

エスペック株式会社  
espec ThermalShock プロバイダ

Version 1.0.0

ユーザーズ ガイド

September 24, 2019

備考：

**【改版履歴】**

バージョン	日付	内容
1.0.0	2019-09-24	初版.

**【動作確認機種】**

機種	バージョン	注意事項
TSA-71L-A	F2BTCCN10.11STD	
TSA-73ES-W (TSA-**3 シリーズ)	----	

## 目次

1. はじめに.....	6
1.1. 参考となる情報源.....	7
2. アプリケーション開発のための環境セットアップ.....	8
2.1. 熱衝撃装置とクライアント PC との接続.....	8
2.1.1. シリアル通信で接続.....	8
2.1.2. Ethernet 通信で接続.....	8
2.2. クライアント PC 開発環境のセットアップ.....	9
2.2.1. espec ThermalShock 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. メソッド/プロパティ一覧.....	10
3.2. メソッド・プロパティ.....	10
3.2.1. CaoWorkspace クラス.....	10
3.2.1.1. AddController メソッド.....	10
3.2.2. CaoController クラス.....	12
3.2.2.1. VariableNames プロパティ.....	12
3.2.2.2. AddExtension メソッド.....	12
3.2.2.3. AddVariable メソッド.....	13
3.2.2.4. Execute メソッド.....	13
3.2.3. CaoExtension クラス.....	14
3.2.3.1. ID プロパティ.....	14
3.2.3.2. VariableNames プロパティ.....	14
3.2.3.3. AddVariable メソッド.....	14
3.2.3.4. Execute メソッド.....	15
3.2.4. CaoVariable クラス.....	15
3.2.4.1. Value プロパティ.....	15
3.3. Execute メソッド一覧.....	15
3.3.1. CaoController クラスコマンド.....	15
3.3.1.1. 機器設定関連コマンド.....	18
3.3.1.2. 機器の運転関連コマンド.....	22
3.3.1.3. 試験パターン情報関連コマンド.....	39

3.3.2. CaoExtension クラスコマンド .....	63
3.3.2.1. 機器設定関連コマンド .....	67
3.3.2.2. 機器の運転関連コマンド .....	68
3.3.2.3. 試験パターン情報関連コマンド .....	73
3.4. 変数一覧 .....	85
3.4.1. CaoController クラス変数 .....	85
3.4.1.1. プロバイダ情報関連変数 .....	86
3.4.1.2. 機器設定関連変数 .....	88
3.4.1.3. 機器運転関連変数 .....	91
3.4.2. CaoExtension クラス変数 .....	97
3.4.2.1. プロバイダ情報関連変数 .....	98
3.4.2.2. 機器設定関連変数 .....	98
3.4.2.3. 機器運転関連変数 .....	99
<b>4. espec ThermalShock プロバイダによるプログラミング .....</b>	<b>103</b>
4.1. RS-485 対応デバイスの試験パターンを変更するサンプルプログラミング .....	103
4.1.1. サンプルプログラム .....	104
4.1.1.1. RS-485 対応デバイスと接続 .....	106
4.1.1.2. 変数の取得/設定 .....	107
4.1.1.3. RS-485 対応デバイスと切断 .....	107
4.2. RS-485 対応デバイスの試験パターンをモニタリングするサンプルプログラミング .....	108
4.2.1. サンプルプログラム .....	109
4.2.1.1. RS-485 対応デバイスと接続 .....	111
4.2.1.2. 試験パターンのモニタリング .....	111
4.2.1.3. RS-485 対応デバイスと切断 .....	111
4.3. Ethernet 対応デバイスの試験パターンを変更するサンプルプログラミング .....	111
4.3.1. サンプルプログラム .....	112
4.3.1.1. Ethernet 対応デバイスと接続 .....	114
4.3.1.2. 変数の取得/設定 .....	114
4.3.1.3. Ethernet 対応デバイスと切断 .....	115
4.4. Ethernet 対応デバイスの試験パターンをモニタリングするサンプルプログラミング .....	115
4.4.1. サンプルプログラム .....	116
4.4.1.1. Ethernet 対応デバイスと接続 .....	118
4.4.1.2. 試験パターンのモニタリング .....	118
4.4.1.3. Ethernet 対応デバイスと切断 .....	118
<b>5. espec ThermalShock プロバイダエラーコード .....</b>	<b>119</b>

---

6. 付録 .....	122
付録 A. ThermalShock プロバイダのコマンドとデバイスのコマンドの対応 .....	122
付録 B. ThermalShock プロバイダの変数とデバイスのコマンドの対応 .....	124
付録 C. 対応機種一覧 .....	125

# 1. はじめに

本書は、エスペック株式会社の冷熱衝撃装置の RS-485 対応シリーズ(表 6-3 RS-485 対応機種一覧の機種)と冷熱衝撃装置の Ethernet 対応シリーズ(表 6-4 Ethernet 対応機種一覧の機種)のデバイスに対してデータの取得及び設定をするプロバイダのユーザーズガイドです。以降エスペック株式会社の冷熱衝撃装置の RS-485 対応シリーズを RS-485 対応デバイス、冷熱衝撃装置の Ethernet 対応シリーズを Ethernet 対応デバイスとそれぞれ呼称します。また、RS-485 対応デバイスと Ethernet 対応デバイスを総称して熱衝撃装置と呼称します。図 1-1 が本プロバイダとデバイスの全体構成図になります。以降本プロバイダを espec ThermalShock プロバイダと呼称します。

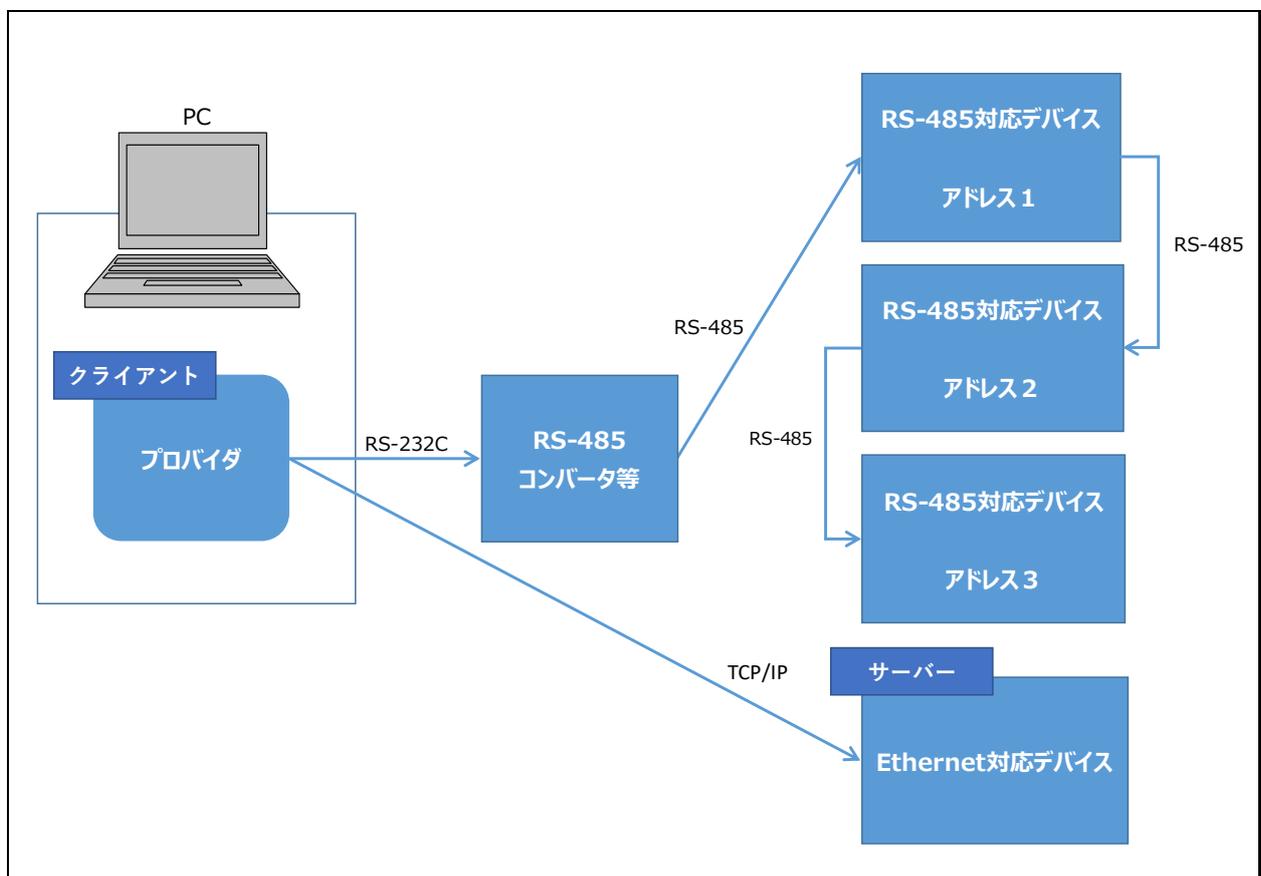


図 1-1 構成図

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。

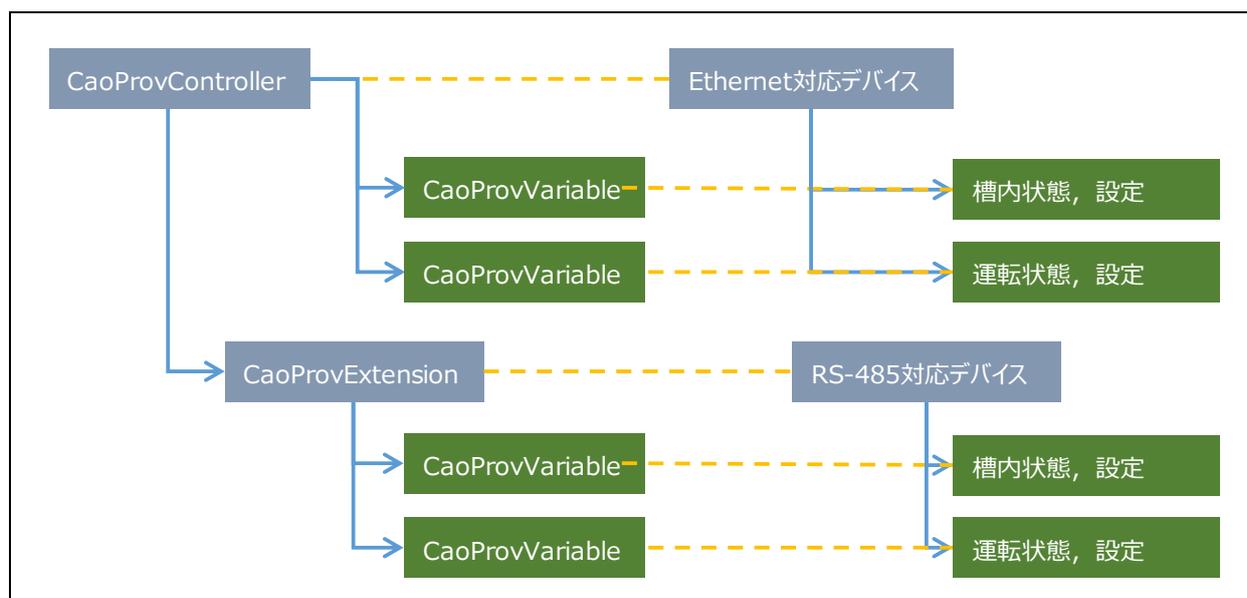


図 1-2 espec ThermalShock プロバイダと熱衝撃装置との対応図

### 1.1. 参考となる情報源

espec ThermalShock プロバイダは、エスペック株式会社から提供されたユーザーズマニュアルを参考に開発しています。以降このマニュアルを ThermalShock RS-485 マニュアルならびに ThermalShock Ethernet マニュアルと呼称します。

以下参考資料

- ユーザーズマニュアル 通信機能 RS-485 (資料番号 4015304029860)
- ユーザーズマニュアル 通信機能 Ethernet (資料番号 4000104007200)

## 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. 熱衝撃装置とクライアント PC との接続

#### 2.1.1. シリアル通信で接続

図 2-1 のように熱衝撃装置 (RS-485 対応デバイス) とクライアント PC は RS-232C を介してシリアル通信にて接続されます。また、データ転送は標準モード及びデリミタは CR+LF で対応しています。接続する RS-485 対応デバイスでは ThermalShock RS-485 マニュアルの「第 2 章 環境設定」を参照して以下の通りに設定して下さい。

設定項目	設定値
RS-485 転送モード設定	標準モード
RS-485 デリミタ設定	CR+LF

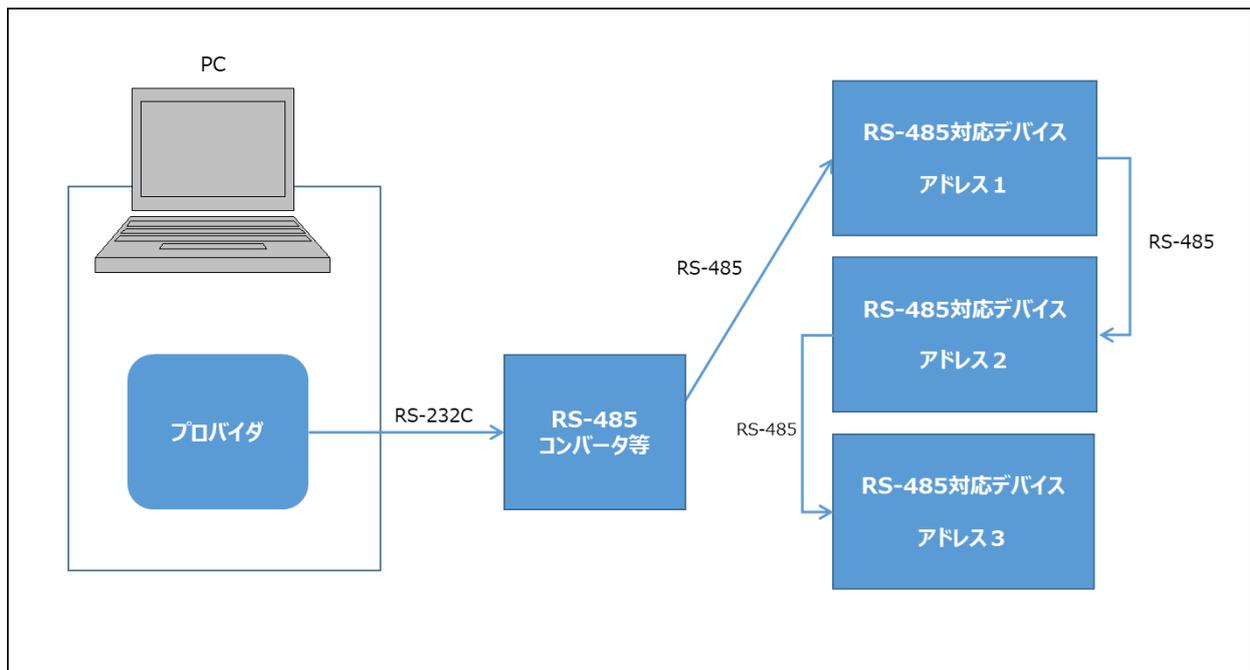


図 2-1 シリアル通信時の構成図

#### 2.1.2. Ethernet 通信で接続

熱衝撃装置 (Ethernet 対応デバイス) とクライアント PC は Ethernet を介して TCP/IP 通信にて接続されます。尚、接続する Ethernet 対応デバイスではデリミタは CR+LF 固定ですので、環境設定で設定する必要はありません。

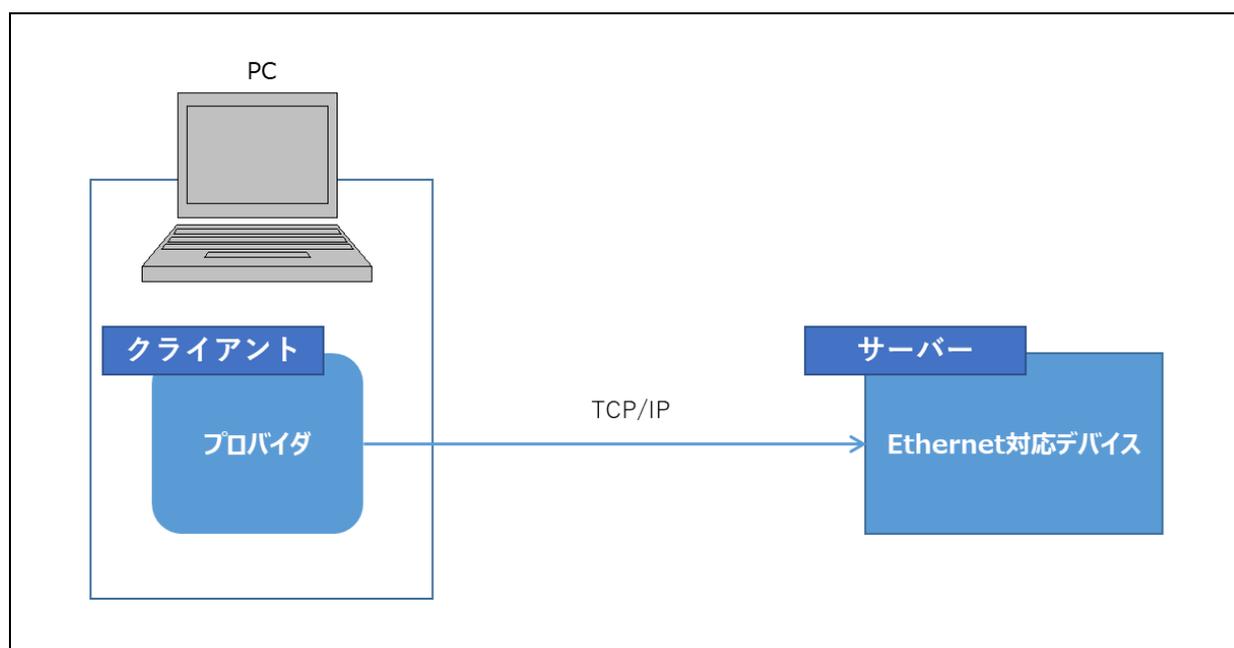


図 2-2 Ethernet 通信時の構成図

## 2.2. クライアント PC 開発環境のセットアップ

### 2.2.1. espec ThermalShock プロバイダの手動インストール

espec ThermalShock プロバイダを使用するために手動でインストールする場合は、下記レジストリ登録を行う必要があります。レジストリ登録を行う場合は、管理者権限でコマンドプロンプトを起動し、regsvr32 コマンドを実行してください。

表 2-1 espec ThermalShock プロバイダ

ファイル名	CaoProvespecThermalShock.dll
ProgID	CaoProv.espec.ThermalShock
レジストリ登録	regsvr32 CaoProvespecThermalShock.dll
レジストリ登録の抹消	regsvr32 /u CaoProvespecThermalShock.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 <sup>1</sup>	機能	参照
<b>CaoWorkspace</b>			
	AddController	M コントローラに接続	P. 10
<b>CaoController</b>			
	VariableNames	P 接続可能な変数名リストの取得	P. 12
	AddExtension	M 拡張ボードオブジェクトの追加	P. 12
	AddVariable	M 変数オブジェクトの追加	P. 13
	Execute	M 拡張コマンドの実行	P. 13
<b>CaoExtension</b>			
	ID	P IDの取得/設定	P. 14
	VariableNames	P 接続可能な変数名リストの取得	P. 14
	AddVariable	M 変数オブジェクトの追加	P. 14
	Execute	M 拡張コマンドの実行	P. 15
<b>CaoVariable</b>			
	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

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 ThermalShock 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

##### AddController

```
(
    "<コントローラ名>", // コントローラ名(任意)
    "CaoProv. espec. ThermalShock", // プロバイダ名(固定)
    "<マシン名>", // プロバイダ実行マシン名
    "<オプション>" // オプション文字列
)
```

<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。

**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	○	熱衝撃装置と接続するためのシリアルまたはイーサネット接続オプションを指定します。詳細は 3.2.1.1.1 を参照して下さい。	--	--
Timeout	--	接続/送受信タイムアウトを ms 単位で指定します。	1 - 4294967295	500

**使用例**

```
Dim caoEng As CaoEngine           ' Engineオブジェクト
Dim caoWor As CaoWorkspace       ' Workspaceオブジェクト
Dim caoCon As CaoController     ' Controllerオブジェクト

Set caoEng = New CaoEngine
Set caoWor = caoEng.Workspaces.Item(0)
Set caoCon = caoWor.AddController("Controller", _
    "CaoProv. espec. ThermalShock", _
    "", _
    "Conn=com:1")
```

**3.2.1.1.1. Conn オプション**

以下に Conn オプションの接続パラメータ文字列を示します。ここで中括弧("[ ]")内は省略可能なことを、各パラメータの解説中の下線部はオプションを指定しなかった時のデフォルト値をそれぞれ示します。

**RS-485 対応デバイスを接続する場合(シリアル通信)**

RS-485 対応デバイスを複数接続する場合(図 1-1 構成図参照)、各デバイスの RS-485 通信速度設定、RS-485 ストップビット設定、RS-485 データビット設定、RS-485 パリティビット設定は同じにして下さい。各デバイスの設定は、ThermalShock RS-485 マニュアルの「第2章 環境設定」及び「第5章 仕様」を参照して下さい。

```
"Conn=Com:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>]]"
```

<COM Port> : COM ポート番号.  
 <BaudRate> : ボーレート. 4800, 9600, 19200  
 <Parity> : パリティ. 'N'-NONE, 'E'-EVEN, 'O'-ODD  
 <DataBits> : データビット. '7'-7bit, '8'-8bit

<StopBits> : ストップビット. '1'-1bit, '2'-2bit

### Ethernet 対応デバイスを接続する場合 (TCP/IP 通信)

Ethernet 対応デバイスは 57732 のポート番号を利用しています.

