

日立アプライアンス株式会社
ヒートショック試験装置プロバイダ

Version 1.1.3

ユーザーズ ガイド

November 20, 2018

備考：

【改版履歴】

バージョン	日付	内容
1.1.0	2018-09-28	以下の変数を追加 @LASTMEAS_SETTING_STATUS @LASTMEAS_SETTING_STATUS_STRING @LASTMEAS_CHECK_STATUS @LASTMEAS_CHECK_STATUS_STRING 以下の変数にリトライ回数オプションを追加 @MEASUREMENT_STATUS @LASTMEAS_SETTING_STATUS @LASTMEAS_CHECK_STATUS SETTING_STATUS CHECK_STATUS
1.1.1	2018-10-01	CaoController::Execute の各コマンドの引数を省略できなっていた不具合を修正.
1.1.2	2018-10-29	CaoController 削除時にメモリリークが発生する問題を修正.
1.1.3	2018-11-30	SETTING_STATUS, CHECK_STATUS の Pattern オプションの不具合を修正. データ取得のロジックを修正. 文言修正.

【動作確認機器】

機種名	ファームウェアバージョン	注意事項
ES-207LHS	-	Web オプション基板装置の場合は、通信のたびに接続/切断が繰り返されます.

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	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. ヒートショック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. HeatShock プロバイダの手動インストール	9
3. コマンドリファレンス	10
3.1. コマンド一覧	10
3.2. メソッド・プロパティ	10
3.2.1. CaoWorkspace::AddController メソッド	10
3.2.2. CaoController::AddVariable メソッド	11
3.2.3. CaoController::GetVariableNames メソッド	12
3.2.4. CaoController::Execute メソッド	12
3.2.4.1. GetMeasurementStatus	13
3.2.4.2. GetMeasurementStatusString	14
3.2.4.3. GetLastMeasSettingStatus	15
3.2.4.4. GetLastMeasSettingStatusString	16
3.2.4.5. GetSettingStatus	16
3.2.4.6. GetSettingStatusString	17
3.2.4.7. GetLastMeasCheckStatus	17
3.2.4.8. GetLastMeasCheckStatusString	18
3.2.4.9. GetCheckStatus	19
3.2.4.10. GetCheckStatusString	20
3.2.5. CaoVariable::get_Value メソッド	20
3.2.6. CaoVariable::put_Value メソッド	20
3.3. 変数一覧	20
3.3.1. コントローラクラス	20
3.3.1.1. @MAKER_NAME	21
3.3.1.2. @VERSION	22
3.3.1.3. @MEASUREMENT_STATUS	22

3.3.1.4. @MEASUREMENT_STATUS_STRING.....	23
3.3.1.5. @LASTMEAS_SETTING_STATUS.....	23
3.3.1.6. @LASTMEAS_SETTING_STATUS_STRING.....	25
3.3.1.7. @LASTMEAS_CHECK_STATUS.....	25
3.3.1.8. @LASTMEAS_CHECK_STATUS_STRING.....	26
3.3.1.9. SETTING_STATUS.....	26
3.3.1.10. SETTING_STATUS_STRING.....	28
3.3.1.11. CHECK_STATUS.....	28
3.3.1.12. CHECK_STATUS_STRING.....	29
4. 付録.....	31
付録 A. エラーコード.....	31
付録 B. 通信プロトコルコマンド対応表.....	31
付録 C. サンプルプログラム.....	32

1. はじめに

本書は、日立アプライアンス株式会社製のヒートショック試験装置に対してデータの読み込みを行う、HITACHI HeatShock プロバイダのユーザーズガイドです。

HITACHI HeatShock 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 通信でヒートショック試験装置と接続します。以降日立アプライアンス株式会社製のヒートショック試験装置をヒートショック試験装置と呼称し、プロバイダを HeatShock プロバイダと呼称します。

