ZKFinger SDK 程序开发手册

ZKFinger SDK
中控科技

ZKFinger SDK 版权声明

使用单位注册

在您过买了本产品后，请认真填写《指纹 SDK 系统用户注册登记表》中如下信息，然后传真或者 EMAIL 至我公司，您就成为本产品的合法用户，可以获得我们全技术支持服务和软件版本升级信息了！

《指纹 SDK 系统用户注册登记表》

1、您购买软件的　日期：____/__/__

地点：_____

_____

经销商：_____

_____

2、您的姓名：_____

职务：_____

称谓：[ ]先生 [ ]小姐

联系电话：_____

电子邮件：_____

联系地址：_____

3、您单位的名称：_____

_____

单位的简称：_____

地址：_____省/市_____市/区

_____

邮政编码：_____

中控科技 http://www.zkteco.com

单位电话：(____)_____

单位人数：[ ] <100 人

[ ] 101 到 200 人

[ ] 201 到 500 人

[ ] 501 到 1000 人

[ ] 1000 人以上

互联网站：_____

电子邮件：_____

4、您是否愿意接收我们的

产品升级通知？ [ ]是

新产品广告？ [ ]是

技术信息快报？ [ ]是

网站更新通知？ [ ]是

请传真：0755-89602194

Email 至：support@biometric.com.cn

目　　录

1. ZKFinger 算法描述

中控科技一直专注于指纹识别算法的研究和产业化推广，已将指纹识别系统应用到各种行业中。随着指纹识别系统越来越广泛的应用，市场对指纹识别算法的精确性，适用性和运算速度等多方面提出更高的要求。为满足这些需求，我们从低质量指纹图像的增强、指纹的特征提取、指纹图像的分类与检索及压缩技术、指纹图像匹配算法等多方面进行优化，推出 ZKFinger10.0 版高速算法。该算法在大规模的数据库上进行了严格的测试，误识率（False Accept Rate, FAR）、拒识率（False Reject Rate, FRR）、拒登率（Error Registration Rate, ERR）等性能都大大提高，对过干、太湿、伤疤、脱皮等低质量的指纹图像处理效果明显增强，算法比对速度提升了 10 倍以上。该算法的指纹模板也同时进行了优化存储，与以前算法版本的指纹模板不兼容。您选择 ZKFinger 10.0 版高速算法后，必须重新登记用户指纹模板，

ZKFinger 算法是一种快速、准确的 1:1 和 1:N 指纹识别算法，面向软件开发商和系统集成商全面开放，客户可以根据需要选择使用 ZKFinger10.0/9.0 算法引擎；在使用 ZKFinger 10.0 引擎进行指纹识别时，不需要对指纹通过姓名、PIN 等预先分类就可以达到 500000 枚/秒(以下测试都在 CPU2.6GHz+ 1GMB 内存环境下进行)。使用 ZKFinger9.0 算法引擎，在 PC 上可以达到 6000 枚/秒；ZKFinger 算法具有以下特点：

1、 ZKFinger 软件开发包能够快速集成到客户系统中，通过开放图像处理接口，可以支持任何扫描设备和指纹 Sensor(图像质量>=300DPI)。

2、 ZKFinger 算法通过自适应的、适合匹配的滤镜和恰当的阀值，减弱噪音，增强脊和谷的对比度，甚至能够从质量很差的指纹(脏、刀伤、疤、痕、干燥、湿润或撕破)中获取适当的全局和局部特征点。

3、 ZKFinger9.0 算法引擎比对时支持指纹平移(>=指纹面积 35%)和 360 度旋转。通过使用特殊技术实现在指纹平移和 360 度旋转时的快速比对（平均速度 3000 枚/秒），即使指纹特征点很少时(<=10，一般手指的特征点)=15)，也可以实现上述功能。

ZKFingr10.0 算法引擎为速度考虑，目前不支持 360 度旋转比对功能，但支持±30 度旋转比对。

4、 ZKFinger 算法不需要指纹必须有全局特征点(核心点、三角点等)，通过局部特征点就可以完成识别。

5、 ZKFinger 通过分类算法(指纹被分成五大类型：拱类、左环类、右环类、尖拱类、旋涡类 “斗”), 预先使用全局特征排序，从而大大的加速指纹匹配过程。

6、 ZKFinger9.0 算法引擎代码相当简洁，数据空间仅需要 350K 内存，因此可以容易的移植到嵌入式系统中。

通过从四种 Sensor(YLC,DFR200,U.ARE.U,Authentec)采集到 2000 枚指纹 ZKFinger 做测试(每种 Sensor 采集 500 枚)， 每枚指纹和其它全部 2000 枚指纹做比对，共进行 4,000,000 次比对，得到下面的测试比对结果：

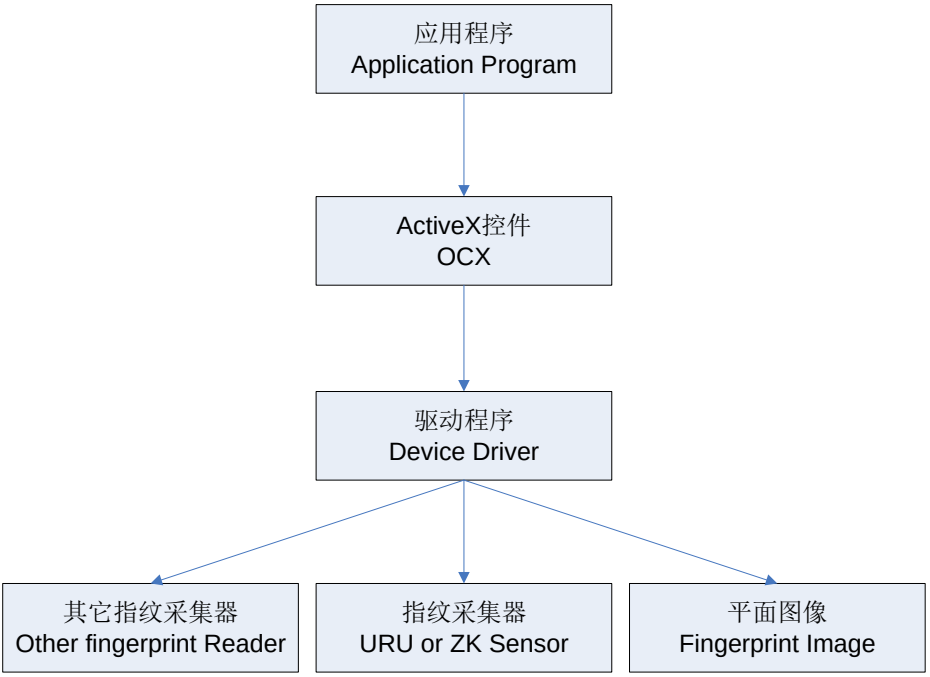
模板大小	310 or 1152 Byte
旋转	0 – 360 度
FAR	<= 0.001%
FRR	<= 2.0%
登记时间	0.5 秒
平均比对速度	2500 枚/秒
图像质量	>=300DPI

ZKFinger10.0 算法引擎需要 2M 以上内存，模板大小在 2k 左右。

2. ZKFinger SDK 架构

ZKFinger SDK(Software Development Kit)主要以 ActiveX 的方式存在，使用者可以使用各种开发语言（例如 VC++, C++Builder, Delphi, VB, Visual Foxpro, PB,C#,VB.net 等）来开发相对于指纹仪的应用程序。

