Uniform Industrial Corp.®

UIC Bezel5 payment card Reader

Programmer's Manual

RS232 & USB Interface

Document #: PM098
Revision 1.4
May. 7th, 2014

Document History

Document Version	Author	Change	Date
1.1	Robin Tang	Initial version	
1.2	Vicky Tuan		16, Dec, 2013
1.3	Stanley Lui	Adjusted some wordings	18, Dec, 2013
1.4	Ting Sun	Adjusted some wordings(Bezel5)	07.May.2014

Table of Contents

List of Tables

List of Figures

NOTICE

The issuer of this manual has made every effort to provide accurate information contained in this manual. The issuer shall not be held liable for any technical and editorial omissions or errors made herein; nor for incidental consequential damages resulting from the furnishing, performance or use of this material.

 The information provided in this manual is subject to change without notice.

AGENCY APPROVED

- *Specification for FCC Class B*
- *Specification for CE Class B, CISPR 22 Class B*

NOTE: 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.*

You are cautioned that any change or modifications to the equipment not expressly approve by the party responsible for compliance could void your authority to operate such equipment.

WARRANTY

This product is served under one-year warranty of defects in material and functionality to the original purchasers. Within the warranty period, if the product found to be defective will be repaired or replaced. This warranty applies to the products only under the normal use of the original purchasers, and in no circumstances covers incidental or consequential damages through consumers' misuse or modification of the product.

PREFACE

This manual provides detailed information relating to the overall operational, electrical, mechanical, environmental and functional aspects of the Bezel5 reader. This document should be read and understood prior to the initial operation of the product.

For ease of installation and programming use, we have addressed everything from its attractive features to its various configurations.

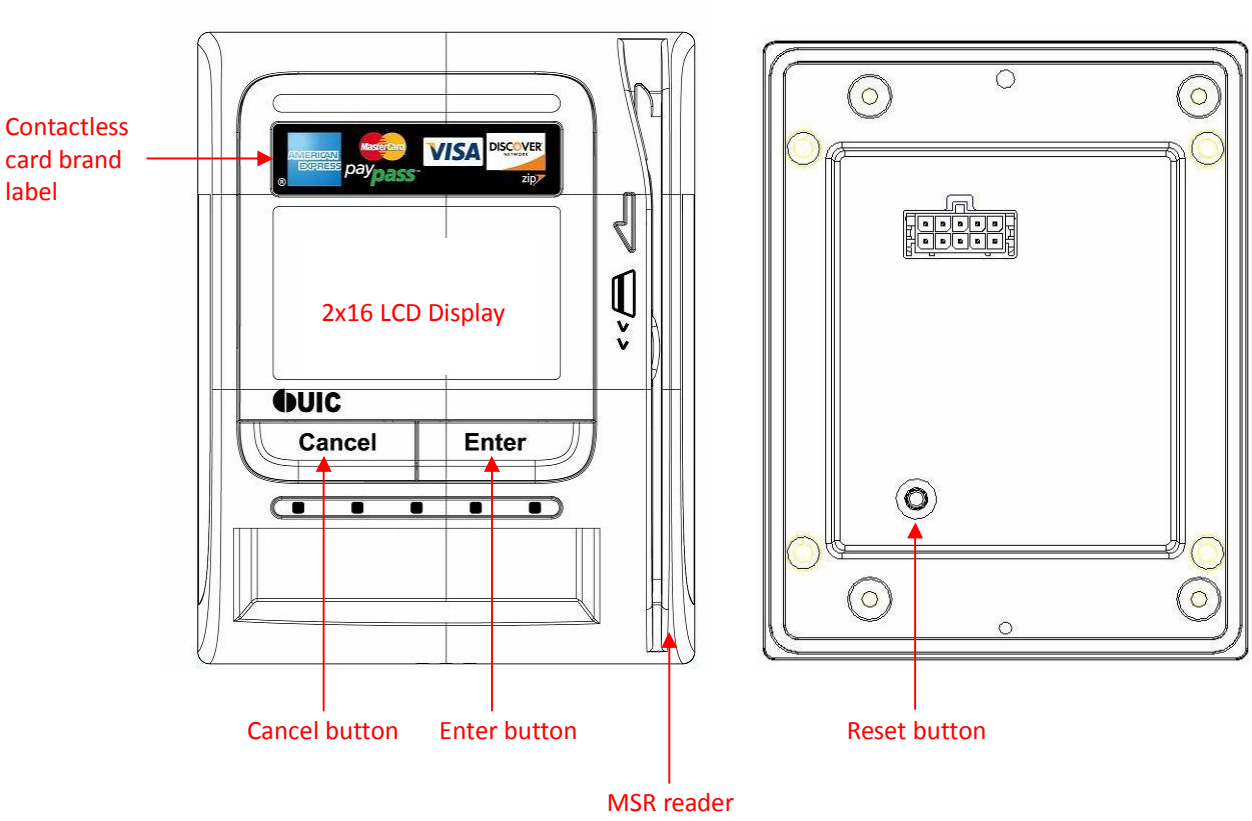
When designing the Bezel5 reader, we selected what we feel are the most useful features and functions. If in some cases you find that your specific needs differ from our existing product, we welcome your comments and suggestions. Custom-designed models are also available.

If further questions do arise, please call for technical support. Our FAE will assist you in any way we can.

1. General Description

This section presents general information about the basic characteristics of the Bezel5.

1.1. Features



Reset Button

The LCD can be refreshed by a short click on the reset button. After the button is released it will generate a short beep to indicate the LCD refresh is complete. For hardware reset, please hold the reset button for 8 seconds. The reader will reset after the button is released.

The *Bezel5* reader provides the following features:

Bezel5	
1	Integrated magnetic stripe reader to read magnetic stripe cards that conform to ISO standard
2	Bi-directional card swipe and triple track read capability
3	64x128 Graphic LCD display with backlight
4	Front: Two Buttons (cancel button and enter button) ; Back: One Button (reset button)
5	LED and Buzzer indicators indicate card status
6	Encrypted card data output (optional)
8	Support RS232, USB 2.0 and serial TTL (optional) interfaces by use of corresponding cables.
9	Supports ISO 14443 & ISO 18092 standard
10	Supports American Express® ExpressPay, MasterCard® PayPassTM (Contactless MagStripe and M/Chip), Visa® PayWave (MSD and qVSDC), and Discover Network Zip Contactless Payments applications, Google Wallet, ISIS Wallet.
11	Reads/Writes NXP MIFARE Plus/Classic/Ultralight/DESFire cards
12	NFC Peer-to-Peer function

Table 1-1 Bezel5 features

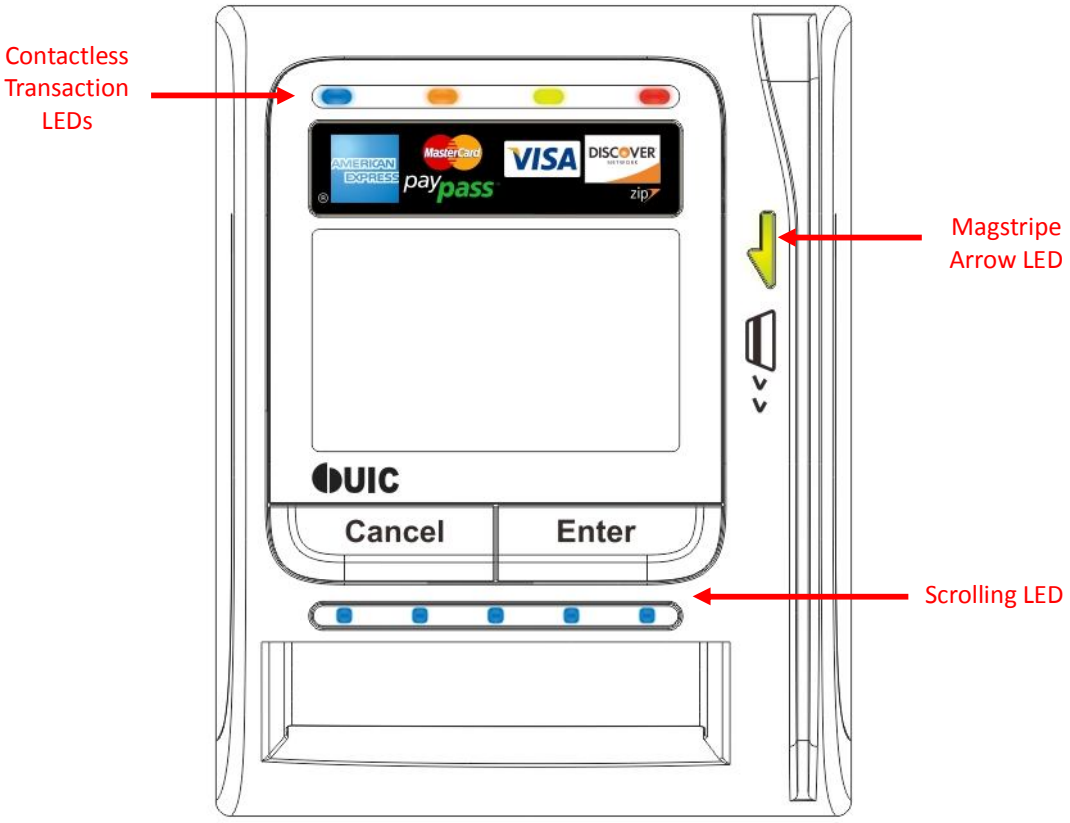
1.2. Application

The contactless smart card payment card reader is mainly used to support the contactless payment operations in the vending machine stations. The reader communicates with a host computer or terminal using a standard RS-232 or USB interface.

1.3. Physical LED Indication

Bezel5 has 3 sets of LED for different indications:

1. Scrolling LED – for catching the attentions of people that the reader is available for accepting payments.
2. Magstripe Ready arrow LED – an indicator to show the Bezel5 is ready to accept Magstripe card swipe
3. Contactless transaction LEDs – leftmost LED indicates Bezel5 is ready to accept contactless card. Rest of LEDs to indicate the transaction status.



There are 2 contactless LED indication modes supported by *Bezel5*:

1. Self-Arm / Host poll Mode: Under this mode, the reader will read and transmit payment card[1]

1 Payment card – the card with MasterCard PayPass, VISA payWave, ExpressPay, or Discover Zip application.

data automatically. The contactless LEDs will be flashing from left to right sequentially.

2. EMV Transaction Mode: Under this mode, the LEDs are lighted during the transaction process. Each LED will represent a different transaction stage in the process.

Idle, ready to accept contactless card

Processing, transaction is processing and do not remove card.

Complete, transaction has been completed by the reader and the card can be removed now.

Card Remove Warning/Bad card read/Transaction Terminated, if card was not remove, red LED will light on to remind cardholder remove the card.

2. Configurations

This section shows the various specifications of the **Bezel5 r**eader.

2.1. Functional Specifications

Basic functions	Read high or low coercivity magnetic stripes (300-4000oe)
	5 LEDs for attention grabbing
	Programmable audio buzzer
	Real time clock /w 5 years battery life
	Contactless communication at 13.56MHz
	4 LEDs for contactless payment indication (optional for 2nd phase development)
Standards	
	ISO 7810/ 7811
	ISO 14443 type A and B compliant
	ISO 18092 compliant
Interfaces	
	RS232 and USB2.0 interfaces by use of corresponding cables.
	USB 2.0 compliant interface configurable to support USB HID MSR, or USB Virtual COM.
	RS232 data output baud rate up to 115.2K BPS
Encrypted card data output (optional)	
	Encrypted card data (AES or Triple DES)
	DUKPT key management with more than 2M keys (model selectable)

Authentication with RSA 2048 bit key

Antennas

Build-in direct matching antenna

Payment applications

American Express ExpressPay

Discover ZIP

MasterCard PayPass/MCHIP

Visa MSD/qVSDC

Google wallet

ISIS wallet

MIFARE applications

Read/Write of MIFARE Plus/Classic/Ultralight/DESFire cards

Support MIFARE higher baud rate up to 424KHz

2.2. Mechanical Specifications

Dimension	Length:	107 mm
	Width:	84 mm
	Depth:	57.5 mm

2.3. Electrical Specifications

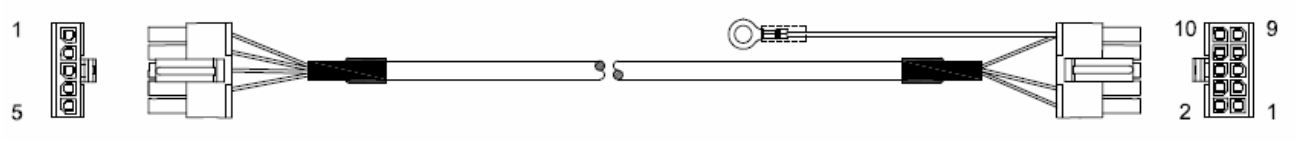
Power Required	7.5 VDC ~ 45 VDC
Power Consumption	75mA in idle mode; 90mA in operating mode (preliminary estimate) at 34 V working voltage
Communication	Standard RS232 signal level Compatible with USB 2.0 specification
Communication Signal (RS232)	Logic 1 = -3 volts to -15 volts Logic 0 = +3 volts to +15 volts

2.4. Environmental Specifications

Temperature	Operating: -20 to 70℃ Storage: -30 to 80℃
Humidity	Operating: 5 to 95% (non condensing)

2.5.Pin Assignment

Interface Pin Assignment



Pin	Signal	Comment
1	VCC	5VDC
2	RXD	
3	TXD	
4	Signal Ground	
5	N/C	

Pin	Signal	Comment
1	VCC	5VDC
2	TXD	
3	RXD	
4	DN	
5	DP	
6	Hi power	7.5VDC ~ 45VDC
7	N/C	
8	N/C	
9	Signal Ground	
10	Shield Ground	

Table 2-1. Pin Assignment of Interface Cable

2.6. Communication

RS232 Interface Data Output

Synchronization
The interface receives and transmits serial asynchronous data at voltage levels compatible with the RS232 specification.

Baud Rate
9600 BPS default (optional: 1200/2400/4800/9600/19200/38400/56000/115.2K BPS)

USB Interface

Compatible with USB specification 2.0
The in/out commands will use the HidD_GetFeature/HidD_SetFeature functions of the Windows standard USB HID driver.

Identification Information

USB Vendor ID: 6352
USB Product ID: BE5A (Virtual COM) / BE5B (HID-MSR)

3. Operation

After power up of the reader, the scrolling LEDs are turned on together with one beep sound, indicating that the reader is ready to operate.

As factory default setting, the *Bezel5* reader is set to Self-Arm mode enabled. Under this mode, the reader will read and transmit payment card[2] data automatically. User needs to disable this mode in order to send contactless card operation commands.

3.1. Reader Default Setting

Item Description	EEPROM Default Value
UART setting	9600-8-N-1
USB Interface	USB HID MSR
Buzzer	Enabled
Protocol format	Protocol 2 (USI2)
Self-Arm mode	Disabled
EMV Mode	Enabled
Administration command protect	Enabled
Data Encryption	Disabled in Protocol 0 / Enabled in Protocol 2
Crypto Algorithm	TDES
DUKPT Key Management Mode	Auto rollover[3]
Contactless smart card manual type (only available in host poll mode)	Type A
Optional functions for the variant versions[3]F	
Google application	Mifare First
Pre-load encryption key	(Customer specific or UIC default)
Pre-load Google Wallet merchant keys	Yes (per merchant request)

Table 3-1. Default Configuration settings

2 Payment cards– the card with MasterCard PayPass, VISA payWave, ExpressPay, or Discover Zip application.
3 Please contact UIC support team for more detailed information.

3.2.Pressing the Buttons and Magnetic Card 'Wiggling'

3.2.1. *Pressing the Cancel Button*

The message "SSS" is transmitted out from the reader after someone presses the Cancel button

3.2.2. *Pressing the Enter Button*

The message "AAA" is transmitted out from the reader after someone presses the Enter Button.

3.2.3. *Magnetic Card is 'Wiggled'*

The message "BBB" is transmitted out from the reader after someone wiggles the magnetic card back and forth.

3.3. Reader Configurations

3.3.1. ***Transmission Protocol***

The user may select from two different protocols: Protocol 0 and 2.
Upon reset, the reader will send out the default power-on character ":", or any character specified by the configuration setting.

Important:
When the Bezel5 reader is working in the USB interface mode, we need to add the header byte C2h and the 2-byte data length before the command.

Protocol 0

In Protocol 0, all characters are transmitted and received using exactly the characters listed in Section 4. There are no headers and Block Check Characters (BCC). Protocol 0 presumes no transmission errors. If the host detects an error, it may request a retransmission.

Example of Protocol 0, RS232 Interface

Host Command	Reader Response	Comment
P		Ready to read
	^	Reader ACK

Example of Protocol 0, USB Interface (Optional)

Host Command	Reader Response	Comment
<C2h><00h><01h>P		Ready to read
	<C2h><00h><01h>^	Reader ACK

Protocol 2

In Protocol 2, all messages are preceded by the ASCII character <SOH>, followed by a one byte reader address, two bytes character count and terminated with a one byte <BCC>.
The <BCC> is an XOR of the characters (8 bits) in the entire message, including <SOH>.

Format: <SOH><ADDRESS><00Hex><COUNT><MESSAGE><BCC>

Example of Protocol 2, RS232 Interface

Host Command	Reader Response	Comment
<01><00><00><01>P<50h>		Ready to read
	<01><00><00><01>^<5Eh>	Reader ACK

Example of Protocol 2, USB Interface

Host Command	Reader Response	Comment
<C2><00><06><01><00><00><01>P<50h>		Ready to read
	<C2><00><06><01><00><00><01>^<5Eh>	Reader ACK

The <ADDRESS> field is for a multi-reader system. This function is not currently supported. The recommended value for this field is NULL (00Hex) but any value will work.

For Protocols 2, if the reader detects an error in an incoming transmission, it will respond with a "Communications Error" message. If the host detects a transmission error, it may request a retransmission.

Protocol 0 is the simplest protocol without adding the redundant data. In order to handle the properly communication, it enforces a 100mSec timeout between characters. In brief, the reader expects the incoming command is ready after 100 ms timeout.

For the applications with the short latency requirement, please choose Protocol 2. The reader processes the incoming command right after received a complete packet.

If the application requests to exchange the binary data, Protocol 2 is recommended.

3.3.2. *Configuration Protocol*

BLP Protocol

In BLP Protocol, all messages are preceded by the ASCII character <HT>, followed by a one byte reader address, one byte character count and terminated with a one byte <BCC>.
<BCC> is an XOR of the 7 data bits, excluding parity, of each character in the entire message, including <HT>.

Format: <HT><00Hex><COUNT><MESSAGE><BCC>

Where HT=09Hex

Example of BLP Protocol, RS232 Interface

Host Command	Reader Response	Comment
<09h><00h><03h>DF<00h><08h>		Load Default
	^	Reader ACK

Example of BLP Protocol, USB Interface

Host Command	Reader Response	Comment
<C2h><00h><07h><09h><00h><03h>DF<00h><08h>		Load Default
	<C2><00><01>^	Reader ACK

3.3.3. *Self – Arm Mode*

The default reader configuration is in "Self-Arm Mode". This allows the payment cards (including VISA MSD, ExpressPay card and the general magnetic stripe credit cards) reading functions to run automatically, reporting the card data to the host without any instruction sent from the host.

With the reader running in the Self-Arm Mode, it can be configured to the "Host Polled Mode" by disabling the Self-Arm Mode. The "Host Polled Mode" allows the card reading functions to be controlled by the relevant host commands.

Card Data Output for Different Types of Card and Reader Configurations

With the reader running in the Self-Arm mode and depending on the configuration set in the reader and the type of card to be read, the reader will output different types of card information. The following table lists out the summary of it:

Type of Card	Reader Configuration Mifare Card Support	
	Disabled (MFxy = 10)[4]	Enabled (MFxy = 11)
Payment Card	Track data	Track data
Mifare Standard 1K	N/A	"M2"
Mifare Standard 4K	N/A	"M3"
Mifare Ultralight	N/A	"M1"
Mifare Ultralight C	N/A	"M1"
Mifare DESFire	N/A	"M4"
Mifare Plus	N/A	"M5"

Card Data Output in Self-Arm and Host-Polled modes

Sending card data under the Self-Arm mode:

Under the Self-Arm mode, the card data output will not include the protocol envelope code. The user can insert the envelope code by utilizing the configuration commands—**SE** and **TO**[5].

[4] Please refer to 4.3.15. *MFxy(4Dh 46h x y) — Set Payment Card and MIFARE Auto-Polling*

[5] Please refer to 4.3.20. *SEx(53h 45h x) — Self-Arm Mode Data Envelope Enable/Disable*
4.3.23. *TOx(54h 4Fh x) — Set Transmitting Data Output Format*

Card data output clear format (Self-Arm mode)

Preamble	Protocol Envelope code	Tk1 prefix	Tk1 Data	Tk1 suffix		
	Separator	Tk2 prefix	Tk2 Data	Tk2 suffix		
	Separator	Tk3 prefix	Tk3 Data	Tk3 suffix	Protocol Envelope code	Postamble

The preamble/postamble is only available in the card data output format under Self-Arm mode. The **Bezel5** reader can be configured to become a secure reader which will output encrypted card data. The data format is as follows:

Encrypted Card data output format[6] (Self-Arm mode)

DUKPT data output format

Encrypt Mode	|	Encrypted Tk1 Data	|	Encrypted Tk2 Data	|	Encrypted Tk3 Data	|	DUKPT KSN	|	Encrypted Session ID	|

RSA data output format

Encrypt Mode	|	Encrypted Tk1 Data	|	Encrypted Tk2 Data	|	Encrypted Tk3 Data	|

Notes Encrypt Mode – 1: DUKPT TDES Mode
2: DUKPT AES Mode
3: RSA Mode

[6] Please refer to section 6 Authentication and Card Data Encryption for more information.

3.3.4. *Host Poll Mode*

Under this mode, user can send out commands manually. Examples like the Q, R, S commands for individual track card data; the commands for controlling the LED and commands for turn on/off antenna power. Host Poll mode is disabled if the reader is configured with default setting.

Read card data using commands in the Host-Polled mode

The reader replies to the so called “Host-Polled” command such as “Transmit Track Data”. The requested message is encapsulated in the protocol envelope.

The response of the Transmit Track Data command is listed as below:

Read TK1 data for command

Protocol Envelope code	Tk1 prefix	Tk1* Data	Tk1 suffix	Protocol Envelope code

Read TK2 data for command

Protocol Envelope code	Tk2 prefix	Tk2* Data	Tk2 suffix	Protocol Envelope code

Read TK3 data for command

Protocol Envelope code	Tk3 prefix	Tk3* Data	Tk3 suffix	Protocol Envelope code

The Protocol Envelope code can be <HEADERS>, <BCC> or NONE, it depends on which protocol is being used.

3.3.5. *EMV Mode*

The EMV transaction is supported by two command groups in ***Bezel5***:

1. Configuration command group
2. General command group.

Usually before the deployment, the configuration commands are set to the bezel with the specific EMV transaction parameters. The settings are stored in the nonvolatile memory and kept until new settings are downloaded. This data is acquirer/issuer related. In other words, once the EMV data has been set, it won't change frequently unless the acquirer/issuer would like to revoke the application or publish new data.

The various general commands are called during the process of the EMV transaction. Each transaction will require several commands. When the transaction is complete, the bezel will return the transaction data.

Note: The configuration commands and the general commands are using different protocol formats. Detailed information can be found in the command section. The bezel can accept both formats at the same time.

The Application Diagram

The below diagram describes how the EMV commands work with the bezel.

EMV Parameters Initialization – They use the bezel configuration commands (BPL protocol). The process is done before deployment but could be updated after installation if the Gateway/Acquirer provides that facility.

In order to process EMV transactions, the bezel must be initialized properly according to the transaction types it has to support. Known as EMV application configuration, the controller needs to configure the bezel with the necessary application related data. All of the application data is stored in the nonvolatile memory of the bezel and is set once before the bezel is deployed to the field site. However, new configuration data can be updated via the remote downloading process if a new application is required to be supported by the bezel.

There are three different groups of reader configurations:

1. Terminal Configuration
2. Application Configuration
3. Public Key

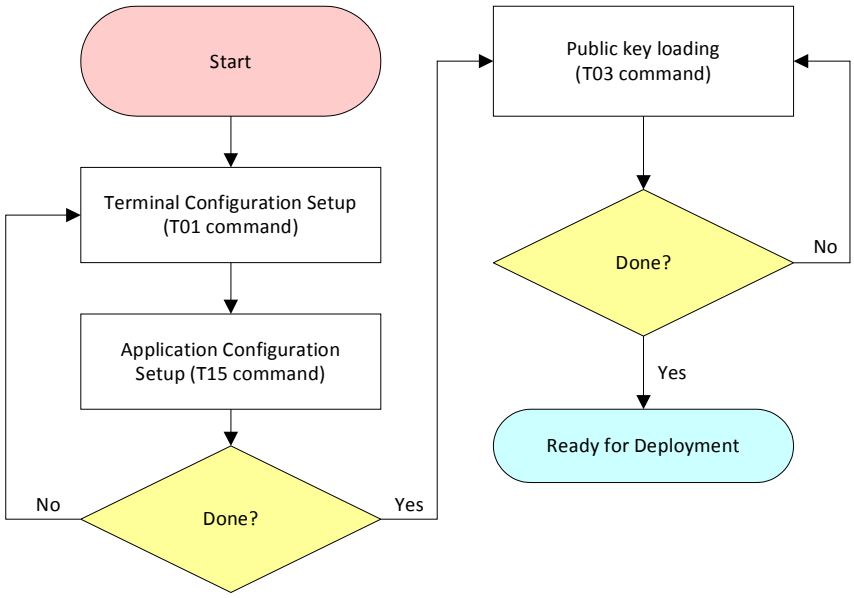


Figure 3-1. EMV Configuration command diagram

EMV Transaction - Using the bezel general commands (Protocol 2). There are many transaction scenarios for EMV transactions. The on-line transaction is shown in the above as one example.

EMV Parameters Maintenance - After the bezel has been deployed, the acquirer may need to update the EMV parameters such as the public key. The parameter update process is similar to the EMV initialization. The controller must be able to handle this kind of process to accept the data from the acquirer. Then converts it to the suitable data format and send to the bezel.

3.3.6. *Details of the Payment Card Tracks Data*

The **Bezel5** reader running at Self-Arm mode will automatically decode the payment card data according to the payment application type. For non-supported payment cards, it is possible to go through the host-polled mode to query the card data.

In general, for the supported payment cards, track 1 and 2 card data will always be present for a successful reading.

Card Data Output Between a MSR Card and a RFID Card

The **Bezel5** reader is able to read two types of payment cards: MSR and RFID. The card data output for each type of card will have a 5-byte string attached before the card data. Their 5-byte strings are pre-defined in the [**Preamble field**] as below:

1) MSR card: '**CARD-**'
2) RFID card: '**RFID-**'

Track 3 Data Format for Magstripe Card

If the magstripe card is swiped and the track 3 is read, the Bezel5 reader will output Track 3 data conform with the ISO 7811 format.

Track 3 Data Format for Contactless Payment

Some contactless payment transactions may require extra information outside track 1 & track 2. The **Bezel5** has introduced a way to reduce the communication time between the host and the reader with the particular information stored in track 3 or in a special Tag. The data is depended on the card type and its application. It is described in the following sections.

The track 3 data of the contactless card is additional card information required for the specific payment transactions. Currently these track 3 data is available for the PayPass-MChip card and the Visa qVSDC/MSD card, and not for other contactless payment cards. These track 3 data is the necessary additional information to be used for System Integration.

To simplify the host application process, this data contains only the value field of the Tag Length Value (TLV) data objects and is expressed in Hex format. The data objects placement is arranged in fixed sequence and are separated by the field separator '='. The transaction data object field is empty if the data object is absent in the card. Moreover, the track 3 data begins with start sentinel and ends with end sentinel.

