

# 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

ARDUINO UNO REV3



Does this MCU come with on chip Bootloader ?

Yes it is !

Does it run whenever MCU undergoes reset ?

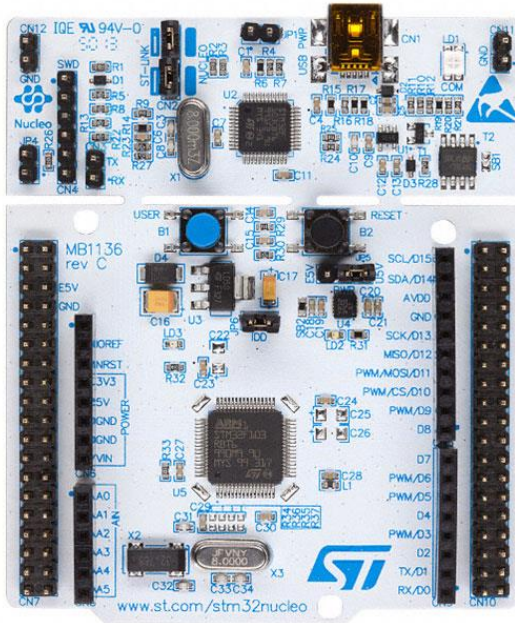
Yes. Upon reset, Arduino bootloader runs first

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 ?

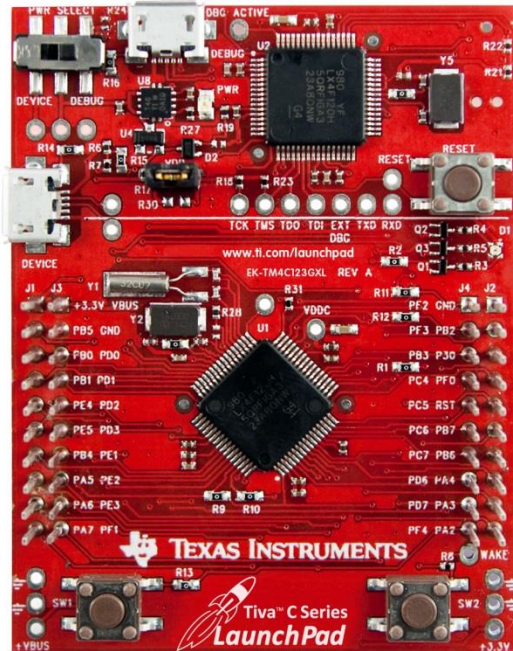
**No. Should activate changing the status of boot pins**

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  $2 \times 16$  bytes.
- Backup RAM of 4KB

