

Beckhoff Automation

TwinCAT3 ADS プロバイダ

Version 1.0.2

ユーザーズ ガイド

November 29, 2018

備考:

本プロバイダはセキュアネットワーク環境下で使用してください。

【改版履歴】

バージョン	日付	内容
1.0.0	2013-10-04	初版.
1.0.1	2016-06-13	WSTRING 型に対応. その他非対応の型を記載.
1.0.1	2017-01-12	空文字を含む文字列/文字列配列型を読み出す際の注意記載.
1.0.2	2018-11-29	CaoController:: get_VariableNames を追加.

【対応機器】

機種	バージョン	注意事項

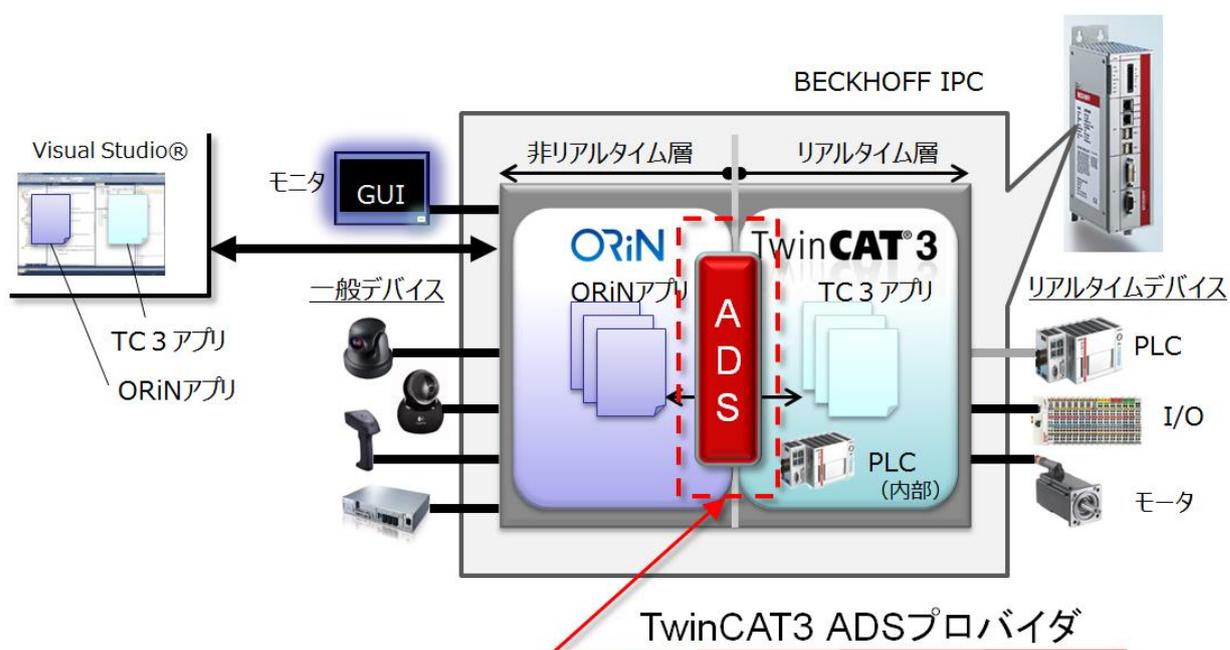
目次

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
2.2.2. CaoController::AddVariable メソッド	7
2.2.3. CaoController:: get_VariableNames プロパティ	10
2.3. 変数一覧	11
2.3.1. コントローラクラス	11
2.4. エラーコード	13
3. サンプルプログラム	15
付録 A. 配列変数へのアクセス	16

1. はじめに

本書は Beckhoff Automation の TwinCAT3 の ADS インターフェイスを利用した 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(CaoProvTC3ADS.dll)を TwinCAT3ADS プロバイダと呼びます。

