# USICF プログラミングマニュアル

**V2.7.1.6**

# I. USICF DLL

USICF DLL はネイティブ USI.DLL の .NET ラッパです。**U**nitech **S**canner **I**nterface for .NET **C**ompact **F**ramework の頭文字を取って名前が付けられています。 そのファイル名は "**USICF.dll**"です。各種の **メソッド、プロパティ**と**イベント**があり、開発者は Unitech PA シリーズポータブルターミナルのスキャナを動作させるために使用することができます。

サンプルプログラムには USICF の使用方法が含まれています。 C# と VB.NET バージョンがあります。

- メソッド

USICF DLL は以下のメソッドがあります。

**a. bool SetWorkingMode(WorkingMode mode)**

パラメータ:
*mode* – スキャナの動作モード。有効な値は *WorkingMode* 列挙で定義されます:

| | |
|---|---|
| SWM_BARCODE | バーコードスキャンモード |
| SWM_IMAGE | イメージキャプチャモード |

戻り値:
true = 成功
false = 失敗

このメソッドはスキャナの動作モードをバーコードスキャナもしくはイメージキャプチャモードのいずれかにセットするために使用されます。

**b. bool EnableScanner(bool enable)**

パラメータ:
*enable* – スキャナをイネーブルまたはディスエーブルにするためのフラグ:

| | |
|---|---|
| true | スキャナをイネーブルにする |
| false | スキャナをディスエーブルにする |

戻り値:
true = 成功

false = 失敗

このメソッドはスキャナをイネーブルもしくはディスエーブルにするために使用されます。

**c. bool StartAutoScan(uint interval)**

パラメータ:
*interval* – 二つのスキャン間の時間(ミリ秒)

戻り値:
true = 成功
false = 失敗

このメソッドは自動スキャンをスタートするために使用されます。

**d. void StopAutoScan()**

パラメータ:
なし

戻り値:
なし

このメソッドは自動スキャンを停止するために使用します。

**e. bool SaveImage(string fileName, int compressionRatio)**

パラメータ:
*fileName* – イメージを保存したいファイルの名前。パスを含めることができます。ファイルの拡張子は以下のいずれかを指定しなければなりません。

| | |
|---|---|
| .bmp | Bitmap フォーマット |
| .jpeg または .jpg | JPEG フォーマット |
| .tiff または .tif | TIFF フォーマット |

*compressionRatio* – イメージの圧縮比。これは拡張子 “.jpeg” または “.jpg”を指定した JPEG フォーマットのみに使用されます。

戻り値:
true = 成功
false = 失敗

このメソッドはキャプチャしたイメージをターミナルのファイルに保存するために使用されます。イメージフォーマットは、 *fileName* パラメータで指定したファイル拡張子で決まります。

**f. bool SetHamster(byte cmdID, byte setting)**

パラメータ:
*cmdID* – Hamster デコーダチップのコマンド ID

*setting* – このコマンドのオペランド

戻り値:
true = 成功
false = 失敗

このメソッドは Hamster デコーダチップにスキャナコマンドを送るために使用されます。コマンドの完全なリストについては、 “PA96x プログラミングマニュアル”をご覧下さい。

**g. TerminatorType GetTerminator()**

パラメータ:
なし

戻り値:
以下の値の一つが *TerminatorType* 列挙で定義されます:
TERMINATOR_ENTER
TERMINATOR_RETURN
TERMINATOR_LINEFEED
TERMINATOR_NONE
TERMINATOR_ENTERENTER

このメソッドはスキャナのターミネータを得るために使用されます。

**h. bool SetTerminator(TerminatorType terminator)**

パラメータ:
*terminator* – *TerminatorType* 列挙で定義された値の一つ。上記参照。

戻り値:
true = 成功
false = 失敗

このメソッドはスキャナのターミネータをセットするために使用されます。

**i. bool SaveCurrentSettings()**

パラメータ:
なし

戻り値:
true = 成功
false = 失敗

このメソッドはスキャナの現在の設定を保存するために使用されるので、設定はターミナルの電源を切り、再度入れても持続します。

**j. bool SaveSettingsToFile(string fileName)**

パラメータ:
*fileName* – 設定を保存したいファイル名。ファイルの拡張子は常に“.usi”.

