Atmel AVR4023: FLIP USB DFU Protocol

Features

- **In System Programming on USB interface**
- **Protocol features**
  - **Read device information**
  - **Read/Write device configuration**
  - **Read/Write internal chip memories**
  - **Read/Write external chip memories**
  - **Security management**
  - **Start application**
- **Atmel® USB DFU**
  - **USB Chapter 9 compliant**
  - **One USB control endpoint required**

8-bit Atmel Microcontrollers

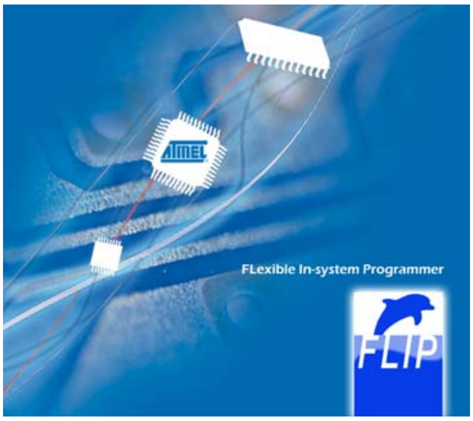
Application Note

1 Introduction

To perform firmware upgrade, Atmel has developed a Flexible in-system programmer (FLIP). This software allows performing In-System Programming from an USB host controller without removing the part from the system or without a pre-programmed application, and without any external programming interface.

Atmel provides USB bootloaders for AVR® parts (both Atmel AVR XMEGA® and UC3 series) including USB hardware interface. These bootloaders use a proprietary USB DFU Protocol, which is described in this application note. By extension this protocol will be named as FLIP protocol, but its covers all host programming application including FLIP, BatchISP, or future Atmel AVR Studio® 5 DFU integration.

Figure 1-1. FLIP.



Rev. 8457A-AVR-01/12

2 Terms and abbreviations

DFU	Device Firmware Upgrade
Firmware	Executable software stored in a write-able, nonvolatile memory on a USB device
FS	USB Full Speed
Upgrade	To overwrite the firmware of a device
	(1) The act of overwriting the firmware of a device
	(2) New firmware intended to replace a device’s existing firmware
Download	To transmit information from host to device
Upload	To transmit information from device to host
USB	Universal Serial Bus
IN	USB transfer packet from device to host
OUT	USB transfer packet from host to device
ZLP	USB Zero Length Packet

3 Related parts

This documentation applies to the all Atmel AVR XMEGA parts with USB module and all ATMEL AVR.

- Atmel AVR XMEGA
- Atmel AVR UC3 Device series

4 Related items

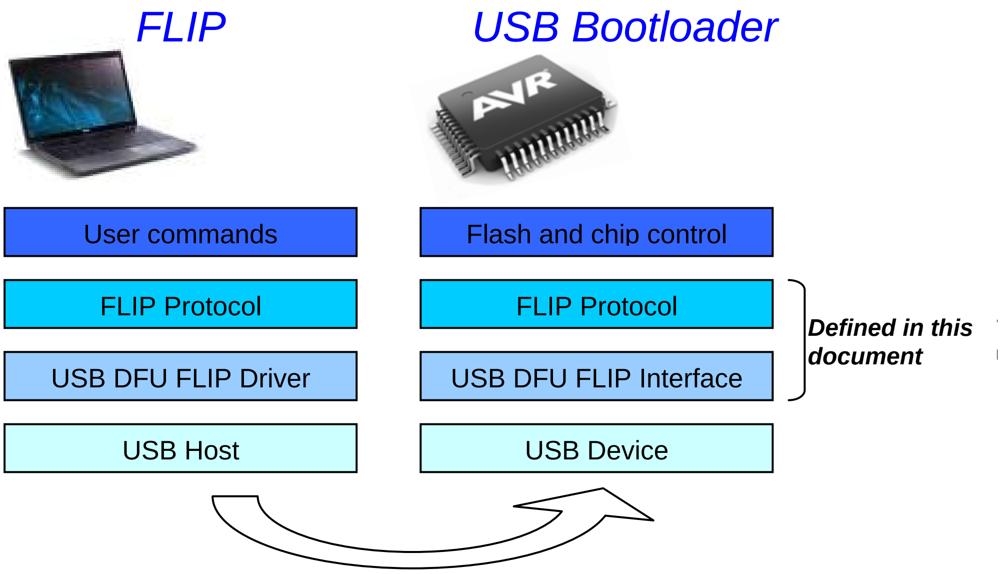
- Atmel FLIP:
    - http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3886
- Atmel AVR Software Framework:
    - http://www.atmel.com/asf
- AVR32784: AVR UC3 USB DFU Bootloader
    - http://www.atmel.com/dyn/resources/prod_documents/doc7745.pdf
- Atmel AVR1916: USB DFU Boot Loader for XMEGA
    - http://www.atmel.com/dyn/resources/prod_documents/doc8429.pdf

