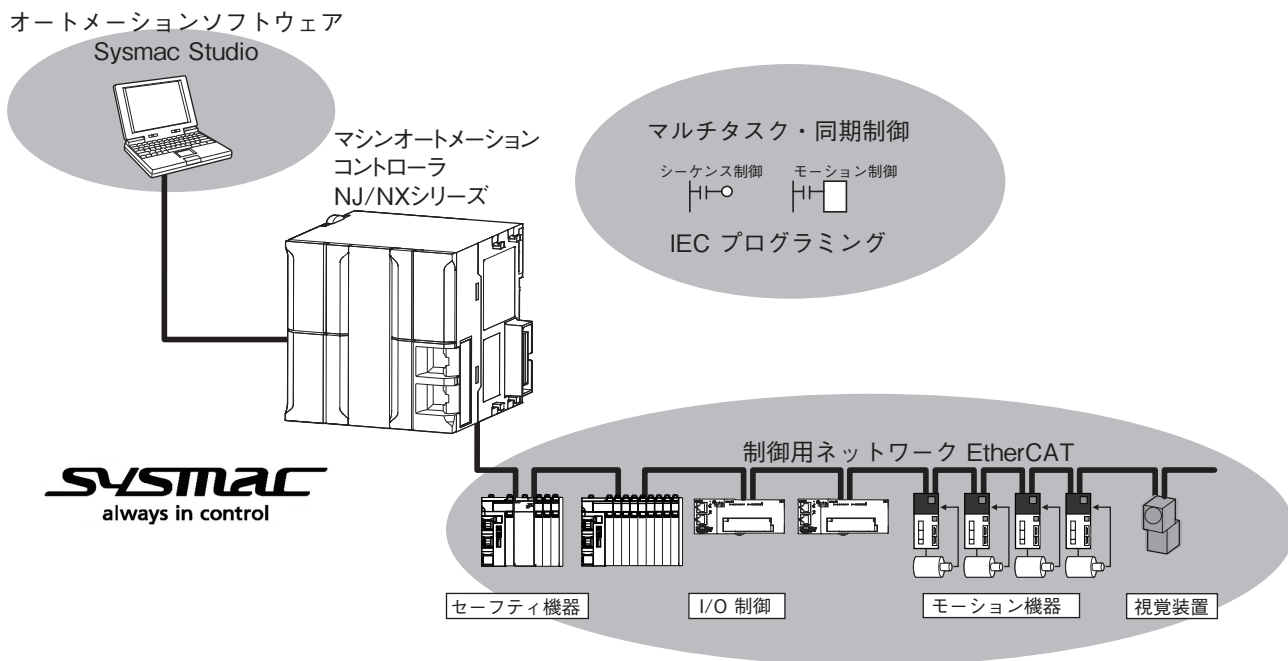


## 概要

### マシンオートメーションコントローラとは

マシンオートメーションコントローラNJ/NX シリーズは、機械制御に必要な高度な機能・高速性能と、産業用コントローラとしての安全性・信頼性・保守性を両立した、次世代コントローラです。

従来のプログラマブルコントローラの機能を包含し、モーション制御に必要な各種機能を付加した統合型コントローラとして、セーフティ機器、視覚装置、モーション機器、I/O などの入出力機器を、高速なEtherCAT® 上で同期して制御できます。



Sysmacは、オムロン株式会社製FA機器製品の日本及びその他の国における商標または登録商標です。  
Microsoft、Visual Basic および Windows は米国 Microsoft Corporationの米国、日本およびその他の国における登録商標または商標です。  
ATI™、Radeon™は、米国Advanced Micro Devices, Inc.の商標です。  
NVIDIA、NVIDIAロゴ、GeForce、GeForceロゴは、米国およびその他の国におけるNVIDIA Corporationの登録商標または商標です。  
EtherCAT®は、Beckhoff Automation GmbH(ドイツ)よりライセンスを受けた特許取得済み技術であり登録商標です。  
EtherNet/IP™およびDeviceNet™はODVAの商標です。  
Celeron、Intel、Intel Coreは、米国およびその他の国におけるIntel Corporationの商標です。  
その他、記載されている会社名と製品名などにつきましては、各社の登録商標または商標です。

## 特長

### ハードウェア上の特長

#### 制御用ネットワーク EtherCAT を標準装備

CPU ユニットはEtherCAT 通信マスタ機能ポートを標準装備しています。

EtherCAT は、Ethernet システムをベースとし、より高速で高効率な通信を実現する高性能な産業用ネットワークシステムです。各ノードがEthernet フレームを高速で転送することにより、短い定周期な通信サイクルタイムを実現することができます。

