

Brother TC Protocol2 プロバイダ

Version 1.1.3

ユーザーズ ガイド

March 23, 2020

備考：

このプロバイダを使用すると、設備のハードウェアが故障する可能性があります。

【改版履歴】

バージョン	日付	内容
1.0.0	2018-05-07	初版.
1.1.0	2018-07-05	キープアライブ追加
1.1.1	2018-07-13	変数名変更
1.1.2	2018-09-18	受信データ解析の不具合修正
1.1.3	2020-03-23	複数接続時の不具合修正

【接続確認済み機器】

■ [機器名]

機種名	ファームウェアバージョン	注意事項
TC-S2D		

内容

1. はじめに.....	4
1.1. 本書が想定している環境とバージョン	5
1.2. 参考となる情報源.....	5
2. アプリケーション開発のための環境セットアップ	6
2.1. 接続方法.....	6
2.2. PC 環境のセットアップ.....	6
2.2.1. Brother TC Protocol2 プロバイダの自動インストール	6
2.2.2. Brother TC Protocol2 プロバイダの手動インストール	6
3. Brother TC Protocol2 プロバイダによるプログラミング	7
3.1. 工具データにアクセスするサンプルプログラミング	7
3.1.1. サンプルプログラム.....	8
4. コマンドリファレンス	12
4.1. コマンド一覧.....	12
4.2. メソッド・プロパティ	12
4.2.1. CaoWorkspace::AddController メソッド.....	12
4.2.2. CaoController::AddVariable メソッド.....	14
4.2.3. CaoController::GetVariableNames メソッド.....	15
4.2.4. CaoController::Execute メソッド	15
4.2.5. CaoVariable::get_Value プロパティ.....	19
4.3. 変数一覧.....	19
4.3.1. コントローラクラス	19
4.3.2. CaoVariable::put_Value プロパティ.....	31

1. はじめに

本書は, brother 社製 CNC のタッピングセンタ (TC-S2D 等) に対して TC 専用通信プロトコル方式 2 を使用してデータの読み込み, 及び書き込みを行なう, Brother TC Protocol2 プロバイダのユーザーズガイドです. 下図 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.

以降 brother 製のタッピングセンタをタッピングセンタと呼称します.

