NFC-1901

PROTOCOL SPECIFICATIONS

NEAR FIELD COMMUNICATION, CONTACTLESS CARD READER FOR NFC, MIFARE, ISO-14443 TYPE A/B, RF-TAG, VISAWAVE, PAYPASS LEVEL 1 and USB HUB FOR EICM-1000 and PDR-1689/MMD-1902

For product information, application engineering assistance or pricing, contact us at:

Peripheral Dynamics Inc.
5150 Campus Drive
Plymouth Meeting, Pennsylvania 19462-1123
Toll-Free: 800-523-0253
Phone: 610-825-7090 Fax: 610-834-7708
Email: sales@pdiscan.com Internet: www.pdiscan.com

March 24, 2014

1. COMMUNICATION PROTOCOL

1.1 Communication Setting

Baudrate : 115200 bps

Data : 8 bits

Stop bit : 1 bit

Parity bit : None

Flow control : None

1.2 Communication Format

From Host to NFC-1901

STX	Status	Command	Length	Data	ETX	BCC

STX : Start of Communication 1byte (0x02)

Status : COMMAND FORMAT 1BYTE (0x01 : INTERNAL EXECUTION, 0x02 : RF CARD COMMAND, 0x02 : IC CARD COMMAND, 0x02 : MS CARD COMMAND)

Command : Host to NFC-1901 command 1byte

Length : Data length 2byte. If there is no Data field, Length is 0.

Data : data field, variable length. If Length field is 0, then Data field is no exist.

ETX : End of Communication 1byte(0x03)

BCC : Checksum byte. Exclusive ORed result from STX to ETX.

From NFC-1901 to Host

STX	Response	Command	Length	Data	ETX	BCC

STX : Start of Communication 1byte (0x02)

Response : NFC-1901 to Host Response 1byte

Length : Data length 2byte. If there is no Data field, Length is 0.

Data : data field, variable length. If Length field is 0, then Data field is no exist.

ETX : End of Communication 1byte(0x03)

BCC : Checksum byte. Exclusive ORed result from STX to ETX.

1.3 Command Set

- Response code Table -

0x00 : SUCCESS
0x01 : COMMAND ERROR
0x02 : PACKET ERROR
0x03 : STATUS ERROR
0x04 : PROCESS_ERROR
0x05 : BCC ERROR
0x06 : CARD NO EXIST
0x07 : LENGTH ERROR
0x08 : PARAMETER ERROR
0x09 : TIMEOUT ERROR
0x0A : FIRMWARE BCC ERROR

Get Version (0xA0)

command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x01	0xA0	0x00	0x00	No Data	0x03	0xA0

* Get Version Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xA0	0x00	0x0E	NFC-1901 V 1.0	0x03	

Card Detect (0xA1)

command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xA1	0x00	0x00	No Data	0x03	0xXX

Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xA1	0x00	0xXX	R-DATA	0x03	0xXX

R-DATA format

Card type 1 byte + Card information n Bytes

- Card type format

0x41 : ISO-14443 Type A
0x42 : ISO-14443 Type B
0x46 : FeliCa

- Card information byte per Card type

ISO-14443 Type A
SAK + UID
SAK value (Refer the each TAG's SAK)
ISO-14443 Type B
PUPI
FeliCa
FeliCa speed (0x01 : 212 kbps, 0x02 : 424kbps) + length 1 byte +
response code 1 byte + UID 8bytes + PAD 8 bytes + system code 2 bytes

<Example>

- Type A

Mifare classic
02 00 A1 00 06 41 08 8C 09 B7 94 03 49

Mifare Ultralight
02 00 A1 00 09 41 00 04 82 9E D9 5B 02 80 03 F0

ISO-14443-4 type A
02 00 A1 00 06 41 28 20 84 9D A3 03 55

NXP Desfire
02 00 A1 00 09 41 20 04 76 89 DA 65 1E 80 03 12Type B
02 00 A1 00 05 42 80 78 A2 50 03 ED

- FeliCa 212kbps

02 00 A1 00 14 46 01 12 01 01 01 04 10 38 0F 4C 01 10 0B 4B 42 84 85 D0 FF 03 B2

Card Activation (0xA2)

command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xA2	0x00	0x02	T-DATA	0x03	0xXX

T-DATA ==> Card Type (1Byte)+ATTR (1Byte)

TAG type	Card Type	ATTR
Mifare	0x41	0x08
Type A	0x41	0x20
Type B	0x42	0x00
Felica 212	0x46	0x01
Felica 424	0x46	0x02

