SONY®

FeliCa Lite-S

FeliCa Lite-S
Technical Note for Software Development

Version 1.11

No. M742-E01-11

Introduction

This document summarizes important information about the development of software (firmware) for contactless IC cards that utilizes FeliCa technology, in terms of maintaining compatibility with FeliCa Lite-S products.

For the differences between the FeliCa Lite documents and the FeliCa Lite-S documents, see "Differences Between FeliCa Lite Documents and FeliCa Lite-S Documents".

Intended audience

This document is written for the following readers:

- Service providers who use FeliCa Lite-S products
- Design engineers of Reader/Writer software (firmware) for FeliCa Lite-S products
- Design engineers of applications for the host controller for FeliCa Lite-S products that communicate with card products via the Reader/Writer
- Middleware design engineers who develop software development kits(SDKs) or similar for FeliCa Lite-S products

Readers of this document are assumed to have enough knowledge of software development for FeliCa technology, acquired from documents concerning card products, Reader/Writer, and SDK.

Purpose

The purpose of this document is to describe the basic information to consider when developing applications and middleware that use FeliCa technology, also when developing Reader/Writer software (firmware).

This document indicates where specifications differ between card products (such as the differences between FeliCa Lite-S and FeliCa Standard), under headings that include the word "Cautions". These cautions, however, do not cover all the differences between all the card products.

Even if you strictly observe these cautions, flawless operation of the application is not guaranteed.

Note: Eventually, you must verify the correct operation of the application at system level (i.e., in an operating environment) by using the card product.

For information about software development for FeliCa Standard products, see the "FeliCa Card Technical Note for Software Development".

Applicable products

The targets of description in this document are FeliCa Lite-S contactless IC cards and IC chip products.

For specific product names, see the following website:

http://www.sony.net/Products/felica/business/tech-support/index.html

In this document, "card products" represents FeliCa Lite-S products and FeliCa Standard products.

Provision of associated information

You can find the latest version of this document and related documents at the following website:
http://www.sony.net/Products/felica/business/tech-support/index.html

Before starting the development of applications or FeliCa card-related software, download the latest information for reference from the website. If you have any questions about any specific product, please contact the supplier of that product.

Contents

1 Cautions on command specifications

1.1 Product-specific specifications

[Users to be especially cautious in this section]

Application design engineers

FeliCa service providers

This section clearly describes the specifications of FeliCa Lite-S products that require special attention compared with FeliCa Standard card products. When you develop software compatible with both FeliCa Lite-S products and FeliCa Standard card products, you are recommended to design the software while taking each item in the following list into consideration:

1) Supported commands

   FeliCa Lite-S products support the following commands:

   - Polling command
   - Read Without Encryption command
   - Write Without Encryption command

   For any command that is not supported, No Response is returned.

   FeliCa Lite-S products

2) System Separation function

   FeliCa Lite-S products do not support System Separation.

3) Number of effective Block

   In FeliCa Lite-S products, the user-available size of the non-volatile memory is 14 Blocks.

4) Number of Block that can be accessed simultaneously by the Read Without Encryption command and the Write Without Encryption command

   The specifications for simultaneously-accessible Block of FeliCa Lite-S products are as follows:

   Read Without Encryption command: 4 Blocks

   Write Without Encryption command: 1 Block

   (Number of Block that can be accessed simultaneously is 2 when performing Write With MAC.)

5) Communication rate

   FeliCa Lite-S products support the fc / 64 (approx. 212 kbps) communication and fc / 32 (approx. 424 kbps) communication.

NOTE fc: Carrier frequency (13.56 MHz)

6) Area

There is no concept of Area in FeliCa Lite-S products.

7) Service

In FeliCa Lite-S products, one Service is registered for Read/Write Access and another one for Read Only Access at the time of shipment.

Regarding the type of Service, FeliCa Lite-S products support Random Service only.

FeliCa Lite-S products support 2-Byte Service Code only. When receiving a command with 4-Byte Service Code, FeliCa Lite-S products return either No Response or an error.

FeliCa Lite-S products have no authentication function for Service, so there is no Service Key.

8) PIN Service

FeliCa Lite-S products do not support PIN Service.

9) System Code

System Code of FeliCa Lite-S products has a fixed value, so it cannot be changed. Additionally, you can add System Code of 12FCh to support NDEF.

10) Check for existence of Service

In FeliCa Lite-S products, the Request Service command is not implemented. Instead, to check for the existence of Service, the Read Without Encryption command is used to read Service Code Block.

1.2 Cautions on command packet structure

[Users to be especially cautious in this section]

Application design engineers
Middleware design engineers
Reader/Writer design engineers

When structuring a card command packet, set only Packet Data Length and the value (as defined in the relevant User's Manual) in it. If any of the following is set, some card products might properly execute the command while others might return either No Response or an error:

- Packet Data Length longer than the defined length (caused when, for example, unnecessary data is added)
- Packet Data Length shorter than the defined length (caused when, for example, necessary data is missing)
- An undefined value (undefined parameter, undefined data, and so on).

1.3 Cautions on the Polling command

[Users to be especially cautious in this section]

Application design engineers
Middleware design engineers
Reader/Writer design engineers

1.3.1 Setting a wildcard for System Code

For System Code of the Polling command, you can set a wildcard (FFh) on a Byte-by-Byte basis. In the course of comparison with System Code of System existing in a card, the Byte with a wildcard is regarded as an arbitrary value. Therefore, using a wildcard, you can capture several cards, regardless of System Code, with a single execution of the Polling command.

When System Code of a logical card (System) is 0123h, for example, the card returns a response to the Polling command having System Code value of 0123h, FF23h, 01FFh, or FFFFh.

