

双葉電子工業 RSC-U485 プロバイダ

Version 1.0.0

ユーザーズ ガイド

June 20, 2018

備考：

【改版履歴】

バージョン	日付	内容
1.0.0	2018-01-23	初版.

【動作確認機種】

機種名	ファームウェアバージョン	注意事項
RSC-U485	*.*.*	

目次

1. はじめに	6
2. プロバイダの概要	7
2.1. 概要.....	8
2.2. メソッド・プロパティ.....	9
2.2.1. CaoWorkspace::AddController メソッド.....	9
2.2.2. CaoController::AddExtension メソッド.....	10
2.2.3. CaoController::get_VariableNames プロパティ.....	11
2.2.4. CaoController::AddVariable メソッド.....	12
2.2.5. CaoExtension:Execute メソッド.....	13
2.2.5.1. CaoExtension::Execute("GetAllParameters").....	14
2.2.5.2. CaoExtension::Execute("InitServo").....	14
2.2.5.3. CaoExtension::Execute("WriteFlashROM").....	15
2.2.5.4. CaoExtension::Execute("SendPacket").....	15
2.2.6. CaoExtension::get_VariableNames メソッド.....	16
2.2.7. CaoExtension::AddVariable メソッド.....	16
2.2.8. CaoVariable::get_Value プロパティ.....	17
2.2.9. CaoVariable::put_Value プロパティ.....	17
2.3. 変数一覧.....	17
2.3.1. コントローラクラス.....	17
2.3.1.1. @MAKER_NAME.....	18
2.3.1.2. @VERSION.....	18
2.3.1.3. ACK<??>.....	18
2.3.1.4. GOAL_POSITIONS<??>.....	19
2.3.1.5. GOAL_TIMES<??>.....	20
2.3.1.6. CURRENT_POSITIONS<??>.....	20
2.3.1.7. CURRENT_TIMES<??>.....	21
2.3.1.8. CURRENT_LOADS<??>.....	21
2.3.1.9. CURRENT_TEMPERATURES<??>.....	22
2.3.2. エクステンションクラス.....	23
2.3.2.1. 共通変数.....	23
2.3.2.1.1. @MODEL_NO.....	25
2.3.2.1.2. @FIRMWARE_VERSION.....	25
2.3.2.1.3. @BAUDRATE.....	26

2.3.2.1.4. @CW_ANGLE_LIMIT	27
2.3.2.1.5. @CCW_ANGLE_LIMIT.....	28
2.3.2.1.6. @TEMPERATURE_LIMIT.....	28
2.3.2.1.7. @CW_MARGIN	29
2.3.2.1.8. @CCW_MARGIN.....	29
2.3.2.1.9. @CW_SLOPE.....	30
2.3.2.1.10. @CCW_SLOPE.....	30
2.3.2.1.11. @PUNCH.....	31
2.3.2.1.12. @GOAL_POS	31
2.3.2.1.13. @GOAL_TIME	32
2.3.2.1.14. @MAX_TORUQU	32
2.3.2.1.15. @TORUQU	32
2.3.2.1.16. @CURRENT_POS	33
2.3.2.1.17. @CURRENT_TIME	33
2.3.2.1.18. @CURRENT_LOAD	33
2.3.2.1.19. @CURRENT_TEMPERATURE	33
2.3.2.2. RS601CR 専用変数	34
2.3.2.2.1. @ACCELERATION_TIME	34
2.3.2.2.2. @LED	35
2.3.2.2.3. @PID_COEFFICIENT.....	35
2.3.2.3. RS405CB,406CB 専用変数.....	36
2.3.2.3.1. @REVERSE	36
2.3.2.3.2. @RETURN_DELAY	37
2.3.2.3.3. @DAMPER	37
2.3.2.3.4. @CURRENT_SPEED.....	38
2.3.2.3.5. @CURRENT_VOLT.....	38
2.3.2.4. RS405CB,402CD 専用変数	39
@ACCELERATION_TIME	39
@PID_COEFFICIENT	39
2.3.2.4.1. @ACCELERATION_TIME	39
2.3.2.4.2. @PID_COEFFICIENT.....	39
2.3.2.5. RS303MR,304MD 専用変数	40
2.3.2.5.1. @REVERSE	40
2.3.2.5.2. @RETURN_DELAY	41
2.3.2.5.3. @CURRENT_SPEED.....	41
2.3.2.5.4. @CURRENT_VOLT.....	41
2.3.2.6. RS301CR,302CD 専用変数	42

2.3.2.6.1. @REVERSE	42
2.3.2.6.2. @RETURN_DELAY	43
2.3.2.6.3. @CURRENT_SPEED.....	43
2.3.2.6.4. @CURRENT_VOLT.....	43
3. エラーコード	44
4. サンプルプログラム.....	45
5. 付録.....	46

1. はじめに

本書は、双葉電子工業製シリアル変換器「RS-U485」と接続している RS485 形式のサーボの制御を行う、Futaba RSC-U485 プロバイダのユーザーズガイドです。

Futaba RSC-U485 プロバイダを用いれば、RSC-U485 デバイスと接続している RS485 形式のサーボの値の読み書きを簡単に行うことができます。

本書は、Futaba RSC-U485 プロバイダの機能と、実装されているメソッドについて説明します。

2. プロバイダの概要

Futaba RSC-U485 プロバイダは RS485 通信方式により RSC-U485 と接続し、RSC-U485 と接続されている各サーボに対して値の読み書きを行います。下図 2-1 が本プロバイダとデバイスの全体構成図になります。

