IDTECH®

Value through Innovation

USER MANUAL

Sign&Pay™

Technical Reference Manual

CE FC

80098502-001-A
09-27-2011

Software & Documentation License Agreement
CAREFULLY READ ALL THE TERMS, CONDITIONS, AND RESTRICTIONS OF THIS LICENSE AGREEMENT BEFORE USING OR INSTALLING THE SOFTWARE. YOUR USE OR INSTALLATION OF THE SOFTWARE PRESUMES YOUR AGREEMENT WITH AND ACCEPTANCE OF THE TERMS, CONDITIONS, AND RESTRICTIONS CONTAINED IN THIS AGREEMENT. IF YOU DO NOT AGREE WITH THESE TERMS, CONDITIONS, AND RESTRICTIONS, PROMPTLY RETURN THE SOFTWARE AND RELATED DOCUMENTATION TO – ID TECH Support, 10721 Walker Street, Cypress, CA 90630.

TERMS, CONDITIONS AND RESTRICTIONS
ID TECH, Incorporated (the "Licensor") owns and has the right to distribute the described software and documentation, collectively referred to as the "Software".
LICENSE: Licensor grants you (the "Licensee") the right to use the Software in conjunction with ID TECH products.

LICENSEE MAY NOT COPY, MODIFY OR TRANSFER THE SOFTWARE and DOCUMENTATION IN WHOLE OR IN PART EXCEPT AS EXPRESSLY PROVIDED IN THIS AGREEMENT. Licensee may not decompile, disassemble, or in any other manner attempt to reverse engineer the Software. Licensee shall not tamper with, bypass, or alter any security features of the software or attempt to do so.

TRANSFER: Licensee may not transfer the Software & Documentation or license the Software to another party without prior written authorization of the Licensor. If Licensee transfers the Software without authorization, all rights granted under this Agreement are automatically terminated.

COPYRIGHT: The Software is copyrighted. Licensee may not copy the Software except to archive the Software or to load the Software for execution purposes. All other copies of the Software are in violation of this Agreement.

TERM: This Agreement is in effect as long as Licensee continues the use of the Software. The Licensor also reserves the right to terminate this Agreement if Licensee fails to comply with any of the terms, conditions, or restrictions contained herein. Should Licensor terminate this Agreement due to Licensee's failure to comply, Licensee agrees to return the Software to Licensor. Receipt of returned Software by the Licensor shall mark the termination.

LIMITED WARRANTY: Licensor warrants to the Licensee that the disk(s) or other media on which the Software is recorded to be free from defects in material or workmanship under normal use. THE SOFTWARE IS PROVIDED AS IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Because of the diversity of conditions and PC hardware under which the Software may be used, Licensor does not warrant that the Software will meet Licensee specifications or that the operation of the Software will be uninterrupted or free of errors.

IN NO EVENT WILL LICENSOR BE LIABLE FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE. Licensee's sole remedy in the event of a defect in material or workmanship is expressly limited to replacement of the Software disk(s) if applicable.

GOVERNING LAW: If any provision of this Agreement is found to be unlawful, void or unenforceable, that provision shall be removed from consideration under this Agreement and will not affect the enforceability of any of the remaining provisions. This Agreement shall be governed by the laws of the State of California and shall insure to the benefit of ID TECH, Incorporated, its successors, or assigns.

ACKNOWLEDGMENT: LICENSEE ACKNOWLEDGES THAT HE HAS READ THIS AGREEMENT, UNDERSTANDS ALL OF ITS TERMS, CONDITIONS, AND RESTRICTIONS AND AGREES TO BE BOUND BY THEM. LICENSEE ALSO AGREES THAT THIS AGREEMENT SUPERSEDES ANY AND ALL, VERBAL AND WRITTEN, COMMUNICATIONS BETWEEN LICENSOR AND LICENSEE OR THEIR ASSIGNS RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

QUESTIONS REGARDING THIS AGREEMENT SHOULD BE ADDRESSED IN WRITING TO ID TECH, INCORPORATED, ATTENTION: CUSTOMER SUPPORT, AT THE ABOVE ADDRESS OR E-MAILED TO: support@idtechproducts.com

Information Provided

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, nor for any infringements or patents or other rights of third parties that may result from its use. The specifications & information described herein were current at the time of publication, but are subject to change at any time without prior notice.

Proprietary & Trademark Statements

This document contains proprietary information of ID TECH. Its receipt or possession does not convey any rights to reproduce or disclose its contents or to manufacture, use or sell anything it may describe. Reproduction, disclosure, or use without specific written authorization from ID TECH is strictly forbidden.

ID TECH
10721 Walker Street
Cypress, CA 90630
(714) 761-6368
www.idtechproducts.com

Revision History

Revision	Date	Description
50	10/06/2010	Initial Release
51	02/16/2011	Minor changes on the command output format
52	05/19/2011	Added commands - 1.4.13 Get BMP Format Signature - 1.4.17 Set BMP Format Signature - 1.5.26 SecureHead Clear Data Output Command - 1.5.27 SecureHead Load DUKPT Command - 1.5.33 Get MSR Card Data Output Status - 1.5.34 Read MSR Card Data Output Format - 1.5.35 Set MSR Data Output Format - 1.6.9 Get Bootloader Check Value Modified commands - 1.6.1 Get Encrypted PIN Online - 1.6.10 Get Pinpad Input as Numeric - 1.6.11 Get Pinpad Input as Amount - 1.6.12 Get Card Account - 1.6.13 Get Encrypted Data Added default unencrypted MSR output format
53	08/11/2011	Modified encrypted MSR output format Added application note on RS232 and USB-HID interface
A	09/27/2011	Added commands (supported in firmware version 1.00.027 and above) - 1.5.28 Secured output structure setting - 1.5.29 Encrypt Option Setting - 1.5.30 Hash Option Setting - 1.5.31 Mask Option Setting - 1.5.36 Get MSR Card Encrypted Data With PIN Key Or Data Key - 1.5.37 Set MSR Card Encrypted Data With PIN Key Or Data Key - 1.6.19 Manual Input Card Data Removed Enable key loading command Initial Release

Sign&Pay Technical Reference Manual

Table of Contents

1.0 Sign&Pay Commands Descriptions

All commands in this section have the following command and response structures:

Host to Device Command Protocol

<STX><LenL><LenH><CommandData><Lrc1><Lrc2><ETX>

<STX>: 0x02. 1 byte.
<LenL><LenH>: size of CommandData. If Length of CommandData is less than 0x8000, LenL + LenH occupies 2 bytes, otherwise it occupies 3 bytes.
<CommandData>: main command string. One or more bytes.
<Lrc1>: Exclusive or of CommandData. 1 byte.
<Lrc2>: Sum of CommandData. 1 byte.
<ETX>: 0x03. 1 byte.

Device to Host Command Protocol

<STX><LenL><LenH><ResponseData><Lrc1><Lrc2><ETX>

<STX>: 0x02. 1 byte.
<LenL><LenH>: size of ResponseData. If Length of ResponseData is less than 0x8000, LenL + LenH occupies 2 bytes, otherwise it occupies 3 bytes.
<ResponseData>: <ACK/NAK> plus the response string. One or more bytes.
<Lrc1>: Exclusive or of ResponseData. 1 byte.
<Lrc2>: Sum of ResponseData. 1 byte.
<ETX>: 0x03. 1 byte.

<ACK>: 0x06
<NAK>: 0x15

All length data described throughout the document uses little-endian format.

1.1 Device Related Settings

1.1.1 Get Firmware Version

COMMAND: <0x78><0x46><0x01>
Get Sign&Pay firmware version.
PARAMETERS:
<0x78><0x46><0x01> is the command head.
RETURN:
IDTECH-SIGN&PAY Vx.xx

1.1.2 Get Serial Number

COMMAND: <0x78><0x46><0x02>

Get Sign&Pay serial number.

PARAMETERS:

<0x78><0x46><0x02> is the command head.

RETURN:

Serial number

1.1.3 Set Serial Number

COMMAND: <0x78><0x46><0x03><Serial number>

Set serial number.

PARAMETERS:

<0x78><0x46><0x03> is the command head.

<Serial number> The length must be eight.

RETURN:

<ACK>

1.1.4 Get Model Number

COMMAND: <0x78><0x46><0x20>

Get model of communication. IDFA-3123 is for RS232 and IDFA-3153 for USB HID.

PARAMETERS:

<0x78><0x46><0x20> is the command head.

RETURN:

IDFA-3123/ IDFA-3153

1.1.5 Reset Device

COMMAND: <0x78><0x46><0x0A><stable number>

<0x78><0x46><0x0A> is the command head.

<stable number> :is defined as <0x49><0x52><0x46><0x57>

1.1.6 Enter Bootloading Mode

COMMAND: <0x78><0x46><0x7A><stable number>

Enter bootloading mode to load firmware/ application on Sign&Pay.

PARAMETERS:

<0x78><0x46><0x7A> is the command head.

<stable number> :is defined as <0x49><0x52><0x46><0x57><8-bytes 0x00>

RETURN:

<ACK>

1.2 LED Control

1.2.1 LED Control

COMMAND: <0x76><0x46><0x01><LED Control Code>

Control the Red LED (R) and Green LED (G).

PARAMETERS:

<0x76><0x46><0x01> is the command head.

< LED Control Code > 1-byte data which is defined as

MSB LSB
B7 B6 B5 B4 B3 B2 B1 B0

Every LED uses two bits:
Left LED: Green& Red B6 B5 B4
Right LED: Green &Red B2 B1 B0
Where

B7 is reserved.

B6 controls the Left LED.

B6=1: Left LED is ON, Then B4 and B5 are selected.

B6=0: Left LED is OFF, Then B4 and B5 are ignored.

B5 controls the Left LED to be Green or Red.

B5=1: Left Green LED is selected.

B5=0: Left Red LED is selected.

B4 controls the Left LED flashing. B6 must be ON to use this control

B4=1: Left Green/Red LED flash.

B4=0: Left Green/Red LED steady.

B3 is reserved.

B2 controls the Right LED.

B2=1: Right LED is ON, Then B1 and B0 are selected.

B2=0: Right LED is OFF, Then B1 and B0 are ignored.

B1 controls the Right LED to be Green or Red.

B1=1: Right Green LED is selected.

B1=0: Right Red LED is selected.

B0 controls the Right LED flashing. B2 must be ON to use this control

B0=1: Rignt Green/Red LED flash.

B0=0: Rignt Green/Red led steady.

RETURN:

<ACK>

1.3 LCD Control

1.3.1 Set Pen Width and Color

COMMAND: <0x8A><0x46><0x10><WIDTH><COLOR>

Set the pen color used to draw line on the LCD.

PARAMETERS:

<0x8A><0x46><0x10> is the command head.

<WIDTH> is the pen's width, 4 bytes long. Must be 0x01.

<COLOR> is the pen's color, defined as <RED><GREEN><BLUE>. Each is 1 byte long.

