

自助机硬件封装技术方案

(版本号 2.00)

修改记录表

序号	版本	修改日期	修改内容描述	修改人
1	1.00	2015-10-10	初版	程达
2	2.00	2016-11-14	第二版，增加新硬件	程达
3	3.00	2016-11-14	第三版，增加新硬件	程达

一、 前言

《自助机硬件封装技术方案》是旭辉卓越健康信息科技有限公司开发的，基于自助机硬件进行封装，以供 his 调用的技术方案。

二、 规范使用对象

- HIS 各厂商

三、 调用方式

1、硬件封装包的文件里会包含 reg.bat、HardwareCom.dll、GGXmlTcp.dll，运行 reg.bat 进行注册

2、以 stdcall 标准调用 GGXmlTcp.dll 里的 XmlTcp 函数

_stdcall int __CALLTYPE XmlTcp(char* xmlbuf, int timeout=0);

C# 范例：

```
//申明：
[DllImport(@"GGXmlTcp.dll")]
public static extern int XmlTcp(StringBuilder xmlbuf, int timeout);
//调用：
StringBuilder xmlBuff= new StringBuilder(1000);
xmlBuff.Append("<invoke name=\"LOCKCLOSE\"><arguments><string
id=\"INDEX\">1</string></arguments></invoke>");
int ret = XmlTcp(xmlBuff, 1000);
MessageBox.Show(xmlBuff.ToString());
```

pb 范例：

```
//申明：
function int XmlTcp(ref string xmlbuf, long timeout) library "GGXmlTcp.dll" alias for
"XmlTcp;Ansi"
//调用：
String xmlbuf = "<invoke>...</invoke>" + space(2048);
Long ret = XmlTcp(xmlBuff, 1000);
```

四、 硬件对象

1.条码阅读器。

提供条码扫描功能。供 his 厂商来进行健康卡读卡以及报告单扫码打印操作。（如有需要）

2.二代身份证识别模块。

提供二代证识别功能。供 his 厂商来进行身份证绑定操作。（如有需要）

3.凭条打印机模块。

提供凭条打印功能。供 his 厂商来进行凭条打印操作。挂号，缴费等（如有需要）
提供凭条纸将尽和凭条纸尽等凭条检测状态操作。（如有需要）

4.摄像头模块。

提供摄像头拍照功能。供 his 厂商来进行监测操作人。（如有需要）

5.发卡器模块

提供发卡器发卡功能，供 his 厂商来进行发卡操作。（如有需要）

6.纸币器模块

提供现金预存功能，供 his 厂商来进行现金预存操作。（如有需要）

7.读卡器模块

提供读卡功能，供 his 厂商来进行读卡操作。（如有需要）

8.门禁模块

提供门禁功能，供 his 厂商来进行门禁灯操作。（如有需要）

9.推杆模块

提供推杆移动功能，供 his 厂商来进行推杆操作。（如有需要）

9.门锁模块

提供开关锁功能，供 his 厂商来进行电子门锁操作。（如有需要）

11.病历打印模块

提供病历打印和状态监控功能，供 his 厂商来进行病历打印操作。（如有需要）

12.激光打印模块

提供激光打印机监控功能，供 his 厂商来进行激光打印操作。（如有需要）

13.金属键盘模块

提供金属键盘输入功能，供 his 厂商来进行金属键盘操作。（如有需要）

五、 硬件封装实现方式

1.条码阅读器。

提供一个扫描接口，调用此接口开始扫描，直到扫描出数据或者时间超时报接口返回。

2.二代身份证识别模块。

提供一个识别接口，调用此接口开始识别，直到识别出数据或者时间超时报接口返回。

3.凭条打印机模块。

提供一个打印接口，调用此接口后凭条打印机打印指定文件。（需事先写好指定文件）

提供一个凭条打印机检测接口，调用此接口后返回凭条打印机状态。（正常，纸将尽，纸已尽）

4.摄像头模块。

提供摄像头拍照功能。供 his 厂商来进行监测操作人。（如有需要）

摄像头拍照之后会存入指定文件夹。

5. 发卡器模块

提供发卡功能。供 his 厂商来进行发卡操作。

提供一个读卡接口，调用此接口来进行移动卡到读卡位及读出卡号。

提供一个写卡接口，调用此接口来进行移动卡到读卡位及写入卡号。

提供一个回收卡接口，调用此接口来进行移动卡到回收槽。

提供一个吐卡接口，调用此接口来进行移动卡到出卡口。

提供一个打印接口，调用此接口来进行打印操作。

提供移到射频位接口。调用此接口移动到射频位接口。

提供写入射频数据接口，调用此接口来写入射频数据。

提供读取射频数据接口，调用此接口来读取射频数据。

6. 纸币器模块

提供打开纸币器接口，调用此接口来开始纸币预存。

提供存钞接口，调用此接口来进行纸币预存。

提供关闭纸币器接口，调用此接口来结束纸币预存。

提供禁止纸币入币接口，调用此接口禁止纸币入币。

提供允许纸币入币接口，调用此接口允许纸币入币。

提供设置纸币入币金额接口，调用此接口设置纸币可入币金额。

7.读卡器模块

提供允许进卡接口，调用此接口来进行允许读卡操作。
提供禁止进卡接口，调用此接口来进行禁止进卡操作。
提供检测卡接口，调用此接口来判断通道是否有卡。
提供读取磁条接口，调用此接口来读取磁道号。
提供移动到 IC 位接口，调用此接口移动到 IC 位接口。
提供移到到射频位接口。调用此接口移动到射频位接口。
提供移到到卡后端接口。调用此接口移动到卡后端接口。
提供退卡接口，调用此接口来退出卡片。
提供读取射频数据接口，调用此接口来读取射频数据。
提供发卡出来的接口，调用此接口来发卡。
提供读取社保卡的接口，调用此接口来读取社保卡。

