

AN プロバイダ デンソー製 HF 帯 RFID プロバイダ

Version 1.0.0

ユーザーズ ガイド

November 11, 2020

備考:

【改版履歴】

バージョン	日付	内容
1.0.0	2019-08-01	初版
	2020-11-11	誤記修正

【動作確認機種】

機種	バージョン	注意事項
AN10R-01	9000801	

【動作確認タグ】

タグ	規格
角型タグ	ISO/IEC 18000-3 mode 1
円筒形タグ	ISO/IEC 18000-3 mode 1

目次

1. はじめに	5
1.1. 参考となる情報源	5
2. アプリケーション開発のための環境セットアップ	6
2.1. リーダライタとクライアント PC との接続	6
2.2. PC 開発環境のセットアップ	6
2.2.1. AN プロバイダの自動インストール	6
2.2.2. AN プロバイダの手動インストール	6
3. AN プロバイダによるプログラミング	7
3.1. RFID タグを取得するするサンプルプログラミング	7
3.1.1. サンプルプログラム	8
3.1.1.1. 接続	9
3.1.1.2. RFID タグの情報取得	10
3.1.1.3. 切断	10
4. コマンドリファレンス	11
4.1. メソッド/プロパティ一覧	11
4.2. メソッド・プロパティ	11
4.2.1. CaoWorkspace クラス	11
4.2.1.1. AddController メソッド	11
4.2.1.1.1. CONN オプション	12
4.2.2. CaoController クラス	13
4.2.2.1. VariableNames プロパティ	13
4.2.2.2. Variables プロパティ	13
4.2.2.3. AddVariable メソッド	14
4.2.2.4. Execute メソッド	14
4.2.2.4.1. RAW コマンド	15
4.2.2.4.2. ReadTag コマンド	15
4.2.2.4.3. WriteTag コマンド	16
4.2.2.4.4. ReadID コマンド	16
4.2.2.4.5. ConnectState コマンド	17
4.2.2.5. OnMessage イベント	17
4.3. 変数一覧	17

4.3.1. コントローラクラス	17
4.3.1.1. @MAKER_NAME	18
4.3.1.2. @VERSION	18
4.3.1.3. @LAST_ERROR	19
4.3.1.4. DEVICE_VERSION	19
4.3.1.5. TAG_DATA	20
4.3.1.6. TAG_ID	21
4.3.1.7. CONNECT_STATE	21
4.4. エラーコード	22

