

日立アプライアンス株式会社  
日立環境試験装置プロバイダ

Version 1.1.3

ユーザーズ ガイド

November 30, 2018

備考：

## 【改版履歴】

バージョン	日付	内容
1.1.0	2018-09-28	<p>以下の変数を追加.</p> <p>@LASTMEAS_PRG_SET_STATUS PRG_SET_STATUS</p> <p>以下の変数にリトライ回数オプションを追加.</p> <p>@MEASUREMENT_STATUS @FIX_SET_STATUS @LASTMEAS_PRG_SET_STATUS PRG_SET_STATUS</p>
1.1.1	2018-10-01	GaoController::Execute の各コマンドの省略可能引数を必須引数として扱っていた不具合を修正.
1.1.2	2018-10-30	GaoController 削除時にメモリリークが発生する問題を修正. 変数のデータ説明を修正.
1.1.3	2018-11-20	Execute メソッドの引数の不具合を修正. モード違いのデータを受信した際の動作を修正. イーサネットオプション基板に接続した場合の動作を修正.
	2018-11-30	文言修正

## 【動作確認機種】

機種	バージョン	注意事項
EC-46MHPES	CPU: 04.04.04 温調基板: 01.03.09 サイクル基板: 04.04.02 LCD 表示器: 05.02.08	Web 基板オプションの場合に頻繁に通信を行うとポートの枯渇が発生する可能性がありますのでご注意ください.
EC-25EXHHS	CPU: 02.11.18 温調基板: 01.04.02 サイクル基板 1/2: 02.10.14 LCD 表示器: 02.02.07	Web 基板オプションの場合に頻繁に通信を行うとポートの枯渇が発生する可能性がありますのでご注意ください.

## 目次

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.....	6
1.2. 参考となる情報源.....	6
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. 日立環境試験装置とクライアント PC との接続.....	7
2.2. PC 開発環境のセットアップ.....	9
2.2.1. ThermoStat プロバイダの自動インストール.....	9
2.2.2. ThermoStat プロバイダの手動インストール.....	9
3. ThermoStat プロバイダによるプログラミング.....	11
3.1. 日立環境試験装置の測定を受信するサンプルプログラミング.....	11
3.1.1. サンプルプログラム.....	12
3.1.1.1. 接続.....	14
3.1.1.2. 測定を取得する.....	15
3.1.1.3. 切断.....	16
4. コマンドリファレンス.....	17
4.1. メソッド/プロパティ一覧.....	17
4.2. メソッド・プロパティ.....	17
4.2.1. CaoWorkspace クラス.....	17
4.2.1.1. AddController メソッド.....	17
4.2.2. CaoController クラス.....	19
4.2.2.1. VariabeNames プロパティ.....	19
4.2.2.2. Variables プロパティ.....	19
4.2.2.3. AddVariable メソッド.....	19
4.2.2.4. Execute メソッド.....	20
4.2.3. CaoVariable クラス.....	26
4.2.3.1. Value プロパティ.....	26
4.3. 変数一覧.....	26
4.3.1. CaoController クラス変数.....	26
4.3.1.1. @MAKER_NAME.....	26

4.3.1.2. @VERSION.....	27
4.3.1.3. @MEASUREMENT_STATUS.....	27
4.3.1.4. @FIX_SET_STATUS.....	29
4.3.1.5. @LASTMEAS_PRG_SET_STATUS.....	30
4.3.1.6. PRG_SET_STATUS .....	32
5. ThermoStat プロバイダエラーコード .....	35
6. 付録 .....	36

## 1. はじめに

本書は、日立アプライアンス株式会社製の日立環境試験装置のイーサネット通信オプション基盤とのネットワーク通信によりデータ受信を行う HITACHI ThermoStat プロバイダのユーザーズガイドです。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 ThermoStat プロバイダと呼称します。

ThermoStat プロバイダは、日立アプライアンス株式会社作成の「イーサネット通信オプション (90R04187) . pdf」に記述されている情報を元に作成されており、TCP/IP 通信で日立環境試験装置と接続します。以降日立環境試験装置を試験装置と呼称します。

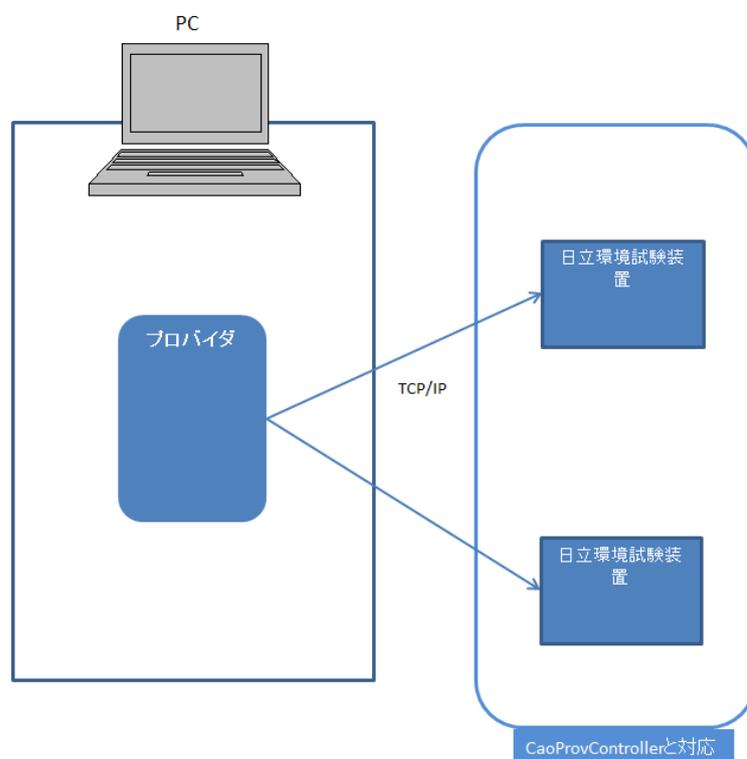


図 1-1構成図

また、本プロバイダ及びデバイスそれぞれの対応を図 1-2に表します。

(※一例です。全てを表しているわけではありません。)



