Value through Innovation

FCC ID:WQJ- IDCA36-0119

USER MANUAL

OmniXpress Reader

Version V1.00

Warning

Changes or modifications not expressly approved by IDTECH can void the user's authority to operate this equipment

Notes: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on. the user is encouraged to try to correct the interference by one or more of the following measures:

—Reorient or relocate the receiving antenna.

—Increase the separation between the equipment and receiver.

—Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

—Consult the dealer or an experienced radio/TV technician for help.

FCC Warning:
This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Installers and users are cautioned not to use any RS-232 cable other than the one supplied with the unit

Installers and users are cautioned not to remove the ferrite cores on the RS-232 and DC power cables supplied with the unit

Limited Warranty

ID TECH warrants this product to be in good working order for a period of one year from the date of purchase. If this product is not in good working order as warranted above, or should this product fail to be in good working order at any time during the warranty period, repair or replacement shall be provided by ID TECH.

This warranty does not cover incidental or consequential damages incurred by consumer misuse, or modification of said product. For limited warranty service during the warranty period, please contact ID TECH to obtain an RMA number and instructions for returning the product.

The information contained herein is provided to the user as a convenience. While every effort has been made to ensure accuracy, ID TECH is not responsible for damages that might occur because of errors or omissions, including any loss of profit or other commercial damage. The specifications described herein were current at the time of publication, but are subject to change at any time without prior notice.

ID TECH is a registered trademark of International Technologies & Systems Corporation. Spectrum and Value through Innovation are trademarks of International Technologies & Systems Corporation.

ID TECH
10721 Walker Street

Cypress, CA 90630-4720
(714)761-6368
www.idtechproducts.com

Table of Contents

Section 1
INTRODUCTION

Description

OmniXpress reader is a hybrid of magnetic stripe reading technology with Proximity smart card reading technology. It provides reading of MagStripe cards, passed through a card slot, and reading proximity cards, held within the reader’s field range.

The reader has a choice of communication interfaces to a host. Communications are through an interface cable. The cable can exit from the end or from the bottom of the base. A slot or channel in the base provides for in-field orientation of the cable exit direction and provides a strain relief function. Compatibility: The reader operates with major POS and ECR terminals old or new, providing the terminals have a compatible interface with the reader and a third party provides a software application and communication protocol layer to accomplish the required operations.

The product is fully capable of outdoor and unattended operations. the reader mounts to a countertop or a horizontal surface. it is mounted using either two threaded inserts on the base. The housing is three parts, a base, a cover, and an overlay applique; The appliqué provides area for logos, contactless symbol, defused LED windows, card stripe orientation, & etc.there is an internal rail for magnetic stripe reading. The reader base provides weep holes so that moisture can exit the housing.

There are 4 green LEDs acting as the read status indicators of the device. It is place at the top of the device; there is also a beeper inside device to provide a single pulse for audio feedback recognition of each key press.

The reader addresses several retail applications for ISO14443 compliant identification technology. The reader supports major financial contactless programs, such as MasterCard's PayPass, Visa's Contactless. The reader reads data from a variety of contactless sources, including credit cards, key fobs, and NFC devices and must support the RF requirements for ISO/IEC 14443A, ISO/IEC 14443 B as outlined in the relevant parts in ISO 18092. In particular, the reader shall meet the MasterCard, PayPass - ISO/IEC 14443 Implementation Specification. In addition, the reader shall be compatible with NFC for POS applications. The reader has the operability & functionality defined in the PayPass Implementation Specification. The reader shall be certified by a PayPass qualified test lab; the reader must meet the criteria to "Function Properly" and pass the test process.

In addition, the reader supports reading and writing to Mifare cards. Mifare cards supported are Standard, Classic, & UltraLite. This support is intended for access control applications, including transportation.

Section 2
INSTALLATION

RS232 Interface

As with a standard serial interface, the host must be configured to accept the data and perform the appropriate processing. For the RS232 interface device, the end user needs care that the host application's RS-232 parameters (baud rate, data bits, Start/Stop characters, parity, and handshaking method) match those expected by the reader. The magnetic reader's output can be formatted with terminating characters and special preamble and/or postamble character strings to match the data format expected by the host.

Section 3
DEFINITIONS

The following are definitions of common terms and abbreviations used throughout this manual.