本プロバイダは Beckhoff Automation 社の IPC に ORiN2 と TwinCAT3 をインストールした環境で動作することを前提としています。



次章に TwinCAT3ADS プロバイダの概要、コマンドリファレンスを記載しています。

2. プロバイダの概要

2.1. 概要

TwinCAT3ADS プロバイダは、BECKHOFF AUTOMATION TwinCAT3 の ADS インターフェイス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。TwinCAT3ADS 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 TwinCAT3ADS プロバイダ

ファイル名	CaoProvTC3ADS.dll
ProgID	CaoProv.BECKHOFF.ADS
レジストリ登録	regsvr32 CaoProvTC3ADS.dll
レジストリ登録の抹消	regsvr32 /u CaoProvTC3ADS.dll

TwinCAT3ADS プロバイダを動作させるには、TwinCAT3 の ADS モジュールである TcAdsDll.dll が必要です。

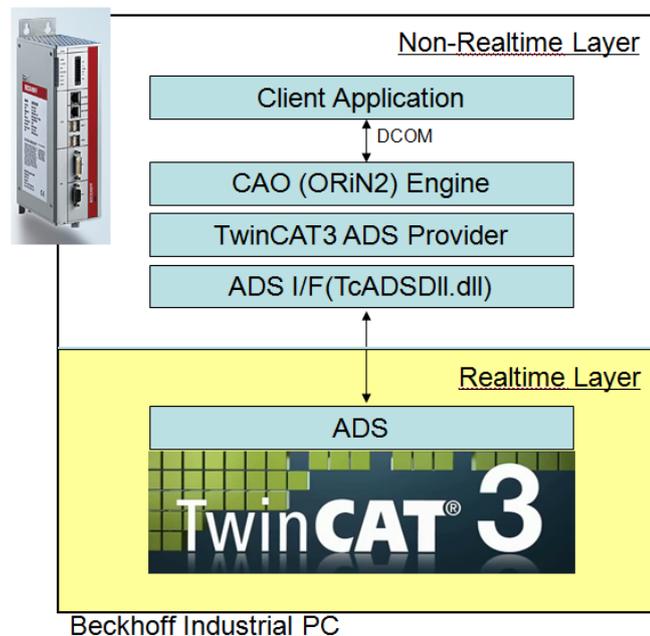


図 2-1 TwinCAT3 ADS プロバイダの構成

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

TwinCAT3ADS プロバイダでは AddController 時にデバイスに接続するパラメータを次の書式で指定します。

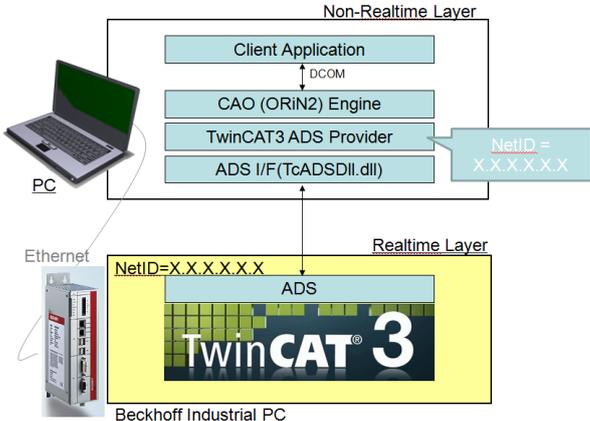
書式

```
AddController(<bstrCtrlName:BSTR>,<bstrProvName>,<bstrPCName:BSTR>,<bstrOption:BSTR>))
```

- <bstrCtrlName> : [in] コントローラ名(VT_BSTR)
任意の文字列を指定
- <bstrProvName> : [in] プロバイダ ProgID(VT_BSTR)
”CaoProv.Beckhoff.ADS”の固定文字列を指定
- <bstrPCName> : [in] PC 名(VT_BSTR)
リモート接続する場合の PC 名を指定.通常のローカル接続の場合は空白文字列(””)を指定.
- <bstrOption> : [in] オプション文字列 (VT_BSTR)
接続を確実にするためのオプション文字列を指定.
各オプションはカンマ(,)区切りで次の書式で指定します.
<OptionName>=<Value>,<OptionName>=<Value>,...
例 : ”NetID=192.168.0.1.1.1, Port = 851”

