UIC680

Contactless Smart Card Reader Module
RS232 & USB Interface

Programmer's Manual

Document #: PM090-C2
Revision 2.3

July 10, 2012

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.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated without the prior written permission of the issuer. 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 UIC680. 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 UIC680, 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 products, 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.

Table of Contents

Section 5 : Commands and Responses 5-1

5.1 Reader Command Description 5-1

5.2 Contactless Card Operation Command Description 5-13

Section 6 : Response to the Host 6-1

Section 7 : General Application Example 7-1

Section 1: General Description

This section presents general information about the basic characteristics of the UIC680.

1.1 Features

The UIC680 provides the following features:

1	Small footprint PCB size: 50 L* 40 W (mm) without antenna board
2	Supports ISO 14443 & ISO 18092 standard
3	Supports American Express® ExpressPay, MasterCard® PayPass^{TM} (Contactless MagStripe and M/Chip), Visa® PayWave (MSD and qVSDC), and Discover Network Zip Contactless Payments applications, Google Wallet (optional).
4	Reads/Writes NXP MIFARE Plus/Classic/Ultralight/DESFire cards
5	Encrypted card data output (optional)
6	Provides options for the direct coupling antenna and the remote antenna
7	Support RS232, USB 2.0 and serial TTL (optional) interfaces by use of corresponding cables.
8	RS232-to-RS232 pass-through (optional)
9	2-SAM or 4-SAM board (optional)

1.2 Application

The contactless smart card reader module UIC680 is mainly to support the contactless payment systems. The small footprint size of the module makes it easy to integrate to the current transaction system such as Point-of-Sale terminal, kiosk, and vending machine station. The module communicates with a host computer or terminal using a standard RS-232, USB or serial TTL (optional) interface.

Section 2: Configurations

This section shows the dimensions and accessories of the UIC680.

2.1 Dimensions of UIC680 PCBA
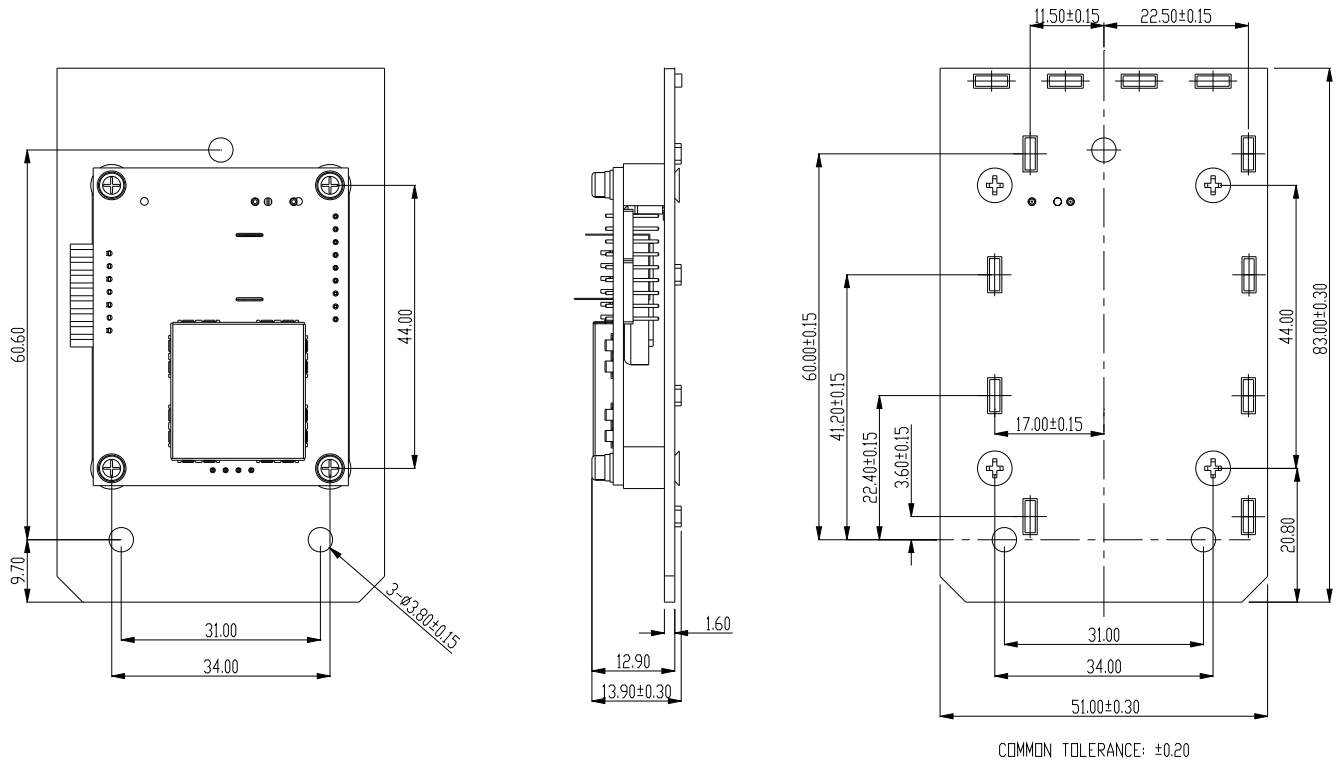
Section 3: Technical Specifications

3.1 Functional Specifications

Basic functions
- Contactless communication at 13.56MHz
- 4 LED indications for vertical or horizontal mount
- Programmable audio buzzer
- Real time clock /w 5 years battery life

