

K720/K750FunctionInterfaceOperationGuide

I.Port OperationFunction

1.Open communicationport-----	3
2.Closecommunicationport-----	3

II.Basic OperationFunction

1.GettheDLLversionnumber-----	4
2.Get device firmware version-----	4
3.Getdeviceaddress-----	4
4.Querydevice status-----	5
5.Cardmoveoperation-----	5

III.Contactless Card Operation

1.S50 Cardoperation-----	7
Searchcard-----	7
Get cardnumber-----	7
KeyCheck-----	7
Readblockdata-----	8
Writeblockdata-----	8
Initializationdata-----	8
E-walletvalueadded-----	9
E-walletimpairment-----	9
Stop activation-----	9
2.S70 Cardoperation-----	9
Stop activation-----	9
Searchcard-----	9
Get cardnumber-----	10
KeyCheck-----	10
Readblockdata-----	10
Writeblockdata-----	11
Initializationdata-----	11
E-walletvalueadded-----	11
E-walletimpairment-----	12
Stop activation-----	12
3.ULCardoperation-----	12
Searchcard-----	12
Get cardnumber-----	12
Readblockdata-----	13

Writeblockdata-----	13
Stop activation-----	13
4.15693 Cardoperation-----	13
GetUid-----	13
Readblockdata-----	14
Writeblockdata-----	14
Selectcard-----	14
Get cardinformation-----	15
Read securitybitdata-----	15
WriteDSFIDdata-----	15
Write AFIdata-----	16
lockblockdata-----	16
Lock AFIdata-----	16
LockDSFIDdata-----	16
5.TypeACardoperation-----	17
Activation-----	17
Apu commandinteraction-----	17
6.TypeBCardoperation-----	17
Activation-----	17
Apu commandinteraction-----	18
Poweroff-----	18

IV.Extension interface

1.Getparameter settinginformation-----	19
2.Movecardinterface-----	19
3.Getdevicestatus-----	20

V.Error Code

1.Errorcode-----	22
------------------	----

I.Port OperationFunction

1.Open communicationport

HANDLE __stdcallK720_CommOpen(char*Port);

Function:Openserialport, defaultbaudrate“9600, andreturnstheporthandle

Parameter: 【in】 Portconnectedportnumber, suchascom1

Returnvalue: right=0, error= not 0, Fordetails, pleaserefertothebottomerrorcode

Returnvalue:Serial port handle > 0 means success, other meansfailure, please refer totheerror codeatthebottomfordetails

HANDLE __stdcallK720_CommOpenWithBaud(char*Port,unsignedint _data);

Function: Open serial port and returnsthe port handle

Parameter: 【in】 Portconnectedportnumber, suchascom1

【in】 Baudateconnectedbandrate, suchas 9600

Returnvalue:Serial port handle > 0 means success, other meansfailure, please refer totheerror codeatthebottomfordetails

2.Close communicationport

int __stdcallK720_CommClose(HANDLE ComHandle);

Function: close current serial port

Parameter: 【in】 ComHandleSerialporthandle

Returnvalue: 0 means success, other meansfailure, please refer to theerror codeatthebottomfordetails

II. Basic Operation Function

1. Get the DLL version number

`int __stdcall K720_GetSysVersion(HANDLE ComHandle, char*strVersion);`

Function: Get the DLL version number

Parameter: **【in】** ComHandleSerialporthandle

【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

2. Get device firmware version

`int __stdcall K720_GetVersion(HANDLE ComHandle, BYTE MacAddr, BYTE Version[20], char*RecordInfo);`

Function: Get device firmware version

Parameter: **【in】** ComHandleSerialporthandle

【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

3. Get device address

`int __stdcall K720_AutoTestMac(HANDLE ComHandle, unsigned char MacAddr, char*RecordInfo);`

Function: Get device address

Parameter: **【in】** ComHandleSerialporthandle

【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(ComHandle,(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

4. Query device status

`int __stdcall K720_Query(HANDLE ComHandle, BYTE MacAddr, BYTE StateInfo[3], char* RecordInfo);`

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

Parameter: 【in】 ComHandle Serial port handle

【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

`int __stdcall K720_SensorQuery(HANDLE ComHandle, BYTE MacAddr, BYTE StateInfo[4], char* RecordInfo);`