制御用ネットワーク EtherCAT を標準採用したことで、マシン制御に必要な I/O システム、サーボドライバ、インバータ、視覚装置などの機器を1つのネットワークで接続できます。

#### EtherCAT スレーブターミナルをサポート

EtherCAT スレーブターミナルを使用することで省スペース化ができます。また、豊富な種類のNXユニットによりフレキシブルなシステム構築ができます。

#### EtherCAT 上にセーフティサブシステムを実現

NX シリーズ セーフティコントロールユニットを使用することによって、シーケンスコントロール、モーションコントロールのシステムに、セーフティコントロールをEtherCAT 上のサブシステムとして統合することができます。

#### CJ シリーズユニットをサポート

NJ シリーズCPU ユニットでは、EtherCAT ネットワークの各種スレーブ以外にも、I/O バスシステム上に、各種CJ シリーズ用ユニット(基本I/O ユニット、高機能ユニット)を装着して使用できます。

#### EtherNet/IP 通信機能ポートを標準装備

CPU ユニットに、EtherNet/IP 通信機能ポートを標準装備しています。

EtherNet/IP は、Ethernet を使用した産業用のマルチベンダネットワークです。コントローラ間のネットワークやフィールドネットワークとして使用することができます。また、標準のEthernet 技術が使用されているため、さまざまな汎用Ethernet 機器とも混在させて使用することができます。

#### USB ポートを標準装備

パソコンにインストールしたサポートソフトからUSB にてCPU ユニットに直接接続できます。

#### SD メモリカードスロットを標準装備

ユーザプログラムから、CPUユニットに装着したSDメモリカードにアクセスすることができます。

#### 高いハードウェア信頼性

PLC 並みのハードウェア信頼性とRAS 機能を用意しています。

#### マルチコアプロセッサによる並列実行

CPU ユニット形NX701-□□□□は、マルチコアプロセッサを搭載しているため、複数のタスクを並列に実行できます。これにより、大規模な装置も高速に制御できます。

### ソフトウェア上の特長

#### シーケンス制御とモーション制御を統合

1台のCPUユニットでシーケンス制御とモーション制御の両方をカバーします。これにより、シーケンス制御と多軸同期制御を同時に実現します。同じ制御周期の中で、シーケンス制御・モーション制御とI/Oリフレッシュが実行されます。また、制御周期はEtherCAT のプロセスデータ通信周期と一致します。これにより、定周期でゆらぎの少ない高精度な、シーケンス制御およびモーション制御を実現できます。

#### マルチタスク対応

複数のタスクにI/Oリフレッシュやユーザプログラム実行などの一連の処理を割り付け、それぞれに実行条件と実行順序を指定し、それらを組み合わせることでアプリケーションに合わせて柔軟な制御が構築できます。

#### 国際標準規格IEC 61131-3準拠のプログラミング言語仕様

IEC 61131-3に準拠した言語仕様を搭載しています。一部、オムロン独自の改良を加えています。PLCopen® に準拠したモーション制御命令をはじめ、IECルールに則った豊富な命令群(POU)を用意しています。

#### メモリマップを意識しない、変数によるプログラミング

パソコン上で高級言語を使用するときと同様に、すべてのデータは変数によりアクセスします。作成した変数はCPUユニットのメモリ上に自動的に割り付けられ、ユーザがそれを意識する必要はありません。

#### 豊富なセキュリティ機能

操作権限の設定、IDによるユーザプログラムの実行制限など、豊富なセキュリティ機能を用意しています。

#### コントローラ全体の監視機能

CPUユニットは、装着されたユニット、EtherCATスレーブを含めたコントローラ全体の事象(イベント)を監視します。異常時の対処方法は、Sysmac StudioまたはNAシリーズ表示器でガイドされます。発生した事象(イベント)は履歴として記憶されます。

#### オートメーションソフトウェア Sysmac Studio

Sysmac Studioは、コントローラだけでなく周辺機器やEtherCAT上の機器までを、1つのソフトウェアでカバーした統合開発環境です。機器の違いにかかわらず、統一した操作性を提供します。設計から、デバッグ、シミュレーション、立ち上げ、運用開始後の変更までの、すべての工程を支援します。