Track 3 Data Format

Start sentinel	Card Type	Transaction Result	Transaction Data Object(s) (card type dependent)	End sentinel
1-byte	1-byte	2-byte	Each object is separated by the field separator.(n Bytes)	1-byte

Table 3-2. Track 3 Data Format

Table of Various Tags with Tag Length Value and their Descriptions

Tag	Description	Card Type*	Data Object Format in Payment Specification Type, Data Length (byte)	Track 3 (ASCII-HEX) , RS232/Vcom Interface Data Length (byte)
+	Start Sentinel			
x	Card Type			
xx	Transaction Result			
=	Field Separator			
50	Application Label	MasterCard	ans, up to 16 bytes	Up to 16 bytes
57	Track 2 Equivalent Data	V/M	Binary, 1~19 var.	2~38 bytes
5A	PAN	V/M	cn, 0~19 var, up to 10 byte.	0~20 bytes
5F20	Cardholder Name	VISA	ans 2~26, 2~26 bytes	2~26 bytes
5F24	Expiry Date	V/M	n 6 (YYMMDD), 3 bytes	6 bytes
5F2A	Transaction Currency Code	V/M	Binary, 2 bytes	4 bytes
5F34	Application PAN Sequence Number	V/M	n 2, 1 byte	2 bytes
82	Application Interchange Profile	V/M	Binary, 2 bytes	4 bytes
84	Dedicated File Name	MasterCard	Binary, 5~16 var.	10~32 var
95	Terminal Verification Results	V/M	Binary, 5 bytes	10 bytes
9A	Transaction Date	V/M	n 6 (YYMMDD), 3 bytes	6 bytes
9B	Transaction Status Information	V/M	Binary, 2 bytes	4 bytes
9C	Transaction Type	V/M	n 2, 1 byte	2 bytes
9F02	Amount, Authorized (Numeric)	V/M	n 12, 6 bytes	12 bytes
9F03	Amount, Other (Numeric)	V/M	n 12, 6 bytes	12 bytes
9F09	Terminal Application Version Number	V/M	Binary, 2 bytes	4 bytes
9F10	Issuer Application Data	V/M	Binary, var. up to 32 bytes	var. up to 64 bytes
9F11	Issuer Code Table Index	MasterCard	n 2, 1 byte	4 bytes

Tag	Description	Card Type*	Data Object Format in Payment Specification Type, Data Length (byte)	Track 3 (ASCII-HEX) , RS232/Vcom Interface Data Length (byte)
9F12	Application Preferred Name	MasterCard	ans, up to 16 bytes	Up to 16 bytes
9F16	Merchant ID	V/M	ans, 15 bytes	30 bytes
9F17	Personal Identification Number (PIN) Try Counter	VISA	Binary, 1 byte	2 bytes
9F1A	Terminal Country Code	V/M	Binary, 2 bytes	4 bytes
9F1E	Interface Device Serial Number (IFD)	V/M	an, 8 bytes	16 bytes
9F26	Application Cryptogram	V/M	Binary, 8 bytes	16 bytes
9F27	Cryptogram Information Data	MasterCard	Binary, 1 byte	2 bytes
9F33	Terminal Capabilities	V/M	Binary, 3 bytes	6 bytes
9F34	Cardholder Verification Method Results	MasterCard	Binary, 3 bytes	6 bytes
9F35	Terminal Type	V/M	n 2, 1 byte	2 bytes
9F36	Application Transaction Counter	V/M	Binary, 2 bytes	4 bytes
9F37	Unpredictable Number	V/M	Binary, 4 bytes	8 bytes
9F40	Additional Terminal Capabilities	V/M	Binary, 5 bytes	10 bytes
9F41	Transaction Sequence Counter	MasterCard	n 4~8 var., 2~4 bytes	4~8 bytes
9F51	Application Currency Code	VISA	n 3, 2 bytes	4 bytes
9F53	Transaction Category Code	MasterCard	Binary, 1 byte	2 bytes
9F54	Cumulative Total Transaction Amount Limit	VISA	n 12, 6 bytes	12 bytes
9F5D	Available Offline Spending Amount	VISA	n 12, 6 bytes	12 bytes
9F66	Terminal Transaction Qualifiers	VISA	Binary, 4 bytes	8 bytes
9F68	Card Additional Processes	VISA	Binary, 4 bytes	8 bytes
9F6B	Card CVM Limit	VISA	n 12, 6 bytes	12 bytes
9F6C	Card Transaction Qualifiers	VISA	Binary, 2 bytes	4 bytes
9F6D	VLP Reset Threshold	VISA	n 12, 6 bytes	12 bytes
9F6E	Form Factor Indicator	VISA	Binary, 4 bytes	8 bytes
9F6E	Third Party Data	MasterCard	Binary, 5-32 var.	10~64 bytes
9F78	VLP Single Transaction Limit	VISA	n 12, 6 bytes	12 bytes
9F79	VLP Available Funds	VISA	n 12, 6 bytes	12 bytes

Tag	Description	Card Type*	Data Object Format in Payment Specification Type, Data Length (byte)	Track 3 (ASCII-HEX) , RS232/Vcom Interface Data Length (byte)
9F7C	Customer Exclusive Data	VISA	Binary, 0~32 var.	0~64 bytes
-	POS Entry Mode	VISA	Binary, 1 byte, VISA only, the value of '91' for MSD transactions. The value of '07' for qVSDC transactions	2 bytes
-	Terminal Entry Capability	VISA	"5" (for readers that also support VSDC contact chip) or "8" (for readers that do not also support VSDC contact chip).	1 byte
?	End Sentinel			

Table 3-3. TLV Tag format and descriptions

TLV (Tag Length Value) Description

[Tag] means the Tag of the TLV item. If the TLV is present in the transaction, it will show in Track 3, else the [Tag] will leave it as empty. If Value of TLV is not alphanumeric or numeric, the data will be shown in Hex Format.
Ex: 2AH will show 2A in ASCII code to be visible.

Data objects moved from the card to the reader are encapsulated in TLV encoded data objects.

Data objects that have the numeric (n) format are BCD encoded, right justified with leading hexadecimal zeros. Data objects that have the compressed numeric (cn) format are BCD encoded, left justified and padded with trailing 'F's.

Note that the length indicator in the numeric and compressed numeric format notations (e.g. n 4) specifies the number of digits and not the number of bytes.

Data objects that have the alphanumeric (an) or alphanumeric special (ans) formats are ASCII encoded, left justified and padded with trailing hexadecimal zeros.

Value of Card Type

Card Type: It indicates that the tag may appear in track 3 by reading that particular card. V/M means VISA and MasterCard. If the card brand doesn't appear in the card type field, it doesn't mean that this card will not support such tag.

Value	Card Type*
0	MChip
1	MagStripe V3.3
2	Amex Express Pay/EP3 (Reserve)
3	Visa(qVSDC, MSD)
4	Interac
5	Discover Zip/D-PAS (Reserve)

Table 3-4. Card Type indication in Track 3

Value of Transaction Result

Value	Transaction Result**
00	Offline Approved
01	Offline Declined
02	Online
03	Switch to other interface
97	Anti-Collision
99	Terminate

Table 3-5. Transaction Result indication in Track 3

3.3.7. *Payment Card Data Output Example*

PayPass–Magstripe3.3

Track 3 data format:

+	Card Type (1-byte)	Transaction Result (2-byte)	[DD_{Card} Track1]=[DD_{Card} Track2]= [9F6E]=[84]=[50] =[9F12]=[9F11]	?

Track Data:

%B5413330056003529^CUST IMP MC 352/^1412205990090990000009990990996992999040 0?;5413330056003529=14122059999999469960?+102=9900 9099000000999099099699299904 00=9999999469960==A0000000041010=ID352 v1 1===?

Parsed Track Data:

Track 1	%B5413330056003529^CUST IMP MC 352/^1412205990090990000009990990996992999040 0?
Track 2	;5413330056003529=14122059999999469960?
Track 3	+102=9900909900000099909909969929990400=9999999469960==A0000000041010=ID352 v1 1===?

Parsed Track 3 Data:

Card Type	Result
1	02
Magstripe	Online Request

Position	1	2	3
Tag	DD_{Card} Track1	DD_{Card} Track2	9F6E
Value	9900909900000099909909969929990400	9999999469960	
Description			PayPass Third Party Data

Position	4	5	6	7
Tag	84	50	9F12	9F11
Value	A0000000041010	ID352 v1 1		
Description	DF Name	Application Label	Application Preferred Name	Issuer Code Table Index

PayPass–MChip

Track 3 data format:

+	Card Type (1-byte)	Transaction Result (2-byte)	=[CVM]=[9F26]=[9F10]=[9F37]=[9F36]=[95]=[9B]=[9A]=[9C]=[9F02]=[5F2A] =[82]=[9F1A]=[9F03]=[9F33]=[9F35]=[84]=[9F09]=[9F1E]=[9F16]=[9F41]=[9F27] =[9F34]=[9F53]=[5A]=[5F24]=[57]=[9F6E]=[50]=[9F12]=[9F11]=[5F34]	?

Track Data:

%B5413330089600119^0056^ETEC/PAYPASS^141231^^?;5413330089600119=1412==?+002=1F=96EB58603A581C2F=0110A0000000000000000000000000000FF0000000000000000000000000000=14A946B2=0001=8000000000==120604=00=000000001500=0978=1880=0056=000000000000=000888=22=A0000000041010=0002=1234567890000000=3030303030303030303030303030303031=00000039=80=1F0300=00=5413330089600119=141231=5413330089600119D14122010123409172==505043204D434420313120763220 31===01?

Parsed Track Data:

Track 1	%B5413330089600119^0056^ETEC/PAYPASS^141231^^?
Track 2	;5413330089600119=1412==?
Track 3	+002=1F=96EB58603A581C2F=0110A0000000000000000000000000000FF0000000000000000000000000000=14A946B2=0001=8000000000==120604=00=000000001500=0978=1880=0056=000000000000=000888=22=A0000000041010=0002=1234567890000000=3030303030303030303030303030303031=00000039=80=1F0300=00=5413330089600119=141231=5413330089600119D14122010123409172==505043204D434420313120763220 31===01?

Parsed Track 3 Data:

Card Type	Result
0	02
M-Chip	Online Request

Position	1	2	3	4
Tag	CVM	9F26	9F10	9F37
Value	1F	96EB58603A581C2F	0110A0000000000000000000000000000000FF0000000000000000000000000000	14A946B2
Description	No CVM	Application Cryptogram	Issuer Application Data	Unpredictable Number

Position	5	6	7	8
Tag	9F36	95	9B	9A
Value	0001	8000000000		120604
Description	Application Transaction Counter	Terminal Verification Results	Transaction Status Information	Transaction Date

Position	9	10	11	12
Tag	9C	9F02	5F2A	82
Value	00	000000001500	0978	1880
Description	Transaction Type	Amount, Authorized	Transaction Currency Code	Application Interchange Profile

Position	13	14	15	16
Tag	9F1A	9F03	9F33	9F35
Value	0056	000000000000	000888	22
Description	Terminal Country Code	Amount, Other	Terminal Capabilities	Terminal Type

Position	17	18	19	20
Tag	84	9F09	9F1E	9F16
Value	A0000000041010	0002	1234567890000000	3030303030303030 30303030303031
Description	DF Name	Terminal Application Version Number	Interface Device Serial Number	Merchant ID

Position	21	22	23	24
Tag	9F41	9F27	9F34	9F53
Value	00000039	80	1F0300	00
Description	Transaction Sequence Counter	Cryptogram Information Data	Cardholder Verification Method Results	Transaction Category Code

Position	25	26	27	28
Tag	5A	5F24	57	9F6E
Value	5413330089600119	141231	5413330089600119D14122 010123409172	
Description	PAN	Expiry Date	Track 2 equivalent Data	Paypass Third Party Data

Position	29	30	31	32
Tag	50	9F12	9F11	5F34
Value	505043204D43442 031312076322031			01
Description	Application Label	Application Preferred Name	Issuer Code Table Index	Card Serial Number

Visa (qVSDC, MSD)

Track 3 data format:

+	Card Type (1-byte)	Transaction Result (2-byte)	=[9F26]=[9F10]=[9F37]=[9F36]=[9F66]=[95]=[9B]=[9A]=[9F02]=[5F2A]=[82] =[9F1A]=[9F03]=[9F33]=[9F35]=[9F09]=[9F1E]=[9F16]=[5F34]=[9F40]=[9F6E] =[9F7C]=[57]=[5A]=[5F20]=[5F24]=[9C]=[9F5D]=[9F68]=[9F6C]=[9F6B]=[9F51] =[9F17]=[9F78]=[9F79]=[9F6D]=[9F54]=[POS Entry Mode]=[Terminal Enter Capability]	?

Track Data:

%B4761739001010010^
/^2012120001231003990300000?;4761739001010010=2012120001233990003 1?+300=AABBCCDDEEFF1122=06011103900000=94018C92=0003=A0804000=0000000000==120604=000000000100=0840=2000=0840=000000000000=000888=22=0000=1234567890000000=3030303030303030303030303031=01=6000000001===4761739001010010D2012120001233990003 1F=4761739001010010==201231=00=000000010000==3000========07=08=40?

Parsed Track Data:

Track 1	%B4761739001010010^ /^2012120001231003990300000?
Track 2	;4761739001010010=2012120001233990003 1?
Track 3	+300=AABBCCDDEEFF1122=06011103900000=94018C92=0003=A0804000=0000000000==120604=000000000100=0840=2000=0840=000000000000=000888=22=0000=1234567890000000=3030303030303030303030303031=01=6000000001===4761739001010010D2012120001233990003 1F=4761739001010010==201231=00=000000010000==3000========07=08=40?

Parsed Track 3 Data:

Card Type	Result
3	02
VISA	Online Request

Position	1	2	3	4
Tag	9F26	9F10	9F37	9F36
Value	AABBCCDDEEFF1122	06011103900000	94018C92	0003
Description	Application Cryptogram	Issuer Application Data	Unpredictable Number	Application Transaction Counter

Position	5	6	7	8
Tag	9F66	95	9B	9A
Value	A0804000	0000000000		120604

Position	5	6	7	8
Description	Terminal Transaction Qualifiers	Terminal Verification Results	Transaction Status Information	Transaction Date

Position	9	10	11	12
Tag	9F02	5F2A	82	9F1A
Value	000000000100	0840	2000	0840
Description	Amount, Authorized	Transaction Currency Code	Application Interchange Profile	Terminal Country Code

Position	13	14	15	16
Tag	9F03	9F33	9F35	9F09
Value	000000000000	000888	22	0000
Description	Amount, Other	Terminal Capabilities	Terminal Type	Application Version Number

Position	17	18	19	20
Tag	9F1E	9F16	5F34	9F40
Value	1234567890000000	30303030303030303030 3030303031	01	6000000001
Description	Interface Device Serial Number	Merchant ID	Application PAN Sequence Number	Additional Terminal Capabilities

Position	21	22	23	24
Tag	9F6E	9F7C	57	5A
Value			4761739001010010D201 21200012339900031F	4761739001010010
Description	Form Factor Indicator	Customer Exclusive Data	Track 2 Equivalent Data	PAN

Position	25	26	27	28
Tag	5F20	5F24	9C	9F5D
Value		201231	00	000000010000
Description	Cardholder Name	Expiry Date	Transaction Type	Available Offline Spending Amount

Position	29	30	31	32
Tag	9F68	9F6C	9F6B	9F51
Value		3000		
Description	Card Additional Processes	Card Transaction Qualifiers	Card CVM Limit	Application Currency Code

Position	33	34	35	36
Tag	9F17	9F78	9F79	9F6D
Value				
Description	PIN Try Counter	VLP Single Transaction Limit	VLP Available Funds	VLP Reset Threshold

Position	37	38	39
Tag	9F54	POS Entry Mode	Terminal Enter Capability
Value		07	08
Description	Cumulative Total Transaction Amount Limit	qVSDC transaction	Always set to 8

4. Commands and Responses

4.1. Common Command Description

Reader Response Code

Response	Meaning
^	Acknowledgement
*	Cannot execute (e.g. out of range)
!	Bad parameter (e.g. incorrect length)
+ (2BH)	No Magnetic Stripe Card Data. Command was received correctly.
? (3FH)	Communication Error. Command was not received correctly.
: (3AH)	Power On report.
~ (7EH)	Unavailable. Hardware is not available to complete this request.

4.1.1. % (25H) - Retransmit

Retransmits the last message sent by the reader.

Example

Host Command	Reader Response Example
%	
	^

Note: This command is ignored if the reader is running in Self-Arm mode.

4.1.2. 70 (37H30H) or 90(39H30H) - Serial Number Report

Gets the reader's serial number that has been stored in the EEPROM

Example

Host Command	Reader Response Example
70	
	00000000

4.1.3. 71 (37H31H) or 91 (39H31H) - Copyright Report

Transmits the version and copyright information

Example

Host Command	Reader Response Example
71	
	131210,BE50131A:V1.G

This command is sent if the user wants to know the version, model and copyright of the currently loaded **Bezel5** firmware. The response is an ASCII string giving the firmware date (yymmdd), reader type and the firmware version number, followed by the firmware copyright statement. The firmware copyright statement is absent in the OEM version.

4.1.4. 7A (37H41H) or 9A (39H41H) - Module Version Report

Transmits the version information

This command is sent to request the version of the module in the **Bezel5 f**irmware currently loaded. The response is a 6-byte ASCII string, reader type and the module version number.

Command Packet

Byte 0-1	Byte 2
7A (37H41H) or 9A (39H41H)	0-7 (See Table 4-1)

Description table

Module	In byte	Example ("-xxxx" is a 4 bytes checksum)
HAL_VERSION	0	BE5H11-xxxx
PAYPASS_VERSION	1	BE5P11-xxxx
AMEX_VERSION	2	BE5A11-xxxx
VISA_VERSION	3	BE5V11-xxxx
DN_VERSION	4	BE5D11-xxxx
L1_VERSION	5	BE5111-xxxx
L2_VERSION	6	BE5211-xxxx
MIFARE_VERSION	7	BE5M11-xxxx

Table 4-1. Module Version Report Description

Example

Host Command	Reader Response Example
7A0	
	BE5H11-xxxx

4.1.5. 7F (37H 46H) – Get Hardware Status

This command can get the hardware status back after powered-on. The host can use this command to check if there's any hardware initialization issue happened during the power-on process. If all the ICs are working well, the response should be 4 bytes of zeros.

Response

Byte 0	Meaning
Bit 8	Reserved
Bit7	Reserved
Bit6	Reserved
Bit5	Reserved
Bit4	Reserved
Bit3	Reserved
Bit2	Reserved
Bit1	Reserved

Byte 1	Meaning
Bit 8	Reserved
Bit7	Reserved
Bit6	Reserved
Bit5	Reserved
Bit4	Reserved
Bit3	Reserved
Bit2	Reserved
Bit1	Reserved

Byte 2	Meaning
Bit 8	Create New Session Key
Bit7	Session Key Error
Bit6	RSA Key Error
Bit5	Interface IC Error

Byte 2	Meaning
Bit4	Create New DUKPT Key
Bit3	DUKPT KEY Error
Bit2	Create New MAC Key
Bit1	MAC Key Error

Byte 3	Meaning
Bit 8	Create New PingPing Key
Bit7	PingPing Key Error
Bit6	Ext Flash Error
Bit5	Create New RSA Key
Bit4	RSA Chip Error
Bit3	Create New Master Key
Bit2	Initial Master Key Error
Bit1	EEPROM Error

4.1.6. ? (3FH) - Select Verbose Responses Command

Most error responses, until the reader receives a reset command, error response will include a short descriptive message.

Example

Host Command	Reader Response Example
?	
	^Verbose responses enabled

4.1.7. $ (24H) – Reader Status Request

Interrogate the reader about its operational status. Two bytes of status information will be returned.

Example

Host Command	Reader Response Example
$	
	`<01>

Reader Response Example = '<01>

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1, see Table 4-2	0	1	1	0	0	0	0	0
Byte 2, see Table 4-3	0	0	0	0	0	0	0	1

First Status Byte

Bit	Value: 0	Value: 1
0	RFU	RFU
1	No Card Present	Card Seated
2	RFU	RFU
3	RFU	RFU
4	No Card status Report	Auto Card status Report
5	always '1'	always '1'
6	Not armed to read	Armed to read
7	RFU	RFU

Table 4-2. First Byte Description of Reader Status Request

Second Status Byte

Bit	Value: 0	Value: 1
0	First LED OFF	First LED ON
1	LED not Flash	LED Flash
2-3	RFU	RFU
4	No RFID Read	RFID Read
5-7	RFU	RFU
Exception: If there is any LED flashes, the bit 0 status will be ignored. (i.e., in this case, bit 0 always = '0')		

Table 4-3. Second Byte Description of Reader Status Request

4.1.8. # (23H) – Configuration Request

Returns single byte or extended 16-byte string representing the configuration of the device.

Example

Host Command	Reader Response Example
#	
	?

Reader Response Example = " ? "

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1, see Table 4-4	0	0	1	1	1	1	1	1

Standard, One Configuration Byte

Bit	Value: 0	Value: 1
0	Track 1 not present	Track 1 present
1	Track 2 not present	Track 2 present
2	Track 3 not present	Track 3 present
3-7	RFU	RFU

Table 4-4. First byte of Configuration Request response

Extended Configuration Bytes (16 bytes)

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-15
Remark	Equip.0	Equip.1	Protocol	Speed	Address	RFU

Equip. 0 — Extended Configuration Byte 0

Bit	Value: 0	Value: 1
0-3	RFU	RFU
4	Track 1 not present	Track 1 present
5	Track 2 not present	Track 2 present
6	Track 3 not present	Track 3 present
7	RFU	RFU

Equip. 1 — Extended Configuration Byte 1

Bit	Value: 0	Value: 1
0-7	Not Used	Not Used

Byte 2 – Byte 4

Byte	Remark	
2	Protocol	00H = USI2; 03H = USI0; 06H = USI1

Byte	Remark	
3	Speed	00H=1200, 01H=2400, 02H=4800, 03H=9600, (Default) 04H=19.2k, 05H=38.4k, 06H=57.6k, 07H=115.2k bps
4	Address	Always 00H.

By using the configuration setting command, users can select the standard or extended format. For the Extend command usage refer to 4.3.8. ECx(45h 43h x) — Extended Configuration Report Enable/Disable.

4.1.9. <CAN> (18H) – Clear Data Buffer

Clears read data buffers.

Example

Host Command	Reader Response Example
<18>	
	^

4.1.10. <7FH> – Warm Reset

It aborts all current actions and causes the device to execute all initialization functions. The device will respond as if in a "power up" cycle; by default it returns a ':' (3AH). This operation will take at least 3 seconds to complete.

Example

Host Command	Reader Response Example
<7F>	
	^

4.1.11. 5 (35H) – Set RTC Time

This command is used to set and read device's RTC Time

Command Packet

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
5	CMD	Date or Time				

CMD Description

[CMD, 1 byte] (ASCII – Hex value)	Description
1 (or 31h)	Read Date
2 (or 32h)	Read Time
3 (or 33h)	RFU
4 (or 34h)	Set Date
5 (or 35h)	Set Time

51 (35H31H) - Read Date

Response data Packet:

Byte 0 – Byte 1	Byte 2	Byte 3	Byte 4
Year	Month	Date	Week
<20*><12>	<12>	<06>	<04>
*The year <20> can be interpreted as space character.			01h=Monday, 02h=Tuesday, … 07h=Sunday

Note: BCD format from 010 (0000BCD = 0h) to 910 (1001BCD = 9h)

Example

Host Command	Reader Response Example
51	
	<20*><12><12><06><04>

52 (35H32H) - Read Time

Response data Packet:

Byte 0 – Byte 1	Byte 2	Byte 3	Byte 4
Hour	Min	Second	Millisecond
<16>	<30>	<00>	<04><90>

Note: BCD format from 010 (0000BCD = 0h) to 910 (1001BCD = 9h)

Example

Host Command	Reader Response Example
52	
	<16>0<00><04><90>

54 (35H34H) - Set Date

Command Packet:

Byte 0-1	Byte 2-3	Byte 4	Byte 5	Byte 6
Command	Year	Month	Date	Week
54	<14><0C>	<0C>	<06>	<04>
Hex value format valid input				01h=Monday, 02h=Tuesday, … 07h=Sunday
Default setting is <20><01><01><01><01>, obtained by Read Date. YYYY: 14h 00h – 1Eh FFh (2000 – 3000) If the 'YYYY' falls out of range, the reader will restore the configuration to default settings after resetting the device.				

Example

Host Command	Reader Response Example
54<14><0C><0C><06><04>	
	^

55 (35H35H) - Set Time

Command Packet: (Hex value format valid input)

Byte 0-1	Byte 2	Byte 3	Byte 4	Byte 5-6
Command	Hour	Min	Second	Millisecond
55	<11>	<0E>	<37>	<00><0A>

Example

Host Command	Reader Response Example
55<11><0E><37><00><0A>	
	^

4.1.12. B (42H) – Buzzer Beep control

Used to let the Buzzer to beep under user's control

Command Packet

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Command	Count	Tone	On Duration	Short Duration
B	<31>	<30>	<7F>	<00>

Example

Host Command	Reader Response Example
B<31><30><7F><00>	
	^

Command Type

Field	Description
Count	0 (30h, ASCII Hex) – long beep **Important: Once 'B0' command starts beeping, NO command can STOP it–unless users send a "Reset" command to stop it.
	1~9, A~F(31h~39h 41h~46h, ASCII Hex) – 1~15 short beeps
Tone	For adjusting the frequency level, 00h~FFh(high → low).
On Duration	The duration of a beep; time unit is 10ms, 00h means 10ms, FFh means 2560ms.
Short Duration	The interval between 2 beeps in unit of 10 milliseconds; 00h means 10ms, FFh means 2560ms.

Note: If Type parameter is omitted, reader will treat it as the ONE SHORT Beep command.

4.1.13. I (49H) – Load RSA Key

This command is used to load RSA Key and query Key Index for PayPass MCHIP and VISA qVSDC applications.

Command Packet

Byte 0	Byte 1 or Byte 1-3	Byte 2~
Command	Type, see Table 4-5	Data, see Table 4-6
I	1	<01><00><A2><05><00><00><00><00><01><01><03><03><01><00><01><80><14><C3><12><D4><88><A7><09><88><A4><F2><19><D5><D6>~y<8F><DC><A0><A7><0D><90>fc<13>:p<98><1E>a&<F9>+(<8B><ED><98><D6><97><82><CC><A8><C5><94><B0><CF>*<B2><EC><E7>9<98><08>WF<88><A1><B8>K<BC><D2><0D>7<E9>-<1C>h<9A>[<BD><84>Z<99><88>Q<0C><9A><96><EE>D]L<1D><A3>W<AD>=<14>-^<8B><C5><D6>DT<92><12>1~z5R'<8B><F8><C6>{_<BF>e<0F><FD><AF>W~<F2>}{3o<EF>k<A6>Sj<DE>;<A1><09><14><DD>>+l<CD>8<CF>Y<99><88>y<F0>X<BF><86><C8>'<E0><9E><91>

Command Type

Command Format (ASCII – Hex)	Description
0[01H 16H] (or 30H 01H 16H)	Show Stored Key's Index and RID
1 (or 31h)	Load RSA Key
2 (or 32h)	Load Authentication RSA Key
5 (or 35h)	Load Test Key and Test RSA Chip
FFH 00H 00H	Erase all Key Entry

Table 4-5. Load RSA Key Type

Command Data

Data	Description
Entry Index	1 byte in binary format, must be 1-16.
Total Len	Total length of rest data, 2 bytes in binary format.
RID Len	1 byte in binary format, must be 5.
RID	5 bytes in binary format
CAPKI Len	1 byte in binary format, must be 1.
CAPKI	Key Index, 1 byte in binary format.
Exp Len	1 byte in binary format, must be 1 or 3.
Exponent	1 or 3 bytes in binary format.

Data	Description
Mod Len	1 byte in binary format, Max is 248.
Modulus	1-248 bytes in binary format.
Sha_1 Len	Len — 1 byte in binary format, must be 0 or 20.
Sha_1	20 bytes in binary format, if present.

Table 4-6. Load RSA Key Data Description

I1 command Example

Data	Value
RID	0000000001
CA Index	03
Modulus	14C312D488A70988A4F219D5D67E798FDCA0A70D906663133A70981E6126F92B288BED98D69782CCA8C594B0CF2AB2ECE7399808574688A1B84BBCD20D37E92D1C689A5BBD845A9988510C9A96EE445D4C1DA357AD3D142D5E8BC5D644549212317E7A3552278BF8C67B5FBF650FFDAF577EF2297B336FEF6BA6536ADE3BA109
Exponent	010001
Sha_1 Value	DD3E2B6CCD38CF59998879F058BF86C827E09E91
Command Form: I1<01><00><A2><05><00><00><00><00><01><01><03><03><01><00><01><80><14><C3><12><D4><88><A7><09><88><A4><F2><19><D5><D6>~y<8F><DC><A0><A7><0D><90>fc<13>:p<98><1E>a&<F9>+(<8B><ED><98><D6><97><82><CC><A8><C5><94><B0><CF>*<B2><EC><E7>9<98><08>WF<88><A1><B8>K<BC><D2><0D>7<E9>-<1C>h<9A>[<BD><84>Z<99><88>Q<0C><9A><96><EE>D]L<1D><A3>W<AD>=<14>-^<8B><C5><D6>DT<92><12>1~z5R'<8B><F8><C6>{_<BF>e<0F><FD><AF>W~<F2>}{3o<EF>k<A6>Sj<DE>;<A1><09><14><DD>>+l<CD>8<CF>Y<99><88>y<F0>X<BF><86><C8>'<E0><9E><91>	

Note: These values are used for testing purposes.

Table 4-7. Load RSA Key example (I1 command)

Command Data Format - (Command I2, Authentication RSA Key)

Data Byte	Field Name	Length	Notes
n	Padding Data	Var.	Padding frame see Table 4-9
16+n	Random	16 Bytes	Issue 90h 10h command to get random.
17+n	Exponent Length	1 Byte	
21+n	Exponent	4 Bytes	
23+n	Modules Length	2 Bytes	
23+n+m	Modules	Var.	Binary format
43+n+m	SHA1	20 Bytes	Padding + Random + Exp Len + Exp + Modules Length + Modules

Table 4-8. Authentication RSA Key data format (I2 command)

Padding Frame - (Command I2, Authentication RSA Key)

Byte 0~1	Byte 2+n	Byte 3+n
00h 01h	Var.	00h

Table 4-9. Padding Frame of Authentication RSA Key command

Example - (Command I2, Authentication RSA Key)

Host Command	Reader Response
I26F5DFC046F37D2CEFAE240A3E1870CA374F34FF9F1D138D5D78B09AA1863E1129F35E2559 4B40205E46EE1C603AD080141B51020892408DB741A58B203A4E8D75A9E98B45FA33AE495F 24F2D6F78048804320216E295E721DED633EFCCCD8CA91B7D12E8AB7FBA8490B5AC87F17E9 3A2E18C4993B52E020ED3C18138CE4A091D2EB0DA846D50C5432E186AB148257884C409A4D DBBF42FB8CFBF778E7966E3704DC8B976945B302D21E82515390FCC6F6BCA4894F6CA29B027 40DD22A1B530DA2CD2F90E9F673E3E0BD1EBCAA3BAC2D5F664F5F77C5193B4A78AF8D5CAC F5344D5E63CF3898D77F96468FA7CBABEE6A4E43E203AF6141E19D3390B9C5565C88F13E091 5EB57034EB4C3788DE6FFEE355364EDDF4E32CABB52DA0DDB816634E58BFA79FFC890B8DE0 F766906C05EBAA2578F85D2D3D3F4D5712722441D1449E40F6BC7205DD281C937E675214D6 63BA69BDB2E5674B4CC8D4D1002814E7BBEED9A96C177A0C8872F59E12593607A440E3ABA D5DF510798B363505F6E81E63FC3F60884404923768C6D1228CAE34289C051418C2FC8C98F5 8CB98F1DED473A7F8F8449682B572EB56758588DD9D6DFD5BC4EC72FF9D3B7E9C6B79F7231 6B593611FF5D0753466621A80EB71BB2D4575AB795C47A1FD4D21B3D702FB296E67F49ED18 07531177900AFE9D7FE8B4DBF19AA520E9BC6D8AA0EE664887C3CD716B5E	
	^

Table 4-10. Load Authentication RSA Key example (I2 command)

4.1.14. w (77H) – Exception File

To add or process the PAN in the Exception File. Primary Account Numbers kept by this black list will be denied for transactions.

