

三菱電機

Mitsubishi EZSocket CNC プロバイダ

Version 1.4.7

ユーザーズ ガイド

March 28, 2023

備考：

CNC のファームウェアが FL 版以降の物になっていることを確認して接続してください。

本プロバイダを使用するためには、三菱 CNC 通信用ソフトウェアランタイムライブラリ (FGSB1224W100) が必要です。本プロバイダはライブラリバージョン A5 を使用して開発されました。

【改版履歴】

バージョン	日付	内容
1.2.2	2018-08-30	初版.
1.3.0	2018-08-31	PLCRANGE 変数を追加しました.
	2018-09-05	文言を修正しました.
1.3.1	2018-10-12	PLCのデバイス種別の要素にアクセスする際のアドレス指定方法の不具合及び、カウンタコイルと 10ms 単位タイマコイルのデバイス名指定方法の不具合を修正しました.
1.3.2	2018-10-26	複数台コントローラの接続時の不具合を修正しました.
1.3.3	2019-06-24	PLCELEM 変数の値取得時の不具合を修正しました.
1.4.0	2019-12-05	内部ロジックを変更しました.
1.4.1	2020-01-15	不具合を修正しました.
1.4.2	2020-02-28	CNC との接続方法を変更しました. エラーコードを追加しました.
1.4.4	2020-06-30	CNC との接続方法を Ver1.4.1 の方法に戻しました. プロバイダのスレッドモデルを STA に変更しました.
	2020-08-03	ユーザーズガイドを修正しました.
	2021-03-09	エラーコードの重複を修正
1.4.5	2021-05-14	通信ライブラリがない状態で CaoWorkspace::AddController メソッドを呼び出した際の不具合を修正しました.
1.4.6	2021-08-31	ログ出力処理の最適化. 内部コード最適化.
	2022-07-07	ユーザーズガイドを修正しました.
1.4.7	2022-08-08	内部処理の修正
	2023-03-28	備考を追記しました.

【動作確認機器】

機種	バージョン	注意事項
M700 シリーズ旋盤		

目次

1. はじめに.....	5
1.1. 本書が想定している環境とバージョン.....	6
1.2. 参考となる情報源.....	6
2. アプリケーション開発のための環境セットアップ.....	7
2.1. CNC とクライアント PC との接続.....	7
2.2. PC 開発環境のセットアップ.....	8
2.2.1. 実行ランタイムの準備.....	8
2.2.2. CNC プロバイダの手動インストール.....	8
3. CNC プロバイダによるプログラミング.....	9
3.1. PLC にアクセスするサンプルプログラミング.....	9
3.1.1. サンプルプログラム.....	10
3.1.1.1. 接続.....	11
3.1.1.2. PLC アドレスのリード・ライト.....	12
3.1.1.3. 切断.....	13
4. コマンドリファレンス.....	14
4.1. メソッド/プロパティ一覧.....	14
4.2. メソッド・プロパティ.....	14
4.2.1. CaoWorkspace クラス.....	14
4.2.1.1. AddController メソッド.....	14
4.2.2. CaoController クラス.....	16
4.2.2.1. Index プロパティ.....	16
4.2.2.2. Name プロパティ.....	16
4.2.2.3. VariableNames プロパティ.....	16
4.2.2.4. Variables プロパティ.....	17
4.2.2.5. AddVariable メソッド.....	17
4.2.3. CaoVariable クラス.....	18
4.2.3.1. Index プロパティ.....	18
4.2.3.2. Name プロパティ.....	19
4.2.3.3. Value プロパティ.....	19

4.3. 変数一覧.....	19
4.3.1. CaoController クラス変数.....	20
4.3.1.1. @MAKER_NAME	20
4.3.1.2. @VERSION.....	21
4.3.1.3. PLCELEM<??>.....	21
4.3.1.4. PLCRANGE<??>.....	23
4.3.1.5. COMMON<??>.....	24
4.3.1.6. TOOL_OFFSET<??>.....	25
4.3.1.7. PRG_NAME<??>.....	27
5. CNC プロバイダエラーコード	29

1. はじめに

本書は、三菱電機社の CNC の各種情報にアクセスするプロバイダのユーザーズガイドです。図 1-1 が本プロバイダとデバイスの全体構成図になります。本プロバイダは、三菱社製の CNC 通信用 S/W ランタイムライブラリを用いて TCP/IP 通信で三菱 CNC と接続します。またマルチスレッドに対応させる場合はマルチスレッド対応のランタイムを PC にインストールする必要があります。以降本プロバイダを CNC プロバイダと呼称します。

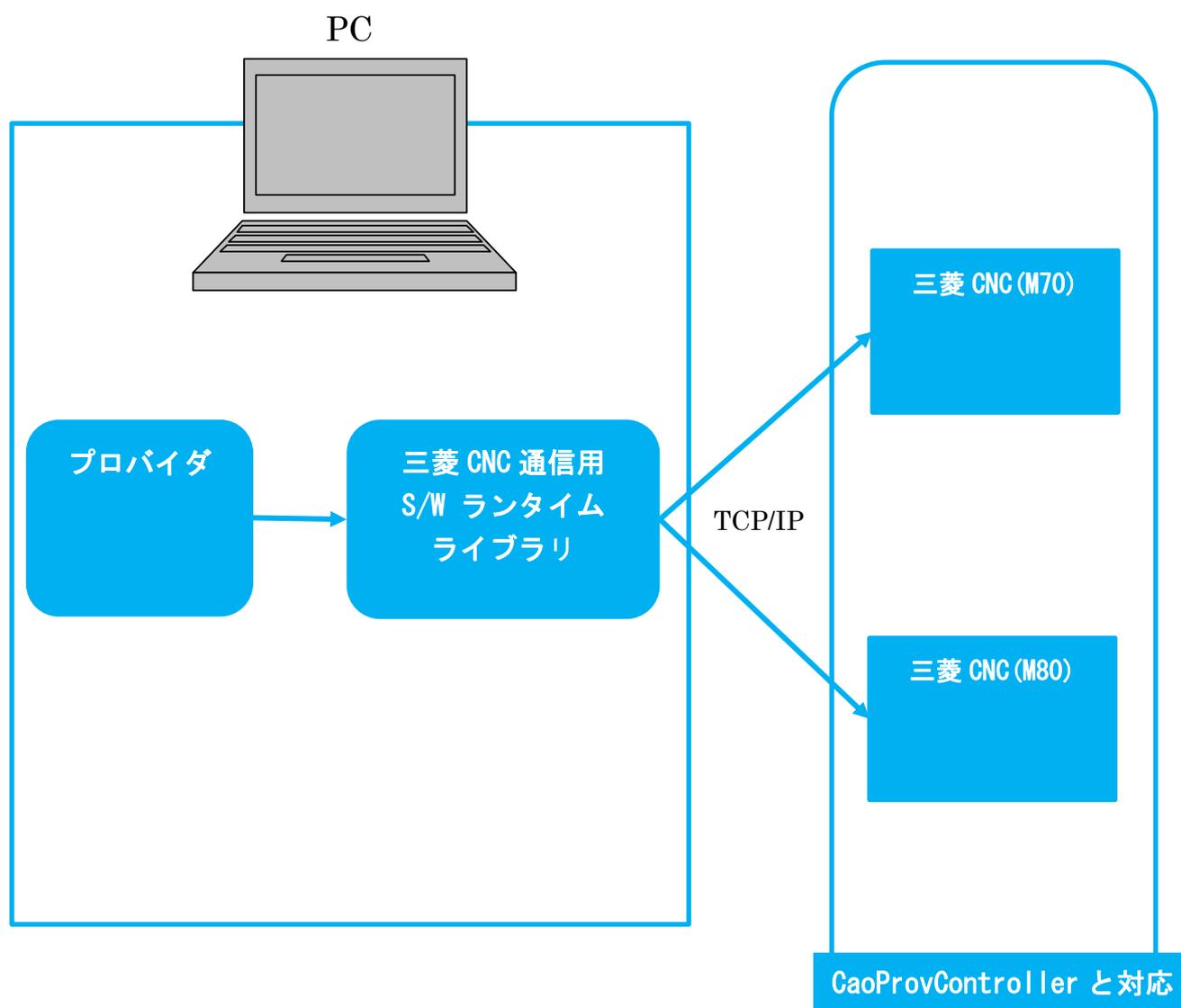


図 1-1 構成図

また、CNCプロバイダ及びデバイスそれぞれの対応を図 1-2に表します。
(※一例です。全てを表しているわけではありません。)

