

# **ID100 身份证阅读器 SDK 使用手册**

## **(V1.0)**

---

**日期：2014 年 11 月 6 日**

# 修订记录

日期	版本	描述	作者
2015-02-05	1.0.0.2	增加获取状态接口及 SDK 加入版本信息	陈建兴
2014-11-06	1.0.0.1	基础版本	陈建兴

目录

简介 ..... 4

隐私权说明 ..... 4

系统需求 ..... 4

ID100 Reader SDK 安装 ..... 4

接口说明 ..... 4

    InitComm ..... 4

    InitCommExt ..... 5

    CloseComm ..... 6

    Authenticate ..... 7

    Read\_Content ..... 8

    GetSAMID ..... 9

    GetBmpPhoto ..... 9

    GetSAMStatus ..... 10

附录 ..... 11

    附录 1 ..... 11

    附录 2 ..... 12

    附录 3 ..... 13

    附录 4 ..... 13

参考文献 ..... 13

# 简介

感谢您使用中控 ID100 身份证阅读器 SDK, 在使用前请仔细阅读 ID100 Reader SDK 概述, 以便您能更快地掌握并使用 ID100 Reader SDK.

本文档描述了中控科技 ID100 身份证 SDK 接口说明以及简单调用例子。

# 隐私权说明

本公司将本软件程序的使用权授予您, 但您必须向本公司作以下保证: 不在本协议规定的条款之外, 使用、拷贝、修改、租赁或转让本系统或其中的任一部份。

# 系统需求

- (1) 操作系统: windows xp 及以上操作系统
- (2) 适用开发语言: C++、C#、VB 等

# ID100 Reader SDK 安装

- (1)、安装 SDK, 将 DLL\_File.dll, license.dat, sdtapi.dll, termb.dll 安装到程序所在目录。
- (2)、安装设备驱动(32 位 Windows 操作系统驱动/64 位 Windows 操作系统驱动)。

# 接口说明

## InitComm

[模块]

**Termb.dll**

[函数]

`int __stdcall InitComm(int nPort);`

[功能]

初始化端口

[参数说明]

nReverd

[IN]端口号；取值范围(1-16 或者 1001-1016)

[返回值]

返回 0，表示搜索失败；返回正数则为端口号

[示例]

**C++:**

```
int nPort = InitComm(1);
if (0 >= nPortRet)
{
    printf("InitComm failed!\r\n");
}
else
{
    printf("InitComm succss,port=%d!\r\n", nPortRet);
}
```

**C#:**

```
int nPort = InitComm(1);
if (0 >= nPortRet)
{
    Console.WriteLine("InitComm failed!");
}
else
{
    Console.WriteLine("InitComm succss,Port={0,D}!", nPortRet);
}
```

## InitCommExt

[模块]

**Termb.dll**

[函数]

```
int __stdcall InitCommExt();
```

[功能]

自动搜索，初始化端口

[返回值]

返回 0，表示搜索失败；返回正数则为端口号

[示例]

**C++:**

```
int nPortRet = InitCommExt();
if (0 >= nPortRet)
{
    printf("InitCommExt failed!\r\n");
}
```

```

    }
    else
    {
        printf("InitCommExt success,port=%d!\r\n", nPortRet);
    }

```

#### **C#:**

```

int nPortRet = InitCommExt();
if (0 >= nPortRet)
{
    Console.WriteLine("InitCommExt failed!");
}
else
{
    Console.WriteLine("InitCommExt success,Port={0,D}!", nPortRet);
}

```

## CloseComm

[模块]

**Termb.dll**

[函数]

`int __stdcall CloseComm()`

[功能]

关闭端口，断开 pc 到阅读器的连接

[返回值]

返回 1 表示关闭端口成功，返回其他值表示错误

[示例]

#### **C++:**

```

int nRet = CloseComm();
if (1 != nRet)
{
    printf("CloseComm failed,errno=%d!\r\n", nRet);
}
else
{
    printf("CloseComm succss!\r\n");
}

```

#### **C#:**

```

int nRet = CloseComm();
if (1 != nRet)
{
    Console.WriteLine("CloseComm failed,errno={0,D}!", nRet);
}

```

```

else
{
    Console.WriteLine("CloseComm succss!");
}

```

## Authenticate

[模块]

**Termb.dll**

[函数]

`int __stdcall Authenticate()`

[功能]

卡认证，本函数用于读卡器和卡片之间的合法身份确认

[返回值]

返回 1 表示正确放置卡，返回其他表示未放卡或者卡片放置不正确

[备注]

调用 **InitComm** 或 **InitCommEx** 方可调用本接口

[示例]

