NAGRAVISION

DIGITAL PAY-TV SYSTEMS

Confidential

EMC

IRD Commands Specification

for new IPPV capable Set-Top Boxes

1.3.16-e

First published, January 2004.

Revised, January 2005.

Part number: EmcIPPVStbCakIrdSpe010316e.doc,

Nagravision S.A. is a member of the Kudelski Group of Companies.
Tel.: (41) (21) 732-0311 Fax: (41) (21) 732-0300

First published, January 2004.

Revised, January 2005.

Part number: EmcIPPVStbCakIrdSpe010316e.doc,

Nagravision S.A. is a member of the Kudelski Group of Companies.
Tel.: (41) (21) 732-0311 Fax: (41) (21) 732-0300

Table Of Contents

List Of Tables

1 Introduction

1.1 Purpose

This document defines the general format of an IRD command as well as generic NagraVision commands, such as "Reset PIN Code" or "Force Tune". It also defines the rules for defining manufacturer's or operator's specific commands.

1.2 Definitions, Acronyms, and Abbreviations

Acronym Abbreviation	Definition
CA	Conditional Access
CAK	Conditional Access Kernel
CRL	Certificate Revocation List
DVB	Digital Video Broadcasting
IRD	Integrated Receiver Decoder
MKY	MovieKey
NVM	Non-volatile memory
STB	Set-Top Box

Table 1 - Definitions, Acronyms, and Abbreviations

1.3 Notational Conventions

All source code occurrences appear in `courier` writing style

1.4 References

[1] Force Identification, Implementation Guidelines V1.0.0
[2] IRD Master/Slave, Solution Overview, Issue 1.0.0
[3] IRD Master/Slave, Implementation Guideline, Issue 0.0.3
[4] ANSI/STCE 41 2003, POD Copy Protection System
[5] NagraVision, Data Item Loader, Application Programming Interface, V 1.0.4 or higher.

1.5 Trademarks

1.6 Overview

IRD commands allow the head-end to send messages to the set-top box in a secured way. IRD commands are carried by EMMs. They can benefit from EMM addressing mode. It means that a message can be addressed either to one single set-top box or to all set-top boxes.

The CA Kernel embedded in the set-top box is not dependent at all on IRD commands. It gets the command from the smartcard and forwards it to the set-top box application without additional processing. The set-top box application is completely responsible for IRD command management. Periodicity of commands (coming from the fact that commands are carried by EMMs) has to be managed by the set-top box application by means of the sequence number. If a command has to be split in several commands due to the EMM length limitation[1], it is also the responsibility of the set-top box application to re-build the original command.

NagraVision has defined a set of generic commands. The table below gives a synopsis of these commands along with the associated command identifier. Refer to §3 for a detailed description.

Name	command_id	operation
Reserved	0x12	0x01
Mail	0xC0	0x01
Force Tune	0xC1	0x01
Reserved	0xC2	0x01
Reserved	0xC4	0x01
Reserved	0xC5	0x01
Set Network ID	0xC6	0x01
Master/Slave Initialization	0xC7	0x01
Master/Slave Cancellation	0xC7	0x02
Master/Slave Single Shot	0xC7	0x03
Reserved	0xC7	0x04
Set PIN Code	0xC8	0x01..0xFF
Reserved	0xC9	0x01
Reserved	0xCA	0x00, 0x01
Reserved	0xCB	0x00
Reserved	0xCB	0x01
Reserved	0xCB	0x02
Reserved	0xCB	0x03
Reserved	0xCC	0x01
Reserved	0xCD	0x01
Reserved	0xCF	0x00..0x01
Reserved	0xD0	0x00
Reserved	0xD1	0x00
Reserved	0xD1	0x01

[1] The maximum size of the `command_body` that can be carried by one IRD-CMD is 75 Bytes for DNASP-3 and 61 Bytes for DNASP-2 (the complete IRD buffer returned by the ICC includes 3 more Bytes, the `EMM_command`, the `length` and the `checksum`).

Name	command_id	operation
Reserved	0xD1	0x02
Reserved	0xD1	0x03
Reserved	0xD1	0x04
Reserved	0xD2	0x00
Reserved	0xD3	0x00
Reserved	0xD4	0x00

Table 2 - Commands Summary

All commands required by a manufacturer or an operator that does not belong to this list may result in a specific command. Refer to §4 for a description of the procedure allowing the definition of a specific command.

2 IRD Command Format

Description

Defines the general format of an IRD command.

