

K720/K750 Function Interface Operation Guide

I.Port Operation Function

1.Open communication port-----	3
2.Close communication port-----	3

II.Basic Operation Function

1.Get device firmware version-----	4
2.Get device address-----	4
3.Query device status-----	4
4.Card move operation-----	5

III.Contactless Card Operation

1.S50 Card operation-----	6
Search card-----	6
Get card number-----	6
Key Check-----	6
Read block data-----	6
Write block data-----	6
Initialization data-----	7
E-wallet value added-----	7
E-wallet impairment-----	7
Stop activation-----	7
2.S70 Card operation-----	7
Stop activation-----	7
Search card-----	7
Get card number-----	7
Key Check-----	8
Read block data-----	8
Write block data-----	8
Initialization data-----	8
E-wallet value added-----	9
E-wallet impairment-----	9
Stop activation-----	9
3.UL Card operation-----	9
Search card-----	9
Get card number-----	9
Read block data-----	9
Write block data-----	10
Stop activation-----	10
4.15693 Card operation-----	10
Get Uid-----	10
Read block data-----	10
Write block data-----	10

Select card-----	11
Get card information-----	11
Read security bit data-----	11
Write DSFID data-----	11
Write AFI data-----	12
lock block data-----	12
Lock AFI data-----	12
Lock DSFID data-----	12
5.TypeA Card operation-----	12
Activation-----	12
Apu command interaction-----	13

IV.Extension interface

1.Get parameter setting information-----	14
2.Move card interface-----	14
3.Get device status-----	15

V.Error Code

1.Error code-----	16
-------------------	----

I. Port Operation Function

1.Open communicationport

a)Public synchronized int K720_CommOpen(String StrPort);

Function:Open serial port, default baud rate"9600

Parameter: **【in】** Port port to be opened, eg.,open com1 , then "Port will save "com1.

Return value: right=0, error= not 0, For details, please refer to the bottom error code

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

b)Public synchronized int K720_CommOpenWithBaud(String StrPort,int Baudate);

Function:Open serial port,and returns the port handle

Parameter: **【in】** Port connected port number , such as"/dev/ttyS0"

【in】 Baudate connected band rate , such as 9600

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

2.Close the communication port

Public synchronized int K720_CommClose();

Function: close current serial port

parameter: none

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

II.Basic Operation Function

1.Get device firmware version

Public synchronized int K720_GetDllVersion(String[] strVersion);

Function: Get the DLL version number

parameter: **【out】** strVersion Return the dll version number, such as "K7X0_DllV20210322"

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_GetVersion(byte MacAddr, byte[] Version, String[] RecordInfo);

Function: Get device firmware version

parameter: **【in】** MacAddr Device address
【out】 Version Return version information
【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

2.Get device address

Public synchronized int K720_AutoTestMac(byte MacAddr, String[] RecordInfo);

Function: Get device address

parameter: **【in】** MacAddr Device address **【0x00~0x0F】**
【out】 RecordInfo Return data interaction record

Note: call reference code

```
for(int i = 0; i < 16; i++){  
    Int nRet = K720_AutoTestMac((byte)i, RecordInfo);  
    If(nRet == 0){  
        //Successfully obtained the device address  
        break  
    }  
}
```

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

3.Query device status

Public synchronized int K720_Query(byte MacAddr, byte[] StateInfo, String[] RecordInfo);

Function: Query device status [Not recommended, the difference from K720_SensorQuery is that one less byte is returned]

parameter: **【in】** MacAddr Device address
【out】 StateInfo Return device status
【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_SensorQuery(byte MacAddr, byte[] StateInfo, String[] RecordInfo);

Function: Query device status

parameter: **【in】** MacAddr Device address
【out】 StateInfo Return device status
【out】 RecordInfo Return data interaction record

Note: Please refer to the document Sensor Status Code.doc for the return status code

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

4.Card move operation

Public synchronized int K720_SendCmd(byte MacAddr, byte[] p_Cmd, int CmdLen, String[] RecordInfo);

