

DENSO

DENSO Robotics

THIRD PARTY PRODUCTS

【サードパーティプロダクト】



PROVIDER MANUAL

プロバイダ取扱説明書

メーカー

(株) キーエンス 製

製品/シリーズ

画像センサ

MODEL:CV-X シリーズ



Vision

はじめに

本書は、「(株)キーエンス製・画像センサ・CV-Xシリーズ」をデンソーロボットコントローラRC8シリーズと接続して使用するためのプロバイダの取扱説明書です。

ご注意：(1) 取扱説明書に記載された内容以外でご使用された場合、機能・性能の保証はできませんのでご注意ください。

(2) 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。

本書が対象としているのはCV-Xシリーズの内、下記の製品です(2021年2月時点)。

(株)キーエンス製 CV-X300 シリーズ、400 シリーズ
(CV-X300/320/350/400/420/450/470)

以下、上記製品をまとめてCV-Xシリーズと明記します。

お願い

ご使用前に「マニュアル・安全上のご注意」をお読みいただき正しく安全にプロバイダをご使用下さい。

お客さまへ

1. ご使用に係わるリスクについて

本製品(ソフトウェア)のシステム組み込み・使用ならびに本製品の使用結果に関する一切のリスクは、本製品の使用者に帰属するものとします。

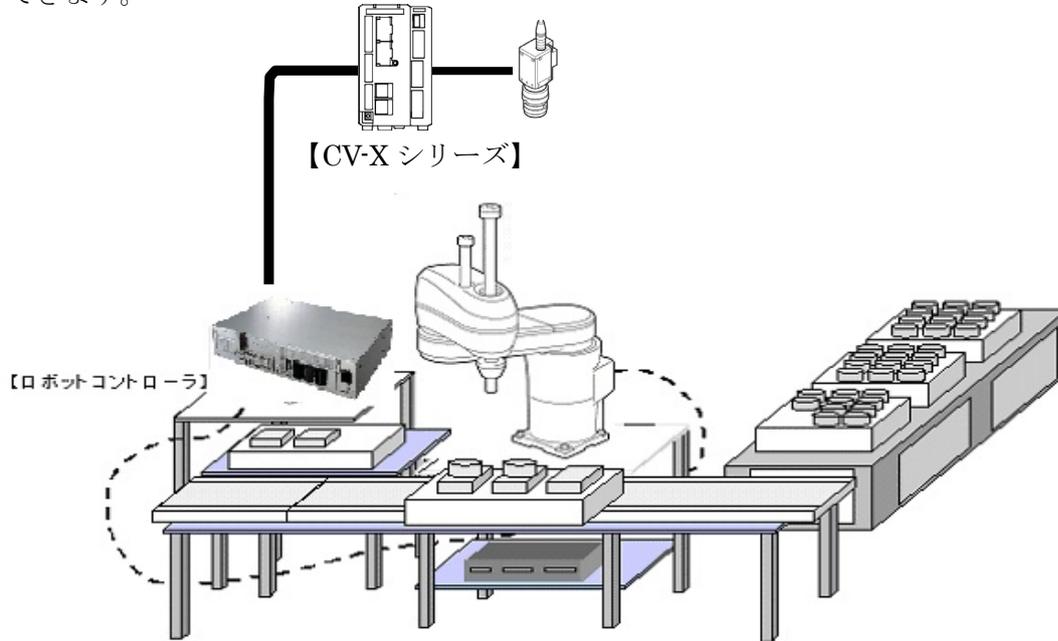
目次

はじめに	2
お願い	2
お客さまへ	2
1. 本製品（プロバイダ）の概要	4
1.1. プロバイダの対象機器	4
1.2. プロバイダの特長	4
1.3. プロバイダの仕組み	5
2. 接続方法	6
2.1. Ethernet 接続例	6
2.2. RS232C 接続例	6
3. 通信設定	7
3.1. Ethernet の場合	7
3.1.1. CV-X シリーズの通信設定	7
3.1.2. ロボットコントローラの通信設定	9
3.2. RS232C の場合	10
3.2.1. CV-X シリーズの通信設定	10
3.2.2. ロボットコントローラの通信設定	10
4. プロバイダ実行手順	11
5. コマンドの説明	12
表 5-1 コマンド一覧	13
6. エラーコード	38
7. サンプルプログラム	38
改訂履歴	39

