

YAMAHA
RCX3 プロバイダ
ユーザーズ ガイド

Version 1.0.0

January 13, 2022

備考:

© 2018 DENSO WAVE INCORPORATED

この取扱説明書の著作権は、株式会社デンソーウェーブにあります。
本書に掲載されている会社名や製品は、一般に各社の商標または登録商標です。

この取扱説明書の一部または全部を無断で複製・転載することはお断りします。

- この説明書の内容は将来予告なしに変更することがあります。
- 本書の内容については、万全を期して作成いたしましたが、万一ご不審の点や誤り、記載もれなど、お気づきの点がありましたらご連絡ください。
- 運用した結果の影響については、上項にかかわらず責任を負いかねますのでご了承ください。

【改版履歴】

バージョン	日付	内容
1.0.0	2022-01-13	初版.

【対応機種】

機種	バージョン	注意事項
RCX320		
RCX340		

【動作確認機種】

機種	バージョン	注意事項
RCX340	V1.83	RCX-Studio 2020 上のエミュレータにて確認

目次

1. はじめに.....	6
1.1. 参考となる情報源.....	7
2. アプリケーション開発のための環境セットアップ.....	8
2.1. 非常停止スイッチの設置.....	8
2.2. RCX3 シリーズコントローラとクライアント PC との接続.....	8
2.2.1. RCX3 シリーズの通信設定.....	9
2.3. PC 開発環境のセットアップ.....	10
2.3.1. RCX3 プロバイダの手動インストール.....	10
3. コマンドリファレンス.....	11
3.1. メソッド/プロパティ一覧.....	11
3.2. メソッド・プロパティ.....	11
3.2.1. CaoWorkspace クラス.....	11
3.2.1.1. AddController メソッド.....	11
3.2.2. CaoController クラス.....	14
3.2.2.1. AddRobot メソッド.....	14
3.2.3. CaoRobot クラス.....	15
3.2.3.1. ID プロパティ.....	15
3.2.3.2. VariableNames プロパティ.....	16
3.2.3.3. Variables プロパティ.....	16
3.2.3.4. AddVariable メソッド.....	16
3.2.3.5. Accelerate メソッド.....	16
3.2.3.6. Change メソッド.....	17
3.2.3.7. Halt メソッド.....	18
3.2.3.8. Move メソッド.....	19
3.2.3.9. Speed メソッド.....	20
3.2.3.10. Execute メソッド.....	21
3.2.4. CaoVariable クラス.....	21
3.2.4.1. Value プロパティ.....	21
3.3. 拡張コマンド一覧.....	22
3.3.1. CaoRobot クラス拡張コマンド.....	22

3.3.1.1. PMOVE コマンド	22
3.3.1.2. SERVO コマンド	22
3.4. 変数一覧	23
3.4.1. CaoRobot クラス変数	23
3.4.1.1. @WHERE	23
3.4.1.2. @WHRXY	24
4. RCX3 プロバイダによるプログラミング	26
4.1. ロボットのアーム現在位置(パルス値)を取得するサンプルプログラミング	26
4.1.1. サンプルプログラム	27
4.1.1.1. 前処理	29
4.1.1.2. ロボットのアーム現在位置取得	31
4.1.1.3. 後処理	31
4.2. ポイントデータを使用しロボットを移動させるサンプルプログラミング	33
4.2.1. サンプルプログラム	33
4.2.1.1. 前処理	35
4.2.1.2. ロボットのサーボ状態を ON	36
4.2.1.3. ロボットを移動	37
4.2.1.4. 後処理	37
5. RCX3 プロバイダエラーコード	38
6. 付録	39
6.1. RCX3 コマンド対応表	39
6.1.1. CaoRobot クラス	39
6.2. オプションの種類	40
6.2.1. 相対速度指定(SPEED)	40
6.2.2. 絶対速度指定(DSPEED)	41
6.2.3. 直線速度指定(VEL)	41
6.2.4. アーチモーション	41
6.2.5. CONT 指定(連結動作)	42
6.2.6. 加速度指定	42
6.2.7. 減速度指定	43
6.2.8. 座標平面指定	43
6.2.9. ポート出力指定	43

1. はじめに

本書は、ヤマハ発動機株式会社のヤマハロボットコントローラ RCX3 シリーズ用の CAO プロバイダのユーザーズガイドです。以降本プロバイダを RCX3 プロバイダと呼称します。また、ヤマハ発動機株式会社のヤマハロボットコントローラ RCX3 シリーズを RCX3 シリーズと呼称します。RCX3 プロバイダは RCX3 シリーズと、シリアル通信又は Ethernet 通信の 2 種類の通信方法にて通信を行います。シリアル通信では RS-232C を使用し、Ethernet 通信では Telnet 通信を使用して通信を行います。

図 1-1 が本プロバイダとデバイスの全体構成図になります。

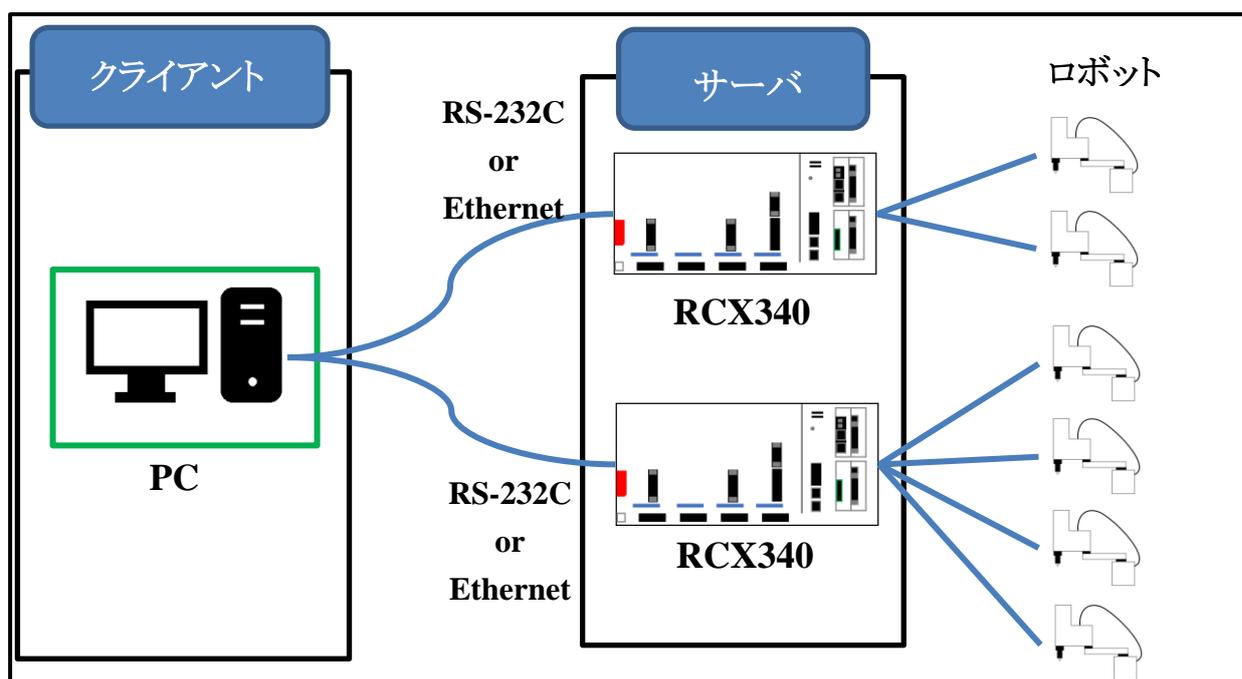


図 1-1 構成図

また、本プロバイダ及びデバイスそれぞれの対応を図 1-2 に表します。

(※一例です。全てを表しているわけではありません。)