Command Packet

Byte 0	Byte 1	Byte 2~
Command	Type	Data
w	2	<10>6011111111111117

Command Type

ASCII – Hex Value	Description
0 (or 30h)	Erase Exception File
1 (or 31h)	Report counts of PANs in the Exception File
2 (or 32h)	Add a PAN to the file, 272 entries max.
3 (or 33h)	Query if a PAN exists in the Exception File

ASCII – Hex Value	Description
4 (or 34h)	Request a certain PAN from the Exception File

Command Data

Type	Description
2 (or 32h)	data length(1 byte) + PAN(up to 19 bytes ASCII '0'~'9')
3 (or 33h)	
4 (or 34h)	2 bytes long, range from 0000h to 010Fh

Response data format

Type	Description
1 (or 31h)	Return 2-byte binary number -- the total number of PANs in the file.
3 (or 33h)	Return '1' if PAN exists; else, return '0'.
4 (or 34h)	Return primary account number; else, return 00h.

w1 Example

Host Command	Reader Response Example
w1	
	<00><02>
w2<10>6011111111111117	
	^
w3<10>6011111111111117	
	1
w4<00>03>	
	<10>6011111111111117

4.1.15. @ (40H) – Display Control

The LCD panel can show eight rows x 18 small ASCII font, four rows x 18 big ASCII font or four rows x 9 Chinese code.

Command Packet

Byte 0	Byte 1	Byte 2 ~
Command	Type	Data
@	<01>	<00>

Command Type

Type	Description
01h	LCD Clear, See Table 4-12
02h	LCD Write Char, See Table 4-13
03h	Graphic picture selection, See Table 4-14
04h	LCD Inverse, See Table 4-15
07h	Cursor Blink, See Table 4-16
08h	Cursor Home
09h	Cursor Display, See Table 4-17
0Ah	Position Cursor, See Table 4-18
0Bh	LCD Blinking, See Table 4-19
0Ch	LCD Blink Time, See Table 4-20
16h	LCD Backlight control, See Table 4-21

Table 4-11. LCD Function Table

Command Data Option

LCD Clear

Data	Description
00h	Clear entire display
01h	Clear line 1
02h	Clear line 2
03h	Clear line 3
04h	Clear line 4

Table 4-12. Clear LCD command option

Example

Display Line	Host Command	Reader Response Example
1	<01><00><00><03>@<01><01>B	
		^
2	<01><00><00><03>@<01><02>A	
		^
3	<01><00><00><03>@<01><03>@	
		^
4	<01><00><00><03>@<01><04>G	

Display Line	Host Command	Reader Response Example
		^

LCD Write Char (5 Fields in binary format)

Field 1	Field 2	Field 3	Field 4	Field 5
1~18	1~64	Length 1	Length 2	Data buffer

Total length = (Length 1 * 256) + Length 2

Table 4-13. Write Characters to LCD

Example

Display Line	Host Command	Reader Response Example
1	<01><00><00><16>@<02><00><00><00><10>1234567890123456C	
		^
2	<01><00><00><16>@<02><00><01><00><10>1234567890123456B	
		^
3	<01><00><00><16>@<02><00><02><00><10>1234567890123456A	
		^
4	<01><00><00><16>@<02><00><03><00><10>1234567890123456@	
		^

Graphic Picture Selection

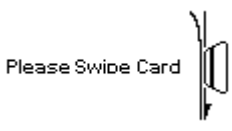
Data	Description
00h	
01h	
02h	
03h	

Table 4-14. Graphic Picture Selection

LCD Inverse

Data	Description
00h	Normal
01h	Light pixels on a dark background

Table 4-15. LCD Inverse Option

Example

	Host Command	Reader Response Example
Enable	<01><00><00><03>@<04><01>G	
		^
Disable	<01><00><00><03>@<04><00>F	
		^

Cursor Blink

Data	Description
00h	No blinking
01h	Cursor blink on

Table 4-16. Cursor Blink Option

Cursor Display

Data	Description
00h	Cursor hides
01h	Cursor display

Table 4-17. Cursor Display Option

Position Cursor

Data	Description
ux and uy	Set cursor position, 2 bytes in binary format

Table 4-18. Cursor Position Set

LCD Blinking

Data	Description
00h	Blinking off
01h	Blinking on

Table 4-19. LCD Blinking Option

LCD Blink Time

Data	Description
00h~0Fh	Setup blinking time, 1 byte in binary format

Table 4-20. Set LCD Blinking Time

LCD Backlight control

Data	Description
00h	Off
01h	Turn on the LCD backlight.

Table 4-21. LCD Backlight Control

Example

	Host Command	Reader Response Example
On	<01><00><00><03>@<16><01>U	
		^
Off	<01><00><00><03>@<16><00>T	
		^

4.1.16. L (4Ch) / l (6Ch) / ((28h)- LED Control

This command is for Contactless LED control. Usually contactless LED don't have to control by the host, but this command can be used for LED function testing or specific purpose.

Command Packet

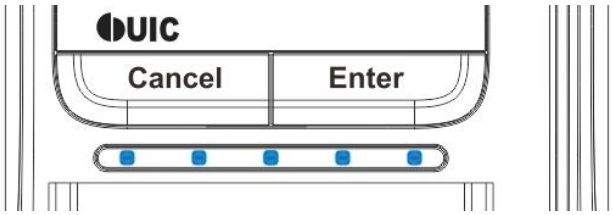
Byte 0	Byte 1 (LED Position)	Description
L	x	LED turn-on
l	x	LED turn-off
(	x	LED Flashing

LED Position

Type	Description
1, (31h)	First LED (Blue)
2, (32h)	Second LED (Orange)
3, (33h)	Third LED (Yellow Green)
4, (34h)	Fourth LED (Red)

4.1.17. LE (4Ch 45h) / LD (4Ch 44h) - Flash LED Control

This command is for the user to recognize that the reader is up and functioning. The default is turned-on upon power up. For specific purpose, it can be disabled and controlled by the host to determine when the LED be turned on.



Command Description

Command	Description
LE	Turn-on Flashing LED
LD	Turn-off Flashing LED

4.2. General Commands Description

The default setting of the ***Bezel5*** reader, Self-Arm mode, is mainly used to simplify the process so that the host does not need to communicate back and forth with the reader. In this situation, the *Bezel5* acts like a general magnetic stripe card reader. Whenever it senses the card it will try to decode the card data automatically and send out the decoded data to the host if the process is successful. Otherwise, error code will be sent out for host to make next activation.

If the application would like to take complete control on the reader, we recommend the user to use the "Host-Polled" mode instead of the "Self-Arm" mode. It can be done by either sending "Self-Arm" disable command or changing the default setting in the reader configuration.

Once the *Bezel5 re*ceives the Self-Arm disable command, 'H0' (see the command description section), it will turn off the auto-read function and then wait for the "Arm-to-Read" command, 'P' (50h) prepared for the next transaction. Since the Self-Arm disable command won't change the EEPROM setting, the *Bezel5* will turn back to the Self-Arm mode in the next power cycling. Besides, the Self-Arm enable command, 'H1', can also bring the *Bezel5* back to the Self-Arm mode.

To disable the Self-Arm mode permanently, the host needs to set its EEPROM value of the *Bezel5*. The configuration command 'SA' (see 4.3.19. *SAx(53h 41h x)* — *Self-Arm Mode Enable/Disable*) saves the setting into the EEPROM of the *Bezel5* and keeps the value until the next change.

We recommend users to use Protocol 2 (USI2) in their "host-polled" applications. This protocol contains the header, message counter and block check character. It is better than using Protocol 0(USI0) as it can prevent the data to be misinterpreted but requires more redundant bytes.

Self-Arm Mode transaction process Example flow

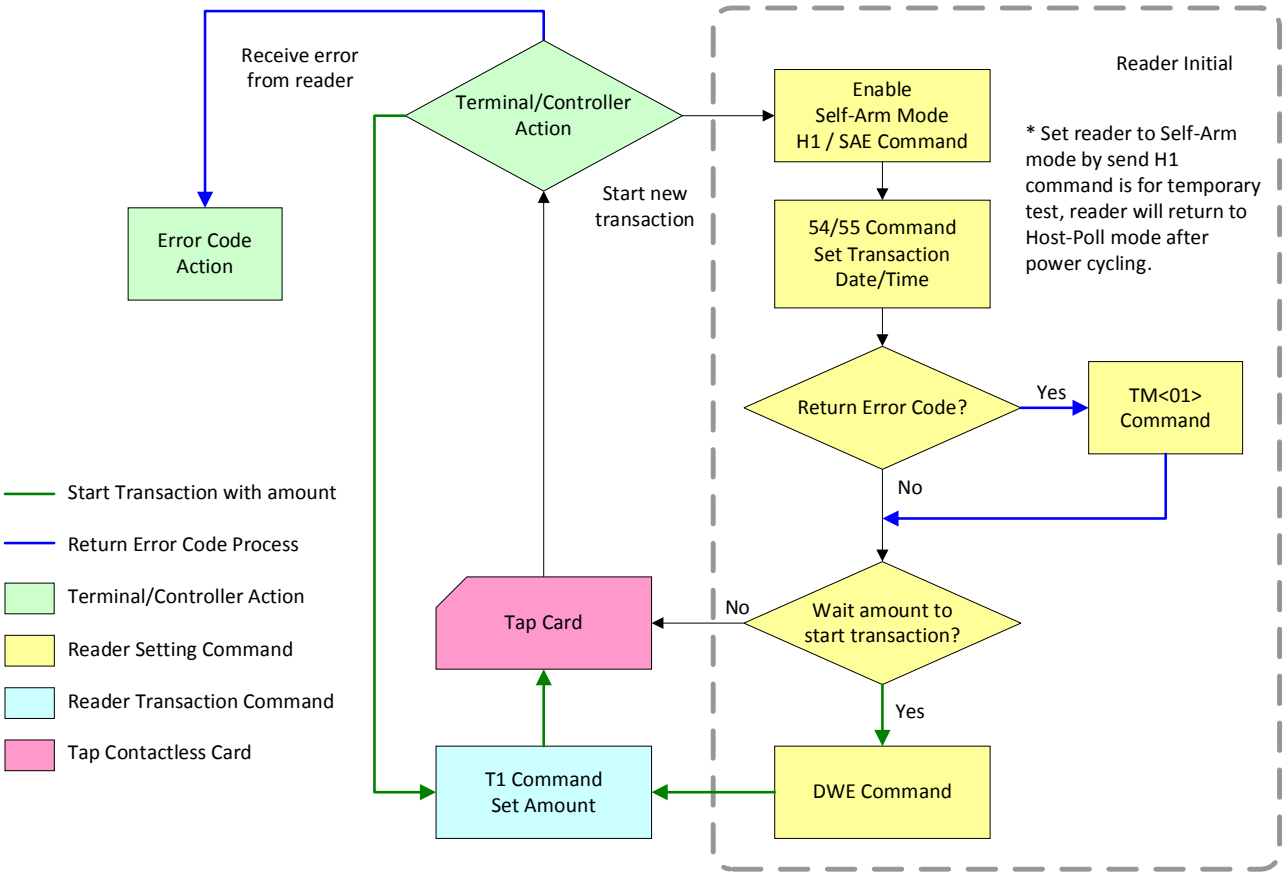


Figure 4-1. Self-Arm Mode Transaction Process Example Flow

Related Commands:

Function	Reference
Self-Arm Mode setting	4.2.1. H (48H) – Self-Arm function disable/enable 4.3.19. SAx(53h 41h x) — Self-Arm Mode Enable/Disable
Set Transaction Date/Time	4.1.11. *5 (35H) – Set RTC Time*
Return Error Code setting	4.3.22. *TMx(54h 4Dh x) — Set Error Code output Enable/ Disable*
Wait Amount mode setting	4.3.7. *DWx(44h 57h x) — Set Wait Amount mode*
Set Amount	4.2.6. *T (54H) – Transaction Comman*

Table 4-22. Commands related to Self-Arm mode transaction example flow

Host Poll Mode transaction process Example flow

In this Example, assume the *Bezel5* is in protocol USI2 and Self-Arm disable mode.

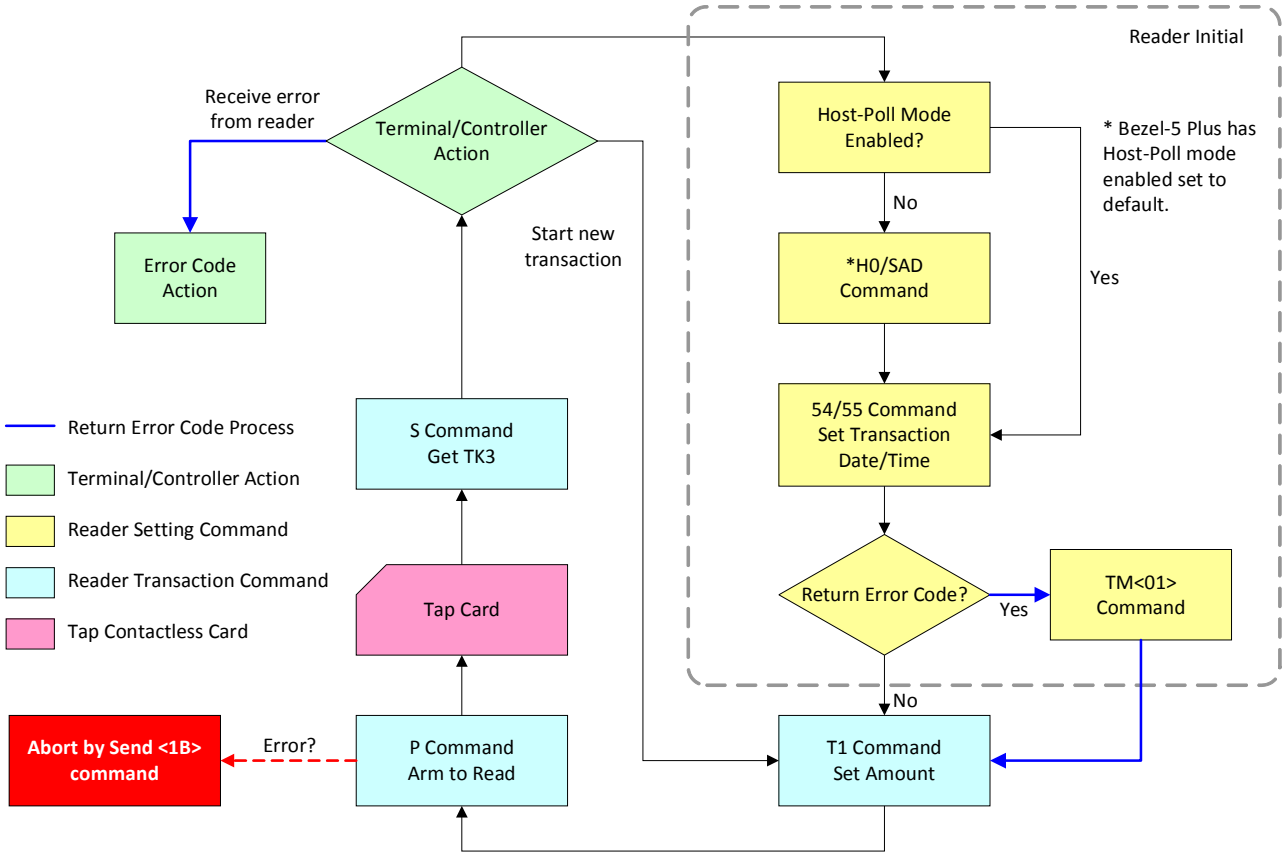


Figure 4-2. Host Poll Mode Transaction Process Example Flow

After the host issues the Arm-to-Read command, the *Bezel5* will check if any payment card is detected in the reading zone or any magnetic stripe card is swiped. No matter the card is decoded successfully or not, it will return the '^' to indicate that the card has been read. The host can issue the 'Q', 'R', 'S' commands to retrieve the card data accordingly.

Related Commands:

Function	Reference
Host-Poll Mode setting	4.2.1. H (48H) – Self-Arm function disable/enable 4.3.19. SAx(53h 41h x) — Self-Arm Mode Enable/Disable
Set Transaction Date/Time	4.1.11. *5 (35H) – Set RTC Time*
Return Error Code setting	4.3.22. *TMx(54h 4Dh x) — Set Error Code output Enable/ Disable*

Function	Reference
Set Arm to Read	4.2.2. *P (50H) – Arm to Read*
Set Amount	4.2.6. *T (54H) – Transaction Comman*
Get Transaction Data	4.2.5. *Q, R, S – Get Transmit Track Data*

Table 4-23. Commands related to Host-Poll mode transaction example flow

4.2.1. H (48H) – Self-Arm function disable/enable

Used for controlling the contactless auto read function temporarily.

Command Packet

Byte 0	Byte 1
H	Type

Command Type

ASCII - Hex 1 Byte Value	Description
0 (or 30h)	Self Arm Disable
1 (or 31h)	Self Arm Enable

Example

Host Command	Reader Response Example
H0	
	^

Note: *Bezel5* cannot perform the Self-Arm enable command for the contactless payment card reading under the following conditions:

1. The payment card is decoded successfully and the *Bezel5* is waiting for the card to be removed from the reading zone.
2. The payment card is failed to decode and the *Bezel5* is waiting for the card to be removed from the reading zone.

4.2.2. P (50H) – Arm to Read

1. Clears data buffers.
2. Transmits command acknowledgement (‘^’ 5EH) if successful.
3. Waiting for and detect for an approaching card.
4. The LED1 will light on and then turn off after a successful reading or a MIFARE card being detected.

Example

Host Command	Reader Response Example
P	
	^

Note:

1. After an Arm to Read command is received and acknowledged the only valid commands that will be accepted for execution are: <ESC> “Abort” and ‘$’ “Status”.
2. Reader will NOT send out track data automatically; the host has to issue the ‘Q’, ‘R’, ‘S’ commands to get the corresponding track data.
3. In the Self-Arm mode, it is not necessary to send this command. If this command is sent, it will temporarily override the Self-Arm mode.

4.2.3. p (70H) – Arm to Read (Used for Manufacturing Test Only)

Equivalent to the 'P' command, except that the card read acknowledgement is not the ‘^’ character.

Example

Host Command	Reader Response Example
p	
	^

It will report a ‘**(**‘ (28H) byte if the card media is detected, and a ‘**)**’ (29H) byte when the media detection goes inactive.

4.2.4. <ESC> (1BH) – Abort Arm to Read

1. Clear the data buffers.
2. Aborts the Arm to Read command.

Example

Host Command	Reader Response Example
<1B>	
	^

4.2.5. Q, R, S – Get Transmit Track Data

Q (51H) Transmit Track 1 data

R (52H) Transmit Track 2 data

S (53H) Transmit Track 3 data

Example

Host Command	Reader Response Example
S	
	+300=AABBCCDDEEFF1122=06011103900000=09861AA8=0003=A0800000=0000000000=0000=121207=000000000000=0840=2000=0840=000000000000=000888=25=0000=3030303030303030=3030303030303030303030303031=01=6000000001=======00=000000010000==2000========07=08?

4.2.6. T (54H) – Transaction Command

This command is used to view and manipulate transaction activities.

Command Packet

Byte 0	Byte 1	Byte 2 ~
Command	Type	Data
T	1	<00><00><00><00><01><00>

Example

Host Command	Reader Response Example
T1<00><00><00><00><01><00>	
	^

Command Type

Parameter	Description	Note
0 (or 30h)	Erase all Transaction records	Command Type 0, 2, B is related to Data Record, Bezel5 default is set to clear record before next transaction start, if transaction records required to be control manually, Host need to send ERx command to enable data record function. (Refer to Section 0
1 (or 31h)	Enter Amount	
2 (or 32h)	Read Transaction Record	
B (or 42h)	Query records Count	
5 (or 35h)	Clear Amount	

Command Data

Command Type	Data Description
T1<Amount>	6 bytes in numeric format, use once only.
T2<Record>	2 bytes in binary format, decide which record to read; range from 0001h to 0186h.
TB<Count> (Get Data)	2 bytes in binary format.

Note: Record data of ‘T2’ command includes Transaction Date and Tracks data.

4.3. Configuration Command Description

The configuration commands related to card brands transaction listed in this section only provide the usage which don't require to have EMV or any card brand certifications. Bezel5 has PayPass 3.0 supported, to process with EMV transaction flow, please use the EMV transaction operating commands (refer to *Section 5 EMV Transaction Operating Command*).

The following are configuration commands executed in BLP format.

BLP Protocol – RS232 Interface

Byte 1	Byte 2,3	Byte 4+n	Byte 5+n
09h	Command Len	Command/Data(n bytes)	BCC

Table 4-24. BLP Configuration Protocol

Response Code

Response	Meaning
^	Acknowledgement
*	Cannot execute (e.g. out of range)
!	Bad parameter (e.g. incorrect length)

4.3.1. CCx(43h 43h x) — Set Code

Command Packet

Byte 0	Byte 1	Byte 2 ~ 3 (or 4)
CC	Type	Data

Command Type

Type	Length	Description	Example (Hex)
1	2 Bytes	Country Code	09h 00h 05h 43h 43h 31h 08h 40h 75h
2	2 Bytes	Currency Code	09h 00h 05h 43h 43h 32h 08h 40h 76h
3	3 Bytes	Terminal Capabilities	09h 00h 06h 43h 43h 33h 00h 08h 88h BCh

Table 4-25. Set Configuration Code Table

4.3.2. CKx – Enable/Disable User CA Key

To enable/disable the user CA key (i.e. load by I command)

Command Packet

Byte 0~1	Byte 2
CK	Parameters, see Table 4-26

Command Type

Parameter	Example (Hex)	Description
00h	09h 00h 03h 43h 4Bh 00h 02h	User CA Key
01h	09h 00h 03h 43h 4Bh 01h 03h	Test CA key (default)

Table 4-26. Public Key switch Table

Note:

1. The host must send CK<00> to set user CA keys enabled if the new CA key is set via T03 command. Otherwise, the default test key is used. It impacts the offline authentication in EMV contactless only. For MSR contactless such as PayPass magstripe or VISA MSD, it has no effect.
2. The reader keeps the setting forever until the next CKx command.

4.3.3. CLx(43h 4Ch x) — Set TRM Parameters

Command Packet

Byte 0 ~ 1	Byte 2	Byte 3+n
CC	Type	Data

Command Type

Parameter	Length	Description	Example (Hex)
0	Var.	Dump setting parameters	09h 00h 03h 43h 4Ch 30h 35h
1	4 Bytes	Terminal Floor Limit	09h 00h 07h 43h 4Ch 31h 31h 30h 30h 32h 33h
2	6 Bytes	Threshold	09h 00h 09h 43h 4Ch 32h 31h 30h 30h 32h 30h 30h 3Dh
3	1 Byte	Target percentage	09h 00h 04h 43h 4Ch 33h 20h 11h
4	1 Byte	Max Target percentage	09h 00h 04h 43h 4Ch 34h 26h 10h

Table 4-27. Set TRM Parameters

Note:

1. Set Threshold data format is numeric, others are binary.
2. Target percentage range: 00h ~ 63h.
3. Max Target percentage range: 00h ~ 63h

4.3.4. CPx(43h 50h x) — PayPass Support

Command Packet

Byte 0 ~ 1	Byte 2
CP	Command Type, see Table 4-28

Command Type

Parameter	Example (Hex)	Description
00h	09h 00h 03h 43h 50h 00h 19h	MagStripe Only
01h	09h 00h 03h 43h 50h 01h 18h	MChip Enable

Table 4-28. Configure PayPass supporting mode

4.3.5. CTx(43h 54h x) — Set Terminal/Transaction Type/Info

Command Packet

Byte 0 ~ 1	Byte 2	Byte 3
CT	Type	Data, see Table 4-29

Command Type

Parameter	Length	Description	Example (Hex)
0	1 Byte	Terminal Type	09h 00h 04h 43h 54h 30h 22h 08h
1	1 Byte	Transaction Type	09h 00h 04h 43h 54h 31h 00h 2Bh
2	1 Byte	Transaction Info	09h 00h 04h 43h 54h 32h 40h 68h

Table 4-29. Set Terminal, Transaction Type/Info Table

4.3.6. DFx(44h 46h x) — Default Setting

The reader will restore the configuration to default manufacture settings after receiving this command.

Command Packet

Byte 0 ~ 1	Byte 2
DF	<00>

Important: It has to power-cycle the reader after this command is processed.

4.3.7. DWx(44h 57h x) — Set Wait Amount mode

This command can only operate in Self-Arm mode. If the wait amount mode is enabled, transaction can only be enabled by reader receiving amount through T1 command.

Command Packet

Byte 0 ~ 1	Byte 2
DW	Command Type, see Table 4-30

Command Type

Parameter	Length	Description	Example (Hex)
D	1 Byte	Disable	09h 00h 03h 44h 57h 44h 5Dh
E	1 Byte	Enable	09h 00h 03h 44h 57h 45h 5Ch

Table 4-30. Set Wait Amount mode

4.3.8. ECx(45h 43h x) — Extended Configuration Report Enable/Disable

Command Packet

Byte 0 ~ 1	Byte 2
EC	Command Type, see Table 4-31

Command Type

Parameter	Length	Description	Example (Hex)
D	1 Byte	Disable, returns one byte with configuration (Default)	09h 00h 03h 45h 43h 44h 4Bh
E	1 Byte	Enable, returns 16 bytes string with configuration	09h 00h 03h 45h 43h 45h 4Ah

Table 4-31. Extended Configuration Report Option

If the “Extended Configuration Report” is enabled, the Configuration Request Command ‘#’ will return an extended 16-byte string with configuration, else it will return a standard one byte.

4.3.9. EGx(45h 47h x) — Output Data Encryption Enable/Disable

Command Packet

Byte 0 ~ 1	Byte 2
EG	Command Type, see Table 4-32

Command Type

Parameter	Length	Description	Example (Hex)
00	1 Byte	Output clear data	09h 00h 03h 45h 47h 00h 08h
01	1 Byte	Output encrypted data	09h 00h 03h 45h 47h 01h 09h

Table 4-32. Output Data Encryption Setup

4.3.10. ERx(45h 52h x) — Record RF card data

Command Packet

Byte 0 ~ 1	Byte 2
ER	Command Type, see Table 4-33

Command Type

Parameter	Length	Description	Example (Hex)
00	1 Byte	Not record	09h 00h 03h 45h 52h 00h 1Dh
01	1 Byte	Record, but stop recording after memory full, continue reading	09h 00h 03h 45h 52h 01h 1Ch
02	1 Byte	Record, but stop reading after memory full, 3 beeps	09h 00h 03h 45h 52h 02h 1Fh
03	1 Byte	Record, but stop reading after memory full, 3 beeps and send '&' out	09h 00h 03h 45h 52h 03h 1Eh

Table 4-33. Record RF card data option

4.3.11. ESx(45h 53h x) — SS/ES Enable/Disable

Command Packet

Byte 0 ~ 1	Byte 2
ES	Command Type, see Table 4-34

Command Type

Parameter	Length	Description	Example (Hex)
D	1 Byte	Disable	09h 00h 03h 45h 53h 44h 58h
E	1 Byte	Enable	09h 00h 03h 45h 53h 45h 59h

Table 4-34. SS/ES Option

If SS/ES is enabled, each track data of magnetic stripe card that sent automatically in Self-Arm mode will be wrapped by the SS/ES character.

Note: This command is only effective in Self-Arming mode.