“Conn=Eth:<IP>[:<Port>]”

<IP> : 接続先 IP アドレス.

<Port> : 接続先ポート. 57732

## 3.2.2. CaoController クラス

### 3.2.2.1. VariableNames プロパティ

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

```
' 接続
Call Connect
' 変数名リスト取得
Dim names as Variant
names = caoCon.VariableNames
' 切断
Call Disconnect
```

### 3.2.2.2. AddExtension メソッド

RS-485 で熱衝撃装置に接続する必要がある場合, CaoController に, 各熱衝撃装置に対応する CaoExtension オブジェクトを追加する必要があります. その際には, オプション文字列に熱衝撃装置に設定されているアドレス番号を指定してください.

以下に, AddExtension メソッドの仕様を示します.

#### 書式

##### AddExtension

```
(
    "<拡張ボード名>", // 拡張ボード名(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	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
Addr	○	接続するデバイスのアドレスを指定します。アドレスは接続されている複数のデバイスを識別するために必要です。各デバイスには一意のアドレスを設定して下さい。(図 1-1 構成図のアドレスを参照)	1 - 16	—

**使用例**

```

' 接続
Call Connect
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
Call caoCon.Extensions.Remove(caoExt.Index)
Set caoExt = Nothing
' 切断
Call Disconnect
    
```

**3.2.2.3. 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 3.4.1 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**

**AddVariable**

```

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

**3.2.2.4. Execute メソッド**

CaoController の拡張コマンドを実行します。Execute で指定できる拡張コマンドについては 3.3.1 に示すコマンドのみ実行可能です。以下に、Execute の仕様を示します。

**書式**

**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)
)
    
```

)

### 3.2.3. CaoExtension クラス

本プロバイダは、シリアル接続された CaoController クラスからのみ本クラスを追加可能です。

#### 3.2.3.1. ID プロパティ

拡張ボードのアドレスを取得します。

##### データ型

型説明	
VT_14	アドレスが返却されます。

##### 使用例

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' ID(アドレス) 取得
Dim address Long
address = caoExt.ID
```

#### 3.2.3.2. VariableNames プロパティ

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

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' 変数名リスト取得
Dim variables as Variant
variables = caoExt.VariableNames
```

#### 3.2.3.3. AddVariable メソッド

CaoExtension に変数オブジェクトを追加します。変数名には 3.4.2 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

##### AddVariable

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

### 3.2.3.4. Execute メソッド

CaoExtension の拡張コマンドを実行します。Execute で指定できる拡張コマンドについては 3.3.2 に示すコマンドのみ実行可能です。以下に、Execute の仕様を示します。

#### 書式

#### 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)
)
```

### 3.2.4. CaoVariable クラス

#### 3.2.4.1. Value プロパティ

接続した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、3.4. 変数一覧を参照してください。

### 3.3. Execute メソッド一覧

各クラスで使用可能なコマンド一覧を定義します。

#### 3.3.1. CaoController クラスコマンド

CaoController クラスで実行する Execute メソッドは、イーサネット接続されている場合のみ有効です。使用例は各コマンドの詳細で記述しています。

表 3-2 CaoController クラスコマンド一覧

コマンド	説明	参照
機器設定関連コマンド		
GetROM	コントローラの ROM バージョンを取得します。	P. 18
GetROMDISP	表示器の ROM バージョンを取得します。	P. 18
GetPOWERCUT	停復電動作モードを取得します。	P. 19
GetSTOPMODE	装置停止モードを取得します。	P. 19
GetDRYTEMP	乾燥運転用設定温度を取得します。	P. 20
GetAMBTIME	常温運転用設定時間を取得します。	P. 20
GetKEYPROTECT	キープロテクト状態を取得します。	P. 21
SetKEYPROTECT	キープロテクト状態を設定します。	P. 21

コマンド	説明	参照
機器の運転関連コマンド		
GetMODE	装置の運転状態を取得します。	P. 22
SetMODE	運転モードを設定します。	P. 23
GetASSIGN	試験パターンを取得します。	P. 25
SetASSIGN	試験パターンを設定します。	P. 25
GetTEMP	さらし温度を取得します。	P. 26
GetPRE	予熱・予冷温度を取得します。	P. 26
GetTIME	さらし残時間を取得します。	P. 27
GetCYCLE	試験サイクル数を取得します。	P. 27
GetMODEL	機種 ID を取得します。	P. 28
GetALARM	アラーム状態を取得します。	P. 28
OPESTANDBY	アイドリング状態にします。	P. 29
OPESETUP	準備運転を開始します。	P. 29
OPESETUPEND	準備後試験運転を開始します。	P. 30
OPETEST	試験を開始します。	P. 30
OPEHALT	試験を中断状態にします。	P. 30
OPERERELEASE	試験の中断状態を解除します。	P. 31
OPECYCLERESET	残サイクルを取り消します。	P. 31
OPEDEFROST	除霜運転をします。	P. 32
OPEDRY	乾燥運転をします。	P. 32
OPEPRESET-SETUP	準備予約タイマーを設定します。	P. 32
OPEPRESET-TEST	試験予約タイマーを設定します。	P. 33
OPEPRESET-ON	タイマー予約をします。	P. 33
OPEPRESET-CLR	タイマー予約を解除します。	P. 34
GetPRESET-HALT-ON	中断予約状況を取得します。	P. 34
OPEPRESET-HALT-ON	中断予約をします。	P. 35
OPEPRESET-HALT-CLR	中断予約を解除します。	P. 36
GetPRESET-HALT	中断予約カウンター情報を取得します。	P. 37
OPEPRESET-HALT	中断予約の詳細を設定します。	P. 38
試験パターン情報関連コマンド		

コマンド	説明	参照
PRGMREAD	参照モードにて試験パターンをオープンします。	P. 40
PRGMREADEND	参照モードでオープンした試験パターンをクローズします。	P. 40
PRGMWRITE	編集モードにて試験パターンをオープンします。	P. 40
PRGMWRITEEND	編集モードでオープンした試験パターンをクローズします。	P. 41
LISTTEMP	さらし温度設定値を取得します。	P. 41
TEMP	さらし温度設定を設定します。	P. 42
LISTPREAI	予熱・予冷モード設定を取得します。	P. 42
PREAI	予熱・予冷モード設定を設定します。	P. 43
LISTPRE	予熱・予冷温度設定を取得します。	P. 43
PRE	予熱・予冷温度設定を設定します。	P. 44
LISTTIME	さらし時間設定値を取得します。	P. 44
TIME	時間設定を設定します。	P. 45
LISTCYCLE	試験サイクル数を取得します。	P. 45
CYCLE	試験サイクル数を設定します。	P. 46
LISTSTARTPOSITION	試験開始位置設定を取得します。	P. 46
STARTPOSITION	試験開始位置を設定します。	P. 47
LISTG-SOAK	安定時間制御設定を取得します。	P. 47
G-SOAK	安定時間制御モードを設定します。	P. 48
LISTEND	試験終了条件設定を取得します。	P. 48
END	終了条件を設定します。	P. 49
LISTNAME	試験パターン名を取得します。	P. 50
NAME	試験パターン名を設定します。	P. 50
LISTSENSOR	センサー位置設定を取得します。	P. 51
SENSOR	センサー位置を設定します。	P. 51
LISTDEFROST	除霜モード設定を取得します。	P. 52
DEFROST	除霜モードを設定します。	P. 52
LISTTEMP-L-DEFROST	除霜待機位置設定を取得します。	P. 53
TEMP-L-DEFROST	除霜待機位置を設定します。	P. 54

コマンド	説明	参照
LISTQ-EXP	さらし時間短縮設定を取得します。	P. 54
Q-EXP	さらし時間短縮モードを設定します。	P. 55
LISTTEMPLIMIT	温度過昇/温度過冷設定値を取得します。	P. 55
TEMPLIMIT	温度過昇/温度過冷設定を設定します。	P. 56
LISTAUXCOOLER	補助冷却設定を取得します。	P. 56
AUXCOOLER	補助冷却モードを設定します。	P. 57
LISTDRYMODESET	前後処理温度・時間設定を取得します。	P. 57
DRYMODESET	前後処理温度・時間設定を設定します。	P. 58
LISTDRYMODE	前後処理モード設定を取得します。	P. 58
DRYMODE	前後処理モードを設定します。	P. 59
LIST-PRE-ECONOMY	エコ運転条件設定を取得します。	P. 60
PRE-ECONOMY	エコ運転条件を設定します。	P. 60
LISTRELAY1	外部出力 1 設定を取得します。	P. 61
RELAY1	外部出力 1 を設定します。	P. 62
LISTRELAY2	外部出力 2 設定を取得します。	P. 62
RELAY2	外部出力 2 を設定します。	P. 63

### 3.3.1.1. 機器設定関連コマンド

#### 3.3.1.1.1. GetROM コマンド

コントローラの ROM バージョンを取得します。

項目	型説明
戻り値	VT_BSTR 装置種別 ROM バージョン

#### 使用例

##### コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

##### GetROMコマンド実行

```
Dim val As String
val = caoCon.Execute("GetROM")
```

#### 3.3.1.1.2. GetROMDISP コマンド

表示器の ROM バージョンを取得します。

項目	型説明	
戻り値	VT_BSTR	装置種別 ROMバージョン

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ GetROMDISPコマンド実行

```
Dim val As String
val = caoCon.Execute("GetROMDISP")
```

3.3.1.1.3. GetPOWERCUT コマンド

停復電動作モードを取得します。

項目	型説明	
戻り値	VT_BSTR	現在の停復電動作モード 設定状態によって以下のいずれかが取得されます。 <ul style="list-style-type: none"> <li>・ STOP : 運転を停止する設定</li> <li>・ CONTINUE : 運転を継続する設定</li> <li>・ DRY : 乾燥運転後, 運転を停止する設定</li> </ul>

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ GetPOWERCUTコマンド実行

```
Dim val As String
val = caoCon.Execute("GetPOWERCUT")
```

3.3.1.1.4. GetSTOPMODE コマンド

装置停止モードを取得します。

項目	型説明	
戻り値	VT_BSTR	現在の装置電源遮断設定状態 設定状態によって以下のいずれかが取得されます。 <ul style="list-style-type: none"> <li>・ ON : 有効</li> <li>・ OFF : 無効</li> </ul>

**使用例**

’ コントローラの追加

Dim caoCon As CaoController

Set caoCon = caoWor.AddController(“Controller”, “CaoProv.espec.ThermalShock”, “”, \_  
“conn=eth:192.168.0.2”)

’ GetSTOPMODEコマンド実行

Dim val As String

val = caoCon.Execute(“GetSTOPMODE”)

### 3.3.1.1.5. GetDRYTEMP コマンド

乾燥運転用設定温度を取得します。

項目	型説明	
戻り値	VT_I4	乾燥運転時の温度設定値

**使用例**

’ コントローラの追加

Dim caoCon As CaoController

Set caoCon = caoWor.AddController(“Controller”, “CaoProv.espec.ThermalShock”, “”, \_  
“conn=eth:192.168.0.2”)

’ GetDRYTEMPコマンド実行

Dim val As Long

val = caoCon.Execute(“GetDRYTEMP”)

### 3.3.1.1.6. GetAMBTIME コマンド

常温運転用設定時間を取得します。

項目	型説明	
戻り値	VT_BSTR	常温運転用設定時間(HH:mm) ・ 熱衝撃装置では、EL タイプ(オプション常温さらし未搭載)の場合、常温運転機能は有していません。 このため、EL タイプでは、本コマンドを使用すると「00:30」(計装内部の初期値)を返しますが、装置は常温運転動作を行いません。

**使用例**

’ コントローラの追加

Dim caoCon As CaoController

Set caoCon = caoWor.AddController(“Controller”, “CaoProv.espec.ThermalShock”, “”, \_  
“conn=eth:192.168.0.2”)

’ GetAMBTIMEコマンド実行

Dim val As String

val = caoCon.Execute(“GetAMBTIME”)

### 3.3.1.1.7. GetKEYPROTECT コマンド

キープロテクト状態を取得します。

項目	型説明	
戻り値	VT_BSTR	キープロテクト状態 設定状態によって以下のいずれかが取得されます。 <ul style="list-style-type: none"> <li>・ ON : データ変更プロテクトもしくは運転変更プロテクトが有効な場合</li> <li>・ OFF : データ変更プロテクト, 運転変更プロテクトのいずれも無効な場合</li> </ul>

#### 使用例

' コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

' GetKEYPROTECTコマンド実行

```
Dim val As String
val = caoCon.Execute("GetKEYPROTECT")
```

### 3.3.1.1.8. SetKEYPROTECT コマンド

キープロテクト状態を設定します。

項目	型説明	
引数	VT_BSTR	キープロテクト状態を指定します。 指定する設定値は以下のようになります。 <ul style="list-style-type: none"> <li>・ ON : 設定変更プロテクト, 運転変更プロテクトが ON になります。</li> <li>・ OFF : 設定変更プロテクト, 運転変更プロテクトが OFF になります。</li> </ul>
戻り値	VT_BSTR	コマンドが正常に実行された場合, 機器からの応答文字列がそのまま格納されます。

#### 使用例

' コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

' SetKEYPROTECTコマンド実行

```
Dim result As String
result = caoCon.Execute("SetKEYPROTECT", "OFF")
```

### 3.3.1.2. 機器の運転関連コマンド

「SetMODE コマンド」は機器の運転状態の変更を行いますが、複合的な運転はできません。一方、コマンド識別子の先頭が「OPE」の場合は、機械の表示機に存在する指令ボタンと同様の動作を行うため、複合的な運転を行わせることができます。

#### 3.3.1.2.1. GetMODE コマンド

現在の装置の運転状態と完了予定日時を取得します。運転状態により、完了予定日時は算出されません。完了予定日時がない場合は、日時はセットされません。運転状態および完了日時の詳細は、「表 3-3 運転状態および完了予定時刻表示一覧」を参照してください。

項目	型説明		
戻り値	VT_ARRAY   VT_BSTR		
	0	VT_BSTR	現在の装置の運転状態
	1	VT_BSTR	完了予定日付 (MM/dd)
	2	VT_BSTR	完了予定時間 (HH:mm)

表 3-3 運転状態および完了予定時刻表示一覧

運転状態	応答データ		TSA シリ ーズ	TSE シリ ーズ	TSB シリ ーズ	TSD シリ ーズ
	運転状態	予定時刻				
受電中	POWER-OFF	(何も返さない)	○	○	○	○
アイドリング中	STANDBY	(何も返さない)	○	○	○	○
液温戻し中	STABLETEMP	液温戻し完了予定時刻	×	×	○	×
回収中	RETRIVING	(何も返さない)	×	×	○	×
常温戻し中	AMB	(何も返さない)	×	×	×	○
手動除霜中	DEFROST	除霜完了予定時刻	○	○	×	○
乾燥運転中	DRY	(何も返さない)	○	×	×	×
準備中	SETUP	準備運転完了予定時刻	○	○	○	○
準備完了	READY	準備完了時刻	○	○	○	○
試験中	TEST	試験終了予定時刻	○	○	○	○
試験中断中	HALT	試験中断時刻	○	○	○	○
試験中断中(常温中)	AMB	試験中断時刻	×	×	×	○
試験中断中(除霜完了待)	HALT	除霜完了予定時刻	○	○	×	○
試験中断 除霜 + 常温中	AMB	試験中断時刻	×	×	×	○
試験中(除霜完了待)	TEST	試験終了予定完了時刻	○	○	×	○
試験終了(準備中)	END-SETUP	試験終了時刻	○	○	○	○

運転状態	応答データ		TSA	TSE	TSB	TSD
	運転状態	予定時刻	シリーズ	シリーズ	シリーズ	シリーズ
試験終了(準備完了)	END-READY	試験終了時刻	○	○	○	○
試験終了(停止中)	END-OFF	試験終了時刻	○	○	○	○
試験終了(乾燥運転中)	END-DRY	試験終了時刻	○	×	×	×
試験終了(常温中)	END-AMB	試験終了時刻	○	×	×	×
試験終了 停止 + 常温中	END-OFF-AMB	試験終了時刻	×	×	×	○
試験終了(除霜中)	END-DEFROST	除霜完了予定時刻	○	○	×	○
試験終了(液温戻し中)	END-STABLETEMP	試験終了時刻	×	×	○	×
試験終了(回収中)	END-REGAIN	試験終了時刻	×	×	○	×
試験終了 除霜 + 常温中	END-DEF-AMB	除霜完了予定時刻	×	×	×	○
準備予約待機中	WAIT-SETUP	試験可能予定時刻	○	○	○	○
試験予約待機中	WAIT-TEST	試験終了予定時刻	○	○	○	○

※○ - 対応      × - 非対応

#### 使用例

##### ’ コントローラの追加

```
Dim caoCon As CaoController
```

```
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _  
"conn=eth:192.168.0.2")
```

##### ’ GetMODEコマンド実行

```
Dim values As Variant
```

```
values = caoCon.Execute("GetMODE")
```

##### ’ 運転状態取得

```
Dim opeState As String
```

```
opeState = values(0)
```

##### ’ 完了予定日付取得

```
Dim day As String
```

```
day = values(1)
```

##### ’ 完了予定時間取得

```
Dim time As String
```

```
time = values(2)
```

### 3.3.1.2.2. SetMODE コマンド

運転モードを設定します。

項目	型説明
----	-----

項目	型説明	
引数	VT_BSTR	運転モードを指定します。指定できる運転モードは単動作のみです。指定可能な運転モードは、「表 3-4 運転モード変更時の送出データ」を参照してください。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

表 3-4 運転モード変更時の送出データ

送出データ	運転モード	TSA シリーズ	TSE シリーズ	TSB シリーズ	TSD シリーズ
OFF	パネル電源 OFF 状態	○	○	○	○
STANDBY	アイドリング状態	○	○	○	○
SETUP	準備運転	○	○	○	○
SETUPEND	準備後試験	○	○	○	○
TEST	試験	○	○	○	○
HALT	試験中断	○	○	○	○
RELEASE	試験中断解除	○	○	○	○
HEATRETURN	液温戻し運転	×	×	○	×
HRET-ON	液温戻し予約	×	×	○	×
HRET-CLR	液温戻し予約解除	×	×	○	×
BOXMOVE-H	試料かご移動要求(左右方向)	×	×	○	×
BOXMOVE-V	試料かご移動要求(上下方向)	×	×	○	×
DEFROST	除霜運転	○	○	×	○
TAREAMOVE	テストエリア移動	×	○	×	○
DRY	乾燥運転	○	×	×	×
AMB-ON	常温戻し運転開始	×	×	×	○
AMB-CLR	常温戻し運転中止	×	×	×	○

※○ - 対応      × - 非対応

**使用例**

’ **コントローラの追加**

Dim caoCon As CaoController

Set caoCon = caoWor.AddController(“Controller”, “CaoProv.espec.ThermalShock”, “”, \_  
“conn=eth:192.168.0.2”)

’ **SetMODEコマンド実行**

Dim result As String

```
result = caoCon.Execute("SetMODE", "OFF")
```

### 3.3.1.2.3. GetASSIGN コマンド

試験パターンを取得します。

項目	型説明	
戻り値	VT_ARRAY   VT_VARIANT	
	0	VT_I4 試験パターンのパターン番号
	1	VT_BSTR 試験パターン名称 最大 14 文字以内で取得されます。

#### 使用例

##### ' コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

##### ' GetASSIGNコマンド実行

```
Dim val As Variant
val = caoCon.Execute("GetASSIGN")
```

##### ' パターン番号取得

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

##### ' 試験パターン名称取得

```
Dim ptrnName As String
ptrnName = val(1)
```

### 3.3.1.2.4. SetASSIGN コマンド

熱衝撃装置の選択中試験パターンを設定します。

項目	型説明	
引数	VT_I4	運転する試験パターンのパターン番号を指定します。 パターン番号は 1~40 の範囲で指定してください。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

##### ' コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ SetASSIGNコマンド実行**

```
Dim result As String
result = caoCon.Execute("SetASSIGN", 1)
```

**3.3.1.2.5. GetTEMP コマンド**

さらし温度を取得します。温度データは、古いものから順にセットされ、温度センサー断線時は 999 がセットされます。

項目	型説明		
戻り値	VT_ARRAY   VT_VARIANT		
	10 秒間隔でサンプリングした温度データ (1 分間のデータとして 6 ケ)		
	0	VT_BSTR	サンプリング開始時刻 (HH:mm)
	1	VT_I4	温度 1
	2	VT_I4	温度 2
	3	VT_I4	温度 3
	4	VT_I4	温度 4
	5	VT_I4	温度 5
6	VT_I4	温度 6	

**使用例**

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ GetTEMPコマンド実行**

```
Dim values As Variant
values = caoCon.Execute("GetTEMP")
```

**’ 最新の温度を取得**

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

**3.3.1.2.6. GetPRE コマンド**

予熱・予冷温度を取得します。

項目	型説明	
戻り値	VT_ARRAY   VT_I4	
	現在の高温槽と低温槽の温度測定値	
	0	VT_I4
1	VT_I4	低温槽測定温度

※温度センサー断線時は 999 がセットされます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ GetPREコマンド実行

```
Dim val() As Long
val = caoCon.Execute("GetPRE")
```

**3.3.1.2.7. GetTIME コマンド**

さらし残時間を取得します。

項目	型説明		
戻り値	VT_ARRAY   VT_BSTR	試験中パターンのさらし状態と残り時間	
	0	VT_BSTR	さらし状態 <ul style="list-style-type: none"> <li>・ さらし状態は以下のように表現されます。                             <ul style="list-style-type: none"> <li>H : 高温さらし運転中</li> <li>L : 低温さらし運転中</li> <li>A : 常温さらし運転中</li> </ul> </li> <li>・ 除霜完了待ち状態の場合は以下のように表現されます。                             <ul style="list-style-type: none"> <li>H : 高温さらし処理中</li> <li>A : 高温さらし処理未実施</li> </ul> </li> </ul>
	1	VT_BSTR	残時間 (HH:mm)

※さらし試験中でない場合は、エラーが返却されます。エラーの詳細に関しては、「表 5-2 デバイスのエラーコード表」のエラーコード[0x8010001F]を参照してください。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ GetTIMEコマンド実行

```
Dim values() As String
values = caoCon.Execute("GetTIME")
```

**3.3.1.2.8. GetCYCLE コマンド**

試験サイクル数を取得します。

項目	型説明	
戻り値	VT_ARRAY   VT_I4	試験中パターンの設定サイクル数と実行サイクル数

項目	型説明		
	0	VT_I4	実行サイクル数
	1	VT_I4	設定サイクル数

※さらし運転中でない場合、および残サイクル数がない場合は、実行サイクル数が「0」となります。

**使用例**

’ **コントローラの追加**

Dim caoCon As CaoController

Set caoCon = caoWor.AddController(“Controller”, “CaoProv.espec.ThermalShock”, “”, \_  
“conn=eth:192.168.0.2”)

’ **GetCYCLEコマンド実行**

Dim values() As Long

values = caoCon.Execute(“GetCYCLE”)

**3.3.1.2.9. GetMODEL コマンド**

機種 ID を取得します。

項目	型説明	
戻り値	VT_BSTR	機種 ID.