1. はじめに

このドキュメントは、デンソー製 HF 帯 RFID リーダライタ用のプロバイダである AN プロバイダのユーザーズガイドです。

AN プロバイダは、リーダライタからデータを出力するプロバイダです。

本書は、この AN プロバイダの機能と実装されているメソッドについて説明します。

図 1-1 構成図は本プロバイダとデバイスの全体構成図になります。AN プロバイダはリーダライタと RS232C によるシリアル通信を行います。

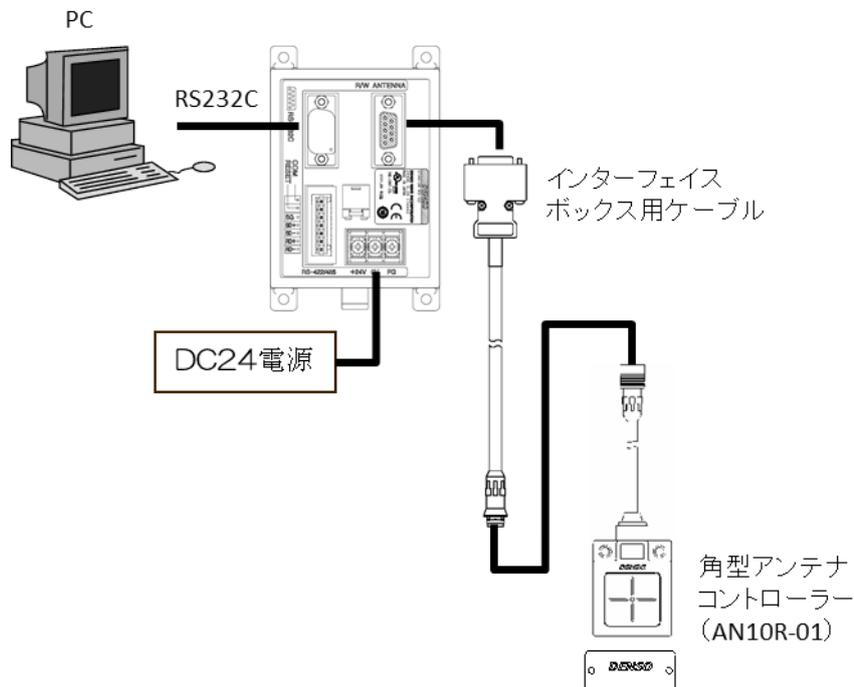


図 1-1 構成図

1.1. 参考となる情報源

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

「ORiN 2 プログラミングガイド」は ORiN2 SDK インストールフォルダの以下のファイルに該当します。

- ORiN2¥CA0¥Doc¥ORiN2_ProgrammersGuide_<lang>.pdf

※<lang>の部分は環境毎の言語文字列に置き換えてお読みください。

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

2. アプリケーション開発のための環境セットアップ

2.1. リーダライタとクライアント PC との接続

リーダーライタとクライアント PC との接続については、お使いの HF 帯リーダーライタ取り扱い説明書を参照してください。

2.2. PC 開発環境のセットアップ

2.2.1. AN プロバイダの自動インストール

ORiN2 SDK がインストールされている環境であれば、リーダーライタに接続するための動作環境（ライタイム）の準備は完了です。

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

2.2.2. AN プロバイダの手動インストール

AN プロバイダを使用するためには手作業で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。

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

表 2-1 AN プロバイダの手動インストール

ファイル名	CaoProvDENSOAN.dll
ProgID	CaoProv.DENSO.AN
レジストリ登録 ¹	regsvr32 CaoProvDENSOAN.dll
レジストリ登録の抹消	regsvr32 /u CaoProvDENSOAN.dll

¹ ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません。

3. AN プロバイダによるプログラミング

AN プロバイダでは、以下の手順でクライアント PC とリーダライタを接続することができます。

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

リーダライタに接続した後は、CaoController の Execute メソッドを使用することで、リーダライタ自身の情報および、リーダライタが読み込んだ RFID タグの情報にアクセスすることができます。