ISO/IEC 14443 - Identification cards, Physical Characteristic
ISO/IEC 14443 - Identification cards, Contactless integrated circuit(s) cards/Proximity cards
PayPass - ISO/IEC 14443 Implementation Specification
Mifare - Philips' Contactless Mifare® Smart Card Ics & operations
PayPass - ISO/IEC 14443 Implementation Specification, Version 1.1 3-31-06
Visa qVSDC & MSD Document Version 1.0.1, May 2006
Visa Cross Reference Guide: ISO14443 versus PayPass, V1.0, January 2005
PayPass - Mag Stripe, Technical Specifications, Version 3.2, October 2006
PayPass - Mag Stripe, Vendor Testing, Version 2.0, November 2005
PayPass - Terminal Implementation Guide, Latest Revision
AMEX - Compatibility Requirements for PCDs, Version 1.0
AMEX - Application Level Interface Spec, Version 1.7
AMEX - Terminal Application Spec, Version 1.5

Section 4
CONFIGURATION PARAMETERS

I. **Command protocol**

Host to device: STX LenL LenH CommandData Lrc1 Lrc2 ETX
STX: 0x02. 1 byte.
LenL+LenH: sizeof(CommandData). 2 bytes.
CommandData: main command string. Several bytes needed.
Lrc1: Exclusive or of CommandData. 1 byte.
Lrc2: Sum of CommandData. 1 byte.
ETX: 0x03. 1 byte.
E.g. Turn application off command is: 0x02 0x02 0x00 0x78 0x16 0x00 0x6E 0x8E 0x03.

Device to host: STX LenL LenH ResponseData Lrc1 Lrc2 ETX
STX: 0x02. 1 byte.
LenL+LenH: sizeof(ResponseData). 2 bytes.
ResponseData: main response string. Several bytes needed.
Lrc1: Exclusive or of ResponseData. 1 byte.
Lrc2: Sum of ResponseData. 1 byte.
ETX: 0x03. 1 byte.

II. **Host to device command set**

1. Test use commands.

To use these commands, the application must be off (Use command **APPLICATION_CONTROL in 3-5).**

1-1. CARRIER

COMMAND: <0xFD><0x01><ON/OFF>.
Make the device to emit.
PARAMETERS:
<0xFD><0x01> is the command head.
<ON/OFF> ON(0x01) means turn on carrier, OFF(0x00) means turn off carrier.
RETURN:
Success: <ACK>.
Fail: This operation will succeed always.

1-2. POLLING

COMMAND: <0xFD><0x02><PollingCount>.
Make the device to poll for one card.
PARAMETERS:
<0xFD><0x02> is the command head.
<PollingCount> is the polling times. 1 byte. 0x00 means forever.
RETURN:
Success: <ACK>.
Fail: This operation will succeed always.

1-3. RESET

COMMAND: <0xFD><0x03><0x00>
Make the device to reset the carrier.
PARAMETERS:
<0xFD><0x03><0x00> is the command head.
RETURN:
Success: <ACK>.
Fail: This operation will succeed always.

1-4. REQA

COMMAND: <0xFD><0x14><0x01>.
Inform the device to send command request to type A card.
PARAMETERS:
<0xFD><0x14><0x01> is the command head.
RETURN:
Success: <ACK><All commands and their responses>.
Fail: This operation will succeed always.

1-5. WUPA

COMMAND: <0xFD><0x14><0x00>.

Inform the device to send command wake up to type A card.

PARAMETERS:

<0xFD><0x14><0x00> is the command head.

RETURN:

Success: <ACK><All commands and their responses>.

Fail: This operation will succeed always.

1-6. HALTA

COMMAND: <0xFD><0x16><0x00>.

Inform the device to send command halt type A card.

PARAMETERS:

<0xFD><0x16><0x00> is the command head.

RETURN:

Success: <ACK><All commands and their responses>.

Fail: This operation will succeed always.

1-7. ACTIVEA

COMMAND: <0xFD><0x18><0x00>.

Inform the device to send commands active type A card.

PARAMETERS:

<0xFD><0x18><0x00> is the command head. 3 bytes.

RETURN:

Success: <ACK><All commands and their responses>.

Fail: This operation will succeed always.

1-8. REQB

COMMAND: <0xFD><0x25><0x01>.

Inform the device to send command request to type B card.

PARAMETERS:

<0xFD><0x25><0x01> is the command head. 3 bytes.

RETURN:

Success: <ACK><All commands and their responses>.

Fail: This operation will succeed always.

