

# 门禁考勤机标准 SDK 函数介绍

版本：V2.3

时间：20190429

# 目录

## 一、 SDK 使用说明

## 二、 串口通讯 comp\_d11 类

1. 打开串口指令 open\_Com
2. 关闭串口指令 Close\_Com
3. 打开设备指令 Device\_Open
4. 关闭设备指令 Device\_Close
5. 获取设备地址指令 Get\_Deviceaddr\_Config
6. 设置设备地址指令 Set\_Deviceaddr\_Config
7. 获取设备网络设置指令 Get\_Network\_Config
8. 设置设备网络指令 Set\_Network\_Config
9. 获取设备韦根指令 Get\_Wiegand\_Config
10. 设置设备韦根指令 Set\_Wiegand\_Config
11. 获取设备接口类型指令 Get\_Porttype\_Config
12. 设置设备接口类型指令 Set\_Porttype\_Config
13. 获取设备连接密码指令 Get\_password\_Config
14. 设置设备连接密码指令 Set\_password\_Config
15. 获取设备 ID 指令 Get\_ID\_Config
16. 获取设备验证模式指令 Get\_verifytype\_Config
17. 设置设备验证模式指令 Set\_verifytype\_Config
18. 获取设备日期时间指令 Get\_time\_Config
19. 设置设备日期时间指令 Set\_time\_Config
20. 获取设备开门时间指令 Get\_opentime\_Config
21. 设置设备开门时间指令 Set\_opentime\_Config
22. 获取设备警报设置指令 Get\_alarmmode\_Config
23. 设置设备警报设置指令 Set\_alarmmode\_Config
24. 获取设备系统语言设置指令 Get\_lng\_Config
25. 设置设备系统语言指令 Set\_lng\_Config
26. 获取用户登记信息同步 Get\_update\_Config
27. 获取所有用户信息指令 Get\_userinfo\_Config
28. 删除指定 ID 用户指令 Set\_Deletid\_Config
29. 删除所有用户指令 Set\_DeletAllid\_Config
30. 获取按指定日期范围获取考勤信息 Get\_kaoqin\_sum\_Config
31. 获取所有考勤信息 Get\_kaoqin\_ALLsum\_Config

- 32.删除所有考勤信息 Set\_DeletALL\_Config:
- 33.下载指定 finger id 模板信息（由设备到上位机） Get\_usermod\_Config
- 34.上传指定 finger id 模板信息（由上位机到设备） set\_usermod\_Config
- 35.上传一个用户登记信息（不包含指静脉模板） Set\_regedit\_Config
- 36.进入 bootloader 命令 Set\_Bootloader\_Config
37. 删除指定模板指令 Set\_DeletMode\_Config

### 三、 网络通讯 client\_dll 类

- 1.打开网口指令 open\_Com
2. 关闭网口指令 Close\_Com
3. 其他指令

### 四、 附录：（用户信息结构）

1. 用户状态:
2. 用户类型
3. 指静脉 1ID
4. 指静脉 2ID
5. 指静脉 3ID
6. 指静脉 4ID
7. 用户密码
8. 用户密码长度
9. 卡号

## 一、 SDK 使用说明

### 1. 总体原则，通过返回字符串反映执行状态：

- "0" 操作成功
- "-1" 操作失败
- "-2" 操作发生异常
- "-3" 操作超时
- "-4" 返回数据错误
- "-5" 输入参数错误
- "-6" 空值

如果要获取具体数据并且操作成功，则返回具体数据字符串

### 2. 网络通讯 TCP/IP 协议, 使用举例：

创建新对象 `COMMUNICATION.client_dll aClass=new COMMUNICATION.client_dll();`  
打开网口 `aClass.open_Com("192.168.1.88","5000");`  
连接设备 `aClass.Device_Open("123456");`  
使用函数 `aClass.Get_Network_Config();`

### 3. 串口通讯, 使用举例：

创建新对象 `COMMUNICATION.comp_dll aClass= new COMMUNICATION.comp_dll();`  
打开串口 `aClass.open_Com("com8");`  
连接设备 `aClass.Device_Open("123456");`  
使用函数 `aClass.Get_Network_Config();`

## 二、 串口通讯 comp\_dll 类

### 1 打开串口指令 `string open_Com(string Address):`

参数：串口名 Address

举例：`aClass.open_Com("COM11");`

说明：使用默认波特率 9600

## 2. 关闭串口指令 `string Close_Com()`:

举例: `aClass.Close_Com()`;

## 3. 打开设备指令 `string Device_Open(string sst)`:

参数: sst 必须为设备密码, 6 位数字字符串

举例: `aClass.Device_Open("123456")`;

## 4. 关闭设备指令 `string Device_Close()`:

举例: `aClass.Device_Close()`;

## 5. 获取设备地址指令 `string Get_Deviceaddr_Config()`:

举例: `aClass.Get_Deviceaddr_Config()`;

返回: 设备地址如:"2"