3.1. RFID タグを取得するサンプルプログラミング

ここでは例としてリーダライタが読み込んだ RFID タグを取得するサンプルプログラムを示します。

表 3-1 にサンプルプログラムの要件、図 3-1 にサンプルプログラムの流れ、3.1.1 に実際のプログラムをそれぞれ記述しています。

表 3-1 サンプルプログラムの要件

要件	説明
接続先	RS232C で接続する
処理内容	TAG_DATA 変数を使って RFID タグのメモリデータを取得する。

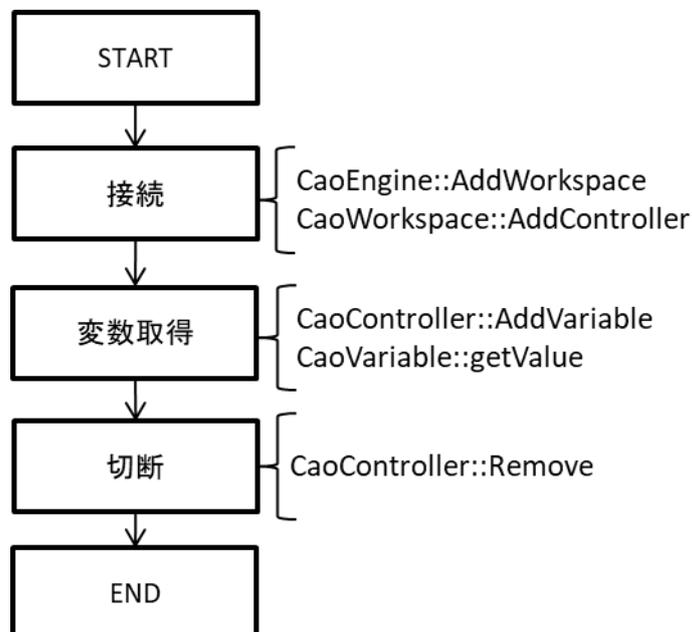


図 3-1 サンプルプログラムの流れ

3.1.1. サンプルプログラム

以下にサンプルプログラムの全体像を示します。

Sample	TAG_ID Sample.vb
--------	------------------

```
Imports CAOLib

Public Class Form1
    Dim engine As CaoEngine
    Dim workspace As CaoWorkspace
    Dim controller As CaoController
    Dim variable As CaoVariable

    Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        ' Create CaoEngine
        engine = New CaoEngine
        ' Create CaoWorkspace
        workspace = engine.Workspaces.Item(0)
        ' Create CaoController
        controller = workspace.AddController("Sample", _
                                           "CaoProv.DENSO.AN", _
                                           ""', _
                                           "Conn=com:1")
        variable = controller.AddVariable("TAG_DATA", _
                                         "machineNo = 1, Addr = 0, Length = 10")

        TextBox1.Text = variable.Value

        ' Delete CaoVariable from CaoController
        Call controller.Variables.Remove(variable.Index)
        variable = Nothing

        ' Delete CaoController from CaoWorkspace
        Call workspace.Controllers.Remove(controller.Index)
        controller = Nothing

        ' Delete CaoWorkspace from CaoEngine
        Call engine.Workspaces.Remove(workspace.Index)
        workspace = Nothing

        ' Delete CaoEngine
        If engine IsNot Nothing Then
            engine = Nothing
        End If
    End Sub
End Class
```

3.1.1.1. 接続

リーダライタと接続するためには、以下の手順を取ります。

- (1) オブジェクトを保持するための変数を用意します。コントローラ接続に必要なオブジェクトは、CaoEngineオブジェクトとCaoWorkspaceオブジェクトとCaoControllerオブジェクトです。

CaoWorkspaceオブジェクトは、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VB.Netでのコード例を示します。

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

- (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
engine = New CaoEngine
```

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

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

- (4) CaoControllerオブジェクトを生成します。CaoControllerオブジェクトを生成するには、使用するプロバイダ名と使用するためのパラメータを設定します。ANプロバイダでは、接続情報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
controller = workspace.AddController("SampleController", _
    "CaoProv. DENSO. AN", _
    "", _
    "conn=com:1")
```

3.1.1.2. RFID タグの情報取得

リーダライタが読み込んだ RFID タグの情報を取得するために、CaoController のユーザー変数である、TAG_ID 生成し値を取得します。詳細は TAG_ID 変数を参照してください。

3.1.1.3. 切断

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

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

4. コマンドリファレンス

4.1. メソッド/プロパティ一覧

表 4-1 メソッド/プロパティ一覧

カテゴリ	メソッド/プロパティ ²		機能	参照
CaoWorkspace				
	Addcontroller	M	コントローラに接続	P. 11
CaoController				
	VariableNames	P	接続可能な変数名リストの取得	P. 13
	Variables	P	コントローラが保持する変数コレクションの取得	P. 13
	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. 14
	OnMessage	E	メッセージ受信イベント	P. 17
CaoVariable				
	Value	P	値の取得/設定	P. 17

4.2. メソッド・プロパティ

4.2.1. CaoWorkspace クラス

4.2.1.1. AddController メソッド

CaoWorkspace に、コントローラオブジェクトを追加します。以下に、AddController メソッドの仕様を示します。

書式

CaoController AddController

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

オプション

² 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。

以下にオプション文字列に指定するオプションを示します。オプション文字列は下記に示す各オプションをカンマ(,)でつなげた文字列となります。

オプション	必須	説明	値範囲	デフォルト値
CONN=<通信パラメータ>	○	接続先情報を指定します。	COM	--
Timeout =<通信タイムアウト>	--	通信タイムアウトを ms 単位で指定します。	0 以上	500
@EventDisable	--	OnMessage イベント (4.2.2.5) の発生を設定します。	True/False	False

