

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 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シリーズの内、下記の製品です(2014年6月時点)。

(株)キーエンス製 CV-3000 シリーズ / CV-5000 シリーズ

以下、上記製品をまとめてCV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。

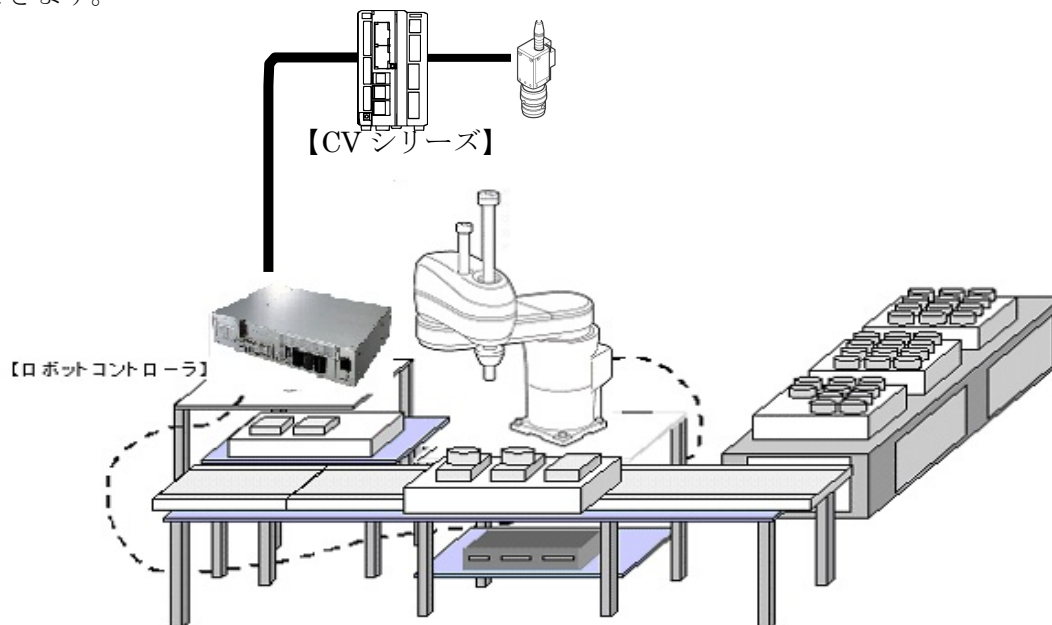
目次

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 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	8
3.2. RS232C の場合	9
3.2.1. CV シリーズの通信設定	9
3.2.2. ロボットコントローラの通信設定	9
4. プロバイダ実行手順	10
5. コマンドの説明	11
表 5-1 コマンド一覧	12
6. エラーコード	48
7. サンプルプログラム	48
改訂履歴	49