なお、接続方法やドライバのインストールの詳細については RSC-U485 の取扱い説明書をご覧ください。

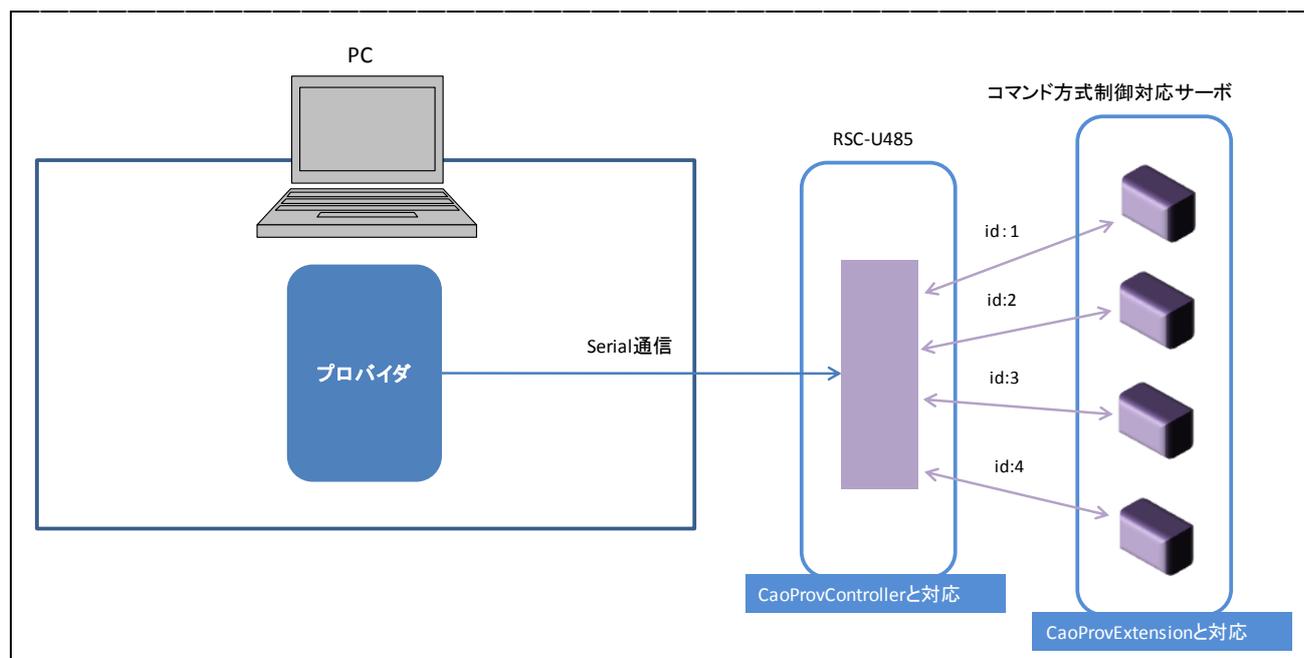


図 2-1 全体構成図

2.1. 概要

Futaba RSC-U485 プロバイダのファイル形式は DLL (Dynamic Link Library) となっており、その詳細は表 2-1 のようになっています。

表 2-1 Futaba RSC-U485 プロバイダ

ファイル名	GaoProvFutabaRSC-U485.dll
ProgID	GaoProv. Futaba. RSC-U485
レジストリ登録	regsvr32 GaoProvFutabaRSC-U485.dll
レジストリ登録の抹消	regsvr32 /u GaoProvFutabaRSC-U485.dll

また、本プロバイダ及びデバイスそれぞれの対応を下図 2-2 に示します。図 2-2 の通り、CaoProvController クラスが RSC-U485 に対応し、CaoProvExtension クラスが RSC-U485 に接続しているサーボモータに対応します。

