Smack Bio1000 Pro OCX

User's Manual

CONTENTS

1. Introduction

This document is a user's manual of the high-speed OEM module Smack Bio1000 Pro(version 2.1) based on the superior fingerprint identification technology. Smack Bio1000 Pro is supplied in the form of OCX and using it users can develop various high performance fingerprint identification systems.
The functions of Smack Bio1000 Pro are as follows.

- Optical sensor control through USB 2.0
- Enrollment, 1:1verification and 1: N identification of fingerprints.
- Enrollment, 1:1verification and 1: N identification of fingerprint files.
- Database management
- Convenient error code.

Technical Specifications

Item	Value
OS	Windows XP / Windows2000
USB	USB 2.0
Image Dimension	256pixel(W) × 256pixel(H)
Capacity of fingerprint enrollment	2000 fingers
Fingerprint data enrollment number	1 ~ 32767
Match Mode	1:1, 1:N match
Template size	1496 Bytes
Threshold control level	LOW, MEDIUM, HIGH
False Acceptance Rate (FAR)	< 0.0001 %
False Rejection Rate (FRR)	< 0.01 %
Enroll time	< 1 sec
Identification time	< 0.5 sec

2. Functions of Smack Bio1000 Pro OCX

2.1 Properties

2.1.1FPBrightness

- type : LONG
- range : 0 ~ 255
- default : 170
- access mode : read / write
- description : Return or set the brightness of a sensor.

2.1.2 FPContrast

- type : LONG
- range : 0 ~ 255
- default : 250
- access mode : read / write
- description : Return or set the contrast of a sensor.

2.1.3 FPDatabaseDir

- type : String
- range :
- default : Application Path\DB
- access mode : read / write
- description : Return or set the name of fingerprint database directory.

2.1.4 FPVerifyID

- type : Long
- range : 32767≥ x > 0
- default : 0
- access mode : read only (Unable to set in design time.)
- description : Return the ID resulting from Enroll, Verify and Identify. An error occurred, it is set as 0.

2.1.5 FPVerifyFingerNumber

- type : Long
- range : 10≥ x > 0
- default : 0
- access mode : read only (Unable to set in design time.)

- description : Return the finger number resulting from Enroll, Verify and Identify. An error occurred, it is set as 0.

2.1.6 FPVerifyLevel

- type : Long
- range : VERIFY_LEVEL_HIGH> V > VERIFY_LEVEL_LOW
- default : VERIFY_LEVEL_HIGH
- access mode : read / write
- description : Return or set a verification threshold level.
  VERIFY_LEVEL_HIGH : verification of high level.
  VERIFY_LEVEL_MEDIUM : verification of medium level.
  VERIFY_LEVEL_LOW : verification of low level.

2.1.7 FPEnrollCount

- type : Long
- range : 4bytes
- default :
- access mode : read only (Unable to set in design time.)
- description : Return the number of data enrolled in database.

2.2 Methods

2.2.1 InitKTFP

- function : Initialize a sensor and OCX.
- declaration : **InitKTFP** ( ) as Long
  long **InitKTFP** ()
- parameter :
- return value :
  - ERROR_INIT_DEVICE : Unable to initialize a device
  - KTFP_SUCCESS : Device initialization success

2.2.2 DeinitKTFP

- function : It disconnects with the sensor and initializes OCX.
- declaration : **DeinitKTFP** ( ) as Long
  long **DeinitKTFP**()
- parameter :
- return value :
  - ERROR_DEINIT_DEVICE : failure
  - KTFP_SUCCESS : success

2.2.3 EnrollStart

- function : Start enroll process when 3 fingerprints enroll.
  User should call this function to enroll 3 fingerprints captured from sensor or 3 fingerprint files and specify ID and finger number to enroll.
  Then he can control 3 fingerprints enroll process by using EnrollProN, EnrollEnd Method.
  (Caution: Before calling this function, using CheckFingerNum or CheckID Method it should be checked whether ID and finger number to enroll are used in database or not.)
- declaration : **EnrollStart(ID** as Long, **FingerNumber** as Long, **Manager** as Boolean) as Long
  long **EnrollStart** (long **ID**, long **FingerNumber**, BOOL **Manager**)
- parameter :
  - ID : ID to enroll (1 ~ 32767)
  - FingerNumber : Finger number to enroll (1 ~ 10)
  - Manager: It describes that the specified ID, finger number is a manager.
    - TRUE: Is a manager.
    - FALSE: Is not a manager.
- return value :
  - ERROR_INVALID_ID: It is an invalid ID.
  - ERROR_INVALID_FINGER_NUM: It is an invalid finger number.
  - ERROR_OVER_DB: In database, there is no place to enroll.
  - KTFP_SUCCESS : success
- example

```
……….
If SBFP1000.CheckFingerNum(nID, nFN) = ERROR_ENROLLED_FINGER Then
   If MsgBox("This ID and fingerNumber has been already enrolled. Continue?",
vbOKCancel, "VB Sample") = vbCancel Then
      lblMessage = "Cancel."
      GoTo err
   End If
   SBFP1000.DeleteData nID, nFN
End If

ret = SBFP1000.EnrollStart(nID, nFN, False)
If ret = SBFP_SUCCESS Then
   'Enrollment proceeding
Else
   'Error treating
End If
……
```

2.2.4 EnrollProN

- function : Proceed enrollment of Nth (1~3) fingerprint when 3 fingerprints enroll.
  This method enrolls currently specified fingerprint image (specified by CaptureImage, ViewFP or SetImageFile) after calling EnrollStart methord .
  (Caution: Make CaptureImage, ViewFP or SetImageFile not to be called again by a careless after capturing fingerprint image.
  If these methods are called, then aimed fingerprint image data is changed.)
  Setting parameter step from 1 to 3 in turn, get 3 fingerprint data needed to enroll.
- declaration : **EnrollProN** (**step** as Long) as Long
  long **EnrollProN** (long **step**)
- parameter :
  - step : number of fingerprint to enroll (1 ~ 3).
- return value :
  - ENROLL_STEP_EXE_ERROR: error

- ENROLL_STEP_EXE_OK: success

2.2.5 EnrollEnd

- function : Enroll the fingerprint data by compounding 3 fingerprint data obtained by EnrollProN methord when 3 fingerprints enroll.
  (Caution: Make CaptureImage, ViewFP or SetImageFile not to be called again by a careless after calling EnrollProN.
  If these functions are called, then aimed fingerprint image data is changed.)
- declaration : **EnrollEnd** ( ) as Long
  long **EnrollProN** ( )
- parameter :.
- return value :
  - ENROLL_DUPLICATE_ERROR: There already exists a fingerprint data to enroll in database.
  - ERROR_ENROLL_FAILED: failure
  - KTFP_SUCCESS: success
- example

```
……
For i = 1 To 3
    lblMessage = "Press Finger(" + Str(i) + ")!"
    If Capture(0) = False Then GoTo err
    SBFP1000.CaptureImage True
    If SBFP1000.EnrollProN(i) <> SBFP_SUCCESS Then GoTo err
    If (i = 3) Then
      If SBFP1000.EnrollEnd <> SBFP_SUCCESS Then GoTo err
      GoTo suc
    End If
    lblMessage = "Takeoff Finger!"
    If WaitTakeoff() = False Then GoTo err
  Next
……
```

2.2.6 Register

- function : It is enrollment using 1 fingerprint or fingerprint image and enrolls currently specified fingerprint or fingerprint image

(specified by CaptureImage, ViewFP or SetImageFile).
If enrollment succeeds, enrolled ID, finger number is reflected to FPVerifyID and VerifyFingerNumber properties.

