Windows Base

HDACS
API Reference Manual

RAC-340/520/820/920/930

Version 2.3

Table of Contents

This API reference manual is provided to help developers' access software via API implementation therefore expediting their software development time. It uses Virtual Studio .NET Professional 2003 C++ as its software development tool.

Chapter 1 System Requirement

1. The DLL (Dynamic Link Library) supports Multi threading technology.
2. The DLL (Dynamic Link Library) support Delphi, VB6 (Visual Basic), VC(Visual C++), and C# development tools.
3. Four types sample code：VC2005, C#2005, Delphi7 and VB6.

Chapter 2 Hardware Supported List

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

Chapter 3 Return Value

Return Value	Description
0	True
1125	Overtime when operate multi-threading programs.
1126	Error during released multi-threading.
1001	Error a sending a parameter. Or device returned an error code. Kindly refer to appendix. ReturnCode。
1002	Socket or communication port read/write error. An error occurred during asynchronous read/write.
1003	Data length too short, device returned an invalid data length.
1103	Length of packet small then reqeust
1004	Invalid control handler received. Invalid hComm value,
1005	Error in the packet no. returned.
1006	Error of 16-bit Cyclic Redundancy Check (CRC-16) returned.
1106	Error of 16-bit Cyclic Redundancy Check (CRC-16) set.
1007	PC sends wrong order to device or device does not support this function.
1008	An error occurred while performing read/write to slave device.
1009	Data length transmitted exceeded max. allowed length.
1010	No data was retrieved
4445	An error while reading device data or records
1025	Operation timed out during asynchronous read/write.
1026	Operation error during asynchronous read/write.
2225	Data was not retrieved during asynchronous read/write.

Chapter 4 RAC Series Functions

4.1 Communication – API (Level 0)

4.1.1 hacOpenChannel (Open Communication Channel)

API	hacOpenChannel
Function	int __stdcall hacOpenChannel (char *sComm, unsigned int iPort, HANDLE *hComm)
Purpose	Open TCP/IP or COM Port
Arguments	For RS-232 Com Port: sComm is COM1~COM128 iPort is Baudrate. Baudrate(1200/2400/4800/9600/19200/38400) hComm is handle value if it returns true. For TCP/IP: sComm is IP addres, like 172.16.1.1 iPort is port number, like 4660 hComm is handle value if it returns true. NOTE: When using TCP/IP converter, it is recommended NOT to stay online with the computer except during polling, perform connection only When upload or download of data is required. Like BF-430 or eP-132, socket connection can only be performed in a short period of time (depending upon the parameter settings of the device), so it is advised to disconnect When data transfer is not needs.
Return Value	Kindly refer to Chapter 3 for return value
Application	
Remark	

4.1.2 hacCloseChannel (Closes Communication Channel)

API	hacCloseChannel	
Function	int __stdcall hacCloseChannel (HANDLE hComm)	
Purpose	Close communication with device	
Arguments	hComm	Handle value to be closed; If handle > 1000 then it is a TCP/IP port.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.1.3 hacReadData (Reads data in Communication Buffer)

API	hacReadData	
Function	int __stdcall hacReadData (char *cBuffer, int *iDataLen, HANDLE hComm, unsigned int iTimeout)	
Purpose	Reads data in communication buffer	
Arguments	cBuffer	Contents of data to be transmitted
	iDataLen	Length of data to be transmitted
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (millisecond) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.1.4 hacWriteData (Sends data to communication channel)

API	hacWriteData	
Function	int __stdcall hacWriteData (char *cBuffer, int iDataLen, int iWrittenLen, HANDLE hComm)	
Purpose	Sends data from cBuffer to COM Port or TCP/IP	
Arguments	cBuffer	Contents of data to be transmitted
	iDataLen	Length of data to be transmitted
	iWriteenLen	Data length of success transmitted
	iTimeout	Wait time (millisecond) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.1.5 hacClearBuffer (Clears data in buffer)

API	hacClearBuffer	
Function	int __stdcall hacClearBuffer (HANDLE hComm)	
Purpose	Clears all data stored in the buffer.	
Arguments	hComm	COM Port or TCP/IP handle value
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.2 Protocol - API (Level1)

4.2.1 hacHWReadCommand (Sends read command to device)

