

# デンソー RFID Scanner UR40 プロバイダ

Version 1.2.0

ユーザーズ ガイド

August 23, 2024



**【改版履歴】**

バージョン	日付	内容
1.0.0	2019-12-02	初版.
1.0.1	2022-07-13	Error メッセージイベントのメッセージ内容を修正. 誤字修正.
1.1.0	2022-10-18	Write, WriteUII, Lock, LockUII, Kill, KillUII コマンド追加. 誤記訂正.
	2024-07-23	付録にタグの仕様を追加. 誤記修正.
1.2.0	2024-08-23	ConnectionLog オプションを追加.

**【対象機種】**

機種	備考
UR40-H-ERU	
UR50-H-ERU	

**【動作確認機種】**

機種	バージョン	注意事項
UR50-H-ERU	Ver. 1.11.00	

## 目次

1. はじめに.....	6
1.1. 本書が想定している開発環境.....	7
1.2. 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. スキャナとクライアント PC との接続.....	8
2.2. 読み取りデータ転送について.....	9
2.3. 接続の設定.....	9
2.4. PC 開発環境のセットアップ.....	9
2.4.1. UR40 プロバイダの自動インストール.....	9
2.4.2. UR40 プロバイダの手動インストール.....	10
3. コマンドリファレンス.....	11
3.1. メソッド/プロパティ一覧.....	11
3.2. メソッド・プロパティ.....	11
3.2.1. CaoWorkspace クラス.....	11
3.2.1.1. AddController メソッド.....	11
3.2.2. CaoController クラス.....	13
3.2.2.1. VariableNames プロパティ.....	13
3.2.2.2. Variables プロパティ.....	13
3.2.2.3. AddVariable メソッド.....	13
3.2.2.4. Execute メソッド.....	14
3.2.3. CaoVariable クラス.....	14
3.2.3.1. Value プロパティ.....	14
3.2.4. OnMessage クラス.....	14
3.3. Execute コマンド.....	15
3.3.1. 概略.....	15
3.3.2. Execute コマンド一覧.....	16
3.3.3. コマンドによるタグ制御.....	17
3.3.4. 設定変更コマンド.....	18
3.3.4.1. TriggerContinuous.....	18
3.3.4.2. TriggerSingle.....	18

3.3.5. RF タグ制御コマンド .....	19
3.3.5.1. Inventory .....	19
3.3.5.2. InventoryUII .....	19
3.3.5.3. Read .....	20
3.3.5.4. ReadUII .....	21
3.3.5.5. Write .....	22
3.3.5.6. WriteUII .....	23
3.3.5.7. Lock .....	24
3.3.5.8. LockUII .....	25
3.3.5.9. Kill .....	26
3.3.5.10. KillUII .....	26
3.3.5.11. CommandRelease .....	27
3.3.6. トリガコマンド .....	27
3.3.6.1. TriggerOn .....	27
3.3.6.2. TriggerOff .....	27
3.3.7. その他のコマンド .....	28
3.3.7.1. Reset .....	28
3.3.7.2. Raw .....	28
3.4. 変数一覧 .....	29
3.4.1. CaoController クラス変数 .....	29
3.4.1.1. @MAKER_NAME .....	29
3.4.1.2. @VERSION .....	29
3.4.1.3. @DEVICE_VERSION .....	30
3.4.1.4. @ID .....	30
3.4.1.5. @READ_CONNECTED .....	30
3.5. イベント一覧 .....	31
3.5.1. Error メッセージ .....	31
3.5.2. Inventory メッセージ .....	32
3.5.3. Read メッセージ .....	33
3.5.4. Write メッセージ .....	34
3.5.5. Lock メッセージ .....	35
3.5.6. Kill メッセージ .....	36
3.5.7. その他メッセージ .....	37
<b>4. UR40 プロバイダによるプログラミング .....</b>	<b>38</b>
4.1. スキャナに接続し RF タグの UII データを読み込むサンプルプログラミング .....	38
4.1.1. サンプルプログラム .....	38

---

4.1.1.1. 接続 .....	40
4.1.1.2. スキャナを UI データの取得モードで実行 .....	41
4.1.1.3. タグ読み込み時のメッセージを受信 .....	41
4.1.1.4. 切断 .....	41
5. UR40 プロバイダエラーコード .....	43

## 1. はじめに

本書は、RFID Scanner の UR40 / UR50 シリーズから RFID タグ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 に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 UR40 プロバイダ、RFID Scanner をスキャナと呼称します。UR40 プロバイダはスキャナと LAN 通信を行います。

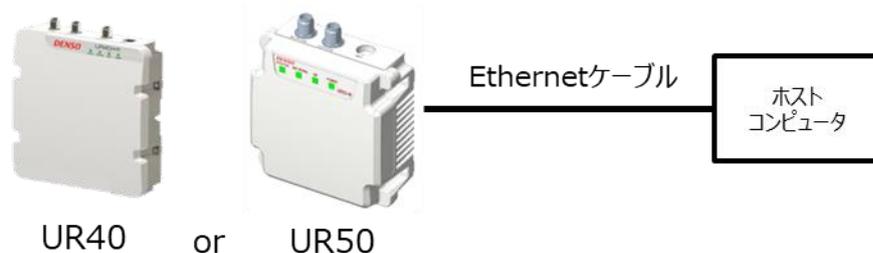


図 1-1 構成図

スキャナに対し、コマンド制御をする際はスキャナを TCPServer に設定してください。読み取りデータ転送は TCPClient に設定してください。プロバイダはコマンド制御については TCPClient、読み取りデータ転送については TCPServer として動作します。図 1-2 接続概要が概要になります。

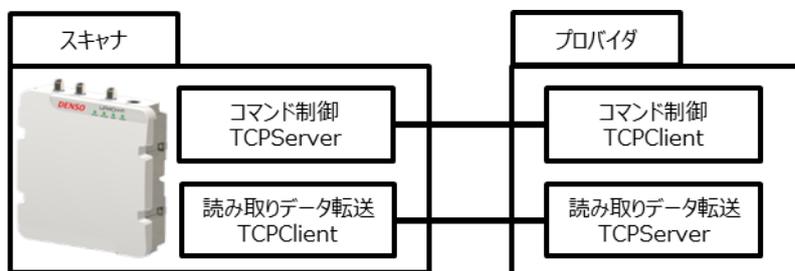


図 1-2 接続概要

また、本プロバイダ及びデバイスそれぞれの対応を図 1-3に表します。

(※一例です。全てを表しているわけではありません。)

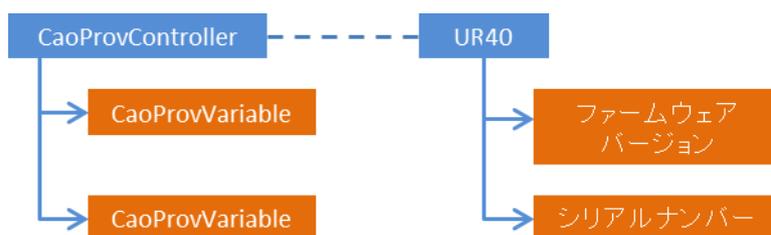


図 1-3 プロバイダの構成とデバイス情報との対応図

### 1.1. 本書が想定している開発環境

クライアント PC が Windows, 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」を参照してください.

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

- ORiN2¥CAO¥Doc¥ORiN2\_ProgrammersGuide\_<lang>.pdf

※<lang>の部分は環境毎の言語文字列に置き換えてお読みください.

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

## 2. アプリケーション開発のための環境セットアップ

### 2.1. スキャナとクライアント PC との接続

スキャナと接続する際は、スキャナの電源が入っている状態でクライアント PC との接続を開始します。スキャナ側は表 2-1 の設定を行ってから接続してください。

スキャナの設定はデンソーウェブの Scanner Setting F から実施してください。Scanner Setting F の画面と設定の対応は 5. 付録 C を参照してください。

表 2-1 プロバイダ利用の際のスキャナの設定

分類		設定項目		
RFID	RF タグ通信動作モード		上位コントロールモード	
通信	読み取りデータ送信インターフェース		Ethernet	
	LAN	IP アドレス設定	IP アドレス	DHCP 以外の任意のアドレス
			サブネットマスク	任意
			デフォルトゲートウェイ	任意
		コマンド制御	プロトコル	TCP Server
	読み取りデータ転送先	プロトコル	TCP Client	
トリガ	トリガタイプ		コマンド	
	トリガコマンド設定	トリガコマンドレスポンス	許可	
		トリガ ON コマンド	TRGON	
		トリガ OFF コマンド	TRGOFF	
失敗通知データ	失敗通知データを出力する	有効		
	失敗通知データキャラクタ	ERROR		
出力フォーマット	セパレータ		カンマ	
	スキャナ ID		許可	
	時刻		許可	
	任意文字列 1		データ無し	
	読み取り条件番号		許可	
	通信識別子		許可	
	レスポンス方式		ASCII レスポンス	
	PC		許可	
RSSI		許可		