なお、各サーボモータの ID は一意の数値です。重複する ID が割り振られている場合は接続に失敗しますのでご注意ください。

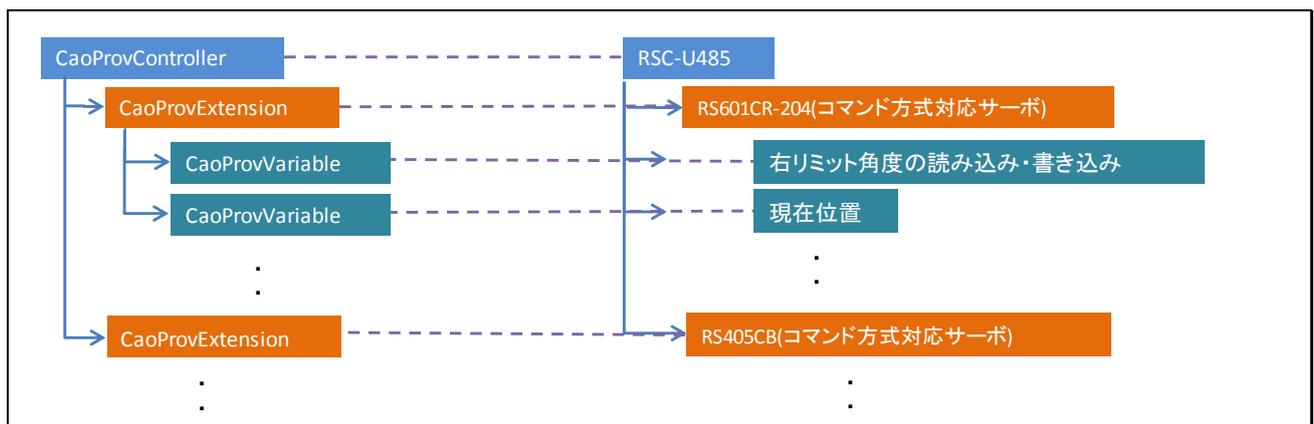


図 2-2 Futaba RSC-U485 プロバイダと RSC-U485 およびサーボのデータ対応図

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

Controller オブジェクトの生成を行います。

各オプションには、デバイスに接続するための情報を指定します。

以下に、AddController の仕様を示します。

書式

AddController

```
(
    “<コントローラ名>”,           // コントローラ名(任意)
    “CaoProv. Futaba. RSC-U485”,   // プロバイダ名(固定)
    “<マシン名>”,                 // プロバイダ実行マシン名(未使用)
    “<オプション>”,               // オプション文字列
)
```

以下にオプション文字列に指定する文字列を示します。

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

オプション	必須	説明	値範囲	デフォルト値
Conn	○	デバイスへの接続オプションを指定します。接続オプションは「Com:<ポート番号>」の形式で指定します。	--	--
Timeout	-	通信タイムアウトを ms 単位で指定します。	1 - 65535	500

使用例 (PacScript)

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

2.2.2. CaoController::AddExtension メソッド

Extension オブジェクトの生成を行います。

以下に、AddExtension の仕様を示します。

書式

AddExtension

```
(
    “<エクステンション名>”, // エクステンション名 (任意)
    “<オプション>”           // オプション文字列
)
```

以下にオプション文字列に指定する文字列を示します。

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

オプション	必須	説明	値範囲	デフォルト値
Id	○	接続するサーボの id 番号を指定します。	1-127	1

使用例 (PacScript)

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

2.2.3. CaoController::get_VariableNames プロパティ

表 2-5 に示す変数名リストを取得します。

以下に、GetVariableNames の仕様を示します。

書式

GetVariableNames

```
(  
    “<オプション>”           // オプション文字列(指定無し)  
)
```

使用例 (PacScript)

```
Dim variablelist As Object  
variablelist = controller.GetVariableNames(“”)
```

2.2.4. CaoController::AddVariable メソッド

CaoController から CaoVariable オブジェクトを生成します。変数名には、表 2-5 に示す変数名のみ使用できます。

以下に、AddVariable の仕様を示します。

書式

AddVariable

```
(  
    “<変数名>” ,           // 変数名  
    “<オプション>” ,       // オプション文字列  
)
```

オプション文字列については、2.3.1 を参照ください。

2.2.5. CaoExtension:Execute メソッド

CaoExtension オブジェクトにてメソッドを実行します。
引数には、メソッドに対して必要な引数を指定します。

以下に、Execute の仕様を示します。

書式

Execute

```
(
    “<メソッド名>”, // メソッド名
    “<引数>”       // 引数
)
```

以下に、Execute で指定できるメソッド一覧を示します。

表 2-4 CaoExtension::Execute メソッド一覧

コマンド名	説明	リンク
GetAllParameters	サーボのメモリマップ No.0~No.59 のパラメータを取得します。	P. 14
InitServo	サーボのメモリマップ No.4~No.29 を初期化します。	P. 14
WriteFlashROM	サーボのフラッシュ ROM 書き込みと再起動を行います。	P. 15
SendPacket	設定したパケットをそのままサーボに送信します。RS485 のパケットについては、サーボ毎のマニュアルを参照してください。	P. 15

2.2.5.1. CaoExtension::Execute("GetAllParameters")

サーボのメモリマップ No. 0~No. 59 のパラメータを取得します。

引数：なし

戻り値：

VT_ARRAY VT_UI1	デバイスのバージョンを取得します。
0 VT_UI1	メモリ No. 0 の値
1 VT_UI1	メモリ No. 1 の値
	⋮
59 VT_UI1	メモリ No. 59 の値

使用例 (PacScript)

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

2.2.5.2. CaoExtension::Execute("InitServo")

サーボのメモリマップ No. 4~No. 29 を初期化します。

引数：なし

戻り値：

VT_BOOL	実行結果 TRUE: 初期化成功 FALSE: 初期化失敗
---------	-------------------------------------

使用例 (PacScript)

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

2.2.5.3. CaoExtension::Execute("WriteFlashROM")

サーボのフラッシュ ROM 書き込みと再起動を行います。

引数：なし

戻り値：

VT_BOOL	実行結果 TRUE: 初期化成功 FALSE: 初期化失敗
---------	-------------------------------------

使用例 (PacScript)

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

2.2.5.4. CaoExtension::Execute("SendPacket")

設定したパケットをそのままサーボに送信します。

RS485 のパケットについては、サーボ毎のマニュアルを参照してください。

引数：

VT_ARRAY VT_VARIANT	
0	VT_ARRAY VT_UI1 送信コマンドを先頭から VT_ARRAY VT_UI1 の配列で設定します。
	0 VT_UI1 設定値
	n VT_UI1 設定値
1	VT_BOOL パケット受信フラグ(オプション) FALSE:サーボからのパケットを受信しない(デフォルト) TRUE:サーボからのパケットを受信する