RETURN:

<ACK>

1.3.2 Draw Line

COMMAND: <0x8A><0x46><0x11><X0><Y0><X1><Y1>

Draw line from point <X0><Y0> to <X1><Y1> using the pen.

PARAMETERS:

<0x8A><0x46><0x11> is the command head.

<X0> is X-coordinate of start point, 2 bytes.

<Y0> is Y-coordinate of start point, 2 bytes.

<X1> is X-coordinate of end point, 2 bytes.

<Y1> is Y-coordinate of end point, 2 bytes.

RETURN:

<ACK>

1.3.3 Draw Rectangle

COMMAND: <0x8A><0x46><0x12><X0><Y0><X1><Y1>

Draw rectangle define by top left point <X0><Y0> and bottom right point <X1><Y1> using the pen.

PARAMETERS:

<0x8A><0x46><0x12> is the command head.

<X0> is X-coordinate of top left point, 2 bytes.

<Y0> is Y-coordinate of top left point, 2 bytes.

<X1> is X-coordinate of bottom right point, 2 bytes.

<Y1> is Y-coordinate of bottom right point, 2 bytes.

RETURN:

<ACK>

1.3.4 Draw Arc

COMMAND: <0x8A><0x46><0x13> <X><Y><Radius>< StartAngle >< SweepAngle >

Draw arc defined by center point, radius, start angle and sweep angle use pen.

PARAMETERS:

<0x8A><0x46><0x13> is the command head.

<X> specifies the x-coordinate of the center of the related circle. 2 bytes.
<Y> specifies the y-coordinate of the center of the related circle. 2 bytes.
<Radius> specifies the radius of the related circle. 2 bytes.
<StartAngle> specifies the starting angle in degrees relative to the x-axis. Unit is 0.1. 2 bytes.
<SweepAngle> specifies the sweep angle in degrees relative to the starting angle. Unit is 0.1. 2 bytes.

RETURN:
<ACK>

1.3.5 Set Brush Color

COMMAND: <0x8A><0x46><0x20><COLOR>
Set the brush's color used to fill region on the LCD.

PARAMETERS:
<0x8A><0x46><0x20> is the command head.
<COLOR> is the pen's color, defined as <RED><GREEN><BLUE>. Each is 1 byte long.

RETURN:
<ACK>

1.3.6 Fill Rectangle

COMMAND: <0x8A><0x46><0x22><X0><Y0><X1><Y1>
Fill rectangle define by top left point <X0><Y0> and bottom right point <X1><Y1> using the brush.

PARAMETERS:
<0x8A><0x46><0x22> is the command head.
<X0> is X-coordinate of top left point, 2 bytes.
<Y0> is Y-coordinate of top left point, 2 bytes.
<X1> is X-coordinate of bottom right point, 2 bytes.
<Y1> is Y-coordinate of bottom right point, 2 bytes.

RETURN:
<ACK>

1.3.7 Fill Arc

COMMAND: <0x8A><0x46><0x23> <X><Y><Radius>< StartAngle >< SweepAngle >
Draw arc defined by center point, radius, start angle and sweep angle use brush.

PARAMETERS:
<0x8A><0x46><0x23> is the command head.
<X> specifies the x-coordinate of the center of the related circle. 2 bytes.
<Y> specifies the y-coordinate of the center of the related circle. 2 bytes.
<Radius> specifies the radius of the related circle. 2 bytes.
<StartAngle> specifies the starting angle in degrees relative to the x-axis. Unit is 0.1. 2 bytes.
<SweepAngle> specifies the sweep angle in degrees relative to the starting angle. Unit is 0.1. 2 bytes.

RETURN:

<ACK>

1.3.8 Set DisplayText Font

COMMAND:<0x8A><0x46><0x40><Height><Width><Weight><Italic><Underline><CharSet>

Select the font for text display on the LCD.

PARAMETERS:

<0x8A><0x46><0x40> is the command head.

<Height> specifies the height of a char. 1 byte. This value defines the vertical gap between characters.

<Width> specifies the width of a char. 1 byte. This value defines the horizontal gap between characters.

<Weight> specifies the weight of the char. 1 byte. Must be 0x00.

<Italic> specifies the italic of the char. 1 byte. Must be 0x00 means no Italic.

<Underline> specifies the underline or not of the char. 1 byte. Must be 0x00 means no underline.

<CharSet> specifies the char set. 1 byte. The valid size is 1 –6, and the corresponding size is: 4x8, 8x16, 12x24, 16x32, 24x48, 32x64.

RETURN:

<ACK>

1.3.9 Set Text Color

COMMAND: <0x8A><0x46><0x41><COLOR>

Set the text's color

PARAMETERS:

<0x8A><0x46><0x41> is the command head.

<COLOR> is the text's color, defined as <RED><GREEN><BLUE>. Each is 1 byte long.

RETURN:

<ACK>

1.3.10 Set Background Color

COMMAND: <0x8A><0x46><0x42><COLOR>

Set the background color when display text on the LCD.

PARAMETERS:

<0x8A><0x46><0x42> is the command head.

<COLOR> is the text's color, defined as <RED><GREEN><BLUE>. Each is 1 byte long.

RETURN:

<ACK>

1.3.11 Set Background Mode

COMMAND: <0x8A><0x46><0x43><MODE>

Select the background mode for text display on the LCD.

PARAMETERS:

<0x8A><0x46><0x43> is the command head.
<MODE> specifies background mode. 1 byte. 0x00 means OPAQUE and 0x01 means TRANSPARENT.

RETURN:
<ACK>

1.3.12 Draw String In Rectangle

COMMAND: <0x8A><0x46><0x4E><X0><Y0><X1><Y1><Length><String>
Draw string using the selected font and colors on the LCD. The string will be displayed in the specified rectangle, from the top left of the rectangle to the right bottom of the rectangle.

PARAMETERS:
<0x8A><0x46><0x4E> is the command head.
<X0> specifies the x-coordinate of the top left point. 2 bytes.
<Y0> specifies the y-coordinate of the top left point. 2 bytes.
<X1> specifies the x-coordinate of the right bottom point. 2 bytes
<Y1> specifies the y-coordinate of the right bottom point. 2 bytes.
<Length> specifies the length of the string in characters. 2 bytes.
<String> specifies the string to be displayed.

RETURN:
<ACK>

1.3.13 Draw String

COMMAND: <0x8A><0x46><0x4F><X><Y><Length><String>
Draw string using the selected font and colors on the LCD.

PARAMETERS:
<0x8A><0x46><0x4F> is the command head.
<X> specifies the x-coordinate of the start point. 2 bytes.
<Y> specifies the y-coordinate of the start point. 2 bytes.
<Length> specifies the length of the string in chars. 2 bytes.
<String> specifies the string to be displayed.

RETURN:
<ACK>

1.3.14 Get Picture on LCD

COMMAND: <0x8A><0x46><0x60><X0><Y0><X1><Y1>
Get picture on the LCD defined by top left point <X0><Y0> and bottom right point <X1><Y1>.

PARAMETERS:
<0x8A><0x46><0x60> is the command head.
<X0> is X-coordinate of top left point, 2 bytes.
<Y0> is Y-coordinate of top left point, 2 bytes.
<X1> is X-coordinate of bottom right point, 2 bytes.
<Y1> is Y-coordinate of bottom right point, 2 bytes.

RETURN:

<ACK><Picture data>
Picture is arranged as top left point first and bottom right end. Each point occupies three bytes defined as: RED GREEN BLUE.

1.3.15 Show Picture on LCD

COMMAND: <0x8A><0x46><0x61><X0><Y0><X1><Y1><Picture Data>
Show picture on the LCD defined by top left point <X0><Y0> and bottom right point <X1><Y1>.

PARAMETERS:
<0x8A><0x46><0x61> is the command head.
<X0> is X-coordinate of top left point, 2 bytes.
<Y0> is Y-coordinate of top left point, 2 bytes.
<X1> is X-coordinate of bottom right point, 2 bytes.
<Y1> is Y-coordinate of bottom right point, 2 bytes.
<Picture Data> is the picture data.

RETURN:
<ACK>

1.3.16 Store Picture on Device

COMMAND: <0x8A><0x46><0x70><ID><TYPE><Picture Data>
Store pictures on the device.

PARAMETERS:
<0x8A><0x46><0x70> is the command head.
<ID> is the identifier for the picture. 2 bytes.
<TYPE> is the picture's type. 2 bytes. 0x00 means RAW format, 0x01 means 24-bit true color BMP format, 0x02 means JPEG format,
<Picture Data> is the picture data.

RETURN:
<ACK>
NOTE: <Picture Data> must be less than 32k bytes

1.3.17 Show Stored Pictures on LCD

COMMAND: <0x8A><0x46><0x71><ID><X0><Y0><X1><Y1>
Show the stored picture on the LCD defined by top left point <X0><Y0> and bottom right point <X1><Y1>.

PARAMETERS:
<0x8A><0x46><0x71> is the command head.
<ID> is the identifier for the picture. 2 bytes.
<X0> is X-coordinate of top left point, 2 bytes.
<Y0> is Y-coordinate of top left point, 2 bytes.
<X1> is X-coordinate of bottom right point, 2 bytes.
<Y1> is Y-coordinate of bottom right point, 2 bytes.

RETURN:

<ACK> if picture exists, otherwise <NAK>.

1.3.18 Retrieve Stored Picture on Device

COMMAND: <0x8A><0x46><0x72>

Retrieve stored picture om the device.

PARAMETERS:

<0x8A><0x46><0x72> is the command head.

RETURN:

<ACK><Picture count (1 byte)><ID(2 bytes)>[<ID(2 bytes)>…]

1.4 Touchpad control

1.4.1 Calibrate Device

COMMAND: <0x7A><0x46><0x01>

Calibrate the device.

PARAMETERS:

<0x7A><0x46><0x01> is the command head.

RETURN:

<ACK>

1.4.2 Set Clip Area

COMMAND: <0x7A><0x46><0x03><Clip area data(8 bytes)><Show Mode><Line Color>

Set new clip area. The max area is (0,0) – (319,239)

Clip area is a rectangle coded as: left (2 bytes) + top (2 bytes) + right (2 bytes) + bottom (2 bytes)

PARAMETERS:

<0x7A><0x46><0x03> is the command head.

<Clip area data(8 bytes)> is the new clip area data.

<Show Mode> is a bitmap for 4 lines. Bit 1 for left line, Bit 2 for right line, Bit 3 for top line and Bit 4 for bottom line. Value 1b means show this line, 0b means do not show this line.

<Line Color> is the rectangle lines (which surround the clip area) color defined as RED (1 byte) GREEN (1 byte) BLUE (1 byte).

RETURN:

<ACK>

1.4.3 Create Active Region on Screen

COMMAND:<0x7A><0x46><0x04><ID><Type><State><X0><Y0><X1><Y1><DataLen><Data>.

Create object like picture, button and text showed on LCD when during signature. The object can be notified when touched.

PARAMETERS:

<0x7A><0x46><0x04> is the command head

<ID> specifies the region's ID. 1 byte.