	アンテナ番号	許可
	偏波	許可
	任意文字列 2	データ無し

## 2.2. 読み取りデータ転送について

プロバイダ起動直後はスキャナとの接続が完了しておらず読み取りデータが受信できない場合があります。@READ\_CONNECTED 変数を取得し、True になっていることを確認してからタグを読み取りください。

## 2.3. 接続の設定

スキャナの IP アドレスとポートの設定とプロバイダの設定の対応は以下のとおりです。

スキャナ		プロバイダ
IP アドレス設定		Conn オプションの IP アドレス
コマンド制御	待ち受けポート番号	Conn オプションのポート番号
読み取りデータ送信	接続先 IP アドレス	プロバイダでの対応はありません。 プロバイダを動作させる PC の IP アドレスと対応します。
	接続先ポート番号	ReadPort オプション

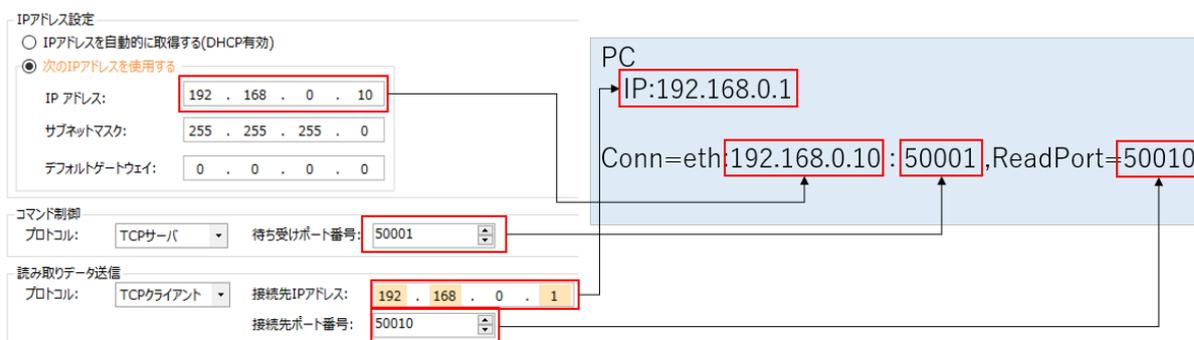


図 2-1 設定ソフトとの対応図

## 2.4. PC 開発環境のセットアップ

### 2.4.1. UR40 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、UR40/UR50 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, 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。

#### 2.4.2. UR40 プロバイダの手動インストール

UR40 プロバイダを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。

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 2-2 UR40 プロバイダ

ファイル名	GaoProvDENSOUR40.dll
ProgID	GaoProv.DENS0.UR40
レジストリ登録	regsvr32 GaoProvDENSOUR40.dll
レジストリ登録の抹消	regsvr32 /u GaoProvDENSOUR40.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. 11
<b>CaoController</b>			
	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. 13
	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. 14
<b>CaoVariable</b>			
	Value	P 値の取得/設定	P. 14

### 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

CaoWorkspace に、コントローラオブジェクトを追加します。UR40 プロバイダでは、AddController メソッド実行時に渡されたパラメータを参照し、該当するスキャナと接続を行います。以下に、AddController メソッドの仕様を示します。

#### 書式

##### CaoController AddController

```
(
    "<コントローラ名>",           // コントローラ名(任意)
    "CaoProv. DENSO. UR40",      // 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
)
```

#### オプション

<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。

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 =<Eth 接続オプション>	○	接続するデバイスの IP アドレスを指定します。	--	--
ReadPort =<読み取りデータ受信ポート>	--	読み取りデータ受信ポートを指定します。	1-65535	50010
MessageMax =<Message オブジェクト生成数>	--	UR40 プロバイダが生成する Message オブジェクト数の制限を設定します。	1-4294967295	100
ConnectionLog =<接続ログ出力フラグ>	--	読み取りデータポートと UR40 との通信が確立していないことを示すログの出力の有無を設定します。	TrueまたはFalse	True

**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  ' Controllerオブジェクト

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("SampleController", _
    "CaoProv. DENSO. UR40", _
    "", _
    "conn=eth:192.168.0.10,ReadPort=50010")
```

**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。なお、本プロバイダでは ETH と指定すると TCP 接続を行います。

**ETH**

```
"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 IP>          : 接続先 IP アドレス. "192.168.0.10" etc..
<接続ポート>      : 接続先ポート番号. 50001
<ローカル IP>     : ローカル IP アドレス.
<ローカルポート> : ローカルポート番号.
```

### 3.2.2. CaoController クラス

#### 3.2.2.1. 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

```
' Connect
Call Connect

' Get variable name list
Dim variables as Variant
variables = controller.VariableNames

' Disconnect
Call Disconnect
```

#### 3.2.2.2. 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

```
' Connect
Call Connect

' Variable collection
Dim variables as CaoVariables
Set variables = controller.Variables

' Get variable
Dim variable as CaoVariable
Set variable = variables.Item(0)

' Disconnect
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

#### 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)
)
```

### 3.2.2.4. Execute メソッド

CaoController 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)
)
```

Execute で指定できる拡張コマンド一覧の詳細は 3.3Execute コマンドを参照してください。使用例は拡張コマンドの詳細で記述しています。

### 3.2.3. CaoVariable クラス

#### 3.2.3.1. Value プロパティ

接続したスキャナからデータを取得/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。

### 3.2.4. OnMessage クラス

スキャナが RF タグ制御中に RF タグを検出した際に通知するデータをメッセージイベントとして発行します。

イベントの詳細は、3.5 イベント一覧を参照してください。

またスキャナの RF タグ制御状態に関しては 3.3.5 を参照してください。

### 3.3. Execute コマンド

#### 3.3.1. 概略

本プロバイダには設定変更コマンド、RF タグと通信可能状態にするタグ制御コマンド、トリガコマンドが実装されています。

プロバイダからスキャナに読み取り動作を実施させる場合、タグ制御コマンドを送信しコマンドを登録後、トリガコマンドによって作動させます。読み取った RF タグは OnMessage により取得できます。図 3-1 Execute コマンドによるタグ制御はプロバイダから Execute コマンドを使ってタグ制御をする一例になります。

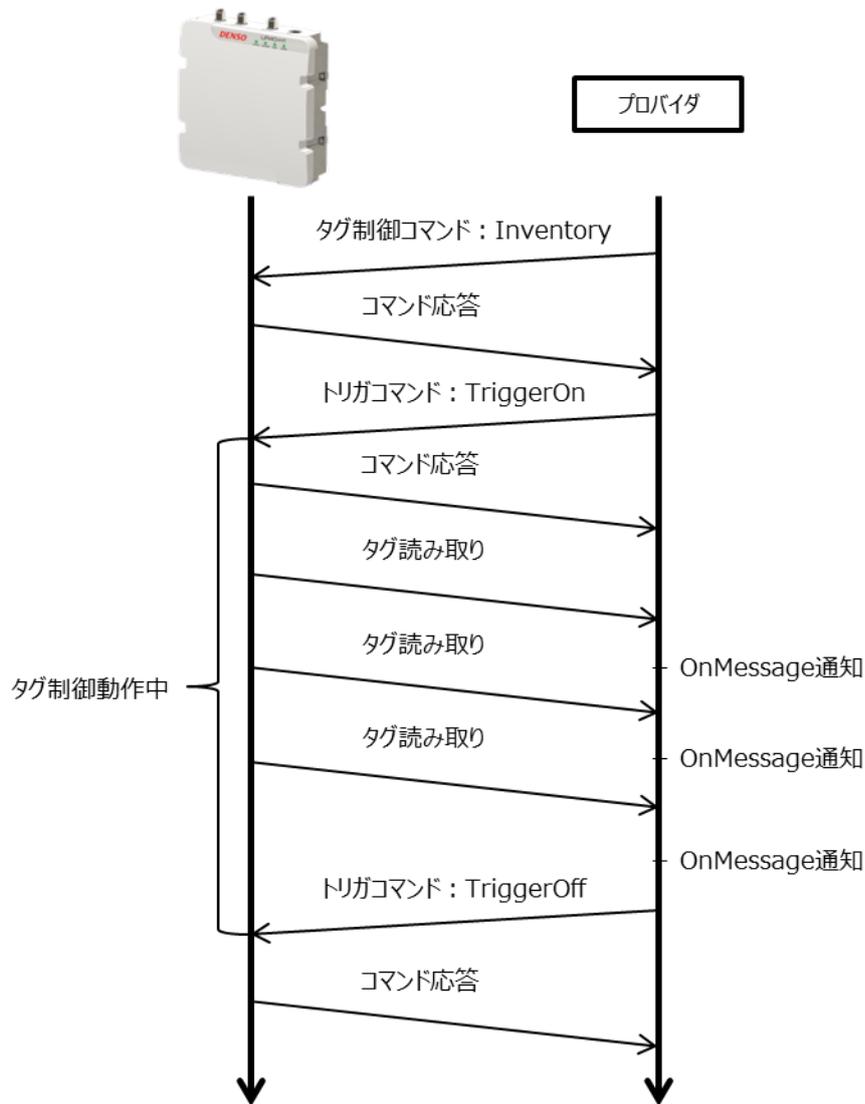


図 3-1 Execute コマンドによるタグ制御 (Inventory)

