



# 目 录

1 简介		5
1.1 1.2 1.3	SmackFace1000 的目标 SmackFace1000 技术特征 SmackFace1000 为面部识别引擎的技术特性	5 5 5
2 Smao	ckFace1000的基本概念	7
2.1 2.2 2.3	SmackFace1000 OEM 模块的要求 2.1.1 系统要求 2.1.2 关于面部图像的建议 用户分类 (用户模式) 用户登记	7 7 8 8
3 Smao	ckFace1000 的外部结构	10
3.1 3.2 3.3	外部结构 连接计算机 标准操作.	10 11 11
4 如何	使用 SmackFace1000 OCX	13
<b>4 如何</b> 4.1	使用 SmackFace1000 OCX	<ol> <li>13</li> <li>13</li> <li>13</li> <li>13</li> </ol>
<b>4 如何</b> 4.1	使用 SmackFace1000 OCX 属性 4.1.1 SFMachineCount 4.1.2 SFVerifyLevel 4.1.3 WorkingOrgMode 4.1.4 SFDatabaseDir <sup>1</sup>	<ol> <li>13</li> <li>13</li> <li>13</li> <li>13</li> <li>14</li> </ol>
<b>4 如何</b> 4.1	使用 SmackFace1000 OCX 属性 4.1.1 SFMachineCount 4.1.2 SFVerifyLevel 4.1.3 WorkingOrgMode 4.1.4 SFDatabaseDir <sup>1</sup> 4.1.5 SFEnrollCount <sup>1</sup>	<ol> <li>13</li> <li>13</li> <li>13</li> <li>13</li> <li>14</li> <li>14</li> </ol>
<b>4 如何</b> 4.1	使用 SmackFace1000 OCX 属性 4.1.1 SFMachineCount 4.1.2 SFVerifyLevel 4.1.3 WorkingOrgMode 4.1.4 SFDatabaseDir <sup>1</sup> 4.1.5 SFEnrollCount <sup>1</sup> 4.1.6 SFManEnrollState <sup>1</sup>	<ol> <li>13</li> <li>13</li> <li>13</li> <li>13</li> <li>13</li> <li>14</li> <li>14</li> <li>14</li> </ol>
<b>4 如何</b> 4.1 4.2	使用 SmackFace1000 OCX	<ol> <li>13</li> <li>13</li> <li>13</li> <li>13</li> <li>13</li> <li>14</li> <li>14</li> <li>14</li> <li>14</li> <li>14</li> <li>14</li> <li>15</li> <li>15</li> </ol>



4.2.6 GetCommMode	15
4.2.7 GetMachineIdx	16
4.2.8 GetMachineNo	16
4.2.9 SetMachineNo	16
4.2.10 GetIPAddr	16
4.2.11 SetIPAddr	17
4.2.12 GetCaptureMode	17
4.2.13 SetCaptureMode	17
4.2.14 GetBrightness	17
4.2.15 SetBrightness	18
4.2.16 CaptureImage	18
4.2.17 GetImageData	18
4.2.18 SaveImage	18
4.2.19 Display	19
4.2.20 IsFaceImage	19
4.2.21 IsFaceImageFile	19
4.2.22 ExtractFeatureFromDev	20
4.2.23 ExtractFeatureFromFile	20
4.2.24 Match	20
4.2.25 SendWiegand	21
4.2.26 CardReaderOn	21
4.2.27 BuzzerOn	21
4.2.28 LEDCardGreenOn	21
4.2.29 LEDCardRedOn	22
4.2.30 LEDFaceGreenOn	22
4.2.31 LEDFaceRedOn	22
4.2.32 SFAction	23
4.2.22 ManEnvallStavt1	<u>0</u> 2
	20
4.2.34 ManEprollStop1	<u> </u>
4.2.34 Manifan Onstop	20
1 2 35 ManCapturol	23
	20
4.2.36 Eproll <sup>1</sup>	24
4.2.00 Eni on	24
4.2.37 OffLineEnroll <sup>1</sup>	21
	4
4.2.38 RegisterItem <sup>1</sup>	25
11-100 ICO210001100111	20



	4.2.39 Delete <sup>1</sup>	25
	4.2.40 DeleteAll <sup>1</sup>	25
	4.2.41 Verify <sup>1</sup>	25
	4.2.42 VerifyFromFile <sup>1</sup>	26
	4.2.43 SearchEmptyID <sup>1</sup>	26
	4.2.44 GetIDFromCardno <sup>1</sup>	26
	4.2.45 GetCardnoFromID <sup>1</sup>	26
	4.2.46 GetUserName <sup>1</sup>	27
	4.2.47 SetUserName <sup>1</sup>	27
	4.2.48 GetUserType <sup>1</sup>	27
	4.2.49 GetFeatureFromDB <sup>1</sup>	27
	4.2.50 SetFeatureToDB <sup>1</sup>	28
	4.2.51 GetLogCount <sup>1</sup>	28
	4.2.52 GetLogInfo <sup>1</sup>	28
	4.2.53 DeleteAllLog¹	29
4.3	事件 4.3.1 OnReceiveCardSign	29 29
	4.3.2 OnVerify <sup>1</sup>	29
5 Smac	ckFace1000 软件包	31
51	包的组成	31
5.2	演示程序 1(Visual Basic)	32



	5.2.1 界面	32
	5.2.2 控制功能	33
	5.2.3 使用	34
5.3	演示程序 2(Visual C++):	36
	5.3.1 界面	36
	5.3.2 功能和控制的使用 :	37



# 1 简介

本手册描述了 Smack Face1000, 面部识别+ ID 卡考勤机和门禁的设计特性。

# 1.1 SmackFace1000 的目标

这是一个结合了 ID 卡和面部识别技术的复杂的个人认证模块,它也是一个组成考勤系统和门禁控制系统的 0EM 模块,其中 ID 卡号和面部图像数据通过 LAN 或 USB 由 Smack Face1000 传感器传送到服务器,它们被已登记在数据库 里的数据验证。由于 ID 卡信息和个人面部信息同时使用,确保了该私人认证 系统的准确性,同时由于通过 LAN 或 USB 和电脑连接,也保证了它对各种系 统结构的灵活性。

