

ifm efector 株式会社
I0-Link プロバイダ

Version 1.0.0

ユーザーズ ガイド

November 30, 2020

備考：

【改版履歴】

| バージョン | 日付 | 内容 |
|-------|------------|--------------------------|
| 1.0.0 | 2020-02-29 | 初版. |
| | 2020-11-30 | IODD ファイルについての説明を記述しました. |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

【動作確認機種】

| 機種 | バージョン | 注意事項 |
|--------|----------------------|------|
| AL1342 | AL1x4x_cn_mo_v2.3.20 | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

内容

| | |
|---|----|
| 1. はじめに..... | 5 |
| 1.1. 本書が想定している環境とバージョン..... | 5 |
| 1.2. 参考となる情報源..... | 5 |
| 2. アプリケーション開発のための環境セットアップ..... | 7 |
| 2.1. IO-Link マスターとプロバイダとの接続..... | 7 |
| 2.1.1. IO-Link マスターの電源供給ポートの接続..... | 7 |
| 2.1.2. IO-Link マスターの通信ポートへの接続..... | 7 |
| 2.1.3. IO-Link マスターとIO-Link デバイスとの接続..... | 7 |
| 2.2. PC 開発環境のセットアップ..... | 7 |
| 2.2.1. プロバイダの自動インストール..... | 7 |
| 2.2.2. プロバイダの手動インストール..... | 8 |
| 3. コマンドリファレンス..... | 9 |
| 3.1. メソッド/プロパティ一覧..... | 9 |
| 3.2. メソッド・プロパティ..... | 9 |
| 3.2.1. CaoWorkspace クラス..... | 9 |
| 3.2.2. CaoController クラス..... | 11 |
| 3.2.3. CaoExtension クラス..... | 14 |
| 3.2.4. CaoVariable クラス..... | 15 |
| 3.3. 変数一覧..... | 15 |
| 3.3.1. CaoController 変数..... | 16 |
| 3.3.2. CaoExtension クラス変数..... | 24 |
| 4. プロバイダによるプログラミング..... | 47 |
| 4.1. タワーライトを制御するサンプルプログラミング..... | 47 |
| 4.1.1. サンプルプログラム..... | 47 |
| 5. プロバイダエラーコード..... | 49 |
| 付録 A. IODD データ型と VARIANT データ型の対応表..... | 50 |
| 付録 B. 通信プロトコルコマンド対応表..... | 51 |

| | |
|--------------------------------|----|
| 付録 C. IO-Link マスタからのエラー一覧..... | 53 |
| 付録 D. IODD ファイル..... | 54 |

1. はじめに

本書は、ifm efector 株式会社の IO-Link マスター (AL1342) に接続し、IO-Link マスターの情報の取得および、マスターに接続されている各 IO-Link デバイスのデータの入出力を行うプロバイダのユーザーズガイドになります。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。

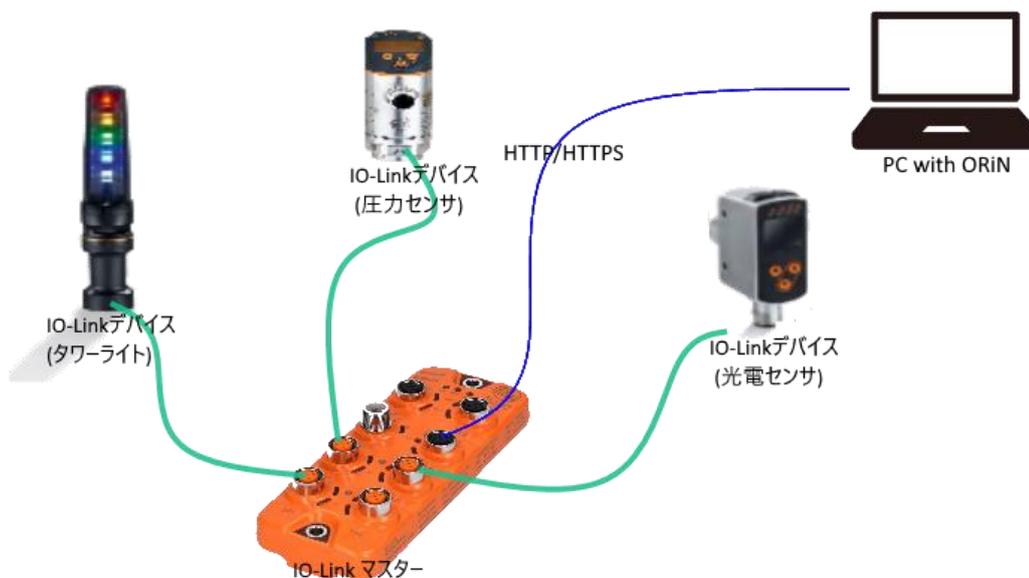


図 1-1 構成図

IO-Link マスターとプロバイダは HTTP/HTTPS 通信により通信を行います。プロバイダは IO-Link デバイスの情報を取得/制御する場合でも IO-Link マスターを介してデータのやり取りを行います。それにより、IO-Link デバイスの製造元を考慮せずに接続することが可能となります。なお、本プロバイダでは IODD ファイルバージョン 1.1 に対応している IO-Link デバイスのみを対象としていますのでご注意ください。

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 CNC がイーサネット接続可能な会社名製デバイス種類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」を参照してください。

「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_</ang>.pdf

※<lang>の部分は環境毎の言語文字列に置き換えてお読みください。

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。

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. I0-Link マスターとプロバイダとの接続

0章で記述した通り、I0-Link マスタとプロバイダとは HTTP/HTTPS 通信にてデータの送受信を行っています。ここでは I0-Link マスター(AL1342)とプロバイダがインストールされたクライアント PC との接続方法を記述します。

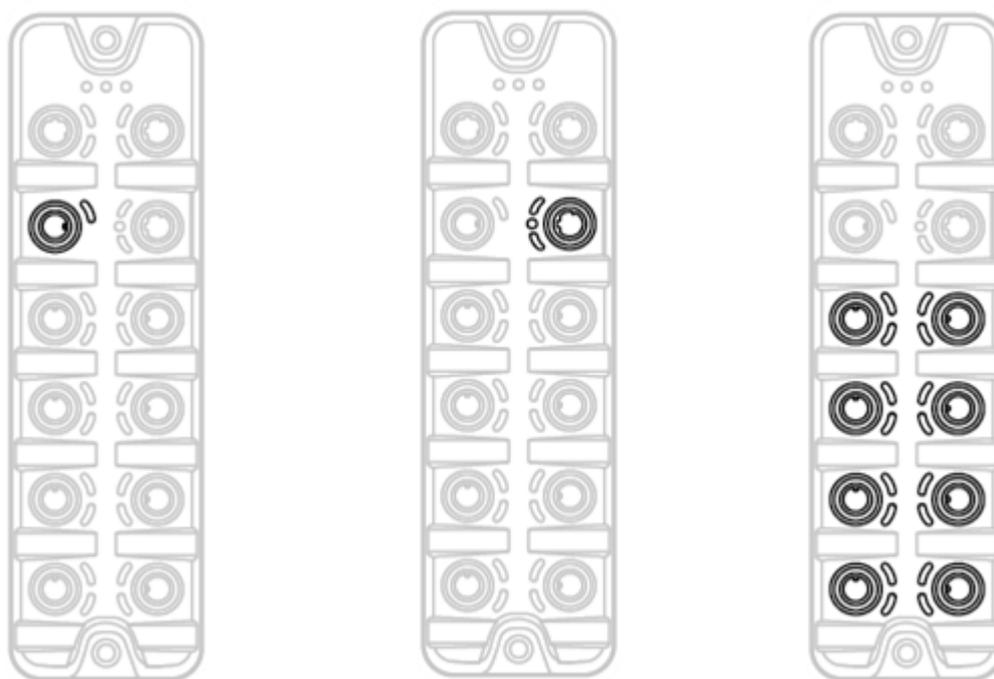


図 2-1 左から順に電源ポート、IoT ポート、I0-Link ポートを示しています。

2.1.1. I0-Link マスターの電源供給ポートの接続

電源供給ポートに電源コネクタケーブルを接続してください。

2.1.2. I0-Link マスターの通信ポートへの接続

IoT ポートに通信コネクタケーブルを接続し、反対側の端子をクライアント PC と接続しているスイッチングハブの空いているポート、もしくはクライアント PC の Lan ポートに直接接続して下さい。

2.1.3. I0-Link マスターと I0-Link デバイスとの接続

I0-Link ポートに接続したい I0-Link デバイスが接続されている I0-Link ケーブルを接続指定ください。

2.2. PC 開発環境のセットアップ

2.2.1. 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、接続するための動作環境（ラインタイム）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ください。

2.2.2. プロバイダの手動インストール

プロバイダを手動でインストールする場合はレジストリ登録が必要です。

レジストリ登録

レジストリ登録を行う場合には、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。

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」の節を参照してください。

表 2-1 プロバイダ情報

| | |
|------------|-----------------------------------|
| ファイル名 | CaoProvifmIO-Link.dll |
| ProgID | CaoProv.ifm.IO-Link |
| レジストリ登録 | regsvr32 CaoProvifmIO-Link.dll |
| レジストリ登録の抹消 | regsvr32 /u CaoProvifmIO-Link.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 | I0-Link マスターへの接続情報の設定 | P. 9 |
| CaoController | | | | |
| | AddExtension | M | I0-Link デバイスの設定 | P. 11 |
| | Extensions | P | コントローラが保持する拡張ボードコレクションの取得 | P. 13 |
| | VariableNames | P | 接続可能な変数名リストの取得 | P. 13 |
| | AddVariable | M | 変数オブジェクトの追加 | P. 13 |
| | 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. 14 |
| CaoExtension | | | | |
| | AddVariable | M | 変数オブジェクトの追加 | P. 14 |
| | VariableNames | P | 接続可能な変数名リストの取得 | P. 14 |
| | Variables | P | タスクが保持する変数コレクションの取得 | P. 15 |
| CaoVariable | | | | |
| | Value | P | 値の取得/設定 | P. 15 |