1. 本製品（プロバイダ）の概要

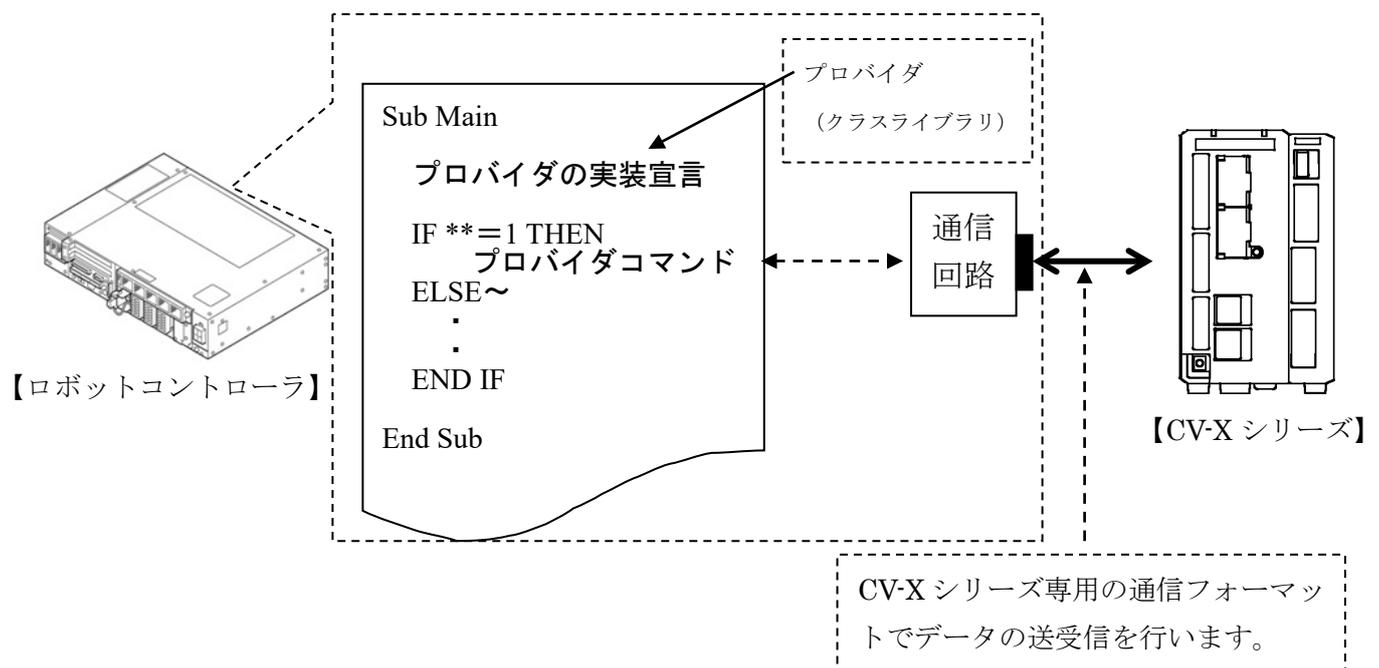
1.1. プロバイダの対象機器

本プロバイダは、デンソーロボットコントローラ（RC8シリーズ）とCV-Xシリーズを接続する時に使用することができます。



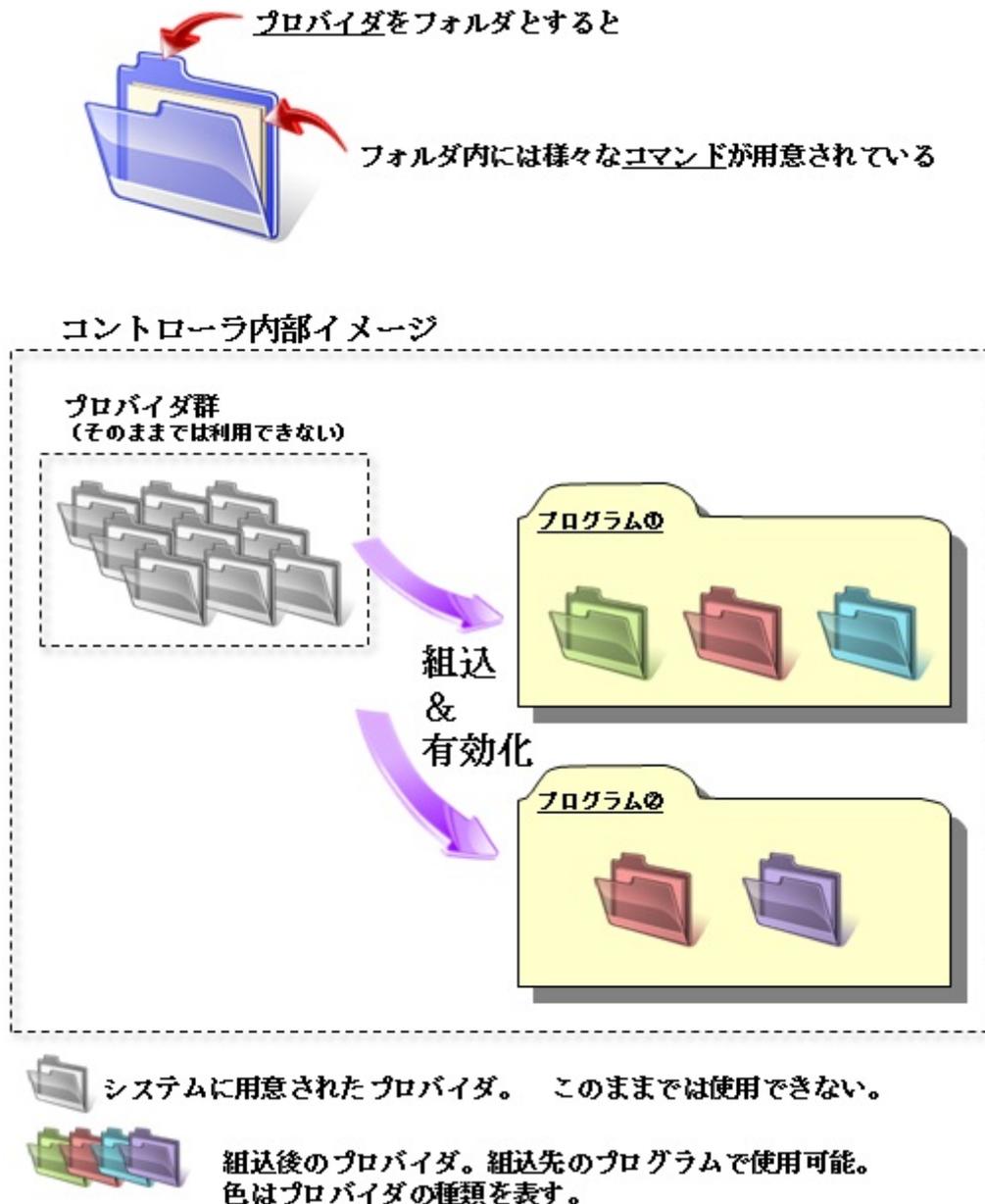
1.2. プロバイダの特長

CV-Xシリーズにアクセスするために必要なCV-Xシリーズのネイティブコマンドを、ロボットプログラムで使用できるプロバイダとして準備しています。本プロバイダを使用することで、CV-Xシリーズの通信部分のプログラムを組むことなく、容易にロボットからの通信を行うことができます。下記にプロバイダの組込関係を示します。



1.3. プロバイダの仕組み

本プロバイダは対象製品の制御を行うための各種プログラムをひとつのプロバイダとして提供しています。使用にあたってはライセンスを有効化するだけで使用する事が出来ます。使用したいプログラムファイルで実装の宣言をすれば、それ以降はプロバイダが用意する関数をコマンドとしてユーザープログラム内で使用する事が可能です。本プロバイダはコントローラ内に予め用意されていますので、インストール作業は不要です。又、違う種類のプロバイダであれば複数個実装する事も可能です。但し、同じ種類のプロバイダは同じプログラム（プロシージャ）内で存在する事は出来ません。



注：上図のプロバイダ  の様に、同一のプロバイダがプログラム別に存在する場合は、プログラム（タスク）間で排他処理を行う必要があります。

※プロバイダは PacScript から使用できるダイナミックリンクライブラリ（以下 DLL）として用意されています。

2. 接続方法

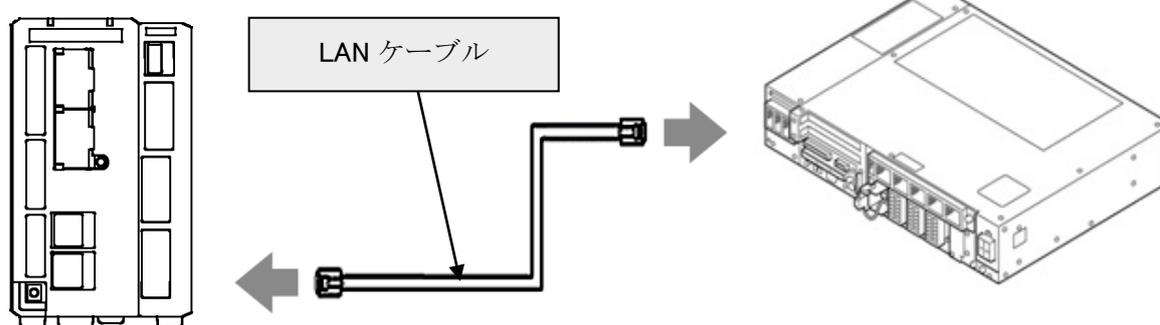
ロボットコントローラとCV-Xシリーズは、EthernetかRS232Cによって通信することができます。接続する際は各種通信用のケーブルで接続します。各種通信用ケーブルの詳細は、キーエンス社のCV-Xシリーズセットアップマニュアルを参照してください。

2.1. Ethernet 接続例

Ethernet通信するには、クロスLANケーブルを使用して下さい。又、スイッチングハブ/ルータを使用する場合は、スイッチングハブ/ルータの仕様に合ったケーブルをご使用下さい。