Function: Device move operation function

parameter: **【in】** MacAddr Device address

【in】 p_Cmd Command parameters

p_Cmd = "DC" Indicates moving to the pickup location

p_Cmd = "CP" Indicates recycling card to recycling box

p_Cmd = "DB" Indicates that the card is recycled to the card issuing box [K750 exclusive command]

p_Cmd = "RS" Indicates card dispenser reset

p_Cmd = "BF" or "FC8" means front-end card

p_Cmd = "BG" Indicates that the front end is prohibited from entering the card [invalidate "BF" or "FC8"]

p_Cmd = "FC6" Indicates moving to the sensor 2 position

p_Cmd = "FC7" Indicates moving to the card reading position

p_Cmd = "FC4" Indicates to move to the card-receiving position [requires force to pull out the card]

p_Cmd = "FC0" Indicates moving to the outside of the bayonet [drop from the bayonet]

p_Cmd = "LP"+0 means setting the LED to be always on, where 0 means the flashing frequency [0 is 0 in hexadecimal, not 0 in ASCII, pay attention to the distinction]

P_Cmd = "LF" means set LED off

P_Cmd = "BE" means set beeping

P_Cmd = "BD" means stop beeping

P_Cmd = "FD0" Indicates that the front-end automatic induction card enters

P_Cmd = "FD1" Indicates that it is necessary to execute the command through "BF" or "FC8" and enter the card at the front end

P_Cmd = "FD2" Indicates that the reset operation is performed, and the channel card does not act.

P_Cmd = "FD3" Indicates that the reset operation is performed, and the channel card is returned to the card issuing box

P_Cmd = "FD4" Indicates that the reset operation is performed, and the channel card is returned to the recycling box

【in】 CmdLen Command parameter length [determined by p_Cmd, 2 or 3]

【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

III.Contactless Card Operation

1.S50 Card Operation

Public synchronized int K720_S50DetectCard(byte MacAddr, String[] RecordInfo);

Function: Search S50 card

parameter: **【in】** MacAddr Device address
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S50GetCardID(byte MacAddr, byte[] _CardID, String[] RecordInfo);

Function: Get S50 card number

parameter: **【in】** MacAddr Device address
 【out】 _CardID Returns the card serial number
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S50LoadSecKey(byte MacAddr, byte SectorAddr, byte _KEYType, byte[] _KEY, String[] RecordInfo);

Function: Import S50 [check] key

parameter: **【in】** MacAddr Device address
 【in】 SectorAddr The sector that needs to verify the secret key [00~0F]
 【in】 _KEYType Type of authentication key required
 _KEYType = 0x30Indicates to verify the KeyA key
 _KEYType = 0x31Indicates to verify the KeyB key
 【in】 _KEY key data
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S50ReadBlock(byte MacAddr, byte SectorAddr, byte BlockAddr, byte[] _BlockData, String[] RecordInfo);

Function: Read S50 read block data

parameter: **【in】** MacAddr Device address
 【in】 SectorAddr Sector address **【0x00~0x0F】**
 【in】 BlockAddr Block address [0x00~0x03]
 【out】 _BlockData Return block data [16 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S50WriteBlock(byte MacAddr, byte SectorAddr, byte BlockAddr, byte[] _BlockData, String[] RecordInfo);

Function: Write S50 write block data

parameter: **【in】** MacAddr Device address
 【in】 SectorAddr Sector address **【0x00~0x0F】**
 【in】 BlockAddr Block address [0x00~0x03]
 【in】 _BlockData Write block data [16 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S50InitValue(byte MacAddr, byte SectorAddr, byte BlockAddr, byte[] _Data, String[] RecordInfo);

Function: Initialization S50 data

parameter: **【in】** MacAddr Device address
 【in】 SectorAddr Sector address **【0x00~0x0F】**
 【in】 BlockAddr Block address [0x00~0x03]
 【in】 _Data Initialize block data [4 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S50Increment(byte MacAddr, byte SectorAddr, byte BlockAddr, byte[] _Data, String[] RecordInfo);

Function: S50 e-wallet value added

parameter: **【in】** MacAddr Device address
 【in】 SectorAddr Sector address **【0x00~0x0F】**
 【in】 BlockAddr Block address [0x00~0x03]
 【in】 _Data Value-added data [4 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S50Decrement(byte MacAddr, byte SectorAddr, byte BlockAddr, byte[] _Data, String[] RecordInfo);