- declaration : **Register** (**ID** as Long, **FingerNumber** as Long, **Manager** as Boolean) as Long
  long **Register** (long **ID**, long **FingerNumber**, BOOL **Manager**)
- parameter :
  - ID : ID to enroll (1 ~ 32767)
  - FingerNumber: finger number to enroll (1 ~ 10)
  - Manager: It describes that the specified ID, finger number is a manager.
    - TRUE: Is a manager.
    - FALSE: Is not a manager.
- return value :
  - ERROR_INVALID_ID: It is an invalid ID.
  - ERROR_INVALID_FINGER_NUM: It is an invalid finger number.
  - ERROR_OVER_DB: In database there is no place to enroll.
  - ENROLL_DUPLICATE_ERROR: The fingerprint data has already been enrolled in database.
  - ERROR_ENROLL_FAILED: failure
  - KTFP_SUCCESS : success

2.2.7 Verify

- function : Verify (1:1 Verify) the input fingerprint image with the specified ID, finger number.
  A fingerprint image to verify is set by CaptureImage, ViewFP or SetImageFile Method.
  If verification succeeds, verified ID, finger number is reflected to FPVerifyID and VerifyFingerNumber properties.
  Then verified fingerprint image is saved in the name of Recimage\result.bmp to the database directory. (FPDatabaseDir property).
- declaration : **Verify** (**ID** as Long, **FingerNum** as Long) as Long
  long **Verify**(long **ID**, long **FingerNum**)
- parameter :
  - ID : ID to verify (1 ~ 32767)
  - FingerNum : finger number to verify (0 ~ 10)
    - 0 : Verify with any finger number of specified ID

▪ Others : Verify with specified finger number of specified ID

- return value :
  - ERROR_INVALID_ID: It is an invalid ID.
  - ERROR_INVALID_FINGER_NUM: It is an invalid finger number.
  - ERROR_VERIFY_FAILED: failure
  - KTFP_SUCCESS : success

2.2.8 Identify

- function : Identify(1:N Identify) the input fingerprint image.
  A fingerprint image to identify is set by CaptureImage, ViewFP or SetImageFile Method.
  If identification succeeds, identified ID, finger number is reflected to FPVerifyID and VerifyFingerNumber properties.
  And identified fingerprint image is saved in the name of Recimage\result.bmp to the database directory.
  (FPDatabaseDir property).
- declaration : **Identify** ( ) as Long
  long **Identify**()
- parameter :
- return value :
  - ERROR_IDENTIFY_FAILED: failure
  - KTFP_SUCCESS : success

2.2.9 ViewFP

- function : After capturing a fingerprint image from a sensor, set and display it in OCX.
  Already set fingerprint image is used for enrollment and identification.
- declaration : **ViewFP**() as Long
  long **ViewFP**()
- parameter :
- return value :
  - ERROR_INIT_DEVICE: Fingerprint sensor is not initialized.
  - ERROR_INSUFFICIENT_RESOURCE: Memory is insufficient.
  - ERROR_GET_IMAGE_DEVICE: It is unable to get image from a sensor.
  - KTFP_SUCCESS: success

2.2.10 CaptureImage

- function : After capturing a fingerprint image from a sensor, set it in OCX.
  Already set fingerprint image can be displayed or not at request and is used for enrollment and identification.

- declaration : **CaptureImage** (**View** as Boolean) as Long
  long **ViewFP** (BOOL **View**)
- parameter :
  - View: Set whether captured fingerprint image should be displayed or not.
    - True: Display a fingerprint image.
    - False: Do not display a fingerprint image.
- return value :
  - ERROR_INIT_DEVICE: Fingerprint sensor is not initialized.
  - ERROR_GET_IMAGE_DEVICE: Unable to get image from the sensor.
  - KTFP_SUCCESS: success
- Example

```
Function Capture(TimeOut As Long) As Boolean
    st = GetTickCount
    ret = SBFP_SUCCESS
      While (1)
        'Verify Finger with sensor
        ret = SBFP1000.CaptureImage(False)
        If TimeOut <> 0 And (GetTickCount() - st) > (TimeOut * 1000) Then GoTo err
        DoEvents
        If SBFP1000.IsFingerprint = True Then
          Capture = True
          Exit Function
        End If
      Wend
    err:
      Capture = False
End Function
```

2.2.11 SetImageFile

- function : Set and display a required fingerprint file in OCX. Fingerprint file is Gray BMP file with size of 256×256. It is used for enrollment and identification of fingerprint files.
- declaration : **SetImageFile** (**FileName** as String) as Long
  long **ViewFP** (LPCTSTR **FileName**)
- parameter :
  - FileName: The file name of fingerprint file to set.
- return value :
  - ERROR_CREATE_FILE: Unable to open specified file.
  - ERROR_INVALID_FILENAME: Is not correct file format.

