FSU1指紋認証
開発キット

ユーザーズマニュアル

第1.0版

RATOC Systems, Inc.

-目次-

1-1．はじめに

SREX-FSU1 用指紋認証開発キット（以下、FSU1-SDK）は、SREX-FSU1 を使用した指紋認証システムの開発を支援するライブラリソフトウェアです。

指紋データの取得、および、照合などの認証プロセスを、自作のアプリケーションプログラムに組み込むことが、可能となります。

1-2．動作環境

パソコン	PC-AT 互換機、NEC PC98-NX シリーズ
OS	Windows2000/XP
対応デバイス	SREX-FSU1（複数接続には対応しておりません）
開発環境	Visual C++ 6.0 以上/Visual BASIC 6.0 以上

1-3．ファイル構成

FSU1LibrarySetup.exe	32 ビットライブラリセットアッププログラム
Sample	サンプルアプリケーションプログラム

1-4．注意事項

①指紋認証技術は完全な本人認証・照合を保証するものではありません。当社では本ソフトウェアを使用されたこと、または使用できなかったことによって生じるいかなる損害に関しても、一切責任を負いかねますのであらかじめご了承ください。

②本書の内容に関しましては、将来予告なしに変更することがあります。
また、本書の内容につきましては万全を期して作成しましたが、万一不審な点や誤りなどお気づきになりましたらご連絡願います。

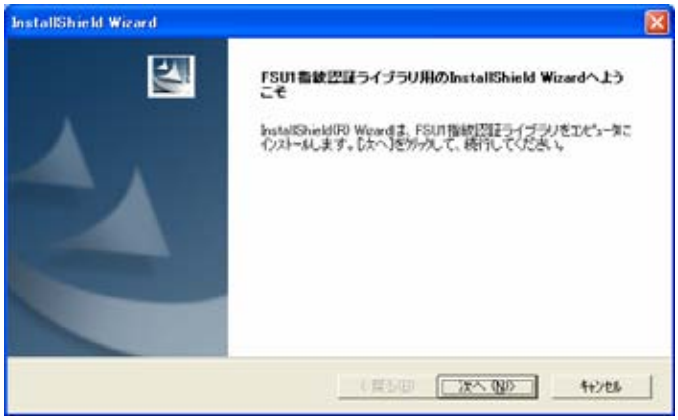
③本製品は日本国内仕様となっており、海外での保守およびサポートは行っておりません。

2-1. セットアップ

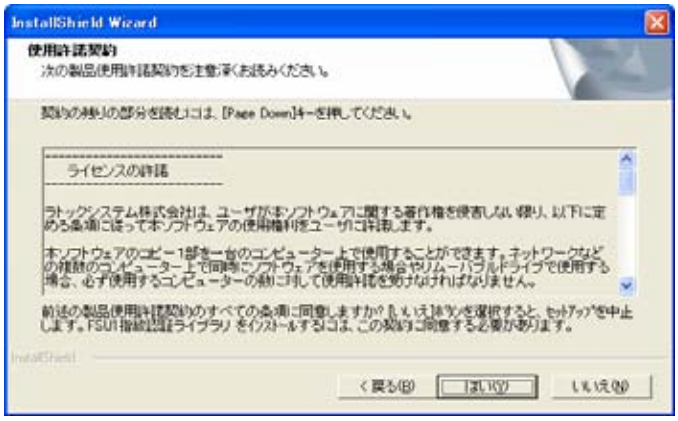
本 FSU1-SDK を使用するには、指紋認証デバイス SREX-FSU1 が必要です。あらかじめ、「SREX-FSU1 ユーザーズマニュアル」にしたがって、SREX-FSU1 のインストールを行ってください。

次に、FSU1-SDK のセットアップを行います。

1. FSU1LibrarySetup.exe を実行すると、右の画面が表示されます。セットアップを続行する場合は、「次へ」を押してください。



2. 使用許諾書の内容をご確認いただき、問題がなければ、「はい」を押してください。



3. 自動的にライブラリのインストールが行われた後、右の画面が表示されます。「完了」を押してください。

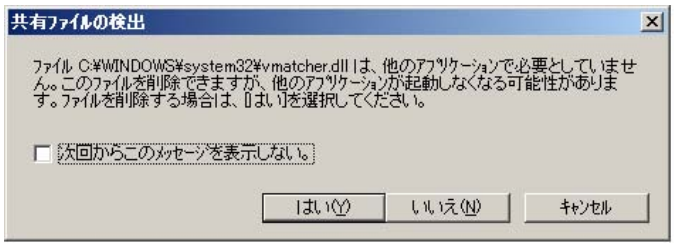
以上で、セットアップは完了です。

2-2．アンインストール

本 FSU1-SDK をアンインストールする場合は、「スタートメニュー」の「コントロールパネル」から「プログラムの追加と削除」を選択します。
「プログラムの追加と削除」の一覧から「FSU1 指紋認証ライブラリ」を選択し、「変更と削除」ボタンを押してアンインストールを行ってください。

1. 右の「ファイル削除の確認」画面が表示されますので、「OK」を押してください。

2. 右のような「共有ファイルの検出」画面が表示された場合、本 FSU1-SDK を使用した指紋認証アプリケーションを継続して使用する場合には、「いいえ」を押してファイルを削除しないようにしてください。
ファイルを必要としない場合は「はい」を押して削除します。
ファイルを削除していいかどうかわからない場合は、「いいえ」を選択してください。
右の画面が表示されない場合は、そのまま 3 へ進んでください。



3. 右の画面で「完了」を押してください。

以上で、アンインストールは完了です。

3-1．アプリケーション開発について

FSU1-SDK には、32 ビットアプリケーション開発を行う際の、ご参考として、サンプルアプリケーションが付属しております。本項では、サンプルアプリケーションを基に、アプリケーションの開発について、説明いたします。
本 FSU1-SDK には、２つの種類のサンプルが付属しています。

① １対１照合・・・指紋照合時において指紋データが複数ある場合、1 つの指紋データとの照合を行います。そのため、どの指紋データと照合を行うかを特定するために固有の ID 等を指紋データに対して割り付ける必要があります。

② １対多照合・・・指紋照合時において指紋データが複数ある場合、複数の指紋データと照合を行います。照合不一致の場合には、照合一致するまでデータベース内の別の指紋データとの照合を順次、行っていきます。指紋データに対して固有 ID 等はとくに必要ありませんが、指紋データ登録時にデータベース内の指紋データに同じデータが存在することの無いよう以下のような注意が必要です。

○ １対多照合を行う上でのアプリケーション作成時の注意点

・1 対多照合を行う場合、他人受理率（他人を本人と判定してしまう確率）が高まる可能性があります。
この対策のため、以下の 3 点を行ってください。

1．指紋登録時に、あらかじめ登録してあるデータベース内の全ての指紋データと照合を行い、一致する場合には、その指紋データを採用しない。
2．指紋登録時に、指紋データの品質の悪いデータについては採用しない。
3．指紋照合時、セキュリティレベルを高く設定しておく。

