Confidentiality:
Reference: Amen-TR-2005031102
Version: 2.0
Project:

A-MEN Technology Corporation

16K SIM Card

Personalization Document

	Name	Title	Date	Singature
Prepared	Otto Hung		2005/03/11	
Reviewed	Tony Chang		2004/06/18	
Approved	Bernard Wang		2004/06/18	

Revision History Sheet (Please add the most recent revision at the beginning of the revision history list)			
Data	**Revision**	**Remark**	**By**
2004/06/18	1.0		Otto Hung
2004/07/03	1.1		Otto Hung
2005/03/11	2.0		Otto Hung

Contents

1. Card Information

Answer to Reset (ATR)

Default ATR value: 3B 34 11 00 6B CX 16 0Z

ATR data：

Byte(s)	Description
1	TS = 0x3B，is Forward Convention, lsb first
2	Format character，has TA1、TB1 and 4 historical bytes
3	TA1 = 0x11，setup up (1 etu = 372 clock)
4	TB1= 0
5	0x6B
6	Type of SIM CARD 0xC2 : Normal SIM + OTA 0xC3 : STK SIM
7	0x16, the EEPROM Size
8	GSM Algorithm 0x01: COMP128-V1 0x02: COMP128-V2 0x03: COMP128-V3 0x0A: Support all COMP128 version

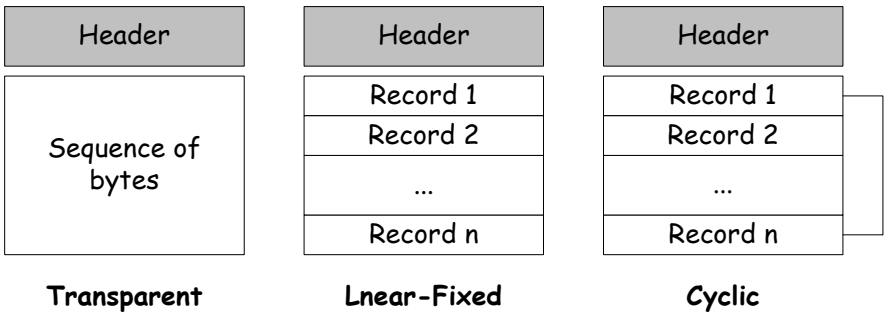
2. File System Structure

The 16K OTA SIM Card files stored in MEMORY are organized in a hierarchical structure. Each file contains a header that indicates the structure and attributes of the file. The file may also contain a data body whose access is handled by the operating system and header information.

The file header generated by off-cards File System Generator. The File System generator also generates a Checksum that used to active the file system by ATCIVE FILE SYSTEM APDU command.

There are two types of files, Dedicated File (DF) and Elementary File (EF). A DF, similar to a “directory” in computer operating systems such as DOS or UNIX, is the “parent” of a group of DF’s and/or EF’s. A DF consists only of a header part. An EF, similar to a “data file” in computer operating systems, is composed of a header and a data body. Data organization of an EF can be “transparent”, “linear fixed”, or “cyclic”. ISO 7816 defines another data organization called “linear variable”, which is not used in GSM 11.11.

A transparent EF consists of a sequence of bytes. Data being read or updated are referenced by an address relative to the first byte of the file. Data length of the body is indicated in the header.



A linear fixed EF consists of a sequence of records of a same length. Data are accessed by record number instead of byte address. Record length and total number of records in a file are indicated in the header.

Cyclic EF is used for storing records of a same length in chronological order. Record storage is managed on a First-in-First-out (FIFO) basis. Each cyclic EF contains no more than 255 records, and each record contains no more than 255 bytes.

A Master File (MF) is the root directory (i.e., root DF) of an application. Using the SELECT command, each DF or EF under the tree can be accessed by its unique ID.

2.1 Contents of the Elementary Files

This clause specifies the Elementary Files (EFs) for the GSM session defining access conditions, data items and coding. A data item is a part of an EF, which represents a complete logical entity.

An elementary file is composed of a header and a body part the structure of file is described in GSM 11.11. If EF has an unassigned valued, it may not be clear from the main test what the value should be. GSM 11.11 suggested contents of the EFs at pre-personalization. Referring to these contents, A-Men Technology Corporation 16K OTA SIM Card sets the default value of the EFs at pre-personalization.

This annex suggests values the file contents at pre-personalization that referring from GSM 11.11 version 7.4.0 Release 1998 Annex D.

Files Description Value

‘2F E2’ ICC identification operator dependant

‘2F 05’* Extended Language preference (this applies only to release 97 and later) ‘FF … FF’

‘6F 05’ Language preference ‘FF’

‘6F 07’ IMSI operator dependant

‘6F 20’ Ciphering key Kc ‘FF… FF07’

‘6F 30’ PLMN selector ‘FF … FF’

‘6F 31’ HPLMN search period ‘FF’

‘6F 37’ ACM maximum value ‘000000’

‘6F 38’ SIM service table operator dependant

‘6F 39’ Accumulated call meter ‘000000’

‘6F 3E’ Group identifier level 1 operator dependant

‘6F 3F’ Group identifier level 2 operator dependant

‘6F 41’ PUCT ‘FFFFFF0000’

‘6F 45’ CBMI ‘FF … FF’

‘6F 46’ Service provider name ‘FF … FF’

‘6F 48’ CBMID ‘FF … FF’

‘6F 49’* Service Dialing Numbers ‘FF … FF’

‘6F 74’ BCCH ‘FF … FF’

‘6F 78’ Access control class operator dependant

‘6F 7B’ Forbidden PLMNs ‘FF … FF’

‘6F 7E’ Location information ‘FFFFFFFF xxFxxx 0000 FF 01’

‘6F AD’ Administrative data operator dependant

‘6F AE’ Phase identification ‘02’

‘6F 3A’ Abbreviated dialing numbers ‘FF … FF’

‘6F 3B’ Fixed dialing numbers ‘FF … FF’

‘6F 3C’ Short messages ‘00FF … FF’

‘6F 3D’ Capability configuration parameters ‘FF … FF’

‘6F 40’ MSISDN storage ‘FF … FF’

‘6F 42’ SMS parameters ‘FF … FF’

‘6F 43’ SMS status ‘FF … FF’

‘6F 44’ Last number dialed ‘FF … FF’

‘6F 4A’ Extension 1 ‘FF … FF’

‘6F 4B’ Extension 2 ‘FF … FF’

‘6F 4C’* Extension 3 ‘FF … FF’

‘6F 4D’* Barred dialing numbers ‘FF … FF’

‘6F 4E’* Extension 4 ‘FF … FF’

*Phase 2+ files

The detail file structure and access description please refer to GSM 11.11.