Standards
- 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 Keyboard, or USB Virtual COM.
- RS232 data output baud rate up to 115.2K BPS
- RS232 pass through baud rate up to 115.2K BPS (optional)
- TTL level for serial data output (optional)

Antennas
- Build-in direct matching antenna
- Remote 50 ohm matching antenna (optional)

Payment applications
- American Express ExpressPay
- Discover ZIP
- MasterCard PayPass/MCHIP
- Visa MSD/qVSDC
- Google wallet (optional)

MIFARE applications
- Read/Write of MIFARE Plus/Classic/Ultralight/DESFire cards
- Support MIFARE higher baud rate up to 424KHz

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

Contact smart card
- Supports 2-SAM or 4-SAM board (optional)

3.2 Mechanical Specifications

- **Dimension**
  Without antenna board
  Length: 50 mm
  Width: 40 mm

3.3 Electrical Specifications

- **Power Required**
  5VDC ± 5%

- **Power Consumption**
  330mA in idle mode; 430mA in operating mode

- **Communication**
  Standard RS232 signal level
  Compatible with USB 2.0 specification
  TTL 5V signal level (optional)

- **Communication Signal (RS232)**
  Logic 1 = -3 volts to -15 volts or TTL level 5 volts
  Logic 0 = +3 volts to +15 volts or TTL level 0 volt

3.4 Environmental Specifications

- **Temperature**
  Operating: -20 to 70℃
  Storage: -30 to 70℃

- **Extended Temperature (Optional Model – Conformal Coating and No Battery.)**
  Operating: -25 to 80℃

  Storage: -30 to 85℃

- **Humidity**
  Operating: 10 to 85% (non condensing)
  Storage: 10 to 90% (non condensing)

3.5 Pin Assignment

Interface J5 Pin Assignment
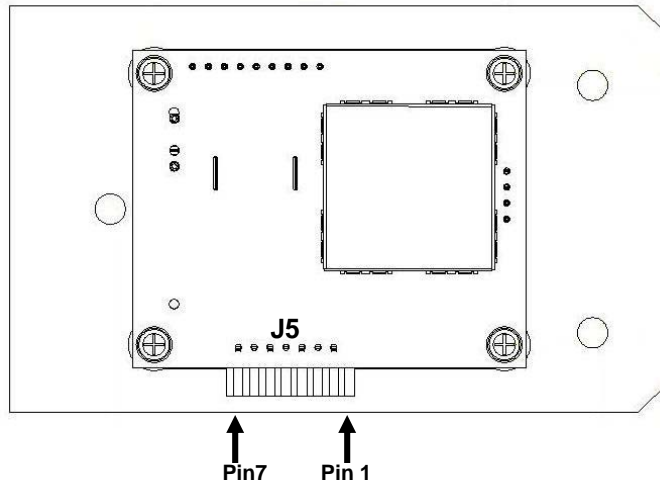
PCB-J5	Direction	Signal
1		GND
2	Serial data to host	TXD_Out
3	Serial data from host	RXD_In
4		VCC
5	USB data or Serial Pass-thru (optional)	USB D- or TXD2 (optional)
6	USB data or Serial Pass-thru (optional)	USB D+ or RXD2 (optional)
7		Shield

Direct Match Antenna Port J2A1 Pin Assignment

Contact number	Signal
1	RX
2	TX
3	GND
4	GND

3.6 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)

- **RS232 Interface Pass Through (Optional)**
  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: 681B(HID Keyboard) / 681C(Virtual COM) / 681D(HID-MSR)

Section 4: Operation

After power up the device, the first right vertical LED is turned on with one beep sound indicating that the reader is ready to operate.

As factory default setting, UIC680 is set to Self-Arm mode enabled. Under this mode, the reader will read and transmit payment card[1] data automatically. User needs to disable this mode in order to send contactless card operation commands.

4.1 Reader Default Setting

Item Description	EEPROM Default Value
UART1 setting (Txd1/Rxd1)	9600-8-N-1
Data pass through mode	Disable
UART2 setting (Txd1/Rxd2)	38400-8-N-1 (optional)
USB Interface	USB HID Keyboard
Buzzer	Enable
Protocol format	Protocol 0 (USI0)
Contactless Self-Arm mode	Enable
Contactless smart card manual type (only available in host poll mode)	Type A
Optional functions for the variant versions[2]	
Google application	Mifare First
Administration command protect	Enable
Data Encryption	Enable
Crypto Algorithm	TDES
DUKPT Key Management Mode	Auto rollover 1
Pre-load encryption key	(Customer specific or UIC default)
Pre-load Google Wallet merchant keys	Yes (per merchant request)

---

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

[2] Please contact UIC support team for more detail information.

4.2 Reader Configurations

- **Transmission Protocol**
  The user may select from three different protocols: Protocol 0, 1, 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 UIC680 is working in the USB interface, 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

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

  Protocol 1
  In Protocol 1, all messages are preceded by the ASCII character <STX> and terminated with the ASCII character <ETX>, followed by a one byte <BCC>.
  <BCC> is an XOR of the 7 data bits, excluding parity, of each character in the entire message, including <STX>.

  Format: <STX><MESSAGE><ETX><BCC>
  where STX=02Hex and ETX=03Hex.

  Example of Protocol 1, RS232 Interface

Host Command	Reader Response	Comment
<02h>P<03h><51h>		Ready to read
	<02h>^<03h><5Fh>	Reader ACK

Example of Protocol 1, USB Interface

Host Command	Reader Response	Comment
<C2h><00h><04h><02h>P<03h><51h>		Ready to read
	<C2h><00h><04h><02h>^<03h><5Fh>	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 1 and 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 1 or 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.