Function: Query device status

Parameter: 【in】 ComHandle Serial port handle

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

Note: Please refer to the document [Sensor StatusCode.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

5. Card move operation

`1. int __stdcall K720_SendCmd(HANDLE ComHandle, unsigned char MacAddr, char* p_Cmd, int CmdLen, char* RecordInfo);`

Function: Device move operation function

Parameter: 【in】 ComHandle Serial port handle

【in】 MacAddr	Device address
【in】 p_Cmd	Command parameter

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 of

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

`int __stdcall K720_S50DetectCard(HANDLE ComHandle, BYTE MacAddr, char*RecordInfo);`

Function: Search S50 card

Parameter: **【in】** ComHandle Serial port handle

【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

`int __stdcall K720_S50GetCardID(HANDLE ComHandle, BYTE MacAddr, BYTE* _CardID, char*RecordInfo);`

Function: Get S50 card number

Parameter: **【in】** ComHandle Serial port handle

【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

`int __stdcall K720_S50LoadSecKey(HANDLE ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE _KEYType, BYTE* _KEY, char*RecordInfo);`

Function: Import S50 [check] key

Parameter: **【in】** ComHandle Serial port handle

【in】 MacAddr Device address

【in】 SectorAddr This sector that needs to verify the secret key [00~0F]

【in】 _KEYType Type of authentication key required

_KEYType = 0x30 Indicate to verify the KeyA key

_KEYType = 0x31 Indicate 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

```
int __stdcall K720_S50ReadBlock(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE BlockAddr, BYTE* _BlockData,
char* RecordInfo);
```

Function: ReadS50 readblockdataParameter: **【in】** ComHandleSerialporthandle

【in】 MacAddr	Device address
【in】 SectorAddr	Sectoraddress 【0x00~0x0F】
【in】 BlockAddr	Blockaddress [0x00~0x03]
【out】 _BlockData	Returnblockdata [16bytes]
【out】 RecordInfo	Return data interaction record

Returnvalue: 0 means success, other meansfailure, please refer to theerror codeatthebottomfordetails

```
int __stdcall K720_S50WriteBlock(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE BlockAddr, BYTE* _BlockData,
char* RecordInfo);
```

Function: WriteS50writeblockdata

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddr	Deviceaddress
【in】 SectorAddr	Sectoraddress 【0x00~0x0F】
【in】 BlockAddr	Blockaddress 【0x00~0x03】
【in】 _BlockData	Writeblockdata [16bytes]
【out】 RecordInfo	Returndatainteractionrecord

Returnvalue: 0 means success, other meansfailure, please refer to theerror codeatthebottomfordetails

```
int __stdcall K720_S50InitValue(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE BlockAddr, BYTE* _Data,
char* RecordInfo);
```

Function: InitializationS50 data

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddr	Device address
【in】 SectorAddr	Sectoraddress 【0x00~0x0F】
【in】 BlockAddr	Blockaddress [0x00~0x03]
【in】 _Data	Initializeblockdata [4bytes]
【out】 RecordInfo	Return data interaction record

Returnvalue: 0 means success, other meansfailure, please refer to theerror codeatthebottomfordetails

```
int __stdcall K720_S50Increment(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE BlockAddr, BYTE* _Data,
char* RecordInfo);
```

Function: S50 e-wallet value added

Parameter: **【in】** ComHandle Serial port handle

【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

```
int __stdcall K720_S50Decrement(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE BlockAddr, BYTE* _Data,
char* RecordInfo);
```

Function: S50 e-wallet impairment

Parameter: **【in】** ComHandle Serial port handle

【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

```
int __stdcall K720_S50Halt(HANDLE ComHandle, BYTE MacAddr, char* RecordInfo);
```

Function: S70 deactivated

Parameter: **【in】** ComHandle Serial port handle

【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

```
int __stdcall K720_S70DetectCard(HANDLE ComHandle, BYTE MacAddr,
char* RecordInfo);
```