図 1-2 プロバイダの構成とデバイス情報との対応図

### 1.1. 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、対象とする試験装置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\_<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。

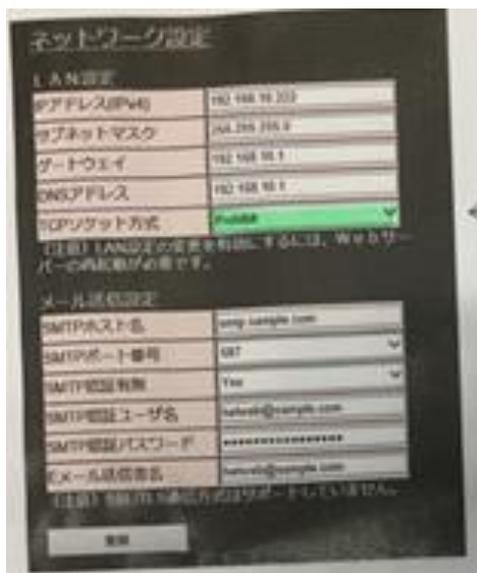
プロバイダを使ったアプリケーションを開発する上で必要となる ORiN2, COM/DCOM の基礎知識や技術に関して例を交えながら解説されています。

## 2. アプリケーション開発のための環境セットアップ

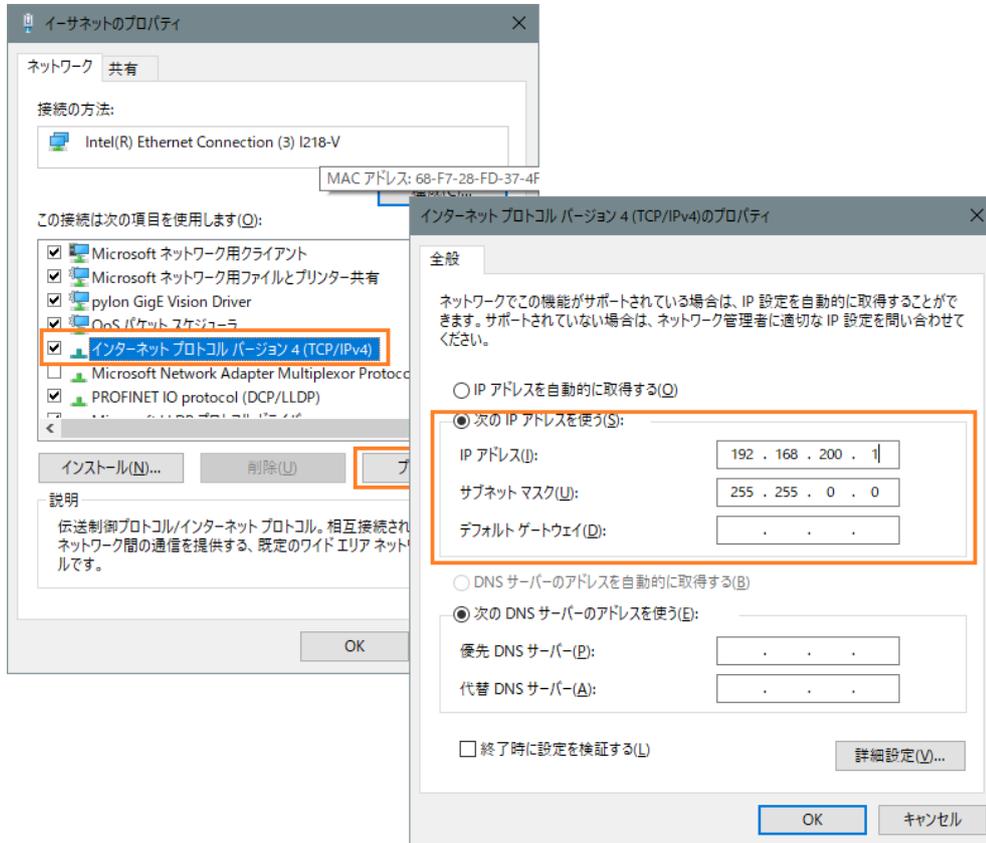
### 2.1. 日立環境試験装置とクライアント PC との接続

日立環境試験装置とクライアント PC を TCP/IP で接続する手順について記述します。

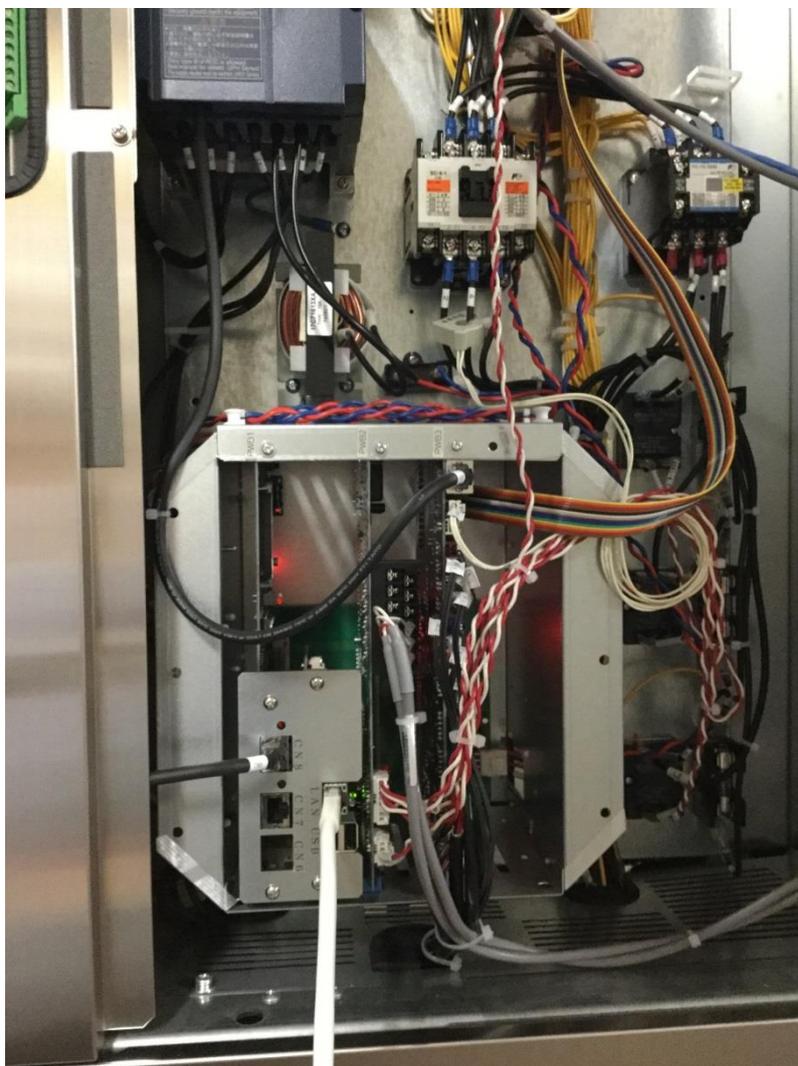
1. 「環境試験装置 Web インターフェースクイックスタートガイド」に記載されている [LAN 設定]  
→ [TCP ソケット方式] に Allow を入力して再起動してください。



2. クライアント PC 上で、使用したい通信ポートのプロパティから、[インターネットプロトコルバージョン 4 (TCP/IPv4)]→[プロパティ]ボタンを実行し、日立環境試験装置の通信設定に合わせて IP アドレス、サブネットマスクの設定を行います。



3. 日立環境試験装置の右側面を開き，下図のように LAN ポートに LAN ケーブル一端を接続し，もう一端をクライアント PC の LAN ポートに接続します。



## 2.2. PC 開発環境のセットアップ

### 2.2.1. ThermoStat プロバイダの自動インストール

ORiN2 SDK インストールされている環境であれば，試験装置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ください。

### 2.2.2. ThermoStat プロバイダの手動インストール

ThermoStat プロバイダの手動インストールを行う場合は下記レジストリ登録を行う必要があります。レジストリ登録を行う場合は，管理者権限でコマンドプロンプトを起動し，regsvr32 コマンド

を実行してください。コマンドを実行する際は絶対パスで指定するか、ファイルがあるフォルダまで移動して実行してください。

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 ThermoStatプロバイダ