1-9. WUPB

COMMAND: <0xFD><0x25><0x00>

Inform the device to send command wake up to type B card.

PARAMETERS:

<0xFD><0x25><0x00> is the command head. 3 bytes.

RETURN:

Success: <ACK><All commands and their responses>.

Fail: This operation will succeed always.

1-10. HALTB

COMMAND: <0xFD><0x27><0x00>

Inform the device to send command halt type B card.

PARAMETERS:

<0xFD><0x27><0x00> is the command head. 3 bytes.

RETURN:

Success: <ACK><All commands and their responses>.

Fail: This operation will succeed always..

1-11. ACTIVEB

COMMAND: <0xFD><0x29><0x00>

Inform the device to send commands active type B card.

PARAMETERS:

<0xFD><0x29><0x00> is the command head. 3 bytes.

RETURN:

Success: <ACK><All commands and their responses>.

Fail: This operation will succeed always.

1-12. APPCOMMAND

COMMAND: <0xFD><0x32><CPDU>

Inform the device to send commands to the active card.

PARAMETERS:

<0xFD><0x32> is the command head. 2 bytes.

<CPDU> is the command send to the card.

RETURN:

Success: <ACK><R_APDU>

Fail: This operation will succeed always.

1-13. APP_TEST_START

COMMAND: <0xFD><0x41>

Inform the device to start digital test.

PARAMETERS:

<0xFD><0x41> is the command head. 2 bytes.

RETURN:

Success: <ACK>.

Fail: This operation will succeed always.

1-14. APP_TEST_STOP

COMMAND: <0xFD><0x42>

Inform the device to stop digital test.

PARAMETERS:

<0xFD><0x42> is the command head. 2 bytes.

RETURN:

Success: <ACK>.

Fail: This operation will succeed always.

Note: Commands and their responses format: C len command R len response C len command R len Response ...

2. Serial port control command set

The default parameters are: 38400 baudrate, 8 data bits, 1 stop bit, no parity, no handshaking.

2-1. SET_BAUDRATE

COMMAND: <0x70><0x01><NewBaudrate>.

Set the device's baudrate.

PARAMETERS:

<0x70><0x01> is the command head. 2 bytes.

<NewBaudrate> is the required serial port communication baudrate. 1 byte: 0x01 for 1200, 0x02 for 2400, 0x03 for 4800, 0x04 for 9600, 0x05 for 14400, 0x06 for 19200, 0x07 for 38400, 0x08 for 57600, 0x09 for 115200.

RETURN:

Success: <ACK><OldBaudrate (1 byte)>.

OldBaudrate is the old baudrate.

Fail: <NAK><Error code>.

2-2 SET_STOPBITS

COMMAND: <0x70><0x02><NewStopBits>.

This command is used to set the device's stop bits.

The total length is 3 bytes

PARAMETERS:

<0x70><0x02> is the command head.

<NewStopBits> is the required serial port communication stop bits. 1 byte: 0x01 for 1 stop bit, 0x02 for 2 stop bits, 0x03 for 1.5 stop bits.

RETURN:

Success: <ACK>< OldStopBits (1 byte)>.

OldStopBits is the old stop bits.

Fail: <NAK><Error code>.

2-3. SET_PARITY

COMMAND: <0x70><0x03><NewParity>.

Set the device's parity style.

PARAMETERS:

<0x70><0x03>is the command head.

<NewParity> is the required serial port communication parity. 1 byte: 0x00 for no parity, 0x01 for odd parity, 0x02 for even parity, 0x03 for mark parity, 0x04 for space parity.

RETURN:
Success: <ACK>< OldParity. (1 byte)>.
OldParity is the old parity.
Fail: <NAK><Error code>.

2-4. SET_HANDSHAKING

COMMAND: <0x70><0x05><NewHandShakingMode>.
Set the device's hand shaking style.
PARAMETERS:
<0x70><0x05> is the command head.

<NewHandShakingMode> is the required serial port communication hand shaking mode. 1 byte: 0x00 for no hand shaking mode, 0x01 for XON/XOFF hand shaking mode, 0x02 for RTS/CTS hand shaking mode.

RETURN:
Success: <ACK>< OldHandShakingMode (1 byte)>.
OldHandShakingMode is the old hand shaking mode
Fail: <NAK><Error code>.

