

FKAttend 使用说明书

(版本 2.876)

2011. 7

目录

| | | |
|--------|--|----|
| 1 | 概要 | 8 |
| 2 | FKAttend.OCX 的接口 | 9 |
| 2.1 | 机器连接与断开 | 9 |
| 2.1.1 | ConnectComm | 9 |
| 2.1.2 | ConnectNet | 9 |
| 2.1.3 | ConnectUSB | 10 |
| 2.1.4 | DisConnect | 10 |
| 2.1.5 | ConnectGetIP | 10 |
| 2.2 | 登记数据管理 | 10 |
| 2.2.1 | GetEnrollData | 10 |
| 2.2.2 | GetEnrollDataWithString | 11 |
| 2.2.3 | PutEnrollData | 11 |
| 2.2.4 | PutEnrollDataWithString | 11 |
| 2.2.5 | SaveEnrollData | 12 |
| 2.2.6 | DeleteEnrollData | 12 |
| 2.2.7 | USBReadAllEnrollDataFromFile | 12 |
| 2.2.8 | USBReadAllEnrollDataCount | 12 |
| 2.2.9 | USBGetOneEnrollData | 13 |
| 2.2.10 | USBGetOneEnrollDataWithString | 13 |
| 2.2.11 | USBSetOneEnrollData | 13 |
| 2.2.12 | USBSetOneEnrollDataWithString | 14 |
| 2.2.13 | USBWriteAllEnrollDataToFile | 14 |
| 2.2.14 | ReadAllUserID | 14 |
| 2.2.15 | GetAllUserID | 15 |
| 2.2.16 | EmptyEnrollData | 15 |
| 2.2.17 | ClearKeeperData | 15 |
| 2.2.18 | BenumbAllManager | 15 |
| 2.2.19 | GetVerifyMode | 16 |
| 2.2.20 | SetVerifyMode | 16 |
| 2.2.21 | USBGetOneEnrollData_1 | 16 |
| 2.2.22 | USBGetOneEnrollDataWithString_1 | 17 |
| 2.2.23 | USBSetOneEnrollData_1 | 17 |
| 2.2.24 | USBSetOneEnrollDataWithString_1 | 17 |
| 2.2.25 | USBReadAllEnrollDataFromFile_Color | 18 |
| 2.2.26 | USBWriteAllEnrollDataToFile_Color | 18 |

| | | |
|--------|--|----|
| 2.2.27 | USBGetOneEnrollData_Color..... | 18 |
| 2.2.28 | USBGetOneEnrollDataWithString_Color..... | 19 |
| 2.2.29 | USBSetOneEnrollData_Color..... | 19 |
| 2.2.30 | USBSetOneEnrollDataWithString_Color..... | 20 |
| 2.3 | 记录数据管理 | 20 |
| 2.3.1 | LoadSuperLogData..... | 20 |
| 2.3.2 | USBLoadSuperLogDataFromFile..... | 20 |
| 2.3.3 | GetSuperLogData..... | 21 |
| 2.3.4 | EmptySuperLogData..... | 21 |
| 2.3.5 | LoadGeneralLogData..... | 22 |
| 2.3.6 | USBLoadGeneralLogDataFromFile..... | 22 |
| 2.3.7 | GetGeneralLogData..... | 22 |
| 2.3.8 | EmptyGeneralLogData..... | 23 |
| 2.3.9 | GetGeneralLogData_1..... | 23 |
| 2.3.10 | GetSuperLogData_1..... | 24 |
| 2.3.11 | GetRealTimeInfo..... | 25 |
| 2.3.12 | SetRealTimeInfo..... | 25 |
| 2.4 | 用户信息管理 | 26 |
| 2.4.1 | EnableUser..... | 26 |
| 2.4.2 | ModifyPrivilege..... | 26 |
| 2.4.3 | GetUserName..... | 26 |
| 2.4.4 | SetUserName..... | 26 |
| 2.4.5 | GetNewsMessage..... | 27 |
| 2.4.6 | SetNewsMessage..... | 27 |
| 2.4.7 | GetUserNewsID..... | 27 |
| 2.4.8 | SetUserNewsID..... | 27 |
| 2.5 | 机器管理 | 29 |
| 2.5.1 | EnableDevice..... | 29 |
| 2.5.2 | PowerOnAllDevice..... | 29 |
| 2.5.3 | PowerOffDevice..... | 29 |
| 2.5.4 | GetDeviceTime..... | 29 |
| 2.5.5 | SetDeviceTime..... | 29 |
| 2.5.6 | GetDeviceStatus..... | 30 |
| 2.5.7 | GetDeviceInfo..... | 30 |
| 2.5.8 | SetDeviceInfo..... | 31 |
| 2.5.9 | GetProductData..... | 31 |
| 2.5.10 | GetDeviceVersion..... | 31 |
| 2.5.11 | GetDeviceTime_1..... | 32 |
| 2.5.12 | SetDeviceTime_1..... | 32 |
| 2.6 | 响铃管理 | 33 |
| 2.6.1 | GetBellTime..... | 33 |

| | | |
|--------|---------------------------------|----|
| 2.6.2 | GetBellTimeWithString..... | 33 |
| 2.6.3 | SetBellTime..... | 33 |
| 2.6.4 | SetBellTimeWithString..... | 33 |
| 2.7 | 门铃管理 | 34 |
| 2.7.1 | GetDoorStatus..... | 34 |
| 2.7.2 | SetDoorStatus..... | 34 |
| 2.7.3 | GetPassTime..... | 34 |
| 2.7.4 | GetPassTimeWithString..... | 35 |
| 2.7.5 | SetPassTime..... | 35 |
| 2.7.6 | SetPassTimeWithString..... | 35 |
| 2.7.7 | GetUserPassTime..... | 35 |
| 2.7.8 | GetUserPassTimeWithString..... | 36 |
| 2.7.9 | SetUserPassTime..... | 36 |
| 2.7.10 | SetUserPassTimeWithString..... | 36 |
| 2.7.11 | GetGroupPassTime..... | 36 |
| 2.7.12 | GetGroupPassTimeWithString..... | 37 |
| 2.7.13 | SetGroupPassTime..... | 37 |
| 2.7.14 | SetGroupPassTimeWithString..... | 37 |
| 2.7.15 | GetGroupMatch..... | 37 |
| 2.7.16 | GetGroupMatchWithString..... | 38 |
| 2.7.17 | SetGroupMatch..... | 38 |
| 2.7.18 | SetGroupMatchWithString..... | 38 |
| 2.8 | 夏天时间制管理..... | 39 |
| 2.8.1 | GetAdjustInfo..... | 39 |
| 2.8.2 | SetAdjustInfo..... | 39 |
| 2.9 | 机器信息管理 | 40 |
| 2.9.1 | GetServerNetInfo..... | 40 |
| 2.9.2 | SetServerNetInfo..... | 40 |
| 2.9.3 | SetUSBModel..... | 41 |
| 3 | FKAttend.DLL 的界面..... | 42 |
| 3.1 | 机器连接与断开..... | 42 |
| 3.1.1 | FK_ConnectComm..... | 42 |
| 3.1.2 | FK_ConnectNet..... | 42 |
| 3.1.3 | FK_ConnectUSB..... | 42 |
| 3.1.4 | FK_DisConnect..... | 42 |
| 3.1.5 | FK_ConnectGetIP..... | 42 |
| 3.2 | 登记数据管理 | 43 |
| 3.2.1 | FK_GetEnrollData..... | 43 |
| 3.2.2 | FK_GetEnrollDataWithString..... | 43 |
| 3.2.3 | FK_PutEnrollData..... | 43 |
| 3.2.4 | FK_PutEnrollDataWithString..... | 43 |

| | | |
|--------|---|----|
| 3.2.5 | FK_SaveEnrollData..... | 43 |
| 3.2.6 | FK_DeleteEnrollData..... | 44 |
| 3.2.7 | FK_USBReadAllEnrollDataFromFile..... | 44 |
| 3.2.8 | FK_USBReadAllEnrollDataCount..... | 44 |
| 3.2.9 | FK_USBGetOneEnrollData..... | 44 |
| 3.2.10 | FK_USBGetOneEnrollDataWithString..... | 44 |
| 3.2.11 | FK_USBSetOneEnrollData..... | 44 |
| 3.2.12 | FK_USBSetOneEnrollDataWithString..... | 45 |
| 3.2.13 | FK_USBWriteAllEnrollDataToFile..... | 45 |
| 3.2.14 | FK_ReadAllUserID..... | 45 |
| 3.2.15 | FK_GetAllUserID..... | 45 |
| 3.2.16 | FK_EmptyEnrollData..... | 45 |
| 3.2.17 | FK_ClearKeeperData..... | 46 |
| 3.2.18 | FK_BenumbAllManager..... | 46 |
| 3.2.19 | FK_GetVerifyMode..... | 46 |
| 3.2.20 | FK_SetVerifyMode..... | 46 |
| 3.2.21 | FK_USBGetOneEnrollData_1..... | 46 |
| 3.2.22 | FK_USBGetOneEnrollDataWithString_1..... | 46 |
| 3.2.23 | FK_USBSetOneEnrollData_1..... | 47 |
| 3.2.24 | FK_USBSetOneEnrollDataWithString_1..... | 47 |
| 3.2.25 | FK_USBReadAllEnrollDataFromFile_Color..... | 47 |
| 3.2.26 | FK_USBWriteAllEnrollDataToFile_Color..... | 47 |
| 3.2.27 | FK_USBGetOneEnrollData_Color..... | 47 |
| 3.2.28 | FK_USBGetOneEnrollDataWithString_Color..... | 48 |
| 3.2.29 | FK_USBSetOneEnrollData_Color..... | 48 |
| 3.2.30 | FK_USBSetOneEnrollDataWithString_Color..... | 48 |
| 3.3 | 记录数据管理 | 49 |
| 3.3.1 | FK_LoadSuperLogData..... | 49 |
| 3.3.2 | FK_USBLoadSuperLogDataFromFile..... | 49 |
| 3.3.3 | FK_GetSuperLogData..... | 49 |
| 3.3.4 | FK_EmptySuperLogData..... | 49 |
| 3.3.5 | FK_LoadGeneralLogData..... | 49 |
| 3.3.6 | FK_USBLoadGeneralLogDataFromFile..... | 49 |
| 3.3.7 | FK_GetGeneralLogData..... | 50 |
| 3.3.8 | FK_EmptyGeneralLogData..... | 50 |
| 3.3.9 | FK_GetGeneralLogData_1..... | 50 |
| 3.3.10 | FK_GetSuperLogData_1..... | 50 |
| 3.3.11 | FK_GetRealTimeInfo..... | 50 |
| 3.3.12 | FK_SetRealTimeInfo..... | 51 |
| 3.4 | 用户信息管理 | 51 |
| 3.4.1 | FK_EnableUser..... | 51 |

| | | |
|--------|------------------------------------|----|
| 3.4.2 | FK_ModifyPrivilege..... | 51 |
| 3.4.3 | FK_GetUserName..... | 51 |
| 3.4.4 | FK_SetUserName..... | 51 |
| 3.4.5 | FK_GetNewsMessage..... | 52 |
| 3.4.6 | FK_SetNewsMessage..... | 52 |
| 3.4.7 | FK_GetUserNewsID..... | 52 |
| 3.4.8 | FK_SetUserNewsID..... | 52 |
| 3.5 | 机器管理 | 52 |
| 3.5.1 | FK_EnableDevice..... | 52 |
| 3.5.2 | FK_PowerOnAllDevice..... | 52 |
| 3.5.3 | FK_PowerOffDevice..... | 53 |
| 3.5.4 | FK_GetDeviceTime..... | 53 |
| 3.5.5 | FK_SetDeviceTime..... | 53 |
| 3.5.6 | FK_GetDeviceStatus..... | 53 |
| 3.5.7 | FK_GetDeviceInfo..... | 53 |
| 3.5.8 | FK_SetDeviceInfo..... | 53 |
| 3.5.9 | FK_GetProductData..... | 54 |
| 3.5.10 | FK_GetProductDataWithString..... | 54 |
| 3.5.11 | FK_GetDeviceVersion..... | 54 |
| 3.5.12 | FK_GetDeviceTime_1..... | 54 |
| 3.5.13 | FK_SetDeviceTime_1..... | 54 |
| 3.6 | 响铃管理 | 55 |
| 3.6.1 | FK_GetBellTime..... | 55 |
| 3.6.2 | FK_GetBellTimeWithString..... | 55 |
| 3.6.3 | FK_SetBellTime..... | 55 |
| 3.6.4 | FK_SetBellTimeWithString..... | 55 |
| 3.7 | 门铃管理 | 55 |
| 3.7.1 | FK_GetDoorStatus..... | 55 |
| 3.7.2 | FK_SetDoorStatus..... | 56 |
| 3.7.3 | FK_GetPassTime..... | 56 |
| 3.7.4 | FK_GetPassTimeWithString..... | 56 |
| 3.7.5 | FK_SetPassTime..... | 56 |
| 3.7.6 | FK_SetPassTimeWithString..... | 56 |
| 3.7.7 | FK_GetUserPassTime..... | 56 |
| 3.7.8 | FK_GetUserPassTimeWithString..... | 57 |
| 3.7.9 | FK_SetUserPassTime..... | 57 |
| 3.7.10 | FK_SetUserPassTimeWithString..... | 57 |
| 3.7.11 | FK_GetGroupPassTime..... | 57 |
| 3.7.12 | FK_GetGroupPassTimeWithString..... | 57 |
| 3.7.13 | FK_SetGroupPassTime..... | 58 |
| 3.7.14 | FK_SetGroupPassTimeWithString..... | 58 |

| | | |
|--------|---------------------------------|----|
| 3.7.15 | FK_GetGroupMatch..... | 58 |
| 3.7.16 | FK_GetGroupMatchWithString..... | 58 |
| 3.7.17 | FK_SetGroupMatch..... | 58 |
| 3.7.18 | FK_SetGroupMatchWithString..... | 58 |
| 3.8 | 夏天时间制管理..... | 60 |
| 3.8.1 | FK_GetAdjustInfo..... | 60 |
| 3.8.2 | FK_SetAdjustInfo..... | 60 |
| 3.9 | 机器信息管理 | 61 |
| 3.9.1 | FK_GetServerNetInfo..... | 61 |
| 3.9.2 | FK_SetServerNetInfo..... | 61 |
| 3.9.3 | FK_SetUSBModel..... | 61 |
| 4 | 备注 | 62 |
| 4.1 | 结构体 | 62 |
| 4.1.1 | BELLINFO 结构体..... | 62 |
| 4.1.2 | PASSCTRLTIME 结构体..... | 62 |
| 4.1.3 | USERPASSINFO 结构体 | 62 |
| 4.1.4 | GROUPPASSINFO 结构体..... | 63 |
| 4.1.5 | GROUPMATCHINFO 结构体..... | 63 |
| 4.1.6 | ADJUSTINFO 结构体 | 63 |
| 4.1.7 | REALTIMEINFO 结构体..... | 63 |
| 4.2 | 错误代码表 | 64 |

1 概要

该使用说明书解释了关于本公司提供的 FK6xx 系列指纹考勤机应用程序的开发方法. 关于 OEM 产品 FKAttend 的说明.

以 FKAttend. ocx, FKAttend. dll, FKViaDev. dll 组成

FKAttend. ocx 是上端软件与机器的接口 OCX.

FKAttend. dll 是上端软件与机器连接部分的 dll, 功能上与 FKAttend. Ocx 一样.

FKViaDev. dll 是与机器通讯的通讯接口 dll.

接口以 7 个大部分组成.

- ① 机器连接与断开 - 该部分说明了与机器连接和断开的部分.
- ② 登记数据管理 - 该部分说明了怎么进行读与删除用户的登记数据.
- ③ 记录数据管理 - 说明怎么读管理记录和进出记录.
- ④ 用户信息管理 - 设置用户的名字, 通知等信息.
- ⑤ 机器管理 - 设置机器的时间, 状态, 信息.
- ⑥ 响铃管理 - 设置响铃时间.
- ⑦ 门铃管理 - 设置门操作.