8.门禁模块

提供开门禁灯接口。
提供关门禁灯接口。
提供门禁灯闪接口。

9.推杆模块

提供推杆打开接口。
提供推杆关闭接口。

10.门锁模块

提供开关锁接口。

11.病历打印模块

提供病历打印机检测状态接口。
提供病历打印机打印接口。

12.激光打印模块

提供激光打印机检测状态接口。

13. 金属键盘模块

提供金属键盘打开输入接口。
提供金属键盘关闭输入接口。

14.找零机器

提供找零功能的接口
提供找零机中余额记录添加的接口
提供找零机中余额记录清零的接口

六、 硬件封装接口

1. 条码阅读器扫描接口

1) 入参:

示例:

```
<invoke name="BARCODESCANNING">
  <arguments>
    </arguments>
</invoke>
```

2) 出参:

```
<return name=" BARCODESCANNING ">
  <arguments>
    < string id=" BARCODE">123456789</string>
  </arguments>
</return>
```

注: 如果扫描失败返回ERROR, 如

```
<return name=" BARCODESCANNING ">
  <arguments>
    <string id="ERROR">TIMEOUT</string>
    或
    <string id="ERROR">DEVEERROR</string>
  </arguments>
</return>
```

2. 二代身份证识别模块接口

1) 入参:

示例:

```
<invoke name="SHENFENZHENG">
    <arguments></arguments>
</invoke>
```

2) 出参:

示例:

```
<return name=" SHENFENZHENG">
    <arguments>
        <string id="IDName">张三</string>
        <string id=" Sex">男</string>
        <string id="Born">19851218</string>
        <string id="Address">上海市闸北区</string>
        <string id="IDCardNo">320681198512187211</string>
    </arguments>
</return>
```

注: 如果读卡失败返回ERROR, 如

```
<return name="SHENFENZHENG">
    <arguments>
        <string id="ERROR">DEVEERROR</ string>
    </arguments>
</return>
```

代码	名称	类型	注释
IDName	姓名	String	
Sex	性别	String	
Born	出生日期	String	
Address	地址	String	
IDCardNo	身份证号	String	
NATION	民族	String	
NATIONCODE	民族代码	String	
ERROR	失败时传	String	DEVEERROR 设备故障

3. 凭条打印机检测状态

1) 入参:

示例:

```
<invoke name="RECEIPTSTATUS">  
  <arguments>  
  </arguments>  
</invoke>
```

2) 出参, 同打印凭条返回相同的结果

示例:

```
<return name="RECEIPTSTATUS">  
  <arguments>  
    正常成功:  
    <string id="ERROR">SUCCESS</string>  
    成功且纸将尽:  
    <string id="ERROR">PAPERNEAREND</string>  
    缺纸:  
    <string id="ERROR">NOPAPER</string>  
    故障:  
    <string id="ERROR">DEVEERROR</string>  
  </arguments>  
</return>
```


4. 凭条打印接口（打印指定文件）

1) 入参:

示例:

```
<invoke name="RECEIPTPRINT">  
  <arguments>  
  </arguments>  
</invoke>
```

2) 出参 :

示例:

```
<return name=" RECEIPTPRINT ">  
  <arguments>  
    正常成功:  
    <string id="ERROR">SUCCESS</string>  
  </arguments>  
</return>
```

5. 摄像头模块接口

一、 一次性拍照

1) 入参:

示例:

```
<invoke name="TAKEPHOTO">
  <arguments>
    <string id="CARDNO">123456</string>
    <string id="NAME">张三</string>
    <string id="USECAMERAINDEX">0</string><!--0 为顶部摄像头，1 为纸币器摄像头-->
  </arguments>
</invoke>
```

2) 出参:

示例:

```
<return name=" TAKEPHOTO ">
```

```
<arguments>
```

正常成功:

```
<string id="ERROR">SUCCESS</string>
```

故障:

```
<string id="ERROR">DEVEERROR</string>
```

```
</arguments>
```

```
</return>
```

二、 连续拍照开始拍照

1) 入参:

示例:

```
<invoke name="TAKEPHOTOSTART">
  <arguments>
    <string id="CARDNO">123456</string>
    <string id="NAME">张三</string>
    <string id="USECAMERAINDEX">0</string><!--0 为顶部摄像头, 1 为纸币器摄像头-->
  </arguments>
</invoke>
```

2) 出参:

示例:

```
<return name=" TAKEPHOTOSTART ">
  <arguments>
    正常成功:
    <string id="ERROR">SUCCESS</string>
    故障:
    <string id="ERROR">DEVEERROR</string>
  </arguments>
</return>
```

三、 连续拍照结束拍照

1) 入参:

示例:

```
<invoke name="TAKEPHOTOEND">
```

```
<arguments>
```

```
    <string id="USECAMERAINDEX">0</string><!--0 为顶部摄像头, 1 为纸币器摄像头-->
```

```
</arguments>
```

```
</invoke>
```