* Response (NFC-1901 → Host)

- Success Data –

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xA2	0x00	0xXX	R-DATA	0x03	

- Error Data –

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x09	0xA2	0x00	0x00	No DATA	0x03	

Mifare Card response

02 00 A2 00 08 00 04 08 04 0B EC 5B 2A 03 35

00 04 → SENS_RES

08 → SAK

04 → UID Length

0B EC 5B 2A → UID

TYPE A Card response

02 00 A2 00 10 00 04 20 04 08 47 91 BC 08 57 80 02 01 10 00 09 03 34

00 04 → SENS_RES

20 → SAK

04 → UID Length

08 47 91 BC → UID

08 57 80 02 01 10 00 09 → RATS

TYPE B Card response

02 00 A2 00 0E 50 71 23 47 BE 04 08 00 00 00 71 C1 01 41 03 AA

50 71 23 47 BE 04 08 00 00 00 71 C1 → ATQB_RES(12 bytes)

01 → ATTRIB_RES Length

41 → ATTRIB_RES

FELICA 212 Card response

02 00 A1 00 16 46 01 14 01 01 01 02 12 36 0E FF 08 10 0B 4B 42 84 85 D0 FF 00 03 03 04

0x46 → Card type, 0x01 → Felica212(0x02 → Felica 424)

0x14 → Length

0x01 → response code byte

01 01 02 12 36 0E FF 08 → UID

10 0B 4B 42 84 85 D0 FF → PAD

00 03 → System Code

- The difference between “Card Detect” and “Card Activation”.

- Card Detect

Determine the card exists or removed.
If the card exists, then identify the card type. It takes time to scan the card type due to various card type.

- Card Activation

To use (activate) the card.
If the card type is already identified, then no need “Card detect” process.

➔ Card Activation Flow after “Card detection”

Card Detect

02 02 A1 00 00 03 A2
02 00 A1 00 06 41 28 20 84 9D A3 03 55

During “Detect process”, SAK is 0x28 that Mifare and type A can be used.

In case of, Mifare Card Select
02 02 A2 00 02 41 08 03 EA
02 00 A2 00 08 03 04 28 04 20 84 9D A3 03 1A

In case of, Type A Card Select
02 02 A2 00 02 41 20 03 C2
02 00 A2 00 13 03 04 28 04 20 84 9D A3 0B 78 80 B1 02 4A 43 4F 50 33 31 03 55

Power off (0xA3)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xA3	0x00	0x00	No Data	0x03	0xA0

* Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xA3	0x00	0x00	No Data	0x03	0xA2

After Card Activation , sending “RF TX” consistently to RF field.
After transaction with the card, “Power off” to stop sending “RF TX”.

Mifare LoadKey(0xA4)

command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xA4	0x00	0x08	T-DATA	0x03	0xXX

T-DATA ==> Sector NUM(1Byte) +KEY Type(1 Byte) + KEY(6Byte)

* Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xA4	0x00	0x00	No Data	0x03	0xA5

- Key Load Value setting to access the Mifare Classic Card

Ex) In case of, #2 sector - key type A, key value 0xff, 0xff, 0xff, 0xff, 0xff, 0xff
-> 02 02 A4 00 08 02 60 FF FF FF FF FF FF 03 CD
<- 02 00 A4 00 00 03 A5

- The method of Mifare Classic Card block Access

Card Activation → Mifare load key → read/write block

- Reference

Sector Trailer Blocks															
Byte #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Key A						Access Bits				Key B					

Sector Trailer Blocks

Access Bits								
Bit #	7	6	5	4	3	2	1	0
Byte 6	_C23	_C22	_C21	_C20	_C13	_C12	_C11	_C10
Byte 7	C13	C12	C11	C10	_C33	_C32	_C31	_C30
Byte 8	C33	C32	C31	C30	C23	C22	C21	C20
Byte 9								

Note **_Cxx** is inversion of **Cxx**

	Access bit			Description
	C1x	C2x	C3x	
Block0	C10	C20	C30	Data block 0 in the sector
Block1	C11	C21	C31	Data block 1 in the sector
Block2	C12	C22	C32	Data block 2 in the sector
Block3	C13	C23	C33	Data block 3 in the sector

