OMRON プログラマブルコントローラ
CJ シリーズ用上位リンクプロバイダ

Version 1.0.1

ユーザーズ ガイド

August 02, 2022

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2019-07-31	初版.
1.0.1	2022-08-02	チャネル指定時の値書込み処理の修正

【通信確認機器】

機種	バージョン	注意事項
CJ2H-CPU64-EIP	Ver1.5	

目次

1. はじめに 4

2. プロバイダの概要 5

2.1. 概要 5
2.2. メソッド・プロパティ 6
2.2.1. CaoWorkspace::AddController メソッド 6
2.2.1.1. Conn オプション 7
2.2.1.2. UnitNo オプション 7
2.2.1.3. FinsParam オプション 8
2.2.2. CaoController::AddVariable メソッド 9
2.2.2.1. 変数名によるアクセス対象のエリア種別/アドレス指定 10
2.2.2.2. Mode オプション 12
2.2.2.3. VT オプション 12
2.2.2.4. アドレスのワイルドカード指定 13
2.2.2.5. 変数名への任意文字列付与 14
2.2.3. CaoVariable:put_Value プロパティ 15
2.2.4. CaoVariable:get_Value プロパティ 15
2.2.5. CaoVariable:put_ID プロパティ 15
2.2.6. CaoVariable:get_ID プロパティ 15
2.3. 変数一覧 16
2.3.1. CaoController クラス 16
2.4. エラーコード 17

付録 A. 変数名設定例 18

付録 B. 変数書き込み時の PLC(CJ シリーズ)内のメモリ割り付け 19

付録 C. チャネルアドレスの VT=BIT, VT=BOOL, VT=UI1, VT=I1 指定時の書き込み処理について 27

1. はじめに

本書は，OMRON 製 PLC(CJ シリーズ)に対しデータの書込み/読出しを行う 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(CaoProvOMRONCJHostLink.dll)を CJHostLink プロバイダと呼びます．

第 2 章に CJHostLink プロバイダの概要，変数の詳細を記載しています．

CJHostLink プロバイダで実装している通信コマンドの対応状況及びデータ列については，通信先となる PLC(CJ シリーズ)に依存します．通信につきましては FINS コマンドのシリアル通信(上位リンク)を使用しています．詳細については OMRON の“ sbca-304r-30_cs1_cj1_cp1_com_cmd.pdf ”並びに“sbca-350l-22_cj2h-cpu6-eip_cj2m-cpu.pdf”を参照してください．

また，シリアルポートへの接続は指定のケーブルを使用してください．ケーブルの詳細については OMRON の“sbca-349r-1_cj2h-cpu6-eip_cj2m-cpu.pdf”を参照してください．

2. プロバイダの概要

2.1. 概要

CJHostLink プロバイダは，OMRON 製 PLC(CJ シリーズ)に対し上位リンク(シリアル接続)で FINS コマンドを用いてデータの書込み/読出しを行う 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．CJHostLink 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 CJHostLink プロバイダ

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

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

CJ プロバイダは AddController 時に通信用の接続パラメータを参照し，通信の接続を行います．(TCP クライアントとして動作します)

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.OMRON.CJHostLink"
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([1])	説明
Conn=<接続パラメータ>	必須．通信形態と接続パラメータ． (参照 2.2.1.1)
UnitNo[=<上位リンク用ユニット No>]	上位リンク用ユニット No．(0～31) (参照 2.2.1.2)
FinsParam=<DNA>:<DA1>:<DA2> [:<GCT>[:<SNA>[:<SA1>[:<SA2>:[<SID>]]]]]	必須(一部省略可)．FINS コマンドのヘッダ構成パラメータ． (参照 2.2.1.3)
Timeout[=<タイムアウト時間>]	送受信時のタイムアウト時間．(ミリ秒) （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)
PutOpt:[=<書込み時動作>]	チャネルアドレスへ BIT, BOOL, UI1, I1 で書き込む際の，データの扱い方法を指定します． (参照 2.4.付録 C)

[1] 角括弧("[ ]")内は省略可能を示します．また，各パラメータの解説中の下線部はオプションを指定しなかったときのデフォルト値になります．

2.2.1.1. 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[1].

シリアルデバイス

"Conn=com:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>]]"

<COM Port>	:	COM ポート番号. '1'-COM1, '2'-COM2, …
<BaudRate>	:	通信速度.
		300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
<Parity>	:	パリティ. 'N'-NONE, 'E'-EVEN, 'O'-ODD.
<DataBits>	:	データビット数. '7'-7bit, '8'-8bit.
<StopBits>	:	ストップビット数. '1'-1bit, '2'-2bit.