2) 出参:

示例:

```
<return name=" TAKEPHOTOEND ">
```

```
<arguments>
```

正常成功:

```
<string id="ERROR">SUCCESS</string>
```

故障:

```
<string id="ERROR">DEVEERROR</string>
```

```
</arguments>
```

```
</return>
```

6. 发卡器模块

一、 读卡

1) 入参:

示例:

```
<invoke name=" CARDSENDERREADCARD ">
```

```
    <arguments>
```

```
</arguments>
</invoke>
```

2) 出参：

代码	名称	类型	注释
CARDNO	卡号	String	
CARDSTACKSTATE	卡槽状态	String	0：无卡 1：有卡
ERROR	错误码	String	Success、NoCard、DevError

3)

示例：

```
<return name="CARDSENDERREADCARD">
  <arguments>
    < string id=" CARDNO">123456789</string>
    卡箱状态
    <string id=" CARDSTACKSTATE ">2</ string>
    成功：
    <string id="ERROR">SUCCESS</string>
    无卡：
    <string id="ERROR">NOCARD</string>
    故障：
    <string id="ERROR">DEVEEROR</string>

  </arguments>
</return>
```

二、 写卡

1) 入参：

示例：

```
<invoke name="CARDSENDERWRITECARD">
  <arguments>
    < string id=" CARDNO">123456789</string>
  </arguments>
</invoke>
```

2) 出参：

代码	名称	类型	注释
----	----	----	----

CARDSTACKSTATE	卡槽状态	String	0: 无卡 1: 有卡
ERROR	错误码	String	SUCCESS、NOCARD、DEVEEROR

3)

示例：

```
<return name="CARDSENDERWRITECARD">
  <arguments>
    <string id=" CARDSTACKSTATE ">0</ string>
    成功:
    <string id="ERROR">SUCCESS</string>
    无卡:
    <string id="ERROR">NOCARD</string>
    故障:
    <string id="ERROR">DEVEEROR</string>
  </arguments>
</return>
```

三、 回收卡

1) 入参：

示例：

```
<invoke name=" CARDSENDERRECYCLE ">
  <arguments>
  </arguments>
</invoke>
```

2) 出参：

示例：

```
<return name=" CARDSENDERRECYCLE ">
  <arguments>
    正常成功:
    <string id="ERROR">SUCCESS</string>
  </arguments>
</return>
```

四、 吐卡

1) 入参：

示例：

```
<invoke name=" CARDSENDEROUTCARD ">
    <arguments>
    </arguments>
</invoke>
```

2) 出参：

示例：

```
<return name="CARDSENDEROUTCARD">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
    </arguments>
</return>
```

五、 打印

1) 入参：

示例：

```
<invoke name=" CARDSENDERPRINT ">
    <arguments>
        <strings>
            <string id="FONTSIZE">9</string>
            <string id=" FONT">黑体</string>
            <string id=" LOCATIONX">15</string>
            <string id=" LOCATIONY">15</string>
            <string id="CONTENT">打印内容</string>
        </strings>
        <strings>
            <string id=" FONTSIZE">9</string>
            <string id=" FONT">黑体</string>
            <string id=" LOCATIONX">15</string>
            <string id=" LOCATIONY ">15</string>
            <string id=" CONTENT">打印内容</string>
        </strings>
    </arguments>
</invoke>
```

2) 出参：

示例：

```
<return name="CARDSENDERPRINT">
    <arguments>
        正常成功：
        <string id="ERROR">SUCCESS</string>
    </arguments>
</return>
```

六、 获取发卡器状态

3) 入参：

示例：

```
<invoke name="CARDSENDERGETSTATUS">
    <arguments>
    </arguments>
</invoke>
```

4) 出参：

示例：

```
<return name="CARDSENDERGETSTATUS">
    <arguments>
    正常成功：
        <string id="ERROR">SUCCESS</string>
        <string id=" ERRORMESSAGE ">SUCCESS</string>
        <string id="CARDBOXSTATUS">SUCCESS</string>
        <string id="CHANELSTATUS">SUCCESS</string>
    </arguments>
</return>
```

代码	名称	类型	注释
ERROR	返回状态	String	SUCCESS 为接口执行正常，具体内容看其它参数。 DEVEERROR 接口执行出错了，具体内容描述看参数 ERRORMESSAGE
ERRORMESSAGE	错误描述	String	为空是状态检测正常。 不为空为读卡器有异常，但不影响参数 CARDBOXSTATUS、CHANELSTATUS 的表达
CARDBOXSTATUS	卡箱状态	String	“0”为没卡，“1”为有卡

CHANELSTATUS	卡槽状态	String	“0”为卡槽没卡，“1”为卡槽有卡
--------------	------	--------	-------------------

七、 移卡到射频位

1) 入参:

示例:

```
<invoke name=" CARDSENDERMOVECARDTORF ">
    <arguments>
    </arguments>
</invoke>
```

2) 出参 :

示例:

```
<return name="CARDSENDERMOVECARDTORF">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
        故障:
        <string id="ERROR">DEVEEROR</string>
    </arguments>
</return>
```

八、 移卡到 IC 位

1) 入参:

示例:

```
<invoke name=" CARDSENDERMOVECARDTOIC ">
    <arguments>
    </arguments>
</invoke>
```