SDK Architecture



3. 软件安装

在安装 ZKFinger SDK 之前，请确定您的操作系统和计算机的配置符合软件运行的要求。

在进行安装之前，如果您的计算机接上了指纹采集器，最好将它拔下来。

1、运行 setup.exe。按照提示点下一步可以完成安装指纹仪驱动。

4.ActiveX 控件参考

ZKFinger SDK 同时具备 1:1 和 1:N 两种功能，以下都采用 VB 语法说明，其中指纹模板 Variant 变量代表一维字节数组。

4.1 属性

4.1.1 Active as Boolean

Read only

当前 SensorIndex 设定的指纹采集器是否准备就绪。

4.1.2 EngineValid as Boolean

Read only

指纹识别系统是否正常工作。调用过函数 initEngine 后会返回有效结果

4.1.3 EnrollIndex As Long

Read only

登记指纹时取样的序号，即表示当前手指登记已经取到的有效次数。

4.1.4 EnrollCount As Long

登记指纹时取样的次数，取值范围为 1、3、4 次。

4.1.5 FPEngineVersion AS String

指纹识别系统算法引擎版本号，默认为 ZKFinger9.0 算法引擎，其值为“9”，若使用 10.0 算法引擎，则在调用其他函数之前首先将该属性设置为“10”，然后调用 InitEngine、CreateFPDBCache 等函数。

4.1.6 ImageHeight AS integer

Read only

指纹图像的高度

4.1.7 ImageWidth AS integer

Read only

指纹图像的宽度

4.1.8 IsRegister As Boolean

Read only

是否正登记指纹

4.1.9 OneToOneThreshold As Boolean

设定 ZKFinger 低速指纹 1：1 比对的识别阀值分数(1-100)，默认为 10，值越大，误判率越低同时拒绝率变大

4.1.10 RegTplFileName As String

设置当事件 OnEnrollToFile 发生时，保存指纹登记模版的文件名称。

4.1.11 SensorCount As Long

Read only

当前连接到计算记得指纹采集器的个数，当 EngineValid 无效时，返回 0

4.1.12 SensorIndex AS Long

连接多个指纹采集器时，选择指纹头的序号，从 0 开始；小于零时指纹采集器不工作。

4.1.13 SensorSN As String

指纹采集器的硬件序列号

4.1.14 TemplateLen As Long

Read only

指纹登记模版的最大字节长度。

注意：ZKFinger 9.0 引擎模板最大长度为 1152 字节，ZKFinger 10.0 算法引擎为 3072；

4.1.15 Threshold As Long

设定指纹识别系统比对识别阀值分数(1-100)，默认为 10，值越大，误判率越低同时拒绝率变大

4.1.16 VerTplFileName As String

设置当事件 OnCaptureToFile 发生时，保存指纹验证模版的文件名称。

4.1.17 LastQuality As Long

Read only

最新一次指纹的质量，在事件 OnFeatureInfo 可以获取。当 LastQuality 小于 LowestQuality 时，指纹质量不合格；-1 表示可疑指纹(假手指)。

4.1.18 LowestQuality As Long

设置允许最低指纹质量，默认值为 60，当 LastQuality 小于 LowestQuality 时，指纹质量不合格。

4.1.19 FakeFunOn As Long

设置防假开关，默认值为 1。1 表示开启，0 表示关闭。

4.2 方法

4.2.1 Sub BeginEnroll()

开始登记指纹，登记结束后发生 OnEnroll 事件。

4.2.2 Sub CancelEnroll()

取消当前的指纹登记状态，即由 BeginEnroll 开始的操作可由此函数中断。

4.2.3 Function DongleIsExist As Boolean

检查加密狗是否存在

*4.0 以后版本已取消此函数的功能，为了兼容，暂保留了此函数。

4.2.4 Function DongleSeed(Byval lp2 As Long, Byval p1, p2, p3, p4 As Integer) As Boolean

得到种子码 lp2 的四个 16 位整数(p1,p2,p3,p4)返回值，加密狗可以通过内部算法计算一个种子码，得到四个返回码。种子码算法是不公开的，可以通过检查返回码是否是期望的值来检查加密狗是否存在。

*4.0 以后版本已取消此函数的功能，为了兼容，暂保留了此函数

4.2.5 Function DongleUserID As Long

读出加密狗中的用户 ID，用户 ID 不会重复相同。保存在加密狗内部特定位置。

*4.0 以后版本已取消此函数的功能，为了兼容，暂保留了此函数

4.2.6 Function DongleMemRead(Byval p1, p2 As Integer, buf) As Boolean

读出加密狗内存区位置 p1 开始的 p2 个字节到 Variant 变量 buf(一维字节数组)。内存区共有 24 个字节，位置为 0-23

*4.0 以后版本已取消此函数的功能，为了兼容，暂保留了此函数

4.2.7 Function DongleMemWrite(Byval p1, p2 As Integer, buf) As Boolean

写入 Variant 变量 buf(一维字节数组)到加密狗内存区位置 p1 开始的 p2 个字节内。内存区共有 24 个字节，位置为 0-23

*4.0 以后版本中已取消此函数的功能，为了兼容，暂保留了此函数

4.2.8 Function GetTemplate()

得到最近一次获得的指纹模板。

4.2.9 Function GetFingerImage(Byval AFingerImage) As Boolean

得到最近一次获得的指纹图像(BMP 格式)。

4.2.10 Function InitEngine() As Long

初始化指纹识别系统。SensorCount、SensorSN、EngineValid、ImageHeight、ImageWidth 等属性需在该函数被调用后才能返回正确结果。返回值:

0 初始化成功

1 指纹识别驱动程序加载失败

2 没有连接指纹识别仪

3 属性 SensorIndex 指定的指纹仪不存在（注意：在调用之前设置属性 SensorIndex）

可以使用方法 EndEngine 释放指纹设备系统

4.2.11 Function VerFinger(byval regTemplate, verTemplate, AdoLearning As Boolean, byval AregFeatureChanged As Boolean) As Boolean

比对两枚指纹的特征模版是否匹配。其中 regTemplate 表示指纹登记特征模版，verTemplate 表示现场采集的指纹验证特征模版，AdoLearning 表示是否进行指纹特征模版学习更新。AregFeatureChanged 表示登记模版 regTemplate 是否改变，两枚指纹匹配时返回 True，不匹配时返回 False

说明：

手指特征随时间可能会发生一定程度的变化，通常不会影响指纹的比对，但通过进行指纹特征模版学习更新，系统会综合得到新的模版，从而可以降低拒绝率。

4.2.12 Function VerFingerFromFile(regTemplateFile As String, verTemplateFile As String, AdoLearning As Boolean, byval AregFeatureChanged As Boolean) As Boolean

比对两枚指纹的特征模版文件是否匹配。其中 regTemplateFile 表示指纹登记特征模版文件，verTemplateFile 表示现场采集的指纹验证特征模版文件，AdoLearning 表示是否进行指纹特征模版学习更新。AregFeatureChanged 表示登记模版文件 regTemplateFile 是否改变。两枚指纹匹配时返回 True，不匹配时返回 False

4.2.13 Function VerRegFingerFile(RegTemplateFile As String, verTemplate, AdoLearning As Boolean, byval AregFeatureChanged As Boolean) As Boolean

比对两枚指纹的特征模版是否匹配。其中 regTemplate 表示由 FileName 指定文件中的以前登记指纹特征模版，verTemplate 表示现场采集的指纹的特