(例 1) "com:1" 通信ポート COM1 (, 9600bps, Even, 7bits, 2bit)
(例 2) "com:2:19200" 通信ポート COM2, 19200bps (,Even, 7bits, 2bit)
(例 3) "com:3:38400:N:8:2" 通信ポート COM3, 38400bps, None, 8bits, 2bit

2.2.1.2. UnitNo オプション

上位コンピュータに接続されている相手先の CPU ユニットを指定するための番号です. CPU ユニットと接続されている場合は, PLC システム設定で設定した号機 No.を設定します.
接続先がシリアルコミュニケーションボード/ユニットの場合は, システム設定(割付 DM エリア)で設定した号機 No.を設定します(図 2-1).
(デフォルト:0)

・ UnitNo の設定(CX-Programmer を使用)

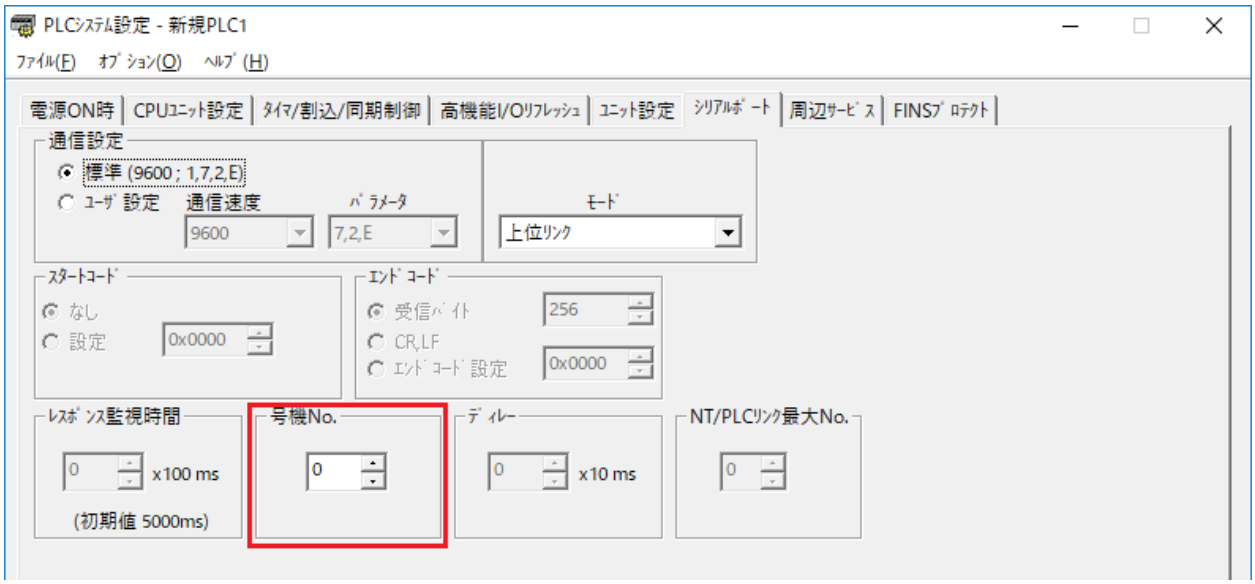


図 2-1 上位リンク用ユニット番号の設定

2.2.1.3. FinsParam オプション

以下に FinsParam オプションの接続パラメータ文字列を示します [1].

"FinsParam=<DNA>:<DA1>:<DA2>[:<GCT>[:<SNA>[:<SA1>[:<SA2>:[<SID>]]]]]"

<DNA> : 相手先ネットワークアドレス(Destination network address)
0 : 自ネットワーク
1～127 :相手先ネットワークアドレス

<DA1> : 相手先ノードアドレス(Destination node number)
0 :自 PLC 内通信
1～32 : Controller Link ネットワークの場合
1～254 : Ethernet の場合
255 : 一斉同報データ送信

<DA2> : 相手先号機アドレス(Destination unit address)
0 : CPU ユニット
254 : 該当 Controller Link ユニット/Ethernet ユニット
16～31 : CPU 高機能ユニット
225 : INNER ボード

<GCT> : 許容ブリッジ通過数(Gateway Count)
2, 7

<SNA> : 発信元ネットワークアドレス(Source network address)
0 : 自ネットワーク
1～127 : 相手先ネットワークアドレス

<SA1> : 発信元ノードアドレス(Source node number)
0 : 自 PLC 内通信
1～32 : Controller Link ネットワークの場合
1～254 : Ethernet の場合

<SA2> : 発信元号機アドレス(Source unit address)
0 : CPU ユニット
16～31 : CPU 高機能ユニット

<SID> : サービス ID(Service ID)
0～255 : 任意の値

2.2.2. CaoController::AddVariable メソッド

CaoController クラスの AddVariable メソッドは, PLC(CJ シリーズ)に対しデータの書込み/読出しを行うための変数オブジェクトを作成するためのメソッドです.

本メソッドでは変数名に 2.2.2.1 に記す書式で指定することで PLC(CJ シリーズ)内のアクセス対象とする IO メモリのエリア種別/アドレスを決定します.

