

Windows 版本

HDACS  
控制軟體 API 參考檔

RAC340/520/820/920/930

Version 2.3

# 目錄

第一章 系統需求.....	1
第二章 支援機型列表.....	1
第三章 錯誤傳回值說明 .....	2
第四章 RAC系列相關函數說明 .....	3
4.1 與通訊相關 API(Level 0) .....	3
4.1.1 hacOpenChannel(開啓通訊) .....	3
4.1.2 hacCloseChannel(關閉通訊).....	4
4.1.3 hacReadData(讀取通訊buffer資料).....	4
4.1.4 hacWriteData(將資料傳送到通訊) .....	5
4.1.5 hacClearBuffer(將Buffer中的資料清除).....	6
4.2. 與協議相關 API (Level1) .....	7
4.2.1 hacHWReadCommand(送出Read 指令到裝置) .....	7
4.2.2 hacHWWRITECommand(送出Write 指令到裝置) .....	8
4.2.3 hacHWReadCommandEX(送出Read 指令到裝置) .....	9
4.4.4 hac34Polling (Polling設備).....	10
4.3 與硬體相關 API(Level 2) .....	12
4.3.1 hac34GetDateTime(讀取裝置上的日期時間) .....	12
4.3.2 hac34SetDateTime(將日期時間寫入裝置) .....	13
4.3.3 hac34GetEEData(讀取裝置EEPROM 值) .....	14
4.3.4 hac34SetEEData (寫入EEPROM 值).....	15
4.3.5 hac34GetVersion(讀取裝置版本) .....	16
4.3.6 hac34AddCard(單筆新增卡號/非壓縮).....	17
4.3.7 hac34DelCard(單筆刪除卡號/非壓縮) .....	18
4.3.8 hac34DelAllCard(刪除所有卡號) .....	19
4.3.9 hac34SetDoorRelayMode (設定開門Relay 動作模式).....	20
4.3.10 hac34SetStatus (設定Reader 狀態) .....	21
4.3.11 hac34SetReadCardParameter(設定mifare讀卡參數).....	22
4.3.12 hac34GetReadCardParameter(讀取mifare讀卡參數) .....	23
4.3.13 hac34GetIO (讀取裝置上目前 IO 狀態) .....	24
4.3.14 hac34Dump( DUMP 裝置上相關資訊) .....	25
第五章 附錄 .....	26
5.1 RAC-340 相關記憶體規劃 .....	26
5.2 RAC-520 相關記憶體規劃 .....	29
5.3 RAC-820/920/930 相關記憶體規劃 .....	31

## 第一章 系統需求

1. 本 DLL 支持多線程。
2. 採用 Virtual Studio .NET Professional 2003 C++ 開發環境。
3. 本 DLL 支援 DELPHI, VB, VC2005, C#開發工具開發。
4. Sample Code 有 VB6, DELPHI7, VC2005, C#2005。

## 第二章 支援機型列表

RAC-340 series : RAC-340PE, RAC-340PM  
RAC-520 series : RAC-520PE, RAC-520PM  
RAC-820 series : RAC-820PE, RAC-820PM  
RAC-920 series : RAC-920PE, RAC-920PM  
RAC-930 series : RAC-930PE, RAC-930PM

### 第三章 錯誤傳回值說明

返回值	說明
0	成功
1125	對多線程同步物件訪問時出現等待超時。
1126	釋放多線程同步物件時出現錯誤。
1001	傳遞的參數有錯誤，或者表示設備回應封包的狀態字元錯誤，具體錯誤資訊參考 ReturnCode。
1002	Socket 讀寫錯誤或者串口讀寫錯誤，執行非同步讀寫的時候發生錯誤。
1003	執行設備資料讀寫的時候，接收或寫入的資料長度有錯誤。
1103	設備回應封包長度小於要求的長度。
1004	傳入的控制控制碼無效，無效的 Handle。
1005	設備回應封包結束符錯誤。
1006	設備回應封包 CRC-16 檢查錯誤。
1106	電腦發送的封包 CRC-16 檢查錯誤。
1007	電腦發送給設備的命令錯誤，或者表示設備不支援此功能。
1008	下層設備進行 R/W 讀寫時出現錯誤。
1009	電腦發送給設備的封包過長。
1010	設備中沒有任何資料，記錄筆數為，或者表示沒有讀取到要求的資料。
4445	讀取設備中的資料或者記錄時發生異常錯誤。
1025	執行非同步讀寫的時候出現超時錯誤，超過等待時間(iTimeout)。
1026	執行非同步讀寫的時候出現等待錯誤。
2225	執行非同步讀寫的時候，沒有接收到任何資料。