5 Overview

The host application software (could be FLIP, BatchISP, or future Atmel AVR Studio 5 extension) receives user input to perform memory operations and translates those requests into a USB communication protocol based on DFU.

The USB bootloader is located in the on-chip Flash memory; it manages the USB communication protocol and performs read/write operations to the on-chip memories.

Figure 5-1. System environment.



The document is divided into the following two sections:

- Atmel USB DFU Class
- Atmel FLIP Protocol

6 Atmel USB DFU Class

6.1 Introduction

The Device Firmware Upgrade (DFU) is a USB class that allows upgrading the on-chip firmware of a USB device. Atmel USB DFU is a vendor class implementation based on part of official USB DFU specification, but does not implement the entire USB DFU class.

The USB DFU FLIP uses only the control endpoint (endpoint 0) and the setup request to communicate with the USB host. The following sections define the USB descriptors and the USB control requests used.

6.2 USB Descriptors Set

The device exports the USB DFU descriptors set, which contains:

- A device descriptor
- A single configuration descriptor
- A single interface descriptor

Table 6-1. USB Device Descriptor.

Offset	Field	Size	Value	Description
0	bLength	1	12h	Size of this descriptor, in bytes
1	bDescriptorType	1	01h	DFU functional descriptor type
2	bcdUSB	2	0100h	USB specification release number in binary coded decimal
4	bDeviceClass	1	00h	See interface
5	bDeviceSubClass	1	00h	See interface
6	bDeviceProtocol	1	00h	See interface
7	bMaxPacketSize0	1	64	Maximum packet size for endpoint zero (limited to 32 due to Host side driver)
8	idVendor	2	03EBh	Atmel Vendor ID
10	idProduct	2	2FXXh	Product ID
12	bcdDevice	2	0x0000	Device release number in binary coded decimal
14	iManufacturer	1	0	Index of string descriptor
15	iProduct	1	0	Index of string descriptor
16	iSerialNumber	1	0	Index of string descriptor
17	bNumConfigurations	1	01h	One configuration only for DFU

The USB configuration descriptor is identical to the standard configuration descriptor described in the USB specification version 1.0, with the exception that the bNumInterfaces field must contain the value 01h.

Table 6-2. USB Interface Descriptor.

Offset	Field	Size	Value	Description
0	bLength	1	09h	Size of this descriptor, in bytes
1	bDescriptorType	1	04h	INTERFACE descriptor type
2	bInterfaceNumber	1	00h	Number of this interface
3	bAlternateSetting	1	00h	Alternate setting
4	bNumEndpoints	1	00h	Only the control pipe is used
5	bInterfaceClass	1	FFh	Vendor specific
6	bInterfaceSubClass	1	00h	No sub class definition
7	bInterfaceProtocol	1	00h	No protocol definition
8	iInterface	1	00h	Index of the String descriptor for this interface

6.3 Specific setup requests

In addition to the USB standard requests, four class-specific requests are used to accomplish the upgrade operations:

Table 6-3. Class-specific requests.

bmRequestType	bRequest	wValue	wIndex	wLength	Data phase
0010 0001b	DFU_DNLOAD	wBlock	0	Length	FLIP Protocol
1010 0001b	DFU_UPLOAD	wBlock	0	Length	FLIP Protocol
1010 0001b	DFU_GETSTATUS	Zero	0	6	Status
0010 0001b	DFU_CLRSTATUS	Zero	0	Zero	none

6.3.1 Device status

Status information is used to ease synchronization between the host application and the USB device. This status gives information on the execution of the previous request.

