SmackFace1000 Module

User's Manual

Table of Contents

1 Introduction

This manual describes about the design specification of Smack Face1000, which is a Face Recognition + ID Card Time&Attendance machine and Access controller.

1.1 The aims of SmackFace1000

This is the complex personal authentication module which is combined with ID card and face recognition technology and it is a OEM module that composes T&A system or Access control system, in which the ID card number and face image data are transmitted from SmackFace1000 sensor to the server through a LAN or USB and they are verified with the data enrolled in the database.
It uses ID card information and personal face information at a time, so it raises the exactness of the private authentication system.
And also it is flexible to the various system constructions by the combination with the computer through LAN or USB.

※If the optical fingerprint sensor is connected with the module instead of the camera, the user can compose the system with fingerprint authentication and ID card identification.

1.2 Technical features of SmackFace1000

- SmackFace1000 module is composed the sensor device and the software (OCX).
- The sensor device is composed a CCD camera to capture a face, a sensor circuit to read ID card and a interface circuit to connect with the computer.
- OCX can manage five sensor devices at most, regardless of type of interface(USB, LAN).
- 10M/100M LAN connector
- USB2.0 interface
- Notification of the authentication result by Wiegand output, a LED and the buzzer.
- Enrollment and management function of the user.
- Face enrollment by a camera and 1:1 verification.

- Face image file enrollment and 1:1 verification.
- ID card enrollment and verification.

1.3 Technical features of the face-recognition engine using in SmackFace1000

- feature size per a face 2240 byte
- whole feature extraction time <= 260ms
  - ◆ face-detection time 140ms
  - ◆ feature extraction time 120ms
- matching time <= 0.01ms
- error rate

Level	**FRR** (False Reject Rate, %)
1	11.90
2	14.52
3	16.07
4	20.66
5	25.31

All mentioned times was measured about image size 640*480, using a computer with Intel P4 2.4GHz CPU and 512Mbytes RAM.

2 Basic concept of SmackFace1000

The programmer who develops the Time & Attendance and access control system using Smack Face1000 module is defined "Secondary Developer" in this document.

The user who has the authority to use the computer in which the program developed by the secondary developer is installed is defined "Manager" in this document.

The user who verifies using the program developed by the secondary developer is defined "User" in this document.

2.1 Requirements of SmackFace1000 OEM module

2.1.1 System requirements

- 128M RAM, 1GHz CPU
- Microsoft Windows 2000 / XP / 2003

2.1.2 Recommendation about face image

The quality of enrollment will play a crucial role especially in face recognition systems.
Complying with following requirements of enrollment, it will be possible to get enjoyable performance.

Enrollment image requirements:

- Sober face with eyes not covered by hair
- Eye looking right at Camera
- Frontal pose without head tilt and pan
- Face is fully visible in image and overall not too dark or bright
- Face size in image : not small than 1/4 of the image size

For immediate verification of enrolled user, followings are recommended.

Verification image recommendation:

Posture

Use the front face (entire face).
Rotation of the head must be less than ±5 degrees fro m frontal in every direction - nodded up/down, rota

ted left/right, tilted right/left.

Distance

Distance (between the camera lens and the front face) must be neither too far nor too near in order to capture the whole front face image as extensive as possible.

The appropriate distance for verification is 30 ~ 70 cm. On enrolling time, especially the distance, 30~40cm is good so that sufficient features are reflected.

Expression which must be prohibited

Closing eyes.

Covering eyes by hair

Frowning

A smile where the inside of the mouth is exposed

Rising up eyebrows.

Looking away from the camera.

Squinting

Rim of glasses covering part of the eye

Lighting

Lighting must be equally distributed on each side of face.

Specially, care must be taken to avoid saturation by extremely brightness or darkness and "hot spots" caused by one, high intensity, focused light source.

Eyeglasses

There should be no lighting artifacts on eyeglasses. This can typically be achieved by increasing the angle between the lighting, subject and camera to 45 degrees or more. If lighting reflections can't be removed, then the eyeglasses themselves should be removed.

Eyeglasses have to be of clear glass and transparent so the eyes and irises are clearly visible. Heavily tinted eyeglasses are not acceptable.

2.2 User classification(User's mode)

Users are divided into the following according to the combination method of the card and face recognition.

User A: Face + Card

User B: Only card

2.3 User enrollment

The user information saved in the database includes the card number, face features and user's mode.

A user has the only one unique ID in the database.

The card number is the secret number which is recorded in the card. In principle, the unique number is recorded in every card.

The face features of the user can be enrolled in the database using one or more face images by SmackFace1000 face-recognition engine. The principle of the face enrollment is to enroll under the actual working environment, namely module's installation position. The device can capture color images with the sizes 640*480 and 320*240, but the size 640*480 is recommended.

On enrolling time, attentions must be paid to the distance, lighting, expression and the rest in order to be extracted better features by face-recognition engine (refer to the Sec. 3.3).

Using several face images in verification, a better performance is achieved generally and the standard enrollment and verification functions of the OCX uses three face images per one user.

The OCX supplying with the device provides On-Line, Off-Line enrollment functions as a standard (refer to the Sec. 3.3). But the secondary developer can customize his own system by using the OCX functions as follows.

① Input the card number for users through SmackFace1000 and then enroll the inputted card number in the database with the personal information (name, birthday, post, etc) and the user's ID.

And also save the information that user's face is not enrolled yet.

② Gives a card to a user and get him to go to the front of the SmackFace1000 sensor and enroll manually, or else the secondary developer can organize the automatic enrollment that the user is insensible of his enrollment.
If the user inputs the card through the sensor, then the signal of input is transmitted to the computer.
At this time, because the information which means that whether the corresponding face is enrolled or not is saved in the database, get him to enroll automatically or save his face image and after that time manager can decides whether to enroll or not.
Because it is possible that a user gives other his card, it is better that the manager verifies the face and then enroll.

3 SmackFace1000 Outside structure

Figure 1 shows the outside structure of SmackFace1000.