**C++:**

```

int nRet = Authenticate();
if (1 != nRet)
{
    printf("Authenticate failed,errno=%d!\r\n", nRet);
}
else
{
    printf("Authenticate succss!\r\n");
}

```

**C#:**

```

int nRet = Authenticate();
if (1 != nRet)
{
    Console.WriteLine("Authenticate failed,errno={0,D}!", nRet);
}
else
{
    Console.WriteLine("Authenticate succss!");
}

```

## Read\_Content

[模块]

**Termb.dll**

[函数]

`int __stdcall Read_Content(int Active)`

[功能]

读卡操作，信息文件存储在应用程序当前目录

[参数说明]

**Active**

[IN] 1 表示读基本信息，形成文字信息文件 **WZ.TXT**、相片文件 **XP.WLT**、**ZP.BMP**，如果有指纹信息形成指纹信息文件 **FP.DAT**；2 表示只读文字信息，形成文字信息文件 **WZ.TXT** 和相片文件 **XP.WLT**；3 读最新住址信息，形成最新住址文件 **NEWADD.TXT**

[返回值]

返回值为 1 正确；0 读卡错误；2 没有最新住址信息；3 正确，并且有指纹信息；-1 相片解码错误；-2 wlt 文件后缀错误；-3 wlt 文件打开错误；-4 wlt 文件格式错误；-5 软件未授权；-11 无效参数

[备注]

调用 **Authenticate** 成功后方可调用

读取到身份证信息以文件形式保存在程序根目录下：

**WZ.TXT**:身份证文字描述信息，具体见附录 2

**XP.WLT**:照片文件，可调用照片解析库获取身份证头像照片文件。

**FP.DAT**:指纹特征码模板文件

[示例]

**C++:**

```
int nRet = Read_Content(1);
if (1 != nRet)
{
    printf("Read_Content failed,errno=%d!\r\n", nRet);
}
else
{
    printf("Read_Content succss!\r\n");
}
```

**C#:**

```
int nRet = Read_Content(1);
if (1 != nRet)
{
    Console.WriteLine("Read_Content failed,errno={0,D}!", nRet);
}
else
{
    Console.WriteLine("Read_Content succss!");
}
```



# GetSAMID

[模块]

**Termb.dll**

[函数]

```
int __stdcall GetSAMID(char *SAMID);
```

[功能]

获取 SAM 模块的 ID 号

[参数说明]

SAMID

[OUT]返回 SAM 编号的字符串缓冲区

[返回值]

返回值为 1 成功；0 为失败。

[备注]

调用 **InitComm** 或 **InitCommEx** 方可调用本接口

[示例]

**C++:**

```
char szSAMID[100] = {0x0};
int nRet = GetSAMID(szSAMID);
if (1 != nRet)
{
    printf("GetSAMID failed,errno=%d!\r\n", nRet);
}
else
{
    printf("GetSAMID succss, Samid=%s!\r\n", szSAMID);
}
```

**C#:**

```
StringBuilder strSAMID = new StringBuilder(100);
int nRet = GetSAMID(strSAMID);
if (1 != nRet)
{
    Console.WriteLine("GetSAMID failed,errno={0,D}!", nRet);
}
else
{
    Console.WriteLine("GetSAMID succss" + strSAMID);
}
```

## GetBmpPhoto

[模块]

### **Termb.dll**

[函数]

```
int __stdcall GetBmpPhoto(char *path);
```

[功能]

解析身份证照片

[参数说明]

path

[IN] **xp.wlt** 文件路径

[返回值]

返回值为 1 成功,并在当前目录生成 zp.bmp(102\*126 24bit); 0 为失败。

[备注]

调用 **Authenticate** 方可调用本接口

[示例]

**C++:**

```
char *pPath = "xp.wlt";
int nRet = GetBmpPhoto(pPath);
if (1 != nRet)
{
    printf("GetBmpPhoto failed, errno=%d!\r\n", nRet);
}
else
{
    printf("GetBmpPhoto succss, image name=zp.bmp!\r\n");
}
```

**C#:**

```
String strPath = "xp.wlt";
int nRet = GetBmpPhoto(strPath);
if (1 != nRet)
{
    Console.WriteLine("GetBmpPhoto failed, errno = {0,D}!", nRet);
}
else
{
    Console.WriteLine("GetBmpPhoto succss, image name=zp.bmp!");
}
```

## GetSAMStatus

[模块]

### **Termb.dll**

[函数]

```
int __stdcall GetSAMStatus();
```

[功能]

获取 SAM 模块状态

[参数说明]

[返回值]

返回值为 1 成功,0 为失败。

[备注]