When setting FFFFh as a wildcard, all the cards can return a response and thereby significantly increase the probability of collision occurrence among responses returned simultaneously from more than one card. If the application can identify System Code of the card, avoid setting a wildcard. Instead, it is recommended to execute the Polling command by setting the value of System Code. For applications that target only FeliCa Lite-S products, it is recommended to execute the Polling command by setting System Code to the fixed value for FeliCa Lite-S products.

1.3.2 Request Code

FeliCa Lite-S products support Request Code shown below. For FeliCa Standard products, supported Request Code depends on the products. When unsupported Request Code is specified, no Request Data (2 Bytes) is added to the Polling response. The application shall be designed assuming that there are cases where no Request Data is added, even when Request Code is specified.

Request Code	Content	Description of Request Data (2 Bytes)
00h	No Request Data	No additional data
01h	Request for System Code	System Code existing in the card product, or, for a separated card, the first matching System Code found during a search which starts from logical card 0 (System 0) of the separated system.
02h	Request for communication performance	1st Byte: 00h 2nd Byte: If the card supports the communication performance, the applicable bit becomes 1. b7(MSB): Supports automatic switching of the communication rate[*1] b6-b2: Reserved (0) b1: Supports fc / 32 (approx. 424 kbps). b0: Supports fc / 64 (approx. 212 kbps).
Others	Reserved	No additional data(at the time of publication, no applicable product exists).

[*1] After the Polling command, data processing is performed by automatically discriminating the communication rate of commands received from the Reader/Writer.

NOTE Data reserved at the time this document was published might not continue to be reserved. Therefore, do not check the value of reserved data with either the Reader/Writer or the application.

1.3.3 Setting up Time Slot

For Time Slot values to be set in the Polling command, specify only the prescribed values (00h, 01h, 03h, 07h, or 0Fh). If any other values are specified, the results can differ depending on which card products you use.

1.4 Cautions on Read/Write commands

[Users to be especially cautious in this section]

Application design engineers
Middleware design engineers

1.4.1 Block List Element

For Block List, either 2-Byte Block List Element or 3-Byte Block List Element can be specified. Mixed use of them in the same Block List is also allowed. Middleware must be designed to support both 2-Byte and 3-Byte data lengths.

1.5 Timeout and retry control

[Users to be especially cautious in this section]

Application design engineers
Middleware design engineers
Reader/Writer design engineers

The command processing time within a card product differs, depending on the command and card product in use. For a stable RF communication, calculate the command processing time and perform retry at the appropriate time.

The command processing time of each card product can be acquired from PMm returned in the Polling response.

It is recommended to calculate the maximum command response time based on PMm, and then perform an appropriate retry process.

1.6 Interrupt process of card commands

[Users to be especially cautious in this section]

Application design engineers
Middleware design engineers

While a card is processing a command, it cannot receive any new command. Instead, It ignores new commands and continues the current command processing. After completing the processing of the command and transmitting the response data, the card returns to its 'listening' state. To transmit a new command, wait until a response is returned from the card, or until the maximum response time of the card has elapsed.[*1]

[*1] You can calculate the maximum response time of the card from PMm. And that of the Polling command from the value of Time Slot, not from PMm.

[Example of card command interruption via the Reader/Writer]

Some Reader/Writer products might be equipped with a command interrupt function. The application, however, is unable to stop command processing within a card using such a function of the Reader/Writer.

Therefore, even if another card command is transmitted to the Reader/Writer immediately after the application uses the command interrupt function of the Reader/Writer, in some cases such a command might not be executed (Figure 1-1). In other cases, an unexpected response might be returned from the Reader/Writer (Figure 1-2).

After issuing the command interrupt instruction to the Reader/Writer, it is recommended that the application waits for a period of [maximum response time of the interrupted card command + response communication time from the card to the Reader/Writer] before transmitting the next card command (Figure 1-3).

You can calculate the maximum response time of a card based on the PMm obtained from the Polling response.

The response communication time from a card to the Reader/Writer depends on the response packet length. The maximum response communication time is approximately 10 ms.

[Caution]

The following examples of the sequence discussed in this section are based on specific circumstances. The sequence may differ depending on the available functions of the Reader/Writer in use. For example, if the Reader/Writer has a command retry function for cards, it can restore the regular sequence of operations only by executing the command retry function, even if it does not confirm with a specific sequence specified in this section.

You are recommended to design your application after checking the available functions of your Reader/Writer.

