

# VPS プロバイダ 富士通 VPS

Version 1.1.1

## ユーザーズ ガイド

July 17, 2012

【備考】

**【改版履歴】**

バージョン	日付	内容
1.0.0.0	2008-03-18	初版.
1.1.0.0	2009-05-19	VPS V20L19R3M0 対応
1.1.1.0	2009-11-11	VPS V20L19R2M0 でも動作するように修正
1.1.1.1	2010-02-11	エラーコード追加
1.1.1	2012-07-17	ドキュメントのバージョンルールを変更

**【対応機器】**

機種	バージョン	注意事項
VPS	V20L19R3M0	
VPS	V20L19R2M0	

## 目次

1. はじめに .....	4
1.1. システム構成 .....	4
2. プロバイダの概要 .....	6
2.1. インストール .....	6
2.2. 概要 .....	6
2.3. メソッド・プロパティ .....	7
2.3.1. CaoWorkspace::AddController メソッド .....	7
2.3.2. CaoController::AddVariable メソッド .....	8
2.3.3. CaoController::Execute メソッド .....	8
2.3.4. CaoVariable::put_Value プロパティ .....	8
2.3.5. CaoVariable::get_Value プロパティ .....	9
2.4. 変数一覧 .....	10
2.4.1. コントローラクラス .....	10
2.5. エラーコード .....	10
3. サンプルプログラム .....	12
4. CaoSQL による接続 .....	13

## 1. はじめに

VPS プロバイダは、FUJITSU 社が開発したデジタル・モックアップ・ツールである VPS (Virtual Product Simulator) と通信を行う ORiN2 CAO プロバイダです。

VPS プロバイダを利用することで ORiN2 アプリケーションは、VPS とリアルタイムにデータ交換したり、実デバイスと同期しながらシミュレーションしたりすることができます。ORiN2 アプリケーションからは VPS の仮想 I/O を実デバイスの I/O と同じように扱えます。これにより図 1-1 のような実デバイスを用いた設備シミュレーションシステムも容易に構築することができます。

本ドキュメントでは、VPS プロバイダの概要と、実装されている CAO インターフェース(関数仕様)について説明しています。

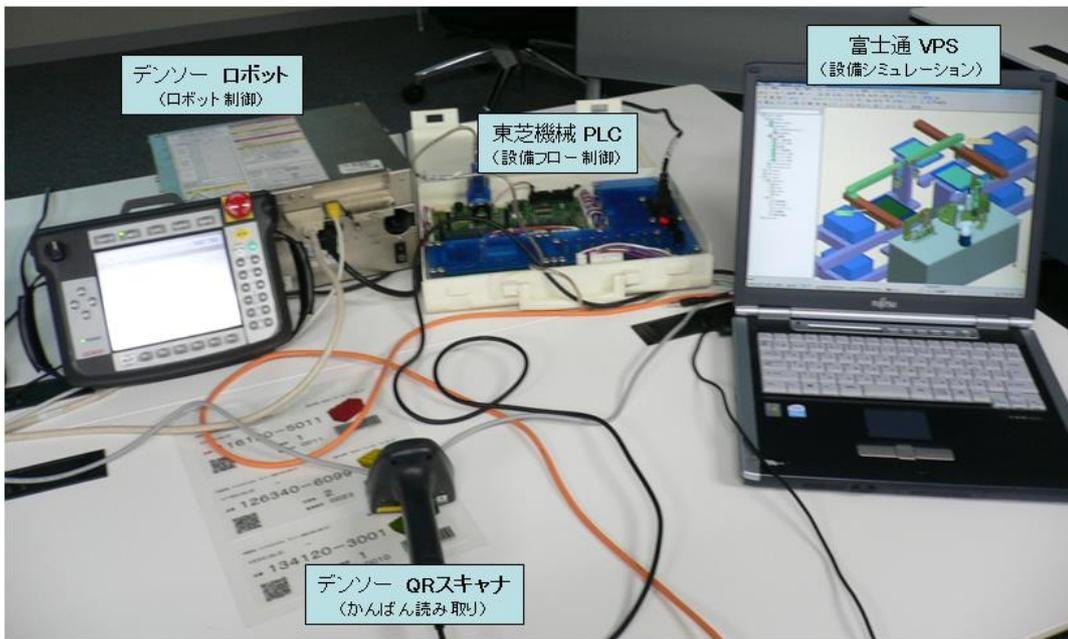


図 1-1 VPS による実デバイスを統合した設備シミュレーション例

### 1.1. システム構成

図 1-2 は ORiN2 と VPS を使った場合のシステム構成を示しています。VPS プロバイダと CaoSQL のリンク機能を使うとプログラムレスで PLC の I/O やレジスタの値、ロボットの I/O や現在位置などを VPS へ送信したり、逆に VPS の値を PLC やロボットへ送信するというようなことができます。この CaoSQL の設定については 4 章を参照してください。

