* **API Reference**

L398界面：(可与E550/E560和L370/L398S共享)

[**PPL398\_InitialDevice**](#PPL398_InitialDevice)

[**PPL398\_UnInitialDevice**](#PPL398_UnInitialDevice)

[**PPL398\_OpenLCD**](#PPL398_OpenLCD)

[**PPL398\_Clear**](#PPL398_Clear)

[**PPL398\_GetHWPenID**](#PPL398_GetHWPenID)

[**PPL398\_GetHWPadID**](#PPL398_GetHWPadID)

[**PPL398\_GetHWVersionID**](#PPL398_GetHWVersionID)

[**PPL398\_DisplayHWVersion**](#PPL398_DisplayHWVersion)

[**PPL398\_SetPenStyle**](#PPL398_SetPenStyle)

[**PPL398\_SetPenWidth**](#PPL398_SetPenWidth)

[**PPL398\_SignPadEventCallback**](#PPL398_SignPadEventCallback)

[**PPL398\_GetTotalInkPointsNumber**](#PPL398_GetTotalInkPointsNumber)

[**PPL398\_GetInkPointData**](#PPL398_GetInkPointData)

[**PPL398\_GetInkPointDataTimeInfo**](#PPL398_GetInkPointDataTimeInfo)

[**PPL398\_FreeInkPointDataTimeInfo**](#PPL398_FreeInkPointDataTimeInfo)

[**PPL398\_SaveDrawingImage**](#PPL398_SaveDrawingImage)

[**PPL398\_SetCropImageMargin**](#PPL398_SetCropImageMargin)

[**PPL398\_GetDeviceInfo**](#PPL398_GetDeviceInfo)

[**PPL398\_AboutBox**](#PPL398_AboutBox)

[**PPL398\_Base64Encode**](#PPL398_Base64Encode)

[**PPL398\_Base64EncodeFree**](#PPL398_Base64EncodeFree)

[**PPL398\_Base64Decode**](#PPL398_Base64Decode)

[**PPL398\_GetTotalDecodeInkPointsNumber**](#PPL398_GetTotalDecodeInkPointsNumber)

[**PPL398\_GetDecodeInkPointData**](#PPL398_GetDecodeInkPointData)

[PPL398\_SetPenColor](#PPL398_SetPenColor)

L500界面:

[**PPL500\_InitialDevice**](#PPL500_InitialDevice)

[**PPL500\_UnInitialDevice**](#PPL500_UnInitialDevice)

[**PPL500\_OpenLCD**](#PPL500_OpenLCD)

[**PPL500\_Clear**](#PPL500_Clear)

[**PPL500\_GetHWPenID**](#PPL500_GetHWPenID)

[**PPL500\_GetHWPadID**](#PPL500_GetHWPadID)

[**PPL500\_GetHWVersionID**](#PPL500_GetHWVersionID)

[**PPL500\_DisplayHWVersion**](#PPL500_DisplayHWVersion)

[**PPL500\_SetPenStyle**](#PPL500_SetPenStyle)

[**PPL500\_SetPenWidth**](#PPL500_SetPenWidth)

[**PPL500\_SignPadEventCallback**](#PPL500_SignPadEventCallback)

[**PPL500\_GetTotalInkPointsNumber**](#PPL500_GetTotalInkPointsNumber)

[**PPL500\_GetInkPointData**](#PPL500_GetInkPointData)

[**PPL500\_GetInkPointDataTimeInfo**](#PPL500_GetInkPointDataTimeInfo)

[**PPL500\_FreeInkPointDataTimeInfo**](#PPL500_FreeInkPointDataTimeInfo)

[**PPL500\_SaveDrawingImage**](#PPL500_SaveDrawingImage)

[**PPL500\_SetCropImageMargin**](#PPL500_SetCropImageMargin)

[**PPL500\_GetDeviceInfo**](#PPL500_GetDeviceInfo)

[**PPL500\_AboutBox**](#PPL500_AboutBox)

[**PPL500\_Base64Encode**](#PPL500_Base64Encode)

[**PPL500\_Base64EncodeFree**](#PPL500_Base64EncodeFree)

[**PPL500\_Base64Decode**](#PPL500_Base64Decode)

[**PPL500\_GetTotalDecodeInkPointsNumber**](#PPL500_GetTotalDecodeInkPointsNumber)

[**PPL500\_GetDecodeInkPointData**](#PPL500_GetDecodeInkPointData)

[PPL500\_SetPenColor](#PPL500_SetPenColor)

L501F界面：(可与L501G共享)

[**PPL501F\_InitialDevice**](#PPL501F_InitialDevice)

[**PPL501F\_UnInitialDevice**](#PPL501F_UnInitialDevice)

[**PPL501F\_OpenLCD**](#PPL501F_OpenLCD)

[**PPL501F\_Clear**](#PPL501F_Clear)

[**PPL501F\_GetHWPenID**](#PPL501F_GetHWPenID)

[**PPL501F\_GetHWPadID**](#PPL501F_GetHWPadID)

[**PPL501F\_GetHWVersionID**](#PPL501F_GetHWVersionID)

[**PPL501F\_DisplayHWVersion**](#PPL501F_DisplayHWVersion)

[**PPL501F\_SetPenStyle**](#PPL501F_SetPenStyle)

[**PPL501F\_SetPenWidth**](#PPL501F_SetPenWidth)

[**PPL501F\_SignPadEventCallback**](#PPL501F_SignPadEventCallback)

[**PPL501F\_GetTotalInkPointsNumber**](#PPL501F_GetTotalInkPointsNumber)

[**PPL501F\_GetInkPointData**](#PPL501F_GetInkPointData)

[**PPL501F\_GetInkPointDataTimeInfo**](#PPL501F_GetInkPointDataTimeInfo)

[**PPL501F\_FreeInkPointDataTimeInfo**](#PPL501F_FreeInkPointDataTimeInfo)

[**PPL501F\_SaveDrawingImage**](#PPL501F_SaveDrawingImage)

[**PPL501F\_SetSaveClipImageMargin**](#PPL501F_SetSaveClipImageMargin)

[**PPL501F\_GetDeviceInfo**](#PPL501F_GetDeviceInfo)

[**PPL501F\_AboutBox**](#PPL501F_AboutBox)

