**Card Encoder User Manual**

**V1.6.1**

**Hangzhou Sciener Intelligent Control Technology**

Apr. 28, 2023

This document is the introduction of APIs for Card Encoder

# Definitions

DLL Name：

CardEncoder.dll

mfc140u.dll

mfc140ud.dll

msvcp140.dll

msvcp140d.dll

ucrtbase.dll

ucrtbased.dll

vcruntime140.dll

vcruntime140d.dll

System Requirement：

Win7 or Win10

Notes：

**All these functions can only be called AFTER the END of the previous one, that is after the returning of the previous function.**

# Guideline

## How to get hotelInfo

Call this api to get hotelInfo：

**China：[https://cnapi.ttlock.com/v3/hotel/getInfo](https://api.sciener.cn/v3/hotel/getInfo?clientId=xxx&clientSecret=xxx&date=1570000000000获取)**

**Others：[https://euapi.ttlock.com/v3/hotel/getInfo](https://api.sciener.cn/v3/hotel/getInfo?clientId=xxx&clientSecret=xxx&date=1570000000000获取)**

API：/v3/hotel/getInfo, GET

Parameters：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | required | type | description | example |
| clientId | Y | String | app\_id | **Create application from [https://euopen.ttlock.com/](https://open.ttlock.com/)** |
| clientSecret | Y | String | app\_secret |
| date | Y | Long | Current timestamp in milsec | like：1570000000000 |

return：

|  |  |
| --- | --- |
| value | description |
| hotelInfo | It is valid for 10 minutes |
| errcode | Error code |
| errmsg | Error message |

## How to config local server

**Get timestamp from server**

**TTLock Open Platform:**

**China：[https://cnapi.ttlock.com/v3/hotel/getServerDateTime](https://api.sciener.cn/v3/hotel/getInfo?clientId=xxx&clientSecret=xxx&date=1570000000000获取)**

**Others：[https://euapi.ttlock.com/v3/hotel/getServerDateTime](https://api.sciener.cn/v3/hotel/getInfo?clientId=xxx&clientSecret=xxx&date=1570000000000获取)**

API：http/https, GET

Parameters：-

return：JSON

|  |  |
| --- | --- |
| value | description |
| date | Server timestamp, unit: hour |

# Error Code

Normal：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Fail |
| 2 | Bad parameter |
| 3 | Communication error (write) |
| 4 | Communication error (read) |
| 5 | Command error (device returns error) |
| 6 | key unconfigured |
| 7 | Not in the status of issuing card |
| 8 | Failed to interrupt |
| 9 | Interrupted |
| 10 | Have not configured the server address |
| 11 | Failed to request server (You can learn more details from log) |
| 12 | Server returning failed |
| 13 | Bad hotelInfo (You can learn more details from log) |
| 14 | Not the hotel’s own card encoder |
| 15 | Card encoder has not been initialized. (1.3.0) |
| 16 | Can’t connect to card encoder. (1.3.0) |
| 21 | Not IC card |
| 26 | Failed to disconnect. (1.3.0) |
| 28 | Failed to config comm. (1.3.0) |
| 31 | Failed to open log file. (1.3.0) |
| 32 | Command not supported. (1.4.0) |
| 33 | CPU card is not supported. (1.4.0) |
| 34 | Sector parse failed(1.6.0, V2) |
| 35 | Sector out of bounds(1.6.0, V2) |
| 36 | No available sector(1.6.0, V2) |
| 37 | Incomplete sector data(1.6.1, V2) |
| 201 | Failed to config key |
| 202 | Failed to config card key |
| 203 | Failed to config hotel info |

Device related:：

|  |  |
| --- | --- |
| 101 | Others |
| 102 | Timeout |
| 104 | Run out of card memory |
| 105 | Failed to decrypt or key unconfigured |
| 106 | Failed to decrypt |
| 107 | No such record |
| 108 | Failed to verification |
| 109 | Have not configured hotel ID |
| 110 | Illegal operation, unable to report the loss of current card |

