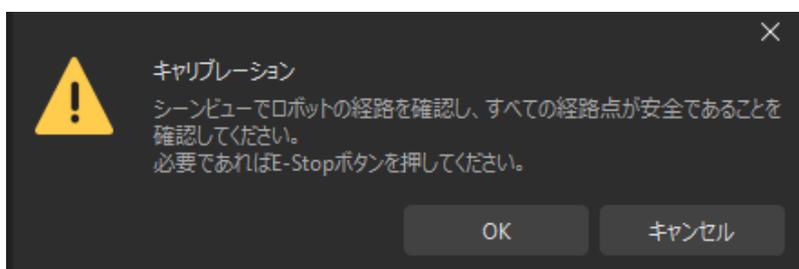
3. 右側の**シーンビュー**では、自動生成された動作経路の各ポイントが周囲と衝突していないことを確認してから、下部にある**[次へ]**をクリックします。

## 画像と位置姿勢を取得

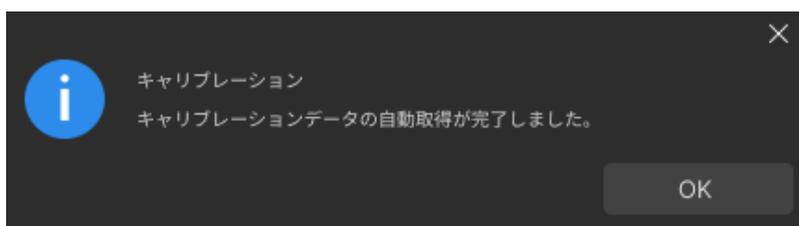
1. **画像と位置姿勢を取得手順**で、**画像を保存**にチェックを入れます。



2. [経路に沿ってロボットを自動で移動させ、画像を撮影]をクリックします。
3. ロボット動作の安全に関する説明をよく読んでから、[OK]をクリックします。

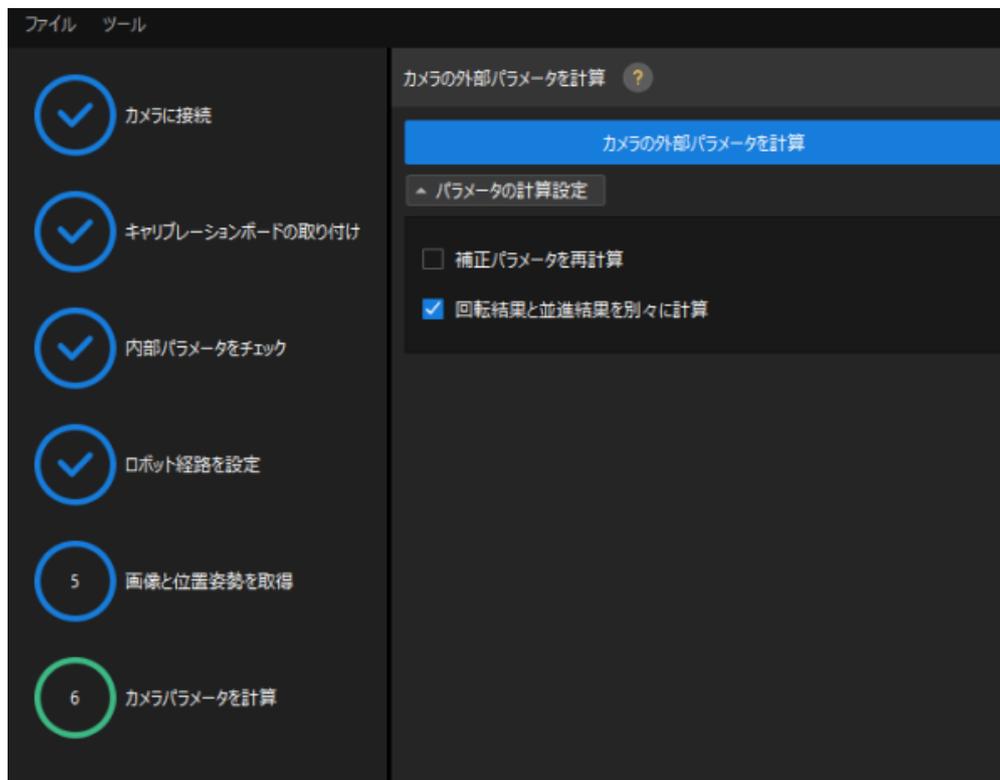


4. ロボットが設定した経路に沿って移動し、カメラが各経路点で画像取得を完了するのを待ちます。右側の画像と位置姿勢一覧には、取得した画像が表示されます。
5. キャリブレーションの自動取得が完了したら、ポップアップウィンドウで[OK]をクリックし、下部にある[次へ]をクリックします。



## カメラパラメータを計算

1. カメラパラメータを計算手順で、[カメラの外部パラメータを計算]をクリックします。



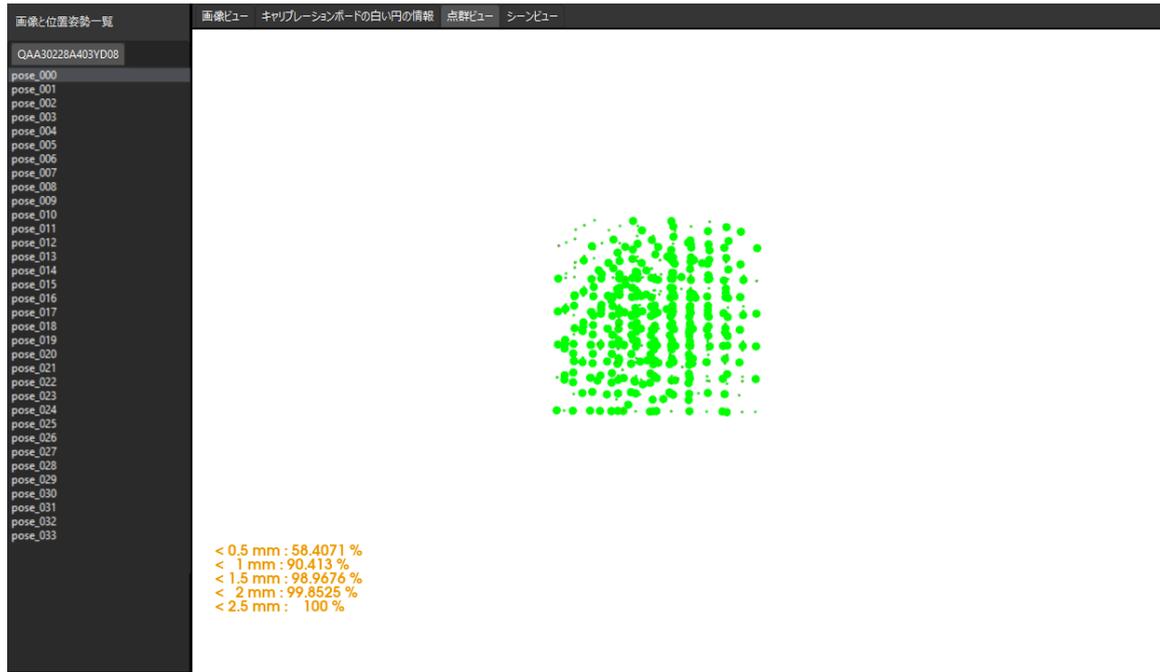
2. キャリブレーションが正常に実行されたことを示すダイアログで、[OK]をクリックします。
3. カメラパラメータを計算の画面でキャリブレーション結果を計算した後、右側の点群ビューパネルでキャリブレーションの誤差点群が表示されます。



誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実際の値と計算値との偏差を示すために使用されます。

4. キャリブレーション精度が要件を満たしていることを確認します。

キャリブレーション精度は、100%を占める誤差の値を求めることで判断できます。例えば、下図は2.5 mm以下の精度を示しています。

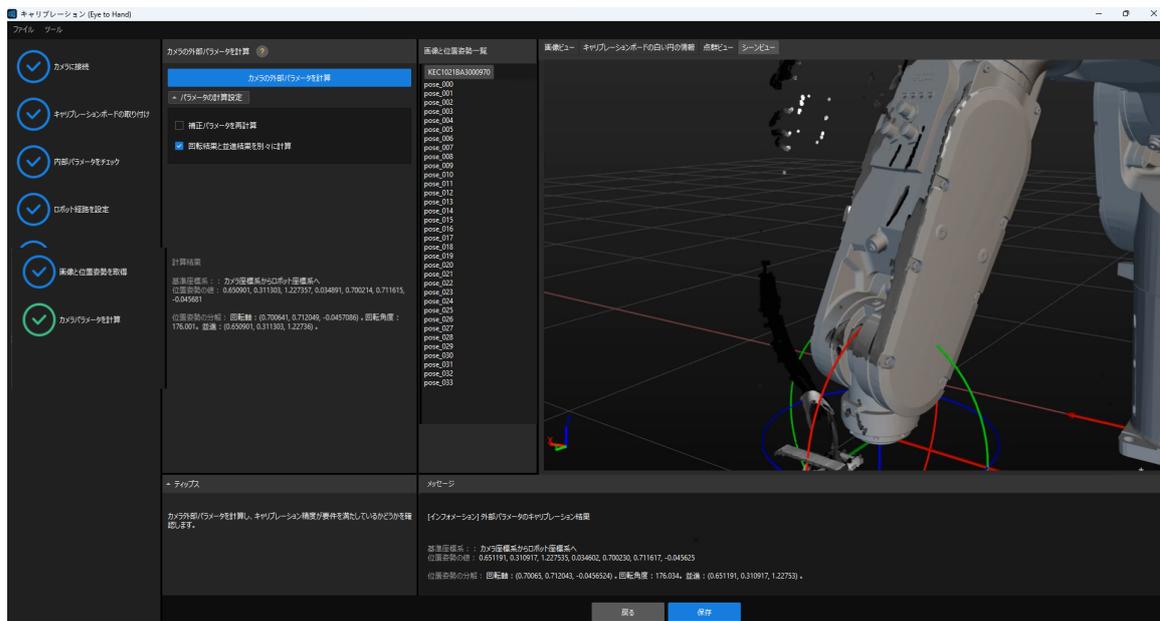


キャリブレーション精度を向上させるには、[キャリブレーション結果の分析](#)をご参照ください。

## キャリブレーション結果の確認と保存

1. キャリブレーションが完了したら、ロボットアームをカメラの視野範囲内に移動させます。
2. カメラパラメータを計算手順で[外部パラメータを再計算]をクリックします。クリックすると、カメラが撮影を行います。
3. シーンビューに切り替え、ロボットの点群がロボットモデルの重なりを確認します。

