

# Brother TC Protocol2 provider

**Version 1.1.3**

**User's guide**

**March 23, 2020**

Remarks:

- This document uses the machine translation.
- **There is a possibility that the hardware of equipment breaks down when this provider is used.**

**【 revision history 】**

Version	Date	Content
1.0.0	2018-05-07	First edition.
1.1.0	2018-07-05	Keep alive addition
1.1.1	2018-07-13	Variable identifier change
1.1.2	2018-09-18	Trouble correction of receive data analysis
1.1.3	2020-03-23	Fixed a problem when connecting multiple devices at the same time.

**【 connected confirmation ending equipment 】****■ Equipment name**

Model name	Firmware version	Notes
TC-S2D		

## Content

<b>1. Introduction.....</b>	<b>4</b>
1.1. Environment and version that this book assumes.....	5
1.2. Source that becomes reference .....	5
<b>2. Environmental setup for application development .....</b>	<b>6</b>
2.1. Connection method.....	6
2.2. Setup of PC environment .....	6
2.2.1. Automated install of Brother TC Protocol2 provider .....	6
2.2.2. Manual installation of Brother TC Protocol2 provider .....	6
<b>3. Programming by Brother TC Protocol2 provider.....</b>	<b>7</b>
3.1. Sample programming accessed tool data.....	7
3.1.1. [Sanpurupurogumu] .....	8
<b>4. Command reference .....</b>	<b>12</b>
4.1. Command list.....	12
4.2. Method property.....	12
4.2.1. CaoWorkspace::AddController method .....	12
4.2.2. CaoController::AddVariable method .....	14
4.2.3. CaoController::GetVariableNames method .....	16
4.2.4. CaoController::Execute method.....	16
4.2.5. CaoVariable::get_Value property .....	20
4.3. Variable list.....	20
4.3.1. Controller class .....	20
4.3.2. CaoVariable::put_Value property .....	34

## 1. Introduction

This book is an user' s guide of the Brother TC Protocol2 provider that reads data by using communication protocol method 2 only for TC for the Topping center made by the brother company CNC (TC-S2D etc.), and writes it. The underFigure1-1However, it becomes a whole of device block diagram with [hon] provider.

The Topping center is named a Topping center at following brother.