API	hacHWReadCommand	
Function	int __stdcall hacHWReadCommand (int iNodeID, int iIndex, char *cSendData, int iSendDataLen, char *cReceiveData, int *iReceiveLen, HANDLE hComm, unsigned int iTimeout)	
Purpose	Sends the value of cSendData from COM Port or TCP/IP to the device and waits for the response packet and checks if the returned value is true.	
Arguments	iNodeID	Device ID
	Int iIndex	Index of Read-Write table
	cSendDataLen	Contents of data to be transmitted, Reserved
	iSendDataLen	Length of data to be transmitted. Reserved
	cReceiveData	Contents of data retrived
	iReceiveDataLen	Length of data retrieved
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (millisecond) for the response of the device. If not specified, system assigns 100 ms wait time.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.2.2 hacHWWriteCommand (Sends write command to device)

API	hacHWWriteCommand	
Function	int __stdcall hacHWWriteCommand (int iNodeID, int iIndex, char *cSendData, int iSendDataLen, char *cReceiveData, int *iReceiveLen, HANDLE hComm, unsigned int iTimeout)	
Purpose	Sends the value of cSendData from COM Port or TCP/IP to the device and waits for the response packet and checks if the returned value is true.	
Arguments	iNodeID	Device ID
	Int iIndex	Index of Read-Write table
	cSendDataLen	Contents of data to be transmitted, Reserved
	iSendDataLen	Length of data to be transmitted. Reserved
	cReceiveData	Contents of data retrived
	iReceiveDataLen	Length of data retrieved
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (millisecond) for the response of the device. If not specified, system assigns 100 ms wait time.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.2.3 hacHWReadCommandEX (Sends read command to device)

API	hacHWReadCommandEX	
Function	int __stdcall hacHWReadCommandEX (int iNodeID, int iIndex, char *cSendData, int iSendDataLen, char *cReceiveData, int *iReceiveLen, HANDLE hComm, unsigned int iTimeout)	
Purpose	Sends the value of cSendData from COM Port or TCP/IP to the device and waits for the response packet and checks if the returned value is true.	
Arguments	iNodeID	Device ID
	Int ilndex	Index of Read-Write table
	cSendDataLen	Contents of data to be transmitted, Reserved
	iSendDataLen	Length of data to be transmitted. Reserved
	cReceiveData	Contents of data retrived. Unlike hacHWReadCommand, cReceiveData of hacHWReadACommandEX does not carry the header and trailer of the packet.
	iReceiveDataLen	Length of data retrieved. Unlike hacHWReadCommand, iReceiveDataLen of iReceiveDataLen does not carry the header and trailer of the packet.
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (millisecond) for the response of the device. If not specified, system assigns 100 ms wait time.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.2.4 hac34Polling (Polling Device)

<table>
<tr><td>API</td><td colspan="2">hac34Polling</td></tr>
<tr><td>Function</td><td colspan="2">int __stdcall hac34Polling (
int iNodeID,
int iPrevRecord,
stPollList *stRecord
int *iRecord,
HANDLE hComm,
unsigned int iTimeout,
int iCardType
)</td></tr>
<tr><td>Purpose</td><td colspan="2">Send polling commands and read swipe card records/events from device.</td></tr>
<tr><td rowspan="7">Arguments</td><td>iNodeID</td><td>Device ID</td></tr>
<tr><td>iPrevRecord</td><td>No. of records previously retrieved</td></tr>
<tr><td>stRecord</td><td>Structure that point to contents of data retrieved</td></tr>
<tr><td>iRecord</td><td>No of events retrieved</td></tr>
<tr><td>hComm</td><td>COM Port or TCP/IP handle value</td></tr>
<tr><td>iTimeout</td><td>Wait time (millisecond) for the response of the device.</td></tr>
<tr><td>iCardType</td><td>Indicates whether the card no. is compressed or not.
0 signifies uncompressed card number
1 signifies compressed card number.</td></tr>
<tr><td>Return Value</td><td colspan="2">Kindly refer to Chapter 3 for return value</td></tr>
<tr><td>Application</td><td colspan="2"></td></tr>
<tr><td>Remark</td><td colspan="2">The data retrieved are decrypted data.
typedef struct stPollList
{
char cEventCode[5]; //Event Code
char cDateTime[20]; //Date and Time
char cCard[20]; //Card number
char cDeviceID[10]; //Device ID
char cReader[10]; //Reader ID
} stPollList;
Event Code List:</td></tr>
</table>

03 01	Case Sensor Alarm
03 05	Forced-door-open Detection Deactivated
03 06	Cold Startup
03 07	Restart
03 11	Forced-door-open Detection Activated
03 12	Door Prop Alarm Activated
03 13	Door Prop Alarm Deactivated
03 18	Door Opened
03 19	Door Closed
03 1A	Exit Button Pressed
01 20	Duress Alarm Activated
01 22	Duress Alarm Deactivated

Event Code list for Card:

0000	Card does not contain data
0001	Duty On-Valid Card Input
0011	Duty On-Valid Card Swiped
0101	Duty Off-Valid Card Input
0111	Duty Off-Valid Card Swiped
0007	Duty On-Invalid Card Input
0017	Duty On-Invalid Card Swiped
0107	Duty Off-Invalid Card Input
0117	Duty Off-Invalid Card Swiped

4.3 Device - API(Level 2)

4.3.1 hac34GetDateTime (Read Device Date and Time)

API	hac34GetDateTime	
Function	int __stdcall hac34GetDateTime (int iNodeID, char *cDate, char *cTime, HANDLE hComm, unsigned int iTimeout)	
Purpose	Retrieve device's date and time	
Arguments	iNodeID	Device ID
	cDate	Date retrieved. Format (YYYYMMDD)+Weekday
	cTime	Time retrieved. Format (HHMMSS)
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (millisecond) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.3.2 hac34SetDateTime (Set Device's Date and Time)

API	hac34SetDateTime	
Function	int __stdcall hac34SetDateTime (int iNodeID, char *cDate, char *cTime, HANDLE hComm, unsigned int iTimeout)	
Purpose	Sets the date and time within the device	
Arguments	iNodeID	Device ID
	cDate	Date retrieved. Format (YYYYMMDD)+Weekday
	cTime	Time retrieved. Format (HHMMSS)
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (millisecond) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.3.3 hac34GetEEData (Read EEPROM's Parameter)

API	hac34GetEEData	
Function	int __stdcall hac34GetEEData (int iNodeID, char *cEEData, int *iReceiveDataLen, int iEEArea, unsigned int *iEEAddr, int *iEELen, HANDLE hComm, unsigned int iTimeout)	
Purpose	Read parameters from EEPROM	
Arguments	iNode ID	Device ID
	cEEData	Contents of data to be transmitted
	iReceiveDataLen	Length of data retrieved
	iEEArea	1: Parameter Section 2: Card Number Section 3: Card Swiped Records Section
	iEEAddr	Location of content retrieved
	iEELen	Length of content retrieved, max. 64 bytes
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (ms) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.3.4 hac34SetEEData (Sets EEPROM Value)

API	hac34SetEEData	
Function	int __stdcall hac34SetEEData (int iNodeID, char *cEEData, unsigned int iEEAddr, int iEELen, int iEEArea, HANDLE hComm, unsigned int iTimeout)	
Purpose	Writes data into the device's EEPROM	
Arguments	iNode ID	Device ID
	cEEData	Contents of data to be set
	iEEArea	1: Parameter Section 2: Card Number Section
	iEEAddr	Location of content to set
	iEELen	Length of content to set, max. 32 bytes
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (ms) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.3.5 hac34GetVersion (Read Device Version)

API	hac34GetVersion	
Function	int __stdcall hac34GetVersion (int iNodeID, char *cData, HANDLE hComm, unsigned int iTimeout)	
Purpose	Read device version	
Arguments	iNodeID	Device ID
	cData	Retrieved device's version, ROM File and date
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (ms) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark	cData is composed of 3 parts: a. Model Number RAC-340PE，RAC-340PM RAC-520PE，RAC-520PM b. ROM Version, Major, Minor V1.00 V2.18	

4.3.6 hac34AddCard (Add a Single Uncompressed Card)

API	hac34AddCard	
Function	int __stdcall hac34AddCard (int iNodeID, char *cCardNo, int iCardLen, char *cPassWord, int iPassLen, int iTimeZone, char cStatus, HANDLE hComm, unsigned int iTimeout)	
Purpose	Add a single uncompressed card	
Arguments	iNodeID	Device ID
	cCardNo	Card number which should be ASCII code.
	char *cOassWird	Password of ASCII code
	iCardLen	Length of card number. Max. 10 digits
	iPassLen	Length of password. Max. 4 digits
	iTimeZone	Time schedule. Reserved.
	cStatus	Status code, Reserved
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (ms) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.3.7 hac34DelCard (Delete a Single Uncompressed Card)

API	hac34DelCard	
Function	int __stdcall hac34DelCard (int iNodeID, char *cCardNo, int iCardLen, HANDLE hComm, unsigned int iTimeout)	
Purpose	Delete a single uncompressed card	
Arguments	iNodeID	Device ID
	cCardNo	Card number to be deleted
	iCardLen	Length of card number
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (ms) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.3.8 hac34DelAllCard (Delete All Valid Card)