- **Self – Arm Mode**

  The default reader configuration is in "Self-Arm Mode". This allows the payment cards (including PayPass Magnetic Stripe, 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.

- **Host Poll Mode**

  Under this mode, user can send out commands manually. Examples like the Q, R, S,U 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.

- **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	
	Mifare Card Support	
	Disabled (MFxy = 10) [3]	**Enabled (MFxy = 11)**
Payment Card	Track data	Track data
Mifare Standard 1K	N/A	"M2"
Mifare Standard 4K	N/A	"M2"
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**[4].

---

[3] Please refer to UIC680 Configuration Guide for the detail information.

[4] Please refer to UIC680 Configuration Guide for the detail information.

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		
	Separator	Tk4 Data			Protocol Envelope code	Postamble

The preamble/postamble is only available in the card data output format under Self-Arm mode. The UIC680 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[5] (Self-Arm mode)
DUKPT data output format

Encrypt Mode			Encrypted Tk1 Data			Encrypted Tk2 Data			Encrypted Tk3 Data			Encrypted Tk4 Data			DUKPT KSN			Encrypted Session ID

RSA data output format

Encrypt Mode			Encrypted Tk1 Data			Encrypted Tk2 Data			Encrypted Tk3 Data			Encrypted Tk4 Data	

Notes   Encrypt Mode – 1: DUKPT TDES Mode

2: DUKPT AES Mode

3: RSA Mode

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

---

[5] Please refer to section 8 for more information.

Read TK4 data (optional) for command

Protocol Envelope code	Tk4* Data	Protocol Envelope code

Tk* Track Data Included:

SS (Optional)	Track Data	ES (Optional)	LRC (Optional)

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

- **Details of the Payment Card Tracks Data**

The UIC680 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. Some transactions may request extra information outside track 1 & track 2. The UIC680 has introduced a way to reduce the communication time between the host and the reader where the particular information can be stored in track 3 or even track 4. This depends on the card type and its application is described in the following sections.

Track 3 Data Format

The track 3 data of the contactless card is mainly provided for the additional information required by the payment transactions. It is currently available for the Paypass-Mchip and Visa cards (qVSDC, MSD) and not for other contactless payment cards. Track 3 data is the necessary additional information to be used for System Integration.

To simplify the host application process, this data only contains 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 of Tag Length Value and Its Description

Data/Tag	Description	Data Object Format (Bytes), Output in ASCII (Bytes)
+	Start Sentinel	
0	Card Type* – M/Chip	
02	Transaction Result** - Online	
=	Field Separator	
9F26	Application Cryptogram b8, 16	
9F10	Issuer Application Data	b var. up to 32, var. up to 64
9F37	Unpredictable Number	b4, 8
9F36	Application Transaction Counter	b2, 4
95	Terminal Verification Results	b5, 10
9B	Transaction Status Information	b2, 4
9A	Transaction Date	n3, 6
9C	Transaction Type	b1, 2
9F02	Amount, Authorized (Numeric)	n6, 12
5F2A	Transaction Currency Code	b2, 4
82	Application Interchange Profile	b2, 4
9F1A	Terminal Country Code	b2, 4
9F03	Amount, Other (Numeric)	n6, 12
9F33	Terminal Capabilities	b3, 6
9F35	Terminal Type	b1, 2
84	Dedicated File Name	b5~16 var. , 10~32 var.
9F09	Terminal Application Version Number	b2, 4
9F1E	Interface Device Serial Number (IFD)	an8, 8
9F16	Merchant ID	ans15, 15
9F41	Transaction Sequence Counter	b2~4 var. , 4~8 var.

Data/Tag	Description	Data Object Format (Bytes), Output in ASCII (Bytes)
9F27	Cryptogram Information Data	b1, 2
9F34	Cardholder Verification Method Results	b3, 3
9F53	Transaction Category Code	b1,2
5A	PAN	cn0~10 var. , 0~20 var.
5F24	Expiry Date	n3, 6
5F34	Application PAN Sequence Number	n1, 2
9F40	Additional Terminal Capabilities	b5, 10
9F6E	Form Factor Indicator	b4, 8
9F7C	Customer Exclusive Data	b 0~32 var. , 0~64 var.
57	Track 2 Equivalent Data	b1~19, 1~19
5F20	Cardholder Name	ans 2~26, 2~26
9F5D	Available Offline Spending Amount	n6, 12
9F68	Card Additional Processes	b4, 8
9F6C	Card Transaction Qualifier	b2, 4
9F6B	Card CVM Limit	n6, 12
9F51	Application Currency Code	b2, 4
9F17	PIN Try Counter	b1, 2
9F78	VLP Single Transaction Limit	n6, 12
9F79	VLP Available Funds	n6, 12
9F6D	VLP Reset Threshold	n6, 12
9F54	CTTA Limit	n6, 12
?	End Sentinel	

Value of Card Type

Value	Card Type*
0	MChip
1	RFU(MagStripe V3.3)
2	RFU
3	Visa(qVSDC, MSD)
4	RFU
5	Contactless VSDC

Value of Transaction Result

Value	Transaction Result**
00	Offline Approved
01	Offline Declined
02	Online
99	Terminate

Example:

Paypass –Mchip in track 3

+	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]	**?**

Visa (qVSDC, MSD) in track 3

+	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 Mode]	**?**

Example of Data Output:

+000=1F=AC99A931E778AE59=021190000324000000000980534968135D5FF=
69449501=0992=00A0000000==010101=00=000000000000=0000=5880=0000=
000000000000=000888=22=A0000000041010=0000=00000000=
303030303030303030303030303031=00000007=40=1F0300=00=
5412340600000005=091231=5412340600000005D09122010000000000000000==
MasterCard=MasterCardCredit=01=00?