2) 出参 :

示例:

```
<return name="CARDSENDERMOVECARDTOIC">
    <arguments>
        正常成功:
```

```

        <string id="ERROR">SUCCESS</string>
        故障:
        <string id="ERROR">DEVEEROR</string>

    </arguments>

</return>

```

九、 写射频卡数据

1) 入参:

代码	名称	类型	注释
CARDNO	卡号	String	
BLOCKADR	块号	String	
NADR	块地址	String	
PASSWORD	认证密码	String	

示例:

```

<invoke name=" CARDSENDERWRITERFCARD ">
    <arguments>
        <string id="CARDNO">1234</string>
        <string id="BLOCKADR">0</string>
        <string id="NADR">0</string>
        <string id="PASSWORD">FFFFFFFFFFFF</string>
    </arguments>
</invoke>

```

2) 出参 :

示例:

```

<return name=" CARDSENDERWRITERFCARD ">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
        故障:
        <string id="ERROR">DEVEEROR</string>

    </arguments>

</return>

```

十、 读射频卡数据

1) 入参:

代码	名称	类型	注释
BLOCKADR	块号	String	
NADR	块地址	String	
PASSWORD	认证密码	String	

示例:

```
<invoke name=" CARDSENDERREADRFCARD ">
  <arguments>
    <string id="BLOCKADR">0</string>
    <string id="NADR">0</string>
    <string id="PASSWORD">FFFFFFFFFFFF</string>
  </arguments>
</invoke>
```

2) 出参 :

CARDNO	卡号	String	
--------	----	--------	--

示例:

```
<return name=" CARDSENDERREADRFCARD ">
  <arguments>
    正常成功:
    <string id="ERROR">SUCCESS</string>
    故障:
    <string id="ERROR">DEVEEROR</string>
    <string id="CARDNO">12345678</string>
  </arguments>
</return>
```

十一、 读社保卡卡内数据

入参:

示例:

```
<invoke
name="READSOCIALSECURITYCARD"><arguments></arguments></invoke>
```

出参：

代码	名称	类型	注释
ERROR	操作结果	String	成功返回 SUCCESS
ERRORMESSAGE	错误描述	String	
CARDN	社保卡号	String	
IDNUMBER	身份证号	String	
NAME	名字	String	
GENDER	性别	String	
NATION	民族	String	
BIRTHPLACE	籍贯	String	
BIRTHDATE	出生日期	String	

示例：

```
<return name="READSOCIALSECURITYCARD">
  <arguments>
    <string id="ERROR">DEVEERROR</string>
    <string id="ERRORMESSAGE">读卡器未初始化</string>
    <string id="CARDNO" />
    <string id="IDNUMBER" />
    <string id="NAME" />
    <string id="GENDER" />
    <string id="NATION" />
    <string id="BIRTHPLACE" />
    <string id="BIRTHDATE" />
  </arguments>
</return>
```

十、卡片回收（具有发卡功能的读卡器）

入参：

示例：

```
<invoke name="ReadCardMoveCardToRecycle"><arguments></arguments></invoke>
```

出参：

示例：

```
<return name="ReadCardMoveCardToRecycle">
  <arguments>
    <string id="ERROR">SUCCESS</string>
  </arguments>
</return>
```

十一、获取卡箱状态（具有发卡功能的读卡器）

入参：

示例：

```
<invoke name="GetCardBoxState"><arguments></arguments></invoke>
```

出参：

示例：

```
<return name="GetCardBoxState">
  <arguments>
    <string id="ERROR">SUCCESS</string>
    <string id="BoxState">0</string>
  </arguments>
</return>
```

参数说明：

代码	名称	类型	注释
BoxState	卡箱状态	String	0 为没卡，1 卡比较少，2 卡足够

十二、发送命令（EVOLIS 发卡机可用）

入参：

示例：

```
<invoke name="CardSenderSendCmd">
  <arguments><string id="cmd">\rSs\n\rSi\n</string></arguments></invoke>
```

参数说明：

cmd，要发送的命令，具体内容如下：

移到磁卡位

\rSi\n

移到射频位

\rSic\n

移到 ic 位

\rSis\n

移到磁卡位、读卡（有点问题，读不出卡号）

\rSs\n\rSi\n

前端出卡

\rSs\n

\rSe\n

要调用接口两次，第一次发 \rSs\n，第二次发 \rSe\n

出参：

示例：

```
<return name="CardSenderSendCmd">
<arguments>
<string id="ERROR">DEVEERROR</string>
<string id="Answer">OK</string>
</arguments>
</return>
```

十三、非接触射频卡激活（EVOLIS、ACT-F3-1300 可用）

入参：

示例：

```
<invoke name=" CARDSENDEROPENCPURFCARD"><arguments></arguments></invoke>
```

出参：

示例：

```
<return name=" CARDSENDEROPENCPURFCARD">
<arguments>
```

正常成功：

```
        <string id="ERROR">SUCCESS</string>
        故障:
        <string id="ERROR">DEVEEROR</string>
    </arguments>
</return>
```

十四、非接触射频卡释放（EVOLIS、ACT-F3-1300 可用）

入参：

示例：

```
<invoke name=" CARDSENDERCLOSECPURFCARD"><arguments></arguments></invoke>
```