<Type> specifies the region's type. 1 byte. 0x01 means BUTTON, 0x02 means PICTURE and 0x03 means TEXT, 0x11 means owner draw button.

Note: when <Type> is been set to 0x03, the <DataLen> must be less than 0x0d. It should include font of text (6 bytes) and color of text (3 bytes) and back mode of LCD (1 byte) and back color of text (3 bytes) and text string.

<State> specifies the region's state. 1 byte.
State:: Bit 0 Exists.
Bit 1 Visable.
Bit 2 Enabled.
Bit 3 Notify.
<X0> is X-coordinate of top left point, 2 bytes.
<Y0> is Y-coordinate of top left point, 2 bytes.
<X1> is X-coordinate of bottom right point, 2 bytes.
<Y1> is Y-coordinate of bottom right point, 2 bytes.
<DataLen> specifies the <Data> length.
<Data> specifies the object/s data.

For Button, <Data> is the text showed on the button.
For Picture, <Data> is the picture data. Picture is arranged as top left point first and bottom right end. Each point occupies three bytes defined as: RED GREEN BLUE.
For Text, <Data> is arranged as: Font(Height 1 byte, Width 1 byte, Weight 1 byte, Italic 1 byte, Underline 1 byte, CharSet 1 byte) TextColour(RED GREEN BLUE) TextBkMode(1 byte) TextBkColour(RED GREEN BLUE) String.
For owner draw button, <Data> is: Font(Height 1 byte, Width 1 byte, Weight 1 byte, Italic 1 byte, Underline 1 byte, CharSet 1 byte) TextColour(RED GREEN BLUE) TextBkMode(1 byte) TextBkColour(RED GREEN BLUE) String Offset(X, Y 4 bytes) String.

RETURN:
<ACK>

1.4.4 Start Capture

COMMAND: <0x7A><0x46><0x10><Capture Mode><Point Interval><Signature Color><Background Color>
Start capture using specified parameters

PARAMETERS:
<0x7A><0x46><0x10> is the command head.
<Capture Mode> specifies the capture mode. 1 byte
0x01: Out signature data using FBP format, pen up is 0x8C and pen down is 0x9C.
0x02: Out signature data (the difference of the current point and previous point) using FBP format, pen up is 0x80 and pen down is 0x90.
0x03: Out signature data using CMP format.
0x04: Out signature data using FBP format, pen up is 0x80 and pen down is 0x90.

0x05: Data is buffered and not send out.

<Point Interval> specifies the maximum points' interval during signature. If exceeds, the signature will be cleared. 1 byte. The unit is second.
<Signature Color> specifies signature's color, defined as RED GREEN BLUE.
<Backgroud Color> specifies background color, defined as RED GREEN BLUE.

RETURN:
<ACK>
During capture, data will be sent out if not buffered.
Notify data for regions: 0x7A ID.
Signature data: FBP format or CMP format.

1.4.5 Continue Capture

COMMAND: <0x7A><0x46><0x11>
Continue capture using specified parameters
PARAMETERS:
<0x7A><0x46><0x11> is the command head.
RETURN:
<ACK>

1.4.6 Pause Capture

COMMAND: <0x7A><0x46><0x12>
Pause capture
PARAMETERS:
<0x7A><0x46><0x12> is the command head.
RETURN:
<ACK><Scipt point count(4 bytes)>

1.4.7 Get Script Point Count

COMMAND: <0x7A><0x46><0x17>
Get script point count
PARAMETERS:
<0x7A><0x46><0x17> is the command head.
RETURN:
<ACK><Script point count(4 bytes)>

1.4.8 Clear Signature

COMMAND: <0x7A><0x46><0x19>
Clear signature
PARAMETERS:
<0x7A><0x46><0x19> is the command head.
RETURN:

<ACK>

1.4.9 Exit Capture

COMMAND: <0x7A><0x46><0x1F>
Exit capture All regions will be deleted.
PARAMETERS:
<0x7A><0x46><0x1F> is the command head.
RETURN:
<ACK>

1.4.10 Get SIG Format Signature

COMMAND: <0x7A><0x46><0x20>
Get buffered signature SIG format data.
PARAMETERS:
<0x7A><0x46><0x20> is the command head.
RETURN:
<ACK><SIG format signature data>

1.4.11 Get CMP Format Signature

COMMAND: <0x7A><0x46><0x21>
Get buffered signature CMP format data.
PARAMETERS:
<0x7A><0x46><0x21> is the command head.
RETURN:
<ACK><CMP format signature data>

1.4.12 Get RAW Format Signature

COMMAND: <0x7A><0x46><0x22>
Get buffered signature RAW format data.
PARAMETERS:
<0x7A><0x46><0x22> is the command head.
RETURN:
<ACK><RAW format signature data>
RAW data format: Three bytes for one point: xxxxxxxx xxxxyyyy yyyyyyyy

1.4.13 Get BMP Format Signature

COMMAND: <0x7A><0x46><0x23>
Get buffered signature BMP format data.
PARAMETERS:
<0x7A><0x46><0x23> is the command head.

RETURN:
<ACK><BMP format signature data>
BMP data format: one bytes for one point: xxxxxxxx

1.4.14 Set SIG Format Signature

COMMAND: <0x7A><0x46><0x30><SIG format signature data>
Send SIG format signature to Sign&Pay.
PARAMETERS:
<0x7A><0x46><0x30> is the command head.
<SIG format signature data>
RETURN:
<ACK>

1.4.15 Set CMP Format Signature

COMMAND: <0x7A><0x46><0x31><CMP format signature data>
Send CMP format signature to Sign&Pay.
PARAMETERS:
<0x7A><0x46><0x02> is the command head.
<CMP format signature data>
RETURN:
<ACK>

1.4.16 Set RAW Format Signature

COMMAND: <0x7A><0x46><0x32><RAW format signature data>
Send RAW format signature to Sign&Pay.
PARAMETERS:
<0x7A><0x46><0x02> is the command head.
<RAW format signature data>
RETURN:
<ACK>

1.4.17 Set BMP Format Signature

COMMAND: <0x7A><0x46><0x33><BMP format signature data>
Send BMP format signature to Sign&Pay.
PARAMETERS:
<0x7A><0x46><0x33> is the command head.
<BMP format signature data>
RETURN:
<ACK>

1.5 MagStripe Reader Control

Magstripe reader is referred to as the IDTECH SecureHead in this section.

1.5.1 Get Reader Output Settings

COMMAND: <0x73><0x52><0x1A>

Get reader output settings.

PARAMETERS:

<0x73><0x52><0x1A> is the command head.

RETURN:

<NAK>or <ACK><0x1A><0x01><Securehead output settings>

<0x1A> is command.

<0x01> is length for <Securehead output Settings>.

<Securehead output Settings> is defined as follow.

0x30: SecureHead Output Disabled

0x31:SecureHead Output Enabled (default)

1.5.2 Get SecureHead Decoding Method Setting

COMMAND: <0x73><0x52><0x1D>

Get SecureHead decoding method

PARAMETERS:

<0x73><0x52><0x1D > is the command head.

RETURN:

<NAK> or <ACK><0x1D><0x01><Decoding Method Settings>

<0x1D> is command.

<0x01> is length for <Decoding Method Settings>.

<Decoding Method Settings> is defined as follow.

0x30: Raw Data Decoding in Both Directions, send out in ID TECH mode

0x31: Decoding in Both Directions**.** If the encryption feature is enabled, the key management method used is DUKPT.

0x32: Moving stripe along head in direction of encoding. If the encryption feature is enabled, the key management method used is DUKPT.

0x33: Moving stripe along head against direction of encoding. If the encryption feature is enabled, the key management method used is DUKPT.

1.5.3 Review All Settings

COMMAND: <0x73><0x52><0x1F>

Get all SecureHead settings

PARAMETERS:

<0x73><0x52>< 0x1f > is the command head.

RETURN:

<NAK> or <ACK><FuncSETBLOCK1>…<FuncSETBLOCKn>

<FuncSETBLOCK> The Format is:

<FuncID><Len><FuncData>

Where:
<FuncID> is 1-byte identifying the setting(s) for the function.
<Len> is a 1-byte length count for the following function-setting block <FuncData>
<FuncData> is the current setting for this function. It has the same format as in the sending command for this function.
<FuncSETBLOCK> are in the order of their Function ID<FuncID>

1.5.4 Get SecureHead Firmware Version

COMMAND:<0x73><0x52><0x22>
Read firmware version of SecureHead.
PARAMETERS:
<0x73><0x52><0x22> is the command head.
RETURN:
<NAK> or <ACK>< firmware version data >

1.5.5 Review SecureHead PrePANID

COMMAND: <0x73><0x52>< 0x49 >
First N Digits in PAN which can be clear data
PARAMETERS:
<0x73><0x52>< 0x49 > is the command head.
RETURN:
<NAK> or <ACK><0x49><0x01><Number>
<Number> is count in PAN which can be clear data at the first of digits.

1.5.6 Review SecureHead PostPANID

COMMAND: <0x73><0x52><0x4A>
Last M Digits in PAN which can be clear data
PARAMETERS:
<0x73><0x52>< 0x4A > is the command head.
RETURN:
<NAK> or <ACK><0x4A><0x01><Number>
<Number> is count in PAN which can be clear data at the last of digits.

1.5.7 Get SecureHead Data Masking Character

COMMAND: <0x73><0x52><0x4B>
Read character that used to mask PAN
PARAMETERS:
<0x73><0x52>< 0x4B > is the command head.
RETURN:
<NAK> or <ACK><0x4B><0x01><Character>
<Character> is used to mask PAN.

1.5.8 Get SecureHead Encryption Algorithm

COMMAND: <0x73><0x52><0x4C>

Read Security Algorithm of SecureHead.

PARAMETERS:

<0x73><0x52><0x4C> is the command head.

RETURN:

<NAK> or <ACK><0x4c><0x01><Encryption Algorithm>

<0x4C> is command.

<0x01> is length of <Encryption Algorithm>.

<Encryption Algorithm > is defined as follow:

0x30: Encryption Disabled (Only works for fixed key, cannot disable DUKPT encryption)

0x31: Enable TDES Encryption

0x32: Enable AES Encryption (Not for Raw Data Decoding in Both Directions, send out in other mode.)

1.5.9 Get SecureHead Serial Number

COMMAND: <0x73><0x52>< 0x4E >

Read Serial Number of SecureHead.

PARAMETERS:

<0x73><0x52>< 0x4E > is the command head.

RETURN:

<NAK> or <ACK><0x4E><data length><1-byte Serial Number Len><8-bytes Serial Number>

1.5.10 Get SecureHead format of display expiration data

COMMAND: <0x73><0x52>< 0x50 >

Display expiration data as mask data or clear data.

PARAMETERS:

<0x73><0x52>< 0x50 > is the command head.

RETURN:

<NAK> or <ACK><0x50><0x01><1-byte format>

<0x50> is command.

<0x01> is length of <1-byte format>

<1-byte format> is defined as follow.