## 6. 设置设备地址指令 `string Set_Deviceaddr_Config(string sst)`:

参数: sst 必须为地址值数字字符串如"3"

举例: `aClass.Set_Deviceaddr_Config("3")`;

## 7. 设置设备网络指令 `string Set_Network_Config(string sst)`:

参数: sst 为: ip 地址 + 端口号 + 子网掩码 + 网关的字符串:

“192,,168,1,88,5000,255,255,255, 0,1,1,1,1”

举例: `aClass.Set_Network_Config("192,168,1,88,5000,255,255,255,0,1,1,1,1")`;

## 8. 获取设备网络设置指令 `string Get_Network_Config()`:

举例: `aClass.Get_Network_Config()`;

返回: 返回 ip 地址 + 端口号 + 子网掩码 + 网关 + MAC 地址的字符串:

“192,,168,1,88,5000,255,255,255, 0,1,1,1,1,48, 255, 214, 5, 77, 70”

## 9. 获取设备韦根指令 `string Get_Wiegand_Config()`:

举例: `aClass.Get_Wiegand_Config()`;

返回: 设备韦根字符串: “1”,代表韦根 26; “2”,代表韦根 34

## 10. 设置设备韦根指令 `string Set_Wiegand_Config(string sst)`:

参数: `sst` 为设备韦根字符串: “1”,代表韦根 26; “2”,代表韦根 34

举例: `aClass.Set_Wiegand_Config("1")`;

## 11. 获取设备接口类型指令 `string Get_Porttype_Config()`:

举例: `aClass.Get_Porttype_Config()`;

返回: 设备接口类型字符串: “1”代表网口; “2”代表 RS485; “3”代表韦根

## 12. 设置设备接口类型指令 `string Set_Porttype_Config(string sst)`:

参数: `sst` 是设备接口类型字符串: “1”代表网口; “2”代表 RS485; “3”代表韦根

举例: `aClass.Set_Porttype_Config("1")`;

## 13. 获取设备连接密码指令 `string Get_password_Config()`:

举例: `aClass.Get_password_Config()`;

返回: 设备连接密码如: "123456"

说明: 默认密码为: 123456

## 14. 设置设备连接密码指令 `string Set_password_Config(string sst)`:

参数: `sst` 必须为 6 位数字字符串

举例: `aClass.Set_password_Config("000000")`;

## 15. 获取设备 ID 指令 `string Get_ID_Config()`:

举例: `aClass.Get_ID_Config()`;

返回: 8 位十六进制字符串, 如 “1805290E”

## 16. 获取设备验证模式指令 `string Get_verifytype_Config()`:

举例: `aClass.Get_verifytype_Config()`;

返回: 设备验证模式字符串:

"1"代表指静脉、密码、IC 卡三选一

"2"代表指静脉 + 密码

"3"代表指指静脉 + IC 卡

## 17. 设置设备验证模式指令 `string Set_verifytype_Config(string sst)`:

参数: "1"代表指静脉、密码、IC 卡三选一

"2"代表指静脉 + 密码

"3"代表指指静脉 + IC 卡

举例: `aClass.Set_verifytype_Config("2")`;

## 18. 获取设备日期时间指令 `string Get_time_Config()`:

举例: `aClass.Get_time_Config()`;

返回: "2018,3,14,17,18,49", 表示: 年-月-日-时-分-秒

## 19. 设置设备日期时间指令 `string Set_time_Config(string sst)`:

参数: sst 必须为日期时间数字字符串,如"2018,2,2,18,0,28", 代表 2018.02.02 18: 00: 28 秒

举例: `aClass.Set_time_Config("2018,3,14,17,18,49")`;

## 20. 获取设备开门时间指令 `string Get_opentime_Config()`:

举例: `aClass.Get_opentime_Config()`;

返回: 设备开门时间字符串如: “8”, 代表开门时间为 8 秒

## 21. 设置设备开门时间指令 `string Set_opentime_Config(string sst):`

参数: sst 必须为设备开门时间数字字符串(取值范围 1-255 秒)

举例: `aClass.Set_opentime_Config("9");`

## 22. 获取设备警报设置指令 `string Get_alarmmode_Config():`

举例: `aClass.Get_alarmmode_Config();`

返回: 设备警报设置字符串:"1"

说明: “1”代表警报开; “2”代表警报关

## 23. 设置设备警报设置指令 `string Set_alarmmode_Config(string sst):`

参数: sst 必须为数字字符串: "1"代表警报开; "2"代表警报关

举例: `aClass.Set_alarmmode_Config("2");`

## 24. 获取设备系统语言设置指令 `string Get_lng_Config():`

举例: `aClass.Get_lng_Config();`

返回: 设备系统语言设置字符串:"1"

说明: “1”英语; “2”代表中文简体; “3”代表中文繁体

## 25. 设置设备系统语言指令 `string Set_lng_Config(string sst):`

参数: sst 必须为数字字符串: "1"英语; "2"代表中文简体; "3"代表中文繁体

举例: `aClass.Set_lng_Config("2");`

## 26. 获取用户登记信息同步 `string Get_update_Config(string flag):`

说明: flag:0x01:获取用户更新 id,返回最近变动的用户 ID 字符串集;