また, オプションでデータの型変換や書込み/読出しするデータ数の指定が可能です. [2]

書式　AddVariable(<bstrVariableName:VT_BSTR>[,<bstrOption:VT_BSTR>])

<bstrVariableName> : [in] 変数名

<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 2-3 CaoController::AddVariable のオプション文字列

オプション	説明
Mode[=<アクセスモード>]	タイマ, カウンタ, タスクフラグにアクセスする際のモードを指定. (参照 2.2.2.2)
VT[=<変数型>]	Put/Get するデータ型を指定. (参照 2.2.2.3)
Elem[=<要素数>]	Put/Get するデータの要素数を指定. データ型が VT_BSTR の場合は文字列のバイト数指定になる. 要素数は 10 進数での指定の他, 16 進数での指定も可能. (0x0A, &h0A, 0AH) (デフォルト:1)
Array[=< True or False >]	Put/Get するデータが 1 要素の場合に配列として扱うかを指定. (デフォルト:False)
ID[=<ID 初期値>]	変数の ID 初期値を指定. (10 進数) (デフォルト:0)

[2] データの読み書きに伴い発生する通信のパケットサイズが大きくなると, FINS コマンドの制限で通信が失敗する場合があります. 1 度に読み書きするデータの要素数を減らすことで回避できます.

2.2.2.1. 変数名によるアクセス対象のエリア種別/アドレス指定

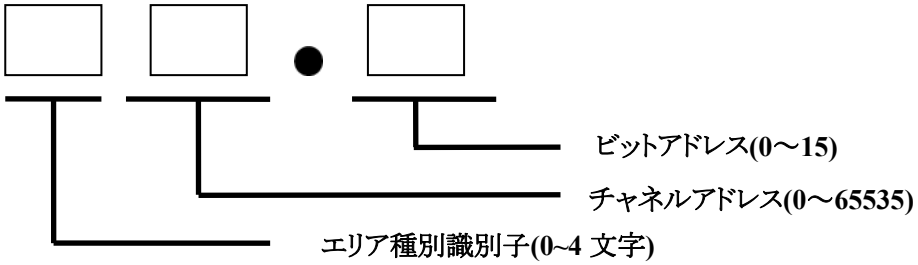
エリア種別識別子(アルファベット 0～4 文字), チャネルアドレス(10 進数の数値 0～65535), ビットアドレス(10 進数の数値 0～15)で構成されます.

アクセス対象 IO メモリのデータ種類がビットの場合はチャネルアドレスの後に“.”(ドット)を挟んでビットアドレスを指定します. チャネル(ワード)の場合はビットアドレスを指定しません.

タイマ, カウンタ, タスクフラグに関してはチャネル(ワード)アドレス指定書式のみを使用し, データ種類は後述の Mode オプションによって指定します. (参照 2.2.2.2)

メモリ種別及びアドレスを指定するための書式を以下に示します.

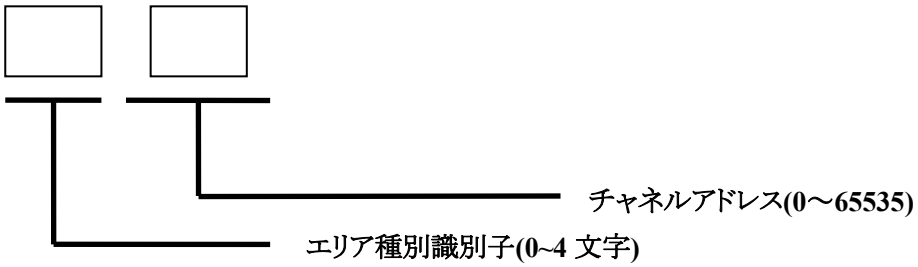
ビットアドレス指定書式



例） チャネル I/O(CIO)のチャネルアドレス 0x0010, ビット 12 にアクセス

⇒ **16.12** 又は **0016.12** （桁の 0 埋めの有無に関わらず同じアドレスが指定されたものとして扱います）

チャネル(ワード)アドレス指定書式



例） 拡張データメモリ EM バンク 4 のチャネルアドレス 0x00FF にアクセス

⇒ **E4_255** 又は **E4_0255**

IO メモリエリア種別毎の識別子一覧を以下に示します.

表 2-4 エリア種別識別子一覧

エリア種別		識別子
チャネル I/O	CIO	なし
内部補助リレー	WR	W
保持リレー	HR	H
特殊補助リレー	AR	A
タイマ	TIM	T
カウンタ	CNT	C
データメモリ	DM	D
拡張データメモリ	EM バンク 0	E0_
	EM バンク 1	E1_
	EM バンク 2	E2_
	EM バンク 3	E3_
	EM バンク 4	E4_
	EM バンク 5	E5_
	EM バンク 6	E6_
	EM バンク 7	E7_
	EM バンク 8	E8_
	EM バンク 9	E9_
	EM バンク A	EA_
	EM バンク B	EB_
	EM バンク C	EC_
	EM バンク D	ED_
	EM バンク E	EE_
	EM バンク F	EF_
	EM バンク 10	E10_
	EM バンク 11	E11_
	EM バンク 12	E12_
	EM バンク 13	E13_
	EM バンク 14	E14_
	EM バンク 15	E15_
	EM バンク 16	E16_

拡張データメモリ	EM バンク 17	E17_
	EM バンク 18	E18_
タスクフラグ	TK	TK

2.2.2.2. Mode オプション

タイマ, カウンタ, タスクフラグを対象とする際にデータ種類を指定するために使用します.

