



159159

- 1 -

IC CARD



G00002

FIELD

Embodiments described herein relate generally to
an IC card.

5

BACKGROUND

Many foreign countries adopt a GSM® (Global System
for Mobile communications) system as a mobile phone
system. In the GSM® system, a SIM (Subscriber Identity
Module) card as one kind of IC cards is indispensable
10 in a mobile phone.

In Japan, a mobile phone system, which adopts a
3GPP (3rd Generation Partnership Project) standard in
place of the GSM®, has been launched. In the 3GPP
standard as well, a USIM (Universal Subscriber Identity
15 Module) card as an IC card similar to the SIM card is
indispensable.

The SIM and USIM cards used in the GSM® and 3GPP
are inserted in mobile phone terminals. The SIM and
USIM cards record key information, encryption
20 algorithms, and various network parameters required to
establish connections to communication networks of
telecommunications carriers. Each mobile phone
terminal sends these pieces of information to an OTA
(Over The Air) server, authentication/management
25 server, and the like of the telecommunications carrier,
and executes authentication with these servers. When
the authentication result is correct, the mobile phone

terminal can receive communication services of that telecommunications carrier.

In standards (ETSI TS 102 223, ETSI TS 102 241), specifications called CAT (Card Application Toolkit) are specified. The CAT is installed in a card (to be referred to as a UIM hereinafter) represented by the SIM, USIM, and UIM cards of IC cards, which are required to identify subscriber information of, for example, mobile phones. As one of the CAT specifications, a function called a Proactive command, which is required to send various operation requests and information from the UIM to a mobile phone (to be referred to as an ME hereinafter) or network server, is specified.

According to the standards, when an Applet called a ToolKit Applet or embedded software is executed in the UIM, a Proactive command send request is often generated in the UIM. At this time, the UIM sends a Proactive command as response data to a Fetch command sent from the ME. However, the UIM cannot output the Proactive command before the ME sends the Fetch command.

On the other hand, in the UIM, another Proactive command send request may be generated by an operation of the Applet independently of this state. For this reason, when a state in which the UIM does not receive any Fetch command (the ME does not send any Fetch

command) is continued, unsent Proactive commands are accumulated in the UIM.

When the ME supplies a reset signal to the UIM, information of unsent Proactive commands is also reset.

5 Hence, a Proactive command to be sent cannot often be sent.

Therefore, there is a need for preventing data to be sent to the ME, which is generated by a specific command in the UIM, from being accumulated while being
10 kept unsent, or preventing data to be sent to the ME, which is generated by a specific command, from being erased involuntarily.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is an exemplary block diagram showing the
15 connection and arrangements of an ME and UIM according to the first embodiment;

FIG. 2 is an exemplary chart showing normal command exchange procedures between the ME and UIM according to the first embodiment;

20 FIG. 3 is an exemplary chart showing communication procedures required to discard accumulated Proactive commands between the ME and UIM according to the first embodiment;

FIG. 4 is an exemplary view showing a T-1 protocol
25 format according to the first embodiment;

FIG. 5 is an exemplary chart showing communication procedures required to re-send accumulated Proactive

commands between the ME and UIM according to the first embodiment;

FIG. 6 is an exemplary flowchart showing the processing sequence of the UIM according to the first embodiment; and

FIG. 7 is an exemplary block diagram for explaining an example of the arrangement of an IC card which incorporates the UIM according to the first embodiment.

10 DETAILED DESCRIPTION

In general, according to one embodiment, an IC card, which interprets and executes a command instruction from an external device, and returns the execution result as a response, comprises: a command generation unit configured to generate a request command which requests the external device to execute a specific operation in response to a specific command instruction from the external device; a storage unit configured to store the generated request command in a memory; a notification unit configured to notify the external device of generation of the request command when a response to the specific command instruction is sent, and notify the external device of the request command when the external device instructs to fetch the request command; and a discard unit configured to discard the request command in the memory when the external device instructs to discard the request

command.