[**PPL501F\_Base64Encode**](#PPL501F_Base64Encode)

[**PPL501F\_Base64EncodeFree**](#PPL501F_Base64EncodeFree)

[**PPL501F\_Base64Decode**](#PPL501F_Base64Decode)

[**PPL501F\_GetTotalDecodeInkPointsNumber**](#PPL501F_GetTotalDecodeInkPointsNumber)

[**PPL501F\_GetDecodeInkPointData**](#PPL501F_GetDecodeInkPointData)

[**PPL501F\_SetPenColor**](#PPL501F_SetPenColor)

[**PPL501F\_SetDisplayFPImageAt**](#PPL501F_SetDisplayedFPImageAt)

[**PPL501F\_GetFPImageBase64Data**](#PPL501F_GetFPImageBase64Data)

[**PPL501F\_FreeFPImageBase64Data**](#PPL501F_FreeFPImageBase64Data)

LC700介面：

[**PPLC700\_InitialDevice**](#PPLC700_InitialDevice)

[**PPLC700\_UnInitialDevice**](#PPLC700_UnInitialDevice)

[**PPLC700\_Clear**](#PPLC700_Clear)

[**PPLC700\_GetHWPadID**](#PPLC700_GetHWPadID)

[**PPLC700\_GetHWVersionID**](#PPLC700_GetHWVersionID)

[**PPLC700\_SetPenStyle**](#PPLC700_SetPenStyle)

[**PPLC700\_SetPenWidth**](#PPLC700_SetPenWidth)

[**PPLC700\_SignPadEventCallback**](#PPLC700_SignPadEventCallback)

[**PPLC700\_GetTotalInkPointsNumber**](#PPLC700_GetTotalInkPointsNumber)

[**PPLC700\_GetInkPointData**](#PPLC700_GetInkPointData)

[**PPLC700\_GetInkPointDataTimeInfo**](#PPLC700_GetInkPointDataTimeInfo)

[**PPLC700\_FreeInkPointDataTimeInfo**](#PPLC700_FreeInkPointDataTimeInfo)

[**PPLC700\_SaveDrawingImage**](#PPLC700_SaveDrawingImage)

[**PPLC700\_SetSaveClipImageMargin**](#PPLC700_SetSaveClipImageMargin)

[**PPLC700\_GetDeviceInfo**](#PPLC700_GetDeviceInfo)

[**PPLC700\_AboutBox**](#PPLC700_AboutBox)

[**PPLC700\_Base64Encode**](#PPLC700_Base64Encode)

[**PPLC700\_Base64EncodeFree**](#PPLC700_Base64EncodeFree)

[**PPLC700\_Base64Decode**](#PPLC700_Base64Decode)

[**PPLC700\_GetTotalDecodeInkPointsNumber**](#PPLC700_GetTotalDecodeInkPointsNumber)

[**PPLC700\_GetDecodeInkPointData**](#PPLC700_GetDecodeInkPointData)

[**PPLC700\_SetPenColor**](#PPLC700_SetPenColor)

L398界面: (可与E550/E560和L370/L398S共享)

**PPL398\_InitialDevice**

设定L398签名板，初始化签名窗口以及设定签名画布大小。

bool PPL398\_InitialDevice(int nDeviceType, QWidget

\*pParentWidget, int nX, int nY, int nWidth, int nHeight)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

设备ID定义：

enum DEVICE\_TYPE\_ID

{

PPL398\_DEVICE = 2,

PPL500\_DEVICE = 5,

PPL501F\_DEVICE = 7,

};

***pParentWidget***

[in] 父窗口指针。

***nX***

[in] 签名窗口起始点x坐标。

***nY***

[in] 签名窗口起始点y坐标。

***nWidth***

[in] 签名窗口宽度。

***nHeight***

[in] 签名窗口高度。

**Remark**

1. 可使用PPL398\_InitialDevice去设定画布大小。
2. 如需重新设定新画布大小时，必须先使用PPL398\_UnInitialDevice后，再使用PPL398\_InitialDevice设置一次。

**Return Value**

参阅回传值定义

**PPL398\_UnInitialDevice**

卸除L398签名板，关闭签名窗口，释放资源。

int PPL398\_UnInitialDevice(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

**Return value**

参阅回传值定义

**PPL398\_OpenLCD**

打开L398签名板LCD

int PPL398\_OpenLCD(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***bOn***

【打开LCD】：bOn = true

【关闭LCD】：bOn = false

**Return Value**

参阅回传值定义

**PPL398\_Clear**

清除L398签名板画布上的笔迹纪录。

int PPL398\_Clear (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

参阅回传值定义

**PPL398\_GetHWPenID**

获取L398签名板Pen ID

int PPL398\_GetHWPenID(int nDeviceType, char\* pPenID, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pPenID***

[out] 输出Pen ID字符串指针

***nMaxCount***

[in] 输出Pen ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPL398\_GetHWPadID**

获取L398签名板Pad ID

int PPL398\_GetHWPadID(int nDeviceType, char\* pPadID, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pPadID***

[out] 输出Pad ID字符串指针

***nMaxCount***

[in] 输出Pad ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPL398\_GetHWVersionID**

获取L398签名板的版本ID

int PPL398\_GetHWVersionID(int nDeviceType, char\* pVersion, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pVersion***

[out] 输出版本ID字符串指针

***nMaxCount***

[in] 输出版本ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPL398\_DisplayHWVersion**

开关L398签名板上版本信息的显示

Int PPL398\_DisplayHWVersion(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***bOn***

【显示】：bOn = true

【关闭】：bOn = false

**Return Value**

参阅回传值定义

**PPL398\_SetPenStyle**

设定L398签名板画笔种类。

int PPL398\_SetPenStyle (int nDeviceType, bool bPressureOn)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

***bPressureOn***

[in] 画笔种类。

【压感笔】：bPressureOn = true

【定宽笔】：bPressureOn = false

**Return value**

参阅回传值定义

**PPL398\_SetPenWidth**

设定L398签名板画笔宽度。

int PPL398\_SetPenWidth (int nDeviceType, int nPenWidth)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

***nPenWidth***

[in] 画笔宽度值。(1～10)