以下に<bstrOption>オプション文字列に指定するオプションリストを示す。

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

オプション	意味
NetID=X.X.X.X.X	<p>AMS NetID の値を指定</p> <p>省略時はローカルの NetID が自動的に設定されます. X には 0~255 までの値を指定します. NetID は IPC に対してリモート接続する場合に使用します.</p> <p>例 : ”NetID=192.168.0.1.1.1”</p> 

Port=<番号>	AMS Port の値を指定 省略時は 851 指定扱いです.Port 番号は TwinCAT3 内部のデバイス選定時に ID 番号指定に該当します. 例 : ” Port = 851”
-----------	---

使用例

```
Dim oCtrl As Object
```

```
Set oCtrl = caoWorkspace.AddController(“Ads”,”CaoProv.Beckhoff.ADS”,”,”,”Port=851”)
```

2.2.2. CaoController::AddVariable メソッド

CaoController クラスの AddVariable メソッドは、変数にアクセスするためのメソッドです。TwinCAT3ADS プロバイダでは、変数名にシステム変数を指定します。

TwinCAT3ADS プロバイダで実装されているシステム変数は表 2-4 を参照ください。



AddVariable(<bstrVariableName:VT_BSTR>[,<vntOption:VT_BSTR>])

<bstrVariableName> : [in] 変数名(VT_BSTR)

1.システム変数="@任意文字列"

システム変数を指定します. 表 2-4 を参照してください.

2.ユーザー変数=先頭が "@" でない文字列

TwinCAT3 のデバイスで定義されている変数 (Name オプション指定時) やメモリ領域 (Addr オプション指定時) へのマッピングを指定します.

<bstrOption> : [in] 変数オプション(VT_BSTR)

ユーザー変数の場合に以下のオプションが指定可能です.

1. VT オプション

変数型の指定. `VARIANT::VARTYPE` の値を数値または文字列で指定します.

VT オプションを明示的に指示しておらず, 且つ, Name オプションが指定されている場合は, Name オプションの名前を利用して, ADS に型情報を問い合わせ, 取得します.

Addr オプションを指定する場合は VT 指定が必須です.

VT=<変数型名> 例:VT=BOOL

または

VT=<VARTYPE の値> 例:VT=11

変数型名	VARTYPE (値)	
"BOOL"	VT_BOOL (11)	真偽値(True/False)
"STRING"	VT_BSTR (8)	文字列(ASCII)
"BSTR"	VT_BSTR (8)	文字列(ASCII)
"WSTRING"	VT_BSTR (8)	文字列(UNICODE)
"I1"	VT_I1 (16)	1バイト整数 (符号付)
"SINT"	VT_I1 (16)	1バイト整数 (符号付)
"I2"	VT_I2 (2)	2バイト整数 (符号付)
"INT"	VT_I2 (2)	2バイト整数 (符号付)
"I4"	VT_I4 (3)	4バイト整数 (符号付)