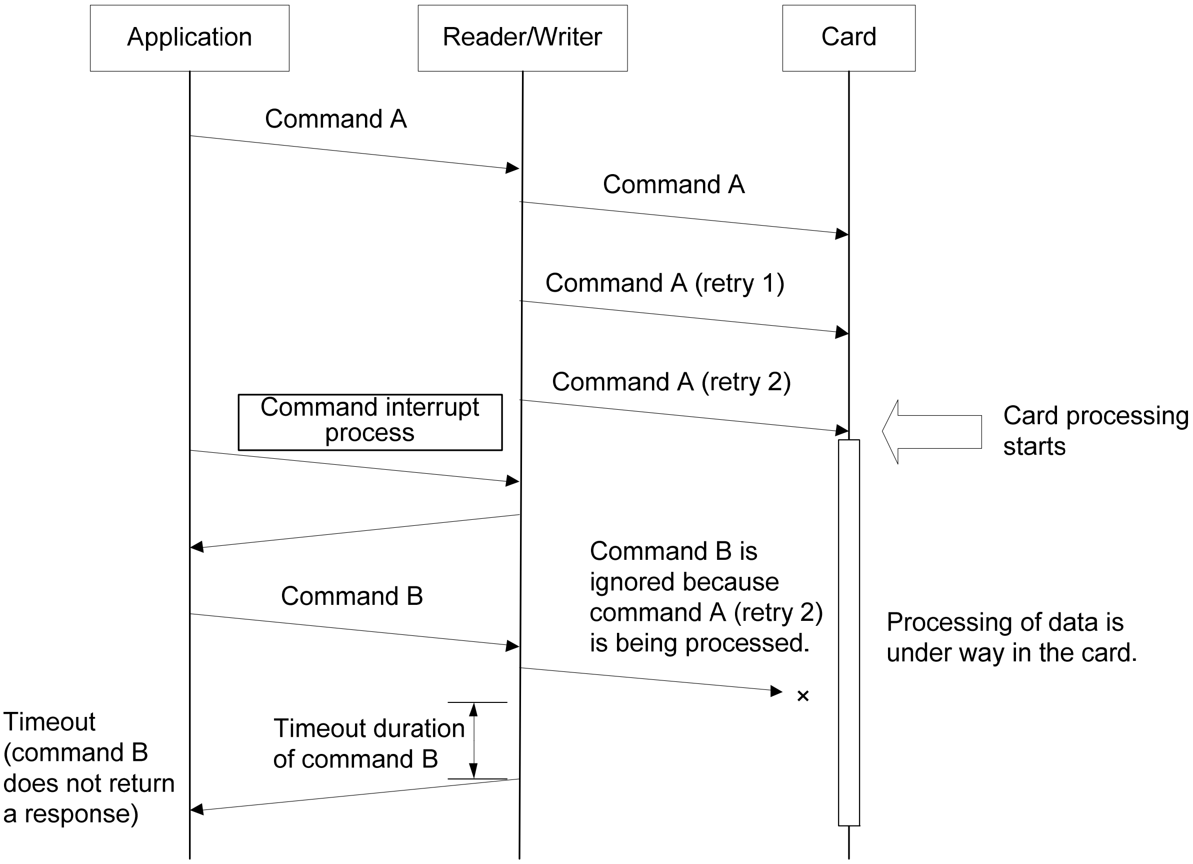

Application
Reader/Writer
Card
Command A
Command A
Command A (retry 1)
Command A (retry 2)
Command interrupt process
Card processing starts
Command B is ignored because command A (retry 2) is being processed.
Command B
Processing of data is under way in the card.
×
Timeout (command B does not return a response)
Timeout duration of command B

Figure 1-1: Example sequence (where No Response is returned)

