* **API Reference**

L398介面：(可與E550/E560和L370/L398S共用)

[**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\_SetSaveClipImageMargin**](#PPL398_SetSaveClipImageMargin)

[**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介面：

[**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\_SetSaveClipImageMargin**](#PPL500_SetSaveClipImageMargin)

[**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介面：(可與L501G共用)

[**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介面: (可與E550/E560和L370/L398S共用)

**PPL398\_InitialDevice**

設定L398簽名板，初始化簽名視窗以及設定簽名畫布大小。

bool PPL398\_InitialDevice(int nDeviceType, QWidget

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

**Parameter**

***nDeviceType***

[in] 設備類型ID。

設備ID定義：

enum DEVICE\_TYPE\_ID

{

PPL398\_DEVICE = 2,

PPL500\_DEVICE = 5,

PPL501F\_DEVICE = 7,

PPLC700\_DEVICE = 3,

};

***pParentWidget***

[in] 父視窗指標。

***nX***

[in] 簽名視窗起始點x座標。

***nY***

[in] 簽名視窗起始點y座標。

***nWidth***

[in] 簽名視窗寬度。

***nHeight***

[in] 簽名視窗高度。

**Remark**

1. 可使用PPL398\_InitialDevice去設定畫布大小。
2. 如需重新設定新畫布大小時，必須先使用PPL398\_UnInitialDevice後，再使用PPL398\_InitialDevice設置一次。

**Return Value**

參閱回傳值定義

**PPL398\_UnInitialDevice**

卸載L398簽名板，關閉簽名視窗，釋放資源。

int PPL398\_UnInitialDevice(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

**Return value**

參閱回傳值定義

**PPL398\_OpenLCD**

打開L398簽名板LCD

int PPL398\_OpenLCD(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***bOn***

【打開LCD】：bOn = true

【關閉LCD】：bOn = false

**Return Value**

參閱回傳值定義

**PPL398\_Clear**

清除L398簽名板畫布上的筆跡紀錄。

int PPL398\_Clear (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

參閱回傳值定義

**PPL398\_GetHWPenID**

獲取L398簽名板Pen ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pPenID***

[out] 輸出Pen ID字串指標

***nMaxCount***

[in] 輸出Pen ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPL398\_GetHWPadID**

獲取L398簽名板Pad ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pPadID***

[out] 輸出Pad ID字串指標

***nMaxCount***

[in] 輸出Pad ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPL398\_GetHWVersionID**

獲取L398簽名板的版本ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pVersion***

[out] 輸出版本ID字串指標

***nMaxCount***

[in] 輸出版本ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPL398\_DisplayHWVersion**

開關L398簽名板上版本資訊的顯示

Int PPL398\_DisplayHWVersion(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***bOn***

【顯示】：bOn = true

【關閉】：bOn = false

**Return Value**

參閱回傳值定義

**PPL398\_SetPenStyle**

設定L398簽名板畫筆種類。

int PPL398\_SetPenStyle (int nDeviceType, bool bPressureOn)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

***bPressureOn***

[in] 畫筆種類。

【壓感筆】：bPressureOn = true

【定寬筆】：bPressureOn = false

**Return value**

參閱回傳值定義

**PPL398\_SetPenWidth**

設定L398簽名板畫筆寬度。

int PPL398\_SetPenWidth (int nDeviceType, int nPenWidth)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

***nPenWidth***

[in] 畫筆寬度值。(1～10)

**Return value**

參閱回傳值定義

**PPL398\_SignPadEventCallback**

設定L398簽名板上按【確定】或【取消】時的回傳函數

LONG PPL398\_SignPadEventCallback)(int nDeviceType,

SIGNPAD\_EVENT\_CALLBACK pfnSignPadEventCallback)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pfnSignPadEventCallback***

[in] 按【確定】或【取消】時的回傳函數位址

回傳函式定義：

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

**Remark**

當L398簽名板按【確定】或【取消】時會調用pfnSignPadEventCallback。

**Return Value**

參閱回傳值定義

**PPL398\_GetTotalInkPointsNumber**

取得L398簽名板目前筆跡陣列點紀錄數量。

int PPL398\_GetTotalInkPointsNumber(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

回傳筆跡陣列點數量, 但如果筆跡點為空或者其他錯誤的話, 將會回傳負的

錯誤值

**PPL398\_GetInkPointData**

獲取L398簽名板單點筆跡資料

Int PPL398\_GetInkPointData(int nDeviceType, INKPOINTDATA&

currentInkPoint)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***currentInkPoint***

[out] 筆跡資料結構體。

筆跡資料結構體:

typedef struct \_INKPOINTDATA

{

int nIndex;

short nX;

short nY;

short nPressureValue;

bool bStrokeEnd;

} INKPOINTDATA, \*PINKPOINTDATA;

