**MQTT PROTOCOL**

Security hint:

This protocol is confidential document. Please do not copy or send to others before permission.

INTRODUCTION

This doc. Describes the communication data format between controller and PC server based on MQTT protocol, and the data packet adopts JSON structure.

THIS DOC. IS SUITABLE FOR:1 DOOR / 2 DOOR/ 4 DOOR/ TURNSTILE CONTROLLER/EMBEDDED

FEATURES:

1. MQTT for IOT is more universal and extensible.
2. We can choose JSON or binary data structure.
3. MQTT and TCP protocol for option.
4. Updated to TLS encrypted communication.
5. JSON with DEMO is suitable for server development.
6. Support bulk downloading
7. Support hart beat PULL command to download data.

MQTT

Access controller connects MQTT server as MQTT client terminal. We need config parameters under built-in web interface.

Note:Access controller does not book and config topic to reduce communication exchange.

COMMUNICATION PROCESS:

1. Controller connects MQTT server.
2. Controller connects MQTT to make connection verification.
3. Controller sends data and Server responses.
4. Server sends command and controller responses.

JSON basic sturcture

Request party:

{

"id":1,

"version":"1.0",

"taskNo":10,

"method":""，

"data"：{}

}

应答方：Response party

{

"id":1,

"version":"1.0",

"code":"200",

" message ":"",

"taskNo":10,

"method":""

}

General field description

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **Description** |
| id | UInt32 | Serial number |
| version | string | Instruction version |
| code | string | Result code after executing command |
| message | string | Error description returned when there is an error |
| taskNo | UInt32 | Task number defined by the initiator. After receiver finishes the task, the original value is returned |
| data | Json object | JSON data object |
| method | String | This Command function is the same for both response and receiver party. |

Special note: All JSON field names, values, and symbols are case sensitive

Time length unit is second.

DATA TYPE DESCRIPTION

|  |  |
| --- | --- |
| **DATA TYPE** | **DESCRIPTION** |
| UInt8 | 8-bit unsigned integer, occupying 1 byte |
| UInt16 | 16 bit unsigned integer, occupying 2 bytes |
| UInt32 | 32 bit unsigned integer, occupying 4 bytes |
| string | string |
| Array | Json structure’s array,square brackets [] |
| Json object | JSON data object，{} or [] |

ERROR CODE TABLE

|  |  |  |
| --- | --- | --- |
| **CODE** | **Meaning** | **Explanations** |
| 0 | Execution success |  |
| 1 | Unsupported command | This model no or version do not support this command |
| 2 | Failure execution |  |
| 3 | JSON not exist | Missing a variable field, incorrect case |
| 4 | ILLEGAL JSON | JSON syntax is incorrect or not recognized. |
| 5 | Illegal JSON value | The value of the field is illegal JSON format |
| 6 | The Value is out of range | The field value is out of expected range |
| 7 | Do not search data |  |
| 8 | Parameters no. Are not consistent | The instruction requires multiple array variables, but the length of the array is inconsistent. |

**Command content**

|  |  |  |
| --- | --- | --- |
| **Command** | **Command definition** | **Explanation** |
| Status | Report controller heartbeat | Must respond first before executing other instructions |
| CardEvent | Report card swiping records |  |
| AlarmEvent | Report alarm records |  |
| OpenDoor | Open door |  |
| CloseDoor | Close door |  |
| OpenDoorLong | Door open long time |  |
| LockDoor | Lock door |  |
| SetAlarm | Alarm |  |
| SetFire | Fire alarm |  |
| SetPass | Control entry and exit |  |
| DeleteTimeZone | Delete time zone |  |
| AddTimezone | Add time zone |  |
| ClearCards | Clear cards |  |
| DeleteCard | Delete card |  |
| AddCard | Add card |  |
| AddAllCards | Bulk add cards---user ID mode | Suggest |
| AddListCards | Bulk add card--no user ID mode |  |
| SearchCard | Search card | Search by card no., user ID,password |
| UpdateCardAPB | update card antipass back |  |
| ClearHolidays | Clear all holidays |  |
| AddHoliday | Add holiday |  |
| TransmitTo485 | send data by RS485 | Support controller with RS485 interface |
| Speak | Voice broadcast | Support controller with RS485 interface |
| UpdateTime | Sync time | can enable NTP alternative, config on web interface |
| Restart | Restart |  |
| Reset | Reset |  |
| SetDoor | Set door parameters |  |
| SetController | Set controller parameters |  |

