

パソコン組み込み型サーボシステムコントローラ CC-Link IE対応シンプルモーションボード/ MELSOFT EM Software Development Kit

2016年7月

新製品ニュース
SV1607-3

シンプルモーションボード MR-EM340GF、新登場。



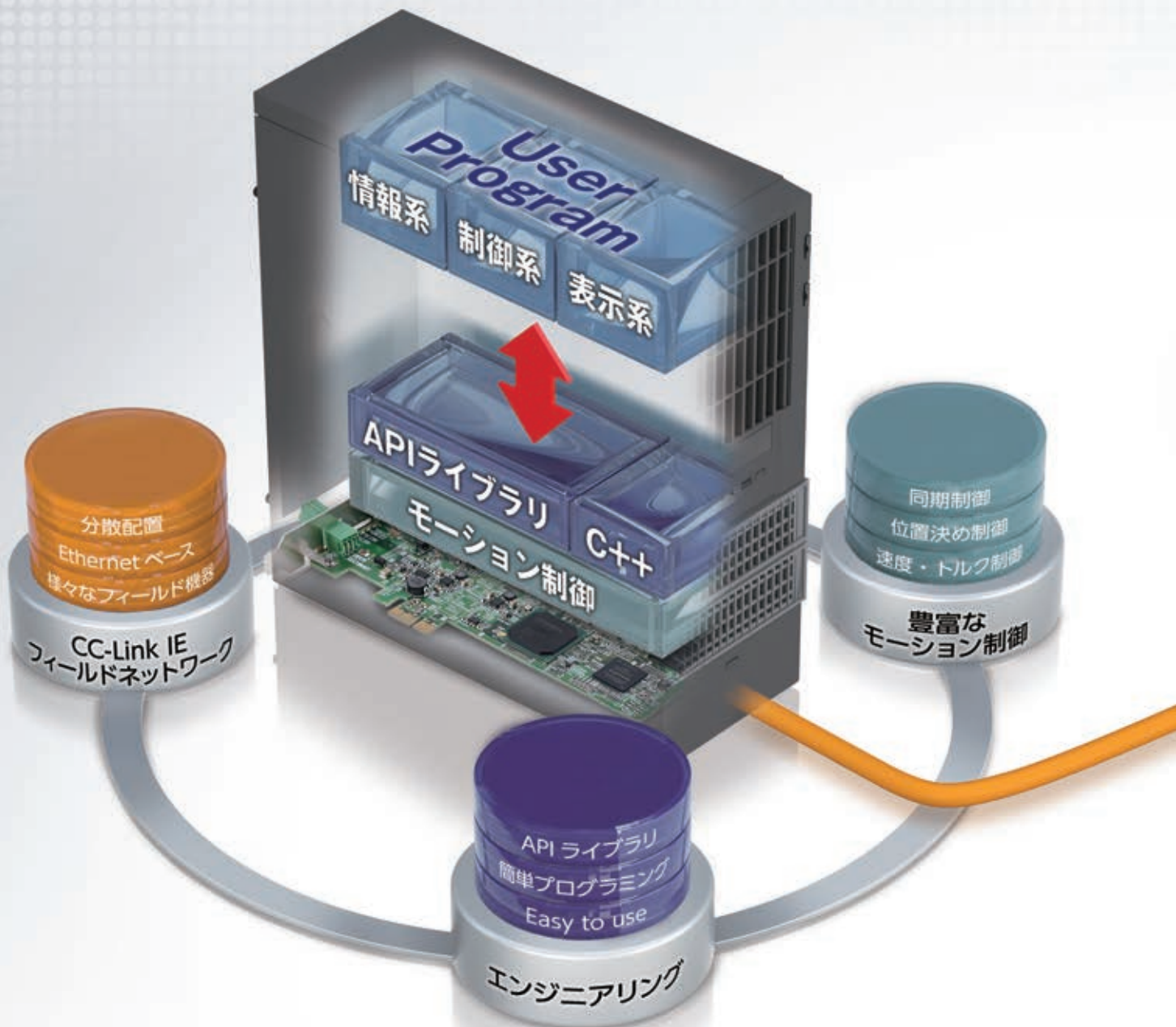
パソコン環境で、CC-Link IEフィールド ネットワークによるモーション制御

- 豊富なモーション制御で最適なソリューションを実現
- 充実した開発・デバッグ環境
MELSOFT EM Software Development Kit
- CC-Link IEフィールドネットワークで様々なフィールド機器を
自由に接続
- Visual C++[®]で簡単プログラミング

パソコン組込み型サーボシステムコントローラ

CC-Link IE対応

シンプルモーションボード





パソコン環境で、豊富な モーション制御を実現

お客様の情報処理(レシピデータ、ロギングデータ)、画像処理を行う産業用パソコン(IPC)にシンプルモーションボードを組み込むことで、豊富なモーション制御が可能になり、様々な装置に適用できます。

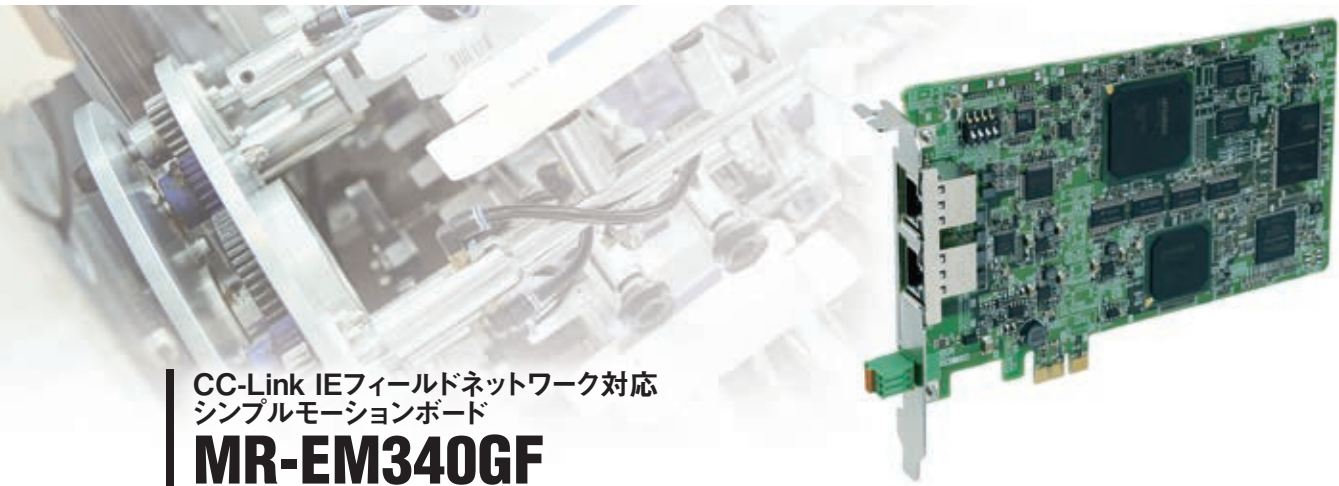
簡単プログラミング & エンジニアリングソフトウェア でデバッグ効率アップ

産業用パソコン(IPC)にAPIライブラリ、PCI Express®デバイスドライバを追加することにより、Visual C++®で簡単にプログラミングができます。
また、エンジニアリングソフトウェアは、サーボアンプ、各種フィールド機器の設定、モニタリングができ、デバッグの効率がアップします。

先進のコンポーネント群を、 ひとつのネットワークに。

CC-Link IEフィールドネットワークはモーション制御に必要な同期性とEthernetの汎用性を両立させたネットワークです。
一つのネットワークでサーボアンプ、I/O、高速カウンタなど様々なフィールド機器を自由に接続できます。

シンプルモーションボード



CC-Link IEフィールドネットワーク対応
シンプルモーションボード

MR-EM340GF

シンプルモーションボードは、PCI Express®対応のパソコンに組み込むことにより、位置決め制御、同期制御、速度・トルク制御などの豊富なモーション制御を実現します。