0x30 Display expiration data as mask data

0x31 Display expiration data as clear data

1.5.11 Review SecureHead KSN and Counter ID

COMMAND: <0x73><0x52>< 0x51 >

Review the Key Serial Number and Encryption Counter

PARAMETERS:

<0x73><0x52>< 0x51 > is the command head.

RETURN:

<NAK> or <ACK><0x51><Parameter length><Data length><Data>
<0x51> is command.
< Parameter length > is length of <Data length> and <Data>.
<Data length> is length of <Data>.
<Data> includes the Initial Key Serial Number in the leftmost 59 bits and a value for the Encryption Counter in the right most 21 bits.

1.5.12 Get SecureHead Key Management setting

COMMAND: <0x73><0x52>< 0x58 >
Get model of key management.
PARAMETERS:
<0x73><0x52>< 0x58 > is the command head.
RETURN:
<NAK> or <ACK><0x58><0x01><Key_management>
<0x58> is command.
<0x01> is length of < Key_management >
< Key_management > is defined as follow:
0x30: Fixed Key
0x31: DUKPT Key

1.5.13 SecureHead External Authenticate Get Random Command

COMMAND: <0x73><0x52>< 0x74 >
Get 8 bytes of TDES-encrypted random data. Then use a fixed key of SecureHead to encrypte these bytes, and send the result to SecureHead. If these steps are ok, then SecureHead allow you to change the fixed key.
PARAMETERS:
<0x73><0x52>< 0x74 > is the command head.
RETURN:
<NAK> or <ACK><8 bytes of TDES-encrypted random data>

1.5.14 Get SecureHead Security Level

COMMAND: <0x73><0x52><0x7E>
Read current SecureHead Security level.
PARAMETERS:
<0x73><0x52><0x7E> is the command head.
RETURN:
<NAK> or <ACK><1-byte Function><1-byte Len><1-byte Security Level>
<1-byte Security Level> is defined as follow:
0x30: Security Level 0;
0x31: Security Level 1;
0x32: Security Level 2;
0x33: Security Level 3.

1.5.15 Change SecureHead to Default Settings

COMMAND: <0x73><0x53><18h >
Set SecureHead to default settings.
PARAMETERS:
<0x73><0x53><18h > is the command head.
RETURN:
<NAK> or <ACK>

1.5.16 SecureHead Output Settings

COMMAND: <0x73><0x53><0x1A><0x01><Securehead outputting Settings>
Enable/Disable SecureHead output.
PARAMETERS:
<0x73><0x53><0x1A> is the command head.
<0x01> is length for <Securehead outputting Settings>.
<Securehead outputting Settings> is defined as follow.
0x30: SecureHead Output Disabled
0x31:SecureHead Output Enabled
RETURN:
<NAK> or <ACK>

1.5.17 SecureHead Decoding Method Settings

COMMAND: <0x73><0x53><0x1D><0x01><Decoding Method Settings>
PARAMETERS:
<0x73><0x53><0x1D > is the command head.
<0x01> is length for <Decoding Method Settings>.

<Decoding Method Settings> is defined as follow.
0x30: Raw Data Decoding in Both Directions, send out in ID TECH mode
0x31: Decoding in Both Directions**.** If the encryption feature is enabled, the key management method used is DUKPT.
0x32: Moving stripe along head in direction of encoding. If the encryption feature is enabled, the key management method used is DUKPT.
0x33: Moving stripe along head against direction of encoding. If the encryption feature is enabled, the key management method used is DUKPT.
RETURN:
<NAK> or <ACK>

1.5.18 Set SecureHead PrePANID

COMMAND: <0x73><0x53><0x49><Number>
First N Digits in PAN which can be clear data
PARAMETERS:
<0x73><0x53>< 0x49 > is the command head.

<Number> is count in PAN which can be clear data at the first of digits.

RETURN:

<NAK> or <ACK>

1.5.19 Set SecureHead PostPANID

COMMAND: <0x73><0x53><0x4A><Number>

Last M Digits in PAN which can be clear data

PARAMETERS:

<0x73><0x53>< 0x4A > is the command head.

<Number> is count in PAN which can be clear data at the last of digits.

RETURN:

<NAK> or <ACK>

1.5.20 Set SecureHead Data Masking Character

COMMAND: <0x73><0x53><0x4B><Character>

Set character that used to mask PAN

PARAMETERS:

<0x73><0x53>< 0x4B > is the command head.

< Character > is used to mask PAN.

RETURN:

<NAK> or <ACK>

1.5.21 Enable SecureHead Encryption

COMMAND: <0x73><0x53><0x4C><0x01><Encryption Settings>

Enable or disable the SecureHead Encryption output.

PARAMETERS:

<0x73><0x53><0x4C> is the command head.

<0x01> is length for <Encryption Settings>.

<Encryption Settings> is defined as follow:

0x30: Encryption Disabled

0x31: Enable TDES Encryption

0x32: Enable AES Encryption (Not for Raw Data Decoding in Both Directions, send out in other mode.)

RETURN:

<NAK> or <ACK>

1.5.22 Set SecureHead Format of Display Expiration Data

COMMAND: <0x73><0x53><0x50><1-byte format>

Display expiration data as mask data or clear data.

PARAMETERS:

<0x73><0x53><0x50> is the command head.

<1-byte format> is defined as follow.
0x30 Display expiration data as mask data
0x31 Display expiration data as clear data

RETURN:
<NAK> or <ACK>

1.5.23 Set SecureHead Key Management

COMMAND: <0x73><0x53><0x58><0x01><Key_management>
Set model of key management.

PARAMETERS:
<0x73><0x53><0x58> is the command head.
<0x01> is length of <Key_management>
<Key_management> is defined as follow:
0x30: Fixed Key
0x31: DUKPT Key

RETURN:
<NAK> or <ACK>

1.5.24 SecureHead Send External Authenticate Command

COMMAND: <0x73><0x53><0x74><0x08><8 bytes of original random data>
Send External Authenticate data to SecureHead.

PARAMETERS:
<0x73><0x53><0x74> is the command head.
<0x08> is length for <8 bytes of original random data>.
<8 bytes of original random data>
After execute the command of SecureHead External Authenticate Get Random Command(0x73 0x52 0x74), Get 8 bytes random data and encrypt these data with Fix key of SecureHead, then get <8 bytes of original random data>.

RETURN:
<NAK> or <ACK>

1.5.25 SecureHead Load Device Key Command

COMMAND: <0x73><0x53><0x76><0x10><16 bytes Device key>
Change Fix Key. If you want to change Fix key, you must execute the command of **SecureHead External Authenticate Get Random Command,** Get 8-bytes Random Data, and encrypt Random data with Fix key of SecureHead, then send encrypt-data to SecureHead. If all is ok, this command can be executed success.

PARAMETERS:
<0x73><0x53><0x10h> is the command head.
<0x10> is length for <16 bytes Device key>.
<16 bytes Device key> is the Fixed key.

RETURN:

<NAK> or <ACK>

1.5.26 SecureHead Clear Data Output Command

COMMAND: <0x73><0x53><0x7e><Len><Fixed Data >

Erase the DUKPT key defined by user and use a random DUKPT key generated by Sign&Pay.

PARAMETERS:

<0x73><0x53><0x7e> is the command head.

< Len > is length of <Fixed Data>, The value is 0x09, 1 byte.

< Fixed Data > is defined as the following:

<0x31,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08>

RETURN:

<NAK> or <ACK>

1.5.27 SecureHead Load DUKPT Key

Note: These commands do not follow the command protocol.

Step 1 : Enable load DUKPT key

COMMAND: <0x55 0x01 0x06 0x08 0x09 0x01 0x5A >

Enable load DUKPT key command

COMMAND:< 0x55 0x01 0x06 0x08 0x01 0x01 0x5A **>**

Cancel DUKPT key loading process.

RETURN:

<NAK> or <ACK>

Step 2: Get SecureHead Key Status

COMMAND: <STX><'F'><'F'> < Command Data(Base64) > <0D>< 0A><ETX><LRC>

< Command Data(Base64) > the original data is <0xff 0x13 0x01 0x02 LRC>

Response:

<ACK/NAK><STX><'F'><'F'>< Respond Data (BASE64)><0x0D><0x0A><ETX><LRC>

<Respond Data> The original data is <0xff 0x00 0x01 0x04 LRC>

Step 3: Load SecureHead KSN

COMMAND: <STX><'F'><'F'> < Command Data(Base64) > <0D>< 0A><ETX><LRC>

< Command Data(Base64) >

the original data is <0xff> <0x0A> <Len> <KSN#><KSN DATA>

<Len>:1 bytes, the value is 0x11.

<KSN#>: 1 bytes, TDES: 0x32; DES:0x0A

<KSN DATA>:16 bytes

Response:

<ACK/NAK><STX><'F'><'F'>< Respond Data (BASE64)><0x0D><0x0A><ETX><LRC>

<Respond Data> 6 bytes data in ASCII format which is converted from the first 3 cipher hex data. These cipher data are generated by encrypting KSN bytes and "00 00 00 00 00 00 00 00".

Step 4: Load SecureHead DUKPT

COMMAND: <STX><'F'><'F'> < Command Data(Base64) > <0D>< 0A><ETX><LRC>

< Command Data(Base64) >

The original data is <0xff> <0x0A> <Len> <KEY#><KEY DATA>

<Len> : 1 bytes, TDES: 0x21; DES:0x11

<KEY#>: 1 bytes, TDES: 0x33; DES:0x0B

<KEY DATA>: TDES: 32 bytes; DES: 16 bytes.

Response:

<ACK/NAK><STX><'F'><'F'>< Respond Data (BASE64)><0x0D><0x0A><ETX> <LRC>

<Respond Data>: 6 bytes data in ASCII format which is converted from the first 3 cipher hex data. These cipher data are generated by encrypting KSN bytes and "00 00 00 00 00 00 00 00".

Notice:

<LRC>: XOR of all data except <STX>.

1.5.28 SecureaHead output structure setting

COMMAND: <0x73><0x53><0x85><Len>< Encrypt Structure >

Set securehead encrypted structure of outputting, SecureHead output structure has two format, one is default format structure which Track 1 and Track 2 is encrypted together with AES or Tri-DES, and Track 3 is clear data; the other is new format structure which every Track is individual encrypted with Aes or Tri-DES

PARAMETERS:

<0x73><0x53><0x85> is the command head.

< Len > is length of < Encrypt Structure >, The value is 0x01, 1 byte.

<Encrypt Structure> is defined as following

0x30: Default: original encrypt output structure

0x31: enhanced encrypt output structure will send bytes 8 and 9 and CardType will be 1xxxxxxx (high bit =1)

RETURN:

<NAK> or <ACK>

1.5.29 Encrypt Option Setting

COMMAND: <0x73><0x53><0x84><Len>< Encrypt Opt >

Only effect encrypted output format of card data in new structure,

PARAMETERS:

<0x73><0x53><0x84> is the command head.

< Len > is length of < Encrypt Opt >, The value is 0x01, 1 byte.