**Return Value**

參閱回傳值定義

**PPL398\_GetInkPointDataTimeInfo**

獲取L398簽名板單點筆跡的時間資訊。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 單點筆跡序號

**Return value**

單點筆跡時間資訊字串。

**PPL398\_FreeInkPointDataTimeInfo**

釋放L398簽名板單點筆跡時間資訊字串。

void PPL398\_FreeInkPointDataTimeInfo(int nDeviceType, char\*

pInkPointDataTimeInfo)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pInkPointDataTimeInfo***

[in] 單點筆跡時間資訊字串

**Return value**

無。

**PPL398\_SaveDrawingImage**

儲存L398簽名板簽名筆跡內容至支援的影像檔案。

int PPL398\_SaveDrawingImage(int nDeviceType, int

nImageTypeID, wchar\_t\* pImageFilePath, int

nDPILevel)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 儲存影像檔案類型

1 : BMP

2 : JPG

3 : PNG

4 : PDF

5 : SVG

***pImageFilePath***

[in] 儲存影像檔案名稱

***nDPILevel***

[in] 保存影像解析度等級。(0～1)

0：150 DPI

1：300 DPI

**Remark**

支援的影像檔案類型總共有BMP/JPG/PNG/PDF/SVG等五種格式，pImageFilePath為儲存影像的完整檔案名稱。

**Return value**

參閱回傳值定義

**PPL398\_SetSaveClipImageMargin**

設定裁切L398簽名板簽名筆跡影像的水平和垂直邊界。

int PPL398\_SetSaveClipImageMargin(int nDeviceType, bool

bClipImage, int nClipMarginX, int nClipMarginY)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***bClipImage***

[in] 是否要裁切筆跡影像?

【是】：bClipImage = true

【否】：bClipImage = false

***nClipMarginX***

[in] 裁切筆跡的同時, 並額外保留的水平邊界。

***nClipMarginY***

[in] 裁切筆跡的同時, 並額外保留的垂直邊界。

**Remark**

1) nClipMarginX，nClipMarginY要有作用，必須bClipImage等於Ture。

2)需搭配PPL398\_SaveDrawingImage API才能儲存裁切筆跡影像。

**Return value**

參閱回傳值定義

**PPL398\_GetDeviceInfo**

獲取L398簽名板相關資訊

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

pOutput)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 設備資訊序號

1 ：設備當前狀態

2 ：簽名設備的X的座標範圍

3 ：簽名設備的Y的座標範圍

4 ：簽名設備的最大壓力值

5 ：簽名設備的廠商名字

6 ：簽名設備的設備名字

7 ：簽名設備的SDK版本

***pOutput***

[out] 獲取的設備資訊

**Return Value**

參閱回傳值定義

**PPL398\_AboutBox**

顯示SDK版本資訊。

void PPL398\_AboutBox(QWidget \*pParent)

**Parameter**

***pParent***

[in] 父視窗指標。

**Return value**

無。

**PPL398\_Base64Encode**

將L398簽名板簽名筆跡以指定的影像格式轉換為Base64編碼的字串。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 欲編碼的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

**Return value**

回傳編碼後的字串, 若失敗會回傳NULL

**PPL398\_Base64EncodeFree**

釋放Base64編碼佔用的記憶體，刪除編碼資料。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pData***

[in] Base64編碼資料指標

**Return value**

無

**PPL398\_Base64Decode**

將Base64編碼的字串解碼為原始筆跡資料。

int PPL398\_Base64Decode(int nDeviceType, int nImageTypeID,

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 編碼的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

***pEncodedData***

[in] 編碼的資料字串

***pImageFilePath***

[in] 解碼資料儲存完整檔案名稱

**Return Value**

參閱回傳值定義

**PPL398\_GetTotalDecodeInkPointsNumber**

獲取解碼後得到的筆跡點數。

int PPL398\_GetTotalDecodeInkPointsNumber (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

回傳筆跡陣列點數量, 但如果筆跡點為空或者其他錯誤的話, 將會回傳負的

錯誤值

**PPL398\_GetDecodeInkPointData**

獲取解碼後得到的單點筆跡資料。

int PPL398\_GetDecodeInkPointData (int nDeviceType, int nIndex,

int nTypeID)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 筆跡點在陣列中的序號

***nTypeID***

[in] 取值種類。

1：取得X座標

2：取得Y座標

3：取得是否為每筆劃結束點狀態

**Return Value**

參閱回傳值定義

**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介面

**PPL500\_InitialDevice**

設定L500簽名板，初始化簽名視窗以及設定簽名畫布大小。

bool PPL500\_InitialDevice(int nDeviceType, QWidget

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

**Parameter**

***nDeviceType***

[in] 設備類型ID。

設備ID定義：

enum DEVICE\_TYPE\_ID

{

PPL398\_DEVICE = 2,

PPL500\_DEVICE = 5,

PPL501F\_DEVICE = 7,

PPLC700\_DEVICE = 3,

};