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

CaoWorkspace にコントローラオブジェクトを追加します。プロバイダはここで指定された接続先情報、通信方式で対応する I0-Link マスタと接続します。以下に、AddController メソッドの仕様を示します。

プロバイダは AddController 時に I0-Link マスターとの接続確認は行いません。そのため、AddController に成功しても I0-Link マスターとの接続が保証されるわけではありません。I0-Link マスターと実際に接続、通信するのは CaoController 変数の Value プロパティにアクセスしたときとなりますのでご注意ください。

書式

CaoController AddController

(

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。

```

"<コントローラ名>", // コントローラ名(任意)
"CaoProv. ifm. I0-Link", // 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(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。なお、表中の値範囲列で下線が引かれている項目はそのオプションを省略した際のデフォルト値を意味します。

| オプション | 必須 | 説明 | 値範囲 |
|----------|----|---|--|
| Server | ○ | 接続する I0-Link マスターの接続先名を指定してください。 | --- |
| SSL | - | SSL 通信を行うかどうかを 2 値で指定してください。 | <u>False: HTTP 接続</u> True:HTTPS 接続 |
| User | - | HTTPS 接続する際のユーザ名を指定してください。 SSL に True を指定する場合のみ必要なオプションです。 SSL に False を指定した場合は、本オプションは無視されます。 省略した場合、" administrator" を使用します。 | 任意の文字列 |
| Password | - | HTTPS 接続する際のパスワードを指定してください。 SSL に True を指定する場合のみ必要なオプションです。 SSL に False を指定した場合は、本オプションは無視されます。 省略した場合、" password" を使用します。 | 任意の文字列 |
| Timeout | - | 接続タイムアウト(ms)を指定してください。 | 1 - デフォルト:500ms |

使用例

以降の使用例は、下のコードが存在することが前提に記述されています。

```

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)

```

' 接続メソッド

```

Private Sub Connect()
' HTTP接続の場合
Set controller = workspace.AddController("Master1", _
" CaoProv. ifm. I0-Link", _
"" , _

```

```
"Server=192.168.0.1")
```

```
' HTTPS接続の場合
```

```
Set controller = workspace.AddController("Master1", _
    "CaoProv.ifm.I0-Link", _
    "", _
    "Server=192.168.0.1, SSL=True, User=usr, Password=pass")
```

```
End Sub
```

```
' 切断メソッド
```

```
Private 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
```

3.2.1.2. セキュリティモード On 時の接続について

接続する I0-Link マスターのセキュリティーモードが On の場合は SSL オプションに True を指定し、UserId オプションと Password オプションにそれぞれユーザー ID とパスワードを指定してください。

I0-Link マスターのセキュリティーモードを Off から On に切り替えた際は、ifm electronic 製の LR DEVICE アプリケーションを利用してユーザ ID とパスワードをはじめに設定しておく必要があります。

3.2.2. CaoController クラス

3.2.2.1. AddExtension メソッド

I0-Link マスターに接続する(もしくは接続している)I0-Link デバイスをプロバイダに設定します。以下に、AddExtension メソッドの仕様を示します。

プロバイダは AddExtension 時にデバイスとの接続確認、およびポート番号の存在有無の確認を行いません。そのため、AddExtension に成功しても I0-Link デバイスとの接続が保証されるわけではありません。実際の接続および通信は GaoExtension 変数の Value プロパティにアクセスする際に行われます。



CaoExtension AddExtension

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

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 |
|-------|----|---|
| Port | ○ | I0-Link デバイスが接続されているポート番号を指定します。I0-Link マスターに存在するポート番号を指定してください。 |
| IODD | - | I0-Link デバイスに対応する IODD ファイルを指定します。IODD ファイルの指定については、P. 12 を参照してください。 本オプションは省略することが可能です。省略した場合は、「3.3.2 CaoExtension クラス変数」のプロセスデータ、I0-Link デバイス設定情報は取得できません。 |

使用例**' 接続**

```
Call Connect
```

' I0-Linkデバイスの追加

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")
```

```
Call controller.Extensions.Remove(extension.Index)
Set caoExt = Nothing
```

' 切断

```
Call Disconnect
```

3.2.2.1.1. IODD ファイルの指定方法**パスの指定方法**

IODD ファイルの指定は相対パスもしくは絶対パスの指定が行えます。IODD ファイル自体については、付録 D を参照してください。

- 相対パス指定
相対パスで指定した場合、下記ディレクトリを基準として IODD ファイルを読込みます。

基準パス

"プロバイダのインストールディレクトリ"/XML

- 絶対パス指定
読込対象の IODD ファイルの絶対パスを指定してください。

3.2.2.2. Extensions プロパティ

コントローラが保持する, CaoExtension のコレクションを取得します。

使用例

' 接続

Call Connect

' 拡張ボードコレクション取得

Dim extensions as CaoExtensions

Set extensions = controller.Extensions

' 拡張ボード取得

Dim extension as CaoExtension

Set extension = extensions.Item(0)

' 切断

Call Disconnect

3.2.2.3. VariableNames プロパティ

接続可能な変数名リストを取得します。取得した変数名は, 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

' 接続

Call Connect

' ファイル名リスト取得

Dim variables as Variant

variables = controller.VariableNames

' 切断

Call Disconnect

3.2.2.4. 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「3.3.1 CaoController 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)
)
```

3.2.2.5. 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**接続**

```
Call Connect
```

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)
```

切断

```
Call Disconnect
```

3.2.3. CaoExtension クラス**3.2.3.1. 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 3.3.2 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)
)
```

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。取得した変数名は 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

’ 接続

```
Call Connect
```

’ 拡張ボードの追加

```
Dim caoExt As CaoExtension  
Set caoExt = controller.AddExtension("Snsor1", _  
                                     "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")
```

’ 変数名リスト取得

```
Dim variables as Variant  
variables = caoExt.VariableNames
```

3.2.3.3. Variables プロパティ

拡張ボードが保持する、変数コレクションを取得します。

使用例

’ 接続

```
Call Connect
```

’ 拡張ボードの追加

```
Dim caoExt As CaoExtension  
Set caoExt = controller.AddExtension("Snsor1", _  
                                     "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")
```

’ 変数コレクション取得

```
Dim variables as CaoVariables  
Set variables = caoExt.Variables
```

’ 変数取得

```
Dim variable as CaoVariable  
Set variable = variables.Item(0)
```

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

接続した IO-Link マスターのポート、もしくはそのポートに接続された IO-Link デバイスからデータを取得/設定します。変数名によって動作が異なります。詳細は、3.3 を参照してください。

3.3. 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。

3.3.1. CaoController 変数

表 3-2 CaoController 変数一覧

| 変数名 | 説明 | 参照 |
|-------------------------|---------------------------------|-------|
| I0-Link マスター基本情報 | | |
| @ProductCode | 品番号を取得します。 | P. 16 |
| @Vendor | 製造元を取得します。 | P. 17 |
| @DeviceFamily | デバイスファミリを取得します。 | P. 17 |
| @HWRevision | ハードウェアバージョンを取得します。 | P. 18 |
| @SerialNumber | シリアル番号を取得します。 | P. 18 |
| @SWRevision | ファームウェアバージョンを取得します。 | P. 18 |
| @BootLoaderRevision | ブートローダーバージョンを取得します。 | P. 19 |
| @ExtensionRevisions | ファームウェアとブートローダーバージョンを取得します。 | P. 19 |
| @FieldbusType | フィールドバス種別を取得します。 | P. 20 |
| @IOLinkMasterName | I0-Link マスター名称を取得/設定します。 | P. 20 |
| I0-Link マスター制御情報 | | |
| @IOLinkMasterVersion | I0-Link マスターのソフトウェアバージョンを取得します。 | P. 20 |
| @IOLinkMasterSWType | I0-Link マスターのソフトウェアタイプを取得します。 | P. 21 |
| @ContainerMaxSize | コンテナエリアの最大サイズ(byte)を取得します。 | P. 21 |
| @ContainerChunkSize | データセグメントのサイズ(byte)を取得します。 | P. 22 |
| @ContainerSize | コンテナエリアのサイズ(byte)を取得します。 | P. 22 |
| 診断データ | | |
| @Temperature | I0-Link マスターの温度(摂氏)を取得します。 | P. 22 |
| @Voltage | I0-Link マスターの電圧(V)を取得します。 | P. 23 |
| @Current | I0-Link マスターの電流(A)を取得します。 | P. 23 |
| @SuperVisionStatus | デバイス供給の診断情報を取得します。 | P. 24 |
| プロバイダ情報 | | |
| @Version | プロバイダのバージョンを取得します。 | P. 24 |

3.3.1.1. I0-Link マスター基本情報

3.3.1.1.1. @ProductCode

品番号を文字列で取得をします。

データ型

型説明

| | |
|---------|-----|
| VT_BSTR | 品番号 |
|---------|-----|

使用例

' 接続

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@ProductCode")

' 値取得

Dim strVal As String

strVal = var.Value

3.3.1.1.2. @Vendor

製造元を文字列で取得をします。

データ型

| 型説明 | |
|---------|-----|
| VT_BSTR | 製造元 |

使用例

' 接続

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@Vendor")

' 値取得

Dim strVal As String

strVal = var.Value

3.3.1.1.3. @DeviceFamily

デバイスファミリを文字列で取得します。

データ型

| 型説明 | |
|---------|----------|
| VT_BSTR | デバイスファミリ |

使用例

' 接続

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@DeviceFamily")

' 値取得