The device responds to the DFU_GETSTATUS request with a payload packet describe in Table 6-4.

Table 6-4. Device status packet.

Offset	Field	Size	Value	Description
0	bStatus	1	Number	An indication of the status resulting from the execution of the most recent request
1	bwPollTimeOut	3	Number	Not used always 0
4	bState	1	Number	An indication of the state that the device is going to enter immediately following transmission of this response
5	iString	1	Index	Not used always 0

The values of ***bStatus*** and ***bState*** possible are described in Table 6-5.

Table 6-5. Status and state values.

Status ref.	Status	State	Description
STATUS_OK	00h	00h	Command successful and device in IDLE mode
STATUS_STALL	0Fh	0Ah	Specific Setup Request unknown
STATUS_MEM_UNKNOW	03h	0Ah	Read or Write memory access not available
STATUS_MEM_PROTECTED	03h	00h	Memory access protected
STATUS_OUTOFRANGE	08h	0Ah	Address out of range or memory ID unknown
STATUS_BLANK_FAIL	05h	00h	Blank check unsuccessful
STATUS_ERASE_ONGOING	09h	04h	Erase on-going

Each time the device detects and reports an error indication status to the host in response to a DFU_GETSTATUS request, it enters the dfuERROR state. After reporting any error status, the device can not leave the dfuERROR state, until it has received a DFU_CLRSTATUS request. Upon receipt of DFU_CLRSTATUS, the device sets status to IDLE mode.

6.3.2 Command request

Four types of commands use DFU_DNLOAD and DFU_UPLOAD setup requests:

- **Single FLIP Command**
- **Empty command** (only used to validated application start command)
- **FLIP Command with data payload to download**
- **FLIP Command with data payload to upload**

Figure 6-1. Single FLIP command.

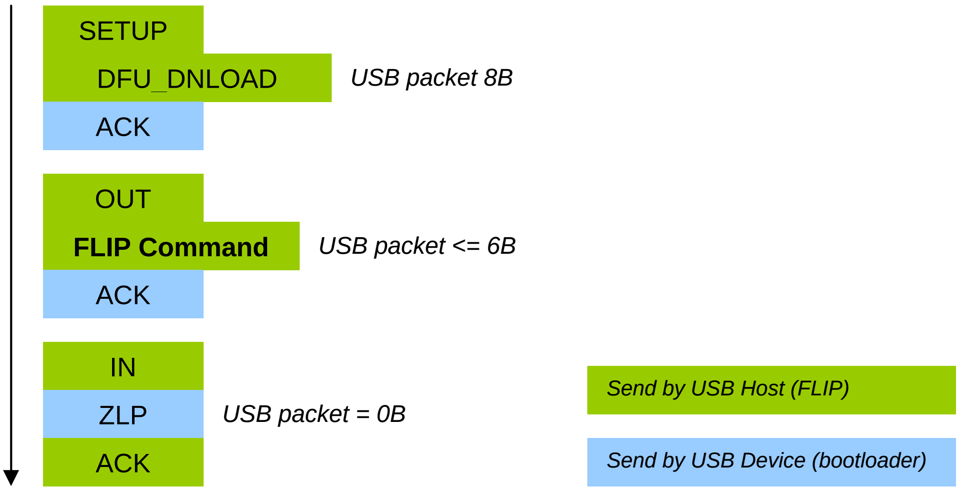


Figure 6-2. Empty command.