※トリガコマンド : TriggerOn→タグ制御コマンド : Inventory の順番でもタグ制御動作が開始されます。

### 3.3.2. Execute コマンド一覧

コマンド	種類	説明	参照
TriggerContinuous	設定変更	トリガモードを連続へ切り替えます。	P. 18
TriggerSingle	設定変更	トリガモードを単独へ切り替えます。	P. 18
Inventory	タグ制御	RF タグの UII を取得する Inventory コマンドを登録します。	P. 19
InventoryUII	タグ制御	RF タグの UII を指定して RF タグの UII を取得する Inventory コマンドを登録します。	P. 19
Read	タグ制御	RF タグのメモリデータを読み出す Read コマンドを登録します。	P. 20
ReadUII	タグ制御	RF タグの UII を指定して RF タグのメモリデータを読み出す Read コマンドを登録します。	P. 21
Write	タグ制御	RF タグのメモリデータに書き込みを行う Write コマンドを登録します。	P. 22
WriteUII	タグ制御	RF タグの UII を指定して RF タグのメモリデータに書き込みを行う Write コマンドを登録します。	P. 23
Lock	タグ制御	RF タグのロック状態の変更を行う Lock コマンドを登録します。	P. 24
LockUII	タグ制御	RF タグの UII を指定して RF タグのロック状態の変更を行う Lock コマンドを登録します。	P. 25
Kill	タグ制御	RF タグを使用不能にする Kill コマンドを登録します。	P. 26
KillUII	タグ制御	RF タグの UII を指定して RF タグを使用不能にする Kill コマンドを登録します。	P. 26
CommandRelease	タグ制御	登録したコマンドを解除します。	P. 27
TriggerOn	トリガ	登録したコマンドを作動させます。	P. 27
TriggerOff	トリガ	作動中のコマンドを停止させます。	P. 27
Reset	その他	スキャナを再起動します。	P. 28
Raw	その他	スキャナに任意のコマンドを送信します。	P. 28

### 3.3.3. コマンドによるタグ制御

RF タグの読み取りはタグ制御コマンドを実行した後、TriggerOn コマンドを実行してください。また、途中でコマンドを切り替える場合は CommandRelease コマンドを送信後、実行したいコマンドを送信してください。

例として Inventory コマンドにより RF タグ読み取りをする際のコマンド送信の手順を記載します。

※スキャナは「上位コントロールモード」で、トリガタイプが「コマンド」である前提です。

1. 電源投入から Inventory を実施する。

手順	操作
1	Inventory コマンドを送信
2	TriggerOn コマンドを送信

2. 1の状態から Read コマンドを実施する。

手順	操作
1	TriggerOff コマンドを送信
2	CommandRelease コマンドを送信
3	Read コマンドを送信
4	TriggerOn コマンドを送信

※手順 1 で TriggerOff コマンドを送信せず、手順 3 を実施すると Read コマンドを送信した段階で Read 操作が実行されます。

### 3.3.4. 設定変更コマンド

パラメータコマンドは、スキャナ本体に関するパラメータの設定取得をします。

タグ制御動作中、または TriggerOn コマンド後はエラーを返します。TriggerOff と CommandRelease コマンドを実行してから設定変更コマンドを実行してください。

#### 3.3.4.1. TriggerContinuous

トリガモードを連続モードへ切り替えます。引数からワンショット時間の設定ができます。

ワンショット時間で指定した時間読み取り動作を実施します。

ワンショット時間終了による動作終了後、再びタグ読み取りさせる場合は、TriggerOff コマンドを送信後、TriggerOn コマンドを送信してください。

項目	型説明	
引数	VT_UI1	ワンショット時間を指定します。 0:ワンショット無効。 1-255:ワンショット有。入力値×100msの時間になります。
戻り値	なし	

#### 使用例

```
' Connect
Call Connect
' Change Trigger mode
controller.Execute("TriggerContinuous", 10)
' Disconnect
Call Disconnect
```

#### 3.3.4.2. TriggerSingle

トリガモードを単独モードへ切り替えます。引数からワンショット時間の設定ができます。

ワンショット時間で指定した時間読み取り動作を実施します。単独モードでは1枚読み取るかワンショット時間経過で動作終了になります。

動作終了後、再びタグ読み取りさせる場合は、TriggerOff コマンドを送信後、TriggerOn コマンドを送信してください。

項目	型説明	
引数	VT_UI1	ワンショット時間を指定します。 0:ワンショット無効。 1-255:ワンショット有。入力値×100msの時間になります。
戻り値	なし	

#### 使用例

```
' Connect
Call Connect
' Change Trigger mode
```

```
Call controller.Execute("TriggerSingle", 10)
' Disconnect
Call Disconnect
```

### 3.3.5. RF タグ制御コマンド

RF タグ制御コマンドでは、コマンドを実行することでスキャナへコマンドを登録します。

登録後 TriggerOn コマンドを送信することで読み取りが実行されます。

スキャナが通信制御中に RF タグを検出した場合は、検出した RF タグの情報を通知するため情報が通知されるごとにメッセージイベントが発行されます。

各コマンドの詳細は下記を参照してください。

#### 3.3.5.1. Inventory

スキャナが RF タグを検出した際に検出した UII データを通知するようになります。

スキャナから UII データが通知された際は、Inventory メッセージイベント (3.5.2 参照) を発行します。

RF タグ制御を開始している状態では本コマンド実行するとエラーが返却されます。

項目	型説明
引数	なし
戻り値	なし

##### 使用例

```
' Connect
Call Connect
' Regist Inventory command
Call controller.Execute("Inventory")
```

#### 3.3.5.2. InventoryUII

スキャナが指定した UII を持つ RF タグを検出した際に検出した UII データを通知するようになります。

スキャナから UII データが通知された際は、Inventory メッセージイベント (3.5.2 参照) を発行します。

RF タグ制御を開始している状態では本コマンド実行するとエラーが返却されます。

項目	型説明
引数	VT_BSTR UII のデータを指定します。 桁数範囲 : 4 - 124 (UII データ長に応じて設定, 最大 62 バイト分の 16 進数文字列)
戻り値	なし

##### 使用例

```

' Connect
Call Connect
Dim UII As String
' UII
UII = "AAAA00001234"
' Regist InventoryUII command
Call controller.Execute("InventoryUII", UII)
    
```

### 3.3.5.3. Read

スキャナがRF タグを検出した際に指定した範囲のメモリデータ内容を通知するようになります。スキャナから読み出しデータが通知された際は、Read メッセージイベント(3.5.3 参照)を発行します。スキャナがRF タグ制御を開始している場合は、実行できません。

項目	型説明	
引数	VT_ARRAY   VT_VARIANT	
	1	VT_I4 バンクを指定 値範囲:0-3 0:Reserved Bank 1:UII Bank 2:TID Bank 3:User Bank
	2	VT_I4 読み出し開始位置を指定 値範囲:0-65534 <sup>2</sup>
	3	VT_I4 読み出しサイズを指定 値範囲:1-4096 <sup>2</sup>
	4	VT_BSTR アクセスパスワードを 16 進数文字列で指定 ("00000000"指定時は認証処理を省略) 値範囲: "00000000"-"FFFFFFFF"
戻り値	なし	

#### 使用例

```

' Connect
Call Connect
' Set Parameter
Dim readParam As Variant
readParam = Array(0, 4, 4, "00000000")
' Regist Read command
Call controller.Execute("Read", readParam)
    
```

<sup>2</sup> 実際の指定できる範囲は、お使いのRF タグおよびスキャナの仕様を確認してください。

### 3.3.5.4. ReadUII

スキャナが指定したUIIを持つRFタグを検出した際に指定した範囲のメモリデータ内容を通知するようになります。スキャナから読み出しデータが通知された際は、Readメッセージイベント（3.5.3参照）を発行します。スキャナがRFタグ制御を開始している場合は、実行できません。