4.3.12. Fxy(46h x y) — Set Track 1, 2, 3 Prefix/Suffix Code, Preamble/Postamble Code

Command Packet

Byte 0	Byte 1	Byte 2 (max 5 bytes)
F	Type	Data (1 ~ 5 Bytes)

Command Type

Parameter (x)	Description	Example (Hex)	Example Data (y)
A	TK1 Prefix	09h 00h 07h 46h 41h 42h 45h 35h 30h 31h 3Ah	'BE501'
	TK1 Prefix Disable (Default)	09h 00h 03h 46h 41h 00h 0Dh	'00'
a	TK1 Suffix	09h 00h 07h 46h 61h 42h 45h 35h 30h 31h 1Ah	'BE501'
	TK1 Suffix Disable (Default)	09h 00h 03h 46h 61h 00h 2Dh	'00'
B	TK2 Prefix	09h 00h 07h 46h 42h 42h 45h 35h 30h 32h 3Ah	'BE502'
	TK2 Prefix Disable (Default)	09h 00h 03h 46h 42h 00h 0Eh	'00'
b	TK2 Suffix	09h 00h 07h 46h 62h 42h 45h 35h 30h 32h 1Ah	'BE502'
	TK2 Suffix Disable (Default)	09h 00h 03h 46h 62h 00h 2Eh	'00'
C	TK3 Prefix	09h 00h 07h 46h 43h 42h 45h 35h 30h 33h 3Ah	'BE503'
	TK3 Prefix Disable (Default)	09h 00h 03h 46h 43h 00h 0Fh	'00'
c	TK3 Suffix	09h 00h 07h 46h 63h 42h 45h 35h 30h 33h 1Ah	'BE503'
	TK3 Suffix Disable (Default)	09h 00h 03h 46h 63h 00h 2Fh	'00'
P	Set Preamble Code	09h 00h 07h 46h 50h 70h 72h 65h 30h 31h 7Eh	'pre01'
	Preamble Code Disable (Default)	09h 00h 03h 46h 50h 00h 1Ch	'00'
p	Set Postamble Code	09h 00h 06h 46h 70h 50h 4Fh 53h 54h 21h	'POST'
	Postamble Code Disable (Default)	09h 00h 03h 46h 70h 00h 3Ch	'00'

Table 4-35. Track Format Configuration Table

4.3.13. LB0x(4Ch 42h 30h x) — Set Read Card Mode

Command Packet

Byte 0 ~ 2	Byte 3
LB0	Command Type, see Table 4-36

Command Type

Parameter	Length	Description	Example (Hex)
00	1 Byte	All cards	09h 00h 04h 4Ch 42h 30h 00h 33h
02	1 Byte	PayPass	09h 00h 04h 4Ch 42h 30h 02h 31h
03	1 Byte	VISA	09h 00h 04h 4Ch 42h 30h 03h 30h
04	1 Byte	AMEX	09h 00h 04h 4Ch 42h 30h 04h 37h
05	1 Byte	Discover	09h 00h 04h 4Ch 42h 30h 05h 36h

Table 4-36. Set Read Card Mode

4.3.14. LCx(4Ch 43h x) — LRC Enable/Disable

If LRC is enabled, each track data sent automatically in self-arm mode will be followed by the LRC character.

Command Packet

Byte 0 ~ 1	Byte 2
LC	Command Type, see Table 4-37

Command Type

Parameter	Length	Description	Example (Hex)
D	1 Byte	Disable (Default)	09h 00h 03h 4Ch 43h 44h 41h
E	1 Byte	Enable	09h 00h 03h 4Ch 43h 45h 40h

Table 4-37. LRC Option

4.3.15. MFxy(4Dh 46h x y) — Set Payment Card and MIFARE Auto-Polling

Command Packet

Byte 0 ~ 1	Byte 2	Byte 3
MF	Type (Payment)	Type (Mifare)

Command Type (Payment)

Parameter	Length	Description	Example (Hex)
0	1 Byte	Disable Payment Card	09h 00h 04h 4Dh 46h 30h 31h 07h
1	1 Byte	Enable Payment Card	09h 00h 04h 4Dh 46h 31h 31h 06h

*Example set to Mifare Card enabled

Command Type (Mifare)

Parameter	Length	Description	Example (Hex)
0	1 Byte	Disable Mifare Card	09h 00h 04h 4Dh 46h 31h 30h 07h
1	1 Byte	Enable Mifare Card	09h 00h 04h 4Dh 46h 31h 31h 06h

*Example set to Payment Card enabled

Once MIFARE Auto-Polling is enabled, the reader will send out the following characters to the host if a MIFARE Card is detected.

Card Type

Response	Description
M1	MIFARE Ultralight
M2	MIFARE 1K
M3	MIFARE 4K
M4	MIFARE DESFire
M5	MIFARE Plus 2K
M6	MIFARE Mini
M7	MPCOS Gemplus
M8	Jewel for Innovision
M9	JCOP31
M0	Not MIFARE card or Not supported card

Table 4-38. Mifare Card Type Response table

4.3.16. PCx(50h 43h x) — Set Host Protocol

Command Packet

Byte 0 ~ 1	Byte 2
PC	Command Type

Command Type

Parameter	Length	Description	Example (Hex)
0	1 Byte	Switch to Protocol 2	09h 00h 03h 50h 43h 30h 29h
3	1 Byte	Switch to Protocol 0	09h 00h 03h 50h 43h 33h 2Ah

* The reader will warm-reset automatically after this command is received

4.3.17. PEx (50h 45h x) — Set Pass-Through Function

Command Packet

Byte 0 ~ 1	Byte 2
PE	Command Type

Command Type

Parameter	Length	Description	Example (Hex)
0	1 Byte	Pass-Through Disabled	09h 00h 03h 50h 45h 30h 2Fh
1	1 Byte	Pass-Through Enabled	09h 00h 03h 50h 45h 31h 2Eh

4.3.18. PHx(50h 48h x) — Set Power On Character

Command Packet

Byte 0 ~ 1	Byte 2
PH	Command Type

Command Type

Parameter	Length	Description	Example (Hex)
3Ah	1 Byte	Set power on character to " : "	09h 00h 03h 50h 48h 3Ah 28h
00h	1 Byte	Disable power on character	09h 00h 03h 50h 48h 00h 12h

*Default power on character is ":"

4.3.19. SAx(53h 41h x) — Self-Arm Mode Enable/Disable

Here is the difference between this command and the "H" command: If the reader is switched to Self-Arm mode enabled by "H" command, it will return to default after power cycling. If the reader is switched to Self-Arm mode by receiving the SAE command, the Self-Arm mode is kept enabled after power cycling.

Command Packet

Byte 0 ~ 1	Byte 2
SA	Command Type

Command Type

Parameter	Length	Description	Example (Hex)
D	1 Byte	Self-Arm mode Disabled	09h 00h 03h 53h 41h 44h 5Ch
E	1 Byte	Self-Arm mode Enabled	09h 00h 03h 53h 41h 45h 5Dh

4.3.20. SEx(53h 45h x) — Self-Arm Mode Data Envelope Enable/Disable

Command Packet

Byte 0 ~ 1	Byte 2
SE	Command Type

Command Type

Parameter	Length	Description	Example (Hex)
D	1 Byte	Disable. The data is not wrapped in the current protocol envelope (default)	09h 00h 03h 53h 45h 44h 58h
E	1 Byte	Enable. The data is wrapped in the current protocol envelope	09h 00h 03h 53h 45h 45h 59h

* In the self-arm mode, the default is not to send any protocol information with the magnetic stripe card data.

4.3.21. TKx(54h 4Bh x) — Set Transmitting Data Tracks

Command Packet

Byte 0 ~ 1	Byte 2
TK	Command Type, see Table 4-39

Command Type

Parameter	Length	Description	Example (Hex)
1	1 Byte	Track 1	09h 00h 03h 54h 4Bh 31h 24h
2	1 Byte	Track 2	09h 00h 03h 54h 4Bh 32h 27h
3	1 Byte	Track 1 & 2	09h 00h 03h 54h 4Bh 33h 26h
4	1 Byte	Track 3	09h 00h 03h 54h 4Bh 34h 21h
5	1 Byte	Track 1 & 3	09h 00h 03h 54h 4Bh 35h 20h
6	1 Byte	Track 2 & 3	09h 00h 03h 54h 4Bh 36h 23h
7	1 Byte	Track 1, 2 & 3(default)	09h 00h 03h 54h 4Bh 37h 22h

Table 4-39. Set Transmitting Data Tracks

4.3.22. TMx(54h 4Dh x) — Set Error Code output Enable/ Disable

Command Packet

Byte 0 ~ 1	Byte 2
TM	Command Type

Command Type

Parameter	Length	Description	Example (Hex)
00h	1 Byte	Disabled	09h 00h 03h 54h 4Dh 00h 13h
01h	1 Byte	Enabled	09h 00h 03h 54h 4Dh 01h 12h

4.3.23. TOx(54h 4Fh x) — Set Transmitting Data Output Format

Command Packet

Byte 0 ~ 1	Byte 2
TO	Command Type

Command Type

Parameter	Length	Description	Example (Hex)
0	1 Byte	Protocol 0	09h 00h 03h 54h 4Fh 30h 21h
2	1 Byte	Protocol 2	09h 00h 03h 54h 4Fh 32h 23h

4.3.24. USBx(55h 53h 42h x) — USB Mode (Optional)

Command Packet

Byte 0 ~ 2	Byte 3
USB	Command Type

Command Type

Parameter	Length	Description	Example (Hex)
00h	1 Byte	HID_KBD	09h 00h 04h 55h 53h 42h 00h 49h
01h	1 Byte	CDC	09h 00h 04h 55h 53h 42h 01h 48h
02h	1 Byte	HID_MSR	09h 00h 04h 55h 53h 42h 02h 4Bh

4.3.25. UTx(55h 54h x) — Set TAC

Command Packet

Byte 0 ~ 1	Byte 2	Byte 3 ~ 7
UT	Command Type, see Table 4-40	TAC Parameters

Command Type

Parameter	Length	Description	Example (Hex)
0	1 Byte	Default	09h 00h 08h 55h 54h 30h 00h 00h 00h 00h 00h 30h
1	1 Byte	Denial	09h 00h 08h 55h 54h 31h 00h 00h 00h 00h 00h 31h
2	1 Byte	Online	09h 00h 08h 55h 54h 32h 00h 00h 00h 00h 00h 32h

Table 4-40. Set TAC Table (for PayPass Only)

4.3.26. VTx(56h 54h x) — VISA Terminal Transaction Qualifier(Tag '9F66') Setting

Command Packet

Byte 0 ~ 1	Byte 2 ~ 5
VT	TTQ parameters

Command Type

Parameter	Length	Description	Example (Hex)
A0 00 00 00	4 Bytes	MSD & qVSDC	09h 00h 06h 56h 54h A0h 00h 00h 00h ADh

4.3.27. VVx(56h 56h x) — VISA Version setting

Command Packet

Byte 0 ~ 1	Byte 2
VV	Command Type

Command Type

Parameter	Length	Description	Example (Hex)
02h	1 Byte	Auto Polling Mode	09h 00h 03h 56h 56h 02h 08h
03h	1 Byte	Visa 2.1 Only	09h 00h 03h 56h 56h 03h 09h

4.3.28. VLx(56h 4Ch x) — VISA CVM Required Limit setting

Command Packet

Byte 0 ~ 1	Byte 2 ~ 7
VL	CVM Required Limit value

Example

Parameter	Length	Description	Example (Hex)
31h 30h 30h 30h 30h 30h	6 Bytes	Amount=$313,030,303,030	09h 00h 08h 56h 4Ch 31h 30h 30h 30h 30h 1Ah

*CVM Required Limit value is fixed at 6 bytes and data format is numeric.

4.4. Contactless Operation Commands Description

4.4.1. G (47H) – ISO 14443 Type Protocol Select

Select which manual command to be operated – ISO 14443 Type A or B.

Command Packet

Byte 0	Byte 1
Command	Type
G	0

Command Type

Type	Description
0 (30h, ASCII Hex)	ISO 14443 Type A
4 (34h, ASCII Hex)	ISO 14443 Type B

Note: The default contactless smart card type is Type A after power up.

Example

Host Command	Reader Response Example
G0	
	^

4.4.2. O (4FH) – Antenna power ON

To apply power to the antenna. This command is for manual command operation.

Example

Host Command	Reader Response Example
O	
	^

Note: If the reader is in Self-Arm mode. The antenna power cannot be turned on by manual command setting.

4.4.3. o (6FH) – Antenna power OFF

To Turns off the antenna power

Example

Host Command	Reader Response Example
o	
	^

Note: If the reader is in Self-Arm mode. The antenna power cannot be turned off by manual command setting.

4.4.4. b (62H) – Request

The 'Request' command.

Command Packet - ISO 14443 type A

Byte 0	Byte 1
Command	Req command
b	52

The request command code is ISO 14443 type A. It can be either 26(REQA) or 52(WUPA).

Note: If the [Req command] field does not appear in the request command, reader will set the request mode to WUPA automatically.

Command Packet - ISO 14443 type B

Byte 0	Byte 1	Byte 2
Command	AFI	PARAM
b	00	00

Command Description

Byte	Description
AFI(optional)	Binary Hex(00h to FFh), please refer to ISO 14443-3 for detailed information.
PARAM(optional)	Binary Hex(00h to FFh), please refer to ISO 14443-3 for detailed information.

If the [AFI] and [PARAM] fields do not appear in the request command, reader will set the request mode to WUPB automatically.

Success Response Data Format

Message Type	Description
ATQA	2 bytes, type A, Binary Hex
ATQB	16 bytes, type B, Binary Hex

Note: If reader response '*' = No card response or No power on the antenna

4.4.5. c (63H) – Anti-collision(type A)/Slot-MARKER(type B)

In type A mode, reader sends the ANTICOLLISION command to the card.
In type B mode, reader sends the Slot-MARKER command to the card.

Command Packet -

Card Type	Byte 0	Byte 1
ISO 14443 type A	c	
ISO 14443 type B	c	APn

Command Description

Byte	Description
APn	Anti-collision Prefix byte, please refer to ISO 14443-3 for detailed information.

Success Response Data Format

Card Type	Description
ISO 14443 type A	PICC serial number for type A(Binary Hex)
ISO 14443 type B	PICC send ATQB(12 bytes, Binary Hex) for type B

Note: If reader response '*' = No card response or No power on the antenna

4.4.6. f (66H) – Select(type A)/Attrib(type B)

In type A mode, reader sends the SELECT command to the card.
In type B mode, reader sends the ATTRIB command to the card.

Example

Host Command	Reader Response (ISO 14443 Type A)	Reader Response (ISO 14443 Type B)
f		
	'^' + SAK(1 byte)	'^' + MBLI/CID(1 byte)

'*' - No card response or No power on the antenna

4.4.7. g (67H) – MIFARE Classic Card Authentication

An authentication command has to be carried out before any operation in order to allow further commands.

Command Packet

Byte 0	Byte 1-3	Byte 4	Byte 5
Command	Block number	Key number	Key type
g	001	0	A

Or

Byte 0	Byte 1-3	Byte 4	Byte 5-16
Command	Block number	Key Type	Key
g	001	A	FFFFFFFFFFFF

Authenticate the card with the key stored in EEPROM.

Block Number – 2 Types

Block Number Type	Data Format
000 to 255	30h30h30h to 32h35h35h, ASCII Hex
B<00><00> to B<00><FF>	42h00h00h to 42h00hFFh, ASCII Hex

Key Information

Field	Length	Description
Key Number	1 Byte	0 to 4(30h to 34h, ASCII Hex)
Key Type	1 Byte	A or B(41h or 42h)
Key	12 Bytes	0 to 9 or A to F(30h to 39h or 41h to 46h, ASCII Hex)

Example

Host Command	Reader Response Example
g001AFFFFFFFFFFFF	
	^

4.4.8. h (68H) – MIFARE Classic Card Read Block(Supports MIFARE Ultralight)

MIFARE Classic card read command.

Command Packet

Byte 0	Byte 1-3
Command	Block number
h	001

Block Number – 2 Types

Block Number Type	Data Format
000 to 255	30h30h30h to 32h35h35h, ASCII Hex
B<00><00> to B<00><FF>	42h00h00h to 42h00hFFh, ASCII Hex

Example

Host Command	Reader Response Example
h001	
	1111111111111111

Response Block data (16 bytes, Binary Hex)

4.4.9. i (69H) – MIFARE Classic Card Write Block(Supports MIFARE Ultralight)

MIFARE Classic card write command.

Command Packet

Byte 0	Byte 1-3	Byte 4-7 or Byte 4-19
Command	Block number	Block data
i	001	1234123412341234

Block Number – 3 Types

Block Number Type	Data Format	Description
000 to 255	30h30h30h to 32h35h35h, ASCII Hex	General MIFARE block
B<00><00> to B<00><FF>	42h00h00h to 42h00hFFh, ASCII Hex	
<00><00> to U<00><FF>	55h00h00h to 55h00hFFh, ASCII Hex	MIFARE Ultralight

Block Data

Card Type	Length
MIFARE Ultralight	4 Bytes
Others	16 Bytes

Example

Host Command	Reader Response Example
i00112341234123412341234	
	^

4.4.10. t (74H) – MIFARE Classic Card Value Operation

Value Block Operation commands.

Command Packet

Byte 0	Byte 1-3	Byte 4	Byte 5-8	Byte 9-11
Command	Block number	Operation mode	Value	Transfer block
t	001	3	00	02

Block Number – 2 Types

Block Number Type	Data Format
000 to 255	30h30h30h to 32h35h35h, ASCII Hex
B<00><00> to B<00><FF>	42h00h00h to 42h00hFFh, ASCII Hex

Operation Mode

ASCII – Hex Value	Description
0 (or 30h)	Decrement
1 (or 31h)	Increment
2 (or 32h)	RFU
3 (or 33h)	Decrement and transfer to the different block
4 (or 34h)	Create MIFARE Value in the block

Others

Field	Description
Value	Binary Hex from 00h to FFh
Transfer block	For option 3 only, the data format is the same as [Block number]. If [transfer block] is not given, reader will regard it as normal Decrement command.

Example

Host Command	Reader Response Example
t00140002	

Host Command	Reader Response Example
	^
t00100001	
	^
t00110001	
	^
t00130001002	
	^

4.4.11. W (57H) – ISO 14443A Detection

To detect the ISO 14443A cards

Response 'M' if detects an ISO 14443A card.

4.4.12. X (58H) – MIFARE Classic Card Activation (Supports MIFARE Ultralight)

Performs the request/anti-collision/select commands to activate the card

It is also can be used for any ISO 14443 compatible cards.

Card Type	Description
Type A	ATQA/SAK/serial number if command executed successfully
Type B	ATQB(12 bytes) if command executed successfully

Example

Host Command	Reader Response Example
X	
	<04><00><08>d<AC>Eq

4.4.13. u (75H) – MIFARE Classic Card Read Sector

MIFARE Classic card read sector command.

Command Packet

Byte 0	Byte 1-3
Command	Block number
u	001

Block Number – 2 Types

Block Number Type	Data Format
000 to 255	30h30h30h to 32h35h35h, ASCII Hex
B<00><00> to B<00><FF>	42h00h00h to 42h00hFFh, ASCII Hex

Note: For MIFARE Classic 4K, sectors 0~31 contains 4 blocks each and sectors 32~39 contains 16 blocks each.
Response Sector data (64/256 bytes, depending on the card) if command executed successfully

4.4.14. v (76H) – MIFARE Classic Card Write Sector

MIFARE Classic card write sector command.

Command Packet

Byte 0	Byte 1-3	Byte 4~
Command	Sector number	Sector Data

Sector Number – 2 Types

Block Number Type	Data Format
000 to 255	30h30h30h to 32h35h35h, ASCII Hex
B<00><00> to B<00><FF>	42h00h00h to 42h00hFFh, ASCII Hex

Sector Data
For MIFARE Classic 4K, sectors 0~31 contains 4 blocks each and sectors 32~39 contains 16 blocks each. That is, the sector data should be 64 bytes for MIFARE 1K card, 64/256 bytes for MIFARE 4K card.

4.4.15. J (4AH) – Activate PICC cpu card

PICC cpu card activation command. The Antenna POWER ON command has to be sent first.
Response ATS (type A) or PUPI (type B) if the command is executed successfully

4.4.16. j (6AH) – Load MIFARE Key(Supports MIFARE Classic only)

Saves up to 5 key sets for MIFARE Classic card application
Note: For security reasons, there is no way to retrieve the keys.

Command Packet

Byte 0	Byte 1	Byte 2~13
Command	Key number	Key data

Key Information

Field	Data Format
Key number	0 to 4 (or 30h to 34h, ASCII Hex)
Key data	0 to 9 or A to F(or 30h to 39h or 41h to 46h, ASCII Hex)

4.4.17. F (58H) – Identify MIFARE Card Type

To reports the MIFARE Card type. It also can be used for any ISO 14443A compatible cards.

Host Command	Reader Response Example
F	
	Card Type*

Card Type*

Response	Description
1 (31H)	MIFARE Ultralight
2 (32H)	MIFARE 1K
3 (33H)	MIFARE 4K
4 (34H)	MIFARE DESFire
5 (35H)	MIFARE Plus 2K
6 (36H)	MIFARE Mini
7 (37H)	MPCOS Gemplus
8 (38H)	Jewel for Innovision
9 (39H)	JCOP31
0 (30H)	Not MIFARE card or Not supported card
’*’	No card response or No power on the antenna

Note: This command is only available after users successfully activate the MIFARE cards (after the ‘f’ or ‘X’ command).

4.4.18. y (79H) – Send DESELECT command

Sends the ISO 14443 layer 4 DESELECT command to the card.

4.4.19. Z (5AH) – I/O to contactless CPU card with APDU format

The command is used to pass an APDU to the card where both data and an ISO status are expected in the response.

Command Packet

Byte 0	Byte 1~ (262 Bytes max)
Command	APDU (Binary hex(00h to FFh))

If successful, the data from the ICC and the two bytes SW1/SW2 ISO 7816-4 response are returned.

If unsuccessful, reader transmits '*'.

APDU Command Structure					
CLA	INS	P1	P2	P3 (Lc or Le)	Data (If Lc present)

APDU Response Structure		
Data (optional)	SW1	SW2

4.4.20. z (7AH) – I/O to contactless card for block data exchange

The command is used to pass a block data to a card.

Command Packet

Byte 0	Byte 1	Byte 2-5	Byte 5~ (384 Bytes max)
Command	CRC mode	Wait time	Block data

CRC Mode

Mode	Description
0 (30h, ASCII Hex)	Block data contain 2 bytes CRC and enable CRC transmission.
1 (31h, ASCII Hex)	No CRC in block data and disable CRC transmission.

Others

Field	Description
Wait time	0000 to 9999(30h30h30h30h to 39h39h39h39h, ASCII Hex) in milliseconds.
Block data	Binary hex(00h to FFh), maximum 384 bytes.

If successful, the data from the ICC are returned.
If unsuccessful, reader transmits '*'.

5. EMV Transaction Operating Command

In order to process EMV transactions, the reader must be initialized properly according to the transactions it has to support. Known as EMV application configuration, the controller needs to configure the reader with the necessary application data. All of the application data is stored in the nonvolatile memory of the reader and is set once before the reader is deployed to the field site. However, it is also possible to update that configuration data via the remote downloading process if a new application is required to be supported by the reader.

There are three different groups of reader configurations:

1. Terminal Configuration: there is only one terminal configuration data set per reader.

2. Application Configuration: multiple applications are allowed to be saved at the reader (up to 11 applications). Each application with a unique AID is associated with its own set of application data.

3. CA Public Key: the RSA public keys are stored at the reader. Up to a maximum of 30 public keys are accommodated in the secure area of the reader.

All of the data in the parameter field of the Terminal configuration and Application configuration must be presented in the TLV binary format. For Example, the tag 9F35 with 1 bytes data length and the data is 22h. It will use 4 bytes as 9Fh, 35h, 01h, and 22h.

Byte 1	Byte 2	Byte 3	Byte 4
9Fh	35h	01h	22h

5.1. Configuration Commands

The following are configuration commands executed in BLP format

BLP Protocol – RS232 Interface

Byte 1	Byte 2,3	Byte 4+n	Byte 5+n
09h	Command Len	Command/Data(n bytes)	BCC

Response Code

Response	Meaning
^	Acknowledgement
*	Cannot execute (e.g. out of range)
!	Bad parameter (e.g. incorrect length)

5.1.1. T01 (54H, 30H, 31H) – Terminal Configuration Setup

The EMV application uses this command to send the Terminal Configuration Data to the reader.

Command

Byte 1,2,3	Byte 4+n
T01	TLV Data Object List (n bytes)

Required TLV Parameters

Tag	Description	Remarks
9F15	Merchant Category	
9F16	Merchant ID	
9F1A	Terminal Country Code	
9F1C	Terminal ID	
9F1E	IFD Serial NO	
9F35	Terminal Type	
9F4E	Merchant Name and Location	Optional

Table 5-1. Terminal Configuration Setup Tag list

Note:
1. The reader will reject the command if the data is in non-TLV format and with invalid coding.
2. For all the unknown tags or tags with incorrect values, it will be ignored by the reader.
3. For the duplicate tags, the reader always overwrites the earlier tag value by the latter tag.
4. The reader accepts partial data update of TLV data.
5. See appendix A for the terminal default value.

Example - Update 9F 1A Terminal Country Code:

Host Command	Reader Response Example
<09><00><08>T01<9F><1A><02><08>@<9B>	
	^

5.1.2. T03 (54H, 30H, 33H) – Certificate Authority Public Key Setup

The EMV application uses this command to send the Certificate Authority Public key data to the bezel. The key will be used in the EMV transaction.

Command

Byte 1,2,3	Byte 4,5,…,13	Byte 14, 15	Byte 16, 17	Byte 18,19,…,57	Byte 58,59	Byte 60,61	Byte 62,63,…, 67	Byte 68+n
T03	RID	PKI	Hash algo	Hash	PK Algo	PK len	PK Exponent	PK Modulus (n bytes)

Parameter description

Parameters	Length	Description
RID	10 bytes	Hexadecimal string, the left 5 bytes of EMV Application ID.
PKI	2 bytes	Public Key Index, hexadecimal string. (Refer to EMV 4.1, tag '9F22')
Hash Algo	2 bytes	Hash Algorithm Index, hexadecimal string. '01': SHA-1 is the only acceptable value.
Hash	40 bytes	(same calculation method issued by the card brand association)
PK Algo	2 bytes	Public Key Algorithm, hexadecimal string. '01': RSA digital signature is the only acceptable value.
PK len	2 bytes	Public Key size, hexadecimal string, for Example '80' = 128 bytes = 1024 bits
PK Exponent	6 bytes	Public Key Exponent's size, hexadecimal, '000003' or '010001'
PK Modulus	Var. bytes	Public Key Modulus, presented in hexadecimal, data length = 2*[PK length]

Table 5-2. Certificate Authority Public Key parameters description

5.1.3. T15 (54H, 31H, 35H) – Contactless Application Configuration Setup

The EMV application uses this command to send one set of EMV application configuration data to the reader. T15 command is acceptable by the reader up to a maximum of 1K bytes for one application. A total of 11 applications can be stored. The command will be rejected if it goes beyond the max number of the application configurations. Please use the T1B command to delete the unnecessary application configuration.

Command Packet

Byte ,1,2,3	Byte 4+n
T15	Data Object (TLV format) (n bytes)

Required TLV Parameters

Tag	Data Object Name	Format	Length (Byte)
Mandatory Tags			
9F06	Application Identifier (AID) –card	b	5-16
9C	Transaction Type	b	1
Group Tags (Can be sent individually or combined with other tags together)			
FFFF8001	Registered Application Provider Identifier (RID)	b	5
FFFF8002	Application Selection Indicator	n	1
FFFF8003	Kernel ID	n	1
FFFF8004	Disable Contactless Transaction Limit	b	1
FFFF8005	Zero allow	b	1
FFFF8006	CVN17 Enable (VISA)	b	1
FFFF8007	Sign Unit Check (VISA)	b	1
FFFF8008	Amount Option 1⁄2 Select (VISA)	b	1
FFFF8009	CVM Require Limit Check Enable (VISA)	b	1
FFFF800A	Reader Contactless Floor Limit Check (VISA)	b	1
FFFF800B	Online Capable Disable (VISA)	b	1
FFFF800C	Exception Check Enable	b	1
FFFF800D	ReFund	b	1
FFFF8101	Terminal Contactless Floor Limit	n12	6
FFFF8102	Terminal Contactless Transaction Limit	n12	6
FFFF8103	CVM Required Limit	n12	6
FFFF8201	Terminal Action Code (Online)	b	5
FFFF8202	Terminal Action Code (Default)	b	5

Tag	Data Object Name	Format	Length (Byte)
FFFF8203	Terminal Action Code (Denial)	b	5
FFFF8204	Terminal Entry Capability (VISA)	b	1
FFFF8205	Time-Out Value	b	4
FFFF8206	Retry Counter for Wait Online Response (DPAS)	b	1
FFFF8207	Time-Out for Wait Online Response	b	4
FFFF8208	Transaction Info	b	1
FFFF8209	Default TDOL	b	n
FFFF820A	Default PDOL	b	n
FFFF8210	Paypass Phone Message Table	b	n
FFFF8211	Certification Revocation List	b	n
FFFF8212	Paypass Signal Out	b	n
FFFF8213	Paypass Message Out	b	n
9F09	Application Version Number (M/Chip) (Value = 00 02)	b	2
9F6D	Application Version Number (MagStripe) (Value = 00 01)	b	2
9F1B	Terminal Floor Limit	b	4
9F33	Terminal Capabilities	b	3
DF2A	Threshold Value for Biased Random Selection	b	6
DF2B	Maximum Target Percentage for Biased Random Selection	b	1
DF2C	Target Percentage for Random Selection	b	1
9F40	Additional Terminal Capabilities	b	6
9F1D	Terminal Risk Management Data	b	1
9F66	Terminal Transaction Qualifiers (TTQ)	b	4
FFFF820B	Application Program IDs (VISA)	b	16
FFFF820C	Single Unit Value	b	1

Table 5-3. Application Configuration Tag List

Note:

1. Tag 9f 06 (AID) and Transaction Type (9C) are the mandatory tag for each T15 command. UIC680 use AID and Transaction Type to identify the group tags to be stored in the proper location.
2. The reader will reject the command if the data is non-TLV format or with invalid coding.
3. For the unknown tags or tags with incorrect values, it will be ignored by the reader.
4. For the duplicate tags, the reader always overwrites the earlier tag value by the latter tag.
5. The reader accepts the partial data update TLV data.
6. See appendix A for the application terminal default value.

Example - Update 9F66 of VISA AID (A0 00 00 00 03 10 10) in the group tags:

Host Command	Reader Response Example
<09><00><17>T15<9F><06><07><A0><00><00><00><03><10><10><9C><01> <00><9F>f<04><80><00><00><00><93>	
	^

5.1.4. T19 (54H, 31H, 39H) – EMV Contactless Configuration Data Query

To retrieve the group ID of the EMV application or the CA public key stored in the rerader.