【CV-X シリーズ】

【ロボットコントローラ】

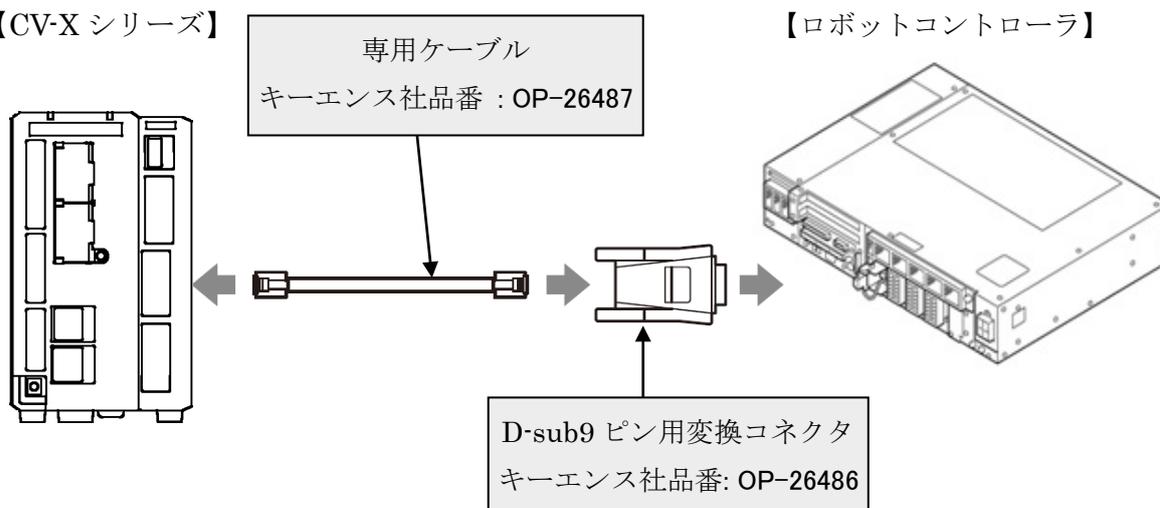


2.2. RS232C 接続例

RS232C通信するには、別売りの専用ケーブルと変換コネクタ（それぞれキーエンス社より発売）を使用します。変換コネクタは2種類ありますが、ロボットコントローラのRS232CコネクタはD-sub9ピンなので、D-sub9ピン用のものを使用します。

【CV-X シリーズ】

【ロボットコントローラ】



3. 通信設定

3.1. Ethernet の場合

3.1.1. CV-X シリーズの通信設定

CV-Xシリーズの通信設定は、CV-Xシリーズ本体に接続したモニター（別売り）に表示される設定画面を、CV-Xシリーズに同梱されているマウスで操作して行います。詳細はキーエンス社のCV-Xシリーズ ユーザーマニュアルを参照してください。

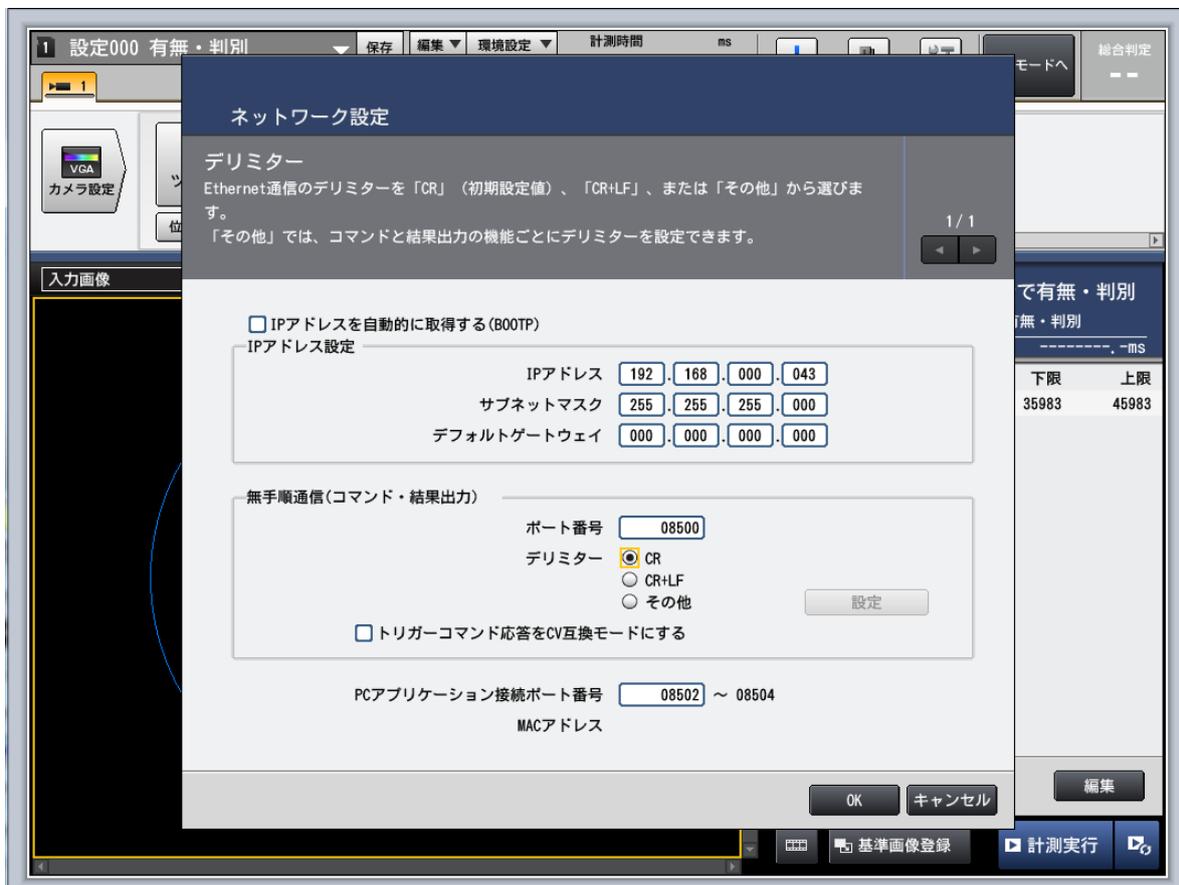
なお、設定項目の中で、一部の項目は、次の設定以外に設定しないでください。

設定項目	設定値
IPアドレスを自動的に取得する(BOOTP)	チェックしない
デリミタ	CR
トリガーコマンド応答をCV互換モードにする	チェックしない

ここでは、CV-Xシリーズの設定例を説明します。

CV-Xシリーズの設定画面を、[環境設定] - [外部入出力設定] - [ネットワーク]と操作すると、ネットワーク設定ウィンドウが表示されます。

- ・ 「IPアドレスを自動的に取得する(BOOTP)」のチェックボックスはチェックしません。
- ・ IPアドレス及び、サブネットマスクはロボットコントローラとCV-Xシリーズが、同一のサブネットマスク内になるように設定して下さい。ここでは、IPアドレスに192.168.0.43を、サブネットマスクに255.255.255.0を設定してします。
- ・ ゲートウェイは必要に応じて設定してください。ここでは0.0.0.0を設定しています。
- ・ ポート番号には任意のポート番号を設定してください。ここで設定したポート番号は、ロボットコントローラのCao.AddControllerコマンドを実行する時にオプションで指定するポート番号となります。ここでは08500を設定しています。
- ・ デリミタは「CR」に設定します。
- ・ 「トリガーコマンド応答をCV互換モードにする」チェックボックスはチェックしません。
- ・ 「PCアプリケーション接続ポート番号」は本プロバイダには関係ありません。