## 第四章 RAC 系列相關函數說明

### 4.1 與通訊相關 API(Level 0)

#### 4.1.1 hacOpenChannel(開啓通訊)

API 名稱	hacOpenChannel
語法	<pre>int __stdcall hacOpenChannel (     char *sComm,     unsigned int iPort,     HANDLE *hComm )</pre>
目的	開啓通訊連接，可以 TCP/IP 或 COM Port
參數說明	<p>開啓 COM 通訊埠時：</p> <p>sComm: COM1-COM128</p> <p>iPort: 傳送速率(1200/2400/4800/9600/19200/38400)</p> <p>hComm: 開啓成功後傳回的 handle 值</p> <p>開啓 TCP/IP 時：</p> <p>sComm: IP 地址, 如 172.16.1.1</p> <p>iPort: 埠號, 如 4660</p> <p>hComm: 開啓成功後, 傳回 socket 值</p> <p>注意：使用 TCP/IP 轉換器時，除了 Polling，否則建議每次通訊時都進行連接，通訊完畢以後關閉連接。如 BF430，進行 SOCKET 連接時只能保持很短一段時間（具體要看設備的參數設定），所以建議每次都重新連接。</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.1.2 hacCloseChannel(關閉通訊)

API 名稱	hacCloseChannel
語法	<pre>int __stdcall hacCloseChannel (     HANDLE hComm )</pre>
目的	關閉通訊連接
參數說明	hComm: 所要關閉的 handle 值, 如果 handle > 1000 表示為 TCP/IP
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.1.3 hacReadData(讀取通訊 buffer 資料)

API 名稱	hacReadData
語法	<pre>int __stdcall hacReadData (     char *cBuffer,     int *iDataLen,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	讀取通訊 buffer 上的資料
參數說明	cBuffer: 存放傳回資料內容 iDataLen: 傳回讀取數據長度 hComm: handle 值, 可以是 COM 或是 TCP/IP iTimeout: 讀取等待時間, 單位為 ms
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.1.4 hacWriteData(將資料傳送到通訊)

API 名稱	hacWriteData
語法	<pre>int __stdcall hacReadData (     char *cBuffer,     int iDataLen,     int iWrittenLen,     HANDLE hComm )</pre>
目的	將資料經由 cBuffer 送到 COM 或是 TCP/IP 上
參數說明	cBuffer: 存放傳送資料內容 iDataLen: 數據長度 iWrittenLen : 成功傳送的資料長度 hComm: handle 值, 可以是 COM 或是 TCP/IP
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.1.5 hacClearBuffer(將 Buffer 中的資料清除)

API 名稱	hacClearBuffer
語法	<pre>int __stdcall hacClearBuffer (     HANDLE hComm )</pre>
目的	清除通訊 Buffer 中的資料
參數說明	hComm: handle 值, 可以是 COM 或是 TCP/IP
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
設備	

## 4.2. 與協議相關 API (Level1)

### 4.2.1 hacHWReadCommand(送出 Read 指令到裝置)

API 名稱	hacHWReadCommand
語法	<pre>int __stdcall hacHWReadCommand (     int iNodeID,     int iIndex,     char *cSendData,     int iSendDataLen,     char *cReceiveData,     int *iReceiveLen,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	將資料 cSendData 經由送到 COM 或是 TCP/IP 上的裝置，等待 Respond 封包，並檢查傳送是否成功
參數說明	iNodeID: 裝置代號 int iIndex : RW 表的索引 cSendData: 存放傳送資料內容，目前保留 iSendDataLen: 傳送資料長度，目前保留 cReceiveData: 存放裝置傳回資料內容 iReceiveDataLen: 傳回資料長度 hComm: handle 值，可以是 COM 或是 TCP/IP iTimeout: 等待逾時，單位為 ms，如果忽略此參數，則強制等待100毫秒
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.2.2 hacHWWWriteCommand(送出 Write 指令到裝置)

API 名稱	<code>hacHWWWriteCommand</code>
語法	<pre>int __stdcall hacHWWWriteCommand (     int iNodeID,     int iIndex,     char *cSendData,     int iSendDataLen,     char *cReceiveData,     int *iReceiveLen,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	將資料 <code>cSendData</code> 經由送到 COM 或是 TCP/IP 上的裝置，等待 Respond 封包，並檢查傳送是否成功
參數說明	<p><code>iNodeID</code>: 裝置代號</p> <p><code>int iIndex</code> : RW 表的索引</p> <p><code>cSendData</code>: 存放傳送資料內容</p> <p><code>iSendDataLen</code>: 傳送資料長度</p> <p><code>cReceiveData</code>: 存放裝置傳回資料內容</p> <p><code>iReceiveDataLen</code>: 傳回數據長度</p> <p><code>hComm</code>: handle 值，可以是 COM 或是 TCP/IP</p> <p><code>iTimeout</code>: 等待逾時，單位為 ms，如果忽略此參數，則強制等待100毫秒</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.2.3 hacHWReadCommandEX(送出 Read 指令到裝置)