Data parsed as follows:

Value	Description
0	Paypass Mchip Card
00	Offline transaction
1F	CVM
AC99A931E778AE59	9F26, Application Cryptogram
021190000324000000000980534968135D5FF	9F10, Issuer Application Data
69449501	9F37, Unpredictable Number
0992	9F36, Application Transaction
00A0000000	95, Terminal Verification Results

Value	Description
	9B, Transaction Status Information
010101	9A, Transaction Date
00	9C, Transaction Type
000000000000	9F02, Amount, Authorized (Numeric)
0000	5F2A, Transaction Currency Code
5880	82, Application Interchange Profile
0000	9F1A, Terminal Country Code
000000000000	9F03, Amount, Other (Numeric)
000888	9F33, Terminal Capabilities
22	9F35, Terminal Type
A0000000041010	84, Dedicated File Name
0000	9F09, Terminal Application Version Number
00000000	9F1E, Interface Device Serial Number (IFD)
3030303030303030303030303031	9F16, Merchant ID
00000007	9F41, Transaction Sequence Counter
40	9F27, Cryptogram Information Data
1F0300	9F34, Cardholder Verification Method Results
00	9F53, Transaction Category Code
5412340600000005	5A, PAN
091231	5F24, Expiry Date
5412340600000005D09122010000000000000	57, Track 2 equivalent Data
	9F6E, Paypass Third Party Data
MasterCard	50, Application Label
MasterCardCredit	9F12, Application Preferred Name
01	9F11, Issuer Code Table Index
00	5F34, Card Serial Number

Track 4 Data Format

The track 4 data of the contactless card is for the additional data or other payment card scheme.

Section 5: Commands and Responses

5.1 Reader Command Description

- **% (25H) – Retransmit**

    COMMENT   Retransmits the last message sent by the reader.

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

- **70 (37H30H) or 90 (39H30H) – Serial Number Report**

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

    EXAMPLE   '00000000'

- **71 (37H31H) or 91 (39H31H) – Copyright Report**

    COMMENT   Transmits version and copyright information.

    This command is sent if the user wants to know the version, model and copyright of the currently loaded UIC680 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 OEM version.

    EXAMPLE   '110811,UIC 68T0161C:V1.C'

- **7A (37H41H) or 9A (39H41H) – Module Version Report**

    COMMENT   Transmits version information.

Send this command when users want to know the version of the module in UIC680 firmware currently loaded. The response is a 6-byte ASCII string, reader type and the module version number.

7A[Module, 1 byte]

Module Description	In Byte	Example
HAL_VERSION	0	68TH11
PAYPASS_VERSION	1	68TP11
AMEX_VERSION	2	68TA11
VISA_VERSION	3	68TV11
DN_VERSION	4	68TD11
L1_VERSION	5	68T111
L2_VERSION	6	68T211
MIFARE_VERSION	7	68TM11

- **? (3FH) – Select Verbose Responses Command**

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

- **$ (24H) – Reader Status Request**

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

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 |

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')

- **# (23H) – Configuration Request**

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

NOTE   By using the configuration setting command, users can select standard or extended format.

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

Extended Configuration Bytes (16 bytes)

Byte	0	1	2	3	4	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

Protocol	00H = USI2; 03H = USI0; 06H = USI1.
Speed	00H=1200, 01H=2400, 02H=4800, 03H=9600, 04H=19.2k, 05H=38.4k, 06H=56k, and 07H=115.2k bps.
Address	Always 00H.

- **<CAN> (18H) – Clear Data Buffer**

  COMMENT   Clears read data buffers.

- **<DC2> (12H) – RS232 pass through enable (optional)**

  COMMENT   Enable RS232 data Pass Through (PT) function temporarily.
  FORMAT   The command characters are followed by an ASCII 'P' to enable the
                 RS232 pass through function. It is only valid in RS232 model.

  NOTE   Once the pass through mode is enabled, UIC680 passes all data between
                 COM1 and COM2 and do nothing. Power cycling (if the default of PT is
                 disable) or send the string '<DC2>P<DC2>a<DC2>s<DC2>S' brings
                 UIC680 back to normal operation. **Before sending this command, user
                 must assure the PT function is desired.**

- **<7FH> – Warm Reset**

  COMMENT   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.

- **P (50H) – Arm to Read**

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

  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 should issue the 'Q', 'R', 'S', 'U' 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.

- **p (70H) – Arm to Read (Used for Manufacturing Test Only)**
  COMMENT
  1. Equivalent to the 'P' command, except the card read acknowledgement which is not the '^' character.
  2. It will report a '('(28H) byte if the card media is detected, and a ')'(29H) byte when the media detection goes inactive.

- **<ESC> (1BH) – Abort Arm to Read**
  COMMENT
  1. Clear the data buffers.
  2. Aborts the **Arm to Read** command.
  3. Transmits the command acknowledgement byte ('^' 5EH) if successful.

