ZKPull SDK Development Guide

Version: 2.0, Rev. A.1

Date: 2021.10.25

Release history:

Date	Version	Change
2021-10-25	2.0, Rev. A.1	New release version

contact information:

Software Park Phase III, Jimei District, Xiamen city, Fujian Province

B02 20-22Zkteco Software

Web:www.zkteco.com

For more branch offices and office addresses, please visit www.zkteco.com

1.Overview of the PullSDK Interfaces

The PullSDK interfaces are a group of functions, which are used to access the data of the C3 and C4 access control panels.

PullSDK enables the developers of final application programs to aces the aces control panel more visually, conveniently, and concisely. The PullSDK interface provides the following functions:

Read and set the controller parameters

Read, set, and delete the related information (fr example, time segment, user information, and holiday information) of the controller

Search for and modify the device information

2.Description of the PullSDK Interface Technology

In the eyes of the developers of final application programs, the PullSDK interfaces are a group of extract interfaces that are used to set and get the data in the aces control panel. It seems that the developers are using the most universal SQL sentences while accessing the user data. In the eyes of the developers of application programs, the PullSDK interfaces seem to be a database server.

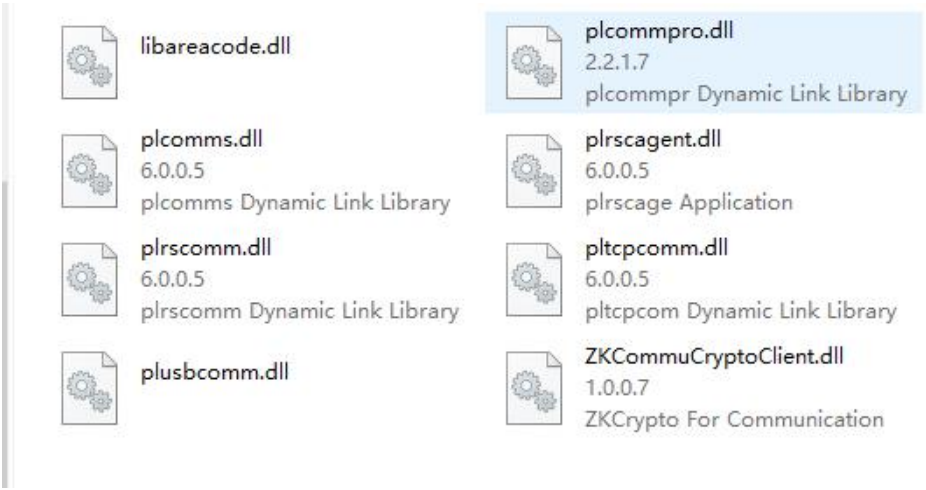
The PullSDK interfaces support the TCP/IP and RS485 communication protocol.

The PullSDK interfaces are developed by using the (language. Data communication is highly optimized, thus turning the PullSDK interfaces into the concise and efficient access interfaces.

Initially, the PullSDK interfaces are designed by referring to the SQL, but the most commonly used service model is the first consideration. Generally, the PullSDK interfaces are a group of elaborately abstracted interfaces, which attain a good balance between design, implementation, and use.

3. Installation of the PullSDK Interface

The PullSDK interface functions are contained in the plcommpro.dll file, which relies upon several other files. You need to copy the following five DLL files together to the system directory under Windows (windows / system32 under 32-bit operating system , windows / syswow64 under 64 bit operating system).After the file is copied, it does not need to be registered. It can be used directly according to the following interface functions.



(Note: Attached table 1 describes the functions of every file).

4. Detailed Description of the PullSDK Interface Functions

4.1 Connect

Long Connect(const char *Parameters)

Connect the device and return the connection handle after the connection is successful.

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
Parameters	Const char *	[in]	Specify connection option parameters

Return

Return Value Description：

Return value	Return Value Description
Greater than 0	connection handle
0	failed

Note

Parameters connection strings are case sensitive.

Use the Parameter Parameter to specify connection options, as in the following example：

"protocol=RS485,port=COM2,baudrate=38400bps,deviceid=1,timeout=50000,passwd=";

"protocol=TCP,ipaddress=192.168.12.154,port=4370,timeout=4000,passwd=";

need to pass device-specific connection parameters to this function to implement the connection function。

protocol Indicates the communication protocol, including RS485 and TCP；

port： device communication port。 Such as， Connected in RS485 mode, port can be set to COM1； The port for TCP communication, if not specially emphasized, is 4370 by default；

deviceid：RS485 communication address of equipment used for serial port；

baudrate：Baud rate used for serial communication；

ipaddress：TCP/IP Indicates the IP address of a device for communication；

timeout： connection timeout in milliseconds。 In case of poor network connection quality, increase the timeout value。generally，“timeout=5000”（5seconds）can meet the basic network use；When the - 2 error code often appears in the query data, increase the timeout value，You can set ：“timeout=20000”（20seconds）。

passwd： Set the connection password for communication. It can be empty to indicate that no password is used.

4.2 ConnectExt

Long ConnectExt(const char *Parameters, int *pErrorCode)

Connect the device and return the connection handle after successful connection， if the connection fails, an error code is returned by *pErrorCode.

Parameter

T he following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
Parameters	Const char *	[in]	Specify connection option parameters
pErrorCode	int *	[in/out]	Error code for connection failure

Return

Return Value Description：

Return value	Return Value Description
Greater than 0	connection handle
0	failed

Note

Parameters connection strings are case sensitive.

Use the Parameter Parameter to specify connection options, as in the following example：

"protocol=RS485,port=COM2,baudrate=38400bps,deviceid=1,timeout=50000,passwd='';

"protocol=TCP,ipaddress=192.168.12.154,port=4370,timeout=4000,passwd='';

need to pass device-specific connection parameters to this function to implement the connection function。

protocol Indicates the communication protocol, including RS485 and TCP；

port：device communication port。For example, if the port is connected in RS485 mode, set port to COM1. The TCP port is used for communication. If not specified, the default port is 4370；

deviceid：RS485 communication address of equipment used for serial port；

baudrate：Baud rate used for serial communication；

ipaddress：IP address of TCP / IP communication related equipment；

timeout：connection timeout in milliseconds。In case of poor network connection quality, increase the timeout value。generally，“timeout=5000”（5seconds）can meet the basic network use；When the - 2 error code often appears in the query data, increase the timeout value，You can set ：“timeout=20000”（20seconds）。

passwd：Set the connection password for communication. It can be empty to indicate that no password is used。

4.3 Disconnect

Void Disconnect(void *handle)

Disconnect from the device.

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
handle	void *	[in]	connection handle

Return

null

Note

4.4 SetDeviceParam

int SetDeviceParam(void *handle, const char *ItemValues)

Set controller parameters, such as equipment number, door magnetic type, lock driving time, card reading interval, etc.

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
handle	void *	[in]	connection handle

ItemValues	const char *	[in]	For the device parameter values to be set, multiple Parameter values can be separated by commas. At most 30 Parameter can be set at one time (see Table 2 for the settable parameter value attributes).

Return

Return Value Description：

Return value	Return Value Description
0	success
Less than 0	failed，see attached table 5 for information on error codes

Note

4.5 GetDeviceParam

int GetDeviceParam(void *handle, char *Buffer, int BufferSize, const char *Items)

Read the controller Parameter, such as equipment number, door magnetic Type, lock driving time, card reading interval, etc

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
handle	void *	[in]	connection handle
Buffer	char *	[in/out]	The buffer used to receive return data. The return data is in text format and may be multiple parameter values. Each parameter is separated by commas.
BufferSize	int	[in]	The size of the buffer used to receive Return data
Items	const char *	[in]	The device parameter names to be read are called the table. Multiple names can be separated by commas. At most 30 parameters can be read at a time (see Table 2 for the readable parameter value attribute).

Return

Return Value Description：

Return value	Return Value Description
0	success
Less than 0	failed，see attached table 5 for information on error codes

Note

4.6 ControlDevice

int ControlDevice(void *handle, LONG OperationID, LONG Param1, LONG Param2, LONG Param3, LONG Param4, const char *Options)

Control controller action。

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
handle	void *	[in]	connection handle
OperationID	LONG	[in]	Operation contents: 1 means lock output or auxiliary output, 2 means cancel alarm, 3 restart the equipment, 4 enable and disable normally open (here, the disabled normally open includes first open normally open, five consecutive card enabled normally open, and remotely enabled normally open).
Param1	LONG	[in]	When OperationID is output operation, if param2 is door output, this parameter represents the number of the door in the equipment; if param2 is auxiliary output, this parameter represents the number of the auxiliary output port in the equipment. For details, see attached table 3; When OperationID is cancel alarm, the default value is 0
Param2	LONG	[in]	When OperationID is

			output operation, this parameter indicates the address type of the equipment output point (1: lock output, 2: auxiliary output). See attached table 3 for details; When OperationID is cancel alarm, the default value is 0; When OperationID is 4, i.e. normally open, this parameter indicates whether to normally open or disable normally open (0: disabled; 1: enabled)
Param3	LONG		When OperationID is output operation, this parameter indicates the door opening time (0 means closed, 255 means normally open, the value range is 1 ~ 60 (seconds)), and the default value is 0
Param4	LONG		Reserved. The default value is 0
Options	const char *		It is empty by default and used for extension

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it indicates success
Less than 0	failed，see attached table 5 for information on error codes

Note

4.7 SetDeviceData

int SetDeviceData(void *handle,const char *TableName,const char *Data，const char *Options)

Setting data to the device is used to set data such as time period, user information and holiday settings. The data can be one record or multiple records. If the primary key of the inserted record is already in the device, the original record will be overwritten。

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
handle	void *	[in]	connection handle
TableName	const char *	[in]	Data table name. Please refer to attached Table IV for the currently available tables
Data	const char *	[in]	The data record indicates that the data is in text format. Multiple records are separated by \ r \ n, and each field = value pair is separated by \ t
Options	const char *	[in]	It is empty by default and used for extension

Return

Return Value Description：

Return value	Return Value Description
0	success
Less than 0	failed，see attached table 5 for information on error codes

Note

4.8 GetDeviceData

int GetDeviceData(void *handle, char *Buffer, int BufferSize, const char *TableName, const char *FieldNames,const char *Filter, const char *Options)

Reading data from the device is used to read card swiping records, time periods, user information, holiday settings and other data. The data can be one record or multiple records.