項目	型説明		
引数	VT_ARRAY   VT_VARIANT		
	1	VT_I4	バンクを指定 値範囲:0-3 0:Reserved Bank 1:UII Bank 2:TID Bank 3:User Bank
	2	VT_I4	読み出し開始位置を指定 値範囲:0-65534 <sup>3</sup>
	3	VT_I4	読み出しサイズを指定 値範囲:1-4096 <sup>3</sup>
	4	VT_BSTR	アクセスパスワードを16進数文字列で指定("00000000"指定時は認証処理を省略) 値範囲:"00000000"-"FFFFFFFF"
5	VT_BSTR	UIIのデータを指定します。 桁数範囲:4 - 124 (UIIデータ長に応じて設定, 最大62バイト分の16進数文字列)	
戻り値	なし		

#### 使用例

```
' Connect
Call Connect
' Set Parameter
Dim readParam As Variant
readParam = Array(0, 4, 4, "00000000", "AAAA00001234")
' Regist ReadUII command
Call controller.Execute("ReadUII", readParam)
```

<sup>3</sup> 実際の指定できる範囲は、お使いのRFタグおよびスキャナの仕様を確認してください。

### 3.3.5.5. Write

スキャナが RF タグを検出した際に指定した範囲のメモリデータに指定した値を書き込みするようになります。データの書き込みが実行された際は、Write メッセージイベント（3.5.4 参照）を発行します。スキャナが RF タグ制御を開始している場合は、実行できません。

項目	型説明		
引数	VT_ARRAY   VT_VARIANT		
	1	VT_I4	バンクを指定 値範囲:0-3 0:Reserved Bank 1:UII Bank 2:TID Bank 3:User Bank
	2	VT_I4	書き込み開始位置を指定 値範囲:0-65534 <sup>4</sup>
	3	VT_BSTR	アクセスパスワードを 16 進数文字列で指定 (“00000000”指定時は認証処理を省略) 値範囲: “00000000”-“FFFFFFFF”
4	VT_BSTR	書き込みを行うデータを 16 進数文字列で指定 <sup>4</sup> 1 バイト (2 文字) 単位で指定してください。	
戻り値	なし		

#### 使用例

```

' Connect
Call Connect
' Set Parameter
Dim writeParam As Variant
writeParam = Array(1, 4, "00000000", "ABCDEF12")
' Regist Read command
Call controller.Execute("Write", writeParam)
    
```

<sup>4</sup> 実際の指定できる範囲は、お使いの RF タグおよびスキャナの仕様を確認してください。

### 3.3.5.6. WriteUII

スキャナが指定したUIIを持つRFタグを検出した際に指定した範囲のメモリデータに指定した値を書き込みするようになります。データの書き込みが実行された際は、Writeメッセージイベント（3.5.4参照）を発行します。スキャナがRFタグ制御を開始している場合は、実行できません。

項目	型説明	
引数	VT_ARRAY   VT_VARIANT	
	1	VT_I4 バンクを指定 値範囲:0-3 0:Reserved Bank 1:UII Bank 2:TID Bank 3:User Bank
	2	VT_I4 書き込み開始位置を指定 値範囲:0-65534 <sup>5</sup>
	3	VT_BSTR アクセスパスワードを16進数文字列で指定("00000000"指定時は認証処理を省略) 値範囲:"00000000"-"FFFFFFFF"
	4	VT_BSTR 書き込みを行うデータを16進数文字列で指定 <sup>5</sup> 1バイト(2文字)単位で指定してください。
5	VT_BSTR UIIのデータを指定します。 桁数範囲:4 - 124 (UIIデータ長に応じて設定, 最大62バイト分の16進数文字列)	
戻り値	なし	

#### 使用例

```
' Connect
Call Connect
' Set Parameter
Dim writeParam As Variant
writeParam = Array(1, 4, "00000000", "ABCDEF12", "AAAA00001234")
' Regist ReadUII command
Call controller.Execute("WriteUII", writeParam)
```

<sup>5</sup> 実際の指定できる範囲は、お使いのRFタグおよびスキャナの仕様を確認してください。

### 3.3.5.7. Lock

スキャナがRF タグを検出した際に指定した領域のロック状態を変更するようになります。ロック状態の変更が実行された際は、Lock メッセージイベント（3.5.5 参照）を発行します。スキャナがRF タグ制御を開始している場合は、実行できません。

項目	型説明		
引数	VT_ARRAY   VT_VARIANT		
	1	VT_UI1 または VT_BSTR	ロックする領域を以下の値の和で指定 VT_BSTR で指定する場合は 16 進数 2 桁の文字列で指定してください。 01h: キルパスワード 02h: アクセスパスワード 10h: UII Bank 20h: TID Bank 40h: User Bank
	2	VT_UI1 または VT_BSTR	ロックタイプを指定 VT_BSTR で指定する場合は 16 進数 2 桁の文字列で指定してください。 00h: アンロック 01h: ロック 10h: 永久アンロック 11h: 永久ロック
	3	VT_BSTR	アクセスパスワードを 16 進数文字列で指定 (“00000000”指定時は認証処理を省略) 値範囲: “00000000”-“FFFFFFFF”
戻り値	なし		

#### 使用例

```

' Connect
Call Connect
' Set Parameter
Dim lockParam As Variant
lockParam = Array(&H11, &H00, "ABCDEF0123")
' Regist Read command
Call controller.Execute("Lock", lockParam)
    
```

### 3.3.5.8. LockUII

スキャナが指定したUIIを持つRFタグを検出した際に指定した領域のロック状態を変更するようになります。ロック状態の変更が実行された際は、Lockメッセージイベント（3.5.5参照）を発行します。スキャナがRFタグ制御を開始している場合は、実行できません。

項目	型説明		
引数	VT_ARRAY   VT_VARIANT		
	1	VT_UI1 または VT_BSTR	ロックする領域を以下の値の和で指定 VT_BSTRで指定する場合は16進数2桁の文字列で指定してください 01h:キルパスワード 02h:アクセスパスワード 10h:UII Bank 20h:TID Bank 40h>User Bank
	2	VT_UI1 または VT_BSTR	ロックタイプを指定 VT_BSTRで指定する場合は16進数2桁の文字列で指定してください。 00h:アンロック 01h:ロック 10h:永久アンロック 11h:永久ロック
	3	VT_BSTR	アクセスパスワードを16進数文字列で指定("00000000"指定時は認証処理を省略) 値範囲:"00000000"-"FFFFFFFF"
4	VT_BSTR	UIIのデータを指定します。 桁数範囲:4 - 124 (UIIデータ長に応じて設定, 最大62バイト分の16進数文字列)	
戻り値	なし		

#### 使用例

```
' Connect
Call Connect
' Set Parameter
Dim lockParam As Variant
lockParam = Array(&H11, &H00, "ABCDEF12", "AAAA00001234")
' Regist ReadUII command
Call controller.Execute("LockUII", lockParam)
```

### 3.3.5.9. Kill

スキャナが RF タグを検出した際に RF タグを使用できない状態にします。RF タグが使用できない状態になった際は、Kill メッセージイベント（3.5.6 参照）を発行します。スキャナが RF タグ制御を開始している場合は、実行できません。

項目	型説明	
引数	VT_ARRAY   VT_VARIANT	
	1	VT_BSTR キルパスワードを 16 進数文字列で指定 値範囲：“00000000”-“FFFFFFFF”
戻り値	なし	

#### 使用例

```
' Connect
Call Connect
' Set Parameter
Dim killParam As Variant
killParam = Array("ABCDEF0123")
' Regist Read command
Call controller.Execute("Kill", killParam)
```

### 3.3.5.10. KillUII

スキャナが指定した UII を持つ RF タグを検出した際に RF タグを使用できない状態にします。RF タグが使用できない状態になった際は、Kill メッセージイベント（3.5.6 参照）を発行します。スキャナが RF タグ制御を開始している場合は、実行できません。

項目	型説明	
引数	VT_ARRAY   VT_VARIANT	
	1	VT_BSTR キルパスワードを 16 進数文字列で指定 値範囲：“00000000”-“FFFFFFFF”
	2	VT_BSTR UII のデータを指定します。 桁数範囲：4 - 124 (UII データ長に応じて設定，最大 62 バイト分の 16 進数文字列)
戻り値	なし	

#### 使用例

```
' Connect
Call Connect
' Set Parameter
Dim killParam As Variant
killParam = Array("ABCDEF12", "AAAA00001234")
' Regist ReadUII command
Call controller.Execute("KillUII", killParam)
```

### 3.3.5.11. CommandRelease

登録した RF タグ制御コマンドを解除します。新しいタグ制御コマンドを登録する前に送信してください。

項目	型説明
引数	なし
戻り値	なし

#### 使用例

```
' Connect
Call Connect
' Regist Inventory command
Call controller.Execute("Inventory")
' Unregist command
Call controller.Execute("CommandRelease")

' Regist Read command
Dim readParam As Variant
readParam = Array(0, 4, 4, "00000000")
Call controller.Execute("Read", readParam)
```