出参：

示例：

```
<return name=" CARDSENDERCLOSECPURFCARD">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
        故障:
        <string id="ERROR">DEVEEROR</string>
    </arguments>
</return>
```

十五、非接触射频卡发送指令（EVOLIS、ACT-F3-1300 可用）

入参：

示例：

```
<invoke name=" CARDSENDEREXCHANGECPURFCARD">
    <arguments>
        <string id=" cmd">发送的命令字符串</string>
    </arguments>
</invoke>
```

出参：

示例：

```
<return name=" CARDSENDEREXCHANGECPURFCARD">
    <arguments>
        正常成功:
```

```
<string id="ERROR">SUCCESS</string>
    故障:
    <string id="ERROR">DEVEEROR</string>
    <string id=" RecvData">收到的命令字符串</string>
</arguments>

</return>
```

7. 纸币器模块

一、 打开纸币器

1) 入参:

示例:

代码	名称	类型	注释
ID	姓名或者卡号	String	用于日志记录唯一标识

```
<invoke name="BILLACCEPTOROPENPORT">
    <arguments>
        <string id="ID"></string>
    </arguments>
</invoke>
```

2) 出参 :

示例:

```
<return name="BILLACCEPTOROPENPORT">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
    </arguments>
</return>
```

二、 获取存钞明细

1) 入参:

示例:

```
<invoke name=" BILLACCEPTORSTACKMONEYDETAIL">
```



```
<arguments>
</arguments>
</invoke>
```

2) 出参：

出参：

代码	名称	类型	注释
STACKMONEY	存钱明细	String	显示当前存钱明细

3)

示例：

```
<return name="BILLACCEPTORSTACKMONEYDETAIL">
<arguments>
  <string id="STACKMONEY">100,123456,2016/2/29 10:35:37|100,123456,2016/2/29 10:
35:43</string>
</arguments></return>
```

三、 关闭纸币器

1) 入参：

示例：

```
<invoke name=" BILLACCEPTORCLOSEPORT ">
  <arguments>
  </arguments>
</invoke>
```

2) 出参：

示例：

```
<return name=" BILLACCEPTORCLOSEPORT ">
  <arguments>
    正常成功：
    <string id="ERROR">SUCCESS</string>
  </arguments>
</return>
```

四、 禁止纸币入币

1) 入参：

示例：

```
<invoke name=" BILLACCEPTORNOTALLOWCASHIN ">
    <arguments>
    </arguments>
</invoke>
```

2) 出参：

示例：

```
<return name=" BILLACCEPTORNOTALLOWCASHIN ">
    <arguments>
        正常成功：
        <string id="ERROR">SUCCESS</string>
    </arguments>
</return>
```

五、 允许纸币入币

1) 入参：

示例：

```
<invoke name=" BILLACCEPTORALLOWCASHIN ">
    <arguments>
    </arguments>
</invoke>
```

2) 出参：

示例：

```
<return name=" BILLACCEPTORALLOWCASHIN ">
    <arguments>
        正常成功：
        <string id="ERROR">SUCCESS</string>
    </arguments>
</return>
```

六、 设置纸币入币金额（在调用“打开纸币器”后才能调用）

1) 入参：

示例：

```
<invoke name=" BILLACCEPTORSETCASHINMONEY ">
    <arguments>
```

```
        <string id=" AVAILABLE ">5,10,20,50,100 </string>
    </arguments>
</invoke>
```

2) 出参：

示例：

```
<return name=" BillAcceptorSetCashInMoney ">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
    </arguments>
</return>
```

七、 获取纸币器状态

1) 入参：

示例：

```
<invoke name=" BILLACCEPTORGETSTATUS">
    <arguments>
    </arguments>
</invoke>
```

2) 出参：

示例：

```
<return name=" BILLACCEPTORGETSTATUS">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
        <string id="STATUS">Error </string>
    </arguments>
</return>
```

8. 读卡器模块

一、 允许进卡

1) 入参：

示例：

```
<invoke name=" READCARDALLOWCARDIN ">  
  <arguments>  
    [<string id="ONLY_MAG">TRUE</string>]  
  </arguments>  
</invoke>
```

2) 出参：

示例：

```
<return name=" READCARDALLOWCARDIN ">  
  <arguments>  
    正常成功：  
    <string id="ERROR">SUCCESS</string>  
  </arguments>  
</return>
```

二、 禁止进卡

1) 入参：

示例：

```
<invoke name=" READCARDNOTALLOWCARDIN ">  
  <arguments>  
  </arguments>  
</invoke>
```

2) 出参：

示例：

```
<return name=" READCARDNOTALLOWCARDIN ">  
  <arguments>  
    正常成功：  
    <string id="ERROR">SUCCESS</string>  
  </arguments>
```

</return>

三、 检测卡

1) 入参:

示例:

```
<invoke name=" READCARDTESTINSERTCARD ">  
    <arguments>  
    </arguments>  
</invoke>
```

2) 出参 :

示例:

代码	名称	类型	注释
CARDSTACKSTATE	卡槽状态	String	0: 无卡 1: 有卡
ERROR	错误码	String	SUCCESS、DEVEERROR

```
<return name=" READCARDTESTINSERTCARD ">  
    <arguments>  
        正常成功:  
        <string id="ERROR">SUCCESS</string>  
        <string id="CARDSTACKSTATE">0</string>  
    </arguments>  
</return>
```

