General Bootloader frame format

- **Little Endiannes**
- **Binary Interface**

byte	*0*	*1*	*2*	*3*	*4*	*5*
	Bootloader Header					

Bootloader Header Definition

byte	*0*	*1*	*2*	*3*	*4*	*5*
	Bootloader Header					
field	**Preamble**		**Source**	**Command**	**Status**	**Len**
value	*0xB0*	*0x07*	*src[7:0]*	*cmd[7:0]*	*stat[7:0]*	*len[7:0]*

Field	Description
Preamble	Fixed value preamble: "***\xB0\x07***"
Source	**Message source** ***0x2B*** : To bootloader, message flow is from PC or other system ***0xB2*** : From bootloader, message data flow goes from bootload system ***other values*** : Ignore
Command	**Command type** ***0x10:*** Connect ***0x11*** **:** Connect response ***0x20*** **:** Prepare ***0x21*** **:** Prepare response ***0x30*** **:** Flash data ***0x31*** **:** Flash data response ***0x41*** : Exit (jump to application) ***0x41:*** Exit response ***0xA0*** : Information command ***0xA1*** : Information response command ***other values*** **:** Ignore

Status	**Message Status** Message status field is being used as a response to command. ***0x00*** : Normal operation (no error) ***0x01*** : Validation error ***0x02*** : Invalid request (wrong sequence) ***0x04*** : Writing to FLASH error ***0x08*** : Preparing (erasing) FLASH error ***0x10*** : Firmware image size error ***0x20*** : Firmware compatibility error ***0x40*** : Reserved ***0x80*** : Reserved
Length	**Payload length** in bytes
CRC	**CRC checksum** of following fields: - Source - Command - Status - Length - Payload (in case of "*Flash data* " command) **CRC-8 calculation details:** - Polynom: *0x07* (CRC-8-CCITT) - Custom seed: *0xB6*

Connect command

byte	*0*	*1*	*2*	*3*	*4*	*5*
	Bootloader Header					
field	**Preamble**		**Source**	**Command**	**Status**	**Len**
value	*0xB0*	*0x07*	*0x2B*	*0x10*	*0x00*	*0x00*

Event	**Description**
On Transmit	The Connect command is being generated by Bootloader Manag
On Receive	On receip of this command bootloader shall evaluate command 1. If bootloader is in "*IDLE* " state, then it shall return "***OK*** " stat 2. If bootloader is not in "*IDLE* " state, then it shall return "***Inva*** Successful command will result to stay in bootloader, until "*boo*

Connect response command

	OK RESPONSE					
byte	*0*	*1*	*2*	*3*	*4*	*5*

	Bootloader Header					
field	Preamble		Source	Command	Status	Len
value	0xB0	0x07	0xB2	0x11	0x00	0x00

Event	Description
On Transmit	Connect Response command is being generated by Bootloader
On Receive	On receipt of that command Bootloader Manager (PC Applicatic 1. If response is "*OK* ", then continue with FW upgrade procedu 2. If response is not "*OK* ", then cancel FW upgrade procedure a

Prepare command

byte	0	1	2	3	4	5
	Bootloader Header					
field	Preamble		Source	Command	Status	Len
value	0xB0	0x07	0x2B	0x20	0x00	0x0C

Field	Description
CRC	CRC-8 calculations based on following fields: ***- Source*** ***- Command*** ***- Status*** ***- Length*** ***- New FW size*** ***- New FW version*** ***- New FW hardware version***
New FW image size	Size of new firmware image size in bytes. *NOTE: That information can be used (configurable via BOOT_CF*
New FW version	New firmware image application version. Version is coded as **he** ***- MM:*** *major software number,* ***- mm:*** *minor software number,* ***- dd:*** *develop software number,* ***- tt:*** *test software number* *NOTE: FW version can be used (configurable via BOOT_CFG_FW* *abort the fw upgrade procedure.*
New FW hardware version	New firmware image hardware support version. Version is code ***- MM:*** *major software number,* ***- mm:*** *minor software number,* ***- dd:*** *develop software number,* ***- tt:*** *test software number* *NOTE: Hardware support version can be used (configurable via E* *exsisting hardware, bootloader shall abort the fw upgrade proce*

Event	Description
On Transmit	The Prepare command is being generated by Bootloader Manag

On Receive	On receip of that command bootlaoder shall evaluate commanc 1. If bootloader is not in "*IDLE* " state, then it shall return "***Inva*** 2. if checking for FW image is enable (***BOOT_CFG_FW_SIZE_CH*** 3. if checking for FW version is enable (***BOOT_CFG_FW_VER_CI*** 4. If checking for HW version is enable(***BOOT_CFG_HW_VER_C*** 5. If preparing (erasing) application flash region result in error,

Prepare response command

OK RESPONSE						
byte	*0*	*1*	*2*	*3*	*4*	*5*
	Bootloader Header					
field	**Preamble**		**Source**	**Command**	**Status**	**Len**
value	*0xB0*	*0x07*	*0xB2*	*0x21*	*0x00*	*0x00*

