What’s a bootloader and why its needed ??

Bootloader :

Bootloader is nothing but a small piece of code stored in the MCU flash or ROM to act as an application loader as well as a mechanism to update the applications whenever required.

ATmega328P

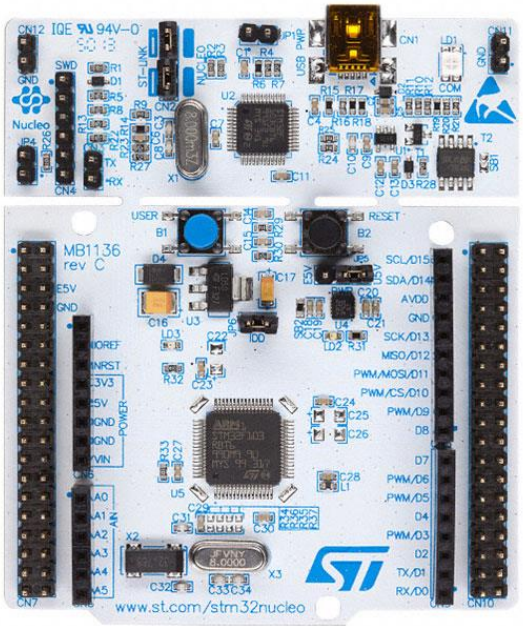
Does this MCU come with on chip Bootloader ?
Yes it is !

Does it run whenever MCU undergoes reset ?
Yes. Upon reset, Arduino bootloader runs first

What's the main use of bootloader ?
To Download Arduino sketches to the board.(IAP)

STM32f446RE

STM32f446RE Nucleo 64



Does this MCU come with on chip Bootloader ?

Yes it is !

Does it run whenever MCU undergoes reset ?

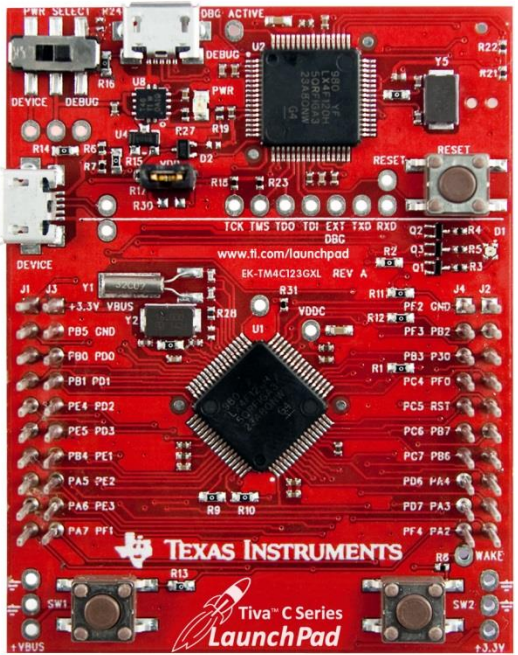
No. Should activate changing the status of boot pins

What's the main use of bootloader ?

To Download/Upload binaries (IAP)

TM4C123G

TM4C123G LaunchPad



Does this MCU come with on chip Bootloader ?

Yes. TivaWare Bootloader

Does it run whenever MCU undergoes reset ?

No. Should activate changing the status of boot pins

What's the main use of bootloader ?

To Download/Upload binaries (IAP)

STM32F42XXX Memory Organization

- Internal Flash memory also called as Embedded Flash memory of 2MB
- Internal SRAM1 of 112KB
- Internal SRAM2 of 16KB
- Internal SRAM3 of 64KB
- System Memory (ROM) of 30KB
- OTP memory of 528 bytes
- Option bytes memory of 2*16bytes.
- Backup RAM of 4KB

Internal Flash memory

- Size is 512KB
- Begins @ 0x0800 0000
- Ends @ 0x081F FFFF
- Used to store your application code and read only data of the program
- Non volatile

Internal SRAM1

- Size is 112KB
- Begins @ 0x2000_0000
- Ends @ 0x2001_BFFF
- Used to store your application global data, static variables
- Also used for Stack and Heap Purpose
- Volatile
- You can also execute code from this memory

Internal SRAM2

- Size is 16KB
- Begins @ 0x2001_C000
- Ends @ 0x2001_FFFF
- Used to store your application global data, static variables
- Also can be used for Stack and Heap Purpose
- Volatile
- You can also execute code from this memory

Internal SRAM3

- Size is 64KB
- Begins @ 0x2002_0000
- Ends @ 0x2002_FFFF
- Used to store your application global data, static variables
- Also can be used for Stack and Heap Purpose
- Volatile
- You can also execute code from this memory