API	hac34DelAllCard	
Function	int __stdcall hac34DelAllCard (int iNodeID, HANDLE hComm, unsigned int iTimeout)	
Purpose	Delete all valid cards within the device	
Arguments	iNodeID	Device ID
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (ms) for the response of the device. This function may need more times. Max. 10 seconds.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.3.9 hac34SetDoorRelayMode (Set Relay Action Mode)

API	hac34SetDoorRelayMode	
Function	int __stdcall hac34SetDoorRelayMode (int iNodeID, int ActionMode, int ActionMask, HANDLE hComm, unsigned int iTimeout)	
Purpose	Sets the relay action mode upon releasing the door.	
Arguments	iNodeID	Device ID
	ActionMode	1. Reader Beep / Alarm Relay On 2. Enable/Disable Auto Learning Function 4. Display Card number or display * for Card No 8. Reserved 16: Language Selection 32: Language Selection 64: Enable/Disable door forced open 128: Enable/Disable door prop detection
	ActionMask	Performs bit masking on ActionMode
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (ms) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.3.10 hac34SetStatus (Set Reader Statis)

<table>
<tr><td>API</td><td colspan="2">hac34SetStatus</td></tr>
<tr><td>Function</td><td colspan="2">int __stdcall hac34SetStatus (
int iNodeID,
int iStatus,
int iMask,
HANDLE hComm,
unsigned int iTimeout
)</td></tr>
<tr><td>Purpose</td><td colspan="2">Read status from RAM of device</td></tr>
<tr><td rowspan="6">Atguments</td><td>iNodeID</td><td>Device ID</td></tr>
<tr><td>iStatus</td><td>Set status.
<table>
<tr><td>Bit0</td><td>0→ LCD Backlight Off
1→ LCD Backlight On</td></tr>
<tr><td>Bit1</td><td>0→ Keypad Backlight Off
1→ Keypad Backlight On</td></tr>
<tr><td>Bit2</td><td>0→ Disable Keypad
1→ Enalbie keypad</td></tr>
<tr><td>Bit3</td><td>0→ Disable Function Key
1→ Enable Function Key</td></tr>
<tr><td>Bit4</td><td>0→ Disable Reader
1→ Enable Reader</td></tr>
<tr><td>Bit5</td><td>0→ Keypad Tones Off
1→ Keypad Tones On</td></tr>
<tr><td>Bit6</td><td>Reserved</td></tr>
<tr><td>Bit7</td><td>0→ Disable Case Sensor
1→ Enable Case Sensor</td></tr>
</table></td></tr>
<tr><td>iMask</td><td>Performs bit mask operation.
0→ Enable 1→ Disable (Bit6 is always disabled)</td></tr>
<tr><td>hComm</td><td>COM Port or TCP/IP handle value</td></tr>
<tr><td>iTimeout</td><td>Wait time (ms) for the response of the device.</td></tr>
<tr><td>Return Value</td><td colspan="2">Kindly refer to Chapter 3 for return value</td></tr>
<tr><td>Application</td><td colspan="2"></td></tr>
<tr><td>Remark</td><td colspan="2"></td></tr>
</table>

3.11 hac34SetReadCardParameter (Set Mifare Parameters)

API	hac34SetReadCardParameter	
Function	int __stdcall hac34SetReadCardParameter (int iNodeID, int iKeyType, int iBlock, int iStartDigit, int iDigitLength, int iCompact, unsigned char *cKeyValue, HANDLE hComm, unsigned int iTimeout)	
Purpose	Set Mifare Parameters	
Arguments	iNodeID	Device ID
	iKeyType	→0 CSN, →1 KeyA, →2 KeyB
	iBlock	Block address (0x00~0x3F) in Mifare card where card number is stored, if iKeyType>0.
	iStartDigit	The starting byte within the block where reading would start (0-15)
	iDigitLength	Length of Card No retrieved (0-12)
	iCompact	Indicates whether the card no. is compressed or not. 0→ Uncompressed 1→ Compressed
	cKeyValue	KeyA, KeyB (6 bytes)
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (ms) for the response of the device.
Return Value	Kindly refer to Chapter 3 for return value	
Application		
Remark		

4.3.12 hac34GetReadCardParameter (Read Mifare Parameters)

API	hac34GetReadCardParameter	
Function	int __stdcall hac34GetReadCardParameter (HANDLE hComm, int iNodeID, int *iKeyType, int *iBlock, int *iStartDigit, int *iDigitLength, int *iCompact, unsigned int iTimeout)	
Purpose	Read Mifare Parameters	
Arguments	hComm	COM Port or TCP/IP handle value
	iNodeID	Device ID
	iKeyType	→0 CSN, →1 KeyA, →2 KeyB
	iBlock	Block address (0x00~0x3F) in Mifare card where card number is stored, if iKeyType>0.
	iStartDigit	The starting byte within the block where reading would start (0-15)
	iDigitLength	Length of Card No retrieved (0-12)
	iCompact	Indicates whether the card no. is compressed or not. 0→ Uncompressed 1→ Compressed
	iTimeout	Wait time (ms) for the response of the device.
Return Value		
Application		
Remark		