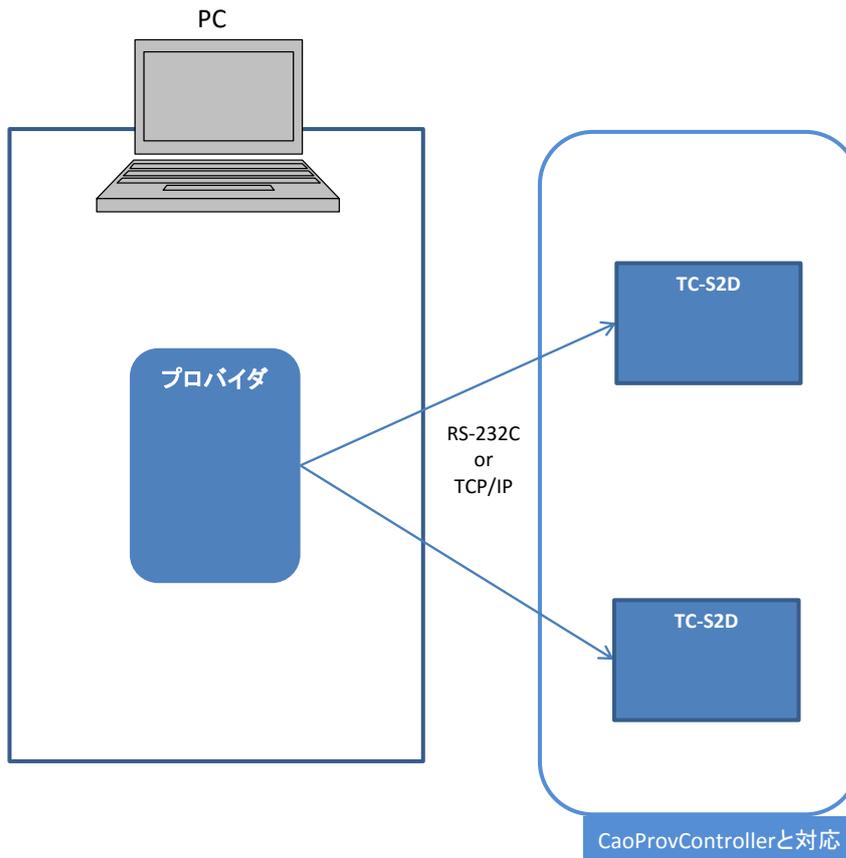


Figure1-1Whole block diagram

Moreover, figure where this provider and correspondence each device were shown is the under. Figure1-2It becomes it.

(\* It is one example. It is not because everything is shown. )

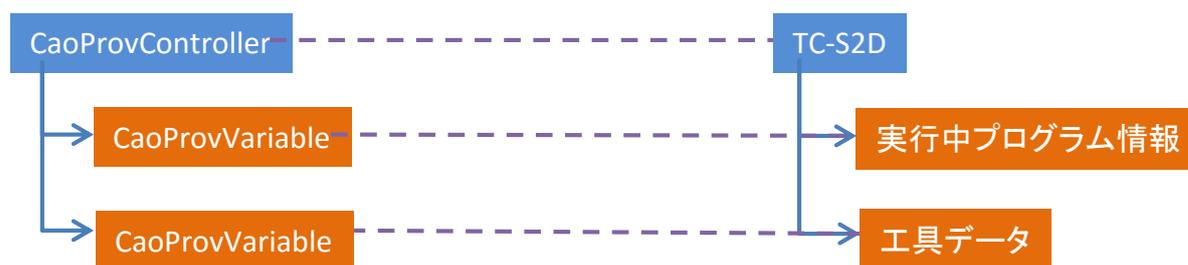


Figure1-2Brother TC Protocol2 provider and figure for data of TC-S2D

### 1.1. Environment and version that this book assumes

Client PC operates on Windows, and the environment whose targeted CNC is possible Ethernet connection's Topping center is assumed. The development setting of PC can be developed in case of the programming environment that supports Component Object Model (COM and Component Object Model).

### 1.2. Source that becomes reference

C++ and Java though all the programming cases with this book have been described with Visual Basic 6.0. It is possible to develop in various program languages such as NET, LabVIEW, and Delphi. Refer to "ORiN 2 programming guide" for use.

"ORiN 2 programming guide" corresponds to the following files of the ORiN2 SDK installation folder.

- ・ ORiN2¥CAO¥Doc¥ORiN2\_ProgrammersGuide\_<lang>.pdf
- Replace with the language character string of each environment and read the part of < lang >.

ORiN2 needed in developing the application that uses the provider and COM/DCOM are explained while exchanging the example for basic knowledge and the technology.

Obtain the communication specification at the Topping center from Brother Industries, Ltd..

## 2. Environmental setup for application development

### 2.1. Connection method

### 2.2. Setup of PC environment

#### 2.2.1. Automated install of Brother TC Protocol2 provider

If it is an environment in which ORiN2 SDK is installed, the preparation for the system requirements (line time) to connect it with the Topping center is completed.

Prepare the programming environment that separately supports Component Object Model (COM and Component Object Model) such as Microsoft Visual Studio 6.0, 2003/2005/2008/2010, and LabVIEW for the setup of the development setting.

#### 2.2.2. Manual installation of Brother TC Protocol2 provider

It is necessary to register the following registry by the hand work when setting it up without using the installer.

Table2-1Brother TC Protocol2 provider

File name	CaoProvBrotherTCProtocol2.dll
ProgID	CaoProv.Brother.TCProtocol2
Registry registration	regsvr32 CaoProvBrotherTCProtocol2.dll
Blotting out of registry registration	regsvr32 /u CaoProvBrotherTCProtocol2.dll

If one regular ORiN2 SDK license of each PC is not registered beforehand so that the CAO engine may work, it doesn't become it. Refer to the paragraph of "Addition and deletion of the license" in the ORiN2 SDK user's guide.

### 3. Programming by Brother TC Protocol2 provider

In the Brother TC Protocol2 provider, PC can be connected with the Topping center according to the following procedures.