ロボットの点群がロボットモデルとほぼ一致すれば、キャリブレーションは成功です。



- 下部にある[保存]をクリックし、ポップアップする**キャリブレーションファイル**を保存画面で[OK]をクリックします。すると、カメラのキャリブレーション結果は、プロジェクトの「calibration」ディレクトリに自動的に保存されます。

## 4.4. 段ボール箱の認識

本ガイドを読む前に、「ハンド・アイ・キャリブレーション」を参照し、「同じ種類の段ボール箱」を使用してMech-Visionプロジェクトを作成する必要があります。

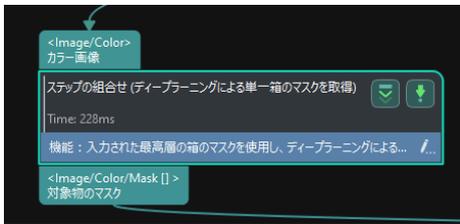
まず、プロジェクトの構築手順を確認し、ステップパラメータの調整とプロジェクト実装により、段ボール箱の位置姿勢を取得してビジョン結果を出力します。

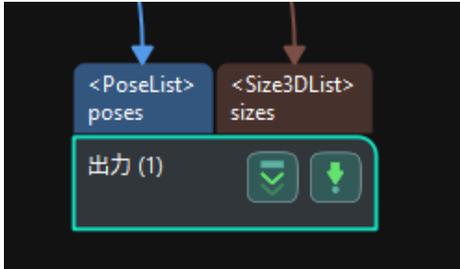
▶ <https://www.youtube.com/watch?v=A-3KNbNCkMY/PLVcMd7cW2rXWbHhTQX8m2R1r6PhX6vrCl> (YouTube video)

動画：段ボール箱の認識

### プロジェクトの構築手順

プロジェクトに使用するステップ・ステップの組合せと機能は下表のとおりです。

| 番号 | ステップ/ステップの組合せ          | サンプル図                                                                                | 機能                                                                                             |
|----|------------------------|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| 1  | カメラから画像を取得             |  | カメラを接続して段ボール箱の画像を取得します。                                                                        |
| 2  | 点群前処理・最高層マスクを取得        |  | 段ボール箱の点群前処理を行い、最高層の箱のマスクを取得します。                                                                |
| 3  | ディープラーニングによる単一箱のマスクを取得 |  | 入力された最高層の箱のマスクを使用し、ディープラーニングによるこの領域内の単一箱のマスクを取得します。これにより、単一箱のマスクを使用して、それが対応する点群を取得することを容易にします。 |

| 番号 | ステップ/ステップの組合せ | サンプル図                                                                               | 機能                                      |
|----|---------------|-------------------------------------------------------------------------------------|-----------------------------------------|
| 4  | 箱の位置姿勢を計算     |   | 箱の位置姿勢を取得し、入力された寸法情報に基づいて結果を確認または調整します。 |
| 5  | 位置姿勢を調整       |   | 箱の位置姿勢の座標系変換を行い、複数箱の位置姿勢を行または行でソートします。  |
| 6  | 出力            |  | ロボット把持のための箱の位置姿勢を出力します。                 |

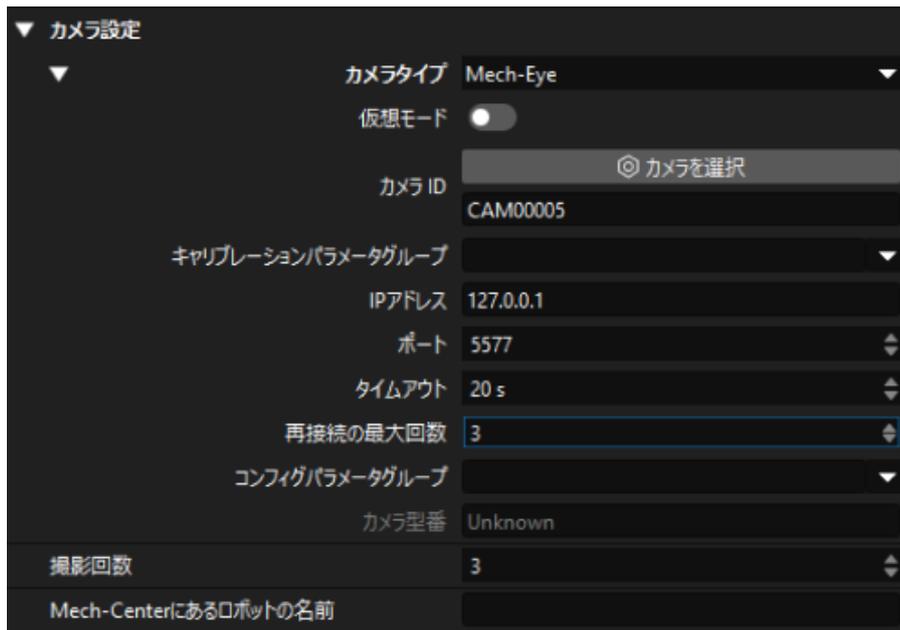
## パラメータの調整説明

以下では、パラメータの調整について説明します。

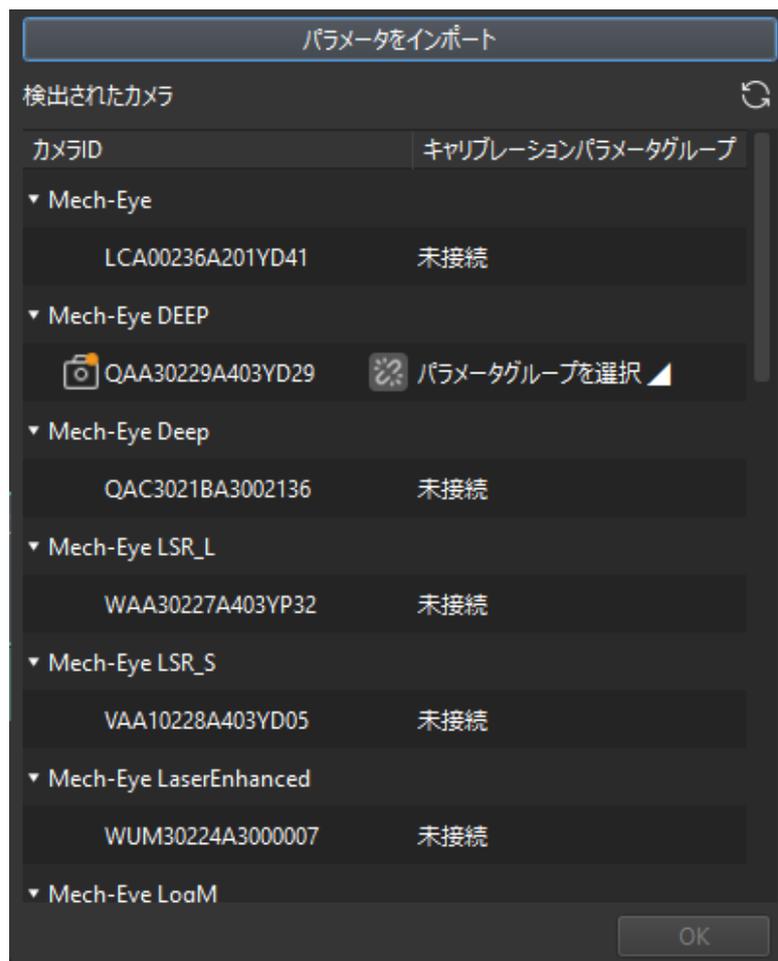
### カメラから画像を取得

「同じ種類の段ボール箱」プロジェクトには仮想データが組み込まれているため、「カメラから画像を取得」のステップパラメータで仮想モードを終了し、カメラ実機に接続する必要があります。

