U Series/H Series Attendance Machine Interface Documentation

Version <2.2>

Created: 2010-04-02

Revision History

date	Version	illustrate	author
2010-04-02	1.0	Create a file	Zou Chunqing
2011-04-22	2.2	Modify and add H series	Zou Chunqing
2011-04-25	2.3	Modify the definition of outgoing and returned data types	Zou Chunqing

Table of contents

1 Introduction 4

2Purpose 5 4

1 Introduction

This document uses the functions and interface details of the U series/H series attendance equipment.

2Purpose

Since there are many types of attendance machines, this document is created to facilitate future searches and comparisons of the differences between the functions and interfaces of various attendance machines.

3 Scope

1. Use U series fingerprint attendance equipment
2. Use H series fingerprint attendance equipment

4. Notes

1) FP_CLOCK.ocx , TMPCCOMM.dll , CH375DLL.DLL

Note: When running for the first time, you should register the OCX component. The registration command is : regsvr32 " the directory where the OCX file is located and its file name ". The registration command is: regsvr32 C:\FP_CLOCK .ocx

5 Details

Using U series/H series attendance machine interface documentation

(Note: All interface functions in this document use VC++ syntax rules)

Serial number	Details	
1	Interface Function	Connecting devices
	Functional Detailed Description	Open the specified port so that the computer can communicate with the device.
	Function declaration	boolean OpenCommPort (long dwMachineNumber);
	Parameter Description	dwMachineNumber: Terminal number
	Function return value	True (failed) False (success)
	Parameter return value	none
	Remark	(1) Open the port so that the device can communicate with the computer. (2) After calling this function and opening the port, you should call EnableDevice() to set the device to a non-attendance state. For function usage, refer to Article 2 . (3) Whenever this interface function is called to open a port and the corresponding operation is performed, the CloseCommPort() interface function must be called to close the port.
2	Interface Function	Whether to allow terminal attendance

	Functional Detailed Description	This interface function sets whether the terminal is allowed to perform attendance
	Function declaration	boolean **EnableDevice** (long dwMachineNumber, BOOL bFlag);
	Parameter Description	dwMachineNumber: terminal number ; bFlag: Whether attendance is allowed , 1 means allowed , 0 means not allowed.
	Function return value	True (failed) False (success)
	Parameter return value	none
	Remark	（1） When the device is not capable of attendance recording, it is in busy state and the user cannot perform attendance recording operations, but vice versa. Before performing any operation, the device should be set to a non-attendance state (bFlag=0), and after performing the corresponding operation, the device should be set to an attendance state (bFlag=1).
3	Interface Function	**Disconnect**
	Functional Detailed Description	Disable communication between terminal and computer
	Function declaration	void **CloseCommPort** ();
	Parameter Description	none
	Function return value	none
	Parameter return value	none

	Remark	（1） After completing the operation of the device, you need to close the port to disconnect the computer from the device. Before disconnecting the port , you should use the EnableDevice() interface function to set the device to attendance enabled state. For function usage, refer to Article 2 .
4	Interface Function	Get the current working status of the terminal
	Functional Detailed Description	This interface function obtains the current working status of the terminal
	Function declaration	boolean GetDeviceStatus (long dwMachineNumber, long dwStatus, long* dwValue);
	Parameter Description	dwMachineNumber: indicates the machine number of the terminal dwStatus: Indicates the type of status information to be obtained . The parameter value is as follows: Value Description 1Total number of administrators registered on the current terminal 2Total number of users currently registered on the terminal 3Total number of fingerprints registered on the current terminal 4Total number of passwords registered on the current terminal (total number of users registered for password verification) 5. The number of new management records on the current terminal 6. The number of new entry and exit records on the current terminal 7Total number of cards currently registered on the terminal dwValue: This variable receives the status information value
	Function return value	True (successful , the specified status information is obtained correctly) False (failed)
	Parameter return value	dwValue : Status information value
	Remark	
5	Interface Function	Get a person's registration information from memory

	Functional Detailed Description	Read back a person's registration information from the memory. Before using this interface function, you must first use the ReadAllUserID() function to read the person's registration information from the device to the internal memory.
	Function declaration	boolean GetAllUserID (long dwMachineNumber, long* dwEnrollNumber, long* dwEMachineNumber, long* dwBackupNumber, long* dwMachinePrivilege, long* dwEnable);

	Parameter Description	dwMachineNumber: The machine number of the terminal dwEnrollNumber: Enroll fingerprint number dwEMachineNumber: The registration machine number of the registration data to be obtained dwBackupNumber: backup fingerprint registration number **fingerprint backup registration number is described as follows:** **Value Description** 0 Fingerprint data No. 0 1 No. 1 fingerprint data 2 No. 2 fingerprint data 3 No. 3 fingerprint data 4 No. 4 fingerprint data 5 No. 5 fingerprint data 6 No. 6 fingerprint data 7 No. 7 fingerprint data 8 No. 8 fingerprint data 9 No. 9 fingerprint data 10 Password data 11Card Data 12All fingerprints, passwords, and card data 13All fingerprint data 20 Faces 21 Face 22 Face 23 Face 24 Face 25 Face 26 Face 27 Face dwMachinePrivilege : Permission dwEnable: Indicates whether the user is allowed to check attendance

