FUTABA
RSC-U485 provider

Version 1.0.0

User's Guide

June 20, 2018

NOTE:

[Revision History]

Version	Date	Description
1.0.0	2018-01-23	First edition

[Operation Checked Device]

Model name	Firmware Version	Note
RSC-U485	*.*.*	

Table of contents

1. Introduction

This manual is a user's guide for Futaba RSC-U485 provider that controls the servos in RS485 format connected to the Futaba Serial Converter "RS-U485".

With Futaba RSC-U485 provider, you can easily read and write servos in RS485 format connected to RSC-U485 devices.

This document describes the features of Futaba RSC-U485 provider.

2. Provider overview

Futaba RSC-U485 provider connect to RSC-U485 using RS485 communication method and read and write data to the servos connected to RSC-U485. Fig. 2-1 shows the overall configuration of this provider and the device.

For details on how to connect and install drivers, refer to RSC-U485's instruction manual.

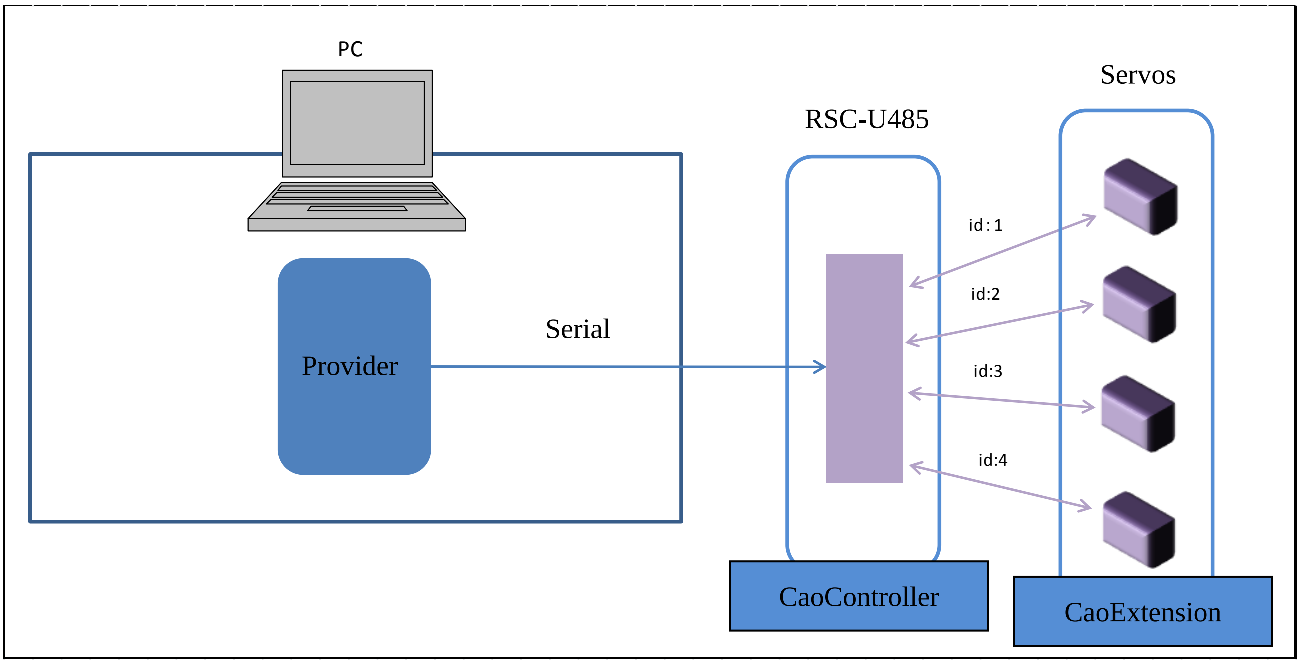

Fig. 2-1 Overall configuration

2.1. Introduction

The file format of Futaba RSC-U485 provider is DLLs (Dynamic Link Library), which are detailed in Table 2-1.

Table 2-1 Futaba RSC-U485 Provider

File name	CaoProvFutabaRSC-U485.dll
ProgID	CaoProv.Futaba.RSC-U485
Registry registration	Regsvr32 CaoProvFutabaRSC-U485.dl
Delete registry registration	Regsvr32 /u CaoProvFutabaRSC-U485.dl

Fig. 2-2 shows the correspondence between this provider and each device. As shown in Fig. 2-2, CaoProvController corresponds to RSC-U485 and CaoProvExtension corresponds to the servo motor connected to RSC-U485.

The ID of each servo motor is a unique numeric value. Note that if duplicate IDs are assigned, the connection will fail.