Event	**Description**
On Transmit	Prepare response command is being generated by Bootloader (
On Receive	On receip of that command Bootloader Manager (PC app) shall 1. If response is "*OK* ", then continue with FW upgrade procedu 2. If response is not "*OK* ", then cancel FW upgrade procedure a

Flash Data command

byte	*0*	*1*	*2*	*3*	*4*	*5*
	Bootloader Header					
field	**Preamble**		**Source**	**Command**	**Status**	**Len**
value	*0xB0*	*0x07*	*0x2B*	*0x30*	*0x00*	*LEN[7:0]*

Field	**Description**
CRC	CRC-8 calculations based on following fields: ***- Source*** ***- Command*** ***- Status*** ***- Length*** ***- Binary data of new firmware***

Binary data of new firmware	Binary stream of new firmware image

Event	Description
On Transmit	The Flash Data command is being generated by Bootloader Mar
On Receive	On receip of that command bootloader shall evaluate commanc 1. If bootloader is not in "*FLASHING* " state, then it shall return 2. If bootloader is in "*FLASHING* " state, then it shall continue tc

Flash Data response command

				OK RESPONSE		
byte	*0*	*1*	*2*	*3*	*4*	*5*
	Bootloader Header					
field	**Preamble**		**Source**	**Command**	**Status**	**Len**
value	*0xB0*	*0x07*	*0xB2*	*0x31*	*0x00*	*0x00*

Event	Description
On Transmit	Flash Data response command is being generated by Bootloade
On Receive	On receip of that command Bootloader Manager (PC app) shall 1. If response is "*OK* ", then continue with FW upgrade procedu 2. If response is not "*OK* ", then cancel FW upgrade procedure a

Exit command

byte	*0*	*1*	*2*	*3*	*4*	*5*
	Bootloader Header					
field	**Preamble**		**Source**	**Command**	**Status**	**Len**
value	*0xB0*	*0x07*	*0x2B*	*0x40*	*0x00*	*0x00*

Event	Description
On Transmit	Exit command shall be generated after final "*Flash Data* " comm
On Receive	On receip of that command bootloader shall evaluate commanc 1. If bootloader is not in "*FLASHING* " state, then it shall return 2. If bootloader is in "*FLASHING* " state, then it shall validate ne *NOTE: Validation of new fw image placed in microcontroller inte*

Exit response command

				OK RESPONSE		
byte	*0*	*1*	*2*	*3*	*4*	*5*
	Bootloader Header					

field	Preamble		Source	Command	Status	Len
value	*0xB0*	*0x07*	*0xB2*	*0x41*	*0x00*	*0x00*

Event	Description
On Transmit	Exit response command is being generated by Bootloader (emb
On Receive	On receip of that command Bootloader Manager (PC app) shall 1. If response is "*OK* ", then FW upgrade is finished with succes 2. If response is not "*OK* ", then cancel FW upgrade procedure a

Information command

byte	*0*	*1*	*2*	*3*	*4*	*5*
	Bootloader Header					
field	**Preamble**		**Source**	**Command**	**Status**	**Len**
value	*0xB0*	*0x07*	*0x2B*	*0xA0*	*0x00*	*0x00*

Event	Description
On Transmit	Information command shall be generated by Bootloader Manag
On Receive	On receip of that command bootloader shall evaluate commanc 1. If bootloader is not in "*IDLE* " state, then it shall return "***Inva*** 2. If bootloader is in "*IDLE* " state, then it shall return with "*Info*

Information response command

						OK RESPONSE
byte	*0*	*1*	*2*	*3*	*4*	*5*
	Bootloader Header					
field	**Preamble**		**Source**	**Command**	**Status**	**Len**
value	*0xB0*	*0x07*	*0xB2*	*0xA1*	*0x00*	*0x04*

Field	Description
CRC	CRC-8 calculations based on following fields: ***- Source*** ***- Command*** ***- Status*** ***- Length*** ***- Bootloader version***
Bootloader version	New firmware image hardware support version. Version is code - ***MM*** *: major software number,* - ***mm*** *: minor software number,* - ***dd*** *: develop software number,* - ***tt*** *: test software number*