2.2 System Extended Files

There are two system extended files of 16K OTA SIM Card, which EF_{PIN} '2FE5' and EF_{Key} '2FE6', that to contain the CHVs, ADMs, Ki and OTA Keys.

2.2.1 EF_{PIN} (PINs)

This file contains the Keys with 16K OTA SIM that is under MF '3F00'. The size of this file by default is 96 bytes with 8 records that contain CHV1, CHV2, PUK1, PUK2, and ADMx for GSM.

Identifier: "2FE5"	Structure: Linear Fixed		Mandatory
Record Size: 12 bytes			
Access Conditions:			
	READ	ADM	
	UPDATE	ADM	
	INVALIDATE	ADM	
	REHABILITATE	ADM	
Bytes	**Description**	**M/O**	**Length**
1	Initialize	M	1
2	Unblock KID	M	1
3	Max Retry number	M	1
4	Try Remain number	M	1
5 – 12	Key Content	M	8

The Key Identifier index, same as the Record Number on file

CHV1:	0x01	ADM1:	0x05
CHV2:	0x02	ADM2:	0x06
PUK1:	0x03	ADM3:	0x07
PUK2:	0x04	ADM4:	0x08

- **Initialize**

  Value 0x01 used to specify this PIN is active.

- **Unblock KID**

  Indicated the Key Identifier of the unblock key.

- **Maximum Retry Number**

  Indicated the number of Key verify retry.

- **Try Remain Number**

  Indicated the number of remain Key verify.

2.2.2 EF_{Key} (Keys)

This file contains the Ki for GSM authentication and OTA's Authentication Keys Kid.

Identifier: "2FE6"	Structure: linear fixed	Mandatory
Record size: 20 bytes		
Access Conditions: READ ADM UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Key ID	M	1
2	RFU	M	1
3	Key Type	M	1
4	Algorithm	M	1
5 – 20	Key Content	M	16

- **Key Identifier**

  Specifies the Key identifier of key, for GSM network the key ID of Ki shall be 0x00.

- **Key Type**

  Contents:

  This field used to specify this key Type, in this version, that COS only supported DES, Triple-DES, and COMP128-X Key.

  Coding:

No	Type (Bit7 ~ Bit4)				Length (Bit3 ~ Bit0)				Description
	B7	b6	b5	b4	B3	b2	b1	b0	
1	0	0	0	0	-	-	-	-	PIN (Personal Identified Number)
2	0	0	0	1	-	-	-	-	DES
3	0	0	1	0	X	x	x	x	TDES
4	0	0	1	0	0	0	1	0	2 Key Length (112 bits)
5	0	0	1	0	0	0	1	1	3 Key Length (168 nits)
6	0	1	0	0	-	-	-	-	Hash (COMP128)
7	0	1	0	1	-	-	-	-	AES

- **Algorithm**

  Contents:

  This field used to specify what the cryptographic algorithm could run.

  Coding:

No	Algorithm				Mode				Description
	B7	b6	b5	B4	b3	b2	b1	b0	
1	0								Symmetric Key Algorithm
2	0	-	-	1	-	-	-	1	Encrypt/Decrypt (EBC Mode)
3	0	-	-	1	-	-	1		Encrypt/Decrypt (CBC Mode)

4	0	-	1	-	-	-	-	-	MAC
5	0	1	-	-	-	-	-	-	Hash COM128
6	0	1	-	-	0	0	0	1	COMP128_1
7	0	1	-	-	0	0	1	0	COMP128_2
8	0	1	-	-	0	0	1	1	COMP123_3
Note1. by default the Symmetric Key always support the EBC mode encryption									

2.3 Toolkit Framework

The Toolkit Framework uses two files to store the toolkit applet, one is SAT Command script, coded **"6Fxx"**, and another is its special variables, coded **"4Fxx"**. Each different type of toolkit applet has its corresponding Applet File Identifier, Applet FID. All of Applet files shall be created under DF_APPLES **"7F0A"**. For reduce the file system space, when one applet has no its own variable, that shall not create variable pool space.

The SAT Commands has fixed command format for each different SAT Command. There are three type of variable object can be used in each SAT command, one is Common variable, one is Applet Variable, another is Edit variable. Each type of variable has its own Pool and different capabilities. For detail see the document SAT Command Format.doc.

Applet FID (6Fxx, 4Fxx)	Description
00 ~ 0F	**Event download**
10 ~ 1F	**Status (Poll interval)**
20 ~ 9F	**Menu Selection**
A0 ~ BF	**RFU**
C0 ~ CF	**Cell broadcast download**
D0 ~ DF	**SMS-PP download**
E0 ~ EF	**Profile Download**
F4	**Call control**
F6	**MO Short message control**

Toolkit Applet Identifier

2.3.1 EF_{AP_xx} (Toolkit Applet)

A Toolkit Applet is a set of SAT Command, called **Script**. A SAT Command was coded LV format, each command have its fixed value format, for detail see SAT Command Format.doc.

Identifier: "6Fxx"	Structure: Transparent	Optional
Record Size: X bytes		
Access Conditions:		
READ	Always	
UPDATE	ADM	
INVALIDATE	ADM	
REHABILITATE	ADM	

Bytes	Description	M/O	Length
1 – X	SAT Command Script	M	X

2.3.2 EF_{VP_xx} (Variable Pool)

This file used to store the Applet's special variables, called **Applet Variable Pool**. Any variable in this pool can store a new value with length less than old value.

Identifier: "4Fxx"	Structure: Transparent		Optional
Record Size: X+2 bytes			
Access Conditions: READ UPDATE INVALIDATE REHABILITATE		Always ADM ADM ADM	
Bytes	**Description**	**M/O**	**Length**
1 – 2	Remain	M	2
3 – X+2	Applet Variable Pool	M	X

2.4 STK Setting Files

There are some files to support Toolkit Framework to register framework capability, e.g. supported Event List, duration interval of Poll, and Toolkit applet menus. To support RFM applet, include COS, there are some files had been created in PERSO Phase. RFM, Remote File Management, can be update file content via Over the Air, OTA. All setting file are stored in DF_STK **"7F0E"** under the MF, Master File "**3F00"**.