1. 本製品（プロバイダ）の概要

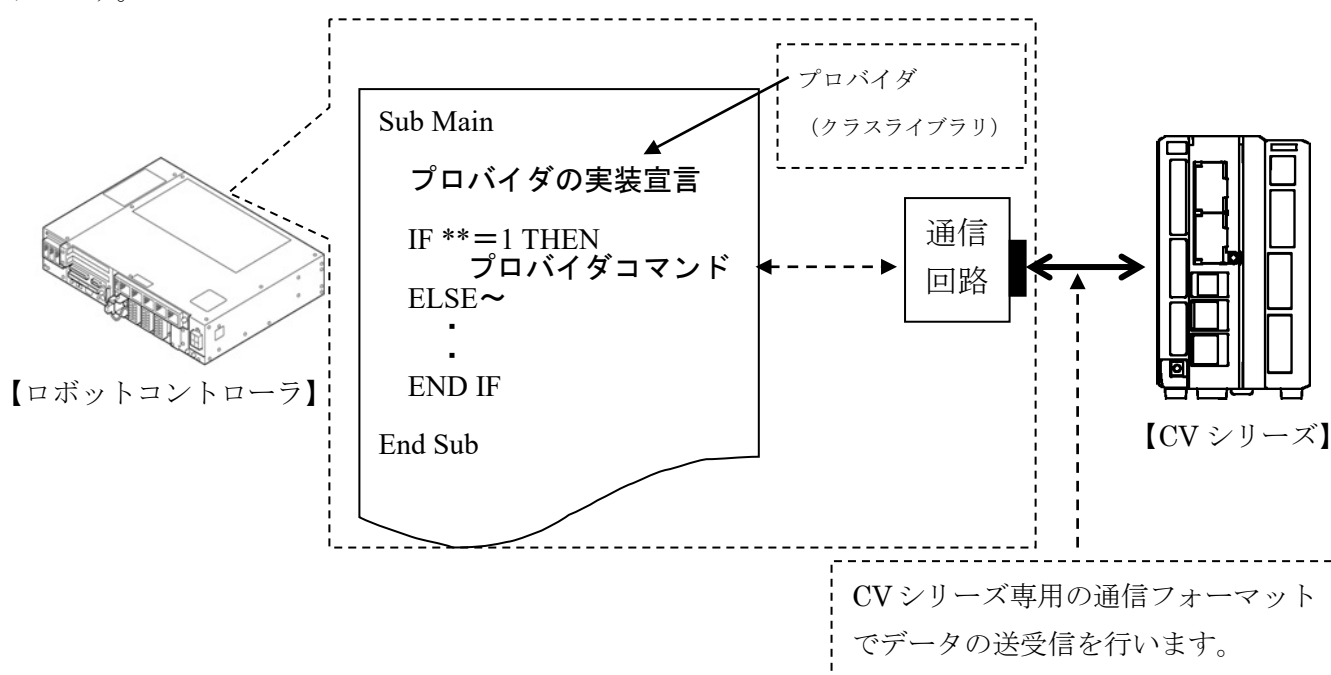
1.1. プロバイダの対象機器

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。



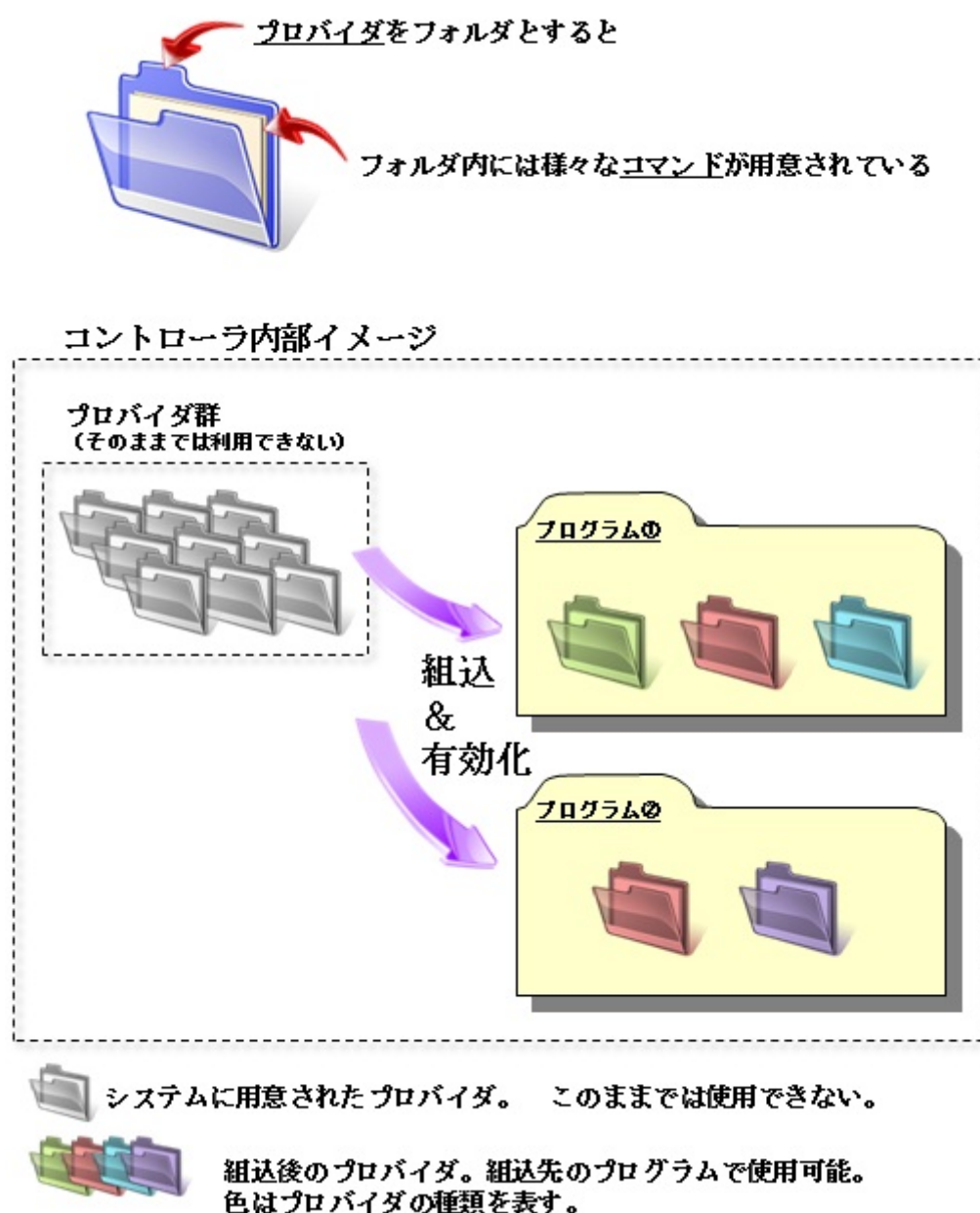
1.2. プロバイダの特長


CVシリーズにアクセスするために必要なCV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

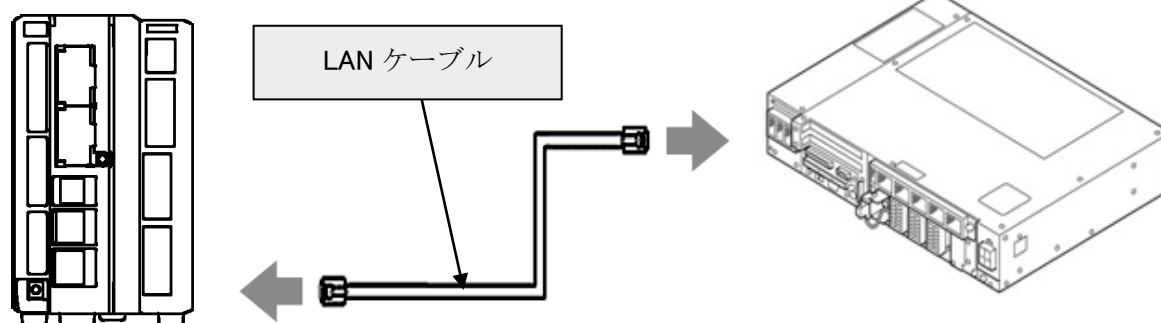
ロボットコントローラとCVシリーズは、EthernetかRS232Cによって通信することができます。接続する際は各種通信用のケーブルで接続します。各種通信用ケーブルの詳細は、キーエンス社のCVシリーズ ユーザマニュアルを参照してください。

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 シリーズ】

【ロボットコントローラ】

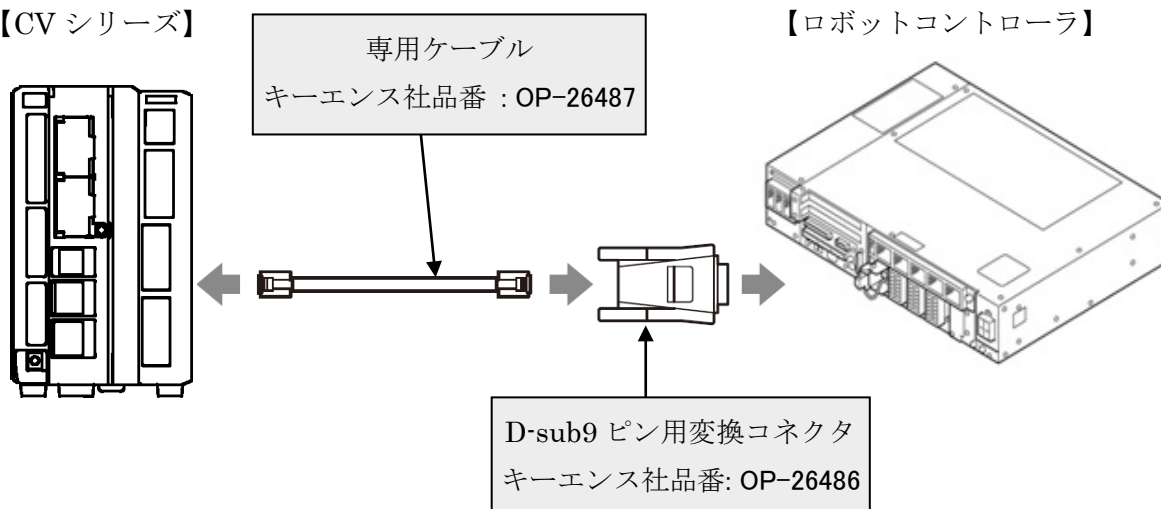


2.2. RS232C 接続例

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 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 シリーズの通信設定

