H2ECTRL.ocx插件接口函数详细说明

2014.05.22

## 接口函数概览

### 触摸传感器

STDMETHODIMP CH2ECtrlAtl::TouchSensorState(BYTE bTouchMode, LONG\* retval)

### 串口操作

STDMETHODIMP CH2ECtrlAtl::OpenCom(LONG\* retval)

### 模板操作

STDMETHODIMP CH2ECtrlAtl::Enroll(WORD wBank, WORD wIdx, UINT uiTimes, LONG lTouchInfo, LONG\* retval)

STDMETHODIMP CH2ECtrlAtl::DeleteTemplete(WORD wBank, WORD wIdx, INT iMode, LONG\* retval)

STDMETHODIMP CH2ECtrlAtl::GetTemplate(WORD wBank, WORD wIdx, VARIANT vTemplate, LONG\* retval)

STDMETHODIMP CH2ECtrlAtl::SetTemplate(WORD wBank, WORD wIdx, VARIANT vTemplate, LONG\* retval)

### 认证

STDMETHODIMP CH2ECtrlAtl::Verify1\_1(WORD wBank, WORD wIdx, INT\* iMatchMemNo)

STDMETHODIMP CH2ECtrlAtl::Verify1\_N(WORD wBank, INT\* iMatchMemNo)

STDMETHODIMP CH2ECtrlAtl::VerifyAcrossBanks(VARIANT vAcrossBanks, WORD wTotalBankNumbers, VARIANT vMatchedBankAndTemplate, LONG\* retval)

## 接口函数详细说明

### 函数OpenCom

Table 1描述了函数OpenCom

Table 1. 函数OpenCom

|  |  |
| --- | --- |
| 函数名 | OpenCom |
| 函数原型 | STDMETHODIMP CH2ECtrlAtl::OpenCom(LONG\* retval) |
| 功能描述 | 打开设备串口。 |
| 输出参数1 | \* retval：返回错误码。 |
| 先决条件 | 无。 |

Javascript示例：

<input type="button" value="OpenCom" onclick="OpenComClick();" />

<script type="text/javascript">

function OpenComClick() {

var OpenComFun = document.getElementById("H2ECtrlAtl");

var returnVal = OpenComFun.OpenCom();

if (returnVal == 0) {

alert("OK! COM Opend!");

}

else {

alert("ErrCode: " + returnVal + " )");

}

}

</script>

### **函数**TouchSensorState

Table 2描述了函数TouchSensorState。

Table 2. 函数TouchSensorState

|  |  |
| --- | --- |
| 函数名 | TouchSensorState |
| 函数原型 | STDMETHODIMP CH2ECtrlAtl::TouchSensorState(BYTE bTouchMode, LONG\* retval) |
| 功能描述 | H2E具有两个检测手指位置放置是否正确的传感器，TouchOut1与TouchOut2 |
| 输入参数1 | bTouchMode：0x00：禁止TouchOut1&2检测；  0x01：使能TouchOut1检测；  0x02：使能TouchOut2检测；  0x03：使能TouchOut1&2检测(推荐)。 |
| 输出参数1 | \* retval：返回错误码。 |
| 先决条件 | 串口已打开。 |

Javascript示例：

<input type="button" value="Touch Sensor test" onclick="TouchSensorStateClick();" />

<script type="text/javascript">

function TouchSensorStateClick() {

var bTouchMode = 1; //b00000011 使能Touch Out 1&2检测

//b00000001 使能Touch Out 1检测

//b00000010 使能Touch Out 2检测

//b00000000 禁止Touch Out检测

var TouchOutFun = document.getElementById("H2ECtrlAtl");

alert("Put your finger ");

var returnVal = TouchOutFun.TouchSensorState(bTouchMode);

if (returnVal == 0) {

alert("Touch Sensor OK! ");

}

else if (returnVal == -11) {

alert("Timeout! ErrCode: " + returnVal + " !");