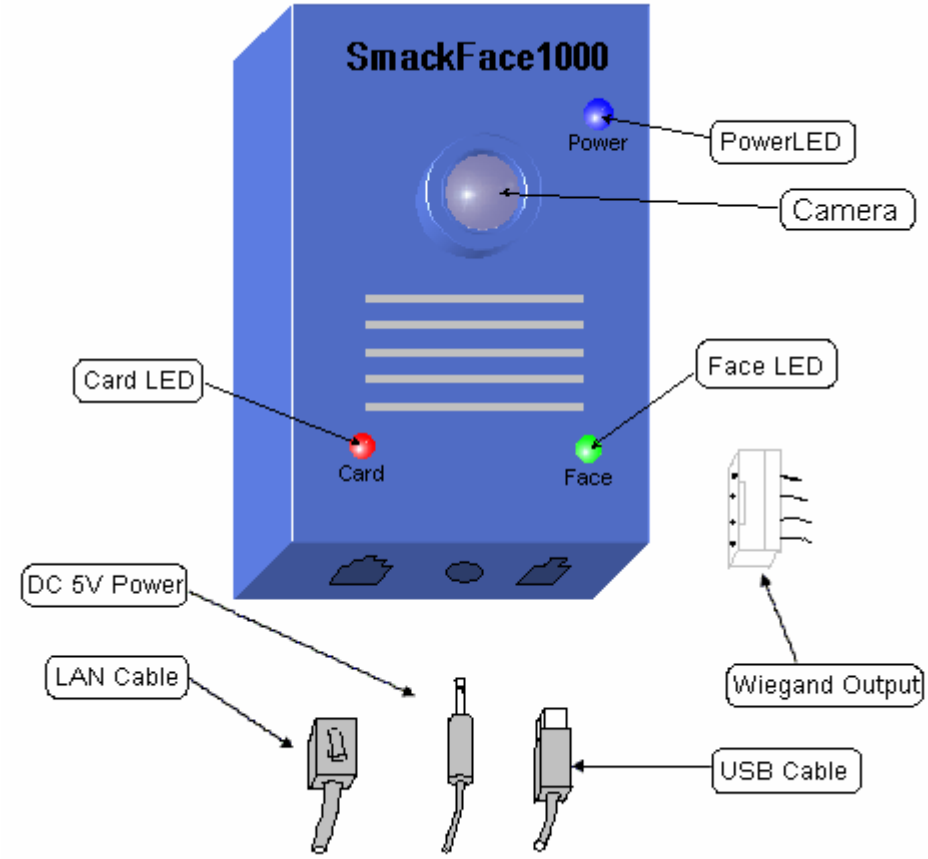

Figure 1. Outside structure of SmackFace1000

3.1 Outside structure

Power LED: This is the Blue LED and informs that the device is been enable to work. It is on when it is connected.

Card LED: This is a LED with two colors, red and green to inform the situation of reading card and identification.

Face LED: This is a LED with two colors, red and green to inform the situation of face verification.

LAN Cable: This is the cable to communicate between the sensor device and computer with Ethernet protocol.

Supply cable: This is the cable to supply DC 5V (1.2A).

USB cable: This is the cable to communicate between the sensor device and computer with USB 2.0 protocol.

Wiegand Output: The device can output the Wiegand signal

according to the command from the server.
The access control can be done with this signal.

3.2 Connection with computer

The device can be connected with computer through 10M/100 M LAN or USB cable. This device is not support DHCP protocol. IP address can be changed by the command from the compute r. The changed IP address is saved in the flash memory of the MCU. The device understands the IP address of the server by t he data from the server to the device.

When the device is connected with the computer through the LAN cable, USB cable should be not connected. And DC 5V power must be supplied to the device when it is connected through the LAN cable.

The device supports USB 2.0. In case of connecting through USB cable, it is not need to supply DC 5V.

3.3 The standard operation.

SmcakFace1000(device) can operate only when it is connected with a computer on which the supplied OCX is installed. If power is supplied, the Power LED of the device is always on.

When it is connected with a computer and on the operating-possible state (card-reading state), only the red lamp of Card LE D is on among all operation-notification LEDs (Card and Face LEDs). This state is called *initial state*.

The standard On-Line Enroll operation is as following.

- If an enroll-start command (OCX function ManEnrollStart) is sent to a device on initial state from the connected computer, the device is waiting for card sign.
- If user puts a card on the device, the device reads the card number and sends it to the computer.
- The computer searches the card number on the database. If it is already enrolled, the computer lets the device turn on the red lamp of its Card LED, emit a sound of long whistling for one second and switch to initial state, finishes the operation with fail. If it is not

enrolled and valid, lets the device turn on the green lamp of its Card LED and emit a sound of short whistling for 150 ms. And lets the device blink red lamp of Face LED.

- Repeat the following operation until three successful face image captures and feature extractions are accomplished or finish the enrollment with fail by using enroll-stop command (OCX function ManEnrollStop).

  The computer sends an image-capture command (OCX function ManCapture) to the device. If an image is transferred from device to the computer and the features are extracted successfully, the computer lets the device turn on green lamp of its Face LED, emit a sound of short whistling two times.

- If three successful face image captures and feature extractions are accomplished, the computer enrolls features to the database, lets the device turn on green lamp of its Face LED, emit a sound of short whistling three times for one second and switch to initial state and finishes the operation with success. If the enrollment is stopped in the middle, the computer lets the device turn on the red lamps of its Face and Card LEDs, emit a sound of long whistling for one second and switch to initial state and finishes the operation with fail.

The standard Verification operation is as following.

- If user puts a card on the device on initial, the device reads the card number and sends it to the computer.
- The computer searches the card number on the database. If it is a non-registered number, the computer lets the device turn on the red lamp of its Card LED, emit a sound of long whistling for one second and switch to initial state, finishes the operation with fail. If it is already enrolled and valid, lets the device turn on the green lamp of its Card LED and emit a sound of short whistling for 150 ms. And lets the device capture images with blinking red lamp of Face LED.
- If images are transferred from device to the computer, the features are extracted successfully, the computer verifies the features with one on the database and the calculated similarity is greater than the setting level, lets the device turn on green lamp of its Face LED, emit a sound of short whistling three times for one second and switch to initial state and finishes the operation with success. If feature extraction is failed or the calculate similarity is small, the

computer lets the device turn on the red lamps of its Face and Card LEDs, emit a sound of long whistling for one second and switch to initial state and finishes the operation with fail.