- Making of CaoEngine
- Making of CaoWorkspace
- Making of CaoController

After it connects it with the Topping center, it can access information in the Topping center by using the Execute method of CaoController or generating the CaoVariable object.

#### 3.1. Sample programming accessed tool data

The sample program that reads the data of the tool data of the data bank as an example is shown here. Table3-1The requirement for [ni] sample programFigure3-1The flow of [ni] sample program is described respectively.

Table3-1Requirement for sample program

Requirement	Explanation
Connection destination	Connect it by TCP/IP.
	Connection destination Internet Protocol address is 192.168.1.2.
	The connection destination port number is 10000.
Content of processing	Read data from the data number 1 of the tool data.

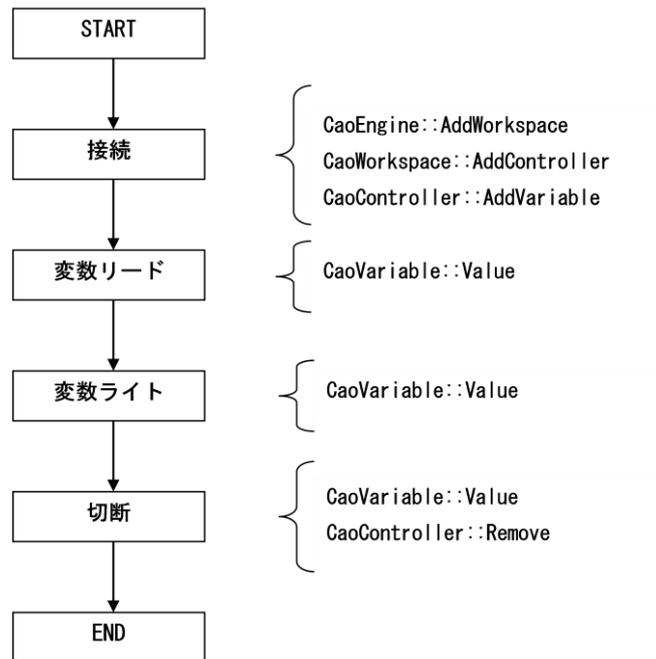


Figure3-1Flow of tool data access

A concrete code is shown from the following paragraphs.

### 3.1.1. [Sanpurupurogumu]

The whole image of the sample program is shown as follows.

#### Sample READ\_TOOL\_DATA.vb

```

Sub Main()
' object
Dim engine As CaoEngine
Dim workspace As CaoWorkspace
Dim ctrl As CaoController
Dim varTool1 As CaoVariable

' generation of CaoEngine object
Set engine = New CaoEngine
' generation of CaoWorkspace object
Set workspace = engine.AddWorkspace("NewWrks", "")
' generation of CaoController object
Set ctrl = workspace.AddController("BrotherTC", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "conn=eth:192.168.1.2, timeout=3000")

' Add CaIVarialbe accessed data number 1 to CaoController.
  
```

---

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

```
' read the value from Tool1.
```

```
Dim var As Variant
var = varTool1.value
```

```
' clear CaoVariable from CaoController.
```

```
Call ctrl.Variables.Clear
```

```
' deletion of CaoVariable
```

```
Set varTool1 = Nothing
```

```
' delete CaoController from CaoWorkspace.
```

```
Call workspace.Controllers.Remove(ctrl.Index)
```

```
' deletion of CaoController
```

```
Set ctrl = Nothing
```

```
' delete CaoWorkspace from CaoEngine.
```

```
Call engine.Workspaces.Remove(workspace.Index)
```

```
' deletion of CaoWorkspace
```

```
Set workspace = Nothing
```

```
' deletion of CaoEngine
```

```
Set engine = Nothing
```

```
End Sub
```

---

### 3.1.1.1. Connection

Take the following procedures to connect it with the Topping center.

- (1) Prepare the variable to maintain the object. The object necessary for the controller connection is CaoEngine object, CaoWorkspace object, and CaoController object. When the CaoController object is acquired from CaoWorkspaces, the CaoWorkspace object need not prepare the variable. Moreover, the CaoVariable object to access the variable is needed. The example of the code in VB6 is shown as follows.