- KTFP_SUCCESS: success

2.2.12 IsFingerprint

- Function : Decide whether current image set by CaptureImage, ViewFP, SetImageFile Method in OCX is a fingerprint image or not.
- declaration : **IsFingerprint** ( ) as Boolean
  BOOL **IsFingerprint** ( )
- parameter :
- return value :
  - False: Is not a fingerprint image.
  - True: Is a fingerprint image.
- Example

```
Function WaitTakeoff() As Boolean
    While (1)
        ret = SBFP1000.CaptureImage(False)
        If SBFP1000.IsFingerprint = False Then
           WaitTakeoff = True
           Exit Function
        End If
    Wend
    err:
    WaitTakeoff = False
End Function
```

2.2.13 SaveImageToFile

- function : Save the input fingerprint image to a file.
- declaration : **SaveImageToFile**(**FileName** as String) as Long
  long **SaveImageToFile** (LPCTSTR **FileName**)
- parameter : FileName : The name of a file to which fingerprint image is saved.
- return value :
  - ERROR_CREATE_FILE : File opening failure
  - ERROR_INIT_DEVICE: Device is not initialized.
  - ERROR_INSUFFICIENT_RESOURCE : Insufficient memory
  - ERROR_GET_IMAGE_DEVICE: Unable to get fingerprint image from device.
  - KTFP_SUCCESS : success

2.2.14 Cls

- function : Clear the interface of control.
- declaration : **Cls**() as Long
  long **Cls**()
- parameter :
- return value :
  - ERROR_INSUFFICIENT_RESOURCE: Unable to do redrawing of control.
  - KTFP_SUCCESS : success

2.2.15 Cancel

- function : Cancel enrollment, verification or identification in progress.
- declaration : **Cancel** ()
  void **Cancel**()
- parameter :
- return value :

2.2.16 ClearDB

- function : Clear fingerprint database.
- type : **ClearDB** () as Long
  long **ClearDB**()
- parameter :
- return value :
  - ERROR_CLEAR_DB : failure
  - KTFP_SUCCESS : success

2.2.17 CheckID

- function : Check whether ID inputted by user is used in database or not.
- declaration : **CheckID**(long ID) as Long
  long **CheckID**(long ID)
- parameter : ID: ID to check (1 ~ 32767 .)
- return value :
  - ERROR_INVALID_ID: It is an invalid ID.
  - ERROR_ENROLLED_ID: It is already enrolled ID.
  - IS_VALID_ID: It is ID enables to enroll.

2.2.18 CheckFingerNum

- function : Check whether fingerprint data is enrolled in specified finger number of specified ID or not.

- declaration : **CheckFingerNum** (**ID** as Long, **FingerNumber** as Long) as Long
  long **CheckFingerNum**(long **ID**, long **FingerNumber**)
- parameter :
  - ID : ID to check (1 ~ 32767)
  - FingerNumber : finger number to check (1 ~ 10)
- return value :
  - ERROR_INVALID_ID: It is an invalid ID.
  - ERROR_INVALID_FINGER_NUM: It is an invalid finger number.
  - ERROR_ENROLLED_FINGER: It is already enrolled finger number.
  - IS_VALID_FINGER: It is finger number enables to enroll.

2.2.19 SearchID

- function : Search the first empty (unenrolled) ID.
- declaration : **SearchID**() as Long
  long **SearchID** ()
- parameter :
- return value : The first empty ID searched or error code
  - ERROR_EMPTY_ID: There is no empty ID.
  - Others : The first empty ID searched

2.2.20 SearchFingerNumber

- function : Search the first empty (unenrolled) finger number of specified ID.
- declaration : **SearchFingerNumber**(**ID** as Long) as Long
  long **SearchFingerNumber**(long **ID**)
- parameter :
  - ID : ID of which the first empty finger number is searched
- return value : The first empty finger number searched or error code
  - ERROR_INVALID_ID: It is an invalid ID.
  - IS_VALID_ID: Any finger number of specified ID is valid.
  - ERROR_EMPTY_FINGER: There is no empty finger number of specified ID.
  - Others : The first empty finger number searched

2.2.21 DeleteData

- function : Delete fingerprint data with specified ID, FingerNum from enrollment database.
- declaration : **DeleteData**(Long **ID**, Long **FingerNum**) as Long

long **DeleteData**(long **ID**, long **FingerNum**)

- parameter :
  - ID : ID of fingerprint data to delete
  - FingerNum : Finger number of fingerprint data to delete (0 ~ 10)
    - 0: Delete all fingerprint data with specified ID.
    - Others: Delete only fingerprint data with specified ID and specified FingerNum
- return value :
  - ERROR_INVALID_ID: It is an invalid ID.
  - ERROR_INVALID_FINGER_NUM: It is an invalid finger number.
  - ERROR_DELETE_DATA : failure
  - KTFP_SUCCESS : success

2.2.22 GetFPData

- function : Get fingerprint enroll data with specified ID, FingerNum.
  If FingerNum is from 1 to 10, then fingerprint data of specified FingerNum is returned.
  At that time user should allocate 1 fingerprint data size of buffer.
  Corresponding enroll image is saved in the name of Recimage\result.bmp to the database directory (FPDatabaseDir property).
  If the specified finger number is not enrolled, then buffer is full of 0.
  If FingerNum is 0, then all data of 10 finger numbers of specified ID are returned
  At that time user should allocate 10 fingerprint data size of buffer.
  As a result, enrolled fingerprint data is saved only in the corresponding finger number place of buffer.
  Corresponding enroll images are not copied.
- declaration : **GetFPData**(**ID** as Long, **FingerNum** as Long, **Buffer** as Long, **Size** as Long) as Long
  long **GetFPData**(long **ID**, long **FingerNum**, long* **Buffer**, long **Size**)
- parameter :
  - ID: ID of fingerprint enroll data to get. (1 ~ 32767 .)
  - FingerNum : finger number of fingerprint enroll data to get (0 ~ 10)
    - 0: Get all fingerprint enroll data with specified ID.

- Others: Get only specified fingerprint enroll data with specified ID.
- Buffer : address of buffer in which enroll data is saved
- Size : size of buffer in which enroll data is saved

- return value :
  - ERROR_INVALID_ID: It is an invalid ID.
  - ERROR_INVALID_FINGER_NUM: It is an invalid finger number.
  - ERROR_INVALID_BUFFER: It is an invalid buffer.
  - ERROR_INVALID_BUFFER_SIZE: The size of buffer is insufficient.
  - ERROR_GET_DATA: failure.
  - KTFP_SUCCESS : success

2.2.23 CheckManager

- function : Check whether specified ID, FingerNumber is a manager or not.
- declaration : **CheckManager**(**ID** as Long, **FingerNumber** Long) as Long
  long **CheckManager**(long **ID**, long **FingerNumber**)
- parameter :
  - ID: ID of data to check. (1 ~ 32767)
  - FingerNumber: finger number of data to check (1 ~ 10)
- return value :
  - ERROR_INVALID_ID: It is an invalid ID.
  - ERROR_INVALID_FINGER_NUM: It is an invalid finger number.
  - ERROR_NOT_MANAGER: Is not a manager.
  - IS_MANAGER: Is a manager.