**使用例**

’ **コントローラの追加**

Dim caoCon As CaoController

Set caoCon = caoWor.AddController(“Controller”, “CaoProv.espec.ThermalShock”, “”, \_  
“conn=eth:192.168.0.2”)

’ **GetMODELコマンド実行**

Dim val As String

val = caoCon.Execute(“GetMODEL”)

**3.3.1.2.10. GetALARM コマンド**

アラーム状態を取得します。

項目	型説明	
戻り値	VT_EMPTY	発生しているアラームがありません
	または	
	VT_ARRAY   VT_I4	最大 6 個までのアラーム情報
	<i>i</i> VT_I4	アラーム番号 アラーム番号の詳細は <a href="#">ThermalShock Ethernet マニュアル</a> 「表 3.4 警報コード一覧」を参照してください。

※*i* - 発生しているアラーム数分(1~6)

**使用例**

**コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**GetALARMコマンド実行**

```
Dim values As Variant
values = caoCon.Execute("GetALARM")
```

**アラーム数分取得**

```
If Not IsEmpty(values) Then
    Dim i As Long
    For i = 0 To UBound(values)
        Dim val As Long
        val = values(i)
    Next i
End If
```

**3.3.1.2.11. OPESTANDBY コマンド**

装置をアイドル状態にします。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

**コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**OPESTANDBYコマンド実行**

```
Dim result As String
result = caoCon.Execute("OPESTANDBY")
```

**3.3.1.2.12. OPESETUP コマンド**

装置を準備開始状態にします。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ OPESETUPコマンド実行**

```
Dim result As String
result = caoCon.Execute("OPESETUP")
```

### 3.3.1.2.13. OPESETUPEND コマンド

装置を準備後試験開始状態にします。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ OPESETUPENDコマンド実行**

```
Dim result As String
result = caoCon.Execute("OPESETUPEND")
```

### 3.3.1.2.14. OPETEST コマンド

装置を試験開始状態にします。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ OPETESTコマンド実行**

```
Dim result As String
result = caoCon.Execute("OPETEST")
```

### 3.3.1.2.15. OPEHALT コマンド

装置を試験中断状態にします。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ OPEHALTコマンド実行

```
Dim result As String
result = caoCon.Execute("OPEHALT")
```

**3.3.1.2.16. OPERERELEASE コマンド**

試験中断状態を解除します。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ OPERERELEASEコマンド実行

```
Dim result As String
result = caoCon.Execute("OPERERELEASE")
```

**3.3.1.2.17. OPECYCLERESSET コマンド**

残サイクルを消去します。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ OPECYCLERESSETコマンド実行

```
Dim result As String
result = caoCon.Execute("OPECYCLERESET")
```

### 3.3.1.2.18. OPEDEFROST コマンド

除霜運転を開始します。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

##### ' コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

##### ' OPEDEFROSTコマンド実行

```
Dim result As String
result = caoCon.Execute("OPEDEFROST")
```

### 3.3.1.2.19. OPEDRY コマンド

乾燥運転を開始します。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

##### ' コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

##### ' OPEDRYコマンド実行

```
Dim result As String
result = caoCon.Execute("OPEDRY")
```

### 3.3.1.2.20. OPEPRESET-SETUP コマンド

準備予約タイマー（試験準備を完了させる時間）を設定します。

項目	型説明	
引数	VT_BSTR	日時 (MM/dd, HH:mm) を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

**コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**OPEPRESET-SETUPコマンド実行**

```
Dim result As String
result = caoCon.Execute("OPEPRESET-SETUP", "03/05, 19:21")
```

**3.3.1.2.21. OPEPRESET-TEST コマンド**

試験予約タイマー(試験を開始させる時間)を設定します。

項目	型説明	
引数	VT_BSTR	日時(MM/dd, HH:mm)を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

**コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**OPEPRESET-TESTコマンド実行**

```
Dim result As String
result = caoCon.Execute("OPEPRESET-TEST", "03/05, 19:21")
```

**3.3.1.2.22. OPEPRESET-ON コマンド**

設定されたタイマーの予約を有効にします。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

**コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**OPEPRESET-ONコマンド実行**

```
Dim result As String
result = caoCon.Execute("OPEPRESET-ON")
```

### 3.3.1.2.23. OPEPRESET-CLR コマンド

予約されているタイマーを解除します。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

##### ’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
                                "conn=eth:192.168.0.2")
```

##### ’ OPEPRESET-CLRコマンド実行

```
Dim result As String
result = caoCon.Execute("OPEPRESET-CLR")
```

### 3.3.1.2.24. GetPRESET-HALT-ON コマンド

中断予約状況を取得します。

項目	型説明		
戻り値	VT_ARRAY   VT_BSTR		
	0	VT_BSTR	さらし後中断 以下のいずれかが取得されます。 ・ ON : 予約中 ・ OFF : 予約中でない
	1	VT_BSTR	サイクル後中断 さらし後中断と同形式の値が取得されます。
	2	VT_BSTR	カウンター1 さらし後中断と同形式の値が取得されます。
	3	VT_BSTR	カウンター2 さらし後中断と同形式の値が取得されます。
	4	VT_BSTR	カウンター3 さらし後中断と同形式の値が取得されます。
	5	VT_BSTR	カウンター4 さらし後中断と同形式の値が取得されます。
6	VT_BSTR	カウンター5 さらし後中断と同形式の値が取得されます。	

項目	型説明	
	7	VT_BSTR カウンター6 さらし後中断と同形式の値が取得されます。

※カウンター中断機能未搭載の製品 (TSD 等) では, カウンター1~カウンター6 は全て「OFF」が取得されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ GetPRESET-HALT-ONコマンド実行

```
Dim val() As String
val = caoCon.Execute("GetPRESET-HALT-ON")
```

3.3.1.2.25. OPEPRESET-HALT-ON コマンド

試験中断予約設定を行います。このコマンドは引数によって、動作が異なります。以下のいずれかを引数に指定してください。

カウンター中断予約を有効にする場合：

項目	型説明	
	VT_ARRAY   VT_VARIANT	
引数	0	VT_I4 予約 No を指定します。 予約 No は, 1~6 の範囲で指定可能です。
	1	VT_BOOL 繰り返し設定を指定します。省略した場合, デフォルト設定となります。 ・ False : 1 回のみ実行(デフォルト) ・ True : 繰り返し
戻り値	VT_BSTR	コマンドが正常に実行された場合, 機器からの応答文字列がそのまま格納されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ OPEPRESET-HALT-ONコマンド実行

```
Dim result As String
’ 繰り返しなしを設定する場合
Dim param As Long
```

```
param = 1
result = caoCon.Execute("OPEPRESET-HALT-ON", param)
```

' 繰り返しを設定する場合

```
Dim params(1) As Variant
params(0) = CLng(1)
params(1) = CBool(True)
result = caoCon.Execute("OPEPRESET-HALT-ON", params)
```

さらし後中断予約またはサイクル後中断予約を有効にする場合：

項目	型説明	
引数	VT_BSTR	以下のいずれかを指定可能です。 <ul style="list-style-type: none"> <li>・ EXP : さらし後中断予約を開始します</li> <li>・ CYC : サイクル後中断予約を開始します</li> </ul>
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

' コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

' OPEPRESET-HALT-ONコマンド実行

```
Dim result As String
```

' さらし後中断予約を開始

```
result = caoCon.Execute("OPEPRESET-HALT-ON", "EXP")
```

' サイクル後中断予約を開始

```
result = caoCon.Execute("OPEPRESET-HALT-ON", "CYC")
```

**3.3.1.2.26. OPEPRESET-HALT-CLR コマンド**

試験中断予約を行います。このコマンドは、引数の値によって動作が異なりますのでご注意ください。

項目	型説明	
引数	VT_I4	予約 No を指定します。予約 No を指定した場合、カウンター中断予約を解除します。予約 No は、1~6 の範囲で指定可能です。 負の値を指定した場合、サイクル後中断予約、さらし後中断予約を解除します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ OPEPRESET-HALT-CLRコマンド実行

```
Dim result As String
```

’ カウンター中断予約を解除する場合

```
result = caoCon.Execute("OPEPRESET-HALT-CLR", 1)
```

’ サイクル後中断予約とさらし後中断予約を解除する場合

```
result = caoCon.Execute("OPEPRESET-HALT-CLR", -1)
```

3.3.1.2.27. GetPRESET-HALT コマンド

中断予約カウンター情報を取得します。

項目	型説明		
引数	VT_I4	予約中断番号(カウンター1~6)	
戻り値	VT_ARRAY   VT_VARIANT		
	0	VT_I4	実行カウンター数
	1	VT_I4	予約カウンター数
	2	VT_BSTR	名称 ※名称未登録時は、空文字が入ります。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ GetPRESET-HALTコマンド実行

```
Dim values As Variant
```

```
values = caoCon.Execute("GetPRESET-HALT", 1)
```

’ 実行カウンター数取得

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

’ 予約カウンター数取得

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

’ 名称取得

```
Dim name As String
Name = values(2)
```

### 3.3.1.2.28. OPEPRESET-HALT コマンド

試験中断カウンターを設定します。このコマンドは引数によって、動作が異なります。以下のいずれかを引数に指定してください。

中断カウンターの設定を行う場合：

項目	型説明		
引数	VT_ARRAY   VT_VARIANT		中断カウンターの設定を行います。
	0	VT_I4	予約 No を指定します。 予約 No は、1～6 の範囲で指定可能です。
	1	VT_I4	中断カウンター設定値を指定します。 中断カウンター設定値は、1～9999 の範囲で指定可能です。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。	

#### 使用例

##### コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

##### OPEPRESET-HALTコマンド実行

```
Dim result As String
Dim params(1) As Long
params(0) = 1
params(1) = 100
result = caoCon.Execute("OPEPRESET-HALT", params)
```

個別カウンターの名称設定を行う場合：

項目	型説明		
引数	VT_ARRAY   VT_VARIANT		個別カウンターの名称設定を行います。
	0	VT_I4	予約 No を指定します。 予約 No は、1～6 の範囲で指定可能です。
	1	VT_BSTR	カウンター名称を指定します。 カウンター名称は、12 文字以内での設定となります。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。	

#### 使用例

##### コントローラの追加

```
Dim caoCon As CaoController
```

```
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ OPEPRESET-HALTコマンド実行**

```
Dim result As String
Dim params(1) As Variant
params(0) = CLng(1)
params(1) = "Count"
result = caoCon.Execute("OPEPRESET-HALT", params)
```

**中断カウンターの実行カウンター数をリセットする場合：**

項目	型説明	
引数	VT_I4	予約 No を指定します。 予約 No は、1～6 の範囲で指定可能です。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ OPEPRESET-HALTコマンド実行**

```
Dim result As String
Dim param As Long
param = 1
result = caoCon.Execute("OPEPRESET-HALT", param)
```

**3.3.1.3. 試験パターン情報関連コマンド**

試験パターン情報を取得するコマンドを使用する場合は、あらかじめ「PRGMREAD コマンド」を実行し、モニターする該当の試験パターンをオープンしておく必要があります。また、モニター中の試験パターンを終了する際は、「PRGMREADEND コマンド」を実行し、モニター中の試験パターンをクローズする必要があります。詳細は、「PRGMREAD コマンド」と、「PRGMREADEND コマンド」を参照してください。

試験パターン情報を設定するコマンドの場合は、あらかじめ「PRGMWRITE コマンド」を実行し、該当する試験パターンをオープンしておく必要があります。また、試験パターン設定が反映されるのは、「PRGMWRITEEND コマンド」を実行し、試験パターンのクローズが行われたときです。クローズする前に、試験パターン設定コマンド以外の設定コマンドを実行した場合は、強制的にクローズされ、それまでの編集内容は破棄されます。詳細は、「PRGMWRITE コマンド」と、「PRGMWRITEEND コマンド」を参照してください。

### 3.3.1.3.1. PRGMREAD コマンド

参照モードにて試験パターンをオープンします。

項目	型説明	
引数	VT_I4	モニターする試験パターンのパターン番号を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

' **コントローラの追加**

```
Dim caoCon As CaoController
```

```
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _  
"conn=eth:192.168.0.2")
```

' **PRGMREADコマンド実行**

```
Dim result As String
```

```
result = caoCon.Execute("PRGMREAD", 1)
```

### 3.3.1.3.2. PRGMREADEND コマンド

参照モードでオープンした試験パターンをクローズします。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

' **コントローラの追加**

```
Dim caoCon As CaoController
```

```
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _  
"conn=eth:192.168.0.2")
```

' **参照モードにて試験パターンをオープン**

```
Dim result As String
```

```
result = caoCon.Execute("PRGMREAD", 1)
```

' **PRGMREADENDコマンド実行**

```
result = caoCon.Execute("PRGMREADEND")
```

### 3.3.1.3.3. PRGMWRITE コマンド

編集モードにて試験パターンをオープンします。

項目	型説明	
引数	VT_I4	編集する試験パターンのパターン番号を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ PRGMWRITEコマンド実行

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

**3.3.1.3.4. PRGMWRITEEND コマンド**

編集モードでオープンした試験パターンをクローズします。

項目	型説明	
引数	VT_I4	編集中の試験パターンのパターン番号を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

’ PRGMWRITEENDコマンド実行

```
result = caoCon.Execute("PRGMWRITEEND", 2)
```

**3.3.1.3.5. LISTTEMP コマンド**

さらし温度設定値を取得します。

項目	型説明	
戻り値	VT_ARRAY   VT_I4	
	0	VT_I4 高温さらし温度設定値
	1	VT_I4 低温さらし温度設定値

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

```
' 参照モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
' LISTTEMPコマンド実行
Dim values() As Long
values = caoCon.Execute("LISTTEMP")
```

### 3.3.1.3.6. TEMP コマンド

さらし温度設定値を設定します。予冷・予熱設定値が自動モードの場合は、予熱・予冷設定値も自動的に再設定されます。

項目	型説明		
引数	VT_ARRAY   VT_I4		
	0	VT_I4	高温さらし温度設定値を指定します。
	1	VT_I4	低温さらし温度設定値を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。	

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
' 編集モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
' TEMPコマンド実行
result = caoCon.Execute("TEMP", Array(100, -65))
```

### 3.3.1.3.7. LISTPREAI コマンド

予熱・予冷モード設定を取得します。

項目	型説明	
戻り値	VT_BSTR	予熱・予冷モード設定 取得される値の詳細は以下のようになります。 <ul style="list-style-type: none"> <li>・ ON : 予熱, 予冷のどちらかが自動モード</li> <li>・ OFF : 予熱, 予冷の両方が手動モード</li> </ul>

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
```

```
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
' 参照モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
' LISTPREAI コマンド実行
Dim val As String
val = caoCon.Execute("LISTPREAI")
```

### 3.3.1.3.8. PREAI コマンド

予熱・予冷設定値の自動モードの設定を行います。

項目	型説明	
引数	VT_BSTR	予熱・予冷モードを指定します。 以下のいずれかを指定してください。 ・ ON : 予熱, 予冷とも自動モード設定となります。 ・ OFF : 予熱, 予冷とも手動モード設定となります。
戻り値	VT_BSTR	コマンドが正常に実行された場合, 機器からの応答文字列がそのまま格納されます。

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
' 編集モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
' PREAI コマンド実行
result = caoCon.Execute("PREAI", "ON")
```

### 3.3.1.3.9. LISTPRE コマンド

予熱・予冷温度設定値を取得します。

項目	型説明	
戻り値	VT_ARRAY   VT_I4	
	0	VT_I4 予熱温度設定値
	1	VT_I4 予冷温度設定値

#### 使用例

```
' コントローラの追加
```

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
' 参照モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
' LISTPREコマンド実行
Dim values() As Long
values = caoCon.Execute("LISTPRE")
```

### 3.3.1.3.10. PRE コマンド

予熱・予冷温度の設定を行います。

項目	型説明		
引数	VT_ARRAY   VT_I4		
	0	VT_I4	予熱温度設定値を指定します。
	1	VT_I4	予冷温度設定値を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。	

※予熱・予冷設定が自動モードの場合は、設定不可となります。

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
' 編集モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
' PREコマンド実行
result = caoCon.Execute("PRE", Array(80, -30))
```

### 3.3.1.3.11. LISTTIME コマンド

さらし時間設定値を取得します。

項目	型説明		
戻り値	VT_ARRAY   VT_BSTR		
	0	VT_BSTR	常温さらし時間(HH:mm)
	1	VT_BSTR	高温さらし時間(HH:mm)
	2	VT_BSTR	低温さらし時間(HH:mm)

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 参照モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
```

’ LISTTIMEコマンド実行

```
Dim values() As String
values = caoCon.Execute("LISTTIME")
```

3.3.1.3.12. TIME コマンド

さらし時間設定を行います。

項目	型説明		
引数	VT_ARRAY   VT_BSTR		
	0	VT_BSTR	常温さらし時間(HH:mm)を指定します。
	1	VT_BSTR	高温さらし時間(HH:mm)を指定します。
	2	VT_BSTR	低温さらし時間(HH:mm)を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。	

※運転中の試験パターンの場合、ゾーン変更となる設定(常温さらし時間 00:00⇔00:00 以外)を行うことはできません。

※熱衝撃装置でELタイプ(オプション常温さらし未搭載)では常温さらし時間は00:00と設定してください。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

’ TIMEコマンド実行

```
result = caoCon.Execute("TIME", Array("00:00", "02:30", "01:00"))
```

3.3.1.3.13. LISTCYCLE コマンド

試験サイクル数を取得します。

項目	型説明	
戻り値	VT_I4	試験サイクル数

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 参照モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
```

’ LISTCYCLEコマンド実行

```
Dim val As Long
val = caoCon.Execute("LISTCYCLE")
```

3.3.1.3.14. CYCLE コマンド

試験パターンの繰り返し回数(サイクル数)を設定します。

項目	型説明	
引数	VT_I4	試験サイクル数を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

’ CYCLEコマンド実行

```
result = caoCon.Execute("CYCLE", 99)
```

3.3.1.3.15. LISTSTARTPOSITION コマンド

試験開始位置設定を取得します。

項目	型説明	
戻り値	VT_BSTR	試験開始位置 設定による応答は以下のようになります。 <ul style="list-style-type: none"> <li>・ H : 高温さらしから開始する設定</li> <li>・ L : 低温さらしから開始する設定</li> </ul>

**使用例**

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")

' 参照モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)

' LISTSTARTPOSITIONコマンド実行
Dim val As String
val = caoCon.Execute("LISTSTARTPOSITION")
```

**3.3.1.3.16. STARTPOSITION コマンド**

試験開始位置を設定します。

項目	型説明	
引数	VT_BSTR	試験開始位置を指定します。 以下のいずれかを指定してください。 ・ H : 高温さらし開始の場合 ・ L : 低温さらし開始の場合
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")

' 編集モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)

' STARTPOSITIONコマンド実行
result = caoCon.Execute("STARTPOSITION", "H")
```

**3.3.1.3.17. LISTG-SOAK コマンド**

安定時間制御設定を取得します。

項目	型説明
----	-----

項目	型説明	
戻り値	VT_BSTR	安定時間制御 設定状態によって、以下のいずれかが取得されます。 <ul style="list-style-type: none"> <li>・ ON : 安定時間制御機能有効</li> <li>・ OFF : 安定時間制御機能無効</li> </ul>

**使用例**

’ **コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ **参照モードにて試験パターンをオープン**

```
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
```

’ **LISTG-SOAKコマンド実行**

```
Dim val As String
val = caoCon.Execute("LISTG-SOAK")
```

**3.3.1.3.18. G-SOAK コマンド**

安定時間制御の設定を行います。

項目	型説明	
引数	VT_BSTR	安定時間制御モードを指定します。指定可能な値は、「LISTG-SOAK コマンド」の戻り値と同様です。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ **コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ **編集モードにて試験パターンをオープン**

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

’ **G-SOAKコマンド実行**

```
result = caoCon.Execute("G-SOAK", "ON")
```

**3.3.1.3.19. LISTEND コマンド**

試験終了条件設定を取得します。

項目	型説明	
戻り値	VT_BSTR	<p>試験終了条件</p> <p>取得した応答の詳細は以下のようになります.</p> <ul style="list-style-type: none"> <li>・ OFF : 停止</li> <li>・ SETUP : 準備</li> <li>・ DEFROST : 除霜後停止</li> <li>・ DRY : 乾燥後停止</li> <li>・ AMB : 常温後停止</li> <li>・ DEF-DRY : 常温後乾燥後停止</li> </ul>

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
```

```
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 参照モードにて試験パターンをオープン