Status、CardEvent、AlarmEvent、AddCard

Develop in sequence

1. Status
2. OpenDoor 、CloseDoor
3. CardEvent、AlarmEvent
4. AddCard
5. ClearCards、DeleteCard、Reset
6. DeleteTimeZone、AddTimezone
7. AddAllCards
8. UpdateTime
9. AddHoliday、ClearHolidays
10. SetDoor、SetController

**Controller heart beat**

Controller sends Topic: /sys/%s/%s/thing/event/property/post

Controller response Topic:

|  |  |  |  |
| --- | --- | --- | --- |
| **控制器请求Controller requests** | | | **服务器应答Controller responses** |
| {  "id": 9,  "version": "1.0",  "method": "Status",  "taskNo":10,  "data": {  "Serial": "1G0095",  "ID": "1C1110",  "DoorStatus": [0,0],  "Input": 0,  "Now": "2022-12-09 12:34:48",  "Firmware": "0.1.121",  "BuildTime": "Dec 9 2022 12:34:08",  "PCBCode": "DD10",  "Model": "G-ZJ1000TCP",  "TimeStamp": 1670589288,  "CardsInPacket": 16,  "SystemOption": 57,  "DoorRelay": [0,0],  "AlarmRelay": [0,0],  "Button": [0,0],  "Sensor": [0,0],  "AlarmIn": 0,  "FireIn": 0,  "TamperIn": 0,  "ResetIn": 0  }  } | | | {  "method": " Status ",  "code": "0",  "taskNo":10,  "version": "1.0",  "id": 9,  "message": "",  "pull":  {  "IndexCmd":1,  "PullCmd":"OpenDoor"  }  } |
| **Field** | **Type** | **Explanation** | |
| Serial | String | 序列号Serial no. | |
| ID | String | The custom device’s identifying string | |
| Now | String | Device current system time, string representation | |
| TimeStamp | UInt32 | Device current system time, time stamp | |
| Firmware | String | Firmware version | |
| BuildTime | String | Firmware compilation time | |
| PCBCode | String | Hardware code, used to distinguish hardware differences | |
| Model | String | Device model no., This product’s function | |
| ModelCode | UInt16 | Model code, see the below model table | |
| CardsInPacket | UInt8 | Maximum number of packages per batch transmission | |
| SystemOption | UInt8 | Function code | |
| DoorStatus | Array | Door status, take 2 door controller as example, Array is [0,0] Starting from the left.The first number is the first door, The second number is the second door. The subsequent state class array will be in this format if there is no specific explanation. | |
| AlarmRelay | Array | Alarm relay status, bit0: Alarm relay, bit1:Fire alarm.0 close, 1 output. | |
| DoorRelay | Array | Door relay status, 0 close, 1 output | |
| Button | Array | Button input position status, 0 open, 1 close | |
| Sensor | Array | Door sensor input position status, 0 open, 1 close | |
| LockOut | Array | Locking status, 1 locking, 0 unlock | |
| Input | UInt32 | Input interface status | |
| AlarmIn | UInt8 | Alarm input status, 0 open, 1 close | |
| FireIn | UInt8 | Fire alarm input status, 0 open, 1 close | |
| TamperIn | UInt8 | Tamper input statu, 0 open, 1 close | |
| ResetIn | UInt8 | Reset key status, 0 release, 1, press down | |
| OEMCODE | UInt16 | Customer exclusive code | |
| IndexCmd | UInt32 | Pull comman serial no., generated by server, controller returns with original value | |
| CmdOK | UInt32 | Pulll command execution result, 0 unsuccessful, 1 succeed | |
| ErrorCode | UInt32 | Error code for pulling instruction execution results | |
| Value | UInt32 | The pull command returns a parameter, which is used to return the package number during batch transmission | |

**Server response with pull command**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Explanation** |
| pull | Json | Optional field, pull instruction content, if not, do not use this field |
| IndexCmd | Uint32 | Pull the instruction number, and the controller returns the original value when the next heartbeat occurs. |
| PullCmd | String | Pull comman name |
| …. | … | For specific pull instruction data, please refer to each specific instruction and add corresponding field parameters |