また、VPS シミュレーションとの同期方法については、「VPS IOConnector ガイド」を参照してください。同様な方法で VPS とデバイス間で同期を取ることができます。

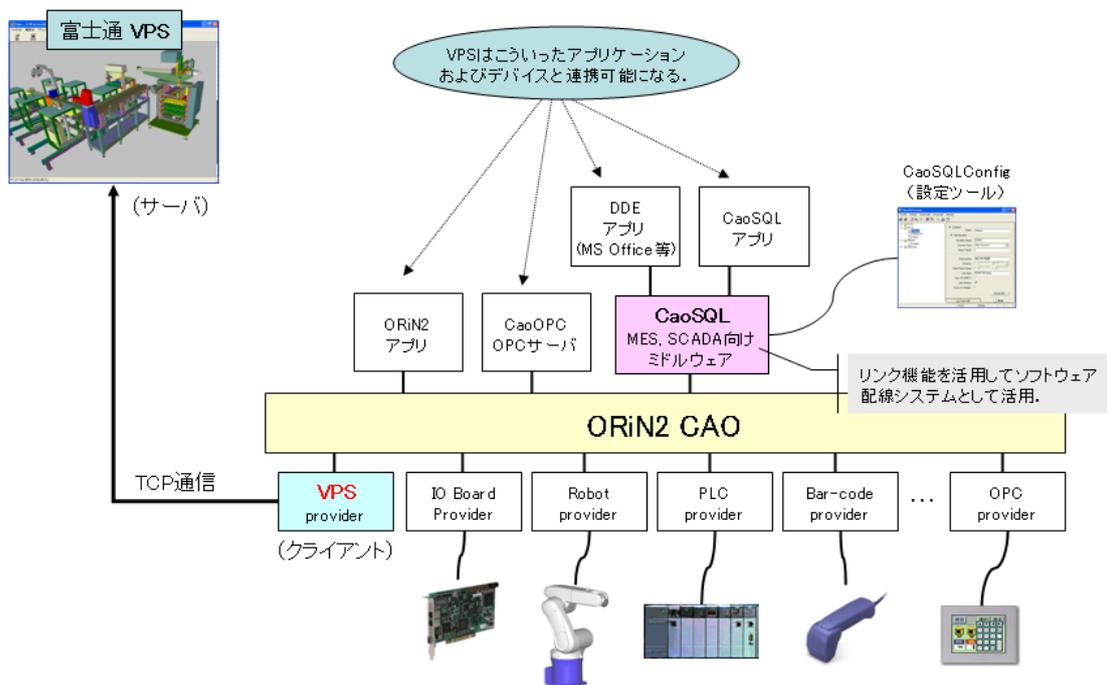


図 1-2 VPS との接続方法の概要

## 2. プロバイダの概要

### 2.1. インストール

VPS プロバイダモジュールは下記の DLL 一つだけで構成されています。ORiN2 SDK のインストーラでインストールした場合はインストール作業は不要ですので、次節のライセンス追加を行ってください。手動でインストールする場合は表 2-1 のように実行してください。

また、VPS 側の設定に関しては「VPS IOConnector ガイド」を参照してください。

表 2-1: VPS プロバイダ

ファイル名	vpsio.dll
ProgID	CaoProv.Fujitsu.VPS
レジストリ登録 <sup>1</sup>	regsvr32 vpsio.dll
レジストリ登録の抹消	regsvr32 /u vpsio.dll

### 2.2. 概要

VPS プロバイダは、TCP/IP で VPS と通信を行います。このとき VPS プロバイダは TCP クライアントとして動作します。

VPS と接続したとき、VPS プロバイダは 4 つの TCP ポートを開きます。それぞれのポートではデジタル入出力 (PI / PO)、アナログ入出力 (AD / DA)、パルス入力 (PC)、文字列送信 (STR)、共有データ領域 (DT) の通信を行います。クライアントアプリケーションから値の書き込みを行った時、VPS に対してパケットを送信します。VPS からの受信があったときには、OnMessage イベントでそのクライアントに通知します。

