

# SLMP プロバイダ

Version 1.0.3

## ユーザーズ ガイド

March 26, 2024

**【備考】**

本書で使用している画像の一部は「SLMP リファレンスマニュアル」から引用しています。

**【改版履歴】**

バージョン	日付	内容
1.0.0	2018-02-01	初版.
1.0.1	2019-09-11	Param オプションの格納処理 バグ修正
1.0.2	2020-01-21	AddController 失敗時の処理修正
1.0.3	2020-04-07	Series オプション追加 AddVariable 時の処理修正 排他制御処理修正
	2020-11-4	付録 他局アクセス 追加
	2021-10-29	指定可能な要求先ユニット I/O 番号, 局番号の一覧を追加. VT 項目の I1, UI1 の注記を修正.
	2023-03-26	誤記修正

**【動作確認機器】**

機種	バージョン	注意事項
R02CPU		MELSEC iQ-R シリーズ
ツール	v1.1	SLMP コンフォーマンステストツール

## 目次

1. はじめに.....	5
2. プロバイダの概要.....	6
2.1. 概要.....	6
2.2. 使用上の注意点.....	6
2.2.1. 機器との接続について.....	6
2.3. メソッド・プロパティ.....	7
2.3.1. CaoWorkspace::AddController メソッド.....	7
2.3.1.1. Conn オプション.....	7
2.3.1.2. Series オプション.....	8
2.3.1.3. サンプルプログラム.....	9
2.3.2. CaoController::AddVariable メソッド.....	10
2.3.2.1. Device オプション.....	11
2.3.2.2. Address オプション.....	12
2.3.2.3. Param オプション.....	15
2.3.2.4. VT オプション.....	17
2.3.2.5. Elem オプション.....	19
2.3.2.6. Array オプション.....	19
2.3.2.7. ランダム読み出し/書き込み.....	20
2.3.2.8. サンプルプログラム.....	21
2.4. エラーコード.....	23
3. コマンドリファレンス.....	24
3.1. Execute コマンド.....	24
3.1.1. CaoController::Execute(“Raw”) コマンド.....	24
3.1.2. サンプルプログラム.....	25
4. 付録.....	28
4.1. 他局アクセス.....	28
4.1.1. 親 PLC の設定.....	29
4.1.1.1. 内蔵 Ethernet ユニットの設定.....	29
4.1.1.2. 拡張 Ethernet ユニット 1 の設定.....	30
4.1.1.3. 拡張 Ethernet ユニット 2 の設定.....	31
4.1.2. 子 PLC の設定.....	32

---

4.1.2.1. 内蔵 Ethernet ユニットの設定.....	32
4.1.2.2. 拡張 Ethernet ユニットの設定.....	33
4.1.3. 接続パラメータ設定.....	34

## 1. はじめに

本書は、相手機器と SLMP(Seamless Message Protocol)<sup>(1)</sup>によるデータ交信機能で、機器のデータの読出し、書込みなどを行う CAO プロバイダのユーザーズガイドです。本書で扱う CAO プロバイダ(CaoProvSLMP.dll)を SLMP プロバイダと呼びます。

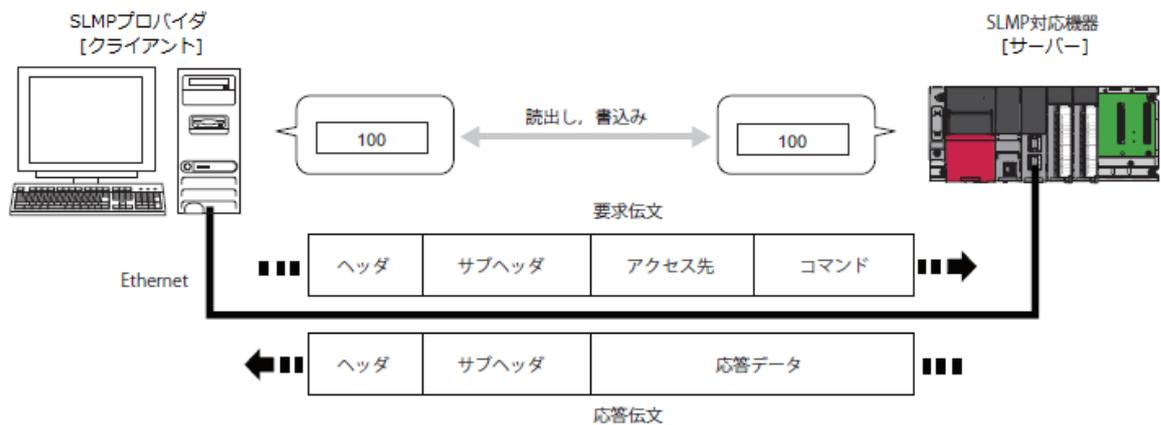
SLMP プロバイダは、Ethernet を使用して、機器や外部機器(パソコンや表示器など)から SLMP 対応機器にアクセスするためのプロトコル通信を行います。

SLMP の制御手順で伝文を送受信できる機器であれば SLMP による交信ができます。

SLMP プロバイダは、SLMP 対応機器(サーバー)に対してクライアントとして機能します。

SLMP プロバイダは、ST 型、バイナリ通信で通信を行っています。

通信の詳細や設定方法については「SLMP リファレンスマニュアル」、各ユニットのマニュアルを参照してください。



第 2 章に SLMP プロバイダの概要、変数の詳細を記載しています。

<sup>1</sup> Seamless Message Protocol (SLMP) は、CC-Link 協会が提唱するシームレス通信用のプロトコルです。

## 2. プロバイダの概要

### 2.1. 概要

SLMP プロバイダは、SLMP 対応機器と通信するためのユニットに依存する部分を吸収し CAO プロバイダ・インターフェース仕様で規定された機能を提供する CAO プロバイダです。そのファイル形式は DLL(Dynamic Link Library)であり、CAO エンジンから使用時に動的にロードされます。SLMP プロバイダを使用するにあたっては ORiN2SDK をインストールするか、下表を参照して手作業でレジストリ登録を行う必要があります。

表 2-1 SLMP プロバイダ

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

### 2.2. 使用上の注意点

#### 2.2.1. 機器との接続について

PC と機器の接続につきましては、使用しているユニットのユーザーズマニュアルを参照して下さい。



## 2.3. メソッド・プロパティ

### 2.3.1. CaoWorkspace::AddController メソッド

SLMP プロバイダでは Controller オブジェクトの生成時に、イーサネット (TCP, UDP) 通信の初期処理を行います。接続時にオプション文字列でデバイスを指定します。

**書式** AddController( <bstrCtrlName:BSTR>,<bstrProvName:BSTR>,  
<bstrPcName:BSTR >,<bstrOption:BSTR> )

bstrCtrlName : [in] コントローラ名  
 bstrProvName : [in] プロバイダ名. 固定値 =”CaoProv.SLMP”  
 bstrPcName : [in] プロバイダの実行マシン名  
 bstrOption : [in] オプション文字列