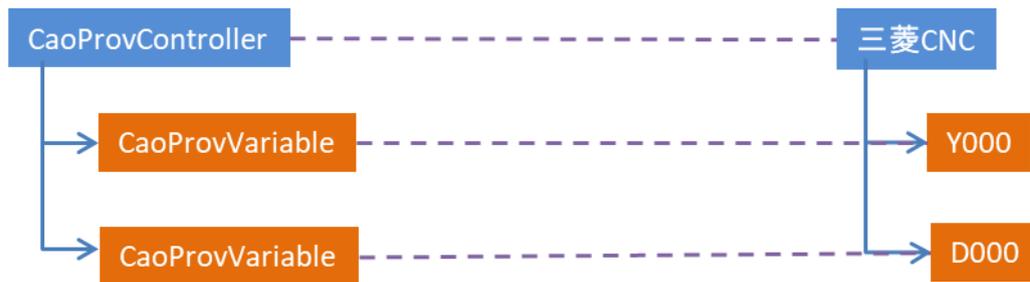


図 1-2 プロバイダの構成とデバイス情報との対応図

1.1. 本書が想定している環境とバージョン

クライアント PC が Windows 上で動作し、対象とする CNC がイーサネット接続可能な会社名製デバイス種類である環境を想定しています。PC の開発環境は、Component Object Model (COM, コンポーネント・オブジェクト・モデル) をサポートするプログラミング環境であれば開発が可能です。

1.2. 参考となる情報源

CNC プロバイダは下記のマニュアルを参照して作成されています。本書で記述する情報は開発時に参照した情報のため、最新情報ではない可能性があります。最新情報については、三菱電機株式会社が公開している最新のリファレンスマニュアルを参照してください。

- 三菱 CNC 用ソフトウェア FCSB1224W000 リファレンスマニュアル IB-1501208-D

本書のプログラミング事例は、すべて Visual Basic 6.0 で記載していますが、C++, Java, .NET などさまざまなプログラム言語で開発が可能です。使用方法に関しては、「ORiN2 プログラミングガイド」を参照してください。

「ORiN 2 プログラミングガイド」は ORiN2 SDK インストールフォルダの以下のファイルに該当します。

- ORiN2¥CAO¥Doc¥ORiN2_ProgrammersGuide_<lang>.pdf

※<lang>の部分は環境毎の言語文字列に置き換えてお読みください。

プロバイダを使ったアプリケーションを開発する上で必要となる ORiN2, COM/DCOM の基礎知識や技術に関して例を交えながら解説されています。

2. アプリケーション開発のための環境セットアップ

2.1. CNC とクライアント PC との接続

CNC プロバイダは三菱 EZSocketCNC API を使用して CNC と接続を行い、三菱 EZSocketCNC API は CNC と接続する際は TCP/IP 通信で接続を行います。クライアント PC と接続している LAN ケーブルを CNC デバイスに差し込みます。図 2.1-1 のように差込口が 2 箇所ありますが左側の赤い丸の差込口に差し込んでください。

※機器の仕様により LAN ポート毎に用途が異なる場合があります。図のように差し込みをしても通信ができない場合、メーカーに問い合わせてください。

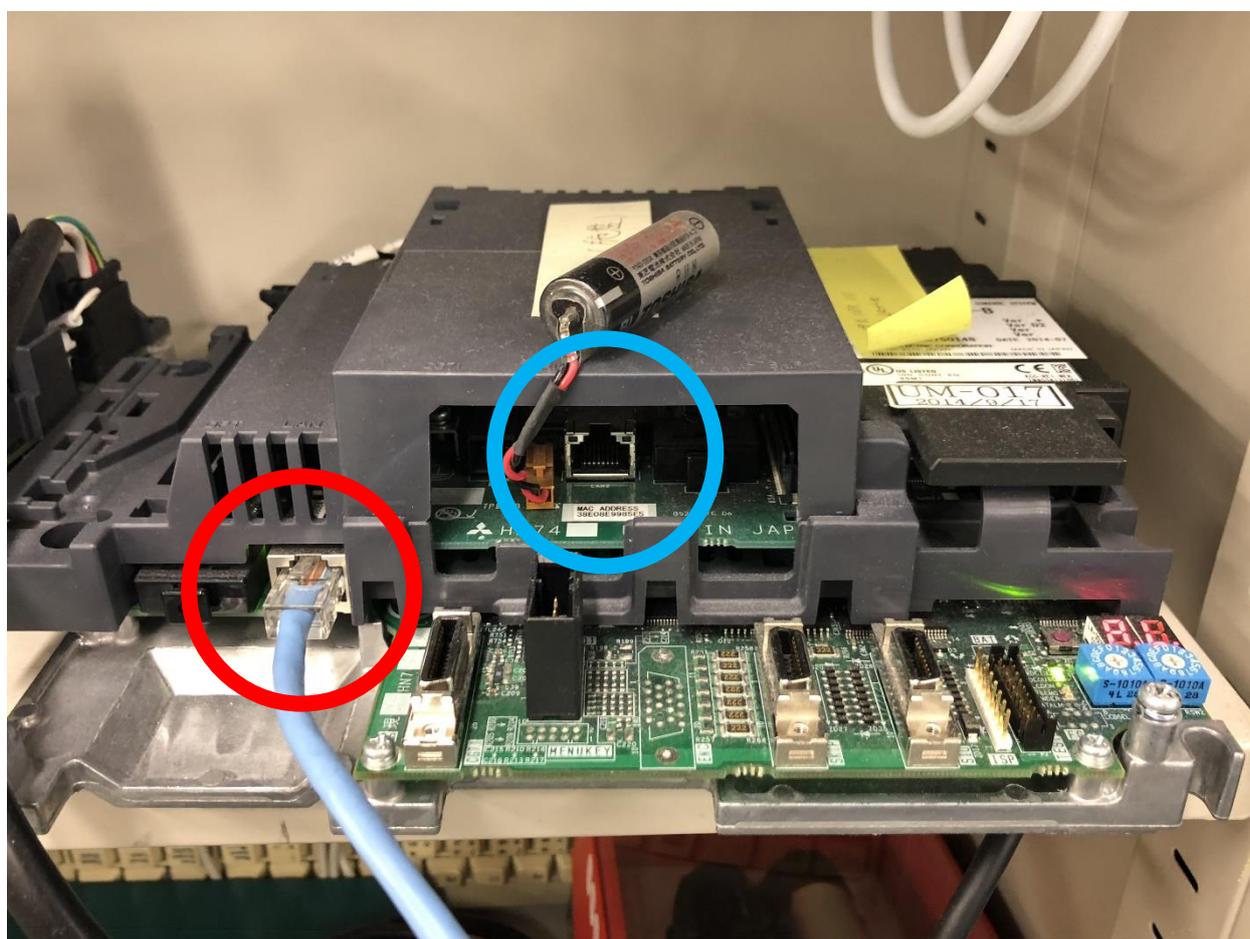


図 2.1-1 実際のデバイス

2.2. PC 開発環境のセットアップ

2.2.1. 実行ランタイムの準備

三菱電機のホームページにある三菱 CNC 通信用 S/W ランタイムライブラリを事前に開発 PC にインストールしてください。マルチスレッドに対応させる場合、マルチスレッド対応版のランタイムライブラリをインストールする必要があります。

