Библиотека pua_fplib.dll предназначена для обслуживания прозрачного режима устройства авторизации LPOS-FP.

pua_fplib.dll – является динамически подключаемой библиотекой (dll), предназначенной для работы под управлением ОС Windows95, “98, ME, NT4, Windows2000, Windows XP.
Библиотека предоставляет такие сервисы:

- динамическая загрузка/выгрузка библиотеки;
- установка драйверов под OS;
- получение качественного шаблона ;
- получение шаблонов через оповещение;
- сравнение двух отпечатков (используя PerfectMatch BSP (STM));

Описание библиотеки для доступа к LPOS-FP устройству авторизации компании POSua

LPOS-FP

Rev. 2-04/10

Содержание

1. Описание процедур и функций библиотеки

Ниже следующие процедуры и функции экспортируются из библиотеки и предназначены для работы совместно с драйвером для доступа к клавиатурам и устройствам ввода POSua.

1.1 Инициализация библиотеки

Перед выполнением операций с библиотекой необходима обязательная процедура инициализации библиотеки. Во время инициализации библиотека производит проверку присутствия драйвера доступа к POSua устройствам, поддержку операционной системы, под которой она запущена, и инициализирует внутренние переменные.

Все вызовы функций библиотеки (кроме функции установки драйверов(RD_InstallDRV), требую проведение инициализации с успешным результатом (ERROR_NO_ERROR)).

Библиотека экспортирует процедуру инициализации RD_Init, которая имеет такой формат:

Для C:
long WINAPI fp_Init(**void**);

Для DELPHI:
Function fp_Init : **integer**; **StdCall**;

Функция не имеет параметров, и возвращает следующие коды ошибок:

Код ошибки	Описание
ERROR_NO_ERROR	Инициализация успешно выполнена.

1.2 Установка POSua PS/2 драйвера

Для доступа к устройствам, библиотека использует специальный драйвер. Библиотека может установить драйвер из своих ресурсов, для этого экспортируется функция fp_InstallDriver, которая имеет такой формат:

Для C:
long WINAPI fp_InstallDriver (**void**);

Для DELPHI:
Function fp_InstallDriver() : **integer**; StdCall;

Возвращаемые коды ошибок:

Код ошибки	Описание
ERROR_NO_ERROR	Завершение прошло успешно
ERROR_I_NODEVICE	В системе отсутствует PS/2 устройство ввода информации
ERROR_I_FAILED	Общая ошибка установки. Может быть вызвана рядом из следующих причин: 2. в ОС отсутствует необходимое API для установки драйвера, либо оно не доступно; 3. недоступен временный каталог для временной распаковки драйвера; 4. отсутствует свободное место для временной распаковки драйвера; 5. пользователь во время установки, своим ответом на один из диалогов ОС об установке драйвер, прервал ее; Более подробную информацию, можно получить из GetLastErro();
ERROR_I_NOACCESS	Пользователь не наделен правами для установки драйверов. Обратитесь к сетевому администратору.

После установки драйвера необходимо перезагрузить компьютер.

1.3. Получение качественного шаблона отпечатка

Для составления базы данных отпечатков пользователей, рекомендуется использовать трехкратное сканирование и обработку отпечатка. Использование таких образцов позволит добиться максимального качестве сравнения отпечатков. Для выполнения сканирования с подключенной LPOS-FP клавиатуры библиотека экспортирует функцию fp_Enroll. Во время выполнения сканирования на экран будет выведен диалог, информирующий пользователя о действиях, которые необходимо выполнить.

Для C:
long WINAPI fp_Enroll (**void*** Buf**, unsigned long*** Size);

Для DELPHI:
Function fp_Enroll(Buf : Pointer; Size : PDword) : **integer**; StdCall;

Где:

Buf – указатель на буфер, куда будет помещен шаблон(Буфер должен иметь размер не менее 1024 байт);

Size – указатель на переменную, куда будет помещен размер шаблона.