2 FKAttend. OCX 的接口

2.1 机器连接与断开

2.1.1 ConnectComm

| | | |
|----|---|--|
| 形式 | long ConnectComm(long nMachineNumber, long nComPort, long nBaudRate, char * pstrTelNumber, long nWaitDialTime, long nLicense) | |
| 功能 | 通过 RS-232/485 线为了机器连接打开 COM 接口。 | |
| 参数 | nMachineNumber | 要连接的机器号码。 |
| | nComPort | COM 接口号码 |
| | nBaudRate | COM 通讯波特率 |
| | pstrTelNumber | 电话号码 |
| | nWaitDialTime | 电话连接待机时间 (单位 ms) |
| | nLicense | 连接许可号码 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | “pstrTelNumber”与“nWaitDialTime”是通过 Modem 连接的时候用的函数, 若不用 Modem 的情况下得输入 0. |
| | 2 | “nLicense”是针对每个代理, 我们公司提供的许可号码. 没有这个许可号码就连接不了机器. |

2.1.2 ConnectNet

| | | |
|----|--|---|
| 形式 | long ConnectNet(long nMachineNumber, char * strIpAddress, long nPort, long nTimeOut, long nProtocolType, long nNetPassword, long nLicense) | |
| 功能 | 通过网线连接机器的时候打开网接口。 | |
| 参数 | nMachineNumber | 要连接的机器号码。 |
| | strIpAddress | 要连接的机器 TCP/IP 地址。 |
| | nPort | 网接口号。 |
| | nTimeOut | 连接待机时间(单位 ms) |
| | nProtocolType | 通讯协议种类 |
| | nNetPassword | 网通讯密码 |
| | nLicense | 连接许可号码。 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 机器没有跟网连接的情况下按照“nTimeOut”时间待机后返回错误代码。 |
| | 2 | “nProtocolType”指定网络通讯的通讯协议种类。 0 : PROTOCOL_TCPIP - TCP/IP 通讯 1 : PROTOCOL_UDP - UDP 通讯。 |
| | 3 | “nLicense”是 《2.1.1 ConnectComm》和一样。 |

2.1.3 ConnectUSB

| | | |
|----|---|------------------------------------|
| 形式 | long ConnectUSB(long nMachineNumber, long nLicense) | |
| 功能 | 为了通过 USB 线连接电脑进行通讯, 打开 USB 接口。 | |
| 参数 | nMachineNumber | 要连接的机器号码。 |
| | nLicense | 连接许可号码。 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | “nLicense”是《2.1.1 ConnectComm》和一样。 |

2.1.4 DisConnect

| | | |
|----|-----------------------|---|
| 形式 | void DisConnect(void) | |
| 功能 | 断开与机器的连接。 | |
| 参数 | | |
| 返回 | 没有 | |
| 其他 | 1 | 关上与 ConnectComm 或 ConnectNet 连接的机器既关上被开着的对应的接口。 |

2.1.5 ConnectGetIP

| | | |
|----|-------------------------------------|------------------------------------|
| 形式 | long ConnectGetIP(BSTR *strComName) | |
| 功能 | 以名称生成 IP 地址的函数 | |
| 参数 | strComName | 待寻找 IP 地址机器的名字 |
| 其他 | 1 | “nLicense”是《2.1.1 ConnectComm》和一样。 |

2.2 登记数据管理

2.2.1 GetEnrollData

| | | |
|----|---|--|
| 形式 | long GetEnrollData(long anEnrollNumber, long anBackupNumber, long *apnMachinePrivilege, long *apnEnrollData, long *apnPassWord) | |
| 功能 | 获取用户的操作授权和登记资料。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| | apnMachinePrivilege | 用户的操作权限的参数指示者 |
| | apnEnrollData | 指纹数据的参数指示者 |
| | apnPassWord | 密码或卡号数据的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 执行成功按照“anBackupNumber”对应的登记数据返回到“apnEnrollData”或“apnPassWord”。 |
| | 2 | 返回到“apnMachinePrivilege”的操作权限含义请参考《2.4.2 ModifyPrivilege》。 |
| | 3 | 每个用户能登记 3 枚指纹和 1 个密码或是卡号, 该登记资料反映在“anBackupNumber”上。 |

| | | |
|--|--|---|
| | | <p>anBackupNumber”的含义如下：</p> <p>0：BACKUP_FP_0 - 被登记第一个指纹区</p> <p>... ..</p> <p>9：BACKUP_FP_9 - 被登记第九个指纹区</p> <p>10：BACKUP_PSW - 被登记密码</p> <p>11：BACKUP_CARD - 被登记卡号</p> |
|--|--|---|

2.2.2 GetEnrollDataWithString

| | | |
|----|---|--|
| 形式 | long GetEnrollDataWithString(long anEnrollNumber, long anBackupNumber, long *apnMachinePrivilege, char * apstrEnrollData) | |
| 功能 | 与 GetEnrollData 指令类似, 登记资料以文字列形式获取. | |
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| | apnMachinePrivilege | 用户的操作权限的参数指示者 |
| | apstrEnrollData | 登记资料的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 跟“anBackupNumber”无关, 则登记资料返回到 “apstrEnrollData”, |
| | 2 | 关于其他参数的说明请参考《2.2.1 GetEnrollData》. |

2.2.3 PutEnrollData

| | | |
|----|---|--|
| 形式 | long PutEnrollData(long anEnrollNumber, long anBackupNumber, long anMachinePrivilege, long *apnEnrollData, long anPassword) | |
| 功能 | 将要登记的用户的操作权限和登记资料传送到机器. | |
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| | anMachinePrivilege | 用户的 操作权限 |
| | apnEnrollData | 指纹数据的参数指示者 |
| | anPassword | 密码或者卡号数据. |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | anBackupNumber”请参考《2.2.1 GetEnrollData》. |
| | 2 | anMachinePrivilege” 请参考《2.4.2 ModifyPrivilege》. |
| | 3 | 按照“anBackupNumber”传送. “apnEnrollData”或者 “anPassword”数据. |
| | 4 | 执行完指令后再执行 SaveEnrollData 指令才能机器上保存传送的数据, SaveEnrollData 指令请参考《2.2.5 SaveEnrollData》. |

2.2.4 PutEnrollDataWithString

| | | |
|----|---|--|
| 形式 | long PutEnrollDataWithString(long anEnrollNumber, long anBackupNumber, long anMachinePrivilege, BSTR apstrEnrollData) | |
| 功能 | 与 PutEnrollData 指令相似, 将登记资料以文字列形式记录. | |

| | | |
|----|---|--|
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| | anMachinePrivilege | 用户的操作权限 |
| | apstrEnrollData | 登记资料的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 不管“anBackupNumber”直接将登记资料登记“apstrEnrollData”上。 |
| | 2 | 关于其他参数的说明请参考《2.2.3 PutEnrollData》。 |

2.2.5 SaveEnrollData

| | | |
|----|---|---|
| 形式 | long SaveEnrollData(void) | |
| 功能 | 用 PutEnrollData 或 PutEnrollDataWithString 指令被传送的登记资料登记到机器上。 | |
| 参数 | | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 利用该指令之前先用 PutEnrollData 或 PutEnrollDataWithString 指令将要登记的数据传送机器上。 |

2.2.6 DeleteEnrollData

| | | |
|----|---|----------------------|
| 形式 | long DeleteEnrollData(long anEnrollNumber, long anBackupNumber) | |
| 功能 | 从删除指定的登记资料。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 如机器上没有指定的数据的话该指令被失败。 |

2.2.7 USBReadAllEnrollDataFromFile

| | | |
|----|--|----------------------------|
| 形式 | long USBReadAllEnrollDataFromFile(char *apstrFilePath) | |
| 功能 | 机器的 USB 存储器理的登记资料文件中将登记资料读到 PC 的内存进行分析。 | |
| 参数 | apstrFilePath | 文件名 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 文件结构不正确的话指令执行被失败。 |
| | 2 | 在机器上 USB 存储器利用方法参考机器使用说明书。 |

2.2.8 USBReadAllEnrollDataCount

| | | |
|----|---|---|
| 形式 | long USBReadAllEnrollDataCount(long *apnValue) | |
| 功能 | 返回通过 USBReadAllEnrollDataFromFile 指令读到 PC 内存的登记资料的个数。 | |
| 参数 | apnValue | 登记资料的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 用该指令之前, 首先用 USBReadAllEnrollDataFromFile 指令读资料。 |

2.2.9 USBGetOneEnrollData

| | | |
|----|---|--|
| 形式 | long USBGetOneEnrollData(long *apnEnrollNumber, long *apnBackupNumber, long *apnMachinePrivilege, long *apnEnrollData, long *apnPassWord, long *apnEnableFlag, BSTR *apnEnrollName) | |
| 功能 | 获取通过 USBReadAllEnrollDataFromFile 指令读的登记资料. | |
| 参数 | apnEnrollNumber | 登记号码参数指示者 |
| | apnBackupNumber | 登记资料 种类号码参数指示者 |
| | apnMachinePrivilege | 用户的操作权限的参数指示者 |
| | apnEnrollData | 指纹数据的参数指示者 |
| | apnPassWord | 密码或卡号数据的参数指示者 |
| | apnEnableFlag | 用户的机器许可的参数指示者 |
| | apnEnrollName | 用户的名字的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 该指令与 GetEnrollData 指令相似. 不一样的部分是不直接连接, 只是用 USB 存储器. 关于 GetEnrollData 指令的说明参考《2.2.1 GetEnrollData》. |
| | 2 | 获取完数据后返回 RUNERR_LOG_END. |
| | 3 | 若通过 USBReadAllEnrollDataFromFile 指令电脑上没有读过来的登记资料数据的话指令被失败. |
| | 4 | "apnEnableFlag"的解释请参考《2.4.1 EnableUser》. |

2.2.10 USBGetOneEnrollDataWithString

| | | |
|----|--|--|
| 形式 | long USBGetOneEnrollDataWithString(long *apnEnrollNumber, long *apnBackupNumber, long *apnMachinePrivilege, BSTR* apstrEnrollData, long *apnEnableFlag, BSTR *apnEnrollName) | |
| 功能 | 与 USBGetOneEnrollData 指令相似, 将登记资料以文字列形式获取. | |
| 参数 | apnEnrollNumber | 登记号码参数指示者 |
| | apnBackupNumber | 登记资料 种类号码参数指示者 |
| | apnMachinePrivilege | 用户的操作权限的参数指示者 |
| | apstrEnrollData | 登记资料参数指示者 |
| | apnEnableFlag | 用户的机器许可的参数指示者 |
| | apnEnrollName | 用户的名字的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 该指令与 GetEnrollDataWithString 指令相似, 不一样的部分是不直接连接, 只是用 USB 存储器. 关于 GetEnrollDataWithString 指令请参考《2.2.2 GetEnrollDataWithString》. |
| | 2 | 其他说明请参考《2.2.9 USBGetOneEnrollData》. |

2.2.11 USBSetOneEnrollData

| | | |
|----|---|--|
| 形式 | long USBSetOneEnrollData(long anEnrollNumber, long anBackupNumber, long anMachinePrivilege, long *apnEnrollData, long anPassWord, long anEnableFlag, LPC TSTR anEnrollName) | |
| 功能 | 为了用户的操作权限和登记资料转换成 USB 存储器用文件, 在 PC 内存上进行形式化. | |

| | | |
|----|---|--|
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| | anMachinePrivilege | 用户的操作权限 |
| | apnEnrollData | 指纹数据的参数指示者 |
| | anPassWord | 密码或卡号数据的参数指示者 |
| | anEnableFlag | 用户的机器许可的参数指示者 |
| | anEnrollName | 用户的名字的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 该指令与 PutEnrollData 指令相似 不一样的部分是不直接连接, 只是用 USB 存储器. 关于 PutEnrollData 指令的说明请参考《2.2.3 PutEnrollData》。 |
| | 2 | “anEnableFlag”的解释请参考《2.4.1 EnableUser》。 |

2.2.12 USBSetOneEnrollDataWithString

| | | |
|----|--|---|
| 形式 | long USBSetOneEnrollDataWithString(long anEnrollNumber, long anBackupNumber, long anMachinePrivilege, BSTR apstrEnrollData, long anEnableFlag, LPCTSTR anEnrollName) | |
| 功能 | 与 USBSetOneEnrollData 指令一样, 将登记资料以文字列形式登记。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| | anMachinePrivilege | 用户的操作权限 |
| | apstrEnrollData | 登记资料的参数指示者 |
| | anEnableFlag | 用户的机器许可的参数指示者 |
| | anEnrollName | 用户的名字的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 该指令与 PutEnrollDataWithString 指令相似, 不一样的部分是不直接连接, 只是用 USB 存储器. 关于 PutEnrollDataWithString 的指令请参考《2.2.4 PutEnrollDataWithString》。 |
| | 2 | 其他说明请参考《2.2.11 USBSetOneEnrollData》。 |

2.2.13 USBWriteAllEnrollDataToFile

| | | |
|----|---|---|
| 形式 | long USBWriteAllEnrollDataToFile(char *apstrFilePath) | |
| 功能 | 通过 USBSetOneEnrollData 或 USBSetOneEnrollDataWithString 指令在 PC 的内存上制作形式性的登记资料文件。 | |
| 参数 | apstrFilePath | 文件名称 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 用该指令之前, 先得有用 USBSetOneEnrollData 或 USBSetOneEnrollDataWithString 指令做好的形式性的数据。 |
| | 2 | 机器上的 USB 存储器的使用方法请参考使用说明书。 |

2.2.14 ReadAllUserID

| | |
|----|--------------------------|
| 形式 | long ReadAllUserID(void) |
|----|--------------------------|

| | | |
|----|--|--|
| 功能 | 所有的机器上的用户信息数据读到 PC 的内存上. | |
| 参数 | | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 被读的用户信息通过 GetAllUserID 指令获取. 关于 GetAllUserID 的指令请参考《2.2.15 GetAllUserID》. |
| | 2 | 没有被登记的用户, 该指令被失败. |

2.2.15 GetAllUserID

| | | |
|----|---|---|
| 形式 | long GetAllUserID(long *apnEnrollNumber, long *apnBackupNumber, long *apnMachinePrivilege, long *apnEnableFlag) | |
| 功能 | 将通过 ReadAllUserID 读过来的用户信息, 一个一个的获取. | |
| 参数 | apnEnrollNumber | 登记号码的参数指示者 |
| | apnBackupNumber | 登记资料种类号码的参数指示者 |
| | apnMachinePrivilege | 用户的操作权限的参数指示者 |
| | apnEnableFlag | 用户的机器许可的参数指示者的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 没有通过 ReadAllUserID 读的用户信息, 该指令内失败. |
| | 2 | 获取完数据后返回 RUNERR_LOG_END. |
| | 3 | 返回到“apnMachinePrivilege”的操作权限的说明请参考《2.4.2 ModifyPrivilege》. |
| | 4 | “apnEnableFlag”的说明请参考《2.4.1 EnableUser》. |

2.2.16 EmptyEnrollData

| | | |
|----|--|-----------------------------------|
| 形式 | long EmptyEnrollData(void) | |
| 功能 | 删除被登记的所有登记资料. | |
| 参数 | | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 因为全部删除所有的登记资料, 执行该指令之前请备份重要的登记资料. |

2.2.17 ClearKeeperData

| | | |
|----|--|--|
| 形式 | long ClearKeeperData(void) | |
| 功能 | 从机器上删除所有登记资料和记录数据. (机器出初始化功能.) | |
| 参数 | | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 因为全部删除所有的登记资料和记录数据, 执行该指令之前请备份重要的登记资料. |