（デフォルト:NORMAL）

NORMAL : データ種別をビットとして扱う.
タイマ, カウンタ, タスクフラグ以外の場合は本設定を無視する.

CURRENT : データ種別を現在値として扱う.
タイマ, カウンタ時のみ指定可能.

STATUS : データ種別をステータスとして扱う.
タスクフラグ時のみ指定可能.

2.2.2.3. VT オプション

Put/Get するデータ型を指定します.

指定可能なデータ型の一覧を以下に示します.

(デフォルト:ビットアドレス指定時=BIT, チャネルアドレス指定時=I2)

表 2-5 VT オプションで指定可能なデータ型一覧

VT	データ型	使用可能種類	説明
BIT	VT_UI1	ビット/チャネル	データを 0/1 の 2 値に変換して書込み/読出し Val==0:0, Val≠0:1
BOOL	VT_BOOL	ビット/チャネル	データを 0/1 の 2 値に変換して書込み/読出し Val==VARIANT_FALSE:0, Val= VARIANT_TRUE:1
BSTR	VT_BSTR	チャネル	BSTR を ASCII として書込み/読出し
I1	VT_I1	チャネル	1 バイトデータとして書込み/読出し
I2	VT_I2	チャネル	2 バイトデータとして書込み/読出し
I4	VT_I4	チャネル	4 バイトデータとして書込み/読出し
I8	VT_I8	チャネル	8 バイトデータとして書込み/読出し
UI1	VT_UI1	チャネル	1 バイトデータとして書込み/読出し
UI2	VT_UI2	チャネル	2 バイトデータとして書込み/読出し
UI4	VT_UI4	チャネル	4 バイトデータとして書込み/読出し
UI8	VT_UI8	チャネル	8 バイトデータとして書込み/読出し
R4	VT_R4	チャネル	4 バイトデータとして書込み/読出し
R8	VT_R8	チャネル	8 バイトデータとして書込み/読出し

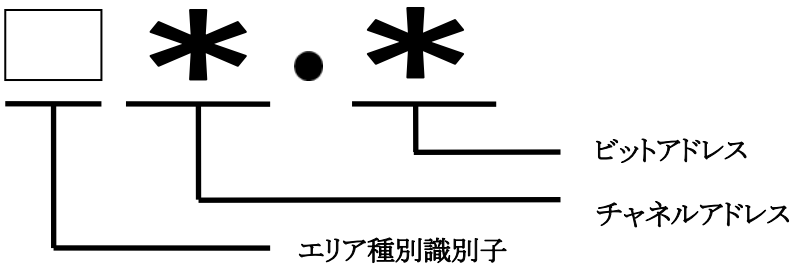
2.2.2.4. アドレスのワイルドカード指定

変数を登録する際にアクセス先のアドレスを固定化せず，後から動的に変更できるようアドレス部分を“*”（アスタリスク）でワイルドカード指定することが可能です．

アドレスは put/get 時の変数の ID で決定します． (参照 2.2.5)

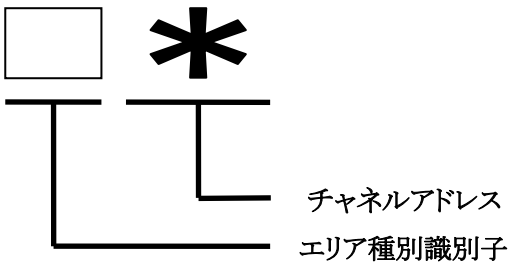
アドレスをワイルドカード指定するための書式を以下に示します．

ビットアドレス指定書式



- 内部補助リレー(WR)にビットアクセス出来るようアドレスをワイルドカード指定
  W*.*

チャネル（ワード）アドレス指定書式



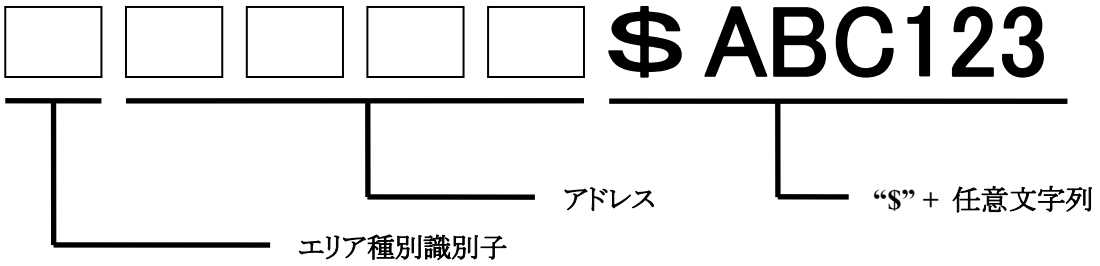
- 保持リレー(HR)にチャネルアクセス出来るようアドレスをワイルドカード指定
  H*

2.2.2.5. 変数名への任意文字列付与

アクセスするエリア種別やアドレスを変えずに複数変数を登録(オプションのみ変更したい場合等に有用)するために“$”(ダラー)を挟んで任意の文字列を付与することが可能です.