2-5. SET_XON_CHAR

COMMAND: <0x70><0x06><NewXONChar>
Set the device's XON char.
PARAMETERS:
<0x70><0x06> is the command head.
<NewXONChar> is the required serial port communication XON char. 1 byte
RETURN:
Success: <ACK><OldXONChar.(1 byte)>.
OldXONChar is the old XOFF char.
Fail: This operation will succeed always.

2-6. SET_XOFF_CHAR

COMMAND: <0x70><0x07><NewXOFFChar>
Set the device's XON char.
PARAMETERS:
<0x70><0x07> is the command head.
<NewXOFFChar> is the required serial port communication XOFF char. 1 byte
RETURN:
Success: <ACK><OldXOFFChar.(1 byte)>.
OldXOFFChar is the old XOFF char.
Fail: This operation will succeed always.

3. System command set

3-1. GET_VERSION

COMMAND: <0x78><0x01>.
Get the system's version.
PARAMETERS:
<0x78><0x01> is the command.
RETURN:
Success: <ACK><version string (42 bytes)>.
Fail: This operation will succeed always.

3-2. GET_SERIAL_NUMBER

COMMAND: <0x78><0x02>
Get the system's serial number.
PARAMETERS:
<0x78><0x02> is the command.
RETURN:
Success: <ACK><serial number string (32 bytes)>.
Fail: This operation will succeed always.

3-3. BEEP

COMMAND: <0x78><0x04><Duration>
Make the system's beeper to beep for a designated time.
PARAMETERS:
<0x78><0x04> is the command header.
<Duration> is the beeper's beep last time in 0.1 second.

RETURN:
Success: <ACK>.
Fail: This operation will succeed always.

3-4. LED_CONTROL

COMMAND: <0x78><0x05><LED><ON/OFF>.
Control the system's led.
PARAMETERS:
<0x78><0x05> is the command header.
<LED> is the led's id. 0x00--0x07 is acceptable.
<ON/OFF>: 0x01(ON) means let led to light on and 0x00(OFF) means let led to light off.
RETURN:
Success: <ACK>.
Fail: This operation will succeed always.

3-5. APPLICATION_CONTROL

COMMAND: <0x78><0x16><APPON/OFF>.
Control the APPLICATION's on or off.
PARAMETERS:
<0x78><0x16> is the command header.
<ON/OFF>: 0x01(ON) means let the application on and 0x00(OFF) means let the application off.
RETURN:
Success: <ACK>.
Fail: This operation will succeed always.

4. Application related commands

4-1. APP_FLUSH_DATA

COMMAND: <0x7E><0x01>.
Flush the unread read data.
PARAMETERS:
<0x7E><0x01> is the command header.
RETURN:
Success: <ACK>.
Fail: This operation will succeed always.

4-2. APP_GET_DATA

COMMAND: <0x7E><0x02><IsFlushData>
Get the unread read data. This command will reset the reader into buffered mode.
PARAMETERS:
<0x7E><0x02> is the command header.
<IsFlushData>: 0x00 is not, others yes.
RETURN:
Success: <ACK><data>.
Fail: <NAK><ERROR_NO_DATA_AVAILABLE>

4-3. APP_SET_AUTOSENDMODE

COMMAND: <0x7E><0x04><MODE>.
Set data send mode.
PARAMETERS:
<0x7E><0x04> is the command header.
<MODE>: 0x00 means buffered mode and 0x01 means auto send mode.
RETURN:
Success: <ACK>.
Fail: This operation will succeed always.

4-4. APP_SET_DATETIME

COMMAND: <0x7E><0x08><DATE(YYYYMMDDHHMMSS)>.
Set the date and time in the system.
PARAMETERS:
<0x7E><0x08> is the command header.
DATE(YYYYMMDDHHMMSS) is the date and time. Eg.0x32 0x30 0x30 0x36 0x30 0x31 0x30 0x32 0x31 0x36 0x33 0x30 0x32 0x35 means 2006-01-02 16:30:25.
RETURN:
Success: <ACK>.

Fail: This operation will succeed always.

4-5. APP_GET_MCHIP_DATA

COMMAND: <0x7E><0x10><IsFlushData>

Get the unread read m/chip data.

PARAMETERS:

<0x7E><0x10> is the command header.

<IsFlushData>: 0x00 is not, others yes.

RETURN:

Success: <ACK><data>.

Fail: <NAK><ERROR_NO_DATA_AVAILABLE>

4-6. APP_GET_TVR

COMMAND: <0x7E><0x20>.