戻り値：

VT_ARRAY VT_UI1 パケット受信フラグを TRUE に設定し、サーボからのパケットを受信できた場合、応答コマンドを先頭から VT_ARRAY VT_UI1 の配列で返します。RS485 の応答パケットについては、サーボ毎のマニュアルを参照してください。	
0	VT_UI1 応答パケット
n	VT_UI1 応答パケット

使用例 (PacScript)

```
Dim vArray As Variant
```

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

2.2.6. CaoExtension::get_VariableNames メソッド

表 2-6, 表 2-7, 表 2-8, 表 2-9, 表 2-10, 表 2-11 の変数名リストを取得します。

2.2.7. CaoExtension::AddVariable メソッド

CaoExtension から CaoVariable オブジェクトを生成します。変数名には、表 2-6, 表 2-7, 表 2-8, 表 2-9, 表 2-10, 表 2-11 に示す変数名のみ使用できます。

以下に、AddVariable の仕様を示します。

書式

AddVariable

```
(  
    “<変数名>”,           // 変数名  
    “<オプション>”,       // オプション文字列  
)
```

オプション文字列については、2.3.2 を参照ください。

2.2.8. CaoVariable::get_Value プロパティ

指定したオプションによってデバイスからデータを取得します。

2.2.9. CaoVariable::put_Value プロパティ

指定したオプションによってデバイスへデータを書き込みます。

2.3. 変数一覧

2.3.1. コントローラクラス

以下に、CaoController で指定できる変数一覧を示します。

表 2-5 コントローラクラス 変数一覧

変数名	説明	Value		リンク
		get	put	
@MAKER_NAME	メーカー名を取得します。	○	-	P. 18
@VERSION	バージョンを取得します。	○	-	P. 18
ACK<??>	サーボの接続確認を実行します。	○	-	P. 18
GOAL_POSITIONS<??>	指定したサーボの目標位置を設定、取得します。	○	○	P. 19
GOAL_TIMES<??>	指定したサーボの目標時間を設定、取得します。	○	○	P. 20
CURRENT_POSITIONS<??>	指定したサーボの現在位置を取得します。	○	-	P. 20
CURRENT_TIMES<??>	指定したサーボの現在時間を取得します。	○	-	P. 21
CURRENT_LOADS<??>	指定したサーボの現在負荷を取得します。	○	-	P. 21
CURRENT_TEMPERATURES<??>	指定したサーボの現在温度を取得します。	○	-	P. 22

※<??>は任意の文字

2.3.1.1. @MAKER_NAME

メーカー名を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_BSTR	固定値：“Futaba”
---------	--------------

2.3.1.2. @VERSION

バージョンを取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_BSTR	現在の DLL のバージョンを表す“*. *.*”文字列
---------	------------------------------

2.3.1.3. ACK<??>

サーボの接続確認を実行します。

オプション：

ID	読み込むサーボの id を指定します。	1 - 127
----	---------------------	---------

get_value 時の戻り値のデータ構成：

VT_BOOL	TRUE: 接続中 FALSE: 未接続
---------	-------------------------

2.3.1.4. GOAL_POSITIONS<??>

指定したサーボの目標位置を設定，取得します。

オプション：

IDS	接続するサーボの id を指定します。 このオプションでは id の複数指定が可能です。 例.) ids=1@2@3@4 一つでも接続できない id があつた場合， 値取得時に失敗します。	1 - 127
-----	--	---------

get_value 時の戻り値のデータ構成：

VT_ARRAY VT_I2	サーボを指示した角度へ動かすことができます。 可動範囲の中央が 0 度で，サーボ上面（銘板のある側）から見て， CW(時計回転)方向が「+」，CCW(反時計回転)が「-」です。 ●取得値/設定値 → サーボ毎の最大リミット値 ~ 最小リミット値 返ってくる値は，ids オプションで指定した id の順に配列に格納されています。
------------------	---

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.1.5. GOAL_TIMES<??>

指定したサーボの目標時間を設定，取得します。

オプション：

IDS	接続するサーボの id を指定します。 このオプションでは id の複数指定が可能です。 例.) ids=1@2@3@4 一つでも接続できない id があつた場合， 値取得時に失敗します。	1 - 127
-----	--	---------

get_value 時の戻り値のデータ構成：

VT_ARRAY VT_I2	目標位置までのサーボ移動時間を設定できます。10ms 単位で設定します。 指令値がサーボの最高速度を超える設定の場合は最高速度で動作します。 返ってくる値は，ids オプションで指定した id の順に配列に格納されています。
------------------	--

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.1.6. CURRENT_POSITIONS<??>

指定したサーボの現在位置を取得します。

オプション：

IDS	接続するサーボの id を指定します。 このオプションでは id の複数指定が可能です。 例.) ids=1@2@3@4 一つでも接続できない id があつた場合， 値取得時に失敗します。	1 - 127
-----	--	---------

get_value 時の戻り値のデータ構成：

VT_ARRAY VT_I2	指定したサーボ毎の現在位置を返します。 返ってくる値は，ids オプションで指定した id の順に配列に格納されています。
------------------	--

2.3.1.7. CURRENT_TIMES<??>

指定したサーボの現在時間を取得します。

オプション :

IDS	接続するサーボの id を指定します。 このオプションでは id の複数指定が可能です。 例.) ids=1@2@3@4 一つでも接続できない id があった場合、 値取得時に失敗します。	1 - 127
-----	--	---------

get_value 時の戻り値のデータ構成 :