下図 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。

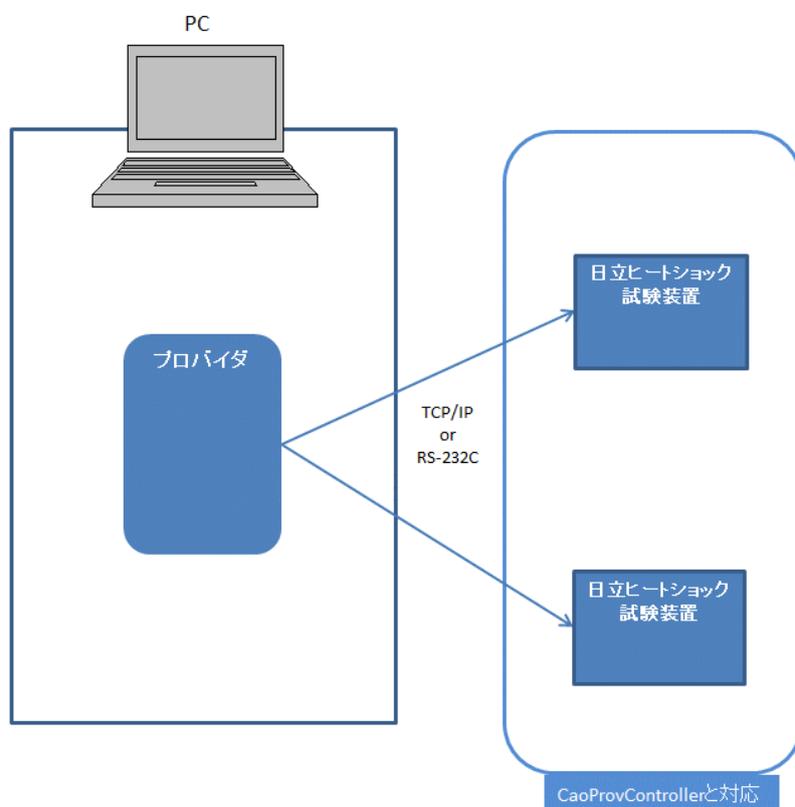


図 1-1 全体構成図

また、本プロバイダ及びデバイスそれぞれの対応を表した図が下図 1-2となります。

(※一例です。全てを表しているわけではありません。)



図 1-2 HeatShock プロバイダと日立ヒートショック試験装置のデータ対応図

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。

「ORiN 2 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. ヒートショック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。

ネットワーク設定

LAN設定

IPアドレス(IPv4)	192.168.10.222
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.10.1
DNSアドレス	192.168.10.1
TCPソケット方式	Prohibit ⇒ Allow

【注意】 LAN設定の変更を有効にするには、Webサーバーの再起動が必要です。

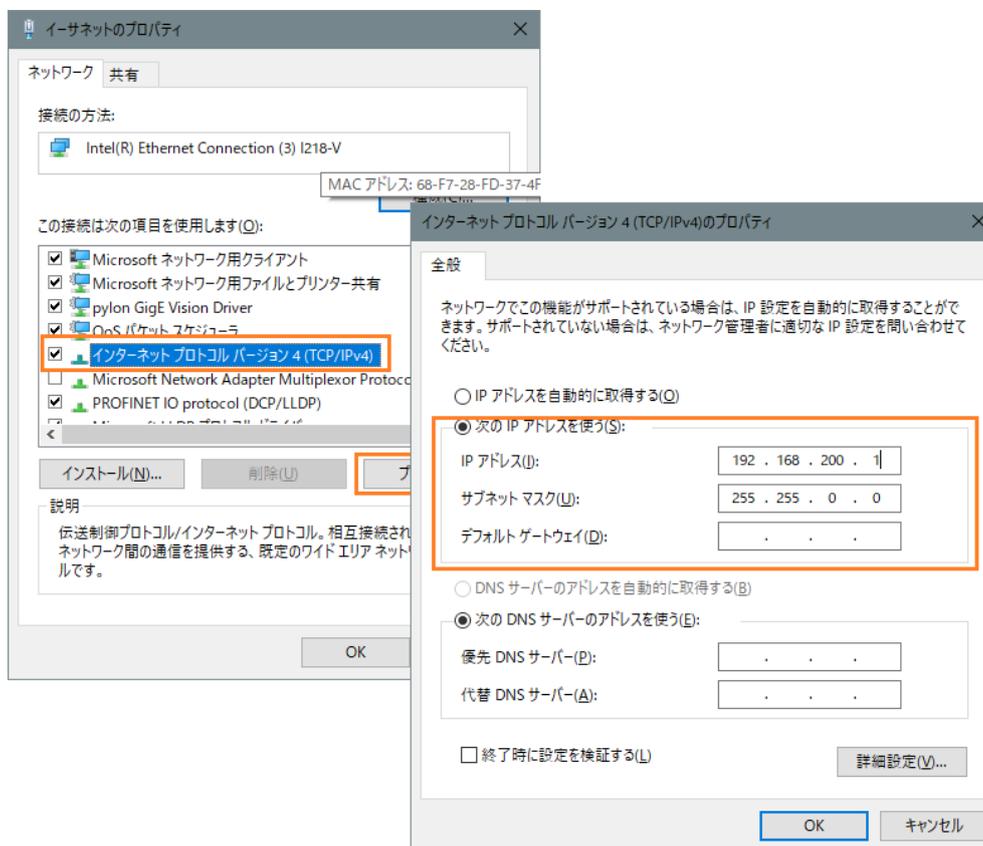
メール送信設定

SMTPホスト名	smtp.sample.com
SMTPポート番号	587
SMTP認証有無	Yes
SMTP認証ユーザ名	hetweb@sample.com
SMTP認証パスワード
Eメール送信者名	hetweb@sample.com