4.3.13 hac34GetIO (Retrieve Device's IO Status)

API	hac34GetIO	
Function	int __stdcall hac34GetIO (int iNodeID, char *cStatus, HANDLE hComm, unsigned int iTimeout)	
Purpose	Retrieve device's IO status	
Arguments	iNodeID	Device ID
	Cstatus	Retrieve IO status, 2 Byte
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (ms) for the response of the device.
Return Value		
Application		
Remark		

4.3.14 hac34Dump (DUMP Info from Device)

API	hac34Dump	
Function	int __stdcall hac34Dump (int iNodeID, int iDumpArea, char *cDumpData, HANDLE hComm, unsigned int iTimeout)	
Purpose	Dump the card number and swiped card records from device to PC.	
Arguments	iNodeID	Device ID
	iDumpArea	1. Card Number 2. Swipe Card Records
	cDumpData	Dump data retrieved
	hComm	COM Port or TCP/IP handle value
	iTimeout	Wait time (ms) for the response of the device.
Return Value		
Application		
Remark		

Chapter 5 Appendix

5.1 RAC-340 series Memory Allocation

Address	Description	Remarks	
0x00	Model Number (Read Only)	0=RAC-340PM 1=RAC-340PE 2=RAC-520PM 3=RAC-520PE 4=Reserved 5=SHR-100 6=RAC-920PM 7=RAC-920PE 8=RAC-820PM 9=RAC-820PE 10=RAC-930PM 10=RAC-930PE	
0x01	Major Version (Read Only)	BCD Format	Ex: Major version 01h, Minor version 10h, which means Version 1.10
0x02	Minor Version (Read Only)	BCD Format	Ex: Major version 11h, Minor version 02h, which means Version 11.02
0x03~0x09	Reserved		
0x0A	Sensor Active	-Bit0: Push Button. 1→ N.O. (Default) 0→ N.C. -Bit1: Reserved -Bit2: Door Sensor. 1→N.C (Default) 0→N.O. -Bit3~7: Reserved	
0x0B	CardJudgeMode	0→ Standard Mode 1→ Random Mode1 (Open door by right Mifare key, system will not save swipe card records) 2→ Random Mode2 (System will check card format, Valid card will becomre to blacklist)	
0x0C~0x0D	Door Relay Action Time	Lower Byte (000.00~655.35 sec)	
0x0E~0x0F	Alarm Relay Action Time	Lower Byte (000.00~655.35 sec)	
0x10~0x11	Door Sensor Time	Lower Byte (000.01~655.35 sec , 0:OFF)	
0x12	Duress Code Length	1~10 digits	
0x13~0x1C	Duress Code	Appends 0xFF if code is insufficient	
0x1D	Disarm Code Length	1~10 digits	
0x1E~0x27	Disarm Code	Appends 0xFF if code is insufficient	
0x28	Advanced Reader Settings 1		

<table>
<tr><td></td><td></td><td><table>
<tr><td>Bit0</td><td>0→ Alarm Relay On
1→ Reader Beep</td></tr>
<tr><td>Bit1</td><td>Reserved</td></tr>
<tr><td>Bit2</td><td>0→ Display Card No
1→ Display * mark</td></tr>
<tr><td>Bit3</td><td>Reserved</td></tr>
<tr><td>Bit4~5</td><td>0→ Traditional Chinese
1→ English
2→ Simplified Chinese</td></tr>
<tr><td>Bit6</td><td>0→ Enable Forced Door Open Detection
1→ Disable Forced Door Open Detection</td></tr>
<tr><td>Bit7</td><td>0→ Enable Door Prop Detection
1→ Disable Door Prop Detection</td></tr>
</table></td></tr>
<tr><td>0x29</td><td>Mifare Retrieved Mode
(Mifare Model Only)</td><td>0→ Auto
1→ Retrieve Fixed Length</td></tr>
<tr><td>0x2A</td><td>Advanced Reader Settings 2</td><td><table>
<tr><td>Bit0</td><td>0→ LCD Backlight Off
1→ LCD Backlight On</td></tr>
<tr><td>Bit1</td><td>0→ Keypad Backlight Off
1→ Keypad Backlight On</td></tr>
<tr><td>Bit2</td><td>0→ Disable Keypad
1→ Enalbie keypad</td></tr>
<tr><td>Bit3</td><td>0→ Disable Function Key
1→ Enable Function Key</td></tr>
<tr><td>Bit4</td><td>0→ Disable Reader
1→ Enable Reader</td></tr>
<tr><td>Bit5</td><td>0→ Keypad Tones Off
1→ Keypad Tones On</td></tr>
<tr><td>Bit6</td><td>Reserved</td></tr>
<tr><td>Bit7</td><td>0→ Disable Case Sensor
1→ Enable Case Sensor</td></tr>
</table></td></tr>
<tr><td>0x2B</td><td>Compare Valid Code</td><td>MSB 4 bit: compare valid code index-1
LSB 4 bit: compare valid code length 0~9,
0 signifies select all 10 digis.</td></tr>
</table>