Struct of Sector Trailer

Access bits			Access condition for				Application
C1	C2	C3	read	write	increment	decrement, transfer, restore	
0	0	0	key A|B[1]	key A|B1	key A|B1	key A|B1	transport configuration
0	1	0	key A|B[1]	never	never	never	read/write block
1	0	0	key A|B[1]	key B[1]	never	never	read/write block
1	1	0	key A|B[1]	key B[1]	key B[1]	key A|B[1]	value block
0	0	1	key A|B[1]	never	never	key A|B[1]	value block
0	1	1	key B[1]	key B[1]	never	never	read/write block
1	0	1	key B[1]	never	never	never	read/write block
1	1	1	never	never	never	never	read/write block
Access bits			Access condition for				Application

[1] if Key B may be read in the corresponding Sector Trailer it cannot serve for authentication (all grey marked lines in previous table). Consequences: If the reader tries to authenticate any block of a sector with key B using grey marked access conditions, the card will refuse any subsequent memory access after authentication.

Access conditions for data blocks

Access bits			Access condition for						Remark
			KEYA		Access bits		KEYB		
C1	C2	C3	read	write	read	write	read	write	
0	0	0	never	key A	key A	never	key A	key A	Key B may be read
0	1	0	never	never	key A	never	key A	never	Key B may be read
1	0	0	never	key B	key A|B	never	never	key B	
1	1	0	never	never	key A|B	never	never	never	
0	0	1	never	key A	key A	key A	key A	key A	Key B may be read, transport configuration
0	1	1	never	key B	key A|B	key B	never	key B	
1	0	1	never	never	key A|B	key B	never	never	
1	1	1	never	never	key A|B	never	never	never	

Remark: the grey marked lines are access conditions where key B is readable and may be used for data.

Mifare Card Read Block(0xA5)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xA5	0x00	0x01	T-DATA	0x03	0xXX

T-DATA ==> BLOCK NUM (1Byte)

* Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xA5	0x00	0x10	R-DATA	0x03	

R-DATA ==> Block Data (16Byte)

Mifare Card Write Block(0xA6)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xA6	0x00	0x11	T-DATA	0x03	0xXX

T-DATA ==> BLOCK NUM (1Byte) + Write Data (16Byte)

* Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xA6	0x00	0x00	No Data	0x03	0xA7

Mifare Card Read Sector(0xA7)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xA7	0x00	0x01	T-DATA	0x03	0xC6

T-DATA ==> Sector NUM (1Byte)

* Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xA7	0x00	0x30	R-DATA	0x03	0xXX

R-DATA ==> Sector Data (48Byte)

Mifare Card Write Sector(0xA8)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xA8	0x00	0x31	T-DATA	0x03	0xXX

T-DATA ==> Sector NUM (1Byte) + Write Data (48Byte)

* Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xA8	0x00	0x00	No Data	0x03	

Mifare Card Increment(0xA9)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xA9	0x00	0x06	T-DATA	0x03	

T-DATA ==> Block NUM(1Byte) + Trans Block NUM(1Byte) + Money Data(4Byte, MSB first)

* Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xA9	0x00	0x00	No Data	0x03	

Mifare Card Decrement(0xAA)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xAA	0x00	0x06	T-DATA	0x03	

T-DATA ==> Block NUM(1Byte) + Trans Block NUM(1Byte) + + Money Data(4Byte, MSB first)

* Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xAA	0x00	0x00	No Data	0x03	

Mifare Card Restore(0xAB)

* M-Card Restore command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xAB	0x00	0x02	T-DATA	0x03	

T-DATA ==> Block NUM(1Byte) + Trans Block NUM(1Byte)

* M-Card Restore Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xAB	0x00	0x00	No Data	0x03	

Type A/B Data Transfer(0xB0)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xB0	0xXX	0xXX	C-APDU	0x03	

* Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xB0	0xXX	0xXX	R-APDU	0x03	

NFC Polling(0xD4)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xD4	0xXX	0xXX	T-DATA	0x03	

T-DATA ==> NDEF format data

* response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xD4	0x00	0xXX	R-DATA	0x03	

R-DATA ==> NDEF format Tag data

If T-DATA has “no data”,

In case of “Tag detect”, read “Tag data” and send to HOST.

In case of “P2P device”, send the data from the device and then send to HOST

If T-DATA has “data”,

In case of “Tag detect”, write T-DATA to the Tag.

In case of “P2P device”, send T-DATA to the device.

CANCEL PROCESS(0xD3)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xD3	0x00	0x00	No Data	0x03	

* response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xD3	0x00	0x00	No Data	0x03	

This command can be used during the process of NFC Tag Detect, NFC Initiator mode, NFC Target mode.