**Return value**

参阅回传值定义

**PPL398\_SignPadEventCallback**

设定L398签名板上按【确定】或【取消】时的回传函数

LONG PPL398\_SignPadEventCallback)(int nDeviceType,

SIGNPAD\_EVENT\_CALLBACK pfnSignPadEventCallback)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pfnSignPadEventCallback***

[in] 按【确定】或【取消】时的回传函数地址

回传函数定义：

typedef void (\*SIGNPAD\_EVENT\_CALLBACK)(int nEventID);

**Remark**

当L398签名板按【确定】或【取消】时会调用pfnSignPadEventCallback。

**Return Value**

参阅回传值定义

**PPL398\_GetTotalInkPointsNumber**

取得L398签名板目前笔迹数组点纪录数量。

int PPL398\_GetTotalInkPointsNumber(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

回传笔迹数组点数量, 但如果笔迹点为空或者其他错误的话, 将会回传负的

错误值

**PPL398\_GetInkPointData**

获取L398签名板单点笔迹数据

Int PPL398\_GetInkPointData(int nDeviceType, INKPOINTDATA&

currentInkPoint)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***currentInkPoint***

[out] 笔迹数据结构体。

笔迹数据结构体:

typedef struct \_INKPOINTDATA

{

int nIndex;

short nX;

short nY;

short nPressureValue;

bool bStrokeEnd;

} INKPOINTDATA, \*PINKPOINTDATA;

**Return Value**

参阅回传值定义

**PPL398\_GetInkPointDataTimeInfo**

获取L398签名板单点笔迹的时间信息。

char\* PPL398\_GetInkPointTime(int nDeviceType, int nIndex)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 单点笔迹序号

**Return value**

单点笔迹时间信息字符串。

**PPL398\_FreeInkPointDataTimeInfo**

释放L398签名板单点笔迹时间信息字符串。

void PPL398\_FreeInkPointDataTimeInfo(int nDeviceType, char\*

pInkPointDataTimeInfo)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pInkPointDataTimeInfo***

[in] 单点笔迹时间信息字符串

**Return value**

无。

**PPL398\_SaveDrawingImage**

储存L398签名板签名笔迹内容至支持的图像文件案。

int PPL398\_SaveDrawingImage(int nDeviceType, int

nImageTypeID, wchar\_t\* pImageFilePath, int

nDPILevel)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 储存影像文件类型

1 : BMP

2 : JPG

3 : PNG

4 : PDF

5 : SVG

***pImageFilePath***

[in] 储存影像文件名

***nDPILevel***

[in] 保存影像分辨率等级。(0～1)

0：150 DPI

1：300 DPI

**Remark**

支持的影像文件类型总共有BMP/JPG/PNG/PDF/SVG等五种格式，pImageFilePath为储存影像的完整文件名。

**Return value**

参阅回传值定义

**PPL398\_SetCropImageMargin**

设定裁切L398签名板签名笔迹影像的水平和垂直边界。

int PPL398\_SetCropImageMargin(int nDeviceType, bool

bCropImage, int nCropMarginX, int nCropMarginY)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***bCropImage***

[in] 是否要裁切笔迹影像?

【是】：bCropImage = true

【否】：bCropImage = false

***nCropMarginX***

[in] 裁切笔迹的同时, 并额外保留的水平边界。

***nCropMarginY***

[in] 裁切笔迹的同时, 并额外保留的垂直边界。

**Remark**

1) nCropMarginX，nCropMarginY要有作用，必须bCropImage等于Ture。

2)需搭配PPL398\_SaveDrawingImage API才能储存裁切笔迹影像。

**Return value**

参阅回传值定义

**PPL398\_GetDeviceInfo**

获取L398签名板相关信息

int PPL398\_GetDeviceInfo(int nDeviceType, int nIndex, void\*

pOutput)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 设备信息序号

1 ：设备当前状态

2 ：签名设备的X的坐标范围

3 ：签名设备的Y的坐标范围

4 ：签名设备的最大压力值

5 ：签名设备的厂商名字

6 ：签名设备的设备名字

7 ：签名设备的SDK版本

***pOutput***

[out] 获取的设备信息

**Return Value**

参阅回传值定义

**PPL398\_AboutBox**

显示SDK版本信息。

void PPL398\_AboutBox(QWidget \*pParent)

**Parameter**

***pParent***

[in] 父窗口指针。

**Return value**

无。

**PPL398\_Base64Encode**

将L398签名板签名笔迹以指定的影像格式转换为Base64编码的字符串。

char\* PPL398\_Base64Encode (int nDeviceType, int nImageTypeID)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 欲编码的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

**Return value**

回传编码后的字符串, 若失败会回传NULL

**PPL398\_Base64EncodeFree**

释放Base64编码占用的内存，删除编码数据。

void PPL398\_Base64EncodeFree(int nDeviceType, char\* pData)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pData***

[in] Base64编码数据指针

**Return value**

无

**PPL398\_Base64Decode**

将Base64编码的字符串译码为原始笔迹数据。

int PPL398\_Base64Decode(int nDeviceType, int nImageTypeID,

const char\* pEncodedData, wchar\_t\* pImageFilePath)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 编码的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

***pEncodedData***

[in] 编码的数据字符串

***pImageFilePath***

[in] 译码数据储存完整文件名

**Return Value**

参阅回传值定义

**PPL398\_GetTotalDecodeInkPointsNumber**

获取解碼后得到的笔迹点数。

int PPL398\_GetTotalDecodeInkPointsNumber (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

回传笔迹数组点数量, 但如果笔迹点为空或者其他错误的话, 将会回传负的

错误值

**PPL398\_GetDecodeInkPointData**

获取译码后得到的单点笔迹数据。

int PPL398\_GetDecodeInkPointData (int nDeviceType, int nIndex,

int nTypeID)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 笔迹点在数组中的序号

***nTypeID***

[in] 取值种类。

1：取得X坐标

2：取得Y坐标

3：取得是否为每笔划结束点状态

**Return Value**

参阅回传值定义

**PPL398\_SetPenColor**

設定L398畫筆顏色。

int PPL398\_SetPenColor (int nDeviceType, int nR, int nG, int nB)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nR***

[in] 畫筆紅色比例值。 (0~255)

***nG***

[in] 畫筆綠色比例值。 (0~255)

***nB***

[in] 畫筆藍色比例值。 (0~255)