0x2C	Reswipe Card Check Time	(1~255sec , 0:OFF)
0x2D	Mifare Read Type (Mifare Model Only)	0: Serial Number 1:KeyA 2:KeyB
0x2E	MifareReadAddressAndLength (Mifare Model Only)	MS4Bit: Address LS4B:Length
0x2F	MifareCompact (Mifare Model Only)	0:Uncompressed 1:Compressed
0x30	MifareReadBlock (Mifare Model Only)	Read Block (0x1~0x3F)
0x31~0x37	Reserved	
0x38	Master Card Length	1~10 digits
0x39~0x42	Master Card Number	Appends 0xFF if code is insufficient
0x43	Open Door Upon Master Card Swipe	0→ Access Denied 1→ Access Granted
0x44	Memory Mode	0→ Standard 1→ Compressed 2→ Shared
0x45	Access Mode	0→ Swipe Card Only 1→ Swipe Card or Input Card No 2→ Swipe Card + Password
0x46	Save Invalid Card Number	0→ Do not save 1→ Save
0x47~0x48	Switch to Duty On	BCD format
0x49~0x4A	Switch to Duty Off	BCD format
0x4B~0x4C	Unrestricted Duration	Lower Byte (1~9999sec)
0x4D	Reserved	
0x4E	Reserved	
0x4F	Reserved	
0x50~0x6F	Siren Time＋ Duration Total of 8 set with 4 Bytes each	BCD format
0x70	Siren Schedule	b0: MON, b1: TUE, b2: WEB, b3: THU, b4: FRI b5: SAT, b6: SUN
0x71	Access Control or Time Attendance Mode	0→ Access Control 1→ Time Attendance

5.2 RAC-520 series Memory Allocation

Address	Description	Remarks	
0x00	Model Number (Read Only)	0=RAC-340PM 1=RAC-340PE 2=RAC-520PM 3=RAC-520PE 4=Reserved 5=SHR-100 6=RAC-920PM 7=RAC-920PE 8=RAC-820PM 9=RAC-820PE 10=RAC-930PM 10=RAC-930PE	
0x01	Major Version (Read Only)	BCD Format	Ex: Major version 01h, Minor version 10h, which means Version 1.10
0x02	Minor Version (Read Only)	BCD Format	Ex: Major version 11h, Minor version 02h, which means Version 11.02
0x03~0x09	Reserved		
0x0A	Sensor Active	-Bit0: Push Button. 1→ N.O. (Default) 0→ N.C. -Bit1: Reserved -Bit2: Door Sensor. 1→N.C (Default) 0→N.O. -Bit3~7: Reserved	
0x0B	CardJudgeMode	0→ Standard Mode 1→ Random Mode1 (Open door by right Mifare key, system will not save swipe card records) 2→ Random Mode2 (System will check card format, Valid card will becomre to blacklist)	
0x0C~0x0D	Door Relay Action Time	Lower Byte (000.00~655.35 sec)	
0x0E~0x0F	Alarm Relay Action Time	Lower Byte (000.00~655.35 sec)	
0x10~0x11	Door Sensor Time	Lower Byte (000.01~655.35 sec , 0:OFF)	
0x12	Duress Code Length	1~10 digits	
0x13~0x1C	Duress Code	Appends 0xFF if code is insufficient	
0x1D	Disarm Code Length	1~10 digits	
0x1E~0x27	Disarm Code	Appends 0xFF if code is insufficient	
0x28	Advanced Reader Settings 1	Bit0: 0→ Alarm Relay On; 1→ Reader Beep	