```
Dim strVal As String
strVal = var.Value
```

3.3.1.1.4. @HWRevision

ハードウェアバージョンを文字列で取得します。

データ型

| 型説明 | |
|---------|-------------|
| VT_BSTR | ハードウェアバージョン |

使用例

```
' 接続
Call Connect
' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@HWRevision")
' 値取得
Dim strVal As String
strVal = var.Value
```

3.3.1.1.5. @SerialNumber

シリアル番号を文字列で取得します。

データ型

| 型説明 | |
|---------|----------|
| VT_BSTR | デバイスファミリ |

使用例

```
' 接続
Call Connect
' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@SerialNumber")
' 値取得
Dim strVal As String
strVal = var.Value
```

3.3.1.1.6. @SWRevision

ファームウェアバージョンを文字列で取得します。

データ型

| 型説明 | |
|---------|--------------|
| VT_BSTR | ファームウェアバージョン |

使用例**' 接続**

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@SWRevision")

' 値取得

Dim strVal As String

strVal = var.Value

3.3.1.1.7. @BootLoaderRevision

ブートローダーバージョンを文字列で取得します。

データ型

| 型説明 | |
|---------|--------------|
| VT_BSTR | ブートローダーバージョン |

使用例**' 接続**

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@BootLoaderRevision")

' 値取得

Dim strVal As String

strVal = var.Value

3.3.1.1.8. @ExtensionRevisions

ファームウェアとブートローダーバージョンを文字列で取得します。

データ型

| 型説明 | |
|---------|----------------------|
| VT_BSTR | ファームウェアとブートローダーバージョン |

使用例**' 接続**

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@ExtensionRevisions")

' 値取得

Dim strVal As String

```
strVal = var.Value
```

3.3.1.1.9. @FieldbusType

フィールドバス種別を文字列で取得します。

データ型

| 型説明 | |
|---------|-----------|
| VT_BSTR | フィールドバス種別 |

使用例

```
' 接続
Call Connect
' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@FieldbusType")
' 値取得
Dim strVal As String
strVal = var.Value
```

3.3.1.1.10. @IOLinkMasterName

I0-Link マスター名を取得/設定します。

データ型

| 型説明 | |
|---------|---------------|
| VT_BSTR | I0-Link マスター名 |

使用例

```
' 接続
Call Connect
' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@IOLinkMasterName")
' 値取得
Dim strVal As String
strVal = var.Value
```

3.3.1.2. I0-Link マスター制御情報

3.3.1.2.1. a@IOLinkMasterVersion

I0-Link マスターのソフトウェアバージョンを文字列で取得します。

データ型

| 型説明 | |
|---------|--------------------------|
| VT_BSTR | I0-Link マスターのソフトウェアバージョン |

使用例**' 接続**

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@IOLinkMasterVersion")

' 値取得

Dim strVal As String

strVal = var.Value

3.3.1.2.2. @IOLinkMasterSWType

IO-Link マスターのソフトウェアタイプを文字列で取得します。

データ型

| 型説明 | |
|---------|------------------------|
| VT_BSTR | IO-Link マスターのソフトウェアタイプ |

使用例**' 接続**

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@IOLinkMasterSWType")

' 値取得

Dim strVal As String

strVal = var.Value

3.3.1.2.3. @ContainerMaxSize

コンテナエリアの最大サイズ(byte)を取得します。

データ型

| 型説明 | |
|-------|---------------|
| VT_I4 | コンテナエリアの最大サイズ |

使用例**' 接続**

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@ContainerMaxSize")

' 値取得

Dim val As Long

val = var.Value

3.3.1.2.4. @ContainerChunkSize

データセグメントのサイズ(byte)を取得します。

データ型

| 型説明 | |
|-------|--------------------|
| VT_I4 | データセグメントのサイズ(byte) |

使用例

```
' 接続
Call Connect
' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@ContainerMaxSize")
' 値取得
Dim val As Long
val = var.Value
```

3.3.1.2.5. @ContainerSize

コンテナエリアのサイズ(byte)を取得します。

データ型

| 型説明 | |
|-------|-------------|
| VT_I4 | コンテナエリアのサイズ |

使用例

```
' 接続
Call Connect
' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@ContainerSize")
' 値取得
Dim val As Long
val = var.Value
```

3.3.1.3. 診断データ

3.3.1.3.1. @Temperatue

I0-Link マスターの温度(°C)を取得します。

データ型

| 型説明 | |
|-------|---------------------|
| VT_I4 | I0-Link マスターの温度(°C) |

使用例**' 接続**

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@Temperature")

' 値取得

Dim val As Long

val = var.Value

3.3.1.3.2. @Voltage

IO-Link マスターの電圧 (V) を取得します。

データ型

| 型説明 | |
|-------|--------------------------|
| VT_I4 | IO-Link マスターの電圧 (V) のサイズ |

使用例**' 接続**

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@Voltage")

' 値取得

Dim val As Long

val = var.Value

3.3.1.3.3. @Current

IO-Link マスターの電流 (A) を取得します。

データ型

| 型説明 | |
|-------|---------------------|
| VT_I4 | IO-Link マスターの電流 (A) |

使用例**' 接続**

Call Connect

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@Current")

' 値取得

Dim val As Long

```
val = var.Value
```

3.3.1.3.4. @SuperVisionStatus

デバイス供給の診断情報を取得します。

データ型

| 型説明 | |
|-------|-------------|
| VT_I4 | デバイス供給の診断情報 |

使用例

```
' 接続
Call Connect
' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@SuperVisionStatus")
' 値取得
Dim val As Long
val = var.Value
```

3.3.1.4. プロバイダ情報

3.3.1.4.1. @VERSION

DLL のバージョンの取得をします。

データ型

| 型説明 | |
|---------|-----------------------------|
| VT_BSTR | DLL のバージョンを取得します。 *. *.* |

使用例

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

3.3.2. CaoExtension クラス変数

表 3-3 CaoExtension クラス変数一覧

| 変数名 | 説明 | 参照 |
|-----|----|----|
|-----|----|----|

| I0-Link ポートの設定 | | |
|------------------------|---|-------|
| @SendProcDataPort | LR_SMARTOBSERVER へプロセスデータを送信する際に使用するポート番号を取得/設定します。 | P. 26 |
| @TransCycleTimePreset | I0-Link ポートでのデータ転送のサイクル時間（マイクロ秒）を取得/設定します。 | P. 26 |
| @TransCycleTimeAct | ポート上の I0-Link マスターと I0-Link デバイス間のデータ転送の現在のサイクル時間を取得します。 | P. 27 |
| @Mode | I0-Link ポートの動作モードを取得/設定します。 | P. 28 |
| @ComSpeed | I0-Link ポートのデータ転送速度を取得します。 | P. 28 |
| @ValidationDataStorage | 新しい I0-Link デバイスが接続されたときの検証方法を取得/設定します。 | P. 28 |
| @VaridationVendorId | 検証する VendorID を取得/設定します。 | P. 30 |
| @VaridationDeviceId | 検証する DeviceID を取得/設定します。 | P. 30 |
| @DataStorageMaxSize | データストレージエリアの最大サイズ (byte) を取得します。 | P. 30 |
| @DataStorageChunkSize | データセグメントのサイズ (byte) を取得します。 | P. 30 |
| @DataStorageSize | データストレージエリアのサイズ (byte) を取得します。 | P. 30 |
| プロセスデータ | | |
| @Pin2Input | I0-Link ポートのピン 2 のデジタル入力値を取得します。 | P. 31 |
| @ProcessDataIn | I0-Link ポートのピン 4 の I0-Link 入力値 (入力プロセスデータ) を取得します。 | P. 31 |
| @ProcessDataOut | I0-Link ポートのピン 4 の I0-Link 出力値 (出力プロセスデータ) を取得/設定します。 | P. 34 |
| I0-Link デバイス設定 | | |
| <I0DD ファイル変数> | I0-Link デバイス設定情報を取得/設定します。I0DD ファイルの SpecialistRoleMenuSet に定義されている変数を指定することができます。変数名については、「3.2.3.2 VariableNames プロパティ」を使用して得ることができます。 | P. 38 |
| I0-Link デバイス情報 | | |
| @Status | I0-Link デバイスのステータスを取得します。 | P. 42 |
| @VendorId | I0-Link デバイスの製造元 ID を取得します。 | P. 43 |
| @DeviceId | I0-Link デバイスのデバイス ID を取得します。 | P. 44 |
| @ProductName | I0-Link デバイスの品名称を取得します。 | P. 44 |

| | | |
|-----------------|----------------------------|-------|
| @Serial | I0-Link デバイスのシリアル番号を取得します。 | P. 45 |
| @ApplicationTag | アプリケーションタグを取得/設定します。 | P. 45 |

3.3.2.1. I0-Link ポートの設定

3.3.2.1.1. @SendProcDataPort

LR_SMARTOBSERVER へプロセスデータを送信する際に使用するポート番号を取得/設定します。Modbus TCP plc への接続が中断された場合のみ、パラメータを編集可能です。値の設定が行えるのは I0-Link デバイスが接続されていないポートに制限されます。

データ型

| 型説明 | |
|-------|---|
| VT_I4 | LR_SMARTOBSERVER へプロセスデータを送信する際に使用するポート番号 値範囲: 0 - 65525 |

使用例

' 接続

