

# UTR-SDKV1 関数一覧

<対象 SDK バージョン : 1.20>

発行日 2026年2月3日  
Ver. 1.21

**タカヤ株式会社**

マニュアル番号 : TDR-MNL-UTR-SDKV1-121

---

---

# はじめに

このたびは、弊社製品「UTR-SDKV1シリーズ」をご利用いただき、誠にありがとうございます。

UTR-SDKV1 シリーズは、UTR シリーズ リーダライタを制御するためのソフトウェア開発キットです。

本書は、ソフトウェア開発キットの備える各種関数（メソッド）及びプロパティについて記載していません。

UTR シリーズは、国際標準規格 ISO/IEC18000-63 及び EPCglobal Class1 Generation2 に対応した製品です。それ以外の規格の RF タグ、IC カードには対応しておりませんのでご注意ください。

本書に記載のメソッドは、基本的にはリーダーライタに搭載されているコマンドを関数化したものです。必要に応じて、対象となるリーダーライタの通信プロトコル説明書もご参照ください。

各種リーダーライタ製品の取扱説明書、ユーティリティソフト、通信プロトコル説明書は以下の URL より最新版をダウンロードすることができます。

<https://www.product.takaya.co.jp/rfid/download.html>

## ご注意

- ・改良のため、お断りなく仕様変更する可能性がありますのであらかじめ御了承ください。
- ・本書の文章の一部あるいは全部を、無断でコピーしないでください。
- ・本書に記載した会社名・商品名などは、各社の商標または登録商標になります。

---

---

# UTR-SDKV1 バージョンアップ履歴

2023/05/30 ver1.2.0

バグ修正

- ・メソッド `Open(int portNo, RFID_BaudRate baudRate)` で通信速度が指定通り設定されない問題を修正
- ・ソケット接続中にまれに例外エラーが発生する問題を修正

バグ修正(サンプルプログラム)

- ・RSSI フィルタの有効/無効設定ができない問題を修正

列挙体追加

- ・ `UHF_ThroughCmdType`

メソッド引数用クラス追加

- ・ `UHF_ThroughCmdOption`

メソッド追加

- ・ `UHF_ThroughCmd`
- ・ `UHF_GetHandle`

2019/11/15 ver1.1.0

`SetTransmitSignal/UHF_BlockErase/UHF_BlockWrite2/UHF_Encode` 関数の追加

`SetTransmitSignal/UHF_BlockErase/UHF_BlockWrite2/UHF_Encode` サンプルソースの追加

2019/02/07 ver1.0.0

新規作成

---

---

# ソフトウェア使用許諾契約

この契約書はタカヤ株式会社（以下、「当社」といいます）が提供するソフトウェア及びその関連資料（以下「本ソフトウェア」といいます）の使用権を許諾する条件を定めたものです。

お客様が本ソフトウェアをダウンロード、インストール、または使用した場合、本条件に同意したものとみなされます。

## 1. 使用許諾

お客様は、本ソフトウェアに対応する当社製品を利用する目的で本ソフトウェアを使用することができます。

## 2. 利用成果物

本ソフトウェアを正当な目的に従って使用した結果として作成されたプログラム、データ、その他の成果物の著作権は、お客様に帰属します。

本ソフトウェアを使用して開発されたアプリケーションの配布および販売について、当社は制限を設けません。

## 3. 著作権

本ソフトウェア及びその複製物の著作権は当社または当社が認めた者が有するものであり、日本国著作権法及び国際条約によって保護されています。

本使用許諾契約に基づき、お客様が本ソフトウェアを複製する場合は、本ソフトウェアに付されていたものと同じの著作権表示がなされることを要します。

## 4. 禁止事項

お客様は、以下の行為を行ってはなりません。

- (a) 本ソフトウェアの全部または一部について、リバースエンジニアリング、逆コンパイル、逆アセンブル、その他の解析を行うこと
- (b) 本ソフトウェアを改変すること
- (c) 本ソフトウェアを有償で第三者に販売または提供すること

## 5. 再配布許可

お客様は、本ソフトウェアを無償で第三者に譲渡、貸与、または再配布することができます。ただし、その場合、本使用許諾契約を改変せずに引き継ぐものとします。

## 6. 無保証

当社は、本ソフトウェアについて、お客様の特定の目的に適合すること、有用であること、瑕疵がないこと、その他本ソフトウェアに関して、いかなる保証も行いません。

## 7. 免責

当社は、いかなる場合においても、本ソフトウェアの使用または使用不能から生ずるいかなる損害（事業利益の損害、事業の中断、事業情報の損失、またはその他金銭的損害）に関して、一切責任を負いません。

## 8. 契約の解除

お客様が本使用許諾契約に違反した場合、当社は本使用許諾契約を解除することができます。その場合、お客様は本ソフトウェアを一切使用しないものとします。

---

---

9. サポート

本ソフトウェアに関するお問い合わせは、以下のお問い合わせフォームよりお願い致します。  
また、同梱のドキュメントファイルを必ずお読み下さい。

お問い合わせフォームURL：

<https://www.product.takaya.co.jp/rfid/contact/sdk-support.html>

10. 保証範囲

本ソフトウェアのインストールはお客様の責任において行って頂きます。  
本ソフトウェアは、予告せず改良、変更することがあります。

11. 著作権者

本ソフトウェアの著作権は、タカヤ株式会社に帰属します。

---

---

# 目次

<b>第 1 章</b>	<b>SDK の仕組み</b> .....	<b>8</b>
1.1	シリアルインターフェースと LAN インターフェース .....	9
1.2	ResponseRFID イベント.....	10
1.3	自動読み取りモードでの応答の受信.....	12
1.4	SDK の内部処理とタイムアウトについて .....	13
<b>第 2 章</b>	<b>通信路の確保と開放</b> .....	<b>16</b>
2.1	Open メソッド.....	17
2.2	Close メソッド.....	18
2.3	Connect メソッド .....	19
2.4	Disconnect メソッド.....	20
2.5	Dispose メソッド .....	21
<b>第 3 章</b>	<b>リーダライタの制御</b> .....	<b>22</b>
3.1	GetError メソッド.....	23
3.2	GetROMVersion メソッド.....	24
3.3	GetCHIPVersion メソッド .....	25
3.4	SetTransmitSignal メソッド.....	26
3.5	CallLEDBuzzer メソッド.....	27
3.6	Restart メソッド.....	29
3.7	CallBuzzer メソッド.....	30
3.8	InitFLASH メソッド.....	31
3.9	UHF_GetHandle メソッド.....	32
<b>第 4 章</b>	<b>リーダライタ動作モードの設定</b> .....	<b>33</b>
4.1	GetActionMode メソッド.....	34
4.2	SetActionMode メソッド.....	35
<b>第 5 章</b>	<b>リーダライタ FLASH の設定</b> .....	<b>36</b>
5.1	ReadFLASH メソッド.....	37
5.2	WriteFLASH メソッド.....	38
<b>第 6 章</b>	<b>リーダライタ設定</b> .....	<b>39</b>
6.1	UHF_GetInventoryParam メソッド .....	40
6.2	UHF_GetSelectParam メソッド.....	42
6.3	UHF_GetExpandSelectParam メソッド.....	43
6.4	UHF_GetAntennaParam メソッド.....	44
6.5	UHF_GetTxPowerParam メソッド.....	45
6.6	UHF_GetFrequencyParam メソッド .....	46
6.7	UHF_GetRFTAGComParam メソッド .....	47
6.8	UHF_GetEPCParam メソッド.....	48
6.9	UHF_SetInventoryParam メソッド .....	49
6.10	UHF_SetSelectParam メソッド.....	50
6.11	UHF_SetExpandSelectParam メソッド.....	51
6.12	UHF_SetAntennaParam メソッド .....	52
6.13	UHF_SetTxPowerParam メソッド .....	53
6.14	UHF_SetFrequencyParam メソッド.....	54
6.15	UHF_SetRFTAGComParam メソッド .....	55
6.16	UHF_SetEPCParam メソッド.....	56

6.17	UHF_SetAccessPassword メソッド	57
<b>第 7 章 RF タグとの通信</b>		<b>58</b>
7.1	UHF_InventoryCmd メソッド	59
7.2	UHF_InventoryReadCmd メソッド	61
7.3	UHF_Read メソッド	65
7.4	UHF_Write メソッド	66
7.5	UHF_BlockWrite メソッド	67
7.6	UHF_Lock メソッド	68
7.7	UHF_Kill メソッド	70
7.8	UHF_BlockErase メソッド	71
7.9	UHF_BlockWrite2 メソッド	72
7.10	UHF_Encode メソッド	74
7.11	UHF_ThroughCmd メソッド	77
<b>第 8 章 汎用メソッド</b>		<b>79</b>
8.1	SendData メソッド	80
8.2	ClearSerialInputBuffer メソッド	81
8.3	ClearSerialOutputBuffer メソッド	82
<b>第 9 章 プロパティ</b>		<b>83</b>
9.1	シリアル通信用プロパティ	84
9.1.1	PortState プロパティ/IsOpen プロパティ	84
9.1.2	PortNumber プロパティ	84
9.1.3	BaudRate プロパティ	84
9.1.4	SerialInputBufferSize プロパティ	84
9.1.5	SerialOutputBufferSize プロパティ	84
9.1.6	SerialInputBufferByteCount プロパティ	85
9.1.7	SerialOutputBufferByteCount プロパティ	85
9.1.8	SerialConnTimeout プロパティ	85
9.1.9	FlowControl プロパティ	85
9.2	TCP/IP 通信用プロパティ	86
9.2.1	Connected プロパティ	86
9.2.2	RemoteHost プロパティ	86
9.2.3	RemotePort プロパティ	86
9.2.4	SocketInputBufferSize プロパティ	86
9.2.5	SocketOutputBufferSize プロパティ	86
9.2.6	SocketInputBufferByteCount プロパティ	87
9.2.7	SocketConnTimeout プロパティ	87
9.3	共通プロパティ	88
9.3.1	Timeout プロパティ	88
<b>第 10 章 データ型</b>		<b>89</b>
10.1	列挙体	90
10.1.1	RFID_BaudRate	90
10.1.2	RFID_FlowControl	90
10.1.3	RFID_LEDMode	90
10.1.4	RFID_ProductSeries	90
10.1.5	RFID_ScanMode	90
10.1.6	RFID_TransmitSignal	90
10.1.7	RFID_AntennaSwitching	91
10.1.8	RFID_FrequencyChannelKounai	92
10.1.9	RFID_FrequencyChannelTokushou	92

10.1.10	RFID_FrequencyChannelActive .....	92
10.1.11	RFID_FrequencyChannel.....	93
10.1.12	RFID_FrequencyScanMode .....	94
10.1.13	RFID_SendCommand .....	95
10.1.14	RFID_UseBuzzer .....	95
10.1.15	UHF_ParamKind .....	95
10.1.16	UHF_Action.....	96
10.1.17	UHF_DR.....	97
10.1.18	UHF_M.....	97
10.1.19	UHF_MemBank.....	97
10.1.20	UHF_Session.....	98
10.1.21	UHF_Sel.....	98
10.1.22	UHF_InventoryTarget.....	98
10.1.23	UHF_Target.....	99
10.1.24	UHF_TRext .....	99
10.1.25	UHF_Truncate .....	99
10.1.26	UHF_ThroughCmdType.....	100
10.2	メソッド引数用クラス .....	101
10.2.1	ActionModeOption .....	101
10.2.2	UHF_SelectOption.....	102
10.2.3	UHF_InventoryOption .....	103
10.2.4	UHF_ExpandSelectOption.....	106
10.2.5	UHF_AntennaParamOption .....	107
10.2.6	UHF_TxPowerParamOption .....	107
10.2.7	UHF_FrequencyParamOption.....	108
10.2.8	UHF_RFTAGComParamOption.....	108
10.2.9	UHF_EPCParamOption .....	109
10.2.10	UHF_TIDLockOption.....	111
10.2.11	UHF_EPCLockOption.....	113
10.2.12	UHF_UserLockOption .....	115
10.2.13	UHF_AccessPwdLockOption.....	117
10.2.14	UHF_KillPwdLockOption .....	119
10.2.15	UHF_EncodeOption .....	121
10.2.16	UHF_ThroughCmdOption .....	122
<b>第 11 章 UTR-SDKV1 プログラミング.....</b>		<b>123</b>
11.1	プロジェクトの作成.....	124
11.2	UTR-SDKV1 への参照の追加 .....	125
11.3	ResponseRFID イベントハンドラの追加.....	126
11.4	リーダライタとの通信処理を記述 .....	128
<b>第 12 章 付録.....</b>		<b>129</b>
12.1	NACK 応答 .....	130
12.2	ResponseRFID イベントパラメータ .....	133
12.2.1	リーダライタの自動読み取りモード .....	133
12.2.2	リーダライタの制御.....	135
12.2.3	リーダライタ動作モードの設定 .....	135
12.2.4	リーダライタ FLASH の設定 .....	135
12.2.5	リーダライタ設定 .....	136
12.2.6	RF タグとの通信 .....	137
<b>変更履歴 .....</b>		<b>138</b>

---

---

# 第1章 SDK の仕組み

本章では、SDK の基本的な仕組みについて説明します。

---

---

## 1.1 シリアルインターフェースとLANインターフェース

UTR-SDKV1 シリーズは、単一のモジュール (DLL) でシリアルインターフェースを持つリーダーライター (USB インターフェース含む) と LAN インターフェースを持つリーダーライターの双方の制御が可能なインターフェースを提供します。

ただし、シリアル通信のインターフェースと LAN 通信のインターフェースを同時に利用することはできません。

シリアルインターフェースを持つリーダーライターと通信する場合には、SDK の **Open** メソッドで通信路の確保を行い、**Close** メソッドを使用して通信路を開放してください。

LAN インターフェースを持つリーダーライターと通信する場合には、SDK の **Connect** メソッドで通信路の確保を行い、**Disconnect** メソッドを使用して通信路を開放してください。

その他のメソッドは、リーダーライターのインターフェースに関わらず共通です。

SDK は、開かれている通信路を確認して適切な通信路でコマンドの送受信を行います。

リーダーライター	シリアルインターフェース	LAN インターフェース
通信路の確保	Open メソッド	Connect メソッド
通信路の開放	Close メソッド	Disconnect メソッド

※USB インターフェースのリーダーライターは、シリアルインターフェースとして接続可能です。

## 1.2 ResponseRFID イベント

SDKでは、リーダライタへのコマンド送信を関数化していますが（関数を呼び出すことでリーダライタへのコマンド送信が行われますが）、リーダライタからの応答は、イベント（ResponseRFID イベント）として上位アプリケーションへ通知されます。  
ResponseRFID イベントのパラメータ（InputEventArgs）は下表のとおりです。

InputEventArgsパラメータ	
SendCommand	<p>リーダライタから受信した応答の種類がセットされます。</p> <p>上位アプリケーションからの関数呼び出しに対する応答の場合は、呼び出した関数名がセットされます。</p> <p>ただし、NACK 応答の場合は、呼び出した関数の如何に関わらず NACK がセットされます。</p> <p>また、不明なコマンド列を受信した場合には <b>Other</b> がセットされます。</p>
InputData	<p>リーダライタから受信したコマンド列がセットされます。</p> <p>InputData はバイト型の配列として定義されています。</p>
TextData	<p>RF タグから読み取ったデータが文字列でセットされます。</p> <p>文字列への変換に使用する文字コードは Shift-JIS です。</p>
BinaryData	<p>RF タグから読み取ったデータがセットされます。</p> <p>BinaryData はバイト型の配列として定義されています。</p>
PC	<p>RF タグから読み取った PC がセットされます。</p> <p>PC はバイト型の配列として定義されています。</p> <p>なお、PC は、上位バイトが配列の先頭に来るようにセットされます。</p>
EPC	<p>RF タグから読み取った EPC(UII)がセットされます。</p> <p>EPC はバイト型の配列として定義されています。</p> <p>なお、EPC(UII)は、上位バイトが配列の先頭に来るようにセットされます。</p>
TID	<p>RF タグから読み取った TID がセットされます。</p> <p>本パラメータにセットされる TID は、[UHF_SetInventoryParam メソッド]を tidRead=true を指定して実行し、リーダライタに[TID を付加する]設定が書き込まれた状態で、[UHF_InventoryReadCmd メソッド]または「UHF 連続イベントリリードモード」で読み取りした際の TID データとなります。</p> <p>TID はバイト型の配列として定義されています。</p> <p>なお、TID は、上位バイトが配列の先頭に来るようにセットされます。</p>
MemBankData	<p>RF タグから読み取った MemBank のデータがセットされます。</p> <p>MemBank のデータとは、UHF 連続イベントリリードモード、または [UHF_InventoryReadCmd メソッド]で読み取り指定した MemBank 領域のデータを示します。</p> <p>MemBankData はバイト型の配列として定義されています。</p> <p>なお、MemBankData は、上位バイトが配列の先頭に来るようにセットされます。</p>
RSSI	<p>RF タグを読み取った時の RSSI 値を 10 倍した値がセットされます。</p> <p>RSSI は int 型の整数値として定義されています。</p> <p>(例) RSSI 値が-41.7 の場合、e.RSSI には-417 がセットされます。</p>



## 1.3 自動読み取りモードでの応答の受信

ISO/IEC18000-63 及び EPCglobal Class1 Generation2 準拠の RF タグは、必ずリーダライタからのコマンドを受信した後でリーダライタにレスポンスを返す仕様です。リーダライタからのコマンドを受信しない限り、RF タグがデータを返すことはありません。このシーケンスを「RTF: Reader Talk First」と呼びます。

しかし、UTR シリーズ リーダライタでは上位機器から制御コマンドを送ることなく RF タグのデータを読み取ることが可能な、自動読み取りモード（連続インベントリモード、連続インベントリリードモード）を備えています。

リーダライタが自動読み取りモードに設定されている場合は、SDK の関数呼び出しが行われなくても RF タグのデータ読み取りが行われ、ResponseRFID イベントが発生します。

※リーダライタ動作モードの詳細は、ご使用の製品に対応した各種通信プロトコル説明書を参照ください。

## 1.4 SDK の内部処理とタイムアウトについて

SDK の内部処理は以下のようなイメージで行われているため、ResponseRFID イベント内で停止（例.MessageBox の表示）、遅延（他のメソッドを実行）させるような処理を記述すると、

SDK の内部処理に影響を与えます。

正常にレスポンスを受信した場合であっても、メソッドの戻り値がタイムアウトとなる場合がありますのでご注意ください。

### <SDK の内部処理概要>

- ①上位アプリケーションからのメソッド受付（例.GetROMVersion）
- ②リーダライタへ送信するコマンドの構築
- ③タイムアウト監視用タイマスタート
- ④リーダライタへのコマンド送信
- ⑤リーダライタからの応答待機
- ⑥リーダライタからの応答受信
- ⑦ResponseRFID イベントの起動（上位アプリケーションへ応答内容を通知）
- ⑧リーダライタからの応答内容判別（ACK or NACK or タイムアウト）
- ⑨メソッドの戻り値（ACK or NACK or タイムアウト）返信

SDK に含まれるほぼ全てのメソッドは、上記①から開始して⑨で戻り値を返します。

また、タイムアウトの管理については、③～⑧の間で行っています。

※④～⑦の間でなんらかの異常動作があった場合にも、メソッドの処理を上位側へ戻せるように管理しています。

そのため、ResponseRFID イベント（⑦）の処理部分（上位側で実装）で処理の停止、遅延が生じた場合は、メソッドはタイムアウトします。

複数のメソッドを続けて実行する場合、以下のような流れで処理を記述することでタイムアウトすることなく制御することができます。

### <制御例>

[UHF\_InventoryCmd メソッド]で RF タグの EPC を取得し、[UHF\_SetSelectParam メソッド]で特定の EPC でマスク指定し、[UHF\_Read メソッド]で特定の RF タグのデータを読み取る場合の処理の一例を示します。

- ・ [UHF\_InventoryCmd メソッド]実行
- ・ ResponseRFID イベント内の処理で、e.EPC の値（読み取った RF タグの EPC）をバイト配列にコピー
- ・ [UHF\_InventoryCmd メソッド]の戻り値を確認
- ↓
- ・ 戻り値が ACK の場合、コピーしておいた EPC を使用して[UHF\_SetSelectParam メソッド]でマスク指定を実行
- ・ [UHF\_SetSelectParam メソッド]の戻り値を確認
- ↓
- ・ 戻り値が ACK の場合、[UHF\_Read メソッド]を実行
- ・ ResponseRFID イベント内の処理で、e.BinaryData の値（読み取った RF タグデータ）をバイト配列にコピー
- ・ [UHF\_Read メソッド]の戻り値を確認
- ・ 戻り値が ACK の場合、バイト配列にコピーした RF タグのデータを使用して次の処理（画面表示や DB 更新など）を実行



<注意事項> 「コマンドタイムアウト時間」の設定によるタイムアウト

リーダーライタの ROM バージョンが 1.090 以降の場合、FLASH 設定により、BlockWrite コマンドと Write コマンドのタイムアウト時間の設定が変更できるようになりました。

「コマンドタイムアウト時間」をメソッドのタイムアウト時間よりも長くすると、リーダーライタからのレスポンスが正しく受信できませんのでご注意ください。

「コマンドタイムアウト時間」の設定は、以下のメソッド実行時に有効となります。

- [UHF\_Write メソッド]
- [UHF\_BlockWrite メソッド]
- [UHF\_Encode メソッド]

---

---

## 第2章 通信路の確保と開放

本章では、通信路の確保と開放及びリソースの解放に対応した関数について説明します。

---

---

## 2.1 Open メソッド

シリアルポートのオープンを行います。

既にシリアルポートがオープンされている場合は、`false` を返します。  
ソケットがオープンされている場合は、ソケットをクローズしてからシリアルポートをオープンします。

```
bool Open();  
bool Open(int portNo);  
bool Open(int portNo, RFID_BaudRate baudRate);
```

### 【パラメータ】

値	説明
portNo	オープンするシリアルポート番号を指定します。 指定しない場合は、 <code>PortNumber</code> プロパティで指定した COM ポートを使用します。
baudRate	通信速度を指定します。 <code>BaudRate115200: 115200bps</code>