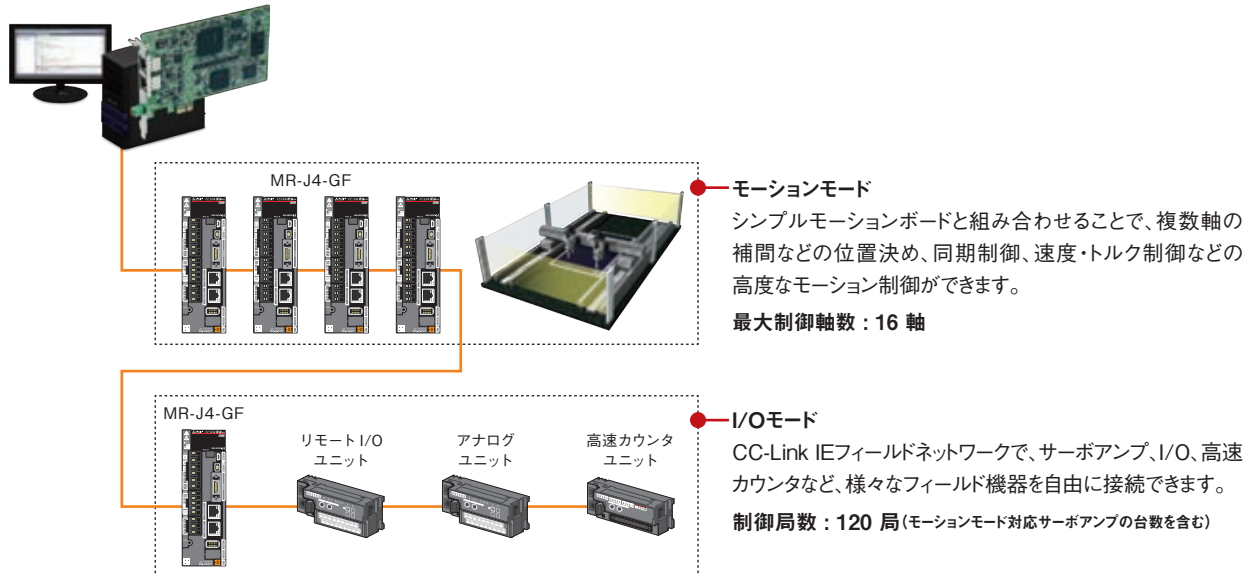
- 一つのネットワークで、サーボアンプ、I/O、高速カウンタなど様々なフィールド機器を自由に接続できます。
- CC-Link IEフィールドネットワークのマスタ局としての機能を搭載しています。
- PCI Express®経由の割り込み機能により、Visual C++®でイベントドリブン方式のプログラミングが可能です。



CC-Link IE

サーボシステムの構築

CC-Link IEフィールドネットワーク対応シンプルモーションボードは、モーション制御だけでなく、CC-Link IEフィールドネットワークのマスタ局としての機能を搭載しています。サーボアンプを含め、最大120局接続可能です。



適用例

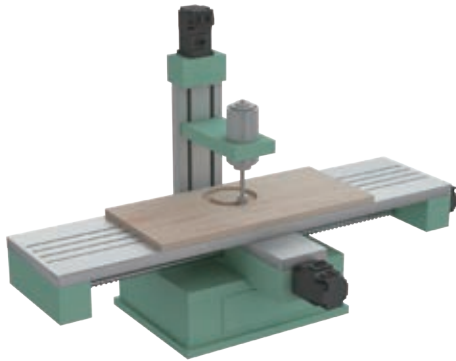
機械に最適な制御や機能を選択することで、最適なソリューションを実現できます。

| | | | |
|---|--|-------------------------------------|---|
| <p>木工機</p> <p>位置決め制御 円弧補間 ヘリカル補間</p> | <p>包装機</p> <p>アドバンス同期 マーク検出</p> | <p>液晶製造装置</p> <p>アドバンス同期</p> | <p>切断機</p> <p>カム自動生成 アドバンス同期</p> |
| <p>検査装置</p> <p>シャッタ出力 位置決め制御</p> | <p>フリップチップボンダ</p> <p>位置決め制御</p> | <p>電子部品組立機</p> <p>位置決め制御</p> | <p>コンバーティング装置</p> <p>速度・トルク制御</p> |

シンプルモーションボード

位置決め制御

- 直線補間、2軸円弧補間、定寸送り、および連続軌跡など、豊富な制御で様々な用途に対応できます。
- APIライブラリで位置決めアドレス、速度などを設定することにより、簡単に自動運転ができます。
- Mコード、スキップ、速度変更、目標位置変更などの充実した補助機能を搭載しています。

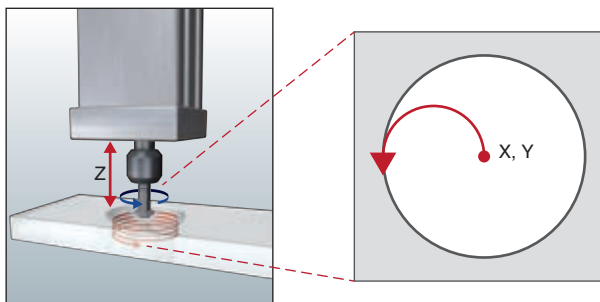


```
void StartPositioningSample( MMC_Axis* axis1 )
{
    unsigned long retCode;

    /* 位置決めデータNo.1による位置決め */
    retCode = axis1->StartPositioning( 1 );
    if( retCode != MMC_OK ) { /* エラー処理 */ }

    /* 位置決め制御完了待機 */
    retCode = axis1->WaitPositioningDone
    ( MMC_POSITIONING_DONE_INP, 10000 );
    if( retCode != MMC_OK ) { /* エラー処理 */ }
}
```

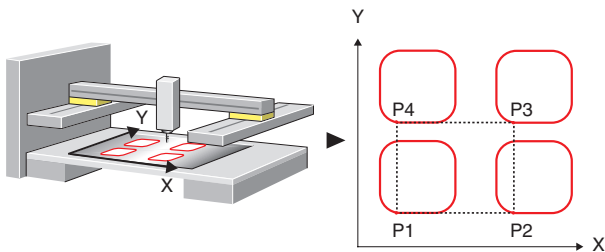
ヘリカル補間



大きな穴をあけるボーリング用途では、X、Y、Zの3軸でヘリカル補間をします。

- フライス加工により、必要とするサイズの穴をXY軸方向に円弧状に開けます。
- 穴の深さはZ軸に沿って、切出位置の偏差を最小限に留めながら、精度の高い加工が可能です。

ブロック始動

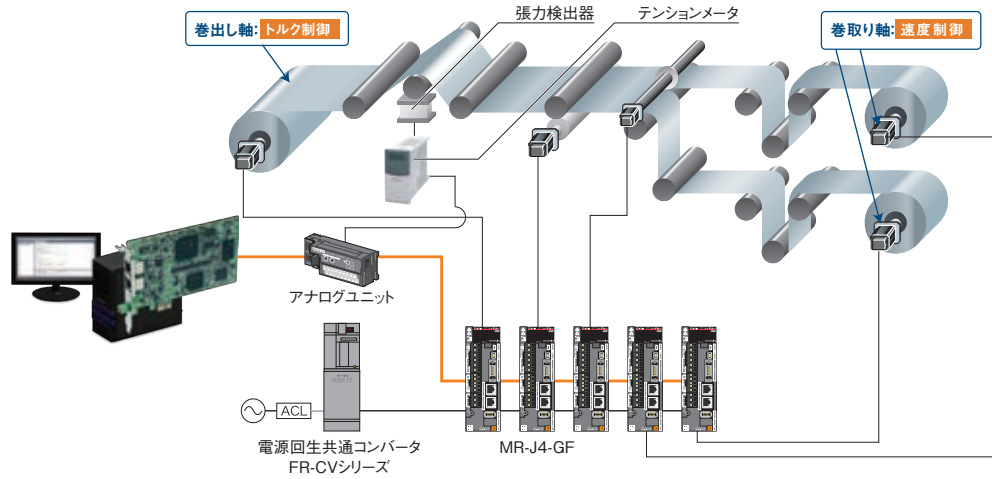


1回の始動で、複数の位置決めデータ群にしたがって順次運転します。同じ軌跡を何度も繰り返す場合などに使用することができます。

始動データNo.1から位置決めを開始して、角丸長方形を4箇所を描画します。

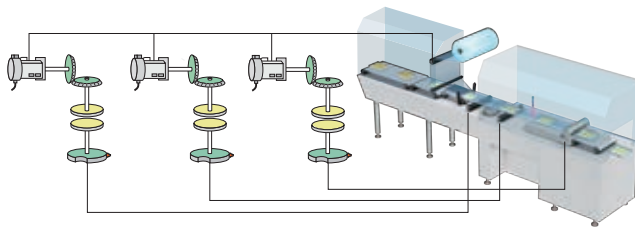
速度・トルク制御

巻出し、巻取りなどの速度・トルク制御にも、シンプルモーションユニットを使用することができます。速度・トルク制御中でも現在位置管理を実施しているため、位置制御に戻した後も絶対位置座標での位置決めが可能です。