```
Call Connect
```

' Extension の追加

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD=ifm-00042A-20180717-I0DD1.1.xml")
```

' 変数追加

```
Dim var As CaoVariable
Set var = extension.AddVariable("@SendProcDataPort")
```

' 値取得

```
Dim value As Long
value = var.Value
```

' 値設定

```
var.Value = 63234
```

3.3.2.1.2. @TransCycleTimePreset

I0-Link ポートでのデータ転送のサイクル時間（マイクロ秒）を取得/設定します。値の設定が行えるのは I0-Link デバイスが接続されていないポートに制限されます。

データ型

| 型説明 | |
|-------|--|
| VT_I4 | I0-Link ポートでのデータ転送のサイクル時間（マイクロ秒） 値範囲: 0 - 132800 (0 は最速のサイクル時間) |

使用例**' 接続**

```
Call Connect
```

' Extension の追加

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD=ifm-00042A-20180717-I0DD1.1.xml")
```

' 変数追加

```
Dim var As CaoVariable
Set var = extension.AddVariable("@TransCycleTimePreset")
```

' 値取得/設定

```
Dim value As Long
value = var.Value
if (value <> 0) Then
    var.Value = 0
EndIf
```

3.3.2.1.3. @TransCycleTimeAct

I0-Link マスターと I0-Link デバイス間のデータ転送の現在のサイクル時間（マイクロ秒）を取得します。

データ型

| 型説明 | |
|-------|---|
| VT_I4 | I0-Link マスターと I0-Link デバイス間のデータ転送の現在のサイクル時間（マイクロ秒） 値範囲: 1 - 132800 |

使用例**' 接続**

```
Call Connect
```

' Extension の追加

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD=ifm-00042A-20180717-I0DD1.1.xml")
```

' 変数追加

```
Dim var As CaoVariable
Set var = extension.AddVariable("@TransCycleTimeAct")
```

' 値取得/設定

Dim value As Long
value = var.Value

3.3.2.1.4. @Mode

I/O-Link ポートの動作モードを取得/設定します。Modbus TCP plc への接続が中断された場合のみ、パラメータを編集可能です。

データ型

| 型説明 | |
|-------|--|
| VT_I4 | 動作モード。 0: deactivated 1: Digital input (DI) 2: Digital output (DO) 3: I/O-Link |

3.3.2.1.5. @ComSpeed

I/O-Link ポートのデータ転送速度を取得します。

データ型

| 型説明 | |
|-------|----------------------|
| VT_I4 | I/O-Link ポートのデータ転送速度 |

3.3.2.1.6. @ValidationDataStorage

新しい I/O-Link デバイスが接続されたときの検証方法を取得/設定します。Modbus TCP plc への接続が中断された場合のみ、パラメータを編集可能です。以下に変数値

データ型

| 型説明 | |
|-------|--------------------------|
| VT_I4 | 検証方法。 詳細は別表を参照してください。 |

| @ValidationDataStorage の値 | I/O-Link デバイスの検証 | パラメータ値の保存 | パラメータ値のリカバリ |
|---------------------------|--|-----------|-------------|
| 0 | 検証しません。 | - | - |
| 1 | I/O-Link Ver1.0 に準拠しているかを検証します。 | - | - |
| 2 | I/O-Link Ver1.1 に準拠しているかを検証します。 併せて@ValidationVendorId, | - | - |

| | | | |
|---|---|--|---|
| | @VaridationDeviceId で指定されているベンダーID, デバイスIDを持つIO-Linkデバイスかも検証します. | | |
| 3 | IODD Ver1.1 に準拠しているかを検証します. 併せて@VaridationVendorId, @VaridationDeviceId で指定されているベンダーID, デバイスIDを持つIO-Linkデバイスかも検証します. | パラメータ値は自動で保存されます. 現在のパラメータ値の変更が保存されます. | 同一のIO-Linkデバイスを接続する場合, 工場出荷時の設定でリカバリされます. |
| 4 | IODD Ver1.1 に準拠しているかを検証します. 併せて@VaridationVendorId, @VaridationDeviceId で指定されているベンダーID, デバイスIDを持つIO-Linkデバイスかも検証します. | パラメータ値は自動で保存されません. | 同一のIO-Linkデバイスを接続する場合, 工場出荷時の設定でリカバリされます. |

使用例

ポート 1 の検証方法を IODD1.1 検証モードにし, Vendor: ifm electronic gmbh, Device: OGD596 以外の接続ができないように制限する方法を記述します.

' 接続

```
Call Connect
```

' Extension の追加

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", "Port=1")
```

' 変数追加

```
Dim validationDataStorage As CaoVariable
Set validationDataStorage = extension.AddVariable("@ValidationDataStorage")
Dim varidationVendorId As CaoVariable
Set varidationVendorId = extension.AddVariable("@VaridationVendorId")
Dim varidationDeviceId As CaoVariable
Set varidationDeviceId = extension.AddVariable("@VaridationDeviceId")
```

' 設定

```
validationDataStorage.Value = 2 ' Ver1.1 準拠
varidationVendorId.Value = 310 ' ifm electronic gmbh
varidationDeviceId.Value = 1066 ' OGD596
```

3.3.2.1.7. @VaridationVendorId

@ValidationDataStorage が 2 以上の時に検証するベンダー ID を取得/設定します。
 @ValidationDataStorage が 2 以上の時、接続された I0-Link デバイスのベンダー ID が本変数の値と異なる場合、I0-Link マスターと I0-Link デバイスの接続はできません。Modbus TCP plc への接続が中断された場合のみ、パラメータを編集可能です。

データ型

| 型説明 | |
|-------|-------------|
| VT_I4 | 検証するベンダー ID |

3.3.2.1.8. @VaridationDeviceId

@ValidationDataStorage が 2 以上の時に検証する DeviceID を取得/設定します。
 @ValidationDataStorage が 2 以上の時、接続された I0-Link デバイスのデバイス ID が本変数の値と異なる場合、I0-Link マスターと I0-Link デバイスの接続はできません。Modbus TCP plc への接続が中断された場合のみ、パラメータを編集可能です。

データ型

| 型説明 | |
|-------|-------------|
| VT_I4 | 検証するデバイス ID |

3.3.2.1.9. @DataStorageMaxSize

データストレージエリアの最大サイズ(byte)を取得します。

データ型

| 型説明 | |
|-------|-------------------------|
| VT_I4 | データストレージエリアの最大サイズ(byte) |

3.3.2.1.10. @DataStorageChunkSize

データセグメントのサイズ(byte)を取得します。

データ型

| 型説明 | |
|-------|--------------------|
| VT_I4 | データセグメントのサイズ(byte) |

3.3.2.1.11. @DataStorageSize

データストレージエリアのサイズ(byte)を取得します。

データ型

| 型説明 | |
|-------|------------------------|
| VT_I4 | データストレージエリアのサイズ (byte) |

3.3.2.2. プロセスデータ

3.3.2.2.1. @Pin2Input

I0-Link ポートのピン 2 のデジタル入力の値を取得します。

データ型

| 型説明 | |
|-------|---------------------------|
| VT_I4 | I0-Link ポートのピン 2 のデジタル入力値 |

使用例

' 接続

```
Call Connect
```

' Extension の追加

```
Dim extension As CaoExtension
```

```
Set extension = controller.AddExtension("sensor", _  
                                     "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")
```

' 変数追加

```
Dim var As CaoVariable
```

```
Set var = extension.AddVariable("@Pin2Input")
```

' 値取得

```
Dim value As Long
```

```
value = var.Value
```

3.3.2.2.2. @ProcessDataIn

I0-Link ポートのピン 4 の I0-Link 入力値 (入力プロセスデータ) を取得します。データ型は、CaoController::AddExtension で指定した IODD ファイルに依存します。

データ型

| 型説明 | |
|------------|---|
| VT_VARIANT | @ProcessDataIn のデータ型は指定された IODD ファイルにより異なります。 どのようなデータ型となるかは、指定した IODD ファイルの "IODevice.ProfileBody.ProcessDataCollection.ProcessDataIn" を参照してください。 |

IODD ファイル例

下図は、IODD ファイル「ifm-00042A-20180717-IODD1.1.xml」での入力プロセスデータの定義です。

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <IODevice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.io-link.
3  <DocumentInfo version="V1.3.14.0" releaseDate="2018-07-17" copyright="Copyright 2018, Bu
4  <ProfileHeader>
16 <ProfileBody>
17 <DeviceIdentity vendorId="310" vendorName="ifm electronic gmbh" deviceId="1066">
34 <DeviceFunction>
35 <Features blockParameter="true" dataStorage="true" profileCharacteristic="1 32768 32
38 <VariableCollection>
513 <ProcessDataCollection>
514 <ProcessData id="V_PdT">
515 <ProcessDataIn id="V_PdInT" bitLength="64">
516 <Datatype xsi:type="RecordT" bitLength="64" subindexAccessSupported="false">
517 <RecordItem bitOffset="48" subindex="1">
518 <SimpleDatatype xsi:type="IntegerT" bitLength="16">
519 <ValueRange lowerValue="25" upperValue="330" />
520 <SingleValue value="-32760">
523 <SingleValue value="32760">
526 <SingleValue value="32764">
529 </SimpleDatatype>
530 <Name textId="TI_PD_VR_IN_1_Name" />
531 <Description textId="TI_PD_VR_IN_1_Descr" />
532 </RecordItem>
533 <RecordItem bitOffset="16" subindex="2">
549 <RecordItem bitOffset="4" subindex="3">
570 <RecordItem bitOffset="1" subindex="4">
582 <RecordItem bitOffset="0" subindex="5">
594 </Datatype>
595 <Name textId="TI_PdIn_Name" />
596 </ProcessDataIn>
597 </ProcessData>
598 </ProcessDataCollection>
599 <ErrorTypeCollection>
614 <EventCollection>
647 <UserInterface>
927 </DeviceFunction>
928 </ProfileBody>
929 <CommNetworkProfile xsi:type="IOLinkCommNetworkProfileT" iolinkRevision="V1.1">
952 <ExternalTextCollection>
1158 <Stamp crc="2899871952"><Checker name="IODD-Checker V1.1.1" version="V1.1.1.0"/></Stamp>
1159 </IODevice>