API 名稱	hacHWReadCommandEX
語法	<pre>int __stdcall hacHWReadCommandEX (     int iNodeID,     int iIndex,     char *cSendData,     int iSendDataLen,     char *cReceiveData,     int *iReceiveLen,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	將資料 cSendData 經由送到 COM 或是 TCP/IP 上的裝置，等待 Respond 封包，並檢查傳送是否成功
參數說明	<p>iNodeID: 裝置代號</p> <p>int iIndex : RW 表的索引</p> <p>cSendData: 存放傳送資料內容，目前保留</p> <p>iSendDataLen: 傳送資料長度，目前保留</p> <p>cReceiveData: 存放裝置傳回資料內容，和 hacHWReadCommand 函數不同的是，此函數傳回來的是去除了起始位元組的真正 Data 部分的內容。</p> <p>iReceiveDataLen: 傳回資料長度，和 hacHWReadCommand 函數不同的是，此函數傳回來的是去除了起始位元組的真正 Data 部分的長度。</p> <p>hComm: handle 值，可以是 COM 或是 TCP/IP</p> <p>iTimeout: 等待逾時，單位為 ms，如果忽略此參數，則強制等待100毫秒</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.4.4 hac34Polling (Polling 設備)

API 名稱	hac34Polling
語法	<pre>int __stdcall hac34Polling (     int iNodeID,     int iPrevRecord,     stPollList *stRecord     int *iRecord,     HANDLE hComm,     unsigned int iTimeout,     int iCardType )</pre>
目的	發出 Poll 指令給設備，返回獲得的紀錄
參數說明	<p>iNodeID: 裝置代號</p> <p>iPrevRecord : 上一次收到的資料筆數</p> <p>stRecord: 存放傳回資料，是個結構</p> <p>iRecord: 傳回的事件筆數</p> <p>hComm: handle 值, 可以是 COM 或是 TCP/IP</p> <p>iTimeout: 等待逾時, 單位為 ms</p> <p>iCardType:是否壓縮，0 表示卡號不壓縮，1 表示卡號壓縮</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	<p>此 Polling 傳回來的資料已經解碼</p> <pre>typedef struct stPollList {     char cEventCode[5]; // Event Code     char cDateTime[20]; // Date &amp; Time     char cCard[20]; // Card Number     char cDeviceID[10]; // Device ID     char cReaderID[10]; // Reader ID } stPollList;</pre> <p>如果是事件，代碼說明：</p> <ul style="list-style-type: none"> <li>03 01 Case Sensor 警報</li> <li>03 05 外力開啓解除</li> <li>03 06 冷開機</li> <li>03 07 重新開機</li> </ul>

	03 11 門被外力開啓
	03 12 門未關妥警告
	03 13 門未關妥解除
	03 18 開門
	03 19 關門
	03 1A 開門鈕觸動
	01 20 脊迫開門警報
	01 22 脊迫開門警報解除

如果是卡號，代碼說明：

0000	沒資料
0001	上班按鍵—合法卡
0011	上班刷卡—合法卡
0101	下班按鍵—合法卡
0111	下班刷卡—合法卡
0007	上班按鍵—非法卡
0017	上班刷卡—非法卡
0107	下班按鍵—非法卡
0117	下班刷卡—非法卡

## 4.3 與硬體相關 API(Level 2)

### 4.3.1 hac34GetDateTime(讀取裝置上的日期時間)

API 名稱	Hac34GetDateTime
語法	<pre>int __stdcall hac34GetDateTime (     int iNodeID,     char *cDate,     char *cTime,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	讀取裝置上的日期時間
參數說明	iNodeID: 裝置代號 cDate: 傳回的日期, 格式為 yyyyymmddw cTime: 傳回的時間, 格式為 hhmmss hComm: handle 值, 可以是 COM 或是 TCP/IP iTimeout: 等待逾時, 單位為 ms
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.3.2 hac34SetDateTime(將日期時間寫入裝置)

API 名稱	Hac34SetDateTime
語法	<pre>int __stdcall hac34SetDateTime (     int iNodeID,     char *cDate,     char *cTime,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	設定裝置上的日期時間
參數說明	iNodeID: 裝置代號 cDate: 傳入的日期, 格式為 yyyyymmddw cTime: 傳入的時間, 格式為 hhmmss hComm: handle 值, 可以是 COM 或是 TCP/IP iTimeout: 等待逾時, 單位為 ms
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.3.3 hac34GetEEData(讀取裝置 EEPROM 值)

API 名稱	Hac34GetEEData
語法	<pre>int __stdcall hac34GetEEData (     int iNodeID,     char *cEEData,     int *iReceiveDataLen,     int iEEArea,     unsigned int *iEEAddr,     int *iEELen,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	讀取裝置 EEPROM 值
參數說明	<p>iNodeID: 裝置代號</p> <p>cEEData: 傳回的 EEPROM 內容</p> <p>iReceiveDataLen : 實際傳回 EEPROM 的長度</p> <p>iEEArea: 1:參數區, 2:卡號區, 3:刷卡記錄區</p> <p>iEEAddr: 要傳回 EEPROM 的地址</p> <p>iEELen: 要傳回 EEPROM 的長度，最長 64 個位元組</p> <p>hComm: handle 值, 可以是 COM 或是 TCP/IP</p> <p>iTimeout: 等待逾時, 單位為 ms</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	