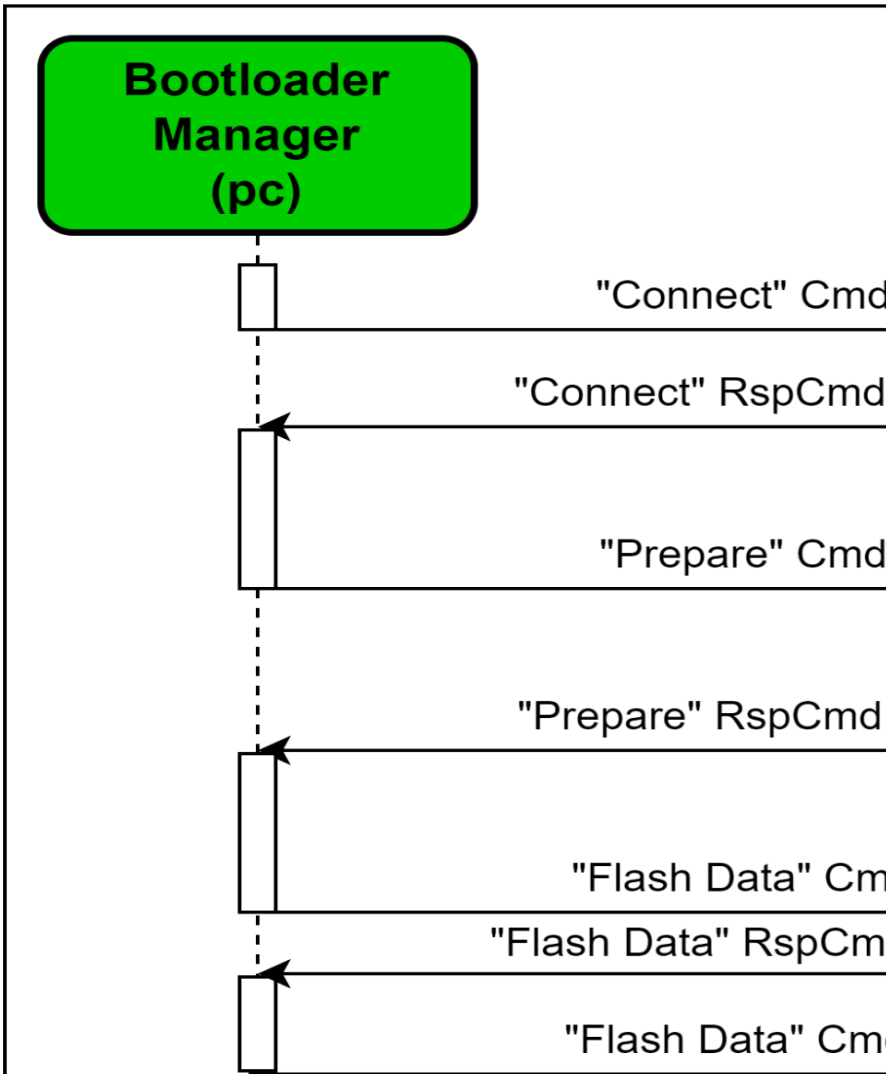
Event	Description
On Transmit	Information response command is being generated by Bootload

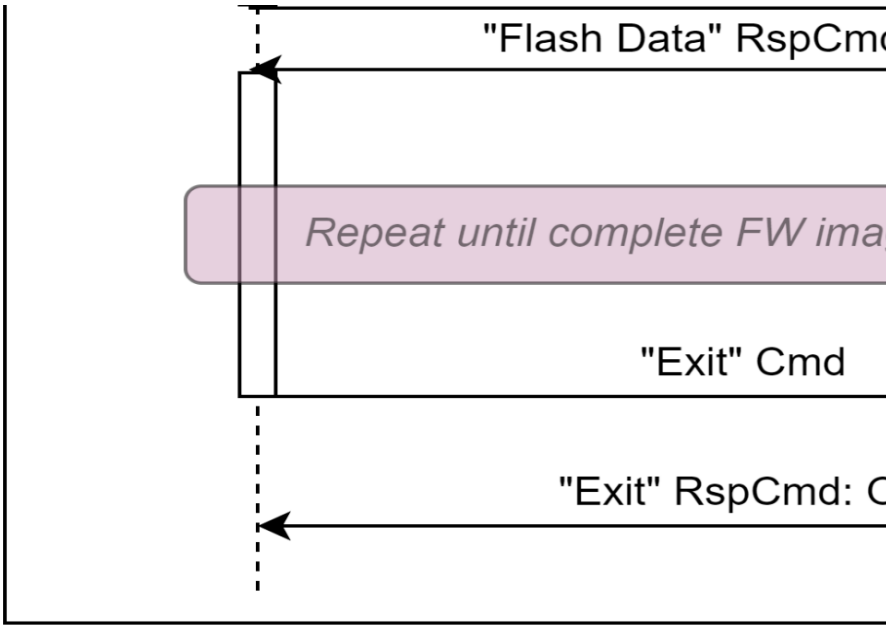
On Receive	On receip of that command Bootloader Manager (PC app) shall 1. If response is "OK", then bootloader information can be obta 2. If response is not "OK", then FW upgrade procudere do not f

6	7	8	9	10	11	...	N
		Bootloader Payload					

6	7
gth	**CRC**
len[15:8]	*crc[7:0]*

to bootloader,
ler to PC or other





6	7
gth	**CRC**
0x00	*0xAF*

ger (PC application) when starting a bootloading procedure.
and response according to:
us
lid request " error

tloader exit timeout " time passes.