### 3.3.6. トリガコマンド

トリガコマンドではスキャナにタグ制御コマンドで登録したコマンドを実行させます。実行したタグ制御コマンドにより読み取ったタグは各コマンドに対応したメッセージイベントが発行されます。

#### 3.3.6.1. TriggerOn

登録されたコマンド動作を実行させます。TriggerOn コマンド送信後に再び TriggerOn コマンドを送信するとエラーが返ります。

項目	型説明
引数	なし
戻り値	なし

#### 使用例

```
' Connect
Call Connect
' Regist Inventory command
Call controller.Execute("Inventory")
' Start RF-tag control (Inventory)
Call controller.Execute("TriggerOn")
```

#### 3.3.6.2. TriggerOff

登録されたコマンド動作を停止させます。

項目	型説明
引数	なし
戻り値	なし

**使用例**

```
' Connect
Call Connect
' Regist Inventory command
Call controller.Execute("Inventory")
' Start RF-tag controll (Inventory)
Call controller.Execute("TriggerOn")
' Stop RF-tag controll (Inventory)
Call controller.Execute("TriggerOff")
```

### 3.3.7. その他のコマンド

#### 3.3.7.1. Reset

接続中のスキャナを再起動させます。再起動後の接続には CaoController 再接続が必要です。

項目	型説明
引数	なし
戻り値	なし

**使用例**

```
' Connect
Call Connect
' Reset scanner
Call controller.Execute("Reset")
' Connect
Call Disconnect
Call Connect
```

#### 3.3.7.2. Raw

スキャナに任意のコマンド文字列を送信します。

コマンドフォーマットはスキャナの取扱説明書を参照ください。

項目	型説明	
引数	VT_BSTR	任意のコマンド文字列
戻り値	VT_BSTR	コマンド応答

**使用例**

```
' Connect
Call Connect
' Send raw command
controller.Execute("Raw", "RFUW, 03, 0000, 00000000, 11223344")
```

### 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。

#### 3.4.1. CaoController クラス変数

変数名	取得データ型	説明	Value		参照
			get	put	
@MAKER_NAME	VT_BSTR	デバイスメーカー名を取得します。	○	-	P. 29
@VERSION	VT_BSTR	プロバイダバージョンを取得します。	○	-	P. 29
@DEVICE_VERSION	VT_BSTR	スキャナのファームウェアバージョンを取得します。	○	-	P. 30
@ID	VT_BSTR	スキャナの ID を取得します。	○	-	P. 30
@READ_CONNECTED	VT_BOOL	読み取りデータ受信ポートとスキャナとの接続状況を取得します。	○	-	P. 30

##### 3.4.1.1. @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

```

' Connect
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@MAKER_NAME")
' Get value
Dim strVal As String
strVal = var.value
    
```

##### 3.4.1.2. @VERSION

プロバイダバージョンの取得をします。

###### データ型

型説明	
VT_BSTR	プロバイダバージョンを取得します。

###### 使用例

```

' Connect
Call Connect
    
```

```
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@VERSION")
' Get value
Dim value As String
value = var.value
```

### 3.4.1.3. @DEVICE\_VERSION

スキャナのファームウェアバージョンの取得をします。

#### データ型

型説明	
VT_BSTR	スキャナのファームウェアバージョンを取得します。

#### 使用例

```
' Connect
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@DEVICE_VERSION")
' Get value
Dim value As String
value = var.value
```

### 3.4.1.4. @ID

スキャナのシリアル番号の取得をします。

#### データ型

型説明	
VT_BSTR	スキャナのファームウェアバージョンを取得します。

#### 使用例

```
' Connect
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@ID")
' Get value
Dim value As String
value = var.value
```

### 3.4.1.5. @READ\_CONNECTED

読み取りデータ受信ポートとスキャナとの接続状況を取得します。プロバイダ起動直後はスキャナ

との接続が完了しておらず読み取りデータが受信できない場合があります。@READ\_CONNECTED 変数が True になっていることを確認してからタグを読み取りください。

**データ型**

型説明	
VT_BOOL	True:接続中 False:切断中.

**使用例**

```

' Connect
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@READ_CONNECTED")
' Get value
Dim value As String
value = var.value
    
```

**3.5. イベント一覧**

スキャナが検出した RF タグの情報を通知します。メッセージは1つのタグに対して1回発行され、複数のタグが読み込まれた場合は読み込まれた回数分のメッセージを通知します。

メッセージ内容は RF タグ制御コマンドごとに異なり、詳細は下記を参照してください。

メッセージ番号	説明	参照
0	エラーコードを通知します。	P. 31
1	Inventory による RF タグ制御実施中に検出された RF タグの情報を通知します。	P. 32
2	Read による RF タグ制御中に検出された RF タグの情報と指定メモリ領域の読み込みデータを通知します。	P. 33
3	Write による RF タグ制御実施中に検出された RF タグの情報を通知します。	P. 34
4	Lock による RF タグ制御実施中に検出された RF タグの情報を通知します。	P. 35
5	Kill による RF タグ制御実施中に検出された RF タグの情報を通知します。	P. 36
99	その他のデータ種を通知します。	P. 37

**3.5.1. Error メッセージ**

タグ制御中のエラーコードおよび受信したデータを通知します。受信データの内容は実行中のタグ制御コマンドにより異なります。エラーコードの内容は「5. UR40 プロバイダエラーコード」を参照ください。

項目	説明		
メッセージ番号	0		
メッセージ内容	VT_BSTR   VT_ARRAY		
	0	VT_BSTR	エラーコード。(HRESULT 形式)
	1 - n	VT_BSTR	スキャナから受信したデータ(下記補足を参照)

**【補足】**

スキャナの「失敗通知データ」→「出力フォーマット設定の適用」の設定により出力内容が異なります。

- 出力フォーマット設定の適用：しない  
失敗通知データキャラクタ (ERROR) が出力されます。
- 出力フォーマット設定の適用：する  
シリアルナンバー, 日付, 失敗通知データキャラクタ (ERROR) が出力されます。

**3.5.2. Inventory メッセージ**

スキャナが Inventory による RF タグ制御中(3.3.5.1 参照)に検出した RF タグの情報を通知します。

項目	説明		
メッセージ番号	1		
メッセージ内容	VT_BSTR   VT_ARRAY		
	1	VT_BSTR	スキャナ ID
	2	VT_BSTR	時刻 (YYYYMMDDhhmmss 形式)
	3	VT_BSTR	読み取り条件番号
	4	VT_BSTR	通信識別子 (RFUGI)
	5	VT_BSTR	UII
	6	VT_BSTR	PC
	7	VT_BSTR	RSSI
	8	VT_BSTR	アンテナ番号
	9	VT_BSTR	偏波

### 3.5.3. Read メッセージ

スキャナが Read による RF タグ制御中 (3.3.5.3 参照)に検出した RF タグの情報を通知します。RF タグ通信結果が正常終了の場合は指定したメモリ領域も一緒に通知します。

項目	説明												
メッセージ番号	2												
メッセージ内容	VT_BSTR   VT_ARRAY												
	1	VT_BSTR	スキャナ ID										
	2	VT_BSTR	時刻 (YYYYMMDDhhmmss 形式)										
	3	VT_BSTR	読み取り条件番号										
	4	VT_BSTR	通信識別子 (RFUGR)										
	5	VT_BSTR	UII										
	6	VT_BSTR	PC										
	7	VT_BSTR	RSSI										
	8	VT_BSTR	アンテナ番号										
	9	VT_BSTR	偏波										
	10	VT_BSTR	エラーコード.										
			<table border="1"> <thead> <tr> <th>エラーコード</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>正常終了</td> </tr> <tr> <td>0203</td> <td>メモリ範囲外へアクセスしました。 指定のアドレス, サイズの見直し, および RF タグの仕様をご確認ください。</td> </tr> <tr> <td>0204</td> <td>ロックされたメモリへのアクセスです。 アクセスパスワードを指定してアクセスするか, 事前にロック状態の解除を実施してください。</td> </tr> <tr> <td>020F</td> <td>RF タグから規定外のエラーが返却されました。 RF タグの仕様をご確認ください。</td> </tr> </tbody> </table>	エラーコード	内容	0000	正常終了	0203	メモリ範囲外へアクセスしました。 指定のアドレス, サイズの見直し, および RF タグの仕様をご確認ください。	0204	ロックされたメモリへのアクセスです。 アクセスパスワードを指定してアクセスするか, 事前にロック状態の解除を実施してください。	020F	RF タグから規定外のエラーが返却されました。 RF タグの仕様をご確認ください。
	エラーコード	内容											
0000	正常終了												
0203	メモリ範囲外へアクセスしました。 指定のアドレス, サイズの見直し, および RF タグの仕様をご確認ください。												
0204	ロックされたメモリへのアクセスです。 アクセスパスワードを指定してアクセスするか, 事前にロック状態の解除を実施してください。												
020F	RF タグから規定外のエラーが返却されました。 RF タグの仕様をご確認ください。												
11	VT_BSTR	読み取りデータ. エラーコードが正常終了以外の場合は空文字が格納されます。											