	Function return value	True (success) : False (failed)
	Parameter return value	dwEnrollNumber: The returned fingerprint number dwEMachineNumber: The registration machine number of the obtained registration data dwBackupNumber: The returned backup fingerprint number dwMachinePrivilege : Returned user privileges dwEnable: User allowed attendance flag
	Remark	（1） Before calling this function, you should first call **ReadAllUserID()** to read all the registered personnel information in the terminal fingerprint device into the internal memory. For usage, please refer to Article 6. Then use this interface function to obtain a user's ID , authority, backup number, etc. from the memory , and call it repeatedly until the return value is False . dwBackupNumber (fingerprint backup registration number) Description: A user can register up to 10 fingerprints with the same registration number on a device. Each fingerprint corresponds to a fingerprint backup number (0 to 9). When the value is 10 to 13 , it represents password, card, fingerprint + password + card, and all fingerprint data respectively .
6	Interface Function	**Read all the registration information in the fingerprint machine into the internal memory**
	Functional Detailed Description	This function reads all fingerprint data from the terminal into memory , and then calls **GetAllUserID()** to retrieve fingerprint data from memory one by one.
	Function declaration	boolean **ReadAllUserID** (long dwMachineNumber);
	Parameter Description	dwMachineNumber: The machine number of the terminal
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	This interface function must be used in conjunction with **GetAllUserID()** .

7	Interface Function	Changing permissions
	Functional Detailed Description	This function changes the specified identification information permissions of the specified user on the specified terminal.
	Function declaration	boolean ModifyPrivilege (long dwMachineNumber, long dwEnrollNumber, long dwEMachineNumber, long dwBackupNumber, long dwMachinePrivilege);
	Parameter Description	dwMachineNumber: The machine number of the terminal dwEnrollNumber : User's fingerprint registration number dwEMachineNumber : indicates the user's registered fingerprint machine number dwBackupNumber : Backup registration number (please refer to Article 5 for parameter description)
		dwMachinePrivilege: The new user privilege to be set 0 : General user 1 : Super administrator (can register personnel and set device information) 2 : Registration administrator (can register registered personnel) 3 : Set up administrator (can set device information)
	Function return value	True (success) False (failed)
	Parameter return value	none

	Remark	(1) This function changes the specified identification information permissions of the specified user on the specified terminal to new permissions. (2) According to the value of dwBackupNumber , allow or disallow the specified user's registered fingerprint or all registered fingerprints and password data to be changed to dwMachinePrivilege value.
8	Interface Function	**Clear all fingerprint data**
	Functional Detailed Description	This function clears all fingerprint data
	Function declaration	boolean **ClearKeeperData** (long dwMachineNumber);
	Parameter Description	dwMachineNumber: The machine number of the terminal
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	This interface function will clear all fingerprint data on the device. Please be careful when using it.
9	Interface Function	**Change the date / time of the terminal**
	Functional Detailed Description	This function changes the date / time of the specified terminal.
	Function declaration	boolean **SetDeviceTime** (long dwMachineNumber);
	Parameter Description	dwMachineNumber: The machine number of the terminal
	Function return value	True (success) False (failed)

	Parameter return value	none
	Remark	Execute this interface function to change the device time to calculation time
10	Interface Function	**Get the terminal date / time**
	Functional Detailed Description	This function gets the date / time of the specified terminal.
	Function declaration	boolean GetDeviceTime (long dwMachineNumber, long* dwYear, long* dwMonth, long* dwDay, long* dwHour, long* dwMinute, long* dwDayOfWeek);
	Parameter Description	dwMachineNumber: The machine number of the terminal dwYear : year dwMonth : Month dwDay : Day dwHour : hour dwMinute : minute dwDayOfWeek : Day of the week (1 for Sunday, 2 for Monday , ... 7 for Saturday)
	Function return value	True (success) False (failed)

	Parameter return value	dwYear : The year returned from the machine dwMonth: Month returned from the machine dwDay: The day returned from the machine dwHour : The time returned from the machine dwMinute: Minute returned from the machine dwDayOfWeek : The day of the week returned from the machine
	Remark	none
11	Interface Function	Get bell time
	Functional Detailed Description	This function is used to obtain the bell time.
	Function declaration	boolean GetBellTime (long dwMachineNumber, long* dwValue, long* dwBellInfo);
	Parameter Description	dwMachineNumber: The machine number of the terminal dwValue: Number of rings (maximum value is 8 , minimum value is 1). dwBellInfo: bell time information (24 Bytes)
	Function return value	True (success) False (failure) , when the return value is False , the return parameter of the function has no meaning.
	Parameter return value	dwValue: The number of times the bell in the device rings dwBellInfo: Bell group time in the device
	Remark	This function only saves the data of the bell group in the computer memory . If you want to obtain the data , you must call RtlMoveMemory (Destination: Pointer; Source: Pointer; Length: integer) function retrieves data from memory. RtlMoveMemory() parameter description, Destination: variable storing data; Source: content to be copied; Length: number of bytes to be copied.
12	Interface Function	Set the bell
	Functional Detailed Description	This function is used to set one or more groups of doorbells.