**Return value**

參閱回傳值定義

L500界面

**PPL500\_InitialDevice**

设定L500签名板，初始化签名窗口以及设定签名画布大小。

bool PPL500\_InitialDevice(int nDeviceType, QWidget

\*pParentWidget, int nX, int nY, int nWidth, int nHeight)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

设备ID定义：

enum DEVICE\_TYPE\_ID

{

PPL398\_DEVICE = 2,

PPL500\_DEVICE = 5,

PPL501F\_DEVICE = 7,

};

***pParentWidget***

[in] 父窗口指针。

***nX***

[in] 签名窗口起始点x坐标。

***nY***

[in] 签名窗口起始点y坐标。

***nWidth***

[in] 签名窗口宽度。

***nHeight***

[in] 签名窗口高度。

**Remark**

1. 可使用PPL500\_InitialDevice去设定画布大小。
2. 如需重新设定新画布大小时，必须先使用PPL500\_UnInitialDevice后，再使用PPL500\_InitialDevice设置一次。

**Return Value**

参阅回传值定义

**PPL500\_UnInitialDevice**

卸除L500签名板，关闭签名窗口，释放资源。

int PPL500\_UnInitialDevice(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

**Return value**

参阅回传值定义

**PPL500\_OpenLCD**

打开L500签名板LCD

int PPL500\_OpenLCD(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***bOn***

【打开LCD】：bOn = true

【关闭LCD】：bOn = false

**Return Value**

参阅回传值定义

**PPL500\_Clear**

清除L500签名板画布上的笔迹纪录。

int PPL500\_Clear (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

参阅回传值定义

**PPL500\_GetHWPenID**

获取L500签名板Pen ID

int PPL500\_GetHWPenID(int nDeviceType, char\* pPenID, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pPenID***

[out] 输出Pen ID字符串指针

***nMaxCount***

[in] 输出Pen ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPL500\_GetHWPadID**

获取L500签名板Pad ID

int PPL500\_GetHWPadID(int nDeviceType, char\* pPadID, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pPadID***

[out] 输出Pad ID字符串指针

***nMaxCount***

[in] 输出Pad ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPL500\_GetHWVersionID**

获取L500签名板的版本ID

int PPL500\_GetHWVersionID(int nDeviceType, char\* pVersion, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pVersion***

[out] 输出版本ID字符串指针

***nMaxCount***

[in] 输出版本ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPL500\_DisplayHWVersion**

开关L500签名板上版本信息的显示

Int PPL500\_DisplayHWVersion(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***bOn***

【显示】：bOn = true

【关闭】：bOn = false

**Return Value**

参阅回传值定义

**PPL500\_SetPenStyle**

设定L500签名板画笔种类。

int PPL500\_SetPenStyle (int nDeviceType, bool bPressureOn)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

***bPressureOn***

[in] 画笔种类。

【压感笔】：bPressureOn = true

【定宽笔】：bPressureOn = false

**Return value**

参阅回传值定义

**PPL500\_SetPenWidth**

设定L500签名板画笔宽度。

int PPL500\_SetPenWidth (int nDeviceType, int nPenWidth)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

***nPenWidth***

[in] 画笔宽度值。(1～10)

**Return value**

参阅回传值定义

**PPL500\_SignPadEventCallback**

设定L500签名板上按【确定】或【取消】时的回传函数

LONG PPL500\_SignPadEventCallback)(int nDeviceType,

SIGNPAD\_EVENT\_CALLBACK pfnSignPadEventCallback)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pfnSignPadEventCallback***

[in] 按【确定】或【取消】时的回传函数地址

回传函数定义：

typedef void (\*SIGNPAD\_EVENT\_CALLBACK)(int nEventID);

**Remark**

当L500签名板按【确定】或【取消】时会调用pfnSignPadEventCallback。

**Return Value**

参阅回传值定义

**PPL500\_GetTotalInkPointsNumber**

取得L500签名板目前笔迹数组点纪录数量。

int PPL500\_GetTotalInkPointsNumber(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

回传笔迹数组点数量, 但如果笔迹点为空或者其他错误的话, 将会回传负的

错误值

**PPL500\_GetInkPointData**

获取L500签名板单点笔迹数据

Int PPL500\_GetInkPointData(int nDeviceType, INKPOINTDATA&

currentInkPoint)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***currentInkPoint***

[out] 笔迹数据结构体。

笔迹数据结构体:

typedef struct \_INKPOINTDATA

{

int nIndex;

short nX;

short nY;

short nPressureValue;

bool bStrokeEnd;

} INKPOINTDATA, \*PINKPOINTDATA;

**Return Value**

参阅回传值定义

**PPL500\_GetInkPointDataTimeInfo**

获取L500签名板单点笔迹的时间信息。

char\* PPL500\_GetInkPointTime(int nDeviceType, int nIndex)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 单点笔迹序号

**Return value**

单点笔迹时间信息字符串。

**PPL500\_FreeInkPointDataTimeInfo**

释放L500签名板单点笔迹时间信息字符串。

void PPL500\_FreeInkPointDataTimeInfo(int nDeviceType, char\*

pInkPointDataTimeInfo)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pInkPointDataTimeInfo***

[in] 单点笔迹时间信息字符串

**Return value**

无。

**PPL500\_SaveDrawingImage**

储存L500签名板签名笔迹内容至支持的图像文件案。

int PPL500\_SaveDrawingImage(int nDeviceType, int

nImageTypeID, wchar\_t\* pImageFilePath, int

nDPILevel)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 储存影像文件类型

1 : BMP

2 : JPG

3 : PNG

4 : PDF

5 : SVG

***pImageFilePath***

[in] 储存影像文件名

***nDPILevel***

[in] 保存影像分辨率等级。(0～1)

0：150 DPI

1：300 DPI

**Remark**

支持的影像文件类型总共有BMP/JPG/PNG/PDF/SVG等五种格式，pImageFilePath为储存影像的完整文件名。

**Return value**

参阅回传值定义

**PPL500\_SetCropImageMargin**

设定裁切L500签名板签名笔迹影像的水平和垂直边界。

int PPL500\_SetCropImageMargin(int nDeviceType, bool

bCropImage, int nCropMarginX, int nCropMarginY)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***bCropImage***

[in] 是否要裁切笔迹影像?