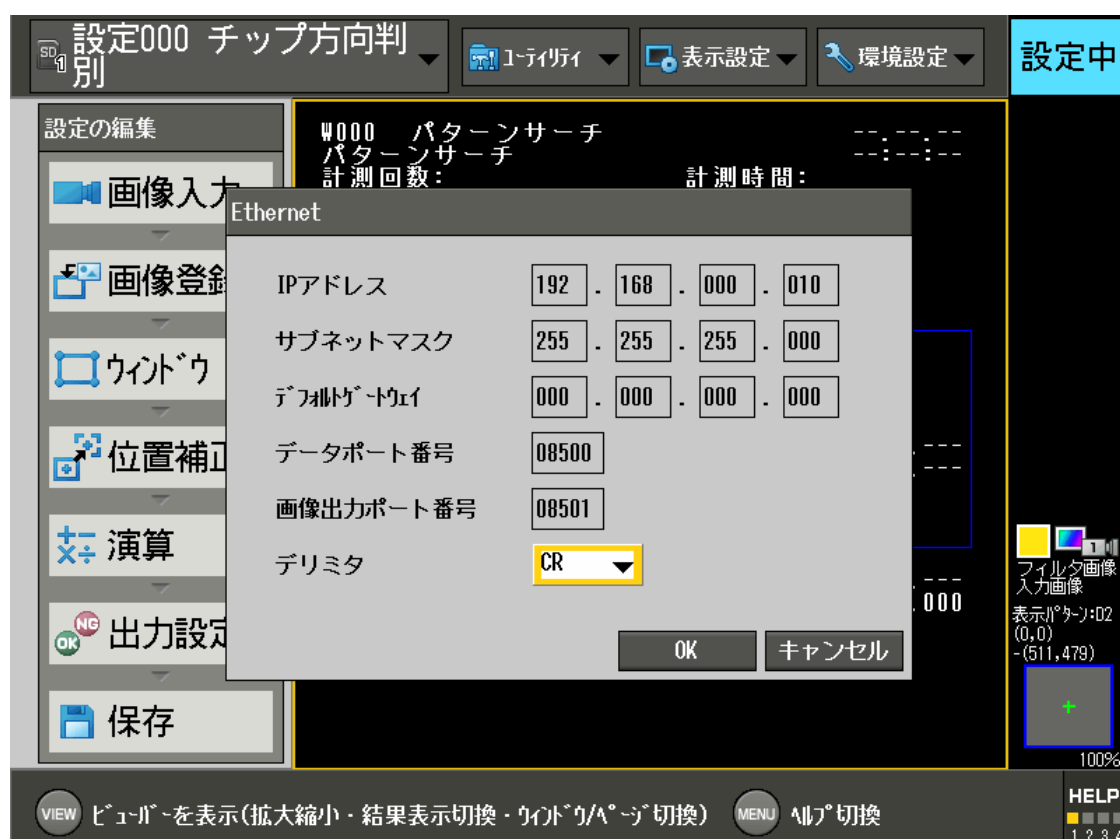
CVシリーズの通信設定は、CVシリーズ本体に接続したモニタ（別売り）に表示される設定画面を、CVシリーズに同梱されているコンソールで操作して行います。詳細はキーエンス社のCVシリーズ ユーザマニュアルを参照してください。

なお、デリミタの設定は「CR」以外には設定しないでください。

ここでは、CV-3000シリーズの設定例を説明します。

CV-3000シリーズの設定画面を、[環境設定] - [Ethernet]と操作すると、Ethernetの通信設定ウィンドウが表示されます。

- ・ 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192.168.0.10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。
- ・ データポート番号、画像出力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。ここではデータポート番号に08500を、画像出力ポート番号に08501を設定しています。
- ・ デリミタは「CR」に設定します。



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サーバーがつなが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。

通信設定

デバイス: イーサネット(192.168.0.1) 読込/書込可

設定内容	値
プロパティ	値
通信権	読込/書込可
DHCP	無効
IPアドレス	192.168.0.1
サブネットマスク	255.255.255.0
ゲートウェイ	0.0.0.0
MACアドレス	B4-B5-2F-B9-1D-18

WINCAPSとロボットコントローラの間で通信を行うための設定です。

Cancel OK

Shortcut

SHIFT 編集

3. 2. RS232C の場合

3. 2. 1. CV シリーズの通信設定

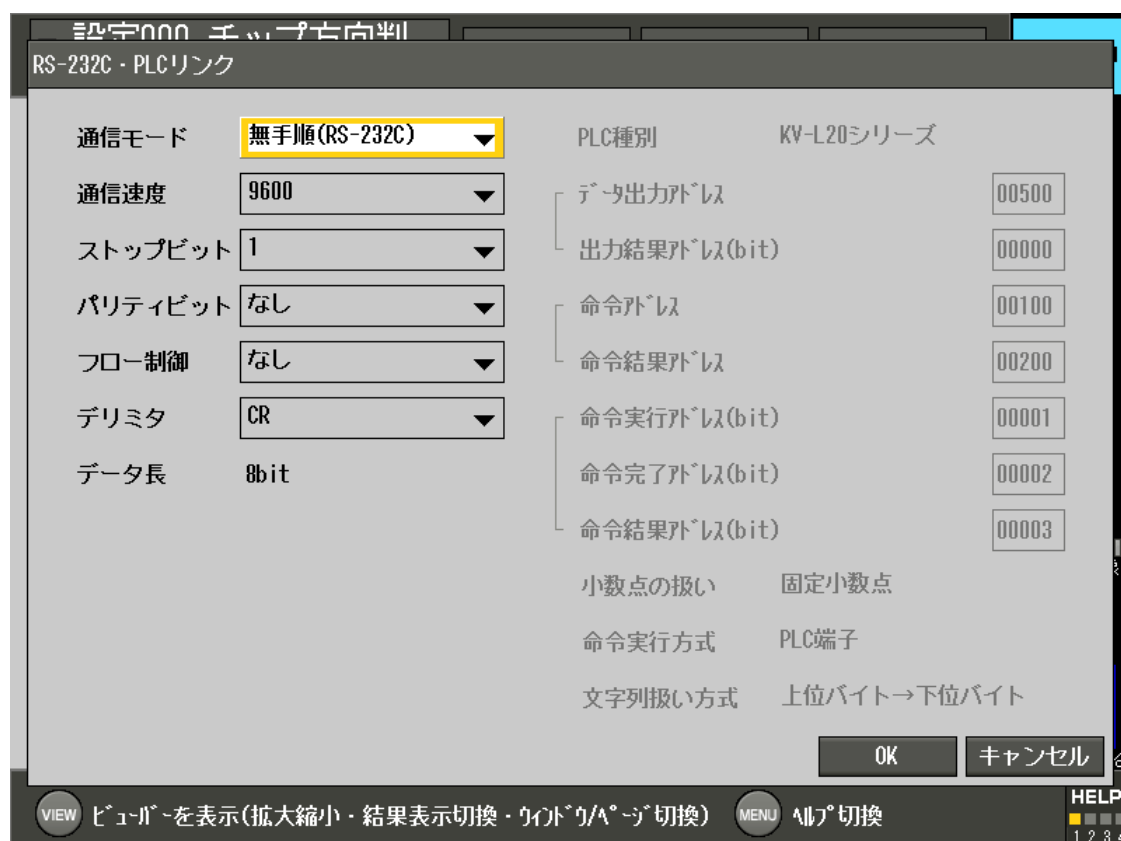
CVシリーズのRS232Cの通信設定は、CVシリーズ本体に接続したモニタ（別売り）に表示される設定画面を、CVシリーズに同梱されているコンソールで操作して行います。詳細はキーエンス社のCVシリーズユーザマニュアルを参照してください。