### 【戻り値】

シリアルポートのオープンに成功した場合は `true`、失敗した場合は `false` を返します。

### 【特記事項】

USB インターフェースのリーダーライタは仮想 COM ポートとして認識されるため、本メソッドで接続することができます。

### 【参照】

Close メソッド、`RFID_BaudRate` 列挙体、`PortNumber` プロパティ

## 2.2 Close メソッド

シリアルポートのクローズを行います。

```
bool Close();
```

### [戻り値]

シリアルポートのクローズに成功した場合は `true`、失敗した場合は `false` を返します。

## 2.3 Connect メソッド

ソケットのオープンを行います。

既にソケットがオープンされている場合は、**false** を返します。  
シリアルポートがオープンされている場合は、シリアルポートをクローズしてからソケットのオープンを行います。

```
bool Connect();  
bool Connect(string remoteHost, int remotePort);  
bool Connect(IPAddress remoteAddress, int remotePort);
```

### 【パラメータ】

値	説明
remoteHost	接続先のホスト名を文字列で指定します。 (DNS を参照するため、IP アドレスを指定した場合より、時間がかかります。) 指定しない場合は、RemoteHost プロパティで指定したホスト名を使用します。
remotePort	接続に使用する TCP ポートの番号を指定します。 指定しない場合は、RemotePort プロパティで指定したポート番号を使用します。
remoteAddress	接続先の IP アドレスを指定します。

### 【戻り値】

ソケットのオープンに成功した場合は **true**、失敗した場合は **false** を返します。

### 【特記事項】

- 本メソッドは、必ず上位側からリーダライタに対して接続処理を行います。  
リーダライタから上位側への自動接続処理はサポートしていませんのでご注意ください。  
UTR-SDKV1-PC を使用して LAN インターフェースのリーダライタを制御する場合、「LAN インターフェース設定ツール : IPSet2」の設定項目が「Active Connect : None」に設定されたリーダライタのみ接続することができます。
- LAN インターフェースのリーダライタは、電源起動後 500ms の間はソケットオープン処理を受け付けませんのでご注意ください。

### 【参照】

Disconnect メソッド、RemoteHost プロパティ、RemotePort プロパティ

## 2.4 Disconnect メソッド

ソケットのクローズを行います。

```
bool Disconnect();
```

### [戻り値]

ソケットのクローズに成功した場合は `true`、失敗した場合は `false` を返します。

## 2.5 Dispose メソッド

保持しているリソースを全て解放します。  
シリアルポートまたはソケットがオープンされている場合は、クローズします。

```
void Dispose();
```

### 【戻り値】

なし

---

---

## 第3章 リーダライタの制御

本章では、リーダーライタの制御に対応した関数について説明します。

---

---

---

## 3.1 GetError メソッド

リーダーライタのエラー情報を読み取ります。

```
int GetError();
```

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand.GetError

e.InputData

6 バイト目 : エラー情報

0x00 : 正常

0x00 以外 : 異常 (リーダーライタ内部のハード的な異常を検出した場合)

## 3.2 GetROMVersion メソッド

リーダーライタの ROM バージョン（ファームウェアバージョン）を読み取ります。

```
int GetROMVersion();
```

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand.GetROMVersion

e.BinaryData

1 バイト目 : メジャーバージョン番号

2~4 バイト目 : マイナーバージョン番号

5~7 バイト目 : シリーズ名

UTR シリーズ (UMP)

8~9 バイト目 : 機種判別用データ

e.TextData

ROM バージョンが文字列でセットされます。

以下の 2 点で、e.BinaryData と内容が異なります。

- ・メジャーバージョン番号とマイナーバージョン番号の間に「.」（ドット）が入る。
- ・マイナーバージョン番号の上位 2 バイトと下位 1 バイトの間に「」（スペース）が入る。

[e.BinaryData] 1080UMP01 → [e.TextData] 1.08 0UMP01

例) UTR-S101 の場合

[e.BinaryData] 1080UMP01

[e.TextData] 1.08 0UMP01

## 3.3 GetCHIPVersion メソッド

リーダーライタの CHIP バージョン（ファームウェアバージョン）を読み取ります。

```
int GetCHIPVersion((bool getserial);
```

### [パラメータ]

値	説明
getserial	false : CHIP バージョンの取得 true : CHIP のシリアル番号取得

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

```
RFID_SendCommand.GetCHIPVersion
```

e.BinaryData

CHIP バージョンの取得の場合

1 バイト目 : メジャーバージョン  
2~4 バイト目 : マイナーバージョン  
5~7 バイト目 : シリーズ名  
UTR シリーズ CHIP (UIC)

8~9 バイト目 : 機種判別用データ

(例) CHIP バージョン番号が 1.080UIC01 の場合

```
[e.BinaryData] 31 30 38 30 55 49 43 30 31
```

CHIP シリアル番号の取得の場合

1~9 バイト目 : シリアル番号

(例) CHIP シリアル番号が A6Q2260195 の場合

```
[e.BinaryData] 41 36 51 32 32 36 30 31 39 35
```

e.TextData

CHIP バージョンの取得の場合

CHIP のバージョンが文字列でセットされます。

以下の 2 点で、e.BinaryData と内容が異なります。

- メジャーバージョン番号とマイナーバージョン番号の間に「.」(ドット)が入る。
- マイナーバージョン番号とチップ名の間に「 」(スペース)が入る。

(例) CHIP バージョン番号が 1.080UIC01 の場合

```
[e.TextData] 1.080 UIC01
```

CHIP シリアル番号の取得の場合

CHIP のシリアル番号が文字列でセットされます。

(例) CHIP シリアル番号が A6Q2260195 の場合

```
[e.TextData] A6Q2260195
```

## 3.4 SetTransmitSignal メソッド

リーダーライタが出力する RF 送信信号（キャリア）の制御を行います。

※リーダーライタの ROM バージョン 1.080 以降対応のメソッドです。

```
int SetTransmitSignal(bool isOn);  
int SetTransmitSignal(RFID_TransmitSignal signal);
```

### 【パラメータ】

値	説明
isOn	RF 送信信号（キャリア）の On/Off を指定します。 true を指定した場合、On になります。 false を指定した場合、Off になります。
signal	RF 送信信号（キャリア）の On/Off/Off→On を指定します。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

```
e.SendCommand  
RFID_SendCommand. SetTransmitSignal
```

### 【参照】

RFID\_TransmitSignal 列挙体

## 3.5 CallLEDBuzzer メソッド

リーダーライタの LED とブザーを同時に制御します。

```
int CallLEDBuzzer(    byte portNo,
                    RFID_LEDMode mode,
                    byte setting,
                    byte buzzerType,
                    byte rumblingTime);
```

### [パラメータ]

値	説明
portNo	制御ポート（点灯させる LED）を指定します。 0x00：制御しない（LED 制御なし） 0x01：汎用ポート 1 の制御（青色 LED の制御） 0x04：汎用ポート 3 の制御（赤色 LED の制御） 0x05：汎用ポート 1 と 3 の制御（青・赤 LED の制御）
mode	LED の動作モード 指定時間の点灯・常時点滅・常時点灯または消灯を指定します。
setting	LED の点灯または消灯時間、点滅間隔を指定します。  <ul style="list-style-type: none"> <li>• mode=RFID_LEDMode.AppointTime の場合 「指定時間の点灯」：setting×200ms 間の点灯</li> <li>• mode=RFID_LEDMode.Blink の場合 「常時点滅」：setting×200 ms 間隔の点滅</li> <li>• mode=RFID_LEDMode.Always の場合 「常時点灯または消灯」 0x00：消灯 0x01：常時点灯</li> </ul>
buzzerType	ブザー音を指定します。 0x00：ピー 0x01：ピッピッピッ 0x02：ピッピー 0x03：ピッピッピー 0x04：ピーー 0x05：ピーピーピーピー 0x06：ピーーーー 0x07：ピッピッピッピッピッ 0x08：ピッピッピッピッ 0xFF：時間指定連続音（ピー）
rumblingTime	ブザーの鳴動有無または鳴動時間を指定します。  <ul style="list-style-type: none"> <li>• buzzerType=0xFF 以外の場合 ブザー音が「時間指定連続音（ピー）」以外の場合 0x00：鳴動しない 0x01：鳴動する</li> <li>• buzzerType=0xFF の場合 ブザー音が「時間指定連続音（ピー）」の場合 rumblingTime×200ms の鳴動</li> </ul>

**【戻り値】**

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

**【レスポンス】**

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand.CallLEDBuzzer

**【特記事項】**

本メソッドで LED とブザーを制御するためには、リーダーライタの汎用ポート 1 および汎用ポート 3 の機能が「汎用ポート」に設定されていることが必要です。  
汎用ポート 1 または汎用ポート 3 の機能が「汎用ポート」でない場合、ブザーと LED が制御できません。（リーダーライタから NACK 応答が返されます）

**【参照】**

RFID\_LEDMode 列挙体

## 3.6 Restart メソッド

リーダーライタをリスタートします。

```
int Restart();
```

### 【戻り値】

値	説明
0	送信成功
-1	送信失敗

### 【レスポンス】

リーダーライタは、本メソッドに対する応答を返しません。  
そのため、本メソッドを実行しても **ResponseRFID** イベントは発生しません。

### 【特記事項】

リーダーライタは、リスタート実行後から一定時間は、次のメソッドに応答できません。  
400ms 以上の時間を空けてください。

## 3.7 CallBuzzer メソッド

リーダーライタのブザーを制御します。

```
int CallBuzzer(byte buzzerType);
int CallBuzzer(byte buzzerType, bool needResponse);
```

### 【パラメータ】

値	説明
buzzerType	ブザー音を指定します。 0x00 : ピー 0x01 : ピッピッピッ 0x02 : ピッピー 0x03 : ピッピッピー 0x04 : ピーー 0x05 : ピーピーピーピー 0x06 : ピーーー 0x07 : ピッピッピッピッピッ 0x08 : ピッピッピッピッ
needResponse	リーダーライタへの応答要求を指定します。 true : 応答を要求する false : 応答を要求しない 指定しなかった場合は、false の指定となります。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

- needResponse=true を指定した場合  
ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand.CallBuzzer
- needResponse=false を指定した場合  
リーダーライタは本メソッドに対する応答を返しません。  
そのため、本メソッドを実行しても ResponseRFID イベントは発生しません。

### 【特記事項】

本メソッドでブザーを制御するためには、リーダーライタの汎用ポート7の機能が「ブザー制御信号出力ポート」に設定されていることが必要です。  
汎用ポート7の機能が「汎用ポート」に設定されている場合は、ブザーが制御できません。  
また、リーダーライタへの応答を要求していない場合、戻り値「0」はACK 応答ではなくコマンドの送信成功を表しています。

### 【例外】

ArgumentOutOfRangeException	
buzzerType	指定可能な値の範囲は 0～8 です。

## 3.8 InitFLASH メソッド

リーダーライタの FLASH 設定を出荷時設定に戻します。  
本メソッド実行後は、Restart メソッドの実行あるいはリーダーライタの電源再起動を実行してください。

```
int InitFLASH();
```

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand.InitFLASH

## 3.9 UHF\_GetHandle メソッド

リーダーライタの内部で保持している「RF タグから最後に取得した Handle データ」を取得します。

```
int UHF_GetHandle();
```

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand.UHFGetHandle

e.BinaryData

RF タグから受信したハンドル値（ワードの値）

※MSB ファーストでセットされます。

### 【特記事項】

- UHF\_ThroughCmd を使用してカスタムコマンドを実行する場合などにおいて、RF タグから取得した Handle データをコマンドパラメータにセットする必要がある場合があります。必要に応じて、本メソッドを使用して、リーダーライタから Handle データを取得してください。

本メソッドを実行して UHF\_ThroughCmd を実行する場合は、以下の手順で処理をおこなう必要があります。

1. RF 送信信号の制御コマンドを実行し、キャリア出力を ON する。（必須処理）
2. Inventory コマンドを以下のパラメータで実行し、RF タグを Open 状態とする。（必須処理）
  - Q 値の自動 UP/DOWN 機能=使用しない
  - Q 値の開始値=0
  - Q 値の最小値=0
  - Q 値の最大値=0※複数の RF タグから 1 枚を指定して処理をおこなう場合は、事前に Select コマンドを実行する設定も必要です。
3. UHF\_GetHandle コマンドを実行して Handle データを取得する。（必要に応じて）  
※Handle を自動で付加する設定の場合は不要な処理です。
4. UHF\_ThroughCmd を使用して RF タグのカスタムコマンドを実行する。
5. RF 送信信号の制御コマンドを実行し、キャリア出力を OFF する。（推奨処理）  
※不要な電波を出し続けると電波干渉の要因となりますので、処理終了後は直ちにキャリア出力 OFF することを推奨します。

---

---

## 第4章 リーダライタ動作モードの設定

本章では、リーダライタの動作モード設定の関数について説明します。

---

---

## 4.1 GetActionMode メソッド

リーダーライタの動作モードを読み取ります。

```
int GetActionMode();
```

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand.GetActionMode

e.InputData

6 バイト目：リーダーライタ動作モード

8 バイト目：リーダーライタ動作モードオプション

6 バイト目（リーダーライタ動作モード）

値	対応する RFID_ScanMode	説明
0x00	CommandScanMode	コマンドモード
0x65	UHF_InventoryMode	UHF 連続インベントリモード
0x66	UHF_InventoryReadMode	UHF 連続インベントリリードモード

8 バイト目（リーダーライタ動作モードオプション）のビット構成

ビット	対応する列挙体	説明
0~1	—	将来拡張のための予約（通常は 0）
2	—	将来拡張のための予約（通常は 0）
3	—	将来拡張のための予約（通常は 0）
4	RFID_UseBuzzer	ブザー 0：鳴らさない 1：鳴らす
5	—	将来拡張のための予約（通常は 0）
6~7	RFID_BaudRate	通信速度 3：115200bps (BaudRate115200)

### [参照]

SetActionMode メソッド、ActionModeOption 構造体

RFID\_UseBuzzer 列挙体、RFID\_BaudRate 列挙体

## 4.2 SetActionMode メソッド

リーダーライタの動作モードを書き込みます。

```
int SetActionMode(RFID_ScanMode mode,  
                 ActionModeOption option,  
                 bool writeFLASH);
```

### 【パラメータ】

値	説明
mode	リーダーライタ動作モードを指定します。
option	リーダーライタ動作モードオプションを指定します。
writeFLASH	設定を FLASH に保存する場合は true を指定します。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand.SetActionMode

### 【参照】

GetActionMode メソッド、ActionModeOption 構造体、RFID\_ScanMode 列挙体

---

---

## 第5章 リーダライタ FLASH の設定

本章では、リーダーライタ FLASH の設定に対応した関数について説明します。

---

---

## 5.1 ReadFLASH メソッド

リーダーライタの FLASH 設定値をアドレス単位（1 バイト単位）で読み取ります。

```
int ReadFLASH( byte address,  
              out byte value);
```

### 【パラメータ】

値	説明
address	読み取りアドレスを指定します。
value	読み取ったアドレスの FLASH 設定値がセットされます。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand.ReadFLASH

### 【参照】

WriteFLASH メソッド

FLASH アドレス及び設定値は、ご使用になるリーダーライタの製品仕様書、または同リーダーライタに対応した通信プロトコル説明書をご参照ください。

## 5.2 WriteFLASH メソッド

リーダーライタの FLASH 設定値をアドレス単位（1 バイト単位）で書き込みます。  
FLASH 設定値変更後は、リーダーライタをリスタートする必要があります。

```
int WriteFLASH( byte address,  
               byte value);
```

### 【パラメータ】

値	説明
address	書き込みアドレスを指定します。
value	書き込みデータを指定します。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand.WriteFLASH

### 【参照】

ReadFLASH メソッド

FLASH アドレス及び設定値は、ご使用になるリーダーライタの製品仕様書、または同リーダーライタに対応した通信プロトコル説明書をご参照ください。

---

---

## 第6章 リーダライタ設定

本章では、リーダーライタ設定コマンドに対応した関数について説明します。

---

---

## 6.1 UHF\_GetInventoryParam メソッド

「コマンドモード」、「UHF 連続インベントリモード」、および「UHF 連続インベントリリードモード」のインベントリ処理におけるパラメータ値の取得を行います。

```
int UHF_GetInventoryParam(
    UHF_ParamKind paramKind,
    out UHF_InventoryOption inventoryOpt,
    out UHF_MemBank membankOpt,
    out uint startWordAddr,
    out byte wordCount,
    out bool tidRead);
```

### [パラメータ]

値	説明
paramKind	パラメータ種類を指定します。 Param_Command     0 : コマンドモード用パラメータ Param_Auto         1 : 自動読み取りモード用パラメータ Param_Flash        2 : FLASH データ
inventoryOpt	インベントリ処理用オプションパラメータがセットされます。
membankOpt	インベントリリード時に読み取るメモリバンクを指定する MemBank オプションがセットされます。
startWordAddr	インベントリリード時の、MemBank で指定した領域の読み取り開始ワードアドレスがセットされます。
wordCount	インベントリリード時の、MemBank で指定した領域の読み取りワード数がセットされます。
tidRead	インベントリリード時の、MemBank で指定した領域に加えて TID も読み取るかどうかセットされます。

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFGetInventoryParam

e.InputData

6 バイト目以降にインベントリ処理用オプションパラメータがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

**【特記事項】**

- ・ 戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

inventoryOpt : UHF\_InventoryOption のデフォルト値  
membankOpt : UHF\_MemBankOption のデフォルト値  
startWordAddr : 0  
wordCount : 0  
tidRead : false

**【参照】**

UHF\_SetInventoryParam メソッド、UHF\_InventoryOption 構造体、  
UHF\_MemBank 列挙体

## 6.2 UHF\_GetSelectParam メソッド

Select コマンドパラメータを読み取ります。

```
int UHF_GetSelectParam(
    UHF_ParamKind paramKind,
    out UHF_SelectOption selectOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command     0：コマンドモード用パラメータ Param_Auto         1：自動読み取りモード用パラメータ Param_Flash        2：FLASH データ
selectOpt	Select コマンド用オプションがセットされます。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFGetSelectParam

e.InputData

6 バイト目以降に Select コマンドパラメータがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【特記事項】

- ・戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

selectOpt           : UHF\_SelectOption のデフォルト値

### 【参照】

UHF\_SetSelectParam メソッド、UHF\_SelectOption 構造体

## 6.3 UHF\_GetExpandSelectParam メソッド

2 回目以降の Select コマンドパラメータを読み取ります。

```
int UHF_GetExpandSelectParam(
    UHF_ParamKind paramKind,
    out UHF_ExpandSelectOption[] selectOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command     0 : コマンドモード用パラメータ Param_Auto         1 : 自動読み取りモード用パラメータ Param_Flash        2 : FLASH データ
selectOpt	2~8 回目までの Select コマンド用オプションがセットされます。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFGetExpandSelectParam

e.InputData

6 バイト目以降に 2~8 回目までの Select コマンドパラメータがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【例外】

ArgumentOutOfRangeException	
paramKind	FLASH データは存在しません。

### 【特記事項】

- ・ 戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
 プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
 は変数が以下の値に上書きされますのでご注意ください。

selectOpt           : UHF\_ExpandSelectOption のデフォルト値

### 【参照】

UHF\_SetExpandSelectParam メソッド、UHF\_ExpandSelectOption 構造体

## 6.4 UHF\_GetAntennaParam メソッド

アンテナ切替時のパラメータを読み取ります。

```
int UHF_GetAntennaParam(
    UHF_ParamKind paramKind,
    out UHF_AntennaParamOption antennaParamOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command     0：コマンドモード用パラメータ Param_Auto         1：自動読み取りモード用パラメータ Param_Flash        2：FLASH データ
antennaParamOpt	アンテナ切替設定のオプションがセットされます。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand  
     RFID\_SendCommand. UHFGetAntennaParam

e.InputData

6 バイト目以降にアンテナ切替設定のパラメータがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【特記事項】

- ・戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
 プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
 は変数が以下の値に上書きされますのでご注意ください。

antennaParamOpt : UHF\_AntennaParamOption のデフォルト値

### 【参照】

UHF\_SetAntennaParam メソッド、UHF\_AntennaParamOption 構造体

## 6.5 UHF\_GetTxPowerParam メソッド

キャリア出力設定のパラメータを読み取ります。

```
int UHF_GetTxPowerParam(
    UHF_ParamKind paramKind,
    out UHF_TxPowerParamOption TxPowerParamOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command      0 : コマンドモード用パラメータ Param_Auto            1 : 自動読み取りモード用パラメータ Param_Flash          2 : FLASH データ
TxPowerParamOpt	出力設定のオプションがセットされます。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFGetTxPowerParam

e.InputData

6 バイト目以降に出力設定のオプションのパラメータがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【特記事項】

- ・戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

TxPowerParamOpt : UHF\_TxPowerParamOption のデフォルト値

### 【参照】

UHF\_SetTxPowerParam メソッド、UHF\_TxPowerParamOption 構造体

## 6.6 UHF\_GetFrequencyParam メソッド

キャリアの周波数関係のパラメータを読み取ります。

```
int UHF_GetFrequencyParam(
    UHF_ParamKind paramKind,
    out UHF_FrequencyParamOption FrequencyParamOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command      0 : コマンドモード用パラメータ Param_Auto            1 : 自動読み取りモード用パラメータ Param_Flash          2 : FLASH データ
FrequencyParamOpt	周波数設定のオプションがセットされます。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFGetFrequencyParam

e.InputData

6 バイト目以降に周波数設定のオプションのパラメータがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【特記事項】

- ・戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

FrequencyParamOpt : UHF\_FrequencyParamOption のデフォルト値

### 【参照】

UHF\_SetFrequencyParam メソッド、UHF\_FrequencyParamOption 構造体

## 6.7 UHF\_GetRFTAGComParam メソッド

RF タグの通信関連パラメータを読み取ります。

```
int UHF_GetRFTAGComParam(
    UHF_ParamKind paramKind,
    out UHF_RFTAGComParamOption RFTAGComParamOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command     0 : コマンドモード用パラメータ Param_Auto         1 : 自動読み取りモード用パラメータ Param_Flash        2 : FLASH データ
RFTAGComParamOpt	RF タグの通信関連パラメータのオプションがセットされます。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFGetRFTAGComParam

e.InputData

6 バイト目以降に RF タグの通信関連パラメータのオプションがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【特記事項】

