

Windows Base  
HDTAS  
API Reference Manual

Version 2.0

# Table of Contents

Chapter 1 System Requirement .....	5
Chapter 2 Hardware Supported List .....	5
Chapter 3 Return Value .....	6
Chapter 4 HTA Series Functions .....	7
4.1 Communication – API (Level 0) .....	7
4.1.1 htaOpenChannel (Open Communication Channel) .....	7
4.1.2 htaCloseChannel (Close Communication Channel) .....	8
4.1.3 hsHTA850WriteTable (Sends table to HTA-850/852/860).....	8
4.1.4 hsHTA850ReadTable (Read the table from HTA-850/852/860).....	9
4.1.5 hsHTA850PollingData (Read Swiped Card Records from 850/852)..	10
4.1.6 htaPolling (Sends Polling Command to HTA-8x0) .....	11
4.1.7 hsHTA860PollingData (Read Swiped Card Records from HTA-860).	12
4.2 Device— API (Level 2) .....	13
HTA-8X0 series .....	13
4.2.1 htaGetDateTime (Read Device Date and Time) .....	13
4.2.2 htaSetDateTime (Set Device Date and Time).....	13
4.2.3 htaGetMemoryData (Reads the SRAM Data).....	14
4.2.4 htaSetMemoryData (Write Data to SRAM) .....	15
4.2.5 htaGetVersion (Reads the Device’s Version).....	15
4.2.6 htaAddCard (Add a Single Uncompressed Card).....	16
4.2.7 htaDelCard (Delete a Single Uncompressed Card).....	17
4.2.8 htaAddZCard (Add a Single Compressed Card) .....	18
4.2.9 htaDelZCard (Delete a Single Compressed Card) .....	19
4.2.10 htaEraseFlash (Format Flash Memory) .....	19
4.2.11 htaDeleteAllLog (Delete All Swipe Card Records) .....	20
4.2.12 htaDeleteAllCard (Delete All Valid Card) .....	20
4.2.13 htaRestart (Restart Device) .....	21
4.2.14 htaGetLogRecord (Read Device’s Decrypted Swiped Card Record).....	22
4.2.15 htaGetLogData (Read Log Files of Swiped Card Records) .....	23
4.2.16 htaGetCardData (Read Valid Card No) .....	24
HTA-850/852/860 series .....	25
4.2.17 hsHTA850ReadParameter (Read Parameter).....	25
4.2.18 hsHTA850WriteParameter (Set Parameter) .....	25
4.2.19 hsHTA850Initial (Device Initialization) .....	26
4.2.20 hsHTA850GetInfo (Read Device Version and Other Info) .....	27
4.2.21 hsHTA850InsertMultiUserRecord (Add Multiple Valid Cards) .....	28

4.2.22 hsHTA850DeleteUserRecord (Delete a Card).....	29
4.2.23 hsHTA850QueryUserRecord (Query Single Card's Auth) .....	29
4.2.24 hsHTA850DeleteAllUserRecord (Delete All Valid Cards).....	30
4.2.25 hsHTA850SetEEPROM (Set EEPROM Parameter).....	30
4.2.26 hsHTA850ReadEEPROM (Read EEPROM's Parameter) .....	31
4.2.27 hsHTA850WriteTime (Set Device Date and Time).....	32
4.2.28 hsHTA850Read Time (Read Device Date and Time) .....	33
4.2.29 hsHTA850SetMifareReader(Set Mifare Reader Parameter).....	33
4.2.30 hsHTA850InsertMultiUserFingerPrinter2 (Add New Cards).....	34
4.2.31 hsHTA850QueryUserFingerPrinter2 (Query Single Auth).....	35
4.2.32 hsHTA850QueryMasterFP (Query Master Fingerprint) .....	36
4.2.33 hsHTA850UpdateMasterFP (Change Master Fingerprint).....	37
<b>Chapter 5 Appendix .....</b>	<b>38</b>
<b>Appendix 1: HTA-8x0 series Description .....</b>	<b>38</b>
5.1.1 Valid Card's Format .....	38
5.1.2 Swiped Card Records .....	38
5.1.3 Date and Time Format.....	39
5.1.4 SRAM Memory Allocation.....	40
5.1.5 LED Message Display Communication Format .....	42
5.1.6 DVR Communication Format.....	42
5.1.7 Printer Communication Format.....	43
<b>Appendix 2: HTA-850/852 series Description .....</b>	<b>44</b>
5.2.1 System Parameter.....	44
5.2.2 Valid Card Format .....	45
5.2.3 Swiped Card Records Format Description .....	46
5.2.4 Memory Allocation .....	48
5.2.5 Siren Timetable .....	52
5.2.6 Work Time Timetable .....	53
5.2.7 Message Display .....	54
5.2.8 Index of Fingerprint machine .....	55
5.2.9 CMS:30 Mifare Settings .....	55
5.2.10 Error Code List .....	57
<b>Appendix 3: HTA-860 series Description .....</b>	<b>58</b>
5.3.1 Valid Card Format .....	58
5.3.2 Swiped Card Records Format Description .....	59
5.3.3 Memory Allocation .....	61
5.3.4 Siren Timetable .....	67
5.3.5 Work Time Timetable .....	68

5.3.6 Message Display .....	69
5.3.7 Reserve Area.....	70
5.3.8 CMS:30 Mifare Settings .....	70
5.3.9 Error Code List .....	72

## Rev History

V2.0	First edition
------	---------------

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, VB (Visual Basic), VC2005 (Visual C++), and C# development tools.
3. Sample code type : C#

## **Chapter 2 Hardware Supported List**

### **HTA-8x0 series includes:**

HTA-810(PE), HTA-810(PM), HTA-810(M), HTA-810(B)  
HTA-820(PE), HTA-820(PM), HTA-820(M), HTA-820(B)  
HTA-830(PE), HTA-830(PM)  
HTA-840(PE), HTA-810(PM)

### **HTA-850 series includes:**

HTA-850(PE), HTA-850(PM),

### **HTA-852 series includes:**

HTA-852(PEF), HTA-852(PMF), HTA-852(PMFC)

### **HTA-860 series includes:**

HTA-860(PE), HTA-860(PM), HTA-860(PEF), HTA-860(PMF), HTA-860(F)

## Chapter 3 Return Value

Define	Return Value	Description
HF_RET_SUCCESS	0	True
HF_ERR_HANDLE_WAIT_TIMEOUT	1125	Overtime when operate multi-threading programs.
HF_ERR_HANDLE_RELEASE	1126	Error during released multi-threading.
HF_ERR_PARAMETER	1001	Error an sending a parameter. Or device returned an error code. Kindly refer to appendix. ReturnCode .
HF_ERR_SOCKET_ERROR	1002	Socket or communication port read/write error. An error occurred during asynchronous read/write.
HF_ERR_DATA_LENGTH	1003	Data length too short, device returned an invalid data length.
HF_ERR RESPOND LENGTH	1103	Length of packet small then request
HF_ERR_HANDLE_INVALID	1004	Invalid control handler received. Invalid hComm value,
HF_ERR RESPOND ENDCHAR	1005	Error in the packet no. returned.
HF_ERR RESPOND_CRC16	1006	Error of 16-bit Cyclic Redundancy Check (CRC-16) returned.
HF_ERR_SEND_CRC16	1106	Error of 16-bit Cyclic Redundancy Check (CRC-16) set.
HF_ERR_SEND_CMD	1007	PC sends wrong order to device or device does not support this function.
HF_ERR_SEND_RW	1008	An error occurred while performing read/write to slave device.
HF_ERR_SEND_OVERLENGTH	1009	Data length transmitted exceeded max. allowed length.
HF_ERR RESPOND_NORECORDS	1010	No data was retrieved
HF_ERR RESPOND_EXCEPT	4445	An error while reading device data or records
HF_ERR_WAIT_TIMEOUT	1025	Operation timed out during asynchronous read/write.
HF_ERR_WAIT_FAILED	1026	Operation error during asynchronous read/write.
HF_ERR_WAIT_NODATA	2225	Data was not retrieved during asynchronous read/write.

# Chapter 4 HTA Series Functions

## 4.1 Communication – API (Level 0)

### 4.1.1 htaOpenChannel (Open Communication Channel)

Hardware	HTA-8x0 series, HTA-850 series, HTA-852 series, HTA-860 series
API	htaOpenChannel
Function	int __stdcall htaOpenChannel (HANDLE *hComm,char *sComm,unsigned int iPort)
Purpose	Open TCP/IP or RS-232
Arguments	<p>RS-232 port only for HTA-810/820 used For RS-232Com Port: hComm is handle value if it returns true. sComm is COM1~COM128 iPort is Baudrate. Baudrate(1200/2400/4800/9600/19200/38400)</p> <p>For TCP/IP: hComm is handle value if it returns true. TCP/IP TCP/IP handle value is actual value plus 256. In order to differentiate between Comport value and TCP/IP value.</p> <p><b>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.</b></p> <p>sComm is IP address, like 172.16.1.1 iPort is port number, like 4660</p>
Return Value	When return “HF_RET_SUCCESS” is true. Kindly refer to Chapter 3 for other return value.
Application	However make order or retrieval data; you need call this function first before you operate other functions.
Remark	Function explanation: Open RS-232 (COM1) and read device time through handle. Open TCP/IP (Port 4660) and read device time through handle. After operate success, please use htaCloseChannel to close the communication.

#### 4.1.2 htaCloseChannel (Close Communication Channel)

Hardware	HTA-8x0 series, HTA-850 series, HTA-852 series, HTA-860 series	
API	htaCloseChannel	
Function	int __stdcall htaCloseChannel (HANDLE hComm)	
Purpose	Close communication port.	
Arguments	hComm	Handle value to be closed
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application	Close communication with device	
Remark		

#### 4.1.3 hsHTA850WriteTable (Sends table to HTA-850/852/860)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850WriteTable	
Function	int __stdcall hsHTA850WriteTable (HANDLE hComm, int iTable,unsigned char *cTableData,int iTableLen,int *iReturnCode,unsigned int iTimeout)	
Purpose	Sends a table to device. Kindly refer to remarks section for the table lists.	
Arguments	hComm	Handle value returned by htaOpenChannel
	iTable	Table to be downloaded to the device. Kindly refer to remarks for description.
	cTableData	Contents of table
	iTableLen	Length of table
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (second) for the response of the device. Recommended to set 10 seconds and above.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark	Table	Description
	1	Siren Timetable (Address:0)
	2	Work Timetable (Address: 0100h)
	3	Message display timetable (Address:0512h)

#### 4.1.4 hsHTA850ReadTable (Read the table from HTA-850/852/860)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850ReadTable	
Function	int __stdcall hsHTA850ReadTable (HANDLE hComm, int iTable,unsigned char *cTableData,int *iTableLen,int *iReturnCode,unsigned int iTimeout)	
Purpose	Reads a table from the device. Kindly refer to the remarks for description.	
Arguments	hComm	Handle value returned by htaOpenChannel
	iTable	Table to be downloaded to the device. Kindly refer to remarks for description.
	cTableData	Contents of table
	iTableLen	Length of table
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (second) for the response of the device. Recommended to set 10 seconds and above.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark	Table	Description
	1	Siren Timetable (Address:0)
	2	Work Timetable (Address: 0100h)
	3	Message display timetable (Address:0512h)