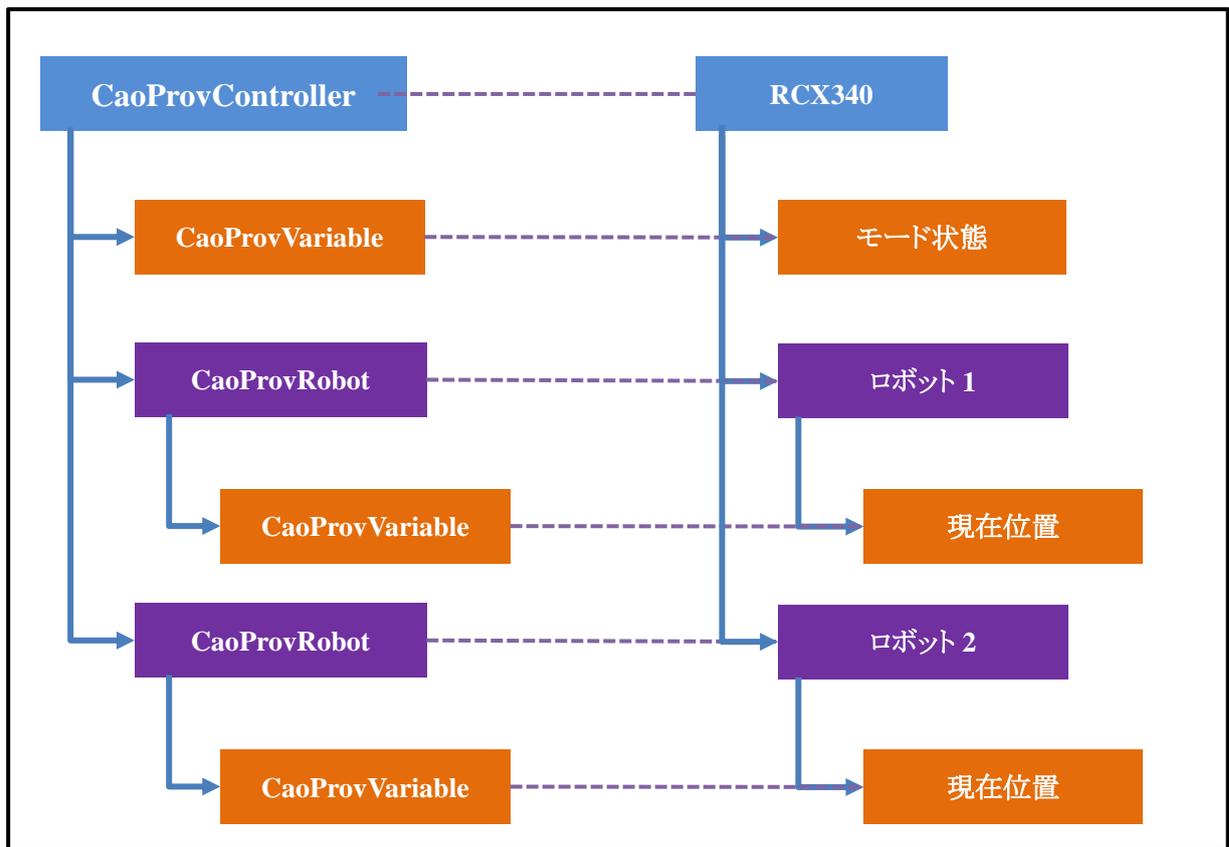


図 1-2 プロバイダの構成とデバイス情報との対応図

1.1. 参考となる情報源

RCX3 プロバイダは、ヤマハ発動機株式会社の「RCX3_プログラミングマニュアル」を参考に開発しております。以降このマニュアルを、RCX3 マニュアルと呼称します。

2. アプリケーション開発のための環境セットアップ

2.1. 非常停止スイッチの設置

ロボットコントローラを使用になる前に、非常の際にただちにロボットの運転を停止できるよう、作業者が容易に操作できる位置に非常停止スイッチを設置してください。

- (1) 非常停止スイッチは、赤色にしてください。
- (2) 非常停止の機能は、作動させたあと自動的に復帰せず、また他の作業者が不用意に復帰させることができないようにしてください。
- (3) 非常停止スイッチは、電源スイッチとは別個に設けてください。

2.2. RCX3 シリーズコントローラとクライアント PC との接続

RCX3 プロバイダは RCX3 シリーズと RS-232C 通信または Ethernet 通信 (Telnet 通信) を利用して接続を行います。RCX3 シリーズとクライアント PC を RS-232C ケーブルまたは LAN ケーブルにて接続してください。RCX3 シリーズとの接続方法の詳細については、ヤマハ発動機株式会社の「RCX3_ユーザーズマニュアル」の「第 6 章 外部通信インターフェース」を参照してください。以降このマニュアルを、RCX3 ユーザーズマニュアルと呼称します。

•RS-232C 接続の場合

PC 側に RS-232C ケーブルを差込み、RCX3 シリーズ側に反対側の RS-232C ケーブルを図 2-1 の赤枠箇所差し込みます。差し込む向きなどを間違えないように差し込んでください。詳細は「RCX3_ユーザーズマニュアル」の「第 6 章 外部通信インターフェース」を参照してください。

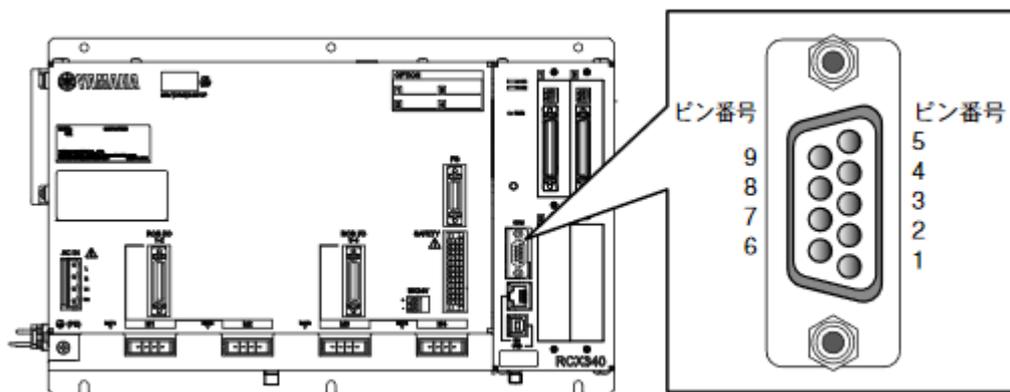


図 2-1 RCX3 シリーズの RS-232C 差込図

•Ethernet 接続の場合

LAN ケーブルは、カテゴリ 5 以上に対応した UTP ケーブル (シールド無しツイストペアケーブル) もしくは STP ケーブル (シールド付きツイストペアケーブル) を使用します。PC 側に LAN ケーブルを差込み、RCX3 シリーズ側に反対側の LAN ケーブルを直接又はハブを介して図 2-2 の赤枠箇所にカチッと音がするまで差し込んでください。差し込む向きなどを間違えないように差し込んでください。詳細は RCX3 ユーザーズマニ

マニュアルの「第 6 章 外部通信インターフェース」を参照してください。

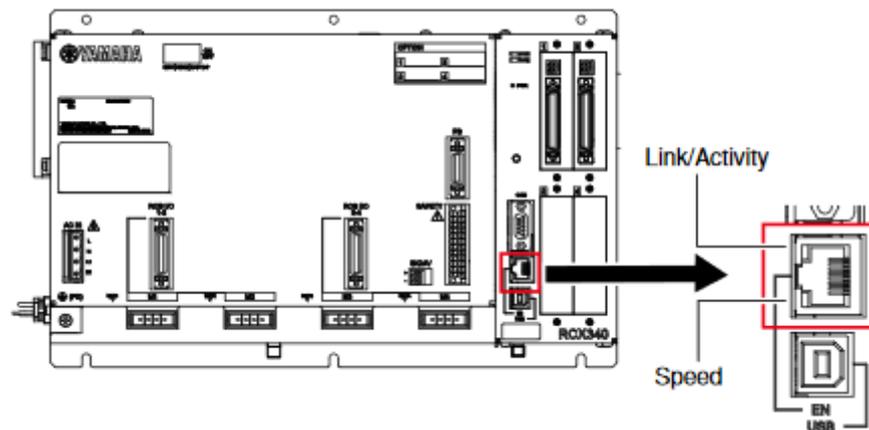


図 2-2 RCX3 シリーズの LAN ケーブル差込図

2.2.1. RCX3 シリーズの通信設定

RCX3 シリーズの通信設定を行う際は、ヤマハ発動機株式会社のプログラミングボックスまたは、ヤマハ発動機株式会社製の RCX3 シリーズのロボットコントローラの支援ソフトウェアである「RCX-Studio 2020」を使用することで設定可能です。プログラミングボックスでの通信設定方法は、RCX3 ユーザーズマニュアルの「第 6 章 外部通信インターフェース」を参照してください。ここでは RCX-Studio 2020 での通信設定方法を記述します。