Function: S50 e-wallet impairment

parameter: **【in】** MacAddr Device address
 【in】 SectorAddr Sector address **【0x00~0x0F】**
 【in】 BlockAddr Block address [0x00~0x03]
 【in】 _Data Impairment data [4 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

2.S70 Card operation

Public synchronized int K720_S50Halt(byte MacAddr, String[] RecordInfo);

Function: S70 deactivated

parameter: **【in】** MacAddr Device address
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S70DetectCard(byte MacAddr, String[] RecordInfo);

Function: Search S70 card

parameter: **【in】** MacAddr Device address
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S70GetCardID(byte MacAddr, byte[] _CardID, String[] RecordInfo);

Function: Get S70 card number

parameter: **【in】** MacAddr Device address

【out】_CardID Returns the card serial number

【out】RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S70LoadSecKey(byte MacAddr, byte SectorAddr, byte _KEYType, byte[] _KEY, String[] RecordInfo);

Function: Import S70 [check] key

parameter: **【in】MacAddr** Device address

【in】_SectorAddr The sector that needs to verify the secret key [00~0x27]

【in】_KEYType Type of authentication key required

_KEYType = 0x30 Indicates to verify the KeyA key

_KEYType = 0x31 Indicates to verify the KeyB key

【in】_KEY key data

【out】RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S70ReadBlock(byte MacAddr, byte SectorAddr, byte BlockAddr, byte[] _BlockData, String[] RecordInfo);

Function: Read S70 read block data

parameter: **【in】MacAddr** Device address

【in】SectorAddr Sector address [0x00~0x27]

【in】BlockAddr Block address [0x00~0x03]

【out】_BlockData Return block data [16 bytes]

【out】RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S70WriteBlock(byte MacAddr, byte SectorAddr, byte BlockAddr, byte[] _BlockData, String[] RecordInfo);

Function: Write S70 write block data

parameter: **【in】MacAddr** Device address

【in】SectorAddr Sector address [0x00~0x27]

【in】BlockAddr Block address [0x00~0x03]

【in】_BlockData Write block data [16 bytes]

【out】RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S70InitValue(byte MacAddr, byte SectorAddr, byte BlockAddr, byte[] _Data, String[] RecordInfo);

Function: Initialization S70 data

parameter: **【in】MacAddr** Device address

【in】SectorAddr Sector address [0x00~0x27]

【in】BlockAddr Block address [0x00~0x03]

【in】_Data Initialize block data [4 bytes]

【out】RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S70Increment(byte MacAddr, byte SectorAddr, byte BlockAddr, byte[] _Data, String[] RecordInfo);

Function: S70 e-wallet value added

parameter: **【in】** MacAddr Device address
 【in】 SectorAddr Sector address [0x00~0x27]
 【in】 BlockAddr Block address [0x00~0x03]
 【in】 _Data Value-added data [4 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S70Decrement(byte MacAddr, byte SectorAddr, byte BlockAddr, byte[] _Data, String[] RecordInfo);

Function: S70 electronic wallet impairment

parameter: **【in】** MacAddr Device address
 【in】 SectorAddr Sector address [0x00~0x27]
 【in】 BlockAddr Block address [0x00~0x03]
 【in】 _Data Impairment data [4 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_S70Halt(byte MacAddr, String[] RecordInfo);

Function: S70 deactivated

parameter: **【in】** MacAddr Device address
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

3.UL Card operation

Public synchronized int K720_ULDetectCard(byte MacAddr, String[] RecordInfo);

Function: Search UL card

parameter: **【in】** MacAddr Device address
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_ULGetCardID(byte MacAddr, byte[] _CardID, String[] RecordInfo);

Function: Get UL card number

parameter: **【in】** MacAddr Device address
 【out】 _CardID Returns the card serial number
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_ULReadBlock(byte MacAddr, byte SectorAddr, byte[] _BlockData, String[] RecordInfo);

Function: Read UL block data

parameter: **【in】** MacAddr Device address
 【in】 SectorAddr sector address
 【out】 _BlockData Return block data [sixteen bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_ULWriteBlock(byte MacAddr, byte SectorAddr, byte[] _BlockData, String[] RecordInfo);

Function: Write UL block data

parameter: **【in】** MacAddr Device address
 【in】 SectorAddr sector address
 【in】 _BlockData Write block data [16 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_ULHalt(byte MacAddr, String[] RecordInfo);