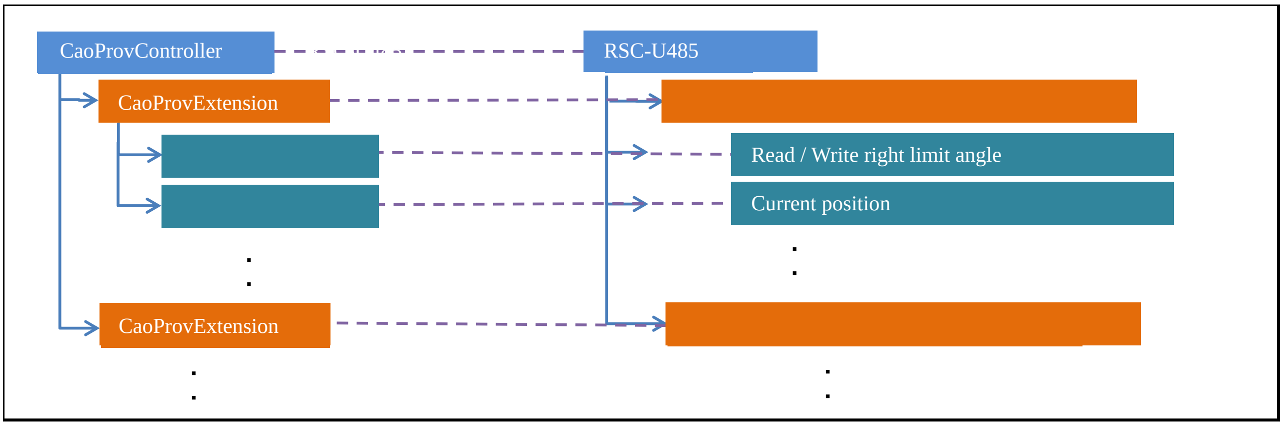

Fig. 2-2 Correspondence diagram of Futaba RSC-U485 Provider and RSC-U485

2.2. Method properties

2.2.1. CaoWorkspace::AddController method

Create Controller objects.

Each option specifies information to connect to the device.

Show AddController format as follows.

FORMAT

```
AddController
(
        "<controller name>",            // Controller name (optional)
        "CaoProv.Futaba.RSC-U485",      // Provider name (fixed)
        "<machine name>",               // Provider execution machine name (unused)
        "<option>"                      // Option character string
)
```

The following are the characters that you specify for Option character string:

Table 2-2 Option character string in the CaoWorkspace::AddController

Option	Required	Description	Value Range	Default Value
Conn	✓	Specify a connection option to the device. A connection option is specified in the format of "Com: <port number>".	--	--
Timeout	-	Specify the communication timeout[ms].	1 - 65535	500

Usage example(PacScript)

```
Dim controller As Object
controller = Cao.AddController("RSC-U485F", "CaoProv.Futaba.RSC-U485", "", "Conn=Com:1, Timeout=500")
```

2.2.2. CaoController::AddExtension method

Create Extenision objects.

Show AddExtension format as follows.

FORMAT

```
AddExtension
(
        "<extension name>",        // Extension name(optional)
        "<option>"                 // Option character string
)
```

The following are the characters that you specify for Option character string:

Table 2-3 Option character string in the CaoController::AddExtension

Option	Required	Description	Value Range	Default Value
Id	✓	Specifies the ID number of the servo to be connected.	1-127	1

Usage example(PacScript)

```
Dim extension As Object
extension = controller.AddExtension("servo1","Id=100")
```

2.2.3. CaoController::property get_VariableNames

Get the variable names shown in Table 2-5.

Show GetVariableNames format as follows.

FORMAT

```
GetVariableNames
(
        "<option>"            // Option character string (unspecified)
)
```

Usage example(PacScript)

```
Dim variablelist As Object
variablelist = controller.GetVariableNames("")
```

2.2.4. CaoController::AddVariable method

Creates a CaoVariable from a CaoController. Only the variable names shown in Table 2-5 can be used.

Show AddVariable format as follows.

FORMAT

```
AddVariable
(
        "<variable name>",       // Variable name
        "<option>"               // Option character string
)
```

For Option character string, see 2.3.1.

2.2.5. CaoExtension::Execute method

Execute the method in CaoExtension object.

For the argument, specify the required argument for the method.

Show Execute format as follows.

FORMAT

```
Execute
(
        "<method name>",  // Method name
        "<argument>"      // Argument
)
```

The following is a list of methods that can be specified in Execute.