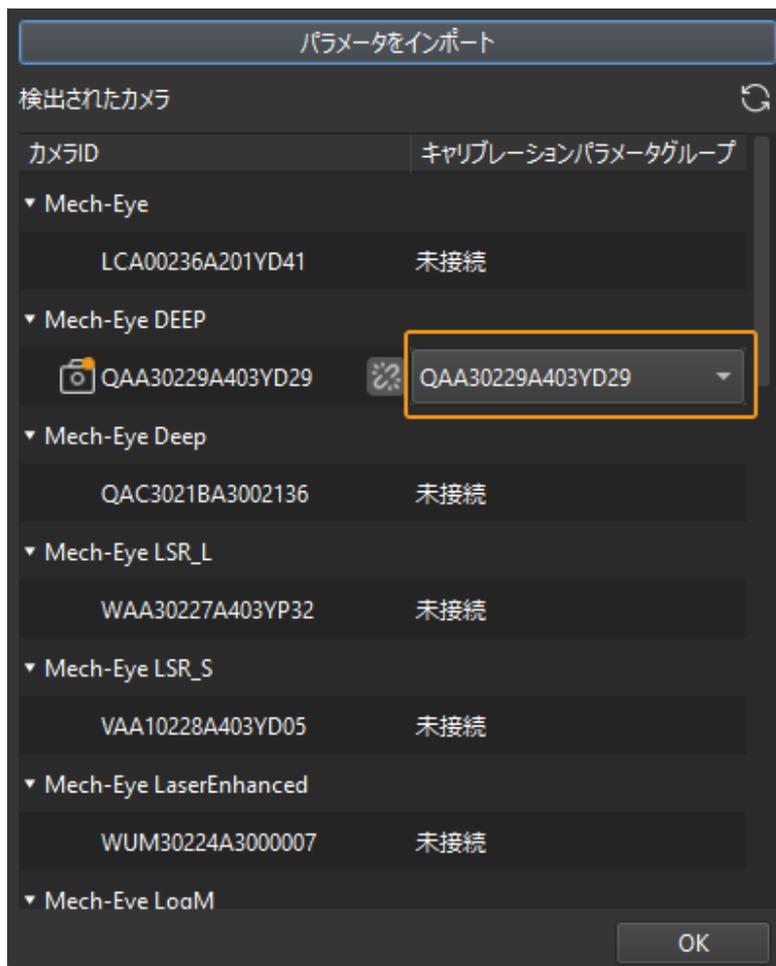
1. 「カメラから画像を取得」をクリックして選択し、ステップパラメータで**仮想モード**をオフにして、**[カメラを選択]**をクリックします。



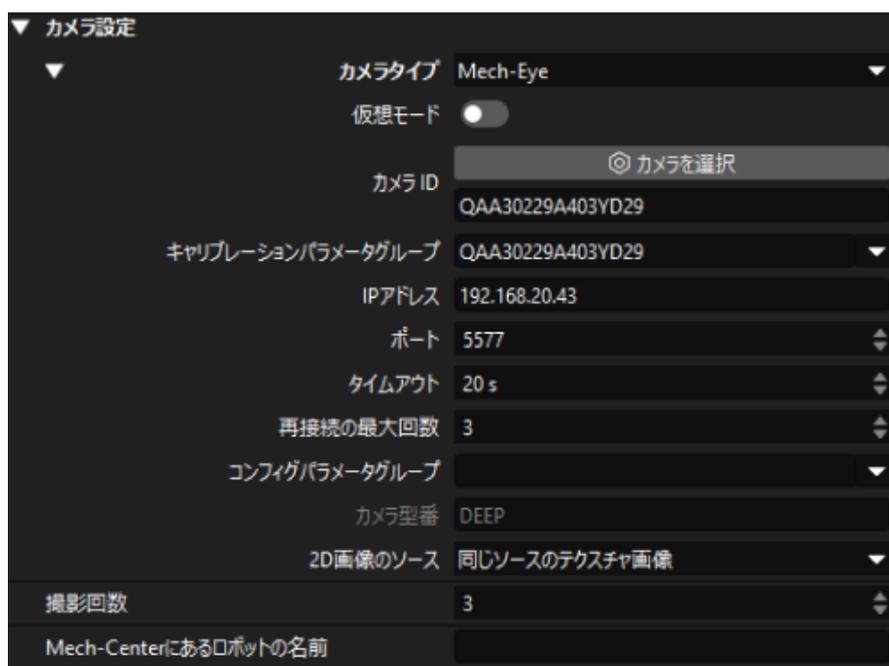
2. 表示される画面でカメラIDの右側にある  をクリックすると、カメラの接続は完了です。カメラが正常に接続された場合、 は  に変わります。



カメラ接続後、**[パラメータグループを選択]** をクリックし、キャリブレーション済みのパラメータグループを選択します。



3. カメラを接続し、パラメータグループを設定すると、キャリブレーションパラメータグループ、IPアドレス、ポートなどのパラメータが自動的に入力されます。それ以外のパラメータ設定は不要です。



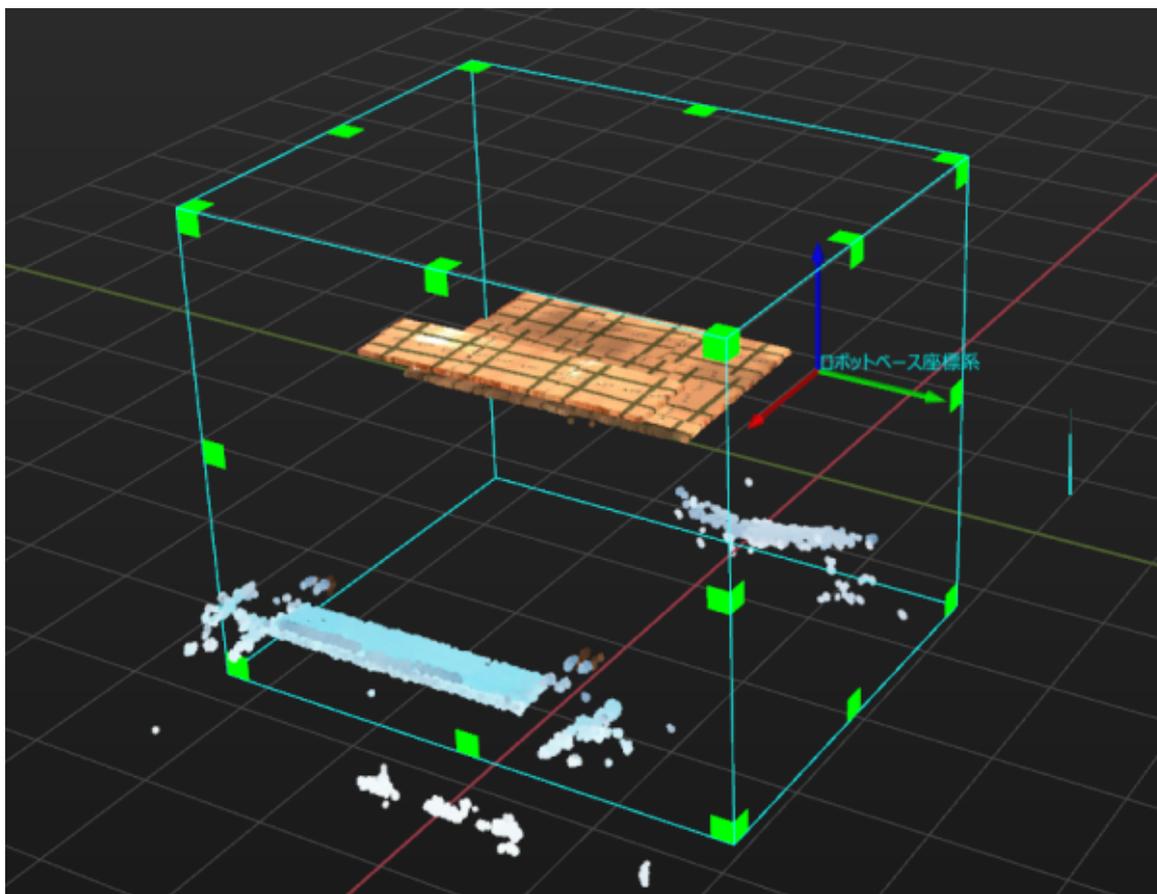
これで、カメラの接続が完了しました。

### 点群前処理・最高層の箱を取得

最高層の箱以外の箱を把持することによるロボットと他の箱との衝突を避けるために、このステップ組合せにより、最高層の箱を取得し、ロボットが最高層の箱を優先的に把持させる必要があります。

このステップ組合せでは、**3D ROI**と**層の高さ**の設定が必要です。

1. ステップパラメータで [**3D ROIを設定**] をクリックします。表示される設定画面で **3D ROI** の設定を行います。



パレットに積みされている箱の高さの上限と下限が3D ROI内に収まる必要があります。また、3D ROIには干渉点群を含まないことも必要です。

2. 下層の箱を認識しないように**層の高さ**を設定する必要があります。層の高さは、単一箱の高さより小さく設定する必要があります。例えば、箱の高さの半分に設定します。通常は推奨値を使用します。