2.2.24 SetFPData

- function : Set fingerprint enroll data with specified ID, FingerNumber.
- declaration : **SetFPData**(**FPData** as Long, **Size** as Long, **ID** as Long, **FingerNumber** as Long) as Long
  long **SetFPData**(long* **FPData**, long **Size**, long **ID**, long **FingerNumber**)
- parameter :
  - FPData : Address of enroll data
  - Size : Size of enroll data
  - ID : ID of data to set. (1 ~ 32767)
  - FingerNumber: Finger number of data to set (1 ~ 10)
- return value :

- ERROR_INVALID_ID: It is an invalid ID.
- ERROR_INVALID_FINGER_NUM: It is an invalid finger number.
- ERROR_INVALID_BUFFER: The address of data is invalid.
- ERROR_INVALID_BUFFER_SIZE: The size of buffer is insufficient.
- ERROR_SET_DATA : failure
- KTFP_SUCCESS : success

2.2.25 Match2FPData

- function : Match 2 inputted fingerprint enroll data.
- declaration : **Match2FPData**(**FPData1** as Long, **DataLen1** as Long, **FPData2** as Long, **DataLen2** as Long)
  long **Match2FPData**(long FAR* **FPData1**, long **DataLen1**, long FAR* **FPData2**, long **DataLen2**)
- parameter :
  - FPData1 : The first fingerprint enroll data to match
  - FPData1 : The second fingerprint enroll data to match
  - DataLen1 : Size of the first enroll data
  - DataLen2 : Size of the second enroll data
- return value :
  - ERROR_INVALID_BUFFER: Buffer of enroll data is invalid.
  - ERROR_INVALID_BUFFER_SIZE: The sizes of enroll data are different.
  - ERROR_DIFF_FINGER: Two fingerprint data are different.
  - IS_SAME_FINGER: Two fingerprint data are same.

2.3 Events

2.3.1 FPError

- function : Report error code and description of an error occurred in Smack Bio1000.
- declaration : **FPError**(**ErrCode** as Long, **Desc** as String)
  void **FPError**(long **ErrCode**, BSTR **Desc**)
- parameter :
  - ErrCode: Error code
  - Desc : Description of error
- example

```
Private Sub SBFP1000_FPError ( ByVal ErrCode As Long, ByVal Desc As String)
        lblMessage.Caption = Desc
End Sub
```

3. Error code and constants table

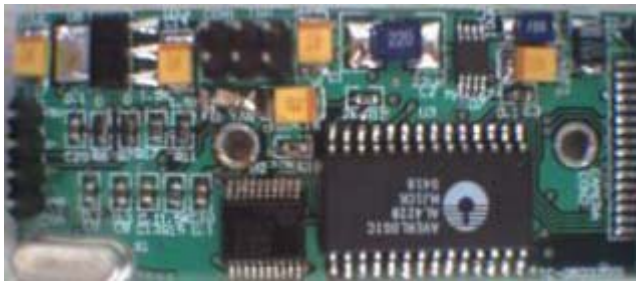
No	Name	Value	Description
1	VERIFY_LEVEL_LOW	0	Fingerprint verification of low level
2	VERIFY_LEVEL_MEDIUM	1	Fingerprint verification of medium level
3	VERIFY_LEVEL_HIGH	2	Fingerprint verification of high level
4	ENROLL_STEP_EXE_OK	0	Former stage of enrollment success
5	ENROLL_STEP_EXE_ERROR	-1	Former stage of enrollment failure
6	ENROLL_DUPLICATE_ERROR	-2	Fingerprint data has already enrolled.
7	SBFP_SUCCESS	0	Process result success
8	ERROR_INSUFFICIENT_RESOURCE	-2	Insufficient system resource
9	ERROR_GET_IMAGE_DEVICE	-3	Unable to get image from a sensor.
10	ERROR_INIT_DEVICE	-4	Unable to initialize a sensor.
11	ERROR_DEINIT_DEVICE	-5	Unable to exit a device.
12	ERROR_BRIGHTNESS_RANGE	-6	Sensor brightness is over the range.
13	ERROR_SET_BRIGHTNESS	-7	Sensor brightness setting error
14	ERROR_CONTRAST_RANGE	-8	Sensor contrast is over the range.
15	ERROR_SET_CONTRAST	-9	Sensor contrast setting error
16	ERROR_ENROLL_FAILED	-11	Enrollment failure.
17	ERROR_ENROLLED_FINGER	-12	Fingerprint data to enroll is already enrolled one.
18	ERROR_VERIFY_FAILED	-13	1:1 verification failure
19	ERROR_IDENTIFY_FAILED	-14	1:N identification failure
20	ERROR_CREATE_FILE	-15	Unable to create temporary file.
21	ERROR_USER_CANCEL	-16	Be cancelled by user.
22	ERROR_CLEAR_DB	-18	Unable to clear database.
23	ERROR_ENROLLED_ID	-19	Is already enrolled ID.
24	ERROR_INVALID_ID	-20	ID value range is invalid.
25	ERROR_INVALID_FINGER_NUM	-21	Finger number is invalid.
26	ERROR_DELETE_DATA	-22	Unable to delete data.
27	ERROR_INVALID_BUFFER	-23	Buffer address is invalid.
28	ERROR_INVALID_BUFFER_SIZE	-24	Buffer size is invalid.
29	ERROR_INVALID_VERIFY_LEVEL	-25	Verification level is over the range.
30	ERROR_ENROLLED_FINGER	-26	Is already enrolled finger number.
31	IS_VALID_ID	0	Usable ID

No	Name	Value	Description
32	**IS_VALID_FINGER**	0	Usable finger number
33	**ERROR_EMPTY_ID**	-29	Unable to find empty ID.
34	**ERROR_EMPTY_FINGER**	-30	There is no empty finger number.
35	**IS_MANAGER**	0	Is a database manager.
36	**ERROR_NOT_MANAGER**	-35	Is not a database manager.
37	**ERROR_SET_DATA**	-37	Unable to set data.
38	**ERROR_GET_DATA**	-38	Unable to get data.
39	**ERROR_INVALID_FILENAME**	-39	Filename is invalid.
40	**IS_SAME_FINGER**	0	Two fingerprint data are same.
41	**ERROR_DIFF_FINGER**	-41	Two fingerprint data are different.
42	**ERROR_INIT_DB**	-42	Database initialization failure.
43	**ERROR_OVER_DB**	-43	Database is full.

4. Installing Smack Bio1000 Pro

4.1 Configuration

Smack Bio1000 Pro (version1.0) consists of optical sensor, OCX and USB driver program.