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
通信モード	無手順(RS-232C)
フロー制御	なし
デリミタ	CR

ここでは、CV-3000シリーズの設定例を説明します。

CV-3000シリーズの設定画面を、[環境設定] → [RS-232C・PLCリンク]と操作すると、RS-232C・PLCリンクの通信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("CV", "CaoProv.KEYENCE.CV", "", "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シリーズ対応コマンドは、CVシリーズに用意されているコマンド（CVシリーズコマンド）と対になっているコマンドです。CVシリーズ対応コマンドに対する、CV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 シリーズプロバイダ コマンド	CV 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 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	トリガを入力します。トリガを入力後、応答結果も受信することもできます。	17
ChangeMode	R0、S0	運転モードまたは、停止モードに移行します。	18
ChangeModeAsync		運転モードまたは、停止モードに非同期で移行します。	18
Reset	RS	各種リセットを行います。	19
StoreSetting	SS	現在選ばれている設定 No.の設定値を保存します。	19
ChangeDisplayPattern	DS	表示設定の切り替えを行います。	20
ChangeCurrentUnit	UW	指定したウィンドウ No.に切り替えます。	20
ReadCurrentUnit	UR	現在のウィンドウ No.を取得します。	21
ReoutputResult	M0	最新の計測結果を再読み出しします。	21
RegistImage	BS	画像を登録します。	22
GetRegistImageList	BL	登録画像の一覧を取得します。	23
SaveAllImage	BT	全画像データを読み出します。	24
GetLibraryList	DL	辞書の一覧を取得します。	25
ChangePassword	PS	パスワードを変更します。	26
InputPseudoConsole	KY	コンソール疑似入力します。	27
SaveStatisticalData	ST	統計データを保存する。	28
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。	29
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。	30
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。	31
ReadToolParameter	DR	指定したウィンドウの公差の上限値または下限値を取得します。	32
ChangeToolParameter	DW	指定したウィンドウの公差の上限値または下限値を設定します。	33
ReadBinaryData	JR	指定したウィンドウの 2 値化フィルタについて、上限値と下限値を取得します。	34
ChangeBinaryData	JW	指定したウィンドウの 2 値化フィルタについて、上限値と下限値を設定します。	35
InitCommandMemory	MI	コマンドメモリの初期設定値として、現在のコマンドメモリの値をまとめて反映します。	35
ReadCommandMemory	MR	指定したコマンドメモリのデータを取得します。	36
ChangeCommandMemory	MW	指定した最大 32 個のコマンドメモリにデータを設定します。	36
RefreshReferencePosition	RR	現在の登録画像で基準値を再計算します。	37
RefreshReferencePositionAsync		非同期で現在の登録画像で基準値を再計算します。	37
EnableTrigger	TE	トリガ入力の許可 / 禁止を設定します。	38
ChangeShutterSpeed	CW	シャッタースピードを変更します。	39
ChangeCameraSensitivity	CW	カメラの感度を変更します。	40

ChangeTriggerDelay	CW	撮像時のディレイ時間を設定します。	40
ChangeLightIntensityLevel	CW	照明のボリュームを変更します。	41
ChangePatternCounter	CW	照明点灯パターンを変更します。	41
独自拡張コマンド			
ExecuteCommand	—	CV シリーズコマンドを CV シリーズコマンドの書式で実行します。	42
ExecuteCommandAsync	—	CV シリーズコマンドを非同期で CV シリーズコマンドの書式で実行します。	43
TriggerAndGetResult	—	トリガを入力し、結果データを取得します。	44
RecievePacket	—	トリガの入力に対する結果を取得します。	45
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。	45
SetTimeout	—	タイムアウト時間を設定します。	46
GetTimeout	—	設定されているタイムアウト時間を取得します。	46
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。	47

Cao.AddController

機能

プロバイダを変数に実装して、CV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.CV"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シリーズのポート番号を指定します。省略可能で、省略すると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。)

用例

Dim caoCtrl as Object

'===== Ethernet通信の場合 =====

'ポートを省略し、タイムアウト時間は指定する場合

```
caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV", _
    "", "conn=eth:192.168.0.10, timeout=1000")
```

'ポートを指定し、タイムアウト時間は指定しない場合

```
caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV", _
    "", "conn=eth:192.168.0.10:8503")
```

'===== RS232C通信の場合 =====

'通信速度以降を省略する場合

```
caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV", _
    "", "conn= com:2")
```

'通信速度以降も指定する場合

```
caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV", _
    "", "conn= com:2:115200:E:8:1:0, timeout=1000")
```

'通信速度以降も指定する場合（パリティのみ指定して他は省略する場合）

```
caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV", _
    "", "conn= com:2::E::")
```


実装変数.Trigger

機能 書式 引数

トリガを入力します。トリガを入力後、応答結果も受信することもできます。

実装変数.Trigger (<トリガ番号>[<モード番号>])

<トリガ番号>

入力対象のトリガ番号を指定します（整数）。

1： トリガ1

2： トリガ2

<モード番号>

結果を受信するかどうかのフラグを設定します（整数）。省略可能で、省略すると1が指定されます。

0： トリガを発行し、応答結果を受信しません。

1： トリガを発行し、応答結果を受信します。

戻り値 説明

結果文字列(モード番号=1の時)