現場では、パレットに積みされている箱の寸法が異なる場合、**層の高さ**は箱の高さの最小値に従って設定されるべきです。



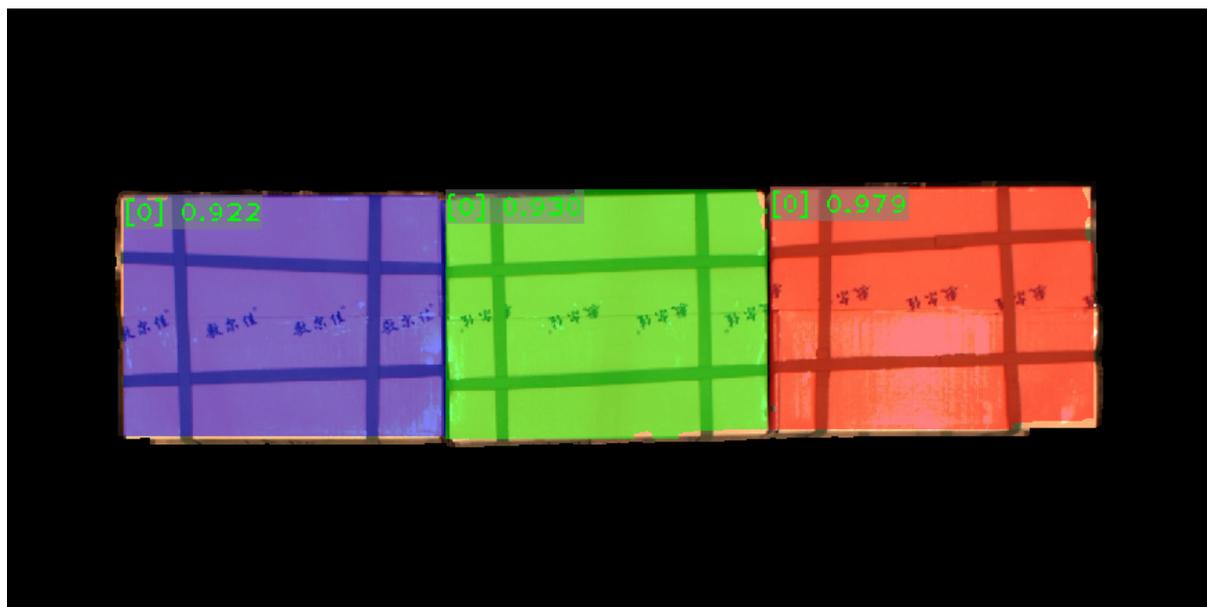
**層の高さ**を正しく設定しないと、最高層以外の箱を認識することになり、ロボットが

把持する際に他の箱と衝突する可能性があります。

## ディープラーニングによる単一箱のマスクを取得

最高層の箱の認識が完了した後、ディープラーニングによる各箱の認識に進みます。

本ガイドで使用されているソリューションには、様々な段ボール箱に適応可能なインスタンスセグメンテーションのモデルが組み込まれています。このステップの組合せを実行すると、下図のような各箱のマスクが取得されます。



結果が要件を満たさない場合は、3D ROIを再設定する必要があります。

## 箱の位置姿勢を計算

各箱の点群を取得した後、箱の位置姿勢の計算が可能になります。また、ビジョン処理による認識の精度を確保するために、箱の寸法を入力します。

「箱の位置姿勢を計算」ステップの組合せでは、**X、Y、Z軸における長さを箱の寸法の誤差許容範囲**を設定します。

- **X、Y、Z軸における長さ**：実際に使用される箱の寸法に応じて設定します。
- **箱の寸法の誤差許容範囲**：初期値30mmを使用します。入力された箱の寸法が認識された箱の寸法と大きく異なる場合は、このパラメータを変更する必要があります。

## 位置姿勢を調整

箱の位置姿勢を取得した後、ロボットが正常に把持できるように、ワークの位置姿勢をカメラ座標系からロボット座標系に変換する必要があります。

また、このステップの組合せにより、位置姿勢を行または列でソートし、ロボットの順次把持を実現できます。

- 昇順（ロボットベース座標系における箱位置姿勢のX値による）：通常は初期値（チェックを入れる）を使用します。チェックを入れると、行はロボットのベース座標系における箱位置姿勢のX値に従って昇順にソートされ、そうでない場合は降順にソートされます。
- 昇順（ロボットベース座標系における箱位置姿勢のY値による）：通常は初期値（チェックを入れる）を使用します。チェックを入れると、列はロボットのベース座標系における箱位置姿勢のY値に従って昇順にソートされ、そうでない場合は降順にソートされます。

## 出力

正確な位置姿勢を取得した後、プロジェクトの結果をバックグラウンドに送信するための「出力」のステップが必要です。

これで、段ボール箱を認識するためのプロジェクトの設定が完了です。

## 4.5. 把持と配置を実行

Mech-Visionソリューションを使用して段ボール箱の位置姿勢を取得した後、ロボットが段ボール箱の把持・配置を繰り返し実行するためにMech-Vizプロジェクトの構築が必要です。

▶ <https://www.youtube.com/watch?v=pL4ZbeoDF8A/PLVcMd7cW2rXWbHhTQX8m2R1r6PhX6vrCl> (YouTube video)

動画：把持と配置

今回作成したプロジェクトは下図のような流れで処理を行います。



### ロボットとシーンの設定

ロボットの把持・配置時に周囲の対象物との衝突を避けるため、衝突検出のためのロボットハンドモデルとシーンモデルをプロジェクトに追加する必要があります。設定手順は下図に示します。