以下にオプション文字列に指定するリストを示します。

**表 2-2 CaoWorkspace::AddController のオプション文字列**

オプション	説明
Conn=<接続パラメータ>	必須. 通信形態とその接続パラメータを設定します。 (参照 2.3.1.1)
Series[=<通信タイプ>]	通信のタイプを指定します。 (参照 2.3.1.2)
Timeout[=<タイムアウト時間>]	送受信時のタイムアウト時間をミリ秒で設定します。 (デフォルト:3000)
Retry[=<リトライ回数>]	送受信時の通信リトライ回数の設定をします。 (デフォルト:0)

#### 2.3.1.1. Conn オプション

以下に Conn オプションの接続パラメータ文字列を示します。

“Conn=ETH:<Dest IP Address>:<Dest Port No>”

“Conn=TCP:<Dest IP Address>:<Dest Port No>”

“Conn=UDP:<Dest IP Address>:<Dest Port No>”

< Dest IP Address > : 接続先の IP アドレス.

< Dest Port No > : 接続先のポート番号.

### 2.3.1.2. Series オプション

通信のタイプを指定します。

設定内容については表 2-3 を参照してください。

Series オプションを省略した場合のデフォルトは 0 となります。

表 2-3 Series オプション設定一覧

値	意味	備考
0	16ビット長アドレス(Q/L シリーズ互換)	・16ビット長アドレスでアクセスします。 MELSEC Q/L シリーズをお使いの場合はこちらを指定してください。 ・16進6桁分のアドレスまでアクセスが可能です
1	32ビット長アドレス	・MELSEC R シリーズ, 32ビット長アドレスに対応している機器をお使いの場合, こちらを指定することにより, 16進8桁分のアドレスまでアクセスが可能になります。

(例) “Conn=ETH:192.168.0.1:1025, Series=0” MELSEC Q/L 互換モード (16ビット長アドレス)

(例) “Conn=ETH:192.168.0.1:1025, Series=1” MELSEC R シリーズ (32ビット長アドレス)

### 2.3.1.3. サンプルプログラム

AddControllerのサンプルプログラムを以下に示します。

```
HRESULT hr = S_OK;
ICaoEngine* pEng = NULL;
ICaoWorkspaces *pWss = NULL;
ICaoWorkspace *pWs = NULL;
ICaoController *pCtrl = NULL;

// CaoEngine の生成
hr = CoCreateInstance(CLSID_CaoEngine,
                    NULL,
                    CLSCTX_LOCAL_SERVER,
                    IID_ICaoEngine,
                    (void **)&pEng);

if (FAILED(hr)) {
    goto EndProc;
}

// CaoWorkspace コレクションの取得
hr = pEng->get_Workspaces(&pWss);
if (FAILED(hr)) {
    goto EndProc;
}

// CaoWorkspace の取得
hr = pWss->Item(CComVariant(0L), &pWs);
if (FAILED(hr)) {
    goto EndProc;
}

// CaoController の生成
hr = pWs->AddController(CComBSTR(L"SLMP_TEST"),
                      CComBSTR(L"CaoProv.SLMP"),
                      CComBSTR(L""),
                      CComBSTR(L"Conn=ETH:169.254.225.206:61442"),
                      &pCtrl);

if (FAILED(hr)) {
    goto EndProc;
}

// ここに必要な処理を入れる
// 値の設定,取得など

EndProc:
if (pCtrl) pCtrl->Release();
if (pWs) pWs->Release();
if (pWss) pWss->Release();
if (pEng) pEng->Release();
```

### 2.3.2. CaoController::AddVariable メソッド

機器情報を取得/設定する変数オブジェクトを作成します。

**書式** AddVariable( <bstrName:BSTR > ,<bstrOption:BSTR> )

bstrName : [in] 変数名. 管理する為の任意で一意的な文字列.

bstrOption : [in] オプション文字列

表 2-4 CaoController::AddVariable のオプション文字列

オプション	意味
Device=<デバイスコード>	必須. アクセス先のデバイスのデバイスコードを指定します. (参照 2.3.2.1)
Address=<先頭アドレス>	必須. アクセス先のデバイスの先頭アドレスを指定します. アドレスは 10 進数, 16 進数値で指定します. (「表 2-5 デバイス一覧」の「アドレス指定方法」を参照) 複数アドレスの指定が可能. (「2.3.2.7 ランダム読み出し」を参照)
Param[=<変数パラメータ>]	変数のパラメータを設定します. (参照 2.3.2.3)
VT[=<データ型>]	デバイスメモリに入出力する場合に使用するデータの型を指定します. (参照 2.3.2.4)
Elem[=<要素数>]	要素数を指定します. 16 進数で指定する場合は以下のフォーマットで入力してください. 0x[0-9,A-F]+, &H[0-9,A-F]+, [0-9,A-F]+H (参照 2.3.2.5)
Array[=< True or False >]	一要素の読み込み時も配列の形式で値を取得するかどうかを指定します. (参照 2.3.2.6)

### 2.3.2.1. Device オプション

Device オプションにデバイスコードを指定することで、機器にアクセスすることができます。

ビットデバイス : SM, X, Y, M, L, F, V, B, TS, TC, LTS, LTC, SS, SC, LSTS, LSTC,  
CS, CC, LCS, LCC, SB, DX, DY

ワードデバイス : SD, D, W, TN, SN, CN, SW, Z, R, ZR, RD

ダブルワードデバイス: LTN, LSTN, LCN, LZ

### 2.3.2.2. Address オプション

読出または書込みするデバイスの先頭アドレスを指定します。

アドレスは、デバイスの種類により 10 進数または 16 進数で指定します。

指定できるアドレスの最大値は Series オプションの設定値によって変わります。

表 2-5 デバイス一覧

デバイス		デバイスコード	種別	アドレス指定方法	
特殊リレー		SM	ビット	10 進数	
特殊レジスタ		SD	ワード	10 進数	
入力		X	ビット	16 進数	
出力		Y		16 進数	
内部リレー		M		10 進数	
ラッチリレー		L		10 進数	
アナンシェータ		F		10 進数	
エッジリレー		V		10 進数	
リンクリレー		B		16 進数	
データレジスタ		D		ワード	10 進数
リンクレジスタ		W			16 進数
タイマ	接点	TS		ビット	10 進数
	コイル	TC			
	現在値	TN	ワード		
ロングタイマ	接点	LTS	ビット	10 進数	
	コイル	LTC			
	現在値	LTN	ダブルワード		
積算タイマ	接点	SS	ビット	10 進数	
	コイル	SC			
	現在値	SN	ワード		
ロング積算タイマ	接点	LSTS	ビット	10 進数	
	コイル	LSTC			
	現在値	LSTN	ダブルワード		
カウンタ	接点	CS	ビット	10 進数	
	コイル	CC			