征模版，AdoLearning 表示是否进行指纹特征模版学习更新。AregFeatureChanged 表示登记模版文件 regTemplateFile 是否改变。两枚指纹匹配时返回 True，不匹配时返回 False

4.2.14 Sub PrintImageAt(HDC As OLE_HANDLE, X As Long, Y As Long, aWidth As Long, aHeight As Long)

在由（x,y）指定的位置上按照（aWidth, aHeight）指定的大小显示指纹图像，HDC 表示要显示指纹的窗口的设备描述表句柄

4.2.15 Sub PrintImageEllipseAt(HDC As OLE_HANDLE, X As Long, Y As Long, aWidth As Long, aHeight As Long, bkColor As OLE_COLOR)

在由（x,y）指定的位置上按照（aWidth, aHeight）指定的大小显示指纹图像，HDC 表示要显示指纹的窗口的设备描述表句柄。这里的指纹图像被一椭圆形包围。

4.2.16 Sub SaveBitmap(FileName As String)

保存最后一次采集到的指纹的图像到 FileName 指定的位图文件中。

4.2.17 Sub SaveJPG(FileName As String)

保存最后一次采集到的指纹的图像到 FileName 指定的 Jpeg 文件中。

4.2.18 Function SaveTemplate(FileName As String, Template) As Boolean

保存 Template 指纹的特征模版到 FileName 指定的文件中。

4.2.19 function EncodeTemplate(ASour, var ADest As String) As Boolean

将控件使用的 Variant 模板 ASour 转换为 BASE64 格式的模板字符串 ADest。

4.2.20 function DecodeTemplate(const ASour As String, ADest) As Boolean

将 BASE64 格式的模板字符串 ASour 转换为控件使用的 Variant 类型

ADset 模板。

以上两个方法主要用于模板的数据库保存，Variant 类型模板是以二进制格式数组方式存放，在 PB, VB 等语言中操作比较困难，方法 EncodeTemplate 可以将 Variant 类型编码转换为字符串类型，方法 DecodeTemplate 可以将字符串类型编码转换为 Variant 类型，需要注意的是，模板变量 BASE64 编码为字符串后，模板长度将变长。

4.2.21 function EncodeTemplate1(ASour) As String

将控件使用的 Variant 模板 ASour 转换为 BASE64 格式的模板字符串。参考 EncodeTemplate，主要方便 PB, VC 中调用

4.2.22 function DecodeTemplate1(const ASour As String) As Variant

将 BASE64 格式的模板字符串 ASour 转换为控件使用的 Variant 类型模板。参考 DecodeTemplate，主要方便 PB, VC 中调用

4.2.23 Sub BeginCapture()

设置当前指纹设备开始取像，可以使用方法 CanncelCapture 禁止当前指纹设备取像。

4.2.24 Sub EndEngine()

释放由方法 InitEngine 初始化的指纹设备，可以使用方法 InitEngine 重新初始化指纹设备。

4.2.25 function VerFingerFromStr(regTemplateStr As String, verTemplateStr As String, AdoLearning As Boolean, byval AregFeatureChanged As Boolean) As Boolean

比对两枚指纹的特征模版是否匹配。其中 regTemplateStr 表示指纹登记特征模版(BASE64 格式的字符串)，verTemplateStr 表示现场采集的指纹验证特征模版(BASE64 格式的字符串)，AdoLearning 表示是否进行指纹特征模版

学习更新。AregFeatureChanged 表示登记模版文件 regTemplateFile 是否改变。两枚指纹匹配时返回 True，不匹配时返回 False

4.2.26 function GetTemplateAsString() As String

得到最近一次获得的指纹验证或者登记模板，可以在 OnCapture，OnEnroll, OnCaptureToFile, OnEnrollToFile 事件中调用，和 GetTemplate 方法功能类似，区别在于本方法返回为转换为 BASE64 格式的模板字符串。

4.2.27 function ControlSensor(ACode As Long; AValue As Long)As Long

ACode 为 11 时控制绿灯，12 时控制红灯，13 时控制蜂鸣

AValue 为 1 表示开，0 表示关

说明：

目前只有 ZK4000 指纹仪及 ZK8000 指纹仪支持此方法。

1：N 控件接口方法：

4.2.28 Function AddRegTemplateToFPCacheDB(fpcHandle As Long, FPID As Long, pRegTemplate) As Long

添加指纹登记模板 pRegTemplate 到指纹识别高速缓冲空间 fpcHandle，FPID 是要添加登记模板的标识。

说明：

fpcHandle 是创建高速缓冲空间的返回值，参见函数 CreateFPCacheDB; fpcHandle= CreateFPCacheDB();

FPID=IdentificationInFPCacheDB(…);当比对成功后返回 FPID;

4.2.29 Function AddRegTemplateFileToFPCacheDB(fpcHandle As Long, FPID As Long, pRegTemplateFile As String) As Long

添加由 pRegTemplateFile 指定文件中的以前指纹登记特征模版到指纹识别高速缓冲空间 fpcHandle，FPID 是要添加登记模板的标识，必须>=0。返回

值为 1 表示成功，0 表示失败

4.2.30 Function CreateFPCacheDB As Long

创建指纹识别高速缓冲空间，进行 1：N 识别时必须首先调用该函数得到指纹识别缓冲空间句柄。

说明：

由于 ZKFinger 1：1 低速比对速度比较慢（在 PII 233 大约 30ms）,所以使用 AddRegTemplateToFPCache 函数加入到缓冲中的 1：1 指纹(质量较差的指纹)不能太多，否则影响比对速度。

通过 IsOneToOneTemplate 可以判断是否属于 1：1 指纹

可以同时创建多个缓冲区，用于分组比对等。

4.2.31 Sub FlushFPImages ()

清空当前指纹设备中的缓冲图像。

4.2.32 Sub FreeFPCacheDB(fpcHandle As Long)

释放指纹识别高速缓冲空间，fpcHandle 是调用该函数 CreateFPCacheDB 得到指纹识别缓冲空间句柄。

4.2.33 Function IdentificationFromFileInFPCacheDB (fpcHandle As Long, pVerTemplateFile As String, Byval Score As Long, Byval ProcessedFPNumber As Long) As Long

将指纹验证模板文件 pVerTemplateFile 和指纹识别高速缓冲空间 fpcHandle 中所有登记模板进行比对，Score 传出 ProcessedFPNumber 次比对中的最高分数，ProcessedFPNumber 传出比对的次数，当识别成功时返回值指纹标识，失败返回-1。

注意：

在识别过程中如果比对分数大于等于属性 Threshold，则认为比对成功，

不再和缓冲空间中剩余的指纹登记模板进行比对，函数返回匹配成功的指纹登记模板的指纹标识；

当指纹验证模板和指纹识别高速缓冲空间中所有指纹登记模板进行比对的分数都没有超过设定的 Threshold，但同时比对的最高分数大于等于 Score，则认为比对也是匹配成功，函数返回比对最高分数的指纹登记模板的标识，推荐设定为 8；

4.2.34 Function IdentificationInFPCacheDB (fpcHandle As Long, pVerTemplate, Byval Score As Long, Byval ProcessedFPNumber As Long) As Long

将指纹验证模板 pVerTemplate 和指纹识别高速缓冲空间 fpcHandle 中所有登记模板进行比对，Score 传出 ProcessedFPNumber 次比对中的最高分数，ProcessedFPNumber 传出比对的次数，当识别成功时返回值指纹标识，失败返回-1。