2.2.18 BenumbAllManager

| | | |
|----|-----------------------------|--|
| 形式 | long BenumbAllManager(void) | |
|----|-----------------------------|--|

| | | |
|----|---|--|
| 功能 | 登陆数据中全部删除管理权限信息，将用户甚至为一般用户。 | |
| 参数 | | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | |

2.2.19 GetVerifyMode

| | | |
|---------------|--|-----------|
| Type | long GetVerifyMode(long anEnrollNumber, long *apnVerifyMode) | |
| Functionality | 分别用于获取用户验证模式的函数。 | |
| Parameter | anEnrollNumber | 登记号码参数 |
| | apnVerifyMode | 验证模式参数指示者 |
| Return | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| Others | 1 | |

2.2.20 SetVerifyMode

| | | |
|---------------|--|-----------|
| Type | long SetVerifyMode(long anEnrollNumber, long anVerifyMode) | |
| Functionality | 分别用于设置用户验证模式的函数。 | |
| Parameter | anEnrollNumber | 登记号码参数 |
| | anVerifyMode | 验证模式参数指示者 |
| Return | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| Others | 1 | |

2.2.21 USBGetOneEnrollData_1

| | | |
|---------------|--|---|
| Type | long USBGetOneEnrollData_1(long *apnEnrollNumber, long *apnBackupNumber, long *apnVerifyMode, long *apnMachinePrivilege, long *apnEnrollData, long *apnPassWord, long *apnEnableFlag, BSTR *apnEnrollName) | |
| Functionality | 获取通过 USBReadAllEnrollDataFromFile 指令读的登记资料。 | |
| Parameter | apnEnrollNumber | 登记号码参数指示者 |
| | apnBackupNumber | 登记资料 种类号码参数指示者 |
| | apnVerifyMode | |
| | apnMachinePrivilege | 用户的操作权限的参数指示者 |
| | apnEnrollData | 指纹数据的参数指示者 |
| | apnPassWord | 密码或卡号数据的参数指示者 |
| | apnEnableFlag | 用户的机器许可的参数指示者 |
| | apnEnrollName | 用户的名字的参数指示者 |
| Return | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| Others | 1 | 该指令与 GetEnrollData 指令相似。不一样的部分是不直接连接，只是用 USB 存储器。关于 GetEnrollData 指令的说明参考《2.2.1 GetEnrollData》。 |
| | 2 | 获取完数据后返回 RUNERR_LOG_END。 |
| | 3 | 若通过 USBReadAllEnrollDataFromFile 指令电脑上没有读过来的登记资料数据的话指令被失败。 |

| | | |
|--|---|--|
| | 4 | 'apnEnableFlag'的解释请参考《2.4.1 EnableUser》。 |
|--|---|--|

2.2.22 USBGetOneEnrollDataWithString_1

| | | |
|---------------|---|---|
| Type | long USBGetOneEnrollDataWithString_1(long *apnEnrollNumber, long *apnBackupNumber, long *apnVerifyMode, long *apnMachinePrivilege, BSTR* apstrEnrollData, long *apnEnableFlag, BSTR *apnEnrollName) | |
| Functionality | 与 USBGetOneEnrollData 指令相似，将登记资料以文字列形式获取。 | |
| Parameter | apnEnrollNumber | 登记号码参数指示者 |
| | apnBackupNumber | 登记资料 种类号码参数指示者 |
| | apnVerifyMode | |
| | apnMachinePrivilege | 用户的操作权限的参数指示者 |
| | apstrEnrollData | 指纹数据的参数指示者 |
| | apnEnableFlag | 用户的机器许可的参数指示者 |
| | apnEnrollName | 用户的名字的参数指示者 |
| Return | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| Others | 1 | 该指令与 PutEnrollData 指令相似 不一样的部分是不直接连接，只是用 USB 存储器. 关于 PutEnrollData 指令的说明请参考《2.2.3 PutEnrollData》。 |
| | 2 | 'anEnableFlag'的解释请参考《2.4.1 EnableUser》。 |

2.2.23 USBSetOneEnrollData_1

| | | |
|---------------|---|---|
| Type | long USBSetOneEnrollData_1(long anEnrollNumber, long anBackupNumber, long anVerifyMode, long anMachinePrivilege, long *apnEnrollData, long anPassWord, long anEnableFlag, LPCTSTR anEnrollName) | |
| Functionality | 为了用户的操作权限和登记资料转换成 USB 存储器用文件, 在 PC 内存上进行形式化。 | |
| Parameter | anEnrollNumber | 登记号码参数指示者 |
| | anBackupNumber | 登记资料 种类号码参数指示者 |
| | anVerifyMode | |
| | anMachinePrivilege | 用户的操作权限的参数指示者 |
| | apnEnrollData | 指纹数据的参数指示者 |
| | anPassWord | 密码或卡号数据的参数指示者 |
| | anEnableFlag | 用户的机器许可的参数指示者 |
| | anEnrollName | 用户的名字的参数指示者 |
| Return | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| Others | 1 | 该指令与 PutEnrollData 指令相似 不一样的部分是不直接连接，只是用 USB 存储器. 关于 PutEnrollData 指令的说明请参考《2.2.3 PutEnrollData》。 |
| | 2 | 'anEnableFlag'的解释请参考《2.4.1 EnableUser》。 |

2.2.24 USBSetOneEnrollDataWithString_1

| | |
|------|---|
| Type | long USBSetOneEnrollDataWithString_1(long anEnrollNumber, long anBackupNumber, long anVerifyMode, long anMachinePrivilege, BSTR apstrEnrollData, long anEnableFlag, LPCTSTR anEnrollName) |
|------|---|

| | | |
|---------------|--|---|
| Functionality | 与 USBSetOneEnrollData 指令一样, 将登记资料以文字列形式登记. | |
| Parameter | anEnrollNumber | 登记号码参数指示者 |
| | anBackupNumber | 登记资料 种类号码参数指示者 |
| | anVerifyMode | |
| | anMachinePrivilege | 用户的操作权限的参数指示者 |
| | apstrEnrollData | 登记数据的参数指示者 |
| | anEnableFlag | 用户的机器许可的参数指示者 |
| | anEnrollName | 用户的名字的参数指示者 |
| Return | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| Others | 1 | 该指令与 PutEnrollDataWithString 指令相似, 不一样的部分是不直接连接, 只是用 USB 存储器. 关于 PutEnrollDataWithString 的指令请参考《2.2.4 PutEnrollDataWithString》. |
| | 2 | 其他说明请参考《2.2.11 USBSetOneEnrollData》. |

2.2.25 USBReadAllEnrollDataFromFile_Color

| | | |
|----|---|----------------------------|
| 形式 | Long USBReadAllEnrollDataFromFile_Color (char *apstrFilePath) | |
| 功能 | 机器的 USB 存储器理的登记资料文件中将登记资料读到 PC 的内存进行分析. | |
| 参数 | apstrFilePath | 文件名 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 文件结构不正确的话指令执行被失败. |
| | 2 | 在机器上 USB 存储器利用方法参考机器使用说明书. |

2.2.26 USBWriteAllEnrollDataToFile_Color

| | | |
|----|---|---|
| 形式 | long USBWriteAllEnrollDataToFile_Color (char *apstrFilePath, long anNewsKind) | |
| 功能 | 通过 USBSetOneEnrollData 或 USBSetOneEnrollDataWithString 指令在 PC 的内存上制作形式性的登记资料文件. | |
| 参数 | apstrFilePath | 文件名称 |
| | anNewsKind | 消息设置类型 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 用该指令之前, 先得有用 USBSetOneEnrollData 或 USBSetOneEnrollDataWithString 指令做好的形式性的数据. |
| | 2 | 机器上的 USB 存储器的使用方法请参考使用说明书. |

2.2.27 USBGetOneEnrollData_Color

| | | |
|----|---|----------------|
| 形式 | long USBGetOneEnrollData_Color (long *apnEnrollNumber, long *apnBackupNumber, long *apnMachinePrivilege, long *apnEnrollData, long *apnPassWord, long *apnEnableFlag, BSTR *apnEnrollName, long anNewsKind) | |
| 功能 | 获取通过 USBReadAllEnrollDataFromFile 指令读的登记资料. | |
| 参数 | apnEnrollNumber | 登记号码参数指示者 |
| | apnBackupNumber | 登记资料 种类号码参数指示者 |
| | apnMachinePrivilege | 用户的操作权限的参数指示者 |

| | | |
|----|---|--|
| | apnEnrollData | 指纹数据的参数指示者 |
| | apnPassWord | 密码或卡号数据的参数指示者 |
| | apnEnableFlag | 用户的机器许可的参数指示者 |
| | apnEnrollName | 用户的名字的参数指示者 |
| | anNewsKind | 消息设置类型 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 该指令与 GetEnrollData 指令相似。不一样的部分是不直接连接, 只是用 USB 存储器。关于 GetEnrollData 指令的说明参考《2.2.1 GetEnrollData》。 |
| | 2 | 获取完数据后返回 RUNERR_LOG_END。 |
| | 3 | 若通过 USBReadAllEnrollDataFromFile 指令电脑上没有读过来的登记资料数据的话指令被失败。 |
| | 4 | “apnEnableFlag”的解释请参考《2.4.1 EnableUser》。 |

2.2.28 USBGetOneEnrollDataWithString_Color

| | | |
|----|--|---|
| 形式 | long USBGetOneEnrollDataWithString_Color (long *apnEnrollNumber, long *apnBackupNumber, long *apnMachinePrivilege, BSTR* apstrEnrollData, long *apnEnableFlag, BSTR *apnEnrollName, long anNewsKind) | |
| 功能 | 与 USBGetOneEnrollData 指令相似, 将登记资料以文字列形式获取。 | |
| 参数 | apnEnrollNumber | 登记号码参数指示者 |
| | apnBackupNumber | 登记资料 种类号码参数指示者 |
| | apnMachinePrivilege | 用户的操作权限的参数指示者 |
| | apstrEnrollData | 登记资料参数指示者 |
| | apnEnableFlag | 用户的机器许可的参数指示者 |
| | apnEnrollName | 用户的名字的参数指示者 |
| | anNewsKind | 消息设置类型 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 该指令与 GetEnrollDataWithString 指令相似, 不一样的部分是不直接连接, 只是用 USB 存储器。关于 GetEnrollDataWithString 指令请参考《2.2.2 GetEnrollDataWithString》。 |
| | 2 | 其他说明请参考《2.2.9 USBGetOneEnrollData》。 |

2.2.29 USBSetOneEnrollData_Color

| | | |
|----|--|---------------|
| 形式 | long USBSetOneEnrollData_Color (long anEnrollNumber, long anBackupNumber, long anMachinePrivilege, long *apnEnrollData, long anPassWord, long anEnableFlag, LPCTSTR anEnrollName, long anNewsKind) | |
| 功能 | 为了用户的操作权限和登记资料转换成 USB 存储器用文件, 在 PC 内存上进行形式化。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| | anMachinePrivilege | 用户的操作权限 |
| | apnEnrollData | 指纹数据的参数指示者 |
| | anPassWord | 密码或卡号数据的参数指示者 |
| | anEnableFlag | 用户的机器许可的参数指示者 |

| | | | |
|----|---|--|-------------|
| | anEnrollName | | 用户的名字的参数指示者 |
| | anNewsKind | | 消息设置类型 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | | |
| 其他 | 1 | 该指令与 PutEnrollData 指令相似 不一样的部分是不直接连接, 只是用 USB 存储器. 关于 PutEnrollData 指令的说明请参考《2.2.3 PutEnrollData》。 | |
| | 2 | anEnableFlag”的解释请参考《2.4.1 EnableUser》。 | |

2.2.30 USBSetOneEnrollDataWithString_Color

| | | |
|----|--|---|
| 形式 | long USBSetOneEnrollDataWithString_Color(long anEnrollNumber, long anBackupNumber, long anMachinePrivilege, BSTR apstrEnrollData, long anEnableFlag, LPC TSTR anEnrollName, long anNewsKind) | |
| 功能 | 与 USBSetOneEnrollData 指令一样, 将登记资料以文字列形式登记. | |
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| | anMachinePrivilege | 用户的操作权限 |
| | apstrEnrollData | 登记资料的参数指示者 |
| | anEnableFlag | 用户的机器许可的参数指示者 |
| | anEnrollName | 用户的名字的参数指示者 |
| | anNewsKind | 消息设置类型 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 该指令与 PutEnrollDataWithString 指令相似, 不一样的部分是不直接连接, 只是用 USB 存储器. 关于 PutEnrollDataWithString 的指令请参考《2.2.4 PutEnrollDataWithString》. |
| | 2 | 其他说明请参考《2.2.11 USBSetOneEnrollData》. |

2.3 记录数据管理

2.3.1 LoadSuperLogData

| | | |
|----|---|---|
| 形式 | long LoadSuperLogData(long anReadMark) | |
| 功能 | 从机器将管理记录读到 PC 的内存上进行分析。 | |
| 参数 | anReadMark | 读标志 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 读过来的数据通过 GetSuperLogData 可以获取。请参考《2.3.3 GetSuperLogData》。 |
| | 2 | anReadMark = 1 的话就读新加的记录数据。 anReadMark = 0 的情况下读所有的记录数据。 |

2.3.2 USBLoadSuperLogDataFromFile

| | | | |
|----|---|--|--|
| 形式 | long USBLoadSuperLogDataFromFile(char *apstrFilePath) | | |
| 功能 | 机器的 USB 存储器理的管理记录数据文件中将记录数据读到 PC 的内存进行分析。 | | |

| | | |
|----|---|---|
| 参数 | apstrFilePath | 文件名称 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 与 LoadSuperLogData 相似, 与 PC 断开的情况下用该指令获取管理记录数据。 |
| | 2 | 文件的结构不正确的情况下指令的执行被失败。 |
| | 3 | 机器的 USB 使用方法参考使用说明书。 |

2.3.3 GetSuperLogData

| | | |
|----|---|---|
| 形式 | long GetSuperLogData(long *apnSEnrollNumber, long *apnGEnrollNumber, long *apnManipulation, long *apnBackupNumber, DATE *apnDateTime) | |
| 功能 | 通过 LoadSuperLogData 或 USBLoadSuperLogDataFromFile 的指令读到 PC 内存的管理记录一个一个的获取。 | |
| 参数 | apnSEnrollNumber | 管理者的登记号码的参数指示者 |
| | apnGEnrollNumber | 被管理者的登记号码的参数指示者 |
| | apnManipulation | 管理对象的识别号码的参数指示者 |
| | apnBackupNumber | 被管理者的登记资料种类号码的参数指示者 |
| | apnDateTime | 管理记录日期, 时间的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 获取完数据后返回 RUNERR_LOG_END. |
| | 2 | 先执行 LoadSuperLogData 或 USBLoadSuperLogDataFromFile 指令, 不然该指令被失败。 |
| | 3 | "apnManipulation"如下 : 3 : LOG_ENROLL_USER - 一般用户登记 4 : LOG_ENROLL_MANAGER - 管理者登记 5 : LOG_ENROLL_DELPASS - 删除指纹数据 6 : LOG_ENROLL_DELPASS - 删除通过密码 7 : LOG_ENROLL_DELCARD - 删除卡数据 8 : LOG_LOG_ALLDEL - 删除所有的记录数据 9 : LOG_SETUP_SYS - 修改机器信息 10 : LOG_SETUP_TIME - 修改机器时间 11 : LOG_SETUP_LOG - 修改记录限制值 12 : LOG_SETUP_COMM - 修改通讯方式 13 : LOG_PASSTIME - 通过时间设置 14 : LOG_SETUP_DOOR - 门操作信息设置 |

2.3.4 EmptySuperLogData

| | | |
|----|---|-------------------------------------|
| 形式 | long EmptySuperLogData(void) | |
| 功能 | 从机器上删除所有的管理记录数据。 | |
| 参数 | | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 因为全部删除所有的管理记录数据, 执行该指令之前请备份重要的登记资料。 |