					INVALID REQUEST ERRO
byte	*0*	*1*	*2*	*3*	
6	*7*				

gth	**CRC**
0x00	*0x67*

				Bootloade
field	**Preamble**		**Source**	**Command**
value	*0xB0*	*0x07*	*0xB2*	*0x11*

(embedded) on reception of *Connect* command.

on) shall evalute command and response according to:

ure

and report error

6	*7*	*8*	*9*	*10*	*11*	*12*	*13*
							Bootloade
gth	**CRC**	**New FW size**					**New FW**
0x00	*crc[7:0]*	*size[7:0]*	*size[15:8]*	*size[23:16]*	*size[31:24]*	*tt[7:0]*	*dd[15:8]*

G_FW_SIZE_CHECK_EN) for checking if bootloader can fit new FW into reserved flash memory space!

exadecimal format 0xMMmmddtt, where:

_VER_CHECK_EN) for checking application compatibility with bootloader. In case bootloader do not sup

ed as **hexadecimal format 0xMMmmddtt**, where:

BOOT_CFG_HW_VER_CHECK_EN) for checking that new firmware is compatible with hardware. In case edure.

ger (PC application) after successfull connection to bootloader.

d and response according to:
lid request " error,
ECK_EN), then it shall check if new FW image fits into application flash size and return "***Firmware ima***
HECK_EN), then it shall check for new FW image compatibility and return "***Firmware compatibility err***
HECK_EN), then it shall check for HW compatibility and return "***Firmware compatibility errorr*** " status
then it shall return "***Preparing (erasing) FLASH error*** " status

6	7
gth	**CRC**
0x00	*0xCE*

INVALID REQUEST ERRO

byte	*0*	*1*	*2*	*3*
				Bootloade
field	**Preamble**		**Source**	**Command**
value	*0xB0*	*0x07*	*0xB2*	*0x21*

FIRMWARE COPATIBILITY ER

byte	*0*	*1*	*2*	*3*
				Bootloade
field	**Preamble**		**Source**	**Command**
value	*0xB0*	*0x07*	*0xB2*	*0x21*

embedded) on reception of *Prepare* command.
evaluate command and response according to:
ıre
and report error

6	7	8	9	…	LEN
		Bootloader Payload			
gth	**CRC**	**Binary data of new firmware**			
LEN[15:8]	*crc[7:0]*	*data[0]*	*data[1]*	…	*data[LEN]*

nager (PC application) after successfull preparation phase (flash erase).
d and response according to:
"***Invalid request*** " error,
o write to flash

6	*7*
gth	**CRC**
0x00	*0xA9*

				INVALID REQUEST ERROR
byte	*0*	*1*	*2*	*3*
				Bootloade
field	**Preamble**		**Source**	**Command**
value	*0xB0*	*0x07*	*0xB2*	*0x31*

r (embedded) after write binary data to flash. Writing to flash is triggered on reception of Flash Data c
evaluate command and response according to:
ıre
and report error

6	*7*
gth	**CRC**
0x00	*0x53*

nand is executed, after complete fw image transfer.
d and response according to:
"***Invalid request*** " error,
w FW image in internal microprocessor flash. If validtion is "*OK* ", then is shall return status "***OK*** ", othe
rnal flash is based on CRC checksum.

6	*7*

				INVALID REQUEST ERROR
byte	*0*	*1*	*2*	*3*
				Bootloade

gth	CRC
0x00	*0x9B*

field	Preamble		Source	Command
value	*0xB0*	*0x07*	*0xB2*	*0x41*

edded) on reception of *Exit* command.
evaluate command and response according to:
s
and report error

6	7
gth	CRC
0x00	*0x37*

er (pc app) in order to acquire bootloader FW version.
d and response according to:
lid request " error,
rmation response " command