PIO, DA, PC は内部キャッシュを持ち、プロバイダ実行中の値を保持します (図 2-1)。内部キャッシュサイズの設定は、レジストリに登録します。この設定は CaoConfig を使用することで変更することができます (表 2-2 CaoConfig の Parameter 文字列)。

<sup>1</sup> ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません。

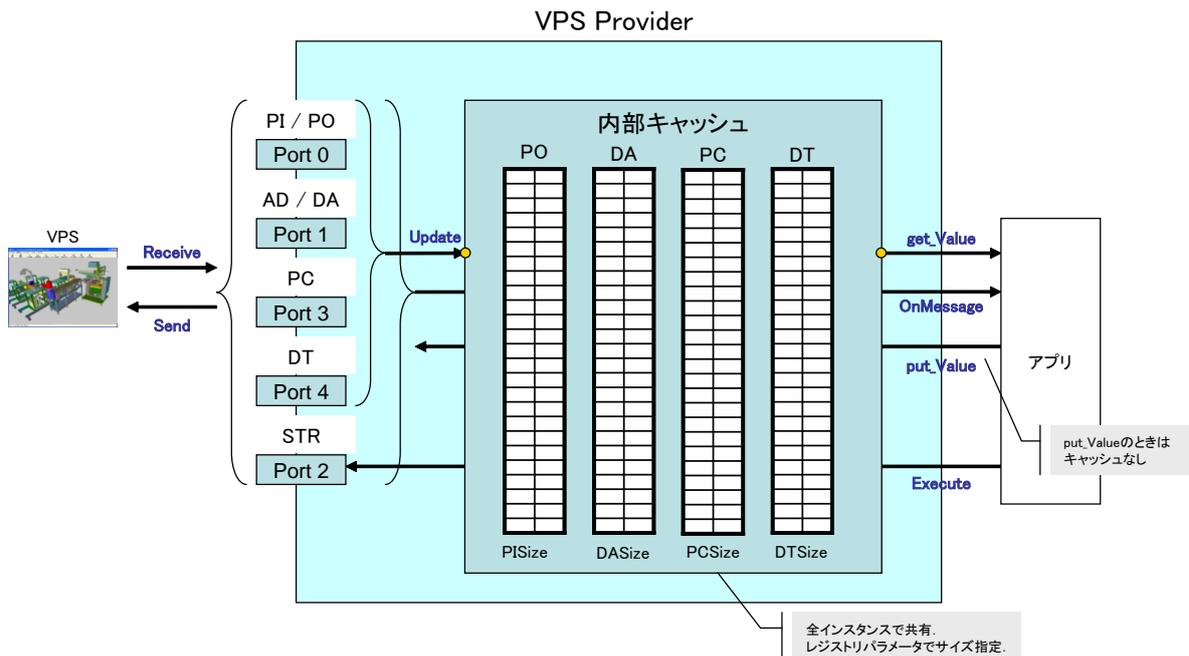


図 2-1 VPS プロバイダの概要

表 2-2 CaoConfig の Parameter 文字列

パラメータ	意味
PISize=<PI 内部キャッシュサイズ>	PI の内部キャッシュサイズを指定します。 (デフォルト:256)
ADSize=<DA 内部キャッシュサイズ>	DA の内部キャッシュサイズを指定します。 (デフォルト:64)
DTSize=<DT 内部キャッシュサイズ>	DT の内部キャッシュサイズを指定します。 (デフォルト:256)

## 2.3. メソッド・プロパティ

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

VPS プロバイダでは AddController 時に、接続先の VPS 及び先頭の TCP ポート番号を指定します。

**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.Fujitsu.VPS".  
 bstrPcName : [in] プロバイダの実行マシン名  
 bstrOption : [in] オプション文字列

表 2-3 CaoWorkspace::AddController のオプション文字列

パラメータ	意味
IP=<IP アドレス>	VPS サーバの IP アドレス(必須)
Port0=<ポート番号>	VPS と通信する TCP ポート番号(必須) 指定したポート番号から連番で 5 つのポートを VPS との通信用にオープンします。 各ポートは以下の目的に使用されます。 Port0 + 0 : PI / PO Port0 + 1 : AD / DA Port0 + 2 : STR Port0 + 3 : PC Port0 + 4 : DT DT 通信用ポート(Port0 + 4)の接続が失敗した場合にはエラーを返しません。