#### 4.1.5 hsHTA850PollingData (Read Swiped Card Records from 850/852)

Hardware	HTA-850 series, HTA-852 series.	
API	hsHTA850PollingData	
Function	int __stdcall hsHTA850PollingData (HANDLE hComm,int iPrevRecord,stPollRecord *stRecord,int *iRecord,int *iReturnCode,unsigned int iTimeout)	
Purpose	Reads swipe card records from the device.	
Arguments	hComm	Handle value returned by htaOpenChannel
	iPrevRecord	No. of records previously retrieved
	stRecord	Structure that points to the contents of data retrieved.
	iRecord	No. of records retrieved
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return “HF_RET_SUCCESS” is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.1.6 htaPolling (Sends Polling Command to HTA-8x0)

Hardware	HTA-8x0 series	
API	htaPolling	
Function	int __stdcall htaPolling (HANDLE hComm,int iNodeID,int iPrevRecord,stPollList *stRecord,int *iRecord,int iCardType,unsigned int iTimeout)	
Purpose	Sends the value of cSendData from COM or TCP/IP to device. Waits for the response packet and checks if the returned value is true,	
Arguments	hComm	Handle value returned by htaOpenChannel
	iNodeID	Device ID
	iPrevRecord	No. of records previously retrieved
	stRecord	Structure that points to the contents of data retrieved.
	iRecord	No. of records retrieved
	iCardType	Indicates Whether the card no. is compressed or not. 0 signifies uncompressed card number 1 signifies compressed card number.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return “HF_RET_SUCESS” is true. Kindly refer to Chapter 3 for other return value.	
Application	<p>The data retrieved are decrypted data.</p> <pre>typedef struct stPollList {     int iClassCode;     int iIllegalCode; // Card type: 0 signifies invalid card.                       // 1 signifies valid card.     char cDateTime[20];// Date and time of card     char cCard[20];     char cDeviceID[10]; // Device ID } stPollList;</pre>	
Remark		

#### 4.1.7 hsHTA860PollingData (Read Swiped Card Records from HTA-860)

Hardware	HTA-860 series	
API	hsHTA850PollingData	
Function	int __stdcall hsHTA860PollingData(HANDLE hComm,int iPrevRecord,stPollRecord *stRecord,int *iRecord,int *iReturnCode,unsigned int iTimeout)	
Purpose	Reads swipe card records from the device.	
Arguments	hComm	Handle value returned by htaOpenChannel
	iPrevRecord	No. of records previously retrieved
	stPollRecord	Structure that points to the contents of data retrieved.
	iRecord	No. of records retrieved
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark	<pre>typedef struct stPollRecord {     char cDate[10]; //Date of card swiped     char cTime[10]; //Date of card swiped     char Reader; //Slave Reader ID     char InputType; //Input Type     char ASection; //Duty Section     char AClass; //Duty Shift     char EventCode; //Event Code     char Card[16]; //Card Number }stPollRecord;</pre>	

## 4.2 Device- API (Level 2)

### HTA-8X0 series

#### 4.2.1 htaGetDateTime (Read Device Date and Time)

Hardware	HTA-8x0 series	
API	htaGetDateTime	
Function	int __stdcall htaGetDateTime (HANDLE hComm,int iNodeID,char *cDate,char *cTime,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	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.2 htaSetDateTime (Set Device Date and Time)

Hardware	HTA-8x0 series	
API	htaSetDateTime	
Function	int __stdcall htaSetDateTime (HANDLE hComm,int iNodeID,char *cDate,char *cTime,unsigned int iTimeout)	
Purpose	Set device's date and time	
Arguments	iNodeID	Device ID
	cDate	Date to set Format (YYYYMMDD)+Weekday
	cTime	Time to set. Format (HHMMSS)
	hComm	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

### 4.2.3 htaGetMemoryData (Reads the SRAM Data)

Hardware	HTA-8x0 series	
API	htaGetMemoryData	
Function	int __stdcall htaGetMemoryData (HANDLE hComm,int iNodeID,char *cMemData,int *iReceiveDataLen,unsigned int iMemAddr,int iMemLen,unsigned int iTimeout)	
Purpose	Reads the SRAM data within the device	
Arguments	hComm	Handle value returned by htaOpenChannel
	iNodeID	Device ID
	cMemData	Contents of SRAM retrieved.
	iReceiveDataLen	Actual length of data of SRAM retrieved
	iMemAddr	Retrieved SRAM location
	iMemLen	Retrieved SRAM length
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application	SRAM address 0000H~1FFFH (Byte1: Lo, Byte2: Hi) Data length (1~255 Byte, 0=256 Byte)	
Remark		

#### 4.2.4 htaSetMemoryData (Write Data to SRAM)

Hardware	HTA-8x0 series	
API	htaSetMemoryData	
Function	int __stdcall htaSetMemoryData (HANDLE hComm,int iNodeID,char *cMemData,unsigned int iMemAddr,int iMemLen,unsigned int iTimeout)	
Purpose	Writes the data to SRAM	
Arguments	hComm	Handle value returned by htaOpenChannel
	iNodeID	Device ID
	cMemData	Contents of SRAM to be set
	iMemAddr	SRAM location to be set
	iMemLen	SRAM length to be set
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark	SRAM address 0000H~1FFFH (Byte1: Lo, Byte2: Hi) Data length (1~255 Byte, 0=256 Byte)	

#### 4.2.5 htaGetVersion (Reads the Device's Version)

Hardware	HTA-8x0 series	
API	htaGetVersion	
Function	int __stdcall htaGetVersion (HANDLE hComm,int iNodeID,char *cData,unsigned int iTimeout)	
Purpose	Reads device's version	
Arguments	hComm	Handle value returned by htaOpenChannel
	iNodeID	Device ID
	cData	Retrieved device's model number and ROM version. Ex: HTA820 V20
	iTimeout	Wait time (ms) for the response of the device.
	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
	Application	
Remark		

## 4.2.6 htaAddCard (Add a Single Uncompressed Card)

Hardware	HTA-8x0 series	
API	htaAddCard	
Function	int __stdcall htaAddCard (HANDLE hComm,int iNodeID,char *cCardNo,int iCardLen,unsigned int iTimeout)	
Purpose	Add a single uncompressed card	
Arguments	iNodeID	Device ID
	cCardNo	Card number. Total length of 14 characters. FF is appended if the card number does not reach 14 characters. AP sends card number12345 to device. The format upon setting onto the device becomes [31][32][33][34][35][FF][FF][FF][FF][FF][FF][FF][FF] [FF]
	iCardLen	Length of cCardNo
	hComm	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return “HF_RET_SUCESS” is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.7 htaDelCard (Delete a Single Uncompressed Card)

Hardware	HTA-8x0 series	
API	htaDelCard	
Function	int __stdcall htaDelCard (HANDLE hComm,int iNodeID,char *cCardNo,int iCardLen,unsigned int iTimeout)	
Purpose	Delete a single uncompressed card	
Arguments	iNodeID	Device ID
	cCardNo	Card number. Total length of 14 characters. FF is appended if the card number does not reach 14 characters. AP sends card number12345 to device. The format upon setting onto the device becomes [31][32][33][34][35][FF][FF][FF][FF][FF][FF][FF][FF] [FF]
	iCardLen	Length of cCardNo
	hComm	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return “HF_RET_SUCESS” is true. Kindly refer to Chapter 3 for other return value.	
Application		

#### 4.2.8 htaAddZCard (Add a Single Compressed Card)

Hardware	HTA-8x0 series	
API	htaAddZCard	
Function	int __stdcall htaAddZCard (HANDLE hComm,int iNodeID,char *cCardNo,int iCardLen,unsigned int iTimeout)	
Purpose	Add a single compress card. The card number should be digits.	
Arguments	iNodeID	Device ID
	cCardNo	Card number. Total length of 14 characters. FF is appended if the card number does not reach 14 characters. AP sends card number12345 to device. The format upon setting onto the device becomes [31][32][33][34][35][FF][FF][FF][FF][FF][FF][FF][FF] [FF]
	iCardLen	Length of cCardNo
	hComm	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return “HF_RET_SUCESS” is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.9 htaDelZCard (Delete a Single Compressed Card)

Hardware	HTA-8x0 series	
API	htaDelZCard	
Function	int __stdcall htaDelZCard (HANDLE hComm,int iNodeID,char *cCardNo,int iCardLen,unsigned int iTimeout)	
Purpose	Delete a single compressed card	
Arguments	iNodeID	Device ID
	cCardNo	Card number. Total length of 14 characters. FF is appended if the card number does not reach 14 characters. AP sends card number12345 to device. The format upon setting onto the device becomes [31][32][33][34][35][FF][FF][FF][FF][FF][FF][FF][FF] [FF]
	iCardLen	Length of cCardNo
	hComm	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
	Return Value	When return “HF_RET_SUCESS” is true. Kindly refer to Chapter 3 for other return value.
Application		
Remark		

#### 4.2.10 htaEraseFlash (Format Flash Memory)

Hardware	HTA-8x0 series	
API	httaEraseFlash	
Function	int __stdcall htaEraseFlash (HANDLE hComm,int iNodeID,unsigned int iTimeout)	
Purpose	initialize the flash memory	
Arguments	hComm	Handle value returned by htaOpenChannel
	iNodeID	Device ID
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return “HF_RET_SUCESS” is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### **4.2.11 htaDeleteAllLog (Delete All Swipe Card Records)**

Hardware	HTA-8x0 series	
API	htaDeleteAllLog	
Function	int __stdcall htaDeleteAllLog (HANDLE hComm,int iNodeID,unsigned int iTimeout)	
Purpose	Delete all swipe card records within the device	
Arguments	iNodeID	Device ID
	hComm	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return “HF_RET_SUCESS” is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### **4.2.12 htaDeleteAllCard (Delete All Valid Card)**

Hardware	HTA-8x0 series	
API	htaDeleteAllCard	
Function	int __stdcall htaDeleteAllCard (HANDLE hComm,int iNodeID,unsigned int iTimeout)	
Purpose	Delete all valid cards within the device	
Arguments	iNodeID	Device ID
	hComm	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return “HF_RET_SUCESS” is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### **4.2.13 htaRestart (Restart Device)**

Hardware	HTA-8x0 series	
API	htaRestart	
Function	int __stdcall htaRestart (HANDLE hComm,int iNodeID,unsigned int iTimeout)	
Purpose	Restart the device	
Arguments	iNodeID	Device ID
	hComm	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return “HF_RET_SUCCESS” is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

