

東洋機械金属株式会社
TOYO-MM PLCS12 プロバイダ

Version 1.1.0

ユーザーズ ガイド

August 6, 2021

備考：

【改版履歴】

バージョン	日付	内容
1.0.0	2018-09-11	初版.
1.1.0	2021-08-06	以下の変数を追加 <ul style="list-style-type: none"> ・ 成形条件 ・ 成形運転状態 CaoWorkspace::AddController メソッドのオプションを修正 <ul style="list-style-type: none"> ・ Delay オプションの削除 ・ CmdTimeout オプション, ConnTimeout オプションの追加 その他文言の修正

【動作確認機種】

機種	バージョン	注意事項
J450C	-	[断線時の動作] 断線復旧し, 再接続に成功しても, 機械側が通信コマンドに対して応答しなくなる場合があります. その場合は, 一度切断し, 一定時間経過後に再接続を試みてください. [複数接続対応] 複数クライアントの接続には対応していません.

目次

1. はじめに.....	4
1.1. 本書が想定している環境とバージョン.....	4
1.2. 参考となる情報源.....	4
2. アプリケーション開発のための環境セットアップ.....	6
2.1. PLCS-12 射出成形機とクライアント PC との接続.....	6
2.2. PC 開発環境のセットアップ.....	11
2.2.1. TOYO-MM PLCS12 プロバイダの自動インストール.....	11
2.2.2. TOYO-MM PLCS12 プロバイダの手動インストール.....	11
3. TOYO-MM PLCS12 プロバイダによるプログラミング.....	12
3.1. モニタデータ1を受信するサンプルプログラミング.....	12
3.1.1. サンプルプログラム.....	13
4. コマンドリファレンス.....	19
4.1. メソッド/プロパティ一覧.....	19
4.2. メソッド・プロパティ.....	20
4.2.1. CaoWorkspace クラス.....	20
4.2.2. CaoController クラス.....	22
4.2.3. CaoVariable クラス.....	25
4.3. 変数一覧.....	27
4.3.1. CaoController クラス変数.....	27
4.4. イベント一覧.....	31
5. PLCS12 プロバイダエラーコード.....	32
付録 A. 通信コマンドとの対応表.....	32

1. はじめに

本書は、東洋機械金属株式会社の射出成形機に対してデータの要求、及び通知をするプロバイダのユーザーズガイドです。以降東洋機械金属株式会社の射出成形機を PLCS-12 射出成形機と呼称します。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。以降本プロバイダを TOYO-MM PLCS12 プロバイダと呼称します。

TOYO-MM PLCS12 プロバイダは TCP/IP プロトコルで PLCS-12 射出成形機とデータの送受信を行います。

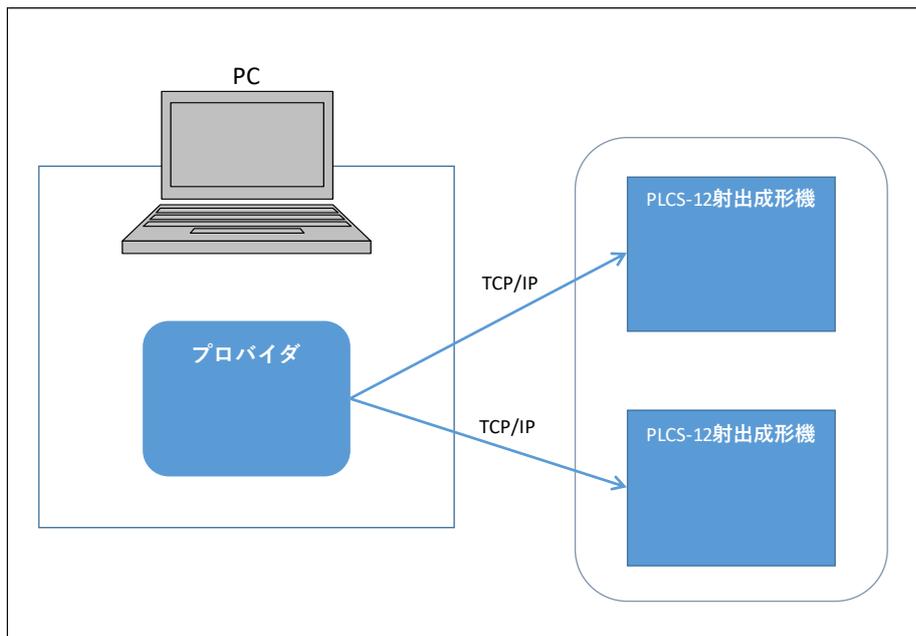


図 1-1 構成図

1.1. 本書が想定している環境とバージョン

クライアント PC が Windows 上で動作し、対象とする PLCS-12 射出成形機がイーサネット接続可能である環境を想定しています。PC の開発環境は、Component Object Model (COM, コンポーネント・オブジェクト・モデル) をサポートするプログラミング環境であれば開発が可能です。

1.2. 参考となる情報源

本書のプログラミング事例は、すべて Visual Basic 6.0 で記載していますが、C++, Java, .NET などさまざまなプログラム言語で開発が可能です。使用方法に関しては、「ORiN2 プログラミングガイド」を参照して下さい。