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

VPS と値の入出力を行うの変数オブジェクトを作成します。変数名には 表 2-4 の変数のみ使用することができます。

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

bstrName : [in] 変数名(表 2-4)

bstrOption : [in] オプション文字列

### 2.3.3. CaoController::Execute メソッド

STR に文字列を送信します。送信時にはパラメータで指定した文字列の頭に“0x”, 後ろに“\n”を付加して送信します。

コマンド名は“PutSTR”, パラメータは“<モータ番号> + <STR コマンド>”を指定してください。

**書式** Execute( <bstrCommand:BSTR>, <vntParam:VARIANT> )

bstrCommand : [in] コマンド名 “PutSTR”

vntParam : [in] パラメータ “<モータ番号> + <STR コマンド>”

### 2.3.4. CaoVariable::put\_Value プロパティ

VPS にデータを送信します。

### 2.3.5. CaoVariable::get\_Value プロパティ

内部キャッシュのデータを読み込みます。

## 2.4. 変数一覧

### 2.4.1. コントローラクラス

表 2-4 コントローラクラスのユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
M<モータ番号>:<コマンド名>:<設定パラメータ>	VT_VARIANT	STR 用変数 設定パラメータ内に複数パラメータがあるときはコロンで区切ります。 例) モータ番号:1 コマンド名:0A 設定パラメータ 第1要素:2 第2要素:1 第3要素:255 "M1:0A:2:1:255"  また, 設定パラメータは1つのみ省略することが可能です. 省略したパラメータは put_Value で値を指定することができます. 例) 設定パラメータの第2要素を省略した場合 "M1:0A:2::255"	-	○
PI<ピン番号>	VT_BOOL	PI 用変数	-	○
PO<ピン番号>	VT_BOOL	PO 用変数	○	-
AD<ピン番号>	VT_R8	AD 用変数	-	○
DA<ピン番号>	VT_R8	DA 用変数	○	-
PC<ピン番号>	VT_UI4	PC 用変数	-	○
DT<ピン番号>	VT_R8	DT 用変数	○	○

## 2.5. エラーコード

VPS 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. 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 2-5 独自エラーコード一覧**

エラー名	エラー番号	説明
E_CAOP_NO_LICENSE	0x80100000	ライセンスがありません。

### 3. サンプルプログラム

VPS プロバイダと Visual Basic 6 を使った簡単なサンプルプログラムを紹介します。このように他のデバイスへアクセスするのと同じプログラミングスタイルで VPS にアクセスできるようになります。また、もう少し詳細なサンプルが下記のフォルダにありますのでそちらも参考にしてください。

<ORiN2>¥CAO¥ProviderLib¥FUJITSU¥VPS¥Sample

#### List A-1

```
' -----  
' 変数宣言  
Dim caoEng as CaoEngine  
Dim caoWs as CaoWorkspace  
Dim caoCtrl as CaoController  
Dim varIO1 as CaoVariable  
Dim varDA1 as CaoVariable  
  
' オブジェクト作成  
Set caoEng = new CaoEngine  
Set caoWs = caoEng.Workspaces(0)  
Set caoCtrl = caoWs.AddController("vps", "GaoProv.Fujitsu.VPS", "", "IP=192.168.0.1")  
Set caoIO1 = caoCtrl.AddVariable("PI01")  
Set caoDA1 = caoCtrl.AddVariable("DA1")  
  
' 値の設定と取得  
caoIO1.Value = 0           ' オフ  
caoIO1.Value = 1         ' オン  
Msgbox caoDA1.Value      ' DA1 の現在値
```

## 4. CaoSQL による接続

CaoSQL (SCADA 向けの CAO 上のミドルウェア) のリンク機能を使うと前節のようなプログラムを記述することなく様々なデバイスとの連携が可能になります。例えば、下記の CaoSQL コンフィギュレーションファイル (\*.csq) を CaoSQLConfig ツールでオープンしてください。

```
<ORiN2>¥CAO¥ProviderLib¥FUJITSU¥VPS¥Bin¥vpstest.csq
```

この例では、ロボットの現在の軸角度(CurJnt)を VPS のモータにリンクしています(図 4-1)。この設定のポイントは下記の通りです。

1. CurJnt は配列なので直接 M 変数へ送ることはできないため、まずは配列の展開リンク機能を使ってバラバラにします。
2. さらに CurJnt の要素は単精度実数であり、M 変数は整数しか扱えないため、定数倍(1000倍)してリンクしています。定数倍されていることを VPS が理解できるように変数名が定義されています。