## 4.2.14 htaGetLogRecord (Read Device's Decrypted Swiped Card Record)

Hardware	HTA-8x0 series	
API	htaGetLogRecord	
Function	int __stdcall htaGetLogRecord (HANDLE hComm,int iNodeID,int iBank,stPollList *stRecord,int *iRecord,int iCardType,unsigned int iTimeout)	
Purpose	Read swipe card records and decrypt the data	
Arguments	hComm	Handle value returned by htaOpenChannel
	iNodeID	Device ID
	iBank	Retrieved blocks (0~319) from log file of device. Only one block can be retrieved at a time.
	stRecord	Structure that points to the contents of data retrieved.
	iRecord	No. of records retrieved
	iCardType	Indicates Whether the card no. is compressed or not. 0 signifies uncompressed card number 1 signifies compressed card number.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark	<p>The data retrieved are decrypted data. Retrieve records should same with Polling data.</p> <pre>typedef struct stPollList {     int iClassCode;     int iIllegalCode; // Card type: 0 signifies valid card.     char cDateTime[20];// Date and time of card     char cCard[20];     char cDeviceID[10]; // Device ID } stPollList;</pre>	

#### 4.2.15 htaGetLogData (Read Log Files of Swiped Card Records)

Hardware	HTA-8x0 series	
API	htaGetLogRecord	
Function	int __stdcall htaGetLogData (HANDLE hComm,int iNodeID,char *cLogData,int *iReceiveDataLen,int iBank,int iCompress,unsigned int iTimeout)	
Purpose	Read log files of swipe card records	
Arguments	iNodeID	Device ID
	cLogData	Contents of log files retrieved
	iReceiveDataLen	Length of log files retrieved
	iBank	Retrieved blocks (0~319) from log file of device. Only one block can be retrieved at a time.
	hComm	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.16 htaGetCardData (Read Valid Card No)

Hardware	HTA-8x0 series	
API	htaGetCardData	
Function	int __stdcall htaGetCardData (HANDLE hComm,int iNodeID,char *cCardData,int *iReceiveDataLen,int iBank,int iCompress,unsigned int iTimeout)	
Purpose	Read the valid card number which stored in the device	
Arguments	iNodeID	Device ID
	cCardData	Contents section of valid card. Contents contain the storage location of data.
	iReceiveDataLen	Length of log files retrieved
	iBank	Section within the valid card to be read. (3-191)
	hComm	Handle value returned by htaOpenChannel
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

## HTA-850/852/860 series

### 4.2.17 hsHTA850ReadParameter (Read Parameter)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850ReadParameter	
Function	int __stdcall hsHTA850ReadParameter (HANDLE hComm,unsigned char *cParaData,int *iParaLen,int *iReturnCode,unsigned int iTimeout)	
Purpose	Read device parameter	
Arguments	hComm	Handle value returned by htaOpenChannel
	cParaData	Contacts of data retrieved
	iParaLen	Length of data retrieved
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

### 4.2.18 hsHTA850WriteParameter (Set Parameter)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850WriteParameter	
Function	int __stdcall hsHTA850WriteParameter (HANDLE hComm,unsigned char *cParaData,int iParaLen,int *iReturnCode,unsigned int iTimeout)	
Purpose	Set device parameter	
Arguments	hComm	Handle value returned by htaOpenChannel
	cParaData	Contents of parameter to be set
	iParaLen	Length of parameter to be set
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.19 hsHTA850Initial (Device Initialization)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series				
API	hsHTA850Initial				
Function	int __stdcall hsHTA850Initial (HANDLE hComm,char cInitFlag,int *iReturnCode,unsigned int iTimeout)				
Purpose	Set device back to default				
Arguments	hComm	Handle value returned by htaOpenChannel			
	cInitFlag	Indicates the data to be initialized			
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.			
	iTimeout	Wait time (ms) for the response of the device.			
Return Value	When return "HF_RET_SUCCESS" is true. Kindly refer to Chapter 3 for other return value.				
Application					
Remark	Initialization Flag Description: bit0: 1 Delete all valid cards bit1:1 Initializes all tables bit2:1 Deletes all swiped cards records bit3:1 Initializes all other parameters but not included system parameters. Bit4~7: Reserved				

## 4.2.20 hsHTA850GetInfo (Read Device Version and Other Info)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series																																			
API	hsHTA850GetInfo																																			
Function	int __stdcall hsHTA850GetInfo (HANDLE hComm,unsigned char *clnfoData,int *iInfoLen,int *iReturnCode,unsigned int iTimeout)																																			
Purpose	Read device version and other information																																			
Arguments	hComm	Handle value returned by htaOpenChannel																																		
	clnfoData	Buffer of data retrieved																																		
	iInfoLen	Length of data retrieved																																		
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.																																		
	iTimeout	Wait time (ms) for the response of the device.																																		
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.																																			
Application	Contents of data to be transmitted																																			
Remark	<table border="1"> <thead> <tr> <th>Length</th> <th>Description</th> <th>Address</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Version type</td> <td>0</td> </tr> <tr> <td>1</td> <td>Major version</td> <td>1</td> </tr> <tr> <td>1</td> <td>Minor version</td> <td>2</td> </tr> <tr> <td>1</td> <td>Beta version</td> <td>3</td> </tr> <tr> <td>1</td> <td>Year</td> <td>4</td> </tr> <tr> <td>1</td> <td>Month</td> <td>5</td> </tr> <tr> <td>1</td> <td>Day</td> <td>6</td> </tr> <tr> <td>4</td> <td>No of valid card stored</td> <td>7</td> </tr> <tr> <td>4</td> <td>No of swipe card records stored</td> <td>11</td> </tr> <tr> <td>1</td> <td>Reserved</td> <td>15</td> </tr> </tbody> </table> <p>Definition:      Version type: 0x00 means Access control function+ Time attendance function.      Major version: Major version of ROM file      Minor version: Minor version of ROM file      Beta version: 0x00 means released version          0x01 means Beta 1          0x02 Beta 2 and so on      Year/Month/Day: ROM file release date (BCD format)</p>			Length	Description	Address	1	Version type	0	1	Major version	1	1	Minor version	2	1	Beta version	3	1	Year	4	1	Month	5	1	Day	6	4	No of valid card stored	7	4	No of swipe card records stored	11	1	Reserved	15
Length	Description	Address																																		
1	Version type	0																																		
1	Major version	1																																		
1	Minor version	2																																		
1	Beta version	3																																		
1	Year	4																																		
1	Month	5																																		
1	Day	6																																		
4	No of valid card stored	7																																		
4	No of swipe card records stored	11																																		
1	Reserved	15																																		

#### 4.2.21 hsHTA850InsertMultiUserRecord (Add Multiple Valid Cards)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850InsertMultiUserRecord	
Function	int __stdcall hsHTA850InsertMultiUserRecord (HANDLE hComm, int iRecord,struct_CardFormat * stRecord, int *iReturnCode,unsigned int iTimeout)	
Purpose	Add multiple valid cards	
Arguments	hComm	Handle value returned by htaOpenChannel
	iRecord	No of new card to be add
	stRecode	List of new card to be add
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark	<pre> typedef struct struct_CardFormat {     char Card[16];     char Stay[2];     char DisplayMsg[16]; }struct_CardFormat; </pre>	

#### 4.2.22 hsHTA850DeleteUserRecord (Delete a Card)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850DeleteUserRecord	
Function	int __stdcall hsHTA850DeleteUserRecord (HANDLE hComm, Int CardLen,char *cCardNo,int *iReturnCode,unsigned int iTimeout)	
Purpose	Delete a card	
Arguments	hComm	Handle value returned by htaOpenChannel
	CardLen	Length of card number
	cCardNo	Card number
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.23 hsHTA850QueryUserRecord (Query Single Card's Auth)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850QueryUserRecord	
Function	int __stdcall hsHTA850QueryUserRecord (HANDLE hComm,Int CardLen,char *cCardNo,unsigned char * cCardFormatData,int * iCardFormatLen,int *iReturnCode,unsigned int iTimeout)	
Purpose	Query single card's authorization	
Arguments	hComm	Handle value returned by htaOpenChannel
	CardLen	Length of card number
	cCardNo	Card number
	cCardFormatData	Contents of data retrieved
	iCardFormatLen	Length of data retrieved
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.24 hsHTA850DeleteAllUserRecord (Delete All Valid Cards)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850DeleteAllUserRecord	
Function	int __stdcall hsHTA850DeleteAllUserRecord (HANDLE hComm,int *iReturnCode,unsigned int iTimeout)	
Purpose	Delete all valid cards	
Arguments	hComm	Handle value returned by htaOpenChannel
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.25 hsHTA850SetEEPROM (Set EEPROM Parameter)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850SetEEPROM	
Function	int __stdcall hsHTA850SetEEPROM (HANDLE hComm,unsigned char cEEData,int iEELen,int *iReturnCode,unsigned int iTimeout)	
Purpose	Set other parameters to EEPROM	
Arguments	hComm	Handle value returned by htaOpenChannel
	cEEData	Contents of data to be set
	iEELen	Length of data
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

## 4.2.26 hsHTA850ReadEEPROM (Read EEPROM's Parameter)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850ReadEEPROM	
Function	int __stdcall hsHTA850ReadEEPROM (HANDLE hComm,unsigned char *cEESendData,int iEESendLen,unsigned char *cEEReceiveData,int *iEEReceiveLen,int *iReturnCode,unsigned int iTimeout)	
Purpose	Read parameters from EEPROM	
Arguments	hComm	Handle value returned by htaOpenChannel
	cEESendData	Contents of data to be transmitted
	iEESendLen	Length of data to be transmitted
	cEEReceiveData	Contents of data retrieved
	iEEReceiveLen	Length of data retrieved
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.27 hsHTA850WriteTime (Set Device Date and Time)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850WriteTime	
Function	int __stdcall hsHTA850WriteTime (HANDLE hComm,char *cDate,char *cTime,int *iReturnCode,unsigned int iTimeout)	
Purpose	Set device time	
Arguments	hComm	Handle value returned by htaOpenChannel
	cDate	Date to set Format (YYYYMMDD)+Weekday
	cTime	Time to set. Format (HHMMSS)
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.28 hsHTA850Read Time (Read Device Date and Time)

Hardware	HTA-850 series, HTA-852 series, HTA-860 series	
API	hsHTA850ReadTime	
Function	int __stdcall hsHTA850ReadTime (HANDLE hComm,int iGCUID, char *cDate,char *cTime,int *iReturnCode,unsigned int iTimeout)	
Purpose	Retrieve device current time and date	
Arguments	hComm	Handle value returned by htaOpenChannel
	cDate	Date retrieved. Format (YYYYMMDD)+Weekday
	cTime	Time retrieved. Format (HHMMSS)
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.29 hsHTA850SetMifareReader(Set Mifare Reader Parameter)

Hardware	HTA-850PM, HTA-852PMF, HTA-860PM. HTA-860PMF	
API	hsHTA850SetMifareReader	
Function	int __stdcall hsHTA850SetMifareReader (HANDLE hComm,unsigned char *cData,int *iLen,int *iReturnCode,unsigned int iTimeout)	
Purpose	Set Mifare reader's parameter	
Arguments	hComm	Handle value returned by htaOpenChannel
	cData	Contents of data to be transmitted
	iLen	Length of data to be transmitted
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.30 hsHTA850InsertMultiUserFingerPrinter2 (Add New Cards)