・1 対多照合を行う場合、照合時、データベース内の複数ある指紋データとの照合を順番に行っていくため、データベース内の登録データ件数が多くなるにつれ、照合に時間がかかる可能性があります。
この対策のため、以下のことを行ってください。

1．照合にかかるトータル時間を短縮するため、データベース内のすべてのデータをあらかじめメモリに展開しておく。

○ アプリケーション開発の注意点

■ 本人拒否率（FRR）と他人受理率（FAR）に関して

「本人拒否率」とは、本人を他人と判定してしまうエラー率のことをいい、「他人受理率」とは、他人を本人と判定してしまうエラー率のことをいいます。
一般的に、「本人拒否率」と、「他人受理率」は相関関係にあり、「他人受理率」を下げると「本人拒否率」は上がり、反対に、「本人拒否率」を下げると他人受理率は上がります。
セキュリティ性を重視する場合、「他人受理率」を小さくする必要がありますが、その場合「本人拒否率」が上がってしまい、使い勝手に影響が出てきます。反対に、利用者の利便性を重視するために「本人拒否率」を下げると、「他人受理率」が上がってしまい、セキュリティ性が低下してしまうという特徴を持っています。
セキュリティ性を保ちつつ、実際にスムーズに運用していくためには、この相関関係を十分に考慮し、導入される現場のニーズにあったアプリケーション開発が必要となります。

■ 本人拒否率の低減

指紋照合時、本人であるにもかかわらず照合一致しない場合があります。
この本人拒否率を下げるため、以下のような点に注意してアプリケーションを作成する必要があります。
1．指紋登録時に、指紋データの品質の悪いデータについては採用しない。
2．指紋登録時、一つの指紋について、複数個の指紋データを登録しておく。
3．指紋登録時、複数の指の指紋データを登録しておく。

■ 指紋データについて

本 FSU1-SDK では、指紋の特徴点データ（Minutiae data）を扱います。実指紋データ（Raw data）を扱うことは出来ません。そのため、指紋データからの指紋復元は出来ません。

■ 指紋データサイズについて

指紋データのサイズは個人差によりますが、通常、数十バイトから数百バイトになります。最大サイズは、1024 バイトです。

○ Visual C++で使用する場合

Visual C++でライブラリ関数を使用するための C ヘッダファイル（FSU1Imp.h)、ライブラリファイル（FSU1Lib.lib）を用意しています。
プロジェクトに上記 2 つのファイルを追加し、C ヘッダファイルをインクルードすることにより、ライブラリ関数を呼び出してください。
ライブラリ関数のインポート宣言は以下のとおりです。（FSU1Imp.h より抜粋）
※ユーザ定義型についての記述方法は、ヘッダファイル FSU1Imp.h を参照してください。

```
#define FSU1LIB_API __declspec(dllimport)

FSU1LIB_API long APIENTRY RS_openDevice(long *hDev);
FSU1LIB_API long APIENTRY RS_closeDevice(long hDev);
FSU1LIB_API long APIENTRY RS_setNotifyType(long hDev,
                                           RS_NotifyType ntype,
                                           void *pCallbackProc,
                                           void *pdata);
FSU1LIB_API long APIENTRY RS_getFP(long hDev, FPPARAM_INFO *fpinfo);
FSU1LIB_API long APIENTRY RS_compareMinu(long hDev,
                                           BYTE *Minu1,
                                           long nMinu1,
                                           BYTE *Minu2,
                                           long nMinu2,
                                           COMPARE_PARAM *cpparam);
FSU1LIB_API long APIENTRY RS_showFPDialog(FPDIALOG_INFO *dialoginfo);
FSU1LIB_API long APIENTRY RS_hideFPDialog(FPDIALOG_INFO *dialoginfo);
FSU1LIB_API long APIENTRY RS_getResult(long *result);
FSU1LIB_API long APIENTRY RS_getVersion(DLLVER_INFO *verInfo);
FSU1LIB_API long APIENTRY RS_terminateFP(long hDev);
```

○ Visual BASICで使用する場合

Visual BASIC でライブラリ関数を使用するための宣言が必要です。
Declare の記述方法は以下のとおりです。（FSU1.bas（FSU1.vb）より抜粋）
※ユーザ定義型についての記述方法は、VB サンプル内のモジュールファイル FSU1.bas（VB.NET の場合、FSU1.vb）を参照してください。

```
‘VB Declare 宣言 (VB6)
Declare Function RS_openDevice Lib "FSU1Lib.DLL" (ByRef hDev As Long) As Long
Declare Function RS_closeDevice Lib "FSU1Lib.DLL" (ByVal hDev As Long) As Long
Declare Function RS_setNotifyType Lib "FSU1Lib.DLL" (ByVal hDev As Long, ByVal
notifyType As Long, ByVal pCallbackProc As Long, ByVal pdata As Long) As Long
Declare Function RS_getFP Lib "FSU1Lib.DLL" (ByVal hDev As Long, ByRef fpinfo As
FPPARAM_INFO) As Long
Declare Function RS_compareMinu Lib "FSU1Lib.DLL" (ByVal hDev As Long,
    ByRef Minu1 As Byte, ByVal nMinu1 As Long,
    ByRef Minu2 As Byte, ByVal nMinu2 As Long,
    ByRef cpparam As COMPARE_PARAM) As Long
Declare Function RS_showFPDialog Lib "FSU1Lib.DLL" (ByRef dialoginfo As
FPDIALOG_INFO) As Long
Declare Function RS_hideFPDialog Lib "FSU1Lib.DLL" (ByRef dialoginfo As
FPDIALOG_INFO) As Long
Declare Function RS_getResult Lib "FSU1Lib.DLL" (ByRef result As Long) As Long
Declare Function RS_getVersion Lib "FSU1Lib.DLL" (ByRef verInfo As DLLVER_INFO)
As Long
Declare Function RS_terminateFP Lib "FSU1Lib.DLL" (ByVal hDev As Long) As Long

‘VB Declare 宣言 (VB.NET)
Declare Function RS_openDevice Lib "FSU1Lib.DLL" (ByRef hDev As Integer) As Integer
Declare Function RS_closeDevice Lib "FSU1Lib.DLL" (ByVal hDev As Integer) As
Integer
Declare Function RS_setNotifyType Lib "FSU1Lib.DLL" (ByVal hDev As Integer, ByVal
notifyType As Integer, ByVal pCallbackProc As Integer, ByVal pdata As Integer) As
Integer
Declare Function RS_getFP Lib "FSU1Lib.DLL" (ByVal hDev As Integer, ByVal ptr As
IntPtr) As Integer
Declare Function RS_compareMinu Lib "FSU1Lib.DLL" (ByVal hDev As Integer,
    ByRef Minu1 As Byte, ByVal nMinu1 As Integer,
    ByRef Minu2 As Byte, ByVal nMinu2 As Integer,
    ByRef cpparam As COMPARE_PARAM) As Integer
Declare Function RS_showFPDialog Lib "FSU1Lib.DLL" (ByRef dialoginfo As
FPDIALOG_INFO) As Integer
Declare Function RS_hideFPDialog Lib "FSU1Lib.DLL" (ByRef dialoginfo As
FPDIALOG_INFO) As Integer
Declare Function RS_getResult Lib "FSU1Lib.DLL" (ByRef result As Integer) As
Integer
Declare Function RS_getVersion Lib "FSU1Lib.DLL" (ByRef verInfo As DLLVER_INFO)
As Integer
Declare Function RS_terminateFP Lib "FSU1Lib.DLL" (ByVal hDev As Integer) As
Integer
```