"DINT"	VT_I4(3)	4バイト整数 (符号付)
"I8"	VT_I8(20)	8バイト整数 (符号付)
"LINT"	VT_I8(20)	8バイト整数 (符号付)
"UI1"	VT_UI1(17)	1バイト整数 (符号なし)
"BYTE"	VT_UI1(17)	1バイト整数 (符号なし)
"USINT"	VT_UI1(17)	1バイト整数 (符号なし)
"UI2"	VT_UI2(18)	2バイト整数 (符号なし)
"WORD"	VT_UI2(18)	2バイト整数 (符号なし)
"UINT"	VT_UI2(18)	2バイト整数 (符号なし)
"UI4"	VT_UI4(19)	4バイト整数 (符号なし)
"DWORD"	VT_UI4(19)	4バイト整数 (符号なし)
"UDINT"	VT_UI4(19)	4バイト整数 (符号なし)
"UI8"	VT_UI8(21)	8バイト整数 (符号なし)
"ULINT"	VT_UI8(21)	8バイト整数 (符号なし)
"R4"	VT_R4(4)	4バイト浮動小数点数
"REAL"	VT_R4(4)	4バイト浮動小数点数
"R8"	VT_R8(5)	8バイト浮動小数点数
"LREAL"	VT_R8(5)	8バイト浮動小数点数

2. Name オプション指定

変数名を指定してアクセスする値を指定します。

Name=<プログラム名>.<変数>

例: Name=Main.bVar

‘TwinCAT3 定義 Main プログラム bVar: BOOL

3. Addr オプション指定

アドレスを指定してアクセスする値を指定します。

Addr=<Group>:<Offset>

例: Addr=0x4021:80, VT=BOOL

‘TwinCAT3 定義

PROGRAM MAIN

VAR

bVar AT%MX10.0: BOOL;

END_VAR

※ Name および Addr オプションを指定しなかった場合は Name=<bstrVariableName>が指定されたものとして扱われます。ORiN の変数名と TwinCAT3 内部のデバイスの変数と対応させることができます。

※ 以下の変数型には対応していません。

LWORD, DATA, DATA_AND_TIME, DT,

TIME_OF_DAY, TOD, TIME, LTIME

※ 配列変数へのアクセスには注意が必要です。

詳細は 3.付録 A を参照してください。

使用例

```
Dim oVar As Object
Set oVar = caoCtrl.AddVariable("Main.bVar", "")
Debug.Print oVar.Value
```

2.2.3. CaoController:: get_VariableNames プロパティ

2.3.1 の変数を取得します。

2.3. 変数一覧

2.3.1. コントローラクラス

表 2-3 コントローラクラス ユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
<プログラム名>.<変数>	変数型依存	TwinCAT3 のデバイスで定義されている変数名と ORiN 変数名を一致させる. Name,Addr オプションは指定しない. 例: caoCtrl.AddVariable("Main.bVar", "")	○	○
任意名	変数型依存	TwinCAT3 のデバイスで定義されている変数名と ORiN 変数名を独立させる. Name または Addr オプションのいずれか一つを必ず指定する. 例: caoCtrl.AddVariable("aaa", "Name=Main.bVar") caoCtrl.AddVariable("bbb", "Addr=0x4021:80, VT=BOOL")	○	○

表 2-4 コントローラクラス 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	メーカー名="Beckhoff Automation" を返す	○	-
@VERSION	VT_BSTR	ADS モジュールのバージョン情報 <Version>.<Revision>.<Build>を返す	○	-
@STATUS	VT_I4	ADS の状態 ADSSTATE_INVALID = 0 ADSSTATE_IDLE = 1 ADSSTATE_RESET = 2 ADSSTATE_INIT = 3 ADSSTATE_START = 4 ADSSTATE_RUN = 5 ADSSTATE_STOP = 6 ADSSTATE_SAVECFG = 7 ADSSTATE_LOADCFG = 8 ADSSTATE_POWERFAILURE = 9 ADSSTATE_POWERGOOD = 10 ADSSTATE_ERROR = 11 ADSSTATE_SHUTDOWN = 12 ADSSTATE_SUSPEND = 13 ADSSTATE_RESUME = 14 ADSSTATE_CONFIG = 15 ADSSTATE_RECONFIG = 16 ※書き込みは状態の値により可能	○	○
@TIMEOUT	VT_I4	ADS 通信のタイムアウト時間(ms) デフォルトでは 5000ms	○	○