ファイル名	CaoProvHITACHIThermoStat.dll
ProgID	CaoProv.HITACHI.ThermoStat
レジストリ登録	regsvr32 CaoProvHITACHIThermoStat.dll
レジストリ登録の抹消	regsvr32 /u CaoProvHITACHIThermoStat.dll

### 3. ThermoStat プロバイダによるプログラミング

ThermoStat プロバイダでは、以下の手順でクライアント PC と試験装置を接続することができます。

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

試験装置に接続した後は、CaoController の Execute メソッドを使用する、もしくは、CaoVariable オブジェクトを生成することで、試験装置の情報にアクセスすることができます。

#### 3.1. 日立環境試験装置の測定を受信するサンプルプログラミング

ここでは例として測定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

要件	説明
接続先	TCP/IP で接続する
	接続先 IP アドレスは 192.168.1.2
	接続先の通信基盤はイーサネット通信オプション基板
	接続先ポート番号は 50000
処理内容	測定を受信する

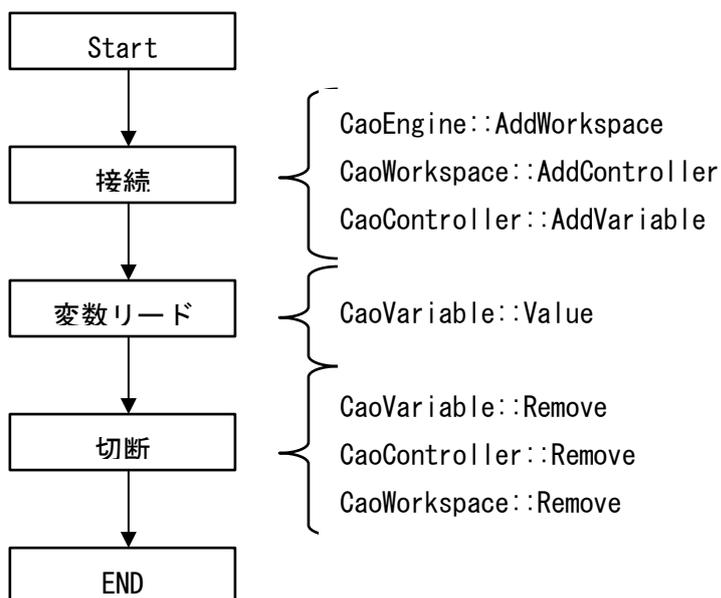


図 3-1 測定受信の流れ

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	ReceiveMeasurement.vb
Sub Main	
' オブジェクト	
Dim engine As CaoEngine	
Dim workspace As CaoWorkspace	
Dim controller As CaoController	
Dim var As CaoVariable	
' 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("ThermoStat", _	
"CaoProv.HITACHI.ThermoStat", _	
""	
, _	
"conn=eth:192.168.1.2")	
' 変数追加	
Set var = controller.AddVariable("@MEASUREMENT_STATUS")	
' 値取得	
Dim values As Variant	

values = var.value

' プログラム番号

Dim programNo As Long

programNo = values(0)