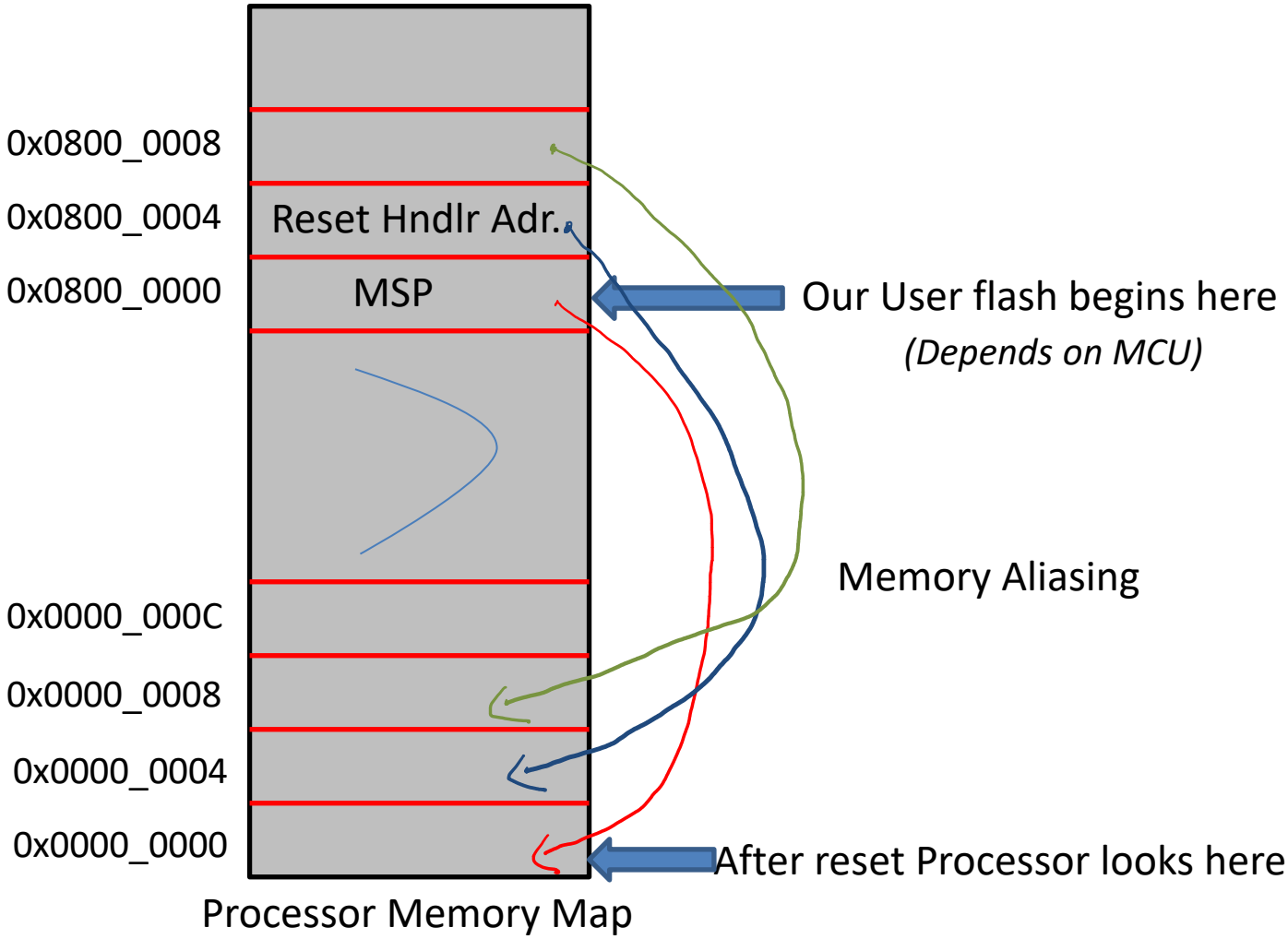
All ARM Cortex M Based MCUs right after reset does,

1) Load value @ Memory addr. **0x0000_0000** in to MSP
2) Load value @ Memory addr. **0x0000_0004** in to PC (Value = Addr of the reset handler)

In STM32 Microcontroller ,

1) MSP value stored at **0x0800_0000**
2) Vector table starts from **0x0800_0004**
3) Address of the reset handler found at **0x0800_0004**

So , Don't you think we should somehow link 0x0800_0000 to 0x0000_0000 ??


0x0800_0008
0x0800_0004
Reset Hndlr Adr.
0x0800_0000
MSP
Our User flash begins here
(Depends on MCU)
Memory Aliasing
0x0000_000C
0x0000_0008
0x0000_0004
0x0000_0000
After reset Processor looks here
Processor Memory Map

System Memory (ROM)

- Size is 30KB
- Begins @ 0x1FFF_0000
- Ends @ 0x1FFF_77FF
- All the ST MCUs store Bootloader in this memory
- This Memory is Read only
- By default MCU will not execute any code from this memory but you can configure MCU to boot or execute bootloader from this memory .