Smack Bio 1000 Module

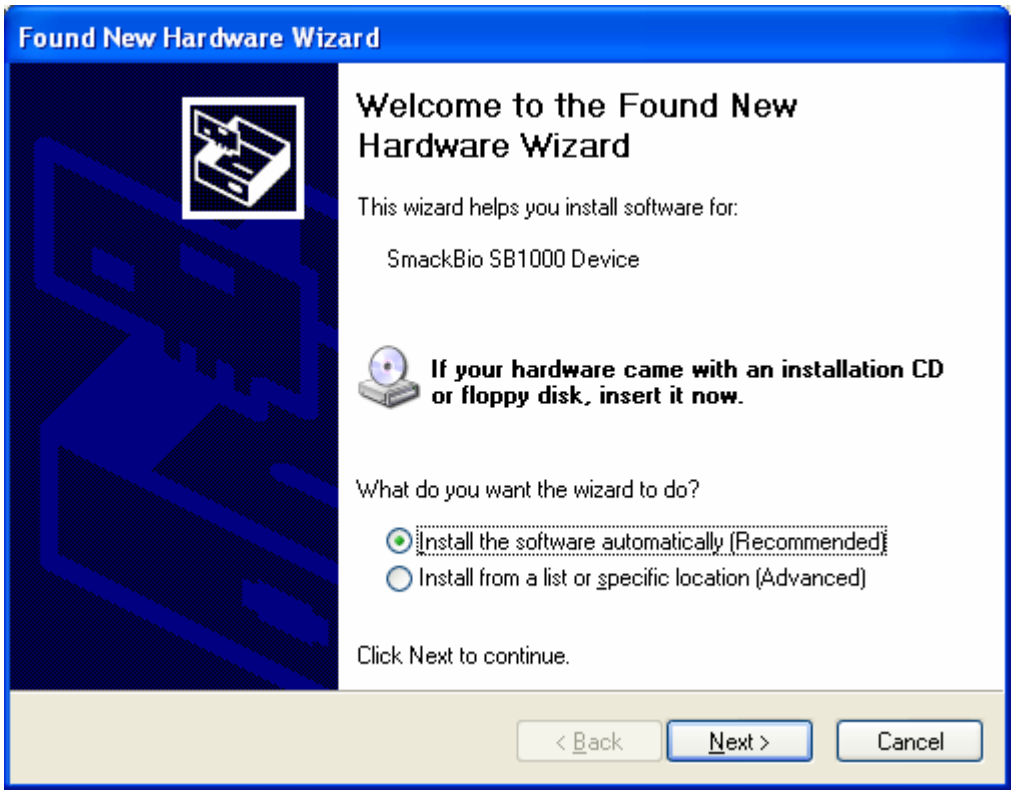
Optical sensor

USB Driver is in the \DRIVER folder of Setup disk and OCX installing program is \Software\Setup.exe.