### ロボットハンドモデルのインポートと設定



- ロボットハンドとは、ロボット先端に取り付けられ、対象物を扱う/把持するための装

置を指します。

- 本ガイドでは、1つグループの吸盤を搭載したロボットハンドを使用しています。

ロボットハンドのインポートと設定を行う目的は、ロボットハンドのモデルを仮想空間に表示し、衝突検出に利用できるようにするためです。

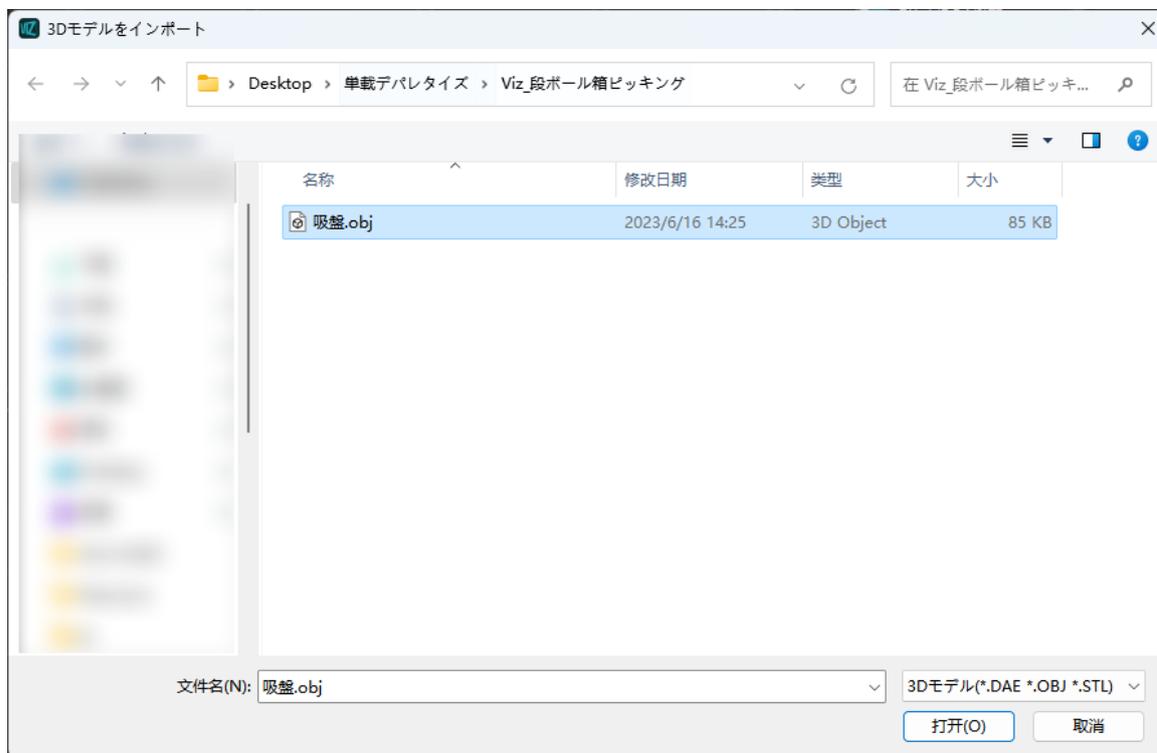
### ロボットハンドのモデルのインポート

1. **リソース** > **モデルライブラリ**の[+]をクリックします。



リソースとは、ロボットやロボットハンド、対象物、シーンの物体など、プロジェクトを構築するための要素を指します。

2. 表示される画面でOBJ形式の衝突モデルのファイルを選択して[開く]をクリックします。

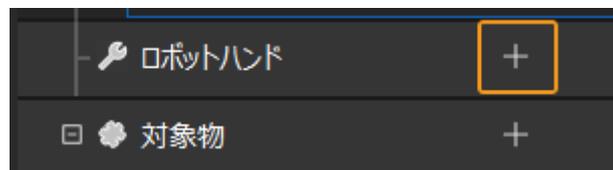


3. モデルのインポートが完了すると、モデルライブラリでインポートしたモデルが表示できるようになります。

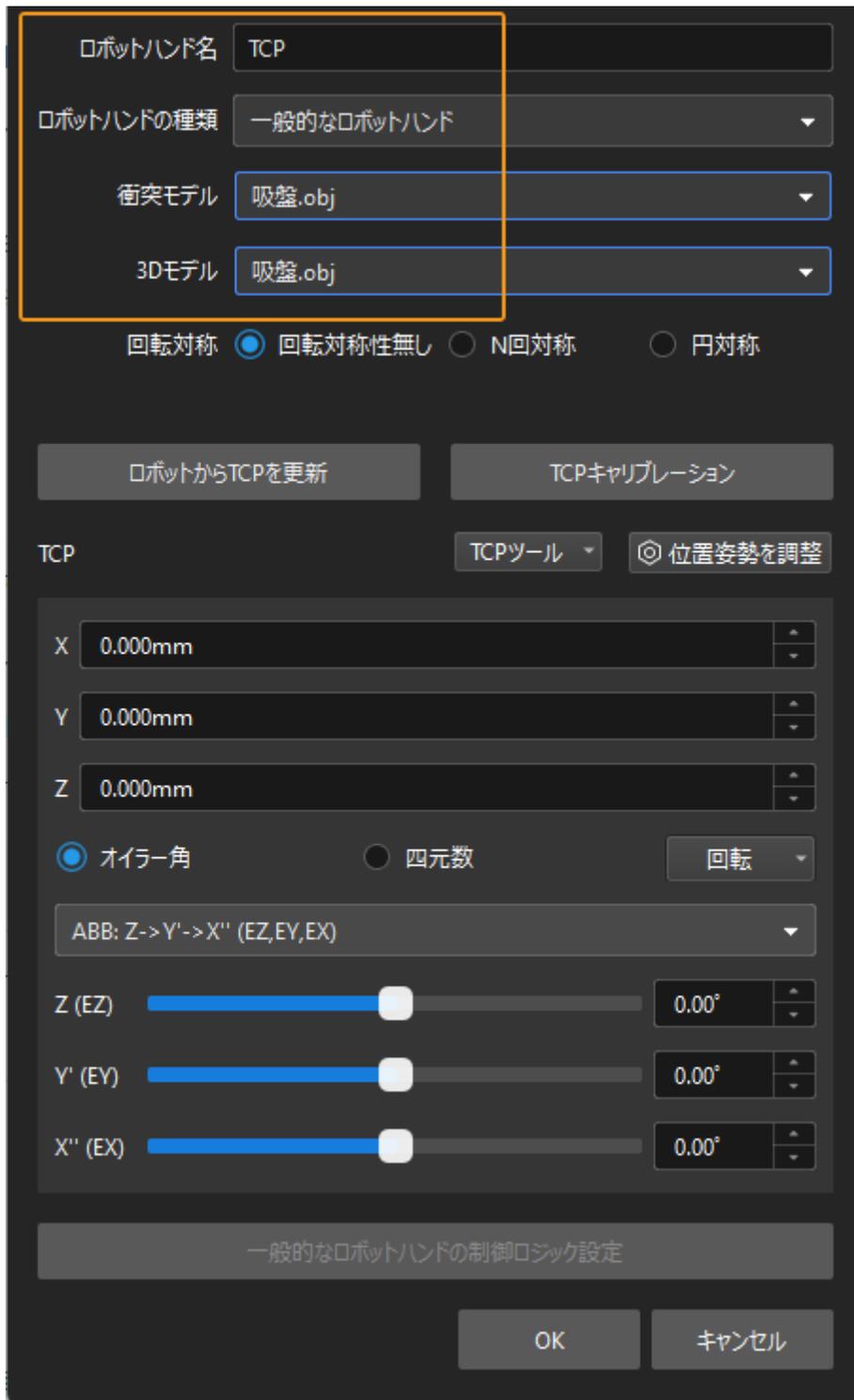


### ロボットハンドモデルの設定

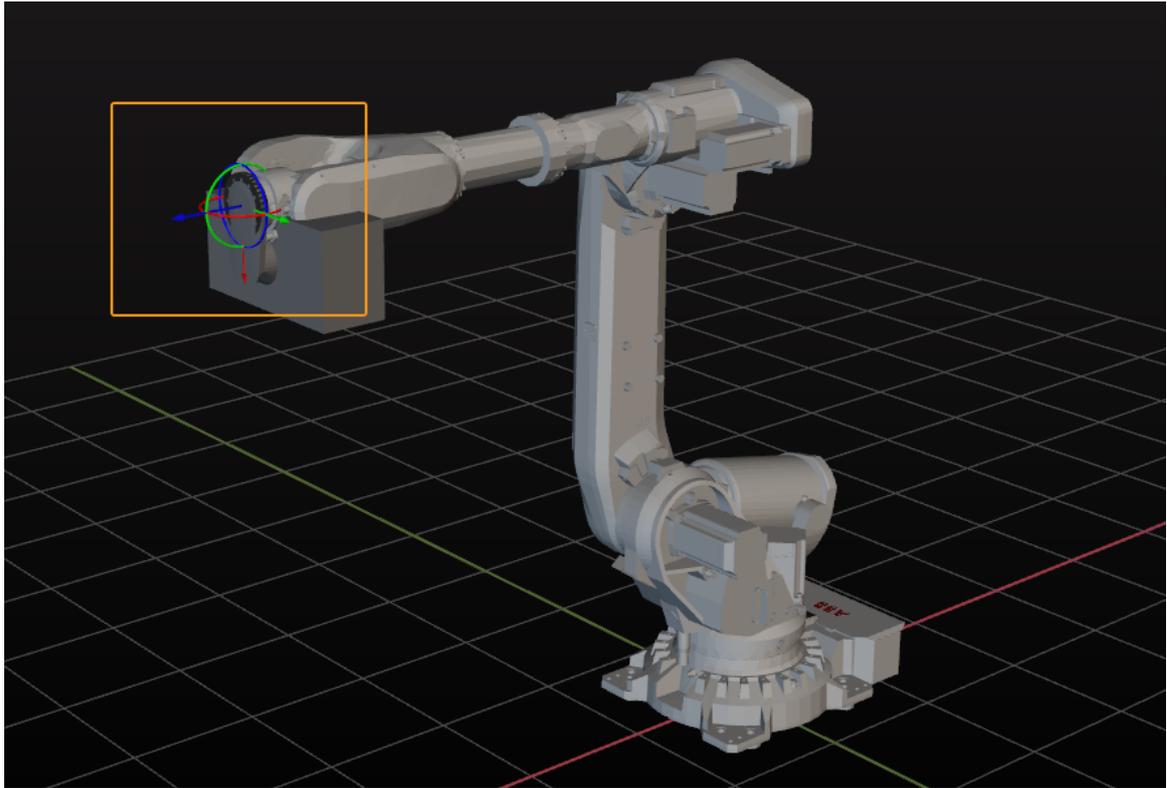
1. リソース > ロボットハンドの[+]をクリックします。



2. 表示される画面にロボットハンドの名前を入力し、インポートしたロボットハンドのモデルファイルを衝突モデルと3Dモデルとして使用し、[OK]をクリックします。



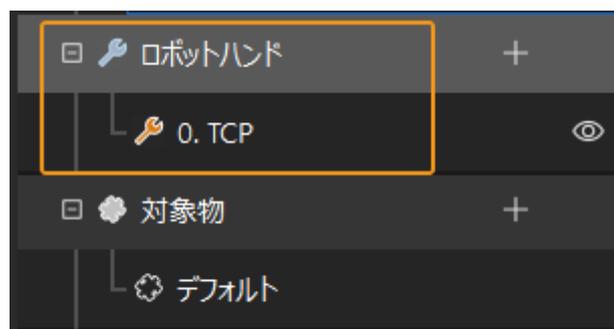
3. ロボットハンドが設定されると、下図のように、設定されたロボットハンドが仮想空間に表示できるようになります。



