



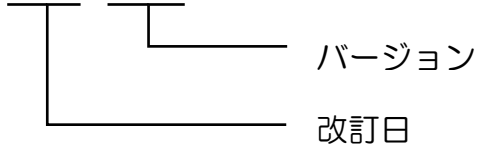
Application Note

E Series EtherCAT Drive Complete Setup
with HIMC3 iA Studio

改訂履歴

マニュアルのバージョンは、表紙の下部にも記載されています。

MD55UJ01-2605_V1.0



日付	バージョン	適用機種	改訂内容
2026 年 5 月	1.0	E シリーズ EtherCAT ドライバー	初版

関連文書

関連資料を通じて、ユーザーはこのマニュアルの位置づけや、マニュアルと製品の関連性を迅速に理解できます。詳細は、HIWIN MIKROSYSTEM の公式サイト→ダウンロード→マニュアル概要 (https://www.hiwinmikro.tw/Downloads/ManualOverview_EN.htm)をご覧ください。

序文

このマニュアルでは、E シリーズ EtherCAT ドライバーを HIMC3 モーションコントローラーと組み合わせて使用する場合の、iA Studio ソフトウェアの操作方法について説明します。E シリーズドライバーの詳細については、関連するユーザーマニュアルを参照してください。

ソフトウェア／ハードウェアの仕様

名称	ソフトウェア/ファームウェアのバージョン
E1 シリーズ EtherCAT ドライバー	ソフトウェア (Thunder) : 1.14.11.0 以降 ファームウェア : 2.14.9 以降 ESI ファイル : HIWIN_MIKROSYSTEM_EDxF_20250725 以降
E2 シリーズ EtherCAT ドライバー	ソフトウェア (Thunder) : 1.14.11.0 以降 ファームウェア : 3.14.9 / 4.14.9 以降 ESI ファイル : HIWIN_MIKROSYSTEM_EDxF_20250725 以降
HIMC3 モーションコントローラー	ソフトウェア (iA Studio) : 3.3.0 以上 ファームウェア : 3.3.0 以上

目次

1. 通信とモジュール設定	1-1
1.1 コントローラーへの接続.....	1-2
1.1.1 接続設定	1-2
1.1.2 イーサネット経由でコントローラーに接続する	1-4
2. パラメーター設定	2-1
2.1 コントローラーの設定.....	2-2
2.2 ESI ファイルのインストール.....	2-3
2.3 モーションコントロール軸の設定.....	2-5
2.4 PDO マッピング.....	2-8
3. テストラン	3-1
3.1 Enable（有効）	3-2
3.2 ホーミング.....	3-3
3.3 ショグ.....	3-5
3.4 相対的な動き.....	3-6
3.5 ポイントツーポイント動作.....	3-7
3.6 HMPL エディタ	3-8
3.6.1 ポイントツーポイント動作	3-11
3.6.2 ホーミング	3-13
4. その他のアプリケーション設定	4-1
4.1 誤差補正機能の設定.....	4-2
4.1.1 1次元誤差補正	4-2
4.1.2 2D 誤差補正.....	4-3

1. 通信とモジュール設定

1.1	コントローラーへの接続.....	1-2
1.1.1	接続設定	1-2
1.1.2	イーサネット経由でコントローラーに接続する	1-4

1.1 コントローラーへの接続

「Connection Setting」ウィンドウでは、ユーザーは指定された通信タイプを使用して iA Studio をコントローラーに接続できます。

1.1.1 接続設定

1. メニューバーの「Controller」をクリックし、「Connection Setting」をクリックして「Connection Setting」ウィンドウを開きます。

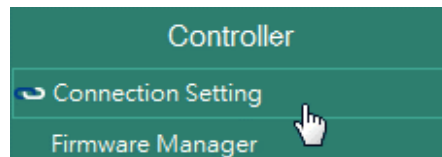


図 1.1.1.1 接続設定

2. コントローラーをコンピューターに接続するためのポート（CN3 または CN4）を選択します。

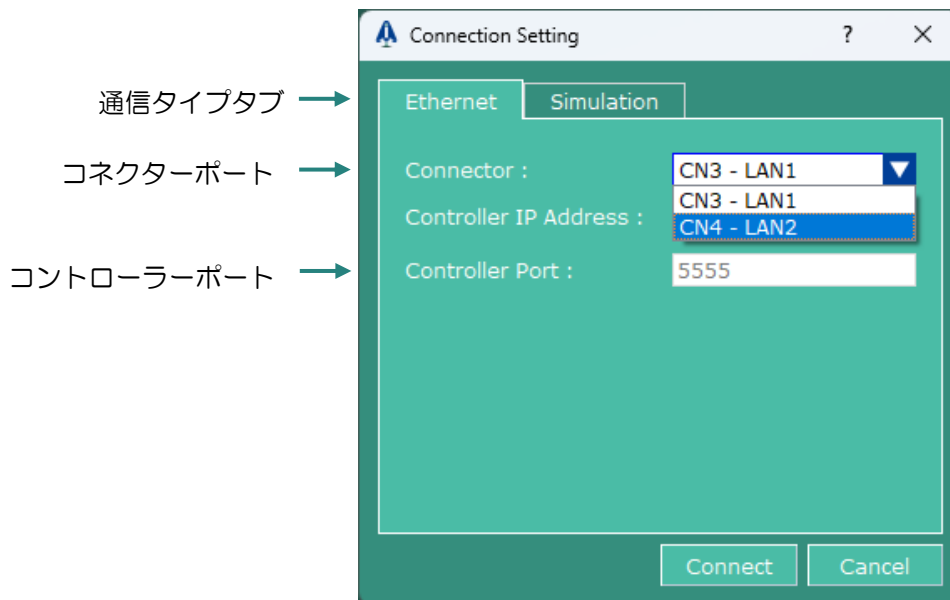


図 1.1.1.2 「Connection Setting」ウィンドウ：EtherCAT

3. CN3 を使用する場合は、コンピューターのネットワークカードとコントローラーを同じ IP サブネット（例：192.168.0.XXX）内に設定する必要があります。CN4 を使用する場合は、IP アドレスは 169.254.188.20 に固定されます。

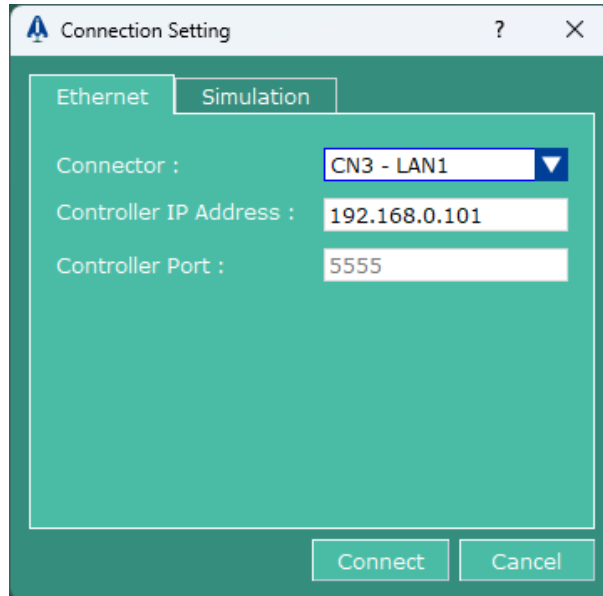


図 1.1.1.3 CN3 - 自己定義 IP 接続

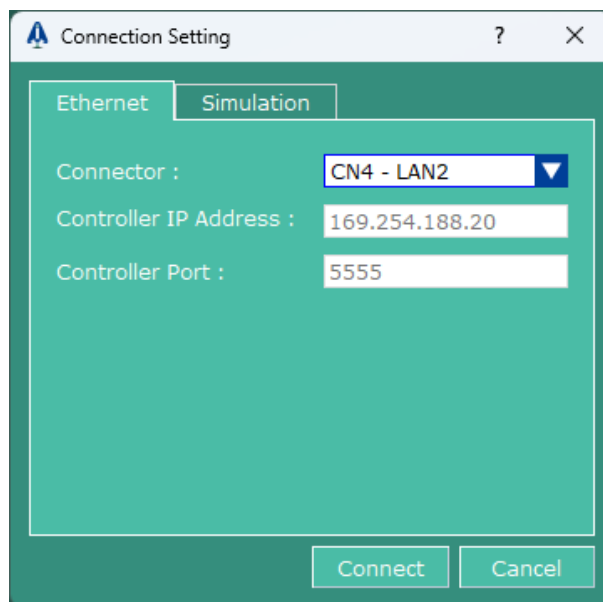


図 1.1.1.4 CN4 - 固定 IP 接続

表 1.1.1.1 「Connection Setting」ウィンドウ

通信タイプタブ	説明
Ethernet	TCP/IP 経由でコントローラーに接続します。
Simulation	シミュレーターに接続します。

1.1.2 イーサネット経由でコントローラーに接続する

コントローラーはイーサネット経由で接続できます。接続を確立するには、以下の手順に従ってください。

1. 「Connection Setting」ウィンドウで「Ethernet」タブを選択します。
2. コントローラーのIPアドレスとIPポート番号を入力します。
3. 「Connect」ボタンをクリックして接続を開始します。接続の進行状況を示すポップアップウィンドウが表示されます。

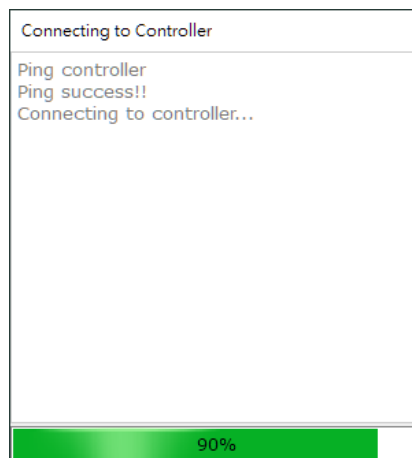


図 1.1.2.1 接続状況ポップアップウィンドウ

接続が正常に確立されると、「Connection Setting」ウィンドウとポップアップウィンドウは自動的に閉じます。接続が確立できない場合は、エラーダイアログが表示されます。表示された場合は、通信ケーブルがコントローラに正しく接続されているか確認してください。

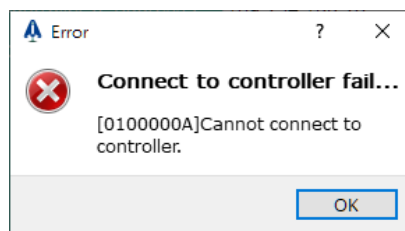


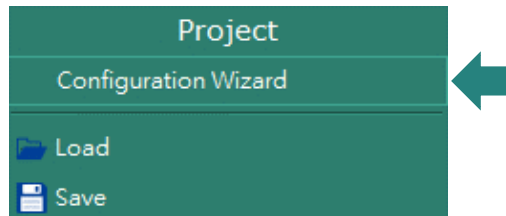
図 1.1.2.2 コントローラーへの接続失敗

2. パラメーター設定


2.1	コントローラーの設定.....	2-2
2.2	ESI ファイルのインストール.....	2-3
2.3	モーションコントロール軸の設定.....	2-5
2.4	PDO マッピング.....	2-8