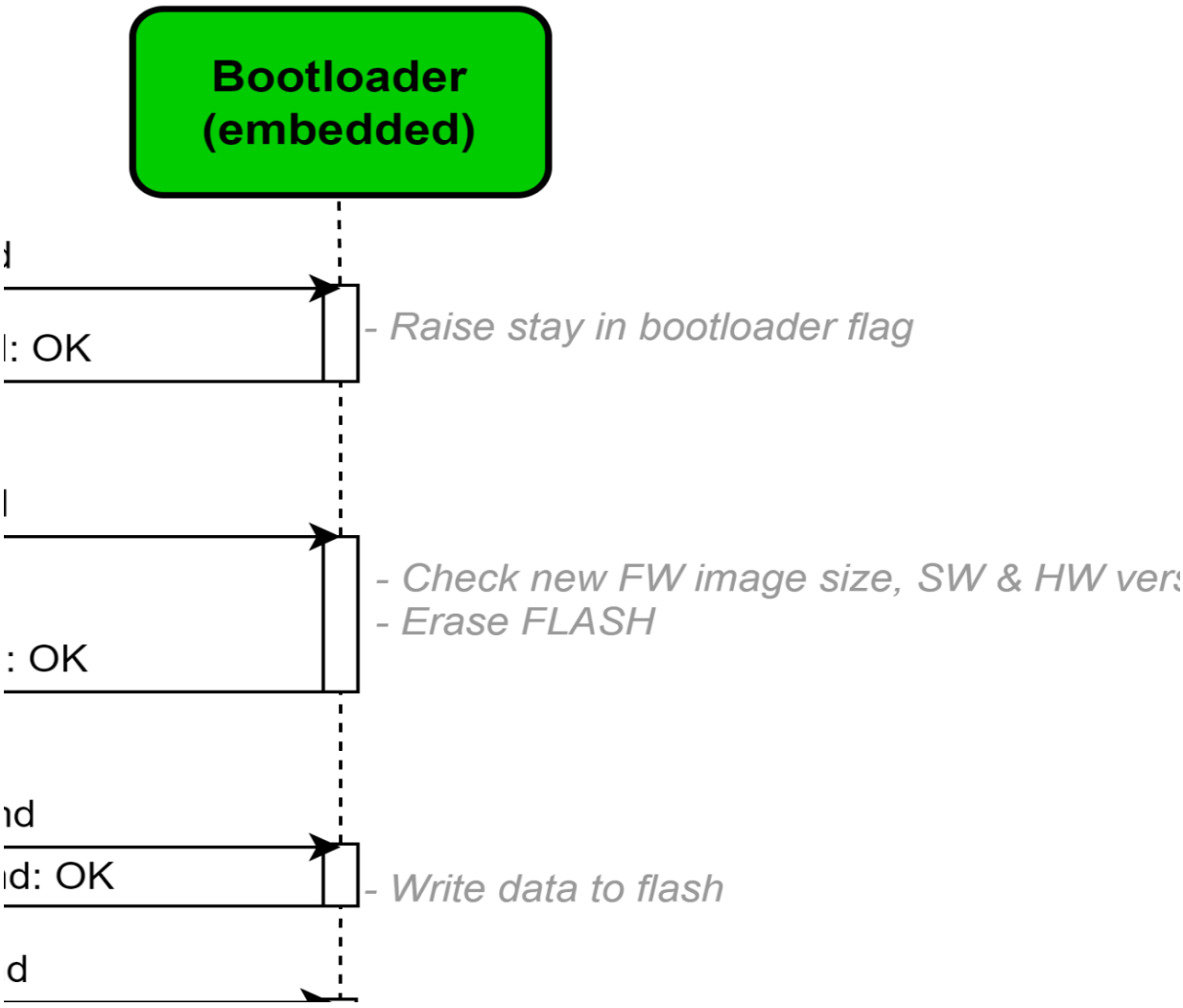
6	7	8	9	10	11
		Bootloader Payload			
gth	CRC	**Bootloader version**			
0x00	*crc[7:0]*	*tt[7:0]*	*dd[15:8]*	*mm[23:16]*	*MM[31:24]*