Function: Search S70 card

Parameter: **【in】** ComHandle Serial port handle

【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

```
int __stdcall K720_S70GetCardID(HANDLE
ComHandle, BYTE MacAddr, BYTE *_CardID, char*RecordInfo);
```

Function: GetS70card number

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddr Device address
【out】 _CardID Returns the card serialnumber
【out】 RecordInfo Return data interaction record

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

```
int __stdcall K720_S70LoadSecKey(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE _KEYType, BYTE *_KEY,
char*RecordInfo);
```

Function: ImportS70 [check] key

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddr Device address
【in】 _SectorAddr Thesector that needstoverifythesecretkey [00~0x27]
【in】 _KEYType Type of authentication key required
 _KEYType = 0x30 IndicatestoverifytheKeyAkey
 _KEYType = 0x31 IndicatestoverifytheKeyBkey
【in】 _KEY keydata
【out】 RecordInfo Return data interaction record

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

```
int __stdcall K720_S70ReadBlock(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE BlockAddr, BYTE *_BlockData,
char*RecordInfo);
```

Function: ReadS70 read block data

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddr Device address
【in】 SectorAddr Sectoraddress [0x00~0x27]
【in】 BlockAddr Blockaddress [0x00~0x03]
【out】 _BlockData Returnblockdata [16bytes]
【out】 RecordInfo Return data interaction record

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

```
int __stdcall K720_S70WriteBlock(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE BlockAddr, BYTE* _BlockData,
char* RecordInfo);
```

Function: WriteS70 writeblockdata

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddr	Device address
【in】 SectorAddr	Sectoraddress [0x00~0x27]
【in】 BlockAddr	Blockaddress [0x00~0x03]
【in】 _BlockData	Writeblockdata[16bytes]
【out】 RecordInfo	Return data interaction record

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

```
int __stdcall K720_S70InitValue(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE BlockAddr, BYTE* _Data,
char* RecordInfo);
```

Function: InitializationS70 data

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddr	Device address
【in】 SectorAddr	Sectoraddress [0x00~0x27]
【in】 BlockAddr	Blockaddress [0x00~0x03]
【in】 _Data	Initializeblockdata [4 bytes]
【out】 RecordInfo	Return data interaction record

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

```
int __stdcall K720_S70Increment(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE BlockAddr, BYTE* _Data,
char* RecordInfo);
```

Function: S70 e-wallet value added

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddr	Device address
【in】 SectorAddr	Sectoraddress [0x00~0x27]
【in】 BlockAddr	Blockaddress [0x00~0x03]
【in】 _Data	Value-added data [4 bytes]
【out】 RecordInfo	Return data interaction record

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

```
int __stdcall K720_S70Decrement(HANDLE
ComHandle, BYTE MacAddr, BYTE SectorAddr, BYTE BlockAddr, BYTE* _Data,
char* RecordInfo);
```

Function: S70e-wallet impairment

Parameter: **【in】** ComHandleSerialporthandle

【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

```
int __stdcall K720_S70Halt(HANDLE ComHandle, BYTE MacAddr, char* RecordInfo);
```

Function: S70 deactivated

Parameter: **【in】** ComHandleSerialporthandle

【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

```
int __stdcall K720_ULDetectCard(HANDLE ComHandle, BYTE MacAddr,
char* RecordInfo);
```

Function: Search UL card

Parameter: **【in】** ComHandleSerialporthandle

【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

```
int __stdcall K720_ULGetCardID(HANDLE
ComHandle, BYTE MacAddr, BYTE* _CardID, char* RecordInfo);
```

Function: Get UL card number

Parameter: **【in】** ComHandleSerialporthandle

【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

```
int __stdcallK720_ULReadBlock(HANDLE
ComHandle,BYTEMacAddr,BYTESectorAddr,BYTE*_BlockData,
char*RecordInfo);
```

Function: ReadULblockdata

Parameter: **【in】** ComHandleSerialporthandle

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

Returnvalue: 0 means success, other meansfailure, please refer to theerror codeatthebottomfordetails

```
int __stdcallK720_ULWriteBlock(HANDLE
ComHandle,BYTEMacAddr,BYTESectorAddr,BYTE*_BlockData,
char*RecordInfo);
```