2.3.5 LoadGeneralLogData

| | | |
|----|---|--|
| 形式 | long LoadGeneralLogData(long anReadMark) | |
| 功能 | 从机器将进出记录数据读到 PC 的内存上进行分析。 | |
| 参数 | anReadMark | 读标志 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 读过来的数据通过 GetGeneralLogData 可以获取。 《2.3.7 GetGeneralLogData》。 |
| | 2 | anReadMark = 1 的情况下读取新的记录数据 anReadMark = 0 的情况下读取全部疾苦数据 |

2.3.6 USBLoadGeneralLogDataFromFile

| | | |
|----|--|--|
| 形式 | long USBLoadGeneralLogDataFromFile(BSTR apstrFilePath) | |
| 功能 | 机器的 USB 存储器里的进出记录数文件中将记录数据读到 PC 的内存进行分析。 | |
| 参数 | apstrFilePath | 文件名称 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 与 LoadGeneralLogData 相似, 跟 PC 断开的情况下为了获取进出记录数据使用该指令。 |
| | 2 | 文件结构不正确的话指令的执行失败。 |
| | 3 | 机器的 USB 使用方法参考使用说明书。 |

2.3.7 GetGeneralLogData

| | | |
|----|---|--------------------------|
| 形式 | long GetGeneralLogData(long *apnEnrollNumber, long *apnVerifyMode, long *apnInOutMode, DATE *apnDateTime) | |
| 功能 | 通过 LoadGeneralLogData 或 USBLoadGeneralLogDataFromFile 指令读到 PC 内存的进出记录数据一个一个的获取。 | |
| 参数 | apnEnrollNumber | 进出用户登记号码的参数指示者 |
| | apnVerifyMode | 确认方式的参数指示者 |
| | apnInOutMode | 进出方式的参数指示者 |
| | apnDateTime | 进出记录日期, 时间的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 获取完数据后返回 RUNERR_LOG_END. |

| | | |
|--|---|---|
| | 2 | <p>apnVerifyMode"如下 :</p> <p>1 : LOG_FPVERIFY - 指纹确认</p> <p>2 : LOG_PASSVERIFY - 密码确认</p> <p>3 : LOG_CARDVERIFY - 卡确认</p> <p>4 : LOG_FPPASS_VERIFY - 指纹+密码确认</p> <p>5 : LOG_FPCARD_VERIFY - 指纹+卡确认</p> <p>6 : LOG_PASSFP_VERIFY - 密码+指纹确认</p> <p>7 : LOG_CARDFP_VERIFY - 卡+指纹确认</p> <p>有控锁功能的版本里利用.(参考《2.7 门铃管理》)</p> <p>10 : LOG_OPEN_DOOR - 确认后发送开门信号</p> <p>11 : LOG_CLOSE_DOOR - 确认后发送关门信号</p> <p>12 : LOG_OPEN_HAND - 通过开门按钮发送开门信号</p> <p>13 : LOG_OPEN_THREAT -防胁迫指纹确认时发送开门信号</p> <p>14 : LOG_PROG_OPEN - 从操作机器发送开门信号</p> <p>15 : LOG_PROG_CLOSE -从操作机器发送关门信号</p> <p>16 : LOG_OPEN_IREGAL - 发送非法性开门信号</p> <p>17 : LOG_CLOSE_IREGAL -发送非法性关门信号</p> <p>18 : LOG_OPEN_COVER - 机器的外壳打开</p> <p>19 : LOG_CLOSE_COVER - 机器的外壳关</p> |
| | 3 | 先用 LoadGeneralLogData 或 USBLoadGeneralLogDataFromFile 指令,不然该指令被失败. |
| | 4 | <p>apnInOutMode"如下 :</p> <p>0 : LOG_IOMODE_IO - 一般方式确认</p> <p>1 : LOG_IOMODE_IN1 - 进的方式 1 确认</p> <p>2 : LOG_IOMODE_IN2 - 进的方式 2 确认</p> <p>3 : LOG_IOMODE_IN3 - 进的方式 3 确认</p> <p>4 : LOG_IOMODE_OUT1 - 出的方式 1 确认</p> <p>5 : LOG_IOMODE_OUT2 - 出的方式 2 确认</p> <p>6 : LOG_IOMODE_OUT3 - 出的方式 3 确认</p> |

2.3.8 EmptyGeneralLogData

| | | |
|----|---|---|
| 形式 | long EmptyGeneralLogData(void) | |
| 功能 | 删除所有的进出记录数据. | |
| 参数 | | |
| 返回 | 执行成功返回 1,失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 因为全部删除所有的进出记录数据, 执行该指令之前请备份重要的登记资料进出记录数据. |

2.3.9 GetGeneralLogData_1

| | |
|----|--|
| 形式 | GetGeneralLogData_1(long *apnEnrollNumber, long pnVerifyMode , long *apnInOutMode, long *apnYear, long *apnMonth, long *apnDay, long *apnHour, long *apnMinute, long *apnSec) |
|----|--|

| | | |
|----|---|--|
| 功能 | 通过 LoadGeneralLogData 或 USBLoadGeneralLogDataFromFile 指令读到 PC 内存的进出记录数据一个一个的获取。 | |
| 参数 | apnEnrollNumber | 进出用户登记号码的参数指示者 |
| | apnVerifyMode | 确认方式的参数指示者 |
| | apnInOutMode | 进出方式的参数指示者 |
| | apnYear, apnMonth apnDay, apnHour apnMinute, apnSec | 进出记录日期, 时间的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 获取完数据后返回 RUNERR_LOG_END. |
| | 2 | <p>“apnVerifyMode”如下 :</p> <p>1 : LOG_FPVERIFY - 指纹确认</p> <p>2 : LOG_PASSVERIFY - 密码确认</p> <p>3 : LOG_CARDVERIFY - 卡确认</p> <p>4 : LOG_FPPASS_VERIFY - 指纹+密码确认</p> <p>5 : LOG_FPCARD_VERIFY - 指纹+卡确认</p> <p>6 : LOG_PASSFP_VERIFY - 密码+指纹确认</p> <p>7 : LOG_CARDFP_VERIFY - 卡+指纹确认</p> <p>有控锁功能的版本里利用.(参考《2.7 门铃管理》)</p> <p>10 : LOG_OPEN_DOOR - 确认后发送开门信号</p> <p>11 : LOG_CLOSE_DOOR - 确认后发送关门信号</p> <p>12 : LOG_OPEN_HAND - 通过开门按钮发送开门信号</p> <p>13 : LOG_OPEN_THREAT -防胁迫指纹确认时发送开门信号</p> <p>14 : LOG_PROG_OPEN - 从操作机器发送开门信号</p> <p>15 : LOG_PROG_CLOSE -从操作机器发送关门信号</p> <p>16 : LOG_OPEN_IREGAL - 发送非法性开门信号</p> <p>17 : LOG_CLOSE_IREGAL -发送非法性关门信号</p> <p>18 : LOG_OPEN_COVER - 机器的外壳打开</p> <p>19 : LOG_CLOSE_COVER - 机器的外壳关</p> |
| | 3 | 先用 LoadGeneralLogData 或 USBLoadGeneralLogDataFromFile 指令, 不然该指令被失败. |
| | 4 | <p>“apnInOutMode”如下 :</p> <p>0 : LOG_IOMODE_IN - 进的方式确认</p> <p>1 : LOG_IOMODE_OUT - 出的方式确认</p> <p>2 : LOG_IOMODE_IO - 一般方式确认</p> |

2.3.10 GetSuperLogData_1

| | | |
|----|---|----------------|
| 形式 | long GetSuperLogData(long *apnSEnrollNumber, long *apnGEnrollNumber, long *apnManipulation, long *apnBackupNumber, long *apnYear, long *apnMonth, long *apnDay, long *apnHour, long *apnMinute, long *apnSec) | |
| 功能 | 通过 LoadSuperLogData 或 USBLoadSuperLogDataFromFile 的指令读到 PC 内存的管理记录一个一个的获取. | |
| 参数 | apnSEnrollNumber | 管理者的登记号码的参数指示者 |

| | | |
|----|---|---|
| | apnEnrollNumber | 被管理者的登记号码的参数指示者 |
| | apnManipulation | 管理对象的识别号码的参数指示者 |
| | apnBackupNumber | 被管理者的登记资料种类号码的参数指示者 |
| | apnYear, apnMonth apnDay, apnHour apnMinute, apnSec | 进出记录日期, 时间的参数指示者 |
| 形式 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 获取完数据后返回 RUNERR_LOG_END. |
| | 2 | 先执行 LoadSuperLogData 或 USBLoadSuperLogDataFromFile 指令, 不然该指令被失败. |
| | 3 | "apnManipulation"如下 : 3 : LOG_ENROLL_USER - 一般用户登记 4 : LOG_ENROLL_MANAGER - 管理者登记 5 : LOG_ENROLL_DELFP - 删除指纹数据 6 : LOG_ENROLL_DELPASS - 删除通过密码 7 : LOG_ENROLL_DELCARD - 删除卡数据 8 : LOG_LOG_ALLDEL - 删除所有的记录数据 9 : LOG_SETUP_SYS - 修改机器信息 10 : LOG_SETUP_TIME - 修改机器时间 11 : LOG_SETUP_LOG - 修改记录限制值 12 : LOG_SETUP_COMM - 修改通讯方式 13 : LOG_PASSTIME - 通过时间设置 14 : LOG_SETUP_DOOR - 门操作信息设置 |

2.3.11 GetRealTimeInfo

| | | |
|----|---|------------|
| 形式 | Long GetRealTimeInfo(long* apGetRealTime) | |
| 功能 | 将传输块所需等待时间和时间段信息读出到 PC 中。 | |
| 参数 | apGetRealTime | 读取数据的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |

2.3.12 SetRealTimeInfo

| | | |
|----|---|------------|
| 形式 | Long SetRealTimeInfo(long* apSetRealTime) | |
| 功能 | 将传输块所需等待时间和时间段信息写入到指纹机中。 | |
| 参数 | apSetRealTime | 写取数据的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |

2.4 用户信息管理

2.4.1 EnableUser

| | | |
|----|--|---|
| 形式 | long EnableUser(long anEnrollNumber, long anBackupNumber, long anEnableFlag) | |
| 功能 | 设置用户对机器的可用/不可用。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| | anEnableFlag | 许可标志 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | anEnableFlag = 0 表示不可用, anEnableFlag = 1 表示可用。 |

2.4.2 ModifyPrivilege

| | | |
|----|---|--|
| 形式 | long ModifyPrivilege(long anEnrollNumber, long anBackupNumber, long anMachinePrivilege) | |
| 功能 | 设置用户的机器操作权限。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | anBackupNumber | 登记资料 种类号码 |
| | anMachinePrivilege | 操作权限 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 用户按照操作机器的权限可分为一般用户和管理者。权限反映在“anMachinePrivilege”上。 anMachinePrivilege”如下： 0 : MP_NONE - 一般用户(在机器上只能做确认的用户) 1 : MP_ALL - 管理者。(可以操作机器的用户) |

2.4.3 GetUserName

| | | |
|----|--|---|
| 形式 | long GetUserName(long anEnrollNumber, char *apstrUserName) | |
| 功能 | 获取分配到用户的名称。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | apstrUserName | 存名称的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | apstrUserName”名称的最大 10byte(最多 10 个英文字母或最多 5 个其他文字)。 |
| | 2 | 没有分配的名称该指令被失败。 |

2.4.4 SetUserName

| | | |
|----|--|------|
| 形式 | long SetUserName(long anEnrollNumber, char *apstrUserName) | |
| 功能 | 给用户分配名称。 | |
| 参数 | anEnrollNumber | 登记号码 |

| | | |
|----|---|---|
| | apstrUserName | 包含名称的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | apstrUserName”里的最大的名称大小 10byte(最多 10 个英文字母或最多 5 个其他文字)。 |
| | 2 | 没有分配的用户的情况下该指令被失败。 |

2.4.5 GetNewsMessage

| | | |
|----|---|--|
| 形式 | long GetNewsMessage(long anNewsId, char *apstrNews) | |
| 功能 | 从机器获取指定的通知文。 | |
| 参数 | anNewsId | 通知文的识别号码 |
| | apstrNews | 通知文数据的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | anNewsId”是指定通知文的号码, 其范围是 '0~255. |
| | 2 | apstrUserName” 里的最大的名称大小是 48byte(最多 48 个英文字母, 其他的最多 24 跟文字)。 |

2.4.6 SetNewsMessage

| | | |
|----|---|---------------------------------|
| 形式 | long SetNewsMessage(long anNewsId, char *apstrNews) | |
| 功能 | 机器上设置通知文。 | |
| 参数 | anNewsId | 通知文的识别号码 |
| | apstrNews | 通知文数据的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考 《2.4.5 GetNewsMessage》。 |

2.4.7 GetUserNewsID

| | | |
|----|--|--|
| 形式 | long GetUserNewsID(long anEnrollNumber, long *apnNewsId) | |
| 功能 | 获取给用户分配的通知文的识别号码。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | apnNewsId | 识别号码의 参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | apnNewsId”是在 《2.4.6 SetNewsMessage》设置的值。 |

2.4.8 SetUserNewsID

| | | |
|----|--|------|
| 形式 | long SetUserNewsID(long anEnrollNumber, long anNewsId) | |
| 功能 | 给用户分配通知文的识别号码。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | anNewsId | 识别号码 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |

| | | |
|----|---|---|
| 其他 | 1 | "apnNewsId" 是在《2.4.6 SetNewsMessage》设置的值. |
|----|---|---|

2.5 机器管理

2.5.1 EnableDevice

| | | |
|----|--|--|
| 形式 | long EnableDevice(long anEnabledFlag) | |
| 功能 | 设置对机器的可用/不可用. | |
| 参数 | anEnabledFlag | 许可标志 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 为与 PC 和机器的通讯停止机器操作的时候利用. |
| | 2 | anEnabledFlag=0, 表示停止显示“工作中”的通知文. anEnabledFlag=1, 表示允许机器操作, 显示正常界面. |

2.5.2 PowerOnAllDevice

| | | |
|----|-----------------------------|-----------------------|
| 形式 | void PowerOnAllDevice(void) | |
| 功能 | 启动连接的机器. | |
| 参数 | | |
| 返回 | 没有 | |
| 其他 | 1 | 该指令进行 RS-485 通讯的时候利用. |

2.5.3 PowerOffDevice

| | | |
|----|--|----------------------|
| 形式 | long PowerOffDevice(void) | |
| 功能 | 关闭机器的电源. | |
| 参数 | | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 执行该指令的话, 机器断开连接关闭电源. |

2.5.4 GetDeviceTime

| | | |
|----|--|--------------|
| 形式 | long GetDeviceTime(DATE* apnDateTime) | |
| 功能 | 获取机器的日期, 时间. | |
| 参数 | apnDateTime | 日期, 时间的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | |

2.5.5 SetDeviceTime

| | | |
|----|--|--------|
| 形式 | long SetDeviceTime(DATE anDateTime) | |
| 功能 | 设置机器的日期, 时间. | |
| 参数 | apnDateTime | 日期, 时间 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |

| | | |
|----|---|--|
| 其他 | 1 | |
|----|---|--|

2.5.6 GetDeviceStatus

| | | |
|----|--|--|
| 形式 | long GetDeviceStatus(long anStatusIndex, long *apnValue) | |
| 功能 | 获取现在机器上存在的状态值. | |
| 参数 | anStatusIndex | 机器状态识别号码 |
| | apnValue | 状态值的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 该指令的作用是在 PC 上了解机器的状态的作用. |
| | 2 | anStatusIndex"如下 : 1 : GET_MANAGERS - 现存在的管理者的个数 2 : GET_USERS - 现存在的一般用户的个数 3 : GET_FPS - 现存在的指纹数据的个数 4 : GET_PSWs - 现存在的密码数据的个数 5 : GET_SLOGS - 现存在的新的管理记录数据的个数 6 : GET_GLOGS - 现存在的新进出记录数据的个数 7 : GET_ASLOGS - 现存在的所有的管理记录数据的个数 8 : GET_AGLOGS - 现存在的所有的进出记录数据的个数 9 : GET_CARDS - 现存在的卡数据的个数 |