< Encrypt Opt > is defined as following

bit0: 1 – tk1 force encrypt *

bit1: 1 – tk2 force encrypt *

bit2: 1 – tk3 force encrypt *

bit3: 1 – tk3 force encrypt when card type is 0

RETURN:

<NAK> or <ACK>

Note:

1) When force encrypt is set, this track will always be encrypt, regardless of card type. No clear/mask text will be sent.

2) If and only if in new encrypt structure, each track encryption is separated, encrypted data length will round up to 8 or 16 bytes.

3) When force encrypt is not set, it encrypts data just like old structure, that is, only T1 and T2 in type zero will be encrypted.

1.5.30 Hash Option Setting

COMMAND: <0x73><0x53><0x5c><Len>< Hash Opt+0x30 >

Only effect encrypted output format of card data in new structure,

PARAMETERS:

<0x73><0x53><0x5c> is the command head.

< Len > is length of < Hash Opt >, The value is 0x01, 1 byte.

< Hash Opt + 0x30> "Hash Opt" is defined as following

bit0: 1 – tk1 hash will be sent if data is encrypted

bit1: 1 – tk2 hash will be sent if data is encrypted

bit2: 1 – tk3 hash will be sent if data is encrypted

RETURN:

<NAK> or <ACK>

1.5.31 Mask Option Setting

COMMAND: <0x73><0x53><0x86><Len>< Mask Opt >

Only effect encrypted output format of card data in new structure.

PARAMETERS:

<0x73><0x53><0x86> is the command head.

< Len > is length of < Mask Opt >, The value is 0x01, 1 byte.

< Mask Opt > is defined as following

bit0: 1 – tk1 mask data allow to send when encrypted

bit1: 1 – tk2 mask data allow to send when encrypted

bit2: 1 – tk3 mask data allow to send when encrypted

RETURN:

<NAK> or <ACK>

Note:

1) When mask option bit is set – if data is encrypted (but not forced encrypted), the mask data will be sent; If mask option is not set, the mask data will not be sent under the same condition.

1.5.32 Enable/Disable SecureHead Card Data Output

COMMAND: <0x73><0x46><0xe1><0x01><Function>

Enable or Disable SecureHead automatic output of Card data.

PARAMETERS:

<0x73><0x46><0xe1> is the command head.

<0x01> is length for <Function>,

<Function> is defined as following.

0x30: Disable SecureHead card data output

0x31: Enable SecureHead card data output, It is default after Power ON.

RETURN:

<NAK> or <ACK>

1.5.33 Get MSR Card Data Output Status

COMMAND: <0x73><0x52><0xe2>

Get SecureHead card data output status.

PARAMETERS:

<0x73><0x52><0xe2> is the command head.

RETURN:

<NAK> or <ACK><0xe2><Len><Function>

<Len> is length for < Function >, The value is 0x01, 1 byte.

< Function > is defined as the following.

0x30: SecureHead output card data with clear data.

0x31: SecureHead output card data with masked data.

1.5.34 Read MSR Card Data Output Format

COMMAND: <0x73><0x52><0xe3>

Get SecureHead card data output format.

PARAMETERS:

<0x73><0x52><0xe3> is the command head.

RETURN:

<NAK> or <ACK><0xe3><Len><Function>

<Len> is length for < Function >, The value is 0x01, 1 byte.

< Function > is defined as the following.

0x30: SecureHead output clear card data with no LRC, and there is no '0x0d' at the end of each track if the track data does not exist.

0x31: SecureHead output clear card data with LRC, and there is '0x0d' at the end of each track.

1.5.35 Set MSR Card Data Output Format

COMMAND: <0x73><53><0xe3><Len><Function>

Set SecureHead card data output format.

PARAMETERS:

<0x73><0x53><0xe3> is the command head.

<Len> is length for < Function >, The value is 0x01, 1 byte.

< Function > is defined as following.

0x30: SecureHead output clear card data with no LRC, and there is no '0x0d' at the end of each track if the track data does not exist.

0x31: SecureHead output clear card data with LRC, and there is '0x0d' at the end of each track.

RETURN:

<NAK> or <ACK>

1.5.36 Get MSR Card Encrypted Data With PIN Key Or Data Key

COMMAND: <0x73><52><0xe4>

Get the unit card endrypted data output format with PIN key or Data key. if user had load MSR DUKPT key, the data encrypt by PIN key while the <Function> is ox31, and by data key while the <Function> is 0x30.

PARAMETERS:

<0x73><0x53><0xe4> is the command head.

RETURN:

<NAK> or <ACK><0xe4><Len><Function>

<Len> is length for < Function >, The value is 0x01, 1 byte.

< Function > is defined as the following.

0x30: the unit always output clear card data if user don't load MSR DUKPT key, or output encrypted card data with Data key if user had loaded MSR DUKPT key.

0x31: the unit always output clear card data if user don't load MSR DUKPT key, or output encrypted card data with PIN key if user had loaded MSR DUKPT key.

1.5.37 Set MSR Card Encrypted Data With PIN Key Or Data Key

COMMAND: <0x73><53><0xe4><Len><Function>

Set the unit card encrypted data output format with PIN key or Data key. if user had load MSR DUKPT key, the encrypted data use PIN key while the <Function> is 0x31, and use data key while the <Function> is 0x30.

PARAMETERS:

<0x73><0x53><0xe4> is the command head.

<Len> is length for < Function >, The value is 0x01, 1 byte.

< Function > is defined as following.

0x30: the unit always output clear card data if user don't load MSR DUKPT key, or output encrypted card data with data key if user had loaded MSR DUKPT key.

0x31: the unit always output clear card data if user don't load MSR DUKPT key, or output encrypted card data with PIN key if user had loaded MSR DUKPT key.

RETURN:

<NAK> or <ACK>

1.6 PIN Pad Control

1.6.1 Get Encrypted PIN online

COMMAND: <0x75><0x46><0x07><Parameter>

Get encrypted PIN.

PARAMETERS:

<0x75><0x46><0x07> is the command head.

<Parameter> is Defined as <KeyType><max_len><min_len><Account_flag>[<Account#>]<LCD Status><BackGround Color(R G B, total 3 bytes)><Input Param> <Message Count (2 bytes)><Message1><Message2>….

<KeyType> :1 byte

- '0' – Master key / Session key (currently not implemented)
- '1' – DUKPT

<max_len(1 byte)> and <min_len(1 byte)> is the max length and min length for the amount. Max length cannot exceed12, while the min length cannot be less than 4, 4<="min_len" <= "max_len" <= 12.

< Account_flag > 0: indicates there is no account#,
1: indicates the account# is included.

<Account#> - if present, is 16 byte ASCII numeric (0x30 - 0x39)
Account number is the first 16 digits of Primary Account Number excluding the last digit, which is the check digit. In cases where there is less than 16 digits, pad the left with zeros.

If no account# is entered, use command 1.6.12 Get Card Account to obtain account number from a card swipe.

<LCD Status> 0x01 means clear display only. 0x02 means show message only. 0x03 means clear display and show message.

<Input Param>is defined: <Font(6 bytes)><Text Color(R G B, total 3 bytes)><Background Mode(1 byte)><Backgroud color(R G B, total 3 bytes)> <X0 (2 bytes)><Y0 (2 bytes)><X1 (2 bytes)><Y1 (2 bytes)><Show Mode (1 byte)>

<MessageX> is defined: <Message Length includes self (2 bytes)><Font(6 bytes)><Text Color(R G B, total 3 bytes)><Background Mode(1 byte)><Backgroud color(R G B, total 3 bytes)> <X(2 bytes)><Y(2 bytes)><String Length(2 bytes)><String>

<X0 (2 bytes)><Y0 (2 bytes)><X1 (2 bytes)><Y1 (2 bytes)> is scope of LCD, 0<="X0" < "X1"< 320, 0<="Y0" < "Y1" < 240.

<Show Mode> is a bitmap for 4 lines. Bit 0 for left line, Bit 1 for right line, Bit 2 for top line and Bit 3 for bottom line, Bit 4~7 reserve. Value 1b means show this line, 0b means do not show this line.

For master/session: <Encrypted PIN block> if succeeded

- Encrypted PIN block: 16 bytes ASCII characters

RETURN:

<NAK> or <ACK><Encrypted data>

< Encrypted data > is encrypted

Get Encrypted PIN Online Command Example

Initially Loaded Key Serial Number (KSN): FFFF9876543210E00000
Initially Loaded PIN Entry Device Key: 6AC292FAA1315B4D 858AB3A3D7D5933A
PIN: 1234
Primary Account Number: 4012345678909 (last digit "9" is the check digit and should not be sent)
Encrypted PIN block output - 1B9C1845EB993A7A

```
02                                                          //STX
80 00                                                       //Length
75 46 07                                                    //command byte
31                                                          //DUKPT
0c                                                          //max PIN length
04                                                          //min PIN length
01                                                          //with account number
30 30 30 30 34 30 31 32 33 34 35 36 37 38 39 30             //account number (no check digit)
03                                                          //clear display and show message
ff ff cc                                                    // BackGround Color
16 18 00 00 00 05
        //font<Height><Width><Weight><Italic><Underline><CharSet>
00 00 00                                                    //PIN color color
01                                                          // TRANSPARENT
ff ff ff                                                    //font color
20 00                                                       //X0
60 00                                                       //Y0
10 01                                                       //X1
90 00                                                       //Y1
0f                                                          // Display 4 lines.
02 00                                                       //two message
1f 00                                                       //message length is 31
10 10 00 00 00 03
        //font<Height><Width><Weight><Italic><Underline><CharSet>
00 00 ff                                                    //Text <RED><GREEN><BLUE>
01                                                          // TRANSPARENT.
ff ff ff                                                    //background color
40 00                                                       //X
20 00                                                       //Y
0a 00                                                       //String length
45 6e 74 65 72 20 50 49 4e 3a                               //Enter PIN:
2e 00                                                       //message length is 46
0c 0c 00 00 00 03                                           //
font<Height><Width><Weight><Italic><Underline><CharSet>
00 00 ff                                                    //Text <RED><GREEN><BLUE>
01                                                          // TRANSPARENT.
ff ff ff                                                    //Back ground color
06 00                                                       //X
b4 00                                                       //Y
19 00                                                       //String length
50 72 65 73 73 20 45 6e 74 65 72 20 4b 65 79 20 57 68 65 6e 20 44 6f 6e 65   //Press Enter Key When Done
46 32                                                       //Checksum and LRC
03                                                          //ETX
```

The response from the unit is

```
02 01 00 06 06 06 03                                        //ACK
```

02 25 00 06 46 46 46 46 39 38 37 36 35 34 33 32 31 30 45 30 30 30 30 31 31 42 39 43 31 38 34 35 45 42 39 39 33 41 37 41 xx xx 03 //Encrypted PIN block in ASCII, xx xx are LRC

1.6.2 Cancel Command Get Encrypted PIN Online or Get Numeric Key

COMMAND: <0x75><0x46><0x09>

Cancel command **Get encrypted PIN online** or **Get numeric key**

PARAMETERS:

<0x75><0x46><0x09> is the command head.

