

Windows 版本

HDACS 控制軟體 API 參考檔

Version 2.4.3

目 錄

第一章 系統需求.....	5
第二章 支援機型列表.....	5
第三章 錯誤傳回值說明	6
第四章 RAC 系列相關函數說明	7
4.1. 與通訊相關 API(Level 0)	7
4.1.1 hacOpenChannelEX(開啟通訊)	7
4.1.2 hacCloseChannel(關閉通訊).....	9
4.2. 與刷卡及事件接收相關 API (Level1).....	10
4.2.1 hacPolling (Polling 設備).....	10
4.3. 與硬體相關 API(Level 2)	13
4.3.1 hacGetDateTime(讀取裝置上的日期時間)	13
4.3.2 hacSetDateTime(將日期時間寫入裝置)	15
4.3.3 hacGetVersion(讀取裝置版本).....	17
4.3.4 hacRelayAction(設定 Relay 狀態)	19
4.3.5 hacAddCard(單筆新增卡號/非壓縮)	21
4.3.6 hacDelCard(單筆刪除卡號/非壓縮)	23
4.3.7 hacGetSensor (讀取 Sensor 狀態).....	25
4.3.8 hacGetEEData(讀取裝置 EEPROM 值)	27
4.3.9 hacSetEEData (寫入 EEPROM 值).....	29
4.3.10 hacAddZCard(單筆新增壓縮卡號)	31
4.3.11 hacDelZCard(單筆刪除壓縮卡號)	33
4.3.12 hacGetRAMData(讀取裝置中 RAM 的內資料)	35
4.3.13 hacSetRAMData (寫入 RAM 值)	37
4.3.14 hacAddVisitorCard(單筆新增來賓卡)	39
4.3.15 hacDelVisitorCard(單筆刪除來賓卡)	41
4.3.16 hacGetFlashData(讀取裝置 Flash 值).....	43
4.3.17 hacSetFlashData (寫入 Flash 值)	45
4.3.18 hacGetParaData(讀取裝置一般 Para 值)	47
4.3.19 hacSetParaData (寫入一般 Para 值).....	49
4.3.20 hacGetSysParaData(讀取裝置系統參數值)	51
4.3.21 hacSetSysParaData (寫入系統參數值).....	53
4.3.22 hacGetMifare(讀取 Mifare 模組讀卡參數值).....	55
4.3.23 hacSetMifare (寫入 Mifare 模組讀卡參數值)	57
4.3.24 hacAddCardEX(單筆新增卡號附加姓名顯示).....	59
4.3.25 hacInitial(初始化設備)	61
4.3.26 hacFingerPrinterQueryMasterFP (查詢 Master 指紋)	62
4.3.27 hacFingerPrinterUpdateMasterFP (更新 Master 指紋).....	63
4.3.28 hacFingerPrinterDeleteFP (刪除 Master 指紋).....	64

4.3.29 hacAddCardFingerPrintEx (上傳指紋特徵值).....	65
4.3.30 hacFingerPrintQueryUser (收回指紋特徵值).....	65
4.4 參數表相關 API(Level 3)	67
4.4.1 hacSetLanMode (設定主動發送模式)	67
4.4.2 hacSetReader (設定 RAC-2200 下 Reader 的參數).....	69
第五章 HDE-100 系列相關函數說明	71
5.1. 通訊相關 API.....	71
5.1.1 hsOpenChannel (開啟通訊聯機).....	71
5.1.2 hsCloseChannel(關閉與 HDE-100 系列聯機的通訊).....	73
5.1.3 hsELWriteTable(傳送表格到 HDE-100 系列)	74
5.1.4 hsELReadTable(讀取 HDE-100 表格).....	76
5.1.5 hsELPolling (讀取 HDE-100 系列主機上的刷卡及事件記錄)	78
5.2. Read/Write 相關 API.....	81
5.2.1 hsELReadParameter(讀取系統參數)	81
5.2.2 hsELWriteParameter (設定系統參數)	83
5.2.3 hsELInitialize (系統初始化)	85
5.2.4 hsELGetInfo (讀取 HDE-100 系列主機版本及其他資訊)	87
5.2.5 hsELAddAuthorization (合法卡多筆新增).....	89
5.2.6 hsELDeleteAuthorization (單筆刪除)	91
5.2.7 hsELQueryAuthorization (查詢單筆卡號許可權)	93
5.2.8 hsELDeleteAllAuthorization (刪除全部合法卡).....	95
5.2.9 hsELReadDeviceInfo(讀取裝置型號)	96
5.2.10 hsELSetTime (設定主機時間)	98
5.2.11 hsELGetTime (讀取主機時間).....	100
5.2.12 hsELReleaseAlarm (解除警報)	102
5.2.13 hsELPublicFloor (設定公共樓層).....	104
5.2.14 hsELSetReader(設定下層讀卡器).....	106
第六章 ECU-680 系列相關函數說明	108
6.1. Read/Write 相關 API.....	108
6.1.1 hsECUReadIO (讀取 I/O 狀態)	108
6.1.2 hsECUReadParamenter (讀取參數表全部資料)	109
6.1.3 hsECUReadPower (讀取電源狀態(帶卡號)).....	110
6.1.4 hsECUAddCard (新增單筆合法卡號)	111
6.1.5 hsECUPolling (Polling ECU-680 設備).....	112
第七章 附錄	113
附錄 1：RAC-2000 系列格式說明	113
1.1 基本格式	113
1.2 RAC-2000 系列 合法卡許可權格式說明	113
1.3 RAC-2000G 合法卡許可權格式說明	114
1.4 RAC-2000 系列刷卡歷史資料	114

1.5 記憶體使用對照表.....	116
1.6 時段表對照說明.....	117
1.7 假日表對照說明.....	118
1.8 來賓卡格式.....	119
1.9 事件碼回傳值(Sensor/Relay)	119
1.10 EEPROM 相關記憶體規劃	120
Wiegand 解碼設定 (位址 1A00H~1BDFH:共 480Bytes)	124
1.11 AntiPassback 設定說明.....	125
1.12 RAC-2000G SRAM 相關記憶體規劃	125
1.13 RAC-2000 系列 SRAM 相關內存規劃	127
附錄 2：HDE-100 格式說明	128
2.1 回傳錯誤代碼表.....	128
2.2 參數表.....	128
2.3 資料格式	129
附錄 3：RAC-960/RAC-970 格式說明	135
3.1 基本格式	135
3.2 卡號權限格式	135
3.3 刷卡歷史資料.....	136
3.4 時段表對照說明.....	138
3.5 持續開門	139
3.6 事件碼回傳值(Sensor/Relay)	139
3.7 Flash Memory 相關記憶體規畫.....	140
附錄 4：RAC-940 格式說明	150
4.1 卡號權限格式	150
4.2 刷卡及事件資料格式	151
4.3 系統參數表 (位址 0000H~0013H：共 20Bytes)	153
4.4 一般參數表 (位址 0014H~00FFH：共 236Bytes)	154
4.5 時段表一 (位址 0100~016F：共 112 Bytes,16 組).....	158
4.6 時間表一 (位址 0480H~04BFH：共 64Bytes ,16 組).....	158
4.7 假日表 (位址 0680H~07FFH：共 384Bytes,64*2 組).....	158
4.8 持續開門時段 (位址 08D8H~08DFH：共 8Bytes, 8 組).....	159
4.9 響鈴時間表 (位址 08E0H~0907H：共 40Bytes, 8 組).....	159
4.10 Anti Reset 時間表 (位址 0980H~098FH：共 16Bytes, 8 組).....	161
4.11 顯示訊息 (位址 0A00H~0A7FH：共 128Bytes, 8 組).....	161
4.12 密碼時段 (位址 0B00H~0B07H：共 8Bytes, 8 組).....	162
4.13 警報時段 (位址 0C00H~0C03H：共 4Bytes, 1 組)	162
4.14 時段表二 (位址 0C04~0C73：共 112 Bytes,16 組).....	162
4.15 時間表二 (位址 0CE4~0EE3：共 512Bytes,16 組).....	163
附錄 5：RAC-2200 格式說明	164
5.1 基本格式	164

5.2 卡號權限格式	164
5.3 刷卡及事件資料格式	165
5.4 時段表對照說明.....	166
5.5 RAC2200 ANTI (下載此 Byte 時請填入 00h).....	169
5.6 事件碼回傳值(Sensor/Relay)	169
5.7 EEPROM 相關記憶體規劃	170
5.8 SRAM 相關記憶體規劃	172
附錄 6：ECU-680 格式說明	173
6.1 合法卡格式.....	173
6.2 刷卡記錄資料格式	174
6.3 參數表 (位址 0000H~001FH:共 32Bytes)	175
6.4 時段表 (位址 0020~0057H:共 56 Bytes,8 組)	176
6.5 時間表 (位址 0058H~00D7H:共 128Bytes,8 組)	177
6.6 不管制時段 (位址 00D8H~00DBH:共 4Bytes, 4 組).....	178
6.7 設定卡 (位址 00DCH~0117H:共 60Bytes).....	178
附錄 7：RAC-820PEF/RAC-820PMF 格式說明	179
7.1 卡號權限格式	179
7.2 刷卡歷史資料	179
7.3 時段表對照說明.....	182
7.4 持續開門	183
7.5 事件碼回傳值(Sensor/Relay)	183
7.6 Flash Memory 相關記憶體規畫	184

手冊版本記錄

版本	修改記錄
V2.0	初版
V2.1	增加 ECU-680
V2.21	修改 hacInitial 定義值錯誤
V2.3	新增 4.3.29 hacAddCardFingerPrintEx (上傳指紋特徵值)
V2.4	新增 RAC-970 系列/RAC-2000WSN 系列
V2.4.2	新增 RAC-852 系列
V2.4.3	新增 RAC-820PxP 系列

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 C#2005。

第二章 支援機型列表

RAC-2000 系列	:	RAC-2000G, RAC-2000P, RAC-2000PS RAC-2000PV, RAC-2000PN, RAC-2000PSN, RAC-2000WS, RAC-2000WSN, HDP-100, HDP-100S
HDE 系列	:	HDE-100
RAC-960 系列	:	RAC-960PE, RAC-960PM RAC-960PEF, RAC-960PMF
RAC-970 系列	:	RAC-970PE, RAC-970PM RAC-970PEF, RAC-970PMF
RAC-940 系列	:	RAC-940PE, RAC-940PM
RAC-2200 系列	:	RAC-2200
RAC-2400	:	RAC-2400(4), RAC-2400(8), RAC-2400(16) 由於與上述其它機型 SDK 使用方法不同，故將 RAC-2400 的部份做獨立說明，請參考 Hdacs_rac2400*.pdf
ECU-680 系列	:	ECU-680PM
RAC-852 系列	:	RAC-852PEFV, RAC-852PMFV, RAC-852PHFV
RAC-820Px F 系列	:	RAC-820PMF/RAC-820PEF

第三章 錯誤傳回值說明

定義	返回值	說明
HF_RET_SUCCESS	0	成功
HF_ERR_HANDLE_WAIT_TIMEOUT	1125	對多線程同步物件訪問時出現等待超時。
HF_ERR_HANDLE_RELEASE	1126	釋放多線程同步物件時出現錯誤。
HF_ERR_PARAMETER	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 。
HF_ERR_SOCKET_ERROR	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。
HF_ERR_DATA_LENGTH	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。
HF_ERR_RESPOND_LENGTH	1103	設備回應封包長度小於要求的長度。
HF_ERR_HANDLE_INVALID	1004	傳入的控制控制碼無效，無效的 Handle 。
HF_ERR_RESPOND_ENDCHAR	1005	設備回應封包結束符錯誤。
HF_ERR_RESPOND_CRC16	1006	設備回應封包 CRC-16 檢查錯誤。
HF_ERR_SEND_CRC16	1106	電腦發送的封包 CRC-16 檢查錯誤。
HF_ERR_SEND_CMD	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。
HF_ERR_SEND_RW	1008	下層設備進行 R/W 讀寫時出現錯誤。
HF_ERR_SEND_OVERLENGTH	1009	電腦發送給設備的封包過長。
HF_ERR_RESPOND_NORECORDS	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。
HF_ERR_RESPOND_EXCEPT	4445	讀取設備中的資料或者記錄時發生異常錯誤。
HF_ERR_WAIT_TIMEOUT	1025	執行非同步讀寫的時候出現超時錯誤，超過等待時間(iTimeout)。
HF_ERR_WAIT_FAILED	1026	執行非同步讀寫的時候出現等待錯誤。
HF_ERR_WAIT_NODATA	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 hacOpenChannelEX(開啟通訊)

支援機型	RAC-2000 系列,RAC-940 系列,RAC-960/970 系列,RAC-2200 系列, RAC-852 系列, RAC-820PxP 系列	
API 名稱	hacOpenChannelEX	
語法	int __stdcall hacOpenChannelEX (char *sComm, unsigned int iPort, int iCheckStatus, HANDLE *hComm, unsigned int iTimeout)	
目的	開啟通訊連接, 可以 TCP/IP 或 COM Port	
參數說明	sComm	開啟 COM 通訊埠時 COM1-COM128 開啟 TCP/IP 時 : sComm : IP 地址, 如 172.16.1.1
	iPort	開啟 COM 通訊埠時 : iPort 為傳輸速度 (1200/2400/4800/9600/19200/38400) RAC-2000 系列默認是 19200 開啟 TCP/IP 時 : iPort : 埠號, 如 4660
	iCheckStatus	當通過網路連接時使用此項參數, 為 2 表示要檢測 網路連接是否被網路設備關閉, 一般使用 BF430 時 要進行此項檢測; 為 1 表示不需要檢測
	hComm	開啟成功後傳回的 handle 值 開啟成功後,傳回 socket 值
	iTimeout	當通過網路連接時使用此項參數, 連接的超時時 間, 單位為毫秒
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	

使用場合	<p>在執行其他功能前，必需先呼叫此功能,方可正常運作，此功能就像一個通道,不管是下命令或是接收資料,都必需先呼叫此功能。</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> <p>如果發現連接不正常，請關閉防火牆再試驗。</p> <p>為保持相容性，hacOpenChannel 函數依然保留</p>
備註	<p>功能說明：</p> <p>開啟 RS-232 通訊埠(COM1), 或者 TCP/IP 通訊介面 (Port 4660), 執行成功後，呼叫 htaCloseChannel 關閉開啟的通訊。</p> <p>VC Code：</p> <pre> int iReturn=0; char clP[20]; memset(clP,0,sizeof(clP)); int iPort=0; int iPort=0; HANDLE ghComm=NULL; iPort=GetDlgItemInt(IDC_EDITPort,NULL); iReturn =hacOpenChannelEX(clP,iPort,2,&ghComm,5000); if (iReturn!=0) { strcpy(sDisplay,"Open Channel： Error Number："); sprintf(sDisplay+strlen(sDisplay),"%04X", (WORD)iReturn); } else { strcpy(sDisplay,"Open Channel： OK!,HANDLE："); sprintf(sDisplay+strlen(sDisplay),"%04X", (WORD)ghComm); } iReturn =hacCloseChannel(ghComm); if (iReturn!=0) </pre>

--	--

4.1.2 hacCloseChannel(關閉通訊)

支援機型	RAC-2000 系列,RAC-940 系列,RAC-960/970 系列,RAC-2200 系列, RAC-852 系列, RAC-820PxP 系列	
API 名稱	hacCloseChannel	
語法	int __stdcall hacCloseChannel (HANDLE hComm)	
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	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註		

4.2. 與刷卡及事件接收相關 API (Level1)

4.2.1 hacPolling (Polling 設備)

支援機型	RAC-2000 系列,RAC-940 系列,RAC-960/970 系列,RAC-2200 系列, RAC-852 系列, RAC-820PxP 系列	
API 名稱	hacPolling	
語法	<pre>int __stdcall hacPolling (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	iNodeID	裝置代號
	iPrevRecord	上一次收到的資料筆數
	stRecord	存放傳回資料，是個結構
	iRecord	傳回的事件筆數
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
	iCardType	此參數低於 4 為表示卡號狀態， 0 表示 RAC-2000 系列卡號不壓縮，1 表示 RAC-2000 系列(RAC-2000G)卡號壓縮，2 表示 RAC2200 卡號接收模式，3 表示 RACC940/960/970；此參數高於 4 位元表示資料接收模式，當此參數值第 4 位為 1 時（0x10），表示設備主動發送模式，注意主動發送模式只有 RAC-2000PN 支援。例如 0x00,0x01 分別表示正常模式下卡號不壓縮和壓縮；0x10,0x11 分別表示主動發送模式下的不壓縮和壓縮。一般都是正常模式。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	

使用場合	<pre> { char cEventCode[5]; //事件代碼 在使用 RAC940/960/970 設備時，如果是 刷卡事件，則此事件代碼的前面兩個字元 表示班別，後面兩個字元表示刷卡狀態。 其他設備的前面兩個字元固定為“00”。 char cDateTime[20]; //日期時間 char cCard[20]; //卡號 char cDeviceID[10]; //設備 ID char cReaderID[10]; //讀頭 ID } stPollList; </pre> <p>注意：RAC-2000P/PV/PS/PN/PSN/WS/WSN, HDP-100/100S 只支援不壓縮卡號，RAC-2000G 只支援壓縮卡號</p>
備註	<p>VC Code：</p> <pre> BOOL Polling(int iDevice,int iCardType) { int iReturn=0; int iRecord=0; int iLoop=0; stPollList stRecord[256]; if (giDeviceType==1) { iReturn =hac34Polling (iDevice,giPrevRecord[iDevice],stRecord, &iRecord,ghComm,100,iCardType); } else { iReturn =hacPolling (iDevice,giPrevRecord[iDevice],stRecord, &iRecord,ghComm,100,iCardType); } if (iReturn!=0) </pre>

	<pre>{ return false; } giPrevRecord[iDevice]=iRecord; for(iLoop=0;iLoop<iRecord;iLoop++) { //Loop AddList AddListViewItems(hWndListView,&stRecord[iLoop]); } return true; }</pre>
--	--

4.3. 與硬體相關 API(Level 2)

4.3.1 hacGetDateTime(讀取裝置上的日期時間)

支援機型	RAC-2000 系列,RAC-940 系列,RAC-960/970 系列,RAC-2200 系列, RAC-852 系列, RAC-820PxP 系列	
API 名稱	HacGetDateTime	
語法	<pre>int __stdcall hacGetDateTime (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 <code>yyyymmddw</code>
	cTime	傳回的時間, 格式為 <code>hhmmss</code>
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 <code>HF_RET_SUCESS</code> , 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>VC Code :</p> <pre>int iReturn=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); char cDate[20]; char cTime[20]; memset(cDate,0,sizeof(cDate)); memset(cTime,0,sizeof(cTime)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); iReturn</pre>	

```

=hacGetDateTime(iNodeID,cDate,cTime,ghComm,1000);
    if (iReturn==0)
    {
        strcpy(sDisplay,"Device Time : ");
        strcat(sDisplay,cDate);
        strcat(sDisplay," ");
        strcat(sDisplay,cTime);
    }
    else
    {
        strcpy(sDisplay,"Get DateTime : Error Number : ");
        sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
    }
    SetDlgItemText(IDC_EDITReturn,sDisplay);

```

4.3.2 hacSetDateTime(將日期時間寫入裝置)

支援機型	RAC-2000 系列,RAC-940 系列,RAC-960/970 系列,RAC-2200 系列, RAC-852 系列, RAC-820PxP 系列	
API 名稱	HacSetDateTime	
語法	<pre>int __stdcall hacSetDateTime (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 <code>yyyymmddw</code>
	cTime	傳入的時間, 格式為 <code>hhmmss</code>
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 <code>HF_RET_SUCESS</code> , 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<pre>VC Code : int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); char cDate[20]; char cTime[20]; memset(cDate,0,sizeof(cDate)); memset(cTime,0,sizeof(cTime)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iPos=0; SYSTEMTIME st; GetLocalTime(&st);</pre>	


```

iPos=0;
sprintf(cDate+iPos,"%04d",st.wYear);
iPos=iPos+4;
sprintf(cDate+iPos,"%02d",st.wMonth);
iPos=iPos+2;
sprintf(cDate+iPos,"%02d",st.wDay);
iPos=iPos+2;
sprintf(cDate+iPos,"%01d",(st.wDayOfWeek));

iPos=0;
sprintf(cTime+iPos,"%02d",st.wHour );
iPos=iPos+2;
sprintf(cTime+iPos,"%02d",st.wMinute );
iPos=iPos+2;
sprintf(cTime+iPos,"%02d",st.wSecond );

iReturn =hacSetDateTime(iNodeID,cDate,cTime,ghComm,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"Set DateTime : OK.");
}

```

4.3.3 hacGetVersion(讀取裝置版本)

支援機型	RAC-2000 系列,RAC-940 系列,RAC-960/970 系列,RAC-2200 系列, RAC-852 系列	
API 名稱	hacGetVersion	
語法	<pre>int __stdcall hacGetVersion (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	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合	第一部分： 機器型號 例如： RAC-2000P 第二部分： 主版本,次版本。例如： V1.00 第三部分： 年,月,日 例如： 2003/01/27	
備註	VC Code : <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); char cReceiveBuff[128]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iLoop=0; int iReadLen=0; int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); iReturn =hacGetVersion(iNodeID,cReceiveBuff,ghComm,1000); if (iReturn==0) {</pre>	

	<pre>strcpy(sDisplay,"Get Version : "); strcat(sDisplay,cReceiveBuff); } else { strcpy(sDisplay,"Get Version : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay);</pre>
--	---

4.3.4 hacRelayAction(設定 Relay 狀態)

支援機型	RAC-2000 系列,RAC-940 系列,RAC-960/970 系列,RAC-2200 系列, RAC-852 系列, RAC-820PxP 系列	
API 名稱	hacRelayAction	
語法	<pre>int __stdcall hacRelayAction (int iNodeID, char cAction, char cMask, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	設定 Relay 狀態, 直接動作 Relay	
參數說明	iNodeID	裝置代號
	cAction	設定 Relay 的動作的方式, (開門 Relay ON/OFF, 警報 Relay ON/OFF) Byte1 : RAC-2000 系列,RAC-2200 系列 : Bit 0 : Relay1 0=OFF 1=ON(Door1) Bit 1 : Relay2 0=OFF 1=ON (Alarm1) Bit 2 : Relay3 0=OFF 1=ON (Door2) Bit 3 : Relay4 0=OFF 1=ON (Alarm2) Bit 4-7 : 不使用 RAC-940 系列/RAC-960/970 系列 : Bit 0 : 開門 Relay (0=OFF 1=ON) Bit 1 : 警報 Relay (0=OFF 1=ON) Bit 2 : 響鈴 Relay (0=OFF 1=ON) Bit 3 : 門鈴 Relay (0=OFF 1=ON) Bit 4~7 : 保留
	cMask	Byte2 : 指定要動作的 Relay RAC-2000 系列,RAC-2200 系列 : Bit 0 : Relay1 0=不動作,1=動作 (Door1) Bit 1 : Relay2 0=不動作,1=動作 (Alarm1) Bit 2 : Relay3 0=不動作,1=動作 (Door2) Bit 3 : Relay4 0=不動作,1=動作 (Alarm2)

		Bit 4-7 : 不使用 RAC-940 系列/RAC-960/970 系列 : Bit 0 : 開門 Relay 0=不動作,1=動作 Bit 1 : 警報 Relay 0=不動作,1=動作 Bit 2 : 響鈴 Relay 0=不動作,1=動作 Bit 3 : 門鈴 Relay 0=不動作,1=動作 Bit 4~7 : 保留
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	VC Code : iReturn =hacRelayAction(1,0x0f,0x0f,ghComm,1000);	

4.3.5 hacAddCard(單筆新增卡號/非壓縮)

支援機型	RAC-2000 系列 (不包含 RAC-2000G) ,RAC-940 系列,RAC-960/970 系列, RAC-2200 系列	
API 名稱	hacAddCard	
語法	<pre>int __stdcall hacAddCard (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/非壓縮	
參數說明	iNodeID	裝置代號
	cCardNo	ASCII 碼的卡號
	cPassWord	ASCII 碼的密碼
	iCardLen	卡號的長度
	iPassLen	密碼的長度
	iTimeZone	時段
	cStatus	狀態代碼, 如是否檢查假日, 檢查時段, 黑名單....
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註		
程式片斷 範例說明	<pre>VC Code : int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay));</pre>	

```

unsigned char cReceiveBuff[280];
memset(cReceiveBuff,0,sizeof(cReceiveBuff));
int iNodeID=0x01;
iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
char cCardNumber[20];
memset(cCardNumber,0,sizeof(cCardNumber));
GetDlgItemText(IDC_EDITCard,cCardNumber,20);
int iLoop=0;
int iReadLen=0;
iReturn
=hacAddCard(iNodeID,cCardNumber,(int)strlen(cCardNumber),"
",0x00,0x00,0x00,ghComm,1000);
    if (iReturn==0)
    {
        strcpy(sDisplay,"(RAC-2000P Support Only)Add Card : OK : ");
        strcat(sDisplay,cCardNumber);
    }
    else
    {
        strcpy(sDisplay,"(RAC-2000P Support Only)Add Card : Error :
");
        sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
    }
    SetDlgItemText(IDC_EDITReturn,sDisplay);

```

4.3.6 hacDelCard(單筆刪除卡號/非壓縮)

支援機型	RAC-2000 系列 (不包含 RAC-2000G) ,RAC-940 系列,RAC-960/970 系列,RAC-2200 系列,RAC-852 系列, RAC-820PxP 系列	
API 名稱	hacDelCard	
語法	<pre>int __stdcall hacDelCard (int iNodeID, char *cCardNo, int iCardLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	單筆刪除卡號/非壓縮	
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	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); char cCardNumber[20]; memset(cCardNumber,0,sizeof(cCardNumber)); GetDlgItemText(IDC_EDITCard,cCardNumber,20); int iLoop=0; int iReadLen=0; iReturn =hacDelCard(iNodeID,cCardNumber,</pre>	

	<pre> (int)strlen(cCardNumber),ghComm,1000); if (iReturn==0) { strcpy(sDisplay,"(RAC-2000P Support Only)Delete Card : OK : "); strcat(sDisplay,cCardNumber); } else { strcpy(sDisplay,"(RAC-2000P Support Only)Delete Card : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	--

4.3.7 hacGetSensor (讀取 Sensor 狀態)