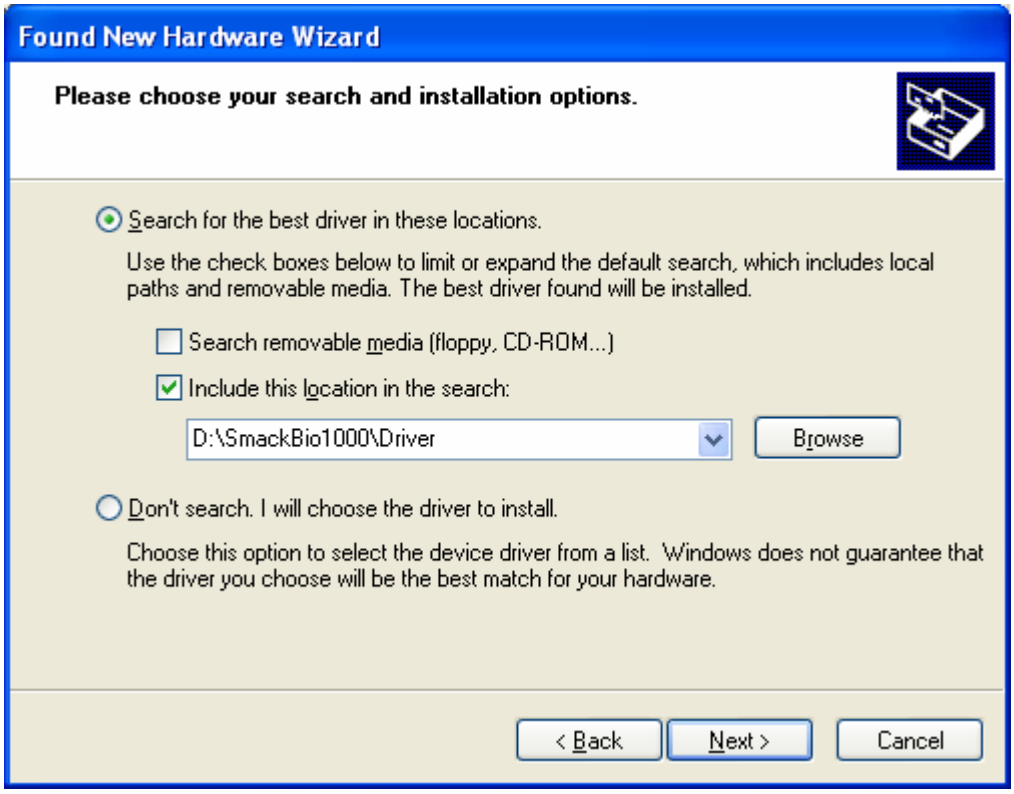
4.2 Installing

4.2.1 Connect an optical sensor to USB connector of computer.

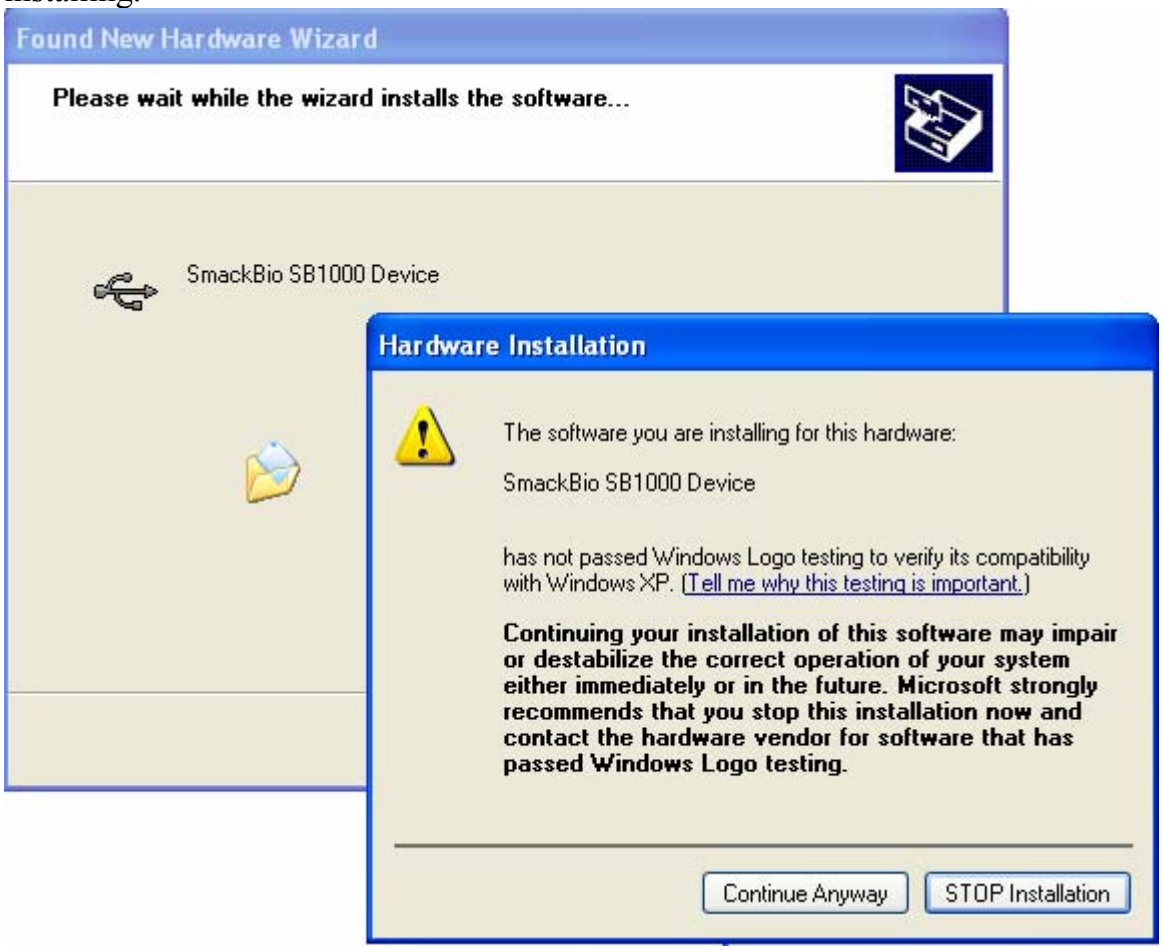
4.2.2 Following wizard appeared, select "Install from a list or specific location (Advanced)" and click "next" button.



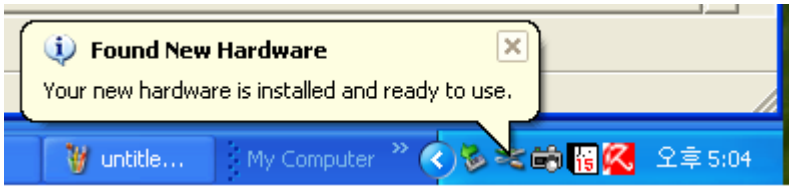
4.2.3 In next dialog select "Include this location in the search", click "Browse" and then appoint the place of Smack Bio1000 USB driver program. (For example D:\SmackBio1000\Driver)
Click "next" button.



4.2.4 Following message box appeared, select "Continue Anyway" and continue installing.



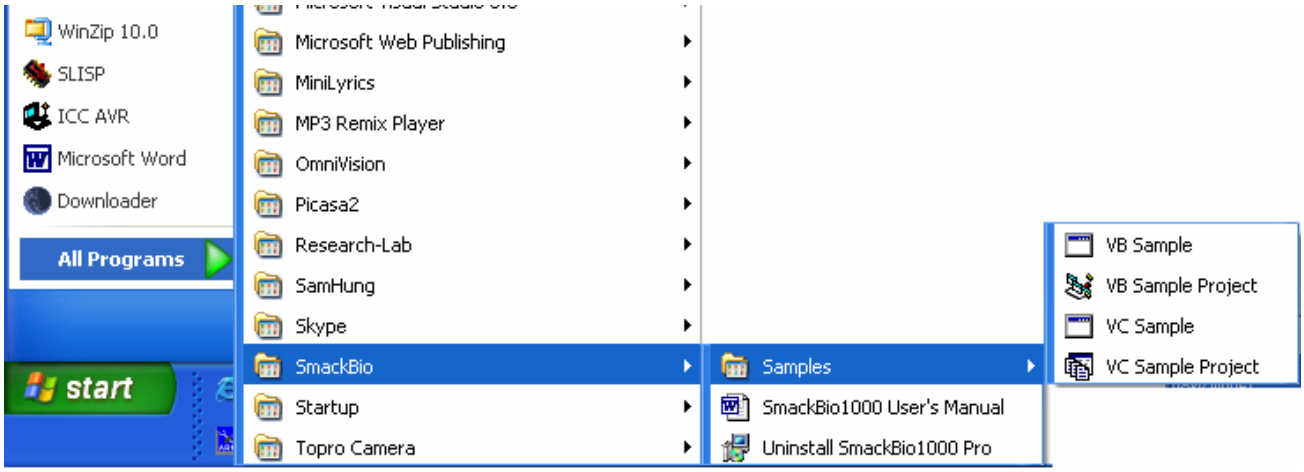
4.2.5 Installation finished normally, following balloon appears in the right bottom side.



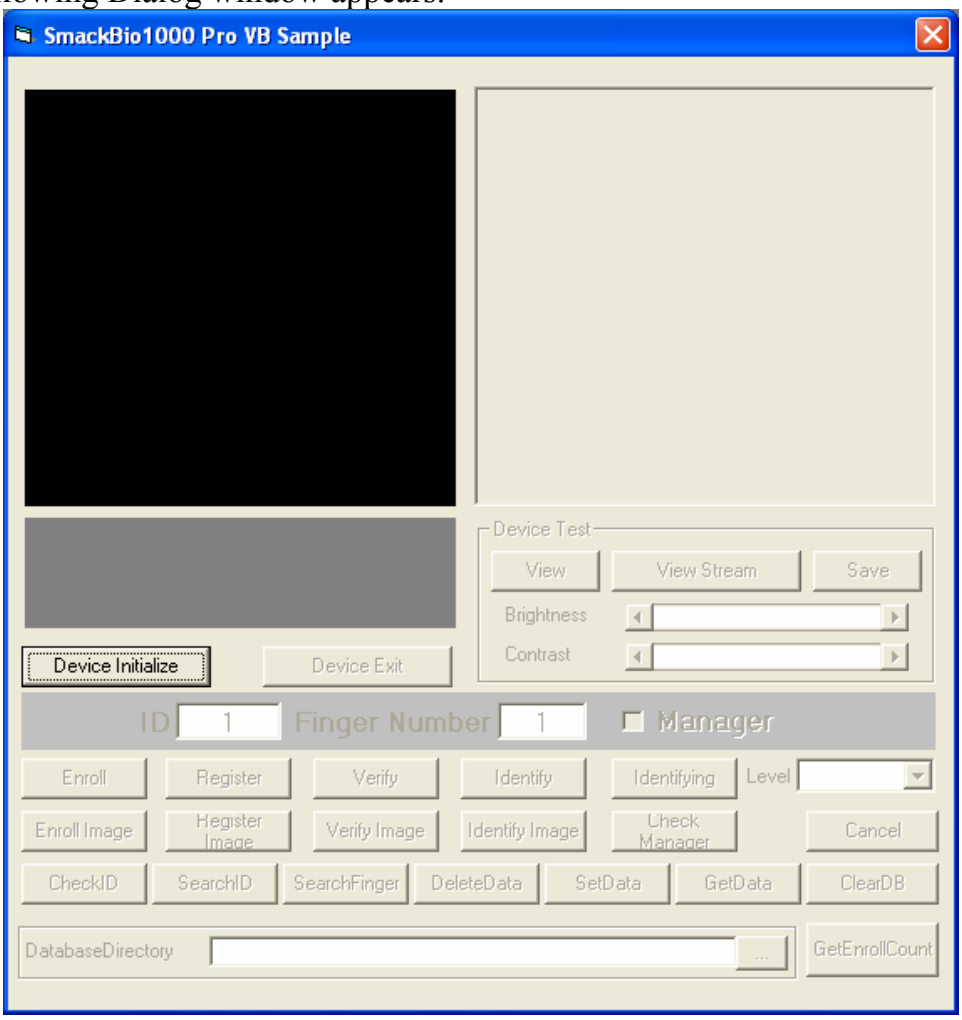
4.2.6 In Explorer open the folder to which OCX installation program belongs (for example, E:\Software) and execute Setup.exe program.