四、 读取磁条卡号

1) 入参:

示例:

```
<invoke name=" READCARDREADMAGCARDNO ">  
    <arguments>  
    </arguments>  
</invoke>
```

2) 出参 :

示例:

```
<return name=" READCARDREADMAGCARDNO ">
  <arguments>
    正常成功:
    <string id="ERROR">SUCCESS</string>
    <string id="CARDNO">123456</string>
  </arguments>
</return>
```

五、 移动到 IC 卡位

1) 入参:

示例:

```
<invoke name=" READCARDMOVECARDTOIC ">
  <arguments>
  </arguments>
</invoke>
```

2) 出参 :

示例:

```
<return name=" READCARDMOVECARDTOIC ">
  <arguments>
    正常成功:
    <string id="ERROR">SUCCESS</string>
  </arguments>
</return>
```

六、 退卡

1) 入参:

示例:

```
<invoke name=" READCARDOUTCARD">
  <arguments>
    [<string id="ONLY_MAG">TRUE</string>]
  </arguments>
</invoke>
```

2) 出参 :

示例:

```
<return name=" READCARDOUTCARD ">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
    </arguments>
</return>
```

七、 移动卡到射频位

1) 入参:

示例:

```
<invoke name=" READCARDMOVECARDTORF">
    <arguments>
    </arguments>
</invoke>
```

2) 出参 :

示例:

```
<return name=" READCARDMOVECARDTORF ">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
    </arguments>
</return>
```

八、 移动卡到卡后端

1) 入参:

示例:

```
<invoke name=" READCARDMOVECARDTOREAR">
    <arguments>
    </arguments>
</invoke>
```

2) 出参 :

示例:

```
<return name=" READCARDMOVECARDTOREAR ">
    <arguments>
```

正常成功：
<string id="ERROR">SUCCESS</string>
</arguments>
</return>

十二、 读射频卡数据

1) 入参：

代码	名称	类型	注释
SECTORNO	扇区号	String	
BLOCKNO	块号	String	
PASSWORD	认证密码	String	

示例：

```
<invoke name=" READCARDREADRFCARD ">  
  <arguments>  
    <string id="SECTORNO">0</string>  
    <string id="BLOCKNO">0</string>  
    <string id="PASSWORD">FFFFFFFFFFFF</string>  
  </arguments>  
</invoke>
```

2) 出参：

CARDNO	卡号	String	
--------	----	--------	--

示例：

```
<return name=" READCARDREADRFCARD ">  
  <arguments>  
    正常成功：  
    <string id="ERROR">SUCCESS</string>  
    故障：  
    <string id="ERROR">DEVEEROR</string>  
    <string id="CARDNO">12345678</string>  
  </arguments>  
</return>
```

十三、 非接触射频卡激活

入参：

示例：

```
<invoke name="READCARDCPUACTIVATE"><arguments></arguments></invoke>
```

出参：

示例：

```
<return name="READCARDREADRFCARD">  
    <arguments>  
        正常成功：  
        <string id="ERROR">SUCCESS</string>  
        故障：  
        <string id="ERROR">DEVEEROR</string>  
    </arguments>  
</return>
```

十四、 非接触射频卡发送命令

入参：

示例：

```
<invoke name="RFCPUSENDDATA">  
    <arguments>  
        <string id="SENDDATA">发送的命令字符串</string>  
    </arguments>  
</invoke>
```

出参：

示例：

```
<return name="RFCPUSENDDATA">  
    <arguments>  
        正常成功：  
        <string id="ERROR">SUCCESS</string>  
        故障：  
        <string id="ERROR">DEVEEROR</string>  
        <string id="RECEIVEDATA">收到的命令字符串</string>  
    </arguments>  
</return>
```

十五、 Ic 卡上电

入参：

示例：

```
<invoke name=" READCARDICPOWERON"><arguments></arguments></invoke>
```

出参：

示例：

```
<return name=" READCARDICPOWERON">
  <arguments>
    正常成功：
    <string id="ERROR">SUCCESS</string>
    故障：
    <string id="ERROR">DEVEEROR</string>
  </arguments>
</return>
```

十六、 Ic 卡下电

入参：

示例：

```
<invoke name=" READCARDICPOWEROFF"><arguments></arguments></invoke>
```

出参：

示例：

```
<return name=" READCARDICPOWEROFF">
  <arguments>
    正常成功：
    <string id="ERROR">SUCCESS</string>
    故障：
    <string id="ERROR">DEVEEROR</string>
  </arguments>
</return>
```

十七、 Ic 卡激活

入参：

示例：

```
<invoke name=" READCARDCPUACTIVATE"><arguments></arguments></invoke>
```

出参：

示例：

```
<return name=" READCARDCPUACTIVATE">
    <arguments>
        正常成功：
        <string id="ERROR">SUCCESS</string>
        故障：
        <string id="ERROR">DEVEEROR</string>
    </arguments>
</return>
```

十八、 Ic 卡释放

入参：

示例：

```
<invoke name=" READCARDCPUDEACTIVATE"><arguments></arguments></invoke>
```

出参：

示例：

```
<return name=" READCARDCPUDEACTIVATE">
    <arguments>
        正常成功：
        <string id="ERROR">SUCCESS</string>
        故障：
        <string id="ERROR">DEVEEROR</string>
    </arguments>
</return>
```