3.1.2. ロボットコントローラの通信設定

ロボットコントローラのEthernetの通信設定は、ティーチングペンダントかミニペンダントのどちらかで行えます。各設定方法はDENSO ROBOT USER MANUALSの下記内容を参照してください。

設定機器	参照箇所
ティーチングペンダント	ティーチングペンダント操作ガイドの「通信設定画面の表示・変更」
ミニペンダント	ミニペンダント操作ガイドの「DHCPの設定」、「IPアドレスの設定」

ここではティーチングペンダントによる設定例を説明します。

[F6設定]・[F5通信と起動権]・[F2ネットワークと通信権]と操作すると、通信設定ウィンドウが表示されます。

- ・ 通信権はCV-Xシリーズとの通信には関係ありません。
- ・ DHCPを有効にすると、IPアドレスが自動的に設定されます（ただし、同じネットワークにDHCPサーバーがつかっている必要があります）。ここでは無効に設定しています。
- ・ DHCPを無効にした場合、ロボットコントローラとCV-Xシリーズが、同一のサブネットマスク内になるようにIPアドレス及び、サブネットマスクを設定して下さい。ここでは、IPアドレスに192.168.0.1を、サブネットマスクに255.255.255.0を設定してします。
- ・ ゲートウェイは必要に応じて設定してください。ここでは0.0.0.0を設定しています。



3.2. RS232C の場合

3.2.1. CV-X シリーズの通信設定

CV-XシリーズのRS232Cの通信設定は、CV-Xシリーズ本体に接続したモニタ（別売り）に表示される設定画面を、CV-Xシリーズに同梱されているマウスで操作して行います。詳細はキーエンス社のCV-Xシリーズ ユーザマニュアルを参照してください。

なお、設定項目の中で、一部の項目は、次の設定以外に設定しないでください。

設定項目	設定値
フロー制御	なし
デリミタ	CR
トリガーコマンド応答をCV互換モードにする	チェックしない

ここでは、CV-Xシリーズの設定例を説明します。

CV-Xシリーズの設定画面を、[環境設定] - [外部入出力設定] - [RS-232C]と操作すると、RS-232C（無手順）の設定ウィンドウが表示されます。上記設定項目以外は任意に設定してください。



3.2.2. ロボットコントローラの通信設定

ロボットコントローラのRS232Cの通信設定は、[Cao.AddController](#) コマンド実行時のオプションで指定します。CV-XシリーズのRS232Cの通信設定にあわせてオプションを指定してください。

なお、ティーチングペンダントまたはミニペンダントでもRS232Cの通信設定を設定できますが、この設定は、Comm.Openコマンドを実行する時に使用される設定で、本プロバイダでは使用されません。

4. プロバイダ実行手順

プロバイダは実装(宣言)→実行が基本の手順になります。本プロバイダは実装時に接続処理を行います。操作は必要な分だけ繰り返す事が出来ます。プログラム例を下記に示します。

Sub Main

```

On Error Goto ErrorProc      ①          '異常処理ルーチンの宣言
Dim caoCtrl as Object        ②          'プロバイダ用変数宣言
Dim vntResult as Variant     ③          '結果取得用変数宣言

caoCtrl = Cao.AddController("CVX", "CaoProv.KEYENCE.CVX", "", "conn=eth:192.168.0.10") ④

「トリガ ~ データ受信処理を記述」      ⑤

```

EndProc:

```

'終了処理
Exit Sub

```

ErrorProc:

```

'異常処理

```

End Sub

- ① 必要があればプロバイダ異常時の処理ルーチンを宣言します。(宣言時の接続異常検出)
- ② プロバイダを実装させる変数を Object 型で宣言します。変数名は任意に指定できます。
- ③ 結果を取得する変数を宣言します。データ型はコマンドにより違います。
- ④ プロバイダ宣言コマンド [Cao.AddController](#) で実装します。設定に必要なパラメータはプロバイダで違います。これ以降は実装変数 caoCtrl を利用してプロバイダコマンドを利用できます。
- ⑤ これ以降は、プロバイダコマンドを使用したプログラムが記述可能です。

5. コマンドの説明

本章では各コマンドについて説明します。コマンドは

- ・ 接続コマンド
- ・ CV-Xシリーズ対応コマンド
- ・ 独自拡張コマンド

に分類されます。CV-Xシリーズ対応コマンドは、CV-Xシリーズに用意されているコマンド（CV-Xシリーズコマンド）と対になっているコマンドです。CV-Xシリーズ対応コマンドに対する、CV-Xシリーズコマンドについては次のコマンド一覧を参照してください。また、CV-Xシリーズコマンドの詳細動作については、キーエンス社のCV-Xシリーズ ユーザーズマニュアルを参照してください。

表 5-1 コマンド一覧

CV-X シリーズプロバイダ コマンド	CV-X シリーズ コマンド	機能	参照 頁
接続コマンド			
Cao.AddController	—	プロバイダを変数に実装して、CV-X シリーズへの接続処理を行います。	14
CV シリーズ対応コマンド			
Trigger	T1、T2、T3、 T4、TA	トリガを入力します。	16
ChangeMode	R0、S0	運転モードまたは、停止モードに移行します。	17
ChangeModeAsync		運転モードまたは、停止モードに非同期で移行します。	17
Reset	RS	各種リセットを行います。	18
Reboot	RB	現在の検査設定を保存し、再起動します。	18
StoreSetting	SS	現在の検査設定、環境設定を保存します。	19
ClearError	CE	エラー状態をクリアします。	19
ChangeDisplayPattern	VW	指定した運転画面に表示を切り換えます。	20
ChangeInspectSetting	PW	指定した SD カードの検査設定 No.に、設定を切り換えます。	21
ChangeInspectSettingAsync		非同期で指定した SD カードの検査設定 No.に、設定を切り換えます。	22
ReadInspectSetting	PR	現在設定されている検査設定 No.とその SD カード番号を取得します。	23
ChangeLightVolume	CLV	指定した照明のボリュームを設定します。	24
ChangeExecuteCondition	EXW	現在有効な実行条件番号を、指定した条件番号に設定します。	24
ReadExecuteCondition	EXR	現在有効な実行条件番号を取得します。	25
WriteCharReg	CW	指定したツール番号の判定文字列または照合パターン文字列を設定します。	26
ReadCharReg	CR	指定したツール番号の判定文字列または照合パターン文字列を取得します。	27
ChangeToolParameter	DW	指定したツールの判定条件の上限値または下限値を設定します。	28
ReadToolParameter	DR	指定したツールの判定条件の上限値または下限値を取得します。	29
ChangeFlawLevel	SLW	指定した傷ツールの傷レベルを設定します。	30
ReadFlawLevel	SLR	指定した傷ツールの傷レベルを取得します。	30
EnableTrigger	TE	トリガ入力の許可 / 禁止を設定します。	31
独自拡張コマンド			
ExecuteCommand	—	CV-X シリーズコマンドを CV-X シリーズコマンドの書式で実行します。	31
ExecuteCommandAsync	—	CV-X シリーズコマンドを非同期で CV-X シリーズコマンドの書式で実行します。	32
TriggerAndGetResult	—	トリガを入力し、結果データを取得します。	33
RecievePacket	—	トリガの入力に対する結果を取得します。	34
ClearPacket	—	ロボットコントローラに蓄積された結果データを消去します。	35
SetTimeout	—	タイムアウト時間を設定します。	35
GetTimeout	—	設定されているタイムアウト時間を取得します。	36
GetCommandResult	—	非同期コマンドの完了待ちを行い、非同期コマンドの戻り値を取得します。	37