2.4.1 $EF_{ENVELOPE}$

This file used to store the ENVELOPE BER-TLV contains which received form ME. This buffer is also used to keep the Proactive Command when the FETCH Command not present.

Identifier: "6F52"	Structure: Transparent		Mandatory
Record Size: 255 bytes			
Access Conditions: READ UPDATE INVALIDATE REHABILITATE		ADM ADM ADM ADM	
Bytes	**Description**	**M/O**	**Length**
1 – 255	Envelope Buffer	M	255

2.4.2 EF_{COMM} (Common Variable Pool)

This file used to store the Common Variable. The variable Identifier must from '01' to '80'. All variable in this pool can not be change. All applets can access this pool when its triggered.

Identifier: "6F56"	Structure: Transparent		Optional
Record Size: X+2 bytes			
Access Conditions: READ ADM UPDATE ADM INVALIDATE NEVER REHABILITATE NEVER			
Bytes	**Description**	**M/O**	**Length**
1 – 2	Remain	M	2
3 – X+2	Common Variable Pool	M	X

2.4.3 EF_{MENUS} (Menu Items)

This file contains Menu Items of SETUP-MENU proactive command.

Identifier: "6F58"	Structure: Linear Fixed		Optional
Record Size: X+2 bytes			
Access Conditions: READ ADM UPDATE ADM INVALIDATE NEVER REHABILITATE NEVER			
Bytes	**Description**	**M/O**	**Length**
1	Length	M	1
2	Identifier of Menu Item	M	1
3 – X+2	Alpha Text String of Item	M	X

- **Length**

  When length is not zero then indicates this record is available. Otherwise, that is to specify the length of text string plus 1 for Identifier of Menu Item.

- **Identifier of menu item**

  The identifier is a single byte between 0x20 and 0x9F. Each item shall have a unique identifier within this file. The identifier is also used to specify the corresponding toolkit applet file of SAT command script.

- **Text String of Item**

  The text string is coded in the same way as the alpha identifier of EF_{ADN}. Any unused bytes at the end of the value part shall be coded 0xFF.

2.4.4 EF_{Event} (Event List)

This file contains all Transparent Files contents shall be updated by selected Phone number.

Identifier: “6F5A”	Structure: Transparent		Optional
File size:			
Access Conditions: READ ADM UPDATE ADM INVALIDATE NEVER REHABILITATE NEVER			
Bytes	**Description**	**M/O**	**Length**
1 – X	Event List	M	X

- **Event List**

  A list of events of variable length. Each byte in the list defines an event. Each event type shall not appear more the one within the list.

  - ‘00’ = MT Call
  - ‘01’ = Call connected
  - ‘02’ = Call disconnected
  - ‘03’ = Location status
  - ‘04’ = User activity
  - ‘05’ = Idle screen available
  - ‘07’ = Language selection

2.4.5 EF_{Poll} (Poll Duration Interval)

This file contains all Transparent Files contents shall be updated by selected Phone number.

Identifier: “6F5D”	Structure: linear fixed		Optional
Record size: 3 bytes			
Access Conditions: READ ADM UPDATE ADM INVALIDATE NEVER REHABILITATE NEVER			
Bytes	**Description**	**M/O**	**Length**
1	Time Unit	M	1
2 – 3	Time Interval	M	1

- **Time Unit**

  Used Time unit; minutes, seconds or tenths of seconds.

- **Time Interval**

  The length of time required, expressed in units. The range is from 1 unit to 255 units.

2.4.6 EF_{LAC} (Local Area Code)

This file contains Local Area String with map to LAC (Local Area Code) in Local Informal. That is provide by PROVIDE LOCAL INFORMATION SAT command. It is allow usedr to change

Identifier: “6F1A”	Structure: linear fixed		Optional
Record size: X+2 bytes			
Access Conditions: READ ADM UPDATE ADM INVALIDATE NEVER REHABILITATE NEVER			
Bytes	**Description**	**M/O**	**Length**
1 – 2	Local Area Code	M	2
3 – X+3	Local Area Text String	M	X

- **Local Area Code**

  The Local Area Code with coding as Local Information on GSM 11.14.

- **Local Area Text String**

  The text string for Local Area .

2.5 RFM setting files

There are some administrative data the OTA has to keep track of. To store the information a set of elementary files is defined. The structure and contents is described below. All elementary file are stored in DF_RFM **‘7F0E’** under the MF, Master File **‘3F00’**.

2.5.1 EF_{TAR} (Toolkit Applet Reference)

The file contains the Toolkit Application Reference values that the OTA listens to. Incoming 03.48 messages that do not contain a TAR value listed in any of the records of this file is discarded.

Identifier: “6F01”	Structure: linear fixed		Mandatory
Record size: 3 bytes			
Access Conditions: READ AWLAYS UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	**Description**	**M/O**	**Length**

1 – 3	TAR	M	3

2.5.2 EF_{SMSH} (SMS Header)

This EF_{SMSH} file contains the Header information of the SMS_SUBMIT command. That used to send the Proof of Precept of 03.48 command packet.

Identifier: “6F04”	Structure: linear fixed		Mandatory
Record size: 14 bytes			
Access Conditions:			
READ	ALWAYS		
UPDATE	ADM		
INVALIDATE	ADM		
REHABILITATE	ADM		
Bytes	**Description**	**M/O**	**Length**
1 – 12	TP-Destination Address	M	12
13	TP-Protocol Identifier	M	1
14	TP-Data Coding Scheme	M	1

- **TP-Destination Address**

  As defined of SM-TL address fields in the GSM 03.40. Unused bytes set to 0xFF.

- TP-Protocol Identifier, TP-Data Coding Scheme AS defined in the GSM 03.40.

2.5.3 EF_{CNTR} (GSM 03.48 Counter)

This file contains the GSM 03.48 OTA Command Package counter.

Identifier: “6F06”	Structure: linear fixed		Optional
Record size: 5 bytes			
Access Conditions:			
READ	AWLAYS		
UPDATE	ADM		
INVALIDATE	ADM		
REHABILITATE	ADM		
Bytes	**Description**	**M/O**	**Length**
1 – 5	Counter (CNTR)	M	5

2.5.4 EF_{CMSB} (Short-Message Buffer)

This EF_{CSMB} file is the buffer of the OTA framework.