2.2.2. CNC プロバイダの手動インストール

CNC プロバイダを使用するためには手作業で下記レジストリ登録を行う必要があります。レジストリ登録を行う場合は、管理者権限でコマンドプロンプトを起動し、regsvr32 コマンドを実行してください。実行する際ファイルのあるパスまで移動するか、ファイルパスを指定して実行してください。

また、CAO エンジンが動作するには予め、PC 毎に正規の ORiN2 SDK ライセンスが 1 つ登録されていなくてはなりません。ORiN2 SDK ユーザーズガイド内にある「ライセンスの追加と削除」の節を参照してください。

表 2-1 CNC プロバイダ

ファイル名	CaoProvMitsubishiEZSocketCNC.dll
ProgID	CaoProv.Mitsubishi.EZSocketCNC
レジストリ登録	regsvr32 CaoProvMitsubishiEZSocketCNC.dll
レジストリ登録の抹消	regsvr32 /u CaoProvMitsubishiEZSocketCNC.dll

3. CNC プロバイダによるプログラミング

CNC プロバイダでは、以下の手順でクライアント PC と CNC を接続することができます。

- CaoEngine の作成
- CaoWorkspace の作成
- CaoController の作成
- CaoController の Connect コマンドを実行

CNC に接続した後は、CaoController に、CaoVariable オブジェクトを生成することで、CNC 内の情報にアクセスすることができます。

3.1. PLC にアクセスするサンプルプログラミング

ここでは例として PLC のレジスタ領域 R の値を読み書きするサンプルプログラムを示します。表 3-1 にサンプルプログラムの要件を、図 3-1 にサンプルプログラムの流れをそれぞれ記述しています。

表 3-1 サンプルプログラムの要件

要件	説明
接続先	TCP/IP で接続する
	接続先 IP アドレスは 10.195.133.20
	接続先ポート番号は 683
処理内容	PLC の R0100 から値を読み込む。
	PLC の R1100 に取得した値 + 1 の値を書込む

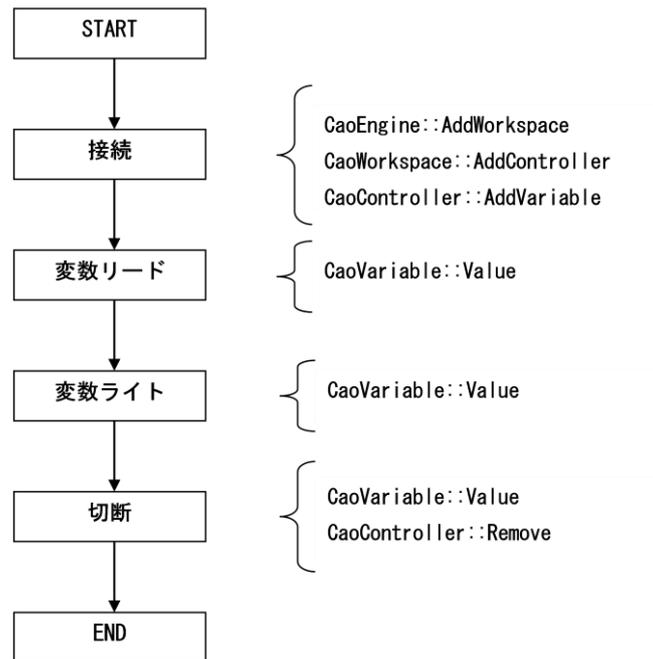


図 3-1 PLC アクセスの流れ

以降の節から具体的なコードを示します。

3.1.1. サンプルプログラム

以下にサンプルプログラムの全体像を示します。

Sample	READ_WRITE_PLG.vb
--------	-------------------

Sub Main

' オブジェクト

```
Dim engine As caoEngine
Dim workspace As caoWorkspace
Dim ctrl As CaoController
Dim varR0100 As CaoVariable
Dim varR1100 As CaoVariable
```

' CaoEngine オブジェクトの生成

```
Set engine = New caoEngine
```

' CaoWorkspace オブジェクトの生成

```
Set workspace = engine.AddWorkspace("NewWrks", "")
```

' CaoController オブジェクトの生成

```
Set ctrl = workspace.AddController("EZSocketCNC", _
    "CaoProv.MITSUBISHI.EZSocketCNC", _
    "", _
    "CONN=ETH:10.195.133.20, TIMEOUT=50")
```

' R0100 にアクセスする CaoVariable を CaoController に追加

```
Set varR0100 = ctrl.AddVariable("PLCELEM_R0100", "DeviceType = R, Number = 100")
```

```

' R1100 にアクセスする CalVariable を CaoController に追加
Set varR1100 = ctrl.AddVariable("PLCELEM_R1100", "DeviceType = R, Number = 1100")

' R0100 から値を読み込む
Dim value As Long
value = varR0100.value

' R0100 から読込んだ値 + 1 を R1100 に書込む
varR1100.value = value + 1

' CaoWorkspace から CaoController を削除
Call workspace.Controllers.Remove(ctrl.Index)
' CaoController の消去
Set ctrl = Nothing

' CaoEngine から CaoWorkspace を削除
Call engine.Workspaces.Remove(workspace.Index)
' CaoWorkspace の消去
Set workspace = Nothing

' CaoEngine の消去
Set engine = Nothing
End Sub

```

3.1.1.1. 接続

CNC と接続するためには、以下の手順を行います。

- (1) オブジェクトを保持するための変数を用意します。コントローラ接続に必要なオブジェクトは、CaoEngineオブジェクトとCaoWorkspaceオブジェクトとCaoControllerオブジェクトです。CaoWorkspaceオブジェクトは、CaoControllerオブジェクトをCaoWorkspacesから取得する場合には変数を用意する必要はありません。また変数にアクセスするためのCaoVariableオブジェクトも必要になります。以下にVB6でのコード例を示します。

```

Dim engine As caoEngine      ' CaoEngineオブジェクト用の変数
Dim workspace As caoWorkspace ' CaoWorkspaceオブジェクト用の変数
Dim ctrl As CaoController   ' CaoControllerオブジェクト用の変数
Dim varR0100 As CaoVariable  ' CaoVariable オブジェクト用の変数
Dim varR1100 As CaoVariable  ' CaoVariable オブジェクト用の変数

```

- (2) CaoEngineオブジェクトを生成します。CaoEngineオブジェクトはNewキーワードを使って生成します。

```

' CaoEngine オブジェクトの生成
Set engine = New caoEngine

```

- (3) CaoWorkspaceオブジェクトを取得もしくは生成します。CaoEngineオブジェクトを生成すると、

デフォルトでCaoWorkspacesオブジェクトとCaoWorkspaceオブジェクトを1つずつ生成しています。以下にCaoWorkspaceオブジェクトを新しく生成するコード例とデフォルトのCaoWorkspaceを示します。

’ CaoWorkspace オブジェクトの生成

```
Set workspace = engine.AddWorkspace("NewWrks", "")
```

- (4) CaoControllerオブジェクトを生成し、CNCと接続します。CaoControllerオブジェクトを生成するには、使用するプロバイダ名と使用するためのパラメータを設定します。CNCプロバイダでは、接続先のコントローラのIPアドレスをオプションで指定します。以下にコード例を示します。

’ CaoController オブジェクトの生成

```
Set ctrl = workspace.AddController("EZSocketCNC", _
    "CaoProv.MITSUBISHI.EZSocketCNC", _
    "", _
    "CONN=ETH:10.195.133.20, TIMEOUT=50")
```