```
Dim result As String
```

```
result = caoCon.Execute("PRGMREAD", 1)
```

’ LISTENDコマンド実行

```
Dim val As String
```

```
val = caoCon.Execute("LISTEND")
```

**3.3.1.3.20. END コマンド**

試験終了条件を設定します.

項目	型説明	
引数	VT_BSTR	<p>試験終了条件を指定します.</p> <p>以下のいずれかを指定してください.</p> <ul style="list-style-type: none"> <li>・ OFF : 停止</li> <li>・ SETUP : 準備状態</li> <li>・ DEFROST : 除霜後停止</li> <li>・ DRY : 乾燥後停止</li> <li>・ AMB : 常温後停止 (EL タイプオプション常温さらし未搭載では使用不可)</li> <li>・ DEFDRY : 除霜/乾燥後停止</li> </ul>
戻り値	VT_BSTR	<p>コマンドが正常に実行された場合, 機器からの応答文字列がそのまま格納されます.</p>

**使用例**

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ 編集モードにて試験パターンをオープン**

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

**’ ENDコマンド実行**

```
result = caoCon.Execute("END", "OFF")
```

### 3.3.1.3.21. LISTNAME コマンド

試験パターン名を取得します。

項目	型説明	
戻り値	VT_BSTR	パターン名称 パターン名称は 14 文字で返却されます (14 文字に満たない場合はスペースがセットされます)。

**使用例**

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ 参照モードにて試験パターンをオープン**

```
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
```

**’ LISTNAMEコマンド実行**

```
Dim val As String
val = caoCon.Execute("LISTNAME")
```

### 3.3.1.3.22. NAME コマンド

試験パターン名を設定します。

項目	型説明	
引数	VT_BSTR	試験パターン名を指定します。14 文字以内の文字列で指定してください。小文字で指定した場合、全て大文字に変換されます。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
```

```

"conn=eth:192.168.0.2")
' 編集モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
' NAMEコマンド実行
result = caoCon.Execute("NAME", "PATTERN001")

```

### 3.3.1.3.23. LISTSENSOR コマンド

センサー位置設定を取得します。

項目	型説明	
戻り値	VT_BSTR	<p>センサー位置。</p> <p>取得した応答の詳細は以下のようになります。</p> <ul style="list-style-type: none"> <li>・ UP : 風上センサー使用時</li> <li>・ DOWN : 風下センサー使用時</li> <li>・ SPECIMENT : 試料温度センサー使用時</li> </ul>

#### 使用例

```

' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
"conn=eth:192.168.0.2")
' 参照モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
' LISTSENSORコマンド実行
Dim val As String
val = caoCon.Execute("LISTSENSOR")

```

### 3.3.1.3.24. SENSOR コマンド

センサー位置を設定します。

項目	型説明	
引数	VT_BSTR	<p>センサー位置を指定します。</p> <p>以下のいずれかを指定してください。</p> <ul style="list-style-type: none"> <li>・ UP : 風上センサーを選択する場合</li> <li>・ DOWN : 風下センサーを選択する場合</li> <li>・ SPECIMENT : 試料センサーを選択する場合</li> </ul>
戻り値	VT_BSTR	<p>コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。</p>

**使用例**

```

' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")

' 編集モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)

' SENSORコマンド実行
result = caoCon.Execute("SENSOR", "UP")
    
```

**3.3.1.3.25. LISTDEFROST コマンド**

除霜モード設定(除霜条件)を取得します。

項目	型説明	
戻り値	VT_I4	サイクル除霜が設定されている場合、除霜サイクル数を取得します。
	または	
	VT_BSTR	除霜条件 設定による応答は以下のようになります。 <ul style="list-style-type: none"> <li>・ AI : 自動除霜設定</li> <li>・ NO : 除霜停止設定</li> </ul>

**使用例**

```

' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")

' 参照モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)

' LISTDEFROSTコマンド実行
Dim val As Variant
val = caoCon.Execute("LISTDEFROST")
    
```

**3.3.1.3.26. DEFROST コマンド**

除霜条件を設定します。

項目	型説明	
引数	VT_I4	サイクル数を指定します。 サイクル除霜を指定したい場合、こちらの引数を指定してください。 サイクル除霜数は1~9999の範囲で指定してください。
	または	

項目	型説明	
	VT_BSTR	除霜モードを指定します。 以下のいずれかが指定できます。 <ul style="list-style-type: none"> <li>・ AI : 自動除霜を指定する場合</li> <li>・ OFF : 停止モードを指定する場合</li> </ul>
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

’ DEFROSTコマンド実行

’ サイクル数を指定

```
Dim param As Long
param = 99
result = caoCon.Execute("DEFROST", param)
```

’ 自動除霜を指定

```
result = caoCon.Execute("DEFROST", "AI")
```

3.3.1.3.27. LISTEMP-L-DEFROST コマンド

除霜待機位置設定を取得します。

項目	型説明	
戻り値	VT_BSTR	除霜待機位置 設定による応答は以下のようになります。 <ul style="list-style-type: none"> <li>・ H : 制御する</li> <li>・ OFF : 制御しない</li> <li>・ A : 制御しない(常温) ※EL タイプ(オプション常温さらし未搭載)は準備待機</li> </ul>

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

```
' 参照モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
' LISTTEMP-L-DEFROSTコマンド実行
Dim val As String
val = caoCon.Execute("LISTTEMP-L-DEFROST")
```

### 3.3.1.3.28. TEMP-L-DEFROST コマンド

除霜完了待ち時のテストエリア制御の設定を行います。

項目	型説明	
引数	VT_BSTR	<p>除霜待機位置を指定します。 以下のいずれかを指定してください。</p> <ul style="list-style-type: none"> <li>・ H : テストエリア制御する</li> <li>・ OFF : テストエリア制御しない</li> <li>・ A : テストエリア制御しない(常温)</li> </ul>
戻り値	VT_BSTR	<p>コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。</p>

※常温さらし機能が未搭載の場合、Aを指定すると、エラーが返却されます。エラーに関しては、「表 5-2 デバイスのエラーコード表」のエラーコード[0x80100006]を参照してください。また、3 ゾーン試験パターンの設定下では、同様のエラーが返却されます。

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
' 編集モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
' TEMP-L-DEFROSTコマンド実行
result = caoCon.Execute("TEMP-L-DEFROST", "A")
```

### 3.3.1.3.29. LISTQ-EXP コマンド

さらし時間短縮設定を取得します。

項目	型説明	
戻り値	VT_BSTR	<p>さらし時間短縮</p> <ul style="list-style-type: none"> <li>・ ON : さらし時間短縮機能有効</li> <li>・ OFF : さらし時間短縮機能無効</li> </ul>

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 参照モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
```

’ LISTQ-EXPコマンド実行

```
Dim val As String
val = caoCon.Execute("LISTQ-EXP")
```

**3.3.1.3.30. Q-EXP コマンド**

さらし時間短縮制御設定を行います。

項目	型説明	
引数	VT_BSTR	さらし時間短縮モードを指定します。指定可能な値は、「LISTQ-EXP コマンド」の戻り値と同様です。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

’ Q-EXPコマンド実行

```
result = caoCon.Execute("Q-EXP", "ON")
```

**3.3.1.3.31. LISTTEMPLIMIT コマンド**

温度過昇/温度過冷設定値を取得します。

項目	型説明		
戻り値	VT_ARRAY   VT_I4		
	0	VT_I4	温度過昇設定値
	1	VT_I4	温度過冷設定値

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
' 参照モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
' LISTTEMPLIMITコマンド実行
Dim values() As Long
values = caoCon.Execute("LISTTEMPLIMIT")
```

### 3.3.1.3.32. TEMPLIMIT コマンド

温度過昇/温度過冷設定を設定します。

項目	型説明	
引数	VT_ARRAY   VT_I4	
	0	VT_I4 温度過昇値を指定します。以下の範囲で設定可能です。 ・ 高温さらし温度設定値 ~ 高温さらし温度設定値+50°C
	1	VT_I4 温度過冷値を指定します。以下の範囲で設定可能です。 ・ 低温さらし温度設定値-50°C ~ 低温さらし温度設定値
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
' 編集モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
' TEMPLIMITコマンド実行
result = caoCon.Execute("TEMPLIMIT", Array(180, -70))
```

### 3.3.1.3.33. LISTAUXCOOLER コマンド

補助冷却設定を取得します。

項目	型説明	
戻り値	VT_BSTR	補助冷却 設定状態によって、以下のいずれかが取得されます。 ・ ON : 補助冷却装置有効 ・ OFF : 補助冷却装置無効

**使用例**

' **コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

' **参照モードにて試験パターンをオープン**

```
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
```

' **LISTAUXCOOLERコマンド実行**

```
Dim val As String
val = caoCon.Execute("LISTAUXCOOLER")
```

**3.3.1.3.34. AUXCOOLER コマンド**

補助冷却装置のモードを設定します。

項目	型説明	
引数	VT_BSTR	補助冷却モードを指定します。指定可能な値は、「LISTAUXCOOLER コマンド」の戻り値と同様です。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

' **コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

' **編集モードにて試験パターンをオープン**

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

' **AUXCOOLERコマンド実行**

```
result = caoCon.Execute("AUXCOOLER", "ON")
```

**3.3.1.3.35. LISTDRYMODESET コマンド**

前後処理温度・時間設定を取得します。

項目	型説明		
戻り値	VT_ARRAY   VT_VARIANT		
	0	VT_I4	前処理温度設定値
	1	VT_BSTR	前処理時間設定値 (HH:mm)
	2	VT_I4	後処理温度設定値
	3	VT_BSTR	後処理時間設定値 (HH:mm)

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 参照モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
```

’ LISTDRYMODESET コマンド実行

```
Dim values() As Variant
values = caoCon.Execute("LISTDRYMODESET")
```

**3.3.1.3.36. DRYMODESET コマンド**

前後処理温度・時間設定を設定します。

項目	型説明		
引数	VT_ARRAY   VT_VARIANT		
	0	VT_I4	前処理温度設定値を指定します。
	1	VT_BSTR	前処理時間設定値 (HH:mm) を指定します。
	2	VT_I4	後処理温度設定値を指定します。
	3	VT_BSTR	後処理時間設定値 (HH:mm) を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。	

※温度設定値は 60～200°C の範囲で設定可能です。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

’ DRYMODESET コマンド実行

```
Dim param(3) As Variant
param(0) = CLng(50)
param(1) = "00:30"
param(2) = CLng(60)
param(3) = "01:15"
result = caoCon.Execute("DRYMODESET", param)
```

**3.3.1.3.37. LISTDRYMODE コマンド**

前後処理モード設定を取得します。

項目	型説明		
戻り値	VT_ARRAY   VT_BSTR		
	0	VT_BSTR	前処理モード <ul style="list-style-type: none"> <li>ON : 処理を実行する</li> <li>OFF : 処理を実行しない</li> </ul>
	1	VT_BSTR	後処理モード. 上記の前処理モードと同形式の値を取得します.

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 参照モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
```

’ LISTDRYMODEコマンド実行

```
Dim values() As String
values = caoCon.Execute("LISTDRYMODE")
```

3.3.1.3.38. DRYMODE コマンド

前後処理モードを設定します.

項目	型説明		
引数	VT_ARRAY   VT_BSTR		
	0	VT_BSTR	前処理モードを指定します. 以下のいずれかを指定してください. <ul style="list-style-type: none"> <li>ON : 処理モードが有効設定となります.</li> <li>OFF : 処理モードが無効設定となります.</li> </ul>
	1	VT_BSTR	後処理モードを指定します. 上記の前処理モードの指定する値と同形式の値が指定可能です.
戻り値	VT_BSTR	コマンドが正常に実行された場合, 機器からの応答文字列がそのまま格納されます.	

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
' DRYMODEコマンド実行
result = caoCon.Execute("DRYMODE", Array("ON", "OFF"))
```

### 3.3.1.3.39. LIST-PRE-ECONOMY コマンド

エコ運転条件設定を取得します。

項目	型説明		
戻り値	VT_ARRAY   VT_BSTR		
	0	VT_BSTR	予熱エコ運転条件 <ul style="list-style-type: none"> <li>・ ON : 予熱エコ設定有効</li> <li>・ OFF : 予熱エコ設定無効</li> </ul>
	1	VT_BSTR	予冷エコ運転条件 <ul style="list-style-type: none"> <li>・ ON : 予冷エコ設定 有効</li> <li>・ OFF : 予冷エコ設定 無効</li> </ul>

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
' 参照モードにて試験パターンをオープン
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
' LIST-PRE-ECONOMYコマンド実行
Dim values() As String
values = caoCon.Execute("LIST-PRE-ECONOMY")
```

### 3.3.1.3.40. PRE-ECONOMY コマンド

エコ運転条件を設定します。

項目	型説明		
引数	VT_ARRAY   VT_BSTR		
	0	VT_BSTR	予熱エコ運転を指定します。 以下のいずれかを指定してください。 <ul style="list-style-type: none"> <li>・ ON : 予熱エコ設定 有効</li> <li>・ OFF : 予熱エコ設定 無効</li> </ul>

項目	型説明	
	1	VT_BSTR 予冷エコ運転を指定します。 以下のいずれかを指定してください。 ・ ON : 予冷エコ設定 有効 ・ OFF : 予冷エコ設定 無効
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

※低温さらし温度設定が-66°C以下で予冷エコ設定を ON にしようとした場合、エラーが返却されます。エラーの詳細に関しては、「表 5-2 デバイスのエラーコード表」のエラーコード[0x80100006]、またはエラーコード[0x80100019]を参照してください。

**使用例**

’ **コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ **編集モードにて試験パターンをオープン**

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

’ **PRE-ECONOMYコマンド実行**

```
result = caoCon.Execute("PRE-ECONOMY", Array("ON", "OFF"))
```

**3.3.1.3.41. LISTRELAY1 コマンド**

外部出力1設定を取得します。

項目	型説明	
戻り値	VT_I4	外部出力1

※設定値の詳細は以下のようになります。

設定値	説明
1	試験中 ON
2	高温さらし運転時 ON
3	低温さらし運転時 ON
4	高温さらし終了後 ON(※1)
5	低温さらし終了後 ON(※1)
6	高温さらし到達後 ON
7	低温さらし到達後 ON
8	一時停止中 ON

設定値	説明
9	出力 OFF
10	常温さらし運転時 ON
11	常温さらし終了後 ON(※1)

※1 出力時間は、画面での設定によります。(1~60 秒)

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 参照モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
```

’ LISTRELAY1 コマンド実行

```
Dim val As Long
val = caoCon.Execute("LISTRELAY1")
```

### 3.3.1.3.42. RELAY1 コマンド

外部出力 No. 1 の設定を行います。

項目	型説明	
引数	VT_I4	外部出力 1 を指定します。 設定値は 1~11 まで指定可能です。設定値の詳細については、「LISTRELAY1 コマンド」の戻り値を参照してください。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

※設定値 10, 11 は熱衝撃装置で EL タイプ(オプション常温さらし未搭載)では使用不可となります。

**使用例**

’ コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

’ RELAY1 コマンド実行

```
result = caoCon.Execute("RELAY1", 8)
```

### 3.3.1.3.43. LISTRELAY2 コマンド

外部出力 2 設定を取得します。

項目	型説明	
戻り値	VT_I4	外部出力 2. 戻り値の詳細については、「LISTRELAY1 コマンド」を参照してください。

**使用例**

**コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**参照モードにて試験パターンをオープン**

```
Dim result As String
result = caoCon.Execute("PRGMREAD", 1)
```

**LISTRELAY2コマンド実行**

```
Dim val As Long
val = caoCon.Execute("LISTRELAY2")
```

**3.3.1.3.44. RELAY2 コマンド**

外部出力 No. 2 を設定します。

項目	型説明	
引数	VT_I4	外部出力 2 を指定します。指定可能な値については、「RELAY1 コマンド」の引数を参照してください。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

**コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**編集モードにて試験パターンをオープン**

```
Dim result As String
result = caoCon.Execute("PRGMWRITE", 2)
```

**RELAY2コマンド実行**

```
result = caoCon.Execute("RELAY2", 8)
```

**3.3.2. CaoExtension クラスコマンド**

CaoExtension クラスで実行する Execute メソッドは、シリアル通信で接続されている場合のみ有効です。使用例は各コマンドの詳細で記述しています。

表 3-5 CaoExtension クラスコマンド一覧

コマンド	説明	参照
機器設定関連コマンド		
GetROM	コントローラの ROM バージョンを取得します。	P. 67
GetROMDISP	表示器の ROM バージョンを取得します。	P. 67
GetPOWERCUT	停復電動作モードを取得します。	P. 67
GetSTOPMODE	装置停止モードを取得します。	P. 67
GetG-SOAKTEMP	安定時間制御用安定条件を取得します。	P. 67
GetDRYTEMP	乾燥運転用設定温度を取得します。	P. 68
GetAMBTIME	常温運転用設定時間を取得します。	P. 68
GetKEYPROTECT	キープロテクト状態を取得します。	P. 68
SetKEYPROTECT	キープロテクト状態を設定します。	P. 68
機器の運転関連コマンド		
GetMODE	運転モードを取得します。	P. 68
SetMODE	運転モードを設定します。	P. 68
GetASSIGN	試験パターンを取得します。	P. 68
SetASSIGN	試験パターンを設定します。	P. 69
GetTEMP	さらし温度を取得します。	P. 70
GetPRE	予熱・予冷温度を取得します。	P. 70
GetTIME	さらし残時間を取得します。	P. 70
GetCYCLE	試験サイクル数を取得します。	P. 70
GetMODEL	機種 ID を取得します。	P. 71
GetALARM	アラーム状態を取得します。	P. 71
OPESTANDBY	アイドリング状態にします。	P. 71
OPESETUP	準備運転を開始します。	P. 71
OPESETUPEND	準備後試験運転を開始します。	P. 72
OPETEST	試験を開始します。	P. 72
OPEHALT	試験を中断状態にします。	P. 72
OPERELASE	試験の中断状態を解除します。	P. 72
OPECYCLERESET	残サイクルを取り消します。	P. 72
OPEDEFROST	除霜運転をします。	P. 72

コマンド	説明	参照
OPEDRY	乾燥運転をします。	P. 72
OPEPRESET-SETUP	準備予約タイマーを設定します。	P. 72
OPEPRESET-TEST	試験予約タイマーを設定します。	P. 72
OPEPRESET-ON	タイマー予約をします。	P. 72
OPEPRESET-CLR	タイマー予約を解除します。	P. 73
試験パターン情報関連コマンド		
PRGMREAD コマンド	参照モードにて試験パターンをオープンします。	P. 73
PRGMREADEND コマンド	参照モードでオープンした試験パターンをクローズします。	P. 74
PRGMWRITE コマンド	編集モードにて試験パターンをオープンします。	P. 74
PRGMWRITEEND コマンド	編集モードでオープンした試験パターンをクローズします。	P. 75
PRGMCREATE コマンド	編集モードにて試験パターンをオープンします。	P. 75
PRGMCREATEEND コマンド	編集モードでオープンした試験パターンをクローズします。	P. 75
LISTTEMP	さらし温度設定値を取得します。	P. 76
TEMP	さらし温度設定を設定します。	P. 76
LISTPREAI	予熱・予冷モード設定を取得します。	P. 76
PREAI	予熱・予冷モード設定を設定します。	P. 76
LISTPRE	予熱・予冷温度設定を取得します。	P. 76
PRE	予熱・予冷温度設定を設定します。	P. 76
LISTTIME	さらし時間設定値を取得します。	P. 76
TIME	時間設定を設定します。	P. 77
LISTCYCLE	試験サイクル数を取得します。	P. 77
CYCLE	試験サイクル数を設定します。	P. 77
LISTSTARTPOSITION	試験開始位置設定を取得します。	P. 77
STARTPOSITION	試験開始位置を設定します。	P. 77
LISTPAUSE	試験中断予約設定を取得します。	P. 77
PAUSE	試験中断予約を設定します。	P. 78
LISTG-SOAK	安定時間制御設定を取得します。	P. 78
G-SOAK	安定時間制御モードを設定します。	P. 79

コマンド	説明	参照
LISTEND	試験終了条件設定を取得します。	P. 79
END	終了条件を設定します。	P. 79
LISTNAME	試験パターン名を取得します。	P. 80
NAME	試験パターン名を設定します。	P. 80
LISTSENSOR	センサー位置設定を取得します。	P. 80
SENSOR	センサー位置を設定します。	P. 80
LISTDEFROST	除霜モード設定を取得します。	P. 80
DEFROST	除霜モードを設定します。	P. 80
LISTTEMP-L-DEFROST	除霜待機位置設定を取得します。	P. 80
TEMP-L-DEFROST	除霜待機位置を設定します。	P. 81
LISTQ-EXP	さらし時間短縮設定を取得します。	P. 82
Q-EXP	さらし時間短縮モードを設定します。	P. 82
LISTTEMPLIMIT	温度過昇/温度過冷設定値を取得します。	P. 82
TEMPLIMIT	温度過昇/温度過冷設定を設定します。	P. 82
LISTAUXCOOLER	補助冷却設定を取得します。	P. 82
AUXCOOLER	補助冷却モードを設定します。	P. 82
LISTDRYMODESET	前後処理温度・時間設定を取得します。	P. 82
DRYMODESET	前後処理温度・時間設定を設定します。	P. 83
LISTDRYMODE	前後処理モード設定を取得します。	P. 83
DRYMODE	前後処理モードを設定します。	P. 84
LIST-PRE-ECONOMY	エコ運転条件設定を取得します。	P. 84
PRE-ECONOMY	エコ運転条件を設定します。	P. 84
LISTRELAY1	外部出力1設定を取得します。	P. 84
RELAY1	外部出力1を設定します。	P. 84
LISTRELAY2	外部出力2設定を取得します。	P. 84
RELAY2	外部出力2を設定します。	P. 84
ECONOMY	エコ運転を設定します。	P. 84

### 3.3.2.1. 機器設定関連コマンド

#### 3.3.2.1.1. GetROM コマンド

コントローラの ROM バージョンを取得します。詳細は、CaoController クラスの「GetROM コマンド」を参照してください。

#### 3.3.2.1.2. GetROMDISP コマンド

表示器の ROM バージョンを取得します。詳細は、CaoController クラスの「GetROMDISP コマンド」を参照してください。

#### 3.3.2.1.3. GetPOWERCUT コマンド

停復電動作モードを取得します。

項目	型説明	
戻り値	VT_BSTR	停復電動作モード 設定状態によって以下のいずれかが取得されます。 <ul style="list-style-type: none"> <li>・ STOP : 運転を停止する設定</li> <li>・ CONTINUE : 運転を継続する設定</li> </ul>

#### 使用例

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' GetPOWERCUTコマンド実行
Dim val As String
val = caoExt.Execute("GetPOWERCUT")
```