1. RCX-Studio 2020 を起動します。
2. 起動後プロジェクトを作成または選択します。
3. 次にコントローラを新規作成又は選択し、オンラインモードにてコントローラのウィンドウを表示します。
4. ターゲットウィンドウの「[システム] - [通信設定]」を開きます。
5. 表示されたウィンドウに RS-232C 及びイーサネットの設定項目が表示されますので設定したい内容を設定し OK を押して下さい。
6. ターゲットウィンドウの「[ツール] - [転送 [PC → コントローラ]]」にて設定内容を RCX3 シリーズに反映させて完了です。

設定方法の詳細は、「RCX-Studio 2020」のヘルプを参照して下さい。

2.3. PC 開発環境のセットアップ

2.3.1. RCX3 プロバイダの手動インストール

RCX3 プロバイダを手動でインストールする場合は下記レジストリ登録を行う必要があります。レジストリ登録を行う場合は、管理者権限でコマンドプロンプトを起動し、regsvr32 コマンドを実行してください。実行する際には、ファイルのあるパスまで移動するか、ファイルパスを指定して実行してください。

表 2-1 RCX3 プロバイダ

ファイル名	CaoProvYAMAHARCX3.dll
ProgID	CaoProv.YAMAHA.RCX3
レジストリ登録	regsvr32 CaoProvYAMAHARCX3.dll
レジストリ登録の抹消	regsvr32 /u CaoProv YAMAHARCX3.dll

3. コマンドリファレンス

3.1. メソッド/プロパティ一覧

表 3-1 メソッド/プロパティ一覧

カテゴリ	メソッド/プロパティ ¹	機能	参照
CaoWorkspace			
	AddController	M コントローラに接続	P.11
CaoController			
	AddRobot	M ロボットオブジェクトの追加	P.14
CaoRobot			
	ID	P ロボット番号の取得/設定	P.15
	VariableNames	P 接続可能な変数名リストの取得	P.16
	Variables	P ロボットが保持する変数コレクションの取得	P.16
	Accelerate	M 加速度及び減速度を設定	P.16
	AddVariable	M 変数オブジェクトの追加	P.16
	Change	M ハンドの変更	P.17
	Execute	M 拡張コマンドの実行	P.21
	Halt	M ロボット動作を途中で停止	P.18
	Move	M ロボットが指定した位置へ移動	P.19
	Speed	M プログラム速度を変更	P.20
CaoVariable			
	Value	P 値の取得/設定	P.21

3.2. メソッド・プロパティ

3.2.1. CaoWorkspace クラス

3.2.1.1. AddController メソッド

RCX3 プロバイダでは AddController 時に通信用の接続パラメータを参照し、通信の接続を行います。このときオプションで通信形態、タイムアウト、Ethernet 通信の場合は Telnet のユーザ名、パスワードを指定します。

Ethernet 通信の場合、YAMAHA コントローラのエコーバック(ECHO)機能を使用すると、パフォーマンスが落ちるだけでなく、応答に対して誤認識する可能性がありますので、この機能を無効にして使用してください。

以下に、AddController メソッドの仕様を示します。

¹ M:メソッド, P:プロパティ, E:イベントをそれぞれ示します。

書式

```
AddController
(
    "<コントローラ名>",           // コントローラ名(任意)
    "CaoProv.YAMAHA.RCX3",       // 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
)
```

オプション

以下にオプション文字列に指定するオプションを示します。オプション文字列は下記に示す各オプションをカンマ(,)でつなげた文字列となります。

オプション	必須	説明	値範囲	デフォルト値
Conn=<接続パラメータ>	○	通信形態とその接続パラメータを設定します。詳細は 3.2.1.1.1 を参照してください。	--	--
User=<ユーザ名>	--	Ethernet 接続で RCX3 コントローラにログインするためのユーザ名を指定します。	--	admin
Password=<パスワード>	--	Ethernet 接続で RCX3 コントローラにログインするためのパスワードを指定します。	--	無し
Timeout=<タイムアウト時間>	--	送受信時のタイムアウト時間(ミリ秒)を指定します。	0 - 4294967295	500

使用例(VBA)

```
Dim caoEng As CaoEngine ' Engine オブジェクト
Dim caoWs As CaoWorkspace ' Workspace オブジェクト
Dim caoCtrl As CaoController ' Controller オブジェクト

' Engine オブジェクト
Set caoEng = New CaoEngine

' Workspace オブジェクト
Set caoWs = caoEng.AddWorkspace("Workspace", "")

' Controller オブジェクト
```

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

3.2.1.1.1. Conn オプション

以下に Conn オプションの接続パラメータ文字列を示します。ここで角括弧(“[]”)内のパラメータは省略可能を示します。また、各パラメータの解説中の下線部はオプション指定を省略した時のデフォルト値を示します。

•Ethernet 通信

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

<IP Address> : 必須。接続先の IP アドレス。

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

<PortNo> : 接続先のポート番号。

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

•RS232C 通信

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

<ComPort> : COM ポート番号。'1'-COM1, '2'-COM2,...

<BaudRate> : 通信速度。4800, 9600, 19200, 38400, 57600, 115200

<ByteSize> : パリティ。'N'-NONE, 'E'-EVEN, 'Q'-ODD

<DataBits> : データビット数。'7'-7bit, '8' - 8bit

<StopBits> : ストップビット数。'1'-1bit, '2'-2bit

<Flow> : フロー制御。'0'-None, '1'-Xon/Xoff, '2'-ハードウェア制御

3.2.1.1.2. 接続時の注意点

Ethernet 通信にてクライアント PC と RCX3 シリーズを接続する際にハブを介して接続する場合、接続中に LAN ケーブルなどの断線が発生した場合、RCX3 シリーズでは断線を検知できない為、再接続する場合は RCX3 シリーズを再起動させ、再度 CaoController オブジェクトを追加し直してください。