4 How to use SmackFace1000 OCX

SmackFace1000 OCX aims to provide the software interface for constructing Face recognition + ID card Time&Attendance system using SmackFace1000 module.
Using this OCX, it is possible to construct a user's own powerful and flexible authentication system based on face recognition and ID card.

※) Among the properties, methods and events that are described below, the ones whose names are followed a symbol "1" are valid only when the property"WorkingOrgMode" has the value, 1.

4.1 Properties

4.1.1 SFMachineCount

- Type of return value : LONG
- Range of return value : 0 ~ 5
- Default value :
- Read/Write property : Read
- Meaning : Count of the machines managed currently.

4.1.2 SFVerifyLevel

- Type of return value : LONG
- Range of return value : 1 ~ 5
- Default value : 2
- Read/Write property : Read/Write
- Meaning : Security level of face verification (refer to the Sec. 1.3).

4.1.3 WorkingOrgMode

- Type of return value : LONG
- Range of value : 1- Default Mode, 0 -

User Mode

- Default value : 1
- Read/Write property : Read/Write
- Meaning : Indicates whether SmackFace1000 monitors the attendance and leaving in the original mode or not.

4.1.4 SFDatabaseDir[1]

- Type of return value : String
- Range of return value :
- Default value : ""
- Read/Write property : Read/Write
- Meaning : Directory path which holds the database.

4.1.5 SFEnrollCount[1]

- Type of return value : LONG
- Range of value :
- Default value :
- Read/Write property : Read
- Meaning : Number of users in the database

4.1.6 SFManEnrollState[1]

- Type of return value : LONG
- Range of value : -1 ~ 2
- Default value :
- Read/Write property : Read
- Meaning : Represents the current state in user enrollment.
  -1 : It is not in user enrollment.
  0 , 1, 2: Notes that a user enrollment is in progress and how many times the

image captures and feature extractions are successful.

4.2 Methods

4.2.1 ConnectAll

●Function : Search all available devices and connect them with OCX.
●Declaration : ConnectAll() as Long
●Parameters :
●Return value : Number of the connected devices.

4.2.2 DisconnectAll

●Function : Disconnect with all of the connected devices.
●Declaration : DisconnectAll() as Long
●Parameters :
●Return value : Number of the disconnected devices.

4.2.3 SearchAvailableMachine

●Function : Search the available device.
●Declaration : SearchAvailableMachine(CommMode as Long) as Long
●Parameters : CommMode – Communication mode of the device to be searched. Refer to GetCommMode.
●Return value : If succeed, returns the machine number of the searched device.
If fails, returns -1.

4.2.4 ConnectMachine

●Function : Connect with the device which has the specified machine number using the specified communication mode.
●Declaration : ConnectMachine(CommMode as Long, M_No as Long) as Long
●Parameters : CommMode – Communication mode.
M_No – Machine number of the device to be connected.
●Return value : If succeed, return 0. If fails, return -1.

4.2.5 DisconnectMachine

●Function : Disconnect with the device which has the specified machine number.
●Declaration : DisconnectMachine(M_No as Long) as Long
●Parameters : M_No – Machine number of the device to be disconnected.
●Return value : If succeed, return 0. If fails, return -1.

4.2.6 GetCommMode

●Function : Get the communication mode of the device which has the specified machine number.
There are two types of communication modes, LAN and USB.
1 : LAN
2 : USB
●Declaration : GetCommMode(M_No as Long) as Long
●Parameters : M_No – Machine number of the device.
●Return value : If succeed, return the communication mode. If fails, return -1.

4.2.7 GetMachineIdx

●Function : Get the linked index of the specified device in the OCX.
●Declaration : GetMachineIdx(M_No as Long) as Long
●Parameters : M_No – Machine number of the device.
●Return value : If succeed, return the index. If fails, return -1.

4.2.8 GetMachineNo

●Function : Get the machine number of the specified device.
●Declaration : GetMachineNo(idx as Long) as Long
●Parameters : idx – Linked index of the device in the OCX.
●Return value : If succeed, return the machine number. If fails, return -1.

4.2.9 SetMachineNo

●Function : Change the machine number of the device which has the specified machine number. The machine number is in between 0 to 255 and not allowed duplicated.
●Declaration : SetMachineNo(oldM_No as Long, newM_No as Long) as Long
●Parameters : oldM_No – The current machine number of the device.
newM_No – The new machine number of the device.
●Return value : If succeed, return 0. If fails, return -1.

4.2.10 GetIPAddr

●Function : Get the IP address of the specified device.
●Declaration : GetIPAddr(M_No as Long) as String
●Parameters : M_No – Machine number of the device.

●Return value : If succeed, return the IP address of the device. If fails, return "0.0.0.0".

4.2.11 SetIPAddr

●Function : Set the IP address of the specified device.
●Declaration : SetIPAddr(M_No as Long, M_IPAddr as String) as Long
●Parameters : M_No – Machine number of the device.
M_IPAddr – IP address to be set.
●Return value : If succeed, return 0. If fails, return -1.

4.2.12 GetCaptureMode

●Function : Get the image capture mode of the specified device.
●Declaration : GetCaptureMode(M_No as Long) as Long
●Parameters : M_No – Machine number of the device.
●Return value : If succeed, return the image capture mode of the device (Refer to SetCaptureMode). If fails, return -1.

4.2.13 SetCaptureMode

●Function : Set the image capture mode of the specified device.
●Declaration : SetCaptureMode(M_No as Long, mode as Long) as Long
●Parameters : M_No – Machine number of the device.
mode – Image capture mode to be set.
1 : 640 * 480
2 : 320 * 240
●Return value : If succeed, return 0. If fails, return -1.

4.2.14 GetBrightness

●Function : Get the brightness of the camera of the specified device.

●Declaration : GetBrightness(M_No as Long) as Long
●Parameters : M_No – Machine number of the device.
●Return value : If succeed, return the brightness of the camera of the device. If fails, return -1.

