YAMAHA
RCX3 providers
User's Guide

Version 1.0.0

January 13, 2022

NOTE:

[Revision History]

Version	Date	Content
1.0.0	2022-01-13	First edition

[Compatible models]

Model	Version	Notes
RCX320		
RCX340		

[Operation check model]

Model	Version	Notes
RCX340	V1.83	Check with the emulator on RCX-Studio 2020

目次

1. Introduction

This manual is a user's guide for the Yamaha Robot Controller RCX3 series of Yamaha Motor Co., Ltd. CAO providers. The providers are referred to as RCX3 providers. The Yamaha Robotic Controller RCX3 series of Yamaha Motor Co., Ltd. is called RCX3 series. RCX3 provider communicates with RCX3 series using two communication methods: serial communication and Ethernet communication. Serial communication uses RS-232C, and Ethernet communication uses Telnet communication to perform communication.

Fig. 1-1 shows the overall configuration of this provider and device.

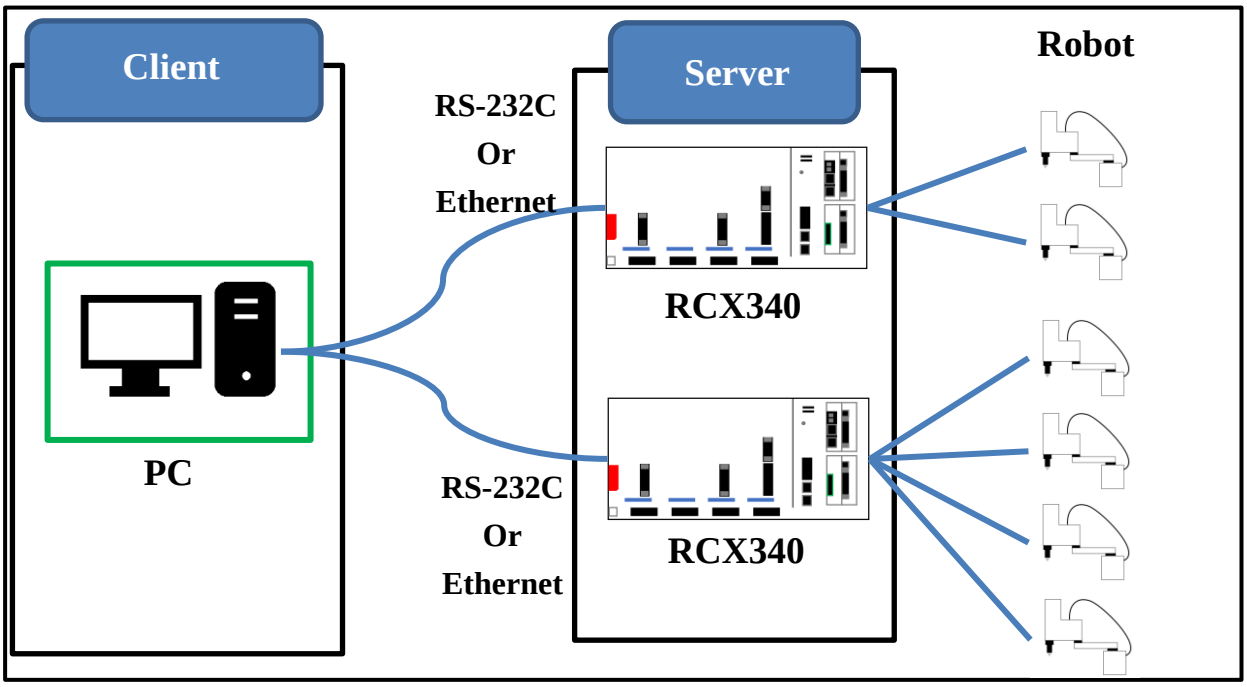


Fig. 1-1 Configuration Diagram

Fig. 1-2 shows the correspondence between this provider and each device.

(* This is an example. It does not represent everything.)

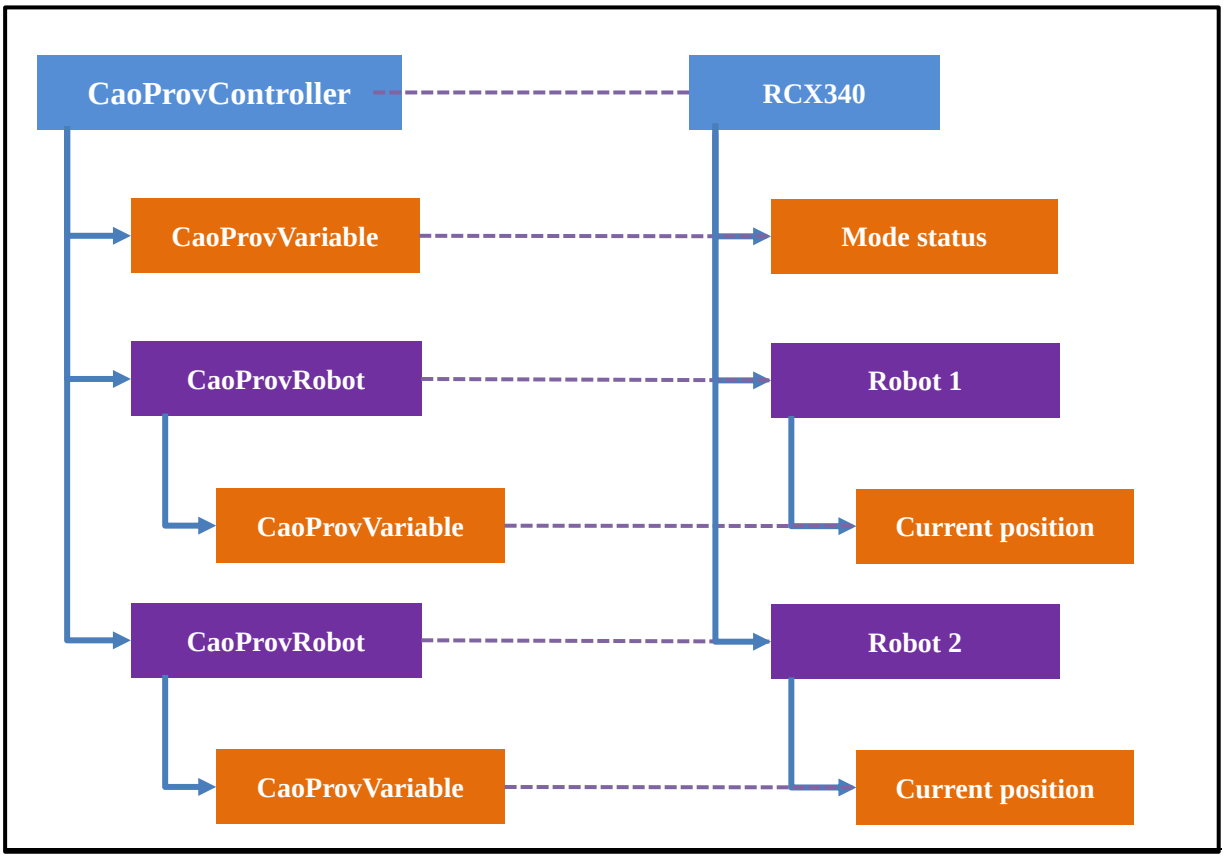


Fig. 1-2 Provider configuration and device information

1.1. Informative sources

RCX3 providers are developed with reference to the "RCX3_ Programming Manual" of Yamaha Motor Co., Ltd. This manual is referred to as RCX3 manual.