}

else if (returnVal == -250) {

alert("没有打开串口哦亲！");

}

}

</script>

### 函数Enroll

Table 3描述了函数**Enroll**。H2E具有两种认证机制，二次认证与三次认证，即对同一个手指进行多次对比认证，增强可靠性，但会占用更多的存储空间。推荐采用二次认证机制，既保证了较高的可靠性，又能存储较多的指静脉信息。PCT-KCC5031在此种模式下可以存储360个手指静脉信息。三次认证模式下可以存储230个手指静脉信息。

注意：1. 每个Bank最多存储15个指静脉模板。

2．只能在空白位置存储模板，如果准备存入的位置已有模板则需要先删除。

Table 3. 函数Enroll

|  |  |
| --- | --- |
| 函数名 | Enroll |
| 函数原型 | STDMETHODIMP CH2ECtrlAtl::Enroll(WORD wBank, WORD wIdx, UINT uiTimes, LONG lTouchInfo, LONG\* retval) |
| 功能描述 | 拍摄指静脉，保存到wBank中的nIdx号模板。 |
| 输入参数1 | wBank：Bank号  0x0000 ~ 0x7FFF： Flash ROM  0x8000 ~ 0xFFFF： SDRAM |
| 输入参数2 | nIdx：模板号（0x0000~ 0xFFFF）； |
| 输入参数3 | uiTimes：第几次采集，例如第一次采集为0 |
| 输入参数4 | lTouchInfo：触摸传感器状态 |
| 输出参数1 | \* retval：返回错误码。 |
| 先决条件 | 串口已打开。 |

Javascript示例：

<input type="button" value="Enroll" onclick="enrollclick();" />

<script type="text/javascript">

function enrollclick() {

var wBank = 0x0000;

var nIdx = 0x0000;

var uiTimes = 0;

var lTouchInfo = 0;

var bTouchMode = 1; //b00000011 使能Touch Out 1&2检测

//b00000001 使能Touch Out 1检测

//b00000010 使能Touch Out 2检测

//b00000000 禁止Touch Out检测

var EnrollFun = document.getElementById("H2ECtrlAtl");

var TouchOutFun = document.getElementById("H2ECtrlAtl");

for (i = 0; i < 2; i++) {

if (i == 1) {

alert("移开手指，重新放上")

}

alert("放上你的手指，采集指静脉(" + (i + 1) + "/2). ");

lTouchInfo = TouchOutFun.TouchSensorState(bTouchMode);

if (lTouchInfo != 0) //未检测到手指放到触摸传感器上

{

if (i > 0) {

//Do dummy enroll opelation.

var RetValEnroll = EnrollFun.Enroll(wBank, nIdx, i, lTouchInfo);

}

alert("指静脉采集超时！ErrCode： " + lTouchInfo);

break;

}

//Enroll sequence

RetValEnroll = EnrollFun.Enroll(wBank, nIdx, i, lTouchInfo);

if (RetValEnroll < 0) {

alert("指静脉采集发生错误，ErrCode： " + RetValEnroll);

break;

}

alert("采集 " + (i + 1) + "/2 OK");

}

if (RetValEnroll < 0 || lTouchInfo != 0) {

alert("指静脉采集错误，请重新采集！");

}

else {

alert("指静脉采集完成！");

}

}

</script>

### 函数Verify1\_1

Table 4描述了函数Verify1\_1

Table 4. 函数Verify1\_1

|  |  |
| --- | --- |
| 函数名 | Verify1\_1 |
| 函数原型 | STDMETHODIMP CH2ECtrlAtl::Verify1\_1(WORD wBank, WORD wIdx, INT\* iMatchMemNo) |
| 功能描述 | 1：1认证。 |
| 输入参数1 | wBank：Bank号  0x0000 ~ 0x7FFF： Flash ROM  0x8000 ~ 0xFFFF： SDRAM |
| 输入参数2 | wIdx：模板号（0x0000 ~ 0xFFFF） |
| 输出参数1 | \* iMatchMemNo：大于等于0为被认证模板号，小于0是错误码 |
| 先决条件 | 串口已打开 |