2.5.7 GetDeviceInfo

| | | |
|----|--|---|
| 形式 | long GetDeviceInfo(long anInfoIndex, long *apnValue) | |
| 功能 | 获取机器信息. | |
| 参数 | anInfoIndex | 机器信息识别号码 |
| | apnValue | 信息值的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | anInfoIndex"如下 : 1 : DI_MANAGERS - 管理者最多登记数 2 : DI_MACHINENUM - 机号 3 : DI_LANGAUGE - 机器上要显示的语言 4 : DI_POWEROFF_TIME - 自定电源关闭 5 : DI_LOCK_CTRL - 控锁操作标志 6 : DI_GLOG_WARNING - 最多进出记录数 记录的时候能登记的数值比该数值小的话发出报警。 7 : DI_SLOG_WARNING - 最多管理记录数, 记录的时候能登记的数值比该数值小的话发出报警。 8 : DI_VERIFY_INTERVALS - 可以进行反复确认的时间隔断。 该时间内进行反复确认时, 未留下记录。 9 : DI_RSCOM_BPS - 串行通讯波特率。没个波特率的值如下。 BPS_9600 = 3 BPS_19200 = 4 |

| | | |
|--|--|--|
| | | BPS_38400 = 5 BPS_57600 = 6 BPS_115200 = 7 10: DI_DATE_SEPARATE- 日期, 时间显示形式 11: DI_VERIFY_KIND: 比对模式设置, 比对模式设置值如下。 0: F / P / C 1: F + P 2: F + C 3: C |
|--|--|--|

2.5.8 SetDeviceInfo

| | | |
|----|--|--|
| 形式 | long SetDeviceInfo(long anInfoIndex, long anValue) | |
| 功能 | 设置机器信息. | |
| 参数 | anInfoIndex | 机器信息识别号码 |
| | anValue | 信息值 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | "anInfoIndex" 的具体值跟 《2.5.7 GetDeviceInfo》一样. |

2.5.9 GetProductData

| | | |
|----|--|---|
| 形式 | long GetProductData(long anProductIndex, char *apstrProductData) | |
| 功能 | 获取卖方注明的产品销售信息. | |
| 参数 | anProductIndex | 销售信息识别号码 |
| | apstrProductData | 销售信息的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | "anProductIndex"如下 : 1 : PRODUCT_SERIALNUMBER - 序列号 2 : PRODUCT_BACKUPNUMBER - 预定号码 3 : PRODUCT_CODE - 版本号 4 : PRODUCT_NAME - 版本名称 5 : PRODUCT_WEB - 销售者的网址 6 : PRODUCT_DATE - 销售日期 7 : PRODUCT_SENDO - 买方名称 |

2.5.10 GetDeviceVersion

| | | |
|----|--|----------|
| 形式 | long GetDeviceVersion(long *apnVersion) | |
| 功能 | 按每个型号获取每个型号上反应的更新版本. | |
| 参数 | apnVersion | 版本的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | |

2.5.11 GetDeviceTime_1

| | | |
|----|---|--------------|
| 形式 | long GetDeviceTime_1(long *apnYear, long *apnMonth, long *apnDay, long *apnHour, long *apnMinute, long *apnSec, long* apnDayOfWeek) | |
| 功能 | 获取机器的日期, 时间. | |
| 参数 | apnYear, apnMonth apnDay, apnHour apnMinute, apnSec, apnDayOfWeek | 日期, 时间的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | |

2.5.12 SetDeviceTime_1

| | | |
|----|---|--------|
| 形式 | long SetDeviceTime_1(long anYear, long anMonth, long anDay, long anHour, long anMinute, long anSec, long anDayOfWeek) | |
| 功能 | 设置机器的日期, 时间. | |
| 参数 | anYear, anMonth anDay, anHour, anMinute, anSec, anDayOfWeek | 日期, 时间 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | |

2.6 响铃管理

2.6.1 GetBellTime

| | | |
|----|---|---|
| 形式 | long GetBellTime(long *apnBellCount, long *aptBellInfo) | |
| 功能 | 获取响铃设置信息. | |
| 参数 | apnBellCount | 响铃个数的参数指示者 |
| | aptBellInfo | 响铃信息结构体的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 一次响的响铃声的个数返回到“apnBellCount”. |
| | 2 | 被设置的响铃个数, 设置时间等, 关于响铃的信息返回到“aptBellInfo”. “aptBellInfo”的说明请参考《4.1.1 BELLINFO》. |

2.6.2 GetBellTimeWithString

| | | |
|----|---|-----------------------------|
| 形式 | long GetBellTimeWithString(long *apnBellCount, char *apstrBellInfo) | |
| 功能 | 与 GetBellTime 指令一样, 响铃信息以文字列形式获取. | |
| 参数 | apnBellCount | 响铃个数的参数指示者 |
| | apstrBellInfo | 响铃信息文字列的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 具体说明请参考《2.6.1 GetBellTime》. |

2.6.3 SetBellTime

| | | |
|----|---|--|
| 形式 | long SetBellTime(long anBellCount, long *aptBellInfo) | |
| 功能 | 机器上设置响铃信息. | |
| 参数 | anBellCount | 响铃个数 |
| | aptBellInfo | 响铃信息结构体的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 在“anBellCount”上一次想的响铃的个数设置. |
| | 2 | 在“aptBellInfo”上设置被设置的响铃个数, 设置时间等, 关于响铃的信息. “aptBellInfo”的说明请参考《4.1.1 BELLINFO》. |

2.6.4 SetBellTimeWithString

| | | |
|----|---|-----------------------------|
| 形式 | long SetBellTimeWithString(long anBellCount, char *apstrBellInfo) | |
| 功能 | 与 SetBellTime 指令一样, 将响铃信息以文字列形式设置. | |
| 参数 | anBellCount | 响铃个数 |
| | apstrBellInfo | 响铃信息结构体的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 具体说明请参考《2.6.3 SetBellTime》. |

2.7 门铃管理

该功能不是所有版本上都有的。

2.7.1 GetDoorStatus

| | | |
|----|--|---|
| 形式 | long GetDoorStatus(long *apnStatusVal) | |
| 功能 | 获取开门状态. | |
| 参数 | apnStatusVal | 状态值的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | “apnStatusVal”:如下 : 0 : DOOR_CONTROLRESET - 机器的门控制状态. 1 : DOOR_OPEN - 门已开. 2 : DOOR_CLOSED - 门已关. 3 : DOOR_COMMNAD-按照门控制指令, 门开了后过了一段时间自动关门. |

2.7.2 SetDoorStatus

| | | |
|----|--|---|
| 形式 | long SetDoorStatus(long anStatusVal) | |
| 功能 | 操作开门状态. | |
| 参数 | anStatusVal | 状态值 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | “anStatusVal”说明请参考 《2.7.1 GetDoorStatus》. |

2.7.3 GetPassTime

| | | |
|----|---|--|
| 形式 | long GetPassTime(long anPassTimeID, long *apnPassTime, long anPassTimeSize) | |
| 功能 | 获取开门关门的时间段的信息(TimeZone). | |
| 参数 | anPassTimeID | 时间段信息识别号码 |
| | apnPassTime | 时间段信息结构体的参数指示者 |
| | anPassTimeSize | 时间段信息结构体的长度 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | “anPassTimeID”是指时间段信息结构体的号码. 最多可以设置 50 个结构体, 该值是 ‘0’~‘49’. |
| | 2 | “apnPassTime”反映 “anPassTimeID”指定的时间段信息结构体的值. 该结构体按照每个星期具有 7 个时间段. 《4.1.2 PASSCTRLTIME 结构体》. |
| | 3 | “anPassTimeSize”是 “apnPassTime”的长度, 为判断 API 的结构体的长度够不够被提供. |

2.7.4 GetPassTimeWithString

| | | |
|----|--|-----------------------------|
| 形式 | long GetPassTimeWithString(long anPassTimeID, char *apstrPassTime) | |
| 功能 | 与 GetPassTime 一样，时间段信息以文字列形式返回。 | |
| 参数 | anPassTimeID | 时间段信息识别号码 |
| | apnPassTime | 时间段信息结构体文字列形式的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考《2.7.3 GetPassTime》。 |

2.7.5 SetPassTime

| | | |
|----|---|-----------------------------|
| 形式 | long SetPassTime(long anPassTimeID, long *apnPassTime, long anPassTimeSize) | |
| 功能 | 设置开门关门的时间段信息 (TimeZone)。 | |
| 参数 | anPassTimeID | 时间段信息识别号码 |
| | apnPassTime | 时间段信息结构体的参数指示者 |
| | anPassTimeSize | 时间段信息结构体的长度 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考《2.7.3 GetPassTime》。 |

2.7.6 SetPassTimeWithString

| | | |
|----|--|-----------------------------|
| 形式 | long SetPassTimeWithString(long anPassTimeID, char *apstrPassTime) | |
| 功能 | 与 SetPassTime 一样，时间段信息以文字列形式输入。 | |
| 参数 | anPassTimeID | 时间段信息识别号码 |
| | apnPassTime | 时间段信息结构体文字列的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考《2.7.3 GetPassTime》。 |

2.7.7 GetUserPassTime

| | | |
|----|---|--|
| 形式 | long GetUserPassTime(long anEnrollNumber, long *apnGroupID, long *apnPassTimeID, long anPassTimeIDSize) | |
| 功能 | 获取给用户分配的时间段信息组合与个别分配的时间段信息。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | apnGroupID | 组合号码的参数指示者 |
| | apnPassTimeID | 时间段信息识别号码结构体的参数指示者 |
| | anPassTimeIDSize | 识别号码结构体的长度 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | apnGroupID”的说明请参考《2.7.11 GetGroupPassTime》。 |
| | 2 | apnPassTimeID”是给用户分配的所有时间段信息识别号码的排列型组合结构体。结构体的定义请参考《4.1.3 USERPASSINFO 结构体》。识别号码的说明请参考《2.7.3 GetPassTime》。 |

| | | |
|--|---|--|
| | 3 | anPassTimeSize”是“apnPassTime”的长度， 被提供于 API 的结构体的长度够不够。 |
|--|---|--|

2.7.8 GetUserPassTimeWithString

| | | |
|----|--|---------------------------------|
| 形式 | long GetUserPassTimeWithString(long anEnrollNumber, long *apnGroupID, char *apstrPassTimeID) | |
| 功能 | 与 GetUserPassTime 指令一样, 将识别号码结构体以文字列形式返回。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | apnGroupID | 组合号码的参数指示者 |
| | apstrPassTimeID | 时间段信息识别号码结构体文字列的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考《2.7.7 GetUserPassTime》。 |

2.7.9 SetUserPassTime

| | | |
|----|---|---------------------------------|
| 形式 | long SetUserPassTime(long anEnrollNumber, long anGroupID, long *apnPassTimeID, long anPassTimeIDSize) | |
| 功能 | 给用户设置时间段信息组合与给各个分配的时间段信息。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | anGroupID | 组合号码 |
| | apnPassTimeID | 时间段信息识别号码结构体的参数指示者 |
| | anPassTimeIDSize | 识别号码结构体的长度 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考《2.7.7 GetUserPassTime》。 |

2.7.10 SetUserPassTimeWithString

| | | |
|----|--|---------------------------------|
| 形式 | long SetUserPassTimeWithString(long anEnrollNumber, long anGroupID, char *apstrPassTimeID) | |
| 功能 | 与 SetUserPassTime 指令一样, 将识别号码结构体以文字列形式输入。 | |
| 参数 | anEnrollNumber | 登记号码 |
| | anGroupID | 组合号码 |
| | apstrPassTimeID | 时间段信息识别号码结构体文字列的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考《2.7.7 GetUserPassTime》。 |

2.7.11 GetGroupPassTime

| | | |
|----|---|--------------------|
| 形式 | long GetGroupPassTime(long anGroupID, long *apnPassTimeID, long anPassTimeIDSize) | |
| 功能 | 获取时间段信息组合里的每个时间段信息的识别号码。 | |
| 参数 | anGroupID | 组合号码 |
| | apnPassTimeID | 时间段信息识别号码结构体的参数指示者 |
| | anPassTimeIDSize | 识别号码结构体的长度 |

| | | |
|----|--|---|
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 在机器上将几个时间段信息结构体组合再组合起来可以使用, “anGroupID”是指该组合的号码. 最多可设置 5 个组合, 该值是 ‘1’~‘5’. |
| | 2 | “apnPassTimeID”是分配到每个组合的时间段信息识别号码的排列型组合结构体, 一个组合里可以设置 3 个结构体识别号码, 结构体的定义参考《4.1.4 GROUPPASSINFO 结构体》, 识别号码的说明请参考《2.7.3 GetPassTime》. |
| | 3 | “anPassTimeIDSize”是 “apnPassTimeID”的长度, 让 API 判断结构体的长度够不够. |

2.7.12 GetGroupPassTimeWithString

| | | |
|----|--|-----------------------------------|
| 形式 | long GetGroupPassTimeWithString(long anGroupID, char *apstrPassTimeID) | |
| 功能 | 与 GetGroupPassTime 指令一样, 将识别号码结构体以文字列形式返回. | |
| 参数 | anGroupID | 组合号码 |
| | apstrPassTimeID | 时间段信息识别号码结构体文字列的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 具体说明请参考《2.7.11 GetGroupPassTime》. |

2.7.13 SetGroupPassTime

| | | |
|----|---|-----------------------------------|
| 形式 | long SetGroupPassTime(long anGroupID, long *apnPassTimeID, long anPassTimeIDSize) | |
| 功能 | 时间段信息组合上设置时间段信息的识别号码. | |
| 参数 | anGroupID | 组合号码 |
| | apnPassTimeID | 时间段信息识别号码结构体的参数指示者 |
| | anPassTimeIDSize | 识别号码结构体的长度 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 具体说明请参考《2.7.11 GetGroupPassTime》. |

2.7.14 SetGroupPassTimeWithString

| | | |
|----|--|-----------------------------------|
| 形式 | long SetGroupPassTimeWithString(long anGroupID, char *apstrPassTimeID) | |
| 功能 | 与 SetGroupPassTime 指令一样, 将识别号码结构体以文字列输入. | |
| 参数 | anGroupID | 组合号码 |
| | apstrPassTimeID | 时间段信息识别号码结构体文字列的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | 具体说明请参考《2.7.11 GetGroupPassTime》. |

2.7.15 GetGroupMatch

| | | |
|----|--|----------------|
| 形式 | long GetGroupMatch(long *apnGroupMatch, long anGroupMatchSize) | |
| 功能 | 获取时间段信息结构体的组合的门操作组合 | |
| 参数 | apnGroupMatch | 组合的组合结构体的参数指示者 |

| | | |
|----|---|---|
| | anGroupMatchSize | 组合结构体的长度 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 将时间段信息结构体的组合组织后用在开门/关门操作。 最多可以组合 10 个组, “apnGroupMatch”是该组合的排列型组合结构体。 结构体的定义请参考《4.1.5 GROUPMATCHINFO 结构体》。 结构体的项目里连续描述组合号码。 比如: 1 号和 3 号同时组合的情况下就是‘13’, 1 号, 3 号和 5 号同时组合的情况下就是 ‘135’。 |
| | 2 | “anPassTimeIDSize”是 “apnPassTimeID” 的长度, 被提供于 API 的结构体的长度够不够。 |

2.7.16 GetGroupMatchWithString

| | | |
|----|---|--------------------------------|
| 形式 | long GetGroupMatchWithString(char *apstrGroupMatch) | |
| 功能 | 与 GetGroupMatchTime 指令一样。将组合结构体以文字列形式返回。 | |
| 参数 | apstrGroupMatch | 组合的组合结构体文字列的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考《2.7.15 GetGroupMatch》。 |

2.7.17 SetGroupMatch

| | | |
|----|--|--------------------------------|
| 形式 | long SetGroupMatch(long *apnGroupMatch, long anGroupMatchSize) | |
| 功能 | 设置时间段信息结构体的组合的门操作组合。 | |
| 参数 | apnGroupMatch | 组合的组合结构体的参数指示者 |
| | anGroupMatchSize | 组合结构体的长度。 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考《2.7.15 GetGroupMatch》。 |

2.7.18 SetGroupMatchWithString

| | | |
|----|---|--------------------------------|
| 形式 | long SetGroupMatchWithString(char *apstrGroupMatch) | |
| 功能 | 与 SetGroupMatch 指令一样, 将组合结构体以文字列登记。 | |
| 参数 | apstrGroupMatch | 组合的组合结构体文字列的参数指示者 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考《2.7.15 GetGroupMatch》。 |

2.8 夏天时间制管理

该功能不是所有版本上都有的.