#### 豊富なシミュレーション機能

仮想コントローラの実行機能、デバッグ機能、タスク実行時間の検証機能など、豊富なシミュレーション機能を用意しています。

## システム構成

### NXシリーズのシステム構成の概要

NXシリーズのシステム構成は、以下のとおりです。

#### 基本構成

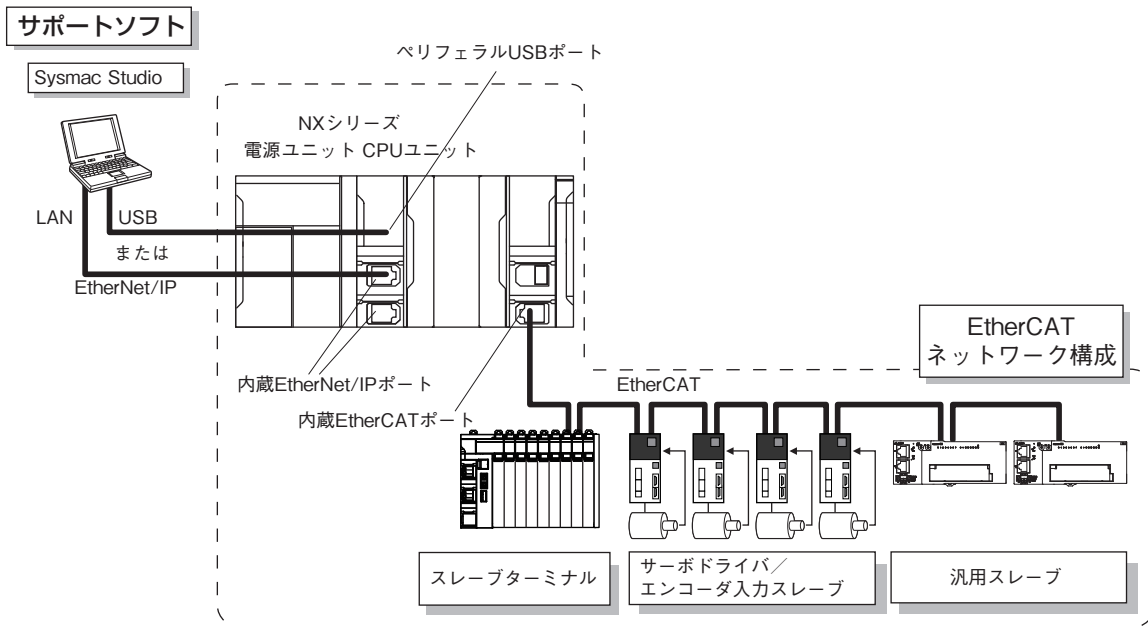
NXシリーズでは、EtherCATネットワーク構成とサポートソフトを基本構成とします。

##### ・EtherCATネットワーク構成

内蔵EtherCATポートを使って、EtherCATスレーブターミナル、デジタルI/O、アナログI/Oなどの汎用スレーブ、およびサーボ/エンコーダ入力スレーブを接続できます。この構成を使用することで、定周期かつゆらぎの少ない高精度な、シーケンス制御およびモーション制御を実現できます。

