

# Tablet USB&RF User Guide

## Interface Specification 1.4

Revised March, 2024

**Target Platform:** Windows

### 1、设备初始化

User Interface	<code>signInitialize</code> 只有初始化成功，才能进行后续操作
Declare	<code>int signInitialize ();</code>
Parameters	/
Return Values	If the function succeeds, the return value is <code>ERR_OK</code> . If the function fails, the return value is <code>ERR_DEVICE_OPENFAIL</code> .
Remarks	

### 2、释放资源

User Interface	<code>signClean</code>
Declare	<code>int signClean ();</code>
Parameters	/
Return Values	资源释放成功，返回 <code>ERR_OK</code> .
Remarks	

### 3、获取设备状态

User Interface	<code>signGetDeviceStatus</code> Find available tablet devices.
Declare	<code>int signGetDeviceStatus ();</code>
Parameters	/
Return Values	设备可用时，返回 <code>ERR_OK</code> . 无找到可用设备时，返回 <code>ERR_DEVICE_NOTFOUND</code> .
Remarks	

### [4]打开签名手写板

User Interface	<code>signOpenDevice</code>
Declare	<code>int signOpenDevice ();</code>
Parameters	/

Return Values	设备打开成功，返回 <a href="#">ERR_OK</a> . 无法打开设备，返回 <a href="#">ERR_DEVICE_XXX</a> . See the definition table for more details.
Remarks	

#### [5] 关闭签名手写板

User Interface	<a href="#">signCloseDevice</a>
Declare	int signCloseDevice ();
Parameters	/
Return Values	The return value is <a href="#">ERR_OK</a> .
Remarks	

#### [6] 获取手写板信息

User Interface	<a href="#">signGetDeviceInfo</a>
Declare	int signGetDeviceInfo (TABLET_DEVICEINFO* lpDeviceInfo);
Parameters	lpDeviceInfo [in] Pointer to the <a href="#">TABLET_DEVICEINFO</a> structure that receives information about the device.
Return Values	If the function succeeds, the return value is <a href="#">ERR_OK</a> . If the function fails, the return value is <a href="#">ERR_DEVICE_XXX</a> . See the definition table for more details.
Remarks	<p><a href="#">TABLET_DEVICEINFO</a></p> <pre>typedef struct tagAXI S {     unsigned long                                 min;     unsigned long    max; }AXIS, *PAXIS;  //device information typedef struct tagTABLET_DEVICEINFO {     AXIS    axisX;                //X 范围     AXIS    axisY;                //Y 范围     unsigned long    pressure;    //压感最大值     char    vendor[32];           //产商名称     char    product[32];          //产品名称 }</pre>

	<pre> unsigned long    version;                //dll 的版本  char    serialnum[32];                //设备序列号 }TABLET_DEVICEINFO,*PTABLET_DEVICEINFO; </pre>

#### [7]注册手写数据回调函数

User Interface	signRegisterDataCallBack
Declare	int signRegisterDataCallBack (PACKDATAPROC lpPackDataProc);
Parameters	<p>lpPackDataProc 指向回调函数</p> <p>The <a href="#">PACKDATAPROC</a> type defines a pointer to this callback function.</p>
Return Values	<p>If the function succeeds, the return value is <a href="#">ERR_OK</a>.</p> <p>If the function fails, the return value is <a href="#">ERR_INVALIDPARAM</a>.</p>
Remarks	<p><a href="#">PACKDATAPROC</a> typedef</p> <pre> int (CALLBACK *                                 (TAPROC) (PDATAPACKET    pktObj);  //    PDATAPACKET    structure typedef struct tagDATAPACKET {     EventType            eventtype;        //事件类型    4     unsigned short    physical_key;        //物理按键    2     unsigned short    virtual_key;        //虚拟按键    2     KeyStatus            keystatus;        //按键状态    4     PenStatus            penstatus;        //笔状态    4     unsigned short    x;                //x 坐标    2     unsigned short    y;                //y 坐标    2     unsigned short    pressure;        //压感    2     short    wheel_direction;        //滚轮    2 </pre>