- (5) CaoVariableオブジェクトを生成します。接続したい変数のCaoVariableオブジェクトを生成します。以下にPLC(R0100)とPLC(R1100)にアクセスする変数オブジェクトを生成するコード例を示します。

```
Set varR0100 = ctrl.AddVariable("PLCELEM_R0100", "DeviceType = R, Number = 100")
Set varR1100 = ctrl.AddVariable("PLCELEM_R1100", "DeviceType = R, Number = 1100")
```

3.1.1.2. PLC アドレスのリード・ライト

接続した PLC アドレスの値を取得/設定するには、CaoVariable オブジェクトの Value プロパティを参照/設定します。PLC アドレスの値を取得/設定する場合は、32 ビット符号あり整数を用意する必要があります。以下にコード例を示します。

```
Dim value As Long
value = varR0100.value           ’ 値の取得
varR1100.value = value + 1      ’ 値の設定
```

3.1.1.3. 切断

コントローラと切断する場合は、Disconnect コマンドを実行した後に、生成したオブジェクトを消去すると共に、オブジェクトを管理するコレクションクラスから消去するオブジェクトを削除します。以下にコード例を示します。

' CaoController から CaoVariable をクリアする

Call ctrl.Variables.Clear

' CaoVariable の消去

Set varR0100 = Nothing

Set varR1100 = Nothing

' CaoWorkspace から CaoController を削除

Call workspace.Controllers.Remove(ctrl.Index)

' CaoController の消去

Set ctrl = Nothing

' CaoEngine から CaoWorkspace を削除

Call engine.Workspaces.Remove(workspace.Index)

' CaoWorkspace の消去

Set workspace = Nothing

' CaoEngine の消去

Set engine = Nothing

4. コマンドリファレンス

4.1. メソッド/プロパティ一覧

表 4-1 メソッド/プロパティ一覧

カテゴリ	メソッド/プロパティ ¹	機能	参照
CaoWorkspace			
	Addcontroller	M コントローラに接続	P. 14
CaoController			
	Index	P コントローラ番号の取得	P. 16
	Name	P コントローラ名の取得	P. 16
	VariableNames	P 接続可能な変数名リストの取得	P. 16
	Variables	P コントローラが保持する変数コレクションの取得	P. 17
CaoVariable			
	Index	P タスク番号の取得	P. 18
	Name	P タスク名の取得	P. 19
	Value	P 値の取得/設定	P. 19

4.2. メソッド・プロパティ

4.2.1. CaoWorkspace クラス

4.2.1.1. AddController メソッド

CaoWorkspace に、コントローラオブジェクトを追加します。CNC プロバイダは AddController メソッド実行時に渡されたパラメータを参照し CNC と接続を行います。

書式

CaoController AddController

```
(
    "<コントローラ名>", // コントローラ名(任意)
    "CaoProv. MITSUBISHI. EZSocketCNC", // プロバイダ名(固定)
    "<マシン名>", // プロバイダ実行マシン名(未使用)
    "<オプション>" // オプション文字列
)
```

使用例

¹ M:メソッド, P:プロパティ, E:イベントをそれぞれ示します。

```

Dim caoEng As CaoEngine      ' Engineオブジェクト
Dim caoWs As CaoWorkspace    ' WorkSpaceオブジェクト
Dim caoCtrl As CaoController ' Controlleオブジェクト

Set caoEng = New CaoEngine
Set caoWS = caoEng.Workspaces.Item(0)
Set caoCtrl = caoWS.AddController("EZSocketCNC",
                                   "CaoProv. MITSUBISHI. EZSocketCNC",
                                   "",
                                   "CONN = ETH:10.195.133.20 , TIMEOUT=50")

```

オプション

以下にオプション文字列に指定するオプションを示します。オプション文字列は下記に示す各オプションをカンマ(,)でつなげた文字列となります。

オプション	必須	説明	値範囲	デフォルト値
Conn=	○	接続用オプション (ETH)	-----	4.2.1.1.1 を参照
Series=	--	NC システムタイプ	5:M700 シリーズ旋盤 6:M700 シリーズマシンニング 8:M800 シリーズ旋盤 9:M800 シリーズマシンニング	5
Machine=	--	NC 制御ユニット番号 ※複数接続する場合、コントローラ毎に一意的な番号を付与する必要があります。	1 - 255	1
Timeout=	--	応答待機時間(100ms)	1 - 3000	10
Multithread=	--	マルチスレッドで通信を行うかを指定します。 true を指定すると複数の NC に対して同時通信可能となります。 ランタイムがマルチスレッド対応していない場合に true を指定すると、プロバイダがクラッシュしますので注意してください。	true / false	false

4.2.1.1.1. Conn オプション

Conn = ETH:<接続先 IP. [:<接続先ポート>[:<ローカル IP>[:<ローカルポート>]]]

<接続先 IP> : 接続先 IP アドレスを***.***.***.***の形式で指定します。

[<接続先ポート>] : 接続先ポート番号を指定します。

デフォルト:683

[<ローカル IP>] : PC 側の IP アドレスを***.***.***.***の形式で指定します.

デフォルト:未指定

[<ローカルポート>]: デフォルト:未指定

4.2.2. CaoController クラス

4.2.2.1. Index プロパティ

コントローラ番号を Long 型 (4 バイト整数型) で取得します。この番号は、該当のコントローラが、CaoWorkspace クラスの保持するコントローラコレクションの何番目のデータかを示します。

使用例

```
Dim caoEng As CaoEngine      ' Engineオブジェクト
Dim caoWs As CaoWorkspace    ' Workspaceオブジェクト
Dim caoCtrl As CaoController ' Controlleオブジェクト

Set caoEng = New CaoEngine
Set caoWS = caoEng.Workspaces.Item(0)
Set caoCtrl = caoWS.AddController("EZSocketCNC",
                                   "CaoProv. MITSUBISHI. EZSocketCNC",
                                   "",
                                   "CONN = ETH:10.195.133.20 , TIMEOUT=50")

' Index取得
Dim index as Long
index = caoCtrl.Index
```

4.2.2.2. Name プロパティ

CaoWorkspace クラスの AddController メソッドで指定されたコントローラ名を取得します。

使用例

```
Dim caoEng As CaoEngine      ' Engineオブジェクト
Dim caoWs As CaoWorkspace    ' Workspaceオブジェクト
Dim caoCtrl As CaoController ' Controlleオブジェクト

Set caoEng = New CaoEngine
Set caoWS = caoEng.Workspaces.Item(0)
Set caoCtrl = caoWS.AddController("EZSocketCNC",
                                   "CaoProv. MITSUBISHI. EZSocketCNC",
                                   "",
                                   "CONN = ETH:10.195.133.20 , TIMEOUT=50")

Debug.Print caoCtrl.Name
```

4.2.2.3. VariableNames プロパティ

接続可能な変数名リストを取得します。本プロパティで取得した変数名は、後述する AddVariable

メソッドの第一引数に使用することができます。

使用例

```
Dim caoEng As CaoEngine      ' Engineオブジェクト
Dim caoWs As CaoWorkspace    ' WorkSpaceオブジェクト
Dim caoCtrl As CaoController ' Controlleオブジェクト

Set caoEng = New CaoEngine
Set caoWS = caoEng.Workspaces.Item(0)
Set caoCtrl = caoWS.AddController("EZSocketCNC",
                                   "CaoProv. MITSUBISHI. EZSocketCNC",
                                   "",
                                   "CONN = ETH:10.195.133.20 , TIMEOUT=50")

' ファイル名リスト取得
Dim variables as Variant
variables = caoCtrl.VariableNames
```

