デンソー
RFID テーブルスキャナ UR20 プロバイダ

Version 1.1.0

ユーザーズ ガイド

July 11, 2024

備考：

【改版履歴】

バージョン	日付	内容
1.0.0	2018-11-27	初版.
1.1.0	2018-12-25	UR20,UR21,UR22 対応 COM 通信対応 Kill コマンド追加
	2020-2-19	StartContinuousRFDetection コマンドの型説明を修正
	2022-1-7	誤字修正
	2024-7-11	キャリア出力強度のコマンドについて日本仕様の UR シリーズと海外仕様の UR シリーズで使用可能なコマンドが違う旨を追記

【動作確認機種】

機種	バージョン	注意事項
UR20	OS: 01.01.02.00 Main AP: 00.01.03.00 RF AP: 00.01.07.00 RF Firm: 0F.0F.06.F0 OEM: 01.02.06.00	端末へのコマンド操作が一定期間(60 秒)無い場合，端末は回線断と判断してソケットを自動的に閉じます. そのため，ソケットが閉じられた場合，ソケットを再接続してください. あるいは，定期的に端末疎通確認コマンド(P.25)を送信して回線を維持してください.
UR21	OS: 01.01.02.00 Main AP: 00.01.03.00 RF AP: 00.01.07.00 RF Firm: 0F.0F.06.F0 OEM: 01.02.06.00	
UR22	OS: 01.02.01.00 Main AP: 00.02.01.00 RF AP: 00.02.03.00 RF Firm: 2D.01.00.00 OEM: 01.06.06.00	

目次

1. はじめに

本書は，RFID テーブルスキャナの UR20 シリーズから RFID タグデータの取得をするプロバイダのユーザーズガイドです．図 1-1 が本プロバイダとデバイスの全体構成図になります．以降本プロバイダを UR20 プロバイダ，UR20 シリーズの RFID テーブルスキャナをスキャナと呼称します．UR20 プロバイダはスキャナと TCP/IP もしくは，USB シリアルによりデータ通信を行います．

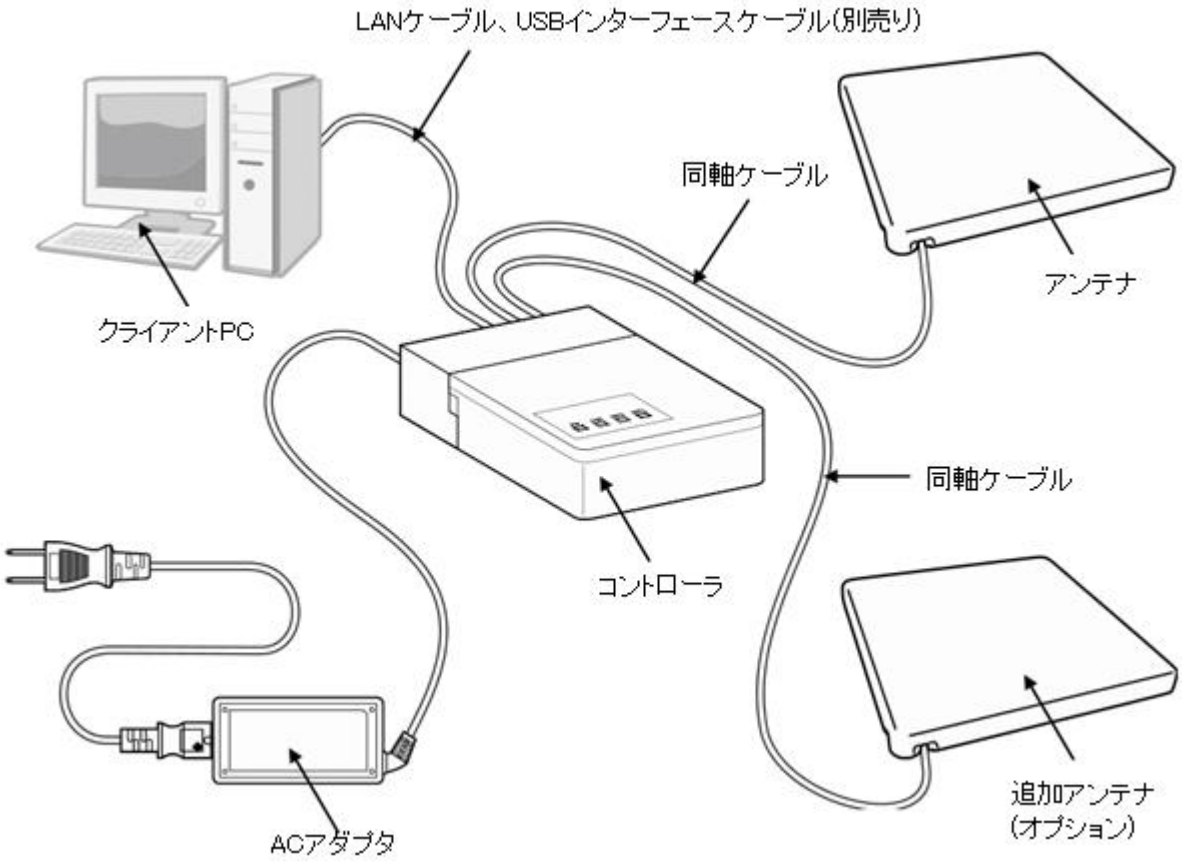

図 1-1 構成図

1.1. 本書が想定している環境とバージョン

クライアント PC が Windows 上で動作し，対象とする CNC がイーサネット接続可能な会社名製デバイス種類である環境を想定しています．PC の開発環境は，Component Object Model(COM，コンポーネント・オブジェクト・モデル)をサポートするプログラミング環境であれば開発が可能です．

1.2. 参考となる情報源

本書のプログラミング事例は，すべて Visual Basic 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」を参照していただきます．

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

- ORiN2¥CAO¥Doc¥ORiN2_ProgrammersGuide_<*lang*>.pdf

※<lang>の部分は環境毎の言語文字列に置き換えてお読みください．

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

2. アプリケーション開発のための環境セットアップ

2.1. スキャナとクライアント PC との接続

スキャナとクライアント PC との接続については，お使いの UR20 シリーズ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. UR20 プロバイダの自動インストール

ORiN2 SDK Ver がインストールされている環境であれば，スキャナに接続するための動作環境（ランタイム）の準備は完了です.