' ステップ番号

Dim stepNo As Long

stepNo = values(1)

' 測定温度

Dim temp As Single

temp = values(2)

' 設定繰り返し数

Dim repeatSetNum As Long

repeatSetNum = values(3)

' 残り時間(分)

Dim minTime As Long

minTime = values(4)

' 残り時間(秒)

Dim secTime As Long

secTime = values(5)

' 設定湿度

Dim setHum As Long

setHum = values(6)

' 戻りステップ番号

Dim returnStepNo As Long

returnStepNo = values(7)

' タイムシグナル

Dim timeSignal As Long

timeSignal = values(8)

' 保持

Dim retention As Long

retention = values(9)

' 槽の状態

Dim tankCondition As Long

tankCondition = values(10)

' アラーム番号

Dim alarmNum As Long

alarmNum = values(11)

' CaoController から変数を削除

Call controller.Variables.Remove(var.Index)

Set var = Nothing

' 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

```

### 3.1.1.1. 接続

試験装置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。CaoWorkspaceオブジェクトは、デフォルトのCaoWorkspaceを使用する場合には変数を用意する必要はありません。また変数にアクセスするためのCaoVariableオブジェクトも必要になります。以下にVB6でのコード例を示します。

```

Dim engine As CaoEngine      ' CaoEngineオブジェクト用の変数
Dim workspace As CaoWorkspace ' CaoWorkspaceオブジェクト用の変数
Dim controller As CaoController ' CaoControllerオブジェクト用の変数
Dim var As CaoVariable       ' CaoVariableオブジェクト用の変数

```

- (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を示します。

```

' CaoWorkspace オブジェクトの生成
Set workspace = engine.AddWorkspace("NewWrks", "")

```

- (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。ThermoStatプロバイダでは、接続先情報、接続先のオプション基板を指定します。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("ThermoStat", _
                                         "CaoProv.HITACHI.ThermoStat", _
                                         "", _
                                         , _

```

---

```
"conn=eth:192.168.1.2")
```

---

- (5) 測定を受信するためのCaoVariableオブジェクトを生成します。ThermoStatプロバイダで測定を受信する変数は、@MEASUREMENT\_STATUSです。

```
' 変数追加
```

```
Set var = controller.AddVariable("@MEASUREMENT_STATUS")
```

---

### 3.1.1.2. 測定を取得する

測定を取得するためには、@MEASUREMENT\_STATUS 変数の Value プロパティを使用します。詳細は、4.3.1.3 を参照してください。

```
' 値取得
```

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

```
' プログラム番号
```

```
Dim programNo As Long  
programNo = values(0)
```

```
' ステップ番号
```

```
Dim stepNo As Long  
stepNo = values(1)
```

```
' 測定温度
```

```
Dim temp As Single  
temp = values(2)
```

```
' 設定繰り返し数
```

```
Dim repeatSetNum As Long  
repeatSetNum = values(3)
```

```
' 残り時間(分)
```

```
Dim minTime As Long  
minTime = values(4)
```

```
' 残り時間(秒)
```

```
Dim secTime As Long  
secTime = values(5)
```

```
' 設定湿度
```

```
Dim setHum As Long  
setHum = values(6)
```

```
' 戻りステップ番号
```

```
Dim returnStepNo As Long  
returnStepNo = values(7)
```

```
' タイムシグナル
```

```
Dim timeSignal As Long  
timeSignal = values(8)
```

```
' 保持
```

```
Dim retention As Long  
retention = values(9)
```

```
' 槽の状態
```

---

```
Dim tankCondition As Long
tankCondition = values(10)
' アラーム番号
Dim alarmNum As Long
alarmNum = values(11)
```

---

### 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。

---

```
' CaoController から変数を削除
Call controller.Variables.Remove(var.Index)
Set var = Nothing
' 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 <sup>1</sup>	機能	参照
CaoWorkspace			
	Addcontroller	M コントローラに接続	P. 17
CaoController			
	VariableNames	P 接続可能な変数名リストの取得	P. 19
	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. 19
	AddVariable	M 変数オブジェクトの追加	P. 19
	Execute	M 拡張コマンドの実行	P. 20
CaoVariable			
	Value	P 値の取得/設定	P. 26

### 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。ThermoStat 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、該当する試験装置と接続を行います。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. HITACHI. ThermoStat", // 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

<sup>1</sup> 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。