トリガを入力します。トリガを入力後、応答結果も受信することもできます。

結果を後で受信する際には先頭に **Trigger** コマンドの応答パケット「T1,」が付与されています。エラーの際には応答パケットが「ER,T1,nn」となります。nn には CV のエラーコードが格納されます。また結果を受信するまで、他のコマンドを使用しないでください。

トリガ番号 1 を発行し、x=11,y=12 の値を出力する例を以下に示します。

用例

```
Dim caoCtrl as Object
```

```
Dim strRet as String
```

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

```
'結果を受信する時
```

```
'トリガ発行&結果取得
```

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

```
'strRet: "+11,+12"
```

```
'結果を後で受信する時
```

```
'トリガだけ発行
```

```
CaoCtrl.Trigger 1,0
```

```
'結果を受信
```

```
strRet = caoCtrl.RecievePacket
```

```
'strRet: " T1,+11,+12"
```

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("CV", "CaoProv.KEYENCE.CV ", "", "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("CV", "CaoProv.KEYENCE.CV ", "", "conn=eth:192.168.0.10")
```

'運転モードに切り替える。

```
caoCtrl.ChangeModeAsync 1
```

'ChangeMoveAsync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("CV"," CaoProv.KEYENCE.CV ", "", "conn=eth:192.168.0.10")
```

'リセットする。

```
caoCtrl.Reset
```

実装変数.StoreSetting

機能 現在選ばれている設定No.の設定値を保存します。

書式 実装変数.StoreSetting

引数 なし。

戻り値 なし。

説明 現在選ばれている設定No.の設定値を保存します。

用例 Dim caoCtrl as Object

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

'現在選ばれている設定No.の設定値を保存する。

```
caoCtrl.StoreSetting
```

実装変数.ChangeDisplayPattern

機能 書式 引数

表示設定の切り替えを行います。

実装変数.ChangeDisplayPattern <切り替え対象>,<パラメータ>

<切り替え対象>

切り替え対象を指定します（文字列）。

PT : 表示パターン

RS : 結果表示

PG : ページ

FC : 画面

<パラメータ>

切り替え対象に対するパラメータを指定します（文字列）。詳細はキーエンス社のCVシリーズ ユーザーズマニュアルのDSコマンドを参照してください。

戻り値 説明 用例

なし。

表示設定の切り替えを行います。

Dim caoCtrl as Object

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

'前のページに移動する。

caoCtrl.ChangeDisplayPattern "PG","P"

実装変数.ChangeCurrentUnit

機能 書式 引数

指定したウィンドウNo.に切り替えます。

実装変数.ChangeCurrentUnit <ウィンドウNo.>

<ウィンドウNo.>

ウィンドウNo.を指定します（整数）。0～127の間で指定してください。

戻り値 説明 用例

なし。

指定したウィンドウNo.に切り替えます。

Dim caoCtrl as Object

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

'ウィンドウNo.2に切り替える。

caoCtrl.ChangeCurrentUnit 2

実装変数.ReadCurrentUnit

機能 現在のウィンドウNo.を取得します。

書式 実装変数.ReadCurrentUnit

引数 なし。

戻り値 <ウィンドウNo.>

現在のウィンドウNo.です（整数）。0～127の間の値です。

説明 現在のウィンドウNo.を取得します。

用例 Dim caoCtrl as Object
Dim iNum as Integer

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

```
'現在のウィンドウNo.を取得する。  
iNum = caoCtrl.ReadCurrentUnit
```

実装変数.ReoutputResult

機能 最新の計測結果を再読み出しします。

書式 実装変数.ReoutputResult

引数 なし

戻り値 <計測結果>

最新の計測結果です（文字列）。

説明 最新の計測結果を再読み出しします。

用例 Dim caoCtrl as Object
Dim bstrResult as String

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

```
'最新の計測結果を取得する。  
bstrResult = caoCtrl.ReoutputResult
```

実装変数.RegistImage

機能 カメラNo.で指定された最新の取り込み画像を、指定した番号の登録画像として登録します。

書式 実装変数.RegistImage <カメラNo.>,<登録画像番号>

引数 <カメラNo.>

カメラNo.を指定します（整数）。1～4の間で指定してください。

<登録画像番号>

登録画像番号を指定します（整数）。0～999の間で指定してください。

戻り値 なし。

説明 カメラNo.で指定された最新の取り込み画像を、指定した番号の登録画像として登録します。

用例 Dim caoCtrl as Object

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

'カメラNo.1に最新の取り込み画像を登録画像番号0で登録する。

```
caoCtrl.RegistImage 1,0
```

実装変数.GetRegistImageList

機能

設定されている登録画像番号を一覧で取得します。

1

現在の設定の登録画像の一覧を取得する場合

書式 実装変数.GetRegistImageList <取得先>,<カメラNo.>

引数 <取得先>

0 : 現在の設定

<カメラNo.>

カメラNo.を指定します（整数）。1～4の間で指定してください。

2

SDカードに保存されている登録画像の一覧を取得する場合

書式 実装変数.GetRegistImageList <取得先>,<登録画像番号>,<カメラNo.>

引数 <取得先>

SDカード番号を指定します（整数）。

1 : SD1

2 : SD2

<登録画像番号>

登録画像番号を指定します（整数）。0～999の間で指定してください。

<カメラNo.>

カメラNo.を指定します（整数）。1～4の間で指定してください。

戻り値

<登録画像番号一覧>

登録画像番号が整数型の配列に格納されます

説明

設定されている登録画像番号を一覧で取得します。

用例

```
Dim caoCtrl as Object
```

```
Dim vntRet as variant
```

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

```
'現在の設定の登録画像番号を一覧で取得する。
```

```
vntRet = caoCtrl. GetRegistImageList 0,1
```

実装変数.SaveAllImage

機能 書式 引数

画像バッファにあるすべての画像データを、SDカードにビットマップ方式で書き込みます。

実装変数.SaveAllImage <カメラNo.>,<圧縮率>,<画像の種類>,<フォルダ名>

<カメラNo.>

カメラNo.を指定します（整数）。1～4の間で指定してください。

<圧縮率>

圧縮率を指定します（整数）。0～3の間で指定してください。

0 : 圧縮なし

1 : 1/2

2 : 1/4

3 : 1/8

<画像の種類>

保存する画像の種類を指定します（文字列）。

AL : 画像バッファにあるすべての画像

NG : 画像バッファにあるNG画像

OK : 画像バッファにあるOK画像

<フォルダ名>

任意のフォルダ名を入力します（文字列）。

戻り値 説明 用例

なし。

画像バッファにあるすべての画像データを、SDカードにビットマップ方式で書き込みます。

```
Dim caoCtrl as Object
```

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

'カメラNo.1の画像バッファにあるすべての画像を1/2に圧縮して保存する。

```
caoCtrl.SaveAllImage 1,1,"AL","ImageFolder"
```