***pParentWidget***

[in] 父視窗指標。

***nX***

[in] 簽名視窗起始點x座標。

***nY***

[in] 簽名視窗起始點y座標。

***nWidth***

[in] 簽名視窗寬度。

***nHeight***

[in] 簽名視窗高度。

**Remark**

1. 可使用PPL500\_InitialDevice去設定畫布大小。
2. 如需重新設定新畫布大小時，必須先使用PPL500\_UnInitialDevice後，再使用PPL500\_InitialDevice設置一次。

**Return Value**

參閱回傳值定義

**PPL500\_UnInitialDevice**

卸載L500簽名板，關閉簽名視窗，釋放資源。

int PPL500\_UnInitialDevice(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

**Return value**

參閱回傳值定義

**PPL500\_OpenLCD**

打開L500簽名板LCD

int PPL500\_OpenLCD(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***bOn***

【打開LCD】：bOn = true

【關閉LCD】：bOn = false

**Return Value**

參閱回傳值定義

**PPL500\_Clear**

清除L500簽名板畫布上的筆跡紀錄。

int PPL500\_Clear (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

參閱回傳值定義

**PPL500\_GetHWPenID**

獲取L500簽名板Pen ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pPenID***

[out] 輸出Pen ID字串指標

***nMaxCount***

[in] 輸出Pen ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPL500\_GetHWPadID**

獲取L500簽名板Pad ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pPadID***

[out] 輸出Pad ID字串指標

***nMaxCount***

[in] 輸出Pad ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPL500\_GetHWVersionID**

獲取L500簽名板的版本ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pVersion***

[out] 輸出版本ID字串指標

***nMaxCount***

[in] 輸出版本ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPL500\_DisplayHWVersion**

開關L500簽名板上版本資訊的顯示

Int PPL500\_DisplayHWVersion(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***bOn***

【顯示】：bOn = true

【關閉】：bOn = false

**Return Value**

參閱回傳值定義

**PPL500\_SetPenStyle**

設定L500簽名板畫筆種類。

int PPL500\_SetPenStyle (int nDeviceType, bool bPressureOn)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

***bPressureOn***

[in] 畫筆種類。

【壓感筆】：bPressureOn = true

【定寬筆】：bPressureOn = false

**Return value**

參閱回傳值定義

**PPL500\_SetPenWidth**

設定L500簽名板畫筆寬度。

int PPL500\_SetPenWidth (int nDeviceType, int nPenWidth)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

***nPenWidth***

[in] 畫筆寬度值。(1～10)

**Return value**

參閱回傳值定義

**PPL500\_SignPadEventCallback**

設定L500簽名板上按【確定】或【取消】時的回傳函數

LONG PPL500\_SignPadEventCallback)(int nDeviceType,

SIGNPAD\_EVENT\_CALLBACK pfnSignPadEventCallback)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pfnSignPadEventCallback***

[in] 按【確定】或【取消】時的回傳函數位址

回傳函式定義：

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

**Remark**

當L500簽名板按【確定】或【取消】時會調用pfnSignPadEventCallback。

**Return Value**

參閱回傳值定義

**PPL500\_GetTotalInkPointsNumber**

取得L500簽名板目前筆跡陣列點紀錄數量。

int PPL500\_GetTotalInkPointsNumber(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

回傳筆跡陣列點數量, 但如果筆跡點為空或者其他錯誤的話, 將會回傳負的

錯誤值

**PPL500\_GetInkPointData**

獲取L500簽名板單點筆跡資料

Int PPL500\_GetInkPointData(int nDeviceType, INKPOINTDATA&

currentInkPoint)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***currentInkPoint***

[out] 筆跡資料結構體。

筆跡資料結構體:

typedef struct \_INKPOINTDATA

{

int nIndex;

short nX;

short nY;

short nPressureValue;

bool bStrokeEnd;

} INKPOINTDATA, \*PINKPOINTDATA;

**Return Value**

參閱回傳值定義

**PPL500\_GetInkPointDataTimeInfo**

獲取L500簽名板單點筆跡的時間資訊。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 單點筆跡序號

**Return value**

單點筆跡時間資訊字串。

**PPL500\_FreeInkPointDataTimeInfo**

釋放L500簽名板單點筆跡時間資訊字串。

void PPL500\_FreeInkPointDataTimeInfo(int nDeviceType, char\*

pInkPointDataTimeInfo)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pInkPointDataTimeInfo***