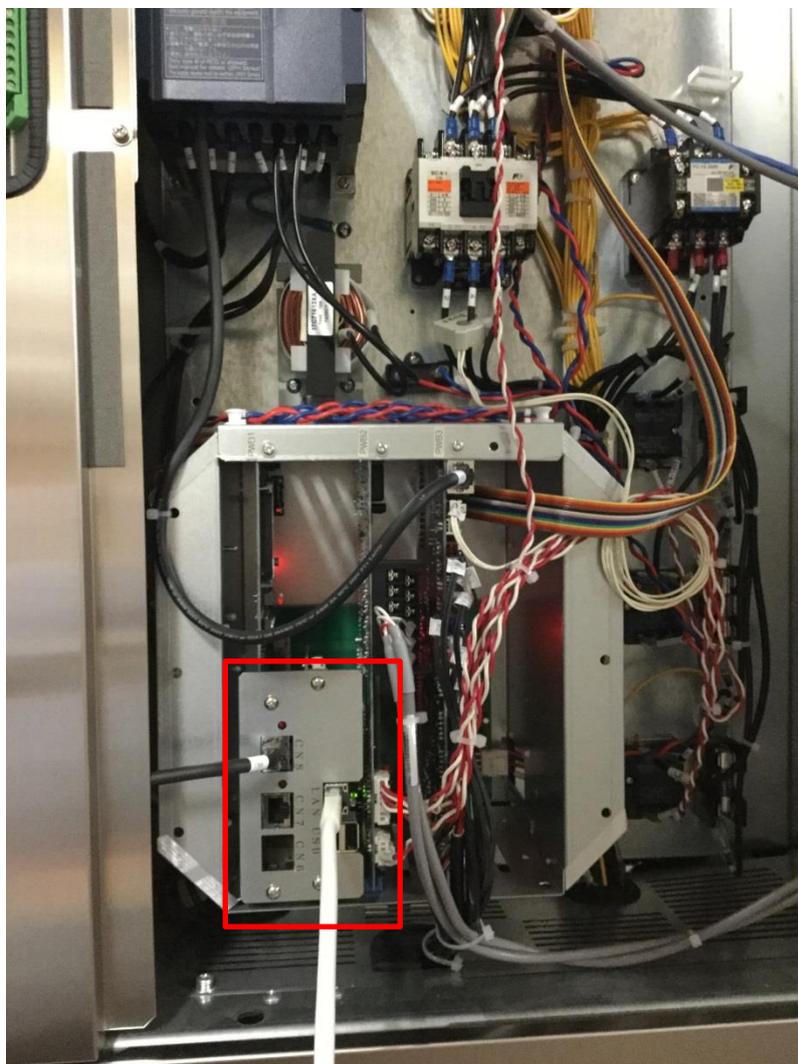
【注意】 SSL/TLS通信方式はサポートしていません。

登録

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、ヒートショック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. ヒートショック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. HeatShock プロバイダの手動インストール

HeatShock プロバイダを手動で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.

表 2-1 HeatShock プロバイダ

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

3. コマンドリファレンス

3.1. コマンド一覧

表 3-1 コマンド一覧

カテゴリ	メソッド/プロパティ	機能	参照
GaoWorkspace	AddController	コントローラに接続	P. 10
CaoController	AddVariable	ユーザ・システム変数に接続	P. 11
	GetVariableNames	ユーザ・システム変数の一覧の取得	P. 12
	Execute	コントローラクラスに実装されているコマンドの実行	P. 12
CaoVariable	get_Value	値の取得	P. 20
	put_Value	値の設定	P. 20

3.2. メソッド・プロパティ

3.2.1. GaoWorkspace::AddController メソッド

Controller オブジェクトの生成時を行います。

各オプションには、デバイスに接続するための情報を指定します。

以下に、AddController の仕様を示します。

書式

AddController

```
(
    "<コントローラ名>",           // コントローラ名(任意)
    "CaoProv. HITACHI. HeatShock", // 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
)
```

以下にオプション文字列に指定する文字列を示します。

表 3-2 GaoWorkspace::AddController のオプション文字列

オプション	必須	説明	値範囲	デフォルト値
CONN= COM:<接続先 COM ポ	○ ¹	接続先の COM ポート番号を環境	-	ボーレート:

¹ Com, ETH のどちらか必須。

オプション	必須	説明	値範囲	デフォルト値
ポート番号>[:<ボーレート>[:<パリティ>:<データビット数>:<ストップビット数>]]		に合わせて指定します。		9600 パリティ: NONE データビット数: 7bit ストップビット数: 1bit
CONN= ETH:<接続先 IP>[:<接続先ポート>]	○ ¹	接続先の IP アドレスおよびポートを環境に合わせて指定します。	-	ポート: 50000
Timeout=<送受信タイムアウト時間>	-	送受信タイムアウト時間 [ms] を設定します。	1000 -	1000
Delimiter=<デリミタ>	-	通信で使用するデリミタを指定します。 ²	0:CR+LF 1:CR 2:LF	1
Board=<通信基板>	-	通信を行う基板を指定します。	0:下記以外 1:Web オプション基板	0
RetryInterval=<リトライ休止時間>	-	通信に失敗した際の再試行するまでの休止時間 (ms) を指定します。	0 - 10000	250
Retry=<リトライ回数>	-	通信に失敗した場合の再試行回数の最大値を指定します。	0 - 50	4
Delay=<送信ディレイ時間>	-	各通信間のディレイ時間 [ms] を設定します。	0 以上	250

使用例

```
AddController("Controller", "CaoProv. HITACHI. HeatShock", "HeatShock", "CONN=ETH:192.168.0.1, Timeout=1000, Delimiter=1, Board=0")
```