Format

```
IRD_command(){
  EMM_command                 8     uimsbf    0x64
  length                      8     uimsbf    7+N, max=71 for Aladin,
                                                max=55 for DNASP2

  command_body(){
    sequence_number           32    uimsbf
    command_id                8     uimsbf
    operation                 8     uimsbf
    for(i=0; i<N; i++){                       Nmax=64 for Aladin, Nmax=48
                                              for DNASP2
      data                    8     bslbf
    }
    checksum                  8     bslbf
  }
}
```

Parameters

`sequence_number` value incremented whenever a command is generated by the head-end.

Since IRD commands are carried by EMMs, the set-top box application may be notified of the same command several times. It is the responsibility of the set-top box application to process the sequence number in order to avoid a command to be run several times.

To do so the sequence number of the last *x* commands run by the application may be stored in NVM. The *x* value depends on the maximum number of different commands that could be broadcast at the same time on the network. It is operator dependent.

`command_id` command identifier.

`operation` used in conjunction with the command_id. The couple (command_id, operation) uniquely identifies a command.

`data` additional data (optional)

`checksum` two's complement of the sum of all bytes from the command_id to the last data byte. The sum of all bytes from the command_id to the checksum must be equal to 0.

For instance, the checksum of the reset PIN code command here after is equal to `` `ED `` (`` `12+01+ED=0 ``)

`` `64 07 00000007 **12 01** ED ``

3 Generic IRD Commands

3.1 Mail

Description

This command provides mail messages to the STB. The management of the messages is the STB responsibility.

Format

```
IRD_command(){
  EMM_command                 8     uimsbf    0x64
  length                      8     uimsbf    10+N
  command_body(){
    sequence_number           32    uimsbf
    command_id                8     uimsbf    0xC0
    operation                 8     uimsbf    0x01
    data{
      mail_id                 10    bslbf     Mail message number
      total_segment           6     bslbf     Total number of segments
      priority                2     bslbf     0 normal priority
                                              1 high priority
                                              2 emergency
                                              3 reserved
      segment_number          6     bslbf
      for (i=0; i<N; i++){
        message               8     bslbf     Mail message body
      }
    }
    checksum                  8     bslbf
  }
}
```

Parameters

`mail_id`	Unique mail number
`total_segment`	Total number of segments required to carry the whole message. It's a 6-bit variable covering the range [1..63]. Each segment may carry up to 45 bytes.
`priority`	Influences the STB behavior. For example, normal priority would not affect the display, while emergency mail would be displayed on the screen without manual intervention.
`segment_number`	Identifies the current segment. The first segment is equal to 0 and the last segment is equal to total_segment-1.

Notes

1. If the total length of a mail is larger than 45 bytes, then the message is split in several segments, each having the same mail id and consecutive segment numbers. As there is at the most 63 segments of 45 bytes per message, the maximum length of a message is equal to 2835 bytes.

3.2 Force Tune

Description

This command forces the STB to tune to a service defined by the network_id/transport_id/service_id. If the STB is able to query the access rights needed for the service, then the tuning should occur only if the subscriber has access to the service.

Format

```
IRD_command(){
  EMM_command                 8     uimsbf    0x64
  length                      8     uimsbf    13
  command_body(){
    sequence_number           32    uimsbf
    command_id                8     uimsbf    0xC1
    operation                 8     uimsbf    0x01
    data{
      network_id              16    uimsbf
      transport_id            16    uimsbf
      service_id              16    uimsbf
    }
    checksum                  8     bslbf
  }
}
```

Parameters

`network_id` corresponds to the network_id as described in the DVB Network Information Table (NIT).

`transport_id` corresponds to the network_id as described in the DVB Network Information Table (NIT).

`service_id` corresponds to the service_id as described in the DVB Service Description Table (SDT). It may also correspond to the program number found in the MPEG Program Map Table (PMT).

3.3 Set Network ID

Description

This command sets the set-top box network ID to a specific value. This allows the set-top box to retrieve the Network Information Table (NIT) defining the topology of a particular local area. This command can also be used to assign testing network ID to specific set-top boxes.

Format

```
IRD_command(){
  EMM_command                 8     uimsbf    0x64
  length                      8     uimsbf    11
  command_body(){
    sequence_number           32    uimsbf
    command_id                8     uimsbf    0xC6
    operation                 8     uimsbf    0x01
    data{
      network_id              16    uimsbf    Network ID
      original_network_id     16    uimsbf    Original network ID
    }
    checksum                  8     bslbf
  }
}
```

Parameters

network_id	Unique identifier indicating the network ID.
original_network_id	Unique identifier indicating the original network ID.

3.4 Master/Slave

Refer to document [2] for a Master/Slave feature solution overview and document [3] for implementation guidelines.

3.4.1 Continuous Mode Initialization

Description

This command is used to set the parameters in order to initialise the Master/Slave continuous mode.