2.8.1 GetAdjustInfo

| | | |
|----|--|--|
| 形式 | long GetAdjustInfo(long* dwAdjustedState, long* dwAdjustedMonth, long* dwAdjustedDay, long* dwAdjustedHour, long* dwAdjustedMinute, long* dwRestoredState, long* dwRestoredMonth, long* dwRestoredDay, long* dwRestoredHour, long* dwRestoredMinute) | |
| 功能 | 获取夏天时间状态. | |
| 参数 | 状态值的参数指示者 | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | “dwAdjustedState, dwRestoredState”:如下 : None : 无变化. -1 : 早一个小时. +1 : 晚一个小时. |

2.8.2 SetAdjustInfo

| | | |
|----|--|--|
| 形式 | long SetAdjustInfo(long dwAdjustedState, long dwAdjustedMonth, long dwAdjustedDay, long dwAdjustedHour, long dwAdjustedMinute, long dwRestoredState, long dwRestoredMonth, long dwRestoredDay, long dwRestoredHour, long dwRestoredMinute) | |
| 功能 | 操作夏天时间制状态. | |
| 参数 | 状态值(参考 4.1.6 结构体) | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | “dwAdjustedState, dwRestoredState”:如下 : None : 无变化. -1 : 早一个小时. +1 : 晚一个小时. |

2.9 机器信息管理

2.9.1 GetServerNetInfo

| | | |
|----|--|---------|
| 形式 | long GetServerInfo(BSTR* astrServerIPAddress, long* apServerPort, long* apServerRequest) | |
| 功能 | 读取从机器服务器的信息 | |
| 参数 | astrServerIPAddress | 服务地址 |
| | apServerPort | 服务端口号码 |
| | apServerRequest | 设定实时间方式 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | |

2.9.2 SetServerNetInfo

| | | |
|----|---|---------|
| 形式 | long SetServerNetInfo(LPCTSTR astrServerIPAddress, long anServerPort, long anServerRequest) | |
| 功能 | 把服务器的信息写取在机器里面 | |
| 参数 | astrServerIPAddress | 服务地址 |
| | anServerPort | 服务端口号码 |
| | anServerRequest | 设定实时间方式 |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码。错误代码的具体说明参考《4.2 错误代码表》。 | |
| 其他 | 1 | |

2.9.3 SetUSBModel

| | | |
|----|--------------------------------------|--|
| 形式 | void SetUSBModel(long anModel) | |
| 功能 | 指定写入（USB Flash）U 盘或从其读出的信息是对应于那款型号机器 | |
| 参数 | anModel | <p>是表示型号信息的 long 型变量</p> <pre> #define FK625_FP1000 2001 #define FK625_FP2000 2002 #define FK625_FP3000 2003 #define FK625_FP5000 2004 #define FK625_FP10000 2005 #define FK625_FP30000 2006 #define FK625_ID30000 2007 #define FK635_FP700 3001 #define FK635_FP3000 3002 #define FK635_FP10000 3003 #define FK635_ID30000 3004 #define FK723_FP1000 4001 #define FK725_FP1000 5001 #define FK725_FP1500 5002 #define FK725_ID5000 5003 #define FK725_ID30000 5004 #define FK735_FP500 6001 #define FK735_FP3000 6002 #define FK735_ID30000 6003 #define FK925_FP3000 7001 #define FK935_FP3000 8001 </pre> |

3 FKAttend.DLL 的界面

FKAttend.DLL 的界面与 FKAttend.OCX 相似。

进行多种连接时，返回第一次通讯的时候获取的接口的识别号码，然后别的接口上也用该识别号码实现与机器的通讯，该功能与 FKAttend.OCX 不一样。

下面解释对应的函数和差异点。

3.1 机器连接与断开

3.1.1 FK_ConnectComm

| | | |
|----|---|------------------------------|
| 形式 | long FK_ConnectComm(long nMachineNo, long nComPort, long nBaudRate, char *pstrTelNumber, long nWaitDialTime, long nLicense) | |
| 功能 | 通过 RS-232/485 线为了机器连接打开 COM 接口。 | |
| 返回 | 执行成功后，返回接口的的识别号码。该值应该比 0 大的值 若失败返回对应的错误代码。 错误代码的具体说明参考 《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考 《2.1.1 ConnectComm》。 |

3.1.2 FK_ConnectNet

| | | |
|----|---|-----------------------------|
| 形式 | long FK_ConnectNet(long nMachineNo, char * pstrIpAddress, long nNetPort, long nTimeOut, long nProtocolType, long nNetPassword, long nLicense) | |
| 功能 | 通过网线连接机器的时候打开网接口。 | |
| 返回 | 执行成功后，返回接口的的识别号码。该值应该比 0 大的值 若失败返回对应的错误代码。 错误代码的具体说明参考 《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考 《2.1.2 ConnectNet》。 |

3.1.3 FK_ConnectUSB

| | | |
|----|--|-----------------------------|
| 形式 | long FK_ConnectUSB(long nMachineNo, long nLicense) | |
| 功能 | 为了通过 USB 线通讯打开 USB 接口。 | |
| 返回 | 执行成功后，返回接口的的识别号码。该值应该比 0 大的值 若失败返回对应的错误代码。 错误代码的具体说明参考 《4.2 错误代码表》。 | |
| 其他 | 1 | 具体说明请参考 《2.1.3 ConnectUSB》。 |

3.1.4 FK_DisConnect

| | | |
|----|---------------------------------------|---|
| 形式 | void FK_DisConnect(long nHandleIndex) | |
| 功能 | 断开与机器的连接。 | |
| 其他 | 1 | nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.1.4 DisConnect》。 |

3.1.5 FK_ConnectGetIP

| | |
|----|-------------------------------------|
| 形式 | long ConnectGetIP(BSTR *strComName) |
|----|-------------------------------------|

| | | |
|----|----------------|-------------------------------------|
| 功能 | 以名称生成 IP 地址的函数 | |
| 参数 | strComName | 待寻找 IP 地址机器的名字 |
| 其他 | 1 | “nLicense”是 《2.1.1 ConnectComm》和一样. |

3.2 登记数据管理

3.2.1 FK_GetEnrollData

| | | |
|----|---|--|
| 形式 | long FK_GetEnrollData(long nHandleIndex, long nEnrollNumber, long nBackupNumber, long * pnMachinePrivilege, void * pnEnrollData, long * pnPassWord) | |
| 功能 | 获取用户的操作授权和登记资料. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码. |
| | 2 | 具体说明请参考 《2.2.1 GetEnrollData》. |

3.2.2 FK_GetEnrollDataWithString

| | | |
|----|---|--|
| 形式 | long FK_GetEnrollDataWithString(long nHandleIndex, long nEnrollNumber, long nBackupNumber, long * pnMachinePrivilege, LPSTR *apstrEnrollData) | |
| 功能 | 与 FK_GetEnrollData 指令象试, 登记资料以文字列形式获取. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码. |
| | 2 | 具体说明请参考 《2.2.2 GetEnrollDataWithString》. |

3.2.3 FK_PutEnrollData

| | | |
|----|---|--|
| 形式 | long FK_PutEnrollData(long nHandleIndex, long nEnrollNumber, long nBackupNumber, long nMachinePrivilege, void * pnEnrollData, long nPassWord) | |
| 功能 | 将要登记的用户的操作权限和登记资料传送到机器. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码. |
| | 2 | 具体说明请参考 《2.2.3 PutEnrollData》. |

3.2.4 FK_PutEnrollDataWithString

| | | |
|----|---|--|
| 形式 | long FK_PutEnrollDataWithString(long nHandleIndex, long nEnrollNumber, long nBackupNumber, long nMachinePrivilege, char *apstrEnrollData) | |
| 功能 | 与 FK_PutEnrollData 指令相似, 将登记资料以文字列形式记录. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码. |
| | 2 | 具体说明请参考 《2.2.4 PutEnrollDataWithString》. |

3.2.5 FK_SaveEnrollData

| | | |
|----|---|--|
| 形式 | long FK_SaveEnrollData(long nHandleIndex) | |
| 功能 | 用 FK_PutEnrollData 或 FK_PutEnrollDataWithString 指令被传送的登记资料登记到机器上. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码. |

| | | |
|--|---|---------------------------------|
| | 2 | 具体说明请参考 《2.2.5 SaveEnrollData》。 |
|--|---|---------------------------------|

3.2.6 FK_DeleteEnrollData

| | | |
|----|---|--|
| 形式 | long FK_DeleteEnrollData(long nHandleIndex, long nEnrollNumber, long nBackupNumber) | |
| 功能 | 从删除指定的登记资料。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.6 DeleteEnrollData》。 |

3.2.7 FK_USBReadAllEnrollDataFromFile

| | | |
|----|--|--|
| 形式 | long FK_USBReadAllEnrollDataFromFile(long nHandleIndex, char * pstrFilePath) | |
| 功能 | 机器的 USB 存储器里的文件中将登记资料读到 PC 的内部存储器里进行分析。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.7 USBReadAllEnrollDataFromFile》。 |

3.2.8 FK_USBReadAllEnrollDataCount

| | | |
|----|--|--|
| 形式 | long FK_USBReadAllEnrollDataCount(long nHandleIndex, long * pnValue) | |
| 功能 | 返回通过 FK_USBReadAllEnrollDataFromFile 指令读到 PC 内存的登记资料的个数。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.8 USBReadAllEnrollDataCount》。 |

3.2.9 FK_USBGetOneEnrollData

| | | |
|----|--|--|
| 形式 | long FK_USBGetOneEnrollData(long nHandleIndex, long * pnEnrollNumber, long * pnBackupNumber, long * pnMachinePrivilege, void * pnEnrollData, long * pnPassword, long * pnEnableFlag, LPSTR * dwEnrollName) | |
| 功能 | 获取通过 FK_USBReadAllEnrollDataFromFile 指令读的登记资料。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.9 USBGetOneEnrollData》。 |

3.2.10 FK_USBGetOneEnrollDataWithString

| | | |
|----|--|--|
| 形式 | long FK_USBGetOneEnrollDataWithString(long nHandleIndex, long * pnEnrollNumber, long * pnBackupNumber, long * pnMachinePrivilege, LPSTR *apstrEnrollData, long * pnEnableFlag, LPSTR * dwEnrollName) | |
| 功能 | 与 FK_USBGetOneEnrollData 指令相似，将登记资料以文字列形式获取。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.10 USBGetOneEnrollDataWithString》。 |

3.2.11 FK_USBSetOneEnrollData

| | | |
|----|--|--|
| 形式 | long FK_USBSetOneEnrollData(long nHandleIndex, long nEnrollNumber, long nBac | |
|----|--|--|

| | | |
|----|---|--|
| | kupNumber, long nMachinePrivilege, void * pnEnrollData, long nPassWord, long nEnableFlag, char *dwEnrollName) | |
| 功能 | 为了用户的操作权限和登记资料转换成 USB 存储器用文件, 在 PC 的内部存储器进行形式化。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.11 USBSetOneEnrollData》。 |

3.2.12 FK_USBSetOneEnrollDataWithString

| | | |
|----|---|--|
| 形式 | long FK_USBSetOneEnrollDataWithString(long nHandleIndex, long nEnrollNumber, long nBackupNumber, long nMachinePrivilege, char *apstrEnrollData, long nEnableFlag, char *dwEnrollName) | |
| 功能 | 与 FK_USBSetOneEnrollData 指令一样, 将登记资料以文字列形式登记。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.12 USBSetOneEnrollDataWithString》。 |

3.2.13 FK_USBWriteAllEnrollDataToFile

| | | |
|----|---|--|
| 形式 | long FK_USBWriteAllEnrollDataToFile(long nHandleIndex, char * pstrFilePath) | |
| 功能 | 通过 FK_USBSetOneEnrollData 或 FK_USBSetOneEnrollDataWithString 指令在 PC 的内存上制作形式性的登记资料文件。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.13 USBWriteAllEnrollDataToFile》。 |

3.2.14 FK_ReadAllUserID

| | | |
|----|--|--|
| 形式 | long FK_ReadAllUserID(long nHandleIndex) | |
| 功能 | 所有的机器上的用户信息数据读到 PC 的内存上。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.14 ReadAllUserID》。 |

3.2.15 FK_GetAllUserID

| | | |
|----|---|--|
| 形式 | long FK_GetAllUserID(long nHandleIndex, long * pnEnrollNumber, long * pnBackupNumber, long * pnMachinePrivilege, long * pnEnableFlag) | |
| 功能 | 将通过 FK_ReadAllUserID 读过来的用户信息, 一个一个的获取。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.15 GetAllUserID》。 |

3.2.16 FK_EmptyEnrollData

| | | |
|----|--|--|
| 形式 | long FK_EmptyEnrollData(long nHandleIndex) | |
| 功能 | 删除被登记的所有登记资料。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |

| | | |
|--|---|-----------------------------------|
| | 2 | 具体说明请参考 《2.2.16 EmptyEnrollData》。 |
|--|---|-----------------------------------|

3.2.17 FK_ClearKeeperData

| | | |
|----|--|--|
| 形式 | long FK_ClearKeeperData(long nHandleIndex) | |
| 功能 | 从机器上删除所有登记资料和记录数据。(机器出初始化功能。) | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.17 ClearKeeperData》。 |

3.2.18 FK_BenumbAllManager

| | | |
|----|---|--|
| 形式 | long FK_BenumbAllManager(long nHandleIndex) | |
| 功能 | 全部删除管理权限信息，将所有的用户设置为一般用户。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.18 BenumbAllManager》。 |

3.2.19 FK_GetVerifyMode

| | | |
|--------|--|--|
| Type | long FK_GetVerifyMode(long nHandleIndex ,long anEnrollNumber, long *apnVerifyMode) | |
| Others | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考“2.2.19 GetVerifyMode”。 |

3.2.20 FK_SetVerifyMode

| | | |
|--------|--|--|
| Type | long FK_SetVerifyMode(long nHandleIndex ,long anEnrollNumber, long anVerifyMode) | |
| Others | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考“2.2.20 SetVerifyMode”。 |

3.2.21 FK_USBGetOneEnrollData_1

| | | |
|--------|---|--|
| Type | long FK_USBGetOneEnrollData_1((long nHandleIndex ,long *apnEnrollNumber, long *apnBackupNumber, long *apnVerifyMode, long *apnMachinePrivilege, long *apnEnrollData, long *apnPassWord, long *apnEnableFlag, BSTR *apnEnrollName) | |
| Others | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考“2.2.21 USBGetOneEnrollData_1”。 |

3.2.22 FK_USBGetOneEnrollDataWithString_1

| | | |
|------|---|--|
| Type | long FK_USBGetOneEnrollDataWithString_1(long nHandleIndex ,long *apnEnrollNumber, long *apnBackupNumber, long *apnVerifyMode, long *apnMachinePrivilege, BSTR* apstrEnrollData, long *apnEnableFlag, BSTR *apnEnrollName) | |
|------|---|--|

| | | |
|--------|---|--|
| Others | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考“2.2.22 USBGetOneEnrollDataWithString_1”。 |