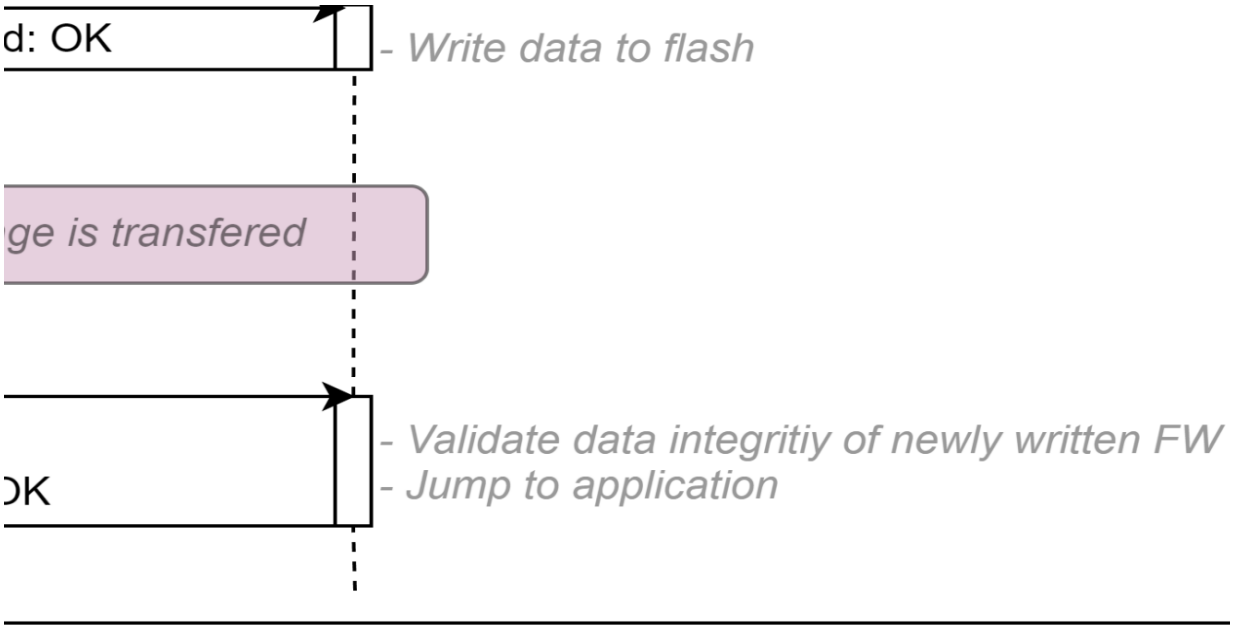
byte
field
value

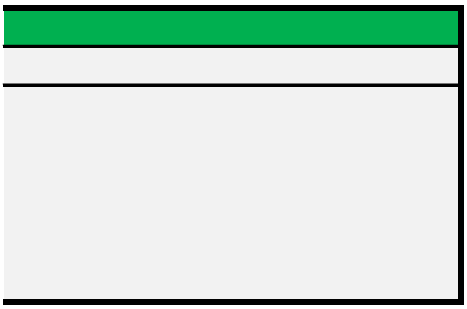
d as hexadecimal format 0xMMmmddtt, where:

ler (embedded) on reception of *Information* command.

evaluate command and response according to: ıin ollow bootloader sequence


Bootloader
(embedded)
- Raise stay in bootloader flag
- Check new FW image size, SW & HW vers
- Erase FLASH
- Write data to flash





R RESPONSE

4	5	6	7

er Header			
Status	**Length**		**CRC**
0x02	*0x00*	*0x00*	*0xB1*

14	*15*	*16*	*17*	*18*	*19*
r Payload					
version		**New FW hardware version**			
mm[23:16]	*MM[31:24]*	*tt[7:0]*	*dd[15:8]*	*mm[23:16]*	*MM[31:24]*

pport given application version it shall

new firmware is not build for

```
ge size error " status,
or " status,
,
```

R RESPONSE

4	5	6	7
er Header			
Status	**Length**		**CRC**
0x02	0x00	0x00	0x18

FIRM

byte	0	1
field	**Preamble**	
value	0xB0	0x07

ROR RESPONSE

4	5	6	7
er Header			
Status	**Length**		**CRC**
0x20	0x00	0x00	0x8D

PR

byte	0	1
field	**Preamble**	
value	0xB0	0x07

R RESPONSE			
4	*5*	*6*	*7*
er Header			
Status	**Length**		**CRC**
0x02	*0x00*	*0x00*	*0x7F*

WR		
byte	*0*	*1*
field	**Preamble**	
value	*0xB0*	*0x07*

ommand.

erwise "***Validation error***"

R RESPONSE			
4	*5*	*6*	*7*
er Header			

byte	*0*	*1*

Status	Length		CRC
0x02	*0x00*	*0x00*	*0x4D*

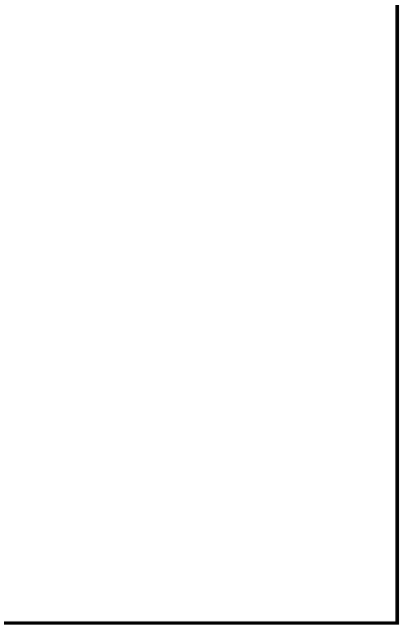
field	Preamble	
value	*0xB0*	*0x07*

INVALID REQUEST ERROR RESPONSE

0	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Bootloader Header							
Preamble		**Source**	**Command**	**Status**	**Length**		**CRC**
0xB0	*0x07*	*0xB2*	*0xA1*	*0x02*	*0x00*	*0x00*	*0x29*

sion compatibility

Validat



WARE IMAGE SIZE ERROR RESPONSE

2	3	4	5	6	7
Bootloader Header					
Source	**Command**	**Status**	**Length**		**CRC**
0xB2	*0x21*	*0x10*	*0x00*	*0x00*	*0x6C*

EPARING FLASH ERROR RESPONSE

2	3	4	5	6	7
Bootloader Header					
Source	**Command**	**Status**	**Length**		**CRC**
0xB2	*0x21*	*0x08*	*0x00*	*0x00*	*0x9F*