Command Packet

Byte 1,2,3	Byte 4
T19	Configuration Type Table 5-4

Configuration Type

Parameter	Description
31h	All the IDs of CA public key, setup by T03.
32h	All the IDs of EMV application data, setup by T15.
33h + AID + Transaction Type	Read data setting of AID and Transaction Type. (Ex: A000000310109C0100)
34h	Read data of terminal

Table 5-4. EMV Contactless Configuration Data Query Type

Response

Result	Response	Description
Success	ID List	The concatenation of IDs. There is a <1C> between each ID. Only present if the result is successful
Failed	*	Bad parameters
	!	Can't execute

Example

Host Command	Reader Response Example
<09><00><04>T191`	
	A00000015201<1C>A00000015203
<09><00><04>T192c	

Host Command	Reader Response Example
	A0000000031010<1C>A0000000999090<1C>A0000000032010<1C>A0000000041010<1C>A0000000043060<1C>B012345678<1C>A00000002501<1C>A0000003241010<1C>A0000001523010
<09><00><13>T193A0000000031010<00><07>	<9F><06><07><A0><00><00><00><03><10><10><9C><01><00><FF><FF><80><02><01><01><FF><FF><80><03><01><03><FF><FF><80><04><01><01><FF><FF><80><05><01><01><9F><1B><04><00><00>'<10><FF><FF><81><01><06><00><00><00><10><00><00><FF><FF><81><02><06><00><00><00>P<00><00><FF><FF><81><03><06><00><00><00><20><00><00><9F><09><02><00><02><9F>3<03><00><08><88><9F>f<04><A0><00><00><00><9F><1A><02><08>@<FF><FF><80><06><01><01><FF><FF><80><07><01><01><FF><FF><80><08><01><00><FF><FF><80><09><01><00><FF><FF><80><0A><01><01><FF><FF><80><0B><01><00><9F>5<01>%
<09><00><04>T194e	
	<9F><15><08>00000000<9F><16><0F>0000000000000001<9F><1A><02><08>@<9F><1C><08>00000000<9F><1E><08>00000000<9F>5<01><00>

5.1.5. T1B (54H, 31H, 42H) – Delete EMV Contactless Configuration Data

To delete the EMV application or the CA public key stored in the reader.

Command Packet

Byte 1,2,3	Byte 4	Byte 5	Byte 6+n
T1B	Configuration Type	<1A> (Optional)	ID List (Optional)* (n bytes)

*The concatenation of IDs. There is a <1C> between each ID.

Configuration Type

Parameter	Description
31h	All the IDs of CA public key, setup by T03.
32h	All the IDs of EMV application data, setup by T15.
33h	Delete all CA public keys.
34h	Delete all EMV application data.
35h	Delete Terminal Setting.

Example

Host Command	Reader Response Example
<09><00><04>T1B4<1E>	
	^

5.1.6. T0C (54H, 30H, 43H) –Configuration Version/Checksum

To retrieve the checksum/version of the EMV application or the CA public key stored in the reader.

Command Packet

Byte 1,2,3	Byte 4	Byte 5
T0C	Mode, see Table 5-5	Options, see Table 5-6

Mode

Parameter	Description
31h	Terminal data checksum request.
32h	EMV Contactless application data checksum request.
33h	Public key data checksum request.

Table 5-5. Configuration Version/Checksum Mode

Options

Parameter	Description
AID/(RID+CAPKI) + Transaction Type	To read EMV application data checksum request, user need to enter AID string and Transaction Type. (Ex. A0000000041010<00>)
AID/(RID+CAPKI)	To read Public key data checksum request, user need to input AID String. (Ex: A0000000031010)

Table 5-6. Configuration Version/Checksum Options

Response

Result	Response	Description
Success	20 bytes SHA1 checksum	Only present if the result is successful
Failed	*	Bad parameters
	!	Can't execute

Example

Host Command	Reader Response Example
<09><00><04>T0C1<1B>	
	<92><00>cX<A6><04>o<0E><8F>y<A1><F0><95>-<20>@<CE>Q<A8>o
<09><00><13>T0C2A0000000031010<00>}	
	<9F><F0><AD><F2><0B>~z<AE><02>om<D0><E8>d<CD><D6><20>B<D9><D7>
<09><00><10>T0C3A00000015201{	
	<B0><80>1<BD><A9><C3>)<1A>><8B><9C>y9<15><F2>G<A3><84>k<8F>

5.1.7. T1C (54H, 31H, 43H) –Terminal and Application List Default Setting

To restore the default terminal and application data in the reader (For testing only)

Command Packet

Byte 1~3
T1C

Example

Host Command	Reader Response Example
T1C	
	^

1. This command will take 15 ~ 20 seconds to update EEPROM.
2. Note 2: Be careful to use this command because the previous data will be changed permanently.

Default terminal and application data

Terminal Configuration Settings

Tag	Length	Value	Description
9F1A	02	0056	Terminal Country Code
9F1C	08	3030303030303031	Terminal Identification
9F1E	08	3030303030303031	Interface Device (IFD) Serial Number
9F4E	08	3030303030303031	Merchant Name and Location
DF811C	02	0000	Max Lifetime of Torn Transaction Log Record
DF811D	01	00	Max Number of Torn Transaction Log Records
FFFF8211	81C4	A0000000045CF85A0000100000110001010001100001110010000010010010100010110011000011010011100011110100000100010100100100110101100011001011010011011011100011101011110000100001B0123456785CF85A0000100000110001010001100001110010000010010010100010110011000011010011100011110100000100010100100100110101000101010101110110000110010110100110110111000111010111100111111100000100001	Certification Revocation List

Table 5-7. Terminal Configuration Settings Tag List

Visa Application Identifier

Tag	Length	Value	Description
9F06	07	A0000000031010	AID Visa
9C	01	00	Transaction Type
FFFF8002	01	01	ASI
DF810C	01	03	Kernel ID
FFFF800F	01	00	Dynamic Reader Limits Enable
FFFF8007	01	01	Status Check(Signal Unit Enable)
FFFF8005	01	01	Zero allow
FFFF8008	01	01	Select Amount Option 1/2
FFFF8004	01	01	Disable Contactless Transaction Limit
DF8124	06	000000003000	Reader Contactless Transaction Limit
FFFF8009	01	01	CVM Required Limit Check
DF8126	06	000000001000	CVM Required Limit
FFFF800A	01	01	Reader Contactless Floor Limit Check
DF8123	06	000000002000	Reader Contactless Floor Limit
9F1B	04	000007D0	Reader Floor Limit
9F09	02	0002	Application Version Number
9F66	04	A6004000	Terminal Transaction Qualifiers
FFFF8006	01	01	CVN17 Enable
FFFF800B	01	01	Online Capable Enable
FFFF800C	01	00	Exception Check Enable
9F35	01	25	Terminal Type
9F1A	02	0840	Country Code
9F33	03	000888	Terminal Capabilities

Table 5-8. Visa Application ID Default Tag Value

PayPass Application Identifier

Tag	Length	Value	Description
9F06	07	A0000000041010	AID MaestroCard
9C	01	00	Transaction Type
FFFF8002	01	01	Application Selection Identifier
FFFF8004	01	01	Disable Contactless Transaction Limit
FFFF8005	01	01	Zero allow

Tag	Length	Value	Description
FFFF8007	01	01	Sign Unit Check
FFFF8009	01	01	CVM Require Limit Check Enable
FFFF8010	01	00	Extended Selection Support flag
FFFF8208	01	40	Transaction Info
5F57	00	N/A	Account Type
9F01	00	N/A	Acquirer Identifier
9F40	05	0000000000	Additional Terminal Capabilities
9F09	02	0002	App Version
DF8117	01	00	Card Data Input Capability
DF8118	01	60	CVM Capability-CVM Required
DF8119	01	08	CVM Capability-No CVM Required
DF811A	03	9F6A04	Default UDOL
DF8130	00	N/A	Hold Time Value
DF811B	01	20	Kernel Configuration
DF810C	01	02	Kernekl ID
9F6D	02	0001	Mag-stripe Application Version Number
DF811E	01	10	Mag-stripe CVM Capability-CVM Required
DF812C	01	00	Mag-stripe CVM Capability-No CVM Required
9F15	02	0001	Merchant Category Code
DF812D	03	000000	Message Hold Time
9F7E	00	N/A	Mobile Support Indicator
DF8123	06	000000010000	Reader Contactless Floor Limit
DF8124	06	000000030000	Reader CTL (No On-device CVM)
DF8125	06	000000050000	Reader CTL (On-device CVM)
DF8126	06	000000001000	CVM Required Limit
DF811F	01	08	Security Capability (CDA)
DF8120	05	0000000000	Terminal Action Code-Default
DF8121	05	0000000000	Terminal Action Code-Denial
DF8122	05	0000000000	Terminal Action Code-Online
9F33	00	N/A	Terminal Capabilities
9F35	01	22	Terminal Type
FFFF8026	01	01	Transaction Type Check
5F36	01	02	Transaction Currency Exponent

Table 5-9. PayPass Application ID Default Tag Value

MaestroCard Application Identifier

Tag	Length	Value	Description
9F06	07	A0000000043060	AID MaestroCard
9C	01	00	Transaction Type
FFFF8002	01	01	Application Selection Identifier
FFFF8004	01	01	Disable Contactless Transaction Limit
FFFF8005	01	01	Zero allow
FFFF8007	01	01	Sign Unit Check
FFFF8009	01	01	CVM Require Limit Check Enable
FFFF8010	01	00	Extended Selection Support flag
FFFF8208	01	40	Transaction Info
5F57	00	N/A	Account Type
9F01	00	N/A	Acquirer Identifier
9F40	05	0000000000	Additional Terminal Capabilities
9F09	02	0002	App Version
DF8117	01	00	Card Data Input Capability
DF8118	01	60	CVM Capability-CVM Required
DF8119	01	08	CVM Capability-No CVM Required
DF811A	03	9F6A04	Default UDOL
DF8130	01	00	Hold Time Value
DF811B	01	A0	Kernel Configuration
DF810C	01	02	Kernekl ID
9F6D	02	0001	Mag-stripe Application Version Number
DF811E	01	10	Mag-stripe CVM Capability–CVM Required
DF812C	01	00	Mag-stripe CVM Capability–No CVM Required
9F15	02	0001	Merchant Category Code
DF812D	03	000000	Message Hold Time
9F7E	00	N/A	Mobile Support Indicator
DF8123	06	000000010000	Reader Contactless Floor Limit
DF8124	06	000000030000	Reader CTL (No On-device CVM)
DF8125	06	000000050000	Reader CTL (On-device CVM)
DF8126	06	000000030000	CVM Required Limit
DF811F	01	08	Security Capability
DF8120	05	0000000000	Terminal Action Code-Default

Tag	Length	Value	Description
DF8121	05	0000000000	Terminal Action Code-Denial
DF8122	05	0000000000	Terminal Action Code-Online
9F33	00	N/A	Terminal Capabilities
9F35	01	22	Terminal Type
FFFF8026	01	01	Transaction Type Check
5F36	01	02	Transaction Currency Exponent

Table 5-10. MaestroCard Application ID Default Tag Value

American Express Application Identifier

Tag	Length	Value	Description
9F06	06	A00000002501	AID American Express
9C	01	00	Transaction Type
FFFF8002	01	03	Application Selection Identifier
DF810C	01	04	Kernel ID
FFFF8004	01	01	Disable Contactless Transaction Limit
FFFF8005	01	01	Zero allow
FFFF8007	01	01	Sign Unit Check
FFFF8009	01	01	CVM Require Limit Check Enable
FFFF8010	01	00	Extended Selection Support flag
FFFF8208	01	40	Transaction Info
5F57	00	N/A	Account Type
9F01	00	N/A	Acquirer Identifier
9F40	05	0000000000	Additional Terminal Capabilities
9F09	02	0002	App Version
9F33	03	000888	Terminal Capabilities
9F35	01	22	Terminal Type
5F36	01	02	Transaction Currency Exponent
DF8123	06	000000010000	Reader Contactless Floor Limit
DF8124	06	000000030000	Reader CTL (No On-device CVM)
DF8125	06	000000050000	Reader CTL (On-device CVM)
DF8126	06	000000001000	CVM Required Limit

Table 5-11. American Express Application ID Default Tag Value

Discover Zip Application Identifier

Tag	Length	Value	Description
9F06	07	A0000003241010	AID Discover Zip
9C	01	00	Transaction Type
FFFF8002	01	01	Application Selection Identifier
DF810C	01	05	Kernel ID
FFFF8004	01	01	Disable Contactless Transaction Limit
FFFF8005	01	01	Zero allow
FFFF8007	01	01	Sign Unit Check
FFFF8009	01	01	CVM Require Limit Check Enable
FFFF8010	01	00	Extended Selection Support flag
FFFF8208	01	40	Transaction Info
5F57	00	N/A	Account Type
9F01	00	N/A	Acquirer Identifier
9F40	05	0000000000	Additional Terminal Capabilities
9F09	02	0002	App Version
9F33	00	N/A	Terminal Capabilities
9F35	01	22	Terminal Type
5F36	01	02	Transaction Currency Exponent
DF8123	06	000000010000	Reader Contactless Floor Limit
DF8124	06	000000030000	Reader CTL (No On-device CVM)
DF8125	06	000000050000	Reader CTL (On-device CVM)
DF8126	06	000000001000	CVM Required Limit

Table 5-12 Discover Zip Application ID Default Tag Value

Interac Application Identifier

Tag	Length	Value	Description
9F06	07	A0000002771010	AID Interac Flash
9C	01	00	Transaction Type
5F57	01	00	Account Type
9F01	06	000000000001	Acquirer Identifier
9F09	02	0002	App Version
9F33	03	000888	Terminal Capabilities
9F35	01	25	Terminal Type

Tag	Length	Value	Description
9F40	05	0000000000	Additional Terminal Capabilities
9F5E	02	E000	Terminal Option Status (TOS)
9F58	01	03	Merchant Type Indicator
9F59	03	D84000	Terminal transaction Information (TTI)
9F5A	01	00	Terminal transaction Type TTT
9F5D	06	000000005000	Terminal Contactless Receipt Required limit
DF2A	01	000000000600	Threshold Value for Biased Random Selection
DF2B	01	00	Maximum Target Percentage for Biased Random Selection
DF2C	01	63	Target Percentage for Random Selection
DF810C	01	06	Kernel ID
DF8120	05	0000000000	Terminal Action Code-Default
DF8121	05	0000000000	Terminal Action Code-Denial
DF8122	05	0000000000	Terminal Action Code-Online
DF8123	06	000000010000	Reader Contactless Floor Limit
DF8126	06	000000001000	CVM Required Limit
FFFF8002	01	01	Application Selection Identifier
FFFF8004	01	01	Disable Contactless Transaction Limit
FFFF8005	01	01	Zero allow
FFFF8007	01	01	Sign Unit Check
FFFF8009	01	01	CVM Require Limit Check Enable
FFFF8208	01	40	Transaction Info

Table 5-13. Interac Flash Application ID Default Tag Value

5.2. General Command

The general command of the reader is for daily transaction purpose. Depending on the reader set for which protocol, the command can be sent by using one of the following protocols: USI0, or USI2 (default).

USI0 – the simplest data format without the header, the trailer or the BCC. The reader relies on a 100ms timeout to determine that a command is received.

USI2 – with 01h (SOH), data length and BCC

Byte 0	Byte 1	Byte 2~3	Byte 4+n	Byte 5+n
01h	Address (not in use)	Command Len	Command/Data (n bytes)	BCC

Note: USI2 do not have the 100ms timeout delay.

5.2.1. (C8H) – Activate/Deactivate Contactless/MSR Reading command

This command can activate the reader to start to read the card. Please be noted that the value in required TLV parameters will impact the transaction result (Refer to Table 5-16)

Command Packet

Byte 0	Byte 1	Byte 2~3 (Optional)	Byte 4+n
C8	Interface Priority, 1 byte See Table 5-14	Time out (mS, Binary Format)	Data Object List in TLV (n bytes) See Table 5-16

Note: "Time Out" is an optional field in the C8 command. If "Time Out" was not set in C8 command, then reader will keep waiting for card tapping, until the host sends the <C8><00> command to cancel the transaction.

Interface Priority

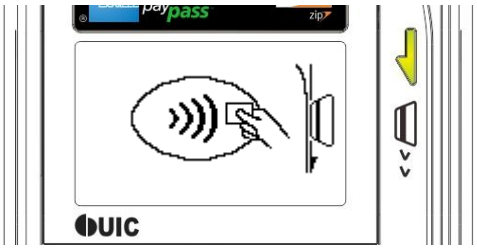
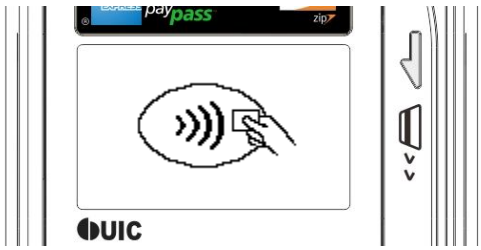
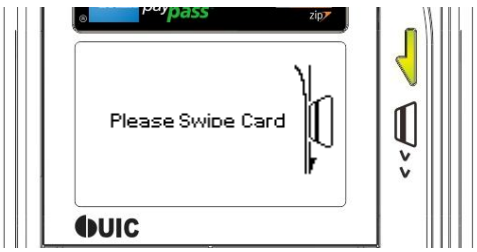
Parameter	Description	Note
00	Cancel	Disable all interface.
01	RFID & Magstripe	Enable both Magstripe and Contactless interface.*
02	RFID	Enable Contactless interface only.
03	Magstripe	Enable Magstripe interface only.

* The controller determines the priority per transaction.

Table 5-14. Interface Priority of Activate Contactless/MSR Reading Command

Default Display picture of Interface Priority

In EMV mode, the display will be empty in the idle mode. After the Host sends the Activate Contactless Reading command to Bezel5, the display will show the picture according to the parameter set in the Interface Priority field.

Parameter	Description	Display
01	RFID & Magstripe	
02	RFID	
03	Magstripe	

Note: The pictures in this table are reference only. For the real position of the signs please refer to the physical unit.

Table 5-15. Display picture reference of Interface Priority

Required TLV Parameters

Tag	Description	Length	Remarks
9F02	Amount, Authorized	6 Bytes	Mandatory
9C	Transaction Type	1 Byte	Options
9A	Transaction Date	3 Bytes	Options
9F21	Transaction Time	3 Bytes	Options
5F2A	Currency Code	2 Bytes	Options

Table 5-16. Required TLV Tags in Activate Contactless Reading Command

Note:

1. The reader will reject the command if the data is non-TLV format or with invalid coding.
2. For the unknown tags or tags with incorrect values, it will be ignored by the reader.
3. For the duplicate tags, the reader always overwrites the earlier tag value by the latter tag.
4. The reader accepts partial data update TLV data.
5. If "Transaction Date" and "Transaction Time" was not set in C8 command, the reader will proceed to the transaction by using RTC time (set by command "54" and "55").

5.2.2. (C9H) – Response of Start Transaction

This command is the return of the result to the Start Transaction command (C8h). The controller should send '^' to acknowledge upon receiving the data.

Command format

Field 1	Field 2	Field 3	Field 4
Byte 0 (1 byte)	Byte 1 (1 byte)	Byte 2 (1 byte)	Byte 3+n (Var. bytes)
C9h	Error Code Table 5-17	POS Entry Table 5-18	Card Data

Error Code

Value	Description	Contactless	Magstripe
00	Successful (If contactless transaction, the value means online request transaction)	v	v
01	Offline Approved	v	
02	Offline Declined	v	
03	Card not support	v	
04	Initiation error	v	v
05	Chip error (No AID)	v	
06	Empty candidate list	v	v
07	Time out	v	v
08	Card block	v	v
09	Application blocked	v	v
0A	Magstripe card data error	v	v
0B	Transaction error	v	v
0C	Authentication error	v	
0E	CVM Failed	v	v
10	Log full	v	v

Value	Description	Contactless	Magstripe
11	Card executing (Wait for the card (either contactless or contact) to be completely removed from reading area or card slot.)	v	v
12	Try Again	v	
20	2nd Tap	v	
33	Switch to Other Interface. (Contactless Used)	v	
37	Multiple Card	v	
39	Terminated. (Contactless Used)	v	
86	Empty candidate list, try other interface	v	
8C	Authentication error, try other interface	v	

Table 5-17. Error Code indication of Transaction Result

POS Entry

Bit 7	Bit 6	Bit 5	Bit 4 – 0	Description
0	0	0	00001	Contactless – qVSDC Card
0	0	0	00010	Contactless – MSD Card
0	0	0	00011	Contactless – PayPass Mchip Card
0	0	0	00100	Contactless – PayPass Magstripe Card
0	0	0	00101	Contactless – AMEX CPU Card
0	0	0	00110	Contactless – AMEX MSD Card
0	0	0	00111	Contactless – Discover DPAS Card
0	1	0	00001	Mag stripe card
1	0	0	00000	No payment card, no additional data available. *
0	1	0	00000	ISIS Sizzle

*Bit 7 = 0, the additional data is available. Bit 7 = 1, no additional data.

Table 5-18. POS Entry indication of Transaction Result

Card Data Scenario

Contactless card read successful: Error code = 00 (successful) and POS Entry = 02 or 03

Clear Data		Encrypted Data					Clear Data		
RFID-	|	TK1	|	TK2	|	TK3	|	DUKPT SN/Counter	|	SID

Fields Description

Value	Length (Byte)	Description
'|'	1	Field separator.
DUKPT SN/Counter	20	DUKPT Key serial number, DUKPT Key using for data encryption can be recognized by this serial number.
SID	16	Encrypted Session ID.

Table 5-19. Field Description of Contactless Transaction Data

Magstripe card read successful, Error code = 00 (successful) and POS Entry = 04

Clear Data		Encrypted Data						Clear Data		
CARD-	|	TK1	|	TK2	|	TK3	|	DUKPT SN/Counter	|	SID

Table 5-20. Field Description of MSR Transaction Data

5.2.3. (CEH) – Return the Specific EMV Tags

The reader will retrieve the data that list on the DOL after the EMV transaction is done.

Response

Response	Description
Result, 1 byte,	Only present if the result is failed
Tag result in TLV format	Only present if the result is successful

Return Data Format

Head	Tag 1			Tag 2			Tag 3~
Total Length	Tag	Length	Value	Tag	Length	Value	Tag

6. Authentication and Card Data Encryption ???

Question: Is this applicable to Bezel5 only? Does it apply to Bezel5?

The Bezel5 can be configured as a secure reader to protect the card holder's privacy. Once the Bezel5 enters into the secure reader mode, the output card data is encrypted. And the administration commands for changing the status or settings of the reader need to be authenticated.

6.1. Data Security and Key Management

The Bezel5 security arrangement involves a cryptography system for supporting end-to-end encryption.

1. Card Data Encryption:
   It uses the symmetric-key encryption, Triple DES (TDES)/or AES, with the Derived Unique Key Per Transaction (DUKPT) key management, as well as the RSA mode, to protect the card data.
2. Authentication for the administration command:
   All of the administration commands must be authenticated before their executions. A challenge-Response mechanism is involved in the process.
3. Google Wallet merchant data update:
   In order to simplify the merchant data update in the field site, the UIC680 will pre-load a TDES injection key. The current data of the merchant key and the merchant id are updated through the application which sends out the merchant data which is protected by the injection key.

Here is the brief summary:

1. The payment card data and the Google Wallet application data are going to be encrypted either by the TDES, AES or RSA mode.
2. The reader leverages the DUKPT for the key management scheme which is used for card data security.
3. The number of card readings is extended to two millions by arranging two key slots.
4. The encrypted output data of the USB HID report is in binary format.

The reader encryption is enabled as default by the factory. It requires an authentication if the user

wants to disable the encryption function. The RSA key length is 2048-bits. TDES will use the double length key and AES will use 128 bits if it is selected.

6.2. Product Life Cycle

The reader will go through several states in related to its data security operations:

State	Operator	Function Description	Security	Remark
Creation	UIC Production	Enable encryption, load initial key, load key management mode, enable command administration protection	No authentication	The key load in clear text.
Operation - Administration	System Integrators or Merchants	Update key, update key management mode, load unique serial number, enable/disable encryption	Authentication / Key is encrypted	
		Google Wallet merchant data update	Key is encrypted	
Operation – General Operation	Merchants	Reader is ready to operate	No Authentication/ Card data is encrypted	
Terminated	UIC RMA/ System Integrator	Key generation reaches to the end, needs to return to the system integrator to re-inject the key.	Authentication / Key is encrypted	

Table 6-1. Data Security Operations

6.3. Operation Flow

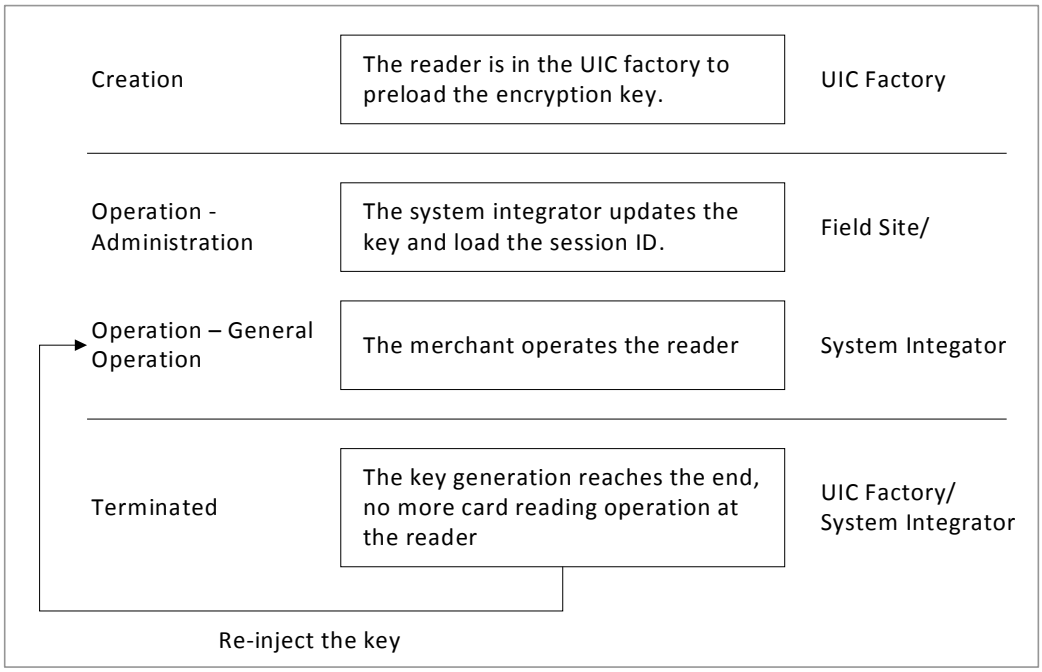


Figure 6-1. Data Security Operation Flow

6.4. Authentication

The command set of the Bezel5 is divided into several groups – the general operation, the administration and Google wallet merchant key update. The general operation commands, Examples like the version report, the encrypted track data query, the serial number query, are allowed to execute in the daily operation without the authentication process required. For the DUKPT key and the sensitive data updates, they belong to the administration commands and the authentication process is mandatory. The final command, Google Wallet merchant command, is protected by the pre-loaded injection key.

The Bezel5 adapts a simple version of entity authentication by using the public key cryptography. The reader pre-loads with the RSA public key. The authentication process uses RSA/SHA-1 to validate the incoming command. At the other end, the host must hold the private key for generating the correct data signature to attach to the command. The process is a kind of unilateral entity authentication, having the reader challenging the host application. No mutual authentication is necessary which reduces the complexity of authentication. In this way, the public key is considered as the part of the sensitive data and thus loaded and saved in the secure storage area of the reader as well.

Authentication involved entities:

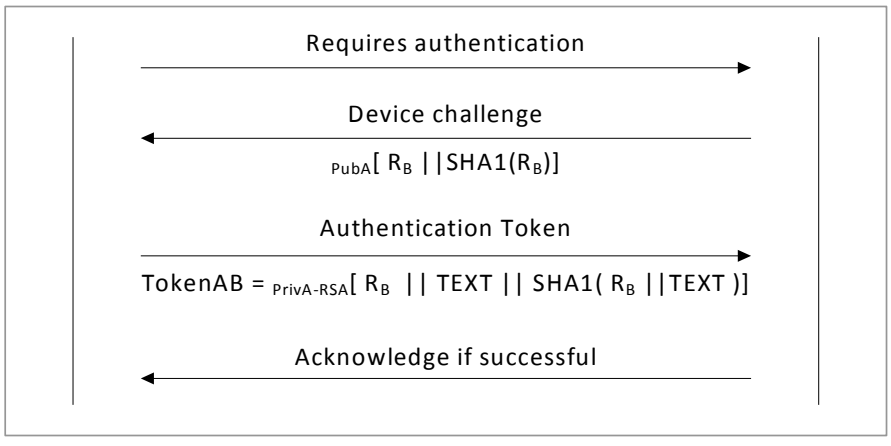
1. The claimant = the host application
2. The verifier = the reader
3. TEXT = the command/or key data to be sent from the host to the reader
4. PrivA = The RSA private key at the host application
5. PubA = The RSA public key at the reader
6. TDES/RSA […] = Data encrypted either by RSA or Triple DES

The authentication process proceeds as follows:

1. The claimant makes an authentication request to the verifier.
2. The verifier generates a random number challenge R_B (16 bytes data generated by the true random number generator).
3. The verifier encrypts R_B and its SHA1 value, $Pub_A[R_B || SHA1(R_B)]$ and sends to the claimant.
4. Upon receiving the challenge, the claimant decrypts R_B and validates SHA1 value. If SHA1 fails, the claimant terminates the process or requests R_B again.
5. The claimant creates an authentication token, TokenAB, by concatenating data and generating a digital signature:

   $$TokenAB = Priv_{A\text{-}RSA}[R_B || TEXT ||SHA1(R_B ||TEXT)]$$

6. The claimant sends the TokenAB to the verifier.
7. The verifier decrypts the TEXT and retrieves the SHA1 value by using the public RSA key, Pub_A.
8. The verifier executes the command data in [TEXT] if the signature is validated OK. The R_B must be used only once to enter the administration mode of the reader.