	現在値	CN	ワード	
ロングカウンタ	接点	LCS	ビット	10 進数
	コイル	LCC		
	現在値	LCN	ダブル ワード	
リンク特殊リレー		SB	ビット	16 進数
リンク特殊レジスタ		SW	ワード	16 進数
ダイレクトアクセス入力		DX	ビット	16 進数
ダイレクトアクセス出力		DY		16 進数
インデックスレジスタ		Z	ワード	10 進数
ロングインデックスレジスタ		LZ	ダブル ワード	10 進数
ファイルレジスタ		R	ワード	10 進数
		ZR		10 進数
ユニットリフレッシュ用レジスタ		RD	ワード	10 進数

- (例) X0 : “Device=X, Address=0”  
 Y1F : “Device=Y, Address=1F”  
 D50 : “Device=D, Address=50”  
 M350 : “Device=M, Address=350”

三菱電機(株)製シーケンサ CPU のロングタイマ、ロング積算タイマのデータ構造は通常のデバイスと異なり、図 2-1 のようになっています。

ロングタイマ、ロング積算タイマはデバイス 1 点あたり 4 ワードのデータ構造なるため、デバイス点数は 4 点単位で指定してください。

### ■ロングタイム、ロング積算タイムデバイス読み出し時の注意事項

ロングタイム、ロング積算タイムの現在値を4ワード単位で指定することで、接点およびコイルも同時に読み出されます。データ構成は下記となります。

応答データ	内容
1ワード目	現在値が格納されます。
2ワード目	
3ワード目	b0: コイルの値が格納されます。 b1: 接点の値が格納されます。 b2~b15: システムで使用
4ワード目	システムで使用

ロングタイム、ロング積算タイムは、上記のようにデバイス1点あたり4ワードのデータ構成となるため、要求データのデバイス点数は、4点単位で指定してください。

#### 例

ロングタイム2点(LT0, LT1)を読み出す場合、先頭デバイスはLTN0、デバイス点数は8点を指定します。

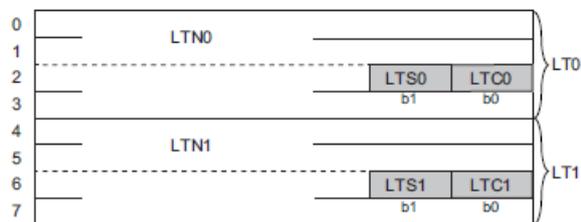


図 2-1 ロングタイム、ロング積算タイムのデータ構造

- (例) LTS : “Device=LTN, Address=0, Elem=4”  
3ワード目の b1 に接点(LTS0)の値が格納されます。
- LTC : “Device=LTN, Address=0, Elem=4”  
3ワード目の b2 にコイル(LTC0)の値が格納されます。
- LTN : “Device=LTN Address=0, Elem=4”  
1, 2ワード目に現在値(LTN0)が格納されています。
- LTN : “Device=LTN Address=0, Elem=8”  
1, 2ワード目に現在値(LTN0), 4, 5ワード目に現在値(LTN1)が格納されます。

### 2.3.2.3. Param オプション

以下に Param オプションの接続パラメータ文字列を示します。ここで角括弧(“[]”)内のパラメータは省略可能を示します。

各要素は設定値を 16 進数で指定桁数分設定する必要があります。不足桁数は 0 埋めしてください。

各項目の内容については SLMP リファレンスマニュアルを参照してください。

“[Param=

[<要求先ネットワーク番号>:<要求先局番>:<要求先ユニット I/O 番号>:<要求先マルチドロップ局番>[:<監視タイマ>]]”

- |                  |   |   |
|------------------|---|---|
| <要求先ネットワーク番号>    | : | アクセス先のネットワーク番号(2 桁).<br>デフォルトは 00 <sub>H</sub> です.      |
| <要求先局番>          | : | アクセス先の局番(2 桁).<br>デフォルトは FF <sub>H</sub> です.            |
| <要求先ユニット I/O 番号> | : | アクセス先のユニット I/O 番号(4 桁).<br>デフォルトは 03FF <sub>H</sub> です. |
| <要求先マルチドロップ局番>   | : | アクセス先のマルチドロップ局番(2 桁).<br>デフォルトは 00 <sub>H</sub> です.     |
| <監視タイマ>          | : | 監視タイマ(4 桁)<br>デフォルトは 0000 <sub>H</sub> (無限待ち)です.        |

(例) “Device=X, Address=0, Param=00:FF:03FF:00:0000” X0 値を取得します

## 指定可能な要求先ユニット I/O 番号一覧

	相手機器のアクセス局	相手機器が指定する要求先ユニット I/O 番号
1	制御系 CPU	03D0 <sub>H</sub>
2	待機系 CPU	03D1 <sub>H</sub>
3	A 系 CPU	03D2 <sub>H</sub>
4	B 系 CPU	03D3 <sub>H</sub>
5	マルチ CPU システム 1 号機	03E0 <sub>H</sub>
6	マルチ CPU システム 2 号機	03E1 <sub>H</sub>
7	マルチ CPU システム 3 号機	03E2 <sub>H</sub>
8	マルチ CPU システム 4 号機	03E3 <sub>H</sub>
9	管理 CPU	03FF <sub>H</sub>
10	マルチドロップ接続上の局の C24 の管理 CPU(最後に経由する CC-Link IE コントローラネットワーク, CC-Link IE フィールドネットワーク, MELSECNET/H, MELSECNET/10 接続局のシーケンサへアクセス時は上記 1~9 を指定)	0000 <sub>H</sub> ~01FF <sub>H</sub>

表 2-6 指定可能な要求先ユニット局番号一覧

	相手機器のアクセス局	相手機器が指定する要求先ユニット局番号
1	下記以外の局	00 <sub>H</sub> (0)
2	マルチドロップ接続上の局(最後に経由する CC-Link IE コントローラネットワーク, CC-Link IE フィールドネットワーク, MELSECNET/H, MELSECNET/10 接続局の Q/L/QnACPU へアクセス時は上記 1 による)	00 <sub>H</sub> ~1F <sub>H</sub> (0~31)

### 2.3.2.4. VT オプション

読み書きするデータ型と一要素あたりの点数を指定します(1点 = 1 bit).

VT オプションの省略時は, Device オプションに指定したデバイスによりデフォルトの値が設定されます. Device オプションにビットデバイスを指定した場合は”BIT”が, ワードデバイスを指定した場合は”I2”が, ダブルワードデバイスを指定した場合は”I4”がデフォルトの値となります.