また、指紋データ取得中のステータスをユーザ定義メッセージにて受け取るための ActiveX コントロール（MboxFSU1.ocx）を用意しています。
VB6 の場合は、メニューの「プロジェクト」→「コンポーネント」→「コントロール」を選択、VB.NET 2003 の場合は、メニューの「ツール」→「ツールボックス アイテムの追加と削除」→「COMコンポーネント」を選択して、図 3-1、 図 3-2のように「FSU1 MBOX OLE Control module」が登録されていることを確認してください。一覧に登録が無い場合は、本マニュアルの 2-1．の セットアップを行ってください。
「FSU1 MBOX OLE Control module」にチェックを入れることで、ツールボックスに登録されますので、フォームに貼り付けて使用できます。
フォームに貼り付けたコントロールをダブルクリックすると、以下のようにメッセージ処理を行うプロシージャのコードが生成されます。
指紋データ取得中のステータス RS_FPState の値は、wParam（VB.NET の場合 e.wParam） により取得できます。詳しくはサンプルコードを参照してください。

```
 (VB6)
Private Sub MBOX1_OnMsgPost(ByVal wParam As Long, ByVal lParam As Long)

End Sub

 (VB.NET)
Private Sub AxMBOX1_OnMsgPost(ByVal sender As System.Object, ByVal e As
        AxMBOXLib._DMBOXEvents_OnMsgPostEvent) Handles AxMBOX1.OnMsgPost

End Sub
```

図 3-1. VB 6 プロジェクト

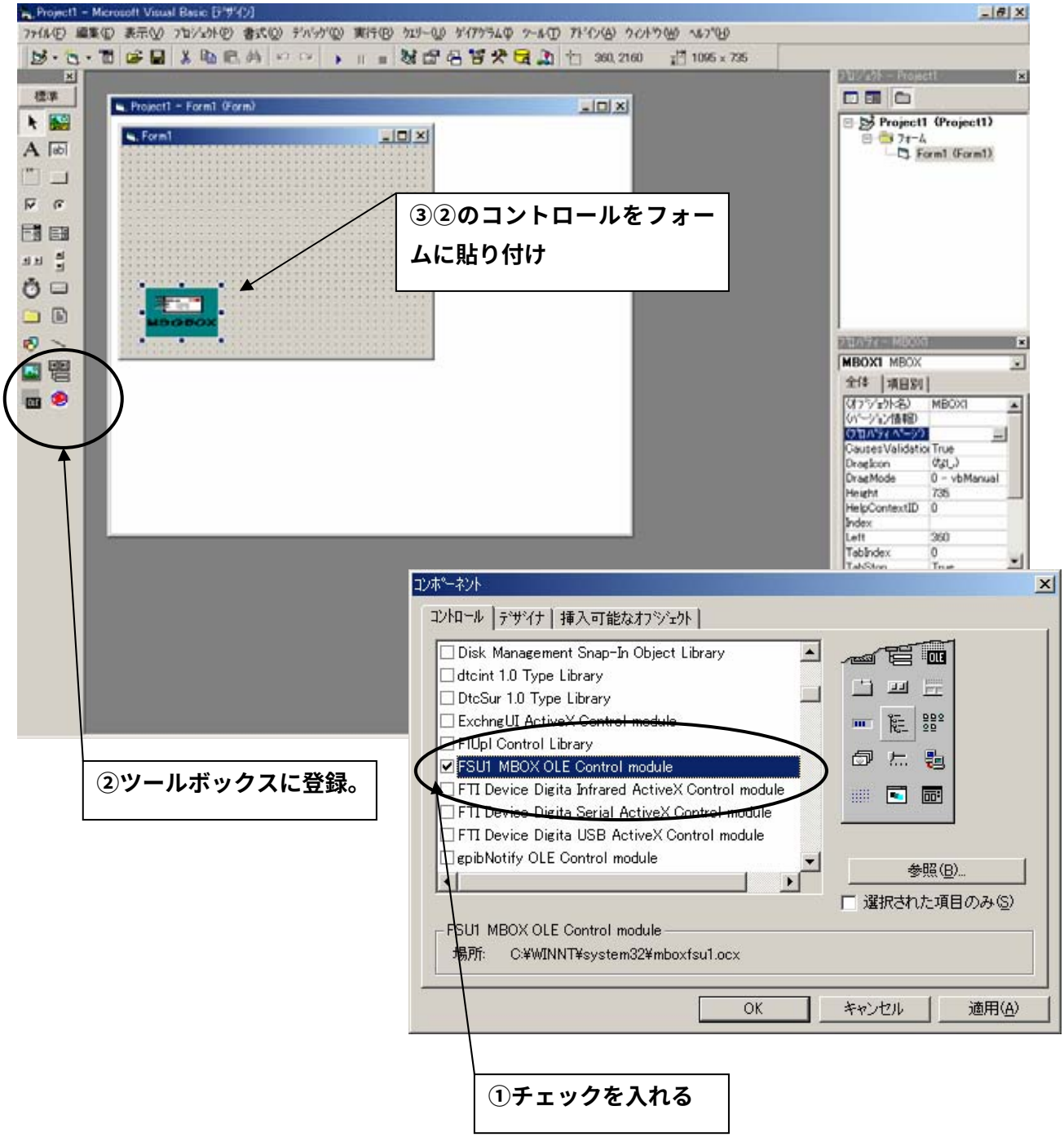
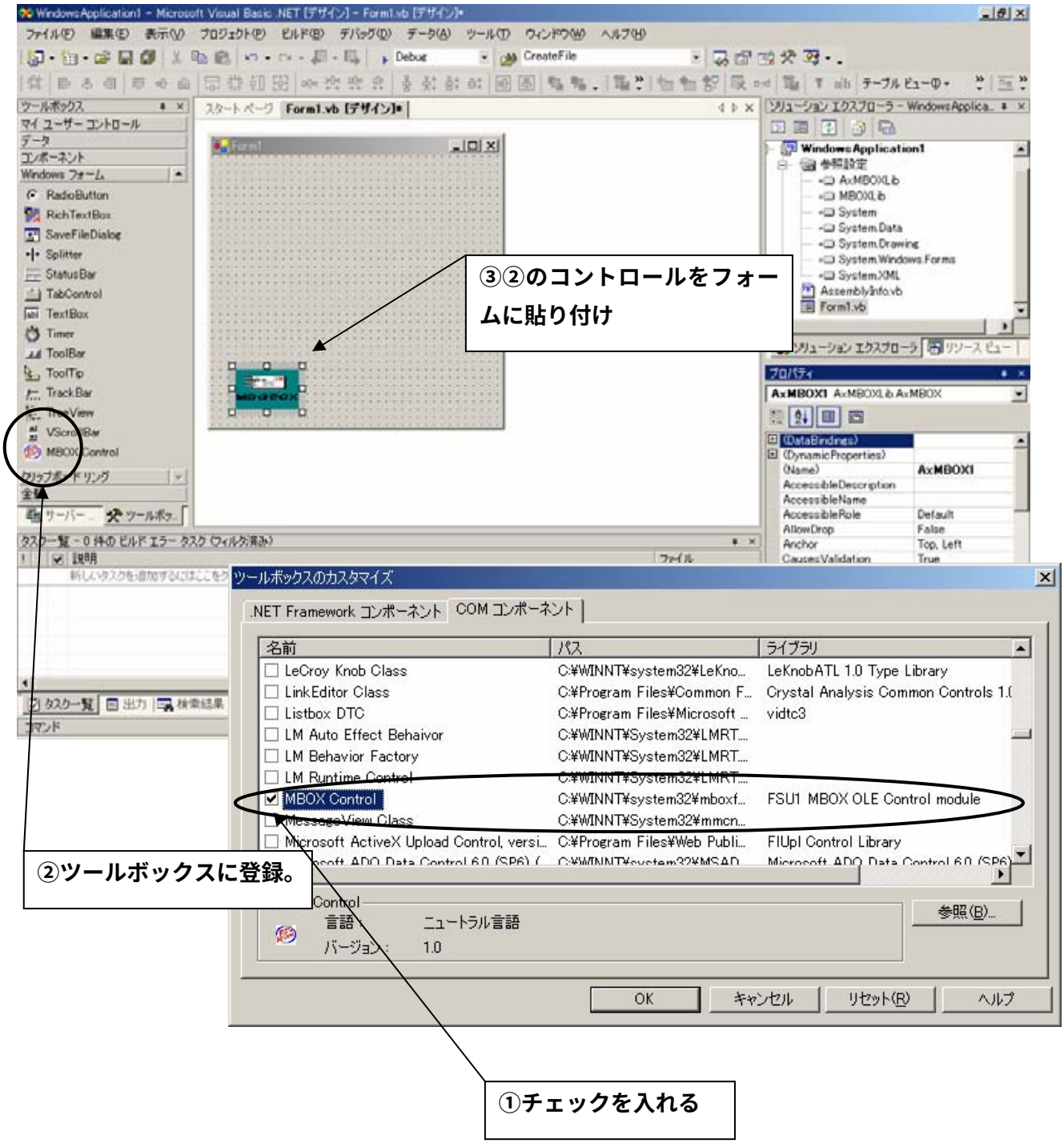
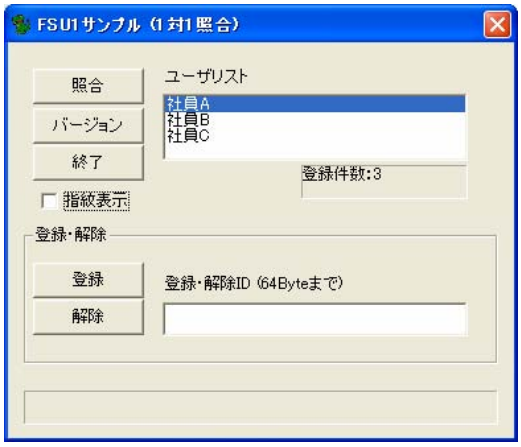