4.2.15 SetBrightness

●Function : Set the brightness of the camera of the specified device. The value of brightness is in between 0 to 255.
●Declaration : SetBrightness(M_No as Long, value as Long) as Long
●Parameters : M_No – Machine number of the device.
value – Value of the brightness to be set.
●Return value : If succeed, return 0. If fails, return -1.

4.2.16 CaptureImage

●Function : Capture the face image data from the device and save into the buffer.
●Declaration : CaptureImage(M_No as Long, Buffer as Long, Size as Long) as Long
●Parameters : M_No – Machine number of the device which will capture an image
Buffer – Pointer of the buffer for captured image. It takes 3 bytes (R, G, B) per a dot.
Size – Size of the Buffer
●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.17 GetImageData

●Function : Pointer of the buffer which contains the last image captured by the specified device. It is upgraded whenever Capture Image, Display, ExtractFeatureFromDev

methods are called or <card+face> verification is proceeded.

●Declaration : GetImageData(M_No as Long) as Long

●Parameters : M_No – Machine number of the device.

●Return value : If succeed, return the pointer of the buffer which contains the image. If fails, return -1.

4.2.18 SaveImage

●Function : Save the last image captured by the specified device as the type of "BMP" (Refer to GetImageData).

●Declaration : SaveImage(M_No as Long, FileName as String) as Long

●Parameters : M_No – Machine number of the device.
FileName – File name to save image.

●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.19 Display

●Function : Capture the face image from the device and display it.

●Declaration : Display(M_No as Long, Long hDC, Long X0, Long Y0, Long Width, Long Height, Long bNew) as Long

●Parameters : M_No – Machine number of the device which will capture an image
hDC – Device context handle of the window to display the captured image.
X0 – X coordinate of the upper-left corner of the image.
Y0 – Y coordinate of the upper-left corner of the image.
Width – Width of the image.
Height – Height of the image.

bNew – Decide whether capture a new image from the device and display (value is 0) or display the last image saved (value is 1).

●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.20 IsFaceImage

●Function : Capture the image data from the device and decide whether it is a face image or not.

●Declaration : IsFaceImage(M_No as Long, Buffer as Long, Size as Long) as Long

●Parameters : M_No – Machine number of the device which will capture an image
Buffer – Pointer of the buffer for the captured image.
Size – Size of the Buffer

●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.21 IsFaceImageFile

●Function : Decide whether a file is the face image file or not.

●Declaration : IsFaceImageFile(FileName as String) as Long

●Parameters : FileName – Image file name.

●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.22 ExtractFeatureFromDev

●Function : Capture face image from the device and extract its features.

●Declaration : ExtractFeatureFromDev(M_No as Long, Buffer as Long, Size as Long) as Long

●Parameters : M_No – Machine number of the device which will capture an image.
Buffer – Pointer of buffer for features. Its size must be more than 2240 bytes.
Size – Size of the buffer.

●Return value : If succeed, return value is 0.
If fails, return value is less than 0.

4.2.23 ExtractFeatureFromFile

●Function : Extract the features from a face image file.

●Declaration : ExtractFeatureFromFile(FileName as String, Buffer as Long, Size as Long) as Long

●Parameters : FileName – Image file path
Buffer – Pointer of buffer for features. Its size must be more than 2240 bytes.
Size – Size of the buffer

●Return value : If succeed, return value is 0.
If fails, return value is less than 0.

4.2.24 Match

●Function : Compare two features and decide whether they are same or not.

●Declaration : Match(Buffer1 as Long, Size1 as Long, Buffer2 as Long, Size2 as Long) as long

●Parameters : Buffer1 – Pointer of the buffer for feature 1.
Size1 – Size of Buffer1
Buffer2 – Pointer of the buffer for feature 2.
Size2 – Size of Buffer2
Buffer1 and Buffer2 must be more than 2240bytes.

●Return value: 1 on the case that the similarity of the two features is bigger than the specified

threshold level (SFVerifyLevel), else 0.

4.2.25 SendWiegand

●Function : Let the device output a wiegand signal.
●Declaration : SendWiegand(long M_No, long ID) as Long
●Parameters : M_No – Machine number of the device.
ID – User's ID.
●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.26 CardReaderOn

●Function : Switch On/Off the card reader module of the device.
●Declaration : CardReaderOn(M_No as Long, On as Long) as Long
●Parameters : M_No – Machine number of the device.
On – If value is 0, switch off the module. Or else switch on it.
●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.27 BuzzerOn

●Function : Switch On/Off the buzzer of the device.
●Declaration : BuzzerOn(long M_No, long On, long Period) as Long
●Parameters : M_No – Machine number of the device.
On – If value is 0, switch off the buzzer. Or else switch on it.
Period – When the buzzer is switched on, this decides how much time it sounds for. The value 0 means that it sounds without limit. Time unit is 50 micro seconds.
●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.28 LEDCardGreenOn

●Function : Switch On/Off or blink the green LED of the card verification LED of the device.
●Declaration : LEDCardGreenOn(M_No as Long, Option as Long) as Long
●Parameters : M_No – Machine number of the device.
Option – The value 0 means switching off. The value 1 means switching on and the value 2 means blinking.
●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.29 LEDCardRedOn

●Function : Switch On/Off or blink the red LED of t he card verification LED of the device.
●Declaration : LEDCardRedOn(M_No as Long, Option as Long) as Long
●Parameters : M_No – Machine number of the device.
Option – The value 0 means switching off. The value 1 means switching on and the value 2 means blinking.
●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.30 LEDFaceGreenOn

●Function : Switch On/Off or blink the green LED of the face recognition LED of the device.
●Declaration : LEDFaceGreenOn(M_No as Long, Option as Long) as Long
●Parameters : M_No – Machine number of the device.
Option – The value 0 means switching off. The value 1 means switching on and the value 2 means blinking.
●Return value : If succeed, return value is 0.

If fails, return value is -1.