Table 2-4 CaoExtension::Execute Methods

Command name	Description	Link
GetAllParameters	Get the parameters of servo memory map No.0 to No.59.	P.14
InitServo	Initialize the servo memory map No.4 to No.29.	P. 14
WriteFlashROM	Write and restart the servo flash ROM.	P.15
SendPacket	Send the set packet directly to the servo. For RS485 packet, refer to the manual for each servo.	P.15

2.2.5.1. **CaoExtension::Execute("GetAllParameters")**

Get the parameters of servo memory map No.0 to No.59.

Argument: None

Return value:

VT_ARRAY | VT_UI1		Get the device version.
0	VT_UI1	Value of memory No.0
1	VT_UI1	Value of memory No.1
	:	
59	VT_UI1	Value of memory No. 59

Usage example(PacScript)

```
Dim vArray As Variant
vArray = extension.Execute("GetAllParameters")
```

2.2.5.2. **CaoExtension::Execute("InitServo")**

Initializes the servo memory map No.4 to No.29.

Argument: None

Return value:

VT_BOOL	Execution result TRUE: initialization succeeded FALSE: initialization failed

Usage example(PacScript)

```
extension.Execute("InitServo")
```

2.2.5.3. **CaoExtension::Execute("WriteFlashROM")**

Write and restart the servo flash ROM.

Argument: None

Return value:

VT_BOOL	Execution result TRUE: initialization succeeded FALSE: initialization failed

Usage example(PacScript)

```
extension.Execute("WriteFlashROM")
```

2.2.5.4. **CaoExtension::Execute("SendPacket")**

Send the set packet as it is to the servo.

For RS485 packets, refer to the manual for each servo.

Argument:

VT_ARRAY | VT_VARIANT			
0	VT_ARRAY | VT_UI1		Set the send command as an array of VT_ARRAY | VT_UI1 from the beginning.
	0	VT_UI1	Set value
	N	VT_UI1	Set value
1	VT_BOOL		Packet Receive Flag (Optional) FALSE: Do not receive packets from the servos (defaults) TRUE: Receiving packets from the servos

Return value:

VT_ARRAY | VT_UI1		When the packet reception flag is set to TRUE and a packet from the servo is received, the response command is started. Returns an array of VT_ARRAY | VT_UI1. For RS485 reply packets, refer to the manual for each servo.
0	VT_UI1	Response packet
N	VT_UI1	Response packet

Usage example(PacScript)

```
Dim vArray As Variant
vArray = extension.Execute("SendPacket", Array(Array(250, 0, 87, 0), 0))
```

2.2.6. CaoExtension::get_VariableNames method

Get the variable name lists in Table 2-6, Table 2-7, Table 2-8, Table 2-9, Table 2-10 and Table 2-11.

2.2.7. CaoExtension::AddVariable method

Creates a CaoVariable from a CaoExtension. Only the variable names listed in Table 2-6, Table 2-7, Table 2-8, Table 2-9, Table 2-10 and Table 2-11 can be used as variable names.

Show AddVariable format as follows.

FORMAT

```
AddVariable
(
        "<variable name>",        // Variable name
        "<option>",               // Option character string
)
```

For Option character string, see 2.3.2.

2.2.8. CaoVariable::get_Value Property

Get data from the device according to the specified options.

2.2.9. CaoVariable::put_Value Property

Write data to the device according to the specified options.

2.3. Variable list

2.3.1. Controller class

The following table lists the variables that can be specified in CaoController.

Table 2-5 List of controller class variables

Variable name	Description	Value		Link
		Get	Put	
@MAKER_NAME	Get the manufacturer's name.	✓	-	P.18
@VERSION	Get the version.	✓	-	P.18
ACK<??>	Execute the servo connection check.	✓	-	P.18
GOAL_POSITIONS<??>	Set or get the target position of the specified servo.	✓	✓	P.19
GOAL_TIMES<??>	Set or get the target time of the specified servo.	✓	✓	P.20
CURRENT_POSITIONS<??>	Gets the current position of the specified servo.	✓	-	P.20
CURRENT_TIMES<??>	Gets the current time of the specified servo.	✓	-	P.21
CURRENT_LOADS<??>	Gets the current load of the specified servo.	✓	-	P.21
CURRENT_TEMPERATURES<??>	Gets the current temperature of the specified servo.	✓	-	P.22

※<??> is any character

2.3.1.1. **@MAKER_NAME**

Get the manufacturer's name.

Option: None

Data structure of return value at get_value:

VT_BSTR	Fixed value: "Futaba"

2.3.1.2. **@VERSION**

Get the version.

Option: None

Data structure of return value at get_value:

VT_BSTR	The current DLL version, "*. *. * "String

2.3.1.3. **ACK<??>**

Execute the servo connection check.