注意：

在识别过程中如果比对分数大于等于属性 Threshold，则认为比对成功，不再和缓冲空间中剩余的指纹登记模板进行比对，函数返回匹配成功的指纹登记模板的指纹标识；

当指纹验证模板和指纹识别高速缓冲空间中所有指纹登记模板进行比对的分数都没有超过设定的 Threshold，但同时比对的最高分数大于等于 Score，则认为比对也是匹配成功，函数返回比对最高分数的指纹登记模板的标识，推荐设定为 8；

4.2.35 Function IsOneToOneTemplate (ATemplate) As Boolean

判断当前指纹特征模版 Atemplate 是否为 ZKFinger 1：1 低速比对特征模版。

4.2.36 Function ModifyTemplate(byval Atemplate, AOneToOne As Boolean) As Boolean

根据 AOneToOne 修改指纹特征模版 Atemplate 为 ZKFinger 1：1 低速比

对特征模版或者高速比对特征模版。

4.2.37 Function RemoveRegTemplateFromFPCacheDB (fpcHandle As Long, FPID As Long) As Long

删除指纹识别高速缓冲空间 fpcHandle 中，指纹标识为 FPID 的指纹登记模板。返回值为 1 表示成功，0 表示失败

4.2.38 Sub CancelCapture()

禁止当前指纹设备取像，可以使用方法 BeginCapture 使指纹设备开始取像。

4.2.39 Function AddRegTemplateStrToFPCacheDB(fpcHandle As Long, FPID As Long, ARegTemplateStr As String) As Long

添加 BASE64 格式的字符串 ARegTemplateStr 指纹登记特征模版到指纹识别高速缓冲空间 fpcHandle，FPID 是要添加登记模板的标识，必须>=0。返回值为 1 表示成功，0 表示失败

4.2.40 Function IdentificationFromStrInFPCacheDB (fpcHandle As Long, AVerTemplateStr As String, Byval Score As Long, Byval ProcessedFPNumber As Long) As Long

将指纹验证模板 AverTemplateStr(BASE64 格式的字符串)和指纹识别高速缓冲空间 fpcHandle 中所有登记模板进行比对，Score 传出 ProcessedFPNumber 次比对中的最高分数，ProcessedFPNumber 传出比对的次数，当识别成功时返回值指纹标识，失败返回-1。

注意：

在识别过程中如果比对分数大于等于属性 Threshold，则认为比对成功，不再和缓冲空间中剩余的指纹登记模板进行比对，函数返回匹配成功的指纹登记模板的指纹标识；

当指纹验证模板和指纹识别高速缓冲空间中所有指纹登记模板进行比对的分数都没有超过设定的 Threshold，但同时比对的最高分数大于等于 Score，

则认为比对也是匹配成功，函数返回比对最高分数的指纹登记模板的标识，推荐设定为 9；

4.2.41 Sub SetAutoIdentifyPara(AutoIdentify As Boolean, fpcHandle As Long, Score As Long)

设置内部高速比对需要的比对方式 AutoIdentify、高速缓冲句柄 fpcHandle 和最低阀值分数 Score，Score 可以参考方法 IdentificationFromInFPCacheDB。当 AutoIdentify = True 时，按压指纹激活 OnCapture 事件，提取到指纹比对模板后，控件内部直接调用优化的高速比对功能进行比对，比对结果可以通过 OnCapture 的参数 ATemplate 或者 GetTemplate 得到，此时不需要再调用 IdentificationINFPCahceDB 函数进行比对。

请参考 OnCapture 事件。

外部图象文件接口方法：

4.2.42 Function AddBitmap(BitmapHandle As OLE_HANDLE, ValidRectX1 As Long, ValidRectY1 As Long, ValidRectX2 As Long, ValidRectY2 As Long, DPI As Long) As Boolean

使用由 BitmapHandle 指定的位图进行登记或比对。ValidRectX1、ValidRectY1、ValidRectX2、ValidRectY2 四个参数指定了图像的有效区域，如果指定的图像区域无效，将取图像取全部区域，DPI 指定了图像的分辨率大小。

4.2.43 Function AddImageFile(FileName As String, DPI As Long) As Boolean

使用由 FileName 指定的指纹图像文件(支持 BMP,JPG 格式)进行登记或比对。DPI 指定了图像的分辨率大小。

以上两个函数在使用前，如果是将图象文件用于指纹登记，首先使用 BeginEnroll,并设定 EnrollCount,如果是用于比对,使用 BeginCapture,然后使用 AddImageFile 或 AddBitmap,系统触发 OnEnroll 或 OnCapture 事件。

以上外部图象接口函数在 ZKFinger SDK Lite Version 中不提供支持。

扩展接口方法：

4.2.44 Function CreateFPCacheDBEx As Integer

功能同 CreateFPCacheDB，区别在于同时创建 9.0 和 10.0 指纹识别高速缓冲空间。

4.2.45 Sub FreeFPCacheDBEx(fpcHandle As Long)

功能同 FreeFPCacheDB，区别在于同时释放 9.0 和 10.0 指纹识别高速缓冲空间。

4.2.46 Function AddRegTemplateStrToFPCacheDBEx(fpcHandle As Integer, FPID As Integer, ARegTemplateStr As Stirng, ARegTemplate10Str As String) As Long

功能同 AddRegTemplateStrToFPCacheDB，区别在于同时添加 9.0 和 10.0 模板到指纹识别高速缓冲空间。

4.2.47 Function AddRegTemplateToFPCacheDBEx (fpcHandle As Integer, FPID As Integer, pRegTemplate, pRegTemplate10) As Long

功能同 AddRegTemplateStrToFPCacheDBEx，区别在于指纹模板为 Variant 类型。

4.2.48 Function AddRegTemplateFileToFPCacheDBEx (fpcHandle As Integer, FPID As Integer, pRegTemplateFile As String, pRegTemplateFile10 As String)

As Long

功能同 AddRegTemplateStrToFPCacheDBEx，区别在于指纹模板从文件中加载。

4.2.49 Function RemoveRegTemplateFromFPCacheDBEx(fpcHandle As Long, FPID As Long) As Long

功能同 RemoveRegTemplateFromFPCacheDB，区别在于同时清空 9.0 和 10.0 指纹模板。

4.2.50 Function GetTemplateEx(AFPEngineVersion As String) As Variant

功能同 GetTemplate，区别在于可以传入 AFPEngineVersion 参数取 9.0 或 10.0 模板。

AFPEngineVersion 为 9 时取 9.0 模板，为 10 取 10.0 模板

4.2.51 Function GetTemplateAsStringEx(AFPEngineVersion As String) As String

功能同 GetTemplateEx，区别在于返回为转换为 BASE64 格式的模板字符串。

EM/Mifare 卡操作方法：

4.2.52 Function MF_GET_SNR(commHandle As Long, deviceAddress As Long, mode As Byte, rDM_halt As Byte, ByRef snr As Byte, ByRef value As Byte) As Boolean