※如果用光学的指纹传感器代替摄像头来和该模块连接,用户能够组成指纹认

证和 ID 卡识别的系统。

## 1.2 SmackFace1000 技术特征

- SmackFace1000 模块由传感器件和软件(OCX)组成。
- 传感器件由一个捕捉面部的 CCD 摄像头,一个读 ID 卡的传感电路和一个连接计算机的接口电路组成。
- OCX 最多能管理 5 个传感器件,和接口的类型(USB, LAN)无关。
- 10M/100M LAN 连接器
- USB2.0 接口
- 通过 Wiegand 输出, 一个 LED 和蜂鸣器对认证结果进行通告。
- 对用户的登记和管理功能。
- 通过摄像头对面部登记和 1:1 验证。
- 面部图像文件登记和 1:1 验证。
- ID 登记和验证。

### 1.3 SmackFace1000 为面部识别引擎的技术特性

-	每张脸的特征尺寸	2240 byte		
-	- 全部特征的提取时间 <= 260			
	◆ 面部检测时间	140ms		
	◆ 特征提取时间	120ms		
-	比对时间	<= 0.01ms		

- 出错率

水平	拒真率 (False Reject Rate, %)
1	11.90
2	14.52
3	16.07
4	20.66
5	25.31

所有上述计量的时间是基于 640\*480 的图像尺寸,用 Intel P4 2.4GHz CPU 和 512Mbytes RAM 的计算机平台测得。



# 2 SmackFace1000 的基本概念

在本文档中,用 Smack Face1000 模块开发考勤和门禁系统的编程人员 被定义为"二次开发者"。

在本文档中,由二次开发者开发的程序被安装到电脑后,有权使用该电脑 的用户被定义为"管理员"。

在本文档中,使用由二次开发者开发的程序来验证的用户被定义为"用 户"。

### 2.1 SmackFace1000 OEM 模块的要求

#### 2.1.1 系统要求

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

#### 2.1.2 关于面部图像的建议

登记注册的质量,特别是在面部识别系统中将发挥至关重要的作用。遵守下列 的注册要求,将可能得到满意的效果。

#### 注册图像的要求:

- 眼睛不被头发盖着的冷静面部
- 眼睛看着摄像头合适的位置
- 头部没有倾斜和平板的前沿姿势
- 面部在图像中完全可见并且总体上要求不能太暗或太亮
- 图像中面部的尺寸:不小于图像尺寸的 1/4

对已注册用户的即时验证,下列事项是被推荐的。

#### 验证图像建议:

#### 姿势

用正面(整个面部).

脑袋上下点头,左右旋转,左右倾斜的幅度必须小于±5 度。.

#### 距离

为了尽可能大的范围捕捉到整个前脸,摄像镜头和前脸之间 的距离必须不能太远或太近。适合于验证的距离是 30 ~ 7 0 厘米。在注册的时候,特别的距离,30~40 厘米是反射充 足特征的最佳距离。



下列表情是必须被禁止的

闭眼 眼睛被头发挡住 皱眉 暴露嘴内侧的笑脸 挑起的眉毛 把脸转离摄像头 斜视 眼镜框挡住部分眼睛

#### 光线

光线必须被均匀地分布在面部的每一侧。特别地,必需小心 避免非常的明亮或特别的黑暗和由一个高亮度聚焦光源所产 生的"热污点"。

#### 眼镜

眼镜上不该有发亮的人造物品。它的典型效果可以通过增加 照明和主体的角度,摄像头转 45 度或更多来实现。如果照 明反射不能被去除掉,那么应该把眼镜拿走。眼镜必须确保 是干净的玻璃和透明来确保眼睛和虹膜的清晰可见。重色彩 的眼镜不能被接受。

### 2.2 用户分类(用户模式)

依据卡和面部识别的结合方法,用户被分为下列两种: 用户 A: 面像+卡 用户 B: 只用卡

### 2.3 用户登记

用户信息包括卡号,面部特征和用户的模式都存在数据库中。一个用 户在数据库中只拥有一个 ID 号。卡号是记录在卡中的密码。原则上,这 唯一的号被记录在每个卡中。通过 SmackFace1000 面部识别引擎对一 幅或多幅面部图像的处理,用户的面部特征被登记到数据库中。面部登 记的原则是在实际的工作环境下登记,即模块的安装位置。该设备能够 捕捉 640\*480 和 320\*240 大小的彩色图像,但是 640\*480 大小的图 像是被推荐的。

在登记注册时,为了使面部识别引擎提取到更多的特征,距离,照 明,表情和状态保持等必须引起注意(参考 3.3 部分)。验证时用几个 面部图像,通常会得到更好的效果。标准的注册和 OCX 验证功能,使用



一个用户的三个面部图像。

和设备一起提供的 OCX 供给了作为标准的在线和离线的注册函数(参考 3.3 部分)。但是二次开发者可以用下列的 OCX 函数来定制他自己的系统。

①通过 SmackFace1000 输入用户的卡号,然后在数据库注册带有 个人信息(名字,生日,岗位等)的已输入卡号和用户的 ID。同时 也存储尚未注册的用户面部信息。

②给用户一个卡并让他到 SmackFace1000 的传感器前进行手工登 记,或者二次的开发者能够组织用户在无知觉的情况下的自动登记。 如果用户通过传感器输入卡,然后输入信号被传送到计算机。这时, 因为意味着相应的面部信息是否已存在了数据库中,给他自动登记或 存储他的面部图像,然后管理人员能够决定是否给予登记。因为用户 可能把自己的卡给别人,管理人员来验证面像和登记会更好些。



# 3 SmackFace1000 的外部结构

图 1显示 AllpaFace1000的外部结构.



图 1. SmackFace1000的外部结构.

- 3.1 外部结构
  - **电源 LED:** 这是蓝色的发光二极管用以告知设备可以工作。被连通时, 它发亮。
  - **卡 LED:** 这是双色的发光二极管,红和绿用来通知读卡和识别的情况。
  - **面部 LED:** 这是双色的发光二极管,红和绿用来通知面部识别的情况。
  - **局域网线**: 该电缆用于传感器或设备和计算机之间基于以太网协议的通讯。
  - 电源线: 该电缆用于 DC 5V (1.2A)的供电。
  - USB 线: 该电缆用于传感器设备和计算机之间基于 USB 2.0 协议的通讯。
  - Wiegand 输出: 该设备能够根据服务器的指令输出 Wiegand 信号,



该信号能实现门禁控制。

### 3.2 连接计算机

该设备能通过 10M/100M LAN 或 USB 电缆和计算机相连。它不支持 DHCP 协议。IP 地址可以通过计算机的命令来改变。改变后的 IP 地址保存在 MCU 的闪存中。通过从服务器传过来的数据,该设备可以解读服务器的 IP 地址。