Javascript示例：

<input type="button" value="Verify 1:1" onclick="verify1\_1click();" />

<script type="text/javascript">

function verify1\_1click() {

var wBank = 0x0000;

var nIdx = 00;

var lTouchInfo = 0;

var bTouchMode = 1; //b00000011 使能Touch Out 1&2检测

//b00000001 使能Touch Out 1检测

//b00000010 使能Touch Out 2检测

//b00000000 禁止Touch Out检测

var verify1\_1Fun = document.getElementById("H2ECtrlAtl");

var TouchOutFun = document.getElementById("H2ECtrlAtl");

alert("放上你的手指，开始指静脉(1:1). ");

lTouchInfo = TouchOutFun.TouchSensorState(bTouchMode);

if (lTouchInfo != 0) //未检测到手指放到触摸传感器上

{

alert("TouchOut传感器超时！ErrCode： " + lTouchInfo);

}

else {

var byMatchMemNo = verify1\_1Fun.Verify1\_1(wBank, nIdx);

if (byMatchMemNo == nIdx) {

alert("身份验证成功 (匹配模板内存编号: " + byMatchMemNo + " )");

}

else {

alert("身份验证失败 ! ErrCode: " + byMatchMemNo);

}

}

}

</script>

### 函数Verify1\_N

Table 5描述了函数Verify1\_N，此函数在某一bank中查询认证指静脉信息，每个bank最大模板数为15。

Table 5. 函数Verify1\_N

|  |  |
| --- | --- |
| 函数名 | Verify1\_N |
| 函数原型 | STDMETHODIMP CH2ECtrlAtl::Verify1\_N(WORD wBank, INT\* iMatchMemNo) |
| 功能描述 | 1：N 认证。 |
| 输入参数1 | wBank：Bank号  0x0000 ~ 0x7FFF： Flash ROM  0x8000 ~ 0xFFFF： SDRAM |
| 输出参数1 | \* iMatchMemNo：大于等于0为被认证模板号，小于0是错误码 |
| 先决条件 | 串口已打开 |

Javascript示例：

<input type="button" value="Verify 1:N" onclick="verify1\_Nclick();" />

<script type="text/javascript">

function verify1\_Nclick() {

var wBank = 0x0000;

var lTouchInfo = 0;

var bTouchMode = 1; //b00000011 使能Touch Out 1&2检测

//b00000001 使能Touch Out 1检测

//b00000010 使能Touch Out 2检测

//b00000000 禁止Touch Out检测

var verify1\_NFun = document.getElementById("H2ECtrlAtl");

var TouchOutFun = document.getElementById("H2ECtrlAtl");

alert("放上你的手指，开始认证(1:N). ");

lTouchInfo = TouchOutFun.TouchSensorState(bTouchMode);

if (lTouchInfo != 0) //未检测到手指放到触摸传感器上

{

alert("TouchOut传感器超时！ErrCode： " + lTouchInfo);

}

else {

var byMatchMemNo = verify1\_NFun.Verify1\_N(wBank);

if (byMatchMemNo >= 0) {

alert("身份验证成功 (匹配模板内存编号: " + byMatchMemNo + " )");

}

else {

alert("身份验证失败 ! ErrCode: " + byMatchMemNo);