表 2-7 指定可能な VT オプションの一覧

VT オプション	データ型	点数/要素数	意味 <sup>(2)</sup>
BIT	VT_I2	1 点	ビット単位(1 点単位)で読み書きします. 注)ビットデバイス(X,Y,M 等)のみに指定できます.
BOOL	VT_BOOL	1 点	ビット単位(1 点単位)で読み書きします. 注)ビットデバイス(X,Y,M 等)のみに指定できます.
I1	VT_I1	8 点	8 点単位で読み書きします. 注)Elem オプションで奇数個の要素数を指定し書き込みをした場合, 偶数個の要素として扱い, 追加した 8 点分を 0 埋めし書き込みを行います. 例)変数名: X0      オプション: VT=I1,Elem=1 の場合 値を代入時に X8~XF はすべて 0 になります.
UI1	VT_UI1	8 点	8 点単位で読み書きします. 注)Elem オプションで奇数個の要素数を指定し書き込みをした場合, 偶数個の要素として扱い, 追加した 8 点分を 0 埋めし書き込みを行います. 例)変数名: X0      オプション: VT=UI1,Elem=1 の場合 値を代入時に X8~XF はすべて 0 になります.
I2	VT_I2	16 点	ワード単位(16 点単位)で読み書きします.
UI2	VT_UI2	16 点	16 点単位で読み書きします.
I4	VT_I4	32 点	32 点単位で読み書きします.
UI4	VT_UI4	32 点	32 点単位で読み書きします.
R4	VT_R4	32 点	32 点単位で読み書きします.
R8	VT_R8	64 点	64 点単位で読み書きします.
BSTR	VT_BSTR	8 点	ASCII(1 文字:8 bit)の文字列を読み書きします. 注)Elem オプションで指定された要素数より短い文字列が書き込みされた場合は, 残りの点を 0 埋めます.

<sup>2</sup> VT=BIT, VT=BOOL, VT=BSTR 以外はワード単位で読み書きします.

“[VT=[<VT オプション文字列>]]”

(例) “Device=X, Address=0, VT=I2” X0 から X15 の値をワード値(2Byte)として読み書きします

### 2.3.2.5. Elem オプション

要素数を 10 進数, または 16 進数で指定します. 10 進数で指定する場合はそのまま数値を指定してください. 16 進数で指定する場合は 0x[0-9,A-F]+, &H[0-9,A-F]+, または [0-9,A-F]+H の形式で指定してください. Elem オプションを省略した場合のデフォルトの値は 1 となります<sup>3)</sup>.

“[Elem = [<要素数>]]”

- |  |                             |
|--|-----------------------------|
| (例) “Device=X, Address=0, Elem=5”          | X0 から X4 の値をビット値として取得します    |
| (例) “Device=D, Address=10, Elem=0x10”      | D10 から D25 の値をワード値として取得します  |
| (例) “Device=D, Address=10, Elem=&H10”      | D10 から D25 の値をワード値として取得します  |
| (例) “Device=D, Address=10, Elem=10H”       | D10 から D25 の値をワード値として取得します  |
| (例) “Device=M, Address=100, VT=I2, Elem=2” | M100 から M131 の値をワード単位で取得します |

### 2.3.2.6. Array オプション

Elem オプションで 1 指定し, 尚且つ VT オプションで BSTR 以外指定した場合, 読み込んだ値を配列の形式で取得するかどうかを指定します. True を指定した場合は配列の形式で, False を指定した場合は指定したデータ型の形式となります. Array オプションを省略した場合のデフォルトの値は False となります.

“[Array = [< True or False >]]”

- |   |                          |
|---|--------------------------|
| (例) “Device=X, Address=0, VT=BOOL, Elem=1, Array=True”  | X0 の値を BOOL 型の配列として取得します |
| (例) “Device=X, Address=0, VT=BOOL, Elem=1, Array=False” | X0 の値を BOOL 型として取得します    |

<sup>3)</sup> 要素数の指定可能範囲については, アクセス先の機器のマニュアルを参照してください.

### 2.3.2.7. ランダム読み出し/書き込み

Address オプションにコロン(:) 区切りで複数のアドレス<sup>4)</sup>を指定することで、連続していないアドレスにアクセスすることができます。このとき、Device オプションは 1 種類のみ、複数指定はできません。その他のオプション(VT, Elem, Array)は無視されます。

ビットデバイスは 16 点単位、ワードデバイスは 1 ワード単位、ダブルワードデバイスは 2 ワード単位でアクセスしてください。

表 2-8 ランダムアクセスのデータ型

デバイス種別	読み出し/書き込みデータ型	備考
ビット	VT_I2   VT_ARRAY	<ul style="list-style-type: none"> <li>データ型はデバイスの初期値。</li> <li>Address オプションで指定したアドレス数分を配列で読み出し/書き込みます。</li> </ul>
ワード	VT_I2   VT_ARRAY	
ダブルワード	VT_I4   VT_ARRAY	

- (例) “Device=M, Address=100:200”      M100～M115, M200～M215 にアクセスします。
- (例) “Device=D, Address=0:2:4”      D0, D2, D4 にアクセスします。
- (例) “Device=LTN, Address=1:3:5:7:9”      LTN1, LTN3, LTN5, LTN7, LTN9 にアクセスします。

<sup>4</sup> 各デバイスのアドレス番号の範囲、および指定可能なアドレス数(アクセス点数)については、アクセス先の機器のマニュアルを参照してください。

### 2.3.2.8. サンプルプログラム

AddVariableのサンプルプログラムを以下に示します。

例) ビットデバイスMの配列(要素数10)に値を設定する。

```
HRESULT hr = S_OK;
ICaoEngine* pEng = NULL;
ICaoWorkspaces *pWss = NULL;
ICaoWorkspace *pWs = NULL;
ICaoController *pCtrl = NULL;
ICaoVariable *pVar = NULL;
CComVariant vntGet;

// CaoEngine の生成
hr = CoCreateInstance(CLSID_CaoEngine,
                    NULL,
                    CLSCTX_LOCAL_SERVER,
                    IID_ICaoEngine,
                    (void **)&pEng);

if (FAILED(hr)) {
    goto EndProc;
}

// CaoWorkspace コレクションの取得
hr = pEng->get_Workspaces(&pWss);
if (FAILED(hr)) {
    goto EndProc;
}

// CaoWorkspace の取得
hr = pWss->Item(CComVariant(0L), &pWs);
if (FAILED(hr)) {
    goto EndProc;
}

// CaoController の生成
hr = pWs->AddController(CComBSTR(L"SLMP_TEST"),
                      CComBSTR(L"CaoProv.SLMP"),
                      CComBSTR(L""),
                      CComBSTR(L"Conn=ETH:169.254.225.206:61442"),
                      &pCtrl);

if (FAILED(hr)) {
    goto EndProc;
}

// 変数の生成
hr = pCtrl->AddVariable(CComBSTR(L"aaa"), CComBSTR(L"Device=M, Address=16, Elem=10"), &pVar);
if (FAILED(hr)) {
    goto EndProc;
}
```

```
// 値の設定, 取得
CComVariant vntPut
vntPut.vt = (VT_I2 | VT_ARRAY);
vntPut.parray = SafeArrayCreateVector(VT_I2, 0, 10);
pVar->put_Value(vntPut);
pVar->get_Value(&vntGet);
```