Model Code

|  |  |  |
| --- | --- | --- |
| **Model code** | **Type** | **Model no. explanation** |
| **0x71** | 1 door | Economic 1 door (built-in PSU) |
| **0x9E** | LCD TCP 1 door controller--Third generation |
| **0x90** | TCP turnstile controller--Third generation |
| **0x91** | TCP 1 door--Third generation |
| **0x72** | 2 door | Economic 2 door (built-in PSU) |
| **0x92** | TCP 2 Door --Third generation |
| **0x94** | 4 door | TCP 4 door--Third generation |
| **0x9C** | Cloud turnstile | Cloud turnstile-Third generation |
| **0xB4** | Cloud plus | Cloud+ B controller--Fourth generation, on one mode only |
| **0xC4** | Cloud+C controller--Fourth generation, on line+ off line mode |
| **0xCB** | Cloud 2 door | Cloud+ 2 door--Fourth generation |
| **0xCD** | Cloud 4 door | Cloud+ 4 door--Fourth generation |

**Controller send events---Event**

Controller sends Topic: /sys/%s/%s/thing/event/post

Controller responses Topic: (server self defined subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Controller request** | | | **Controller response** |
| {  "id": 159,  "version": "1.0",  "taskNo":10,  "method": "CardEvent",  "data": {  "Serial": "1G0095",  "Time": "20221208104151",  "DataType": 0,  "Card":"9527",  "EventType": 1,  "Door": 1,  "Reader": 0,  "Exist": 0,  "Pass": 0,  "APBInOut": 0,  "APBOn": 0,  "Count": 0,  "Index": 1  }  } | | | {  "method": "CardEvent",  "code": "0",  "taskNo":10,  "version": "1.0",  "id": 159,  "message": "",  "data": {  "Index": 1  }  } |
| **Field** | **Data type** | **Explanation** | |
| Serial | String | Controller serial no. | |
| Door | UInt8 | Door no. Starting from 1 | |
| Reader | UInt8 | This record was swiped on that reader, starting from 0 | |
| Card | String | Card no. | |
| DataType | UInt8 | card data type, 0 card | |
| Time | String | Event time | |
| EventType | UInt8 | Event type, see event table | |
| Index | UInt32 | Event record transmission flow, the response must be the same, and upon receiving the same return, the controller will delete this record | |
| Count | UInt32 | Current record quantity | |
| Exist | UInt8 | The card no. Of this event Exists in the controller or not | |
| Pass | UInt8 | This card swiping is allowed to pass or not. | |
| APBOn | UInt8 | This card swiping is enabled anti passback or not | |
| APBInOut | UInt8 | If enable anti passback, and valid pass, entry or exit, 0 entry, 1 exit | |

**The controller sends event-alarm records**

Controller sends Topic: /sys/%s/%s/thing/event/post

Server replies Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Controller request** | | | **Server reply** |
| {  "id": 1967,  "version": "1.0",  "taskNo":10,  "method": "AlarmEvent",  "data": {  "Serial": "1G0095",  "Time": "20221208115704",  "EventType": 59,  "Door": 1,  "Count": 0,  "Index": 2  }  } | | | {  "method": " AlarmEvent",  "code": "0",  "taskNo":10,  "version": "1.0",  "message": "",  "id": 1967,  "data": {  "Index": 2  }  } |
| **Fields** | **Data type** | **Notes** | |
| Serial | String | Serial number of the controller | |
| door | UInt8 | Door number, starting at 1 | |
| EventType | UInt8 | Event type, see Event Type Table description (excel file) | |
| Time | String | The time of the event | |
| Index | UInt32 | Event record transmission pipeline, the response must be the same, receive the same return, the controller will delete this record | |
| Count | UInt32 | Current number of records. | |

**Server sends command - open and close the door**

Server sends Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 2017,  "data": {  "Door": 0,  "Open": 1  },  "version": "1.0",  "taskNo":10,  "method": "OpenDoor"  } | | | {  "id": 2017,  "code": 0,  "taskNo":10,  "method": "OpenDoor",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| door | UInt8 | The gate number starts at 0 | |
| Open | UInt8 | Action, 0 close the door, 1 open the door | |

**Server sends command - long open door**

Server send Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 2042,  "data": {  "Door": 0  },  "version": "1.0",  "taskNo":10,  "method": " OpenDoorLong"  } | | | {  "id": 2043,  "code": 0,  "taskNo":10,  "method": " OpenDoorLong",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| door | UInt8 | The gate number starts at 0 | |

**The server sends the instruction - lock (unlock) the door**

Server sends Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 2058,  "data": {  "Door": 0,  "Lock": 1  },  "version": "1.0",  "method": "LockDoor",  "taskNo":10  } | | | {  "id": 2059,  "code": 0,  "taskNo":10,  "method": "LockDoor",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| door | UInt8 | Door number, starting at 0 | |
| Lock | UInt8 | Action, 0 off, 1 on | |