3.2.2. CaoController::AddVariable メソッド

CaoController から CaoVariable オブジェクトを生成します。変数名には、表 3-4 に示す変数名の

² CONN オプションで ETH を指定した場合、指定の有無に関わらずデフォルト値が適用されます。

み使用できます。

以下に、AddVariable の仕様を示します。

書式

AddVariable

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

HeatShock プロバイダにはオプションを指定可能な変数はありません。

使用例

```
AddVariable("MEASUREMENT_STATUS", "")  
// 測定状態の変数を追加
```

3.2.3. CaoController::GetVariableNames メソッド

AddVariable メソッドで指定できる変数名とシステム変数名の一覧を取得します。

3.2.4. CaoController::Execute メソッド

Controller にてメソッドを実行します。

引数には、メソッドに対して必要な引数を指定します。

以下に、Execute の仕様を示します。

書式

Execute

```
(  
    "<メソッド名>", // メソッド名  
    "<引数>"       // 引数  
)
```

以下に、Execute で指定できるメソッド一覧を示します。

表 3-3 CaoController::Execute メソッド一覧

メソッド名	説明	参照
GetMeasurementStatus	測定状態を取得します。	P. 13
GetMeasurementStatusString	測定状態を、通信で得られた生値(文字列)取得します。	P. 14
GetLastMeasSettingStatus	直前の測定状態で取得したパターン番号の設定状態を取得します。	P. 15
GetLastMeasSettingStatusString	直前の測定状態で取得したパターン番号の設定状態を、通信で得られた生値(文字列)で取得します。	P. 16
GetSettingStatus	設定状態を取得します。	P. 16
GetSettingStatusString	設定状態を、通信で得られた生値(文字列)で取得します。	P. 17
GetLastMeasCheckStatus	直前の測定状態で取得したパターン番号の確認状態を取得します。	P. 17
GetLastMeasCheckStatusString	直前の測定状態で取得したパターン番号の確認状態を、通信で得られた生値(文字列)で取得します。	P. 18
GetCheckStatus	確認状態を取得します。	P. 19
GetCheckStatusString	確認状態を、通信で得られた生値(文字列)で取得します。	P. 20

3.2.4.1. GetMeasurementStatus

測定状態を取得します。

項目	型説明		
vntParam	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
pVal	VT_ARRAY VT_I4		
	0	VT_I4	パターン番号が返却されます。
	1	VT_I4	試験エリアが返却されます。 0:低温, 1:常温, 2:高温

	2	VT_I4	測定試験室温度が返却されます。
	3	VT_I4	測定予熱温度が返却されます。
	4	VT_I4	測定予冷温度が返却されます。
	5	VT_I4	経過試験時間が返却されます。
	6	VT_I4	経過サイクル数が返却されます。
	7	VT_I4	経過除霜サイクル数が返却されます。
	8	VT_I4	装置状態が返却されます。
	9	VT_I4	準備運転状態が返却されます。 0:準備運転以外, 1:準備運転, 2:準備完了
	10	VT_I4	除霜運転状態が返却されます。 0:除霜運転以外, 1:除霜運転
	11	VT_I4	試験完了状態が返却されます。 0:試験未完, 1:試験完了
	12	VT_I4	扉開放状態が返却されます。 0:扉閉, 1:扉開
	13	VT_I4	アラーム番号が返却されます。

使用例

```
Dim vArray() As Long
vArray = caoController.Execute("GetMeasurementStatus")
```

3.2.4.2. GetMeasurementStatusString

測定状態を、通信で得られた生値(文字列)で取得します。

項目	型説明	
vntParam	なし	
pVal	VT_BSTR	測定状態文字列が返却されます。

使用例

```
Dim vValue As String
vValue = caoController.Execute("GetMeasurementStatusString")
```

3.2.4.3. GetLastMeasSettingStatus

直前の測定状態で取得したパターン番号の設定状態を取得します。一度も測定状態を取得していない場合は、パターン番号 1 の設定状態を取得します。

項目	型説明		
vntParam	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
pVal	VT_ARRAY VT_I4		
	0	VT_I4	パターン番号が返却されます。
	1	VT_I4	試験エリアが返却されます。 0: 低温, 1: 常温, 2: 高温
	2	VT_I4	設定試験室温度が返却されます。
	3	VT_I4	設定予熱温度が返却されます。
	4	VT_I4	設定予冷温度が返却されます。
	5	VT_I4	設定試験時間が返却されます。
	6	VT_I4	設定サイクル数が返却されます。
	7	VT_I4	設定除霜サイクル数が返却されます。
	8	VT_I4	装置状態が返却されます。
	9	VT_I4	準備運転状態が返却されます。 0: 準備運転以外, 1: 準備運転, 2: 準備完了
	10	VT_I4	除霜運転状態が返却されます。 0: 除霜運転以外, 1: 除霜運転
	11	VT_I4	試験完了状態が返却されます。 0: 試験未完, 1: 試験完了
	12	VT_I4	扉開放状態が返却されます。 0: 扉閉, 1: 扉開
13	VT_I4	アラーム番号が返却されます。	

使用例

```
Dim vArray() As Long
vArray = caoController.Execute("GetLastMeasSettingStatus")
```

3.2.4.4. GetLastMeasSettingStatusString

直前の測定状態で取得したパターン番号の設定状態を、通信で得られた生値(文字列)で取得します。一度も測定状態を取得していない場合は、パターン番号1の設定状態を取得します。