図 3-2. VB.NETプロジェクト

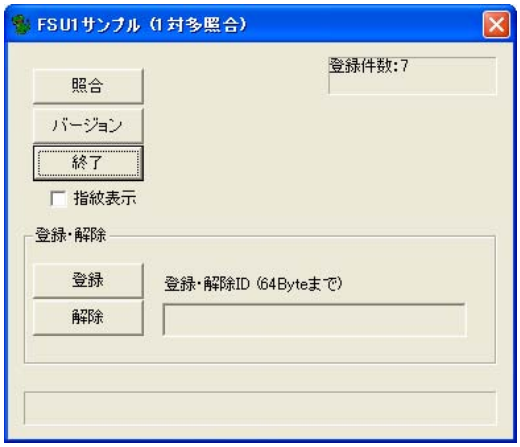


3-2．サンプルアプリケーションについて

下図はサンプルアプリケーションの画面です。
主な機能として、
「指紋登録」「指紋解除」「指紋照合」の操作を行うことが出来ます。



１対１ 照合 サンプル

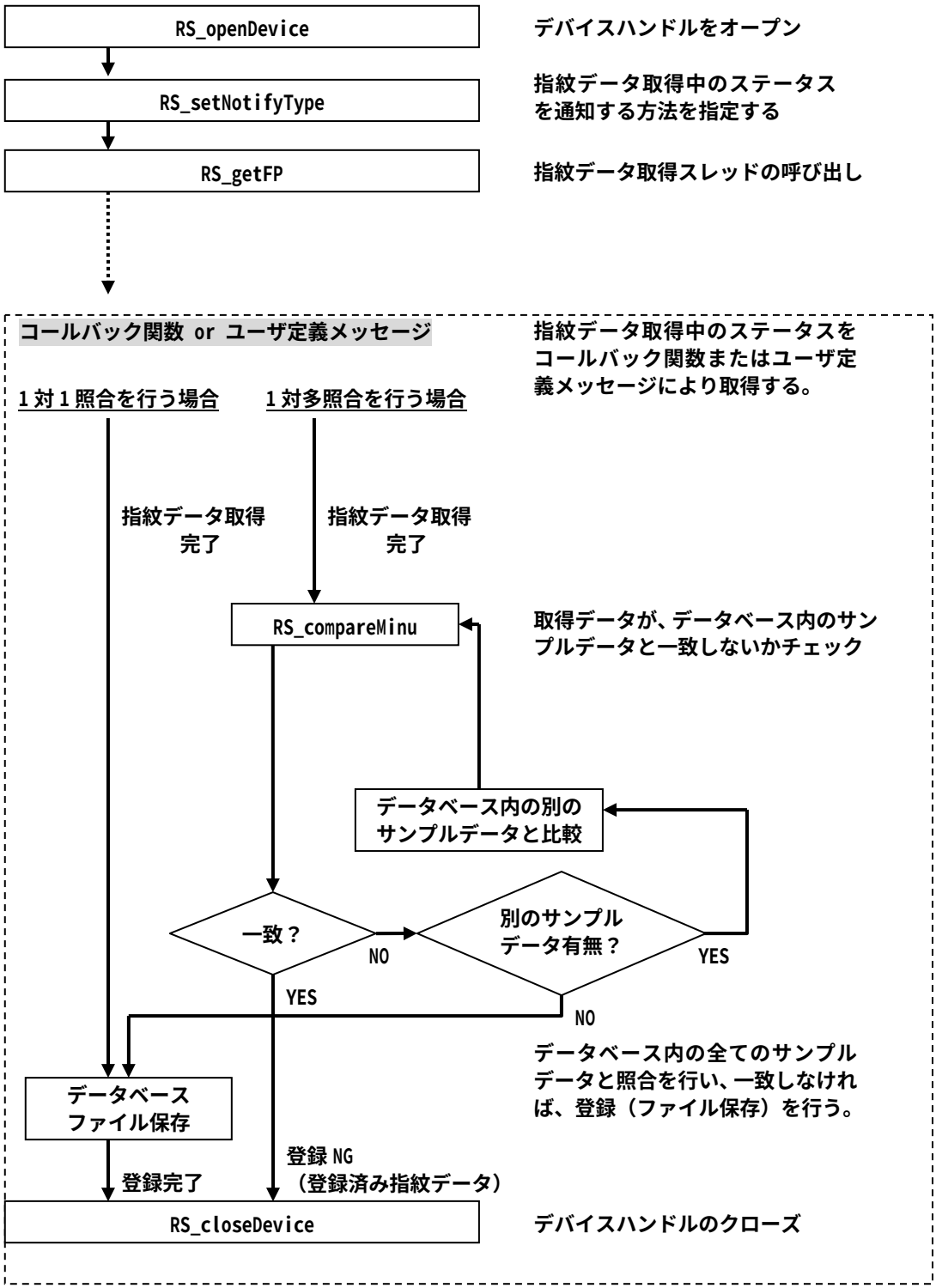


１対多 照合 サンプル

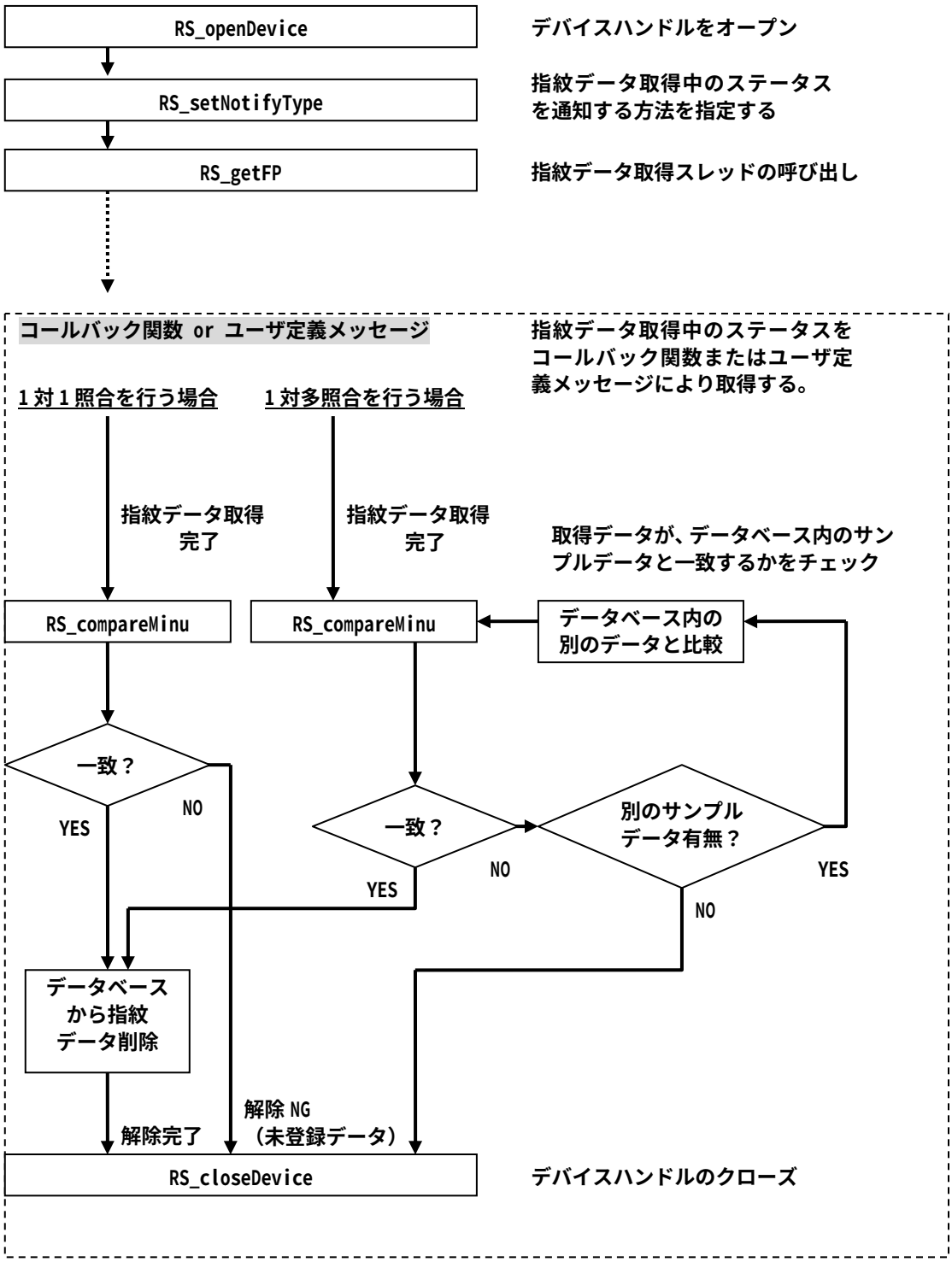
「指紋登録」・・・FSU1 指紋センサから取得した指紋データをデータベースファイルに保存します。
「指紋解除」・・・指定の指紋データをデータベースファイルから削除します。
「指紋照合」・・・FSU1 指紋センサから取得した指紋データとデータベースファイルから取得した指紋サンプルデータとの照合を行います。

