

IT-100MU

非接觸介面軟體開發文件

WinNfc API

Version 1.0



訊想科技股份有限公司

中華民國九十九年十一月

文件制/修訂紀錄

版次	日期	制/修訂說明	作者
V 1.00	99/11/03	First Version	Roman

目 錄

1. WinNfc API 函式清單.....	5
1.1 建立/釋放函式庫	5
NfcEstablishContext.....	5
NfcReleaseContext	6
1.2 讀卡機相關函式	7
NfcConnect.....	7
NfcDisConnect	8
NfcLEDControl.....	9
NfcBuzzerControl.....	10
1.3 卡片相關函式	11
NfcSelectCard	11
NfcDeSelectCard	12
1.4 MIFARE 卡相關函式	13
Mifare_GetCardID	13
Mifare_KeyAuthority	14
Mifare_ReadBlock.....	15
Mifare_WriteBlock	16
Mifare_ReadValue.....	17
Mifare_WriteValue	18
Mifare_IncreaseValue.....	19
Mifare_DecreaseValue.....	20
Mifare_TransferValue	21
Mifare_RestoreValue	22
1.5 ISO14443-4A/B	23
ISO14443_SendAPDU	23
2. 錯誤代碼(Error Code)	25
0.....	25
0x80100001L	25
0x80100003L	25
0x80100004L	25
0x80100006L	25
0x80100009L	25
0x8010000AL	25
0x8010000CL	25
0x8010000DL	25
0x80100011L	25

0x80100014L	25
0x80100017L	25
0x8010001AL	25
0x8010001CL	25
0x8010002EL	25
3.API 詢問窗口	26

1.WinNfc API 函式清單

1.1 建立/釋放函式庫

NfcEstablishContext

建立並初始化 WinNfc 函式庫。

```
WINNFC_API LONG WINAPI  
NfcEstablishContext(  
    IN  DWORD dwScope,  
    IN  LPCVOID pvReserved1,  
    IN  LPCVOID pvReserved2,  
    OUT LPNFCCONTEXT phContext);
```

參數：

dwScope:個人或系統模式，此參數暫時無用途。

pvReserved1:保留參數。

pvReserved2: 保留參數。

phContext:回傳 WinNfc Handle，後續操作其他函式使用。

回覆值：

成功： NFC_S_SUCCESS

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
NFCCONTEXT phCtx = NULL;  
LONG rc = 0;  
  
//initialize NFC functions  
rc = NfcEstablishContext(0, 0, 0, &phCtx);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
  
printf("initialize NFC functions successfully\n", rc);
```

NfcReleaseContext

結束並釋放 WinNfc 函式庫,

```
WINNFC_API LONG WINAPI  
NfcReleaseContext(  
    IN    NFCCONTEXT hContext);
```

參數:

phContext: WinNfc Handle。

回覆值:

成功: NFC_S_SUCCESS

失敗: 錯誤代碼(詳見 Error Code 清單)

範例:

```
NfcReleaseContext(ctx);  
printf("release NFC functions successfully\n", rc);
```

1.2 讀卡機相關函式

NfcConnect

連接讀卡機。

```
WINNFC_API LONG WINAPI
NfcConnect(
    IN      NFCCONTEXT hContext,
    IN      LPCSTR szReader,
    IN      DWORD dwShareMode);
```

參數：

hContext: WinNfc Handle。

szReader: 讀卡機名稱，輸入 NULL 自動尋找讀卡機；輸入 "InfoThink IT-100MU 0" 代表連接第一台讀卡機；輸入 "InfoThink IT-100MU 1" 代表連接第二台讀卡機，以此內推。

dwShareMode: 連接模式，此參數暫時無用途。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//Auto connect to IT-100MU reader
rc = NfcConnect(phCtx, NULL, 0);

if (rc) {
    ShowError(phCtx, rc);
    return 0;
}

printf("auto connect NFC reader successfully\n");
```

NfcDisconnect

中斷讀卡機連線。

```
WINNFC_API LONG WINAPI  
NfcDisconnect(  
    IN    NFCCONTEXT hContext,  
    IN    DWORD dwDisposition);
```

參數：

hContext: WinNfc Handle。

dwDisposition: 中斷模式，此參數暫時無用途。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//disconnect IT-100MU reader  
rc = NfcDisconnect(phCtx, 0);  
  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
  
printf("disconnect successfully\n");
```


NfcLEDControl

開關 IT-100MU 讀卡機左上方藍色和紅色 LED 燈。