- **Q (51H) – Transmit Track 1 data**

- **R (52H) – Transmit Track 2 data**

- **S (53H) – Transmit Track 3 data**

- **U (55H) – Transmit Track 4 data**
  COMMENT
  1. Detects card type automatically and processes data in the read buffer.
  2. If error is detected, transmits relevant error response ('*' or '+') (see **Response to the Host** section)
  3. Else, transmits the track data in ASCII.
  4. The DUKPT (TDes and AES) encrypted output data format:
     | Encrypted track data | DUKPT KSN | Encrypted Session ID |
  5. The RSA encrypted output data format:
     Base64[RSA Encrypt Data]

- **H (48H) – Self-Arm function disable/enable**
  COMMENT   Used for controlling Contactless auto read function temporarily.

  H[Enable/Disable, 1 byte]
  [Enable/Disable]

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

  RESPONSE   '^'   - Acknowledgement
  　　　　　　　'*'   - Cannot execute
  　　　　　　　'!'   - Bad parameter

Note that the UIC680 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 UIC680 is waiting for the card to be removed from the reading zone.
2. The payment card is failed to decode and the UIC680 is waiting for the card to be removed from the reading zone.

- **5 (35H) – Set RTC Time**

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

    5[CMD, 1 byte][Date or Time, 5 bytes]

    [CMD, 1byte]

[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

    Command: 1, 31h

    [Read Date, 5 bytes BCD]

    Command   ***<35><31>***

    Response   **<Year, 2 bytes><Month, 1 byte><Day, 1 byte><Week, 1 byte>**

    Note: BCD format from 0_{10} (0000_{BCD} = 0h) to 9_{10} (1001_{BCD} = 9h)

    EXAMPLE   <20*><12><07><09><01>

    20h 12h = Year 2012

    07h   = Month July

    09h   = Day 09

    01h   = Week Day Monday

    (01h – Monday, 02h – Tuesday, … 07h – Sunday)

    * The year <20> can be interpreted as space character.

    Command: 2, 32h

    [Read Time, 5 bytes BCD]

    Command   ***<35><32>***

    Response   **<Hour, 1 byte><Min., 1 byte><Sec., 1 byte><Millisecond, 2 bytes>**

    Note: BCD format from 0_{10} (0000_{BCD} = 0h) to 9_{10} (1001_{BCD} = 9h)

    EXAMPLE   <14><43><37><08><91>

    14h   = Hour 2pm

    43h   = Minute 43

37h   = Second 37

08h 91h   = Milliseconds 891

Command: 4, 34h

[Set Date, 5 bytes Hex]

Format	**\<Year, 2 bytes>\<Month, 1 byte >\<Day, 1 byte >\<Week, 1 byte >**
Note	Hex value format valid input:
	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.
	Default setting is \<20>\<01>\<01>\<01>\<01>, obtained by Read Date.
	MM: 01h – 0Ch (01 – 12)
	DD: 01h – 1Fh (01 – 31)
	WW: 01h – 07h (Monday – Sunday)
EXAMPLE	20 12, July 09, Monday
	20 12→14h 0Ch = Year 2012
	07 → 07h = Month July
	09 → 09h = Day 09
	01 → 01h = Week Day Monday
Command	***\<35>\<34>\<14>\<0C>\<07>\<09>\<01>***
Response	'^' – Acknowledgement
	'*' – Cannot execute (e.g. out of range)
	'!' – Bad parameter (e.g. incorrect length)

Command: 5, 35h

[Set Time, 5 bytes Hex]

Format	**\<Hour, 1 byte>\<Min., 1 byte>\<Sec., 1 byte>\<Millisecond, 2 bytes>**
Note	Hex value format valid input:
	Hour: 00h – 17h
	Minute: 00h – 3Bh
	Seconds: 00h – 3Bh
	Millisecond: 00h 00h – 03h E7h
EXAMPLE	14:26:30 10 ms
	14 → 0Eh = Hour
	26 → 1Ah = Minutes
	30 → 1Eh = Seconds
	10 → 00h 0Ah = Millisecond

Command   *<35><35><0E><1A><1E><00><0A>*

Response   '^' – Acknowledgement

                  '*' – Cannot execute (e.g. out of range)

                  '!' – Bad parameter (e.g. incorrect length)

- **B (42H) – Buzzer Beep control**

  COMMENT   Used to let Buzzer beep under user's control.

  B[Type, 1~4 bytes] – [Count, 1 byte][Tone, 1 byte(optional)][OnDuration, 1 byte(optional)][ShortDuration, 1 byte(optional)]

  [Count]

0 (30h, ASCII Hex) – long beep
1~9, A~F(31h~39h 41h~46h, ASCII Hex) – 1~15 short beeps

  Important:

  Once 'B0' command starts beeping, NO command can STOP it–unless users send a "Reset" command to stop it.

  [Tone]

  For adjusting the frequency level, 00h~FFh(high → low).

  [OnDuration]

  The duration of a beep; time unit is 10ms, 00h means 10ms, FFh means 2560ms.

  [ShortDuration]

  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.

  RESPONSE   '^' – Acknowledgement

                    '*'- Failed, the reader is in Self-Arm mode. The antenna power cannot be turned on.

- **I (49H) – Load RSA Key**
  COMMENT   This command is used to load RSA Key and query Key Index for PayPass
  MCHIP and VISA qVSDC applications.

I[Command, 1 or 3 bytes][Data, optional]

[Command]

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
5 (or 35h)	Load Test Key and Test RSA Chip
FFH 00H 00H	Erase all Key Entry

Command: 1, 31H

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

EXAMPLE

	Value
RID	0000000001
CA Index	03
Modulus	14C312D488A70988A4F219D5D67E798FDCA0A70D906663133A70981E6126F92B28 8BED98D69782CCA8C594B0CF2AB2ECE7399808574688A1B84BBCD20D37E92D1C 689A5BBD845A9988510C9A96EE445D4C1DA357AD3D142D5E8BC5D644549212317 E7A3552278BF8C67B5FBF650FFDAF577EF2297B336FEF6BA6536ADE3BA109
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>>+I<CD>8<CF>Y<99><88>y<F0>X<BF><86><C8
>'<E0><9E><91>

Note: These values are used for testing purposes.