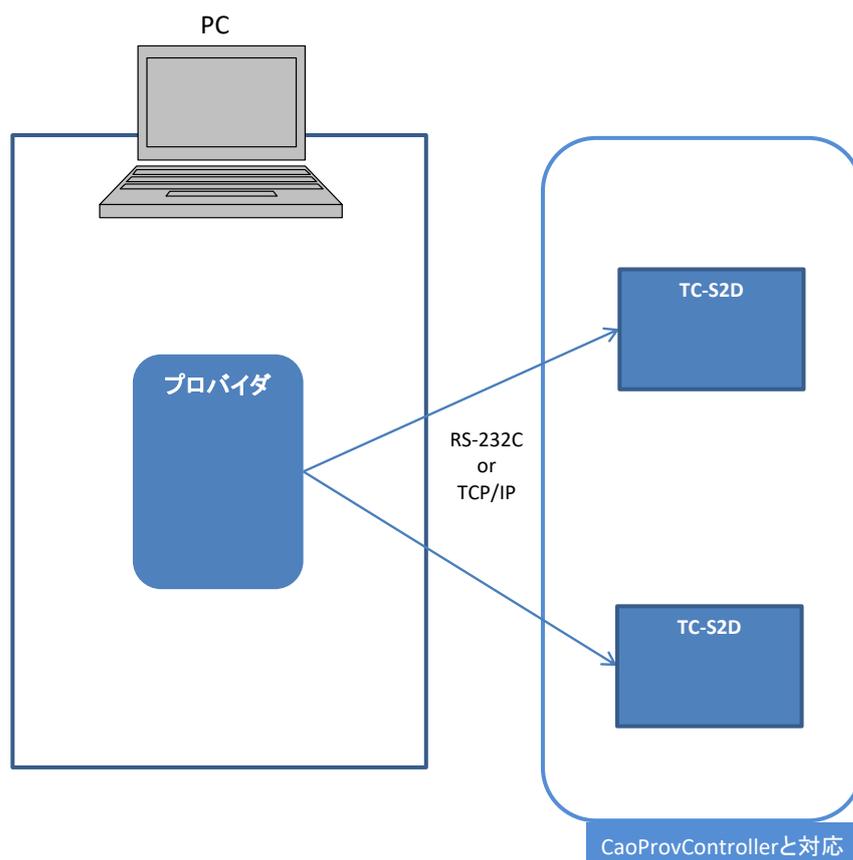


図 1-1 全体構成図

また, 本プロバイダ及びデバイスそれぞれの対応を表した図が下図 1-2となります.

(※一例です. 全てを表しているわけではありません.)

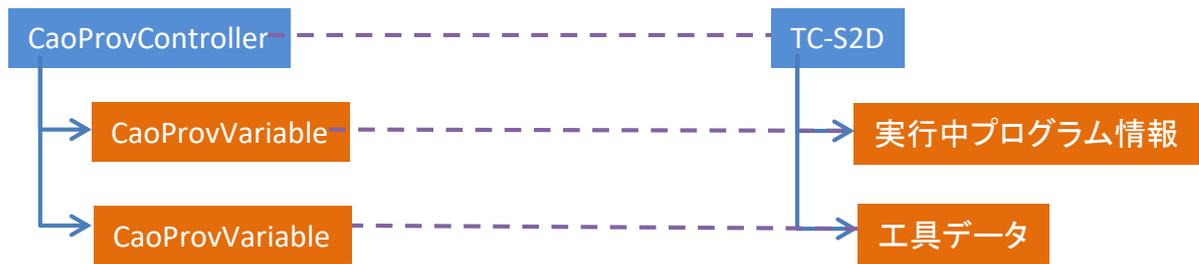


図 1-2 Brother TC Protocol2 プロバイダと TC-S2D のデータ対応図

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タッピングセンタ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

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, LabVIEW, Delphi 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、「ORiN 2 プログラミングガイド」を参照ください。

「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。

タッピングセンタの通信仕様に関しては、ブラザー工業株式会社より入手してください。

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. 接続方法

2.2. PC 環境のセットアップ

2.2.1. Brother TC Protocol2 プロバイダの自動インストール

ORiN2 SDK がインストールされている環境であれば、タッピングセンタに接続するための動作環境（ラインタイム）の準備は完了です。

開発環境のセットアップは別途、Microsoft Visual Studio 6.0, 2003/2005/2008/2010, LabVIEW など 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をご準備してください。

2.2.2. Brother TC Protocol2 プロバイダの手動インストール

インストーラを使用せずにセットアップを行う場合は手作業で下記レジストリ登録を行う必要があります。

表 2-1 Brother TC Protocol2 プロバイダ

ファイル名	CaoProvBrotherTCProtocol2.dll
ProgID	CaoProv.Brother.TCProtocol2
レジストリ登録	regsvr32 CaoProvBrotherTCProtocol2.dll
レジストリ登録の抹消	regsvr32 /u CaoProvBrotherTCProtocol2.dll

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。

3. Brother TC Protocol2 プロバイダによるプログラミング

Brother TC Protocol2 プロバイダでは、以下の手順で PC とタッピングセンタを接続することができます。

- CaoEngine の作成
- CaoWorkspace の作成
- CaoController の作成

タッピングセンタに接続した後は、CaoController の Execute メソッドを使用する、もしくは、CaoVariable オブジェクトを生成することで、タッピングセンタ内の情報にアクセスすることができます。