RETURN:

<ACK>

1.6.3 Activate PinPad

step 1

COMMAND: <0x75><0x46><0x0D><0x00>

Activate PinPad step 1: get 16-byte random number.

PARAMETERS:

<0x75><0x46><0x0D><0x00> is the command head.

RETURN:

<ACK><16 bytes Random Number>

step 2

COMMAND: <0x75><0x46><0x0D><0x01><Encrypted data>

Activate PinPad step 2

PARAMETERS:

<0x75><0x46><0x0D><0x01> is the command head.

<Encrypted data> is encrypted data for “IDTECHSH” using 16 bytes random got in step 1.

RETURN:

<ACK> or <NAK>

1.6.4 Set Public Key

COMMAND: <0x75><0x46><0x16><128 bytes n><128 bytes d><20bytes hash with sha-1>

Set public key used in setting numeric key and account key.

PARAMETERS:

<0x75><0x46><0x16> is the command head.

<128 bytes n>

<128 bytes d>

<20bytes hash with sha-1>

RETURN:

<ACK> or <NAK>

1.6.5 Set Numeric Key

COMMAND: <0x75><0x46><0x18><384 bytes encrypted data>

Set numeric key. This key will be used by command **Get pinpad input AS numeric** and Get pinpad input AS amount.

PARAMETERS:

<0x75><0x46><0x18> is the command head.

<384 bytes encrypted data> Original data include two field of <128 bytes encrypted n> and <128 bytes encrypted e>, they are defined as follow.

<128 bytes encrypted n> is encrypted by public key. The valid original data length must be no more than 1024 bits.

<128 bytes encrypted e> is encrypted by public key. The valid original data length must be no more than 128 bits.

RETURN:

<ACK> or <NAK>

1.6.6 Set Account Key

COMMAND: <0x75><0x46><0x19><384 bytes encrypted data>

Set account key. This key will be used by command **Get encrypted PIN online**.

PARAMETERS:

<0x75><0x46><0x19> is the command head.

<384 bytes encrypted data> Original data include two field of <128 bytes encrypted n> and <128 bytes encrypted e>, they are defined as follow.

<128 bytes encrypted n> is encrypted by public key. The valid original data length must be no more than 1024 bits.

<128 bytes encrypted e> is encrypted by public key. The valid original data length must be no more than 128 bits.

RETURN:

<ACK> or <NAK>

1.6.7 Set Firmware Key

COMMAND: <0x75><0x46><0x20><384 bytes encrypted data>

Set firmware key. This key will be used by command **firmware bootload**.

PARAMETERS:

<0x75><0x46><0x20> is the command head.

<384 bytes encrypted data> Original data include two field of <128 bytes encrypted n> and <128 bytes encrypted e>, they are defined as follow.

<128 bytes encrypted n> is encrypted by public key. The valid original data length must be no more than 1024 bits.

<128 bytes encrypted e> is encrypted by public key. The valid original data length must be no more than 128 bits.

RETURN:

<ACK> or <NAK>

1.6.8 Set Check Value

COMMAND: <0x75><0x46><0x21><128 bytes check value >

Set check value. This value will be used by firmware and bootload.

PARAMETERS:

<0x75><0x46><0x21> is the command head.

<128 bytes check value >

RETURN:

<ACK> or <NAK>

1.6.9 Get Bootloader Check Value

COMMAND: <0x75><0x46><0x33>

Get the bootloader check value stored in device.

PARAMETERS:

<0x75><0x46><0x33> is the command head.

RETURN:

<NAK> or <ACK><8-bytes Bootloader check value>

1.6.10 Get Pinpad Input as Numeric

COMMAND: <0x75><0x46><0x22><max_len><min_len><Parameter>++

Get numeric pinpad input.

PARAMETERS:

<0x75><0x46><0x22> is the command head.

<max_len> and <min_len> is the max length and min length for amount. Max length cannot exceed 20, while the min length cannot be less than 1, 0<"min_len"<="max_len"<=20.

<Parameter> is encrypted data using numeric key. Be make sure the original data length must be less than 380 bytes.

The original data is: <LCD Status 1 byte><BackGround Color(R G B, total 3 bytes)><Input Param 22 bytes> <Message Count (2 bytes)><Message1><Message2>…..

<Input Param>is defined: <Font(6 bytes)><Text Color(R G B, total 3 bytes)><Background Mode(1 byte)><Backgroud color(R G B, total 3 bytes)> <X0 (2 bytes)><Y0 (2 bytes)><X1 (2 bytes)><Y1 (2 bytes)><Show Mode (1 byte)>

<MessageX> is defined: <Message Length includes self (2 bytes)><Font(6 bytes)><Text Color(R G B, total 3 bytes)><Background Mode(1 byte)><Backgroud color(R G B, total 3 bytes)><X(2 bytes)><Y(2 bytes)><String Length(2 bytes)><String>

<LCD Status> 0x01 means clear display only. 0x02 means show message only. 0x03 means clear display and show message.

<X0 (2 bytes)><Y0 (2 bytes)><X1 (2 bytes)><Y1 (2 bytes)> is scope of LCD, 0<="X0" < "X1"< 320, 0<="Y0" < "Y1" < 240.

<Show Mode> is a bitmap for 4 lines. Bit 0 for left line, Bit 1 for right line, Bit 2 for top line and Bit 3 for bottom line. Value 1b means show this line, 0b means do not show this line.

RETURN:

<ACK> or <NAK>

Input is 16 bytes <len><keys0><keys1>…<FF>

Where:
len: the number of numeric keys.
keys0, keys1 … : two numeric keys, every key is the nibble of this keys. Besides, 'F' is the padding of the keys.

1.6.11 Get Pinpad Input as Amount

COMMAND: <0x75><0x46><0x23><max_len><min_len><Parameter>
Get pinpad input AS AMOUNT.
PARAMETERS:
<0x75><0x46><0x23> is the command head.
<max_len> and <min_len> is the max length and min length for amount. Max length cannot exceed 20, while the min length cannot be less than 1, 0<"min_len"<="max_len"<=20.
<Parameter> is encrypted data using numeric key. Be make sure the original data length must be less than 380 bytes.
The original data is: <LCD Status><BackGround Color(R G B, total 3 bytes)><Input Param> <Message Count (2 bytes)><Message1><Message2>…..
<Input Param>is defined: <Font(6 bytes)><Text Color(R G B, total 3 bytes)><Background Mode(1 byte)><Backgroud color(R G B, total 3 bytes)> <X0 (2 bytes)><Y0 (2 bytes)><X1 (2 bytes)><Y1 (2 bytes)><Show Mode (1 byte)>
<MessageX> is defined: <Message Length includes self (2 bytes)><Font(6 bytes)><Text Color(R G B, total 3 bytes)><Background Mode(1 byte)><Backgroud color(R G B, total 3 bytes)><X(2 bytes)><Y(2 bytes)><String Length(2 bytes)><String>
<LCD Status> 0x01 means clear display only. 0x02 means show message only. 0x03 means clear display and show message.
<X0 (2 bytes)><Y0 (2 bytes)><X1 (2 bytes)><Y1 (2 bytes)> is scope of LCD, 0<="X0" < "X1"< 320, 0<="Y0" < "Y1" < 240.
<Show Mode> is a bitmap for 4 lines. Bit 0 for left line, Bit 1 for right line, Bit 2 for top line and Bit 3 for bottom line. Value 1b means show this line, 0b means do not show this line.

RETURN:
<ACK> or <NAK>
Input is 16 bytes <len><keys0><keys1>…<FF>
Where:
len: the number of numeric keys.
keys0, keys1 … : two numeric keys, every key is the nibble of this keys. Besides, 'F' is the padding of the keys.

1.6.12 Get Card Account

COMMAND: <0x75><0x46><0x24><Parameter>
Get card Account
PARAMETERS:
<0x75><0x46><0x24> is the command head.
<Parameter> defined as follow:

<max_len><min_len><LCD Status><BackGround Color(R G B, total 3 bytes)><Input Param> <Message Count (2 bytes)><Message1><Message2>…..

<Input Param>is defined: <Font(6 bytes)><Text Color(R G B, total 3 bytes)><Background Mode(1 byte)><Backgroud color(R G B, total 3 bytes)> <X0 (2 bytes)><Y0 (2 bytes)><X1 (2 bytes)><Y1 (2 bytes)><Show Mode (1 byte)>

<MessageX> is defined: <Message Length includes self (2 bytes)><Font(6 bytes)><Text Color(R G B, total 3 bytes)><Background Mode(1 byte)><Backgroud color(R G B, total 3 bytes)><X(2 bytes)><Y(2 bytes)><String Length(2 bytes)><String>

<max_len> and <min_len> is the max length and min length for amount. Max length cannot exceed 20, while the min length cannot be less than 12.

<LCD Status> 0x01 means clear display only. 0x02 means show message only. 0x03 means clear display and show message.

<X0 (2 bytes)><Y0 (2 bytes)><X1 (2 bytes)><Y1 (2 bytes)> is scope of LCD, 0<="X0" < "X1"< 320, 0<="Y0" < "Y1" < 240.

<Show Mode> is a bitmap for 4 lines. Bit 0 for left line, Bit 1 for right line, Bit 2 for top line and Bit 3 for bottom line. Value 1b means show this line, 0b means do not show this line.

RETURN:

<NAK> or <ACK><Encrypted Data><KSN>

Where

<KSN> is a 10 bytes string, in the case of fixed key management, use serial number plus two bytes null characters instead of KSN.

1.6.13 Get Encrypted Data

COMMAND: <0x75><0x46><0x25><Parameter>

Get encrypted data from keypad.

PARAMETERS:

<0x75><0x46><0x25> is the command head.

<Parameter> is defined: <End flag><max_len><min_len><LCD Status><BackGround Color(R G B, total 3 bytes)><Input Param> <Message Count (2 bytes)><Message1><Message2>…..

<End flag> End_flag = 0 : Not to send back encrypted key entry. Must followed by another Get encrypted data command.

End_flag = 1: Final key entry command. Data will be encrypted and sent back.

<Input Param>is defined: <Font(6 bytes)><Text Color(R G B, total 3 bytes)><Background Mode(1 byte)><Backgroud color(R G B, total 3 bytes)> <X0 (2 bytes)><Y0 (2 bytes)><X1 (2 bytes)><Y1 (2 bytes)><Show Mode (1 byte)>

<MessageX> is defined: <Message Length includes self (2 bytes)><Font(6 bytes)><Text Color(R G B, total 3 bytes)><Background Mode(1 byte)><Backgroud color(R G B, total 3 bytes)><X(2 bytes)><Y(2 bytes)><String Length(2 bytes)><String>

<Show Mode> is a bitmap for 4 lines. Bit 0 for left line, Bit 1 for right line, Bit 2 for top line and Bit 3 for bottom line. Value 1b means show this line, 0b means do not show this line.