Data parsing error：

|  |  |
| --- | --- |
| 301 | Other errors of parsing data |
| 304 | Run out of sector memory |
| 305 | Failed to decrypt with key or key unconfigured |
| 307 | No IC card data |
| 308 | Need to operate next sector(1.6.0, V2) |
| 309 | Params error(1.6.0, V2) |
| 310 | Search sector data error(1.6.0, V2) |
| 420 | failed to return data |

# Functions

## bool CE\_ConfigServer (const char \*url)

Description：The DLL will get current time from ntp.org by default. You can use your own server by calling this function. Otherwise, just ignore it.

The DLL will only get time from server when issue master card, building card, or floor card. It will not call server when issue card for room.

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*url | Your time server |  |

Return：

|  |  |
| --- | --- |
| Value | Description |
| true | Success |
| false | Failed |

## int CE\_ConnectCommOnPromise (const wchar\_t \*portName, bool useReverseCardNo)

Description：Connect to the port to which to card encoder is connected,

Parameter：portName (COM1~COM256)

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const wchar\_t \*portName | Port name | COM1~COM256 |
| bool useReverseCardNo | Use reversed card number. |  |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Failed |
| 26 | Failed to disconnect. (1.3.0) |
| 28 | Failed to config comm. (1.3.0) |

## int CE\_ConnectComm (const wchar\_t \*portName)

Description：Connect to the port to which to card encoder is connected, **Use reversed card number.**

Parameter：portName (COM1~COM256)

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const wchar\_t \*portName | Port name | COM1~COM256 |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Failed |
| 26 | Failed to disconnect. (1.3.0) |
| 28 | Failed to config comm. (1.3.0) |

## int CE\_DisconnectComm(void)

Description：Disconnect from the port

Parameter：null

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Failed |

## int CE\_InitCardEncoder(const char \*hotelInfo)

Description： Initialize the card encoder to be used for specific hotel. (Use network)

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | Assigned or get from server |  |

Return：

|  |  |  |
| --- | --- | --- |
| Value | Description |  |
| 0 | Successs |  |
| 2 | Bad parameter |  |
| 11 | Failed to request server |  |
| 12 | Server returning failed |  |
| 13 | Bad hotelInfo |  |
| 201 | Failed to config key |  |
| 202 | Failed to config card key |  |
| 203 | Failed to config hotel info |  |
| [More information](#_Error Code) | | |

## CE\_InitCard(const char \*hotelInfo)

Description：Initialize an empty card to make it usable in specific hotel

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | Assigned or get from server |  |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| 3 | Failed to send command or device disconnected |
| 4 | Communication error or interrupted |
| 5 | Command error |
| 7 | Not in the status of issuing card |
| 11 | Failed to request server |
| 12 | Server returning failed |
| 13 | Bad hotelInfo |
| [More information](#_Error Code) | |

## int CE\_StopInitCard(void)

Description：This function will stop initializing card. **You MUST call it in a separated thread, or there may be exceptions.**

Parameter： null

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Failed |
| [More information](#_Error Code) | |

## int CE\_WriteCard(const char \*hotelInfo, int buildNo, int floorNo, const char \*mac, unsigned long timestamp, bool allowLockOut)

Description：Write the access permission into the card. After this, the card can be used to unlock.