「ORiN2 プログラミングガイド」は ORiN2 SDK インストールフォルダの以下のファイルに該当します。

- ORiN2¥CAO¥Doc¥ORiN2_ProgrammersGuide_<lang>.pdf

※<lang>の部分は環境毎の言語文字列に置き換えてお読み下さい。

プロバイダを使ったアプリケーションを開発する上で必要となる ORiN2, COM/DCOM の基礎知識や技術に関して例を交えながら解説されています。

TOYO-MM PLCS12 プロバイダは、東洋機械金属株式会社から提供された「PLCS-12 監視ソフト イーサネット通信仕様書（ファイル名：C07058P P-12 イーサネット通信仕様.pdf）」を参考に開発しています。以降この仕様書を PLCS-12 イーサネット通信仕様書と呼称します。

以下参考資料

C07058P P-12 イーサネット通信仕様.pdf

C14365P イーサネット通信仕様.pdf

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. PLCS-12 射出成形機とクライアント PC との接続

図 2-1 は PLCS-12 射出成形機の制御盤の一部です。

PLCS-12 射出成形機とクライアント PC は Ethernet 通信 (TCP プロトコル) にて接続されます。

PC の LAN ポートと PLCS-12 射出成形機の LAN ポート (CMA26) 図 2-2 を LAN ケーブルにて接続します。

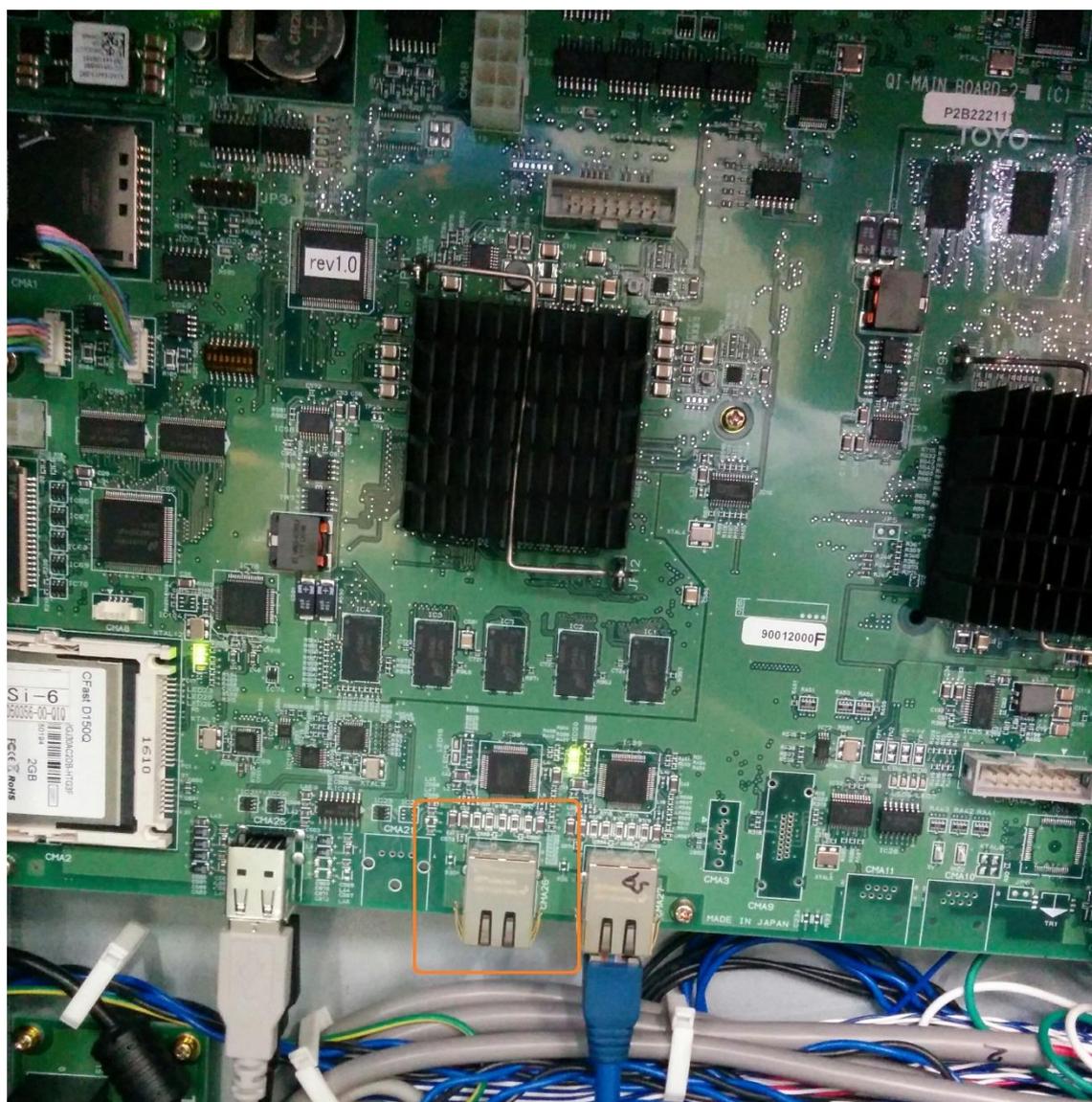


図 2-1 制御盤

LAN ケーブルを接続した後、PLCS-12 射出成形機の画面設定にて IP アドレスを設定します。
MENU1 を押下し MENU1 の画面図 2-3 を表示させます。