Optional:

ID	Specifies the id of the servo to be read.	1 - 127

Data structure of return value at get_value:

VT_BOOL	TRUE: Connecting FALSE: Not connected

2.3.1.4. **GOAL_POSITIONS<??>**

Set or get the target position of the specified servo.

Optional:

IDS	Specify the ID of the servo to be connected. This option allows you to specify multiple ids. Examples.) ids=1@2@3@4 If at least one ID cannot be connected, the ID fails when get the value.	1 - 127

Data structure of return value at get_value:

VT_ARRAY | VT_I2	The servo can be moved to the indicated angle. The center of the movable range is 0 degrees, when viewed from the top surface of the servo (on the side with the nameplate), CW (clockwise) direction is "+" and CCW (counterclockwise) is "-". ● Acquired value/set value → Maximum limit value to minimum limit value for each servo The returned values are stored in the array in the order of id specified by the ids option.

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.1.5. **GOAL_TIMES<??>**

Set or get the target time of the specified servo.

Optional:

IDS	Specifies the ID of the servo to be connected. This option allows you to specify multiple ids. Examples.) ids=1@2@3@4 If at least one ID cannot be connected, the ID fails when the value is acquired.	1 - 127

Data structure of return value at get_value:

VT_ARRAY | VT_I2	You can set the servo movement time to the target position. Set in units of 10 ms. When the command value exceeds the maximum speed of the servo, it operates at the maximum speed. The returned values are stored in the array in the order of id specified by the ids option.

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.1.6. **CURRENT_POSITIONS<??>**

Gets the current position of the specified servo.

Optional:

IDS	Specifies the ID of the servo to be connected. This option allows you to specify multiple ids. Examples.) ids=1@2@3@4 If at least one ID cannot be connected, the ID fails when the value is acquired.	1 - 127

Data structure of return value at get_value:

VT_ARRAY | VT_I2	Returns the current position of each specified servo. The returned values are stored in the array in the order of id specified by the ids option.

2.3.1.7. **CURRENT_TIMES<??>**

Gets the current time of the specified servo.

Optional:

IDS	Specifies the ID of the servo to be connected. This option allows you to specify multiple ids. Examples.) ids=1@2@3@4 If at least one ID cannot be connected, the ID fails when the value is acquired.	1 - 127

Data structure of return value at get_value:

VT_ARRAY | VT_I2	Returns the current time of each specified servo. The returned values are stored in the array in the order of id specified by the ids option.

2.3.1.8. **CURRENT_LOADS<??>**

Gets the current load of the specified servo.

Optional:

IDS	Specifies the ID of the servo to be connected. This option allows you to specify multiple ids. Examples.) ids=1@2@3@4 If at least one ID cannot be connected, the ID fails when the value is acquired.	1 - 127

Data structure of return value at get_value:

VT_ARRAY | VT_I2	Returns the current load for each specified servo. The returned values are stored in the array in the order of id specified by the ids option.

2.3.1.9. **CURRENT_TEMPERATURES<??>**

Gets the current temperature of the specified servo.

Optional:

IDS	Specifies the ID of the servo to be connected. This option allows you to specify multiple ids. Examples.) ids=1@2@3@4 If at least one ID cannot be connected, the ID fails when the value is acquired.	1 - 127

Data structure of return value at get_value:

VT_ARRAY | VT_I2	Returns the current temperature for each specified servo. The returned values are stored in the array in the order of id specified by the ids option.

2.3.2. Extension class

Some variables can be used and some cannot be used depending on the device to which they are connected.

See the following chapters:

2.3.2.1. **Common variable**

The following table lists the common variables that can be specified in CaoExtenision.

Table 2-6 List of extension class common variables

Variable name	Description	Value		Link
		Get	Put	
@MODEL_NO	Gets the model number of the connected servo.	✓	-	P.25
@FIRMWARE_VERSION	Gets the firmware version of the connected servo.	✓	-	P.25
@BAUDRATE	Reads and writes the baud rate of the connected servo.	✓	✓	P.26
@CW_ANGLE_LIMIT	Reads and writes the right limit angle of the connected servo.	✓	✓	P.27
@CCW_ANGLE_LIMIT	Reads and writes the left limit angle of the connected servo. ● Acquired value/set value	✓	✓	P.28
@TEMPERATURE_LIMIT	Gets the temperature limit value of the connected servo.	✓	-	P.28
@CW_MARGIN	Specifies the allowable range of the connected servo stop position. (CW)	✓	✓	P.2929
@CCW_MARGIN	Specifies the range to adjust the torque to return to the target position when the current position of the connected servo deviates from the target position. (CCW)	✓	✓	P.29
@CW_SLOPE	Specifies the range to adjust the torque to return to the target position when the current position of the connected servo is out of alignment with the target position. (CW)	✓	✓	P.30
@CCW_SLOPE	Specifies the range to adjust the torque to return to the target position when the current position of the connected servo is out of alignment with the target position. (CCW)	✓	✓	P.30
@PUNCH	The minimum current applied to the internal motor when driving the connected servo can be set.	✓	✓	P.31
@GOAL_POS	The servo can be moved to the specified angle. The center of the movable range is 0 degrees, and the CW (clockwise rotation) direction is "+" and CCW (counterclockwise rotation) is "-" when viewed from the servo top surface (side with nameplate).	✓	✓	P.31