【是】：bCropImage = true

【否】：bCropImage = false

***nCropMarginX***

[in] 裁切笔迹的同时, 并额外保留的水平边界。

***nCropMarginY***

[in] 裁切笔迹的同时, 并额外保留的垂直边界。

**Remark**

1) nCropMarginX，nCropMarginY要有作用，必须bCropImage等于Ture。

2)需搭配PPL500\_SaveDrawingImage API才能储存裁切笔迹影像。

**Return value**

参阅回传值定义

**PPL500\_GetDeviceInfo**

获取L500签名板相关信息

int PPL500\_GetDeviceInfo(int nDeviceType, int nIndex, void\*

pOutput)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 设备信息序号

1 ：设备当前状态

2 ：签名设备的X的坐标范围

3 ：签名设备的Y的坐标范围

4 ：签名设备的最大压力值

5 ：签名设备的厂商名字

6 ：签名设备的设备名字

7 ：签名设备的SDK版本

***pOutput***

[out] 获取的设备信息

**Return Value**

参阅回传值定义

**PPL500\_AboutBox**

显示SDK版本信息。

void PPL500\_AboutBox(QWidget \*pParent)

**Parameter**

***pParent***

[in] 父窗口指针。

**Return value**

无。

**PPL500\_Base64Encode**

将L500签名板签名笔迹以指定的影像格式转换为Base64编码的字符串。

char\* PPL500\_Base64Encode (int nDeviceType, int nImageTypeID)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 欲编码的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

**Return value**

回传编码后的字符串, 若失败会回传NULL

**PPL500\_Base64EncodeFree**

释放Base64编码占用的内存，删除编码数据。

void PPL500\_Base64EncodeFree(int nDeviceType, char\* pData)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pData***

[in] Base64编码数据指针

**Return value**

无

**PPL500\_Base64Decode**

将Base64编码的字符串译码为原始笔迹数据。

int PPL500\_Base64Decode(int nDeviceType, int nImageTypeID,

const char\* pEncodedData, wchar\_t\* pImageFilePath)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 编码的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

***pEncodedData***

[in] 编码的数据字符串

***pImageFilePath***

[in] 译码数据储存完整文件名

**Return Value**

参阅回传值定义

**PPL500\_GetTotalDecodeInkPointsNumber**

获取解碼后得到的笔迹点数。

int PPL500\_GetTotalDecodeInkPointsNumber (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

回传笔迹数组点数量, 但如果笔迹点为空或者其他错误的话, 将会回传负的

错误值

**PPL500\_GetDecodeInkPointData**

获取译码后得到的单点笔迹数据。

int PPL500\_GetDecodeInkPointData (int nDeviceType, int nIndex,

int nTypeID)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 笔迹点在数组中的序号

***nTypeID***

[in] 取值种类。

1：取得X坐标

2：取得Y坐标

3：取得是否为每笔划结束点状态

**Return Value**

参阅回传值定义

**PPL500\_SetPenColor**

設定L500畫筆顏色。

int PPL500\_SetPenColor (int nDeviceType, int nR, int nG, int nB)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nR***

[in] 畫筆紅色比例值。 (0~255)

***nG***

[in] 畫筆綠色比例值。 (0~255)

***nB***

[in] 畫筆藍色比例值。 (0~255)

**Return value**

參閱回傳值定義

L501F界面：(可与L501G共享)

**PPL501F\_InitialDevice**

设定L501F签名板，初始化签名窗口以及设定签名画布大小。

bool PPL501F\_InitialDevice(int nDeviceType, QWidget

\*pParentWidget, int nX, int nY, int nWidth, int nHeight)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

设备ID定义：

enum DEVICE\_TYPE\_ID

{

PPL398\_DEVICE = 2,

PPL500\_DEVICE = 5,

PPL501F\_DEVICE = 7,

};

***pParentWidget***

[in] 父窗口指针。

***nX***

[in] 签名窗口起始点x坐标。

***nY***

[in] 签名窗口起始点y坐标。

***nWidth***

[in] 签名窗口宽度。

***nHeight***

[in] 签名窗口高度。

**Remark**

1. 可使用PPL501F\_InitialDevice去设定画布大小。
2. 如需重新设定新画布大小时，必须先使用PPL501F\_UnInitialDevice后，再使用PPL501F\_InitialDevice设置一次。

**Return Value**

参阅回传值定义

**PPL501F\_UnInitialDevice**

卸除L501F签名板，关闭签名窗口，释放资源。

int PPL501F\_UnInitialDevice(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

**Return value**

参阅回传值定义

**PPL501F\_OpenLCD**

打开L501F签名板LCD

int PPL501F\_OpenLCD(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***bOn***

【打开LCD】：bOn = true

【关闭LCD】：bOn = false

**Return Value**

参阅回传值定义

**PPL501F\_Clear**

清除L501F签名板画布上的笔迹纪录。

int PPL501F\_Clear (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

参阅回传值定义

**PPL501F\_GetHWPenID**

获取L501F签名板Pen ID

int PPL501F\_GetHWPenID(int nDeviceType, char\* pPenID, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pPenID***

[out] 输出Pen ID字符串指针

***nMaxCount***

[in] 输出Pen ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPL501F\_GetHWPadID**

获取L501F签名板Pad ID

int PPL501F\_GetHWPadID(int nDeviceType, char\* pPadID, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pPadID***

[out] 输出Pad ID字符串指针

***nMaxCount***

[in] 输出Pad ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPL501F\_GetHWVersionID**

获取L501F签名板的版本ID

int PPL501F\_GetHWVersionID(int nDeviceType, char\* pVersion, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pVersion***

[out] 输出版本ID字符串指针

***nMaxCount***

[in] 输出版本ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPL501F\_DisplayHWVersion**

开关L501F签名板上版本信息的显示

Int PPL501F\_DisplayHWVersion(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***bOn***

【显示】：bOn = true

【关闭】：bOn = false

**Return Value**

参阅回传值定义

**PPL501F\_SetPenStyle**

设定L501F签名板画笔种类。

int PPL501F\_SetPenStyle (int nDeviceType, bool bPressureOn)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

***bPressureOn***

[in] 画笔种类。

【压感笔】：bPressureOn = true

【定宽笔】：bPressureOn = false

**Return value**

参阅回传值定义

**PPL501F\_SetPenWidth**

设定L501F签名板画笔宽度。

int PPL501F\_SetPenWidth (int nDeviceType, int nPenWidth)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