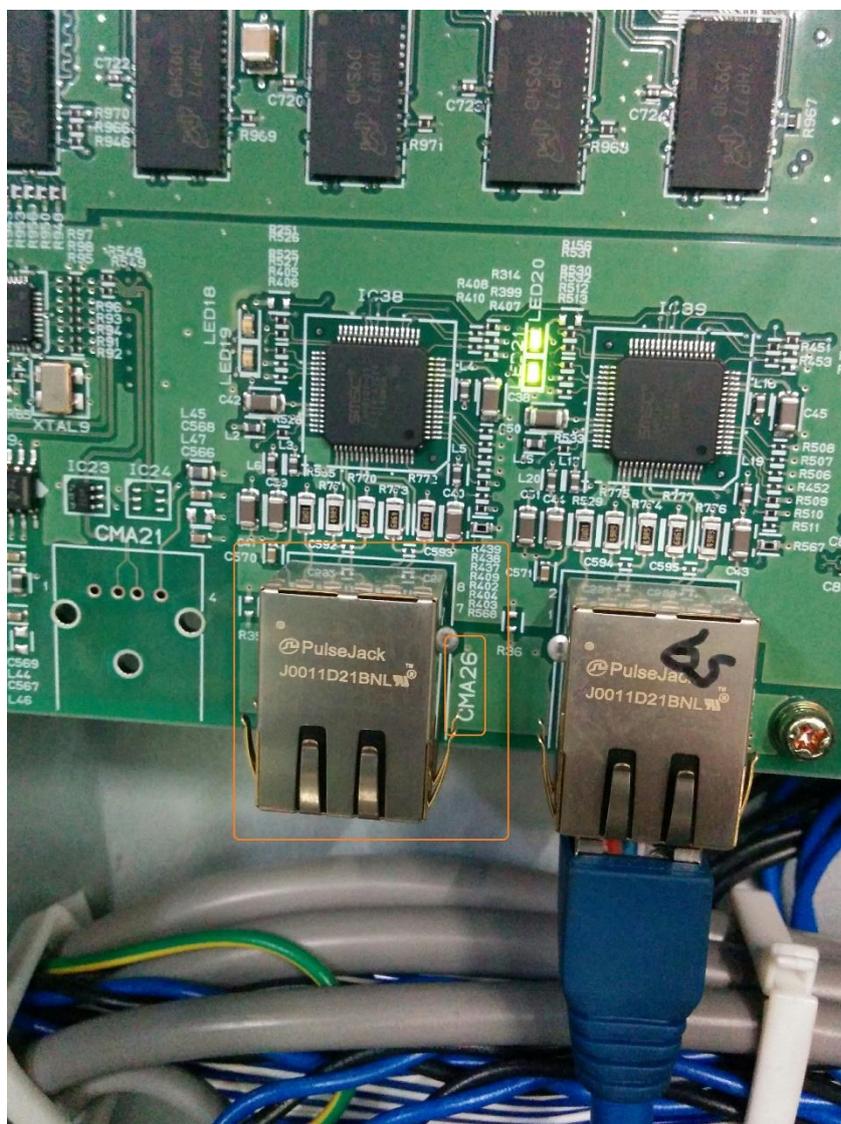


図 2-2 LAN ポート

通信ポート設定を選択して通信ポート設定の画面図 2-4 を表示させます。



図 2-3 MENU1 の画面

通信ポートにて IP アドレスを設定します。設定した IP アドレスとクライアント PC のイーサネットの IP アドレス（左 3 桁）及びサブネットマスクを同じにしてください。（図 2-5 を参照）

No. 102 通信ポート設定 Si-180-6 H450D Φ50 2018-06-29 (金) 10:21:29

スクリー位置	射出速度	射出圧力	回転速度	型開閉位置	エジェクタ位置
0.00 mm	0 mm/s	0.1 R MPa	0 min ⁻¹	350 mm	0.0 mm
サイクルタイム	クッション位置	最小クッション位置	1 次射出時間	可塑化時間	項目選択
0.00 s	3.99 mm	3.99 mm	0.00 s	0.00 s	

プリンタ USB MEMORY 条件印字 COLOR

指定画面印字	プリント		Default		
No.	110	12	74	81	43
No.	112	40	140	21	22

通信ポート1	設定			
IPアドレス	192	168	5	232
サブネットマスク	255	255	255	0
デフォルトゲートウェイ	192	168	5	254