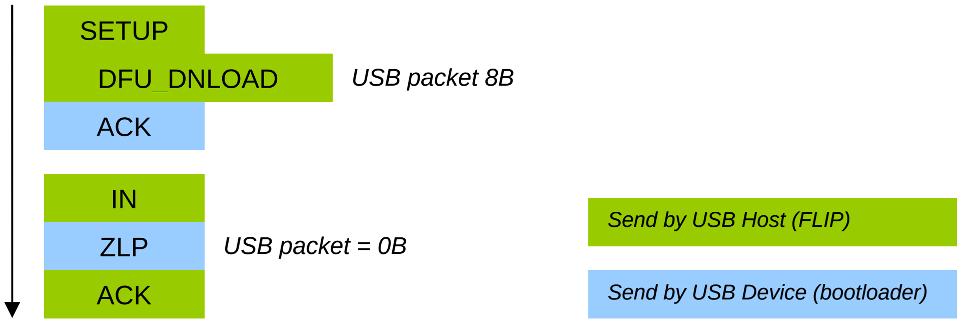


Figure 6-3. FLIP command with data to download.

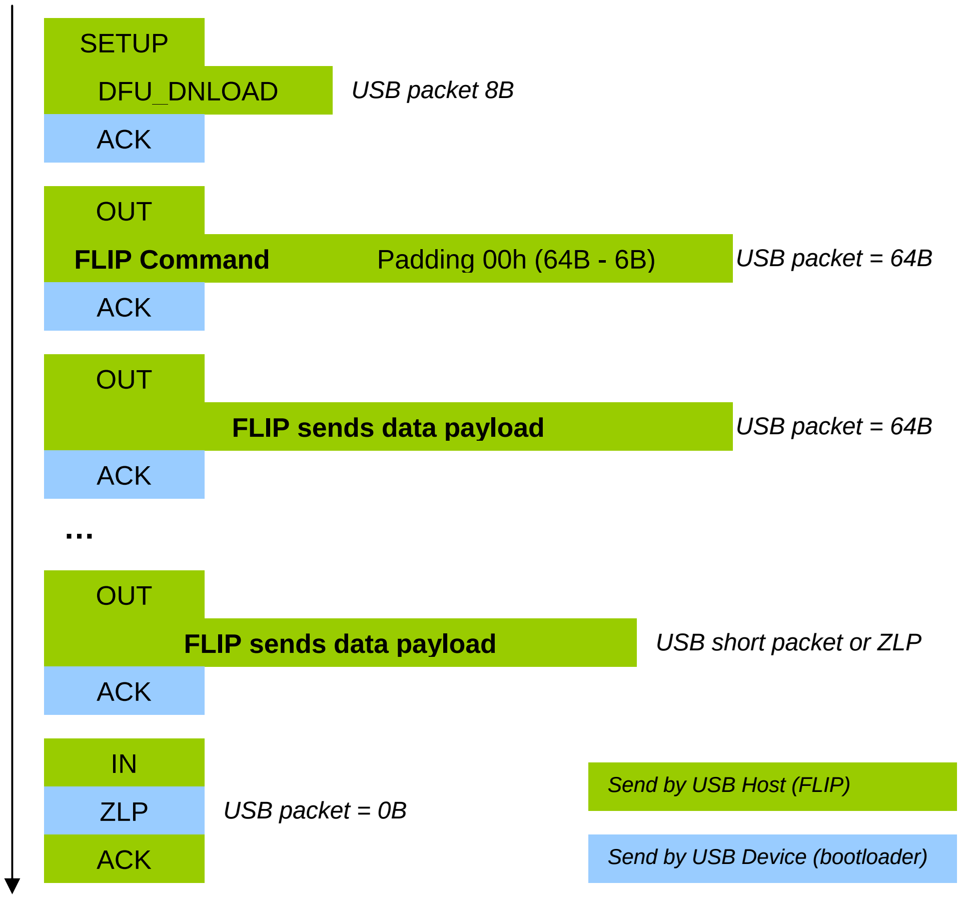
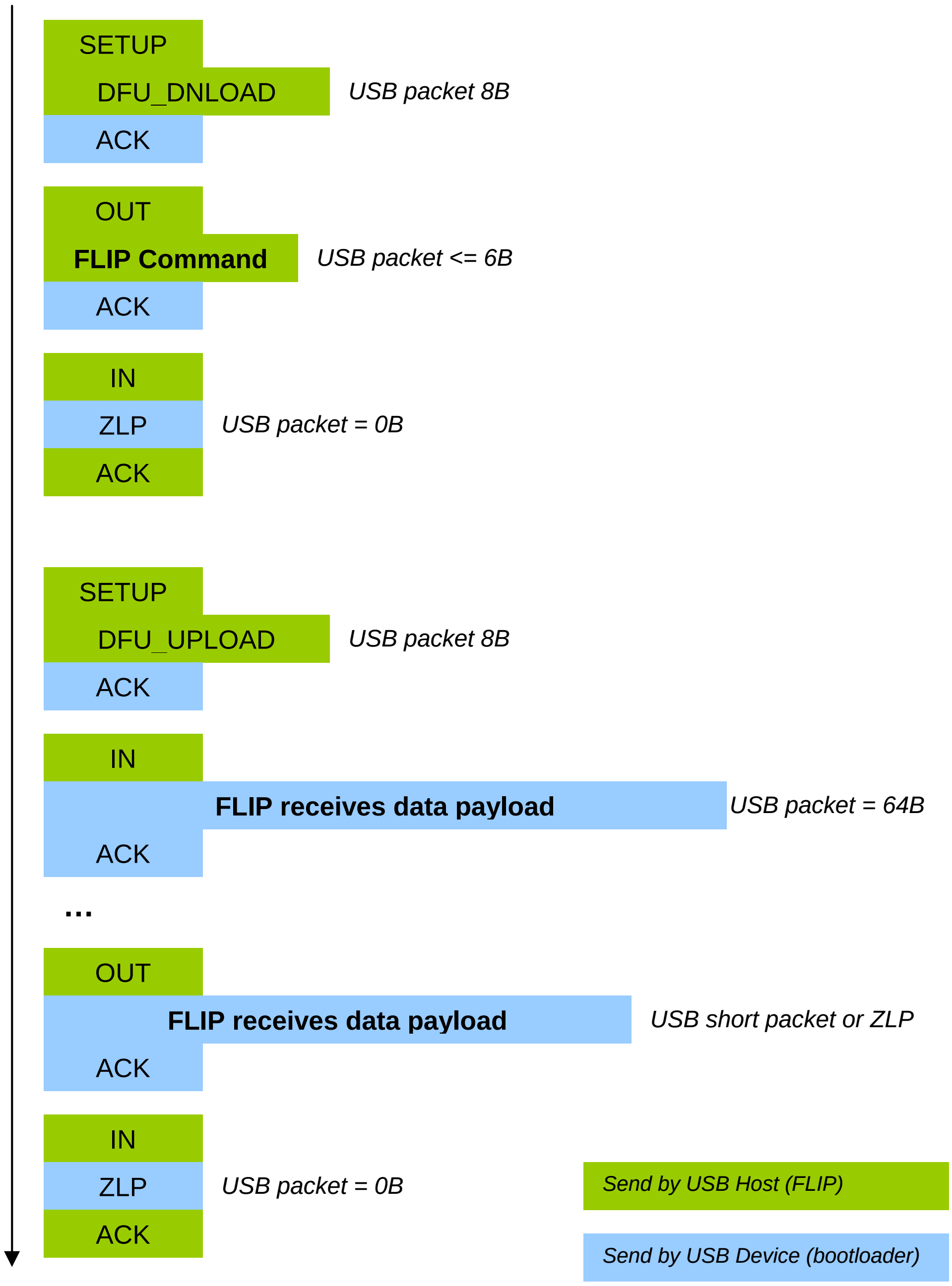


Figure 6-4. FLIP command with data to upload.



7 FLIP protocol

The Atmel FLIP protocol is generic and may support other physical layers than USB.