開発環境のセットアップは別途，Microsoft Visual Studio 6.0, 2003/2005/2008/2010, LabVIEW など Component Object Model（COM，コンポーネント・オブジェクト・モデル）をサポートする，プログラミング環境をご準備してください.

2.2.2. UR20 プロバイダの手動インストール

UR20 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 UR20 プロバイダの手動インストール

ファイル名	CaoProvDENSOUR20.dll
ProgID	CaoProv.DENSO.UR20
レジストリ登録	regsvr32 CaoProvDENSOUR20.dll
レジストリ登録の抹消	regsvr32 /u CaoProvDENSOUR20.dll

3. UR20 プロバイダによるプログラミング

UR20 プロバイダでは，以下の手順でクライアント PC とスキャナを接続することができます.

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

スキャナに接続した後は，CaoController の Execute メソッドを使用することで，スキャナ自身の情報および，スキャナが読み込んだ RFID タグの情報にアクセスすることができます.

3.1. RFID タグを取得するするサンプルプログラミング

ここでは例としてスキャナが読み込んだ 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を,図 3-1 にサンプルプログラムの流れをそれぞれ記述しています.

表 3-1 サンプルプログラムの要件

要件	説明
接続先	TCP/IP で接続する
	接続先 IP アドレスは 192.168.0.2
	接続先ポート番号は 20000
	AP 名は UHFTAGRW
処理内容	ReadUII コマンドにて RFID タグ情報を取得する.

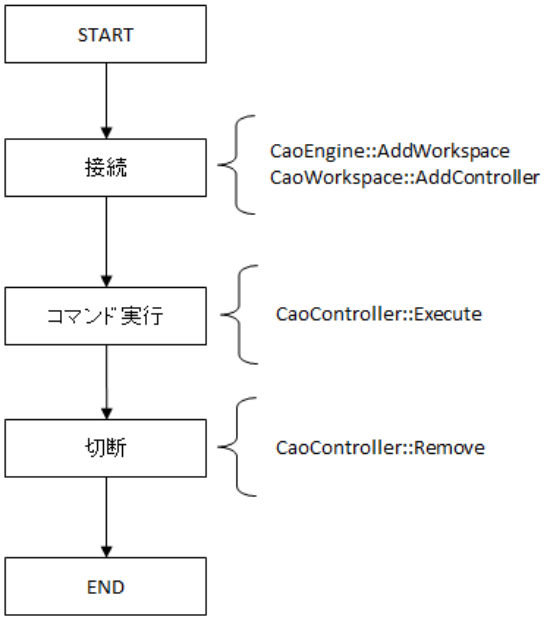

図 3-1 サンプルプログラムの流れ

以降の節から具体的なコードを示します.

3.1.1. サンプルプログラム

以下にサンプルプログラムの全体像を示します.

Sample	ReadUIISample.vb

```
Sub Main
    ' オブジェクト
    Dim engine As CaoEngine
    Dim workspace As CaoWorkspace
    Dim controller As CaoController

    ' CaoEngine オブジェクトの生成
    Set engine = New CaoEngine
    ' CaoWorkspace オブジェクトの生成
    Set workspace = engine.AddWorkspace("NewWrks", "")
    ' CaoController オブジェクトの生成
    Set controller = workspace.AddController("SampleController", _
                                             "CaoProv.DENSO.UR20", _
                                             "", _
                                             "conn=eth:192.168.0.2")

    ' RFID データ取得
    Dim uiis As Variant
    uiis = controller.Execute("ReadUII", False)
    If Not IsEmpty(uiis) Then
        ' 検出された個数を取得
        Dim detectedElem As Long
```

```
        detectedElem = UBound(uiis)

        ' 検出されたRFIDタグ数分繰り返す
        Dim i As Long
        For i = 0 To detectedElem
            ' PCデータ取得
            Dim pcData As Long
            pcData = uiis(i)(0)

            ' UIIデータを取得
            Dim uiiData() As Byte
            uiiData = uiis(i)(1)
            ' UIIデータサイズを取得
            Dim uiiDataSize As Long
            uiiDataSize = UBound(uiiData)
            ' UIIデータサイズ分データ抽出
            Dim j As Long
            For j = 0 To uiiDataSize
                Dim value As Byte
                value = uiiData(0)
            Next j
        Next i
    End If

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

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

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

3.1.1.1. 接続

スキャナ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．CaoWorkpace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 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
Set engine = New CaoEngine
```

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

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

(4) CaoControllerオブジェクトを生成します．CaoControllerオブジェクトを生成するには，使用するプロバイダ名と使用するためのパラメータを設定します．UR20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
Set controller = workspace.AddController("SampleController", _
                                          "CaoProv.DENSO.UR20", _
                                          "", _
                                          "conn=eth:192.168.0.2")
```

3.1.1.2. RFID タグの情報取得

スキャナが読み込んだ RFID タグの情報を取得するために，CaoController の拡張コマンドである，ReadUII コマンドを実行します．詳細は ReadUII コマンド(P.25)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
Set controller = Nothing
' CaoEngine から CaoWorkspace を削除
Call engine.Workspaces.Remove(workspace.Index)
' CaoWorkspace の消去
Set workspace = Nothing
' CaoEngine の消去
Set engine = Nothing
```

4. コマンドリファレンス

4.1. メソッド/プロパティ一覧

表 4-1 メソッド/プロパティ一覧

カテゴリ	メソッド/プロパティ[1]		機能	参照
CaoWorkspace				
	Addcontroller	M	コントローラに接続	P.13
CaoController				
	VariableNames	P	接続可能な変数名リストの取得	P.15
	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.15
	AddVariable	M	変数オブジェクトの追加	P.16
	Execute	M	拡張コマンドの実行	P.16
CaoVariable				
	Value	P	値の取得/設定	P.35

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.UR20",           // プロバイダ名 (固定)
        "<マシン名>",                   // プロバイダ実行マシン名(未使用)
        "<オプション>"                  // オプション文字列(省略可能)
)
```

オプション

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

[1] M:メソッド，P:プロパティ，E:イベントをそれぞれ示します．

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.	ETH or COM	--
APName=<AP 名称>	--	接続先の AP 名称を指定します. 16 文字以内で指定してください. 16 文字以上の AP 名称を指定した場合, 17 文字以降の文字は無視されます.	--	UHFTAGRW
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	2000

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

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("SampleController", _
                                          "CaoProv.DENSO.UR20", _
                                          "", _
                                          "conn=eth:192.168.0.2")
```

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.