- **T (54H) – Transaction Command**

  COMMENT   This command is used to view and manipulate transaction activities.

T[Command, 1 byte][Data, optional]

[Command]

ASCII – Hex Value	Description
0 (or 30h)	Erase all Transaction records
1 (or 31h)	Enter Amount
2 (or 31h)	Read Transaction Record
B (or 42h)	Query records Count
5 (or 35h)	Clear Amount

[Data]

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**.

NOTE   Record data of 'T2' command includes Transaction Date and Tracks data.

RESPONSE   '^' - Acknowledgement
                '*' - Cannot execute
                '!' - Bad parameter
                [Count] - Response for 'TB' command, 2 bytes in binary format.

- **w (77H) – Exception File**
  COMMENT   Adds or processes PAN in the Exception File. Primary Account Numbers
  kept by this black list will be denied for transactions.

  w[Command, 1 byte][Data, optional]

  [Command]

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
4 (or 34h)	Request a certain PAN from the Exception File

  [Data]
  For option 2 and 3, the format is data length(1 byte) + PAN(up to 19 bytes ASCII '0'~'9').
  For option 4, the format is 2 bytes long, range from 0000h to 010Fh.

  RESPONSE   '^' - Acknowledgement
                  '*' - Cannot execute
                  For 'w1' - return 2-byte binary number -- the total number of PANs in the file.
                  For 'w3' - return '1' if PAN exists; else, return '0'.

For 'w4' - return primary account number; else, return 00h.

EXAMPLE:

2 Bytes	Len (1 Byte)	PAN (Max: 19 Bytes)
W2	09h	"123456789"

5.2 Contactless Card Operation Command Description

- **G (47H) – ISO 14443 Type Protocol Select**

  COMMENT   Selects which manual command operated, ISO 14443 Type A or B.

  G[Type, 1 byte]

  [Type]

  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.

  RESPONSE   '^' - Acknowledgement

  '*' - Cannot execute

- **O (4FH) – Antenna power ON**

  COMMENT   Applies power on the antenna. This command is for manual command operation.

  RESPONSE   '^' - Acknowledgement

  '*' - Failed, the reader is in Self-Arm mode. The antenna power cannot be turned on.

- **o (6FH) – Antenna power OFF**

  COMMENT   Turns the antenna power off.

  Response   '^' - Acknowledgement

'*' - Failed, the reader is in Self-Arm mode. The antenna power cannot be turned off.

- **b (62H) – Request**

COMMENT   Request command.

b[Req command, 1 byte] - If reader is set to ISO 14443 type A

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

b[AFI, 1 byte][PARAM, 1 byte] - If reader is set to ISO 14443 type B
[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.

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

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

Response   ATQA(2 bytes, type A, Binary Hex) or ATQB(16 bytes, type B, Binary Hex) if command executed successfully
'*' - No card response or No power on the antenna

- **c (63H) – Anticollision(type A)/Slot-MARKER(type B)**
COMMENT   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.

c - ISO 14443 type A

c[APn, 1 byte] - ISO 14443 type B

[APn]

Anticollision Prefix byte, please refer to ISO 14443-3 for detailed information.

NOTE   Slot-MARKER is not a mandatory command for type B card.

Response   PICC serial number for type A(Binary Hex) or PICC send ATQB(12 bytes, Binary Hex) for type B if command executed successfully

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

'!' - Bad parameter

- **f (66H) – Select(type A)/Attrib(type B)**

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

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

  Response   '^' + SAK(1 byte, type A) or '^' + MBLI/CID(1 byte, type B) If command executed successfully

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

- **g (67H) – MIFARE Classic Card Authentication**

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

  g[Block number, 3 bytes][Key number, 1 byte][Key type, 1 byte]

  Authenticate the card with the key stored in EEPROM.

         or

  g[Block number, 3 bytes][Key type, 1 byte][Key, 12 bytes]

  Authenticate the card with the key in [Key] field.

  [Block number]

  There are two types of block number:

  1.   000 to 255 (30h30h30h to 32h35h35h, ASCII Hex)

             or

  2.   B<00><00> to B<00><FF>(42h00h00h to 42h00hFFh, ASCII Hex)

  [Key number]

0 to 4(30h to 34h, ASCII Hex)

[Key]
0 to 9 or A to F(30h to 39h or 41h to 46h, ASCII Hex)

[Key type]
A or B(41h or 42h)

Response   '^' - Acknowledgement

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

                   '!' - Bad parameter

- **h (68H) – MIFARE Classic Card Read Block(Supports MIFARE Ultralight)**
  COMMENT   MIFARE Classic card read command.

  h[Block number, 3 bytes]

  [Block number]
  There are two types of block number:
  1. 000 to 255 (30h30h30h to 32h35h35h, ASCII Hex)

                 or
  2. B<00><00> to B<00><FF>(42h00h00h to 42h00hFFh, ASCII Hex)

  Response   Block data(16 bytes, Binary Hex) if command executed successfully

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

- **i (69H) – MIFARE Classic Card Write Block(Supports MIFARE Ultralight)**
  COMMENT   MIFARE Classic card write command.

  i[Block number, 3 bytes][Block data, 4 bytes or 16 bytes]

  [Block number]
  There are three types of block number:
  1. 000 to 255(30h30h30h to 32h35h35h, ASCII Hex) - General MIFARE block

                 or

2.   B<00><00> to B<00><FF>(42h00h00h to 42h00hFFh, ASCII Hex) - General MIFARE
      block
                  or
3.   <00><00> to U<00><FF>(55h00h00h to 55h00hFFh, ASCII Hex) - MIFARE Ultralight

[Block data]
For MIFARE Ultralight, the block data should be 4 bytes; others using 16 bytes block data.