4.2.2.4. Variables プロパティ

コントローラが保持する、変数コレクションを取得します。

使用例

```
Dim caoEng As CaoEngine      ' Engineオブジェクト
Dim caoWs As CaoWorkspace    ' WorkSpaceオブジェクト
Dim caoCtrl As CaoController ' Controlleオブジェクト

Set caoEng = New CaoEngine
Set caoWS = caoEng.Workspaces.Item(0)
Set caoCtrl = caoWS.AddController("EZSocketCNC",
                                   "CaoProv. MITSUBISHI. EZSocketCNC",
                                   "",
                                   "CONN = ETH:10.195.133.20 , TIMEOUT=50")

' 変数コレクション取得
Dim variables as CaoVariables
Set variables = caoCtrl.Variables

' 変数取得
Dim variable as CaoVariable
Set variable = Variables.Item(0)
```

4.2.2.5. AddVariable メソッド

CaoController に変数オブジェクトを追加します。変数名には4.3.1に示すもののみ使用できます。以下に、AddVariable の仕様を示します。

書式

CaoVariable AddVariable

```
(
    "<変数名>",           // 変数名
    "<オプション>"       // オプション文字列
)
```

使用例

```
Dim caoEng As CaoEngine      ' Engineオブジェクト
Dim caoWs As CaoWorkspace   ' Workspaceオブジェクト
Dim caoCtrl As CaoController ' Controllerオブジェクト

Set caoEng = New CaoEngine
Set caoWS = caoEng.Workspaces.Item(0)
Set caoCtrl = caoWS.AddController("EZSocketCNC",
                                   "GaoProv. MITSUBISHI. EZSocketCNC",
                                   "",
                                   "CONN = ETH:10.195.133.20 , TIMEOUT=50")

' Variableオブジェクト
Dim var as CaoVariable
Set var = caoCtrl.AddVariable("PLCELEM<D0100>", "DeviceType = D, Number = 100")
```

4.2.3. CaoVariable クラス**4.2.3.1. Index プロパティ**

変数番号を Long 型 (4 バイト整数型) で取得します。この番号は、該当の変数が、CaoController クラスの保持する変数コレクションの何番目のデータかを示します。

使用例

```
Dim caoEng As CaoEngine      ' Engineオブジェクト
Dim caoWs As CaoWorkspace   ' Workspaceオブジェクト
Dim caoCtrl As CaoController ' Controllerオブジェクト

Set caoEng = New CaoEngine
Set caoWS = caoEng.Workspaces.Item(0)
Set caoCtrl = caoWS.AddController("EZSocketCNC",
                                   "GaoProv. MITSUBISHI. EZSocketCNC",
                                   "",
                                   "CONN = ETH:10.195.133.20 , TIMEOUT=50")

' Variableオブジェクト
Dim var as CaoVariable
Set var = caoCtrl.AddVariable("PLCELEM<D0100>", "DeviceType = D, Number = 100")
```

Index取得

```
Dim index As Long
Index = var.Index
```

4.2.3.2. Name プロパティ

CaoController クラスの AddVariable メソッドで指定された変数名を取得します。

使用例

```
Dim caoEng As CaoEngine      ' Engineオブジェクト
Dim caoWs As CaoWorkspace    ' WorkSpaceオブジェクト
Dim caoCtrl As CaoController ' Controlleオブジェクト

Set caoEng = New CaoEngine
Set caoWS = caoEng.Workspaces.Item(0)
Set caoCtrl = caoWS.AddController("EZSocketCNC",
                                   "CaoProv. MITSUBISHI. EZSocketCNC",
                                   "",
                                   "CONN = ETH:10.195.133.20 ,TIMEOUT=50")

' Variableオブジェクト
Dim var as CaoVariable
Set var = caoCtrl.AddVariable("PLCELEM<D0100>", "DeviceType = D, Number = 100")

Debug.Print varD0100.Name
```

4.2.3.3. Value プロパティ

接続した CNC からデータを取得/設定します。変数名によって動作が異なります。詳細は、4.3.1 を参照してください。

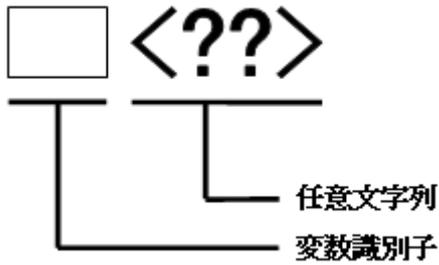
4.3. 変数一覧

各クラスで使用可能な変数一覧を定義します。なお変数は、CaoVariable クラスのオブジェクトを指します。

なお変数は、CaoVariable クラスのオブジェクトを指します。複数変数を登録（オプションのみ変更したい場合等に有用）するために任意の文字列を付与することが可能です。

変数名に任意文字列を付与するための書式を以下に示します。

複数変数共通指定書式



4.3.1. CaoController クラス変数

変数名	説明	Value		参照
		get	put	
@MAKER_NAME	メーカー名を取得します。	○	-	P. 20
@VERSION	DLL バージョンを取得します。	○	-	P. 21
PLCELEM<??>	PLC のデバイスの一要素ずつにアクセスします。	○	○	P. 21
PLCRANGE<??>	PLC のデバイスに範囲アクセスします。	○	○	P. 23
COMMON<??>	コモン変数にアクセスします。	○	-	P. 23
TOOL_OFFSET<??>	工具補正量を取得します。	○	-	P. 25
PRG_NAME<??>	現在、サーチ完了または自動運転中のプログラム番号を返します。	○	-	P. 27

4.3.1.1. @MAKER_NAME

メーカー名の取得をします。

使用例

```
Dim engine As caoEngine
Dim workspace As caoWorkspace
Dim ctrl As CaoController
Set engine = New caoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set ctrl = workspace.AddController("EZSocketCNC", _
    "CaoProv.MITSUBISHI.EZSocketCNC", _
    "", _
    "CONN=ETH:10.195.133.20, TIMEOUT=50")

' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("@MAKER_NAME")

' 値取得
Dim strVal As String
strVal = var.value
```

データ型

型説明	
VT_BSTR	メーカー名を取得します。

4.3.1.2. @VERSION

DLL のバージョンの取得をします。

使用例

```
Dim engine As caoEngine
Dim workspace As caoWorkspace
Dim ctrl As CaoController
Set engine = New caoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set ctrl = workspace.AddController("EZSocketCNC", _
    "CaoProv. MITSUBISHI. EZSocketCNC", _
    "", _
    "CONN=ETH:10.195.133.20, TIMEOUT=50")

' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("@VERSION")

' 値取得
Dim value As String
value = var.value
```

データ型

型説明	
VT_BSTR	DLL のバージョンを取得します。 *. *.*

4.3.1.3. PLCELEM<??>

PLC のデバイスの一要素ずつにアクセスします。PLCELEM の後に任意の文字列を入力してください。以下にオプションと取得/設定時の戻り値を記します。

使用例

```
Dim engine As caoEngine
Dim workspace As caoWorkspace
Dim ctrl As CaoController
Set engine = New caoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set ctrl = workspace.AddController("EZSocketCNC", _
    "CaoProv. MITSUBISHI. EZSocketCNC", _
    "", _
    "CONN=ETH:10.195.133.20, TIMEOUT=50")

' 変数追加
```

```
Dim var As CaoVariable
```

```
Set var = ctrl.AddVariable("PLCELEM<R0100>", "DeviceType=R, Number=100")
```

’ 値取得

```
Dim value As Long
```

```
value = var.value
```

オプション

オプション	必須	説明	値範囲	デフォルト値
DeviceType=	○	アクセスデバイスを指定します。	----	----
Number=	○	アクセス先頭アドレスを 16 進数で指定します。	0 -	----
DataType=	--	Value プロパティのデータ型を指定します。 省略した場合は、アクセスデバイスのデフォルトの型となります。	1: Bit 8: Byte 16: Word 32: Dword	アクセスデバイスによります。
Elem=	--	何点読み出すかを指定します。	1 -	1
IsArray=	--	Elem が 1 の時に配列にするかを指定します。	true/false	false