}

}

}

</script>

### 函数VerifyAcrossBanks

Table 6描述了函数VerifyAcrossBanks。

Table 6. 函数VerifyAcrossBanks

|  |  |
| --- | --- |
| 函数名 | VerifyAcrossBanks |
| 函数原型 | STDMETHODIMP CH2ECtrlAtl::VerifyAcrossBanks(VARIANT vAcrossBanks, WORD wTotalBankNumbers, VARIANT vMatchedBankAndTemplate, LONG\* retval) |
| 功能描述 | 此函数在指定的多个bank中查询认证指静脉信息。 |
| 输入参数1 | vAcrossBanks：Bank号数组  0x0000 ~ 0x7FFF： Flash ROM  0x8000 ~ 0xFFFF： SDRAM |
| 输入参数2 | wTotalBankNumbers：Bank数量 |
| 输入参数3 | vMatchedBankAndTemplate：数组元素0为Bank，1为Template |
| 输出参数1 | \* retval：错误码 |
| 先决条件 | 串口已打开 |

Javascript示例：

<input type="button" value="VerifyAcrossBanks" onclick="VerifyAcrossBanksClick();" />

<script type="text/javascript">

function VerifyAcrossBanksClick() {

var vBanks = new Array(0x0000, 0x0001, 0x0002);//需和当前H2E中的Bank号一致

var Counts = 3;//H2E中Bank数量

var MatchedBankAndTemplate = new Array(0xff, 0xff);//元素0为Bank，1为Template

var lTouchInfo = 0;

var bTouchMode = 0; //b00000011 使能Touch Out 1&2检测

//b00000001 使能Touch Out 1检测

//b00000010 使能Touch Out 2检测

//b00000000 禁止Touch Out检测

var VerifyAcrossBanksFun = document.getElementById("H2ECtrlAtl");

var TouchOutFun = document.getElementById("H2ECtrlAtl");

alert("放上你的手指，开始认证(1:ALL). ");

lTouchInfo = TouchOutFun.TouchSensorState(bTouchMode);

if (lTouchInfo != 0) //未检测到手指放到触摸传感器上

{

alert("TouchOut传感器超时！ErrCode： " + lTouchInfo);

}

else

{

var retval = VerifyAcrossBanksFun.VerifyAcrossBanks(vBanks, Counts, MatchedBankAndTemplate);

if (retval == 0) {

alert("身份验证成功! Bank: " + MatchedBankAndTemplate[0] + ", Template: " + MatchedBankAndTemplate[1]);

}

else {

alert("ErrCode: " + retval);

}

}

}

</script>

### **函数**GetTemplate

Table 7描述了函数GetTemplate。获取的位置必须有模板，否则函数行为不确定。

Table 7. 函数GetTemplate

|  |  |
| --- | --- |
| 函数名 | DeleteTemplete |
| 函数原型 | STDMETHODIMP CH2ECtrlAtl::GetTemplate(WORD wBank, WORD wIdx, VARIANT vTemplate, LONG\* retval) |
| 功能描述 | 获取存储在H2E指定位置的模板 |
| 输入参数1 | wBank：Bank号  0x0000 ~ 0x7FFF： Flash ROM  0x8000 ~ 0xFFFF： SDRAM |
| 输入参数2 | wIdx：模板号（0x0000 ~ 0xFFFF） |
| 输入参数3 | vTemplate：函数会将获取到的1080字节模板填充到vTemplate。 |
| 输出参数1 | \* retval：返回错误码。 |
| 先决条件 | 串口已打开 |

Javascript示例：

<input type="button" value="GetTemplate" onclick="GetTemplateClick()" />

<script type="text/javascript">

function GetTemplateClick() {

var wBank = 0x0000;

var nIdx = 00;

var vTemplete = new Array(1080);

for (i = 0; i < vTemplete.length; i++) {

vTemplete[i] = 0;

}

var GetTmpFun = document.getElementById("H2ECtrlAtl");

var RetVal = GetTmpFun.GetTemplate(wBank, nIdx, vTemplete);

for (i = 0; i < 1080; i++) {

document.write(vTemplete[i] + ",")

}

//if (RetVal == 0) {

// alert("Get OK! ");

//}

//else {

// alert("ErrCode: " + RetVal);