Hardware	HTA-852 series, HTA-860PEF, HTA-860PMF, HTA-860F				
API	hsHTA850InsertMultiUserFingerPrinter2				
Function	int __stdcall hsHTA850InsertMultiUserFingerPrinter2(HANDLE hComm,int CardLen,int MsgLen,int iRecord,struct_FingerPrinterFormat2 *stRecord,int *iReturnCode,unsigned int iTimeOut)				
Purpose	Add card to fingerprint machine				
Arguments	hComm	Handle value returned by htaOpenChannel			
	iRecord	No of new card to be add			
	stFPRecord	List of new card to be add			
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.			
	iTimeout	Wait time (ms) for the response of the device.			
Return Value	When return "HF_RET_SUCCESS" is true. Kindly refer to Chapter 3 for other return value.				
Application					
Remark	<pre> typedef struct struct_FingerPrinterFormat2 {     char Card[16];     char Stay[2];     char DisplayMsg[16];     unsigned char FingerPrinter1[386];     unsigned char FingerPrinter2[386]; }struct_FingerPrinterFormat2; </pre>				

#### 4.2.31 hsHTA850QueryUserFingerPrinter2 (Query Single Auth)

Hardware	HTA-852 series, HTA-860PEF, HTA-860PMF, HTA-860F	
API	hsHTA850QueryUserFingerPrinter2	
Function	int __stdcall hsHTA850QueryUserFingerPrinter2(HANDLE hComm,int CardLen,char *cCardNo,unsigned char *cFingerPrinterData1,unsigned char *cFingerPrinterData2,int *iCardFormatLen,int *iReturnCode,unsigned int iTimeOut)	
Purpose	Query single authorization	
Arguments	hComm	Handle value returned by htaOpenChannel
	CardLen	Length of card number
	cCardNo	Card number
	cFingerPrinterData1	First fingerprint template retrieved
	cFingerPrinterData1	Second fingerprint template retrieved
	iCardFormatLen	Length of data retrieved
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.32 hsHTA850QueryMasterFP (Query Master Fingerprint)

Hardware	HTA-852 series, HTA-860PEF, HTA-860PMF, HTA-860F	
API	hsHTA850QueryMasterFP	
Function	int __stdcall hsHTA850QueryMasterFP(HANDLE hComm, unsigned char *cFingerPrinterData1,unsigned char *cFingerPrinterData2,int *iReturnCode,unsigned int iTimeOut)	
Purpose	Query Master fingerprint	
Arguments	hComm	Handle value returned by htaOpenChannel
	CardLen	Length of card number
	cFingerPrinterData1	First fingerprint template retrieved
	cFingerPrinterData1	Second fingerprint template retrieved
	iCardFormatLen	Length of data retrieved
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

#### 4.2.33 hsHTA850UpdateMasterFP (Change Master Fingerprint)

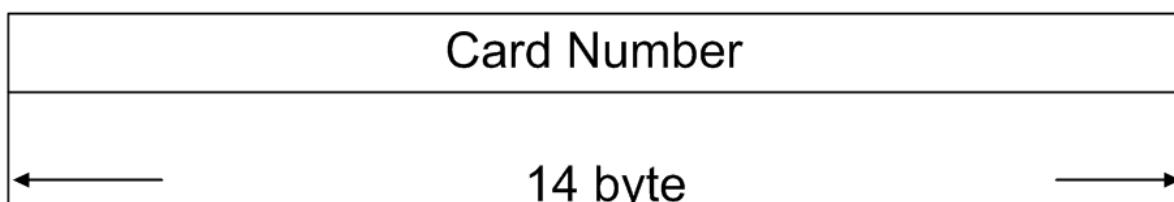
Hardware	HTA-852 series, HTA-860PEF, HTA-860PMF, HTA-860F	
API	hsHTA850UpdateMasterFP	
Function	int __stdcall hsHTA850UpdateMasterFP(HANDLE hComm, unsigned char *cFingerPrinterData1,unsigned char *cFingerPrinterData2,int *iReturnCode,unsigned int iTimeOut)	
Purpose	Change Master fingerprint	
Arguments	hComm	Handle value returned by htaOpenChannel
	cFingerPrinterData1	First fingerprint template retrieved
	cFingerPrinterData2	Second fingerprint template retrieved
	iReturnCode	Error code returned by the device. Kindly refer to Appendix for description.
	iTimeout	Wait time (ms) for the response of the device.
Return Value	When return "HF_RET_SUCESS" is true. Kindly refer to Chapter 3 for other return value.	
Application		
Remark		

## Chapter 5 Appendix

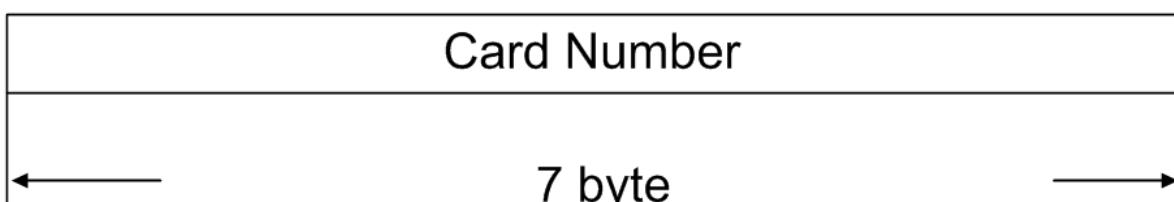
### Appendix 1: HTA-8x0 series Description

#### 5.1.1 Valid Card's Format

**Uncompressed:** Length is 14 byte, each character occupies 1 byte, appends 1 to every bit if there's no data.

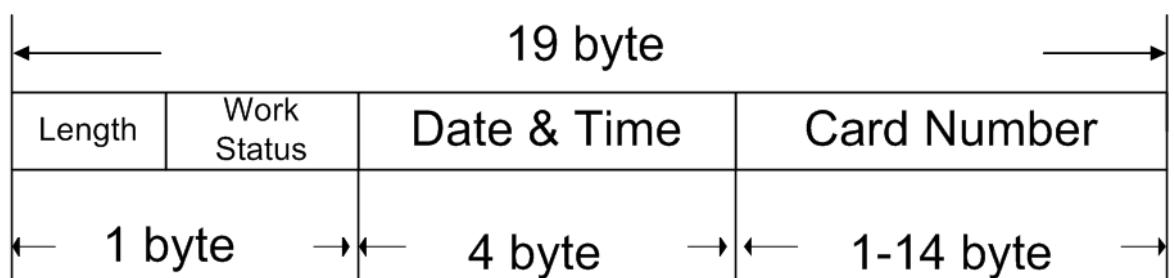


**Compressed:** Length is 7 byte, stores a digit in the first and last 4 bit of every byte, appends 1 to every bit if there is no data.



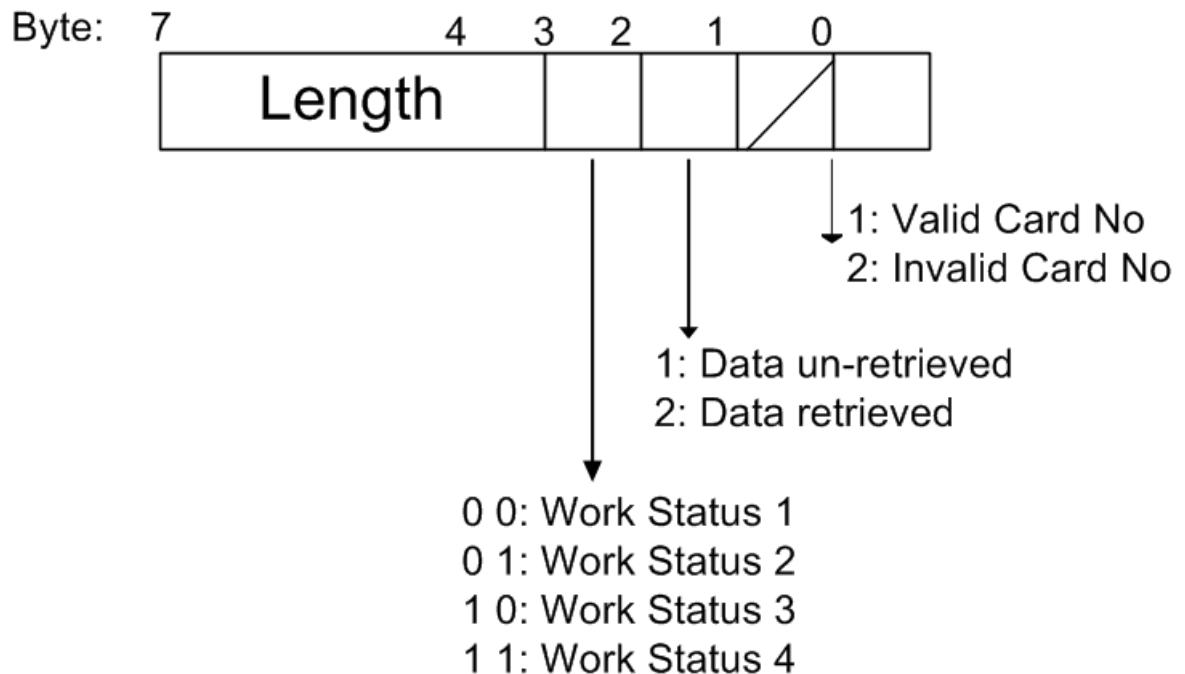
#### 5.1.2 Swiped Card Records

**Uncompressed:**

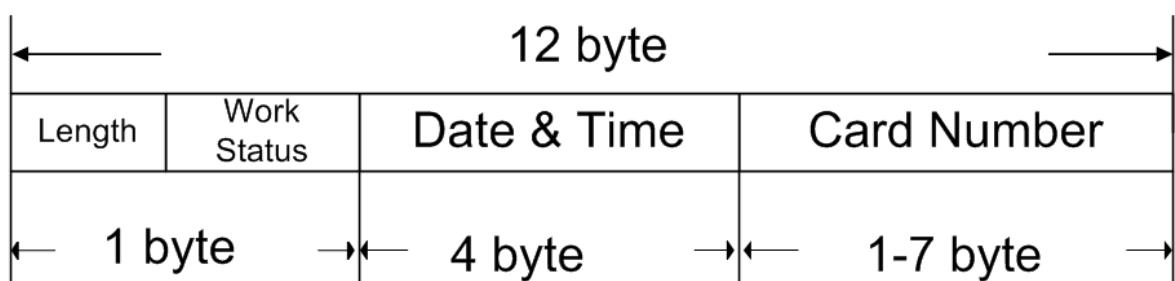


Length: 4 bit defines card number's length.