Cao.AddController

機能

プロバイダを変数に実装して、CV-Xシリーズへの接続処理を行います。

書式

Cao.AddController(<コントローラ名>,<プロバイダ名>,
<プロバイダの実行マシン名>,<オプション>)

引数

<コントローラ名>

任意名を付けて下さい（名前で管理しています）（文字列）。

<プロバイダ名>

"CaoProv.KEYENCE.CVX"を指定してください（文字列）。

<プロバイダの実行マシン名>

""を指定してください（文字列）。

<オプション>

次の項目を指定します（文字列）。

書式

"Conn=<接続パラメータ>,Timeout=<タイムアウト時間>"

引数

<接続パラメータ>

通信方法によって異なります。次の「接続パラメータの説明」を参照してください。

<タイムアウト時間>

本プロバイダのコマンド実行時に、CV-Xシリーズからの応答を待つ時間（ミリ秒）を指定します。省略可能で、省略すると500（ミリ秒）が指定されます。

接続パラメータの説明

Ethernetの場合

書式

"eth:<IPアドレス>:<ポート番号>"

引数

<IPアドレス>

接続するCV-XシリーズのIPアドレスを指定します。

<ポート番号>

接続するCV-Xシリーズのポート番号を指定します。省略可能で、省力すると8500が設定されます。

RS232Cの場合

書式

com:<COMポート番号>:<通信速度>:<パリティ>
:<データビット数>:<ストップビット数>:<フロー制御>

引数

<COMポート番号>

接続するCV-XシリーズにつながっているロボットコントローラのCOMポート番号を指定します。

指定した番号がそのままCOMポートの番号となります。たとえば1を指定した場合、COM1を指定したこととなります。

ロボットコントローラのフロント側のシリアル通信用コネクタを使用する場合で増設RS232Cを使用していない場合は、2を指定してください。

<通信速度>

接続するCV-Xシリーズの通信速度に合わせて指定してください。

4800、9600、19200、38400、57600、115200 (bps) のいずれかを指定することができます。

省略可能で、省略した場合、9600が指定されます。

<パリティ>

接続するCV-Xシリーズのパリティに合わせて指定してください。指定方法は次の通りです。

N: なし

E: 偶数パリティ

O: 奇数パリティ

省略可能で、省略するとNが指定されます。

<データビット数>

接続するCV-Xシリーズのデータビット数に合わせて指定してください。指定方法は次の通りです。

7: 7ビット

8: 8ビット

省略可能で、省略すると8が指定されます。

<ストップビット数>

接続するCV-Xシリーズのストップビット数に合わせて指定してください。指定方法は次の通りです。

1: 1ビット

2: 2ビット

省略可能で、省略すると1が指定されます。

<フロー制御>

フロー制御の方法を指定することができますが、CV-Xシリーズと通信する場合は、フロー制御なしを指定してください。指定方法は次の通りです。

0: フロー制御なし

1: Xon / Xoff

2: ハードウェア制御

省略可能で、省略するとフロー制御なしが指定されます。

戻り値 説明

実装したオブジェクトです (Object)。

プロバイダを変数に実装すると同時に有効にします。これ以降は実装したObject型変数を使用してプロバイダにアクセスします。(実装された変数を"実装変数"と呼びます。)

実装変数.ChangeMode

機能 運転モードまたは、停止モードに移行します。

書式 実装変数.ChangeMode <モード番号>

引数 <モード番号>

変更先のモードを指定します（整数）。

0：停止モード

1：運転モード

戻り値 なし

説明 運転モードまたは、停止モードに移行します。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX","CaoProv.KEYENCE.CVX",_
    "", "conn=eth:192.168.0.10")
```

'運転モードに切り替える。

```
caoCtrl.ChangeMode 1
```

実装変数.ChangeModeAsync

機能 運転モードまたは、停止モードに非同期で移行します。

書式 実装変数.ChangeModeAsync <モード番号>

引数 <モード番号>

変更先のモードを指定します（整数）。

0：停止モード

1：運転モード

戻り値 なし

説明 運転モードまたは、停止モードに非同期で移行します。

コマンドの戻り値はGetCommandResultコマンドで取得し、確認してください。

用例 Dim caoCtrl as Object

```
Dim vntResult as Variant
```

```
caoCtrl=Cao.AddController("CVX","CaoProv.KEYENCE.CVX",_
    "", "conn=eth:192.168.0.10")
```

'運転モードに切り替える。

```
caoCtrl.ChangeModeAsync 1
```

'ChangeModeAsyncコマンドの戻り値取得

```
vntResult = caoCtrl.GetCommandResult
```

実装変数.Reset

機能 各種リセットを行います。

書式 実装変数.Reset

引数 なし。

戻り値 なし。

説明 各種リセットを行います。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX", "CaoProv.KEYENCE.CVX ", _  
                        "", "conn=eth:192.168.0.10")
```

```
'リセットする。  
caoCtrl.Reset
```

実装変数.Reboot

機能 現在の検査設定を保存し、再起動します。

書式 実装変数.Reboot

引数 なし。

戻り値 なし。

説明 現在の検査設定を保存し、再起動します。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX", "CaoProv.KEYENCE.CVX ", _  
                        "", "conn=eth:192.168.0.10")
```

```
'現在の検査設定を保存し、再起動する。  
caoCtrl.Reboot
```

実装変数.StoreSetting

機能 現在の検査設定、環境設定を保存します。

書式 実装変数.StoreSetting

引数 なし。

戻り値 なし。

説明 現在の検査設定、環境設定を保存します。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _  
                        "", "conn=eth:192.168.0.10")
```

'現在の検査設定、環境設定を保存する。
caoCtrl.StoreSetting

実装変数.ClearError

機能 エラー状態をクリアします。

書式 実装変数.ClearError

引数 なし。

戻り値 なし。

説明 エラー状態をクリアします。エラー状態でないときも、正常終了します。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _  
                        "", "conn=eth:192.168.0.10")
```

'エラー状態をクリアする。
caoCtrl.ClearError

実装変数.ChangeDisplayPattern

機能 指定した運転画面に表示を切り換えます。
書式 実装変数.ChangeDisplayPattern <画面種別>,<画面番号>
引数 <画面種別>

画面種別を指定します（整数）。

0：カメラ画面

1：運転画面

<画面番号>

画面番号を指定します（整数）。

0～4 ： カメラNo. (1～4、0は全てのカメラ)

0～9 ： 運転画面No. (S00～S09)

戻り値 なし。

説明 指定した運転画面に表示を切り換えます。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX","CaoProv.KEYENCE.CV",_
    "", "conn=eth:192.168.0.10")
```

'カメラ画面をカメラ1に切り換える。

```
caoCtrl.ChangeDisplayPattern 0,1
```


実装変数.ChangeInspectSettingAsync

機能 非同期で指定したSDカードの検査設定No.に、設定を切り換えます。
書式 実装変数.ChangeInspectSettingAsync <SDカード番号>,<検査設定No.>