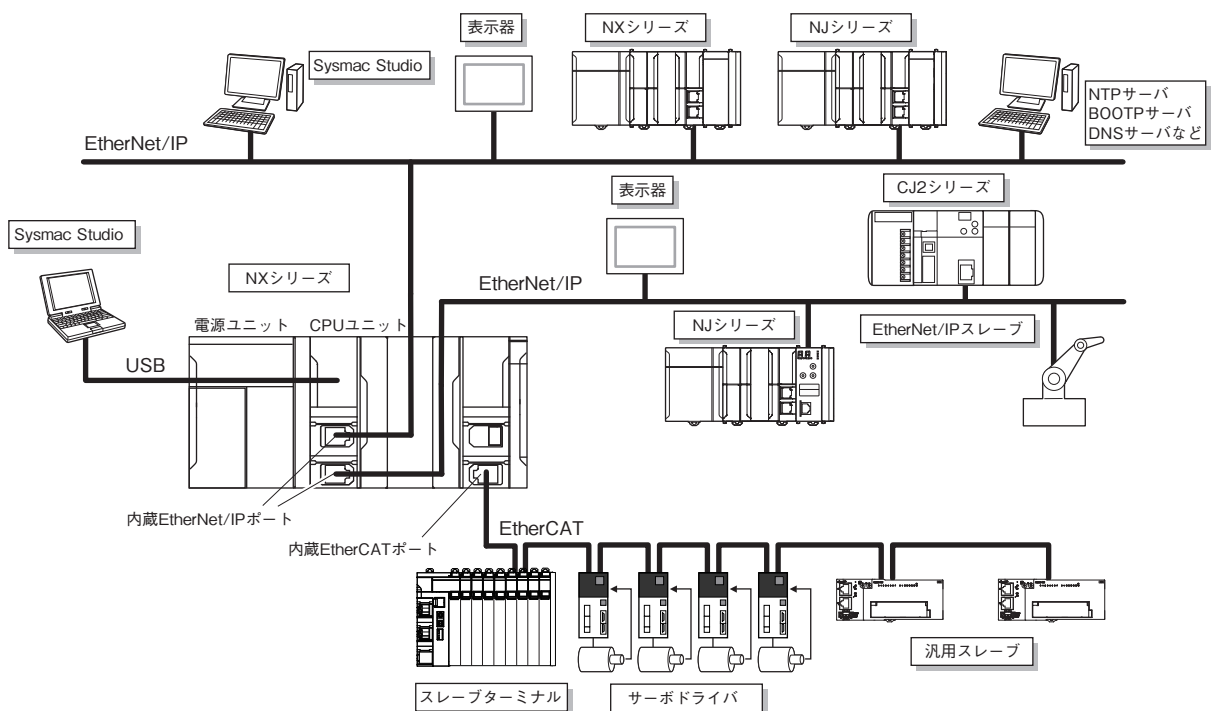
##### ・サポートソフト

CPUユニットのペリフェラル USBポートに市販USBケーブルで接続します。または内蔵EtherNet/IPポートにEthernetケーブルで接続します。



### NXシリーズシステム構成例

NXシリーズシステムは、以下の通信システムを構築できます。



## NJシリーズのシステム構成の概要

NJシリーズのシステム構成は、以下のとおりです。

### 基本構成

NJシリーズでは、EtherCATネットワーク構成、CJユニット構成およびサポートソフトを基本構成とします。

#### ・EtherCATネットワーク構成

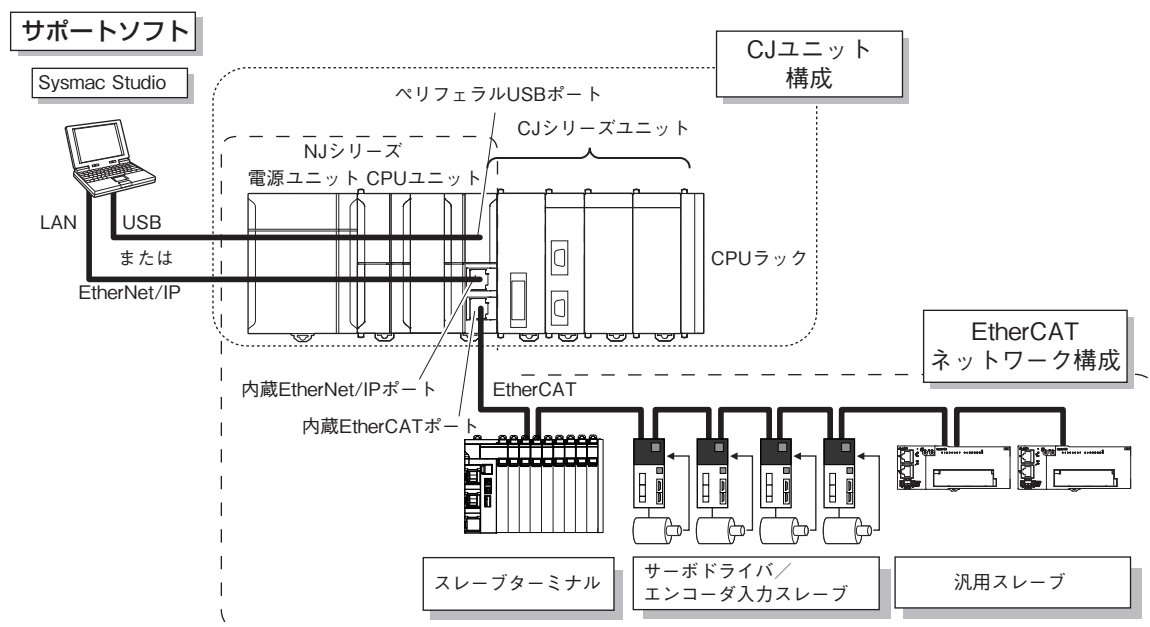
内蔵EtherCATポートを使って、EtherCATスレーブターミナル、デジタルI/O、アナログI/Oなどの汎用スレーブ、およびサーボ/エンコーダ入力スレーブを接続できます。この構成を使用することで、定周期かつゆらぎの少ない高精度な、シーケンス制御およびモーション制御を実現できます。