3.2.23 FK_USBSetOneEnrollData_1

| | | |
|--------|--|--|
| Type | long USBSetOneEnrollData(long nHandleIndex ,long anEnrollNumber, long anBackupNumber, long anVerifyMode, long anMachinePrivilege, long *apnEnrollData, long anPassWord, long anEnableFlag, LPCTSTR anEnrollName) | |
| Others | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考“2.2.23 USBSetOneEnrollData_1”。 |

3.2.24 FK_USBSetOneEnrollDataWithString_1

| | | |
|--------|---|--|
| Type | long FK_USBSetOneEnrollDataWithString_1(long nHandleIndex ,long anEnrollNumber, long anBackupNumber, long anVerifyMode, long anMachinePrivilege, BSTR apstrEnrollData, long anEnableFlag, LPCTSTR anEnrollName) | |
| Others | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考“2.2.24 USBSetOneEnrollDataWithString_1”。 |

3.2.25 FK_USBReadAllEnrollDataFromFile_Color

| | | |
|----|---|--|
| 形式 | long FK_USBReadAllEnrollDataFromFile_Color (long nHandleIndex, char * pstrFilePath) | |
| 功能 | 机器的 USB 存储器里的文件中将登记资料读到 PC 的内部存储器里进行分析。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.25 USBReadAllEnrollDataFromFile_Color》。 |

3.2.26 FK_USBWriteAllEnrollDataToFile_Color

| | | |
|----|---|--|
| 形式 | long FK_USBWriteAllEnrollDataToFile_Color (long nHandleIndex, char * pstrFilePath, long anNewsKind) | |
| 功能 | 通过 FK_USBSetOneEnrollData 或 FK_USBSetOneEnrollDataWithString 指令在 PC 的内存上制作形式性的登记资料文件。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.26 USBWriteAllEnrollDataToFile_Color》。 |

3.2.27 FK_USBGetOneEnrollData_Color

| | | |
|----|--|--|
| 形式 | long FK_USBGetOneEnrollData_Color (long nHandleIndex, long * pnEnrollNumber, long * pnBackupNumber, long * pnMachinePrivilege, void * pnEnrollData, long * pnPassWord, long * pnEnableFlag, LPSTR * dwEnrollName, long anNewsKind) | |
| 功能 | 获取通过 FK_USBReadAllEnrollDataFromFile 指令读的登记资料。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.27 USBGetOneEnrollData_Color》。 |

3.2.28 FK_USBGetOneEnrollDataWithString_Color

| | | |
|----|---|--|
| 形式 | long FK_USBGetOneEnrollDataWithString(long nHandleIndex, long * pnEnrollNumber, long * pnBackupNumber, long * pnMachinePrivilege, LPSTR *apstrEnrollData, long * pnEnableFlag, LPSTR * dwEnrollName, long anNewsKind) | |
| 功能 | 与 FK_USBGetOneEnrollData 指令相似，将登记资料以文字列形式获取。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.28 USBGetOneEnrollDataWithString_Color》。 |

3.2.29 FK_USBSetOneEnrollData_Color

| | | |
|----|---|--|
| 形式 | long FK_USBSetOneEnrollData_Color (long nHandleIndex, long nEnrollNumber, long nBackupNumber, long nMachinePrivilege, void * pnEnrollData, long nPassword, long nEnableFlag, char *dwEnrollName, long anNewsKind) | |
| 功能 | 为了用户的操作权限和登记资料转换成 USB 存储器用文件，在 PC 的内部存储器进行形式化。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.29 USBSetOneEnrollData_Color》。 |

3.2.30 FK_USBSetOneEnrollDataWithString_Color

| | | |
|----|--|--|
| 形式 | long FK_USBSetOneEnrollDataWithString_Color(long nHandleIndex, long nEnrollNumber, long nBackupNumber, long nMachinePrivilege, char *apstrEnrollData, long nEnableFlag, char *dwEnrollName, long anNewsKind) | |
| 功能 | 与 FK_USBSetOneEnrollData 指令一样，将登记资料以文字列形式登记。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.2.30 USBSetOneEnrollDataWithString_Color》。 |

3.3 记录数据管理

3.3.1 FK_LoadSuperLogData

| | | |
|----|---|--|
| 形式 | long FK_LoadSuperLogData(long nHandleIndex, long nReadMark) | |
| 功能 | 从机器将管理记录读到 PC 的内存上进行分析. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.1 LoadSuperLogData》. |

3.3.2 FK_USBLoadSuperLogDataFromFile

| | | |
|----|--|--|
| 形式 | long FK_USBLoadSuperLogDataFromFile(long nHandleIndex, char *astrFilePath) | |
| 功能 | 机器的 USB 存储器里的管理记录数据文件中将记录数据读到 PC 的内部存储器里进行分析. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.2 USBLoadSuperLogDataFromFile》. |

3.3.3 FK_GetSuperLogData

| | | |
|----|--|--|
| 形式 | long FK_GetSuperLogData(long nHandleIndex, long *pnSEnrollNumber, long *pnGEnrollNumber, long *pnManipulation, long *pnBackupNumber, DATE *pnDateTime) | |
| 功能 | 通过 LoadSuperLogData 或 USBLoadSuperLogDataFromFile 的指令读到 PC 内存的管理记录一个一个的获取. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.3 GetSuperLogData》. |

3.3.4 FK_EmptySuperLogData

| | | |
|----|--|--|
| 形式 | long FK_EmptySuperLogData(long nHandleIndex) | |
| 功能 | 从机器上删除所有的管理记录数据. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.4 EmptySuperLogData》. |

3.3.5 FK_LoadGeneralLogData

| | | |
|----|---|--|
| 形式 | long FK_LoadGeneralLogData(long nHandleIndex, long nReadMark) | |
| 功能 | 从机器将进出记录数据读到 PC 的内存上进行分析. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.5 LoadGeneralLogData》. |

3.3.6 FK_USBLoadGeneralLogDataFromFile

| | | |
|----|---|--|
| 形式 | long FK_USBLoadGeneralLogDataFromFile(long nHandleIndex, char * pstrFilePath) | |
|----|---|--|

| | | |
|----|--|--|
| 功能 | 机器的 USB 存储器的进出记录数据文件中将登记资料读到 PC 的内存进行分析。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.6 USBLoadGeneralLogDataFromFile》。 |

3.3.7 FK_GetGeneralLogData

| | | |
|----|---|--|
| 形式 | long FK_GetGeneralLogData(long nHandleIndex, long *pnEnrollNumber, long *pnVerifyMode, long *pnInOutMode, DATE *pnDateTime) | |
| 功能 | 通过 FK_LoadGeneralLogData 或 FK_USBLoadGeneralLogDataFromFile 指令读到 PC 内存的进出记录数据一个一个的获取。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.7 GetGeneralLogData》。 |

3.3.8 FK_EmptyGeneralLogData

| | | |
|----|--|--|
| 形式 | long FK_EmptyGeneralLogData(long nHandleIndex) | |
| 功能 | 删除所有的进出记录数据。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.8 EmptyGeneralLogData》。 |

3.3.9 FK_GetGeneralLogData_1

| | | |
|----|---|--|
| 形式 | FK_GetGeneralLogData_1(long nHandleIndex ,long *apnEnrollNumber,long pnVerifyMode,long *apnInOutMode, long *apnYear, long *apnMonth, long *apnDay, long *apnHour, long *apnMinute, long *apnSec) | |
| 功能 | 通过 FK_LoadGeneralLogData 或 FK_USBLoadGeneralLogDataFromFile 指令读到 PC 内存的进出记录数据一个一个的获取。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.79 GetGeneralLogData_1》。 |

3.3.10 FK_GetSuperLogData_1

| | | |
|----|---|--|
| 形式 | long FK_GetSuperLogData_1(long nHandleIndex ,long *apnSEnrollNumber, long *apnGEnrollNumber, long *apnManipulation, long *apnBackupNumber, long *apnYear, long *apnMonth, long *apnDay, long *apnHour, long *apnMinute, long *apnSec) | |
| 功能 | 通过 LoadSuperLogData 或 USBLoadSuperLogDataFromFile 的指令读到 PC 内存的管理记录一个一个的获取。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.310 GetSuperLogData_1》。 |

3.3.11 FK_GetRealTimeInfo

| | |
|----|---|
| 形式 | FK_GetRealTimeInfo(long* apGetRealTime) |
| 功能 | 将传输块所需等待时间和时间段信息读出到 PC 中。 |

| | | |
|----|---|--|
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.11 GetRealTimeInfo》。 |

3.3.12 FK_SetRealTimeInfo

| | | |
|----|---|--|
| 形式 | FK_SetRealTimeInfo(long* apSetRealTime) | |
| 功能 | 将传输块所需等待时间和时间段信息写入到指纹机中。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.3.12 SetRealTimeInfo》。 |

3.4 用户信息管理

3.4.1 FK_EnableUser

| | | |
|----|---|--|
| 形式 | long FK_EnableUser(long nHandleIndex, long nEnrollNumber, long nBackupNumber, long nEnableFlag) | |
| 功能 | 设置用户对机器的可用/不可用。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.4.1 EnableUser》。 |

3.4.2 FK_ModifyPrivilege

| | | |
|----|--|--|
| 形式 | long FK_ModifyPrivilege(long nHandleIndex, long nEnrollNumber, long nBackupNumber, long nMachinePrivilege) | |
| 功能 | 设置用户的机器操作权限。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.4.2 ModifyPrivilege》。 |

3.4.3 FK_GetUserName

| | | |
|----|---|--|
| 形式 | long FK_GetUserName(long nHandleIndex, long nEnrollNumber, LPSTR *pstrUserName) | |
| 功能 | 获取分配到用户的名称。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.4.3 GetUserName》。 |

3.4.4 FK_SetUserName

| | | |
|----|--|--|
| 形式 | long FK_SetUserName(long nHandleIndex, long nEnrollNumber, char *pstrUserName) | |
| 功能 | 给用户分配名称。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |

| | | |
|--|---|------------------------------|
| | 2 | 具体说明请参考 《2.4.4 SetUserName》。 |
|--|---|------------------------------|

3.4.5 FK_GetNewsMessage

| | | |
|----|--|--|
| 形式 | long FK_GetNewsMessage(long nHandleIndex, long nNewsId, LPSTR *pstrNews) | |
| 功能 | 从机器获取指定的通知文。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.4.5 GetNewsMessage》。 |

3.4.6 FK_SetNewsMessage

| | | |
|----|--|--|
| 形式 | long FK_SetNewsMessage(long nHandleIndex, long nNewsId, char * pstrNews) | |
| 功能 | 机器上设置通知文。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.4.6 SetNewsMessage》。 |

3.4.7 FK_GetUserNewsID

| | | |
|----|---|--|
| 形式 | long FK_GetUserNewsID(long nHandleIndex, long nEnrollNumber, long * pnNewsId) | |
| 功能 | 获取给用户分配的通知文的识别号码。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.4.7 GetUserNewsID》。 |

3.4.8 FK_SetUserNewsID

| | | |
|----|--|--|
| 形式 | long FK_SetUserNewsID(long nHandleIndex, long nEnrollNumber, long nNewsId) | |
| 功能 | 给用户分配通知文的识别号码。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.4.8 SetUserNewsID》。 |

3.5 机器管理

3.5.1 FK_EnableDevice

| | | |
|----|--|--|
| 形式 | long FK_EnableDevice(long nHandleIndex, unsigned char nEnableFlag) | |
| 功能 | 设置对机器的可用/不可用。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.1 EnableDevice》。 |

3.5.2 FK_PowerOnAllDevice

| | | |
|----|---|--|
| 形式 | void FK_PowerOnAllDevice(long nHandleIndex) | |
| 功能 | 启动连接的机器。 | |

| | | |
|----|---|--|
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.2 PowerOnAllDevice》。 |

3.5.3 FK_PowerOffDevice

| | | |
|----|---|--|
| 形式 | long FK_PowerOffDevice(long nHandleIndex) | |
| 功能 | 关闭机器的电源. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.3 PowerOffDevice》。 |

3.5.4 FK_GetDeviceTime

| | | |
|----|---|--|
| 形式 | long FK_GetDeviceTime(long nHandleIndex, DATE * pnDateTime) | |
| 功能 | 获取机器的日期, 时间. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.4 GetDeviceTime》。 |

3.5.5 FK_SetDeviceTime

| | | |
|----|--|--|
| 形式 | long FK_SetDeviceTime(long nHandleIndex, DATE nDateTime) | |
| 功能 | 设置机器的日期, 时间. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.5 SetDeviceTime》。 |

3.5.6 FK_GetDeviceStatus

| | | |
|----|--|--|
| 形式 | long FK_GetDeviceStatus(long nHandleIndex, long nStatusIndex, long *pnValue) | |
| 功能 | 获取现在机器上存在的状态值. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.6 GetDeviceStatus》。 |

3.5.7 FK_GetDeviceInfo

| | | |
|----|--|--|
| 形式 | long FK_GetDeviceInfo(long nHandleIndex, long nInfoIndex, long *pnValue) | |
| 功能 | 获取机器信息. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.7 GetDeviceInfo》。 |

3.5.8 FK_SetDeviceInfo

| | | |
|----|--|--|
| 形式 | long FK_SetDeviceInfo(long nHandleIndex, long nInfoIndex, long nValue) | |
| 功能 | 设置机器信息. | |

| | | |
|----|---|--|
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.8 SetDeviceInfo》。 |

3.5.9 FK_GetProductData

| | | |
|----|---|--|
| 形式 | long FK_GetProductData(long nHandleIndex, long nDataIndex, char *pstrValue) | |
| 功能 | 获取卖方注明的产品销售信息。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.9 GetProductData》。 |

3.5.10 FK_GetProductDataWithString

| | | |
|----|--|--|
| 形式 | long FK_GetProductDataWithString(long nHandleIndex, long nDataIndex, BSTR *apstrValue) | |
| 功能 | 与 FK_GetProductData 指令一样, 获取卖方注明的产品销售信息以文字列形式。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.9 GetProductData》。 |

3.5.11 FK_GetDeviceVersion

| | | |
|----|--|--|
| 形式 | long FK_GetDeviceVersion(long nHandleIndex, long *pnVersion) | |
| 功能 | 按每个型号获取每个型号上反应的更新版本。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.10 GetDeviceVersion》。 |

3.5.12 FK_GetDeviceTime_1

| | | |
|----|--|--|
| 形式 | long FK_GetDeviceTime_1(long nHandleIndex, long *apnYear, long *apnMonth, long *apnDay, long *apnHour, long *apnMinute, long apnSec, long *apnDayOfWeek) | |
| 功能 | 获取机器的日期, 时间。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.411 GetDeviceTime_1》。 |

3.5.13 FK_SetDeviceTime_1

| | | |
|----|--|--|
| 形式 | long SetDeviceTime_1(long nHandleIndex, long anYear, long anMonth, long anDay, long anHour, long anMinute, long anSec, long anDayOfWeek) | |
| 功能 | 设置机器的日期, 时间。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.5.512 SetDeviceTime_1》。 |

3.6 响铃管理

3.6.1 FK_GetBellTime

| | | |
|----|---|--|
| 形式 | long FK_GetBellTime(long nHandleIndex, long * pnBellCount, long * ptBellInfo) | |
| 功能 | 获取响铃设置信息. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.6.1 GetBellTime》. |

3.6.2 FK_GetBellTimeWithString

| | | |
|----|---|--|
| 形式 | long FK_GetBellTimeWithString(long nHandleIndex, long *pnBellCount, LPSTR *apstrBellInfo) | |
| 功能 | 与 FK_GetBellTime 指令一样, 响铃信息以文字列形式获取. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.6.2 GetBellTimeWithString》. |

3.6.3 FK_SetBellTime

| | | |
|----|--|--|
| 形式 | long FK_SetBellTime(long nHandleIndex, long nBellCount, long * ptBellInfo) | |
| 功能 | 机器上设置响铃信息. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.6.3 SetBellTime》. |

3.6.4 FK_SetBellTimeWithString

| | | |
|----|--|--|
| 形式 | long FK_SetBellTimeWithString(long nHandleIndex, long nBellCount, char *apstrBellInfo) | |
| 功能 | 与 SetBellTime 指令一样, 将响铃信息以文字列形式设置. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.6.4 SetBellTimeWithString》. |

3.7 门铃管理

该功能不是所有版本上都有的.