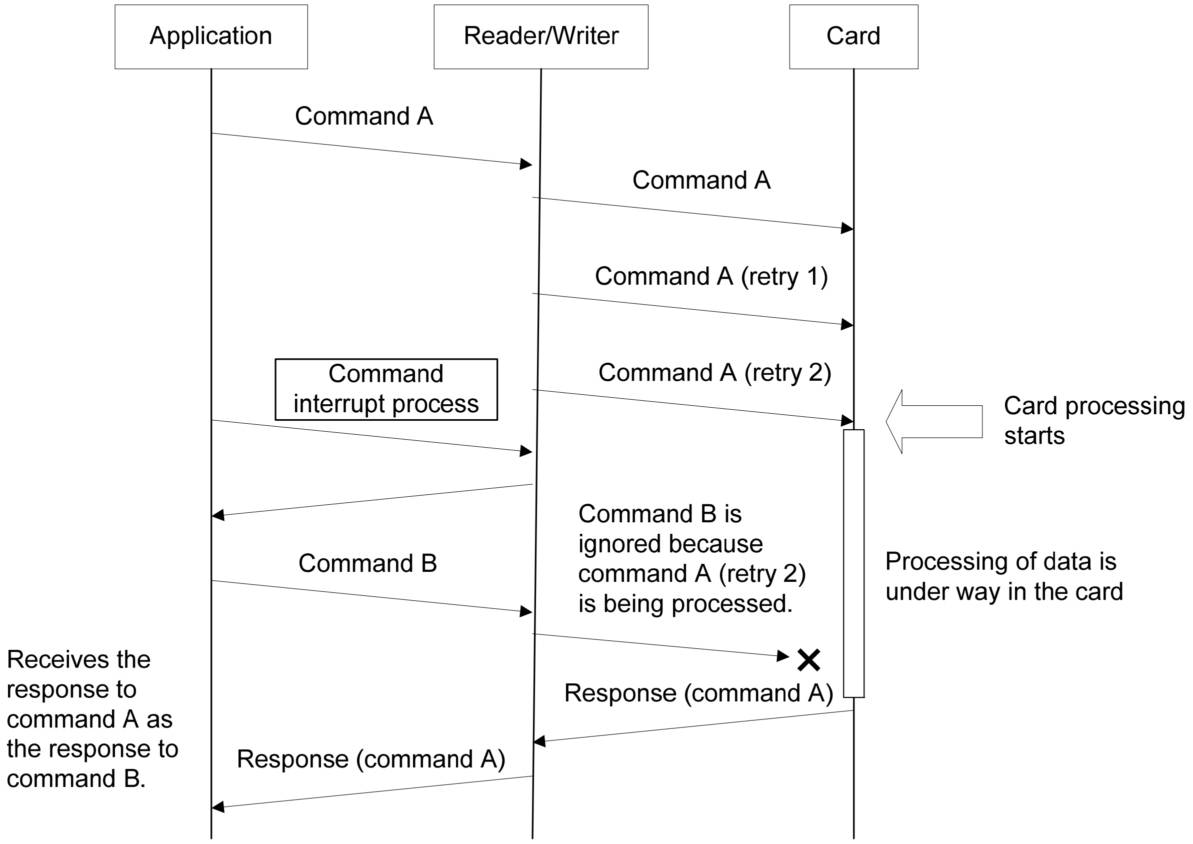
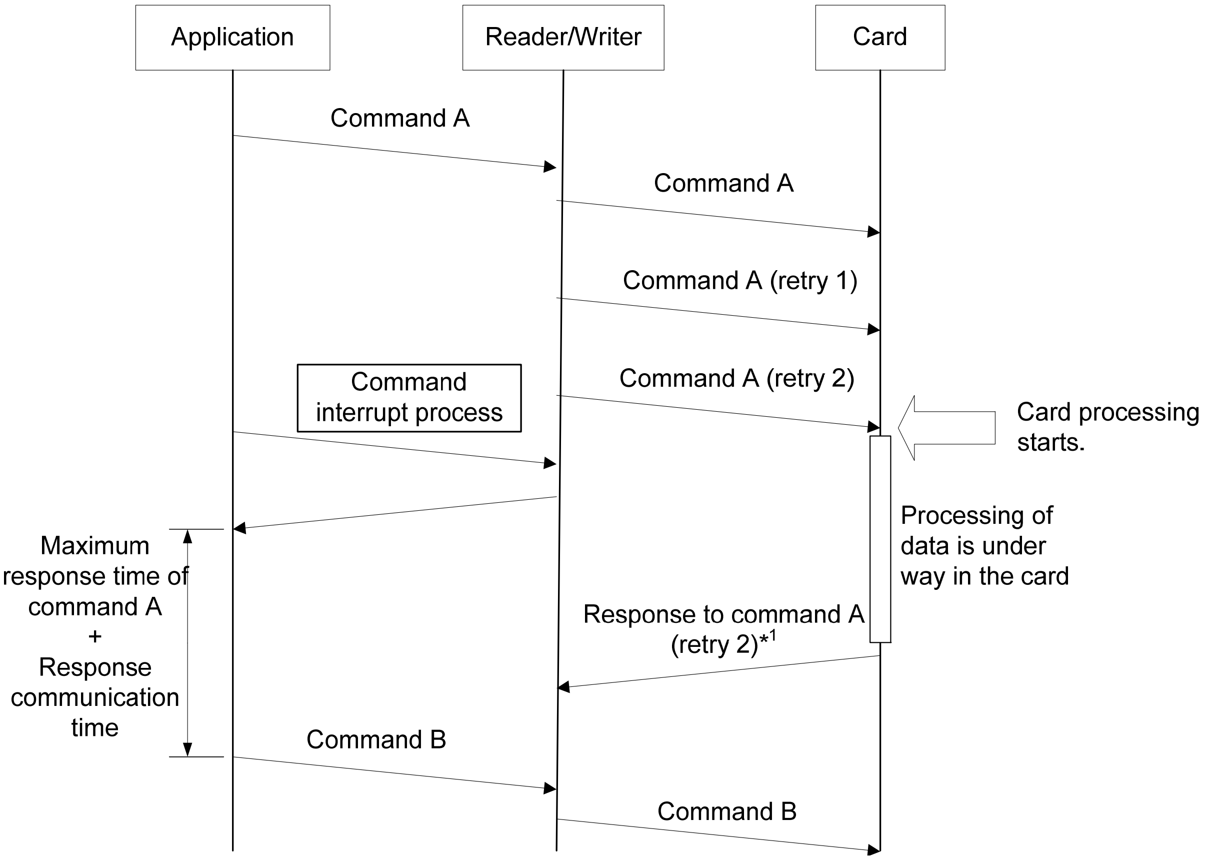

Application
Reader/Writer
Card
Command A
Command A
Command A (retry 1)
Command interrupt process
Command A (retry 2)
Card processing starts
Command B is ignored because command A (retry 2) is being processed.
Command B
Processing of data is under way in the card
Receives the response to command A as the response to command B.
Response (command A)
Response (command A)

Figure 1-2: Example sequence (where an unexpected response is returned)


Application
Reader/Writer
Card
Command A
Command A
Command A (retry 1)
Command interrupt process
Command A (retry 2)
Card processing starts.
Processing of data is under way in the card
Maximum response time of command A + Response communication time
Response to command A (retry 2)*1
Command B
Command B

[1] Processing method of response depends on implementation of the Reader/Writer.

Figure 1-3: Example sequence (regular behavior)

1.7 Sequence of input/output data for DES operations

[Users to be especially cautious in this section]

Application design engineers
Middleware design engineers

The exchange of DES operation data (plaintext, cyphertext, and key) is performed between the card and the Reader/Writer. For FeliCa Lite-S products, this exchange of DES operation data occurs during Read With MAC, Write With MAC, Internal Authentication, or External Authentication. For FeliCa Standard products, this exchange of DES operation data occurs during Mutual Authentication or encrypted communication between the card and the Reader/Writer.

When performing the DES operation, some differences exist between FeliCa Lite-S products and FeliCa Standard card products, as follows:

- FeliCa Lite-S products

  After sorting Block Data or the data extracted from the command packet, DES operation is performed. The result of DES operation is sorted again, and stored in response packet or Block Data.

- FeliCa Standard products

  The DES operation is performed without sorting Block Data or the data extracted from the command packet. The result of the DES operation is stored (without sorting) in the response packet or Block Data.

Figure 1-4 illustrates the concept of the differences mentioned above. For details of the DES operation in specific products, see the relevant user documentation supplied with those products.

The DES operation function contained in the SDK for FeliCa is implemented in a manner that is compatible with FeliCa Standard products. Therefore, if you use the DES operation function contained in the SDK for FeliCa Lite-S products, the sorting of data (permutation of the upper Byte and the lower Byte) for every 8 Bytes is required.