2. Setting Up Your Environment for Application Development

2.1. Installation of emergency stop switch

Before using the robot controller, install an emergency stop switch in a position that can be easily operated by the operator so that the robot operation can be stopped immediately in an emergency.

(1) The emergency stop switch should be red.

(2) The emergency stop function should not be automatically restored after operation, and should be prevented from being carelessly restored by other operators.

(3) Provide an emergency stop switch separately from the power switch.

2.2. Connecting RCX3 Series Controller to a Client-PC

RCX3 provider connects RS-232C series with Ethernet communication (Telnet communication). Connect RCX3 series with the client PC with RS-232C cable or LAN cable. For details on how to connect with RCX3 series, refer to "Chapter 6 External Communication Interface" of "RCX3_User's Manual" of Yamaha Motor Co., Ltd. This manual is referred to as RCX3 User's Manual.

・For RS-232C connectivity

Insert RS-232C cable into the PC. Insert the other end of RCX3 cable into the red frame in Fig. 2-1. Insert the cable in the wrong direction, etc. so as not to make a mistake. For details, see "CHAPTER 6 EXTERNAL COMMUNICATION INTERFACE" in "RCX3_USER'S MANUAL".

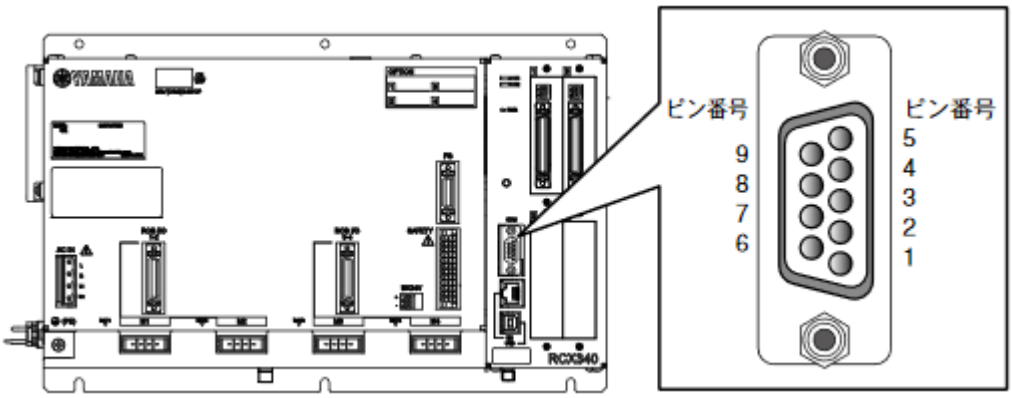


Fig. 2-1 RCX3 Series RS-232C Insertion Diagram

・For Ethernet connectivity

Use a UTP cable (unshielded twisted pair cable) or STP cable (shielded twisted pair cable) that supports Category 5 or higher. Insert the LAN cable into.PC until it clicks into the red frame of Fig. 2-2 by inserting the LAN cable on the other end of RCX3 series directly or through the hub. Insert the cable in the wrong direction, etc. so as not to make a mistake. For more information, see "CHAPTER 6 EXTERNAL COMMUNICATION INTERFACE" in RCX3 User's Manual.

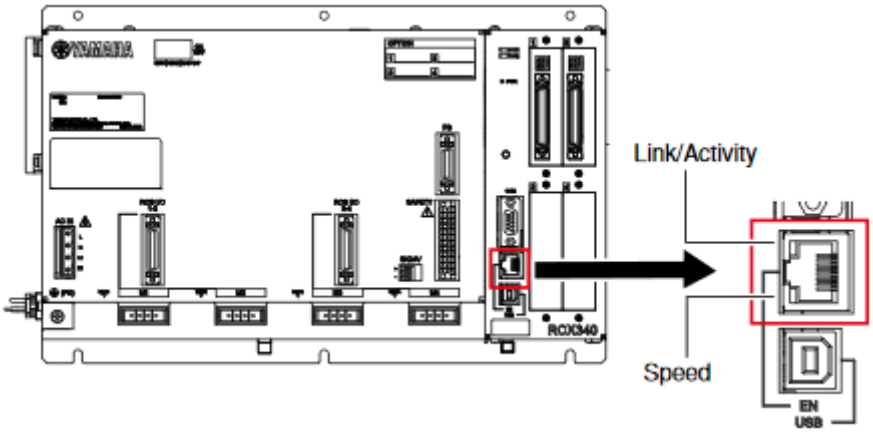


Fig. 2-2 LAN Cable Insertion Diagram for RCX3 Series

2.2.1. RCX3 SeriesCommunication Settings

The communication settings of RCX3 series can be set by using the programming box of Yamaha Motor Co., Ltd. or the support software "RCX-Studio 2020" of the robot controller of RCX3 series manufactured by Yamaha Motor Co., Ltd. For the communication setting method in the programming box, refer to "CHAPTER 6 EXTERNAL COMMUNICATION INTERFACE" in RCX3 USER'S MANUAL. This section describes how to set up communication in RCX-Studio 2020.

1. Start RCX-Studio 2020.
2. Create or select a post-launch project.
3. Next, create or select a new controller and display the controller window in online mode.
4. Open [System]-[Communication Settings] in the target window.
5. RS-232C and Ethernet settings are displayed in the displayed window. Set the desired settings and press OK.
6. In the target window, "[Tools]-[Transfer [PC → Controller]]" is completed by applying the settings to RCX3 series.

For more information on how to do this, see RCX-Studio 2020 Help.

2.3. Setting up a PC development environment

2.3.1. Installing RCX3 Providers Manually

If you install RCX3 providers manually, you must register the registry as shown below. To register the registry, start the command prompt with administrator privileges and execute regsvr32 command. When executing the command, either move to the path where the file is located or specify the file path.

Table2-1 RCX3 providers

File name	CaoProvYAMAHARCX3.dll
ProgID	CaoProv.YAMAHA.RCX3
Registry registration	Regsvr32 CaoProvYAMAHARCX3.dll
Deletion of Registry Registration	Regsvr32 /u CaoProv YAMAHARCX3.dll

3. Command Reference

3.1. Method/Property List

Table 3-1 List of methods and properties

Category	Methods/Properties[1]		Function	Reference
CaoWorkspace				
	AddController	M	Connected to controller	P.11
CaoController				
	AddRobot	M	Adding Robot Objects	P.14
CaoRobot				
	ID	P	Obtain/Set Robot Number	P.15
	VariableNames	P	Get a list of variable names that can be connected	P.15
	Variables	P	Retrieving Variable Collections Holded by the Robot	P.15
	Accelerate	M	Set acceleration and deceleration	P.16
	AddVariable	M	Adding Variable Objects	P.16
	Change	M	Change Hand	P.17
	Execute	M	Execute Extended Commands	P.21
	Halt	M	Stop robot operation halfway	P.17
	Move	M	Move to the position specified by the robot	P.19
	Speed	M	Change the program speed	P.21
CaoVariable				
	Value	P	Get/set value	P.22

3.2. Method properties

3.2.1. CaoWorkspace classes

3.2.1.1. AddController method

RCX3 providers refer to the connection parameters for communication during AddController and connect communication. In this case, optionally specify the communication format, timeout, and Telnet username/password for Ethernet communication.