# Internal Flash memory

- Size is 2MB
- 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

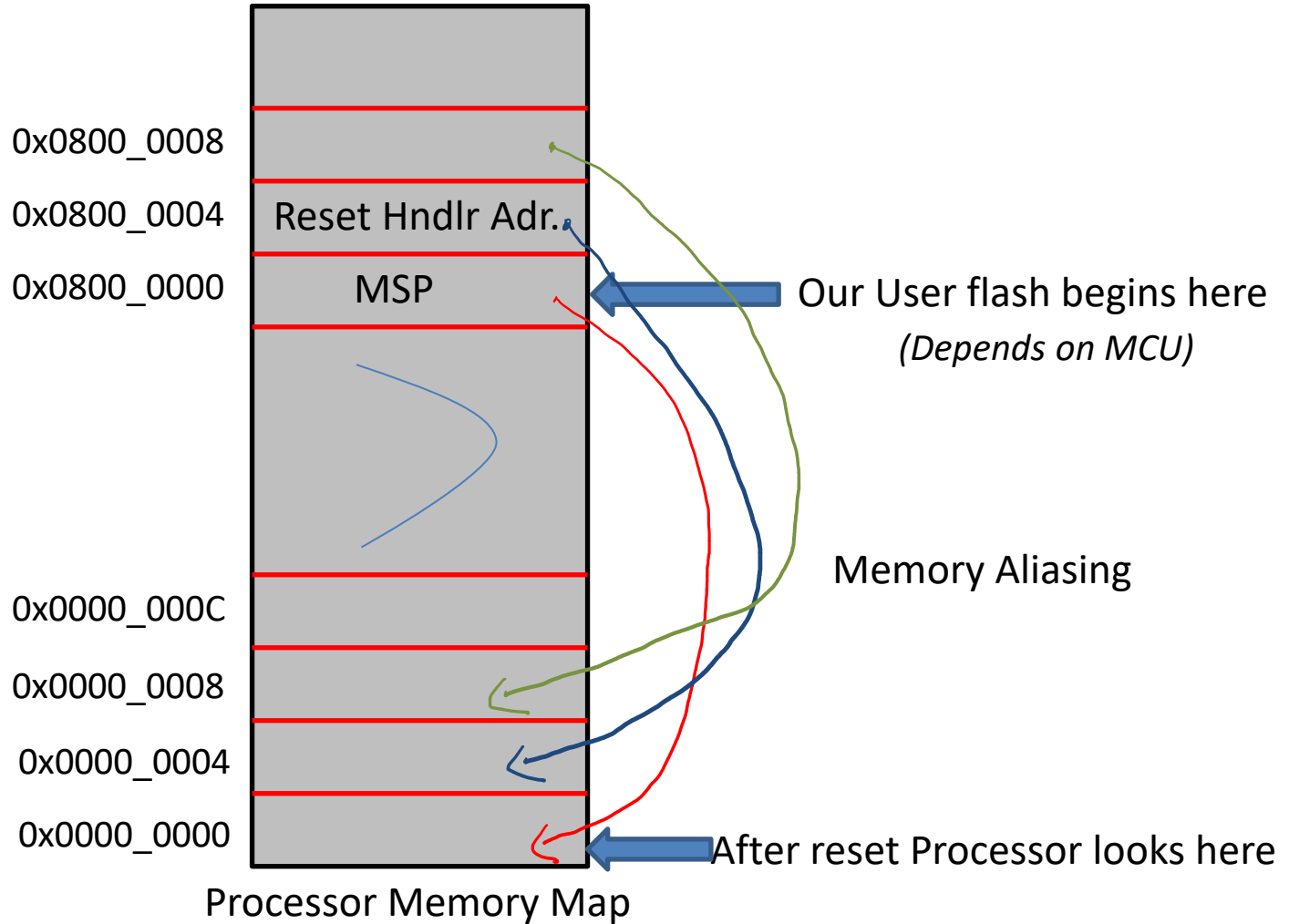
**All ARM Cortex M Based MCUs right after reset does,**

- 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 ??**



# System Memory ( ROM)

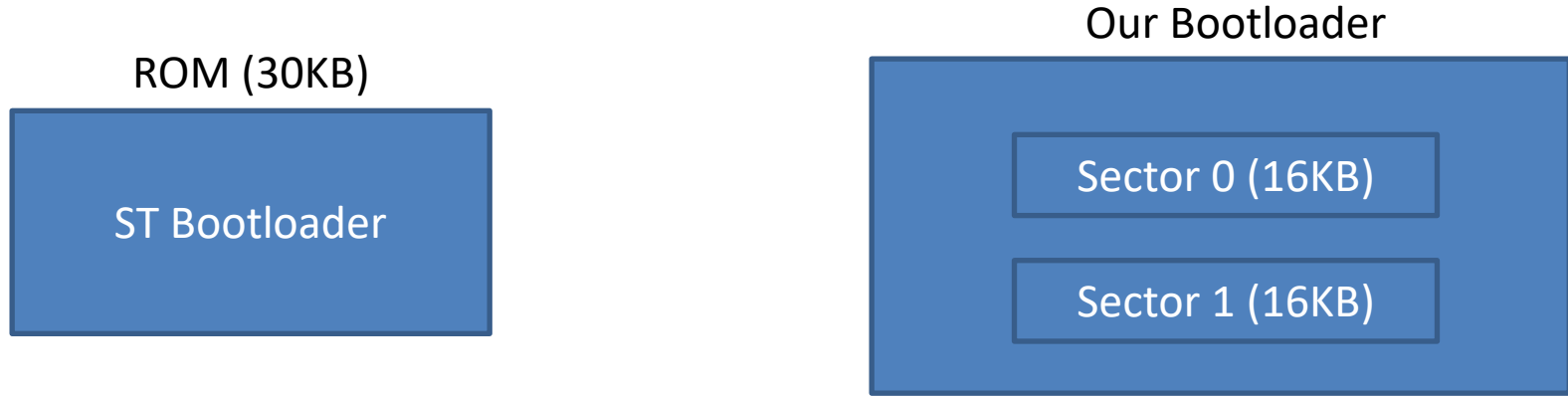
- Size is 30KB
- Begins @ 0x1FFF\_0000
- Ends @ 0x1FFF\_77FF
- All the ST MCUs store Bootloader in this memory
- This Memory is Read only
- By default MCU will not execute any code from this memory but you can configure MCU to boot or execute bootloader from this memory .