実装変数.GetLibraryList

機能

設定されている辞書番号を一覧で取得します。

- 1 現在の設定No.の辞書番号の一覧を取得する場合
 書式 実装変数.GetLibraryList <取得先>
 引数 <取得先>
 0 : 現在の設定
- 2 SDカードに保存されている辞書番号の一覧を取得する場合
 書式 実装変数.GetLibraryList<取得先>,<辞書番号の設定No.>
 引数 <取得先>
 SDカード番号を指定します（整数）。
 1 : SD1
 2 : SD2
 <辞書番号の設定No.>
 取得する辞書番号の設定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

<辞書番号一覧>

辞書番号が整数型の配列に格納されます。

説明

設定されている辞書番号を一覧で取得します。

用例

```
Dim caoCtrl as Object
Dim vntRet as variant
```

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

```
'現在の設定にある辞書番号の一覧を取得する。
vntRet = caoCtrl. GetLibraryList 0
```

実装変数.ChangePassword

機能

パスワードを変更します。

書式

実装変数.ChangePassword <旧パスワード>,<新パスワード>

引数

<旧パスワード>

旧パスワードを入力します（整数）。

<新パスワード>

新パスワードを指定します（整数）。0000～9999の間で指定してください。

戻り値

なし。

説明

パスワードを変更します。

用例

```
Dim caoCtrl as Object
```

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

パスワードを「0000」から「1111」に変更する。

```
caoCtrl.ChangePassword 0000,1111
```

実装変数.InputPseudoConsole

機能 書式 引数

疑似的にコンソール入力を行うことができます。

実装変数.InputPseudoConsole <コンソール入力コード>

<コンソール入力コード>

コンソール入力コードを指定します（文字列）。

FN	: FUNCTIONボタン
ES	: ESCAPEボタン
TG	: TRGボタン
SC	: SCREENボタン
VI	: VIEWボタン
MN	: MENUボタン
EN	: ENTERボタン
UP	: ENTERボタン上
DN	: ENTERボタン下
LT	: ENTERボタン左
RT	: ENTERボタン右
LU	: ENTERボタン左上
LD	: ENTERボタン左下
RU	: ENTERボタン右上
RD	: ENTERボタン右下
FU	: FUNCTION+ENTER上
FD	: FUNCTION+ENTER下
FL	: FUNCTION+ENTER左
FR	: FUNCTION+ENTER右
FLU	: FUNCTION+ENTERボタン左上
FLD	: FUNCTION+ENTERボタン左下
FRU	: FUNCTION+ENTERボタン右上
FRD	: FUNCTION+ENTERボタン右下
RS	: 運転 / 設定切り換え
FV	: FUNCTION+VIEW
FT	: FUNCTION+TRG
FM	: FUNCTION+MENU
FE	: FUNCTION+ENTER
FS	: FUNCTION+ESCAPE
SL	: SCREEN+ENTER左

SR : SCREEN+ENTER右

戻り値

なし。

説明

疑似的にコンソール入力を行うことができます。

用例

Dim caoCtrl as Object

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

'疑似的にコンソールでFunctionボタンを押す。

```
caoCtrl.InputPseudoConsole "FN"
```

実装変数.SaveStatisticalData

機能

統計解析で保存されているすべての統計データを、カンマ区切りのテキスト形式でSDカードに書き込みます。

書式

実装変数.SaveStatisticalData <SDカードの出力先ディレクトリ>

引数

<SDカードの出力先ディレクトリ>

SDカードの出力先ディレクトリを指定します（文字列）。

戻り値

なし。

説明

統計解析で保存されているすべての統計データを、カンマ区切りのテキスト形式でSDカードに書き込みます。

用例

Dim caoCtrl as Object

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

'統計データをフォルダ「stat」にテキスト形式で書き込む。

```
caoCtrl.SaveStatisticalData "/CV/stat"
```

実装変数.ChangeInspectSetting

機能 書式 引数

指定したSDカードの検査設定No.に、設定を切り換えます。

実装変数.ChangeInspectSetting <SDカード番号>,<検査設定No.>

<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 説明 用例

なし。

指定したSDカードの検査設定No.に、設定を切り換えます。

Dim caoCtrl as Object

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

'SD1の検査設定No.1に設定を切り換える。

caoCtrl.ChangeInspectSetting 1,1

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("CV"," CaoProv.KEYENCE.CV ", "", "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("CV","CaoProv.KEYENCE.CV","", "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.ReadToolParameter

機能 書式 引数

指定したウィンドウの公差の上限値または下限値を取得します。

実装変数.ReadToolParameter (<ウィンドウ>,<公差種別>,<上限 / 下限>)

<ウィンドウ>

ウィンドウを指定します（文字列）。

Wnnn : 計測ウィンドウを指定します。nnnには数字が入ります。W000～W127の間で指定してください。

Cnnn : 演算ウィンドウを指定します。nnnには数字が入ります。C000～C127の間で指定してください。

<公差種別>

公差種別を指定します（文字列）。詳細はキーエンス社のCVシリーズ ユーザーズマニュアルの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（文字列）。

HL : 上限値

LL : 下限値

戻り値

<公差設定値>

引数で指定した公差の設定値です。（倍精度実数）。

説明 用例

指定したウィンドウの公差の上限値または下限値を取得します。

```
Dim caoCtrl as Object
```

```
Dim dblMargin as Double
```

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

'計測ウィンドウ005(パターンサーチに設定済み)のX座標下限値を読み出す。

```
dblMargin = caoCtrl.ReadToolParameter("W005","X","LL")
```


実装変数.ChangeToolParameter

機能 書式 引数

指定したウィンドウの公差の上限値または下限値を設定します。

実装変数.ChangeToolParameter <ウィンドウ>,<公差種別>,<上限 / 下限>,<公差設定値>

<ウィンドウ>

ウィンドウを指定します（文字列）。

Wnnn : 計測ウィンドウを指定します。nnnには数字が入ります。W000～W127の間で指定してください。

Cnnn : 演算ウィンドウを指定します。nnnには数字が入ります。C000～C127の間で指定してください。

<公差種別>

公差種別を指定します（文字列）。詳細はキーエンス社のCVシリーズ ユーザーズマニュアルのDW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（文字列）。