7.1 Overview

The commands are include in a group command and four bytes are allowed to add argument, see Table 7-1.

Table 7-1. FLIP command structure.

Offset [byte]	Field	Size [byte]
0	Group identifier	1
1	Command identifier	1
2	Arguments	4

There are four groups (Table 7-2) described in following sections. For each commands the FLIP command structure is described, the optional data to upload or download and the status possible after the command sending.

Table 7-2. FLIP group command.

Value [byte]	Group	Description
01h	CMD_GROUP_DOWNLOAD	To program a memory
03h	CMD_GROUP_UPLOAD	To read or check a memory
04h	CMD_GROUP_EXEC	To erase chip or start user application
06h	CMD_GROUP_SELECT	To select a memory and the memory area

7.2 Select group

Prior to any read or program operation, a memory target must be selected as well as the page offset inside this memory.

This is achieved by sending the SELECT_MEMORY_UNIT command and the SELECT_MEMORY_PAGE command.

7.2.1 Selecting memory unit

Table 7-3. Select memory unit command.

Field	Value	Description
Group identifier	06h	Select Group
Command identifier	03h	Select Memory Command
Argument 1	00h	Select Memory unit
Argument 2	Memory Unit	Memory identifier to select see
Argument 3	00h	Reserved
Argument 4	00h	Reserved
Data payload	*None*	*None*

Table 7-4. Memory unit available.

Value	Description
00h	FLASH
01h	EEPROM
02h	SECURITY
03h	CONFIGURATION
04h	BOOTLOADER
05h	SIGNATURE
06h	USER
07h	INT_RAM
08h	EXT_MEM_CS0
09h	EXT_MEM_CS1
0Ah	EXT_MEM_CS2
0Bh	EXT_MEM_CS3
0Ch	EXT_MEM_CS4
0Dh	EXT_MEM_CS5
0Eh	EXT_MEM_CS6
0Fh	EXT_MEM_CS7
10h	EXT_MEM_DF

Table 7-5. Select memory unit status.

Status ref.	Description
STATUS_OK	Command successful and device in IDLE mode
STATUS_OUTOFRANGE	Memory ID unknown

7.2.2 Selecting memory page

Table 7-6. Select memory page command.

Field	Value	Description
Group identifier	06h	Select Group
Command identifier	03h	Select Memory Command
Argument 1	01h	Select Memory page
Argument 2	Page MSB	64KB Memory page number
Argument 3	Page LSB	
Argument 4	00h	Reserved
Data payload	*None*	*None*

Table 7-7. Select memory page status.

Status ref.	Description
STATUS_OK	Command successful and device in IDLE mode
STATUS_OUTOFRANGE	Address out of range

7.3 Download group

The Download Group includes one command, which is used to program the selected memory.

7.3.1 Program start

BatchISP and FLIP have internal ISP buffers (one buffer per device memory). Writing a memory is always done from the buffer contents. The ISP buffer content is irrelevant to the FLIP script user.

The Program Start command transfers the ISP buffer from FLIP to selected memory.

Table 7-8. Program start command.

Field	Value	Description
Group identifier	01h	Download Group
Command identifier	00h	Program Start Command
Argument 1	Page MSB	Start Memory Address
Argument 2	Page LSB	
Argument 3	Page MSB	End Memory Address
Argument 4	Page LSB	
Download Data payload	*FLIP buffer*	*A prefix* [1] *is added at FLIP buffer*

Note: 1. Data Payload Prefix.

In order to be in accordance with the memory write entity (page size), X non-significant bytes may be added before the first byte to program. The X number is calculated to align the beginning of the firmware with the memory write entity.

NOTE Currently FLIP application split her internal buffer to have a data payload size (include prefix) which does not exceed 2KB.

Table 7-9. Program start status.

Status ref.	Description
STATUS_OK	Command successful and device in IDLE mode
STATUS_MEM_UNKNOW	Write memory access not available
STATUS_MEM_PROTECTED	Memory access protected
STATUS_OUTOFRANGE	Address out of range

NOTE If a status error occurs then the data payload to download must be stalled by USB protocol.