```
WINNFC_API LONG WINAPI  
NfcLEDControl(  
    IN NFCCONTEXT hContext,  
    IN BYTE bState);
```

參數：

hContext: WinNfc Handle。

bState:開關 LED 燈參數，bit-1 控制紅燈 ON 或 OFF, bit-2 控制藍燈 ON 或 OFF。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//LED test  
printf("LED BLUE ON\n");  
rc = NfcLEDControl(phCtx, LED_BLUE);  
Sleep(500);  
printf("LED RED ON\n");  
rc = NfcLEDControl(phCtx, LED_RED);  
Sleep(500);  
printf("LED ALL ON\n");  
rc = NfcLEDControl(phCtx, LED_ALL);  
Sleep(500);  
printf("LED ALL OFF\n");  
  
rc = NfcLEDControl(phCtx, LED_OFF);
```

NfcBuzzerControl

開關 IT-100MU 讀卡機蜂鳴器。

```
WINNFC_API LONG WINAPI  
NfcBuzzerControl(  
    IN NFCCONTEXT hContext,  
    IN BYTE bState);
```

參數：

hContext: WinNfc Handle。

bState: 開關 Buzzer 參數，0x00 打開 Buzzer 或 0x08 關閉 Buzzer。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//BUZZER test  
NfcBuzzerControl(phCtx, BUZZER_ON);  
Sleep(50);  
  
NfcBuzzerControl(phCtx, BUZZER_OFF);
```

1.3 卡片相關函式

NfcSelectCard

由於天線的訊號控制在一定的範圍，當卡片距離讀卡機 5,6 公分內，呼叫此函式搜尋卡片，找到卡後即可透過 **NfcTransmit** 函式與卡片交換資料。若天線範圍內有多張卡片，IT-100MU 同時間只能存取 1 張卡。

```
WINNFC_API LONG WINAPI
NfcSelectCard(
    IN    NFCCONTEXT hContext,
    IN    DWORD dwCardTypes,
    IN    PBYTE pbInitData,
    IN    DWORD dwInitDataLen);
```

參數：

hContext: WinNfc Handle。

dwCardTypes: 卡別。

pbInitData: 輸入資料(保留參數)。

dwInitDataLen: 輸入資料長度(保留參數)。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 **Error Code** 清單)

範例：

```
//Find card and select the first card
rc = NfcSelectCard(phCtx, NFC_CARDTYPE_MIFARE, NULL, 0);

if (rc) {
    ShowError(phCtx, rc);
    return 0;
}

printf("select card successfully\n");
```

NfcDeSelectCard

將 myKEY II 資料清除化，清除後 myKEY II 會回到空白狀態，管理者密碼(SO PIN) 驗證通過後才成清除 myKEY II 。

```
WINNFC_API LONG WINAPI  
NfcDeSelectCard(  
    IN        NFCCONTEXT hContext);
```

參數：

hContext: WinNfc Handle 。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//deselect the card if you don't use it anymore  
rc = NfcDeSelectCard(phCtx);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
  
printf("DeSelect card successfully\n");
```

1.4 MIFARE 卡相關函式

MIFARE 卡格式的相關文件可以參考：

(1) M001053_MF1ICS50_rev5_3.pdf: NXP MIFARE 卡規格文件

Mifare_GetCardID

取得 MIFARE 卡號。

```
WINNFC_API LONG WINAPI  
Mifare_GetCardID(  
    IN  NFCCONTEXT hContext,  
    IN  LPSTR mszID);
```

參數：

hContext: WinNfc Handle。

mszID: 卡號字串(8 個 bytes)。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//get Mifare Card ID  
char szCardID[9] = {0};  
rc = Mifare_GetCardID(phCtx, szCardID);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
  
printf("get MIFARE Card ID: %s \n", szCardID);
```

Mifare_KeyAuthority

MIFARE 卡每個 Sector 有 4 個 Block，由兩把金鑰分別為 KeyA 和 KeyB 控管權限，此函式可用來認證 KeyA 或 KeyB，驗證成功後即會擁有特定權限(請參考 MIFARE 規格)。

```
WINNFC_API LONG WINAPI  
Mifare_KeyAuthority(  
    IN  NFCCONTEXT hContext,  
    IN  BYTE bBlock,  
    IN  BYTE bKeyType,  
    IN  LPCBYTE pbKey);
```

參數：

hContext: WinNfc Handle。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//authenticate key A on Block N given  
rc = Mifare_KeyAuthority(phCtx, Block, KEY_TYPE_A, KeyA);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
  
printf("authenticate key A successfully\n");
```