6.5. Double DUKPT

In order to support 2 million times of card reading, the Double DUKPT (D-DUKPT) solution is being used in the Bezel5 reader. There are two DUKPT key slots available inside the reader. They can be combined in different modes to fit the user application for achieving the 2 million times of operations. In this way it can extend the life cycle of the reader without the need to return the reader to the factory for key re-injection. The host application chooses either Triple DES or AES as the crypto engine to protect the card data.

Key Management Mode

Mode	Function Description
Auto rollover 1 (Factory Default)	Under this mode, the user only needs to load the initial key/ key serial number (KSN) to slot 1. The reader will duplicate the same key and KSN to slot 2. When the slot 1 key generation reaches the maximum 1M iterations, the reader will roll over the key management to slot 2 to continue the work. Note: Loading the key to slot 2 is prohibited. Note: the EC of KSN will start over when the reader switches the key management to slot 2.
Auto rollover 2	Under this mode, the user needs to load the initial key/key serial number (KSN) to slot 1 and slot 2 separately. The key management starts at slot 1. When the slot 1 key reaches the end, the reader will roll over the key management to slot 2 if the key is available. If slot 2 has no key, the reader enters the terminated state. Otherwise it continues to work at slot 2. The key and KSN can be different in both slots. Note: it is allowed to update the key to any slot regardless the slot is active or not. The simple rule is that the reader always chooses the lower number of the key slot for the key management if it is available. For Example, if the active key slot is 2 and slot 1 reaches the end, and then the user updates slot 1. For the next card swipe, the reader will choose slot 1 for the key management and leave slot 2 unchanged. Once the new key in slot 1 is running over, the reader will go back to slot 2 (assuming no key update in slot 2) to continue the work.
Traditional mode	Reserved and not in use.

Table 6-2. Key Management Mode

D-DUKPT Rules:

1. There are two DUKPT key management slots in the reader.
2. Each key slot has 3 different states:
   Empty: No key is loaded.
   Active: Key is loaded and is able to do the key management.
   Terminated: Passed 1M key iterations. No more function is allowed unless the key is re-injected.
3. No matter the reader is in what mode, it will always examine the key slot starting at the lower number (i.e. slot 1 then slot 2). If slot 1 is active with the key available, the reader will use the key for the data encryption. If slot 1 is inactive (empty or terminated) but slot 2 is active, the

reader will get the key from slot 2. If both key slots are inactive and the data encryption is enabled, the reader is in the terminated state then no data output is available.

6.5.1. *Auto Rollover 1: key generation*

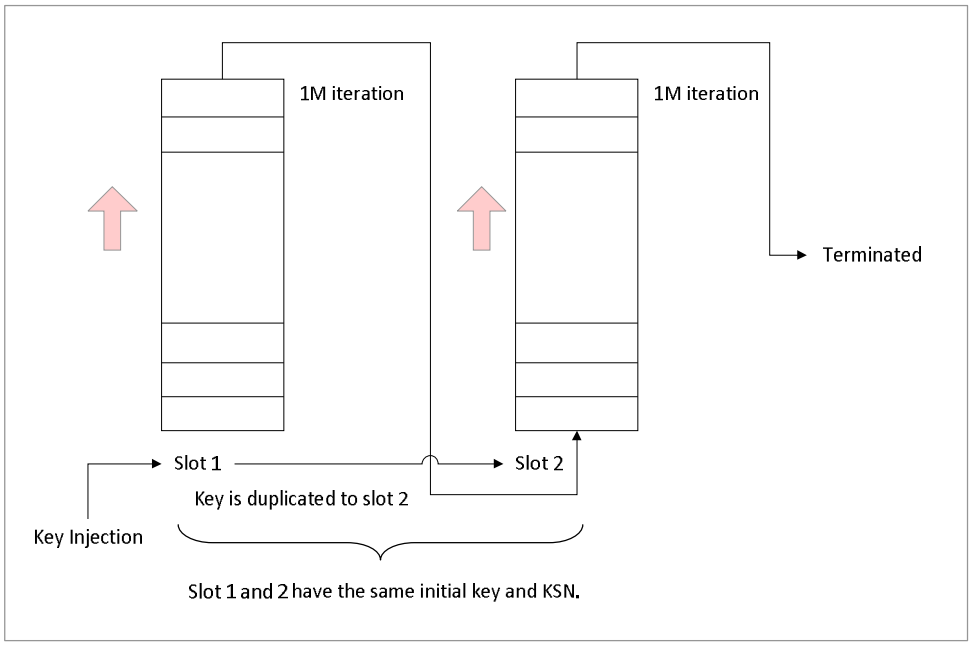


Figure 6-2. Auto Rollover 1: Key Generation

6.5.2. *Auto Rollover 2: key generation*

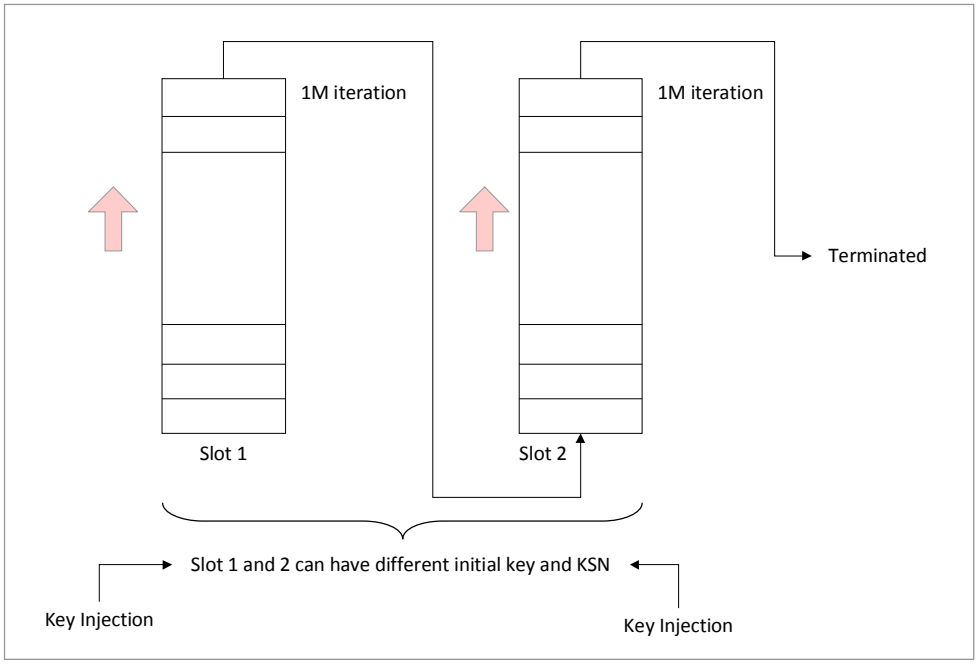


Figure 6-3. Auto Rollover 2: Key Generation

6.6. Track Output format (Self-Arm)

The encrypted data output varies according to the encryption mode and the interface type of the reader.

The card data can be encrypted with one of the following three modes:

1. DUKPT TDES
2. DUKPT AES
3. RSA

The following two interface groups will give totally different output formats of the encrypted card data:

1. RS232/USB Virtual COM
2. HID MSR

They are described in the following sections.

6.6.1. RS232/USB Virtual

DUKPT data output format

Encrypt Mode	| (7Ch)	Encrypt Tk1 Data	|	Encrypt Tk2 Data	|	Encrypt Tk3 Data	|	KSN	|	Encrypt Session ID	|

EXAMPLE

1|2B06FD66BF9896C0DD0207B8DFBA25745EC15069ECBE88E65738E6DADA2C7311859568DDDE08437775C8D703FA753BC9F57944404A2A5187E554D5B2FDB8E565**|**894C04A6BCF008448A6A6766A840502DABFC15308971492519C9E09C3F3EE839400432CDFE924CCB123731F078E29B58**|**84CF9EBE8E1C2CD04E5A72BCC63142C2300806552ED9C2EDED593EDC703EB00C39FC75DE6314F8A3C395A44A3B69F3B951838FA0479C7BA55438E1FE56DFF5A5DDC1C840547A4D430EE5B3DEFBF935AC**||**55494330303031000044**|**C9738D7244D6BC57**|**

RSA data output format

Encrypt Mode	| (7Ch)	Encrypted Tk1 Data	|	Encrypted Tk2 Data	|	Encrypted Tk3 Data	|

EXAMPLE

3|ZBJPirNzGDqKfOZ9Iar1MB12qQo+Sm4NJZZi5RVyEZ30O0qh1sc4Cq+IctcOQdNLNLlJEAx9bM/O59dV60v4upM5V7hpe
ROaQQCu1H2OQfvz/lEwqgpWEWySTqjWg/w/hh8c2yqT8ZZrWvJjddx0+tJCRpmtLZbmaYdFH7AvCdUQobEpamNnltg1vb
XoZ2OXYAIt9Wps3E6M4ogsoI0wDI87TYXgUXDl3Onz1Tjz+dnXYtoet/lKp4++n8B8MEm/U4wHfmxgjzNwl/HM/ZxhSH5IwL
rOJh/vhTN/SyWZ4jqr/+qUxTLmcITyOtKli3X6+2m5443p5nyQ6GTwsWbrxg==|vZ/M4tMk+1SY8Lyjy7UGzVBO0rOTYZZDY
WMMnnS+c8cEO0kHbnz26j3wbyWkI/rZoq07VTeviCenqFU79QFs/s8ZLGRugWLa6jzJW4rqjB4UT2zlQBtQPtMZ2hVjBi+M
XYg3T9Wwfaj5aa0COTMMhrPHrDLSIYW3CYHN/dPRo8i187HIIwAzmsw+N0cG7gaMz89jgwOpPIehA0E8m1YurqOe0ScdY
wl3zWJH8KMf+3AjKnmRfSAHFjDRvWHkKVLJTwC8jDj/dITyuSCHbD3RpBpq9hBmp9d5VI+0V7YpUyQpakHqzMcix04FCnZ
malaG6ToNjqvY1WVP6QfjQp8hFw==|htdHYRIKrppRiYyS8Lk0s1BG10hpUNMjwAcFxjIGKU9Wb+6a4av+R79NvUiUACNK
RHP3hvwiGzlqpdw1jS7jokroIF8Vjvgn78CxNwN+QRgmII+d4ke0adzY4X39br7iPCqnQau6uZWV/ubokBbvGTjytxLQCgxzG3l
X9iZwqfPQTF504Gn+y6npic/Y5xp/zfWuXoLR2v0SPZMmbNap1givc5W3d93r6lJehmRbWJCJoW16n4dDMoZwiKSE7K8gK
QAsQB1FcNRzFYfHWU4EvpJnBBZ48myzwgreQrhYt2BI8JAGlYRIUNpSOu43TPlwTAFZbadT22fEvDqJU4fv4g==|

6.6.2. HID MSR (Optional)

Offset	Usage Name	Description
0	Track 1 status	00h→decode error, 01h→decode ok
1	Track 2 status	00h→decode error, 01h→decode ok
2	Track 3 status	00h→decode error, 01h→decode ok
4-5	Track 1 encrypted data length	Track 1 encrypted data length
6-7	Track 2 encrypted data length	Track 2 encrypted data length
8-9	Track 3 encrypted data length	Track 3 encrypted data length
12	Card type	See note *.
13	Encrypt Mode	0:Not Encrypt 1:DUKPT TDES 2:DUKPT AES 3: RSA
14-23	DUKPT serial number/counter	DUKPT serial number and counter
24-31	Device serial number	Serial number for device
32-47	Encrypted Session ID	Session ID use DUKPT to encrypt
48-303	Track 1 encrypted data	Original TK1 data use the crypto algorithm to encrypt.
304-559	Track 2 encrypted data	Original TK2 data use the crypto algorithm to encrypt.
560-1071	Track 3 encrypted data	Original TK3 data use the crypto algorithm to encrypt.

* Note (offset 12) Card type

Table 6-3. HID MSR Offset Table

Low Nibble	0 ISO/ABA:ISO/ABA encode format for all contactless payments	
High Nibble	Bit 7, 6, 5 – payment instrument status	000 – No payment solution is available or its traditional magistrate card data
		001 – Track of Google payment MID available in track 1 and 2
		010 – PayPass Magstripe & Mchip available in track 1, 2 and 3
		011 – Visa MSD & qVSDC available in track 1, 2 and 3
		100 – Amex available in track 1 and 2
		101 – Discover ZIP available in track 1 and 2
		111 – Other contactless payment solution (reserved for future)

Example
0000: Google Wallet not available and no other contactless payment instrument available.
0001: Google Wallet available only but neither Google payment MID nor other contactless payment available.
0100: Google Wallet not available but Contactless PayPass available.
0011: Google Wallet is available and the data of Google payment MID is available in track 1 and 2.

6.7. Administration Commands

6.7.1. 90H 02H – Load Session ID

This command is used to load 8 bytes Session ID to device.

Command Pocket

Byte 0~1	Byte 2~9	Byte 10~17
Command	Encrypted Random	Encrypted Session ID
90h 02h	Issue 90h 03h command to get random	For creating new keys

6.7.2. 90H 03H – Get KSN & Encrypted Random

This command is used to get the DUKPT Key Serial Number and encrypted random number.

Command Pocket

Byte 0~1
Command, 90h 03h

Response

Response	Byte 0~9	Byte 10~17
Success	DUKPT serial number and counter	Random number in encrypted format
Bad Parameters	*	

Example

Host Command	Reader Response Example
<90><03>	
	<55><49><43><30><30><30><31><00><00><05><FF><AE><E7><96><F3><02><15><2D>

6.7.3. 90H 04H – Select DUKPT Key Slot

This command is used to select DUKPT encrypt key slot of device.

Command Pocket

Byte 0~1	Byte 2
Command, 90h 04h	Data

Command Data

Data	Description
01h	Select key slot 1
02h	Select key slot 2

Response Code

Response	Meaning
^	Acknowledgement
*	Cannot execute (e.g. out of range)
!	Bad parameter (e.g. incorrect length)

6.7.4. 90H 05H – Select DUKPT Management Mode

This command is used to select DUKPT Management Mode of device.

Command Pocket

Byte 0~1	Byte 2
Command, 90h 05h	Data

Command Data

Data	Description
01h	Select Mode 1
02h	Select Mode 2

Response Code

Response	Meaning
^	Acknowledgement
*	Cannot execute (e.g. out of range)
!	Bad parameter (e.g. incorrect length)

6.7.5. 90H 06H – DUKPT Key Iteration Test

This message is designed to do DUKPT key iteration test. The reader will return 71 assuming a PIN of '1234' and pack the data in ANSI X9.8 PIN block format. This command can be used to verify the key being loaded properly or not.

Command Pocket

Byte 0~1	Byte 2~11	Byte 12	Byte 13	Byte 14~17
Command, 90h 04h	Account	FS	DC Ind	Amount

Parameters

Field	Description
Account	Primary account number, "1234567890" (31H 32H 33H 34H 35H 36H 37H 38H 39H 30H) – **Fixed data, don't change**
FS	<1C>, field separator
DC Ind	'D' (44H) – **Fixed data, don't change**
Amount	"4567" (34H 35H 36H 37H) – **Fixed data, don't change**

Response element

Field	Length	Value and description
71	2	Message ID
Reserved	1	Always '0'
Key Serial#	10..20	Key Serial number used in encrypting PIN. Included only when PIN is entered. Format: hexadecimal string.
[PIN]	16	Encrypted PIN block. Format: hexadecimal string.

Error codes

Code	Meaning
'0'	Null Account input field.
'2'	Account number shorter than 8 digits.
'3'	Account number longer than 19 digits.
'4'	Account number have character other than '0'-'9'.
'5'	[D/C Ind] field not exist or format error.
'6'	Timeout value error.
'8'	Amount string format error.
'A'	No DUKPT key injected.

Code	Meaning
'B'	Flash read/write error.
'C'	Memory buffer allocation error.
'F'	DUKPT operation limit (1 million) reached, program stop.

Example

Parameter	
Initial Key	"554E49464F524D5F44454641554C5421"
Account number	"1234567890"

Usage	
Send command	"<90><06>1234567890<1C>D4567"
Gets PIN Block	"710554943303030310003E32FF2D3C47BF9F87E"
Find out current key by using of initial key and serial number/counter. (DUKPT scheme)	Current Key = "A51DD67C06A54E7F0ADA776534532772"
Decrypt PIN Block by using the current key	"2FF2D3C47BF9F87E"
Clear PIN Block	"041234FEDCBA9876"
Take max. 12 bytes of account number from the next-to-last digit, then pad zeroes on the left to match the length of 16 bytes	"0000000123456789"
XOR the clear PIN Block with account number	"041234FFFFFFFFFF"
"04" means 4 bytes data follows	PIN = "1234"

6.7.6. 90H 07H – Get Encrypted Status

This command is used to Get Encrypted Status of device.

Command Pocket

Byte 0~1
Command, 90h 07h

Response

Response	Byte 0~5
Success	Encrypt Mode (1 byte) + DUKPT Key Slot (1 byte) + DUKPT Management Mode (1 byte) + DUKPT Key Slot 1 Status (1 byte) + DUKPT Key Slot 2 Status (1 byte)
Bad Parameters	*

Response element

Encrypt Mode

Code	Meaning
00	Not Encrypted
01	DUKPT Mode
02	AES Mode
03	RSA Mode

DUKPT Key Slot

Code	Meaning
00	Key Slot 1
01	Key Slot 2

DUKPT Management Mode

Code	Meaning
00	Mode 1
01	Mode 2

DUKPT Key Slot 1 Status

Code	Meaning
00	DUKPT Key Empty
01	DUKPT Key Active
02	DUKPT Key Terminated

DUKPT Key Slot 2 Status

Code	Meaning
00	DUKPT Key Empty
01	DUKPT Key Active
02	DUKPT Key Terminated

6.7.7. 90H 10H – Get Challenge

This command is used to get challenge from the reader.

Command Pocket

Byte 0~1
Command, 90h 10h

RSA Encrypted Response

Response	Byte n	Byte 0~15 +n	Byte 16~35 +n
Success	Padding Data See Table 6-4	Random	SHA1 (Padding + Random + Exp Len + Exp + Modules Length + Modules)
	The return length is upon RSA key length.		
Bad Parameters	*		

Padding Frame

Byte 0~1	Byte 2+n	Byte 3+n
00h 02h	Var.	00h

Table 6-4. Get Challenge Padding Frame

6.7.8. 90H 11H – Load Encrypt Initial Key

This command is used to load initial key to device.

Command Pocket

Byte 0~1	Data
Command, 90h 11h	Var. Bytes

Data Format

Data Byte	Field Name	Length	Notes
n	Padding Data	Var.	Padding frame see Table 6-5
16+n	Random	16 Bytes	Issue 90h 10h command to get random.
17+n	Encrypt Mode	1 Byte	See Table 6-6
18+n	DUKPT Key Slot	1 Byte	See Table 6-7

Data Byte	Field Name	Length	Notes
19~20 +n	Key Length	2 Bytes	2 bytes in binary format
21~28 +n	Key Data	8~16 Bytes	Initial DUKPT Key, must be 8 or 16 bytes
29~48 +n	SHA1	20 Bytes	Padding + Random + Encrypt Mode + Key Slot + Key Length + Key Data

Padding Frame

Byte 0~1	Byte 2+n	Byte 3+n
00h 01h	Var.	00h

Table 6-5. Load Initial Key Padding Frame

Encrypt Mode

Code	Meaning
01	DUKPT Mode
02	Google Wallet merchant symmetry key

Table 6-6. Encrypt Mode of Load Initial Key

DUKPT Key Slot

Code	Meaning
01	DUKPT Key Slot 1
02	DUKPT Key Slot 2

Table 6-7. DUKPT Key Slot of Load Initial Key

6.7.9. 90H 12H – Change Encrypt Mode for Data Output Format

This command is used to change the encryption mode of data output format for the device.

Command Pocket

Byte 0~1	Data
Command, 90h 12h	Var. Bytes

Data Format

Data Byte	Field Name	Length	Notes
n	Padding Data	Var.	Padding frame see Table 6-8
16+n	Random	16 Bytes	Issue 90h 10h command to get random.

Data Byte	Field Name	Length	Notes
17+n	Encrypt Mode	1 Byte	See Table 6-9
18~37+n	SHA1	20 Bytes	Padding + Random + Encrypt Mode

Padding Frame

Byte 0~1	Byte 2+n	Byte 3+n
00h 01h	Var.	00h

Table 6-8. Padding Frame of Change Encrypt Mode for Data Output Format

Encrypt Mode

Code	Meaning
30h	None Encrypted Mode
31h	DUKPT TDES Mode
32h	DUKPT AES Mode
33h	RSA Mode

Table 6-9. Encrypt Mode of Data Output Format

6.8. Load Session ID

USAGE:

1. Issue the 90h 03h command to get encrypted random number.
2. Decrypt “Encrypted Random” using the current key and gets “Random”.
3. Generate an encryption key by XORing current key and “Random”.
4. Use encryption key to encrypt [(Random) + (New Session ID)].

Put the result into the 90h 02h command packet.

EXAMPLE

Parameter	
Initial Key	“554E49464F524D5F44454641554C5421”
New Session ID	“0102030405060708”
Usage	
get encrypted random number from the 90h 03h command	"554943303030310003E103EA68415833B363 "
Find out the current key by using the initial key and the serial number/counter. (DUKPT scheme)	Current Key : "13B9BB5C4136505FB9B1335223CC9291"
Random number	“03EA68415833B363 "
using the current key to gets Clear Random	Clear Random : "495AF134649D561E"
Generate an encryption key by XORing the current key and the clear Random	Encryption Key: "5AE34A6825AB0641F0EBC2664751C48F"
Use the encryption key to encrypt (TDES) [(Random)+(New Session ID)]	Encrypted data: "82CEEAF1C6502358686C954C121E65E6"
Put the result into the 90h 02h command packet	90h 02h 82h CEh EAh F1h C6h 50h 23h 58h 68h 6Ch 95h 4Ch 12h 1Eh 65h E6h

Table 6-10. Example of Load Session ID

6.9. Load DUKPT Key

USAGE:

1. Issue the 90h 10h command to get the encrypted challenge data.
2. Decrypt the “Encrypted Challenge” by using the “RSA Private Key” and then get the “Random”.
3. Use the command format data to get the SHA 1(20 bytes).
4. Use the “RSA Private Key” to encrypt the command format data.
5. Put the result into the 90h 11h command packet.

EXAMPLE

Parameter	
New KSN	55494330303031000000
New Initial Key	“554E49464F524D5F44454641554C5421”
Exp	“00010001”
RSA Private Key	“7AD86A3E9BEBCE15EAE06EAC8CEAFF119E8584B0A24AADDDF6827A2ED46AA9D78FC7B9CE262CAF5CC17BFA3DF074C9E7B79577BDF530784DB3EB57CD455CA2BA5F9CDDA5B38380C89B1136BE1A1BE82DE9A4ABA2CBC6F0E8F75208EF1B77AA7D4FC7A8642A0C268DC6A012B908F3D8A646246F70236FAACE67FCF638E75E7EBFAD71D52405EAC4F04D9530BDAC54D97BB37C9BF229D2F18F140AB071BC7C144F9255947A5C55DFF8B1A465621E64447CA6AA5D50876F2B22CCEC68EF629AAE7AD78CA9D3D3BC1A72E92FDDACFFC4A347240ED9FDF245AE0D3545D2249553DD5A4758D58A44E642736B60E6D5B4C2A940C194F4109F458C9D2636535EC63A82A1”
Modules	“BDEA7BE96D7CC049C6D68EFB7AFF404810C23AC88866E744C5E27016E415D3787F57EF835B84A5AAC8D550C99E3C2955472525A7AB40C5190CF42351AD41386BD8238A5474637332BFF35A7B7CF1C3173FA424F466DD574C23B84ED9B748D7350F26BF17D5014EDAEDAD5917991427C5D3859D16312DEEE2E2A5B2287856CDB435B8B5D3E7C68E70B31398EEC34C45524EB54DD4153438A3BE50D4EEA7BD54E088873C173F5023AD18FD4AAC9068DA786A9A5AD7462683CEDA8B862AC3CC19F7715AD37A19E1A9C2AC9169D58283674041B66D7A2E69D4920E45D4B75AA745DFC0C1C654FDD47E526298F86DEB0DC80BBB6DC26793C48BDC8D99CB31FFEA42BD”
Usage	
Get Challenge Data from 90h 10h command	"115BBAC1EFF299152081B7C81858820D8A13D6630CB0F322D20110F2FFDF1ED759411AB528998CCA983ED0503D306B7E17E08B0B2A196983BE52E95BF2554F690F3FC08C50ADBE94E115BAA44ECBFD9C9E55FA6749981782340B749794F08ACF92271240E8A0EE382863C29F7455CDAABC9F666D3270580465F2522C6A83FE0EA52D06C48AD93CE9DC803FFF38DD4EB33966E8618B39C9D580F9F6313A67884C1DC42203EE78CAC9A0029FD41E8F55A54C4AD65BDDEE605A15C309B608307CC7C86940ADC65BFC260718D236C3DC1D887BFA655EA7D22EDF6453BDA365CB81F64B5477C53BCCB4792D2C8B09D3C72AB5A55982E5B8369E49853514E3A6A9EBFF"
Using the RSA Private key to decrypt 90h	"7AD86A3E9BEBCE15EAE06EAC8CEAFF119E8584B0A24AADDDF6827A2ED46AA9D78FC7B9CE262CAF5CC17BFA3DF074C9E7B79577BDF530784DB3EB57CD455CA2BA5F9CDDA5B38380C89B1136BE1A1BE82DE9A4ABA2CBC6F0E8F75208EF1B77AA7D4FC7A8642A0C268DC6

10h command result	A012B908F3D8A646246F70236FAACE67FCF638E75E7EBFAD71D52405EAC4F04D9530BDA C54D97BB37C9BF229D2F18F140AB071BC7C144F9255947A5C55DFF8B1A465621E64447C A6AA5D50876F2B22CCEC68EF629AAE7AD78CA9D3D3BC1A72E92FDDACFFC4A347240ED9 FDF245AE0D3545D2249553DD5A4758D58A44E642736B60E6D5B4C2A940C194F4109F458 C9D2636535EC63A82A1"
gets clear format	“000213128446848055107333536843043063130225127201117370049836628096318451 1345200232425145528526184503136810365209537777444951615531096409926153192 2782409A52271033378951605115421705215522657720984452132328205851634102206 26135747447445225318556615053620101236A5256349552436400251106038551298032 5691462212205272480213494445154801158639191173674445039854305894846751965 1414232611163199405454121077815089534523380341246349153232583010410797517 2007E7DF70595A4D1571AEC6AC83F4580A97F1EDC26824B9B6AC828DEB6A5A88F738D61 D10C”
Random data	“7E7DF70595A4D1571AEC6AC83F4580A9”
Complement the data length	“0001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF007E7DF70595A4D1571 AEC6AC83F4580A901010010554E49464F524D5F44454641554C542155494330303031000 000”
get SHA 1	“6170126C75D798FAD28E93FCEBC9A5AC24956BA6”
Combine data	“0001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF007E7DF70595A4D1571 AEC6AC83F4580A901010010554E49464F524D5F44454641554C542155494330303031000 0006170126C75D798FAD28E93FCEBC9A5AC24956BA6”
Using the RSA Private key to encrypt	“AFD7EA2D13FBDBC74DEAB0C427A95612FC38CAAC0F89ABF8F61B20139A82285C97B321 742EE19524A7A2066819F32DCD2AD46CFDE70680896E095238858377BDB7651AA467779 18728E38F0238D6522CEB2D0096A9FD734A08C4C7736EB63E7B5DFBD3D9E738369C90FE 6981923D3C38285EA2ECEADAD9B880C65EAFF1B05775F6B3E2B44BB1C695AEF5DE086FD 1FABAFCE3CDB80F8DE6FA2701B42998B7F3D568DF0248190898CA6A83A2755B0100FC9B C8F2F73C9E5A399A39B0A76DDB0BE342B605EFC4F687E97244AC60C33337F684765E4C95 8C1F4AC636DDEE9820E3E84BB4E24DD47AB5FE95747A9355F4AB09E72147FAE4E18BD12 968FE69037E9CAA”
Put the result into the 90h 11h command packet	“<90><11>AFD7EA2D13FBDBC74DEAB0C427A95612FC38CAAC0F89ABF8F61B20139A8228 5C97B321742EE19524A7A2066819F32DCD2AD46CFDE70680896E095238858377BDB7651 AA46777918728E38F0238D6522CEB2D0096A9FD734A08C4C7736EB63E7B5DFBD3D9E738 369C90FE6981923D3C38285EA2ECEADAD9B880C65EAFF1B05775F6B3E2B44BB1C695AEF 5DE086FD1FABAFCE3CDB80F8DE6FA2701B42998B7F3D568DF0248190898CA6A83A2755B 0100FC9BC8F2F73C9E5A399A39B0A76DDB0BE342B605EFC4F687E97244AC60C33337F684 765E4C958C1F4AC636DDEE9820E3E84BB4E24DD47AB5FE95747A9355F4AB09E72147FAE4 E18BD12968FE69037E9CAA”

Table 6-11. Example of Load DUKPT Key

6.10. Load Google Wallet Merchant Symmetry Key

USAGE:

1. Issue the 90h 10h command to get the encrypted challenge data.
2. Decrypt the “Encrypted Challenge” using the the “RSA Private Key” and then get the “Random”.
3. Use the command format data to get the SHA 1(20 bytes).
4. Use the “RSA Private Key” to encrypt the command format data.
5. Put the result into the 90h 11h command packet.