次に、処理の流れを図に示します。

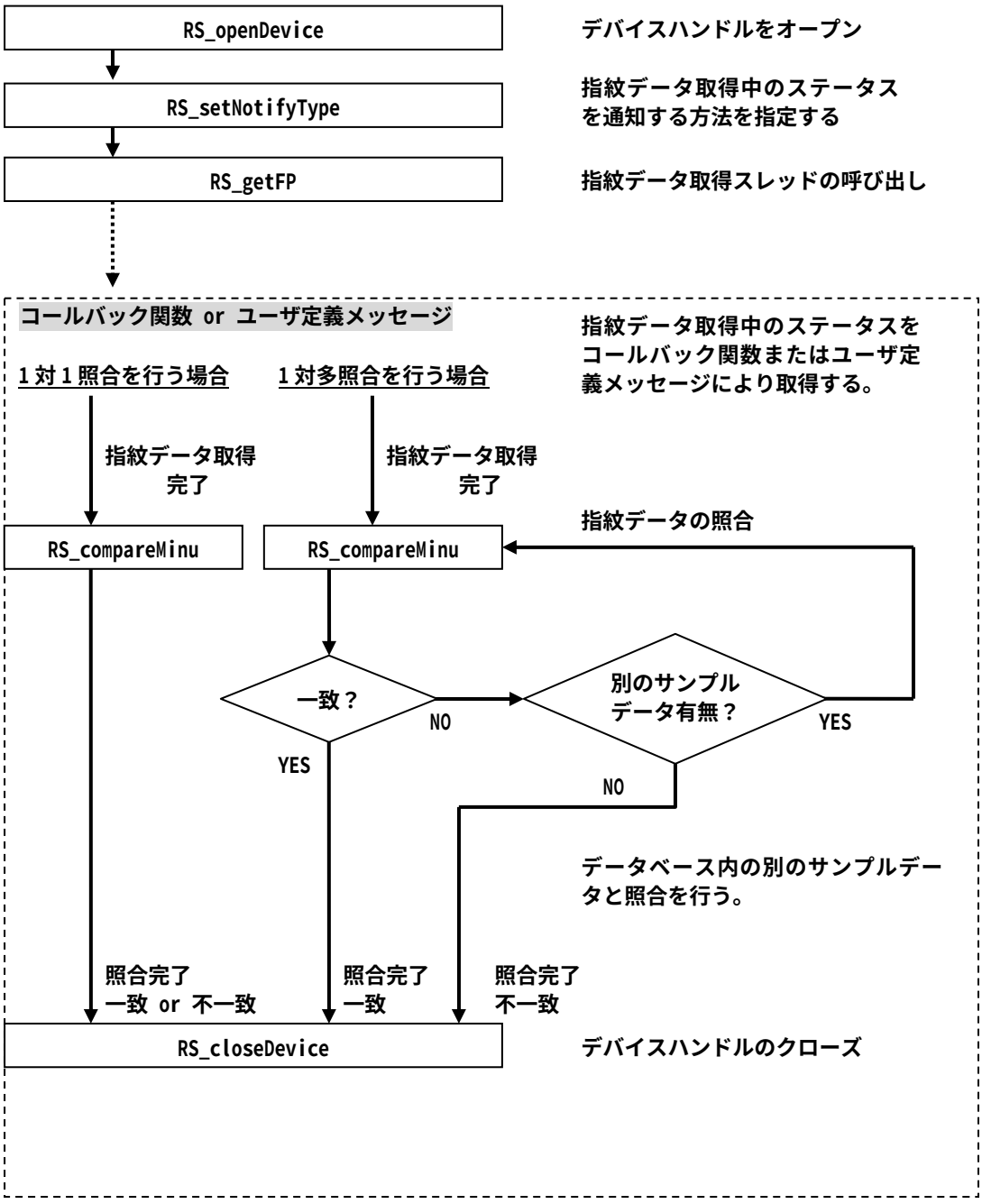
＜指紋登録＞


RS_openDevice
デバイスハンドルをオープン
RS_setNotifyType
指紋データ取得中のステータス
を通知する方法を指定する
RS_getFP
指紋データ取得スレッドの呼び出し
コールバック関数 or ユーザ定義メッセージ
指紋データ取得中のステータスを
コールバック関数またはユーザ定
義メッセージにより取得する。
1 対 1 照合を行う場合
1 対多照合を行う場合
指紋データ取得
完了
指紋データ取得
完了
RS_compareMinu
取得データが、データベース内のサン
プルデータと一致しないかチェック
データベース内の別の
サンプルデータと比較
一致？
NO
別のサンプル
データ有無？
YES
YES
NO
データベース内の全てのサンプル
データと照合を行い、一致しなけれ
ば、登録（ファイル保存）を行う。
データベース
ファイル保存
登録 NG
（登録済み指紋データ）
登録完了
RS_closeDevice
デバイスハンドルのクローズ

＜指紋解除＞


RS_openDevice
デバイスハンドルをオープン
RS_setNotifyType
指紋データ取得中のステータス
を通知する方法を指定する
RS_getFP
指紋データ取得スレッドの呼び出し
コールバック関数 or ユーザ定義メッセージ
指紋データ取得中のステータスを
コールバック関数またはユーザ定
義メッセージにより取得する。
1 対 1 照合を行う場合
1 対多照合を行う場合
指紋データ取得
完了
指紋データ取得
完了
取得データが、データベース内のサン
プルデータと一致するかをチェック
RS_compareMinu
RS_compareMinu
データベース内の
別のデータと比較
一致？
NO
YES
一致？
別のサンプル
データ有無？
NO
YES
YES
NO
データベース
から指紋
データ削除
解除 NG
（未登録データ）
解除完了
RS_closeDevice
デバイスハンドルのクローズ

＜指紋照合＞


RS_openDevice
デバイスハンドルをオープン
RS_setNotifyType
指紋データ取得中のステータス
を通知する方法を指定する
RS_getFP
指紋データ取得スレッドの呼び出し
コールバック関数 or ユーザ定義メッセージ
指紋データ取得中のステータスを
コールバック関数またはユーザ定
義メッセージにより取得する。
1 対 1 照合を行う場合
1 対多照合を行う場合
指紋データ取得
完了
指紋データ取得
完了
指紋データの照合
RS_compareMinu
RS_compareMinu
一致？
NO
別のサンプル
データ有無？
YES
YES
NO
データベース内の別のサンプルデー
タと照合を行う。
照合完了
一致 or 不一致
照合完了
一致
照合完了
不一致
RS_closeDevice
デバイスハンドルのクローズ

3-3．ライブラリ関数仕様

関数一覧

RS_openDevice	デバイスハンドルのオープン
RS_closeDevice	デバイスハンドルのクローズ
RS_setNotifyType	指紋データの取得時のステータス通知方法を設定
RS_getFP	指紋データの取得
RS_compareMinu	指紋データの照合を行う
RS_getResult	RS_getFP の結果を返す
RS_showFPDialog	指紋画像を表示するダイアログを表示
RS_hideFPDialog	指紋画像を表示するダイアログを非表示
RS_getVersion	各種 DLL ライブラリのバージョン情報を取得
RS_terminateFP	指紋取得処理を強制終了

書 式　VC ➢　long **RS_openDevice**(long *hDev)
VB ➢　Function **RS_openDevice**(ByRef hDev As Long) As Long
VB.NET ➢　Function **RS_openDevice**(ByRef hDev As Integer) As Integer