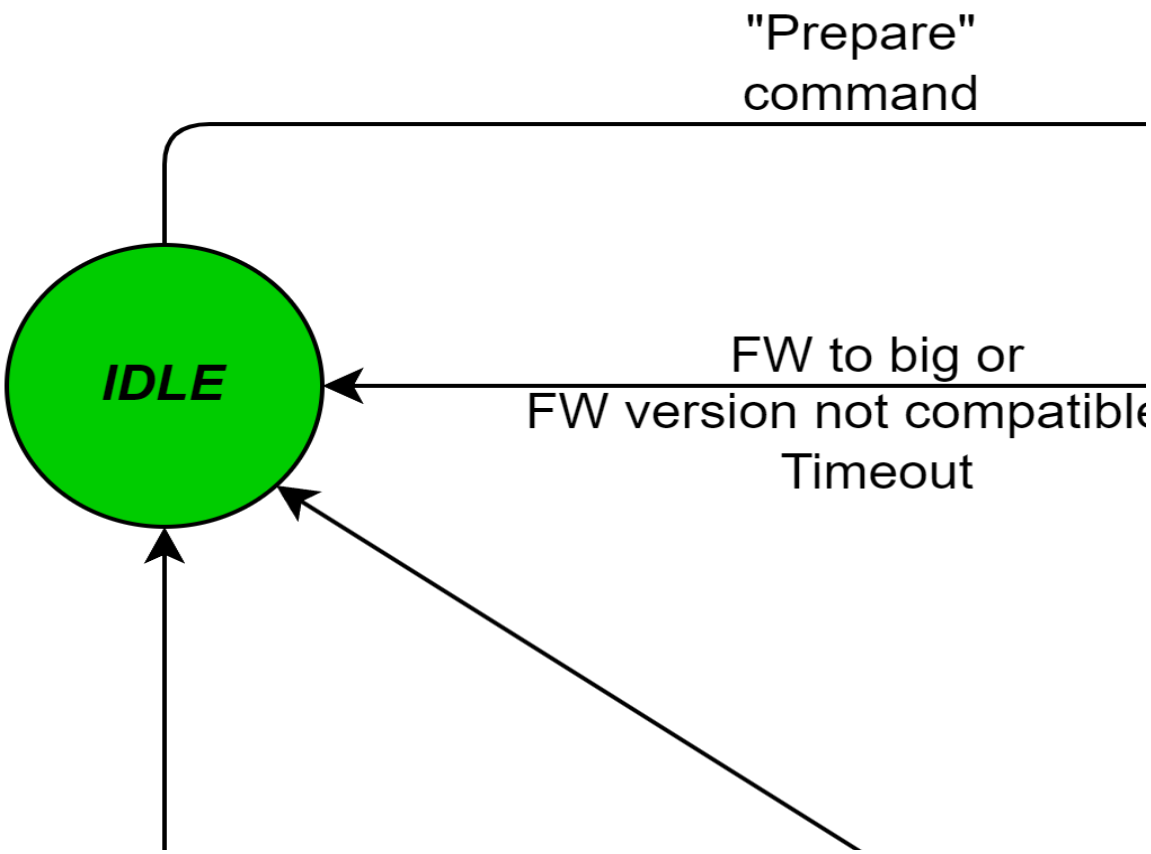
ITING TO FLASH ERROR RESPONSE

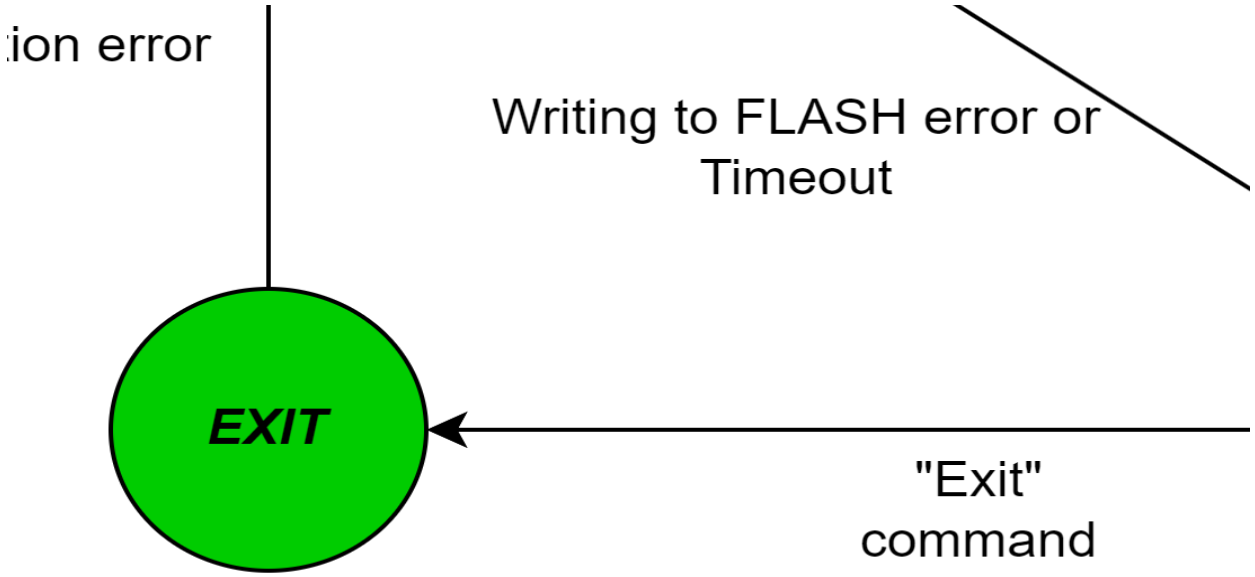
2	3	4	5	6	7
Bootloader Header					
Source	**Command**	**Status**	**Length**		**CRC**
0xB2	*0x31*	*0x04*	*0x00*	*0x00*	*0x02*