### 3.5.4. Write メッセージ

スキャナが Write による RF タグ制御中 (3.3.5.5 参照) に検出した RF タグの情報を通知します。

項目	説明		
メッセージ番号	3		
メッセージ内容	VT_BSTR   VT_ARRAY		
	1	VT_BSTR スキャナ ID	
	2	VT_BSTR 時刻 (YYYYMMDDhhmmss 形式)	
	3	VT_BSTR 読み取り条件番号	
	4	VT_BSTR 通信識別子 (RFUGR)	
	5	VT_BSTR UII	
	6	VT_BSTR PC	
	7	VT_BSTR RSSI	
	8	VT_BSTR アンテナ番号	
	9	VT_BSTR 偏波	
	10	VT_BSTR エラーコード.	
		<b>エラーコード</b>	<b>内容</b>
		0000	正常終了
		0203	メモリ範囲外へアクセスしました。 指定のアドレス, サイズの見直し, および RF タグの仕様をご確認ください。
	0204	ロックされたメモリへのアクセスです。 アクセスパスワードを指定してアクセスするか, 事前にロック状態の解除を実施してください。	
	020F	RF タグから規定外のエラーが返却されました。 RF タグの仕様をご確認ください。	

### 3.5.5. Lock メッセージ

スキャナが Lock による RF タグ制御中 (3.3.5.7 参照) に検出した RF タグの情報を通知します。

項目	説明												
メッセージ番号	4												
メッセージ内容	VT_BSTR   VT_ARRAY												
	1	VT_BSTR	スキャナ ID										
	2	VT_BSTR	時刻 (YYYYMMDDhhmmss 形式)										
	3	VT_BSTR	読み取り条件番号										
	4	VT_BSTR	通信識別子 (RFUGR)										
	5	VT_BSTR	UII										
	6	VT_BSTR	PC										
	7	VT_BSTR	RSSI										
	8	VT_BSTR	アンテナ番号										
	9	VT_BSTR	偏波										
	10	VT_BSTR	エラーコード.										
			<table border="1"> <thead> <tr> <th>エラーコード</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>正常終了</td> </tr> <tr> <td>0203</td> <td>メモリ範囲外へアクセスしました。 指定のアドレス, サイズの見直し, および RF タグの仕様をご確認ください。</td> </tr> <tr> <td>0204</td> <td>ロックされたメモリへのアクセスです。 アクセスパスワードを指定してアクセスするか, 事前にロック状態の解除を実施してください。</td> </tr> <tr> <td>020F</td> <td>RF タグから規定外のエラーが返却されました。 RF タグの仕様をご確認ください。</td> </tr> </tbody> </table>	エラーコード	内容	0000	正常終了	0203	メモリ範囲外へアクセスしました。 指定のアドレス, サイズの見直し, および RF タグの仕様をご確認ください。	0204	ロックされたメモリへのアクセスです。 アクセスパスワードを指定してアクセスするか, 事前にロック状態の解除を実施してください。	020F	RF タグから規定外のエラーが返却されました。 RF タグの仕様をご確認ください。
	エラーコード	内容											
	0000	正常終了											
0203	メモリ範囲外へアクセスしました。 指定のアドレス, サイズの見直し, および RF タグの仕様をご確認ください。												
0204	ロックされたメモリへのアクセスです。 アクセスパスワードを指定してアクセスするか, 事前にロック状態の解除を実施してください。												
020F	RF タグから規定外のエラーが返却されました。 RF タグの仕様をご確認ください。												

### 3.5.6. Kill メッセージ

スキャナが Kill による RF タグ制御中 (3.3.5.9 参照) に検出した RF タグの情報を通知します。

項目	説明											
メッセージ番号	4											
メッセージ内容	VT_BSTR   VT_ARRAY											
	1	VT_BSTR スキャナ ID										
	2	VT_BSTR 時刻 (YYYYMMDDhhmmss 形式)										
	3	VT_BSTR 読み取り条件番号										
	4	VT_BSTR 通信識別子 (RFUGR)										
	5	VT_BSTR UII										
	6	VT_BSTR PC										
	7	VT_BSTR RSSI										
	8	VT_BSTR アンテナ番号										
	9	VT_BSTR 偏波										
10	VT_BSTR	エラーコード.										
		<table border="1"> <thead> <tr> <th>エラーコード</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>正常終了</td> </tr> <tr> <td>0203</td> <td>メモリ範囲外へアクセスしました。 指定のアドレス, サイズの見直し, および RF タグの仕様をご確認ください。</td> </tr> <tr> <td>0204</td> <td>ロックされたメモリへのアクセスです。 アクセスパスワードを指定してアクセスするか, 事前にロック状態の解除を実施してください。</td> </tr> <tr> <td>020F</td> <td>RF タグから規定外のエラーが返却されました。 RF タグの仕様をご確認ください。</td> </tr> </tbody> </table>	エラーコード	内容	0000	正常終了	0203	メモリ範囲外へアクセスしました。 指定のアドレス, サイズの見直し, および RF タグの仕様をご確認ください。	0204	ロックされたメモリへのアクセスです。 アクセスパスワードを指定してアクセスするか, 事前にロック状態の解除を実施してください。	020F	RF タグから規定外のエラーが返却されました。 RF タグの仕様をご確認ください。
	エラーコード	内容										
	0000	正常終了										
	0203	メモリ範囲外へアクセスしました。 指定のアドレス, サイズの見直し, および RF タグの仕様をご確認ください。										
0204	ロックされたメモリへのアクセスです。 アクセスパスワードを指定してアクセスするか, 事前にロック状態の解除を実施してください。											
020F	RF タグから規定外のエラーが返却されました。 RF タグの仕様をご確認ください。											

### 3.5.7. その他メッセージ

スキャナから Inventory, Read 以外の読み取りデータの内容を通知します。

項目	説明	
メッセージ番号	99	
メッセージ内容	VT_BSTR   VT_ARRAY	
	1 - n	VT_BSTR スキャナから通知された要素数 n 分だけ配列を確保し、各要素は VT_BSTR 型で格納されます。

## 4. UR40 プロバイダによるプログラミング

UR40 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、もしくは、CaoVariable オブジェクトを生成することで、スキャナの情報にアクセスすることができます。

### 4.1. スキャナに接続し RF タグの UII データを読み込むサンプルプログラミング

ここでは例としてスキャナのパラメータを設定しフラッシュメモリに保存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、4.1.1 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	ETH
	接続先 IP は 192.168.0.10、読み取りデータ転送先ポートを 50010。その他項目は省略。
処理内容	スキャナを Inventory モードで通信制御を開始する。
	スキャナから UII データの読み込みが走るとメッセージを表示する。
	メッセージ表示後切断する。

#### 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。

サンプルは VB.Net で記載しています。

Sample	SampleInventory.vb
	<pre> ' Object Dim engine As CaoEngine Dim workspace As CaoWorkspace Dim WithEvents controller As CaoController Dim MessageBuf As Object  ' Connect Private Sub Connect()     ' Create CaoEngine     engine = New CaoEngine     ' Create CaoWorkspace     workspace = engine.AddWorkspace("Newwrks", "")     ' Create CaoController     controller = workspace.AddController("SampleController", _ </pre>

```
        "CaoProv. DENS0. UR40", _
        ""
        ' _
        "conn=eth:192.168.0.10,ReadPort=50010")
    ' Regist OnMessage handler
    AddHandler controller.OnMessage, AddressOf controller_OnMessage
End Sub

' Disconnect
Private Sub Disconnect()
    ' Send TriggerOff
    If controller IsNot Nothing Then
        Call controller.Execute("TriggerOff")
    End If

    ' Delete OnMessage handler
    RemoveHandler controller.OnMessage, AddressOf controller_OnMessage
    ' Delete CaoController from CaoWorkspace
    Call workspace.Controllers.Remove(controller.Index)
    ' Delete CaoController
    System.Runtime.InteropServices.Marshal.ReleaseComObject(controller)
    controller = Nothing
    ' Delete CaoWorkspace from CaoEngine
    Call engine.Workspaces.Remove(workspace.Index)
    ' Delete CaoWorkspace
    System.Runtime.InteropServices.Marshal.ReleaseComObject(workspace)
    workspace = Nothing
    ' Delete CaoEngine
    System.Runtime.InteropServices.Marshal.ReleaseComObject(engine)
    engine = Nothing
End Sub

' Inventory
Private Sub StartInventory()
    ' Connect
    Call Connect()
    ' Stop current command
    controller.Execute("TriggerOff")
    ' Unregist current command
    controller.Execute("CommandRelease")

    ' Regist Inventory command
    controller.Execute("Inventory")
    ' Start Inventory
    controller.Execute("TriggerOn")

End Sub

' Message event from provider.
```

```

Private Sub controller_OnMessage(ByVal pICaoMess As CAOLib.CaoMessage)
    MessageBuf = pICaoMess.Value
    ' UII data
    Dim uiiData As String
    uiiData = MessageBuf(4)

    ' PC
    Dim pc As String
    pc = MessageBuf(5)

    ' RSSI
    Dim rssi As String
    rssi = MessageBuf(6)

    ' Antenna
    Dim antenna As String
    antenna = MessageBuf(7)

    ' Display UII data
    MsgBox(uiiData)

End Sub

