



PC/SC PROTOCOL SPECIFICATION

DUALi Inc.

Document Version: 2.14

Last Revised Date: 09. Jan. 2018

Copyright © 2016 DUALi Inc. All rights reserved. You are strictly prohibited to copy, disclose, distribute, or use this document in part or as a whole for any purposes other than those for which this document is disclosed. This document is copyrighted and contains confidential information and other intellectual property rights of DUALi Inc. Any unauthorized use, copy, disclosure or distribution constitutes infringement of DUALi's intellectual property rights.

DUALi Inc. reserves the right to make changes to its applications or services to discontinue any application or service at any time without notice. DUALi provides customer assistance in various technical areas, but does not have full access to data concerning the use and applications of customer's products.

Therefore, DUALi assumes no liability and is not responsible for customer applications or software design or performance relating to systems or applications incorporating DUALi products. In addition, DUALi assumes no liability and is not responsible for infringement of patents and/or any other intellectual or industrial property rights of third parties, which may result from assistance provided by DUALi.

Composition of the information in this manual has been done to the best of our knowledge. DUALi does not guarantee the correctness and completeness of the details given in this manual and may not be held liable for damages ensuing from incorrect or incomplete information. Since, despite all our efforts, errors may not be completely avoided, we are always grateful for your useful tips.

We have our development center in South Korea to provide technical support. For any technical assistance can contact our technical support team as below;

Tel: +82 31 213 0074

e-mail : duali@duali.com

Revision History

- 2011.09.22(Ver. 1.00) : First Release (HTY)
- 2011.09.30(Ver. 1.01) : Separate Protocol between Windows and Linux (HTY)
Add Escape Command Enable process on Windows(HTY)
- 2011.10.05(Ver. 1.02) : MiFare Ultra Light Card Write and FeliCa Function added,
Type A/B ATR Modified (HTY)
- 2011.10.26(Ver. 1.03) : LoadKey Key Type Add (HTY)
- 2011.12.29(Ver. 1.04) : Mifare Ultra Light function for PS/CS Add (HTY)
- 2012.02.24(Ver. 1.10) : Add FeliCa Batch Command(HTY)
Add commands for non-ISO cards.
Add method to call vendor commands (Chapter 9)
- 2012.03.08(Ver. 1.11) : Change document name from "CCID PROTOCOL ...". (HTY)
- 2012.03.22(Ver. 1.12) : Correct miss-typing of RF parameter command from 0x10(HTY)
- 2012.04.06(Ver. 1.13) : Save MIFARE key to nonvolatile memory
Add more Card(Tag) Types to ATR(HTY)
- 2012.09.11(Ver. 1.14) : Save MIFARE key to nonvolatile memory
- 2012.10.23(Ver. 1.15) : Location means key location(HTY)
- 2012.12.11(Ver. 1.16) : Add explanation for MIFARE PLUS in chapter 5.6(HTY)
- 2013.02.06(Ver. 1.17) : Add explanation for control Buzzer/LED in chapter 6.5(SYH)
- 2013.03.22(Ver. 1.18): Saving MIFARE key to volatile memory was added
- 2013.08.06(Ver. 2.00): Support 3 ATR types and Add PCSC MIFARE commands
- 2013.09.03(Ver. 2.01): Add example of MIFARE Ultra Light Read/Write
- 2013.09.11(Ver. 2.02): Change TD1 of ATR for MIFARE card to 0x80 from 0x81
- 2014.01.28(Ver. 2.03): Add function for PS/CS checking card type change(chapter 7.8)
- 2014.02.13(Ver. 2.04): PC/SC mode(mode1) ATR change to meet specification
- 2014.04.03(Ver. 2.05): Add explain to be quiet(not beep) when card is detected
- 2014.04.11(Ver. 2.06): Error fix for PS/CS checking card type change(chapter 7.8)
- 2014.12.16(Ver. 2.07): Add USB PID & VID of DRAGON
- 2015.01.23(Ver. 2.08): Add Linux Driver install process
- 2015.07.27(Ver. 2.09): Add explanation for Buzzer control with Tone and time
- 2016.04.04(Ver. 2.10): Add Authentication
- 2016.08.09(Ver. 2.11): Add explanation for use key(chapter 6.5)
- 2017.08.08(Ver. 2.12): ISO-14443 vendor mode ATR TA3 explanation change (chapter 3.4)
- 2017.08.14(Ver. 2.13): Add ATR mode 3 for memory card like MIFARE (chapter 3.7)

- 2018.01.09(Ver. 2.14): Add USB PID of DE-ABCM6 (chapter 11.2)

CONTENTS

1.	INTRODUCTION	7
2.	COMMUNICATION INTERFACE	8
2.1	COMMUNICATION SCHEME OF PC/SC MODE	8
3.	ATR GENERATION.....	9
3.1	ATR MODE	9
3.2	ATR FORMAT FOR MIFARE, ISO15693 AND FEILCA (MODE 0, VENDOR)	10
3.3	ATR FORMAT FOR MIFARE, ISO15693 AND NFC BARCODE (MODE 1, PC/SC)	10
3.4	ATR FORMAT FOR ISO14443 SMART CARD (MODE 0 & 3, VENDOR)	12
3.5	ATR FORMAT FOR ISO14443 (MODE 1, PC/SC)	13
3.6	ATR FORMAT ALL CARDS (MODE 2, CCID)	14
3.7	ATR FORMAT FOR MIFARE, ISO15693 AND FEILCA (MODE 3, VENDOR)	14
4.	APDU COMMANDS FORMAT	16
4.1	EXCEPT LE	16
4.2	INCLUDE LE.....	16
4.3	EXTENDED LE.....	16
5.	APDU COMMANDS FOR NON-ISO CARDS (DUALI COMMAND)	17
5.1	LOAD KEYS FOR MIFARE(NONVOLATILE MEMORY).....	17
5.2	READ BLOCK FOR MIFARE.....	18
5.3	WRITE BLOCK FOR MIFARE	19
5.4	INCREMENT AND TRANSFER VALUE BLOCK FOR MIFARE.....	20
5.5	DECREMENT AND TRANSFER VALUE BLOCK FOR MIFARE.....	21
5.6	NON-ISO COMMAND(MIFARE DESFIRE AND PLUS)	22
5.7	TYPE-A TRANSPARENT COMMAND	23
5.8	TYPE-B TRANSPARENT COMMAND.....	25
5.9	READ BLOCK FOR MIFARE.....	26
5.10	WRITE BLOCK FOR MIFARE	27
5.11	INCREMENT AND TRANSFER VALUE BLOCK FOR MIFARE.....	28
5.12	DECREMENT AND TRANSFER VALUE BLOCK FOR MIFARE.....	29
5.13	NON-ISO COMMAND(MIFARE DESFIRE AND PLUS)	30
5.14	TYPE-A TRANSPARENT COMMAND	31
5.15	TYPE-B TRANSPARENT COMMAND.....	33
6.	APDU COMMANDS FOR NON-ISO CARDS (PC/SC COMMAND).....	34
6.1	GET UID AND HISTORICAL DATA	34

6.2	LOAD KEYS FOR MIFARE	34
6.3	READ BLOCK FOR MIFARE.....	36
6.4	WRITE BLOCK FOR MIFARE	36
6.5	GENERAL AUTHENTICATE COMMAND.....	37
6.6	OTHER COMMANDS FOR MIFARE	38
7.	APDU COMMANDS FOR READER CONTROL.....	39
7.1	GET READER VERSION	39
7.2	CONTROL AUTO-POLLING	40
7.3	CONTROL RF AND CONTACT	41
7.4	GET CARD STATUS	42
7.5	CONTROL BUZZER/LED.....	43
7.6	RF PARAMETER CONTROL	44
7.7	FLASH MEMORY READ/WRITE	47
7.8	CHECKING CARD TYPES SET	48
8.	APDU COMMANDS FOR ICC OR SAM CONTROL	49
8.1	ICC POWER ON	49
8.2	ICC TRANCEIVE	50
8.3	ICC POWER OFF	51
9.	APDU COMMANDS FOR FELICA CARD CONTROL	52
9.1	FELICA SAM AUTHENTICATION	52
9.2	FELICA MUTUAL AUTHENTICATION.....	53
9.3	FELICA MUTUAL AUTHENTICATION RWSAM.....	54
9.4	FELICA COMMAND.....	55
9.5	FELICA TRANSPARENT	56
9.6	FELICA EXAMPLE.....	57
10.	VENDOR COMMANDS.....	59
10.1	COMMAND CALL METHOD	59
11.	CCID CONTROL CODE	60
11.1	CONTROL CODE TABLE	60
11.2	CONTROL CODE USAGE ENABLE ON LINUX	62
11.3	CONTROL CODE USAGE ENABLE ON WINDOWS XP	65
12.	STATUS(RESPONSE) CODE DEFINITION.....	67

1. Introduction

This document defines the USB PC/SC (include CCID) protocol for special function between DUALi's readers and a host computer.

DUALi's readers and modules support ISO7816, ISO14443 type A/B, ISO18092, my-d™, Mifare®, DESFire® FeliCa™, ISO15693 and I-CODE cards and high communication speed depending on the chip used in the reader. But there is some limit when we use it as USB CCID reader.