#### 4.3.4 hac34SetEEData (寫入 EEPROM 值)

API 名稱	hacSetEEData
語法	<pre>int __stdcall hacSetEEData (     int iNodeID,     char *cEEData,     unsigned int iEEAddr,     int iEELen,     int iEEArea,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	寫入資料到裝置 EEPROM 內
參數說明	<p>iNodeID: 裝置代號</p> <p>cEEData: 要寫入的 EEPROM 內容，其內容為實際存放的值</p> <p>iEEArea: 1:參數區, 2:卡號區</p> <p>iEEAddr: 要寫入 EEPROM 的位址</p> <p>iEELen: 要寫入 EEPROM 的長度，最長 32 個位元組</p> <p>hComm: handle 值，可以是 COM 或是 TCP/IP</p> <p>iTimeout: 等待逾時，單位為 ms</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.3.5 hac34GetVersion(讀取裝置版本)

API 名稱	hac34GetVersion
語法	<pre>int __stdcall hac34GetVersion (     int iNodeID,     char *cData,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	讀取裝置版本
參數說明	iNodeID: 裝置代號 cData: 傳回裝置的版本/ROM File/日期, hComm: handle 值, 可以是 COM 或是 TCP/IP iTimeout: 等待逾時, 單位為 ms
回傳值	請參考第三章 錯誤回傳值說明
使用場合	第一部分: 機器型號 RAC-340PE , RAC-340PM RAC-520PE , RAC-520PM 第二部分: 主版本,次版本 V1.00 V2.18
備註	

#### 4.3.6 hac34AddCard(單筆新增卡號/非壓縮)

API 名稱	hac34AddCard
語法	<pre>int __stdcall hac34AddCard (     int iNodeID,     char *cCardNo,     int iCardLen,     char *cPassWord,     int iPassLen,     int iTimeZone,     char cStatus,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	單筆新增卡號
參數說明	<p>iNodeID: 裝置代號</p> <p>cCardNo: ASCII 碼的卡號</p> <p>char *cPassWord: ASCII 碼的密碼</p> <p>iCardLen: 卡號的長度，最大 10 位</p> <p>iPassLen: 密碼的長度，最大 4 位</p> <p>iTimeZone: 時段，保留不用填</p> <p>cStatus: 狀態碼，保留不用填</p> <p>hComm: handle 值，可以是 COM 或是 TCP/IP</p> <p>iTimeout: 等待逾時，單位為 ms</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.3.7 hac34DelCard(單筆刪除卡號/非壓縮)

API 名稱	hac34DelCard
語法	<pre>int __stdcall hac34DelCard (     int iNodeID,     char *cCardNo,     int iCardLen,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	單筆刪除卡號
參數說明	iNodeID: 裝置代號 cCardNo: 所要刪除的卡號 iCardLen: cCardNo 的長度 hComm: handle 值, 可以是 COM 或是 TCP/IP iTimeout: 等待逾時, 單位為 ms
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.3.8 hac34DelAllCard(刪除所有卡號)

API 名稱	hac34DelAllCard
語法	<pre>int __stdcall hac34DelAllCard (     int iNodeID,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	刪除所有合法卡號
參數說明	iNodeID: 裝置代號 hComm: handle 值, 可以是 COM 或是 TCP/IP iTimeout: 等待逾時, 單位為 ms, 注意此操作需要比較多時間, 最少 10 秒
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.3.9 hac34SetDoorRelayMode (設定開門 Relay 動作模式)

API 名稱	hac34SetDoorRelayMode
語法	<pre>int __stdcall hac34SetDoorRelayMode (     int iNodeID,     int ActionMode,     int ActionMask,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	單筆刪除壓縮卡號
參數說明	<p>iNodeID: 裝置代號</p> <p>ActionMode:</p> <ul style="list-style-type: none"> <li>1: Reader Beep / Alarm Relay ON</li> <li>2: 關閉緊急卡自動學習/開啟</li> <li>4: 顯示星號 / 顯示卡號</li> <li>8: 保留</li> <li>16: 語系</li> <li>32: 語系</li> <li>64: 外力破壞</li> <li>128: 門未關妥</li> </ul> <p>ActionMask: 遮蔽 ActionMode 用</p> <p>hComm: handle 值, 可以是 COM 或是 TCP/IP</p> <p>iTimeout: 等待逾時, 單位為 ms</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.3.10 hac34SetStatus (設定 Reader 狀態)