<table>
<tr><td></td><td></td><td><table>
<tr><td>Bit1~5</td><td>Reserved</td></tr>
<tr><td>Bit6</td><td>0→ Enable Forced Door Open Detection
1→ Disable Forced Door Open Detection</td></tr>
<tr><td>Bit7</td><td>0→ Enable Door Prop Detection
1→ Disable Door Prop Detection</td></tr>
</table></td></tr>
<tr><td>0x29</td><td>Mifare Retrieved Mode
(Mifare Model Only)</td><td>0→ Auto
1→ Retrieve Fixed Length</td></tr>
<tr><td>0x2A</td><td>Advanced Reader Settings 2</td><td><table>
<tr><td>Bit0</td><td>Reserved</td></tr>
<tr><td>Bit1</td><td>0→ Keypad Backlight Off
1→ Keypad Backlight On</td></tr>
<tr><td>Bit2</td><td>0→ Disable Keypad
1→ Enalbie keypad</td></tr>
<tr><td>Bit3</td><td>Reserved</td></tr>
<tr><td>Bit4</td><td>0→ Disable Reader
1→ Enable Reader</td></tr>
<tr><td>Bit5</td><td>0→ Keypad Tones Off
1→ Keypad Tones On</td></tr>
<tr><td>Bit6</td><td>Reserved</td></tr>
<tr><td>Bit7</td><td>0→ Disable Case Sensor
1→ Enable Case Sensor</td></tr>
</table></td></tr>
<tr><td>0x2B</td><td>Compare Valid Code</td><td>MSB 4 bit: compare valid code index-1
LSB 4 bit: compare valid code length 0~9,
0 signifies select all 10 digis.</td></tr>
<tr><td>0x2C</td><td>Reswipe Card Check Time</td><td>(1~255sec , 0:OFF)</td></tr>
<tr><td>0x2D</td><td>Mifare Read Type
(Mifare Model Only)</td><td>0: Serial Number 1:KeyA 2:KeyB</td></tr>
<tr><td>0x2E</td><td>MifareReadAddressAndLength
(Mifare Model Only)</td><td>MS4Bit: Address LS4B:Length</td></tr>
<tr><td>0x2F</td><td>MifareCompact
(Mifare Model Only)</td><td>0:Uncompressed 1:Compressed</td></tr>
<tr><td>0x30</td><td>MifareReadBlock
(Mifare Model Only)</td><td>Read Block (0x1~0x3F)</td></tr>
<tr><td>0x31~0x37</td><td>Reserved</td><td></td></tr>
<tr><td>0x38</td><td>Master Card Length</td><td>1~10 digits</td></tr>
</table>

0x39~0x42	Master Card Number	Appends 0xFF if code is insufficient
0x43	Open Door Upon Master Card Swipe	0→ Access Denied 1→ Access Granted
0x44	Memory Mode	0→ Standard 1→ Compressed 2→ Shared
0x45	Access Mode	0→ Swipe Card Only 1→ Swipe Card or Input Card No 2→ Swipe Card + Password
0x46	Save Invalid Card Number	0→ Do not save 1→ Save
0x47~0x48	Reserved	
0x49~0x4A	Reserved	
0x4B~0x4C	Unrestricted Duration	Lower Byte (1~9999sec)

5.3 RAC-820/920/930 series Memory Allocation

Address	Description	Remarks	
0x00	Model Number (Read Only)	0=RAC-340PM 1=RAC-340PE 2=RAC-520PM 3=RAC-520PE 4=Reserved 5=SHR-100 6=RAC-920PM 7=RAC-920PE 8=RAC-820PM 9=RAC-820PE 10=RAC-930PM 10=RAC-930PE	
0x01	Major Version (Read Only)	BCD Format	Ex: Major version 01h, Minor version 10h, which means Version 1.10
0x02	Minor Version (Read Only)	BCD Format	Ex: Major version 11h, Minor version 02h, which means Version 11.02
0x03~0x09	Reserved		
0x0A	Sensor Active	-Bit0: Push Button. 1→ N.O. (Default) 0→ N.C. -Bit1: Reserved -Bit2: Door Sensor. 1→N.C (Default) 0→N.O. -Bit3~7: Reserved	
0x0B	CardJudgeMode	0→ Standard Mode 1→ Random Mode1 (Open door by right Mifare key, system will not save swipe card records) 2→ Random Mode2 (System will check card format, Valid card will becomre to blacklist)	
0x0C~0x0D	Door Relay Action Time	Lower Byte (000.00~655.35 sec)	
0x0E~0x0F	Alarm Relay Action Time	Lower Byte (000.00~655.35 sec)	
0x10~0x11	Door Sensor Time	Lower Byte (000.01~655.35 sec , 0:OFF)	
0x12	Duress Code Length	1~10 digits	
0x13~0x1C	Duress Code	Appends 0xFF if code is insufficient	
0x1D	Disarm Code Length	1~10 digits	
0x1E~0x27	Disarm Code	Appends 0xFF if code is insufficient	
0x28	Advanced Reader Settings 1	Bit0: 0→ Alarm Relay On 1→ Reader Beep (Default)	