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | Hotel info |  |
| int buildNo | Building Number | 0 ~ 254 |
| int floorNo | Floor Number | 0 ~ 255 |
| const char \*mac | Lock MAC | Mac address without colon |
| unsigned long timestamp | Timestamp of expired time | In second |
| bool allowLockOut | If allow to unlock double locked |  |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| 3 | Failed to send command or device disconnected |
| 4 | Communication error or interrupted |
| 5 | Command error |
| 10 | Have not configured the server address |
| 11 | Failed to request server |
| 12 | Server returning failed |
| 13 | Bad hotelInfo |
| 101 | Others |
| 102 | Timeout |
| 104 | Run out of card memory |
| 105 | Failed to decrypt or key unconfigured |
| 106 | Failed to decrypt |
| 109 | Have not configured hotel ID |
| [More information](#_Error Code) | |

Description of some parameters：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| data type | buildNo | floorNo | mac | 备注 |
| master card | 0 | 0 | “000000000000” | Offline operations are not allowed |
| building card | √ | 0 | “000000000000” |
| floor card | √ | √ | “000000000000” |
| room card | √ | √ | √ | Offline operations are allowed |

## int CE\_ClearCard(const char \*hotelInfo)

Description：Clear all the access permission in the card

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | Assigned or get from server |  |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| 3 | Failed to send command or device disconnected |
| 4 | Communication error or interrupted |
| 5 | Command error |
| 10 | Have not configured the server address |
| 11 | Failed to request server |
| 12 | Server returning failed |
| 13 | Bad hotelInfo |
| 101 | Others |
| 102 | Timeout |
| 105 | Failed to decrypt or key unconfigured |
| 106 | Failed to decrypt |
| [More information](#_Error Code) | |

## int CE\_ReadCard(const char \*hotelInfo, char \*\*hotelArray)

Description：Read all the access permission from the card

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | Assigned or get from server |  |
| char \*\*hotelArray | The data from the card | JSON string |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| 3 | Failed to send command or device disconnected |
| 4 | Communication error or interrupted |
| 5 | Command error |
| 101 | Others |
| 102 | Timeout |
| 105 | Failed to decrypt or key unconfigured |
| 106 | Failed to decrypt |
| 109 | Have not configured hotel ID |
| [More information](#_Error Code) | |

## int CE\_GetCardNo(char \*\*cardNumber)

Description：Get the card number

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| char \*\*cardNumber | Card number | char \*cardnumber = NULL, decimal system |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Failed |
| 21 | Not IC card |
| [More information](#_Error Code) | |

## char \*CE\_GetCardID() (1.4.0)

Description：Get the card number

Parameter：none

Return：JSON

|  |  |  |
| --- | --- | --- |
| Key | Description |  |
| errorCode |  |  |
| data | Card ID |  |
| type | Card type | 0 -M1  2 -CPU |
| transformData | Transformed Card ID |  |

Error code:

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Failed |
| 21 | Not IC card |
| [More information](#_Error Code) | |

## int CE\_Beep(int voiceLen, int interval, int voiceCount)

Description：Manage the beep behavior. You can set the voice length, interval and times.

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| int voiceLen | The length of each beep | 1-10000（millisecond） |
| int interval | The interval | 1-10000（millisecond） |
| int voiceCount | Time | 1-50 |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Failed |
| 2 | Bad parameter |
| [More information](#_Error Code) | |

## int CE\_GetVersion(char \*\*versions)

Description：Get the version of card encoder

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| char \*\*versions | The version | JSON string |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Failed |
| [More information](#_Error Code) | |

## int CE\_InitConstructionCard()

Description：Initialize Construction Cards. These cards can be used to unlock all locks which have not been initialized

Parameter：null

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| [More information](#_Error Code) | |

## int CE\_DeInitCard(const char \*hotelInfo)

Description： Deinitialize the card into empty card, then the card can be initialized to any other hotels

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | The original hotelinfo |  |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| [More information](#_Error Code) | |

## int CE\_CancelCard(const char \*hotelInfo, const char \*cardNumber, unsigned long timestamp)

Description： Write the number of card which need to be cancelled.