任意文字列を付与するための書式を以下に示します.

ビット/チャネル(ワード)アドレス共通指定書式



- カウンタ(CNT)のアドレス 0x00A0 のアップフラグと現在値を取得するために変数を 2 つ登録 **C0160$flag** と **C0160$val** (”$”以降は解析されず別変数として登録できます)

2.2.3. CaoVariable:put_Value プロパティ

引数渡しされた値をオプション指定に従い変換した後，変数名で指定したメモリ領域に対し書込みをするFINS コマンド（コマンドコード：01 02）を送出します．

2.2.4. CaoVariable:get_Value プロパティ

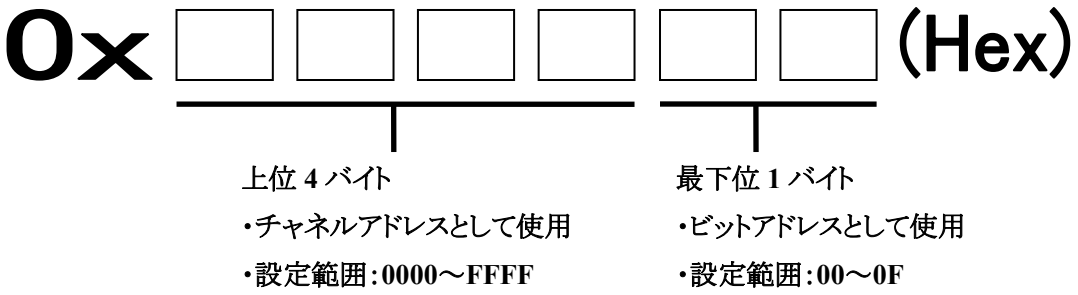
変数名で指定したメモリ領域からオプションで指定されたサイズになるように読出しするFINSコマンド（コマンドコード：01 01）を送出します．読み出した結果はオプションで指定された型に変換し返されます．

2.2.5. CaoVariable:put_ID プロパティ

変数の ID を設定します．

この ID は変数名によるアドレスをワイルドカード指定した際に参照されます．(参照 2.2.2.4)

ワイルドカード指定されていない場合は単純なIDとして機能します．ワイルドカード指定された場合，ID 値は以下の意味と設定範囲を持ちます．



※ チャネルアドレスのみ指定したい場合は最下位 1 バイトを 0x00 として ID をセット

2.2.6. CaoVariable:get_ID プロパティ

現在設定されている変数の ID を取得します．

2.3. 変数一覧

2.3.1. CaoController クラス

表 2-6 CaoController クラス ユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
任意	変数型依存	PLC(CJ シリーズ)内のアクセス対象とする IO メモリのエリア種別/アドレスを指定する. (参照 2.2.2)	○	○

表 2-7 CaoController クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@MAKER_NAME	VT_BSTR	メーカ名=”OMRON”を返す.	○	－
@VERSION	VT_BSTR	バージョン情報.	○	－
@LAST_ERROR	VT_VARIANT | VT_ARRAY	直前の通信異常応答詳細情報. [0] VT_UI2 ： 発生コマンドコード [1] VT_BOOL: 中継異常 [2] VT_BOOL: 本体停止異常 [3] VT_BOOL: 本体継続異常 [4] VT_UI1 ： メインレスポンスコード [5] VT_UI1 ： サブレスポンスコード [6] VT_UI1 ： 異常発生ネットワークアドレス [7] VT_UI1 ： 異常発生ノードアドレス	○	－

2.4. エラーコード

CJ 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．

0x801001xx のエラーが返された場合，システム変数“@LAST_ERROR”を読み出すことでエラー情報を取得することが可能です．

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 2-8 固有エラーコード

エラー名	エラー番号	説明
上位リンク応答異常 (フォーマットエラー)	0x80100000	上位リンク通信の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．
受信データ欠落	0x80100001	受信データが欠落しており，解析ができなかった場合に返されます．
異なるコマンド応答	0x80100002	送信した FINS コマンドと異なるコマンド応答を受けた場合に返されます．
中継異常発生	0x80100100	通信経路に異常があり，中継異常の応答があった場合に返されます．
エラー応答	0x80100101	接続先からエラー応答を受けた場合に返されます．

付録A. 変数名設定例

以下にいくつかの変数名の設定例を紹介します.