**Server sends command - alarm output**

Server sends Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 2114,  "taskNo":10,  "data": {  "Open": 1  },  "version": "1.0",  "method": "SetAlarm"  } | | | {  "id": 2115,  "taskNo":10,  "code": 0,  "method": "SetAlarm",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| Open | UInt8 | Action, 0 off, 1 on | |

**Server sends command - Start (turn off) fire alarm**

Server sends Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 2170,  "taskNo":10,  "data": {  "Open": 1  },  "version": "1.0",  "method": "SetFire"  } | | | {  "id": 2171,  "taskNo":10,  "code": 0,  "method": "SetFire",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| Open | UInt8 | Action, 0 off, 1 on | |

**Server sends command - Direction control (disallow card reading)**

Server send Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 2170,  "taskNo":10,  "data": {  "Door": 1,  "Reader": 1,  "Keepout": 1  },  "version": "1.0",  "method": " SetPass"  } | | | {  "id": 2171,  "taskNo":10,  "code": 0,  "method": " SetPass",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| door | UInt8 | Door number, starting at 0 | |
| Reader | UInt8 | The direction of the control, 0 in, 1 out | |
| Keepout | UInt8 | 1 is forbidden to pass, 0 is allowed to pass | |

**Server sends instructions - sync time**

Server send Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 2208,  "taskNo":10,  "data": {  "Second": 17,  "Day": 8,  "Minute": 5,  "Month": 12,  "Hour": 12,  "Year": 2022  },  "version": "1.0",  "method": "UpdateTime"  } | | | {  "id": 2209,  "taskNo":10,  "code": 0,  "method": "UpdateTime",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| Second | UInt8 | Seconds, in base 10 | |
| Minute | UInt8 | Minute, base 10 | |
| Hour | UInt8 | Hour, base 10 | |
| Day | UInt8 | Day, base 10 | |
| Month | UInt8 | Month, base 10 | |
| Year | UInt16 | Year, base 10 | |

**Server sends the command - controller restarts**

Server send Topic: (server defines its own subscription)

|  |  |
| --- | --- |
| **Server instructions** | **Controller reply** |
| {  "id": 2222,  "taskNo":10,  "data": {},  "version": "1.0",  "method": "Restart"  } | {  "id": 2223,  "taskNo":10,  "code": 0,  "method": "Restart",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |

**The server sends instructions - controller initialization**

Server send Topic: (server defines its own subscription)

|  |  |
| --- | --- |
| **Server instructions** | **Controller reply** |
| {  "id": 13,  "taskNo":10,  "data": {},  "version": "1.0",  "method": "Reset"  } | {  "id": 14,  "taskNo":10,  "code": 0,  "method": "Reset",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |

**Server send instruction - send data through 485**

Server send Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 18,  "taskNo":10,  "data": {  "Data": "MTIzNDU2QUJD"  },  "version": "1.0",  "method": "TransmitTo485"  } | | | {  "id": 19,  "taskNo":10,  "code": 0,  "method": "TransmitTo485",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| Data | string | A Base64-encoded string of the data to send via 485 | |

**The server sends the command - broadcast the voice**

Server sends Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 18,  "taskNo":10,  "data": {  "Data": "abc"  },  "version": "1.0",  "method": "Speak"  } | | | {  "id": 19,  "taskNo":10,  "code": 0,  "method": "Speak",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **fields** | **Data type** | **Instructions** | |
| Encoding | Uint8 | The code for the broadcast text, 0=GB2312, 4=UTF8 | |
| Ring | Uint8 | Ring before broadcast, 0 does not ring, nothing defaults to 1 | |
| Data | string | Broadcast message string, maximum length 128 bytes | |

**Server sends instruction - delete all cards**

Server send Topic: (server defines its own subscription)

|  |  |
| --- | --- |
| **Server instructions** | **Controller reply** |
| {  "id": 32,  "taskNo":10,  "data": {},  "version": "1.0",  "method": "ClearCards"  } | {  "id": 33,  "taskNo":10,  "code": 0,  "method": "ClearCards",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |

This instruction takes a different amount of time to execute depending on the model, ranging from 0.2 seconds -15 seconds, but an answer will be received immediately after sending the instruction. It is recommended that the software wait until the next heartbeat before issuing other commands.