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | The original hotelinfo |  |
| const char \*cardNumber | Card number | decimal system |
| unsigned long timestamp | Timestamp of expired time | In second |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| [More information](#_Error Code) | |

## int CE\_ReadCancellationInfo(const char \*hotelInfo, char \*\*infoArray)

Description：Read all the cancelled information from the card

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | Assigned or get from server |  |
| char \*\*infoArray | The data from the card | JSON string |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| [More information](#_Error Code) | |

## int CE\_SetSectors(const char \*sectors)

Description：Set the sector configuration to the card encoder

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*sectors | Tell the card encoder which sectors on the card are available or unavailable | It is a string consists with “0” and “1”. The length is 16.  Like ”0000111100001111”.  Each digit represents a sector.  1 is available, 0 unavailable.  Tips：all 1 or all 0 means all the sectors are available |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| 101 | Others |
| [More information](#_Error Code) | |

## int CE\_GetSectors(char \*\*sectorStr)

Description：Read the sector configuration from the card encoder

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| char \*\*sectorStr | Read from the card encoder which sectors on the card are available or unavailable | It is a string consists with “0” and “1”. The length is 16.  Like ”0000111100001111”.  Each digit represents a sector.  1 is available, 0 unavailable.  Tips：all 1 or all 0 means all the sectors are available  Initialize fist：char \*infoArray = NULL  The value is in JSON |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| 101 | Others |
| [More information](#_Error Code) | |

## int CE\_ParseResData(const char \*hotelInfo, unsigned char \*sectorData, bool isLowestSector, char \*\*hotelArray) (1.2.0)

Description：Parse the data of a sector. For 3rd party encoders（1.2.0）

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | Assigned or get from server |  |
| unsigned char \*sectorData | The data of current selected sector | Fixed 48 bits |
| bool isLowestSector | Is it the lowest sector |  |
| char \*\*hotelArray | The data from the card | JSON string |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| [More information](#_Error Code) | |

## int CE\_GenerateSectorData(const char \*hotelInfo, unsigned char \*sectorData, bool isLowestSector, int buildNo, int floorNo, const char \*mac, unsigned long timestamp, bool allowLockOut) (1.2.0)

Description：Generate data for one sector. It is for 3rd party encoders. Please make sure you have configured the server url before calling this function.（1.2.0）

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | Assigned or get from server |  |
| unsigned char \*sectorData | Sector data | Fixed 48 bits |
| bool isLowestSector | Is it the lowest sector |  |
| int buildNo | Building Number | 0 ~ 254 |
| int floorNo | Floor Number | 0 ~ 255 |
| const char \*mac | Lock MAC | Mac address without colon，end with’\0’，total length 13 characters |
| unsigned long timestamp | Timestamp of expired time | In second |
| bool allowLockOut | If allow to unlock double locked |  |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| [More information](#_Error Code) | |

Description of some parameters：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| data type | buildNo | floorNo | mac | Description |
| master card | 0 | 0 | “000000000000” | Offline operations are not allowed |
| building card | √ | 0 | “000000000000” |
| floor card | √ | √ | “000000000000” |
| room card | √ | √ | √ | Offline operations are allowed |

## int CE\_GenerateClearData(unsigned char \*srcBytes) (1.2.0)

Description：Generate data to clear one sector data of a card. It is for 3rd party encoders.（1.2.0）

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| unsigned char \*srcBytes | Writing the generated data to card will clear current data | Fixed 48 bits |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Fail |
| [More information](#_Error Code) | |

## int CE\_GenerateCancelCardData(const char \*hotelInfo, unsigned char \*sectorData, bool isLowestSector, const char \*cardNo, unsigned long timestamp) (1.2.1)

Description：Generate data to cancel a card. It is for 3rd party encoders.

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | Assigned or get from server |  |
| unsigned char \*sectorData | Sector data | Fixed 48 bits |
| bool isLowestSector | Is it the lowest sector |  |
| const char \*cardNumber | Card number | decimal system |
| unsigned long timestamp | Timestamp of expired time | In second |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Fail |
| [More information](#_Error Code) | |

## int CE\_ParseLossData(const char \*hotelInfo, unsigned char \*sectorData, bool isLowestSector, char \*\*infoArray) (1.2.1)

Description：Parse the CancelCardData. It is for 3rd party encoders.