返回 1 个字节的单卡或多卡标识，4 个字节的卡号。

commHandle：操作的串口，0

deviceAddress：设备地址，0

mode：模式控制 0x26 一次只对一张卡操作，0x52 一次可对多张卡操作

API_halt：是否需要 halt 卡，00 不需要，01 需要

snr：返回的 1 个字节的单卡或多卡标识(读卡失败 snr[0]错误代码)

value：返回的 4 个字节的卡号

说明：

目前此方法只支持 ZK8000 指纹仪

4.2.53 Function MF_GetSerNum(commHandle As Long, deviceAddress As Long, ByRef buffer As Byte) As Boolean

读取 1 个字节的读卡器地址和 8 个字节读卡器序列号。

commHandle：操作的串口，0

deviceAddress：设备地址，0

buffer：buffer[0]读写器地址，buffer[1...8]8 个字节的读写器序列号

说明：

目前此方法只支持 ZK8000 指纹仪

4.2.54 Function MF_SetSerNum(commHandle As Long, deviceAddress As Long, ByRef newValue As Byte, ByRef buffer As Byte) As Boolean

设置 8 个字节的读卡器序列号

commHandle：操作的串口，0

deviceAddress：设备地址，0

newValue：8 个字节的读写器序列号

buffer：操作失败时返回错误代码

说明：

目前此方法只支持 ZK8000 指纹仪

4.2.55 Function MF_GetVersionNum(commHandle As Long, deviceAddress As Long, ByRef versionNum As Byte) As Boolean

读取读卡器的版本号

commHandle：操作的串口，0

deviceAddress：设备地址，0

versionNum：读写器的版本号，操作失败 versionNum[0] 为错误代码

说明：

目前此方法只支持 ZK8000 指纹仪

4.2.56 Function MF_PCDRead(commHandle As Long, deviceAddress As Long, mode As Byte, blkIndex As Byte, blkNum As Byte, ByRef key As Byte, ByRef buffer As Byte) As Boolean

读取卡内数据

commHandle：操作的串口，0

deviceAddress：设备地址，0

mode：0(keyA+单卡) 1(keyA+多卡) 2(keyB+单卡) 3(keyB+多卡)

blkIndex：块索引

blkNum：块数目

key：六个字节的密钥，成功时返回 4 字节卡号

buffer：失败时 buffer[0]为错误代码，成功时为读取到的数据

说明：

目前此方法只支持 ZK8000 指纹仪

错误代码：

0x80: 参数设置成功

0x81：参数设置失败

0x82：通讯超时

0x83：卡不存在

0x84：接收卡数据出错

0x87：未知的错误

0x85：输入参数或者输入命令格式错误

0x8A：在对于卡块初始化命令中出现错误

0x8B：在防冲突过程中得到错误的序列号

0x8C：密码认证没通过

0x90：卡不支持这个命令

0x91：命令格式有错误

0x92：在命令的 FLAG 参数中，不支持 OPTION 模式

0x93：要操作的 BLOCK 不存在
0x94：操作的对象已经被锁定，不能进行修改
0x95：锁定操作不成功
0x96：写操作不成功

4.2.57 Function MF_PCDWrite(commHandle As Long, deviceAddress As Long, mode As Byte, blkIndex As Byte, blkNum As Byte, ByRef key As Byte, ByRef buffer As Byte) As Boolean

往卡内写数据
commHandle：操作的串口，0
deviceAddress：设备地址，0
mode：0(keyA+单卡) 1(keyA+多卡) 2(keyB+单卡) 3(keyB+多卡)
blkIndex：块索引
blkNum：块数目
key：六个字节的密钥，成功时返回 4 字节卡号
buffer：待写入数据，失败时 buffer[0]为错误代码(见 4.2.56)
说明：
目前此方法只支持 ZK8000 指纹仪

4.3 事件

4.3.1 OnCapture(ActionResult AS Boolean, ATemplate)

AutoIdentify = False 时，取到用于比对的指纹验证模板 ATemplate，ActionResult =true 表示成功取到指纹模版；False 表示失败。

AutoIdentify = True 时，返回指纹比对结果(一维数组)，请参考下面定义：

ATemplate[0] 代表 ID 值 -1 代表查找失败

ATemplate[1] 返回 1:N 比对分数 即原来 Identification 函数中的 Score 参数

ATemplate[2] 1:N 指纹比对数

ATemplate[3] 1:1 指纹比对数

请参考 SetAutoIdentifyPara 方法。

4.3.2 OnCaptureToFile(ActionResult AS Boolean)

取到用于比对的指纹验证模板，模版保存到文件中，文件名称为属性 VerTplFileName 设置， ActionResult =true 表示成功取到指纹模版；False 表示失败，如果 VerTplFileName 没有设置或者为空，则不产生保存文件，但本事件仍然会触发。

4.3.3 OnEnroll(ActionResult AS Boolean, ATemplate)

用户登记指纹结束时调用该事件， ActionResult =true 表示成功登记，用 pTemplate 属性可取得指纹特征模版；False 表示失败。

4.3.4 OnEnrollToFile(ActionResult AS Boolean)

用户登记指纹结束时调用该事件， ActionResult =true 表示成功登记，指纹特征模板保存到文件中，文件名称为属性 RegTplFileName 设置；False 表示失败，如果 RegTplFileName 没有设置或者为空，则不产生保存文件，但本事件仍然会触发。。

4.3.8 OnFeatureInfo(AQuality As Long)

取得指纹初始特征，Quality 表示该指纹特征的质量，有如下可能值：

0: 好的指纹特征

1: 特征点不够

2: 其它原因导致不能取到指纹特征

-1: 可疑指纹

4.3.6 OnImageReceived(byval AImageValid As Boolean)

设备取到指纹图像或者通过 AddImageFile 和 AddBitmap 加入指纹图像时调用该事件，AImageValid 表示是否进行模板提取，设置为 False 后，系统在取到指纹图像后返回，不进行模板提取。