For Ethernet communication, disable the echo back (ECHO) function of YAMAHA controller. This function can not only reduce performance but also cause false recognition of responses.

[1] M:Indicates methods, P: properties, and E: events, respectively.

The following are the specifics of AddController method:

Format

```
AddController
(
    "<controller name>",                    // Controller name (optional)
    "CaoProv.YAMAHA.RCX3",// Provider name (fixed)
    "<machine name>",                       // Provider execution machine name (unused)
    "<Option>"                  // Option string
)
```

Option

The following options are specified in the option string: The option string is a string consisting of the following options separated by a comma (,).

Option	Required	Description	Value Range	Default value
Conn = <connection parameter>	✓	Set the communication mode and its connection parameters. Refer to 3.2.1.1.1 for details.	--	--
User = <username>	--	Specifies the username for logging in to RCX3 controllers with Ethernet connectivity.	--	Admin
Password = <password>	--	Specifies the password to log in to RCX3 controllers with Ethernet connectivity.	--	None
Timeout = <timeout>	--	Specifies the timeout (in milliseconds) for transmission and reception.	0 - 4294967295	500

Usage example (VBA)

```
Dim caoEng As CaoEngine         ' Engine
Dim caoWs As CaoWorkspace       ' Workspace
Dim caoCtrl As CaoController    ' Controller

' Engine
Set caoEng = New CaoEngine
```

```
' Workspace
Set caoWs = caoEng.AddWorkspace("Workspace", "")
' Controller
Set caoCtrl = caoWs.AddController("RCX3", _
                                  "CaoProv.YAMAHA.RCX3", _
                                  "", _
                                  "Conn=eth:127.0.0.1,Timeout=1000")
```

3.2.1.1.1. Conn Optional

The following is a Conn optional connection parameter string: Parameters within square brackets ("[]") are optional. In the explanation of each parameter, the underlined part shows the default value when the option specification is omitted.

- **Ethernet communication**

"Conn=eth:<IP Address>[:<PortNo>]"

<IP Address>: Required. The IP address to connect to.

Example : "Conn=eth:127.0.0.1", "Conn=eth:192.168.0.1"

<PortNo>: Port number to connect to.

Example : "Conn=eth:127.0.0.1:23", "Conn=eth:192.168.0.1:5010"

- **RS232C communication**

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

<ComPort>: COM-port number. '1'-COM1, '2'-COM2,…

<BaudRate>: Baud rate. 4800, 9600, 19200, 38400, 57600, 115200

<ByteSize>: Parity. 'N'-NONE, 'E'-EVEN, 'O'-ODD

<DataBits>: Number of data bits. '7'-7bit, '8' – 8bit

<StopBits>: Number of stop bits. '1'-1bit, '2'-2bit

<Flow>: Flow-control. ' 0'-None, '1'-Xon/Xoff, '2'-Hardware Control

3.2.1.1.2. Notes on Connection

When connecting RCX3 series with the client PC via Ethernet communication, if a disconnection such as a LAN cable occurs during connection, RCX3 series cannot detect the disconnection. To reconnect, restart RCX3 series and re-add CaoController again.

Same IP address: Port No. cannot be connected to other applications and this provider at the same time.

3.2.2. CaoController classes

3.2.2.1. AddRobot method

Add a robotic object to CaoController. The following are the specifics of AddRobot method:

Format

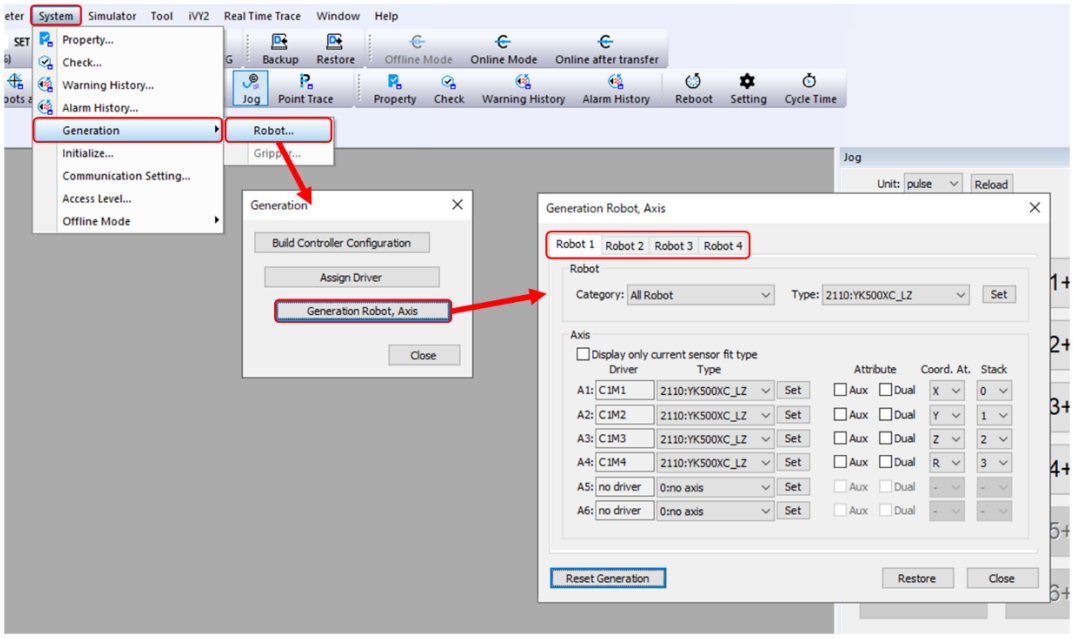
```
AddRobot
(
"<Robot name>",                    // Robot name (optional)
"<Option>"                 // Option string (optional)
)
```

Option

The following options are specified in the option string:

Option	Required	Description	Value Range	Default value
ID=<robot number>	--	Set the number of the target robot. This setting can be changed using CaoRobot::ID property.	1 - 4	1

※ To check the robot number of the robot connected to the controller, display the generation window from [System]-[Generation]-[Robot] in the target window of "RCX-Studio 2020", and display the [Generation Robot, Axis] window. The robot number of the robot displayed on the [Robot 1] tab is "1", the robot number of the robot displayed on the [Robot 2] tab is "2", the robot number of the robot displayed on the [Robot 3] tab is "3", and the robot number of the robot displayed on the [Robot 4] tab is "4".



Usage example (VBA)

```
Dim caoRbt As CaoRobot          ' Robot

' Generate CaoRobot
Set caoRbt = caoCtrl.AddRobot("MF20","ID=1")
```

3.2.3. CaoRobot classes

3.2.3.1. ID property

Gets/sets the robot number of the robot object.

Data type

Type	Description
VT_I4	Robot number

Usage example (VBA)

```
' obtains ID
Dim id As Integer
Id = caoRbt.id
' ID set.
Id = 2
CaoRbt.id = id
```

3.2.3.2. VariableNames Properties

Gets a list of variable names that can be connected. The variable name obtained by this property can be used as the first argument of AddVariable method described later.AddVariable method

Usage example (VBA)

```
' Get list of variable names
Dim variableNames() As String
VaribaleNames = caoRbt.variableNames
```

3.2.3.3. Variables Properties

Gets the variable collection held by the robot.