Mifare_ReadBlock

讀取第 N 個 Block 裡的資料。

```
WINNFC_API LONG WINAPI  
Mifare_ReadBlock(  
    IN  NFCCONTEXT hContext,  
    IN  BYTE bBlock,  
    IN OUT LPBYTE pbData,  
    IN OUT LPDWORD pcbLength);
```

參數：

hContext: WinNfc Handle。

bBlock: 卡片 Block 編號位置。

pbData: 讀取資料的 Buff。

pcbLength: 讀取資料長度。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//read data from Block N  
BYTE data[16] = {0};  
DWORD dwLen = 16;  
rc = Mifare_ReadBlock(phCtx, Block, data, &dwLen);  
  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
  
printf("read data from block %d successfully\n", Block);  
ShowData(data, 16);
```

Mifare_WriteBlock

寫入資料到第 N 個 Block。

```
WINNFC_API LONG WINAPI  
Mifare_WriteBlock(  
    IN  NFCCONTEXT hContext,  
    IN  BYTE bBlock,  
    IN  LPCBYTE pbData,  
    IN  DWORD cbLength);
```

參數：

hContext: WinNfc Handle。

bBlock: 卡片 Block 編號位置。

pbData: 寫入資料的 Buff。

cbLength: 寫入資料長度。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//write data to Block N  
PBYTE pData = (PBYTE)"\x01\x02\x03\x04\x05\x06\x07\x08\x01\x02\x03\x04\x05\x06\x07\x08";  
rc = Mifare_WriteBlock(phCtx, Block, pData, 16);  
  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
  
printf("write data to block %d successfully\n", Block);
```


Mifare_ReadValue

MIFARE 定義一種 16 bytes 的 Value Block，可用來儲存數值(通常為數位錢包，這種 Value Block 可以透過特定指令來加值(Increase Value)或扣款(Decrease Value)，此函式可以用來讀取 Value Block 的數值。

```
WINNFC_API LONG WINAPI  
Mifare_ReadValue(  
    IN  NFCCONTEXT hContext,  
    IN  BYTE bBlock,  
    IN  LPDWORD pcbValue);
```

參數：

hContext: WinNfc Handle。

bBlock: 卡片 Block 編號位置。

pcbValue: Block 裡所存放的數值。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
dwBalance = 0;  
rc = Mifare_ReadValue(phCtx, Block, &dwBalance);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
  
printf("now block %d balance is %d\n", Block, dwBalance);
```

Mifare_WriteValue

寫入數值到第 N 個 Value Block。

```
WINNFC_API LONG WINAPI  
Mifare_WriteValue(  
    IN  NFCCONTEXT hContext,  
    IN  BYTE bBlock,  
    IN  LPDWORD pcbValue);
```

參數：

hContext: WinNfc Handle。

bBlock: 卡片 Block 編號位置。

pcbValue: 寫入的數值。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//write value on Block N given  
rc = Mifare_WriteValue(phCtx, Block, Value);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
  
printf("write value %d successfully\n", Value);
```

Mifare_IncreaseValue

針對第 N 個 Value Block 增加數值(加值)，此函式會加值的結果暫時放在記憶體空間，等呼叫 Mifare_TransferValue 後才會把結果存放到指定的 Block。

```
WINNFC_API LONG WINAPI  
Mifare_IncreaseValue(  
    IN NFCCONTEXT hContext,  
    IN BYTE bBlock,  
    IN DWORD pcbValue);
```

參數：

hContext: WinNfc Handle。

bBlock: 卡片 Block 編號位置。

pcbValue: 欲增加的數值。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
/* 加值100 元/點*/  
rc = Mifare_IncreaseValue(phCtx, Block, 200);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
rc = Mifare_TransferValue(phCtx, Block);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
printf("add 200 to block %d\n", Block);
```

Mifare_DecreaseValue

針對第 N 個 Value Block 減少數值(扣款)，此函式會扣款的結果暫時放在記憶體空間，等呼叫 Mifare_TransferValue 後才會把結果存放到指定的 Block。

```
WINNFC_API LONG WINAPI
Mifare_DecreaseValue(
    IN  NFCCONTEXT hContext,
    IN  BYTE bBlock,
    IN  DWORD pcbValue);
```

參數：

hContext: WinNfc Handle。

bBlock: 卡片 Block 編號位置。

pcbValue: 欲減少的數值。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
/* 扣款100 元/點*/
rc = Mifare_DecreaseValue(phCtx, Block, 100);
if (rc) {
    ShowError(phCtx, rc);
    return 0;
}
rc = Mifare_TransferValue(phCtx, Block);
if (rc) {
    ShowError(phCtx, rc);
    return 0;
}

printf("remove 100 from block %d\n", Block);
```