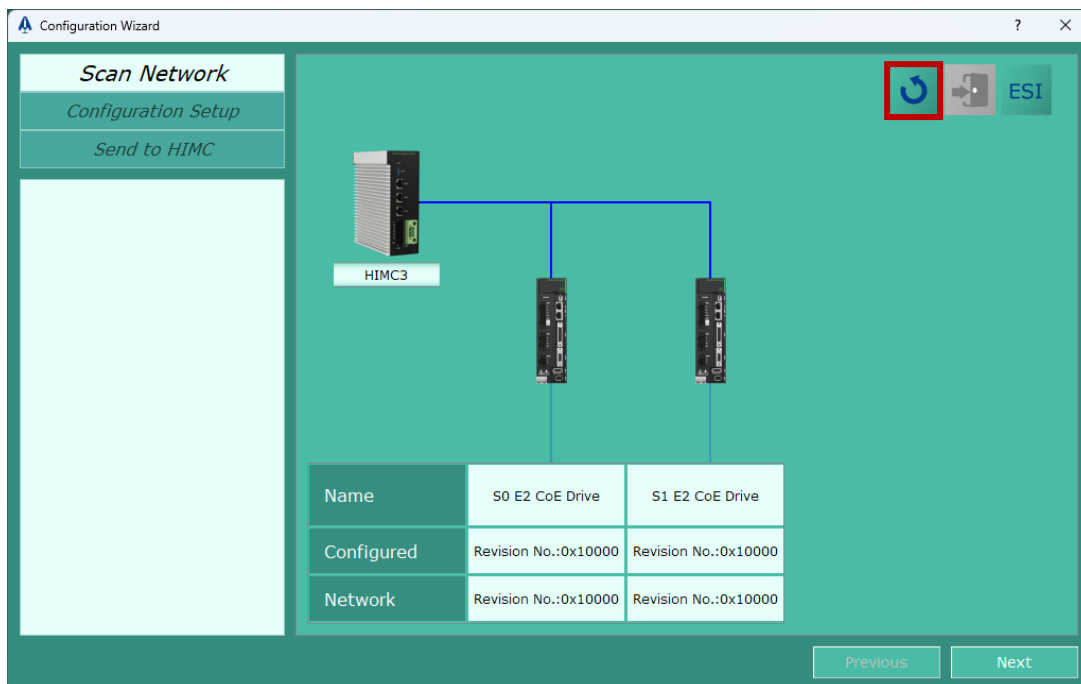
2.1 コントローラーの設定

1. メニューバーの「Project」をクリックし、「Configuration Wizard」をクリックします。



☒ 2.1.1 Configuration Wizard

2.  をクリックしてドライバーをスキャンします。



☒ 2.1.2 ドライバーのスキャン

2.2 ESI ファイルのインストール



重要

- (1) EシリーズドライバーのESIファイル名：HIWIN_MIKROSYSTEM_EDxF_yyyymmdd.xml
- (2) EシリーズドライバーのESIファイルは、ドライバーのThunderソフトウェアのインストールパス（Thunder/doc/ESI Files）から取得できます。

1. 「Configuration Wizard」ウィンドウで **ESI** をクリックします。

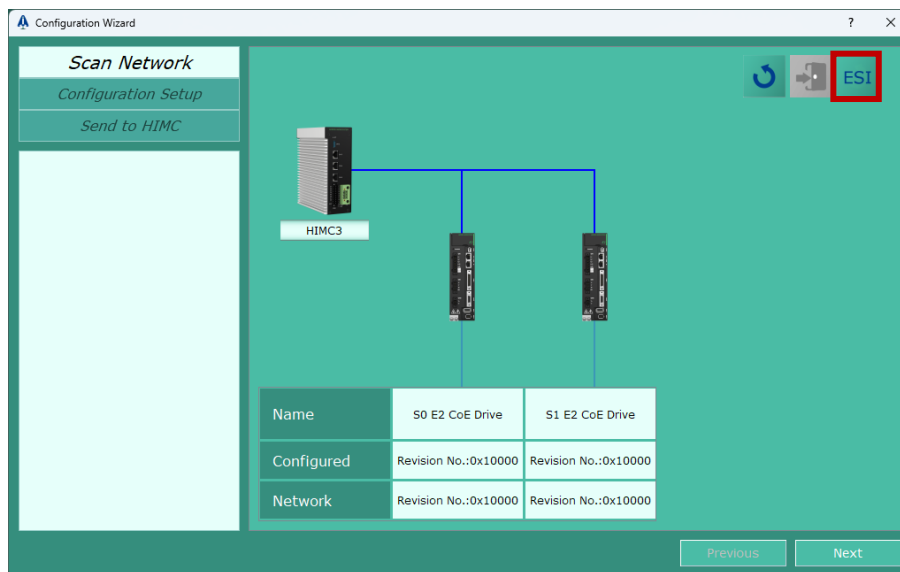


図 2.2.1 ESI ファイルの設定

2. [Add]をクリックして、ファームウェアのバージョンに対応する ESI ファイルを追加します。

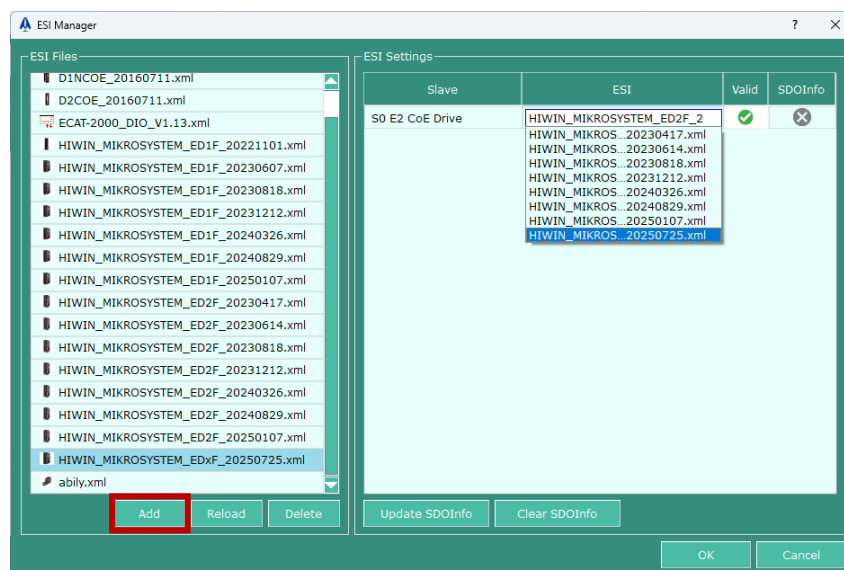


図 2.2.2 ESI ファイルの追加

3. 読み込む ESI ファイル (.xml) を選択し、「Load」をクリックします。

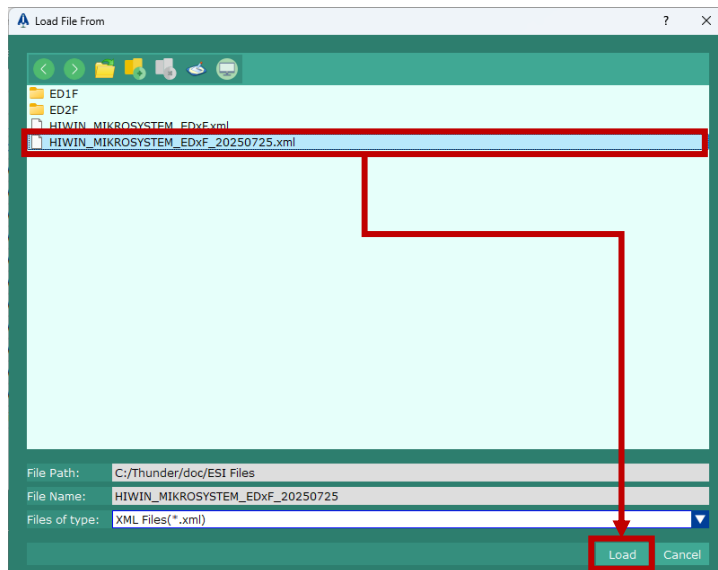


図 2.2.3 ESI ファイルの読み込み

4. 必要な ESI ファイル (.xml) のバージョンを選択し、「OK」をクリックしてインストールを完了します。

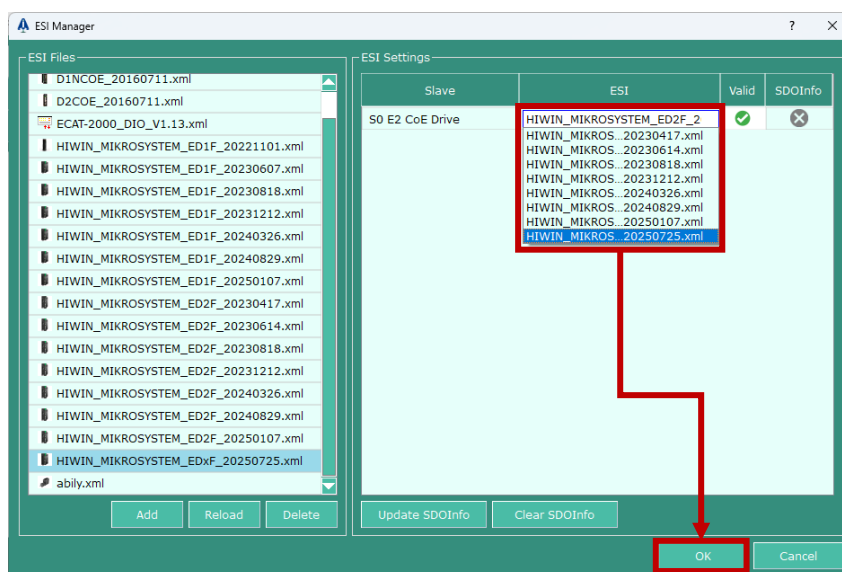



図 2.2.4 ESI ファイルの選択

2.3 モーションコントロール軸の設定

1. 「Configuration Setup」タブで、スレーブ軸の前の  をクリックします。

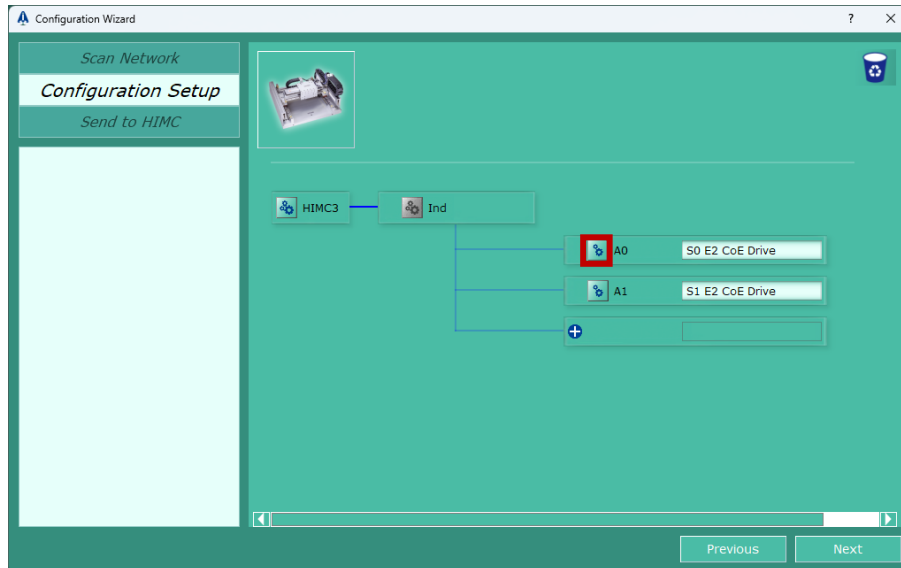


図 2.3.1

2. 「Axis」タブで、動作モード、モータータイプ、駆動設定、安全設定など、軸関連のパラメーターを設定します。

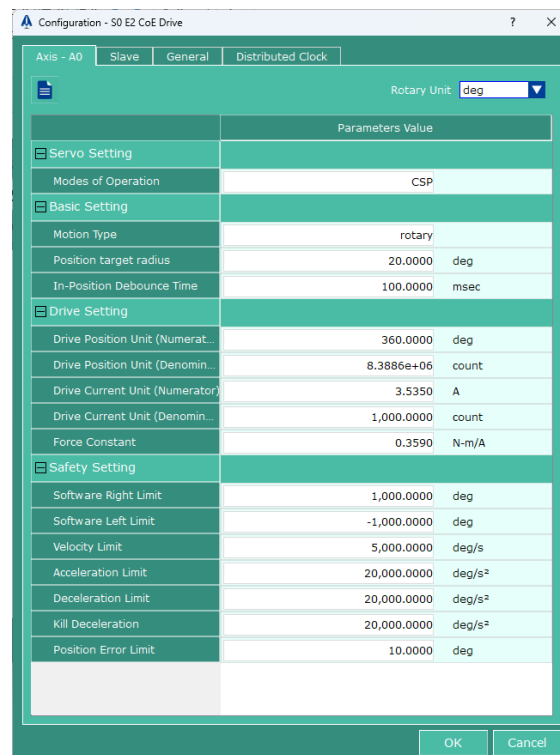


図 2.3.2

3. 動作タイプが正しいか確認してください。



重要

非回転式ダイレクトドライブアーキテクチャ（回転モーターとネジを組み合わせたアプリケーションを含む）の場合、ネジ送り定数はドライバー内の電子ギア比で設定し、リニア制御ユニットに変換する必要があります。そのため、制御異常を防ぐために、コントローラーの設定時にはモーションタイプをリニアに設定する必要があります。