Usage example (VBA)

```
' Variable Collection Retrieval
Dim variables As CaoVariables
```

```
Set variables = caoRbt.Variables

' Variable acquisition
Dim variable As CaoVariable
Set variable = variables.Item(0)
```

3.2.3.4. AddVariable method

Adds a variable object to CaoRobot. Only the variable names shown in 3.4.1 can be used.

AddVariable is specifi ed as follows.

Format

```
AddVariable
(
    "<variable name>",          // Variable Name
    "<Option>"                  // Option string (optional)
)
```

3.2.3.5. Accelerate method

Set the acceleration and deceleration. Accelerate is specifi ed as follows.

Format

```
Accelerate
(
    "<axis number>",            // Axis number (range: 0 to 6)
    "<Acceleration>"",          // Acceleration (Range: 0 to 100)
    "<Deceleration>"            // Deceleration (range: 0 to 100)
)
```

Axis number

Set the acceleration and deceleration of the specified axis. When 0 is specified as the axis number, acceleration and deceleration are set for all axes.

Acceleration

Set the acceleration. If 0 is specified as the acceleration, the acceleration is not set. To set only deceleration, set this item to 0.

Deceleration

Sets the deceleration. If 0 is specified for deceleration, no deceleration setting is made. Therefore, if you only want to set the acceleration, set this item to 0.

Usage example (VBA)

```
' Set the acceleration of axis 1 to 60 and the deceleration to 50.
CaoRbt.Accelerate 1, 60, 50
' Only the acceleration of axis 2 is set to 30.
CaoRbt.Accelerate 2, 30, 0
' Only deceleration of all axes is set to 40.
CaoRbt.Accelerate 0, 0, 40
```

3.2.3.6. Change method

Switches the robot hand data. Change is specifi ed as follows.

Format

```
Change
(
    "<Hand number>"          // Hand No. (Range: 0 to 31 or "OFF")
)
```

Hand number

Switches to the hand data of the specified hand number. When "OFF" is specified, no hand data is set.

Usage example (VBA)

```
' Set to Hand 1
CaoRbt.Change 1
' Set No Hand Data
CaoRbt.Change "OFF"
```

3.2.3.7. Halt method

Stops the robot operation halfway. Halt is specifi ed as follows.

Format

```
Halt
(
    "<Option string>"          // Option string (unused)
)
```

Usage example (VBA)

```
' Stop the robot
CaoRbt.Halt
```

3.2.3.8. Move method

Moves the robot to the specified position. Move command type is specified as the first argument, and the destination information and speed are specified as the second argument in VARIANT array. The specifi cations of Move are shown below.

Format

```
Move
(
    "<interpolation specification>",        // Interpolation specification (range: 1 to 7)
    "<Specify Move Destination>",           // Specify destination
    "<Option>"                  // Option (optional)
)
```

Interpolation specification

The content of this parameter specifies the interpolation specification. For details of interpolation specification, refer to "RCX3 Series Programming Manual".

Details of interpolation specification that can be specified are shown below.

Specified value	Command name	Move Type	Description
1	MOVE	P	Move the robot through the path in which each axis can move as short as possible from the position where the robot is currently stopped to the position of the specified point.
2	MOVE	L	The robot is moved in a route in which two points according to the position where the robot is currently stopped and the specified point position are connected by a straight line.
3	MOVE	C	Moves the robot in the path where the robot is currently stopped, the specified point position in the middle, and the three points according to the specified point position of the target are connected by an arc. Therefore, an even number of points must be specified.
4	MOVEI	P	Move the robot through the path in which each axis with the position plus the specification point coordinates from the position where the robot is currently stopping can move in the shortest distance.

Specified value	Command name	Move Type	Description
5	MOVEI	L	The robot is moved in a route in which two points at the position plus the specified point coordinates are connected by a straight line from the position where the robot is currently stopping.
6	MOVET	P	Move the robot through the path in which each axis with the position plus by the tool coordinate can move as short as possible from the position where the robot is currently stopping.
7	MOVET	L	Move the robot in a path connecting two points with a straight line from the position where the robot is currently stopped to the position where the robot is plus by the tool coordinates.

※ P : Point to Point, L : Linear interpolation, C : Circular interpolation

Specify destination

Specify the destination as a string. When specifying a value directly, enter a space between the values to be set for each axis. For details on how to specify, refer to the "RCX3 Series Programming Manual".

※ When specifying directly by numerical value, specify 0 for axes that do not exist in the target robot.

Option

Specify each option separated by a comma. Options are optional. For more information about the options, see 6.2. Types of Options.

Usage example (VBA)

・Using Point Data to Execute Move Method

```
' Only the acceleration of axis 2 is set to 30.
CaoRbt.Move 1, "P13", "S=50,CONT"          ' 1 = Specifies PTP, relative speed (50%), and specifies CONT.
CaoRbt.Move 2, "P11"                       ' 2 = Linear interpolation (move from current position to P11)
CaoRbt.Move 3, "P10,P11,P12,P13"           ' 3 = Circle interpolation
CaoRbt.Move 4, "P11", "S=50"               ' Move P11=170mm/deg from current position
```

・Execute Move method using direct numeric values (up to axis 4 of robots to Move)

```
CaoRbt.Move 1, "152.997 301.000 70.000 75.000 0 0", ""          ' 1 = PTP, optional omitted
```

3.2.3.9. Speed method

Change the program speed. The speed of all axes is changed. Speed is specifi ed as follows.

Format

```
Speed
(
    "<axis number>",            // Axis number (unused)
    "<Speed>"                   // Speed (range: 1 to 100)
)
```

Speed

Specifies the program speed. The movement speed of the robot is determined by the product of the automatic movement speed and the program movement speed specified by this method.

Movement speed = Automatic movement speed × Program movement speed

Example) Automatic moving speed (80%) × Program moving speed (50%) = Moving speed (40%)

Usage example (VBA)

```
' Set program speed to 50%
CaoRbt.Speed 1,50
```

3.2.3.10. Execute method

Execute CaoRobot extended command. Only the extended commands listed in 3.3.1.CaoRobot Class Extended Commands can be used for the extended commands that can be specified in Execute. Execute is specifi ed as follows.

Format

```
Execute
(
    "<extension command name>",         // Extended command name
    "<Option string>"           // Option string (optional)
)
```

3.2.4. CaoVariable classes

3.2.4.1. Value Properties

Acquires/sets data from the connected RCX3 series controller. The behavior depends on the variable name. For details, refer to section 3.4, Variable List.

3.3. Extended command list

Defines the list of commands available for each class. The usage examples are described in detail for each command.

3.3.1. CaoRobot class-extension command

The extended commands that operate the robot in this chapter will wait until the robot operation is complete or a timeout occurs.

Table 3-2 CaoRobot class extension command list

Command	Description	Reference
PMOVE	Pallet move command	P.23
SERVO	Servo status setting	P.23

3.3.1.1. PMOVE Commands

Execute palletizing motion command. The following arguments are provided.