API 名稱	hac34SetStatus																
語法	<pre>int __stdcall hac34SetStatus (     int iNodeID,     int iStatus,     int iMask,     HANDLE hComm,     unsigned int iTimeout )</pre>																
目的	讀取裝置 RAM 中的資料																
參數說明	<p>iNodeID: 裝置代號</p> <p>iStatus: 設定之狀態值, 各位元代表意義下</p> <table> <tbody> <tr> <td>-Bit 0: 0→LCD 背光 Off</td> <td>1→LCD 背光 On</td> </tr> <tr> <td>-Bit 1: 0→鍵盤背光 Off</td> <td>1→鍵盤背光 On</td> </tr> <tr> <td>-Bit 2: 0→Disable 數字鍵</td> <td>1→Enable 數字鍵</td> </tr> <tr> <td>-Bit 3: 0→Disable 功能鍵</td> <td>1→Enable 功能鍵</td> </tr> <tr> <td>-Bit 4: 0→Disable Reader</td> <td>1→Enable Reader</td> </tr> <tr> <td>-Bit 5: 0→按鍵時不要嗶聲</td> <td>1→按鍵時要嗶聲</td> </tr> <tr> <td>-Bit 6: Don't Care</td> <td></td> </tr> <tr> <td>-Bit 7: 0→Disable Case Sensor</td> <td>1→Enable Case Sensor</td> </tr> </tbody> </table> <p>iMask: 對應 iStatus 之相對位元是否動作 1→動作 0→不動作 (Don't Care 的位元都是不動作)</p> <p>hComm: handle 值, 可以是 COM 或是 TCP/IP</p> <p>iTimeout: 等待逾時, 單位為 ms</p>	-Bit 0: 0→LCD 背光 Off	1→LCD 背光 On	-Bit 1: 0→鍵盤背光 Off	1→鍵盤背光 On	-Bit 2: 0→Disable 數字鍵	1→Enable 數字鍵	-Bit 3: 0→Disable 功能鍵	1→Enable 功能鍵	-Bit 4: 0→Disable Reader	1→Enable Reader	-Bit 5: 0→按鍵時不要嗶聲	1→按鍵時要嗶聲	-Bit 6: Don't Care		-Bit 7: 0→Disable Case Sensor	1→Enable Case Sensor
-Bit 0: 0→LCD 背光 Off	1→LCD 背光 On																
-Bit 1: 0→鍵盤背光 Off	1→鍵盤背光 On																
-Bit 2: 0→Disable 數字鍵	1→Enable 數字鍵																
-Bit 3: 0→Disable 功能鍵	1→Enable 功能鍵																
-Bit 4: 0→Disable Reader	1→Enable Reader																
-Bit 5: 0→按鍵時不要嗶聲	1→按鍵時要嗶聲																
-Bit 6: Don't Care																	
-Bit 7: 0→Disable Case Sensor	1→Enable Case Sensor																
回傳值	請參考第三章 錯誤回傳值說明																
使用場合																	
備註																	

#### 4.3.11 hac34SetReadCardParameter(設定 mifare 讀卡參數)

API 名稱	hac34SetReadCardParameter
語法	<pre>int __stdcall hac34SetReadCardParameter (     int iNodeID,     int iKeyType,     int iBlock,     int iStartDigit,     int iDigitLength,     int iCompact,     unsigned char *cKeyValue,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	設定 mifare 的讀卡相關參數
參數說明	<p>iNodeID: 裝置代號</p> <p>iKeyType: =0 讀 CSN        =1 KeyA        =2KeyB</p> <p>iBlock: 當 cKeyType&gt;0 時指出要讀哪個 Block ( 0x00~0x3F )</p> <p>iStartDigit: 由該 Block 第幾 Byte 開始讀取 (0~15)</p> <p>iDigitLength: 讀取幾位數 (0~12)</p> <p>iCompact: 卡號是否為壓縮格式    =0 非壓縮    =1 壓縮</p> <p>cKeyValue: 指到 mifare Key 值之指標, mifare Key 值為 6 bytes</p> <p>hComm: handle 值, 可以是 COM 或是 TCP/IP</p> <p>iTimeout: 等待逾時, 單位為 ms</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.3.12 hac34GetReadCardParameter(讀取 mifare 讀卡參數)

API 名稱	hac34GetReadCardParameter
語法	<pre>int __stdcall hac34GetReadCardParameter (     HANDLE hComm,     int iNodeID,     int *iKeyType,     int *iBlock,     int *iStartDigit,     int *iDigitLength,     int *iCompact,     unsigned int iTimeout )</pre>
目的	讀取 mifare 的讀卡相關參數
參數說明	<p>hComm: handle 值, 可以是 COM 或是 TCP/IP  iNodeID: 裝置代號  iKeyType: =0 讀 CSN        =1 KeyA        =2KeyB  iBlock: 當 cKeyType&gt;0 時指出要讀哪個 Block ( 0x00~0x3F )  iStartDigit: 由該 Block 第幾 Byte 開始讀取 (0~15)  iDigitLength: 讀取幾位數 (0~12)  iCompact: 卡號是否為壓縮格式        =0 非壓縮        =1 壓縮  iTimeout: 等待逾時, 單位為 ms</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.3.13 hac34GetIO (讀取裝置上目前 IO 狀態)