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| const char \*hotelInfo | Assigned or get from server |  |
| unsigned char \*sectorData | Sector Data | Fixed 48 bits |
| bool isLowestSector | Is it the lowest sector |  |
| char \*\*infoArray | The data from the card | JSON string |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Fail |
| [More information](#_Error Code) | |

## int CE\_GenerateConstructionCardData(unsigned char \*blockData, const char \*cardNo) (1.2.1)

Description：Generate data to create construction card.The data should be written to 2 block of 0 sector(the block is start from 0). It is for 3rd party encoders., **Use reversed card number.**

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| unsigned char \*blockData | Sector data | Fixed 16 bits |
| const char \*cardNo | Card number | decimal system |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Fail |
| [More information](#_Error Code) | |

## int CE\_GetCPUCardSupport() (1.4.0)

Description：Judge whether the encoder supports CPU card operation

Parameter：none

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Support. |
| 1 | Fail |
| 32 | Command not supported. |
| 33 | CPU card is not supported. |
| [More information](#_Error Code) | |

## int CE\_ReadSectorRawData(int sectorNum, int blockNum, bool isEncrypted, const unsigned char \*IC\_KEY, unsigned char \*blockData) (1.5.0)

Description：Read block data from current card.

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| int sectorNum | Sector number | Integer, [0, 15] |
| int blockNum | Block Index | [0, 2], it can’t be 0 while sectorNum is 0 |
| bool isEncrypted | Wheather the block is encrypted |  |
| unsigned char \*IC\_KEY | Block IC key |  |
| unsigned char \*blockData | Block data from card | Fixed 16 bits |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Fail |
| [More information](#_Error Code) | |

## int CE\_WriteSectorRawData(int sectorNum, int blockNum, bool isEncrypted, const unsigned char \*IC\_KEY, unsigned char \*blockData) (1.5.0)

Description：Write block into current card.

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| int sectorNum | Sector number | Integer, [0, 15] |
| int blockNum | Block Index | [0, 2], it can’t be 0 while sectorNum is 0 |
| bool isEncrypted | Wheather the block is encrypted |  |
| unsigned char \*IC\_KEY | Block IC key |  |
| unsigned char \*blockData | Block data write into card | Fixed 16 bits |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Fail |
| [More information](#_Error Code) | |

## int CE\_ParseResData\_V2( const char \*hotelInfo, unsigned char \*sectorData, int sectorCount, int timezoneRawOffset, char \*\*HotelArray) (V2, 1.6.0)

Description：Parse the data of the card. For 3rd party encoders, only used for locks support start time.

Parameter：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Value | IN/OUT |
| const char \*hotelInfo | Assigned or get from server |  | IN |
| unsigned char \*sectorData | The data of sector (except sector 0). | Mutiples of 48 bytes. At least 192 bytes | IN |
| int sectorCount | Number of sectors contained in sectorData | [4, 15] | IN |
| int timezoneRawOffset | Time zone of lock | Eg: GMT+0800 - 28800 | IN |
| char \*\*HotelArray | The data from the card | JSON string | IN |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| [More information](#_Error Code) | |

HotelArray：

|  |  |
| --- | --- |
| 值 | 意义 |
| count | hotelInfo length |
| hotelInfo | Card Info |