アドバンスト同期制御

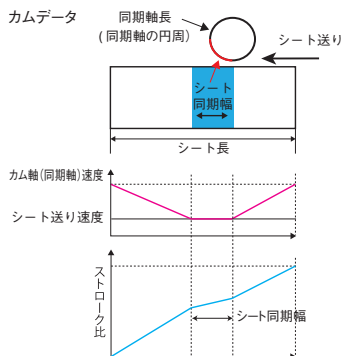
ギア、シャフト、クラッチ、変速機、カムなどの機械機構をソフトウェアに置き換えた同期制御が可能です。カム自動生成機能によりカム生成を簡単に実現します。軸ごとに同期制御の始動、停止ができ、同期制御軸と位置決め制御軸の混在が可能です。



```
void SynchronizationSample
(MMC_Controller *controller,
 MMC_Axis *axis1, MMC_Axis *axis4 )
{
    unsigned long retCode;
    /* 同期制御始動 */
    retCode = axis1->StartSync();
    if( retCode != MMC_OK ) { /* エラー処理 */ }

    /* 仮想サーボ軸のJOG運転を開始 */
    retCode = axis4->StartJog( 20000 );
    if( retCode != MMC_OK ) { /* エラー処理 */ }
    :
    /* 同期制御終了 */
    axis1->StopSync( );
    if( retCode != MMC_OK ) { /* エラー処理 */ }
}
```

カム自動生成



パラメータを設定するだけで、ロータリーカッター用カムデータを自動生成できます。



ソフトウェア開発キット

ソフトウェア開発キット

MELSOFT EM Software Development Kit

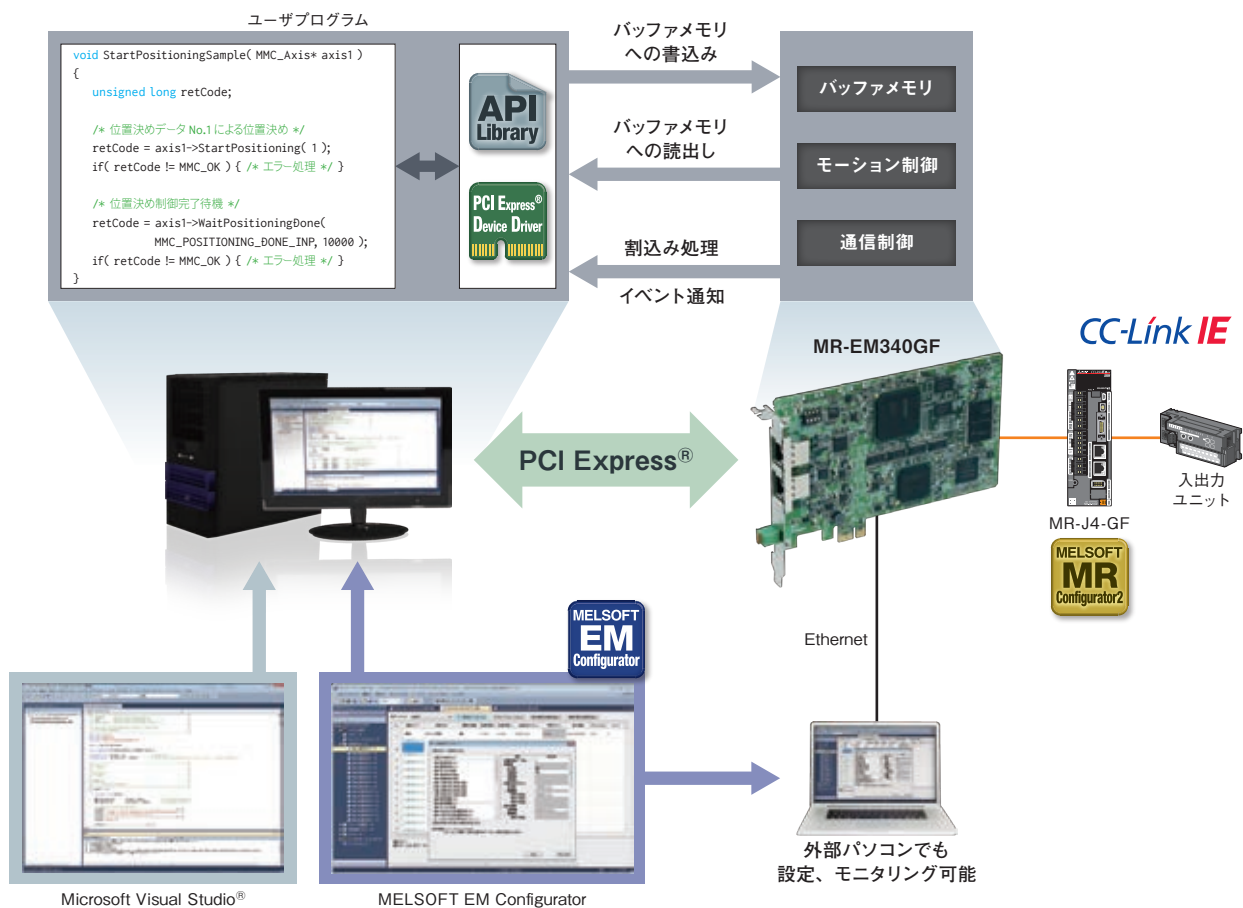


MELSOFT EM Software Development Kitは、シンプルモーションボードのシステム設計・プログラミングから、デバッグ、保守まで、すべてのシーンに対応できるソフトウェア開発キットです。

- | | |
|------------|---------------------------|
| [同梱ソフトウェア] | ●MELSOFT EM Configurator |
| | ●MELSOFT MR Configurator2 |
| | ●APIライブラリ |
| | ●PCI Express®デバイスドライバ |

開発環境、デバッグ環境

Microsoft Visual Studio®のプロジェクトに、モーション制御用APIライブラリを追加してユーザプログラムを作成します。



注：OSおよび開発環境はお客様にて準備してください。



MELSOFT EM Configurator

システム設計・プログラミングからデバッグ、保守まで、すべてのシーンに対応できます。



MELSOFT MR Configurator2

サーボアンプと接続してサーボ調整、モニタ、診断などを簡単に行うことができます。



APIライブラリ

コントローラクラスや軸クラスなどの機能(メソッド)・ラベル(メンバ)を使用して、簡単にVisual C++®でプログラミングができます。



PCI Express®デバイスドライバ

ユーザプログラムからPCI Express®バス経由で、シンプルモーションボードにアクセスするソフトウェアです。



MELSOFT EM Configurator (シンプルモーションボード設定ツール)

システム設計が簡単

システム構成もパラメータもマニュアルレスで設定

- システム構成からパラメータまで、MELSOFT EM Configuratorで設定。
- ワンポイントヘルプによりマニュアルレスで簡単設定。

[CC-Link IE構成]



[電子ギア設定]

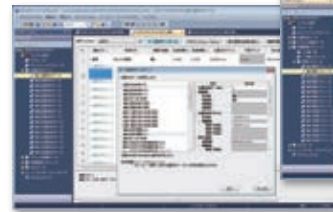


モーション制御が簡単

ワンランク上の位置決めや同期制御の編集

- 豊富な補助機能で位置決めデータを簡単に作成。
- パラメータを設定するだけで、簡単に同期制御。
- カムデータはマウスでドラッグ&ドロップ、数値入力で簡単に作成。

[位置決めデータ]



[同期制御パラメータ]



System Design

Programming

Debug

Maintenance

立上げ簡単

[軸モニタ]



[テスト運転]

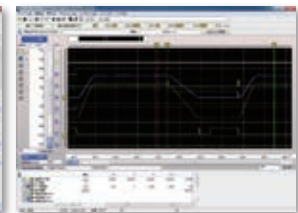


メンテナンスが簡単

[ネットワーク診断]



[デジタルオシロ]