このような設定だけでロボットのオフラインポイント教示が可能になります。CaoSQL の詳細は「CaoSQL ユーザーズガイド」を参照してください。

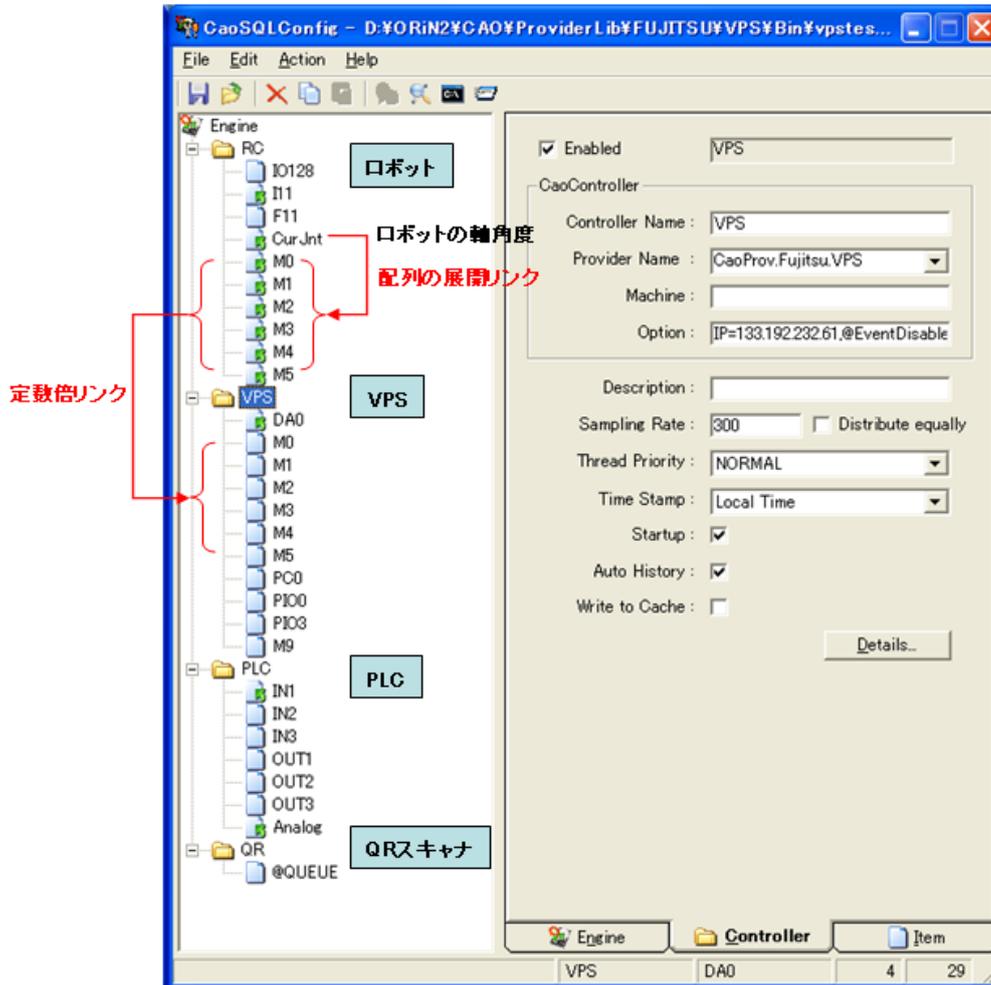


図 4-1: CaoSQLConfig ツールによるリンク設定

## 付録A. Ver. 1.1.\*の変更点

### [変更点 1] CaoConfig のパラメータ

各変数の内部キャッシュサイズを指定する CaoConfig の Parameter 文字列が変更されました。  
変更の詳細は以下のようになります。

表 4-1 CaoConfig の Parameter 文字列変更点

変更前	変更後	備考
PIOSize	PISize	
DASize	ADSize	
<del>PCSize</del>	<del>PCSize</del>	<del>変更なし</del> Ver.1.1.1 より廃止
-	DTSize	新規追加

### [変更点 2] 変数の追加・変更

変数クラスで使用可能な変数名及び実装されている属性の内容が変更されました。  
変更の詳細は以下のようになります。

表 4-2 変数クラスの実装状況変更点

変更前			変更後			備考
変数名	属性		変数名	属性		
	get	put		get	put	
-	-	-	DT	○	○	新規追加
PIO	○	○	PI	-	○	
			PO	○	-	
DA	○	○	AD	-	○	
			DA	○	-	
PC	○	○	PC	<del>○</del>	○	Ver.1.1.1 より get 不可, put 可
M	-	○	M	-	○	変更なし