Function: UL deactivation

parameter: **【in】** MacAddr Device address
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

4.15693 Card operation

Public synchronized int K720_15693GetUid(byte MacAddr, byte[] Uid, String[] RecordInfo);

Function: Get 15693 card Uid

parameter: **【in】** MacAddr Device address
 【out】 UID Return Uid data [8 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_15693ReadData(byte MacAddr,boolean bUid,byte[] Uid,byte BlockAddr, byte BlockLen, byte[] _BlockData,byte[] ReadBlockLen, String[] RecordInfo);

Function: Read 15693 Card block data

parameter: **【in】** MacAddr Device address
 【in】 bUid Whether to bring Uid [false means without uid, true means with uid]
 【in】 Uid Uid data [8 bytes]
 【in】 BlockAddr block address
 【in】 BlockLen block data length
 【out】 _BlockData block data
 【out】 ReadBlockLen Returns the length of the block data read
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_15693WriteData(byte MacAddr,boolean bUid,byte[] Uid,byte BlockAddr, byte BlockLen, byte[] _BlockData,byte [] WriteBlockLen, String[] RecordInfo);

Function: Write 15693 Card block data

parameter: **【in】** MacAddr Device address
 【in】 bUid Whether to bring Uid [false means without uid, true means with uid]
 【in】 Uid Uid data [8 bytes]
 【in】 BlockAddr block address
 【in】 BlockLen block data length
 【in】 _BlockData block data

【out】 WriteBlockLen Returns the length of the block data written

【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_15693ChooseCard(byte MacAddr,boolean bUid,byte[] Uid, String[] RecordInfo);

Function: Select 15693 card

parameter: **【in】 MacAddr** Device address

【in】 bUid Whether to bring Uid [false means without uid, true means with uid]

【in】 Uid Uid data [8 bytes]

【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_15693GetMessage(byte MacAddr,boolean bUid,byte[] Uid,byte[] Message, String[] RecordInfo);

Function: Get 15693 card information

parameter: **【in】 MacAddr** Device address

【in】 bUid Whether to bring Uid [false means without uid, true means with uid]

【in】 Uid Uid data [8 bytes]

【out】 Message Return data

【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_15693ReadSafeBit(byte MacAddr,boolean bUid,byte[] Uid,byte BlockAddr, byte BlockLen,byte[] ReadBlockLen,byte[] BlockLockStatus, String[] RecordInfo);

Function: Read 15693 card security bit data

parameter: **【in】 MacAddr** Device address

【in】 bUid Whether to bring Uid [false means without uid, true means with uid]

【in】 Uid Uid data [8 bytes]

【in】 BlockAddr block address

【in】 BlockLen block data length

【out】 ReadBlockLen Returns the data length in safe bits

【out】 BlockLockStatus return security bit data

【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_15693WriteDSFID(byte MacAddr,boolean bUid,byte[] Uid,byte WriteBit, String[] RecordInfo);

Function: write 15693 card DSFID data

parameter: **【in】 MacAddr** Device address

【in】 bUid Whether to bring Uid [false means without uid, true means with uid]

【in】 Uid Uid data [8 bytes]

【in】 WriteBit data input

【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_15693WriteAFI(byte MacAddr,boolean bUid,byte[] Uid,byte WriteBit, String[] RecordInfo);

Function: Write 15693 card AFI data

parameter: **【in】** MacAddr Device address
 【in】 bUid Whether to bring Uid [false means without uid, true means with uid]
 【in】 Uid Uid data [8 bytes]
 【in】 WriteBit data input
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_15693LockBlock(byte MacAddr,boolean bUid,byte[] Uid,byte LockAddress, String[] RecordInfo);

Function: Lock 15693 card block data

parameter: **【in】** MacAddr Device address
 【in】 bUid Whether to bring Uid [false means without uid, true means with uid]
 【in】 Uid Uid data [8 bytes]
 【in】 LockAddress block data address
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_15693LockAFI(byte MacAddr,boolean bUid,byte[] Uid, String[] RecordInfo);

Function: Lock 15693 card AFI data

parameter: **【in】** MacAddr Device address
 【in】 bUid Whether to bring Uid [false means without uid, true means with uid]
 【in】 Uid Uid data [8 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_15693LockDSFID(byte MacAddr,boolean bUid,byte[] Uid, String[] RecordInfo);