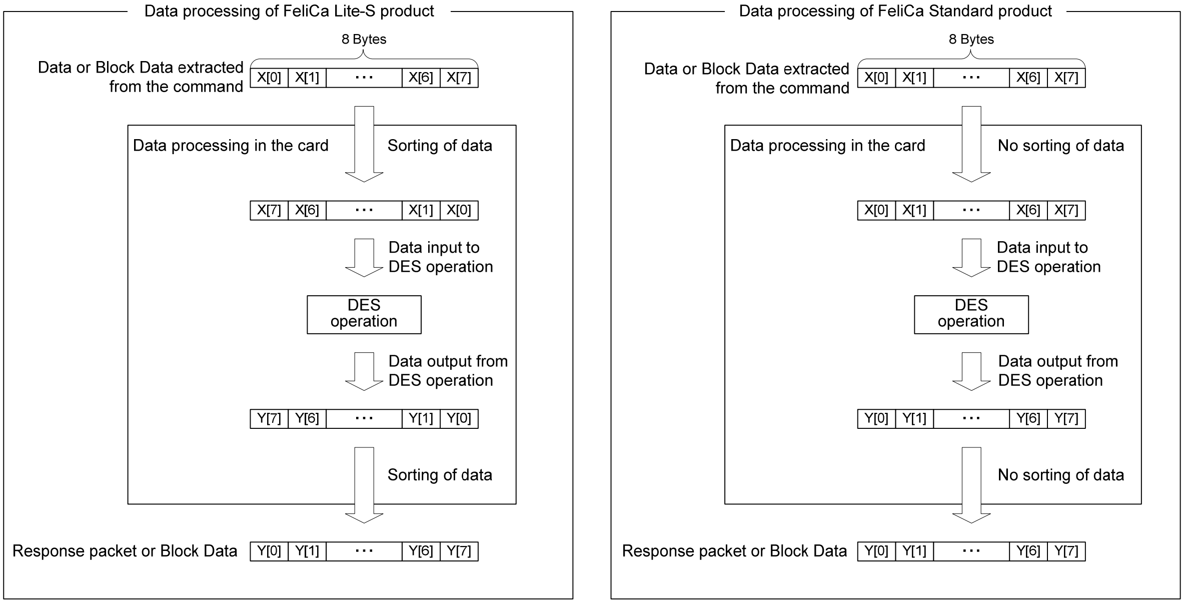

Data processing of FeliCa Lite-S product
8 Bytes
Data or Block Data extracted from the command
X[0] X[1] ··· X[6] X[7]
Data processing in the card
Sorting of data
X[7] X[6] ··· X[1] X[0]
Data input to DES operation
DES operation
Data output from DES operation
Y[7] Y[6] ··· Y[1] Y[0]
Sorting of data
Response packet or Block Data
Y[0] Y[1] ··· Y[6] Y[7]
Data processing of FeliCa Standard product
8 Bytes
Data or Block Data extracted from the command
X[0] X[1] ··· X[6] X[7]
Data processing in the card
No sorting of data
X[0] X[1] ··· X[6] X[7]
Data input to DES operation
DES operation
Data output from DES operation
Y[0] Y[1] ··· Y[6] Y[7]
No sorting of data
Response packet or Block Data
Y[0] Y[1] ··· Y[6] Y[7]

Figure 1-4: Concept diagram showing how to process DES operation data

2 Cautions on RF interface

2.1 Cautions on magnetic field control

[Users to be especially cautious in this section]

Reader/Writer design engineers
Application design engineers

When a card product enters the magnetic field generated by the Reader/Writer, the card product activates the IC within the card and starts initialization. After the initialization process completes, the card product enters the listening state.

Taking the magnetic field rise time into consideration, it is recommended that the Reader/Writer should continue to generate the magnetic field for at least 20.4 ms after the start of magnetic field generation, and should transmit the Polling command after this period of time. The Reader/Writer can transmit the Polling command before 20.4 ms has elapsed. In such cases, however, it is recommended to repeat Polling command transmission retries until 20.4 ms has elapsed while magnetic field generation continues, because the card product can fail to return a response to the commands before 20.4 ms.

2.2 Guard time

[Users to be especially cautious in this section]

Reader/Writer design engineers

After the end of reception of Response Packet Data from a card product (or, in the case of the Polling command, after the maximum response time has elapsed), it is recommended that the Reader/Writer application shall wait for at least (106 × 64 + 16) / fc (approx. 501 μs) before starting transmission of the top data of the preamble of the next command packet.

After the end of transmission of Command Packet Data to a card product, it is recommended that the Reader/Writer shall be ready to receive the top data of the preamble of the response packet from the card product within (42 × 64 - 16) / fc (approx. 197 μs).