This document is dedicated for all readers and modules. So, when you use some readers of DUALi, those readers have a possibility to return code UNKNOWN COMMAND ERROR(23, 0x17), it means your reader or module doesn't support that command.

my-d™ is registered trademarks of Infineon Technologies AG

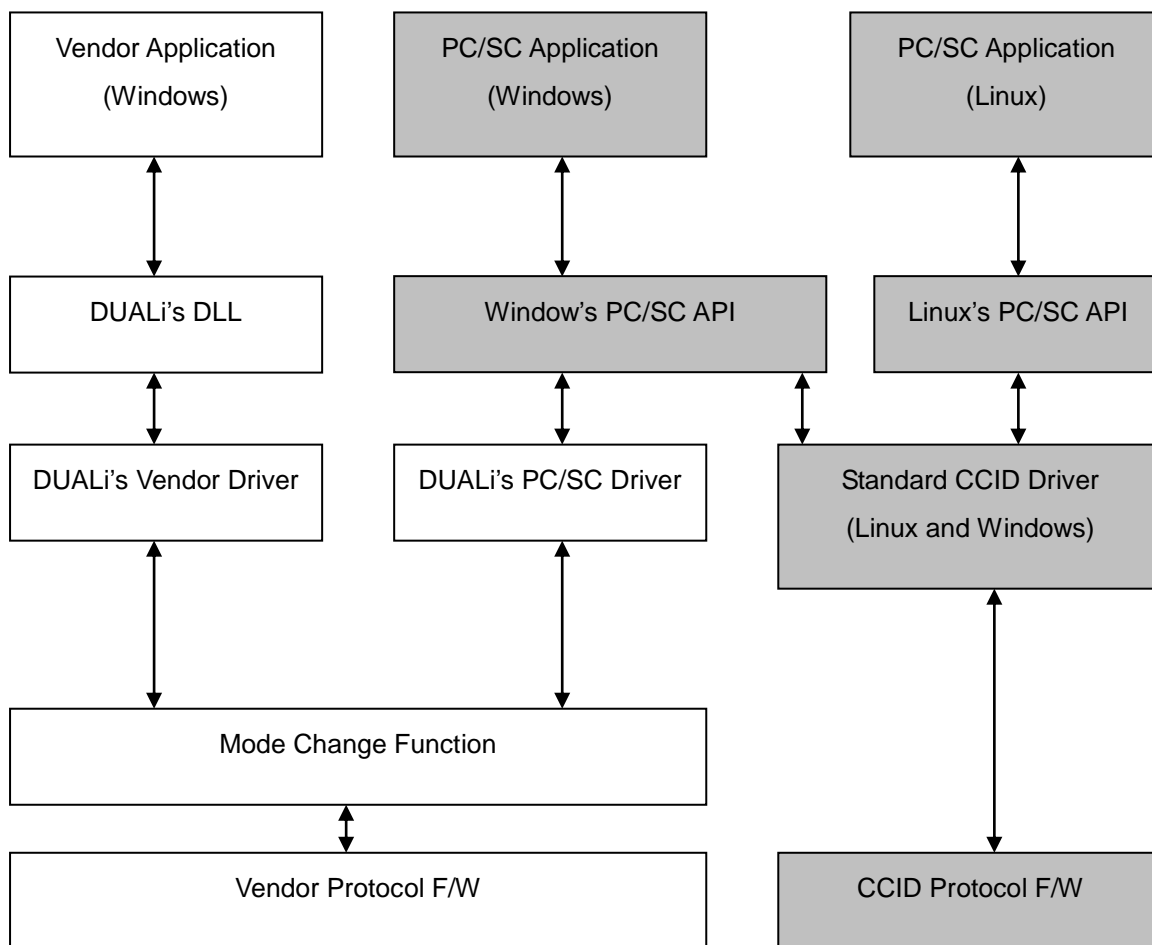
FeliCa™ is registered trademark of SONY Corporation

Mifare® and DesFire® are registered trademarks of NXP Semiconductors

2. Communication Interface

2.1 Communication scheme of PC/SC Mode

DUALi's reader supports 3 protocols (proprietary or PC/SC and CCID) depends on firmware. This document explains protocol of PC/SC mode include CCID. In this mode, reader has a compatibility with standard Microsoft API. It supports card detection/removal, auto-selection and ATR generation. Reader can be controlled by pseudo APDU or Control-Code (SCardControl function). You can refer this document if you use **USB PC/SC driver later than "2012-02-02"** version. For more information about this scheme, please contact DUALi Technical Support Team.



3. ATR Generation

If the reader detects a contactless card, an ATR will be generated and sent to the driver for identification of the card. It bypasses ATR for contact card.

3.1 ATR Mode

DUALi readers support 4 types of ATR. Mode 0 and 3 is original DUALi format. Mode 1 is PC/SC format. And mode 2 is CCID recommended format ATR without historical bytes. In mode 2 Windows will never ask to install Smart Card driver.

You can change ATR format with command 0x15.

Pseudo APDU format (7 byte) – Mode Set					
CLASS	INS	P1	P2	LC	DATA IN
0xFE	0x15	0xFE	0xFE	0x02	0x04, 0x00 (mode 0, Vendor) 0x04, 0x01 (mode 1, PC/SC) 0x04, 0x02 (mode 2, CCID) 0x04, 0x03 (mode 3, Original Vendor, Optional)

Pseudo APDU format (6 byte) – Mode Request					
CLASS	INS	P1	P2	LC	DATA IN
0xFE	0x15	0xFE	0xFE	0x01	0x04

3.2 ATR Format for MIFARE, ISO15693 and FeilCa (mode 0, Vendor)

Byte	Value(HEX)	Designation	Description
0	3B	TS	
1	8n 4 < n < 16	T0	Higher nibble 8 means : only TD1 is following Lower nibble n means : length of historical bytes(Tk)
2	80 Or 0x81	TD1	Higher nibble 8 means : only TD2 is following 0x80 is default and 0x81 is optional to maintain original format, depends on firmware.
3	01	TD2	Higher nibble 0 means : no data is following Lower nibble 1 means : T=1 supported
4 ~ 3+n	0xHH	Tk	Byte for card type 0xF0 or 0x01 - Mifare card 0xFD or 0x02 – ISO15693 card 0xFC or 0x03 – FeliCa card 0xF1 - Topaz, NFC Type1 Tag 0xFB – NFC Barcode(KOVIO)
	UID[5~]		UID for MiFare Cascade level1 : 4byte Cascade level2 : 7byte Cascade level3 : 10byte UID for FeilCa & ISO15693 : 8byte UID for NFC Barcode : 14byte
	0xHH		MIFARE Type 0x30 : MIFARE 1K, 0x31 : MIFARE Ultra Light 0x32 : MIFARE 4K, 0x33 : MIFARE Mini 0x34 : MIFARE PLUS 2K 0x35 : MIFARE PLUS 4K
4+n	0xHH	TCK	Exclusive-oring of all the bytes T0 to TK

3.3 ATR Format for MIFARE, ISO15693 and NFC Barcode (mode 1, PC/SC)

Byte	Value(HEX)	Desig	Description
------	------------	-------	-------------

		nation	
0	3B	TS	
1	8F	T0	Higher nibble 8 means : only TD1 is following Lower nibble F means : length of historical bytes(Tk)
2	80	TD1	Higher nibble 8 means : only TD2 is following Lower nibble 0 means : T=0 not supported
3	01	TD2	Higher nibble 0 means : no data is following Lower nibble 1 means : T=1 supported
4 ~ 18	80 4F 0C A0 00 00 03 06 03 or 01 or 09 or 11 00 HH 00 00 00 00	Tk	Category indicator byte Application identifier Presence indicator length RID 5 bytes Byte for Standard (03:MIFARE, 11: FeliCa, 09:15693, 01:NFC Barcode(KOVIO) and Topaz(NFC T1Tag) Bytes for Card Name(00HH), 2 bytes 0001 : MIFARE 1K, 0002 : MIFARE 4K 0003 : ULTRA LIGHT, 0026 : MIFARE Mini 0036 : MIFARE PLUS 2K 0037 : MIFARE PLUS 4K 0016 : ISO 15693 Tag 0030 : Topaz, NFC T1Tag 003B : FeliCa 0000 : Others (include NFC Barcode) RFU 4 bytes
19	0xHH	TCK	Exclusive-oring of all the bytes T0 to TK

3.4 ATR Format for ISO14443 Smart Card (mode 0 & 3, Vendor)

Byte	Value (HEX)	Designation	Description
0	3B	TS	Fixed value
1	Fn 4 < n < 16	T0	Higher nibble F means : TA1, TB1, TC1, TD1 is following Lower nibble n means : length of historical bytes(Tk)
2	91	TA1	Fixed value
3	00	TB1	Fixed value
4	FF	TC1	Fixed value
5	91	TD1	Higher nibble 9 means : TA2, TD2 is following Lower nibble 1 means : T=0 not supported
6	81	TA2	Fixed value
7	71	TD2	Higher nibble 9 means : TA3, TB3, TC3 is following Lower nibble 1 means : T=1 supported
8	xx	TA3	FSCI in case of T=1
9	40	TB3	Higher nibble 4 means : BWI Lower nibble 0 means : CWI
10	00	TC3	Error detection code : using LRC
11 ~ 10+n	0xHH	Tk	Historical byte Tk[0] 'A'(0x41) or 0x0A - type A Historical Data + PUPI[4~] Tk[0] 'B'(0x42) or 0x0B - type B App data[4]+protocol info[3] +80+ PUPI[4]
11+n	0xHH	TCK	Exclusive-OR(XOR) of all the bytes T0 to TK

3.5 ATR Format for ISO14443 (mode 1, PC/SC)

Byte	Value (HEX)	Designation	Description
0	3B	TS	Fixed value
1	8n 1 < n < 15	T0	Higher nibble F means : TD1 is following Lower nibble n means : length of historical bytes(Tk)
2	80	TD1	Fixed value
3	01	TD2	Fixed value
4 ~ 3+n	0xHH	Tk	Historical byte <i>Type A: ATS, Reply of the RATS(4~15 bytes)</i> <i>Type B (8 bytes):</i> - <i>Byte 0~3: Application data from ATQB,</i> - <i>Byte 4~6: protocol info byte from ATQB,</i> - <i>Byte 7:</i> <i>high nibble - MBLI from ATTRIB,</i> <i>low nibble - 0</i>
4+n	0xHH	TCK	Exclusive-OR(XOR) of all the bytes T0 to TK

