Tablet USB&RF User Guide

Interface Specification 1.3

Revised July, 2020

Target Platform: Windows

[1]

User Interface	signInitialize Initialize the device application environment.
Declare	int signInitialize ();
Parameters	/
Return Values	If the function succeeds, the return value is ERR_OK. If the function fails, the return value is ERR_DEVICE_OPENFAIL.
Remarks	

[2]

User Interface	signClean Close the device application environment.
Declare	int signClean ();
Parameters	/
Return Values	The return value is ERR_OK.
Remarks	

[3]

User Interface	signGetDeviceStatus Find available tablet devices.
Declare	int signGetDeviceStatus ();
Parameters	/
Return Values	If the function succeeds, the return value is ERR_OK. If the function fails, the return value is ERR_DEVICE_NOTFOUND.
Remarks	

[4]

User Interface	signOpenDevice Open a usable tablet device.
Declare	int signOpenDevice ();
Parameters	/
Return Values	If the function succeeds, the return value is ERR_OK. If the function fails, the return value is ERR_DEVICE_XXX. See the definition table for more details.
Remarks	

[5]

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

[6]

User Interface	signGetDeviceInfo Get information about the device.
Declare	int signGetDeviceInfo (TABLET_DEVICEINFO* lpDeviceInfo);
Parameters	lpDeviceInfo [in] Pointer to the TABLET_DEVICEINFO structure that receives information about the device.
Return Values	If the function succeeds, the return value is ERR_OK. If the function fails, the return value is ERR_DEVICE_XXX. See the definition table for more details.

Remarks	TABLET_DEVICEINFO
	typedef struct tagAXIS
	{
	unsigned long min;
	unsigned long max;
	}AXIS,*PAXIS;
	
	//device information
	typedef struct tagTABLET_DEVICEINFO
	{

	AXIS axisX; //X range
	AXIS axisY; //Y range
	unsigned long pressure; //pressure
	char vendor[32]; //verdor name
	char product[32]; //product name
	unsigned long version; //driver version
	char serialnum[32]; //serial number
	}TABLET_DEVICEINFO,*PTABLET_DEVICEINFO;

[7]

User Interface	signRegisterDataCallBack Register a pen data callback function.
Declare	int signRegisterDataCallBack (PACKDATAPROC lpPackDataProc);
Parameters	lpPackDataProc [in]Pointer to the callback function. The PACKDATAPROC type defines a pointer to this callback function.
Return Values	If the function succeeds, the return value is ERR_OK. If the function fails, the return value is ERR_INVALIDPARAM.

Remarks	PACKDATAPROC
	typedef int (CALLBACK * PACKDATAPROC)(PDATAPACKET pktObj);
	
	// PDATAPACKET structure
	typedef struct tagDATAPACKET
	{
	EventType eventtype; //event type 4
	unsigned short physical_key; //physical key 2
	unsigned short virtual_key; //virtual key2
	KeyStatus keystatus; //key status 4
	PenStatus penstatus; //pen status 4
	unsigned short x; //x 2
	unsigned short y; //y 2
	unsigned short pressure; //pressure 2
	short wheel_direction; //wheel 2
	unsigned short button; //pen button 2
	}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,
	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&(0x1<<0);
	bool Pkey_02 = physical_key&(0x1<<1);
	bool Pkey_03 = physical_key&(0x1<<2);
	bool Pkey_04 = physical_key&(0x1<<3);
	...
	if virtual _key is greater than 0, get the key number
	int Vkey = virtual_key;

[8]

User Interface	signUnregisterDataCallBack Unregister the pen data callback function.
Declare	void signUnregisterDataCallBack (long handler);
Parameters	handler [in] Handle returned by the bleRegisterDataCallBack callback function.
Return Values	/
Remarks	

[9]

User Interface	signRegisterDevNotifyCallBack Register a status callback function.
Declare	int signRegisterDevNotifyCallBack (DEVNOTIFYPROC lpDevNotifyProc);
Parameters	lpDevNotifyProc [in]Pointer to the callback function. The DEVNOTIFYPROC type defines a pointer to this callback function.
Return Values	If the function succeeds, the return value is ERR_OK. If the function fails, the return value is ERR_INVALIDPARAM.

Remarks	DEVNOTIFYPROC
	`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;`

[10]

User Interface	signUnregisterDevNotifyCallBack Unregister the status callback function.
Declare	void signUnregisterDevNotifyCallBack (long handler);
Parameters	handler [in] Handle returned by the bleRegisterDataCallBack callback function.
Return Values	/
Remarks	

[11]

User Interface	signRegisterTouchCallBack Register a touch data callback function.
Declare	void signRegisterTouchCallBack (long handler);
Parameters	lpDevNotifyProc [in]Pointer to the callback function. The TOUCHPROC type defines a pointer to this callback function.
Return Values	If the function succeeds, the return value is ERR_OK. If the function fails, the return value is ERR_INVALIDPARAM.

Remarks	TOUCHPROC
	`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;`

[12]

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

[13]

User Interface	signChangeDeviceMode The function changes the running mode of the device.
Declare	int signChangeDeviceMode(int mode);
Parameters	mode [in] running mode
Return Values	If the function succeeds, the return value is ERR_OK. If the function fails, the return value is ERR_ ERR_NOSUPPORTED or ERR_DEVICE_OPENFAIL.

Remarks	//run mode
	enum DeviceRunMode
	{
	DeviceRunMode_Mouse = 1, //system mouse
	DeviceRunMode_Pen = 2, //pen data
	DeviceRunMode_MousePen = 3, //system mouse and pen data
	DeviceRunMode_StdPen = 4 //standard pen
	};

[14]

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

[15]

User Interface	signMouseControl The function enables or disables the virtual mouse.
Declare	bool signMouseControl(bool bControlled);
Parameters	bControlled [in] If this parameter is true, the mouse is enabled. If the parameter is false, the mouse is disabled.
Return Values	Returns the current mouse available status.
Remarks	

[16]

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

[17]

User Interface	signRotateMode 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 ERR_OK. If the function fails, the return value is ERR_NOSUPPORTED

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; //0 degrees
	Mode=1; //90 degrees
	Mode=2; //180 degrees
	Mode=3; //270 degrees

Constant Definition		
ERR_OK	0	Is ok.
ERR_DEVICE_NOTFOUND	-1	No available devices were found.
ERR_DEVICE_OPENFAIL	-2	The function fails.
ERR_DEVICE_NOTCONNECTED	-3	Not connected.
ERR_INVALIDPARAM	-101	Invalid parameter.
ERR_NOSUPPORTED	-102	This operation is not supported.

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