Get the last terminal verification result.

PARAMETERS:

<0x7E><0x20>. is the command header.

RETURN:

Success: <ACK><TVR>

Fail: <NAK><ERROR_NO_DATA_AVAILABLE>

4-7. APP_GET_TSI

COMMAND: <0x7E><0x22>.

Get the last transaction status information.

PARAMETERS:

<0x7E><0x22> is the command head.

RETURN:

Success: <ACK><TSL>

Fail: <NAK><ERROR_NO_DATA_AVAILABLE>

4-6. APP_GET_TVRTSI

COMMAND: <0x7E><0x24>.

Get the last terminal verification result and transaction status information.

PARAMETERS:

<0x7E><0x24> is the command head.

RETURN:

Success: <ACK><TVR><TSL>

Fail: <NAK><ERROR_NO_DATA_AVAILABLE>

4-7. APP_GET_TRAN_RESULT

COMMAND: <0x7E><0x26>

Get the last transaction result information.

PARAMETERS:

<0x7E><0x26> is the command head.

RETURN:

Success: <ACK><Result>

<Result> is 1 byte, see below definition.

# TRANSACTION_TWOCARDS	0x7A
# TRANSACTION_TERMINATE	0x7E
# TRANSACTION_DECLINE	0x81
# TRANSACTION_ACCEPT_OFFLINE	0x82
# TRANSACTION_GO_ONLINE	0x84
# TRANSACTION_COMPLETE	0x80

Fail: <NAK><ERROR_NO_DATA_AVAILABLE>

4-8. APP_GET_CVMRESULT

COMMAND: <0x7E><0x28>.

Get the last terminal verification CVM result.

PARAMETERS:

<0x7E><0x28> is the command head.

RETURN:

Success: <ACK><CVMResult>

Fail: <NAK><ERROR_NO_DATA_AVAILABLE>

4-9. APP_GET_DATETIME

COMMAND: <0x7E><0x18>

Get date and time.

PARAMETERS:

<0x7E><0x18> is the command head.

RETURN:

Success: <ACK>< DATE(YYYYMMDDHHMMSS)> Eg.0x32 0x30 0x30 0x36 0x30 0x31 0x30 0x32 0x31 0x36 0x33 0x30 0x32 0x35 means 2006-01-02 16:30:25.

Fail: This operation will succeed always.

4-10. APP_GET_TRANSACTION_COUNT

COMMAND: <0x7E><0x30>

Get the transaction count.

PARAMETERS:

<0x7E><0x30> is the command head.

RETURN:

Success: <ACK><Transaction count(4 bytes)>

Fail: This operation will succeed always.

Data format: Application type(1 byte) track1 track2.
track1 is encapsuled by start sentinel '%' and end sentinel '?'.
track2 is encapsuled by start sentinel ';' and end sentinel '?'.
Auto send mode data will not be capsuled by STX and ETX.

Application type:

MASTER_APPLICATION	0x01
VISA_APPLICATION	0x02
ExpressPay_APPLICATION	0x03
Contact Magstripe	0xA1

5. Data setting commands.

To use these commands, the application must be off.

5-1. RETRIEVE_TERMINAL_DATA

COMMAND: <0x7E><0xF0><0x01><Tag>.

Retrieve the terminal's data element.

PARAMETERS:

<0x7E><0xF0><0x01> is the command head.

<Tag> is the desired data element's tag. 1 or 2 bytes

Note: Action code's tag is coded as '0x01'.

RETURN:

Success: <ACK><Requested data element (TLV)>

Fail: This operation will succeed always.

5-2. UPDATE_TERMINAL_DATA

COMMAND: <0x7E><0xF0><0x02><Data element (TLV)>.

Update the terminal's data element temporary.

PARAMETERS:

<0x7E><0xF0><0x02> is the command head..

<Data element (TLV)>: TLV format data

RETURN:

Success: <ACK><Status code (1 byte)>.

Fail: This operation will succeed always.

5-3. UPDATE_TERMINAL_DATA_SAVE

COMMAND: <0x7E><0xF0><0x03><Data element (TLV).>

Update the terminal's data element.

PARAMETERS:

<0x7E><0xF0><0x03> is the command head.

<Data element (TLV)>: TLV format data

RETURN:

Success: <ACK><Status code (1 byte)>

Fail: This operation will succeed always.