3.6 ATR Format all Cards (mode 2, CCID)

Byte	Value (HEX)	Designation	Description
0	3B	TS	Fixed value
1	F0	T0	Higher nibble F means : TA1, TB1, TC1, TD1 is following Lower nibble n means : length of historical bytes(Tk)
2	18	TA1	Fixed value
3	00	TB1	Fixed value
4	02	TC1	Fixed value
5	C1	TD1	Higher nibble C means : TA2, TD2 is following Lower nibble 1 means : T=1 supported
6	05	TC2	Fixed value
7	B1	TD2	Higher nibble B means : TA3, TC3, TD3 is following Lower nibble 1 means : T=1 supported
8	40	TA3	Fixed value
9	38	TC3	Fixed value
10	1F	TD3	Higher nibble 1 means : TA4 is following
11	03	TA4	Fixed value
12	0xHH	TCK	Exclusive-OR(XOR) of all the bytes T0 to TK

3.7 ATR Format for MIFARE, ISO15693 and FeilCa (mode 3, Optional Vendor)

Byte	Value(HEX)	Designation	Description
0	3B	TS	
1	8n 4 < n < 16	T0	Higher nibble 8 means : only TD1 is following Lower nibble n means : length of historical bytes(Tk)
2	80	TD1	Higher nibble 8 means : only TD2 is following Lower nibble 0 means : T=1 supported
3	01	TD2	Higher nibble 0 means : no data is following Lower nibble 1 means : T=1 supported

4 ~ 3+n	0xHH	Tk	Byte for card type 0x01 - Mifare card 0x02 – ISO15693 or other cards 0x03 – FeliCa card
	UID[5~]		UID for MiFare Cascade level1 : 4byte Cascade level2 : 7byte Cascade level3 : 10byte UID for FeilCa & ISO15693 : 8byte UID for NFC Barcode : 14byte
4+n	0xHH	TCK	Exclusive-oring of all the bytes T0 to TK

Example : ATR for Mifare = {3B 85 81 01 01 D4 82 89 AE 75}

ATR						
TS	T0	TD1	TD2	Card byte	UID	TCK
3B	85	81	01	01(Mifare)	D4 82 89 AE	75

4. APDU Commands Format

4.1 Except Le

In Windows, depending on driver version, you must not send last Le byte. This is command example of Pseudo-APDU.

Pseudo APDU format (N+5 byte)					
CLASS	INS	P1	P2	LC	DATA IN
0xFF Or 0xFD Or 0xFE		0xFE	0xFE	N	Data[0..N-1]

4.2 Include Le

Normally you need to send last Le byte. This is command example of Pseudo-APDU.

Pseudo APDU format (N+6 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le
0xFF 0xFD Or 0xFE		0xFE	0xFE	N	Data[0..N-1]	0x00

4.3 Extended Le

Data can be more than 255 when use extended Lc. This is command example of Pseudo-APDU.

Pseudo APDU format (N+7 byte)					
CLASS	INS	P1	P2	LC[3]	DATA IN
0xFF Or 0xFD Or 0xFE		0xFE	0xFE	0x00 N/256 N%256	Data[0..N-1]

5. APDU Commands for NON-ISO Cards (DUALi command)

5.1 Load keys for MIFARE(nonvolatile memory)

The “Load keys command” will load authentication keys into the reader. The load keys are used to authenticate the particular sector of the MiFare memory card. It saves key at nonvolatile and volatile memory location depend on location.

Load keys APDU format (12byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFD	0x2F	Key type	Locations	0x06	Key[6]	0x00

Key type :

-0x00 : Key is used as type A, -0x04 : Key is used as type B

Locations : 0x00 ~0x0F, 0x10~0x1F

-0x00~0x0F : nonvolatile memory

-0x10 : volatile memory

Load keys response format (3byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

e.g.

Send: FD2F000006 112233445566 00

Receive: 00 9000

5.2 Read block for MIFARE

The “Read block command” is used for retrieving a data block from the MiFare memory card.

Read block APDU format (7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFD	0x35	Key type	Locations	0x01	Block no	0x00

Key type :

-0x00 : Key is used as type A, -0x04 : Key is used as type B

Locations : key location, 0x00 ~0x0F, 0xFF(for Ultra Light)

Block no : The block to be accessed

Read block response code (19 byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Block data[16]	0x90	0x00

Block data : retrieved data from accessed block.

e.g.

Send: FD35000001 01 00 : read block 1 with A-Key, Number 0

Receive: 00 11223344556677889900112233445566 9000

e.g. (Ultra Light)

Send: FD3500FF01 01 00 : read block 1 of Ultra Light

Receive: 00 F1D30280A04800001020304066666666 9000

5.3 Write block for MIFARE

The “Write block command” is used for writing a data block into the Mifare memory card.

Write block APDU format (23 byte)							
CLASS	INS	P1	P2	LC	DATA IN	DATA IN	Le(option)
0xFD	0x37	Key type	Locations	0x11	Block no	Block data[16]	0x00

Key type :

-0x00: Key is used as type A, -0x04: Key is used as type B

Locations: key location, 0x00 ~0x0F, 0xFF(for Ultra Light)

Block no: The block to be accessed

Block data: The data to be written into the accessed block.(16byte or 4byte[Ultra Light])

Write block response format (3byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

e.g.

Send: FD37000011 01 11223344556677889900112233445566 00

Receive: 00 9000

e.g. (Ultra Light)

Send: FD37000011 01 11223344556677889900112233445566 00

Receive: 00 9000

5.4 Increment and transfer value block for MIFARE

The “Increment and transfer value block command” is used for increment the value block by the specified value in command and transfer this value to the particular block.

Increment and transfer value block APDU format (12 byte)								
CLASS	INS	P1	P2	LC	DATA IN	DATA IN	DATA IN	Le(option)
0xFD	0x33	Key type	Locations	0x06	Inc Block no	Value[4]	Trs Block no	0x00

Key type :

-0x00: Key is used as type A, -0x04: Key is used as type B

Locations: key location, 0x00 ~0x0F

Inc Block no: The block to be increased by the Value[4].

Value: The value used for value manipulation. This variable is an unsigned long integer.

e.g. : if 1, 0x01 0x00 0x00 0x00
 if 255, 0xFF 0x00 0x00 0x00

Trs Block no : The block used to transfer the manipulated value.

Increment and transfer value block response format (3byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

5.5 Decrement and transfer value block for MIFARE

The “Decrement and transfer value block command” is used for decrement the value block by the specified value in command and transfer this value to the particular block.

Decrement and transfer value block APDU format (12 byte)								
CLASS	INS	P1	P2	LC	DATA IN	DATA IN	DATA IN	Le(option)
0xFD	0x34	Key type	Locations	0x06	Dec Block no	Value[4]	Trs Block no	0x00

Key type:

-0x00: Key is used as type A, -0x04: Key is used as type B

Locations : key location, 0x00 ~0x0F

Dec Block no: The block to be Decreased by the Value[4].

Value: The value used for value manipulation. This variable is a unsigned long integer.

e.g. : if 1, 0x01 0x00 0x00 0x00
 if 255, 0xFF 0x00 0x00 0x00

Trs Block no: The block used to transfer the manipulated value.

Decrement and transfer value block response format (3byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

5.6 NON-ISO COMMAND(MIFARE DESFIRE and PLUS)

You can use this command when you want to control DESFIRE (EV1) or MIFARE PLUS card in non-ISO mode. Reader controls PCB and CID automatically.

Reader stops checking card existence when this command was executed. So, user must send 'card checking enable command(0x81)' after all transaction.

Non-ISO Command APDU format (n+7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x63	0xFE	0xFE	N+1	Data[0..n]	0x00

Non-ISO Command response format (n+4byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Data[0..n]	0x90	0x00

e.g. (DESFIRE)

Send: FE63FEFE01 60 00 : Get Version1
Receive: 00 AF04010101001605 9000

Send: FE63FEFE01 AF 00 : Get Version2
Receive:.....

Send: FE63FEFE01 AF 00 : Get Version3
Receive:.....

Send: FE63FEFE04 5A010000 00 : Select Application
Receive:00 00 9000

Send: FE81FEFE01 00 00 : **Start Card Check again**
Receive:00 9000

e.g. (PLUS, refer chapter 5.7 first)

Send: FE63FEFE13 A800900000000000000000000000000000 00 : Write
Perso(Write personalization of AES Kets and all blocks)
Receive: 00 90 9000

5.7 Type-A Transparent COMMAND

You can use this command when you want to send any commands to all type-A cards. All data include PCB, CID and Rx-Timeout must be controlled by user. Refer to '8.5 FeliCa Transparent' for more data of Timeout.

User must disable card existence checking first. And, user must send 'card checking enable command(0x81)' after all transaction.

Type-A Transparent APDU format (n+7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x41	0xFE	0xFE	N+1	Data[0..n]	0x00

Data: include PCB, CID, CMD, DATA and Rx Timeout,

Type-A Transparent response format (n+4byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Data[0..n]	0x90	0x00

e.g. (MIFARE PLUS)

Send: FE81FEFE01 FF 00 : **Disable card existence checking**
Receive: 00 9000

Send: FE11FEFE00 00 : RF OFF, Refer to "9.Vendor Command".
Receive: 00 9000

Send: FE10FEFE00 00 : RF ON, Refer to "9.Vendor Command".
Receive: 00 9000

Send: FE21FEFE00 00 : REQA, Refer to "9.Vendor Command".
Receive: 00 4403 9000

Send: FE3DFEFE00 00 : ANTICOLLISION-SELECT, Refer to "9.Vendor Command".
Receive: 00 20 04472AA1431C80 9000

Send: FE41FEFE03 E080 10 00 : RATS, Timeout=16(0x10)millisecond.
Receive: 00 20 04472AA1431C80 9000

Send: FE41FEFE04 0A0060 10 00 : Get Version1 PCB=0x0A, CID=0x00, CMD=0xA8,
Rx Timeout=0x10.