項目	型説明	
vntParam	なし	
pVal	VT_BSTR	設定状態文字列が返却されます。

使用例

```
Dim vValue As String
vValue = caoController.Execute("GetLastMeasSettingStatusString")
```

3.2.4.5. GetSettingStatus

設定状態を取得します。

項目	型説明		
vntParam	VT_ARRAY VT_I4		
	0	VT_I4	パターン番号を指定します。
pVal	1	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
	0	VT_I4	パターン番号が返却されます。
	1	VT_I4	試験エリアが返却されます。 0:低温, 1:常温, 2:高温
	2	VT_I4	設定試験室温度が返却されます。
	3	VT_I4	設定予熱温度が返却されます。
	4	VT_I4	設定予冷温度が返却されます。
	5	VT_I4	設定試験時間が返却されます。
	6	VT_I4	設定サイクル数が返却されます。
	7	VT_I4	設定除霜サイクル数が返却されます。
	8	VT_I4	装置状態が返却されます。
9	VT_I4	準備運転状態が返却されます。 0:準備運転以外, 1:準備運転, 2:準備完了	

	10	VT_I4	除霜運転状態が返却されます。 0: 除霜運転以外, 1: 除霜運転
	11	VT_I4	試験完了状態が返却されます。 0: 試験未完, 1: 試験完了
	12	VT_I4	扉開放状態が返却されます。 0: 扉閉, 1: 扉開
	13	VT_I4	アラーム番号が返却されます。

使用例

```
Dim vArray() As Long
vArray = caoController.Execute("GetSettingStatus", 1)
```

3.2.4.6. GetSettingStatusString

設定状態を、通信で得られた生値(文字列)で取得します。

項目	型説明	
vntParam	VT_I4	パターン番号を指定します。
pVal	VT_BSTR	設定状態文字列が返却されます。

使用例

```
Dim vValue As String
vValue = caoController.Execute("GetSettingStatusString", 1)
```

3.2.4.7. GetLastMeasCheckStatus

直前の測定状態で取得したパターン番号の確認状態を取得します。一度も測定状態を取得していない場合は、パターン番号1の確認状態を取得します。

項目	型説明	
vntParam	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
pVal	VT_ARRAY VT_I4	
	0	VT_I4

1	VT_14	試験エリアが返却されます。 0:低温, 1:常温, 2:高温
2	VT_14	設定試験室温度が返却されます。
3	VT_14	設定予熱温度が返却されます。
4	VT_14	設定予冷温度が返却されます。
5	VT_14	設定試験時間が返却されます。
6	VT_14	設定サイクル数が返却されます。
7	VT_14	設定除霜サイクル数が返却されます。
8	VT_14	装置状態が返却されます。
9	VT_14	準備運転状態が返却されます。 0:準備運転以外, 1:準備運転, 2:準備完了
10	VT_14	除霜運転状態が返却されます。 0:除霜運転以外, 1:除霜運転
11	VT_14	試験完了状態が返却されます。 0:試験未完, 1:試験完了
12	VT_14	扉開放状態が返却されます。 0:扉閉, 1:扉開
13	VT_14	アラーム番号が返却されます。

使用例

```
Dim vArray() As Long
vArray = caoController.Execute("GetLastMeasCheckStatus")
```

3.2.4.8. GetLastMeasCheckStatusString

直前の測定状態で取得したパターン番号の確認状態を、通信で得られた生値(文字列)で取得します。一度も測定状態を取得していない場合は、パターン番号1の確認状態を取得します。

項目	型説明	
vntParam	なし	
pVal	VT_BSTR	設定状態文字列が返却されます。

使用例

```
Dim vValue As String
vValue = caoController.Execute("GetLastMeasCheckStatusString")
```

3.2.4.9. GetCheckStatus

確認状態を取得します。

項目	型説明		
vntParam	VT_ARRAY VT_I4		
	0	VT_I4	パターン番号を指定します。
	1	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
pVal	VT_ARRAY VT_I4		
	0	VT_I4	パターン番号が返却されます。
	1	VT_I4	試験エリアが返却されます。 0: 低温, 1: 常温, 2: 高温
	2	VT_I4	設定試験室温度が返却されます。
	3	VT_I4	設定予熱温度が返却されます。
	4	VT_I4	設定予冷温度が返却されます。
	5	VT_I4	設定試験時間が返却されます。
	6	VT_I4	設定サイクル数が返却されます。
	7	VT_I4	設定除霜サイクル数が返却されます。
	8	VT_I4	装置状態が返却されます。
	9	VT_I4	準備運転状態が返却されます。 0: 準備運転以外, 1: 準備運転, 2: 準備完了
	10	VT_I4	除霜運転状態が返却されます。 0: 除霜運転以外, 1: 除霜運転
	11	VT_I4	試験完了状態が返却されます。 0: 試験未完, 1: 試験完了
	12	VT_I4	扉開放状態が返却されます。 0: 扉閉, 1: 扉開
13	VT_I4	アラーム番号が返却されます。	

使用例

```
Dim vArray() As Long
vArray = caoController.Execute("GetCheckStatus", 1)
```