---

```
Dim engine As caoEngine      ' variable for CaoEngine object
Dim workspace As caoWorkspace ' variable for CaoWorkspace object
Dim ctrl As CaoController    ' variable for CaoController object
Dim varTool1 As CaoVariable  ' variable for CaoVariable object
```

---

(2) Generate the CaoEngine object. The CaoEngine object uses and generates the New key word.

---

```
' generation of CaoEngine object
Set engine = New caoEngine
```

---

(3) Acquire the CaoWorkspace object or generate it. When the CaoEngine object is generated, the CaoWorkspaces object and the CaoWorkspace object are generated with default one by one. The example of the code of newly generating the CaoWorkspace object and CaoWorkspace of default are shown as follows.

---

```
' generation of CaoWorkspace object
Set workspace = engine.AddWorkspace("NewWrks", "")
```

---

(4) Generate the CaoController object. Set the CaoController object and set the provider name used and the parameter to use it to generate it. In the Brother TC Protocol2 provider, specify Internet Protocol address of the controller of the connection destination in the option. The example of the code is shown as follows.

---

```
' generation of CaoController object
Set ctrl = workspace.AddController("BrotherTC", _
                                "CaoProv.Brother.TCProtocol2", _
                                "", _
                                "conn=eth:192.168.1.2, timeout=3000")
```

---

(5) Generate the CaoVariable object. Generate the CaoVariable object of the variable to be connected. The example of the code of generating the variable object accessed the tool data is shown as follows.

---

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

---

### 3.1.1.2. Acquisition of tool data

Refer for acquisition to the Value property of the CaoVariable object for the connected tool data. When the tool data acquires it, it is necessary to prepare the Variant type variable. The example of the code is shown as follows.

---

```
Dim var As Variant
var = varTool1.value ' acquisition of value
```

---

### 3.1.1.3. Cutting

Delete the generated object, and delete the object deleted from the collection class that manages the object when cutting it with the controller. The example of the code is shown as follows.

---

```
' clear CaoVariable from CaoController.
Call ctrl.Variables.Clear
' deletion of CaoVariable
Set varTool1 = Nothing
' delete CaoController from CaoWorkspace.
Call workspace.Controllers.Remove(ctrl.Index)
' deletion of CaoController
Set ctrl = Nothing
' delete CaoWorkspace from CaoEngine.
Call engine.Workspaces.Remove(workspace.Index)
' deletion of CaoWorkspace
Set workspace = Nothing
' deletion of CaoEngine
Set engine = Nothing
```

---

## 4. Command reference

### 4.1. Command list

Table4-1Command list

Category	Method/property	Function	Reference
CaoWorkspace	AddController	Connect it with the controller.	P. 12
CaoController	AddVariable	Connect it with the user system variable.	P. 14
	GetVariableNames	Acquisition of list of user system variable	P. 16
	Execute	Execution of command being mounted by controller class	P. 16
CaoVariable	get_Value	Acquisition of value	P. 20

### 4.2. Method property

#### 4.2.1. CaoWorkspace::AddController method

Generate the Controller object.

Optional, each specify information to connect it with the device.

The specification of AddController is shown as follows.

#### Format

#### AddController

```
(
    "< controller name >"          ' Controller name (arbitrariness)
    "GaoProv. Brother. TCProtocol2", ' Provider name (fixation)
    "< machine name >"            ' Provider execution machine name (unused)
    "< option >"                  ' Optional character string
)
```

**Usage example**

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

' when connecting it with the Topping center by using serial communications
Set ctrl = workspace.AddController("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=COM:2, Timeout=500")