Receive: 00 0A00AF04010101001605 9000

:

Send: FE81FEFE01 00 00 : **Start Card Check again**

Receive:00 9000

5.8 Type-B Transparent COMMAND

You can use this command when you want to send any commands to all type-B cards. All data include PCB, CID and Rx-Timeout must be controlled by user. Refer to '8.5 FeliCa Transparent' for more data of Timeout.

User must disable card existence checking first. And, user must send 'card checking enable command(0x81)' after all transaction.

Type-B Transparent APDU format (n+7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x60	0xFE	0xFE	N+1	Data[0..n]	0x00

Data: include PCB, CID, CMD, DATA and Timeout.

Type-B Transparent response format (n+4byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Data[0..n]	0x90	0x00

e.g.

Send: FE81FEFE01 FF 00 : **Disable card existence checking**
Receive:00 9000

Send: FE11FEFE00 00 : RF OFF, Refer to "9.Vendor Command".
Receive:00 9000

Send: FE10FEFE00 00 : RF ON, Refer to "9.Vendor Command".
Receive:00 9000

Send: FE60FEFE04 05000005 00 : Write Perso REQB command, Rx Timeout=0x05
Receive: 00 507505717500014B40008181 9000

:

Send: FE81FEFE01 00 00 : **Start Card Check again**
Receive:00 9000

5.9 Read block for MIFARE

The “Read block command” is used for retrieving a data block from the MiFare memory card.

Read block APDU format (7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFD	0x35	Key type	Locations	0x01	Block no	0x00

Key type :

-0x00 : Key is used as type A, -0x04 : Key is used as type B

Locations : key location, 0x00 ~0x0F, 0xFF(for Ultra Light)

Block no : The block to be accessed

Read block response code (19 byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Block data[16]	0x90	0x00

Block data : retrieved data from accessed block.

e.g.

Send: FD35000001 01 00 : read block 1 with A-Key, Number 0

Receive: 00 11223344556677889900112233445566 9000

5.10 Write block for MIFARE

The “Write block command” is used for writing a data block into the Mifare memory card.

Write block APDU format (23 byte)							
CLASS	INS	P1	P2	LC	DATA IN	DATA IN	Le(option)
0xFD	0x37	Key type	Locations	0x11	Block no	Block data[16]	0x00

Key type :

-0x00: Key is used as type A, -0x04: Key is used as type B

Locations: key location, 0x00 ~0x0F, 0xFF(for Ultra Light)

Block no: The block to be accessed

Block data: The data to be written into the accessed block.(16byte or 4byte[Ultra Light])

Write block response format (3byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

e.g.

Send: FD37000011 01 11223344556677889900112233445566 00

Receive: 00 11223344556677889900112233445566 9000

5.11 Increment and transfer value block for MIFARE

The “Increment and transfer value block command” is used for increment the value block by the specified value in command and transfer this value to the particular block.

Increment and transfer value block APDU format (12 byte)								
CLASS	INS	P1	P2	LC	DATA IN	DATA IN	DATA IN	Le(option)
0xFD	0x33	Key type	Locations	0x06	Inc Block no	Value[4]	Trs Block no	0x00

Key type :

-0x00: Key is used as type A, -0x04: Key is used as type B

Locations: key location, 0x00 ~0x0F

Inc Block no: The block to be increased by the Value[4].

Value: The value used for value manipulation. This variable is an unsigned long integer.

e.g. : if 1, 0x01 0x00 0x00 0x00
 if 255, 0xFF 0x00 0x00 0x00

Trs Block no : The block used to transfer the manipulated value.

Increment and transfer value block response format (3byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

5.12 Decrement and transfer value block for MIFARE

The “Decrement and transfer value block command” is used for decrement the value block by the specified value in command and transfer this value to the particular block.

Decrement and transfer value block APDU format (12 byte)								
CLASS	INS	P1	P2	LC	DATA IN	DATA IN	DATA IN	Le(option)
0xFD	0x34	Key type	Locations	0x06	Dec Block no	Value[4]	Trs Block no	0x00

Key type:

-0x00: Key is used as type A, -0x04: Key is used as type B

Locations : key location, 0x00 ~0x0F

Dec Block no: The block to be Decreased by the Value[4].

Value: The value used for value manipulation. This variable is a unsigned long integer.

e.g. : if 1, 0x01 0x00 0x00 0x00
if 255, 0xFF 0x00 0x00 0x00

Trs Block no: The block used to transfer the manipulated value.

Decrement and transfer value block response format (3byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

5.13 NON-ISO COMMAND(MIFARE DESFIRE and PLUS)

You can use this command when you want to control DESFIRE (EV1) or MIFARE PLUS card in non-ISO mode. Reader controls PCB and CID automatically.

Reader stops checking card existence when this command was executed. So, user must send 'card checking enable command(0x81)' after all transaction.

Non-ISO Command APDU format (n+7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x63	0xFE	0xFE	N+1	Data[0..n]	0x00

Non-ISO Command response format (n+4byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Data[0..n]	0x90	0x00

e.g. (DESFIRE)

Send: FE63FEFE01 60 00 : Get Version1
Receive: 00 AF04010101001605 9000

Send: FE63FEFE01 AF 00 : Get Version2
Receive:.....

Send: FE63FEFE01 AF 00 : Get Version3
Receive:.....

Send: FE63FEFE04 5A010000 00 : Select Application
Receive:00 00 9000

Send: FE81FEFE01 00 00 : **Start Card Check again**
Receive:00 9000

e.g. (PLUS, refer chapter 5.7 first)

Send: FE63FEFE13 A800900000000000000000000000000000 00 : Write
Perso(Write personalization of AES Kets and all blocks)
Receive: 00 90 9000

5.14 Type-A Transparent COMMAND

You can use this command when you want to send any commands to all type-A cards. All data include PCB, CID and Rx-Timeout must be controlled by user. Refer to '8.5 FeliCa Transparent' for more data of Timeout.

User must disable card existence checking first. And, user must send 'card checking enable command(0x81)' after all transaction.

Type-A Transparent APDU format (n+7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x41	0xFE	0xFE	N+1	Data[0..n]	0x00

Data: include PCB, CID, CMD, DATA and Rx Timeout,

Type-A Transparent response format (n+4byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Data[0..n]	0x90	0x00

e.g. (MIFARE PLUS)

Send: FE81FEFE01 FF 00 : **Disable card existence checking**
Receive: 00 9000

Send: FE11FEFE00 00 : RF OFF, Refer to "9.Vendor Command".
Receive: 00 9000

Send: FE10FEFE00 00 : RF ON, Refer to "9.Vendor Command".
Receive: 00 9000

Send: FE21FEFE00 00 : REQA, Refer to "9.Vendor Command".
Receive: 00 4403 9000

Send: FE3DFEFE00 00 : ANTICOLLISION-SELECT, Refer to "9.Vendor Command".
Receive: 00 20 04472AA1431C80 9000

Send: FE41FEFE03 E080 10 00 : RATS, Timeout=16(0x10)millisecond.
Receive: 00 20 04472AA1431C80 9000

Send: FE41FEFE04 0A0060 10 00 : Get Version1 PCB=0x0A, CID=0x00, CMD=0xA8,
Rx Timeout=0x10.

Receive: 00 0A00AF04010101001605 9000

:

Send: FE81FEFE01 00 00 : **Start Card Check again**

Receive:00 9000

5.15 Type-B Transparent COMMAND

You can use this command when you want to send any commands to all type-B cards. All data include PCB, CID and Rx-Timeout must be controlled by user. Refer to '8.5 FeliCa Transparent' for more data of Timeout.

User must disable card existence checking first. And, user must send 'card checking enable command(0x81)' after all transaction.

Type-B Transparent APDU format (n+7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x60	0xFE	0xFE	N+1	Data[0..n]	0x00

Data: include PCB, CID, CMD, DATA and Timeout.

Type-B Transparent response format (n+4byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Data[0..n]	0x90	0x00

e.g.

Send: FE81FEFE01 FF 00 : **Disable card existence checking**
Receive: 00 9000

Send: FE11FEFE00 00 : RF OFF, Refer to "9.Vendor Command".
Receive: 00 9000

Send: FE10FEFE00 00 : RF ON, Refer to "9.Vendor Command".
Receive: 00 9000

Send: FE60FEFE04 05000005 00 : Write Perso REQB command, Rx Timeout=0x05
Receive: 00 507505717500014B40008181 9000

:

Send: FE81FEFE01 00 00 : **Start Card Check again**
Receive: 00 9000

6. APDU Commands for NON-ISO Cards (PC/SC command)

6.1 Get UID and Historical Data

This command can be used to get full UID of activated card.

Get UID APDU format						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFF	0xCA	0x00 0x01	0x00	0x00	none	0x00

Get UID response		
DATA OUT	SW1	SW2
UID[4~32] Historical Data[1..20]	0x90	0x00

6.2 Load keys for MIFARE

The “Load keys command” will load authentication keys into the reader. The load keys are used to authenticate the particular sector of the MiFare memory card. It saves key at nonvolatile and volatile memory location depend on location.

Load keys APDU format (12byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFF	0x82	Key No	Key Type	0x06	Key[6]	0x00

Key No :

-0x20~0x2F : nonvolatile memory location 0x00 ~ 0x0F

-0x00 : volatile memory

- 0xff : Indicate MIFARE Ultra Light, Key will be ignored

Key Type :

-0x60 : Key is used as type A, -0x61 : Key is used as type B

Load keys response format (3byte)	
SW1	SW2
0x90	0x00

Error Code

	SW1	SW2	Meaning
Error	0x69	0x83	Authentication cannot be done
Error	0x69	0x85	Secured transmission not supported
Error	0x69	0x88	Key number not valid
Error	0x6A	0x05	Authentication cannot be done

6.3 Read block for MIFARE

The “Read block command” is used for retrieving a data block from the MiFare memory card.