3.2.4.10. GetCheckStatusString

確認状態を、通信で得られた生値(文字列)で取得します。

項目	型説明	
vntParam	VT_I4	パターン番号を指定します。
pVal	VT_BSTR	確認状態文字列が返却されます。

使用例

```
Dim vValue As String
vValue = caoController.Execute("GetCheckStatusString", 1)
```

3.2.5. CaoVariable::get_Value メソッド

オブジェクトに対応している変数の値を取得します。

変数の実装状況およびデータ型は「表 3-4 コントローラクラス 変数一覧」を参照してください。

3.2.6. CaoVariable::put_Value メソッド

オブジェクトに対応している変数に値を設定します。

変数の実装状況およびデータ型は「表 3-4 コントローラクラス 変数一覧」を参照してください。

3.3. 変数一覧

3.3.1. コントローラクラス

CaoController クラスで使用できる変数一覧を以下に示します。

表 3-4 コントローラクラス 変数一覧

変数名	データ型	説明	Value	
			get	put
@MAKER_NAME	VT_BSTR	メーカー名の取得を取得します。	○	—
@VERSION	VT_BSTR	DLL のバージョンの取得します。	○	—
@MEASUREMENT_STATUS	VT_ARRAY VT_I4	測定状態の取得します。	○	—

@MEASUREMENT_STATUS_STRING	VT_BSTR	測定状態を、通信で得られた生値(文字列)の取得します。	○	-
@LASTMEAS_SETTING_STATUS	VT_ARRAY VT_I4	直前の測定状態で取得したパターン番号の設定状態を取得します。	○	-
@LASTMEAS_SETTING_STATUS_STRING	VT_BSTR	直前の測定状態で取得したパターン番号の設定状態を、通信で得られた生値(文字列)で取得します。	○	-
@LASTMEAS_CHECK_STATUS	VT_ARRAY VT_I4	直前の測定状態で取得したパターン番号の確認状態を取得します。	○	-
@LASTMEAS_CHECK_STATUS_STRING	VT_BSTR	直前の測定状態で取得したパターン番号の確認状態を、通信で得られた生値(文字列)で取得します。	○	-
SETTING_STATUS	VT_ARRAY VT_I4	設定状態の取得	○	-
SETTING_STATUS_STRING	VT_BSTR	設定状態を、通信で得られた生値(文字列)の取得します。	○	-
CHECK_STATUS	VT_ARRAY VT_I4	確認状態の取得	○	-
CHECK_STATUS_STRING	VT_BSTR	確認状態を、通信で得られた生値(文字列)の取得します。	○	-

3.3.1.1. @MAKER_NAME

メーカー名の取得をします。

データ型

型説明	
VT_BSTR	メーカー名を取得します。

使用例

```
Dim makerName As CaoVariable
Set makerName = controller.AddVariable("@MAKER_NAME")
```

```
Dim makerNameValue As String
makerNameValue = makerName.Value
```

3.3.1.2. @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

```
Dim version As CaoVariable
Set version = controller.AddVariable("@VERSION")
```

```
Dim versionValue As String
versionValue = version.Value
```

3.3.1.3. @MEASUREMENT_STATUS

測定状態の取得をします。

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_I4			
0	VT_I4	パターン番号が返却されます。	
1	VT_I4	試験エリアが返却されます。 0:低温, 1:常温, 2:高温	
2	VT_I4	測定試験室温度が返却されます。	
3	VT_I4	測定予熱温度が返却されます。	
4	VT_I4	測定予冷温度が返却されます。	
5	VT_I4	経過試験時間が返却されます。	
6	VT_I4	経過サイクル数が返却されます。	
7	VT_I4	経過除霜サイクル数が返却されます。	

8	VT_I4	装置状態が返却されます。
9	VT_I4	準備運転状態が返却されます。 0:準備運転以外, 1:準備運転, 2:準備完了
10	VT_I4	除霜運転状態が返却されます。 0:除霜運転以外, 1:除霜運転
11	VT_I4	試験完了状態が返却されます。 0:試験未完, 1:試験完了
12	VT_I4	扉開放状態が返却されます。 0:扉閉, 1:扉開
13	VT_I4	アラーム番号が返却されます。

使用例

```
Dim measurement As CaoVariable
Set measurement = controller.AddVariable("@MEASUREMENT_STATUS")

Dim measurementValue() As Long
measurementValue = measurement.Value
```

3.3.1.4. @MEASUREMENT_STATUS_STRING**データ型****型説明**

VT_BSTR	測定状態文字列が返却されます。
---------	-----------------

使用例

```
Dim measurementStr As CaoVariable
Set measurementStr = controller.AddVariable("@MEASUREMENT_STATUS_STRING")

Dim measurementValue As String
measurementValue = measurementStr.Value
```

3.3.1.5. @LASTMEAS_SETTING_STATUS

直前の測定状態で取得したパターン番号の設定状態を取得します。一度も測定状態を取得していない場合は、パターン番号1の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
-------	----	----	-----	--------