HotelArray.hotelInfo参数说明：

|  |  |
| --- | --- |
| 值 | 意义 |
| buildNo | Building Number |
| floorNo | Floor Number |
| mac | Lock MAC |
| startDate | Start date |
| endDate | End date |
| timeMark | Time mark  0 - Enforcement  1-255 Weight  卡里的该时间标记必须循环大于等于锁里面的时间标记10以内（该规则认为 1-9 均大于255）该条记录才判定为有效，如果锁判断卡里时间标记有效则会将卡里的标记更新到锁内 |
| allowLockOut | If allow lock out, 1 or 0 |
| cycleInfo(?) | Cycle info |

hotelArray.hotelInfo.cycleInfo参数说明：

|  |  |
| --- | --- |
| 值 | 意义 |
| cycleType | Cycle type， 1 -weekly，2 -daily，3 -monthly |
| cycleDays | Cycle days.  cycleType=1: return array between [1-7]  eg. [1,4,6] means Mon. Thur. Sat.  cycleType=2: no return  cycleType=3: return array between [1-31]  Eg. [3,28]. |
| startTime | Cycle start minute  Eg: 75 means 1:15 |
| endTime | Cycle end minute  Eg: 90 means 1:30 |

## int CE\_GenerateSectorData\_V2( const char \*hotelInfo, unsigned char \*sectorData, int sectorCount, unsigned char TimeMark, int buildNo, int floorNo, const char \*mac, unsigned long startDate, unsigned long endDate, bool allowLockOut, int timezoneRawOffset, bool isCycle, unsigned char cycleType, int cycleCount, unsigned char \*cycleDays, unsigned int startTime, unsigned int endTime) (V2, 1.6.0)

Description：Write one record into card. It is for 3rd party encoders. Please make sure you have configured the server url before calling this function, only used for locks support start time.

Parameter：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Value | IN/OUT |
| const char \*hotelInfo | Assigned or get from server |  | IN |
| unsigned char \*sectorData | The data of sector (except sector 0). | Mutiples of 48 bytes. At least 192 bytes | IN, OUT |
| int sectorCount | Number of sectors contained in sectorData | [4, 15] | IN |
| unsigned char TimeMark | Time Mark | 0 - Enforcement  1-255 Weight  卡里的该时间标记必须循环大于等于锁里面的时间标记10以内（该规则认为 1-9 均大于255）该条记录才判定为有效，如果锁判断卡里时间标记有效则会将卡里的标记更新到锁内 | IN |
| int buildNo | Building Number | 0 ~ 254 | IN |
| int floorNo | Floor Number | 0 ~ 255 | IN |
| const char \*mac | Lock MAC | Mac address without colon，end with’\0’，total length 13 characters | IN |
| unsigned long startDate | Timestamp of start time | In second | IN |
| unsigned long endDate | Timestamp of expired time | In second | IN |
| bool allowLockOut | If allow lock out |  | IN |
| int timezoneRawOffset | Time zone of lock | Eg: GMT+0800 - 28800 | IN |
| bool isCycle |  |  | IN |
| unsigned char cycleType [IN] | Cycle Type | 1 -weekly，2 -daily，3 -monthly | IN |
| int cycleCount [IN] | Length of cycleDays | [1,9] | IN |
| unsigned char \*cycleDays [IN] | Cycle Days | cycleType=1: return array between [1-7]  eg. [1,4,6] means Mon. Thur. Sat.  cycleType=2: no return  cycleType=3: return array between [1-31]  Eg. [3,28]. | IN |
| unsigned int startTime | Start cycle second | [0,1438], A multiple of 15 | IN |
| unsigned int endTime |  | [1,1439], A multiple of 15 | IN |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 2 | Bad parameter |
| [More information](#_Error Code) | |

Description of some parameters：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| data type | buildNo | floorNo | mac | Description |
| master card | 0 | 0 | “000000000000” | Offline operations are not allowed |
| building card | √ | 0 | “000000000000” |
| floor card | √ | √ | “000000000000” |
| room card | √ | √ | √ | Offline operations are allowed |

## int CE\_GenerateClearCardData\_V2(const char \*hotelInfo, unsigned char \*sectorData, int sectorCount) (V2, 1.6.0)

Description：Generate data to clear card. It is for 3rd party encoders, only used for locks support start time.