***nPenWidth***

[in] 画笔宽度值。(1～10)

**Return value**

参阅回传值定义

**PPL501F\_SignPadEventCallback**

设定L501F签名板上按【确定】或【取消】时的回传函数

LONG PPL501F\_SignPadEventCallback)(int nDeviceType,

SIGNPAD\_EVENT\_CALLBACK pfnSignPadEventCallback)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pfnSignPadEventCallback***

[in] 按【确定】或【取消】时的回传函数地址

回传函数定义：

typedef void (\*SIGNPAD\_EVENT\_CALLBACK)(int nEventID);

**Remark**

当L501F签名板按【确定】或【取消】时会调用pfnSignPadEventCallback。

**Return Value**

参阅回传值定义

**PPL501F\_GetTotalInkPointsNumber**

取得L501F签名板目前笔迹数组点纪录数量。

int PPL501F\_GetTotalInkPointsNumber(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

回传笔迹数组点数量, 但如果笔迹点为空或者其他错误的话, 将会回传负的

错误值

**PPL501F\_GetInkPointData**

获取L501F签名板单点笔迹数据

Int PPL501F\_GetInkPointData(int nDeviceType, INKPOINTDATA&

currentInkPoint)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***currentInkPoint***

[out] 笔迹数据结构体。

笔迹数据结构体:

typedef struct \_INKPOINTDATA

{

int nIndex;

short nX;

short nY;

short nPressureValue;

bool bStrokeEnd;

} INKPOINTDATA, \*PINKPOINTDATA;

**Return Value**

参阅回传值定义

**PPL501F\_GetInkPointDataTimeInfo**

获取L501F签名板单点笔迹的时间信息。

char\* PPL501F\_GetInkPointTime(int nDeviceType, int nIndex)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 单点笔迹序号

**Return value**

单点笔迹时间信息字符串。

**PPL501F\_FreeInkPointDataTimeInfo**

释放L501F签名板单点笔迹时间信息字符串。

void PPL501F\_FreeInkPointDataTimeInfo(int nDeviceType, char\*

pInkPointDataTimeInfo)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pInkPointDataTimeInfo***

[in] 单点笔迹时间信息字符串

**Return value**

无。

**PPL501F\_SaveDrawingImage**

储存L501F签名板签名笔迹内容至支持的图像文件案。

int PPL501F\_SaveDrawingImage(int nDeviceType, int

nImageTypeID, wchar\_t\* pImageFilePath, int

nDPILevel)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 储存影像文件类型

1 : BMP

2 : JPG

3 : PNG

4 : PDF

5 : SVG

***pImageFilePath***

[in] 储存影像文件名

***nDPILevel***

[in] 保存影像分辨率等级。(0～1)

0：150 DPI

1：300 DPI

**Remark**

支持的影像文件类型总共有BMP/JPG/PNG/PDF/SVG等五种格式，pImageFilePath为储存影像的完整文件名。

**Return value**

参阅回传值定义

**PPL501F\_SetSaveClipImageMargin**

设定裁切L501F签名板签名笔迹影像的水平和垂直边界。

int PPL501F\_SetClipImageMargin(int nDeviceType, bool

bClipImage, int nClipMarginX, int nClipMarginY)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***bClipImage***

[in] 是否要裁切笔迹影像?

【是】：bClipImage = true

【否】：bClipImage = false

***nClipMarginX***

[in] 裁切笔迹的同时, 并额外保留的水平边界。

***nClipMarginY***

[in] 裁切笔迹的同时, 并额外保留的垂直边界。

**Remark**

1) nClipMarginX，nClipMarginY要有作用，必须bClipImage等于Ture。

2)需搭配PPL501F\_SaveDrawingImage API才能储存裁切笔迹影像。

**Return value**

参阅回传值定义

**PPL501F\_GetDeviceInfo**

获取L501F签名板相关信息

int PPL501F\_GetDeviceInfo(int nDeviceType, int nIndex, void\*

pOutput)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 设备信息序号

1 ：设备当前状态

2 ：签名设备的X的坐标范围

3 ：签名设备的Y的坐标范围

4 ：签名设备的最大压力值

5 ：签名设备的厂商名字

6 ：签名设备的设备名字

7 ：签名设备的SDK版本

***pOutput***

[out] 获取的设备信息

**Return Value**

参阅回传值定义

**PPL501F\_AboutBox**

显示SDK版本信息。

void PPL501F\_AboutBox(QWidget \*pParent)

**Parameter**

***pParent***

[in] 父窗口指针。

**Return value**

无。

**PPL501F\_Base64Encode**

将L501F签名板签名笔迹以指定的影像格式转换为Base64编码的字符串。

char\* PPL501F\_Base64Encode (int nDeviceType, int nImageTypeID)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 欲编码的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

**Return value**

回传编码后的字符串, 若失败会回传NULL

**PPL501F\_Base64EncodeFree**

释放Base64编码占用的内存，删除编码数据。

void PPL501F\_Base64EncodeFree(int nDeviceType, char\* pData)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pData***

[in] Base64编码数据指针

**Return value**

无

**PPL501F\_Base64Decode**

将Base64编码的字符串译码为原始笔迹数据。

int PPL501F\_Base64Decode(int nDeviceType, int nImageTypeID,

const char\* pEncodedData, wchar\_t\* pImageFilePath)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 编码的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

***pEncodedData***

[in] 编码的数据字符串

***pImageFilePath***

[in] 译码数据储存完整文件名

**Return Value**

参阅回传值定义

**PPL501F\_GetTotalDecodeInkPointsNumber**

获取解碼后得到的笔迹点数。

int PPL501F\_GetTotalDecodeInkPointsNumber (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

回传笔迹数组点数量, 但如果笔迹点为空或者其他错误的话, 将会回传负的

错误值

**PPL501F\_GetDecodeInkPointData**

获取译码后得到的单点笔迹数据。

int PPL501F\_GetDecodeInkPointData (int nDeviceType, int nIndex,

int nTypeID)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 笔迹点在数组中的序号

***nTypeID***

[in] 取值种类。

1：取得X坐标

2：取得Y坐标

3：取得是否为每笔划结束点状态

**Return Value**

参阅回传值定义

**PPL501F\_SetPenColor**

设定画笔颜色。

int PPL501F\_SetPenColor (int nDeviceType, int nR, int nG, int nB)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nR***