EndProc:

```
if (pVar) pVar->Release();
if (pCtrl) pCtrl->Release();
if (pWs) pWs->Release();
if (pWss) pWss->Release();
if (pEng) pEng->Release();
```

## 2.4. エラーコード

SLMP プロバイダでは、以下の固有エラーコードが定義されています。また、ORiN2 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 2-9 固有エラーコード

エラー名	エラー番号	説明
ユニットエラー	0x8010xxxx	機器でエラーが発生した場合は、機器のエラーコードを xxxx の箇所に入れて返します。エラーコードの内容については機器のリファレンスを参照してください。
応答異常 (フォーマットエラー)	0x80110000	応答パケットが想定外の異常なフォーマットであった場合に返ります。
デバイス名不正	0x80110001	存在しないデバイス名を指定した場合等に発生します。
アドレスフォーマット異常	0x80110002	アドレスの指定が正しくない場合に返ります。 10 進デバイスのアドレスに 16 進指定をしている場合等に発生します。
アドレス範囲外	0x80110003	開始アドレスを指定不可のアドレスを指定した場合に発生します。 Series オプション(2.3.1.2)を使用することで回避できることもあります。
ビット単位不許可アクセス	0x80110004	ワードデバイスまたはダブルワードデバイスでビット単位アクセスを指定した場合に発生します。 デバイス, VT オプションをご確認ください。

### 3. コマンドリファレンス

本章では CaoController::Execute メソッドの各コマンドについて解説します。

表 3-1 CaoController::Execute コマンド一覧

コマンド	機能	内容
Raw	生データ送受信	生データ(バイナリコード)の送受信を行います。

#### 3.1. Execute コマンド

##### 3.1.1. CaoController::Execute( “Raw” ) コマンド

生データの送受信を行います。

**書式** Raw ( <vntParam>, <pVal> )

<vntParam> : [in] 要求伝文 (VT\_UI1 | VT\_ARRAY)  
 <pVal> : [out] 応答伝文 (VT\_UI1 | VT\_ARRAY)  
 戻り値 : HRESULT

S_OK	正常終了
S_OK 以外	異常終了

備考 : 引数で指定した要求伝文(バイナリコード)を機器に送信します。  
 応答伝文をバイナリコードで返します。  
 このとき送受信で要求伝文/応答伝文の内容は一切加工しません。

### 3.1.2. サンプルプログラム

Executeコマンド(Raw)のサンプルプログラムを以下に示します。

例) ①Y100を読み出します。

②X110に値 (ON) を書き込みます。

```
HRESULT func1()
{
    HRESULT hr = S_OK;
    ICaoEngine* pEng = NULL;
    ICaoWorkspaces *pWss = NULL;
    ICaoWorkspace *pWs = NULL;
    ICaoController *pCtrl = NULL;
    ICaoVariable *pVar = NULL;
    CComVariant vntParam;
    CComVariant vntRet;
    SAFEARRAYBOUND boundR = { 21, 0 };
    BYTE bSendRData[21] = { 0x50, 0x00, 0x00, 0xFF, 0xFF, 0x03, 0x00, 0x0C, 0x00, 0x00,
                          0x00, 0x01, 0x04, 0x01, 0x00, 0x00, 0x01, 0x00, 0x9D, 0x01,
                          0x00 };
    SAFEARRAYBOUND boundW = { 22, 0 };
    BYTE bSendWData[22] = { 0x50, 0x00, 0x00, 0xFF, 0xFF, 0x03, 0x00, 0x0D, 0x00, 0x00,
                          0x00, 0x01, 0x14, 0x01, 0x00, 0x10, 0x01, 0x00, 0x9C, 0x01,
                          0x00, 0x10 };

    // CaoEngine の生成
    hr = CoCreateInstance(CLSID_CaoEngine,
                        NULL,
                        CLSCTX_LOCAL_SERVER,
                        IID_ICaoEngine,
                        (void **)&pEng);

    if (FAILED(hr)) {
        goto EndProc;
    }

    // CaoWorkspace コレクションの取得
    hr = pEng->get_Workspaces(&pWss);
    if (FAILED(hr)) {
        goto EndProc;
    }

    // CaoWorkspace の取得
    hr = pWss->Item(CComVariant(0L), &pWs);
    if (FAILED(hr)) {
        goto EndProc;
    }
}
```

```
// CaoController の生成
hr = pWs->AddController(CComBSTR(L"SLMP_TEST"),
                        CComBSTR(L"CaoProv.SLMP"),
                        CComBSTR(L""),
                        CComBSTR(L"Conn=ETH:169.254.225.206:61442"),
                        &pCtrl);

if (FAILED(hr)) {
    goto EndProc;
}

////////////////////////////////////
// ①Y100 を読み出します.
////////////////////////////////////
SAFEARRAY* pSa = SafeArrayCreate(VT_UI1, 1, &boundR);
// SAFEARRAY にアクセスする.
BYTE* pbyRead;

SafeArrayAccessData(pSa, (void*)&pbyRead);
for(int i = 0; i < 21; i++){
    pbyRead[i] = bSendRData[i];
}

SafeArrayUnaccessData(pSa);

vntParam.Clear();
vntParam.parray = pSa;
vntParam.vt = VT_ARRAY | VT_UI1;

vntRet.Clear();
hr = pCtrl->Execute(CComBSTR("Raw"), vntParam, &vntRet);
if (FAILED(hr)) {
    goto EndProc;
}

////////////////////////////////////
// ②X110 に値(ON)を書き込みます.
////////////////////////////////////
// SAFEARRAY の実体が存在しないので作成する.
pSa = SafeArrayCreate(VT_UI1, 1, &boundW);
// SAFEARRAY にアクセスする.
BYTE* pbyWrite;

SafeArrayAccessData(pSa, (void*)&pbyWrite);
for(int i = 0; i < 22; i++){
    pbyWrite[i] = bSendWData[i];
}
SafeArrayUnaccessData(pSa);
```

```
vntParam.Clear();
vntParam.parray = pSa;
vntParam.vt = VT_ARRAY | VT_UI1;

vntRet.Clear();
hr = pCtrl->Execute(CComBSTR("Raw"), vntParam, &vntRet);
if (FAILED(hr)) {
    goto EndProc;
}
std::cout << "Raw(Write) = 0x" << std::hex << std::setw(8) << std::setfill('0') << hr << "\n";

EndProc:

// オブジェクトの解放
if (pVar) pVar->Release();
if (pCtrl) pCtrl->Release();
if (pWs) pWs->Release();
if (pWss) pWss->Release();
if (pEng) pEng->Release();
return hr;
}
```