//}

}

</script>

### **函数**SetupTemplete

Table 8描述了函数SetTemplate

Table 8. 函数SetTemplate

|  |  |
| --- | --- |
| 函数名 | SetTemplate |
| 函数原型 | STDMETHODIMP CH2ECtrlAtl::SetTemplate(WORD wBank, WORD wIdx, VARIANT vTemplate, LONG\* retval) |
| 功能描述 | 下载指静脉模板到H2E指定位置。 |
| 输入参数1 | wBank：Bank号  0x0000 ~ 0x7FFF： Flash ROM  0x8000 ~ 0xFFFF： SDRAM |
| 输入参数2 | wIdx：模板号（0x0000 ~ 0xFFFF）； |
| 输入参数3 | vTemplete：要下载的指静脉模板。 |
| 输出参数1 | \* retval：返回错误码。 |
| 先决条件 | 串口已打开 |

Javascript示例：

<input type="button" value="SetupTemplete" onclick="SetTemplateClick();" />

<script type="text/javascript">

function SetTemplateClick() {

var wBank = 0x0000;

var nIdx = 00;

var vTemplete = new Array(100, 61, 177, 64, 84, 10, 128, 34, 227, 56, 201, 150, 99, 54, 221, 158, 91, 117, 17, 194, 68, 72, 186, 43, 66, 187, 80, 35, 110, 226, 61, 132, 157, 249, 188, 152, 225, 218, 89, 49, 192, 200, 57, 228, 233, 10, 194, 195, 76, 62, 201, 9, 36, 34, 24, 42, 159, 240, 91, 110, 216, 166, 188, 177, 97, 21, 51, 4, 138, 238, 240, 16, 107, 80, 149, 187, 148, 219, 175, 100, 218, 128, 139, 0, 237, 145, 104, 17, 39, 145, 126, 117, 165, 242, 163, 182, 106, 11, 119, 67, 18, 181, 184, 62, 120, 121, 176, 1, 219, 154, 239, 63, 145, 99, 13, 14, 247, 51, 219, 64, 30, 103, 17, 34, 176, 57, 156, 175, 105, 125, 205, 164, 160, 192, 44, 133, 155, 124, 4, 122, 241, 212, 226, 231, 224, 52, 232, 206, 48, 100, 57, 236, 153, 174, 106, 251, 252, 73, 135, 103, 121, 120, 118, 192, 248, 30, 98, 240, 243, 3, 15, 84, 155, 213, 66, 80, 238, 171, 60, 29, 1, 153, 54, 251, 22, 67, 253, 167, 96, 218, 159, 82, 219, 206, 71, 123, 214, 207, 244, 239, 207, 17, 26, 155, 33, 241, 219, 235, 79, 46, 217, 127, 8, 209, 66, 87, 196, 27, 15, 230, 87, 215, 207, 44, 140, 223, 182, 115, 118, 7, 85, 158, 82, 111, 136, 39, 84, 31, 38, 246, 57, 210, 174, 225, 243, 100, 209, 243, 36, 133, 210, 104, 10, 109, 106, 113, 248, 218, 158, 51, 114, 2, 83, 218, 246, 29, 170, 229, 236, 158, 22, 65, 57, 40, 53, 87, 23, 255, 178, 75, 171, 18, 67, 95, 78, 189, 157, 125, 73, 188, 208, 81, 46, 205, 102, 204, 58, 157, 236, 220, 204, 241, 21, 18, 38, 233, 249, 241, 161, 49, 14, 10, 254, 82, 102, 209, 191, 115, 65, 119, 187, 227, 121, 244, 210, 108, 93, 184, 64, 35, 12, 32, 178, 187, 3, 198, 2, 136, 171, 234, 69, 32, 171, 198, 42, 244, 237, 228, 30, 203, 136, 172, 106, 250, 50, 226, 250, 76, 13, 160, 84, 211, 102, 11, 73, 147, 5, 243, 67, 78, 7, 31, 189, 40, 73, 61, 51, 150, 35, 161, 140, 212, 123, 227, 91, 154, 213, 244, 248, 187, 134, 49, 102, 233, 70, 112, 38, 199, 58, 201, 182, 165, 132, 214, 99, 44, 42, 67, 232, 123, 177, 76, 229, 142, 130, 200, 241, 226, 67, 65, 147, 109, 127, 245, 169, 29, 105, 152, 187, 213, 247, 41, 76, 159, 63, 252, 202, 207, 155, 45, 90, 49, 36, 59, 161, 250, 97, 149, 221, 17, 117, 163, 96, 161, 35, 193, 200, 158, 5, 229, 56, 177, 