#### 3.3.2.1.4. GetSTOPMODE コマンド

装置停止モードを取得します。詳細は、CaoController クラスの「GetSTOPMODE コマンド」を参照してください。

#### 3.3.2.1.5. GetG-SOAKTEMP コマンド

安定時間制御用安定条件を取得します。

項目	型説明	
戻り値	VT_I4	安定制御用の安定幅(温度)。温度は 1~10 までの範囲で取得されます。

#### 使用例

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
```

```
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' GetG-SOAKTEMPコマンド実行
Dim val As Long
val = caoExt.Execute("GetG-SOAKTEMP")
```

### 3.3.2.1.6. GetDRYTEMP コマンド

乾燥運転時の温度設定値を取得します。詳細は、CaoController クラスの「GetDRYTEMP コマンド」を参照してください。

### 3.3.2.1.7. GetAMBTIME コマンド

常温運転用設定時間を取得します。詳細は、CaoController クラスの「GetAMBTIME コマンド」を参照してください。

### 3.3.2.1.8. GetKEYPROTECT コマンド

キープロテクト状態を取得します。詳細は、CaoController クラスの「GetKEYPROTECT コマンド」を参照してください。

### 3.3.2.1.9. SetKEYPROTECT コマンド

キープロテクト状態を設定します。詳細は、CaoController クラスの「SetKEYPROTECT コマンド」を参照してください。

## 3.3.2.2. 機器の運転関連コマンド

### 3.3.2.2.1. GetMODE コマンド

現在の装置の運転状態と完了予定日時を取得します。詳細は、CaoController クラスの「GetMODE コマンド」を参照してください。

### 3.3.2.2.2. SetMODE コマンド

運転モードを設定します。詳細は、CaoController クラスの「SetMODE コマンド」を参照してください。

### 3.3.2.2.3. GetASSIGN コマンド

試験パターンを取得します。

項目	型説明	
戻り値	VT_ARRAY   VT_VARIANT	
	0	VT_I4

項目	型説明		
	1	VT_I4	メモリ種別 以下のいずれかが取得されます。 ・ 0 : RAM ・ 1 : ROM
	2	VT_BSTR	試験名称. 14 文字以内で取得されます.

**使用例**

```

' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' GetASSIGNコマンド実行
Dim values As Variant
values = caoExt.Execute("GetASSIGN")

' パターン番号取得
Dim ptrnNo As Long
ptrnNo = values(0)

' メモリ種別取得
Dim memory As Long
Memory = values(1)

' 試験名称
Dim name As String
name = values(2)
    
```

**3.3.2.2.4. SetASSIGN コマンド**

試験パターンを設定します.

項目	型説明		
引数	VT_ARRAY   VT_I4		試験パターンを指定します.
	0	VT_I4	試験パターンのパターン番号
	1	VT_I4	メモリ種別 以下のいずれかを指定してください。 ・ 0 : RAM ・ 1 : ROM
戻り値	VT_BSTR		コマンドが正常に実行された場合, 機器からの応答文字列がそのまま格納されます.

**使用例**

```

' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' SetASSIGNコマンド実行
Dim result As String
result = caoExt.Execute("SetASSIGN", Array(1,0))
    
```

### 3.3.2.2.5. GetTEMP コマンド

さらし温度を取得します。詳細は、CaoController クラスの「GetTEMP コマンド」を参照してください。

### 3.3.2.2.6. GetPRE コマンド

予熱・予冷温度を取得します。詳細は、CaoController クラスの「GetPRE コマンド」を参照してください。

### 3.3.2.2.7. GetTIME コマンド

さらし残時間を取得します。

項目	型説明		
戻り値	VT_ARRAY   VT_BSTR		
	0	VT_BSTR	さらし状態 ・ さらし状態は以下のように表現されます。 H : 高温さらし運転中 L : 低温さらし運転中 A : 常温さらし運転中
	1	VT_BSTR	残時間(HH:mm)

#### 使用例

```

' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' GetTIMEコマンド実行
Dim values() As String
values = caoExt.Execute("GetTIME")
    
```

### 3.3.2.2.8. GetCYCLE コマンド

試験サイクル数を取得します。詳細は、CaoController クラスの「GetCYCLE コマンド」を参照してください。

### 3.3.2.2.9. GetMODEL コマンド

機種 ID を取得します。詳細は, CaoController クラスの「GetMODEL コマンド」を参照してください。

### 3.3.2.2.10. GetALARM コマンド

アラーム状態を取得します。

項目	型説明	
戻り値	VT_EMPTY	発生しているアラームがありません
	または	
	VT_ARRAY   VT_I4	最大 6 個までのアラーム情報
	<i>i</i> VT_I4	アラーム番号 アラーム番号の詳細は <u>ThermalShock RS-485 マニュアル内の「表 3.21 警報コード」</u> を参照してください。

※ *i* - 発生しているアラーム数分 (1~6)

#### 使用例

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' GetALARMコマンド実行
Dim values As Variant
values = caoExt.Execute("GetALARM")

' アラーム数分取得
If Not IsEmpty(values) Then
    Dim i As Long
    For i = 0 To UBound(values)
        Dim val As Long
        val = values(i)
    Next i
End If
```

### 3.3.2.2.11. OPESTANDBY コマンド

装置をアイドル状態にします。詳細は, CaoController クラスの「OPESTANDBY コマンド」を参照してください。

### 3.3.2.2.12. OPESETUP コマンド

装置を準備開始状態にします。詳細は, CaoController クラスの「OPESETUP コマンド」を参照してください。

### 3.3.2.2.13. OPESETUPEND コマンド

装置を準備後試験開始状態にします。詳細は、CaoController クラスの「OPESETUPEND コマンド」を参照してください。

### 3.3.2.2.14. OPETEST コマンド

装置を試験開始状態にします。詳細は、CaoController クラスの「OPETEST コマンド」を参照してください。

### 3.3.2.2.15. OPEHALT コマンド

装置を試験中断状態にします。詳細は、CaoController クラスの「OPEHALT コマンド」を参照してください。

### 3.3.2.2.16. OPERERELEASE コマンド

試験中断状態を解除します。詳細は、CaoController クラスの「OPERERELEASE コマンド」を参照してください。

### 3.3.2.2.17. OPECYCLERESET コマンド

残サイクルを消去します。詳細は、CaoController クラスの「OPECYCLERESET コマンド」を参照してください。

### 3.3.2.2.18. OPEDEFROST コマンド

除霜運転を開始します。詳細は、CaoController クラスの「OPEDEFROST コマンド」を参照してください。

### 3.3.2.2.19. OPEDRY コマンド

乾燥運転を開始します。詳細は、CaoController クラスの「OPEDRY コマンド」を参照してください。

### 3.3.2.2.20. OPEPRESET-SETUP コマンド

準備予約タイマー（試験準備を完了させる時間）を設定します。詳細は、CaoController クラスの「OPEPRESET-SETUP コマンド」を参照してください。

### 3.3.2.2.21. OPEPRESET-TEST コマンド

試験予約タイマー（試験を開始させる時間）を設定します。詳細は、CaoController クラスの「OPEPRESET-TEST コマンド」を参照してください。

### 3.3.2.2.22. OPEPRESET-ON コマンド

設定されたタイマを起動させます。詳細は、CaoController クラスの「OPEPRESET-ON コマンド」を

参照してください。

### 3.3.2.2.3. OPEPRESET-CLR コマンド

起動中のタイマを解除します。詳細は、CaoController クラスの「OPEPRESET-CLR コマンド」を参照してください。

### 3.3.2.3. 試験パターン情報関連コマンド

試験パターン情報の取得を行うコマンドを使用する場合は、あらかじめ「PRGMREAD コマンド」を実行し、モニターする該当の試験パターンをオープンしておく必要があります。また、モニター中の試験パターンを終了する際は、「PRGMREADEND コマンド」を実行し、モニター中の試験パターンをクローズする必要があります。詳細は、「PRGMREAD コマンド」と、「PRGMREADEND コマンド」を参照してください。

試験パターン情報の設定を行うコマンドを使用する場合は、あらかじめ「PRGMWRITE コマンド」または「PRGMCREATE コマンド」を実行し、該当する試験パターンをオープンしておく必要があります。また、試験パターン情報の設定が反映されるのは、「PRGMWRITEEND コマンド」または「PRGMCREATEEND コマンド」を実行し、試験パターンのクローズが行われたときです。クローズする前に、試験パターン設定コマンド以外の設定コマンドを実行した場合は、強制的にクローズされ、それまでの編集内容は破棄されます。「PRGMWRITE コマンド」を実行して試験パターンをオープンした場合、「PRGMWRITEEND コマンド」を実行してクローズしてください。また、「PRGMCREATE コマンド」を実行して試験パターンをオープンした場合、「PRGMCREATEEND コマンド」を実行してクローズしてください。詳細は、「PRGMWRITE コマンド」、「PRGMWRITEEND コマンド」、「PRGMCREATE コマンド」、「PRGMCREATEEND コマンド」を参照してください。

#### 3.3.2.3.1. PRGMREAD コマンド

参照モードにて試験パターンをオープンします。

項目	型説明		
引数	VT_ARRAY   VT_I4		
	0	VT_I4	モニターする試験パターンのパターン番号
	1	VT_I4	メモリ種別 以下のいずれかを指定可能してください。 ・ 0 : RAM ・ 1 : ROM
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。	

**使用例**

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' PRGMREADコマンド実行
Dim result As String
result = caoExt.Execute("PRGMREAD", Array(1, 0))
```

**3.3.2.3.2. PRGMREADEND コマンド**

参照モードでオープンした試験パターンをクローズします。

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' 参照モードにて試験パターンをオープン
Dim result As String
result = caoExt.Execute("PRGMREAD", Array(1, 0))
' PRGMREADENDコマンド実行
result = caoExt.Execute("PRGMREADEND")
```

**3.3.2.3.3. PRGMWRITE コマンド**

編集モードにて試験パターンをオープンします。

項目	型説明	
引数	VT_I4	編集する試験パターンのパターン番号
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' PRGMWRITEコマンド実行
Dim result As String
result = caoExt.Execute("PRGMWRITE", 2)
```

### 3.3.2.3.4. PRGMWRITEEND コマンド

編集モードでオープンした試験パターンをクローズします。

項目	型説明	
引数	VT_I4	編集中の試験パターンのパターン番号
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' 編集モードにて試験パターンをオープン
Dim result As String
result = caoExt.Execute("PRGMWRITE", 2)
' PRGMWRITEENDコマンド実行
result = caoExt.Execute("PRGMWRITEEND", 2)
```

### 3.3.2.3.5. PRGMCREATE コマンド

編集モードにて試験パターンをオープンします。

項目	型説明	
引数	VT_I4	編集する試験パターンのパターン番号
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' PRGMCREATEコマンド実行
Dim result As String
result = caoExt.Execute("PRGMCREATE", 2)
```

### 3.3.2.3.6. PRGMCREATEEND コマンド

編集モードでオープンした試験パターンをクローズします。

項目	型説明	
引数	VT_I4	編集中の試験パターンのパターン番号
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")

' 編集モードにて試験パターンをオープン
Dim result As String
result = caoExt.Execute("PRGMCREATE", 2)
' PRGMCREATEEND コマンド実行
result = caoExt.Execute("PRGMCREATEEND", 2)
```

**3.3.2.3.7. LISTTEMP コマンド**

さらし温度設定値を取得します。詳細は、CaoController クラスの「LISTTEMP コマンド」を参照してください。

**3.3.2.3.8. TEMP コマンド**

さらし温度設定値を設定します。詳細は、CaoController クラスの「TEMP コマンド」を参照してください。

**3.3.2.3.9. LISTPREAI コマンド**

予熱・予冷モード設定値を取得します。詳細は、CaoController クラスの「LISTPREAI コマンド」を参照してください。

**3.3.2.3.10. PREAI コマンド**

予熱・予冷モード設定を設定します。詳細は、CaoController クラスの「PREAI コマンド」を参照してください。

**3.3.2.3.11. LISTPRE コマンド**

予熱・予冷温度設定値を取得します。詳細は、CaoController クラスの「LISTPRE コマンド」を参照してください。

**3.3.2.3.12. PRE コマンド**

予熱・予冷温度を設定します。詳細は、CaoController クラスの「PRE コマンド」を参照してください。

**3.3.2.3.13. LISTTIME コマンド**

さらし時間設定値を取得します。詳細は、CaoController クラスの「LISTTIME コマンド」を参照してください。

### 3.3.2.3.14. TIME コマンド

時間設定を設定します。詳細は、CaoController クラスの「TIME コマンド」を参照してください。

### 3.3.2.3.15. LISTCYCLE コマンド

試験サイクル数を取得します。詳細は、CaoController クラスの「LISTCYCLE コマンド」を参照してください。

### 3.3.2.3.16. CYCLE コマンド

試験パターンの繰り返し回数(サイクル数)を設定します。詳細は、CaoController クラスの「CYCLE コマンド」を参照してください。

### 3.3.2.3.17. LISTSTARTPOSITION コマンド

試験開始位置設定を取得します。詳細は、CaoController クラスの「LISTSTARTPOSITION コマンド」を参照してください。

### 3.3.2.3.18. STARTPOSITION コマンド

試験開始位置を設定します。詳細は、CaoController クラスの「STARTPOSITION コマンド」を参照してください。

### 3.3.2.3.19. LISTPAUSE コマンド

試験中断予約設定を取得します。

項目	型説明		
戻り値	VT_ARRAY   VT_VARIANT		
	0	VT_BSTR	中断予約設定 設定による応答は以下のようになります。 <ul style="list-style-type: none"> <li>・ ON : 試験中断予約有効の場合</li> <li>・ OFF : 試験中断予約無効の場合</li> </ul>
	1	VT_I4	中断予約サイクル数。試験中断予約無効の場合は0が取得されます。

#### 使用例

' 拡張ボードの追加

```
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

' 参照モードにて試験パターンをオープン

```
Dim result As String
result = caoExt.Execute("PRGMREAD", Array(1,0))
```

' LISTPAUSEコマンド実行

```
Dim values As Variant
values = caoExt.Execute("LISTPAUSE")
```

' 中断予約設定を取得

```
Dim val1 As String
val1 = values(0)
```

' 中断予約サイクル数を取得

```
Dim val2 As Long
val1 = values(1)
```

### 3.3.2.3.20. PAUSE コマンド

試験中断予約を設定します。

項目	型説明	
引数	VT_I4	試験中断予約有効を指定する場合、中断予約サイクル数を指定します。
	または	
	VT_BSTR	試験中断予約無効を指定する場合、「OFF」を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

' 拡張ボードの追加

```
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

' 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoExt.Execute("PRGMWRITE", 2)
```

' PAUSEコマンド実行

' 試験中断予約有効を指定する場合

```
Dim param As Long
param = 10
result = caoExt.Execute("PAUSE", param)
```

' 試験中断予約無効を指定する場合

```
result = caoExt.Execute("PAUSE", "OFF")
```

### 3.3.2.3.21. LISTG-SOAK コマンド

安定時間制御設定を取得します。詳細は、CaoController クラスの「LISTG-SOAK コマンド」を参照

してください。

### 3.3.2.3.22. G-SOAK コマンド

安定時間制御モードを設定します。詳細は、CaoController クラスの「G-SOAK コマンド」を参照してください。

### 3.3.2.3.23. LISTEND コマンド

試験終了条件設定を取得します。

項目	型説明	
戻り値	VT_BSTR	試験終了条件 取得した応答の詳細は以下のようになります。 <ul style="list-style-type: none"> <li>・ OFF : 停止</li> <li>・ HEATRETURN : 液温戻し後停止</li> <li>・ SETUP : 準備</li> <li>・ DEFROST : 除霜後停止</li> <li>・ DRY : 乾燥後停止</li> <li>・ AMB : 常温後停止</li> </ul>

#### 使用例

' 拡張ボードの追加

```
Dim caoExt As CaoExtension
```

```
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

' 参照モードにて試験パターンをオープン

```
Dim result As String
```

```
result = caoExt.Execute("PRGMREAD", Array(1,0))
```

' LISTENDコマンド実行

```
Dim val As String
```

```
val = caoExt.Execute("LISTEND")
```

### 3.3.2.3.24. END コマンド

終了条件を設定します。

項目	型説明	
引数	VT_BSTR	試験終了条件を指定します。指定可能な値は、「LISTEND コマンド」の戻り値と同様です。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")

' 編集モードにて試験パターンをオープン
Dim result As String
result = caoExt.Execute("PRGMWRITE", 2)

' ENDコマンド実行
result = caoExt.Execute("END", "OFF")
```

**3.3.2.3.25. LISTNAME コマンド**

試験パターン名を取得します。詳細は、CaoController クラスの「LISTNAME コマンド」を参照してください。

**3.3.2.3.26. NAME コマンド**

試験パターン名を設定します。詳細は、CaoController クラスの「NAME コマンド」を参照してください。

**3.3.2.3.27. LISTSENSOR コマンド**

センサー位置設定を取得します。詳細は、CaoController クラスの「LISTSENSOR コマンド」を参照してください。

**3.3.2.3.28. SENSOR コマンド**

センサー位置を設定します。詳細は、CaoController クラスの「SENSOR コマンド」を参照してください。

**3.3.2.3.29. LISTDEFROST コマンド**

除霜モード設定(除霜条件)を取得します。詳細は、CaoController クラスの「LISTDEFROST コマンド」を参照してください。

**3.3.2.3.30. DEFROST コマンド**

除霜モードを設定します。詳細は、CaoController クラスの「DEFROST コマンド」を参照してください。

**3.3.2.3.31. LISTTEMP-L-DEFROST コマンド**

除霜待機位置設定を取得します。

項目	型説明	
戻り値	VT_BSTR	除霜待機位置 設定による応答は以下のようになります。 ・ H : 高温待機 ・ A : 常温待機

**使用例**

’ 拡張ボードの追加

```
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

’ 参照モードにて試験パターンをオープン

```
Dim result As String
result = caoExt.Execute("PRGMREAD", Array(1,0))
```

’ LISTTEMP-L-DEFROSTコマンド実行

```
Dim val As String
val = caoExt.Execute("LISTTEMP-L-DEFROST")
```

3.3.2.3.32. TEMP-L-DEFROST コマンド

除霜待機位置を設定します。

項目	型説明	
引数	VT_BSTR	除霜待機位置を指定します。 以下のいずれかを指定してください。 ・ H : 除霜待機位置を高温に設定します。 ・ A : 除霜待機位置を常温に設定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

**使用例**

’ 拡張ボードの追加

```
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

’ 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoExt.Execute("PRGMWRITE", 2)
```

’ TEMP-L-DEFROSTコマンド実行

```
result = caoExt.Execute("TEMP-L-DEFROST", "A")
```

**3.3.2.3.33. LISTQ-EXP コマンド**

さらし時間短縮設定を取得します。詳細は、CaoController クラスの「LISTQ-EXP コマンド」を参照してください。

**3.3.2.3.34. Q-EXP コマンド**

さらし時間短縮モードを設定します。詳細は、CaoController クラスの「Q-EXP コマンド」を参照してください。

**3.3.2.3.35. LISTTEMPLIMIT コマンド**

温度過昇/温度過冷設定値を取得します。詳細は、CaoController クラスの「LISTTEMPLIMIT コマンド」を参照してください。

**3.3.2.3.36. TEMPLIMIT コマンド**

温度過昇/温度過冷設定を設定します。詳細は、CaoController クラスの「TEMPLIMIT コマンド」を参照してください。

**3.3.2.3.37. LISTAUXCOOLER コマンド**

補助冷却設定を取得します。詳細は、CaoController クラスの「LISTAUXCOOLER コマンド」を参照してください。

**3.3.2.3.38. AUXCOOLER コマンド**

補助冷却モードを設定します。詳細は、CaoController クラスの「AUXCOOLER コマンド」を参照してください。

**3.3.2.3.39. LISTDRYMODESET コマンド**

前後処理設定値を取得します。

項目	型説明		
戻り値	VT_ARRAY   VT_VARIANT		
	0	VT_I4	温度設定値
	1	VT_BSTR	時間設定値 (HH:mm)

**使用例**

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

```
' 参照モードにて試験パターンをオープン
```

```
Dim result As String
result = caoExt.Execute("PRGMREAD", Array(1,0))
```

#### ' LISTDRYMODESET コマンド実行

```
Dim values As Variant
values = caoExt.Execute("LISTDRYMODESET")
```

#### ' 温度設定値を取得

```
Dim val1 As Long
val1 = values(0)
```

#### ' 時間設定値を取得

```
Dim val2 As String
val2 = values(1)
```

### 3.3.2.3.40. DRYMODESET コマンド

前後処理温度・時間設定を設定します。

項目	型説明		
引数	VT_ARRAY   VT_VARIANT		
	0	VT_I4	温度設定値を指定します。
	1	VT_BSTR	時間設定値(HH:mm)を指定します。
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。	

#### 使用例

#### ' 拡張ボードの追加

```
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

#### ' 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoExt.Execute("PRGMWRITE", 2)
```