使用例

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

engine = New CaoEngine
workspace = engine.Workspaces.Item(0)
controller = workspace.AddController("SampleController",
                                     "CaoProv. DENSO. AN", _
                                     "", _
                                     "conn=com:1")
```

4.2.1.1.1. CONN オプション

以下に Conn オプションの接続パラメータ文字列を示します。ここで角括弧("[]")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。

RS232C

"Conn=COM:<COM Port>[<:BaudRate>]"

<COM Port> : COM ポート番号. '1' -COM1, '2' - COM2, ...
 <BaudRate> : 通信速度. 9600, 19200, 38400, 115200

4.2.2. CaoController クラス

4.2.2.1. VariableNames プロパティ

接続可能な変数名リストを取得します。本プロパティで取得した変数名は、後述する AddVariable メソッドの第一引数に使用することができます。

使用例

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

engine = New CaoEngine
workspace = engine.Workspaces.Item(0)
controller = workspace.AddController("SampleController", _
                                     "CaoProv. DENSO. AN", _
                                     "" _
                                     , _
                                     "conn=com:1")

' ファイル名リスト取得
Dim variables as Variant
variables = controller.VariableNames
```

4.2.2.2. Variables プロパティ

コントローラが保持する、変数コレクションを取得します。

使用例

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

engine = New CaoEngine
workspace = engine.Workspaces.Item(0)
controller = workspace.AddController("SampleController", _
                                     "CaoProv. DENSO. AN", _
                                     "" _
                                     , _
                                     "conn=com:1")

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

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

4.2.2.3. AddVariable メソッド

CaoController に変数オブジェクトを追加します。変数名には 4.3 変数一覧に示すもののみ使用できます。

以下に、AddVariable の仕様を示します。

書式

CaoVariable AddVariable

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

4.2.2.4. Execute メソッド

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

書式

Variant Execute

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

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

表 4-2 Execute コマンド一覧

コマンド	説明	参照
RAW	電文を送信しその応答を返します。	P. 15
ReadTag	タグからデータを読み取ります。	P. 15
WriteTag	タグへデータを書き込みます。	P. 16
ReadID	最大 4 枚までの UID を読み取ります。	P. 16
ConnectState	接続状態を取得します。	P. 17

4.2.2.4.1. RAW コマンド

入力された電文をリーダライタに送信します。

項目	型説明	
引数	VT_BSTR	電文形式の文字列。 16 進数文字列かつ 2 の倍数の長さで入力してください。
戻り値	VT_BSTR	応答電文。

電文形式は HF 帯リーダライタマニュアルを参照ください。

なお、本書では電文は以下の形式を想定しています。

表 4-3 電文形式

項目	STF	ADDR	CTL	LEN	DATA					ETF	FCS
					CMD	Para1	Para2	Para3	Data		
長さ	2	1	1	2	1	1	1	1	n	2	2

使用例

' コマンド電文を送信

```
Dim command As string
command = controller.Execute("RAW", "AA55010000047100000055AA0274")
```

```
End If
```

4.2.2.4.2. ReadTag コマンド

タグからデータを読み取ります。

項目	型説明	
引数	VT_UI2 VT_ARRAY	
	1	機器 No. 範囲 : 1 - 31
	2	読み取開始アドレス. 範囲 : 0 - 65535
	3	読み取りデータ長. 範囲 : 1 - 65535
戻り値	VT_BSTR	読み取りデータ.

使用例

' コマンド電文を送信

```
Dim param As Array
Dim data As string
param = {1, 0, 10}
```

```
string = controller.Execute("ReadTag", param)
End If
```

4.2.2.4.3. WriteTag コマンド

タグヘータを書き込みます。

項目	型説明	
引数	VT_VARIANT VT_ARRAY	
	1	VT_UI2 機器 No. 範囲 : 1 - 31
	2	VT_UI2 書き込み開始アドレス. 範囲 : 0 - 65535
	3	VT_BSTR 書き込みデータ 16 進数表記かつ 2 の倍数の長さでの入力にのみ対応しています.
戻り値	VT_BOOL	書き込み結果

使用例

```
' タグヘータ書き込み
Dim result As bool
Dim param As Array
param = {1, 0, "AABBCCDDEE"}

result = controller.Execute("WriteTag", param)

End If
```