- ・戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

RFTAGParamOpt : UHF\_RFTAGComParamOption のデフォルト値

### 【参照】

UHF\_SetRFTAGComParam メソッド、UHF\_RFTAGComParamOption 構造体

## 6.8 UHF\_GetEPCParam メソッド

EPC(UII)関連パラメータを読み取ります。

```
int UHF_GetEPCParam(
    UHF_ParamKind paramKind,
    out UHF_EPCParamOption EPCParamOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command     0：コマンドモード用パラメータ Param_Auto         1：自動読み取りモード用パラメータ Param_Flash        2：FLASH データ
EPCParamOpt	EPC(UII)関連パラメータのオプションがセットされます。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFGetEPCParam

e.InputData

6 バイト目以降に EPC(UII)関連パラメータのオプションがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【特記事項】

- ・戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

EPCParamOpt : UHF\_EPCParamOption のデフォルト値

### 【参照】

UHF\_SetEPCParam メソッド、UHF\_EPCParamOption 構造体

## 6.9 UHF\_SetInventoryParam メソッド

EPC 自動読取モードパラメータを書き込みます。

```
int UHF_SetInventoryParam(
    UHF_ParamKind paramKind,
    UHF_InventoryOption inventoryOpt,
    UHF_MemBank membankOpt,
    uint startWordAddr,
    byte wordCount,
    bool tidRead);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command     0 : コマンドモード用パラメータ Param_Auto         1 : 自動読み取りモード用パラメータ Param_Flash        2 : FLASH データ
inventoryOpt	インベントリ処理用オプションを指定します。
membankOpt	インベントリリード時に読み取るメモリバンクの MemBank オプションを指定します。
startWordAddr	インベントリリード時、MemBank で指定した領域の読み取り開始ワードアドレスを指定します。
wordCount	インベントリリード時、MemBank で指定した領域の読み取りワード数を指定します。
tidRead	インベントリリード時、MemBank で指定した領域に加えて TID も読み取る場合は true を指定します。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand. UHFSetInventoryParam

### 【例外】

ArgumentOutOfRangeException	
inventoryOpt.Q	指定可能な値の範囲は 0~15 です。
inventoryOpt.Qmin	指定可能な値の範囲は 0~15 です。
inventoryOpt.Qmax	指定可能な値の範囲は 0~15 です。
wordCount	指定可能な値の範囲は 1~32 です。

### 【参照】

UHF\_GetInventoryParam メソッド、UHF\_InventoryOption 構造体、UHF\_MemBank 列挙体

## 6.10 UHF\_SetSelectParam メソッド

インベントリ処理時に実行する Select コマンドのパラメータを書き込みます。

```
int UHF_SetSelectParam(
    UHF_ParamKind paramKind,
    UHF_SelectOption selectOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command     0：コマンドモード用パラメータ Param_Auto         1：自動読み取りモード用パラメータ Param_Flash        2：FLASH データ
selectOpt	インベントリ処理時に実行する Select コマンド用オプションを指定します。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ  
 e.SendCommand  
 RFID\_SendCommand. UHFSetSelectParam

### 【例外】

<b>ArgumentOutOfRangeException</b>	
selectOpt.maskLength	指定可能な値の範囲は 0～128 です。
<b>ArgumentNullException</b>	
selectOpt.maskValue	maskLength に 0 以外が設定された場合、null を指定することはできません。

### 【参照】

UHF\_GetSelectParam メソッド、UHF\_SelectOption 構造体

## 6.11 UHF\_SetExpandSelectParam メソッド

2 回目以降の Select コマンド用のパラメータ値の設定を行うメソッドです。  
UHF\_SetSelectParam メソッドにて指定したマスクに加えて、別のエリアのマスク対象を指定するために使用します。

```
int UHF_SetExpandSelectParam(
    UHF_ParamKind paramKind,
    UHF_ExpandSelectOption[] selectOpt,
    byte paramCount);
```

### [パラメータ]

値	説明
paramKind	パラメータ種類を指定します。 (※) Param_Command 0 : コマンドモード用パラメータ Param_Auto 1 : 自動読み取りモード用パラメータ
selectOpt	インベントリ処理時に実行する 2 回目以降の Select コマンド用オプションを指定します。
paramCount	2 回目以降のパラメータの個数(マスクデータ数)を指定します。

※UHF\_SetExpandSelectParam メソッドのパラメータの paramKind に Param\_FLASH を指定することはできません。指定すると[例外]が返ります。

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand. UHFSetExpandSelectParam

### [例外]

ArgumentOutOfRangeException	
paramKind	FLASH データは存在しません。
selectOpt[].maskLength	指定可能な値の範囲は 0~128 です。
paramCount	指定可能な値の範囲は 1~7 です。
ArgumentNullException	
selectOpt[].maskLength	指定可能な値の範囲は 0~128 です。
selectOpt[].maskValue	maskLength に 0 以外が設定された場合、null を指定することはできません。

### [参照]

UHF\_GetExpandSelectParam メソッド、UHF\_ExpandSelectOption 構造体

## 6.12 UHF\_SetAntennaParam メソッド

アンテナ切替時のパラメータを書き込みます。

```
int UHF_SetAntennaParam(  
    UHF_ParamKind paramKind,  
    UHF_AntennaParamOption antennaParamOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command    0 : コマンドモード用パラメータ Param_Auto        1 : 自動読み取りモード用パラメータ Param_Flash       2 : FLASH データ
antennaParamOpt	アンテナ切替設定のオプションをセットします。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFSetAntennaParam

e.InputData

6 バイト目以降にアンテナ切替設定のパラメータがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【特記事項】

- ・戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

antennaParamOpt : UHF\_AntennaParamOption のデフォルト値

### 【参照】

UHF\_GetAntennaParam メソッド、UHF\_AntennaParamOption 構造体

## 6.13 UHF\_SetTxPowerParam メソッド

キャリア出力関係のパラメータを書き込みます。

```
int UHF_SetTxPowerParam(
    UHF_ParamKind paramKind,
    UHF_TxPowerParamOption TxPowerParamOpt)
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command     0：コマンドモード用パラメータ Param_Auto         1：自動読み取りモード用パラメータ Param_Flash        2：FLASH データ
TxPowerParamOpt	出力設定のオプションがセットされます。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFSetTxPowerParam

e.InputData

6 バイト目以降に出力設定のオプションのパラメータがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【特記事項】

- ・戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

TxPowerParamOpt : UHF\_TxPowerParamOption のデフォルト値

### 【参照】

UHF\_GetTxPowerParam メソッド、UHF\_TxPowerParamOption 構造体

## 6.14 UHF\_SetFrequencyParam メソッド

キャリアの周波数関係のパラメータを書き込みます。

```
int UHF_SetFrequencyParam(
    UHF_ParamKind paramKind,
    UHF_FrequencyParamOption frequencyParamOpt)
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command    0 : コマンドモード用パラメータ Param_Auto        1 : 自動読み取りモード用パラメータ Param_Flash       2 : FLASH データ
frequencyParamOpt	キャリアの周波数関係パラメータのオプションをセットします。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFSetFrequencyParam

e.InputData

6 バイト目以降に周波数設定のオプションのパラメータがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【特記事項】

- ・戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

frequencyParamOpt : UHF\_FrequencyParamOption のデフォルト値

### 【参照】

UHF\_GetFrequencyParam メソッド、UHF\_FrequencyParamOption 構造体

## 6.15 UHF\_SetRFTAGComParam メソッド

RF タグの通信関連パラメータを書き込みます。

```
int UHF_SetRFTAGComParam(
    UHF_ParamKind paramKind,
    UHF_RFTAGComParamOption RFTAGComParamOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command     0 : コマンドモード用パラメータ Param_Auto           1 : 自動読み取りモード用パラメータ Param_Flash          2 : FLASH データ
RFTAGComParamOpt	RF タグの通信関連パラメータのオプションをセットします。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFSetRFTAGComParam

e.InputData

6 バイト目以降に RF タグの通信関連パラメータのオプションがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【例外】

ArgumentOutOfRangeException	
RFTAGComParamOpt.InventoryRetryCount	指定可能な値の範囲は 0～15 です。
RFTAGComParamOpt.ReadWriteRetryCount	指定可能な値の範囲は 0～15 です。

### 【特記事項】

- ・ 戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

RFTAGComParamOpt : UHF\_RFTAGComParamOption のデフォルト値

### 【参照】

UHF\_GetRFTAGComParam メソッド、UHF\_RFTAGComParamOption 構造体

## 6.16 UHF\_SetEPCParam メソッド

EPC(UII)関連パラメータを書き込みます。

```
int UHF_SetEPCParam(
    UHF_ParamKind paramKind,
    UHF_EPCParamOption EPCParamOpt);
```

### 【パラメータ】

値	説明
paramKind	パラメータ種類を指定します。 Param_Command    0 : コマンドモード用パラメータ Param_Auto        1 : 自動読み取りモード用パラメータ Param_Flash       2 : FLASH データ
EPCParamOpt	EPC(UII)関連パラメータのオプションをセットします。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFSetEPCParam

e.InputData

6 バイト目以降に EPC(UII)関連パラメータのオプションがセットされます。

設定値は全て出力パラメータにセットされますので、出力パラメータを参照してください。

### 【特記事項】

- ・戻り値が ACK 応答以外の場合、出力パラメータには以下の値がセットされます。  
プログラム内で使用している変数を出力パラメータにセットすると、ACK 応答以外の場合  
は変数が以下の値に上書きされますのでご注意ください。

EPCParamOpt : UHF\_EPCParamOption のデフォルト値

### 【参照】

UHF\_GetEPCParam メソッド、UHF\_EPCParamOption 構造体

## 6.17 UHF\_SetAccessPassword メソッド

リーダーライタに Access パスワードを設定します。

```
int UHF_SetAccessPassword(byte[] password)
```

### [パラメータ]

値	説明
password	アクセスパスワードを指定します。 配列長 4 のバイト配列で指定します。 MSB ファーストでセットします。

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHF\_SetAccessPassword

### [例外]

<b>ArgumentOutOfRangeException</b>	
password	配列長が 4 ではありません。
<b>ArgumentNullException</b>	
password	null を指定することはできません。

### [特記事項]

UHF\_SetAccessPassword は、リーダーライタに Access パスワードを設定する関数です。

UHF\_SetAccessPassword で 0 以外のパスワードを設定すると、以下のコマンド実行時にタグに対して Access コマンドが発行され、パスワードでロックされたタグへのアクセスが可能になります。

- UHF\_Read
  - UHF\_Write
  - UHF\_Lock
  - UHF\_BlockWrite
- リーダライタに UHF\_SetAccessPassword() 関数でアクセスパスワードを設定すると、Access コマンドに対応していない RF タグへのデータの読み書きができなくなります。Access コマンドを使用する必要のない場合は、必ず 0 を設定するようにしてください。
- Access コマンドの発行がなくなった時点で、必ず 0 を設定するようにしてください。0 にしない限りパスワードでロックされた RF タグに対してアクセス可能なままの状態となります。
- コマンドモード用パラメータです。FLASH へは書き込まれませんので、リスタートや電源を [OFF→ON] すると 0 に戻ります。

---

---

## 第7章 RF タグとの通信

本章では、RF タグとの通信に対応した関数について説明します。

---

---

## 7.1 UHF\_InventoryCmd メソッド

インベントリ処理を行い、RF タグの EPC(UII)を読み取ります。

事前に[UHF\_SetSelectParam メソッド]を実行してマスク条件を設定することで、複数の RF タグの中から特定の RF タグだけを選択して処理を行うこともできます。

```
int UHF_InventoryCmd();
```

```
int UHF_InventoryCmd(uint timeout);
```

### [パラメータ]

値	説明
timeout	コマンドの応答を待機する時間をミリ秒単位で指定します。 指定しない場合は、Timeout プロパティの値が有効になります。

#### • timeout について

本メソッド実行後、ACK 応答または NACK 応答を受けるまでに、timeout 以上の時間何も応答を受けない時間が経過すると、戻り値 2 のタイムアウトが返ります。

timeout の値は、100～65535 までの値で指定します。100 未満の値が指定された場合は、100 に補正され、65535 よりも大きい値が指定された場合は、65535 に補正されてメソッドが実行されます。

[UHF\_SetInventoryParam メソッド]で inventoryOpt.Q の値を大きくしすぎると、正常に動作していても以下のような場合にタイムアウトと判定されます。

- 1 件目の EPC(UII)データが timeout 時間より遅く返る場合
- 2 件目以降にあがってくる EPC(UII)データまたは ACK レスポンスの間隔が timeout 時間より遅くなる場合
- NACK レスポンスが timeout 時間よりも遅くなる場合

[UHF\_SetInventoryParam メソッド]で inventoryOpt.Q=10 以上で実行した場合、timeout=1000 (Timeout プロパティ初期値) に設定した場合でも timeout になる可能性があります。

(RF タグの枚数が少ない場合、NACK 応答の場合などタイムアウトする場合があります)

このような場合、メソッドの戻り値は「2: タイムアウト」となりますが、その後もリーダーライタが処理を継続している場合は ResponseRFID イベントが発生しますので、必要に応じて適切な timeout 時間を設定してください。

また、ACK または NACK の ResponseRFID イベントを受ける前に戻り値がタイムアウトになった場合、その後も指定時間内は ResponseRFID イベントを待つなど、必要に応じてご検討ください。

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

- 本メソッドでは、[RF タグ読み取りデータ]を返すイベントと、[RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK]を返すイベントが、別々に発生します。
- [RF タグ読み取りデータ]を返すイベントは、RF タグの読み取り枚数が 1 以上の場合に、読み取った RF タグの枚数と同じ数だけ発生します。
- [RF タグ読み取りデータ]を返すイベントと、[RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK]を返すイベントは、e.SendCommand で区別することができます。
- RF タグが 1 枚も読み取りできなかった場合には、[RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK]を返すイベントのみ発生します。

### [RF タグ読み取りデータ]を返す場合の ResponseRFID イベントパラメータ

#### e.SendCommand

RFID\_SendCommand.UHFInventoryCmd

#### e.PC

読み取った RF タグの PC データ

※MSB ファーストでセットされます。

RF タグのメモリは、ビットアドレスの小さいほうが MSB となります。

#### e.EPC

読み取った RF タグの EPC(UII)データ

※MSB ファーストでセットされます。

RF タグのメモリは、ビットアドレスの小さいほうが MSB となります。

#### e.RSSI

RF タグを読み取った時の RSSI 値を 10 倍した値が、int 型でセットされます。

(例) RSSI 値が-41.7 の場合、e.RSSI には-417 がセットされます。

### [RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK]を返す場合の ResponseRFID イベントパラメータ

#### e.SendCommand

RFID\_SendCommand. UHFInventoryCmdCount

#### e.BinaryData

RF タグ読取枚数 (2 バイトのデータを 16bit の数値として参照。先頭が下位バイト)、および、読み取り時のキャリアのチャンネル番号 (1 バイト)

<注意事項> 「インベントリタイムアウト時間」を設定した場合のレスポンス

リーダーライタの FLASH 設定で「インベントリタイムアウト時間」を設定した場合、「インベントリタイムアウト時間」が経過するまでの間、Inventory コマンドが連続実行され、同じ RF タグであっても読み取りするたびに[RF タグ読み取りデータ]が返ります。

[RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK]は、「インベントリタイムアウト時間」を経過した場合に、「RF タグの読み取り回数および読み取り時のキャリアのチャンネル番号を含む ACK」として、1 回のみ返ります。

※「インベントリタイムアウト時間」は、リーダーライタの ROM バージョンが 1.090 以降の場合に設定できます。

### [参照]

UHF\_SetInventoryParam メソッド、UHF\_SetSelectParam メソッド、

UHF\_SetExpandSelectParam メソッド、UHF\_MemBank 列挙体、Timeout プロパティ

## 7.2 UHF\_InventoryReadCmd メソッド

インベントリ処理を行い、RF タグの EPC(UII)と指定メモリバンクのデータを読み取ります。事前に[UHF\_SetSelectParam メソッド]を実行してマスク条件を設定することで、複数の RF タグの中から特定の RF タグだけを選択して処理を行うこともできます。

```
int UHF_InventoryReadCmd( UHF_MemBank membankOpt,
                          uint startWordAddr,
                          byte wordCount,
                          bool tidRead);
```

```
int UHF_InventoryReadCmd( UHF_MemBank membankOpt,
                          uint startWordAddr,
                          byte wordCount,
                          bool tidRead,
                          uint timeout);
```

### [パラメータ]

値	説明
membankOpt	読み取るメモリバンクの MemBank オプションを指定します。
startWordAddr	MemBank で指定した領域の読み取り開始ワードアドレスを指定します。
wordCount	MemBank で指定した領域の読み取りワード数を指定します。 wordCount=0 を指定した場合、指定した MemBank の全領域を読み取ります。
tidRead	MemBank で指定した領域に加えて TID も読み取る場合は true を指定します。
timeout	コマンドの応答を待機する時間をミリ秒単位で指定します。 指定しない場合は、Timeout プロパティの値が有効になります。

#### • timeout について

本メソッド実行後、ACK 応答または NACK 応答を受けるまでに、timeout 以上の時間何も応答を受けない時間が経過すると、戻り値 2 のタイムアウトが返ります。

timeout の値は、100~65535 までの値で指定します。100 未満の値が指定された場合は、100 に補正され、65535 よりも大きい値が指定された場合は、65535 に補正されてメソッドが実行されます。

[UHF\_SetInventoryParam メソッド]で inventoryOpt.Q の値を大きくしすぎると、正常に動作していても以下のような場合にタイムアウトと判定されます。

- 1 件目のタグデータが timeout 時間より遅く返る場合
- 2 件目以降にあがってくるタグデータまたは ACK レスポンスの間隔が timeout 時間より遅くなる場合
- NACK レスポンスが timeout 時間よりも遅くなる場合

※RF タグがレスポンスを返すスロットはランダムですので、本メソッドの timeout 時間は、[RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK]が返るまでの時間を下回らないように設定してください。

**[UHF\_SetInventoryParam メソッド]で inventoryOpt.Q=10 以上で実行した場合、timeout=1000 (Timeout プロパティ初期値) に設定した場合でも timeout になる可能性があります。**

**(RF タグの枚数が少ない場合、NACK 応答の場合などタイムアウトする場合があります)**



このような場合、メソッドの戻り値は「2:タイムアウト」となりますが、その後もリーダライタが処理を継続している場合は **ResponseRFID** イベントが発生しますので、必要に応じて適切な **timeout** 時間を設定してください。

また、**ACK** または **NACK** の **ResponseRFID** イベントを受ける前に戻り値がタイムアウトになった場合、その後も指定時間内は **ResponseRFID** イベントを待つなど、必要に応じてご検討ください。

#### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

#### [レスポンス]

- 本メソッドでは、**[RF タグ読み取りデータ]**を返すイベントと、**[RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK]**を返すイベントが、別々に発生します。
- **[RF タグ読み取りデータ]**を返すイベントは、**RF タグの読み取り枚数が 1 以上の場合に、読み取った RF タグの枚数と同じ数だけ発生します。**
- **[RF タグ読み取りデータ]**を返すイベントと、**[RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK]**を返すイベントは、**e.SendCommand** で区別することができます。
- **RF タグが 1 枚も読み取りできなかった場合には、[RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK]を返すイベントのみ発生します。**

#### [RF タグ読み取りデータ]を返す場合の ResponseRFID イベントパラメータ

##### e.SendCommand

RFID\_SendCommand. UHFInventoryReadCmd

##### e.PC

読み取った RF タグの PC データ

※MSB ファーストでセットされます。

RF タグのメモリは、ビットアドレスの小さいほうが MSB となります。

##### e.EPC

読み取った RF タグの EPC(UII)データ

※MSB ファーストでセットされます。

RF タグのメモリは、ビットアドレスの小さいほうが MSB となります。

##### e.RSSI

RF タグを読み取った時の RSSI 値を 10 倍した値が、int 型でセットされます。

(例) RSSI 値が-41.7 の場合、e.RSSI には-417 がセットされます。

##### e.MemBankData

読み取った RF タグの MemBank データ

※MSB ファーストでセットされます。

RF タグのメモリは、ビットアドレスの小さいほうが MSB となります。

##### e.TID

読み取った RF タグの TID データ

tidRead=true で実行した場合のみセットされます。

※MSB ファーストでセットされます。

RF タグのメモリは、ビットアドレスの小さいほうが MSB となります。

[RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK]を返す場合の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFInventoryReadCmdCount

e.BinaryData

RF タグ読取枚数 (2 バイトのデータを 16bit の数値として参照。先頭が下位バイト)、  
および、読み取り時のキャリアのチャンネル番号 (1 バイト)

<注意事項> 「インベントリタイムアウト時間」を設定した場合のレスポンス

リーダーライタの FLASH 設定で「インベントリタイムアウト時間」を設定した場合、「インベントリタイムアウト時間」が経過するまでの間、Inventory コマンドが連続実行され、同じ RF タグであっても読み取りするたびに [RF タグ読み取りデータ] が返ります。

[RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK] は、「インベントリタイムアウト時間」を経過した場合に、「RF タグの読み取り回数および読み取り時のキャリアのチャンネル番号を含む ACK」として、1 回のみ返ります。

※ 「インベントリタイムアウト時間」は、リーダーライタの ROM バージョンが 1.090 以降の場合に設定できます。

[例外]

ArgumentOutOfRangeException	
wordCount	指定可能な値の範囲は 1～32 です。

[参照]

UHF\_SetInventoryParam メソッド、UHF\_SetSelectParam メソッド、  
UHF\_SetExpandSelectParam メソッド、UHF\_MemBank 列挙体、Timeout プロパティ

## 7.3 UHF\_Read メソッド

メモリバンクとアドレスを指定し、RF タグのデータをワード単位で読み取ります。

```
int UHF_Read(UHF_MemBank membankOpt,
             uint startWordNo,
             byte wordCount);
```

### [パラメータ]

値	説明
membankOpt	読み取るメモリバンクの MemBank オプションを指定します。
startWordNo	MemBank で指定した領域の読み取り開始ワード番号を指定します。
wordCount	MemBank で指定した領域の読み取りワード数を指定します。 wordCount=0 を指定した場合、指定した MemBank の全領域を読み取ります。

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFRead

e. MemBankData

読み取った RF タグの MemBank データ

※MSB ファーストでセットされます。

RF タグのメモリは、ビットアドレスの小さいほうが MSB となります。

### [例外]

ArgumentOutOfRangeException	
wordCount	指定可能な値の範囲は 1~32 です。

### [参照]

UHF\_SetInventoryParam メソッド、UHF\_MemBank 列挙体

## 7.4 UHF\_Write メソッド

メモリバンクとアドレスを指定し、RF タグにワード単位でデータを書き込みます。