	Function declaration	boolean SetBellTime (long dwMachineNumber, long dwValue, long* dwBellInfo);
	Parameter Description	dwMachineNumber: The machine number of the terminal dwValue: Number of rings (maximum value is 8 , minimum value is 1) dwBellInfo: bell time information (24 Bytes)
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	none
13	Interface Function	Turn off the device
	Functional Detailed Description	This function is used to close a terminal device.
	Function declaration	boolean PowerOffDevice (long dwMachineNumber);
	Parameter Description	dwMachineNumber: The machine number of the terminal
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	none
14	Interface Function	Read new attendance records to the internal memory
	Functional Detailed Description	This function reads all newly recorded in/out log data from the specified terminal and saves it in the internal memory. This function is used together with the function GetGeneralLogData () .

	Function declaration	boolean **ReadGeneralLogData** (long dwMachineNumber);
	Parameter Description	dwMachineNumber: The machine number of the terminal
	Function return value	True (success) False (failed, the terminal with the specified number does not exist or there is no new entry and exit record in the specified terminal, or a data communication error occurs)
	Parameter return value	none
	Remark	This function is used in conjunction with **GetGeneralLogData ()** and should be called before **GetGeneralLogData ()** .
15	Interface Function	Read a new attendance record from the internal memory
	Functional Detailed Description	Before calling this function, you should first call **ReadGeneralLogData ()** to get the input and output record data from the internal memory one by one. This function can only read new records and will not read the data that has been collected. It needs to be called repeatedly until it returns False
	Function declaration	boolean **GetGeneralLogData** (long dwMachineNumber, long* dwTMachineNumber, long* dwEnrollNumber, long* dwEMachineNumber, long* dwVerifyMode, long* dwYear, long* dwMonth, long* dwDay, long* dwHour, long* dwMinute);

	Parameter Description	dwMachineNumber: The machine number of the terminal dwTMachineNumber : This variable receives the terminal number value that the user passes through (to record attendance records) dwEnrollNumber : Pointer to a long variable that receives the enrollment number of the user who has been clocked in dwEMachineNumber : A pointer to a long variable that receives the registered machine number value of the attendance user. dwVerifyMode : A pointer to a long variable that receives the verification mode value of the attendance user. Parameter Description: 1: Fingerprint 2: Password 3: Induction card 4: Return 5: Go out 6: Open the door by door button 7: Open the door by software 8: Open the door for a long time (forced door opening) 9: Forced door closing 10: Identification successful but the door is not opened 11: Illegal door opening alarm 12: Go to work13: Get off work14: Work overtime15: Get off work after overtime dwYear : year dwMonth : month dwDay : day dwHour : hour dwMinute : minute
	Function return value	True (success) False (failed, the terminal with the specified number does not exist or there is no new entry and exit record in the specified terminal, or a data communication error occurs , in which case the return value of the function is meaningless)

	Parameter return value	dwTMachineNumber : The terminal number of the user (recording attendance records) dwEnrollNumber : The enrollment number value of the user who has been checked in dwEMachineNumber : The registered machine number value of the attendance user dwVerifyMode : The confirmation mode value of the user who has checked in dwYear : User attendance year dwMonth : User attendance month dwDay : User attendance day dwHour : User attendance time dwMinute : User attendance minutes
	Remark	Before using this function, use the **ReadGeneralLogData()** interface function to read the device data into the computer's memory, and then use this function to obtain data from the internal memory one by one . Each time valid data is successfully obtained, the function returns a value of TRUE, and returns FALSE when the data is read or an error occurs . If the value of ReadMark property is TRUE, after all the data stored in the internal memory is read out by this function, these data can no longer be read out by ReadGeneralLogData () function. If the ReadGeneralLogData () function is called next time, only the management record data newly recorded in the terminal can be read out. If the management record data in the internal memory cannot be read out for some reason and returns FALSE when the function is used to read the management record data in the internal memory one by one, the data will be read again the next time the ReadGeneralLogData () function is called. If the value of ReadMark property is FALSE, after the data stored in the internal memory is read out by this function, the record data can be read out by ReadGeneralLogData () function on the terminal . Before using this function, pre-allocate the register area for the BSTR type parameter value.
16	Interface Function	**Read all attendance records into internal memory**
	Functional Detailed Description	This function reads all the input and output record data from the specified terminal and saves it in the internal memory. This function must be used in conjunction with GetAllGLogData ().
	Function declaration	boolean **ReadAllGLogData** (long dwMachineNumber);

	Parameter Description	dwMachineNumber: The machine number of the terminal
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	This function only reads the recorded data into the computer's memory. After executing this function, you must call the GetAllGLogData() function to retrieve the data from the internal memory one by one.
17	Interface Function	Read an attendance record from the internal memory
	Functional Detailed Description	Before calling this function, you should call ReadAllGLogData () first. This function gets the input and output record data one by one from the internal memory. It needs to be used in conjunction with ReadAllGLogData (). It needs to be called repeatedly until it returns False.
	Function declaration	boolean GetAllGLogData (long dwMachineNumber, long* dwTMachineNumber, long* dwEnrollNumber, long* dwEMachineNumber, long* dwVerifyMode, long* dwYear, long* dwMonth, long* dwDay, long* dwHour, long* dwMinute);