同一の IP アドレス:ポート番号へは、他アプリケーション及び本プロバイダ上で同時に接続することはできません。

3.2.2. CaoController クラス

3.2.2.1. AddRobot メソッド

CaoController に、ロボットオブジェクトを追加します。以下に、AddRobot メソッドの仕様を示します。

書式

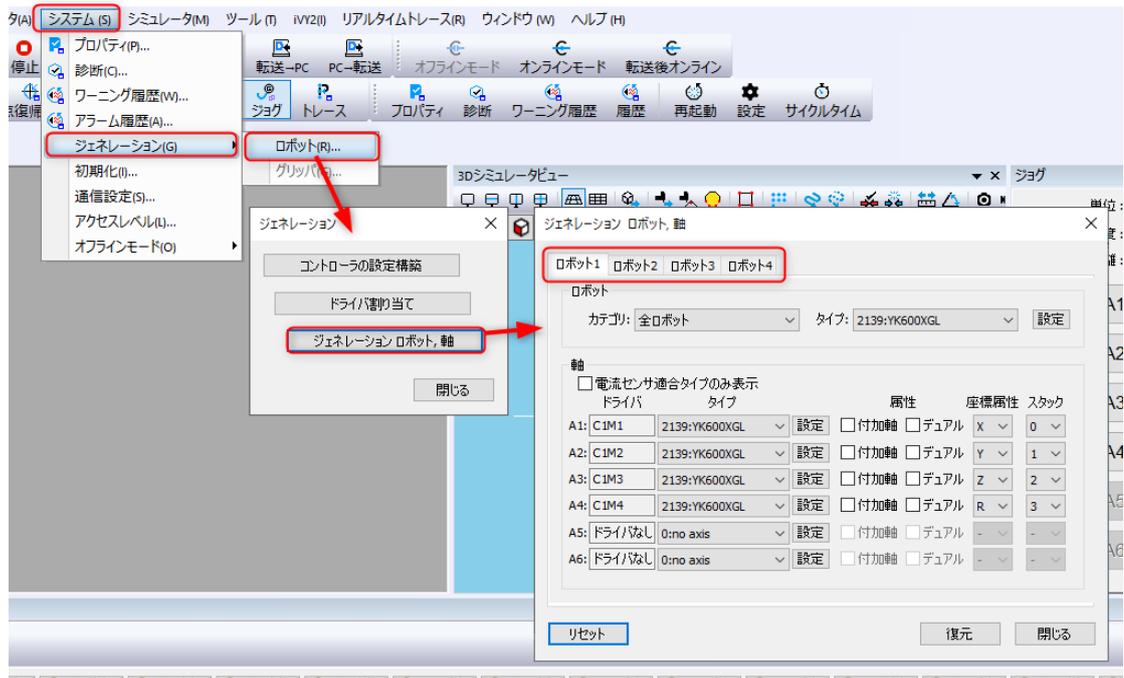
```
AddRobot
(
  "<ロボット名>",           // ロボット名(任意)
  "<オプション>"           // オプション文字列(省略可能)
)
```

オプション

以下にオプション文字列に指定するオプションを示します。

オプション	必須	説明	値範囲	デフォルト値
ID=<ロボット番号>	--	対象となるロボットの番号を設定します。この設定は CaoRobot::ID プロパティを使用して変更することができます。	1 - 4	1

※ コントローラに接続されているロボットのロボット番号を確認するには、「RCX-Studio 2020」のターゲットウィンドウの[システム]-[ジェネレーション]-[ロボット]からジェネレーションウィンドウを表示し、[ジェネレーション ロボット,軸]ウィンドウを表示させ、[ロボット 1]タブに表示されているロボットのロボット番号が「1」、[ロボット 2]タブに表示されているロボットのロボット番号が「2」、[ロボット 3]タブに表示されているロボットのロボット番号が「3」、[ロボット 4]タブに表示されているロボットのロボット番号が「4」となります。



使用例(VBA)

Dim caoRbt As CaoRobot ' Robot オブジェクト

' CaoRobot オブジェクトの生成

Set caoRbt = caoCtrl.AddRobot("MF20","ID=1")

3.2.3. CaoRobot クラス

3.2.3.1. ID プロパティ

ロボットオブジェクトの対象のロボット番号を取得/設定します。

データ型

型説明	
VT_I4	ロボット番号

使用例(VBA)

' ID を取得

Dim id As Integer

id = caoRbt.id

' ID を設定

id = 2

caoRbt.id = id

3.2.3.2. VariableNames プロパティ

接続可能な変数名リストを取得します。本プロパティで取得した変数名は、後述する AddVariable メソッドの第一引数に使用することができます。

使用例(VBA)

```
' 変数名一覧取得
Dim variableNames() As String
variableNames = caoRbt.variableNames
```

3.2.3.3. Variables プロパティ

ロボットが保持する、変数コレクションを取得します。

使用例(VBA)

```
' 変数コレクション取得
Dim variables As CaoVariables
Set variables = caoRbt.Variables

' 変数取得
Dim variable As CaoVariable
Set variable = variables.Item(0)
```

3.2.3.4. AddVariable メソッド

CaoRobot に変数オブジェクトを追加します。変数名には 3.4.1 に示すもののみ使用できます。

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(省略可能)
)
```

3.2.3.5. Accelerate メソッド

加速度及び減速度を設定します。以下に、Accelerate の仕様を示します。

書式

```
Accelerate
(
```

```

"<軸番号>",           // 軸番号(範囲 : 0 ~ 6)
"<加速度>",          // 加速度(範囲 : 0 ~ 100)
"<減速度>"           // 減速度(範囲 : 0 ~ 100)
)

```

軸番号

指定した軸の加速度及び減速度を設定します。軸番号に0を指定した場合は全ての軸に対して加速度及び減速度を設定します。

加速度

加速度を設定します。加速度に 0 を指定した場合は、加速度の設定を行いません。このため減速度のみを設定したい場合は、この項目を 0 に設定してください。

減速度

減速度を設定します。減速度に 0 を指定した場合は、減速度の設定を行いません。このため加速度のみを設定したい場合は、この項目を 0 に設定してください。

使用例(VBA)

```
' 軸 1 の加速度を 60, 減速度を 50 に設定
```

```
caoRbt.Accelerate 1, 60, 50
```

```
' 軸 2 の加速度のみを 30 に設定
```

```
caoRbt.Accelerate 2, 30, 0
```

```
' 軸すべての減速度のみを 40 に設定
```

```
caoRbt.Accelerate 0, 0, 40
```

3.2.3.6. Change メソッド

ロボットのハンドデータを切り替えます。以下に、Change の仕様を示します。

書式

```

Change
(
    "<ハンド番号>"      // ハンド番号(範囲 : 0 ~ 31 or "OFF")
)

```

ハンド番号

指定したハンド番号のハンドデータへ切り替えます。"OFF"を指定した場合、ハンドデータ設定なしとなります。

使用例(VBA)

```
' ハンド 1 に設定
```

```
caoRbt.Change 1
```

```
' ハンドデータ設定なしを設定
```

```
caoRbt.Change "OFF"
```

3.2.3.7. Halt メソッド

ロボット動作を途中で停止させます。以下に, Halt の仕様を示します。

書式

```
Halt
```

```
(
```

```
    "＜オプション文字列＞"
```

```
    // オプション文字列(未使用)
```

```
)
```

使用例(VBA)

```
' ロボットを停止させる
```

```
caoRbt.Halt
```

3.2.3.8. Move メソッド

ロボットを指定した位置へ移動させます。第1引数に Move コマンドの種類、第2引数に VARIANT 配列で移動先情報、速度を指定します。以下に Move の仕様を示します。

書式