[in] 画笔红色比例值。 (0~255)

***nG***

[in] 画笔绿色比例值。 (0~255)

***nB***

[in] 画笔蓝色比例值。 (0~255)

**Return value**

参阅回传值定义

**PPL501F\_SetDisplayedFPImageAt**

设定指纹影像显示位置。

int PPL501F\_SetDisplayedFPImageAt(int nDeviceType, QWidget \*pParentWidget, int nCanvasWidth, int nCanvasHeight, int nFPLeft, int nFPTop, int nFPWidth, int nFPHeight);

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pParentWidget***

[in] 父窗口句柄。

***nCanvasWidth***

[in] 指纹影像显示在画布的宽度

***nCanvasHeight***

[in] 指纹影像显示在画布的高度

***nFPLeft***

[in] 指纹影像显示在父窗口或画布的起始点x坐标

***nFPTop***

[in] 指纹影像显示在父窗口或画布的起始点y坐标

***nFPWidth***

[in] 指纹影像显示在父窗口或画布的宽度

***nFPHeight***

[in] 指纹影像显示在父窗口或画布的高度

**Return value**

参阅回传值定义

**PPL501F\_GetFPImageBase64Data**

获取Base64编码的指纹影像。

char\* PPL501F\_GetFPImageBase64Data(int nDeviceType, int nImageTypeID, bool bFPImageOnly);

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 欲编码的影像文件格式。

1 : BMP

2 : JPG

3 : PNG

***bFPImageOnly***

[in] TRUE : 只取得指纹影像

FALSE : 取得包含指纹的父窗口或画布的影像

**Remark**

使用本API前，请先呼叫PPL501F\_SetDisplayedFPImageAt() API。

**Return value**

回传编码后的字符串, 若失败会回传NULL

**PPL501F\_FreeFPImageBase64Data**

释放编码占用的内存。

void PPL501F\_FreeFPImageBase64Data(int nDeviceType, char\* pFPImageBase64Data);

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pFPImageBase64Data***

[in] 编码数据储存缓冲区指针

**Return value**

无

LC700界面

**PPLC700\_InitialDevice**

设定LC700签名板，初始化签名窗口以及设定签名画布大小。

bool PPLC700\_InitialDevice(int nDeviceType, QWidget

\*pParentWidget, int nX, int nY, int nWidth, int nHeight)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

设备ID定义：

enum DEVICE\_TYPE\_ID

{

PPL398\_DEVICE = 2,

PPL500\_DEVICE = 5,

PPL501F\_DEVICE = 7,

PPLC700\_DEVICE = 3,

};

***pParentWidget***

[in] 父窗口指针。

***nX***

[in] 签名窗口起始点x坐标。

***nY***

[in] 签名窗口起始点y坐标。

***nWidth***

[in] 签名窗口宽度。

***nHeight***

[in] 签名窗口高度。

**Remark**

1. 可使用PPLC700\_InitialDevice去设定画布大小。
2. 如需重新设定新画布大小时，必须先使用PPLC700\_UnInitialDevice后，再使用PPLC700\_InitialDevice设置一次。

**Return Value**

参阅回传值定义

**PPLC700\_UnInitialDevice**

卸除LC700签名板，关闭签名窗口，释放资源。

int PPLC700\_UnInitialDevice(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

**Return value**

参阅回传值定义

**PPLC700\_Clear**

清除LC700签名板画布上的笔迹纪录。

int PPLC700\_Clear (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

参阅回传值定义

**PPLC700\_GetHWPadID**

获取LC700签名板Pad ID

int PPLC700\_GetHWPadID(int nDeviceType, char\* pPadID, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pPadID***

[out] 输出Pad ID字符串指针

***nMaxCount***

[in] 输出Pad ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPLC700\_GetHWVersionID**

获取LC700签名板的版本ID

int PPLC700\_GetHWVersionID(int nDeviceType, char\* pVersion, int

nMaxCount)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pVersion***

[out] 输出版本ID字符串指针

***nMaxCount***

[in] 输出版本ID字符串的最大长度

**Return Value**

参阅回传值定义

**PPLC700\_SetPenStyle**

设定LC700签名板画笔种类。

int PPLC700\_SetPenStyle (int nDeviceType, bool bPressureOn)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

***bPressureOn***

[in] 画笔种类。

【压感笔】：bPressureOn = true

【定宽笔】：bPressureOn = false

**Return value**

参阅回传值定义

**PPLC700\_SetPenWidth**

设定LC700签名板画笔宽度。

int PPLC700\_SetPenWidth (int nDeviceType, int nPenWidth)

**Parameter**

***nDeviceType***

[in] 设备类型ID。

***nPenWidth***

[in] 画笔宽度值。(1～10)

**Return value**

参阅回传值定义

**PPLC700\_SignPadEventCallback**

设定LC700签名板上按【确定】或【取消】时的回传函数

LONG PPLC700\_SignPadEventCallback)(int nDeviceType,

SIGNPAD\_EVENT\_CALLBACK pfnSignPadEventCallback)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pfnSignPadEventCallback***

[in] 按【确定】或【取消】时的回传函数地址

回传函数定义：

typedef void (\*SIGNPAD\_EVENT\_CALLBACK)(int nEventID);

**Remark**

当LC700签名板按【确定】或【取消】时会调用pfnSignPadEventCallback。

**Return Value**

参阅回传值定义

**PPLC700\_GetTotalInkPointsNumber**

取得LC700签名板目前笔迹数组点纪录数量。

int PPLC700\_GetTotalInkPointsNumber(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

回传笔迹数组点数量, 但如果笔迹点为空或者其他错误的话, 将会回传负的

错误值

**PPLC700\_GetInkPointData**

获取LC700签名板单点笔迹数据

Int PPLC700\_GetInkPointData(int nDeviceType, INKPOINTDATA&

currentInkPoint)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***currentInkPoint***

[out] 笔迹数据结构体。

笔迹数据结构体:

typedef struct \_INKPOINTDATA

{

int nIndex;

short nX;

short nY;

short nPressureValue;

bool bStrokeEnd;