データ型

DataType	型説明	
1	VT_BOOL	VT_BOOL 型で読み書きします。
8	VT_UI1	VT_UI1 型で読み書きします。
16	VT_I2	VT_I2 型で読み書きします。
32	VT_I4	VT_I4 型で読み書きします。

※ Elem に 2 以上を指定する、もしくは、IsArray に True を指定すると配列になります。

デバイス種類

デバイス名	名称
B	カウンタ (固定カウンタ)
C ※ 3	カウンタコイル
D	データレジスタ
E	特殊リレー
F	アラームメッセージインタフェース。一時記憶。
G	一時記憶
I	I デバイス

デバイス名	名称
J	J デバイス
L	ラッチリレー (バックアップメモリ)
M	一時記憶
Q	Q デバイス
R	ファイルレジスタ※1
SM	特殊リレー※1
SB	リンク用特殊リレー
SD	特殊レジスタ
ST	積算タイマ
SW	リンク用特殊レジスタ
T ※3	10ms 単位タイマコイル
U	PLC への入力信号 2 系統用※1
V	V デバイス
W	PLC への出力信号 2 系統用※1
X	PLC への入力信号※1
Y	PLC への出力信号※1
ZR	ファイルレジスタ

※ 1 : このデバイスは用途が決まっています。機械側との入出力信号に対応したデバイス以外は未定義の空きデバイスでも使用しないでください。

※ 2 : これらのデバイスは読み出し専用です。

※ 3 : C 及び T のデバイスでは、CNC 上では C は CI と T は TI と表示されますが、ここでは C と T で指定してください。

4.3.1.4. PLCRANGE<??>

PLC のデバイスに範囲アクセスします。PLCRANGE の後に任意の文字列を入力してください。以下にオプションと取得/設定時の戻り値を記します。

使用例

```
Dim engine As caoEngine
Dim workspace As caoWorkspace
Dim ctrl As CaoController
Set engine = New caoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set ctrl = workspace.AddController("EZSocketCNC", _
    "CaoProv.MITSUBISHI.EZSocketCNC", _
    "", _
    "CONN=ETH:10.195.133.20, TIMEOUT=50")
```

変数追加

```
Dim var As CaoVariable
```

```
Set var = ctrl.AddVariable("PLCRANGE<R0100>", "DeviceType=R, Number=100, Elem=100")
```

値取得

```
Dim value(99) As Long
```

```
value = var.value
```

オプション

オプション	必須	説明	値範囲	デフォルト値
DeviceType=	○	アクセスデバイスを指定します。	----	----
Number=	○	アクセス先頭アドレスを 16 進数で指定します。	0 -	----
DataType=	---	Value プロパティのデータ型を指定します。 省略した場合は、アクセスデバイスのデフォルトの型となります。	1: Bit 8: Byte 16: Word 32: Dword	アクセスデバイスによります。
Elem=	---	何点読み出すかを指定します。 DataType 毎に指定できる最大値が異なります。	Bit: 2 - 1280 Byte: 2 - 1280 Word: 2 - 640 Dword: 2 - 320	2

データ型

DataType	型説明
1	VT_BOOL VT_ARRAY VT_BOOL 型で読み書きします。
8	VT_UI1 VT_ARRAY VT_UI1 型で読み書きします。
16	VT_I2 VT_ARRAY VT_I2 型で読み書きします。
32	VT_I4 VT_ARRAY VT_I4 型で読み書きします。

デバイス種類

PLCELEM<??>と同様です。

4.3.1.5. COMMON<??>

コモン変数にアクセスします。COMMON の後に任意の文字列を入力してください。以下にオプションと値取得時の戻り値を記します。

使用例

```
Dim engine As caoEngine
```

```
Dim workspace As caoWorkspace
```

```
Dim ctrl As CaoController
Set engine = New caoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set ctrl = workspace.AddController("EZSocketCNC", _
    "CaoProv.MITSUBISHI. EZSocketCNC", _
    "", _
    "CONN=ETH:10.195.133.20, TIMEOUT=50")
```

’ 変数追加

```
Dim var As CaoVariable
Set var = ctrl.AddVariable("COMMON<100>", "Number=100")
```

’ 値取得

```
Dim value As Double
value = var.value
```

オプション

オプション	必須	説明	値範囲	デフォルト値
Head=	--	システムを指定します。	1 -	1
Number=	○	コモン変数番号を 10 進数で指定します。	1 -	----
Elem=	--	何点読み出すかを指定します。	1 -	1
IsArray=	--	Elem が 1 の時に配列にするかを指定します。	true/false	false

データ型

型説明			
VT_ARRAY VT_VARIANT			
i	VT_VARIANT	読み出したコモン変数の状況により取得される型が異なります。 VT_EMPTY: コモン変数の値が未設定時 VT_R8: コモン変数に値が設定されている場合	

4.3.1.6. TOOL_OFFSET<??>

工具補正量を取得します。TOOL_OFFSET の後に任意の文字列を入力してください。接続 CNC によって工具データが異なるので、接続 CNC ごとの工具補正量の戻り値を別表に示します。

使用例

```
Dim engine As caoEngine
Dim workspace As caoWorkspace
Dim ctrl As CaoController
Set engine = New caoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set ctrl = workspace.AddController("EZSocketCNC", _
```

```
"CaoProv.MITSUBISHI. EZSocketCNC", _
""
"CONN=ETH:10.195.133.20, TIMEOUT=50")
```

' 変数追加

```
Dim var As CaoVariable
```

```
Set var = ctrl.AddVariable("TOOL_OFFSET<1>", "Number = 1")
```

' 値取得

```
Dim values() As Double
```

```
values = var.value
```

オプション

オプション	必須	説明	値範囲	デフォルト値
Head=	--	システムを指定します.	1 -	1
Number=	○	工具番号を 10 進数で指定します.	1 -	----
Elem=	--	読み出し工具数	1 -	1
isArray=	--	Elem が 1 の時に配列にするかを指定します.	true/false	false

データ型

表 4-2 マシニングセンタ 1

型説明			
VT_VARIANT VT_ARRAY		読み出したすべての工具の補正量	
i	VT_ARRAY VT_R8		工具 1 つのデータ
	0	VT_R8	工具補正量

※ $0 \leq i < \text{Elem}$

表 4-3 マシニングセンタ 2

型説明			
VT_VARIANT VT_ARRAY		読み出したすべての工具の補正量	
i	VT_ARRAY VT_R8		工具 1 つのデータ
	0	VT_R8	工具長寸法補正量
	1	VT_R8	工具長摩耗補正量
	2	VT_R8	工具径寸法補正量
	3	VT_R8	工具径摩耗補正量

※ $0 \leq i < \text{Elem}$

表 4-4 旋盤

型説明			
VT_VARIANT VT_ARRAY		読み出したすべての工具の補正量	
i	VT_ARRAY VT_R8		工具 1 つのデータ

型説明			
	0	VT_R8	刃先摩耗補正量 X
	1	VT_R8	刃先摩耗補正量 Z
	2	VT_R8	刃先摩耗補正量 C(Z)
	3	VT_R8	刃先寸法補正量 X
	4	VT_R8	刃先寸法補正量 Z
	5	VT_R8	刃先寸法補正量 C(Z)
	6	VT_R8	工具半径 R
	7	VT_R8	工具半径 R 摩耗量
	8	VT_R8	仮想刃先点番号 P (0 - 8)

※ $0 \leq i < \text{Elem}$

4.3.1.7. PRG_NAME<??>

現在、サーチ完了または自動運転中のプログラム番号を返します。PRG_NAME の後に任意の文字列を入力してください。

使用例

```
Dim engine As caoEngine
Dim workspace As caoWorkspace
Dim ctrl As CaoController
Set engine = New caoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set ctrl = workspace.AddController("EZSocketCNC", _
    "CaoProv.MITSUBISHI.EZSocketCNC", _
    "", _
    "CONN=ETH:10.195.133.20, TIMEOUT=50")
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("PRG_NAME<1>")
' 値取得
Dim value As String
value = var.value
```