	Parameter Description	dwMachineNumber: The machine number of the terminal dwTMachineNumber : This variable receives the terminal number value that the user passes through (to record attendance records) dwEnrollNumber : Pointer to a long variable that receives the enrollment number of the user who has been logged in dwEMachineNumber : A pointer to a long variable that receives the registered machine number value of the attendance user. dwVerifyMode : A pointer to a long variable that receives the verification mode value of the user who has logged in . For details, please refer to Article 15 . dwYear : year dwMonth : month dwDay : day dwHour : hour dwMinute : minute
	Function return value	True (success) False (failed, the terminal with the specified number does not exist or there is no new entry and exit record in the specified terminal, or a data communication error occurs)
	Parameter return value	dwTMachineNumber : The terminal number of the user (recording attendance records) dwEnrollNumber : The enrollment number value of the user who has been checked in dwEMachineNumber : The registered machine number value of the attendance user dwVerifyMode : The confirmation mode value of the user who has checked in dwYear : User attendance year dwMonth : User attendance month dwDay : User attendance day dwHour : User attendance time dwMinute : User attendance minutes

	Remark	（1） gets data from the internal memory one by one . Before using this function, you must first use the ReadAllGLogData() function to read all attendance records from the specified terminal and save them to the internal memory. （2） The usage of this function is the same as GetGeneralLogData () . For detailed usage , please refer to Article 15 . This function works independently of the value of the ReadMark property.
18	Interface Function	Get backup data
	Functional Detailed Description	This function is used for the terminal device serial number.
	Function declaration	long GetBackupNumber(long dwMachineNumber);
	Parameter Description	dwMachineNumber: The machine number of the terminal
	Function return value	Backing up your data
	Parameter return value	
	Remark	none
19	Interface Function	Delete a registration data
	Functional Detailed Description	This function is used to delete the registration data of the specified registration number on the specified terminal.

	Function declaration	boolean **DeleteEnrollData** (long dwMachineNumber, long dwEnrollNumber, long dwEMachineNumber, long dwBackupNumber);
	Parameter Description	dwMachineNumber: The machine number of the terminal dw EnrollNumber : indicates the registration number of the registration data to be deleted dwEMachineNumber : indicates the registration machine number of the registration data to be deleted dwBackupNumber : indicates the backup registration number of the registration data to be deleted . For the specific meaning, please refer to Article 5 .
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	none
20	Interface Function	**Register user data**
	Functional Detailed Description	This function is used to register user data.

	Function declaration	boolean SetEnrollData (long dwMachineNumber, long dwEnrollNumber, long dwEMachineNumber, long dwBackupNumber, long dwMachinePrivilege, VARIANT* dwEnrollData, long dwPassWord);
	Parameter Description	dwMachineNumber: The machine number of the terminal dw EnrollNumber : indicates the registration number of the registration data to be transmitted dwEMachineNumber : Indicates the registration machine number of the registration data to be transmitted dwBackupNumber : indicates the backup registration number of the registration data to be transmitted . Please refer to Article 5 for its specific meaning . dwMachinePrivilege : Indicates the permission of the registration data to be transmitted . For details, please refer to Article 7 . dwEnrollData : Fingerprint data, indicating a long pointer to the buffer that receives the enrollment data value to be obtained
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	none
tw	Interface Function	Get user registration data

en ty on e	Function declaration	boolean **GetEnrollData** (long dwMachineNumber, long dwEnrollNumber, long dwEMachineNumber, long dwBackupNumber, long* dwMachinePrivilege, VARIANT* dwEnrollData, long* dwPassWord);

	Parameter Description	dwMachineNumber: indicates the machine number of the terminal dw EnrollNumber : indicates the registration number of the registration data to be obtained dwEMachineNumber : indicates the registration machine number of the registration data to be obtained dwBackupNumber : indicates the fingerprint backup registration number to be obtained The **fingerprint backup registration number is described as follows:** **Value Description** 0 Fingerprint data No. 0 1 No. 1 fingerprint data 2 No. 2 fingerprint data 3 No. 3 fingerprint data 4 No. 4 fingerprint data 5 No. 5 fingerprint data 6 No. 6 fingerprint data 7 No. 7 fingerprint data 8 No. 8 fingerprint data 9 No. 9 fingerprint data 10 Password Data 11 Card Data 12 All fingerprints, passwords, and card data 13 All fingerprint data 20 Faces 21 Face 22 Face 23 Face 24 Face 25 Face 26 Face 27 Face dwMachinePrivilege : A long pointer to a variable that receives the machine privilege value of the registration data to be obtained . The parameters are as follows:

		Value Description 0 General users 1 Administrator (registration , machine settings) [Level 1] 2 Administrator (Registration) [Level 2] 3 Administrator (machine settings) [Level 3] dwEnrollData : Fingerprint data, indicating a long pointer to the buffer that receives the enrollment data value to be obtained dwPassWord : Password/card data, indicating a long pointer to a variable that receives the password value of the registration data to be obtained
	Function return value	True (success) False (failed, when the return value is FALSE, the parameter return value is meaningless)
	Parameter return value	dwMachinePrivilege : The machine privilege value of the obtained registration data dwEnrollData : Obtained enrollment data value dwPassWord : The password value of the obtained registration data
	Remark	(1) This function reads the specified fingerprint registration data and password data from the terminal. (2) When calling this function, if the value of dwBackupNumber is between 0 and 9 , the function will read the specified fingerprint registration data from the terminal. At this time, the variable value specified by dwPassWord is meaningless. If the value of dwBackupNumber is 10 when calling this function , the function will read the specified password registration data from the terminal. At this time, the variable value specified by dw EnrollData is meaningless.
	Function return value	True (success) False (failed, when the return value is FALSE, the parameter return value is meaningless)
tw en ty	Interface Function	Get new management record data
	Functional Detailed Description	This function is used to obtain the new management record data on the specified terminal and save it in the computer. It must be used with GetSuperLogData ()
	Function declaration	boolean ReadSuperLogData (long dwMachineNumber);