VT_ARRAY VT_I2	指定したサーボ毎の現在時間を返します。 返ってくる値は、ids オプションで指定した id の順に配列に格納されています。
------------------	--

2.3.1.8. CURRENT_LOADS<??>

指定したサーボの現在負荷を取得します。

オプション :

IDS	接続するサーボの id を指定します。 このオプションでは id の複数指定が可能です。 例.) ids=1@2@3@4 一つでも接続できない id があった場合、 値取得時に失敗します。	1 - 127
-----	--	---------

get_value 時の戻り値のデータ構成 :

VT_ARRAY VT_I2	指定したサーボ毎の現在負荷を返します。 返ってくる値は、ids オプションで指定した id の順に配列に格納されています。
------------------	--

2.3.1.9. CURRENT_TEMPERATURES<??>

指定したサーボの現在温度を取得します。

オプション :

IDS	接続するサーボの id を指定します。 このオプションでは id の複数指定が可能です。 例.) ids=1@2@3@4 一つでも接続できない id があった場合、 値取得時に失敗します。	1 - 127
-----	--	---------

get_value 時の戻り値のデータ構成 :

VT_ARRAY VT_12	指定したサーボ毎の現在温度を返します。 返ってくる値は、ids オプションで指定した id の順に配列に格納されています。
------------------	--

2.3.2. エクステンションクラス

接続先のデバイスによって使用できる変数と使用できない変数があります。
以下の各章を参照してください。

2.3.2.1. 共通変数

以下に、CaoExtension で指定できる共通変数一覧を示します。

表 2-6 エクステンションクラス 共通変数一覧

変数名	説明	Value		リンク
		get	put	
@MODEL_NO	接続したサーボのモデル番号を取得します。	○	-	P. 25
@FIRMWARE_VERSION	接続したサーボのファームウェアバージョンを取得します。	○	-	P. 25
@BAUDRATE	接続したサーボのボーレートの読み書きを行います。	○	○	P. 26
@CW_ANGLE_LIMIT	接続したサーボの右リミット角度の読み書きを行います。	○	○	P. 27
@CCW_ANGLE_LIMIT	接続したサーボの左リミット角度の読み書きを行います。 ● 取得値/設定値	○	○	P. 28
@TEMPERATURE_LIMIT	接続したサーボの温度リミット値を取得します。	○	-	P. 28
@CW_MARGIN	接続したサーボ停止位置の許容範囲を指定します。(CW)	○	○	P. 29
@CCW_MARGIN	接続したサーボの、現在位置が目標位置とずれている時に、目標位置へ戻ろうとするトルクを調整する範囲を指定します。(CCW)	○	○	P. 29
@CW_SLOPE	接続したサーボの現在位置が目標位置とずれている時に、目標位置へ戻ろうとするトルクを調整する範囲を指定します。(CW)	○	○	P. 30
@CCW_SLOPE	接続したサーボの現在位置が目標位置とずれている時に、目標位置へ戻ろうとするトルクを調整する範囲を指定します。(CCW)	○	○	P. 30
@PUNCH	接続したサーボを駆動するときに、内部のモータにかける最小電流を設定できます。	○	○	P. 31
@GOAL_POS	サーボを指示した角度へ動かすことが出来ます。可動範囲の中央が 0 度で、サーボ上面(銘板のある側)から見て、CW(時計回転)方向が「+」、CCW(反時計回転)が「-」です。	○	○	P. 31

@GOAL_TIME	目標位置までのサーボ移動時間を設定できます。 10ms 単位で設定します。 指令値がサーボの最高速度を超える設定の場合は最高速度で動作します。	○	○	P. 32
@MAX_TORQU	サーボが出力する最大トルクを設定できます。 サーボのトルクを 100%として 1%単位で設定できます。	○	○	P. 32
@TORQU	サーボのトルクを ON, OFF できます。	○	○	P. 32
@CURRENT_POS	サーボの現在の角度を取得します。	○	-	P. 33
@CURRENT_TIME	現在時間は、サーボが指令を受信し、移動を開始してからの経過時間です。移動が完了すると最後の時間を保持します。	○	-	P. 33
@CURRENT_LOAD	サーボの負荷(電流)を mA 単位で表します。 サーボに供給されている電流を計測しているため、トルク OFF 状態でも 0 にはなりません。	○	-	P. 33
@CURRENT_TEMPERATURE	サーボの基板上の温度を表します。 温度センサには個体差があり、おおよそ±3°C程度の誤差があります。	○	-	P. 33

※<??>は任意の文字

2.3.2.1.1. @MODEL_NO

接続したサーボのモデル番号を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	モデル番号
	・ 24592 : RS601CR
	・ 16464 : RS405CB
	・ 16480 : RS406CB
	・ 16400 : RS401CR
	・ 16416 : RS402CD
	・ 12336 : RS303MD
	・ 12352 : RS304MD
	・ 12304 : RS301CR
	・ 12320 : RS302CR
・ 上記以外 : 未知のサーボ	

2.3.2.1.2. @FIRMWARE_VERSION

接続したサーボのファームウェアバージョンを取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	製造時のバージョン
--------	-----------

2.3.2.1.3. @BAUDRATE

接続したサーボのボーレートの読み書きを行います。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	ボーレート
	・ 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	