```
int UHF_Write(UHF_MemBank membankOpt,
              uint writeWordNo,
              byte[] writeData);
```

### 【パラメータ】

値	説明
membankOpt	書き込みを行うメモリバンクの MemBank オプションを指定します。
writeWordNo	MemBank で指定した領域の、書き込みを行うワード番号を指定します。
writeData	書き込みデータを指定します。 配列長 2 のバイト配列で指定します。 MSB ファーストでセットします。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand. UHFWrite

### 【例外】

<b>ArgumentNullException</b>	
writeData	null を指定することはできません。
<b>ArgumentOutOfRangeException</b>	
writeData	配列長が 2 ではありません。

### 【参照】

UHF\_SetInventoryParam メソッド、UHF\_MemBank 列挙体

## 7.5 UHF\_BlockWrite メソッド

メモリバンクとアドレスを指定し、RF タグに連続する複数ワードのデータを書き込みます。

```
int UHF_BlockWrite(UHF_MemBank membankOpt,
                  bool blockwriteuse,
                  uint startWordAddr,
                  ushort wordCount,
                  byte[] writeData);
```

### [パラメータ]

値	説明
membankOpt	書き込みを行うメモリバンクの MemBank オプションを指定します。
blockwriteuse	BlockWrite コマンドを使用して書き込む場合は true を指定します。 ※false を指定した場合は、Write コマンドを wordCount の値に応じて複数回実行します。
startWordAddr	MemBank で指定した領域の書き込み開始ワード番号を指定します。
wordCount	MemBank で指定した領域の書き込みワード数を指定します。
writeData	書き込みデータを指定します。MSB ファーストでセットします。 wordCount×2 の配列長となるバイト配列で指定します。 例) 2ワード書き込む場合  1 バイト目: 1ワード目の上位バイト 2 バイト目: 1ワード目の下位バイト 3 バイト目: 2ワード目の上位バイト 4 バイト目: 2ワード目の下位バイト

※BlockWrite コマンドは ISO18000-63 では RF タグのオプションコマンドのため、一部の RF タグでは対応していません。対応/非対応は、UTR 通信プロトコル説明書の「4.2.3 RF タグオプションコマンド対応表」、または使用する RF タグのデータシートを参照ください。

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand. UHFBlockWrite

### [例外]

ArgumentNullException	
writeData	null を指定することはできません。
ArgumentOutOfRangeException	
writeData	配列長が適切な値(wordCount×2)ではありません。
wordCount	書き込みワード数を 32 ワードを超えて指定することはできません。

### [参照]

UHF\_SetInventoryParam メソッド、UHF\_MemBank 列挙体

## 7.6 UHF\_Lock メソッド

RF タグのメモリやパスワードをロックします。  
EPC(UII)、TID、User はライトロックすることができます。  
Access パスワード、Kill パスワードはリード/ライトロックすることができます。  
具体的には、以下の処理を実行することができます。

処理対象	ロック	Perma ロック
EPC(UII)領域	Write ロック／解除	Write プロテクト状態のロック
TID 領域	Write ロック	Write プロテクト状態のロック
User 領域	Write ロック／解除	Write プロテクト状態のロック
Access パスワード	Read/Write ロック／解除	Read/Write プロテクト状態のロック
Kill パスワード	Read/Write ロック／解除	Read/Write プロテクト状態のロック

※TID は、通常は ReadOnly 領域のため、本メソッドを実行しても状態は変わりません。

```
int UHF_Lock(UHF_EPCLockOption epcLockOpt);

int UHF_Lock(UHF_TIDLKOption tidLockOpt);

int UHF_Lock(UHF_UserLockOption userLockOpt);

int UHF_Lock(UHF_AccessPwdLockOption accessLockOpt);

int UHF_Lock(UHF_KillPwdLockOption killLockOpt);

int UHF_Lock(UHF_EPCLockOption epcLockOpt,
             UHF_UserLockOption userLockOpt);

int UHF_Lock(UHF_AccessPwdLockOption accessLockOpt,
             UHF_KillPwdLockOption killLockOpt);

int UHF_Lock(UHF_EPCLockOption epcLockOpt,
             UHF_TIDLKOption tidLockOpt,
             UHF_UserLockOption userLockOpt,
             UHF_AccessPwdLockOption accessLockOpt,
             UHF_KillPwdLockOption killLockOpt);
```

### [パラメータ]

値	説明
epcLockOpt	EPC(UII)領域の Lock 処理用オプションを指定します。 本パラメータを省略すると、全て false がセットされます。
tidLockOpt	TID 領域の Lock 処理用オプションを指定します。 本パラメータを省略すると、全て false がセットされます。
userLockOpt	User 領域の Lock 処理用オプションを指定します。 本パラメータを省略すると、全て false がセットされます。
accessLockOpt	Access パスワードの Lock 処理用オプションを指定します。 本パラメータを省略すると、全て false がセットされます。
killLockOpt	Kill パスワードの Lock 処理用オプションを指定します。 本パラメータを省略すると、全て false がセットされます。

**【戻り値】**

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

**【レスポンス】**

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand.UHFlock

**【参照】**

UHF\_EPCLockOption 構造体、UHF\_TIDLlockOption 構造体、  
UHF\_UserLockOption 構造体、UHF\_AccessPwdLockOption 構造体、  
UHF\_KillPwdLockOption 構造体

## 7.7 UHF\_Kill メソッド

タグをキル（無効化）するメソッドです。

タグをキル（無効化）するためには、タグ（Bank00）Reserved 領域のキルパスワード（アドレス 00h から 4 バイト）が設定された状態で、そのパスワードと本メソッドで指定するキルパスワードが一致する必要があります。

タグへ設定するキルパスワードは、「UHF\_Write」もしくは「UHF\_BlockWrite」メソッドを使用して行います。

### 【パラメータ】

値	説明
password	Kill パスワードはタグ（Bank00）Reserved 領域のキルパスワード（アドレス 00h から 4 バイト）を指定します。 MSB ファーストでセットします。

### 【戻り値】

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### 【レスポンス】

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand.UHFKill

### 【例外】

<b>ArgumentNullException</b>	
password	null を指定することはできません。
<b>ArgumentOutOfRangeException</b>	
password	配列長が 4 ではありません。

## 7.8 UHF\_BlockErase メソッド

RF タグの連続する複数ワードのデータを消去します。  
MemBank と、消去するアドレスおよびワード数を指定して実行します。

※リーダライタの ROM バージョン 1.080 以降対応のメソッドです。

```
int UHF_BlockErase(UHF_MemBank membankOpt,
                  uint startWordAddr,
                  ushort wordCount);
```

### [パラメータ]

値	説明
membankOpt	消去するメモリバンクの MemBank オプションを指定します。
startWordAddr	MemBank で指定した領域の消去開始ワード番号を指定します。
wordCount	MemBank で指定した領域の消去ワード数を指定します。

※BlockErase コマンドは ISO18000-63 では RF タグのオプションコマンドで、一部の RF タグでは対応していません。対応/非対応は、UTR 通信プロトコル説明書の「4.2.3 RF タグオプションコマンド対応表」、または使用する RF タグのデータシートを参照ください。

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ

```
e.SendCommand
  RFID_SendCommand. UHFBlockErase
```

### [参照]

UHF\_SetInventoryParam メソッド、UHF\_MemBank 列挙体

## 7.9 UHF\_BlockWrite2 メソッド

メモリバンクとアドレスを指定し、RFタグに連続して複数ワードのデータを書き込むメソッドです。BlockWriteコマンドを複数回連続で送信するような大容量データを書き込む場合、本メソッドを使用することで、処理時間を短くすることができます。

※リーダライタのROMバージョン 1.080 以降対応のメソッドです。

```
int UHF_BlockWrite2(UHF_MemBank membankOpt,
                   bool writeContinue,
                   uint startWordAddr,
                   ushort wordCount,
                   byte[] writeData);
```

### [パラメータ]

値	説明
membankOpt	書き込みを行うメモリバンクの MemBank オプションを指定します。
writeContinue	UHF_BlockWrite2メソッドによる書き込みを継続するかどうかを指定します。 <ul style="list-style-type: none"> <li>連続書き込みを継続する場合は、「false」をセットします。</li> <li>連続書き込みを終了する場合は、「true」をセットします。</li> </ul> <注意事項> 連続書き込みの継続に「false」を指定して、UHF_BlockWrite2メソッドによる書き込みを[継続する]とした場合でも、RFタグへの書き込みに失敗してリーダライタからの応答がNACK応答となった場合、連続書き込みの継続は自動的に終了します。
startWordAddr	MemBank で指定した領域の書き込み開始ワード番号を指定します。
wordCount	MemBank で指定した領域の書き込みワード数を指定します。
writeData	書き込みデータを指定します。 wordCount×2 の配列長となるバイト配列で指定します。 MSB ファーストでセットします。  例) 2ワード書き込む場合 1 バイト目：1ワード目の上位バイト 2 バイト目：1ワード目の下位バイト 3 バイト目：2ワード目の上位バイト 4 バイト目：2ワード目の下位バイト

※BlockWrite コマンドは ISO18000-63 では RF タグのオプションコマンドのため、一部の RF タグでは対応していません。対応/非対応は、UTR 通信プロトコル説明書の「4.2.3 RF タグオプションコマンド対応表」、または使用する RF タグのデータシートを参照ください。

※本メソッドは、高速書き込みを実現するために、[UHF\_BlockWrite メソッド]と内部処理が異なります。そのため、BlockWrite コマンドに対応した RF タグであっても書き込みできない場合があります。あらかじめ書き込み精度の動作確認をおこなった上でご使用ください。

**【戻り値】**

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

**【レスポンス】**

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand. UHFBlockWrite2

**【例外】**

ArgumentNullException	
writeData	null を指定することはできません。
ArgumentOutOfRangeException	
writeData	配列長が適切な値(wordCount×2)ではありません。
wordCount	書き込みワード数を 124 ワードを超えて指定することはできません。

**【参照】**

UHF\_SetInventoryParam メソッド、UHF\_MemBank 列挙体

## 7.10 UHF\_Encode メソッド

RF タグの複数の MemBank に対して、1 回のコマンド送信で複数ワードのデータを書き込むメソッドです。同時に RF タグのメモリをロックすることもできます。

※リーダライタの ROM バージョン 1.080 以降対応のメソッドです。

※「タカヤ独自メソッド」です。

[UHF\_BlockWriteメソッド]、[UHF\_Lockメソッド]、[UHF\_SetAccessPasswordメソッド]の複合メソッドです。

RF送信信号(キャリア)の状態により、内部処理で[SetTransmitSignalメソッド]が実行される場合があります。メソッド実行後は、実行前の送信信号(キャリア)の状態に戻ります。従来は複数のメソッドを送受信する必要があったメソッド群を1つのメソッドとすることで、上位機器とのデータ通信時間を節約し、高速な書き込みが可能です。

```
int UHF_Encode(
    bool blockwriteuse,
    bool keeppassword,
    UHF_EncodeOption encodeOpt,
    ushort reservedWordCount,
    uint reservedStartWordAddr,
    byte[] reservedWriteData,
    ushort epcWordCount,
    uint epcStartWordAddr,
    byte[] epcWriteData,
    ushort userWordCount,
    uint userStartWordAddr,
    byte[] userWriteData,
    UHF_EPCLockOption epcLockOpt,
    UHF_TIDLLockOption tidLockOpt,
    UHF_UserLockOption userLockOpt,
    UHF_AccessPwdLockOption accessLockOpt,
    UHF_KillPwdLockOption killLockOpt);
```

### [パラメータ]

値	説明
blockwriteuse	RF タグへの書き込み時の BlockWrite コマンド使用の有無を指定します。 <ul style="list-style-type: none"> <li>• true を指定した場合、BlockWrite コマンドを使用します。</li> <li>• false を指定した場合、wordCount の値に応じて Write コマンドを複数回実行します。</li> </ul>
keeppassword	メソッド実行前後でのリーダライタの Access Password の保持の有無を指定します。 <ul style="list-style-type: none"> <li>• true を指定した場合、メソッド実行後も、メソッド実行前の Access Password を保持します。</li> <li>• false を指定した場合、メソッド実行後は、reservedWriteData で指定した Access Password となります。</li> </ul> reservedWriteData を指定しない場合には、Access Password は、[00000000]になります。
encodeOpt	RF タグの Lock 処理時に各 MemBank の Lock[する/しない]の指定、および、RF タグの Lock 済みの MemBank にアクセスする際に、Access コマンドを発行[する/しない]を指定するためのオプションです。

(次ページへ続く)

(前ページからの続き)

値	説明
reservedWordCount	Reserved 領域の書き込みワード数を指定します。
reservedStartWordAddr	Reserved 領域の書き込み開始ワード番号を指定します。
reservedWriteData	Reserved 領域の書き込みデータを指定します。 reservedWordCount ×2 の配列長となるバイト配列で指定します。MSB ファーストでセットします。  例) 2 ワード書き込む場合 1 バイト目: 1 ワード目の上位バイト 2 バイト目: 1 ワード目の下位バイト 3 バイト目: 2 ワード目の上位バイト 4 バイト目: 2 ワード目の下位バイト
epcWordCount	EPC 領域の書き込みワード数を指定します。
epcStartWordAddr	EPC 領域の書き込み開始ワード番号を指定します。
epcWriteData	EPC 領域の書き込みデータを指定します。 epcWordCount ×2 の配列長となるバイト配列で指定します。MSB ファーストでセットします。  例) 2 ワード書き込む場合 1 バイト目: 1 ワード目の上位バイト 2 バイト目: 1 ワード目の下位バイト 3 バイト目: 2 ワード目の上位バイト 4 バイト目: 2 ワード目の下位バイト
userWordCount	User 領域の書き込みワード数を指定します。
userStartWordAddr	User 領域の書き込み開始ワード番号を指定します。
userWriteData	User 領域の書き込みデータを指定します。 userWordCount ×2 の配列長となるバイト配列で指定します。MSB ファーストでセットします。  例) 2 ワード書き込む場合 1 バイト目: 1 ワード目の上位バイト 2 バイト目: 1 ワード目の下位バイト 3 バイト目: 2 ワード目の上位バイト 4 バイト目: 2 ワード目の下位バイト
epcLockOpt	EPC(UIT)領域の Lock 処理用オプションを指定します。
tidLockOpt	TID 領域の Lock 処理用オプションを指定します。
userLockOpt	User 領域の Lock 処理用オプションを指定します。
accessLockOpt	Access パスワードの Lock 処理用オプションを指定します。
killLockOpt	Kill パスワードの Lock 処理用オプションを指定します。

**[戻り値]**

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

**[レスポンス]**

ACK 応答時の ResponseRFID イベントパラメータ  
e.SendCommand  
RFID\_SendCommand.UHFEncode

**[例外]**

ArgumentNullException	
reservedWriteData	null を指定することはできません。
epcWriteData	null を指定することはできません。
userWriteData	null を指定することはできません。
ArgumentOutOfRangeException	
reservedWriteData	配列長が適切な値(reservedWordCount×2)ではありません。
epcWriteData	配列長が適切な値(epcWordCount×2)ではありません。
userWriteData	配列長が適切な値(userWordCount×2)ではありません。
resevedWordCount	書き込みワード数を 123 ワードを超えて指定することはできません
epcWordCount	書き込みワード数を 123 ワードを超えて指定することはできません
userWordCount	書き込みワード数を 123 ワードを超えて指定することはできません
全パラメータ	データ長が 255 バイトを超えます。

**[参照]**

UHF\_EncodeOption 構造体、UHF\_EPCLockOption 構造体、UHF\_TIDLlockOption 構造体、  
UHF\_UserLockOption 構造体、UHF\_AccessPwdLockOption 構造体、  
UHF\_KillPwdLockOption 構造体

## 7.11 UHF\_ThroughCmd メソッド

RF タグと直接交信を行います。

リーダライタは、本メソッドの引数に指定したコマンドをそのまま RF タグへ送信します。

```
int UHF_ThroughCmd(UHF_ThroughCmdOption option,
                   ushort rcvLength,
                   ushort sendLength,
                   byte[] command);
```

```
int UHF_ThroughCmd(UHF_ThroughCmdOption option,
                   ushort rcvLength,
                   ushort sendLength,
                   byte[] command,
                   ushort timeout);
```

### [パラメータ]

値	説明
option	UHF_ThroughCmd 専用のオプションを指定します。
rcvLength	RF タグから受信するデータビット長 (Header から CRC まで) を指定します。
sendLength	RF タグへ送信するコマンドデータの有効ビット長を指定します。 自動で Handle を付加する場合や自動で CRC を付加する場合は、それらのデータ長は含めません。
command	RF タグへ送信するコマンドデータをバイトデータとしてセットします。 送信データビット長がバイト単位に収まらず、端数 bit が生じる場合は最終バイトの上位側に詰めてセットします。
timeout	受信タイムアウト時間をミリ秒単位で 1~20 の範囲で指定します。 本フィールドは、パラメータのオプションでコマンド種別=7(受信待ち時間指定)を指定した場合のみ有効になります。 本フィールドを省略した場合は 20 が指定されます。

### [戻り値]

値	説明
0	ACK 応答
1	NACK 応答
2	タイムアウト
-1	送信失敗

### [レスポンス]

ACK 応答時の ResponseRFID イベントパラメータ

e.SendCommand

RFID\_SendCommand.UHFThroughCmd

e.BinaryData

RF タグから受信したデータ (Header から CRC まで)

※MSB ファーストでセットされます。

### [例外]

ArgumentNullException	
command	null を指定することはできません。

**【特記事項】**

- 本メソッドは、RF タグの状態が **Open** 状態、もしくは **Secured** 状態の場合に使用可能です。  
そのため、本メソッドを実行する場合は、以下の手順で処理をおこなう必要があります。
  1. RF 送信信号の制御コマンドを実行し、キャリア出力を **ON** する。(必須処理)
  2. **Inventory** コマンドを以下のパラメータで実行し、RF タグを **Open** 状態とする。  
(必須処理)
    - Q 値の自動 UP/DOWN 機能=使用しない
    - Q 値の開始値=0
    - Q 値の最小値=0
    - Q 値の最大値=0※複数の RF タグから 1 枚を指定して処理をおこなう場合は、事前に **Select** コマンドを実行する設定も必要です。
  3. **UHF\_GetHandle** コマンドを実行して **Handle** データを取得する。(必要に応じて)  
※**Handle** を自動で付加する設定の場合は不要な処理です。
  4. **UHF\_ThroughCmd** を使用して RF タグのカスタムコマンドを実行する。
  5. RF 送信信号の制御コマンドを実行し、キャリア出力を **OFF** する。(推奨処理)  
※不要な電波を出し続けると電波干渉の要因となりますので、処理終了後は直ちにキャリア出力 **OFF** することを推奨します。
  
- **command** のデータ長が 247 バイトより大きい場合は、超過分が破棄されます。

**【参照】**

UHF\_ThroughCmdOption 構造体

---

---

## 第8章 汎用メソッド

本章では、汎用メソッドについて説明します。

---

---

## 8.1 SendData メソッド

リーダーライタへ任意のバイナリデータを送信します。  
SDKにおいて関数化されていないコマンドなど、任意のコマンド列をリーダーライタへ直接送信することができます。

```
int SendData(byte[] command);
```

### 【パラメータ】

値	説明
command	バイナリデータを指定します。

### 【戻り値】

値	説明
0	送信成功
-1	送信失敗

### 【レスポンス】

リーダーライタからレスポンスを受けた場合は、ResponseRFID イベントが発生します。  
ただし、e.SendCommand には Other がセットされます。

### 【例外】

ArgumentNullException	
command	null を指定することはできません。

### 【特記事項】

本メソッドは、リーダーライタからの応答を待機しません。  
バイナリデータの送信完了時点で戻り値が返されます。  
リーダーライタからレスポンスを受ける場合は、上位アプリケーションにて ResponseRFID イベントを待機してください。

## 8.2 ClearSerialInputBuffer メソッド

シリアル受信バッファに含まれるデータをクリアします。

```
bool ClearSerialInputBuffer();
```

### [戻り値]

シリアル受信バッファのクリアに成功した場合は **true**、失敗した場合は **false** を返します。

## 8.3 ClearSerialOutputBuffer メソッド

シリアル送信バッファに含まれるデータをクリアします。

```
bool ClearSerialOutputBuffer();
```

### [戻り値]

シリアル送信バッファのクリアに成功した場合は **true**、失敗した場合は **false** を返します。

---

---

## 第9章 プロパティ

本章では、SDK に含まれるプロパティについて説明します。

---

---

## 9.1 シリアル通信用プロパティ

### 9.1.1 PortState プロパティ/IsOpen プロパティ

COM ポートのオープン状態を取得します。  
COM ポートをオープンしている場合は、true を返します。

```
bool PortState{ get; }  
bool IsOpen{ get; }
```

### 9.1.2 PortNumber プロパティ

COM ポート番号を取得または設定します。  
Open メソッドのパラメータを指定しなかった場合、本プロパティの値を使用して COM ポートをオープンします。

```
int PortNumber{ get; set; }
```

[デフォルト値]

1

### 9.1.3 BaudRate プロパティ

通信速度を取得または設定します。  
設定可能な値は、115200 のみです。

```
uint BaudRate{ get; set; }
```

[デフォルト値]

115200

### 9.1.4 SerialInputBufferSize プロパティ

シリアル通信時の受信バッファサイズを取得または設定します。  
COM ポートオープン中は本プロパティ値を変更することはできません。

```
int SerialInputBufferSize{ get; set; }
```

[デフォルト値]

8192

### 9.1.5 SerialOutputBufferSize プロパティ

シリアル通信時の送信バッファサイズを取得または設定します。  
COM ポートオープン中は本プロパティ値を変更することはできません。

```
int SerialOutputBufferSize{ get; set; }
```

[デフォルト値]

8192

#### 9.1.6 SerialInputBufferByteCount プロパティ

シリアル受信バッファ内のデータサイズ（バイト数）を取得します。  
取得に失敗した場合は-1 を返します。

```
int SerialInputBufferByteCount{ get; }
```

#### 9.1.7 SerialOutputBufferByteCount プロパティ

シリアル送信バッファ内のデータサイズ（バイト数）を取得します。  
取得に失敗した場合は-1 を返します。

```
int SerialOutputBufferByteCount{ get; }
```

#### 9.1.8 SerialConnTimeout プロパティ

COM ポートの通信タイムアウト時間をミリ秒単位で取得または設定します。  
本プロパティは、COM ポート制御時のタイムアウト時間に関するプロパティです。  
COM ポートオープン中は本プロパティ値を変更することはできません。  
設定可能な値の範囲は 100~65535 です。