[in] 單點筆跡時間資訊字串

**Return value**

無。

**PPL500\_SaveDrawingImage**

儲存L500簽名板簽名筆跡內容至支援的影像檔案。

int PPL500\_SaveDrawingImage(int nDeviceType, int

nImageTypeID, wchar\_t\* pImageFilePath, int

nDPILevel)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 儲存影像檔案類型

1 : BMP

2 : JPG

3 : PNG

4 : PDF

5 : SVG

***pImageFilePath***

[in] 儲存影像檔案名稱

***nDPILevel***

[in] 保存影像解析度等級。(0～1)

0：150 DPI

1：300 DPI

**Remark**

支援的影像檔案類型總共有BMP/JPG/PNG/PDF/SVG等五種格式，pImageFilePath為儲存影像的完整檔案名稱。

**Return value**

參閱回傳值定義

**PPL500\_SetSaveClipImageMargin**

設定裁切L500簽名板簽名筆跡影像的水平和垂直邊界。

int PPL500\_SetClipImageMargin(int nDeviceType, bool

bClipImage, int nClipMarginX, int nClipMarginY)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***bClipImage***

[in] 是否要裁切筆跡影像?

【是】：bClipImage = true

【否】：bClipImage = false

***nClipMarginX***

[in] 裁切筆跡的同時, 並額外保留的水平邊界。

***nClipMarginY***

[in] 裁切筆跡的同時, 並額外保留的垂直邊界。

**Remark**

1) nClipMarginX，nClipMarginY要有作用，必須bClipImage等於Ture。

2)需搭配PPL500\_SaveDrawingImage API才能儲存裁切筆跡影像。

**Return value**

參閱回傳值定義

**PPL500\_GetDeviceInfo**

獲取L500簽名板相關資訊

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

pOutput)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 設備資訊序號

1 ：設備當前狀態

2 ：簽名設備的X的座標範圍

3 ：簽名設備的Y的座標範圍

4 ：簽名設備的最大壓力值

5 ：簽名設備的廠商名字

6 ：簽名設備的設備名字

7 ：簽名設備的SDK版本

***pOutput***

[out] 獲取的設備資訊

**Return Value**

參閱回傳值定義

**PPL500\_AboutBox**

顯示SDK版本資訊。

void PPL500\_AboutBox(QWidget \*pParent)

**Parameter**

***pParent***

[in] 父視窗指標。

**Return value**

無。

**PPL500\_Base64Encode**

將L500簽名板簽名筆跡以指定的影像格式轉換為Base64編碼的字串。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 欲編碼的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

**Return value**

回傳編碼後的字串, 若失敗會回傳NULL

**PPL500\_Base64EncodeFree**

釋放Base64編碼佔用的記憶體，刪除編碼資料。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pData***

[in] Base64編碼資料指標

**Return value**

無

**PPL500\_Base64Decode**

將Base64編碼的字串解碼為原始筆跡資料。

int PPL500\_Base64Decode(int nDeviceType, int nImageTypeID,

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 編碼的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

***pEncodedData***

[in] 編碼的資料字串

***pImageFilePath***

[in] 解碼資料儲存完整檔案名稱

**Return Value**

參閱回傳值定義

**PPL500\_GetTotalDecodeInkPointsNumber**

獲取解碼後得到的筆跡點數。

int PPL500\_GetTotalDecodeInkPointsNumber (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

回傳筆跡陣列點數量, 但如果筆跡點為空或者其他錯誤的話, 將會回傳負的

錯誤值

**PPL500\_GetDecodeInkPointData**

獲取解碼後得到的單點筆跡資料。

int PPL500\_GetDecodeInkPointData (int nDeviceType, int nIndex,

int nTypeID)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 筆跡點在陣列中的序號

***nTypeID***

[in] 取值種類。

1：取得X座標

2：取得Y座標

3：取得是否為每筆劃結束點狀態

**Return Value**

參閱回傳值定義

**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介面：(可與L501G共用)

**PPL501F\_InitialDevice**

設定L501F簽名板，初始化簽名視窗以及設定簽名畫布大小。

bool PPL501F\_InitialDevice(int nDeviceType, QWidget

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

**Parameter**

***nDeviceType***

[in] 設備類型ID。

設備ID定義：

enum DEVICE\_TYPE\_ID

{

PPL398\_DEVICE = 2,

PPL500\_DEVICE = 5,

PPL501F\_DEVICE = 7,

PPLC700\_DEVICE = 3,

};

***pParentWidget***

[in] 父視窗指標。

***nX***

[in] 簽名視窗起始點x座標。

***nY***

[in] 簽名視窗起始點y座標。

***nWidth***

[in] 簽名視窗寬度。

***nHeight***

[in] 簽名視窗高度。

**Remark**

1. 可使用PPL501F\_InitialDevice去設定畫布大小。
2. 如需重新設定新畫布大小時，必須先使用PPL501F\_UnInitialDevice後，再使用PPL501F\_InitialDevice設置一次。