#### ' DRYMODESET コマンド実行

```
Dim param(1) As Variant
param(0) = CLng(50)
param(1) = "00:30"
result = caoExt.Execute("DRYMODESET", param)
```

### 3.3.2.3.41. LISTDRYMODE コマンド

前後処理モード設定を取得します。詳細は、GaoController クラスの「LISTDRYMODE コマンド」を参照してください。

**3.3.2.3.42. DRYMODE コマンド**

前後処理モードを設定します。詳細は、CaoController クラスの「DRYMODE コマンド」を参照してください。

**3.3.2.3.43. LIST-PRE-ECONOMY コマンド**

エコ運転条件設定を取得します。詳細は、CaoController クラスの「LIST-PRE-ECONOMY コマンド」を参照してください。

**3.3.2.3.44. PRE-ECONOMY コマンド**

エコ運転条件を設定します。詳細は、CaoController クラスの「PRE-ECONOMY コマンド」を参照してください。

**3.3.2.3.45. LISTRELAY1 コマンド**

外部出力1の設定値を取得します。詳細は、CaoController クラスの「LISTRELAY1 コマンド」を参照してください。

**3.3.2.3.46. RELAY1 コマンド**

外部出力1を設定します。詳細は、CaoController クラスの「RELAY1 コマンド」を参照してください。

**3.3.2.3.47. LISTRELAY2 コマンド**

外部出力2の設定値を取得します。詳細は、CaoController クラスの「LISTRELAY2 コマンド」を参照してください。

**3.3.2.3.48. RELAY2 コマンド**

外部出力2を設定します。詳細は、CaoController クラスの「RELAY2 コマンド」を参照してください。

**3.3.2.3.49. ECONOMY コマンド**

節約運転を設定します。

項目	型説明	
引数	VT_BSTR	節約運転設定を指定します。 以下のいずれかを指定してください。 <ul style="list-style-type: none"> <li>・ ON : 節約運転機能有効</li> <li>・ OFF : 節約運転機能無効</li> </ul>

項目	型説明	
戻り値	VT_BSTR	コマンドが正常に実行された場合、機器からの応答文字列がそのまま格納されます。

#### 使用例

' 拡張ボードの追加

```
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

' 編集モードにて試験パターンをオープン

```
Dim result As String
result = caoExt.Execute("PRGMWRITE", 2)
```

' ECONOMYコマンド実行

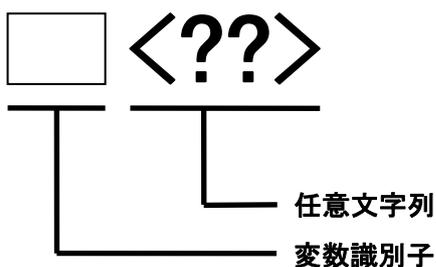
```
result = caoExt.Execute("ECONOMY", "ON")
```

### 3.4. 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。複数変数を登録（オプションのみ変更したい場合等に有用）するために任意の文字列を付与することが可能です。

変数名に任意文字列を付与するための書式を以下に示します。

#### 複数変数共通指定書式



#### 3.4.1. CaoController クラス変数

CaoController クラスの「@MAKER\_NAME」、 「@VERSION」を除いた変数は、イーサネット接続した場合のみ有効な変数です。使用例は各コマンドの詳細で記述しています。

表 3-6 CaoController クラス変数一覧

変数名	説明	Value		参照
		get	put	
<b>プロバイダ情報関連変数</b>				
@MAKER_NAME	メーカー名を取得します。	○	—	P. 86
@VERSION	プロバイダバージョンを取得します。	○	—	P. 87
@LASTDEVERR	最後に発生したデバイスエラーを取得します。	○	—	P. 87
<b>機器設定関連変数</b>				
@ROM	コントローラの ROM バージョンを取得します。	○	—	P. 88
@ROMDISP	表示器の ROM バージョンを取得します。	○	—	P. 89
@POWERCUT	停復電動作モードを取得します。	○	—	P. 89
@STOPMODE	装置停止モードを取得します。	○	—	P. 89
@DRYTEMP	乾燥運転用設定温度を取得します。	○	—	P. 90
@AMBTIME	常温運転用設定時間を取得します。	○	—	P. 90
@KEYPROTECT	キープロテクト状態を取得/設定します。	○	○	P. 91
<b>機器の運転関連変数</b>				
@MODE	運転モードを取得/設定します。	○	○	P. 91
@ASSIGN	試験パターンを取得/設定します。	○	○	P. 92
@TEMP	さらし温度を取得します。	○	—	P. 93
@PRE	予熱・予冷温度を取得します。	○	—	P. 93
@TIME	さらし残時間を取得します。	○	—	P. 94
@CYCLE	試験サイクル数を取得します。	○	—	P. 94
@MODEL	機種 ID を取得します。	○	—	P. 95
@ALARM	アラーム状態を取得します。	○	—	P. 95
@PRESET-HALT-ON	中断予約状況を取得します。	○	—	P. 96
PRESET-HALT	中断予約カウンター情報を取得します。	○	—	P. 96

### 3.4.1.1. プロバイダ情報関連変数

本プロバイダが保持している情報を取得する変数です。

#### 3.4.1.1.1. @MAKER\_NAME

メーカー名を取得します。

**データ型**

型説明	
VT_BSTR	メーカー名

**使用例**

```

' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")

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

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

**3.4.1.1.2. @VERSION**

プロバイダバージョンを取得します。

**データ型**

型説明	
VT_BSTR	プロバイダバージョン *. *.*

**使用例**

```

' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")

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

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

**3.4.1.1.3. @LASTDEVERR**

最後に発生したデバイスエラーを取得します。

**データ型**

項目	型説明		
取得	VT_ARRAY   VT_VARIANT		
	0	VT_BSTR	エラーメッセージ

項目	型説明	
	1	VT_I4 エラーコード

#### 使用例

##### ' コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

##### ' 変数追加

```
Dim var As CaoVariable
Set var = caoCon.AddVariable("@LASTDEVERR")
```

##### ' 値取得

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

##### ' エラーメッセージ取得

```
Dim message As String
message = values(0)
```

##### ' エラーコード取得

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

### 3.4.1.2. 機器設定関連変数

機器に設定されている設定値を取得/設定する変数です。

#### 3.4.1.2.1. @ROM

コントローラの ROM バージョンを取得します。

#### データ型

取得するデータについては、CaoController クラスの「GetROM コマンド」の戻り値を参照してください。

#### 使用例

##### ' コントローラの追加

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

##### ' 変数追加

```
Dim var As CaoVariable
Set var = caoCon.AddVariable("@ROM")
```

##### ' 値取得

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

### 3.4.1.2.2. @ROMDISP

表示器の ROM バージョンを取得します。

#### データ型

取得するデータについては、CaoController クラスの「GetROMDISP コマンド」の戻り値を参照してください。

#### 使用例

```
' コントローラの追加
```

```
Dim caoCon As CaoController
```

```
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _  
                                "conn=eth:192.168.0.2")
```

```
' 変数追加
```

```
Dim var As CaoVariable
```

```
Set var = caoCon.AddVariable("@ROMDISP")
```

```
' 値取得
```

```
Dim val As String
```

```
val = var.value
```

### 3.4.1.2.3. @POWERCUT

停復電動作モードを取得します。

#### データ型

取得するデータについては、CaoController クラスの「GetPOWERCUT コマンド」の戻り値を参照してください。

#### 使用例

```
' コントローラの追加
```

```
Dim caoCon As CaoController
```

```
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _  
                                "conn=eth:192.168.0.2")
```

```
' 変数追加
```

```
Dim var As CaoVariable
```

```
Set var = caoCon.AddVariable("@POWERCUT")
```

```
' 値取得
```

```
Dim val As String
```

```
val = var.value
```

### 3.4.1.2.4. @STOPMODE

装置停止モードを取得します。

#### データ型

取得するデータについては、CaoController クラスの「GetSTOPMODE コマンド」の戻り値を参照して

ください。

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")

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

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

#### 3.4.1.2.5. @DRYTEMP

乾燥運転用設定温度を取得します。

#### データ型

取得するデータについては、CaoController クラスの「GetDRYTEMP コマンド」の戻り値を参照してください。

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")

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

' 値取得
Dim val As Long
val = var.value
```

#### 3.4.1.2.6. @AMBTIME

常温運転用設定時間を取得します。

#### データ型

取得するデータについては、CaoController クラスの「GetAMBTIME コマンド」の戻り値を参照してください。

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
```

```
"conn=eth:192.168.0.2")
```

```
' 変数追加
Dim var As CaoVariable
Set var = caoCon.AddVariable("@AMBTIME")
' 値取得
Dim val As String
val = var.value
```

### 3.4.1.2.7. @KEYPROTECT

キープロテクト状態を取得/設定します。取得/設定するデータの詳細については、CaoController クラスの「GetKEYPROTECT コマンド」の戻り値と「SetKEYPROTECT コマンド」の引数を参照してください。

#### データ型

項目	型説明	
取得/設定	VT_BSTR	キープロテクト状態

#### 使用例

```
' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
' 変数追加
Dim var As CaoVariable
Set var = caoCon.AddVariable("@KEYPROTECT")
' 値取得
Dim val As String
val = var.value
' 値設定
var.value = "OFF"
```

### 3.4.1.3. 機器運転関連変数

機器の現在の状態を取得/設定する変数です。

#### 3.4.1.3.1. @MODE

運転モードを取得/設定します。取得データの詳細についてはCaoController クラスの「GetMODE コマンド」の戻り値を、設定するデータの詳細については「SetMODE コマンド」の引数を参照してください。

#### データ型

項目	型説明		
取得/設定	VT_ARRAY   VT_BSTR		
	0	VT_BSTR	現在の装置の運転状態

項目	型説明		
	1	VT_BSTR	完了予定日付 (MM/dd)
	2	VT_BSTR	完了予定時間 (HH:mm)

※設定する際には完了予定日付と完了予定時間は無視されます。

**使用例**

’ **コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ **変数追加**

```
Dim var As CaoVariable
Set var = caoCon.AddVariable("@MODE")
```

’ **値取得**

```
Dim values() As String
values = var.value
```

’ **値設定**

```
var.value = "OFF"
```

**3.4.1.3.2. @ASSIGN**

試験パターンを取得/設定します。

**データ型**

項目	型説明		
取得/設定	VT_ARRAY   VT_VARIANT		
	0	VT_I4	試験パターンのパターン番号
	1	VT_BSTR	試験パターン名称. 最大 14 文字以内で取得されます. ただし, 設定する際にはこの項目は無視されます.

**使用例**

’ **コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

’ **変数追加**

```
Dim var As CaoVariable
Set var = caoCon.AddVariable("@ASSIGN")
```

’ **値取得**

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

---

**’ パターン番号取得**

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

**’ 試験パターン名称取得**

```
Dim patternName As String  
patternName = values(1)
```

**’ 値設定**

```
Dim patternNo As Long  
patternNo = 2  
var.value = patternNo
```

---

### 3.4.1.3.3. @TEMP

さらし温度を取得します。

**データ型**

取得するデータについては、CaoController クラスの「GetTEMP コマンド」の戻り値を参照してください。

**使用例****’ コントローラの追加**

```
Dim caoCon As CaoController  
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _  
                                "conn=eth:192.168.0.2")
```

**’ 変数追加**

```
Dim var As CaoVariable  
Set var = caoCon.AddVariable("@TEMP")
```

**’ 値取得**

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

**’ 最新の温度を取得**

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

---

### 3.4.1.3.4. @PRE

予熱・予冷温度を取得します。

**データ型**

取得するデータについては、CaoController クラスの「GetPRE コマンド」の戻り値を参照してください。

**使用例**

---

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ 変数追加**

```
Dim var As CaoVariable
Set var = caoCon.AddVariable("@PRE")
```

**’ 値取得**

```
Dim values() As Long
values = var.value
```

---

### 3.4.1.3.5. @TIME

さらし残時間を取得します。

**データ型**

取得するデータについては、CaoController クラスの「GetTIME コマンド」の戻り値を参照してください。

**使用例**

---

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ 変数追加**

```
Dim var As CaoVariable
Set var = caoCon.AddVariable("@TIME")
```

**’ 値取得**

```
Dim values() As String
values = var.value
```

---

### 3.4.1.3.6. @CYCLE

試験サイクル数を取得します。

**データ型**

取得するデータについては、CaoController クラスの「GetCYCLE コマンド」の戻り値を参照してください。

**使用例**

---

**’ コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**’ 変数追加**

```
Dim var As CaoVariable
Set var = caoCon.AddVariable("@CYCLE")
```

---

---

**値取得**

```
Dim values() As Long
values = var.value
```

---

### 3.4.1.3.7. @MODEL

機種 ID を取得します。

**データ型**

取得するデータについては、CaoController クラスの「GetMODEL コマンド」の戻り値を参照してください。

**使用例****コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**変数追加**

```
Dim var As CaoVariable
Set var = caoCon.AddVariable("@MODEL")
```

**値取得**

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

---

### 3.4.1.3.8. @ALARM

アラーム状態を取得します。

**データ型**

取得するデータについては、CaoController クラスの「GetALARM コマンド」の戻り値を参照してください。

**使用例****コントローラの追加**

```
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv.espec.ThermalShock", "", _
    "conn=eth:192.168.0.2")
```

**変数追加**

```
Dim var As CaoVariable
Set var = caoCon.AddVariable("@ALARM")
```

**値取得**

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

**アラーム数分取得**

```
If Not IsEmpty(values) Then
    Dim i As Long
```

---

```

    For i = 0 To UBound(values)
        Dim val As Long
        val = values(i)
    Next i
End If

```

### 3.4.1.3.9. @PRESET-HALT-ON

中断予約状況を取得します。

#### データ型

取得するデータについては、CaoController クラスの「GetPRESET-HALT-ON コマンド」の戻り値を参照してください。

#### 使用例

```

' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")

' 変数追加
Dim var As CaoVariable
Set var = caoCon.AddVariable("@PRESET-HALT-ON")

' 値取得
Dim values() As String
values = var.value

```

### 3.4.1.3.10. PRESET-HALT

中断予約カウンター情報を取得します。

#### オプション

オプション	必須	説明	値範囲	デフォルト値
No	○	予約中断番号	1~	--

#### データ型

取得するデータについては、CaoController クラスの「GetPRESET-HALT コマンド」の戻り値を参照してください。

#### 使用例

```

' コントローラの追加
Dim caoCon As CaoController
Set caoCon = caoWor.AddController("Controller", "CaoProv. espec. ThermalShock", "", _
    "conn=eth:192.168.0.2")

' 変数追加
Dim var As CaoVariable

```

```

Set var = caoCon.AddVariable("PRESET-HALT", "No=1")
' 値取得
Dim values As Variant
values = var.value

' 実行カウンター数
Dim val1 As Long
val1 = values(0)
' 予約カウンター数
Dim val2 As Long
val2 = values(1)
' 名称
Dim val3 As String
val3 = values(2)
    
```

### 3.4.2. CaoExtension クラス変数

CaoExtension クラスの変数は、シリアル通信で接続されている場合のみ有効です。使用例は各コマンドの詳細で記述しています。

表 3-7 CaoExtension クラス変数一覧

変数名	説明	Value		参照
		get	put	
<b>プロバイダ情報関連変数</b>				
@LASTDEVERR	最後に発生したデバイスエラーを取得	○	—	P. 98
<b>機器設定関連変数</b>				
@ROM	コントローラの ROM バージョンを取得します。	○	—	P. 98
@ROMDISP	表示器の ROM バージョンを取得します。	○	—	P. 98
@POWERCUT	停復電動作モードを取得します。	○	—	P. 98
@STOPMODE	装置停止モードを取得します。	○	—	P. 99
@G-SOAKTEMP	安定時間制御用安定条件を取得します。	○	—	P. 99
@DRYTEMP	乾燥運転用設定温度を取得します。	○	—	P. 99
@AMBTIME	常温運転用設定時間を取得します。	○	—	P. 99
@KEYPROTECT	キープロテクト状態を取得/設定します。	○	○	P. 99
<b>機器の運転関連変数</b>				
@MODE	運転モードを取得/設定します。	○	○	P. 100
@ASSIGN	試験パターンを取得/設定します。	○	○	P. 100
@TEMP	さらし温度を取得します。	○	—	P. 101

変数名	説明	Value		参照
		get	put	
@PRE	予熱・予冷温度を取得します。	○	—	P. 101
@TIME	さらし残時間を取得します。	○	—	P. 101
@CYCLE	試験サイクル数を取得します。	○	—	P. 101
@MODEL	機種 ID を取得します。	○	—	P. 101
@ALARM	アラーム状態を取得します。	○	—	P. 101

### 3.4.2.1. プロバイダ情報関連変数

本プロバイダが保持している情報を取得する変数です。

#### 3.4.2.1.1. @LASTDEVERR

最後に発生したデバイスエラーを取得します。詳細は、CaoController クラスの変数「@LASTDEVERR」を参照してください。

### 3.4.2.2. 機器設定関連変数

機器に設定されている設定値を取得/設定する変数です。

#### 3.4.2.2.1. @ROM

コントローラの ROM バージョンを取得します。詳細は、CaoController クラスの変数「@ROM」を参照してください。

#### 3.4.2.2.2. @ROMDISP

表示器の ROM バージョンを取得します。詳細は、CaoController クラスの変数「@ROMDISP」を参照してください。

#### 3.4.2.2.3. @POWERCUT

停復電動作モードを取得します。

##### データ型

取得するデータについては、CaoExtension クラスの「GetPOWERCUT コマンド」の戻り値を参照してください。

##### 使用例

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' 変数追加
Dim var As CaoVariable
Set var = caoExt.AddVariable("@POWERCUT")
```

---

**’ 値取得**

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

---

**3.4.2.2.4. @STOPMODE**

装置停止モードを取得します。詳細は、CaoController クラスの変数「@STOPMODE」を参照してください。

**3.4.2.2.5. @G-SOAKTEMP**

安定時間制御用安定条件を取得します。

**データ型**

取得するデータについては、CaoExtension クラスの「GetG-SOAKTEMP コマンド」の戻り値を参照してください。

**使用例****’ 拡張ボードの追加**

```
Dim caoExt As CaoExtension  
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

**’ 変数追加**

```
Dim var As CaoVariable  
Set var = caoExt.AddVariable("@G-SOAKTEMP")
```

**’ 値取得**

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

---

**3.4.2.2.6. @DRYTEMP**

乾燥運転時の温度設定値を取得します。詳細は、CaoController クラスの変数「@DRYTEMP」を参照してください。

**3.4.2.2.7. @AMBTIME**

常温運転用設定時間を取得します。詳細は、CaoController クラスの変数「@AMBTIME」を参照してください。

**3.4.2.2.8. @KEYPROTECT**

キープロテクト状態を取得/設定します。詳細は、CaoController クラスの変数「@KEYPROTECT」を参照してください。

**3.4.2.3. 機器運転関連変数**

機器の現在の状態を取得/設定する変数です。

### 3.4.2.3.1. @MODE

運転モードを取得/設定します。詳細は、CaoController クラスの変数「@MODE」を参照してください。

### 3.4.2.3.2. @ASSIGN

試験パターンを取得/設定します。

#### データ型

項目	型説明		
取得/設定	VT_ARRAY   VT_VARIANT		
	0	VT_I4	試験パターンのパターン番号
	1	VT_I4	メモリ種別 以下のいずれかを指定してください。 ・ 0 : RAM ・ 1 : ROM
	2	VT_BSTR	試験名称. 14 文字以内で取得されます。ただし、設定する際にはこの項目は無視されます。

#### 使用例

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' 変数追加
Dim var As CaoVariable
Set var = caoExt.AddVariable("@ASSIGN")
' 値取得
Dim values As Variant
values = var.value

' パターン番号
Dim val1 As Long
val1 = values(0)

' メモリ種別
Dim val2 As Long
val2 = values(1)

' 試験名称
Dim val3 As String
val3 = values(2)

' 値設定
```

---

```
Dim pattern(2) As Long
pattern(0) = 1
pattern(1) = 0
var.value = pattern
```

---

#### 3.4.2.3.3. @TEMP

さらに温度を取得します。詳細は、CaoController クラスの変数「@TEMP」を参照してください。

#### 3.4.2.3.4. @PRE

予熱・予冷温度を取得します。詳細は、CaoController クラスの変数「@PRE」を参照してください。

#### 3.4.2.3.5. @TIME

さらに残時間を取得します。

##### データ型

取得するデータについては、CaoExtension クラスの「GetTIME コマンド」の戻り値を参照してください。

##### 使用例

---

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' 変数追加
Dim var As CaoVariable
Set var = caoExt.AddVariable("@TIME")
' 値取得
Dim values() As String
values = var.value
```

---

#### 3.4.2.3.6. @CYCLE

試験サイクル数を取得します。詳細は、CaoController クラスの変数「@CYCLE」を参照してください。

#### 3.4.2.3.7. @MODEL

機種 ID を取得します。詳細は、CaoController クラスの変数「@MODEL」を参照してください。

#### 3.4.2.3.8. @ALARM

アラーム状態を取得します。

##### データ型

取得するデータについては、CaoExtension クラスの「GetALARM コマンド」の戻り値を参照してください。

さい。

#### 使用例

```
' 拡張ボードの追加
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
' 変数追加
Dim var As CaoVariable
Set var = caoExt.AddVariable("@ALARM")
' 値取得
Dim values As Variant
values = var.value