```
uint SerialConnTimeout{ get; set; }
```

[デフォルト値]

1000

#### 9.1.9 FlowControl プロパティ

シリアル通信時のフロー制御方式を取得または設定します。  
ポートオープンのパラメータとして使用するプロパティです。  
設定可能な値は NONE（無手順）、RTSCTS（RTS/CTS フロー制御）のいずれかです。  
RTSCTS は一部の機種のみサポートしていますので、対応可否はリーダーライタの仕様書をご参照ください。  
リーダーライタがサポートしていない値を設定すると通信できませんのでご注意ください。

```
RFID_FlowControl FlowControl{get; set;}
```

[デフォルト値]

NONE

## 9.2 TCP/IP 通信用プロパティ

### 9.2.1 Connected プロパティ

ソケットのオープン状態を取得します。  
ソケットをオープンしている場合は、true を返します。

```
bool Connected{ get; }
```

### 9.2.2 RemoteHost プロパティ

接続先のホスト名を取得または設定します。  
Connect メソッドのパラメータを指定しなかった場合、本プロパティの値を使用してソケットをオープンします。

```
string RemoteHost{ get; set; }
```

[デフォルト値]

127.0.0.1

### 9.2.3 RemotePort プロパティ

接続に使用するポート番号を取得または設定します。  
Connect メソッドのパラメータを指定しなかった場合、本プロパティの値を使用してソケットをオープンします。

```
int RemotePort{ get; set; }
```

[デフォルト値]

9004

### 9.2.4 SocketInputBufferSize プロパティ

ソケット通信時の受信バッファサイズを取得または設定します。  
ソケットオープン中は本プロパティ値を変更することはできません。

```
int SocketInputBufferSize{ get; set; }
```

[デフォルト値]

8192

### 9.2.5 SocketOutputBufferSize プロパティ

ソケット通信時の送信バッファサイズを取得または設定します。  
ソケットオープン中は本プロパティ値を変更することはできません。

```
int SocketOutputBufferSize{ get; set; }
```

[デフォルト値]

8192

### 9.2.6 SocketInputBufferByteCount プロパティ

ソケット受信バッファ内のデータサイズ (バイト数) を取得します。  
取得に失敗した場合は-1 を返します。

```
int SocketInputBufferByteCount{ get; }
```

### 9.2.7 SocketConnTimeout プロパティ

ソケットオープン時の通信タイムアウト時間をミリ秒単位で取得または設定します。  
ソケットオープン中は本プロパティ値を変更することはできません。  
設定可能な値の範囲は 100~65535 です。

```
uint SocketConnTimeout{ get; set; }
```

[デフォルト値]

10000

## 9.3 共通プロパティ

### 9.3.1 Timeout プロパティ

通信のタイムアウト時間（コマンド送信後、リーダーライタからの応答を待機する時間）をミリ秒単位で設定または取得します。  
設定可能な値の範囲は 100～65535 です。