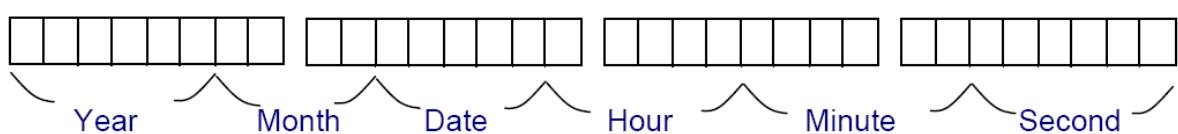
Work Status: 4 bit



**Compressed:**



### 5.1.3 Date and Time Format



### 5.1.4 SRAM Memory Allocation

Length	Address	Description	Default
1	0000H	Device's code, please do not modify it.	'H"
1	0001H	Device's Model, 1:810, 2:820, 3:830, 4: 840	Depends on device
1	0002H	Save card no. (Index 1~128), indicates the starting location of the card number to save.	01 H
1	0003H	Length of the card no to save 0~14, 0 signifies the max length that could be saved.	00 H
1	0004H	Do not save Re-swipe Card Check Time (sec)	3C H
1	0005H	Overwrite data or not 0= Overwrite, 1=Do not overwrite	1
1	0006H	Lock keypad 0=Unlock, 1=Lock	0
1	0007H	'V'	76 H
1	0008H	Major Version	02 H
1	0009H	Minor Version	00 H
1	000AH	Reserved	00 H
1	000BH	Reserved	00 H
1	000CH	Reserved	00 H
16*5	000DH	Set work time timetable 16 sets*5byte byte 1: Time start (Hour) byte 2: Time Start (minute) byte 3: Time End (Hour) byte 4: Time End (Minute) byte 5: Work status 1~4  HTA_820/830/840 will change the LED to light as well as the storage location of the work status records, accordingly to the settings in the work in timetable. If the program has located the time schedule, it will not search for another time schedule.	
32*4	005DH	Set alarm timetable 32 sets*4byte 1 <sup>st</sup> byte: Time Start (Hour) 2 <sup>nd</sup> byte: Time Start (Minute) 3 <sup>rd</sup> byte: Prolong Time (Minute) 4 <sup>th</sup> byte: Prolong Time (Seconds)  HTA 8x0 will activate the relay according to the alarm time set. If the program has located the time schedule, it will not search for another time schedule.	
1	00DDH	Baudrate 1~6 =1200,2400,4800,9600,19200,38400	4
2	00DEH	Total number of valid cards recorded in Flash memory. byte 1: Hi byte 2: Lo	0,0

2	00E0H	Reserved	
2	00E2H	Total number of swipe card records in Flash memory. byte 1: Hi byte 2: Lo	0,0
2	00E4H	Reserved	
1	00E6H	Compress card number or not. 0 = Uncompress, 1 = Compress Please refer to card number's format	1
1	00E7H	Bit7 = 1 Save invalid card number	1
1	0xE8	Work Status	1
1	0xE9	Default ID	1
32	0xF0H	Set alarm timetable, 32sets (byte) Activation of alarm relay on the specified day of the week. Every byte's bit0 is for Monday, bit1 is for Tuesday, ....bit 6 is Sunday, bit7 is reserved. 0 means holiday timetable is activated, 1 means deactivated.	
2	0x110	Reserved	
1	0x112	Daylight saving function 00=Deactivated (Default) 01= Advance 02= Delay	00
1	0x113	Adjust time-Hour 00H~23H represents 0~23 hour	00
1	0x114	Adjust time-Minute 00H~59H represents 0~59 Minute	00
1	0x115	Starting Month, range 01H~12H, represents 1~12 month.	01
1	0x116	Starting Date, range 01H~31H, represents 1~31 days.	01
1	0x117	Starting Hour, range 00H~23H, represents 0~23 hours.	00
1	0x118	Starting Minute, range 00H~59H, represents 0~59 minutes.	00
1	0x119	Ending Month, range 01H~12H, represents 1~12 month.	01
1	0x11A	Ending date, range 01H~31H, represents 1~31 days.	01
1	0x11Bb	Ending Hour, range 00H~23H, represents 0~23 hours.	00
1	0x11C	Ending Minute, range 00H~59H, represents 0~59 minutes.	00
1	0x11D	States whether daylight saving function has been activated. 0= Disabled 1= Enabled	00
2	0x11E	Device's code, please do not modify it.	'H', '2'
	0x200	Reserved	

## 5.1.5 LED Message Display Communication Format

(Baudrate: 9600, N, 8, 1)

### Send:

Data = 0xED 0xED 0xED 0xFF 0x00 TXT(8 byte) 0x00

Explanation:

TXT: 8 byte big 5 or ASCII code

### Response:

LED message display does not have response.

## 5.1.6 DVR Communication Format

(Baudrate: 9600, N, 8, 1)

### Send:

Data structure as below:

Message	Size	Description
Start character of message	1 byte	0xF6, Every message must start from this character.
Length	1 byte	HEX. Do not include start and end character and length.
Device ID	4 byte	HEX. Send lo byte first and then hi byte, Fixed send 0x01, 0x00, 0x00, 0x00
Recorder of card swiped	2 byte	Lo byte recorder ID Hi byte will be fixed to send 0x00
Date and time of card swiped	7 byte	BCD code for Year, Month, date, Hour, Minute and Second. ( Ex. 2006/04/12 08:05:55 will be 0x20, 0x06, 0x04, 0x12, 0x08, 0x05, 0x55)
Card number	1~13 byte	Card number
Check Sum	1 byte	Total amount of data. Do not include start and end character.
End character of message	1 byte	0xF7, Every message must end by this character.

### Response:

DVR does not have response.

Explanation:

Sample 1

Recorder ID, inside swipe card reader =0, Date and time of card swiped = 2006/05/30 15:20:45, Card No=0000163003(10 digits)

Transmit data:

0xF6,0x18,0x01,0x00,0x00,0x00,0x00,0x00,0x20,0x06,0x05,0x30,0x15,0x20,0x45,  
0x30,0x30,0x30,0x30,0x31,0x36,0x33,0x30,0x30,0x33,0xDB,0xF7

Sample 2

Recorder ID, outside swipe card reader =1, Date and time of card swiped = 2005/12/31 09:11:15, Card No=00925121(8 digits)

Transmit data:

0xF6,0x16,0x01,0x00,0x00,0x00,0x01,0x00,0x20,0x05,0x12,0x31,0x09,0x11,0x15,  
0x30,0x30,0x39,0x32,0x35,0x31,0x32,0x31,0x43,0xF7

### 5.1.7 Printer Communication Format

(Baudrate: 9600, N, 8, 1)

**Send:**

Data = Date:2006/12/27

Time:08:00:55

Shift: 1

ID:0000147258

**Response:**

Printer does not have response.

## Appendix 2: HTA-850/852 series Description

### 5.2.1 System Parameter

byte 1	byte 2	.....	byte 12
--------	--------	-------	---------

byte 1:	Set maximum total amount of valid card. 0~1: (Default: 0x0) 0x0: Do not have valid card. It signifies all cards are valid. 0x1: 10,000 Valid cards.
byte 2:	Set maximum total amount of swipe card record. 0~3: (Default: 0x3) 0x3: 30,000 swipe card records
byte 3:	Length of card number in the Card No Division. 1~16 byte: (Default: 0x10) 0x1~0x10: Length of card number in the Card No Division. All cards' length should be same. Note: this parameter also use in the swipe card records format.
byte 4:	Maximum length of message display division. 0~16 byte; (Default: 0x10) 0 signifies do not have message display division. To display message division should end by 0x00.
byte 5~9:	User-define 1~5 length. (Default: 0x0) Maximum size is 8 byte per division. Note: this parameter also use in the swipe card records format.
Byte 10~12:	Reserved ( Default: 0x0)

## 5.2.2 Valid Card Format

### Standard Model: HTA-850PE/PM

Card Number	Reserved	Message Display
← 16 byte →	← 2 byte →	← 16 byte →

1. Card Number: 16 byte, FF is appended if the card number does not reach 16 characters. 0x00
2. Reserved: 2 byte, input 0x00
3. Message Display: 16 byte, Message will display in the LCD screen after card swiped. 0x20 is appended if the message does not reach 16 characters.

### Fingerprint Model: HTA-852PEF/PMF

Card No	Status	Message Display	Fingerprint Index
← 16 byte →	← 2 byte →	← 16 byte →	← 4 byte →

1. Card Number: 16 byte, FF is appended if the card number does not reach 16 characters. 0x00
2. Status: 2 byte

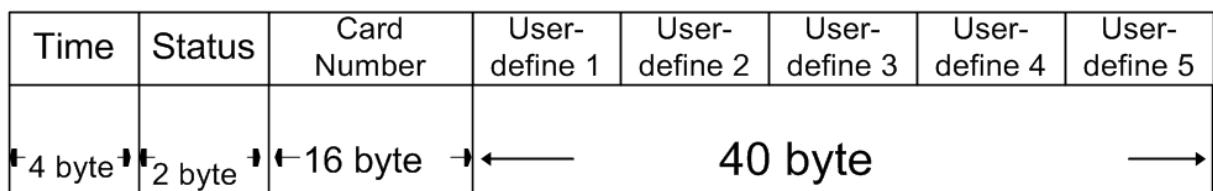
First byte: 0x00 signifies normal, 0x01 signifies add cards from device.

Second byte:

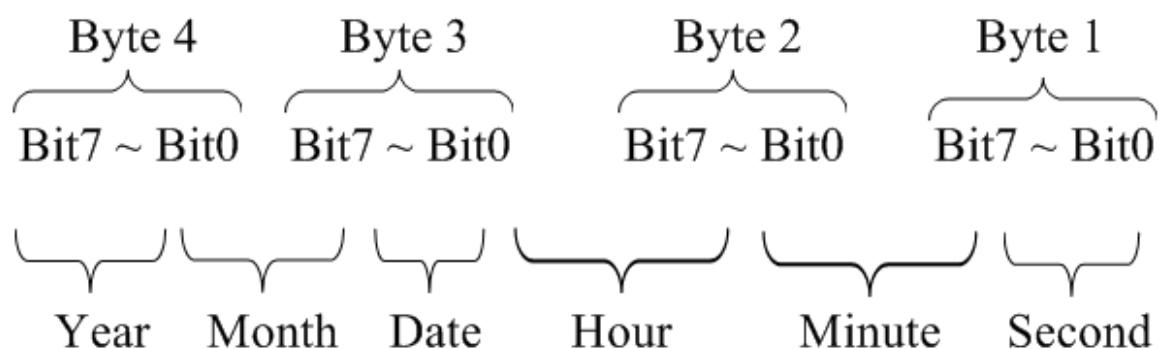
bit0	0: Do not need fingerprint 1: Request fingerprint
bit1	0: Download card number by software 1: Add card numbers by device

3. Message Display: 16 byte, Message will display in the LCD screen after card swiped. 0x20 is appended if the message does not reach 16 characters.
4. Fingerprint Index: 4 byte. Input 0xFF (Index saves in the module, Lo~Hi)

### 5.2.3 Swiped Card Records Format Description



1. Time (4byte) byte1, byte2, byte3, byte4



2. Status: 2 byte

Byte1: (Bit7 = 1)

bit 6~bit 4	Bit 3~bit 1	bit 0
0= DUTY ON	0= Input from keypad	0=Reader 1 (Inside Reader)
1= DUTY OFF	1= Input from T2 reader	0=Reader 1 (Outside Reader)
2= BREAK OUT	2= Input from Wiegand reader	
3= BREAK IN	3= Input from Mifare reader	
4= OVR_T_STR	4= Input from fingerprint	
5= OVR_T_END	5~7= Reserved	
6~7= Reserved		

Byte 2:

Reserved (bit 7~bit 4)	bit 3~ bit 0
	0= Valid card swipe
	1= Card number does not exist
	2= Error in the length of card number
	3= Invalid card number (Non-numeric character detected)
	4= Mifare write error
	5= Check fingerprint failed
	6= Blacklist
	7= Invalid time schedule
	8= Reserved
	9= Fingerprint Match
	10= Master card
	11= Invalid master card
	12= Add card from fingerprint recorder
	13= Delete card from fingerprint recorder
	14= Reserved
	15= Reserved

3. Card number: Length of card number will follow byte 3 of system parameter.
- (a) Once length is longer than pre-set length, system will cut redundant character, and only display pre-set length.
  - (b) Once length is shorter than pre-set length, 0x00 will be appended to make length is same with pre-set length.
4. User-defind1~5: Maximum has 5 divisions. 8 byte per division, length will follow byte 5~9 of system parameter. Once system parameter is 0, it signifies do not need reserve those divisions.