' アラーム数分取得
If Not IsEmpty(values) Then
    Dim i As Long
    For i = 0 To UBound(values)
        Dim val As Long
        val = values(i)
    Next i
End If
```

## 4. espec ThermalShock プロバイダによるプログラミング

espec ThermalShock プロバイダでは、クライアント PC と RS-485 対応デバイスまたは、Ethernet 対応デバイスを接続することができます。

初めに、以下の手順でクライアント PC と RS-485 対応デバイスをシリアル通信で接続することができます。

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

RS-485 対応デバイスに接続した後は、CaoExtension 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、RS-485 対応デバイスの情報にアクセスすることができます。

次に、以下の手順でクライアント PC と Ethernet 対応デバイスをイーサネット通信で接続することができます。

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

Ethernet 対応デバイスに接続した後は、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、Ethernet 対応デバイスの情報にアクセスすることができます。

### 4.1. RS-485 対応デバイスの試験パターンを変更するサンプルプログラミング

ここでは例として試験パターンを変更するサンプルプログラムを示します。表 4-1 にサンプルプログラムの要件を、図 4-1 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 4-1 サンプルプログラムの要件

要件	説明
接続先	シリアルで接続する
	接続先 COM ポートは COM1
処理内容	現在の試験パターンを取得する
	試験パターン No2 に変更する

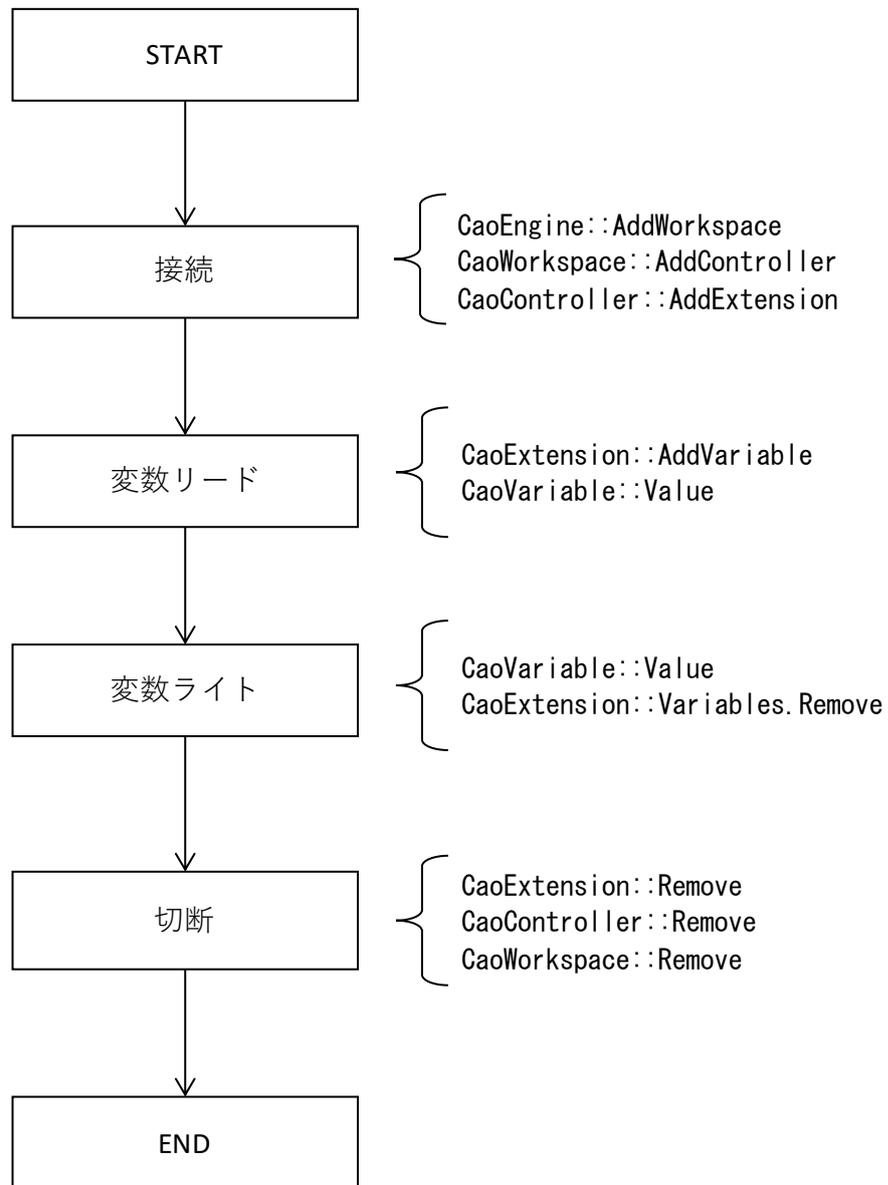


図 4-1 試験パターン変更の流れ

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**      **ChangeTestPattern\_TSA-1.vb**

' オブジェクト

```

Dim caoEng As CaoEngine
Dim caoWor As CaoWorkspace
Dim caoCon As CaoController
Dim caoExt As CaoExtension
    
```

```
Private Sub Main
```

```
    ' 接続
```

```
    Call Connect
```

```
    ' CaoVariable オブジェクトの生成
```

```
    Dim varASSIGN As CaoVariable
```

```
    Set varASSIGN = caoExt.AddVariable("@ASSIGN")
```

```
    ' 試験パターンを取得
```

```
    Dim values As Variant
```

```
    values = varASSIGN.value
```

```
    ' 試験パターンを設定
```

```
    Dim pattern(2) As Long
```

```
    pattern(0) = 2
```

```
    pattern(1) = 0
```

```
    varASSIGN.value = pattern
```

```
    ' CaoExtension から CaoVariable を削除
```

```
    Call caoExt.Variables.Remove(varASSIGN.Index)
```

```
    ' CaoVariable の消去
```

```
    Set varASSIGN = Nothing
```

```
    ' 切断
```

```
    Call Disconnect
```

```
End Sub
```

```
' 接続メソッド
```

```
Private Sub Connect()
```

```
    ' CaoEngine オブジェクトの生成
```

```
    Set caoEng = New CaoEngine
```

```
    ' CaoWorkspace オブジェクトの生成
```

```
    Set caoWor = caoEng.AddWorkspace("Workspace", "")
```

```
    ' CaoController オブジェクトの生成
```

```
    Set caoCon = caoWor.AddController("Controller", _  
                                     "CaoProv.espec.ThermalShock", _  
                                     "", _  
                                     "conn=com:1")
```

```
    ' CaoExtension オブジェクトの生成
```

```
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

```
End Sub
```

```
' 切断メソッド
```

```
Private Sub Disconnect()
```

```
    ' CaoController から CaoExtension を削除
```

```
    Call caoCon.Extensions.Remove(caoExt.Index)
```

```
    ' CaoExtension の消去
```

```

Set caoExt = Nothing

' CaoWorkspace から CaoController を削除
Call caoWor.Controllers.Remove(caoCon.Index)
' CaoController の消去
Set caoCon = Nothing

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

' CaoEngine の消去
Set caoEng = Nothing
End Sub

```

#### 4.1.1.1. RS-485 対応デバイスと接続

RS-485 対応デバイスと接続するためには、以下の手順を行います。

- (1) オブジェクトを保持するための変数を用意します。デバイス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オブジェクトとCaoExtensionオブジェクトです。CaoWorkspaceオブジェクトは、CaoControllerオブジェクトをCaoWorkspacesから取得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 caoEng As CaoEngine      ' CaoEngineオブジェクト用の変数
Dim caoWor As CaoWorkspace  ' CaoWorkspaceオブジェクト用の変数
Dim caoCon As CaoController ' CaoControllerオブジェクト用の変数
Dim caoExt As CaoExtension  ' CaoExtension オブジェクト用の変数
Dim varASSIGN 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 caoEng = New CaoEngine

```

- (3) CaoWorkspaceオブジェクトを取得もしくは生成します。CaoEngineオブジェクトを生成すると、デフォルトでCaoWorkspacesオブジェクトとCaoWorkspaceオブジェクトを1つずつ生成しています。以下にCaoWorkspaceオブジェクトを新しく生成するコード例とデフォルトのCaoWorkspaceを示します。

```

' CaoWorkspace オブジェクトの生成
Set caoWor = caoEng.AddWorkspace("Workspace", "")

```

- (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。espec ThermalShockプロバイダでは、接続先のCOMポートをオプションで指定します。以下にコード例を示します。

---

**’ CaoController オブジェクトの生成**

```
Set caoCon = caoWor.AddController("Controller", _  
                                "CaoProv. espec. ThermalShock", _  
                                "" , _  
                                "conn=com:1")
```

---

- (5) CaoExtensionオブジェクトを生成します。CaoExtensionオブジェクトを生成するには、使用するためのパラメータを設定します。espec ThermalShockプロバイダでは、接続先のアドレスをオプションで指定します。以下にコード例を示します。

---

**’ CaoExtensionオブジェクトの生成**

```
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

---

#### 4.1.1.2. 変数の取得/設定

変数の値を取得/設定するには、アクセスしたい変数の CaoVariable オブジェクトを生成し、CaoVariable オブジェクトの Value プロパティを参照/設定します。Value プロパティを参照/設定する場合は、Value プロパティに合わせた型変数を用意する必要があります。以下にコード例を示します。

---

**’ CaoVariableオブジェクトの生成**

```
Dim varASSIGN As CaoVariable  
Set varASSIGN = caoExt.AddVariable("@ASSIGN")
```

**’ 試験パターンを取得**

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

**’ 試験パターンを設定**

```
Dim pattern(1) As Long  
pattern(0) = 2  
pattern(1) = 0  
varASSIGN.value = pattern
```

**’ CaoExtension から CaoVariable を削除**

```
Call caoExt.Variables.Remove(varASSIGN.Index)
```

**’ CaoVariable の消去**

```
Set varASSIGN = Nothing
```

---

#### 4.1.1.3. RS-485 対応デバイスと切断

RS-485 対応デバイス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 CaoExtension を削除
Call caoCon. Extensions. Remove(caoExt. Index)
' CaoExtension の消去
Set caoExt = Nothing
' CaoWorkspace から CaoController を削除
Call caoWor. Controllers. Remove(caoCon. Index)
' CaoController の消去
Set caoCon = Nothing
' CaoEngine から CaoWorkspace を削除
Call caoEng. Workspaces. Remove(caoWor. Index)
' CaoWorkspace の消去
Set caoWor = Nothing
' CaoEngine の消去
Set caoEng = Nothing
    
```

#### 4. 2. RS-485 対応デバイスの試験パターンをモニタリングするサンプルプログラミング

ここでは例として試験パターンをモニタリングするサンプルプログラムを示します。表 4-2 にサンプルプログラムの要件を、図 4-2 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 4-2 サンプルプログラムの要件

要件	説明
接続先	シリアルで接続する
	接続先 COM ポートは COM1
処理内容	参照モードにて試験パターン No3 をオープンします
	さらし温度設定値を取得します
	さらし時間設定値を取得します
	参照モードにて試験パターン No3 をクローズします

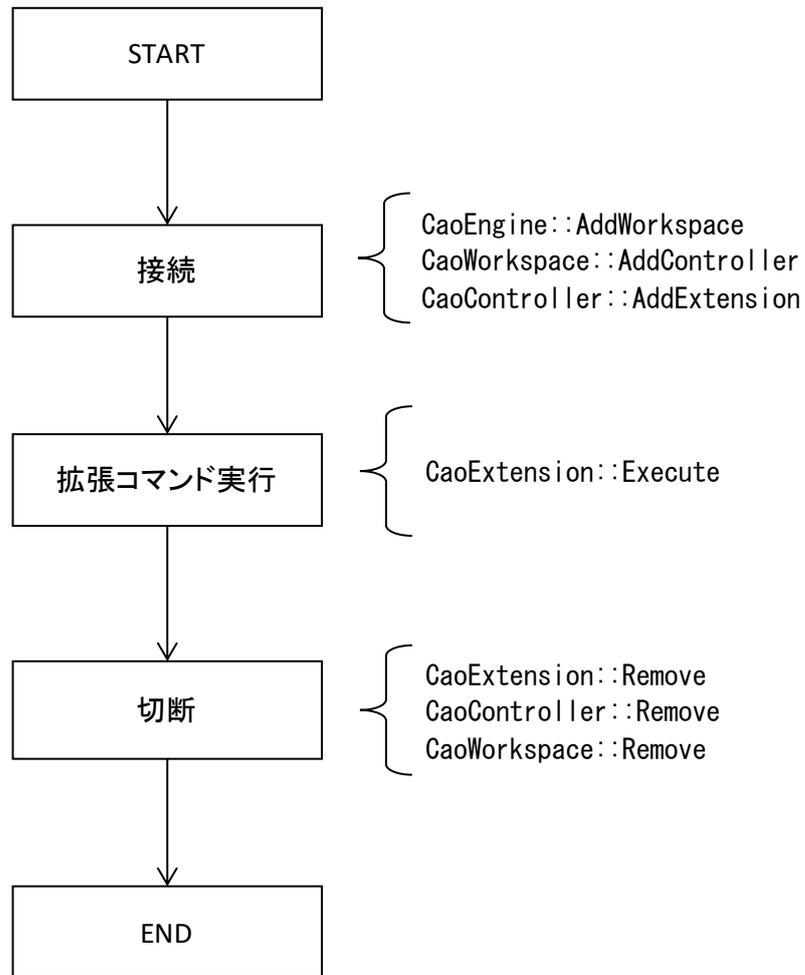


図 4-2 試験パターンモニタリングの流れ

以降の節から具体的なコードを示します。

#### 4.2.1. 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。

<b>Sample</b>	<b>MonitoringTestPattern_TSA-1. vb</b>
---------------	--

```

' オブジェクト
Dim caoEng As CaoEngine
Dim caoWor As CaoWorkspace
Dim caoCon As CaoController
Dim caoExt As CaoExtension

Private Sub Main
    ' 接続
    Call Connect

    Dim result As String
    ' 参照モードにて試験パターンをオープン
  
```

```
result = caoExt.Execute("PRGMREAD", Array(3, 0))

' さらし温度設定値を取得
Dim tempVal() As Long
tempVal = caoExt.Execute("LISTTEMP")

' さらし時間設定値を取得
Dim timeVal() As String
timeVal = caoExt.Execute("LISTTIME")

' 参照モードにて試験パターンをクローズ
result = caoExt.Execute("PRGMREADEND")

' 切断
Call Disconnect
End Sub

' 接続メソッド
Private Sub Connect()
' CaoEngine オブジェクトの生成
Set caoEng = New CaoEngine
' CaoWorkspace オブジェクトの生成
Set caoWor = caoEng.AddWorkspace("Workspace", "")
' CaoController オブジェクトの生成
Set caoCon = caoWor.AddController("Controller", _
                                "CaoProv. espec. ThermalShock", _
                                "", _
                                "conn=com:1")
' CaoExtension オブジェクトの生成
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
End Sub

' 切断メソッド
Private Sub Disconnect()
' CaoController から CaoExtension を削除
Call caoCon.Extensions.Remove(caoExt.Index)
' CaoExtension の消去
Set caoExt = Nothing

' CaoWorkspace から CaoController を削除
Call caoWor.Controllers.Remove(caoCon.Index)
' CaoController の消去
Set caoCon = Nothing

' CaoEngine から CaoWorkspace を削除
Call caoEng.Workspaces.Remove(caoWor.Index)
' CaoWorkspace の消去
Set caoWor = Nothing
```

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

#### 4.2.1.1. RS-485 対応デバイスと接続

RS-485 対応デバイスとの接続は 4.1.1.1 RS-485 対応デバイスと接続を参照して下さい。

#### 4.2.1.2. 試験パターンのモニタリング

CaoExtension オブジェクトの Execute メソッドを実行します。Execute メソッド実行時の引数ならびに Execute メソッド実行後の戻り値を参照する場合は、Execute メソッドに合わせた型変数を用意する必要があります。以下にコード例を示します。

```
Dim result As String
' 参照モードにて試験パターンをオープン
result = caoExt.Execute("PRGMREAD", Array(3,0))

' さらに温度設定値を取得
Dim tempVal() As Long
tempVal = caoExt.Execute("LISTTEMP")

' さらに時間設定値を取得
Dim timeVal() As String
timeVal = caoExt.Execute("LISTTIME")

' 参照モードにて試験パターンをクローズ
result = caoExt.Execute("PRGMREADEND")
```

#### 4.2.1.3. RS-485 対応デバイスと切断

RS-485 対応デバイスとの切断は 4.1.1.3 RS-485 対応デバイスと切断を参照して下さい。

### 4.3. Ethernet 対応デバイスの試験パターンを変更するサンプルプログラミング

ここでは例として試験パターンを変更するサンプルプログラムを示します。表 4-3 にサンプルプログラムの要件を、図 4-3 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 4-3 サンプルプログラムの要件

要件	説明
接続先	イーサネット (TCP/IP) で接続する
	接続先 IP アドレスは 192.168.0.2
処理内容	現在の試験パターンを取得する
	試験パターン No4 に変更する

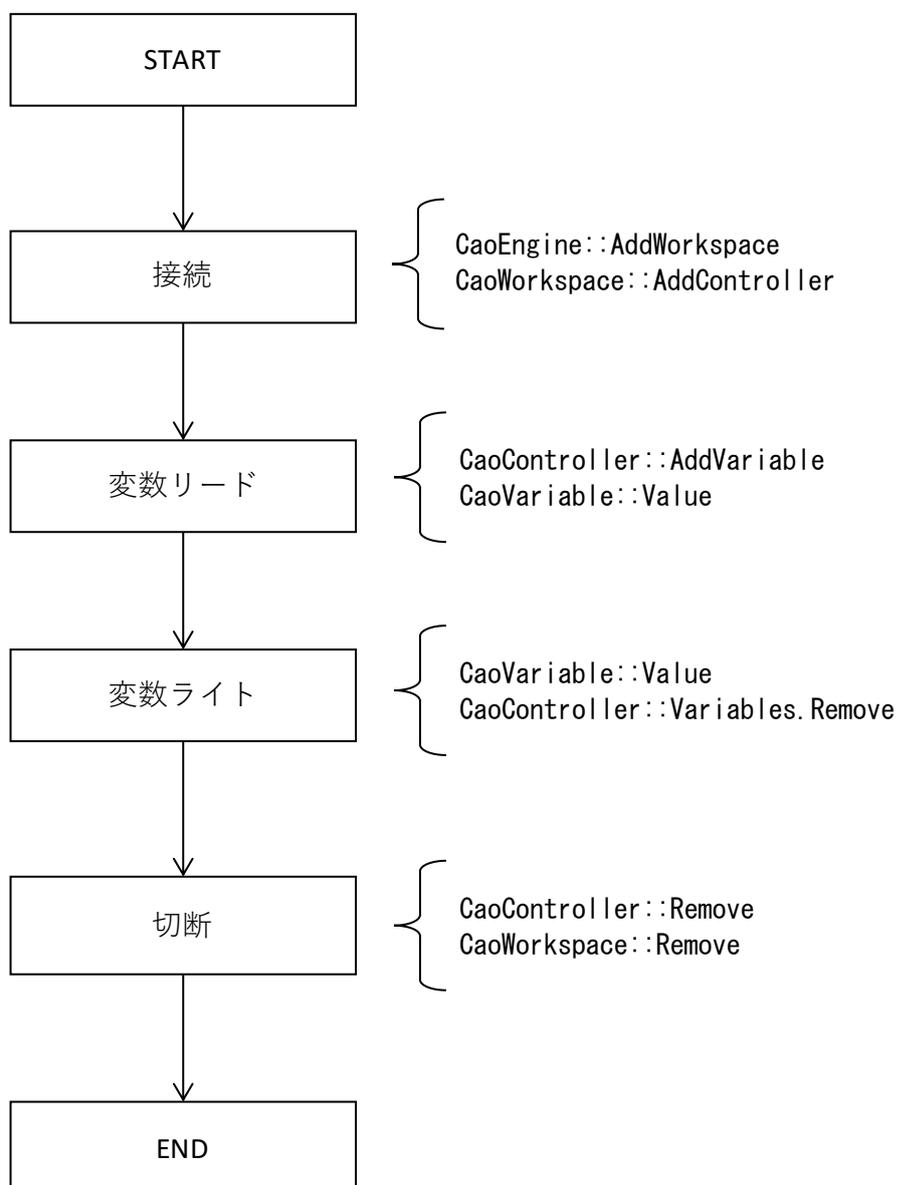


図 4-3 試験パターン変更の流れ

以降の節から具体的なコードを示します。

### 4.3.1. 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** `ChangeTestPattern_TSA-3.vb`

' オブジェクト

```

Dim caoEng As CaoEngine
Dim caoWor As CaoWorkspace
Dim caoCon As CaoController
    
```

```
Private Sub Main
```

```
    ' 接続
```

```
    Call Connect
```

```
    ' CaoVariable オブジェクトの生成
```

```
    Dim varASSIGN As CaoVariable
```

```
    Set varASSIGN = caoCon.AddVariable("@ASSIGN")
```

```
    ' 試験パターンを取得
```

```
    Dim values As Variant
```

```
    values = varASSIGN.value
```

```
    ' 試験パターンを設定
```

```
    varASSIGN.value = 4
```

```
    ' CaoExtension から CaoVariable を削除
```

```
    Call caoCon.Variables.Remove(varASSIGN.Index)
```

```
    ' CaoVariable の消去
```

```
    Set varASSIGN = Nothing
```

```
    ' 切断
```

```
    Call Disconnect
```

```
End Sub
```

```
' 接続メソッド
```

```
Private Sub Connect()
```

```
    ' CaoEngine オブジェクトの生成
```

```
    Set caoEng = New CaoEngine
```

```
    ' CaoWorkspace オブジェクトの生成
```

```
    Set caoWor = caoEng.AddWorkspace("Workspace", "")
```

```
    ' CaoController オブジェクトの生成
```

```
    Set caoCon = caoWor.AddController("Controller", _  
                                     "CaoProv.espec.ThermalShock", _  
                                     "", _  
                                     "conn=eth:192.168.0.2")
```