Item	Type Description		
Argument	VT_ARRAY | VT_VARIANT		
	0	VT_I4	Pallet number (0 to 39)
	1	VT_I4	Point No. (1 to 32767)
	2	VT_BSTR	Option string Specify each option separated by a comma. Options are optional. For more information about the options, see 6.2.Types of Options.

Usage example (VBA)

```
' Pallet moving by specifying pallet number 1, point number 5, and program speed at 50%
Call caoRbt.Execute("PMove", Array(1, 5, "S=50"))
```

3.3.1.2. SERVO Commands

Set the servo status. To operate the robot, the servo status must be set to ON. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY | VT_VARIANT	Movement position specification

Item	Type Description		
	0	VT_BSTR	Servo status. Specify one of the following: ON OFF FREE
	1	VT_I4	Axis number (1 to 6). If omitted or 0 is specified, all axes are specified.

Usage example (VBA)

```
' Turns ON the servo status of axis 1.
Call caoRbt.Execute("SERVO", array("ON", 1))

' Turns the servo status of all axes OFF.
Call caoRbt.Execute("SERVO", array("OFF"))
```

3.4. Variable list

Defines a list of variables that can be used in each class. Variables refer to objects of CaoVariable classes.

3.4.1. CaoRobot class-variable

Variable Name	Description	Value		Reference
		Get	Put	
@WHERE	Gets the current position of the robot pulse coordinate system.	✓	-	P.24
@WHRXY	Gets the current position of the robot's millimeter system of units.	✓	-	P.25

3.4.1.1. @WHERE

Obtains the current arm position (pulse coordinate) of the robot.

Data type

Type Description	
VT_I4 | VT_ARRAY	Current Position of Robot Pulse Coordinate System
0	Current position of 1st axis
1	Current position of the 2nd axis

2	Current position of the 3rd axis
3	Current position of the 4th axis
4	Current position of the 5th axis
5	Current position of axis 6

Usage example (VBA)

```
Dim whereVariable As CaoVariable
Dim whereValues() As Long

Set whereVariable = caoRbt.AddVariable("@WHERE")
WhereValues = whereVariable.Value

' Acquisition of values for each axis
If Not IsEmpty(whereValues) Then
    Dim i As Integer
    For i = 0 To UBound(whereValues)
        Dim value As Integer
        Value = whereValues(i)
    Next i
End If
```

3.4.1.2. @WHRXY

Obtains the current arm position (Cartesian coordinate) of the robot.

Data type

Type	Description
VT_R4 | VT_ARRAY	Cartesian coordinate system current position of the robot
0	Current position of 1st axis
1	Current position of the 2nd axis
2	Current position of the 3rd axis
3	Current position of the 4th axis
4	Current position of the 5th axis
5	Current position of axis 6

Usage example (VBA)

```
Dim whrxyVariable As CaoVariable
```

```
Dim whrxyValues() As Single

Set whrxyVariable = caoRbt.AddVariable("@WHRXY")
WhrxyValues = whrxyVariable.Value

' Acquisition of values for each axis
If Not IsEmpty(whrxyValues) Then
    Dim i As Integer
    For i = 0 To UBound(whrxyValues)
        Dim value As Single
        Value = whrxyValues(i)
    Next i
End If
```

4. Programming by RCX3 providers

RCX3 providers prepare to communicate with the equipment as follows:

- Creating a CaoEngine
- Creating a CaoWorkspace
- Creating a CaoController

After you complete the steps, you can communicate with RCX3 series and access the device information by using Execute method of CaoController or by obtaining Value properties of CaoVariable object.

In the example program, the program language "VBA (Visual Basic for Applications)" is used for programming.

4.1. Sample programming to acquire robot arm current position (pulse value)

This example shows a sample program that reads the current arm position of the robot. Table 4-1 describes the requirements of the sample program, and Fig. 4-1 describes the flow of the sample program.

Table 4-1 Sample program requirements

Requirements	Description
Process Description	Load the current position of the robot arm from RCX340.

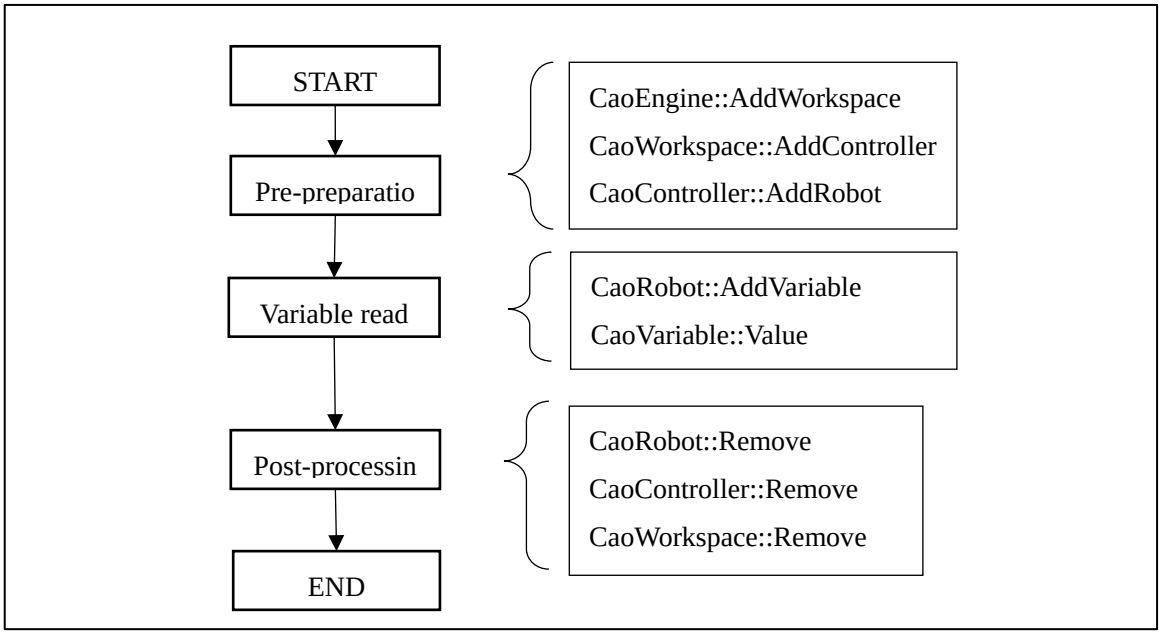


Fig. 4-1 Flow of Loading the Robot's Arm Current Position

Specific codes are given in the following sections.

4.1.1. Sample program

The following is an overview of the sample program.