HEATER ON OFF E J 後退限 NZタッチ完了 ログインユーザ pcs

MOTOR ON ショット数 722 稼働回数 17645
ロット数 10000 総稼働時間 240:47:34
製品名 tt 総積算 121.23

RESET

EJ後退 EJ前進 型開 型閉 射出 可塑化 ノズル前進 ノズル後退 型厚後退 型厚前進

図 2-4 通信ポートの画面

PLCS-12 射出成形機と同じにします。

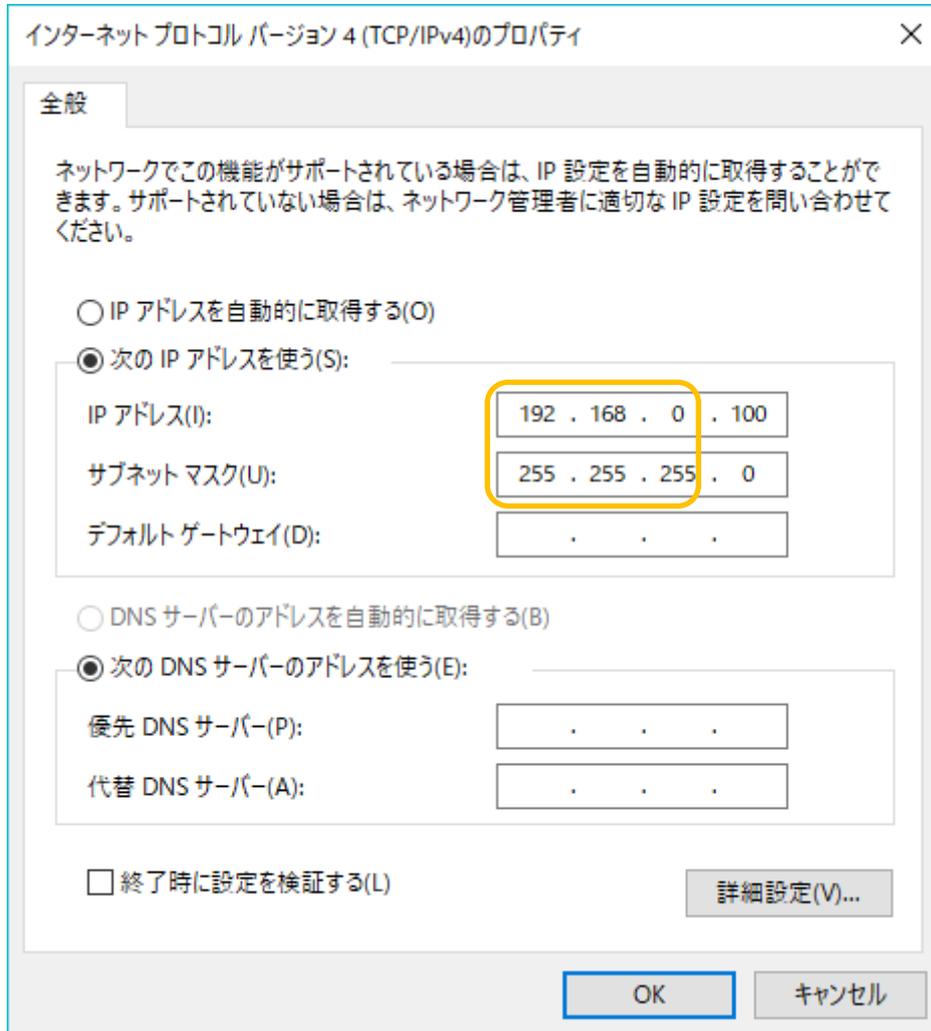


図 2-5 TCP/IPv4 のプロパティ

2. 2. PC 開発環境のセットアップ

2. 2. 1. TOYO-MM PLCS12 プロバイダの自動インストール

ORiN2 SDK がインストールされている環境であれば、PLCS-12 射出成形機に接続するための動作環境の準備は完了です。

開発環境のセットアップは別途、Microsoft Visual Studio 6.0, 2003/2005/2008/2010, LabVIEW など Component Object Model (COM, コンポーネント・オブジェクト・モデル) をサポートする、プログラミング環境をご準備して下さい。

2. 2. 2. TOYO-MM PLCS12 プロバイダの手動インストール

TOYO-MM PLCS12 プロバイダを使用するために手作業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下さい。

また、CAO エンジンが動作するには予め、PC 毎に正規の ORiN2 SDK ライセンスが 1 つ登録されていなくてはなりません。ORiN2 SDK ユーザーズガイド内にある「ライセンスの追加と削除」の節を参照して下さい。

表 2-1 TOYO-MM PLCS12 プロバイダ

ファイル名	CaoProvTOYO-MMPLCS12.dll
ProgID	CaoProv.TOYO-MM.PLCS12
レジストリ登録	regsvr32 CaoProvTOYO-MMPLCS12.dll
レジストリ登録の抹消	regsvr32 /u CaoProvTOYO-MMPLCS12.dll

3. TOYO-MM PLCS12 プロバイダによるプログラミング

TOYO-MM PLCS12 プロバイダでは、以下の手順でクライアント PC と PLCS-12 射出成形機を接続することができます。

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

PLCS-12 射出成形機に接続した後は、CaoController の OnMessage イベントを利用することで、PLCS-12 射出成形機の情報にアクセスすることができます。

3.1. モニタデータ 1 を受信するサンプルプログラミング

ここでは例として PLCS-12 射出成形機からの情報を受信するサンプルプログラムを示します。表 3-1 にサンプルプログラムの要件を、図 3-1 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 3-1 サンプルプログラムの要件

	説明
接続先	接続先 IP アドレスは 192.168.1.201
処理内容	PLCS-12 射出成形機からのモニタデータ 1 を OnMessage で受信する。

