	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	1/16

## CRT-571动态库接口函数使用说明

版权所有：深圳市创自技术有限公司

网址：www.china-creator.com

电话：0755-26710345 26710456

动态库说明文档版本：V 1.0

日期：2010-01-28

### WINDOWS 动态库文件说明：

我公司提供WINDOWS 平台下的WINDOWS 32 动态库，文件名称如下：

1. CRT\_571.H            32位动态库的头文件
2. CRT\_571.LIB        32位动态库的输入库
3. CRT\_571.DLL        WINDOWS 32位动态库

### CRT-571可以读写操作的卡片类型：

1. 射频卡，支持ISO/IEC 14443 TYPE A / B协议和MIFARE one(S50, S70,UL)
2. 接触式IC 卡：
  - SLE4428、SLE4442
  - 24C01A、24C02、24C04、24C08、24C16、24C32、24C64
  - 接触式CPU 卡（T=0/T=1）
  - SAM/SIM 卡（T=0/T=1）

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	2/16

目录

1 函数参数组成说明 ..... 3

    1.1 卡机分机地址 ..... 3

    2.1 根据通讯协议来说明函数参数组成如下： ..... 4

2 API 参考说明 ..... 5

    2.1 打开串口函数(默认波特率为 38400bps) ..... 5

    2.2 用指定的波特率打开串口函数 ..... 6