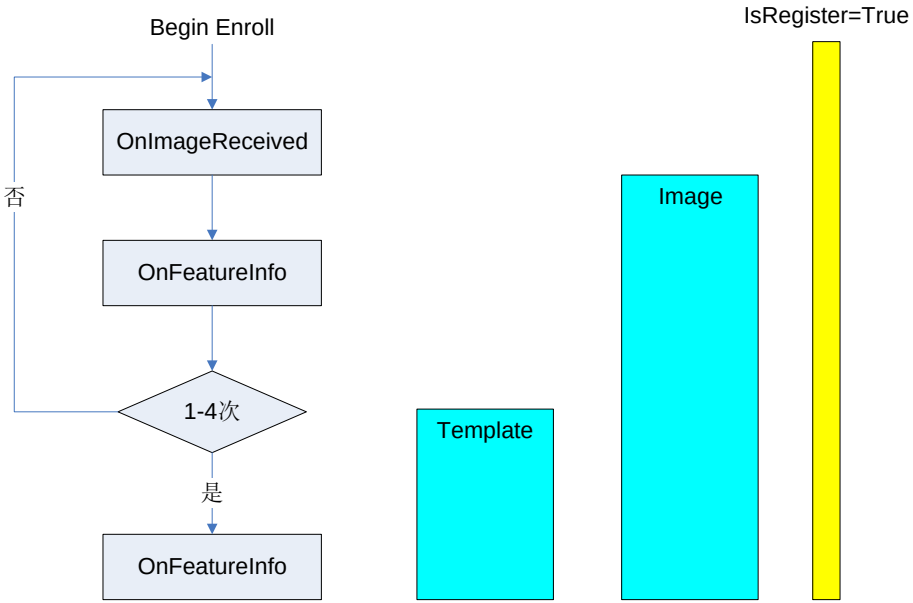
4.3.7 OnFingerTouching

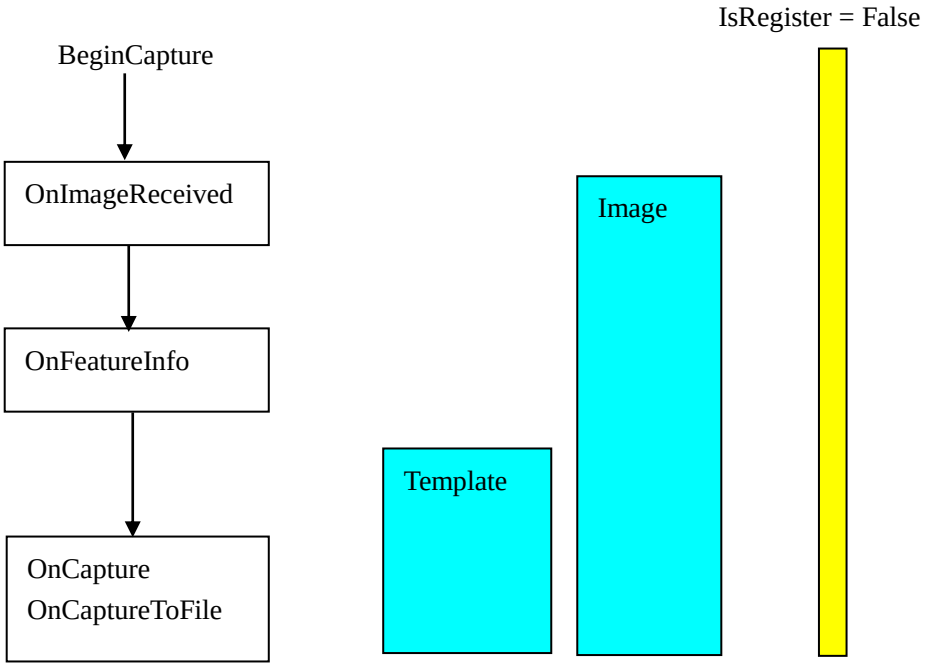
当手指按压指纹取像设备时调用该事件。

4.3.8 OnFingerLeaving

当手指移开指纹取像设备时调用该事件。

5.工作流程说明





工作流程说明：

指纹仪初始化进入工作状态后，调用 BeginEnroll 处于登记指纹状态，调用 BeginCapture 处于指纹验证状态。控件的工作方式是基于事件驱动，触发事件的顺序参考上面示意图。

指纹登记一般需要按同一手指 1-4 次，然后由识别系统综合处理得到一个指纹登记模板，按压登记指纹次数由控件属性 EnrollCount 设置，达到设定次数后会触发 OnEnroll 和 OnEnrollToFile 事件。

指纹验证时，按压手指后会触发 OnCapture 和 OnCaptureToFile 事件，此时可以调用 VerFinger 或者 IdentificationInFPCacheDB 进行 1:1 或者 1:N 比对。

需要注意每次按压手指都会触发 OnFeatureInfo 事件，如果按压手指的指纹模板质量不合格，则本次取像无效，需要重新按压手指

6.常见问题说明

6.1 1：1 和 1：N 的应用

1：1 函数主要用于需要进行 1:1 验证的开发项目，一般需要预先输入当前验证客户的标识，然后得到他已经登记的一个或几个模板与现场采集的模版进行验证；而 1：N 函数主要用于不输入客户标识，直接通过客户指纹从已经登记的指纹模板中找出自己。

1：1 主要目标是高的通过率和相对高的准确率；1：N 主要目标是高的比对速度和相对高的准确率.。

6.2 数据库中指纹模板写入和读出

SDK 中指纹模板是以 Vairant 变量的方式保存和传递，其存储的是一维二进制字节数组，不能像字符串一样直接用 SQL 语句写入和读出，有下面处理方法：

1、 EncodeTemplate 和 DecodeTemplate 方法可以在 Variant 变量和字符串变量之间互相进行 BASE64 编码转换，转换为

字符串后，模板长度会增加大约 1/3。

2、 在 OnEnroll 事件和 OnCapture 事件中直接调用 GetTemplateAsString 方法得到相应字符串形式的登记模板和比对模板。

3、直接操作 Variant 变量，下面是操作示例：

Delphi, CB：

```
procedure TFPProcess.SaveFPData(AQuery: TADOQuery; AFingerID: Integer; AFPData: OleVariant);
var
    pData: PChar;
begin
    with AQuery do
    begin
        Close;
        SQL.Clear;
        SQL.Add('SELECT * FROM zkFingerPrint WHERE FingerID = ' + IntToStr(AFingerID));
        Open;
        if IsEmpty then
            Append
        else
            Edit;
```

```
        FieldByName('FingerID').Value := AFingerID;
        //保存指纹模板
        with TBlobStream(CreateBlobStream(FieldByName('Template'), bmWrite)) do
        begin
            pData := VarArrayLock(AFPData);
            try
                Write(pData^, VarArrayHighBound(AFPData, 1) - VarArrayLowBound(AFPData, 1) + 1);
            finally
                VarArrayUnlock(AFPData);
            end;
            Free;
        end;
        Post;
        Close;
    end;
end;
procedure TFPProcess.GetFPData(AQuery: TADOQuery; AFingerID: Integer; var AFPData: OleVariant);
var
    pData: PChar;
```

```
begin
  with AQuery do begin
    Close;
    SQL.Clear;
    SQL.Add('SELECT * FROM zkFingerPrint WHERE FingerID = ' + IntToStr(AFingerID));
    Open;
    //读取数据
    if not IsEmpty then
      with TBlobStream(CreateBlobStream(FieldByName('Template'), bmRead)) do begin
        AFPData := VarArrayCreate([0, Size + 1], varByte);
        pData := VarArrayLock(AFPData);
        try
          Read(pData^, Size);
        finally
          VarArrayUnlock(AFPData);
        end;
        Free;
      end;
    Close;
```

```
    end;
end;
```

其它语言请参考www.zkteco.com上技术讨论论坛。

6.3 软件加密狗和授权许可文件

4.0 以后版本已取消

6.4 1：N 高速缓冲空间的使用

在 1：N 比对时，需要对比对模板进行分类，同时为了得到最高速度，SDK 需要创建内存空间，然后将已登记指纹加入到内存空间，高速缓冲空间实际上是内存空间，使用时需要首先用方法 CreateFPCahceDB 创建，然后用方法 AddRegTemplateToFPCahceDB，RemoveRegTemplateFromFPCacheDB 等加入或者删除指纹登记模板，最后可以使用方法 FreeFPCacheDB 释放内存空间。

可以同时创建多个高速缓冲空间以用于实现分组查询等功能。

6.5 使用平面指纹图像

在一些工程项目中，很多时候要求保存指纹图像，或者从扫描仪直接扫描得到平面指纹图像，因此 SDK Starndard Version 或 SDK Project Versioin 提供方法 AddImageFile 等可以直接从平面指纹图像得到指纹登记模板或比对模版的方法，但需要注意必须将图像的分辨率则正确传入此方法，要求不能低于 350DPI。