例1） チャネル I/O(CIO)のチャネルアドレス 0x0010, ビット 12 にアクセス

16.12 又は **0016.12** （桁の 0 埋めの有無に関わらず同じアドレスが指定されたものとして扱います）

例2） 拡張データメモリ EM バンク 4 のチャネルアドレス 0x00FF にアクセス

E4_255 又は **E4_0255**

例3） 内部補助リレー(WR)にビットアクセス出来るようアドレスをワイルドカード指定

W*.*

アドレス 0x0010, ビット 12 へアクセスするよう ID 設定

ID=4108 (0x001012)

例4） 保持リレー(HR)にチャネルアクセス出来るようアドレスをワイルドカード指定

H*

アドレス 0x0010 へアクセスするよう ID 設定

ID=4096 (0x001000)

例5） カウンタ(CNT)のアドレス 0x00A0 のアップフラグと現在値を取得するために変数を 2 つ登録

C0160$flag と **C0160$val** （”$”以降は解析されず別変数として登録できます）

付録B. 変数書き込み時の PLC(CJ シリーズ)内のメモリ割り付け

put_Value プロパティを用いてメモリ書き込みを行った際に，PLC（CJ シリーズ）内のメモリにどのようにマッピングされるか例を示します．(PLC（CJ シリーズ）内のメモリが全て 0 クリアされている前提で説明します)

例1） 保持リレー(HR) チャネルアドレス 0x0010 の Bit5，Bit7 を ON (1) にする

- 設定用変数を 2 つ登録する

  AddVariable

  <変数 1>

  Name ： H0016.05

  Option ： Mode=NORMAL, VT=BOOL, Elem=1, Array=FALSE

  <変数 2>

  Name ： H0016.07

  Option ： Mode=NORMAL, VT=BOOL, Elem=1, Array=FALSE

- 各変数で値を書き込む

  put_Value

  <変数 1>

  VT_BOOL ： TRUE

  <変数 2>

  VT_BOOL ： TRUE

- PLC（CJ シリーズ）のメモリ内の状態（CX-Programmer を使用）

H0016 の値が 00A0（0000000010100000）

・ 確認のために同チャネルアドレスを読み出す変数を登録する

AddVariable

<変数 3>

Name ： H0016

Option ： Mode=NORMAL, VT=UI2, Elem=1, Array=FALSE

・ 値を読み出す

get_Value

<変数 3>

VT_UI2 ： 160 (0x00A0)

例2) データメモリ(DM) チャネルアドレス 0x0100 に 1 バイト分データを書き込む

- 設定用変数を 1 つ登録する

  AddVariable

  <変数 4>

  Name : D0100

  Option : Mode=NORMAL, VT=UI1, Elem=1, Array=FALSE

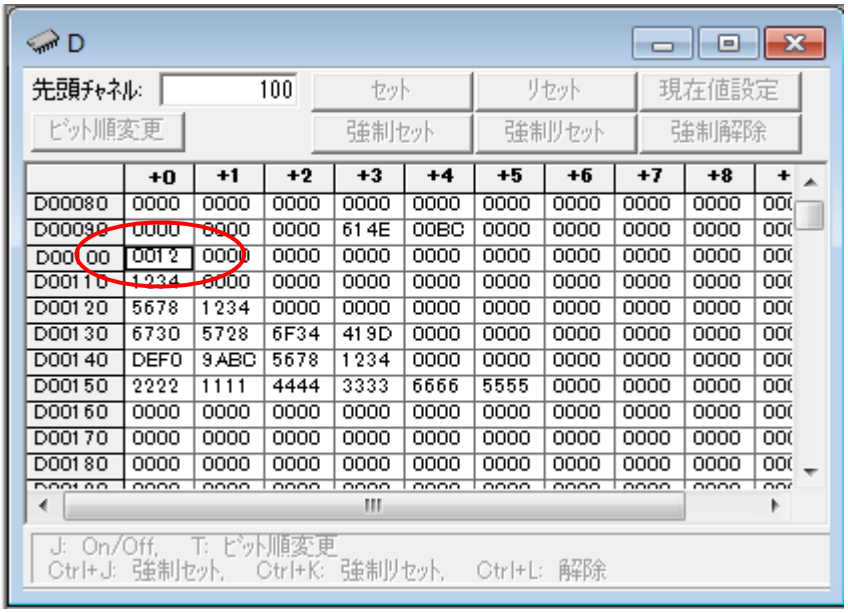
- 値を書き込む

  put_Value

  <変数 4>

  VT_UI1 :18 (0x12)

- PLC(CJ シリーズ)のメモリ内の状態(CX-Programmer を使用)



D0100 の値が 0012(0000000000010010)

- 確認のために同チャネルアドレスを読み出す

  get_Value

  <変数 4>

  VT_UI1 :18 (0x12)