HL : 上限値

LL : 下限値

<公差設定値>

公差設定値を指定します（文字列）。

戻り値 説明 用例

なし。

指定したウィンドウの公差の上限値または下限値を設定します。

```
Dim caoCtrl as Object
```

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

'演算ウィンドウ010の下限値を-142.214に設定する。

```
caoCtrl.ChangeToolParameter "C010","MS","LL","-142.214"
```

実装変数.ReadBinaryData

機能 指定したウィンドウの2値化フィルタについて、上限値と下限値を取得します。

書式 実装変数.ReadBinaryData (<ウィンドウNo.>)

引数 <ウィンドウNo.>

ウィンドウNo.を指定します（整数）。0～127の間のNo.で指定してください。

戻り値 次の2つの値が整数型の配列に格納されます。

<2値化フィルタ上限>

2値化フィルタの上限値です。

<2値化フィルタ下限>

2値化フィルタの下限値です。

説明 指定したウィンドウの2値化フィルタについて、上限値と下限値を取得します。

用例

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

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

'ウィンドウ3の2値化フィルタの上限値、下限値を取得する。

'iaryParams(0)に上限値が

'iaryParams(1)に下限値が格納される。

```
vntRet = caoCtrl.ReadBinaryData(3)
```

```
iaryParams(0) = vntRet(0)
```

```
iaryParams(1) = vntRet(1)
```

実装変数.ChangeBinaryData

機能	指定したウィンドウの2値化フィルタについて、上限値と下限値を設定します。
書式	実装変数.ChangeBinaryData <ウィンドウNo.>,<上限値>,<下限値>
引数	<ウィンドウNo.> ウィンドウNo.を指定します（整数）。0～127の間のNo.で指定してください。 <上限値> 上限値を指定します（整数）。0～255の間の値で指定してください。 <下限値> 下限値を指定します（整数）。0～255の間の値で指定してください。

戻り値	なし。
説明	指定したウィンドウの2値化フィルタについて、上限値と下限値を設定します。
用例	Dim caoCtrl as Object

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

```
'ウィンドウ3の2値化フィルタの上限値を200に、下限値を100に設定する。  
caoCtrl.ChangeBinaryData 3,200,100
```

実装変数.InitCommandMemory

機能	コマンドメモリの初期設定値として、現在のコマンドメモリの値をまとめて反映します。
書式	実装変数.InitCommandMemory
引数	なし。
戻り値	なし。
説明	コマンドメモリの初期設定値として、現在のコマンドメモリの値をまとめて反映します。
用例	Dim caoCtrl as Object

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

```
'コマンドメモリの初期設定値として、現在のコマンドメモリの値をまとめて反映する。  
caoCtrl.InitCommandMemory
```

実装変数.ReadCommandMemory

機能	指定したコマンドメモリのデータを取得します。
書式	実装変数.ReadCommandMemory (<コマンドメモリNo.>)
引数	<コマンドメモリNo.> コマンドメモリNo.を指定します（整数）。0～999の間のNo.で指定してください。
戻り値	<コマンドメモリのデータ> コマンドメモリのデータです（整数）。
説明	指定したコマンドメモリのデータを取得します。
用例	<pre>Dim caoCtrl as Object Dim iParam as Integer caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", "conn=eth:192.168.0.10") 'コマンドメモリ4のデータを取得する。 iParam = caoCtrl.ReadCommandMemory(4)</pre>

実装変数.ChangeCommandMemory

機能	指定した最大32個のコマンドメモリにデータを設定します。
書式	実装変数.ChangeCommandMemory <コマンドメモリNo.>,<データ>, <コマンドメモリNo.>,<データ>.....
引数	次の2つの引数をペアで指定してください。最大32組指定できます。 <コマンドメモリNo.> コマンドメモリNo.を指定します（整数）。0～999の間のNo.で指定してください。 <データ> データを指定します（整数）。-2147483648～2147483647の間の値で指定してください。
戻り値	なし。
説明	指定した最大32個のコマンドメモリにデータを設定します。
用例	<pre>Dim caoCtrl as Object caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", "conn=eth:192.168.0.10") 'コマンドメモリ000の値を1、コマンドメモリ100の値を-1000に設定する。 caoCtrl.ChangeCommandMemory 0,1,100,-1000</pre>

実装変数.RefreshReferencePosition

機能 現在の登録画像で基準値を再計算します。

書式 実装変数.RefreshReferencePosition

引数 なし。

戻り値 なし。

説明 現在の登録画像で基準値を再計算します。

用例 Dim caoCtrl as Object

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

'現在の登録画像で基準値を再計算する。

```
caoCtrl.RefreshReferencePosition
```

実装変数.RefreshReferencePositionAsync

機能 非同期で現在の登録画像で基準値を再計算します。

書式 実装変数.RefreshReferencePositionAsync

引数 なし。

戻り値 なし。

説明 非同期で現在の登録画像で基準値を再計算します。

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("CV"," CaoProv.KEYENCE.CV ", "", "conn=eth:192.168.0.10")
```

'現在の登録画像で基準値を再計算する。

```
caoCtrl.RefreshReferencePositionAsync
```

'RefreshReferencePositionAsyncの戻り値取得

```
vntResult = caoCtrl.GetCommandResult
```

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("CV"," CaoProv.KEYENCE.CV ", "", "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.ChangeShutterSpeed

機能 書式 引数

指定したカメラのシャッタースピードを変更することができます。

実装変数.ChangeShutterSpeed <カメラNo.>, <シャッタースピード>

<カメラNo.>

カメラNo.を指定します（整数）。1～4の間で指定してください。

<シャッタースピード>

シャッタースピードを指定します（整数）。0～10の間で指定してください。

0	: 1/15
1	: 1/30
2	: 1/60
3	: 1/120
4	: 1/240
5	: 1/500
6	: 1/1000
7	: 1/2000
8	: 1/5000
9	: 1/10000
10	: 1/20000

戻り値 説明 用例

なし。

指定したカメラのシャッタースピードを変更することができます。

```
Dim caoCtrl as Object
```

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

'カメラNo.1のシャッタースピードを1/30に変更する。