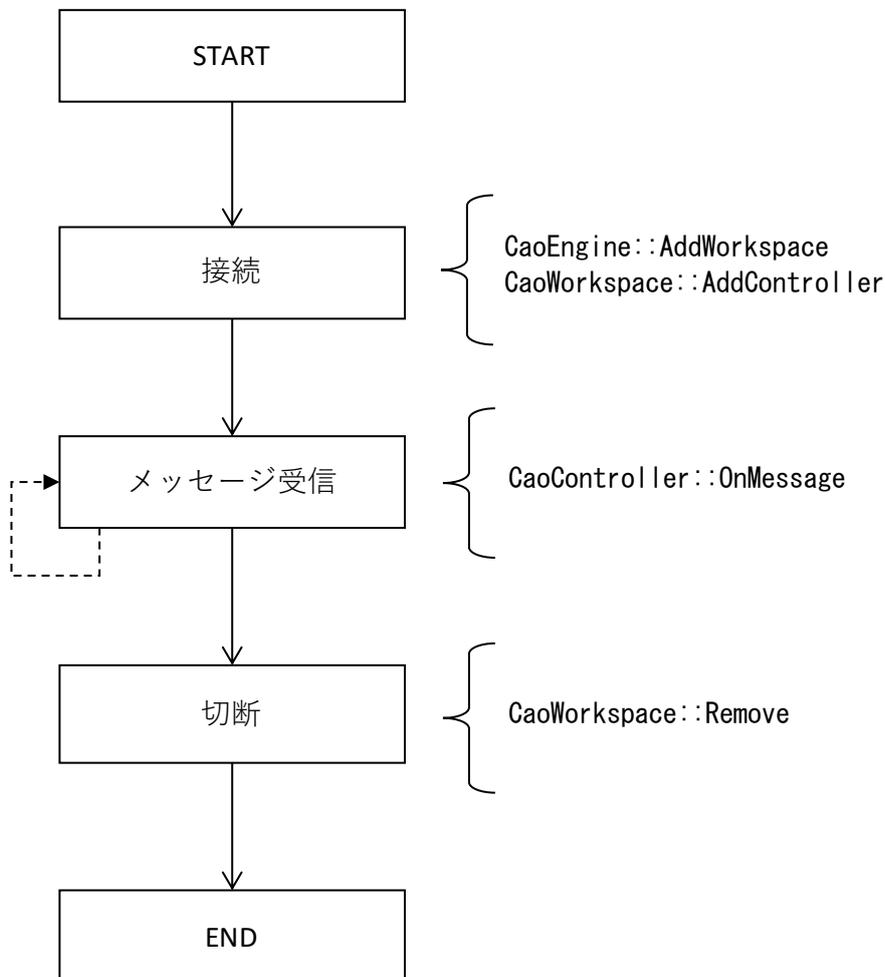


図 3-1 モニタデータ 1 受信の流れ

以降の節から具体的なコードを示します。

3.1.1. 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	ReceiveCheckMonitorData1.vb
<pre> ' オブジェクト Private engine As CaoEngine Private workspace As CaoWorkspace Private WithEvents controller As CaoController ' 接続メソッド Sub Connect () ' CaoEngineオブジェクトの生成 Set engine = New CaoEngine ' CaoWorkspaceオブジェクトの生成 Set workspace = engine.AddWorkspace("NewWrks", "") </pre>	

```
' CaoController オブジェクトの生成
Set controller = workspace.AddController("Controller", _
                                        "CaoProv. TOYO-MM. PLCS12", _
                                        ""', _
                                        "Conn=Tcp:192.168.1.201")

End Sub

' 切断メソッド
Sub Disconnect()
' CaoWorkspace から CaoController を削除
Call workspace.Controllers.Remove(controller.Index)
' CaoController の消去
Set controller = Nothing

' CaoEngine から CaoWorkspace を削除
Call engine.Workspaces.Remove(workspace.Index)
' CaoWorkspace の消去
Set workspace = Nothing

' CaoEngine の消去
Set engine = Nothing
End Sub

' メッセージ受信メソッド
Private Sub controller_OnMessage(ByVal pICaoMess As CAOLib.ICaoMessage)
' 受信メッセージ番号がモニタデータ 1 であればメッセージ本文を表示
If pICaoMess.Number = 33 Then
    Dim moniValues As Variant
    moniValues = pICaoMess.Value

    ' 年月日
    Dim moniDate As String
    moniDate = moniValues(0) & "/" & moniValues(1) & "/" & moniValues(2)
    Call MsgBox(moniDate)

    ' 時分秒
    Dim moniTime As String
    moniTime = moniValues(3) & ":" & moniValues(4) & ":" & moniValues(5)
    Call MsgBox(moniTime)

    ' ショット数
    Dim moniShots As Integer
    moniShots = moniValues(6)
    Call MsgBox(moniShots)

    ' モニタデータ
    For i = 1 To 96 Step 1
```

```
    Dim moniData As Integer
    moniData = moniValues(6 + i)
    Call MsgBox(monData)
Next i

' 不良マーク (0 . . . OK, 1 . . . 不良)
For i = 1 To 96 Step 1
    Dim moniFailure As Integer
    moniFailure = moniValues(102 + i)
    Call MsgBox(monFailure)
Next i
End If
End Sub
```