注意，在普通版中 SDK Lite Version 不提供此方法。

6.6 指纹识别阀值的设定

属性 Threshold 推荐值为 10，此时误判率大约 0.001%，拒绝率大约 1.5-2%之间。

属性 OneToOneThreshold 的推荐值为 10。

Score 的推荐值为 8

6.7 1：N 识别中低质量指纹模板的处理方法

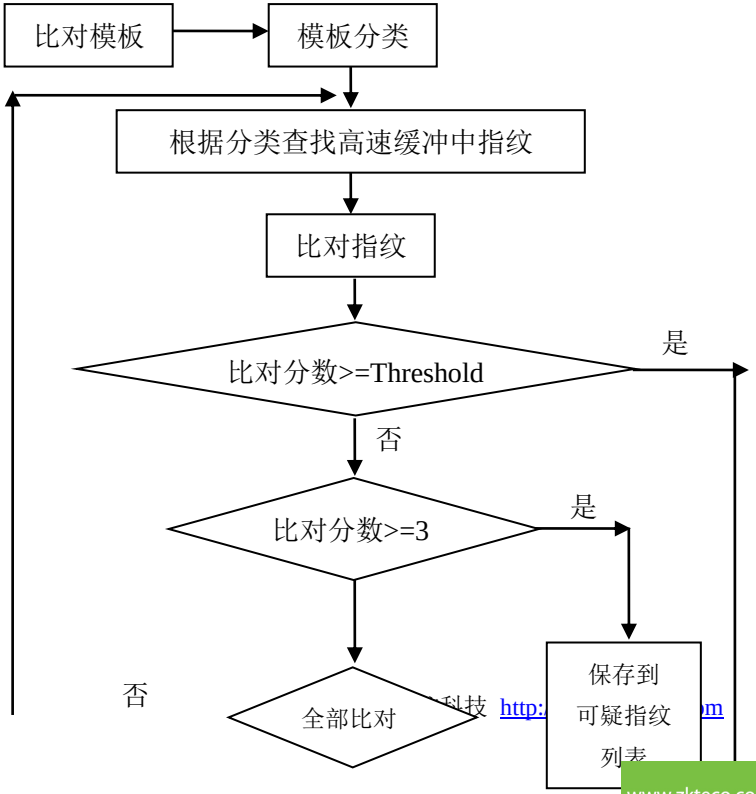
在 1：N 验证时，在指纹登记时系统自动将指纹模板按照质量的好坏分类标识和保存在模板中，质量差的登记模板称为 ZKFinger 1：1 低速比对特征模版,质量好的登记模板称为 ZKFinger 高速比对特征模版。

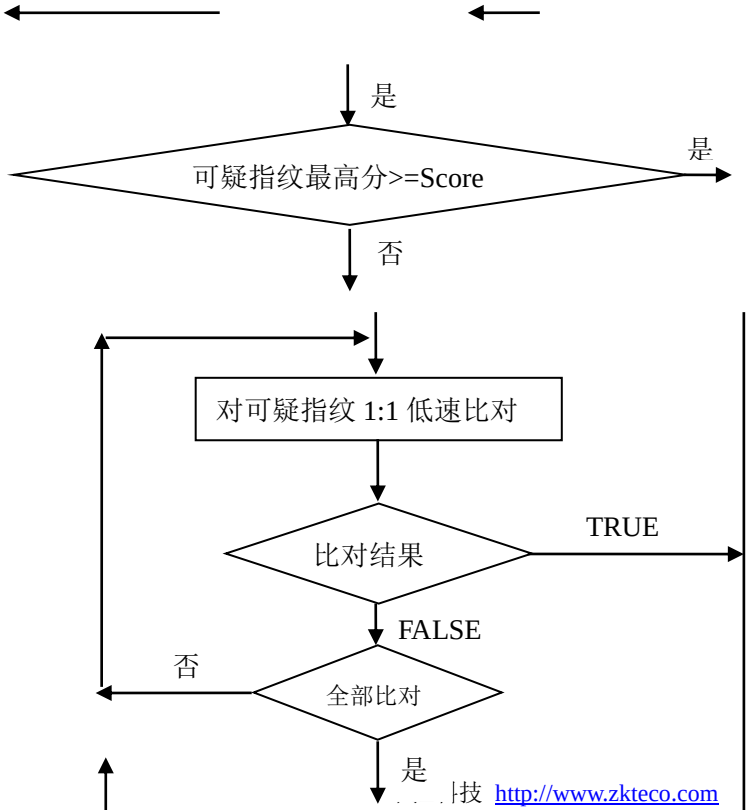
在一般的应用环境中，大约有 5%的登记指纹模板会被标识为低速比对特征模板，可以用方法 IsOneToOneTemplate 判断是否是低速比对特征模版，用方法 ModifyTemplate 可以人为强行改变质量的好坏分类标识。

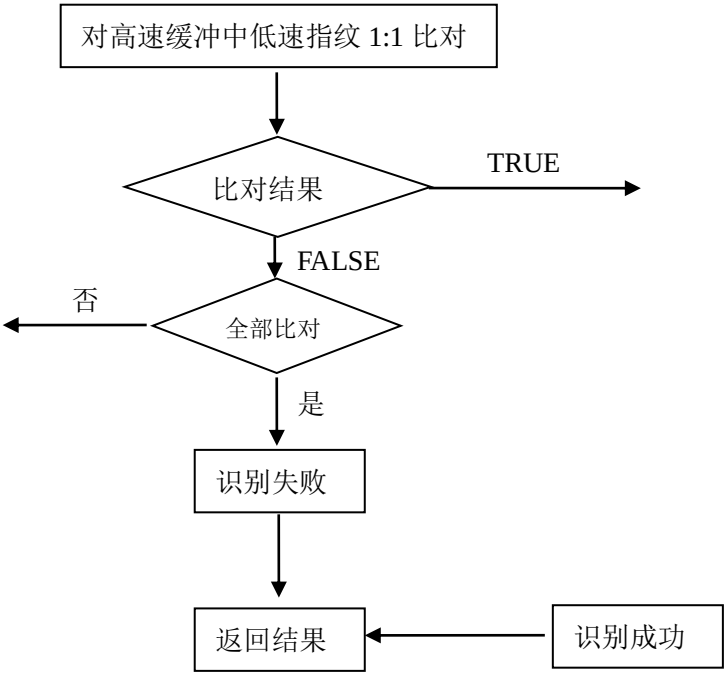
由于 ZKFinger 1：1 低速比对速度比较慢（在 PII 233 大约 30ms）,所以使用方法 AddRegTemplateToFPCache 加入到高速缓冲空间中的低速比对特征模版不能太多，否则影响比对速度。

在 1：N 验证时，使用 IdentificationInFPCacheDB 进行指纹识别的流程如下图：

IdentificationFromFileInFPCacheDB (fpcHandle As Long, pVerTemplateFile As String, Byval Score As Long, Byval ProcessedFPNumber As Long) As Long







6.8 模板操作的文件/字符串/变体变量方法

为了在不同开发语言中，方便操作模板的登记，比对等，可以选择文件方式、字符串和变体变量等三种方式进行。以下分别列出，请参考：

模板 Variant 和 String 之间的转换

EncodeTemplate, EncodeTemplate1 : Variant -> String

DecodeTemplate, DecodeTemplate1 : String -> Variant

添加模板到高速缓冲区

AddRegTemplateToFPCacheDB	Variant
AddRegTemplateStrToFPCacheDB	String
AddRegTemplateFileToFPCacheDB	File

在高速缓冲识别指纹

IdentificationFromInFPCacheDB	Variant
IdentificationFromStrInFPCacheDB	String
IdentificationFromFileCacheDB	File