@GOAL_TIME	The servo travel time to the target position can be set. Set in units of 10 ms. When the command value exceeds the maximum speed of the servo, it operates at the maximum speed.	✓	✓	P.32
@MAX_TORUQU	The maximum torque output by the servo can be set. The servo torque can be set as 100% in 1% increments.	✓	✓	P.32
@TORUQU	Torque of servo can be turned ON/ OFF.	✓	✓	P.32
@CURRENT_POS	Gets the current angle of the servo.	✓	-	P.33
@CURRENT_TIME	The current time is the elapsed time since the servo received the command and started moving. When the move is complete, the last time is retained.	✓	-	P.33
@CURRENT_LOAD	Expresses the load (current) of the servo in mA. Since the current supplied to the servo is measured, it does not become 0 even in the torque OFF status.	✓	-	P.33
@CURRENT_TEMPERATURE	Indicates the temperature on the board of the servo. The temperature sensor has an individual difference and has an error of about ±3°C.	✓	-	P.33

※<??> is any character

2.3.2.1.1. @MODEL_NO

Gets the model number of the connected servo.

Option: None

Data structure of return value at get_value:

VT_UI2	Model number ・24592 : RS601CR ・16464 : RS405CB ・16480 : RS406CB ・16400 : RS401CR ・16416 : RS402CD ・12336 : RS303MD ・12352 : RS304MD ・12304 : RS301CR ・12320 : RS302CR ・Other: Unknown servos

2.3.2.1.2. @FIRMWARE_VERSION

Gets the firmware version of the connected servo.

Option: None

Data structure of return value at get_value:

VT_UI1	Production version

2.3.2.1.3. @BAUDRATE

Reads and writes the baud rate of the connected servo.

Option: None

Data structure of return value at get_value:

VT_UI1	Baud rate 00H : 9,600bps 01H : 14,400bps 02H : 19,200bps 03H : 28,800bps 04H : 38,400bps 05H : 57,600bps 06H : 76,800bps 07H : 115,200bps 08H : 153,600bps 09H : 230,400bps 0AH : 460,800bps 0BH : 691,200bps 0CH : 1,382,400bps

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.4. @CW_ANGLE_LIMIT

Reads and writes the right limit angle of the connected servo.

Option: None

Data structure of return value at get_value:

VT_UI2	Right limit angle ・RS301CR, RS302CD, RS303MR, RS304MD, RS405CB, RS406CB → 0 ~ 1500 ・RS401CR, RS402CD → 0 ~ 900 ・RS601CR → 0 ~ 1230 ・Other → It operates at the set value. (If a value greater than or equal to the maximum value specified for each servo is set, the servo may be damaged.)

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.5. @CCW_ANGLE_LIMIT

Reads and writes the left limit angle of the connected servo.

Option: None

Data structure of return value at get_value:

VT_I2	Left limit angle ・RS301CR, RS302CD, RS303MR, RS304MD, RS405CB, RS406CB → 0 ~ -1500 ・RS401CR, RS402CD → 0 ~ -900 ・RS601CR → 0 ~ -1230 ・Other → It operates at the set value. (If a value greater than or equal to the maximum value specified for each servo is set, the servo may be damaged.)

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.6. @TEMPERATURE_LIMIT

Gets the temperature limit value of the connected servo.

Option: None

Data structure of return value at get_value:

VT_UI2	Temperature limit

2.3.2.1.7. @CW_MARGIN

Specifies the allowable range of the connected servo stop position. (CW)

Option: None

Data structure of return value at get_value:

VT_UI1	Right compliance margin ・RS601CR, RS401CR, RS402CD (units: 0.08 degrees) → 00H to FFH (Approx. 20.4 degrees) ・Other than the above (unit: 0.1 degree) → 00H to FFH (Approx. 25.5 degrees)

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.8. @CCW_MARGIN

Specifies the range to adjust the torque to return to the target position when the current position of the connected servo deviates from the target position. (CCW)

Option: None

Data structure of return value at get_value:

VT_UI1	・RS601CR, RS401CR, RS402CD (units: 0.08 degrees) → 00H to FFH (Approx. 20.4 degrees) ・Other than the above (unit: 0.1 degree) → 00H to FFH (Approx. 25.5 degrees)