```
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 ' Controlleオブジェクト

Set engine = New CaoEngine
Set workspace = engine.CaoWorkspaces.Item(0)
Set controller = workspace.AddController("ThermoStat", _
    "CaoProv. HITACHI. ThermoStat", _
    "" , _
    "conn=eth:192.168.1.2")
```

### 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=	○	通信パラメータを指定します。指定方法については 4.2.1.1.1 をご参照ください	-----	-----
Timeout=	--	コマンドを送信してから応答が来るまでプロバイダが待機する最大時間(ms)を指定します。	1000 - 65535	1000
Delimiter=	--	デリミタの指定 Conn オプションで ETH を指定した場合は、無視されます。	0:CR+LF 1:CR 2:LF	1
Board=	--	イーサネット(ソケット)通信時の接続先基板の指定。Conn オプションで COM を指定した場合は無視されます。	0:イーサネット通信 オプション基板 1:Web オプション基板	0
Retry=	--	通信失敗時の最大リトライ回数を指定します。	0 - 50	4
RetryInterval=	--	通信失敗時のリトライごとの休止時間(ms)を指定します。	0 - 10000	250
Delay=	--	通信間のディレイ時間(ms)を指定します。	0 -	250

#### 4.2.1.1.1. Conn オプション

前述したとおり、試験装置は TCP/IP による通信に対応しています。本プロバイダはこの2つの通信方式に対応するため、Conn オプションで指定された通信方式で試験装置との接続を試みます。

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。

##### ・TCP/IP で接続する場合

“Conn=ETH:<接続先 IP>[:<接続先ポート>[:<ローカル IP>[:<ローカルポート>]]]”

<接続先 IP> : 接続先 IP アドレスを\*\*\*.\*\*\*.\*\*\*.\*\*\*の形式で指定します。

この項目は必ず指定してください。

<接続先ポート> : 接続先ポート番号を指定します。50000

<ローカル IP> : PC 側の IP アドレスを\*\*\*.\*\*\*.\*\*\*.\*\*\*の形式で指定します。

デフォルト:未指定

<ローカルポート>: デフォルト:未指定

#### 4.2.1.1.2. 接続時の注意点

Web オプション基板の試験装置は通常クライアント PC と接続することができません。接続する前に、日立アプライアンス株式会社作成の「環境試験装置 Web インターフェースクイックスタートガイド」に記載されている[LAN 設定]→[TCP ソケット方式]に Allow を入力して再起動してください。

また、Web オプション基板と接続した場合、データの送受信のたびに通信の接続/切断を繰り返します。頻繁なデータの送受信はクライアント PC のポートが枯渇する可能性がありますので注意してください。

#### 4.2.2. CaoController クラス

##### 4.2.2.1. 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

```
' 変数名リスト取得
```

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

##### 4.2.2.2. 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

```
' 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)
```

##### 4.2.2.3. 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 4.3.1 に示すもののみ使用できます。以下に、AddVariable の仕様を示します。

**書式****CaoVariable AddVariable**

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

**4.2.2.4. Execute メソッド**

ConController の拡張コマンドを実行します。以下に、Execute の仕様を示します。

**書式****Variant Execute**

```
(
    "<拡張コマンド名>",           // 拡張コマンド名
    "<オプション文字列>"         // オプション文字列(省略可能)
)
```

以下に、Execute で指定できる拡張コマンド一覧を示します。使用例は拡張コマンドの詳細で記述しています。

コマンド	説明	参照
GetMeasurementData	測定状態を取得します。	P. 20
GetFixSettings	定値モードの運転設定状態を取得する。	P. 22
GetLastMeasPrgSetttings	直前に取得したプログラムモードでの測定状態に対応するプログラムの運転設定状態を取得します。	P. 22
GetPrgSettings	指定したプログラム番号、ステップ番号の運転設定を取得します。	P. 24

**4.2.2.4.1. GetMeasurementData コマンド**

測定状態を取得します。

項目	型説明	
引数	VT_I4	受信データのモード異常 (0x80110007) が発生した際のリトライ回数を指定します。省略可能です。 値範囲：0 - 50 デフォルト値：4
戻り値	VT_ARRAY   VT_VARIANT	

項目	型説明		
	0	VT_I4	プログラム番号が返却されます。
	1	VT_I4	ステップ番号が返却されます。
	2	VT_R4	測定温度が返却されます。
	3	VT_I4	残り繰り返し数が返却されます。
	4	VT_I4	残り時間(分)が返却されます。
	5	VT_I4	残り時間(秒)が返却されます。
	6	VT_I4	測定湿度が返却されます。
	7	VT_I4	戻りステップ番号が返却されます。
	8	VT_I4	タイムシグナルが返却されます。
	9	VT_I4	保持が返却されます。
	10	VT_I4	槽の状態が返却されます。 STP:0, RUN=1, ALARM=2
	11	VT_I4	アラーム番号が返却されます。

### 使用例

' GetMeasurementData実行

```
Dim val As Variant
val = controller.Execute("GetMeasurementData")
```

' プログラム番号

```
Dim programNo As Long
programNo = val(0)
```

' ステップ番号

```
Dim stepNo As Long
stepNo = val(1)
```

' 測定温度

```
Dim temp As Single
temp = val(2)
```

' 設定繰り返し数

```
Dim repeatSetNum As Long
repeatSetNum = val(3)
```

' 残り時間(分)

```
Dim minTime As Long
minTime = val(4)
```

' 残り時間(秒)