3.7.1 FK_GetDoorStatus

| | | |
|----|--|--|
| 形式 | long FK_GetDoorStatus(long nHandleIndex, long *apnStatusVal) | |
| 功能 | 获取开门状态. | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |

| | | |
|--|---|--------------------------------|
| | 2 | 具体说明请参考 《2.7.1 GetDoorStatus》。 |
|--|---|--------------------------------|

3.7.2 FK_SetDoorStatus

| | | |
|----|--|--|
| 形式 | long FK_SetDoorStatus(long nHandleIndex, long anStatusVal) | |
| 功能 | 操作开门状态。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.2 SetDoorStatus》。 |

3.7.3 FK_GetPassTime

| | | |
|----|---|--|
| 形式 | long FK_GetPassTime(long nHandleIndex, long anPassTimeID, long *apnPassTime, long anPassTimeSize) | |
| 功能 | 获取开门关门的时间段的信息(TimeZone)。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.3 GetPassTime》。 |

3.7.4 FK_GetPassTimeWithString

| | | |
|----|---|--|
| 形式 | long FK_GetPassTimeWithString(long nHandleIndex, long anPassTimeID, LPSTR *apstrPassTime) | |
| 功能 | 与 GetPassTime 一样，时间段信息以文字列形式返回。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.4 GetPassTimeWithString》。 |

3.7.5 FK_SetPassTime

| | | |
|----|---|--|
| 形式 | long FK_SetPassTime(long nHandleIndex, long anPassTimeID, long *apnPassTime, long anPassTimeSize) | |
| 功能 | 设置开门关门的时间段信息(TimeZone)。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.5 SetPassTime》。 |

3.7.6 FK_SetPassTimeWithString

| | | |
|----|--|--|
| 形式 | long FK_SetPassTimeWithString(long nHandleIndex, long anPassTimeID, char *apstrPassTime) | |
| 功能 | 与 FK_SetPassTime 一样，时间段信息以文字列形式输入。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.6 SetPassTimeWithString》。 |

3.7.7 FK_GetUserPassTime

| | | |
|----|---|--|
| 形式 | long FK_GetUserPassTime(long nHandleIndex, long anEnrollNumber, long *apnGroupID, long *apnPassTimeID, long anPassTimeIDSize) | |
|----|---|--|

| | | |
|----|-----------------------------|--|
| 功能 | 获取给用户分配的时间段信息组合与个别分配的时间段信息。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.7 GetUserPassTime》。 |

3.7.8 FK_GetUserPassTimeWithString

| | | |
|----|---|--|
| 形式 | long FK_GetUserPassTimeWithString(long nHandleIndex, long anEnrollNumber, long *apnGroupID, LPSTR *apstrPassTimeID) | |
| 功能 | 与 GetUserPassTime 指令一样, 将识别号码结构体以文字列形式返回。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.8 GetUserPassTimeWithString》。 |

3.7.9 FK_SetUserPassTime

| | | |
|----|---|--|
| 形式 | long FK_SetUserPassTime(long nHandleIndex, long anEnrollNumber, long anGroupID, long *apnPassTimeID, long anPassTimeIDSize) | |
| 功能 | 给用户设置时间段信息组合与给各个分配的时间段信息。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.9 SetUserPassTime》。 |

3.7.10 FK_SetUserPassTimeWithString

| | | |
|----|--|--|
| 形式 | long FK_SetUserPassTimeWithString(long nHandleIndex, long anEnrollNumber, long anGroupID, char *apstrPassTimeID) | |
| 功能 | 与 FK_SetUserPassTime 指令一样, 将识别号码结构体以文字列形式输入。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.10 SetUserPassTimeWithString》。 |

3.7.11 FK_GetGroupPassTime

| | | |
|----|---|--|
| 形式 | long FK_GetGroupPassTime(long nHandleIndex, long anGroupID, long *apnPassTimeID, long anPassTimeIDSize) | |
| 功能 | 获取时间段信息组合里的每个时间段信息的识别号码。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.11 GetGroupPassTime》。 |

3.7.12 FK_GetGroupPassTimeWithString

| | | |
|----|---|--|
| 形式 | long FK_GetGroupPassTimeWithString(long nHandleIndex, long anGroupID, LPSTR *apstrPassTimeID) | |
| 功能 | 与 FK_GetGroupPassTime 指令一样, 将识别号码结构体以文字列形式返回。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.12 GetGroupPassTimeWithString》。 |

3.7.13 FK_SetGroupPassTime

| | | |
|----|---|--|
| 形式 | long FK_SetGroupPassTime(long nHandleIndex, long anGroupID, long *apnPassTimeID, long anPassTimeIDSize) | |
| 功能 | 时间段信息组合上设置时间段信息的识别号码。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.13 SetGroupPassTime》。 |

3.7.14 FK_SetGroupPassTimeWithString

| | | |
|----|--|--|
| 形式 | long FK_SetGroupPassTimeWithString(long nHandleIndex, long anGroupID, char *apstrPassTimeID) | |
| 功能 | 与 FK_SetGroupPassTime 指令一样, 将识别号码结构体以文字列输入。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.14 SetGroupPassTimeWithString》。 |

3.7.15 FK_GetGroupMatch

| | | |
|----|--|--|
| 形式 | long FK_GetGroupMatch(long nHandleIndex, long *apnGroupMatch, long anGroupMatchSize) | |
| 功能 | 获取时间段信息结构体的组合的门操作组合 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.15 GetGroupMatch》。 |

3.7.16 FK_GetGroupMatchWithString

| | | |
|----|--|--|
| 形式 | long FK_GetGroupMatchWithString(long nHandleIndex, LPSTR *apstrGroupMatch) | |
| 功能 | 与 FK_GetGroupMatchTime 指令一样。将组合结构体以文字列形式返回。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.16 GetGroupMatchWithString》。 |

3.7.17 FK_SetGroupMatch

| | | |
|----|--|--|
| 形式 | long FK_SetGroupMatch(long nHandleIndex, long *apnGroupMatch, long anGroupMatchSize) | |
| 功能 | 设置时间段信息结构体的组合的门操作组合。 | |
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.17 SetGroupMatch》。 |

3.7.18 FK_SetGroupMatchWithString

| | | |
|----|---|--|
| 形式 | long FK_SetGroupMatchWithString(long nHandleIndex, char *apstrGroupMatch) | |
| 功能 | 与 FK_SetGroupMatch 指令一样, 将组合结构体以文字列登记。 | |

| | | |
|----|---|--|
| 其他 | 1 | “nHandleIndex”是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.7.18 SetGroupMatchWithString》。 |

3.8 夏天时间制管理

该功能不是所有版本上都有的.

3.8.1 FK_GetAdjustInfo

| | | |
|----|--|--|
| 形式 | long FK_GetAdjustInfo(long nHandleIndex, long* dwAdjustedState, long* dwAdjustedMonth, long* dwAdjustedDay, long* dwAdjustedHour, long* dwAdjustedMinute, long* dwRestoredState, long* dwRestoredMonth, long* dwRestoredDay, long* dwRestoredHour, long* dwRestoredMinute) | |
| 功能 | 获取夏天时间状态. | |
| 参数 | 状态值的参数指示者 | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | “dwAdjustedState, dwRestoredState”:如下 : None : 无变化. -1 : 早一个小时. +1 : 晚一个小时. |

3.8.2 FK_SetAdjustInfo

| | | |
|----|--|--|
| 形式 | long FK_SetAdjustInfo(long nHandleIndex, long dwAdjustedState, long dwAdjustedMonth, long dwAdjustedDay, long dwAdjustedHour, long dwAdjustedMinute, long dwRestoredState, long dwRestoredMonth, long dwRestoredDay, long dwRestoredHour, long dwRestoredMinute) | |
| 功能 | 操作夏天时间制状态. | |
| 参数 | 状态值(参考 4.1.6 结构体) | |
| 返回 | 执行成功返回 1, 失败返回对应的错误代码. 错误代码的具体说明参考《4.2 错误代码表》. | |
| 其他 | 1 | “dwAdjustedState, dwRestoredState”:如下 : None : 无变化. -1 : 早一个小时. +1 : 晚一个小时. |

3.9 机器信息管理

3.9.1 FK_GetServerNetInfo

| | | |
|----|---|---|
| 形式 | long FK_GetServerNetInfo(long nHandleIndex, BSTR* astrServerIPAddress, long* apServerPort, long* apServerRequest) | |
| 功能 | 读取从机器服务器的信息 | |
| 其他 | 1 | “nHandleIndex” 是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.9.1 GetServerNetInfo》. |

3.9.2 FK_SetServerNetInfo

| | | |
|----|---|---|
| 形式 | long FK_SetServerNetInfo(long nHandleIndex, LPCTSTR astrServerIPAddress, long anServerPort, long anServerRequest) | |
| 其他 | 1 | “nHandleIndex” 是 通过 FK_ConnectComm 或者 FK_ConnectNet 被返回的通讯接口识别号码。 |
| | 2 | 具体说明请参考 《2.9.2 SetServerNetInfo》. |

3.9.3 FK_SetUSBModel

| | | |
|----|--------------------------------------|---|
| 形式 | void FK_SetUSBModel(long anModel) | |
| 功能 | 指定写入（USB Flash）U 盘或从其读出的信息是对应于那款型号机器 | |
| 参数 | 1 | “anModel” 是表示型号信息的 long 型变量 |
| | | <div>#define FK625_FP1000 2001 #define FK625_FP2000 2002 #define FK625_FP3000 2003 #define FK625_FP5000 2004 #define FK625_FP10000 2005 #define FK625_FP30000 2006 #define FK625_ID30000 2007 #define FK635_FP700 3001 #define FK635_FP3000 3002 #define FK635_FP10000 3003 #define FK635_ID30000 3004 #define FK723_FP1000 4001 #define FK725_FP1000 5001 #define FK725_FP1500 5002 #define FK725_ID5000 5003 #define FK725_ID30000 5004 #define FK735_FP500 6001 #define FK735_FP3000 6002 #define FK735_ID30000 6003 #define FK925_FP3000 7001 #define FK935_FP3000 8001</div> |

4 备注

4.1 结构体

4.1.1 BELLINFO 结构体

```
#define MAX_BELLCOUNT_DAY 24

typedef struct tagBELLINFO {
    BYTE    mValid[MAX_BELLCOUNT_DAY];    // 响铃设置有效标志
    BYTE    mHour[MAX_BELLCOUNT_DAY];    // 响铃时间(时)
    BYTE    mMinute[MAX_BELLCOUNT_DAY];    // 响铃时间(分)
} BELLINFO; // 响铃 信息
```

4.1.2 PASSCTRLTIME 结构体

```
#define MAX_PASSTIMECOUNT 6            // Pass Count Max Value
#define MAX_TIMEGROUP 255              // Time Group Max Count
typedef struct tagDORRIME {
    BYTE    StartHour;    //开门时间(时)
    BYTE    StartMinute;    // 开门时间(分)
    BYTE    EndHour;    // 关门时间(时)
    BYTE    EndMinute;    // 关门时间(分)
} DOORTIME; // 时间段信息(TimeZone)

typedef struct tagPASSCTRLTIME {
    PASSTIME    mPassCtrlTime[MAX_PASSCTRL_COUNT];    // 每天的时间段信息
} PASSCTRLTIME; // 一周的时间段信息(TimeZone)
```

4.1.3 USERPASSINFO 结构体

```
#define MAX_USERPASSINFO_COUNT 3

typedef struct tagUSERPASSINFO {
    BYTE    UserPassID[MAX_USERPASSINFO_COUNT];    // 时间段信息识别号码
} USERPASSINFO; // 给用户设置的时间段信息识别号码
```

4.1.4 GROUPPASSINFO 结构体

```
#define MAX_GROUPPASSKIND_COUNT    5

#define MAX_GROUPPASSINFO_COUNT    3

typedef struct tagGROUPPASSINFO {
    BYTE    GroupPassID[MAX_GROUPPASSINFO_COUNT];    // 时间段信息识别号码
} GROUPPASSINFO; // 时间段信息组合
```

4.1.5 GROUPMATCHINFO 结构体

```
#define MAX_GROUPMATCHINFO_COUNT    10

typedef struct tagGroupMatchInfo {
    BYTE    GroupMatch[MAX_GROUPMATCHINFO_COUNT]; // 时间段信息组合的识别号码
} GROUPMATCHINFO; // 时间段信息组合的组合
```

4.1.6 ADJUSTINFO 结构体

```
typedef struct tagCHANGE_DATE {
    BYTE    Month;    // Month
    BYTE    Day;      // Day
    BYTE    Hour;     // Hour
    BYTE    Minute;   // Minute
} CHANGEDATE;

typedef struct tagADJUSTINFO {
    unsigned char    AdjustedState;    //夏令时状态
    unsigned char    Reserve1[1];      // Reserve
    unsigned short   AdjustedFlag;     //夏令时旗号
    CHANGEDATE       Adjusted;         //夏令时数据
    unsigned char    RestoredState;    //恢复正常的时间状态
    unsigned char    Reserve2[1];     // Reserve
    unsigned short   RestoredFlag;     //恢复正常的时间旗号
    CHANGEDATE       Restored;         //恢复正常的时间数据
} ADJUSTINFO;
```

4.1.7 REALTIMEINFO 结构体

```
#define MAX_REAL_TIME    4

typedef struct tagGroupMatchInfo {
    BYTE    Valid;    // 传送方式
    BYTE    AckTime;  // 回答时间
```

```

    BYTE  WaitTime; // 等待时间

    BYTE  Reserve; // reserve

    BYTE  SendPos; // 传送的位置

    BYTE  Hour[MAX_REAL_TIME]; // 时间段的时间

    BYTE  Minute[MAX_REAL_TIME]; // 时间段的分

} REALTIMEINFO; //是设置传输块所需等待时间和时间段的结构体

```

4.2 错误代码表

| 值 | 型号 | 解释 |
|-----|-----------------------|----------------------------|
| 1 | RUN_SUCCESS | 表示指令成功的通知 |
| 0 | RUNERR_NOSUPPORT | 表示机器不支持对应指令的错误代码 |
| -1 | RUNERR_UNKNOWNERROR | 表示不明错误代码 |
| -2 | RUNERR_NO_OPEN_COMM | 表示机器没有连接的错误代码 |
| -3 | RUNERR_WRITE_FAIL | 表示机器上不能传送数据的错误代码 |
| -4 | RUNERR_READ_FAIL | 表示从机器上不能读数据的错误代码 |
| -5 | RUNERR_INVALID_PARAM | 表示读的参数不正确的错误代码 |
| -6 | RUNERR_NON_CARRYOUT | 表示指令失败的错误代码 |
| -7 | RUNERR_DATAARRAY_END | 表示读完数据的通知 |
| -8 | RUNERR_DATAARRAY_NONE | 表示数据不存在的错误代码 |
| -9 | RUNERR_MEMORY | 表示 PC 的内存不足的错误代码 |
| -10 | RUNERR_MIS_PASSWORD | 表示连接机器许可号码是不正确的错误代码 |
| -11 | RUNERR_MEMORYOVER | 表示机器上不足存储器的错误代码 |
| -12 | RUNERR_DATADOUBLE | 表示要登记的号码已经在机器数据里存在的错误代码 |
| -14 | RUNERR_MANAGEROVER | 表示机器里没有空间，未能登记管理者. 数据的错误代码 |
| -15 | RUNERR_FPDATAVERSION | 表示要用的指纹数据的版本不正确的错误代码 |