TCP/IP で接続する場合

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

<接続先 IP> : 接続先 IP アドレスを***.***.***.***の形式で指定します.
この項目は必ず指定してください.

<接続先ポート> : 接続先ポート番号を指定します. 50000

RS232C

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

<COM Port>	:	COM ポート番号. '1' -COM1, '2' - COM2, ...
<BaudRate>	:	通信速度. 4800, 9600, 19200, 38400, 57600, 115200
<Parity>	:	パリティ. 'N'-NONE, 'E'-EVEN, 'O'-ODD
<DataBits>	:	データビット数. '7'-7bit, '8'-8bit
<StopBits>	:	ストップビット数. '1'-1bit, '2'-2bit
<Flow>	:	フロー制御. '0'-None, '1'-Xon/Xoff, '2'-ハードウェア制御

OR をとって指定できます.

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

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("SampleController", _
                                          "CaoProv.DENSO.UR20", _
                                          "", _
                                          "conn=eth:192.168.0.2")

' 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

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("SampleController", _
                                          "CaoProv.DENSO.UR20", _
                                          "", _
                                          "conn=eth:192.168.0.2")
```

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

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

4.2.2.3. AddVariable メソッド

CaoController に変数オブジェクトを追加します. 変数名には 0 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.

コマンド	説明	参照
SetCarrierOutputIntensityLevel	キャリア出力レベルを設定します. 設定した値は切断/端末電源 OFF で初期化されます	P.18
GetCarrierOutputIntensityLevel	キャリア出力強度レベルを取得します.	P.18
SetCarrierOutputIntensitydBm	キャリア出力強度を dBm×10 の単位で設定します. 設定した値は切断/端末電源 OFF で初期化されます.	P.18
GetCarrierOutputIntensitydBm	キャリア出力強度を dBm×10 の単位で取得しま	P.19

コマンド	説明	参照
	す.	
SetFrequency	キャリア出力を許可する周波数をビットフラグで設定します. 設定した値は切断/端末電源 OFF で初期化されます.	P.19
GetFrequency	キャリア出力を許可する周波数をビットフラグで取得します.	P.19
SetAntennaPort	RFID タグ通信を行う際のキャリアを出力するアンテナポートを設定します. 設定した値は切断/端末電源 OFF で初期化されます.	P.20
GetAntennaPort	RFID タグ通信を行う際のキャリアを出力するアンテナポートを取得します.	P.20
SetRereadPrevention	UII 読み取り処理, タグ通信処理での二度読み防止機能の有効/無効を設定します. 設定した値は切断/端末電源 OFF で初期化されます.	P.21
GetRereadPrevention	UII 読み取り処理, タグ通信処理での RFID タグ検出完了判定の有効/無効を取得します.	P.21
SetQValue	UII 取得時にタグからの応答が衝突する確率を調整するために使用する Q 値を設定します. 設定した値は切断/端末電源 OFF で初期化されます.	P.21
GetQValue	UII 取得時にタグからの応答が衝突する確率を調整するために使用する Q 値を取得します.	P.22
SetSession	タグの UII 取得時に使用するセッションを設定します. 設定した値は切断/端末電源 OFF で初期化されます.	P.22
GetSession	タグの UII 取得時に使用するセッションを取得します.	P.23
ResetTerminal	処理を中断します.	P.23
Reboot	RFID テーブルスキャナを再起動します. 再起動後は, 接続処理を行ってください.	P.23
GetSerialNumber	RFID テーブルスキャナの製造番号情報を取得します.	P.24
GetVersion	RFID テーブルスキャナのバージョン情報を取得します.	P.24
KeepAlive	RFID テーブルスキャナに端末疎通コマンドを送信します.	P.25
ReadUII	RFID タグの検出を行い, UII データを読み取ります.	P.25
StartContinuousRFDetection	連続 UII 読み取りを開始します.	P.26
ReadContinuousUII	StartContinuousRFDetection を実行後から検出された RFID タグの UII データを読み取ります.	P.27
StopContinuousRFDetection	連続 UII 読み取りを停止します.	P.28
ReadMemory	RFID タグの指定メモリのデータを読み出します.	P.29
WriteMemory	RFID タグの指定メモリのデータを書き込みます.	P.31
Lock	タグのロック状態を変更します.	P.32
Kill	タグを使用不可にします.	P.33

4.2.2.4.1. SetCarrierOutputIntensityLevel コマンド

キャリア出力強度レベルを設定します. このコマンドは日本向け UR20 および UR21 でのみ使用可能です. 設定した値は切断/端末電源 OFF で初期化されます(初期値 6).

項目	型説明	
引数	VT_I4	キャリア出力強度レベルを指定します. 値範囲: 1 - 8
戻り値	なし	

使用例

GetCarrierOutputIntensityLevel コマンドを参照してください

4.2.2.4.2. GetCarrierOutputIntensityLevel コマンド

キャリア出力強度レベルを取得します. このコマンドは日本向け UR20 および UR21 でのみ使用可能です.

項目	型説明	
引数	なし	
戻り値	VT_I4	キャリア出力強度レベルを保持します.

使用例

```
' キャリア出力強度を取得
Dim carrierOutputIntensity As Long
carrierOutputIntensity = controller.Execute("GetCarrierOutputIntensityLevel")

' キャリア出力強度を設定
If carrierOutputIntensity <> 8 Then
    Call controller.Execute("SetCarrierOutputIntensityLevel", 8)
End If
```

4.2.2.4.3. SetCarrierOutputIntensitydBm コマンド

キャリア出力強度を dBm×10 の単位で設定します. このコマンドは 日本向け UR22 および海外向け UR20,UR21,UR22 で使用可能です. 設定した値は切断/端末電源 OFF で初期化されます(初期値 200).

項目	型説明	
引数	VT_I4	キャリア出力強度(dBm x 10)を指定します. 値範囲: 50 - 200
戻り値	なし	

使用例

GetCarrierOutputIntensitydBm コマンドを参照してください.

4.2.2.4.4. GetCarrierOutputIntensitydBm コマンド

キャリア出力強度を dBm×10 の単位で取得します. このコマンドは 日本向け UR22 および海外向け UR20,UR21,UR22

項目	型説明	
引数	なし	
戻り値	VT_I4	キャリア出力強度(dBm x 10)を保持します.

使用例