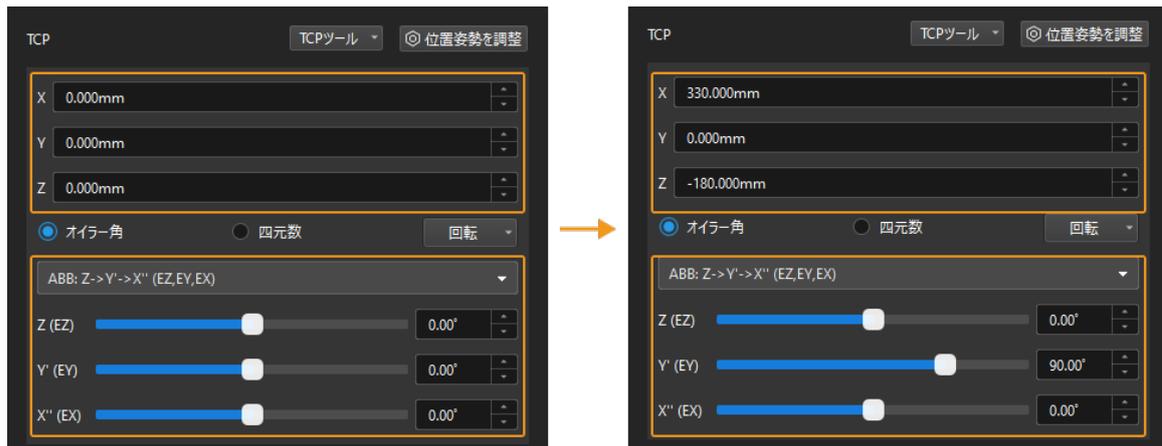
### ロボットハンドの調整

上図では、ロボットに対するロボットハンドの位置が間違っていますので、ロボットハンドの位置調整が必要です。

1. モデルライブラリ内のロボットハンドモデルをダブルクリックします。

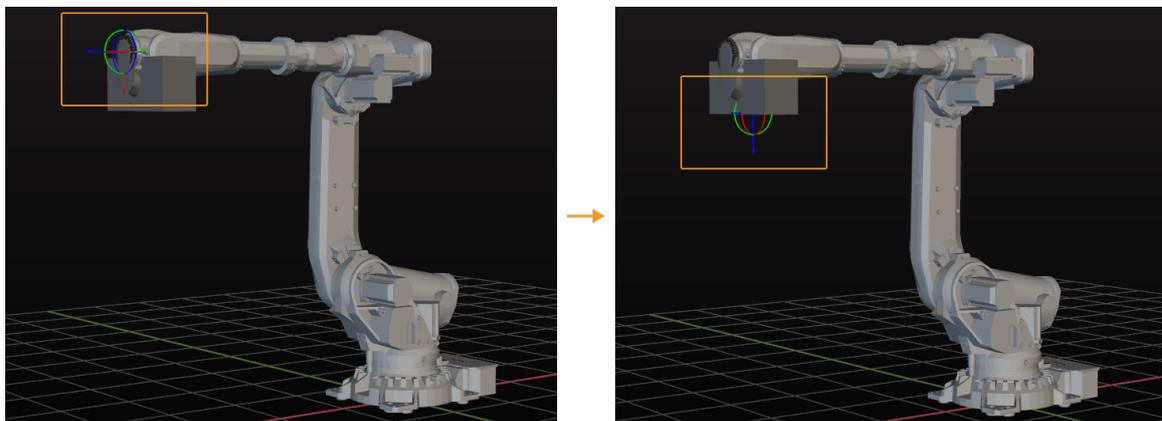


2. 表示されるモデル設定画面でTCP（ツール中心点）を調整します。



TCPは通常、ロボットハンドの先端にある点のことを指します。本ガイドでは、TCPは吸盤の中心点となります。

3. 調整前と調整後のロボットハンドの位置を下图に示します。



## シーンモデルのインポートと設定

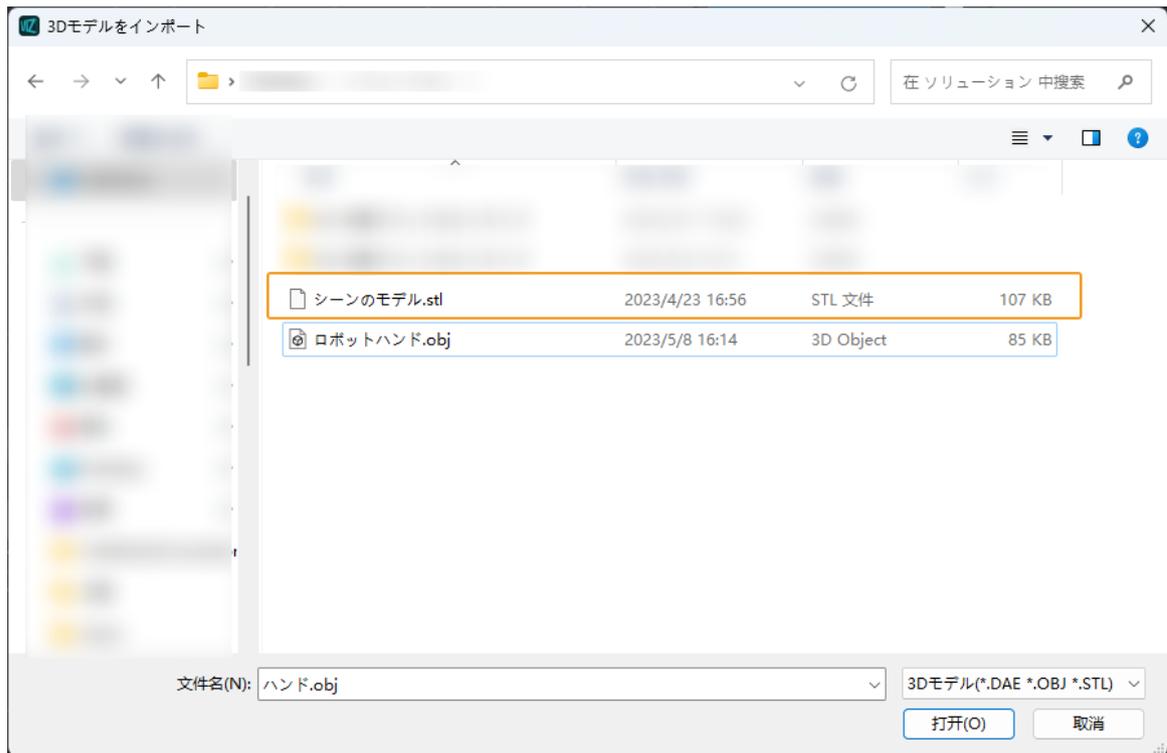
シーンモデルをインポートして設定する目的は、実際の作業現場を再現することで、ロボットの動作経路を計画できるようにすることです。