機 能　デバイスハンドルをオープンします。本 SDK を使用する前に、本関数を必ず呼び出す必要があります。

引 数　hDev　(OUT)　デバイスハンドルを受け取る変数ポインタ

戻 値　0：正常終了
-1：下位ライブラリ関数呼び出しエラー
-4：デバイスハンドル取得エラー
-5：FSU1 未検出

書 式　VC ➢　long **RS_closeDevice**(long hDev)
VB ➢　Function **RS_closeDevice**(ByVal hDev As Long) As Long
VB.NET ➢　Function **RS_closeDevice**(ByVal hDev As Integer) As Integer

機 能　デバイスハンドルをクローズします。RS_openDevice でオープンしたハンドルは使用後、必ずクローズするようにしてください。

引 数　hDev　(IN)　デバイスハンドル

戻 値　0：正常終了
-2：ハンドルクローズエラー

書 式

VC ➢
```
long APIENTRY RS_setNotifyType(long hDev, RS_NotifyType ntype,
                               void *pCallbackProc, void *pdata)
```

VB ➢
```
Function RS_setNotifyType(ByVal hDev As Long,
                          ByVal ntype As Long,
                          ByVal pCallbackProc As Long,
                          ByVal pdata As Long) As Long
```

VB.NET ➢
```
Function RS_setNotifyType(ByVal hDev As Integer,
                          ByVal ntype As Integer,
                          ByVal pCallbackProc As Integer,
                          ByVal pdata As Integer) As Integer
```

機 能　指紋データの取得時のステータス通知方法を設定します。
ただし、Visual Basic ではコールバック関数による通知方法は、使用しないでください。

引 数

hDev	(IN)	デバイスハンドル
ntype	(IN)	ステート通知方法（RS_NotifyType から選択）
pCallbackProc	(IN)	登録するコールバック関数へのポインタ
pdata	(IN)	コールバック関数に渡すユーザデータ。ユーザ定義メッセージ使用時は、ウィンドウハンドルを渡してください。

戻 値　0：通知方法設定の正常終了。
-1：デバイスハンドル指定エラー
-2：指紋データ取得中
-3：コールバック関数の登録エラー
-4：通知方法指定エラー
-5：ウィンドウハンドル指定エラー

```
//
// ステート通知方法
//
typedef enum{
    RS_Notify_None              = 0,   // 実行状態通知しない
    RS_Notify_Callback          = 1,   // コールバック関数により実行状態通知
    RS_Notify_UserMsg           = 2    // ユーザメッセージにより実行状態通知

} RS_NotifyType;
```

書 式　VC ➢　long **RS_getFP**(long hDev, FPPARAM_INFO *fpinfo)

VB ➢　Function **RS_getFP**(ByVal hDev As Long,
ByRef fpinfo As FPPARAM_INFO) As Long

VB.NET ➢　Function **RS_getFP**(ByVal hDev As Integer,
ByVal ptr As IntPtr) As Integer

機 能　FSU1 指紋センサから指紋データ（特徴点データ）の取得を行います。本関数は DLL 内部で指紋取得を行うスレッドを呼び出し、すぐに制御を返します。指紋データ取得中のステータスは、コールバック関数またはユーザ定義メッセージまたは RS_getResult にて取得可能です。

引 数

hDev	(IN)	デバイスハンドル
fpinfo	(OUT)	指紋データ用パラメータ構造体のポインタ

戻 値　0：指紋データ取得開始（ステート通知なし）
1：指紋データ取得開始（コールバック関数による通知）
2：指紋データ取得開始（ユーザ定義メッセージによる通知）
-1：デバイスハンドル指定エラー
-2：指紋データ取得中
-3：スレッド開始のエラー

```
//
// 指紋データ用パラメータ
//
typedef struct _FPPARAM_INFO{
    BYTE Minutiae[1024]; // 特徴点データ格納バッファ 1024Byte
    long lengthMinutiae; // 特徴点データサイズ
    long nMinutiae; // 特徴点データの特徴点の数
    long lMinuQuality; // 特徴点データの品質を数値化
    long lImageQuality; // 指紋イメージの品質を数値化
    float fscore; // 指紋画像のクオリティを小数値化
    BYTE times; // 指紋データ取得の試行回数（指紋登録時は、3 回以上を推奨します。）
    BYTE timeouts; // 指の検出試行を終了するまでのタイムアウト時間
    BYTE chkparam; // 0 を指定してください

} FPPARAM_INFO, *PFPPARAM_INFO;
```

書 式　VC ➢

```
long APIENTRY RS_compareMinu(long hDev,
                             BYTE *Minu1, long nMinu1,
                             BYTE *Minu2, long nMinu2,
                             COMPARE_PARAM *cpparam)
```

VB ➢

```
Function RS_compareMinu(ByVal hDev As Long,
                        ByRef Minu1 As Byte, ByVal nMinu1 As Long,
                        ByRef Minu2 As Byte, ByVal nMinu2 As Long,
                        ByRef cpparam As COMPARE_PARAM) As Long
```

VB.NET ➢

```
Function RS_compareMinu(ByVal hDev As Integer,
                        ByRef Minu1 As Byte, ByVal nMinu1 As Integer,
                        ByRef Minu2 As Byte, ByVal nMinu2 As Integer,
                        ByRef cpparam As COMPARE_PARAM) As Integer
```

機 能　２つの指紋データの照合を行い、照合一致・不一致の結果を返します。
照合が一致した場合、戻り値に特徴点の一致した数を返します。
1 対多照合を行う場合には、あらかじめ登録時に、登録しようとするデータと一致するサンプルデータがデータベースに存在しないことを本関数により、チェックするべきです。(他人受理率の低減)
1 対多照合を行う場合には、Minu1 にデータベース上のサンプルデータ、Minu2 に照合対象のデータを指定してください。2 回目以降の照合については、nMinu2 を 0 にすることで、スピードのパフォーマンスが得られます。

引 数

hDev	(IN)	デバイスハンドル
Minu1	(IN)	特徴点データ１を格納するバッファポインタ
nMinu1	(IN)	特徴点データ１の特徴点数
Minu2	(IN)	特徴点データ２を格納するバッファポインタ
nMinu2	(IN)	特徴点データ２の特徴点数
cpparam	(IN)	照合用パラメータをセットした構造体

戻 値　1 以上の整数：正常終了。指紋データが一致。
0：正常終了。指紋データが不一致。
-1：デバイスハンドル指定エラー
-2：下位ライブラリ関数呼び出しエラー

```
//
// 照合用パラメータ
//
typedef struct _COMPARE_PARAM{
    BYTE secuLevel; // セキュリティレベル 0～4 の値を設定してください。
                    // 0：セキュリティ高　4：セキュリティ低
    long judge; // 内部判定（0 を指定してください）
    long matchscore; // 一致度数ポイントを受け取る変数
} COMPARE_PARAM, *PCOMPARE_PARAM;
```