2.4. エラーコード

TwinCAT3ADS プロバイダでは、固有のエラーコードを下記に示します。

エラーの詳細に関してはエラーコードの下位4桁(16bit)をTwinCAT3のADS DLLのドキュメント記載のコードを照らし合わせて参照ください。(0x8010XXXXのXXXXはADSのエラー番号に該当します。)

その他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。

Error Macro	Error Number	Description
ADSERR_DEVICE_ERROR	0x80100700	Error class < device error >
ADSERR_DEVICE_SRVNOTSUPP	0x80100701	Service is not supported by server
ADSERR_DEVICE_INVALIDGRP	0x80100702	invalid indexGroup
ADSERR_DEVICE_INVALIDOFFSET	0x80100703	invalid indexOffset
ADSERR_DEVICE_INVALIDACCESS	0x80100704	reading/writing not permitted
ADSERR_DEVICE_INVALIDSIZE	0x80100705	parameter size not correct
ADSERR_DEVICE_INVALIDDATA	0x80100706	invalid parameter value(s)
ADSERR_DEVICE_NOTREADY	0x80100707	device is not in a ready state
ADSERR_DEVICE_BUSY	0x80100708	device is busy
ADSERR_DEVICE_INVALIDCONTEXT	0x80100709	invalid context (must be InWindows)
ADSERR_DEVICE_NOMEMORY	0x8010070A	out of memory
ADSERR_DEVICE_INVALIDPARM	0x8010070B	invalid parameter value(s)
ADSERR_DEVICE_NOTFOUND	0x8010070C	not found (files, ...)
ADSERR_DEVICE_SYNTAX	0x8010070D	syntax error in comand or file
ADSERR_DEVICE_INCOMPATIBLE	0x8010070E	objects do not match
ADSERR_DEVICE_EXISTS	0x8010070F	object already exists
ADSERR_DEVICE_SYMBOLNOTFOUND	0x80100710	symbol not found
ADSERR_DEVICE_SYMBOLVERSIONINVALID	0x80100711	symbol version invalid
ADSERR_DEVICE_INVALIDSTATE	0x80100712	server is in invalid state
ADSERR_DEVICE_TRANSMODENOTSUPP	0x80100713	AdsTransMode not supported
ADSERR_DEVICE_NOTIFYHNDINVALID	0x80100714	Notification handle is invalid
ADSERR_DEVICE_CLIENTUNKNOWN	0x80100715	Notification client not registered
ADSERR_DEVICE_NOMOREHDLS	0x80100716	no more notification handles
ADSERR_DEVICE_INVALIDWATCHSIZE	0x80100717	size for watch to big
ADSERR_DEVICE_NOTINIT	0x80100718	device not initialized
ADSERR_DEVICE_TIMEOUT	0x80100719	device has a timeout
ADSERR_DEVICE_NOINTERFACE	0x8010071A	query interface failed
ADSERR_DEVICE_INVALIDINTERFACE	0x8010071B	wrong interface required

ADSERR_DEVICE_INVALIDCLSID	0x8010071C	class ID is invalid
ADSERR_DEVICE_INVALIDOBJID	0x8010071D	object ID is invalid
ADSERR_DEVICE_PENDING	0x8010071E	request is pending
ADSERR_DEVICE_ABORTED	0x8010071F	request is aborted
ADSERR_DEVICE_WARNING	0x80100720	signal warning
ADSERR_DEVICE_INVALIDARRAYIDX	0x80100721	invalid array index
ADSERR_CLIENT_ERROR	0x80100740	Error class < client error >
ADSERR_CLIENT_INVALIDPARG	0x80100741	invalid parameter at service call
ADSERR_CLIENT_LISTEMPTY	0x80100742	polling list is empty
ADSERR_CLIENT_VARUSED	0x80100743	var connection already in use
ADSERR_CLIENT_DUPLINVOKEID	0x80100744	invoke id in use
ADSERR_CLIENT_SYNCSTIMEOUT	0x80100745	timeout elapsed
ADSERR_CLIENT_W32ERROR	0x80100746	error in win32 subsystem
ADSERR_CLIENT_TIMEOUTINVALID	0x80100747	Timeout is invalid.
ADSERR_CLIENT_PORTNOTOPEN	0x80100748	Port is not opened.
ADSERR_CLIENT_NOAMSADDR	0x80100749	No AMS address.
ADSERR_CLIENT_SYNCINTERNAL	0x80100750	internal error in ads sync
ADSERR_CLIENT_ADDHASH	0x80100751	hash table overflow
ADSERR_CLIENT_REMOVEHASH	0x80100752	key not found in hash table
ADSERR_CLIENT_NOMORESVM	0x80100753	no more symbols in cache
ADSERR_CLIENT_SYNCRESINVALID	0x80100754	invalid response received
ADSERR_CLIENT_SYNCPORTLOCKED	0x80100755	sync port is locked