Возвращаемые коды ошибок:

Код ошибки	Описание
ERROR_NO_ERROR	Завершение прошло успешно
ERROR_NOT_INITED	Не произведена инициализация библиотеки.
ERROR_NO_DRIVER	PS/2 драйвер не установлен

ERROR_NO_DEVICE	Устройство отсутствует либо ошибка обмена

1.4 Установка функции оповещения о успешном сканировании отпечатка

Функция оповещения – единственный способ получения отпечатка, предназначенного для верификации. Библиотека вызывает функцию оповещения, при получении нового шаблона от клавиатуры. При этом как параметры передается указатель на шаблон полученного отпечатка и его размер. Для установки функции оповещения библиотека экспортирует следующую функцию:

Для C:
long WINAPI fp_SetCallback(**FPCallBack** cb);
typedef long (**FPCallBack**)(void* data, long size);

Для DELPHI:
Function fp_SetCallback(cb : TFPCallBack) : **integer**; StdCall;
type **TFPCallBack** = Procedure(Data : Pointer; Size : Integer); StdCall;

Где cb – указатель на функцию, которая имеет следующий прототип:

Возвращаемые коды ошибок:

Код ошибки	Описание
ERROR_NO_ERROR	Завершение прошло успешно
ERROR_NOT_INITED	Не произведена инициализация библиотеки.
ERROR_NO_DRIVER	PS/2 драйвер не установлен

1.5 Сравнение двух шаблонов

После получения шаблона отпечатка через функцию оповещение, необходимо провести его верификацию с эталонными шаблонами, выполнить это можно вызвав экспортируемую библиотекой функцию fp_CompareTemplates. Эта функция производит верификацию с помощью PerfectMactch ® (STm).

Для C:
long WINAPI fp_CompareTemplates(void* tmpl1, void* tmpl2);
typedef long (**FPCallBack**)(void* data, long size);

Для DELPHI:
Function fp_CompareTemplates (tmpl1, templ2 : Pointer) : **integer**; StdCall;
type **TFPCallBack** = Procedure(Data : Pointer; Size : Integer); StdCall;

Где
tmpl1, tmpl2 – указатели на данные шаблона для сравнения и шаблона-эталона.

Возвращаемые коды ошибок:

Код ошибки	Описание
ERROR_NO	Завершение прошло успешно

_ERROR	
ERROR_NO_LIB	Отсутствует PerfectMatch библиотека
ERROR_LIB_ERROR	Ошибка при выполнении сравнения(возможно поврежден шаблон)
ERROR_FP_MATCH	Совпадение отпечатков (результат сравнения)
ERROR_FP_NOMATCH	Отпечатки разные (результат сравнения)

Приложение 1. Файл-определений для библиотеки

```
unit pua_fplib;

// Описание функций экспортируемых из pua_fplib

interface
Uses Windows;

Type
    TFPCallBack = Procedure(Data : Pointer; Size : Integer); StdCall;

Const
    ERROR_NO_ERROR           = 0;
    ERROR_NOT_INITED         = -100;

    ERROR_OS_NOTSUPPORTED    = -1001;
    ERROR_NO_DRIVER          = -1002;
    ERROR_I_FAILED           = -1003;
    ERROR_I_NOACCESS         = -1004;
    ERROR_I_NODEVICE         = -1005;
    ERROR_DRIVER_FAIL        = -1006;
    ERROR_NO_DEVICE          = -1100;

    ERROR_NO_LIB             = -2000;
    ERROR_LIB_ERROR          = -2001;

    ERROR_INTERNAL_001       = -3000;
    ERROR_INTERNAL_002       = -3001;
    ERROR_INTERNAL_003       = -3002;
    ERROR_INTERNAL_004       = -3004;

    ERROR_FP_MATCH       = 100;
    ERROR_FP_NOMATCH     = 101;


function fp_Init() : Integer; StdCall; External 'pua_fplib.dll';
Function fp_InstallDriver() : Integer; StdCall; External 'pua_fplib.dll';
Function fp_Enroll(Buf : Pointer; Size : PDword) : Integer; StdCall; External
'pua_fplib.dll';
Function fp_SetCallback(cb : TFPCallBack) : Integer; StdCall; External 'pua_fplib.dll';
Function fp_CompareTemplates(tmpl1, tmpl2 : Pointer) : Integer; StdCall;External
'pua_fplib.dll';

implementation

end.
```