5-4. SELECT_TERMINAL_APPLICATION

COMMAND: <0x7E><0xF0><0x05><Application type>.

Select the terminal's application, used for test only.

PARAMETERS:

<0x7E><0xF0><0x05> is the command head.

Application type: 1 byte.

RETURN:

Success: <ACK><Old application type (1 byte)>.

Fail: This operation will succeed always.

5-5. RETRIEVE_COUNTRY_CODE_TABLE

COMMAND: <0x7E><0xF0><0x14><PageNumber>

Retrieve the terminal's country code table data.

PARAMETERS:

<0x7E><0xF0><0x14> is the command head.

<PageNumber> is the desired page number. 0x01--0x08.

RETURN:

Success: <ACK><Country code table data>

Fail: This operation will succeed always.

Note: Country code is 2 chars + 3 chars + 3 digits. Total 256 items. Total 8 pages, 32 items (256 bytes) per page.

5-6. UPDATE_COUNTRY_CODE_TABLE

COMMAND: <0x7E><0xF0><0x18><PageNumber><Country_code_table_data>.

Update the terminal's saved country code table data.

PARAMETERS:

<0x7E><0xF0><0x18> is the command head.

<PageNumber> is the desired page number. 0x01--0x08.

<Country_code_table_data> is the data.

RETURN:

Success: <ACK><Status code(1 byte)>

Fail: This operation will succeed always.

5-9. RETRIEVE_PUBLIC_KEY

COMMAND: <0x7E><0xF1><0x01><RID(5 BYTES)><Index(1 byte)>.

Retrieve the terminal's SAVED public key of a special RID and its index.

PARAMETERS:

<0x7E><0xF1><0x01> is the command head.

<RID(5 BYTES)> is the applications's RID.

<Index(1 byte)> is the key index.

RETURN:

Success: <ACK><RID(5)><Index(1)><Hash Indicator(1)><Algorithm Indicator(1)> <Exponent(4)><Key Module Length(1)><Key Hash(20)><Module Data(var)>.

5-10. REVOKE_PUBLIC_KEY

COMMAND: <0x7E><0xF1><0x02><RID(5 BYTES)><Index(1 byte)>.

Revoke the terminal's SAVED public key of a special RID and its index.

PARAMETERS:

<0x7E><0xF1><0x02> is the command head.

<RID(5 BYTES)> is the applications's RID.

<Index(1 byte)> is the key index.

RETURN:

Success: <ACK>

Fail: This operation will succeed always.

5-11. INVOKE_PUBLIC_KEY

COMMAND: <0x7E><0xF1><0x04><DATA>.

Invoke a public key of a special RID and its index in the terminal.

PARAMETERS:

<0x7E><0xF1><0x04> is the command head.

<DATA>: RID(5) Index(1) Hash Indicator(1) Algorithm Indicator(1) Exponent(4) Key Module Length(1) Key Hash(20) Pad(29) Module Data(var).

RETURN:

Success: <ACK>

Fail: This operation will succeed always.

5-12. RETRIVE_PUBLIC_KEY_COUNT

COMMAND: <0x7E><0xF1><0x08><RID (5 BYTES)>

Retrieve the count of the terminal's SAVED public key of a special RID.

PARAMETERS:

<0x7E><0xF1><0x08> is the command head.

<RID (5 BYTES)>

RETURN:

Success: <ACK><Count(1 byte)>.

Fail: This operation will succeed always.

6. Device direct control commands. Command header: 0xFE.

To use these commands, the application must be off (Use command **APPLICATION_CONTROL in 3-5).**

6-1. SEARCH CARDS

COMMAND: <0xFE><0x01><A/B><AFI>

Detect cards in the field.

PARAMETERS:

<0xFE><0x01> is the command head..

<A/B> is card type: 0 for type A and type B, 1 for type A and 2 for type B. 1 byte.

<AFI> is application family identifier for type B. 0 for type A. 1 byte.

RETURN:

Success:

<ACK><CardCount>[<CardType><UIDLen><UID>][<CardType><UIDLen><UID>]...

<CardCount> is the count of found cards in the field.

[<CardType><UIDLen><UID>] will happen <CardCount> times.

< CardType > is the card's type: 1 for type A and 2 for type B.

<UIDLen> is the following UID's length.

<UID> is the found card's UID (for type A card) or PUID (for type B card).

Fail: <ACK><0x00>.