    2.3 关闭串口函数 ..... 7

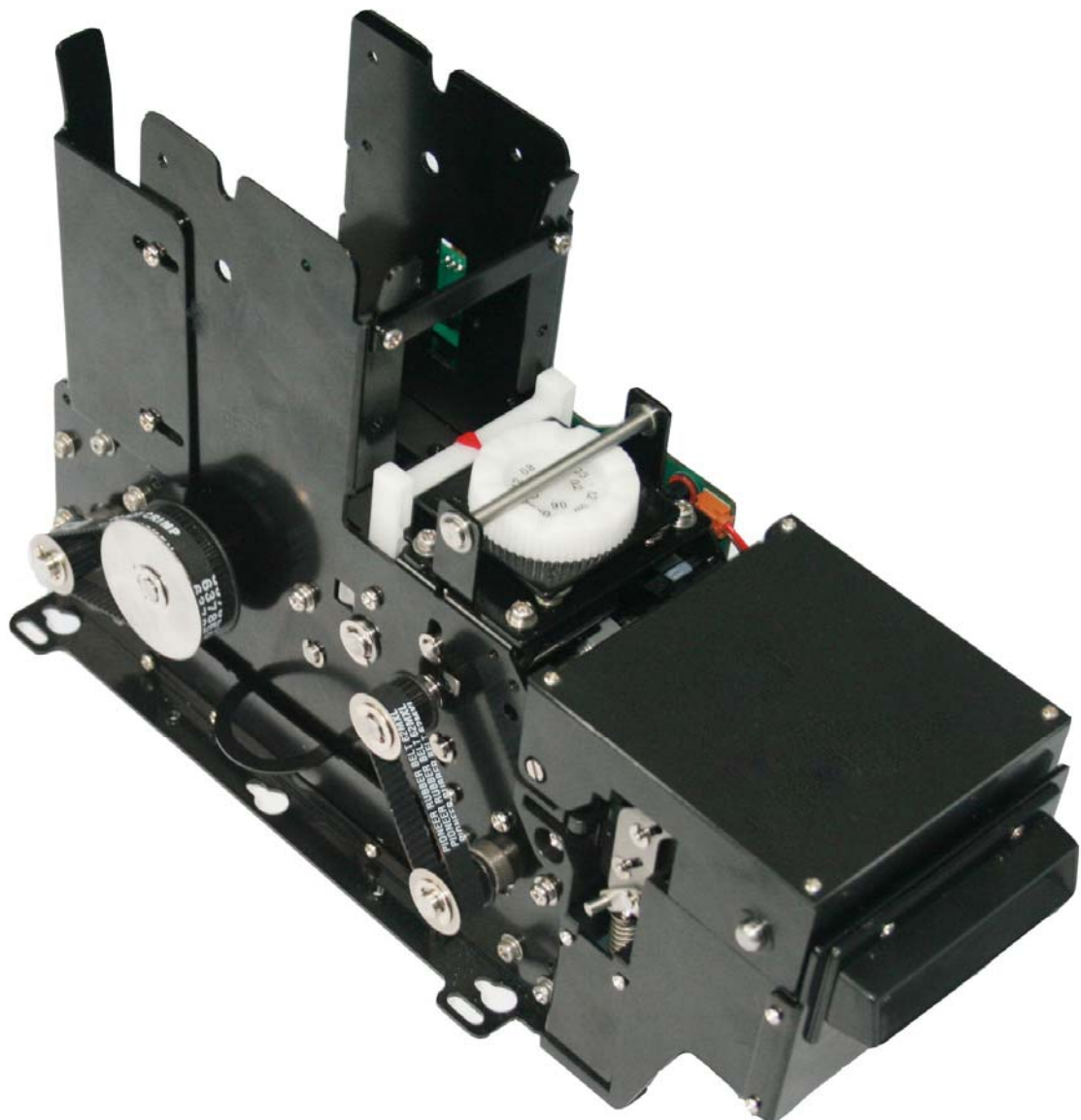
    2.4 执行命令函数 ..... 8

    2.5 IC 卡传输命令函数 ..... 9

3 示例代码 ..... 10

    3.1 部分常用命令列表 ..... 10

    3.2 示例代码 (接触式 CPU 卡操作过程) ..... 11



	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	3/16

# 1 函数参数组成说明

## 1.1 卡机分机地址

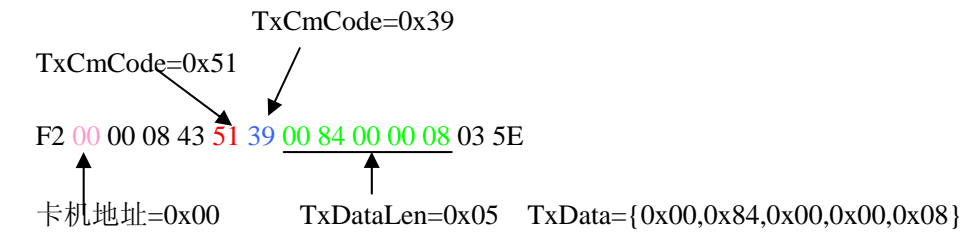
通过主板上的 4 位拨码开关来设定每一台分机地址：

拨码开关				设定卡机地址
4	3	2	1	
ON	ON	ON	ON	0x00
ON	ON	ON	OFF	0x01
ON	ON	OFF	ON	0x02
ON	ON	OFF	OFF	0x03
ON	OFF	ON	ON	0x04
ON	OFF	ON	OFF	0x05
ON	OFF	OFF	ON	0x06
ON	OFF	OFF	OFF	0x07
OFF	ON	ON	ON	0x08
OFF	ON	ON	OFF	0x09
OFF	ON	OFF	ON	0x0A
OFF	ON	OFF	OFF	0x0B
OFF	OFF	ON	ON	0x0C
OFF	OFF	ON	OFF	0x0D
OFF	OFF	OFF	ON	0x0E
OFF	OFF	OFF	OFF	0x0F

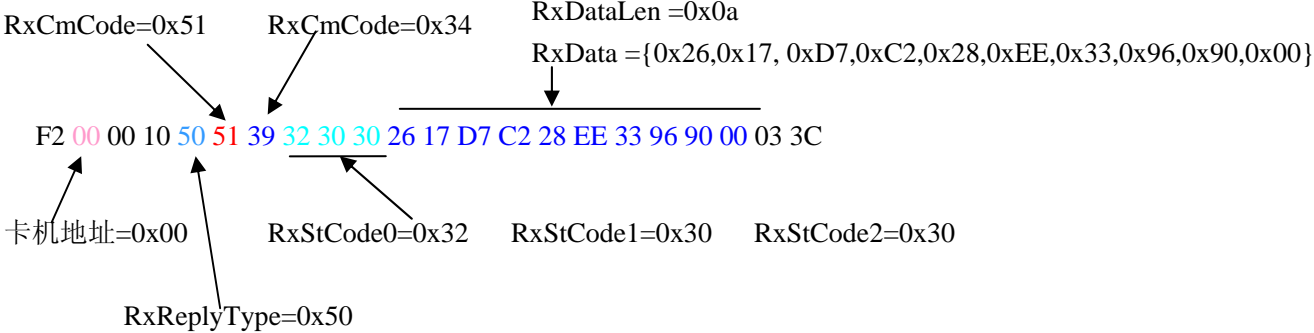
	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	4/16