#### ・CJユニット構成

EtherCATネットワーク以外にもCJシリーズユニット(基本I/Oユニット、高性能ユニット)を装着できます。CPUユニットが装着されているCPUラック以外にも、増設ラックを使ってCJシリーズユニットを増設できます。

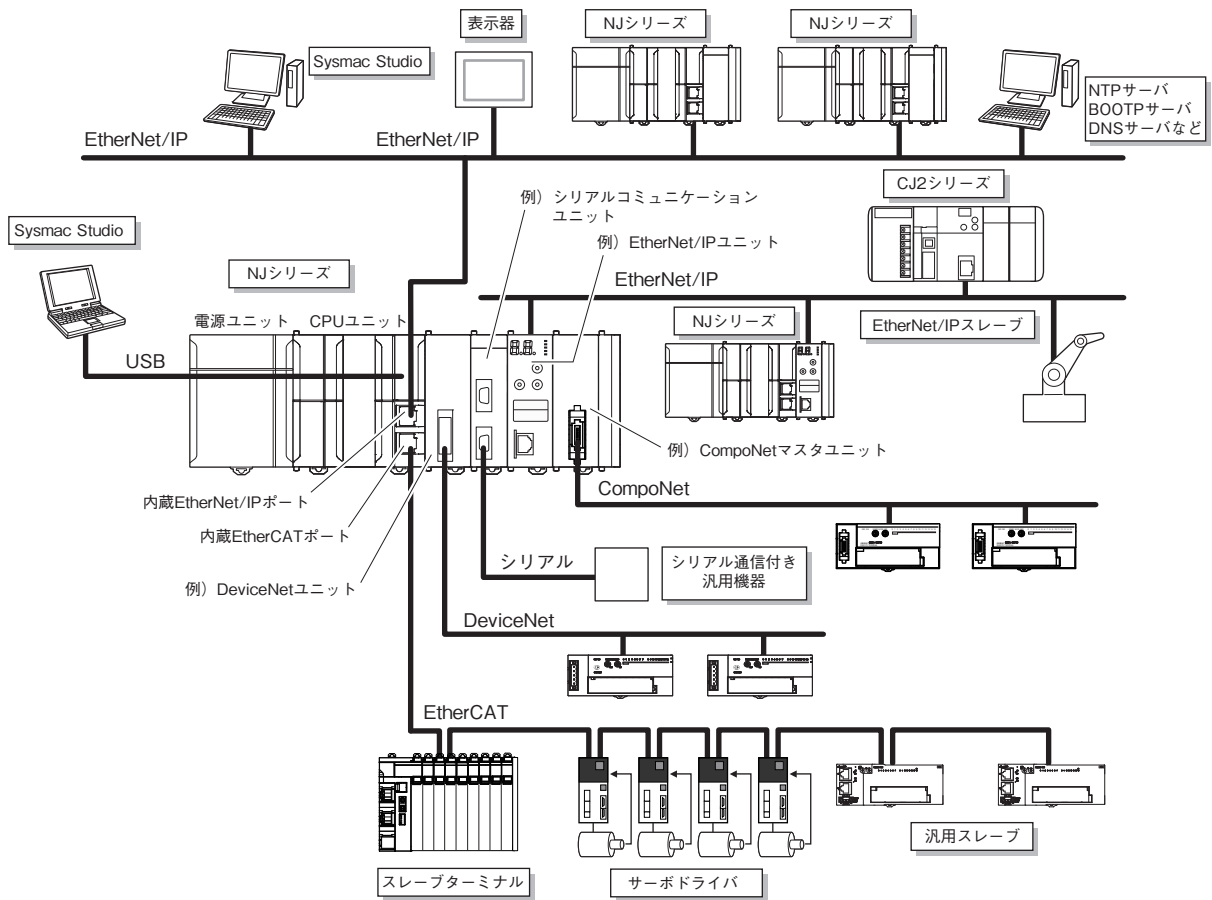
#### ・サポートソフト

CPUユニットのペリフェラル USBポートに市販USBケーブルで接続します。または内蔵EtherNet/IPポートにEthernetケーブルで接続します。



### NJシリーズシステム構成例

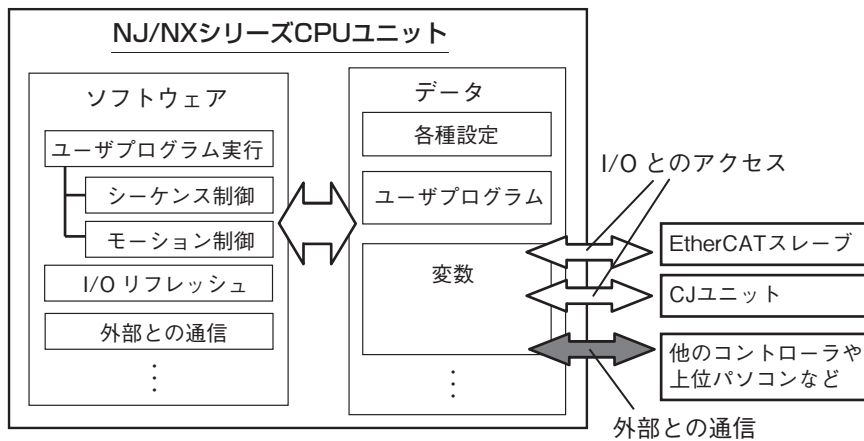
NJシリーズシステムは、以下の通信システムを構築できます。



### CPUユニットの動作

NJ/NXシリーズCPUユニットは、シーケンス制御およびモーション制御のためのユーザプログラム実行、I/Oリフレッシュ、外部との通信などの処理を行います。これらの処理は、CPUユニットのソフトウェアが行います。

また、CPUユニットには、各種設定、ユーザプログラム、変数などのデータがあります。CPUユニットはこれらのデータを使って処理を行います。CPUユニットとI/Oとのアクセス、および外部との通信には、これらのデータのうち、変数を使用します。CPUユニットは、ソフトウェアおよび、変数を使ったI/Oとのアクセスにより、シーケンス制御とモーション制御の両方を実行することができます。



注. CJユニットを使用できるのは、NJシリーズCPUユニットだけです。

## プログラムの構成単位(POU)

POU(Program Organization Unit)とは、IEC 61131-3におけるユーザプログラムの実行単位です。