支援機型	RAC-2000 系列,RAC-940 系列,RAC-960/970 系列,RAC-2200 系列, RAC-852 系列	
API 名稱	hacGetSensor	
語法	<pre>int __stdcall hacGetSensor (int iNodeID, int * iSensor, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	讀取 Sensor 的狀態	
參數說明	iNodeID	裝置代號
	iSensor	傳回裝置的 Sensor 狀態
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合	<p>RAC-2000 系列</p> <p>-Bit 0 : 讀取 Push Button1 的現狀 , 1→Close 0→Open</p> <p>-Bit 1 : 讀取 Push Button2 的現狀 , 1→Close 0→Open</p> <p>-Bit 2 : 讀取 Door Sensor1 的現狀 , 1→Close 0→Open</p> <p>-Bit 3 : 讀取 Door Sensor2 的現狀 , 1→Close 0→Open</p> <p>-Bit 4 : 讀取 Case Sensor1 的現狀 , 1→Close 0→Open</p> <p>-Bit 5 : 讀取 Case Sensor2 的現狀 , 1→Close 0→Open</p> <p>-Bit 6 : 讀取 Case Sensor3 的現狀 , 1→Close 0→Open</p> <p>-Bit 7 : 讀取 Case Sensor4 的現狀 , 1→Close 0→Open</p> <p>RAC-2200 系列</p> <p>-Bit 0 : 讀取 IN1 的現狀 , 1→Close 0→Open (RAC-2200 消防連動)</p> <p>-Bit 1 : 讀取 IN2 的現狀 , 1→Close 0→Open (RAC-2200 Alarm 連動)</p> <p>-Bit 2~7 : 保留</p> <p>RA-C940 系列/RAC-960/970 系列</p> <p>Byte1 : Sensor 狀態</p> <p>-Bit 0 : Case Sensor 的現狀 , 1→Close 0→Open</p>	

	<p>-Bit 1 : Door Sensor 的現狀 , 1→Close 0→Open</p> <p>-Bit 2 : Push Button 的現狀 , 1→Close 0→Open</p> <p>-Bit 3~7 : 保留</p> <p>Byte2 : Relay 狀態</p> <p>-Bit 0 : 開門 Relay , 1→動作 0→未動作</p> <p>-Bit 1 : 警報 Relay , 1→動作 0→未動作</p> <p>-Bit 2 : 響鈴 Relay , 1→動作 0→未動作</p> <p>-Bit 3 : 門鈴 Relay , 1→動作 0→未動作</p> <p>-Bit 4~7 : 保留</p>
備註	<p>VC Code :</p> <pre> int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iSensor=0x00; int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); iReturn =hacGetSensor(iNodeID,&iSensor,ghComm,1000); if (iReturn==0) { strcpy(sDisplay,"Read Sensor : "); sprintf(sDisplay+strlen(sDisplay),"%02X ",(BYTE)iSensor); } else { strcpy(sDisplay,"Read Sensor : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>

4.3.8 hacGetEEData(讀取裝置 EEPROM 值)

支援機型	RAC-2000 系列, RAC-2200 系列	
API 名稱	hacGetEEData	
語法	<pre>int __stdcall hacGetEEData (int iNodeID, char *cEEData, int *iReceiveDataLen, unsigned int *iEEAddr, int *iEELen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	讀取裝置 EEPROM 值	
參數說明	iNodeID	裝置代號
	cEEData	傳回的 EEPROM 內容
	iReceiveDataLen	實際傳回 EEPROM 的長度
	iEEAddr	要傳回 EEPROM 的地址
	iEELen	要傳回 EEPROM 的長度，最長 240 個位元組
	hComm	handle 值，可以是 COM 或是 TCP/IP
	iTimeout	等待逾時，單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS，否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0; int iReadLen=0;</pre>	

	<pre> iReturn =hacGetEEData(iNodeID,cReceiveBuff, &iReadLen,0x0069,0x08,ghComm,1000); if (iReturn==0) { strcpy(sDisplay,"Read EEPROM : "); for(iLoop=0;iLoop<iReadLen;iLoop++) { sprintf(sDisplay+strlen(sDisplay),"%02X", (BYTE)cReceiveBuff[iLoop]); } } else { strcpy(sDisplay,"Read EEPROM : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	---

4.3.9 hacSetEEData (寫入 EEPROM 值)

支援機型	RAC-2000 系列, RAC-2200 系列	
API 名稱	hacSetEEData	
語法	<pre>int __stdcall hacSetEEData (int iNodeID, char *cEEData, unsigned int iEEAddr, int iEELen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	寫入資料到裝置 EEPROM 內	
參數說明	iNodeID	裝置代號
	cEEData	要寫入的 EEPROM 內容, 其內容為實際存放的值
	iEEAddr	要寫入 EEPROM 的位址
	iEELen	要寫入 EEPROM 的長度, 最長 240 個位元組
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); cReceiveBuff[0]=0x05; iReturn =hacSetEEData(iNodeID,(unsigned char*)cReceiveBuff,0x0069,0x01,ghComm,1000); if (iReturn==0)</pre>	

	<pre>{ strcpy(sDisplay,"Write EEPROM : OK!"); } else { strcpy(sDisplay,"Write EEPROM : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay);</pre>
--	---

4.3.10 hacAddZCard(單筆新增壓縮卡號)

支援機型	RAC-2000G	
API 名稱	hacAddZCard	
語法	<pre>int __stdcall hacAddZCard (int iNodeID, char *cCardNo, int iCardLen, char *cPassword, int iPassLen, char cStatus, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	單筆新增壓縮卡號，壓縮卡號必需皆為數位	
參數說明	iNodeID	裝置代號
	cCardNo	ASCII 碼的卡號
	iCardLen	cCardNo 的長度
	cPassword	ASCII 碼的密碼
	iPassLen	cPassword 長度，無 password 則設為 0
	cStatus	狀態代碼，如是否檢查假日，檢查時段，黑名單....
	hComm	handle 值，可以是 COM 或是 TCP/IP
	iTimeout	等待逾時，單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS，否則會有相對應的錯誤訊息出現.	
使用場合		
備註	VC Code : <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL);</pre>	


```

char cCardNumber[20];
memset(cCardNumber,0,sizeof(cCardNumber));
GetDlgItemText(IDC_EDITCard,cCardNumber,20);
int iLoop=0;
int iReadLen=0;

iReturn =hacAddZCard(iNodeID,cCardNumber,
(int)strlen(cCardNumber),"",0x00,0x00,ghComm,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"(RAC-2000G Support Only)Add Compress
Card : OK : ");
    strcat(sDisplay,cCardNumber);
}
else
{
    strcpy(sDisplay,"(RAC-2000G Support Only)Add Compress
Card : Error!");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

4.3.11 hacDelZCard(單筆刪除壓縮卡號)

支援機型	RAC-2000G	
API 名稱	hacDelZCard	
語法	<pre>int __stdcall hacDelZCard (int iNodeID, char *cCardNo, int iCardLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	單筆刪除壓縮卡號	
參數說明	iNodeID	裝置代號
	cCardNo	ASCII 碼的卡號
	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	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); char cCardNumber[20]; memset(cCardNumber,0,sizeof(cCardNumber)); GetDlgItemText(IDC_EDITCard,cCardNumber,20); int iLoop=0; int iReadLen=0; iReturn =hacDelZCard(iNodeID,cCardNumber, (int)strlen(cCardNumber),ghComm,1000);</pre>	

	<pre> if (iReturn==0) { strcpy(sDisplay,"(RAC-2000G Support Only)Delete Compress Card : OK : "); strcat(sDisplay,cCardNumber); } else { strcpy(sDisplay,"(RAC-2000G Support Only)Delete Compress Card : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	---

4.3.12 hacGetRAMData(讀取裝置中 RAM 的內資料)

支援機型	RAC-2000 系列,RAC-960/970 系列,RAC-2200 系列, RAC-820PxP 系列	
API 名稱	hacGetRAMData	
語法	<pre>int __stdcall hacGetRAMData (int iNodeID, char *cRAMData, int *iReceiveDataLen , unsigned int iRAMAddr, int iRAMLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	讀取裝置 RAM 中的資料	
參數說明	iNodeID	裝置代號
	cRAMData	傳回的 RAM 內容
	iReceiveDataLen	實際傳回 RAM 的長度
	iRAMAddr	要傳回 RAM 的地址
	iRAMLen	要傳回 RAM 的長度，最長 240 個位元組
	hComm	handle 值，可以是 COM 或是 TCP/IP
	iTimeout	等待逾時，單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS，否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0; int iReadLen=0;</pre>	

```

        iReturn = hacGetRAMData(iNodeID, cReceiveBuff,
&iReadLen, 0x00, 0x40, ghComm, 1000);
        if (iReturn == 0)
        {
            strcpy(sDisplay, "Read RAM : ");
            for(iLoop=0; iLoop<iReadLen; iLoop++)
            {
                sprintf(sDisplay+strlen(sDisplay), "%02X",
                (BYTE)cReceiveBuff[iLoop]);
            }
        }
        else
        {
            strcpy(sDisplay, "Read RAM : Error : ");
            sprintf(sDisplay+strlen(sDisplay), "%04X ", (WORD)iReturn);
        }
        SetDlgItemText(IDC_EDITReturn, sDisplay);

```

4.3.13 hacSetRAMData (寫入 RAM 值)

支援機型	RAC-2000 系列,RAC-960/970 系列,RAC-2200 系列, RAC-820PxP 系列	
API 名稱	hacSetRAMData	
語法	<pre>int __stdcall hacSetRAMData (int iNodeID, char *cRAMData, unsigned int iRAMAddr, int iRAMLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	寫入資料到裝置 RAM 內	
參數說明	iNodeID	裝置代號
	cRAMData	要寫入的 RAM 內容, 其內容為實際存放的值
	iRAMAddr	要寫入 RAM 的位址
	iRAMLen	要寫入 RAM 的長度, 最長 240 個位元組
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); cReceiveBuff[0]=0x01; cReceiveBuff[1]=0x01; iReturn =hacSetRAMData(iNodeID,(unsigned</pre>	

	<pre>char*)cReceiveBuff,0x000020,0x02,ghComm,1000); if (iReturn==0) { strcpy(sDisplay,"Write RAM : OK!"); } else { strcpy(sDisplay,"Write RAM : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay);</pre>
--	--

4.3.14 hacAddVisitorCard(單筆新增來賓卡)

支援機型	RAC-2000 系列,RAC-2200 系列, RAC-820PxP 系列	
API 名稱	hacAddVisitorCard	
語法	<pre>int __stdcall hacAddVisitorCard (int iNodeID, char *cCardNo, int iCardStart, int iCardLen, char *cStartDate, char *cStartTime, char * cEndDate, char * cEndTime, int iWeek, int iTimes, int iSerial,, int iAnti, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	單筆新增來賓卡	
參數說明	iNodeID	裝置代號
	cCardNo	ASCII 碼的卡號
	iCardStart	卡號的起始位
	iCardLen	卡號的長度
	cStartDate	卡片有效的開始日期格式為 <code>yyyymmdd</code>
	cStartTime	卡片有效的開始時間，格式為 <code>hhmmss</code>
	cEndDate	卡片有效的開始日期格式為 <code>yyyymmdd</code>
	cEndTime	卡片有效的開始時間，格式為 <code>hhmmss</code>
	iWeek	從高位到低位元元，分別表示週一到周日，最低位請填 0
	iTimes	刷卡有效次數
	iSerial	有 0 到 49 筆來賓卡，增加到哪個位置
	iAnti	Anti 狀態代碼，請參考附件
	hComm	handle 值，可以是 COM 或是 TCP/IP
	iTimeout	等待逾時，單位為 ms

回傳值	當此功能成功會返回 HF_RET_SUCESS , 否則會有相對應的錯誤訊息出現.
使用場合	
備註	<p>VC Code :</p> <pre> int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); char cCardNumber[20]; memset(cCardNumber,0,sizeof(cCardNumber)); GetDlgItemText(IDC_EDITCard,cCardNumber,20); int iLoop=0; int iReadLen=0; iReturn =hacAddVisitorCard(iNodeID,cCardNumber, 0x01,(int)strlen(cCardNumber), "20010101","000000","20151231","235959", 0xff,2000,0x00,0x01,ghComm,1000); if (iReturn==0) { strcpy(sDisplay,"Add Visitor Card : OK : "); strcat(sDisplay,cCardNumber); } else { strcpy(sDisplay,"Add Visitor Card : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>

4.3.15 hacDelVisitorCard(單筆刪除來賓卡)

支援機型	RAC-2000 系列,RAC-2200 系列, RAC-820PxP 系列	
API 名稱	hacDelVisitorCard	
語法	<pre>int __stdcall hacDelVisitorCard (int iNodeID, int iSerial,, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	單筆刪除卡號/非壓縮	
參數說明	iNodeID	裝置代號
	iSerial	有 0 到 49 筆來賓卡，刪除哪個位置的卡
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>RAC-2000 系列來賓卡為 50 組, 其相關資訊請參考附件 1 下的 1.8 ,1.12 及 1.13</p> <p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); char cCardNumber[20]; memset(cCardNumber,0,sizeof(cCardNumber)); GetDlgItemText(IDC_EDITCard,cCardNumber,20); int iLoop=0; int iReadLen=0; iReturn =hacDelVisitorCard(iNodeID,0x01,ghComm,1000);</pre>	

	<pre> if (iReturn==0) { strcpy(sDisplay,"Delete Visitor Card : OK : "); strcat(sDisplay,cCardNumber); } else { strcpy(sDisplay,"Delete Visitor Card : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	---

4.3.16 hacGetFlashData(讀取裝置 Flash 值)

支援機型	RAC-940 系列,RAC-960/970 系列,RAC-852 系列	
API 名稱	hacGetFlashData	
語法	<pre>int __stdcall hacGetFlashData (int iNodeID, char *cFlashData, int *iReceiveDataLen, unsigned int *iFlashAddr, int *iFlashLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	讀取裝置 Flash 值	
參數說明	iNodeID	裝置代號
	cFlashData	傳回的 Flash 內容
	iReceiveDataLen	實際傳回 Flash 的長度
	iFlashAddr	要傳回 Flash 的地址
	iFlashLen	要傳回 Flash 的長度，最長 240 個位元組
	hComm	handle 值，可以是 COM 或是 TCP/IP
	iTimeout	等待逾時，單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS，否則會有相對應的錯誤訊息出現。	
使用場合		
備註	<p>功能說明：</p> <p>利用 hacGetFlashData 功能取得裝置中某一段 Flash 中的資料。</p> <p>VC Code：</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL);</pre>	

```

        int iLoop=0;
        int iReadLen=0;
        iReturn
=hacGetFlashData(iNodeID,cReceiveBuff,&iReadLen,0x0069,0x08,
ghComm,1000);
        if (iReturn==0)
        {
            strcpy(sDisplay,"Read Flash : ");
            for(iLoop=0;iLoop<iReadLen;iLoop++)
            {
                sprintf(sDisplay+strlen(sDisplay),"%02X
",(BYTE)cReceiveBuff[iLoop]);
            }
        }
        else
        {
            strcpy(sDisplay,"Read Flash : Error : ");
            sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
        }
        SetDlgItemText(IDC_EDITReturn,sDisplay);

```

4.3.17 hacSetFlashData (寫入 Flash 值)

支援機型	RAC-940 系列,RAC-960/970 系列,RAC-852 系列	
API 名稱	hacSetFlashData	
語法	<pre>int __stdcall hacSetFlashData (int iNodeID, char *cFlashData, unsigned int iFlashAddr, int iFlashLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	寫入資料到裝置 Flash 內	
參數說明	iNodeID	裝置代號
	cFlashData	要寫入的 Flash 內容, 其內容為實際存放的值
	iFlashAddr	要寫入 Flash 的位址
	iFlashLen	要寫入 Flash 的長度, 最長 240 個位元組
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>功能說明：</p> <p>利用 hacSetFlashData 功能寫入資料到裝置 Flash 中.</p> <p>VC Code：</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); cReceiveBuff[0]=0x05; iReturn =hacSetFlashData(iNodeID,(unsigned</pre>	

	<pre> char*)cReceiveBuff,0x0069,0x01,ghComm,1000); if (iReturn==0) { strcpy(sDisplay,"Write Flash : OK!"); } else { strcpy(sDisplay,"Write Flash : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	--

4.3.18 hacGetParaData(讀取裝置一般 Para 值)

支援機型	RAC-940 系列,RAC-960/970 系列,RAC-852 系列, RAC-820Px F 系列	
API 名稱	hacGetParaData	
語法	<pre>int __stdcall hacGetParaData (int iNodeID, char *cFlashData, int *iReceiveDataLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	讀取裝置 Flash 值	
參數說明	iNodeID	裝置代號
	cFlashData	傳回的 Flash 內容
	iReceiveDataLen	實際傳回 Flash 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	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>功能說明：</p> <p>利用 hacGetParaData 功能取得裝置中某一段 Para 中的資料。</p> <p>VC Code：</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0; int iReadLen=0; iReturn =hacGetFlashData(iNodeID,cReceiveBuff,&iReadLen,0x0069,0x08,</pre>	


```

ghComm,1000);
    if (iReturn==0)
    {
        strcpy(sDisplay,"Read Flash : ");
        for(iLoop=0;iLoop<iReadLen;iLoop++)
        {
            sprintf(sDisplay+strlen(sDisplay),"%02X
", (BYTE)cReceiveBuff[iLoop]);
        }
    }
    else
    {
        strcpy(sDisplay,"Read Flash : Error : ");
        sprintf(sDisplay+strlen(sDisplay),"%04X ", (WORD)iReturn);
    }
    SetDlgItemText(IDC_EDITReturn,sDisplay);

```

4.3.19 hacSetParaData (寫入一般 Para 值)

支援機型	RAC-940 系列,RAC-960/970 系列,RAC-852 系列, RAC-820Px F 系列	
API 名稱	hacSetParaData	
語法	<pre>int __stdcall hacSetParaData (int iNodeID, char *cFlashData, int iFlashLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	寫入資料到裝置 Flash 內	
參數說明	iNodeID	裝置代號
	cFlashData	要寫入的 Flash 內容, 其內容為實際存放的值
	iFlashLen	要寫入 Flash 的長度, 最長 240 個位元組
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>功能說明：</p> <p>利用 hacSetParaData 功能寫入資料到裝置 Para 中.</p> <p>VC Code：</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); cReceiveBuff[0]=0x05; iReturn =hacSetFlashData(iNodeID,(unsigned char*)cReceiveBuff,0x0069,0x01,ghComm,1000); if (iReturn==0)</pre>	

	<pre>{ strcpy(sDisplay,"Write Flash : OK!"); } else { strcpy(sDisplay,"Write Flash : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay);</pre>
--	---

4.3.20 hacGetSysParaData(讀取裝置系統參數值)

支援機型	RAC-940 系列,RAC-960/970 系列,RAC-852 系列, RAC-820PxP 系列	
API 名稱	hacGetSysParaData	
語法	<pre>int __stdcall hacGetSysParaData (int iNodeID, char *cFlashData, int *iReceiveDataLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	讀取裝置系統參數設定值	
參數說明	iNodeID	裝置代號
	cFlashData	傳回的系統參數內容
	iReceiveDataLen	實際傳回系統參數的長度
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>功能說明：</p> <p>利用 hacGetSysParaData 功能取得裝置中某一段系統參數中的資料。</p> <p>VC Code：</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0; int iReadLen=0; iReturn =hacGetFlashData(iNodeID,cReceiveBuff,&iReadLen,0x0069,0x08,</pre>	

	<pre> ghComm,1000); if (iReturn==0) { strcpy(sDisplay,"Read Flash : "); for(iLoop=0;iLoop<iReadLen;iLoop++) { sprintf(sDisplay+strlen(sDisplay),"%02X", (BYTE)cReceiveBuff[iLoop]); } } else { strcpy(sDisplay,"Read Flash : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	---

4.3.21 hacSetSysParaData (寫入系統參數值)

支援機型	RAC-940 系列,RAC-960/970 系列,RAC-852 系列, RAC-820Px F 系列	
API 名稱	hacSetSysParaData	
語法	<pre>int __stdcall hacSetSysParaData (int iNodeID, char *cFlashData, int iFlashLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	寫入系統參數到裝置內	
參數說明	iNodeID	裝置代號
	cFlashData	要寫入的系統參數內容，其內容為實際存放的值
	iFlashLen	要寫入系統參數的長度，最長 240 個位元組
	hComm	handle 值，可以是 COM 或是 TCP/IP
	iTimeout	等待逾時，單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS，否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>功能說明：</p> <p>利用 hacSetSysParaData 功能寫入系統參數到裝置中.</p> <p>VC Code：</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); cReceiveBuff[0]=0x05; iReturn =hacSetFlashData(iNodeID,(unsigned char*)cReceiveBuff,0x0069,0x01,ghComm,1000); if (iReturn==0)</pre>	

	<pre>{ strcpy(sDisplay,"Write Flash : OK!"); } else { strcpy(sDisplay,"Write Flash : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay);</pre>
--	---

4.3.22 hacGetMifare(讀取 Mifare 模組讀卡參數值)

支援機型	RAC-940 系列,RAC-960/970 系列,RAC-852 系列, RAC-820PxP 系列	
API 名稱	hacGetMifare	
語法	<pre>int __stdcall hacGetMifare (int iNodeID, int iKeyType, int *iBlock, int *iStartDigit, int *iDigitLength, int *iCompact, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	讀取裝置 Mifare 模組讀卡參數值	
參數說明	iNodeID	裝置代號
	iKeyType	0 表示讀 CSN(序號) , 1 讀 KeyA , 2 讀 KeyB
	iBlock	當 iKeyType>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
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>功能說明：</p> <p>利用 hacGetMifare 功能取得 Mifare 模組讀卡參數值.</p> <p>VC Code：</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280];</pre>	


```

memset(cReceiveBuff,0,sizeof(cReceiveBuff));
int iNodeID=0x01;
iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
int iLoop=0;
int iReadLen=0;
iReturn
=hacGetFlashData(iNodeID,cReceiveBuff,&iReadLen,0x0069,0x08,
ghComm,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"Read Flash : ");
    for(iLoop=0;iLoop<iReadLen;iLoop++)
    {
        sprintf(sDisplay+strlen(sDisplay),"%02X
",(BYTE)cReceiveBuff[iLoop]);
    }
}
else
{
    strcpy(sDisplay,"Read Flash : Error : ");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

4.3.23 hacSetMifare (寫入 Mifare 模組讀卡參數值)

支援機型	RAC-940 系列,RAC-960/970 系列,RAC-852 系列, RAC-820PxP 系列	
API 名稱	hacSetMifare	
語法	<pre>int __stdcall hacSetMifare (int iNodeID, int iKeyType, int iBlock, int iStartDigit, int iDigitLength, int iCompact, unsigned char *cKeyValue, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	寫入 Mifare 模組讀卡參數值到裝置內	
參數說明	iNodeID	裝置代號
	iKeyType	0 表示讀 CSN(序號) , 1 讀 KeyA , 2 讀 KeyB
	iBlock	當 iKeyType>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
	cKeyValue	Mifare Key , 6 個位元組
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<p>功能說明：</p> <p>利用 hacSetMifare 功能寫入 Mifare 模組讀卡參數值.</p> <p>VC Code：</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256];</pre>	

	<pre> memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); cReceiveBuff[0]=0x05; iReturn =hacSetFlashData(iNodeID,(unsigned char*)cReceiveBuff,0x0069,0x01,ghComm,1000); if (iReturn==0) { strcpy(sDisplay,"Write Flash : OK!"); } else { strcpy(sDisplay,"Write Flash : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	--

4.3.24 hacAddCardEX(單筆新增卡號附加姓名顯示)

支援機型	RAC-940 系列,RAC-960/970 系列,RAC-852 系列	
API 名稱	hacAddCardEX	
語法	<pre>int __stdcall hacAddCardEX(int iNodeID, char *cCardNo, int iCardLen, char *cPassWord, int iPassLen, char *cName, int iNameLen, int iTimeZone, char cStatus, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	單筆新增卡號及姓名(要確認系統參數設定為卡號加姓名,否則裝置會傳回錯誤代碼)	
參數說明	iNodeID	裝置代號
	cCardNo	ASCII 碼的卡號
	iCardLen	卡號的長度
	cPassWord	ASCII 碼的密碼
	iPassLen	密碼的長度
	cName	ASCII 碼的姓名，只有 RAC940/960/970 才有姓名
	iNameLen	姓名的長度
	iTimeZone	時段
	cStatus	狀態代碼，如是否檢查假日，檢查時段，黑名單....
	hComm	handle 值，可以是 COM 或是 TCP/IP
	iTimeout	等待逾時，單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS，否則會有相對應的錯誤訊息出現.	
使用場合		
備註	功能說明：	

利用 hacAddCardEX 功能新增一筆卡號許可權。

VC Code :

```
int iReturn=0;
int iReturnCode=0;
char sDisplay[256];
memset(sDisplay,0,sizeof(sDisplay));
unsigned char cReceiveBuff[280];
memset(cReceiveBuff,0,sizeof(cReceiveBuff));
int iNodeID=0x01;
iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
char cCardNumber[20];
memset(cCardNumber,0,sizeof(cCardNumber));
GetDlgItemText(IDC_EDITCard,cCardNumber,20);
char cName[20];
memset(cName,0,sizeof(cName));
GetDlgItemText(IDC_EDITCard4,cName,20);
int iLoop=0;
int iReadLen=0;

iReturn
=hacAddCardEX(iNodeID,cCardNumber,(int)strlen(cCardNumber),
"",0x00,cName,(int)strlen(cName),0x00,0x00,ghComm,1000);

if (iReturn==0)
{
    strcpy(sDisplay,"(RAC940/960/970 Support Only)Add
CardEX:OK:");
    strcat(sDisplay,cCardNumber);
}
else
{
    strcpy(sDisplay,"(RAC940/960/970 Support Only)Add
CardEX:Error:");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
```

	SetDlgItemText(IDC_EDITReturn,sDisplay);
--	--

4.3.25 haclInitial(初始化設備)

支援機型	RAC-940 系列,RAC-960/970 系列,RAC-852 系列, RAC-820PxP 系列	
API 名稱	haclInitial	
語法	<pre>int __stdcall haclInitial (int iNodeID, int iDeviceType, int iClearFlag, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	將機器到恢復到原始出廠狀態	
參數說明	iNodeID	裝置代號
	iDeviceType	此參數值為 3 表示 RAC960/970,4 表示 RAC940
	iClearFlag	表示要初始化設備的哪些資料 Bit 0 : 1 刪除所有卡號 Bit 1 : 1 初始化所有表格 Bit 2 : 1 刪除刷卡及事件記錄 Bit 3 : 1 一般參數初始化 Bit 4 : 1 只有清除黑名單 Bit 5~7 未使用
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	功能說明： 利用 haclInitial 功能初始化設備。 VC Code： <pre>iReturn =haclInitial(1,0x03,0xff,ghComm,1000);</pre>	