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
handle	void *	[in]	connection handle
Buffer	char *	[in]	The buffer used to receive return data. The return data is in text format and may be multiple records. Each record is separated by \r\n
BufferSize	int	[in]	The size of the buffer used to receive return data
TableName	const char *	[in]	Data table name. see attached Table IV for

			currently available table names
FieldNames	const char *	[in]	A list of field names. Multiple fields are separated by \t, and "*" indicates all fields. At this time, the first row of the return data field is the field name
Filter	const char *	[in]	Conditions for reading data. When a single string composed of "field name operator value", multiple conditions can be supported, separated by \ T, as follows: < field name > = < value > ("=" symbol cannot have spaces on both sides)
Options	const char *	[in]	Currently, it is only valid when downloading the data of access control event record table. When the value is "new record", download new records; when it is empty, Download all records. When downloading other table data, this field can be set to an empty string

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or a positive number, it indicates the success of the operation, and its value is the number of records
Less than 0	failed，see attached table 5 for information on error codes

Note

4.9 GetDeviceDataCount

int GetDeviceDataCount(void *Handle, const char *TableName, const char *Filter,const char *Options)

Read the total number of records in the device. Return specifies the number of records of the data.

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
handle	void *	[in]	connection handle
TableName	const char *	[in]	Data table name. see attached Table IV for currently available table names
Filter	const char *	[in]	It is empty by default and used for extension；
Options	const char *	[in]	It is empty by default and used for extension；

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or a positive number, it indicates the success of the operation, and its value is the number of records
Less than 0	failed，see attached table 5 for information on error codes

Note

4.10 DeleteDeviceData

int DeleteDeviceData(void *handle, const char *TableName,const char *Data,const char *Options)

Delete data in the device, such as user information, time period, etc.

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
handle	void *	[in]	connection handle
TableName	const char *	[in]	Data table name. see attached Table IV for currently available table names.
Data	const char *	[in]	For deleted conditions, the data record indicates that the data is in text format, and each "condition field = value" pair is separated by \t, * means delete all.
Options	const char *	[in]	It is empty by default and used for extension；

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it indicates success
Less than 0	failed， see attached table 5 for information on error codes

Note

4.11 GetRTLog

int GetRTLog(void *handle,char *Buffer, int BufferSize)

Obtain the real-time event records generated by the equipment, as well as the door status and alarm status of the equipment.

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
handle	void *	[in]	connection handle
Buffer	char *	[in]	The buffer used to receive return data. The return data is in text format. There are two types of data stored in the buffer, one is real-time event recording, and the other is door status / alarm status. The data returned with this function can only be one of them at a time.If it is a real-time event record, multiple event records can be returned at the same time (depending on the number of event records in the real-time monitoring buffer in the device at the current time).see attached table 7 for detailed data format in buffer。
BufferSize	int	[in]	The size of the buffer used to receive return data

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it is the number of records receiving data
Less than 0	failed， see attached table 5 for information on error codes

Note

4.12 GetRTLogExt

int GetRTLogExt(void *handle,char *Buffer, int BufferSize)

Obtain the real-time event records generated by the equipment, as well as the door status and alarm status of the equipment. Unlike GetRTLog, this function obtains a record in PUSH format。

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
handle	void *	[in]	connection handle
Buffer	char *	[in]	The buffer used to receive return data. The return data is in text format. There are two types of data stored in the buffer, one is real-time event recording, and the other is door status / alarm status.The data returned with this function can only be one of them at a time. If it is a real-time event record, multiple event records can be returned at the same time (depending on the number of event records in the real-time monitoring buffer in the device at the current time).see attached table 8 for detailed data format in buffer。
BufferSize	int	[in]	The size of the buffer used to receive return data

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it is the number of records receiving data
Less than 0	failed，see attached table 5 for information on error codes

Note

4.13 SearchDevice

int SearchDevice(char *CommType,char *Address, char *Buffer)

Search access controller in LAN.

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
CommType	char *	[in]	If the communication type is "UDP" (or Ethernet), the device with the specified communication type will be searched
Address	char *	[in]	Broadcast address: the LAN devices within the specified IP address range will be searched. The default is 255.255.255.255, that is, the whole network broadcast;
Buffer	char *	[in]	The buffer used to store the searched devices. The user shall determine the requested memory value according to the number of devices in the network. For example, it is recommended to apply for 32K memory within 50 devices and 64K memory within 100 devices.

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it is the number of access control controllers searched
Less than 0	failed，see attached table 5 for information on error codes

Note

Because this method searches the access control controller in the LAN by UDP broadcast, the UDP broadcast packet cannot pass through the router, so the controller and the server cannot be

separated by a router.
In addition, if the device and server are not in the same network segment, but Ping cannot find the controller IP address through this method, please try to adjust the controller IP address and server address to the same subnet (not necessarily the same network segment). For the network settings in the specific network, consult the corresponding network management to obtain the correct IP address, Subnet mask and gateway.

4.14 ModifyIPAddress

int ModifyIPAddress(char *CommType,char *Address, char *Buffer)
Modify the IP address of the controller through UDP broadcasting (considering the security of the device, only the IP address, subnet mask and gateway of the controller without password can be modified).
Parameter
The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
CommType	char *	[in]	Communication mode used when searching access controller。Here is "UDP" (or Ethernet)；
Address	char *	[in]	The broadcast address is 255.255.255.255 by default
Buffer	char *	[in]	Used to store the MAC address and new IP address of the target device; In addition to the IP address, please set the subnet mask and gateway according to the current network.

Return
Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it is the number of records receiving data
Less than 0	failed，see attached table 5 for information on error codes

Note
the current device has set the device communication password, in order to ensure the device security, the IP address of the device cannot be modified through this method. To modify the IP address, use SetDeviceParam to set the IP address, gateway and subnet mask.If you forget the communication password, you can use the dial switch to initialize the communication parameter, and the communication password will also be cancelled. The 7th bit of the dial switch is off by default; When you dial it up and down for three times within 10 seconds, and finally dial it back to

the off bit, restart the device to recover the IP address, gateway, subnet mask and communication password.

4.15 PullLastError

int PullLastError()
The function is used to Obtain the returned error code.If an error code return fails by using other error codes,this function can be called to obtain the error code.For example,if 0 is returned when an equipment connection fails by calling Connect(),you can run this function to obtain current error code.
Parameter description
None
Return
Return error ID number
Note

4.16 SetDeviceFileData

int SetDeviceFileData(void *Handle, const char *FileName, char *Buffer,int BufferSize,const char *Options)
Transfer files from PC to device。
Parameter
The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
Handle	void *	[in]	connection handle
FileName	const char *	[in]	The file name passed to the device, such as the emfw.cfg file
Buffer	char *	[in]	Data buffer for files to be transferred；
BufferSize	int	[in]	Length of transmitted data；
Options	const char *	[in]	It is empty by default and used for extension

Return
Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it indicates success
Less than 0	failed，see attached table 5 for information on error codes

Note

4.17 GetDeviceFileData

int GetDeviceFileData(void *Handle,char *Buffer,int *BufferSize,const char *FileName,const char *Options)

Get files from device to PC.

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
Handle	void *	[in]	connection handle
Buffer	char *	[in]	Buffer for receiving data
BufferSize	int *	[in]	Length of received data
FileName	const char *	[in]	The file name obtained from the device, such as the main file
Options	const char *	[in]	It is empty by default and used for extension

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it indicates success
Less than 0	failed，see attached table 5 for information on error codes

Note

4.18 ProcessBackupData

int ProcessBackupData(const unsigned char *revBuf,int fileLen,char *outBuf,int outSize)

Files used to process device backup, such as backup files in SD card, etc

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
revBuf	const unsigned char *	[in]	Is the content of the file passed in
fileLen	int	[in]	Is the file length
outBuf	char *	[in]	Is the data to receive the return
outSize	int	[in]	Is the maximum length of accepted data

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it indicates success
Less than 0	failed，see attached table 5 for information on error codes

Note

4.19 BufferToProtocolData

int BufferToProtocolData(char *ProStruct, char *Buffer, int Datalen);

Converts a byte stream to a formatted string。

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
ProStruct	char *	[in]	Table structure information
Buffer	char *	[in]	When calling a function, the buffer of the byte stream to be converted and the format string after the function call
Datalen	int	[in]	Data length of byte stream

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it indicates success. Return is the number of records in the format string
Less than 0	failed，see attached table 5 for information on error codes

Note

4.20 ProtocolDataToBuffer

int ProtocolDataToBuffer(char *Buffer, char *Datas, char *ProStruct, char *TableName);

Converts a formatted string to a byte stream。

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
Buffer	char *	[in]	Buffer of converted byte stream
Datas	char *	[in]	The content of the format string that needs to be converted

ProStruct	char *	[in]	Table structure information
TableName	char *	[in]	Table name

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it indicates success, and the length of the byte stream of return
Less than 0	failed，see attached table 5 for information on error codes

Note

4.21 SetParameters

int SetParameters(void *Handle, const char *Parameters)