### シーンモデルのインポート

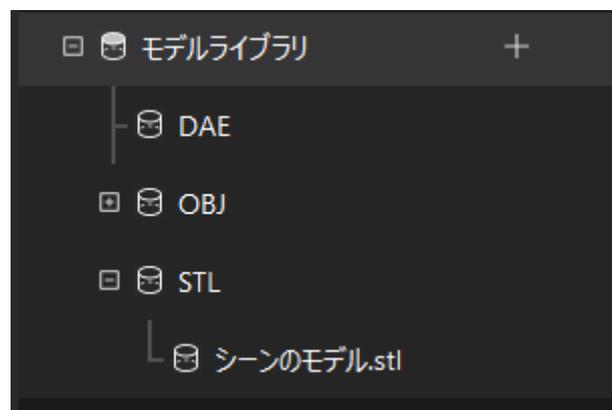
1. リソース > モデルライブラリの[+]をクリックします。



2. 表示される画面でシーンの物体のモデルファイルを選択して[開く]をクリックします。

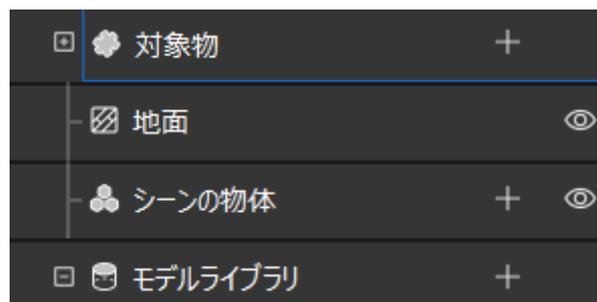


3. モデルのインポートが完了すると、モデルライブラリでインポートしたモデルが表示できるようになります。



### シーンモデルの設定

1. リソース > シーンの物体の[+]をクリックします。



2. 表示される画面に物体の名前を入力し、「外部モデル」を「シーンのモデル」とし、インポ

ートしたシーンの物体のモデルファイルを衝突モデルと3Dモデルとして使用し、[OK]をクリックします。

物体設定 物体の位置姿勢

物体名  
シーンのモデル

シーンモデル  
外部モデル

衝突モデル  
シーンのモデル.stl

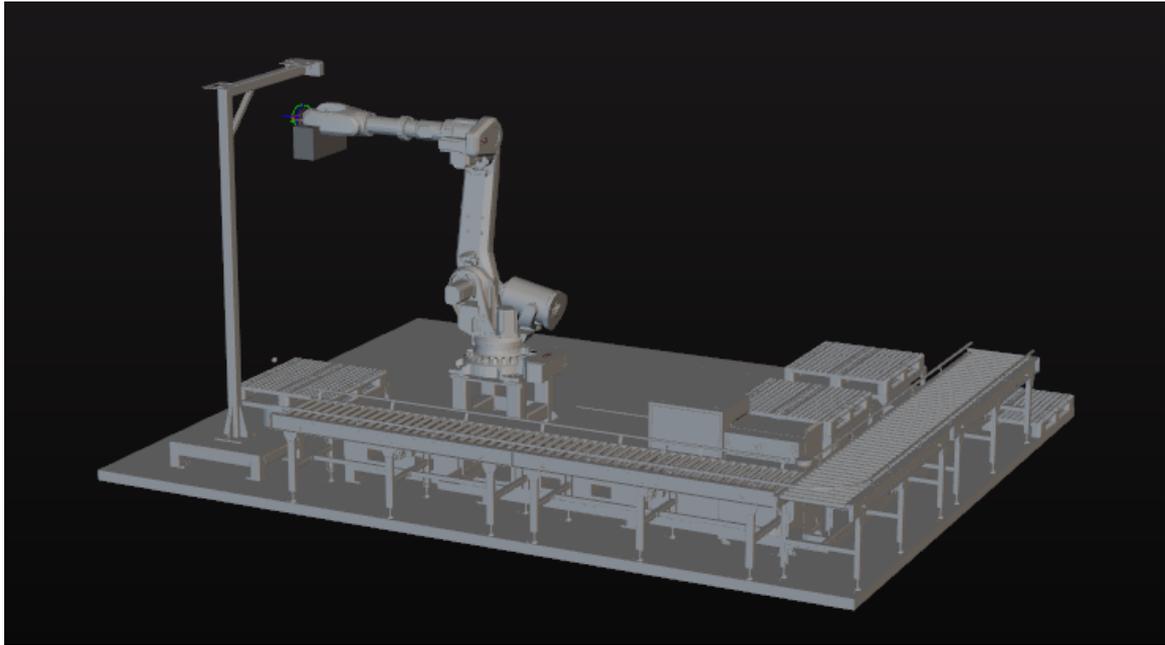
3Dモデル  
シーンのモデル.stl

衝突検出に使用する

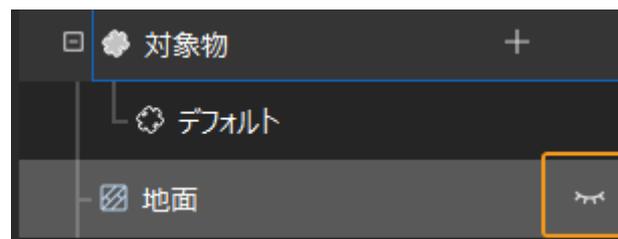
モデル選択可能

OK キャンセル

3. 設定されると、シーンの物体が仮想空間に表示できるようになります。



シーンモデルのみを表示する場合は、地面を非表示に設定することができます。



上記のモデルの設定が完了後、ツールバーで[ **ロボットを同期させる** ]をクリックし、仮想ロボットとロボット実機の姿勢を一致させます。

## ワークフローの構築

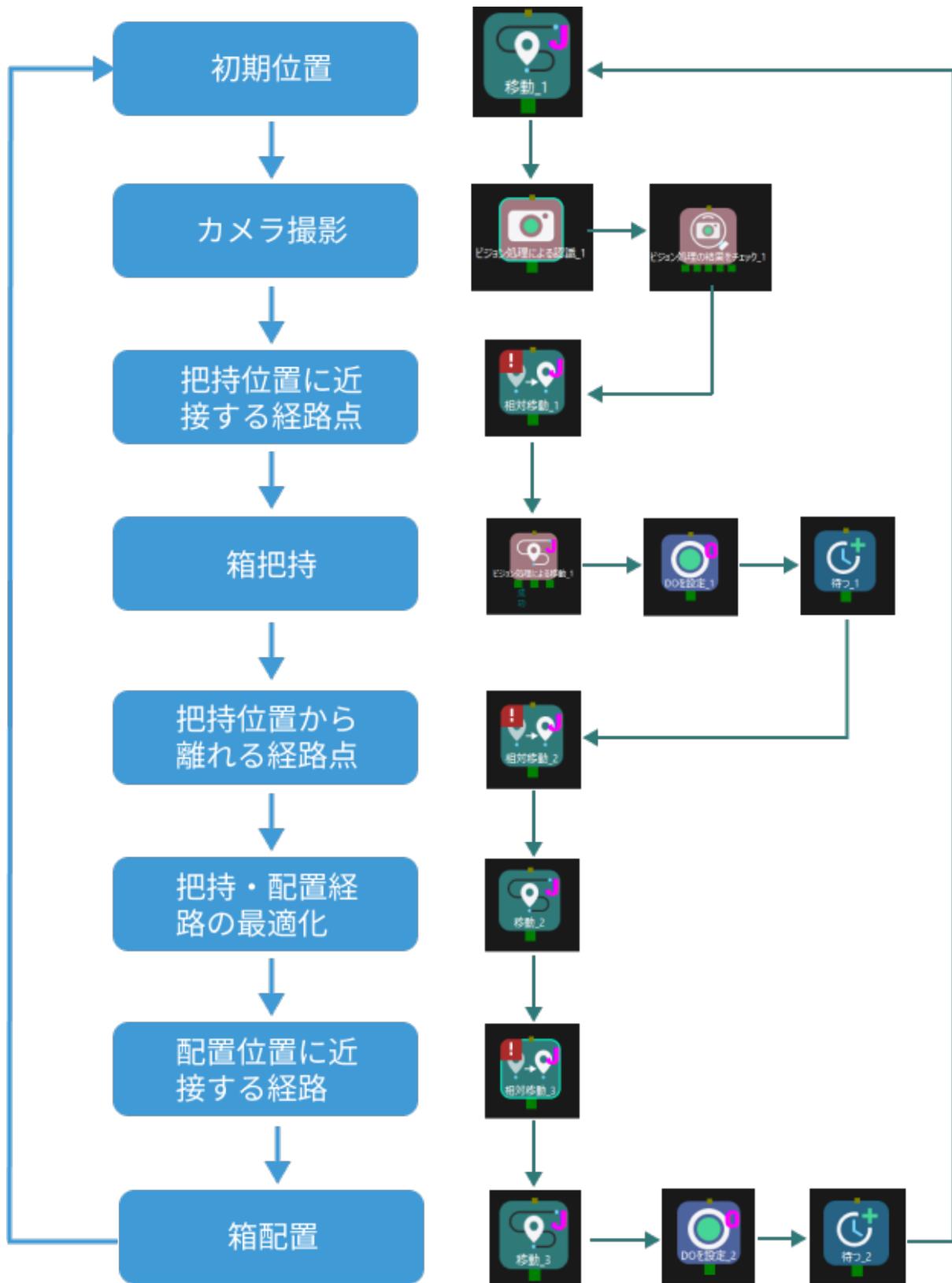
モデルの設定が完了したら、ワークフローの構築を行います。プリセットプログラムの機能は、ステップライブラリからプロジェクト編集エリアにステップをドラッグし、ステップパラメータを設定し、ステップをつなぐことで実現します。



- ワークフローとは、Mech-Vizでフローチャートの形式で作成されたロボット動作制御プログラムです。
- ステップとは、ロボットをプログラミングするための機能モジュールです。

### ワークフローの構築手順

下図のような手順でワークフローの構築を行います。



### 「初期位置」に移動

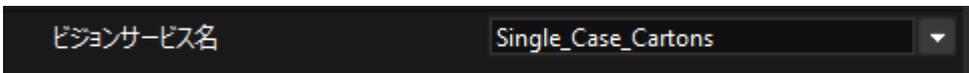
初期位置はロボット動作の開始位置だけでなく、安全な位置でもあります。この位置で、ロボットが把持する対象物や周囲の機器から離れ、カメラの視野を遮らないようにする必要があります。

ます。

ロボットをカスタイズされた初期位置に移動させた後、ステップライブラリから**移動**を選択し、プロジェクト編集エリアにドラッグして「初期位置」という名前を付けます。ツールバーの[**ロボットを同期させる**]をクリックし、現在ロボットにいる位置姿勢を記録します。



### 画像取得とビジョン処理を実行

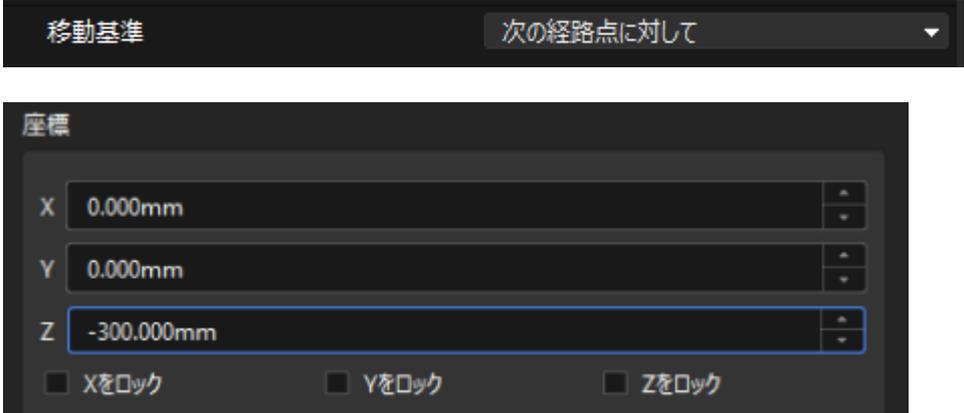
|         |                                                                                      |
|---------|--------------------------------------------------------------------------------------|
| ステップ    | ビジョン処理による認識                                                                          |
| 説明      | Mech-Visionプロジェクトを実行し、ビジョン結果を取得します。                                                  |
| 操作      | ステップライブラリから「ビジョン処理による移動」を選択し、プロジェクト編集エリアにドラッグします。                                    |
| パラメータ設定 | <b>ビジョンサービス名</b> のドロップダウンリストで <b>Single_Case_Cartons</b> を選択します。                     |
| サンプル図   |  |

ビジョン結果があるかどうかを確認するために、「ビジョン処理による移動」の後に**ビジョン処理の結果をチェック**を追加します。

|         |                                                      |
|---------|------------------------------------------------------|
| ステップ    | ビジョン処理の結果をチェック                                       |
| 説明      | ビジョン結果があるかどうかを確認します。                                 |
| 操作      | ステップライブラリから「ビジョン処理の結果をチェック」を選択し、プロジェクト編集エリアにドラッグします。 |
| パラメータ設定 | パラメータの調整は不要です。                                       |

### 把持位置に近接する経路点に移動

段ボール箱を把持する際、ロボットが初期位置から把持位置まで直接移動すると、段ボール箱と衝突する可能性があります。したがって、ビジョン結果を取得した後、**相対移動**ステップを使用してロボットを経路点に近接する位置に移動させます。

|         |                                                                                            |
|---------|--------------------------------------------------------------------------------------------|
| ステップ    | 相対移動                                                                                       |
| 説明      | ビジョン結果に従って移動します。                                                                           |
| 操作      | ステップライブラリから「相対移動」を選択し、プロジェクト編集エリアにドラッグして「近接する経路点1」という名前を付けます。                              |
| パラメータ設定 | <b>移動基準</b> を次の経路点に対してに設定し、 <b>目標点タイプ</b> をロボットハンドに設定し、 <b>Z軸</b> の座標を-300mmなどの適切な値に設定します。 |
| サンプル図   |          |

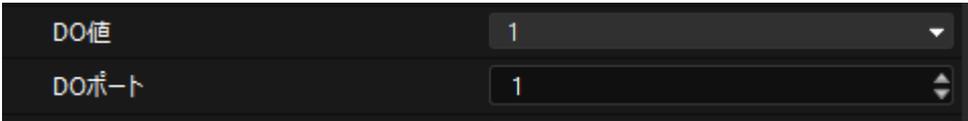
### 段ボール箱の把持

ロボットが把持位置に近接する経路点に移動すると、次の2つの手順で箱を把持できます。

1. 手順1：**ビジョン処理による移動**ステップを使用して、ロボットを段ボール箱の把持位置に移動させます。
2. 手順2：**DOを設定**ステップを使用して、ロボットが吸盤を通して段ボール箱の把持をさせます。