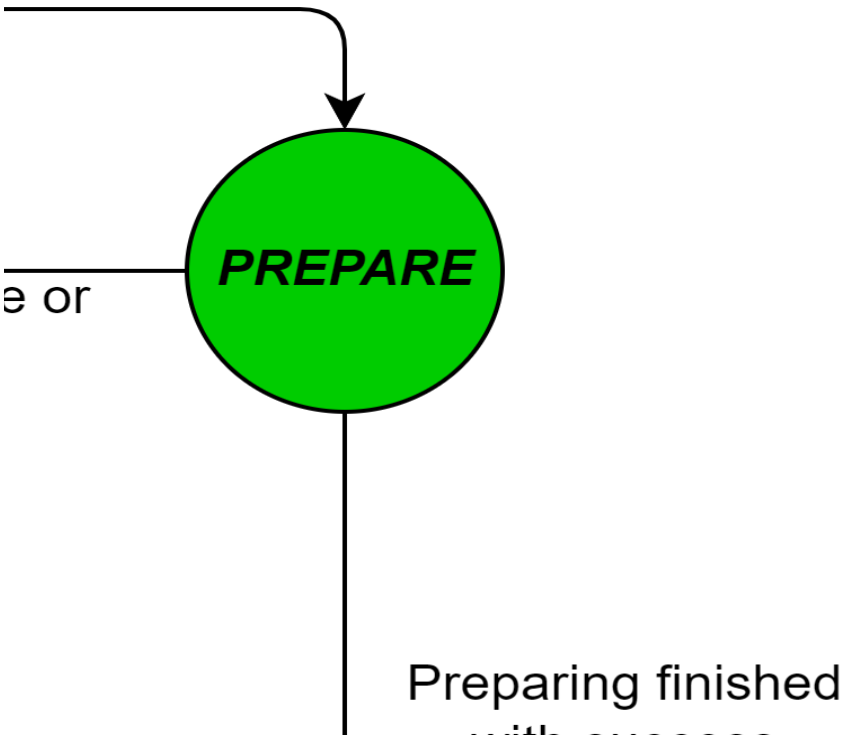
VALIDATION ERROR RESPONSE

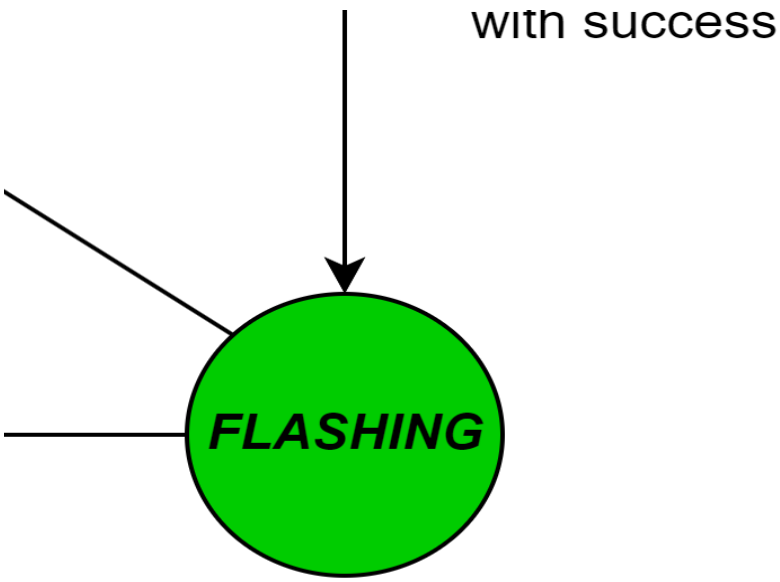
2	3	4	5	6	7
Bootloader Header					

Source	Command	Status	Length		CRC
0xB2	*0x41*	*0x01*	*0x00*	*0x00*	*0xF0*


"Prepare"
command
IDLE
FW to big or
FW version not compatible
Timeout


ion error
Writing to FLASH error or
Timeout
EXIT
"Exit"
command


e or
PREPARE
Preparing finished


with success
FLASHING