Set SDK related Parameter

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
Handle	void *	[in]	connection handle
Parameters	const char *	[in]	The SDK Parameter needs to be set

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it indicates success
Less than 0	failed，see attached table 5 for information on error codes

Note

At present, only setting communication timeout is supported，and

”timeout=**”

4.22 GetPullSDKVersion

int GetPullSDKVersion(unsigned char *buffer, int size);

Get pull SDK version

Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
buffer	unsigned char *	[in]	Data cache
size	int	[in]	data size

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it indicates success. The length of the buffer of return
Less than 0	failed，see attached table 5 for information on error codes

Note

4.23 GetDeviceFileDataToPath

int GetDeviceFileDataToPath(void *Handle, const char *SrcFileName,const char *DecFileName,const char *Options)

Obtain the file from the device and save it to the specified directory, including the file name to be saved Parameter

The following table describes the parameters：

Parameter names	Type	Parameter properties	Parameter description
Handle	void *	[in]	connection handle
SrcFileName	const char *	[in]	The file name obtained from the device, such as the main file
DecFileName	const char *	[in]	File name to save, full path
Options	const char *	[in]	It is empty by default and used for extension

Return

Return Value Description：

Return value	Return Value Description
0	When return is 0 or positive, it indicates success
Less than 0	failed，see attached table 5 for information on error codes

Note

5. Appendix

5.1 Attached Table 1：Detailed Description of Interface Files

File Name	Description
plcommpro.dll	Dynamic connection database interface of the PullSDK function
plcomms.dll	Database on which the PullSDK interfaces rely
plrscomm.dll	Database on which the PullSDK interfaces rely
pltcpcomm.dll	Database on which the PullSDK interfaces rely
plrscagent.dll	Database on which the PullSDK interfaces rely

plusbcomm.dll	Database on which the PullSDK interfaces rely
ZKCommuCryptoClient.dll	Database on which the PullSDK interfaces rely
libareacode.dll	Database on which regional

5.2 Attached Table 2 : Description of Controller Parameters

Note：Due to the large number of controller versions, the parameters cannot be listed one by one.
If necessary, contact technical support for help

Attribute Name	Parameter	Read /write Type	Remarks
SerialNumber of device	~SerialNumber	Read only	
MAC address	MAC	Read only	
Number of doors	LockCount	Read only	the number of locks on the control board
Number of read heads	ReaderCount	Read only	Only the number of Wiegand read heads
Auxiliary input quantity	AuxInCount	Read only	
Auxiliary onput quantity	AuxOutCount	Read only	
Communication password	ComPwd	Read /write	Default:null character string.Maximum: 15-bit characters (including digits and letters).
IP address	IPAddress	Read /write	Default :192.168.1.201
Gateway	GATEIPAddress	Read /write	Default value is IPAddress
Baud rate	RS232BaudRate	Read /write	Default : 38400
Subnet mask	NetMask	Read /write	Default: 255.255.255.0
Anti-passback rule	AntiPassback	Read /write	One-door and two-way controller 1:Enable the anti-passback function between the readers of Door 1 Two-door and single-way controller 1 Enable the anti-passback function between the Door1 and Door 2 Two-door and two-way controller 1:Enable the anti-passback function between the readers of Door 1

			2:Enable the anti-passback function between the readers of Door 2 3:Enable the anti-passback function between the readers of Door1 and between the readers of Door2 respectively. 4:Enable the anti-passback function between Doo1 and Door2. Four-door and single-way controller (anti-passback between reading heads here only refers to anti submarine between inbio reading heads) 1 :Enable the anti-passback function between Door1 and Door2 2 :Enable the anti-passback function between Door3 and Door4 3 :Enable the anti-passback function between Door1 and Door2,and between Door3 and Door4 4:Enable the anti-passback function between Door1,2and Door3,4 5:Enable the anti-passback function between Door1 and Door2,3. 6:Enable the anti-passback function between Door1 and Door2,3,4 16:denotes that only supports anti-passback function between the readers of Door1 32:denotes that only supports anti-passback function between the readers of Door2 64:denotes that only supports anti-passback function between the readers of Door3 128 :denotes that only supports anti-passback function between the readers of Door4 Other options： 48 :denotes that Door1 and 2 support concurrent anti-passback among their respective readers. 80:denotes that Door1 and 2 support concurrent anti-passback among their respective readers.

			144 :denotes that Door1 and 4 support concurrent anti-passback among their respective readers. 96 :denotes that Door2 and 3 support concurrent anti-passback among their respective readers. 160 :denotes that Door2 and 4 support concurrent anti-passback among their respective readers. 196:denotes that Door3 and 4 support concurrent anti-passback among their respective readers. 112 :denotes that Door1,2 and 3 support concurrent anti-passback among their respective readers. 176 :denotes that Door1,2 and 4 support concurrent anti-passback among their respective readers. 208:denotes that Door1,3 and 4 support concurrent anti-passback among their respective readers. 224 :denotes that Door2,3 and 4 support concurrent anti-passback among their respective readers. 240 :denotes that Door1,2,3 and 4 support concurrent anti-passback among their respective readers. (Choose and configure the preceding options as required)
Interlock	InterLock	Read /write	Two-door controller 1 :InterLock Door 1 and Door 2 mutually Four-door controller 1 :InterLock Door 1 and Door 2 mutually 2 :InterLock Door 3 and Door 4 mutually 3 :InterLock Door 1,Door 2 and Door 3 mutually

			4:InterLock Door 1 and Door 2 mutually,and interLock Door 3 and Door 4 mutually 5:InterLock Door 1 ,Door 2,Door 3 and Door 4 mutually
Coercion code	Door1ForcePassWord Door2ForcePassWord Door3ForcePassWord Door4ForcePassWord	Read /write	Maximum 8 bits
Emergency password	Door1SupperPassWord Door2SupperPassWord Door3SupperPassWord Door4SupperPassWord	Read /write	Maximum 8 bits
Lock at door closing	Door1CloseAndLock Door2CloseAndLock Door3CloseAndLock Door4CloseAndLock	Read /write	1: Enabled 0 : Disabled
Door senor type	Door1SensorType Door2SensorType Door3SensorType Door4SensorType	Read /write	0:Not available 1: Normal open 2:Normal closed
Lock drive time length	Door1Drivertime Door2Drivertime Door3Drivertime Door4Drivertime	Read /write	The set range（0~255） 0 :Normal closed 255 : Normal open 1~254 :Door-opening duration Note：The ladder control has 120 doors
Timeout alarm duration of door magnet	Door1Detectortime Door2Detectortime Door3Detectortime Door4Detectortime	Read /write	The set range（0~255） Unit :second
Verify mode	Door1VerifyType Door2VerifyType Door3VerifyType Door4VerifyType	Read /write	1:Fingerprint 4 :Card 6:Card or fingerprint 10:Card and fingerprint 11: Card and Password
Multi-card open door Enable	Door1MultiCardOpenDoor Door2MultiCardOpenDoor Door3MultiCardOpenDoor Door4MultiCardOpenDoor	Read /write	0 :Disabled 1: Enabled

Opening the door through the first card	Door1FirstCardOpenDoor Door2FirstCardOpenDoor Door3FirstCardOpenDoor Door4FirstCardOpenDoor	Read /write	0: Disabled 1: First card normal open
Active time segment of the door (time segment in which a valid punch)	Door1ValidTZ Door2ValidTZ Door3ValidTZ Door4ValidTZ	Read /write	The default value of 0 indicates that the lock is not activated Note: the ladder control has 120 doors
Normal-open time segement of the door	Door1KeepOpenTimeZone Door2KeepOpenTimeZone Door3KeepOpenTimeZone Door4KeepOpenTimeZone	Read /write	Default 0(the parameter is not set) Note: the ladder control has 120 doors
Punch interval	Door1Intertime Door2Intertime Door3Intertime Door4Intertime	Read /write	0 means no interval (unit:second)
MCU watchdog	WatchDog	Read /write	0 : Disabled 1: Enabled
Synchronization controller time	DateTime	Write only	DateTime= ((Year-2000)*12*31 + (Month -1)*31 + (Day-1))*(24*60*60) + Hour* 60 *60 + Minute*60 + Second; For example, the time to be set is 2010-10-26 20:54:55 after conversion DateTime=347748895;Resolution method: If you get “DateTime = 347748895”; Then: Second = DateTime % 60； Minute = (DateTime / 60) % 60； Hour = (DateTime / 3600) % 24； Day = (DateTime / 86400) % 31 + 1； Month= (DateTime / 2678400) % 12 + 1； Year = (DateTime / 32140800) + 2000；
Four doors to two	Door4ToDoor2	Read /write	0: Disabled 1: Enabled
Cancel door normally open date	Door1CancelKeepOpenDay Door2CancelKeepOpenDay	Read only	Date saved when canceling normally open

	Door3CancelKeepOpenDay Door4CancelKeepOpenDay		Note: the ladder control has 120 doors
SD card backup time	BackupTime	Read /write	1~24，Set to integer
Display parameters of daylight saving time	~DSTF	Read /write	0:Never Show(default) 1:show
Enablement parameters of daylight saving time	DaylightSavingTimeOn	Read /write	0:Never start(default) 1:start
Enable mode of daylight saving time	DLSTMode	Read /write	0:mode 1 1:mode 2
Daylight saving time marker	CurTimeMode	Read /write	1 is currently daylight saving time 2 is not daylight saving time, and the firmware is used internally
Start time of daylight saving time mode	DaylightSavingTime	Read /write	The value have 4 bytes:"month-date-hour-minute"
Daylight saving time mode 1 end time	StandardTime	Read /write	The value have 4 bytes:"month-date-hour-minute"
Daylight saving time mode 2 start time: month	WeekOfMonth1	Read /write	The value range are 1 ~ 12
Start of daylight saving time mode 2: weeks	WeekOfMonth2	Read /write	The value range are 1 ~ 6
Daylight saving time mode 2 start: day of week	WeekOfMonth3	Read /write	The value range are 1~7
Daylight saving time mode 2 start :Hours	WeekOfMonth4	Read /write	The value range are 0~23
Start of daylight saving time mode 2: minutes	WeekOfMonth5	Read /write	The value range are 0~59
Daylight saving time mode 2 end time: month	WeekOfMonth6	Read /write	The value range are 1~12
End of daylight saving time mode 2: what week	WeekOfMonth7	Read /write	The value range are1~6