自动读卡软件可能中间设备状态异常或者拔插,可通过判断该状态主动断开并重连设备。

[示例]

**C++:**

```
int nRet = GetSAMStatus();
if (1 != nRet)
{
    printf("GetSAMStatus failed,errno=%d!\r\n", nRet);
}
else
{
    printf("GetSAMStatus success!\r\n");
}
```

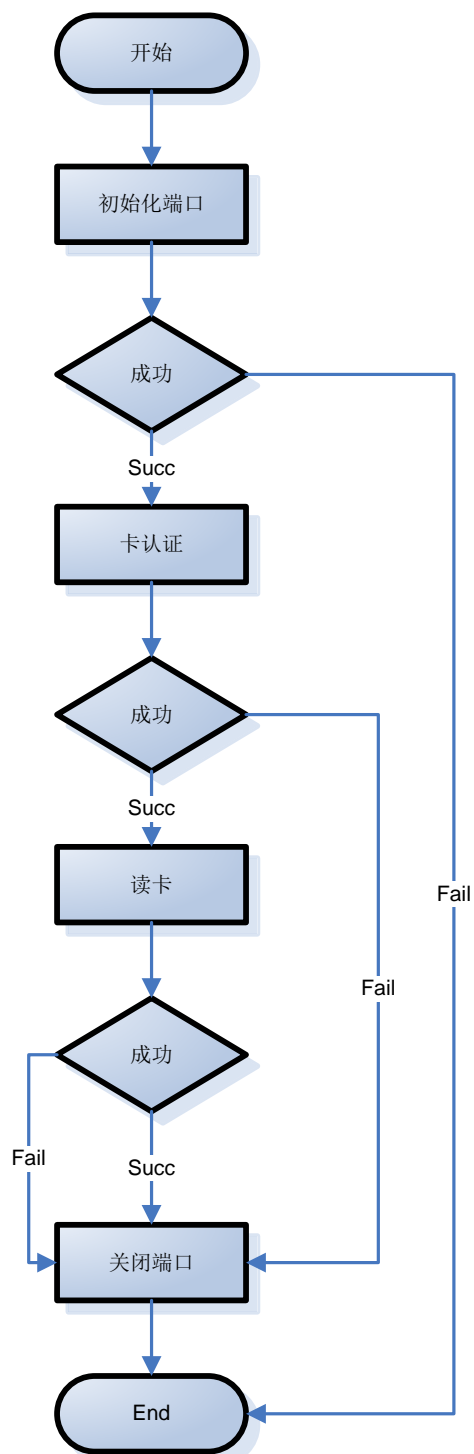
**C#:**

```
int nRet = GetSAMStatus();
if (1 != nRet)
{
    Console.WriteLine("GetSAMStatus failed,errno={0,D}!", nRet);
}
else
{
    Console.WriteLine("GetSAMStatus succss!");
}
```

## 附录

### 附录 1

调用流程



## 附录 2

文字信息文件(**wz.txt**)大小 256 字节, 采用 GB 13000 的 UCS-2 编码格式

字段名	长度 (字节)	描述
姓名	30	姓名
性别代码	2	见附录 4

民族代码	4	见附录 3
出生日期	16	年月日: YYYMMDD
住址	70	汉字和数字
公民身份号码	36	数字
签发机关	30	汉字
有效期起始日期	16	年月日: YYYMMDD
有效期截止日期	16	年月日: YYYMMDD 有效期为长期时存储“长期”
	36	预留区

### 附录 3

民族代码对照表

01	汉	15	土家	29	柯尔克孜	43	乌孜别克
02	蒙古	16	哈尼	30	土	44	俄罗斯
03	回	17	哈萨克	31	达斡尔	45	鄂温克
04	藏	18	傣	32	仫佬	46	德昂
05	维吾尔	19	黎	33	羌	47	保安
06	苗	20	傈僳	34	布朗	48	裕固
07	彝	21	佤	35	撒拉	49	京
08	壮	22	畲	36	毛南	50	塔塔尔
09	布依	23	高山	37	仡佬	51	独龙
10	朝鲜	24	拉祜	38	锡伯	52	鄂伦春
11	满	25	水	39	阿昌	53	赫哲
12	侗	26	东乡	40	普米	54	门巴
13	瑶	27	纳西	41	塔吉克	55	珞巴
14	白	28	景颇	42	怒	56	基诺

### 附录 4

性别代码对照表

0	未知	2	女
01	男	9	未说明

## 参考文献

[1] 《居民身份证机读信息规范》-GA 中华人民共和国公共安全行业标准-GA 490—2012。中华人民共和国公安部，2012 年 9 月。