Function: WriteULblockdata

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddr	Device address
【in】 SectorAddr	sector address
【in】 _BlockData	Writeblockdata[16bytes]
【out】 RecordInfo	Return data interaction record

Returnvalue: 0 means success, other meansfailure, please refer to theerror codeatthebottomfordetails

```
int __stdcallK720_ULHalt(HANDLE ComHandle,BYTEMacAddr, char*RecordInfo);
```

Function: UL deactivation

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddrDeviceaddress	
【out】 RecordInfo	Returndatainteractionrecord

Returnvalue: 0 means success, other meansfailure, please refer to theerror codeatthebottomfordetails

4.15693 Card operation

```
int __stdcallK720_15693GetUid(HANDLE
ComHandle,BYTEMacAddr,BYTEUID[8],char*RecordInfo);
```

Function: Get 15693 cardUid

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddrDeviceaddress	
【out】 UIDReturnUiddata [8 bytes]	
【out】 RecordInfo	Returndatainteractionrecord

Returnvalue: 0 means success, other meansfailure, please refer to theerror codeatthebottomfordetails

```
int __stdcall K720_15693ReadData(HANDLE
ComHandle, BYTE MacAddr, bool bUid, BYTE Uid[8], BYTE BlockAddr, BYTE BlockLen,
BYTE _BlockData[], BYTE*
ReadBlockLen, char* RecordInfo);
```

Function: Read 15693 Card block data

Parameter: **【in】** ComHandle Serial port handle

【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

```
int __stdcall K720_15693WriteData(HANDLE
ComHandle, BYTE MacAddr, bool bUid, BYTE Uid[8], BYTE BlockAddr, BYTE BlockLen,
BYTE _BlockData[], BYTE
*WriteBlockLen, char* RecordInfo);
```

Function: Write 15693 Card block data

Parameter: **【in】** ComHandle Serial port handle

【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

```
int __stdcall K720_15693ChooseCard(HANDLE
ComHandle, BYTE MacAddr, bool bUid, BYTE Uid[8], char* RecordInfo);
```

Function: Select 15693 card

Parameter: **【in】** ComHandle Serial port handle

【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

```
int __stdcall K720_15693GetMessage(HANDLE
ComHandle, BYTE MacAddr, bool bUid, BYTE Uid[8], BYTE Message[15],
char* RecordInfo);
```

Function: Get 15693 card information

Parameter: **【in】** ComHandleSerialporthandle

【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 in interaction record

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

```
int __stdcall K720_15693ReadSafeBit(HANDLE
ComHandle, BYTE MacAddr, bool bUid, BYTE Uid[8], BYTE BlockAddr, BYTE BlockLen,
BYTE* ReadBlockLen, BYTE
BlockLockStatus[], char* RecordInfo);
```

Function: Read 15693 card security bit data

Parameter: **【in】** ComHandleSerialporthandle

【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 Return the data length in safe bits
【out】 BlockLockStatus return security bit data
【out】 RecordInfo Return data in interaction record

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

```
int __stdcall K720_15693WriteDSFID(HANDLE
ComHandle, BYTE MacAddr, bool bUid, BYTE Uid[8], BYTE WriteBit,
char* RecordInfo);
```

Function: write 15693 card DSFID data

Parameter: **【in】** ComHandleSerialporthandle

【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 in interaction record

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

```
int __stdcallK720_15693WriteAFI(HANDLE
ComHandle, BYTE MacAddr, bool bUid, BYTE Uid[8], BYTE WriteBit,
char* RecordInfo);
```

Function: Write 15693 cardAFI data

Parameter: **【in】** ComHandleSerialporthandle

【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 in interaction record

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

```
int __stdcallK720_15693LockBlock(HANDLE
ComHandle, BYTE MacAddr, bool bUid, BYTE Uid[8], BYTE LockAddress,
char* RecordInfo);
```

Function: Lock 15693 card block data

Parameter: **【in】** ComHandleSerialporthandle

【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 in interaction record

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

```
int __stdcallK720_15693LockAFI(HANDLE
ComHandle, BYTE MacAddr, bool bUid, BYTE Uid[8], char* RecordInfo);
```