Identifier: “6F12”	Structure: Transparent	Mandatory
Record size: 136 bytes		
Access Conditions:		
READ	ALWAYS	

UPDATE	ADM		
INVALIDATE	ADM		
REHABILITATE	ADM		
Bytes	**Description**	**M/O**	**Length**
1	Active	M	1
2	Length	M	1
3 – 134	Segment of Concatenate SM	M	134

- **Active**

  Indicates this record is active.

- **Length**

  Indicates the length of concatenate short message segment.

- **Segment of Concatenate SM**

  This field stores the partial contain of the concatenate short message. In the case of uncompressed 8-bit data, the maximum length of the short message within the TP-UD field is 134 Bytes.

2.6 SIM Protected

To protect the SIM , there are two mechanisms provide by A_MEN SIM Card.

One is the RUN GSM ALGORITHM Counter, if the EF_AlgorithmCounter has exist, that every time the RUN GSM ALGORITHM is sent then the counter had reduce one. If the counter is zero the SIM card will block.

Another is Anti-Clone, when EF_AntiClone has exist and between two RUN GSM ALGORITHM has some same bytes, then the counter will reduce one. When this counter is zero then this SIM card will block. Two elementary files are stored in DF_GSM ‘7F20’ under the MF, Master File ‘3F00’.

2.6.1 $EF_{AlgorithmCounter}$ (GSM Algorithm Counter)

This file contains remain number to run the GSM Algorithm. While

Identifier: “8F1B”	Structure: Algorithm		Optional
Record size: 4 bytes			
Access Conditions:			
READ	ADM		
UPDATE	ADM		
INVALIDATE	ADM		
REHABILITATE	ADM		
Bytes	**Description**	**M/O**	**Length**
1 – 4	Counter	M	4

2.6.2 $EF_{AntiClone}$ (Anti Clone)

This $EF_{AntiClone}$ file is the counter and buffer of Anti Clone of RUN GSM ALGORITHM. While the random have 10 Bytes same as previous Random number the counter

Identifier: "8F1C"	Structure: Transparent		Mandatory
Record size: 136 bytes			
Access Conditions: READ ADM UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	**Description**	**M/O**	**Length**
1 – 4	Counter	M	4
5 – 20	Random number	M	16

- **Counter**

  Indicates the remain number of Anti Clone counter.

- **Random number**

  This field stores the random of previous RUN GSM ALGORITHM command.

2.7 File access conditions

All files protected by certain access conditions for different commands. Every file has its own specific access condition for each command. The relevant access condition of the last selected file shall be fulfilled before the requested action can take place.

For the command SEEK the same access condition is valid as set for the command READ. For the commands SELECT and STATUS the access conditions for the commands READ, UPDATE, DOWNLOAD, INVALIDATE and REHABILITATE are specified for each file. The DOWNLOAD access condition allows remote management by Over the Air (OTA). The access right shall specify to ADMx for operator controlled.

The access condition levels which are not hierarchical are defined in the following table:

Level	Access Condition
0	Always
1	CHV1
2	CHV2
3	RFU
4,…,7	ADM1, …,ADM8
8,…,14	RFU
15	Never

Always: Access is granted at all times requested

CHV1 (=Card Holder Verification 1):
One of the following three conditions has to be fulfilled in order to get access:
- the CHV1 value has already been verified during the current session
- the CHV1 is Disable able.
- the CHV1 has been successfully UNBLOCKed

CHV2 (=Card Holder Verification 2):
One of the following two conditions has to be fulfilled in order to get access:
- the CHV2 value has already been verified during the current session
- the CHV2 has been successfully UNBLOCKed

ADMx: This access condition can be used on a proprietary basis in agreement between the network operator and the SIM manufacturer.

NEVER Access from outside the 16K OTA SIM Card is granted at no time. Only the SIM internally is allowed to access this item.

3. Interface Commands

To execute the command of 16K OTA SIM Card by used Application Protocol Data Units (APDUs) which transmission protocol T=0. The command APDU has format (CLA, INS, P1, P2, P3, [data]), and the response APDU has format ([data], SW1, SW2). Coding of the class (CLA), instruction (INS) and status words (SW1, SW2) are fully compliant with the GSM 11.11.

3.1 GSM Commands

Except the Normal GSM Commands, the16K OTA SIM Card extended a VERIFY ADM command. The detail description of these APDU command introduce in **APPENDIX A**.

3.1.1 Summary of commands.

Table 1. GSM APDU Commands list

Command	CLA	INS	P1	P2	P3	Data S/R
SELECT[#]	'A0'	'A4'	'00'	'00'	'02'	S/R
STATUS[#]	'A0'	'F2'	'00'	'00'	length	R
READ BINARY[#]	'A0'	'B0'	offset high	offset low	length	R
UPDATE BINARY[#]	'A0'	'D6'	offset high	offset low	length	S
READ RECORD[#]	'A0'	'B2'	rec NO.	mode	length	R
UPDATE RECORD[#]	'A0'	'DC'	rec NO.	mode	length	S

SEEK[#]	‘A0’	‘A2’	‘00’	type/mode	length	S/R
INCREASE[#]	‘A0’	‘32’	‘00’	‘00’	‘03’	S/R
VERIFY ADM *[#]	‘A0’	‘2A’	‘00’	ADM No.	‘08’	S
VERIFY CHV	‘A0’	‘20’	‘00’	CHV No.	‘08’	S
CHANGE CHV	‘A0’	‘24’	‘00’	CHV No.	‘10’	S
DISABLE CHV	‘A0’	‘26’	‘00’	‘01’	‘08’	S
ENABLE CHV	‘A0’	‘28’	‘00’	‘01’	‘08’	S
UNBLOCK CHV[#]	‘A0’	‘2C’	‘00’	see note1	‘10’	S
INVALIDATE[#]	‘A0’	‘04’	‘00’	‘00’	‘00’	-
REHABILITATE[#]	‘A0’	‘44’	‘00’	‘00’	‘00’	-
RUN GSM ALGORITHM	‘A0’	‘88’	‘00’	‘00’	‘10’	S/R
SLEEP	‘A0’	‘FA’	‘00’	‘00’	‘00’	-
GET RESPONSE[#]	‘A0’	‘C0’	‘00’	‘00’	length	R

Note:

1. If the UNBLOCK CHV command applies to CHV1 then P2 is coded ‘00’; if it applies to CHV2 then P2 is coded ‘02’.
2. ‘ * ’ this command is not included in GSM 11.11 specification.
2. ‘ # ’ this command supported by OTA (Over the Air).