```
' キャリア出力強度を取得
Dim carrierOutputIntensity As Long
carrierOutputIntensity = controller.Execute("GetCarrierOutputIntensitydBm")

' キャリア出力強度を設定
If carrierOutputIntensity <> 200 Then
    Call controller.Execute("SetCarrierOutputIntensitydBm", 200)
End If
```

4.2.2.4.5. SetFrequency コマンド

キャリア出力を許可する周波数をビットフラグで設定します.設定した値は切断/端末電源 OFF で初期化されます(初期値は全対応 ch 有効を表す, 0x7FF8).

項目	型説明	
引数	VT_I4	キャリア出力を許可する周波数をビットフラグで指定します. 値範囲: 3bit(0x0008) - 14bit(0x7FF8)
戻り値	なし	

使用例

GetFrequency コマンドを参照してください.

4.2.2.4.6. GetFrequency コマンド

キャリア出力を許可する周波数をビットフラグで取得します.

項目	型説明	
引数	なし	
戻り値	VT_I4	キャリア出力を許可する周波数のビットフラグを保持します.

使用例

```
' キャリア出力を許可する周波数を取得
Dim frequencyFlg As Long
frequencyFlg = controller.Execute("GetFrequency")

' キャリア出力を許可する周波数を設定
If frequencyFlg <> &H7FF8 Then
```

```
    Call controller.Execute("SetFrequency", &H7FF8)
End If
```

4.2.2.4.7. SetAntennaPort コマンド

RFID タグ通信を行う際のキャリアを出力するアンテナポートをビットフラグで設定します. 設定した値は切断/端末電源 OFF で初期化されます(初期値は 0x01). このコマンドは UR21 と UR22 のみ使用できます. UR20 にてアンテナポート 2 を含む設定を行った場合, 読み取りが行えませんのでご注意ください.

項目	型説明	
引数	VT_I4	アンテナポートを示すビットフラグを指定します. 0x01: アンテナポート 1 0x02: アンテナポート 2 0x03: アンテナポート 1, 2
戻り値	なし	

使用例

GetAntennaPort コマンドを参照してください.

4.2.2.4.8. GetAntennaPort コマンド

RFID タグ通信を行う際のキャリアを出力するアンテナポートを取得します.

項目	型説明	
引数	なし	
戻り値	VT_I4	アンテナポートを示すビットフラグを保持します.

使用例

```
' キャリア出力を許可する周波数を取得
Dim antennaFlg As Long
antennaFlg = controller.Execute("GetAntennaPort")

' キャリア出力を許可する周波数を設定
If antennaFlg <> &H3 Then
    Call controller.Execute("SetAntennaPort", &H3)
End If
```

4.2.2.4.9. SetRereadPrevention コマンド

UII 読み取り処理，タグ通信処理での二度読み防止機能の有効/無効を設定します．二度読み防止機能が有効の場合は，一度検出した RF タグは ResetTerminal コマンドを実行するか，再接続を行わない限り再検出されません．設定した値は切断/端末電源 OFF で初期化されます(初期値は 0)．

項目	型説明	
引数	VT_I4	二度読み防止機能の有効/無効を指定します． 0: 無効 1: 有効
戻り値	なし	

使用例

GetRereadPrevention コマンドを参照してください．

4.2.2.4.10. GetRereadPrevention コマンド

UII 読み取り処理，タグ通信処理での RFID タグ検出完了判定の有効/無効を取得します．

項目	型説明	
引数	なし	
戻り値	VT_I4	二度読み防止機能の有効/無効を保持します．

使用例

```
'二度読み防止機能の有効/無効を取得
Dim rereadPrevention As Long
rereadPrevention = controller.Execute("GetRereadPrevention")

'二度読み防止機能の有効/無効を設定
If rereadPrevention <> 0 Then
    Call controller.Execute("SetRereadPrevention", 0)
End If
```

4.2.2.4.11. SetQValue コマンド

UII 取得時にタグからの応答が衝突する確率を調整するために使用する Q 値を設定します．設定した値は切断/端末電源 OFF で初期化されます(初期値は 4)．

項目	型説明	
引数	VT_I4	Q 値を指定します． 範囲: 0 - 7
戻り値	なし	

使用例

GetQValue コマンドを参照してください．

4.2.2.4.12. GetQValue コマンド

UII 取得時にタグからの応答が衝突する確率を調整するために使用する Q 値を取得します.

項目	型説明	
引数	なし	
戻り値	VT_I4	Q 値を保持します.

使用例

```
' Q値を取得
Dim qValue As Long
qValue = controller.Execute("GetQValue")

' Q値を設定
If qValue <> 4 Then
    Call controller.Execute("SetQValue", 0)
End If
```

4.2.2.4.13. SetSession コマンド

タグの UII 取得時に使用するセッションを設定します. これを設定することにより, タグが再度 UII 取得に応答する条件を変更することができます. 詳細は『ISO/IEC18000-63 (GS1 Gen2)』を参照してください. 設定した値は切断/端末電源 OFF で初期化されます(初期値は 0).

項目	型説明	
引数	VT_I4	タグの UII 取得時に使用するセッションを指定します. 0: RFID タグ通信待機状態後, 再度同じ RF タグの UII 取得が可能です. 1: RFID タグの UII データ取得後, 規定時間以上経過すれば再度同じ RFID タグの UII データの取得が可能となります. 2: RFID タグ通信待機状態後, 規定時間以上経過すれば再度同じ RFID タグの UII 取得が可能となります. 3:2 と同じです.
戻り値	なし	

使用例

GetSession コマンドを参照してください.

4.2.2.4.14. GetSession コマンド

タグの UII 取得時に使用するセッションを取得します.

項目	型説明	
引数	なし	
戻り値	VT_I4	セッションを保持します.

使用例

```
' セッションを取得
Dim session As Long
session = controller.Execute("GetSession")

' セッションを設定
If session <> 0 Then
    Call controller.Execute("SetSession", 0)
End If
```

4.2.2.4.15. ResetTerminal コマンド

実行中の処理(UII 読み取り, タグ通信)を中断します. 本処理を実行すると, 実行中の処理結果は破棄されます. 二度読み防止機能が有効の場合, 読み取り済みのデータもクリアされます.

項目	型説明
引数	なし
戻り値	なし

使用例