```
Dim secTime As Long
secTime = val(5)
```

' 測定湿度

```
Dim setHum As Long
setHum = val(6)
```

## ` 戻りステップ番号

Dim returnStepNo As Long

returnStepNo = val (7)

## ` タイムシグナル

Dim timeSignal As Long

timeSignal = val (8)

## ` 保持

Dim retention As Long

retention = val (9)

## ` 槽の状態

Dim tankCondition As Long

tankCondition = val (10)

## ` アラーム番号

Dim alarmNum As Long

alarmNum = val (11)

**4.2.2.4.2. GetFixSettings コマンド**

定値モードの運転設定状態を取得します。

項目	型説明		
引数	VT_I4		受信データのモード異常 (0x80110007) が発生した際のリトライ回数を指定します。省略可能です。 値範囲: 0 - 50 デフォルト値: 4
戻り値	VT_ARRAY   VT_VARIANT		
	0	VT_R4	設定温度が返却されます。
	1	VT_I4	設定時間(分)が返却されます。
	2	VT_I4	設定時間(秒)が返却されます。
	3	VT_I4	設定湿度が返却されます。
	4	VT_I4	保持が返却されます。
	5	VT_I4	槽の状態が返却されます。 STP:0, RUN=1, ALARM=2
	6	VT_I4	アラーム番号が返却されます。

**使用例**

## ` GetFixSettings実行

Dim val As Variant

val = controller.Execute("GetFixSettings")

**4.2.2.4.3. GetLastMeasPrgSettings コマンド**

直前に取得したプログラムモードでの測定状態に対応するプログラムの運転設定状態を取得します。

直前に取得したプログラムモードでの測定状態がない場合は、プログラム番号:1, ステップ番号:1の運転設定状態を取得します。

項目	型説明		
引数	VT_I4		受信データのモード異常(0x80110007)が発生した際のリトライ回数を指定します。省略可能です。 値範囲: 0 - 50 デフォルト値: 4
戻り値	VT_ARRAY   VT_VARIANT		
	0	VT_I4	プログラム番号が返却されます。
	1	VT_I4	ステップ番号が返却されます。
	2	VT_R4	設定温度が返却されます。
	3	VT_I4	設定繰り返し数が返却されます。
	4	VT_I4	設定時間(分)が返却されます。
	5	VT_I4	設定時間(秒)が返却されます。
	6	VT_I4	設定湿度が返却されます。
	7	VT_I4	戻りステップ番号が返却されます。
	8	VT_I4	タイムシグナルが返却されます。
	9	VT_I4	保持が返却されます。
	10	VT_I4	槽の状態が返却されます。 STP:0, RUN=1, ALARM=2
11	VT_I4	アラーム番号が返却されます。	

#### 使用例

```
' GetLastMeasPrgSettings実行
```

```
Dim val As Variant
```

```
val = controller.Execute("GetLastMeasPrgSettings")
```

```
' プログラム番号
```

```
Dim programNo As Long
```

```
programNo = val(0)
```

```
' ステップ番号
```

```
Dim stepNo As Long
```

```
stepNo = val(1)
```

```
' 設定温度
```

```
Dim temp As Single
```

```
temp = val(2)
```

```
' 設定繰り返し数
```

```
Dim repeatSetNum As Long
```

```
repeatSetNum = val(3)
```

## ' 設定時間(分)

Dim minTime As Long

minTime = val(4)

## ' 設定時間(秒)

Dim secTime As Long

secTime = val(5)

## ' 設定湿度

Dim setHum As Long

setHum = val(6)

## ' 戻りステップ番号

Dim returnStepNo As Long

returnStepNo = val(7)

## ' タイムシグナル

Dim timeSignal As Long

timeSignal = val(8)

## ' 保持

Dim retention As Long

retention = val(9)

## ' 槽の状態

Dim tankCondition As Long

tankCondition = val(10)

## ' アラーム番号

Dim alarmNum As Long

alarmNum = val(11)

## 4.2.2.4.4. GetPrgSettings コマンド

指定したプログラム番号, ステップ番号のの運転設定状態を取得します.

項目	型説明		
引数	VT_I4		プログラム番号を指定します(1以上を指定してください).
	VT_I4		ステップ番号を指定します(1以上を指定してください.) 省略可能です. 省略した場合はデフォルト値1で動作します.
	VT_I4		受信データのモード異常(0x80110007)が発生した際のリトライ回数を指定します. 省略可能です. 値範囲: 0 - 50 デフォルト値: 4
戻り値	VT_ARRAY   VT_VARIANT		
	0	VT_I4	プログラム番号が返却されます.
	1	VT_I4	ステップ番号が返却されます.
	2	VT_R4	設定温度が返却されます.
	3	VT_I4	設定繰り返し数が返却されます.
	4	VT_I4	設定時間(分)が返却されます.

項目	型説明		
	5	VT_I4	設定時間(秒)が返却されます。
	6	VT_I4	設定湿度が返却されます。
	7	VT_I4	戻りステップ番号が返却されます。
	8	VT_I4	タイムシグナルが返却されます。
	9	VT_I4	保持が返却されます。
	10	VT_I4	槽の状態が返却されます。 STP:0, RUN=1, ALARM=2
	11	VT_I4	アラーム番号が返却されます。

### 使用例

```
' GetPrgSettings実行
```

```
Dim val As Variant
```

```
val = controller.Execute("GetPrgSettings", Array(1, 1))
```

```
' プログラム番号
```

```
Dim programNo As Long
```

```
programNo = val(0)
```

```
' ステップ番号
```

```
Dim stepNo As Long
```

```
stepNo = val(1)
```

```
' 設定温度
```

```
Dim temp As Single
```

```
temp = val(2)
```

```
' 設定繰り返し数
```

```
Dim repeatSetNum As Long
```

```
repeatSetNum = val(3)
```

```
' 設定時間(分)
```

```
Dim minTime As Long
```

```
minTime = val(4)
```

```
' 設定時間(秒)
```

```
Dim secTime As Long
```

```
secTime = val(5)
```

```
' 設定湿度
```

```
Dim setHum As Long
```

```
setHum = val(6)
```

```
' 戻りステップ番号
```

```
Dim returnStepNo As Long
```

```
returnStepNo = val(7)
```

```
' タイムシグナル
```

```
Dim timeSignal As Long
```

```
timeSignal = val(8)
```

```
' 保持
```

```
Dim retention As Long
retention = val(9)
' 槽の状態
Dim tankCondition As Long
tankCondition = val(10)
' アラーム番号
Dim alarmNum As Long
alarmNum = val(11)
```

### 4.2.3. CaoVariable クラス

#### 4.2.3.1. Value プロパティ

接続した試験装置からデータを取得/設定します。変数名によって動作が異なります。詳細は、4.3. 変数一覧を参照してください。

### 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. 26
@VERSION	DLL バージョンを取得します。	○	-	P. 27
@MEASUREMENT_STATUS	測定状態を取得します。	○	-	P. 27
@FIX_SET_STATUS	定値モードの運転設定状態を取得します。	○	-	P. 29
@LASTMEAS_PRG_SET_STATUS	直前に取得したプログラムモードでの測定状態に対応するプログラムの運転設定状態を取得します。	○	-	P. 30
PRG_SET_STATUS	指定したプログラム番号, ステップ番号の運転設定を取得します。	○	-	P. 32

##### 4.3.1.1. @MAKER\_NAME

メーカー名の取得をします。

#### 使用例