1:1 比对模板

VerFinger Variant

VerFingerFromStr String

VerFingerFromFile File

7.ZKFinger SDK 开发许可协议

所有的订购和使用由中控科技股份有限公司（或它的子公司，下称“中控科技”）提供的产品（包括但不限于：ZKFinger 开发包、磁盘、CD-ROM、加密狗和开发指南），也应该服从这份协议中所提及的条款。（ZKFinger 为中控科技核心技术注册商标）

打开包含 ZKFinger 开发包和开发指南的密封包，或者在您的计算机上安装下述软件或者使用这个软件，或者使用任何中控科技的产品的同时，您就自动接受了这份协议并同时接受它的条款的约束。

如果您不愿意受它的条款的约束，您应当立即（至少在收到这个信息包之日内 7 天之内）将未使用的 ZKFinger 开发包和开发指南归还中控科技，您将收到退款。

1. 标题和所有权

这是一个许可协议而不是一个销售协议。中控科技在此授权您，您因此接受一个个人的或公司法人，不可转让的，非独家的使用中控科技产品的许可（转让许可和转卖的权利只能根据在这里清楚地规定的条款），这些许可由此处的条

款所陈述。

中控科技产品的软件组件部分，包括任何修订、更正、修改、增加或升级、中控科技开发指南，任何其他文件或软件有关的用户指南，将保留中控科技的所有权。

所有软件（包括但不限于：软件代码和根据本协议第二部分执行的工作产品）包含或表明了与之相关的知识产权（包含但不限于：版权、商业秘密，商标等），中控科技的开发指南和任何其他文件是，而且应该是唯一由中控科技所拥有。在任何法律下本协议中的内容没有构成对中控科技的知识产权的自动放弃。

2. 许可

您允许有限地使用仅有本协议所规定的而且仅仅是可执行文件格式的软件。

a) 您可以在您办公地点的计算机上安装并使用该软件。

b) 您被允许制作合理数量的软件拷贝用于开发和备份目的，但不能多于 3 个。

c) 您可以将该软件连接和拷贝到您的计算机程序中，仅用于该计算机程序开发目的，就如中控科技开发指南中描述的那样；但是，连接和拷贝另一计算机程序的任何部分将视为是衍生的作品并将继续遵守本协议的条款。

3. 转让许可

在您的计算机程序中，根据第二部分的说明，插入该软件后，您可以依据本协议的条款，把插入的软件和中控科技卖给您的加密狗的软件组成部分，向销售商或用户转让许可。转让许可之前，您应该将中控科技在条款说明的担保、

弃权和许可条款插入您和销售商或用户签定的协议中去，或者向销售商或用户直接提供上述条款。

4. 禁止使用

除了上面的 1、2 部分允许的之外，您须同意不得有如下行为：

d) 除了本协议特别授权之外，使用，修改，插入，或转让软件或任何中控科技其他产品（包括但不限于：试用组件）。

e) 在本许可下向其他人出售，转让许可，出租，转让，抵押或分享您的权利。

f) 修改，反汇编，反编译，反相工程，修改或增加该软件或企图推导该软件的源代码。

g) 将该软件放到一个服务器上，以便他人从一个公共网得到它。

h) 使用该软件的备份或档案拷贝（或允许他人使用这样的拷贝）的任何目的，除非用于替换一个损坏的或有故障的原始拷贝。

如果您是欧共体成员，这份协议不会影响您在合法执行 EC 委员会关于计算机软件保护方面的指导时所拥有的权利。如果您寻求这个指导的含义的任何信息，您可以主动与中控科技联系。

5. 有限的担保

中控科技在产品递交给您后，担保 12 个月（担保期），如下：

i) 该软件递交给您时，假设它使用的计算机的硬件和操作系统与为之设计的匹配，将 依照开发指南运行。

j) 固化该软件的磁介质没有材料和工艺上的重大缺陷。

k) ZKFinger 加密狗没有材料和工艺的重大缺陷。

6. 非担保内容

中控科技不担保它的任何产品都满足您的要求，不担保它的操作不会被打断或无错误。

在法律上允许的程度上，中控科技特别不承认所有未在这里陈述的特殊担保和所有隐含的担保，这些担保包括但不限于：适销的和用于某特定目的的隐含担保。

7. 修理的限度

违背这些担保时，中控科技唯一的责任是替换或修理，任何产品或它的部件不符合前述的有限担保要求时，将免费服务。保修索赔单必须在索赔期内写好，或故障发现日内连同令人信服的证据交给中控科技。所有的产品应当从它们的购买地返还给中控科技并且应当由返还部门负责运费和保险，该产品或其部件必须连同您的收据复印件一起返还。

8. 间接损坏的排除

双方承认该软件和中控科技产品的内在复杂性，并可能不是完全没有错误，中控科技将不对您，您的销售商，您

的软件用户或任何第三方的任何损失或损坏（包括不直接的，特定的或间接的损坏）负责（不论是否根据合同，侵权行为（包括疏忽）或其他规定），包括但不限于：由于任何使用该软件或中控科技产品或您的软件程序导致的任何商业收入的损失，利益损失，数据的损坏或丢失，或文件的丢失，即使中控科技被建议有这种损害的可能性。

9. 责任限度

在某种情况下，尽管有本协议的条款，中控科技仍对任何其产品的故障或不符合所造成的损害负责，对每个有故障产品的总责任费用将不超过您购买该产品而付给中控科技的费用。

10. 没有其他担保

除了这里指出的外，中控科技不再提供特别的或隐含的担保，也不再对本协议序 言中所描述的产品负责，包括它们的质量，性能和对某一特定目的的适应性。

11. 终止

您不能遵守本协议的条款时将终止您的许可和本协议，根据中控科技的这份协议：

l) 这份协议授予您的许可将失效，您不能继续使用（包括转让许可）许可的软件和其他许可产品。

m) 您应该立即将所有代表中控科技知识产权的有形资产及其拷贝返还中控科技，或将其包含的此类信息以电子

形式删除。

尽管本协议终止，第 1，4，5，6，7，8，9，10 和 11 部分将保留。

12. 管辖法律与仲裁权

本协议受中国政府法律的管辖，仅由中国的法庭拥有本协议之外产生的冲突和争端的仲裁权。

中控科技有限公司

8.软件售后服务

感谢您对本产品的关注，为提供给您完善的服务，请登录本公司技术论坛，并填写完整的注册信息，以便我们及时与您联系。

本公司周一至周五，上午九点到下午六点上班，周六值班，法定假日及、周日不上班。

我们随时恭候您的来电，为您迅速解答问题。

当您准备来电询问前，请先确定已经依照手册的步骤操作，且确认已经关闭所使用的其它应用程序。

来函请寄：深圳市坂田华为基地五和大道北中控大厦

邮政编码：518129

来电请拨：0755-89602345 89602346 89602347 89602348 89602670

传真服务：0755-89602194

E-mail：support@zksoftware.com

若您有关于本套产品任何技术上的问题，请详尽备妥下述资料，以便我们在最短的时间内为您解决问题，提供服

务：

1、 软件名称

2、 您的电脑配备（包括厂牌、机型、CPU、内存、光驱、主板厂牌等）

3、 XP/Vista/7/8/10 版本或其它环境

4、 任何您所使用的应用程序

5、 详细描述问题的情况

您也可以登录我们公司主页www.zkteco.com，访问技术论坛并提出您们的问题和意见，我们将尽快给您满意的答复。