End of daylight saving time mode 2: day of week	WeekOfMonth8	Read /write	The value range are 1~7
Daylight saving time mode 2 ends:Hours	WeekOfMonth9	Read /write	The value range are 0~23
End of daylight saving time mode 2: minutes	WeekOfMonth10	Read /write	The value range are 0~59
Fingerprint comparison threshold	MThreshold	Read /write	The value range are 0~100

Unique parameter of ladder control：

Master configuration parameter	MachineType	Read /write	11 indicates a ladder controlled machine
Master configuration parameter	IsMasterBoard		1 Indicates that the machine is the master computer
If C4 is used as an expansion board, the expansion board needs to be configured	IsExtBoard		1 Indicates an expansion board
Activate host and expansion board	ExtBoard1ID		1 Indicates that the lock of the host is activated
	ExtBoard2ID		1 Indicates that the lock of expansion board 1 is activated, and the 485 address of corresponding expansion board 1 is 2, and so on
	..		
	ExtBoard12ID		1 Indicates that the lock of the expansion board 11 is activated
Configuration method of relay	RelayReverseFunOn		1 Indicates that the relay is on and the digital key is inactive. If the controller controls the door lock for other purposes, the parameter is configured as 0, indicating normal use.
Configure parameter for the number of locks on the expansion boardr	ExtLockCount		16 indicates 16 relay extension board. If C4 is used as an expansion board, this parameter may not be configured

Calculation method of controllable lock			Host lock + expansion board lock*ExtLockCount （such as 10 + 2*16 = 42）
Read head configuration			The 485 address of 485 read head must be configured as 1 before it can be used. The four Wiegand heads do not distinguish the number one. The access status in ladder control is none。

5.3 Attached Table 3：Description of ControlDevice Parameters

Operation ID	Description	Param1	Param2	Param3	Param4	Options
1	Output operation	Door number or auxiliary output number	1: Door output 2: auxiliary output（the address type of output operation）	0: disable 255: normal open state 1~60: normal open or the duration of normal open （If Param2=1, the value of Param3 makes sense）	reserved	Expansion parameter is null
2	Cancel alarm	0（null）	0（null）	0（null）	reserved	Expansion parameter is null
3	reboot device	0（null）	0（null）	0（null）	reserved	Expansion parameter is null
4	Enable/ disable normal	Door number	0：Disable 1：Enable	0（null）	reserved	Expansion parameter is null

	open state					

Note: If OperationID=1, Param2 determine the Param1 value is door number or auxiliary output number. If Param1 is door number, the max value is the door number that the device permitted. If the Param1 is auxiliary output number, the max value is the auxiliary output number that the device permitted.

5.4 Attached Table 4: Description of Structure of Function Tables

5.4.1 Description of ladder control meter structure

5.4.1.1 Card number information table structure (user)

Table name	user				
Field ID	Field name	width	Type	constraint	remarks
1	UID	2	int		User ID, firmware internal self increment field
2	CardNo	4	int		Card number
3	Pin	4	int	Primary key	Personnel number, nine digit code, can only be numbers
4	Password	8	string		password
5	Group	4	int		Multicard door opener group
6	StartTime	4	int		Validity start time YYYYMMDD， such as: 20100823；
7	EndTime	4	int		Expiration date YYYYMMDD， such as: 20100823；
8	Name	24	string		User name
9	SuperAuthorize	4	int		1 Superuser privileges

5.4.1.2 pin authorization table (userauthorize)

Table name	userauthorize				
Field ID	Field name	width	Type	constraint	remarks
1	Pin	4	int	Primary key	
2	AuthorizeTimezoneId	4	int	Primary key	timezoneID
3	AuthorizeDoorId	30	string	Primary key	AuthorizeDoorId indicates which floors of the equipment are

					included in this permission. The value is a string of 30 byte hexadecimal string. Every 2 hexadecimal characters are converted into binary, indicating 8 floor permissions. If the corresponding binary bit is 1, this floor belongs to this permission. For details, please refer to the following: 0F000000000000000000000000000000 Indicates that layers 1 ~ 4 have permission F000000000000000000000000000000 Indicates that layers 5 ~ 8 have permission … And so on (if the number of layers is insufficient, you don't need to fill in the following zeros, just reserve the front part)

5.4.1.3 Holiday table (holiday)

Table name	holiday				
Field ID	Field name	width	Type	constraint	remarks
1	Holiday	4	int	Primary key	20100101 means January 1, 2010
2	HolidayType	4	int		Holiday Type，only 1，2，3
3	Loop	4	int		1 means annual cycle, and the month and day can be equal. 2 means that the month, month and day must be equal

5.4.1.4 timezone table (timezone)

Table name	timezone				
Field ID	Field name	width	Type	constraint	remarks
1	TimezoneId	4	int	Primary key	Index number
2	SunTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
3	SunTime2	4	int		
4	SunTime3	4	int		
5	MonTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
6	MonTime2	4	int		
7	MonTime3	4	int		
8	TueTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
9	TueTime2	4	int		
10	TueTime3	4	int		
11	WedTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
12	WedTime2	4	int		
13	WedTime3	4	int		
14	ThuTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
15	ThuTime2	4	int		
16	ThuTime3	4	int		
17	FriTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
18	FriTime2	4	int		
19	FriTime3	4	int		
20	SatTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
21	SatTime2	4	int		
22	SatTime3	4	int		
23	Hol1Time1	4	int		Holiday 1 // For example, the value from 8:30 to 12:00 is (830 <

					< 16) + 1200, that is 0x33e04b0
24	Hol1Time2	4	int		Holiday 2 // For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
25	Hol1Time3	4	int		Holiday 3 // For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
26	Hol2Time1	4	int		
27	Hol2Time2	4	int		
28	Hol2Time3	4	int		
29	Hol3Time1	4	int		
30	Hol3Time2	4	int		
31	Hol3Time3	4	int		

5.4.1.5Event table (transaction)

Table name	transaction				
Field ID	Field name	width	Type	constraint	remarks
1	Cardno	4	Int		Card number
2	Pin	4	Int	Primary key	
3	Verified	1	char		Authentication method: 3 means "password only" authentication; 4 stands for "card only" verification; 11 represents "card plus password" authentication; 200 stands for "other"
4	DoorID	1	char		Door ID
5	EventType	1	char		Event type See attached table 6 for details
6	InOutState	1	char		Access status 0: out 1: in 2: other

7	Time_second	4	Int		Time stamp
					
					If you want to take it out for analysis, the analysis formula is as follows:
					second = t % 60;
					t /= 60;

					minute = t % 60;
					t /= 60;
					hour = t % 24;
					t /= 24;
					day = t % 31 + 1;
					t /= 31;
					month = t % 12 + 1;
					t /= 12;
					year = t + 2000;）

5.4.1.6 First card door opening table (firstcard)

Table name	firstcard				
Field ID	Field name	width	Type	constraint	remarks
1	Pin	4	int	Primary key	
2	DoorID	30	string	Primary key	It indicates which floors of the equipment are included in this permission. The value is a string of 30 byte hexadecimal string. Every 2 hexadecimal characters are converted into binary, indicating 8 floor permissions. If the corresponding binary bit is 1, this floor belongs to this permission. For details, please refer to the following: 0F000000000000000000000000000000 Indicates that layers 1 ~ 4 have permission F000000000000000000000000000000 Indicates that layers 5 ~ 8 have permission … and so on

3	TimezoneID	4	int	Primary key	

5.4.1.7 Multicard door opening combination table (multimcard)

Table name	multimcard				
Field ID	Field name	width	Type	constraint	remarks
1	Index	4	int	Primary key	
2	DoorId	4	Int		
3	Group1	4	int		Group number of multi card door opening
4	Group2	4	int		Group number of multi card door opening
5	Group3	4	int		Group number of multi card door opening
6	Group4	4	int		Group number of multi card door opening
7	Group5	4	int		Group number of multi card door opening

5.4.1.8 Linkage control i / o table (inoutfun)

Table name	inoutfun				
Field ID	Field name	width	Type	constraint	remarks
1	Index	2	int	Primary key	Trigger event index
2	EventType	1	char		All event types. When the event type is 220 (auxiliary input point off) and 221 (auxiliary input point short circuit), the input point is auxiliary input; When the event type is other than the above two types, the input point is the door:
3	InAddr	1	char		The input point InAddr is the door: 0 any; 1 door 1; 2 doors 2;

					3 door 3; 4 doors 4; The input point InAddr is an auxiliary input: 0 any; 1 auxiliary input 1; 2 auxiliary input 2; 3 auxiliary input 3; 4 auxiliary input 4;
4	OutType	1	char		Output type 0 refers to door lock and 1 refers to auxiliary output
5	OutAddr	1	char		OutAddr indicates lock: 1 door lock 1; 2 door lock 2; 3 door lock 3; 4 door lock 4; Whenthe output type ‘OutType’ is 1, the output point ‘OutAddr ’represents auxiliary output: 1 auxiliary output 1; 2 auxiliary output 2; 3 auxiliary output 3; 4 auxiliary output 4; 5 auxiliary output 5; 6auxiliary output 6