Sample GetCurrentData.vb

```
' Object
Dim caoEng As CaoEngine
Dim caoWs As CaoWorkspace
Dim caoCtrl As CaoController
Dim caoRbt As CaoRobot
Dim whereVaraible As CaoVariable

Private Sub Main()
    ' Pre-processing
    Call Connect

    Dim whereValues() As Integer
    WhereValues = whereVaraible.Value

    ' Acquisition of values for each axis
    If Not IsEmpty(whereValues) Then
        Dim i As Integer
        For i = 0 To UBound(whereValues)
            Dim value As Single
            Value = whereValues(i)
        Next i
    End If

    ' Post-processing
    Call Disconnect
End Sub

' Preprocessing method
Private Sub Connect()
    ' Generate CaoEngine
    Set caoEng = New CaoEngine
    ' Generate CaoWorkspace
    Set caoWs = caoEng.AddWorkspace("Workspace", "")
```

```
    ' Generate CaoController
    Set caoCtrl = caoWs.AddController("RCX3", _
                                      "CaoProv.YAMAHA.RCX3", _
                                      "", _
                                      " Conn=eth:192.168.0.10,Timeout=1000")
    ' Generate CaoRobot
    Set caoRbt = caoCtrl.AddRobot("MF20")
    ' Generate CaoVariable
    Set whereVaraible = caoRbt.AddVariable("@WHERE")
End Sub

' Post-processing method
Private Sub Disconnect()
    ' Remove CaoVariable from CaoController
    Call caoCtrl.variables.Remove(whereVaraible.Index)
    ' Clear CaoVariable
    Set whereVaraible = Nothing
    ' Remove CaoController from CaoWorkspace
    Call caoWs.Controllers.Remove(caoCtrl.Index)
    ' Clear CaoController
    Set caoCtrl = Nothing
    ' Remove CaoWorkspace from CaoEngine
    Call caoEng.Workspaces.Remove(caoWs.Index)
    ' Clear CaoWorkspace
    Set caoWs = Nothing
    ' Clear CaoEngine
    Set caoEng = Nothing
End Sub
```

4.1.1.1. Pre-processing

To communicate with the instrument, proceed as follows:

(1) Prepare a variable to hold the object. The objects required to connect to the controller are CaoEngine object, CaoWorkspace object, and CaoController object. CaoWorkpace object does not need to have a variable to obtain CaoController object from CaoWorkspaces. You also need a CaoRobot object to specify the target robot and a CaoVariable object to access the variable. The following is a code example for VBA.

```
' Variables for CaoEngine Objects
Dim caoEng As CaoEngine
' Variables for CaoWorkspace Objects
Dim caoWs As CaoWorkspace
' Variables for CaoController Objects
Dim caoCtrl As CaoController
' Variables for CaoRobt Objects
Dim caoRbt As CaoRobot
' Variables for CaoVariable Objects
Dim whereVaraible As CaoVariable
```

(2) Creates a CaoEngine object. CaoEngine object is generated using the New keyword.

```
' Generate CaoEngine
Set caoEng = New CaoEngine
```

(3) Gets or generates a CaoWorkspace object. When you create a CaoEngine object, it defaults to one CaoWorkspaces object and one object. The following is a sample code/default CaoWorkspace for creating a new CaoWorkspace.

```
' Generate CaoWorkspace
Set caoWs = caoEng.AddWorkspace("Workspace", "")
```

(4) Create a CaoController object. To generate a CaoController object, set the provider name to use and the parameters to use. For RCX3 providers, optionally specify the destination information and timeout. The following is a code example:

```
' Generate CaoController
Set caoCtrl = caoWs.AddController("RCX3", _
                                  "CaoProv.YAMAHA.RCX3", _
                                  "", _
                                  " Conn=eth:192.168.0.10,Timeout=1000")
```

(5) Create a CaoRobot object. To create a CaoRobot object, set the robot name and parameters. For RCX3 providers, optionally specify the robot number. The following is a code example:

```
' Generate CaoRobot
Set caoRbt = caoCtrl.AddRobot("MF20")
```

(6) Creates a CaoVariable. Create a CaoVariable for the variable you want to retrieve. The following code

example generates a variable object that accesses the arm current position of the robot.

```
' Generate CaoVariable
Set whereVaraible = caoRbt.AddVariable("@WHERE")
```

4.1.1.2. Acquire robot arm current position

To get the arm current position of the robot, refer to Value properties of CaoVariable object. Here, the data of each axis is referenced. The following is a code example:

```
Dim whereValues() As Integer
WhereValues = whereVariable.Value
' Acquisition of values for each axis
If Not IsEmpty(whereValues) Then
    Dim i As Integer
    For i = 0 To UBound(whereValues)
        Dim value As Single
        Value = whereValues(i)
    Next i
End If
```

4.1.1.3. Post-processing

To perform post-processing, you can erase the generated objects and delete the objects that you want to erase from the collection class that manages the objects. The following is a code example:

```
' Remove CaoVariable from CaoRobot
Call caoRbt.Variables.Remove(whereVariable.Index)
' Clear CaoVariable
Set whereVariable = Nothing
' Remove CaoRobot from CaoController
Call caoCtrl.Robots.Remove(caoRbt.Index)
' Clear CaoController
Set caoRbt = Nothing
' Remove CaoController from CaoWorkspace
Call caoWs.Controllers.Remove(caoCtrl.Index)
' Clear CaoController
Set caoCtrl = Nothing
' Remove CaoWorkspace from CaoEngine
Call caoEng.Workspaces.Remove(caoWs.Index)
' Clear CaoWorkspace
```

```
    Set caoWs = Nothing
    ' Clear CaoEngine
    Set caoEng = Nothing
```

4.2. Sample Programming to Move the Robot Using Point Data

Here is an example program for moving a RCX3 series robot. Table 4-2 describes the requirements of the sample program, and Fig. 4-2 describes the flow of the sample program.

Table 4-2 Sample program requirements

Requirements	Description
Process Description	Move RCX340's robotics.

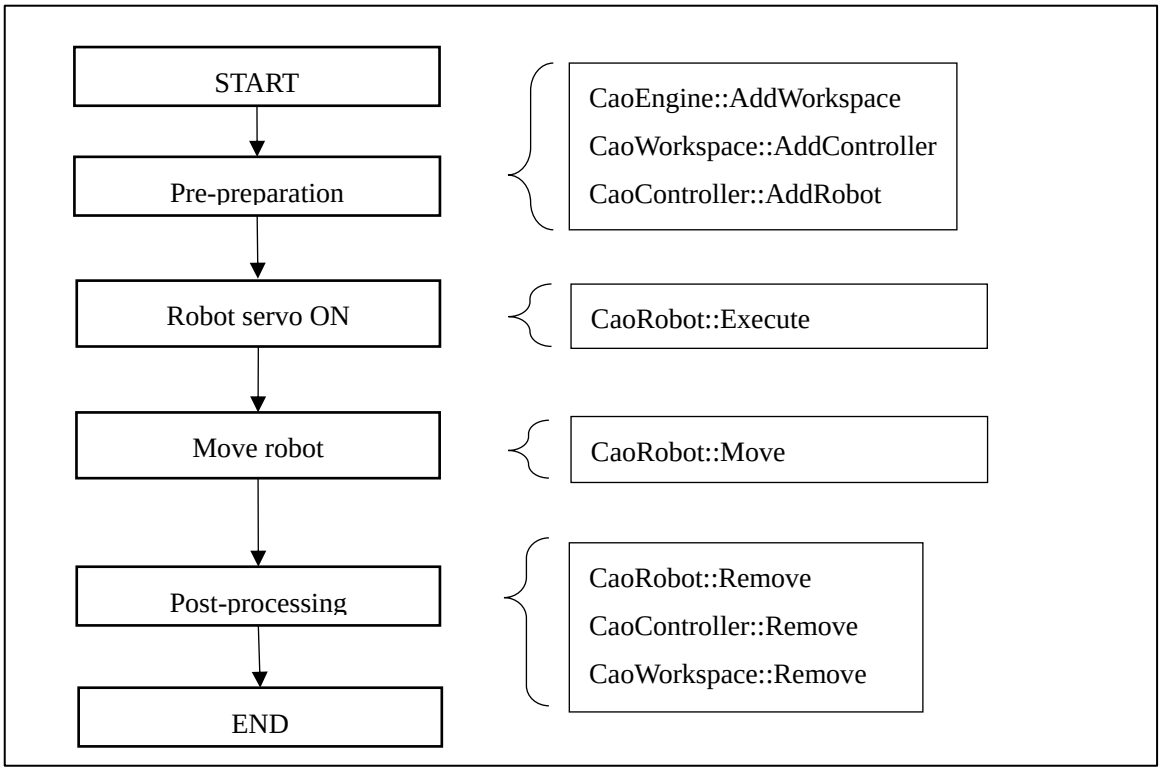


Fig. 4-2 Flow of Moving the Robot

Specific codes are given in the following sections.