```

図 3-1 実際の ProcessDataIn の定義

ProcessDataIn が RecordT 型であり、レコードが 5 つあることがわかります。

データ型例

ifm-00042A-20180717-IODD1.1.xml を指定した場合のデータ型例を下に示します

型説明

| 型説明 | | |
|------------------------|---------|---|
| VT_VARIANT VT_ARRRAY | | 例ではプロセスデータは RecordT 型となるため、各 subindex に対応する RecordItem 型に対応するデータ型を保持しています。 なお、データの並び順は IODD ファイルに定義された順番ではなく、subindex の昇順になっている点に注意してください。 データ名称は、各 "RecordItem" の "Name" に定義されている文字列を表しています。 |
| 0 | VT_I2 | データ名称: Distance |
| 1 | VT_I2 | データ名称: Reflectivity |
| 2 | VT_UI1 | データ名称: Device Status |
| 3 | VT_BOOL | データ名称: OUT2 |
| 4 | VT_BOOL | データ名称: OUT1 |

使用例

Port 1 に接続された ifm 製光学センサ「0GD596」から入力プロセスデータを取得する例を示します。

接続

```
Call Connect
```

Extension の追加

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")
```

変数追加

```
Dim var As CaoVariable
Set var = extension.AddVariable("@ProcessDataIn")
```

値取得

```
Dim value As Variant
value = var.Value
```

Distance 取得

```
Dim distance As Long
distance = value(0)
```

Reflectivity 取得

```
Dim reflectivity As Long
Reflectivity = value(1)
```

DeviceStatus 取得

```
Dim deviceStatus As Byte
deviceStatus = value(2)
```

' OUT2 取得

```
Dim out2 As Boolean
out2 = value(3)
```

' OUT1 取得

```
Dim out1 As Boolean
out1 = value(4)
```

3.3.2.2.3. @ProcessDataOut

IO-Link ポートのピン 4 の IO-Link 出力値(出力プロセスデータ)を取得/設定します。データ型は、CaoController::AddExtension で指定した IODD ファイルに依存します。

データ型

| 型説明 | |
|------------|---|
| VT_VARIANT | @ProcessDataOut のデータ型は指定された IODD ファイルにより異なります。 どのようなデータ型となるかは、指定した IODD ファイルの "IODevice.ProfileBody.ProcessDataCollection.ProcessDataOut" を参照してください。 |

IODD ファイル例

下図は、IODD ファイル「Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml」での出力プロセスデータの定義です。

```

30 <DeviceFunction>
31 <Features blockParameter="true" dataStorage="true">
34 <DatatypeCollection>
156 <VariableCollection>
252 <ProcessDataCollection>
253 <ProcessData id="V_PdT">
254 <ProcessDataIn id="V_PdInSegmentT" bitLength="8">
263 <ProcessDataOut id="V_Pd_OutSegmentT" bitLength="64">
264 <Datatype xsi:type="RecordT" bitLength="64">
265 <RecordItem subindex="1" bitOffset="59">
269 <RecordItem subindex="2" bitOffset="56">
273 <RecordItem subindex="3" bitOffset="63">
277 <RecordItem subindex="4" bitOffset="60">
281 <RecordItem subindex="5" bitOffset="51">
285 <RecordItem subindex="6" bitOffset="48">
289 <RecordItem subindex="7" bitOffset="55">
293 <RecordItem subindex="8" bitOffset="52">
297 <RecordItem subindex="9" bitOffset="43">
301 <RecordItem subindex="10" bitOffset="40">
302 <DatatypeRef datatypeId="DT_Color" />
303 <Name textId="TI_PD_Color_Segment5" />
304 </RecordItem>
305 <RecordItem subindex="11" bitOffset="32">
309 <RecordItem subindex="12" bitOffset="24">
313 <RecordItem subindex="13" bitOffset="16">
317 <RecordItem subindex="14" bitOffset="8">
321 <RecordItem subindex="15" bitOffset="0">
325 </Datatype>
326 <Name textId="TI_PD_Out" />
327 </ProcessDataOut>
328 </ProcessData>
329 </ProcessDataCollection>
330 <EventCollection>

```

eXtensible Marki length : 26,952 lines : 510 Ln : 1 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8-BOM INS

図 3-2 実際の ProcessDataOut の定義

ProcessDataOut が RecordT 型であり、レコードが 15 あることがわかります。
また、各レコードのデータ型は DataTypeCollection の要素を参照しています。

データ型例

Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml を指定した場合のデータ型例を下に示します

型説明

| 型説明 | | |
|-------------------------|---------|--|
| VT_VARIANT VT_ARRRAY | | <p>例ではプロセスデータは RecordT 型となるため、各 subindex に対応する RecordItem 型に対応するデータ型を保持しています。</p> <p>なお、データの並び順は IODD ファイルに定義された順番ではなく、subindex の昇順になっている点に注意してください。</p> <p>データ名称は、各 "RecordItem" の "Name" に定義されている文字列を表しています。</p> |
| 0 | VT_BOOL | <p>データ名称: Segment 1 blink/dominant</p> <p>FALSE: 点滅なし</p> <p>TRUE: 点滅あり</p> |
| 1 | VT_UI1 | <p>データ名称: Segment 1 color / Background color</p> <p>0: OFF</p> <p>1: 緑</p> <p>2: 赤</p> <p>3: 黄</p> <p>4: 青</p> <p>5: オレンジ</p> <p>6: ユーザー定義</p> <p>7: 白</p> |
| 2 | VT_BOOL | <p>データ名称: Segment 2 blink/dominant</p> <p>FALSE: 点滅なし</p> <p>TRUE: 点滅あり</p> |
| 3 | VT_UI1 | <p>データ名称: Segment 2 color / Running color</p> <p>0: OFF</p> <p>1: 緑</p> <p>2: 赤</p> <p>3: 黄</p> <p>4: 青</p> <p>5: オレンジ</p> <p>6: ユーザー定義</p> <p>7: 白</p> |
| 4 | VT_BOOL | <p>データ名称: Segment 3 blink/dominant</p> <p>FALSE: 点滅なし</p> <p>TRUE: 点滅あり</p> |

| 型説明 | | |
|-----|---------|--|
| 5 | VT_UI1 | データ名称: Segment 3 color / Running color 0: OFF 1: 緑 2: 赤 3: 黄 4: 青 5: オレンジ 6: ユーザー定義 7: 白 |
| 6 | VT_BOOL | データ名称: Segment 4 blink/dominant FALSE: 点滅なし TRUE: 点滅あり |
| 7 | VT_UI1 | データ名称: Segment 4 color 0: OFF 1: 緑 2: 赤 3: 黄 4: 青 5: オレンジ 6: ユーザー定義 7: 白 |
| 8 | VT_BOOL | データ名称: Segment 5 blink/dominant FALSE: 点滅なし TRUE: 点滅あり |
| 9 | VT_UI1 | データ名称: Segment 5 color 0: OFF 1: 緑 2: 赤 3: 黄 4: 青 5: オレンジ 6: ユーザー定義 7: 白 |

| 型説明 | | |
|-----|--------|---|
| 10 | VT_UI1 | データ名称: Operating mode 0: セグメントモード 1: レベルモード 2: ランライトモード |
| 11 | VT_UI1 | データ名称: Number of segments / Level type |
| 12 | VT_UI1 | データ名称: Blinking mode / Level value low byte |
| 13 | VT_UI1 | データ名称: Blinking frequency / Level value high byte / Running speed |
| 14 | VT_UI1 | データ名称: Reserved |

使用例

Port 2 に接続された BALLUFF 製制御ランプ「BNI0082」を、ランニングモードでオレンジ色に光らせるサンプルを記述します。

接続

```
Call Connect
```

Extension の追加

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD= Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml ")
```

変数追加

```
Dim var As CaoVariable
Set var = extension.AddVariable("@ProcessDataOut")
```

値取得

```
Dim processDataOutValues As Variant
processDataOutValues = var.Value
```

オペレーティングモードをランニングモードに設定する

```
processDataOutValues(10) = 4
```

セグメント1の背景カラーをオレンジに設定する

```
processDataOutValues(1) = 5
```

値をデバイスに書き込む

```
var.Value = processDataOutValues
```

3.3.2.3. IO-Link デバイス設定**3.3.2.3.1. <IODD ファイル変数>**

IO-Link デバイス設定情報を取得/設定します。IODD ファイルの SpecialistRoleMenuSet に定義されている変数を指定することができます。