```
End Sub
```

```
' 切断メソッド
```

```
Private Sub Disconnect()
```

```
    ' CaoWorkspace から CaoController を削除
```

```
    Call caoWor.Controllers.Remove(caoCon.Index)
```

```
    ' CaoController の消去
```

```
    Set caoCon = Nothing
```

```
    ' CaoEngine から CaoWorkspace を削除
```

```
    Call caoEng.Workspaces.Remove(caoWor.Index)
```

```
    ' CaoWorkspace の消去
```

```
    Set caoWor = Nothing
```

```

' CaoEngine の消去
Set caoEng = Nothing
End Sub

```

---

#### 4.3.1.1. Ethernet 対応デバイスと接続

Ethernet 対応デバイスと接続するためには、以下の手順を行います。

- (1) オブジェクトを保持するための変数を用意します。デバイス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、CaoControllerオブジェクトをCaoWorkspacesから取得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 caoEng As CaoEngine           ' CaoEngineオブジェクト用の変数
Dim caoWor As CaoWorkspace        ' CaoWorkspaceオブジェクト用の変数
Dim caoCon As CaoController       ' CaoControllerオブジェクト用の変数
Dim varASSIGN 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 caoEng = New CaoEngine

```

---

- (3) CaoWorkspaceオブジェクトを取得もしくは生成します。CaoEngineオブジェクトを生成すると、デフォルトでCaoWorkspacesオブジェクトとCaoWorkspaceオブジェクトを1つずつ生成しています。以下にCaoWorkspaceオブジェクトを新しく生成するコード例とデフォルトのCaoWorkspaceを示します。

```

' CaoWorkspace オブジェクトの生成
Set caoWor = caoEng.AddWorkspace("Workspace", "")

```

---

- (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。espec ThermalShockプロバイダでは、接続先のCOMポートをオプションで指定します。以下にコード例を示します。

```

' CaoController オブジェクトの生成
Set caoCon = caoWor.AddController("Controller", _
    "CaoProv. espec. ThermalShock", _
    "", _
    "conn=eth:192.168.0.2")

```

---

#### 4.3.1.2. 変数の取得/設定

変数の値を取得/設定するには、アクセスしたい変数の CaoVariable オブジェクトを生成し、CaoVariable オブジェクトの Value プロパティを参照/設定します。Value プロパティを参照/設定する場合は、Value プロパティに合わせた型変数を用意する必要があります。以下にコード例を示します。

```
' CaoWorkspaceからCaoControllerを削除
Dim varASSIGN As CaoVariable
Set varASSIGN = caoCon.AddVariable("@ASSIGN")

' 試験パターンを取得
Dim values As Variant
values = varASSIGN.value

' 試験パターンを設定
Dim patternNo As Long
patternNo = 4
varASSIGN.value = patternNo

' CaoExtensionからCaoVariableを削除
Call caoCon.Variables.Remove(varASSIGN.Index)
' CaoVariableの消去
Set varASSIGN = Nothing
```

#### 4.3.1.3. Ethernet 対応デバイスと切断

Ethernet 対応デバイスと切断する場合には、生成したオブジェクトを消去すると共に、オブジェクトを管理するコレクションクラスから消去するオブジェクトを削除します。以下にコード例を示します。

```
' CaoWorkspaceからCaoControllerを削除
Call caoWor.Controllers.Remove(caoCon.Index)
' CaoControllerの消去
Set caoCon = Nothing
' CaoEngineからCaoWorkspaceを削除
Call caoEng.Workspaces.Remove(caoWor.Index)
' CaoWorkspaceの消去
Set caoWor = Nothing
' CaoEngineの消去
Set caoEng = Nothing
```

#### 4.4. Ethernet 対応デバイスの試験パターンをモニタリングするサンプルプログラミング

ここでは例として試験パターンをモニタリングするサンプルプログラムを示します。表 4-4 にサンプルプログラムの要件を、図 4-4 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 4-4 サンプルプログラムの要件

要件	説明
接続先	イーサネット (TCP/IP) で接続する
	接続先 IP アドレスは 192.168.0.2
処理内容	参照モードにて試験パターン No5 をオープンします
	さらし温度設定値を取得します
	さらし時間設定値を取得します
	参照モードにて試験パターン No5 をクローズします

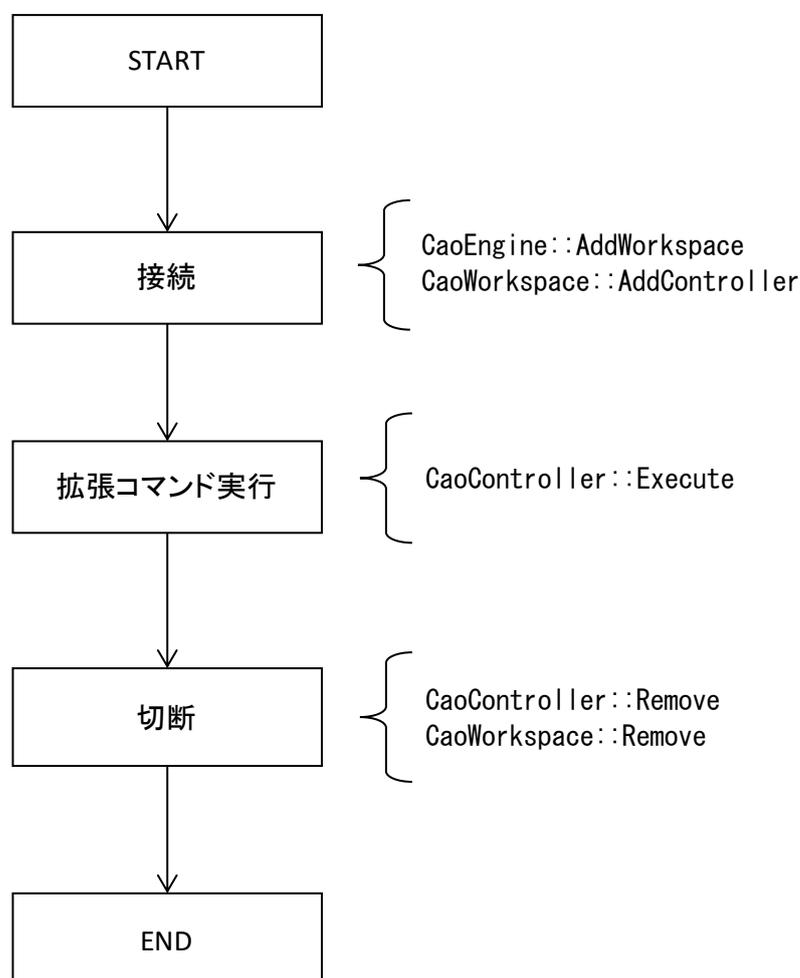


図 4-4 試験パターンモニタリングの流れ

以降の節から具体的なコードを示します。

#### 4.4.1. 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**      **MonitoringTestPattern\_TSA-3. vb**

```
' オブジェクト
Dim caoEng As CaoEngine
Dim caoWor As CaoWorkspace
Dim caoCon As CaoController

Private Sub Main
    ' 接続
    Call Connect

    Dim result As String
    ' 参照モードにて試験パターンをオープン
    result = caoCon.Execute("PRGMREAD", 5)

    ' さらし温度設定値を取得
    Dim tempVal() As Long
    tempVal = caoCon.Execute("LISTTEMP")

    ' さらし時間設定値を取得
    Dim timeVal() As String
    timeVal = caoCon.Execute("LISTTIME")

    ' 参照モードにて試験パターンをクローズ
    result = caoCon.Execute("PRGMREADEND")

    ' 切断
    Call Disconnect
End Sub

' 接続メソッド
Private Sub Connect()
    ' CaoEngine オブジェクトの生成
    Set caoEng = New CaoEngine
    ' CaoWorkspace オブジェクトの生成
    Set caoWor = caoEng.AddWorkspace("Workspace", "")
    ' CaoController オブジェクトの生成
    Set caoCon = caoWor.AddController("Controller", _
        "GaoProv. espec. ThermalShock", _
        "", _
        "conn=eth:192.168.0.2")
End Sub

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

```
' CaoEngine から CaoWorkspace を削除
Call caoEng.Workspaces.Remove(caoWor.Index)
' CaoWorkspace の消去
Set caoWor = Nothing

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

---

#### 4.4.1.1. Ethernet 対応デバイスと接続

Ethernet 対応デバイスとの接続は 4.3.1.1 Ethernet 対応デバイスと接続を参照して下さい。

#### 4.4.1.2. 試験パターンのモニタリング

CaoExtension オブジェクトの Execute メソッドを実行します。Execute メソッド実行時の引数ならば Execute メソッド実行後の戻り値を参照する場合は、Execute メソッドに合わせた型変数を用意する必要があります。以下にコード例を示します。

---

```
Dim result As String
' 参照モードにて試験パターンをオープン
result = caoCon.Execute("PRGMREAD", 5)

' さらに温度設定値を取得
Dim tempVal() As Long
tempVal = caoCon.Execute("LISTTEMP")

' さらに時間設定値を取得
Dim timeVal() As String
timeVal = caoCon.Execute("LISTTIME")

' 参照モードにて試験パターンをクローズ
result = caoCon.Execute("PRGMREADEND")
```

---

#### 4.4.1.3. Ethernet 対応デバイスと切断

Ethernet 対応デバイスとの切断は 4.3.1.3 Ethernet 対応デバイスと切断を参照して下さい。

## 5. espec ThermalShock 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	アドレスオプションがありません。
0x80110002	アドレスが既に登録されています。
0x80110003	オブジェクトを追加できません。

また、本プロバイダは、デバイスのエラーコードを「0x8010\*\*\*\*」でマスクして返します。デバイスから送られてきたエラーメッセージを元に以下のエラーコードとなっています。（表 5-2 デバイスのエラーコード表参照）

表 5-2 デバイスのエラーコード表

エラー番号	エラーメッセージ	説明
0x80100001	CMD ERR	コマンドデータとして認識できない
0x80100002	CONTROLLER NOT READY-1	試験時にのみ対応可能なコマンドを試験以外の状態で使用した
0x80100003	CONTROLLER NOT READY-2	パネル電源OFF状態で、プロテクト設定を行おうとした
0x80100004	CONTROLLER NOT READY-3	設定コマンド使用時に、装置がその要求を処理不可能な運転状態の場合
0x80100005	CONTROLLER NOT READY-4	湿度対応でない装置に湿度コマンドを使用した
0x80100006	PARA ERR	付加されたパラメーターに誤りがある
0x80100007	DATA OUT OF RANGE	データ範囲外
0x80100008	PROTECT ON	リモートプロテクト設定により、通信機能での設定処理が実施できない
0x80100009	PRGM READ ERR-1	参照モードでオープンせずに、試験パターンを参照しようとした場合
0x8010000A	PRGM READ ERR-2	参照モードで試験パターンをオープン中に、運転変更もしくは試験パターン編集を行おうとした場合
0x8010000B	PRGM READ ERR-3	参照モードで試験パターンオープン中に、再度参照モードで試験パターンオープンを行った場合
0x8010000C	PRGM READ ERR-20	STT機能を使用しない試験パターン参照時に、STT機能を使用する試験パターン用パラメータを使用した場合

エラー番号	エラーメッセージ	説明
0x8010000D	PRGM READ ERR-21	STT機能を使用する試験パターン参照時に、STT機能をしていない試験パターンしか使用できないパラメータを使用した場合
0x8010000E	PRGM WRITE ERR-1	編集モードでオープンせずに、試験パターンを編集しようとした場合
0x8010000F	PRGM WRITE ERR-2	編集モードで試験パターンをオープン中に、運転変更もしくは試験パターン参照を行おうとした場合
0x80100010	PRGM WRITE ERR-3	参照モードで試験パターンオープン中に、再度編集モードで試験パターンオープンを行った場合
0x80100011	PRGM WRITE ERR-4	ROMパターンを編集しようとした場合
0x80100012	PRGM WRITE ERR-5	オープンしている試験パターン以外のパターンをクローズしようとした場合
0x80100013	PRGM WRITE ERR-6	画面にて試験パターン編集/参照中に、通信コマンドにて編集しようとした場合
0x80100014	PRGM WRITE ERR-7	自動予熱・予冷モード設定時に、予熱・予冷設定値の編集を行おうとした場合
0x80100015	PRGM WRITE ERR-8	節約運転有効時もしくは安定時間制御有効時に、さらし時間短縮設定を有効にしようとした場合
0x80100016	PRGM WRITE ERR-9	さらし時間短縮有効に、安定時間制御設定を有効にしようとした場合
0x80100017	PRGM WRITE ERR-10	さらし時間短縮有効に、エコ運転制御設定を有効にしようとした場合
0x80100018	PRGM WRITE ERR-11	補助冷却オプション未搭載時に、補助冷却設定を行おうとした場合
0x80100019	PRGM WRITE ERR-14	低温さらし温度設定が-66℃以下で予冷設定をONした場合
0x8010001A	PRGM WRITE ERR-15	エコ運転設定が無効の場合
0x8010001B	PRGM WRITE ERR-20	STT機能を使用しない試験パターン編集時に、STT機能を使用する試験パターンパラメータを使用した場合
0x8010001C	PRGM WRITE ERR-21	STT機能を使用する試験パターン編集時に、STT機能をしていない試験パターンしか使用できないパラメータを使用した場合
0x8010001D	CMD_ERR	メインコマンドに誤りがある
0x8010001E	INVALID REQ	装置が対応できない機能を指定された場合
0x8010001F	CHB NOT READY	装置状態により、コマンドが処理できない場合
0x80100020		ThermalShock RS-485マニュアルの「1.3 通信機能で扱うデータ」の「表1.1 エラー

エラー番号	エラーメッセージ	説明
		メッセージ」及びThermalShock Ethernetマニュアル「1.3.3 エラーメッセージ」記載されていないエラーメッセージを受信した

## 6. 付録

### 付録A. ThermalShock プロバイダのコマンドとデバイスのコマンドの対応

本プロバイダのGaoControllerクラスのコマンドならびにGaoExtensionクラスのコマンドとデバイスのコマンドの対応は以下の通りです。なお、TSA シリーズはすべてのコマンドに対応しています。

表 6-1 プロバイダのコマンドとデバイスのコマンド対応

プロバイダのコマンド		デバイスのコマンド	TSE	TSB	TSD
GaoController クラス	GaoExtension クラス		シリーズ	シリーズ	シリーズ
GetROM		ROM?	○	○	○
GetROMDISP		ROM?, DISP	○	○	○
GetPOWERCUT		POWERCUT?	○	○	○
GetSTOPMODE		STOPMODE?	○	○	○
-	GetG-SOAKTEMP	G-SOAKTEMP?	○	○	×
GetDRYTEMP		DRYTEMP?	×	×	×
GetAMBTIME		AMBTIME?	×	×	×
GetKEYPROTECT		KEYPROTECT?	○	○	○
GetMODE		MODE?	○	○	○
GetASSIGN		ASSIGN?	○	○	○
GetTEMP		TEMP?	○	○	○
GetPRE		PRE?	○	○	○
GetTIME		TIME?	○	○	○
GetCYCLE		CYCLE?	○	○	○
GetMODEL		MODEL?	○	○	○
GetALARM		ALARM?	○	○	○
GetPRESET-HALT-ON	-	PRESET-HALT-ON?	×	×	○
GetPRESET-HALT	-	PRESET-HALT?	×	×	×
LISTTEMP		LISTTEMP?	○	○	○
LISTPREAI		LISTPREAI?	○	○	○
LISTPRE		LISTPRE?	○	○	○
LISTTIME		LISTTIME?	○	○	○
LISTCYCLE		LISTCYCLE?	○	○	○
LISTSTARTPOSITION		LISTSTARTPOSITION?	○	○	○
-	LISTPAUSE	LISTPAUSE?	○	○	○
LISTG-SOAK		LISTG-SOAK?	○	○	○

プロバイダのコマンド		デバイスのコマンド	TSE シリ ーズ	TSB シリ ーズ	TSD シリ ーズ
GaoController クラス	GaoExtension クラス				
LISTEND		LISTEND?	○	○	○
LISTNAME		LISTNAME?	○	○	○
LISTSENSOR		LISTSENSOR?	○	×	○
LISTDEFROST		LISTDEFROST?	○	×	○
LISTTEMP-L-DEFROST		LISTTEMP-L-DEFROST?	×	×	×
LISTQ-EXP		LISTQ-EXP?	○	×	○
LISTTEMPLIMIT		LISTTEMPLIMIT?	○	○	○
LISTAUXCOOLER		LISTAUXCOOLER?	○	×	○
LISTDRYMODESET		LISTDRYMODESET?	×	×	×
LISTDRYMODE		LISTDRYMODE?	×	×	×
LIST-PRE-ECONOMY		LIST-PRE-ECONOMY?	×	×	×
LISTRELAY1		LISTRELAY1?	○	○	○
LISTRELAY2		LISTRELAY2?	○	○	○
SetKEYPROTECT		KEYPROTECT	○	○	○
SetMODE		MODE	○	○	○
SetASSIGN		ASSIGN	○	○	○
OPESTANDBY		OPESTANDBY	○	○	○
OPESETUP		OPESETUP	○	○	○
OPESETUPEND		OPESETUPEND	○	○	○
OPETEST		OPETEST	○	○	○
OPEHALT		OPEHALT	○	○	○
OPERELASE		OPERELASE	○	○	○
OPECYCLERESET		OPECYCLERESET	○	○	○
OPEDEFROST		OPEDEFROST	○	×	○
OPEDRY		OPEDRY	×	×	×
OPEPRESET-SETUP		OPEPRESET-SETUP	○	○	○
OPEPRESET-TEST		OPEPRESET-TEST	○	○	○
OPEPRESET-ON		OPEPRESET-ON	○	○	○
OPEPRESET-CLR		OPEPRESET-CLR	○	○	○
OPEPRESET-HALT-ON	-	OPEPRESET-HALT-ON	×	×	×
OPEPRESET-HALT-CLR	-	OPEPRESET-HALT-CLR	×	×	×
OPEPRESET-HALT	-	OPEPRESET-HALT	×	×	×
PRGMREAD		PRGMREAD	○	○	○

プロバイダのコマンド		デバイスのコマンド	TSE シリ ーズ	TSB シリ ーズ	TSD シリ ーズ
GaoController クラス	GaoExtension クラス				
PRGMREADEND		PRGMREADEND	○	○	○
PRGMWRITE		PRGMWRITE	○	○	○
PRGMWRITEEND		PRGMWRITEEND	○	○	○
-	PRGMCREATE	PRGMCREATE	○	○	○
-	PRGMCREATEEND	PRGMCREATEEND	○	○	○
TEMP		TEMP	○	○	○
PREAI		PREAI	○	○	○
PRE		PRE	○	○	○
TIME		TIME	○	○	○
CYCLE		CYCLE	○	○	○
STARTPOSITION		STARTPOSITION	○	○	○
DEFROST		DEFROST	○	×	○
TEMP-L-DEFROST		TEMP-L-DEFROST	×	×	×
-	PAUSE	PAUSE	○	○	○
Q-EXP		Q-EXP	○	×	○
G-SOAK		G-SOAK	○	○	○
-	ECONOMY	ECONOMY	○	○	○
SENSOR		SENSOR	○	×	○
AUXCOOLER		AUXCOOLER	○	×	○
END		END	○	○	○
NAME		NAME	○	○	○
TEMPLIMIT		TEMPLIMIT	○	○	○
RELAY1		RELAY1	○	○	○
RELAY2		RELAY2	○	○	○
DRYMODESET		DRYMODESET	×	×	×
DRYMODE		DRYMODE	×	×	×
PRE-ECONOMY		PRE-ECONOMY	×	×	×

※○ - 対応      × - 非対応

### 付録B. ThermalShock プロバイダの変数とデバイスのコマンドの対応

本プロバイダのGaoController クラス変数ならびにGaoExtension クラス変数とデバイスのコマンドの対応は以下の通りです。

表 6-2 プロバイダの変数とデバイスのコマンド対応

プロバイダの変数		デバイスのコマンド		TSE	TSB	TSD
CaoController クラス	CaoExtension クラス	get_Value	put_Value	シリー ズ	シリー ズ	シリー ズ
@ROM		ROM?	—	○	○	○
@ROMDISP		ROM?, DISP	—	○	○	○
@POWERCUT		POWERCUT?	—	○	○	○
@STOPMODE		STOPMODE?	—	○	○	○
—	@G-SOAKTEMP	G-SOAKTEMP?	—	○	○	×
@DRYTEMP		DRYTEMP?	—	×	×	×
@AMBTIME		AMBTIME?	—	×	×	×
@KEYPROTECT		KEYPROTECT?	KEYPROTECT	○	○	○
@MODE		MODE?	MODE	○	○	○
@ASSIGN		ASSIGN?	ASSIGN	○	○	○
@TEMP		TEMP?	—	○	○	○
@PRE		PRE?	—	○	○	○
@TIME		TIME?	—	○	○	○
@CYCLE		CYCLE?	—	○	○	○
@MODEL		MODEL?	—	○	○	○
@ALARM		ALARM?	—	○	○	○
@PRESET-HALT-ON	—	PRESET-HALT-ON?	—	×	×	○
PRESET-HALT	—	PRESET-HALT?	—	×	×	×

※○ - 対応      × - 非対応

### 付録C. 対応機種一覧

ThermalShock RS-485 マニュアルならびに ThermalShock Ethernet マニュアルに記載されている対応機種一覧を記載します。なお本プロバイダでは、TSA シリーズ以外の機種では対応していないコマンドがあります。詳細は、付録の「表 6-1 プロバイダのコマンドとデバイスのコマンド対応」と「表 6-2 プロバイダの変数とデバイスのコマンド対応」を参照してください。

表 6-3 RS-485 対応機種一覧

機種	注意事項
TSA-41L	
TSA-71L	
TSA-71S	
TSA-71H	

TSA-101L	
TSA-101S	
TSA-201S	
TSA-301L	
TSE-11	
TSB-21	
TSB-51	
TSD-100	

表 6-4 Ethernet 対応機種一覧

機種	注意事項
TSA-**3 シリーズ	
TSD-101-W	