```
Move
(
  "<補間指定>",           // 補間指定(範囲 : 1 ~ 7)
  "<移動先指定>",       // 移動先指定
  "<オプション>"        // オプション(省略可能)
)
```

補間指定

このパラメータの内容は補間指定を指定します。補間指定の詳細については「RCX3 シリーズ プログラミング説明書」を参照してください。

以下に指定可能な補間指定の詳細を示します。

指定値	コマンド名	移動種別	説明
1	MOVE	P	現在ロボットが停止している位置から指定ポイントの位置まで各軸が最短で移動できる経路を通してロボットを移動させます。
2	MOVE	L	現在ロボットが停止している位置と指定ポイント位置による 2 点を直線で結んだ経路でロボットを移動させます。
3	MOVE	C	現在ロボットが停止している位置と中間の指定ポイント位置、目標の指定ポイント位置による 3 点を円弧で結んだ経路でロボットを移動させます。そのため、ポイント指定は必ず偶数個でなければなりません。
4	MOVE I	P	現在ロボットが停止している位置から指定ポイント座標分プラスした位置による各軸が最短で移動できる経路を通してロボットを移動させます。
5	MOVE I	L	現在ロボットが停止している位置から指定ポイント座標分プラスした位置による 2 点を直線で結んだ経路でロボットを移動させます。
6	MOVET	P	現在ロボットが停止している位置からツール座標でプラスされた位置による各軸が最短で移動できる経路を通してロボットを移動させます。

指定値	コマンド名	移動種別	説明
7	MOVET	L	現在ロボットが停止している位置からツール座標でプラスされた位置による2点を直線で結んだ経路でロボットを移動させます。

※ P : Point to Point, L : 直線補間, C : 円弧補間

移動先指定

移動先を文字列で指定します。直接数値で指定する際は各軸に設定する数値間の間にスペースを入力してください。詳しい指定方法については「RCX3 シリーズ プログラミング説明書」を参照してください。

※ 直接数値にて指定する際、対象のロボットに存在しない軸には0を指定してください。

オプション

各オプション間はカンマ区切りで指定してください。オプションは省略可能です。オプションの詳細については、6.2.オプションの種類を参照してください。

使用例(VBA)

・ポイントデータを使用して Move メソッド実行

' 軸2の加速度のみを30に設定

caoRbt.Move 1, "P13", "S=50,CONT"

' 1=PTP,相対速度を指定(50%),CONT 指定

caoRbt.Move 2, "P11"

' 2=直線補間(現在位置から P11 へ移動)

caoRbt.Move 3, "P10,P11,P12,P13"

' 3=円弧補間

caoRbt.Move 4, "P11", "S=50"

' 現在位置から P11=170mm/deg 移動

・直接数値を使用して Move メソッド実行(Move させるロボットは軸4まで存在)

caoRbt.Move 1, "152.997 301.000 70.000 75.000 0 0", "" ' 1=PTP,オプション省略

3.2.3.9. Speed メソッド

プログラム速度を変更します。全ての軸のスピードが変更されます。以下に、Speed の仕様を示します。

書式

Speed

(

"<軸番号>", // 軸番号(未使用)

"<速度>" // 速度(範囲 : 1 ~ 100)

)

速度

プログラム速度を指定します。ロボットの移動速度は自動移動速度と本メソッドで指定されたプログラム移動速度の積によって決まります。

移動速度 = 自動移動速度 × プログラム移動速度

例) 自動移動速度(80%) × プログラム移動速度(50%) = 移動速度(40%)

使用例(VBA)

```
' プログラム速度を 50%に設定
```

```
caoRbt.Speed 1,50
```

3.2.3.10. Execute メソッド

CaoRobot の拡張コマンドを実行します。Execute で指定できる拡張コマンドについては 3.3.1.CaoRobot クラス拡張コマンドに示すもののみ使用可能です。以下に、Execute の仕様を示します。

書式

```
Execute
```

```
(
```

```
    "<拡張コマンド名>",           // 拡張コマンド名
```

```
    "<オプション文字列>"        // オプション文字列(省略可能)
```

```
)
```

3.2.4. CaoVariable クラス**3.2.4.1. Value プロパティ**

接続した RCX3 シリーズコントローラからデータを取得/設定します。変数名によって動作が異なります。詳細は、3.4.変数一覧を参照してください。

3.3. 拡張コマンド一覧

各クラスで使用可能なコマンド一覧を定義します。使用例は各コマンドの詳細で記述しています。

3.3.1. CaoRobot クラス拡張コマンド

本章のロボットを動作させる拡張コマンドはロボットの動作が完了するかもしくはタイムアウトが発生するまで、本プロバイダは待機します。

表 3-2 CaoRobot クラス拡張コマンド一覧

コマンド	説明	参照
PMOVE	パレット移動命令	P.22
SERVO	サーボ状態の設定	P.22

3.3.1.1. PMOVE コマンド

パレット移動命令を実行します。以下に引数を示します。

項目	型説明	
引数	VT_ARRAY VT_VARIANT	
	0	VT_I4 パレット番号(0~39)
	1	VT_I4 ポイント番号(1~32767)
	2	VT_BSTR オプション文字列。 各オプション間はカンマ区切りで指定してください。オプションは省略可能です。オプションの詳細については、6.2.オプションの種類を参照してください。

使用例(VBA)

' パレット番号 1, ポイント番号 5, プログラム速度を 50%で指定しパレット移動

```
Call caoRbt.Execute("PMove", Array(1, 5, "S=50"))
```

3.3.1.2. SERVO コマンド

サーボ状態を設定します。ロボットの動作させる際はサーボ状態 ON に設定しておく必要があります。以下に引数を示します。

項目	型説明	
引数	VT_ARRAY VT_VARIANT	移動位置指定

項目	型説明	
0	VT_BSTR	サーボ状態. 以下のいずれかを指定してください. ON OFF FREE
1	VT_I4	軸番号(1~6). 省略または 0 を指定した場合は, 全軸指定となります.

使用例(VBA)

' 軸 1 のサーボ状態を ON

```
Call caoRbt.Execute("SERVO", array("ON", 1))
```

' 全軸のサーボ状態を OFF

```
Call caoRbt.Execute("SERVO", array("OFF"))
```

3.4. 変数一覧

各クラスで使用可能な変数一覧を定義します. なお変数は, `CaoVariable` クラスのオブジェクトを指します.

3.4.1. CaoRobot クラス変数

変数名	説明	Value		参照
		get	put	
@WHERE	ロボットのパルス座標系現在位置を取得します.	○	-	P.23
@WHRXY	ロボットのミリ単位系現在位置を取得します.	○	-	P.24

3.4.1.1. @WHERE

ロボットのアーム現在位置(パルス座標)を取得します.

データ型

型説明	
VT_I4 VT_ARRAY	ロボットのパルス座標系現在位置
0	第 1 軸の現在位置
1	第 2 軸の現在位置
2	第 3 軸の現在位置
3	第 4 軸の現在位置

4	第 5 軸の現在位置
5	第 6 軸の現在位置

使用例(VBA)

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

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

' 各軸の値取得

```
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

ロボットのアーム現在位置(直交座標)を取得します。

データ型

型説明	
VT_R4 VT_ARRAY	ロボットの直交座標系現在位置
0	第 1 軸の現在位置
1	第 2 軸の現在位置
2	第 3 軸の現在位置
3	第 4 軸の現在位置
4	第 5 軸の現在位置
5	第 6 軸の現在位置