当该设备通过 LAN 电缆和计算机相连时,USB 电缆不能被连上。且 当它通过 LAN 电缆连接时,DC 5V 电源必须供给该设备。本设备支持 U SB 2.0;当通过 USB 电缆连接时,它不需要直流 5V 的供电。

### 3.3 标准操作.

只有当它和已装有提供 0CX 的计算机相连时, SmcakFace1000(设备)才能操作。若已供电,该设备的电源发光二极管常亮。

当它连接到计算机且在可能操作的状态(读卡状态),所有操作提示的发光二极管(包括卡和面像的发光二极管)中,只有卡的发光二极管 (卡 LED)中红灯亮。这个状态叫*初始状态*。

标准在线登记操作如下:

- 如果在设备的初始状态有一个 enroll-start 登陆命令(OCX 函数为 ManE nrollStart)从所连接的电脑传来,该设备等待卡的签入。.
- 如果用户把卡放在部件上,该部件读卡号并把它送给计算机。
- 计算机在数据库中查找卡号。如果该号已被登陆,计算机会让设备中卡的 发光二极管(卡 LED)的红灯亮,发出一两秒的长哨声,转到初始状态,完 成了失败的操作。如果它没有登陆和有效,就让设备中卡的发光二极管(卡 LED)的绿灯亮且发出150毫秒的短哨声,并让设备的面部发光二极管(面 部 LED)的红灯闪亮。
- 重复下列的操作直到三个面像成功地被捕捉和特征被充分地被提取,或者 由 enroll-stop 命令(OCX 函数为 ManEnrollStop)来结束失败的登记。

计算机送一个 image-capture 命令(OCX 函数为 ManCapture)给设备; 如果一个图像成功地由设备转送到计算机并完成特征提取,计算机会让设备 的面像发光二极管(面部 LED)的绿灯亮,发出一个两次的短哨声。

 如果完成了三个成功的面像捕捉和特征提取,计算机把特征登记到数据库中, 让设备的面像发光二极管(面部 LED)的绿灯亮,一秒内发出一个三次的短哨 声,转到初始状态并成功地完成操作。如果登记过程在中间停止,计算机会让 设备的面像发光二极管(面部 LED)和卡的发光二极管(卡 LED)的红灯亮,一



秒内发出一个长哨声,转到初始状态并完成失败的操作。

标准验证操作如下:

- 如果用户初始把卡放在设备上,设备读卡号并把它送给计算机。
- 计算机在数据库中查找卡号。如果该号未被登记,计算机会让设备中卡的发光二极管(卡 LED)的红灯亮,发出一秒的长哨声,转到初始状态,完成了失败的操作。如果它已经登陆和有效,就让卡的发光二极管(卡 LED)的绿灯亮且发出一个150毫秒的短哨声,且让设备在面像发光二极管(面部 L ED)的红灯闪烁下捕捉图像。
- 如果图像成功地从设备被传送到计算机并使特征得到提取,计算机会在数据库中验证一个特征,如果计算的相似点大于设置水平,计算机就让设备上的面部发光二极管(面部 LED)的绿灯亮,一秒内发出三次短哨声并转到初始状态,成功地完成操作。如果特征提取失败,或计算的相似点小,计算机会让设备上的面部发光二极管(面部 LED)和卡的发光二极管(卡 LED) 红灯亮,发出一个一秒的长哨声并转到初始状态,完成失败的操作。



# 4 如何使用 SmackFace1000 OCX

SmackFace1000 OCX 的目标是为构成一个面部识别+使用 SmackFace1 000 模块的 ID 卡考勤系统提供一个软件界面。

使用这个 OCX,将会使用户有可能来构建自己的基于面像识别和 ID 卡的强有力和灵活的认证系统。

※)在下列描述的所有属性,方法和事件中,那些名字后面标有符号

"1"的,只有当"WorkingOrgMode"属性的值为1时,才有效。

### 4.1 属性

### 4.1.1 SFMachineCount

•	返回值的类型	:	LONG
•	返回值的范围	:	$0 \sim 5$
•	默认值	:	
•	读/写属性	:	读
$\bullet$	意义	:	计算目前被管理的机器数

### 4.1.2 SFVerifyLevel

•	返回值的类型	:	LONG
•	返回值的范围	:	1~~~5
•	默认值	:	2
•	读/写属性	:	读/写
•	意义	:	面像验证的安全水平(参
		考1.3	3部分)

### 4.1.3 WorkingOrgMode

•	返回值的类型		:	LONG	
•	返回值的范围	:	1-	默认模式, 0 -	
		用户模	式		
•	默认值	:	1		
•	读/写属性 :	读/写			
•	意义		:	显示 SmackF	ace1000
		是否在原	反始相	莫式监视出席和离3	干.



## 4.1.4 SFDatabaseDir<sup>1</sup>

•	返回值的类型:	String	
•	返回值的范围	:	
•	默认值	• • • • • • • • • • • • • • • • • • • •	
•	读/写属性 :	读/写	
•	意义	:	保存数据库的目录路径
4.1.5 \$	SFEnrollCount <sup>1</sup>		
•	返回值的类型:	LONG	
•	值的范围	:	
•	默认值	:	
•	读/写属性 :	读	
•	意义	:	在数据库中的用户数

### 4.1.6 SFManEnrollState<sup>1</sup>

•	返回值的类型:	LONG		
•	值的范围	:	-1 ~ 2	2
•	默认值	:		
•	读/写属性 :	读		
•	意义		:	在用户登记时表示当前的
		状态	5.	
		-1	:在用户	<sup>白</sup> 登记时无效.

0,1,2:记录有一个用户登记在进 行中和有多少次图像捕捉和特 征被成功地提取.

## 4.2 方法

### 4.2.1 ConnectAll

- ●功能 : 搜索所有有效设备并由 OCX 连接它们.
- ●样式 : ConnectAll() as Long
- ●参数
- ●返回值 : 连接设备的数目.

:



### 4.2.2 DisconnectAll

- ●功能 : 断开所有已连接的设备.
- ●样式 : DisconnectAll() as Long
- ●参数
- ●返回值 : 断开设备的数目.

:

### 4.2.3 SearchAvailableMachine

- ●功能 : 搜索有效设备.
- ●样式 : SearchAvailableMachine(CommMode as Long) as Long
- ●参数 : CommMode 设备的通讯模式被搜索. 参考 GetCommMode.
- ●返回值 : 若成功,返回被搜索设备的机器号;若失败,返回值为-1.