## 4. 付録

### 4.1. 他局アクセス

三菱電機製シーケンサ Q シリーズは他局アクセス(親 PLC 経由で子 PLC の値を取得/書込)を行うことができます。本項では他局アクセスための設定手順を **図 4-1** **図 4-1** の構成を例に説明します。なお、親 PLC-子 PLC は汎用 Ethernet で接続していることを想定しています。**接続は必ず Ethernet ユニットを使用してください。内蔵 Ethernet ユニット経由でのアクセスはできません。** Ethernet ユニット間はハブまたはクロスケーブルを用いて接続してください。

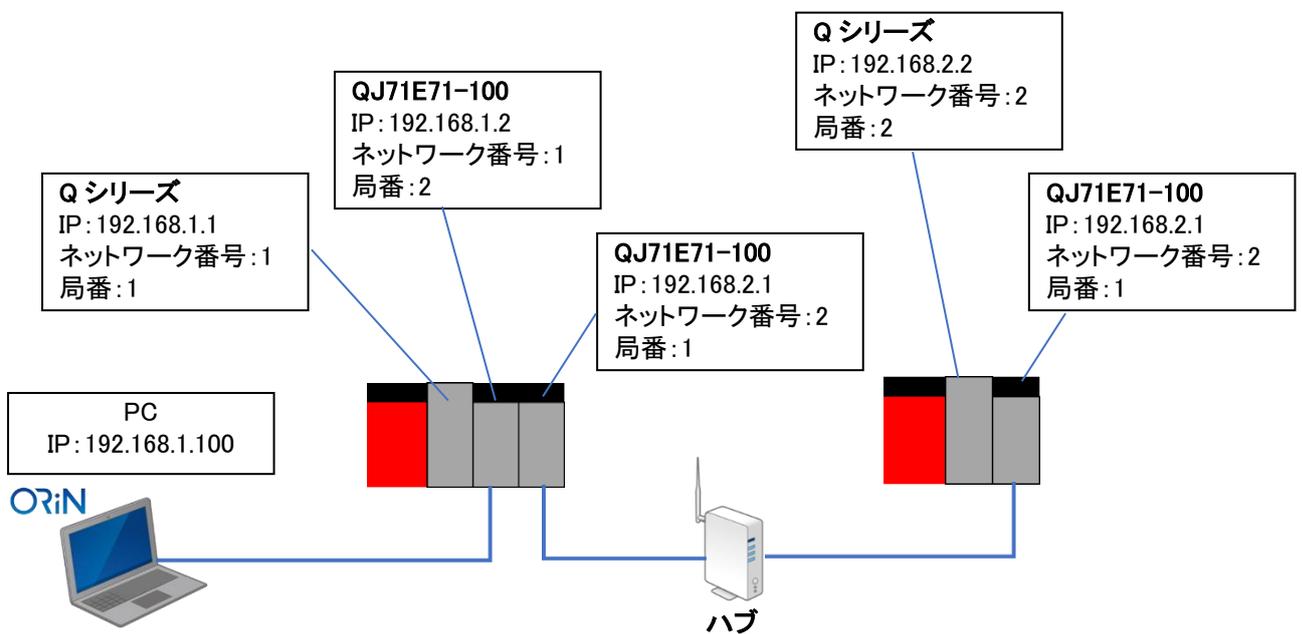


図 4-1 構成図

## 4.1.1. 親 PLC の設定

### 4.1.1.1. 内蔵 Ethernet ユニットの設定

図 4-2 の①～④に内蔵 Ethernet ユニットの設定手順を示します。

- ① PLC 接続後、ナビゲーションツリーメニューの「PC パラメータ」をクリックします。
- ② Q パラメータ設定画面で IP アドレスとサブネットマスクを入力します。
- ③ MELSOFT 接続拡張設定を開きます。
- ④ 「MELSOFT 接続拡張設定:使用する」を選択し、ネットワーク No. と局番を入力してください。

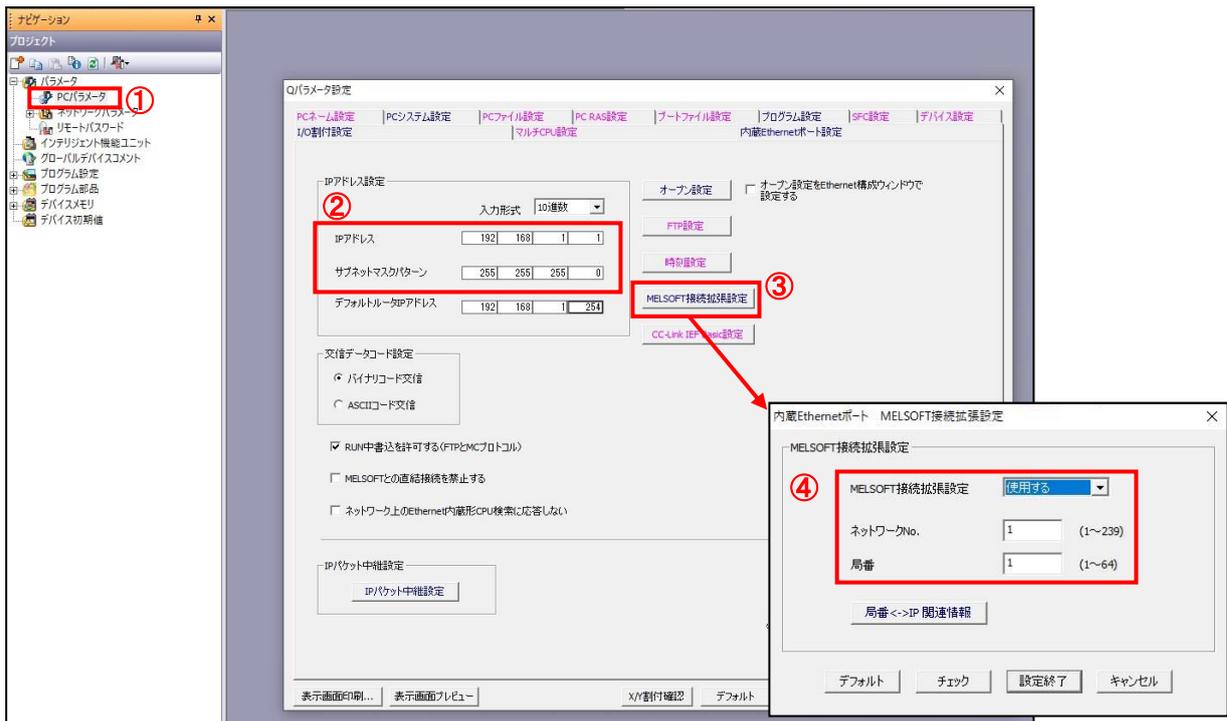


図 4-2 内蔵 Ethernet ユニットパラメータ設定(親 PLC)