使用例(VBA)

```
Dim whrxyVariable As CaoVariable
Dim whrxyValues() As Single
```

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

```
' 各軸の値取得
```

```
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. RCX3 プロバイダによるプログラミング

RCX3 プロバイダでは、以下の手順で機器との通信の準備を行います。

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

手順を実行した後は、CaoController の Execute メソッドを使用する、もしくは、CaoVariable オブジェクトの Value プロパティの取得を行うことで、RCX3 シリーズとの通信を行い機器の情報にアクセスすることができます。

また、サンプルプログラムではプログラム言語「VBA(Visual Basic for Applications)」を使用してプログラムしています。

4.1. ロボットのアーム現在位置(パルス値)を取得するサンプルプログラミング

ここでは例としてロボットのアーム現在位置を読み込むサンプルプログラムを示します。表 4-1 にサンプルプログラムの要件を、図 4-1 にサンプルプログラムの流れをそれぞれ記述しています。

表 4-1 サンプルプログラムの要件

要件	説明
処理内容	RCX340 からロボットのアーム現在位置を読み込む。

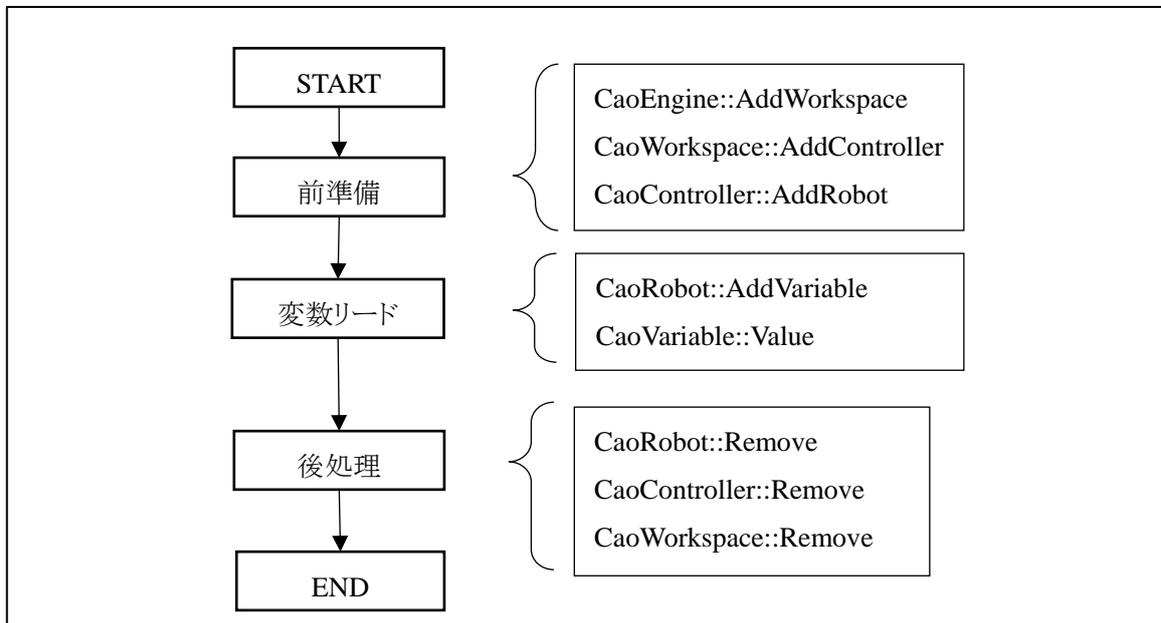


図 4-1 ロボットのアーム現在位置読み込みの流れ

以降の節から具体的なコードを示します。

4.1.1. サンプルプログラム

以下にサンプルプログラムの全体像を示します。

Sample	GetCurrentData.vb
--------	-------------------

' オブジェクト

```

Dim caoEng As CaoEngine
Dim caoWs As CaoWorkspace
Dim caoCtrl As CaoController
Dim caoRbt As CaoRobot
Dim whereVariable As CaoVariable
  
```

Private Sub Main()

' 前処理

Call Connect

Dim whereValues() As Integer

whereValues = whereVariable.Value

```
' 各軸の値取得
```

```
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
```

```
' 後処理
```

```
Call Disconnect  
End Sub
```

```
' 前処理メソッド
```

```
Private Sub Connect()  
    ' CaoEngine オブジェクトの生成  
    Set caoEng = New CaoEngine  
    ' CaoWorkspace オブジェクトの生成  
    Set caoWs = caoEng.AddWorkspace("Workspace", "")  
    ' CaoController オブジェクトの生成  
    Set caoCtrl = caoWs.AddController("RCX3", _  
        "CaoProv. YAMAHA. RCX3", _  
        "", _  
        " Conn=eth:192.168.0.10, Timeout=1000")  
    ' CaoRobot オブジェクトの生成  
    Set caoRbt = caoCtrl.AddRobot("MF20")  
    ' CaoVariable オブジェクトの生成  
    Set whereVariable = caoRbt.AddVariable("@WHERE")  
End Sub
```

```
' 後処理メソッド
```

```
Private Sub Disconnect()  
    ' CaoController から CaoVariable を削除
```

```
Call caoCtrl.variables.Remove(whereVariable.Index)
' CaoVariable の消去
Set whereVariable = Nothing
' CaoWorkspace から CaoController を削除
Call caoWs.Controllers.Remove(caoCtrl.Index)
' CaoController の消去
Set caoCtrl = Nothing
' CaoEngine から CaoWorkspace を削除
Call caoEng.Workspaces.Remove(caoWs.Index)
' CaoWorkspace の消去
Set caoWs = Nothing
' CaoEngine の消去
Set caoEng = Nothing
End Sub
```

4.1.1.1. 前処理

機器と通信するためには、以下の手順を取ります。

- (1) オブジェクトを保持するための変数を用意します。コントローラ接続に必要なオブジェクトは、CaoEngine オブジェクトと CaoWorkspace オブジェクトと CaoController オブジェクトです。CaoWorkspace オブジェクトは、CaoController オブジェクトを CaoWorkspaces から取得する場合には変数を用意する必要はありません。また対象のロボットを指定するために CaoRobot オブジェクトと、変数にアクセスするための CaoVariable オブジェクトも必要になります。以下に VBA でのコード例を示します。

```
' CaoEngine オブジェクト用の変数
Dim caoEng As CaoEngine
' CaoWorkspace オブジェクト用の変数
Dim caoWs As CaoWorkspace
' CaoController オブジェクト用の変数
Dim caoCtrl As CaoController
' CaoRobt オブジェクト用の変数
Dim caoRbt As CaoRobot
' CaoVariable オブジェクト用の変数
```

Dim whereVariable As CaoVariable

- (2) CaoEngine オブジェクトを生成します。CaoEngine オブジェクトは New キーワードを使って生成します。

' CaoEngine オブジェクトの生成

```
Set caoEng = New CaoEngine
```

- (3) CaoWorkspace オブジェクトを取得もしくは生成します。CaoEngine オブジェクトを生成すると、デフォルトで CaoWorkspaces オブジェクトと CaoWorkspace オブジェクトを 1 つずつ生成しています。以下に CaoWorkspace オブジェクトを新しく生成するコード例とデフォルトの CaoWorkspace を示します。

' CaoWorkspace オブジェクトの生成

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

- (4) CaoController オブジェクトを生成します。CaoController オブジェクトを生成するには、使用するプロバイダ名と使用するためのパラメータを設定します。RCX3 プロバイダでは、接続先情報とタイムアウトをオプションで指定します。以下にコード例を示します。

' CaoController オブジェクトの生成

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

- (5) CaoRobot オブジェクトを生成します。CaoRobot オブジェクトを生成するには、ロボット名とパラメータを設定します。RCX3 プロバイダでは、ロボット番号をオプションで指定します。以下にコード例を示します。

' CaoRobot オブジェクトの生成

```
Set caoRbt = caoCtrl.AddRobot("MF20")
```

- (6) CaoVariable オブジェクトを生成します。取得したい変数の CaoVariable オブジェクトを生成します。以下にロボットのアーム現在位置にアクセスする変数オブジェクトを生成するコー

ド例を示します。

```
' CaoVariable オブジェクトの生成
```

```
Set whereVariable = caoRbt.AddVariable("@WHERE")
```

4.1.1.2. ロボットのアーム現在位置取得

ロボットのアーム現在位置を取得するには、CaoVariable オブジェクトの Value プロパティを参照します。ここでは各軸のデータを参照します。以下にコード例を示します。

```
Dim whereValues() As Integer
whereValues = whereVariable.Value
' 各軸の値取得
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. 後処理

後処理を行う場合には、生成したオブジェクトを消去すると共に、オブジェクトを管理するコレクションクラスから消去するオブジェクトを削除します。以下にコード例を示します。

```
' CaoRobot から CaoVariable を削除
Call caoRbt.Variables.Remove(whereVariable.Index)
' CaoVariable の消去
Set whereVariable = Nothing
' CaoController から CaoRobot を削除
Call caoCtrl.Robots.Remove(caoRbt.Index)
' CaoController の消去
Set caoRbt = Nothing
' CaoWorkspace から CaoController を削除
Call caoWs.Controllers.Remove(caoCtrl.Index)
' CaoController の消去
Set caoCtrl = Nothing
' CaoEngine から CaoWorkspace を削除
Call caoEng.Workspaces.Remove(caoWs.Index)
' CaoWorkspace の消去
```

Set caoWs = Nothing

'CaoEngine の消去

Set caoEng = Nothing

4.2. ポイントデータを使用しロボットを移動させるサンプルプログラミング

ここでは例として RCX3 シリーズのロボットを移動させるサンプルプログラムを示します。表 4-2 にサンプルプログラムの要件を、図 4-2 にサンプルプログラムの流れをそれぞれ記述しています。