4.2.31 LEDFaceRedOn

●Function : Switch On/Off or blink the red LED of the face recognition LED of the device.
●Declaration : LEDFaceRedOn(M_No as Long, Option as Long) as Long
●Parameters : M_No - Machine number of the device.
Option - The value 0 means switching off. The value 1 means switching on and the value 2 means blinking.
●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.32 SFAction

●Function : Let the device perform the pre-defined signal actions using the buzzer and LEDs.
●Declaration : SFAction(long M_No, long ActionNo) as Long
●Parameters : M_No - Machine number of the device.
ActionNo - The value 94 means that the device emits a sound of short whistling two times (350ms). The value 95 means that the device emits a sound of short whistling three times (350ms).
●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.33 ManEnrollStart[1]

●Function : Start a manual enrollment.
●Declaration : ManEnrollStart(ID as Long, name as String, mode as Long, bSave as Long) as Long
●Parameters : ID - User's ID to enroll
name - User's name to enroll. (max 20 bytes)

mode – Verification mode
bSave – Decide whether image data will be saved as a file or not. If it is TRUE, the image data will be saved as the name "xxxxxx_y.bmp" in the "img" subdirectory of the database directory. Here, "xxxxxx" indicates the ID value and "y" indicates the number of the image among the 3 images used in enrollment.

●Return value : If succeed, return 0. If fails, return -1.

4.2.34 ManEnrollStop[1]

●Function : Stop the manual enrollment in progress.
●Declaration : ManEnrollStop() as Long
●Parameters :
●Return value : If succeed, return 0. If fails, return -1.

4.2.35 ManCapture[1]

●Function : In a manual enrollment, capture an image from the device and extract features. This function must be used after a manual enrollment is started and a card sign is inputted from a device. Whenever a successful call is done, SFManEnrollState value is updated. After three successful call of ManCapture, OCX registers the user information and face features to the database and finishes the manual enrollment with success.
●Declaration : ManCapture() as Long
●Parameters :
●Return value : If succeed, return 0. If fails, return -1.

4.2.36 Enroll[1]

●Function : Read the ID card from the device and enroll the user on inputted ID with the inputted verification mode.
In the case of the <ID + face> verification mode, also read the face image data from the device, extract features and enroll.

●Declaration : Enroll(ID as Long, name as String, mode as Long, bSave as Long) as Long

●Parameters : ID – User's ID to enroll
name – User's name to enroll. (max 20 bytes)
mode – Verification mode
bSave – Decide whether image data will be saved as a file or not. Refer to ManEnrollStart.

●Return value : If succeed, return value is more than 0. If fails, return value is -1.

4.2.37 OffLineEnroll[1]

●Function : Enroll with the ID, card number and verification mode.
In case of the <ID + Face> verification mode, read the image data from the files, extract the features and enroll.

●Declaration : OffLineEnroll(ID as Long, CardNo as Long, name as String, mode as Long, filenames as Variant, bSave as Long) as Long

●Parameters : ID – User's ID to enroll
CardNo – User's card number to enroll
name – User's name to enroll (max 20bytes)
mode – User's verification mode to use

filenames – The path of the face image files.
It is valid when mode value is 3, namely <ID + Face> mode.
bSave – Decide whether save the image data as a file or not. Refer to ManEnrollStart

●Return value : If succeed, return value is more than 0.
If fails, return value is -1.

4.2.38 RegisterItem[1]

●Function : Enroll the card number in the database.
●Declaration : RegisterItem(ID as Long, name as String, CardNo as Long, mode as Long) as Long
●Parameters : ID – User's ID to enroll
name – User's name to enroll
CardNo – User's card number to enroll
Mode – User's verification mode.
Refer to GetUserType
●Return value : If success, return value is 0.
If fails, return value is less than -1.

4.2.39 Delete[1]

●Function : Delete the ID and card number in the database.
●Declaration : Delete(ID as Long, CardNo as Long) as Long
●Parameters : ID – User's ID enrolled in the database.
CardNo – Card number on the ID.
●Return value : If succeed, return 0.
If fails, return -1.

4.2.40 DeleteAll[1]

●Function : Delete all data in the database.

●Declaration : DeleteAll() as Long
●Parameters :
●Return value : If succeed, return 0.
If fails. return -1.

4.2.41 Verify[1]

●Function : Capture the image data from the device, extract the features and verify one to one with the features in the database.
●Declaration : Verify(M_No as Long, ID as Long, CardNo as Long, TryCount as Long) as Long
●Parameters : M_No – Machine number of the device which will capture an image
ID – User's ID to verify
CardNo – Card number
TryCount – Retry count on fail.
●Return value : If succeed, return value is 1.
If fails, return value is 0.

4.2.42 VerifyFromFile[1]

●Function : Extract the features from the face image file and verify one to one with the features in the database.
●Declaration : VerifyFromFile(ID as Long, CardNo as Long, FileName as String) as Long
●Parameters : ID – User's ID to verify.
CardNo – Card number
FileName – The face image file path
●Return value : If succeed, return value is 1.
If fails, return value is 0.

4.2.43 SearchEmptyID[1]

●Function : Search a new ID to enroll in the database.
●Declaration : SearchEmptyID() as Long

●Parameters :
●Return value : ID value

4.2.44 GetIDFromCardno[1]

●Function : Get the ID which the specified card number was enrolled with by searching the database.
●Declaration : GetIDFromCardno(CardNo as Long) as Long
●Parameters : CardNo – Card number to search
●Return value : Return ID if the card number was enrolled or 0 otherwise.

4.2.45 GetCardnoFromID[1]

●Function : Get the card number which the specified ID was enrolled with by searching the database.
●Declaration : GetCardnoFromID(ID as Long) as Long
●Parameters : ID – ID of a user enrolled in the database
●Return value : Return the card number if it was enrolled or 0 otherwise.

4.2.46 GetUserName[1]

●Function : Get user's name.
●Declaration : GetUserName(ID as Long) as String
●Parameters : ID – User's ID enrolled in the database
●Return value : User's name

4.2.47 SetUserName[1]

●Function : Set user's name.
●Declaration : SetUserName(ID as Long, name as String) as Long
●Parameters : ID – User's ID enrolled in the database
name – User's name to be set.

●Return value : If succeed, return value is 0.
If fails, return value is -1.

4.2.48 GetUserType[1]

●Function : Get the user's verification mode.
Verification mode has two types of <ID card + face> mode and <ID card> mode.
●Declaration : GetUserType(ID as Long) as Long
●Parameters : ID – User's ID enrolled in the database
●Return value : If return value is 1, <ID card> mode
If return value is 3, <ID card + face> mode.