Mifare_TransferValue

將執行 Mifare_IncreaseValue 和 Mifare_DecreaseValue 的結果(暫存的記憶體)存放到指定的 Block。

```
WINNFC_API LONG WINAPI  
Mifare_TransferValue(  
    IN  NFCCONTEXT hContext,  
    IN BYTE bBlock);
```

參數：

hContext: WinNfc Handle。

bBlock: 卡片 Block 編號位置。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
/* 扣款100 元/點*/  
rc = Mifare_DecreaseValue(phCtx, Block, 100);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
rc = Mifare_TransferValue(phCtx, Block);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
  
printf("remove 100 from block %d\n", Block);
```

Mifare_RestoreValue

在 MIFARE 電子錢包的設計上，為了安全起見會採用兩個 **Block** 來存放同一個數值，另一個數值作為備份用途，同時若有交易不明確的情況下，可以取消交易並將資料從備份的 **Block** 復原回去。

```
WINNFC_API LONG WINAPI  
Mifare_RestoreValue(  
    IN  NFCCONTEXT hContext,  
    IN BYTE bBlock);
```

參數：

hContext: WinNfc Handle。

bBlock: 卡片 Block 編號位置。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
rc = Mifare_RestoreValue(phCtx, Block + 1);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}  
rc = Mifare_TransferValue(phCtx, Block);  
if (rc) {  
    ShowError(phCtx, rc);  
    return 0;  
}
```

1.5 ISO14443-4A/B

ISO14443_SendAPDU

符合 ISO1443-4A/B 規格的卡片，可透過 T=CL 的通訊協定將 APDU 指令傳送至卡片。

ISO1443-4 詳細規格可參考 ISOIEC 14443-4.pdf

```
WINNFC_API LONG WINAPI  
ISO14443_SendAPDU(  
    IN  NFCCONTEXT hContext,  
    IN  LPCBYTE pbSendBuffer,  
    IN  DWORD cbSendLength,  
    OUT LPBYTE pbRecvBuffer,  
    IN  OUT LPDWORD pcbRecvLength);
```

參數：

hContext: WinNfc Handle。

pbSendBuffer: 送出資料。

cbSendLength: 送出資料長度。

pbRecvBuffer: 接收資料。

pcbRecvLength: 接收資料長度。

回覆值：

成功： 0

失敗： 錯誤代碼(詳見 Error Code 清單)

範例：

```
//Select AID APDU  
BYTE SelectAID[ ] = {0x00,0xA4,0x04,0x00,0x07,0xA0,0x00,0x00,0x00,0x03,0x00,0x00,0x00};  
  
BYTE Rec[262] = {0};  
DWORD dwLen = 262;
```

```
printf("Select AID \n");
ShowData(SelectAID, 13);

//transmit C=TL APDU command
rc = ISO14443_SendAPDU(phCtx, (LPCBYTE)SelectAID, 13, Rec, &dwLen);
if (rc) {
    ShowError(phCtx, rc);
    return 0;
}

printf("get response data successfully\n");
ShowData(Rec, dwLen);
```


2.錯誤代碼(Error Code)

NFC_S_SUCCESS	0	函式呼叫正確
NFC_F_INTERNAL_ERROR	0x80100001L	內部錯誤
NFC_E_INVALID_HANDLE	0x80100003L	傳入的 Handle 值錯誤
NFC_E_INVALID_PARAMETER	0x80100004L	傳入參數錯誤
NFC_E_NO_MEMORY	0x80100006L	記憶體不足
NFC_E_UNKNOWN_READER	0x80100009L	無法辨識的讀卡機
NFC_E_TIMEOUT	0x8010000AL	函式逾時
NFC_E_NO_SMARTCARD	0x8010000CL	找不到卡片
NFC_E_UNKNOWN_CARD	0x8010000DL	無法辨識的卡片
NFC_E_INVALID_VALUE	0x80100011L	傳入參數值錯誤
NFC_F_UNKNOWN_ERROR	0x80100014L	未知型錯誤
NFC_E_READER_UNAVAILABLE	0x80100017L	目前無法存取讀卡機
NFC_E_READER_UNSUPPORTED	0x8010001AL	不支援的讀卡機
NFC_E_CARD_UNSUPPORTED	0x8010001CL	不支援的卡片
NFC_E_NO_READERS_AVAILABLE	0x8010002EL	找不到讀卡機

3.API 詢問窗口

若您有採購合法的 SDK 軟體開發工具，有任何使用本 API 的問題，均可來信詢問，詢問窗口為 roman@ittec.com.tw