デバッグ・調整を効率アップ

- カスタマイズできる軸モニタで立上げの効率アップ。
- テスト運転により、プログラム作成前にサーボモータの動作チェック。

豊富な診断機能

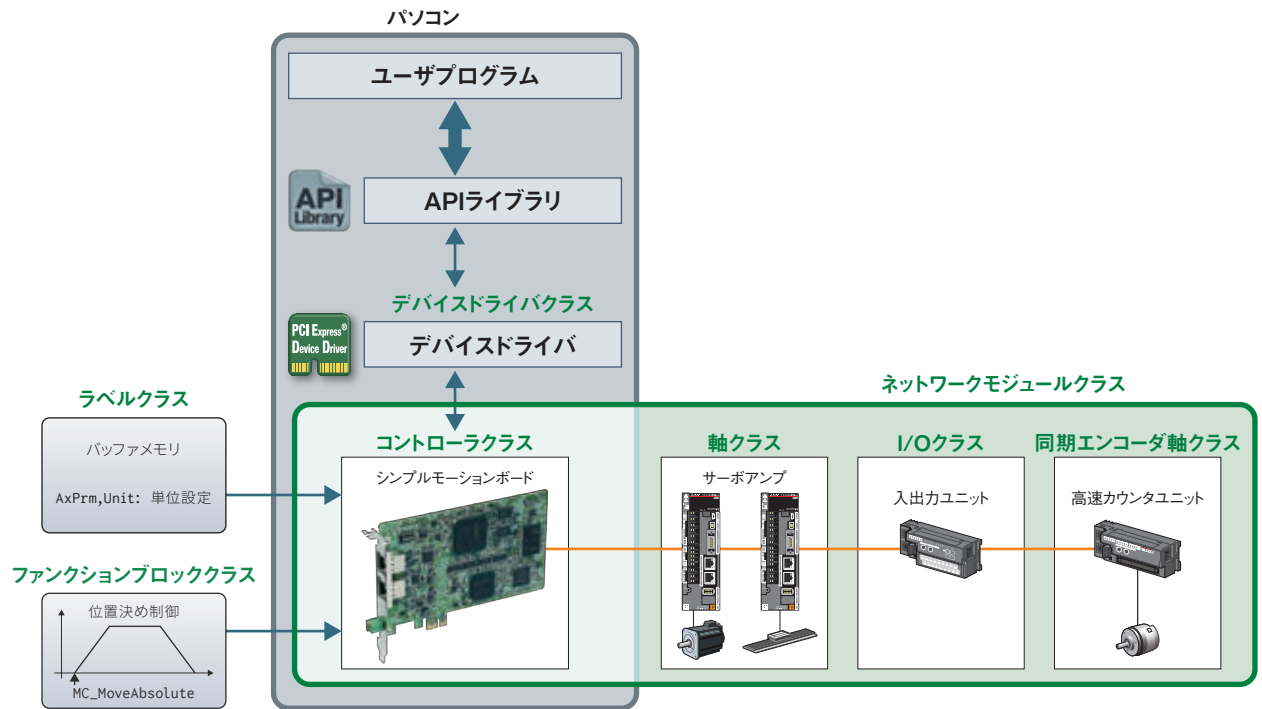
- ネットワーク診断で異常箇所を表示。 対応予定
- 波形表示によりトラブルシューティングをサポート。

APIライブラリ (C++モーションクラスライブラリ)

APIライブラリでは、コントローラクラスや軸クラスなどの機能(メソッド)・ラベル(メンバ)を使用してプログラミングを行います。

- クラスライブラリにより、可読性の高いプログラミングを実現します。
- Microsoft Visual Studio®のIntelliSense®活用により、コーディング時間を短縮します。
- 割り込み機能の条件に任意のビット・データを指定し、イベントドリブン方式のプログラミングとの親和性を向上します。
- PLCopen® Motion Control FBの仕様にあわせ、スキャン処理に適したライブラリを準備しています。

[APIライブラリで使用するクラス概念図]



モーション機能・軸ラベルを使用したプログラミング

```
void ObjectSample( MMC_DeviceDriver *pciDev,
MMC_Controller* controller, MMC_Axis* axis1 )
{
    long data;
    unsigned long retCode;
    /* PCIe デバイスオブジェクト作成 */
    retCode = MmfCreatePciDevice( boardID, &pciDev );

    /* コントローラオブジェクト作成 */
    retCode = MmfCreateEM340GF( pciDev,
(MMC_EM340GF **)&controller );
    /* 軸オブジェクト取得 */
    retCode = controller->GetAxis( 1, &axis1 );

    /* 加速時定数0のパラメータ設定 */
    axis1->AxPrm.AccelerationTime0 = 1000;
    /* 実現在値の取得 */
    data = axis1->AxMntr.ActualPosition;

    /* 位置決めデータNo.1による位置決め */
    retCode = axis1->StartPositioning( 1 );
    if( retCode != MMC_OK ) { /* エラー処理 */ }
}
```

Microsoft Visual Studio® IntelliSense®

```
void Sample( MMC_Axis* axis1 )
{
    axis1->AxMntr.
}
```

軸クラスのモーション機能(メソッド)・軸ラベル(メンバ)の入力候補を表示

イベントドリブン方式のプログラミング [CPUリソースを最大限に活用]

```

void InterruptSample( MMC_Axis* axis1 )
{
    unsigned long retCode;
    MMST_PositioningData positioningData = { 0 };

    /* 位置決めデータNo.1の作成 */
    positioningData.OperationPattern = 0;          /* [Da.1] 運転パターン */
    positioningData.ControlMethod = 0x01;        /* [Da.2] 制御方式 */
    positioningData.AccelerationTimeNo = 0;      /* [Da.3] 加速時間No. */
    positioningData.DecelerationTimeNo = 0;     /* [Da.4] 減速時間No. */
    positioningData.PositioningAddress = -1000000; /* [Da.6] 位置決めアドレス */
    positioningData.CommandSpeed = 20000;       /* [Da.8] 指令速度 */

    /* 位置決めデータNo.1の設定 */
    axis1->SetPositioningData( 1, positioningData );

    /* 位置決め完了の割り込みイベントを非シグナル状態にする */
    retCode = axis1->ResetPositioningDoneIntEvent( MMC_POSITIONING_DONE_INP );
    if( retCode != MMC_OK ) { /* エラー処理 */ }

    /* 位置決め開始(位置決めデータNo.1) */
    retCode = axis1->StartPositioning( 1 );
    if( retCode != MMC_OK ) { /* エラー処理 */ }

    /* 位置決め完了待ち */
    retCode = axis1->WaitPositioningDoneIntEvent( MMC_POSITIONING_DONE_INP, 10000 );
    if( retCode != MMC_OK ) { /* エラー処理 */ }
}

```

位置決め完了の割り込み信号待ち

スキャン処理方式のプログラミング [PLCopen® Motion Control FBのインターフェースにあわせてライブラリ]

スキャン時間を保つためにメソッド内部で待機できない場合や、ST言語からC言語への置換え時に有効です。

```

void FunctionBlockSample( MC_MoveAbsolute *fbMC_MoveAbsolute, AXIS_REF *axis, int phase )
{
    switch( phase )
    {
        case 0:

            /* MC_MoveAbsoluteを実行 */
            fbMC_MoveAbsolute->Axis = axis;          /* 軸情報 */
            fbMC_MoveAbsolute->PositionDataNo = 1;   /* 位置決めデータNo. */
            fbMC_MoveAbsolute->Position = -1000.0;   /* 目標位置 */
            fbMC_MoveAbsolute->Velocity = 20.00;     /* 速度 */
            fbMC_MoveAbsolute->Acceleration = 1000;  /* 加速時間 */
            fbMC_MoveAbsolute->Deceleration = 1000;  /* 減速時間 */
            fbMC_MoveAbsolute->Direction = 1;        /* 回転方向 */
            fbMC_MoveAbsolute->Execute = true;       /* 実行指令ON */
            fbMC_MoveAbsolute->Update();             /* ファンクションブロック実行 */
            if( fbMC_MoveAbsolute->Error ) { /* エラー処理 */ }
            if( fbMC_MoveAbsolute->Done )
            {
                phase = 1;
            }
            break;

        case 1:
            fbMC_MoveAbsolute->Execute = false;     /* 実行指令OFF */
            fbMC_MoveAbsolute->Update();             /* ファンクションブロック実行 */
            phase = 2;
            break;

        case 2:
            :
            :
            break;
    }
}

```

ソフトウェア開発キット

位置決め始動プログラミング

APIライブラリで位置決めデータをシンプルモーションボードに設定するだけで、位置決めを始動できます。

```

void InterruptSample( MMC_Axis* axis1 )
{
    unsigned long retCode;
    MMST_PositioningData positioningData = { 0 };

    /* 位置決めデータNo.1の作成 */
    positioningData.OperationPattern = 0;          /* [Da.1] 運転/パターン */
    positioningData.ControlMethod = 0x01;         /* [Da.2] 制御方式 */
    positioningData.AccelerationTimeNo = 0;       /* [Da.3] 加速時間No. */
    positioningData.DecelerationTimeNo = 0;      /* [Da.4] 減速時間No. */
    positioningData.PositioningAddress = -1000000; /* [Da.6] 位置決めアドレス */
    positioningData.CommandSpeed = 20000;        /* [Da.8] 指令速度 */

    /* 位置決めデータNo.1の設定 */
    axis1->SetPositioningData( 1, positioningData );

    /* 位置決め完了の割り込みイベントを非シグナル状態にする */
    retCode = axis1->ResetPositioningDoneIntEvent( MMC_POSITIONING_DONE_INP );
    if( retCode != MMC_OK ) { /* エラー処理 */ }

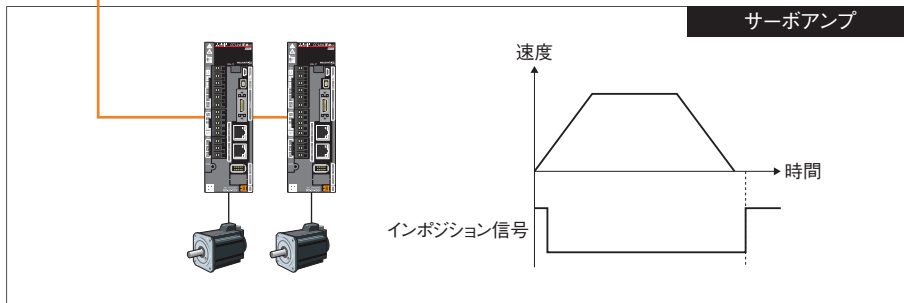
    /* 位置決め開始(位置決めデータNo.1) */
    retCode = axis1->StartPositioning( 1 );
    if( retCode != MMC_OK ) { /* エラー処理 */ }

    /* 位置決め完了待ち */
    retCode = axis1->WaitPositioningDoneIntEvent( MMC_POSITIONING_DONE_INP, 10000 );
    if( retCode != MMC_OK ) { /* エラー処理 */ }
}
    
```