### 5.2.4 Memory Allocation

Total: 128 bytes

Length	Address (HEX)	Description	Default	Remark
1	00H	Swiped card records	0x02	0= Do not record 1= Overwrite record 2= Do not overwrite record
1	01H	Reserved	0x01	
16	02H	Master Card	"30191000",0x00 ,0x00, 0x00,0x00,0x00, 0x00, 0x00,0x00	Master card, appends 0x00 if length did not reach 16.
1	12H	Language Selection	0x00	0= English 1=Traditional Chinese 2. Simplified Chinese
1	13H	LCD date format	0x00	0=YYYY/MM/DD 1=MM/DD/YYYY 2=DD/MM/YYYY
1	14H	Reader Type	0x00	Standard Model: 0= Mifare RS-232 1= T2 2= Wiegand  Fingerprint Model: 0= 852PMF 1= 852PEF 2= 852F 3= 852PMFC
1	15H	Invalid card records	0x01	0= Do not record 1=Record
2	16H	Relay 1 status settings: bit0~bit11 = Pulse time Pulse=0: refer to other relay action Pulse = 1~4095: indicates 0.1~409.5 sec  bit12~bit14: Reserved	0x28,0x00	Lo byte, Hi byte 0x0028=40 40*0.1 sec=4 sec

		bit15= other relay action (Pulse time =0) 0=Toggle 1= Latch		
1	18H	Check card number length	0x00	0= Ignore 1~16= Length of card number to check
1	19H	Check digit of card number	0x00	0= Ignore 1= Inspect
1	1AH	Retrieve valid card number (Index)	0x01	1-16
1	1BH	Retrieve valid card number (Length)	0x00	0= get the length of card number swiped 1-16= Length specified
1	1CH	Keypad Settings: bit0= keypad status (0= Disable, 1= Enable)  bit1~3= Reserved  bit4~5= keypad backlight (0= Auto, 1= Enable, 2= Disable)  bit6-7= Reserved	0x01	Default value: Keypad status: Enable  Keypad backlight: Auto
1	1DH	Reserved	0x00	
1	1EH	Re-swi pe card check time	0x00	0= Disabled 1~255= Inspection time 1~255 seconds
1	1FH	Comport 4 RS-232 output. Refer to Chapter 5.1.5	0x00	0= Disabled 1= LED Display 2= DVR 3= Printer 4= Other
10	20H	Mifare reader settings	0x00,0x00,0x00, 0x80, 0xFF,0xFF,0xFF ,	Refer to Chapter 5.2.9 CMD30 (P0-P3, K0-K5)
2	2AH	Lunch break (min) (deducts the time indicated from the total work hours)	0x00,0x00	Lo byte, hi byte

1	2CH	LCD display settings bit0= LCD message display (0=Disable, 1=Enable)  bit1~3= Reserved  bit4~5= LCD backlight (0=Auto, 1=Enable, 2=Disable)  bit6~7= Reserved	0x10	Default value: LCD message display: Disable  LCD backlight: Enable
1	2DH	March Threshold	0x05	0= Auto 1~5= Security A~E
2	2EH	Reserved	0x00,0x00	
6	30H	Reserved	All are 0xFF	
1	36H	Work shift auto resume	0x00	0= Disable 1= Enable
1	37H	Daylight saving settings bit0= Daylight saving options (0= Advance time, 1= Delay time)  bit1= Adjustment status (0= Unadjusted, 1= Adjusted)  bit2= Enable/disable daylight saving function (0=Disable, 1=Enable)  Bit3~7= Reserved	0x00	Default: Disable
4	38H	Daylight saving function time start	0x00,0x00,0x00,0x00	Month, Date, Hour, Min, BCD format.
4	3CH	Daylight saving function time End	0x00,0x00,0x00,0x00	Month, Date, Hour, Min, BCD format.
2	40H	Daylight saving function adjust time	0x00,0x00	Hour, Min, BCD format.
1	42H	Error frequency	0x00	Error processing
1	43H	Error alarm frequency bit0~5= Frequency (0=Disable, 1~63= No of frequency).  bit6= Deactivate relay3	0x00	Error processing

		mode (0= Valid card, 1=Master card)  bit7= Relay 3 activation (0=Disable, 1=Enable)		
2	44H	Reserved	0x28,0x00	
2	46H	Relay 3 status settings: bit0~11 = Pulse time Pulse=0: refer to other relay action Pulse = 1~4095: indicates 0.1~409.5 sec  bit12~bit14: Reserved  bit15= other relay action (Pulse time =0) 0=Toggle 1= Latch	0x00,0x80	Latch
1	48H	Audio Settings bit0~2=Volume (Max 0~7 min)  bit3= Audio prompt (0=Disable, 1=Enable)  bit4= Siren output (0=Disable, 1=Enable)  bit5~7= Reserved	0x0B	
1	49H	Fingerprint Settings bit0= 1:1 mode (0=Auto, 1= 1:1 mode)  bit1= Master card can use fingerprint. (0= Disable, 1= Enable)  bit2~7= Reserved	0x00	
1	4AH	Previous 100 sets holiday schedule	0x08	BCD code
5	4BH	Reserved		
10	50H	Write records onto Card function	0x05,0x06,0x00, 0x00,0xFF,0xFF, 0xFF,0xFF, 0xFF, 0xFF	Refer to Chapter 5.2.9 CMD30 (w0-w3, y0-y5)
38	5AH	Reserved		

## 5.2.5 Siren Timetable

(Address 0: Total 32\*5byte=160 bytes)

Group	Action time (2 byte)	Duration (2 byte)	Week date(1 byte)
1 <sup>st</sup> Address:0000H	HH:MM	MM:SS	bit7~0= x Sun, Sat, Fri, Thu, Wed, Tue, Mon
2 <sup>nd</sup> Address:0005H	HH:MM	MM:SS	bit7~0= x Sun, Sat, Fri, Thu, Wed, Tue, Mon
3 <sup>rd</sup> Address:000AH	HH:MM	MM:SS	bit7~0= x Sun, Sat, Fri, Thu, Wed, Tue, Mon
...	...	...	...
...	...	...	...
...	...	...	...
32 <sup>nd</sup> Address: 009BH	HH:MM	MM:SS	bit7~0= x Sun, Sat, Fri, Thu, Wed, Tue, Mon

1. Action time: 2 byte, First byte for hour, second byte for minute

Example: 13:45 → 0x13, 0x45 (BCD format)

2. Duration: 2 byte, First byte minute, second byte for second.

Example: 00:12→ 0x00, 0x12 (BCD format)

3. Week date: 1 byte

bit7: Reserved, bit6~0: Sun, Sat, Fri, Thu, Wed, Tue, Mon

4. Week date: 1 byte

bit7: Reserved, bit6~0: Sun, Sat, Fri, Thu, Wed, Tue, Mon

0= Disable, 1= Enable

## 5.2.6 Work Time Timetable

(Address 256: Total 32\*4byte=128 bytes)

Group	Time (2 byte)	Work Status (2 byte)	Work Shift(1 byte)
1 <sup>st</sup> Address:0100H	HH:MM	Class	Shift
2 <sup>nd</sup> Address:0104H	HH:MM	Class	Shift
3 <sup>rd</sup> Address:0108H	HH:MM	Class	Shift
...	...	...	...
...	...	...	...
...	...	...	...
32 <sup>nd</sup> Address: 017CH	HH:MM	Class	Shift

1. Time: 2 byte, First byte for hour, Second byte for minute.

Example: 13:45 → 0x13, 0x45 (BCD format)

18:23 → 0x18, 0x23

2. Work Status

0=DUTY ON	1=DUTY OFF
2=BREAK OUT	3=BREAK IN
4=OT_START	5=OT_END

3. Work shift: 1 byte

0→ Hide

1→ Display the shift number.

### 5.2.7 Message Display

(Address 512: Total 16\*16byte=256 bytes)

Group	Message	LCD Display Length	Remark
1	DUTY ON	10	Duty on
2	DUTY OFF	10	Duty off
3	BREAK OUT	10	Away
4	BREAK IN	10	Returned
5	OT_START	10	Overtime starts
6	OT_END	10	Overtime end
7	HUNDURE	16	Message displayed during ready status
8	TEST1	16	User-define message
9	TEST2	16	User-define message
10			Reserved
11			Reserved
12			Reserved
13			Reserved
14			Reserved
15			Reserved
16			Reserved

If message to be displayed did not reach the expected length, kindly append 0X20.

## 5.2.8 Index of Fingerprint machine

(Address 2048~10240, T/ byte)

Reserve 0~8191 fingerprint index, please do not download any data to those address.

0xFF: Signifies the index is reserved

0x0F: Signifies the index is under working

0x00: Signifies the index has been used and deleted.

## 5.2.9 CMS:30 Mifare Settings

Send:

Data = C0+P0+P1+P2+P3+K0+K1+K2+K3+K4+K5

+w0+w1+w2+w3+y0+y1+y2+y3+y4+y5 → Hexadecimal code format

C0: 0x00 is the Mifare reader's command which similar with write table 54.

For P0~P3, kindly refer to table below:

P0	Select read card type	0x00 (Default)	0x00 → Read series no of Mifare card (byte2~byte10 ignore) 0x01 → Use key A 0x02 → Use key B
P1	Retrieve valid card	0x10 (Default)	bit4~7: Index of retrieve valid card bit0~3: Length of retrieve valid card
P2	Compress or Uncompress selection	0x00 (Default)	0x00 → Uncompress card no 0x01 → Compress card no
P3	Read block	0x00 (Default)	0x00~0x3F

K0~K5 are for keys when block read.

For w0~w3, kindly refer to table below: (C0 is 0x70 or 0x00, the setting of w0~y5 are same)

w0	Write swiped card. Records onto card: index block	0x00~0x3F	0x00 → Disable (byte2~byte10 ignore) 0x01~0x3F → Write swipe card records on to card index block. (w0 should be the first block of the sector)
w1	Write swiped card.	0x00~0x3F	0x00~0x3F → Write swiped card.