戻り値:
true = 成功
false = 失敗

このメソッドは現在の設定をファイルに保存するために使用します。

**k. bool LoadSettingsFromFile(string fileName)**
**bool LoadSettingsFromFile(string fileName, bool formulaOnly)**

パラメータ:
*fileName* – 設定を読み込みたいファイル名。

*formulaOnly* – TRUE の場合、データ編集式のみが読み込まれ、他の設定は変更されないままです。

戻り値:
true = 成功
false = 失敗

このメソッドはファイルから設定を読み込むために使用されます。二つのバージョンがあることにご注意下さい。

**l. EchoMode GetGoodReadEcho(out string soundFileName)**

パラメータ:

*soundFileName* – 戻り時に、現在の Good-Read-Echo モードが GRE_PLAYSOUND と同じ場合、サウンドファイルのファイル名を含みます。Good-Read-Echo モードが GRE_BEEP または GRE_NONE と同じ場合は無効です。

戻り値:
*EchoMode* 列挙で定義された以下の値の一つ:
GRE_PLAYSOUND
GRE_BEEP
GRE_NONE

このメソッドはスキャナの Good-Read-Echo モードとサウンドファイル名(適当な場合)を得るために使用します。

**m. bool SetGoodReadEcho(EchoMode mode, string soundFileName)**

パラメータ:
*mode* – *EchoMode* 列挙で定義された値の一つ。上記参照。
*soundFileName* – *mode* が GRE_PLAYSOUND にセットされた場合、サウンドファイル(.WAV ファイル)の名前

戻り値:
true = 成功
false = 失敗

このメソッドはスキャナの Good-Read-Echo モードとサウンドファイル名(適当な場合)をセットするために使用します。GRE_BEEP または GRE_NONE モードについては、*soundFileName* について *null* (C#) または *vbNullString* (VB.NET)を渡すことができます。

**n. bool Reset()**

パラメータ:
なし

戻り値:
true = 成功
false = 失敗

このメソッドはスキャナを動作モードにセットし、通信コントロールをリセットするために使用されます。

**o. bool SetReplaceDataChar(bool enable, byte oldChr, byte newChr)**

パラメータ:
Enable – データ置換機能を有効/無効にする
oldChr – 置き換えられる古い文字
newChr – 置き換える新しい文字

戻り値:
true = 成功
false = 失敗

Enable パラメータが true にセットされた場合、バーコードの oldChar は newChr に置き換わります。

**p. bool GetReplaceDataChar(ref byte p_oldChr, ref byte p_newChr)**

パラメータ:
p_oldChr – 置き換えられる古い文字を検索
p_newChr – 置き換える新しい文字を検索

戻り値:
true = 成功
false = 失敗

True が戻ると置換が有効であることを意味し、そうでなければ無効です。p_oldChr と p_newChr は、NULL がセットされます。

**q. Void Dispose()**

パラメータ:
USI の登録を解除してリソースを解放します。

戻り値:
なし

- プロパティ

USICF DLL は以下のプロパティを持っています。

**a. IsAutoScanning**

Type: bool

Value:
true = 自動スキャンがオン
false = 自動スキャンがオフ

このプロパティは自動スキャンがオンかどうかを決めます。これは読み取り専用プロパティです。

**b. PromptMessage**

Type: bool
Value:
true = USI プロンプトメッセージ有効
false = USI プロンプトメッセージ無効

このプロパティはネイティブ USI DLL から作業情報のポップアップウインドウを有効/無効にするために使用されます。これは書き込み専用プロパティです。

**c. ErrorMessage**

Type: bool
Value:
true = USI エラーメッセージ有効
false = USI エラーメッセージ無効

このプロパティはネイティブ USI DLL からエラー情報ポップアップウインドウを有効/無効にするために使用されます。これは書き込み専用プロパティです。

- イベント

USICF DLL は以下のイベントをサポートしています。

**a. DataReady**

このイベントはスキャナがバーコードを正しくデコードしたときに発生します。アプリケーションは、ユーザがバーコードをスキャンするときに通知するためにこのイベントを見ることができます。実際のバーコード文字列とバーコードタイプは、このイベントと共に入る ***USIEventArgs*** パラメータから得ることができます。