3.1.1.1. 接続

PLCS-12 射出成形機と接続するためには、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。VB6では、CaoControllerのイベントをハンドルするためにはCaoController型変数を“WithEvents”付きで宣言する必要があります。

```
' CaoEngineオブジェクト用の変数
Private engine As CaoEngine
' CaoWorkspaceオブジェクト用の変数
Private workspace As CaoWorkspace
' CaoControllerオブジェクト用の変数
Private WithEvents controller As CaoController
```

- (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 engine = 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 workspace = engine.AddWorkspace("NewWrks", "")
```

```
' デフォルトの CaoWorkspace オブジェクトの取得
Set workspace = engine.Workspaces.Item(0)
```

- (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。TOYO-MM PLCS12プロバイダでは、接続先のIPアドレス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 controller = workspace.AddController("Controller", _
    "CaoProv. TOYO-MM. PLCS12", _
    "", _
    "Conn=Tcp:192.168.1.201")
```

3.1.1.2. メッセージ受信

PLCS-12 射出成形機の情報 は CaoController::OnMessage イベントをハンドルすることで取得可能となります。サンプルプログラムでは、controller_OnMessage イベントハンドラを用意して CaoController::OnMessage イベントをハンドルしています。各メッセージの仕様については、4.4 イベント一覧を参照してください。

' 受信メッセージ番号がモニターデータ 1 であればメッセージ本文を表示

```
If pICaoMess.Number = 33 Then
    Dim moniValues As Variant
    moniValues = pICaoMess.Value
```

' 年月日

```
Dim moniDate As String
moniDate = moniValues(0) & "/" & moniValues(1) & "/" & moniValues(2)
Call MsgBox(moniDate)
```

' 時分秒

```
Dim moniTime As String
moniTime = moniValues(3) & ":" & moniValues(4) & ":" & moniValues(5)
Call MsgBox(moniTime)
```

' ショット数

```
Dim moniShots As Integer
moniShots = moniValues(6)
Call MsgBox(moniShots)
```

' モニターデータ

```
For i = 1 To 96 Step 1
    Dim moniData As Integer
    moniData = moniValues(6 + i)
    Call MsgBox(moniData)
Next i
```

' 不良マーク (0 . . . OK, 1 . . . 不良)

```
For i = 1 To 96 Step 1
    Dim moniFailure As Integer
    moniFailure = moniValues(102 + i)
    Call MsgBox(monifailure)
Next i
```

```
End If
```

3.1.1.3. 切断

コントローラと切断する場合には、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。

```
' CaoWorkspace から CaoController を削除
Call workspace.Controllers.Remove(controller.Index)
' CaoController の消去
Set controller = Nothing
' CaoEngine から CaoWorkspace を削除
Call engine.Workspaces.Remove(workspace.Index)
' CaoWorkspace の消去
Set workspace = Nothing
' CaoEngine の消去
Set engine = Nothing
```

4. コマンドリファレンス

4.1. メソッド/プロパティ一覧

表 4-1 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. 20
CaoController			
	Index	P コントローラ番号の取得	P. 22
	Name	P コントローラ名の取得	P. 22
	VariableNames	P 接続可能な変数名リストの取得	P. 22
	Variables	P コントローラが保持する変数コレクションの取得	P. 23
	AddVariable	M 変数オブジェクトの追加	P. 24
	OnMessage	E メッセージ受信イベント	P. 24
CaoVariable			
	Index	P 変数番号の取得	P. 25
	Name	P 変数名の取得	P. 25
	Value	P 値の取得/設定	P. 26

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.

4.2. メソッド・プロパティ

4.2.1. CaoWorkspace クラス

4.2.1.1. AddController メソッド

CaoWorkspace に、コントローラオブジェクトを追加します。TOYO-MM PLCS12 プロバイダでは、AddController メソッド実行時に渡されたパラメータを参照し、該当する PLCS-12 射出成形機と接続を行います。以下に、AddController メソッドの仕様を示します。

書式

CaoController AddController

```
(
    "<コントローラ名>",           // コントローラ名(任意)
    "CaoProv. TOYO-MM. PLCS12",   // 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
)
```

使用例

```
Dim engine As CaoEngine           ' Engineオブジェクト
Dim workspace As CaoWorkspace     ' Workspaceオブジェクト
Dim controller As CaoController  ' Controllerオブジェクト

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("Controller", _
                                         "CaoProv. TOYO-MM. PLCS12", _
                                         "", _
                                         "Conn=Tcp:192.168.1.201")
```

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	○	PLCS-12 射出成形機と接続するための TCP 接続オプションを指定します。詳細は 4.2.1.1.1. Conn オプションを参照して下さい。	--	--

オプション	必須	説明	値範囲	デフォルト値
ConnTimeout ²	--	接続/送受信タイムアウトを ms 単位で指定します。	1 - 65534	500
CmdTimeout	--	値取得時のタイムアウトを ms 単位で指定します。 ConnTimeout 以下の指定はできません。	2 - 65535	1000

4.2.1.1.1. 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。ここで中括弧(“[]”)内は省略可能なことを、各パラメータの解説中の下線部はオプションを指定しなかった時のデフォルト値をそれぞれ示します。

確認した PLCS-12 射出成形機ではポート番号の変更は確認出来なかったため、接続先ポートは指定しなくても構いません。

下線はデフォルト値を示します。

“Conn=Tcp:<IP>[:<Port>]”