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.1.4. @CW_ANGLE_LIMIT

接続したサーボの右リミット角度の読み書きを行います。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	<p>右リミット角度</p> <ul style="list-style-type: none"> ・ RS301CR, RS302CD, RS303MR, RS304MD, RS405CB, RS406CB → 0 ~ 1500 ・ RS401CR, RS402CD → 0 ~ 900 ・ RS601CR → 0 ~ 1230 ・ その他 → 設定された値で動作します。 <p>(サーボ毎に既定されている最大値以上の値を設定した場合、サーボが破損する可能性があります。)</p>
--------	---

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.1.5. @CCW_ANGLE_LIMIT

接続したサーボの左リミット角度の読み書きを行います。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_I2	左リミット角度 ・ RS301CR, RS302CD, RS303MR, RS304MD, RS405CB, RS406CB → 0 ~ -1500 ・ RS401CR, RS402CD → 0 ~ -900 ・ RS601CR → 0 ~ -1230 ・ その他 → 設定された値で動作します。 (サーボ毎に既定されている最大値以上の値を設定した場合、サーボが破損する可能性があります。)
-------	--

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.1.6. @TEMPERATURE_LIMIT

接続したサーボの温度リミット値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	温度リミット
--------	--------

2.3.2.1.7. @CW_MARGIN

接続したサーボ停止位置の許容範囲を指定します。(CW)

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	右コンプライアンスマージン ・ RS601CR, RS401CR, RS402CD(単位：0.08 度) → 00H ~ FFH(約 20.4 度) ・ 上記以外(単位：0.1 度) → 00H ~ FFH(約 25.5 度)
--------	---

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.1.8. @CCW_MARGIN

接続したサーボの、現在位置が目標位置とずれている時に、目標位置へ戻ろうとするトルクを調整する範囲を指定します。(CCW)

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	・ RS601CR, RS401CR, RS402CD(単位：0.08 度) → 00H ~ FFH(約 20.4 度) ・ 上記以外(単位：0.1 度) → 00H ~ FFH(約 25.5 度)
--------	--

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.1.9. @CW_SLOPE

接続したサーボの現在位置が目標位置とずれている時に、目標位置へ戻ろうとするトルクを調整する範囲を指定します。(CW)

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	右コンプライアンスロープ ・ RS301CR, RS302CD → 00H ~ 96H ・ 上記以外 → 00H ~ FFH
--------	--

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.1.10. @CCW_SLOPE

接続したサーボの現在位置が目標位置とずれている時に、目標位置へ戻ろうとするトルクを調整する範囲を指定します。(CCW)

オプション：なし

get_value 時の戻り値のデータ構成：

VT_I1	左コンプライアンスロープ ・ RS601CR, RS401, RS402 → 00H ~ 6A5H ・ 上記以外 → 00H ~ 2710H
-------	--

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.1.11. @PUNCH

接続したサーボを駆動するときに、内部のモータにかける最小電流を設定できます。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	パンチ ・ RS601CR, RS401, RS402 → 00H ~ 6A5H ・ 上記以外 → 00H ~ 2710H
--------	---

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.1.12. @GOAL_POS

サーボを指示した角度へ動かすことができます。可動範囲の中央が 0 度で、サーボ上面（銘板のある側）から見て、CW(時計回転)方向が「+」、CCW(反時計回転)が「-」です

オプション：なし

get_value 時の戻り値のデータ構成：

VT_I2	目標位置 サーボ毎の最大リミット値 ~ 最小リミット値 ※サーボ毎の最大リミット値以上の値を設定した場合は、サーボに設定された最大リミット値まで動作します。
-------	--

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.1.13. @GOAL_TIME

目標位置までのサーボ移動時間を設定できます。10ms 単位で設定します。
 指令値がサーボの最高速度を超える設定の場合は最高速度で動作します。

オプション :

get_value 時の戻り値のデータ構成 :

VT_UI2	目標時間
--------	------

put_value 時の設定値のデータ構成 :

get_value の戻り値のデータ構成と同じ

2.3.2.1.14. @MAX_TORUQU

サーボが出力する最大トルクを設定できます。
 サーボのトルクを 100%として 1%単位で設定できます。

オプション : なし

get_value 時の戻り値のデータ構成 :

VT_UI1	最大トルク数 00H ~ 64H
--------	---------------------

put_value 時の設定値のデータ構成 :

get_value の戻り値のデータ構成と同じ

2.3.2.1.15. @TORUQU

サーボのトルクを ON, OFF できます。

オプション : なし

get_value 時の戻り値のデータ構成 :

VT_UI1	トルク On /Off <ul style="list-style-type: none"> ・ 0 : OFF ・ 1 : ON ・ 2 : Brake
--------	---

put_value 時の設定値のデータ構成 :

get_value の戻り値のデータ構成と同じ

2.3.2.1.16. @CURRENT_POS

サーボの現在の角度を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_I2	現在位置
-------	------

2.3.2.1.17. @CURRENT_TIME

現在時間はサーボが指令を受信し、移動を開始してからの経過時間です。

移動が完了すると最後の時間を保持します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	現在時間
--------	------

2.3.2.1.18. @CURRENT_LOAD

サーボの負荷(電流)を mA 単位で表します。

サーボに供給されている電流を計測しているため、トルク OFF 状態でも 0 にはなりません。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	現在負荷
--------	------