```
Call controller.Execute("ResetTerminal")
```

4.2.2.4.16. Reboot コマンド

RFID テーブルスキャナを電源再起動します.

項目	型説明
引数	なし
戻り値	なし

使用例

```
Call controller.Execute("Reboot")
```

4.2.2.4.17. GetSerialNumber コマンド

RFID テーブルスキャナの製造番号情報を取得します.

項目	型説明		
引数	なし		
戻り値	VT_ARRAY | VT_BSTR		
	1	VT_BSTR	メイン基板製造番号を保持します.
	2	VT_BSTR	RF 基板製造番号を保持します.

使用例

```
' 製造番号取得
Dim serialNumbers As Variant
serialNumbers = controller.Execute("GetSerialNumber")

' メイン基板製造番号
Dim mainSerial As String
mainSerial = serialNumbers(0)
' RF基板製造番号
Dim rfSerial As String
rfSerial = serialNumbers(1)
```

4.2.2.4.18. GetVersion コマンド

RFID テーブルスキャナのバージョン情報を取得します.

項目	型説明		
引数	なし		
戻り値	VT_ARRAY | VT_VARIANT		
	0	VT_I4	OS バージョンを保持します.
	1	VT_I4	メインアプリバージョンを保持します.
	2	VT_I4	RF チップソフトウェアバージョンを保持します.
	3	VT_I4	OEM ソフトウェアバージョンを保持します.

使用例

```
' バージョン取得
Dim versions() As Long
versions = controller.Execute("GetVersion")

' OSバージョン
Dim osVer As Long
osVer = versions(0)
' メインアプリバージョン
Dim mainAppVer As Long
mainAppVer = versions (1)
' RFチップソフトウェアバージョン
Dim rfTipVer As Long
```

```
    rfTipVer = versions(2)
    ' OEMソフトウェアバージョン
    Dim oemVer As Long
    oemVer = versions(3)
```

4.2.2.4.19. KeepAlive コマンド

RFID テーブルスキャナに端末疎通コマンドを送信します. TCP/IP 接続の場合, スキャナとの通信が 60 秒間ない場合, スキャナは回線断と判断してソケットを閉じるため, それ以内に本コマンドを送信してください.

項目	型説明
引数	なし
戻り値	なし

使用例

```
' キープアライブ実行メソッド 10秒に1回KeepAliveコマンドを実行します.
Private Sub DoKeepAlive()
    ' KeepAliveコマンドを実行
    Call controller.Execute("KeepAlive")
    ' 10秒に1回KeepAliveを実行するのを繰り返す
    Call Application.OnTime(Now + TimeSerial(0, 0, 10), "DoKeepAlive")
End Sub
```

4.2.2.4.20. ReadUII コマンド

RFID タグの検出を行い, UII データを読み取ります. 実行されたときに複数の RFID タグが検出可能範囲に存在していればそのすべてが読み取れます.

項目	型説明			
引数	VT_BOOL			付加情報も合わせて取得するかを指定します. False: 付加情報は取得しません. True: 付加情報も取得します.
戻り値	VT_ARRAY | VT_VARIANT			
	i	VT_ARRAY | VT_VARIANT		1 つの RFID タグに対応するデータを保持します.
		0	VT_I4	i 番目の RFID タグの PC データを保持します.
		1	VT_UI1 | VT_ARRAY	i 番目の RFID タグの UII データをバイト配列で保持します.
		2	VT_I4	付加情報取得時のみ存在します. データ受信時の RSSI 値を保持します.
		3	VT_I4	付加情報取得時のみ存在します. データ受信時のアンテナ番号を保持します.

使用例

```
    Dim uiis As Variant
    uiis = controller.Execute("ReadUII", False)
    If Not IsEmpty(uiis) Then
        ' 検出された個数を取得
        Dim detectedElem As Long
        detectedElem = UBound(uiis)

        ' 検出されたRFIDタグ数分繰り返す
        Dim i As Long
        For i = 0 To detectedElem
            ' PCデータ取得
            Dim pcData As Long
            pcData = uiis(i)(0)

            ' UIIデータを取得
            Dim uiiData() As Byte
            uiiData = uiis(i)(1)
            ' UIIデータサイズを取得
            Dim uiiDataSize As Long
            uiiDataSize = UBound(uiiData)
            ' UIIデータサイズ分データ抽出
            Dim j As Long
            For j = 0 To uiiDataSize
                Dim value As Byte
                value = uiiData(0)
            Next j
        Next i
    End If
```

4.2.2.4.21. StartContinuousRFDetection コマンド

連続 UII 読み取りを開始します. ReadContinuousUII と StopContinuousRFDetection コマンドは本コマンド実行後に実行してください.

項目	型説明	
引数	VT_UI1	反応モードを指定します. 0: 連続反応モード コマンド内で同一 UII を持つ RFID タグを何度でも検出します. 1: 二度読み防止モード 同一の UII を持つ RFID タグを 1 度だけ検出します.
戻り値	なし	

使用例

ReadContinuousUII コマンドを参照してください.

4.2.2.4.22. ReadContinuousUII コマンド

StartContinuousRFDetection を実行後から検出された RFID タグの UII データを読み取ります.

項目	型説明			
引数	VT_BOOL			付加情報も合わせて取得するかを指定します. False: 付加情報は取得しません. True: 付加情報も取得します.
戻り値	VT_ARRAY | VT_VARIANT			
	i	VT_ARRAY | VT_VARIANT		1 つの RFID タグに対応するデータを保持します.
		0	VT_I4	i 番目の RFID タグの PC データを保持します.
		1	VT_UI1 | VT_ARRAY	i 番目の RFID タグの UII データをバイト配列で保持します.
		2	VT_I4	付加情報取得時のみ存在します. データ受信時の RSSI 値を保持します.
		3	VT_I4	付加情報取得時のみ存在します. データ受信時のアンテナ番号を保持します.

使用例

```
' <summary> 連続読出しが開始しているか否か </summary>
Private isRunning As Boolean

' <summary>
' 連続読出し開始メソッド
' </summary>
Public Sub StartContinuousRF()
    Call controller.Execute("StartContinuousRFDetection", 0)
    isRunning = True
    Call DoReadContinuousUII
End Sub

' <summary>
' 連続読出し停止メソッド
' </summary>
Public Sub StopContinuousRF()
    isRunning = False
    Call controller.Execute("StopContinuousRFDetection")
End Sub