4.3.26 hacFingerPrinterQueryMasterFP (查詢 Master 指紋)

支援機型	RAC960/970 系列,RAC-852 系列	
API 名稱	hacFingerPrinterQueryMasterFP	
語法	int __stdcall hacFingerPrinterQueryMasterFP(int iNodeID,HANDLE hComm, unsigned char *cFingerPrinterData1,unsigned char *cFingerPrinterData2,int *iReturnCode,unsigned int iTimeout)	
目的	查詢 Master 指紋	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iNodeID	裝置代號
	CardLen	卡號長度
	cCardNo	卡號
	cFingerPrinterData1	傳回第一枚指紋的資料內容。
	cFingerPrinterData2	傳回第二枚指紋的資料內容。
	iReturnCode	主機回傳的錯誤碼,當有錯誤發生時, 錯誤代碼請參照附件 1 說明。
	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註		

4.3.27 hacFingerPrinterUpdateMasterFP (更新 Master 指紋)

支援機型	RAC960/970 系列,RAC-852 系列	
API 名稱	hacFingerPrinterUpdateMasterFP	
語法	int __stdcall hacFingerPrinterUpdateMasterFP(int iNodeID,HANDLE hComm, unsigned char *cFingerPrinterData1,unsigned char *cFingerPrinterData2,int *iReturnCode,unsigned int iTimeout)	
目的	更新卡鐘 Master 指紋	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iNodeID	裝置代號
	cFingerPrinterData1	第一枚指紋的資料內容。
	cFingerPrinterData2	第二枚指紋的資料內容。
	iReturnCode	主機回傳的錯誤碼,當有錯誤發生時, 錯誤代碼請參照附件 1 說明。
	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註		

4.3.28 hacFingerPrinterDeleteFP (刪除 Master 指紋)

支援機型	RAC960/970 系列,RAC-852 系列	
API 名稱	hacFingerPrinterDeleteFP	
語法	int __stdcall hacFingerPrinterDeleteFP(int iNodeID,HANDLE hComm, int *iReturnCode,unsigned int iTimeout)	
目的	刪除主機 Master 指紋	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iNodeID	裝置代號
	iReturnCode	主機回傳的錯誤碼,當有錯誤發生時, 錯誤代碼請參照附件 1 說明。
	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註		

4.3.29 hacAddCardFingerPrintEx (上傳指紋特徵值)

支援機型	RAC960/970 系列,RAC-852 系列, RAC-820Px F 系列	
API 名稱	hacAddCardFingerPrintEx	
語法	Int __stdcall hacAddCardFingerPrintEx (int iNodeID,char *cCardNo,int iCardLen,char *cPassword,int iPassLen,int iTimeZone,char cStatus,unsigned char *cFingerPrinterData1,unsigned char *cFingerPrinterData2,HANDLE hComm, unsigned int iTimeout)	
目的	上傳指紋特徵值到機器上	
參數說明	iNodeID	裝置代號
	cCardNo	ASCII 碼的卡號
	iCardLen	卡號的長度
	cPassWord	ASCII 碼的密碼
	iPassLen	密碼的長度
	iTimeZone	時段
	cStatus	狀態代碼. 如是否檢查假日, 檢查時段, 黑名單...
	cFingerPrinterData1	第 1 枚特徵
	cFingerPrinterData2	第 2 枚特徵
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註		

4.3.30 hacFingerPrintQueryUser (收回指紋特徵值)

支援機型	RAC960/970 系列,RAC-852 系列, RAC-820Px F 系列	
API 名稱	hacFingerPrintQueryUser	
語法	Int __stdcall hacFingerPrintQueryUser (int iNodeID, HANDLE hComm, int iCardLen, char *cCardNo ,unsigned char *cFingerPrinterData1,unsigned char *cFingerPrinterData2,int *iCardFormatLen, int *iReturnCode, unsigned int iTimeout)	
目的	收回指紋特徵值	
參數說明	iNodeID	裝置代號
	iCardLen	卡號長度

	cCardNo	ASCII 碼的卡號
	cFingerPrinterData1	傳回第一枚指紋
	cFingerPrinterData2	傳回第二枚指紋
	iCardFormatLen	傳回指紋長度
	iReturnCode	傳回碼
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註		

4.4 參數表相關 API(Level 3)

4.4.1 hacSetLanMode (設定主動發送模式)

支援機型	RAC-2000PN/PSN	
API 名稱	hacSetLanMode	
語法	<pre>int __stdcall hacSetLanMode (int iNodeID, char cMode, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	設定 RAC-2000PN 主動發送模式	
參數說明	iNodeID	裝置代號
	cMode	0 表示取消主動發送模式，1 表示啟用主動發送模式
	hComm	handle 值，可以是 COM 或是 TCP/IP
	iTimeout	等待逾時，單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<pre>VC Code : int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); cReceiveBuff[0x0]=0x01; iReturn=hacSetLanMode(iNodeID,1,ghComm,1000); if (iReturn==0) { strcpy(sDisplay,"Enabled Active Send : OK!"); } else</pre>	

	<pre>{ strcpy(sDisplay,"Enabled Active Send : Error : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay);</pre>
--	--

4.4.2 hacSetReader (設定 RAC-2200 下 Reader 的參數)

支援機型	RAC-2200 系列	
API 名稱	hacSetReader	
語法	<pre>int __stdcall hacSetReader (int iNodeID, int iReaderID, int iIndex, unsigned char *cSendData, int iSendDataLen, HANDLE hComm, unsigned int iTimeout)</pre>	
目的	設定 RAC-2200 下 Reader 的參數	
參數說明	iNodeID	裝置代號
	iReaderID	RAC2200 下的 Reader ID,0~3,16~19
	iIndex	Reader 的 Read/Write Index
	cSendData	要傳送到 Reader 的參數
	iSendDataLen	要發送的數據長度
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iTimeout	等待逾時, 單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合	此功能只有 RAC2200 控制器才有提供	
備註	<p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); //控制讀卡器發出叫聲 cReceiveBuff[0x0]=0x00;</pre>	

```

cReceiveBuff[0x1]=0x00;
cReceiveBuff[0x2]=0x0D;
cReceiveBuff[0x3]=0x0F;
cReceiveBuff[0x4]=0x01;
cReceiveBuff[0x5]=0x02;
cReceiveBuff[0x6]=0x00;
iReturn=hacSetReader(iNodeID,1,23,cReceiveBuff,7,
ghComm,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"Set Reader:OK!");
}
else
{
    strcpy(sDisplay,"Set Reader:Error:");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

第五章 HDE-100 系列相關函數說明

5.1. 通訊相關 API

5.1.1 hsOpenChannel (開啟通訊聯機)

支援機型	HDE-100 系列	
API 名稱	hsOpenChannel	
語法	<pre>int __stdcall hsOpenChannel(HANDLE *hComm, char *sComm, unsigned int iPort);</pre>	
目的	開啟通訊連接, 可以 TCP/IP 或 COM Port	
參數說明	hComm	開啟成功後傳回的 handle 值
	sComm	COM1-COM128 或者是 IP 地址
	iPort	傳輸速度(1200/2400/4800/9600/19200/38400) 或者是 TCP/IP Port 號
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<p>開啟 TCP/IP 時 :</p> <p>sComm : IP 地址, 如 172.16.1.1</p> <p>iPort : 埠號, 如 4660</p> <p>hComm 開啟成功後,傳回 socket 值</p> <p>ps. TCP/IP hComm 值為實際值加上 1000, 以防它與 COM Port 傳回的 handle 值相同</p> <p>VC Code :</p> <pre>int iReturn=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); char clP[20]; memset(clP,0,sizeof(clP)); int iPort=0;</pre>	

	<pre> GetDlgItemText(IDC_EDITIP,cIP,20); iPort=GetDlgItemInt(IDC_EDITPort,NULL); iReturn =hsELOpenChannel(&ghComm,cIP,iPort); if (iReturn!=0) { strcpy(sDisplay,"Open Channel : Error Number : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } else { strcpy(sDisplay,"Open Channel : OK!,HANDLE : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)ghComm); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	---

5.1.2 hsCloseChannel(關閉與 HDE-100 系列聯機的通訊)

支援機型	HDE-100 系列	
API 名稱	hsCloseChannel	
語法	int __stdcall hsCloseChannel(HANDLE hComm);	
目的	關閉與 HDE-100 系列的通訊聯機	
參數說明	hComm	使用 hsOpenChannel Function 時所傳回成功的 handle 值.
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機	如果 hCOMM 值 < 1000 表示是以 232 方式通訊.	
備註	<p>VC Code :</p> <pre> int iReturn=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); iReturn =hsELCloseChannel(ghComm); if (iReturn!=0) { strcpy(sDisplay,"Close Channel : Error Number : "); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } else { strcpy(sDisplay,"Close Channel : OK!"); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>	

5.1.3 hsELWriteTable(傳送表格到 HDE-100 系列)

支援機型	HDE-100 系列	
API 名稱	hsELWriteTable	
語法	<pre>int __stdcall hsELWriteTable(HANDLE hComm, int iELID, unsigned char *cTableData, int iTableLen, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	傳送單張表格到 HDE-100 系列 主機, 其命令請參考備註說明.	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cTableData	傳送表格資料的內容
	iTableLen	傳送表格資料的長度。
	iReturnCode	主機回傳的錯誤碼, 當有錯誤發生時, 錯誤代碼請參照附件 2.2.1 說明。
	iTimeout	等待主機回應的時間. 單位為毫秒, 建議這個值在 10 秒以上。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	表格	說明
	假日表	(地址 0-600)
	星期表	(地址 601-2378)
	時區表	(地址 2379-6474)
	最大設定值為 240 Bytes 的資料 VC Code : <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280];</pre>	

```

//unsigned char cReceiveBuff[301];
int iNodeID=0x01;
iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
int iLoop=0;
int iWrittenLen=240;
memset(cReceiveBuff,0xff,sizeof(cReceiveBuff));

cReceiveBuff[0]=0x00;
cReceiveBuff[1]=0x00;
cReceiveBuff[2]=0xF0;
cReceiveBuff[3]=0x00;

cReceiveBuff[4]=0X07 ;
cReceiveBuff[5]=0X31 ;
cReceiveBuff[6]=0X31 ;
cReceiveBuff[7]=0X31 ;
cReceiveBuff[8]=0X32 ;
cReceiveBuff[9]=0X40 ;
cReceiveBuff[10]=0X31 ;

iReturn=hsELWriteTable(ghComm,iNodeID,cReceiveBuff,iWritten
Len,&iReturnCode,5000);

if (iReturn==0)
{
    strcpy(sDisplay,"Write Table : OK!");
}
else
{
    strcpy(sDisplay,"Write Table : Error : ");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

5.1.4 hsELReadTable(讀取 HDE-100 表格)

支援機型	HDE-100 系列	
API 名稱	hsELReadTable	
語法	<pre>int __stdcall hsELReadTable(HANDLE hComm, int iELID, unsigned char *cTableData, int *iTableLen, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	從 HDE-100 系列主機讀取單一張表格，其表格請參考備註說明。	
參數說明	hComm	handle 值，可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cTableData	傳回讀取表格資料的內容
	iTableLen	傳回讀取表格資料的長度。
	iReturnCode	主機回傳的錯誤碼，當有錯誤發生時，錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間。單位為毫秒，建議這個值在 10 秒以上。
回傳值	當此功能成功會返回 HF_RET_SUCESS，否則會有相對應的錯誤訊息出現。	
使用時機		
備註	表格	說明
	假日表	(地址 0-600)
	星期表	(地址 601-2378)
	時區表	(地址 2379-6474)
	<p>每次讀取長度最大值為 240 Bytes 資料</p> <p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280];</pre>	

```

memset(cReceiveBuff,0,sizeof(cReceiveBuff));
int iNodeID=0x01;
iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
int iLoop=0;
int iReadLen=0;
int iTTotal=0;
iTotal=2 ;
memset(cReceiveBuff,0x00,sizeof(cReceiveBuff));

cReceiveBuff[0]=0x59 ;
cReceiveBuff[1]=0x02 ;

cReceiveBuff[2]=0x60 ;
cReceiveBuff[3]=0x02 ;
iReturn=hsELMatrixReadTable(ghComm,iNodeID,iTotal,cReceive
Buff,&iReadLen,&iReturnCode,1000);

if (iReturn==0)
{
    strcpy(sDisplay,"Matrix Read Table : ");
    for(iLoop=0;iLoop<iReadLen;iLoop++)
    {
        sprintf(sDisplay+strlen(sDisplay),"%02X
",(BYTE)cReceiveBuff[iLoop]);
    }
}
else
{
    strcpy(sDisplay,"Matrix Read Table : Error : ");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

5.1.5 hsELPolling (讀取 HDE-100 系列主機上的刷卡及事件記錄)

支援機型	HDE-100 系列	
API 名稱	hsELPolling	
語法	<pre>int __stdcall hsELPolling(HANDLE hComm, int iELID, int iPrevRecord, stPollList *stRecord, int *iRecord, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	讀取 HDE-100 系列主機上的刷卡及事件記錄	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	iPrevRecord	傳入上次 hsPollingData 傳回的事件筆數 (IReturnEvents)
	stRecord	傳送資料的結構 (請參考備註)
	iRecord	傳送回幾筆資料的長度。
	iReturnCode	主機回傳的錯誤碼, 當有錯誤發生時, 錯誤代碼請參照附件 2.2.1 說明。
	iTimeout	等待主機回應的時間. 單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<p>stRecord 結構：</p> <pre>char cEventCode[10]; char cDateTime[20]; char cCard[20]; char cDeviceID[10]; char cStatusCode[10];</pre> <p>VC Code：</p> <pre>int wmlId, wmEvent; switch (message) {</pre>	

```

case WM_INITDIALOG :
    hWndListView = GetDlgItem(hDlg, IDC_LIST1);
    SetFocus(hWndListView);
    InitListViewColumns(hWndListView);
    // FALSE since we set focus to hWndTreeView
    return FALSE;
    break;
case WM_DESTROY :
    break;
case WM_NOTIFY :
    // Process notification messages.
    switch (((LPNMHDR) lParam)->code)
    {

        // Fill subitems.
        case LVN_GETDISPINFO :
            //OnGetDispInfo((NMLVDISPINFO *) lParam);
            break;

        // Change labels.
        case LVN_ENDLABELEDIT :
            break;
    }
    return 0;
case WM_COMMAND :
    wParam = LOWORD(wParam);
    lParam = HIWORD(wParam);
    switch(wParam)
    {
        case IDB_PollingOnce :
            PollingLoop(hDlg);
            break;
        case IDB_ClearList :
            ListView_DeleteAllItems(hWndListView);
            break;
        case IDB_Close :

```


	<pre> EndDialog(hDlg,wmlId); return TRUE; break; case 2 : EndDialog(hDlg,wmlId); return TRUE; } break; } return FALSE;</pre>
--	---

5.2. Read/Write 相關 API

5.2.1 hsELReadParameter(讀取系統參數)

支援機型	HDE-100 系列	
API 名稱	hsELReadParameter	
語法	<pre>int __stdcall hsELReadParamenter(HANDLE hComm, int iELID, unsigned char *cParaData, int *iParaLen, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	讀取系統參數	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cParaData	傳送及接收資料的 buffer。
	iTableLen	傳回接收資料的長度
	iReturnCode	主機回傳的錯誤碼, 當有錯誤發生時, 錯誤代碼請參照附件 2.2.1 說明。
	iTimeout	等待主機回應的時間. 單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<pre>VC Code : int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0; int iReadLen=4; cReceiveBuff[0x0]=0x00;</pre>	

```

cReceiveBuff[0x1]=0x00;
cReceiveBuff[0x2]=0x13;
cReceiveBuff[0x3]=0x00;

iReturn=hsELReadParameter(ghComm,iNodeID,
cReceiveBuff,&iReadLen,&iReturnCode,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"Read Para : ");
    for(iLoop=0;iLoop<iReadLen;iLoop++)
    {
        sprintf(sDisplay+strlen(sDisplay),"%02X
",(BYTE)cReceiveBuff[iLoop]);
    }
}
else
{
    strcpy(sDisplay,"Read Para : Error : ");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

5.2.2 hsELWriteParameter (設定系統參數)

支援機型	HDE-100 系列	
API 名稱	hsELWriteParameter	
語法	<pre>int __stdcall hsELWriteParameter(HANDLE hComm, int iELID, unsigned char *cParaData, int iParaLen, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	設定系統參數	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cParaData	傳送資料的內容
	iParaLen	傳送資料的長度。
	iReturnCode	主機回傳的錯誤碼, 當有錯誤發生時, 錯誤代碼請參照附件 2.2.1 說明。
	iTimeout	等待主機回應的時間. 單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<pre>VC Code : int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0; int iReadLen=0; int iWrittenLen=23; cReceiveBuff[0x0]=0x00; cReceiveBuff[0x1]=0x00;</pre>	

```

cReceiveBuff[0x2]=0x13;
cReceiveBuff[0x3]=0x00;

cReceiveBuff[0x4]=0x00;
cReceiveBuff[0x5]=0x00;
cReceiveBuff[0x6]=0xE8;
cReceiveBuff[0x7]=0x03;

cReceiveBuff[0x8]=0x00;
cReceiveBuff[0x9]=0x30;
cReceiveBuff[0xa]=0x30;
cReceiveBuff[0xb]=0x30;
cReceiveBuff[0xc]=0x30;
cReceiveBuff[0xd]=0x00;
cReceiveBuff[0xe]=0x00;
cReceiveBuff[0xf]=0x00;
cReceiveBuff[0x10]=0x00;
cReceiveBuff[0x11]=0x00;
cReceiveBuff[0x12]=0x00;
cReceiveBuff[0x13]=0x00;
cReceiveBuff[0x14]=0x00;

cReceiveBuff[0x15]=0x00;
cReceiveBuff[0x16]=0x00;
iReturn=hsELWriteParameter(ghComm,iNodeID,
cReceiveBuff,iWrittenLen,&iReturnCode,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"Write Para : OK!");
}
else
{
    strcpy(sDisplay,"Write Para : Error : ");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}

```

	SetDlgItemText(IDC_EDITReturn,sDisplay);
--	--

5.2.3 hsELInitialize (系統初始化)

支援機型	HDE-100 系列	
API 名稱	hsELInitialize	
語法	<pre>int __stdcall hsELInitialize(HANDLE hComm, int iELID, char cInitFlag, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	系統初始化	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cInitFlag	傳送資料的內容(參考備註)
	iReturnCode	主機回傳的錯誤碼,當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<p>bit0 : 1 刪除所有卡號 bit1 : 1 初始化所有表格 bit2 : 1 刪除刷卡及事件記錄 bit3 : 1 參數初始化 bit4 : 1 只有清除黑名單 bit5~7 未使用</p> <p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff));</pre>	

```

int iNodeID=0x01;
iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
int iLoop=0;
int iReadLen=0;
int iWrittenLen=12;
iReturn=hsELInitialize(ghComm,iNodeID,(char)31,&
iReturnCode,10000);
if (iReturn==0)
{
    strcpy(sDisplay,"Initial RAC2000EL : OK!");
}
else
{
    strcpy(sDisplay,"Initial RAC2000EL : Error!");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

5.2.4 hsELGetInfo (讀取 HDE-100 系列主機版本及其他資訊)

支援機型	HDE-100 系列	
API 名稱	hsELGetInfo	
語法	<pre>int __stdcall hsELGetInfo(HANDLE hComm, int iELID, unsigned char *cInfoData, int *iInfoLen, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	讀取主機版本及其他資訊	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cInfoData	接收資料的 buffer(備註)。
	iInfoLen	傳回接收資料的長度
	iReturnCode	主機回傳的錯誤碼,當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<p>共 16 Byte</p> <p>Byte1 : 版本別</p> <p>Byte2 : 主版本</p> <p>Byte3 : 次版本</p> <p>Byte4 : Bate 版</p> <p>Byte5 : 年</p> <p>Byte6 : 月</p> <p>Byte7 : 日</p> <p>Byte8~11 : 已存放合法卡筆數</p> <p>Byte12~15 : 已存放刷卡筆數</p> <p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0;</pre>	


```

char sDisplay[256];
memset(sDisplay,0,sizeof(sDisplay));
unsigned char cReceiveBuff[280];
memset(cReceiveBuff,0,sizeof(cReceiveBuff));
int iNodeID=0x01;
iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
int iLoop=0;
int iReadLen=0;
iReturn=hsELReadDeviceInfo(ghComm,iNodeID,
cReceiveBuff,&iReadLen,&iReturnCode,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"Get Device Info : ");
    for(iLoop=0;iLoop<iReadLen;iLoop++)
    {
        sprintf(sDisplay+strlen(sDisplay),"%02X
", (BYTE)cReceiveBuff[iLoop]);
    }
}
else
{
    strcpy(sDisplay,"Get Device Info : Error : ");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

5.2.5 hsELAddAuthorization (合法卡多筆新增)

支援機型	HDE-100 系列	
API 名稱	hsELAddAuthorization	
語法	<pre>int __stdcall hsELAddAuthorization(HANDLE hComm, int iELID, int iRecord, struct_CardFormat * stRecord, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	合法卡多筆新增	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	iRecord	傳送資料結構的總筆數
	stRecord	傳送資料結構的內容 (結構內容如備註)
	iReturnCode	主機回傳的錯誤碼, 當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間. 單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<pre>int iType; char cCardNo[12]; int iHoliday; int iTime; unsigned char cActiveFloor[8]; VC Code : int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01;</pre>	

```

iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
int iLoop=0;
int iReadLen=0;
struct _CardFormat stCard[16];
memset(stCard,0,sizeof(stCard));

stCard[0].iType=0x00;
strcpy(stCard[0].cCardNo,"0000135724");
stCard[0].iHoliday=0;
stCard[0].iTime=255;
memset((char*)stCard[0].cActiveFloor,0xff,8);
iReturn=hsELAddAuthorization(ghComm,
iNodeID,0x01,stCard,&iReturnCode,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"Insert Card : OK!");
}
else
{
    strcpy(sDisplay,"Insert Card : Error!");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

5.2.6 hsELDeleteAuthorization (單筆刪除)

支援機型	HDE-100 系列	
API 名稱	hsELDeleteAuthorization	
語法	<pre>int __stdcall hsELDeleteAuthorization(HANDLE hComm, int iELID, char *cCardNo, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	單筆刪除	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cCardNo	傳送卡號數據的內容
	iReturnCode	主機回傳的錯誤碼,當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0; int iReadLen=0; iReturn=hsELDeleteAuthorization(ghComm, iNodeID,"0000135724",&iReturnCode,5000); if (iReturn==0) { strcpy(sDisplay,"Delete Card : OK!");</pre>	

	<pre> } else { if (iReturnCode==0x06) { strcpy(sDisplay,"Delete Card : Not Exist!"); } else { strcpy(sDisplay,"Delete Card : Error!"); sprintf(sDisplay+strlen(sDisplay),"%04X ", (WORD)iReturn); } } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	---

5.2.7 hsELQueryAuthorization (查詢單筆卡號許可權)

支援機型	HDE-100 系列	
API 名稱	hsELQueryAuthorization	
語法	<pre>int __stdcall hsELQueryAuthorization(HANDLE hComm, int iELID, char *cCardNo, unsigned char * cCardFormatData,</pre>	
目的	查詢單筆卡號許可權	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cCardNo	傳送卡號數據的內容
	iReturnCode	主機回傳的錯誤碼,當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0; int iReadLen=0; iReturn=hsELQueryAuthorization(ghComm,iNodeID, "0000135724",cReceiveBuff,&iReadLen,&iReturnCode,1000); if (iReturn==0) { if (memcmp(cReceiveBuff+0x01,"0000135724",10)==0) {</pre>	

	<pre> strcpy(sDisplay,"Query Card Found : OK!"); } else { strcpy(sDisplay,"Query Card Not Found : OK!"); } } else { if (iReturnCode==0x06) { strcpy(sDisplay,"Query Card : Not Exist!"); } else { strcpy(sDisplay,"Query Card : Error!"); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	--

5.2.8 hsELDeleteAllAuthorization (刪除全部合法卡)

支援機型	HDE-100 系列	
API 名稱	hsELDeleteAllAuthorization	
語法	<pre>int __stdcall hsELDeleteAllAuthorization(HANDLE hComm, int iELID, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	刪除全部合法卡	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	iReturnCode	主機回傳的錯誤碼,當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0; int iReadLen=0; iReturn=hsELDeleteAllAuthorization(ghComm, iNodeID,&iReturnCode,1000); if (iReturn==0) { strcpy(sDisplay,"Delete All Card : OK!"); } else</pre>	

	<pre> { strcpy(sDisplay,"Delete All Card : Error!"); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	---

5.2.9 hsELReadDeviceInfo(讀取裝置型號)

支援機型	HDE-100 系列	
API 名稱	hsELReadDeviceInfo	
語法	<pre> int __stdcall hsELReadDeviceInfo(HANDLE hComm, int iELID, unsigned char *cInfoData, int *iInfoLen, int * iReturnCode, unsigned int iTimeout); </pre>	
目的	讀取裝置型號	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cInfoData	接收資料的 buffer
	iInfoLen	傳回接收資料的長度
	iReturnCode	主機回傳的錯誤碼,當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<pre> VC Code : int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); </pre>	

```

int iNodeID=0x01;
iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
int iLoop=0;
int iReadLen=0;
iReturn=hsELReadDeviceInfo(ghComm,iNodeID,
cReceiveBuff,&iReadLen,&iReturnCode,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"Get Device Info : ");
    for(iLoop=0;iLoop<iReadLen;iLoop++)
    {
        sprintf(sDisplay+strlen(sDisplay),"%02X",
        (BYTE)cReceiveBuff[iLoop]);
    }
}
else
{
    strcpy(sDisplay,"Get Device Info : Error : ");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

5.2.10 hsELSetTime (設定主機時間)