API 名稱	hac34GetIO
語法	<pre>int __stdcall hac34GetIO (     int iNodeID,     char *cStatus,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	讀取裝置上目前 IO 狀態
參數說明	iNodeID: 裝置代號 cStatus: 傳回 IO 狀態, 2 個 Byte hComm: handle 值, 可以是 COM 或是 TCP/IP iTimeout: 等待逾時, 單位為 ms
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

#### 4.3.14 hac34Dump( DUMP 裝置上相關資訊)

API 名稱	hac34Dump
語法	<pre>int __stdcall hac34Dump (     int iNodeID,     int iDumpArea,     char *cDumpData,     HANDLE hComm,     unsigned int iTimeout )</pre>
目的	將裝置上的卡號,刷卡記錄 DUMP 回電腦
參數說明	<p>iNodeID: 裝置代號</p> <p>iDumpArea: 1 卡號, 2 刷卡記錄</p> <p>cDumpData: 傳回 Dump 的資料</p> <p>hComm: handle 值, 可以是 COM 或是 TCP/IP</p> <p>iTimeout: 等待逾時, 單位為 ms</p>
回傳值	請參考第三章 錯誤回傳值說明
使用場合	
備註	

## 第五章 附錄

### 5.1 RAC-340 相關記憶體規劃

參數地址	說明	備註
0x00	裝置型號 (唯讀)	0=RAC-340PM 1=RAC-340PE 2=RAC-520PM 3=RAC-520PE 4=保留 5=SHR-100 6=RAC-920PM 7=RAC-920PE 8=RAC-820PM 9=RAC-820PE 10=RAC-930PM 11=RAC-930PE
0x01	主版本 (唯讀)	BCD 格式
0x02	次版本 (唯讀)	BCD 格式
0x03~0x09	保留	
0x0A	Sensor 動作	-Bit0 : Push Button 1→NO 0→NC (內定值 1) -Bit1 : 未使用 -Bit2 : Door Sensor 1→NC 0→NO (內定值 1) -Bit3~7 : 未使用
0x0B	CardJudgeMode	=0 正常模式 =1 任意卡模式 1(mifare Key 對即合法,沒有刷卡紀錄) =2 任意卡模式 2(會檢查卡號格式,合法卡空間變為黑名單)
0x0C~0x0D	開門 Relay 時間	低位在前, (000.00~655.35 sec)
0x0E~0x0F	警報 Relay 時間	低位在前, (000.00~655.35 sec)
0x10~0x11	門偵測時間	低位在前, (000.01~655.35 sec , 0:OFF)
0x12	脅迫碼長度	( 1~10)
0x13~0x1C	脅迫碼	不足請補 0xFF
0x1D	解除碼長度	( 1~10)
0x1E~0x27	解除碼	不足請補 0xFF
0x28	控制功能 1	-Bit 0: 0→Alarm Relay On 1→Reader Beep(預設) -Bit 1: 保留 -Bit 2: 0→顯示卡號(預設) 1→顯示星號 -Bit 3: 保留 -Bit 4~5: 0 繁體 1 英語(預設) 2 簡體 -Bit 6: 0→門被外力破壞 Enable(預設)

		1→門被外力破壞 Disable -Bit 7: 0→門未關妥功能 Enable(預設) 1→門未關妥功能 Disable
0x29	Mifare 取碼模式 (僅支援 Mifare 機種)	Mifare 取碼方式 0: 自動取碼 1:固定長度取碼
0x2A	控制功能 2	-Bit 0: 0→LCD 背光 Off 1→LCD 背光 On -Bit 1: 0→鍵盤背光 Off 1→鍵盤背光 On -Bit 2: 0→Disable 數位鍵 1→Enable 數位鍵 -Bit 3: 0→Disable 功能鍵 1→Enable 功能鍵 -Bit 4: 0→Disable Reader 1→Enable Reader -Bit 5: 0→按鍵時不要嗶聲 1→按鍵時要嗶聲 -Bit 6: 保留 -Bit 7: 0→Disable Case Sensor 1→Enable Case Sensor
0x2B	比較有效位數	MSB 4 bit : 比較起始位數-1 LSB 4 bit : 比較位數 0~9 碼 (0 代表 10 碼全部比較)
0x2C	重複刷卡檢查時間	(1~255sec , 0:OFF)
0x2D	MifareReadType (僅支援 Mifare 機種)	0:序號 1:KeyA 2:KeyB
0x2E	MifareReadAddressAndLength (僅支援 Mifare 機種)	MS4Bit:取碼地址 LS4B:取碼長度
0x2F	MifareCompact (僅支援 Mifare 機種)	0:非壓縮 1:壓縮
0x30	MifareReadBlock (僅支援 Mifare 機種)	讀取 Block ( 0x1~0x3F )
0x31~0x37	保留	
0x38	母卡長度	( 1~10 )
0x39~0x42	母卡號碼	不足請補 0xFF
0x43	刷母卡開門模式	0 不開門 1 開門
0x44	卡片模式	0 標準 1 精簡 2 共通
0x45	刷卡開門模式	0 刷卡開門 1 刷卡或按鍵 2 加按 PIN
0x46	是否儲存非法卡	0 不儲存 1 儲存