' <summary>
' 連続読出し実行
' </summary>
Private Sub DoReadContinuousUII()
    If Not isRunning Then
        Exit Sub
```

```
    End If

    Dim uiis As Variant
    uiis = controller.Execute("ReadContinuousUII", False)
    If Not IsEmpty(uiis) Then
        ' 検出された個数を取得
        Dim detectedElem As Long
        detectedElem = UBound(uiis)

        ' 検出されたRFIDタグ数分繰り返す
        Dim i As Long
        For i = 0 To detectedElem
            ' PCデータ取得
            Dim pcData As Long
            pcData = uiis(i)(0)

            ' UIIデータを取得
            Dim uiiData() As Byte
            uiiData = uiis(i)(1)
            ' UIIデータサイズを取得
            Dim uiiDataSize As Long
            uiiDataSize = UBound(uiiData)
            ' UIIデータサイズ分データ抽出
            Dim j As Long
            For j = 0 To uiiDataSize
                Dim value As Byte
                value = uiiData(0)
            Next j
        Next i
    End If
    executeTime = Now + TimeSerial(0, 0, 1)
    Call Application.OnTime(executeTime, "DoReadContinuousUII")
End Sub
```

4.2.2.4.23. StopContinuousRFDetection コマンド

連続 UII 読み取りを停止します.

項目	型説明
引数	なし
戻り値	なし

使用例

ReadContinuousUII コマンドを参照してください.

4.2.2.4.24. ReadMemory コマンド

RFID タグの指定メモリのデータを読み出します.

項目	型説明			
引数	VT_VARIANT| VT_ARRAY			
	0	VT_I4		出力するアンテナポートをビットフラグで指定します. 0x00: パラメータ設定の内容に従います. 0x01: アンテナポート 1 を使用します. 0x02: アンテナポート 2 を使用します.
	1	VT_I4		読み取りを行う RFID タグのメモリバンクを指定します. 0x00: Rederved Bank 0x01: UII Bank 0x02: TID Bank 0x03: User Bank
	2	VT_I4		読み取りバイト数を 2 バイト単位で指定します. 範囲: 2 - 256
	3	VT_I4		読み取り開始ポインタを 2 バイト単位で指定します. 範囲: タグに依存します.
	4	VT_I4		アクセスパスワードを指定します. 読み取りロック状態領域へのアクセス以外は 0 を指定してください.
	5	VT_VARIANT | VT_ARRAY		省略可能です. 読込み対象の RFID 情報を指定します. 指定しなかった場合は検出したすべての RFID を対象とします.
		0	VT_I4	RFID タグの PC データを指定します.
		1	VT_UI1 | VT_ARRAY	RFID タグの UII データをバイト配列で指定します.
戻り値	VT_VARIANT | VT_ARRAY			
	j	VT_VARIANT | VT_ARRAY		RFID タグ 1 つ分のデータを保持します.
		0	VT_I4	RFID タグの PC タグを保持します.
		1	VT_UI1 | VT_ARRAY	RFID タグの UII データをバイト配列で保持します.
		2	VT_UI1 | VT_ARRAY	読み込んだメモリデータをバイト配列で保持します.

使用例

```
    ' 検出される RFID すべてを読み込み対象にする.
    ' アンテナポート: 設定に従う
    ' 読み取り対象のメモリバンク: Reserved Bank
    ' 読み取りバイト数: 4Byte
    ' 読み取り開始位置: 0
    ' アクセスパスワード: 0
    Dim readParam() As Variant
    readParam = Array(0, 0, 4, 0, 0)

    Dim memory As Variant
    memory = controller.Execute("ReadMemory", readParam)
    If Not IsEmpty(memory) Then
        ' 検出された個数を取得
        Dim detectedElem As Long
        detectedElem = UBound(memory)

        ' 検出された RFID タグ数分繰り返す
        Dim i As Long
        For i = 0 To detectedElem
            ' メモリデータを取得
            Dim memoryData() As Byte
            memoryData = memory(i)(2)

            ' メモリサイズを取得
            Dim memorySize As Long
            memorySize = UBound(memoryData)
            ' メモリサイズ分データ抽出
            Dim j As Long
            For j = 0 To memorySize
              ' 現在の値+1 を行う
              memoryData(j) = memoryData(j) + 1
            Next j

            ' 読み込んだ RFID タグを特定してメモリに値を書込む
            Dim rfidData(1) As Variant
            rfidData(0) = memory(i)(0)
            rfidData(1) = memory(i)(1)
            Dim writeParam() As Variant
            writeParam = Array(0, 0, 4, 0, 0, memoryData, rfidData)
            Call controller.Execute("WriteMemory", writeParam)
        Next i

    End If
```

4.2.2.4.25. WriteMemory コマンド

RFID タグの指定メモリのデータを書き込みます.

項目	型説明			
引数	VT_VARIANT| VT_ARRAY			
	0	VT_I4		出力するアンテナポートをビットフラグで指定します. 0x00: パラメータ設定の内容に従います. 0x01: アンテナポート 1 を使用します. 0x02: アンテナポート 2 を使用します.
	1	VT_I4		書き込みを行う RFID タグのメモリバンクを指定します. 0x00: Rederved Bank 0x01: UII Bank 0x02: TID Bank 0x03: User Bank
	2	VT_I4		書き込みバイト数を 2 バイト単位で指定します. 範囲: 2 - 64
	3	VT_I4		書き込み開始ポインタを 2 バイト単位で指定します. 範囲: タグに依存します.
	4	VT_I4		アクセスパスワードを指定します. 書き込みロック状態領域へのアクセス以外は 0 を指定してください.
	5	VT_UI1 | VT_ARRAY		書き込みデータ
	6	VT_VARIANT | VT_ARRAY		省略可能です. 書き込み対象の RFID 情報を指定します. 指定しなかった場合は検出したすべての RFID を対象とします.
		0	VT_I4	RFID タグの PC データを指定します.
		1	VT_UI1 | VT_ARRAY	RFID タグの UII データをバイト配列で指定します.
戻り値	VT_VARIANT | VT_ARRAY			
	j	VT_VARIANT | VT_ARRAY		RFID タグ 1 つ分のデータを保持します.
		0	VT_I4	RFID タグの PC タグを保持します.
		1	VT_UI1 | VT_ARRAY	RFID タグの UII データをバイト配列で保持します.

使用例

ReadMemory コマンドを参照してください.

4.2.2.4.26. Lock コマンド

タグのロック状態を変更します.

項目	型説明		
引数	VT_VARIANT| VT_ARRAY		
	0	VT_I4	出力するアンテナポートをビットフラグで指定します. 0x00: パラメータ設定の内容に従います. 0x01: アンテナポート 1 を使用します. 0x02: アンテナポート 2 を使用します.
	1	VT_I4	対象を指定します. 対象がパスワードの場合は Read/Write 両方に対して, その他は Write に対してのみロックが適用されます. 0x01: キルパスワード 0x02: アクセスパスワード 0x10: UII Bank 0x20: TID Bank 0x40: User Bank
	2	VT_I4	ロック状態を指定します. 0x00: アンロック 0x01: ロック 0x10: 永久アンロック 0x11: 永久ロック ※ 永久アンロック / ロックを行ったタグはそれ以降ロック状態の変更ができなくなります.
	3	VT_I4	アクセスパスワードを設定します.
	4	VT_VARIANT | VT_ARRAY	省略可能です. 読込み対象の RFID 情報を指定します. 指定しなかった場合は検出したすべての RFID を対象とします.
		0 VT_I4	RFID タグの PC データを指定します.
		1 VT_UI1 | VT_ARRAY	RFID タグの UII データをバイト配列で指定します.
戻り値	VT_VARIANT | VT_ARRAY		
	j	VT_VARIANT | VT_ARRAY	RFID タグ 1 つ分のデータを保持します.
		0 VT_I4	RFID タグの PC タグを保持します.
		1 VT_UI1 | VT_ARRAY	RFID タグの UII データをバイト配列で保持します.

使用例