} INKPOINTDATA, \*PINKPOINTDATA;

**Return Value**

参阅回传值定义

**PPLC700\_GetInkPointDataTimeInfo**

获取LC700签名板单点笔迹的时间信息。

char\* PPLC700\_GetInkPointTime(int nDeviceType, int nIndex)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 单点笔迹序号

**Return value**

单点笔迹时间信息字符串。

**PPLC700\_FreeInkPointDataTimeInfo**

释放LC700签名板单点笔迹时间信息字符串。

void PPLC700\_FreeInkPointDataTimeInfo(int nDeviceType, char\*

pInkPointDataTimeInfo)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pInkPointDataTimeInfo***

[in] 单点笔迹时间信息字符串

**Return value**

无。

**PPLC700\_SaveDrawingImage**

储存LC700签名板签名笔迹内容至支持的图像文件案。

int PPLC700\_SaveDrawingImage(int nDeviceType, int

nImageTypeID, wchar\_t\* pImageFilePath, int

nDPILevel)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 储存影像文件类型

1 : BMP

2 : JPG

3 : PNG

4 : PDF

5 : SVG

***pImageFilePath***

[in] 储存影像文件名

***nDPILevel***

[in] 保存影像分辨率等级。(0～1)

0：150 DPI

1：300 DPI

**Remark**

支持的影像文件类型总共有BMP/JPG/PNG/PDF/SVG等五种格式，pImageFilePath为储存影像的完整文件名。

**Return value**

参阅回传值定义

**PPLC700\_SetSaveClipImageMargin**

设定裁切LC700签名板签名笔迹影像的水平和垂直边界。

int PPLC700\_SetClipImageMargin(int nDeviceType, bool

bClipImage, int nClipMarginX, int nClipMarginY)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***bClipImage***

[in] 是否要裁切笔迹影像?

【是】：bClipImage = true

【否】：bClipImage = false

***nClipMarginX***

[in] 裁切笔迹的同时, 并额外保留的水平边界。

***nClipMarginY***

[in] 裁切笔迹的同时, 并额外保留的垂直边界。

**Remark**

1) nClipMarginX，nClipMarginY要有作用，必须bClipImage等于Ture。

2)需搭配PPLC700\_SaveDrawingImage API才能储存裁切笔迹影像。

**Return value**

参阅回传值定义

**PPLC700\_GetDeviceInfo**

获取LC700签名板相关信息

int PPLC700\_GetDeviceInfo(int nDeviceType, int nIndex, void\*

pOutput)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 设备信息序号

1 ：设备当前状态

2 ：签名设备的X的坐标范围

3 ：签名设备的Y的坐标范围

4 ：签名设备的最大压力值

5 ：签名设备的厂商名字

6 ：签名设备的设备名字

7 ：签名设备的SDK版本

***pOutput***

[out] 获取的设备信息

**Return Value**

参阅回传值定义

**PPLC700\_AboutBox**

显示SDK版本信息。

void PPLC700\_AboutBox(QWidget \*pParent)

**Parameter**

***pParent***

[in] 父窗口指针。

**Return value**

无。

**PPLC700\_Base64Encode**

将LC700签名板签名笔迹以指定的影像格式转换为Base64编码的字符串。

char\* PPLC700\_Base64Encode (int nDeviceType, int nImageTypeID)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 欲编码的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

**Return value**

回传编码后的字符串, 若失败会回传NULL

**PPLC700\_Base64EncodeFree**

释放Base64编码占用的内存，删除编码数据。

void PPLC700\_Base64EncodeFree(int nDeviceType, char\* pData)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***pData***

[in] Base64编码数据指针

**Return value**

无

**PPLC700\_Base64Decode**

将Base64编码的字符串译码为原始笔迹数据。

int PPLC700\_Base64Decode(int nDeviceType, int nImageTypeID,

const char\* pEncodedData, wchar\_t\* pImageFilePath)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nImageTypeID***

[in] 编码的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

***pEncodedData***

[in] 编码的数据字符串

***pImageFilePath***

[in] 译码数据储存完整文件名

**Return Value**

参阅回传值定义

**PPLC700\_GetTotalDecodeInkPointsNumber**

获取解碼后得到的笔迹点数。

int PPLC700\_GetTotalDecodeInkPointsNumber (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 设备类型ID

**Return value**

回传笔迹数组点数量, 但如果笔迹点为空或者其他错误的话, 将会回传负的

错误值

**PPLC700\_GetDecodeInkPointData**

获取译码后得到的单点笔迹数据。

int PPLC700\_GetDecodeInkPointData (int nDeviceType, int nIndex,

int nTypeID)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nIndex***

[in] 笔迹点在数组中的序号

***nTypeID***

[in] 取值种类。

1：取得X坐标

2：取得Y坐标

3：取得是否为每笔划结束点状态

**Return Value**

参阅回传值定义

**PPLC700\_SetPenColor**

设定LC700画笔颜色。

int PPLC700\_SetPenColor (int nDeviceType, int nR, int nG, int nB)

**Parameter**

***nDeviceType***

[in] 设备类型ID

***nR***

[in] 画笔红色比例值。 (0~255)

***nG***

[in] 画笔绿色比例值。 (0~255)

***nB***

[in] 画笔蓝色比例值。 (0~255)

**Return value**

参阅回传值定义

**回传值:**

|  |  |  |
| --- | --- | --- |
| 定义名称 | 定义值 | 描述 |
| ARI\_SUCCESS | 0 | 成功 |
| ARI\_FAIL | -1 | 失败 |
| ARI\_INVALID\_PARAMETERS | -2 | 传入参数错误 |
| ARI\_DRIVER\_INIT\_FAIL | -3 | 【驱动】初始化失败 |
| ARI\_DRIVER\_OPEN\_DEVICE\_FAIL | -4 | 【驱动】打开设备失败 |
| ARI\_DRIVER\_CLAIM\_INTERFACE\_FAIL | -5 | 【驱动】宣告接口失败 |
| ARI\_DRIVER\_CREATE\_THREAD\_FAIL | -6 | 【驱动】建立Thread失败 |
| ARI\_DRIVER\_SENT\_CMD\_FAIL | -7 | 【驱动】下命令失败 |
| ARI\_INK\_EMPTY | -8 | 笔迹数组点为空 |