**Server sends command - delete 1 card**

Server send Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 64,  "taskNo":10,  "data": {  "Card": 444855280,  "CardIndex": 0  },  "version": "1.0",  "method": "DeleteCard"  } | | | {  "id": 65,  "taskNo":10,  "code": 0,  "method": "DeleteCard",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {  "CardIndex": 0  }  } |
| **fields** | **Data type** | **Instructions** | |
| Card | UInt32 | Card number to delete, in base 10 | |
| CardIndex | UInt32 | Optional argument, deletes data with specified number, Card is not needed if this field is present. This parameter is preferred and recommended | |

**Note:**

Regarding card management, there are 2 method modes, with and without card numbers. This is decided by the management software, if the card number is used, the card number is used for deletion, addition, and batch addition, otherwise the card number is not used. Using the card number can improve performance, especially when the number of cards is large, the **card number mode is strongly recommended**, but some software architectures may not be suitable for using card numbers. Card numbering in this protocol uses the field name "CardIndex".

**Noun interpretation: Card number**

A list of people needs to be maintained in the computer software. Each record represents the information of a person and generates a unique ID, which is the card number. The card number must start from 0, must not be interrupted, must not be repeated, and must not exceed the controller's card capacity (about 30,000-60,000). So when the software deletes a card, it just leaves out the number. When you add a card, you need to use the number that was previously removed.

**The server sends the instruction - add 1 card**

Server Send Topic: (server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instructions** | | | **Controller reply** |
| {  "id": 44,  "taskNo":10,  "data": {  "CardIndex": 0  "TZ1": 1,  "TZ2": 0,  "Card": 444855280,  "Password": "87654321",  "BeginTime": "3007232359",  "EndTime": "2212161647",  "Name": "Card A",  "Status": 1  },  "version": "1.0",  "method": "AddCard"  } | | | {  "id": 45,  "taskNo":10,  "code": 0,  "method": "AddCard",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {  "CardIndex": 0  }  } |
| **Fields** | **Data type** | **Notes** | |
| CardIndex | UInt32 | Optional parameter, starts at 0 and does not exceed controller capacity -1. Save by this specified number if there is one, otherwise search for card number coverage, and automatically search for blank position to save if not present. | |
| Card | UInt32 | Card number, 4 byte integer in decimal | |
| TZ1 | UInt8/UInt16 | Opening time- The first gate of 124 door controller, expressed in binary bits,  Each bit represents 1 opening time, corresponding to the whole opening schedule in order, 0 no choice no permission,1 choice | |
| TZ2 | UInt8/UInt16 | Opening time - 24 door controller 2nd door | |
| TZ3 | UInt8 | Open time selection - 4 door controller controller 3rd gate | |
| TZ4 | UInt8 | Opening time Selection - 4 door controller 4th gate, 8 bits, each bit represents 1 opening time, corresponding to the whole opening schedule in sequence. | |
| Password | string | Optional argument, password no more than 8 bits | |
| BeginTime | String | Optional parameter, expiration start yyMMddHHmm; Decimal string; The year, month and day in byte order, such as 3001012359:30 for 2030,  01 for month 01, 01 for day 01,  23 for 23:00, 59 for 59 minutes | |
| EndTime | String | Optional parameter, expiration yyMMddHHmm; 10 string, the year needs to be subtracted by 2000. | |
| Name | string | Optional parameter, person name | |
| Status | UInt8 | Optional parameter, enabled status of personnel 1 available, 0 not available | |

**Return results**

|  |  |  |
| --- | --- | --- |
| CardIndex | UInt32 | Returns the argument, the actual saved card number, which is the location of the card |

**Optimization tip**: If you need to download all or most of the cards, you can delete all the cards first, and then download, so that the performance can be improved at least by more than 1 times. Add after removing all the cards, the performance is higher. The same works for bulk additions.

**The server sends instructions - update card status, anti-pass back function**

The server sends Topic: (The server defines the subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instruction** | | | **Controller answer** |
| {  "id": 44,  "taskNo":10,  "data": {  "CardIndex": 0,  "Card": 444855280,  "APBInOut": 0,  "APBOn": 0,  "Times": [1,2,3,4,5],  "Time": "221216164735"  },  "version": "1.0",  "method": "UpdateCardAPB"  } | | | {  "id": 45,  "taskNo":10,  "code": 0,  "method": "UpdateCardAPB",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {  "CardIndex": 0  }  } |
| **Fields** | **Type** | **Instructions** | |
| CardIndex | UInt32 | This parameter is optional,starts from 0 and does not exceed controller capacity -1. If there is, it will be saved by this specified number, otherwise it will automatically search for a blank place to save. | |
| APBOn | UInt8 | Indicates whether anti-pass back is enabled for this card. 0 indicates that anti-pass back is not enabled and 1 indicates that anti-pass back  is enabled | |
| APBInOut | UInt8 | Set whether the card is in or out. 0 in, 1 out. | |
| Times | Array | Optional parameter, the number of swipes per the first 5 opening times | |
| Time | String | Optional parameter, time of entry and exit. It is a character string in base 10. The year must be subtracted by 2000 first. | |