Boot Configuration of STM32F42xxx

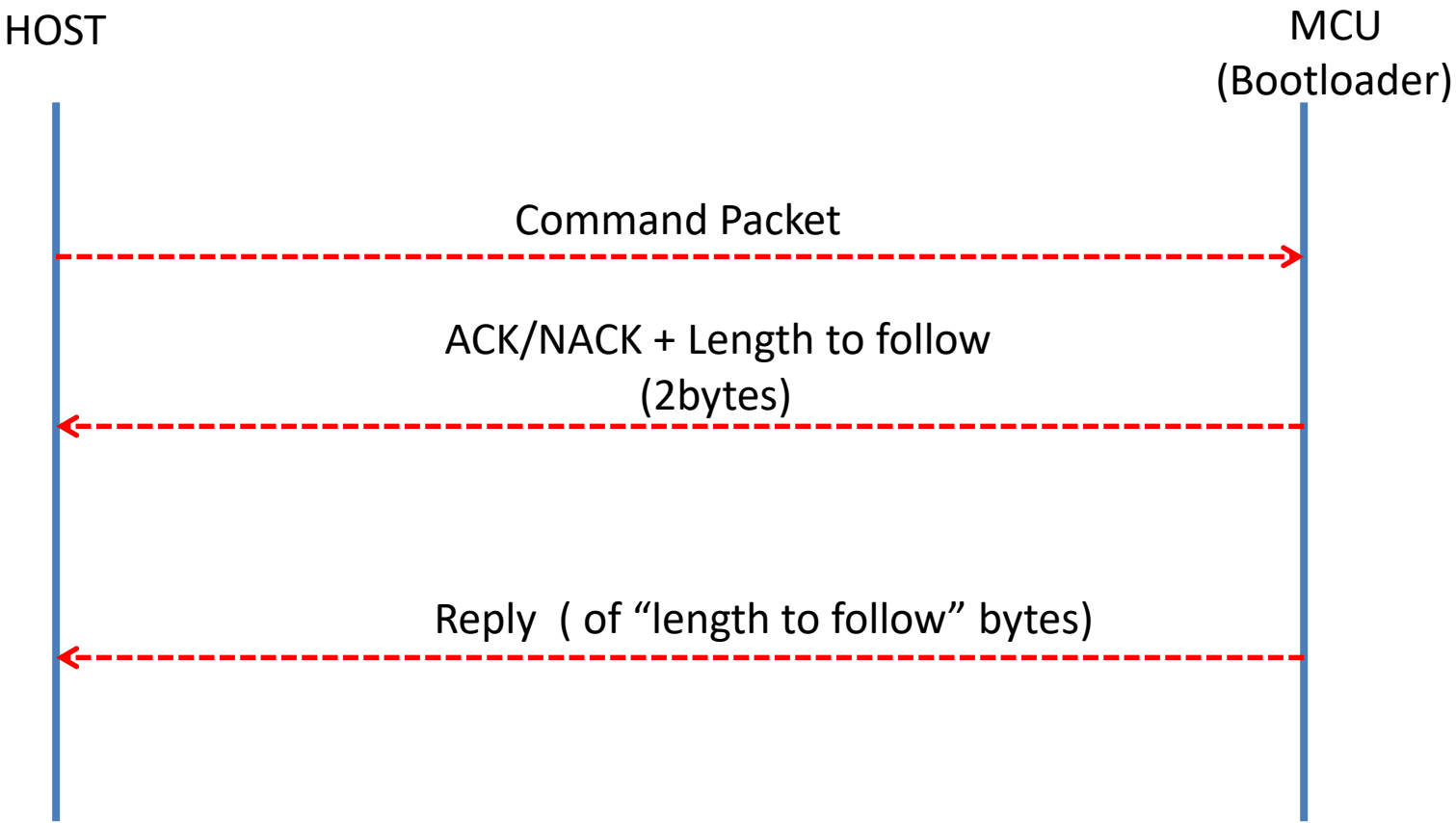
Table 2. Boot modes

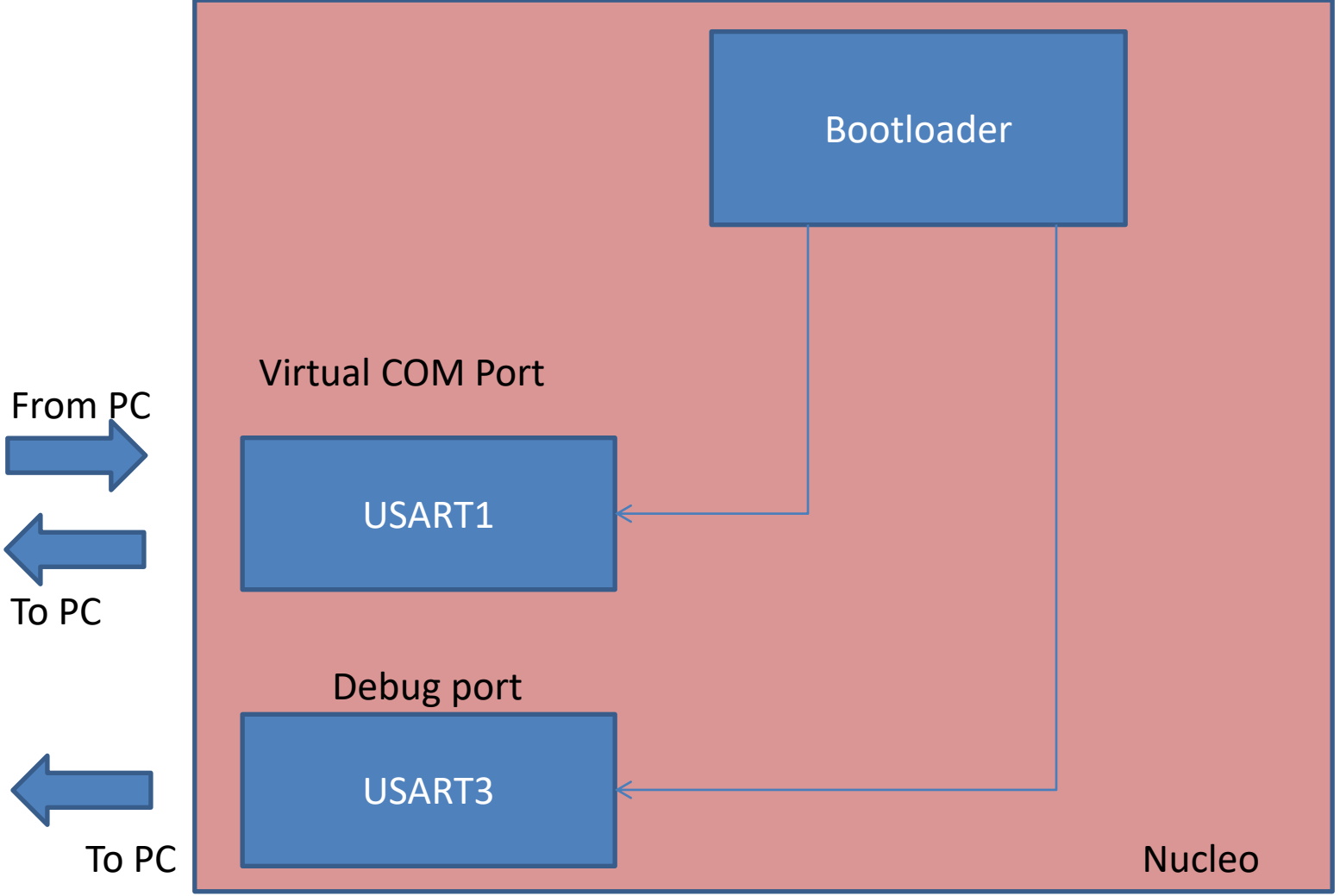
Boot mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
x	0	Main Flash memory	Main Flash memory is selected as the boot area
0	1	System memory	System memory is selected as the boot area
1	1	Embedded SRAM	Embedded SRAM is selected as the boot area

Bootloader : Code Placement in Flash

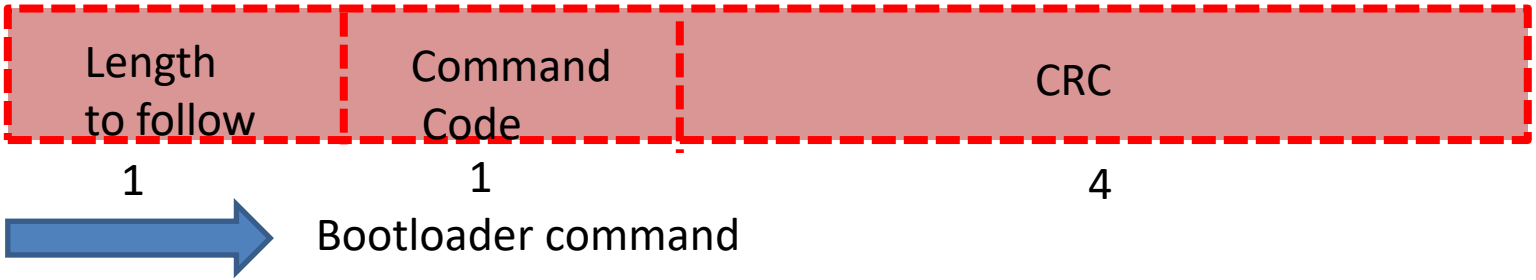
Sector-2 to Sector-7 will be used for storing user application