**Return Value**

參閱回傳值定義

**PPL501F\_UnInitialDevice**

卸載L501F簽名板，關閉簽名視窗，釋放資源。

int PPL501F\_UnInitialDevice(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

**Return value**

參閱回傳值定義

**PPL501F\_OpenLCD**

打開L501F簽名板LCD

int PPL501F\_OpenLCD(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***bOn***

【打開LCD】：bOn = true

【關閉LCD】：bOn = false

**Return Value**

參閱回傳值定義

**PPL501F\_Clear**

清除L501F簽名板畫布上的筆跡紀錄。

int PPL501F\_Clear (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

參閱回傳值定義

**PPL501F\_GetHWPenID**

獲取L501F簽名板Pen ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pPenID***

[out] 輸出Pen ID字串指標

***nMaxCount***

[in] 輸出Pen ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPL501F\_GetHWPadID**

獲取L501F簽名板Pad ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pPadID***

[out] 輸出Pad ID字串指標

***nMaxCount***

[in] 輸出Pad ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPL501F\_GetHWVersionID**

獲取L501F簽名板的版本ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pVersion***

[out] 輸出版本ID字串指標

***nMaxCount***

[in] 輸出版本ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPL501F\_DisplayHWVersion**

開關L501F簽名板上版本資訊的顯示

Int PPL501F\_DisplayHWVersion(int nDeviceType, bool bOn)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***bOn***

【顯示】：bOn = true

【關閉】：bOn = false

**Return Value**

參閱回傳值定義

**PPL501F\_SetPenStyle**

設定L501F簽名板畫筆種類。

int PPL501F\_SetPenStyle (int nDeviceType, bool bPressureOn)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

***bPressureOn***

[in] 畫筆種類。

【壓感筆】：bPressureOn = true

【定寬筆】：bPressureOn = false

**Return value**

參閱回傳值定義

**PPL501F\_SetPenWidth**

設定L501F簽名板畫筆寬度。

int PPL501F\_SetPenWidth (int nDeviceType, int nPenWidth)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

***nPenWidth***

[in] 畫筆寬度值。(1～10)

**Return value**

參閱回傳值定義

**PPL501F\_SignPadEventCallback**

設定L501F簽名板上按【確定】或【取消】時的回傳函數

LONG PPL501F\_SignPadEventCallback)(int nDeviceType,

SIGNPAD\_EVENT\_CALLBACK pfnSignPadEventCallback)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pfnSignPadEventCallback***

[in] 按【確定】或【取消】時的回傳函數位址

回傳函式定義：

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

**Remark**

當L501F簽名板按【確定】或【取消】時會調用pfnSignPadEventCallback。

**Return Value**

參閱回傳值定義

**PPL501F\_GetTotalInkPointsNumber**

取得L501F簽名板目前筆跡陣列點紀錄數量。

int PPL501F\_GetTotalInkPointsNumber(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

回傳筆跡陣列點數量, 但如果筆跡點為空或者其他錯誤的話, 將會回傳負的

錯誤值

**PPL501F\_GetInkPointData**

獲取L501F簽名板單點筆跡資料

Int PPL501F\_GetInkPointData(int nDeviceType, INKPOINTDATA&

currentInkPoint)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***currentInkPoint***

[out] 筆跡資料結構體。

筆跡資料結構體:

typedef struct \_INKPOINTDATA

{

int nIndex;

short nX;

short nY;

short nPressureValue;

bool bStrokeEnd;

} INKPOINTDATA, \*PINKPOINTDATA;

**Return Value**

參閱回傳值定義

**PPL501F\_GetInkPointDataTimeInfo**

獲取L501F簽名板單點筆跡的時間資訊。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 單點筆跡序號

**Return value**

單點筆跡時間資訊字串。

**PPL501F\_FreeInkPointDataTimeInfo**

釋放L501F簽名板單點筆跡時間資訊字串。

void PPL501F\_FreeInkPointDataTimeInfo(int nDeviceType, char\*

pInkPointDataTimeInfo)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pInkPointDataTimeInfo***

[in] 單點筆跡時間資訊字串

**Return value**

無。

**PPL501F\_SaveDrawingImage**

儲存L501F簽名板簽名筆跡內容至支援的影像檔案。

int PPL501F\_SaveDrawingImage(int nDeviceType, int

nImageTypeID, wchar\_t\* pImageFilePath, int

nDPILevel)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 儲存影像檔案類型

1 : BMP

2 : JPG

3 : PNG

4 : PDF

5 : SVG

***pImageFilePath***

[in] 儲存影像檔案名稱

***nDPILevel***

[in] 保存影像解析度等級。(0～1)

0：150 DPI

1：300 DPI

**Remark**

支援的影像檔案類型總共有BMP/JPG/PNG/PDF/SVG等五種格式，pImageFilePath為儲存影像的完整檔案名稱。

**Return value**

參閱回傳值定義

**PPL501F\_SetSaveClipImageMargin**

設定裁切L501F簽名板簽名筆跡影像的水平和垂直邊界。

int PPL501F\_SetClipImageMargin(int nDeviceType, bool

bClipImage, int nClipMarginX, int nClipMarginY)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***bClipImage***

[in] 是否要裁切筆跡影像?