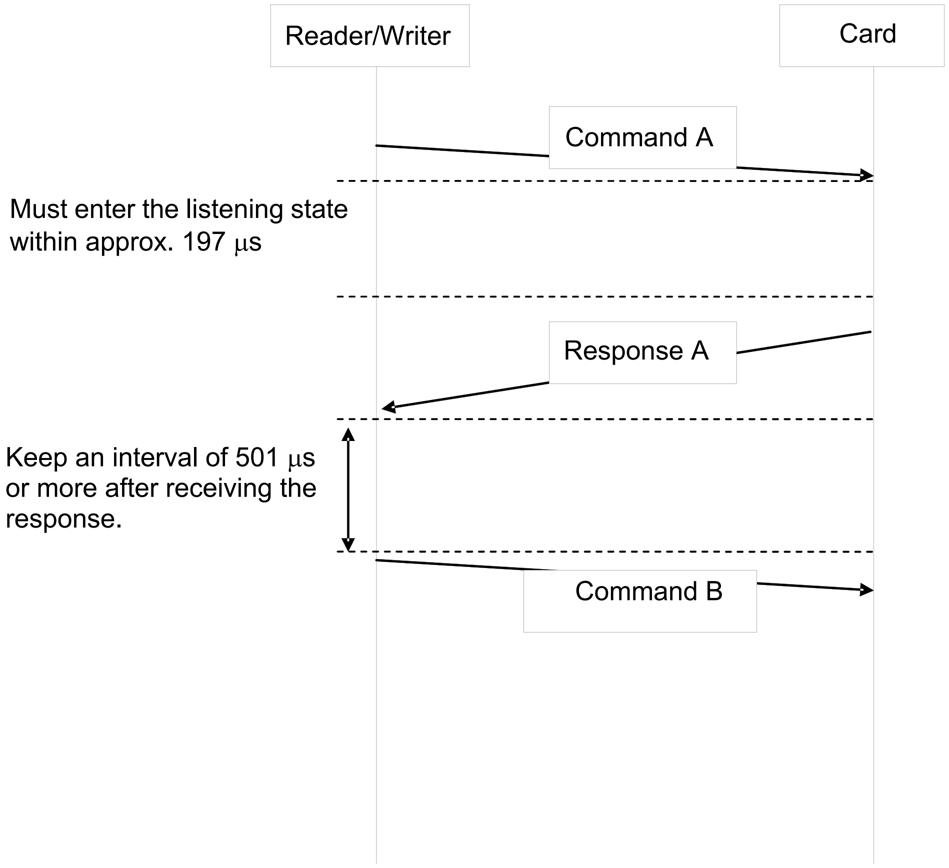

Reader/Writer
Card
Command A
Must enter the listening state within approx. 197 μs
Response A
Keep an interval of 501 μs or more after receiving the response.
Command B

Figure 2-1: Guard time for the Reader/Writer and card product

2.3 Cautions on magnetic field control during External Authentication

When magnetic field generation stops during External Authentication, FeliCa Lite-S is reset, causing the authentication process to be stopped on the way or the authentication result to be canceled. During External Authentication, it is necessary for the Reader/Writer to maintain the magnetic field.

When the Reader/Writer is implemented in a PC or a mobile phone, the middleware can be implemented to allow multiple applications or middleware programs to use the Reader/Writer simultaneously by time sharing. If such middleware is implemented, the Reader/Writer might stop magnetic field generation according to an instruction from an application or middleware while another application is performing External Authentication, causing the authentication process to be interrupted. To prevent this type of problem, configure the middleware setting so that the Reader/Writer can be used exclusively for a specific period of time if so required by an application.

2.4 Cautions on Polling Disable function

[Users to be especially cautious in this section]

Reader/Writer design engineers
Middleware design engineers
Application design engineers

Use of the Polling Disable function enables you to prevent FeliCa Lite-S from responding to Polling. When the magnetic field from the Reader/Writer stops, FeliCa Lite-S is reset, and this function is disabled (i.e. Polling is enabled.). The Reader/Writer must continue magnetic field generation while multiple cards are captured by using the Polling Disable function.

The Reader/Writer or middleware can be configured so that it sends the Polling command regularly to check for the existence of the card beyond the control of the application. If it detects No Response to the Polling command, it judges that the card is removed and stops magnetic field generation. This causes the Polling Disable function setting to be canceled. If you use your Reader/Writer or middleware in this way, configure it so that the card existence check function can be canceled by an application.

The Polling Disable function is one way to identify a specific card while more than one card is presented to the Reader/Writer simultaneously. This function can be used together with the Time Slot method.

2.5 Noise reception countermeasures

[Users to be especially cautious in this section]

Reader/Writer design engineers
Middleware design engineers
Application design engineers

The Reader/Writer may inadvertently start signal reception due to noise in the communication environment, the magnetic field generated by another Reader/Writer nearby, or power fluctuations caused by card operations. If this happens when the Reader/Writer is waiting for Response Packet Data, the Reader/Writer may not be able to receive Response Packet Data from the card (see Figure 2-2).

In order for the Reader/Writer to complete the interaction with a card even if it is inadvertently affected by noise in such cases, it is recommended that you take the following countermeasures:

- Command retry process
- Re-reception of Response Packet Data
- Ensuring better reception accuracy using preamble

The following is also recommended for processing in which data is written into the card:

- Recovery process

2.5.1 Command retry process

Calculate the maximum response time of the card based on PMm data and perform an appropriate retry process (see Figure 2-3). We recommend that you set the maximum possible number of retries allowed in your use case.

2.5.2 Re-reception of Response Packet Data

We recommend that you employ a reception process as in the following example in order for the Reader/Writer to be able to receive correct Response Packet Data even if it starts data reception inadvertently (see Figure 2-4). Set the waiting period of Response Packet Data to the maximum response time calculated from PMm data.

- Return to the waiting-to-receive state if the received packet has a CRC error.
- Stop receiving Response Packet Data and return to the waiting-to-receive state if Response Code of the received packet is not an expected value.
- Stop receiving Response Packet Data and return to the waiting-to-receive state if Data Length (LEN) of the received packet is not an expected value.

2.5.3 Ensuring better reception accuracy using preamble

One example of the countermeasures for preventing the Reader/Writer from inadvertently starting data reception due to noise is to have preamble in addition to a sync code (B24Dh) as a reception starting requisite (see Figure 2-5). For example, combine one-Byte preamble (00h) and a sync code (B24Dh) to use the resulting 3-Byte code (00B24Dh) as a reception starting requisite. It should be noted that careful evaluation is required to use this method because it may affect the communication characteristics or may not be as effective as you expected depending on the hardware.

2.5.4 Recovery process

If the Reader/Writer could not receive Response Packet Data even after the command retry process, it is recommended that the Reader/Writer reads out the data written into the card to confirm that the writing has succeeded.