5. Examples of using Smack Bio1000 Pro

Start up Smack Bio\Samples\VB Sample of Start menu.

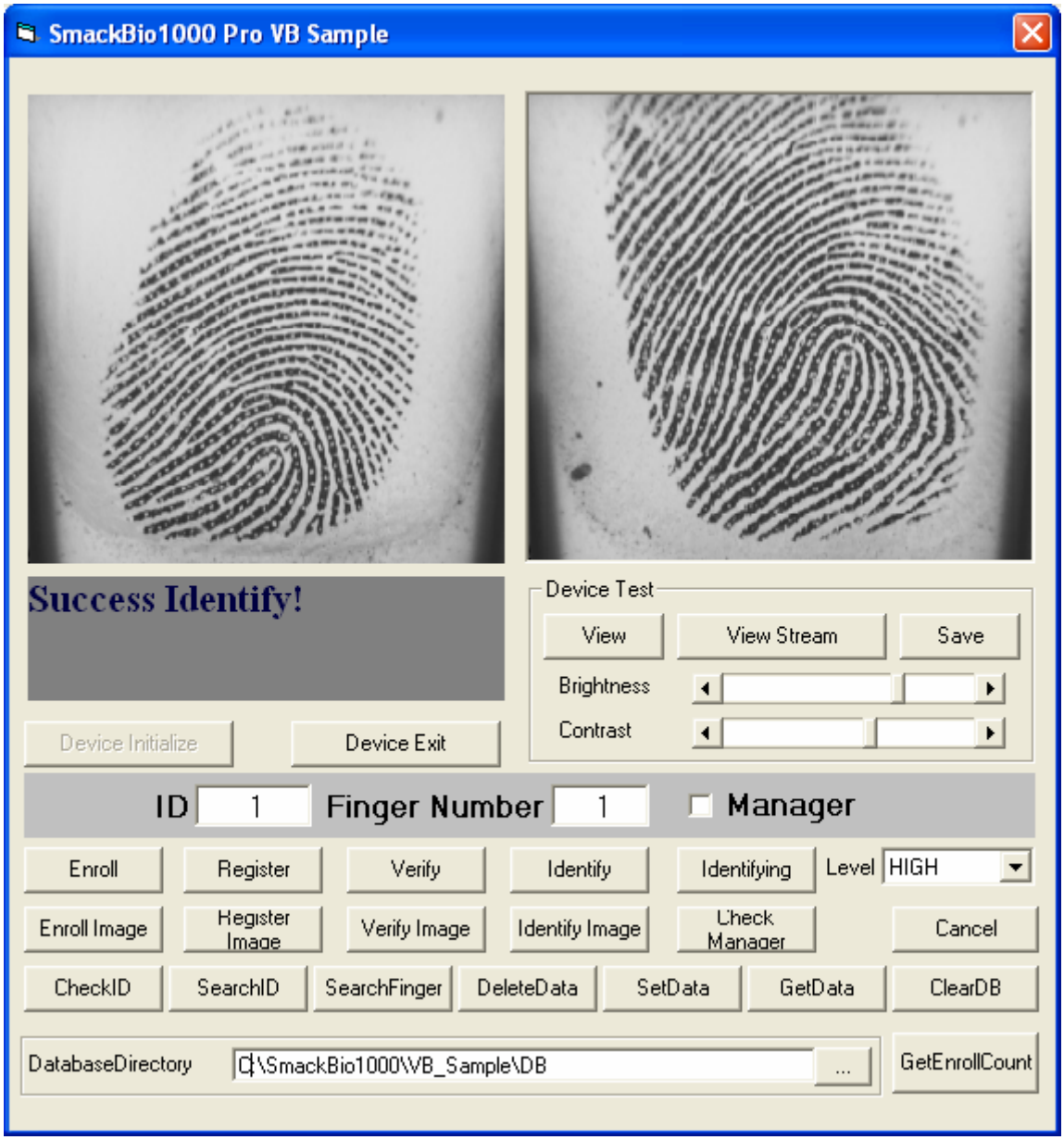


Then following Dialog window appears.



First click "Device Initialize" button.

The message as "Device Initialize OK!" appeared, user can initialize the sensor and use fingerprint identification module.



Functions of the buttons are as follows.

Device Exit : It disconnects with the sensor and initializes OCX.

View : Capture fingerprint image from the sensor and display it on the window of Smack Bio1000 Pro OCX.
To capture fingerprint image, user should click this button taking his finger on the sensor. (Refer to ViewFP Method.)

View Stream : Capture images seriously from the sensor and display them on the window. When user clicks this button, caption of button turns to "Stop Stream" and

images captured from the sensor are displayed on the window of Smack Bio1000 Pro OCX in real time.
If user clicks this button again, then caption turns to "View Stream" and displaying images is stopped.

Save : Display the image captured from the sensor on the window of Smack Bio1000 Pro OCX and save the image in the name specified by user.
To save fingerprint image, user should click this button taking his finger on the sensor. (Refer to SaveImageToFile Methord.)

Brightness : Control the brightness of fingerprint sensor. This control value is applied to all images captured from sensor thereafter.
User can verify the quality of the image using this function after turning to view status by clicking View Stream button (Refer to FPBrightness property.)