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.9. @CW_SLOPE

Specifies the range to adjust the torque to return to the target position when the current position of the connected servo is out of alignment with the target position. (CW)

Option: None

Data structure of return value at get_value:

VT_UI1	Right compliance slope ・RS301CR, RS302CD → 00H ~ 96H ・Other than the above → 00H ~ FFH

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.10. @CCW_SLOPE

Specifies the range to adjust the torque to return to the target position when the current position of the connected servo is out of alignment with the target position. (CCW)

Option: None

Data structure of return value at get_value:

VT_I1	Left Compliance Rope ・RS601CR, RS401, RS402 → 00H ~ 6A5H ・Other than the above → 00H ~ 2710H

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.11. @PUNCH

The minimum current applied to the internal motor when driving the connected servo can be set.

Option: None

Data structure of return value at get_value:

VT_UI2	Punch ・RS601CR, RS401, RS402 → 00H ~ 6A5H ・Other than the above → 00H ~ 2710H

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.12. @GOAL_POS

The servo can be moved to the specified angle. The center of the movable range is 0 degrees, and the CW (clockwise) direction is "+" and the CCW (counterclockwise) direction is "-" when viewed from the top surface of the servo (side with nameplate).

Option: None

Data structure of return value at get_value:

VT_I2	Target position Maximum limit value to minimum limit value for each servo ※If a value greater than or equal to the maximum limit value for each servo is set, the servo operates up to the maximum limit value set in the servo.

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.13. @GOAL_TIME

You can set the servo movement time to the target position. Set in units of 10 ms.

When the command value exceeds the maximum speed of the servo, it operates at the maximum speed.

Optional:

Data structure of return value at get_value:

VT_UI2	Target time

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.14. @MAX_TORUQU

The maximum torque output by the servo can be set.

The servo torque can be set as 100% in 1% increments.

Option: None

Data structure of return value at get_value:

VT_UI1	Maximum number of torques 00H ~ 64H

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.15. @TORUQU

Torque of servo can be turned ON/ OFF.

Option: None

Data structure of return value at get_value:

VT_UI1	Torque On /Off ・0 : OFF ・1 : ON ・2 : Brake

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.1.16. @CURRENT_POS

Gets the current angle of the servo.

Option: None

Data structure of return value at get_value:

VT_I2	Current position

2.3.2.1.17. @CURRENT_TIME

The current time is the elapsed time after the servo receives a command and starts moving.

When the move is complete, the last time is retained.

Option: None

Data structure of return value at get_value:

VT_UI2	Current time

2.3.2.1.18. @CURRENT_LOAD

Expresses the load (current) of the servo in mA.

Since the current supplied to the servo is measured, it does not become 0 even in the torque OFF status.

Option: None

Data structure of return value at get_value:

VT_UI2	Current load

2.3.2.1.19. @CURRENT_TEMPERATURE

Indicates the temperature on the board of the servo.

The temperature sensor has an individual difference and has an error of about ±3°C.

Option: None

Data structure of return value at get_value:

VT_UI2	Current temperature

2.3.2.2. **RS601CR Only Variables**

The following table lists RS601CR specificvariables that can be specified in CaoExtenision.

Table 2-7 Dedicated extension class RS601CR variables

Variable name	Description	Value		Link
		Get	Put	
@ACCELERATION_TIME	The time of acceleration and deceleration can be specified to smoothen the movement at the start and stop of the servo. Specify the time in units of 10 ms.	✓	✓	P.34
@LED	You can turn on/off LED on RS601CR unit.	✓	✓	P.35
@PID_COEFFICIENT	The control coefficient of the motor can be changed.	✓	✓	P.35

2.3.2.2.1. @ACCELERATION_TIME

The time of acceleration and deceleration can be specified to smoothen the movement at the start and stop of the servo.

Specify the time in units of 10 ms.

Option: None

Data structure of return value at get_value:

VT_UI1	Acceleration/deceleration time (in units of 10ms) 00H ~ FFH

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.2.2. @LED

You can turn on/off LED on RS601CR unit.

Option: None

Data structure of return value at get_value:

VT_UI1	LED ・0 : OFF ・1 : Green LED ・2 : Red LED ・3 : Both LEDs

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.2.3. @PID_COEFFICIENT

The control coefficient of the motor can be changed.

Option: None

Data structure of return value at get_value:

VT_UI1	PID adjustments 00H ~ FFH

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.3. **RS405CB, 406CB-specific variables**

The following table lists RS405CB and 406CB-specific variables that can be specified with CaoExtenision.