表 4-2 サンプルプログラムの要件

要件	説明
処理内容	RCX340 のロボットを移動させる。

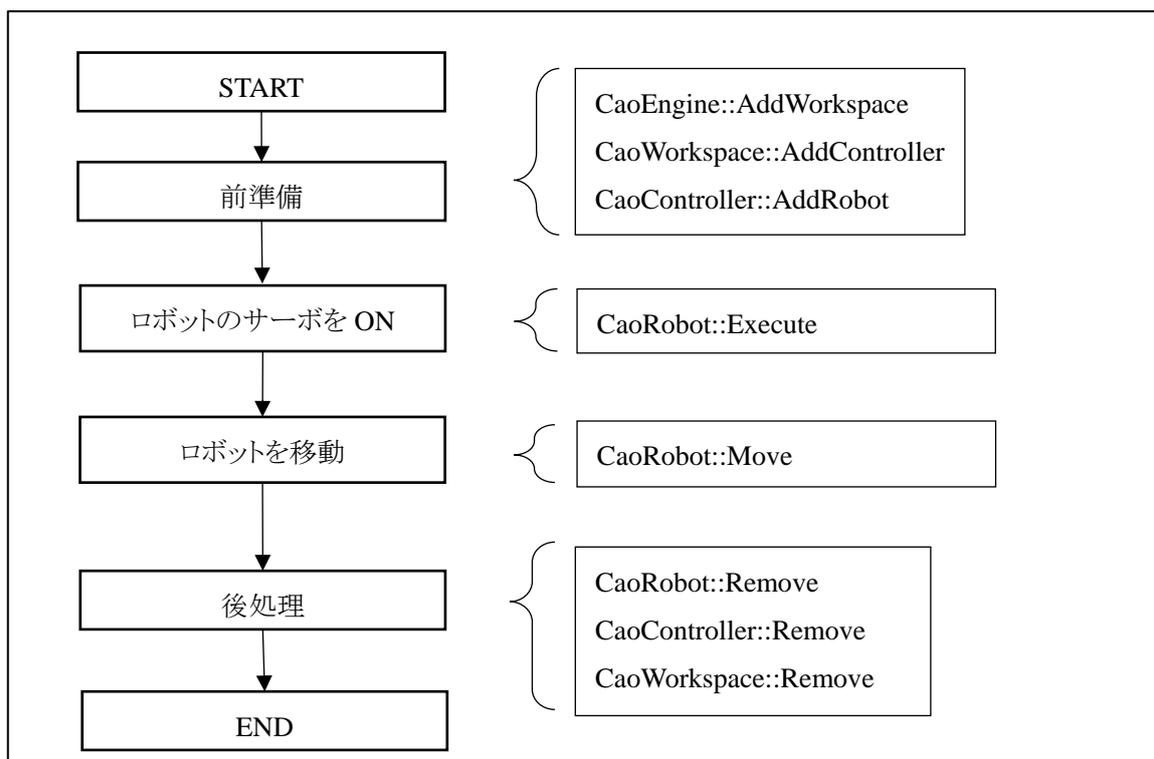


図 4-2 ロボットを移動させる流れ

以降の節から具体的なコードを示します。

4.2.1. サンプルプログラム

以下にサンプルプログラムの全体像を示します。

Sample

GetCurrentData.vb

' オブジェクト

Dim caoEng As CaoEngine

Dim caoWs As CaoWorkspace

Dim caoCtrl As CaoController

Dim caoRbt As CaoRobot

Private Sub Main()

' 準備

Call Connect

' ロボットの全軸のサーボ状態を ON にする

Call caoRbt.Execute("SERVO", Array("ON"))

' 数値を使用しロボットを移動させる

caoRbt.Move 1, "152.997 301.000 70.000 75.000 0 0", "S=80"

' 後処理

Call Disconnect

End Sub

' 準備メソッド

Private Sub Connect()

' CaoEngine オブジェクトの生成

Set caoEng = New CaoEngine

' CaoWorkspace オブジェクトの生成

Set caoWs = caoEng.AddWorkspace("Workspace", "")

' CaoController オブジェクトの生成

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

' CaoRobot オブジェクトの生成

Set caoRbt = caoCtrl.AddRobot("MF20")

End Sub

' 後処理メソッド

Private Sub Disconnect()

' CaoController から CaoRobot を削除

Call caoCtrl.Robots.Remove(caoRbt.Index)

' CaoRobot の消去

```
Set caoRbt = Nothing
' CaoWorkspace から CaoController を削除
Call caoWs.Controllers.Remove(caoCtrl.Index)
' CaoController の消去
Set caoCtrl = Nothing
' CaoEngine から CaoWorkspace を削除
Call caoEng.Workspaces.Remove(caoWs.Index)
' CaoWorkspace の消去
Set caoWs = Nothing
' CaoEngine の消去
Set caoEng = Nothing
End Sub
```

4.2.1.1. 前処理

機器と通信するためには、以下の手順を取ります。

- (1) オブジェクトを保持するための変数を用意します。コントローラ接続に必要なオブジェクトは、CaoEngine オブジェクトと CaoWorkspace オブジェクトと CaoController オブジェクトです。CaoWorkspace オブジェクトは、CaoController オブジェクトを CaoWorkspaces から取得する場合には変数を用意する必要はありません。また対象のロボットを指定するために CaoRobot オブジェクトが必要になります。以下に VBA でのコード例を示します。

```
' CaoEngine オブジェクト用の変数
Dim caoEng As CaoEngine
' CaoWorkspace オブジェクト用の変数
Dim caoWs As CaoWorkspace

' CaoController オブジェクト用の変数

Dim caoCtrl As CaoController

' CaoRobt オブジェクト用の変数

Dim caoRbt As CaoRobot
```

- (2) CaoEngine オブジェクトを生成します。CaoEngine オブジェクトは New キーワードを使って生成します。

'CaoEngine オブジェクトの生成

```
Set caoEng = New CaoEngine
```

- (3) CaoWorkspace オブジェクトを取得もしくは生成します。CaoEngine オブジェクトを生成すると、デフォルトで CaoWorkspaces オブジェクトと CaoWorkspace オブジェクトを1つずつ生成しています。以下に CaoWorkspace オブジェクトを新しく生成するコード例とデフォルトの CaoWorkspace を示します。

'CaoWorkspace オブジェクトの生成

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

- (4) CaoController オブジェクトを生成します。CaoController オブジェクトを生成するには、使用するプロバイダ名と使用するためのパラメータを設定します。RCX3 プロバイダでは、接続先情報とタイムアウトをオプションで指定します。以下にコード例を示します。

'CaoController オブジェクトの生成

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

- (5) CaoRobot オブジェクトを生成します。CaoRobot オブジェクトを生成するには、ロボット名とパラメータを設定します。RCX3 プロバイダでは、ロボット番号をオプションで指定します。以下にコード例を示します。

'CaoRobot オブジェクトの生成

```
Set caoRbt = caoCtrl.AddRobot("MF20")
```

4.2.1.2. ロボットのサーボ状態を ON

ロボットのサーボ状態を ON にするには、CaoRobot オブジェクトの Execute メソッドの「SERVO」コマンドを実行します。ここでは全軸を ON にします。以下にコード例を示します。

'ロボットの全軸のサーボ状態を ON にする

```
Call caoRbt.Execute("SERVO", Array("ON"))
```

4.2.1.3. ロボットを移動

ロボットを移動させる数値を使用して、ロボットを移動させます。ロボットを移動させるには、CaoRobot オブジェクトの Move メソッドを実行します。直接数値を使用し、Point to Point 移動を実行します。以下にコード例を示します。

```
' 定義したポイントデータを使用しロボットを移動させる
caoRbt.Move 1, "152.997 301.000 70.000 75.000 0 0", "S=80"
```

4.2.1.4. 後処理

後処理を行う場合には、生成したオブジェクトを消去すると共に、オブジェクトを管理するコレクションクラスから消去するオブジェクトを削除します。以下にコード例を示します。

```
' CaoController から CaoRobot を削除
Call caoCtrl.Robots.Remove(caoRbt.Index)
' CaoRobot の消去
Set caoRbt = Nothing
' CaoWorkspace から CaoController を削除
Call caoWs.Controllers.Remove(caoCtrl.Index)
' CaoController の消去
Set caoCtrl = Nothing
' CaoEngine から CaoWorkspace を削除
Call caoEng.Workspaces.Remove(caoWs.Index)
' CaoWorkspace の消去
Set caoWs = Nothing
' CaoEngine の消去
Set caoEng = Nothing
```

5. RCX3 プロバイダエラーコード

RCX3 プロバイダには、独自のエラーコードは存在せず、デバイスからのエラーコードを 0x801*****でマスクした以下のプロバイダエラーコードが存在します(表 5-1 プロバイダエラーコード表参照).