' when connecting it with the Topping center by using the TCP/IP communication
Set ctrl = workspace.AddController("Controller", _
    "CaoProv. Brother. TCProtocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")
```

The character string specified for an optional character string is shown as follows.

Table4-2Optional character string of CaoWorkspace::AddController

Option	Indispensability	Explanation	Range of value	Default value
Conn=com:<port no> [<baud rate>: <parity>: <data bit>: <stop bit>]	✓ <sup>1</sup>	Specify it when connecting it with the Topping center by using serial communications. Specify the serial port number that can be used on the PC side.	-	Baud rate: 9600 Parity: NONE data bits: 7bit stop bits: 1bit
Conn=eth:<IP address> [:<port>]	✓ <sup>1</sup>	Specify it when connecting it with the Topping center by using the TCP/IP communication. Specify Internet Protocol address and the port number of the connection destination.	-	Port: 10000
Timeout = <Sending and receiving time-out	-	Set sending and receiving time-out time ms.	1 - 65535	500

<sup>1</sup> com or eth option required.

Option	Indispensability	Explanation	Range of value	Default value
time>				
KeepaliveTime=< Keep alive standby time >	-2	Set keep alive standby time ms.	1 - 42949672 95	10000
KeepaliveInterval=< Interval time of Keep alive>	-2	Set keep alive time of cycle ms.	1 - 42949672 95	2000

#### 4.2.2. CaoController::AddVariable method

Create CaoVariable object from CaoController. For the variable name, specify the variable name shown in Table 4-4.

The specification of AddVariable is shown as follows.

##### Format

##### AddVariable

```
(
    "< variable identifier >",      ' variable identifier
    "< option >"                  ' Optional character string
)
```

##### Usage example

```
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")
```

Add ' CalVarialbe accessed data number 1 to CaoController.

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

<sup>2</sup> Only required when using the eth option



### 4.2.3. CaoController::GetVariableNames method

Acquire the list of the variable identifier and the system variable identifier that can be specified by the AddVariable method.

### 4.2.4. CaoController::Execute method

Execute the method with Controller.

Specify a necessary argument for the argument for the method.

The specification of Execute is shown as follows.

#### Format

#### Execute

```
(
    "< method name >",      '      Method name
    "< argument >"         '      argument
)
```

The method list that can be specified with Execute is shown as follows. Each method is detailed and the usage example is described.

Table4-3CaoController::Execute method list

Method name	Explanation	Command type + function
GetProductionData4	Acquire "Data (production data 4) for the machine monitor" file data.	LOD
GetMemoryOperationData	Acquire "Memory driving data" file data.	LOD
GetProductionData2	Acquire "Production data 2(time display)" file data.	LOD
GetToolData	Acquire "Tool data" file data.	LOD
GetMacroVariable	Acquire "Macro variable" file data.	LOD

#### 4.2.4.1. GetProductionData4

Acquire data in "Data (production data 4) for the machine monitor" file data.

Item	Type explanation		
Argument	VT_ARRAY   VT_VARIANT		
	0	VT_BOOL	Specify the data refreshing execution existence.
Return value	VT_ARRAY   VT_VARIANT		
	0	VT_BSTR	The driving program number is returned.
	1	VT_BSTR	The editor program number is returned.

#### Usage example

```

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")
' acquire "Data (production data 4) for the machine monitor" file data.
Dim vntProductionData4 As Variant
vntProductionData4 = ctrl.Execute("GetProductionData4", Array(True))
    
```

#### 4.2.4.2. GetMemoryOperationData

Acquire data in "Memory driving data" file data.

Item	Type explanation		
Argument	VT_ARRAY   VT_VARIANT		
	0	VT_BOOL	Specify the data refreshing execution existence.
Return value	VT_ARRAY   VT_VARIANT		
	0	VT_BSTR	The driving program number is returned.
	1	VT_I4	Driving is returned. (0: Reset and 1: Driving and 2: Temporary stop 3: Block stop)
	2	VT_I4	The state of an inside palette is returned. 1 .. un. . . (0:.. crack and going out. . : the 1st side and 2: the 2nd on the side. )
	3	VT_I4	A preliminary tool is returned. (0: 1 not used: Be using it. )

	4	VT_I4	The mode is returned.
	5	VT_I4	Enhancing is returned. (0: 1 usually: Enhancing)

**Usage example**

```

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")
' acquire "Memory driving data" file data.
Dim vntMemoryOperationData As Variant
vntMemoryOperationData = ctrl.Execute("GetMemoryOperationData", Array(True))
    