4.2.2.4.4. ReadID コマンド

最大 4 枚までのタグから UID を読み取ります。

項目	型説明	
引数	VT_UI2	機器 No. 範囲 : 1 - 31
戻り値	VT_ARRAY VT_VARIANT	
	1	VT_UI2 読み取った UID データの個数
	2	VT_BSTR 読み取った UID データ 1 個目 ※データが無い場合は空文字
	3	VT_BSTR 読み取った UID データ 2 個目 ※データが無い場合は空文字
	4	VT_BSTR 読み取った UID データ 3 個目 ※データが無い場合は空文字
	5	VT_BSTR 読み取った UID データ 4 個目 ※データが無い場合は空文字

使用例

```
' UID 読み取り
```

```

Dim result As Array
result = controller.Execute("ReadID", 1)
TextBox1.Text = result(0).ToString()
                + result(1).ToString()
                + result(2).ToString()
                + result(3).ToString()
                + result(4).ToString()

End If

```

4.2.2.4.5. ConnectState コマンド

タグヘータを書き込みます。

項目	型説明	
引数	VT_UI2	機器 No. 範囲 : 1 - 31
戻り値	VT_BSTR	接続可能な場合:"OK" 接続可能な場合:"NG"

使用例

```

' 接続状態を取得
Dim state As string
state = controller.Execute("ConnectState", 1)

End If

```

4.2.2.5. OnMessage イベント

本プロバイダではコマンド実行時にデバイスからエラーが返ったタイミングで, CaoController クラスの OnMessage イベントとしてクライアントにエラー詳細を受け渡します。エラー詳細の内容は @LAST_ERROR 変数で取得できるものと同じです。

表 4-4 OnMessage 内容

Number プロパティ	Value プロパティ	データ型
1	エラー詳細	VT_BSTR

4.3. 変数一覧

4.3.1. コントローラクラス

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。

表 4-5 システム変数一覧

変数名	説明	属性	
		get	put
@MAKER_NAME	メーカー名を取得します。	○	-
@VERSION	プロバイダバージョンを取得します。	○	-
@LAST_ERROR	最後に起こったエラーを返します。	○	-

4.3.1.1. @MAKER_NAME

メーカー名の取得をします。

データ型

項目	型説明	
戻り値	VT_BSTR	メーカー名を取得します。

使用例

```
' 変数追加
Dim variable As CaoVariable
variable = controller.AddVariable("@MAKER_NAME")
' 値取得
Dim strVal As String
strVal = variable.value
```

4.3.1.2. @VERSION

プロバイダバージョンの取得をします。

データ型

項目	型説明	
戻り値	VT_BSTR	プロバイダバージョンを取得します。

使用例

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

4.3.1.3. @LAST_ERROR

最後に起こったエラーを取得します。

データ型

項目	型説明
戻り値	VT_BSTR 最後に起こったエラーの 16 進数文字列。

エラーの内容はリーダーライタの取扱説明書を参照ください。

使用例

```

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

```

表 4-6 ユーザー変数一覧

変数名	説明	属性	
		get	put
DEVICE_VERSION	デバイスのファームウェアバージョンを取得します。	○	-
TAG_DATA	RFID タグのメモリを読み書きします。	○	○
TAG_ID	RFID の UII を読み取ります。	○	-
CONNECT_STATE	接続状態を取得します。	○	-

4.3.1.4. DEVICE_VERSION

デバイスのバージョンの取得をします。

オプション	必須	説明	設定できる値範囲	デフォルト
MachineNo	○	機器 No	1 - 31	-

データ型

項目	型説明
----	-----

戻り値	VT_BSTR	デバイスのバージョン.
-----	---------	-------------

使用例

```
' 変数追加
```

```
Dim variable As GaoVariable
variable = controller.AddVariable("DEVICE_VERSION", "MachineNo = 1")
```

```
' 値取得
```

```
Dim strVal As String
strVal = variable.value
```

4.3.1.5. TAG_DATA

RFID タグのデータの読み書きをします.

オプション	必須	説明	設定できる値範囲	デフォルト
MachineNo	○	機器 No	1 - 31	-
Addr	○	読み取り/書き込み開始位置	0 - 65535	-
Length	○	読み取りデータ長 ※書き込み時は未使用	1 - 65535	-

データ型

項目	型説明
戻り値	VT_BSTR 読み取った/書き込みデータ. 書き込みの場合は 16 進数表記かつ 2 の倍数の長さでの入力にのみ対応しています. Length オプションで指定された長さにかかわらず入力できます.