```
    ' 検出される RFID すべてを読み込み対象にする.
    ' アンテナポート: 設定に従う
    ' ロック対象: アクセスパスワード
    ' ロック状態: ロック
    ' アクセスパスワード: 0
    Dim readParam() As Variant
    readParam = Array(0, 0, 4, 0, 0)

    Dim memory As Variant
    memory = controller.Execute("ReadMemory", readParam)
    If Not IsEmpty(memory) Then
        ' 検出された個数を取得
        Dim detectedElem As Long
        detectedElem = UBound(memory)

        ' 検出された RFID タグ数分繰り返す
        Dim i As Long
        For i = 0 To detectedElem
            ' 読み込んだ RFID タグを特定してロックをおこなう
            Dim rfidData(1) As Variant
            rfidData(0) = memory(i)(0)
            rfidData(1) = memory(i)(1)
            Dim lockParam() As Variant
            lockParam = Array(0, 1, 1, 0,  rfidData)
            Call controller.Execute("Lock", lockParam)
        Next i

    End If
```

4.2.2.4.27. Kill コマンド

タグを使用不可にします. このコマンドを実行した RFID タグはその後, 使用できなくなりますのでご注意ください.

項目	型説明		
引数	VT_VARIANT| VT_ARRAY		
	0	VT_I4	出力するアンテナポートをビットフラグで指定します. 0x00: パラメータ設定の内容に従います. 0x01: アンテナポート 1 を使用します. 0x02: アンテナポート 2 を使用します.
	1	VT_I4	キルパスワードを指定します.

項目	型説明			
	2	VT_VARIANT | VT_ARRAY		省略可能です. 読込み対象の RFID 情報を指定します. 指定しなかった場合は検出したすべての RFID を対象とします.
		0	VT_I4	RFID タグの PC データを指定します.
		1	VT_UI1 | VT_ARRAY	RFID タグの UII データをバイト配列で指定します.
戻り値	VT_VARIANT | VT_ARRAY			
	j	VT_VARIANT | VT_ARRAY		RFID タグ 1 つ分のデータを保持します.
		0	VT_I4	RFID タグの PC タグを保持します.
		1	VT_UI1 | VT_ARRAY	RFID タグの UII データをバイト配列で保持します.

使用例

```
' 検出される RFID すべてを対象にする.
' アンテナポート: 設定に従う
' キルパスワード: 1
' 指定の UII タグ: 検出した UII タグデータ

Dim readParam() As Variant
readParam = Array(0, 0, 4, 0, 0)

Dim memory As Variant
memory = controller.Execute("ReadMemory", readParam)
If Not IsEmpty(memory) Then
    ' 検出された個数を取得
    Dim detectedElem As Long
    detectedElem = UBound(memory)

    ' 検出された RFID タグ数分繰り返す
    Dim i As Long
    For i = 0 To detectedElem
        ' 読み込んだ RFID タグを特定して個別に Kill をおこなう
        Dim rfidData(1) As Variant
        rfidData(0) = memory(i)(0)
        rfidData(1) = memory(i)(1)
        Dim writeParam() As Variant
        KillParam = Array(0, 1, rfidData)
        Call controller.Execute("Kill", KillParam)
    Next i

End If
```

4.2.3. CaoVariable クラス

4.2.3.1. Value プロパティ

接続したスキャナからデータを取得します. 変数名によって動作が異なります. 詳細は, 4.3.変数一覧を参照してください.

4.3. 変数一覧

各クラスで使用可能な変数一覧を定義します. なお変数は, CaoVariable クラスのオブジェクトを指します.

CaoController クラス変数

変数名	説明	Value		参照
		get	put	
@MAKER_NAME	メーカー名を取得します.	○	-	P.35
@VERSION	プロバイダバージョンを取得します.	○	-	P.36
@SERIAL	製造番号を取得します	○		P.37
@LASTDEVICEERROR	直前のデバイスエラー情報を取得します. 0x80100001 が発生していない場合は Empty となります.	○	-	P.38

4.3.1.1. @MAKER_NAME

メーカー名の取得をします.

データ型

型説明	
VT_BSTR	メーカー名を取得します.

使用例

```
    ' 変数追加
    Dim variable As CaoVariable
    Set 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を取得します.