3.1.2 Status Conditions

This subclause specifies the coding of the status words SW1 and SW2.

Responses to commands which are correctly executed

SW1 SW2 Description

'90' '00' - normal ending of the command

'9F' 'XX' - length 'XX' of the response data

Memory management

SW1 SW2 Error description

'92' '0X' - command successful but after using an internal update retry routine 'X' times

'92' '40' - memory problem

Referencing management

SW1 SW2 Error description

'94' '00' - no EF selected

'94' '02' - out of range (invalid address)

'94' '04' - file ID not found

- pattern not found

'94' '08' - file is inconsistent with the command

Security management

SW1 SW2 Error description

'98' '02' - no CHV initialized

'98' '04' - access condition not fulfilled

- unsuccessful CHV verification, at least one attempt left
- unsuccessful UNBLOCK CHV verification, at least one attempt left
- authentication failed (see note)

'98' '08' - in contradiction with CHV status

'98' '10' - in contradiction with invalidation status

'98' '40' - unsuccessful CHV verification, no attempt left

- unsuccessful UNBLOCK CHV verification, no attempt left
- CHV blocked
- UNBLOCK CHV blocked

'98' '50' - increase cannot be performed, Max value reached

3.2 Proprietary commands

Pre-Personalization/Personalization commands also known as proprietary commands hereafter. The proprietary commands have three categories: accessing memory, configuring chip, and initializing COS. CLA for these commands is **H'70**.

Table 2. Proprietary Commands List for 16K OTA SIM

Command	CLA	INS	P1	P2	P3	Data Send/Return
VERIFY TRANSPORT KEY	'70'	'02'	'00'	'00'	'08'	S
ERASE ALL	'70'	'14'	'00'	'00'	'00'	-
WRITE MEMORY	'70'	'D0'	addr high	addr low	length	S
READ MEMORY	'70'	'B4'	addr high	addr low	length	R
LOAD KEY	'70'	'D8'	'00'	Kid	length	S
ACTIVE FILE SYSTEM	'70'	'50'	'00'	'00'	'01'	S

VERIFY ICCID	'70'	'56'	'00'	'00'	'10'	S
LOAD SYSTEM FLAG	'70'	'2C'	'00'	Type	for Type	S

3.2.1 Verify Transport Key

Command	CLA	INS	P1	P2	P3
VERIFY PROPRIETARY KEY	'70'	'02'	'00'	'00'	length

Description:

Verify Transport Key to permit execution of Proprietary commands.

Command parameters/data P3:

Byte(s)	Description	Length
1 – 8	Transport Key	8

3.2.2 ERASE ALL

Command	CLA	INS	P1	P2	P3
ERASE ALL	'70'	'14'	'00'	type	'00'

Description:

This command used to erase entire space of EEPROM.

3.2.3 WRITE MEMORY

Command	CLA	INS	P1	P2	P3
WRITE MEMORY	'70'	'D0'	addr. high	addr. low	length

Description:

Write 'length' byte(s) to EEPROM memory.

Command parameters/data P3:

Byte(s)	Description	Length
1 – length	Data to be written to the address specified in command.	length

3.2.4 READ MEMORY

Command	CLA	INS	P1	P2	P3
READ MEMORY	'70'	'B4'	addr. high	Addr. low	length

Description:

Read 'length' byte(s) from EEPROM memory.

Response parameters/data:

Byte(s)	Description	Length
1 – length	Data to be read from the EEPROM address specified in command.	length

3.2.5 PUT KEY

Command	CLA	INS	P1	P2	P3
PUT KEY	'A0'	'D8'	Type	kid	length

Description:

Put PIN or Key into EF_{PIN} or EF_{KEY}.

- **PUT PIN**

  Parameter P1 = 0x00,
  Parameter P2 specifies key identifier, KID:
  '01' = CHV1
  '02' = CHV2
  '03' = PUK1
  '04' = PUK2
  '05' ~ '08'= ADM1 ~ ADM4

  Command parameters/data:

Byte(s)	Description	Length
1	Unblock Key ID	1
2	Maximum retry number	1
3 – 10	Key content	8

- **PUT KEY**

  Parameter P1 = 0x80,
  Parameter P2, Key Identifier
  0x00: Ki for GSM network
  0x01 ~ 0x0F: for OTA Command Packet used.

  Command parameters/data:

Byte(s)	Description	Length
1	Key type	1
2	Key algorithm	1
3 – 18	Key content	16

3.2.6 ACTIVE FILE SYSTEM

Command	CLA	INS	P1	P2	P3
GSM ENABLE	'70'	'50'	'00'	'00'	'01'

Description:

Activate File System and GSM application (FDN function).

Command parameters/data P3：

Byte(s)	Description	Length
1	Total File number	1

To specify how many file had been created in File System by Off-Card file system generator.

3.2.7 VERIFY ICCID

Command	CLA	INS	P1	P2	P3
GSM ENABLE	'70'	'56'	'00'	'00'	'10'

Description:

Verify ICCID to permit execution of RUN GSM ALGORITHM command.

Command parameters/Data P3：

Byte(s)	Description	Length
1 – 16	ICCID Code	16

- ICCID Code
  This code must give form A-MEN.

3.2.8 LOAD SYSTEM FLAG

Command	CLA	INS	P1	P2	P3
LOAD SYSTEM FLAG	'70'	'2C'	'00'	Type	For P2

Description:

Load System flag for RiSIM STK-16 COS.

Command parameters/data P2:

P2	Description
0x03	Clock Stop Mode
0x04	Record Access Mode

- Clock Stop Mode Function:
  Command parameters/data P3：

Byte(s)	Description	Length
1	Allow the terminal can stop external clock	1

When Mode = 0x01, then this card allow the ME to stop the clock, than this card into the IDLE mode. To wake up this card by a new APDU Command is received. By default, this card set to allow stop the extern clock.

- Record Access Mode:
  Command parameters/data P3：

Byte(s)	Description	Length
1	The P3 of READ RECORD comm. Can less than selected file record length.	1

When Mode = 0x0A, then this card the P3 of READ RECORD Command less than the record length of selected current file.