#### 4.2.4 ConnectMachine

●功能	:	用指定的通讯模式连接有指定机器号的设备.
●样式	:	ConnectMachine(CommMode as Long,
		M_No as Long) as Long
●参数	:	CommMode - 通讯模式.
		M_No - 被连接设备的机器号.
●返回值	:	若成功, 返回值为 0. 若失败, 返回值为-1.

### 4.2.5 DisconnectMachine

- ●功能 : 断开有指定机器号的设备.
  ●样式 : DisconnectMachine(M\_No as Long) as Long
- ●参数 : M\_No -被断开设备的机器号.
- ●返回值 : 若成功,返回值为0.若失败,返回值为-1.

### 4.2.6 GetCommMode

●功能 : 获取有指定机器号设备的通讯模式.



有两种类型的通讯模式, LAN 和 USB.

1:LAN

2: USB

- ●样式 : GetCommMode(M\_No as Long) as Long
- ●参数 : M\_No 设备的机器号.
- ●返回值 : 若成功, 返回通讯模式. 若失败, 返回 -1.

### 4.2.7 GetMachineIdx

- ●功能 : 在 OCX 获取指定设备的连接索引.
  ●样式 : GetMachineIdx(M\_No as Long) as Long
- ●参数 : M\_No 设备的机器号.
- ●返回值 : 若成功, 返回索引. 若失败, 返回-1.

### 4.2.8 GetMachineNo

●功能	: 获取指定设备的机器号.
●样式	: GetMachineNo(idx as Long) as Long
●参数	: idx - 在 OCX 中设备的连接索引.
●返回值	:若成功,返回机器号.若失败,返回 -1.

### 4.2.9 SetMachineNo

●功能	: 改变有指定机器号设备的机器号. 机器号介于
	0 到 255 之间,且不允许复制的.
●样式	: SetMachineNo(oldM_No as Long,
	newM_No as Long) as Long
●参数	: oldM_No - 设备的当前机器号.
	newM_No – 设备的新机器号.
●返回值	:若成功, 返回 0. 若失败, 返回 –1.

### 4.2.10 GetIPAddr

●功能	:	获取指定设备的 IP 地址.	
			``

- ●样式 : GetIPAddr(M\_No as Long) as String
- ●参数 : M\_No 设备的机器号.
- ●返回值 : 若成功, 返回设备的 IP 地址. 若失败, 返回 "0.0.0.0".



### 4.2.11 SetIPAddr

●功能	:	设置指定设备的 IP 地址.
●样式	:	SetIPAddr(M_No as Long, M_IPAddr as
		String) as Long
●参数	:	M_No – 设备的机器号.

- M\_IPAddr 被设置的 IP 地址.
- ●返回值 : 若成功, 返回 0. 若失败, 返回-1.

### 4.2.12 GetCaptureMode

●功能	:	获取指定设备的图像捕捉模式.
- 124 - 124		

- ●样式 : GetCaptureMode(M\_No as Long) as Long
- ●参数 : M\_No -设备的机器号.
- ●返回值 : 若成功,返回设备的图像捕捉模式(参考 SetCaptureMode).若失败,返回-1.

### 4.2.13 SetCaptureMode

●功能	: 设置指定设备的图像捕捉模式.
●样式	: SetCaptureMode(M_No as Long, mode as
	Long) as Long
●参数	: M_No -设备的机器号.
	mode - 图像捕捉模式的设定.
	1:640*480
	2:320 * 240
●返回值	:若成功, 返回 0. 若失败, 返回 -1.

## 4.2.14 GetBrightness

- ●功能 : 获取指定设备照相机的亮度.
- ●样式 : GetBrightness(M\_No as Long) as Long
- ●参数 : M\_No 设备的机器号.
- ●返回值 : 若成功,返回设备照相机的亮度.若失败,返回 -1.



### 4.2.15 SetBrightness

●功能	: 设置指定设备照相机的亮度.亮度值介于0到
	255 之间.
●样式	: SetBrightness(M_No as Long, value as
	Long) as Long
●参数	: M_No – 设备的机器号.
	value – 被设置的亮度值.
●返回值	:若成功,返回 0. 若失败,返回 -1.

### 4.2.16 CaptureImage

●功能	:	从设备捕捉面部图像数据并保存到缓冲器中.
●样式	:	CaptureImage(M_No as Long, Buffer as
		Long, Size as Long) as Long
●参数	:	M_No – 将捕捉一幅图像的设备机器号
		Buffer – 捕捉到图像的缓冲器指针. 每个点取
		三个字节(R, G, B).
		Size - 缓冲器的大小
●返回值		:若成功,返回值为0;若失败,返回值为-1.

### 4.2.17 GetImageData

●功能	:	包含由指定设备捕捉的最后一张图像的缓冲器
		指针.它用以加固 CaptureImage, Display,
		ExtractFeatureFromDev 方法的随时调用或
		是<卡+面部>验证继续进行.

- ●样式 : GetImageData(M\_No as Long) as Long
- ●参数 : M\_No 设备的机器号.
- ●返回值 : 若成功, 返回包含图像的缓冲器指针. 若失败, 返回 -1.

### 4.2.18 SaveImage

●功能	:	以"BMP"的类型保存由指定设备捕捉的最后一
		张图像(参考 GetImageData).
●样式	:	SaveImage(M_No as Long, FileName as
		String) as Long



●参数	:	M_No	- 设备的机器	<b>容号</b> .	
		FileNa	me - 保存图	像的文件	名.
●返回值	:	若成功,	返回值为0;	若失败,	返回值为-1.

## 4.2.19 Display

●功能	: 从设备捕捉面部图像并显示它.
●样式	: Display(M_No as Long, Long hDC, Long
	X0, Long Y0, Long Width, Long Height, L
	ong bNew) as Long
●参数	: M_No – 将捕捉一幅图像的设备机器号
	hDC – 设备上下处理显示捕捉到图像的窗口.
	X0 - 图像左上角的 X 坐标.
	YO - 图像左上角的 Y 坐标.
	Width -图像的宽度.
	Height -图像的高度.
	bNew -判断是否从设备中重新抓取一个新的
	图像并显示该图像(值为 0), 或显示存储的
	最后一个图像(值为1).
●返回值	:若成功,返回值为 0;若失败,返回值为-1.

## 4.2.20 IsFaceImage

●功能	:	从设备中捕捉图像数据,判断它是否是面部图
		像.
●样式	:	IsFaceImage(M_No as Long, Buffer as
		Long, Size as Long) as Long
●参数	:	M_No – 将捕捉一幅图像的设备机器号
		Buffer – 用以捕捉图像缓冲器的指针.
		Size -缓冲器的大小
●返回值	: 5	若成功,返回值为 0;若失败,返回值为−1.