【是】：bClipImage = true

【否】：bClipImage = false

***nClipMarginX***

[in] 裁切筆跡的同時, 並額外保留的水平邊界。

***nClipMarginY***

[in] 裁切筆跡的同時, 並額外保留的垂直邊界。

**Remark**

1) nClipMarginX，nClipMarginY要有作用，必須bClipImage等於Ture。

2)需搭配PPL501F\_SaveDrawingImage API才能儲存裁切筆跡影像。

**Return value**

參閱回傳值定義

**PPL501F\_GetDeviceInfo**

獲取L501F簽名板相關資訊

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

pOutput)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 設備資訊序號

1 ：設備當前狀態

2 ：簽名設備的X的座標範圍

3 ：簽名設備的Y的座標範圍

4 ：簽名設備的最大壓力值

5 ：簽名設備的廠商名字

6 ：簽名設備的設備名字

7 ：簽名設備的SDK版本

***pOutput***

[out] 獲取的設備資訊

**Return Value**

參閱回傳值定義

**PPL501F\_AboutBox**

顯示SDK版本資訊。

void PPL501F\_AboutBox(QWidget \*pParent)

**Parameter**

***pParent***

[in] 父視窗指標。

**Return value**

無。

**PPL501F\_Base64Encode**

將L501F簽名板簽名筆跡以指定的影像格式轉換為Base64編碼的字串。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 欲編碼的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

**Return value**

回傳編碼後的字串, 若失敗會回傳NULL

**PPL501F\_Base64EncodeFree**

釋放Base64編碼佔用的記憶體，刪除編碼資料。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pData***

[in] Base64編碼資料指標

**Return value**

無

**PPL501F\_Base64Decode**

將Base64編碼的字串解碼為原始筆跡資料。

int PPL501F\_Base64Decode(int nDeviceType, int nImageTypeID,

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 編碼的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

***pEncodedData***

[in] 編碼的資料字串

***pImageFilePath***

[in] 解碼資料儲存完整檔案名稱

**Return Value**

參閱回傳值定義

**PPL501F\_GetTotalDecodeInkPointsNumber**

獲取解碼後得到的筆跡點數。

int PPL501F\_GetTotalDecodeInkPointsNumber (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

回傳筆跡陣列點數量, 但如果筆跡點為空或者其他錯誤的話, 將會回傳負的

錯誤值

**PPL501F\_GetDecodeInkPointData**

獲取解碼後得到的單點筆跡資料。

int PPL501F\_GetDecodeInkPointData (int nDeviceType, int nIndex,

int nTypeID)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 筆跡點在陣列中的序號

***nTypeID***

[in] 取值種類。

1：取得X座標

2：取得Y座標

3：取得是否為每筆劃結束點狀態

**Return Value**

參閱回傳值定義

**PPL501F\_SetPenColor**

設定畫筆顏色。

int PPL501F\_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

**PPL501F\_SetDisplayedFPImageAt**

設定指紋影像顯示位置。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pParentWidget***

[in] 父視窗控制碼。

***nCanvasWidth***

[in] 指紋影像顯示在畫布的寬度

***nCanvasHeight***

[in] 指紋影像顯示在畫布的高度

***nFPLeft***

[in] 指紋影像顯示在父視窗或畫布的起始點x座標

***nFPTop***

[in] 指紋影像顯示在父視窗或畫布的起始點y座標

***nFPWidth***

[in] 指紋影像顯示在父視窗或畫布的寬度

***nFPHeight***

[in] 指紋影像顯示在父視窗或畫布的高度

**Return value**

參閱回傳值定義

**PPL501F\_GetFPImageBase64Data**

獲取Base64編碼的指紋影像。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 欲編碼的影像檔案格式。

1 : BMP

2 : JPG

3 : PNG

***bFPImageOnly***

[in] TRUE : 只取得指紋影像

FALSE : 取得包含指紋的父視窗或畫布的影像

**Remark**

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

**Return value**

回傳編碼後的字串, 若失敗會回傳NULL

**PPL501F\_FreeFPImageBase64Data**

釋放編碼佔用的記憶體。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pFPImageBase64Data***

[in] 編碼資料儲存緩衝區指標

**Return value**

無

LC700介面

**PPLC700\_InitialDevice**

設定LC700簽名板，初始化簽名視窗以及設定簽名畫布大小。

bool PPLC700\_InitialDevice(int nDeviceType, QWidget

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

**Parameter**

***nDeviceType***

[in] 設備類型ID。

設備ID定義：

enum DEVICE\_TYPE\_ID

{

PPL398\_DEVICE = 2,

PPL500\_DEVICE = 5,

PPL501F\_DEVICE = 7,

PPLC700\_DEVICE = 3,

};