6	OutTime	1	char		Output action time: 0 off, 1 ~ 254 on, N seconds, 255 normally open
7	Reserved	1	char		Reserve to ensure alignment

5.4.1.9 9.0Fingerprint template table (template)

Table name	template				
Field ID	Field name	width	Type	constraint	remarks
1	Size	2	Int		Fingerprint template length
2	Pin	2	Int	Primary key	Personnel pin
3	FingerID	1	char		Finger number: if the fingerprint is an ordinary fingerprint, the value is 0 ~ 9; if it is a coercive fingerprint, the value is 16 ~ 25 (the finger number of the coercive fingerprint is the finger number of the normal fingerprint + 16);
4	Valid	1	char		0: invalid flag; 1: Valid signs; 3: Stress markers
5	Template	608	string		Fingerprint template

5.4.1.10 10.3Fingerprint template table (only on devices supporting fingerprint 10.0) (templatev10)

Table name	templatev10				
Field ID	Field name	width	Type	constraint	remarks
1	Size	2	int		Fingerprint template length
2	UID	2	Int		Firmware internal use
3	Pin	4	int	Primary key	Personnel pin
4	FingerID	1	char		Finger number: if the fingerprint is an ordinary fingerprint, the value is 0 ~ 9; if it is a coercive fingerprint, the value is 16 ~ 25 (the finger number of the coercive fingerprint is the finger number of

					the normal fingerprint + 16);
5	Valid	1	char		0: invalid flag; 1: Valid signs; 3: Stress markers
6	Template	2080	LONG_BYTE_T		Fingerprint template
7	Resverd	1	char		reservefield
8	EndTag	1	char		Firmware internal use

5.4.1.11 Loss report card (losscard)

Table name	losscard				
Field ID	Field name	width	Type	constraint	remarks
1	CardNo	4	int	Primary key	Card number
2	Reserved	4	int	Primary key	

5.4.2 Old schema table structure instructions

5.4.2.1 Card number information table structure (user)

Table name	user				
Field ID	Field name	width	Type	constraint	remarks
1	UID	2	int		User ID, firmware internal self incrementing field
2	CardNo	4	int		Card number. The maximum is 2147483647
3	Pin	4	int	Primary key	Personnel number, nine digit code, can only be numbers
4	Password	8	string		password
5	Group	4	int		Multi card door opener group
6	StartTime	4	int		Validity period start time YYYYMMDD, for example: 20100823;
7	EndTime	4	int		Expiration date: YYYYMMDD, for example: 20100823;
8	Name	24	string		User name
9	SuperAuthorize	4	int		

5.4.2.2 Pin authorization table (userauthorize)

Table name	userauthorize				
Field ID	Field name	width	Type	constraint	remarks
1	Pin	4	int	Primary key	
2	AuthorizeTimezoneId	4	int	Primary key	timezoneID
3	AuthorizeDoorId	4	int	Primary key	AuthorizeDoorId indicates which doors of the device are included in the permission. The value is obtained through binary coding. Each door is represented by a binary bit. If the bit is 1, the door belongs to the permission. For details, please refer to the following: 1 denotes lock1; 2 denotes lock2; 3 denotes lock1 and lock2; 4 denotes lock3; 5 denotes lock1 and lock3; 6 denotes lock2 and lock3; 7 denotes lock1, lock2 and lock3; 8 denotes lock4; 9 denotes lock1 and lock4; 10 denotes lock2 and lock4; 11 denotes lock1, lock2 and lock4; 12denotes lock3 and lock4; 13 denotes lock1, lock3 and

					lock4; 14 denotes lock2, lock3 and lock4; 15 denotes lock1, lock2, lock3 and lock4 15 it can be calculated that the numbers of the four doors are 1, 2, 3 and 4 respectively, then: 1<<(1-1)+1<<(2-1)+1<<(3-1)+1<<(4-1)=15 or（1111）$_2$=（15）$_{10}$

5.4.2.3 holiday table (holiday)

Table name	holiday				
Field ID	Field name	width	Type	constraint	remarks
1	Holiday	4	int	Primary key	20100101 means January 1, 2010
2	HolidayType	4	int		Holiday type, can only be 1, 2, 3
3	Loop	4	int		1 means annual cycle, and the month and day can be equal. 2 means that the month, month and day must be equal

5.4.2.4 timezone table (timezone)

Table name	timezone				
Field ID	Field name	width	Type	constraint	remarks
1	TimezoneId	4	int	Primary key	index
2	SunTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
3	SunTime2	4	int		
4	SunTime3	4	int		
5	MonTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0

6	MonTime2	4	int		
7	MonTime3	4	int		
8	TueTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
9	TueTime2	4	int		
10	TueTime3	4	int		
11	WedTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
12	WedTime2	4	int		
13	WedTime3	4	int		
14	ThuTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
15	ThuTime2	4	int		
16	ThuTime3	4	int		
17	FriTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
18	FriTime2	4	int		
19	FriTime3	4	int		
20	SatTime1	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
21	SatTime2	4	int		
22	SatTime3	4	int		
23	Hol1Time1	4	int		Holiday 1 // For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
24	Hol1Time2	4	int		Holiday 2 // For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
25	Hol1Time3	4	int		Holiday 3 // For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
26	Hol2Time1	4	int		
27	Hol2Time2	4	int		
28	Hol2Time3	4	int		
29	Hol3Time1	4	int		
30	Hol3Time2	4	int		
31	Hol3Time3	4	int		

5.4.2.5Event table (transaction)

Table name	transaction				
Field ID	Field name	width	Type	constraint	remarks
1	Cardno	4	Int		Card number
2	Pin	4	Int	Primary key	
3	Verified	1	char		Authentication method: 3 means "password only" authentication; 4 stands for "card only" verification; 11 represents "card plus password" authentication; 200 stands for "other" 1 stands for "fingerprint"
4	DoorID	1	char		Door ID
5	EventType	1	char		Event type See attached table VI for details
6	InOutState	1	char		

7	Time_second	4	Int		Time stamp
					
					If you want to take it out for analysis, the analysis formula is as follows:
					second = t % 60;
					t /= 60;
					minute = t % 60;
					t /= 60;
					hour = t % 24;
					t /= 24;
					day = t % 31 + 1;
					t /= 31;
					month = t % 12 + 1;
					t /= 12;
					year = t + 2000;)

5.4.2.6 First card door opening table (firstcard)

Table name	firstcard				
Field ID	Field name	width	Type	constraint	remarks
1	Pin	4	int	Primary key	
2	DoorID	30	string	Primary key	It indicates which floors of the equipment are included in this permission. The value is a string of 30 byte hexadecimal string. Every 2 hexadecimal characters are converted into binary, indicating 8 floor permissions. If the corresponding binary bit is 1, this floor belongs to this permission. For details, please refer to the following: 0F000000000000000000000000000000 Indicates that layers 1 ~ 4 have permission F000000000000000000000000000000 Indicates that layers 5 ~ 8 have permission … and so on
3	TimezoneID	4	int	Primary key	Max 20

5.4.2.7 Multicard door opening combination table (multimcard)

Table name	multimcard				
Field ID	Field name	width	Type	constraint	remarks
1	Index	4	int	Primary key	
2	DoorId	4	Int		

3	Group1	4	int		Group number of multi card door opening
4	Group2	4	int		Group number of multi card door opening
5	Group3	4	int		Group number of multi card door opening
6	Group4	4	int		Group number of multi card door opening
7	Group5	4	int		Group number of multi card door opening

5.4.2.8 Linkage control i/o table (inoutfun)

Table name	inoutfun				
Field ID	Field name	width	Type	constraint	remarks
1	Index	2	int	Primary key	Trigger event index
2	EventType	1	char		All event types. When the event type is 220 (auxiliary input point off) and 221 (auxiliary input point short circuit), the input point is auxiliary input; When the event type is other than the above two types, the input point is the door:
3	InAddr	1	char		The input point InAddr is the door: 0 any; 1 door 1; 2 doors 2; 3 door 3; 4 doors 4; The input point InAddr is an auxiliary input: 0 any; 1 auxiliary input 1;

					2 auxiliary input 2; 3 auxiliary input 3; 4 auxiliary input 4；
4	OutType	1	char		Output type 0 refers to door lock and 1 refers to auxiliary output
5	OutAddr	1	char		OutAddr indicates lock: 1 door lock 1; 2 door lock 2; 3 door lock 3; 4 door lock 4; When the output type 'OutType' is 1, the output point 'OutAddr' represents auxiliary output: 1 auxiliary output 1; 2 auxiliary output 2; 3 auxiliary output 3; 4 auxiliary output 4; 5 auxiliary output 5; 6auxiliary output 6；
6	OutTime	1	char		Output action time: 0 off, 1 ~ 254 on, n seconds, 255 normally open
7	Reserved	1	char		Reserve to ensure alignment

5.4.2.9 9.0 Fingerprint template table (template)

Table name	template				
Field ID	Field name	width	Type	constraint	remarks
1	Size	2	Int		Fingerprint template length
2	Pin	2	Int	Primary key	Personnel Pin

3	FingerID	1	char		Finger number: if the fingerprint is an ordinary fingerprint, the value is 0 ~ 9; if it is a coercive fingerprint, the value is 16 ~ 25 (the finger number of the coercive fingerprint is the finger number of the normal fingerprint + 16);
4	Valid	1	char		0: invalid flag; 1: Valid signs; 3: Stress markers
5	Template	608	string		Fingerprint template

5.4.2.10 10.3Fingerprint template table (only on devices supporting fingerprint 10.0) (templatev10)