例3) データメモリ(DM) チャネルアドレス 0x0110 に 2 バイト分データを書き込む

・ 設定用変数を 1 つ登録する

AddVariable

<変数 5>

Name : D0110

Option : Mode=NORMAL, VT=UI2, Elem=1, Array=FALSE

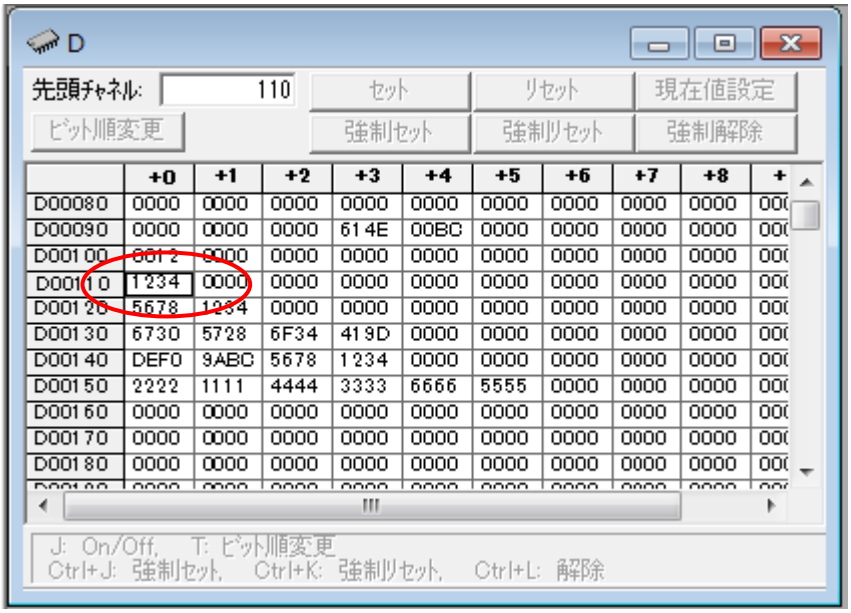
・ 値を書き込む

put_Value

<変数 5>

VT_UI2 :4660 (0x1234)

・ PLC(CJ シリーズ)のメモリ内の状態(CX-Programmer を使用)



D0110 の値が 1234(0001001000110100)

・ 確認のために同チャネルアドレスを読み出す

get_Value

<変数 5>

VT_UI2 :4660 (0x1234)

例4) データメモリ(DM) チャネルアドレス 0x0120 に 4 バイト分データを書き込む

・ 設定用変数を 1 つ登録する

AddVariable

<変数 6>

Name : D0120

Option : Mode=NORMAL, VT=I4, Elem=1, Array=FALSE

・ 値を書き込む

put_Value

<変数 6>

VT_I4 :305419896 (0x12345678)

・ PLC(CJ シリーズ)のメモリ内の状態(CX-Programmer を使用)

D0120 の値が 5678(0101011001111000)

D0121 の値が 1234(0001001000110100)

・ 確認のために同チャネルアドレスを読み出す

get_Value

<変数 6>

VT_I4 :305419896 (0x12345678)

例5) データメモリ(DM) チャネルアドレス 0x0140 に 8 バイト分データを書き込む

- 設定用変数を 1 つ登録する

  AddVariable

  <変数 7>

  Name : D0140

  Option : Mode=NORMAL, VT=I8, Elem=1, Array=FALSE

- 値を書き込む

  put_Value

  <変数 7>

  VT_I8 :1311768467463790320 (0x123456789ABCDEF0)

- PLC(CJ シリーズ)のメモリ内の状態(CX-Programmer を使用)

D0140 の値が DEF0(1101111011110000)

D0141 の値が 9ABC(1001101010111100)

D0142 の値が 5678(0101011001111000)

D0143 の値が 1234(0001001000110100)

- 確認のために同チャネルアドレスを読み出す

  get_Value

  <変数 7>

  VT_I8 :1311768467463790320 (0x123456789ABCDEF0)

例6) データメモリ(DM) チャネルアドレス 0x0150 に 4 バイト分データを 3 つ書き込む

・ 設定用変数を 1 つ登録する

AddVariable

<変数 8>

Name : D0150

Option : Mode=NORMAL, VT=I4, Elem=3, Array=TRUE

・ 値を書き込む

put_Value

<変数 8>

VT_ARRAY | VT_I4 :

286335522 (0x11112222),

858997828 (0x33334444),

1431660134 (0x55556666)

・ PLC(CJ シリーズ)のメモリ内の状態(CX-Programmer を使用)

D0150 の値が 2222(1101111011110000)
D0151 の値が 1111(1001101010111100) } 要素 0

D0152 の値が 4444(0101011001111000)
D0153 の値が 3333(0001001000110100) } 要素 1

D0154 の値が 6666(0001001000110100)
D0155 の値が 5555(0001001000110100) } 要素 2

・ 確認のために同チャネルアドレスを読み出す

get_Value
<変数 8>
VT_ARRAY | VT_I4 :
286335522 (0x11112222),
858997828 (0x33334444),
1431660134 (0x55556666)

付録C. チャネルアドレスの VT=BIT, VT=BOOL, VT=UI1, VT=I1 指定時の書き込み処理について

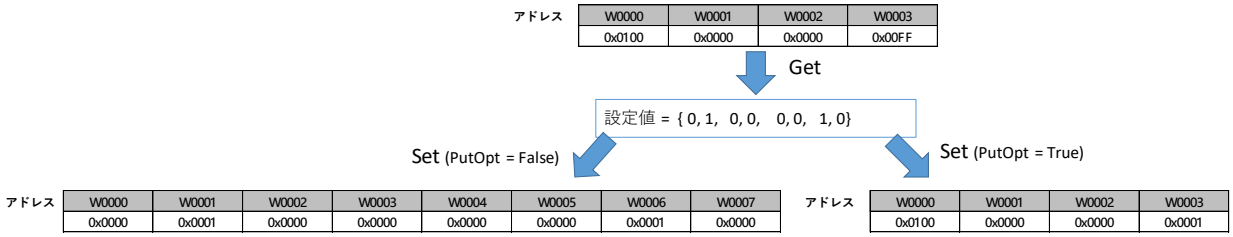
チャネルアドレス指定時に VT=BIT, VT=BOOL, VT=UI1, VT=I1 を指定した場合の挙動を v1.0.1 から選択できるようになりました. v1.0.1 以降に作成する場合は本オプションに True を指定してください.