```
' 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**

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**

```
' 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**

型説明	
VT_BSTR	DLL のバージョンを取得します。 *. *.*

**4.3.1.3. @MEASUREMENT\_STATUS**

測定状態を取得します。

**オプション**

オプション	必須	説明	値範囲	デフォルト値
Retry	-	受信データのモード異常 (0x80110007) が発生した際の、リトライ回数を指定します。	0 - 50	4

**データ型**

型説明			
VT_ARRAY   VT_VARIANT			
	0	VT_I4	プログラム番号.
	1	VT_I4	ステップ番号.
	2	VT_R4	測定温度.
	3	VT_I4	残り繰り返し数.

型説明			
	4	VT_I4	残り時間(分).
	5	VT_I4	残り時間(秒).
	6	VT_I4	測定湿度.
	7	VT_I4	戻りステップ番号.
	8	VT_I4	タイムシグナル.
	9	VT_I4	保持.
	10	VT_I4	槽の状態. 0:STP, 1:RUN, 2:ALARM
	11	VT_I4	アラーム番号.

### 使用例

' 変数追加

Dim var As CaoVariable

Set var = controller.AddVariable("@MEASUREMENT\_STATUS")

' 値取得

Dim values As Variant

values = var.value

' プログラム番号

Dim programNo As Long

programNo = values(0)

' ステップ番号

Dim stepNo As Long

stepNo = values(1)

' 測定温度

Dim temp As Single

temp = values(2)

' 設定繰り返し数

Dim repeatSetNum As Long

repeatSetNum = values(3)

' 残り時間(分)

Dim minTime As Long

minTime = values(4)

' 残り時間(秒)

Dim secTime As Long

secTime = values(5)

' 測定湿度

Dim setHum As Long

setHum = values(6)

' 戻りステップ番号

Dim returnStepNo As Long

returnStepNo = values(7)

' タイムシグナル

```
Dim timeSignal As Long
timeSignal = values(8)
' 保持
Dim retention As Long
retention = values(9)
' 槽の状態
Dim tankCondition As Long
tankCondition = values(10)
' アラーム番号
Dim alarmNum As Long
alarmNum = values(11)
```

#### 4.3.1.4. @FIX\_SET\_STATUS

定値モードの運転状態を取得する。

##### オプション

オプション	必須	説明	値範囲	デフォルト値
Retry	-	受信データのモード異常 (0x80110007) が発生した際の、リトライ回数を指定します。	0 - 50	4

##### データ型

型説明			
VT_ARRAY   VT_VARIANT			
	0	VT_R4	設定温度.
	1	VT_I4	設定時間(分).
	2	VT_I4	設定時間(秒).
	3	VT_I4	設定湿度.
	4	VT_I4	保持.
	5	VT_I4	槽の状態. 0:STP, 1:RUN, 2:ALARM
	6	VT_I4	アラーム番号.

##### 使用例

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

' 値取得
Dim values As Variant
values = var.value
```

## ' 設定温度

Dim temp As Single  
temp = values(0)

## ' 設定時間(時)

Dim minutes As Long  
minutes = values(1)

## ' 設定時間(秒)

Dim seconds As Long  
seconds = values(2)

## ' 設定湿度

Dim humidity As Long  
humidity = values(3)

## ' 保持

Dim retention As Long  
retention = values(4)

## ' 槽の状態

Dim tankCondition As Long  
tankCondition = values(5)

## ' アラーム番号

Dim alarmNum As Long  
alarmNum = values(6)

## 4.3.1.5. @LASTMEAS\_PRG\_SET\_STATUS

直前に取得したプログラムモードでの測定状態に対応するプログラムの運転設定状態を取得します。  
直前に取得したプログラムモードでの測定状態がない場合は、プログラム番号:1, ステップ番号:1  
の運転設定状態を取得します。

## オプション

オプション	必須	説明	値範囲	デフォルト値
Retry	-	受信データのモード異常 (0x80110007) が発生した際の、リトライ回数を指定します。	0 - 50	4

## データ型

型説明			
VT_ARRAY   VT_VARIANT			
	0	VT_I4	プログラム番号.

型説明			
1	VT_I4	ステップ番号.	
2	VT_R4	設定温度.	
3	VT_I4	設定繰り返し数.	
4	VT_I4	設定時間(分).	
5	VT_I4	設定時間(秒).	
6	VT_I4	設定湿度.	
7	VT_I4	戻りステップ番号.	
8	VT_I4	タイムシグナル.	
9	VT_I4	保持.	
10	VT_I4	槽の状態. 0:STP, 1:RUN, 2:ALARM	
11	VT_I4	アラーム番号.	