[First Embodiment]

FIG. 1 is a block diagram showing the connection and arrangements of a mobile phone terminal (ME) 1 and UIM 2 according to the first embodiment. The UIM 2 is a portable electronic device, and is attached to the ME 1 to handle data when a mobile phone application is used.

The ME 1 includes a control unit 11, display unit 12, communication unit 14, and input unit 15. The control unit 11 systematically controls the operations of the respective units in the ME 1. The display unit 12 displays information for the user. The communication unit 14 executes data communications with the UIM 2. The input unit 15 inputs instructions from the user.

The UIM 2 includes a communication interface (I/F) unit 21, control unit 22, RAM 23, ROM 24, and nonvolatile memory 25.

The communication I/F unit 21 receives data sent from the ME 1, and outputs the received data to the control unit 22. The control unit 22 executes arithmetic operations and control, and systematically controls the UIM 2. The ROM 24 is a read-only memory. The ROM 24 stores basic software such as an operating system (OS), applications, and data. The RAM 23 is used as a work area in the processing of the control

unit 22. The nonvolatile memory 25 is a rewritable nonvolatile memory such as an EEPROM and flash memory, and is normally used as a work area and program area of the user. An Applet to be described later is stored in the nonvolatile memory 25.

FIG. 2 is a chart showing normal command exchange procedures between the ME 1 and UIM 2 according to the first embodiment.

In step S01, the ME 1 sends an Envelope command to the UIM 2. In this case, a plurality of types of formats are specified in the Envelope command by the aforementioned standards. "Eventdownload" is one of these formats, and has contents which notify the UIM 2 of generation of a predetermined event in the ME 1.

In the UIM 2, an Applet corresponding to the received Envelope command is launched, and a program described in the Applet is executed. In this case, when the UIM 2 causes the ME 1 to take an arbitrary action, for example, when the UIM 2 causes the ME 1 to execute processing for displaying a screen on the display unit 12 of the ME 1, a Proactive command which designates screen display processing has to be sent to the ME 1. Hence, in step S02, the UIM 2 sends a status word "91XX" to the ME 1. This status word "91XX" represents a Proactive command send request. That is, this status word represents that there is a Proactive command to be sent in the UIM 2.

Upon reception of the status word "91XX", the ME 1 outputs a Fetch command to the UIM 2 in step S03. Upon reception of the Fetch command, the UIM 2 sends a Proactive command and status word "9000" to the ME 1 in
5 step S04. Upon reception of the Proactive command, the ME 1 executes processing corresponding to that Proactive command. For example, the ME 1 executes an operation for outputting sound, displaying a screen, originating a phone call, or the like.

10 In step S05, the ME 1 outputs an execution result to the UIM 2 as a Terminal Response. For example, the ME 1 returns a response indicating that execution of the processing is complete, the processing cannot be executed since a user's operation is in progress,
15 execution of the processing is interrupted due to occurrence of an error, or the like. In step S06, the UIM 2 executes subsequent processing according to the program of the Applet. When there is no subsequent Proactive command to be processed by the Applet, the
20 UIM 2 sends a status word "9000".

Next, procedures executed when a Proactive command cannot be normally sent and is accumulated will be described below.

FIG. 3 is a chart showing communication procedures
25 required to discard accumulated Proactive commands between the ME 1 and UIM 2 according to the first embodiment.

In step S11, the ME 1 sends an Envelope (Event Download) command. The UIM 2 launches the Applet using reception of this command as a trigger. Then, the UIM 2 sends a status word (SW) = 91XX indicating that there is a Proactive command to be sent in step S12. The processes executed so far are the same as those in the procedures described using FIG. 2.

Normally, the ME 1 sends a Fetch command to the UIM 2. However, when the ME executes processing which cannot send any Fetch command, the Fetch command is not sent to the UIM 2, and another command associated with the processing in execution is sent. In step S13, the ME 1 sends a Select command. The UIM 2 receives the Select command, and normally executes processing. Next, the UIM 2 checks whether or not there is a Proactive command to be sent every time a command is sent. Since there is an unsent Proactive command, the UIM 2 sends SW = 91XX in step S14.