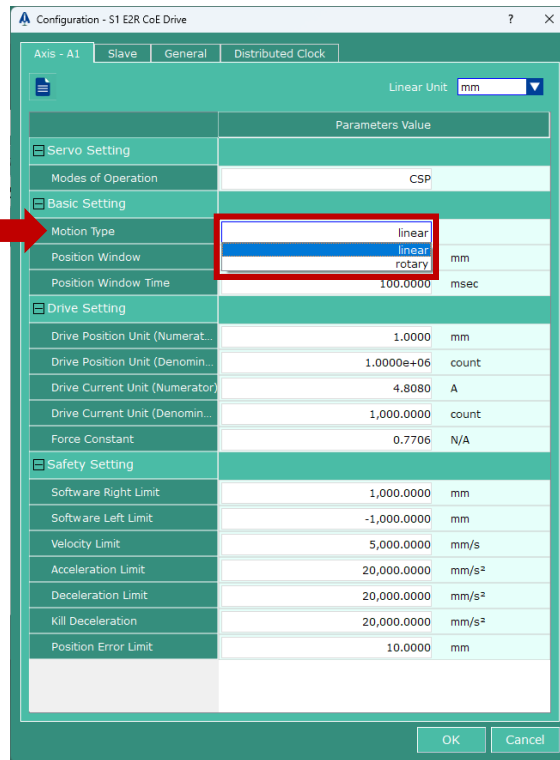


図 2.3.3

4. ドライバーとコントローラーの制御ユニットが同じかどうかを確認します。



重要

装置機構の電子ギア比が既に駆動装置で設定されている場合は、駆動装置とコントローラーの制御ユニットが同じであることを確認するだけで済みます。装置機構の電子ギア比が駆動装置で1:1に設定されている場合、機構の電子ギア比の変換をコントローラーで設定する必要があります。

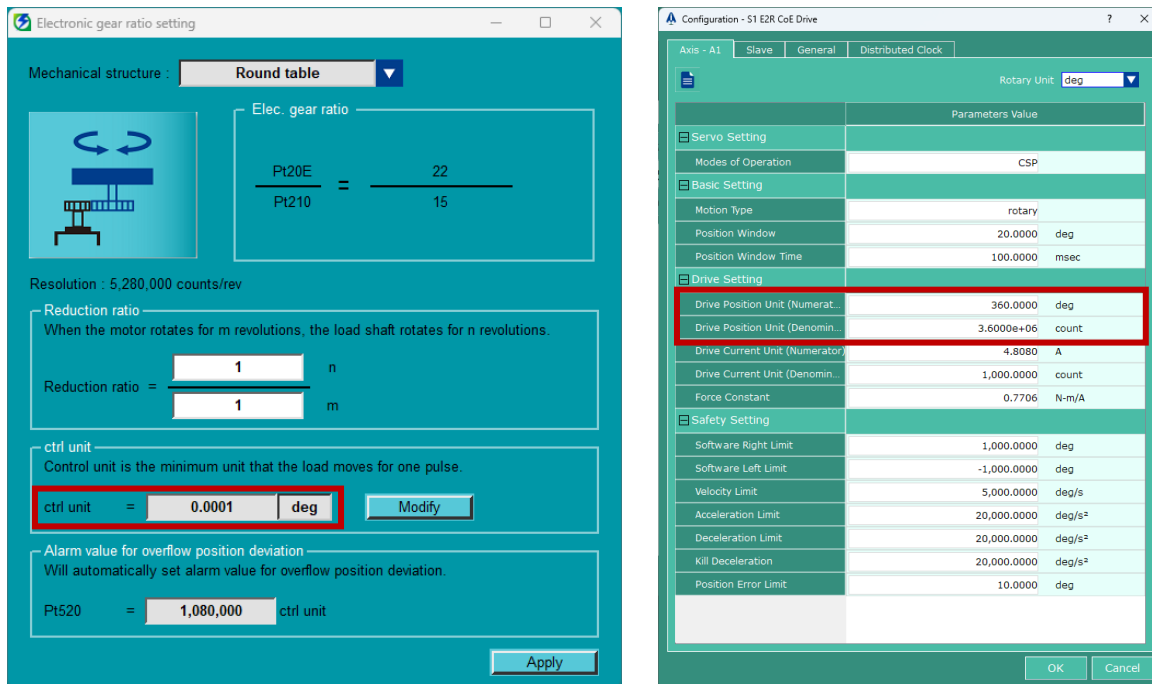


図 2.3.4

- 「Send to HIMC」タブに切り替えます。変更したパラメーターが正しいことを確認したら、「Send to HIMC」をクリックして設定を完了します。

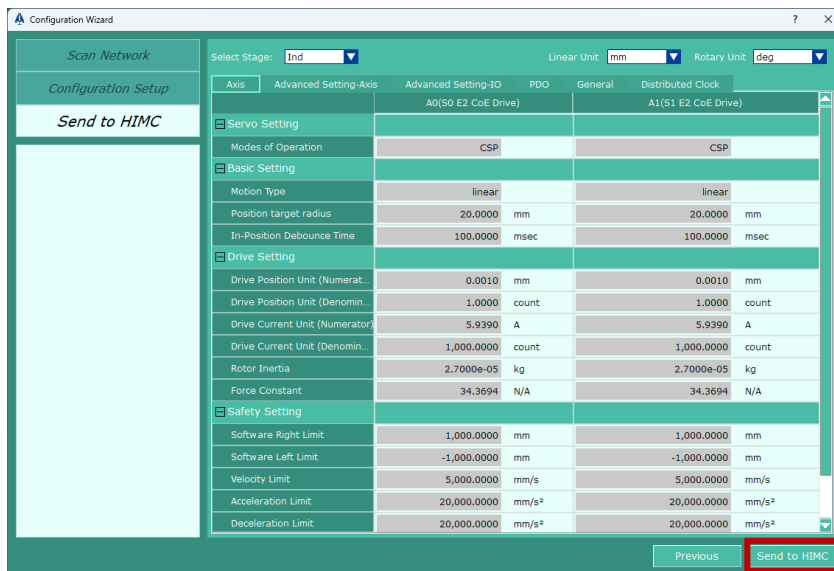


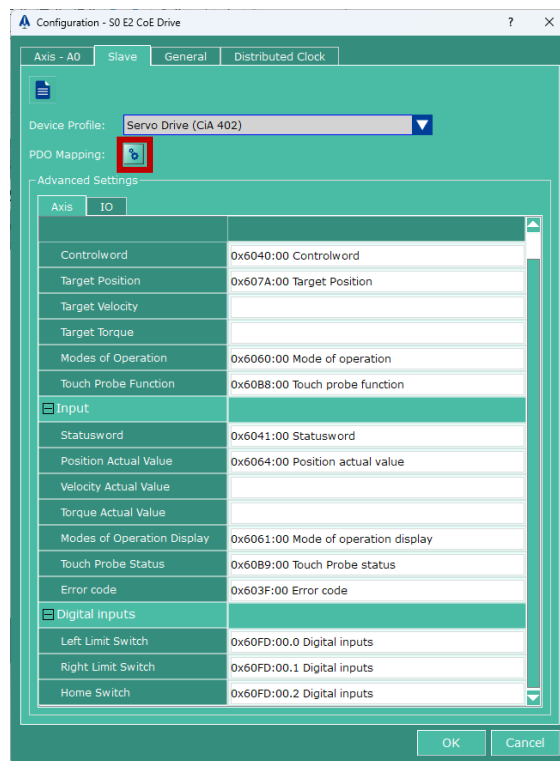
図 2.3.5

2.4 PDO マッピング




- (1) いずれかのPDOグループをクリックすると、「Edit PDO Map Settings」ウィンドウの右側にそのグループのデフォルトのPDOオブジェクトが表示されます。
- (2) 「Add PDO Entry」をクリックすると、グループに他のオブジェクトが追加されます。「Delete PDO Entry」をクリックすると、グループから既存のオブジェクトが削除されます。
- (3) EシリーズドライバーのRxPDOおよびTxPDOオブジェクトの最大数はそれぞれ10です。