ユーザプログラム

シンプルモーションボード

| No. | 軸1 | | 軸2 | | [Da.5] 位置決め アドレス | [Da.8] 指令速度 | ... |
|-----|----------------------|----------------|-----------------------|-----------------------|------------------------|----------------|-----|
| | [Da.1] 運転 パターン | [Da.2] 制御方式 | [Da.3] 加速時間 No. | [Da.4] 減速時間 No. | | | |
| 1 | 0 | 0x01 | 0:1000 | 0:1000 | -1000000 | 20000 | 0 |
| 2 | 0 | 0x08 | 0:1000 | 0:1000 | 200000 | 10000 | 0 |



■ 主なAPIライブラリー一覧

MMC_Controllerクラス

| オブジェクト取得メソッド群 | |
|--------------------------------|----------------------------|
| GetAxis | 軸クラスのオブジェクトを取得します。 |
| GetSlavelo | I/Oクラスのオブジェクトを取得します。 |
| GetSyncEncoder | 同期エンコーダ軸クラスのオブジェクトを取得します。 |
| システムメソッド群 | |
| ResetController | リモートRESETを実行します。 |
| SetUserProgramReady | ユーザプログラムレディ信号[Y0]をセットします。 |
| 割り込みメソッド群 | |
| SetInterruptParameter | 割り込みパラメータを設定します。 |
| EnableInterrupt | 割り込み出力を有効にします。 |
| DisableInterrupt | 割り込み出力を無効にします。 |
| 同期制御メソッド群 | |
| CalcCamCommandPosition | カム軸送り現在値を計算します。 |
| CalcCamCommandPositionPerCycle | カム軸1サイクル現在値を計算します。 |
| MakeRotaryCutterCam | ロータリーカッタ用カム(中央基準)を自動生成します。 |
| MakeEasyStrokeRatioCam | 簡易ストローク比カムを自動生成します。 |
| MakeAdvancedStrokeRatioCam | 詳細ストローク比カムを自動生成します。 |

MMC_Axisクラス

| 位置決めデータメソッド群 | |
|------------------------------|-----------------------------------|
| SetPositioningData | 位置決めデータを設定します。 |
| SetBlockStartData | ブロック始動データを設定します。 |
| SetBlockConditionData | ブロック始動で使用する条件データを設定します。 |
| GetPositioningData | 位置決めデータを取得します。 |
| GetBlockStartData | ブロック始動データを取得します。 |
| GetBlockConditionData | ブロック始動で使用する条件データを取得します。 |
| 運転メソッド群 | |
| StartPositioning | 位置決め制御の始動を行います。 |
| StartBlockPositioning | 高度な位置決め制御の始動を行います。 |
| StopPositioning | 軸停止を行います。 |
| RestartPositioning | 停止中の軸の再始動を行います。 |
| WaitPositioningDone | 位置決め制御が完了するまで待機します。 |
| ResetPositioningDoneIntEvent | 位置決め完了の割り込みイベントを非シグナル状態に設定します。 |
| SetPositioningDoneIntEvent | 位置決め完了の割り込みイベントをシグナル状態に設定します。 |
| WaitPositioningDoneIntEvent | 位置決め完了の割り込みイベントがシグナル状態になるまで待機します。 |
| StartJog | JOG運転を開始します。 |
| StopJog | JOG運転を停止します。 |
| EnableMPG | 手動ハルス運転の許可を行います。 |
| DisableMPG | 手動ハルス運転の不許可を行います。 |
| ChangeControlMode | 制御モードを切替えます。 |
| 変更メソッド群 | |
| ChangeSpeed | 速度および加減速時間の変更を行います。 |
| ChangePosition | 目標位置および指令速度の変更を行います。 |
| 割り込みメソッド群 | |
| SetInterruptParameter | 割り込みパラメータを設定します。 |
| ResetIntEvent | 割り込みイベントを非シグナル状態に設定します。 |
| SetIntEvent | 割り込みイベントをシグナル状態に設定します。 |
| WaitIntEvent | 割り込みイベントがシグナル状態になるまで待機します。 |
| 同期制御メソッド群 | |
| StartSync | 同期制御を始動します。 |
| StopSync | 同期制御を終了します。 |
| ChangeSyncPosition | 同期制御中に現在値変更を行います。 |
| MoveCamPosition | 同期制御中にカム軸の移動を行います。 |

MC_FunctionBlockクラス

| | |
|------------------|---|
| MC_Power | 指定した軸のサーボアンプを運転可能状態に切替えます。 |
| MCv_Home | 指定した軸の原点復帰を実行します。 |
| MC_Stop | 指定した軸を停止させます。 |
| MC_MoveAbsolute | 指定した軸の絶対位置の目標位置を指定し、位置決めを実行します。 |
| MC_MoveRelative | 現在位置から指定された距離を移動します。 |
| MC_Reset | 指定した軸のエラー、ワーニングを解除します。 |
| MC_MoveAdditive | 指定した軸の直前の位置決め指令に指定された相対位置を付加して位置決めを実行します。 |
| MC_MoveVelocity | 指定した軸を指定された速度で速度制御を実行します。 |
| MC_TorqueControl | 指定した軸を指定されたトルクでトルク制御します。 |
| MC_SetPosition | 指定した軸の現在位置(指令位置、フィードバック位置)を変更します。 |
| MC_SetOverride | 指定した軸の目標速度の変更を実行します。 |



CC-Link IEフィールドネットワーク対応
サーボアンプ

MR-J4-GF MR-J4-GF-RJ

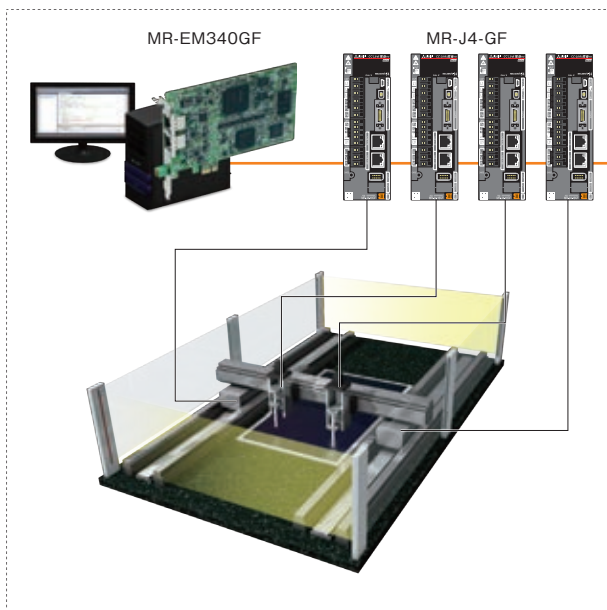
シンプルモーションボードと組み合わせて、最適なソリューションを実現し、生産性向上に貢献します。

- 業界をリードする基本性能
業界最高レベルの基本性能により、装置の高タクト化に貢献します。
- 高度なサーボゲイン調整を簡単に実現
先進の振動抑制機能を手軽に駆使することで、マシンの性能を最大限に発揮させます。
- 幅広い容量・シリーズをラインアップ
回転型サーボモータからリニアサーボモータ、ダイレクトドライブモータまで幅広く対応し、装置のパフォーマンスを大幅に向上できます。
- 予防保全
ネットワーク経由でサーボアンプの情報を読み出すことができ、機械診断などの予防保全に役立ちます。