複数のPOUを組み合わせることで、ユーザプログラム全体を構成します。

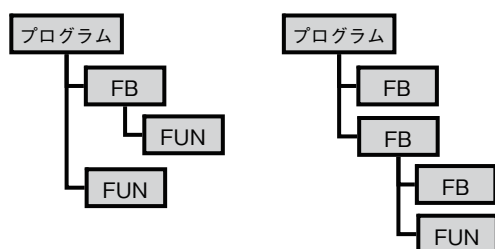
POUの構成要素には、以下の3種類があります。

POUの構成要素	説明
プログラム	メインルーチンに相当するアルゴリズムを記載する主となる要素です。アルゴリズムに、すべての命令、ファンクション、ファンクションブロックを記述することができます。
ファンクションブロック (以降「FB」と省略します)	入力の値が同じでも、異なる値の出力ができる要素です。プログラムまたは別のFB から呼び出されることで起動します。プログラム上でFBを使用するには、「インスタンス」としてプログラムに配置する必要があります。また、内部変数の値を保持することができます。そのため、タイマやカウンタのように状態を持つことができます。
ファンクション (以降「FUN」と省略します)	入力の値が同じであれば、常に均一の値を出力できる要素です。プログラム、FB、またはFUN から呼び出されることで起動します。

### 視認性のよいプログラミング

あるPOUから他のPOUを呼び出すことにより、プログラムを階層的に構成することができます。たとえば、制御単位で階層化することで、プログラムの視認性を高めることができます。

下図は階層化した場合の例です。



### 再利用性の高いプログラミング

FBおよびFUNは、プログラムを部品化するためのしくみです。ある処理をFBとして部品化しておくことで、同様な処理が必要な別の装置でも、FBのインスタンスを呼び出すことで再利用できます。