3.1. 工具データにアクセスするサンプルプログラミング

ここでは例としてデータバンクの工具データのデータを読み込み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 192. 168. 1. 2
	接続先ポート番号は 10000
処理内容	工具データのデータ番号 1 からデータを読み込む。

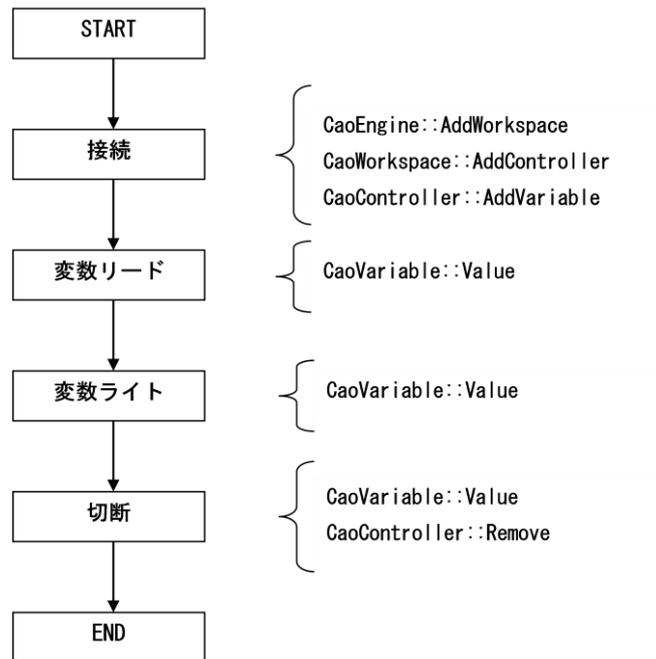


図 3-1 工具データアクセスの流れ

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_TOOL_DATA.vb
--------	-------------------

```

Sub Main()
  ' オブジェクト
  Dim engine As CaoEngine
  Dim workspace As CaoWorkspace
  Dim ctrl As CaoController
  Dim varTool1 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("BrotherTC", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "conn=eth:192.168.1.2, timeout=3000")

  ' データ番号1にアクセスするCaoVariableをCaoControllerに追加
  
```

```

Set varTool1 = ctrl.AddVariable("TOOL_DATA_1", "REFRESH=True, UNIT=1, No=1")

' Tool1 から値を読み込む
Dim var As Variant
var = varTool1.value

' CaoController から CaoVariable をクリアする
Call ctrl.Variables.Clear
' CaoVariable の消去
Set varTool1 = 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
End Sub

```

3.1.1.1. 接続

タッピングセンタとの接続を行うには、以下の手順を取ります。

- (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 varTool1 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。Brother TC Protocol2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("BrotherTC", _
    "GaoProv. Brother. TCProtocol2", _
    "", _
    "conn=eth:192.168.1.2, timeout=3000")
```

- (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。以下に工具データ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 varTool1 = ctrl.AddVariable("TOOL_DATA_1", "REFRESH=True, UNIT=1, No=1")
```

3.1.1.2. 工具データの取得

接続した工具データの取得するには、CaoVariableオブジェクトのValueプロパティを参照します。工具データの取得する場合は、Variant型変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 var As Variant
var = varTool1.value
```

’ 値の取得

3.1.1.3. 切断

コントローラと切断する場合には、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 varTool1 = 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. コマンド一覧

表 4-1 コマンド一覧

カテゴリ	メソッド/プロパティ	機能	参照
CaoWorkspace	AddController	コントローラに接続	P. 12
CaoController	AddVariable	ユーザ・システム変数に接続	P. 14
	GetVariableNames	ユーザ・システム変数の一覧の取得	P. 15
	Execute	コントローラクラスに実装されているコマンドの実行	P. 15
CaoVariable	get_Value	値の取得	P. 19

4.2. メソッド・プロパティ

4.2.1. CaoWorkspace::AddController メソッド

Controller オブジェクトの生成時を行います。

各オプションには、デバイスに接続するための情報を指定します。

以下に、AddController の仕様を示します。

書式

AddController