tw o	Parameter Description	dwMachineNumber : terminal machine number
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	ReadSuperLogData() function reads the newly recorded management record data from the specified terminal and saves it in the internal memory . This function only reads the new management records in the device, that is, it does not read the management record data that has been read. After using this function to read the management records into the internal memory, use the GetSuperLogData() function to obtain them one by one until the function return value is False .
tw en	Interface Function	Read a new management record from internal memory
	Functional Detailed Description	This function is used to obtain a new management data from the internal memory of the specified terminal. Before calling this function, you should first call ReadSuperLogData() to read the new management data into the internal memory . When reading data, you should call it repeatedly until it returns False.

ty th re e	Function declaration	boolean **GetSuperLogData** (long dwMachineNumber, long* dwTMachineNumber, long* dwSEnrollNumber, long* dwSMachineNumber, long* dwGEnrollNumber, long* dwGMachineNumber, long* dwManipulation, long* dwBackupNumber, long* dwYear, long* dwMonth, long* dwDay, long* dwHour, long* dwMinute);

	Parameter Description	dwMachineNumber : terminal machine number dwTMachineNumber : The terminal number that receives and records the data dwSEnrollNumber : This variable receives the registration number value of the administrator who performs the management operation ; when there is no registered administrator, the parameter value is 0 . dwSMachineNumber : This variable receives the registered machine number value of the administrator who performs the management operation dwGEnrollNumber : This variable receives the registration number value of the management operation object ; if the operation object is the terminal itself (i.e., the management operation to modify the terminal system information), the parameter value is 0. dwGMachineNumber : This variable receives the registered machine number value of the management operation object ; if the operation object is the terminal itself (i.e., the management operation to modify the terminal system information), the parameter value is 0. dwManipulation : This variable receives the type value of the management operation performed on the terminal The description of the management operation type values is as follows: **Value Description** 3 Registered a new user on the terminal. 4 Registered a new administrator on the terminal. 5The fingerprint registration data was deleted on the terminal. 6. The password registration data was deleted on the terminal. 7. The card registration data was deleted on the terminal . 8 Deleted all registration data on the terminal 9 Modified the system settings information on the terminal. 10 The date/time was modified in the terminal . 11 Modified the record setting information on the terminal. 12 The communication setting information was modified on the terminal.

		dwBackupNumber : This variable receives the fingerprint registration backup number of the registration data The backup registration number values are explained as follows: **Value Description** 0 Fingerprint data No. 0 1 No. 1 fingerprint data 2 No. 2 fingerprint data 3 No. 3 fingerprint data 4 No. 4 fingerprint data 5 No. 5 fingerprint data 6 No. 6 fingerprint data 7 No. 7 fingerprint data 8 No. 8 fingerprint data 9 No. 9 fingerprint data 10 Password Data 11 Card Data 12 All fingerprints, passwords, and card data 13 All fingerprint data 20 Faces 21 Face 22 Face 23 Face 24 Face 25 Face 26 Face 27 Face dwYear : year dwMonth : month dwDay : day dwHour : hour dwMinute : minute

	Function return value	True (success) False (failed)
	Parameter return value	dwSEnrollNumber : The registration number value of the administrator who performs the management operation dwSMachineNumber : The registered machine number value of the administrator who performs management operations dwGEnrollNumber : The registration number value of the management operation object dwGMachineNumber : The registered machine number value of the management operation object dwManipulation : The type of management operation performed on the terminal
		dwBackupNumber : Backup number of registered data dwYear : Return year dwMonth : Returned month dwDay : Returned day dwHour : Returned hour dwMinute : The minute returned
	Remark	gets data from the internal memory one by one . Before calling this function, you need to call the ReadSuperLogData() function to read the management records from the specified terminal and save them to the internal memory. Each time valid data is obtained, the function returns TRUE . After reading the data or when an error occurs, it returns FALSE. When using the ReadSuperLogData() and GetSuperLogData() interface functions, only new management records can be retrieved from the device. If the value of ReadMark property is TRUE, after all the data stored in the internal memory are read out by this function, these data can no longer be read out by ReadSuperLogData() function. If the ReadSuperLogData() function is called next time, only the management record data newly recorded in the terminal can be read out. If the management record data in the internal memory cannot be read out for some reason and returns FALSE when the function is used to read the management record data in the internal memory one by one, the data will be read again the next time the ReadSuperLogData() function is called . If the value of ReadMark property is FALSE, after the data stored in the internal memory are read out by this function, the record data can be read out by ReadSuperLogData () function on the terminal . Before using this function, pre-allocate the register area for the BSTR type parameter value.

twenty four	Interface Function	Read all management record data
	Functional Detailed Description	This function is used to read all management record data on the specified terminal and save it in the terminal's internal memory. GetAllSLogData () usage
	Function declaration	boolean ReadAllSLogData (long dwMachineNumber);
	Parameter Description	dwMachineNumber : terminal number
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	After using this interface function, you must call the GetAllSLogData() function to read back all management data.
25	Interface Function	Read a management record data from the internal memory
	Functional Detailed Description	This function is used to read a management record data from the internal memory. Before calling this function, you should call ReadAllSLogData () to read all management records into the internal memory of the fingerprint machine. This function needs to be called cyclically until it returns False.