Host – Bootloader Communication




Bootloader
Virtual COM Port
From PC
USART1
To PC
Debug port
USART3
To PC
Nucleo

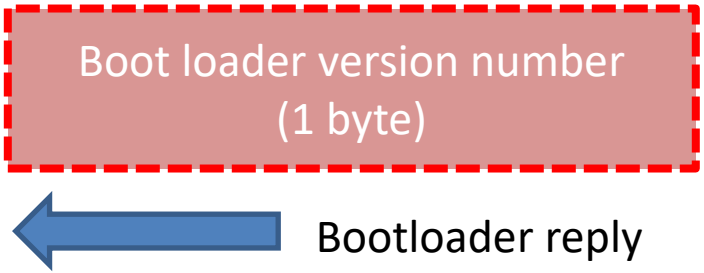
Command Name : BL_GET_VER



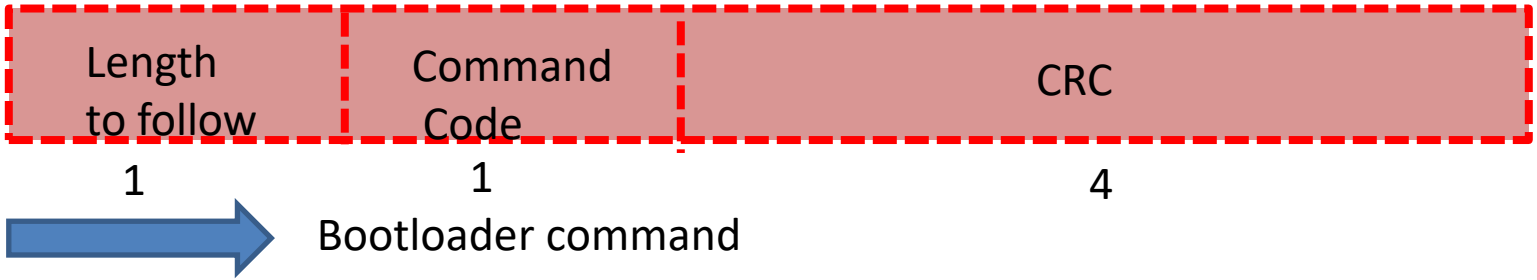
Total Bytes of the packet = 6

“Length to follow ” field will contain the value : 5

Command Code : 0x51



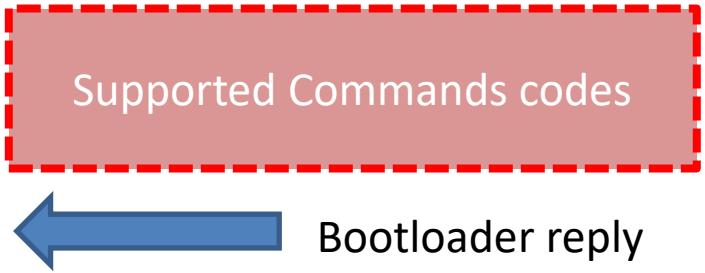
Command Name : BL_GET_HELP



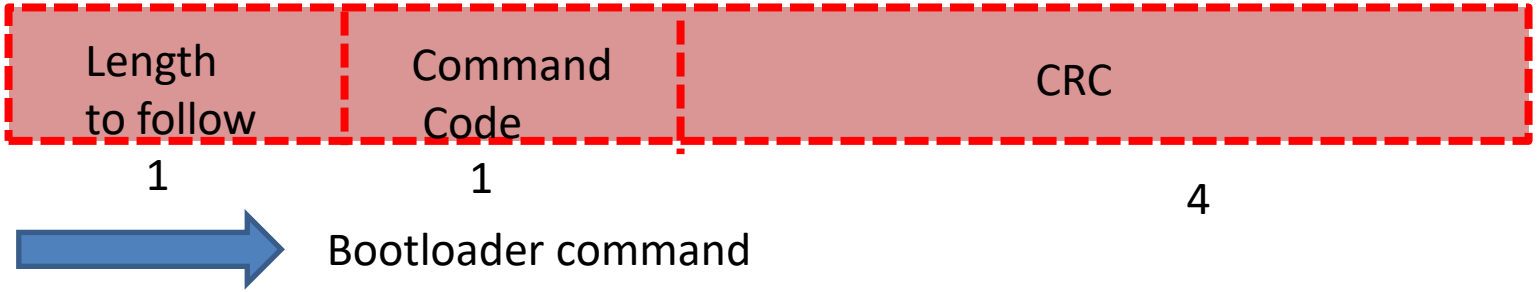
Total Bytes of the packet = 6

“Length to follow ” field will contain the value : 5

Command Code : 0x52



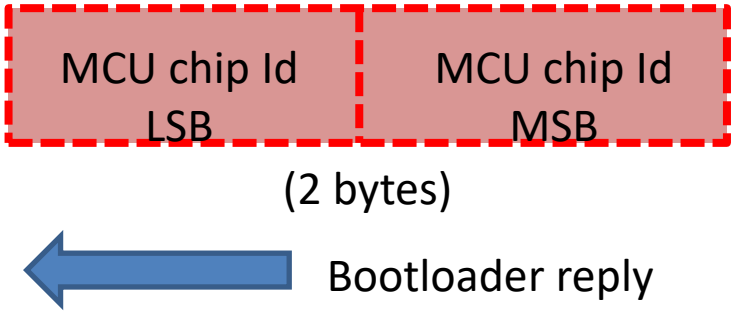
Command Name : BL_GET_CID



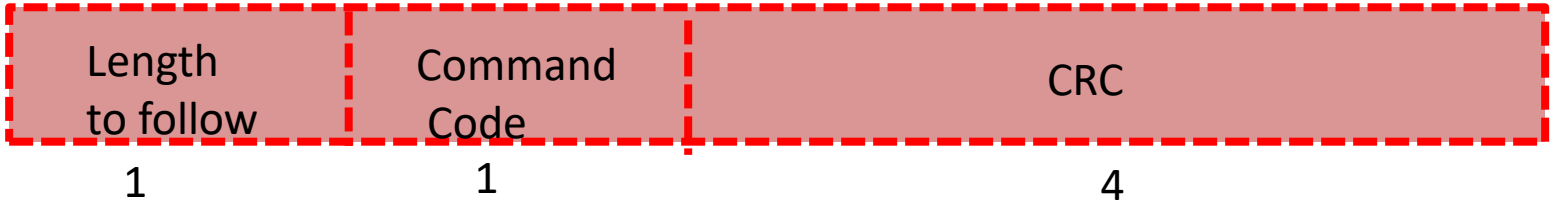
Total Bytes of the packet = 6

“Length to follow ” field will contain the value : 5

Command Code : 0x53