## 変数とI/Oアクセス

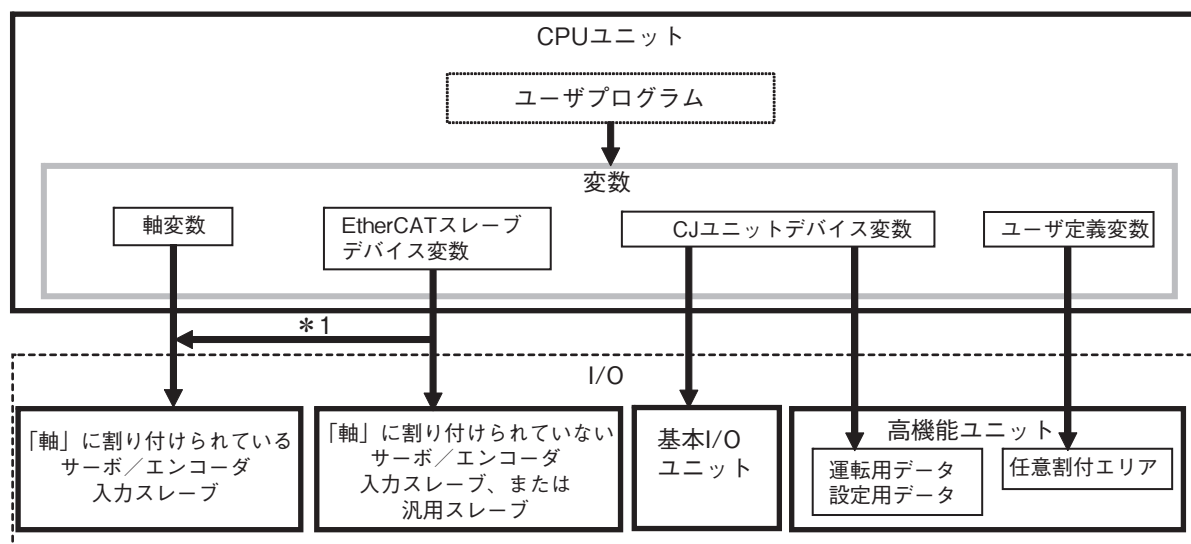
CPUユニットは、ユーザプログラムで変数を使用します。変数は、割り付けられたI/Oのデータとアクセスします。CPUユニットでは、以下のようにI/Oと変数を割り付けます。CJユニットの高機能ユニットは、アクセスするデータによって使用する変数が異なります。

	I/O	データ	変数
EtherCATスレーブ	「軸」に割り付けられていないEtherCATスレーブ*1	—	EtherCATスレーブデバイス変数
	「軸」に割り付けられているEtherCATスレーブ	—	軸変数
CJ ユニット*2	基本I/Oユニット	—	CJユニットデバイス変数
	高機能ユニット	・運転用データ ・設定用データ	CJユニットデバイス変数
		割付任意エリアのデータ*3	ユーザ定義変数

\*1. Sysmac Studio Ver.1.09以降では、「軸」に割り付けられたEtherCATスレーブも、EtherCATスレーブデバイス変数のアクセス対象となります。

\*2. CJユニットを使用できるのは、NJシリーズCPUユニットだけです。

\*3. 拡張機能やスレーブの入出力などを、ユーザがメモリエリアに割り付けて使用するデータです。割付任意エリアのデータは、デバイス変数でアクセスできません。





## 用語解説

### ●AT(割付先)指定

変数の属性のひとつ

変数の割付先をユーザが指定する属性

I/OポートまたはCJユニット用メモリのアドレスを指定できる

### ●FB

ファンクションブロックの略

### ●FUN

ファンクションの略

### ●I/Oポート

CPUユニットが外部のデバイス(スレーブ/ユニット)と、データをやりとりするための論理的なインターフェース

### ●I/Oマップ設定

I/Oポートに対して、変数を割り付けた設定

I/Oポートと変数間の割付情報

### ●PDO通信

プロセスデータ通信の略

プロセスデータ通信周期(=プライマリ定周期タスクのタスク周期)で、サイクリックに、マスタスレーブ間でデータ交換をする通信

### ●POU(Program Organization Unit)

IEC 61131-3におけるプログラムの実行モデルの単位  
ユーザプログラム全体を構成する基本的な単位として、アルゴリズムとローカル変数テーブルを記述するプログラム、ファンクションブロック、ファンクションの3種類がある

### ●SDO通信

EtherCAT通信の一種で、任意のタイミングで情報伝達を行うサービスデータオブジェクト(Service Data Objects: SDO)を使用した通信

### ●基本データ型

IEC 61131-3で定義されているデータ型のこと  
ブール型、ビット列型、整数型、実数型、持続時間型、日付型、時刻型、日付時刻型、文字列型がある  
ユーザが定義するデータ型である「派生データ型」の対義語である