```

**4.2.4.3. GetProductionData2**

Acquire data in "Production data 2(time display)" file data.

Item	Type explanation			
Argument	VT_ARRAY   VT_VARIANT			
	0	VT_BOOL	Specify the data refreshing execution existence.	
Return value	VT_ARRAY   VT_VARIANT			
	i (0-19)	VT_ARRAY   VT_VARIANT		Record i+1 at operating time
		0	VT_BSTR	The date is returned.
		1	VT_BSTR	Operating time is returned.
		2	VT_I4	The driving frequency is returned.
3		VT_R8	Utilization rates is returned.	

**Usage example**

```

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")
' acquire "Production data 2(time display)" file data.
Dim vntProductionData2 As Variant
    
```

---

```
vntProductionData2 = ctrl.Execute("GetProductionData2", Array(True))
```

---

#### 4.2.4.4. GetToolData

Acquire the tool data in "Tool data" file data.

Item	Type explanation			
Argument	VT_ARRAY   VT_VARIANT			
	0	VT_BOOL	Specify the data refreshing execution existence.	
	1	VT_UI1	Specify the unit system. (1: Metric and 2: Inch)	
	2	VT_UI1	Specify the data number. (1-10: Data number)	
Return value	VT_ARRAY   VT_VARIANT			
	i (0-98)	VT_ARRAY   VT_VARIANT		Tool number i+1
		0	VT_BSTR	The tool name is returned.
		1	VT_R8	The tool length offset is returned.
		2	VT_R8	The wear-out of the length of the tool correction is returned.
		3	VT_R8	The cutter compensation is returned.
		4	VT_R8	The tool diameter wear-out correction is returned.
		5	VT_I4	The unit of longevity is returned.
		6	VT_I4	Initial longevity/end longevity is returned.
		7	VT_I4	The previous notice longevity is returned.
8	VT_I4	Longevity is returned.		

#### Usage example

---

```
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")
' acquire metric and data number 1 of "Tool data" file data.
Dim vntToolData As Variant
vntToolData = ctrl.Execute("GetToolData", Array(True, 1, 1))
```

---

#### 4.2.4.5. GetMacroVariable

Acquire macro variable (#500-#999) in "Macro variable" file data.

Item	Type explanation		
Argument	VT_ARRAY   VT_VARIANT		
	0	VT_BOOL	Specify the data refreshing execution existence.
	1	VT_UI1	Specify the unit system. (1: Metric and 2: Inch)
	2	VT_UI1	Specify the data number. (1-10: Data number)
Return value	VT_ARRAY   VT_R8		
	0	VT_R8	Macro variable (#500) is returned.
	~	~	~
	499	VT_R8	Macro variable (#999) is returned.

#### Usage example

```
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")
' acquire metric and data number 1 of "Macro variable" file data.
Dim vntMacroVariable As Variant
vntMacroVariable = ctrl.Execute("GetMacroVariable", Array(True, 1, 1))
```

### 4.2.5. CaoVariable::get\_Value property

Acquire the value of the variable that corresponds to the object.

The mounting situation and the data type of the variable are "Table4-4Refer to".

## 4.3. Variable list

### 4.3.1. Controller class

The variable list that can be used in the CaoController class is shown below.

Table4-4Controller class variable list

Variable identifier	Data type	Explanation	Value	
			get	put
@MAKER_NAME	VT_BSTR	Acquisition of manufacturer name	✓	-

@VERSION	VT_BSTR	Acquisition of version of DLL	✓	-
PROGRAM_NUMBER<??>	VT_ARRAY   VT_VARIANT	Acquire the program number.	✓	-
MEMORY<??>	VT_ARRAY   VT_VARIANT	Acquire the memory driving data.	✓	-
OPERATING_TIME<??>	VT_ARRAY   VT_VARIANT	Acquire the record at operating time.	✓	-
TOOL_DATA<??>	VT_ARRAY   VT_VARIANT	Acquisition of tool data	✓	-
MACRO<??>	VT_ARRAY   VT_R8	Acquisition of macro variable (#500-#999)	✓	-

#### 4.3.1.1. @MAKER\_NAME

Acquire the manufacturer name.

##### Usage example

```

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")

```

##### Variable ' addition

```

Dim var As CaoVariable
Set var = ctrl.AddVariable("@MAKER_NAME")

```

##### Value ' acquisition