Table 2-8 List of extension class RS405CB and 406CB parameters

Variable name	Description	Value		Link
		Get	Put	
@REVERSE	Indicates the rotation direction of the servo.	✓	✓	P.36
@RETURN_DELAY	Indicates the response delay time when a return packet is requested.	✓	✓	P.37
@DAMPER	When a large load is applied, such as when a heavy object is moved, hunting (a phenomenon that moves as a servo cramps) is hard to occur by suppressing overshoots, etc. due to inertia.	✓	✓	P.37
@CURRENT_SPEED	Gets the present rotational speed of the servo in deg/sec.	✓	-	P.38
@CURRENT_VOLT	Indicates the voltage of the power currently supplied to the servo in units of 10mV.	✓	-	P.38

2.3.2.3.1. @REVERSE

Indicates the rotation direction of the servo.

The initial value is forward at 00H and reverse at 01H.

Option: None

Data structure of return value at get_value:

VT_UI1	Invert ・00H: Forward ・01H: Invert

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.3.2. @RETURN_DELAY

Indicates the response delay time when a return packet is requested.

Wait 100µs after receiving data at setting 0, and then the servo issues a return packet.

The parameter No. 7 is in units of 001H = 50µs .

To set the reply delay time to 1ms, write 18 (12H).

(1ms=100µs+18x50µs)

Option: None

Data structure of return value at get_value:

VT_UI1	Reply delay time

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.3.3. @DAMPER

To overshoot due to inertia when heavy load is applied, such as when moving a heavy object

Hold it down to prevent hunting (a phenomenon in which the servo moves to cramp) from occurring.

Option: None

Data structure of return value at get_value:

VT_UI1	Damper 00H ~ FFH

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.3.4. @CURRENT_SPEED

Gets the present rotational speed of the servo in deg/sec.

Option: None

Data structure of return value at get_value:

VT_UI2	Current speed

2.3.2.3.5. @CURRENT_VOLT

Indicates the voltage of the power currently supplied to the servo in units of 10mV .

Option: None

Data structure of return value at get_value:

VT_UI2	Current voltage

2.3.2.4. **RS405CB, 402CD only**

The following table lists RS405CB and 402CD-only variables that can be specified with CaoExtenision.

Table 2-9 List of extension class RS405CB and 402CD-only variables

Variable name	Description	Value		Link
		Get	Put	
@ACCELERATION_TIME	The time of acceleration and deceleration can be specified to smoothen the movement at the start and stop of the servo.	✓	✓	P.39
@PID_COEFFICIENT	The control coefficient of the motor can be changed.	✓	✓	P.39

2.3.2.4.1. @ACCELERATION_TIME

The time of acceleration and deceleration can be specified to smoothen the movement at the start and stop of the servo.

Option: None

Data structure of return value at get_value:

VT_UI1	Acceleration and Deceleration time 00H ~ FFH

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.4.2. @PID_COEFFICIENT

The control coefficient of the motor can be changed.

Option: None

Data structure of return value at get_value:

VT_UI1	PID adjustment value 00H ~ FFH

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.5. **RS303MR, 304MD-only variable**

The following table lists RS303MR and 304MD-specific variables that can be specified with CaoExtenision.

Table 2-10 List of extension class RS303MR and 304MD-only variables

Variable name	Description	Value		Link
		Get	Put	
@REVERSE	Indicates the rotation direction of the servo.	✓	✓	P. 40
@RETURN_DELAY	Indicates the response delay time when a return packet is requested.	✓	✓	P.41
@CURRENT_SPEED	Gets the present rotational speed of the servo in deg/sec.	✓	-	P.41
@CURRENT_VOLT	Indicates the voltage of the power currently supplied to the servo in units of 10mV .	✓	-	P.41

2.3.2.5.1. @REVERSE

Indicates the rotation direction of the servo.

The initial value is forward at 00H and reverse at 01H.

Option: None

Data structure of return value at get_value:

VT_UI1	Invert ・00H: Forward ・01H: Invert

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.5.2. @RETURN_DELAY

Indicates the response delay time when a return packet is requested.

Wait 100μs after receiving data at setting 0, and then the servo issues a return packet.

The parameter No. 7 is in units of 001H = 50μs.

To set the reply delay time to 1ms, write 18 (12H).

(1ms=100μs+18x50μs)

Option: None

Data structure of return value at get_value:

VT_UI1	Reply delay time

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.5.3. @CURRENT_SPEED

Gets the present rotational speed of the servo in deg/sec.

Option: None

Data structure of return value at get_value:

VT_UI2	Current speed

2.3.2.5.4. @CURRENT_VOLT

Indicates the voltage of the power currently supplied to the servo in units of 10mV.