<IP> : 接続先 IP アドレス。

<Port> : 接続先ポート。 (8200)

² ConnTimeout オプションは、Ver1.0.0 に存在した Timeout オプションの代替オプションです。Ver1.1.0 以降は ConnTimeout オプションの使用を推奨いたします。また、Timeout オプションと ConnTimeout を同時に指定した場合は ConnTimeout が優先されます。

4.2.2. CaoController クラス

4.2.2.1. Index プロパティ

コントローラ番号を Long 型 (4 バイト整数型) で取得します。この番号は、該当のコントローラが、CaoWorkspace クラスの保持するコントローラコレクションの何番目のデータかを示します。

使用例

```
Dim engine As CaoEngine           ' Engineオブジェクト
Dim workspace As CaoWorkspace     ' Workspaceオブジェクト
Dim controller As CaoController   ' Controllerオブジェクト

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("Controller", _
                                         "CaoProv. TOYO-MM. PLCS12", _
                                         "", _
                                         "Conn=Tcp:192.168.1.201")

' Index取得
Dim index as Long
index = controller.Index
```

4.2.2.2. Name プロパティ

CaoWorkspace クラスの AddController メソッドで指定されたコントローラ名を取得します。

使用例

```
Dim engine As CaoEngine           ' Engineオブジェクト
Dim workspace As CaoWorkspace     ' Workspaceオブジェクト
Dim controller As CaoController   ' Controllerオブジェクト

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("Controller", _
                                         "CaoProv. TOYO-MM. PLCS12", _
                                         "", _
                                         "Conn=Tcp:192.168.1.201")

Debug.Print controller.Name
```

4.2.2.3. 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

```
Dim engine As CaoEngine      ' Engineオブジェクト
Dim workspace As CaoWorkspace ' Workspaceオブジェクト
Dim controller As CaoController ' Controllerオブジェクト

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("Controller", _
    "CaoProv. TOYO-MM. PLCS12", _
    "", _
    "Conn=Tcp:192.168.1.201")

' 変数名リスト取得
Dim variables as Variant
variables = controller.VariableNames
```

4.2.2.4. Variables プロパティ

コントローラが保持する、変数コレクションを取得します。

使用例

```
Dim engine As CaoEngine      ' Engineオブジェクト
Dim workspace As CaoWorkspace ' Workspaceオブジェクト
Dim controller As CaoController ' Controllerオブジェクト

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("Controller", _
    "CaoProv. TOYO-MM. PLCS12", _
    "", _
    "Conn=Tcp:192.168.1.201")

' 変数コレクション取得
Dim variables as CaoVariables
Set variables = controller.Variables

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

4.2.2.5. AddVariable メソッド

CaoController に変数オブジェクトを追加します。変数名には 4.3.1CaoController クラス変数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

CaoVariable AddVariable

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

4.2.2.6. OnMessage イベント

コントローラのエラー通知や状態の変化を OnMessage イベントとして受け取ることが可能です。受け取れるイベントについては 4.4. イベント一覧を参照して下さい。

4.2.3. CaoVariable クラス

4.2.3.1. Index プロパティ

変数番号を Long 型 (4 バイト整数型) で取得します。この番号は該当の変数が、CaoController クラスの保持する変数コレクションの、何番目のデータかを示します。

使用例

```
Dim engine As CaoEngine          ' Engineオブジェクト
Dim workspace As CaoWorkspace    ' Workspaceオブジェクト
Dim controller As CaoController  ' Controllerオブジェクト

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("Controller", _
                                         "CaoProv. TOYO-MM. PLCS12", _
                                         "", _
                                         "Conn=Tcp:192.168.1.201")

' 変数の追加
Dim variable As CaoVariable
Set variable = controller.AddVariable("@MAKER_NAME")

' Index取得
Dim index as Long
index = variable.Index
```

4.2.3.2. Name プロパティ

CaoController クラスの AddVariable メソッドで指定された変数名を取得します。

使用例

```
Dim engine As CaoEngine          ' Engineオブジェクト
Dim workspace As CaoWorkspace    ' Workspaceオブジェクト
Dim controller As CaoController  ' Controllerオブジェクト

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("Controller", _
                                         "CaoProv. TOYO-MM. PLCS12", _
                                         "", _
                                         "Conn=Tcp:192.168.1.201")

' 変数の追加
Dim variable As CaoVariable
Set variable = controller.AddVariable("@MAKER_NAME")

Debug.Print variable.Name
```

4.2.3.3. Value プロパティ

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、4.3. 変数一覧を参照して下さい。

4.3. 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。

4.3.1. CaoController クラス変数

変数名	説明	Value		参照
		get	put	
@MAKER_NAME	メーカー名を取得します。	○	-	P. 27
@VERSION	DLL バージョンを取得します。	○	-	P. 27
@LAST_ERROR	最後に発生したエラーを取得/設定します。	○	○	P. 28
@MOLDING_CONDITION	成形条件を取得します。	○	-	P. 29
@MOLDING_OPERATION_STATE	成形運転状態を取得します。	○	-	P. 29

4.3.1.1. @MAKER_NAME

メーカー名の取得をします。

使用例