EXAMPLE

Parameter	
New Symmetry Key	112233445566778899AABBCCDDEEFF11
Exponent	“00010001”
RSA Private Key	“7AD86A3E9BEBCE15EAE06EAC8CEAFF119E8584B0A24AADDDF6827A2ED46AA9D78FC7B9CE262CAF5CC17BFA3DF074C9E7B79577BDF530784DB3EB57CD455CA2BA5F9CDDA5B38380C89B1136BE1A1BE82DE9A4ABA2CBC6F0E8F75208EF1B77AA7D4FC7A8642A0C268DC6A012B908F3D8A646246F70236FAACE67FCF638E75E7EBFAD71D52405EAC4F04D9530BDAC54D97BB37C9BF229D2F18F140AB071BC7C144F9255947A5C55DFF8B1A465621E64447CA6AA5D50876F2B22CCEC68EF629AAE7AD78CA9D3D3BC1A72E92FDDACFFC4A347240ED9FDF245AE0D3545D2249553DD5A4758D58A44E642736B60E6D5B4C2A940C194F4109F458C9D2636535EC63A82A1”
Modules	“BDEA7BE96D7CC049C6D68EFB7AFF404810C23AC88866E744C5E27016E415D3787F57EF835B84A5AAC8D550C99E3C2955472525A7AB40C5190CF42351AD41386BD8238A5474637332BFF35A7B7CF1C3173FA424F466DD574C23B84ED9B748D7350F26BF17D5014EDAEDAD5917991427C5D3859D16312DEEE2E2A5B2287856CDB435B8B5D3E7C68E70B31398EEC34C45524EB54DD4153438A3BE50D4EEA7BD54E088873C173F5023AD18FD4AAC9068DA786A9A5AD7462683CEDA8B862AC3CC19F7715AD37A19E1A9C2AC9169D58283674041B66D7A2E69D4920E45D4B75AA745DFC0C1C654FDD47E526298F86DEB0DC80BBB6DC26793C48BDC8D99CB31FFEA42BD”
Usage	
Use 90h 10h command to get the encrypted challenge data	"553B9A174DFE7B9D23C3C888AF16F658AAED690BC32900F99647A2E41A2B206D050DB02032A789EEB577C65DE1B8EDA36DEE1ECC13E55E4BB315EF2F2A4B3648B29D975C9516C180A14A939FF05AB648D3795E8957E47C1BDD2704350B3D9F463A3D27D5BA3E4F738420C1AB2A36F49CD7DF3CD8A9B3628C07BA8E64D58AE0762420E27A574B0D7412B26770682963ECB1BB28F73717C650EF13AEFD969A383C9A8D9B5586B943D9A09EFCFAE75C2B90D64C4B7B336AF1D2D796E2D2EF3107EE0DC64586B88AD3A5397496A3B1B06B820D10AD2866BD8BEE0DFFFA566CFCBAEE96F2EF264C76F4959F0B06C11BC65B40CD24E0D4F7780C6AA16D05F9C3228C12"
Using the RSA Private key to decrypt challenge data then gets clear	“00022112159426448623353271879254069460501139399393913239544505253854624137953645633003543040633304148145328094650843214206818526505452254323194372987535730243527681408210576453834025294324914553401003421337519483235237

format	5246048350230716256119815505945234153193597310555761037509143415560395999 1042127044404248709333974621853650201647513489037542275934139063120931881 3020127526814474630130205503555125012019698125060363012279079380506308573 9007284558CF716CA567844661192AD23C99DEE8B5255D018714140BFE60D122467ECF17 5A7”
Random data	“7284558CF716CA567844661192AD23C9”
Complement the data length[7]	“0001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0 07284558CF716CA567844661192AD23C90201001011223344556677889 9AABBCCDDEEFF 111”
Get SHA 1	“5AE7E8F57264369B47FC6E06703712EEECC11FF”
Combine data	“0001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0 07284558CF716CA567844661192AD23C90201001011223344556677889 9AABBCCDDEEFF 1115AE7E8F57264369B47FC6E06703712EEECC11FF”
Using the RSA Private key to encrypt the combine data	“312AD2AD1849FE8ACC2A769D4BE22F0FAD504C85D9BD98C3C1B9E27F2055727709E16C DB6F37F009032BBE33E34D475DB8FB1EA2C94995A44A144E3E91A740065FD5FAECEB134 1A92896A1C087B038D2360F048A4F6EAF1647C19C4DEBF2F58770B3B8D5651C5E3FDED5 C3857135D3B405AD0E77DA24DED419F88D3CB37A177F24C3440B033D61331CD3813C86 A102970464842DD7ED261A76E023F73BEEB48AC69763F6EFABAFC32416796E62169DB453 7038EAE3C9ADD13141763B20193392D812151B8F24183E8DB31CEA2D30DAF0CB9600A5 B395744E946A0AEF94FD7497F628EB6D145DF023E8A349CDA1F5C790C84B58D4E1D1A44 F1525696BB2C37E01”
Put the result into 90h 11h command packet	“<90><11>312AD2AD1849FE8ACC2A769D4BE22F0FAD504C85D9BD98C3C1B9E27F205572 7709E16CDB6F37F009032BBE33E34D475DB8FB1EA2C94995A44A144E3E91A740065FD5F AECEB1341A92896A1C087B038D2360F048A4F6EAF1647C19C4DEBF2F58770B3B8D5651C 5E3FDED5C3857135D3B405AD0E77DA24DED419F88D3CB37A177F24C3440B033D61331C D3813C86A102970464842DD7ED261A76E023F73BEEB48AC69763F6EFABAFC32416796E62 169DB4537038EAE3C9ADD13141763B20193392D812151B8F24183E8DB31CEA2D30DAF0 CB9600A5B395744E946A0AEF94FD7497F628EB6D145DF023E8A349CDA1F5C790C84B58D 4E1D1A44F1525696BB2C37E01”

Table 6-12. Example of Load Google Wallet Merchant Symmetry Key

7 If Data Length < RSA Private Key Length, then
complement data length = RSA Private Key Length – sha1(20 bytes) – 2 bytes (head and end)
If Data Length > RSA Private Key Length, then
complement data length = Multiple (RSA Private Key Length) – sha1(20 bytes) – 2*n bytes (head and End)；n= qty' of page

6.11. Load Authentication RSA Key

USAGE:

1. Issue the 90h 10h command to get the encrypted challenge data.
2. Decrypt the “Encrypted Challenge” using the “RSA Private Key” and then get the “Random”.
3. Use the command format data to get the SHA 1(20 bytes).
4. Use the “RSA Private Key” to encrypt the command format data.
5. Put the result into the “I2” command packet.

EXAMPLE

Parameter	
New Modules (First byte should be greater then “6A”)	“C827FF33BD1C24C6A2919F8B182975F1399697F460514B2B67BB7822DB9A4D11457F0A10EE420011D96A42F91BA42D1DEDB5EA4B6B7A32E3EBE67574211E68D78FAB65994A6D9AD3343CDCC5C28F0E46AE391054811EE4B1D11DE4EAB6EF9EAF79750F049DA24678D835C06587A9101B0AE1344D71D5D58E469F7FE352AD61A587924F47A8E5EECD9911440E9C09CF2625CD34CB9B4907A19C7EEFE3DC460759AEDBDC902174D2A8F5D21E35B690EEFB756E6C1A88D0B8B9D243C1C0785617FC21B8D4B441F3341B00566A05AEFE31D3277EF8E3B0A7E8660C9C7278E9418DB5BF2924B50FB84CEE4E9A03250DBA83FD3B9245F0727FAFCF85C71B9ED87BE01B”
New Exponent	“00010001”
Exponent	“00010001”
RSA Private Key	“7AD86A3E9BEBCE15EAE06EAC8CEAFF119E8584B0A24AADDDF6827A2ED46AA9D78FC7B9CE262CAF5CC17BFA3DF074C9E7B79577BDF530784DB3EB57CD455CA2BA5F9CDDA5B38380C89B1136BE1A1BE82DE9A4ABA2CBC6F0E8F75208EF1B77AA7D4FC7A8642A0C268DC6A012B908F3D8A646246F70236FAACE67FCF638E75E7EBFAD71D52405EAC4F04D9530BDAC54D97BB37C9BF229D2F18F140AB071BC7C144F9255947A5C55DFF8B1A465621E64447CA6AA5D50876F2B22CCEC68EF629AAE7AD78CA9D3D3BC1A72E92FDDACFFC4A347240ED9FDF245AE0D3545D2249553DD5A4758D58A44E642736B60E6D5B4C2A940C194F4109F458C9D2636535EC63A82A1”
Modules	“BDEA7BE96D7CC049C6D68EFB7AFF404810C23AC88866E744C5E27016E415D3787F57EF835B84A5AAC8D550C99E3C2955472525A7AB40C5190CF42351AD41386BD8238A5474637332BFF35A7B7CF1C3173FA424F466DD574C23B84ED9B748D7350F26BF17D5014EDAEDAD5917991427C5D3859D16312DEEE2E2A5B2287856CDB435B8B5D3E7C68E70B31398EEC34C45524EB54DD4153438A3BE50D4EEA7BD54E088873C173F5023AD18FD4AAC9068DA786A9A5AD7462683CEDA8B862AC3CC19F7715AD37A19E1A9C2AC9169D58283674041B66D7A2E69D4920E45D4B75AA745DFC0C1C654FDD47E526298F86DEB0DC80BBB6DC26793C48BDC8D99CB31FFEA42BD”
Usage	
Use 90h 10h command to get the encrypted challenge data	"A08B1810E85D8D5B9DD8E324A6D204DD2E6C3ED6DA2706EEE461469567DD9B3EA9053F60CB48168922161E640340C782FEF919B5BF1B293EDD2F0C5B7449543134877B150FACF558ACFCF7719473DAA20C7E389B17C3159D3DBDF6CEFD3CA15652EB916D8B8252077AFB32CF5416D12FC79F06E8AB9ED2834CE6CE5AD98018BC5C62A4074389004B04AE4BC7FEF027F87694F45912DD238A6043FD6AFA38F6F9CD2E307FB3186C784F3D3C093BB665FF41

	C923CB0FEF00EC2A9BE648E952AFB7502733D11EBA0CE4D3642EDAC412F58773AA793C9784BACD3068A8CF76CC65175001AE879E78F437D67051CE604839C3BA4177407B353FEB0887A3FFC2A47940"
Using the RSA Private key to decrypt challenge data then gets clear format	“0002905032825913855082218433915032704170403503617877258702104585355653434642821230554179571228398206205 6A51036851078355538126206100847095152811160037930080451625357213530957558147293393101182408247185240470195245305292294510361970931605044744353414A50813018473203320205440432634915120754405435481020789980776144018183134441927 28A5519015035423445145959930802455804102802102167805213430040332245874763306210614450553892004126320754546845624 0046C66E3F4D4AA86FFF55132FBB71C845179D53D3B8BC9210DA8CAA1319E835A12FC78114”
Random data	“46C66E3F4D4AA86FFF55132FBB71C845”
Complement the data length[8]	“0001FFF0046C66E3F4D4AA86FFF55132FBB71C84504000100010100C827FF33BD1C24C6A2919F8B182975F1399697F460514B2B67BB7822DB9A4D11457F0A10EE420011D96A42F91BA42D1DEDB5EA4B6B7A32E3EBE67574211E68D78FAB65994A6D9AD3343CDCC5C28F0E46AE391054811EE4B1D11DE4EAB6EF9EAF79750F049DA24678D835C06587A9101B0AE1344D71D5D58E469F7FE352AD61A587924F47A8E5EECD9911440E9C09CF2625CD34CB9B4907A19C7EEFE3DC460759AEDBDC902174D2A8F5D21E35B690EEFB756E6C1A88D0B8B9D243C1C0785617FC21B8D4B441F3341B00566A05AEFE31D3277EF8E3B0A7E8660C9C7278E9418DB5BF2924B50FB84CEE4E9A03250DBA83FD3B9245F0727FAFCF85C71B9ED87BE01B”
Get SHA 1	“9A2A410325F5ECE9A251A3316B449E7DF649BDD2”
Combine data	“**0001FF00**46C66E3F4D4AA86FFF55132FBB71C84504000100010100C827FF33BD1C24C6A2919F8B182975F1399697F460514B2B67BB7822DB9A4D11457F0A10EE420011D96A42F91BA42D1DEDB5EA4B6B7A32E3EBE67574211E68D78FAB65994A6D9AD3343CDCC5C28F0E46AE391054811EE4B1D11DE4EAB6EF9EAF79750F049DA24678D835C06587A9101B0AE1344D71D5D58E469F7FE352AD61A587924F47A8E5EECD9911440E9C09CF2625CD34CB9B4907A19C7EEFE3DC460759AEDBDC902174D2A8F5D21E35B690EEFB756E6C1A88D0B8B9D243C1C0785617FC21B8D4B441F3341B00566A05AEFE31D3277EF8E3B0A7E8660C9C7278E9418DB5BF2924B50FB84CEE4E9A03250DBA83FD3B9245F0727FAFCF85C71B9ED87BE01B9A2A410325F5ECE9A251A3316B449E7DF649BDD2”

8 If Data Length < RSA Private Key Length, then
complement data length = RSA Private Key Length – sha1(20 bytes) – 2 bytes (head and end)
If Data Length > RSA Private Key Length, then
complement data length = Multiple (RSA Private Key Length) – sha1(20 bytes) – 2*n bytes (head and End)；n= qty' of page

Base on Private Key Length – 2 bytes, add 2 bytes of Head (6A) and End (BC) to each package.	“**6A**0001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFF0046C66E3F4D4AA86FFF55132FBB71C84504000100010100C827FF 33BD1C24C6A2919F8B182975F1399697F46051**BC”** “**6A**4B2B67BB7822DB9A4D11457F0A10EE420011D96A42F91BA42D1DEDB5EA4B6B7A32E 3EBE67574211E68D78FAB65994A6D9AD3343CDCC5C28F0E46AE391054811EE4B1D11DE4 EAB6EF9EAF79750F049DA24678D835C06587A9101B0AE1344D71D5D58E469F7FE352AD6 1A587924F47A8E5EECD9911440E9C09CF2625CD34CB9B4907A19C7EEFE3DC460759AEDB DC902174D2A8F5D21E35B690EEFB756E6C1A88D0B8B9D243C1C0785617FC21B8D4B441F 3341B00566A05AEFE31D3277EF8E3B0A7E8660C9C7278E9418DB5BF2924B50FB84CEE4E9 A03250DBA83FD3B9245F0727FAFCF85C71B9ED87BE01B9A2A410325F5ECE9A251A3316B 449E7DF649BDD2**BC”**
Using the RSA Private key to encrypt the combine data	“60B09C88578CC3E8C299F87FECDA9FD95D1314CDBB4C65254F9FEB3C368F0259BB6BDD 50F33EAFEB6A73E4D96630268A5AF0AFC3E9DB696B882FD175144C6DE997FD0B4966DED 0A0B43F866FF4BF0AC368D25CE032652AE29F72B2B3D3459CB36AC9B26B1922C0B7CEBC 8643E18A0EDAF0F0D1652FC6D21DD518483FDA29F81125713BF0221436CB6071E1E3185 08E91D8B827D100652AFCAB47C84963351E7C8DEB41DD4D1B278C1A964C20A3DE07E6F 6B1394981C2FD910FD53EB8B084FAC5D0B4F82E716A06A933DF80E7B49F62A4CA11408A 5CDBF874C8A877CA03BDC13914CF7D01018F424624242F48E7427D1693AA3098999482E 9A9A0C49139D8A361” “777D160D8F040298DAD7911CD81C9113961B6358240F6D83025537E6BE5AA42EDFBD3C 5BEE250136FB90C5B3B58B1D4AE088197B34B15152E252A721E5FD89C629DB8A8DA564E 59B7611C0F8F1DCEE344197C34ED5EA5033516F7E740ECEF8E50B4C10EE3FD3591E4806D 1F1F4367CA05FFD2684EA7325C64E82F01C2A6285221F3C6F4507B0135F5DC978C363B5F AE61A4817067ACE52774817C028AA4EE293AC7B584A3524D50A49AA2B94BA2F7D3B16F6 B016A89A3A6A453705464E74BF1541DC0CC51B49F051891C6DE0280A435A043C7DAD4A 45EF8B4E5D4983208EAA19DDDC4C3932E1B6511EE8A743F830FCAC4AF692BE5EB9AF8A8 80AC4E9A0137396”
Put the result into I2 command packet	I260B09C88578CC3E8C299F87FECDA9FD95D1314CDBB4C65254F9FEB3C368F0259BB6BD D50F33EAFEB6A73E4D96630268A5AF0AFC3E9DB696B882FD175144C6DE997FD0B4966DE D0A0B43F866FF4BF0AC368D25CE032652AE29F72B2B3D3459CB36AC9B26B1922C0B7CEB C8643E18A0EDAF0F0D1652FC6D21DD518483FDA29F81125713BF0221436CB6071E1E318 508E91D8B827D100652AFCAB47C84963351E7C8DEB41DD4D1B278C1A964C20A3DE07E6 F6B1394981C2FD910FD53EB8B084FAC5D0B4F82E716A06A933DF80E7B49F62A4CA11408 A5CDBF874C8A877CA03BDC13914CF7D01018F424624242F48E7427D1693AA3098999482 E9A9A0C49139D8A361777D160D8F040298DAD7911CD81C9113961B6358240F6D830255 37E6BE5AA42EDFBD3C5BEE250136FB90C5B3B58B1D4AE088197B34B15152E252A721E5F D89C629DB8A8DA564E59B7611C0F8F1DCEE344197C34ED5EA5033516F7E740ECEF8E50B 4C10EE3FD3591E4806D1F1F4367CA05FFD2684EA7325C64E82F01C2A6285221F3C6F4507 B0135F5DC978C363B5FAE61A4817067ACE52774817C028AA4EE293AC7B584A3524D50A4 9AA2B94BA2F7D3B16F6B016A89A3A6A453705464E74BF1541DC0CC51B49F051891C6DE0 280A435A043C7DAD4A45EF8B4E5D4983208EAA19DDDC4C3932E1B6511EE8A743F830FC AC4AF692BE5EB9AF8A880AC4E9A0137396”

Table 6-13. Example of Load Authentication RSA Key

6.12. Change Encrypt Mode for Data Output Format

USAGE

1. Issue the 90h 10h command to get the encrypted challenge data.
2. Decrypt the “Encrypted Challenge” using the the “RSA Private Key” and then get the “Random”.
3. Use the command format data to get the SHA 1 (20 bytes).
4. Use the “RSA Private Key” to encrypt the command format data.
5. Put the result into the 90h 12h command packet.

EXAMPLE

Parameter	
Select Encrypt Mode	31 (DUKPT TDES Mode)
Exponent	“00010001”
RSA Private Key	“7AD86A3E9BEBCE15EAE06EAC8CEAFF119E8584B0A24AADDDF6827A2ED46AA9D78FC7B9CE262CAF5CC17BFA3DF074C9E7B79577BDF530784DB3EB57CD455CA2BA5F9CDDA5B38380C89B1136BE1A1BE82DE9A4ABA2CBC6F0E8F75208EF1B77AA7D4FC7A8642A0C268DC6A012B908F3D8A646246F70236FAACE67FCF638E75E7EBFAD71D52405EAC4F04D9530BDAC54D97BB37C9BF229D2F18F140AB071BC7C144F9255947A5C55DFF8B1A465621E64447CA6AA5D50876F2B22CCEC68EF629AAE7AD78CA9D3D3BC1A72E92FDDACFFC4A347240ED9FDF245AE0D3545D2249553DD5A4758D58A44E642736B60E6D5B4C2A940C194F4109F458C9D2636535EC63A82A1”
Modules	“BDEA7BE96D7CC049C6D68EFB7AFF404810C23AC88866E744C5E27016E415D3787F57EF835B84A5AAC8D550C99E3C2955472525A7AB40C5190CF42351AD41386BD8238A5474637332BFF35A7B7CF1C3173FA424F466DD574C23B84ED9B748D7350F26BF17D5014EDAEDAD5917991427C5D3859D16312DEEE2E2A5B2287856CDB435B8B5D3E7C68E70B31398EEC34C45524EB54DD4153438A3BE50D4EEA7BD54E088873C173F5023AD18FD4AAC9068DA786A9A5AD7462683CEDA8B862AC3CC19F7715AD37A19E1A9C2AC9169D58283674041B66D7A2E69D4920E45D4B75AA745DFC0C1C654FDD47E526298F86DEB0DC80BBB6DC26793C48BDC8D99CB31FFEA42BD”
Usage	
Use 90h 10h command to get the encrypted challenge data	"4151FDFB7D5D0C6546E980D0BBE220A5703F36CD3A6A7307EA9303BE7CECF781973F37737CD5A439EB3DE0D687A8E5B38B3450D211E62B1EDABA5A9A81B89D8280B3C6E2C6A97B2B619C5CE762E6556B33F7C0F181FBE769C272E20CDF1696D40B856B019678D20CD3BE8F6A5979DB7E6AB26AEBF4FAFE09B2C2D28B5C846BC74E33372023D2C249BC24AD9D113DA9E1C5B56880074C2891BA037BB137EFE1BAF3CF5E96841B966E374ADAAE076BFA48ACC3375A155C1502959434FA58B8B4CB59D98CD749384CB10F789BB39A8B2989C3C1B7FEF3EC85E42479C0511A1EB328FBA05E70CCE4D9119454E575E2809280D48142FF2E86DC177F2084C6A2D30809"
Using the RSA Private key to decrypt challenge	“00023506633308938685050275474304591191241992954489054526951544580433474432704727410351051512156471649550512537 17A59321516113509277384655112543158

data then gets clear format	2261382065027843430440595310358901384310502624329203409175009185065590105 7752232055503031414823180337450550305385945111913229101984324536531003011 5167151413692235302595123933765030701823657750310331305823732689042341455 1257281483427422206143305936656063250116625831111935366452330928410225178 5000591231B27E3E125BE4B88B68D451A2EEB5A5734ECB97987B0C26873D84F13CA52267 898”
Random data	“0591231B27E3E125BE4B88B68D451A2E”
Complement the data length[9]	“0001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000591231B27E3E125BE4B88B68D451A2E31”
Get SHA 1	“5AE7E8F57264369B47FC6E06703712EEECC11FF”
Combine data	“0001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000591231B27E3E125BE4B88B68D451A2E3182 CD8E4AACB8966CE3FA3D8A579A93B049DABC67”
Using the RSA Private key to encrypt the combine data	“905EB9A2240901536FC4CE8EDEE3F8551D7157238DDE2600733E4AEB54C7E993989BA4E 92F8A1ADBF71FEF288E714E353AD5490234B2672AF9791C213B766744855F1A416B61AF0 CB40517BA3A85D06C51E834AB129EE1C0E551555C5BAB828D8A354F80DC2953785F65C3 55202FA6FD0A7F72288AF171084F305EEAA4CF6C2D99B25780790D0BFCB9CAC6897BA3D 6B21B1B8229CEA4F49F4A0113A3D98208B549EE965D7BC706BC0731662921E53EB6BF58 CEC4D30C6CA5ED79937483EB1FF45F5C157BAAF6E1269200081304458734C6B4B397A868 C35BDA3F7E715F02B5F6015EC12EDDA09FFD407EB3698A324E20D31715FE0817E2DDD0F 540DB097932CE”
Put the result into 90h 12h command packet	“<90><12>905EB9A2240901536FC4CE8EDEE3F8551D7157238DDE2600733E4AEB54C7E99 3989BA4E92F8A1ADBF71FEF288E714E353AD5490234B2672AF9791C213B766744855F1A4 16B61AF0CB40517BA3A85D06C51E834AB129EE1C0E551555C5BAB828D8A354F80DC2953 785F65C355202FA6FD0A7F72288AF171084F305EEAA4CF6C2D99B25780790D0BFCB9CAC6 897BA3D6B21B1B8229CEA4F49F4A0113A3D98208B549EE965D7BC706BC0731662921E53 EB6BF58CEC4D30C6CA5ED79937483EB1FF45F5C157BAAF6E1269200081304458734C6B4B 397A868C35BDA3F7E715F02B5F6015EC12EDDA09FFD407EB3698A324E20D31715FE0817 E2DDD0F540DB097932CE”

Table 6-14 Example of Change Encrypt Mode for Data Output Format

9 If Data Length < RSA Private Key Length, then

complement data length = RSA Private Key Length – sha1(20 bytes) – 2 bytes (head and end)

If Data Length > RSA Private Key Length, then

complement data length = Multiple (RSA Private Key Length) – sha1(20 bytes) – 2*n bytes (head and End)；n= qty’ of page

7. Google Wallet

Google Wallet is an application utilizing the mobile phone as the payment instrument. It provides several services in one wallet including payments, offers and rewards. The general working scenario is just to have the user to tap the phone at the **Bezel5** reader. By passing all necessary information to the host application, the user can purchase the goods to earn the loyalty points, get the discount price or pay by the gift card.

There are two different phases proposed by Google Wallet - Legacy and Next Generation. The current available service is Legacy which is fully supported by the Bezel5 reader.

7.1. Track Output Scenarios

The Legacy service is composed of payment (compliant with EMV contactless specification) and value-added services over MIFARE. Bezel5 reads all services in one tap and outputs the card data in the following scenarios.

CASE 1:

- Card contains active payment instrument in PPSE and active payment instrument in Mifare.
- VAID containing payment MID is present in PPSE.
  Output: Tag FFFF820E. Track 1 and 2 contain Google Wallet payment MID track data if any. There is no track 3 data.

CASE 2:

- Card contains active payment instrument in PPSE and no payment instrument in Mifare.
- VAID containing payment MID is not present in PPSE.
  Output: Track 1 and 2 (or 3) with contactless payment data, and Tag FFFF820E if any.

CASE 3:

- Card contains no active payment instrument in PPSE and an active payment instrument in Mifare.
- VAID containing payment MID is present in PPSE
  Output: Tag FFFF820E. Track 1 and 2 contain Google Wallet payment MID track data if any. There is no track 3 data.

CASE 4:

- Card contains no active payment instrument in PPSE and no active payment instrument in Mifare.
- VAID containing payment MID is not present in PPSE
  Output: Tag FFFF820E, if any.

Track 1~3, Tag FFFF820E Information

Track #	Description	SS/ES	Data
1	PayPass/payWave/Amex/Discover ZIP/Google Wallet Payment	%/?	Emulate magnetic stripe track 1: PAN, Card holder name, Expiration Date, Track 1 Discretionary Data.
2	PayPass/payWave/Amex/Discover ZIP /Google Wallet Payment	;/?	Emulate magnetic stripe track 2: PAN, Expiration Date, and Track 2 Discretionary Data.
3	PayPass/payWave	+/?	UIC proprietary data output for extra contactless payment information.
Tag	**Description**	**SS/ES**	**Data**
FFFF820E	Google Wallet applets	$/?	UIC proprietary data output for Google Wallet.

Table 7-1. Track/Tag information of Google Wallet Transaction Format

7.2. Configuration Option

In order to integrate the Google Wallet with other payment schemes, **Bezel5** provides the following selectable configurations.

Mode	Description
Google application deactivated	Google application (and Mifare functionality) is deactivated
Mifare First (default)	Google application (and Mifare functionality) is activated and Mifare is read first
Select PPSE First	Google application (and Mifare functionality) is activated and Select PPSE is done first

Table 7-2. Selectable Configuration of Google Wallet transaction mode

Card Data Output for Different Types of Card and Reader Configurations

With the reader running in the Self-Arm mode, depending on the configuration set in the reader and the type of card to be read, the reader will output different types of card information. The following table lists out the summary of it:

Type of Card	Reader Configuration				
	Google Wallet Support			Mifare Card Support	
	Disabled	PPSE First	Mifare First	Disabled (MFxy = 10)	Enabled (MFxy = 11)
Payment Card	Track data	Track data	Track data	Track data	Track data
Google Wallet with payment instrument	Track data	Google data	Google data	Google data	Google data
Google Wallet without payment instrument	Track data	Track data /w Google data	Track data /w Google data	Track data /w Google data	Track data /w Google data
Mifare Standard 1K	N/A	N/A	N/A	N/A	“M2”
Mifare Standard 4K	N/A	N/A	N/A	N/A	“M3”
Mifare Ultralight	N/A	N/A	N/A	N/A	“M1”
Mifare Ultralight C	N/A	N/A	N/A	N/A	“M1”
Mifare DESFire	N/A	N/A	N/A	N/A	“M4”
Mifare Plus	N/A	N/A	N/A	N/A	“M5”

Table 7-3. Card Data Output mode for different types of card and reader configurations

7.3. Tag FFFF820E Data Format

Tag FFFF820E data format is reserved for Google Wallet application only (Tag FFFF820E data format is being used for Google Wallet application). This data can be represented in ASCII-HEX values. If the data is in ASCII, it is embraced by […]. The data begins with start sentinel "$" and ends with end sentinel "?".