支援機型	HDE-100 系列	
API 名稱	hsELSetTime	
語法	<pre>int __stdcall hsELSetTime(HANDLE hComm, int iELID, char *cDate, char *cTime, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	設定 HDE-100 系列主機時間	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cDate	日期格式 (YYYYMMDD) + 星期
	cTime	時間格式 (HHNNSS)
	iReturnCode	主機回傳的錯誤碼, 當有錯誤發生時, 錯誤代碼請參照附件 2.2.1 說明。
	iTimeout	等待主機回應的時間. 單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<p>cDate 中要加上星期 如: 2007/02/10 星期一 cDate = 200702101 最後的 1 是代表星期 cTime = 120000</p> <p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); char cDate[20]; char cTime[20]; memset(cDate,0,sizeof(cDate)); memset(cTime,0,sizeof(cTime));</pre>	

```

int iNodeID=0x01;
iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
int iPos=0;
SYSTEMTIME st;
GetLocalTime(&st);
iPos=0;
sprintf(cDate+iPos,"%04d",st.wYear);
iPos=iPos+4;
sprintf(cDate+iPos,"%02d",st.wMonth);
iPos=iPos+2;
sprintf(cDate+iPos,"%02d",st.wDay);
iPos=iPos+2;
sprintf(cDate+iPos,"%01d",(st.wDayOfWeek));

iPos=0;
sprintf(cTime+iPos,"%02d",st.wHour );
iPos=iPos+2;
sprintf(cTime+iPos,"%02d",st.wMinute );
iPos=iPos+2;
sprintf(cTime+iPos,"%02d",st.wSecond );

iReturn
=hsELSetTime(ghComm,iNodeID,cDate,cTime,&iReturnCode,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"Set DateTime : OK.");
}
else
{
    strcpy(sDisplay,"Set DateTime : Error Number : ");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

5.2.11 hsELGetTime (讀取主機時間)

支援機型	HDE-100 系列	
API 名稱	hsELGetTime	
語法	<pre>int __stdcall hsELGetTime(HANDLE hComm, int iELID, char *cDate, char *cTime, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	讀取主機時間	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cDate	日期格式 (YYYYMMDD) + 星期
	cTime	時間格式 (HHNNSS)
	iReturnCode	主機回傳的錯誤碼, 當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間. 單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<p>cDate 中要加上星期 如：2007/02/10 星期一 cDate = 200702101 最後的 1 是代表星期 cTime = 120000</p> <p>VC Code：</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); char cDate[20]; char cTime[20]; memset(cDate,0,sizeof(cDate)); memset(cTime,0,sizeof(cTime));</pre>	

```

        int iNodeID=0x01;
        iNodeID=GetDlgItemInt(IDC_EDITID,NULL);

        iReturn
=hsELGetTime(ghComm,iNodeID,cDate,cTime,&iReturnCode,500);
        if (iReturn==0)
        {
            strcpy(sDisplay,"Device Time : ");
            strcat(sDisplay,cDate);
            strcat(sDisplay," ");
            strcat(sDisplay,cTime);
        }
        else
        {
            strcpy(sDisplay,"Get DateTime : Error Number : ");
            sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
        }
        SetDlgItemText(IDC_EDITReturn,sDisplay);
    }

```

5.2.12 hsELReleaseAlarm (解除警報)

支援機型	HDE-100 系列	
API 名稱	hsELReleaseAlarm	
語法	<pre>int __stdcall hsELReleaseAlarm(HANDLE hComm, int iELID, char cAlarmFlag, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	解除警報	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	cAlarmFlag	傳送資料的內容 (備註)
	iReturnCode	主機回傳的錯誤碼, 當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間. 單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<p>cAlarmFlag</p> <p>bit0 : 1 解除黑名單警報</p> <p>bit1 : 1 解除讀卡超次警報</p> <p>bit2~7 未使用</p> <p>VC Code :</p> <pre>int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0; int iReadLen=0;</pre>	

	<pre> int iWrittenLen=12; iReturn=hsELReleaseAlarm(ghComm, iNodeID,(char)31,&iReturnCode,10000); if (iReturn==0) { strcpy(sDisplay,"Release Alarm RAC2000EL : OK!"); } else { strcpy(sDisplay,"Release Alarm RAC2000EL : Error!"); sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn); } SetDlgItemText(IDC_EDITReturn,sDisplay); </pre>
--	--

5.2.13 hsELPublicFloor (設定公共樓層)

支援機型	HDE-100 系列	
API 名稱	hsELPublicFloor	
語法	<pre>int __stdcall hsELPublicFloor(HANDLE hComm, int iELID, int iTotal, unsigned char *cFloorData, int *iReadLen, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	設定公共樓層	
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	iTotal	設定公共樓層總共幾樓
	cFloorData	傳送資料的內容 (備註)
	iReadLen	傳回接收資料的長度
	iReturnCode	主機回傳的錯誤碼, 當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
參數說明	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤	
使用時機		
備註	<p>cFloorData : 8 Bytes</p> <p>Byte 1 bit0 第 1 層樓</p> <p>Byte 1 bit1 第 2 層樓</p> <p>Byte 1 bit2 第 3 層樓</p> <p>.....</p> <p>Byte 8 bit6 第 63 層樓</p> <p>Byte 8 bit7 第 64 層樓</p> <p>0->不可開啟公共樓層 1->開啟公共樓層</p> <p>VC Code :</p> <pre>int iTotal=0; int iReturn=0; int iReturnCode=0;</pre>	

```

char sDisplay[256];
memset(sDisplay,0,sizeof(sDisplay));
unsigned char cReceiveBuff[280];
memset(cReceiveBuff,0,sizeof(cReceiveBuff));
int iNodeID=0x01;
iNodeID=GetDlgItemInt(IDC_EDITID,NULL);
int iLoop=0;
int iReadLen=0;
memset(cReceiveBuff,0x00,sizeof(cReceiveBuff));
iTotal=8;
cReceiveBuff[0]=0xff ;
cReceiveBuff[1]=0xfe ;
cReceiveBuff[2]=0xff ;
cReceiveBuff[3]=0xff ;
cReceiveBuff[4]=0xff ;
cReceiveBuff[5]=0xff ;
cReceiveBuff[6]=0xff ;
cReceiveBuff[7]=0xff ;

iReturn=hsELPublicFloor(ghComm,iNodeID,iTotal,
cReceiveBuff,&iReadLen,&iReturnCode,1000);
if (iReturn==0)
{
    strcpy(sDisplay,"Set Public Floor  :  OK!");
}
else
{
    strcpy(sDisplay,"Set Public Floor  :  Error  :  ");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

5.2.14 hsELSetReader(設定下層讀卡器)

支援機型	HDE-100 系列	
API 名稱	hsELSetReader	
語法	<pre>int __stdcall hsELSetReader(HANDLE hComm, int iELID, int iReaderID, int iIndex, unsigned char * cSendData, int iSendDataLen, int * iReturnCode, unsigned int iTimeout);</pre>	
目的	設定公共樓層	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iELID	裝置代號
	iReaderID	讀卡器編號
	iIndex	R/W 表編號
	cSendData	發送資料的內容
	iSendDataLen	發送資料的長度
	iReturnCode	主機回傳的錯誤碼,當有錯誤發生時, 錯誤代碼請參照附件 2,2.1 說明。
	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用時機		
備註	<pre>VC Code : int iReturn=0; int iReturnCode=0; char sDisplay[256]; memset(sDisplay,0,sizeof(sDisplay)); unsigned char cReceiveBuff[280]; memset(cReceiveBuff,0,sizeof(cReceiveBuff)); int iNodeID=0x01; iNodeID=GetDlgItemInt(IDC_EDITID,NULL); int iLoop=0;</pre>	

```

int iReadLen=0;
memset(cReceiveBuff,0x00,sizeof(cReceiveBuff));
cReceiveBuff[0x0]=0x00;
cReceiveBuff[0x1]=0x00;
cReceiveBuff[0x2]=0x0D;
cReceiveBuff[0x3]=0x0F;
cReceiveBuff[0x4]=0x01;
cReceiveBuff[0x5]=0x02;
cReceiveBuff[0x6]=0x00;

iReturn=hsELSetReader(ghComm,
iNodeID,1,23,cReceiveBuff,7,&iReturnCode,2000);
if (iReturn==0)
{
    strcpy(sDisplay,"Set Reader  :  OK!");
}
else
{
    strcpy(sDisplay,"Set Reader  :  Error : ");
    sprintf(sDisplay+strlen(sDisplay),"%04X ",(WORD)iReturn);
}
SetDlgItemText(IDC_EDITReturn,sDisplay);

```

第六章 ECU-680 系列相關函數說明

6.1. Read/Write 相關 API

6.1.1 hsECUReadIO (讀取 I/O 狀態)

支援機型	ECU-680 系列	
API 名稱	hsECUReadIO	
語法	int __stdcall hsECUReadIO(HANDLE hComm,int iECUID,int *cSensor,int *cRelay,int *iReturnCode,unsigned int iTimeout)	
目的	讀取 I/O 狀態	
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iECUID	裝置代號
	cSensor	傳回 Sensor 狀態值
	cRelay	傳回 Relay 值
	iReturnCode	保留
參數說明	iTimeout	等待主機回應的時間,單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤	
使用時機		
備註	<p>Sensor 狀態：</p> <p>Bit 0 — Sensor 1 的現狀, 1→Close 0→Open</p> <p>Bit 1 — Sensor 2 的現狀, 1→Close 0→Open</p> <p>Bit 2~7 — 保留</p> <p>Relay 狀態：</p> <p>Bit 0 — Relay 1, 1→動作 0→未動作</p> <p>Bit 1 — Relay 2, 1→動作 0→未動作</p> <p>Bit 2~7 — 保留</p>	

6.1.2 hsECUReadParamenter (讀取參數表全部資料)

支援機型	ECU-680 系列	
API 名稱	hsECUReadParamenter	
語法	int __stdcall hsECUReadParamenter(HANDLE hComm,int iECUID,unsigned char *cParaData,int *iParaLen,int *iReturnCode,unsigned int iTimeout)	
目的	讀取參數表全部資料	
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iECUID	裝置代號
	cParaData	傳回 參數內容
	iParaLen	傳回 參數長度
	iReturnCode	保留
參數說明	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤	
使用時機		
備註		

6.1.3 hsECUReadPower (讀取電源狀態(帶卡號))

支援機型	ECU-680 系列	
API 名稱	hsECUReadPower	
語法	int __stdcall hsECUReadPower(HANDLE hComm,int iECUID,char *cPower,unsigned char *cCardNo,int * iReturnCode,unsigned int iTimeout)	
目的	讀取電源狀態(帶卡號)	
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iECUID	裝置代號
	cPower	傳回 電源狀態
	cCardNo	傳回 使用中卡號
	iReturnCode	保留
參數說明	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤	
使用時機		
備註	當 Power ON 時卡號為空值, 表示卡片已抽出,進入延遲關閉電源	

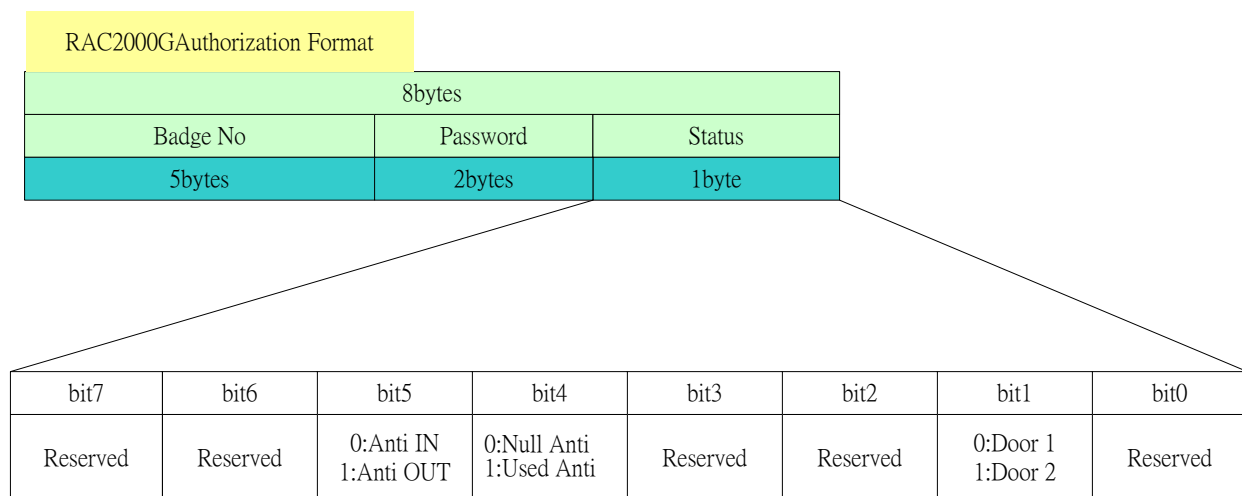
6.1.4 hsECUAddCard (新增單筆合法卡號)

支援機型	ECU-680 系列	
API 名稱	hsECUAddCard	
語法	int __stdcall hsECUAddCard(HANDLE hComm,int iECUID,char *cCardNo,int iCardLen,int iTimeZone,int Validity,int ValidityYear,int ValidityMonth,int ValidityDay,int ValidityHour,int ValidityMinute , unsigned int iTimeout)	
目的	新增單筆合法卡號	
	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iECUID	裝置代號
	cPower	傳回 電源狀態
	cCardNo	傳回 使用中卡號
	iReturnCode	保留
參數說明	iTimeout	等待主機回應的時間.單位為毫秒(ms)。
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤	
使用時機		
備註	當 Power ON 時卡號為空值, 表示卡片已抽出,進入延遲關閉電源	

6.1.5 hsECUPolling (Polling ECU-680 設備)

支援機型	ECU-680 系列	
API 名稱	hsECUPolling	
語法	int __stdcall hsECUPolling(HANDLE hComm,int iECUID,int iPrevRecord,stPollList *stRecord,int *iRecord,int *iReturnCode,unsigned int iTimeout)	
目的	發出 Poll 指令給設備，返回獲得的紀錄	
參數說明	hComm	handle 值, 可以是 COM 或是 TCP/IP
	iECUID	裝置代號
	iPrevRecord	上一次收到的資料筆數
	stRecord	存放傳回資料，是個結構
	iRecord	傳回的事件筆數
	iReturnCode	保留
	iTimeout	等待逾時，單位為 ms
回傳值	當此功能成功會返回 HF_RET_SUCESS, 否則會有相對應的錯誤訊息出現.	
使用場合		
備註	<pre>typedef struct stPollList { char cEventCode[5]; //Event Code char cDateTime[20]; //Date Time char cCard[20]; //Card Number char cDeviceID[10]; //Device ID char cReaderID[10]; //Reader ID } stPollList;</pre>	

1.3 RAC-2000G 合法卡許可權格式說明



-卡號與密碼：不足位元或沒有密碼補 FF。

-卡號與密碼內容固定為壓縮格式, 2 個數位為 1 個 byte。

1.4 RAC-2000 系列刷卡歷史資料

History Log Format				
19bytes				
Len	Status1	Status2	Datetime	Badge No
1byte	1byte	1byte	4bytes	1-13 byte

長度： 4bit 說明卡號長度

狀態 1：4bit

bit0 =0：擊鍵卡號

1：刷卡

bit2,1=00：reader 1

01：reader 2

10：reader 3

11：reader 4

bit3 =0：卡號事件

1：有事件代碼的事件(無卡號) 事件代碼在第 7 第 8 個 byte

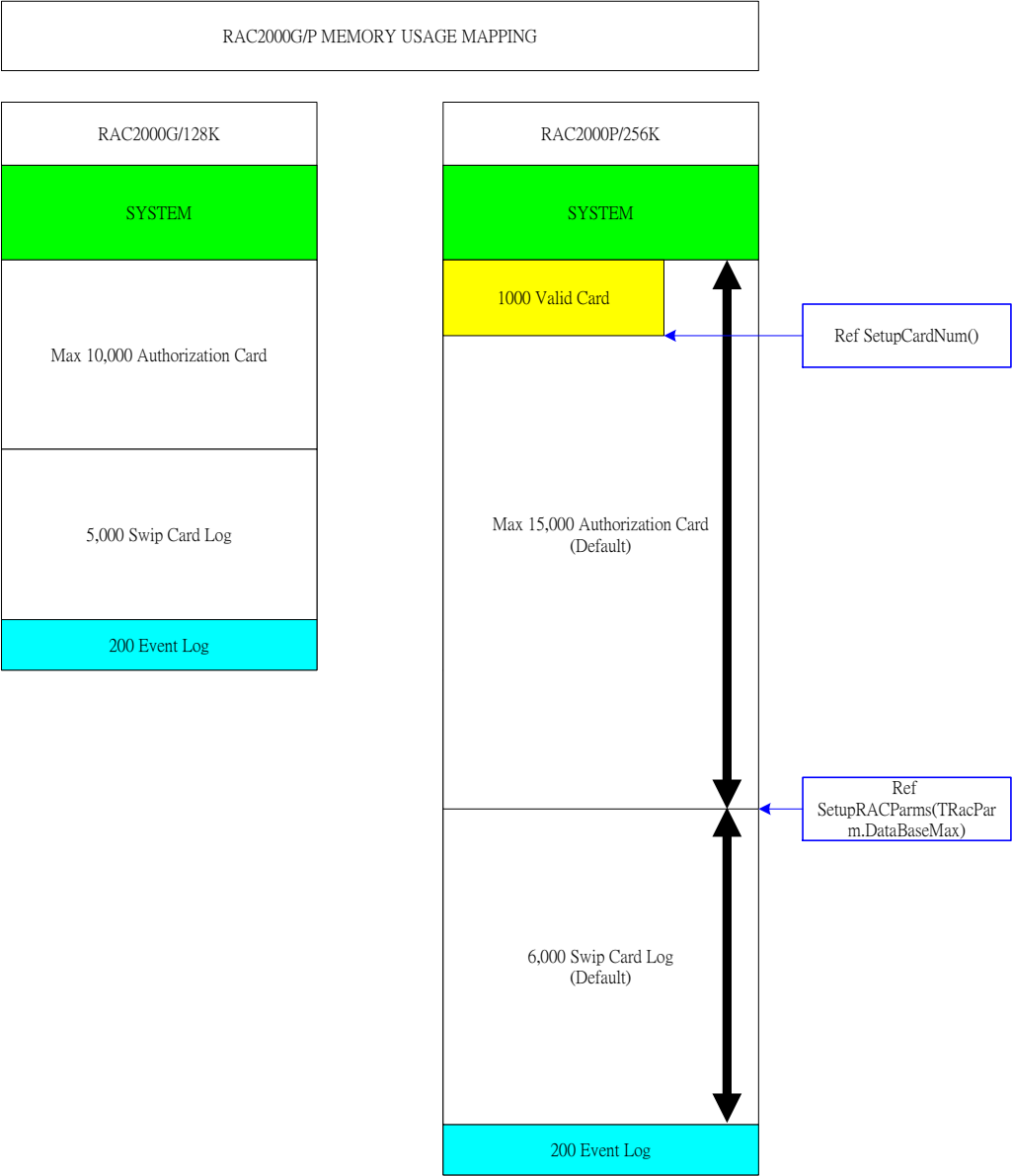
卡號依長度來存數據

狀態 2：

- 0 = 正常刷卡
- 1 = Master Card
- 2 = Release Code
- 3 = 脅迫卡
- 4 = 脅迫碼
- 5 = 臨時卡
- 6 = 黑名單
- 10 = 來賓卡
- 11 = 來賓卡(不限次數)
- 20 = 沒有找到卡號
- 61=刷卡正確+密碼正確
- 62=刷卡正確+密碼不正確
- 63=刷卡+密碼時,刷卡是用擊鍵
- 67=Anti 錯誤

日期時間：共 4Byte(傳輸時 Lo Byte 在前),0~5bit 代表秒,6~11bit 代表分鐘,12~16 代表小時,17~21 代表日期,22~25 代表月份,26~31 代表年份(2000+N)。

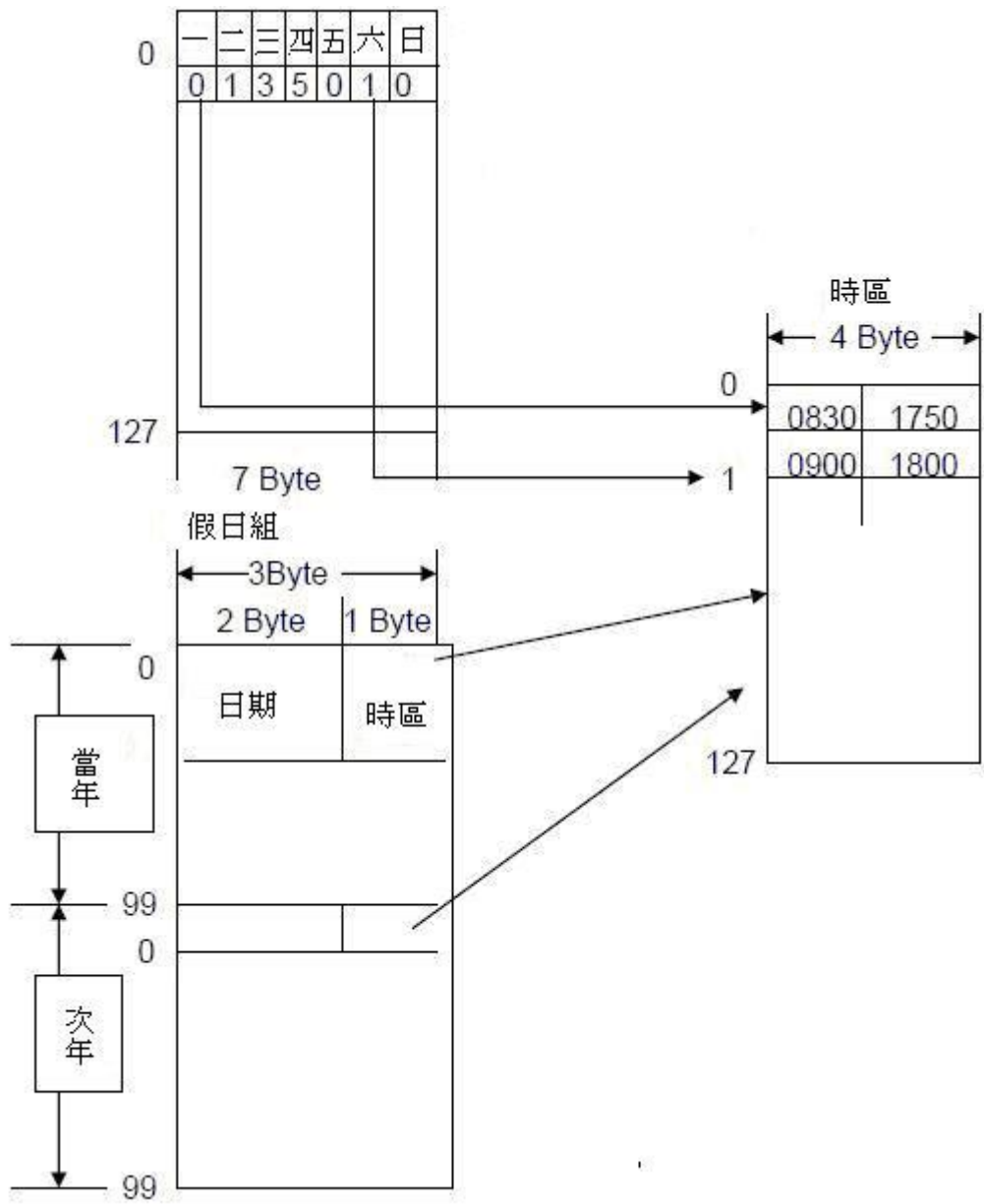
1.5 記憶體使用對照表



1.6 時段表對照說明

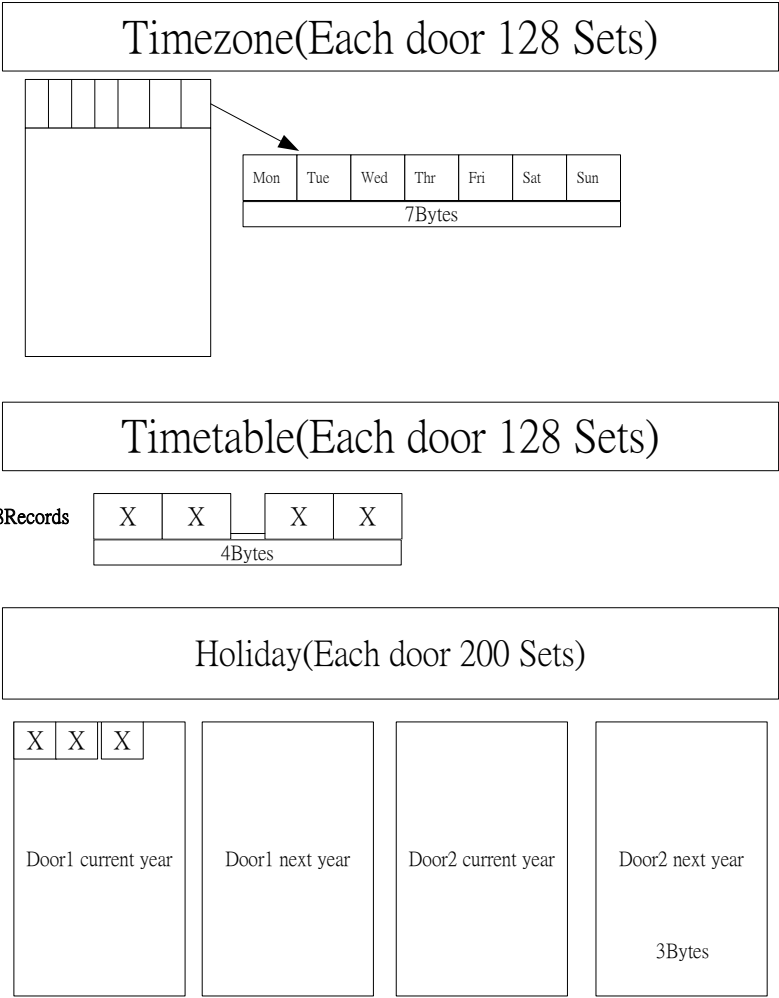
時段,時區,假日

時段：



-不使用的時段組，假日組，時區組填 FF。

1.7 假日表對照說明



Timetable(Each door 128 Sets)

128Records

X

X

X

X

4Bytes

Holiday(Each door 200 Sets)