Function: Lock 15693 cardAFI data

Parameter: **【in】** ComHandleSerialporthandle

【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 in interaction record

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

```
int __stdcallK720_15693LockDSFID(HANDLE
ComHandle, BYTE MacAddr, bool bUid, BYTE Uid[8], char* RecordInfo);
```

Function: Lock 15693 cardDSFID data

Parameter: **【in】** ComHandleSerialporthandle

【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 in 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

`int __stdcall K720_CPUCardPowerOn(HANDLE ComHandle, BYTE MacAddr, BYTE* szATR, char* RecordInfo);`

Function: Power-on activation of Type A card

Parameter: **【in】** ComHandle Serial port handle

【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

`int __stdcall K720_CPUAPDU(HANDLE ComHandle, BYTE MacAddr, BYTE SCH, int _dataLen, BYTE _APDUData[], BYTE* RCH, BYTE _exData[], int* _exdataLen, char* RecordInfo);`

Function: Type A card Apdu command interaction

Parameter: **【in】** ComHandle Serial port handle

【in】 MacAddr	Device address
【in】 SCH	Reserved for use, the default is 0
【in】 _dataLen	Send Apdu command length
【in】 _APDUData	Send Apdu command
【out】 RCH	Reserved for use, a byte pointer can be passed in by default
【out】 _exData	Apdu command response data
【out】 _exdataLen	Apdu 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

5.TypeB Card operation

`int __stdcall K720_CPUTypeBCardPowerOn(HANDLE ComHandle, BYTE MacAddr, BYTE* szATR, char* RecordInfo);`

Function: Power-on activation of Type B card

Parameter: **【in】** ComHandle Serial port handle

【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

```
int __stdcall K720_CPUTypeBAPDU(HANDLE ComHandle, BYTE MacAddr, BYTE SCH, int
_dataLen, BYTE _APDUData[], BYTE* RCH, BYTE _exData[], int* _exdataLen,
char* RecordInfo);
```

Function : Type B card Adu command interaction

Parameter : 【in】 ComHandle Serial port handle

【in】 MacAddr	Device address
【in】 SCH	Reserved for use, the default is 0
【in】 _dataLen	Send Adu command length
【in】 _APDUData	Send Adu command
【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

```
int __stdcall K720_CPUTypeBCardPowerOff(HANDLE
ComHandle, BYTE MacAddr, char* RecordInfo);
```

Function : Type B deactivation

Parameter : 【in】 ComHandle Serial port handle

【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

IV.Extension interface

1.Get parameter settinginformation

```
int __stdcallK720_CheckSetting(HANDLE  
ComHandle,BYTEMacAddr,BYTEStateInfo[], char*RecordInfo);
```

Function: Get parameter setting information

Parameter: **【in】** ComHandleSerialporthandle

【in】 MacAddrDeviceaddress

【out】 StateInfoReturnparameterinformation

StateInfo[0] = 0x01 Indicatesthatthefront-endcardneedstobe
enteredthroughthe"BF"or"FC8"command

StateInfo[0] = 0x02 Indicatesauto-sensingfront-endcardentry

StateInfo[1] = 0x01 Indicatesresetwithoutaction

StateInfo[1] = 0x02Indicatesthatifthereisacardinthereset
channel, it is returned tothecardissuingbox

StateInfo[1] = 0x03Indicatesthatifthereisacardinthereset
channel, it is recycledtotherecyclingbox

【out】 RecordInfoReturndatainteractionrecord

Returnvalue: 0 means success, other meansfailure, please refer to theerror codeatthebottomfordetails

*/*Thefollowing interfaces are encapsulated interfaces,which include operationssuchascard
movement and query.Thefunction returnvaluewill only be returned aftertheoperationis
completed or timed out.The preconditionfor callingthefollowing interfaces isthatthedevice has
been established and the Device addresshasbeenobtained*/*

2.Move cardinterface

```
int __stdcallK720_MoveCard(BYTE _PM);
```

Function: Card move operation [This interfacewill return
aftertheactioniscompletedortimedout,which is theencapsulated actionflow]

Parameter: **【in】** _PM = 0x30Indicatesthatthecardboxisissuedtothecard
_PMreading position