十九、 Ic 卡检测类型

入参：

示例：

```
<invoke name=" GETICCARDTYPE"><arguments></arguments></invoke>
```

出参：

示例：

```
<return name=" GETICCARDTYPE">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
        故障:
        <string id="ERROR">DEVEEROR</string>
        <string id=" IccType">ic 卡类型</string>
    </arguments>
</return>
```

二十、 Ic 卡发送命令

入参:

示例:

```
<invoke name=" READCARDPUTTRANSMIT">
    <arguments>
        <string id=" cmd">发送的命令字符串</string>
    </arguments>
</invoke>
```

出参 :

示例:

```
<return name=" READCARDPUTTRANSMIT">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
        故障:
        <string id="ERROR">DEVEEROR</string>
        <string id=" RecvData">收到的命令字符串</string>
    </arguments>
</return>
```

二十一、 读卡器发卡功能

入参:

示例:

```
<invoke name=" TAKEINFROMREAR">
    <arguments>
    </arguments>
</invoke>
```

出参：

示例：

```
<return name=" TAKEINFROMREAR ">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
        故障:
        <string id="ERROR">DEVEEROR</string>
    </arguments>
</return>
```

二十二、 读卡器读取社保卡功能

入参：

示例：

```
<invoke name=" READSOCIALCARDNO_DIRECTLY">
    <arguments>
    </arguments>
</invoke>
```

出参：

示例：

```
<return name=" READSOCIALCARDNO_DIRECTLY">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
        <string id=" SOCIALCARDNO ">12345678</string>
        故障:
        <string id="ERROR">DEVEEROR</string>
    </arguments>
</return>
```

9. 门禁灯模块

一、 开门禁灯

1) 入参:

示例:

```
<invoke name=" DOORLIGHTOPEN ">  
  <arguments>  
    <string id=" INDEX ">1</string>  
  </arguments>  
</invoke>
```

2) 出参 :

示例:

```
<return name=" DOORLIGHTOPEN ">  
  <arguments>  
    正常成功:  
    <string id="ERROR">SUCCESS</string>  
  </arguments>  
</return>
```

二、 关门禁灯

1) 入参:

示例:

```
<invoke name=" DOORLIGHTCLOSE ">  
  <arguments>  
    <string id=" INDEX ">1</string>  
  </arguments>  
</invoke>
```

2) 出参 :

示例:

```
<return name=" DOORLIGHTCLOSE ">  
  <arguments>  
    正常成功:  
    <string id="ERROR">SUCCESS</string>  
  </arguments>  
</return>
```

三、 闪门禁灯

1) 入参:

示例:

```
<invoke name=" DOORLIGHTFLUSH ">  
    <arguments>  
        <string id="INDEX">1</string>  
    </arguments>  
</invoke>
```

2) 出参 :

示例:

```
<return name=" DOORLIGHTFLUSH ">  
    <arguments>  
        正常成功:  
        <string id="ERROR">SUCCESS</string>  
    </arguments>  
</return>
```

10.推杆模块

一、 开推杆

1) 入参:

示例:

```
<invoke name=" DOORTGOPEN ">  
    <arguments>  
    </arguments>  
</invoke>
```

2) 出参 :

示例:

```
<return name=" DOORTGOPEN ">  
    <arguments>  
        正常成功:  
        <string id="ERROR">SUCCESS</string>
```

```
        </arguments>
    </return>
```

二、 关推杆

1) 入参:

示例:

```
<invoke name=" DOORTGCLOSE ">
    <arguments>
    </arguments>
</invoke>
```

2) 出参 :

示例:

```
<return name=" DOORTGCLOSE ">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
    </arguments>
</return>
```

11.门禁锁模块

一、 开门禁锁

1) 入参:

示例:

```
<invoke name=" LOCKOPEN">
    <arguments>
        <string id="INDEX">1</string>

    </arguments>
</invoke>
```

2) 出参 :

示例:

```
<return name=" LOCKOPEN">
```



```
        <arguments>
            正常成功:
            <string id="ERROR">SUCCESS</string>
        </arguments>
    </return>
```

二、 关门禁锁

1) 入参:

示例:

```
<invoke name="LOCKCLOSE">
    <arguments>
        <string id="INDEX">1</string>
    </arguments>
</invoke>
```

2) 出参 :

示例:

```
<return name=" LOCKCLOSE ">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
    </arguments>
</return>
```

12.病历打印模块

一、 病历打印机获取状态

1) 入参:

示例:

```
<invoke name="BLPRINTGETSTATUS">
</invoke>
```

2) 出参 :

示例:

```
<return name="BLPRINTGETSTATUS">
```

```
<arguments>
    正常成功:
    <string id="ERROR">SUCCESS</string>
    <string id="STATUS"></string>

</arguments>

</return>
```

二、 病历打印机打印(指定文件)

1) 入参:

示例:

```
<invoke name="BLPRINTPRINT">
</invoke>
```

2) 出参 :

示例:

```
<return name="BLPRINTPRINT">
    <arguments>
        正常成功:
        <string id="ERROR">SUCCESS</string>
        <string id="STATUS"></string>
    </arguments>

</return>
```

13.激光打印模块

一、 激光打印机获取状态

1) 入参:

示例:

```
<invoke name=" LASERPRINTSTATUS">
</invoke>
```

2) 出参 :