	<pre>unsigned short  button;      //笔身按键      2  char tiltX;      //tilt X  char tiltY;      //tilt Y  }DATAPACKET, *PDATAPACKET;  enum EventType {     EventType_Pen = 1,     EventType_Key = 2,     EventType_Eraser = 3,     EventType_Wheel = 4,     EventType_ALL = 0xfe };  enum PenStatus {     PenStatus_Hover,     PenStatus_Down,     PenStatus_Move,     PenStatus_Up,</pre>
	<pre>PenStatus_Leave };  enum KeyStatus {     KeyStatus_Up,     KeyStatus_Down };  When the eventtype is EventType_Key, if physical_key is greater than 0, gets the physical key mask status bool Pkey_01 = physical_key&amp;(0x1&lt;&lt;0); bool Pkey_02 = physical_key&amp;(0x1&lt;&lt;1); bool Pkey_03 = physical_key&amp;(0x1&lt;&lt;2); bool Pkey_04 = physical_key&amp;(0x1&lt;&lt;3); ...  if virtual _key is greater than 0, get the key number int Vkey = virtual_key;</pre>

#### 4、注销手写笔数据回调函数

User Interface	signUnregisterDataCallBack
Declare	void signUnregisterDataCallBack (long handler);
Parameters	handler 指向注册手写数据回调函数句柄
Return Values	/
Remarks	

#### [9]注册状态回调函数

User Interface	signRegisterDevNotifyCallBack
Declare	int signRegisterDevNotifyCallBack (DEVNOTIFYPROC lpDevNotifyProc);
Parameters	lpDevNotifyProc 指向回调函数
Return Values	If the function succeeds, the return value is <b>ERR_OK</b> . If the function fails, the return value is <b>ERR_INVALIDPARAM</b> .
Remarks	<b>DEVNOTIFYPROC</b> <pre>typedef          int (CALLBACK * DEVNOTIFYPROC) (PENVPACKET pktObj);  typedef struct tagSTATUSPACKET {     INT          penAlive;     INT          penBattery;     INT          status; //0 DISCONNECTED 1 CONNECTED 2                   SLEEP 3 AWAKE 4 BATTERY }STATUSPACKET,* PSTATUSPACKET;</pre>

#### [10]注销状态回调函数

User Interface	signUnregisterDevNotifyCallBack
Declare	void signUnregisterDevNotifyCallBack (long handler);
Parameters	handler 指向注册时状态回调函数句柄
Return Values	/
Remarks	

#### [11]注册触摸回调函数

User Interface	signRegisterTouchCallBack
Declare	void signRegisterTouchCallBack (long handler);
Parameters	lpDevNotifyProc [in]Pointer to the callback function.

	The TOUCHPROC type defines a pointer to this callback function.
<b>Return Values</b>	If the function succeeds, the return value is ERR_OK. If the function fails, the return value is ERR_INVALIDPARAM.
<b>Remarks</b>	<p>TOUCHPROC</p> <pre>typedef          int (_stdcall * TOUCHPROC) (TOUCHDATA td);  enum TouchStatus {     TouchStatus_Up,     TouchStatus_Down,     TouchStatus_Move };  typedef struct tagTOUCHDATA {     TouchStatus      status[10];     unsigned int     x[10];     unsigned int     y[10]; } TOUCHDATA, *PTOUCHDATA;</pre>

#### [12]注销触摸回调函数

<b>User Interface</b>	<p>signUnregisterTouchCallBack</p> <p>Unregister the touch data callback function.</p>
<b>Declare</b>	void signUnregisterTouchCallBack (long handler);
<b>Parameters</b>	<p>handler</p> <p>[in] Handle returned by the signRegisterTouchCallBackcallback function.</p>
<b>Return Values</b>	/
<b>Remarks</b>	

#### [13]设备切换模式

<b>User Interface</b>	signChangeDeviceMode
<b>Declare</b>	int signChangeDeviceMode(int mode);
<b>Parameters</b>	mode 模式值