4.2.49 GetFeatureFromDB[1]

●Function : Get the face features which is corresponding to the ID and card number from the database.
●Declaration : GetFeatureFromDB(ID as Long, CardNo as Long, Buffer as Long, Size as Long) as Long
●Parameters : ID – ID to get the features.
CardNo – Card number.
Buffer – Pointer of the buffer for the enrollment data.
Its size must be more than 2240 bytes.
Size – Size of the buffer.
●Return value : If succeed, return value is 0.
If fails, return value is less than 0.

4.2.50 SetFeatureToDB[1]

●Function : Set the features on the ID and card number in the database.
●Declaration : SetFeatureFromDB(ID as Long, CardNo as Long,Buffer as Long, Size as Long) as Long

●Parameters : ID – User's ID to set
CardNo – Card number
Buffer – Pointer of the buffer for the features.
Size – Size of the buffer

●Return value : If succeed, return value is 0.
If fails, return value is less than 0.

4.2.51 GetLogCount[1]

●Function : Get the count of the log data in the database.

●Declaration : GetLogCount() as Long

●Parameters :

●Return value : If succeed, return value is the count.
If fails, return value is -1.

4.2.52 GetLogInfo[1]

●Function : Get a log data in the database.

●Declaration : GetLogInfo(no as Long, pLogInfo as Long) as Long

●Parameters : no – Log data number to get.
pLogInfo – Pointer of the buffer for the log data.

●Return value : The log data number

Log data type is the following.

Type LOGITEM

Id As Long	'User's ID of the log data
CardNo As Long	'User's Card number
mode As Long	'Verification mode
vYear As Long	'Year
vMonth As Long	'Month
vDate As Long	'Date
vHour As Long	'Hour
vMin As Long	'Minute
vSec As Long	'Second

```
    wTime As Long          'Response time (ms)
    Ok As Long             'Verification result
                           (success-1, fail-0)
    tryCount As Long       'Number of verification
                           attempt
    M_No As Long           'Machine number of the used
                           device

End Type
```

4.2.53 DeleteAllLog[1]

- ●Function : Delete all of the log data in the database.
- ●Declaration : DeleteAllLog() as Long
- ●Parameters :
- ●Return value : If succeed, return value is 0.
  If fails, return value is -1.

4.3 Events

4.3.1 OnReceiveCardSign

- ●Function : Informs the card number when read the card from the device.
- ●Declaration : OnReceiveCardSign(M_No as Long, CardNo as Long)
- ●Parameters : M_No - Machine number of the device that read the card sign.
  CardNo - Card number read

4.3.2 OnVerify[1]

- ●Function : Informs the verification result after the verification if WorkingOrgMode = 1. On the case that a card sign is inputted after a manual enrollment starts, it also informs

whether the card number is possible to be registered.

●Declaration : OnVerify(M_No as Long, ID as Long, Result as Long)

●Parameters : M_No – Machine number of the used device.

ID – Verified ID.

Result – Result

On verification.

-1 : fail.

1 : <ID card + Face> verification success

2 : <ID card> verification success.

On manual enrollment.

-1 : The inputted card number is impossible to be registered.

1 : <ID card + Face > enrollment is possible.

2 : <ID card> enrollment is successful.

5 SmackFace1000 Software Package

5.1 Package components

SmackFace1000 Software Package consists of SF1000PC OCX, USB2.0 driver for SmackFace1000 device, two program sources that are written in Visual Basic and Visual C++ and user's manual.

Each component is placed at the following paths.

- [SysDir]₩ - SF1000PC OCX.
- [AppDir]₩ - User's manual.
- [AppDir]₩Driver - USB2.0 driver for SmackFace1000 device.
- [AppDir]₩VBSample - the program sources written in Visual Basic.
- [AppDir]₩SF1000CustomDemo - the program sources written in Visual C++.

※) Here, [SysDir] is the system directory of Windows system and [AppDir] is the setup directory of the Package.

5.2 Demo program1(Visual Basic)

This demo program uses the standard operation mode and is written in Visual Basic (property WorkingOrgMode = 1).

It performs the data management, user enrollment, verification and result notification using the standard functions of the OCX and so it is simple relatively to demo program2.

5.2.1 Interface

Figure 2 shows the interface of the demo program1.