2.1 根据通讯协议来说明函数参数组成如下：

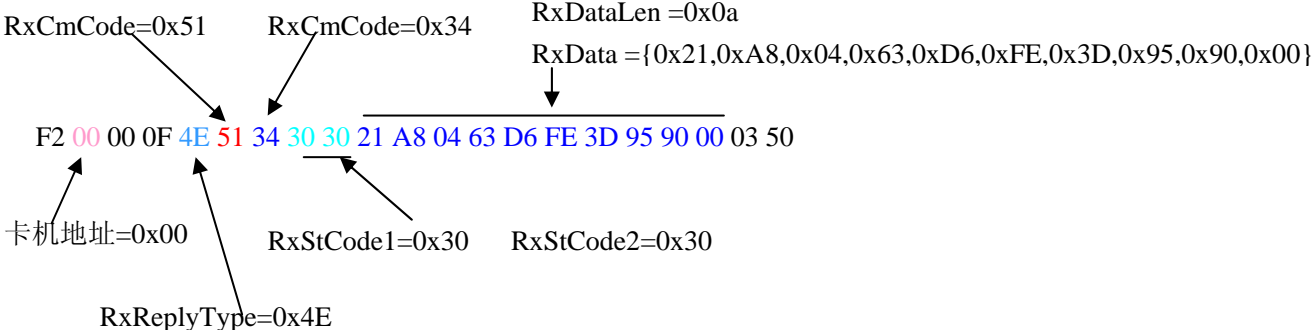
1.发送命令(Send command format )




2.接收成功命令(Receive command format)



3.接收失败命令(Receive command format)



注：  
接收成功命令： RxReplyType=0x50， RxStCode0， RxStCode1， RxStCode2 为卡机状态码  
接收失败命令： RxReplyType=0x4E， RxStCode1， RxStCode2 为失败原因码

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	5/16

## 2 API 参考说明

### 2.1 打开串口函数(默认波特率为 38400bps)

HANDLE APIENTRY CommOpen(char \*Port);

参数:

**Port:** 串口号字符串

例如: **CommOpen("COM1");**

返回:

0 打开串口失败

可能出现的原因:

① 无效的串口号

② 该串口被其它设备占用

<>0 串口文件句柄, 打开串口成功

备注:必须先调用此函数, 获得指定串口的串口文件句柄, 才可调用其他函数。  
可以同时打开多个串口, 获得多个串口文件句柄, 但不能多次打开同一个串口。  
使用完毕后, 必须调用**CommClose()**关闭串口。

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	6/16

## 2.2 用指定的波特率打开串口函数

HANDLE APIENTRY CommOpenWithBaut(char \*Port, unsigned int Baudrate);

参数:

Port: 串口号字符串

Baudrate 波特率

波特率=9600, 19200, 38400, 57600, 115200

例如: CommOpenWithBaut ("COM1", 38400);

返回:

串口文件句柄

0 打开串口失败

可能出现的原因:

- ① 无效的串口号
- ② 该串口被其它设备占用

<>0 串口文件句柄, 打开串口成功

备注: 必须先调用此函数, 获得指定串口的串口文件句柄, 才可调用其他函数。

可以同时打开多个串口, 获得多个串口文件句柄, 但不能多次打开同一个串口。

使用完毕后, 必须调用 CommClose() 关闭串口。

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	7/16

## 2.3 关闭串口函数

```
int APIENTRY CommClose(HANDLE ComHandle);;
```

参数:

ComHandle: 串口句柄。

返回:

- =0 成功
- <>0 失败

备注:与 CommOpen()或 CommOpenWithBaut 函数配套使用，并且在使用串口完毕后必须调用此函数关闭串口。

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	8/16

2.4 执行命令函数

```
int APIENTRY ExecuteCommand(HANDLE ComHandle,BYTE TxAddr,
                             BYTE TxCmCode,BYTE TxPmCode,
                             int TxDataLen,BYTE TxData[],
                             BYTE *RxReplyType,
                             BYTE *RxStCode0,BYTE *RxStCode1,BYTE *RxStCode2,
                             int *RxDataLen,BYTE RxData[])
```

功能：  
执行命令并返回执行结果

参数：

ComHandle: 串口句柄。

TxAddr:卡机地址，取值范围 0x00~0x0f

TxCmCode:命令代码

TxPmCode:参数代码

TxDataLen:命令附加的数据包长度

TxData: 命令附加的数据包

RxReplyType:返回的应答类型

0x50 : 执行成功

0x4E : 执行失败

0x10 : 下位机取消通讯 (NAK )

0x20 : 通讯错误

0x30 : 上位机取消命令 (DLE, EOT)

RxStCode0: 返回的状态代码0 //具体状态码见通讯协议中的状态码和错误码部分

RxStCode1: 返回的状态代码1 //具体状态码见通讯协议中的状态码和错误码部分

RxStCode2: 返回的状态代码2 //具体状态码见通讯协议中的状态码和错误码部分

RxDataLen: 返回的数据包长度

RxData: 返回的数据包

返回：

=0 成功

<>0 失败

注：若 RxReplyType 值为 0x50 时，则 RxStCode0，RxStCode1，RxStCode2 为卡机状态码

若 RxReplyType 值为 0x4E 时，则 RxStCode1，RxStCode2 为失败原因码，RxStCode0 值无意义



	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	9/16

2.5 IC 卡传输命令函数

```
int APIENTRY ICCardTransmit(HANDLE ComHandle,BYTE TxAddr,
                             BYTE TxCmCode,BYTE TxPmCode,
                             int TxDataLen,BYTE TxData[],
                             BYTE *RxReplyType,
                             BYTE *RxCmCode,BYTE *RxPmCode,
                             BYTE *RxStCode0,BYTE *RxStCode1,BYTE *RxStCode2,
                             int *RxDataLen,BYTE RxData[]);
```

功能：  
执行命令并返回执行结果  
注：相对于执行命令函数（ExecuteCommand）就是在其中增加了两个参数(RxCmCode和RxPmCode)

参数:

ComHandle: 串口句柄。

TxAddr:卡机地址，取值范围 0x00~0x0f

TxCmCode:命令代码

TxPmCode:参数代码

TxDataLen:命令附加的数据包长度

TxData: 命令附加的数据包

RxReplyType:返回的应答类型

- 0x50 : 执行成功
- 0x4E : 执行失败
- 0x10 : 下位机取消通讯 NAK (15H)
- 0x20 : 通讯错误
- 0x30 : 上位机取消命令 (DLE, EOT (10H, 04H))

RxCmCode: 返回的命令代码

RxPmCode: 返回的参数代码

RxStCode0: 返回的状态代码0 //具体状态码见通讯协议中的状态码和错误码部分

RxStCode1: 返回的状态代码1 //具体状态码见通讯协议中的状态码和错误码部分

RxStCode2: 返回的状态代码2 //具体状态码见通讯协议中的状态码和错误码部分

RxDataLen: 返回的数据包长度

RxData: 返回的数据包


返回:

=0 成功

<>0 失败

注：若 RxReplyType 值为 0x50 时，则 RxStCode0, RxStCode1, RxStCode2 为卡机状态码

若 RxReplyType 值为 0x4E 时，则 RxStCode1, RxStCode2 为失败原因码，RxStCode0 值无意义


	<b>SPECIFICATION</b>	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	10/16

## 3 示例代码

### 3.1 部分常用命令列表

cm: Command code pm: Parameters

Command	cm	Function	pm	Details of operation
INITIALIZE	30H	Initialize ICRW	30H	If card is inside ICRW, move card to gate
			31H	If card is inside ICRW, capture card to bin
			33H	If card is inside ICRW, does not move the card
			34H	Same as 30H. And Retract counter will work
			35H	Same as 31H. And Retract counter will work
			37H	Same as 33H. And Retract counter will work
STATUS REQUEST	31H	Inquire status	30H	Report presence of card and its position
			31H	Report presence of sensor status in detail
CARD MOVE	32H	Card move	30H	Move card to Gate and with hold card
			31H	Move card to IC contact position
			32H	Move card to RF
			33H	Capture card to BIN
			39H	Move card to Gate and without hold card
CARD IN	33H	Disable/Enable	30H	Accepts all card In
			31H	Disable
TEST CARD TYPE	50H	AutoTest card type	30H	AutoTest IC card type
			31H	AutoTest RF card type
CPU CARD	51H	CPU Card Operation	30H	CPU card cold reset
			31H	CPU card power off
			32H	CPU Cards Status Inquiry
			33H	CPU cards T = 0 protocol APDU data exchange
			34H	CPU cards T = 1 protocol APDU data exchange
			38H	CPU Card Hot Reset
			39H	Automatically distinguish between T = 0 / T = 1 protocols CPU card APDU data exchange
RFCARD	60H	RF Card Operation	30H	RF Card Activation
			31H	RF Card Deactivation
			32H	RF card operation status query
			33H	Mafare one Card Standard Operation
			34H	Type A standard T = CL card APDU data exchange
			35H	Type B standard T = CL card APDU data exchange

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	11/16

## 3.2 示例代码 (接触式 CPU 卡操作过程)

```

HANDLE hCom;    // Target device's HANDLE
int rc=0;       //Result
unsigned char Addr;
unsigned char CmCode;
unsigned char PmCode;
int  CmDataLen;
unsigned char CmData[1024];

unsigned char ReType;
unsigned char St2;
unsigned char St1;
unsigned char St0;
int  ReDataLen;
unsigned char ReData[1024];
// Open Comm. port
{
    hCom=CommOpenWithBaut("COM1",38400);
    if(hCom<=0)
    {
        // failed
        ...
    }
    else
    {
        // successfully.
        ...
    }
}
// Initialize
{
    memset(CmData,0x00,sizeof(CmData));
    Addr=0x00;
    CmCode=0x30; // Initialize command
    PmCode=0x30; // Parameter code
    CmDataLen=1; // Data size (bytes)
    CmData[0]=0x30;
    rc=ExecuteCommand(hCom,Addr,CmCode,PmCode,CmDataLen,CmData,
                      &ReType,&St0,&St1,&St2,&ReDataLen,ReData);
    if(rc==0)
        // Initialize command successfully finished.

```

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	12/16

```
if (ReType==0x50)
{
    // Received positive reply
    ...
}
else (ReType==0x4e)
{
    // Received negative reply
    ...
}
else
{
    //Communication Error
    // Initialize command failed.
    ...
}
}

// Setting Card In                                or                                // Move card to IC contact position
{
    memset(CmData,0x00,sizeof(CmData));
    CmCode= 0x33; // Card In
    PmCode= 0x30; // Enable
    CmDataLen= 0; // Data size
}

rc=ExecuteCommand(hCom,Addr,CmCode,PmCode,CmDataLen,CmData,
                  &ReType,&St0,&St1,&St2,&ReDataLen,ReData);
if(rc!=0) || ReType != 0x50)
{
    // Command sending failed or command execution failed
    ResErrMsg(St1,St2);    // ResErrMsg 是一个函数，用来显示错误原因
                          //具体错误状态码见通讯协议中的错误码部分,
    ...
}
}

// Request Status
{
    memset(CmData,0x00,sizeof(CmData));
    CmCode= 0x31; // Status request command
    PmCode= 0x30; // Parameter code
    CmDataLen=0; // Data size
    rc=ExecuteCommand(hCom,Addr,CmCode,PmCode,CmDataLen,CmData,
```

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	13/16

&ReType,&St0,&St1,&St2,&ReDataLen,ReData);

if(rc==0)

if (ReType==0x50)

{

//Execute Ok

if(St0 == '2') // There is a card at IC contact position

{

// status code="2xx"

// Detected a card inside of Card Reader/Writer

...