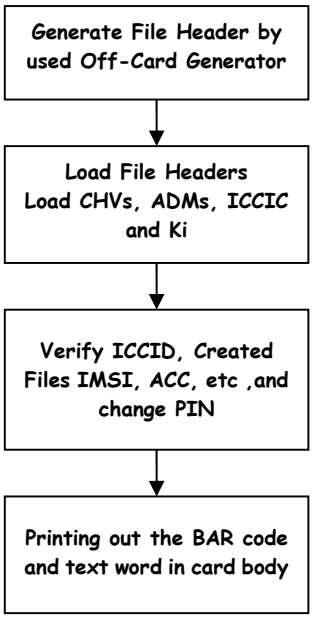
4. Personalization

In Personalization phase, there are some additional features:

- All GSM 11.11 SIM commands can be used under this mode.
- Ki store of EF_{KEY}
- The “Update Record” command can update record with data length less than record size or data length equal record, that setup up by LOAD SYSTTEM FLAG.
- The “Read Record” command can read record with data length less than record size or data length equal record size, setup up by LOAD SYSTEM FLAG.
- Only after Proprietary KEY verification, those proprietary commands are executable.

4.1 Personalization flow

To mass-produce same requirement of a big quantity order chips or cards. It is better to load CHVs, create all the files, put default values into files, download applets, and activate applets during wafer testing stage. If this is done, the personalization flow can be as follows:



These files have different contents from card to card.

4.2 Personalization command sample

//**********Personalization**********

RESET
3B 30 11 00

// Verify Proprietary KEY
70 02 00 00 02 E2 43 D5 71 B1 34 70 28
%90 00

// Clear the EEPROM
70 14 00 00 00
%90 00

// Create File System which Out-Card File system generator
70 D0 41 40 10 27 00 43 C0 46 E5 03 26 EA 3F 0A 40 00 00 00 00
%90 00

70 D0 41 50 80 3F 00 01 80 00 00 00 26 0D 66 44 44 44 04 03 00 2F E5 14 80 01 00 49 1A 00 60 44 FF FF 0C 08 00 2F E6 14 80 02 01 48 07 00 3C 44 FF FF 14 03 00 2F E2 04 80 03 02 47 65 00 10 04 FF 44 00 00 00 27 0E 02 80 04 03 00 08 00 00 44 44 44 00 04 00 6F 01 14 80 05 00 46 F2 00 03 44 FF 44 03 01 00 6F 04 04 80 06 05 47 47 00 0E 44 FF 44 00 00 00 6F 06 14 80 07 06 47 02 00 05 44 FF 44 05 01 00
%90 00

70 D0 41 D0 70 6F 52 04 80 08 07 49 FC 00 B0 44 FF 44 00 00 00 7F 10 02 80 09 04 00 13 00 00 44 44 44 00 0A 00 6F 3A 14 80 0A 00 4B B0 14 50 11 F4 22 1A C8 00 6F 3B 14 80 0B 0A 49 7A 00 82 12 F4 44 1A 05 00 6F 3C 14 80 0C 0B EA 40 0D C0 11 F4 44 B0 14 00 6F 3D 14 80 0D 0C 48 84 00 46 11 F4 44 0E 05 00 6F 44 34 80 0E 0D 4A AC 01 04 11 F4 44 1A 0A 01
%90 00

70 D0 42 40 80 6F 40 14 80 0F 0E 47 D3 00 34 11 F4 44 1A 02 00 6F 42 14 80 10 0F 48 CA 00 50 11 F4 44 28 02 00 6F 43 04 80 11 10 46 EA 00 02 11 F4 44 00 00 00 6F 4A 14 80 12 11 48 43 00 41 11 F4 44 0D 05 00 6F 4B 14 80 13 12 47 3A 00 0D 12 F4 44 0D 01 00 7F 20 02 80 14 09 00 25 00 00 44 44 44 00 11 00 6F 05 04 80 15 00 46 F9 00 04 01 F4 44 00 00 00 6F 07 04 80 16 15 47 10 00 09 14 F4 14 00 00 00
%90 00

70 D0 42 B0 70 6F 20 04 80 17 16 47 07 00 09 11 F4 44 00 00 00 6F 30 04 80 18 17 47 75 00 18 11 F4 44 00 00 00 6F 31 04 80 19 18 46 E7 00 01 14 F4 44 00 00 00 6F 37 04 80 1A 19 46 EF 00 03 12 F4 44 00 00 00 6F 38 04 80 1B 1A 47 19 00 0A 14 F4 44 00 00 00 6F 39 34 88 1C 1B 47 8D 00 1E 12 14 44 03 0A 01 6F 41 04 80 1D 1C 46 FD 00 05 12 F4 44 00 00 00

%90 00

70 D0 43 30 90 6F 48 14 80 1E 1D 46 F5 00 04 14 44 44 02 02 00 6F 45 04 80 1F 1E 47 AB 00 28 11 F4 44 00 00 00 6F 74 04 80 20 1F 47 55 00 10 11 F4 44 00 00 00 6F 78 04 80 21 20 46 E8 00 02 14 F4 44 00 00 00 6F 7B 04 80 22 21 47 2E 00 0C 11 F4 44 00 00 00 6F 7E 04 80 23 22 47 23 00 0B 11 F4 14 00 00 00 6F AD 04 80 24 23 46 EC 00 03 04 F4 44 00 00 00 6F AE 04 80 25 24 46 E6 00 01 04 F4 44 00 00 00 7F 21
%90 00

// Write the ICCID by WRITE MENORY Command
70 D0 $$ $$ 09 988896XXXXXXXXXXXXXX
%90 00

// Load System Flag
// Allow the Record Access mode with LEQ record size.
70 2C 00 04 01 0A
%9000

// Active the File system, xx to specify number of files had been created
70 50 00 00 01 xx
%9000

// Load PIN1
70 D8 00 01 0A 03 03 30303030FFFFFFFF
%90 00

// Load PIN2
70 D8 00 02 0A 04 03 38383838FFFFFFFF
%90 00

// Load PUK1
70 D8 00 03 0A 03 0A 3030303030303030
%90 00

// Load PUK2
70 D8 00 04 0A 04 0A 3838383838383838
%90 00

// Load ADM
70 D8 00 05 0A 05 0A 3838383838383838
%90 00

// Load Ki and use COMP128-1 Algorithm
70 D8 80 81 12 40 41 KKKKKKKK KKKKKKKK KKKKKKKK KKKKKKKK
%90 00

// Verify and Initial File Content

RESET
3B 30 11 00

// Verify PIN1
A0 20 00 01 08 30303030 FFFFFFFF
%90 00

// Verify PIN2
A0 20 00 02 08 38383838 FFFFFFFF
%90 00

// Verify ADM
A0 2A 00 05 08 38383838 38383838
%90 00

// Create EF ICCID
A0 A0 00 00 02 2FE2
%9F 0F

//Update ICCID * each byte High/Low nibble swap
A0 B0 00 00 0A 988896XXXXXXXXXXXXXX
%90 00