Option: None

Data structure of return value at get_value:

VT_UI2	Current voltage

2.3.2.6. **RS301CR, 302CD-only**

The following table lists RS303MR and 304MD-specific variables that can be specified with CaoExtenision.

Table 2-11 List of extension class RS301CR and 302CD-only variables

Variable name	Description	Value		Link
		Get	Put	
@REVERSE	Indicates the rotation direction of the servo.	✓	✓	P. 42
@RETURN_DELAY	Indicates the response delay time when a return packet is requested.	✓	✓	P. 43
@CURRENT_SPEED	Gets the present rotational speed of the servo in deg/sec.	✓	-	P. 43
@CURRENT_VOLT	Indicates the voltage of the power currently supplied to the servo in units of 10mV.	✓	-	P. 43

2.3.2.6.1. @REVERSE

Indicates the rotation direction of the servo.

The initial value is forward at 00H and reverse at 01H.

Option: None

Data structure of return value at get_value:

VT_UI1	Invert ・00H: Forward ・01H: Invert

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.6.2. @RETURN_DELAY

Indicates the response delay time when a return packet is requested.

Wait 100µs after receiving data at setting 0, and then the servo issues a return packet.

The parameter No. 7 is in units of 001H = 50µs.

To set the reply delay time to 1ms, write 18 (12H).

(1ms=100µs+18x50µs)

Option: None

Data structure of return value at get_value:

VT_UI1	Reply delay time

Data configuration of set values at put_value:

Same as the data structure of the return value of get_value

2.3.2.6.3. @CURRENT_SPEED

Gets the present rotational speed of the servo in deg/sec.

Option: None

Data structure of return value at get_value:

VT_UI2	Current speed

2.3.2.6.4. @CURRENT_VOLT

Indicates the voltage of the power currently supplied to the servo in units of 10mV.

Option: None

Data structure of return value at get_value:

VT_UI2	Current voltage

3. Error code

This provider has its own error code. Refer to Table 3-1 Unique error codes below for details.

For ORiN2 common errors, refer to the Error Codes section of ORiN2 Programming Guide.

Table 3-1 Unique error codes

Error Number	Description
0x80110001	Required option is not set.
0x80110002	The connection failed.
0x80110003	Packet is invalid.

4. Sample program

Describes the sample to perform data communication with the device.

List 3-1 **Sample.pcs**

```
Sub Main
        ' Object
        Dim caoCtrl as Object
        Dim caoExt as Object
        Dim caoCtrlVariable as Object
        Dim caoExtVariable as Object
        Dim AckValue as Object
        Dim ModelValue as Object

        ' Controller creation
        caoCtrl = cao.AddController("RSC-U485F", "CaoProv.Futaba.RSC-U485", "", "Conn=Com:1, Timeout=500")

        ' Extension creation
        caoExt = caoCtrl.AddExtension("10")

        ' Controller::Variable creation
         caoCtrlVariable = caoCtrl.AddVariable("ACK1", "id=1")

        'Get Value of Controller::Variable
         AckValue = caoCtrlVariable.Value

        ' Extension::Variable creation
         caoExtVariable = caoExt.AddVariable("@MODEL_NO", "")

        'Get Value of Extension::Variable
         ModelValue = caoExtVariable .Value

End Sub
```

5. Appendix

Specifications for each servo are described below.

■ RS601CR

Communication method	RS485 half duplex communication
Bits per second	115.2 [kbps] (Available in 9.6 [kbps] to 1.3 [Mbps])
Data bit [bit]	8
Parity	None
Stop bit [bit]	1
Flow control	None

■ RS405CB/RS406CB

Communication method	RS485 half duplex communication
Bits per second	115.2 [kbps] (Can be set from 9.6 [kbps] to 460.8 [Mbps])
Data bit [bit]	8
Parity	None
Stop bit [bit]	1
Flow control	None

■ RS401CR/RS402CD

Communication method	RS485 half duplex communication
Bits per second	115.2 [kbps] (Available in 9.6 [kbps] to 1.3 [Mbps])
Data bit [bit]	8
Parity	None
Stop bit [bit]	1
Flow control	None

■ RS303MR/RS304MD

Communication method	RS485 half duplex communication
Bits per second	115.2 [kbps] (Can be set from 9.6 [kbps] to 460.8 [Mbps])
Data bit [bit]	8
Parity	None
Stop bit [bit]	1
Flow control	None

■ RS301CR/RS302CD

Communication method	RS485 half duplex communication
Bits per second	115.2 [kbps] (Can be set from 9.6 [kbps] to 460.8 [Mbps])
Data bit [bit]	8
Parity	None
Stop bit [bit]	1
Flow control	None