	Function declaration	boolean **GetAllSLogData** (long dwMachineNumber, long* dwTMachineNumber, long* dwSEnrollNumber, long* dwSMachineNumber, long* dwGEnrollNumber, long* dwGMachineNumber, long* dwManipulation, long* dwBackupNumber, long* dwYear, long* dwMonth, long* dwDay, long* dwHour, long* dwMinute);

	Parameter Description	dwMachineNumber : terminal machine number dwTMachineNumber : This variable is used to receive the terminal number that records the data. dwSEnrollNumber : This variable is used to receive the registration number value of the administrator who performs the management operation dwSMachineNumber : This variable is used to receive the registered machine number value of the administrator who performs management operations dwGEnrollNumber : This variable is used to receive the registration number value of the management operation object dwGMachineNumber : This variable is used to receive the registration machine number value of the management operation object dwManipulation : This variable is used to receive the type value of the management operation performed on the terminal dwBackupNumber : This variable is used to receive the backup number of the registration data wYear : year dwMonth : month dwDay : day dwHour : hour dwMinute : minute
	Function return value	True (success) False (failed)
	Parameter return value	dwTMachineNumber : The terminal number that records the data dwSEnrollNumber : The registration number value of the administrator who performs the management operation dwSMachineNumber : The registered machine number value of the administrator who performs management operations dwGEnrollNumber : The registration number value of the management operation object dwGMachineNumber : The registered machine number value of the management operation object dwManipulation : The type of management operation performed on the terminal dwBackupNumber : Backup number of registered data

		dwYear : Return year dwMonth : Returned month dwDay : Returned day dwHour : Returned hour dwMinute : The minute returned
	Remark	one by one from the internal memory . The data is read from the specified terminal using the ReadAllSLogData() function and saved to the internal memory. **The usage of this function is the same as GetSuperLogData () ,** please refer to Article 23 **for details .** This function works independently of the value of the ReadMark property.
26	Interface Function	**Get the last error message**
	Functional Detailed Description	This function is used to read the last error message on the specified terminal.
	Function declaration	boolean **GetLastError** (long* dwErrorCode);
	Parameter Description	dwErrorCode : This variable receives the error code value **The parameters are described as follows:** **value illustrate** 0 Operation successful 1 Unable to open COM interface 2 An error occurred while sending data 3 An error occurred while receiving data. 4 Operation failed 5 All data in the memory have been read
	Function return value	True (success) False (failed)
	Parameter return value	dwErrorCode : error code

	Remark	The following is the function for reading record data: { GetSuperLogData () , GetAllSLogData () , GetGeneralLogData () , GetAllGLogData ()} If the return value of this function is FALSE , the function GetLastError is called . If the error code value is 5, it means that all the record data is read from the storage.
27	Interface Function	**Get Username**
	Functional Detailed Description	This function is used to obtain the corresponding user name of the specified registration number on the specified terminal.
	Function declaration	boolean **GetUserName** (long DeviceKind, long dwMachineNumber, long dwEnrollNumber, long dwEMachineNumber, VARIANT* lpszUserName);
	Parameter Description	DeviceKind : device type (temporary value is 0) dwMachineNumber : terminal number dwEnrollNumber : fingerprint number (i.e. registration number) dwEMachineNumber : indicates the registration machine number of the registration data to be obtained lpszUserName : indicates the user name to be obtained
	Function return value	True (success) False (failed)
	Parameter return value	dwEMachineNumber : The registration machine number of the registration data lpszUserName : obtained user name
	Remark	none
28	Interface Function	**Set Username**

	Functional Detailed Description	This function is used to obtain the user name on the specified terminal.
	Function declaration	boolean **SetUserName** (long DeviceKind, long dwMachineNumber, long dwEnrollNumber, long dwEMachineNumber, VARIANT* lpszUserName);
	Parameter Description	DeviceKind : device type (temporary value is 0) dwMachineNumber : terminal number dwEnrollNumber : Fingerprint number (i.e. user registration number) dwEMachineNumber : Registration machine number lpszUserName : Username
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	none
29	Interface Function	**Set the terminal IP address**
	Functional Detailed Description	This function is used to set the IP address of the terminal.
	Function declaration	boolean **SetIPAddress** (BSTR* lpszIPAddress, long dwPortNumber, long dwPassWord);

	Parameter Description	lpszIPAddress : IP address, such as: 192.168.10.10 dwPortNumber : port number (1-65535) dwPassWord : Password
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	When using TCPIP communication, you must first set the IP address of the terminal to be communicated.
30	Interface Function	**Get terminal settings information**
	Functional Detailed Description	This interface function is used to obtain the terminal's setting information
	Function declaration	boolean **GetDeviceInfo** (long dwMachineNumber, long dwInfo, long* dwValue);
	Parameter Description	dwMachineNumber : terminal number dwInfo : indicates the type of setting information to be obtained The parameter values are as follows: **value illustrate** 1The maximum number of managers that can be registered on the terminal. (The data range of this value is 0 ~ 10.) 2 The terminal number (the value range is 1 ~ 255.) 3Languages **Value Description** 0 English 1 SChinese (Simplified Chinese) 2 TChinese (Traditional Chinese) 3 Korean 4 Automatic shutdown time (the value range is 0 ~ 255. The unit is "minutes".)