<max_len> and <min_len> is the max length and min length for strings. Max length cannot exceed 20; while the min length cannot be less than 1, 0< "min_len" <= "max_len"<=20.

<LCD Status> 0x01 means clear display only. 0x02 means show message only. 0x03 means clear display and show message.

Note: This function is for encrypting the data entered from the key pad. This function will end if the end_flag is 1 and the enter key or cancel key is pressed.

The data in buffer will be erased if a different command is sent or there is a 30 seconds time out between two commands.

Maximum number of Get encrypted data command allowed is 10.
If there is an error or invalid command is sent, the data in buffer will also be erased.

RETURN:

<NAK> or <ACK> or <ACK><Encrypted Data><KSN>

Where

<KSN> is a 10 bytes string,

<Encrypted Data>: Entered data encrypted by DUKPT 3DES key. When the encrypted data is decrypted, it contains cleat text of all the keys entered separated by '/'.

Format: <First entered key string> '/' <Second entered key string> '/' ....<Last entered key string> .

This command will response <0x06> as soon as the unit received correct command.

When "Enter" key is pressed:
If end_flag = 0 - Return value = <ACK>
If end_flag = 1 - Return value = <ACK><encrypted data><KSN>
When Cancel key is pressed:
Return: <NAK>

1.6.14 Check DUKPT Key

COMMAND: <0x75><0x46><0x34>

Check to see if the DUKPT key exists.

PARAMETERS:

<0x75><0x46><0x34> is the command head.

RETURN:

<ACK><0x00> if no key exists.
<ACK><0x01> if key exists.

1.6.15 Get FPGA Version

COMMAND: <0x75><0x46><0x36>

Get FPGA version

PARAMETERS:

<0x75><0x46><0x36> is the command head.

RETURN:

<ACK><Version 1 byte>

1.6.16 Get Key Pad buffered non-numeric key

COMMAND: <0x75><0x46><0x81>

Get one Key Pad buffered pressed non-numeric key.

PARAMETERS:

<0x75><0x46><0x81> is the command head.

RETURN:

<ACK><Buffered Key 1 byte> or <ACK><0x00> if no key pressed.

1.6.17 Clear Key Pad buffer

COMMAND: <0x75><0x46><0x82>

Clear Key Pad buffer

PARAMETERS:

<0x75><0x46><0x82> is the command head.

RETURN:

<ACK>

1.6.18 Invalidate Public Key

COMMAND: <0x75><0x46><0x84>

Make the numeric key and account key invalid.

PARAMETERS:

<0x75><0x46><0x84> is the command head.

RETURN:

<ACK>

1.6.19 Manual Input Card Data

COMMAND: <0x75><0x46><0x40>

Get manual input card data, only support format of new structure with PIN key, and no LRC to set for every track data.

PARAMETERS:

<0x75><0x46><0x40> is the command head.

RETURN:

<ACK>

When finish inputting, then return data as follow.

Output ISO/ABA card data format, please see title 2.4.

1.7 Audio Control

1.7.1 Audio Control

COMMAND:<0x7B><0x46><0x01><0x00/0x01>

Sign&Pay audio Control. 0x00 means OFF and 0x01 ON.

PARAMETERS:

<0x7B><0x46><0x01> is the command head.

RETURN: <ACK>

1.7.2 Generate Tone

COMMAND:<0x7B><0x46><0x02><Frequency Low Byte>< Frequency High Byte><Duration Low Byte><Duration High Byte>

PARAMETERS:

5 < Frequency < 40,000

RETURN: <ACK>

2.0 Magstripe Card Data Output Format

2.1 Unencrypted MSR Data Output Format

Track 1: <SS1><T_1 Data><ES><CR>*
Track 2: <SS2><T_2 Data><ES><CR>*
Track 3: <SS3><T_3 Data><ES><CR>*

where: SS1(start sentinel track 1) = %
SS2(start sentinel track 2) = ;
SS3(start sentinel track 3) = ; for ISO, ! for CDL, % for AAMVA
ES(end sentinel all tracks) = ?

Start or End Sentinel: Characters in encoding format which come before the first data character (start) and after the last data character (end), indicating the beginning and end, respectively, of data.

Track Separator: A designated character which separates data tracks.

Terminator: A designated character which comes at the end of the last track of data, to separate card reads.

LRC: Check character, following end sentinel.

CDL: Old California Drivers License format.

CR: Carriage Return.

**Note: The <CR> characters (shown above) between tracks 1 & 2 and 2 & 3 denote the default character for this position, the Track Separator position. The <CR> characters shown for track 3 denotes the default character for this position, the Terminator position.*

Unencrypted MSR setting:
0x30: clear text card data with no LRC, '0x0d' at the end of each data track only if it exists (default setting)

0x31: clear text card data with LRC, '0x0d' at the end of each track when the track data does not exist.

2.2 Encrypted MSR Data Output Format

2.2.1 Original Encryption Format

<STX><LenL><LenH><Card Data><CheckLRC><CheckSum><ETX>

Where <STX> = 02h, <ETX> = 03h
<LenL><LenH> is a two byte length of <Card Data>.
<CheckLRC> is a one byte Exclusive-OR sum calculated for all <Card Data>.
<CheckSum> is a one byte Sum value calculated for all <Card data>.
<Card Data> card data format is shown below.

ISO/ABA Data Output Format:

- card encoding type (0: ISO/ABA, 4: for Raw Mode)
- track status (bit 0,1,2:T1,2,3 decode, bit 3,4,5:T1,2,3 sampling)
- track 1 unencrypted length (1 byte, 0 for no track1 data)
- track 2 unencrypted length (1 byte, 0 for no track2 data)
- track 3 unencrypted length (1 byte, 0 for no track3 data)
- track 1 masked (Omitted if in Raw mode)
- track 2 masked (Omitted if in Raw mode)
- track 3 data (Omitted if in Raw mode)
- track 1 encrypted (AES/TDES encrypted data)
- track 2 encrypted (AES/TDES encrypted data)
- track 3 encrypted (Only used in Raw mode)
- track 1 hashed (20 bytes SHA1-Xor)
- track 2 hashed (20 bytes SHA1-Xor)
- DUKPT serial number (10 bytes)

Non ISO/ABA Data Output Format

- card encoding type (1: AAMVA, 3: Others)
- track status (bit 0,1,2:T1,2,3 decode, bit 3,4,5:T1,2,3 sampling)
- track 1 length (1 byte, 0 for no track1 data)
- track 2 length (1 byte, 0 for no track2 data)
- track 3 length (1 byte, 0 for no track3 data)
- track 1 data
- track 2 data
- track 3 data

2.2.2 Original Encryption Format Decryption Example

Decryption of a three track ABA card with the original encryption format.
Sign&Pay with default settings

Original encryption format can be recognized because the high bit of the fourth byte underlined (00) is 0.

027D0100 3F48236B252A343236362A2A2A2A2A2A2A2A393939395E42555348204A522F47454
F52474520572E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2
A2A2A2A2A2A2A2A3F2A3B343236362A2A2A2A2A2A2A2A393939393D2A2A2A2A2A2A2A2A
2A2A2A2A2A2A2A2A3F2A3B3333333333333333333333736373637363037303730373736373637363
3333333333333333333373637363736303730373037373637363736333333333333333333333373637
3637363037303730373736373637363333333333333333333333736373637363037303730373F32863E9E
3DA28E455B28F7736B77E47A64EDDA3BF03A06E44F31D1818C0BCD7A353FB1AD70EFD30
FFC3DA08A4FBC9372E57E8B40848BAEAA3FE724B3550E2F4B223E6BF264BEAE9E39142B
648CDB51FB8DAF8EA5B63913D29419B67582FCCCE9B372660F03668CC453216D9449C6B67
EF33418AC88F65E1DB7ED4D10973F99DFC8463FF6DF113B6226C4898A9D355057ECAF11A
5598F02CA3162994901190000000001399F03

STX, Length (LSB, MSB), card type, track status, length track 1, length track 2, length track 3
02 7D01 00 3F 48 23 6B

The above broken down and interpreted
02—STX character
7D—low byte of total length
01—high byte of total length
00—card type byte (interpretation old format ABA card)
3F—3 tracks of data all good
48—length of track 1
23—length of track 2
6B—length of track 3

Track 1 data masked (length 0x48)
252A343236362A2A2A2A2A2A2A2A393939395E42555348204A522F47454F52474520572E4D5
25E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A3
F2A

Track 2 data in hex masked (length 0x23)
3B343236362A2A2A2A2A2A2A2A393939393D2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A3F2
A

Track 3 data unencrypted (length 0x6B)

3B33333333333333333333736373637363037303730373736373637363333333333333333333337
36373637363037303730373736373637363333333333333333333337363736373630373037303737
63736373633333333333333333333337363736373630373037373F32

Track 1 & 2 encrypted length 0x48+0x23 rounded up to 8 bytes =0x6B -> 0x70 (112 decimal)
863E9E3DA28E455B28F7736B77E47A64EDDA3BF03A06E44F31D1818C0BCD7A35
3FB1AD70EFD30FFC3DA08A4FBC9372E57E8B40848BAEAA3FE724B3550E2F4B22
3E6BF264BEAE9E39142B648CDB51FB8DAF8EA5B63913D29419B67582FCCCE9B3
72660F03668CC453216D9449C6B67EF3

Track 1 hashed
3418AC88F65E1DB7ED4D10973F99DFC8463FF6DF

Track 2 hashed
113B6226C4898A9D355057ECAF11A5598F02CA31

KSN
62994901190000000001

LRC, checksum and ETX
39 9F 03

Masked Data:
Track 1 data masked in ASCII:
%*4266********9999^BUSH JR/GEORGE W.MR^********************************?*
Track 2 data masked in ASCII:
;4266********9999=****************?*
Track 3 data unencrypted in ASCII:
;33333333376767607070776767633333333337676760707077676763333333333767676070707767
76763333333333767676070?2

Key Value: F8 2A 7A 0D 7C 67 46 F1 96 18 9A FB 54 2C 65 A3
KSN: 62 99 49 01 19 00 00 00 00 01

Decrypted Data in ASCII:
%B4266841088889999^BUSH JR/GEORGE
W.MR^080910110000110000000046000000?!;4266841088889999=0809101100000046?0
;33333333376767607070776767633333333337676760707077676763333333333767676070707767
76763333333333767676070?2

Decrypted Data in Hex:
254234323636383431303838383839393935E42555348204A522F47454F52474520572E4D525E3
038303931303131303030303131303030303030303034363030303030303F213B3432363638343
13038383838393939393D303830393130311303030303034363F300000000000

2.2.3 Enhanced Encryption Format

This mode is used when all tracks must be encrypted, or encrypted OPOS support is required, or when the tracks must be encrypted separately or when cards other than type 0 (ABA bank cards) must be encrypted or when track 3 must be encrypted. This format is the standard encryption format, but not yet the default encryption format.