## 4.2.21 IsFaceImageFile

●功能	:	判断面部图像是否是个文件.
●样式	:	IsFaceImageFile(FileName as String) as
		Long
●参数	:	FileName - 图像文件名.



●返回值 : 若成功,返回值为 0;若失败,返回值为-1.

### 4.2.22 ExtractFeatureFromDev

●功能 :	从设备中抓取图像并提取特征	征.
-------	---------------	----

- ●样式 : ExtractFeatureFromDev(M\_No as Long, Buffer as Long, Size as Long) as Long
- ●参数 : M\_No -将捕捉一幅图像的设备机器号.
   Buffer -特征缓冲器的指针,它的大小需大于 2240 bytes.
   Size -缓冲器的大小.
- ●返回值 : 若成功,返回值为 0;若失败,返回值小于 0.

### 4.2.23 ExtractFeatureFromFile

●功能	:	从面部图像文件中提取特征.
●样式	:	ExtractFeatureFromFile(FileName as
		String, Buffer as Long, Size as Long) as
		Long
●参数	:	FileName – 图像文件路径
		Buffer - 特征缓冲器的指针,它的大小需大于
2240 bytes.		
		Size -缓冲器的大小
●返回值	: ;	若成功,返回值为 0;若失败,返回值小于 0.

### 4.2.24 Match

●功能	: 比较两个特征并判断它们是否相同.
●样式	: Match(Buffer1 as Long, Size1 as Long,
	Buffer2 as Long, Size2 as Long) as long
●参数	: Buffer1 - 特征1缓冲器的指针.
	Size1 - 缓冲器 1 的大小
	Buffer2 - 特征2缓冲器的指针.
	Size2 - 缓冲器 2 的大小
	缓冲器 1 和缓冲器 2 的大小一定要大于 2240
	bytes.
●返回值	:在两个特征的相似度大于设定阀值水平(SFVer

fyLevel)的情况下,返回值为1;否则为0.

20



### 4.2.25 SendWiegand

●功能 : 让设备输出一个韦根信	夛.
------------------	----

- ●样式 : SendWiegand(long M\_No, long ID) as Long
- ●参数 : M\_No 设备的机器号. ID - 用户的 ID.
- ●返回值 : 若成功, 返回值为 0; 若失败, 返回值为-1.

#### 4.2.26 CardReaderOn

●功能	: 打开/关闭设备的读卡器模式.
●样式	: CardReaderOn(M_No as Long, On as Lo
	ng) as Long
●参数	: M_No – 设备的机器号.
	On – 若值为 0, 关闭模式.否则为打开.
▲近回佐	• 龙虎马 近回传头 0 龙虎啦 近回传头 1

### ●返回值 : 若成功,返回值为 0;若失败,返回值为-1.

### 4.2.27 BuzzerOn

●功能	:	打开/关闭设备的蜂鸣器.
●样式	:	BuzzerOn(long M_No, long On, long Peri
		od) as Long

- ●参数 : M\_No 设备的机器号. On - 若值为 0,关闭蜂鸣器.否则为打开. Period - 当蜂鸣器被打开时,它决定蜂鸣器 响多少次.当它的值为 0,意味着响多少次无 限制.时间单元是 50 微秒.
- ●返回值 : 若成功,返回值为 0;若失败,返回值为-1.

### 4.2.28 LEDCardGreenOn

●功能	:	设备的打开/关闭或是卡验证 LED 的绿 LED
		闪烁.
●样式	:	LEDCardGreenOn(M_No as Long, Option
		as Long) as Long
●参数	:	M_No – 设备的机器号.



Option - 值为 0 意味着关闭. 值为 1 意味着 打开, 值为 2 意味着闪烁.

●返回值 : 若成功,返回值为 0;若失败,返回值为-1.

## 4.2.29 LEDCardRedOn

●功能	: 设备的打开/关闭或是卡验证 LED 的红 LED
	闪烁.
●样式	: LEDCardRedOn(M_No as Long, Option as
	Long) as Long
●参数	: M_No – 设备的机器号.
	Option – 值为 0 意味着关闭. 值为 1 意味着
	打开,值为2意味着闪烁.
●返回值	:若成功,返回值为 0;若失败,返回值为-1.

### 4.2.30 LEDFaceGreenOn

●功能	: 设备的打开/关闭或是面部识别 LED 的绿 LE	Ď
	闪烁.	
●样式	: LEDFaceGreenOn(M_No as Long, Option	1
	as Long) as Long	
●参数	: M_No - 设备的机器号.	
	Option - 值为0意味着关闭. 值为1意味着	
	打开,值为2意味着闪烁.	
●返回值	:若成功,返回值为 0;若失败,返回值为-1.	

### 4.2.31 LEDFaceRedOn

●功能	: 设备的打开/关闭或是面部识别 LED 的红 LED
	闪烁.
●样式	: LEDFaceRedOn(M_No as Long, Option as
	Long) as Long
●参数	: M_No – 设备的机器号.
	Option −值为 0 意味着关闭. 值为 1 意味着打
	开,值为2意味着闪烁.
●返回值	:若成功,返回值为 0;若失败,返回值为-1.



### 4.2.32 SFAction

●功能	: 用蜂鸣器和 LED 让设备执行预定义信号的动
	作.
●样式	: SFAction(long M_No, long ActionNo) as
	Long
●参数	: M_No – 设备的机器号.
	ActionNo – 值为 94, 意味着设备发出一声短
	哨声(350ms)两次; 值为 95, 意味着设备发
	出一声短哨声(350ms)三次.

### ●返回值 : 若成功, 返回值为 0; 若失败, 返回值为-1.

### 4.2.33 ManEnrollStart<sup>1</sup>

●功能	: 开启手动登记.
●样式	: ManEnrollStart(ID as Long, name as Strin
	g, mode as Long, bSave as Long) as Long
●参数	: ID - 登记的用户 ID
	name – 登记的用户名字. (最大为 20 bytes)
	mode – 验证模式
	bSave – 决定图像数据是否被存为一个文
	件,若是,图像数据将以"xxxxxx_y.bmp"
	的名字存储在数据库目录的"img"子目录中.
	这里"xxxxxx"指的是 ID 值,"y"指的是三
	张用于登记图像中的图像号码.
●返回值	:若成功,返回 0. 若失败,返回 -1.

## 4.2.34 ManEnrollStop<sup>1</sup>

●功能 :	停止手动登记的进程.
-------	------------

:

- ●样式 : ManEnrollStop() as Long
- ●参数
- ●返回值 : 若成功, 返回 0. 若失败, 返回 -1.