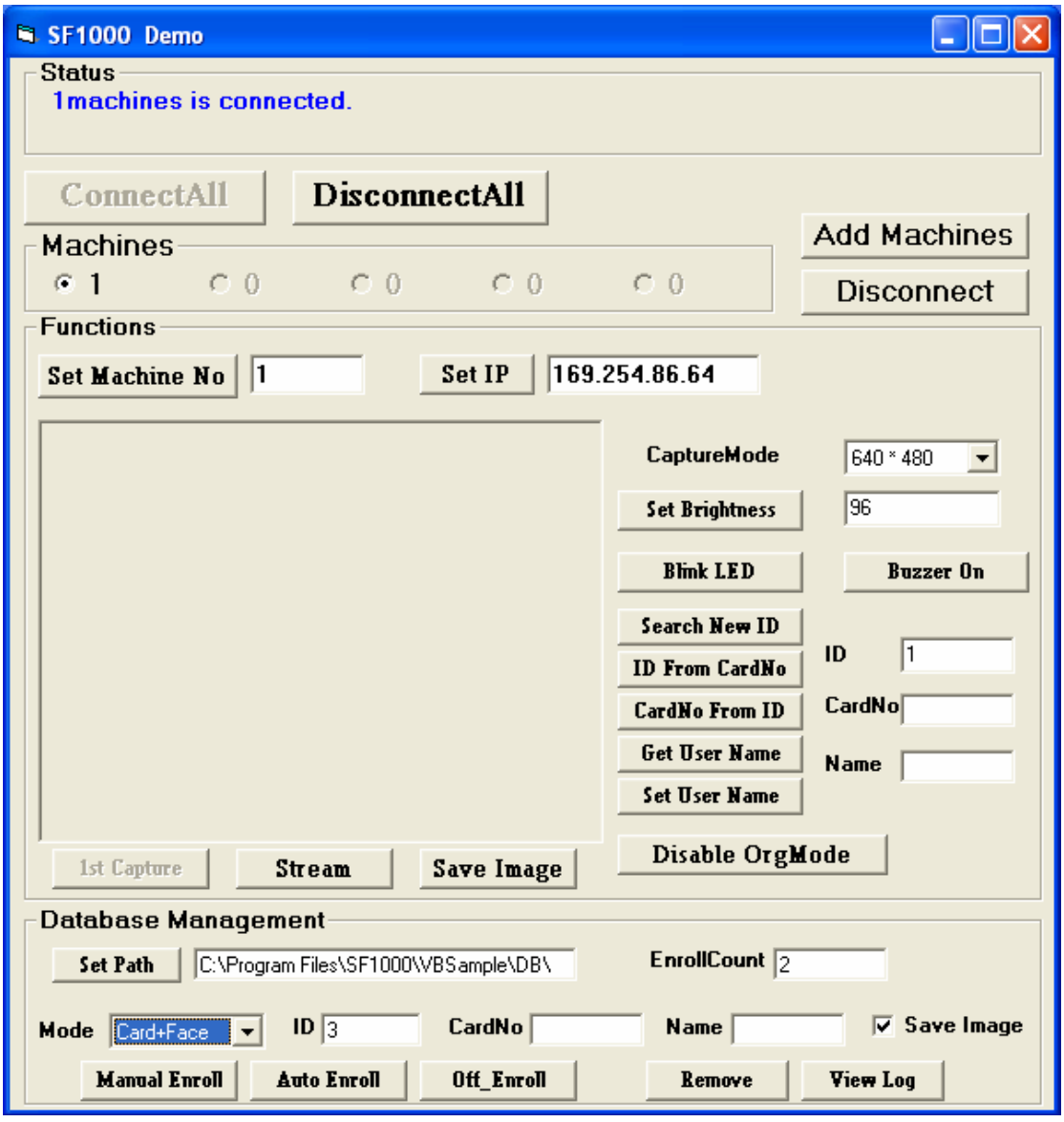

Figure 2. Demo program1 using OCX

In the part Status, messages about connecting and disconnecting of the devices, incoming card sign, verification result and enrolling and removing of a user are displayed.

The part Machines shows the devices connected with the OCX and user can select a device that he wants to control.

In the part Functions, there are the controls for execution of the OCX functions related with device.

In the part Database Management, there are the controls for execution of the OCX functions related with database management.

5.2.2 Functions of the controls.

"ConnectAll" button connects all the available devices with the OCX.

"DisconnectAll" button disconnects all of the devices connected with the OCX.

"Disconnect" button disconnects the selected machine with the OCX.

"Add Machines" button finds an available device and connects it with the OCX.

"Set Machine No" button sets the machine number of a device.

"Set IP" button sets the IP address of a device.

"Capture Mode" combobox sets the image capturing mode of a device.

"Set Brightness" button sets the brightness of the camera in a device.

"Blink LED" button lets the red lamp of the Face LED of a device blink. Press it again and then the blinking is stopped.

"Buzzer On" button lets the buzzer of a device sound. Press it again and then the sounding is stopped.

"Search New ID" button searches the minimum ID possible to register.

"ID From CardNo" button searches the ID which the specified card number is registered with.

"CardNo From ID" button searches the card number registered with the specified ID.

"Get User Name" button searches the name of the user registered with the specified ID.

"Set User Name" button sets the name of the user registered with the specified ID.

"Disable OrgMode" button enables or disables the standard verification function.

"1st Capture" buttons captures an image from a device and extracts features on manual enrollment.

"Stream" button captures images from a device and displays them continuously. Press it again and then the capturing is stopped.

"Save Image" button captures an image from a device and saves it with the specified filename.

"Set Path" button sets the path of the database of the OCX.

"Mode" combobox sets the verification mode on user enrollment.

"Save Image" checkbox decides whether the images will be saved as files on enrollment.

"Manual Enroll" button starts or stops a manual enrollment.

"Auto Enroll" button performs an automatic enrollment.

"Off_Enroll" button performs an off-line enrollment.

"Remove" button removes a user registered with the specified ID from the database.

"View Log" button shows the logs stored in the database.

5.2.3 Using.

- Connection and disconnection of devices.

Connect the devices with the computer by using supply cable and LAN cable or USB cable according to their communication mode.

Power on the devices and then power LED is turned on and the devices emit a sound of short whistling three times.

Press "ConnectAll" button and then all the available devices are connected with the OCX, emitting a sound of short whistling two times.

When a device is to be added later, connect the device with the computer using cable first, power on , press "Add

Machines" button next and then it is connected with the device in addition. An OCX can manage five devices at most.
Press "DisconnectAll" button and then all of the connected devices are disconnected emitting a sound of short whistling two times.
Unlike this "Disconnect" button disconnects only the selected machine.

- Enrolling and removing a user.

Three face images are used in the enrollment of a user. The buttons in the part Management are used.

•Manual enrollment

Input ID (required, not duplicated) and name (if needed) to be registered in the ID edit box and Name edit box first.
Press "Manual Enroll" button. Then the caption of this button changes to "Stop Enrolling" and it is possible to stop the enrollment at any time by using it. Put the card on the device and then the captured images are displayed continuously. When the fine full face images are captured, press "1st Capture" button. If the first image enrollment is successful, the caption of the button changes to "2nd Capture" and it is possible second image enrollment. (If the feature extraction from the image fails, the caption of the button does not change.) If three image enrollments are successful with "1st Capture" button changing to "2nd Capture", "3rd Capture" button in this way, the user enrollment is finished successfully.

•Automatic enrollment

Input ID and Name as in the manual enrollment.
Press "Auto Enroll" button. If user puts the card on the device within 5 seconds, from that time the OCX captures three images continuously with a certain interval (about 0.5 second) and enrolls them.