オプション

オプション	必須	説明	値範囲	デフォルト値
Head=	--	システムを指定します。	1 -	1
ProgramType=	--	プログラムの種類を指定します。	0: メイン 1: サブ	0

データ型**型説明**

VT_BSTR	現在, サーチ完了または自動運転中のプログラム番号を UNICODE 文字列として返却されます.
---------	--------------------------------------------------

5. CNC プロバイダエラーコード

本プロバイダには、以下の独自エラーコードとワーニングコードが存在します。API からのエラーコードについては付録 B に記載します。

ORiN2 の共通エラーについては、「[ORiN2 プログラミングガイド](#)」のエラーコードの章を参照してください。

表 5-1 独自エラーコード表

エラー番号	説明
0x80110001	必須のオプションが入力されていません。
0x80110002	指定した値が範囲外です。
0x80110003	サポート外のデバイスを指定しています。
0x80110004	排他制御に失敗しました。このエラーが頻発する場合は、AddControllerのTimeoutにより長い時間を指定して再度お試しください。
0x80110005	設定されたTimeout時間内に、コールバックメソッドが呼び出されませんでした。このエラーが頻発する場合は、AddControllerのTimeoutにより長い時間を指定して再度お試しください。
0x80110006	実行エラーです。このエラーコードを受け取った場合は開発元へお問い合わせください。
0x80110007	値取得時に想定外の例外が発生しました。
0x80110008	値設定時に想定外の例外が発生しました。
0x8011000A	存在しない系統番号が指定されています。存在する系統番号を指定して、変数の再作成してください。

付録A. API 対応表

CaoWorkspace::CaoAddController	
API 関数名	
IEZNCCommunication2::SetTCP/IPProtocol	
IEZNCCommunication3::Open2	
IEZNCCommunication3::Close	※CaoWorkspace::AddController のオプションで Multithread=false を指定されたとき
IEZNCCommunication3::Close2	※CaoWorkspace::AddController のオプションで Multithread=true を指定されたとき

CaoVariable

変数名	Get_Value	Set_Value
PLCELEM<??>	IEZNCDevice::SetDevice IEZNCDevice::ReadDevice	IEZNCDevice::SetDevice IEZNCDevice::WriteDevice
PLCRANGE<??>	IEZNCDevice::ReadBlockDevice	IEZNCDevice::WriteBlockDevice
COMMON<??>	IEZNCCommunication::GetHead IEZNCCommunication::SetHead IEZNCSystem::GetSystemInformation IEZNCCommonVariable2::CommonVRead	---
TOOL_OFFSET<??>	IEZNCCommunication::GetHead IEZNCCommunication::SetHead IEZNCSystem::GetSystemInformation IEZNCTool::GetOffset IEZNCTool::GetType	---
PRG_NAME<??>	IEZNCCommunication::GetHead IEZNCCommunication::SetHead IEZNCSystem::GetSystemInformation	---

付録B. API からのエラー一覧

エラー番号	説明
0x80A00101	通信回線がオープンされていません
0x80A00104	2重オープンエラー
0x80A00105	引数のデータタイプが不正
0x80A00106	引数のデータ範囲が不正

エラー番号	説明
0x80A00107	サポートしていない
0x80A00109	通信回線がオープンできません
0x80A0010A	引数がNULLポインタです
0x80A0010B	引数のデータ不正
0x80A0010C	COMMポートハンドルエラー
0x80B00101	メモリの確保ができない
0x80B00102	EZSocketPcのエラーが取得できない
0x80B00201	モード指定不正
0x80B00202	未ファイルオープン
0x80B00203	ファイルが既に存在する
0x80B00204	既にファイルオープンしている
0x80B00205	テンポラリファイルを作成できない
0x80B00206	書き込みモード指定でファイルオープンしていない
0x80B00207	書き込みデータサイズ不正
0x80B00208	書き込みできない状態
0x80B00209	読み出しモード指定でファイルオープンしていない
0x80B0020A	読み出しできない状態
0x80B0020B	テンポラリファイルを作成できない
0x80B0020C	ファイルが存在しない (READモード)
0x80B0020D	ファイルがオープンできない
0x80B0020E	ファイルのパスが不正
0x80B0020F	読み出しファイルが不正
0x80B00210	書き込みファイルが不正
0x80B00301	オートメーション呼び出しでローカル接続時のホスト名が不正
0x80B00302	TCP/IP通信が設定されていない
0x80B00303	既に通信中なので設定できない
0x80B00304	下位モジュールがない
0x80B00305	EZSocketPcオブジェクトが生成できない
0x80B00401	データが存在しない
0x80B00402	データ重複
0x80B00501	パラメータ情報ファイルがない
0x80020190	NCカード番号不正
0x80020102	デバイスがオープンされていない
0x80020132	コマンド不正

エラー番号	説明
0x80020133	通信パラメータデータ範囲不正
0x80030143	ファイルシステムに異常がある
0x80030191	ディレクトリが存在しない
0x8003019B	ドライブが存在しない
0x800301A2	ディレクトリが存在しない
0x800301A8	ドライブが存在しない
0x80050D90	系統, 軸指定が不正
0x80050D02	アラーム種類が不正
0x80050D03	NCとPC間の通信データにエラーがある
0x80041194	寿命管理データの種類の指定不正
0x80041195	設定データ範囲オーバ
0x80041196	設定工具番号不一致
0x80041197	指定工具番号が仕様外
0x80040190	系統, 軸指定が不正
0x80040191	大区分番号不正
0x80040192	小区分番号不正
0x80040196	アプリケーションが用意したバッファに入りきらない
0x80040197	データタイプ不正
0x8004019D	データが読み出せない状態にある
0x8004019F	書き込み専用データ
0x800401A0	軸指定不正
0x800401A1	データ番号不正
0x800401A3	読み出しデータなし
0x8004019A	読み出しデータ範囲不正
0x80040290	系統, 軸指定が不正
0x80040291	大区分番号不正
0x80040292	小区分番号不正
0x80040296	アプリケーションが用意したバッファに入りきらない
0x80040297	データタイプ不正
0x8004029B	読み出し専用データ
0x8004029E	データが書き込めない状態にある
0x800402A0	軸指定不正
0x8004024D	安全パスワードロック中
0x800402A2	SRAM開放パラメータ不正によりフォーマット中止した

エラー番号	説明
0x800402A4	編集ファイルを登録できない(既に編集中)
0x800402A5	編集ファイルを解除できない
0x800402A3	書き込み先データなし
0x8004029A	書き込みデータ範囲不正
0x800402A6	安全パスワード未設定
0x800402A7	安全データ整合性チェックエラー
0x800402A9	安全用データタイプ不
0x800402A8	工具データソート中で書き込みできない
0x80040501	高速読み出し登録されていない
0x80040402	プライオリティ指定不正
0x80040401	登録数をオーバした
0x80040490	アドレス不正
0x80040491	大区分番号不正
0x80040492	小区分番号不正
0x80040497	データタイプ不正
0x8004049B	読み出し専用データ
0x8004049D	データが読み出せない状態にある
0x8004049F	書き込み専用データ
0x800404A0	軸指定不正
0x80040BA3	再ねじ切り位置設定なし
0x80030101	既に別ディレクトリがオープンされている
0x80030103	データサイズオーバ
0x80030148	ファイル名が長い
0x80030198	ファイル名フォーマットが不正
0x80030190	オープンされていない
0x80030194	ファイル情報リードエラー
0x80030102	すでに別ディレクトリがオープンされている(PCのみ)
0x800301A0	オープンされていない
0x800301A1	ファイルが存在しない
0x800301A5	ファイル情報リードエラー
0x80030447	コピーできない状態にある(運転中)
0x80030403	登録本数オーバ
0x80030401	コピー先ファイルが既に存在する
0x80030443	ファイルシステムに異常がある