#### 使用例

```

' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@LASTMEAS_PRG_SET_STATUS")
' 値取得
Dim values As Variant
values = var.value
' プログラム番号
Dim programNo As Long
programNo = values(0)
' ステップ番号
Dim stepNo As Long
stepNo = values(1)
' 設定温度
Dim temp As Single
temp = values(2)
' 設定繰り返し数
Dim repeatSetNum As Long
repeatSetNum = values(3)

' 設定時間(分)
Dim minTime As Long
minTime = values(4)
' 設定時間(秒)
Dim secTime As Long
secTime = values(5)
' 設定湿度
Dim setHum As Long

```

```

setHum = values(6)
' 戻りステップ番号
Dim returnStepNo As Long
returnStepNo = values(7)
' タイムシグナル
Dim timeSignal As Long
timeSignal = values(8)
' 保持
Dim retention As Long
retention = values(9)
' 槽の状態
Dim tankCondition As Long
tankCondition = values(10)
' アラーム番号
Dim alarmNum As Long
alarmNum = values(11)

```

#### 4.3.1.6. PRG\_SET\_STATUS

指定したプログラム番号、ステップ番号の運転設定を取得します。PRG\_SET\_STATUS の後に任意の文字列を入力して変数名としてください(例. PRG\_SET\_STATUS\_1\_1)。

##### オプション

オプション	必須	説明	値範囲	デフォルト値
ProgNo	○	取得するプログラム番号を指定してください。	1 -	1
StepNo	-	取得するステップ番号を指定してください。	1 -	1
Retry	-	受信データのモード異常 (0x80110007) が発生した際の、リトライ回数を指定します。	0 - 50	4

##### データ型

型説明			
VT_ARRAY   VT_VARIANT			
	0	VT_I4	プログラム番号.
	1	VT_I4	ステップ番号.
	2	VT_R4	設定温度.
	3	VT_I4	設定繰り返し数.
	4	VT_I4	設定時間(分).
	5	VT_I4	設定時間(秒).
	6	VT_I4	設定湿度.

型説明			
	7	VT_I4	戻りステップ番号.
	8	VT_I4	タイムシグナル.
	9	VT_I4	保持.
	10	VT_I4	槽の状態. 0:STP, 1:RUN, 2:ALARM
	11	VT_I4	アラーム番号.

### 使用例

' 変数追加

```
Dim var As CaoVariable
```

```
Set var = controller.AddVariable("PRG_SET_STATUS_24_2", "PROGNO=24, STEPNO=2")
```

' 値取得

```
Dim values As Variant
```

```
values = var.value
```

' プログラム番号

```
Dim programNo As Long
```

```
programNo = values(0)
```

' ステップ番号

```
Dim stepNo As Long
```

```
stepNo = values(1)
```

' 設定温度

```
Dim temp As Single
```

```
temp = values(2)
```

' 設定繰り返し数

```
Dim repeatSetNum As Long
```

```
repeatSetNum = values(3)
```

' 設定時間(分)

```
Dim minTime As Long
```

```
minTime = values(4)
```

' 設定時間(秒)

```
Dim secTime As Long
```

```
secTime = values(5)
```

' 設定湿度

```
Dim setHum As Long
```

```
setHum = values(6)
```

' 戻りステップ番号

```
Dim returnStepNo As Long
```

```
returnStepNo = values(7)
```

' タイムシグナル

```
Dim timeSignal As Long
```

```
timeSignal = values(8)
```

' 保持

```
Dim retention As Long
```

```
retention = values(9)
' 槽の状態
Dim tankCondition As Long
tankCondition = values(10)
' アラーム番号
Dim alarmNum As Long
alarmNum = values(11)
```

---

## 5. ThermoStat 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。
0x80110002	GaoWorkspace::AddControllerのConnオプションの指定に誤りがあります。ConnオプションにEthもしくはCOMを指定してください。
0x80110003	Timeoutオプションが範囲外です。
0x80110004	Delimiter オプションが範囲外です。
0x80110005	Boardオプションが範囲外です。
0x80110006	受信データの数値が未設定です。ヒートショック試験装置の初期化状態の可能性がありますが、製造元にお問い合わせください。
0x80110007	受信したデータのモードが要求したモードと異なります <sup>2</sup> 。頻発する場合はリトライ回数をより大きな値にして指定してください。
0x80110008	RetryIntervalオプションが範囲外です。
0x80110009	Retryオプションが範囲外です。
0x8011000A	想定外のデータを受信しました。対応機種でない可能性があります。製造元にお問い合わせください。
0x8011000B	Delayオプションが範囲外です。
0x8011000C	Prognosticオプションが範囲外です。
0x8011000D	Stepnoオプションが範囲外です。

<sup>2</sup> 発生原因は不明ですが、プログラム状態受信コマンド(CF=P)を送信した際に、プログラムモードにならず、定値モードになる現象が確認されています。

## 6. 付録

### 付録A. 通信プロトコルコマンド対応表

参考にした通信仕様書に記述されているコマンドとプロバイダのコマンドとの対応表を記述します。  
通信コマンド名列の記述は通信仕様書の記述に従っています。

#### CaoController::Execute

コマンド名	通信コマンド名
GetMeasurementData	C2 (測定状態にする)
GetFixSettings	CF=T (定値モードの確認状態にする)
GetLastMeasPrgSetttings	CF=P (プログラムモードの確認状態にする)
GetPrgSettings	C4=プログラム番号 (プログラム番号を指定する) C5=ステップ番号 (ステップ番号を指定する)

#### CaoVariable

変数名	通信コマンド名
@MEASUREMENT_STATUS	C2 (測定状態にする)
@FIX_SET_STATUS	CF=T (定値モードでの確認状態にする)
@LASTMEAS_PRG_SET_STATUS	CF=P (プログラムモードの確認状態にする)
PRG_SET_STATUS	C4=プログラム番号 (プログラム番号を指定する) C5=ステップ番号 (ステップ番号を指定する)