```

Dim strVal As String
strVal = var.value

```

#### ● Data type

Item	Type explanation	
Return value	VT_BSTR	Acquire the manufacturer name.

#### 4.3.1.2. @VERSION

Acquire the version of DLL.

##### Usage example

```
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")
```

##### Variable ' addition

```
Dim var As CaoVariable
Set var = ctrl.AddVariable("@VERSION")
```

##### Value ' acquisition

```
Dim value As String
value = var.value
```

- Data type

Item	Type explanation	
Return value	VT_BSTR	Acquire the version of DLL.

**4.3.1.3. PROGRAM\_NUMBER<??>**

Acquire the program number in "Data (production data 4) for the machine monitor" file data.

**Usage example**

```
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")
```

**Variable ' addition**

```
Dim var As CaoVariable
Set var = ctrl.AddVariable("PROGRAM_NUMBER", "REFRESH=True")
```

**Value ' acquisition**

```
Dim value As Variant
value = var.value
```

● Option

Option	Indispensability	Explanation	Range of value	Default value
REFRESH= <Specify the file data update existence >	-	Specify the presence of the file data update processing.	0 - 1	-

● Data type

Item	Type explanation		
Return value	VT_ARRAY   VT_VARIANT		
	0	VT_BSTR	The driving program number is returned.
	1	VT_BSTR	The editor program number is returned.

● Correspondence command

Command type + function	LOD
-------------------------	-----

4.3.1.4. MEMORY<??>

Acquire data in "Memory driving data" file data.

Usage example

```
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")
```

Variable ' addition

```
Dim var As CaoVariable
Set var = ctrl.AddVariable("MEMORY", "REFRESH=True")
```

Value ' acquisition

```
Dim value As Variant
value = var.value
```

● Option

Option	Indispensability	Explanation	Range of value	Default value
REFRESH= <Specify the file data update existence >	-	Specify the presence of the file data update processing.	0 - 1	-

● Data type

Item	Type explanation		
Return value	VT_ARRAY   VT_VARIANT		
	0	VT_BSTR	The driving program number is returned.
	1	VT_I4	Driving is returned. (0: Reset and 1: Driving and 2: Temporary stop 3: Block stop)
	2	VT_I4	The state of an inside palette is returned. 1..un... (0:... crack and going out... the 1st side and 2: the 2nd on the side. )
	3	VT_I4	A preliminary tool is returned. (0: 1 not used: Be using it. )
	4	VT_I4	The mode is returned.

---

	5	VT_I4	Enhancing is returned. (0: 1 usually: Enhancing)
--	---	-------	---

- Correspondence command

Command type + function	LOD
----------------------------	-----

### 4.3.1.5. OPERATING\_TIME<??>

Acquire the record of operating time in "Production data 2(time display)" file data.

**Usage example**

```
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")
```

**Variable ' addition**

```
Dim var As CaoVariable
Set var = ctrl.AddVariable("OPERATING_TIME", "REFRESH=True")
```

**Value ' acquisition**

```
Dim value As Variant
value = var.value
```

● Option

Option	Indispensability	Explanation	Range of value	Default value
REFRESH= <Specify the file data update existence >	-	Specify the presence of the file data update processing.	0 - 1	-

● Data type

Item	Type explanation			
Return value	VT_ARRAY   VT_VARIANT			
	i (0-19)	VT_ARRAY   VT_VARIANT		Record i+1 at operating time
		0	VT_BSTR	The date is returned.
		1	VT_BSTR	Operating time is returned.
		2	VT_I4	The driving frequency is returned.
	3	VT_R8	Utilization rates is returned.	

● Correspondence command

Command type + function	LOD
----------------------------	-----



### 4.3.1.6. TOOL\_DATA<??>

Acquire the tool data in "Tool data" file data.

**Usage example**

```
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")
```

**Variable ' addition**

```
Dim var As CaoVariable
Set var = ctrl.AddVariable("TOOL_DATA_M_1", "REFRESH=True, UNIT=1, NO=1")
```

**Value ' acquisition**