**USIEventArgs のプロパティ**

**i) BarcodeData**
Type: string
String としてバーコードデータを返します

**ii) BarcodeLength**
Type: uint
データ長を返します

**iii) BarcodeType**
Type: uint
uint としてバーコードタイプを返します

**iv) BarcodeName**
Type: string
バーコードシンボル名を返します

**v)** PrimalDataLength
Type: uint
最初のバーコードデータ長を戻します

**vi)** PrimalData
Type: byte[]
バイナリでバーコードの最初のデータを戻します

**b. ImageTrigger**

このイベントはユーザがイメージをキャプチャするためにトリガを押したときに発生します。スキャナはこのイベントを発生する前にはイネーブルにして、イメージキャプチャモードになければなりません。アプリケーションは、通常はイメージ読み取りと表示のための準備を行うことによってこのイベントを処理します。

**c. ImageProgress**

このイベントは *ImageTrigger* イベントの後で発生し、アプリケーションにイメージ読み取りの進行をレポートします。進行のパーセントが、このイベントと共に入る ***ImageProgressEventArgs*** パラメータから読むことができます。

**ImageProgressEventArgs のプロパティ:**

**i) Percentage**
Type: int
int として進行をパーセントで返します。

**d. ImageReady**

このイベントは、イメージキャプチャされた全体がスキャナから読み込まれたことをアプリケーションに通知します。 *SaveImage* メソッドはイメージをファイルに保存するために呼び出すことができます。

**e. ErrorEvent**

名前の通り、このイベントはエラーが発生したときにアプリケーションに通知します。エラーの情報はこのイベントと共に入る ***ErrorEventArgs*** パラメータから読むことができます。

**ErrorEventArgs のプロパティ:**

**i) ErrorInfo**
Type: string
エラーの説明を戻します。

**ii) ErrorCode**
Type: uint
エラーが USI エラーか、システムエラーかによって、USI エラーコードまたはシステムエラーコードのいずれかを戻します。

# II. C# プログラミングモデル

C# アプリケーションは USICF DLL を呼ぶために以下のプログラミングモデルを使用することができます。

```
using USICF;

public class Form1 : System.Windows.Forms.Form
{
      ..............

      private USIClass myUSI;

      public Form1()
      {
            //
            // Required for Windows Form Designer support
            //
            InitializeComponent();

            //
            // TODO: Add any constructor code after
            //InitializeComponent call
            //
            myUSI = new USIClass(this);
            myUSI.DataReady += new
            USIClass.USIEventHandler(myUSI_DataReady);
            myUSI.ErrorEvent += new
            USIClass.ErrorEventHandler(myUSI_ErrorEvent);

      }

      private void myUSI_DataReady(object sender, USIEventArgs e)
      {
            MessageBox.Show(e.BarcodeData);
      }

      private void myUSI_ErrorEvent(object sender, ErrorEventArgs
      e)
      {
            MessageBox.Show(e.ErrorInfo, "USI Error",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation,
            MessageBoxDefaultButton.Button1);
      }

      ........................
}
```

# III. VB.NET プログラミングモデル

VB.NET アプリケーションは、USICF DLL を呼ぶために以下のプログラミングモデルを使用することができます。

```
Imports USICF

Public Class Form1
      Inherits System.Windows.Forms.Form
      ................
      Private WithEvents myUSI As USIClass

#Region " Windows Form Designer generated code "

    Public Sub New()

        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent()
        'call
        myUSI = New USIClass(Me)

    End Sub

      ................

#End Region

    Private Sub myUSI_DataReady(ByVal sender As Object, ByVal e
    As USIEventArgs) Handles myUSI.DataReady
        MessageBox.Show(e.BarcodeData)
    End Sub

    Private Sub myUSI_ErrorEvent(ByVal sender As Object, ByVal e
    As ErrorEventArgs) Handles myUSI.ErrorEvent
        MessageBox.Show(e.ErrorInfo, "USI Error",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation, _
        MessageBoxDefaultButton.Button1)
    End Sub

      ................

End Class
```

# IV. サンプルプログラム

すべてのサンプルプログラムは、 “**Bin**” フォルダ下の“**USICF.dll**” ファイルを参照します。

## 1. CSDemo & VBDemo

シングルフォームで USICF を使用する方法を示します。また各種のメソッドの使いかたを示します。

## 2. CSDemo2 & VBDemo2

マルチフォームで USICF を使用する方法を示します。

## 3. CSDemo3 & VBDemo3

イメージキャプチャの扱い方を示します。