データ型

| 型説明 | |
|------------|---|
| VT_VARIANT | @ProcessDataOut のデータ型は指定された IODD ファイルにより異なります。 どのようなデータ型となるかは、指定した IODD ファイルの "IODevice.ProfileBody.ProcessDataCollection.ProcessDataOut" を参照してください。 |

IODD ファイル例

下図は、IODD ファイル「Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml」での IODD ファイル変数の定義です。

```

34
156 <DatatypeCollection>
157 <VariableCollection>
158 <StdVariableRef id="V_DirectParameters_1" />
159 <StdVariableRef id="V_DirectParameters_2" />
160 <StdVariableRef id="V_SystemCommand" />
161 <StdVariableRef id="V_DeviceAccessLocks" />
162 <StdVariableRef id="V_VendorName" default="BALLUFF" />
163 <StdVariableRef id="V_VendorText" default="www.balluff.com" />
164 <StdVariableRef id="V_ProductName" default="BNI IOL-802-102-Z036" />
165 <StdVariableRef id="V_ProductID" default="BNI0082" />
166 <StdVariableRef id="V_ProductText" default="Smart Light 5 segment" />
167 <StdVariableRef id="V_HardwareRevision" />
168 <StdVariableRef id="V_FirmwareRevision" />
169 <StdVariableRef id="V_ApplicationSpecificTag" />
170 <Variable id="V_Resolution" index="67" accessRights="rw" default="4" />
171 <Variable id="V_LevelLimit12" index="73" accessRights="rw" dynamic="true" default="0" />
172 <Variable id="V_LevelLimit23" index="74" accessRights="rw" dynamic="true" default="0" />
173 <Variable id="V_LevelLimit34" index="75" accessRights="rw" dynamic="true" default="0" />
174 <Variable id="V_LevelLimit45" index="76" accessRights="rw" dynamic="true" default="0" />
175 <Variable id="V_SupplyMonitor" index="80" accessRights="ro" />
176 <Variable id="V_Brightness" index="81" accessRights="rw" />
177 <Variable id="V_UserColor" index="252" accessRights="rw" />
178 <Variable id="V_LimitType" index="253" accessRights="rw" default="0" />
179 </VariableCollection>
252 <ProcessDataCollection>
330 <EventCollection>
333 <UserInterface>
334 <MenuCollection>
335 <Menu id="M OR MR SR Ident">
336 <VariableRef variableId="V_VendorName" />
337 <VariableRef variableId="V_VendorText" />
338 <VariableRef variableId="V_ProductName" />
339 <VariableRef variableId="V_ProductID" />
340 <VariableRef variableId="V_ProductText" />
341 <VariableRef variableId="V_HardwareRevision" />
342 <VariableRef variableId="V_FirmwareRevision" />
343 <VariableRef variableId="V_ApplicationSpecificTag" />
344 </Menu>
345 <Menu id="M OR Param">
346 <VariableRef variableId="V_DeviceAccessLocks" displayFormat="Hex" accessRightRestriction="ro" />
347 <VariableRef variableId="V_Resolution" accessRightRestriction="ro" displayFormat="Hex" />
348 <VariableRef variableId="V_LevelLimit12" accessRightRestriction="ro" displayFormat="Hex" />
349 <VariableRef variableId="V_LevelLimit23" accessRightRestriction="ro" displayFormat="Hex" />
350 <VariableRef variableId="V_LevelLimit34" accessRightRestriction="ro" displayFormat="Hex" />
351 <VariableRef variableId="V_LevelLimit45" accessRightRestriction="ro" displayFormat="Hex" />
352 <VariableRef variableId="V_SupplyMonitor" accessRightRestriction="ro" displayFormat="Hex" />
353 <VariableRef variableId="V_Brightness" accessRightRestriction="ro" displayFormat="Hex" />
354 <VariableRef variableId="V_UserColor" accessRightRestriction="ro" displayFormat="Hex" />
355 <VariableRef variableId="V_LimitType" accessRightRestriction="ro" displayFormat="Hex" />
356 </Menu>
357 <Menu id="M MR Param">
358 </Menu>
359 <Menu id="M SR Param">
360 </Menu>
361 </MenuCollection>
362 <ObserverRoleMenuSet>
363 <MaintenanceRoleMenuSet>
364 <SpecialistRoleMenuSet>
365 <IdentificationMenu menuId="M OR MR SR Ident" />
366 <ParameterMenu menuId="M SR Param" />
367 </SpecialistRoleMenuSet>
368 </UserInterface>
369 </DeviceFunction>

```

図 3-3 実際の IODD 変数の定義の様子

SpecialistRoleMenuSet からどのように IODD ファイル変数を特定するかの参照ルートが矢印で示しています。プロバイダには SpecialistRoleMenuSet 内に定義されている menuId に対応する Menu 内の、VariableRef および RecordItemRef で参照されている VariableCollection 内の Variable もしくは RecordItem を IODD ファイル変数として追加できます。

変数名決定方法

IODD ファイル変数の変数名は以下のルールに従ってそれぞれのアイテム内の Name 要素で参照されている、ExternalTextCollection.PrimaryLanguage 内のテキスト文字列を使用して作成されます。

VariableRef の参照先が Record 型でない場合、または Record 型でも subindexAccessSupport 属性に false が指定されている場合、変数名は「Variable の名称」となります。アイテムが RecordItemRef

の場合、または、VariableRef の参照先が Record 型で subindexAccessSupported 属性に false が指定されていない場合には、変数名は「Variable の名称.RecordItem の名称」となります。

なお、複数の IODD ファイル変数で同じ名前になってしまう場合は、重複した変数名にインデックスが付与された変数名（“_[index]”が付与）になります（下例を参照）。

- 重複した場合の変数名の例

例えば以下のように別々の ID を持つ Variable / RecordItem にも関わらず同じ変数名になってしまった場合には、重複した Variable 要素の定義順に、変数名にインデックスが付与されます。同じ ID を持つ Variable / RecordItem が複数の Menu で参照されている場合は、最初に定義されたものを採用しています。

| IODD での Variable ID | IODD での定義名 | プロバイダでの変数名 |
|---------------------|------------|------------|
| dummy_1 | Dummy | Dummy |
| dummy_2 | Dummy | Dummy_2 |
| dummy_3 | Dummy | Dummy_3 |

実際の IODD フィル変数例

下表は、IODD ファイル「Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml」を読み込んだ際の IODD ファイル変数の一覧です。併せてそれぞれの変数のデータ型も記載しています。

| プロバイダ変数名 | データ型 |
|--------------------------|---|
| Vendor Name | VT_BSTR |
| Vendor Text | VT_BSTR |
| Product Name | VT_BSTR |
| Product ID | VT_BSTR |
| Product Text | VT_BSTR |
| Hardware Version | VT_BSTR |
| Firmware Version | VT_BSTR |
| Application Specific Tag | VT_BSTR |
| Device Access Locks | VT_ARRAY VT_VARIANT |
| | 0 VT_BOOL: Parameter (write) Access Lock false: Unlock true: Lock |
| | 1 VT_BOOL: Data Storage Lock false: Unlock true: Lock |
| | 2 VT_BOOL: Local Parameterization Lock false: Unlock |

| プロバイダ変数名 | データ型 |
|---------------------------------------|--|
| | true: Lock |
| | 3 VT_BOOL: Local User Interface Lock false: Unlock true: Lock |
| Standard Command | VT_UI1 |
| Resolution of level indicator | VT_UI1 0: 8 bit 1: 10 bit 2: 12 bit 3: 14 bit 4: 16 bit |
| Level mode, limit segment 1 and 2 | VT_UI2 |
| Level mode, limit segment 2 and 3 | VT_UI2 |
| Level mode, limit segment 3 and 4 | VT_UI2 |
| Level mode, limit segment 4 and 5 | VT_UI2 |
| Supply monitoring. Supply Voltage | VT_BOOL false: OK true: Low Us |
| Supply monitoring. Supply Voltage LED | VT_BOOL false: OK true: Low LED Voltage |
| Brightness. Red channel brightness | VT_UI1 |
| Brightness. Green channel brightness | VT_UI1 |
| Brightness. Blue channel brightness | VT_UI1 |
| User defined color. Red channel | VT_UI1 |
| User defined color. Green channel | VT_UI1 |
| User defined color. Blue channel | VT_UI1 |
| Limit register type | VT_UI1 0: Percent 1: Absolute |

3.3.2.4. I0-Link デバイス情報

3.3.2.4.1. @Status

接続されている I0-Link デバイスのステータスを取得します。

データ型

| 型説明 | |
|-------|---|
| VT_I4 | IO-Link デバイスのステータス 0x00: 接続されていない 0x02: 接続されている |

使用例**' 接続**

```
Call Connect
```

' Extension の追加

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")
```

' 変数追加

```
Dim var As CaoVariable
Set var = extension.AddVariable("@Status")
```

' 値取得

```
Dim value As Long
value = var.Value
```

3.3.2.4.2. @VendorId

接続されている IO-Link デバイスの製造元 ID を取得します。

データ型

| 型説明 | |
|-------|---------------------|
| VT_I4 | IO-Link デバイスの製造元 ID |

使用例**' 接続**

```
Call Connect
```

' Extension の追加

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")
```

' 変数追加

```
Dim var As CaoVariable
Set var = extension.AddVariable("@VendorId")
```