After that, the ME 1 does not send a Fetch command, and sends another Envelope (Menu Selection) command different from that in step S11 to the UIM 2 in step S15. The UIM 2 launches the Applet using this Envelope command as a trigger. Then, the Applet issues a Proactive command send request to the UIM 2. In step S16, the UIM 2 sends SW = 91XX to the ME 1.

In the aforementioned case, the two Proactive commands, that is, the Proactive command generated upon

launching the Applet for the first time and that
generated upon launching the Applet for the second time
are accumulated in the UIM 2. In this way, after the
Proactive command is received, when the ME 1 does not
5 send any Fetch command for the reason on the ME 1 side,
a state in which the Proactive commands are accumulated
occurs in the UIM 2.

Next, a correspondence on the ME 1 side will be
described below. For example, a case will be examined
10 below wherein the user is executing an operation for
inputting mail text in the ME 1. At this time, when a
Proactive command is received, and a screen
corresponding to that command is displayed, a function
currently used by the user is temporarily interrupted,
15 resulting in a quite inconvenient situation.
Therefore, the ME 1 inhibits reception of the Proactive
command at that time. That is, when the ME 1 is not
ready to receive and process the Proactive command, the
ME 1, which received SW = 91XX from the UIM 2, can
20 request the UIM 2 to discard the Proactive commands.

In step S17, the ME 1 sends a Select command which
indicates discarding of the Proactive commands to the
UIM 2. This Select command is sent in response to
reception of SW = 91XX in step S16 during execution of
25 a predetermined job in the ME 1. When the command data
indicates discarding of the Proactive commands, the UIM
2 discards the currently accumulated Proactive

commands. Note that a method of discarding the Proactive commands can be designated by the Select command. For example, the Select command can designate to discard the oldest Proactive command or to discard all the accumulated Proactive commands. The UIM 2 discards the Proactive commands, and sends a status word in step S18. When there is no Proactive command, the UIM 2 sends SW = 9000; when the Proactive command is left, it returns SW = 91XX.

10 When the Proactive command discard instruction is issued, a command is sent as an I-block of a T-1 protocol, and its PCB field can indicate discarding of the Proactive commands.

15 FIG. 4 shows the T-1 protocol format according to the first embodiment.

20 In the T-1 protocol format (ISO07816), a command is configured by NAD, PCB, LEN, INF, and EDC fields. The NAD field represents an address. The PCB field represents a control code indicating a mode. The LEN field represents a length of the command. The INF field represents a type (Select, Proactive, etc.) of the command. The EDC field represents an error detection check code.

25 When the mode is "I-block", the PCB field is configured by 8 bytes, and right five bytes are unused. Hence, data indicating discarding of a command can be described in this unused data field.

FIG. 5 is a chart showing communication procedures required to re-send accumulated Proactive commands between the ME 1 and UIM 2 according to the first embodiment.

5 Since the procedures in steps S21 to S26 are the same as those in steps S11 to S16 shown in FIG. 3, a detailed description thereof will not be repeated.

 When the ME 1 restarts reception of a Proactive command, it sends a Reset command to the UIM 2 in step
10 S31. This Reset command is sent to initialize the UIM 2 when, for example, the power supply of the ME 1 is switched from OFF to ON. Therefore, the UIM 2 which received the Reset command clears the RAM 23. Conventionally, since a Proactive command is
15 accumulated in the RAM 23, it is discarded. However, in this embodiment, since a Proactive command is accumulated in the nonvolatile memory 25, it is not discarded. In step S32, the UIM 2 returns ATR (answer to reset) as a response to the Reset command.

20 In step S33, the ME 1 sends a Select command. The UIM 2 receives the Select command, and normally executes processing. Next, the UIM 2 checks whether or not there is a Proactive command to be sent. Since the nonvolatile memory 25 accumulates the Proactive command
25 to be sent, the UIM 2 sends SW = 91XX to the ME 1 in step S34.