■ 制御モード

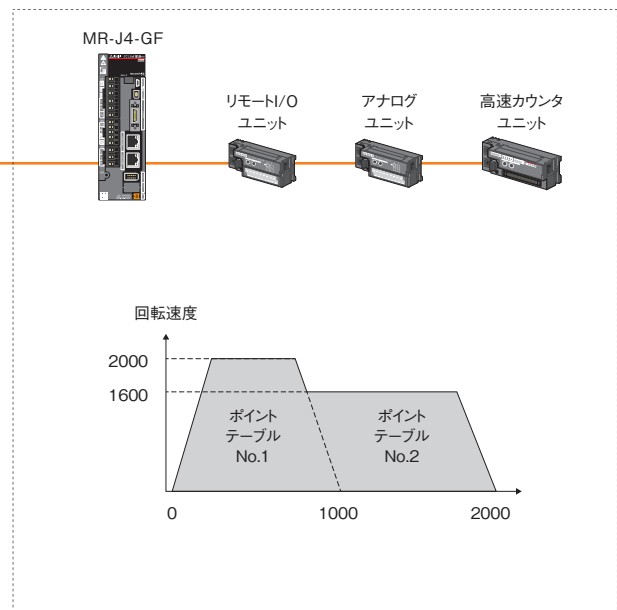
複数軸の位置決め、同期制御など幅広いモーション制御ができるモーションモードと、1軸での位置決めができるI/Oモードがあり、用途により使い分けができます。

モーションモード



モーションモードは複数軸の位置決め、同期制御、速度・トルク制御などの高度なモーション制御ができます。

I/Oモード

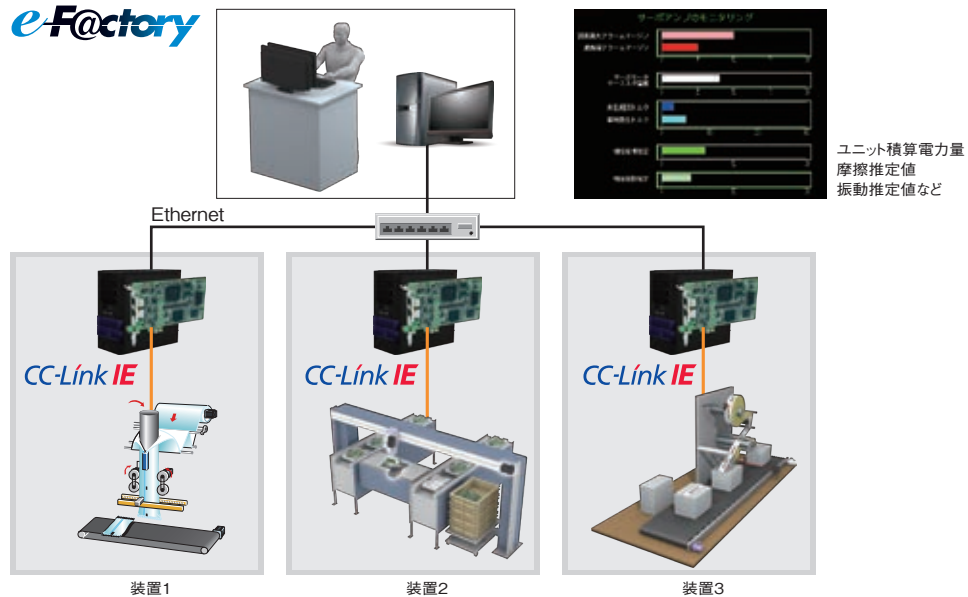


I/Oモードはサーボアンプに内蔵している位置決め機能を使って、ベルトコンベア、回転テーブル、ボールねじなどを簡単に駆動できます。

ITシステムとの情報連携

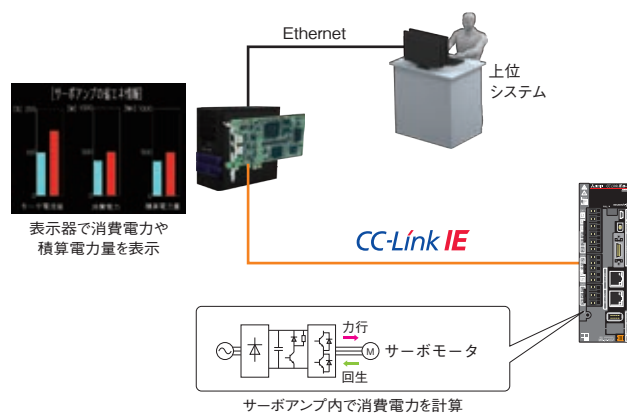
各装置のサーボアンプ、サーボモータの情報をCC-Link IEフィールドネットワーク経由で収集できます。収集した情報を一括管理することにより、ライン全体の状況を把握することができます。CC-Link IEフィールドネットワーク対応サーボシステムは、装置のIoT*1化を支援します。

*1: IoT (Internet of Things)



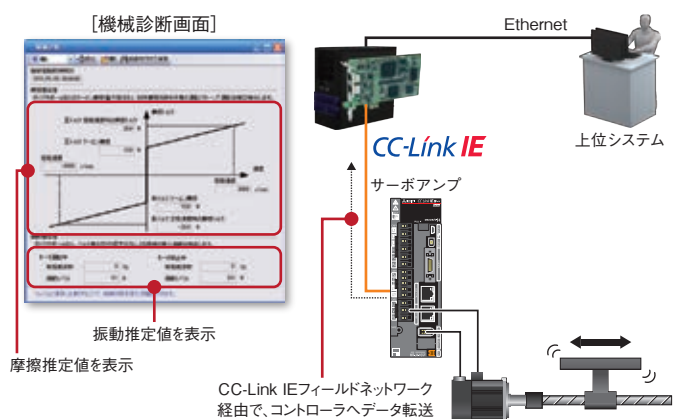
サーボデータのモニタリング

運転中に最大50項目のサーボデータを逐次変更・モニタリングが可能です。CC-Link IEフィールドネットワーク経由で取得したサーボアンプ、サーボモータの運転状態を上位システムへ転送・表示できます。