**The server sends the command - search for 1 card**

Server sends Topic: (Server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instruction** | | | **Controller answer** |
| {  "id": 82,  "taskNo":10,  "data": {  "Card": 444855280  },  "version": "1.0",  "method": "SearchCard"  } | | | {  "id": 83,  "taskNo":10,  "code": 0,  "method": "SearchCard",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {  "CardIndex": 0,  "Card": 444855280,  "Status": 1,  "TZ1": 1,  "BeginTime": "3007232359",  "EndTime": "3001012359",  "Password": "87654321",  "Name": "Card A"  }  } |
| **fields** | **Data type** | **Instructions** | |
| Card | UInt32 | Search card number, which is a decimal number, optional | |
| Password | String | Optional argument, the password of the search | |
| CardIndex | UInt32 | Card number, that is, the position of the card, is optional | |

Choose one of 3 parameters!

Return results

|  |  |  |
| --- | --- | --- |
| Card | UInt32 | Card number, decimal number |
| CardIndex | UInt32 | Card number, that is, the location of the card |
| Status | UInt8 | Search results - State |
| TZ1 | UInt8/UInt16 | Opening Time selection - 124-door controller door 1, see Add card instructions |
| TZ2 | UInt8/UInt16 | Opening Time selection - 24 door controller 2nd door |
| TZ3 | UInt8 | Opening Time Selection - 4 door controller 3rd door |
| TZ4 | UInt8 | Opening Time Selection - 4 door controller 4th door |
| BeginTime | String | Expiration date begins. A string in base 10; The year, month and day in byte order, such as 3001012359:  30 represents the year 2030,  01 for 01 month, 01 for 01 day,  23 for 23 o 'clock, 59 for 59 minutes |
| EndTime | String | This string is valid until it expires. It is a character string in base 10. The year must be reduced by 2000. |
| Password | String | Password |
| Name | String | Name |

**Server send instructions - Batch add cards (with numbered mode, recommended)**

Server send Topic: (Server defines subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instruction** | | | **Controller answer** |
| {  "id": 106,  "taskNo":10,  "data":  {  "CardofPack": 16,  "PackIndex": 0,  "TZ1" :,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 [1],  "TZ2" :,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 [1],  "Card": [  2001200 2200 3200 4200 5200 6200 7200 8200 9,  2010201 1201 2201 3201 4201 5201 6],  "Password": [  "5000", "5001", "5002", "5003", "5004", "5005", "5006", "5007",  "5008", "5009", "5010", "5011", "5012", "5013", "5014", "5015"].  "BeginTime": ["3001012359","3001012359","3001012359","3001012359",  "3001012359", "3001012359", "3001012359", "3001012359",  "3001012359", "3001012359", "3001012359", "3001012359",  "3001012359", "3001012359", "3001012359", "3001012359"].  "EndTime": ["3001012359","3001012359","3001012359","3001012359",  "3001012359", "3001012359", "3001012359", "3001012359",  "3001012359", "3001012359", "3001012359", "3001012359",  "3001012359", "3001012359", "3001012359", "3001012359"].  "Name": ["Card A1","Card A2","Card A3","Card A4",  "Card A5","Card A6","Card A7","Card A8",  "Card A9","Card A10","Card A11","Card A12",  "Card A13","Card A14","Card A15","Card A16"]  },  "version": "1.0",  "method": "AddAllCards"  } | | | {  "id": 107,  "taskNo":10,  "code": 0,  "method": "AddAllCards",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {  "PackIndex": 0  }  } |
| **Fields** | **Data type** | **Instructions** | |
| CardofPack | Uint8 | The current number of cards in the packet, the size does not exceed the CardsInPacket value in the heartbeat. | |
| PackIndex | Uint32 | Batch transmission, batch serial number, starting from 0. The original controller answer indicates that the transmission is successful. | |
| Card | Array | Uint32 Array of integers, the decimal number of the card | |
| TZ1 | Array | Array of Uint8/Uint16 integers,  Permission opening time selection - Door 1 of the 124-door controller, see Add card instructions | |
| TZ2 | Array | Uint8/Uint16 integer array, permission Open time selection - 24 door controller 2nd door | |
| TZ3 | Array | Uint8 integer array, permission Open time selection - 4 door controller 3rd door | |
| TZ4 | Array | Uint8 integer array, permission Open time selection - 4-door controller 4th door | |
| Password | Array | string array, each person's password | |
| Name | Array | string array, each person's name | |
| BeginTime | Array | string array, a string in base 10 at the beginning of the validity period; The year, month and day in byte order, such as 3001012359:30 for 2030,  01 for 01 month, 01 for 01 day,  23 for 23 o 'clock, 59 for 59 minutes | |
| EndTime | Array | It's valid until the end of time. It is a character string in base 10. The year must be reduced by 2000. | |