6-2. SELECT ONE CARD

COMMAND: <0xFE><0x02><CardType><LenOfUID/PUID><UID/PUPI>

Select one card.

PARAMETERS:

<0xFE><0x02> is the command head.

<CardType> is the card's type: 1 for type A and 2 for type B. 1 byte.

<LenOfUID/PUID> is the length of the selected UID for type A or PUPI for type B. 1 byte.

<UID/PUPI>: UID for type A. PUPI for type B.

RETURN:

Success: <ACK><CardHandle>

<CardHandle> is the selected card's handle, it will be used by other commands to communicate with a specified card.

Fail: <ACK><0x00>

6-3. DELETE ONE CARD

COMMAND: <0xFE><0x03><CardHandle>.

Delete one card from the system.

PARAMETERS:

<0xFE><0x03> is the command head.

<CardHandle> is the Card's handle returned by **SELECT ONE CARD** command.

RETURN:

Success: <ACK><Free slots available>

Fail: This operation will succeed always.

6-4. DELETE ALL CARDS

COMMAND: <0xFE><0x04>

Delete all cards from the system.

PARAMETERS:

<0xFE><0x04> is the command head.

RETURN:

Success: <ACK><Free slots available>. 2 bytes

Fail: This operation will succeed always.

6-5. GET USING CARDS COUNT

COMMAND: <0xFE><0x05>

Get the cards count using by the device system.

PARAMETERS:

<0xFE><0x05> is the command head.

RETURN:

Success: <ACK><Using cards count by the device>. 2 bytes

Fail: This operation will succeed always.

6-6. GET FREE SLOTS

COMMAND: <0xFE><0x06>.

Get the free slots available in the device system.

PARAMETERS:

<0xFE><0x06> is the command head.

RETURN:

Success: <ACK><Free slots available>. 2 bytes

Fail: This operation will succeed always.

6-7. GET CARD CID

COMMAND: <0xFE><0x07><CardHandle>.

Retrieve one card’s current CID used in the device system.

PARAMETERS:

<0xFE><0x07> is the command head.

<CardHandle> is the Card's handle returned by **SELECT ONE CARD** command.

RETURN: <ACK><Card's CID> if available. CID is between 1 and 14.

This operation will return <NAK><ERROR_INVALID_CID_RETURNED> if failed.

6-8. DISABLE CARD CID

COMMAND: <0xFE><0x08><CardHandle>

Inform the device do not use card’s CID in the system.

PARAMETERS:

<0xFE><0x08> is the command head.

<CardHandle> is the Card's handle returned by **SELECT ONE CARD** command.

RETURN:

Success: <ACK>.

Fail: This operation will succeed always.

6-9. DISABLE CARD NAD

COMMAND: <0xFE><0x09><CardHandle>

Inform the device do not use card’s NAD in the system.

PARAMETERS:

<0xFE><0x09> is the command head.

<CardHandle> is the Card's handle returned by **SELECT ONE CARD** command.

RETURN:

Success: <ACK>.

Fail: This operation will succeed always.

6-10. SEND COMMAND TO CARD

COMMAND: <0xFE><0x0A><CardHandle><CommandBlock>

Make the device to communicate with the card in the system.

PARAMETERS:

<0xFE><0x0A> is the command head.

<CardHandle> is the Card's handle returned by **SELECT ONE CARD** command.

<CommandBlock> is the input command.

RETURN:

Success: <ACK><Card returned data>

Fail: <NAK><Error code>

6-11. Special commands for one card control only

COMMAND: <0xFE><0x0D><A/B><CommandBlock>

These commands are used to let the device to communicate with only one card in the system.
The total length is 3+LenOfCmd bytes.

PARAMETERS:

<0xFE><0x0D> is the command head..

<CommandBlock> is the input command. See below for detail.

6-11-1. REQUEST A CARD

CommandBlock: <0x00>

Inform the device to send command to request a card.

RETURN: <ACK><Card response responses>

6-11-2. <WAKE UP A CARD>

CommandBlock: <0x01>

This command is used to let the device to send command wake up a card.

PARAMETERS:

0x01 is the command head. 1 byte.

RETURN: <ACK><Card response>

6-11-3. <SELECT A CARD>

CommandBlock: <0x02>.

Inform the device to send command to select a card.

RETURN: <ACK><Card response>

6-11-4. <HALT A CARD>

CommandBlock: <0x04>

Inform the device to send command to halt a card.

RETURN: <ACK><Card response>