7.4 Upload group

This group of commands allows reading the content as well as checking the blank state of the selected memory.

7.4.1 Read memory

BatchISP and FLIP have internal ISP buffers (one buffer per device memory). Reading a memory updates the buffer from the memory contents. During the verify operation, the target memory is read and its contents is compared to the buffer one.

Table 7-10. Read memory command.

Field	Value	Description
Group identifier	03h	Upload Group
Command identifier	00h	Read memory Command
Argument 1	Page MSB	Start Memory Address
Argument 2	Page LSB	
Argument 3	Page MSB	End Memory Address
Argument 4	Page LSB	
Upload Data payload	*Memory content*	*Memory content corresponding at selected memory and Memory address*

NOTE Currently FLIP application split Read Memory command to have a data payload size which does not exceed 1KB.

Table 7-11. Read memory status.

Status ref.	Description
STATUS_OK	Command successful and device in IDLE mode
STATUS_MEM_UNKNOW	Read memory access not available
STATUS_MEM_PROTECTED	Memory access protected
STATUS_OUTOFRANGE	Address out of range

NOTE If STATUS_MEM_UNKNOW or STATUS_OUTOFRANGE occurs, then the ZLP of DOWNLOAD request must be stalled.

NOTE If STATUS_MEM_PROTECTED occurs, then the next UPLOAD request must be stalled by USB protocol.

7.4.2 Blank check memory

During the blank check memory command, the memory content is compared to FFh.

Table 7-12. Blank check memory command.

Field	Value	Description
Group identifier	03h	Upload Group
Command identifier	01h	Blank Check Command
Argument 1	Page MSB	Start Memory Address
Argument 2	Page LSB	
Argument 3	Page MSB	End Memory Address
Argument 4	Page LSB	
Data payload	*None*	*None*

Table 7-13. Blank check status.

Status ref.	Description
STATUS_OK	Command successful and device in IDLE mode
STATUS_MEM_UNKNOW	Read memory access not available
STATUS_OUTOFRANGE	Address out of range
STATUS_BLANK_FAIL	Blank check unsuccessful

NOTE If STATUS_MEM_UNKNOW or STATUS_OUTOFRANGE occurs, then the ZLP of DOWNLOAD request must be stalled.

7.5 Excec group

This group of commands allows to erase the whole Flash memory or to start the application.

7.5.1 Chip erase

The Chip erase command erases the whole Flash Memory.

Table 7-14. Chip erase command.

Field	Value	Description
Group identifier	04h	Exec Group
Command identifier	00h	Erase Command
Argument 1	FFh	Chip erase
Argument 2	00h	Reserved
Argument 3	00h	Reserved
Argument 4	00h	Reserved
Data payload	*None*	*None*

Table 7-15. Chip erase status.

Status ref.	Description
STATUS_OK	Command successful and device in IDLE mode
STATUS_ERASE_ONGOING	Erase on-going. The Chip erase command must be resend to finish erase.

7.5.2 Starting the application

The Start Application command resets the device and the application is started.

Table 7-16. Start application command.

Field	Value	Description
Group identifier	04h	Exec Group
Command identifier	03h	Start Application Command
Argument 1	00h	Hardware Reset
Argument 2	00h	Reserved
Argument 3	00h	Reserved
Argument 4	00h	Reserved
Data payload	*None*	*None*

Table 7-17. Start application status.

Status ref.	Description
STATUS_OK	Command successful and device in IDLE mode

NOTE

To complete the Start Application command, an empty FLIP command must be send (see Figure 6-2).

8 Table of contents

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Milennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
16F, Shin Osaki Kangyo Bldg.
1-6-4 Osaki Shinagawa-ku
Tokyo 104-0032
JAPAN
Tel: (+81) 3-6417-0300
Fax: (+81) 3-6417-0370

© 2012 Atmel Corporation. All rights reserved.

Atmel®, Atmel logo and combinations thereof, AVR®, AVR Studio®, XMEGA®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.