#### 4.1.1.2. 拡張 Ethernet ユニット 1 の設定

図 4-3 の①～④に拡張 Ethernet ユニット 1 の設定手順を示します。

- ① ナビゲーションツリーメニューの「Ethernet/CC IE/MELSECNET」をクリックします。
- ② ユニット 1 の設定で「ネットワーク種別:Ethernet」を選択し、ネットワーク No. と局番を入力します。
- ③ 動作設定を開きます。
- ④ IP アドレスを入力してください。

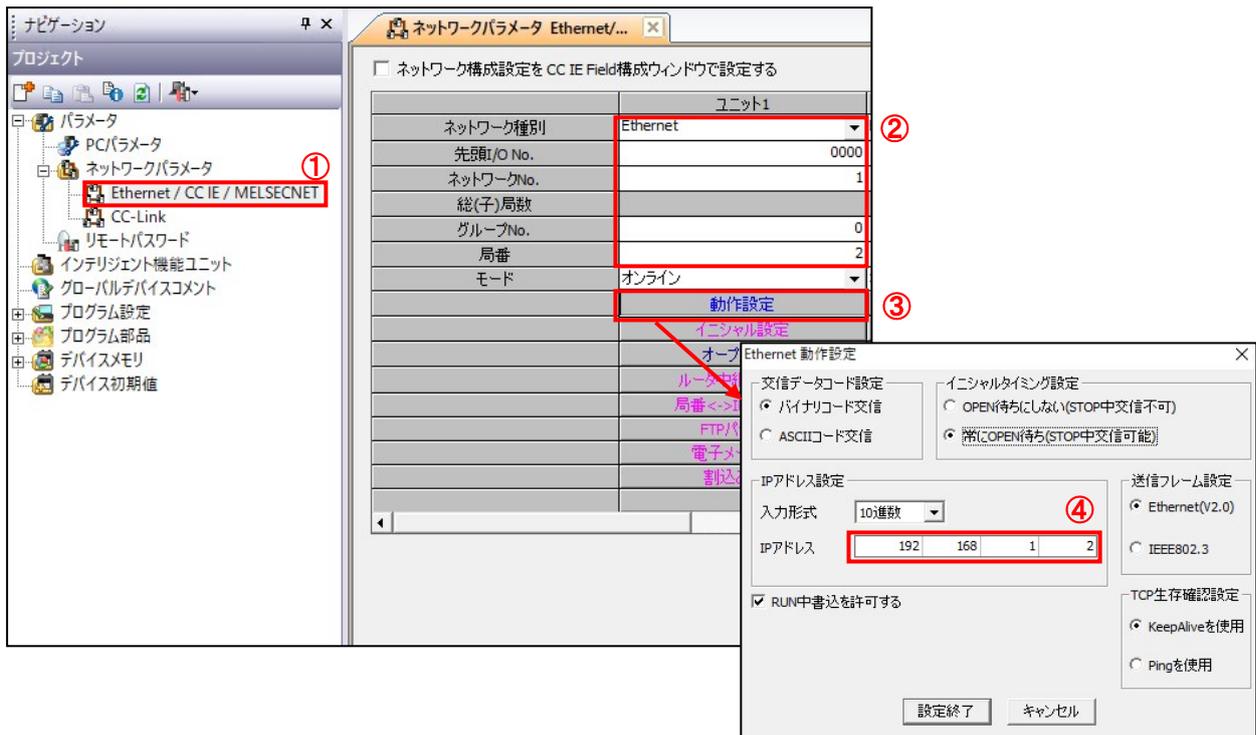


図 4-3 拡張 Ethernet ユニット 1 パラメータ設定

## 4.1.1.3. 拡張 Ethernet ユニット 2 の設定

図 4-4 の①～⑤に拡張 Ethernet ユニット 2 の設定手順を示します。設定画面を表示する手順は 4.1.1.2 の手順①と共通です。

- ① ユニット 2 の設定で「ネットワーク種別:Ethernet」を選択し、ネットワーク No. と局番を入力します。
- ② 動作設定を開きます。
- ③ IP アドレスを入力してください。
- ④ 局番<->IP 関連情報を開きます。
- ⑤ 「局番<->IP 関連情報設定方式:テーブル変換方式」を選択し、子 PLC の拡張 Ethernet ユニットのネットワーク No. と局番と IP アドレスを入力してください。

	ユニット1	ユニット2
ネットワーク種別	Ethernet	Ethernet
先頭I/O No.	0000	0100
ネットワークNo.	1	2
総(子)局数		
グループNo.	0	0
局番	2	1
モード	オンライン	オンライン
	動作設定	動作設定
	イニシャル設定	イニシャル設定
	オープン設定	オープン設定
	ルータ中継パラメータ	ルータ中継パラメータ
	局番<->IP関連情報	局番<->IP関連情報
	FTPパラメータ	
	電子メール設定	
	割込み設定	

No.	ネットワークNo.	局番	IPアドレス
1	2	3	192.168.2.3
2			
3			
4			
5			
6			
7			
8			
9			
10			

図 4-4 拡張 Ethernet ユニット 2 パラメータ設定

## 4.1.2. 子 PLC の設定

### 4.1.2.1. 内蔵 Ethernet ユニットの設定

図 4-5 の①～④に内蔵 Ethernet ユニットの設定手順を示します。

- ① PLC 接続後、ナビゲーションツリーメニューの「PC パラメータ」をクリックします。
- ② Q パラメータ設定画面で IP アドレスとサブネットマスクを入力します。
- ③ MELSOFT 接続拡張設定を開きます。
- ④ 「MELSOFT 接続拡張設定:使用する」を選択し、ネットワーク No. と局番を入力してください。