1. 「Slave」タブに切り替えた後、PDO マッピングの  をクリックします。



2.4.1

2. 希望するデフォルトの PDO を選択します。設定を調整するには、右上隅の  をクリックします。設定が完了したら、「OK」をクリックします。

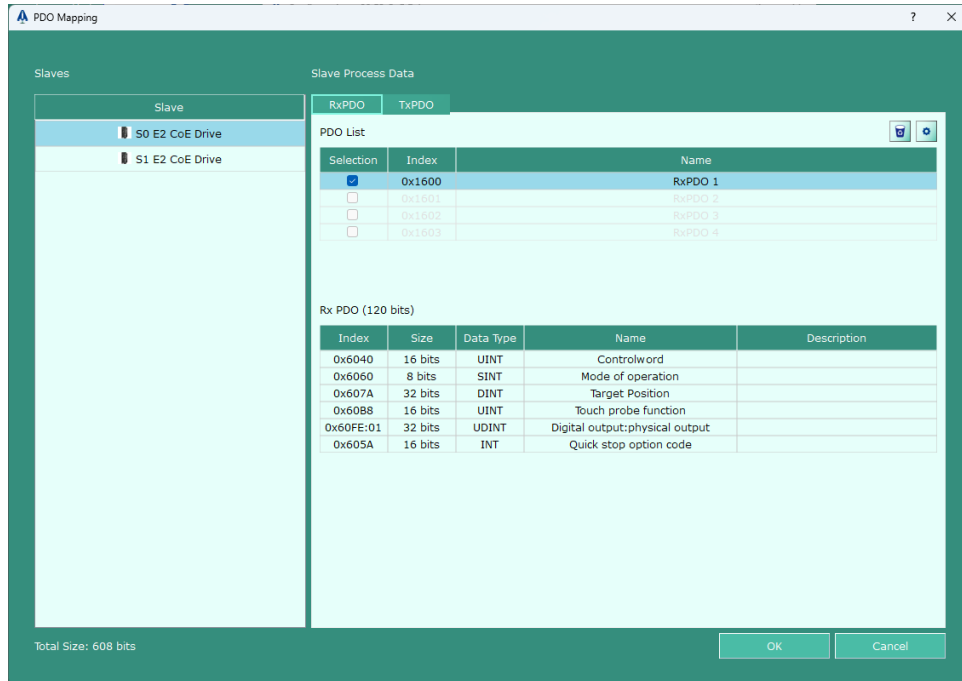


図 2.4.2 デフォルトの PDO を選択

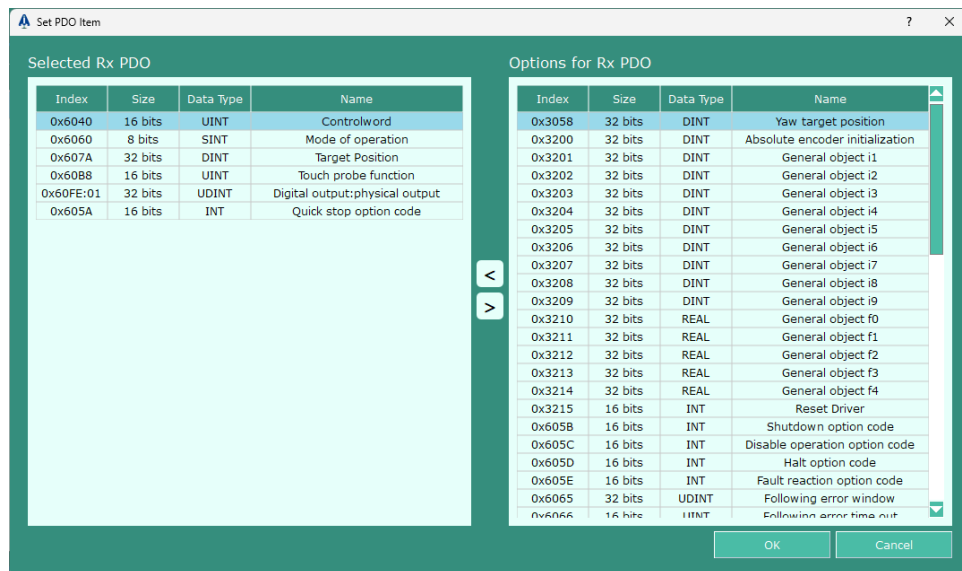
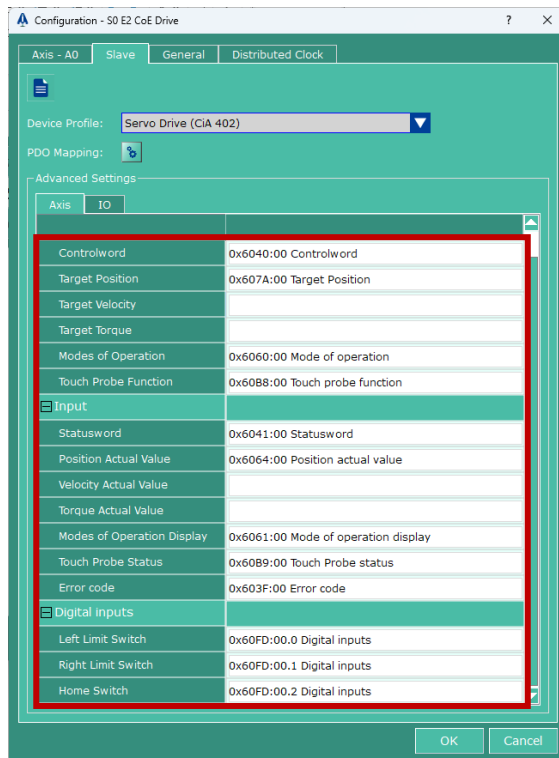


図 2.4.3 自己定義 PDO の選択

3. 最後に、「Slave」タブで PDO マッピングを実行します。



☒ 2.4.4

3. テストラン

3.1	Enable (有効)	3-2
3.2	ホームイング.....	3-3
3.3	ジョグ.....	3-5
3.4	相対的な動き.....	3-6
3.5	ポイントツーポイント動作.....	3-7
3.6	HMPL エディタ	3-8
3.6.1	ポイントツーポイント動作	3-11
3.6.2	ホームイング	3-13

本章では、iA Studio のインターフェースと HMPL のシンプルなプログラムを用いてテストを実行する方法を紹介します。



重要

このマニュアルでは基本的な機能のみを紹介しています。その他の機能については、「HIMCシリーズ iA Studio ユーザーガイド」および「HIMCシリーズ HMPL ユーザーガイド」を参照してください。

3.1 Enable（有効）

1. メニューバーの「Tools」をクリックし、「Motion Manager」をクリックします。

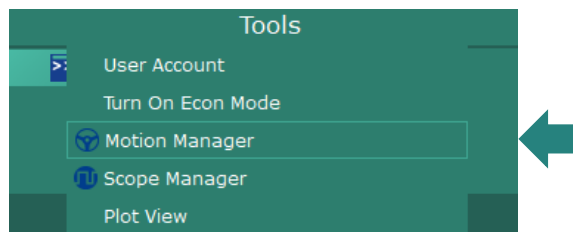


図 3.1.1 Motion Manager

2. 「Motion Manager」ウィンドウで「Enable（有効） / Disable（無効）」をクリックして、モーターを有効/無効にします。「Enable（有効）」のインジケータランプを確認して、操作が成功したかどうかを確認してください。