### 4.2.35 ManCapture<sup>1</sup>

●功能 : 在手动登记时,从设备捕捉图像和提取特征.该功能需在手动登记被启动和从设备签入卡信



	号后才能使用. 无论何时一个成功的访问完成,
	SFManEnrollState 的值会被更新.在三次成
	功地调用 ManCapture 后,OCX 登记用户信
	息和面部特征到数据库并成功地完成手动注册.
●样式	: ManCapture() as Long
●参数	:
●返回值	:若成功,返回 0. 若失败,返回 –1.
4.2.36 Enroll <sup>1</sup>	
●功能	: 从设备中读取 ID 卡,并用输入的验证模式在
	输入的 ID 卡上登记用户,在《ID 卡+ 面像》
	验证的模式下,也从设备或一个文件里读取面
	部图像,提取特征并登记.
<b>-</b> 124 - D	

- •样式 : Enroll(ID as Long, name as String, mode as Long, bSave as Long) as Long
- ●参数 : ID -登记用户的 ID name -登记用户的姓名(最大 20 字节) mode -验证模式 bSave -判断图像文件数据是否要存储成一个 文件.参考 ManEnrollStart.
- ●返回值 : 若成功, 返回值大于 0. 若失败, 返回值为-1.

### 4.2.37 OffLineEnroll<sup>1</sup>

●功能	:	用 ID,卡号和验证模式来登记。在〈ID+面
		像〉验证模式的情况下,从文件里读取图像数
		据,提取特征并进行登记.
●样式	:	OffLineEnroll(ID as Long, CardNo as Lon
		g, name as String, mode as Long,
		filenames as Variant, bSave as Long) as
		Long
●参数	:	ID – 登记用户的 ID
		CardNo - 登记用户的卡号
		name – 登记用户的姓名(最大 20 字节)
		mode – 采用用户的验证模式
		filenames - 面部图像文件的路径。当模式值
		为 3, 即 〈ID+ 面像〉模式时, 它是有效的.



bSave -判断是否存储该图像数据为一个文件. 参考 ManEnrollStart

●返回值 : 若成功, 返回值大于 0; 若失败, 返回值为-1.

### 4.2.38 RegisterItem<sup>1</sup>

●功能	: 在数据库里登记卡号.
●样式	: RegisterItem(ID as Long, name as String,
	CardNo as Long, mode as Long) as Long
●参数	: ID - 登记的用户 ID
	name – 登记的用户姓名
	CardNo - 登记的用户卡号
	Mode – 用户的验证模式.
	参考 GetUserType
●汳□佶	· 芋戓功 近回值为 0. 芋牛叻 近回值小王-1

### ●返回值 : 右成切,返回值为 0; 右矢败,返回值小于-1.

### 4.2.39 Delete<sup>1</sup>

●功能	:	删除数据库中的 ID 和卡号.
●样式	:	Delete(ID as Long, CardNo as Long) as
		Long
●参数	:	ID -用户在数据库里登记的 ID.
		CardNo -在 ID 里的卡号.
●返回值	:	若成功,返回值为0;若失败,返回值为-1.

### 4.2.40 DeleteAll<sup>1</sup>

●功能	:	删除数据库中的所有数据.
DV B		

●样式 : DeleteAll() as Long

:

- ●参数
- ●返回值 : 若成功,返回值为0;若失败.返回值为-1.

### 4.2.41 Verify<sup>1</sup>

●功能	:	从设备中抓取图像数据,提取特征点,并用数
		据库里的特征点一个一个验证.
●样式	:	Verify(M_No as Long, ID as Long, CardNo

F式 : Verify(M\_No as Long, ID as Long, CardNo as Long, TryCount as Long) as Long



●参数	:	M_No	- 将捕捉图僔	象的设备标	机器号
		ID - 张	验证用户的 ID		
		CardN	0-卡号		
		TryCo	unt – 失败时	重新计数	ξ.
●返回值	: 3	若成功,	返回值为1;	若失败,	返回值为 0.

### 4.2.42 VerifyFromFile<sup>1</sup>

●功能	: 从面部图像文件中提取特征,用数据库里的特
	征一个一个验证.
●样式	: VerifyFromFile(ID as Long, CardNo as Lo
	ng, FileName as String) as Long
●参数	: ID - 验证用户的 ID.
	CardNo - 卡号
	FileName – 面部图像文件路径
●返回值	:若成功,返回值为 1;若失败,返回值为 0.

### 4.2.43 SearchEmptyID<sup>1</sup>

●功能	:	在数据库中搜索一个新的 ID 来登记.
●样式	:	SearchEmptyID() as Long
●参数	:	
●返回值	:	ID 值

### 4.2.44 GetIDFromCardno<sup>1</sup>

●功能 :		获取通过搜索数据库后登记的指定卡号的 ID.
-------	--	------------------------

- ●样式 : GetIDFromCardno(CardNo as Long) as Long
- ●参数 : CardNo 搜索的卡号
- ●返回值 : 若卡号已登记,则返回值为 ID; 否则返回值为 0.

### 4.2.45 GetCardnoFromID<sup>1</sup>

●功能	:	获取通过搜索数据库后登记的指定 ID 的卡号.
●样式	:	GetCardnoFromID(ID as Long) as Long

●参数 : ID - 数据库中已登记用户的 ID



●返回值 :若已登记,则返回卡号;否则返回值为0.

### 4.2.46 GetUserName<sup>1</sup>

- ●功能 : 获取用户的名字.
- ●样式 : GetUserName(ID as Long) as String
- ●参数 : ID 数据库中已登记用户的 ID
- ●返回值 : 用户的名字

### 4.2.47 SetUserName<sup>1</sup>

- ●功能 : 设置用户的名字.
- •样式 : SetUserName(ID as Long, name as String) as Long
- ●参数 : ID 数据库中已登记用户的 ID name 被设置的用户名字.
- ●返回值 : 若成功,返回值为 0;若失败,返回值为-1.

### 4.2.48 GetUserType<sup>1</sup>

●功能	:	获取用户的验证模式,验证模式有两种类型:
		《ID 卡+ 面像》模式和《ID 卡》模式.
●样式	:	GetUserType(ID as Long) as Long
●参数	:	ID – 数据库中己登记用户的 ID

●返回值 : 若返回值为 1, <ID 卡>模式; 若返回值为 3,
 <ID 卡 + 面像>模式.