ORiN2 の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参照してください.

表 5-1 プロバイダエラーコード表

エラー番号	説明
0x801xxyyy	RCX3 のコマンド実行時にエラーが発生した場合は, xx の箇所にアラームコードのグループ番号, yyy の箇所にアラームコードの分類番号を 16 進数で返します. アラームコードの詳細な内容については RCX3 のユーザーズマニュアルのトラブルシューティングを参照してください.

6. 付録

6.1. RCX3 コマンド対応表

6.1.1. CaoRobot クラス

表 6-1 CaoRobot::Execute メソッド-RCX3 コマンド対応表

コマンド名	RCX3 コマンド
PMOVE	PMOVE[i]
SERVO	SERVO[i]

※ i = ロボット番号

表 6-2 CaoRobot::Execute 以外のメソッド-RCX3 コマンド対応表

メソッド	RCX3 コマンド
Accelerate	ACCEL[i]
	DECEL[i]
Change	CHANGE[i]
Halt	^C
Move	MOVE[i]
	MOVEI[i]
Speed	SPEED[i]

※ i = ロボット番号

表 6-3 CaoRobot の変数オブジェクト-RCX3 コマンド対応表

変数名	RCX3 コマンド
@WHERE	?WHERE[i]
@WHRXY	?WHRXY[i]

※ i = ロボット番号

6.2. オプションの種類

ロボットを移動させるコマンド実行時に指定可能なオプションがいくつか存在します。各コマンドに指定可能なオプションは表 6-4 を参照してください。

表 6-4 指定可能なオプション対応表

オプション	オプション文字列	MOVE P	MOVE L	MOVE C	MOVEI P	PMOVE	参照
相対速度指定 (SPEED)	"S="	○	○	○	○	○	P.40
絶対速度指定 (DSPEED)	"DS="	○	○	○	○	×	P.41
直線速度指定 (VEL)	"VEL="	×	○	○	×	×	P.41
アーチモーショ ン	"Ax=" (x:軸番号)	○	×	×	○	○	P.41
CONT 指定	"CONT"	○	○	○	○	○	P.42
加速度指定	"ACC="	○	○	○	○	×	P.42
減速度指定	"DEC="	○	○	○	×	×	P.43
座標平面指定	"XY" or "YZ" or "ZX"	×	×	○	×	×	P.43
ポート出力指定	"DOm(b,···,b)=" or "MOm(b,···,b)=" or "SOm(b,···,b)=" (m:ポート番号 b:ビット指定)	×	○	○	×	×	P.43

以下に指定可能なオプションの指定方法、詳細を記述します。

6.2.1. 相対速度指定(SPEED)

プログラム移動速度(S)を比率で指定します。この速度設定は、現在設定されている自動移動速度に対してさらに設定速度を変更します。

ロボットの動作速度 =

ロボットの最高速度(mm/秒または度/秒) × 自動移動速度(%) × プログラム移動速度 S(%)

動すると、その他の軸が目標位置に移動します。その他の軸が移動を終了する時点で、残り移動距離が残りのアーチ距離 2 になるように目標位置に移動します。すべての軸がアウト有効位置範囲以内に入ると命令を終了します。アーチモーションの詳細については、「RCX3 シリーズ プログラミング説明書」を参照してください。

オプション形式

"x=式{式 1,式 2}" {式 1,式 2}は省略可能

- x: アーチ軸(指定軸 : A1 ~ A6)
- 式 : アーチ位置 (直接数値指定 or 変数指定)
- 式 1: 始点に対する直線距離 (アーチ距離 1)
- 式 2: 終点に対する直線距離 (アーチ距離 2)

使用例(VBA)

'現在位置から A3 軸が 0.00mm に移動. その後, その他の軸が移動し最後に A3 軸が P1 に移動
caoRob.Move 1, "P1", "A3=0.00{50.00,70.00}"

6.2.5. CONT 指定(連結動作)

CONT 指定の移動を実行した場合、移動対象軸の動作完了(公差範囲内に到着)を待たずに次の命令を開始します。次の命令が移動命令の場合、2 つの移動経路が減速と加速部分を接続することで連結され、途中で停止することなく移動が継続されます。CONT 指定の詳細については、「RCX3 シリーズ プログラミング説明書」を参照してください。

オプション形式

"CONT"

使用例(VBA)

caoRob.Move 1, "P1", "CONT" 'CONT 指定

6.2.6. 加速度指定

加速度係数パラメータによって決定されたロボットの加速度に対する比率をオプション形式の式で指定し、動作時の加速度を決定します。

動作時の加速度 = 加速度係数パラメータ × 加速度 ACC(%)

オプション形式

"ACC=式" 式には 1~100 を指定(単位:%)

使用例(VBA)

 caoRob.Move 1, "P1", "ACC=10"

'加速度比率 10%で移動'

6.2.7. 減速度指定

加速度係数パラメータによって決定されたロボットの加速度に対する比率をオプション形式の式で指定し、動作時の減速度を決定します。

動作時の減速度 = 加速度係数パラメータ × 減速度 DEC(%)

オプション形式

"DEC=式"

式には 1~100 を指定(単位:%)

使用例(VBA)

 caoRob.Move 1, "P1", "DEC=20"

'減速度比率 20%で移動'

6.2.8. 座標平面指定

座標指定の円弧補間を実行した場合、指定座標平面への投射が円となるような円弧補間を実行します。座標平面指定の詳細については、「RCX3 シリーズ プログラミング説明書」を参照してください。

オプション形式

"XY"

XY 座標平面

or

"YZ"

YZ 座標平面

or

"ZX"

ZX 座標平面

使用例(VBA)

'P10 から P14 で生成される XY 平面状の円弧を移動'

caoRob.Move 3, "P10,P11,P13,P14", "XY"

6.2.9. ポート出力指定

直線補間または円弧補間を実行時、移動開始位置からオプション形式の式 2 の距離(単位:mm)だけ離れるタイミングで、指定した出力ポートに式 1 の値を出力します。(ハード的に存在しないポート番号を指定した場合、何も出力されません。)

このオプションはオプション内に 2 回まで指定可能です。

ポート出力指定の詳細については、「RCX3 シリーズ プログラミング説明書」を参照してください。

オプション形式

"DOm(b,···,b)=式 1@式 2"

b,···,b を省略した場合, 8bit 全てが対象

or

"MOm(b,···,b)=式 1@式 2"

or

"SOm(b,···,b)=式 1@式 2"

m : ポート番号(2~7,10~17,20~27)

b : ビット指定(0~7)

式 1 : 指定ポートに出力する値 (整数のみ有効)

式 2 : ポート出力を行う位置(小数点以下 3 桁まで指定可能)※移動開始点を中心とした円の半径

使用例(VBA)

'ロボットが P1 へ直線補間で移動中, 移動前の位置から 28.85mm 離れるタイミングで DO2()に 105(&B01101001)を出力

```
caoRob.Move 2, "P1", "DO2()=105@25.85"
```