// Create DF GSM
A0 A4 00 00 02 7F20
%9F 17

// Create EF IMSI
A0 A4 00 00 02 6F07
%90 00

//Update IMSI value * each byte High/Low nibble swap after fix value 08x9*
A0 D6 00 00 09 084966**********
%90 00

// Create EF ACC
A0 A4 00 00 02 6F78
%90 00

//Update ACC value * Last Digit of IMSI*
A0 D6 00 00 02 $$$$
%90 00

//*********END !**********

Appendix A GSM APDU Commands

Table A.1 GSM Command Set - Quick Reference Table

Command	CLA	INS	P1	P2	P3	Data S/R
SELECT#	'A0'	'A4'	'00'	'00'	'02'	S/R
STATUS#	'A0'	'F2'	'00'	'00'	length	R
READ BINARY#	'A0'	'B0'	offset high	offset low	length	R
UPDATE BINARY#	'A0'	'D6'	offset high	offset low	length	S
READ RECORD#	'A0'	'B2'	rec NO.	mode	length	R
UPDATE RECORD#	'A0'	'DC'	rec NO.	mode	length	S
SEEK#	'A0'	'A2'	'00'	type/mode	length	S/R
INCREASE#	'A0'	'32'	'00'	'00'	'03'	S/R
VERIFY ADM *#	'A0'	'2A'	'00'	ADM No.	'08'	S
VERIFY CHV	'A0'	'20'	'00'	CHV No.	'08'	S
CHANGE CHV	'A0'	'24'	'00'	CHV No.	'10'	S
DISABLE CHV	'A0'	'26'	'00'	'01'	'08'	S
ENABLE CHV	'A0'	'28'	'00'	'01'	'08'	S
UNBLOCK CHV#	'A0'	'2C'	'00'	see note1	'10'	S
INVALIDATE#	'A0'	'04'	'00'	'00'	'00'	-
REHABILITATE#	'A0'	'44'	'00'	'00'	'00'	-
RUN GSM ALGORITHM	'A0'	'88'	'00'	'00'	'10'	S/R
SLEEP	'A0'	'FA'	'00'	'00'	'00'	-
GET RESPONSE#	'A0'	'C0'	'00'	'00'	length	R

Note:

1. If the UNBLOCK CHV command applies to CHV1 then P2 is coded '00'; if it applies to CHV2 then P2 is coded '02'.
2. ' * ' this command is not included in GSM 11.11 specification.
2. ' # ' this command supported by OTA (Over the Air).

A.1 SECLECT

Command	CLA	INS	P1	P2	P3
SELECT	'A0'	'A4'	'00'	'00'	'02'

Description:

Select a file in GSM 11.11 specifications.

Command parameters/data P3:

Byte(s)	Description	Length
1 – 2	File ID.	2

Response parameters/data in case of an MF or DF:

Byte(s)	Description	Length
1 – 2	RFU	2
3 – 4	Total number of unused bytes under the current directory	2
5 – 6	File ID	2
7	Type of file：MF = 01，DF = 02	1
8	RFU	1
9 – 11	Access conditions byte 9：RFU byte 10：DELETE FILE | CREATE FILE byte 11：RFU	3
12	File Status： bit 0：0 = invalidated，1 = not invalidated bit 1 – bit 7：RFU	1
13	Length of the following data (0x0A)	1
14	File characteristics 0x10 = CHV1 enable; clock stop not allowed. 0x90 = CHV1 disabled; clock stop not allowed.	1
15	Number of DFs which are a direct child of the current directory	1
16	Number of EFs which are a direct child of the current directory	1
17	Number of CHVs，UNBLOCK CHVs and administrative codes	1
18	RFU	1
19	CHV1 status 0x8n = CHV1 initialised；'n'：0 – 3 false presentation remaining 0x0n = CHV1 not initialised	1
20	UNBLOCK CHV1 status 0x8n =UNBLOCK CHV1 initialised； 'n'：0 – A false presentation remaining 0x0n = UNBLOCK CHV1 not initialised	1
21	CHV2 status 0x8n = CHV2 initialised；'n'：0 – 3 false presentation remaining 0x0n = CHV2 not initialised	1
22	UNBLOCK CHV2 status 0x8n =UNBLOCK CHV2 initialised； 'n'：0 – A false presentation remaining 0x0n = UNBLOCK CHV2 not initialised	1
23	RFU	1

Response parameters/data in case of an EF:

Byte(s)	Description	Length
1 – 2	RFU	2
3 – 4	File size (not include file header)	2
5 – 6	File ID	2
7	Type of file：EF = 04	1
8	Cyclic increase flag 0x00：INCREASE not allowed on cyclic EF 0x40：INCREASE allowed on cyclic EF	1
9 – 11	Access conditions byte 9：READ | UPDATE byte 10：INCREASE | RFU byte 11：REHABILITATE | INVALIDATE	3
12	File Status： bit 0：0 = invalidated，1 = not invalidated bit 1 – bit 7：RFU	1
13	Length of the following data (0x02)	1
14	Structure of EF 0x00：transparent 0x01：linear fixed 0x03：cyclic	1
15	Length of a record (transparent EF = 0x00)	1

A.2 STATUS

Command	CLA	INS	P1	P2	P3
STATUS	‘A0’	‘F2’	‘00’	‘00’	length

Description:

The command returns information concerning the current directory.

Response parameters/data:

Byte(s)	Description	Length
1 – length	Response parent file information (MF or DF)	length

A.3 READ BINARY

Command	CLA	INS	P1	P2	P3
READ BINARY	‘A0’	‘B0’	offset high	offset low	length

Description:

Read ‘length’ byte(s) from current selected Transparent EF.

Response parameters/data:

Byte(s)	Description	Length
1 – length	Data to be read from Transparent EF.	length

A.4 UPDATE BINARY

Command	CLA	INS	P1	P2	P3
UPDATE BINARY	‘A0’	‘D6’	offset high	offset low	length

Description:

Update ‘length’ byte(s) to current selected Transparent EF.