示例:

```
<return name=" LASERPRINTSTATUS ">
    <arguments>
```

正常成功:

```
<string id="ERROR">SUCCESS</string>
```

```
<string id="STATUS"></string>
```

```
</arguments>
```

```
</return>
```

可能的正常值: 空|Null|正常|Ready|准备就绪|正在打印|休眠|墨粉将用完|硒鼓寿命将到

14.金属键盘模块

一、 打开金属键盘开始输入

1) 入参:

示例:

```
<invoke name=" OPENPINPAD">
```

```
</invoke>
```

2) 出参 :

示例:

```
<return name=" OPENPINPAD ">
```

```
<arguments>
```

正常成功:

```
<string id="ERROR">SUCCESS</string>
```

```
</arguments>
```

```
</return>
```

二、 关闭金属键盘停止输入

1) 入参:

示例:

```
<invoke name="CLOSEPINPAD">
```

```
</invoke>
```

2) 出参 :

示例:

```
<return name=" CLOSEPINPAD ">
```

```
<arguments>
```

正常成功：

```
<string id="ERROR">SUCCESS</string>
</arguments>
</return>
```

15.找零机器

一、找零

1) 入参：

参数	名称	类型	注释
SUM	要找零的金额	String	

示例：

```
<invoke name="RECHARGE">
  <arguments>
    <string id="SUM">33</string>
  </arguments>
</invoke>
```

2) 出参：

参数	名称	类型	注释
REALSUM	实际找出的金额	String	返回结果先看该值
ERROR	错误码	String	REALSUM 和入参不一致时，描述错误的内容

示例：

```
<return name="RECHARGE">
  <arguments>
    <string id="ERROR">1 元硬币找零失败，找币超时|success</string>
    <string id="REALSUM">30</string>
  </arguments>
</return>
```

二、设置找零机中剩余纸币、硬币的张数，在原来基础上增加，如果对应的张数值不够，找零时会失败。

1) 入参:

参数	名称	类型	注释
COUNT	新加的现金数量	String	格式为: 硬币个数,10 元纸币张数,20 元纸币张数,30 元纸币张数

示例:

```
<invoke name="ADDCASHCOUNT">
  <arguments>
    <string id="COUNT">1,2,3,4</string>
  </arguments>
</invoke>
```

2) 出参 :

参数	名称	类型	注释
ERROR	错误码	String	成功: SUCCESS 失败: 错误描述
COUNTINFO	当前现金的数量	String	成功时, 格式为: 硬币个数,10 元纸币张数,20 元纸币张数,30 元纸币张数。 可以设置入参"0,0,0,0" 来获取当前的现金数量。

示例:

```
<return name="ADDCASHCOUNT">
  <arguments>
    <string id="ERROR">SUCCESS</string>
    <string id="COUNTINFO">2,8,14,20</string>
  </arguments>
</return>
```

三、清空 找零机中剩余纸币、硬币的张数。

1) 入参:

示例：

```
<invoke name="CLEARCASHCOUNT"><arguments></arguments></invoke>
```

2) 出参：

参数	名称	类型	注释
ERROR	错误码	String	成功：SUCCESS 失败：错误描述

示例：

```
<return name="ADDCASHCOUNT">  
<arguments><string id="ERROR">SUCCESS</string></arguments>  
</return>
```

七、 配置说明

随包会提供名为 SysConfig.xml 的参数配置文件，使用前先确保参数配置正确，具体参数配置说明如下：

Card2Port：二代证端口号，1~16 表示串口号，1001 表示 USB 口，0 表示自动选择，默认 1001

Card2TimeOut：二代证超时时间 毫秒，默认 20000 毫秒（20 秒）

SnapshotSavePath：视频截图保存位置，默认 C:\\XUHUI_ZZJ\\CameraSave

SnapshotInterval：视频截图时间间隔，单位毫秒，默认 100

SnapshotTime：视频截图时长 0 表示只取 1 张就返回 其他毫秒为单位，默认 10000 毫秒（10 秒）

TopCameraIndex：摄像头索引位置（0 或者 1）默认 0，可以通过设置 0 或者 1 看拍摄的照片确定正确的索引

BottomCameraIndex：摄像头索引位置（0 或者 1）默认 0，可以通过设置 0 或者 1 看拍摄的照片确定正确的索引

ScannnerPort：扫描仪端口号，默认 5，即 COM5

ScannnerTimeOut：扫描仪自动超时时间 毫秒

PrintTextPath：打印内容文件，默认 C:\\\\print.txt，当调用打印接口时，就回去打印读取指定的打印文件打印。

PrintStatePort: 凭条纸将尽检测端口号, 默认 1

LogSavePath: 日志保存路径, 默认 C:\\\\XUHUI_ZZJ\\\\Log

PostPort: 纸币器端口

ReadCardPort: 读卡器端口

ReadCardType: 读卡器类型

CardSenderName: 发卡器打印机名称

TGPort: 推杆端口

LockPort: 门锁端口

CardSenderPotential: 高抗低抗配置 0 为低抗, 1 为高抗

UserRecharge: 找零机是否启用, 1 为启用, 否则为禁用

CashRechargeType: 找零纸币器类型, GBM10, 广电运通为 GRGCDM

CoinRechargeType: 找零硬币器类型, NVHopper, , MK2Hopper

八、 测试说明

配置好配置文件后, 点击测试程序确保硬件测试通过。