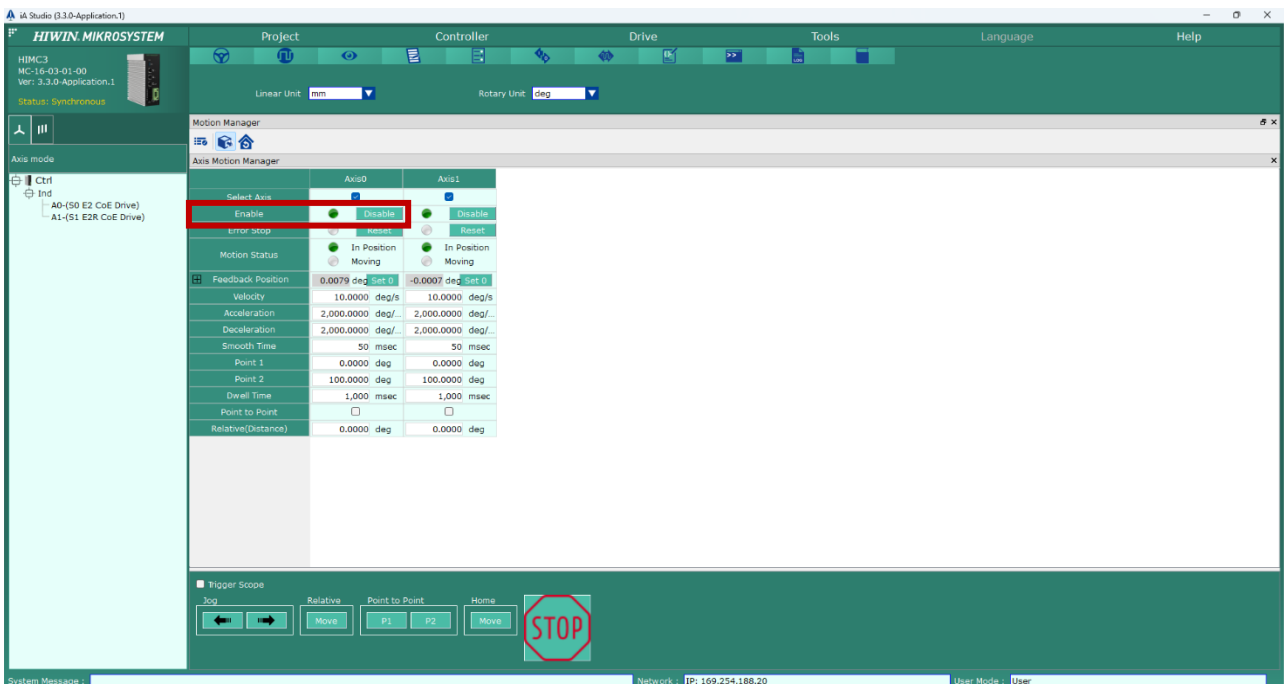



図 3.1.2 「Motion Manager」ウィンドウ – Enable（有効） / Disable（無効）

3.2 ホーミング

1. 「Motion Manager」ウィンドウで  をクリックして、ホーミング機能ウィンドウに切り替えます。

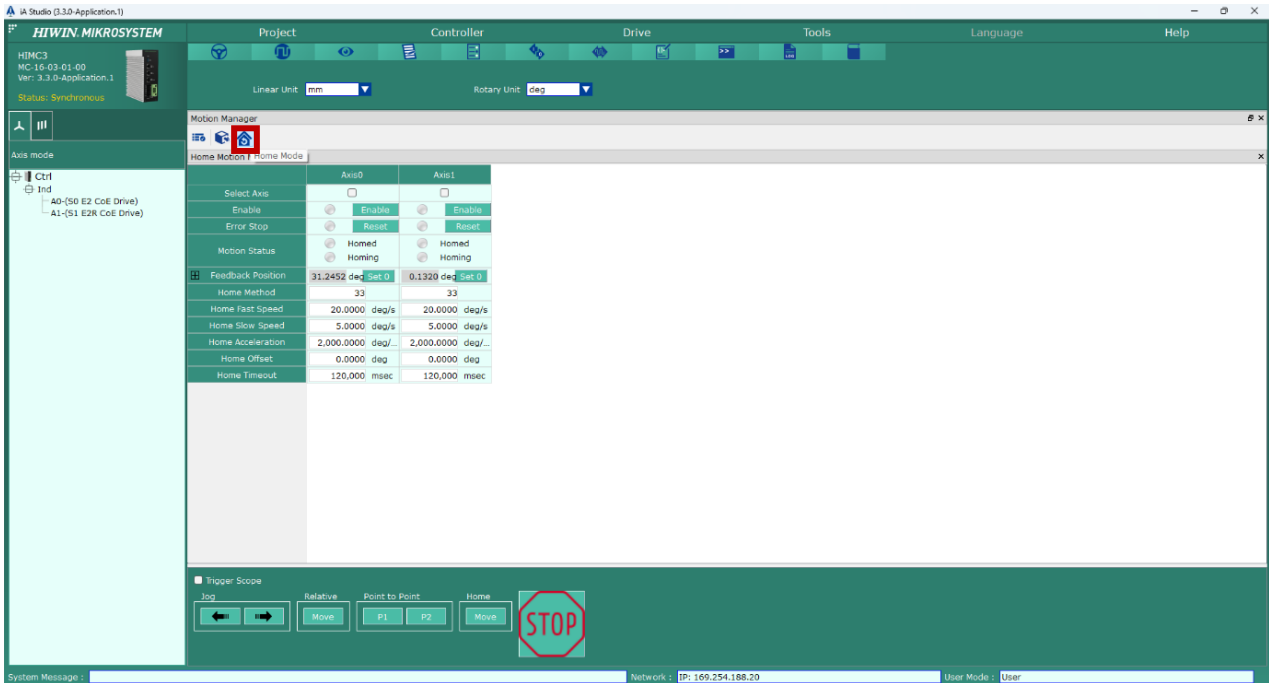


図 3.2.1 「Motion Manager」ウィンドウ - ホーミング

2. ホーミング方法、速度、加速度、オフセットなど、アプリケーションの要件に基づいてホーミングパラメータを設定します。

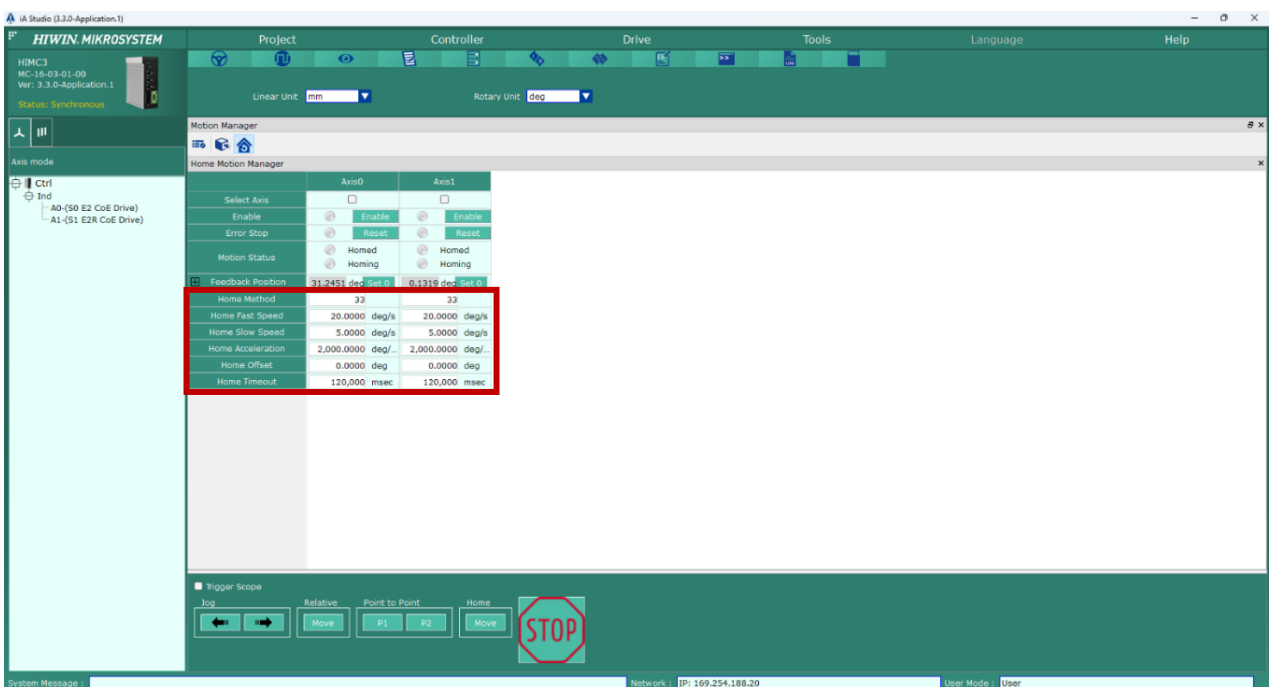


図 3.2.2 ホーミング条件の設定

3. 原点復帰させる軸を選択し、「Enable」をクリックして、原点復帰の「Move」をクリックします。

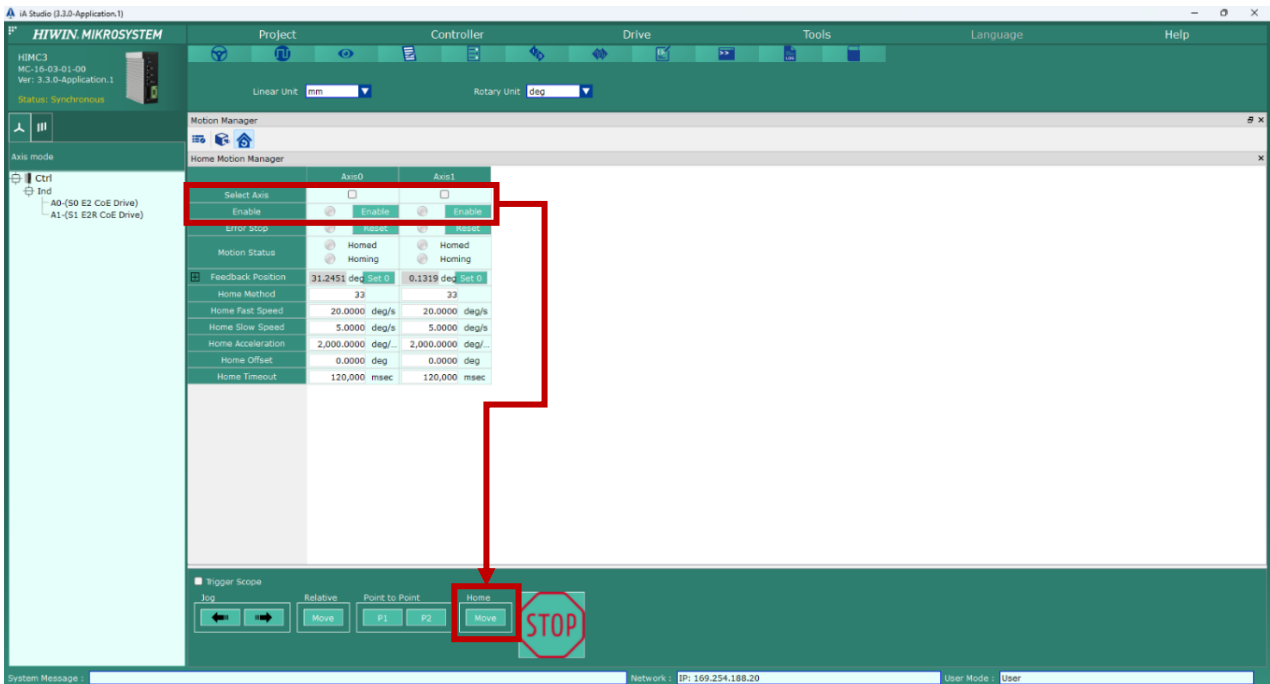


図 3.2.3 ホーミングを有効にする

4. Homed ランプが緑色に点灯したら、ホーミングは完了です。

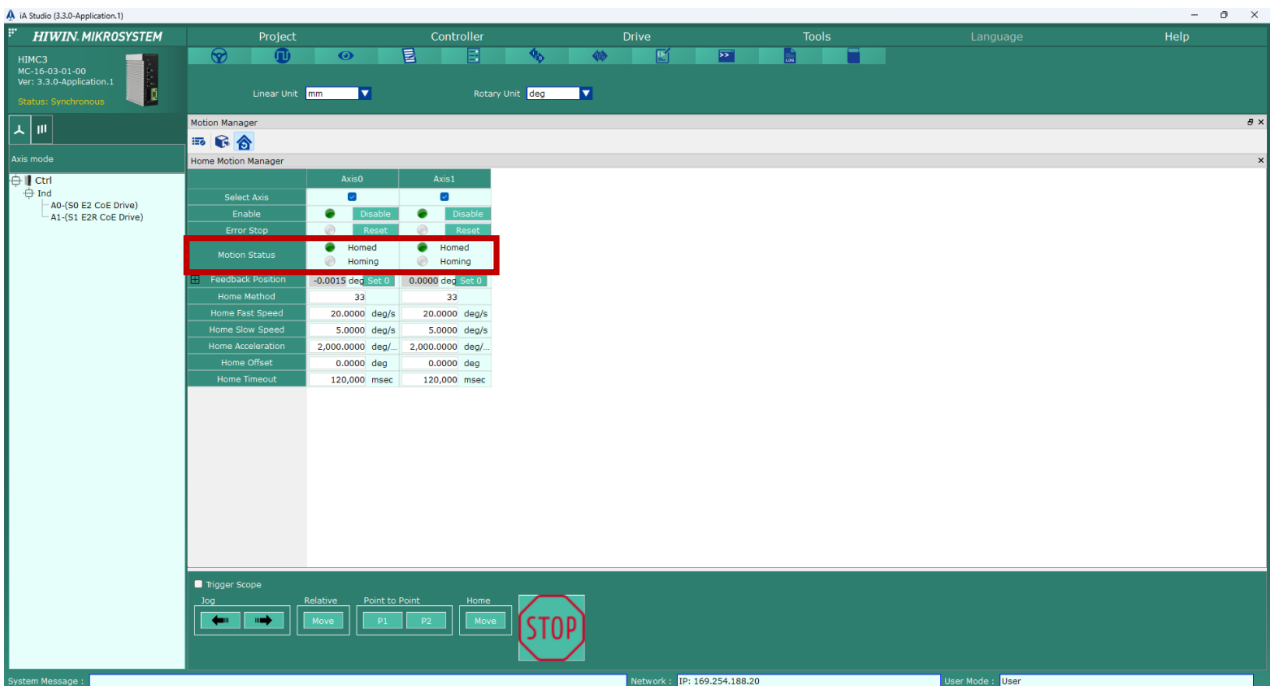


図 3.2.4 ホーミング完了

3.3 ジョグ

「Motion Manager」ウィンドウの「Jog」の矢印をクリックすると、前進または後退のジョグ操作を実行できます。

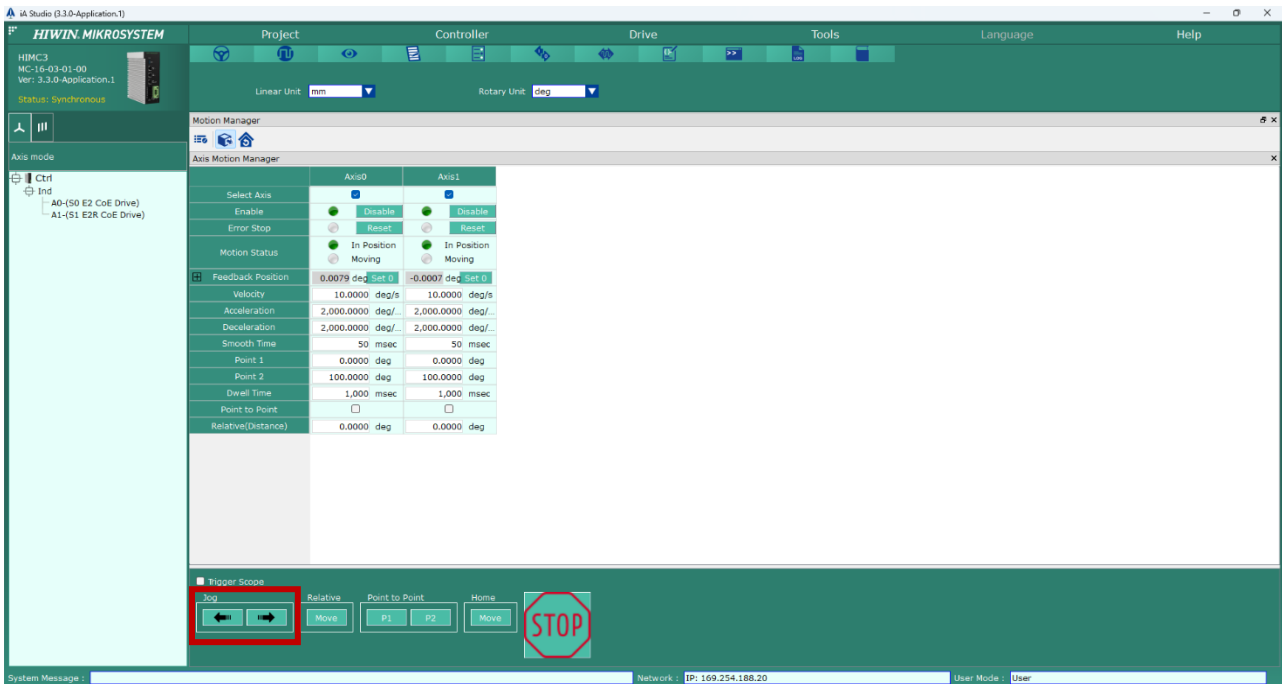


図 3.3.1 「Motion Manager」ウィンドウ - ジョグ

3.4 相対的な動き

「Motion Manager」ウィンドウで、Relative(Distance)を設定し、「Move of Relative」をクリックして相対移動を実行します。