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_I4			
0	VT_I4	パターン番号が返却されます。	
1	VT_I4	試験エリアが返却されます。 0:低温, 1:常温, 2:高温	
2	VT_I4	設定試験室温度が返却されます。	
3	VT_I4	設定予熱温度が返却されます。	
4	VT_I4	設定予冷温度が返却されます。	
5	VT_I4	設定試験時間が返却されます。	
6	VT_I4	設定サイクル数が返却されます。	
7	VT_I4	設定除霜サイクル数が返却されます。	
8	VT_I4	装置状態が返却されます。	
9	VT_I4	準備運転状態が返却されます。 0:準備運転以外, 1:準備運転, 2:準備完了	
10	VT_I4	除霜運転状態が返却されます。 0:除霜運転以外, 1:除霜運転	
11	VT_I4	試験完了状態が返却されます。 0:試験未完, 1:試験完了	
12	VT_I4	扉開放状態が返却されます。 0:扉閉, 1:扉開	
13	VT_I4	アラーム番号が返却されます。	

使用例

```
Dim lastMeasSetting As CaoVariable
Set lastMeasSetting = controller.AddVariable("@LASTMEAS_SETTING_STATUS")

Dim lastMeasSettingValue() As Long
lastMeasSettingValue = lastMeasSetting.Value
```

3.3.1.6. @LASTMEAS_SETTING_STATUS_STRING

直前の測定状態で取得したパターン番号の設定状態を、通信で得られた生値(文字列)で取得します。一度も測定状態を取得していない場合は、パターン番号1の設定状態を取得します。

データ型

型説明	
VT_BSTR	設定状態文字列が返却されます。

使用例

```
Dim lastMeasSettingStr As CaoVariable
Set lastMeasSettingStr = controller.AddVariable("@LASTMEAS_SETTING_STATUS_STRING")

Dim lastMeasSettingValue As String
lastMeasSettingValue = lastMeasSettingStr.Value
```

3.3.1.7. @LASTMEAS_CHECK_STATUS

直前の測定状態で取得したパターン番号の確認状態を取得します。一度も測定状態を取得していない場合は、パターン番号1の確認状態を取得します。

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_I4			
0	VT_I4	パターン番号が返却されます。	
1	VT_I4	試験エリアが返却されます。 0:低温, 1:常温, 2:高温	
2	VT_I4	設定試験室温度が返却されます。	
3	VT_I4	設定予熱温度が返却されます。	
4	VT_I4	設定予冷温度が返却されます。	
5	VT_I4	設定試験時間が返却されます。	
6	VT_I4	設定サイクル数が返却されます。	

7	VT_I4	設定除霜サイクル数が返却されます。
8	VT_I4	装置状態が返却されます。
9	VT_I4	準備運転状態が返却されます。 0: 準備運転以外, 1: 準備運転, 2: 準備完了
10	VT_I4	除霜運転状態が返却されます。 0: 除霜運転以外, 1: 除霜運転
11	VT_I4	試験完了状態が返却されます。 0: 試験未完, 1: 試験完了
12	VT_I4	扉開放状態が返却されます。 0: 扉閉, 1: 扉開
13	VT_I4	アラーム番号が返却されます。

使用例

```
Dim lastMeasCheck As CaoVariable
Set lastMeasCheck = controller.AddVariable("@LASTMEAS_CHECK_STATUS")

Dim lastMeasCheckValue() As Long
lastMeasCheckValue = lastMeasCheck.Value
```

3.3.1.8. @LASTMEAS_CHECK_STATUS_STRING

直前の測定状態で取得したパターン番号の確認状態を、通信で得られた生値(文字列)で取得します。一度も測定状態を取得していない場合は、パターン番号1の確認状態を取得します。

データ型

型説明	
VT_BSTR	設定状態文字列が返却されます。

使用例

```
Dim lastMeasCheckStr As CaoVariable
Set lastMeasCheckStr = controller.AddVariable("@LASTMEAS_CHECK_STATUS_STRING")

Dim lastMeasCheckValue As String
lastMeasCheckValue = lastMeasCheckStr.Value
```

3.3.1.9. SETTING_STATUS

設定状態の取得をします。

オプション

オプション	必須	説明	値範囲	デフォルト値
Pattern	○	取得するパターン番号を指定します。	1 - 2147483647	-
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_I4			
0	VT_I4	パターン番号が返却されます。	
1	VT_I4	試験エリアが返却されます。 0:低温, 1:常温, 2:高温	
2	VT_I4	設定試験室温度が返却されます。	
3	VT_I4	設定予熱温度が返却されます。	
4	VT_I4	設定予冷温度が返却されます。	
5	VT_I4	設定試験時間が返却されます。	
6	VT_I4	設定サイクル数が返却されます。	
7	VT_I4	設定除霜サイクル数が返却されます。	
8	VT_I4	装置状態が返却されます。	
9	VT_I4	準備運転状態が返却されます。 0:準備運転以外, 1:準備運転, 2:準備完了	
10	VT_I4	除霜運転状態が返却されます。 0:除霜運転以外, 1:除霜運転	
11	VT_I4	試験完了状態が返却されます。 0:試験未完, 1:試験完了	
12	VT_I4	扉開放状態が返却されます。 0:扉閉, 1:扉開	
13	VT_I4	アラーム番号が返却されます。	