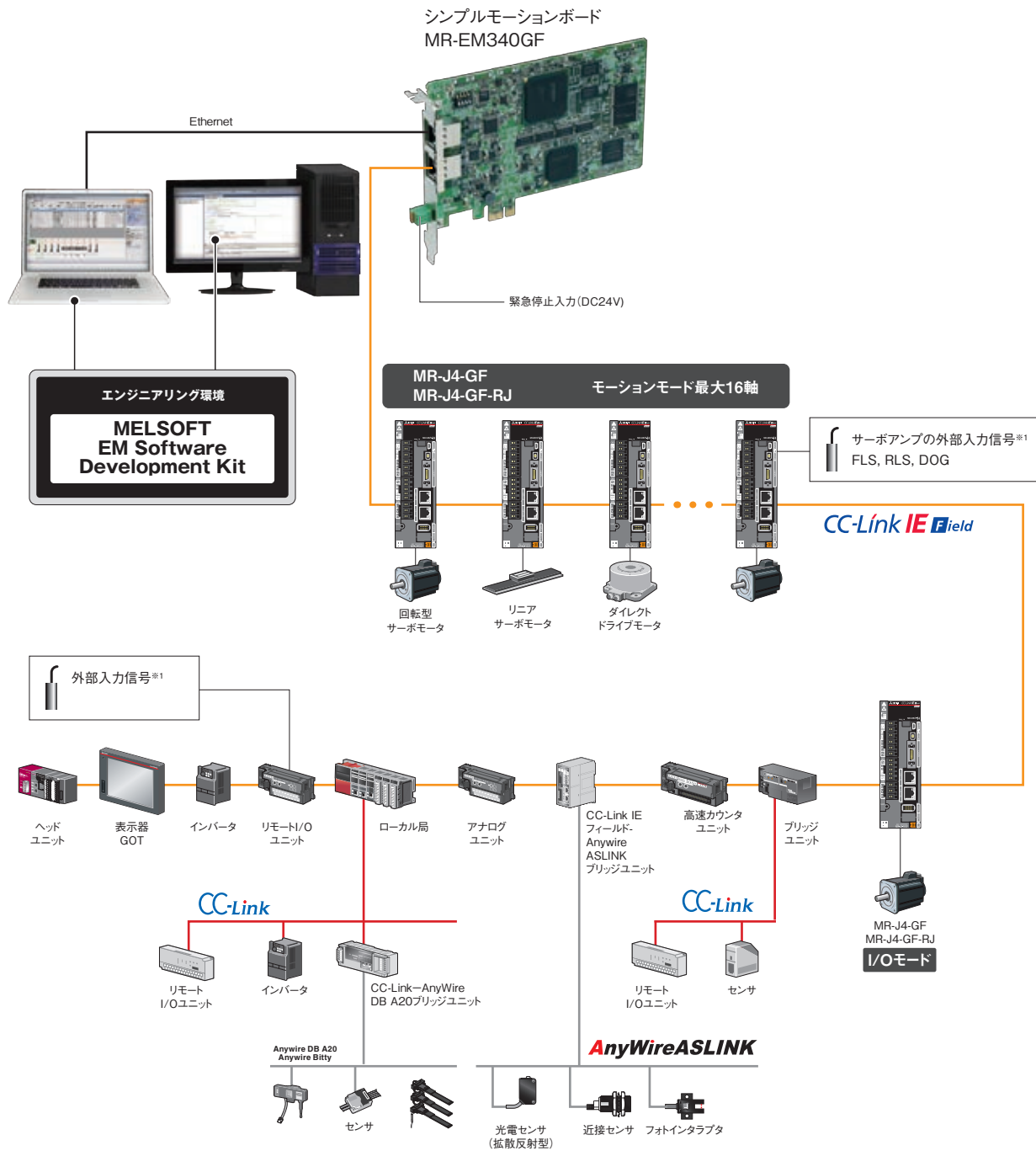
予防保全

サーボアンプの内部データから、装置の摩擦、負荷慣性モーメント、アンバランストルク、振動成分の変化を解析し、機械部品（ボールねじ、ガイド、軸受、ベルトなど）の変化を検出できます。駆動部のタイムリーなメンテナンスを支援します。また、上位システムへデータを転送することができ、装置全体の管理ができます。



システム構成

システム構成



※1: 外部入力信号 (FLS, RLS, DOG) の入力先はパラメータで切替えます。

スレーブ局: 120局 (モーションモード対応サーボアンプの台数を含む)
 注: スター型配線の場合、スイッチングハブが必要です。

■制御仕様

| 項目 | | 仕様 | |
|-------------------------|------------------------------------|---|-----------|
| | | MR-EM340GF | |
| 最大制御軸数(仮想サーボアンプ軸を含む) | | 16軸 | |
| 演算周期(演算周期設定) | | 0.5ms, 1.0ms, 2.0ms, 4.0ms | |
| 補間機能 | | 直線補間(最大4軸), 2軸円弧補間, ヘリカル補間(3軸) | |
| 制御方式 | | 位置決め制御, 軌跡制御(直線, 円弧, ヘリカルとも設定可), 速度制御, 速度・トルク制御 | |
| 加減速処理 | | 台形加減速, S字加減速 | |
| 補正機能 | | 電子ギア, バックラッシュ, 近傍通過 | |
| 同期制御 | | 同期エンコーダ入力, カム, 位相補正, カム自動生成 | |
| 制御単位 | | mm, inch, degree, pulse | |
| 位置決めデータ数 | | 600データ/軸 | |
| バックアップ | | パラメータ, 位置決めデータ, ブロック始動データはフラッシュROMで保存可(バッテリーレス) | |
| 原点復帰 | 原点復帰方式 | ドライバ原点復帰式 | |
| | 高速原点復帰 | あり | |
| | 補助機能 | あり(サーボアンプの補助機能を使用) | |
| 位置決め制御 | 直線制御 | 直線補間制御(最大4軸) ^{※1} (合成速度, 基準軸速度) | |
| | 定寸送り | 定寸送り制御 | |
| | 2軸円弧補間 | 補助点指定, 中心点指定 | |
| | 速度制御 | 速度制御 | |
| | 速度位置切換え | INCモード, ABSモード | |
| | 位置速度切換え | INCモード | |
| | 現在値変更 | 位置決めデータ指定, 現在値変更用始動番号指定 | |
| | NOP命令 | あり | |
| | JUMP命令 | 条件付き, 無条件 | |
| | LOOP, LEND | あり | |
| 高度な位置決め | ブロック始動, 条件始動, ウェイト始動, 同時始動, 繰り返し始動 | | |
| 手動制御 | JOG運転 | あり | |
| | インテグレーション | あり | |
| | 手動パルス | 1台接続可能(インクリメンタル), 単位倍率(1~10000倍) リンクデバイス経由 | |
| 拡張制御 | 速度・トルク制御 | 位置ループを含まない速度制御, トルク制御 | |
| 絶対位置システム | | サーボアンプにバッテリー装着にて対応可能 | |
| 同期エンコーダインタフェース | | 16CH | |
| 制御を制限する機能 | バッファメモリ経由 | あり(インクリメンタル) | |
| | リンクデバイス | あり(インクリメンタル) | |
| | サーボアンプ経由 | 16CH | |
| | 速度制限 | 速度制限値, JOG速度制限値 | |
| | トルク制限 | トルク制限値同一指定, トルク制限値個別指定 | |
| | | 緊急停止 | 内蔵インタフェース |
| バッファメモリ経由 | | | あり |
| リンクデバイス経由 | あり | | |
| ソフトウェアストロークリミット | 送り現在値での可動範囲チェック, 送り機械値で可動範囲チェック | | |
| ハードウェアストロークリミット | あり | | |
| 制御内容を変更する機能 | 速度変更 | あり | |
| | オーバーライド | 0~300% | |
| | 加減速時間変更 | あり | |
| | トルク変更 | あり | |
| | 目標位置変更 | 目標位置のアドレス, 目標位置への速度の変更が可能 | |
| その他機能 | Mコード出力 | WITHモード/AFTERモード | |
| | ステップ機能 | 減速単位ステップ, データNo. 単位ステップ | |
| | スキップ機能 | バッファメモリ経由, 外部指令信号経由 | |
| | ティーチング機能 | あり | |
| パラメータ初期化機能 | | あり | |
| 外部入力信号設定機能 | バッファメモリ経由 | あり | |
| | リンクデバイス | あり | |
| | サーボアンプ経由 | あり | |
| アンプなし運転機能(仮想サーボアンプ機能) | | あり | |
| マーク検出機能 | | 常時モード, 指定回数モード, リングバッファモード | |
| | マーク検出信号 | 最大16点 ^{※3} | |
| | マーク検出設定 | 16設定 | |
| デジタルオシロ機能 ^{※2} | ビットデータ | 16CH | |
| | ワードデータ | 16CH | |

※1: 4軸直線補間制御は基準軸速度のみ有効です。

※2: ワード8CH, ビット8CHまでのデータをリアルタイム波形表示可能です。

※3: 当社製のリモート入力ユニットが必要です。

仕様

■ シンプルモーションボード仕様

| 項 目 | | 仕 様 |
|-------------------|---------------------------|---|
| | | MR-EM340GF |
| サーボアンプ接続方式 | | CC-Link IEフィールドネットワーク |
| 最大局間距離[m] | | 100 |
| 周辺装置インタフェース | | Ethernet(100BASE) |
| 緊急停止入力信号 (EMI) | 入力点数 | 1点 |
| | 入力方式 | プラスコモン/マイナスコモン共用(フォトカプラ絶縁) |
| | 定格入力電圧/電流 | DC24V/約2.4mA |
| | 使用電圧範囲 | DC20.4~26.4V(DC24V +10%/−15%, リップル率 5%以内) |
| | ON電圧/電流 | DC17.5V 以上/2.0mA 以上 |
| | OFF電圧/電流 | DC1.8V 以下/0.18mA 以下 |
| | 入力抵抗 | 約10kΩ |
| | 応答時間 | 1ms 以下(OFF→ON, ON→OFF) |
| | 使用電線サイズ[mm ²] | 0.08~0.5(AWG20~28) |
| パソコン1台当たりの装着台数 | | 4台 |
| バス仕様 | | PCI Express® 2.0 × 1 |
| | 形状[mm] | ショートサイズ(167.65 × 111.15) |
| 電源電圧 | | DC12V / DC3.3V |
| 消費電力[A] | DC12V | 0.4 |
| | DC3.3V | 0.6 |
| 質量[kg] | | 0.13 |

■ MELSOFT EM Software Development Kit の動作環境

| 項 目 | 内 容 | |
|-------------|--|--|
| パソコン本体 | パソコン | Microsoft® Windows® が動作するパーソナルコンピュータ |
| | OS | Microsoft® Windows® 8.1 (Pro, Enterprise)日本語版/英語版(64bit/32bit) Microsoft® Windows® 7 (Professional, Ultimate, Enterprise)日本語版/英語版(64bit/32bit) [Service Pack 1] |
| | CPU | デスクトップパソコン: インテル® Celeron® プロセッサ 2.8GHz 以上推奨 ノートパソコン: インテル® Pentium® Mプロセッサ 1.7GHz 以上推奨 |
| | 必要メモリ | 1GB以上推奨 (32ビット版の場合) 2GB以上推奨 (64ビット版の場合) |
| ハードディスク空き容量 | インストール時: HDDの空き容量 3GB 以上 動作時: 仮想メモリの空き容量 512MB 以上 | |
| ディスクドライブ | DVD-ROM 対応ディスクドライブ | |
| ディスプレイ | 解像度 1024 × 768ドット 以上 | |
| 通信用インタフェース | PCI Express®バス Ethernetポート | |