引数 <SDカード番号>
SDカード番号を指定します（整数）。
1 : SD1
2 : SD2
<検査設定No.>
検査設定No.を指定します（整数）。0～999の間で指定してください。

戻り値 なし。

説明 非同期で指定したSDカードの検査設定No.に、設定を切り換えます。
コマンドの戻り値はGetCommandResultコマンドで取得し、確認してください。

用例 Dim caoCtrl as Object
Dim vntResult as Variant

caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _
"", "conn=eth:192.168.0.10")

'SD1の検査設定No.1に設定を切り換える。
caoCtrl.ChangeInspectSettingAsync 1,1

'ChangeInspectionSettingAsyncコマンドの戻り値取得
vntResult = caoCtrl.GetCommandResult

実装変数.ReadInspectSetting

機能 現在設定されている検査設定No.とそのSDカード番号を取得します。

書式 実装変数.ReadInspectSetting

引数 なし

戻り値 次の2つの値が整数型の配列に格納されます。

<SDカード番号>

現在設定されているSDカード番号です。

1 : SD1

2 : SD2

<検査設定No. >

現在設定されている検査設定No.です。

説明 現在設定されている検査設定No.とそのSDカード番号を取得します。

用例

```
Dim caoCtrl as Object
Dim vntRet as Variant
Dim iaryData(1) as Integer
```

```
caoCtrl=Cao.AddController("CVX","CaoProv.KEYENCE.CVX","", _
    "conn=eth:192.168.0.10")
```

'現在設定されている検査設定No.とそのSDカード番号を取得する。

'iaryData(0)にはSDカード番号が

'iaryData(1)には検査設定No.が格納される。

```
vntRet = caoCtrl.ReadInspectSetting
```

```
iaryData(0) = vntRet(0)
```

```
iaryData(1) = vntRet(1)
```

実装変数.ChangeLightVolume

機能	指定した照明のボリュームを設定します。
書式	実装変数.ChangeLightVolume <照明No.>,<照明ボリューム値>,<撮像回または撮像箇所>,<複数回撮像時の照明>
引数	<p><照明No.></p> <p>照明No.を指定します（整数）。1～8の間で指定してください。</p> <p><照明ボリューム値></p> <p>照明ボリューム値を指定します（整数）。0～255の間で指定してください。</p> <p><撮像回または撮像箇所></p> <p>撮像回または撮像箇所を指定します（整数）。1～8の間で指定してください。省略可能で、省略すると、撮像回または撮像箇所は変更されません。</p> <p><複数回撮像時の照明></p> <p>複数回撮像時の照明を指定します（整数）。1～2の間で指定してください。省略可能で、省略すると、複数回撮像時の照明は変更されません。</p>
戻り値	なし。
説明	指定した照明のボリュームを設定します。
用例	<pre>Dim caoCtrl as Object caoCtrl=Cao.AddController("CVX","CaoProv.KEYENCE.CVX",_ "", "conn=eth:192.168.0.10") '照明1のボリュームを30に設定する。 caoCtrl.ChangeLightVolume 1,30</pre>

実装変数.ChangeExecuteCondition

機能	現在有効な実行条件番号を、指定した条件番号に設定します。
書式	実装変数.ChangeExecuteCondition <実行条件番号>
引数	<p><実行条件番号></p> <p>実行条件番号を指定します（整数）。0～99の間で指定してください。</p>
戻り値	なし。
説明	現在有効な実行条件番号を、指定した条件番号に設定します。
用例	<pre>Dim caoCtrl as Object caoCtrl=Cao.AddController("CVX","CaoProv.KEYENCE.CVX",_ "", "conn=eth:192.168.0.10") '実行条件番号を1に設定する。 caoCtrl.ChangeExecuteCondition 1</pre>

実装変数.ReadExecuteCondition

機能 現在有効な実行条件番号を取得します。

書式 実装変数.ReadExecuteCondition

引数 なし。

戻り値 <実行条件番号>

取得した実行条件番号です（整数）。

説明 現在有効な実行条件番号を取得します。

用例 Dim caoCtrl as Object
Dim iParam as Integer

```
caoCtrl=Cao.AddController("CVX","CaoProv.KEYENCE.CVX",_
                        "", "conn=eth:192.168.0.10")
```

'現在有効な実行条件番号を取得する。

```
iParam = caoCtrl.ReadExecuteCondition
```

実装変数. WriteCharReg

機能 指定したツール番号の判定文字列または照合パターン文字列を設定します。

書式 実装変数. WriteCharReg <ツール番号>, <行番号 / 照合条件番号>,
<判定文字列 / 照合パターン文字列>

引数 <ツール番号>

ツール番号を指定します（整数）。100～499の間で指定してください。

<行番号 / 照合条件番号>

行番号または照合条件番号を指定します（整数）。指定するツール番号のツールが、OCRツールの場合は1を、1Dコードリーダーツールまたは2Dコードリーダーツールの場合は1～16を指定してください。