' 値取得

```
Dim value As Long
```

```
value = var.Value
```

3.3.2.4.3. @DeviceId

接続されている I0-Link デバイスのデバイス ID を取得します。

データ型

| 型説明 | |
|-------|----------------------|
| VT_I4 | I0-Link デバイスのデバイス ID |

使用例

```
' 接続
```

```
Call Connect
```

```
' Extension の追加
```

```
Dim extension As CaoExtension
```

```
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")
```

```
' 変数追加
```

```
Dim var As CaoVariable
```

```
Set var = extension.AddVariable("@DeviceId")
```

```
' 値取得
```

```
Dim value As Long
```

```
value = var.Value
```

3.3.2.4.4. @ProductName

接続されている I0-Link デバイスの品名称を取得します。

データ型

| 型説明 | |
|---------|------------------|
| VT_BSTR | I0-Link デバイスの品名称 |

使用例

```
' 接続
```

```
Call Connect
```

```
' Extension の追加
```

```
Dim extension As CaoExtension
```

```
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")
```

```
' 変数追加
```

```
Dim var As CaoVariable
```

```
Set var = extension.AddVariable("@ProductName")
```

```
' 値取得
```

```
Dim value As String
value = var.Value
```

3.3.2.4.5. @Serial

接続されている I0-Link デバイスのシリアル番号を取得します。

データ型

| 型説明 | |
|---------|---------------------|
| VT_BSTR | I0-Link デバイスのシリアル番号 |

使用例

```
' 接続
```

```
Call Connect
```

```
' Extension の追加
```

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
    "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")
```

```
' 変数追加
```

```
Dim var As CaoVariable
Set var = extension.AddVariable("@Serial")
```

```
' 値取得
```

```
Dim value As String
value = var.Value
```

3.3.2.4.6. @ApplicationTag

接続されている I0-Link デバイスのアプリケーションタグを取得/設定します。

データ型

| 型説明 | |
|---------|-------------------------|
| VT_BSTR | I0-Link デバイスのアプリケーションタグ |

使用例

```
' 接続
```

```
Call Connect
```

```
' Extension の追加
```

```
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
```

“Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml”

’ 変数追加

```
Dim var As CaoVariable  
Set var = extension.AddVariable("@ApplicationTag")
```

’ 値取得

```
Dim value As String  
value = var.Value
```

4. プロバイダによるプログラミング

プロバイダは前述したとおり各変数の取得/設定のたびに I0-Link マスタと通信します。CaoController/CaoExtension の作成時には通信確認は行っていません。

4.1. タワーライトを制御するサンプルプログラミング

ここでは例として Port 2 に接続された BALLUFF 製制御ランプ「BNI0082」を、ランニングモードでオレンジ色に光らせるサンプルプログラムを示します。下記にサンプルプログラムの要件を記述しています。

表 4-1 サンプルプログラムの要件

| 要件 | 説明 |
|------|---|
| 接続先 | 192.168.0.1 |
| | ポート番号は 1 |
| | I0-Link デバイスは, Vendor: Balluff, Device: BNI0082 |
| 処理内容 | プロセスデータアウトの値を書き換える。 |

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 ControlBNI0082.vb
' オブジェクト
Dim engine As CaoEngine
Dim workspace As CaoWorkspace
Dim controller As CaoController
Dim extension As CaoExtension

Private Sub Main
    Dim processDataOut As CaoVariable

    ' 初期化
    Call Initialize

    Set processDataOut = extension.AddVariable("@ProcessDataOut")

    ' 値取得
    Dim processDataOutValues As Variant
    processDataOutValues = var.Value
    ' オペレーティングモードをランニングモードに設定する
    processDataOutValues(10) = 4

```

```
' セグメント 1 の背景カラーをオレンジに設定する
processDataOutValues(1) = 5
' 値をデバイスに書き込む
var.Value = processDataOutValues

Call extension.Variables.Remove(processDataOut.Index)
Set processDataOut = Nothing

' 終了処理
Call Finalize
End Sub

' 初期化メソッド
Private Sub Initialize()
' 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", "")
' CaoController オブジェクトの生成
Set controller = workspace.AddController("Master1", _
                                         "CaoProv.ifm.IO-Link", _
                                         "", _
                                         "Server=192.168.0.1")

' CaoExtension オブジェクトの生成
Set extension = controller.AddExtension("BNI0082", _
                                         "Port=1, IODD= Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml ")

End Sub

' 終了メソッド
Private Sub Finalize()
' CaoController から CaoExtension を削除
Call controller.Extensions.Remove(extension.Index)
Set extension = Nothing

' CaoWorkspace から CaoController を削除
Call workspace.Controllers.Remove(controller.Index)
Set controller = Nothing

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

' CaoEngine の消去
Set engine = Nothing
End Sub
```

5. 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。

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 独自エラーコード表

| エラー番号 | 説明 |
|-------------------------|---|
| 0x80110001 | 必須オプションが指定されていません。ユーザーズガイドで必須と指定されているオプションは省略できません。 |
| 0x80110002 | 指定されたオプションが値範囲外です。ユーザーズガイドで示された値範囲内の値の指定してください。 |
| 0x80110003 | 書込みデータのデータ型が不正なため変換できません。ユーザーズガイドで示されたデータ型、もしくはメモリの的に変換可能なデータ型で書込みをしてください。 |
| 0x80110004 | 受信したデータのデータ型が想定と異なります。このエラーが発生した場合はお手数ですがお問い合わせください。 |
| 0x80110005 | 同じポート番号を指定されたCaoExtensionがすでに存在しています。異なるExtensionで同一のポート番号は指定できません。どちらかをいったん削除してください。 |
| 0x80111000 | 指定されたIODDファイル、もしくはIODD-StandardDefinitions1.1.xmlが存在しません。パスの指定を見直してください。 IODD-StandardDefinitions1.1.xmlファイルが存在しない場合は、Bin¥XML¥以下に配置してください。 |
| 0x80112000 | 指定されたIODDファイルのバージョンが異なる、もしくは指定されたファイルがIODDファイルではありません。 IODDバージョンは1.1のみ対応しています。複数バージョンがある場合はバージョン1.1のIODDファイルを指定してください。 |
| 0x801130** ² | IODDファイルのProcessDataInデータの解析中に想定外のエラーが発生しました。ご使用のIODDファイルが認可を受けたファイルかを確認してください。 |
| 0x801140** ² | IODDファイルのProcessDataOutデータの解析中に想定外のエラーが発生しました。ご使用のIODDファイルが認可を受けたファイルかを確認してください。 |
| 0x801150** ² | IODDファイルのVariableデータの解析中に想定外のエラーが発生しました。ご使用のIODDファイルが認可を受けたファイルかを確認してくだ |

² **には具体的にどのようなエラーがあったかを示すコードが入ります。詳細コードは別表を参照してください。

| エラー番号 | 説明 |
|-------------|--|
| | さい。 |
| 0x801160**2 | IDDファイルのTextデータの解析中に想定外のエラーが発生しました。ご使用のIODDファイルが認可を受けたファイルかを確認してください。 |
| 0x801170**2 | IODDファイルのVariableCollectionデータの解析中に想定外のエラーが発生しました。ご使用のIODDファイルが認可を受けたファイルかを確認してください。 |
| 0x801180**2 | IDDファイルのDataTypeデータの解析中に想定外のエラーが発生しました。ご使用のIODDファイルが認可を受けたファイルかを確認してください。 |

表 5-2 IODD ファイルデータ解析中の詳細エラーコード

| エラー番号 | 説明 |
|-------|------------------------------|
| 0x01 | 必須属性が存在しません。 |
| 0x02 | 必須要素が存在しません。 |
| 0x03 | 属性値が範囲外です。 |
| 0x04 | 要素値が範囲外です。 |
| 0x05 | 指定されたテキストが存在しません。 |
| 0x06 | 指定されProcessDataが存在しません。 |
| 0x07 | 指定されたVariableが存在しません。 |
| 0x08 | 指定されたSubindexを持つ要素が存在しません。 |
| 0x09 | 指定されたDatatypeが存在しません。 |
| 0x0A | サポートされていないDatatypeが指定されています。 |

また、本プロバイダは、IO-Link マスターからのエラーコードを「0x8010****」で、Http エラーコードを「0x8012****」でマスクして返します。

付録A. IODD データ型と VARIANT データ型の対応表

IODD の各データ型に対応する VARIANT データ型の対応表をここに定義します。

| IODD データ型 | VARIANT データ型 |
|-----------|--|
| BooleanT | VT_BOOL |
| UIntegerT | $2 \leq \text{bitlength} \leq 8$: VT_UI1 $8 < \text{bitlength} \leq 16$: VT_UI2 $16 < \text{bitlength} \leq 32$: VT_UI4 $32 < \text{bitlength} \leq 64$: VT_UI8 |

| IODD データ型 | VARIANT データ型 |
|--------------|---|
| IntegerT | 2 ≤ <i>bitlength</i> ≤ 8: VT_I1 8 < <i>bitlength</i> ≤ 16: VT_I2 16 < <i>bitlength</i> ≤ 32: VT_I4 32 < <i>bitlength</i> ≤ 64: VT_I8 |
| Float32T | VT_R4 |
| StringT | VT_BSTR |
| OctetStringT | VT_UI1 VT_ARRAY |
| TimeT | VT_DATE |
| TimeSpanT | VT_I8 |
| RecordT | VT_VARIANT VT_ARRAY |
| ArrayT | 定義されているデータ型の配列 |