•Off-Line enrollment

Input ID and Name as in the manual enrollment.
User must input a card number manually in the CardNo edit

box instead of input from a device because of off-line enrollment. (Whenever one puts a card on the device, the card number is displayed in the CardNo edit box and so it can be used on off-line enrollment later.)
Press "Off_Enroll" button and then a file open dialog is opened. Select an image file and press "Open" button and then a file open dialog is opened. If three files are selected in this way, the enrollment is finished.
Before off-line enrollment, the face images that will be used in the enrollment can be captured and saved using this program. Press "Stream" button in the part Functions and then images are displayed continuously. Press "Save Image" button beside it and then a file save dialog is opened and the captured image on the moment can be saved with the specified file name. Press "Stop" button which "Stream" button changed to and then the image capturing is stopped.

•Enrollment can fail on the following cases.

When the feature extraction from an image was failed

When the specified ID or card number is registered already

Not duplicated ID number must be inputted into the ID edit box. The demo program searches an available ID and inputs it into the ID edit box by itself whenever it is connected with devices or an enrollment is successful. And it is possible to find an available ID by using "Search New ID" button in the part Functions.

When user doesn't put a card on the device within a certain time (on automatic enrollment)

- Verification.

When a user puts the card on the device, the OCX reads the card number and investigates whether it is a registered card. If it is registered, OCX captures images from the device, extracts features, verifies with the registered features and informs the result as a message. On success, the green lamps of the two LED are turned on and a short whistle is sounded three times. On fail the red lamps of the two LED are turned on and a long whistle is sounded. The result is

displayed in the part Status.

It is possible to disable the standard verification function of the OCX by pressing "Disable OrgMode" button in the part Functions. On this time, the caption of the button changes to "Enable OrgMode" and the OCX only informs the card number without verification. Press this button again and then the standard verification function is enabled. This function is useful when a user wants to use a custom verification designed by himself.

- The others.

Other functions of OCX can be used by using the controls in the part Functions. On this time, a device to be controlled must be selected correctly in the part Machines before setting its machine number, IP address, image capture mode, brightness of the camera and so on.

5.3 Demo Program2(Visual C++)

This demo program doesn't use the standard operation mode and is written in Visual C++ (property **WorkingOrgMode** = 0).

It performs the same functions as ones of the standard operation mode and describes that a user defines the data management, user enrollment, verification and result notification in his own way and uses them.

5.3.1 Interface

Figure 3 shows the interface of the demo program2.

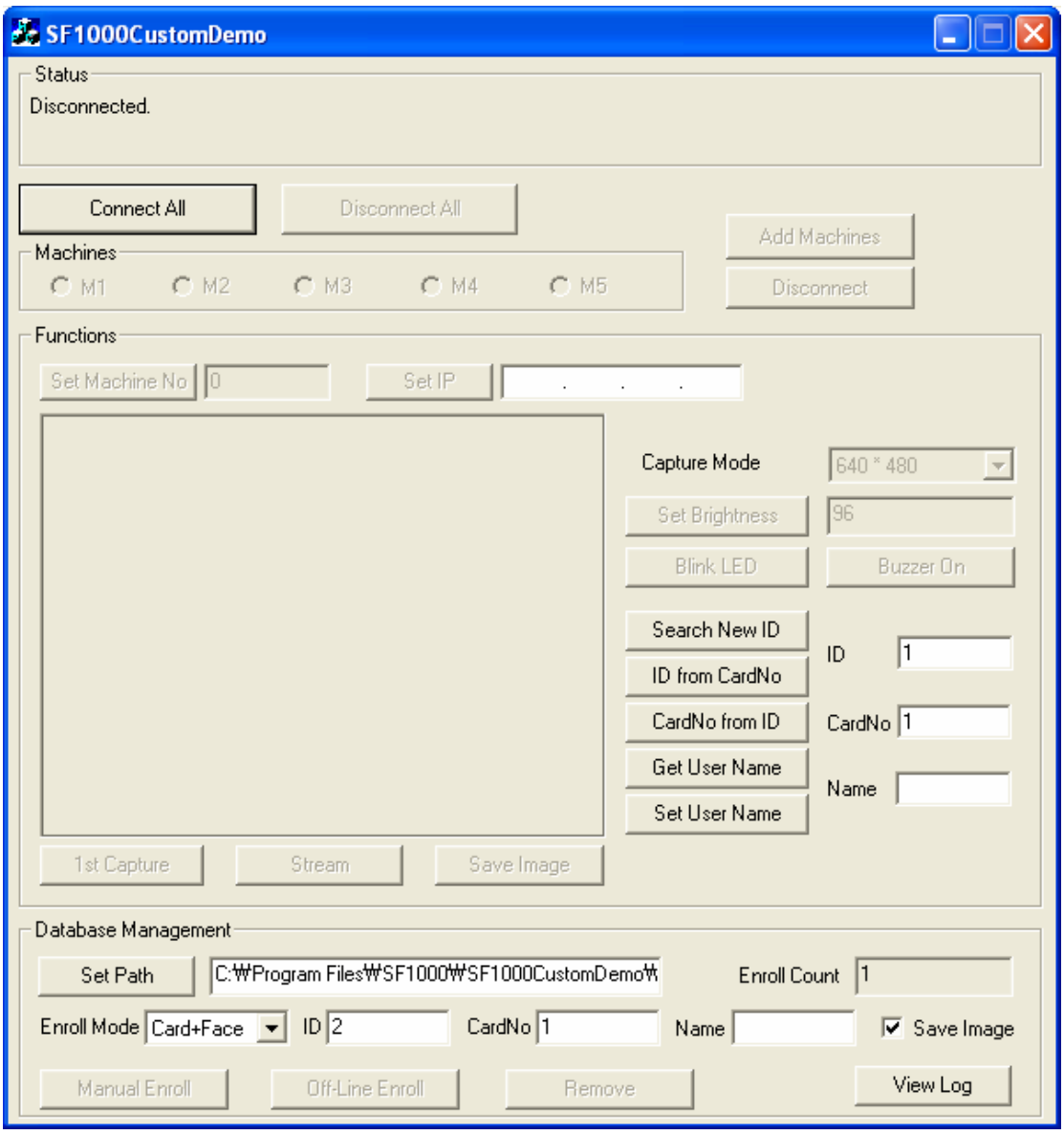

Figure 3. Demo program2 using OCX

5.3.2 Functions and use of the controls

The function and use of each control are just same as ones in the demo program1.