Table name	templatev10				
Field ID	Field name	width	Type	constraint	remarks
1	Size	2	int		Fingerprint template length
2	UID	2	Int		Firmware internal use
3	Pin	4	int	Primary key	Personnel Pin
4	FingerID	1	char		Finger number: if the fingerprint is an ordinary fingerprint, the value is 0 ~ 9; if it is a coercive fingerprint, the value is 16 ~ 25 (the finger number of the coercive fingerprint is the finger number of the normal fingerprint + 16)；
5	Valid	1	char		0: invalid flag; 1: Valid signs; 3: Stress markers
6	Template	2080	LONG_BYTE_T		Fingerprint template
7	Resverd	1	char		reservefield
8	EndTag	1	char		Firmware internal use

5.4.3 New schema table structure instructions

5.4.3.1 Card number information table structure (user)

Table name	user				
Field ID	Field name	width	Type	constraint	remarks
1	UID	2	int		User ID, firmware internal self incrementing field
2	CardNo	4	int		Card Number
3	Pin	24	string	Primary key	Personnel number, nine digit code, can only be numbers
4	Password	8	string		password
5	Group	4	int		Multicard door opener group
6	StartTime	4	int		Validity period start time YYYYMMDD, for example: 20100823；
7	EndTime	4	int		Expiration date: YYYYMMDD, for example: 20100823;
8	Name	24	string		User name
9	SuperAuthorize	4	int		
10	Disable	1	char		Enable and Disable

5.4.3.2 Pin authorization table (userauthorize)

Table name	userauthorize				
Field ID	Field name	width	Type	constraint	remarks
1	Pin	24	string	Primary key	
2	AuthorizeTimezoneId	4	int	Primary key	timezoneID
3	AuthorizeDoorId	4	int	Primary key	AuthorizeDoorId indicates which doors of the device are included in the permission. The value is obtained through binary coding. Each door is represented by a binary bit. If the bit is 1, the door belongs to the permission. For details, please refer to the following: 1 denotes lock1;

					2 denotes lock2; 3 denotes lock1 and lock2; 4 denotes lock3; 5denotes lock1 and lock3; 6 denotes lock2 and lock3; 7 denotes lock1, lock2 and lock3; 8 denotes lock4; 9 denotes lock1 and lock4; 10 denotes lock2 and lock4; 11 denotes lock1, lock2 and lock4; 12 denotes lock3 and lock4; 13 denotes lock1, lock3 and lock4; 14 denotes lock2, lock3 and lock4; 15 denotes lock1, lock2, lock3 and lock4 15 it can be calculated that the numbers of the four doors are 1, 2, 3 and 4 respectively, then: 1<<(1-1)+1<<(2-1)+1<<(3-1)+1<<(4-1)=15 or $(1111)_2=(15)_{10}$

5.4.3.3 holiday table (holiday)

Table name	holiday				
Field ID	Field name	width	Type	constraint	remarks

1	Holiday	4	int	Primary key	20100101 means January 1, 2010
2	HolidayType	4	int		Holiday type, can only be 1, 2, 3
3	Loop	4	int		1 means annual cycle, and the month and day can be equal. 2 means that the month, month and day must be equal

5.4.3.4 timezone table (timezone)

Table name	timezone				
Field ID	Field name	width	Type	constraint	remarks
1	**TimezoneId**	4	int	Primary key	Index
2	**SunTime1**	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
3	**SunTime2**	4	int		
4	**SunTime3**	4	int		
5	**MonTime1**	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
6	**MonTime2**	4	int		
7	**MonTime3**	4	int		
8	**TueTime1**	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
9	**TueTime2**	4	int		
10	**TueTime3**	4	int		
11	**WedTime1**	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
12	**WedTime2**	4	int		
13	**WedTime3**	4	int		
14	**ThuTime1**	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
15	**ThuTime2**	4	int		
16	**ThuTime3**	4	int		
17	**FriTime1**	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
18	**FriTime2**	4	int		

19	**FriTime3**	4	int		
20	**SatTime1**	4	int		For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
21	**SatTime2**	4	int		
22	**SatTime3**	4	int		
23	**Hol1Time1**	4	int		Holiday 1 // For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
24	**Hol1Time2**	4	int		Holiday 2 // For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
25	**Hol1Time3**	4	int		Holiday 3// For example, the value from 8:30 to 12:00 is (830 < < 16) + 1200, that is 0x33e04b0
26	**Hol2Time1**	4	int		
27	**Hol2Time2**	4	int		
28	**Hol2Time3**	4	int		
29	**Hol3Time1**	4	int		
30	**Hol3Time2**	4	int		
31	**Hol3Time3**	4	int		

5.4.3.5 Event table (transaction)

Table name	transaction				
Field ID	Field name	width	Type	constraint	remarks
1	Pin	24	string	Primary key	
2	Verified	1	char		Authentication method: 3 means "password only" authentication; 4 stands for "card only" verification; 11 represents "card plus password" authentication; 200 stands for "other"
3	DoorID	1	char		Door ID
4	EventType	1	char		Event options 'EventType' See attached table VI for details
5	InOutState	1	char		

6	Time_second	4	int		Time stamp
					
					If you want to take it out for analysis, the analysis formula is as follows:
					second = t % 60;
					t /= 60;
					minute = t % 60;
					t /= 60;
					hour = t % 24;
					t /= 24;
					day = t % 31 + 1;
					t /= 31;
					month = t % 12 + 1;
					t /= 12;
					year = t + 2000;）

7	Index	4	Int		Configuration ID number of linkage
8	Cardno	8	Long long		Card number
9	Sitecode	4	int		

5.4.3.6 First card door opening table (firstcard)

Table name	firstcard				
Field ID	Field name	width	Type	constraint	remarks
1	Pin	24	string	Primary key	
2	DoorID	4	int	Primary key	
3	TimezoneID	4	int		

5.4.3.7 Multicard door opening combination table (multimcard)

Table name	multimcard				
Field ID	Field name	width	Type	constraint	remarks
1	Index	4	int	Primary key	
2	DoorId	4	Int		
3	Group1	4	int		Group number of multi card door opening
4	Group2	4	int		Group number of multi card door opening
5	Group3	4	int		Group number of multi card door

					opening
6	Group4	4	int		Group number of multi card door opening
7	Group5	4	int		Group number of multi card door opening

5.4.3.8 Linkage control i/o table (inoutfun)

Table name	inoutfun				
Field ID	Field name	width	Type	constraint	remarks
1	Index	4	int	Primary key	Trigger event index
2	EventType	1	char		All event types. When the event type is 220 (auxiliary input point off) and 221 (auxiliary input point short circuit), the input point is auxiliary input; When the event type is other than the above two types, the input point is the door:
3	InAddr	1	char		The input point InAddr is the door: 0 any; 1 door 1; 2 doors 2; 3 door 3; 4 doors 4; The input point InAddr is an auxiliary input: 0 any; 1 auxiliary input 1; 2 auxiliary input 2; 3 auxiliary input 3; 4 auxiliary input 4;
4	OutType	1	char		Output type 0 refers to door lock and 1 refers to auxiliary output
5	OutAddr	1	char		OutAddr indicates lock: 1 door lock 1; 2 door lock 2; 3 door lock 3; 4 door lock 4; When the output type 'OutType'

					is 1, the output point 'OutAddr' represents auxiliary output: 1 auxiliary output 1; 2 auxiliary output 2; 3 auxiliary output 3; 4 auxiliary output 4; 5 auxiliary output 5; 6auxiliary output 6
6	OutTime	1	char		Output action time: 0 off, 1 ~ 254 on, N seconds, 255 normally open
7	Reserved	1	char		Reserve to ensure alignment

5.4.3.9 9.0 Fingerprint template table (template)

Table name	template				
Field ID	Field name	width	Type	constraint	remarks
1	Size	2	Int		Fingerprint template length
2	Pin	24	string	Primary key	Personnel Pin
3	FingerID	1	char		Finger number: if the fingerprint is an ordinary fingerprint, the value is 0 ~ 9; if it is a coercive fingerprint, the value is 16 ~ 25 (the finger number of the coercive fingerprint is the finger number of the normal fingerprint + 16);
4	Valid	1	char		0: invalid flag; 1: Valid signs; 3: Stress markers
5	Template	608	string		Fingerprint template

5.4.3.10 10.3 Fingerprint template table (only on devices supporting fingerprint 10.0) (templatev10)

Table name	templatev10				
Field ID	Field name	width	Type	constraint	remarks
1	Size	2	int		Fingerprint template length
2	UID	2	Int		Firmware internal use
3	Pin	24	string	Primary	Personnel Pin

				key	
4	FingerID	1	char		Finger number: if the fingerprint is an ordinary fingerprint, the value is 0 ~ 9; if it is a coercive fingerprint, the value is 16 ~ 25 (the finger number of the coercive fingerprint is the finger number of the normal fingerprint + 16);
5	Valid	1	char		0: invalid markers; 1: Valid markers; 3: Duress markers
6	Template	2080	LONG_BYTE_T		Fingerprint template
7	Resverd	1	char		Reserve field
8	EndTag	1	char		Firmware internal use

5.4.3.11 Loss report card(losscard)

Table name	losscard				
Field ID	Field name	width	Type	constraint	remark
1	CardNo	4	int	Primary key	Card number
2	Reserved	4	int	Primary key	

5.4.3.12 Antisubmarine table (antipassback)

Table name	antipassback				
Field ID	Field name	width	Type	constraint	remark
1	Index	4	int	Primary key	
2	AddrID	4	int		
3	ConditionAddrID	4	int		

5.4.3.13Card number information table (cardinfo)

Table name	cardinfo				
Field ID	Field name	width	Type	constraint	remark
1	Pin	24	string	Primary	

				key	
2	CardNo	4	int		Card number

5.4.3.14 Extended user (extuser)

Table name	extuser				
Field ID	Field name	width	Type	constraint	remark
1	Pin	24	string	Primary key	
2	FunSwitch	4	int		

5.4.3.15 One person with more cards (mulcarduser)

Table name	mulcarduser				
Field ID	Field name	width	Type	constraint	remark
1	Pin	24	string		
2	CardNo	8	Long long	Primary key	Card number
3	SiteCode	2	int	Primary key	Location code

5.4.3.16 Auxiliary output setting table (outrelaysetting)

Table name	outrelaysetting				
Field ID	Field name	width	Type	constraint	remark
1	Num	1	char	Primary key	Output point
2	OutType	1	char		0 indicates door and 1 indicates auxiliary output
3	ActionType	1	char		Indicates 0 none, 2 normally closed, 1 normally open
4	TimezoneId	4	int		Is the id number of timezone.dat

5.4.3.17 Daylight saving time (time setting) table(DSTSetting)

Table name	DSTSetting				
Field ID	Field name	width	Type	constraint	remark
1	Year	4	int	Primary key	
2	StartTime	4	int		
3	EndTime	4	int		
4	Loop	4	int		

Note: please the case of the field in the Note table.