```
uint Timeout{ get; set; }
```

[デフォルト値]

1000

---

---

# 第10章 データ型

本章では、SDK に含まれるデータ型について説明します。

---

---

## 10.1 列挙体

### 10.1.1 RFID\_BaudRate

リーダライタとの通信速度を定義します。

定数名	値	説明
BaudRate115200	0x03	115200bps

### 10.1.2 RFID\_FlowControl

シリアル通信のフロー制御方式を定義します。

定数名	値	説明
NONE	0x00	無手順(フロー制御なし)
RTSCTS	0x01	RTS/CTS フロー制御

### 10.1.3 RFID\_LEDMode

LED の点灯モードを定義します。

定数名	値	説明
AppointTime	0x00	指定時間の点灯
Blink	0x01	常時点滅
Always	0x02	常時点灯または消灯

### 10.1.4 RFID\_ProductSeries

製品シリーズを定義します。

定数名	値	説明
UTRS01	0x04	UTR 据置型 特定小電力タイプ

### 10.1.5 RFID\_ScanMode

リーダライタ動作モードを定義します。

定数名	値	説明
CommandScanMode	0x00	コマンドモード
UHFInventoryMode	0x65	UHF 連続インベントリモード
UHFInventoryReadMode	0x66	UHF 連続インベントリリードモード

### 10.1.6 RFID\_TransmitSignal

RF 送信信号 (キャリア) の設定を定義します。

定数名	値	説明
Off	0x00	送信キャリア OFF
On	0x01	送信キャリア ON
Reset	0x02	一度 OFF し、ON に戻します。

### 10.1.7 RFID\_AntennaSwitching

リーダライタのアンテナ切替方式を定義します。

自動読み取りモード時に、アンテナを自動的に切り替える方法を指定します。  
パラメータによりアンテナの切替の順番、レスポンスの順番及び内容が異なります。

定数名	値	説明
NoControl	0x00	制御しない
Control	0x01	制御する
SpecialControl	0x02	制御する(複数のアンテナを一つのアンテナとして扱う)

- NoControl : 「0x00 : 制御しない」

選択されたアンテナ 1 台のみを使用してキャリア出力を行います。  
複数のアンテナを「使用する」に設定していても、同時に選択されているアンテナは 1 台のみとなります。

※選択されたアンテナを切り替える際には、[コマンドモード用パラメータ]に、「使用する」アンテナを書き込むことを行います。

- Control : 「0x01 : 制御する」

「使用する」に設定されたアンテナを、アンテナ番号の小さい順に順次切り替えてキャリア出力を行います。

- SpecialControl : 「0x02 : 制御する (複数のアンテナを一つのアンテナとして扱う)」

「使用する」に設定されたアンテナを、1 つのアンテナと見立ててアンテナ番号の小さい順に順次切り替えてキャリア出力を行います。  
1 つのアンテナと見立てていますので、読み取り枚数等のレスポンスは最後のアンテナの読み取りの後に返ります。

#### <注意事項>

アンテナの切替制御を行う場合、キャリア OFF → キャリアセンス → キャリア ON の処理が入るため、読み取りが遅くなります。

また、キャリアの周波数は、設定された一連のアンテナ切替が終了するまで、同じ周波数となります。ただし、キャリアセンスにかかった場合はアンテナ切替の途中でも周波数が切り替わります。

### 10.1.8 RFID\_FrequencyChannelKounai

構内無線局優先の周波数設定のオプションを定義します。

定数名	値	説明
ChNone	0x00	指定なし
Ch5	0x01	CH5 916.8MHz
Ch11	0x02	CH11 918.0MHz
Ch17	0x04	CH17 919.2MHz
Ch23	0x08	CH23 920.4MHz
Ch24	0x10	CH24 920.6MHz
Ch25	0x20	CH25 920.8MHz
ChALL	0x3F	構内無線局優先の全周波数

### 10.1.9 RFID\_FrequencyChannelTokushou

特定小電力無線局優先の周波数設定のオプションを定義します。

定数名	値	説明
ChNone	0x00	指定なし
Ch26	0x40	CH26 921.0MHz
Ch27	0x80	CH27 921.2MHz
Ch28	0x01	CH28 921.4MHz
Ch29	0x02	CH29 921.6MHz
Ch30	0x04	CH30 921.8MHz
Ch31	0x08	CH31 922.0MHz
Ch32	0x10	CH32 922.2MHz
ChALL	0xDF	特定小電力無線局優先の全周波数

### 10.1.10 RFID\_FrequencyChannelActive

アクティブタグ優先の周波数設定のオプションを定義します。

定数名	値	説明
ChNone	0x00	指定なし
Ch33	0x40	CH33 922.4MHz
Ch34	0x80	CH34 922.6MHz
Ch35	0x01	CH35 922.8MHz
Ch36	0x02	CH36 923.0MHz
Ch37	0x04	CH37 923.2MHz
ChALL	0xE3	アクティブタグ優先の全周波数

### 10.1.11 RFID\_FrequencyChannel

周波数設定のオプションを定義します。

定数名	値	説明
ChNone	0x00	指定なし
Ch5	0x05	CH5 916.8MHz
Ch11	0x0B	CH11 918.0MHz
Ch17	0x11	CH17 919.2MHz
Ch23	0x17	CH23 920.4MHz
Ch24	0x18	CH24 920.6MHz
Ch25	0x19	CH25 920.8MHz
Ch26	0x1A	CH26 921.0MHz
Ch27	0x1B	CH27 921.2MHz
Ch28	0x1C	CH28 921.4MHz
Ch29	0x1D	CH29 921.6MHz
Ch30	0x1E	CH30 921.8MHz
Ch31	0x1F	CH31 922.0MHz
Ch32	0x20	CH32 922.2MHz
Ch33	0x21	CH33 922.4MHz
Ch34	0x22	CH34 922.6MHz
Ch35	0x23	CH35 922.8MHz
Ch36	0x24	CH36 923.0MHz
Ch37	0x25	CH37 923.2MHz

### 10.1.12 RFID\_FrequencyScanMode

周波数のスキャンモードを定義します。

リーダライタがキャリアを出力する際には、出力を開始しようとしている周波数で、他の機器がキャリアを出していないかを確認します。これをキャリアセンスと言います。電波法の規定上、リーダライタは、一定時間(5ms)以上のキャリアセンスを行ってからキャリアの出力を行います。キャリアセンスを行った際に、他の機器のキャリアが検出された場合の動作は、周波数のスキャンモードにより異なり、以下の動作を行います。

定数名	値	説明
FrequecneyFix	0x00	指定周波数固定
FrequencyHopping	0x01	周波数ホッピング有効
CarrierSensePriority	0x02	キャリアセンス優先

- **FrequecneyFix**

「指定周波数固定」の場合

キャリアセンスにかかった場合、その周波数で引き続きキャリアセンスを繰り返します。一定時間 (5ms 以上) キャリアが検出されなくなるまで待ち、キャリアの出力を行います。

- **FrequencyHopping**

「周波数ホッピング有効」の場合

キャリアセンスにかかった場合、「周波数設定の書き込み」で使用許可された周波数の中から周波数をランダムに切替、キャリアセンスを再度行います。切り替えた先でさらにキャリアセンスにかかった場合には、引き続き周波数をランダムに切り替えてキャリアセンスを繰り返します。また、キャリアの出力を開始しようとするタイミングで、周波数が毎回ランダムに切り替わります。

- **CarrierSensePriority**

「キャリアセンス優先」の場合

キャリアセンスにかかった場合、「周波数設定の書き込み」で使用許可された周波数の中から周波数をランダムで切替、キャリアセンスを再度行います。切り替えた先でさらにキャリアセンスにかかった場合には、引き続き周波数をランダムで切り替えてキャリアセンスを繰り返します。また、次にキャリアの出力を開始しようとする場合には、前回キャリアを出力した周波数でキャリアセンスを行います。

### 10.1.13 RFID\_SendCommand

コマンドの種類を定義します。

ResponseRFID イベントの SendCommand プロパティにはこの値がセットされます。

定数名は、各メソッド名と同名です。

例：Inventory メソッドのレスポンスは、RFID\_SendCommand.Inventory

なお、対応するメソッドのない不明なレスポンスは、RFID\_SendCommand.Other がセットされます。

### 10.1.14 RFID\_UseBuzzer

リーダライタ動作モード設定時のブザーに関するオプションを定義します。

定数名	値	説明
Unuse	0x00	鳴らさない
Use	0x01	鳴らす

### 10.1.15 UHF\_ParamKind

パラメータ種類を定義します。

定数名	値	説明
Param_Command	0x00	コマンドモード用パラメータ
Param_Auto	0x01	自動読取モード用パラメータ
Param_Flash	0x02	FLASH データ用パラメータ

10.1.16 UHF\_Action

Select コマンドで使用する Action 値(3bit)を定義します。

定数名	値	説明
Act_000b	0x00	<マスク条件 一致の場合> Inventoried フラグを A にセット、または SL フラグをセット
		<マスク条件 不一致の場合> Inventoried フラグを B にセット、または SL フラグをリセット
Act_001b	0x01	<マスク条件 一致の場合> Inventoried フラグを A にセット、または SL フラグをセット
		<マスク条件 不一致の場合> なにもしない
Act_010b	0x02	<マスク条件 一致の場合> なにもしない
		<マスク条件 不一致の場合> Inventoried フラグを B にセット、または SL フラグをリセット
Act_011b	0x03	<マスク条件 一致の場合> Inventoried フラグを反転、または SL フラグを反転
		<マスク条件 不一致の場合> なにもしない
Act_100b	0x04	<マスク条件 一致の場合> Inventoried フラグを B にセット、または SL フラグをリセット
		<マスク条件 不一致の場合> Inventoried フラグを A にセット、または SL フラグをセット
Act_101b	0x05	<マスク条件 一致の場合> Inventoried フラグを B にセット、または SL フラグをリセット
		<マスク条件 不一致の場合> なにもしない
Act_110b	0x06	<マスク条件 一致の場合> なにもしない
		<マスク条件 不一致の場合> Inventoried フラグを A にセット、または SL フラグをセット
Act_111b	0x07	<マスク条件 一致の場合> なにもしない
		<マスク条件 不一致の場合> Inventoried フラグを反転、または SL フラグを反転

**[Action 値説明]**

マスク条件が一致した RF タグに対し、Target 値で指定したフラグの状態をどのように変化させるかを指定するパラメータです。

### 10.1.17 UHF\_DR

Inventory 処理で使用する DR 値(1bit)を定義します。

定数名	値	説明
DR8	0x00	0b : 8
DR64_3	0x01	1b : 64/3

#### [DR 値説明]

RF タグからの応答で使用するサブキャリア周波数を指定します。

### 10.1.18 UHF\_M

Inventory 処理で使用する M 値(2bit)を定義します。

定数名	値	説明
M1	0x00	00b : FM0
M2	0x01	01b : M2
M4	0x02	10b : M4
M8	0x03	11b : M8

#### [M 値説明]

RF タグからの応答信号の符号化方式を指定します。

### 10.1.19 UHF\_MemBank

Inventory 処理および Select コマンドで使用する MemBank(2bit)を定義します。

定数名	値	説明
Reserved	0x00	00b : Reserved
EPC	0x01	01b : EPC(UII)
TID	0x02	10b : TID
User	0x03	11b : User

#### [MemBank 説明]

処理の対象 (Select コマンドのマスク対象、UHF\_Read の読み取り対象など) となるメモリバンクを指定します。

### 10.1.20 UHF\_Session

Inventory 処理で使用する Session 値(2bit)を定義します。

定数名	値	説明
S0	0x00	00b : S0
S1	0x01	01b : S1
S2	0x02	10b : S2
S3	0x03	11b : S3

#### [Session 値説明]

Inventory 処理の対象となる Session を指定します。

Inventory 処理された RF タグは、指定された Session の Inventoried フラグを A から B、または B から A に変更します。

※S0 を指定した場合、読み取り後に RF タグをアンテナから外せば Inventoried フラグはすぐに A に戻ります。

※S2 を指定した場合、読み取り後に RF タグを一定時間以上アンテナから外しておかないと Inventoried フラグは A に戻りません。

### 10.1.21 UHF\_Sel

Inventory 処理で使用する Sel 値(2bit)を定義します。

定数名	値	説明
ALL_00b	0x00	00b : ALL
ALL_01b	0x01	01b : ALL
NotSL	0x02	10b : ~SL
SL	0x03	11b : SL

#### [Sel 値説明]

インベントリ処理の対象となる RF タグの SL フラグステータスを指定します。

1 枚の RF タグだけを対象とする場合など、事前の UHF\_Select 処理で SL フラグをセットした場合は「SL」を指定します。

複数の RF タグを対象として処理を行う場合、通常は「ALL\_00b」を指定します。

### 10.1.22 UHF\_InventoryTarget

Inventory 処理で使用する Target 値(1bit)を定義します。

定数名	値	説明
InventoryA	0x0	0b : InventoryA
InventoryB	0x1	1b : InventoryB

#### [Inventory 処理の Target 値説明]

Inventory 処理の対象となるフラグを指定します。

RF タグは Session ごとに Inventory フラグを持っており、フラグは A または B の状態を保持しています。本設定では、Inventory 処理を行う際に、RF タグが持つ Inventoried フラグ(A/B)のうち、どちらのフラグの RF タグを読み取り対象にするかを指定します。

### 10.1.23 UHF\_Target

Select コマンドで使用する Target 値(3bit)を定義します。

定数名	値	説明
Inventoried_S0	0x00	000b : Inventoried(S0)
Inventoried_S1	0x01	001b : Inventoried(S1)
Inventoried_S2	0x02	010b : Inventoried(S2)
Inventoried_S3	0x03	011b : Inventoried(S3)
SL	0x04	100b : SL
RFU_101b	0x05	101b : 将来のための予約(未サポート)
RFU_110b	0x06	110b : 将来のための予約(未サポート)
RFU_111b	0x07	111b : 将来のための予約(未サポート)

#### [Select コマンドの Target 値説明]

Select コマンドの対象となるフラグを指定します。

マスク条件が一致した RF タグに対して、ここで指定したフラグの状態を Action 値で指定した状態に変更します。

Target 値 (定数名)	Select 対象フラグ
Inventoried_S0	Session0 の Inventoried フラグ
Inventoried_S1	Session1 の Inventoried フラグ
Inventoried_S2	Session2 の Inventoried フラグ
Inventoried_S3	Session3 の Inventoried フラグ
SL	SL フラグ

### 10.1.24 UHF\_TRext

Inventory 処理で使用する TRext 値(1bit)を定義します。

定数名	値	説明
NoPilotTone	0x00	0b : No Pilot Tone
UsePilotTone	0x01	1b : Use Pilot Tone

#### [TRext 値説明]

RF タグからの応答のプリアンブル (同期信号) に「pilot tone」を含むかどうかの設定です。通常は「No Pilot Tone」を指定します。

### 10.1.25 UHF\_Truncate

Select コマンドで使用する Truncate 値(1bit)を定義します。

定数名	値	説明
Disable	0x00	0b : Disable
Enable	0x01	1b : Enable(未サポート)

#### [Truncate 値説明]

Select コマンド後に実行する Inventory 処理において、マスクされた RF タグから返される EPC(UII)データの値を切り詰めるかどうかの設定です。

UTR シリーズリーダーライタは「Disable」のみをサポートしていますので、その他の設定は指定しないでください。

### 10.1.26 UHF\_ThroughCmdType

ThroughCmd で使用するコマンド種別を定義します。

定数名	値	説明
Send	0x00	000b : コマンド送信のみ
Read	0x01	001b : リード系コマンド
Write	0x02	010b : ライト系コマンド
SpecifyTimeout	0x07	111b : 受信待ち時間指定

## 10.2 メソッド引数用クラス

### 10.2.1 ActionModeOption

リーダライタ動作モードオプションです。

GetActionMode メソッドの応答に含まれる動作モードオプション(e.InputData の 8 バイト目)をコンストラクタに渡して、初期化することができます。

[プロパティ]

データ型	プロパティ名	説明
RFID_BaudRate	BaudRate	通信速度 デフォルト値 : BaudRate115200
RFID_UseBuzzer	UseBuzzer	ブザー (鳴らさない : Unuse、鳴らす : Use) デフォルト値 : Unuse

### 10.2.2 UHF\_SelectOption

UHF\_SelectParam メソッド用オプションです。

[プロパティ]

データ型	プロパティ名	説明
UHF_MemBank	MemBank	MemBank デフォルト値 : EPC
UHF_Action	Action	Action 値 デフォルト値 : Act_000b
UHF_Target	Target	Target デフォルト値 : SL
UHF_Truncate	Truncate	Truncate デフォルト値 : Disable
uint	startMaskAdd	マスク開始アドレス(bit アドレス) デフォルト値 : 0
byte	maskLength	マスク bit 数(最大 128bit まで) デフォルト値 : 0
byte[]	maskValue	マスクデータ デフォルト値 : 0

- MemBank  
「10.1.19 UHF\_MemBank」を参照してください。
- Action  
「10.1.16 UHF\_Action」を参照してください。
- Target  
「10.1.23 UHF\_Target」を参照してください。
- Truncate  
「10.1.25 UHF\_Truncate」を参照してください。
- startMaskAdd  
MemBank で指定したメモリ領域のマスク開始アドレスを bit 単位で指定します。  
※0Word 目の先頭 bit を 0bit 目と数えます。
- maskLength  
マスク開始アドレスからマスクする bit 長を指定します。  
上限 128bit まで指定することができます。
- maskValue  
マスクデータを byte 単位で指定します。  
上限 16 byte まで指定することができます。

### 10.2.3 UHF\_InventoryOption

Inventory 処理用オプションです。

[プロパティ]

データ型	プロパティ名	説明
UHF_InventoryTarget	InventoryTarget	Inventory 処理の Target デフォルト値 : InventoryA
UHF_Session	Session	Session 値 デフォルト値 : S2
UHF_Sel	Sel	Sel 値 デフォルト値 : SL
UHF_TRext	TRext	TRext 値 デフォルト値 : NoPilotTone
UHF_M	M	M 値 デフォルト値 : M4
UHF_DR	DR	DR 値 デフォルト値 : DR64_3
byte	Q	Q 値の開始値 指定範囲 : 0~15 デフォルト値 : 3
byte	Qmin	Q 値の最小値 指定範囲 : 0~15 デフォルト値 : 1
byte	Qmax	Q 値の最大値 指定範囲 : 0~15 デフォルト値 : 8
bool	UseQ_AutoResize	Q 値の自動 UP/DOWN 機能 (使用しない : false、使用する : true) デフォルト値 : true 使用する
bool	UseAntiCollision	アンチコリジョン機能 (使用しない : false、使用する : true) デフォルト値 : true 使用する
bool	UseSelectCmd	インベントリ処理時の Select コマンド (使用しない : false、使用する : true) デフォルト値 : true 使用する

(注意事項)

インベントリ処理の対象となる RF タグを、Session と Sel で指定します。

Sel の値を「UHF\_Sel.SL」に設定した場合、RF タグの SL フラグが SL になっている場合でも、指定した Session の Inventoried フラグが A になっていないと、その RF タグは読み取ることができません。

Session の条件と Sel の条件は AND 条件となりますのでご注意ください。

- InventoryTarget  
「10.1.22 UHF\_InventoryTarget」を参照してください。
- Session  
「10.1.20 UHF\_Session」を参照してください。
- Sel  
「10.1.21 UHF\_Sel」を参照してください。
- TRext  
「10.1.24 UHF\_TRext」を参照してください。
- M  
「10.1.18 UHF\_M」を参照してください。
- DR  
「10.1.17 UHF\_DR」を参照してください。
- Q (Q 値の初期値)  
インベントリ処理の中で使用するスロット数 (=2 の Q 乗) を指定します。  
「UseQ\_AutoResize=true」に設定した場合、Q の値を初期値として処理を開始します。

※スロット数

RF タグが応答を返すタイムスロット数を指定します。

全ての RF タグが、指定した範囲内のいずれかのスロット番号で応答を返します。

RF タグの枚数と比較してスロット数が少なすぎると、RF タグのデータが正常に受信できません。

以下の条件を目安としてください。(スロット数が RF タグ枚数の半分程度)

1 回の処理で読み取る RF タグの最大枚数	Q 値	スロット数
1 枚	0	1 (2 の 0 乗)
~10 枚	2	4 (2 の 2 乗)
~20 枚	3	8 (2 の 3 乗)
~30 枚	4	16 (2 の 4 乗)
~50 枚	5	32 (2 の 5 乗)

スロット数が小さすぎると、読みこぼしにつながります。

読み取りが安定しない場合は Q 値を 1 増やしてお試しください。

また、スロット数が大きすぎると、処理時間が遅くなります。

Q 値を必要以上に大きくしないでください。

- Qmin (Q 値の最小値)  
「UseQ\_AutoResize=true」に設定した場合、リーダライタ内部で Q 値を変更しながら処理を行いますが、その場合の Q 値の下限値を指定します。
- Qmax (Q 値の最大値)  
「UseQ\_AutoResize=true」に設定した場合、リーダライタ内部で Q 値を変更しながら処理を行いますが、その場合の Q 値の上限値を指定します。

- **UseQ\_AutoResize**  
インベントリ処理のスロット数を、動的に変更しながら処理を行う場合は「true」を選択します。  
自動制御を行う場合、開始値の「Q」の設定で処理を開始し、RF タグの応答状況に応じて最小値「Qmin」から最大値「Qmax」の範囲で Q 値を変更しながら処理を行います。
- **UseAntiCollision**  
「使用しない」に設定した場合、Inventory 処理において、コリジョンが発生したスロットは、タグの読み取りを行いません。  
コリジョン処理を行わない分、読み取り速度が上がりますが、読み取り枚数にバラツキが発生します。  
コリジョンが発生したことを検出しませんので、Q 値の自動 UP/DOWN 機能も動作しなくなります。Q 値の設定が適切でない場合には、読み取りが不安定になります。
- **UseSelectCmd**  
UHF 帯の RFID では、リーダライタから RF タグへのコマンドは、一般的に、Select コマンド、Inventory コマンド、Access コマンド (Read/Write など) を順番に発行し RF タグとの通信を行います。  
「使用する」に設定した場合、自動読み取りモードや RF タグ通信コマンドにおいて、Inventory 処理の前に Select コマンドが発行されます。  
MemBank の一部領域をマスクして読み取りを行う際には、Select コマンドの使用が必須となりますので、「使用する」を選択します。  
Select コマンドのパラメータは、「UHF\_SetSelectParam」コマンドの内容が反映されます。

#### 10.2.4 UHF\_ExpandSelectOption

UHF\_ExpandSelectParam メソッド用オプションです。

[プロパティ]

データ型	プロパティ名	説明
UHF_MemBank	MemBank	MemBank デフォルト値 : EPC
UHF_Action	Action	Action 値 デフォルト値 : Act_000b
UHF_Target	Target	Select コマンドの Target デフォルト値 : SL
UHF_Truncate	Truncate	Truncate デフォルト値 : Disable
uint	startMaskAdd	マスク開始アドレス(bit アドレス) デフォルト値 : 0
byte	maskLength	マスク bit 数(最大 128bit まで) デフォルト値 : 0
byte[]	maskValue	マスクデータ デフォルト値 : 0

- MemBank  
「10.1.19 UHF\_MemBank」を参照してください。
- Action  
「10.1.16 UHF\_Action」を参照してください。
- Target  