_PM = 0x31 Indicatestomovethecardtothepick-upposition _PM =
0x32 Indicatesthatthecardismovedoutofthebayonet _PM =

0x33 Indicatesmovingthecardtotherecyclingbox

_PM = 0x34 Indicatesthatthecardismovedtothecardissuingbox

Returnvalue: 0 means success, other meansfailure, please refer to theerror codeatthebottomfordetails

int __stdcall K720_RetainToCardBox(BOOL 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 whether 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

1. int

__stdcall 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 Indicatesthatthecardboxispre-empty and thecardisless

TransportStatus= 0x32 Indicatesthatthecardboxhasenough cards

TransportStatus= 0x33 Indicatesthatthecardboxispre-full,almostfull

TransportStatus= 0x34 Indicatesthatthecardboxisfull

【out】 *RetainBoxStatus* Return to recycle box status

RetainBoxStatus= 0x30 Indicatesthattherecyclingboxisnot full

RetainBoxStatus= 0x31 Indicatesthattherecyclingboxisfull

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

4.read barcode data

1.int __stdcall K720_ReadBarcode(HANDLE ComHandle, BYTE MacAddr, BYTE data[], char *RecordInfo);

Function: read barcode data

Parameter: 【in】 ComHandle port handle value

【in】 MacAddr machine address

【out】 data barcode data

【out】 RecordInfo communication data

Returnvalue: 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//Portnotopen
#define Bad_CommSet           -102//Theportparametersettingiswrong,(102) also
                               meansthat theCommSetfunction iswrong

#define Bad_CommTimeouts      -999
#define Bad_CommQueue         -998
#define Bad_CommClose         -105//Portclosederror
#define Bad_Sendbyte          -106//Serialdatawritefailed
#define Bad_SendData_Len      -107//Serialdatawritefailed
#define                       -103
Bad_ReceiveData_ReadFile#define -104
Bad_ReceiveData_Len         -200//Incomingparametererror (beyonddefinition)
#define Bad_Parameter

#define CardBoxEmpty          0xA0//Thecardissuingboxisemptyandthecardcannot
                               beissued
#define CardBoxFull           0xA1//The card issuing box isfull, and
                               thecardscannotbereturned totheissuingbox
#define RetainBoxFull         0xA2//The recycling bin isfull and
                               cannotberecycledtothe recyclingbox
#define HaveCard              0xA3//Thereisacardinthechannel,andthecardissuing
                               action cannot be performed.
#define HaveNoCard            0xA4//Thereisnocardinthechannel,andactionssuchas
                               bullet card and recycling cannot be performed.
#define MoveCardTimeOut       0xA5//Card movement timeout, including card issuance,
                               card ejection, recycling andotheractions
#define EnterCardTimeOut      0xA6//Whenthecardisreturnedtothecardissuingbox,
                               thefront-end cardfeedingtimes out

#define SendCard_Time          15000//Cardissuancetimeout
#define EjectCard_Time         4000//Cardissuancetimeout
#define RetainCard_Time        4000//Recycletimeout

//device status
#define IDLE                   0x30//idle
#define FAULT                  0x31//devicefailure
#define READY_FAILED           0x32//Failedtopreparecard
#define SENDINGCARD            0x33//Issuingcard
#define RETAININGCARD          0x34//receivingcard
#define SENDCARD_FAILED        0x35//Cardissuanceerror
#define RETAINCARD_FAILED      0x36//Cardcollectionerror
```

```

//Channel status
#define OVERLAP          0x30//OverlappingCards
#define JAM              0x31//Blockage
#define MEDIAPRESENT     0x32//Thereisacardintheareawherethecardisread
#define MEDIANOTPRESENT  0x33//Nocardinchannel
#define MEDIAENTERING    0x34//cardonthe front

//Card box status
#define CARDBOXEMPTY     0x30//Emptycard
#define CARDBOXLESS      0x31//Pre-empty, lesscard
#define CARDBOXMORE      0x32//Enoughcard
#define CARDBOXNOTFULL   0x33//Prefilled
#define CARDBOXFULL      0x34//Full

//Card box status
#define RETAINBOXNOTFULL 0x30//Notfull
#define RETAINBOXFULL    0x31//Full

```