		5 Lock control mode **Value** **Description** 0Start lock. 1Do not activate the lock. 6 The number of input and output records for which input and output record warnings are issued (the data range of this value is 0 ~ 1500) . 7 The number of management records for which management record warnings are issued (the value range is 0 ~ 255) . 8 Confirmation interval time (the value range is 0 ~ 255) . 9 Baud rate **Value** **Description** 0 1200 bps 1 2400 bps 2 4800 bps 3 9600 bps 4 19200 bps 5 38400 bps 6 57600 bps 7 115200 bps
	Function return value	True (success) False (failed)
	Parameter return value	dwValue : Returns the terminal's setting information
	Remark	none
31	Interface Function	Set company name

	Functional Detailed Description	This function is used to set the company name.
	Function declaration	boolean **SetCompanyName** (long dwMachineNumber, long vKind, VARIANT* dwCompanyName);
	Parameter Description	dwMachineNumber : terminal number vKind : vKind = 1 to set; vKind = 0 to delete dwCompanyName : Company name
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	This will display the company name on the device.
32	Interface Function	**Get company name**
	Functional Detailed Description	This function is used to obtain the company name.
	Function declaration	boolean **GetCompanyName** (long dwMachineNumber, VARIANT* dwCompanyName);
	Parameter Description	dwMachineNumber : terminal number dwCompanyName : Company name
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	This operation returns the company name displayed on the device.
33	Interface Function	**Get access control status**

	Functional Detailed Description	This function is used to obtain the access control status.
	Function declaration	boolean **GetDoorStatus** (long dwMachineNumber, long* dwValue);
	Parameter Description	dwMachineNumber : terminal number dwValue : Returns the access control status value 1 (Forced door opening) 2 (Forced to close) 3 (Software Open) 4 (Restore automatic control)
		5 Restart the fingerprint machine 6 Cancel alarm
	Function return value	True (success) False (failed)
	Parameter return value	dwValue : status value
	Remark	
34	Interface Function	**Set access control status**
	Functional Detailed Description	This function is used to set the access control status.
	Function declaration	boolean **SetDoorStatus** (long dwMachineNumber, long dwValue);

	Parameter Description	dwMachineNumber : terminal number dwValue : Status value 1 (Forced door opening) 2 (Forced closing) 3 (Software Open) 4 (Resume automatic control) 5 Restart the fingerprint machine 6 Cancel alarm
	Function return value	True (success) False (failed)
	Parameter return value	dwValue: Access control status return value
	Remark	
35	Interface Function	**Clear fingerprint data**
	Functional Detailed Description	This function is used to clear fingerprint data.
	Function declaration	boolean **EmptyEnrollData** (long dwMachineNumber);
	Parameter Description	dwMachineNumber : terminal number
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	Executing this function will clear all fingerprint data on the device, so be careful when operating.
36	Interface Function	**Clear the general operation log**

	Functional Detailed Description	This function is used to clear the normal operation log.
	Function declaration	boolean **EmptyGeneralLogData** (long dwMachineNumber);
	Parameter Description	dwMachineNumber : terminal number
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	This operation is to delete the common fingerprint attendance data.
37	Interface Function	**Clear the administrator operation log**
	Functional Detailed Description	This function is used to clear the administrator operation log.
	Function declaration	boolean EmptySuperLogData(long dwMachineNumber);
	Parameter Description	dwMachineNumber : terminal number
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	This operation clears the administrator's management records.
38	Interface Function	**U disk data file operation starts**

	Functional Detailed Description	This function is used to initialize some variables about the U disk file in the memory.
	Function declaration	void UsbEnrollDataStart();
	Parameter Description	none
	Function return value	none
	Parameter return value	none
	Remark	Before running all the U disk file import and export data, you must first run this function to initialize the relevant variables of the memory. Please refer to DEMO.
	Interface Function	**Obtain user registration data (U disk method)**
	Function declaration	boolean GetUsbEnrollData(long* dwEnrollNumber, long* dwBackupNumber, long* dwMachinePrivilege, VARIANT* dwEnrollData, long* dwPassWord, VARIANT* lpszUserName);

	Parameter Description	dw EnrollNumber : indicates the registration number of the registration data to be obtained dwBackupNumber : indicates the fingerprint backup registration number to be obtained The **fingerprint backup registration number is described as follows:** **Value Description** 0 Fingerprint data No. 0 1 No. 1 fingerprint data 2 No. 2 fingerprint data 3 No. 3 fingerprint data 4 No. 4 fingerprint data 5 No. 5 fingerprint data 6 No. 6 fingerprint data 7 No. 7 fingerprint data 8 No. 8 fingerprint data 9 No. 9 fingerprint data 10 Password Data 11 Card Data 12 All fingerprints, passwords, and card data 13 All fingerprint data 20 Faces 21 Face 22 Face 23 Face 24 Face 25 Face 26 Face 27 Face dwMachinePrivilege : A long pointer to a variable that receives the machine privilege value of the registration data to be obtained . The parameters are as follows: **Value Description** 0 General users