### 4.2.49 GetFeatureFromDB<sup>1</sup>

- ●功能 : 从数据库里获得与 ID 和卡号相应的面部特征.
- ●样式 : GetFeatureFromDB(ID as Long, CardNo as Long, Buffer as Long, Size as Long) as Long
- 参数 : ID 获取特征的 ID. CardNo - 卡号. Buffer - 登记数据缓冲器的指针. 它的大小一定要大于 2240 bytes. Size - 缓冲器的大小.
   美成功,近回值为 0. 美生败,近回值小司
- ●返回值 : 若成功,返回值为 0;若失败,返回值小于 0.



## 4.2.50 SetFeatureToDB<sup>1</sup>

●功能 :		数据库中设置有关 ID 和卡号的特征.
-------	--	---------------------

●样式	:	SetFeatureFromDB(ID as Long, CardNo
		as Long,Buffer as Long, Size as Long) as
		Long

●参数	:	ID – 设置用户的 ID
		CardNo - 卡号
		Buffer - 特征缓冲器的指针.
		Size – 缓冲器的大小
●返回值	:	若成功,返回值为0;若失败,返回值小于0.

### 4.2.51 GetLogCount<sup>1</sup>

●功能	: 获取数据库中记录数据数.
●样式	: GetLogCount() as Long
●参数	:
●返回值	:若成功,返回值为计数;若失败,返回值为-1.

## 4.2.52 GetLogInfo<sup>1</sup>

●切肥 · 获取级据件中一个吃水级据	
●样式 : GetLogInfo(no as Long, pLogInfo as	s Lon
g) as Long	
●参数 : no - 获取记录数据号.	
pLogInfo - 记录数据号缓冲器的指针.	
●返回值 : 记录数据号	

## 记录数据类型如下.

### Type LOGITEM

Id As Long	'记录数据的用户 ID
CardNo As Long	'用户的卡号
mode As Long	'验证模式
vYear As Long	'年
vMonth As Long	'月
vDate As Long	"日
vHour As Long	'小时



vMin As Long
vSec As Long
wTime As Long
Ok As Long
tryCount As Long
M_No As Long
End Type

'分 '和

'秒
'响应时间(ms)
'验证结果(成功-1, 失败-0)
'尝试验证数
'已用设备的机器数

### 4.2.53 DeleteAllLog<sup>1</sup>

- ●功能 : 删除数据库中所有的记录数据.●样式 : DeleteAllLog() as Long
- ●参数 :
- ●返回值 :若成功,返回值为0;若失败,返回值 is -1.

## 4.3 事件

### 4.3.1 OnReceiveCardSign

●功能	:	从设备中读卡时,告知卡号.
●样式	:	OnReceiveCardSign(M_No as Long,
		CardNo as Long)
2. SH		

●参数 : M\_No - 读卡签入的设备机器号.
 CardNo - 被读的卡号

### 4.3.2 OnVerify<sup>1</sup>

若WorkingOrgMode = 1条件下,验证后通
知验证结果. 在启动手动登记后, 卡签入被输
入的情况下,它也告知卡号有无可能被登记.

- ●样式 : OnVerify(M\_No as Long, ID as Long, Result as Long)
- ●参数 : M\_No -已用设备的机器号.
  - ID 验证的 ID.

Result - 结果

在验证时.

- -1:失败.
- 1: <ID 卡 + 面像> 验证 成功



2: <ID 卡>验证 成功.

在手动登记时.

- -1: 输入的卡号不能被登记.
- 1: <ID 卡 + 面像 > 登记许可.
- 2: <ID 卡> 登记成功.



# 5 SmackFace1000 软件包

## 5.1 包的组成

SmackFace1000 软件包由 SF1000PC OCX, SmackFace1000 设备的 USB2.0 驱动程序,两个由 Visual Basic 和 Visual C++编写的源程序和用户 手册组成.

每个组件被放在下列的路径.

- **[SysDir]₩** SF1000PC OCX.
- [AppDir]₩ 用户手册.
- [AppDir]₩Driver SmackFace1000 设备的 USB2.0 驱动程序.
- [AppDir]₩VBSample 用 Visual Basic 编写的源程序.
- [AppDir]₩SF1000CustomDemo 用 Visual C++编写的源程序.

※)这里, [SysDir]是 Windows 系统的系统目录, [AppDir]是软件包的 设置目录.



## 5.2 演示程序 1(Visual Basic)

该演示程序用标准操作模式,使用 Visual Basic (属性 WorkingOrgMode = 1)编写.

它演示了数据管理,用户登记,验证和用 OCX 标准函数的结果告知等;相对于演示程序 2,该程序要简单一些.

#### 5.2.1 界面

图 2 演示程序 1 的界面显示.

🖻 SF1000 Demo
1个设备已被连接。
<b>今</b> 郭连接 <b>今</b> 郭折开
传感器议备
○1 C0 C0 C0 C0 拆开
设置设备号 1 设置IP地址 255.255.255.255
<b>抓取图像模式</b> 640 × 480 ▼
没罢去度 [96
LED闪烁 打开蜂鸣器
搜索新的ID
从卡号中获取ID <sup>ID</sup> 1
从ID中获取卡号 卡号
第一次抓取 动态像 保存图像 不能够原创模式
必要の A C\Program Files\CE1000\\/PCample\DD\
模式     ★     ID 3     卡号     姓名     ☑ 保存图像
手动登记 自动登记 脱机登记 移除 查看记录

图 2. 使用 OCX 的演示程序

在"状态"部分, 主要是关于设备的连接和断开, 卡的输入, 验证结



果和注册或取消用户的显示信息.

在"**传感器设备**"部分,显示设备和 OCX 的连接以及用户能够选择 他想要控制的设备.

在"功能"部分,是和设备关联的 OCX 函数的执行控制.

在"**数据库管理**"部分,是和数据库管理关联的 OCX 函数的执行控制.

5.2.2 控制功能.

"全部连接"键--把所有可用的设备和 OCX 相联.

"全部拆开"键---断开所有设备和 OCX 的连接.

"拆开"键---断开所选设备和 OCX 的连接.

"添加传感器设备"键一找到一个可用设备并把它和 OCX 连接.

"设置设备号"键---设置设备的机器号.

"设置 IP 地址"键一设置设备的 IP 地址.

"抓取图像模式"组合框--设置设备的图像捕捉模式.

"设置亮度"键一设置设备中照相机的亮度.

"LED 闪烁"键一让设备中面像发光二极管(Face LED)的红灯 闪烁,再按一次则停止闪烁.

"打开蜂鸣器"键一让设备的蜂鸣器发声.再按一次则发声停止.