エラー番号	説明
0x80030448	ファイル名が長い
0x80030498	ファイル名フォーマットが不正
0x80030404	メモリ容量オーバ
0x80030491	ディレクトリが存在しない
0x8003049B	ドライブが存在しない
0x80030442	ファイルが存在しない
0x80030446	コピーできない状態にある (PLC動作中)
0x80030494	転送元ファイルが読めない
0x80030495	転送先ファイルに書き込めない
0x8003044A	コピーできない状態にある (プロテクト中)
0x80030405	照合エラー
0x80030449	照合機能をサポートしていない
0x8003044C	ファイルコピー中
0x80030490	ファイルがオープンされていない
0x8003044D	安全パスワードロック中
0x8003049D	ファイルフォーマット不正
0x8003049E	パスワードが異なる
0x800304A4	ファイルが生成できない (PCのみ)
0x800304A3	ファイルをオープンできない (PCのみ)
0x80030402	コピー先ファイルが既に存在する
0x800304A7	ファイル名フォーマットが不正
0x800304A2	ディレクトリが存在しない
0x800304A8	ドライブが存在しない
0x800304A1	ファイルが存在しない
0x800304A5	転送元ファイルが読めない
0x800304A6	転送先ファイルに書き込めない
0x80030406	ディスク容量オーバ
0x800304A0	ファイルがオープンされていない
0x80030201	削除できないファイル
0x80030242	ファイルが存在しない
0x80030243	ファイルシステムに異常がある
0x80030247	削除できない状態にある (運転中)
0x80030248	ファイル名が長い
0x8003024A	ファイルが削除できない状態にある (プロテクト中)

エラー番号	説明
0x80030291	ディレクトリが存在しない
0x80030298	ファイル名フォーマットが不正
0x8003029B	ドライブが存在しない
0x80030202	削除できないファイル
0x800302A7	ファイル名フォーマットが不正
0x800302A2	ディレクトリが存在しない
0x800302A8	ドライブが存在しない
0x800302A1	ファイルが存在しない
0x80030301	新ファイル名が既に存在する
0x80030342	ファイルが存在しない
0x80030343	ファイルシステムに異常がある
0x80030347	リネームできない状態にある(運転中)
0x80030348	ファイル名が長い
0x8003034A	リネームできない状態にある(プロテクト中)
0x80030391	ディレクトリが存在しない
0x80030398	ファイル名フォーマットが不正
0x8003039B	ドライブが存在しない
0x80030303	リネームできない
0x80030305	新旧ファイル名が同じ
0x80030302	新ファイル名が既に存在する
0x800303A7	ファイル名フォーマットが不正
0x800303A2	ディレクトリが存在しない
0x800303A8	ドライブが存在しない
0x800303A1	ファイルが存在しない
0x80030691	ディレクトリが存在しない
0x8003069B	ドライブが存在しない
0x80030643	ファイルシステムに異常がある
0x80030648	ファイル名フォーマットが不正
0x800306A2	ディレクトリが存在しない(PCのみ)
0x800306A8	ドライブが存在しない(PCのみ)
0x80030701	アプリケーションが用意したバッファに入りきらない
0x80030794	ドライブ情報リードエラー
0x82020001	すでにオープンされている
0x82020002	オープンされていない

エラー番号	説明
0x82020004	カードが存在しない
0x82020006	チャンネル番号不正
0x82020007	ファイルディスクプリタ不正
0x8202000A	コネクタされていない
0x8202000B	クローズされていない
0x82020014	タイムアウト
0x82020015	データ不正
0x82020016	キャンセル要求により終了した
0x82020017	パケットサイズ不正
0x82020018	タスク終了により終了した
0x82020032	コマンド不正
0x82020033	設定データ不正
0x80060001	データリードキャッシュが無効
0x80060090	アドレス不正
0x80060091	大区分番号不正
0x80060092	小区分番号不正
0x80060097	データタイプ不正
0x8006009A	データ範囲不正
0x8006009D	データが読み出せない状態にある
0x8006009F	データタイプ不正
0x800600A0	軸指定不正
0x80070140	作業領域を確保できない
0x80070142	ファイルをオープンできない
0x80070147	ファイルがオープンできない状態にある(運転中)
0x80070148	ファイルパスが長い
0x80070149	未サポート (CF未対応)
0x80070192	すでにオープンされている
0x80070199	最大ファイルオープン数を越えた
0x8007019F	工具データソート中でオープンができない
0x800701B0	安全パスワードが未認証
0x80070290	ファイルがオープンされていない
0x80070340	作業領域を確保できない
0x80070347	ファイルが生成できない状態にある(運転中)
0x80070348	ファイルパスが長い

エラー番号	説明
0x80070349	未サポート (CF未対応)
0x80070392	すでに生成されている
0x80070393	ファイルを生成できない
0x80070399	最大ファイルオープン数を越えた
0x8007039B	ドライブが存在しない
0x80070490	ファイルがオープンされていない
0x80070494	ファイル情報リードエラー
0x80070549	書き込み不可
0x80070590	ファイルがオープンされていない
0x80070595	ファイル書き込みエラー
0x80070740	ファイル削除エラー
0x80070742	ファイルが存在しない
0x80070747	ファイルが削除できない状態にある (運転中)
0x80070748	ファイルパスが長い
0x80070749	未サポート (CF未対応)
0x80070792	ファイルがオープンされている
0x8007079B	ドライブが存在しない
0x80070842	ファイルが存在しない
0x80070843	リネームできないファイル
0x80070848	ファイルパスが長い
0x80070849	未サポート (CF未対応)
0x80070892	ファイルがオープンされている
0x80070899	最大ファイルオープン数を越えた
0x8007089B	ドライブが存在しない
0x80070944	コマンド不正 (未対応)
0x80070990	オープンされていない
0x80070994	リードエラー
0x80070995	ライトエラー
0x80070996	アプリケーションが用意したバッファに入りきらない
0x80070997	データタイプ不正
0x80070949	未サポート (CF未対応)
0x80070A40	作業領域を確保できない
0x80070A47	ディレクトリがオープンできない状態にある (運転中)
0x80070A48	ファイルパスが長い

エラー番号	説明
0x80070A49	未サポート (CF未対応)
0x80070A91	ディレクトリが存在しない
0x80070A92	すでにオープンされている
0x80070A99	最大ディレクトリオープン数を越えた
0x80070A9B	ドライブが存在しない
0x80070B90	ディレクトリがオープンされていない
0x80070B91	ディレクトリが存在しない
0x80070B96	アプリケーションが用意したバッファに入りきらない
0x80070D90	ディレクトリがオープンされていない
0x80070E48	ファイルパスが長い
0x80070E49	サポート (CF未対応)
0x80070E94	ファイル情報読み込みエラー
0x80070E99	最大ファイルオープン数を越えた
0x80070E9B	ドライブが存在しない
0x80070F48	ファイルパスが長い
0x80070F49	未サポート (CF未対応)
0x80070F94	ファイル情報読み込みエラー
0x80070F90	ファイルがオープンされていない
0x80070F9B	ドライブが存在しない
0x8007099C	SRAM開放パラ不正でフォーマット中止
0xF00000FF	引数が不正
0xFFFFFFFF	データが読み出せない/書き込めない状態
0x01XXXXXX	詳細につきましては、下記マニュアルを参照してください。 ・ EZSocketスタンダードリファレンスマニュアル (MELSEC編) (BAD-801Q013) ・ EZSocket プロFX CPU対応 リファレンスマニュアル (MELSEC編) (BAD-801Q025)
0x02XXXXXX	
0x03XXXXXX	
0x04XXXXXX	
0x10XXXXXX	