<table>
<tr><td></td><td></td><td><table>
<tr><td>Bit1~5</td><td>Reserved</td></tr>
<tr><td>Bit6</td><td>0→ Enable Forced Door Open Detection
1→ Disable Forced Door Open Detection</td></tr>
<tr><td>Bit7</td><td>0→ Enable Door Prop Detection
1→ Disable Door Prop Detection</td></tr>
</table></td></tr>
<tr><td>0x29</td><td>Mifare Retrieved Mode
(Mifare Model Only)</td><td>0→ Auto
1→ Retrieve Fixed Length</td></tr>
<tr><td>0x2A</td><td>Advanced Reader Settings 2</td><td><table>
<tr><td>Bit0</td><td>Reserved</td></tr>
<tr><td>Bit1</td><td>0→ Keypad Backlight Off
1→ Keypad Backlight On</td></tr>
<tr><td>Bit2</td><td>0→ Disable Keypad
1→ Enalbie keypad</td></tr>
<tr><td>Bit3</td><td>Reserved</td></tr>
<tr><td>Bit4</td><td>0→ Disable Reader
1→ Enable Reader</td></tr>
<tr><td>Bit5</td><td>0→ Keypad Tones Off
1→ Keypad Tones On</td></tr>
<tr><td>Bit6</td><td>Reserved</td></tr>
<tr><td>Bit7</td><td>0→ Disable Case Sensor
1→ Enable Case Sensor</td></tr>
</table></td></tr>
<tr><td>0x2B</td><td>Compare Valid Code</td><td>MSB 4 bit: compare valid code index-1
LSB 4 bit: compare valid code length 0~9,
0 signifies select all 10 digis.</td></tr>
<tr><td>0x2C</td><td>Reswipe Card Check Time</td><td>(1~255sec , 0:OFF)</td></tr>
<tr><td>0x2D</td><td>Mifare Read Type
(Mifare Model Only)</td><td>0: Serial Number 1:KeyA 2:KeyB</td></tr>
<tr><td>0x2E</td><td>MifareReadAddressAndLength
(Mifare Model Only)</td><td>MS4Bit: Address LS4B:Length</td></tr>
<tr><td>0x2F</td><td>MifareCompact
(Mifare Model Only)</td><td>0:Uncompressed 1:Compressed</td></tr>
<tr><td>0x30</td><td>MifareReadBlock
(Mifare Model Only)</td><td>Read Block (0x1~0x3F)</td></tr>
<tr><td>0x31~0x37</td><td>Reserved</td><td></td></tr>
<tr><td>0x38</td><td>Master Card Length</td><td>1~10 digits</td></tr>
</table>

<table>
<tr><td>0x39~0x42</td><td>Master Card Number</td><td>Appends 0xFF if code is insufficient</td></tr>
<tr><td>0x43</td><td>Open Door Upon Master Card Swipe</td><td>0→ Access Denied
1→ Access Granted</td></tr>
<tr><td>0x44</td><td>Memory Mode</td><td>0→ Standard
1→ Compressed
2→ Shared</td></tr>
<tr><td>0x45</td><td>Access Mode</td><td>0→ Swipe Card Only
1→ Swipe Card or Input Card No
2→ Swipe Card + Password</td></tr>
<tr><td>0x46</td><td>Save Invalid Card Number</td><td>0→ Do not save
1→ Save</td></tr>
<tr><td>0x47~0x48</td><td>Reserved</td><td></td></tr>
<tr><td>0x49~0x4A</td><td>Reserved</td><td></td></tr>
<tr><td>0x4B~0x4C</td><td>Unrestricted Duration</td><td>Lower Byte (1~9999sec)</td></tr>
<tr><td>0x4D</td><td>Slave reader and Anti setting</td><td><table><tr><td>Bit0</td><td>0→ Slave reader doesn’t need check password.
1→ Slave reader need check password.</td></tr><tr><td>Bit1</td><td>Reserved</td></tr><tr><td>Bit2</td><td>0→ Disable Anit-passback function.
1→ Enable Anti-passback function.</td></tr></table></td></tr>
<tr><td>0x4E</td><td>Built-in Relay Function</td><td>0→ Door relay
1→ Door bell relay (ACU-30 relay1 for door, Relay 2 for Alarm/siren)</td></tr>
</table>