"搜索新的 ID"键--搜索能被注册的最小 ID.

"从卡号中获取 ID"键一搜索注册的指定卡号的 ID.

"从 ID 中获取卡号"键一搜索指定 ID 已注册的卡号.

"获取用户姓名"键一搜索指定 ID 已注册的用户名字.

"设置用户姓名"键一设置指定 ID 已注册的用户名字.

"不能够原创模式"键--使能或不能启动标准严整功能.

"第一次抓取"键---手动登记时,从设备捕捉一幅图像并提取特征.

"动态像"键--从设备捕捉图像并连续显示它们. 再按一次则停止 捕捉.

"保存图像"键一从设备捕捉一幅并把它存到指定的文件名.

"设置路径"键---设置 OCX 数据库的路径.

"模式"组合框--在用户登记时设置验证模式.

"保存图像"组合框--判定图像在登记时是否被保存为文件.

"手动登记"键--启动或停止手动登记.

"自动登记"键--启动自动登记.

"脱机登记"键--启动脱机登记.

"移除"键--从数据库中将指定 ID 的注册用户注销.

"查看记录"键--数据库中存储的记录.



#### 5.2.3 使用.

#### - 连接和断开设备.

依据通讯模式把设备和计算机由 LAN 或 USB 等电缆连接起来. 给设备上电,电源的发光二极管亮,然后设备发出三次的短哨声. 按下"**全部连接**"键,所有可用的设备都和 OCX 相连,发出二次的 短哨声.

当有一个设备后来增加时,先用电缆把设备和计算机相连,上电, 然后按下"添加传感器设备"键,这个后加的设备就联上了。一个 OCX 最多能够管理 5 个设备.

按下"**全部拆开**"键,所有连接的设备都被断开,并发出二次的短 哨声.与此不同,"**拆开**"键只断开所选择的机器.

#### - 注册和注销用户.

一个用户的三个面部图像用于注册,"**数据库管理**"部分的按键用 于实现这部分功能.

#### •手动注册

首先,输入 ID (要求,不重复的)和名字 (若需要的话)登记在 ID 编辑栏和名字编辑栏.

按下"**手动登记**"键. 然后该键的字幕变为"**停止登记**"这意味着随时可以用它来终止注册过程.把卡放在设备上然后被捕捉的图像连续显示. 当完整良好的图像被捕捉时,按下"**第一次抓取**"键. 若第一幅图像注册成功,该键的字幕变为"**第二次抓取**"并可以启动第二幅图像的注册.(如果由图像的特征提取失败,该键的字幕不变.) 如果三幅图像注册成功地经由"**第一次抓取**"键变为"**第二次抓取**"和"**第三次抓取**"键,则表明用户注册已成功完成.

#### •自动注册

和手动注册一样, 输入 ID 和名字.

按下"自动登记"键. 若在 5 秒内用户把卡放在设备上,从那时起,OCX 以一定的时间间隔(大约 0.5 秒)连续捕捉三张图像并注册它们.

#### •脱机注册

和手动注册一样,输入 ID 和名字. 因为是脱机注册,用户必须在**卡号**编辑栏手动地输入一个卡号以 代替从设备输入.(无论何时,当一个人把卡防在设备上时,卡号都



会被显示在**卡号**编辑栏,这样它就可用于随后的脱机注册.) 按下"**脱机登记**"键后,一个文件对话框被打开.选择一个图像文件 并按下"**打开**"键,一个文件对话框被打开.若以这种方式三个文件 被选择,注册已完成.

在脱机注册前,将被用来注册的面部图像可以用这个程序被捕捉和 保存.在**功能部分**按下"**动态像**"键,图像被连续显示.按下"**保存图** 像"键,除了它,还有一个文件保存对话框被打开,此时捕捉的图 像可以被存到指定的文件名中.按下"**停止**"键,则 改变"**动态像**" 键的功能,图像捕捉被终止.

#### •下列情况下注册会失败.

当由图像提取特征失败时

当指定 ID 或卡号已经注册,不同的卡号必须被输入到 ID 编辑栏. 无论何时它和设备相连或是注册成功,该演示程序由自己搜索一 个可用的 ID 并把它输入到 ID 编辑栏.用**功能部分**的"**搜索新的** ID"键能够找到可用的 ID.

当用户没有在某一时间内把卡放在设备上(自动注册时)

#### - 验证.

当用户把卡放在设备上, OCX 读卡号并检查它是否是已注册的卡. 若已注册, OCX 从设备捕捉图像, 提取特征, 和注册的特征做验证 并以信息的方式告知结果. 成功时, 两个发光二极管的绿灯亮并发 出三次的短哨声. 失败时, 两个发光二极管的红灯亮并发出一声长 哨声. 结果显示在**状态部分**.

在**功能部分**按下"**不能够原创模式**"键可以使 OCX 终止标准验证 功能.此时,该键的字幕改变为"能够原创模式"且 OCX 只告知卡 号而无验证.再次按下该键则标准验证功能打开.当用户要用自己 定制设计的验证时,该功能非常有用.

#### - 其它.

在**功能部分** OCX 的其它函数可以由用以控制来使用.此时,在设置 它的机器号, IP 地址,图像捕捉模式,照相机的亮度等之前,被控制 设备必须在**传感器设备部分**被正确地选择.

# 5.3 演示程序 2(Visual C++)

该演示程序不使用标准操作模式且它是由 Visual C++编写的(属性 WorkingOrgMode = 0).

它执行了和标准操作模式下一样的功能且描述了用户定义数据管理,用户登记,验证和按用户自己的方式提示结果和使用它们.

### 5.3.1 界面

图 3 显示演示程序 2 的界面.

💑 SF1000CustomDemo		
─状态 1个设备已被连接.		
全部连接     全部拆开       传感器设备     ● 1     ● M2     ● M3     ● M4     ● M5	添加传感器设备 拆 开	
切能 设置设备号 1 设置IP地址 255 . 255 . 255 . 255		
	抓取图像模     640×480       设置亮度     96       LED闪烁     打开蜂鸣器       搜索新的ID     打开蜂鸣器       从卡号中获取ID     1       从ID中获取卡号     卡号       获取用户姓名     姓名	
第一次抓取 动态像 保存图像		
数据库管理 设置路径 E:\SF1000CustomDemo\CustomDB\	登记数 2	
	姓名     ▶     ✔     保存图像       余     查看记录	

图 3. 使用 OCX 的演示程序 2



## 5.3.2 功能和控制的使用

功能和每个控制的使用和演示程序1的一样.