0x02:删除用户更新, 用户信息状态由其它状态(修改, 添加, 删除), 变成一般状态。

举例: `aClass.Get_update_Config("1");`



返回: "24,26,27"

举例: `aClass.Get_update_Config("2");`

返回: "0"

## 27. 获取所有用户信息指令 `string Get_userinfo_Config(string sst):`

举例: `aClass.Get_userinfo_Config("2");`

返回: "0,3,21,1,0,0,0,0000000000000000,0,000000"

说明: 返回用户信息数据结构

"flag,userType,userid,FingerID1,FingerID2,FingerID3,FingerID4,password,password\_len,rfid"

关于用户信息数据结构见附录。

## 28. 删除指定 ID 用户指令 `string Set_Deletid_Config(string sst):`

参数: sst 必须为代表用户 ID 的数字字符串

举例: `aClass.Set_Deletid_Config("21");`

## 29. 删除所有用户指令 `string Set_DeletAllid_Config():`

举例: `aClass.Set_DeletAllid_Config();`

说明: 同时会删除所有的模板

## 30. 获取按指定日期范围获取考勤信息

`string Get_kaoqin_sum_Config(string fl, string sst)`

参数: fl:为“1”时获取考勤记录条数, sst 必须是表示时间字符串: "date\_1, date\_2"。

(date\_1:起始日期, date\_2:结束日期。date\_2 必须大于或等于 date\_1)

fl:为“2”时, 获取一般考勤信息, sst 必须是第几帧记录的字符串

首先, 要获取用户考勤信息数量:

举例: `aClass.Get_kaoqin_sum_Config("1","2018,03,01,2018,03,13");`

返回: "5"

其次, 获取考勤信息:每条命令对应一帧考勤信息:

举例: `aClass.Get_kaoqin_sum_Config("2","3");`

返回: 22,2018,5,3,17,12,6,1;22,2018,5,3,17,12,7,1

说明: "22"是 ID 号 "2018,3,13,11,44,2"表示时间 "1"是用户类型 用分号;分开不同单元的考勤记录  
每帧考勤记录最多 32 条 (256 字节), 不足 32 条按实际长度发送。

### 31. 获取所有考勤信息 string Get\_kaoqin\_ALLsum\_Config(string fl, string sst)

参数: fl:为 "1" 时获取考勤记录条数, sst 可以为任意字符串

fl:为 "2" 时, 获取一般考勤信息, sst 必须是第几帧记录的字符串

首先, 要获取用户考勤信息数量:

举例: aClass.Get\_kaoqin\_ALLsum\_Config("1","0");

返回: "5"

其次, 获取考勤信息;每条命令对应一帧考勤信息:

举例: aClass.Get\_kaoqin\_ALLsum\_Config("2","3");

返回: 22,2018,5,3,17,12,6,1;22,2018,5,3,17,12,7,1

说明: "22"是 ID 号 "2018,3,13,11,44,2"表示时间 "1"是用户类型 用分号;分开不同单元的考勤记录  
每帧考勤记录最多 32 条 (256 字节), 不足 32 条按实际长度发送。

### 32. 删除所有考勤信息 string Set\_DeletALL\_Config():

举例: aClass.Set\_DeletALL\_Config();

### 33. 下载指定 finger id 模板信息 (由设备到上位机) string Get\_usermod\_Config(string a)

参数: a 代表指纹模板编号

举例: aClass.Get\_usermod\_Config("1");

说明: 返回的指纹模板为 16 进制字符串每两个字符表示一个 byte 的数据, 返回正确字符串大小为固定长度  
6704

### 34. 上传指定 finger id 模板信息 (由上位机到设备) string set\_usermod\_Config(string finger\_id, string a)

参数: finger\_id 代表指纹模板编号, a 表示指纹模板字符串

举例: aClass.set\_usermod\_Config(10, Mrecord);

说明: a 为 16 进制字符串, 每两个字符表示一个 byte 的数据, 正确的数据大小为固定长度 6704



## 四、 附录：（用户信息结构）

### 1. 用户状态：

flag(1 个字节)

"0"代表一般，此项标记的用户数据，表示无变动，同步时不会上传

"1"代表修改

"2"代表添加

"3"代表删除

### 2. 用户类型

userType(1 个字节)

"1"代表管理员

"2"代表临时用户

"3"代表普通用户

### 3. 指静脉 1ID

FingerID1(4 个字节)

### 4. 指静脉 2ID

FingerID2(4 个字节)

### 5. 指静脉 3ID

FingerID1(4 个字节)

### 6. 指静脉 4ID

FingerID1(4 个字节)

### 7. 用户密码

password(16 个字节)

## 8. 用户密码长度

password\_len(1 个字节)

## 9. 卡号

rfid(6 个字节)