***pParentWidget***

[in] 父視窗指標。

***nX***

[in] 簽名視窗起始點x座標。

***nY***

[in] 簽名視窗起始點y座標。

***nWidth***

[in] 簽名視窗寬度。

***nHeight***

[in] 簽名視窗高度。

**Remark**

1. 可使用PPLC700\_InitialDevice去設定畫布大小。
2. 如需重新設定新畫布大小時，必須先使用PPLC700\_UnInitialDevice後，再使用PPLC700\_InitialDevice設置一次。

**Return Value**

參閱回傳值定義

**PPLC700\_UnInitialDevice**

卸載LC700簽名板，關閉簽名視窗，釋放資源。

int PPLC700\_UnInitialDevice(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

**Return value**

參閱回傳值定義

**PPLC700\_Clear**

清除LC700簽名板畫布上的筆跡紀錄。

int PPLC700\_Clear (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

參閱回傳值定義

**PPLC700\_GetHWPadID**

獲取LC700簽名板Pad ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pPadID***

[out] 輸出Pad ID字串指標

***nMaxCount***

[in] 輸出Pad ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPLC700\_GetHWVersionID**

獲取LC700簽名板的版本ID

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

nMaxCount)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pVersion***

[out] 輸出版本ID字串指標

***nMaxCount***

[in] 輸出版本ID字串的最大長度

**Return Value**

參閱回傳值定義

**PPLC700\_SetPenStyle**

設定LC700簽名板畫筆種類。

int PPLC700\_SetPenStyle (int nDeviceType, bool bPressureOn)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

***bPressureOn***

[in] 畫筆種類。

【壓感筆】：bPressureOn = true

【定寬筆】：bPressureOn = false

**Return value**

參閱回傳值定義

**PPLC700\_SetPenWidth**

設定LC700簽名板畫筆寬度。

int PPLC700\_SetPenWidth (int nDeviceType, int nPenWidth)

**Parameter**

***nDeviceType***

[in] 設備類型ID。

***nPenWidth***

[in] 畫筆寬度值。(1～10)

**Return value**

參閱回傳值定義

**PPLC700\_SignPadEventCallback**

設定LC700簽名板上按【確定】或【取消】時的回傳函數

LONG PPLC700\_SignPadEventCallback)(int nDeviceType,

SIGNPAD\_EVENT\_CALLBACK pfnSignPadEventCallback)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pfnSignPadEventCallback***

[in] 按【確定】或【取消】時的回傳函數位址

回傳函式定義：

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

**Remark**

當LC700簽名板按【確定】或【取消】時會調用pfnSignPadEventCallback。

**Return Value**

參閱回傳值定義

**PPLC700\_GetTotalInkPointsNumber**

取得LC700簽名板目前筆跡陣列點紀錄數量。

int PPLC700\_GetTotalInkPointsNumber(int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

回傳筆跡陣列點數量, 但如果筆跡點為空或者其他錯誤的話, 將會回傳負的

錯誤值

**PPLC700\_GetInkPointData**

獲取LC700簽名板單點筆跡資料

Int PPLC700\_GetInkPointData(int nDeviceType, INKPOINTDATA&

currentInkPoint)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***currentInkPoint***

[out] 筆跡資料結構體。

筆跡資料結構體:

typedef struct \_INKPOINTDATA

{

int nIndex;

short nX;

short nY;

short nPressureValue;

bool bStrokeEnd;

} INKPOINTDATA, \*PINKPOINTDATA;

**Return Value**

參閱回傳值定義

**PPLC700\_GetInkPointDataTimeInfo**

獲取LC700簽名板單點筆跡的時間資訊。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 單點筆跡序號

**Return value**

單點筆跡時間資訊字串。

**PPLC700\_FreeInkPointDataTimeInfo**

釋放LC700簽名板單點筆跡時間資訊字串。

void PPLC700\_FreeInkPointDataTimeInfo(int nDeviceType, char\*

pInkPointDataTimeInfo)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pInkPointDataTimeInfo***

[in] 單點筆跡時間資訊字串

**Return value**

無。

**PPLC700\_SaveDrawingImage**

儲存LC700簽名板簽名筆跡內容至支援的影像檔案。

int PPLC700\_SaveDrawingImage(int nDeviceType, int

nImageTypeID, wchar\_t\* pImageFilePath, int

nDPILevel)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 儲存影像檔案類型

1 : BMP

2 : JPG

3 : PNG

4 : PDF

5 : SVG

***pImageFilePath***

[in] 儲存影像檔案名稱

***nDPILevel***

[in] 保存影像解析度等級。(0～1)

0：150 DPI

1：300 DPI

**Remark**

支援的影像檔案類型總共有BMP/JPG/PNG/PDF/SVG等五種格式，pImageFilePath為儲存影像的完整檔案名稱。

**Return value**

參閱回傳值定義

**PPLC700\_SetSaveClipImageMargin**

設定裁切LC700簽名板簽名筆跡影像的水平和垂直邊界。

int PPLC700\_SetClipImageMargin(int nDeviceType, bool

bClipImage, int nClipMarginX, int nClipMarginY)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***bClipImage***

[in] 是否要裁切筆跡影像?