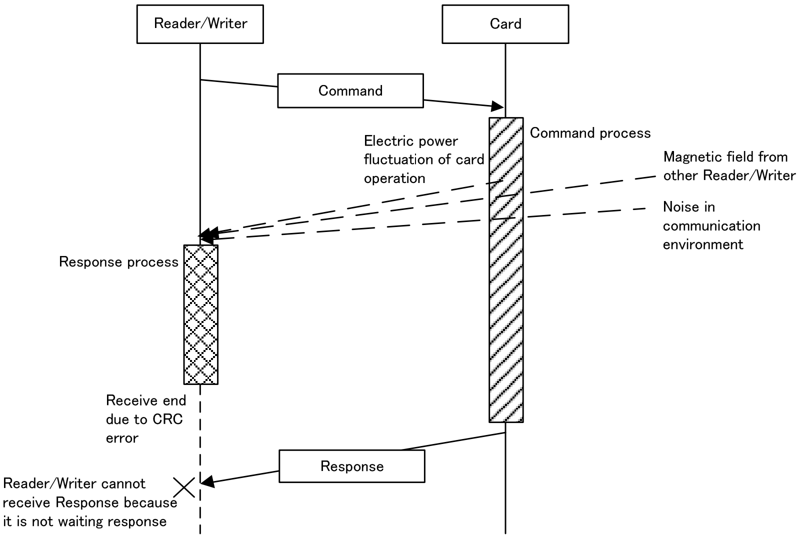

Reader/Writer
Card
Command
Command process
Electric power fluctuation of card operation
Magnetic field from other Reader/Writer
Noise in communication environment
Response process
Receive end due to CRC error
Response
Reader/Writer cannot receive Response because it is not waiting response

Figure 2-2: Example in which the Reader/Writer cannot receive Response Packet Data

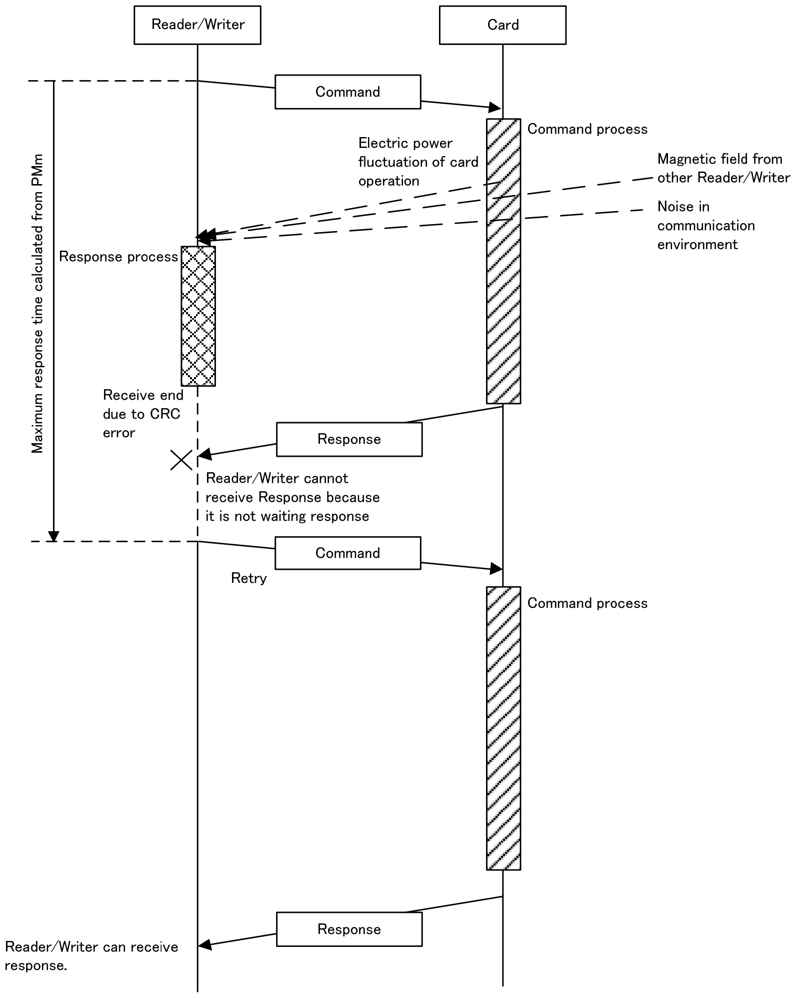

Reader/Writer
Card
Command
Command process
Electric power fluctuation of card operation
Magnetic field from other Reader/Writer
Noise in communication environment
Maximum response time calculated from PMm
Response process
Receive end due to CRC error
Response
Reader/Writer cannot receive Response because it is not waiting response
Command
Retry
Command process
Response
Reader/Writer can receive response.

Figure 2-3: Example in which the Reader/Writer can receive Response Packet Data by retry process

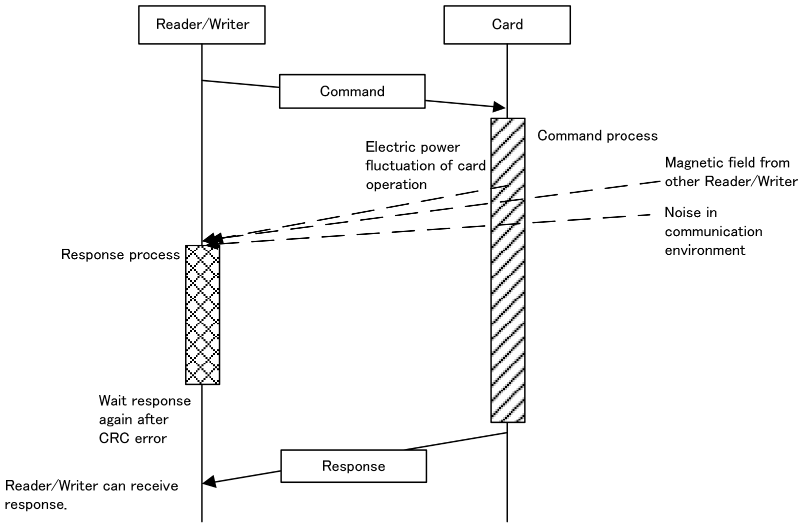

Reader/Writer
Card
Command
Command process
Electric power fluctuation of card operation
Magnetic field from other Reader/Writer
Noise in communication environment
Response process
Wait response again after CRC error
Response
Reader/Writer can receive response.