```
Dim value As Variant
value = var.value
```

● Option

Option	Indispensability	Explanation	Range of value	Default value
REFRESH= <Specify the file data update existence >	-	Specify the presence of the file data update processing.	0 - 1	-
UNIT= <Specify the unit system >	-	Specify the unit system of the file data. (1: Metric and 2: Inch)	1 ? 2	1
DATA_NO= <Specify the data number >	-	Specify the data number of the file data. (1-10: Data number)	1 ? 10	1

● Data type

Item	Type explanation			
Return value	VT_ARRAY   VT_VARIANT			
	i (0-98)	VT_ARRAY   VT_VARIANT		Tool number i+1
		0	VT_BSTR	The tool name is returned.
		1	VT_R8	The tool length offset is returned.
	2	VT_R8	The wear-out of the length of the tool	

			correction is returned.
	3	VT_R8	The cutter compensation is returned.
	4	VT_R8	The tool diameter wear-out correction is returned.
	5	VT_I4	The unit of longevity is returned.
	6	VT_I4	Initial longevity/end longevity is returned.
	7	VT_I4	The previous notice longevity is returned.
	8	VT_I4	Longevity is returned.

● Correspondence command

Command type + function	LOD
----------------------------	-----

4.3.1.7. MACRO<??>

Acquire macro variable (#500~#999) in "Macro variable" file data.

Usage example

```
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")
```

Variable ' addition

```
Dim var As CaoVariable
Set var = ctrl.AddVariable("MACRO_M_1", "REFRESH=True, UNIT=1, NO=1")
```

Value ' acquisition

```
Dim value As Variant
value = var.value
```

● Option

Option	Indispensability	Explanation	Range of value	Default value
REFRESH= <Specify the file data update existence >	-	Specify the presence of the file data update processing.	0 - 1	-
UNIT= <Specify the unit system >	-	Specify the unit system of the file data. (1: Metric and 2: Inch)	1 ? 2	1
DATA_NO= <Specify the data number >	-	Specify the data number of the file data. (1-10: Data number)	1 ? 10	1

● Data type

Item	Type explanation		
Return value	VT_ARRAY   VT_R8		
	0	VT_R8	Macro variable (#500) is returned.
	~	~	~
	499	VT_R8	Macro variable (#999) is returned.

- Correspondence command

Command type + function	LOD
----------------------------	-----

## Appendix A. Error code

This provider has its own error code. For details, refer to Table 4-5 Original error code table. Please refer to it.

Refer to the chapter of the error code of "ORiN2 programming guide" for the ORiN2 commonness error.

Table4-5Original error code table

Error number	Explanation
0x80110003	The checksum of the received message is illegal.
0x80110004	The file data acquired data while unacquired.

Moreover, this provider does the mask by "0x8010\*\*\*\*" and returns the completion code from the equipment (TC-S2D etc.) that communicates with TC Protocol2.

Refer to "Manual of operation (data)" of the brother Co. for the completion code.

## Appendix B. Mechanism of reading file data

In communication protocol method 2 only for TC, there is a command to acquire the file data that the object device maintains. The method and the variable whose command type + function is "LOD" in this provider correspond.

Table4-6Table for method of file data and variable

File data	Method	Variable
MONTR	GetProductionData4	PROGRAM_NUMBER<??>
MEM	GetMemoryOperationData	MEMORY<??>
PRD2	GetProductionData2	OPERATING_TIME<??>
TOLN	GetToolData	TOOL_DATA<??>
MCRS	GetMacroVariable	MACRO<??>

Maintain the file data of the object in the memory of PC that uses this provider, and acquire the value there in the method of the above-mentioned and the variable. When the value is acquired because the file data never acquired is not maintained in the memory of PC, it becomes an error.

The data maintained in the memory of PC can be updated by using method (Refresh\*\*) of updating data. Moreover, the presence of the update of the file data when the argument and the variable of the method of acquiring part or all of the file data are acquired in the option can be specified.

Output the value based on the data updated there even when updating it by acquiring another data in the same file data to update data in each file data.

### 4.3.2. CaoVariable::put\_Value property

Set the value to the variable that corresponds to the object.

The mounting situation and the data type of the variable are "Table4-4Refer to".