Function: Lock 15693 card DSFID data

parameter: **【in】** MacAddr Device address
 【in】 bUid Whether to bring Uid [false means without uid, true means with uid]
 【in】 Uid Uid data [8 bytes]
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

5.TypeA Card operation

Public synchronized int K720_CPUCardPowerOn(byte MacAddr, byte [] szATR, String[] RecordInfo);

Function: Power-on activation of TypeA card

parameter: **【in】** MacAddr Device address
 【out】 szATR Return answer-to-reset data
 【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_CPUAPDU(byte MacAddr, byte SCH, int _dataLen, byte[] _APDUData, byte[] RCH, byte[] _exData, int [] _exdataLen, String[] RecordInfo);

Function: TypeA card Adu command interaction

parameter: 【in】 MacAddr Device address

【in】 SCH Reserved for use, the default is 0

【in】 _datalen Send Adu command length

【in】 _APDUData Send Adu command length

【out】 RCH Reserved for use, a byte pointer can be passed in by default

【out】 _exData Adu command response data

【out】 _exdataLen Adu Command response data length

【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

IV.Extension interface

1.Get parameter setting information

Public synchronized int K720_CheckSetting(byte MacAddr, byte[] StateInfo, String[] RecordInfo);

Function: Get parameter setting information

parameter:

【in】 MacAddr Device address
【out】 StateInfo Return parameter information
 StateInfo[0] = 0x01 Indicates that the front-end card needs to be entered through the "BF" or "FC8" command
 StateInfo[0] = 0x02 Indicates auto-sensing front-end card entry
 StateInfo[1] = 0x01 Indicates reset without action
 StateInfo[1] = 0x02 Indicates that if there is a card in the reset channel, it is returned to the card issuing box
 StateInfo[1] = 0x03 Indicates that if there is a card in the reset channel, it is recycled to the recycling box
【out】 RecordInfo Return data interaction record

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

*/*The following interfaces are encapsulated interfaces, which include operations such as card movement and query. The function return value will only be returned after the operation is completed or timed out. The precondition for calling the following interfaces is that the device has been established and the Device address has been obtained*/*

2.Move card interface

Public synchronized int K720_MoveCard(byte _PM);

Function: Card move operation [This interface will return after the action is completed or timed out, which is the encapsulated action flow]

parameter: **【in】** _PM _PM = 0x30 Indicates that the card box is issued to the card reading position
 _PM = 0x31 Indicates to move the card to the pick-up position
 _PM = 0x32 Indicates that the card is moved out of the bayonet
 _PM = 0x33 Indicates moving the card to the recycling box
 _PM = 0x34 Indicates that the card is moved to the card issuing box

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

Public synchronized int K720_RetainToCardBox(boolean flag, int EnterTimeOut, int TimeOut);

Function: Recycle the card to the card issuing box [the difference from K720_MoveCard is that the reserved parameter can be set whether the card is inserted in the front or not]

parameter: **【in】** flag Do you need to enter the card from the front
 flag = true means that the front-end card entry function will be performed inside the interface, and the card can be inserted within the EnterTimeOut time, otherwise it will return to the card entry timeout
 flag=false means that it is necessary to ensure that there is a card in the channel before recycling the card to the card issuing box, otherwise it will return no card error
【in】 EnterTimeOut Front-end card entry timeout time

【in 】 TimeOut *The timeout period for recycling to the card box [does not include the timeout period for incoming cards]*

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

3.Get device status

Public synchronized int K720_CheckCardPosition(byte [] DeviceStatus, byte [] TransportStatus, byte [] CardBoxStatus, byte [] RetainBoxStatus);

Function: Query device status [same function as K720_SensorQuery, the difference is that the returned value is processed]

parameter: **【out 】** DeviceStatus *Return device status*

DeviceStatus = 0x30 Indicates that the current action is idle

DeviceStatus = 0x31 Indicates equipment failure [reset operation is required]