X

X

X

Door1 current year

Door1 next year

Door2 current year

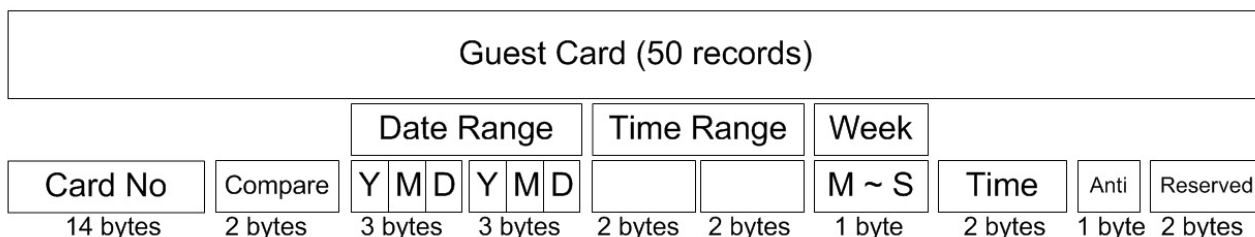
Door2 next year

3Bytes

-不使用的時段組，假日組，時區組填 FF。

1.8 來賓卡格式

RAC-2000 RMA Usaged



1.9 事件碼回傳值(Sensor/Relay)

0120 H	脅迫碼通過
0122 H	解除碼解除警報
0123 H	Master Card (Code) 修改
0301 H	偵測外力破壞(Tamper) → Case Sensor
0306 H	冷啟動(Cold Startup)
0307 H	重新啟動(Reset)
0311 H	門被外力開啟 → Door Sensor 門被破壞
0312 H	門持續開啟警告
0313 H	門持續開啟警告解除
0318 H	開門(Door Opened) → 一般合法開門
0319 H	關門(Door Closed) → Door Relay Close
031A H	開門鈕觸動(Button Pressed) → Push Button
0401 H	記憶體已滿(Memory Full)
041B H	系統初始化

備註：以上皆為 RAC-2000 系列一個門的記憶體規劃(除了卡號許可權格式是共用)。

1.10 EEPROM 相關記憶體規劃

長度	地址	說明	預設值	備註
2	0000H	EEPROM Test	5AH,A5H	開機測試用
2	0002H	機器代碼	'R', '2'	G=R1,P=R2
2	0004H	取碼有效位數	01H, 00H	GetCard_CODE
2	0006H	比較有效位數	00H, 00H	CompCard_CODE
14	0008H	Master Card(不用的位數補 FFH)	'30191000'	MasterCard_CODE
14	0016H	脅迫卡(不用的位數補 FFH)	1190	ForceCard_CODE
1	0024H	刷脅迫卡後是否要輸入脅迫碼 bit0,4 0=不需要,1=需要 輸入脅迫碼後是否要刷卡 Bit1,5 0=不需要,1=需要	00H	ForcePin_CODE
1	0025H	刷脅迫卡後 Relay 是否要指定輸出 0=不用(1,2,4,8 Relay1,2,3,4)	82H	ForceRelay_CODE
14	0026H	脅迫碼(不用的位數補 FFH)	1190	ForceCode_CODE
1	0034H	Door1 動作時間	04H	Door1Time_CODE
1	0035H	Door2 動作時間	04H	Door2Time_CODE
14	0036H	解除碼(不用的位數補 FFH)	'0000'	ReleaseCode_CODE
16	0044H	持續開門時段 8 組*2 門	FFH	SustainedOpenBlock_CODE
2	0054H	假日表的年份	0	HolydayYear, 每個門一個位元組, 如果是 2011 年, 應該寫入: 0x11,0x11
2	0056H	合法卡號在 SRAM 的筆數	0	DataBasePoint_CODE(Hi,Lo)
2	0058H	合法卡號的最大筆數	10000	DataBaseMax_CODE(Hi,Lo)
1	005AH	內定的 ID(當 DIP SW = 0)	1	NodeID_CODE
1	005BH	鮑率 1-6=1200,2400,4800,9600, 19200,38400	5	BaudRateSet_CODE
1	005CH	保留	0	RAC2000P=0,RAC2000G=1
1	005DH	黑名單是否要啟動 Realy bit0,4: 0 不啟動,1 啟動	11H	BlacklistRelay_CODE

1	005EH	解除碼要帶卡號 0=不帶卡號,1=帶卡號	0	ReleaseWithCard_CODE
1	005FH	Door Sensor 是否要啟動警報 bit0,4 = 1 Reader Buzzer 叫,bit1,5 = 1 Alarm Relay 動作	0	DoorSenserMode_CODE
1	0060H	設定 Door Sensor1 偵測的秒數	0	DoorSenser1Sec_CODE
1	0061H	設定 Door Sensor2 偵測的秒數	0	DoorSenser2Sec_CODE
1	0062H	bit0=1 儲存資料達 90%警告 bit1=1 儲存資料不可覆蓋 bit2=1 儲存資料滿時起動 Alarm Relay	0	HistoryMemorySet_CODE
1	0063H	設定錯誤次數	0	ErrorTime_CODE
1	0064H	達到錯誤次數時系統要停止操作的秒數	0	ErrorHaltTime_CODE
1	0065H	Sensor 輸入是否要反相 0=不要,1=要 bit0,1,2,3=Case1,2,3,4 bit4,6=PushButton1,2 bit5,7=DoorSensor1,2 // bit0,1=Case1,2 bit4,6=PushButton1,2 bit5,7=DoorSensor1,2	0	SensorAction_CODE
1	0066H	Case Sensor 是否要啟動警報 1 Byte bit0,4 = 1 Reader Buzzer 叫,bit1,5 = 1 Alarm Relay 動作 // bit0,4 = 1 Reader Buzzer 叫,bit1,5 = 1 Alarm Reader 叫	0	CaseSensorMode_CODE
1	0067H	保留	0	

1	0068H	0=無 Anti (2000P/G) 1=Anti By Door (2000P/G) 只動作 Door1 Relay IN/OUT Reader1-2 IN,Reader3-4 OUT 2=Anti By Reader (2000P) Door1 Relay IN/OUT Reader1 IN,Reader2 OUT Door2 Relay IN/OUT Reader3 IN,Reader4 OUT 3=Anti By Door(2000P) Door1 Relay IN, Door2 Relay OUT Reader1-2 IN,Reader3-4 OUT	0	AntiMode_CODE*
1	0069H	重複刷卡檢查的秒數 0=關閉 1~255=1~255 秒	0	
1	006AH	保留	10	
1	006BH	保留	10	
1	006CH	0=Disable,1=Enable bit0=雙門互鎖	0	
1	006DH	保留	0	
1	006EH	保留	0	
1	006FH	Reader 狀態 bit0=保留 bit1=保留 bit2=0→Disable 數字鍵 1→Enable 數字鍵 bit3=0→Disable 功能鍵 1→Enable 功能鍵 bit4=保留 bit5=保留 bit6=保留 bit7=0→Disable Case Sensor 1→Enable Case Sensor	7FH	

1	0070H	Door Relay 恢復時間設定 0=Door Relay 動作時間結束才關閉 Door Relay 1=DOOR SENSOR 恢復即關閉 Door Relay	0	
1	0071H	副刷卡槽不檢查時段 0=檢查時段 1=不檢查時段	0	
1	0072H	保留	60	
1	0073H	保留	0	
1	0074H	Wiegand 26Bit 8 碼卡,前面補 2 碼 0 成為 10 碼卡 1=補 2 碼 0	0	
1	0075H	無合法卡時 0=刷任意卡開門, 1=不開門	0	
5	0076H	76H-7AH 保留	FFH	
1	007BH	Weigand 讀頭是否啟動同位元檢查(設定為 00H 為不檢查, 其他值就檢查)	01H	
1	007CH	保留	60	
3	007DH	7DH-7FH 保留	FFH	
8	0080H	保留	FFH	
8	0088H	密碼管制時段代碼		
16	0090H	每門八組密碼管制時段代碼, 對應到時段表 0090H~0097H: 門一 0098H~009FH: 門二		不支援跨日(跨日需拆成兩個時段, 例如 2100~2359, 0000~0700) 各門第一個 Byte 為 0xFF, 則應用一般密碼管制作法, 否則時段內才要用密碼
1	00A0H	日光節約時間啟動 Bit6~Bit0 =0 關閉(Disable) =1 調前 =2 調後 (其他值一律視為 Disable) Bit7: 調整的狀態 (請不要變更此 Bit 狀態) =0 尚未調整 =1 已經調整	00H	
4	00A1H	日光節約時間起始月日時分	均為 00H	

		(MMDDhhmm (BCD 格式))		
4	00A5H	日光節約時間結束月日時分 (MMDDhhmm (BCD 格式))	均為 00H	
2	00A9H	日光節約調整時間 hhmm (BCD 格式)	均為 00H	
	00ABH	ABH-F9H 保留	FFH	
6	00FAH	FAH-FFH 系統使用,請勿修改此資料		
128*7	0100H	時段表 128 組*7Byte(門 1)		TimeBlock1
128*7	0480H	時段表 128 組*7Byte(門 2)		TimeBlock2
128*4	0800H	時間表 128 組*4Byte(門 1)		TimeTable1
128*4	0A00H	時間表 128 組*4Byte(門 2)		TimeTable2
200*3	0C00H	假日表 200 組*3Byte(門 1)		Holyday1
200*3	0E58H	假日表 200 組*3Byte(門 2)		Holyday2
	10B0H	10B0H-10FFH 系統使用, 勿修改		Blank2
	1100H	1100H-19FFH 保留		
480	1A00H	Wiegand 解碼設定		(RAC-2000WS/RAC-2000WSN Only)
	1BE0H	1BE0H-1FFFH 保留		

Wiegand 解碼設定 (位址 1A00H~1BDFH:共 480Bytes)

各種 bit 解碼設定,每種包含 10 Byte,狀態 1Byte , 每個 bit Enable/Disable 狀態 9Byte
狀態 1Byte : bit0 是否解碼 1=Enable / 0=Disable

bit1 卡號解碼規則

1=特別解碼規則,依客戶需求增加

0=一般解碼規則,取最大 32 bit 轉成 10 碼 10 進制卡號

bit2-7 = 0 (保留)

bit Enable/Disable 狀態 9Byte : 初始值都是 FFh , (1=Enable / 0=Disable)

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
11111111

bit1bit72

下表各 bit 解碼初始值

	狀態	Enable/Disable (bit1 – bit72)
26 Bit	03h	7Fh,FFh,FFh,80h,00h,00h,00h,00h,00h
27 Bit	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh
28 Bit	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh

29 Bit	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh
30 Bit	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh
31 Bit	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh
32 Bit	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh
33 Bit	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh
34 Bit	01h	7Fh,FFh,FFh,FFh,80h,00h,00h,00h,00h
35 Bit	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh
36 Bit	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh
.....
72 Bit	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh
保留	00h	FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh,FFh

1.11 AntiPassback 設定說明

如果要啟動 AntiPassback 可以使用 hacSetEEData 的功能，對位址 0068H 寫入參數即可。

ANTI(4Bit)

AntiMode = 1 By Door 只動作 Door1 Relay IN/OUT Reader1-2 IN,Reader3-4 OUT	AntiMode = 2 By Reader Door1 Relay IN/OUT Reader1 IN Reader2 OUT	AntiMode = 2 By Reader Door2 Relay IN/OUT Reader3 IN Reader4 OUT
AntiMode = 3 By Door Door1 Relay IN, Door2 Relay OUT Reader1-2 IN,Reader3-4 OUT		
bit4 : 0 : Anti 啟動的初值 1 : 此卡號已經有使用過	bit4 : 0 : Anti 啟動的初值 1 : 此卡號已經有使用過	bit6 : 0 : Anti 啟動的初值 1 : 此卡號已經有使用過
bit5 : 0 : 上次刷卡是 IN 1 : 上次刷卡是 OUT	bit5 : 0 : 上次刷卡是 IN 1 : 上次刷卡是 OUT	Bit7 : 0 : 上次刷卡是 IN 1 : 上次刷卡是 OUT

1.12 RAC-2000G SRAM 相關記憶體規劃

地址範圍	長度	說明	備註
000000H-00001FH	32	資料指標位址	
000020H-00065FH	1600	來賓卡地址(50 筆*32Byte)	
000660H-000C9FH	1600	儲存 Event 位址(200 筆*8Byte)	
000CA0H-01451FH	10000	合法卡地址(10000 筆*8Byte)	
014520H-01FFFFH		刷卡紀錄位址	

備註：以上內存規劃僅支援 RAC-2000G。

1.13 RAC-2000 系列 SRAM 相關內存規劃

地址範圍	長度	說明	備註
000000H-00001FH	32	資料指標位址	
000020H-00065FH	1600	來賓卡地址(50 筆*32Byte)	
000660H-000C9FH	1600	儲存 Event 位址(200 筆*8Byte)	
000CA0H-027D9FH	10000	合法卡地址(10000 筆*16Byte)	
027DA0H-03FFFFH		刷卡紀錄位址	

備註：以上內存規劃支援 RAC-2000P/PV/PS/PN/PSN/WS/WSN, HDP-100/100S 等機種。

附錄 2：HDE-100 格式說明

2.1 回傳錯誤代碼表

代碼(16 進制)	說明
0001	數據長度與合法卡長度不符
0002	合法卡長度不符
0003	已超過可容納的筆數
0006	卡號不存在
0007	設定數據長度過長
0008	讀取數據長度過長
0009	數據長度不符
000F	輸入資料錯誤

2.2 參數表

長度	地址	參數內容	說明
1	0	刷卡紀錄是覆蓋	0 — 沒有提示聲(default) 1 — 當紀錄容量達 90%時，Reader 會有 嗶嗶的提示聲
1	1	沒有合法卡時是否開門	0 — 不開門 1 — 開門 (default)
2	2	樓層按鈕開啟時間 (Lo,Hi)	兩個 Byte(Lo,Hi) 單位時間 10ms (default : 10 秒)
1	4	不管制時段索引	(保留不用)
12	5	解除碼	當發生警報時使用解除碼，解除警報。 非壓縮，未滿 12 Byte 後面補 0 (default : 0000) 解除碼可解除： 1.黑名單警報 2.讀卡超次警報
1	17	重複讀卡警報	可設定是否輸出警報 0 — 不輸出警報 (default) 1 — 輸出警報的秒數
1	18	重複讀卡延遲時間	0 — 系統不停止 (default) 1 ~ 255 — 延遲時間(秒)

8	19	公共樓層	64 bit 指向 64 個主機 ID (梯號)
1	27	不管制模式	(保留不用)
12	28	Master Card	a.不分時段可到各樓層 b.解除警報 c.解除維護或停用狀態 一次一個
1	40	設定密碼模式(保留)	
1	41	平常密碼時段索引(保留)	
1	42	假日密碼時段索引(保留)	
		未使用	

2.3 資料格式

2.3.1 主機上的合法卡格式

卡別(1)	卡號(12)	假日(1)	時段(1)	動作樓層(8)
-------	--------	-------	-------	---------

- 卡別： (1 byte)

說明：

0x00： 一般卡
進入比對流程,並記錄比對結果
0x01： 黑名單發警報
觸發警報，並記錄為黑名單發警報

- 卡號： 12 Byte，非壓縮，未滿 12 Byte 後面補 0。
- 假日：
 - Bit0： 0 -- 假日不管制
1 -- 假日管制
 - Bit 1~7 保留
- 時段：時段的索引,0 表示不通過,255 表示通過。
- 動作樓層：當合法時所要觸動的樓層 $8 * 8 = 64$ bit 可控制 64 樓層。
 - 當 bit =0 – 不可開啟的樓層
 - bit=1 – 可開啟的樓層

Byte1, Byte2, Byte3, Byte4, Byte5, Byte6, Byte7, Byte8,

Byte 1 bit 0 第 1 層樓

Byte 1 bit 1 第 2 層樓

Byte 1 bit 2 第 3 層樓

Byte 1 bit 3 第 4 層樓

.....

Byte 8 bit 0 第 57 層樓

.....

Byte 8 bit 7 第 64 層樓

2.3.2 卡片上的合法卡格式

標準版 電梯mifare 卡片規劃 V1.0



Block 1:

卡別: Byte 1
 = 03 - 漢軍員工卡
 = 04 - 漢軍臨時卡

有效日期: Byte 2, Byte 3

Byte 2: bit 7....bit 0
 0000000 0
 1-127年

Byte 3: bit 7....bit 0
 000 00000
 1-12月 1-31日

有效日期

起迄時分: Byte 4, Byte 5, Byte 6

Byte 4: bit 7....bit 0
 00000 000
 0-23時

Byte 5: bit 7....bit 0
 000 00000
 0-59分

Byte 6: bit 7....bit 0
 000000 00
 0-23時 0-59分

起始時分 結束時分

※當有效日期Byte2 及Byte3 為0xff,0xff 則 此卡片永久有效

下區指示: Byte 7

指出下個欲讀取 Sector, 0xff 表示沒有下個 Sector, 若有下個 Sector, 則下個 Sector 的前3個Block的權限資料規劃為第1Byte為下區指示, 第2bytes格式為 1bytes梯號+8bytes樓層權限, 共可儲存 5 梯的權限

公共梯號 bitmask: Byte 9~14

當 bit = 0 - 不可開啓公共樓層的梯號
 = 1 - 可開啓公共樓層的梯號

Byte 9 bit 0 梯號0
 Byte 9 bit 1 梯號1
 Byte 9 bit 2 梯號2
 Byte 9 bit 3 梯號3

 Byte 14 bit 0 梯號46
 Byte 14 bit 7 梯號47

梯號A: Byte 15

當主機的ID與梯號A相同, 則才有A梯權限

A梯權限: Byte 16, Block 2 Byte 1~7

即為可開啓的樓層, 格式Byte 1的Bit0為最低樓層, Byte 7的Bit7為最高樓層

梯號B, B梯權限 同上

Block 3~4: 保留

2.3.3 假日表

(地址 0：共 601 bytes)

年份								
今年 明年	第 1 個假日(3 Byte)					第 100 個假日	
	月	日	時段索引					

年份(1 byte)：今年是西元幾年?,例如：西元 2010=>0x10 (BCD 格式)

時段索引：0 表示無時段
 1~128 索引至時段表
 255 全時段(全天)

註明：1.月,日為 BCD 格式例如 12 月 1 日 =>0x12,0x01

2.每一個索引可以有一百個假日,在月,日填入 0xFF,0xFF 表示不用再比對下去。

2.3.4 星期表格式

(地址 601： 共 1778 bytes)

星期						
一 二 三 四 五 六 日						
索引 1 2 . . 254	時段索引					

時段索引：0 表示無時段
 1~128 索引至時段表
 255 全時段(全天)

2.3.5 時段表格式

(起始位址：2379, 共 4096 bytes)

索引	第一組			第八組	
	開始時間	結束時間				
	HHMM	HHMM				
	2					
	.					
128						

開始時間：兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

13：25 => 0x13,0x25 (BCD 格式)

結束時間：兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

17：01 => 0x17,0x01 (BCD 格式)

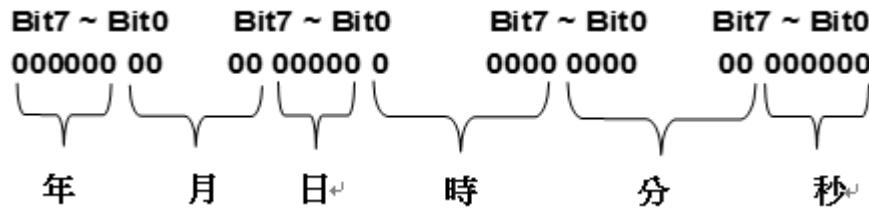
註明：在一天中最多可以對應到八個時段,在”開始時間”填入 0xFF,0xFF 表時不用再比對下去。

2.3.6 刷卡及事件記錄格式

裝置 ID(1byte)+時間(4byte)+狀態(1byte) +卡號(12 byte)
+事件碼(12 byte)

裝置 ID(1byte)： 當 ID 為 255 時則此筆記錄為主機所產生。

時間(4byte) Byte1,Byte2,Byte3,Byte4



● 狀態

0x00	表示為事件,則後面為事件碼
0x01	合法卡

0x04	許可權不合
0x05	假日許可權不合
0x06	卡片時效已過
0x07	無此卡號
0x0B	黑名單發警報
0x0C	解除警報
0x16	在常開時區刷卡
0x18	無任何卡號開門

- 事件碼

事件碼	說明
031B01	黑名單發警報(主機)
031B02	讀卡超次發警報(主機)
020101	PUSH1 ON(主機)
031C01	解除黑名單警報(主機)
031C02	解除讀卡超次警報(主機)
020201	PUSH1 OFF(主機)
0301	Case Sensor 警報
0305	Case Sensor 警報解除
0306	冷啟動(主機)
020102	PUSH2 ON
020202	PUSH2 OFF
0117	刷 master card

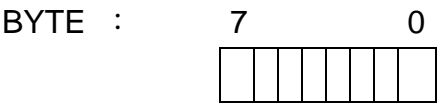
- 卡號：未滿 12 Byte 時本主機會在後面補 0。
- 事件碼：未滿 12 Byte 時本主機會在後面補 0。

附錄 3：RAC-960/RAC-970 格式說明

3.1 基本格式

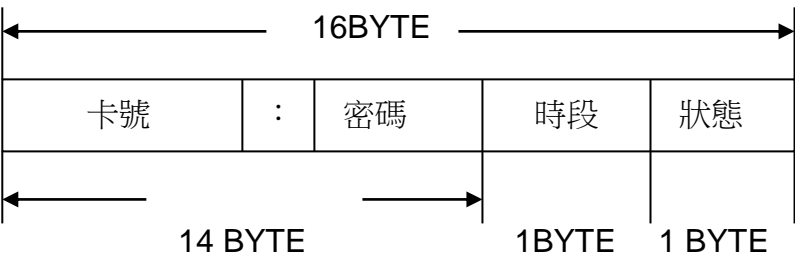
RAC-960/970 ID 範圍 1-255 (RS-232/485 通訊模式)

系統 BYTE 表示：

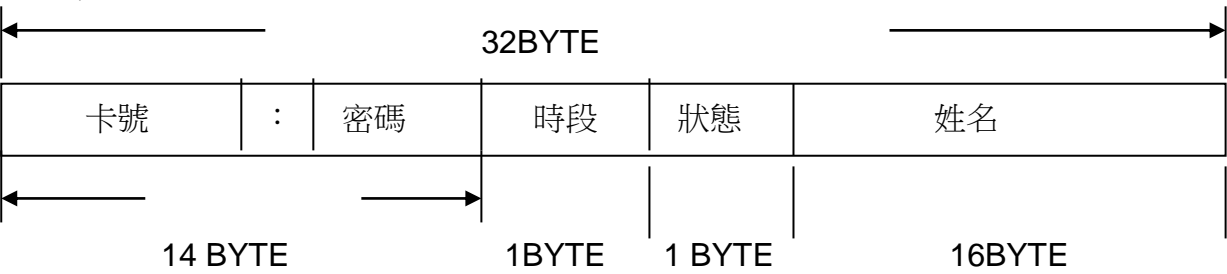


3.2 卡號權限格式

標準模式：



合法卡 + 姓名顯示：



相關資訊需參照”參數區”中的位址 0013H

1	0013H	主機模式 0=標準模式 (同 RAC-2000) 1=合法卡 + 姓名顯示	00H	
---	-------	---	-----	--

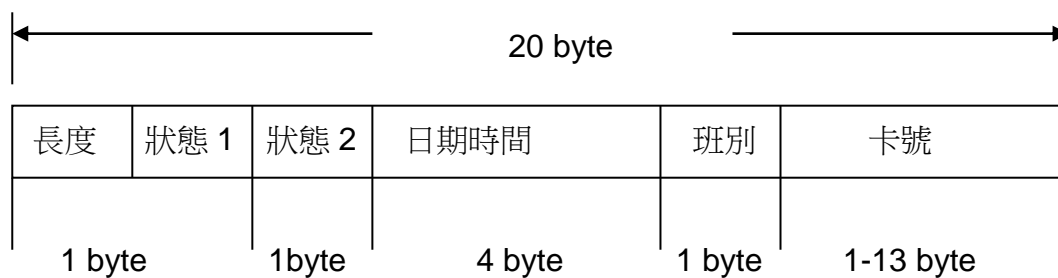
- 卡號 ASCII
- 密碼：不足位或沒有密碼補 FFH
- 密碼內容固定為壓縮格式, 2 個數字為 1 個 byte

- 卡號與密碼用' : '[冒號] 分隔
- 時段 128 組 (0~127)
- 狀態說明：

Bit0	0=正常卡 1=黑名單
Bit1	960/970 保留
Bit2	0=假日不檢查 1=假日檢查
Bit3	0=時段不檢查 1=時段檢查
Bit4~5	960/970 保留
Bit6~7	保留

- 姓名：不顯示的部分請填入 20H

3.3 刷卡歷史資料



- 長度： 4bit 說明卡號長度
- 狀態 1：

Bit0	0=按鍵輸入卡號 1=刷卡
Bit1~2	0=內 Reader 1=外 Reader 2= 960/970 保留 3= 960/970 保留
Bit3	0=刷卡紀錄 1=事件紀錄

- 狀態 2：

Bit0~7	0 = 正常刷卡
	1 = 母卡
	2 = 解除碼
	3 = 脅迫碼
	5 = 臨時卡
	6 = 黑名單
	10 = 來賓卡
	11 = 來賓卡(不限次數)
	20 = 沒有找到卡號
	21 = 卡號時段錯誤
	61 = 刷卡正確+密碼正確
	62 = 刷卡正確+密碼不正確
	63 = 刷卡+密碼時,刷卡是用按鍵輸入
	67 = Anti 錯誤
	75 = 比較有效位數設定的長度錯誤

- 日期時間：共 4Byte(傳輸時 Lo Byte 在前),0~5bit 代表秒,6~11bit 代表分鐘,12~16 代表小時,17~21 代表日期,22~25 代表月份,26~31 代表年份(2000+N)。
- 時間(4byte) Byte1,Byte2,Byte3,Byte4