「10.1.23 UHF\_Target」を参照してください。
- M  
「10.1.18 UHF\_M」を参照してください。
- DR  
「10.1.17 UHF\_DR」を参照してください。
- startMaskAdd  
MemBank で指定したメモリ領域のマスク開始アドレスを bit 単位で指定します。  
※0Word 目の先頭 bit を 0bit 目と数えます。
- maskLength  
マスク開始アドレスからマスクするビット長を指定します。  
上限 128bit まで指定することができます。
- maskValue  
マスクデータを Byte 単位で指定します。  
上限 16 Byte まで指定することができます。

### 10.2.5 UHF\_AntennaParamOption

UHF\_AntennaParam メソッド用オプションです。

[プロパティ]

データ型	プロパティ名	説明
RFID_Antenna Switching	AntennaSwitching	アンテナ切替方式 NoControl 0:制御しない Control 1:制御する SpecialControl 2:制御する(複数アンテナを一つのアンテナとして扱う) デフォルト値 : NoControl: 制御しない
bool	Ant0_en	アンテナ 0 の使用の有無 (使用しない : false、使用する : true) デフォルト値 : true 使用する
bool	Ant1_en	アンテナ 1 の使用の有無 (使用しない : false、使用する : true) デフォルト値 : false 使用しない
bool	Ant2_en	アンテナ 2 の使用の有無 (使用しない : false、使用する : true) デフォルト値 : false 使用しない
bool	Ant3_en	アンテナ 3 の使用の有無 (使用しない : false、使用する : true) デフォルト値 : false 使用しない
bool	AntennaIDOut	アンテナ ID 出力 (出力しない : false、出力する : true) デフォルト値 : true 出力する

※リーダライタにより接続できるアンテナの数量が異なりますので、ご使用になるリーダライタの製品仕様書をご参照のうえ、機器に合わせて使用するアンテナを選択してください。

- AntennaSwitching  
「10.1.7 RFID\_AntennaSwitching」を参照してください。

### 10.2.6 UHF\_TxPowerParamOption

UHF\_TxPowerParam メソッド用オプションです。

[プロパティ]

データ型	プロパティ名	説明
int	Tx_Power	キャリア出力レベル(dBm*10) 設定する出力レベルを 10 倍して指定 デフォルト値 : 240 (24.0dBm)
int	Tx_OnTime	キャリア送信時間(ms) デフォルト値 : 2000
int	Tx_OffTime	キャリア休止時間(ms) デフォルト値 : 50
int	LBT_Time	キャリアセンス待ち時間(ms) デフォルト値 : 200

※リーダライタにより設定できる出力の上限値および下限値が異なりますので、ご使用になるリーダライタの製品仕様書をご参照のうえ、機器に合わせて Tx\_Power 値を設定してください。

※設定できない Tx\_Power 値を指定した場合、NACK 応答が返ります。

### 10.2.7 UHF\_FrequencyParamOption

UHF\_FrequencyParam メソッド用オプションです。

[プロパティ]

データ型	プロパティ名	説明
RFID_FrequencyChannel	StartFrequencyChannel	開始チャンネル番号 デフォルト値 : Ch26
RFID_FrequencyChannel	NowFrequencyChannel	現在設定されているチャンネル番号 デフォルト値 : Ch26
RFID_FrequencyChannel Kounai	FrequencyUseChannel Kounai	構内無線局優先使用チャンネル デフォルト値 : ChNone
RFID_FrequencyChannel Tokushou	FrequencyUseChannel Tokushou	特定小電力無線局優先使用 チャンネル デフォルト値 : ChALL
RFID_FrequencyChannel Active	FrequencyUseChannel Active	アクティブタグ優先使用 チャンネル デフォルト値 : ChNone

### 10.2.8 UHF\_RFTAGComParamOption

UHF\_RFTAGComParam メソッド用オプションです。

[プロパティ]

データ型	プロパティ名	説明
bool	RSSIFilterFunc	RSSI フィルタ機能 (使用しない : false、使用する : true) デフォルト値 : false 使用しない
RFID_FrequencyScanMode	FrequencyScanMode	周波数のスキャンモード デフォルト値 : CarrierSensePriority
bool	TargetABAutoChange	Inventory の Target A/B 自動切替 (切替しない : false、切替する : true) デフォルト値 : true 切替する
sbyte	RSSI	RSSI フィルタのしきい値 デフォルト値 : -65
int	InventoryRetryCount	インベントリ処理のリトライ回数 指定範囲 : 0~15 デフォルト値 : 0
int	ReadWriteRetryCount	リードライト処理のリトライ回数 指定範囲 : 0~15 デフォルト値 : 0

- RFID\_FrequencyScanMode  
「10.1.5 RFID\_ScanMode」を参照してください。

### 10.2.9 UHF\_EPCParamOption

UHF\_EPCParam メソッド用オプションです。

[プロパティ]

データ型	プロパティ名	説明
bool	EPCBuffering	EPC データのバッファリング処理(重複禁止) (使用しない : false、使用する : true) デフォルト値 : false 使用しない
bool	InventoryCycleResponse	自動読み取りモード時の読み取りサイクル終了時のレスポンス (返さない : false、返す : true) デフォルト値 : false 返さない
bool	AntennaSwitchingResponse	アンテナ自動切り替え時のレスポンス (返さない : false、返す : true) デフォルト値 : false 返さない
bool	CarrierSenseResponse	キャリアセンスにかかった時のレスポンス (返さない : false、返す : true) デフォルト値 : true 返す

- EPCBuffering

EPCデータのバッファリング処理

Inventory処理のレスポンスデータを返す処理の過程で、同じタグデータを返すかどうかの指定を設定します。

「true 使用する」に設定すると、1回のアンチコリジョンの間に同じタグのデータを返さなくなります。

本設定は、EPC(UID)データがユニークである前提で使用可能な設定です。

異なるRFタグに同じEPC(UID)データを書き込んでいる場合、本設定は「false 使用しない」を選択する必要があります。

- InventoryCycleResponse

自動読み取りモード時の読み取りサイクル終了時のレスポンス

Inventory処理のレスポンスデータを返す処理の過程で、1回のアンチコリジョンの終了時にレスポンスを返すかどうかの設定をします。

「true 返す」に設定すると、Inventory処理終了時に、[RFタグの読取枚数]、および[送信したキャリアのチャンネル番号]を含むレスポンスを返します。

- AntennaSwitchingResponse

アンテナ自動切替終了時のレスポンス

Inventory処理のレスポンスデータを返す処理の過程で、アンテナ自動切替終了時のタイミングでレスポンスを返すかどうかの指定を設定します。「true 返す」に設定すると、アンテナ自動切替終了時にレスポンスを返します。

- **CarrierSenseResponse**

キャリアセンスにかかった時のレスポンス

リーダライタがキャリアを出そうとしている周波数で、他の機器から既にキャリアが出力されていた場合、電波法の規定上、リーダライタはキャリアを出すことができず、タグを読み取ることができません。本説明書ではこの状態を「キャリアセンスにかかった」と表現しています。

自動読み取りモード時に、キャリアセンスにかかった場合に、リーダライタからレスポンスを返すかどうかの指定を設定します。

**CarrierSenseResponse = true**に設定すると、キャリアセンスにかかっている間、最小200msの間隔でレスポンスが返されます。

コマンドモード時には、本項目の設定によらず、キャリアセンスにかかった時のレスポンスは返りません。[キャリアセンス待ち時間]を越えてもキャリアの出力が開始できなかった場合には、コマンドのレスポンスとしてNACK応答が返されます。

※補足説明

キャリアセンスにかかった時の動作は、「周波数のスキャンモード」により異なります。

「周波数のスキャンモード」の詳細は、**RFID\_FrequencyScanMode** 列挙体を参照してください。

### 10.2.10 UHF\_TIDLKOption

TID 領域の Lock 処理用オプションです。

[プロパティ]

データ型	プロパティ名	説明
bool	TID_PermaLock_Mask	TID PermaLock Mask 値 (OFF : false、ON : true) デフォルト値 : false
bool	TID_PermaLock_Action	TID PermaLock Action 値 (OFF : false、ON : true) デフォルト値 : false
bool	TID_PasswordWrite_Mask	TID PasswordWrite Mask 値 (OFF : false、ON : true) デフォルト値 : false
bool	TID_PasswordWrite_Action	TID PasswordWrite Action 値 (OFF : false、ON : true) デフォルト値 : false

#### [TID PermaLock について]

TID のロック状態を変更不可 (Permanent Lock) とする機能です。

ただし、TID は、通常は ReadOnly 領域のため、本パラメータによらずロック状態を変更することはできません。

TID\_PermaLock\_Mask を true で実行した場合のみ、  
TID\_PermaLock\_Action の設定値が RF タグに書き込まれます。

- TID\_PermaLock\_Mask  
false : PermaLock の Action 設定値は書き込まれません。  
true : PermaLock の Action 設定値が書き込まれます。
- TID\_PermaLock\_Action  
false : TID の PermaLock は解除されません。  
true : TID が PermaLock されます。

※PermaLock 実行後は、Lock 状態を変更することができません。

- TID\_PasswordWrite の Lock を設定した状態で TID\_PermaLock を実行した場合  
→TID 領域に対する Write ができなくなり、TID\_PasswordWrite の Lock 状態を解除することができなくなります。
- TID\_PasswordWrite の Lock を解除した状態で TID\_PermaLock を実行した場合  
→TID 領域に対する Write はできますが (通常はできません)、TID\_PasswordWrite の Lock を設定することができなくなります。

**[TID PasswordWrite について]**

TID をライトロックする機能です。

ただし、TID は、通常は ReadOnly 領域のため、本パラメータによらずロック状態を変更することはできません。

TID\_PasswordWrite\_Mask を true で実行した場合のみ、

TID\_PasswordWrite\_Action の設定値が RF タグに書き込まれます。

- TID\_PasswordWrite\_Mask  
false : PasswordWrite の Action 設定値は書き込まれません。  
true : PasswordWrite の Action 設定値が書き込まれます。
- TID\_PasswordWrite\_Action  
false : TID のライトロックが解除されます。(通常は解除されません)  
true : TID がライトロックされます。

### 10.2.11 UHF\_EPCLockOption

EPC(UII)領域の Lock 処理用オプションです。

[プロパティ]

データ型	プロパティ名	説明
bool	EPC_PermaLock_Mask	EPC PermaLock Mask 値 (OFF : false、ON : true) デフォルト値 : false
bool	EPC_PermaLock_Action	EPC PermaLock Action 値 (OFF : false、ON : true) デフォルト値 : false
bool	EPC_PasswordWrite_Mask	EPC PasswordWrite Mask 値 (OFF : false、ON : true) デフォルト値 : false
bool	EPC_PasswordWrite_Action	EPC PasswordWrite Action 値 (OFF : false、ON : true) デフォルト値 : false

#### [EPC(UII) PermaLock について]

EPC(UII)のロック状態を変更不可 (Permanent Lock) とする機能です。

PermaLock を実行しなければ、EPC(UII)のライトロック状態を何度でも変更することができます。

PermaLock 実行後は、ライトロック状態を変更することができません。

EPC\_PermaLock\_Mask を true で実行した場合のみ、

EPC\_PermaLock\_Action の設定値が RF タグに書き込まれます。

- EPC\_PermaLock\_Mask  
false : PermaLock の Action 設定値は書き込まれません。  
true : PermaLock の Action 設定値が書き込まれます。
- EPC\_PermaLock\_Action  
false : EPC(UII)の ParmaLock は解除されません。  
true : EPC(UII)が ParmaLock されます。
- EPC\_PasswordWrite の Lock を設定した状態で EPC\_PermaLock を実行した場合  
→EPC(UII)領域に対する Write ができなくなり、EPC\_PasswordWrite の Lock 状態を解除することができなくなります。
- EPC\_PasswordWrite の Lock を解除した状態で EPC\_PermaLock を実行した場合  
→EPC(UII)領域に対する Write はできますが、EPC\_PasswordWrite の Lock を設定することができなくなります。

### [EPC(UII) PasswordWrite について]

EPC(UII)をライトロックする機能です。

ライトロックとは、リードはできるがライトはできない状態です。

ただし、リーダライタに Access パスワードを書き込んで Access パスワードの認証を行った場合は、ライトが可能となります。

EPC(UII)\_PasswordWrite\_Mask を true で実行した場合のみ、

EPC(UII)\_PasswordWrite\_Action の設定値が RF タグに書き込まれます。

- EPC(UII)\_PasswordWrite\_Mask  
false : PasswordWrite の Action 設定値は書き込まれません。  
true : PasswordWrite の Action 設定値が書き込まれます。
- EPC(UII)\_PasswordWrite\_Action  
false : EPC(UII)のライトロックが解除されます。  
true : EPC(UII)がライトロックされます。

### 10.2.12 UHF\_UserLockOption

User 領域の Lock 処理用オプションです。

[プロパティ]

データ型	プロパティ名	説明
bool	User_PermaLock_Mask	User PermaLock Mask 値 (OFF : false、ON : true) デフォルト値 : false
bool	User_PermaLock_Action	User PermaLock Action 値 (OFF : false、ON : true) デフォルト値 : false
bool	User_PasswordWrite_Mask	User PasswordWrite Mask 値 (OFF : false、ON : true) デフォルト値 : false
bool	User_PasswordWrite_Action	User PasswordWrite Action 値 (OFF : false、ON : true) デフォルト値 : false

#### [User PermaLock について]

User 領域のロック状態を変更不可 (Permanent Lock) とする機能です。

PermaLock を実行しなければ、User 領域のライトロック状態を何度でも変更することができます。

PermaLock 実行後は、ライトロック状態を変更することができません。

User\_PermaLock\_Mask を true で実行した場合のみ、

User\_PermaLock\_Action の設定値が RF タグに書き込まれます。

- User\_PermaLock\_Mask  
false : PermaLock の Action 設定値は書き込まれません。  
true : PermaLock の Action 設定値が書き込まれます。
- User\_PermaLock\_Action  
false : User 領域の PermaLock は解除されません。  
true : User 領域が PermaLock されます。
- User\_PasswordWrite の Lock を設定した状態で User\_PermaLock を実行した場合  
→User 領域に対する Write ができなくなり、User\_PasswordWrite の Lock 状態を解除することができなくなります。
- User\_PasswordWrite の Lock を解除した状態で User\_PermaLock を実行した場合  
→User 領域に対する Write はできますが、User\_PasswordWrite の Lock を設定することができなくなります。

### [User PasswordWrite について]

User 領域をライトロックする機能です。

ライトロックとは、リードはできるがライトはできない状態です。  
ただし、リーダライタに Access パスワードを書き込んで Access パスワードの認証を行った場合は、ライトが可能となります。

User\_PasswordWrite\_Mask を true で実行した場合のみ、  
User\_PasswordWrite\_Action の設定値が RF タグに書き込まれます。

- User\_PasswordWrite\_Mask  
false : PasswordWrite の Action 設定値は書き込まれません。  
true : PasswordWrite の Action 設定値が書き込まれます。
- User\_PasswordWrite\_Action  
false : User 領域のライトロックが解除されます。  
true : User 領域がライトロックされます。

### 10.2.13 UHF\_AccessPwdLockOption

Access パスワードの Lock 処理用オプションです。

[プロパティ]

データ型	プロパティ名	説明
bool	AccessPwd_PermaLock_Mask	AccessPassword PermaLock Mask 値 (OFF : false、ON : true) デフォルト値 : false
bool	AccessPwd_PermaLock_Action	AccessPassword PermaLock Action 値 (OFF : false、ON : true) デフォルト値 : false
bool	AccessPwd_PwdReadWrite_Mask	AccessPassword PasswordReadWrite Mask 値 (OFF : false、ON : true) デフォルト値 : false
bool	AccessPwd_PwdReadWrite_Action	AccessPassword PasswordReadWrite Action 値 (OFF : false、ON : true) デフォルト値 : false

#### [AccessPassword PermaLock について]

Access パスワードのロック状態を変更不可 (Permanent Lock) とする機能です。

PermaLock を実行しなければ、Access パスワードのリードライトロック状態を何度でも変更することができます。

PermaLock 実行後は、リードライトロック状態を変更することができません。

AccessPwd\_PermaLock\_Mask を true で実行した場合のみ、  
AccessPwd\_PermaLock\_Action の設定値が RF タグに書き込まれます。

- AccessPwd\_PermaLock\_Mask  
false : PermaLock の Action 設定値は書き込まれません。  
true : PermaLock の Action 設定値が書き込まれます。
- AccessPwd\_PermaLock\_Action  
false : Access パスワードの PermaLock は解除されません。  
true : Access パスワードの PermaLock が「設定」されます。
- AccessPwd\_PwdReadWrite の Lock を設定した状態で AccessPwd\_PermaLock を実行した場合  
→Access パスワードの Read / Write ができなくなり、Access パスワードの Lock 状態を解除することができなくなります。
- AccessPwd\_PwdReadWrite の Lock を解除した状態で AccessPwd\_PermaLock を実行した場合  
→Access パスワードの Read / Write はできますが、Access パスワードの Lock を設定することができなくなります。

### [AccessPassword PasswordReadWrite について]

Access パスワードをリードライトロックする機能です。

リードライトロックとは、リードもライトもできない状態です。  
ただし、リーダライタに Access パスワードを書き込んで Access パスワードの認証を行った場合は、リードおよびライトが可能となります。

AccessPwd\_PwdReadWrite\_Mask を true で実行した場合のみ、  
AccessPwd\_PwdReadWrite\_Action の設定値が RF タグに書き込まれます。

- AccessPwd\_PwdReadWrite\_Mask  
false : PasswordReadWrite の Action 設定値は書き込まれません。  
true : PasswordReadWrite の Action 設定値が書き込まれます。
- AccessPwd\_PwdReadWrite\_Action  
false : Access パスワードのリードライトロックが「解除」されます。  
true : Access パスワードのリードライトロックが「設定」されます。

### 10.2.14 UHF\_KillPwdLockOption

Kill パスワードの Lock 処理用オプションです。

[プロパティ]

データ型	プロパティ名	説明
bool	KillPwd_PermaLock_Mask	KillPassword PermaLock Mask 値 (OFF : false、ON : true) デフォルト値 : false
bool	KillPwd_PermaLock_Action	KillPassword PermaLock Action 値 (OFF : false、ON : true) デフォルト値 : false
bool	KillPwd_PwdReadWrite_Mask	KillPassword PasswordReadWrite Mask 値 (OFF : false、ON : true) デフォルト値 : false
bool	KillPwd_PwdReadWrite_Action	KillPassword PasswordReadWrite Action 値 (OFF : false、ON : true) デフォルト値 : false

#### [KillPassword PermaLock について]

Kill パスワードのロック状態を変更不可 (Permanent Lock) とする機能です。

PermaLock を実行しなければ、Kill パスワードのリードライトロック状態を何度でも変更することができます。

PermaLock 実行後は、リードライトロック状態を変更することができません。

KillPwd\_PermaLock\_Mask を true で実行した場合のみ、

KillPwd\_PermaLock\_Action の設定値が RF タグに書き込まれます。

- KillPwd\_PermaLock\_Mask  
false : PermaLock の Action 設定値は書き込まれません。  
true : PermaLock の Action 設定値が書き込まれます。
- KillPwd\_PermaLock\_Action  
false : Kill パスワードの PermaLock は解除されません。  
true : Kill パスワードの PermaLock が「設定」されます。
- KillPwd\_PwdReadWrite の Lock を設定した状態で KillPwd\_PermaLock を実行した場合  
→Kill パスワードの Read / Write ができなくなり、Kill パスワードの Lock 状態を解除することができなくなります。
- KillPwd\_PwdReadWrite の Lock を解除した状態で KillPwd\_PermaLock を実行した場合  
→Kill パスワードの Read / Write はできますが、Kill パスワードの Lock を設定することができなくなります。

**[KillPassword PasswordReadWrite について]**

Kill パスワードをリードライトロックする機能です。

リードライトロックとは、リードもライトもできない状態です。  
ただし、リーダライタに Access パスワードを書き込んで Access パスワードの認証を行った場合は、リードおよびライトが可能となります。

KillPwd\_PwdReadWrite\_Mask を true で実行した場合のみ、  
KillPwd\_PwdReadWrite\_Action の設定値が RF タグに書き込まれます。

- KillPwd\_PwdReadWrite\_Mask  
false : PasswordReadWrite の Action 設定値は書き込まれません。  
true : PasswordReadWrite の Action 設定値が書き込まれます。
- KillPwd\_PwdReadWrite\_Action  