45, 113, 82, 144, 82, 183, 98, 77, 99, 192, 247, 100, 215, 14, 12, 149, 210, 165, 191, 175, 114, 29, 12, 87, 80, 244, 75, 146, 89, 33, 119, 57, 192, 195, 80, 105, 242, 178, 105, 31, 8, 117, 143, 214, 193, 76, 134, 214, 24, 159, 102, 78, 217, 108, 233, 223, 17, 9, 111, 198, 132, 2, 250, 64, 8, 49, 57, 81, 171, 70, 232, 115, 230, 79, 31, 229, 135, 84, 102, 192, 38, 11, 83, 208, 75, 190, 62, 246, 97, 45, 231, 79, 254, 217, 103, 10, 11, 67, 39, 64, 86, 93, 61, 136, 25, 64, 200, 171, 110, 211, 27, 115, 49, 168, 182, 192, 79, 43, 64, 218, 23, 195, 229, 0, 147, 129, 0, 121, 81, 63, 8, 39, 2, 75, 45, 227, 176, 1, 51, 64, 92, 3, 50, 81, 99, 116, 93, 65, 79, 34, 95, 11, 51, 61, 161, 138, 137, 170, 79, 4, 60, 253, 52, 218, 181, 214, 43, 85, 82, 143, 232, 86, 123, 151, 65, 195, 181, 245, 104, 73, 251, 219, 42, 254, 247, 154, 238, 14, 26, 252, 4, 88, 219, 253, 56, 148, 2, 140, 80, 50, 97, 77, 58, 21, 19, 205, 12, 78, 168, 158, 10, 78, 3, 203, 62, 102, 214, 255, 124, 159, 102, 75, 218, 82, 201, 61, 102, 28, 34, 11, 43, 116, 192, 57, 116, 235, 21, 205, 72, 172, 183, 199, 57, 103, 110, 198, 126, 168, 39, 18, 233, 175, 217, 242, 162, 172, 211, 81, 150, 152, 189, 91, 111, 224, 63, 105, 186, 205, 166, 152, 77, 144, 211, 199, 54, 85, 154, 4, 176, 239, 58, 32, 255, 4, 85, 254, 87, 99, 198, 220, 63, 152, 13, 123, 68, 42, 213, 147, 71, 146, 59, 84, 163, 42, 59, 247, 88, 124, 119, 188, 210, 39, 215, 128, 109, 213, 30, 180, 5, 35, 196, 31, 54, 72, 193, 245, 153, 183, 199, 156, 107, 159, 20, 116, 184, 27, 111, 9, 18, 253, 154, 0, 59, 119, 66, 60, 41, 73, 180, 193, 30, 59, 73, 158, 137, 52, 172, 225, 215, 54, 209, 239, 28, 207, 78, 222, 48, 19, 184, 31, 202, 37, 63, 230, 163, 156, 10, 33, 29, 193, 190, 150, 0, 201, 77, 219, 104, 119, 28, 7, 60, 47, 230, 19, 103, 138, 48, 57, 189, 231, 145, 100, 122, 5, 190, 180, 122, 91, 62, 166, 136, 158, 78, 196, 249, 19, 2, 38, 175, 127, 25, 61, 167, 76, 110, 251, 141, 161, 41, 255, 87, 179, 238, 26, 6, 38, 106, 180, 151, 244, 166, 196, 167, 185, 239, 83, 224, 47, 233, 90, 195, 207, 160, 151, 43, 161, 143, 221, 84, 54, 83, 179, 4, 242, 167, 230, 99, 86, 222, 140, 34, 254, 222, 129, 65, 118, 249, 180, 218, 203, 146, 130, 139, 180, 232, 166, 97, 83, 101, 51, 155, 70, 166, 58, 76, 224, 203, 189, 164, 165, 87, 182, 55, 125, 174, 99, 100, 113, 240, 169, 216, 238, 121, 136, 71, 46, 198, 6, 20, 182, 221, 151, 204, 118, 88, 29, 129, 231, 185, 63, 224, 162, 14, 82, 247, 2, 30, 234, 64, 180, 219, 43, 127, 241, 247, 226, 41, 209, 242, 226, 97, 220, 208, 4, 102, 246, 203, 225, 15, 94, 50, 75, 182, 185, 102, 171, 94, 8, 164, 115, 134, 107, 110, 128, 68, 206, 109, 102, 238, 48, 210, 99, 110, 49, 209, 60, 147, 29, 242, 235, 162, 159, 60, 205, 27, 177, 66, 45, 164, 116, 66, 71, 51, 220, 141, 39, 96, 23, 194, 242, 56, 153, 130, 203, 234, 226, 222, 21);