In this mode, the performance is highest, the number of packets must be CardsInPacket except the last packet, and the data is stored consecutively in numbered order. The number of the first person in each packet is PackIndex\*CardsInPacket, and the number of subsequent people in the packet is increased by 1. This mode is applicable to the management of processing personnel by number. Each person has a corresponding fixed number and is consecutive. Or it can be used to download all personnel.

**Server sends instructions - Batch add people (no numbered mode, not recommended)**

Server send Topic: (Server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instruction** | | | **Controller answer** |
| {  "id": 106,  "taskNo":10,  "data":  {  "CardofPack": 16,  "PackIndex": 0,  "TZ1" :,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 [1],  "TZ2" :,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 [1],  "Card": [  2001200 2200 3200 4200 5200 6200 7200 8200 9,  2010201 1201 2201 3201 4201 5201 6],  "Password": [  "5000", "5001", "5002", "5003", "5004", "5005", "5006", "5007",  "5008", "5009", "5010", "5011", "5012", "5013", "5014", "5015"].  "BeginTime": ["3001012359","3001012359","3001012359","3001012359",  "3001012359", "3001012359", "3001012359", "3001012359",  "3001012359", "3001012359", "3001012359", "3001012359",  "3001012359", "3001012359", "3001012359", "3001012359"].  "EndTime": ["3001012359","3001012359","3001012359","3001012359",  "3001012359", "3001012359", "3001012359", "3001012359",  "3001012359", "3001012359", "3001012359", "3001012359",  "3001012359", "3001012359", "3001012359", "3001012359"].  "Name": ["Card A1","Card A2","Card A3","Card A4",  "Card A5","Card A6","Card A7","Card A8",  "Card A9","Card A10","Card A11","Card A12",  "Card A13","Card A14","Card A15","Card A16"]  },  "version": "1.0",  "method": "AddListCards"  } | | | {  "id": 107,  "taskNo":10,  "code": 0,  "method": "AddListCards",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {  "PackIndex": 0  }  } |
| **Fields** | **Data type** | **Instructions** | |
| CardofPack | Uint8 | The current number of cards in the packet, the size does not exceed the CardsInPacket value in the heartbeat. | |
| PackIndex | Uint32 | The value starts from 0. The original controller response indicates that the transmission is successful. | |
| Card | Array | Uint32 Array of integers, the decimal number of the card | |
| TZ1 | Array | Uint8/Uint16 integer array, permission open time selection -- 124-door controller door 1,  See Add card instructions | |
| TZ2 | Array | Uint8/Uint16 integer array, permission Open time selection - 24 door controller 2nd door | |
| TZ3 | Array | Uint8 integer array, permission Open time selection - 4 door controller 3rd door | |
| TZ4 | Array | Uint8 integer array, permission open time selection - 4 gate controller 4 gate | |
| Password | Array | string array, each person's password | |
| Name | Array | string array, each person's name | |
| BeginTime | Array | string array, a string in base 10 at the beginning of the validity period; The year, month and day in byte order, such as 3001012359:30 for 2030,  01 for 01 month, 01 for 01 day,  23 for 23 o 'clock, 59 for 59 minutes | |
| EndTime | Array | It's valid until the end of time. It is a character string in base 10. The year must be reduced by 2000. | |

The number of elements in the array must be the same. This mode is suitable for unnumbered management, personnel according to the card number as the only key value, search by the card number, if it exists, it is overwritten, otherwise save to a blank position. It takes 3-5 seconds for each instruction to execute, and the performance improvement in this mode is limited, so it is not recommended.