Response   '^' - Acknowledgement
                   '*' - No card response or No power on the antenna
                   '!' - Bad parameter

- **t (74H) – MIFARE Classic Card Value Operation**
  COMMENT   Value Block Operation commands.

t[Block number, 3 bytes][Operation mode, 1 byte][Value, 4 bytes][transfer block, 3 bytes]

[Block number]
There are two types of block number:
1.   000 to 255(30h30h30h to 32h35h35h, ASCII Hex)
                  or
2.   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)	Restore
3 (or 33h)	Decrement and transfer to the different block
4 (or 34h)	Create MIFARE Value in the block

[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.

Response   '^' - Acknowledgement
                '*' - No card response or No power on the antenna
                '!' - Bad parameter

- **W (57H) – ISO 14443A Detection**
  COMMENT   Detects ISO 14443A cards.

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

- **X (58H) – MIFARE Classic Card Activation(Supports MIFARE Ultralight)**
  COMMENT   Performs request/anticollision/select command to activate the card.
                 It is also can be used for any ISO 14443 compatible cards.

  Response   ATQA/SAK/serial number if command executed successfully for type A card
                 ATQB(12 bytes) if command executed successfully for type B card
                 '*' - No card response or No power on the antenna

- **x (78H) – Card HALT**
  COMMENT   Card halt command.

  Response   '^' - Acknowledgement
                 '*' - No power on the antenna

- **u (75H) – MIFARE Classic Card Read Sector**
  COMMENT   MIFARE Classic card read sector command.

  u[Sector number, 3 bytes]

[Sector number]

There are two types of sector number:

1.   000 to 255(30h30h30h to 32h35h35h, ASCII Hex)

         or

2.   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
                '*' - No card response or No power on the antenna

- **v (76H) – MIFARE Classic Card Write Sector**
  COMMENT   MIFARE Classic card write sector command.

  v[Sector number, 3 bytes][Sector data]

  [Sector number]

  There are two types of sector number:

  1.   000 to 255(30h30h30h to 32h35h35h, ASCII Hex)

           or

  2.   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.

  Response   '^' - Acknowledgement
                  '*' - No card response or No power on the antenna
                  '!' - Bad parameter

- **J (4AH) – Activate PICC cpu card**

  COMMENT   PICC cpu card activation command. The Antenna POWER ON command
                 has to be sent first.

  Response   ATS (type A) or PUPI (type B) if command executed successfully

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

- **j (6AH) – Load MIFARE Key(Supports MIFARE Classic only)**

  COMMENT   Saves up to 5 key sets for MIFARE Classic card application.

  NOTE   For security reasons, there is no way to retrieve the keys.

  j[Key number, 1 byte][Key data, 12 bytes]

  [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)

  RESPONSE   '^' - Acknowledgement

                 '*' - No power on the antenna

                 '!' - Bad parameter

- **F (58H) – Identify MIFARE Card Type**

  COMMENT   Reports MIFARE Card type.
                 It also can be used for any ISO 14443A compatible cards.

  RESPONSE   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).

- **y (79H) – Send DESELECT command**
  COMMENT   Sends ISO 14443 layer 4 DESELECT command to the card.

  Response   '^' - Acknowledgement

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

- **Z (5AH) – I/O to contactless CPU card with APDU format**
  COMMENT   The command is used to pass an APDU to the card where both data and
  an ISO status are expected in the response.

  Z[APDU, variant, 262 bytes max]
  [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

- **z (7AH) – I/O to contactless card for block data exchange**
  COMMENT   The command is used to pass a block data to a card.

  z[CRC mode, 1 byte][Wait time, 4 bytes][Block data, 384 bytes max]

[CRC mode]

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.

[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 '*'.

Section 6: Response to the Host

- **^ (5EH) Acknowledge. Acknowledges correct completion of most recent command.**

- *** (2AH) Command was received correctly but could not be completed.**

- **+ (2BH) No Magnetic Stripe Card Data. Command was received correctly.**

- **? (3FH) Communication Error. Command was not received correctly.**

- **! (21H) Invalid Command or bad parameter.**

- **: (3AH) Power On report.**

- **~ (7EH) Unavailable. Hardware is not available to complete this request.**

Section 7: General Application Example

The default setting of the UIC680 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 UIC680 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, no information is sent out.

If the application would like to take the whole 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 default setting in the reader configuration.

Once the UIC680 receives 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 UIC680 will turn back to the Self-Arm mode in the next power cycling. Besides, the Self-Arm enable command, 'H1', can also bring the UIC680 back to the Self-Arm mode.

To disable the Self-Arm mode permanently, the host needs to set the EEPROM value of the UIC680. The configuration command 'SA' (see the Configuration Guide) saves the setting into the EEPROM of the UIC680 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. This is better than using Protocol 0(USI0) or Protocol 1(USI1) at it can prevent the data to be misinterpreted but requires more redundant bytes.