Card data is sent out in the following format
<STX><LenL><LenH><Card Data><CheckLRC><CheckSum><ETX>

0	STX
1	Data Length low byte
2	Data Length high byte
3	Card Encode Type[1]
4	Track 1-3 Status[2]
5	Track 1 data length
6	Track 2 data length
7	Track 3 data length
8	Clear/masked data sent status [3]
9	Encrypted/Hash data sent status [4]
10	Track 1 clear/mask data
	Track 2 clear/mask data
	Track 3 clear/mask data
	Track 1 encrypted data
	Track 2 encrypted data
	Track 3 encrypted data
	Session ID (8 bytes) (Security level 4 only)
	Track 1 hashed (20 bytes each) (if encrypted and hash track 1 allowed)
	Track 2 hashed (20 bytes each) (if encrypted and hash track 2 allowed)
	Track 3 hashed (20 bytes each) (if encrypted and hash track 3 allowed)
	KSN (10 bytes)
	CheckLRC
	CheckSum
	ETX

Where <STX> = 02h, <ETX> = 03h

Note 1 : Card Encode Type

Card Type will be 8x for enhanced encryption format and 0x for original encryption format

Value	Encode Type Description
00h / 80h	ISO/ABA format
01h / 81h	AAMVA format

03h / 83h Other
04h / 84h Raw; un-decoded format

For Type 04 or 84 Raw data format, all tracks are encrypted and no mask data is sent. No track indicator '01', '02' or '03' in front of each track. Track indicator '01','02' and '03' will still exist for non-encrypted mode.

Note 2: Track 1-3 status byte

Field 4:
Bit 0: 1— track 1 decoded data present
Bit 1: 1— track 2 decoded data present
Bit 2: 1— track 3 decoded data present
Bit 3: 1— track 1 sampling data present
Bit 4: 1— track 2 sampling data present
Bit 5: 1— track 3 sampling data present
Bit 6, 7 — Reserved for future use

Note 3: Clear/mask data sent status

Field 8 (Clear/mask data sent status) and field 9 (Encrypted/Hash data sent status) will only be sent out in enhanced encryption format.

Field 8: Clear/masked data sent status byte:
Bit 0: 1 —track 1 clear/mask data present
Bit 1: 1— track 2 clear/mask data present
Bit 2: 1— track 3 clear/mask data present
Bit 3: 0— reserved for future use
Bit 4: 0— reserved for future use
Bit 5: 0— reserved for future use

Note 4: Encrypted/Hash data sent status

Field 9: Encrypted data sent status
Bit 0: 1— track 1 encrypted data present
Bit 1: 1— track 2 encrypted data present
Bit 2: 1— track 3 encrypted data present
Bit 3: 1— track 1 hash data present
Bit 4: 1— track 2 hash data present
Bit 5: 1— track 3 hash data present
Bit 6: 1—session ID present
Bit 7: 1—KSN present

2.2.4 Enhanced Encryption Format Decryption Example

Example of decryption of a three track ABA card with the enhanced encryption format. Sign&Pay with default settings except enhanced encryption structure format.

Enhanced encryption Format (this can be recognized because the high bit of the fourth byte underlined (80) is 1.

029801803F48236B03BF252A343236362A2A2A2A2A2A2A2A393939395E42555348204A522F4
7454F52474520572E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A
2A2A2A2A2A2A2A2A2A3F2A3B343236362A2A2A2A2A2A2A2A393939393D2A2A2A2A2A2A2A
A2A2A2A2A2A2A2A2A2A3F2ADA7F2A52BD3F6DD8B96C50FC39C7E6AF22F06ED1F033BE0F
B23D6BD33DC5A1F808512F7AE18D47A60CC3F4559B1B093563BE7E07459072ABF8FAAB5
338C6CC8815FF87797AE3A7BEAB3B10A3FBC230FBFB941FAC9E82649981AE79F2632156E
775A06AEDAFAF6F0A184318C5209E55AD44A9CCF6A78AC240F791B63284E15B4019102BA
6C505814B585816CA3C2D2F42A99B1B9773EF1B116E005B7CD8681860D174E6AD316A0EC
DBC687115FC89360AEE7E430140A7B791589CCAADB6D6872B78433C3A25DA9DDAE83F12
FEFAB530CE405B701131D2FBAAD970248A456000933418AC88F65E1DB7ED4D10973F99DF
C8463FF6DF113B6226C4898A9D355057ECAF11A5598F02CA31688861C157C1CE2E0F72CE0
F3BB598A614EAABB162994901190000000000206E203

STX, Length(LSB, MSB), card type, track status, length track 1, length track 2, length track 3
02 9801 80 3F 48-23-6B 03BF

The above broken down and interpreted
02—STX character
98—low byte of total length
01—high byte of total length
80—card type byte (interpretation new format ABA card)
3F—3 tracks of data all good
48—length of track 1
23—length of track 2
6B—length of track 3
03—tracks 1 and 2 have masked/clear data
BF—bit 7=1—KSN included
Bit 6=0—no Session ID included so not level 4 encryption
Bit 5=1—track 3 hash data present
Bit 4=1—track 2 hash data present
Bit 3-1—track 1 hash data present
Bit 2=1—track 3 encrypted data present
Bit 1=1—track 2 encrypted data present
Bit 0=1—track 1 encrypted data present

Track 1 data masked (length 0x48)
252A343236362A2A2A2A2A2A2A393939395E42555348204A522F47454F52474520572E4D5
25E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A3
F2A

Track 1 masked data in ASCII
%*4266********9999^BUSH JR/GEORGE W.MR^********************************?*

Track 2 data in hex masked (length 0x23)
3B343236362A2A2A2A2A2A2A393939393D2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A3F2
A

Track2 masked data in ASCII
;4266********9999=****************?*

In this example there is no Track 3 data either clear or masked (encrypted and hashed data is below)

Track 1 encrypted length 0x48 rounded up to 8 bytes = 0x48 (72 decimal)
DA7F2A52BD3F6DD8B96C50FC39C7E6AF22F06ED1F033BE0FB23D6BD33DC5A1F8
08512F7AE18D47A60CC3F4559B1B093563BE7E07459072ABF8FAAB5338C6CC88
15FF87797AE3A7BE

Track 2 encrypted length 0x32 rounded up to 8 bytes =0x38 (56 decimal)
AB3B10A3FBC230FBFB941FAC9E82649981AE79F2632156E775A06AEDAFAF6F0A
184318C5209E55AD

Track 3 encrypted length 0x6B rounded up to 8 bytes =0x70 (64 decimal)
44A9CCF6A78AC240F791B63284E15B4019102BA6C505814B585816CA3C2D2F42
A99B1B9773EF1B116E005B7CD8681860D174E6AD316A0ECDBC687115FC89360A
EE7E430140A7B791589CCAADB6D6872B78433C3A25DA9DDAE83F12FEFAB530CE
405B701131D2FBAAD970248A45600093

Track 1 data hashed length 20 bytes
3418AC88F65E1DB7ED4D10973F99DFC8463FF6DF

Track 2 data hashed length 20 bytes
113B6226C4898A9D355057ECAF11A5598F02CA31

Track 3 data hashed length 20 bytes
688861C157C1CE2E0F72CE0F3BB598A614EAABB1

KSN length 10 bytes
62994901190000000002

LCR, check sum and ETX
06E203

Clear/Masked Data in ASCII:
Track 1: %*4266********9999^BUSH JR/GEORGE
W.MR^********************************?*
Track 2: ;4266********9999=***************?*

Key Value: 1A 99 4C 3E 09 D9 AC EF 3E A9 BD 43 81 EF A3 34
KSN: 62 99 49 01 19 00 00 00 00 02

Decrypted Data:
Track 1 decrypted
%B4266841088889999^BUSH JR/GEORGE W.MR^080910110000110000000046000000?!
Track 2 decrypted
;4266841088889999=08091011000046?0
Track 3 decrypted
;33333333376767607070776767633333333337676760707077676763333333333767676070707767676333333333376760707077?2

Track 1 decrypted data in hex including padding zeros (but there are no pad bytes here)
2542343236363834313038383838393939395E42555348204A522F47454F52474520572E4D525E3
038303931303131303030303131303030303030303030343630303030303F21

Track 2 decrypted data in hex including padding zeros
3B34323636383431303838383839393939D30383039313031313030303034363F300000000000

Track 3 decrypted data in hex including padding zeros
3B33333333333333333333373637363736303730373037373637363736333333333333333333333337
3637363736303730373037373637363736333333333333333333333373637363736303730373037373
6373637363333333333333333333333373637363736303730373F320000000000

3.0 List of Error Code

ERROR_PARAMETER	0xE100
ERROR_LOWOUTBUFFER	0xE200
ERROR_CARD_NOT_FOUND	0xE300
ERROR_COLLISION_CARD_EXIST	0xE400
ERROR_TOOMANY_CARDS_EXIST	0xE500
ERROR_SAVED_DATA_NOT_EXIST	0xE600
ERROR_NO_DATA_AVAILABLE	0xE800
ERROR_INVALID_CID_RETURNED	0xE900
ERROR_INVALID_CARD_EXIST	0xEA00
ERROR_COMMAND_UNSUPPORTED	0xEC00
ERROR_COMMAND_PROCESS	0xED00
ERROR_INVALID_COMMAND	0xEE00
ERROR_BAD_COMMAND	0x6A00
ERROR_NO_KEY	0x0400
ERROR_KEY_TYPE	0x0300
ERROR_DUKPT_OVER	0x0500
ERROR_KEY_EXIST	0x0D00
ERROR_SECUREHEAD_RESPONSE_DATA	0xC000
ERROR_SECUREHEAD_STATUS_BUSY	0xC100
ERROR_SECUREHEAD_DAV_ALWAYS_HIGH	0xC200
ERROR_SECUREHEAD_NO_RESPONSE*	0xC300

*Note: If the unit always returns ERROR_SECUREHEAD_NO_RESPONSE, please restart the unit

4.0 Application Note

RS232 Interface:
Default serial port parameters are: baud rate 38400, 8 data bits, 1 stop bit, no parity.

USB HID Interface:

1. **VID =0x0ACD**; **PID =0x2310**.
2. Sign&Pay uses report size 64 bytes for both input report and output report.

3. Report definition:
   1). The first byte is a status byte:
   MSB (bit 7) equals 0 means last or only report, 1 means following report(s) follow.
   Bit 0 --- bit 6 defines the valid bytes in this report.
   2). Other bytes construct data or padding data.

4. Commands and responses are exchanged with Sign&Pay using common Win32 functions like **CreateFile**, **ReadFile**, **WriteFile** and **CloseHandle**. A class written in Visual C++ implementation, which communicates with Sign&Pay is available.
   Below is the process of handling the Sign&Pay device.
   1) Look for the Sign&Pay device using VID (0x0ACD) & PID (0x2310).
   2) Establish connection to Sign&Pay using **CreateFile.**
   3) Communicate with Sign&Pay using **ReadFile** & **WriteFile.**
   4) Destroy connection to Sign&Pay using **CloseHandle**.