4.2.1. Sample program

The following is an overview of the sample program.

Sample **GetCurrentData.vb**

```
' Object
Dim caoEng As CaoEngine
Dim caoWs As CaoWorkspace
Dim caoCtrl As CaoController
Dim caoRbt As CaoRobot
```

```
Private Sub Main()
    ' Preparation
    Call Connect

    ' Set the servo status of all axes of the robot to ON.
    Call caoRbt.Execute("SERVO", Array("ON"))

    ' Moving the Robot Using Numerical Values
    CaoRbt.Move 1, "152.997 301.000 70.000 75.000 0 0", "S=80"

    ' Post-processing
    Call Disconnect
End Sub

' Preparation method
Private Sub Connect()
    ' Generate CaoEngine
    Set caoEng = New CaoEngine
    ' Generate CaoWorkspace
    Set caoWs = caoEng.AddWorkspace("Workspace", "")
    ' Generate CaoController
    Set caoCtrl = caoWs.AddController("RCX3", _
                                      "CaoProv.YAMAHA.RCX3", _
                                      "", _
                                      " Conn=eth:192.168.0.10,Timeout=1000")
    ' Generate CaoRobot
    Set caoRbt = caoCtrl.AddRobot("MF20")
End Sub

' Post-processing method
Private Sub Disconnect()
    ' Remove CaoRobot from CaoController
    Call caoCtrl.Robots.Remove(caoRbt.Index)
    ' Clear CaoRobot
    Set caoRbt = Nothing
    ' Remove CaoController from CaoWorkspace
    Call caoWs.Controllers.Remove(caoCtrl.Index)
```

```
    ' Clear CaoController
    Set caoCtrl = Nothing
    ' Remove CaoWorkspace from CaoEngine
    Call caoEng.Workspaces.Remove(caoWs.Index)
    ' Clear CaoWorkspace
    Set caoWs = Nothing
    ' Clear CaoEngine
    Set caoEng = Nothing
End Sub
```

4.2.1.1. Pre-processing

To communicate with the instrument, proceed as follows:

(1) Prepare a variable to hold the object. The objects required to connect to the controller are CaoEngine object, CaoWorkspace object, and CaoController object. CaoWorkpace object does not need to have a variable to obtain CaoController object from CaoWorkspaces. In addition, a CaoRobot is required to specify the target robot. The following is a code example for VBA.

```
' Variables for CaoEngine Objects
Dim caoEng As CaoEngine
' Variables for CaoWorkspace Objects
Dim caoWs As CaoWorkspace
' Variables for CaoController Objects
Dim caoCtrl As CaoController
' Variables for CaoRobt Objects
Dim caoRbt As CaoRobot
```

(2) Creates a CaoEngine object. CaoEngine object is generated using the New keyword.

```
' Generate CaoEngine
Set caoEng = New CaoEngine
```

(3) Gets or generates a CaoWorkspace object. When you create a CaoEngine object, it defaults to one CaoWorkspaces object and one object. The following is a sample code/default CaoWorkspace for creating a new CaoWorkspace.

```
' Generate CaoWorkspace
Set caoWs = caoEng.AddWorkspace("Workspace", "")
```

(4) Create a CaoController object. To generate a CaoController object, set the provider name to use and the parameters to use. For RCX3 providers, optionally specify the destination information and timeout. The following is a code example:

```
' Generate CaoController
Set caoCtrl = caoWs.AddController("RCX3", _
                                  "CaoProv.YAMAHA.RCX3", _
                                  "", _
                                  " Conn=eth:192.168.0.10,Timeout=1000")
```

(5) Create a CaoRobot object. To create a CaoRobot object, set the robot name and parameters. For RCX3 providers, optionally specify the robot number. The following is a code example:

```
' Generate CaoRobot
Set caoRbt = caoCtrl.AddRobot("MF20")
```

4.2.1.2. Turns on the servo status of the robot.

To activate the servo status of the robot, execute the "SERVO" command in Execute method of CaoRobot. In this case, all axes are set to ON. The following is a code example:

```
' Set the servo status of all axes of the robot to ON.
Call caoRbt.Execute("SERVO", Array("ON"))
```

4.2.1.3. Move robot

Move the robot using the number to move the robot. To move the robot, execute CaoRobot object's Move method. Performs a Point to Point move using a direct numeric value. The following is a code example:

```
' Moving the Robot Using Defined Point Data
CaoRbt.Move 1, "152.997 301.000 70.000 75.000 0 0", "S=80"
```

4.2.1.4. Post-processing

To perform post-processing, you can erase the generated objects and delete the objects that you want to erase from the collection class that manages the objects. The following is a code example:

```
' Remove CaoRobot from CaoController
Call caoCtrl.Robots.Remove(caoRbt.Index)
' Clear CaoRobot
Set caoRbt = Nothing
' Remove CaoController from CaoWorkspace
Call caoWs.Controllers.Remove(caoCtrl.Index)
```

```
' Clear CaoController
Set caoCtrl = Nothing
' Remove CaoWorkspace from CaoEngine
Call caoEng.Workspaces.Remove(caoWs.Index)
' Clear CaoWorkspace
Set caoWs = Nothing
' Clear CaoEngine
Set caoEng = Nothing
```

5. RCX3 Provider Error Codes

RCX3 providers do not have their own error codes and have the following provider error codes masked with the 0x801***** from the device (see Table 5-1 Provider Error Codes Table).

For information about common ORiN2 errors, see the Error Codes section of ORiN2 Programming Guide.

Table 5-1 Provider Error Codes Table

Error Number	Description
0x801xxyyy	If an error occurs when RCX3 command is executed, the group number of the alarm code is returned in xx and the classification number of the alarm code is returned in yyy in hexadecimal. For more information about alarm codes, see Troubleshooting in RCX3 User's Manual.