Format

```
IRD_command(){
  EMM_command                8      uimsbf    0x64
  length                     8      uimsbf    14
  command_body(){
    sequence_number          32     uimsbf
    command_id               8      uimsbf    0xC7
    operation                8      uimsbf    0x01
    data{
      masterSmartcard        32     uimsbf
      validationPeriod       8      uimsbf    in days
      randomPeriod           8      uimsbf    in days
      timeout                8      uimsbf    in hours
    }
    checksum                 8      bslbf
  }
}
```

Parameters

`masterSmartcard` this is the Smartcard ID of the master Smartcard without checksum.

`validationPeriod` this value define the average time, expressed in days, between two validation procedures.

`randomPeriod` the next validation procedure will occur in `validationPeriod` days +/- `randomPeriod` days. The targeted day will be randomly chosen in this bracket of time.

`timeout` the timeout is the period of time during which the customer has to succeed with the validation procedure (insert the master Smartcard in the slave STB). At the end of the `timeout` period, the STB will stop playing video and/or audio signal.

3.4.2 Cancellation

Description

This command id used to disable the IRD Master/Slave mode continuous and single shot mode.

Format

```
IRD_command(){
  EMM_command                8      uimsbf    0x64
  length                     8      uimsbf    7
  command_body(){
    sequence_number          32     uimsbf
    command_id               8      uimsbf    0xC7
    operation                8      uimsbf    0x02
    checksum                 8      bslbf     0x37
  }
}
```

Parameters

None

3.4.3 Single Shot

Description

This command is used to set the parameters in order to initialise the single shot Master/Slave command. This is not possible to disable this command only; all Master/Slave modes must be disabled in order to cancel it. In other words, it's not possible to cancel a single shot command without cancelling the continuous mode.

Format

```
IRD_command(){
  EMM_command                 8     uimsbf    0x64
  length                      8     uimsbf    12
  command_body(){
    sequence_number           32    uimsbf
    command_id                8     uimsbf    0xC7
    operation                 8     uimsbf    0x03
    data{
      masterSmartcard         32    uimsbf
      timeout                 8     uimsbf    in hours
    }
    checksum                  8     bslbf
  }
}
```

Parameters

`masterSmartcard` this is the Smartcard ID of the master Smartcard without checksum.

`timeout` the timeout is the period of time during which the customer has to succeed with the validation procedure (insert the master Smartcard in the slave STB). At the end of the `timeout` period, the STB will stop playing video and/or audio signal.

3.5 Set PIN Code

Description

This command allows the head-end to change the set-top box PIN code. The operation field identifies the PIN code that has to be modified in case the set-top box manages several PIN codes.

Format

```
IRD_command(){
  EMM_command                 8     uimsbf    0x64
  length                      8     uimsbf    8+N
  command_body(){
    sequence_number           32    uimsbf
    command_id                8     uimsbf    0xC8
    operation                 8     uimsbf    0x01..0xFF
    data{
      pin_length              8     uimsbf    PIN length
      for(i=0; i<N; i++){
        character             8     uimsbf    PIN character
      }
    }
    checksum                  8     bslbf
  }
}
```

Parameters

`pin_length`	Number of bytes the PIN code is composed of.
`character`	ASCII code of each character composing the PIN code.

Example

The following command will change the PIN code number 1 to "1234".

```
IRD_command = `640C00000007C801043132333469
```

In this example the 4-byte sequence number is equal to `00000007.

4 Specific IRD Commands

For any specific commands required by a manufacturer that doesn't belongs to the set of generic commands defined in §3, the procedure here after has to be followed:

- The manufacturer has to issue a formal document specifying the format and the behavior of the specific command. The command must comply with the general format defined in §2, but is restricted to the definition of the *operation* and *data* fields:

```
IRD_command(){
  EMM_command              8
  length                   8     uimsbf
  command_body{
    sequence_number        32    uimsbf
    command_id             8     uimsbf
    operation              8     uimsbf
    for(i=0; i < N; i++){
      data                 8     bslbf
    }
    checksum               8     bslbf
  }
}
```

- The specification is provided to NagraVision for approval by sending an email to the following address:

  cak@nagra.com

- NagraVision evaluates the specification to know whether it is acceptable and assign a value to the *command_id* field. This allows to guarantee a global consistency all over the networks and allows to avoid conflicts between different commands.
  NagraVision reserves the right to modify the command and move it in the set of generic commands if its usage suits a wider scope.
- In case the command remains a specific command, the manufacturer updates the specification with the *command_id* assigned by NagraVision and publishes a new version of the document.
- In case the command becomes a generic command, NagraVision updates the present document with the new command and publishes a new version.

If the request for a specific commands comes from an operator instead of a manufacturer, the procedure here above remains the same, except that the specification is written by the operator. It is then provided to manufacturers providing set-top boxes over the operator network for implementation.

—— END OF DOCUMENT ——