3. サンプルプログラム

// TwinCAT3 の 851 ポート:Main プログラムの bVar(BOOL)変数へのアクセス.

‘TwinCAT3 定義 -----

PROGRAM MAIN

VAR

 bVar AT%MX10.0: BOOL;

END_VAR

List 3-1

Sample.frm

```
Private caoEng As CaoEngine
Private caoWS As CaoWorkspace
Private TwinCAT3ADSctrl As CaoController

Private Sub Form_Load()
    ' CAO エンジンと CAO ワークスペースの作成
    Set caoEng = New CaoEngine
    Set caoWS = caoEng.Workspaces.Item(0)

    ' TwinCAT3ADS の CAO コントローラ
    Set TwinCAT3ADSctrl = caoWS.AddController("Ads", "CaoProv. Beckhoff.ADS", "", "Port=851")

    ' 変数オブジェクトを取得
    Set TwinCAT3ADSbVar = TwinCAT3ADSctrl.AddVariable("Main.bVar", "")

    TwinCAT3ADSbVar.Value = True

    Debug.Print TwinCAT3ADSbVar.Value

    TwinCAT3ADSbVar.Value = False

    Debug.Print TwinCAT3ADSbVar.Value
End Sub
```

付録A. 配列変数へのアクセス

TwinCAT3ADS プロバイダでは TwinCAT3 上に定義された配列変数に対してもアクセスすることが可能です。(一部制限のある変数型があります)