付録B. 通信プロトコルコマンド対応表

プロバイダは Rest API の POST メソッドを用いて IO-Link マスタと通信しています。以下の表には各変数の Get/Put で adr フィールドに

| 変数名 | JSON adr フィールドの値 | |
|---------------------|---------------------------------------|----------------------------------|
| | Get | Put |
| CaoController 変数 | | |
| @ProductCode | deviceinfo/productcode/getdata | - |
| @Vendor | deviceinfo/vendor/getdata | - |
| @DeviceFamily | deviceinfo/devicefamily/getdata | - |
| @HWRevision | deviceinfo/hwrevision/getdata | - |
| @SerialNumber | deviceinfo/serialnumber/getdata | - |
| @SWRevision | deviceinfo/swrevision/getdata | - |
| @BootLoaderRevision | deviceinfo/bootloaderrevision/getdata | - |
| @ExtensionRevisions | deviceinfo/extensionrevisions/getdata | - |
| @FieldbusType | deviceinfo/fieldbustype/getdata | - |
| @IOLinkMasterName | devicetag/applicationtag/getdata | devicetag/applicationtag/setdata |

| | | |
|------------------------|--|--|
| @IOLinkMasterVersion | firmware/version/getdata | - |
| @IOLinkMasterSWType | firmware/type/getdata | - |
| @ContainerMaxSize | firmware/container/maxsize/getdata | - |
| @ContainerChunkSize | firmware/container/chunksize/getdata | - |
| @ContainerSize | firmware/container/size/getdata | - |
| @Temperature | processdatamaster/temperature/getdata | - |
| @Voltage | processdatamaster/voltage/getdata | - |
| @Current | processdatamaster/current/getdata | - |
| @SuperVisionStatus | processdatamaster/supervisionstatus/getdata | - |
| @Version | - | - |
| CaoExtension 変数 | | |
| @SendProcDataPort | iolinkmaster/port[n]/senddatatosmob/getdata | iolinkmaster/port[n]/senddatatosmob/setdata |
| @TransCycleTimePreset | iolinkmaster/port[n]/mastercycletime_preset/getdata | - |
| @TransCycleTimeAct | iolinkmaster/port[n]/mastercycletime_actual/getdata | - |
| @Mode | iolinkmaster/port[n]/mode/getdata | iolinkmaster/port[n]/mode/setdata |
| @ComSpeed | iolinkmaster/port[n]/comspeed/getdata | iolinkmaster/port[n]/comspeed/setdata |
| @ValidationDataStorage | iolinkmaster/port[n]/validation_datastorage_mode/getdata | iolinkmaster/port[n]/validation_datastorage_mode/setdata |
| @ValidationVendorId | iolinkmaster/port[n]/validation_vendor_id/getdata | iolinkmaster/port[n]/validation_vendor_id/setdata |
| @ValidationDeviceId | iolinkmaster/port[n]/validation_deviceid/getdata | iolinkmaster/port[n]/validation_deviceid/setdata |
| @DataStorageMaxSize | iolinkmaster/port[n]/datastorage_maxsize/getdata | - |

| | | |
|-----------------------|--|--|
| @DataStorageChunkSize | iolinkmaster/port[n]/datastorage/chunksize/getdata | - |
| @DataStorageSize | iolinkmaster/port[n]/datastorage/size/getdata | - |
| @Pin2Input | iolinkmaster/port[n]/pin2in/getdata | - |
| @ProcessDataIn | iolinkmaster/port[n]/iolinkdevice/pdin/getdata | - |
| @ProcessDataOut | iolinkmaster/port[n]/iolinkdevice/pdout/getdata | iolinkmaster/port[n]/iolinkdevice/pdout/setdata |
| <IODD ファイル変数> | iolinkmaster/port[n]/iolinkdevice/iolreadacyclic | iolinkmaster/port[n]/iolinkdevice/iolwriteacyclic |
| @Status | iolinkmaster/port[n]/iolinkdevice/status/getdata | - |
| @VendorId | iolinkmaster/port[n]/iolinkdevice/vendorid/getdata | - |
| @DeviceId | iolinkmaster/port[n]/iolinkdevice/deviceid/getdata | - |
| @ProductName | iolinkmaster/port[n]/iolinkdevice/productname/getdata | - |
| @Serial | iolinkmaster/port[n]/iolinkdevice/serial/getdata | - |
| @ApplicationTag | iolinkmaster/port[n]/iolinkdevice/applicationtag/getdata | iolinkmaster/port[n]/iolinkdevice/applicationtag/setdata |

付録C. IO-Link マスタからのエラー一覧

| エラー番号 ()内は 10 進数値 | 説明 |
|-----------------------|--|
| 0x0190 (400) | 無効なリクエストです。 |
| 0x0191 (401) | 認証エラーです。 SSLの設定, ユーザ名, パスワード名が異なっている可能性があります。 |
| 0x0193 (403) | アクセス拒否されました。 SSLの設定, ユーザ名, パスワード名が異なっている可能性があります。 |
| 0x0194 (404) | アドレスが見つかりません。 |
| 0x01F4 (500) | 内部サーバーエラーです。 |

| エラー番号 ()内は 10 進数値 | 説明 |
|-----------------------|--|
| 0x01F7 (503) | サービスが動いていません。 ポートにI0-Linkデバイスが接続されていない状態でI0-Linkデバイスの情報にアクセスしている可能性があります。 |
| 0x0212 (530) | 入力データが異常です。 |
| 0x0213 (531) | I0-Linkエラーです。 I0-Linkマスタ, I0-Linkデバイスが異常状態です。 |
| 0x0214 (532) | PLC接続エラーです。 I0-LinkマスタがすでにフィールドバスPLCと接続されています。 |

付録D. IODD ファイル

IODD ファイルの入手方法

I0-Link デバイスと接続するためには、IODD (I0 Device Description) ファイルが必要です。指定するファイルは下記 URL よりダウンロードして使用してください。

<https://ioddfinder.io-link.com>

なお、プロバイダが対応しているスキーマバージョンは 1.1 のため、ダウンロードする際にはご注意ください。下図は上記 URL から IODD ファイルを検索した際の例です。

| SEARCH_TABLE.VENDOR_NAME | SEARCH_TABLE.PRODUCT_NAME | SEARCH_TABL | SEARCH_TABL | SEARCH_TABL | SEARCH_TABL | SEARCH_TA |
|--------------------------|------------------------------|-------------|-------------|-------------|-------------|-----------|
| Balluff GmbH | BOD 24K-LPI07-S4 | 50142212 | 200710 | 1.1 | V0.1 | |
| Balluff GmbH | BAE SA-CS-027-YI-BP00,3-GS04 | BAE00LC | 459009 | 1.1 | V1.1 | |
| Balluff GmbH | BAE PS-XA-1S-24-100-103 | BAE00LJ | 66821 | 1.1 | V1.1 | |
| Balluff GmbH | BAE PS-XA-1S-24-200-104 | BAE00M3 | 66822 | 1.1 | V1.1 | |
| Balluff GmbH | BAE PS-XA-1S-24-050-102 | BAE00T4 | 66819 | 1.1 | V1.1 | |

IODD ファイルの見分け方

IODD ファイルは下図のようにルート要素が <IODevice> となっており、<IODevice>要素の xsi:schemaLocation 属性には、"http://www.io-link.com/IODD/2010/10 IODD1.1.xsd"と記述されています。IODD ファイルを指定しているにも関わらずエラーとなる場合は、指定したファイルの中身を確認し、正常な IODD ファイルかを確かめてください。

また、プロバイダインストール時に標準で同封されている"IODD-StandardDefinitions.xml"は、IODD ファイルではなく、すべての IODD ファイルが参照する定義ファイルとなります。このファイルを IODD オプションに指定しないでください。

```
<?xml version="1.0" encoding="utf-8"?>
<IODevice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.io-link.com/IODD/2010/10" xsi:schemaLocation="http://www.io-link.com/IODD/2010/10 IODD1.1.xsd" >
  <DocumentInfo version="V1.1" releaseDate="2016-11-02" copyright="Copyright (c) 2016 by Weiss Robotics GmbH & Co. KG" />
  <ProfileHeader>
    <ProfileIdentification>IO Device Profile</ProfileIdentification>
    <ProfileRevision>1.1</ProfileRevision>
    <ProfileName>Device Profile for IO Devices</ProfileName>
    <ProfileSource>IO-Link Consortium</ProfileSource>
    <ProfileClassID>Device</ProfileClassID>
    <ISO15745Reference>
      <ISO15745Part>1</ISO15745Part>
      <ISO15745Edition>1</ISO15745Edition>
    </ISO15745Reference>
    <ProfileTechnology>IODD</ProfileTechnology>
  </ProfileHeader>
  <ProfileBody>
    <DeviceIdentity vendorId="815" vendorName="Weiss Robotics" deviceId="20">
      <VendorText textId="TN_VendorText" />
      <VendorUrl textId="TN_VendorUrl" />
      <VendorLogo name="WeissRobotics-logo.png" />
      <DeviceName textId="TN_DeviceName" />
      <DeviceFamily textId="TN_DeviceFamily" />
      <DeviceVariantCollection>
        <DeviceVariant productId="IEG 55-020" deviceSymbol="WeissRobotics-IEG-55-pic.png" deviceIcon="WeissRobotics-IEG-55-pic.png">
          <Name textId="TN_Variant_IEG55" />
          <Description textId="TD_Variant_IEG55" />
        </DeviceVariant>
        <DeviceVariant productId="IEG 76-030" deviceSymbol="WeissRobotics-IEG-76-pic.png" deviceIcon="WeissRobotics-IEG-76-pic.png">
          <Name textId="TN_Variant_IEG76" />
          <Description textId="TD_Variant_IEG76" />
        </DeviceVariant>
      </DeviceVariantCollection>
    </DeviceIdentity>
  </ProfileBody>
</IODevice>
```