```

#### 4.1.1.1. 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。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 WithEvents を変数名の前につけることで CaoController から OnMessage イベントを受信することができるようになります。

```

Dim engine As CaoEngine           ' Variable CaoEngine object
Dim workspace As CaoWorkspace    ' Variable CaoWorkspace object
Dim WithEvents controller As CaoController ' Variable CaoController object

```

- (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。

```

' Create CaoEngine object
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。

```
' Create CaoWorkspace object
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。UR40プロバイダでは、スキャナと接続した際にあらかじめレスポンスフォーマッ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。

```
' Create CaoController object
Set controller = workspace.AddController("SampleController", _
    "CaoProv. DENSO. UR40", _
    "", _
    "conn=eth:192.168.0.10,ReadPort=50010")
```

#### 4.1.1.2. スキャナを UII データの取得モードで実行

スキャナを UII 取得モード (Inventory) で開始します。

```
' Register Inventory command
controller.Execute("Inventory")
' Start Inventory
controller.Execute("TriggerOn")
```

#### 4.1.1.3. タグ読み込み時のメッセージを受信

スキャナからタグの情報を受信したら OnMessage イベントが発行されます。OnMessage イベントを受信したらメッセージ内容から UII データを MsgBox で表示します。

```
Private Sub controller_OnMessage(ByVal pICaoMess As CAOLib.CaoMessage)
    MessageBuf = pICaoMess.Value
    ' Display UII data
    MsgBox MessageBuf(4)
End Sub
```

#### 4.1.1.4. 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。

```
' Delete OnMessage handler
RemoveHandler controller.OnMessage, AddressOf controller_OnMessage
' Delete CaoController from CaoWorkspace
Call workspace.Controllers.Remove(controller.Index)
' Delete CaoController
System.Runtime.InteropServices.Marshal.ReleaseComObject(controller)
controller = Nothing
' Delete CaoWorkspace from CaoEngine
Call engine.Workspaces.Remove(workspace.Index)
```

---

' Delete CaoWorkspace

System.Runtime.InteropServices.Marshal.ReleaseComObject(workspace)

workspace = Nothing

' Delete CaoEngine

System.Runtime.InteropServices.Marshal.ReleaseComObject(engine)

engine = Nothing

---

## 5. UR40 プロバイダエラーコード

本プロバイダには、0x8010\*\*\*\*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

エラー名	エラー番号	説明
E_INVALID_PACKET	0x80100001	読み取りデータ、コマンド応答の packets が壊れています。接続やスキャナの設定を確認してください。
E_OPT_NONCONN	0x80100002	Conn オプションで eth を指定しなかった、またはオプション設定が仕様と異なります。AddController オプションを確認してください。
E_NOCLIENT	0x80100003	読み取りデータ転送先ポートがスキャナと繋がっていません。接続を確認してください。
E_MISMATCH_FORMAT	0x80100004	デバイスのフォーマット設定がプロバイダ設定と不一致になっています。デバイスのフォーマットを「2.1スキャナとクライアントPGとの接続」を参照し設定してください。
E_UNREAD_TAG	0x80100005	タグ制御動作中にタグが読めなかった。 ※プロバイダでこのエラーを検出するには、スキャナの設定で「失敗通知データを出力する」を有効にし、失敗通知データキャラクタを「ERROR」に設定しておく必要があります。
デバイスからのエラー	0x8010A300	動作モードエラー
	0x8010A301	実行条件エラー
	0x8010A303	フォーマットエラー
	0x8010A304	引数範囲外エラー
	0x8010A400	実行エラー
	0x8010AA01	トリガモードエラー
	0x8010****	想定外の異常

## 付録A. 通信プロトコルコマンド対応表

CaoController::Execute

コマンド	通信コマンド
TriggerContinuous	RFUU7
TriggerSingle	RFUU6
Inventory	RFUI
InventoryUII	RFUIU
Read	RFUR
ReadUII	RFURU
Write	RFUW
WriteUII	RFUWU
Lock	RFUL
LockUII	RFULU
Kill	RFUK
KillUII	RFUKU
CommandRelease	RFUS
TriggerOn	TRGON
TriggerOff	TRGOFF
Reset	RESET

## 付録B. 通信可能な RF タグ

スキャナで通信可能な RF タグは ISO/IEC 18000-63 (GS1 EPC Gen2) 準拠の RF タグです。

RF タグのメモリの有無やサイズなどの詳細情報については、ご使用になる RF タグの説明書を参照いただくか、タグメーカーにお問い合わせください。

### 付録B.1. RF タグのメモリ構造

バンク領域	開始アドレス [byte]	サイズ [byte]	内容	説明
User Bank	0	n <sup>6</sup>	ユーザメモリ	ユーザがデータを保存できる領域
TID Bank	0	n <sup>6</sup>	TID	RF タグの製造者識別子やシリアル、 拡張方法等（タグメーカーが設定）
UIIBank	4	n <sup>7</sup>	UII	RF タグ固有の識別子
	2	2	PC (StoredPC)	プロトコル制御情報 ● 10h-14h の 5bit で UII 長を規定
	0	2	GRC-16	Inventory 通信時のチェックデジット (ユーザ書き換え不可)
Reserved Bank	8	n <sup>6</sup>	規定なし	-
	4	4	アクセスパスワード	Read, Write, Lock 処理のアクセス認証に使用
	0	4	キルパスワード	Kill 処理の認証に使用

<sup>6</sup> RF タグの仕様により規定されます。

<sup>7</sup> PC の先頭 5bit (bit アドレス: 10h-14h) により決定されます。

## 付録C. スキャナの設定

2.1 スキャナとクライアント PC との接続 で設定するスキャナの設定について、Scanner Setting F を使って設定する例を示します。

- RFID

分類		設定項目
RFID	RF タグ通信動作モード	上位コントロールモード

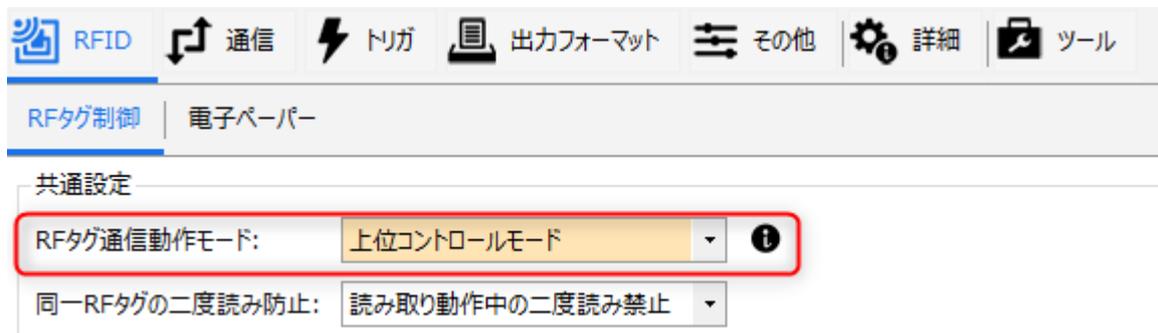


図 5-1 RFID の設定

- 通信

分類		設定項目		
通信	読み取りデータ送信インターフェース		Ethernet	
	LAN	IP アドレス設定	IP アドレス	DHCP 以外の任意のアドレス
			サブネットマスク	任意
			デフォルトゲートウェイ	任意
		コマンド制御	プロトコル	TCP Server
	読み取りデータ転送先	プロトコル	TCP Client	