Mifare UltraLight Read Block(0xB6)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xB6	0x00	0x01	T-DATA	0x03	

T-DATA : Read Block number 1 byte

* response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xB6	0x00	0x10	R-DATA	0x03	

R-DATA ==> Block data 16 bytes

- Error Data –

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x09	0xB6	0x00	0x00	No Data	0x03	

Mifare UltraLight Write Block(0xB7)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xB7	0x00	0x05	T-DATA	0x03	

T-DATA : Read Block number 1 byte + Write Block 4 Bytes

* response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xB7	0x00	0x00	No data	0x03	

- Error Data –

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x09	0xB7	0x00	0x00	No Data	0x03	

NFC TAG TYPE 4 Card Emulation(0xD5)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x02	0xD5	0xxx	0xxx	T-DATA	0x03	

T-DATA ==> after Tag detection, sending NDEF format data

In case of cancellation, send the command CANCEL PROCESS

In case of Tag Write and data received from "p2p device",

* Response (NFC-1901 → Host)

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x20	0xD5	0xXX	0xXX	R-DATA	0x03	

IC Card Check(0xCC)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x03	0xCC	0x00	0x00	No Data	0x03	

* Response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xCC	0x00	0x01	R-DATA	0x03	

R-DATA ==> 0x01 : Card Exist
0x00 : Card No Exist

IC Card DirectCommand(0xCB)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x03	0xCB	0xXX	0xXX	T-DATA	0x03	

T-DATA ==> EICM-1000 format Request data (Refer the spec of EICM-1000)

Ex) 02 03 cb 00 0b 65 00 00 00 00 01 00 00 00 00 64 03 c2

* response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xCB	0x00	0xXX	R-DATA	0x03	

R-DATA ==> EICM-1000 format Response data (Refer the spec of EICM-1000)

Ex) 02 00 cb 00 0b 81 00 00 00 00 01 00 41 fe 01 3e 03 c1

Command format of MSG_ PC_to_RDR_GetSlotStatus:

Offset	Field	Size	Value	Description
0	bMessageType	1	65h	MSG_PC_to_RDR_Get_SlotStatus
1	dwLength	4	00000000h	Message-specific data length
5	bSlot	1	01h	Identifies the slot number for this command (Assume 1)
6	bSeq	1	00h	Sequence number for command. (Assume 0)
7	abRFU	3	000000h	Reserved for Future Use

Data for transferring would be MSG_ PC_to_RDR_GetSlotStatus + LRC:

bMessageType	dwLength (LSB first)				bSlot	bSeq	abRFU (LSB first)			LRC
65h	00h	00h	00h	00h	01h	00h	00h	00h	00h	64h

MS Card DirectCommand(0xC1)

* command (Host → NFC-1901)

STX	STATUS	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x04	0xC1	0xXX	0xXX	T-DATA	0x03	

T-DATA ==> MS card format Request data

Ex) 02 04 c1 00 06 02 16 31 00 03 26 03 c2

Command format

STX	Class	Function	Length	TX Data	ETX	BCC
(1byte)	(1byte)	(1byte)	(1byte)	(n byte)	(1byte)	(1byte)

* STX : Command Start Character (0x02)
* Class : Class of command
* Function : Requested function for the chip to conduct
* Length : Number of bytes in Data field
* TX Data : Data
* ETX : End Character (0x03)
* BCC : Check Sum (Exclusive OR value from STX to ETX)

* response (NFC-1901 → Host)

- Success Data -

STX	RESPONSE	COMMAND	LENGTH_H	LENGTH_L	DATA	ETX	BCC
0x02	0x00	0xC1	0x00	0xXX	R-DATA	0x03	

R-DATA ==> MS card format Request data

Ex) 02 00 c1 00 0d 02 16 31 06 06 04 00 00 00 20 10 03 12 03 cd

Response format

STX	Class	Function	Length	Status	RX Data	ETX	BCC
(1byte)	(1byte)	(1byte)	(1byte)	(1byte)	(n byte)	(1byte)	(1byte)

* STX : Start Character for Response (0x02)
* Class : Re-transmission of the Class sent in the command
* Function : Re-transmission of the Function sent in the command
* Length : Number of bytes in RX Data field
* Status : Status Check Byte
  ACK (0x06): Successful execution of the command
  NAK (0x15): Failed execution of the command
* RX Data : Response Value and error code as response
* ETX : End Character for response (0x03)
* BCC : Check sum (Exclusive OR value from STX to ETX)

<The contents can be changed without prior notice.>

N/A