変数型毎のアクセス可否とアクセスする際の注意事項を以下に示します。

表 3-1 配列にした際の変数型毎のアクセス可否

変数型	アクセス可否		備考
	PUT	GET	
BOOL	○	○	
BYTE	○	○	
WORD	○	○	
DWORD	○	○	
LWORD	×	×	LWORD 型自体が非対応.
SINT	○	○	
USINT	○	○	
INT	○	○	
UINT	○	○	
DINT	○	○	
UDINT	○	○	
LINT	○	○	
ULINT	○	○	
REAL	○	○	
LREAL	○	○	
STRING	×	○	GET のみ対応.
WSTRING	×	○	GET のみ対応.
DATE	×	×	DATE 型自体が非対応.
DATE_AND_TIME	×	×	DATE_AND_TIME 型自体が非対応.
DT	×	×	DT 型自体が非対応.
TIME_OF_DAY	×	×	TIME_OF_DAY 型自体が非対応.
TOD	×	×	TOD 型自体が非対応.
TIME	×	×	TIME 型自体が非対応.
LTIME	×	×	LTIME 型自体が非対応.

```
PROGRAM MAIN
VAR (* TwinCAT3 配列変数定義例 *)

(*BOOL*)
bArray: ARRAY[0..1] OF BOOL := [TRUE, FALSE];

(*BYTE*)
byArray: ARRAY[0..2] OF BYTE := [1, 2, 3];

(*WORD*)
wArray: ARRAY[0..2] OF WORD := [1, 2, 3];

(*DWORD*)
dwArray: ARRAY[0..2] OF DWORD := [1, 2, 3];

(*SINT*)
siArray: ARRAY[0..2] OF SINT := [-127, 0, 127];

(*USINT*)
usiArray: ARRAY[0..1] OF USINT := [0, 255];

(*INT*)
iArray: ARRAY[0..2] OF INT := [-32768, 0, 32767];

(*UINT*)
uiArray: ARRAY[0..1] OF UINT := [0, 65535];

(*DINT*)
diArray: ARRAY[0..2] OF DINT := [-134217728, 0, 134217727];

(*UDINT*)
udiArray: ARRAY[0..1] OF UDINT := [0, 4294967295];

(*LINT*)
liArray: ARRAY[0..2] OF LINT := [-9223372036854775807, 0, 9223372036854775807];

(*ULINT*)
uliArray: ARRAY[0..1] OF ULINT := [0, 18446744073709551615];

(*REAL*)
rArray: ARRAY[0..1] OF REAL := [1.234, 5.678];

(*LREAL*)
lrArray: ARRAY[0..1] OF LREAL := [1.234, 5.678];

(*STRING*)
sArray1: ARRAY[0..3] OF STRING := ['aaa', 'bbb', 'ccc'];
sArray2: ARRAY[0..3] OF STRING(2) := ['aa', 'bb', 'cc'];

(*WSTRING*)
wsArray1: ARRAY[0..3] OF WSTRING := ["あいうえお", "かきくけこ", "さしすせそ"];
wsArray2: ARRAY[0..3] OF WSTRING(2) := ["ああ", "いい", "うう"];

END_VAR
```

■ 配列内の 1 要素辺りのサイズと値の範囲について

配列として変数が定義されている場合、定義された要素数を超えての PUT を行うことはできません。

また、TwinCAT3ADS プロバイダは VARIANT 型から各変数の型への変換を行った上でアクセスを行います。(GET 時は各対応型の配列として取得されます)

各型の対応と値の範囲に関しては 2.2.2 及び BECKHOFF AUTOMATION TwinCAT3 のマニュアルを参照してください。

■ 文字列を扱う変数について

ASCII 文字列を扱う STRING 型、UNICODE 文字列を扱う WSTRING 型共に変数の定義の仕方によって文字数(TwinCAT3 内に確保されるバッファサイズ)が可変となります。

単純に STRING や WSTRING と型定義した場合は 80 文字(STRING:80+1 バイト, WSTRING:160+2 バイト)分の領域が確保されます。STRING(2)や WSTRING(2)のように型定義した場合は文字数指定となり、この場合は 2 文字(STRING:2+1 バイト, WSTRING:4+2 バイト)分の領域が確保されます。

確保された領域サイズより少ない文字数のデータを PUT した場合は残りの領域を NULL(0x00)でパディングします。

文字列の配列に関しては TwinCAT3 側から 1 要素辺りの文字数情報を取得することができないため、2 要素目の書き込み位置を TwinCAT3ADS プロバイダでは特定できなく PUT は非対応となっています。GET 時は取得データが文字終端の NULL を含んで複数データが存在する場合に限り、パディング分を含む文字終端(NULL)でデータを分割し、(VT_ARRAY | VT_BSTR)の配列データとして取得結果を返します。

ただし、文字列変数の値が空文字である、または文字列配列中の要素に空文字を含む場合は読み出した値が正しく解釈されないため注意が必要です。

例1) 配列変数定義例の sArray1 を GET した場合

(VT_ARRAY | VT_BSTR)

[0] VT_BSTR : aaa

[1] VT_BSTR : bbb

[2] VT_BSTR : ccc

例2) 配列変数定義例の wsArray1 を GET した場合

(VT_ARRAY | VT_BSTR)

[0] VT_BSTR : あいうえお

[1] VT_BSTR : かきくけこ

[2] VT_BSTR : さしすせそ