	Records onto card: Terminating block		Records onto card: Terminating block (w1 must big than w0 and must input it in last block (password block) of the sector)
w2	Key type	0x00 or 0x80	0x00 → Key A 0x80 → Key B
w3	Reserved	0x00	

y0 ~ y5 the key value needed in order to write the records onto the block.

Response:

status=0 : OK

Data= NULL

Status=1 : Error

Data=Error Code (2 byte)

Note: 1. The block for P1 and w0 should be different.

2. Once write swiped card records onto card function selection, kindly refer to table below:

Block 0	Serial number of Mifare card	
...		
...		
P1 Block	Card No (16 byte)	
...		
...		
w0 Block	Index Block for saving swipe card record (2 Byte)	
w0 Block + 1 Start Block for saving swipe card record		
...		
...		
w1 Block		
Terminating Block for saving swipe card record  (must input it in third block of the sector)		Swipe card records occupy at least a sector. 8 records=2 Blcok*4 records Swipe card records occupy maximum 9 sectors. 104 records=(2 Block*4 records) +(8 Sector*12 records)

Example:

Swipe card record write on to card Block 8~Block15

W0= 0x08, w1= 0x0F

Index Block for saving swipe card record should set 0x09 0x00

0x09→ Block for saving swipe card record

0x00→ What record which is stored in the block(0x00~0x03)

### 5.2.10 Error Code List

Hex Code	Description
0x0001	Invalid length of card number
0x0002	Allocated memory space for card number exceeded. Length of valid card number mismatched.
0x003	Exceeded the max number of valid cards.
0x0005	Wrong system parameter value
0x0006	Card number does not exist
0x0007	Table length transmitted is too long
0x0008	Table length retrieved is too long
0x0009	Invalid data length
0x000F	Error in the table transmitted to the device
0x0013	Unable to write in the Flash memory
0x0030	Failed to set Mifare reader settings
0x0031	Fingerprint recorder is currently busy and unable to set data
0x0032	Unable to enroll fingerprint
0x0033	Unable to delete fingerprint
0x0034	Reader type error
0x0035	Incorrect Mifare reader settings
0x0036	Failed to read the fingerprint module

## Appendix 3: HTA-860 series Description

### 5.3.1 Valid Card Format

#### Standard Model: HTA-860PE/PM

Card Number	Reserved	Message Display
← 16 byte	→ 2 byte	← 16 byte →

1. Card Number: 16 byte, FF is appended if the card number does not reach 16 characters. 0x00
2. Reserved: 2 byte, input 0x00
3. Message Display: 16 byte, Message will display in the LCD screen after card swiped. 0x20 is appended if the message does not reach 16 characters.

#### Fingerprint Model: HTA-860PEF/PMF

Card No	Status	Message Display	Fingerprint Index
← 16 byte	→ 2 byte	← 16 byte →	← 4 byte →

1. Card Number: 16 byte, FF is appended if the card number does not reach 16 characters. 0x00
2. Status: 2 byte

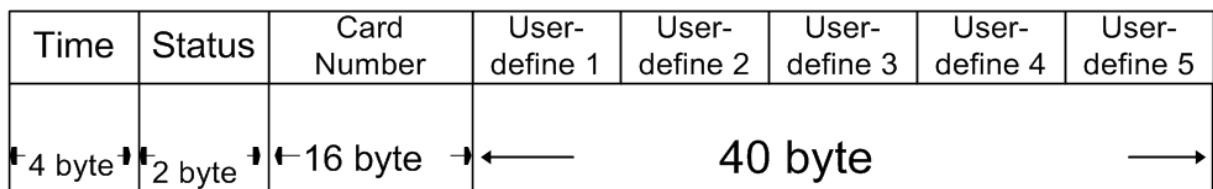
First byte: 0x00 signifies normal, 0x01 signifies add cards from device.

Second byte:

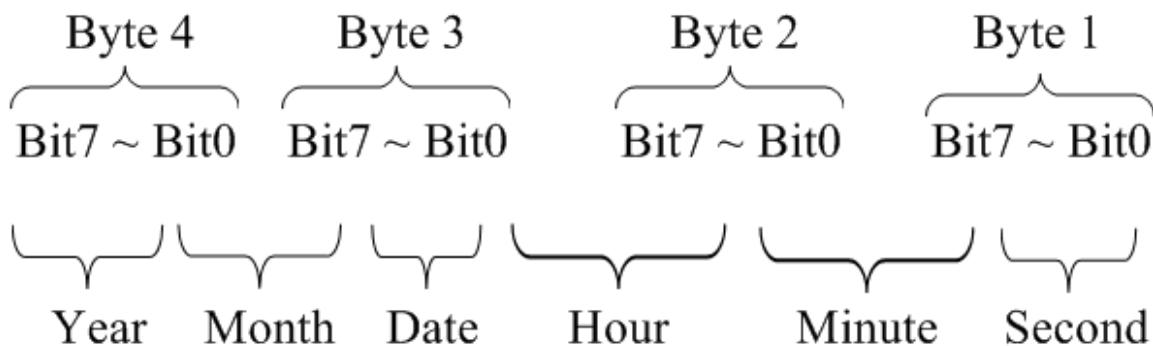
bit0	0: Do not need fingerprint 1: Request fingerprint
bit1	0: Download card number by software 1: Add card numbers by device

3. Message Display: 16 byte, Message will display in the LCD screen after card swiped. 0x20 is appended if the message does not reach 16 characters.
4. Fingerprint Index: 4 byte. Input 0xFF (Index saves in the module, Lo~Hi)

### 5.3.2 Swiped Card Records Format Description



1. Time (4byte) byte1, byte2, byte3, byte4



2. Status: 2 byte

Byte1: (Bit7 = 1)

bit 6~bit 4	Bit 3~bit 1	bit 0
0= DUTY ON	0= Input from keypad	0=Reader 1 (Inside Reader)
1= DUTY OFF	1= Input from T2 reader	0=Reader 1 (Outside Reader)
2= BREAK OUT	2= Input from Wiegand reader	
3= BREAK IN	3= Input from Mifare reader	
4= OVR_T_STR	4= Input from fingerprint	
5= OVR_T_END	5~7= Reserved	
6~7= Reserved		

Byte 2:

Reserved (bit 7~bit 4)	bit 3~ bit 0
	0= Valid card swipe
	1= Card number does not exist
	2= Error in the length of card number
	3= Invalid card number (Non-numeric character detected)
	4= Mifare write error
	5= Check fingerprint failed
	6= Blacklist
	7= Invalid time schedule
	8= Reserved
	9= Fingerprint Match
	10= Master card
	11= Invalid master card
	12= Add card from fingerprint recorder
	13= Delete card from fingerprint recorder
	14= Reserved
	15= Reserved

3. Card number: Length of card number will follow byte 3 of system parameter.
- (a) Once length is longer than pre-set length, system will cut redundant character, and only display pre-set length.
  - (b) Once length is shorter than pre-set length, 0x00 will be appended to make length is same with pre-set length.
4. User-defind1~5: Maximum has 5 divisions. 8 byte per division, length will follow byte 5~9 of system parameter. Once system parameter is 0, it signifies do not need reserve those divisions.

### 5.3.3 Memory Allocation

Address 0x0000~0x00FF, totally 256 bytes and 0x0000~0x001F belong system parameter.

Length	Address	Description	Default	Remark
1	0x0000	Communication way bit7= 0: RS-485/RS-232 bit0~6: Baudrate (1=9600, 2=19200, 3=38400, 4=115200)  bit7=1: TCP/IP bit0~6: Reserved	0x80	
1	0x0001	Reader Sensor Type Bit0: Recorder sensor 0=Mifare, 1= EM  Bit1~3 Reserved Bit4~7: Slave reader sensor 0=T1/T2, 1= Wiegand 26(8 digits) 2= Wiegand 26(10 digits) 3= Wiegand 34(10 digits) 4~15: Reserved	0x01	
4	0x0002	IP address	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	Maximum total amount of valid card. (Base on Flash memory property, once modify amount of valid card, it may need a sector. Adjustable range is 0 or 6023~15058)	0xD2, 0x3A	Lo byte, Hi byte 0x3AD2=15058 cards. (Once select "All cards are invalid", system will change to 0 automatically.)
1	0x0012	Recorder ID	0x01	For RS-232/RS-485 used
1	0x0013	Device type 0= All cards are invalid 1= Valid card + Display Name	0x01	Once select type0, system will adjust maximum total

				amount of valid card to 0.
1	0x0014	Swiped card records	0x00	1= Overwrite record Other= Do not overwrite record
1	0x0015	Reserved		
16	0x0016	Master Card	30191000	Master card appends 0x00 if length did not reach 16.
1	0x0026	Language Selection	0x00	0= English 1= Tradition Chinese 2= Simplified Chinese
1	0x0027	LCD date format	0x00	0= YYYYMM/DD 1= MM/DD/YYYY 2= DD/MM/YYYY
1	0x0028	Reserved		
1	0x0029	Invalid card records	0x01	0= Do not record 1= Record
2	0x002A	Relay 1 status settings  (Door): bit0~11 = Pulse time Pulse=0: refer to other relay action Pulse = 1~4095: indicates 0.1~409.5 sec  bit12~bit14: Reserved  bit15= other relay action (Pulse time =0) 0=Toggle 1= Latch (Do not have this mode)	0x28, 0x00	Lo byte, Hi byte 0x0028=40 40*0.1 sec=4 sec
1	0x002C	Check card number length	0x00	0= Ignore 1~16= Length of card number to check
1	0x002D	Check digit of card number	0x00	0= Ignore 1= Inspect
1	0x002E	Retrieve valid card number (Index)	0x01	1-16
1	0x002F	Retrieve valid card	0x00	0= get the length

		number (Length)		of card number swiped 1-16= Length specified
1	0x0030	Keypad Settings: bit0= keypad status (0= Refer to bit1~2 settings. 1= Enable)  bit1= Digital keypad status (0=Disable, 1= Enable)  bit2= Function key status (0=Disable, 1= Enable)  bit3= Reserved  bit4~5= keypad backlight (0= Auto, 1= Enable, 2= Disable)  bit6-7= Reserved	0x01	Default value: Keypad status: Enable  Keypad backlight: Auto
1	0x0031	Reserved		
1	0x0032	Re-swipecard check time	0x00	0= Disabled 1~255= Inspection time 1~255 seconds
1	0x0033	Set RS-485 output Refer to Chapter 5.1.5	0x00	0= Disabled 1= LED Display 2= DVR 3= Printer
10	0x0034	Mifare reader settings Byte1: 0x00 read serial number of card (byte2~byte10 ignore) 0x01: Read Key A 0x02: Read Key B  byte2: Retrieve valid card bit4~7: Index of retrieve valid card bit0~3: Length of retrieve valid card  byte4: Read block (0x00-0x3F)	0x00,0x10,0x00,0x00, 0xFF,0xFF,0xFF, 0xFF,0xFF,0xFF	Refer to Chapter 5.3.8 CMD30 (P0-P3, K0-K5)