Contrast : Control the contrast of the fingerprint sensor. This control value is applied to all images captured from sensor thereafter.
User can verify the quality of the image using this function after turning to view status by clicking View Stream button (Refer to FPContrast property.)

ID, Finger Number : Display or set enroll ID and finger number using when enroll or identify.
ID can be specified from 1 to 32767 and fingerprint number from 1 to 10.
When enroll, user sets required ID and finger number on this TextBox and clicks Enroll button.
When 1:1 verify, user sets required ID and finger number on this TextBox and clicks Enroll button.
When 1:N identify, identified ID and finger number are displayed on this TextBox in the case of identification success.
And when use functions of CheckManager, CheckID, SearchID, SearchFinger, SetData, GetData, this TextBox is also used. (In detail refer to corresponding items.)

Enroll : It enrolls 3 fingerprints captured from a sensor to database.
User sets required ID and finger number on ID and FingerNumber TextBox separately and clicks this button.
Then according to the message, user may take on or off his finger on the sensor.
To cancel enroll process, click Cancel button.

In the case of enrollment failure, description about occurred error is displayed on the message box (Refer to EnrollStart Methord.)

Register : It enrolls 1 fingerprint captured from a sensor to database.
User sets required ID and finger number on ID and FingerNumber TextBox separately and clicks this button.
Then according to the message, user may take on or off his finger on the sensor.
To cancel enroll process, click Cancel button.
In the case of registration failure, description about occurred error is displayed on the message box (Refer to Register Methord.)

Verify : Verify (1:1 Verify) the fingerprint captured from a sensor with fingerprint of ID, finger number specified by user.
User sets required ID and finger number on ID and FingerNumber TextBox separately and clicks this button.
Then according to the message, user may take on or off his finger on the sensor.
To cancel verification process, click Cancel button.
If verification succeeds, then verified fingerprint image is displayed on the right PictureBox.
In the case of verification failure, description about occurred error is displayed on the message box (Refer to Verify Methord.)

Identify : Identify (1: N Identify) the fingerprint captured from a sensor with all fingerprint data in database.
First user clicks this button.
Then according to the message, user may take on or off his finger on the sensor.
To cancel identification process, click Cancel button.
If identification succeeds, then identified fingerprint image is displayed on the right PictureBox.
In the case of identification failure, description about occurred error is displayed on the message box (Refer to Identify Methord.)

Identifying : Seriously identify (1: N Identify) the fingerprints captured from a sensor.
First user clicks this button.
In this status, seriously taking on or off his finger on the sensor user can continue 1: N identification. .

To cancel serious identification process, click Cancel button.
If identification succeeds, then ID and finger number of identified fingerprint data are displayed on ID and FingerNumber TextBox and identified fingerprint image is displayed on the right PictureBox.
In the case of identification failure, 0 is displayed on ID and FingerNumber TextBox.

Enroll Image : It enrolls 3 specified fingerprint files to database.
User sets required ID and finger number on ID and FingerNumber TextBox and clicks this button.
Then select the files to enroll in turn on FileOpen Dialog.
To cancel enroll process, click Cancel button of FileOpen Dialog.
Files to enroll are Gray BMP files with size of 256×256.
In the case of identification failure, description about occurred error is displayed on the message box.

Register Image : It enrolls 1 specified fingerprint file to database.
User sets required ID and finger number on ID and FingerNumber TextBox and clicks this button.
Then select a file to enroll on FileOpen Dialog.
To cancel enroll process, click Cancel button of FileOpen Dialog.
Files to enroll are Gray BMP files with size of 256×256.
In the case of identification failure, description about occurred error is displayed on the message box.

Verify Image : Verify (1:1Verify) an input fingerprint file with the specified ID, finger number.
User sets required ID and finger number on ID and FingerNumber TextBox and clicks this button.
Then select a file to verify on FileOpen Dialog.
To cancel verification process, click Cancel button of FileOpen Dialog.
Files to verify are Gray BMP files with size of 256×256.
If verification succeeds, then verified fingerprint image is displayed on the right PictureBox.
In the case of verification failure, description about occurred error is displayed on the message box.

Identify Image : Identify (1: N Identify) an input fingerprint file with all fingerprint data enrolled in database.

After clicking this button, user selects a file to identify on FileOpen Dialog. To cancel identification process, click Cancel button of FileOpen Dialog. Files to identify are Gray BMP files with size of 256×256.
If verification succeeds, then ID and finger number of identified fingerprint data are displayed on ID and FingerNumber TextBox and the identified fingerprint image on the right PictureBox.
In the case of verification failure, 0 is displayed on ID and FingerNumber TextBox and description about occurred error is displayed on the message box.

Check Manager : Check whether fingerprint data of specified ID, FingerNumber is a manager of database or not.
User sets ID and finger number to check on ID and FingerNumber TextBox and clicks this button.
Then check result is displayed on the message box.
(Enrollment of database manager is set when fingerprint enrollment.
In this sample program, default setting when enroll is no manager.
User of Smack Bio1000 Pro OCX can specify the database manager when use the Register, Enroll, RegisterImage and EnrollImage functions. Refer to CheckManagerMethord)

CkeckID : Check whether ID inputted by user is enrolled in database or not.
User sets ID to check on ID TextBox and clicks this button.
According to this ID is already enrolled or not, "This ID has been enrolled" or "Is valid ID" is displayed on the message box. (Refer to CheckID Methord).

SearchID : Search empty (unenrolled) ID.
In the case of check success, checked ID is displayed on ID TextBox (Refer to SearchID Methord.)

SearchFinger : Search empty (unenrolled) finger number of specified ID.
In the case of check success, checked finger number is displayed on Finger Number TextBox (Refer to SearchFingerNumber Methord.)

DeleteData : Delete fingerprint data with specified ID, FingerNum from enrollment database. User sets ID and finger number to delete on ID and FingerNumber TextBox and clicks this button.

SetData : Insert fingerprint enroll data with specified ID, FingerNumber into database.
User sets ID and finger number to enroll on ID and FingerNumber TextBox

and clicks this button.
(In this sample program, empty fingerprint data is enrolled. User can insert any fingerprint data into database using SetFPData Methord of OCX.)

GetData : Read fingerprint enroll data with specified ID, FingerNum from database. User sets ID and finger number to get on ID and FingerNumber TextBox and clicks this button.
(In this sample program, obtained fingerprint data can not be used. User can get required fingerprint data from database using GetFPData Methord of OCX.)

ClearDB : Clear all fingerprint enroll data from database.(Refer to ClearDB Methord.)

DatabaseDirectory : Set the fingerprint database directory.(Refer to FPDatabaseDir Property.)