使用例

```
Dim setting As CaoVariable
```

```
Set setting = controller.AddVariable("SETTING_STATUS_1", "Pattern=1")
```

```
Dim settingValue() As Long
```

```
settingValue = setting.Value
```

3.3.1.10. SETTING_STATUS_STRING

設定状態(文字列)の取得をします。

オプション

オプション	必須	説明	値範囲	デフォルト値
Pattern	○	取得するパターン番号を指定します。	1 - 2147483647	-

データ型

型説明	
VT_BSTR	設定状態文字列が返却されます。

使用例

```
Dim settingStr As CaoVariable
```

```
Set settingStr = controller.AddVariable("SETTING_STATUS_STRING_1", "Pattern=1")
```

```
Dim settingValue As String
```

```
settingValue = settingStr.Value
```

3.3.1.11. CHECK_STATUS

確認状態の取得をします。

オプション

オプション	必須	説明	値範囲	デフォルト値
Pattern	○	取得するパターン番号を指定します。	1 - 2147483647	-
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_I4	

0	VT_14	パターン番号が返却されます。
1	VT_14	試験エリアが返却されます。 0:低温, 1:常温, 2:高温
2	VT_14	設定試験室温度が返却されます。
3	VT_14	設定予熱温度が返却されます。
4	VT_14	設定予冷温度が返却されます。
5	VT_14	設定試験時間が返却されます。
6	VT_14	設定サイクル数が返却されます。
7	VT_14	設定除霜サイクル数が返却されます。
8	VT_14	装置状態が返却されます。
9	VT_14	準備運転状態が返却されます。 0:準備運転以外, 1:準備運転, 2:準備完了
10	VT_14	除霜運転状態が返却されます。 0:除霜運転以外, 1:除霜運転
11	VT_14	試験完了状態が返却されます。 0:試験未完, 1:試験完了
12	VT_14	扉開放状態が返却されます。 0:扉閉, 1:扉開
13	VT_14	アラーム番号が返却されます。

使用例

```
Dim checkStatus As CaoVariable
```

```
Set checkStatus = controller.AddVariable("CHECK_STATUS_1", "Pattern=1")
```

```
Dim checkStatusValue() As Long
```

```
checkStatusValue = checkStatus.Value
```

3.3.1.12. CHECK_STATUS_STRING

確認状態(文字列)の取得をします。

オプション

オプション	必須	説明	値範囲	デフォルト値
Pattern	○	取得するパターン番号を指定します。	1 - 2147483647	-

データ型

型説明

VT_BSTR	確認状態文字列が返却されます。
---------	-----------------

使用例

```
Dim checkStr As CaoVariable
Set checkStr = controller.AddVariable("CHECK_STATUS_STRING_1", "Pattern=1")

Dim checkStrValue As String
checkStrValue = checkStr.Value
```

4. 付録

付録A. エラーコード

本プロバイダには、独自のエラーコードが存在します。詳細は以下の表 4-1 独自エラーコード表をご参照ください。

ORiN2 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ご参照ください。

表 4-1 独自エラーコード表

エラー番号	説明
0x80110001	必須オプションが存在しません。
0x80110002	Connオプションの指定に誤りがあります。
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。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 ³ 。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	受信データが想定外です。イーサネット通信
0x8011000B	Delayオプションが範囲外です。
0x8011000C	Patternオプションが範囲外です。

付録B. 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
GetMeasurementStatus	C2(測定状態にする)
GetMeasurementStatusString	

³ 発生原因は不明ですが、設定状態取得コマンド(C3)、確認状態取得コマンド(CD)を送信した際に、測定状態データを受信する現象が発生することが確認されています。

GetLastMeasSettingStatus	C3 (設定状態にする)
GetLastMeasSettingStatusString	C4=パターン番号 (パターン番号を設定する)
GetSettingStatus	
GetSettingStatusString	
GetLastMeasCheckStatus	CD (確認状態にする)
GetLastMeasCheckStatusString	C4=パターン番号 (パターン番号を設定する)
GetCheckStatus	
GetCheckStatusString	

CaoVariable

変数名	通信コマンド名
@MEASUREMENT_STATUS	C2 (測定状態にする)
@MEASUREMENT_STATUS_STRING	
@LASTMEAS_SETTING_STATUS	C3 (設定状態にする)
@LASTMEAS_SETTING_STATUS_STRING	C4=パターン番号 (パターン番号を設定する)
SETTING_STATUS	
SETTING_STATUS_STRING	
@LASTMEAS_CHECK_STATUS	CD (確認状態にする)
@LASTMEAS_CHECK_STATUS_STRING	C4=パターン番号 (パターン番号を設定する)
CHECK_STATUS	
CHECK_STATUS_STRING	

付録C. サンプルプログラム

デバイスとデータ通信を行うサンプルを記述します。

Sample**Sample.pcs**

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("ヒートショック", _  
    "CaoProv.HITACHI.HeatShock", _  
    "", _  
    "conn=eth:10.4.122.125, timeout=15000, Board=1")
```

' 測定状態変数作成

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

' 測定状態を取得

```
Dim Value() As Long
```

```
Value = caoVariable.Value
```

' 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
```