- 班別： 參數區的主機模式設定

0=空白	1=DUTY ON
2=DUTY_OFF	3=BREAK OUT
4=BREAK_IN	5=OT_START
6=OT_END	

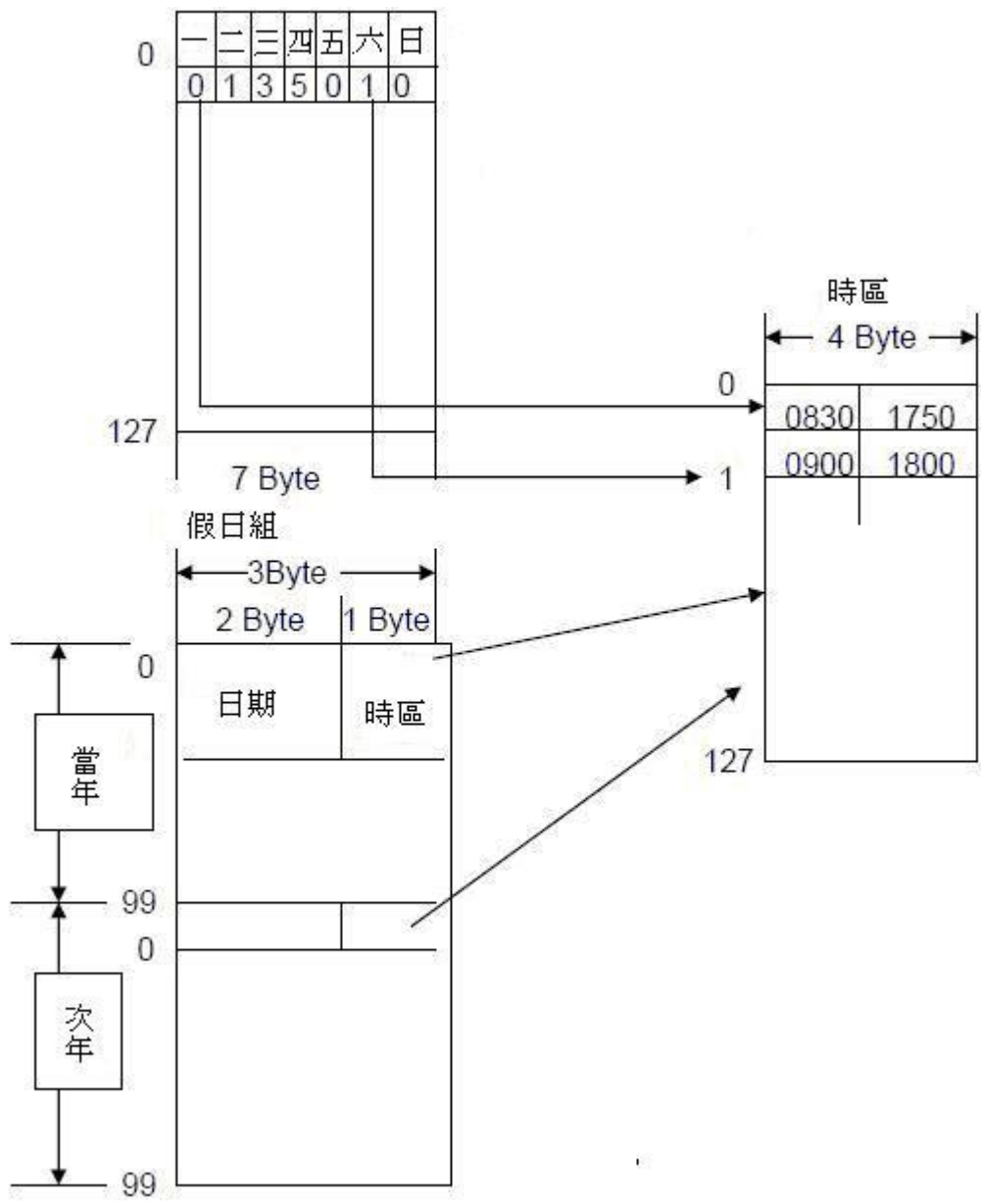
- 卡號依長度來存資料。

Polling 有班別資料請用改用 1FH Command,原本 10H Command 不會有班別資料。

3.4 時段表對照說明

時段/時區/假日

時段：



不使用的時段組，假日組，時區組填 FF

3.5 持續開門



3.6 事件碼回傳值(Sensor/Relay)

0120H 脅迫碼通過

0122H 解除碼解除警報

0123H Master Card (Code)修改

0301H 偵測外力破壞(Tamper) → Case Sensor

0306H 冷開機(Cold Startup)

0307H 重新啟動(Reset)

0311H 門被外力開啟 → Door Sensor 門被破壞

0312H 門持續開啟警告

0313H 門持續開啟警告解除

0318H 開門(Door Opened) → 一般合法開門

0319H 關門(Door Closed) → Door Sensor Close

031AH 開門鈕觸動(Button Pressed) → Push Button

0401H 記憶體已滿(Memory Full)

041BH 系統初始化

3.7 Flash Memory 相關記憶體規畫

(位址 0000H~00FFH：共 256Bytes),0000H~0013H 屬於系統參數

長度	位址	說明	預設值	備註
1	0000H	通訊方式 Bit7 = 0：RS-485/232 Bit0~6：鮑率 1=9600 2=19200 3=38400 4=115200 Bit7 = 1：TCP/IP Bit0~6：保留	02H	
1	0001H	讀頭型態 0=Mifare,副讀卡機 T1 1=EM,副讀卡機 T1 2=無內讀頭,副讀卡機 Wiegand	00H	
4	0002H	IP		
4	0006H	Mask		
4	000AH	Gateway		
2	000EH	Port		
2	0010H	合法卡最大筆數 (因 Flash Memory 的特性,筆數調整後需可被 256 筆整除,可調整的範圍 6144~15104)	00H,28H	Lo Byte,Hi Byte 2800H = 10240 筆
1	0012H	機器的 ID	01H	RS-232/485 使用
1	0013H	主機模式 0=標準模式 (同 RAC-2000) 1=合法卡 + 姓名顯示	00H	
1	0014H	主機狀態設定 Bit0：副刷卡槽不檢查時段 0=不檢查時段 1=檢查時段 Bit1：室內/室外模式選擇	00H	

		0=室內 1=室外 Bit2：黑名單是否要啟動 Alarm Realy 0=Disable 1=Enable Bit3:副讀頭於刷卡加密模式時,不需 輸入密碼 0=Disable (需按密碼) 1=Enable (不需按密碼) Bit4:卡號顯示 0=Disable (不顯示卡號) 1=Enable (顯示卡號) Bit5：非法卡是否儲存刷卡紀錄 0=儲存 1=不儲存 Bit6：儲存資料不可覆蓋 0=覆蓋 1=不覆蓋且儲存資料達 90%以上 警告 Bit7：儲存資料滿時起動 Alarm Relay 0=Disable 1=Enable		
1	0015H	假日表的年份	09H	
10	0016H	Mifare 設定值	00H,00H,00H,00H, FFH,FFH,FFH, FFH,FFH,FFH	參考 R/W Table 54
1	0020H	LCD 日期顯示格式 0=YYYY/MM/DD 1=MM/DD/YYYY 2=DD/MM/YYYY	00H	
1	0021H	LCD 狀態設定 Bit0~3 = 保留 Bit4~5 = LCD 背光模式 (0=Auto,1=開啟,2=關閉)	10H	初始值 LCD 背光模 式開啟

		Bit6~7 = 保留		
1	0022H	LCD 回復 Ready 時間 0=內定值 8 秒 1~255=1~255 秒	00H	
1	0023H	按鍵狀態設定 Bit0 = 按鍵狀態 (0=關閉,1=開啟) Bit1~3 = 保留 Bit4~5 = 按鍵背光模式 (0=Auto,1=開啟,2=關閉) Bit6~7 = 保留	01H	初始值 按鍵狀態開啟 按鍵背光模式 Auto
2	0024H	取碼有效位數	01H, 00H	
2	0026H	比較有效位數	00H, 00H	
1	0028H	0=Anti Disable 1=Anti Enable	00H	
1	0029H	語系選擇 0 = English 1 = 繁體 2 = 簡體	00H	
1	002AH	Sensor 動作狀態 (1=Close,0=Open) Bit0 = 0 (Case Sensor) Bit1 = 0 (Door Sensor) Bit2 = 1 (Push Button) Bit3~7 = 保留	04H	
1	002BH	Case Sensor 是否要啟動 Alarm Relay Bit0 = Buzzer 警報 Bit1 = Alarm Relay 動作 Bit2~7 = 保留	01H	
1	002CH	設定 Door Sensor 偵測的秒數 0=關閉 Door Sensor 1~255=偵測秒數 1~255 秒	00H	
1	002DH	Door Sensor 是否要啟動警報 Bit0 = Buzzer 警報 Bit1 = Alarm Relay 動作	01H	

		Bit2~7 = 保留																						
1	002EH	設定錯誤次數 0=關閉 1~255=錯誤次數 1~255 次	00H																					
1	002FH	達到錯誤次數時系統要停止操作的 秒數 0=關閉 1~255=1~255 秒	00H																					
14	0030H	母卡(不用的位數補 FFH)	'30191000'																					
1	003EH	保留																						
1	003FH	保留																						
14	0040H	解除碼(不用的位數補 FFH)	'0000'																					
1	004EH	保留																						
1	004FH	Relay 動作模式選擇 <table border="1"> <tr> <td></td><td>Relay0</td><td>Relay1</td><td>Relay2</td><td>Relay3</td></tr> <tr> <td>模式 0</td><td>Door</td><td>Alarm</td><td>響鈴</td><td>門鈴</td></tr> <tr> <td>模式 1</td><td>X</td><td>Door</td><td>Alarm</td><td>門鈴</td></tr> <tr> <td>模式 2</td><td>X</td><td>Door</td><td>Alarm</td><td>響鈴</td></tr> </table>		Relay0	Relay1	Relay2	Relay3	模式 0	Door	Alarm	響鈴	門鈴	模式 1	X	Door	Alarm	門鈴	模式 2	X	Door	Alarm	響鈴	00H	Relay0 代表 主機上的 Relay Relay1~3 代 表 ACU-60 的 Relay1~3
	Relay0	Relay1	Relay2	Relay3																				
模式 0	Door	Alarm	響鈴	門鈴																				
模式 1	X	Door	Alarm	門鈴																				
模式 2	X	Door	Alarm	響鈴																				
14	0050H	脅迫碼(不用的位數補 FFH)	'1190'																					
1	005EH	Alarm Relay 重複觸發模式 Bit0~5 0 = Alarm Relay 動作後不重複觸 發 1~63=Alarm Relay 動作幾分鐘後 關閉 Bit6~7 = 保留	00H	Alarm Relay Latch 模式才 有作用																				
1	005FH	Alarm Relay 重複觸發模式關閉 Relay 的時間及重複觸發次數 Bit0~4 1~31= Relay 關閉 1~31 分鐘後再 次觸發	00H	Alarm Relay Latch 模式才 有作用																				

		Bit5~7 1~7 = 重複觸發 1~7 次		
2	0060H	Relay0 狀態設定 (Door) Bit0~11 = Pulse 時間 0=參考其他模式 1~4095=0.1~409.5 秒 Bit12~14 = 保留 Bit15 = 其他模式 (Pulse 時間為 0) 0=Toggle 1=Latch (無此模式)	28H,00H	Lo Byte,Hi Byte 0028H=40 40*0.1 秒=4 秒
2	0062H	Relay1 狀態設定 (Alarm) Bit0~11 = Pulse 時間 0=參考其他模式 1~4095=0.1~409.5 秒 Bit12~14 = 保留 Bit15 = 其他模式 (Pulse 時間為 0) 0=Toggle 1=Latch	00H,80H	Lo Byte,Hi Byte 0028H=40 40*0.1 秒=4 秒
2	0064H	保留 (響鈴無法更改動作模式)		
2	0066H	Relay3 狀態設定 (門鈴) Bit0~11 = Pulse 時間 0=參考其他模式 1~4095=0.1~409.5 秒 Bit12~14 = 保留 Bit15 = 其他模式 (Pulse 時間為 0) 0=Toggle (無此模式) 1=Latch (無此模式)	28H,00H	Lo Byte,Hi Byte 0028H=40 40*0.1 秒=4 秒
	0068H	0068H~00FFH 保留		

3.7.1 時段表 (位址 0100H~047FH : 共 896Bytes)

128*7	0100H	時段表 128 組*7Byte		
-------	-------	-----------------	--	--

Byte1 : 星期一對應到的時間表 (00H~7FH 128 組)

Byte2 : 星期二對應到的時間表 (00H~7FH 128 組)

.....

Byte7 : 星期日對應到的時間表 (00H~7FH 128 組)

不使用請填入 FFH。

3.7.2 時間表 (位址 0480H~067FH : 共 512Bytes)

128*4	0480H	時間表 128 組*4Byte		
-------	-------	-----------------	--	--

起始時間 : 2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

13 : 45 => 13H,45H (BCD 格式)

結束時間 : 2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

20 : 12 => 20H,12H (BCD 格式)

不使用請填入 FFH。

3.7.3 假日表 (位址 0680H~08D7H : 共 600Bytes)

200*3	0680H	假日表 200 組*3Byte		今年 0~99 組 明年 100~199 組
-------	-------	-----------------	--	------------------------------

假日日期 : 2 byte,第一個 byte 為月,第二個 byte 為日,例如

12 月 25 日 => 12H,25H (BCD 格式)

對應到的時間表 : 1 byte 例如

第二組時間表 => 01H (00H~7FH 128 組)

不使用請填入 FFH。

3.7.4 持續開門時段 (位址 08D8H~08DFH : 共 8Bytes)

8*1	08D8H	持續開門時段 8 組		
-----	-------	------------	--	--

不使用請填入 FFH。

3.7.5 響鈴時間表 (位址 08E0H~097FH : 共 160Bytes)

32*5	08E0H	響鈴時間表 32 組*5Byte		
------	-------	------------------	--	--

起始時間 : 2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

08 : 30 => 08H,30H (BCD 格式)

動作時間 : 2 byte,第一個 byte 為分鐘,第二個 byte 為秒,例如

00 : 12 => 00H,12H (BCD 格式)

動作星期 : 1 byte

bit7 : 保留,bit6~0 : 日六五四三二一

不使用請填入 FFH。

3.7.6 Anti Reset 時間表 (位址 0980H~099FH : 共 32Bytes)

16*2	0980H	定時清除 Anti 標記,16 組* 2Byte		
------	-------	--------------------------	--	--

時間 : 2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

23 : 50 => 23H,50H (BCD 格式)

不使用請填入 FFH。

3.7.7 班別時間表 (位址 09A0H~09FFH : 共 96Bytes)

32*3	09A0H	32 組* 3Byte		
------	-------	-------------	--	--

組別	時間(2byte)	班別(1byte)
第 1 組位址 0090H	HH : MM	Class
第 2 組位址 0093H	HH : MM	Class
第 3 組位址 0096H	HH : MM	Class
.	.	.
.	.	.
.	.	.
第 32 組位址 00EDH	HH : MM	Class

時間 : 2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

08 : 10 => 08H,10H (BCD 格式)

班別 : 1 byte

0=空白	1=DUTY ON
2=DUTY_OFF	3=BREAK OUT
4=BREAK_IN	5=OT_START
6=OT_END	

不使用請填入 FFH。

3.7.8 顯示訊息 (位址 0A00H~0A7FH : 共 128Bytes)

8*16	0A00H	訊息 8 組* 16Byte		
------	-------	----------------	--	--

組別	訊息內容	LCD 顯示長度	說明
0	HUNDURE	10	Ready 顯示

1	DUTY ON	10	上班
2	DUTY OFF	10	下班
3	BREAK OUT	10	外出
4	BREAK IN	10	返回
5	OT_START	10	加班開始
6	OT_END	10	加班結束
7		10	空白

每個訊息預留 16 Byte,目前只會顯示前 10 Byte。

不使用請填入 20H。

3.7.9 保留區 (位址 0A80H~1FFFH)

	0A80H	0A80H-1FFFH 空白		
--	-------	----------------	--	--

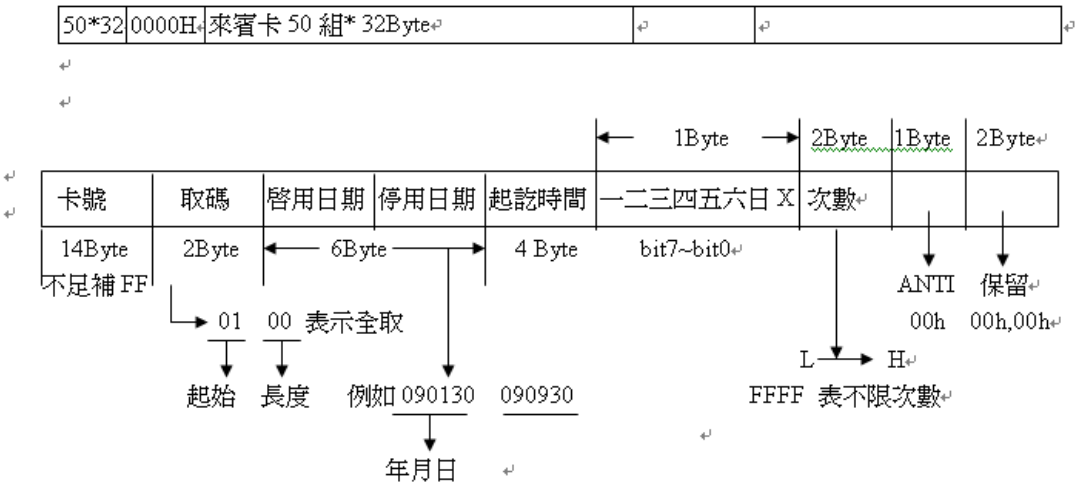
3.7.10 合法卡區 (位址 002000H~027FFFH)

(10240 筆 * 16Byte)/4KB 大約佔 40 Sector。

3.7.11 刷卡資料紀錄區 (位址 028000H~07FFFFH)

3.7.12SRAM 相關記憶體規劃 (R/W Table 15,16)

• 來賓卡 50 筆 (位址 0000H~063FH:共 1600Bytes)



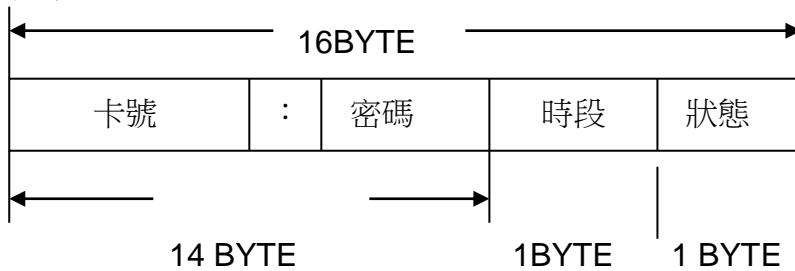
3.7.13 事件紀錄 200 筆 (位址 0640H~0C7FH : 共 1600Bytes)

200*8	0640H	事件紀錄 200 組* 8Byte		
-------	-------	-------------------	--	--

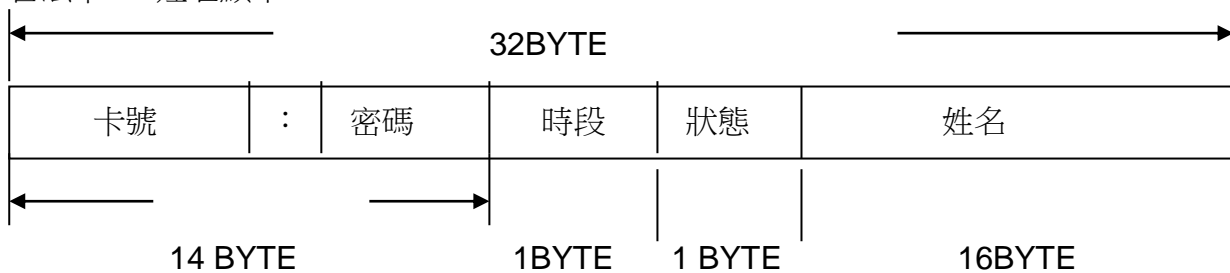
附錄 4：RAC-940 格式說明

4.1 卡號權限格式

標準模式：



合法卡 + 姓名顯示：



- 卡號與密碼：(14 byte)
- 卡號為 ASCII 碼。
- 密碼內容固定為壓縮格式, 2 個數字為 1 個 byte, 不足位或沒有密碼補 FFH
- 卡號與密碼用' ':'[冒號] 分隔。
- 時段：0~15 索引至時段表一。128~143 索引至時段表二。
 1. 時段表一及時段表二 的差別在於時段表一只能對應一組時間,而時段表二可以對應最多八組時間。
 2. 當權限會使用到一天多個時間組時,建議使用時段表二, 效率比較好也不佔合法卡空間。
- 狀態：
 - Bit 0：0—正常卡
1—黑名單
 - Bit 1：保留
 - Bit 2：0—假日不檢查
1—假日檢查

Bit 3：0－時段不檢查

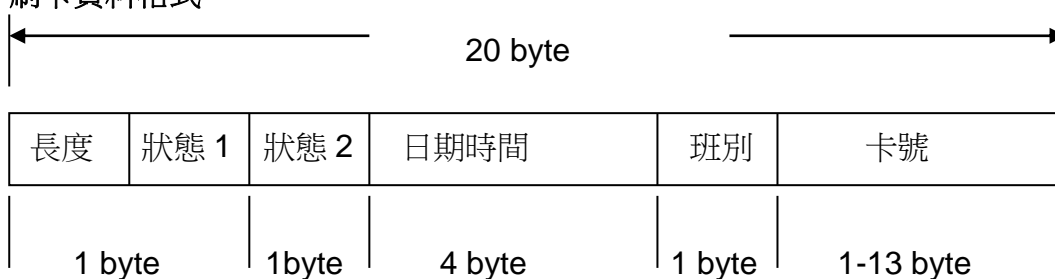
1－時段檢查

Bit 4~7：保留

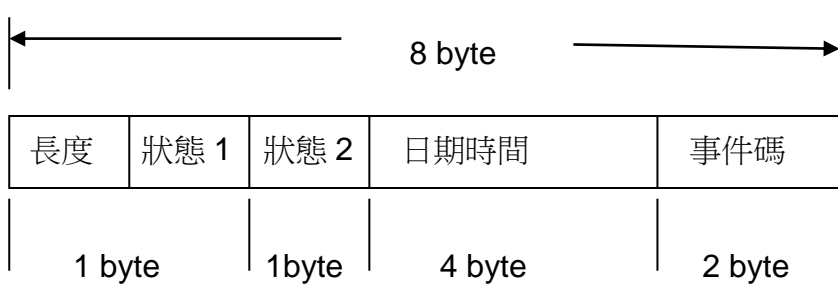
- 姓名：不顯示的部分請填入 20H。

4.2 刷卡及事件資料格式

刷卡資料格式



事件資料格式



長度：bit 7~4 說明卡號長度

狀態 1：

Bit0	0=按鍵輸入卡號 1=刷卡
Bit1	0=內 Reader 1=外 Reader
Bit2	保留
Bit3	0=刷卡紀錄 1=事件紀錄

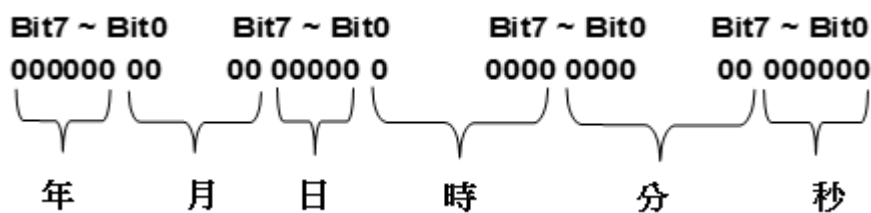
狀態 2：

Bit0~7	0 = 正常刷卡
--------	----------

1 =	母卡
2 =	解除碼
3 =	脅迫碼
6 =	黑名單
20 =	沒有找到卡號
21 =	卡號時段錯誤
61 =	刷卡正確+密碼正確
62 =	刷卡正確+密碼不正確
63 =	刷卡+密碼時,刷卡是用按鍵輸入
67 =	Anti 錯誤
75 =	比較有效位數設定的長度錯誤

日期時間：共 4Byte(傳輸時 Lo Byte 在前),0~5bit 代表秒,6~11bit 代表分鐘,12~16 代表小時,17~2 代表日期,22~25 代表月份,26~31 代表年份(2000+N)。

時間(4byte) Byte1,Byte2,Byte3,Byte4



班別： 參數區的主機模式設定

0=空白	1=DUTY ON
2=DUTY_OFF	3=BREAK OUT
4=BREAK_IN	5=OT_START
6=OT_END	

事件碼格式(Sensor/Relay)

- 0120H 脅迫碼通過
- 0122H 解除碼解除警報
- 0123H Master Card (Code) 修改
- 0301H 偵測外力破壞(Tamper) → Case Sensor
- 0306H 冷開機(Cold Startup)
- 0307H 重新啟動(Reset)
- 0311H 門被外力開啟 → Door Sensor 門被破壞

0312H 門持續開啟警告

0313H 門持續開啟警告解除

0318H 開門(Door Opened) → 一般合法開門

0319H 關門(Door Closed) → Door Sensor Close

031AH 開門鈕觸動(Button Pressed) → Push Button

041BH 系統初始化

4.3 系統參數表 (位址 0000H~0013H : 共 20Bytes)

長度	位址	說明	預設值	備註
1	0000H	通訊方式 Bit0~6 : 鮑率 1=9600 2=19200 3=38400 4=115200 Bit7 : 保留	02H	
1	0001H	保留		
4	0002H	保留		
4	0006H	保留		
4	000AH	保留		
2	000EH	保留		
2	0010H	合法卡最大筆數包含黑名單,最多 5000 筆	88H,13H	Lo Byte,Hi Byte 1388H = 5000 筆
1	0012H	機器的 ID	01H	
1	0013H	主機模式 0=標準模式 (同 RAC-2000) 1=合法卡 + 姓名顯示	00H	

4.4 一般參數表 (位址 0014H~00FFH : 共 236Bytes)

長度	位址	說明	預設值	備註
1	0014H	主機狀態設定 Bit0 : 副刷卡槽不檢查時段 0=不檢查時段 1=檢查時段 Bit1 : 室內/室外模式選擇 0=室內 1=室外 Bit2 : 黑名單是否要啟動 Alarm Realy 0=Disable 1=Enable Bit3~4 : 保留 Bit5 : 非法卡是否儲存刷卡紀錄 0=儲存 1=不儲存 Bit6 : 儲存資料不可覆蓋 0=覆蓋 1=不覆蓋且儲存資料達 90%以上警告 Bit7 : 儲存資料滿時起動 Alarm Relay 0=Disable 1=Enable	00H	
1	0015H	假日表的年份	09H	
10	0016H	保留		
1	0020H	LCD 日期顯示格式 0=YYYY/MM/DD 1=MM/DD/YYYY 2=DD/MM/YYYY	00H	
1	0021H	LCD 狀態設定 Bit0~3 = 保留 Bit4~5 = LCD 背光模式 (0=Auto,1=開啟,2=關閉) Bit6~7 = 保留	10H	初始值 LCD 背光模式開啟
1	0022H	LCD 回復 Ready 時間	00H	