詳細は以下の通りです。

|         |                                                                                      |
|---------|--------------------------------------------------------------------------------------|
| ステップ    | ビジョン処理による移動                                                                          |
| 説明      | ビジョン結果に従って移動します。                                                                     |
| 操作      | ステップライブラリから「ビジョン処理による移動」を選択し、プロジェクト編集エリアにドラッグします。                                    |
| パラメータ設定 | <b>ビジョンサービス名</b> のドロップダウンリストで <b>Single_Case_Cartons</b> を選択します。                     |
| サンプル図   |  |
| ステップ    | DOを設定                                                                                |

|         |                                                                                    |
|---------|------------------------------------------------------------------------------------|
| 説明      | 吸盤を制御して段ボール箱を吸着します。                                                                |
| 操作      | ステップライブラリから「DOを設定」を選択し、プロジェクト編集エリアにドラッグして「箱吸着」という名前を付けます。                          |
| パラメータ設定 | <b>DO値</b> と <b>DOポート</b> を <b>1</b> に設定します。このパラメータ値は参考値であり、実際の現場要件に応じて設定してください。   |
| サンプル図   |  |

ロボットが移動する前に吸盤が箱をしっかりと吸着するために、**待つ**ステップを追加します。

|         |                                                                                      |
|---------|--------------------------------------------------------------------------------------|
| ステップ    | 待つ                                                                                   |
| 説明      | ロボットが箱の把持に失敗するのを防ぎます。                                                                |
| 操作      | ステップライブラリから「待つ」を選択し、プロジェクト編集エリアにドラッグして「把持待ち」という名前を付けます。                              |
| パラメータ設定 | <b>待ち時間</b> を <b>1000ms</b> に設定します。                                                  |
| サンプル図   |  |

### 把持位置から離れる経路点に移動

ロボットが箱を把持した後、箱とシーン物体との衝突を防ぐための**相対移動**を使用して、ロボットを把持位置から離れる経路点に移動させます。

|         |                                                                                                                                                       |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| ステップ    | 相対移動                                                                                                                                                  |
| 説明      | ビジョン結果に従って移動します。                                                                                                                                      |
| 操作      | ステップライブラリから「相対移動」を選択し、プロジェクト編集エリアにドラッグして「経路点から離れる」という名前を付けます。                                                                                         |
| パラメータ設定 | <b>移動基準</b> を <b>次の経路点に対して</b> に設定し、 <b>目標点タイプ</b> を <b>ロボットハンド</b> に設定し、 <b>Z軸</b> の座標を箱の高さより大きく設定します。例えば、段ボール箱の高さが500mmの場合、 <b>Z</b> を-600mmに設定します。 |



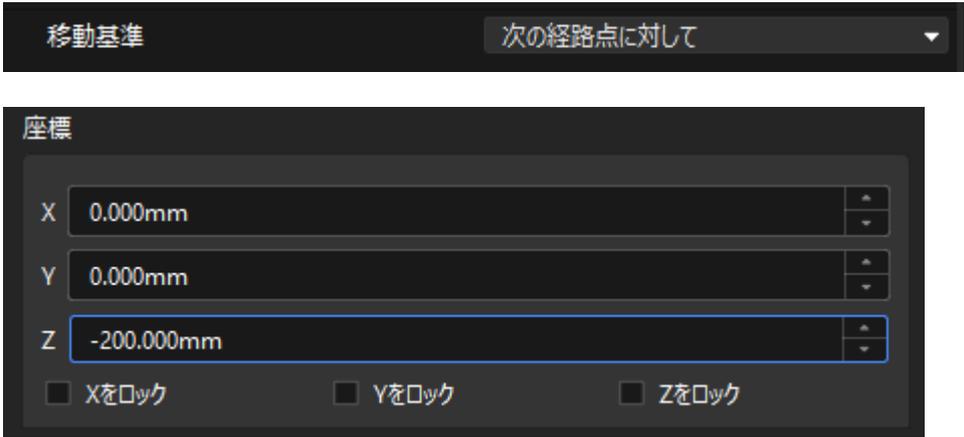
### 把持・配置経路の最適化

衝突を避けるために、経路点から離れた後に中間点を追加するための**移動ステップ**が必要です。これにより、把持・配置経路の最適化を実現しています。

|         |                                                                                      |
|---------|--------------------------------------------------------------------------------------|
| ステップ    | 移動                                                                                   |
| 説明      | 把持・配置経路を最適化します。                                                                      |
| 操作      | ステップライブラリから「移動」を選択し、プロジェクト編集エリアにドラッグして「中間点」という名前を付けます。                               |
| パラメータ設定 | 目標点タイプを関節角度を設定し、関節角度の各値を設定します。                                                       |
| サンプル図   |  |

### 配置位置に近接する経路点に移動

ロボットが箱を配置する前に、箱とシーンとの衝突を防ぐための**相対移動**ステップを使用して、ロボットを配置位置に近接する経路点に移動させます。

|         |                                                                                                            |
|---------|------------------------------------------------------------------------------------------------------------|
| ステップ    | 相対移動                                                                                                       |
| 説明      | ビジョン結果に従って移動します。                                                                                           |
| 操作      | ステップライブラリから「相対移動」を選択し、プロジェクト編集エリアにドラッグして「近接する経路点2」という名前を付けます。                                              |
| パラメータ設定 | <b>移動基準</b> を次の経路点に対してに設定し、 <b>目標点タイプ</b> をロボットハンドに設定し、実際の状況に応じて <b>Z軸</b> の座標（例えば、 <b>-200mm</b> ）を設定します。 |
| サンプル図   |                         |

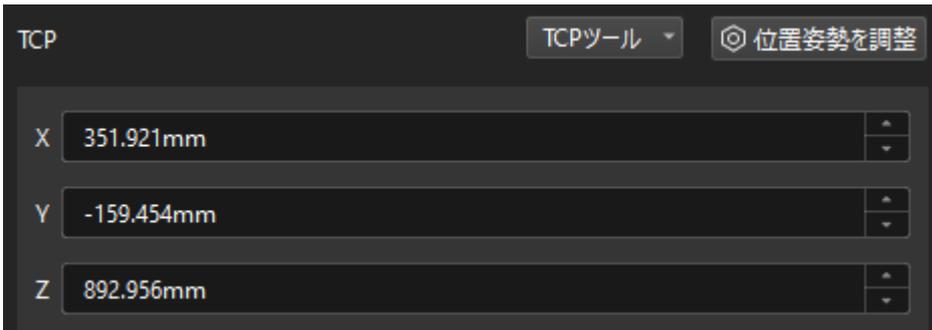
## 段ボール箱の配置

ロボットが配置位置に近接する経路点に移動すると、次の2つの手順で手順で箱を配置できます。

1. 手順1：**移動**ステップを使用して、ロボットを配置位置に移動させます。
2. 手順2：**DOを設定**ステップを使用して、吸盤の吸着を解除して箱を配置します。

詳細は以下の通りです。

|         |                                                         |
|---------|---------------------------------------------------------|
| ステップ    | 移動                                                      |
| 説明      | ロボットを配置位置に移動させます。                                       |
| 操作      | ステップライブラリから「移動」を選択し、プロジェクト編集エリアにドラッグして「配置位置」という名前を付けます。 |
| パラメータ設定 | <b>TCP位置姿勢</b> を配置位置とするために、それを正しく設定する必要があります。           |

|       |                                                                                    |
|-------|------------------------------------------------------------------------------------|
| サンプル図 |  |
|-------|------------------------------------------------------------------------------------|

|         |                                                                                     |
|---------|-------------------------------------------------------------------------------------|
| ステップ    | DOを設定                                                                               |
| 説明      | 吸着を解除して箱を配置します。                                                                     |
| 操作      | ステップライブラリから「DOを設定」を選択し、プロジェクト編集エリアにドラッグして「箱吸着解除」という名前を付けます。                         |
| パラメータ設定 | <b>DO値を0に、DOポートを1に設定</b> します。このパラメータ値は参考値であり、実際の現場要件に応じて設定してください。                   |
| サンプル図   |  |

吸盤の吸着解除に時間がかかるので、ロボットが箱の配置に失敗するのを防ぐために**待つ**ステップを追加する必要があります。

|         |                                                                                      |
|---------|--------------------------------------------------------------------------------------|
| ステップ    | 待つ                                                                                   |
| 説明      | ロボットが箱の配置に失敗するのを防ぎます。                                                                |
| 操作      | ステップライブラリから「待つ」を選択し、プロジェクト編集エリアにドラッグして「配置待ち」という名前を付けます。                              |
| パラメータ設定 | <b>待ち時間を1000msに設定</b> します。                                                           |
| サンプル図   |  |

## ステップ接続

上記のステップが追加されたら、順番につなぎます。ロボットが段ボール箱の把持と配置を繰り返し実行するために、「配置待ち」の出口を「初期位置」の入口につなぎます。

## シミュレーションと実行

1. ツールバーの[シミュレート]をクリックすると、作成したMech-Vizプロジェクトがシミュ

レーションで実行されます。



2. シミュレーション効果が期待通りにある場合、Mech-Vizツールバーの[実行]ボタンをクリックして、ロボット実機を動作させます。



ロボットを低い速度で動作させることをお勧めします。緊急時には緊急停止ボタンを押してください。