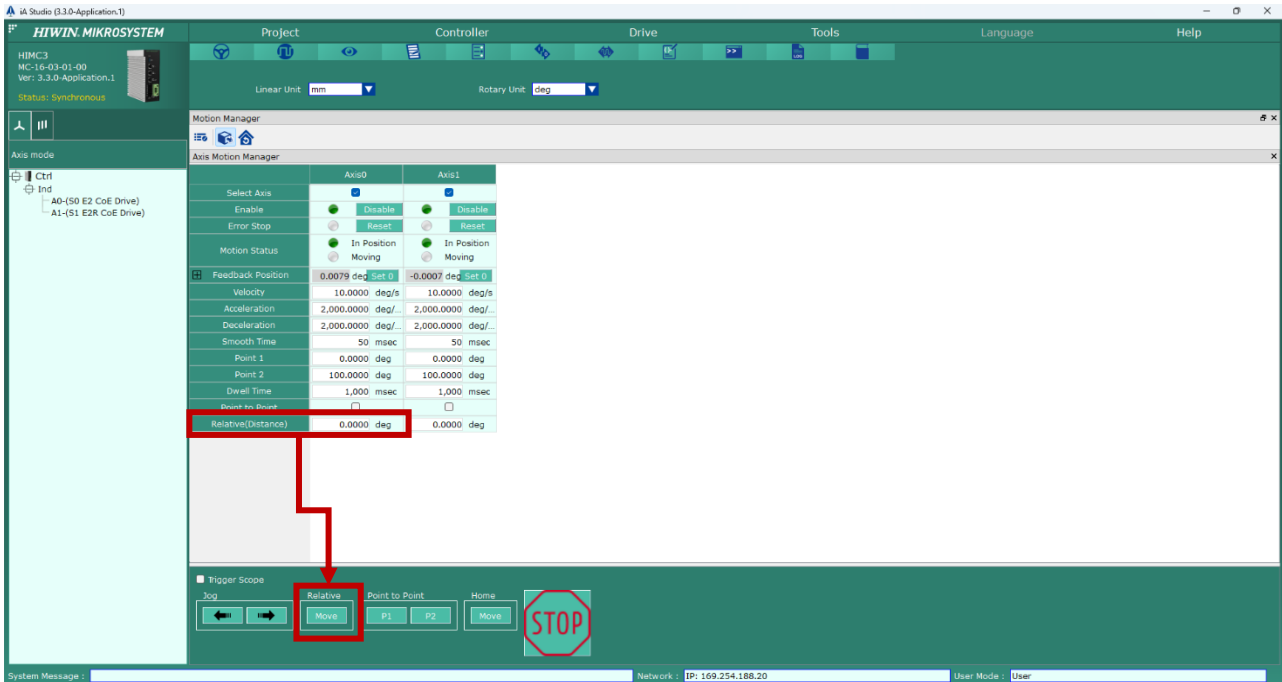


図 3.4.1 「Motion Manager」ウィンドウ - 相対移動

3.5 ポイントツーポイント動作

1. 「Motion Manager」ウィンドウで、ポイント 1、ポイント 2、および停止時間を設定します。
2. ポイントツーポイントの P1 または P2 をクリックして、モーターをポイント 1 またはポイント 2 の位置に移動します。
3. 「Point to Point」にチェックを入れ、P1 または P2 をクリックすると、モーターがポイント 1 とポイント 2 の位置間を繰り返し移動します。