Command Name : BL_GET_RDP_STATUS



Bootloader command

Total Bytes of the packet = 6

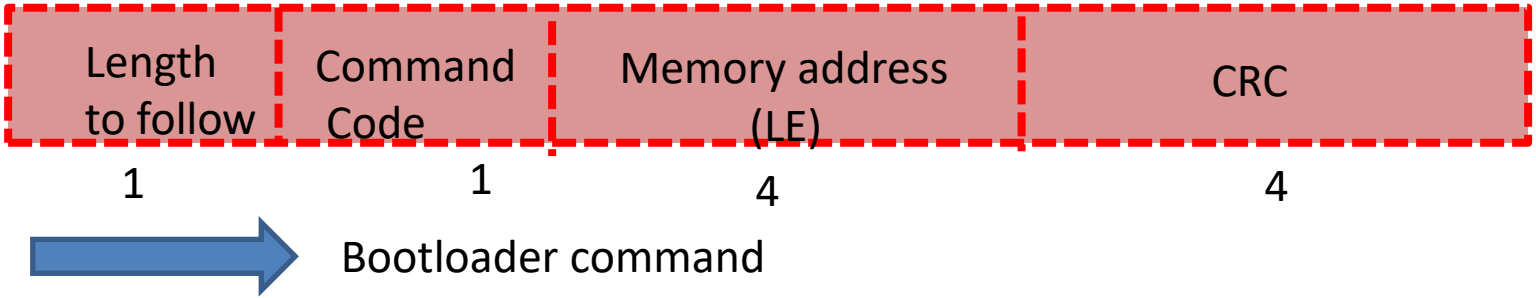
“Length to follow ” field will contain the value : 5

Command Code : 0x54

(1 bytes)

Bootloader reply

Command Name : BL_GO_TO_ADDR



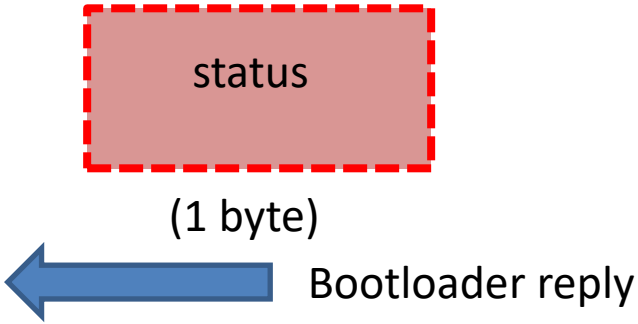
Total Bytes of the packet = 10

“Length to follow ” field will contain the value : 9

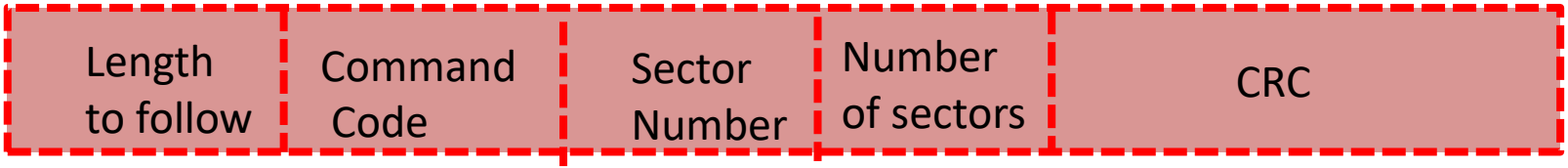
Command Code : 0x55

Base Memory Addr. :

4 Byte base address to jump



Command Name : BL_FLASH_ERASE



Bootloader command

Total Bytes of the packet = 8

“Length to follow ” field will contain the value : 7

Command Code : 0x56

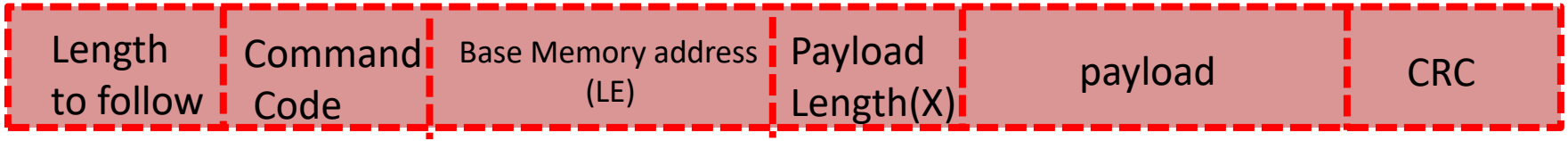
Sector number : 0 , 1 , 2 , 3 , 4 ,5 , 6 ,7

Number of sectors : 0 to 7

(1 byte)