```
Dim engine As CaoEngine
Dim workspace As CaoWorkspace
Dim controller As CaoController
Set engine = New CaoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set controller = workspace.AddController("Controller", _
    "CaoProv. TOYO-MM. PLCS12", _
    "", _
    "Conn=Tcp:192.168.1.201")

' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@MAKER_NAME")

' 値取得
Dim strVal As String
strVal = var.value
```

データ型

データ型	説明
VT_BSTR	メーカー名を取得します。

4.3.1.2. @VERSION

DLL のバージョンの取得をします。

使用例

```
Dim engine As CaoEngine
```

```

Dim workspace As CaoWorkspace
Dim controller As CaoController
Set engine = New CaoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set controller = workspace.AddController("Controller", _
    "CaoProv. TOYO-MM. PLCS12", _
    "", _
    "Conn=Tcp:192.168.1.201")

' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@VERSION")
' 値取得
Dim value As String
value = var.value

```

データ型

データ型	説明
VT_BSTR	DLL のバージョンを取得します。 *. *.*

4.3.1.3. @LAST_ERROR

最後に発生したエラーを取得/設定します。

使用例

```

Dim engine As CaoEngine
Dim workspace As CaoWorkspace
Dim controller As CaoController
Set engine = New CaoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set controller = workspace.AddController("Controller", _
    "CaoProv. TOYO-MM. PLCS12", _
    "", _
    "Conn=Tcp:192.168.1.201")

' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@LAST_ERROR")
' 値取得
Dim strVal As String
strVal = var.value

' 値設定
var.value = ("FFFFFFFF")
' ゼロクリア
var.value = ("00000000")

```

データ型

データ型	説明
VT_BSTR	最後に発生したエラーを取得します。(8桁の16進数文字列)

4.3.1.4. @MOLDING_CONDITION

成形条件を取得します。

使用例

```
Dim engine As CaoEngine
Dim workspace As CaoWorkspace
Dim controller As CaoController
Set engine = New CaoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
Set controller = workspace.AddController("Controller", _
    "CaoProv. TOYO-MM. PLCS12", _
    "", _
    "Conn=Tcp:192.168.1.201")

' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@MOLDING_CONDITION")
' 値取得
Dim conditions As String()
conditions = var.value
```

データ型

データ型	説明
VT_BSTR VT_ARRAY	成形条件を保持しています。
$0 \leq i \leq 75$	射出設定
$76 \leq i \leq 132$	計量設定
$133 \leq i \leq 163$	温度設定
$164 \leq i \leq 221$	型開閉・EJ 設定
$222 \leq i \leq 250$	型厚・タイムカウンタ
$251 \leq i \leq 284$	単位設定

4.3.1.5. @MOLDING_OPERATION_STATE

成形運転状態を取得します。

使用例

```
Dim engine As CaoEngine
Dim workspace As CaoWorkspace
Dim controller As CaoController
Set engine = New CaoEngine
```

```

Set workspace = engine.AddWorkspace("NewWrks", "")
Set controller = workspace.AddController("Controller", _
    "CaoProv. TOYO-MM. PLCS12", _
    "", _
    "Conn=Tcp:192.168.1.201")
' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@MOLDING_OPERATION_STATE")
' 値取得
Dim states As String()
states = var.value

```

データ型

データ型	説明
VT_BSTR VT_ARRAY	メーカー名を取得します。
0	日付
1	時間
2	機種名
3	運転モードコード
4	運転モード名
5	製品名
6	金型 No.
7	ショット数
8	サイクルタイム
9	稼働時間
10	停止時間
11	不良数
12	不良率
13	生産数
14	生産率
15	稼働率
16	進捗度
17 ~ 31	予備

4.4. イベント一覧

コントローラのエラー通知や状態の変化を OnMessage イベントとして受け取ることが可能です。

受信メッセージは、Number プロパティに通信コマンドの識別子、Value プロパティにその識別子に付随するデータ列を格納しています。また、データ列はデータ列内の区切り文字（, :カンマ）で分解した文字列配列（VT_BSTR | VT_ARRAY）となっています。

識別子、データ列の詳細は PLCS-12 イーサネット通信仕様書を参照して下さい。

以下に受け取るメッセージ番号を示します。

メッセージ番号	説明
30	アラームデータ
32	モードデータ
33	モニタデータ 1
34	モニタデータ 2
36	成形条件
46	成形運転状態
54	設定値来歴データ
55	半固定値来歴データ
999	通信異常

5. PLCS12 プロバイダエラーコード

本プロバイダには、0x8011****でマスクした以下の独自エラーコードが存在します。（表 5-1 独自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下さい。

表 5-1 独自エラーコード表

エラー番号	説明
0x80110001	受信したパケットが不正です。
0x80110002	受信したパケットのチェックサムが不正です。
0x80110003	ミューテックスの作成に失敗しました。

付録A. 通信コマンドとの対応表

変数

変数	通信コマンド識別子
@MOLDING_CONDITION	要求時: 35
	受信: 36
@MOLDING_OPERATION_STATE	要求時: 45
	受信: 46

メッセージ

メッセージ	通信コマンド識別子
アラームデータ	30
モードデータ	32
モニタデータ 1	33
モニタデータ 2	34
成形条件	36
成形運転状態	46
設定値来歴データ	54
半固定値来歴データ	55