<判定文字列 / 照合パターン文字列>

判定文字列または照合パターン文字列を指定します（文字列）。

戻り値 なし。

説明 指定したツール番号のツールが、OCRツールの場合は判定文字列を、1Dコードリーダーツールまたは2Dコードリーダーツールの場合は照合パターン文字列を設定します。判定文字列または照合パターン文字列を指定しない場合は、そのツールの最新の読み取り結果を設定します。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX", "CaoProv.KEYENCE.CVX", _  
    "", "conn=eth:192.168.0.10")
```

ツール番号101のOCRツールの判定文字列を"DEF"に設定する。
caoCtrl.WriteCharReg 101,1,"DEF"

実装変数.ReadCharReg

機能	指定したツール番号の判定文字列または照合パターン文字列を取得します。
書式	実装変数.ReadCharReg (<ツール番号>,<行番号 / 照合条件番号>)
引数	<ツール番号> ツール番号を指定します（整数）。100～499の間で指定してください。 <行番号 / 照合条件番号> 行番号または照合条件番号を指定します（整数）。指定するツール番号のツールが、OCRツールの場合は1を、1Dコードリーダーツールまたは2Dコードリーダーツールの場合は1～16を指定してください。
戻り値	<判定文字列 / 照合パターン文字列> 取得した判定文字列または照合パターン文字列です（文字列）。
説明	指定したツール番号のツールが、OCRツールの場合は判定文字列を、1Dコードリーダーツールまたは2Dコードリーダーツールの場合は照合パターン文字列を取得します。
用例	<pre>Dim caoCtrl as Object Dim bstrParam as String caoCtrl=Cao.AddController("CVX", " CaoProv.KEYENCE.CVX ", _ "", "conn=eth:192.168.0.10") 'ツール番号101のOCRツールの判定文字列を取得する。 bstrParam = caoCtrl.WriteCharReg(101,1)</pre>

実装変数.ChangeToolParameter

機能	指定したツールの判定条件の上限値または下限値を設定します。
書式	実装変数.ChangeToolParameter <ツール番号>,<判定条件種別の項目ID>,<上限 / 下限>,<判定条件の値>
引数	<p><ツール番号></p> <p>ツール番号を指定します（整数）。</p> <p><判定条件種別の項目ID></p> <p>判定条件種別の項目を指定します（整数）。詳細はキーエンス社のCV-Xシリーズ ユーザーマニュアルのDWコマンドを参照してください。</p> <p><上限 / 下限.></p> <p>上限または下限.を指定します（整数）。</p> <p>0： 上限値</p> <p>1： 下限値</p> <p><判定条件の値.></p> <p>判定条件の値を指定します（文字列）。</p>
戻り値	なし。
説明	指定したツールの判定条件の上限値または下限値を設定します。
用例	<pre>Dim caoCtrl as Object caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _ "", "conn=eth:192.168.0.10") 'ツール番号100のエッジツールの下限値を123.456に設定する。 caoCtrl.ChangeToolParameter 100,82,1,"123.456"</pre>

実装変数.ReadToolParameter

機能	指定したツールの判定条件の上限値または下限値を取得します。
書式	実装変数.ReadToolParameter (<ツール番号>,<判定条件種別の項目ID>,<上限 / 下限>)
引数	<ツール番号> ツール番号を指定します（整数）。 <判定条件種別の項目ID > 判定条件種別の項目を指定します（整数）。詳細はキーエンス社のCV-Xシリーズ ユーザーマニュアルのDRコマンドを参照してください。 <上限 / 下限.> 上限または下限.を指定します（整数）。 0： 上限値 1： 下限値
戻り値	<判定条件の値.> 取得した判定条件の値です（文字列）。
説明	指定したツールの判定条件の上限値または下限値を取得します。
用例	<pre>Dim caoCtrl as Object Dim bstrParam as String caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _ "", "conn=eth:192.168.0.10") 'ツール番号100のエッジツールの下限値を取得する。 bstrParam = caoCtrl.ReadToolParameter(100,82,1)</pre>

実装変数.ChangeFlawLevel

機能 指定した傷ツールの傷レベルを設定します。

書式 実装変数.ChangeFlawLevel <ツール番号>,<傷レベルの値>

引数 <ツール番号>
ツール番号を指定します（整数）。
<傷レベルの値>
傷レベルの値を指定します（整数）。

戻り値 なし。

説明 指定した傷ツールの傷レベルを設定します。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _  
                        "", "conn=eth:192.168.0.10")
```

'ツール番号102の傷レベルを200に設定する。
caoCtrl.ChangeFlawLevel 102,200

実装変数.ReadFlawLevel

機能 指定した傷ツールの傷レベルを取得します。

書式 実装変数. ReadFlawLevel (<ツール番号>)

引数 <ツール番号>
ツール番号を指定します（整数）。

戻り値 <傷レベルの値>
取得した傷レベルの値です（文字列）。

説明 指定した傷ツールの傷レベルを取得します。

用例 Dim caoCtrl as Object
Dim bstrParam as String

```
caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _  
                        "", "conn=eth:192.168.0.10")
```

'ツール番号102の傷レベルを取得する。
bstrParam = caoCtrl.ReadFlawLevel(102)

実装変数.EnableTrigger

機能 トリガ入力の許可 / 禁止を設定します。

書式 実装変数.EnableTrigger <許可 / 禁止>

引数 <許可 / 禁止>

トリガの許可 / 禁止を指定します (整数)。

0 : トリガ禁止

1 : トリガ許可

戻り値 なし。

説明 トリガ入力の許可 / 禁止を設定します。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX", "CaoProv.KEYENCE.CVX", _
    "", "conn=eth:192.168.0.10")
```

'トリガ入力を禁止に設定する。

```
caoCtrl.EnableTrigger 0
```

実装変数.ExecuteCommand

機能 CV-XシリーズコマンドをCV-Xシリーズコマンドの書式で実行します。

書式 実装変数.ExecuteCommand (<CV-Xシリーズコマンド書式>)

引数 <CV-Xシリーズコマンド書式>

CV-Xシリーズコマンド書式を指定します (文字列)。

戻り値 < CV-Xシリーズコマンド実行後の受信データ>

CV-Xシリーズコマンド実行後の受信データです (文字列)。

説明 CV-XシリーズコマンドをCV-Xシリーズコマンドの書式で実行します。CV-Xシリーズコマンドについては、キーエンス社のCV-Xシリーズ ユーザーズマニュアルを参照してください。

用例 Dim caoCtrl as Object

```
Dim strRet as String
```

```
caoCtrl=Cao.AddController("CVX", "CaoProv.KEYENCE.CVX", _
    "", "conn=eth:192.168.0.10")
```

'ツール番号100のエッジツールの下限値を123.456に設定する。

'コマンドが正常に実行されれば、strRetには"DW"が格納される。

'コマンドが失敗すると、strRetには

"ER,DW,nn" (nnにはエラーコードが入る。) が格納される。

```
strRet = caoCtrl.ExecuteCommand("DW,100,82,1,123.456")
```

実装変数.ExecuteCommandAsync

機能	CV-Xシリーズコマンドを非同期でCV-Xシリーズコマンドの書式で実行します。
書式	実装変数.ExecuteCommandAsync <CV-Xシリーズコマンド書式>
引数	<CV-Xシリーズコマンド書式> CV-Xシリーズコマンド書式を指定します（文字列）。
戻り値	なし
説明	CV-Xシリーズコマンドを非同期でCV-Xシリーズコマンドの書式で実行します。CV-Xシリーズコマンドについては、キーエンス社のCV-Xシリーズ ユーザーズマニュアルを参照してください。 コマンドの戻り値はGetCommandResultコマンドで取得し、確認してください。
用例	<pre>Dim caoCtrl as Object Dim vntResult as Variant caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _ "", "conn=eth:192.168.0.10") 'ツール番号100のエッジツールの下限値を123.456に設定する。 ' ExecuteCommandAsync コマンドの戻り値取得 vntResult = caoCtrl.GetCommandResult</pre>

実装変数.TriggerAndGetResult

機能	トリガを入力し、結果データを取得します。
書式	実装変数.TriggerAndGetResult (<トリガ番号>)
引数	<トリガ番号> トリガ番号を指定します（整数）。 1~4 : トリガ1~4
戻り値	<結果データ> 取得した結果データです（文字列）。
説明	トリガを入力した後、結果データを取得します。結果データがCV-Xシリーズから送信されていない場合、タイムアウト時間（ Cao.AddController コマンドのオプションまたは、 SetTimeout コマンドで設定 ）まで待ち、それでも送信されない場合はエラーとなります。トリガを入力した後、結果データが送信されるまでの時間に、他の処理を行いたい場合は、 Trigger コマンドでトリガを入力した後、任意の処理を実行し、その後、 RecievePacket コマンドで結果データを取得してください。
用例	<pre>Dim caoCtrl as Object Dim strRet as String caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _ "", "conn=eth:192.168.0.10") 'トリガ1にトリガを入力し、結果を取得する。 strRet = caoCtrl.TriggerAndGetResult(1)</pre>

実装変数.RecievePacket

機能 トリガの入力に対する結果を取得します。

書式 実装変数.RecievePacket

引数 なし。

戻り値 <結果データ>

取得した結果データです（文字列）。

説明

トリガの入力に対する結果を取得します。

CV-Xシリーズの設定が、トリガ入力に対し結果を出力しない設定になっている場合、結果データがCV-Xシリーズから送信されないため、タイムアウト時間（[Cao.AddController](#)コマンドのオプションまたは、[SetTimeout](#)コマンドで設定）経過後にエラーとなります。

また、トリガの入力を行った後、[RecievePacket](#)コマンドを実行しないで、再びトリガの入力を行うと、2回分の結果データがロボットコントローラに蓄積されます。その後[RecievePacket](#)コマンドを実行すると、最初のトリガの入力に対する結果データを取得することになります。したがって、トリガの入力回数に対し、[RecievePacket](#)コマンドの実行回数が伴わなくなってしまった場合は、[ClearPacket](#)コマンドでロボットコントローラに蓄積された結果データを消去し、再度トリガを入力した後、[RecievePacket](#)コマンドで結果データを取得してください。

用例