 The ME 1 which received the status word "91XX"

outputs a Fetch command to the UIM 2 in step S35. Upon
reception of the Fetch command, the UIM 2 sends a
Proactive command and status word "9000" to the ME 1 in
step S36. The ME 1 which received the Proactive
5 command executes processing corresponding to that
Proactive command. For example, the ME 1 executes an
operation for outputting sound, displaying a screen,
originating a phone call, or the like.

In step S37, the ME 1 outputs an execution result
10 to the UIM 2 as a Terminal Response. For example, the
ME 1 returns a response indicating that execution of
the processing is complete, the processing cannot be
executed since a user's operation is in progress,
execution of the processing is interrupted due to
15 occurrence of an error, or the like. The UIM 2 which
received the Terminal Response checks whether or not
there is another Proactive command to be sent. If
there is a Proactive command to be sent, the UIM 2
sends SW = 91XX to the ME 1 in step S38.

20 After that, the UIM 2 repetitively executes the
processes in steps S35 to S38 as many as the number of
remaining Proactive commands.

In step S41, the ME 1 which received the status
word "91XX" outputs a Fetch command to the UIM 2. Upon
25 reception of the Fetch command, the UIM 2 sends a
Proactive command and status word "9000" to the ME 1 in
step S42. The ME 1 which received the Proactive

command executes processing corresponding to that
Proactive command. In step S43, the ME 1 outputs an
execution result to the UIM 2 as a Terminal Response.
Upon reception of the Terminal Response, the UIM 2
5 sends SW = 9000 indicating that there is no Proactive
command to be sent to the ME 1 in step S44.

According to the procedures described using
FIGS. 3, 4, and 5, the Proactive commands can be
discarded at a desired timing, and can be restarted
10 after the reset operation.

The processing sequence of the UIM 2 required to
implement the aforementioned communication procedures
will be described below with reference to FIG. 6. Note
that "UIM-control unit" shown in FIG. 6 indicates the
15 control unit 22, and "UIM - Applet unit" indicates the
Applet stored in the nonvolatile memory 25.

In step S50, the ME 1 sends a command. The
UIM-control unit receives the command in step T01, and
interprets the command in step T02. The UIM-control
20 unit judges in step T03 whether or not the received
command instructs to discard a Proactive command.

If the received command instructs to discard a
Proactive command, the UIM-control unit discards the
Proactive command accumulated in the nonvolatile memory
25 in step T04. Then, the process advances to step
T05. If the received command does not instruct to
discard a Proactive command, the UIM-control unit

interprets the command in step T05. The UIM-control unit judges in step T06 whether or not execution of the Applet is required.

5 For example, in the aforementioned example, if the command is an Envelope or Fetch command, the UIM-control unit judges that execution of the Applet is required; if the command is a Terminal Response, it judges that execution of the Applet is not required.

10 If the UIM-control unit judges in step T06 that execution of the Applet is not required, it executes the command in step T10. The UIM-control unit checks in step T11 whether or not Proactive commands are accumulated in the nonvolatile memory 25.

15 If no Proactive command is accumulated in the nonvolatile memory 25, the UIM-control unit sets SW = 9000 in step T12. If Proactive commands are accumulated in the nonvolatile memory 25, the UIM-control unit sets SW = 91XX in step T13.

20 In step T14, the UIM-control unit sends a response. In step S51, the ME 1 receives the response.

If the UIM-control unit judges in step T06 that execution of the Applet is required, it executes the Applet in step P01, and interprets the command in step P02. The UIM-control unit checks in step P03 if the
25 command is a Fetch command.

If the command is not a Fetch command, the UIM-control unit executes the Applet in step P04. The

UIM-control unit judges in step P05 whether or not a Proactive command is required to be sent.

If a Proactive command is not required to be sent, the UIM-control unit sets SW = 9000 in step T12. In
5 step T14, the UIM-control unit sends a response.