Bootloader reply

Command Name : BL_MEM_WRITE



Bootloader command

Total Bytes of the packet = 11+X

“Length to follow ” field will contain the value : 10+X

Command Code : 0x57

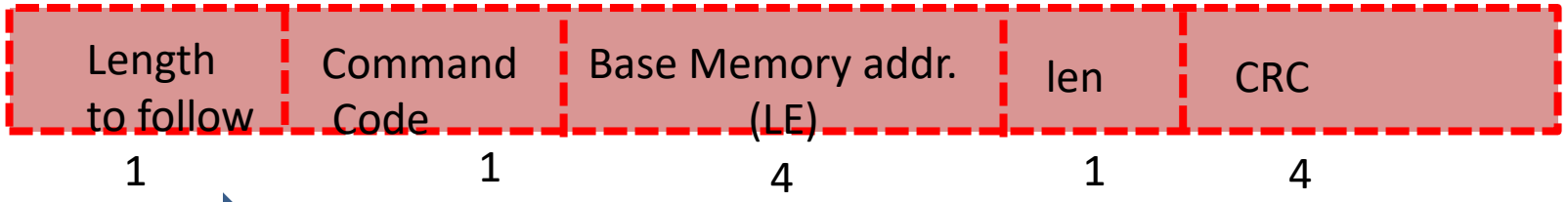
Base Memory Addr. :
4 Byte base address

Payload len: No. of bytes to write

Payload : bytes to write

Bootloader reply

Command Name : BL_MEM_READ



Bootloader command

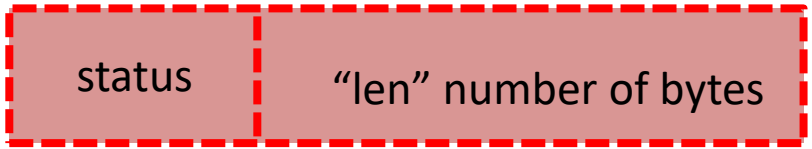
Total Bytes of the packet = 11

“Length to follow ” field will contain the value : 10

Command Code : 0x59

Base Memory Addr. :
4 Byte base address from which data has to be read

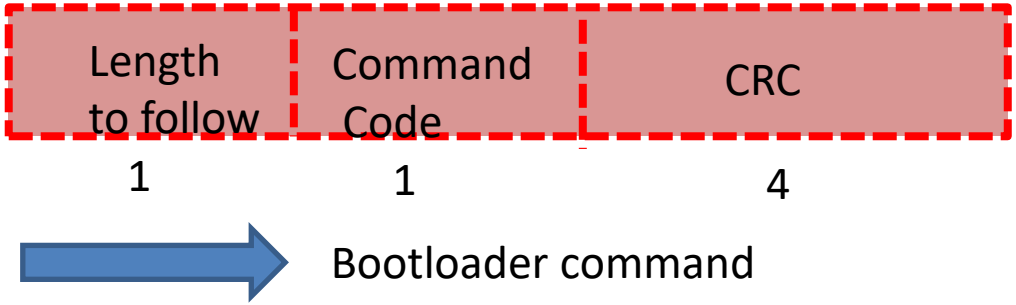
len : No. of bytes to read



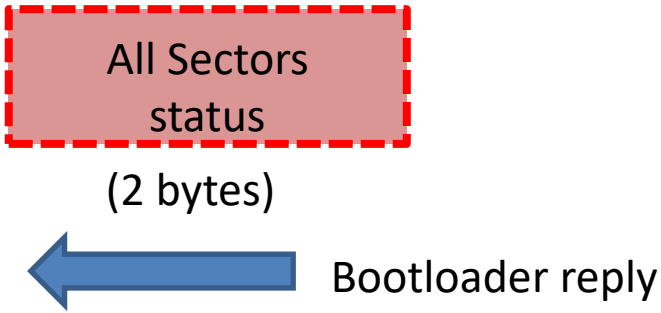
(1+”len” bytes)

Bootloader reply

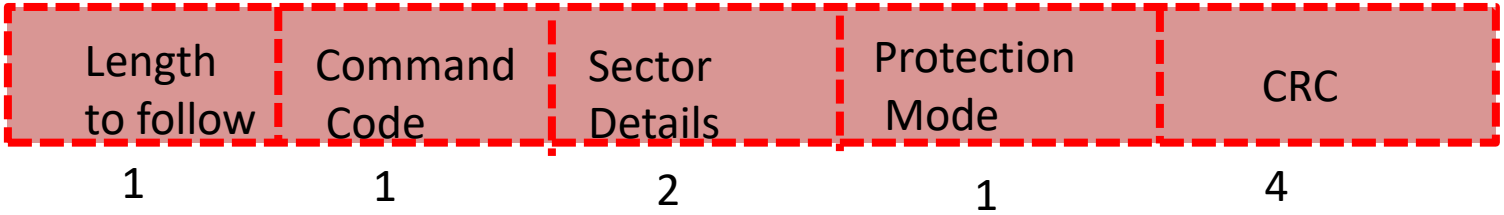
Command Name : BL_READ_SECTOR_STATUS



Total Bytes of the packet = 6
“Length to follow ” field will contain the value : 5
Command Code : 0x5A



Command Name : BL_EN_R_W_PROTECT



Bootloader command

Total Bytes of the packet = 9

"Length to follow " field will contain the value : 8

Command Code : 0x58

Sector details : sector numbers encoded in 8bits (ex: 0^{th} bit is sector 0)

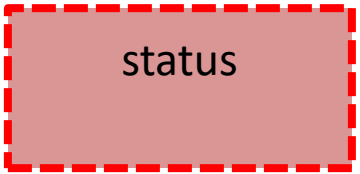
1: protection

0: No protection

Protection Mode :

1(Write Protection)

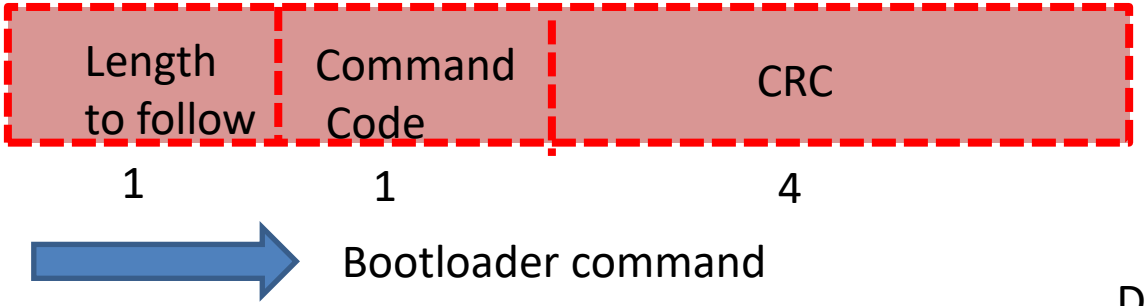
2 (R/W protection)



(1 bytes)