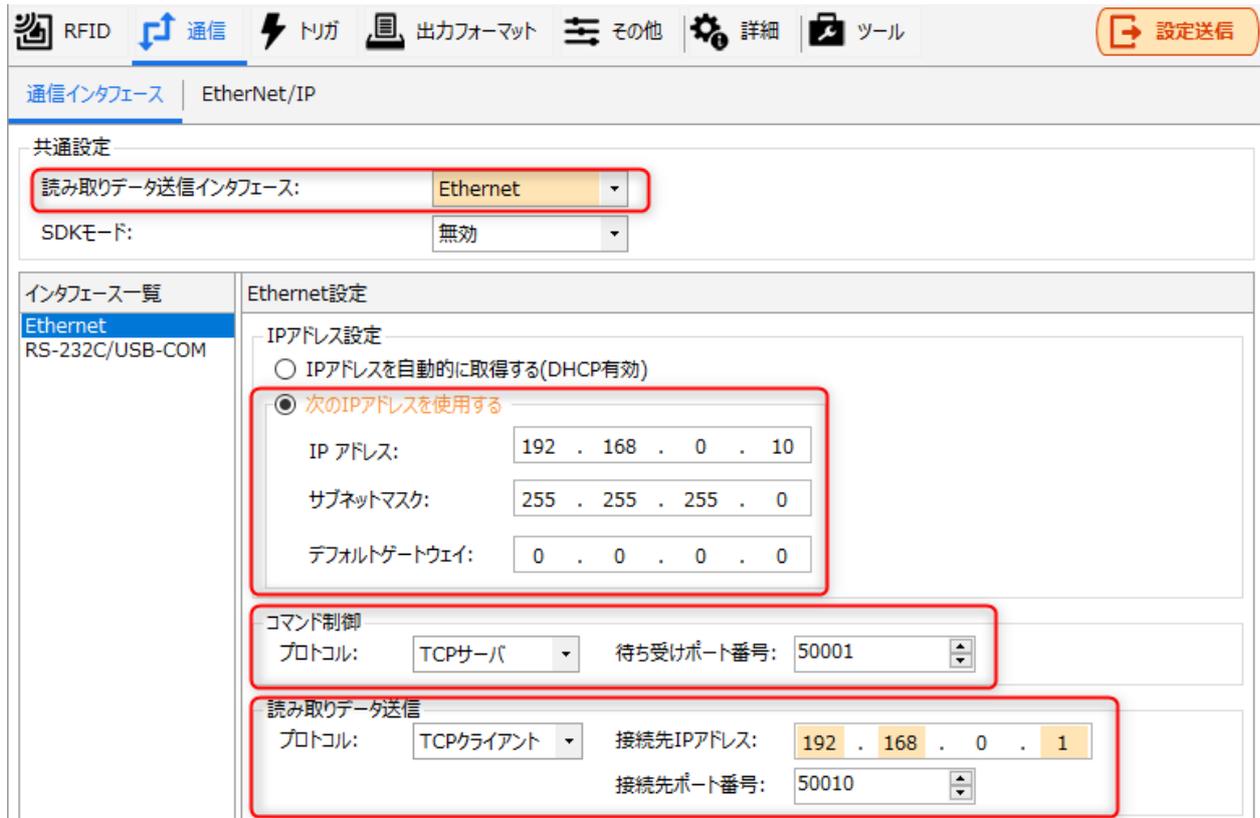


図 5-2 通信の設定

接続先 IP アドレスはプロバイダが起動している PC の IP アドレスを指定してください。スキャナの IP アドレスは PC と接続できる IP アドレスに適切に設定してください。

● トリガ

分類		設定項目	
トリガ	トリガタイプ	コマンド	
	トリガコマンド設定	トリガコマンドレスポンス	許可
		トリガ ON コマンド	TRGON
		トリガ OFF コマンド	TRGOFF
	失敗通知データ	失敗通知データを出力する	有効
失敗通知データキャラクタ		ERROR	

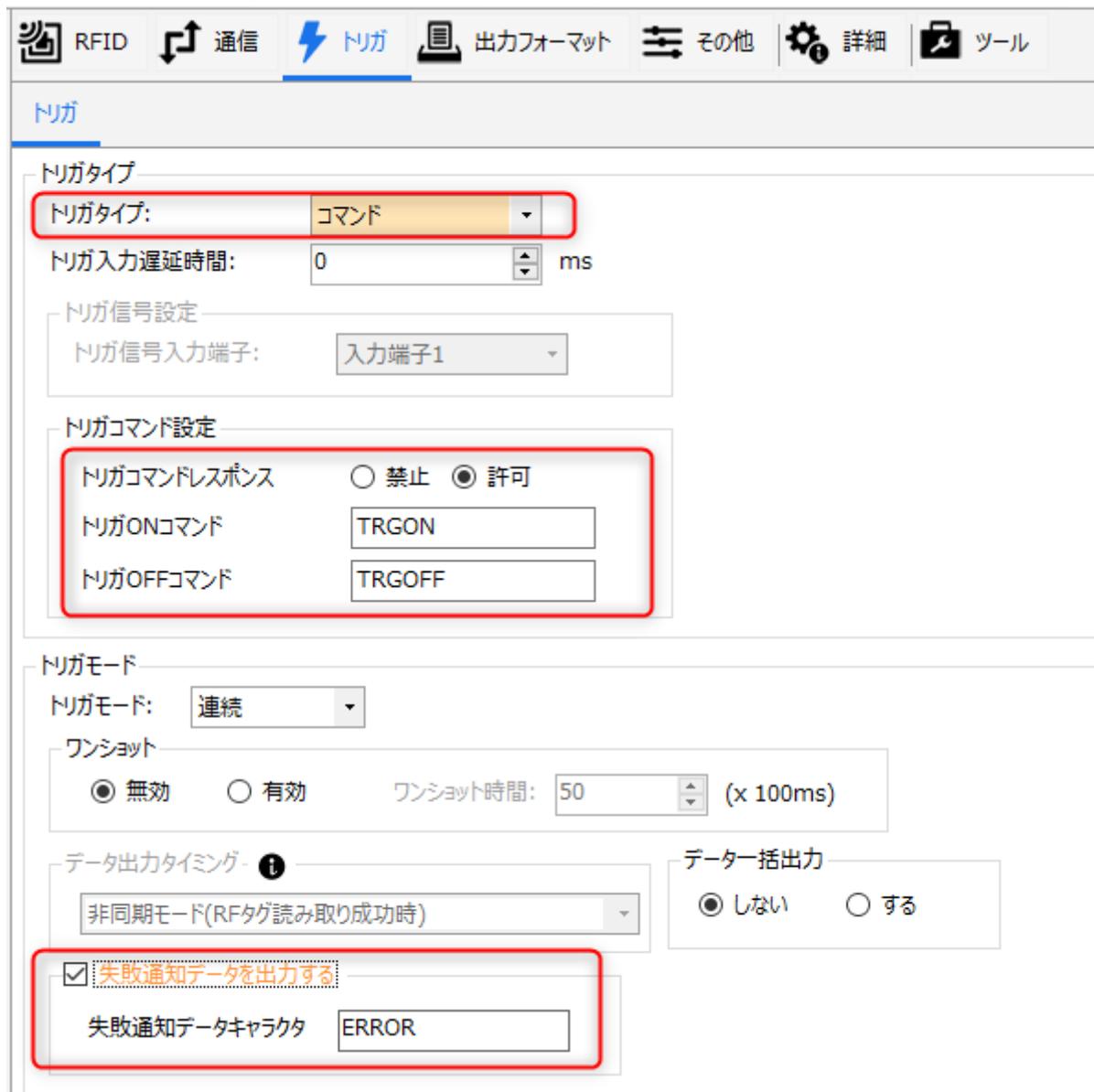


図 5-3 トリガの設定

● 出力フォーマット

分類		設定項目
出力フォーマット	セパレータ	カンマ
	スキャナ ID	許可
	時刻	許可
	任意文字列 1	データ無し
	読み取り条件番号	許可
	通信識別子	許可
	レスポンス方式	ASCII レスポンス

PC	許可
RSSI	許可
アンテナ番号	許可
偏波	許可
任意文字列 2	データ無し

The screenshot shows the '出力フォーマット' (Output Format) settings screen. The interface includes a top navigation bar with icons for RFID, 通信 (Communication), トリガ (Trigger), 出力フォーマット (Output Format), その他 (Others), 詳細 (Details), and ツール (Tools). The '出力フォーマット' section is active and contains the following settings:

- 共通設定 (Common Settings):**
  - セパレータ:  なし  カンマ  ユーザ指定 <設定なし>
  - デリミタ:  なし  カンマ  ユーザ指定 <設定なし>
- 読み取りデータ出力フォーマット (Data Output Format):**
  - スキャナーID:  無効  有効
  - 時刻:  無効  有効
  - 任意文字列1: [データ無し] 入力...
  - 読み取り条件番号:  無効  有効
  - 通信識別子:  無効  有効
- RFタグ通信データ部 (RF Tag Communication Data Section):**
  - レスポンス方式: ASCIIレスポンス
  - PC:  無効  有効
  - RSSI:  無効  有効
  - アンテナ番号:  無効  有効
  - 偏波:  無効  有効
- 任意文字列2:** [データ無し] 入力...
- BCC:**  無効  有効

図 5-4 出力フォーマットの設定