```
(
    "<コントローラ名>",           'コントローラ名(任意)'
    "CaoProv. Brother. TCProtocol2", '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

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

' シリアル通信を使用してタッピングセンタと接続する場合

```
Set ctrl = workspace.AddController("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=COM:2, Timeout=500")
```

' TCP/IP 通信を使用してタッピングセンタと接続する場合

```
Set ctrl = workspace.AddController("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
```

以下にオプション文字列に指定する文字列を示します。

表 4-2 CaoWorkspace::AddController のオプション文字列

オプション	必須	説明	値範囲	デフォルト値
CONN= COM:<接続先 COM ポート番号>[:<ボーレート>[:<パリティ>:<データビット数>:<ストップビット数>]]	○ ¹	シリアル通信を使用してタッピングセンタと接続する場合指定してください。 PC 側で使用できるシリアルポート番号を指定します。	-	ボーレート: 9600 パリティ: NONE データビット数: 7bit ストップビット数: 1bit
CONN= ETH:<接続先 IP>[:<接続先ポート>]	○ ¹	TCP/IP 通信を使用してタッピングセンタと接続する場合指定してください。 接続先の IP アドレスおよびポート番号を指定します。	-	ポート:10000
Timeout=<送受信タイムアウト時間>	-	送受信タイムアウト時間 [ms] を設定します。	1 - 65535	500

¹ Com, ETH のどちらか必須.

オプション	必須	説明	値範囲	デフォルト値
KeepaliveTime=<キープアライブ待機時間>	⌋ ²	キープアライブ待機時間 [ms] を設定します。	1 - 4294967295	10000
KeepaliveInterval=<キープアライブ周期時間>	⌋ ²	キープアライブ周期時間 [ms] を設定します。	1 - 4294967295	2000

4.2.2. CaoController::AddVariable メソッド

CaoController から CaoVariable オブジェクトを生成します。変数名には、表 4-4 に示す変数名のみ使用できます。

以下に、AddVariable の仕様を示します。

書式

AddVariable

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

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")

' データ番号 1 にアクセスする CalVariable を CaoController に追加
Dim varTool1 As CaoVariable
Set varTool1 = ctrl.AddVariable("TOOL_DATA_1", "REFRESH=True, UNIT=1, No=1")
```

² ETH の時のみ

4.2.3. CaoController::GetVariableNames メソッド

AddVariable メソッドで指定できる変数名とシステム変数名の一覧を取得します。

4.2.4. CaoController::Execute メソッド

Controller にてメソッドを実行します。

引数には、メソッドに対して必要な引数を指定します。

以下に、Execute の仕様を示します。

書式

Execute

```
(
    "＜メソッド名＞", 'メソッド名
    "＜引数＞"      '引数
)
```

以下に、Execute で指定できるメソッド一覧を示します。使用例は各メソッドの詳細で記述しています。

表 4-3 CaoController::Execute メソッド一覧

メソッド名	説明	コマンド種別+機能
GetProductionData4	「機械モニタ用データ（生産データ 4）」ファイルデータを取得する。	LOD
GetMemoryOperationData	「メモリ運転データ」ファイルデータを取得する。	LOD
GetProductionData2	「生産データ 2(時間表示)」ファイルデータを取得する。	LOD
GetToolData	「工具データ」ファイルデータを取得する。	LOD
GetMacroVariable	「マクロ変数」ファイルデータを取得する。	LOD

4.2.4.1. GetProductionData4

「機械モニタ用データ（生産データ 4）」ファイルデータ内のデータを取得します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_ARRAY VT_VARIANT		
	0	VT_BSTR	運転プログラム番号が返却されます。
	1	VT_BSTR	編集プログラム番号が返却されます。

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
' 「機械モニタ用データ（生産データ 4）」ファイルデータを取得
Dim vntProductionData4 As Variant
vntProductionData4 = ctrl.Execute("GetProductionData4", Array(True))
```