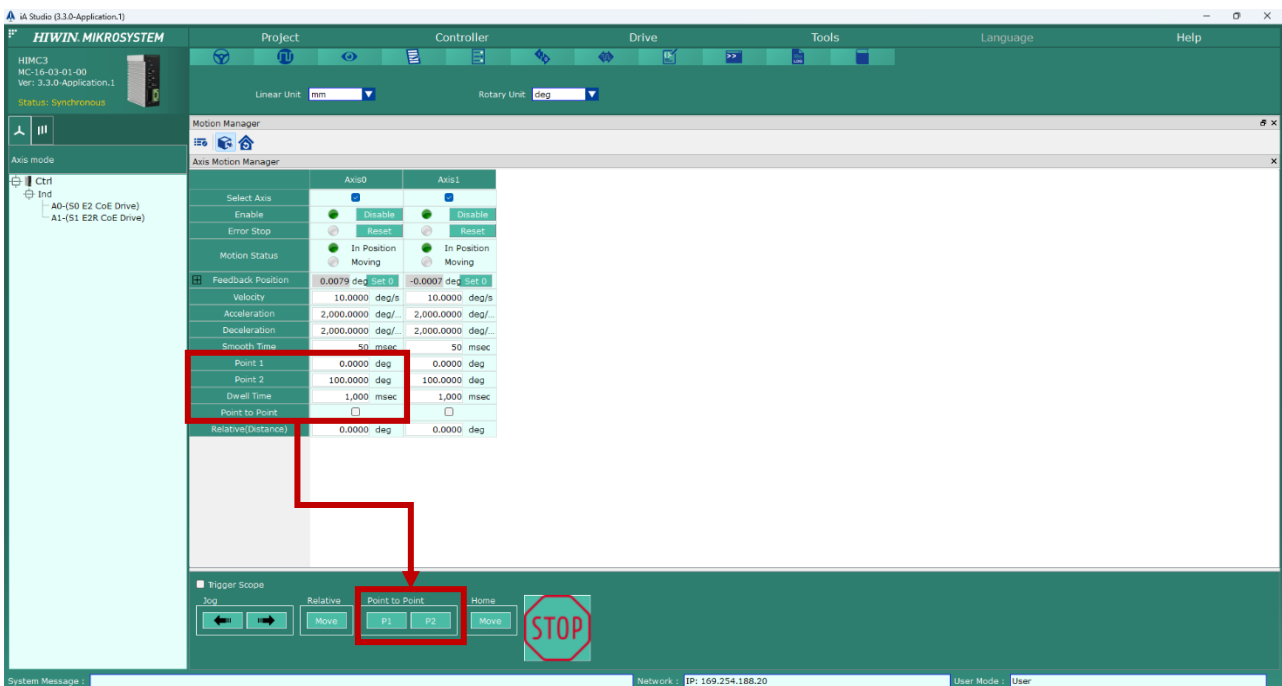


図 3.5.1 「Motion Manager」ウィンドウ - ポイントツーポイントモーション

3.6 HMPL エディタ

1. ニューバーの「Tools」をクリックし、「HMPL Editor」をクリックして HMPL 編集ウィンドウを開きます。

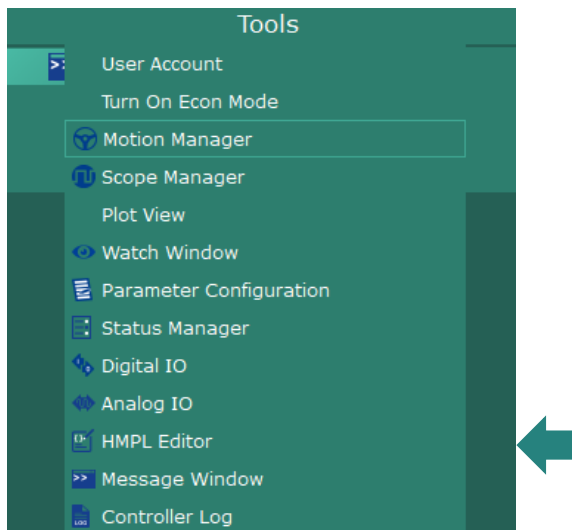


図 3.6.1 HMPL Editor

2. task をダブルクリックして、プログラム編集ページを開きます。

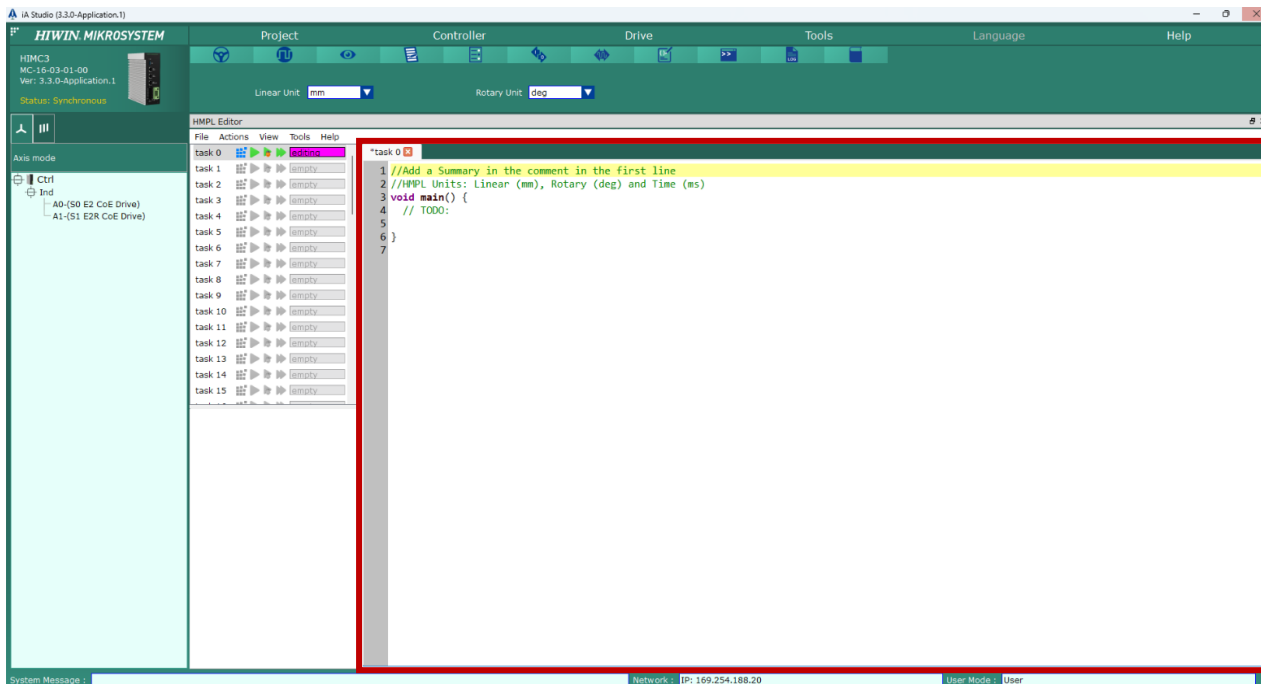



図 3.6.2 プログラム編集

3.  (F7) をクリックして、作成したプログラムをコンパイルします。

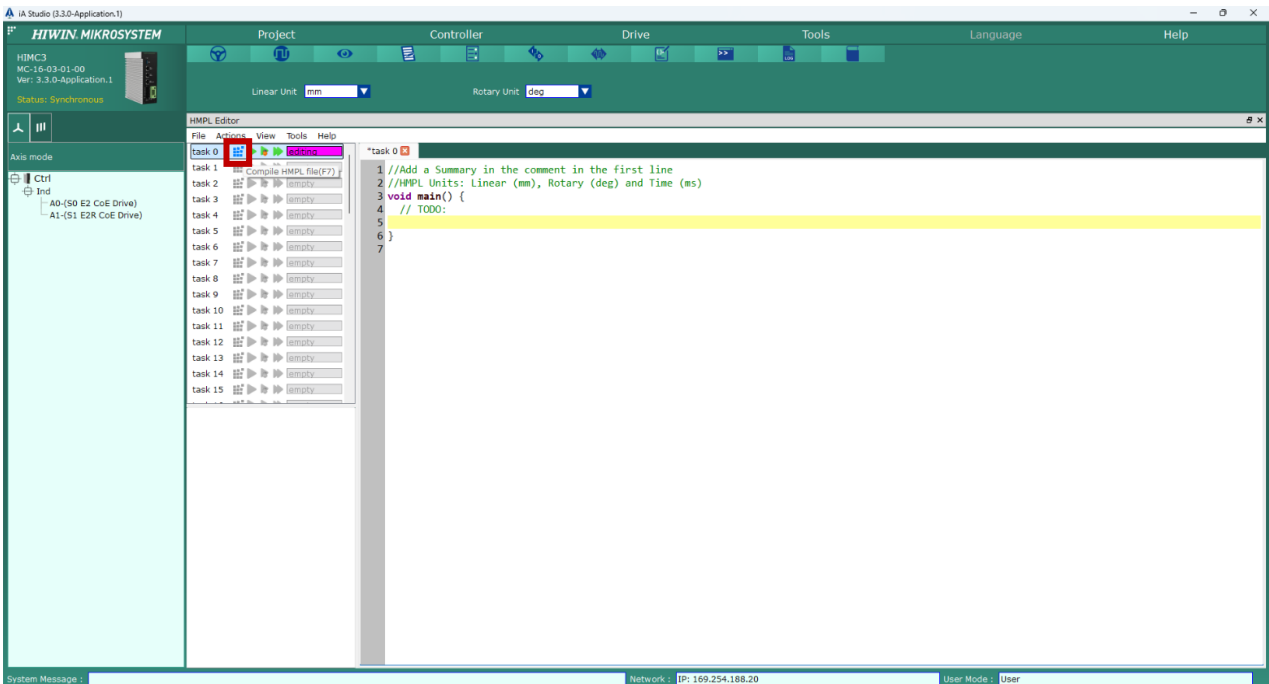



図 3.6.3 プログラムのコンパイル

4.  をクリックして、タスクをコントローラーに保存します。

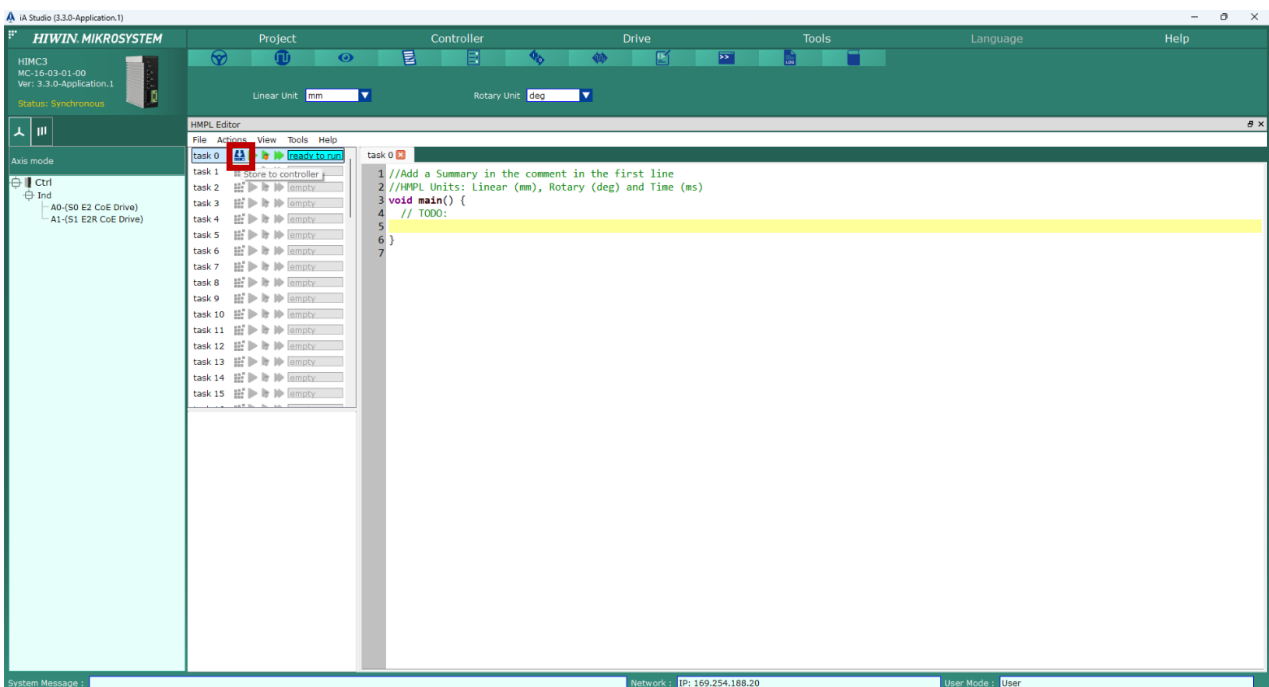



図 3.6.4 コントローラーに書き込まれたプログラム

5.  キー（Ctrl+F5）をクリックして、タスク内のプログラムを実行します。

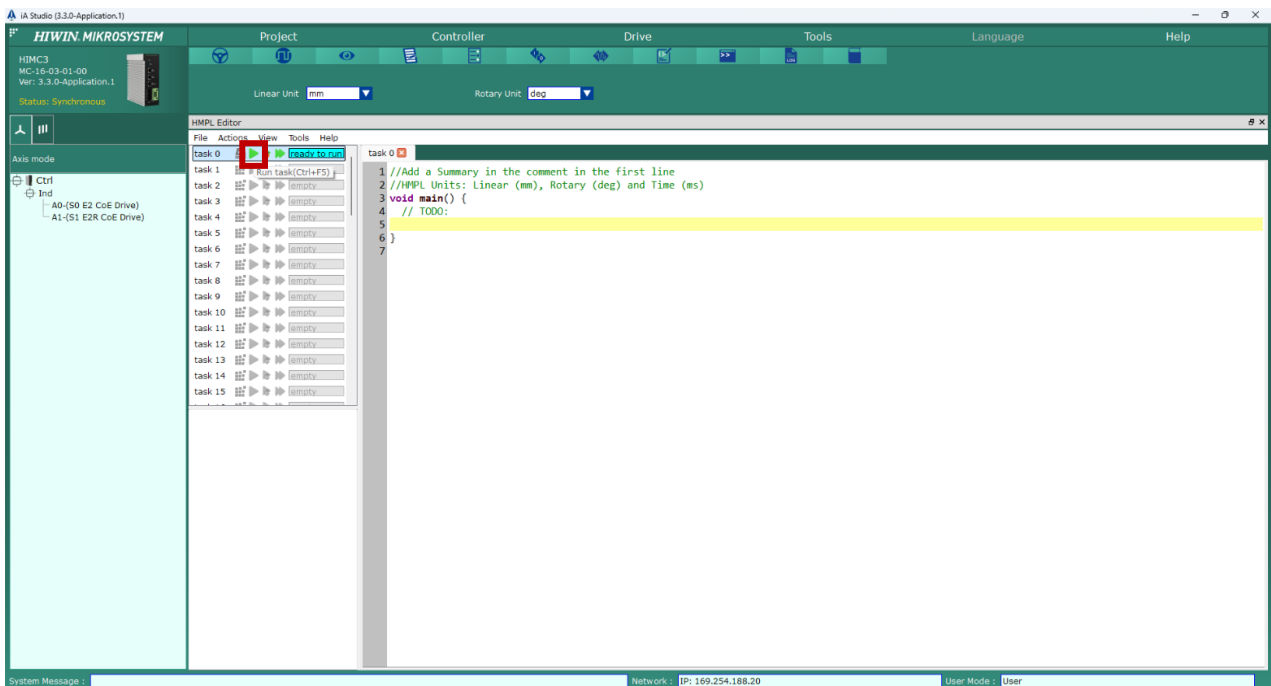


図 3.6.5 プログラムの実行

3.6.1 ポイントツーポイント動作

1. 新しいタスクを開始します。
2. タスク内でポイントツーポイントのプログラムを作成します（P1/P2 の位置、速度、加速度/減速度などを設定します）。
3. プログラムをコンパイルし、コントローラーに書き込みます。
4. プログラムを実行します。

```
int axis_x = 0; // Axis index
int P1 = 0; // P1 position Unit: linear (mm), rotary (deg)
int P2 = 30; // P2 position Unit: linear (mm), rotary (deg)
int Dwell_Time = 1000; // Dwell time (ms)

void main()
{
    SetVel(axis_x, 50); // Set velocity Unit: linear (mm/s), rotary (deg/s)
    SetAcc(axis_x, 2000); // Set acceleration Unit: linear (mm/s^2), rotary (deg/s^2)
    SetDec(axis_x, 2000); // Set deceleration Unit: linear (mm/s^2), rotary (deg/s^2)

    Enable(axis_x); // Enable the axis
    Till(IsEnabled(axis_x)); // Wait until it is enabled
    Sleep(500);

    while(IsEnabled(axis_x)){
        MoveAbs(axis_x, P1); // Move to P1
        Till(IsInPos(axis_x)); // Wait until it is in-position
        Sleep(Dwell_Time); // Dwell time
        MoveAbs(axis_x, P2); // Move to P2
        Till(IsInPos(axis_x)); // Wait until it is in-position
        Sleep(Dwell_Time); // Dwell time
    }
}
```

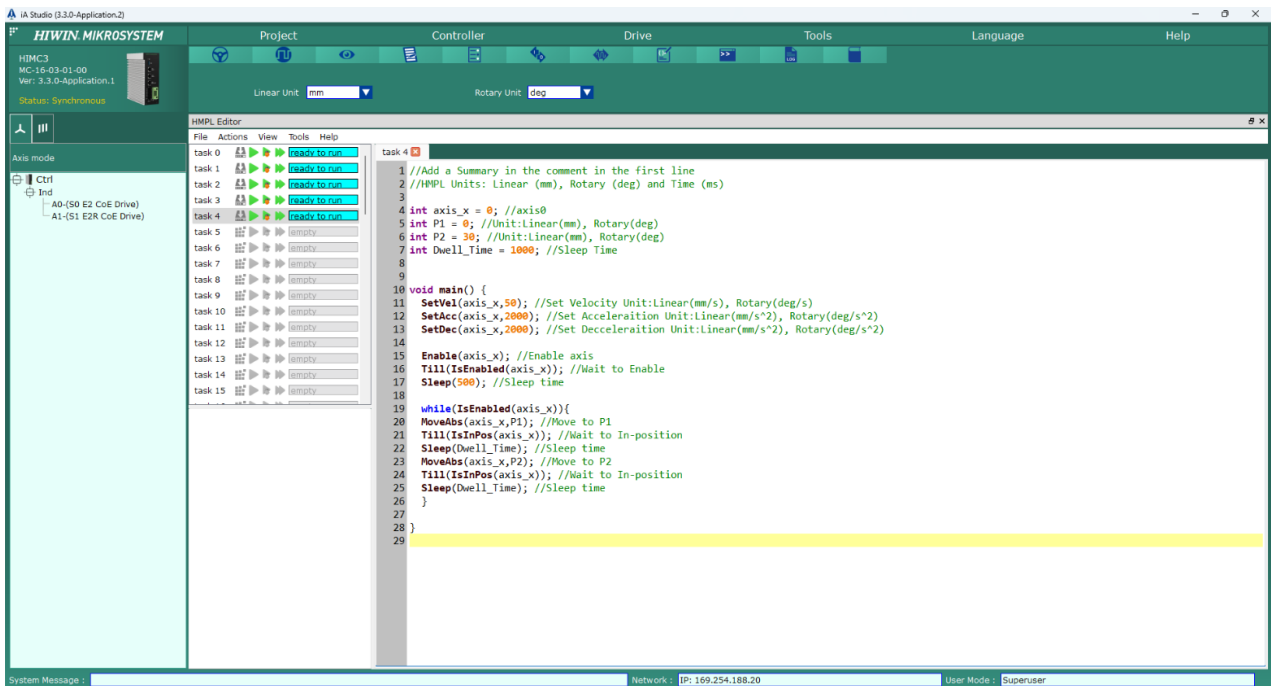


図 3.6.1.1 コントローラーに書き込まれたポイントツーポイントプログラム

3.6.2 ホーミング



重要

ユーザーは、iA Studioのインストールパスからサンプルプログラムを読み込むことができます：
C:\Program Files (x86)\HIWIN MIKROSYSTEM\iA Studio (version)\examples\HMPL

1. 新しいタスクを開始します。
2. タスク内でホーミングプログラムを作成します（ホーミング方法、速度などを設定します）。
3. プログラムをコンパイルし、コントローラーに書き込みます。
4. プログラムを実行します。

```
void main()
{
    int axis_id = 0; // Axis index
    int home_method = 33; // Homing method
    double fast_vel = 20; // Homing velocity (search for Limit Switch) Unit: linear (mm/s), rotary (deg/s)
    double slow_vel = 2; // Homing velocity (search for Index) Unit: linear (mm/s), rotary (deg/s)
    double acc = 2000; // Homing acceleration Unit: linear (mm/s^2), rotary (deg/s^2)
    double home_offsets = 0; // Home offset Unit: linear (mm), rotary (deg)
    int time_out = 10000; // Timeout (ms)

    Enable(axis_id);
    Till(IsEnabled(axis_id));

    SetHomedStatus(axis_id, false); // Set homing status
    SetHomeMethod(axis_id, home_method); // Set homing method
    SetHomeSwitchVel(axis_id, fast_vel); // Set homing velocity (search for Limit Switch)
    SetHomeZeroVel(axis_id, slow_vel); // Set homing velocity (search for Index)
    SetHomeAcc(axis_id, acc); // Set homing acceleration
    SetHomeOffset(axis_id, home_offsets); // Set home offset
    SetHomeTimeout(axis_id, time_out); // Set timeout
    int result = MoveHome(axis_id);
    Till(!IsHoming(axis_id)); // Wait until homing is done

    if (!IsHoming(axis_id)) {
        Print("Home success.");
    } else {
        Print("Home fail:%d.", result);
    }
}
```

```

}
}
    
```