6. Appendix

6.1. RCX3 command reference table

6.1.1. CaoRobot classes

Table 6-1 CaoRobot::Execute Method-RCX3 Command Correspondence Table61

Command name	RCX3 command
PMOVE	PMOVE[i]
SERVO	SERVO[i]

※ i = robot number

Table 6-Non 2 CaoRobot::Execute Methods-RCX3 Command Correspondence Table62

Method	RCX3 command
Accelerrate	ACCEL[i]
	DECEL[i]
Change	CHANGE[i]
Halt	^C
Move	MOVE[i]
	MOVEI[i]
Speed	SPEED[i]

※ i = robot number

Table 6-3 CaoRobot Variable Object-RCX3 Command Correspondence Table63

Variable Name	RCX3 command
@WHERE	?WHERE[i]
@WHRXY	?WHRXY[i]

※ i = robot number

6.2. Option Type

There are several options that you can specify when you execute a command to move the robot. For details about the options that can be specified for each command, see Table 6-4.Table 6-4: Correspondence between options that can be specified64

Table 6-4: Correspondence between options that can be specified64

Option	Option string	MOVE P	MOVE L	MOVE C	MOVEI P	PMOVE	Reference
Relative-speed specification (SPEED)	"S="	✓	✓	✓	✓	✓	P.41
Absolute speed specification (DSPEED)	"DS="	✓	✓	✓	✓	-	P.41
Linear speed specification (VEL)	"VEL="	-	✓	✓	-	-	P.41
Arch motion	"Ax=" (x:Axis number)	✓	-	-	✓	✓	P.42
CONT specification	"CONT"	✓	✓	✓	✓	✓	P.42
Acceleration specification	"ACC="	✓	✓	✓	✓	-	P.43
Deceleration specification	"DEC="	✓	✓	✓	-	-	P.43
Coordinate Plane Specification	"XY" Or "YZ" Or "ZX"	-	-	✓	-	-	P.43
Port output specification	"DOm(b,...,b)=" Or "MOm(b,...,b)=" Or "SOm(b,...,b)=" (m:m: port number b:Bit designation)	-	✓	✓	-	-	P.44

The following describes the specification method and details of options that can be specified.

6.2.1. Relative-speed specification (SPEED)

Specifies the program movement speed (S) as a ratio. This speed setting further changes the set speed relative to the currently set auto speed.

Robot operation speed =

Maximum speed of the robot (mm/sec or degree/sec) × Automatic movement speed (%) × Program movement speed S (%)

Option format

"S = ratio"	Specify 1 to 100 for the ratio (unit: %).

Usage example (VBA)

```
Dim caoRob As Object
Set caoRob = caoCtrl.AddRobot("MF20")
' Move robot at relative speed specification (50%)
CaoRob.Move 1, "P1", "S=50"
```

6.2.2. Absolute speed specification (DSPEED)

Specifies the ratio to the maximum speed of the robot (cannot be used in conjunction with relative speed specification).

Motion speed of robot = Maximum speed of robot (mm/sec or degrees/sec) × Movement speed DS (%)

Option format

"DS= Ratio"	Specify 0.01 to 100.00 for the ratio (unit: %).

Usage example (VBA)

```
Dim caoRob As Object
' Move robot at absolute speed specification (10.01%)
CaoRob.Move 1, "P1", "DS=10.01"
```

6.2.3. Linear speed specification (VEL)

Specifies the maximum combined XYZ axis speed of the robot in mm/sec. This speed setting operates at the specified speed regardless of the automatic movement speed.

Option format

"VEL = maximum speed" For maximum speed, specify the maximum speed of 1 to each model (in mm/sec.)

Usage example (VBA)

```
' Move the robot at the specified linear speed (100 mm/s).
CaoRob.Move 2, "P1", "VEL=100"
```

6.2.4. Arch motion

When the axis specified in the option format starts moving to the arch position and the specified axis moves by one or more arch distances, the other axes move to the target position. When the other axis finishes moving, it moves to the target position so that the remaining distance becomes the remaining arch distance 2. When all axes fall within the Out valid position range, the command ends. For more information on arch motion, refer to RCX3 Series Programming Manual.

Option format

"x = expression {expression 1, expression 2}" {Equation 1, Equation 2} is optional

- x : x : Arch axis (A1 to A6)
- Expression: Arch position (direct numeric or variable specification)
- Equation 1: Linear Distance to Starting Point (Arch Distance 1)
- Equation 2: Linear Distance to End Point (Arch Distance 2)

Usage example (VBA)

```
'The A3 axis moves to 0.00mm from the current position. After that, the other axes move, and finally, the A3 axis moves to P1.
CaoRob.Move 1, "P1", "A3=0.00{50.00,70.00}"
```

6.2.5. CONT specification (concatenation operation)

When CONT specified movement is executed, the next instruction starts without waiting for the operation completion (arrival within the tolerance range) of the moving object axis. If the next instruction is a moving instruction, the two moving paths are connected by connecting the deceleration and acceleration parts, and the movement continues without stopping on the way. For more information on specifying CONT, refer to RCX3 Series Programming Manual.

Option format

"CONT"

Usage example (VBA)

```
CaoRob.Move 1, "P1", "CONT"          ' CONT specification
```

6.2.6. Acceleration specification

The ratio to the acceleration of the robot determined by the acceleration coefficient parameter is specified by an optional expression, and the acceleration during operation is determined.

Acceleration during operation = Acceleration coefficient parameter × Acceleration ACC (%)

Option format

"ACC= Expression"	Specify 1 to 100 in the expression (unit: %).

Usage example (VBA)

```
CaoRob.Move 1, "P1", "ACC=10"          'Move at 10% acceleration ratio
```

6.2.7. Deceleration specification

The ratio to the acceleration of the robot determined by the acceleration coefficient parameter is specified by an optional expression, and the deceleration during operation is determined.

Deceleration in operation = Acceleration coefficient parameter × Deceleration DEC (%)

Option format

"DEC= Expression"	Specify 1 to 100 in the expression (unit: %).

Usage example (VBA)

```
CaoRob.Move 1, "P1", "DEC=20"          'Move at 20% deceleration ratio
```

6.2.8. Coordinate Plane Specification

When arc interpolation with coordinate specification is executed, arc interpolation is executed so that the projection to the specified coordinate plane becomes a circle. For more information on specifying the coordinate plane, refer to RCX3 Series Programming Manual.

Option format

"XY"	XY Coordinate planes
Or	
"YZ"	YZ Coordinate planes
Or	
"ZX"	ZX Coordinate planes

Usage example (VBA)

```
'Move XY planar arc generated at P10 to P14
CaoRob.Move 3, "P10,P11,P13,P14", "XY"
```

6.2.9. Port output specification

When linear interpolation or arc interpolation is executed, the value of Expression 1 is output to the specified output port at a timing that is separated from the movement start position by the distance (unit: mm) of Expression 2 in the option format. (If you specify a port number that does not exist in hardware, nothing is output.)

This option can be specified up to two times within the option.

For more information on specifying port outputs, see RCX3 Series Programming Manual.

Option format

"DOm(b,...,b) = Equation 1@Equation 2" b, If … or b is omitted, all 8 bits are subject to processing.

Or

"MOm(b,...,b) = Equation 1@Equation 2"

Or

"SOm(b,...,b) = Equation 1@Equation 2"

m : Port number (2~7,10~17,20~27)

b : Bit specification (0 to 7)

Expression 1: Value to be output to the specified port (valid only for integers)

Equation 2: Port output position (up to three decimal places can be specified) * Radius of the circle centered on the movement start point

Usage example (VBA)

```
' 105 (&B01101001) is outputted to DO2 () at 28.85mm away from the position prior to movement while the robot is moving to P1 in linear interpolation.
CaoRob.Move 2, "P1", "DO2()=105@25.85"
```