Read block APDU format (7 byte)					
CLASS	INS	P1	P2	LC	Le(option)
0xFF	0xB0	0x00	Block No	0x00 Or 0x10 0x04(MIFARE Ultra Light)	0x00

Block No : The block to be accessed

Read block response code		
DATA OUT	SW1	SW2
Block data[16] Block data[4] –Ultra Light	0x90	0x00

Block data : retrieved data from accessed block.

Error Code

	SW1	SW2	Meaning
Error	0x69	0x83	Authentication cannot be done
Error	0x6A	0x05	Authentication cannot be done

6.4 Write block for MIFARE

The “Write block command” is used for writing a data block into the Mifare memory card.

Write block APDU format (23 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFF	0xD6	0x00	Block no	0x10 : MIFARE 0x04 : Ultra Light	Block data[16] Block data[4]	0x00

Block no: The block to be accessed

Block data: The data to be written into the accessed block.(16byte or 4byte[Ultra Light])

Load keys response format (3byte)	
SW1	SW2
0x90	0x00

6.5 General Authenticate Command

The application provides the number of the key used for the authentication. The specific key must be already in the reader.

General Authenticate Command APDU format									
CLASS	INS	P1	P2	LC	Ver	Addr MSB	Addr LSB	Key Type	Key Nr
0xFF	0x86	0x00	0x00	0x05	0x01				

Key Type :

-0x60 : Key is used as type A, -0x61 : Key is used as type B

Key Number :

-0x00~0x0F : nonvolatile memory

-0x10 : volatile memory

General Authenticate Command response	
SW1	SW2
0x90	0x00

Error Code

	SW1	SW2	Meaning
Error	0x69	0x86	Key type not known
Error	0x69	0x88	Key number not valid

6.6 Other Commands for MIFARE

You can use “5. APDU Commands for NON-ISO Cards (DUALi command)” to execute value related commands.

7. APDU Commands For Reader Control

Le(=0x00) can be followed behind DATA when send pseudo APDU.

7.1 Get Reader Version

The “Get reader version command” is used for retrieving the version of reader.

Get reader version APDU format (6 byte)					
CLASS	INS	P1	P2	LC	Le(option)
0xFE	0x16	0xFE	0xFE	0x00	0x00

Get reader version response code (n+3 byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Version data[n]	0x90	0x00

Version data: retrieved version data from reader.

e.g.

Send: FE16FEFE00 00

Receive: 00 44452D3632305F3131303931375F43434944 9000

→ ASCII : DE-620_APP_110917_CCID

7.2 Control auto-polling

The “Control auto-polling command” is used to stop/start auto-polling function in the reader.

When auto-polling is stopped, reader can't support card detection/removal function.

Control auto-polling APDU format (7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x81	0xFE	0xFE	0x01	Data[0]	0x00

Auto-polling stop: Data[0] =0xFF

Auto-polling start: Data[0] =0x00

Control auto-polling response code (3 byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

e.g.

Send: FE81FEFE01 FF 00: auto polling stop


Receive: 00 9000

7.3 Control RF and Contact

This “Control RF and Contact command” is used to enable/disable RF and CONTACT slot in the reader.

Control RF and Contact APDU format (7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x82	0xFE	0xFE	0x01	Data[0]	0x00

Stop checking RF and Disable Contact slot: Data[0] =0x80 (useful when control FeliCa or ISO-15693 card)

 **Reader disconnects previous connection with card if it receives below data. It needs to reconnect to card after sending 0x01, 0x02 and 0x03.**

Enable RF and Disable Contact slot: Data[0] =0x01

Disable RF and Enable Contact Slot: Data[0] =0x02

Enable both RF and Contact slot: Data[0] =0x03

Control RF and Contact response code (3 byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

e.g.

Send: FE82FEFE01 01 00: Enable RF and Disable Contact

Receive: 00 9000

7.4 Get card status

The “Get card status command” is used for retrieving the interface information between reader and card.

Get card status APDU format (6 byte)					
CLASS	INS	P1	P2	LC	Le(option)
0xFE	0x17	0xFE	0xFE	0x00	0x00

Get card status response code (22 byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Status data[19]	0x90	0x00

.Status data : retrieved interface information between reader and card	
Data[0]	Auto-polling (0x01:disable, 0x00:enable)
Data[1]	RF status (0x01:rf on, 0x00:rf off)
Data[2]	Card type (0x41: type A, 0x42: type B, 0x4d: ISO14443 Part3(MiFare), 0x43: FeliCa, 0x49: ISO15693, 0xff: No card)
Data[3]	Reader's Max Bit rates(Tx/Rx max) (0:106 1:212 2:424 3:847) Kbit/s In case of 847 for Tx/Rx, reader returns: 0x03.
Data[4]	Card Bit rates(Tx max) (0:106 1:212 2:424 3:847) Kbit/s
Data[5]	Card Bit rates(Tx current) (0:106 1:212 2:424 3:847) Kbit/s
Data[6]	Card Bit rates(Rx max) (0:106 1:212 2:424 3:847) Kbit/s
Data[7]	Card Bit rates(Rx current) (0:106 1:212 2:424 3:847) Kbit/s
Data[8]	Cascade level (Type A:0x01,0x02,0x03 Type B, FeliCa, ISO15693:0xff)
Data[9~18]	UID (default buffer value:0xff) Ex1] cascade level 1 => Data[9~12] : UID, last bytes(Data[13~18]) will be 0xff. Ex2] cascade level 2 => Data[9~15] : UID, last bytes(Data[16~18]) will be 0xff. Ex3] cascade level 3 => Data[9~18] : UID. Ex4] type B => Data[9~12] : UID, last bytes(Data[13~18]) will be 0xff. Ex5] FeliCa => Data[9~16] : UID, last bytes(Data[17~18]) will be 0xff. Ex6] ISO15693 => Data[9~16] : UID, last bytes(Data[17~18]) will be 0xff.

7.5 Control buzzer/LED

The “Control buzzer/LED command” is used to stop/start buzzer beep or turn on/off LED in the reader.

Control buzzer APDU format (7 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x13	0xFE	0xFE	n	Data[n]	0x00

- Data

LC	Data[0]	Buzzer/LED Control
0x01	0x00	Buzzer Turn On
0x01	0x01	Buzzer Turn Off
0x03	0x02	Buzzer Turn On with Tone(optional) Data[1] : Octave : 0~7 Data[2] : Frequency : 0~11(Do~Si)
0x04	0x03	Buzzer Turn On with Tone and time Data[1] : Octave : 0~7 Data[2] : Frequency : 0~11(Do~Si) Data[3] : time in scale of 10millisecond
0x03	0x10	LED Control (optional) Data[1] : LED position(0x00 or 0x01) Data[2] : 0(off) or 1(on)

Control buzzer response code (3 byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

e.g.

Send: FE13FEFE01 00 00 : Buzzer On

Receive: 00 9000

Send: FE13FEFE03 10 00 01 : 1st LED On

Receive: 00 9000

7.6 RF Parameter Control

This command Changes RF related parameters.

RF Parameter Control APDU format (7 byte)								
CLASS	INS	P1	P2	LC	DATA IN			Le(option)
0xFE	0x1D	0xFE	0xFE	0x01 Or 0x03	Data[0] RF Type	Data[1] Para Type	Data[2..] Para Value	0x00

Data[0] : RF Type

Data[0]	Description
0x10	Read all Parameter Data
0x11	Type A parameter set
0x12	Type B parameter set
0x13	FeliCa parameter set
0x14 ~ 0x18	RFU
0x19	Type A 212/424/848K parameter set(gain,threshold)
0x1B	Special parameter
0x1F	Return to default parameter values

Data[1] : Parameter Type

Data[1]	Description
0x00	Maximum speed set (Tx and Rx Common)
0x01	CwConductance, RF field strength set Strictly prohibited to change
0x02	ModConductance, Modulation set Strictly prohibited to change
0x03	Rx signal gain set Strictly prohibited to change
0x04	Rx threshold level set Strictly prohibited to change
0x05	RF Level set Strictly prohibited to change

Data[2] : Parameter Value

Data[1]	Data[2] Description
0x00	Maximum speed set (Tx and Rx Common) 0(106Kbps), 1(212Kbps), 2(424Kbps), 3(848Kbps)
0x01	CwConductance, RF field strength set 0(Minimum RF strength)~63(Maximum RF strength) Strictly prohibited to change
0x02	ModConductance, Modulation set 0(Maximum Modulation) ~ 63(Minimum Modulation) Strictly prohibited to change
0x03	Rx signal gain set 1(lowest gain)~3(highest gain) Strictly prohibited to change
0x04	Rx threshold level set Refer to PN512 datasheet Strictly prohibited to change
0x05	RF Level set Refer to PN512 datasheet Strictly prohibited to change

Special parameter(Data[0] = 0x1B)

Data[1]	Data[2].. Description
0x00	Data[2] ~Data[5] : Serial communication baudrate(RS-232,TTL), Little endian
0x01	Data[2] :Maximum FWI value set, Card Response Wait Time. 1~9 (less than 300msecond) 10(0.31 second) 11(0.62 second) 12(1.24 second) 13(2.47 second) 14(4.9485 second), Common for all RF type If change, recommend to use 11 or 12
0x02	Data[2] :Reader number set
0x03	15693 option(not use)
0x04	PCSC Buzzer Action Option (0:Act, 1:Quiet)

Response frame when transmitted Data[0] is not 0x10

RF Parameter Control response code (3 byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

Response frame when transmitted Data[0] is 0x10

RF Parameter Control response code (33 byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Data[0..29] Parameters	0x90	0x00

Parameters Format

Data[0...3]	Device Type[Proprietary(0x00) or PCSC]
Data[4...9]	Type-A parameters for 106Kbps, Refer ot Parameter Type Table. (From Data[1] is 0x00 to 0x05)
Data[10...15]	Type-B parameters, refer ot Parameter Value Table
Data[16...21]	FeliCa parameters, refer ot Parameter Value Table
Data[22...27]	RFU
Data[28...33]	RFU
Data[34...39]	RFU
Data[40...45]	RFU
Data[46...51]	RFU
Data[52...57]	Type A 212/424/848K parameters(gain,threshold)
Data[58...63]	RFU
Data[64...67]	Serial communication baudrate(RS-232,TTL), Little endian
Data[68]	Maximum FWI
Data[69]	Reader Address
Data[70]	RFU
Data[71...80]	RFU

e.g.