		1 Administrator (registration , machine settings) [Level 1] 2 Administrator (Registration) [Level 2] 3 Administrator (machine settings) [Level 3] dwEnrollData : Fingerprint data, indicating a long pointer to the buffer that receives the enrollment data value to be obtained dwPassWord : Password/card data, indicating a long pointer to a variable that receives the password value of the registration data to be obtained lpszUserName : User name, indicating a character pointer to a buffer that receives the name of the registrant to be obtained
	Function return value	True (success) False (failed, when the return value is FALSE, the parameter return value is meaningless)
	Parameter return value	dwMachinePrivilege : The machine privilege value of the obtained registration data dwEnrollData : Obtained enrollment data value dwPassWord : The password value of the obtained registration data
	Remark	(1) Before calling this function , you need to call EnrollDataReadFromFile to read the enrollment data into memory. (2) This function reads the specified fingerprint registration data and password data from the memory . (3) When calling this function, if the value of dwBackupNumber is between 0 and 9 , the function will read the specified fingerprint registration data from the terminal. At this time, the variable value specified by dwPassWord is meaningless. If the value of dwBackupNumber is 10 when calling this function , the function will read the specified password registration data from the terminal. At this time, the variable value specified by dw EnrollData is meaningless.
	Function return value	True (success) False (failed, when the return value is FALSE, the parameter return value is meaningless)
39	Interface Function	**Register user data (U disk method)**
	Functional Detailed Description	This function is used to create a USB disk file for registering user data.

	Function declaration	boolean SetUsbEnrollData (long dwEnrollNumber, long dwBackupNumber, long dwMachinePrivilege, VARIANT* dwEnrollData, long dwPassWord, VARIANT* lpszUserName) ;
	Parameter Description	dw EnrollNumber : indicates the registration number of the registration data to be transmitted dwBackupNumber : indicates the backup registration number of the registration data to be transmitted . Please refer to Article 5 for its specific meaning . dwMachinePrivilege : Indicates the permission of the registration data to be transmitted . For details, please refer to Article 7 . dwEnrollData : Fingerprint data, indicating the long pointer to the buffer where the enrollment data value is to be transmitted dwPassWord : Password value to send registration data lpszUserName : User name, indicating the character pointer to the buffer where the name of the registrant is to be sent
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	This function writes a registration data into the memory. When the loop finishes writing the registration data, you need to call EnrollDataSaveTOFile () to write it into the file.
40	Interface Function	Register data to create a USB file
	Functional Detailed Description	This function is used to write the data registered in the memory to the U disk file

	Function declaration	boolean EnrollDataSaveTOFile(LPCTSTR LPSZFileName);
	Parameter Description	LPSZFileName : The file name and path of the USB disk file to be created, in string format
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	
41	Interface Function	**Read the registration data of the USB file**
	Functional Detailed Description	This function is used to read the registration data of the U disk file into the memory
	Function declaration	boolean EnrollDataReadFromFile(LPCTSTR LPSZFileName);
	Parameter Description	LPSZFileName : The file name and path of the USB disk file to be read, in string format
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	
42	Interface Function	**Remove all administrators**
	Functional Detailed Description	This function is used to cancel all administrators on the terminal.
	Function declaration	boolean BenumbAllManager(long dwMachineNumber);

	Parameter Description	dwMachineNumber : device number
	Function return value	True (success) False (failed)
	Parameter return value	none
	Remark	It is used when the administrator leaves the company and the administrator's rights are not revoked normally. It uses violent means to revoke all administrator rights.
43	Interface Function	Setting up USB communication
	Functional Detailed Description	Set whether to use USB communication
	Function declaration	boolean IsUSB
	Parameter Description	IsUSB = True; The communication method is USB IsUSB = False; The communication method is not USB
	Function return value	
	Parameter return value	
	Remark	When IsUSB = True;, the communication mode between the interface and the attendance machine is USB, and there is no need to set other communication mode parameters such as communication serial port number, IP address, etc.
44	Interface Function	Set the communication serial port number
	Functional Detailed Description	Set the communication serial port number

	Function declaration	Integer variable CommPort
	Parameter Description	
	Function return value	
	Parameter return value	
	Remark	
45	Interface Function	Setting the baud rate
	Functional Detailed Description	Setting the baud rate
	Function declaration	Integer variable Baudrate
	Parameter Description	
	Function return value	
	Parameter return value	
	Remark	
46	Interface Function	ReadMark

	Functional Detailed Description	A Flag indicating whether the recorded data can be read again on the terminal after being read by the GetGeneralLogData () function and the GetSuperLogData () function. If this property value is TRUE, you cannot use the GetGeneralLogData () function and the GetSuperLogData () function to read the log data that has been read once. However, you can use the GetAllGLogData () function and the GetAllSLogData () function to read all the log data. If this property is FALSE, you can use the GetGeneralLogData () function and the GetSuperLogData () function to read the recorded data.
	Function declaration	Boolean variable ReadMark
	Parameter Description	
	Function return value	
	Parameter return value	
	Remark	

END