		0=內定值 8 秒 1~255=1~255 秒		
1	0023H	按鍵狀態設定 Bit0 = 按鍵狀態 (0=關閉,1=開啟) Bit1~3 = 保留 Bit4~5 = 按鍵背光模式 (0=Auto,1=開啟,2=關閉) Bit6~7 = 保留	01H	初始值 按鍵狀態開啟 按鍵背光模式 Auto
2	0024H	取碼有效位數	01H, 00H	
2	0026H	比較有效位數	00H, 00H	
1	0028H	0=Anti Disable 1=Anti Enable	00H	
1	0029H	語系選擇 0 = English 1 = 繁體 2 = 簡體	00H	
1	002AH	Sensor 動作狀態 (1=Close,0=Open) Bit0 =保留 Bit1 = 0 (Door Sensor) Bit2 = 1 (Push Button) Bit3~7 = 保留	04H	
1	002BH	Case Sensor 是否要啟動 Alarm Relay Bit0 = Buzzer 警報 Bit1 = Alarm Relay 動作 Bit2~7 = 保留	01H	
1	002CH	設定 Door Sensor 偵測的秒數 0=關閉 Door Sensor 1~255=偵測秒數 1~255 秒	00H	
1	002DH	Door Sensor 是否要啟動警報 Bit0 = Buzzer 警報 Bit1 = Alarm Relay 動作 Bit2~7 = 保留	01H	
1	002EH	設定錯誤次數 0=關閉 1~255=錯誤次數 1~255 次	00H	

1	002FH	達到錯誤次數時系統要停止操作的秒數 0=關閉 1~255=1~255 秒	00H																					
14	0030H	母卡(不用的位數補 FFH)	'30191000'																					
1	003EH	保留																						
1	003FH	保留																						
14	0040H	解除碼(不用的位數補 FFH)	'0000'																					
1	004EH	保留																						
1	004FH	Relay 動作模式選擇 <table border="1"> <thead> <tr> <th></th><th>Relay0</th><th>Relay1</th><th>Relay2</th><th>Relay3</th></tr> </thead> <tbody> <tr> <td>模式 0</td><td>Door</td><td>Alarm</td><td>響鈴</td><td>門鈴</td></tr> <tr> <td>模式 1</td><td>X</td><td>Door</td><td>Alarm</td><td>門鈴</td></tr> <tr> <td>模式 2</td><td>X</td><td>Door</td><td>Alarm</td><td>響鈴</td></tr> </tbody> </table>		Relay0	Relay1	Relay2	Relay3	模式 0	Door	Alarm	響鈴	門鈴	模式 1	X	Door	Alarm	門鈴	模式 2	X	Door	Alarm	響鈴	00H	Relay0 代表主機上的 Relay Relay1~3 代表 ACU-60 的 Relay1~3
	Relay0	Relay1	Relay2	Relay3																				
模式 0	Door	Alarm	響鈴	門鈴																				
模式 1	X	Door	Alarm	門鈴																				
模式 2	X	Door	Alarm	響鈴																				
14	0050H	脅迫碼(不用的位數補 FFH)	'1190'																					
1	005EH	Alarm Relay 重複觸發模式 Bit0~5 0 = Alarm Relay 動作後不重複觸發 1~63=Alarm Relay 動作幾分鐘後關閉 Bit6~7 = 保留	00H	Alarm Relay Latch 模式才有作用																				
1	005FH	Alarm Relay 重複觸發模式關閉 Relay 的時間及重複觸發次數 Bit0~4 1~31= Relay 關閉 1~31 分鐘後再次觸發 Bit5~7 1~7 = 重複觸發 1~7 次	00H	Alarm Relay Latch 模式才有作用																				
2	0060H	Relay0 狀態設定 (Door) Bit0~11 = Pulse 時間 0=參考其他模式 1~4095=0.1~409.5 秒 Bit12~14 = 保留 Bit15 = 其他模式 (Pulse 時間為 0) 0=Toggle	28H,00H	Lo Byte,Hi Byte 0028H=40 40*0.1 秒=4 秒																				

		1=Latch (無此模式)		
2	0062H	Relay1 狀態設定 (Alarm) Bit0~11 = Pulse 時間 0=參考其他模式 1~4095=0.1~409.5 秒 Bit12~14 = 保留 Bit15 = 其他模式 (Pulse 時間為 0) 0=Toggle 1=Latch	00H,80H	Lo Byte,Hi Byte 0028H=40 40*0.1 秒=4 秒
2	0064H	保留		
2	0066H	Relay3 狀態設定 (門鈴) Bit0~11 = Pulse 時間 0=參考其他模式 1~4095=0.1~409.5 秒 Bit12~14 = 保留 Bit15 = 其他模式 (Pulse 時間為 0) 0=Toggle (無此模式) 1=Latch (無此模式)	28H,00H	Lo Byte,Hi Byte 0028H=40 40*0.1 秒=4 秒
1	0068H	黑名單最大筆數 0~255	64H	
	0069H	0069H~00FFH 保留		

4.5 時段表一 (位址 0100~016F：共 112 Bytes,16 組)

		星期					
索引		一	二	三	四	五	六 日
0	時段索引						
1							
.							
.							
15							

時段索引：

0~15 索引至時間表一

255 未使用

4.6 時間表一 (位址 0480H~04BFH：共 64Bytes ,16 組)

索引	2 byte	2 byte
0	開始時間 HHMM	結束時間 HHMM
1		
.	.	
.	.	
15		

- 開始時間：兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如 13:25 => 13H,25H (BCD 格式)
- 結束時間：兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如 17:01 => 17H,01H (BCD 格式)

4.7 假日表 (位址 0680H~07FFH：共 384Bytes,64*2 組)

第 1 個假日(3 Byte)					第 64 個假日		
今年	月	日	時間表索引					
明年								

時段索引：

0~15 索引至時間表一
128~143 索引至時間表二

註明：1.月,日為 BCD 格式例如 12 月 1 日 =>12H,01H。

2.在月,日填入 FFH,FFH 表示結束。

4.8 持續開門時段 (位址 08D8H~08DFH：共 8Bytes, 8 組)

1	2	3	4	5	6	7	8
時段索引							

時段索引：

0~15 索引至時段表一
128~143 索引至時段表二
255 結束

4.9 響鈴時間表 (位址 08E0H~0907H：共 40Bytes, 8 組)

	2 Byte	2 Byte	1 Byte
	開始時間 HHMM	動作時間 MMSS	動作星期
0			
1			
.			
.	.		
.	.		
7			

- 起始時間：2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如 08:30 => 08H,30H (BCD 格式)
- 動作時間：2 byte,第一個 byte 為分鐘,第二個 byte 為秒,例如 00:12 => 00H,12H (BCD 格式)
- 動作星期：1 byte
bit7：保留

bit6~0： 日六五四三二一

結束請填入 FFH。

4.10 Anti Reset 時間表 (位址 0980H~098FH：共 16Bytes, 8 組)

	2 Byte
0	起動時間 HHMM
1	
•	•
•	•
6	
7	

起動時間：2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

23：50 => 23H,50H (BCD 格式)

結束請填入 FFH。

4.11 顯示訊息 (位址 0A00H~0A7FH：共 128Bytes, 8 組)

	16 Byte
0	顯示訊息
1	
•	•
•	•
6	
7	

組別	訊息內容	LCD 顯示長度	說明
0	HUNDURE	10	Ready 顯示
1	DUTY ON	10	上班
2	DUTY OFF	10	下班
3	BREAK OUT	10	外出
4	BREAK IN	10	返回
5	OT_START	10	加班開始
6	OT_END	10	加班結束
7		10	空白

每個訊息預留 16 Byte,目前只會顯示前 10 Byte。

不使用請填入 20H。

4.12 密碼時段 (位址 0B00H~0B07H : 共 8Bytes, 8 組)

1	2	3	4	5	6	7	8
時段索引							

時段索引：

0~15 索引至時段表一

128~143 索引至時段表二

255 結束

4.13 警報時段 (位址 0C00H~0C03H : 共 4Bytes, 1 組)

2 byte	2 byte
開始時間 HHMM	結束時間 HHMM

開始時間：兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

13 : 25 => 13H,25H (BCD 格式)

結束時間：兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

17 : 01 => 17H,01H (BCD 格式)

4.14 時段表二 (位址 0C04~0C73 : 共 112 Bytes,16 組)

索引	星期						
	一	二	三	四	五	六	日
128	時段索引						
129							
.							
.							
143							

時段索引：

0~15 索引至時間表二

255 未使用

4.15 時間表二 (位址 0CE4~0EE3：共 512Bytes,16 組)

索引	第一組			第八組	
	開始時間 HHMM	結束時間 HHMM				
0						
2						
.	.					
.	.					
15						

開始時間：兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

13：25 => 13H,25H (BCD 格式)

結束時間：兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

17：01 => 17H,01H (BCD 格式)

註明：在一天中最多可以對應到八個時間組,在”開始時間”填入 0xFF,0xFF 表示結束。

5.1 基本格式

系統 BYTE 表示：

--	--	--	--	--	--	--	--

	Outside Reader ID	Inside Reader ID
Door 1	0	16
Door 2	1	17
Door 3	2	18
Door 4	3	19

Diagram illustrating the structure of the 16-byte data field:

16BYTE				
卡號	:	密碼	時段	狀態
14 BYTE		1BYTE	1 BYTE	

0：上次刷卡是 IN

1：上次刷卡是 OUT

bit6~7：

0：第一個門

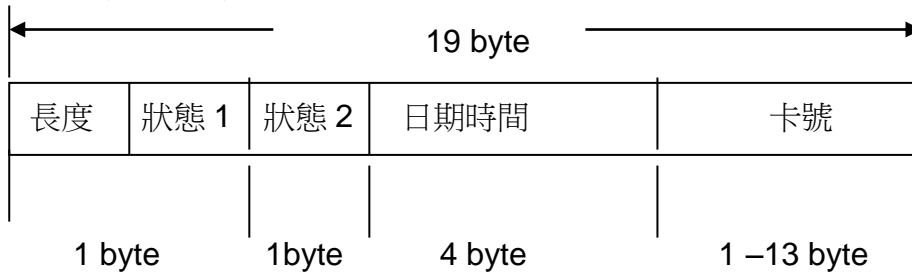
1：第二個門

2：第三個門

3：第四個門

- 密碼：不足位或沒有密碼補 FF
- 密碼內容固定為壓縮格式, 2 個數字為 1 個 byte
- 卡號與密碼用 ':' [冒號] 分隔

5.3 刷卡及事件資料格式



長度：4bit 說明卡號長度

狀態 1：4bit

bit0~2 = 000 reader 1

001 reader 2

010 reader 3

011 reader 4

100 reader 5

101 reader 6

110 reader 7

111 reader 8

bit3 = 0 卡號

= 1 Event (在第 7 與第 8 個 byte)

卡號依長度來存資料

狀態 2：0 = 正常刷卡

1 = Master Card

2 = Release Code

3 = 脅迫卡

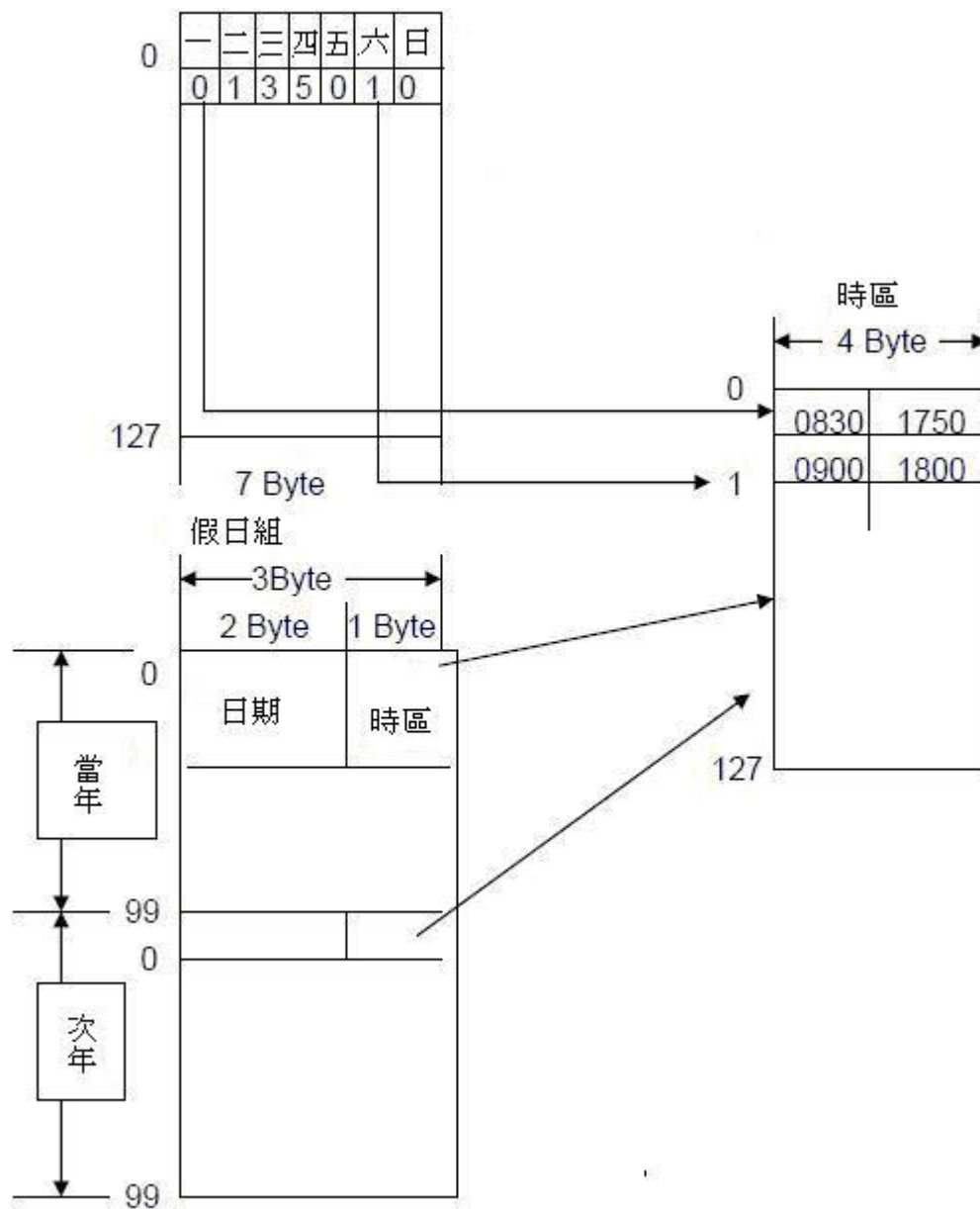
5 =臨時卡
6 =黑名單
10 =來賓卡
11 =來賓卡(不限次數)
20 =沒有找到卡號
21 =卡號時段錯誤
61=刷卡正確+密碼正確
62=刷卡正確+密碼不正確
63=刷卡+密碼時,刷卡是用按鍵輸入
67=Anti 錯誤

日期時間：共 4Byte(傳輸時 Lo Byte 在前),0~5bit 代表秒,6~11bit 代表分鐘,12~16 代表小時,17~21 代表日期,22~25 代表月份,26~31 代表年份(2000+N)。

次數：刷卡時儲存卡片上的次數(FFFFH 不限定次數) (Hi,Lo)。

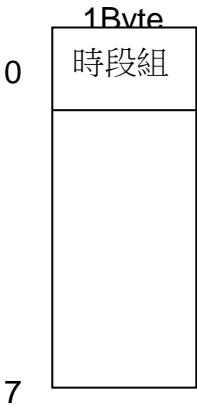
5.4 時段表對照說明

時段：

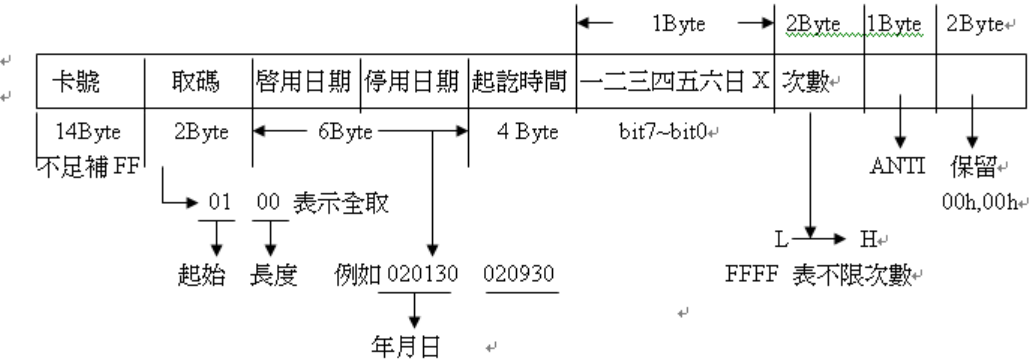


-不使用的時段組，假日組，時區組填 FF

持續開門： 8 組



➤ 來賓卡 50 筆 (50*32Bytes)



5.5 RAC2200 ANTI (下載此 Byte 時請填入 00h)

AntiMode = 1 By Door Acivate Door1 Relay IN/OUT Reader ID 0/16/2/18 for IN, Reader ID 1/17/3/19 for OUT AntiMode = 3 By Door Door1 Relay IN, Door2 Relay OUT Reader ID 0/16/2/18 for IN Reader ID 1/17/3/19 for OUT	AntiMode = 2 By Reader Door1 Relay IN/OUT Door1 Relay Reader ID 0 for IN,Reader ID 16 for OUT Door2 Relay Reader ID 1 for IN,Reader ID 17 for OUT Door3 Relay Reader ID 2 for IN,Reader ID 18 for OUT Door4 Relay Reader ID 3 for IN,Reader ID 19 for OUT
bit4 : 0 : Anti 啟動的初值 1 : 此卡號已經有使用過	bit4~7 : 0 : Anti 啟動的初值 1 : 此卡號已經有使用過 bit4 : Door1 bit5 : Door2 bit6 : Door3 bit7 : Door4
bit5 : 0 : 上次刷卡是 IN 1 : 上次刷卡是 OUT	bit0~3 : 0 : 上次刷卡是 IN 1 : 上次刷卡是 OUT bit0 : Door1 bit1 : Door2 bit2 : Door3 bit3 : Door4

5.6 事件碼回傳值(Sensor/Relay)

0120 H 脅迫碼通過
0122 H 解除碼解除警報
0123 H Master Card (Code) 修改
0306 H 冷開機(Cold Startup)
0307 H 重新啟動(Reset)
0401 H 記憶體已滿(Memory Full)
041B H 系統初始化

5.7 EEPROM 相關記憶體規劃

長度	位址	說明	預設值	備註
2	0000H	EEPROM Test	5AH,A5H	開機測試用
1	0002H	Node ID	1	
1	0003H	假日表的年份	08H	
14	0004H	Master Card(不用的位數補 FFH)	'30191000'	
14	0012H	脅迫碼(不用的位數補 FFH)	1190	
14	0020H	解除碼(不用的位數補 FFH)	'0000'	
1	002EH	設定錯誤次數	0	
1	002FH	達到錯誤次數時系統要停止操作的秒數	0	
2	0030H	合法卡號在 SRAM 的筆數	0	
2	0032H	合法卡號的最大筆數	10000	
1	0034H	Door1 動作時間	04H	
1	0035H	Door2 動作時間	04H	
1	0036H	Door3 動作時間	04H	
1	0037H	Door4 動作時間	04H	
1	0038H	Door1 Door Sensor 偵測時間 0=關閉 1~255=1~255 秒	00H	
1	0039H	Door2 Door Sensor 偵測時間 0=關閉 1~255=1~255 秒	00H	
1	003AH	Door3 Door Sensor 偵測時間 0=關閉 1~255=1~255 秒	00H	
1	003BH	Door4 Door Sensor 偵測時間 0=關閉 1~255=1~255 秒	00H	
1	003CH	Door1 重複刷卡檢查的秒數 0=關閉 1~255=1~255 秒	00H	
1	003DH	Door2 重複刷卡檢查的秒數 0=關閉	00H	

		1~255=1~255 秒		
1	003EH	Door3 重複刷卡檢查的秒數 0=關閉 1~255=1~255 秒	00H	
1	003FH	Door4 重複刷卡檢查的秒數 0=關閉 1~255=1~255 秒	00H	
8	0040H	Door1 持續開門時段 8 組	FFH	
8	0048H	Door2 持續開門時段 8 組	FFH	
8	0050H	Door3 持續開門時段 8 組	FFH	
8	0058H	Door4 持續開門時段 8 組	FFH	
1	0060H	0=無 Anti 1=Anti By Door 只動作 Door1 Relay IN/OUT Reader1-2 IN,Reader3-4 OUT Reader5-6 IN,Reader7-8 OUT 2=Anti By Reader Door1 Relay IN/OUT Reader1 IN,Reader2 OUT Door2 Relay IN/OUT Reader3 IN,Reader4 OUT Door3 Relay IN/OUT Reader5 IN,Reader6 OUT Door4 Relay IN/OUT Reader7 IN,Reader8 OUT 3=Anti By Door Door1 Relay IN, Door2 Relay OUT Reader1-2 IN,Reader3-4 OUT Reader5-6 IN,Reader7-8 OUT	0	
1	0061H	黑名單是否要啟動 Realy bit0 : 0 不啟動,1 啟動	01H	
1	0062H	bit0=1 儲存資料達 90%警告 bit1=1 儲存資料不可覆蓋	0	
1	0063H	空白		

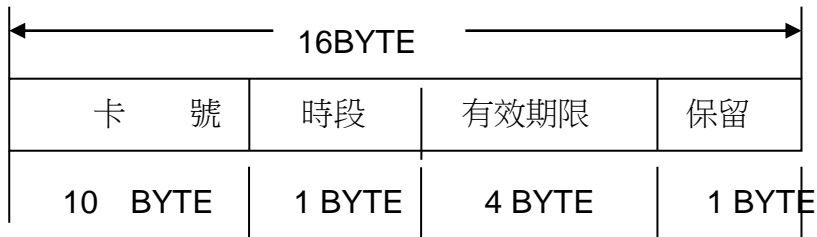
1	0064H	Door1 Reader Keypad 狀態設定 0=關閉 1=開啟	01H	
1	0065H	Door2 Reader Keypad 狀態設定 0=關閉 1=開啟	01H	
1	0066H	Door3 Reader Keypad 狀態設定 0=關閉 1=開啟	01H	
1	0067H	Door4 Reader Keypad 狀態設定 0=關閉 1=開啟	01H	
152	0068H	空白		
256*7	0100H	時段表 128 組*7Byte		
256*4	0800H	時間表 128 組*4Byte		
100*3	0C00H	今年假日表 100 組*3Byte		
100*3	0D2CH	明年假日表 100 組*3Byte		
	0E58H	0E58H-1FFFFH 空白		

5.8 SRAM 相關記憶體規劃

位址範圍	長度	說明	備註
000000H-00001FH	32	資料指標位址	
000020H-00065FH	1600	來賓卡位址(50 筆*32Byte)	
000660H-000C9FH	1600	儲存 Event 位址(200 筆*8Byte)	
000CA0H-027D9FH	10000	合法卡位址(10000 筆*16Byte)	
027DA0H-03FFFFH		刷卡紀錄位址	

附錄 6：ECU-680 格式說明

6.1 合法卡格式



1. 卡號：(10 byte)

卡號為 ASCII 碼,不足位 00H

2. 時段：(1 byte)

Bit 0~2:

0 ~ 7 時段表索引

Bit 3:

0 — 時段管制

1 — 時段不管制

Bit 4:

0 — 通訊方式加卡

1 — 設定卡加卡

Bit 5~7:保留

3. 有效期限：Byte 1,2,3,4 (4 byte)

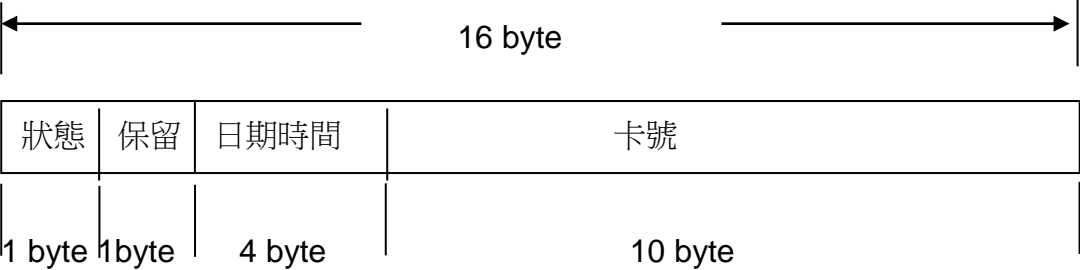
bit 7....bit0	bit 7....bit0	bit 7....bit0	bit 7....bit0
00000 000	0000 0000	00000 000	00 000000
└─┘	└─┘└─┘└─┘	└─┘└─┘└─┘	└─┘
啟用	年	月	日 時 分

啟用:00000 表示啟用有效期限,其他值表示不使用有效期限

4. 保留：(1 BYTE)保留

6.2 刷卡記錄資料格式

刷卡資料格式

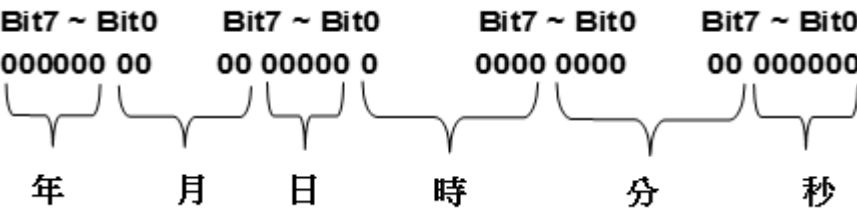


狀態:

Bit0	0=抽出卡片 1=插入卡片
Bit1	0=非法卡 1=合法卡
Bit2~6	保留
Bit7	0=已讀取 1=未讀取

日期時間:共 4Byte(傳輸時 Lo Byte 在前),0~5bit 代表秒,6~11bit 代表分鐘,12~16 代表小時,17~21 代表日期,22~25 代表月份,26~31 代表年份(2000+N)。

時間(4byte) Byte1,Byte2,Byte3,Byte4



6.3 參數表 (位址 0000H~001FH:共 32Bytes)