Google Wallet data is output in Tag FFFF820E according to the following format:

Byte 0	Byte 1~2	Byte 3	Byte 4~5	Byte 6
[$], Start sentinel (1-byte)	Total length of track (2-bytes)	Num of applications (1-byte)	CB	02

Byte 7~8	Byte 9	Byte 10~12	Byte 12+n
App 1 Schema-code (2-byte)	App 1 tag (1-byte)	Length of all records for App1 (1-3 byte)	Data of records for App 1 (var bytes)

Byte n								
CB	02	App 2 Schema-code (2-byte)	App 2 tag (1-byte)	Length of all records for App1 (1-3 byte)	Data of records for App 2 (var bytes)	…	CB	02

Byte n				
App n Schema- code (2-byte)	App n tag (1-byte)	Length of all records for App n (1-3 byte)	Data of records for App n (var bytes)	[?], End sentinel (1-byte)

Tag FFFF820E Data Object Format:

Data/Tag	Description	Data Object Format (Bytes)
$	Start Sentinel	1 byte
0023	Total Length of track	2 bytes
03	Number of application	1 byte
CB02	[CB][02]	2 bytes
1002	App 1 Schema-code	2 bytes
C5	App 1 tag	1 byte
05	Length of all records for App 1	1~3 byte
6502530000	Data of all records for App 1	var bytes
CB02	[CB][02]	2 bytes

Data/Tag	Description	Data Object Format (Bytes)
2002	App 2 Schema-code	2 bytes
C5	App 2 tag	1 byte
04	Length of all records for App 2	1~3 byte
12340001	Data of all records for App 2	var bytes
CB02	[CB][02]	2 bytes
4002	App 3 Schema-code	2 bytes
D1	App 3 tag	1 byte
07	Length of all records for App 3	1~3 byte
12FF34FF056789	Data of all records for App 3	var bytes
?	End Sentinel	1 byte

Table 7-4. Google Wallet Data Transmission Tag Format

Google Wallet Sample Data Format

$[Total Length][Number of application][CB][02][App1 Schema-code][App1 tag][length of record for App1][Data of records for App1] [CB][02][App2 Schema-code][App2 tag][length of record for App2][Data of records for App2]…[CB][02][App n Schema-code][App n tag][length of record for App n][Data of records for App n]?

Example (in Hex String)

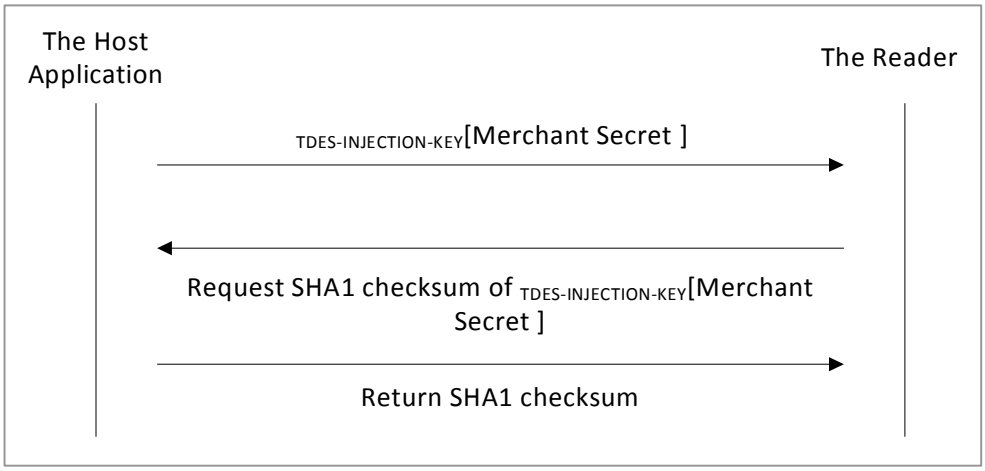
$002303CB021002C5056502530000CB022002C50412340001CB024002D10712FF34FF056789?

Important note: If the Google Wallet contains incorrect data such as CRC error at one application, the reader will ignore (discard) the related application. Then the reader will continue to read the next application in the wallet. There is no output data if the CRC of all applications are erroneous.

The data sequence for multiple bytes value is in big-endian. For Example, [01 02] = 1 x 256 + 2 = 258.

7.4. Google Wallet Merchant Key Update

The Google Wallet merchant key is protected by a pre-loaded injection key by using TDES crypto algorithm. Since the key is always encrypted, the authentication is not required. The reader will use the SHA1 value for verification purpose.



The reader only accepts the merchant key to follow the Google MIFARE key packet format. In general, Google will generate the new key which is protected by the merchant defined symmetry key (TDES key). The normal procedure is that the encrypted data is sent to the merchant. Once the merchant application receives the data, it will convert the data to the suitable coding format according to the reader's command to be sent to the reader. Then the reader will use the pre-loaded symmetry key to decrypt the merchant key and install it.

The symmetry key is able to be updated after the unit is deployed to the field site. The details of the symmetry key update process are not described in this section. Please refer to the **Bezel5** command authentication section for more information.

In the meantime, the reader will perform SHA1 on the whole encrypted merchant key data block and store the value in the nonvolatile memory. The application can verify the checksum to assure the key is installed properly.

The format of the packaged merchant key is

<len_enc_X>, the length of enc_X, (1 byte)	<enc_X>, the encrypted list of merchant secrets, (n bytes)

7.5. Google Wallet Commands

7.5.1. D (44H) – Google Card Operation

The command is used to operate Google card.

Command Packet

Byte 0	Byte 1	Byte 2+n
D (44h)	Command Type	Data, (Optional)

Command Type

Command	Description
03h	Read transmission log
04h	Clear transmission log
07h	Load Google wallet MIFARE secret key
08h	Get SHA1 value of MIFARE key
09h	Get Google Polling Mode
0Ah	Get SHA1 value of All Encrypt MIFARE key

Table 7-5. Google Wallet Operation Command Type

D<03> (44H 03H) - Read transmission log

Example

Host Command	Reader Response Example	Comment
The Google Wallet card on top of the card reader		Do not remove
H0		Self-Arm Disable, see 4.1.15.
	^	Reader ACK
O		Antenna power On, see 4.4.2.
	^	Reader ACK
D<03>		Read transmission log

Host Command	Reader Response Example	Comment
	<FC><E3><02><01><30><30><30><30><30><30><30><30><F0><12><01><00><5A><F1><02><01><00><30><F0><31><00><00><00><97><F0><12><01><00><5A><F1><02><01><00><30><F0><31><00><00><00><97><F0><12><01><00>	Transmission log of the card (40 bytes)
OR		
D<03>		Read transmission log
	* (2Ah)	Card not detected

D<04> (44H 04H) - Clear transmission log

Example

Host Command	Reader Response Example	Comment
The Google Wallet card on top of the card reader		Do not remove
H0		Self-Arm Disable, see 4.1.15.
	^	Reader ACK
O		Antenna power On, see 4.4.2.
	^	Reader ACK
D<03>		Read transmission log
	<FC><E3><02><01><30><30><30><30><30><30><30><30><F0><12><01><00><5A><F1><02><01><00><30><F0><31><00><00><00><97><F0><12><01><00><5A><F1><02><01><00><30><F0><31><00><00><00><97><F0><12><01><00>	Transmission log of the card (48 bytes)
D<04>		Clear transmission log
	^ (5Eh)	Successful
D<03>		Read transmission log
	<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>	Clear (48 bytes)
OR		
D<04>		Clear transmission log
	* (2Ah)	Card not detected

D<07> (44H 07H) – Load Google wallet MIFARE secret key

Command Packet

Byte 0-1	Byte 2	Byte 3~
Command	Total Encrypt length	Encrypt data
D<07>	<38>	3FAF3B31B3DDDBA6964DF8BBB81A3F828BEF5FAEED91B5B2856E44E4E0C30ED930502694CD20EE81E43B0846FDC8DD7924B7A04BA6248C5E

Example Keys

Symmetry Key		
	11223344556677889AABBCCDDEEFF11	
Secret Key		MID
1	20E103626A70A92B3AD3FDE04429C3B6	0000
2	642DDA067A4A1725C6F3B22F51E607EE	FFFE
3	7A244D16EAB80D7CBB5329E0653A09CD	F010

Load MIFARE Key

Get clear text data	36||20E103626A70A92B3AD3FDE04429C3B60000642DDA067A4A1725C6F3B22F51E607EEFFFE7A244D16EAB80D7CBB5329E0653A09CDF010||80
Encrypt the clear text data by the symmetry Key doing TDES Encrypt Clear Data	3FAF3B31B3DDDBA6964DF8BBB81A3F828BEF5FAEED91B5B2856E44E4E0C30ED930502694CD20EE81E43B0846FDC8DD7924B7A04BA6248C5E Total Encrypt Data Length = 38h
Send the command to the reader	D<07>383FAF3B31B3DDDBA6964DF8BBB81A3F828BEF5FAEED91B5B2856E44E4E0C30ED930502694CD20EE81E43B0846FDC8DD7924B7A04BA6248C5E

*[Encrypt Data]: $_{\text{Symmetry Key encrypted}}$ [Total Clear Key Length (1 byte) + Key1 (32 bytes) + MID1 (2 bytes) + Key2 (32 bytes) + MID2 (2 bytes) + Key3 (32 bytes) + MID 3 (2 bytes) + padding (80h 00h 00h....)]

Example

Host Command	Reader Response Example
D<07> 383FAF3B31B3DDDBA6964DF8BBB81A3F828BEF5FAEED91B5B2856E44E4E0C30ED930502694CD20EE81E43B0846FDC8DD7924B7A04BA6248C5E	
	^ (5Eh) Successful

D<08> (44H 08H) –Get SHA1 value of MIFARE key

Command Packet

Byte 0-1	Byte 2
Command	Key Index
D<08>	1 (31h) ~ 8 (38h)

Example Keys

Symmetry Key		
	112233445566778899AABBCCDDEEFF11	
Secret Key		MID
1	20E103626A70A92B3AD3FDE04429C3B6	0000
2	642DDA067A4A1725C6F3B22F51E607EE	FFFE
3	7A244D16EAB80D7CBB5329E0653A09CD	F010

Load MIFARE Key

Get clear text data	36||20E103626A70A92B3AD3FDE04429C3B60000642DDA067A4A1725C6F3B22F51E607EEFFFE7A244D16EAB80D7CBB5329E0653A09CDF010||80
Encrypt the clear text data by the symmetry Key doing TDES Encrypt Clear Data	3FAF3B31B3DDDBA6964DF8BBB81A3F828BEF5FAEED91B5B2856E44E4E0C30ED930502694CD20EE81E43B0846FDC8DD7924B7A04BA6248C5E Total Encrypt Data Length = 38h
Send the command to the reader	D<07>383FAF3B31B3DDDBA6964DF8BBB81A3F828BEF5FAEED91B5B2856E44E4E0C30ED930502694CD20EE81E43B0846FDC8DD7924B7A04BA6248C5E

*[Encrypt Data]: $_{\text{Symmetry Key encrypted}}$ [Total Clear Key Length (1 byte) + Key1 (32 bytes) + MID1 (2 bytes) + Key2 (32 bytes) + MID2 (2 bytes) + Key3 (32 bytes) + MID 3 (2 bytes) + padding (80h 00h 00h....)]

Get MIFARE Key SHA1 value

Send the command string to the reader	D<08H>1
Return SHA1	<12><C7><B4><D7><5A><6E><5C><A1><3A><17><41><72><3F><A6><3F><0E><6C><30><E9><B2>
Verify the SHA1 value	Key1 = 20e103626a70a92b3ad3fde04429c3b6 Symmetry Key = 112233445566778899AABBCCDDEEFF11
	Encrypt Data = AF0F842C5E9DE3C5983943B326264075
	[Encrypt Data] SHA1 = 12C7B4D75A6E5CA13A1741723FA63F0E6C30E9B2

Example

Host Command	Reader Response Example	Comment
D<08>1		
	<12><C7><B4><D7><5A><6E><5C><A1><3A><17><41><72><3F><A6><3F><0E><6C><30><E9><B2>	SHA1 (20 bytes)
	<FF><FF><FF><FF><FF><FF><FF><FF><FF><FF><FF><FF><FF><FF><FF><FF><FF><FF><FF><FF>	Empty (20 bytes)

D<09> (44H 09H) –Get Google Polling Mode

Example

Host Command	Reader Response Example	Comment
D<09>		
	0 (30h)	Google Polling Disable
	1 (31h)	PPSE First
	2 (32h)	MIFARE First
	* (2Ah)	Bad parameters

D<0A> (44H 0AH) –Get SHA1 value of All Encrypt MIFARE key

Command Packet

Byte 0-1
Command
D<0A>

Example Keys

Symmetry Key		
	112233445566778899AABBCCDDEEFF11	
Secret Key		MID
1	20E103626A70A92B3AD3FDE04429C3B6	0000
2	642DDA067A4A1725C6F3B22F51E607EE	FFFE
3	7A244D16EAB80D7CBB5329E0653A09CD	F010

Load MIFARE Key

Get clear text data	36||20E103626A70A92B3AD3FDE04429C3B60000642DDA067A4A1725C6F3B22F51E607EEFFFE7A244D16EAB80D7CBB5329E0653A09CDF010||80
Encrypt the clear text data by the symmetry Key doing TDES Encrypt Clear Data	3FAF3B31B3DDDBA6964DF8BBB81A3F828BEF5FAEED91B5B2856E44E4E0C30ED930502694CD20EE81E43B0846FDC8DD7924B7A04BA6248C5E Total Encrypt Data Length = 38h
Send the command to the reader	D<07>383FAF3B31B3DDDBA6964DF8BBB81A3F828BEF5FAEED91B5B2856E44E4E0C30ED930502694CD20EE81E43B0846FDC8DD7924B7A04BA6248C5E

*[Encrypt Data]: $_{\text{Symmetry Key encrypted}}$ [Total Clear Key Length (1 byte) + Key1 (32 bytes) + MID1 (2 bytes) + Key2 (32 bytes) + MID2 (2 bytes) + Key3 (32 bytes) + MID 3 (2 bytes) + padding (80h 00h 00h....)]

Get MIFARE Key SHA1 value

Send the command string to the reader	D<0AH>
Encrypt the clear text data by the symmetry Key doing SHA1 Value	383FAF3B31B3DDDBA6964DF8BBB81A3F828BEF5FAEED91B5B2856E44E4E0C30ED930502694CD20EE81E43B0846FDC8DD7924B7A04BA6248C5E
Return SHA1	BB8EB10C7521C547E6D74643D5559A6C1080ABF9

Example

Host Command	Reader Response Example	Comment
D<0A>		
	BB8EB10C7521C547E6D74643D5559A6C1080ABF9	SHA1 (40 bytes)

8. ISIS Wallet

ISIS Wallet is proposed by the wireless service providers, AT&T Mobility, T-Mobile USA, and Verizon Wireless that mainly target at the payment solutions. The payment application name is called the Sizzle Apps which runs on the phone to manage the payment information, loyalty data or the coupon offer.

The reader plays the role to gather all transaction data from the Sizzle Apps and pass it to the POS applications. According to the transaction requirements, there are two major payment methods to be used for a transaction:

1. Single Tap
2. Double Tap

The Single Tap is the simplest way to fit most of the transactions. The user just taps the phone once at the reader to complete the transaction. All the required payment information is sent to the POS application. The user will experience no difference from the current contactless payment schemes such as PayPass or PayWave.

For the Double Tap, the user is requested to tap at the reader twice. The first tap is same as the Single Tap for passing the payment information. The second tap is required for some types of transaction as for them the first tap cannot determine the final amount of the transaction.

At the moment, ***Bezel5*** only supports the Single Tap mode. It is expected to add the Double Tap support in the future.

ISIS Wallet Transaction Style

Operation	Support Function	Remark
Single Tap	Support	
Double Tap	Not support	Will support in future version

ISIS AID (Application Identifier)
The ISIS AID is hardcoded in the reader firmware and is listed as below:

Application Name	Application Identifier
Sizzle AID	A00000048510010101

8.1. Track Output Concept

Bezel5 is designed to eliminate the burdens of the communication between the reader and the host application. When the user taps the phone, the reader gathers all the data within the track format and output to the host just in one time. It does not require the application to send any command to get the card data. For the tracks data arrangement, Track 1, Track 2 and Track 3 contain the regular payment information, while Track 4 contains the ISIS wallet data of loyalty data or coupons. This arrangement of sending all card data out not only can save the communication time, but also make it easy for the *Bezel5* to work with the host applications like the Web POS applications which can only accept keyboard interface input.

This data output arrangement for the ISIS wallet data in the *Bezel5* is consistent to the other wallet applications and the contactless payment schemes being implemented in the same platform. It creates a uniform programming experience for the POS applications in dealing with different contactless payment solutions.

However, the working behavior of the *Bezel5* is a little bit different from the scenario described in the ISIS wallet specification. It requires the reader to output the data in two separate times. The *Bezel5* should be able to meet the ISIS wallet specification because it is up to the POS application to process the track data. The POS application can parse the Tag data to obtain the necessary information to calculate the total amount, and then parse the Track 1/2/3 to get the payment information to complete the transaction.

Track 1~3 and Tag FFFF820E Information

Track #	Description	SS/ES	Data
1	PayPass/payWave/Amex/Discover ZIP/Google Wallet Payment	%/?	Emulate magnetic stripe track 1: PAN, Card holder name, Expiration Date, Track 1 Discretionary Data.
2	PayPass/payWave/Amex/Discover ZIP /Google Wallet Payment	;/?	Emulate magnetic stripe track 2: PAN, Expiration Date, and Track 2 Discretionary Data.
3	PayPass/payWave	+/?	UIC proprietary data output for extra contactless payment information.
Tag	**Description**	**SS/ES**	**Data**
FFFF820E	ISIS Wallet application	#/?	UIC proprietary data output for ISIS Wallet.

Table 8-1 Track/Tag information of Google Wallet Transaction Format

8.2. Configuration Option

There are some configuration settings to be done before the unit is deployed to the field

Mode	Description
Wallet application deactivated	ISIS wallet is deactivated
Mifare First (default)	Google application (and Mifare functionality) is activated and Mifare is read first
ISIS Wallet	ISIS wallet is activated*. The operation PPSE or Sizzle AID is determined by TERMINAL_STARTUP_MODE. And the host can choose for 'ISIS only', or 'ISIS with the payment in MERCHANT_CAPABILITIES'.

* ISIS requests two different ways to start the application – manual or automatic (default).

8.3. Tag FFFF820E Output Format

The wallet data is output in Tag FFFF820E packed in a series of application data units according to the following format:

The data of all records are expressed in ASCII-HEX values. For Example, the character 'B' is expressed as '42'. If the data is in ASCII, it is embraced by […]. The Tag FFFF820E must be present if the tap is from the phone. It can either be the normal track or the empty track in cases of no application or read back with error. The data begins with the start sentinel "#" and ends with the end sentinel "?".

* Tag FFFF820E is always in clear text

ISIS Wallet data is output in Tag FFFF820E according to the following format:

Byte 0	Byte 1~4	Byte 5~6	Byte 7~8	Byte 9~10	Byte 11	Byte 12+n	Byte 13+n		Byte n	Byte n+1
[#], Start sentinel	Total length of track	Num of Tags	Num of Loyalty	Num of Offer	=	Tag^1	=	…	Tag^n	[?], End sentinel

Table 8-2. ISIS Wallet Tag Data Output Format

Empty Tag Format

Byte 0	Byte 1~4	Byte 5
[#], Start sentinel	0000	[?], End sentinel

The empty track has two different meanings

1. It may indicate the user taps the ISIS wallet phone for the payment but there is no ISIS wallet data. Usually, it comes after track 1 & 2 payment card data.
2. It may indicate the POST TRANSACTION command is sent to the ISIS wallet phone successfully.

Error Tag Format

Byte 0	Byte 1~4	Byte 5~8	Byte 9
[#], Start sentinel	Fxxx (Error code)	Status code, Optional (only available for some error codes)	[?], End sentinel

The error track data is starting with FXXXX to indicate ISIS wallet error reading. Or the error returned from POST TRANSACTION command.

Error code List

Error Code	Status Code	Description
F101	XXXX*	Get SmarTap Data Error
F102	N/A	Get SmarTap Data Error - buffer overflow
F103	N/A	Get SmarTap Data Error - Command Timeout
F104	N/A	Get SmarTap Data Error - the length of Card Response is wrong
F111	XXXX*	Post Transaction Data Error
F112	N/A	Post Transaction Data Error - buffer overflow
F113	N/A	Post Transaction Data Error - Command Timeout
F114	N/A	Post Transaction Data Error - the length of Card Response is wrong

* XXXX is from the ISIS wallet to be the status code other than 9000 (success) such as 6909 (ISIS wallet internal error). The programmer must refer to ISIS technical document for further information.

Tag FFFF820E Data Object Format

Data/Tag	Description	Data Object Format (Bytes) Output in ASCII (Bytes)
#	Start Sentinel	1 byte
0000	Total length of track	2 bytes

Data/Tag	Description	Data Object Format (Bytes) Output in ASCII (Bytes)
Xx	Num of tags	01 to 99
xx	Num of Loyalty	If num = 00, DF41/ DF43 doesn't present in track 4, If it exists, the number is LoyaltyID#x
Xx	Num of Offer	If num = 00, DF51/ DF53/ DF55 doesn't present in track 4 If it exists, the number is Offer_Type_Code#x
=	Field separator	
DF21	Customer ID	var bytes, if missed, it is empty field
=	Field separator	
DF41	LoyaltyID #1	var bytes, if missed, it is empty field
=	Field separator	
DF43	Loyalty Account #1	
=	Field separator	
…	…	…
DF41	LoyaltyID #x	var bytes, if missed, it doesn't present.
=	Field separator	
DF43	Loyalty Account #x	
=	Field separator	
DF51	Offer_ID#1	var bytes, if missed, it doesn't present.
=	Field separator	
DF53	Offer_Type_Code#1	var bytes, if missed, it doesn't present.
=	Field separator	
DF55	Offer signature#1	var bytes, if missed, it doesn't present.
…		
DF51	Offer_ID#x	var bytes, if missed, it doesn't present.
=	Field separator	
DF53	Offer_Type_Code#x	var bytes, if missed, it doesn't present.
=	Field separator	
DF55	Offer signature#x	var bytes, if missed, it doesn't present.
?	End Sentinel	1 byte

Table 8-3. ISIS Wallet Data Transmission Tag Format

8.4. ISIS Commands

Bezel5 supports the BLP command format and is mainly to update the EEPROM setting. In general, the factory or the system integrator uses the BLP protocol to configure the reader before deploying it to the field site. It can always be set back to a known state by the BLP protocol if the user doesn't know the current setting of the reader.

8.4.1. Configuration Command Protocol

The BLP protocol is used to store the configuration settings to the nonvolatile memory. The host can use the configuration commands to configure the bezel purposely to access the EMV card by the BLP protocol.

BLP Protocol – RS232 Interface

Byte 0	Byte 1~2	Byte 3+n	Byte 4+n
09h	Command Len	Command/Data	BCC

BLP Protocol - USB Interface (adding the header C2h and Len)

Byte 0	Byte 1~2	Byte 3	Byte 4~5	Byte 6+n	Byte 7+n
C2h	Len	09h	Command Len	Command/Data	BCC

* BCC is the LRC calculated the first byte to the last byte before BCC.
* Address field is not using, please use 00h
* If Command LEN is 00h, the bezel assumes the length of Command/Data field to be 3.

8.4.2. Activate/or deactivate wallet application

Command	Data Format/Example	Description
ISE	09 00 03 49 53 45 55	enable ISIS wallet
ISD	09 00 03 49 53 44 54	disable ISIS wallet

8.4.3. Merchant ID

The Merchant ID is a value assigned by ISIS that can be loaded into the NFC reader.

Load Merchant ID

Command	Data Format/Example	Description
ISM<Len, 1 byte><Merchant ID, var bytes>	09 00 0C 49 53 4D 08 11 22 33 44 55 66 77 01 5B	Set Merchant ID, Ex: set 11 22 33 44 55 66 77 01

Get Merchant ID

Command	Data Format/Example	Description
ISm	09 00 03 49 53 6D 7D	Get Merchant ID
(Response) <len, 1 byte>< Merchant ID, var bytes>	08 11 22 33 44 55 66 77 01	Returned Merchant ID, Ex: get 11 22 33 44 55 66 77 01

8.4.4. Merchant Store ID

The Merchant store ID is a value assigned by ISIS that can be loaded to the NFC reader.

Load Merchant Store ID

Command	Data Format/Example	Description
ISS<Len, 1 byte><Merchant Store ID, var bytes>	09 00 0C 49 53 53 08 11 22 33 44 55 66 77 88 CC	Set Merchant Store ID, Ex: set 11 22 33 44 55 66 77 88

Get Merchant Store ID

Command	Data Format/Example	Description
ISs	09 00 03 49 53 73 63	Get Merchant Store ID
(Response) <Len, 1 byte>< Merchant Store ID, var bytes>	08 11 22 33 44 55 66 77 88	Returned Merchant ID, Ex: get 11 22 33 44 55 66 77 88

8.4.5. Load Loyalty ID

The Loyalty Identifier number is a number assigned by ISIS that can be loaded to the NFC reader. It reserves a maximum of 50 records of the loyalty ID (total bytes should not exceed 256 bytes).

Load Loyalty ID

Command	Data Format/Example	Description
ISL<Len, 1 bytes>< Loyalty ID, var bytes>	09 00 0C 49 53 4C 08 11 22 33 44 55 66 77 01 5A	Add Loyalty ID, Ex: set 11 22 33 44 55 66 77 01

Get Loyalty ID

Command	Data Format/Example	Description
ISl	09 00 03 49 53 6C 7C	Get Loyalty ID
(Response) <Len, 1 byte>< Loyalty ID, var bytes>	08 11 22 33 44 55 66 77 01	Returned Loyalty ID, Ex: get 11 22 33 44 55 66 77 01

Erase Loyalty ID

Command	Data Format/Example	Description
ISRL	09 00 04 49 53 52 4C 09	Erase the last one in Loyalty ID list

Important: MEI 4-in-1 Plus accepts multiple Loyalty IDs. However, the new added Loyalty ID appended at the end of the list. The erase command will erase the last Loyalty ID from the list.

8.4.6. Load OFFER_TYPE_CODES

The Offer Type Codes are values assigned by ISIS that can be loaded to the NFC reader. It reserves a maximum of 50 records of the offer (total bytes should not exceed 256 bytes).

Load OFFER_TYPE_CODES

Command	Data Format/Example	Description
ISO<Len, 1 byte>< OFFER_TYPE_CODES, var bytes >	09 00 0D 49 53 4F 09 01 11 22 33 44 55 66 77 01 58	Add OFFER_TYPE_CODES, Ex: set 11 22 33 44 55 66 77 01

Get OFFER_TYPE_CODES

Command	Data Format/Example	Description
ISo	09 00 03 49 53 6F 7F	Get OFFER_TYPE_CODES
(Response) <Len, 1 byte>< OFFER_TYPE_CODES , car bytes>	09 01 11 22 33 44 55 66 77 01	Returned OFFER_TYPE_CODES, Ex: get 01 11 22 33 44 55 66 77 01

Erase OFFER_TYPE_CODES

Command	Data Format/Example	Description
ISRO	09 00 04 49 53 52 4F 0A	Erase the last one in OFFER_TYPE_CODES list

Important: MEI 4-in-1 Plus accepts multiple OFFER_TYPE_CODES. However, the new added OFFER_TYPE_CODES appended at the end of the list. The erase command will erase the last OFFER_TYPE_CODES from the list.

8.4.7. Load MERCHANT_CAPABILITIES

To accept ISIS wallet only or not.

Command	Data Format/Example	Description
IS1< MERCHANT_CAPABILITIES, 2 bytes >	09 00 05 49 53 31 F8 00 DF	Set Merchant Capabilities, Ex: set F8 00

The Merchant Capabilities value represents the different SmarTap features supported by the merchant.

Byte	Bit	Value	NFC Reader function
1	8 MSB	1=Merchant Loyalty Support 0=No	If Bit 8 is on, the reader will send the required Merchant ID and optionally a Loyalty ID for the merchant.
1	7	1=Secondary Loyalty 0=No	Bit 8 must be on. Additional Loyalty IDs will be included in the Get SmarTap Data request.
1	6	1=Offers Support 0=No	Offer Type fields will be included in the Get SmarTap Data request.
1	5	1=Post Transaction Data support	The NFC reader will receive redemption data from the ECR and forward it to the handset.
1	4	1=Contactless Payment support 0=No	Bit set to 0 will have the reader to accept the SmarTap -only operation but not other contactless payments
1	3-1	0	Reserved for Future Use
2	8-1	0	Reserved for Future Use

Note: BYTE 1 BIT 5 isn't changeable. It is always 1.

8.4.8. Load TERMINAL_STARTUP_MODE

The Terminal Start mode will determine in the reader the mechanism to be used to start the SmarTap application on the NFC reader.

Command	Data Format/Example	Description
IS2< TERMINAL_STARTUP_MODE, 2 bytes >	09 00 05 49 53 32 90 00 B4	Set TERMINAL_STARTUP_MODE, Ex: set 90 00

Byte	Bit	Value	NFC Reader function
1	8 MSB	1=Auto Start 0=No	At the start of the check the reader will quest for the SmarTap AID at the first TAP
1	7	1=Manual Start 0=No	The reader will only select a SmarTap AID after some user intervention.
1	6	1=Payment with Post Transaction Data 0=No	Payment and Post Transaction Data will occur on TAP 2
1	5	Payment First	Payment PPSE will precede in the Auto or Manual start modes
1	4-1	0	Reserved for Future Use
2	8-1	0	Reserved for Future Use

Note: Bits 7 and 8 are exclusive and only one bit can be turned on at a time.
Note: Bit 6 is fixed and can't be changed.
Note: Power cycling is needed after change the start mode.

8.4.9. Set SmarTap Application Version

Command	Data Format/Example	Description
IS2< SmarTap Application Version, 2 bytes >	09 00 05 49 53 30 01 01 26	Set SmarTap Application Version, Ex: set 01 01