}

}

else (ReType==0x4e)

{

//Command execution failed

ResErrMsg(St1,St2);

...

}

else

{

//Communication Error

...

}

}

// CPU card cold reset

{

memset(CmData,0x00,sizeof(CmData));

CmCode= 0x51; // CPU card

PmCode=0x30; // Cold reset

CmDataLen=1; // Data size

If (true) //Choice Mode

{

CmData[0]=0x30; //EMV2000 V4.0 Vcc=5v

}

else

{

CmData[0]=0x33; // ISO/IEC7816-3


}

rc=ExecuteCommand(hCom,ICRWaddr,CmCode,PmCode,CmDataLen,CmData,&ReType,&St0,&St1,&St2,&ReDataLen,ReData);

if(rc==0)

if (ReType==0x50)


{

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	14/16

```

CString str,t;
CString TempBuf="";
for(int n=1; n<ReDataLen; n++)  //
{
    t.Format("%02x",ReData[n]);
    TempBuf += t;
    t="";
}
if (ReData[0]==48)
{
    MessageBox("CPU Card(T=0) Deactivate Successed and CPU card information meet the EMV mode
ATR ");
}
else
{
    MessageBox("CPU Card(T=1) Deactivate Successed and CPU card information meet the EMV mode
ATR");
}
}
else if((ReType==0x4e) && (ReDataLen>0))
{
    CString str,t;
    CString TempBuf="";
    for(int n=1; n<ReDataLen; n++)  //
    {
        t.Format("%02x",ReData[n]);
        TempBuf += t;
        t="";
    }
    if (ReData[0]==48)
    {
        MessageBox("CPU Card(T=0) Deactivate Successed,But CPU card information does not meet the
EMV mode ATR ");
    }
    else
    {
        MessageBox("CPU Card(T=1) Deactivate Successed,But CPU card information does not meet the
EMV mode ATR");
    }
}
else
{
    SANKYOErrMsg(St1,St2);
}

```

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	15/16

```

    }
else
{
    MessageBox("Communication Error");
}
}

// Exchanges data between the Host Computer and IC card
{
    memset(CmData,0x00,sizeof(CmData));
    CmCode= 0x51; // IC card control
    PmCode=0x39; // Deactivate
    CmDataLen=5; // Data size
    CmData[0]=0x00;
    CmData[1]=0x84;
    CmData[2]=0x00;
    CmData[3]=0x00;
    CmData[4]=0x08;
    rc=ExecuteCommand(hCom,Addr,CmCode,PmCode,CmDataLen,CmData,
                      &ReType,&St0,&St1,&St2,&ReDataLen,ReData);
    if(rc!=0)
    {
        // ICardTransmit failed
        ...
    }
    else
    {
        if (ReType==0x50)
        {
            CString t;
            CString TempBuf="";
            for(int n=0; n<ReDataLen; n++) // R-APDU= start from n=0
            {
                t.Format("%02x",ReData[n]);
                TempBuf += t;
                t="";
            }
            // TempBuf is the buffer of R-APDU data
        }
        else //(ReType==0x4e)
        {
            // Command execution failed
            SANKYOErrMsg(St1,St2);
        }
    }
}

```

	SPECIFICATION	Model No.	CRT-571 读卡器
		Date	2010/01/28
	动态库说明	Ver.	v20100128
		Page	16/16

```
    }
}
// CPU card power off
{
    memset(CmData,0x00,sizeof(CmData));
    CmCode=0x51; // CPU card power off
    PmCode=0x31; // Power off
    CmDataLen=0; // Data size
    rc=ExecuteCommand(hCom,Addr,CmCode,PmCode,CmDataLen,CmData,
                      &ReType,&St0,&St1,&St2,&ReDataLen,ReData);
    if(rc!=0) || ReType != 0x50)
    {
        // Command sending failed or command execution failed
        ...
    }
}
_EXIT1:
// Closes communications between the Host Computer and the Card Reader/Writer
CommClose(hCom);
_EXIT:
```