長度	位址	說明	預設值	備註
2	0000H	延遲電源關閉時間	0F,00H	單位秒
1	0002H	電源關閉提示聲	00H	單位秒
1	0003H	模式設定 Bit 0 = 刷卡聲 0 — 關閉 1 — 開啟 Bit 1 =不合法刷卡記錄 0 — 不保留 1 — 保留 Bit 2 =Sensor 1 0 — 關閉 1 — 開啟 Bit 3 =Sensor 2 0 — 關閉 1 — 開啟 Bit4~7 = 保留	00	
2	0004H	擴充 Relay 功能: 0000H — 模式同電源 Relay 0001H~FFFFH — Pulse 模式 0.01~655.35 秒	90H,01H	單位 0.01 秒
1	0006H	Sensor1 秒數	0	單位秒
1	0007H	Sensor2 秒數	0	單位秒
4	0008H	Mifare 模組參數	00H,1AH,00H,01H	
20	000CH	保留	FF~	

6.4 時段表 (位址 0020~0057H:共 56 Bytes,8 組)

星期		一	二	三	四	五	六	日
索引	0	時段索引						
	1							
	·							
	·							
	7							

時段索引:

- 0~7 索引至時間表
- 255 未使用

6.5 時間表 (位址 0058H~00D7H:共 128Bytes,8 組)

索引	第一組			第四組	
	開始時間 HHMM	結束時間 HHMM				
0						
1						
.	.					
.	.					
7						

開始時間:兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘例如

13:25 => 13H,25H (BCD 格式)

結束時間:兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘例如

17:01 => 17H,01H (BCD 格式)

註明:在一天中最多可以對應到四個時間組,在"開始時間"填入 0xFF,0xFF 表示結束。

6.6 不管制時段 (位址 00D8H~00DBH:共 4Bytes, 4 組)

1	2	3	4
時段索引			

時段索引:

0~7 索引至時段表
255 結束

6.7 設定卡 (位址 00DCH~0117H:共 60Bytes)

0	主設定卡(10 byte)
1	設定卡(10 byte)
2	設定卡(10 byte)
3	設定卡(10 byte)
4	設定卡(10 byte)
5	設定卡(10 byte)

一張主設定卡,五張設定卡,刪除時補滿 0

附錄 7：RAC-820PEF/RAC-820PMF 格式說明

7.1 卡號權限格式

卡號	:	密碼	時段	狀態	姓名	系統保留	
←	14	Bytes	→	1 Byte	1 Byte	14 Bytes	2 Bytes

-卡號 ASCII

-密碼:不足位或沒有密碼補 0xFF

-密碼內容固定為壓縮格式, 2 個數字為 1 個 byte

-卡號與密碼用 ':' [冒號] 分隔

-時段 128 組 (0~127)

820Px F 合法卡只能有一張

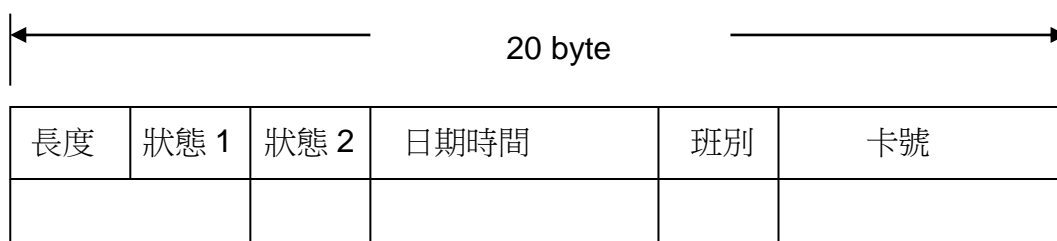
-狀態說明:

Bit0	0=正常卡 1=黑名單
Bit1	0=使用者不需按指紋 1=使用者需按指紋
Bit2	0=假日不檢查 1=假日檢查
Bit3	0=時段不檢查 1=時段檢查
Bit4	保留
Bit5	保留
Bit6	820Px F 0:雙重比對 B 群 1:雙重比對 A 群
Bit7	0:不播放語音 1:播放語音

姓名：因 RAC-820Px F 不支援顯示功能，若無姓名資料請填入 0x20，共 14 Bytes

系統保留: 2 Bytes

7.2 刷卡歷史資料



1 byte

1byte

4 byte

1 byte

1-13 byte

- 長度：4bit 說明卡號長度

- 狀態 1：

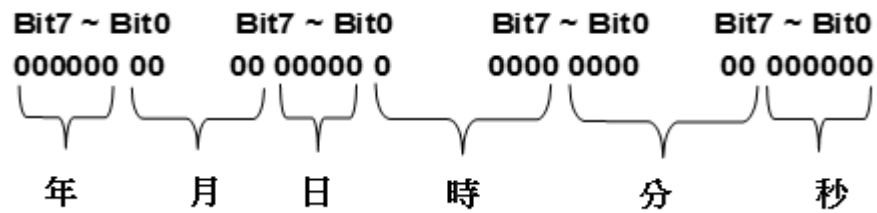
Bit0	0=按鍵輸入卡號 1=刷卡
Bit1	0=主機內部讀頭 1=外部讀頭
Bit2	0=一般卡號 1=DesFire 序號 (7Byte/16 進制碼)
Bit3	0=刷卡紀錄 1=事件紀錄

- 狀態 2：

Bit0~7	0 = 正常刷卡 1 = 母卡 2 = 解除碼 3 = 脅迫碼 5 = 臨時卡 6 = 黑名單 8 = 指紋比對失敗 9 = 指紋比對通過 10 = 來賓卡 11 = 來賓卡(不限次數) 20 = 沒有找到卡號 21 = 卡號時段錯誤 30 = 密碼開門 31 = 操作錯誤 61 = 刷卡正確+密碼正確 62 = 刷卡正確+密碼不正確 63 = 刷卡+密碼時,刷卡是用按鍵輸入 67 = Anti 錯誤 75 = 比較有效位數設定的長度錯誤 76 = 雙重比對錯誤
--------	--

- 日期時間：共 4Byte(傳輸時 Lo Byte 在前),0~5bit 代表秒,6~11bit 代表分鐘,12~16 代表小時,17~21 代表日期,22~25 代表月份,26~31 代表年份(2000+N)。

- 時間(4byte) Byte1,Byte2,Byte3,Byte4



- 班別： 參數區的主機模式設定

0=空白	1=DUTY ON
2=DUTY_OFF	3=BREAK OUT
4=BREAK_IN	5=OT_START
6=OT_END	

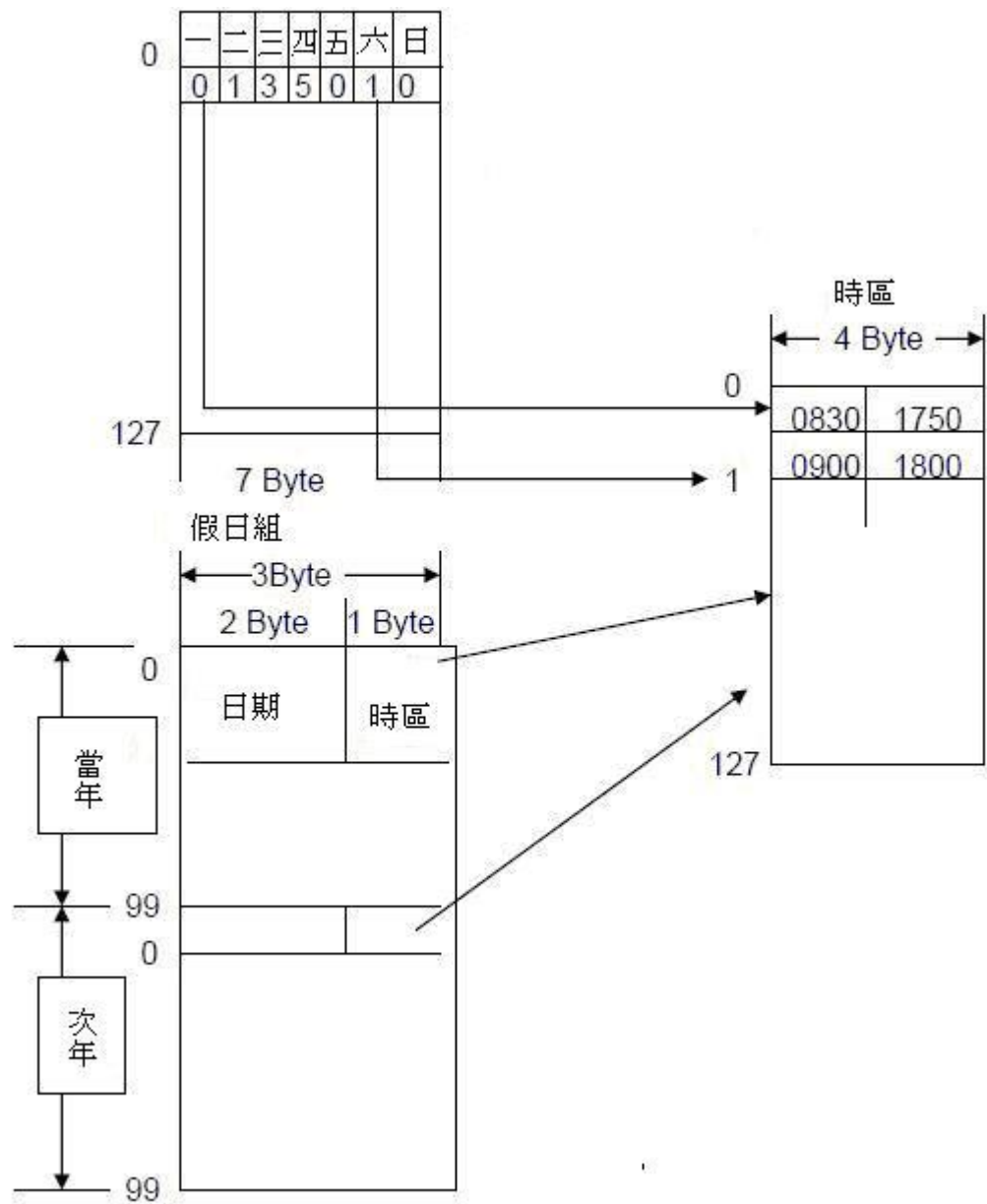
- 卡號依長度來存資料。

Polling 有班別資料請用改用 1FH Command,原本 10H Command 不會有班別資料。

7.3 時段表對照說明

時段/時區/假日

時段：



不使用的時段組，假日組，時區組填 FF

7.4 持續開門



7.5 事件碼回傳值(Sensor/Relay)

0120H 脅迫碼通過

0122H 解除碼解除警報

0123H Master Card (Code)修改

0301H 偵測外力破壞(Tamper) → Case Sensor

0306H 冷開機(Cold Startup)

0307H 重新啟動(Reset)

0311H 門被外力開啟 → Door Sensor 門被破壞

0312H 門持續開啟警告

0313H 門持續開啟警告解除

0318H 開門(Door Opened) → 一般合法開門

0319H 關門(Door Closed) → Door Sensor Close

031AH 開門鈕觸動(Button Pressed) → Push Button

0401H 記憶體已滿(Memory Full)

041BH 系統初始化

7.6 Flash Memory 相關記憶體規畫

(位址 0000H~00FFH：共 256Bytes),0000H~0013H 屬於系統參數

長度	位址	說明	預設值	備註
1	0x0000	通訊方式 Bit7 = 1:TCP/IP Bit0~6:保留	0x80	
1	0x0001	讀頭型態 主機讀卡型態依照硬體與搭配對 應韌體決定 Bit1~3:保留 Bit4~7:副讀頭 0=T1/T2 1=Wiegand26 (8 碼) 2=Wiegand26 (10 碼) 3=Wiegand34 (10 碼) 4~15:保留	0x01	
4	0x0002	IP	172.16.250.100	
4	0x0006	SubMask	255.255.0.0	
4	0x000A	Gateway	0.0.0.0	
2	0x000E	Port	4660	Lo Byte,Hi Byte
2	0x0010	合法卡最大筆數	0x80, 0x46	Lo Byte,Hi Byte 0x4680=18048 筆
1	0x0012	機器的 ID	0x01	
1	0x0013	主機模式 1=合法卡	0x01	
1	0x0014	主機狀態設定 Bit0:外出不檢查時段 0=不檢查時段 1=檢查時段 Bit1:室內/室外模式選擇 0=室外 1=室內 Bit2:黑名單是否要啟動 Alarm Realy	0x00	

		0=Disable (不啟動 Relay) 1=Enable (啟動 Relay) Bit3:副讀頭於刷卡加密模式時,不需 輸入密碼 0=Disable (需按密碼) 1=Enable (不需按密碼) Bit4:卡號顯示 不顯示卡號 Bit5:非法卡是否儲存刷卡紀錄 0=儲存 1=不儲存 Bit6:儲存資料不可覆蓋 0=覆蓋 1=不覆蓋且儲存資料達 90%以上警 告 Bit7:儲存資料滿時起動 Alarm Relay 0=Disable (不啟動 Relay) 1=Enable (啟動 Relay)		
1	0x0015	假日表的年份	0x16	
10	0x0016	Mifare 設定值 (設定 KeyA/KeyB 的值只能用 Write Table 54 設定)	0x00,0x10,0x00,0x00, 0xFF,0xFF,0xFF, 0xFF,0xFF,0xFF	參考 Write Table 54
2	0x0024	取碼有效位數	0x01, 0x00	
2	0x0026	比較有效位數	0x00, 0x00	
1	0x0028	0=Anti Disable 1=Anti Enable	0x00	
1	0x0029	語系選擇 0 = English 1 = 繁體 2 = 簡體	0x00	
1	0x002A	Sensor 動作狀態 (1=Close,0=Open) Bit0 = 0 (Case Sensor) Bit1 = 0 (Door Sensor) Bit2 = 1 (Push Button) Bit3~7 = 保留	0x04	
1	0x002B	Case Sensor 是否要啟動 Alarm Relay Bit0 = Buzzer 警報	0x00	

		Bit1 = Alarm Relay 動作 Bit2~7 = 保留		
1	0x002C	設定 Door Sensor 偵測的秒數 0=關閉 Door Sensor 1~255=偵測秒數 1~255 秒	0x00	
1	0x002D	Door Sensor 是否要啟動警報 Bit0 = Buzzer 警報 Bit1 = Alarm Relay 動作 Bit2~7 = 保留	0x01	
1	0x002E	設定錯誤次數 0=關閉 1~255=錯誤次數 1~255 次	0x00	
1	0x002F	達到錯誤次數時系統要停止操作的秒數 0=關閉 1~255=1~255 秒	0x00	
14	0x0030	母卡(不用的位數補 0xFF)	'30191000'	
1	0x003E	指紋機狀態設定 Bit0 = 指紋機進入 1:1 模式 (0=Auto,1=1:1) Bit1 = Master Card 可用指紋取代 (0=Disable,1=Enable) Bit2 = 雙重比對 (0=Disable,1=Enable) Bit3 = 語音狀態 (0=開啟,1=關閉) Bit4 = 第二組 RS-485 模式 0=Polling 副指紋機 1=DVR Bit5 = 卡號比對通過後不需比對指紋 (0=Disable,1=Enable) Bit6~7 = 保留	0x00	
1	0x004E	主機參數設定 Bit0:LCD 顯示班別時是否自動恢復 0=班別顯示不恢復 1=班別顯示恢復 Bit1:達到錯誤次數時啟動 Alarm Relay 0=Disable (不啟動 Relay) 1=Enable (啟動 Relay)	0x01	

		Bit2:鐵捲門模式 0:門禁模式 1:鐵捲門模式 Bit3~7:保留														
1	0x004F	Relay 動作模式選擇 <table><tr><td></td><td>Relay0</td><td>Relay1</td><td>Relay2</td></tr><tr><td>模式 0</td><td>Door</td><td>Door</td><td>Alarm/ 響鈴</td></tr><tr><td>模式 1</td><td>門鈴</td><td>Door</td><td>Alarm/ 響鈴</td></tr></table>		Relay0	Relay1	Relay2	模式 0	Door	Door	Alarm/ 響鈴	模式 1	門鈴	Door	Alarm/ 響鈴	0x00	Relay0 代表主機上的 Relay Relay1~2 代表 ACU-30 的 Relay1~2
	Relay0	Relay1	Relay2													
模式 0	Door	Door	Alarm/ 響鈴													
模式 1	門鈴	Door	Alarm/ 響鈴													
14	0x0050	脅迫碼(不用的位數補 0xFF)	'1190'													
1	0x005E	Alarm Relay 重複觸發模式 Bit0~5 0 = Alarm Relay 動作後不重複觸發 1~63=Alarm Relay 動作幾分鐘後關閉 Bit6~7 = 保留	0x00	Alarm Relay Latch 模式才有作用												
1	0x005F	Alarm Relay 重複觸發模式關閉 Relay 的時間及重複觸發次數 Bit0~4 1~31= Relay關閉 1~31 分鐘後再次觸發 Bit5~7 1~7 = 重複觸發 1~7 次	0x00	Alarm Relay Latch 模式才有作用												
2	0x0060	Relay0 狀態設定 (Door) Bit0~11 = Pulse 時間 0=參考其他模式 1~4095=0.1~409.5 秒 Bit12~14 = 保留 Bit15 = 其他模式 (Pulse 時間為 0) 0=Toggle 1=Latch (無此模式)	0x28,0x00	Lo Byte,Hi Byte 0x0028=40 40*0.1 秒=4 秒												
2	0x0062	Relay1 狀態設定 (Alarm) Bit0~11 = Pulse 時間 0=參考其他模式 1~4095=0.1~409.5 秒 Bit12~14 = 保留 Bit15 = 其他模式 (Pulse 時間為 0) 0=Toggle	0x00,0x80	Lo Byte,Hi Byte												

		1=Latch														
1	0x004E	主機參數設定 Bit0:LCD 顯示班別時是否自動恢復 0=班別顯示不恢復 1=班別顯示恢復 Bit1:達到錯誤次數時啟動 Alarm Relay 0=Disable (不啟動 Relay) 1=Enable (啟動 Relay) Bit2:鐵捲門模式 0:門禁模式 1:鐵捲門模式 Bit3~7:保留	0x01													
1	0x004F	Relay 動作模式選擇 <table><tr><td></td><td>Relay0</td><td>Relay1</td><td>Relay2</td></tr><tr><td>模式 0</td><td>Door</td><td>Door</td><td>Alarm/ 響鈴</td></tr><tr><td>模式 1</td><td>門鈴</td><td>Door</td><td>Alarm/ 響鈴</td></tr></table>		Relay0	Relay1	Relay2	模式 0	Door	Door	Alarm/ 響鈴	模式 1	門鈴	Door	Alarm/ 響鈴	0x00	Relay0 代表主機上的 Relay Relay1~2 代表 ACU-30 的 Relay1~2
	Relay0	Relay1	Relay2													
模式 0	Door	Door	Alarm/ 響鈴													
模式 1	門鈴	Door	Alarm/ 響鈴													
1	0x005E	Alarm Relay 重複觸發模式 Bit0~5 0 = Alarm Relay 動作後不重複觸發 1~63=Alarm Relay 動作幾分鐘後關閉 Bit6~7 = 保留	0x00	Alarm Relay Latch 模式才有作用												
1	0x005F	Alarm Relay 重複觸發模式關閉 Relay 的時間及重複觸發次數 Bit0~4 1~31= Relay關閉 1~31 分鐘後再次觸發 Bit5~7 1~7 = 重複觸發 1~7 次	0x00	Alarm Relay Latch 模式才有作用												
2	0x0060	Relay0 狀態設定 (Door) Bit0~11 = Pulse 時間 0=參考其他模式 1~4095=0.1~409.5 秒 Bit12~14 = 保留 Bit15 = 其他模式 (Pulse 時間為 0) 0=Toggle 1=Latch (無此模式)	0x28,0x00	Lo Byte,Hi Byte 0x0028=40 40*0.1 秒=4 秒												

2	0x0062	Relay1 狀態設定 (Alarm) Bit0~11 = Pulse 時間 0=參考其他模式 1~4095=0.1~409.5 秒 Bit12~14 = 保留 Bit15 = 其他模式 (Pulse 時間為 0) 0=Toggle 1=Latch	0x00,0x80	Lo Byte,Hi Byte
1	0x005E	Alarm Relay 重複觸發模式 Bit0~5 0 = Alarm Relay 動作後不重複觸發 1~63=Alarm Relay 動作幾分鐘後關閉 Bit6~7 = 保留	0x00	Alarm Relay Latch 模式才有 作用
1	0x0069	1=Enable/0=Disable Bit0 Mifare Card Enable/Disable Bit1 CPU Card Enable/Disable Bit2-7 保留	0x03	Mifare/CPU Card Enable (PMF Only)
1	0x006A	RTC 微調 Bit0-6:每多少小時調整 1 秒 Bit7:0=調慢,1=調快	0x08	每 8 小時調慢 1 秒
1	0x006B	重複刷卡檢查秒數 0=關閉此功能 範圍 1-255 秒	0x00	

7.7.1 時段表 (位址 0100H~047FH : 共 896Bytes)

128*7	0100H	時段表 128 組*7Byte		
-------	-------	-----------------	--	--

Byte1 : 星期一對應到的時間表 (00H~7FH 128 組)

Byte2 : 星期二對應到的時間表 (00H~7FH 128 組)

.....

Byte7 : 星期日對應到的時間表 (00H~7FH 128 組)

不使用請填入 FFH。

7.7.2 時間表 (位址 0480H~067FH : 共 512Bytes)

128*4	0480H	時間表 128 組*4Byte		
-------	-------	-----------------	--	--

起始時間 : 2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

13 : 45 => 13H,45H (BCD 格式)

結束時間 : 2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

20 : 12 => 20H,12H (BCD 格式)

不使用請填入 FFH。

7.7.3 假日表 (位址 0680H~08D7H : 共 600Bytes)

200*3	0680H	假日表 200 組*3Byte		今年 0~99 組 明年 100~199 組
-------	-------	-----------------	--	------------------------------

假日日期 : 2 byte,第一個 byte 為月,第二個 byte 為日,例如

12 月 25 日 => 12H,25H (BCD 格式)

對應到的時間表 : 1 byte 例如

第二組時間表 => 01H (00H~7FH 128 組)

不使用請填入 FFH。

7.7.4 持續開門時段 (位址 08D8H~08DFH : 共 8Bytes)

8*1	08D8H	持續開門時段 8 組		
-----	-------	------------	--	--

不使用請填入 FFH。

7.7.5 響鈴時間表 (位址 08E0H~097FH : 共 160Bytes)

32*5	08E0H	響鈴時間表 32 組*5Byte		
------	-------	------------------	--	--

起始時間 : 2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

08 : 30 => 08H,30H (BCD 格式)

動作時間 : 2 byte,第一個 byte 為分鐘,第二個 byte 為秒,例如

00 : 12 => 00H,12H (BCD 格式)

動作星期 : 1 byte

bit7 : 保留,bit6~0 : 日六五四三二一

不使用請填入 FFH。

7.7.6 Anti Reset 時間表 (位址 0980H~099FH : 共 32Bytes)

16*2	0980H	定時清除 Anti 標記,16 組* 2Byte		
------	-------	--------------------------	--	--

時間 : 2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

23 : 50 => 23H,50H (BCD 格式)

不使用請填入 FFH。

7.7.7 班別時間表 (位址 09A0H~09FFH : 共 96Bytes)

32*3	09A0H	32 組* 3Byte		
------	-------	-------------	--	--

組別	時間(2byte)	班別(1byte)
第 1 組位址 0090H	HH : MM	Class
第 2 組位址 0093H	HH : MM	Class
第 3 組位址 0096H	HH : MM	Class
.	.	.
.	.	.
.	.	.
第 32 組位址 00EDH	HH : MM	Class

時間 : 2 byte,第一個 byte 為小時,第二個 byte 為分鐘,例如

08 : 10 => 08H,10H (BCD 格式)

班別 : 1 byte

0=空白	1=DUTY ON
2=DUTY_OFF	3=BREAK OUT
4=BREAK_IN	5=OT_START
6=OT_END	

不使用請填入 FFH。

7.7.8 加密時段 (位址 0x0A80~0x0A87:共 8Bytes)

8*1	0x0A80	加密時段 8 組		
-----	--------	----------	--	--

* 不使用請填入 0xFF

7.7.9 警報時段 (位址 0x0A88~0x0A8B:共 4Bytes)

4*1	0x0A88	警報時段 1 組		
-----	--------	----------	--	--

2 byte	2 byte
開始時間 HHMM	結束時間 HHMM

開始時間:兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘例如

13:25 => 13H,25H (BCD 格式)

結束時間:兩個 byte,第一個 byte 為小時,第二個 byte 為分鐘例如

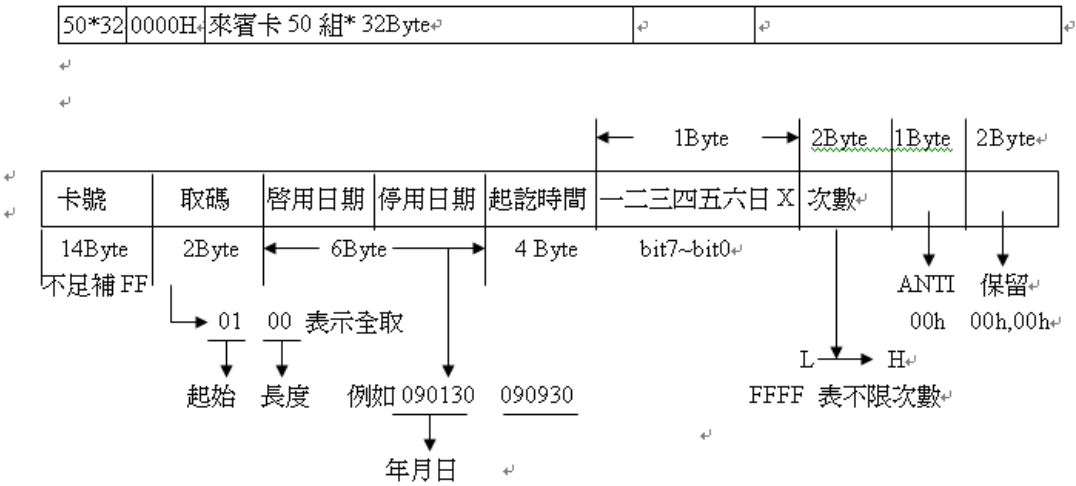
17:01 => 17H,01H (BCD 格式)

* 不使用請填入 0xFF

7.7.11 刷卡資料紀錄區 (位址 028000H~07FFFFH)

7.7.12SRAM 相關記憶體規劃 (R/W Table 15,16)

•來賓卡 50 筆 (位址 0000H~063FH:共 1600Bytes)



7.7.13 事件紀錄 200 筆 (位址 0640H~0C7FH : 共 1600Bytes)

200*8	0640H	事件紀錄 200 組* 8Byte		
-------	-------	-------------------	--	--