false : Kill パスワードのリードライトロックが「解除」されます。  
true : Kill パスワードのリードライトロックが「設定」されます。

### 10.2.15 UHF\_EncodeOption

RF タグの Lock 処理時に各 MemBank の Lock[する／しない]の指定、および、RF タグの Lock 済みの MemBank にアクセスする際に、Access コマンドを発行[する／しない]を指定するためのオプションです。

UHF\_Encode メソッド用オプションです。

[プロパティ]

データ型	プロパティ名	説明
bool	Lock	Lock コマンドの発行 (発行しない : false、発行する : true) デフォルト値 : false 発行しない
bool	Reserved_Rewrite	Reserved 領域書き込み時の Access コマンドの発行 (発行しない : false、発行する : true) デフォルト値 : false 発行しない
bool	EPC_Rewrite	EPC 領域書き込み時の Access コマンドの発行 (発行しない : false、発行する : true) デフォルト値 : false 発行しない
bool	User_Rewrite	User 領域書き込み時の Access コマンドの発行 (発行しない : false、発行する : true) デフォルト値 : false 発行しない

- Lock  
Lock コマンドを実行するかどうかを指定します。
  - true を指定した場合、各 MemBank に対して設定した Lock の処理内容を実行します。
  - false を指定した場合、Lock 処理は実行されません。
- Reserved\_Rewrite  
UHF\_Encode メソッドは複数の MemBank に対して書き込みをおこなうため、MemBank により WriteLock が掛かっている場合と掛かっていない場合があります。  
Reserved 領域への Write 時に Access コマンドを「発行する／発行しない」を指定します。
- EPC\_Rewrite  
UHF\_Encode メソッドは複数の MemBank に対して書き込みをおこなうため、MemBank により WriteLock が掛かっている場合と掛かっていない場合があります。  
EPC 領域への Write 時に Access コマンドを「発行する／発行しない」を指定します。
- User\_Rewrite  
UHF\_Encode メソッドは複数の MemBank に対して書き込みをおこなうため、MemBank により WriteLock が掛かっている場合と掛かっていない場合があります。  
User 領域への Write 時に Access コマンドを「発行する／発行しない」を指定します。

### 10.2.16 UHF\_ThroughCmdOption

スルーコマンド (UHF\_ThroughCmd メソッド) 用オプションです。

[プロパティ]

データ型	プロパティ名	説明
UHF_ThroughCmdType	ThroughCmdType	コマンド種別 デフォルト値 : Read (値 : 0x01)
bool	RcvGainUp	受信ゲインアップを行う (行わない : false、行う : true) デフォルト値 : false
bool	NotAddHandleToTx	RF タグへの送信データに自動で Handle を付加しない (付加する : false、付加しない : true) デフォルト値 : false
bool	NotReturnCRCFromTag	RF タグから受信した CRC を上位へ返さない (返す : false、返さない : true) デフォルト値 : false
bool	NotCheckCRCFromTag	RF タグから受信した CRC のチェックを行わない (行う : false、行わない : true) デフォルト値 : false
bool	AddCRCToTx	RF タグへの送信データに自動で CRC を付加する (付加しない : false、付加する : true) デフォルト値 : true

- UHF\_ThroughCmdType  
「10.1.26 UHF\_ThroughCmdType」を参照してください。

---

---

# 第11章 UTR-SDKV1 プログラミング

本章では、UTR-SDKV1 を利用したプログラミング方法について説明します。

---

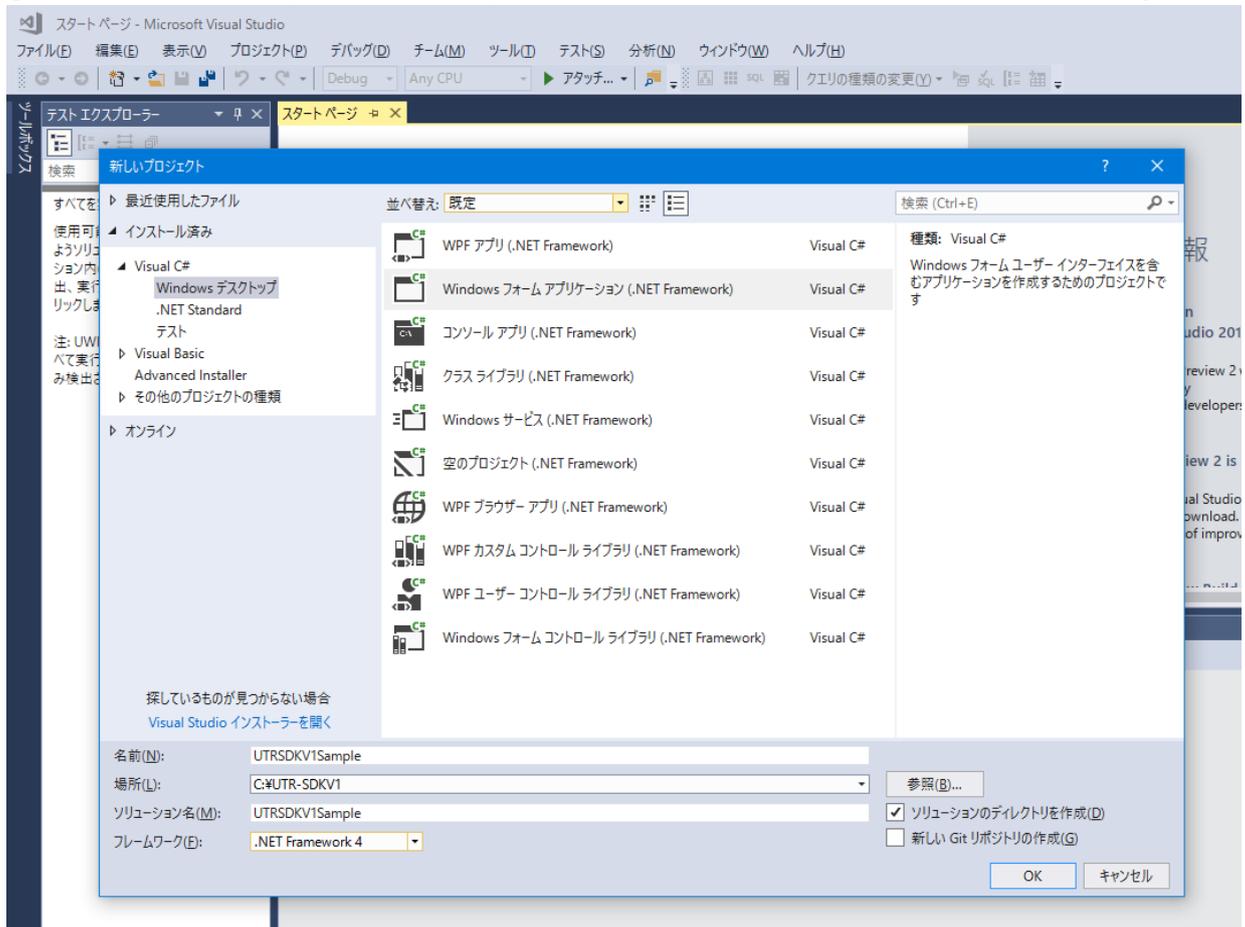
---

本章では、Microsoft Visual Studio 2017 (C#) を利用して Windows アプリケーションを作成する場合の開発手順を説明します。

(その他のバージョンの Microsoft Visual Studio、及び UTR-SDKV1 を利用した場合も手順は同様です)

## 11.1 プロジェクトの作成

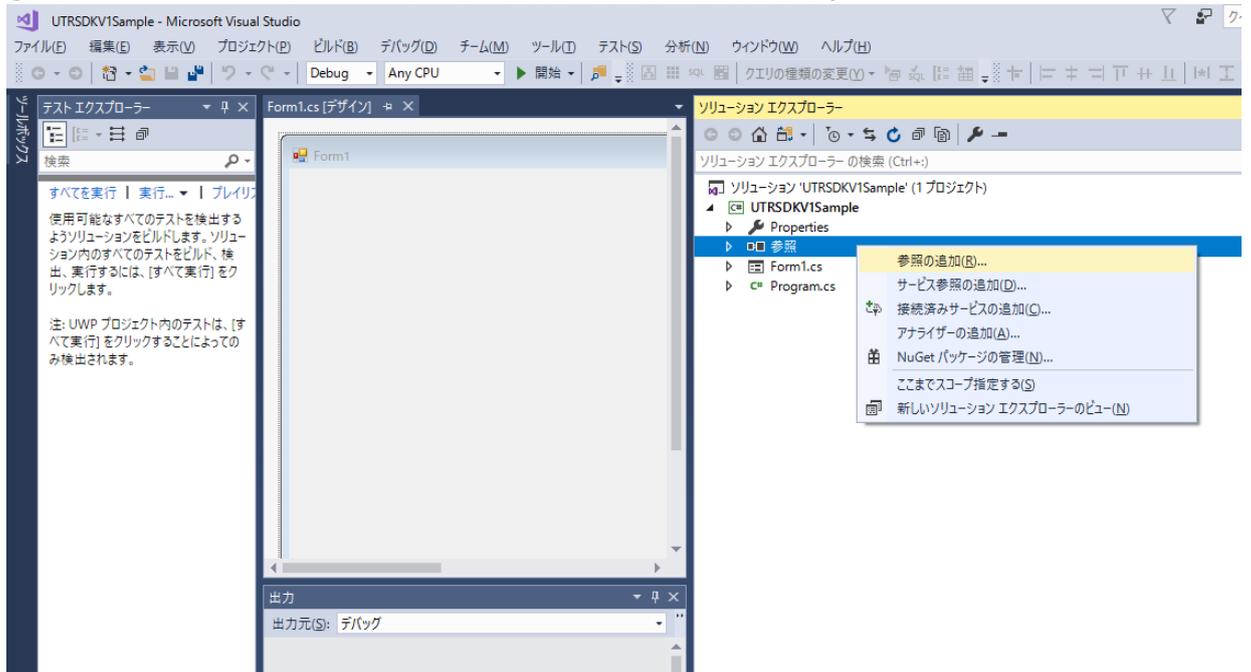
- ① Microsoft Visual Studio 2017 を起動します。
- ② プロジェクトの新規作成を選択し、Windows フォームアプリケーションを作成します。



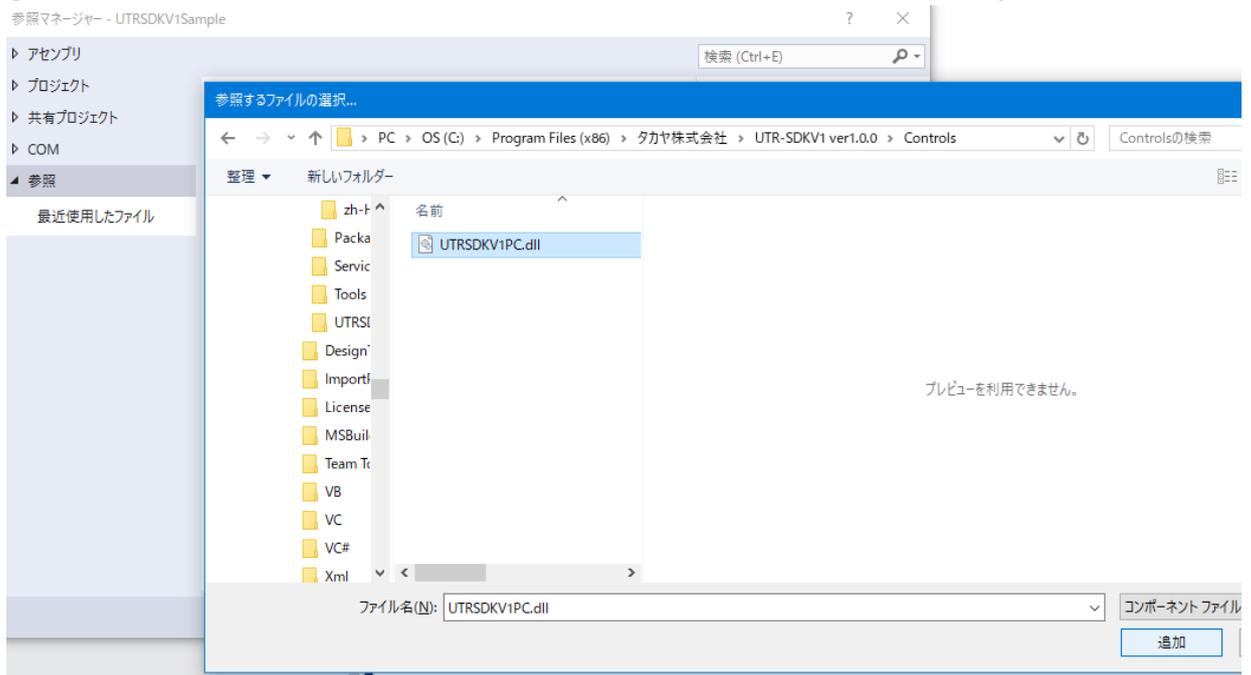
プロジェクトの種類 : VisualC#-Windows  
テンプレート : Windows フォームアプリケーション  
プロジェクト名 : UTRSDKV1Sample  
場所 : C:\¥UTR-SDKV1  
ソリューション名 : UTRSDKV1Sample  
フレームワーク名 : .NET Framework 4

## 11.2 UTR-SDKV1 への参照の追加

- ① メニューバー — [プロジェクト]—[参照の追加]を選択します。

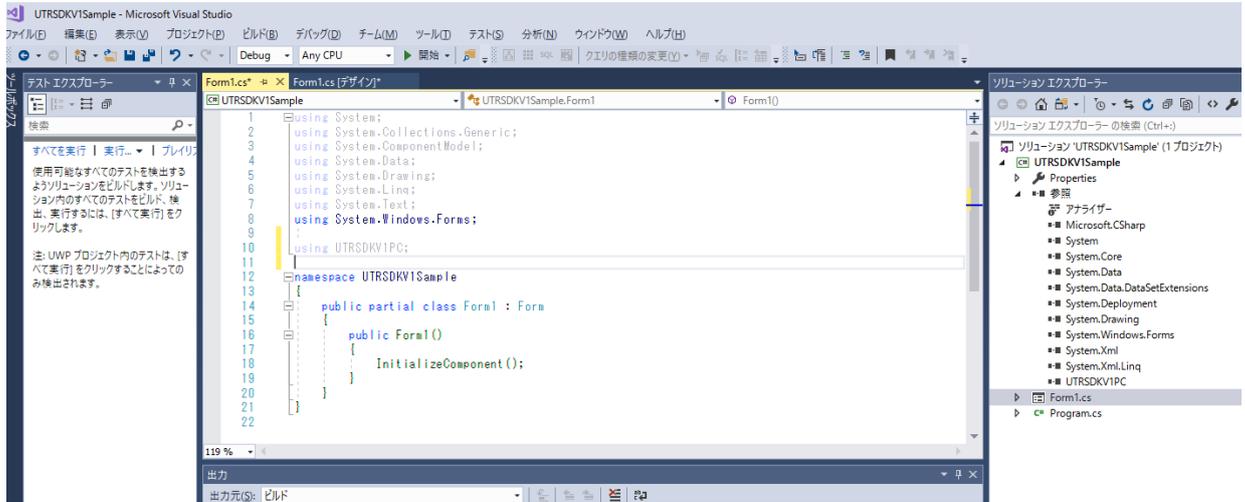


- ② UTR-SDKV1 のインストールフォルダの UTRSDKV1PC.dll を選択します。

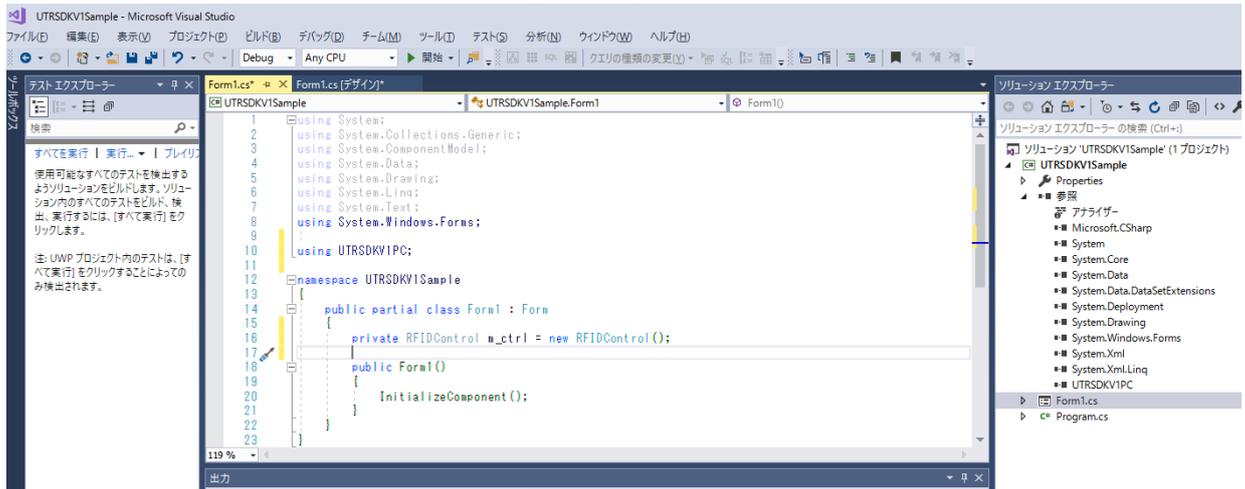


## 11.3 ResponseRFID イベントハンドラの追加

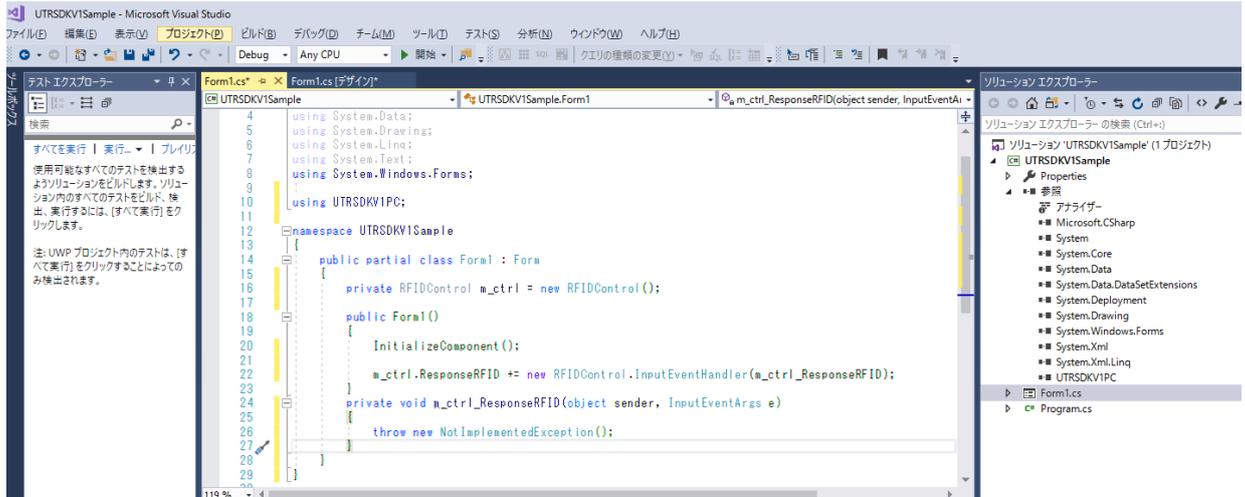
### ① UTRSDKV1 をインポートします。



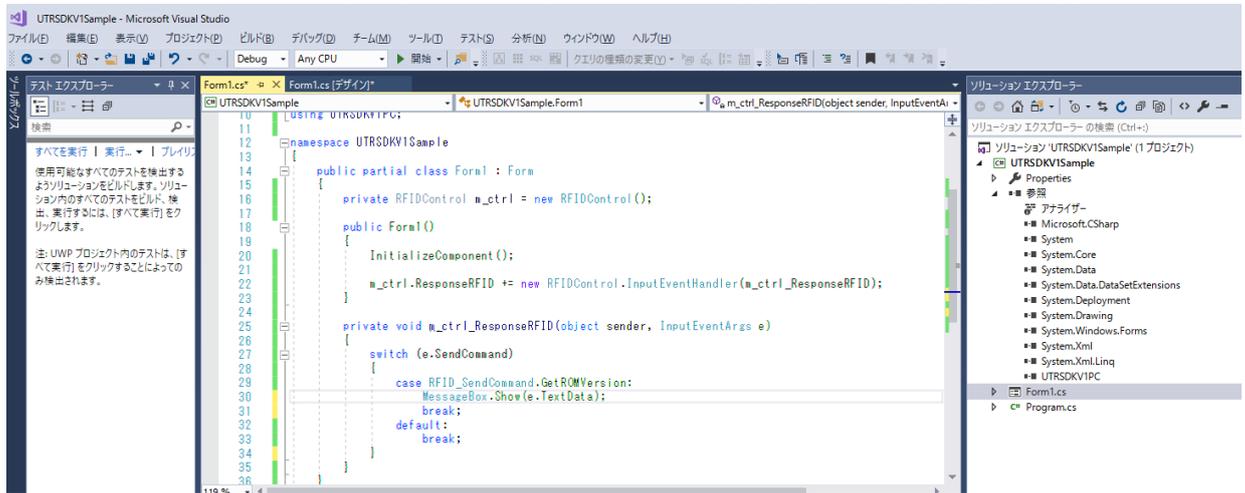
### ② RFIDControl 型のオブジェクトを作成します。



③ ResponseRFID イベントハンドラを追加します。

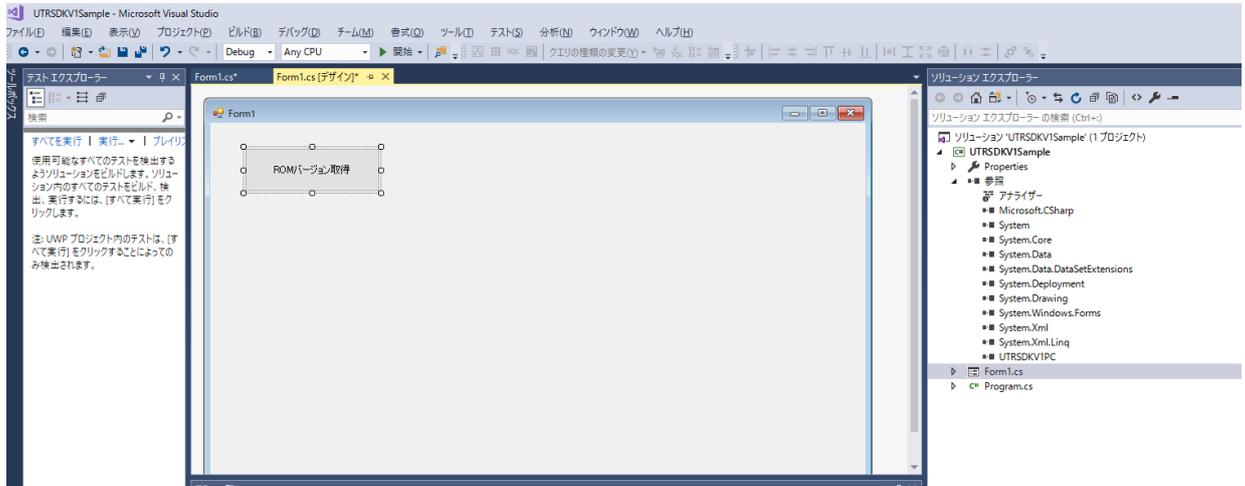


④ イベントハンドラ内にリーダライタからのレスポンス受信時の処理を記述します。

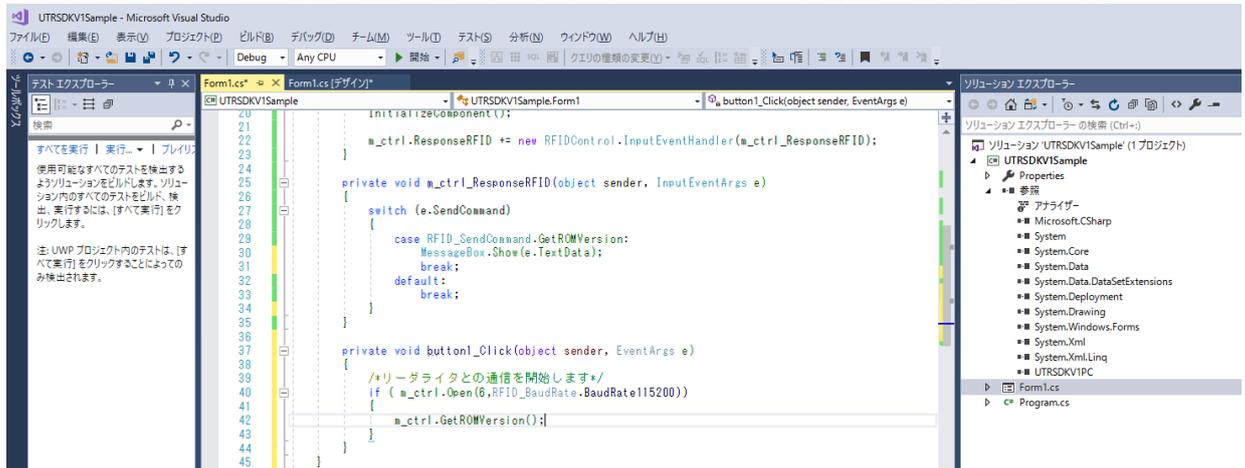


## 11.4 リーダライタとの通信処理を記述

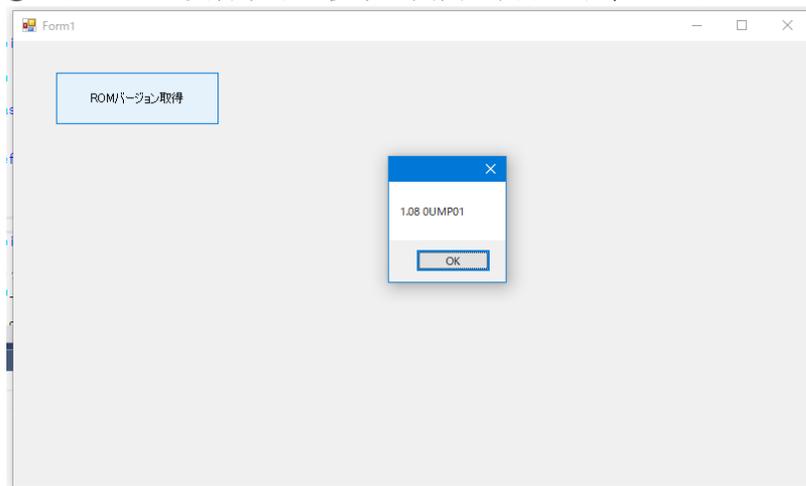
### ⑤ 実行するためのボタンをフォームに配置



### ⑥ リーダライタとの通信開始処理とリーダーライタへのコマンド送信処理を記述します。



### ⑦ F5 キーで実行すると以下の画面が表示され、ROM バージョンが取得できます。



---

---

## 第12章 付録

---

---

## 12.1 NACK 応答

ResponseRFID イベントのパラメータが「e.SendCommand = NACK」の場合、「e.InputData」にはエラーコードを含む NACK 応答がセットされています。  
また、「e.TextData」にはエラーの説明がセットされています。  
NACK 応答時の「e.InputData」のフォーマット (NACK 応答のフォーマット) について解説します。

・ NACK 応答の e.InputData の例

※UHF\_BlockWrite2 メソッドで NACK 応答となった例を示しています。

02 00 31 0A 1D 0A 20 0A 81 00 00 00 00 00 03 12 0D

- ・ エラーコード 1 (NACK 応答の 6 バイト目) : 0Ah  
…タグアクセス時、リーダライタ内蔵チップが返すエラー
- ・ エラーコード 2 (NACK 応答の 7 バイト目) : 20h …  
…前回のタグアクセス時、Write に失敗
- ・ エラーコード 3 (NACK 応答の 8 バイト目) : 0Ah …  
…タグアクセス時、リーダライタ内蔵チップが返すエラー
- ・ エラーコード 4 (NACK 応答の 9 バイト目) : 81h …  
…今回のタグアクセス時、ハンドル取得失敗

<NACK 応答のフォーマットについて>

使用するメソッド、及び RF タグからのエラー情報により、応答のフォーマットが異なります。

- (1) NACK 応答の 6 バイト目はエラーコード 1 がセットされます。以下のメソッド実行時に NACK 応答となった場合に値がセットされます。エラーの内容は、(表 1)をご参照ください。
  - ・リーダライタの制御メソッド 全般
  - ・リーダライタ動作モードの設定メソッド 全般
  - ・リーダライタ FLASH の設定メソッド 全般
  - ・リーダライタ設定メソッド 全般
  - ・RF タグとの通信メソッド 全般
- (2) NACK 応答の 7 バイト目のエラーコード 2 は、エラーコード 1 に「0Ah」がセットされている場合に値がセットされます。エラーコード 2 の内容は、主にタグアクセスの際に発生するエラーです。  
エラーの内容は、(表 2)をご参照ください。
- (3) NACK 応答の 8 バイト目のエラーコード 3 は、以下の場合に値がセットされます。
  - ・ UHF\_BlockWrite2 メソッド実行時に、writeContinue=1 をセットした場合  
→ 今回の書き込み結果が NACK 応答となった場合に値がセットされます。  
エラーの内容は、(表 1)をご参照ください。
  - ・ UHF\_Encode メソッド実行時に、NACK 応答となった場合  
→ どのメソッド実行時にエラーが発生したかがセットされます。  
エラーの内容は、(表 3)をご参照ください。
- (4) NACK 応答の 9 バイト目のエラーコード 4 は、UHF\_BlockWrite2 メソッド実行時に、writeContinue=1 をセットした場合、エラーコード 3 に「0Ah」がセットされている場合に、値がセットされます。  
エラーコード 4 の内容は、主にタグアクセスの際に発生するエラーです。  
エラーの内容は、(表 2)をご参照ください。

(表 1) エラーコード 1、または、[UHF\_BlockWrite2 メソッド] 実行時のエラーコード 3

種別	エラーコード	シンボル	説明
RF タグ アクセス 異常	01h	CMD_CRC_ERROR	RF タグから受信したデータの CRC を 検査した結果、一致しない。
	02h	CMD_TIME_OVER	RF タグからの受信データが途中で 途切れた。
	03h	CMD_RX_ERROR	アンチコリジョン処理中にエラーが 発生した。
	04h	CMD_RXBUSY_ERROR	RF タグからの応答がない。
	07h	CMD_ERROR	コマンド実行中にリーダーライタ内部で エラーが発生。
	0Ah	CMD_UHF_IC_ERROR	タグアクセス時、リーダーライタ内蔵チップが 返すエラー。 エラーコード 2 を参照。
キャリア 異常	60h	CMD_LBT_ERROR	キャリアセンス時、タイムアウトエラーでキ ャリアを送信出来なかった。
アンテナ 接続異常	68h	CMD_ANT_ERROR	起動時のアンテナ断線確認から、アンテナ断 線エラーになった場合に、RF タグにアクセ スするコマンドを送信した場合のエラー
コマンド 形式異常	42h	SUM_ERROR	上位機器から送信されたコマンドの SUM 値が不正。
	44h	FORMAT_ERROR	上位機器から送信されたコマンドの フォーマットまたはパラメータが不正。

(表 2) エラーコード 2、または、[UHF\_BlockWrite2 メソッド] 実行時のエラーコード 4

種別	エラーコード	説明
RF タグ エラー	01h	サポートされていません
	02h	権限が不十分
	03h	メモリオーバーラン
	04h	メモリロック
	05h	暗号違反
	06h	コマンドはカプセル化されません
	07h	レスポンスバッファオーバーフロー
	08h	セキュリティアウト
	0Bh	不十分な電力
	0Fh	非特定のエラー
	20h	Write に失敗
	22h	Kill に失敗
	23h	Lock に失敗
	80h	検出されない
	81h	ハンドル取得失敗
	82h	Access パスワードエラー
	90h	CRC エラー
上記以外	処理失敗	

(表 3) [UHF\_Encode メソッド] 実行時のエラーコード 3

種別	エラーコード	説明
エラー発生箇所	01h	Reserved 領域への書き込み時にエラー
	02h	EPC(UII)領域への書き込み時にエラー
	03h	User 領域への書き込み時にエラー
	05h	Lock コマンド発行時にエラー

## 12.2 ResponseRFID イベントパラメータ

### 12.2.1 リーダライタの自動読み取りモード

動作モード	ResponseRFID イベントパラメータ					
連続インベントリ モード	e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI	
	UHFInventory ※RF タグ読み取りデータを返すイベント	—	○	○	○	
		e.MemBankData		e.BinaryData		
		—				—
		e.InputData		e.TextData		
	○				—	
	e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI	
	UHFInventoryCount ※RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK を返すイベント (※1)	—	—	—	—	
		e.MemBankData		e.BinaryData		
		—				○ (※2)
		e.InputData		e.TextData		
	○				—	
	e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI	
	UHFInventoryAntenna Change ※アンテナ切り替え終了時、ACK を返すイベント (※3)	—	—	—	—	
		e.MemBankData		e.BinaryData		
		—				—
e.InputData		e.TextData				
○				—		
e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI		
UHFInventoryCarrierSense ※キャリア検知時、ACK を返すイベント (※4)	—	—	—	—		
	e.MemBankData		e.BinaryData			
	—				○ (※5)	
	e.InputData		e.TextData			
○				—		

※1：リーダライタが、「自動読み取りモード時の読み取りサイクル終了時のレスポンス：返す」に設定されている場合に、本 ResponseRFID イベントが発生します。  
設定の詳細は、[6.8 UHF\_GetEPCParam メソッド]または[6.16 UHF\_SetEPCParam メソッド]を参照ください。

※2：RF タグ読取枚数 (2 バイト、LSB ファースト)、および、読み取り時のキャリアのチャンネル番号(1 バイト)が、e.BinaryData にセットされます。

※3：リーダライタが「アンテナ自動切替終了時のレスポンス：返す」に設定されており、「アンテナ切替方式」が「制御する」または「制御する (複数アンテナを一つのアンテナとして扱う)」に設定されている場合に、本 ResponseRFID イベントが発生します。

「アンテナ自動切替終了時のレスポンス」の設定は、[6.8 UHF\_GetEPCParam メソッド]または[6.16 UHF\_SetEPCParam メソッド]を参照ください。

「アンテナ切替方式」の設定は、[6.4 UHF\_GetAntennaParam メソッド]または[6.12 UHF\_SetAntennaParam メソッド]を参照ください。

※4：リーダライタが「キャリアセンスにかかった時のレスポンス：返す」に設定されており、キャリアセンス時に他のリーダライタからの電波を検知したタイミングで、本 ResponseRFID イベントが発生します。

※5：キャリアセンスにかかった時のチャンネル番号が、e.BinaryData にセットされます。

動作モード	ResponseRFID イベントパラメータ					
連続インベントリ リードモード	e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI	
	UHFInventoryRead ※RF タグ読み取りデータを返すイベント	○/-(※1)	○	○	○	
		e.MemBankData		e.BinaryData		
		-		-		
		e.InputData		e.TextData		
	○		-			
	e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI	
	UHFInventoryReadCount ※RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK を返すイベント (※2)	-	-	-	-	
		e.MemBankData		e.BinaryData		
		-		○ (※3)		
		e.InputData		e.TextData		
	○		-			
	e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI	
	UHFInventoryRead AntennaChange ※アンテナ切り替え終了時、ACK を返すイベント (※4)	-	-	-	-	
		e.MemBankData		e.BinaryData		
		-		-		
e.InputData		e.TextData				
○		-				
e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI		
UHFInventoryReadCarrier Sense ※キャリア検知時、ACK を返すイベント (※5)	-	-	-	-		
	e.MemBankData		e.BinaryData			
	-		○ (※6)			
	e.InputData		e.TextData			
○		-				

- ※1 : tidRead = true で実行した場合、読み取った RF タグの TID が e.TID にセットされます。  
tidRead = false で実行した場合、e.TID はセットされません。
- ※2 : リーダライタが、「自動読み取りモード時の読み取りサイクル終了時のレスポンス：返す」に設定されている場合に、本 ResponseRFID イベントが発生します。  
設定の詳細は、[6.8 UHF\_GetEPCParam メソッド]または[6.16 UHF\_SetEPCParam メソッド]を参照ください。
- ※3 : RF タグ読取枚数 (2 バイト、LSB ファースト)、および、読み取り時のキャリアのチャンネル番号(1 バイト)が、e.BinaryData にセットされます。
- ※4 : リーダライタが「アンテナ自動切替終了時のレスポンス：返す」に設定されており、「アンテナ切替方式」が「制御する」または「制御する (複数アンテナを一つのアンテナとして扱う)」に設定されている場合に、本 ResponseRFID イベントが発生します。  
  
「アンテナ自動切替終了時のレスポンス」の設定は、[6.8 UHF\_GetEPCParam メソッド]または[6.16 UHF\_SetEPCParam メソッド]を参照ください。  
「アンテナ切替方式」の設定は、[6.4 UHF\_GetAntennaParam メソッド]または[6.12 UHF\_SetAntennaParam メソッド]を参照ください。
- ※5 : リーダライタが「キャリアセンスにかかった時のレスポンス：返す」に設定されており、キャリアセンス時に他のリーダーライタからの電波を検知したタイミングで、本 ResponseRFID イベントが発生します。
- ※6 : キャリアセンスにかかった時のチャンネル番号が、e.BinaryData にセットされます。

12.2.2 リーダライタの制御

参照項	ResponseRFID イベントパラメータ			
	e.SendCommand	e.TextData	e.InputData	e.BinaryData
3.1	GetError	—	○	—
3.2	GetROMVersion	○	○	○
3.3	GetCHIPVersion	○	○	○
3.4	SetTransmitSignal	—	○	—
3.5	CallLEDBuzzer	—	○	—
3.7	CallBuzzer (※1)	—	○	—
3.8	InitFLASH	—	○	—
3.9	UHFGetHandle	—	○	○

※いずれの場合も、e.TID、e.PC、e.EPC、e.RSSI、e.MemBankData はセットされません。

※1：needResponse=false を指定してメソッドを実行した場合、ResponseRFID イベントは発生しません。

12.2.3 リーダライタ動作モードの設定

参照項	ResponseRFID イベントパラメータ			
	e.SendCommand	e.TextData	e.InputData	e.BinaryData
4.1	GetActionMode	—	○	—
4.2	SetActionMode	—	○	—

※いずれの場合も、e.TID、e.PC、e.EPC、e.RSSI、e.MemBankData はセットされません。

12.2.4 リーダライタ FLASH の設定

参照項	ResponseRFID イベントパラメータ			
	e.SendCommand	e.TextData	e.InputData	e.BinaryData
5.1	ReadFLASH	—	○	—
5.2	WriteFLASH	—	○	—

※いずれの場合も、e.TID、e.PC、e.EPC、e.RSSI、e.MemBankData はセットされません。

12.2.5 リーダライタ設定

参照項	ResponseRFID イベントパラメータ			
	e.SendCommand	e.TextData	e.InputData	e.BinaryData
6.1	UHFGetInventoryParam	—	○	—
6.2	UHFGetSelectParam	—	○	—
6.3	UHFGetExpandSelectParam	—	○	—
6.4	UHFGetAntennaParam	—	○	—
6.5	UHFGetTxPowerParam	—	○	—
6.6	UHFGetFrequencyParam	—	○	—
6.7	UHFGetRFTAGComParam	—	○	—
6.8	UHFGetEPCParam	—	○	—
6.9	UHFSetInventoryParam	—	○	—
6.10	UHFSetSelectParam	—	○	—
6.11	UHFSetExpandSelectParam	—	○	—
6.12	UHFSetAntennaParam	—	○	—
6.13	UHFSetTxPowerParam	—	○	—
6.14	UHFSetFrequencyParam	—	○	—
6.15	UHFSetRFTAGComParam	—	○	—
6.16	UHFSetEPCParam	—	○	—
6.17	UHFSetAccessPassword	—	○	—

※いずれの場合も、e.TID、e.PC、e.EPC、e.RSSI、e.MemBankData はセットされません。

12.2.6 RF タグとの通信

参照項	ResponseRFID イベントパラメータ					
	e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI	
7.1	UHFInventoryCmd ※RF タグ読み取りデータを返すイベント	—	○	○	○	
		e.MemBankData		e.BinaryData		
		—	—	—	—	
		e.InputData		e.TextData		
		—	—	—	—	
		e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI
	UHFInventoryCmdCount ※RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK を返すイベント	—	—	—	—	
		e.MemBankData		e.BinaryData		
	—	—	○	—		
	e.InputData		e.TextData			
	○	—	—	—		
7.2	UHFInventoryReadCmd ※RF タグ読み取りデータを返すイベント	○/— (※1)	○	○	○	
		e.MemBankData		e.BinaryData		
		○	—	—	—	
		e.InputData		e.TextData		
		○	—	—	—	
		e.SendCommand	e.TID	e.PC	e.EPC	e.RSSI
	UHFInventoryReadCmdCount ※RF タグ読取枚数および読み取り時のキャリアのチャンネル番号を含む ACK を返すイベント	—	—	—	—	
		e.MemBankData		e.BinaryData		
	—	—	○	—		
	e.InputData		e.TextData			
	○	—	—	—		
7.3	UHFRead	—	—	—	—	
		e.MemBankData		e.BinaryData		
		○	—	—	—	
		e.InputData		e.TextData		
		○	—	—	—	

※1 : tidRead = true で実行した場合、読み取った RF タグの TID が e.TID にセットされます。  
tidRead = false で実行した場合、e.TID はセットされません。

参照項	ResponseRFID イベントパラメータ			
	e.SendCommand	e.BinaryData	e.InputData	e.TextData
7.4	UHFWrite	—	○	—
7.5	UHFBlockWrite	—	○	—
7.6	UHFlock	—	○	—
7.7	UHFKill	—	○	—
7.8	UHFBlockErase	—	○	—
7.9	UHFBlockWrite2	—	○	—
7.10	UHFEncode	—	○	—
7.11	UHFThroughCmd	○	○	—

※いずれの場合も、e.TID、e.PC、e.EPC、e.RSSI、e.MemBankData はセットされません。

## 変更履歴

Ver. No.	日付	内容
1.00	2019/2/20	新規作成
1.10	2019/12/18	以下の4種のメソッドを追加 ・ SetTransmitSignal メソッド ・ UHF_BlockErase メソッド ・ UHF_BlockWrite2 メソッド ・ UHF_Encode メソッド
1.11	2020/4/20	リーダライタのROMバージョン1.090以降で「インベントリタイムアウト時間」と「コマンドタイムアウト時間」を設定時の注意事項を追加
1.20	2023/6/19	メソッド追加 ・ UHF_ThroughCmd ・ UHF_GetHandle 列挙体追加 ・ UHF_ThroughCmdType メソッド引数用クラス追加 ・ UHF_ThroughCmdOption
1.21	2026/2/3	バージョンアップ履歴を追加 ソフトウェア使用許諾契約の内容変更

---

---

タカヤ株式会社 RF 事業部

[URL] <https://www.takaya.co.jp/>

[Mail] [rfid@takaya.co.jp](mailto:rfid@takaya.co.jp)

---

---

仕様については、改良のため予告なく変更する場合がありますので、あらかじめご了承ください。