Figure 2-4: Example in which the Reader/Writer returns to waiting-to-receive state and can receive Response Packet Data

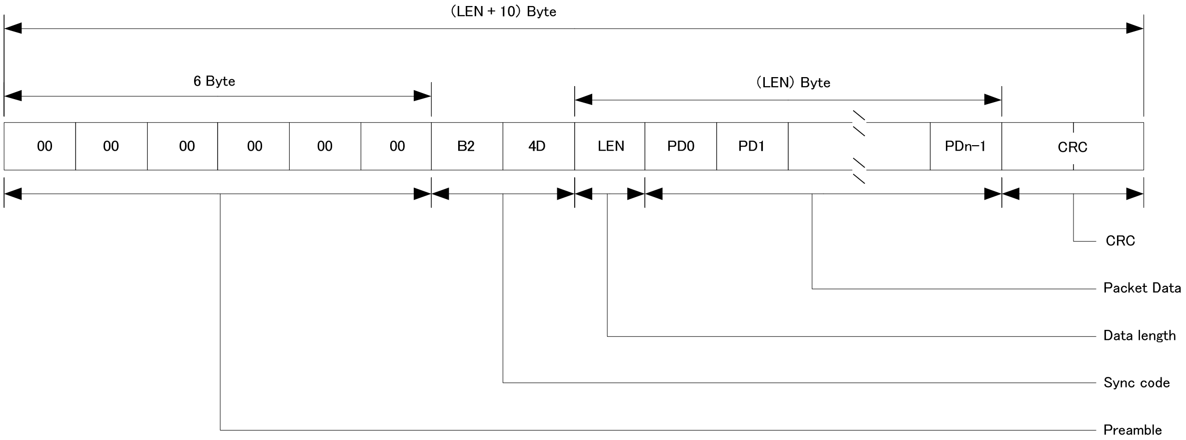


Figure 2-5: Packet structure

3 Cautions on authentication

3.1 Differences in authentication function

[Users to be especially cautious in this section]

Application design engineers

The Mutual Authentication (Internal Authentication and External Authentication) function implemented in FeliCa Lite-S products is performed using the Write Without Encryption command and the Read Without Encryption command.

The Mutual Authentication function implemented in FeliCa Standard products is performed using the Authentication1 command and the Authentication2 command.

Care should be taken when you create an application that commands used for authentication in FeliCa Lite-S is different from that in FeliCa Standard.

4 Cautions on issuance

4.1 Differences in issuance method

[Users to be especially cautious in this section]

Application design engineers for card issuance

For FeliCa Lite-S products, the data write for card issuance is performed using the Write Without Encryption command.

Care should be taken when you create an application that issuance of FeliCa Lite-S is different from that of FeliCa Standard.

5 Cautions on using Status Flag

5.1 Use of common specifications

[Users to be especially cautious in this section]

Application design engineers
Reader/Writer design engineers

During operation, if both Status Flag1 and Status Flag2 are 00h it is recommended to make sure that the operation completed successfully. In other cases, it is recommended to respond appropriately depending on the values of Status Flag.

When determining the occurrence of an error based on the values of Status Flag, it is recommended to use 80h - FFh of Status Flag2 (these are card-specific values) only for debugging purposes during application development.

The meanings of 00h - 7Fh of Status Flag2 are common to all card products. The implemented error determination methods differ for each card product, however, so the behavior (i.e., the values) can also differ even if the condition seems the same, as seen from the system level.

6 Cautions on keys

6.1 Differences in usage of keys

[Users to be especially cautious in this section]

Application design engineers

For FeliCa Lite-S products, only a single key can be registered to each card. This key is used to generate the MAC necessary for Internal/External Authentication.

For FeliCa Standard products, a specific key can be registered for each Area or Service. This key is used for Mutual Authentication.

Care should be taken when you create an application that FeliCa Lite-S uses keys differently from FeliCa Standard.

6.2 Cautions on rewriting Card Key and Card Key Version of FeliCa Lite-S

[Users to be especially cautious in this section]

Reader/Writer design engineers

Application design engineers

For FeliCa Lite-S products, both Card Key and Card Key Version cannot be rewritten simultaneously. You must prevent any mismatch between Card Key and Card Key Version.

An example of how to rewrite Card Key and Card Key Version follows:

This example is based on the assumption that Card Key Version must be rewritten after Card Key is changed and that Card Key Version must be increased. If a successful response cannot be received for rewriting Card Key, the following process is also effective.

Conduct Internal Authentication using Card Key to be written next. If the MAC values are matched, rewrite Card Key Version. If the settings of Card Key Block and Card Key Version Block are configured so that they can be changed by Write With MAC, Card Key Version Block is rewritten by Write With MAC using Card Key to be written next.

FeliCa Lite-S

FeliCa Lite-S Technical Note for Software Development Version 1.11

March 2012 First Edition FeliCa Business Division

April 2017 Revision

Sony Imaging Products & Solutions Inc.

No. M742-E01-11