### ●共用体型

派生データ型の一つで、同じデータを異なるデータ型として扱えるようにしたデータ型のこと

### ●グローバル変数

すべてのPOU(プログラム、ファンクションブロック、ファンクション)から読み書き可能な変数のこと

### ●構造体型

派生データ型の一つで、複数の異なるデータ型を階層的にひとつにまとめたデータ型のこと

### ●コンスタント

変数の属性のひとつ

コンスタント指定を行った変数は、命令、ST言語の演算子、またはCIPメッセージ通信によって値を書き込むことができなくなる

### ●軸

モーション制御機能モジュール内の機能単位のこと  
外部のサーボドライバの駆動機構、エンコーダ入力スレーブの検出機構などを割り付ける

### ●軸グループ

モーション制御機能モジュール内の、複数の「軸」をまとめた機能単位のこと

### ●軸グループ変数

軸グループごとの各種ステータス情報や一部の軸グループパラメータ設定情報などを構造体で定義したシステム定義変数

モーション制御命令の軸グループの指定、および軸グループの指令補間速度や異常情報などのモニタに使用される

### ●軸変数

軸ごとの各種ステータス情報や一部の軸パラメータ設定情報などを構造体で定義したシステム定義変数

モーション制御命令の軸の指定、および軸の指令現在位置や異常情報などのモニタに使用される

### ●システム定義変数

すべての属性がシステムによって定義されており、ユーザはそれを一切変更できない変数のこと

### ●初期値

変数の属性のひとつ

以下のときに設定される変数の値である

- ・電源投入時
- ・運転モードへの切替時
- ・ユーザプログラム転送時に初期化する指定のとき
- ・全停止フォールトレベルのコントローラ異常発生時

### ●タスク

「プログラム実行時の属性」のこと

### ●タスク周期

プライマリ定周期タスクおよび定周期タスクの実行間隔

### ●定周期タスク

周期ごとに、I/Oリフレッシュとユーザプログラム実行をするタスク

### ●デバイス変数

I/Oポートを介して特定のデバイスにアクセスするための変数

### ●名前空間

ファンクション、ファンクションブロック定義、データ型の名称を、分割・階層化して管理する仕組み

### ●ネットワーク公開

変数の属性のひとつ

CIPメッセージ通信またはタグデータリンク機能によって、その変数を、他のコントローラや上位パソコンなどからも読み書きできるようにする

### ●配列指定

変数の指定のひとつ

同じデータ型の要素を複数個まとめて、ひとつの変数にしたもの  
ひとつひとつの要素は、先頭からの通し番号(添え字)で指定する



### ●派生データ型

仕様をユーザが定義するデータ型のこと  
構造体型、共用体型、列挙型がある

### ●ファンクション

演算処理など、入力に対して出力が一意に決まる部品を作成するとき使用するPOU

### ●ファンクションブロック

タイマやカウンタなど、状態により、同じ入力に対して出力が異なる場合がある部品を作成するとき使用するPOU

### ●プライマリ定周期タスク

最も優先度が高いタスク

### ●プログラム

ファンクション、ファンクションブロックと並ぶ、3種類のPOUのひとつ  
タスクに割り付けられて動作する

### ●プロセスデータ通信

EtherCAT通信の一種で、定周期でリアルタイムの情報交換を行うプロセスデータオブジェクト(Process Data Objects: PDO)を使用した通信  
「PDO通信」とも呼ぶ

### ●変数

ユーザプログラム中で使用する、任意の数値や文字列を表現する概念のこと  
任意の値を代入することによって、変数の値を変えることができる。常に値が変わらない「定数」の対語である

### ●変数メモリ

AT(割付先)指定がない変数の現在値  
「AT(割付先)指定なし」属性の変数でのみアクセスする

### ●保持

変数の属性のひとつ  
以下のときに、本属性を設定した場合は変数の値を保持し、設定しない場合は初期値になる  
・電断後の電源投入時  
・運転モードへの切替時  
・ユーザプログラム転送時で初期化しない指定のとき

### ●モーション制御命令

モーション制御機能を実行するためのファンクションブロック定義の命令  
モーション制御機能モジュールでは、PLCopen®のモーション制御用ファンクションブロックに準拠した命令とモーション制御機能モジュール独自の命令を持つ

### ●ユーザ定義変数

すべての変数属性がユーザによって定義されており、ユーザによる変更が可能な変数

### ●リテラル

ユーザプログラム内に使用される定数表現のこと

### ●列挙型

派生データ型の一つ  
あらかじめ用意された名前リスト(列挙子)内のひとつを値とするデータ型

### ●列挙子

列挙型の変数が取り得る複数の値を、それぞれ文字で表現したものの  
列挙型の変数の値は、列挙子の中のどれかひとつをとる

### ●ローカル変数

定義されたPOU内だけからアクセス可能な変数  
複数のPOUからアクセス可能な「グローバル変数」の対語である  
ローカル変数には、内部変数、入力変数、出力変数、入出力変数、外部変数が含まれる