var SetupFun = document.getElementById("H2ECtrlAtl");

var retval = SetupFun.SetTemplate(wBank, nIdx, vTemplete);

if (retval == 0) {

alert("Set OK! ");

}

else {

alert("ErrCode: " + retval);

}

}

</script>

### **函数**DeleteTemplete

Table 9描述了函数DeleteTemplete

Table 9. 函数DeleteTemplete

|  |  |
| --- | --- |
| 函数名 | DeleteTemplete |
| 函数原型 | STDMETHODIMP CH2ECtrlAtl::DeleteTemplete(WORD wBank, WORD wIdx, INT iMode, LONG\* retval) |
| 功能描述 | 删除指定模板。 |
| 输入参数1 | wBank：Bank号  0x0000 ~ 0x7FFF： Flash ROM  0x8000 ~ 0xFFFF： SDRAM |
| 输入参数2 | wIdx：模板号（0x0000 ~ 0xFFFF） |
| 输入参数3 | iMode：删除选项  0：删除指定的模板号；  1：删除当前Bank 的全部模板；  2：删除全部Bank 的全部模板 。 |
| 输出参数1 | \* retval：返回错误码。 |
| 先决条件 | 串口已打开 |

Javascript示例：

<input type="button" value="DeleteTemplete" onclick="deleteclick();" />

<script type="text/javascript">

function deleteclick() {

var wBank = 0x0000;

var nIdx = 00;

var iMode = 0; //0:删除指定的模板号；

//1:删除当前Bank 的全部模板；

//2:删除全部Bank 的全部模板 。

var del = document.getElementById("H2ECtrlAtl");

var RetVal = del.DeleteTemplete(wBank, nIdx, iMode);

if (RetVal == 0) {

alert("Delete OK! ");

}

else {

alert("ErrCode: " + RetVal);

}

}

</script>

## 错误码

### 串口操作错误码

|  |  |
| --- | --- |
| 错误码 | 描述 |
| 0 | 操作成功 |
| -200 | 串口打开错误 |
| -201 | 串口关闭错误 |
| -202 | 串口读错误 |
| -203 | 串口写错误 |
| -204 | 没有收到数据 |
| -205 | 收到无效数据，一般是手指没放好 |
| -206 | 串口错误 |
| -207 | Add校验码错误 |
| -208 | Xor校验码错误 |
| -250 | 串口没有打开 |

### 认证操作错误码

|  |  |
| --- | --- |
| 错误码 | 描述 |
| 0 | 操作成功 |
| -10 | 发送超时 |
| -11 | 触摸传感器超时 |
| -101 | 无效参数 |
| -104 | 认证失败 |