2.3.2.1.19. @CURRENT_TEMPERATURE

サーボの基板上の温度を表します。

温度センサには個体差があり、おおよそ±3°C程度の誤差があります。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	現在温度
--------	------

2.3.2.2. RS601CR 専用変数

以下に、CaoExtension で指定できる RS601CR 専用変数一覧を示します。

表 2-7 エクステンションクラス RS601CR 専用変数一覧

変数名	説明	Value		リンク
		get	put	
@ACCELERATION_TIME	加速と減速の時間を指定して、サーボの動き始めと停止時の動きを滑らかにできます。 10ms 単位の時間で指定します。	○	○	P. 34
@LED	RS601CR 本体の LED を点灯/消灯させることができます。	○	○	P. 35
@PID_COEFFICIENT	モータの制御係数を変更することができます。	○	○	P. 35

2.3.2.2.1. @ACCELERATION_TIME

加速と減速の時間を指定して、サーボの動き始めと停止時の動きを滑らかにできます。
10ms 単位の時間で指定します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	加減速時間(10ms 単位) 00H ~ FFH
--------	-----------------------------

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.2.2. @LED

RS601CR 本体の LED を点灯/消灯させることができます.

オプション : なし

get_value 時の戻り値のデータ構成 :

VT_UI1	LED ・ 0 : OFF ・ 1 : 緑色 LED ・ 2 : 赤色 LED ・ 3 : 両方 LED
--------	--

put_value 時の設定値のデータ構成 :

get_value の戻り値のデータ構成と同じ

2.3.2.2.3. @PID_COEFFICIENT

モータの制御係数を変更することができます.

オプション : なし

get_value 時の戻り値のデータ構成 :

VT_UI1	PID 調整値 00H ~ FFH
--------	----------------------

put_value 時の設定値のデータ構成 :

get_value の戻り値のデータ構成と同じ

2.3.2.3. RS405CB,406CB 専用変数

以下に、CaoExtension で指定できる RS405CB, 406CB 専用変数一覧を示します。

表 2-8 エクステンションクラス RS405CB, 406CB 専用変数一覧

変数名	説明	Value		リンク
		get	put	
@REVERSE	サーボの回転方向を表します。	○	○	P. 36
@RETURN_DELAY	リターンパケットを要求された時の返信ディレイ時間を示します。	○	○	P. 37
@DAMPER	重たいものを動かす場合など大きい負荷がかかるときに、慣性によるオーバーシュート等を抑えてハンチング(サーボが痙攣するように動く現象)を起こりにくくします。	○	○	P. 37
@CURRENT_SPEED	サーボの現在回転スピードを deg/sec 単位で取得します。	○	-	P. 38
@CURRENT_VOLT	現在サーボに供給されている電源の電圧を 10mV 単位で示します。	○	-	P. 38

2.3.2.3.1. @REVERSE

サーボの回転方向を表します。

初期値は 00H で正転, 01H で反転になります。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	反転 ・ 00H：正転 ・ 01H：反転
--------	----------------------------

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.3.2. @RETURN_DELAY

リターンパケットを要求された時の返信ディレイ時間を示します。

設定 0 でデータ受信後 100 μ s 待ってから、サーボがリターンパケットを出します。

No. 7 のパラメータは 001H = 50 μ s の単位になります。

返信ディレイ時間を 1ms にしたい場合は 18 (12H) を書き込みます。

(1ms=100 μ s+18 x 50 μ s)

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	返信遅延時間
--------	--------

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.3.3. @DAMPER

重たいものを動かす場合など大きい負荷がかかるときに、慣性によるオーバーシュート等を抑えてハンチング（サーボが痙攣するように動く現象）を起こりにくくします。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	ダンパー 00H ~ FFH
--------	-------------------

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.3.4. @CURRENT_SPEED

サーボの現在回転スピードを deg/sec 単位で取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	現在スピード
--------	--------

2.3.2.3.5. @CURRENT_VOLT

現在サーボに供給されている電源の電圧を 10mV 単位で示します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	現在電圧
--------	------

2.3.2.4. RS405CB,402CD 専用変数

以下に、CaoExtension で指定できる RS405CB, 402CD 専用変数一覧を示します。

表 2-9 エクステンションクラス RS405CB, 402CD 専用変数一覧

変数名	説明	Value		リンク
		get	put	
@ACCELERATION_TIME	加速と減速の時間を指定して、サーボの動き始めと停止時の動きを滑らかにできます。	○	○	P. 39
@PID_COEFFICIENT	モータの制御係数を変更することができます。	○	○	P. 39

2.3.2.4.1. @ACCELERATION_TIME

加速と減速の時間を指定して、サーボの動き始めと停止時の動きを滑らかにできます。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	加減速時間 00H ~ FFH
--------	--------------------

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.4.2. @PID_COEFFICIENT

モータの制御係数を変更することができます。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	PID 調整値 00H ~ FFH
--------	----------------------

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.5. RS303MR,304MD 専用変数

以下に、CaoExtension で指定できる RS303MR, 304MD 専用変数一覧を示します。

表 2-10 エクステンションクラス RS303MR, 304MD 専用変数一覧

変数名	説明	Value		リンク
		get	put	
@REVERSE	サーボの回転方向を表します。	○	○	P. 40
@RETURN_DELAY	リターンパケットを要求された時の返信ディレイ時間を示します。	○	○	P. 41
@CURRENT_SPEED	サーボの現在回転スピードを deg/sec 単位で取得します。	○	-	P. 41
@CURRENT_VOLT	現在サーボに供給されている電源の電圧を 10mV 単位で示します。	○	-	P. 41