DeviceStatus = 0x32 Indicates that the card preparation failed

DeviceStatus = 0x33 Indicates that the card is being issued

DeviceStatus = 0x34 Indicates that the card is being collected

DeviceStatus = 0x35 Indicates an error in issuing the card

DeviceStatus = 0x36 Indicates a card error

【out 】 TransportStatus *Return channel status*

TransportStatus= 0x30 Indicates that the channel has overlapping cards

TransportStatus= 0x31 Indicates that the channel is blocked

TransportStatus= 0x32 Indicates that there is a card in the card reader

TransportStatus= 0x33 Indicates that the channel has no card

TransportStatus= 0x34 Indicates that there is a card in the front

【out 】 CardBoxStatus *Return to card box status*

TransportStatus= 0x30 Indicates that the card box is empty

TransportStatus= 0x31 Indicates that the card box is pre-empty and the card is less

TransportStatus= 0x32 Indicates that the card box has enough cards

TransportStatus= 0x33 Indicates that the card box is pre-full, almost full

TransportStatus= 0x34 Indicates that the card box is full

【out 】 RetainBoxStatus *Return to recycle box status*

RetainBoxStatus= 0x30 Indicates that the recycling box is not full

RetainBoxStatus= 0x31 Indicates that the recycling box is full

Return value: 0 means success, other means failure, please refer to the error code at the bottom for details

V.Error code

```
/[] Card reader communication error code[] /
#define Bad_CommOpen           -101//Port not open
#define Bad_CommSet            -102//The port parameter setting is wrong, (102) also
                                means that the CommSet function is wrong

#define Bad_CommTimeouts       -999
#define Bad_CommQueue          -998
#define Bad_CommClose          -105//Port closed error
#define Bad_Sendbyte           -106//Serial data write failed
#define Bad_SendData_Len       -107//Serial data write failed
#define Bad_ReceiveData_ReadFile -103
#define Bad_ReceiveData_Len    -104
#define Bad_Parameter           -200//Incoming parameter error (beyond definition)


#define CardBoxEmpty            0xA0//The card issuing box is empty and the card cannot
                                be issued
#define CardBoxFull             0xA1//The card issuing box is full, and the cards cannot be
                                returned to the issuing box
#define RetainBoxFull           0xA2//The recycling bin is full and cannot be recycled to
                                the recycling box
#define HaveCard                0xA3//There is a card in the channel, and the card issuing
                                action cannot be performed.
#define HaveNoCard              0xA4//There is no card in the channel, and actions such as
                                bullet card and recycling cannot be performed.
#define MoveCardTimeOut         0xA5//Card movement timeout, including card issuance,
                                card ejection, recycling and other actions
#define EnterCardTimeOut        0xA6//When the card is returned to the card issuing box,
                                the front-end card feeding times out


#define SendCard_Time            15000//Card issuance timeout
#define EjectCard_Time           4000//Card issuance timeout
#define RetainCard_Time          4000//Recycle timeout


//device status
#define IDLE                     0x30//idle
#define FAULT                    0x31//device failure
#define READY_FAILED             0x32//Failed to prepare card
#define SENDINGCARD              0x33//Issuing card
#define RETAININGCARD            0x34//receiving card
#define SENDCARD_FAILED          0x35//Card issuance error
#define RETAINCARD_FAILED        0x36//Card collection error


//Channel status
#define OVERLAP                  0x30//Overlapping Cards
#define JAM                      0x31//Blockage
```

<i>#define MEDIAPRESENT</i>	<i>0x32//There is a card in the area where the card is read</i>
<i>#define MEDIANOTPRESENT</i>	<i>0x33//No card in channel</i>
<i>#define MEDIAENTERING</i>	<i>0x34//card on the front</i>
 <i>//Card box status</i>	
<i>#define CARDBOXEMPTY</i>	<i>0x30//Empty card</i>
<i>#define CARDBOXLESS</i>	<i>0x31//Pre-empty, less card</i>
<i>#define CARDBOXMORE</i>	<i>0x32//Enough card</i>
<i>#define CARDBOXNOTFULL</i>	<i>0x33//Prefilled</i>
<i>#define CARDBOXFULL</i>	<i>0x34//Full</i>
 <i>//Card box status</i>	
<i>#define RETAINBOXNOTFULL</i>	<i>0x30//Not full</i>
<i>#define RETAINBOXFULL</i>	<i>0x31//Full</i>

Timer: 2021/04/21