<b>Return Values</b>	<p>If the function succeeds, the return value is <code>ERR_OK</code>.</p> <p>If the function fails, the return value is <code>ERR_</code> <code>ERR_NOSUPPORTED</code> or <code>ERR_DEVICE_OPENFAIL</code>.</p>
<b>Remarks</b>	<pre>//run mode enum DeviceRunMode {     DeviceRunMode_Mouse = 1,        //鼠标模式     DeviceRunMode_Pen = 2,          //笔模式     DeviceRunMode_MousePen = 3,     //鼠标和笔模式     DeviceRunMode_StdPen = 4        //standard pen };</pre>

#### [14]获取签名区域

<b>User Interface</b>	<p><code>signGetScreenRect</code></p> <p>The function retrieves the dimensions of the bounding rectangle of the signaturescreen.</p>
<b>Declare</b>	<code>int signGetScreenRect(RECT* lpRect);</code>
<b>Parameters</b>	<p><code>lpRect</code></p> <p>[in] Pointer to a RECT structure that receives the screen coordinates.</p>
<b>Return Values</b>	<p>If the function succeeds, the return value is <code>ERR_OK</code>.</p> <p>If the function fails, the return value is <code>ERR_</code> <code>ERR_NOSUPPORTED</code>.</p>
<b>Remarks</b>	

#### [15]设置笔触为鼠标控制

<b>User Interface</b>	<code>signMouseControl</code>
<b>Declare</b>	<code>bool signMouseControl(bool bControlled);</code>
<b>Parameters</b>	<p><code>bControlled =true</code> 时，笔触转为鼠标显示；</p> <p><code>bControlled =false</code> 时，笔触不为鼠标显示。</p>
<b>Return Values</b>	Returns the current mouse available status.
<b>Remarks</b>	

#### [16]鼠标扩展屏显示

<b>User Interface</b>	<p><code>signSetExtendDisplay</code></p> <p>The function changes the mouse to extend mode.</p>
<b>Declare</b>	<code>void signSetExtendDisplay(bool bExtendDisplay);</code>
<b>Parameters</b>	<p><code>bExtendDisplay</code></p> <p>[in] If this parameter is true, the mouse show on the extend screen.</p> <p>If the parameter is false, the mouse show on the primary screen.</p>

Return Values	/
Remarks	

#### [17]设备旋转角度设置

User Interface	<b>signRotateMode</b> The function changes the running rotation angle of the device for pen.
Declare	int signRotateMode(int mode);
Parameters	mode [in] rotation smode
Return Values	If the function succeeds, the return value is <b>ERR_OK</b> . If the function fails, the return value is <b>ERR_NOSUPPORTED</b>
Remarks	Set the screen rotation angle. The default angle is 0 degrees. [parameter] angle: the values can be specified as 0, 90, 180, 270. Mode=0; //顺时针旋转 0 度 Mode=1; //顺时针旋转 90 度 Mode=2; //顺时针旋转 180 度 Mode=3; //顺时针旋转 270 度

#### Constant Definition 错误定义

<b>ERR_OK</b>	0	Is ok.
<b>ERR_DEVICE_NOTFOUND</b>	-1	无法找到可用设备
<b>ERR_DEVICE_OPENFAIL</b>	-2	执行失败
<b>ERR_DEVICE_NOTCONNECTED</b>	-3	无连接
<b>ERR_INVALIDPARAM</b>	-101	无效参数
<b>ERR_NOSUPPORTED</b>	-102	模式不支持



**Support Device List**

Type	Pen	Physics Key	Virtual key	Display	Touch
CS01	yes	no	no	no	no
CS03	yes	no	no	no	no
EX07	yes	yes	no	no	no
EX08	yes	yes	no	no	no
UG05	yes	no	yes	yes	no
TabletA5	yes	no	yes	no	no
ET0A4KW	yes	yes	yes	no	no
101NF	yes	no	no	yes	no
101TF	yes	no	no	yes	yes
UD13	yes	yes	no	yes	no