Send: FE1DFEFE03 1B010B 00 : Type-A, Response Wait for 0.62 second.

Receive: 00 9000

Send: FE1DFEFE03 1B0401 00 : Do not beep when card is detected.

Receive: 00 9000

7.7 Flash Memory Read/Write

This command reads or writes user data from/to reader flash memory. (Maximum size is 128 byte)

Flash Memory Read/Write APDU format (6 byte or more)								
CLASS	INS	P1	P2	LC	DATA IN			Le(option)
0xFE	0x1F	0xFE	0xFE	Data Length+2 Or 0x01	Data[0] 0x00 Or 0xFF	Data[1] offset	Data[2..]	0x00

Flash Write:

Data[0] : 0xFF

Data[1] : Start Position

Data[2 ...] : Data to Write

Flash Read :

Data[0] : 0x00

Data[1] : Start Position

Data[2] : Length to Read

Flash Memory Read/Write response code (3 byte or more)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Data[0..]	0x90	0x00

DATA OUT exists only read flash.

e.g.

Send: FE1FFEFE03 00000A 00 : Read 10 byte from first position.

Receive: 00 FFFFFFFFFFFFFFFFFFFFFFFF 9000

7.8 Checking Card Types Set

This command changes checking card types among Type-A, Type-B, FeliCa, NFC Barcode and ISO-15693 Tag.

Checking Card Types Set APDU format (7 byte)							
CLASS	INS	P1	P2	LC	DATA IN		Le(option)
0xFE	0x15	0xFE	0xFE	0x02	Data[0] 0x05	Data[1] Checking option	0x00

Data[1] :

Bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2~0
Value	0x80	0x40	0x20	0x10	0x08	
Card Type	Type A	Type B	FeliCa	NFC Barcode	ISO 15693	RFU

Checking Card Types Set response code (3 byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

e.g.

Send: FE15FEFE02 05C0 00 : Check only Type-A and Type-B card. (Do not check FeliCa, NFC Barcode and ISO-15693 tags)

Receive: 00 9000

8. APDU Commands For ICC or SAM Control

Le(=0x00) can be followed behind DATA when send pseudo APDU.

8.1 ICC Power On

Reader resets IC or SAM card from appointed slot and receives ATR.

ICC Power On APDU format (8 byte)							
CLASS	INS	P1	P2	LC	DATA IN		Le(option)
0xFE	0xC0	0xFE	0xFE	0x02	Data[0] Slot	Data[1] 0x00	0x00

Slot

0: First Slot (Contact Card slot, SAM1 slot for DE-ABCM)

1: Second Slot (SAM1 slot, SAM2 slot for DE-ABCM)

2: Second Slot (SAM2 slot)

ICC Power On response code (n+3 byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	ATR[n]	0x90	0x00

ATR data: retrieved ATR data from Contact card or SAM.

e.g.

Send: FEC0FEFE02 0100 00 : Second Slot, Power On

Receive: 00 3B6900000806355528083079000 9000

8.2 ICC Transceive

Reader sends APDU+data(n) (5+n bytes) to the ICC or SAM slot and receives data and status words.

Card Transive APDU format (n+7 byte)							
CLASS	INS	P1	P2	LC	Data[0]	Data[1..n]	Le(option)
0xFE	0xC9	0xFE	0xFE	n+1	Slot	APDU+DATA	0x00

Slot

0: First Slot (Contact Card slot, SAM1 slot for DE-ABCM)

1: Second Slot (SAM1 slot, SAM2 slot for DE-ABCM)

2: Second Slot (SAM2 slot)

Card Transive response code (3 byte)			
STATUS CODE	Data	SW2	SW2
Status	Response from card(Data+SW)	0x90	0x00

e.g.

Send: FEC9FEFE08 01 00A4000002 1122 00 : Second Slot, Select File(Applet)

Receive: 00 6A82 9000 : SW=0x6A82, File Not Exist

8.3 ICC Power Off

Reader disables contact card in the slot.

Power Off APDU format (7 byte)						
CLASS	INS	P1	P2	LC	Data[0]	Le(option)
0xFE	0x82	0xFE	0xFE	0x01	Slot	0x00

Slot

0: First Slot (Contact Card slot, SAM1 slot for DE-ABCM)

1: Second Slot (SAM1 slot, SAM2 slot for DE-ABCM)

2: Second Slot (SAM2 slot)

Power Off response code (3 byte)		
STATUS CODE	SW1	SW2
Status	0x90	0x00

e.g.

Send: FEC5FEFE01 01 00 : Second Slot, Power Off

Receive: 00 9000

9. APDU Commands For FeliCa Card Control

Le(=0x00) can be followed behind DATA when send pseudo APDU.

9.1 FeliCa SAM Authentication

This is a authentication between the FeliCa USER and RC-S251(SAM) based on a 3-way mutual authentication mechanism using shared symmetrical keys. User must submit this symmetrical a 24-byte key to RC-S251. The default CBC is all 0x00s. The method of the authentication is not disclosed and executed inside the reader.

Reader stops checking card existence when this command was executed. So, user must send card checking enable command(0x81) after all transaction.

FeliCa SAM Authentication format (39 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x56	0xFE	0xFE	33	ENC Mode Default Key[24] CBC[8]	0x00

ENC Mode: this mode is maintained until finish all transactions.

0: authentication for unencrypted communication

1: authentication for encrypted communication

Default Key : symmetrical a 24-byte TDES key

CBC : default CBC is {0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}

FeliCa SAM AuthTransparent response code (13 byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	IDM[8] IDt[2]	0x90	0x00

9.2 FeliCa Mutual Authentication

This is a authentication between the FeliCa card and RC-S251(SAM). User must submit global key and user key. The method of the mutual authentication is not disclosed and executed inside the reader.

FeliCa SAM Authentication format (26 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x57	0xFE	0xFE	20	Area Code[2] Service Code[2] Global Key[8] User Key[8]	0x00

FeliCa SAM AuthTransparent response code (3 byte)		
STATUS CODE		SW1
		SW2
Status		0x90
		0x00

9.3 FeliCa Mutual Authentication RWSAM

This is a authentication between the FeliCa card and RC-S251(SAM) using the keys stored in SAM. User doesn't need to submit keys but need to submit key codes and versions. The method of the mutual authentication is not disclosed and executed inside the reader.

FeliCa SAM Authentication format (16 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x58	0xFE	0xFE	10	System Code[2] GSK Code and Version[4] USK Code and Version[4]	0x00

FeliCa SAM AuthTransparent response code (3 byte)		
STATUS CODE		SW2
Status	0x90	0x00

9.4 FeliCa Command

This command is used to exchange commands with the FeliCa card after authentication. IDtr is controlled inside the reader. User must submit all other data. User can read or write each block using this command. The method of the process is not disclosed and executed inside the reader.

FeliCa SAM Authentication format (n+9 byte)						
CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0x59	0xFE	0xFE	33	Header[4] Input Data[0..n]	0x00

Header[4]: Dispatcher[0], Reserved[2], Command Code[1]

Input Data: all other data except IDtr

FeliCa SAM AuthTransparent response code (n+4 byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Data[0..n]	0x90	0x00

9.5 FeliCa Transparent

Reader receives data from Host and sends data with computed CRC to the FeliCa card and transfer response from FeliCa card. **To control FeliCa Card, auto card removal detection of reader must be disabled by sending 'Stop Card Checking' command (0x81).**

FeliCa Transparent APDU format (8 byte)							
CLASS	INS	P1	P2	LC	DATA IN		Le(option)
0xFE	0x50	0xFE	0xFE	N	Data[0..N-2] CMD + Data	Data[N-1] TIMEOUT	0x00

TIMEOUT

TOUT Value in hexadecimal	Time Value	FWI
0x02	1.208 msec	2
0x03	2.416 msec	3
0x05	4.832 msec	4
0x0A	9.664 msec	5
0x14	19.32 msec	6
0x27	38.66 msec	7
0x4E	77.3 msec	8
0x9B	154.6 msec	9
0xB0 ~	309.3 msec	10
0xC0 ~	618.6 msec	11
0xD0 ~	1.2371 sec	12
0xE0 ~	2.4742 sec	13
0xF0 ~	4.9485 sec	14
0xFF	9.897 sec	15

FeliCa Transparent response code (n+3 byte)			
STATUS CODE	DATA OUT	SW1	SW2
Status	Data[n]	0x90	0x00

Data[n]: retrieved data from FeliCa card.