4.2.4.2. GetMemoryOperationData

「メモリ運転データ」ファイルデータ内のデータを取得します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_ARRAY VT_VARIANT		
	0	VT_BSTR	運転プログラム番号が返却されます。
	1	VT_I4	運転状態が返却されます。 (0:リセット, 1:運転, 2:一時停止 3:ブロック停止)
	2	VT_I4	内側パレット状態が返却されます。 (0:未割出, 1:内側 1 番, 2:内側 2 番)
	3	VT_I4	予備工具が返却されます。 (0:使用していない, 1:使用中)
	4	VT_I4	モードが返却されます。
5	VT_I4	拡張が返却されます。	

(0:通常, 1:拡張)

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
' 「メモリ運転データ」ファイルデータを取得
Dim vntMemoryOperationData As Variant
vntMemoryOperationData = ctrl.Execute("GetMemoryOperationData", Array(True))

```

4.2.4.3. GetProductionData2

「生産データ 2(時間表示)」ファイルデータ内のデータを取得します。

項目	型説明			
引数	VT_ARRAY VT_VARIANT			
	0	VT_BOOL	データリフレッシュ実行有無を指定します。	
戻り値	VT_ARRAY VT_VARIANT			
	i (0-19)	VT_ARRAY VT_VARIANT	稼働時間記録 i+1	
		0	VT_BSTR	年月日が返却されます。
		1	VT_BSTR	稼働時間が返却されます。
		2	VT_I4	運転回数が返却されます。
	3	VT_R8	稼働率が返却されます。	

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
' 「生産データ 2(時間表示)」ファイルデータを取得
Dim vntProductionData2 As Variant
vntProductionData2 = ctrl.Execute("GetProductionData2", Array(True))

```

4.2.4.4. GetToolData

「工具データ」ファイルデータ内の工具データを取得します。

項目	型説明			
引数	VT_ARRAY VT_VARIANT			
	0	VT_BOOL	データリフレッシュ実行有無を指定します。	
	1	VT_UI1	単位系を指定します。(1:メトリック, 2:インチ)	
	2	VT_UI1	データ番号を指定します。(1~10:データ番号)	
戻り値	VT_ARRAY VT_VARIANT			
	i (0-98)	VT_ARRAY VT_VARIANT		工具番号 i+1
		0	VT_BSTR	工具名が返却されます。
		1	VT_R8	工具長オフセットが返却されます。
		2	VT_R8	工具長摩耗補正が返却されます。
		3	VT_R8	工具径補正が返却されます。
		4	VT_R8	工具径摩耗補正が返却されます。
		5	VT_I4	寿命単位が返却されます。
		6	VT_I4	初期寿命/終了寿命が返却されます。
		7	VT_I4	予告寿命が返却されます。
8	VT_I4	寿命が返却されます。		

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
' 「工具データ」ファイルデータのメトリック, データ番号1を取得
Dim vntToolData As Variant
vntToolData = ctrl.Execute("GetToolData", Array(True, 1, 1))
```

4.2.4.5. GetMacroVariable

「マクロ変数」ファイルデータ内のマクロ変数(#500~#999)を取得します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_BOOL	データリフレッシュ実行有無を指定します。
	1	VT_UI1	単位系を指定します。(1:メトリック, 2:インチ)

	2	VT_UI1	データ番号を指定します。(1~10:データ番号)
戻り値	VT_ARRAY VT_R8		
	0	VT_R8	マクロ変数(#500)が返却されます。
	~	~	~
	499	VT_R8	マクロ変数(#999)が返却されます。

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
' 「マクロ変数」ファイルデータのメトリック、データ番号1を取得
Dim vntMacroVariable As Variant
vntMacroVariable = ctrl.Execute("GetMacroVariable", Array(True, 1, 1))
```