5.5 Attached Table5：Description of Error Codes in the Returned Values

（1）Error Code of PullSDK and Firmware By provided

Error code	Description
-1	The command is not sent successfully
-2	The command has no response
-3	The buffer is not enough
-4	The decompression fails
-5	The length of the read data is not correct
-6	The length of the decompressed data is not consistent with the expected length
-7	The command is repeated
-8	The connection is not authorized
-9	Data error: The CRC result is failure
-10	Data error: PullSDK cannot resolve the data
-11	Data parameter error
-12	The command is not executed correctly
-13	Command error: This command is not available
-14	The communication password is not correct
-15	Fail to write the file
-16	Fail to read the file
-17	The file does not exist
-18	Insufficient equipment space
-19	Checksum error
-20	The received data length is inconsistent with the given data length
-21	Platform parameter is not set in the device
-22	During firmware upgrade, the platform of the transmitted firmware is inconsistent with the local platform
-23	The upgraded firmware version is older than the firmware version in the device
-24	Error upgrading file ID
-25	During firmware upgrade, the file name passed is incorrect, that is, it is not emfw.cfg
-26	The length of the transmitted fingerprint template is 0
-27	The pin number of the fingerprint passed is wrong. The user cannot be found
-28	Execute door opening command in normally open period

-99	unknown error
-100	The table structure does not exist
-101	In the table structure, the **Condition** field does not exit
-102	The total number of fields is not consistent
-103	The sequence of fields is not consistent
-104	Real-time event data error
-105	Data errors occur during data resolution.
-106	Data overflow: The delivered data is more than 4 MB in length
-107	Fail to get the table structure
-108	Invalid options
-112	PC incoming data receive buffer insufficient
-201	Load Library failure
-202	Fail to invoke the interface
-203	Communication initialization fails
-206	The serial port agent fails to start because the serial port does not exist or the serial port is occupied
-301	Requested TCP/IP version error
-302	Incorrect version number
-303	Fail to get the protocol type
-304	Invalid socket
-305	Socket error
-306	Host error
-307	connection timed out

（2）Some common winsocket error codes

10035	Resources temporarily unavailable. This error is returned from operations on nonblocking sockets that cannot be completed immediately, for example, recv (Wsapiref_2i9e.asp) when no data is queued to be read from the socket. It is a non-fatal error, and the operation should be retried later. It is normal for WSAEWOULDBLOCK to be reported as the result from calling connect on a nonblocking SOCK_STREAM socket (Wsapiref_8m7m.asp), since some time must elapse for the connection to be established.
10038	An operation was attempted on something that is not a socket. Ether the socket handle parameter did not reference a valid socket, or for select, a member of an fd_set was no valid.

10054	Connection reset by peer. An existing connection was forcibly closed by the remote host. This normally results if the peer application on the remote host is suddenly stopped, the host is rebooted, the host or remote network interface is disabled, or the remote host uses a hard close (See setsockopt (Wsapiref_94aa.asp) for more information on the SO_LINGER option on the remote socket). This error may also result if a connection was broken due to keep-alive activity detecting a failure while one or more operations are in progress. Operations that were in progress fail with WSAENETRESET. Subsequent operations fail with WSAECONNRESET.
10060	Connection timed out. A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.
10061	Connection refused. No connection could be made because the target machine actively refused it. This usually results from trying to connect to a server that is inactive on the foreign host — that is, one with no server application running.
10065	No route to host. A socket operation was attempted to an unreachable host. See WSAENETUNREACH.

5.6 Attached Table 6: Description of Event Types and Code

Code	Event Types	Description
0	Normal Punch Open	In [Card Only] verification mode, the person has open door permission punch the card and triggers this normal event of open the door.
1	Punch during Normal Open Time Zone	At the normally open period (set to normally open period of a single door or the door open period after the first card normally open), or through the remote normal open operation, the person has open door permission punch the effective card at the opened door to trigger this normal events.
2	First Card Normal Open (Punch Card)	In [Card Only] verification mode, the person has first card normally open permission, punch card at the setting first card normally open period but the door is not opened, and trigger the normal event.
3	Multi-Card Open (Punching Card)	In [Card Only] verification mode, multi-card combination can be used to open the door. After the last piece of card verified, the system trigger this normal event.

4	Emergency Password Open	The password (also known as the super password) set for the current door can be used for door open. It will trigger this normal event after the emergency password verified.
5	Open during Normal Open Time Zone	If the current door is set a normally open period, the door will open automatically after the setting start time, and trigger this normal event.
6	Linkage Event Triggered	When the linkage setting the system takes effect, trigger this normal event.
7	Cancel Alarm	When the user cancel the alarm of the corresponding door, and the operation is success, trigger this normal event.
8	Remote Opening	When the user opens a door from remote and the operation is successful, it will trigger this normal event.
9	Remote Closing	When the user close a door from remote and the operation is successful, it will trigger this normal event.
10	Disable Intraday Normal Open Time Zone	When the door is in Normally Open (NO) state, swipe your valid card five times through the reader or call ControlDevice to disable the NO period on that day. In this case, trigger this normal event.
11	Enable Intraday Normal Open Time Zone	When the door's NO period is disabled, swipe your valid card (held by the same user) five times through the reader or call ControlDevice to enable the NO period on that day. In this case, trigger this normal event.
12	Open Auxiliary Output	If the output point address is set to a specific auxiliary output point and the action type is set enabled in a linkage setting record, then this normal event will be triggered as long as this linkage setting takes effect.
13	Close Auxiliary Output	Events that are triggered when you disable the auxiliary input through linkage operations or by calling ControlDevice.
14	Press Fingerprint Open	Normal events that are triggered after any person authorized to open the door presses his fingerprint and passes the verification in "Fingerprint only" or "Card/Fingerprint" verification modes.
15	Multi-Card Open (Press Fingerprint)	Multi-card open(Fingerprint required): normal events that are triggered when the last person opens the door with his fingerprint in "Finger print" verification mode.
16	Press Fingerprint during Normal Open Time Zone	Normal events that are triggered after any person authorized to open the door presses his valid fingerprint during the NO duration (including the NO durations set for single doors and the first-card NO duration) and through remote operations.

17	Card plus Fingerprint Open	Normal events that are triggered after any person authorized to open the door swipes his card and presses his fingerprint to pass the verification in the “Card + Fingerprint” verification mode.
18	First Card Normal Open (Press Fingerprint)	Normal events that are triggered after any person authorized to open the door becomes the first one to press his fingerprint and pass the verification during the preset first-card NO duration and in either the “Fingerprint only” or the “Card/Fingerprint” verification mode.
19	First Card Normal Open (Card plus Fingerprint)	Normal events that are triggered after any person authorized to open the door becomes the first one to swipe his card and press his fingerprint to pass the verification during the preset first-card NO duration and in the “Card + Fingerprint” verification mode.
20	Too Short Punch Interval	When the interval between two card punching is less than the interval preset for the door, trigger this abnormal event.
21	Door Inactive Time Zone (Punch Card)	In [Card Only] verification mode, the user has the door open permission, punch card but not at the door effective period of time, and trigger this abnormal event.
22	Illegal Time Zone	The user with the permission of opening the current door, punches the card during the invalid time zone, and triggers this abnormal event.
23	Access Denied	The registered card without the access permission of the current door, punch to open the door, triggers this abnormal event.
24	Anti-Passback	When the anti-pass back setting of the system takes effect, triggers this abnormal event.
25	Interlock	When the interlocking rules of the system take effect, trigger this abnormal event
26	Multi-Card Authentication (Punching Card)	Use multi-card combination to open the door, the card verification before the last one (whether verified or not), trigger this normal event
27	Unregistered Card	Refers to the current card is not registered in the system, trigger this abnormal event.
28	Opening Timeout:	The door sensor detect that it is expired the delay time after opened, if not close the door, trigger this abnormal event
29	Card Expired	The person with the door access permission, punch card to open the door after the effective time of the access control, can not be verified and will trigger this abnormal event.
30	Password Error	Use card plus password, duress password or emergency password to open the door, trigger this event if the password is wrong.