- 読み取り時

使用例

```
' 変数追加
```

```
Dim variable As GaoVariable
Set variable = controller.AddVariable("TAG_DATA", "MachineNo=1, Addr=0, Length=100")
```

```
' 値取得
```

```
Dim strVal As String
strVal = variable.value
```

【補足】

読み取りは 400 バイトごとに約 600ms かかります.

- 書き込み時

使用例

変数追加

```
Dim variable As CaoVariable
variable = controller.AddVariable("TAG_DATA", "MachineNo=1, Addr=0, Length=100")
```

値取得

```
Dim strVal As String
variable.value = "00112233445566778899AABBCCDDEEFF"
```

4.3.1.6. TAG_ID

RFID タグの UID の読み書きをします。

オプション	必須	説明	設定できる値範囲	デフォルト
MachineNo	○	機器 No	1 - 31	-

データ型

項目	型説明		
戻り値	VT_ARRAY VT_VARIANT		
	1	VT_UI2	読み取った UID データの個数
	2	VT_BSTR	読み取った UID データ 1 個目 ※データが無い場合は空文字
	3	VT_BSTR	読み取った UID データ 2 個目 ※データが無い場合は空文字
	4	VT_BSTR	読み取った UID データ 3 個目 ※データが無い場合は空文字
	5	VT_BSTR	読み取った UID データ 4 個目 ※データが無い場合は空文字

使用例**変数追加**

```
Dim variable As CaoVariable
variable = controller.AddVariable("TAG_ID", "MachineNo=1")
```

値取得

```
Dim arrayUID As Array
arrayUID = variable.value
```

4.3.1.7. CONNECT_STATE

機器と接続可能か返します。

オプション	必須	説明	設定できる値範囲	デフォルト
MachineNo	○	機器 No	1 - 31	-

データ型

項目	型説明
----	-----

戻り値	VT_BSTR	接続可能な場合: "OK" 接続可能な場合: "NG"
-----	---------	--------------------------------

使用例

```
' 変数追加
```

```
Dim variable As CaoVariable  
variable = controller.AddVariable("CONNECT_STATE", "MachineNo=1")
```

```
' 値取得
```

```
Dim state As string  
state = variable.value
```

4.4. エラーコード

AN プロバイダで定義されている独自のエラーコードを表 4-7 に示します。

表 4-7 エラーコード

エラー名	エラー番号	説明
機器No異常	0x80100001	MachineNo オプションの範囲外が指定されています。 1 - 31 で設定してください。
データアドレス設定異常	0x80100002	Addr オプションの範囲外が指定されています。 0 - 65535 で設定してください。
データ長設定異常	0x80100003	Length オプションの範囲外が指定されています。 1 - 65535 で設定してください。
送信先機器No異常	0x80100004	受信した機器Noとオプション指定した機器Noが異なります。 機器かオプションを確認してください。
書き込みデータ長さ異常	0x80100005	書き込みデータの長さが 2byte 単位になっていません。 2byte 単位に設定してください。
書き込みデータ異常	0x80100006	書き込みデータに 16 進数以外が含まれています。 00~FF までの 16 進数文字列を指定してください。
パケット異常	0x80100007	FCS が不一致です。 機器メーカーにお問い合わせください。

デバイスからのエラー	0x8010FF01 0x8010FF02 0x8010FF03 0x8010FF65 0x8010FF71	コマンド実行時にデバイスからエラーが返りました。 下 2 桁はプロバイダから送信したコマンドを示します。 プロバイダで使用しているコマンドとデバイスコマンドとの対応は表 4-8 コマンド対応表 を参照してください。
------------	--	---

表 4-8 コマンド対応表

プロバイダ側の処理			デバイス側コマンド
変数	TAG_DATA	Get	0x01
		Put	0x02
	TAG_ID	Get	0x03
	CONNECT_STATE	Get	0x65
	DEVICE_VERSION	Get	0x71
Execute メソッド	ReadTag		0x01
	WriteTag		0x02
	ConnectState		0x65

その他は ORiN2 共通エラーについては、「[ORiN2 プログラミングガイド](#)」のエラーコードの章を参照してください。