```
caoCtrl.ChangeShutterSpeed 1,1
```

実装変数.ChangeCameraSensitivity

機能	指定したカメラの感度を変更することができます。
書式	実装変数.ChangeCameraSensitivity <カメラNo.>,<感度>
引数	<p><カメラNo.></p> <p>カメラNo.を指定します（整数）。1～4の間で指定してください。</p> <p><感度></p> <p>カメラの感度を指定します（整数）。10～90の間で指定してください。</p>
戻り値	なし。
説明	指定したカメラの感度を変更することができます。指定した値の1/10の値がカメラ感度として設定されます。
用例	<pre>Dim caoCtrl as Object caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", "conn=eth:192.168.0.10") 'カメラNo.1の感度を1.0に変更する。 caoCtrl.ChangeCameraSensitivity 1,10</pre>

実装変数.ChangeTriggerDelay

機能	トリガ入力に対して、実際に撮像が開始するまでのディレイ時間を設定することができます。
書式	実装変数.ChangeTriggerDelay <カメラNo.>,<ディレイ時間>
引数	<p><カメラNo.></p> <p>カメラNo.を指定します（整数）。1～4の間で指定してください。</p> <p><ディレイ時間></p> <p>ディレイ時間を指定します（整数）。0～999（ms）の間で指定してください。</p>
戻り値	なし。
説明	トリガ入力に対して、実際に撮像が開始するまでのディレイ時間を設定することができます。
用例	<pre>Dim caoCtrl as Object caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", "conn=eth:192.168.0.10") 'カメラNo.1のトリガ入力時のディレイ時間を100msに変更する。 caoCtrl.ChangeTriggerDelay 1,100</pre>

実装変数.ChangeLightIntensityLevel

機能	指定した照明のボリュームを変更することができます。
書式	実装変数.ChangeLightIntensityLevel <照明No.>,<照明ボリューム値>
引数	<照明No.> 照明No.を指定します（整数）。1～8の間で指定してください。 <照明ボリューム値> 照明のボリュームを指定します（整数）。0～255の間で指定してください。

戻り値 なし。

説明 指定した照明のボリュームを変更することができます。

用例 Dim caoCtrl as Object

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

```
'照明No.1の照明のボリュームを50に変更する。  
caoCtrl.ChangeLightIntensityLevel 1,50
```

実装変数.ChangePatternCounter

機能	マルチパターン発光のパターンカウンタを変更し、照明点灯パターンを変更することができます。
書式	実装変数.ChangePatternCounter <カウンタ値>
引数	<カウンタ値> カウンタ値を指定します（整数）。0～3の間で指定してください。

戻り値 なし。

説明 マルチパターン発光のパターンカウンタを変更し、照明点灯パターンを変更することができます。

用例 Dim caoCtrl as Object

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

```
'マルチパターン発光のパターンカウンタを1に変更する。  
caoCtrl.ChangePatternCounter 1
```

実装変数.ExecuteCommand

機能	CVシリーズコマンドをCV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シリーズコマンド書式>)
引数	<CVシリーズコマンド書式> 実行するCVシリーズコマンドを、CVシリーズコマンドの書式で指定します（文字列）。
戻り値	< CVシリーズコマンド実行後の受信データ> CV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シリーズコマンドをCVシリーズコマンドの書式で実行します。 CVシリーズコマンドについては、キーエンス社のCV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	<pre>Dim caoCtrl as Object Dim strRet as String caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", "conn=eth:192.168.0.10") '表示パターンをスルー画面に切り替える。 'コマンドが正常に実行されれば、strRetには"DS"が格納される。 'コマンドが失敗すると、strRetには "ER,DS,nn"（ nnにはエラーコードが入る。 ）が格納される。 strRet = caoCtrl.ExecuteCommand("DS,PT,0")</pre>

実装変数.ExecuteCommandAsync

機能	CVシリーズコマンドを非同期でCV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シリーズコマンド書式>
引数	<CVシリーズコマンド書式> 実行するCVシリーズコマンドを、CVシリーズコマンドの書式で指定します（文字列）。
戻り値	なし
説明	CVシリーズコマンドを非同期でCVシリーズコマンドの書式で実行します。 コマンド実行の成否や戻り値は、 GetCommandResult コマンドで取得します。 CVシリーズコマンドについては、キーエンス社のCV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("CV"," CaoProv.KEYENCE.CV ", "", "conn=eth:192.168.0.10") '表示パターンをスルー画面に切り替える。 'コマンドの戻り値はGetCommandResultコマンドで取得する。 caoCtrl.ExecuteCommandAsync "DS,PT,0" 'ExecuteCommandAsync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： トリガ1

2： トリガ2

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。

トリガをかけた後に結果の受信を行います。出力結果が受信できない場合は、Timeout時間まで待ちます。

出力結果にはトリガの応答パッケージが含まれます。

例) x=10, y=11 の値を出力設定した時

“T1,+11,+12”

用例 Dim caoCtrl as Object

Dim strRet as String

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", "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)

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う場合は、[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("CV", CaoProv.KEYENCE.CV, "", "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("CV", CaoProv.KEYENCE.CV, "", "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	<pre>Dim caoCtrl as Object caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", "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</pre>

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("CV"," CaoProv.KEYENCE.CV ", "", "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	<p>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。</p> <p>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。</p> <p>また、同期コマンドの後で使用了場合は、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。</p> <p>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。</p> <p>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。</p> <p>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。</p>
用例	<pre>Dim caoCtrl as Object Dim vntResult as variant caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", "conn=eth:192.168.0.10") '表示パターンをスルー画面に切り替える。 caoCtrl.ExecuteCommandAsync "DS,PT,0" 'コマンドの戻り値をGetCommandResultコマンドで取得する。 vntResult = caoCtrl.GetCommandResult</pre>

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、[ChangeCurrentUnit](#)コマンドを実行する時に、ウィンドウNo.に127を超えた値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シリーズ ユーザマニュアルのUW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 / 桁数 / 範囲のいずれかが誤っている。"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	CV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 シリーズプロバイダの実装
```

```
caoCtrl = Cao.AddController("CV", "CaoProv.KEYENCE.CV", "", _  
"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 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 シリーズ

バージョン	対応RC8	改訂内容
Ver.1.0.0	Ver.1.8.6	初版
Ver.1.0.1	Ver.1.13.0	Triggerコマンド引数拡張 TriggerAndGetResultコマンドバグ修正 非同期コマンド (ChangeModeAsync, ChangeInspectionSettingAsync, RefreshReferencePositionAsync)追加 GetCommandResultコマンド追加 エラーコード一覧追加
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.2.8.*	同期コマンド (RegistImage, GetRegistImageList, SaveAllImage, GetLibraryList, ChangePassword, InputPseudoConsole, SaveStatisticalData, ChangeShutterSpeed, ChangeCameraSensitivity, ChangeTriggerDelay, ChangeLightIntensityLevel, ChangePatternCounter)追加

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。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。

DENSO Robotics

THIRD PARTY PRODUCTS

株式会社 デンソーウェーブ