書 式　VC ➢　long **RS_getResult**(long *result)
VB ➢　Function **RS_getResult**(ByRef result As Long) As Long
VB.NET ➢　Function **RS_getResult**(ByRef result As Integer) As Integer

機 能　RS_getFP の結果（指紋データ取得中のステータス）を返します。RS_FPState の値のいずれかを返します。
指紋データ取得中のステータスは、ほかにもコールバック関数またはユーザ定義メッセージにより取得可能です。

引 数　result　(OUT)　RS_FPState の値を受け取る変数ポインタ

戻 値　常に 0 を返します。

```
//
// 指紋データ取得時 ステータス
//
typedef enum{
    RS_SUCCESS_FPGET, // 指紋データ取得に成功した
    RS_FAILED_FPGET, // 指紋データ取得に失敗した
    RS_FAILED_FP_TIMEOUT, // タイムアウト

    RS_PENDING_STARTED_FP_GETTING, // 指紋データ取得開始
    RS_PENDING_END_FP_GETTING, // 指紋データ取得終了

    RS_PENDING_NO_FINGER, // 指を検出していない
    RS_PENDING_FINGER_DETECT, // 指を検出した

    RS_PENDING_TOO_HIGH, // 指が上方にある
    RS_PENDING_TOO_LOW, // 指が下方にある
    RS_PENDING_TOO_LEFT, // 指が左側にある
    RS_PENDING_TOO_RIGHT, // 指が右側にある
    RS_PENDING_BAD_QUALITY, // 品質がよくない
    RS_PENDING_FP_DIFFERENT, // 指紋データが異なる

} RS_FPState;
```

書 式　VC ➢　long **RS_showFPDialog**(FPDIALOG_INFO *dialoginfo)

VB ➢　Function **RS_showFPDialog**(
ByRef dialoginfo As FPDIALOG_INFO) As Long

VB.NET ➢　Function **RS_showFPDialog**(
ByRef dialoginfo As FPDIALOG_INFO) As Integer

機 能　指紋画像を表示するダイアログを表示します。

引 数　dialoginfo　(IN)　ダイアログ設定情報をもつ構造体のポインタ

戻 値　0：正常終了。
-1：位置設定エラー。

書 式　VC ➢　long **RS_hideFPDialog**(FPDIALOG_INFO *dialoginfo)

VB ➢　Function **RS_hideFPDialog**(
ByRef dialoginfo As FPDIALOG_INFO) As Long

VB.NET ➢　Function **RS_hideFPDialog**(
ByRef dialoginfo As FPDIALOG_INFO) As Integer

機 能　指紋画像を表示するダイアログを非表示にします。

引 数　dialoginfo　(IN)　ダイアログ設定情報をもつ構造体のポインタ

戻 値　常に 0 を返します。

```
//
// 指紋画像表示用ダイアログパラメータ
//
typedef struct _FPDIALOG_INFO{
    long left; // ダイアログ表示位置の左上隅の x 座標。絶対座標の pixel 単位で指定。
    long top; // ダイアログ表示位置の左上隅の y 座標。絶対座標の pixel 単位で指定。
    long right; // ダイアログ表示位置の右下隅の x 座標。絶対座標の pixel 単位で指定。
    long bottom; // ダイアログ表示位置の右下隅の y 座標。絶対座標の pixel 単位で指定。
    COLORREF dlgrgb; // ダイアログ色（RGB 値）（0xFFFFFFFF を指定で既定値）
    COLORREF bitmapbkrgb; // ビットマップ背景色（RGB 値）（0xFFFFFFFF を指定で既定値）
    long penwidth; // 内部判定描画用ペンの太さ（ピクセル指定、0 指定で既定値 20pixel）
    char *pDlgTitle; // ダイアログタイトルの名前が格納されたバッファポインタ（NULL 終端）

} FPDIALOG_INFO, *PFPDIALOG_INFO;
```

書 式	VC ➢	long **RS_getVersion**(DLLVER_INFO *verInfo)
	VB ➢	Function **RS_getVersion**(ByRef verInfo As DLLVER_INFO) As Long
	VB.NET ➢	Function **RS_getVersion**(ByRef verInfo As DLLVER_INFO) As Integer

機 能　各種 DLL ライブラリのバージョン情報を取得します。

引 数

verInfo	(OUT)	バージョン情報（文字列）を格納するバッファポインタ

戻 値　0：正常終了
-1：リソース検出エラー
-2：バージョン取得エラー

```
//
// DLL ライブラリバージョン情報（文字列）を受け取るバッファポインタ構造体
//
typedef struct _DLLVER_INFO{
    char *RS_Lib;
    char *FujiCap_Lib;
    char *fpfltr5_Lib;
    char *minu_Lib;
    char *kyQual_Lib;
    char *vmatch_Lib;
} DLLVER_INFO, *PDLLVER_INFO;
```

書 式　VC ➢　long **RS_terminateFP**(long hDev)

VB ➢　Function **RS_terminateFP**(ByVal hDev As Long) As Long

VB.NET ➢　Function **RS_terminateFP**(ByVal hDev As Integer) As Integer

機 能　指紋データ取得中の処理を強制的に終わらせます。本関数呼出し後は、指紋取得ステータス RS_FPState を返しません。
また、本関数呼出し後の指紋データは保障されません。
指紋データ取得中でない場合に呼び出した場合には、特に何も起こりません。

引 数

hDev	(IN)	デバイスハンドル

戻 値　0：正常終了
-1：デバイスハンドル指定エラー

書 式　VC ➢

```
long CALLBACK FuncCallback (RS_FPState fpstatus,
                            FPPARAM_INFO *fpinfo,
                            void *udata)
```

VB ➢

機 能　指紋データ取得中のステータスを受け取るコールバック関数です。RS_FPState の値のいずれかを返します。
本関数はアプリケーションにて実装します。
ただし、Visual Basic では使用できません。

引 数

fpstatus	(IN)	指紋データ取得中の ステータス
fpinfo	(IN)	指紋データ情報をもつ構造体のポインタ
udata	(IN)	RS_setNotifyType()にて渡すユーザデータ

戻 値　0 を返します。

（空白ページ）

FAX：06-6633-8285

FSU1 指紋認証開発キット 質問用紙

●下記ユーザ情報をご記入願います。

法人登録の方のみ	会社名・学校名			
	所属部署			
ご担当者名				
E-Mail				
住所	〒			
TEL		FAX		
製品型番		シリアル№		
ご購入情報	販売店名		購入日	

●下記運用環境情報とお問い合わせ内容をご記入願います。

【パソコン/マザーボードのメーカ名と機種名】
【ご利用の OS】
【ライブラリバージョン】
【お問合せ内容】
【添付資料】

⚠ 個人情報取り扱いについて

ご連絡いただいた氏名、住所、電話番号、メールアドレス、その他の個人情報は、お客様への回答など本件に関わる業務のみに利用し、他の目的では利用致しません。