■ 開発環境

| 項 目 | 内 容 |
|--------------|---|
| ユーザプログラム動作OS | MELSOFT EM Software Development Kitの動作環境と同じ |
| ソフトウェア開発環境 | Microsoft® Visual C++® 2013/2012/2010 |
| APIライブラリ | クラスライブラリ形式(C++コンパイルのみ対応) |

■CC-Link IEフィールドネットワーク性能仕様

| 項目 | | 仕様 | |
|--------------------|---------------|--|-----------------------|
| | | MR-EM340GF | |
| 1ネットワークあたりの最大リンク点数 | RX | 16k点 (16384点, 2k/バイト) | |
| | RY | 16k点 (16384点, 2k/バイト) | |
| | RWr | 8k点 (8192点, 16k/バイト) | |
| | RWw | 8k点 (8192点, 16k/バイト) | |
| 1局あたりの最大リンク点数 | マスタ局 | RX | 16k点 (16384点, 2k/バイト) |
| | | RY | 16k点 (16384点, 2k/バイト) |
| | | RWr | 8k点 (8192点, 16k/バイト) |
| | | RWw | 8k点 (8192点, 16k/バイト) |
| | ローカル局 | RX | 2k点 (2048点, 256/バイト) |
| | | RY | 2k点 (2048点, 256/バイト) |
| | | RWr | 256点, 512バイト |
| | | RWw | 256点, 512バイト |
| | インテリジェントデバイス局 | RX | 2k点 (2048点, 256/バイト) |
| | | RY | 2k点 (2048点, 256/バイト) |
| | | RWr | 256点, 512バイト |
| | | RWw | 256点, 512バイト |
| リモートデバイス局 | RX | 128点, 16/バイト | |
| | RY | 128点, 16/バイト | |
| | RWr | 64点, 128/バイト | |
| | RWw | 64点, 128/バイト | |
| Ethernet部 | 通信速度 | 1Gbps | |
| | 接続ケーブル | 1000BASE-Tの規格を満たすEthernetケーブル※1: カテゴリ5e 以上 (二重シールド付・STP) ストレートケーブル | |
| | 最大局間距離[m] | 最大100(ANSI/TIA/EIA-568-B(Category 5e)に準拠) | |
| | 伝送路形式 | ライン型, スター型(ライン型とスター型の混在も可能) | |
| 総延長距離 | ライン接続時[m] | 12000(マスタ局1台, スレーブ局120台接続時) | |
| | スター接続時※2 | システム構成による | |
| 1ネットワークあたりの最大接続局数 | | 121台(マスタ局1台, スレーブ局120台) | |
| 最大ネットワーク数 | | 239 | |

※1: CC-Link IEフィールドネットワークの配線には、CC-Link 協会の推奨配線部品を使用してください。
 CC-Link IEフィールドネットワークには、CC-Link IEコントローラネットワーク用のケーブルは使用できません。

※2: スター型配線の場合、スイッチングハブが必要です。

Ethernetケーブル仕様

| 項目 | | 内容 |
|--------------|------|---|
| Ethernetケーブル | 規格 | カテゴリ5e以上(二重シールド付き・STP) ストレートケーブル 下記を満たすケーブル IEEE802.3 1000BASE-T ANSI/TIA/EIA-568-B(Category 5e) |
| | コネクタ | シールド付きRJ-45 |

■紹介品

Ethernetケーブル

| 品名 | 仕様 | | |
|--------------|--------|----------------|----------------------|
| Ethernetケーブル | 屋内用 | SC-E5EW-S_M | 二重シールド付きケーブル(カテゴリ5e) |
| | 屋内稼働部用 | SC-E5EW-S_M-MV | |
| | 屋内・屋外用 | SC-E5EW-S_M-L | |

価格及び詳細については、下記の三菱電機システムサービス株式会社までお問い合わせください。

■北日本支社 Tel: 022-353-7814 ■北海道支店 Tel: 011-890-7515 ■東京機電支社 Tel: 03-3454-5511 ■中部支社 Tel: 052-722-7602
 ■北陸支店 Tel: 076-252-9519 ■関西支店 Tel: 06-6454-0281 ■中四国支社 Tel: 082-285-2111 ■四国支店 Tel: 087-831-3186
 ■九州支社 Tel: 092-483-8208

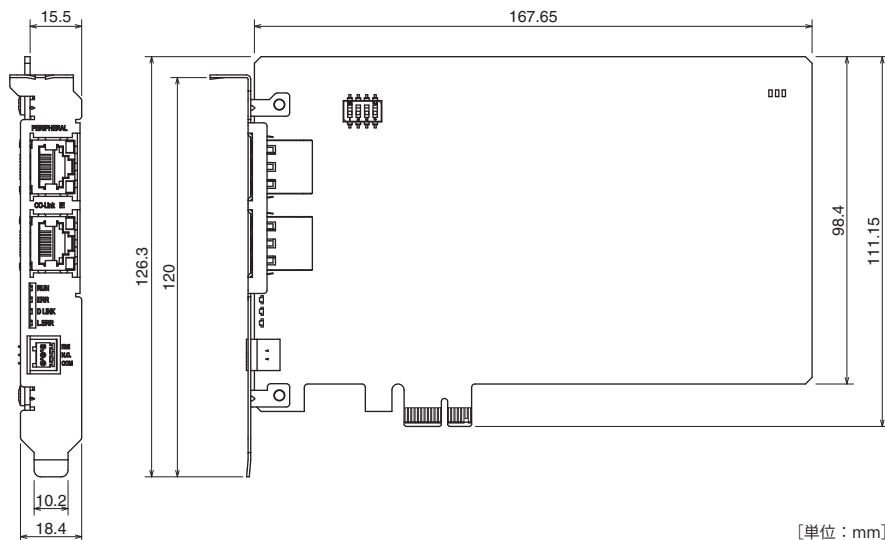
URL: <http://www.melco.co.jp/business/>

Microsoft, Windows, Visual C++, Visual StudioおよびIntelliSenseは、米国Microsoft Corporation の米国およびその他の国における登録商標です。
 Intel, Pentium, Celeronは、Intel Corporationの米国およびその他の国における登録商標です。
 PCI ExpressはPCI-SIG社の登録商標です。
 Ethernetは、米国Xerox Corporation の登録商標です。
 その他、本文中における会社名、商品名は、各社の商標または登録商標です。

パソコン組み込み型サーボシステムコントローラ CC-Link IE対応シンプルモーションボード/ MELSOFT EM Software Development Kit

■外形図

シンプルモーションボード MR-EM340GF



[単位: mm]

■機器一覧

| 品名 | 形名 | 仕様 | 適用海外規格 | 標準価格(円) |
|-------------------------------------|----------------|---|------------|---------|
| シンプルモーションボード | MR-EM340GF | 最大16軸 | CE, UL, KC | — |
| MELSOFT EM Software Development Kit | SW1DND-EMSDK-B | <ul style="list-style-type: none"> ● MELSOFT EM Configurator(設定ツール) ● MELSOFT MR Configurator2 ● APIライブラリ(C++モーションクラスライブラリ) ● PCI Express®デバイスドライバ(割り込み用ドライバ含む) | — | — |

三菱 FA

検索

www.MitsubishiElectric.co.jp/fa

メンバー
登録無料!

インターネットによる情報サービス「三菱電機FAサイト」

三菱電機FAサイトでは、製品や事例などの技術情報に加え、トレーニングスクール情報や各種お問い合わせ窓口をご提供しています。また、メンバー登録いただくとマニュアルやCADデータ等のダウンロード、eラーニングなどの各種サービスをご利用いただけます。

三菱電機株式会社

〒100-8310 東京都千代田区丸の内2-7-3(東京ビル)

本社機器営業部… (03)3218-6740
 北海道支社 …… (011)212-3793
 東北支社 …… (022)216-4546
 関東支社 …… (048)600-5835
 新潟支店 …… (025)241-7227

神奈川支社 …… (045)224-2623
 北陸支社 …… (076)233-5502
 中部支社 …… (052)565-3326
 豊田支店 …… (0565)34-4112
 関西支社 …… (06)6486-4120

中国支社 …… (082)248-5445
 四国支社 …… (087)825-0055
 九州支社 …… (092)721-2251

商標、登録商標について

本文中における会社名、商品名は、各社の商標または登録商標です。

▲安全に関するご注意

本資料に記載された製品を正しくお使いいただくためご使用前に必ず「マニュアル」をお読みください。