表 2-9 CaoWorkspace::AddController のオプション文字列

オプション	説明
PutOpt[=< True or False >]	・FALSE (v1.0.1 未満のバージョン動作) データを書き込む際に上位バイトを 0 で補完して書き込みます. ・TRUE (v1.0.1 以降で追加された動作) 下位バイトから順にデータを格納して書き込みます. (デフォルト:FALSE)

以下にそれぞれの動作イメージを記述します.

<変数>
Name : W0
Option : VT=BIT, Elem=8



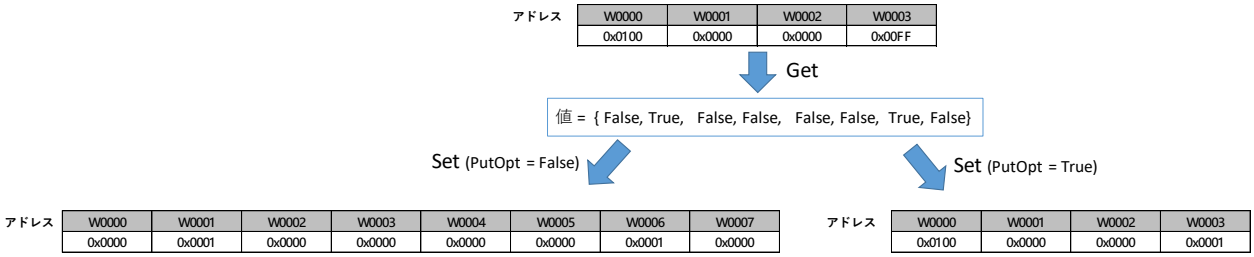
FALSE 時: 読み込み時は, 1 チャネルデータを上位バイトと下位バイトデータに分け, それぞれ 0 以外であれば 1 とします. 書き込み時は, 1 ビットデータを 1 チャネルデータとして扱い, 上位バイトは 0 で補完してデバイスに書き込みます.

TRUE 時: 読み込み時は, 1 チャネルデータを上位バイトと下位バイトデータに分け, それぞれ 0 以外であれば 1 とします. 書き込み時は, 2 ビットデータを 1 チャネルデータとして扱い, 下位バイトから順にデータを格納してデバイスに書き込みます.

<変数>

Name : W0

Option : VT=BOOL, Elem=8



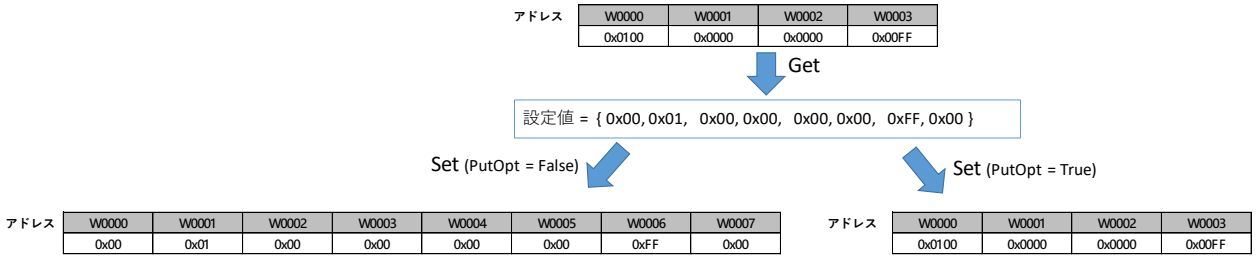
FALSE 時: 読み込み時は, 1 チャネルデータを上位バイトと下位バイトデータに分け, それぞれ 0 以外であれば True とします. 書き込み時は, 1 ブールデータを 1 チャネルデータとして扱い, 上位バイトは 0 で補完してデバイスに書き込みます.

TRUE 時: 読み込み時は, 1 チャネルデータを上位バイトと下位バイトデータに分け, それぞれ 0 以外であれば True とします. 書き込み時は, 2 ブールデータを 1 チャネルデータとして扱い, 下位バイトから順にデータを格納してデバイスに書き込みます.

<変数>

Name : W0

Option : VT=UI1 or I1, Elem=8



FALSE 時: 読み込み時は, 1 チャネルデータを上位バイトと下位バイトデータに分けて値を取得します. 書き込み時は, 1 バイトデータを 1 チャネルデータとして扱い, 上位バイトは 0 で補完してデバイスに書き込みます.

TRUE 時: 読み込み時は, 1 チャネルデータを上位バイトと下位バイトデータに分けて値を取得します. 書き込み時は, 2 バイトのデータを 1 チャネルデータとして扱い, 下位バイトから順にデータを格納してデバイスに書き込みます.