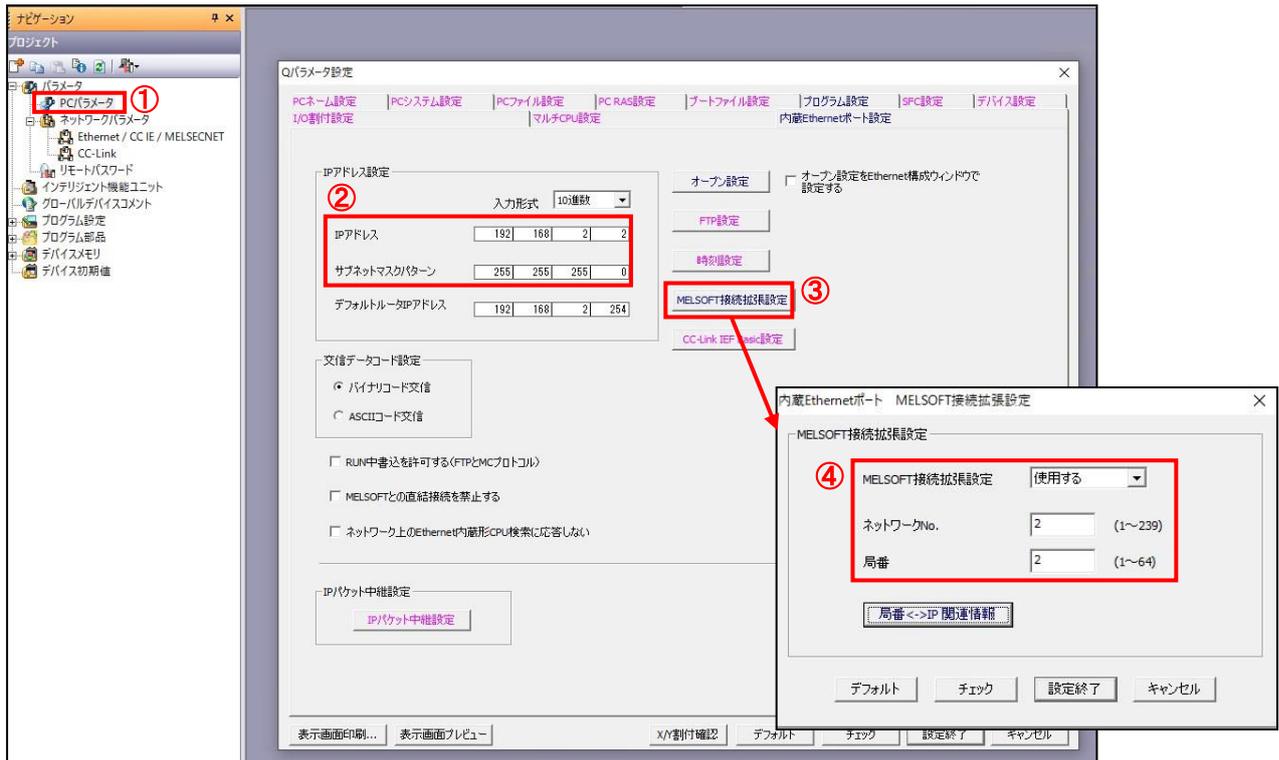


図 4-5 内蔵 Ethernet ユニットの設定(子 PLC)

#### 4.1.2.2. 拡張 Ethernet ユニットの設定

図 4-6 の①～④に拡張 Ethernet ユニット 1 の設定手順を示します。

- ① ナビゲーションツリーメニューの「Ethernet/CC IE/MELSECNET」をクリックします。
- ② ユニット 1 の設定で「ネットワーク種別:Ethernet」を選択し、ネットワーク No. と局番を入力します。
- ③ 動作設定を開きます。
- ④ IP アドレスを入力してください。

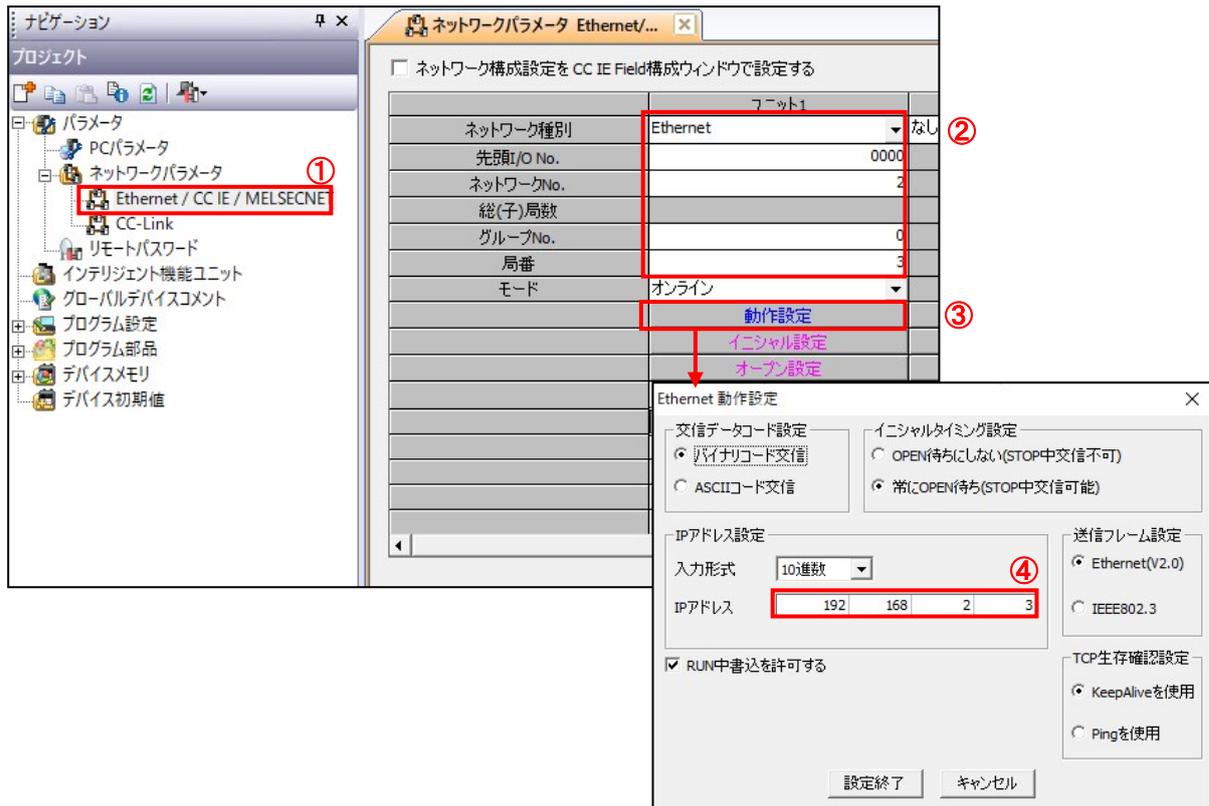


図 4-6 拡張 Ethernet ユニットの設定(子 PLC)

### 4.1.3. 接続パラメータ設定

他局アクセスするためには変数オブジェクト作成時に以下のパラメータを設定してください。

"Param=

<要求先ネットワーク番号>:<要求先局番>:<要求先ユニット I/O 番号>:<要求先マルチドロップ局番>[:<監視タイマ>]"

<要求先ネットワーク番号>	:子 PLC のネットワーク番号(16 進数).
<要求先局番>	:子 PLC の局番(16 進数).
<要求先ユニット I/O 番号>	: 03FF(管理 CPU).
<要求先ユニット 局番号>	: 00(非マルチドロップ接続).
<監視タイマ>	:任意.

(例) " Device=X, Address=0, Param=00:FF:03FF:00:0000"

ネットワーク番号 2, 局番 3 の PLC の X0 値を取得します