Host Polled Application Example

A. UIC680 setting changed

In this example, we assume that the UIC680 is in factory default setting and the user would like to change the setting to protocol USI2 and Self-Arm disable mode.

```
        ┌──────────────────┐
        │   Initial State  │
        └──────────────────┘
                 │
                 ▼
 ┌───────────────────────────────────────┐
 │  1.  Send the configuration command    │
 │      to set theUIC680TG to protocol 2. │
 │  2.  Send the configuration command    │
 │      to set the UIC680TG to Self-ARM   │
 │      disable.                          │
 └───────────────────────────────────────┘
                 │
                 ▼
        ┌──────────────────┐
        │       Done       │
        └──────────────────┘
```
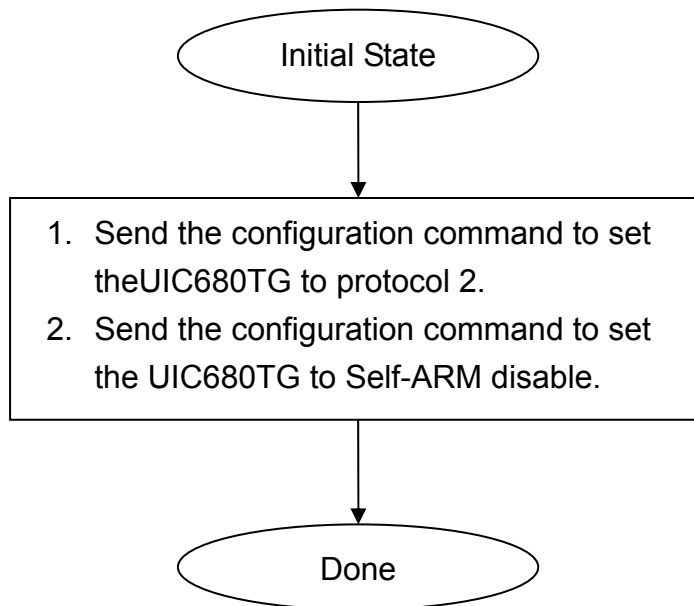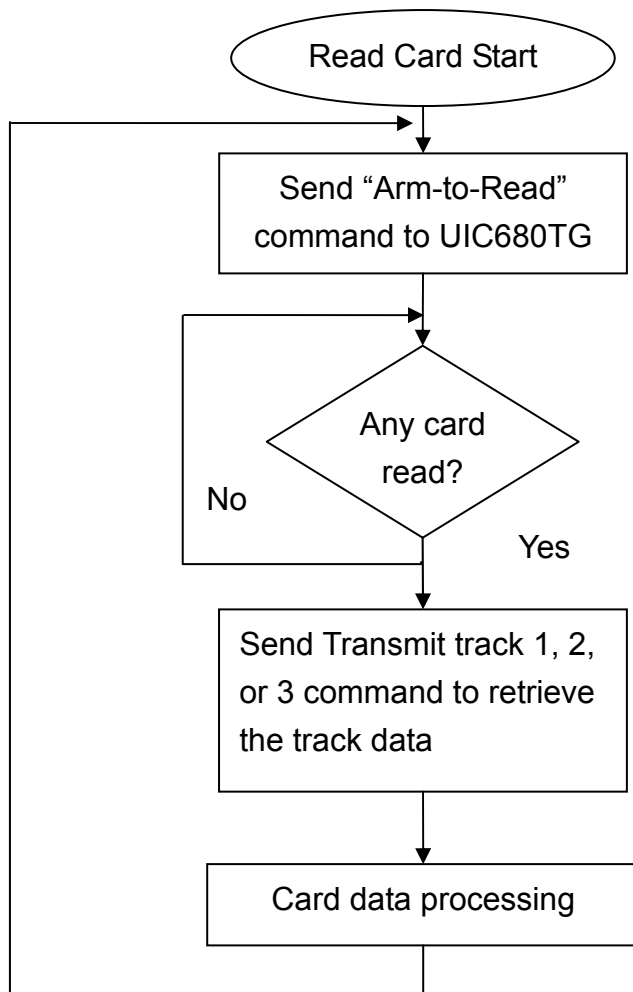
This process should be done only once to set the UIC680 to the proper state.

Due to the EEPROM life limitation, it is recommended not to change the EEPROM settings very frequently. Normally, the configuration setting is done in the factory production stage. Please contact the UIC Sales representative for more detail information.

B. Operation Example

In this example, assume the UIC680 is in protocol USI2 and Self-Arm disable mode.

```
        ( Read Card Start )
                 │
                 ▼
        ┌──────────────────────┐
   ┌───▶│  Send "Arm-to-Read"  │
   │    │ command to UIC680TG  │
   │    └──────────────────────┘
   │                │
   │                ▼
   │         ╱──────────╲
   │        ╱  Any card  ╲   No
   │   ┌───◀   read?      ◀──┐
   │   │    ╲            ╱    │
   │   │     ╲──────────╱     │
   │   │          │           │
   │   └──────────┘  Yes      │
   │                ▼
   │    ┌──────────────────────┐
   │    │ Send Transmit track  │
   │    │ 1, 2, or 3 command   │
   │    │ to retrieve the      │
   │    │ track data           │
   │    └──────────────────────┘
   │                │
   │                ▼
   │    ┌──────────────────────┐
   │    │ Card data processing │
   │    └──────────────────────┘
   │                │
   └────────────────┘
```

For next transaction

After the host issues the Arm-to-Read command, the UIC680 will check if any payment card is detected in the reading zone or any magnetic stripe card is swiped (the magnetic stripe reader is optional device). 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.