31	Too Short Fingerprint Pressing Interval	When the interval between two consecutive fingerprints is less than the interval preset for the door, trigger this abnormal event.
32	Multi-Card Authentication (Press Fingerprint)	In either the “Fingerprint only” or the “Card/Fingerprint” verification mode, when any person presses his fingerprint to open the door through the multi-card access mode and before the last verification, trigger this event regardless of whether the verification attempt succeeds.
33	Fingerprint Expired	When any person fails to pass the verification with his fingerprint at the end of the access control duration preset by himself, trigger this event.
34	Unregistered Fingerprint	Events that are triggered when any fingerprints are not registered in the system or registered but not synchronized to the device.
35	Door Inactive Time Zone (Press Fingerprint)	Abnormal events that are triggered when any person authorized to open the door presses his fingerprint during the preset valid duration.
36	Door Inactive Time Zone (Exit Button)	Abnormal events that are triggered when any person fails to open the door by pressing the Unlock button during the preset valid duration.
37	Failed to Close during Normal Open Time Zone	Abnormal events that are triggered when any person fails to close the door in NO state by calling **ControlDevice**.
38	Card has report the loss of	Event triggered by brushing the corresponding Card when the Card number is reported as lost
39	blacklist	When the user number is blacklisted, this event will occur when the user makes any comparison.
40	Multi fingerprint verification failed	During combined verification, multiple users perform fingerprint comparison. This event is generated when one user's fingerprint verification fails
41	Validation method error	This event occurs when the authentication method used by the user is inconsistent with the setting
42	Wiegand format error	The event is triggered when the number of digits of the card is inconsistent with the configuration
43	Background verification	
44	Background validation failed	
45	Background verification timeout	
46	Background authentication event	
47	Send command failed	It is defined in ladder control and used for the event of communication failed prompt when the master control sends a command to the sub control

48	Multi card door opening failed	Events prompted when combining validation failed
100	Tamper alarm	An event uploaded when the machine is dismantled
101	Duress Password Open	Use the duress password of current door verified and triggered alarm event.
102	Opened Accidentally	Except all the normal events (normal events such as user with door open permission to punch card and open the door, password open door, open the door at normally open period, remote door open, the linkage triggered door open), the door sensor detect the door is opened, that is the door is unexpectedly opened.
103	Duress Fingerprint Open	Use the duress fingerprint of current door verified and triggered alarm event.
200	Door Opened Correctly	When the door sensor detects that the door has been properly opened, triggering this normal event.
201	Door Closed Correctly	When the door sensor detects that the door has been properly closed, triggering this normal event.
202	Exit button Open	User press the exit button to open the door within the door valid time zone, and trigger this normal event.
203	Multi-Card Open (Card plus Fingerprint)	Normal events that are triggered when any person passes the verification with his card and fingerprint in multi-card access mode.
204	Normal Open Time Zone Over	After the setting normal open time zone, the door will close automatically. The normal open time zone include the normal open time zone in door setting and the selected normal open time zone in first card setting.
205	Remote Normal Opening	Normal events that are triggered when the door is set to the NO state for remote opening operations.
206	Device Start	When the device is being activated, this normal event is triggered.
207	Password open the door	The user uses the password to generate a door opening event
208	Super user open the door	Event generated when the user is a super user
209	The door is locked	When the door is configured as locked state, the door cannot be opened by using the door switch button and this event will be triggered.
210	Fire open	Fire protection function, use all doors are always open
211	Shutdown of superuser	Use event in ladder control. After the super user starts normally open, swipe the card again to close normally open state.
220	Auxiliary Input Disconnected	When any auxiliary input point breaks down, this normal event is triggered.
221	Auxiliary Input Shorted	When any auxiliary input point has short circuited, this normal event is triggered.

222	Background verification success	
223	Background verification network instability	
224	Enable antisubmarine in the background	
255	Actually that obtain door status and alarm status	See Attachment 7

5.7 Attached Table 7：Description of the data format returned by the parameter Buffer in the GetRTLog function

When the data in the buffer is resolved and detected to be:

- Multiple realtime event records: separate those records into single ones with “\r\n”.
- Door and alarm status recorded in single entries: separate those single records with a comma considering

that the data of single records is separated with a comma.

When you resolve single records, make adjustments according to bit 4 of the separated data. If bit 4 is 255, this record

contains the door status and alarm status only; otherwise, this record contains realtime event records.

The following table compares the data structures of these two records.

	Bit 0	Bit 1	Bit 2	Bit 3	Bit4	Bit 5	Bit6
Door/Alarm Status	Time	DSS status (0: no DSS; 1: door closed; 2: door open)	Alarm status (1: alarm; 2: door opening timeout)	Temporarily not in use	255	Temporarily not in use	200 (Indicates that the verification mode is “none”); not in use
Realtime Event Records	Time	Pin (Employee No.)	Card No.	Door No., namely lock number	Event type code. See Attachment 6 for details.	Entry/Exit status: (0: entry; 1: exit: 2: none)	The verification mode is the same as the door opening mode of controller

							parameters described in Attachment 2.

Note:

(1) The device can temporarily save a maximum of 30 realtime event records. You can call **GetRTLog** to check whether the cache contains event records. If so, the device returns all records (30 entries at most) in the current cache; otherwise, the device returns the door and alarm status events referred above.

(2) The door status records contain the open/closed status of current door (on the premise that the DSS is connected). Additionally, you can judge the current door status through “Door already open” (Event code: 200) and “Door already closed” (Event code: 201).

(3) When the record adopts the door/alarm status, the door status contained in all records actually is the door status (four doors at most) of all doors of the device. 4 bytes are respectively represents four door status, arranged in an ascending order separately represent doors 1 to 4. For example, if this value is 0x01020001, door No.1 is closed, door No.2 is not configured with the DSS, door No.3 is door opened, and door No.4 is door closed. Contained in the alarm status (and Opening Timeout) (The Second place) the same that 4 bytes are respectively represents four door status, behind two place of Each byte respectively represents whether that have alarm or door open is overtime, arranged in an ascending order separately represent alarm or door opening timeout. For example, if this value is 0x01020001, door No.1 is closed, door No.3 means door opening timeout, door No.2 and No.4 means alarm.

(4)When the record adopts “realtime event” status and type of event is Triggered Linkage Event (the code of type event: 6), the sixth place saved Linkage Event Triggered, and the second is for reuse of Linkage ID, It have software for the device synchronous linkage setting (usually the linkage in the ID value of software end database).

(5)The values of the three types in the event point number can be corresponding to the silkscreen on the access control panel. For example, door number 1 corresponds to Lock1 on the control panel, auxiliary input 1 corresponds to Aux In1 on the control panel, and auxiliary output 2 corresponds to Aux Out2 on the control panel (Note: The screen printing contents may be different depending on the device model, but the numbers correspond.

5.8 Attached Table 8：Description of data format returned by parameter Buffer in GetRTLogExt function

The data in Buffer is a string record in PUSH format. If the record is a real-time event record, it contains multiple single records separated by '\r\n'. If the data Type obtained is the door state and alarm state, the contained record is already a single record.

A single record is divided into two string formats of real-time event record and door state

alarm state, and each format is divided into multiple fields segmented by '\ t'.The two string formats for a single record are as follows:

	String format
Door/Alarm Status	type=rtstate\ttime=%s\tsensor=%02X\trelay=%02X\talarm=%02X%02X%02X%02X%\r\n Description： type=rtstate\ttime=xx\tsensor=AA\trelay=CC\talarm=DDEEFFGGHHIIJJKK\r\n 'time' indicates the current time in the format of %04d-%02d-%02d %02d:%02d:%02d; 'sensor' represents the status of door status, each door occupies two binary bits, AA represents 1-4 doors, and 1 occupies the first and second bits of the first byte, and so on. 0b00 indicates that the Type of the current door status is set to no door status, 0b01 indicates that the current door is closed (with door status), and 0b10 is open (without door status). 'relay' represents the relay state, each door occupies a binary bit, 0b0 represents the relay closing, 0b1 represents the relay disconnection, the first digit is the relay state of 1 door, and so on; (Both are currently 0) 'Alarm' indicates the alarm status. Each door occupies one byte and can express up to 8 alarms. DD is the alarm status of one door, and so on. At present, the alarm is defined as follows: No. 1: Unexpected door opening event No. 2: anti-demolition alarm No. 3: Duress password alarm No. 4: Duress fingerprint alarm No. 5: door magnetic timeout alarm No. 6:-No. 8:： reserved field
Real time event recording	type=rtlog\ttime=%s\tpin=%u\tcardno=%u\teventaddr=%d\tevent=%d\tinoutstatus=%d\tverifytype=%d\r\n Description： 'time' indicates the current time in the format of %04d-%02d-%02d %02d:%02d:%02d; 'pin' indicates a person number; 'cardno' indicates card number; 'Eventaddr' indicates the event point number,Including door number (i.e. lock

	number), auxiliary input number, auxiliary output number; 'event' event Type code, see **Attached Table 6;** 'inoutstatus' indicates in and out status (0 is in, 1 is out, and 2 is none); 'verifytype' indicates the verification mode, which is the same as the opening mode in the description of controller parameters in Attached Table 2.

Note:

(1) The device can temporarily save a maximum of 30 realtime event records. You can call **GetRTLog** to check whether the cache contains event records. If so, the device returns all records (30 entries at most) in the current cache; otherwise, the device returns the door and alarm status events referred above.

(2) The door status records contain the open/closed status of current door (on the premise that the DSS is connected). Additionally, you can judge the current door status through "Door already open" (Event code: 200) and "Door already closed" (Event code: 201).

(3)When the record adopts "realtime event" status and type of event is Triggered Linkage Event (the code of type event: 6), the sixth place saved Linkage Event Triggered, and the second is for reuse of Linkage ID, It have software for the device synchronous linkage setting (usually the linkage in the ID value of software end database).

(4)The values of the three types in the event point number can be corresponding to the silkscreen on the access control panel. For example, door number 1 corresponds to Lock1 on the control panel, auxiliary input 1 corresponds to Aux In1 on the control panel, and auxiliary output 2 corresponds to Aux Out2 on the control panel (Note: The screen printing contents may be different depending on the device model, but the numbers correspond.

(5) When parsing a single record, it is necessary to parse the string format records in the Buffer.