4.2.5. CaoVariable::get_Value プロパティ

オブジェクトに対応している変数の値を取得します。

変数の実装状況およびデータ型は「表 4-4 コントローラクラス 変数一覧」を参照してください。

4.3. 変数一覧

4.3.1. コントローラクラス

CaoController クラスで使用できる変数一覧を以下に示します。

表 4-4 コントローラクラス 変数一覧

変数名	データ型	説明	Value	
			get	put
@MAKER_NAME	VT_BSTR	メーカー名の取得	○	-
@VERSION	VT_BSTR	DLL のバージョンの取得	○	-
PROGRAM_NUMBER<??>	VT_ARRAY VT_VARIANT	プログラム番号を取得	○	-
MEMORY<??>	VT_ARRAY VT_VARIANT	メモリ運転データを取得	○	-
OPERATING_TIME<??>	VT_ARRAY VT_VARIANT	稼働時間記録を取得	○	-
TOOL_DATA<??>	VT_ARRAY VT_VARIANT	工具データの取得	○	-
MACRO<??>	VT_ARRAY VT_R8	マクロ変数(#500~#999)の	○	-

		取得		
--	--	----	--	--

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
' 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

項目	型説明	
戻り値	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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
' 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

項目	型説明	
戻り値	VT_BSTR	DLL のバージョンを取得します。

4.3.1.3. PROGRAM_NUMBER<??>

「機械モニタ用データ（生産データ 4）」ファイルデータ内のプログラム番号を取得します。

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("PROGRAM_NUMBER", "REFRESH=True")
' 値取得
Dim value As Variant
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
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

項目	型説明		
戻り値	VT_ARRAY VT_VARIANT		
	0	VT_BSTR	運転プログラム番号が返却されます。
	1	VT_BSTR	編集プログラム番号が返却されます。

● 対応コマンド

コマンド種別+機能	LOD

4.3.1.4. MEMORY<??>

「メモリ運転データ」ファイルデータ内のデータを取得します。

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")

' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("MEMORY", "REFRESH=True")

' 値取得
Dim value As Variant
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
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

項目	型説明		
戻り値	VT_ARRAY VT_VARIANT		
	0	VT_BSTR	運転プログラム番号が返却されます。
	1	VT_I4	運転状態が返却されます。 (0:リセット, 1:運転, 2:一時停止 3:ブロック停止)
	2	VT_I4	内側パレット状態が返却されます。 (0:未割出, 1:内側1番, 2:内側2番)
	3	VT_I4	予備工具が返却されます。 (0:使用していない, 1:使用中)
	4	VT_I4	モードが返却されます。
	5	VT_I4	拡張が返却されます。 (0:通常, 1:拡張)

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.5. OPERATING_TIME<??>

「生産データ 2(時間表示)」ファイルデータ内の稼働時間記録を取得します。

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")

' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("OPERATING_TIME", "REFRESH=True")

' 値取得
Dim value As Variant
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
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

項目	型説明			
戻り値	VT_ARRAY VT_VARIANT			
	i (0-19)	VT_ARRAY VT_VARIANT		稼働時間記録 i+1
		0	VT_BSTR	年月日が返却されます。
		1	VT_BSTR	稼働時間が返却されます。
		2	VT_I4	運転回数が返却されます。
	3	VT_R8	稼働率が返却されます。	

● 対応コマンド

コマンド種別+機能	LOD

4.3.1.6. TOOL_DATA<??>

「工具データ」ファイルデータ内の工具データを取得します。

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")

' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("TOOL_DATA_M_1", "REFRESH=True, UNIT=1, NO=1")

' 値取得
Dim value As Variant
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
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-
UNIT= <単位系を指定>	-	ファイルデータの単位系を指定し ます。(1:メトリック, 2:インチ)	1 - 2	1
DATA_NO= <データ番号を指定>	-	ファイルデータのデータ番号を指 定します。(1~10:データ番号)	1 - 10	1