Bootloader reply

Command Name : BL_DIS_R_W_PROTECT

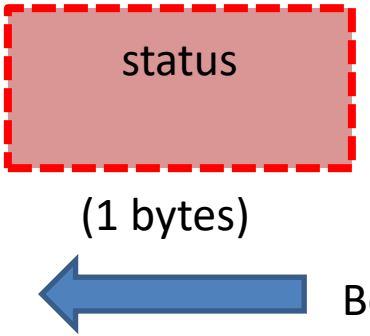


Disables Active protection on all the sectors (resumes to default state)

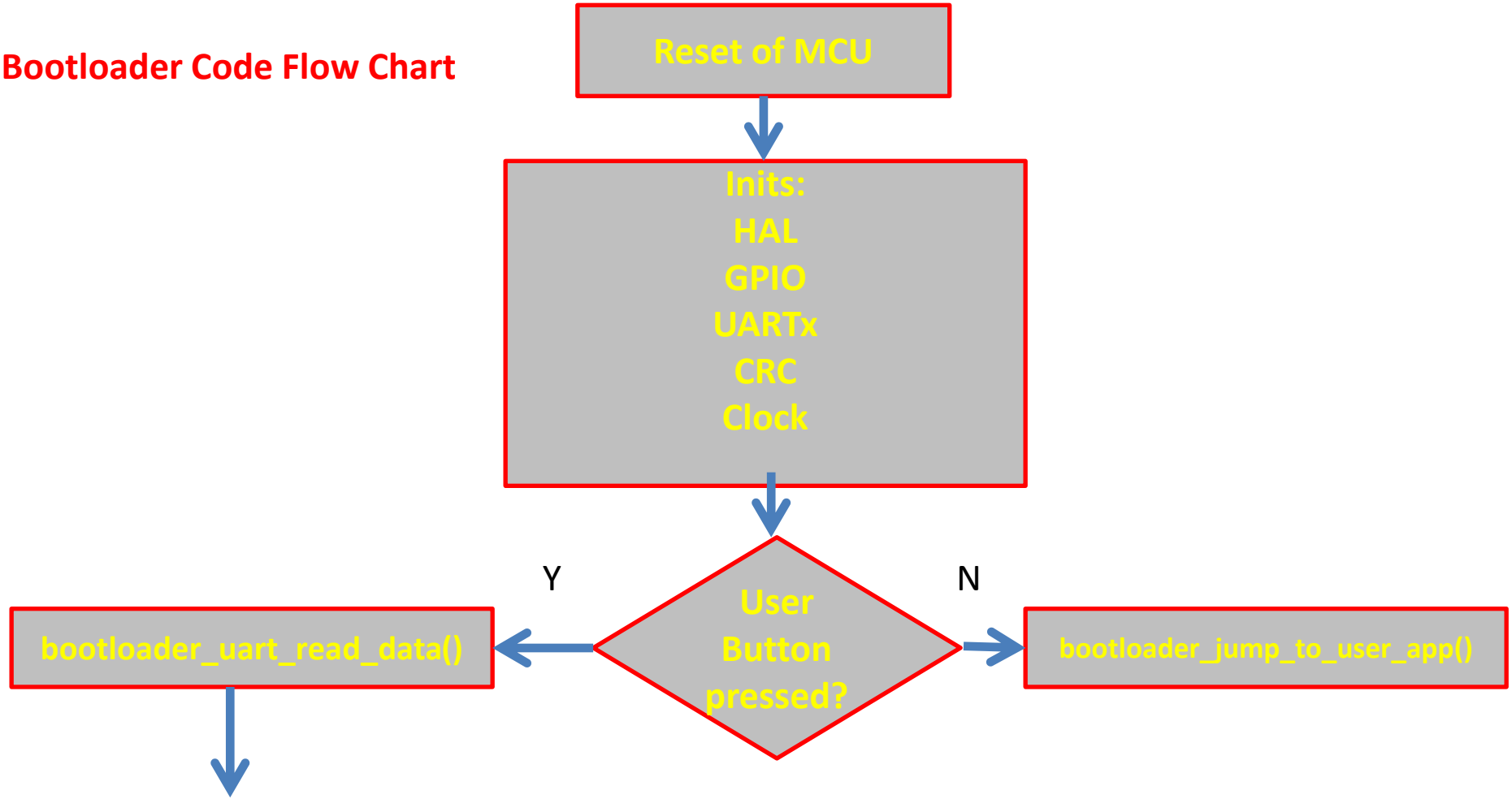
Total Bytes of the packet = 6

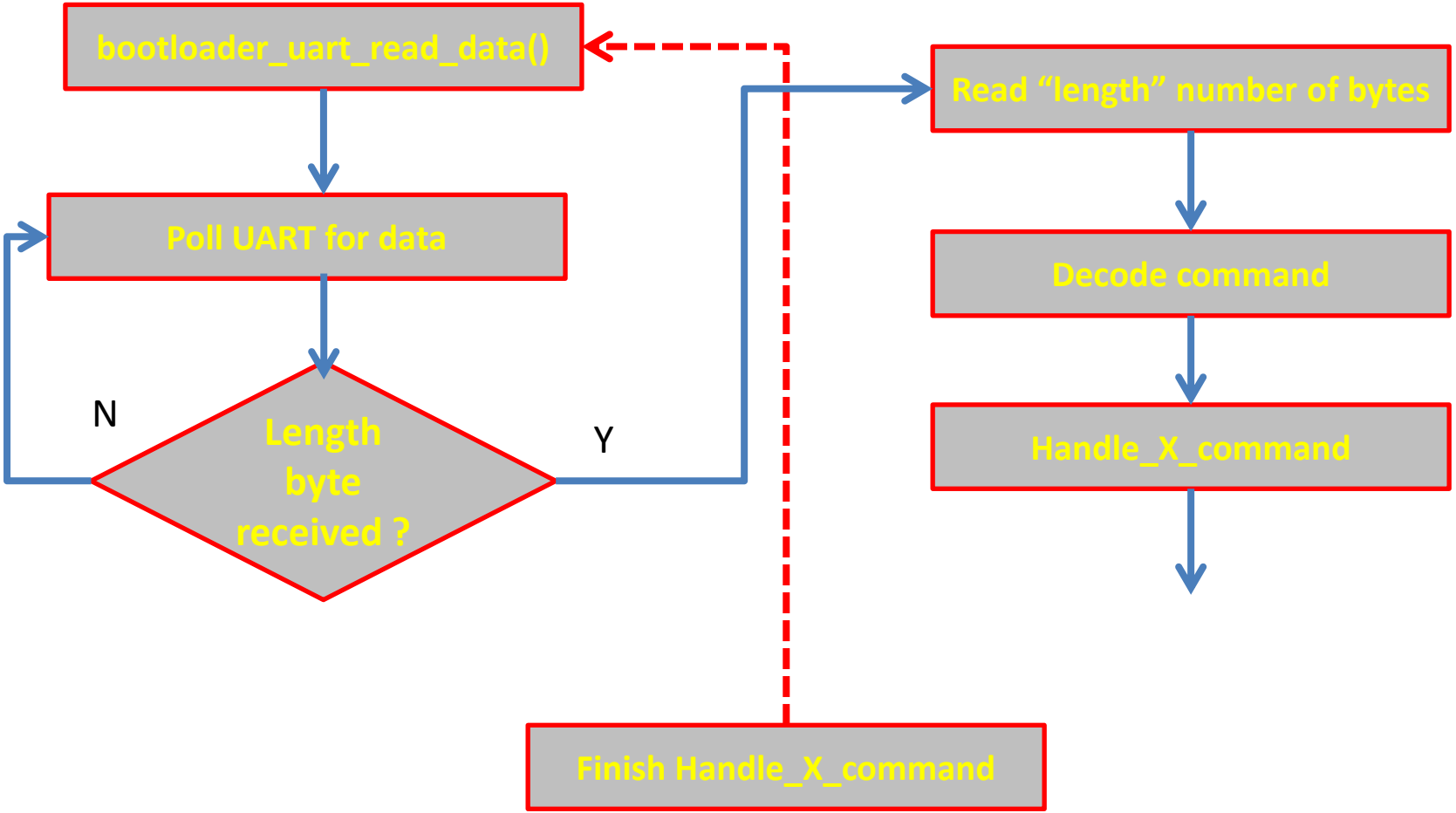
“Length to follow ” field will contain the value : 5