If a Proactive command is required to be sent, the UIM-control unit stores the Proactive command in the nonvolatile memory 25 in step P06. In step T13, the UIM-control unit sets SW = 91XX. In step T14, the
10 UIM-control unit sends a response.

If the UIM-control unit determines in step P03 that the command is a Fetch command, it sets a Proactive command and SW = 91XX in data to be sent in step P07. In step P08, the UIM-control unit deletes
15 the Proactive command to be sent from the nonvolatile memory 25. In step T14, the UIM-control unit sends a response.

Note that in the flowchart shown in FIG. 6, if it is determined in step T03 that the received command
20 instructs to discard a Proactive command, the Proactive command accumulated in the nonvolatile memory 25 is discarded in step T04. In this case, the oldest Proactive command may be discarded in place of all Proactive commands in the nonvolatile memory 25.

25 Note that the aforementioned UIM 2 is configured to be built in an IC card and to exchange data with a card reader/writer.

FIG. 7 is a block diagram for explaining an example of the arrangement of an IC card 100 which incorporates the UIM of the first embodiment.

As shown in FIG. 7, the IC card 100 includes a
5 card-like main body 101 formed of, for example,
plastic, and an IC module 102 incorporated in the main
body 101. The IC module 102 includes one or a
plurality of IC chips 103 and a communication I/F unit
21. Each IC chip 103 and communication I/F unit 21 are
10 embedded and formed in the IC module 102 while being
connected to each other. The UIM of this embodiment
can be configured as the IC module 102.

Note that the functions described in the
aforementioned embodiment is configured using hardware.
15 The functions can also be implemented by loading
programs which describe the respective functions using
software into a computer. The respective functions can
be configured by selecting software or hardware as
needed.

20 Note that the present invention is not limited to
the aforementioned embodiment intact, and can be
embodied by modifying components without departing from
its scope and spirit when it is practiced.

While certain embodiments have been described,
25 these embodiments have been presented by way of example
only, and are not intended to limit the scope of the
inventions. Indeed, the novel embodiments described

herein may be embodied in a variety of other forms;
furthermore, various omissions, substitutions and
changes in the form of the embodiments described herein
may be made without departing from the spirit of the
5 inventions. The accompanying claims and their
equivalents are intended to cover such forms or
modifications as would fall within the scope and spirit
of the inventions.

WHAT IS CLAIMED IS:

1. An IC card, which interprets and executes a command instruction from an external device, and returns the execution result as a response, comprising:

5 a command generation unit configured to generate a request command which requests the external device to execute a specific operation in response to a specific command instruction from the external device;

10 a storage unit configured to store the generated request command in a memory;

15 a notification unit configured to notify the external device of generation of the request command when a response to the specific command instruction is sent, and notify the external device of the request command when the external device instructs to fetch the request command; and

 a discard unit configured to discard the request command in the memory when the external device instructs to discard the request command.

20 2. The IC card according to claim 1, wherein the notification unit notifies the external device of generation of the request command when a command instruction other than the command instruction which instructs to discard the request command is received
25 and when the request command is stored in the memory.

 3. The IC card according to claim 2, wherein the discard unit discards all request commands in the

memory.

4. The IC card according to claim 2, wherein the discard unit discards an oldest request command in the memory.

5 5. The IC card according to claim 2, further comprising:

 a reset unit configured to initialize the IC card when a command instruction which instructs to initialize is received from the external device,
10 wherein the reset unit does not discard any request command in the memory.

6. The IC card according to claim 5, wherein the storage unit stores the generated request command in a nonvolatile memory.

15 7. An IC card comprising:

 an IC module which includes a command generation unit configured to generate a request command which requests an external device to execute a specific operation in response to a specific command
20 instruction from the external device, a storage unit configured to store the generated request command in a memory, a notification unit configured to notify the external device of generation of the request command when a response to the specific
25 command instruction is sent, and notify the external device of the request command when the external device instructs to fetch the request

command; and

an IC card main body which houses the IC module.