# Tablet USB&RF User Guide

## Interface Specification 1.3

Revised September, 2020

**Target Platform:** Windows

### 1、设备初始化

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

### 2、 释放资源

| | |
|---|---|
| **User Interface** | signClean |
| **Declare** | int signClean (); |
| **Parameters** | / |
| **Return Values** | 资源释放成功，返回 ERR_OK. |
| **Remarks** | |

### 3、获取设备状态

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

**[4]打开签名手写板**

| | |
|---|---|
| **User Interface** | signOpenDevice |

| Declare | int signOpenDevice (); |
|---|---|
| Parameters | / |
| Return Values | 设备打开成功，返回 ERR_OK.<br>无法打开设备，返回 ERR_DEVICE_XXX.<br>See the definition table for more details. |
| Remarks | |

**[5] 关闭签名手写板**

| User Interface | signCloseDevice |
|---|---|
| Declare | int signCloseDevice (); |
| Parameters | / |
| Return Values | The return value is ERR_OK. |
| Remarks | |

**[6]获取手写板信息**

| User Interface | signGetDeviceInfo |
|---|---|
| Declare | int signGetDeviceInfo (TABLET_DEVICEINFO* lpDeviceInfo); |
| Parameters | lpDeviceInfo<br>[in] Pointer to the TABLET_DEVICEINFO structure<br>that receives information about the device. |
| Return Values | If the function succeeds, the return value is ERR_OK.<br>If the function fails, the return value is ERR_DEVICE_XXX.<br>See the definition table for more details. |
| Remarks | TABLET_DEVICEINFO<br>`typedef struct  tagAXIS`<br>`{`<br>`    unsigned long    min;`<br>`    unsigned long    max;`<br>`}AXIS,*PAXIS;`<br><br>`//device information`<br>`typedef struct tagTABLET_DEVICEINFO`<br>`{`<br>`    AXIS    axisX;                    //X 范围`<br>`    AXIS    axisY;                    //Y 范围`<br>`    unsigned long    pressure;        //压感最大值`<br>`    char    vendor[32];               //产商名称`<br>`    char    product[32];              //产品名称`<br>`    unsigned long    version;         //dll 的版本`<br>`    char    serialnum[32];            //设备序列号`<br>`}TABLET_DEVICEINFO,*PTABLET_DEVICEINFO;` |
| | |

**[7]注册手写数据回调函数**

| User Interface | signRegisterDataCallBack |
|---|---|
| Declare | int signRegisterDataCallBack (PACKDATAPROC lpPackDataProc); |
| Parameters | lpPackDataProc **指向回调函数**<br>The PACKDATAPROC type defines a pointer to this callback function. |
| Return Values | If the function succeeds, the return value is ERR_OK.<br>If the function fails, the return value is ERR_INVALIDPARAM. |
| Remarks | PACKDATAPROC<br>typedef int (CALLBACK * PACKDATAPROC)(PDATAPACKET pktObj);<br><br>// PDATAPACKET structure<br>typedef struct tagDATAPACKET<br>{<br>EventType eventtype; //事件类型 4<br>unsigned short physical_key; //物理按键 2<br>unsigned short virtual_key; //虚拟按键 2<br>KeyStatus keystatus; //按键状态 4<br>PenStatus penstatus; //笔状态 4<br>unsigned short x; //x坐标 2<br>unsigned short y; //y坐标 2<br>unsigned short pressure; //压感 2<br>short wheel_direction; //滚轮 2<br>unsigned short button; //笔身按键 2<br>}DATAPACKET, *PDATAPACKET;<br><br>enum EventType<br>{<br>EventType_Pen = 1,<br>EventType_Key = 2,<br>EventType_Eraser = 3,<br>EventType_Wheel = 4,<br>EventType_ALL = 0xfe<br>};<br><br>enum PenStatus<br>{<br>PenStatus_Hover,<br>PenStatus_Down,<br>PenStatus_Move,<br>PenStatus_Up, |

| | |
|---|---|
| | PenStatus_Leave<br>};<br><br>enum KeyStatus<br>{<br>KeyStatus_Up,<br>KeyStatus_Down<br>};<br><br>When the eventtype is EventType_Key, if physical_key is greater than 0, gets the physical key mask status<br>bool Pkey_01 = physical_key&(0x1<<0);<br>bool Pkey_02 = physical_key&(0x1<<1);<br>bool Pkey_03 = physical_key&(0x1<<2);<br>bool Pkey_04 = physical_key&(0x1<<3);<br>...<br>if virtual _key is greater than 0, get the key number<br>int Vkey = virtual_key; |
| | |

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 ERR_OK.<br>If the function fails, the return value is ERR_INVALIDPARAM. |
| Remarks | DEVNOTIFYPROC<br>typedef int (CALLBACK * DEVNOTIFYPROC)(PENVPACKET pktObj);<br><br>typedef struct tagSTATUSPACKET<br>{<br>INT penAlive;<br>INT penBattery;<br>INT status; //0 DISCONNECTED 1 CONNECTED 2 SLEEP 3 AWAKE 4 BATTERY<br>}STATUSPACKET,* PSTATUSPACKET; |

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

**[12]注销触摸回调函数**

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

**[13]设备切换模式**

| | |
|---|---|
| **User Interface** | signChangeDeviceMode |
| **Declare** | int signChangeDeviceMode(int mode); |
| **Parameters** | mode 模式值 |
| **Return Values** | If the function succeeds, the return value is ERR_OK.<br>If the function fails, the return value is ERR_ ERR_NOSUPPORTED or ERR_DEVICE_OPENFAIL. |
| **Remarks** | //run mode<br>enum DeviceRunMode<br>{<br>DeviceRunMode_Mouse = 1, //鼠标模式<br>DeviceRunMode_Pen = 2, //笔模式<br>DeviceRunMode_MousePen = 3, //鼠标和笔模式<br>DeviceRunMode_StdPen = 4 //standard pen<br>}; |

**[14]获取签名区域**

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

**[15]设置笔触为鼠标控制**

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

**[16]鼠标扩展屏显示**

| | |
|---|---|
| **User Interface** | signSetExtendDisplay<br><br>The function changes the mouse to extend mode. |
| **Declare** | void signSetExtendDisplay(bool bExtendDisplay); |
| **Parameters** | bExtendDisplay<br>[in] If this parameter is true, the mouse show on the extend screen.<br>If the parameter is false, the mouse show on the primary screen. |
| **Return Values** | / |
| **Remarks** | |

**[17]设备旋转角度设置**

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

## Constant Definition 错误定义

| | | |
|---|---|---|
| ERR_OK | 0 | Is ok. |
| ERR_DEVICE_NOTFOUND | -1 | 无法找到可用设备 |
| ERR_DEVICE_OPENFAIL | -2 | 执行失败 |
| ERR_DEVICE_NOTCONNECTED | -3 | 无连接 |
| ERR_INVALIDPARAM | -101 | 无效参数 |
| ERR_NOSUPPORTED | -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 |