		byte5-10: KeyA / keyB password.		
2	0x003E	Reserved		
1	0x0040	LCD display settings bit0= LCD message display (0=Disable, 1=Enable) bit1~3= Reserved bit4~5= LCD backlight (0=Auto, 1=Enable, 2=Disable) bit6~7= Reserved	0x10	Default value: LCD message display: Disable  LCD backlight: Enable
9	0x0041	Reserved		
1	0x004A	Work shift auto resume	0x01	0= Disable 1= Enable
1	0x004B	Daylight saving settings bit0= Daylight saving options (0= Advance time, 1= Delay time)  bit1= Adjustment status (0= Unadjusted, 1= Adjusted)  bit2= Enable/disable daylight saving function (0=Disable, 1=Enable)  Bit3~7= Reserved	0x00	Default is disable
4	0x004C	Daylight saving function time start	0x00,0x00,0x00,0x00	Month, Date, Hour, Min, BCD format.
4	0x0050	Daylight saving function time End	0x00,0x00,0x00,0x00	Month, Date, Hour, Min, BCD format.
2	0x0054	Daylight saving function adjust time	0x00,0x00	Hour, Min, BCD format.
1	0x0056	Error frequency	0x00	Error processing
1	0x0057	Error alarm frequency bit0~3= Frequency (0=Disable, 1~15= No of frequency).  bit4~5= Reserved bit6= How to	0x00	Error processing

		deactivated relay 3? (0= Valid card, 1= Master card)  Bit4~15= Deactivate relay3 mode (0= Valid card, 1=Master card)  bit7= Relay 3 activation (0=Disable, 1=Enable)		
2	0x0058	Reserved	0x28,0x00	
2	0x005A	Relay 3 status settings: bit0~11 = Pulse time Pulse=0: refer to other relay action Pulse = 1~4095: indicates 0.1~409.5 sec  bit12~bit14: Reserved  bit15= other relay action (Pulse time =0) 0=Toggle 1= Latch	0x00, 0x80	Lo byte, Hi byte Latch
1	0x005C	Audio Settings bitf0~2 = Reserved  bit3= Audio prompt (0=Disable, 1=Enable)  bit4= Siren output (0=Disable, 1=Enable)  bit5~7= Reserved	0x08	Default : Enable Audio prompt and enable siren output.
1	0x005D	Fingerprint Settings bit0= 1:1 mode (0=Auto, 1= 1:1 mode)  bit1= Master card can use fingerprint. (0= Disable, 1= Enable)  bit2~7= Reserved	0c00	When set fingerprint as Master card, pleas set communication by Com35.
1	0x005E	Previous 100 sets holiday schedule	0x10	BCD code
1	0x005F	Reserved		

1	0x0060	Door relay action when re-swipe card bit0= Door relay deactivated when Re-swipe card. (0=Disable, 1= Enable)  bit1~7= Reserved	0x00	
3	0x0061	Reserved		
10	0x0064	Write swipe card records back to Mifare card.  byte1: 0x00 close writes back function. (byte2~byte10 ignore) 0x01~0x3F: Start block of write back function.  byte2: 0x01~0x3F End block of write back function.  byte3: 0x00 Key A 0x80 Key B byte 4: Reserved  byte 5~10: KeyA / Key B password	0x00,0x00,0x00,0x00, 0xFF,0xFF,0xFF, 0xFF,0xFF,0xFF	Refer to Chapter 5.3.8 CMS30 (w0-w3, y0-y5)
1	0x006E	Select built-in relay action 0= Door relay 1= Siren relay	0x00	External ACU-30, relay 1 = Door relay Relay 2= Alarm relay
		Relay 0      Relay 1      Relay 2 Mode0      Door      Door      Alarm Mode 1      Siren      Door      Alarm		
1	0x006F	Duration of LCD return to ready status Default is 10 seconds 1~255= 1~255 seconds.	0x00	The duration cannot less than 8 seconds for fingerprint recorder.
1	0x0070	Siren Timetable 2 byte      2 bye Start      End Time      Time HHMM      HHMM	0xFF, 0xFF,0xFF, 0xFF	Hour (HH), Minute(MM), BCD format.
	0x0074	Reserved (0x74~0xFF)		

### 5.3.4 Siren Timetable

(Address 0x0100~0x019F: Total 160 bytes)

32*5	0x0100	Siren timetable 32 sets*5byte		
------	--------	-------------------------------	--	--

Group	Action time (2 byte)	Duration (2 byte)	Week date(1 byte)
1 <sup>st</sup> Address:0100H	HH:MM	MM:SS	bit7~0= x, Sun, Sat, Fri, Thu, Wed, Tue, Mon
2 <sup>nd</sup> Address:0105H	HH:MM	MM:SS	bit7~0= x ,Sun, Sat, Fri, Thu, Wed, Tue, Mon
3 <sup>rd</sup> Address:010AH	HH:MM	MM:SS	bit7~0= x, Sun, Sat, Fri, Thu, Wed, Tue, Mon
...	...	...	...
...	...	...	...
...	...	...	...
32 <sup>nd</sup> Address: 019BH	HH:MM	MM:SS	bit7~0= x ,Sun, Sat, Fri, Thu, Wed, Tue, Mon

1. Action time: 2 byte, First byte for hour, second byte for minute

Example: 08:30 → 0x08, 0x30 (BCD format)

2. Duration: 2 byte, First byte minute, second byte for second.

Example: 00:12 → 0x00, 0x12 (BCD format)

3. Week date: 1 byte

bit7: Reserved, bit6~0: Sun, Sat, Fri, Thu, Wed, Tue, Mon

0= Disable, 1= Enable

\* Please input 0xFF if select disable siren timetable.

### 5.3.5 Work Time Timetable

(Address 0x0200~0x027F: Total 128 bytes)

32*4	0x020	Work time timetable 32 sets*4byte		
------	-------	-----------------------------------	--	--

Group	Time (2 byte)	Work Status (2 byte)	Reserved(1 byte)
1 <sup>st</sup> Address:0x0200	HH:MM	Class	
2 <sup>nd</sup> Address:0x0204	HH:MM	Class	
3 <sup>rd</sup> Address:0x0208	HH:MM	Class	
...	...	...	
...	...	...	
...	...	...	
32 <sup>nd</sup> Address: 0x027C	HH:MM	Class	

1. Time: 2 byte, First byte for hour, Second byte for minute.

Example: 13:45 → 0x13, 0x45 (BCD format)

18:23 → 0x18, 0x23

2. Work Status

0= DUTY ON	1= DUTY OFF
2= BREAK OUT	3= BREAK IN
4= OT_START	5=OT_END

\* Please input 0xFF if select disable siren timetable.

### 5.3.6 Message Display

(Address 0x0300~0x03FF: Total 256bytes)

16*16	0x0300	Message display 16 sets* 16byte		
-------	--------	---------------------------------	--	--

Group	Message	LCD Display Length	Remark
1	DUTY ON	10	Duty on
2	DUTY OFF	10	Duty off
3	BREAK OUT	10	Away
4	BREAK IN	10	Returned
5	OT_START	10	Overtime starts
6	OT_END	10	Overtime end
7	Ready!	16	Message displayed during ready status
8	TEST1	16	User-define message
9	TEST2	16	User-define message
10			Reserved
11			Reserved
12			Reserved
13			Reserved
14			Reserved
15			Reserved
16			Reserved

If message to be displayed did not reach the expected length, kindly append 0X20.

Each group has 16 byte, append 0X20 if message to be displayed did not reach to 16 byte.

Group 1~6 will only display 10 byte.

### 5.3.7 Reserve Area

(Address 0x0400~0x1FFF)

0x0400 0x0400~0x01FFF Null

### 5.3.8 CMS:30 Mifare Settings

Send:

Data = C0+P0+P1+P2+P3+K0+K1+K2+K3+K4+K5

+w0+w1+w2+w3+y0+y1+y2+y3+y4+y5 → Hexadecimal code format

C0: 0x00 is the Mifare reader's command which similar with write table 54.

For P0~P3, kindly refer to table below:

P0	Select read card type	0x00 (Default)	0x00 → Read series no of Mifare card (byte2~byte10 ignore) 0x01 → Use key A 0x02 → Use key B
P1	Retrieve valid card	0x10 (Default)	bit4~7: Index of retrieve valid card bit0~3: Length of retrieve valid card
P2	Compress or Uncompress selection	0x00 (Default)	0x00 → Uncompress card no 0x01 → Compress card no
P3	Read block	0x00 (Default)	0x00~0x3F

K0~K5 are for keys when block read.

For w0~w3, kindly refer to table below: (C0 is 0x70 or 0x00, the setting of w0~y5 are same)

w0	Write swiped card. Records onto card: index block	0x00~0x3F	0x00 → Disable (byte2~byte10 ignore) 0x01~0x3F → Write swipe card records on to card index block. (w0 should be the first block of the sector)
w1	Write swiped card. Records onto card: Terminating block	0x00~0x3F	0x00~0x3F → Write swiped card. Records onto card: Terminating block (w1 must big than w0 and must input it in thrid block of the sector)
w2	Key type	0x00 or 0x80	0x00 → Key A 0x80 → Key B
w3	Reserved	0x00	

y0 ~ y5 the key value needed in order to write the records onto the block.

Response:

status=0 : OK

Data= NULL

Status=1 : Error

Data=Error Code (2 byte)

Note: 1. The block for P1 and w0 should be different.

2. Once write swiped card records onto card function selection, kindly refer to table below:

Block 0	Serial number of Mifare card	
...		
...		
P1 Block	Card No (16 byte)	
...		
...		
w0 Block	Index Block for saving swipe card record (2 Byte)	
w0 Block + 1 Start Block for saving swipe card record		
...		
...		
w1 Block		
Terminating Block for saving swipe card record  (must input it in third block of the sector)		Swipe card records occupy at least a sector. 8 records=2 Blcok*4 records Swipe card records occupy maximum 9 sectors. 104 records=(2 Block*4 records) +(8 Sector*12 records)

Example:

Swipe card record write on to card    Block 8~Block15

W0= 0x08, w1= 0x0F

Index Block for saving swipe card record should set 0x09 0x00

0x09→ Block for saving swipe card record

0x00→ What record which is stored in the block(0x00~0x03)

### 5.3.9 Error Code List

Hex Code	Description
0x0001	Invalid length of card number
0x0002	Allocated memory space for card number exceeded. Length of valid card number mismatched.
0x003	Exceeded the max number of valid cards.
0x0005	Wrong system parameter value
0x0006	Card number does not exist
0x0007	Table length transmitted is too long
0x0008	Table length retrieved is too long
0x0009	Invalid data length
0x000F	Error in the table transmitted to the device
0x0013	Unable to write in the Flash memory
0x0030	Failed to set Mifare reader settings
0x0031	Fingerprint recorder is currently busy and unable to set data
0x0032	Unable to enroll fingerprint
0x0033	Unable to delete fingerprint
0x0034	Reader type error