データ型

型説明		
VT_ARRAY | VT_I4		
0	VT_I4	OS バージョン
1	VT_I4	メインアプリバージョン
2	VT_I4	RF アプリバージョン
3	VT_I4	RF チップソフトウェアバージョン
4	VT_I4	OEM ソフトウェアバージョン

使用例

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

    Dim version As Variant
    version = variable.value

    ' OS バージョン
    If Not IsEmpty(varsion(0)) Then
        Dim os As Integer
        os = version(0)
    End If
    ' メインアプリバージョン
    If Not IsEmpty(version(1)) Then
        Dim mainAp As Integer
        mainAp = version(1)
    End If
    ' RF アプリバージョン
    If Not IsEmpty(version(2)) Then
        Dim rfAp As Integer
        rfAp = version(2)
    End If
    ' RF チップソフトソフト
    If Not IsEmpty(version(3)) Then
        Dim rfFirm As Integer
        rfFirm = version(3)
    End If
    ' OEM ソフトウェアバージョン
    If Not IsEmpty(version(4)) Then
        Dim oem As Integer
        oem = version(4)
    End If
```

4.3.1.3. @SERIAL

プロバイダバージョンを取得します.

データ型

型説明		
VT_ARRAY | VT_BSTR		
0	VT_BSTR	メイン基板製造番号
1	VT_BSTR	RF 基盤製造番号

使用例

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

    Dim serial As Variant
    serial = variable.value

    ' メイン基板製造番号
    If Not IsEmpty(serial(0)) Then
        Dim mainNo As String
        mainNo = serial(0)
    End If
    ' RF 基盤製造番号
    If Not IsEmpty(serial(1)) Then
        Dim rfNo As String
        rfNo = serial(1)
    End If
```

4.3.1.4. @LASTDEVICEERROR

直前のデバイスエラー情報を取得します. 0x80100001 が発生していない場合は Empty となります.

データ型

型説明		
VT_ARRAY | VT_VARIANT		
0	VT_I2	SW の値を保持します.
1	VT_I2	Result の値を保持します. ※ Result が存在しない場合は VT_EMPTY になります.
2	VT_I2	Error1 の値を保持します. ※ Result が存在しない場合は VT_EMPTY になります.
3	VT_I2	Error2 の値を保持します. ※ Result が存在しない場合は VT_EMPTY になります.

使用例

```
    ' 変数追加
    Dim variable As CaoVariable
    Set variable = controller.AddVariable("@LASTDEVICEERROR")
    ' 値取得
    Dim lastError As Variant
    lastError = variable.Value
    If Not IsEmpty(lastError) Then
        ' スイッチ部
        Dim sw As Integer
        sw = lastError(0)

        ' Result データ
        If Not IsEmpty(lastError(1)) Then
            Dim result As Integer
            result = lastError(1)
        End If
        ' Error1
        If Not IsEmpty(lastError(2)) Then
            Dim error1 As Integer
            error1 = lastError(3)
        End If
        ' Error2
        If Not IsEmpty(lastError(3)) Then
            Dim error2 As Integer
            error2 = lastError(3)
        End If
    End If
```

5. UR20 プロバイダエラーコード

本プロバイダには，0x8011****でマスクした以下の独自エラーコードが存在します．（表 5-1 独自エラーコード表参照）

ORiN2 の共通エラーについては，「ORiN2 プログラミングガイド」のエラーコードの章を参照してください．

表 5-1 独自エラーコード表

エラー番号	説明
0x80110001	必須オプションが未定義です．必須オプションを指定してください．
0x80110002	想定外のレスポンスを受信しました．対応機種かを確認してください．
0x80110003	受信データのチェックサムが不正です．通信状態を確認してください．

付録A. 通信プロトコルコマンド対応表

CaoWorkspace

メソッド	通信コマンド
AddController	APロード
	RF Open
Controllers::Remove	RF Close
	AP アンロード

CaoController::Execute

コマンド	通信コマンド
SetCarrierOutputIntensityLevel	パラメータ設定＜キャリア出力強度設定(レベル指定)＞
GetCarrierOutputIntensitydLevel	パラメータ取得＜キャリア出力強度設定(レベル指定)＞
SetCarrierOutputIntensitydBm	パラメータ設定＜キャリア出力強度設定(dBm指定)＞
GetCarrierOutputIntensitydBm	パラメータ取得＜キャリア出力強度設定(dBm指定)＞
SetFrequency	パラメータ設定＜周波数設定＞
GetFrequency	パラメータ取得＜周波数設定＞
SetAntennaPort	パラメータ設定＜アンテナポート設定＞
GetAntennaPort	パラメータ取得＜アンテナポート設定＞
SetRereadPrevention	パラメータ設定＜二度読み防止設定＞
GetRereadPrevention	パラメータ取得＜二度読み防止設定＞
SetQValue	パラメータ設定＜Q 値設定＞
GetQValue	パラメータ取得＜Q 値設定＞
SetSession	パラメータ設定＜セッション設定＞
GetSession	パラメータ取得＜セッション設定＞
ResetTerminal	処理中断
Reboot	端末リセット
GetSerialNumber	製造番号取得
GetVersion	バージョン取得
KeepAlive	端末疎通確認
ReadUII	UII読み取り(開始)
	UII読み取り(結果確認)
	UII読み取り(バッファ取り出し)
	UII読み取り(付加情報付きバッファ取り出し)

StartContinuousRFDetection	連続UII読み取り(開始)
ReadContinuousUII	連続UII読み取り(データ取得)
	連続UII読み取り(付加情報付きデータ取得)
StopContinuousRFDetection	連続UII読み取り(終了)
ReadMemory	タグ通信(開始) メモリリード
	タグ通信(結果確認)
	タグ通信(バッファ取り出し) メモリリード
WriteMemory	タグ通信(開始) メモリライト
	タグ通信(結果確認)
	タグ通信(結果確認) メモリライト, ロック, キル
Lock	タグ通信(開始) ロック
	タグ通信(結果確認)
	タグ通信(結果確認) メモリライト, ロック, キル
Kill	タグ通信(開始) キル
	タグ通信(結果確認)
	タグ通信(結果確認) メモリライト, ロック, キル

付録B. スキャナからのエラー一覧

UR20 プロバイダはスキャナからのエラーを 0x8010****でマスクした値で返します.

エラー番号	説明
0x80100001	デバイスからのエラーです. @LASTDEVICEERROR変数の値を確認してください.
0x801001xx	RFタグとの通信中エラーです. 通信環境の見直しをおこなってください. 指定パスワードを確認してください.
0x80100203	メモリ範囲外へのアクセスです. 指定したパラメータを確認してください.
0x80100204	ロックメモリへのアクセスです. パスワード認証を行ってください. (永久ロックの場合は行えません.)
0x8010020B	メモリライト電力不足です.
0x8010020F	その他のエラーです.