Parameter：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Value | IN/OUT |
| const char \*hotelInfo | Assigned or get from server |  | IN |
| unsigned char \*sectorData | The data of sector (except sector 0). | **Must be whole used sector info.**  Mutiples of 48 bytes. At least 192 bytes | IN, OUT |
| int sectorCount | Number of sectors contained in sectorData | [4, 15] | IN |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Fail |
| [More information](#_Error Code) | |

## int CE\_ConfigProxyServer(bool useHTTPS, const char \*pszAgent, const char \*serverName, const char \*IPAddress, int serverPort, bool useProxy, unsigned long dwAuthSchemeType, const char \*userName, const char \*password, const char \*proxyServer, const char \*path) (1.5.2)

Description：Config server and proxy.

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| bool useHTTPS | Wheather use HTTPS. |  |
| const char \*pszAgent | Agent name. | Less than 300 characters. |
| const char \*serverName | Server name. | Less than 100 characters |
| const char \*IPAddress | IP address. | Less than 50 characters |
| int serverPort | Server port |  |
| bool useProxy | Wheather use proxy. |  |
| unsigned long dwAuthSchemeType | <https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpqueryauthschemes> | Default: WINHTTP\_AUTH\_SCHEME\_BASIC |
| LPCSTR userName | Proxy username | Less than 100 characters |
| LPCSTR password | Proxy passcode | Less than 100 characters |
| LPCSTR proxyServer | Proxy server address | Less than 120 characters |
| LPCSTR path | Interface path | Like ”/v3/hotel/getServerDatetime” |

Return：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success |
| 1 | Fail |
| 11 | Failed to request server (You can learn more details from log) |
| 12 | Server returning failed |
| [More information](#_Error Code) | |

## bool CE\_GetLastHttpError(unsigned long \*execErrorCode, unsigned long \*httpErrorCode, unsigned long \*httpStatus) (1.5.2)

Description：Get last http request information.

Parameter：

|  |  |  |
| --- | --- | --- |
| Name | Description | Value |
| unsigned long \*execErrorCode | DLL request error. |  |
| unsigned long \*httpErrorCode | Win HTTP/HTTPS error. | https://docs.microsoft.com/en-us/windows/win32/winhttp/error-messages |
| unsigned long \*httpStatus | Http status. | -1 or 0 means failed. |

Detail from execErrorCode：

|  |  |
| --- | --- |
| Value | Description |
| 0 | Success. |
| 1 | Proxy params error. |
| 2 | Server params error. |
| 3 | Request path error. |
| 4 | Busy. |
| 5 | Open session failed. |
| 6 | Http connect failed. |
| 7 | Init request failed. |
| 8 | HTTPS config failed. |
| 9 | Headers config failed. |
| 10 | Proxy config failed. |
| 11 | Send request error. |
| 12 | Receive response error. |
| 13 | Check response lenth failed. |
| 14 | Response error, not confirmed. |
| 15 | Out of memory. |
| 16 | Get response data failed. |
| 17 | Data parse error. |

Return：

|  |  |
| --- | --- |
| Value | Description |
| TRUE | Success |
| FALSE | Fail |

# FAQ

## What are the possible causes for 106 error code?

1. It is an uninitialized card. Please initialize it first.
2. It is a card initialized in another hotel or another hotel account. Card can only work in hotel from which it was initialized.
3. You are trying to create a construction card from an initialized hotel card.
4. The sector configuration of the card and card encoder are not the same.

## Why failed to open door with card?

1. Wrong lock
2. The lock time is not correct
3. Configuration of the card and lock are not the same
4. The card has been reported loss

## Why error code 3/4/5 ?

Failed to communicate with card encoder. Please re-plug the cable and try again.

## Why error code 13 ?

1. Hotelinfo expired
2. Hotelinfo has been modified

## Why error code 101 ?

The card may not be put on the right position

## Why got error code 1 when connect to card encoder?

1. The card encoder has not been plugged in, or the comm is not correct
2. The card encoder driver has not been installed
3. The card encoder has been occupied another software