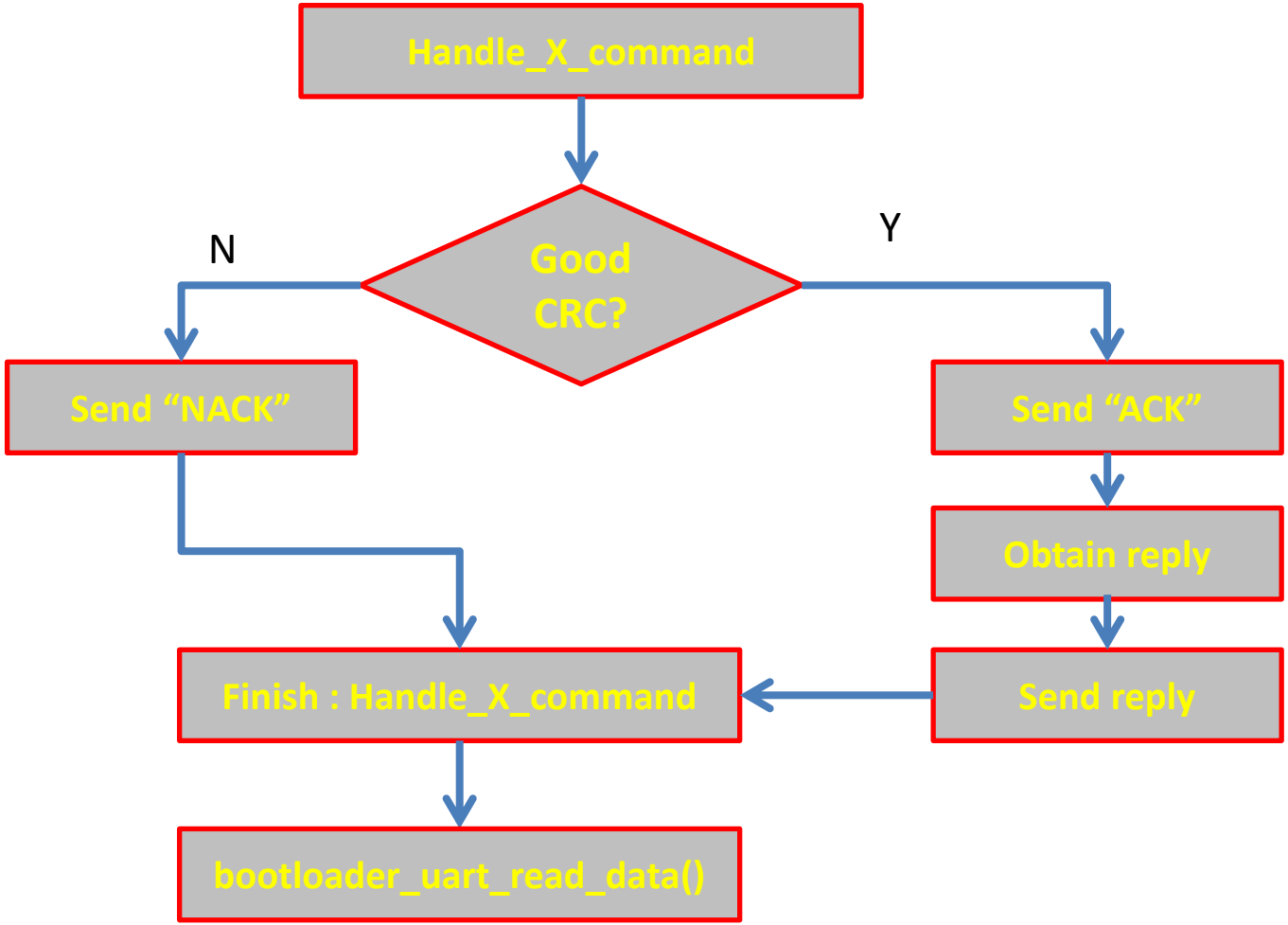
Command Code : 0x5C



Bootloader Code Flow Chart




bootloader_uart_read_data()
Poll UART for data
Length byte received ?
N
Y
Read “length” number of bytes
Decode command
Handle_X_command
Finish Handle_X_command


Handle_X_command
Good CRC?
N
Y
Send "NACK"
Send "ACK"
Obtain reply
Send reply
Finish : Handle_X_command
bootloader_uart_read_data()