0x47~0x48	上班切換時分	BCD 格式
0x49~0x4A	下班切換時分	BCD 格式
0x4B~0x4C	持續開門秒數	低位在前 (1~9999sec)
0x4D	保留	
0x4E	保留	
0x4F	保留	
0x50~0x6F	響鈴時間 + 響鈴分秒 每組 4 Bytes, 共 8 組	BCD 格式
0x70	星期幾要響鈴	
0x71	門禁/考勤模式設定	0 門禁      1 考勤

## 5.2 RAC-520 相關記憶體規劃

參數地址	說明	備註	
0x00	裝置型號 (唯讀)	0=RAC-340PM 2=RAC-520PM 4=保留 6=RAC-920PM 8=RAC-820PM 10=RAC-930PM	1=RAC-340PE 3=RAC-520PE 5=SHR-100 7=RAC-920PE 9=RAC-820PE 11=RAC-930PE
0x01	主版本 (唯讀)	BCD 格式	Ex: 主版次 01h, 副版次 10h 代表 v1.10
0x02	次版本 (唯讀)	BCD 格式	Ex: 主版次 11h, 副版次 02h 代表 v11.02
0x03~0x09	保留		
0x0A	Sensor 動作	-Bit0 : Push Button 1→NO 0→NC (內定值 1) -Bit1 : 未使用 -Bit2 : Door Sensor 1→NC 0→NO (內定值 1) -Bit3~7 : 未使用	
0x0B	CardJudgeMode	=0 正常模式 =1 任意卡模式 1(mifare Key 對即合法,沒有刷卡紀錄) =2 任意卡模式 2(會檢查卡號格式,合法卡空間變為黑名單)	
0x0C~0x0D	開門 Relay 時間	低位在前, (000.00~655.35 sec)	
0x0E~0x0F	警報 Relay 時間	低位在前, (000.00~655.35 sec)	
0x10~0x11	門偵測時間	低位在前, (000.01~655.35 sec , 0:OFF)	
0x12	脅迫碼長度	( 1~10)	
0x13~0x1C	脅迫碼	不足請補 0xFF	
0x1D	解除碼長度	( 1~10)	
0x1E~0x27	解除碼	不足請補 0xFF	
0x28	控制功能 1	-Bit 0: 0→Alarm Relay On 1→Reader Beep(預設) -Bit 1~5: 保留 -Bit 6: 0→門被外力破壞 Enable(預設) 1→門被外力破壞 Disable -Bit 7: 0→門未關妥功能 Enable(預設) 1→門未關妥功能 Disable	
0x29	Mifare 取碼模式 (僅支援 Mifare 機種)	Mifare 取碼方式 0: 自動取碼 1:固定長度取碼	
0x2A	控制功能 2	-Bit 0: 保留	

		<ul style="list-style-type: none"> <li>-Bit 1: 0→鍵盤背光 Off 1→鍵盤背光 On</li> <li>-Bit 2: 0→Disable 數字鍵 1→Enable 數字鍵</li> <li>-Bit 3: 保留</li> <li>-Bit 4: 0→Disable Reader 1→Enable Reader</li> <li>-Bit 5: 0→按鍵時不要嗶聲 1→按鍵時要嗶聲</li> <li>-Bit 6: 保留</li> <li>-Bit 7: 0→Disable Case Sensor 1→Enable Case Sensor</li> </ul>
0x2B	比較有效位數	MSB 4 bit : 比較起始位數-1 LSB 4 bit : 比較位數0~9 碼 (0 代表10碼全部比較)
0x2C	重複刷卡檢查時間	(1~255sec , 0:OFF)
0x2D	MifareReadType (僅支援 Mifare 機種)	0:序號 1:KeyA 2:KeyB
0x2E	MifareReadAddressAndLength (僅支援 Mifare 機種)	MS4Bit:取碼地址 LS4B:取碼長度
0x2F	MifareCompact (僅支援 Mifare 機種)	0:非壓縮 1:壓縮
0x30	MifareReadBlock (僅支援 Mifare 機種)	讀取 Block ( 0x1~0x3F)
0x31~0x37	保留	
0x38	母卡長度	( 1~10)
0x39~0x42	母卡號碼	不足請補 0xFF
0x43	刷母卡開門模式	0 不開門 1 開門
0x44	卡片模式	0 標準 1 精簡 2 共通
0x45	刷卡開門模式	0 刷卡開門 1 刷卡或按鍵 2 加按 PIN
0x46	是否儲存非法卡	0 不儲存 1 儲存
0x47~0x48	保留	
0x49~0x4A	保留	
0x4B~0x4C	持續開門秒數	低位在前 (1~9999sec)