2.3.2.5.1. @REVERSE

サーボの回転方向を表します。

初期値は 00H で正転, 01H で反転になります。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	反転 ・ 00H：正転 ・ 01H：反転
--------	----------------------------

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.5.2. @RETURN_DELAY

リターンパケットを要求された時の返信ディレイ時間を示します。
 設定 0 でデータ受信後 100 μ s 待ってから、サーボがリターンパケットを出します。
 No. 7 のパラメータは 001H = 50 μ s の単位になります。
 返信ディレイ時間を 1ms にしたい場合は 18 (12H) を書き込みます。
 (1ms=100 μ s+18 x 50 μ s)

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	返信遅延時間
--------	--------

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.5.3. @CURRENT_SPEED

サーボの現在回転スピードを deg/sec 単位で取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	現在スピード
--------	--------

2.3.2.5.4. @CURRENT_VOLT

現在サーボに供給されている電源の電圧を 10mV 単位で示します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	現在電圧
--------	------

2.3.2.6. RS301CR,302CD 専用変数

以下に、CaoExtension で指定できる RS303MR, 304MD 専用変数一覧を示します。

表 2-11 エクステンションクラス RS301CR, 302CD 専用変数一覧

変数名	説明	Value		リンク
		get	put	
@REVERSE	サーボの回転方向を表します。	○	○	P. 42
@RETURN_DELAY	リターンパケットを要求された時の返信ディレイ時間を示します。	○	○	P. 43
@CURRENT_SPEED	サーボの現在回転スピードを deg/sec 単位で取得します。	○	-	P. 43
@CURRENT_VOLT	現在サーボに供給されている電源の電圧を 10mV 単位で示します。	○	-	P. 43

2.3.2.6.1. @REVERSE

サーボの回転方向を表します。

初期値は 00H で正転, 01H で反転になります。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	反転 ・ 00H：正転 ・ 01H：反転
--------	----------------------------

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.6.2. @RETURN_DELAY

リターンパケットを要求された時の返信ディレイ時間を示します。
 設定 0 でデータ受信後 100 μ s 待ってから、サーボがリターンパケットを出します。
 No. 7 のパラメータは 001H = 50 μ s の単位になります。
 返信ディレイ時間を 1ms にしたい場合は 18 (12H) を書き込みます。
 (1ms=100 μ s+18 x 50 μ s)

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI1	返信遅延時間
--------	--------

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.6.3. @CURRENT_SPEED

サーボの現在回転スピードを deg/sec 単位で取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	現在スピード
--------	--------

2.3.2.6.4. @CURRENT_VOLT

現在サーボに供給されている電源の電圧を 10mV 単位で示します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	現在電圧
--------	------

3. エラーコード

本プロバイダには、独自のエラーコードが存在します。詳細は以下の表 3-1 独自エラーコード表をご参照ください。

ORiN2 共通エラーについては、「ORiN2 プログラミングガイド」のエラーコードの章をご参照ください。

表 3-1 独自エラーコード表

エラー番号	説明
0x80110001	必須オプションが設定されていません。
0x80110002	接続に失敗した。
0x80110003	パケットが不正。

4. サンプルプログラム

デバイスとデータ通信を行うサンプルを記述します。

List 3-1 Sample.pcs

Sub Main

' オブジェクト

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オブジェクト作成

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

' Extensionオブジェクト作成

caoExt = caoCtrl.AddExtension("servo1", "Id=100")

' Controller::Variableオブジェクト作成

caoCtrlVariable = caoCtrl.AddVariable("ACK1", "id=1")

' Controller::VariableのValueを取得

AckValue = caoCtrlVariable.Value

' Extension::Variableオブジェクト作成

caoExtVariable = caoExt.AddVariable("@MODEL_NO", "")

' Extension::VariableのValueを取得

ModelValue = caoExtVariable.Value

End Sub

5. 付録

サーボ毎の仕様を以下に記載します。

■ RS601CR

通信方式	RS485 半二重通信
ビット/秒	115.2[kbps] (9.6[kbps]~1.3[Mbps]で設定可能)
データビット[bit]	8
パリティ	なし
ストップビット[bit]	1
フロー制御	なし

■ RS405CB/RS406CB

通信方式	RS485 半二重通信
ビット/秒	115.2[kbps] (9.6[kbps]~460.8[Mbps]で設定可能)
データビット[bit]	8
パリティ	なし
ストップビット[bit]	1
フロー制御	なし

■ RS401CR/RS402CD

通信方式	RS485 半二重通信
ビット/秒	115.2[kbps] (9.6[kbps]~1.3[Mbps]で設定可能)
データビット[bit]	8
パリティ	なし
ストップビット[bit]	1
フロー制御	なし

■ RS303MR/RS304MD

通信方式	RS485 半二重通信
ビット/秒	115.2[kbps] (9.6[kbps]~460.8[Mbps]で設定可能)
データビット[bit]	8
パリティ	なし
ストップビット[bit]	1
フロー制御	なし

■ RS301CR/RS302CD

通信方式	RS485 半二重通信
ビット/秒	115.2[kbps] (9.6[kbps]~460.8[Mbps]で設定可能)
データビット[bit]	8
パリティ	なし
ストップビット[bit]	1
フロー制御	なし