```
Dim caoCtrl as Object
```

```
Dim strRet as String
```

```
caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _  
    "", "conn=eth:192.168.0.10")
```

```
'トリガ1にトリガを入力する。
```

```
caoCtrl.Trigger 1
```

```
'結果データを取得する。
```

```
strRet = caoCtrl.RecievePacket
```

実装変数.ClearPacket

機能 ロボットコントローラに蓄積された結果データを消去します。

書式 実装変数.ClearPacket

引数 なし。

戻り値 なし。

説明 ロボットコントローラに蓄積された結果データを消去します。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _  
    "", "conn=eth:192.168.0.10")
```

'結果データを消去する。

```
caoCtrl.ClearPacket
```

実装変数.SetTimeout

機能 タイムアウト時間を設定します。

書式 実装変数. SetTimeout <タイムアウト時間>

引数 <タイムアウト時間>

タイムアウト時間を指定します（整数）。単位はミリ秒です。

戻り値 なし。

説明 タイムアウト時間は、[Cao.AddController](#) コマンド実行時に指定しますが、[Cao.AddController](#) コマンド実行後にタイムアウト時間を設定したい場合に使用します。

用例 Dim caoCtrl as Object

```
caoCtrl=Cao.AddController("CVX"," CaoProv.KEYENCE.CVX ", _  
    "", "conn=eth:192.168.0.10")
```

'タイムアウト時間を1000ミリ秒に設定する。

```
caoCtrl.SetTimeout 1000
```

実装変数.GetTimeout

機能	設定されているタイムアウト時間を取得します。
書式	実装変数.GetTimeout
引数	なし。
戻り値	<タイムアウト時間> 設定されているタイムアウト時間です（整数）。単位はミリ秒です。
説明	設定されているタイムアウト時間を取得します。
用例	<pre>Dim caoCtrl as Object Dim iTimeout as Integer caoCtrl=Cao.AddController("CVX", "CaoProv.KEYENCE.CVX ", _ "", "conn=eth:192.168.0.10") 'タイムアウト時間を取得する。 iTimeout = caoCtrl.GetTimeout</pre>

実装変数.GetCommandResult

機能 非同期コマンドの完了待ちを行い、非同期コマンドの戻り値を取得します。

書式 実装変数.GetCommandResult

引数 なし。

戻り値 <非同期コマンドの戻り値>

非同期コマンドの戻り値が格納されます。

説明 非同期コマンドの完了待ちを行い、非同期コマンドの戻り値を取得します。

戻り値がない非同期コマンドを実行した場合、戻り値はありません。

また、同期コマンドの後で使用した場合は、**GetCommandResult**コマンド実行時に結果取得エラー(0x80100003)になり戻り値はありません。

非同期コマンドの実行でエラーが発生した場合、非同期コマンドの実行時にはエラーは発生せず**GetCommandResult**コマンド実行時にエラーとなります。

非同期コマンドの完了待ちの際、設定されているタイムアウト時間以内に応答がない場合タイムアウトエラー (0x80000900) が発生します。

非同期コマンド実行後に別のコマンドを実行した場合は、先に実行した非同期コマンドの結果は削除されますのでご注意ください。

用例 Dim caoCtrl as Object

Dim vntResult as variant

```
caoCtrl=Cao.AddController("CVX","CaoProv.KEYENCE.CVX",_
    "", "conn=eth:192.168.0.10")
```

'ツール番号100のエッジツールの下限値を123.456に設定する。

```
caoCtrl.ExecuteCommandAsync "DW,100,82,1,123.456"
```

'コマンドの戻り値をGetCommandResultコマンドで取得する。

```
vntResult = caoCtrl.GetCommandResult
```

6. エラーコード

プロバイダのエラーの見方に関しては、DENSO ROBOT USER MANUALSの「プロバイダガイド」の「プロバイダエラーの見方」を参照してください。

プロバイダのエラーの中で、CV-Xシリーズから送信されたエラーに関しては、オリジナルナンバが80108000（16進数）～80108063（16進数）の範囲の番号となり、下2桁がCV-Xシリーズから送信されたエラーコードを表します。例えば、[ChangeInspectSetting](#) コマンドを実行する時に、SDカード番号に5を指定して実行すると、ロボットコントローラのエラーのオリジナルナンバは80108016（16進数）となります。下2桁の16（16進数）は10進数に変換すると22であり、キーエンス社のCV-Xシリーズ ユーザマニュアルのPWコマンドの説明を参照すると、エラーコード22は"パラメータ数、パラメータ範囲が違うとき"ということがわかります。

エラー名	エラー番号	説明
E_CVERROR_CVERR	0x80108000～0x80108063	CVXシリーズ固有エラー
E_CVERROR_LENGTH	0x80100000	パケット長エラー
E_CVERROR_PACKET	0x80100001	パケット異常エラー
E_COMMAND_EXECUTING	0x80100002	コマンド実行中に別コマンドを実行しました。
E_GET_COMMAND_RESULT	0x80100003	同期コマンド実行後に GetCommandResult コマンドを実行しました。

7. サンプルプログラム

Sub Main

```
Dim caoCtrl As Object
```

```
Dim strRet As String
```

```
'CV-X シリーズプロバイダの実装
```

```
caoCtrl = Cao.AddController("CVX", "CaoProv.KEYENCE.CVX", "", _  
                             "conn=eth:192.168.0.3, timeout=1000")
```

```
'トリガ1にトリガを入力し結果データを取得する。
```

```
strRet = caoCtrl.TriggerAndGetResult(1)
```

```
'結果データをティーチングペンダントのメッセージ出力ウィンドウに出力する。
```

```
PrintDbg strRet
```

```
' CV-X シリーズプロバイダを切断し削除する。
```

```
cao.Cotrollers.Remove caoCtrl.Index
```

```
caoCtrl = Nothing
```

End Sub

改訂履歴

デンソーロボット
プロバイダ
取扱説明書
株式会社キーエンス製 画像センサ CV-X シリーズ

バージョン	対応RC8	改訂内容
Ver.1.0.0	Ver.1.1.2	初版
Ver.1.0.1	Ver.1.13.0	非同期処理の追加
Ver.1.0.2	Ver.1.13.0	サンプルプログラムの修正
Ver.1.0.3	Ver.2.3.*	誤記訂正
Ver.1.0.4	Ver.2.13.*	誤記訂正

株式会社デンソーウェーブ

- 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。万が一不審の点や誤り、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。

DENSO Robotics
THIRD PARTY PRODUCTS

株式会社 デンソーウェーブ