**The server sends instructions - initializing all open hours**

Server sends Topic: (Server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instruction** | | | **Controller answer** |
| {  "id": 256,  "taskNo":10,  "data": {  "Door": 0  },  "version": "1.0",  "method": "DeleteTimeZone"  } | | | {  "id": 257,  "taskNo":10,  "code": 0,  "method": "DeleteTimeZone",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| door | UInt8 | Door number, starting from 0 | |

**Server sends instruction - Add opening hours**

Server sends Topic: (Server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instruction** | | | **Controller response** |
| {  "id": 274,  "taskNo":10,  "data": {  "Door": 0,  "Index": 0,  "Identify": 0,  "Group": 0,  "FrmHour": 0,  "FrmMinute": 0,  "ToMinute": 59,  "ToHour": 23,  "Passback": 0,  "Holiday": 1,  "Year": 30,  "Month": 1,  "Day": 1,  "Week": [1,1,1,1,1,1]  },  "version": "1.0",  "method": "AddTimezone"  } | | | {  "id": 275,  "taskNo":10,  "code": 0,  "method": "AddTimezone",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| door | UInt8 | Serial number of the door, starting from 0 | |
| Index | UInt8 | Serial number of opening hours, starting from 0 | |
| Identify | UInt8 | Identification method, pick only one  1: Single card identification  2: card plus PIN  3: Open and close the door on a regular basis  7: Regular alarm  8: Password  9: Card or PIN  17: Button time | |
| Group | UInt16 | Group number | |
| FrmHour | UInt8 | Start time hour , decimal | |
| FrmMinute | UInt8 | Start time minutes, decimal | |
| ToHour | UInt8 | End time hour ,decimal | |
| ToMinute | UInt8 | End time minutes, decimal | |
| Year | UInt16 | End date of opening hours year, decimal | |
| Month | UInt8 | Month, decimal | |
| Day | UInt8 | Day, decimal | |
| Holiday | UInt8 | Holiday enabled or not 1 Yes 0 no | |
| Week | Array | Week Select array, in order Sunday, Monday, Tuesday... Saturday; [1,1,1,1,1,1] 1 Disable for enable 0 | |
| Passback | UInt8 | anti-passback 1 Enable 0 Disable | |

**Server sends instruction - Update gate parameters**

Server sends Topic: (Server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instruction** | | | **Controller answer** |
| {  "id": 286,  "taskNo":10,  "data": {  "MCardsInOut": 0,  "OutTime": 5,  "AlarmMask" :,1,1,1 [1],  "MCards": 0,  "Door": 0,  "OpenTime": 1,  "AlarmTime": 10,  "ToolongAlarm": 0  },  "version": "1.0",  "method": "SetDoor"  } | | | {  "id": 287,  "taskNo":10,  "code": 0,  "method": "SetDoor",  "message": "",  "Serial": "1G0095",  version: 1.0,  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| door | UInt8 | Door number, starting from 0 | |
| OpenTime | UInt16 | Open time | |
| OutTime | UInt8 | The time range of the door opening timeout detection after the door is closed | |
| AlarmMask | Array | Alarm type Generates alarm output (alarm relay action), 4 options, 1 enabled, 0 disabled; Order: [door alarm, open the door for too long alarm, invalid card alarm, invalid time alarm] | |
| AlarmTime | UInt16 | Alarm output alarm time | |
| ToolongAlarm | UInt8 | Whether to enable the function of keeping the door open for too long: 1 enabled, 0 disabled | |
| MCards | UInt8 | Multi-card opens the door, with a value greater than 1 indicating enabled | |
| MCardsInOut | UInt8 | When the multi-card door is enabled, the enabled directions are 1 in direction, 2 out direction, and 3 in and out of both directions | |

**The server sends instructions - update controller parameters**

The server sends Topic: (Server defines its own subscription)

|  |  |  |  |
| --- | --- | --- | --- |
| **Server instruction** | | | **Controller answer** |
| {  "id": 298,  "taskNo":10,  "data": {  "FireTime": 0,  "LockEach": 0,  "Duress": "5874",  "AlarmTime": 10  },  "version": "1.0",  "method": "SetController"  } | | | {  "id": 299,  "taskNo":10,  "code": 0,  "method": "SetController",  "message": "",  "Serial": "1G0095",  "version": "1.0",  "data": {}  } |
| **Fields** | **Data type** | **Instructions** | |
| LockEach | UInt8 | 0: no interlock, single door, turnstile controller does not have this field;  1:1-2 door interlock  2:3-4 door interlock  3:1-2-3 Doors interlock  4:1-2-3-4 doors interlock | |
| FireTime | UInt16 | Length of time the alarm relay output after the fire alarm is triggered | |
| Duress | String | Hijack password, string of numbers | |
| AlarmTime | UInt16 | The length of time the alarm relay outputs after the alarm input is triggered | |
| Title | string | The title displayed on the LCD interface. Only the standalone access controller has this field | |