【是】：bClipImage = true

【否】：bClipImage = false

***nClipMarginX***

[in] 裁切筆跡的同時, 並額外保留的水平邊界。

***nClipMarginY***

[in] 裁切筆跡的同時, 並額外保留的垂直邊界。

**Remark**

1) nClipMarginX，nClipMarginY要有作用，必須bClipImage等於Ture。

2)需搭配PPLC700\_SaveDrawingImage API才能儲存裁切筆跡影像。

**Return value**

參閱回傳值定義

**PPLC700\_GetDeviceInfo**

獲取LC700簽名板相關資訊

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

pOutput)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 設備資訊序號

1 ：設備當前狀態

2 ：簽名設備的X的座標範圍

3 ：簽名設備的Y的座標範圍

4 ：簽名設備的最大壓力值

5 ：簽名設備的廠商名字

6 ：簽名設備的設備名字

7 ：簽名設備的SDK版本

***pOutput***

[out] 獲取的設備資訊

**Return Value**

參閱回傳值定義

**PPLC700\_AboutBox**

顯示SDK版本資訊。

void PPLC700\_AboutBox(QWidget \*pParent)

**Parameter**

***pParent***

[in] 父視窗指標。

**Return value**

無。

**PPLC700\_Base64Encode**

將LC700簽名板簽名筆跡以指定的影像格式轉換為Base64編碼的字串。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 欲編碼的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

**Return value**

回傳編碼後的字串, 若失敗會回傳NULL

**PPLC700\_Base64EncodeFree**

釋放Base64編碼佔用的記憶體，刪除編碼資料。

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***pData***

[in] Base64編碼資料指標

**Return value**

無

**PPLC700\_Base64Decode**

將Base64編碼的字串解碼為原始筆跡資料。

int PPLC700\_Base64Decode(int nDeviceType, int nImageTypeID,

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

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nImageTypeID***

[in] 編碼的影像格式。

1 : BMP

2 : JPG

3 : PNG

4 : INK

***pEncodedData***

[in] 編碼的資料字串

***pImageFilePath***

[in] 解碼資料儲存完整檔案名稱

**Return Value**

參閱回傳值定義

**PPLC700\_GetTotalDecodeInkPointsNumber**

獲取解碼後得到的筆跡點數。

int PPLC700\_GetTotalDecodeInkPointsNumber (int nDeviceType)

**Parameter**

***nDeviceType***

[in] 設備類型ID

**Return value**

回傳筆跡陣列點數量, 但如果筆跡點為空或者其他錯誤的話, 將會回傳負的

錯誤值

**PPLC700\_GetDecodeInkPointData**

獲取解碼後得到的單點筆跡資料。

int PPLC700\_GetDecodeInkPointData (int nDeviceType, int nIndex,

int nTypeID)

**Parameter**

***nDeviceType***

[in] 設備類型ID

***nIndex***

[in] 筆跡點在陣列中的序號

***nTypeID***

[in] 取值種類。

1：取得X座標

2：取得Y座標

3：取得是否為每筆劃結束點狀態

**Return Value**

參閱回傳值定義

**PPLC700\_SetPenColor**

設定LC700畫筆顏色。

int PPLC700\_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

**回傳值:**

|  |  |  |
| --- | --- | --- |
| 定義名稱 | 定義值 | 描述 |
| ARI\_SUCCESS | 0 | 成功 |
| ARI\_FAIL | -1 | 失敗 |
| ARI\_INVALID\_PARAMETERS | -2 | 傳入參數錯誤 |
| ARI\_DRIVER\_INIT\_FAIL | -3 | 【驅動】初始化失敗 |
| ARI\_DRIVER\_OPEN\_DEVICE\_FAIL | -4 | 【驅動】打開設備失敗 |
| ARI\_DRIVER\_CLAIM\_INTERFACE\_FAIL | -5 | 【驅動】宣告介面失敗 |
| ARI\_DRIVER\_CREATE\_THREAD\_FAIL | -6 | 【驅動】建立Thread失敗 |
| ARI\_DRIVER\_SENT\_CMD\_FAIL | -7 | 【驅動】下命令失敗 |
| ARI\_INK\_EMPTY | -8 | 筆跡陣列點為空 |