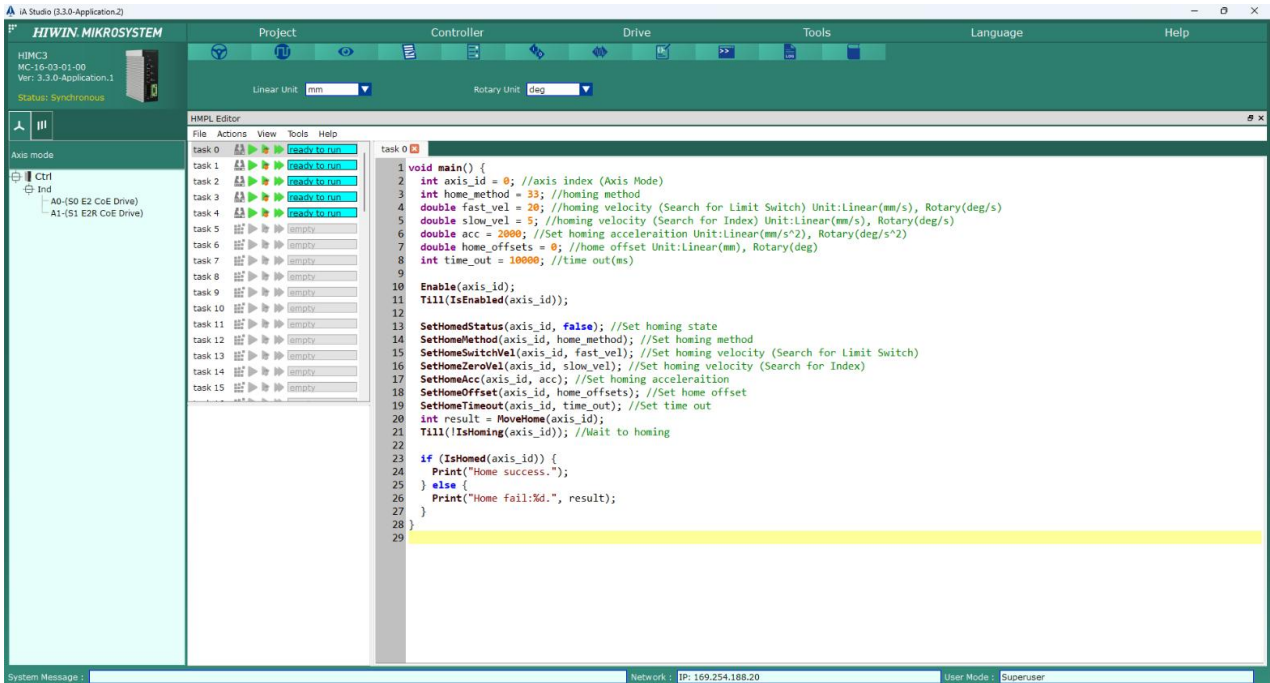


図 3.6.2.1 コントローラーに書き込まれたホーミングプログラム

4. その他のアプリケーション設定

4.1 誤差補正機能の設定.....	4-2
4.1.1 1次元誤差補正.....	4-2
4.1.2 2D 誤差補正.....	4-3

4.1 誤差補正機能の設定

機構の設置および動作中に誤差が発生します。精度を向上させるには、レーザー測定器、ダイヤルゲージ、校正プレートなどの測定ツールを使用して誤差値を取得し、誤差補正を行うことができます。誤差補正機能、誤差補正関数、およびサンプルプログラムの詳細については、「HIMC シリーズ HMPL ユーザーガイド」の第 14 章「動的誤差補正機能」を参照してください。

4.1.1 1 次元誤差補正

1D 誤差補正は、1 つの軸を基準として単軸補正を実行します。以下の例では、X 軸を基準として Y 軸補正を行う例を示します。補正開始位置は-100、間隔は 200、補正ポイントは 5 です。補正値は、-0.2、0.1、-0.3、0.4、-0.1 の順に変化します。詳細なプログラム例については、「HIMC シリーズ HMPL ユーザーガイド」第 14 章「動的誤差補正機能」の「例 1：1 次元動的誤差補正」を参照してください。

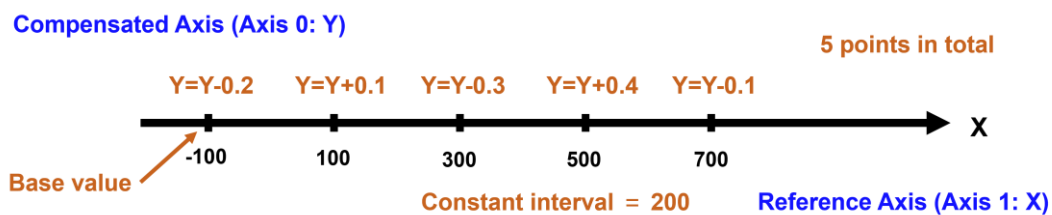


図 4.1.1.1 1 次元誤差補償の位置図

HMPL 経由で補正設定を有効にする：

```
void main() {
    // Set up error map
    double data[5] = {-0.2, 0.1, -0.3, 0.4, -0.1};
    SetUserTable(1, 5, data); // Write data to user table

    SetupComp(
        0, // axis to be compensated
        1, // start index in user table
        -100, // base value
        200, // interval
        5, // number of points (base data included)
        1 // reference axis (input)
    );
}
```

```

EnableComp(0); // Enable compensation on axis 0
Enable(0); // Enable axis 0 to activate compensation
}
    
```

HMPL による補償を無効にする：

```

void main() {
    Disable(0); // Disable axis 0
    Till(!IsEnabled(0));
    DisableComp(0); // Disable compensation on axis 0
}
    
```

4.1.2 2D 誤差補正

2D 誤差補正は、2つの軸を同時に参照する方式で、一般的には XYZ 平面で用いられます。X 軸、Y 軸、Z 軸の補正は、X 軸と Y 軸の両方を参照することで行われます。以下の例では、X 軸と Y 軸の両方を同時に参照して Y 軸補正を行う方法を示します。詳細なプログラム例については、「HIMC シリーズ HMPL ユーザーガイド」第 14 章「動的誤差補正関数」の「例 2：2次元動的誤差補正」を参照してください。



重要

コントローラーにはテーブルが1つしかないため、補正時には両軸とも同じテーブルを共有します。補正值を作成する際は、各軸の開始位置とポイント数をテーブルに書き込む際に注意し、互いの値が上書きされないようにしてください。

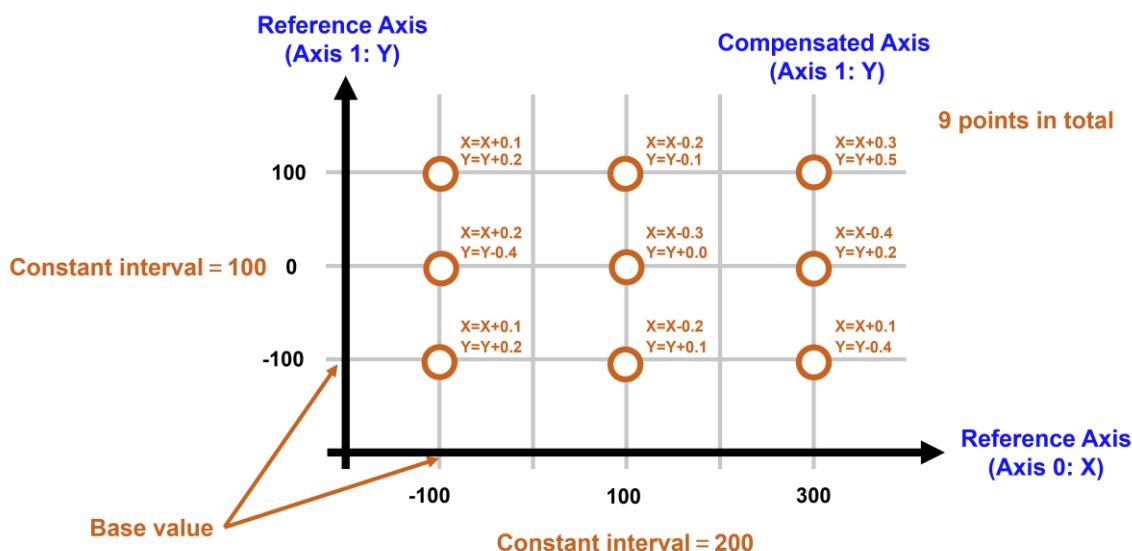


図 4.1.2.1 2次元誤差補償の位置図

HMPL 経由での補正設定と有効化：

```
void main() {
    // Set up error map
    double datay[9] = {0.2, 0.1, -0.4, -0.4, 0.0, 0.2, 0.2, -0.1, 0.5}; // compensation values (from left to right,
bottom to top)
    double datax[9] = {0.1, -0.2, 0.1, 0.2, -0.3, -0.4, 0.1, -0.2, 0.3}; // compensation values (from left to right,
bottom to top)

    SetUserTable(1, 9, datay); // Write data to user table
    SetUserTable(10, 9, datay); // Write data to user table

    double base[2] = {-100, -100};
    double interval[2] = {200, 100};
    int num_pt[2] = {3, 3};
    int ref_axis[2] = {0, 1};
    SetupComp2D(
        1, // axis to be compensated (axis 1: Y)
        1, // start index in user table
        base, // base value
        interval, // interval
        num_pt, // number of points (base data included)
        ref_axis // reference axis (input)
    );

    SetupComp2D(
        0, // axis to be compensated (axis 0: X)
        10, // start index in user table
        base, // base value
        interval, // interval
        num_pt, // number of points (base data included)
        ref_axis // reference axis (input)
    );

    EnableComp(0); // Enable compensation on (axis 0: X)
    Enable(0); // Enable (axis 0: X) to activate compensation
    EnableComp(1); // Enable compensation on (axis 1: Y)
    Enable(1); // Enable (axis 1: Y) to activate compensation
}
```

HMPL による補正を無効にする：

```
void main() {
    Disable(1); // Disable axis 1
    Till(!IsEnabled(1));
    DisableComp(1); // Disable compensation on axis 1
    Disable(0); // Disable axis 0
    Till(!IsEnabled(0));
    DisableComp(0); // Disable compensation on axis 0
}
```

補正值を確認するには、スコープマネージャで変数 `hcv.axis[0].pos_fb_comp` (補正後の位置) と `hcv.axis[0].pos_fb` (元のエンコーダー位置) を監視します。



重要

- (1) 補正位置=元のエンコーダー位置+補正值
- (2) 括弧[]内の数値を変更して、異なる軸を監視します。



図 4.1.2.1 エンコーダーのフィードバックと補正值の監視

(このページはブランクになっています)

Application Note
E Series EtherCat Drive Complete Setup
with HIMC3 iA Studio
バージョン 1.0 2026 年 5 月初版

-
1. HIWIN は HIWIN Mikrosystem Corp., HIWIN Technologies Corp., ハイウィン株式会社の登録商標です。ご自身の権利を保護するため、模倣品を購入することは避けてください。
 2. 実際の製品は、製品改良等に対応するため、このカタログの仕様や写真と異なる場合があります。
 3. HIWIN は「貿易法」および関連規制の下で制限された技術や製品を販売・輸出しません。制限された HIWIN 製品を輸出する際には、関連する法律に従って、所管当局によって承認を受けます。また、核・生物・化学兵器やミサイルの製造または開発に使用することは禁じます。
-