Command parameters/data P3:

Byte(s)	Description	Length
1 – length	Update ‘length’ byte(s) data to transparent EF.	length

A.5 READ RECORD

Command	CLA	INS	P1	P2	P3
READ RECORD	‘A0’	‘B2’	Rec. No.	Mode	length

Description:

Read one record data from Linear fixed EF or Cyclic EF.
Parameter P2 specifies the mode:
-'02' = next record;
-'03' = previous record;
-'04' = absolute mode/current mode, the record number is given in P1 with P1='00' denoting the current record.
For the modes "next" and "previous" P1 has no significance and shall be set to '00' by the ME. To ensure phase compatibility between Phase 2 SIMs and Phase 1 MEs, the SIM shall not interpret the value given by the ME. Parameter P3 specifies the length of reading that can one complete record or less than one record size decided on Create MF phase (see the command detail from the section Create MF).

Response parameters/data :

Byte(s)	Description	Length
1 – length	The data of the record	length

A.6 UPDATE RECORD

Command	CLA	INS	P1	P2	P3
UPDATE RECORD	‘A0’	‘DC’	Rec. No.	Mode	length

Description:

Update Record data to Linear fixed EF or Cyclic EF.
Parameter P2 specifies the mode:
-'02' = next record;

-'03' = previous record;
-'04' = absolute mode/current mode, the record number is given in P1 with P1='00' denoting the current record.
For the modes "next" and "previous" P1 has no significance and shall be set to '00' by the ME. To ensure phase compatibility between Phase 2 SIMs and Phase 1 MEs, the SIM shall not interpret the value given by the ME. Parameter P3 specifies the length of reading that can one complete record or less than one record size decided on Create MF phase (see the command detail from the section Create MF).

Command parameters/data P3:

Byte(s)	Description	Length
1 – length	Data	length

A.7 SEEK

Command	CLA	INS	P1	P2	P3
SEEK	'A0'	'A2'	'00'	Type/Mode	length

Description:
The command searches through the current Linear fixed EF to find a record starting with the given pattern.
Parameter P2 specifies type and mode:
-'x0' = from the beginning forward;
-'x1' = from the end backward;
-'x2' = from the next location forward;
-'x3' = from the previous location backward
with x='0' specifies type 1 and x='1' specifies type 2 of the SEEK command.

Command parameters/data P3:

Byte(s)	Description	Length
1 – length	Pattern	length

There are no response parameters/data for a type 1. A type 2 SEEK function returns the following

Response parameters/data:

Byte(s)	Description	Length
1	Record number	1

A.8 INCREASE

Command	CLA	INS	P1	P2	P3
INCREASE	'A0'	'32	'00'	'00'	'03'

Description:
Adding the value given by the ME to the value of the last increase /

updated record of the current selected cyclic EF.

Command parameters/data P3

Byte(s)	Description	Length
1 – 3	Value to be added	3

Response parameters/data :

Byte(s)	Description	Length
1 – X	Value of the increased record	X
X+1 – X+3	Value which has been added	3

NOTE: X denotes the length of the record.

A.9 VERIFY ADM

Command	CLA	INS	P1	P2	P3
VERIFY CHV	'A0'	'2A'	'00'	ADM No.	'08

Description:

Verify ADM (P2= 05) ~ ADM4 (P2= 08).

Command parameters/data P3:

Byte(s)	Description	Length
1 – 8	ADM Value	8

A.10 VERIFY CHV

Command	CLA	INS	P1	P2	P3
VERIFY CHV	'A0'	'20'	'00'	CHV No.	'08

Description:

Verify CHV1 (P2= 01) or CHV2 (P2= 02).

Command parameters/data P3:

Byte(s)	Description	Length
1 – 8	CHV value	8

A.11 CHANGE CHV

Command	CLA	INS	P1	P2	P3
CHANGE CHV	'A0'	'24'	'00'	CHV No.	'10'

Description:

Verify CHV1 (P2= 01) or CHV2 (P2= 02).

Command parameters/data P3:

Byte(s)	Description	Length
1 – 8	Old CHV value	8
9 – 16	New CHV value	8

A.12 DISABLE CHV

Command	CLA	INS	P1	P2	P3
DISABLE CHV	'A0'	'26'	'00'	'01'	'08'

Description:

Disable CHV1.

Command parameters/data P3:

Byte(s)	Description	Length
1 – 8	CHV1 value	8

A.13 ENABLE CHV

Command	CLA	INS	P1	P2	P3
ENABLE CHV	'A0'	'28'	'00'	'01'	'08'

Description:

Enable CHV1.

Command parameters/data P3:

Byte(s)	Description	Length
1 – 8	CHV1 value	8

A.14 UNBLOCK CHV

Command	CLA	INS	P1	P2	P3
UNBLOCK CHV	'A0'	'2C'	'00'	CHV No.	'10'

Description:

Unblock CHV1 (P2= 00) or CHV2 (P2= 02).

Command parameters/data P3:

Byte(s)	Description	Length
1 – 8	Unblock CHV value	8
9 – 16	New CHV value	8

A.15 INVALIDATE

Command	CLA	INS	P1	P2	P3
INVALIDATE	‘A0’	‘04’	‘00’	‘00’	‘00’

Description:

The command invalidates the current EF.

A.16 REHABILITATE

Command	CLA	INS	P1	P2	P3
REHABILITATE	‘A0’	‘44’	‘00’	‘00’	‘00’

Description:

The command rehabilitates the invalidated current EF.

A.17 RUN GSM ALGORITHM

Command	CLA	INS	P1	P2	P3
RUN GSM ALGORITHM	‘A0’	‘88’	‘00’	‘00’	‘10’

Description:

Add the value given by the ME to the value of the last increase / updated record of the current cyclic EF.

Command parameters/data P3:

Byte(s)	Description	Length
1 – 16	RAND	16

Response parameters/data:

Byte(s)	Description	Length
1 – 4	SRES	4
5 – 12	Cipher Key Kc	8

Note: significant bit of SRES is coded on bit 8 of byte 1.
The most significant bit of Kc is coded on bit 8 of byte 5.

A. 18 SLEEP

Command	CLA	INS	P1	P2	P3
SLEEP	‘A0’	‘FA’	‘00’	‘00’	‘00’

Description:

The command is used by Phase 1 MEs only.

A. 19 GET RESPONSE

Command	CLA	INS	P1	P2	P3
GET RESOPNSE	‘A0’	‘C0’	‘00’	‘00’	length

Description:

The command get response data depends on the preceding command.

Response parameters/data:

Byte(s)	Description	Length
1 – length	Response data	length