9.6 FeliCa Example

FeliCa Read/Write Example

Send: FE56FEFE21
0157A16232E4F97DD6E53463F0B1CD1B1784869C5940CC09120000000000000000
00 : Authenticate SAM

Receive: 00 01010701360DEE080000 9000

Send: FE57FEFE14 00000810 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx 00: Mutual
Authentication
Receive: 00 9000

Send: FE59FEFE190000008A000001800011223344556677889900112233445566
00 : Write Block 0
Receive: 00060015034147F40800006E78E4BD6533D81C55EAE5D6A8E9000

Send: FE59FEFE0B 0000008800000280008001 00 : Read Block 0 and Block 1
Receive: 00 080015034147F408000002 11223344556677889900112233445566
000102030405060708090A0B0C0D0E0F C5087E9200E09BB83B14A74C44 9000

Send: FE81FEFE01 00 00 : **Start Card Check again**
Receive: 00 9000

FeliCa Transparent Example

Send: FE81FEFE01 FF 00 : Stop RF Checking
Receive: 00 9000

Send: FE50FEFE07 0600FFFF0000 64 00 : Polling, TIMEOUT=100mSecond
Receive: 00 1201012200E1E90F03000120220427674EFF 9000

Send: FE50FEFEyy xx....xx 64 00 : FeliCa Command(xx....xx), TO=100mSec

Receive: 00 xx....xx 9000

Send: FE50FEFEyy xx....xx 64 00 : FeliCa Command(xx....xx), TO=100mSec

Receive: 00 xx....xx 9000

Send: FE81FEFE01 00 00 : Start Checking RF and Contact Card

Receive: 00 9000

10. Vendor Commands

10.1 Command Call Method

There are various vendor commands for card and reader control. All the commands are explained at “RW_Protocol_spec_XXXX.pdf”. These all commands could be executed in CCID mode. This is the rule to call vendor commands in CCID mode. Some commands are not supported because reader doesn’t support that command. This restriction caused by small program memory size.

Vendor Protocol(Send)

Name	STX	LEN-H	LEN-L	CMD	Data	LRC
Values	0x02	0xHH	0xHH	0xHH	Data[N-1]	0xHH
Length.	1-byte	N		1-byte	n-byte	1-byte

CCID COMMAND(Send)

CLASS	INS	P1	P2	LC	DATA IN	Le(option)
0xFE	0xHH	0xFE	0xFE	N-1	Data[N-1]	0x00

Vendor Protocol(Receive)

Name	STX	LEN-H	LEN-L	Resp	Data	LRC
Values	0x02	0xHH	0xHH	0xHH	Data[N-1]	0xHH
Length.	1-byte	N		1-byte	n-byte	1-byte

CCID COMMAND(Receive)


Resp	DATA IN	SW
STATUS	Data[0..n]	0x9000

Refer to “11. STATUS(Response) Code Definition” for STATUS information.

11. CCID Control Code

11.1 Control Code Table

Developer can use pseudo APDU or Control-Code when card was activated. But developer can use Control-Code although card was not activated. Command and Data format for Control Codes are same with pseudo APDU. You can refer previous chapters for pseudo APDU for detail data format. You also can refer “RW_Protocol_spec_XXXXX.pdf” for more control codes. All the vendor commands could be supported as control code. Some commands(control code) could not be supported depend on program memory size of reader/module.

Control Code Detail				
Control Code	Data	Description	Reference Chapter	
0x12		Reboot Reader		
0x13	0x00 0x01	Buzzer ON Buzzer OFF	6.5	
0x16		Reader Version	6.1	
0x17		Get Card Status	6.4	
0x1D		RF Parameter Set	6.6	
0x1F	0xFF 0x00	Flash Write Flash Read	6.7	
0x81	0xFF 0x00	RF Polling Stop RF Polling Start	6.2	
0x82	0x80 0x01 0x02 0x03	Stop RF Checking  Reader disconnects previous connection with card if it receives below data. Enable RF, Disable Contact slot Enable Contact Slot, Disable RF Enable RF and Contact Slot	6.3	Bit7(0x80):RF checking Bit0(0x01):RF Enable Bit1(0x02):Contact Enable It needs to connect to card again after sending 0x01,0x02 and 0x03.
0xC0	Data[0] Data[1]	ICC Power ON Data[0] –Slot No Data[1]- Fix to 0x00	7.1	
0xC9	Data[n]	Send APDU and Data to ICC Data[0..4] : APDU Data[5..(n-1)] : Data	7.2	
0xC5	Slot No.	ICC Power Off	7.3	

e.g. 1

Send: 0x8201: Enable RF, Disable Contact Slot

Receive: 0x00: (Response Code)

e.g. 2

Send: 0x16: Reader Firmware Version Request

Receive: 0x00 44452D3632305F3131303931375F43434944 (Response
Code+Data)

→ ASCII : DE-620_APP_110917_CCID

11.2 Control Code Usage Enable on Linux

You don't need to refer this chapter if you use DUALi's CCID driver.

When you want to use standard CCID driver, you need to change option data for 'Control Code Enable' and add reader data (VID, PID and NAME) to '/etc/libccid_Info.plist'.

Driver Install Process:

```
apt-get remove pcscd
apt-get remove libccid
apt-get remove pcsc tools

apt-get install pcscd
apt-get install libccid
apt-get install pcsc tools

apt-get install libpcsclite1
apt-get install libpcsclite-dev
(Edit or copy "libccid_Info.plist" and reboot Linux)
pcscd
pcsc_scan
```

libccid_Info.plist edit:

```
<key>ifdDriverOptions</key>
  <string>0x0001</string>
...
<key>ifdVendorID</key>
<array>
...
<string>0x1DB2</string>
</array>

<key>ifdProductID</key>
<array>
...
<string>0x0802</string>
</array>

<key>ifdFriendlyName</key>
<array>
```

```
...  
<string>DUALi DE-ABCM Combi</string>  
</array>
```

This is list of DUALi's readers.

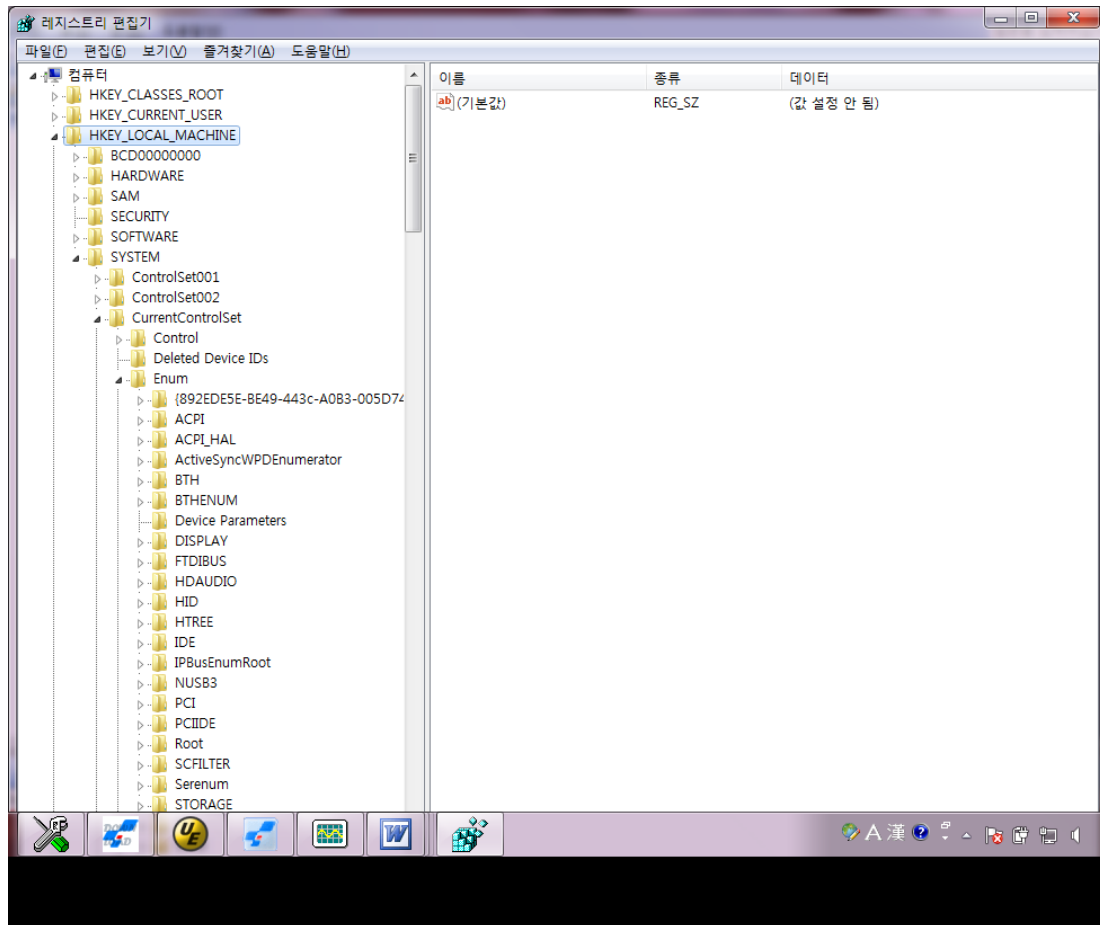
```
//VID  PID  NAME  
0x1DB2:0x0801:DUALi DE-620 Combi Reader  
0x1DB2:0x0802:DUALi DE-ABCM Combi Reader  
0x1DB2:0x0803:DUALi DE-620L Combi Reader  
0x1DB2:0x0804:DUALi DE-ABM4 Contactless Reader  
0x1DB2:0x0805:DUALi DE-ABM4S Contactless Reader  
0x1DB2:0x0806:DUALi DE-ABM5 Contactless Reader  
0x1DB2:0x0807:DUALi DE-ABCM2 Contactless Reader  
0x1DB2:0x0808:DUALi DE-620R Combi Reader  
0x1DB2:0x0809:DUALi DE-EPASS10 Contactless Reader  
0x1DB2:0x080B: DUALi DRAGON NFC Reader  
0x1DB2:0x088B: DUALi DRAGON NFC Reader2  
0x1DB2:0x080C: DUALi DE-ABCM6 Contactless Reader
```

Log when reader and card is detected:

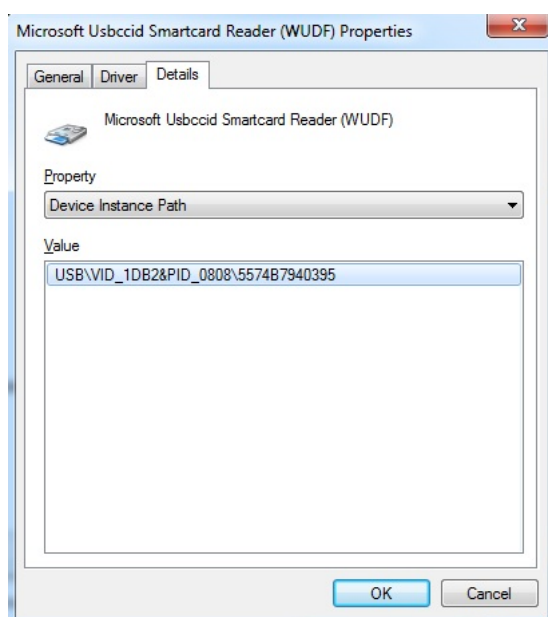
```
duali@ubuntu64bit-desktop:~$ pcsc_scan  
PC/SC device scanner  
V 1.4.16 (c) 2001-2009, Ludovic Rousseau <ludovic.rousseau@free.fr>  
Compiled with PC/SC lite version: 1.5.3  
Scanning present readers...  
0: DUALi Dragon NFC Reader2 (823121300740) 00 00  
  
Fri Jan 23 20:33:48 2015  
Reader 0: DUALi Dragon NFC Reader2 (823121300740) 00 00  
Card state: Card removed,  
  
Fri Jan 23 20:33:55 2015  
Reader 0: DUALi Dragon NFC Reader2 (823121300740) 00 00  
Card state: Card inserted,  
ATR: 3B F9 91 00 FF 91 81 71 3C 40 00 0A 80 04 66 24 C1 93 1B 80 8F
```


11.3 Control Code Usage Enable on Windows XP

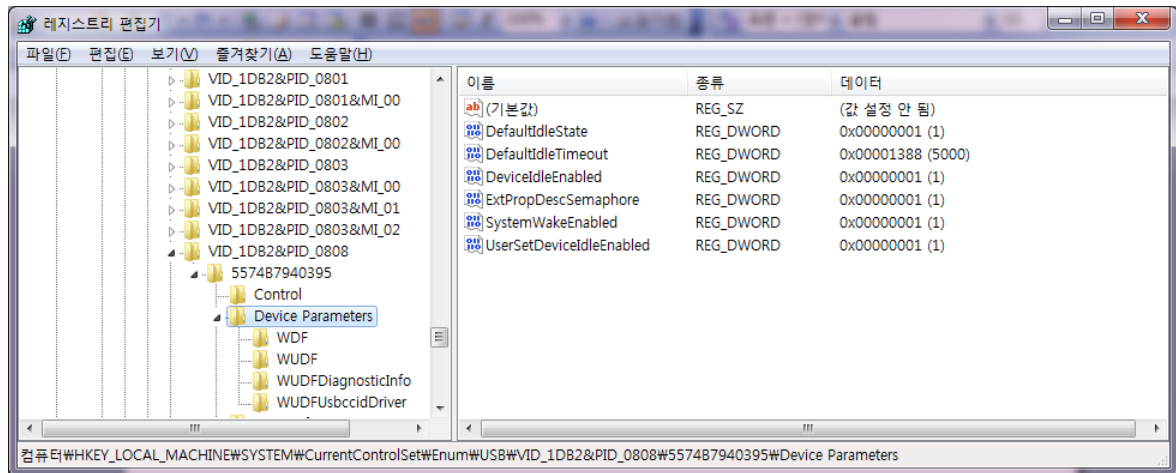
- Execute "regedit".
- Go to My Computer > HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Enum > USB



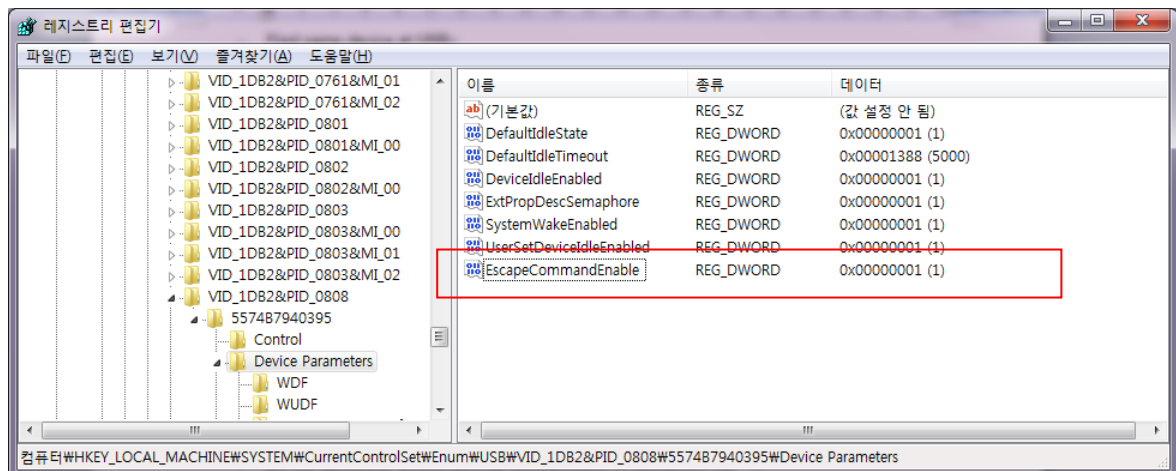
- Check "Device Instance Path" of the reader at Device Manager.



- Find same device at “My Computer > HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Enum > USB”



- Create new DWORD(32bit) Value parameter named “EscapeCommandEnable” at **Device Parameters**.
- Double click the parameter and change the value to “1”.



- Reboot PC. And then you can use SCardControl Function.
- It's not allowed to access through SCardControl on Windows7 yet.

12. STATUS(Response) Code Definition

OK	: 0 (0x00)
NO TAG ERROR	: 2 (0x02)
CRC ERROR	: 3 (0x03)
EMPTY (NO IC CARD ERROR)	: 4 (0x04)
AUTHENTICATION ERROR	: 5 (0x05)
NO POWER	: 5 (0x05)
PARITY ERROR	: 6 (0x06)
CODE ERROR	: 7 (0x07)
SERIAL NUMBER ERROR	: 8 (0x08)
KEY ERROR	: 9 (0x09)
NOT AUTHENTICATION ERROR	: 10 (0x0A)
BIT COUNT ERROR	: 11 (0x0B)
BYTE COUNT ERROR	: 12 (0x0C)
TRANSFER ERROR	: 14 (0x0E)
WRITE ERROR	: 15 (0x0F)
INCREMENT ERROR	: 16 (0x10)
DECREMENT ERROR	: 17 (0x11)
READ ERROR	: 18 (0x12)
OVERFLOW ERROR	: 19 (0x13)
POLLING ERROR	: 20 (0x14)
FRAMING ERROR	: 21 (0x15)
ACCESS ERROR	: 22 (0x16)
UNKNOWN COMMAND ERROR	: 23 (0x17)
ANTICOLLISION ERROR	: 24 (0x18)
INITIALIZATION(RESET) ERROR	: 25 (0x19)
INTERFACE ERROR	: 26 (0x1A)
ACCESS TIMEOUT ERROR	: 27 (0x1B)
NO BITWISE ANTICOLLISION ERROR	: 28 (0x1C)
FILE ERROR	: 29 (0x1D)
INVAILD BLOCK ERROR	: 32 (0x20)
ACK COUNT ERROR	: 33 (0x21)
NACK DESELECT ERROR	: 34 (0x22)
NACK COUNT ERROR	: 35 (0x23)
SAME FRAME COUNT ERROR	: 36 (0x24)
RCV BUFFER TOO SMALL ERROR	: 49 (0x31)
RCV BUFFER OVERFLOW ERROR	: 50 (0x32)
RF ERROR	: 51 (0x33)

PROTOCOL_ERROR	: 52 (0x34)
USER_BUFFER_FULL_ERROR	: 53 (0x35)
BUADRATE_NOT_SUPPORTED	: 54 (0x36)
INVAILD_FORMAT_ERROR	: 55 (0x37)
LRC_ERROR	: 56 (0x38)
FRAMERR	: 57 (0x39)
WRONG_PARAMETER_VALUE	: 60 (0x3C)
INVAILD_PARAMETER_ERROR	: 61 (0x3D)
UNSUPPORTED_PARAMETER	: 62 (0x3E)
UNSUPPORTED_COMMAND	: 63 (0x3F)
INTERFACE_NOT_ENABLED	: 64 (0x40)
ACK_SUPPOSED	: 65 (0x41)
NACK_RECEIVED	: 66 (0x42)
BLOCKNR_NOT_EQUAL	: 67 (0x43)
TARGET_SET_TOX	: 68 (0x44)
TARGET_RESET_TOX	: 69 (0x45)
TARGET_DESELECTED	: 70 (0x46)
TARGET_RELEASED	: 71 (0x47)
ID_ALREADY_IN_USE	: 72 (0x48)
INSTANCE_ALREADY_IN_USE	: 73 (0x49)
ID_NOT_IN_USE	: 74 (0x4A)
NO_ID_AVAILABLE	: 75 (0x4B)
OTHER_ERROR	: 76 (0x4C)
INVALID_STATE	: 77 (0x4D)
MI_JOINER_TEMP_ERROR	: 78 (0x4C)
NOTYET_IMPLEMENTED	: 100(0x64)
FIFO_ERROR	: 109(0x6D)
WRONG_SELECT_COUNT	: 114(0x72)
WRONG_VALUE	: 123(0x7B)
VALERR	: 124(0x7C)
RE_INIT	: 126(0x7E)
NO_INIT	: 127(0x7F)