● データ型

項目	型説明		
戻り値	VT_ARRAY VT_VARIANT		
	i	VT_ARRAY VT_VARIANT	工具番号 i+1
	(0-98)	0	VT_BSTR 工具名が返却されます。
		1	VT_R8 工具長オフセットが返却されます。
		2	VT_R8 工具長摩耗補正が返却されます。
		3	VT_R8 工具径補正が返却されます。
		4	VT_R8 工具径摩耗補正が返却されます。
		5	VT_I4 寿命単位が返却されます。
		6	VT_I4 初期寿命/終了寿命が返却されます。

	7	VT_I4	予告寿命が返却されます.
	8	VT_I4	寿命が返却されます.

- 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.7. MACRO<??>

「マクロ変数」ファイルデータ内のマクロ変数(#500~#999)を取得します。

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("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("MACRO_M_1", "REFRESH=True, UNIT=1, NO=1")
' 値取得
Dim value As Variant
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
REFRESH= 〈ファイルデータ更新有無を指定〉	○	ファイルデータ更新処理の有無を指定します。	0 - 1	-
UNIT= 〈単位系を指定〉	-	ファイルデータの単位系を指定します。(1:メトリック, 2:インチ)	1 - 2	1
DATA_NO= 〈データ番号を指定〉	-	ファイルデータのデータ番号を指定します。(1~10:データ番号)	1 - 10	1

● データ型

項目	型説明		
戻り値	VT_ARRAY VT_R8		
	0	VT_R8	マクロ変数(#500)が返却されます。
	~	~	~
	499	VT_R8	マクロ変数(#999)が返却されます。

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

付録A. エラーコード

本プロバイダには、独自のエラーコードが存在します。詳細は以下の表 4-5 独自エラーコード表をご参照ください。

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ご参照ください。

表 4-5 独自エラーコード表

エラー番号	説明
0x80110003	受信したメッセージのチェックサムが不正です。
0x80110004	ファイルデータが未取得の状態でのデータの取得を行いました。

また、本プロバイダは、TC Protocol2 で通信を行う機器 (TC-S2D 等) からの完了コードを "0x8010****" でマスクして返します。

完了コードについては、brother 社の「操作説明書(データ)」をご参照ください。

付録B. ファイルデータの読み込みの仕組み

TC 専用通信プロトコル方式 2 では、対象デバイスの保持しているファイルデータを取得するコマンドがあります。本プロバイダではコマンド種別+機能が「LOD」となっているメソッド、変数が対応しています。

表 4-6 ファイルデータ - メソッド, 変数対応表

ファイルデータ	メソッド	変数
MONTR	GetProductionData4	PROGRAM_NUMBER<??>
MEM	GetMemoryOperationData	MEMORY<??>
PRD2	GetProductionData2	OPERATING_TIME<??>
TOLN	GetToolData	TOOL_DATA<??>
MCRS	GetMacroVariable	MACRO<??>

上記メソッド、変数では、本プロバイダを使用している PC のメモリ上に対象のファイルデータを保持し、そこから値を取得します。一度も取得していないファイルデータは PC のメモリ上に保持されていないため、値を取得しようとするとうエラーとなります。

PC のメモリ上に保持されるデータは、データを更新するメソッド(Refresh**)を使用することで更新することができます。また、ファイルデータの一部、もしくは全部を取得するメソッドの引数、変数のオプションで取得時のファイルデータの更新の有無を指定できます。

データの更新はファイルデータ単位で行うため、同一ファイルデータ内の別データの取得によって更新した場合でも、そこで更新したデータを元に値を出力します。

4.3.2. CaoVariable::put_Value プロパティ

オブジェクトに対応している変数に値を設定します。

変数の実装状況およびデータ型は「表 4-4 コントローラクラス 変数一覧」を参照してください。