### 5.3 RAC-820/920/930 相關記憶體規劃

參數地址	說明	備註	
0x00	裝置型號 (唯讀)	0=RAC-340PM 2=RAC-520PM 4=保留 6=RAC-920PM 8=RAC-820PM 10=RAC-930PM	1=RAC-340PE 3=RAC-520PE 5=SHR-100 7=RAC-920PE 9=RAC-820PE 11=RAC-930PE
0x01	主版本 (唯讀)	BCD 格式	Ex: 主版次 01h, 副版次 10h 代表 v1.10
0x02	次版本 (唯讀)	BCD 格式	Ex: 主版次 11h, 副版次 02h 代表 v11.02
0x03~0x09	保留		
0x0A	Sensor 動作	-Bit0 : Push Button 1→NO 0→NC (內定值 1) -Bit1 : 未使用 -Bit2 : Door Sensor 1→NC 0→NO (內定值 1) -Bit3~7 : 未使用	
0x0B	CardJudgeMode	=0 正常模式 =1 任意卡模式 1(mifare Key 對即合法,沒有刷卡紀錄) =2 任意卡模式 2(會檢查卡號格式,合法卡空間變為黑名單)	
0x0C~0x0D	開門 Relay 時間	低位在前, (000.00~655.35 sec)	
0x0E~0x0F	警報 Relay 時間	低位在前, (000.00~655.35 sec)	
0x10~0x11	門偵測時間	低位在前, (000.01~655.35 sec , 0:OFF)	
0x12	脅迫碼長度	( 1~10)	
0x13~0x1C	脅迫碼	不足請補 0xFF	
0x1D	解除碼長度	( 1~10)	
0x1E~0x27	解除碼	不足請補 0xFF	
0x28	控制功能 1	-Bit 0: 0→Alarm Relay On 1→Reader Beep(預設) -Bit 1~5: 保留 -Bit 6: 0→門被外力破壞 Enable(預設) 1→門被外力破壞 Disable -Bit 7: 0→門未關妥功能 Enable(預設) 1→門未關妥功能 Disable	
0x29	Mifare 取碼模式 (僅支援 Mifare 機種)	Mifare 取碼方式 0: 自動取碼 1:固定長度取碼	
0x2A	控制功能 2	-Bit 0: 保留	

		<ul style="list-style-type: none"> <li>-Bit 1: 0→鍵盤背光 Off 1→鍵盤背光 On</li> <li>-Bit 2: 0→Disable 數字鍵 1→Enable 數字鍵</li> <li>-Bit 3: 保留</li> <li>-Bit 4: 0→Disable Reader 1→Enable Reader</li> <li>-Bit 5: 0→按鍵時不要嗶聲 1→按鍵時要嗶聲</li> <li>-Bit 6: 保留</li> <li>-Bit 7: 0→Disable Case Sensor 1→Enable Case Sensor</li> </ul>
0x2B	比較有效位數	MSB 4 bit : 比較起始位數-1 LSB 4 bit : 比較位數0~9 碼 (0 代表10碼全部比較)
0x2C	重複刷卡檢查時間	(1~255sec , 0:OFF)
0x2D	MifareReadType (僅支援 Mifare 機種)	0:序號 1:KeyA 2:KeyB
0x2E	MifareReadAddressAndLength (僅支援 Mifare 機種)	MS4Bit:取碼地址 LS4B:取碼長度
0x2F	MifareCompact (僅支援 Mifare 機種)	0:非壓縮 1:壓縮
0x30	MifareReadBlock (僅支援 Mifare 機種)	讀取 Block ( 0x1~0x3F)
0x31~0x37	保留	
0x38	母卡長度	( 1~10)
0x39~0x42	母卡號碼	不足請補 0xFF
0x43	刷母卡開門模式	0 不開門 1 開門
0x44	卡片模式	0 標準 1 精簡 2 共通
0x45	刷卡開門模式	0 刷卡開門 1 刷卡或按鍵 2 加按 PIN
0x46	是否儲存非法卡	0 不儲存 1 儲存
0x47~0x48	保留	
0x49~0x4A	保留	
0x4B~0x4C	持續開門秒數	低位在前 (1~9999sec)
0x4D	副卡機及 Anti 設定	<ul style="list-style-type: none"> <li>bit 0: =0 副卡機不需檢查 =1 副卡機需檢查密碼</li> <li>bit 1: 保留</li> <li>bit 2: =0 不 Anti-Passback =1 啓動 Anti-Passback</li> </ul>

0x4E	內建 Relay 功能	=0 開門 Relay =1 門鈴 Relay (ACU-30 Relay1: Door , Relay 2: Alarm/響鈴)
------	-------------	--