# 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

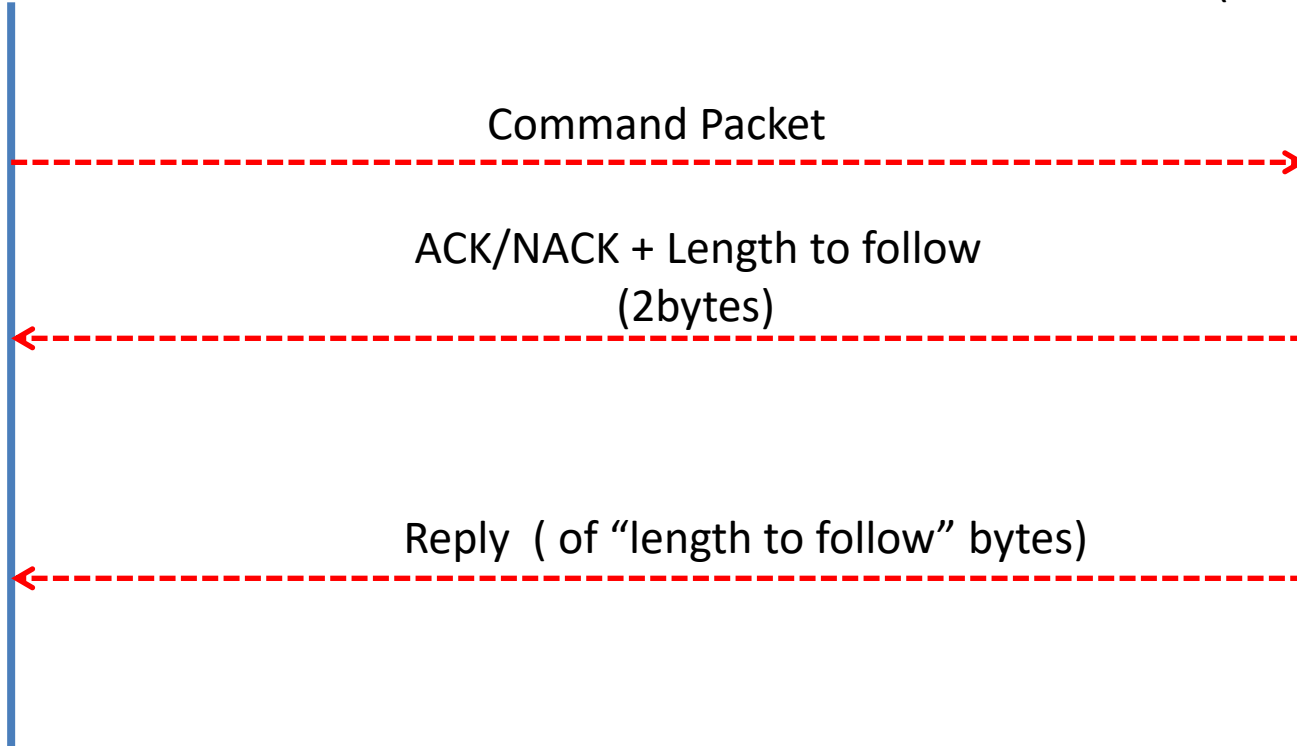


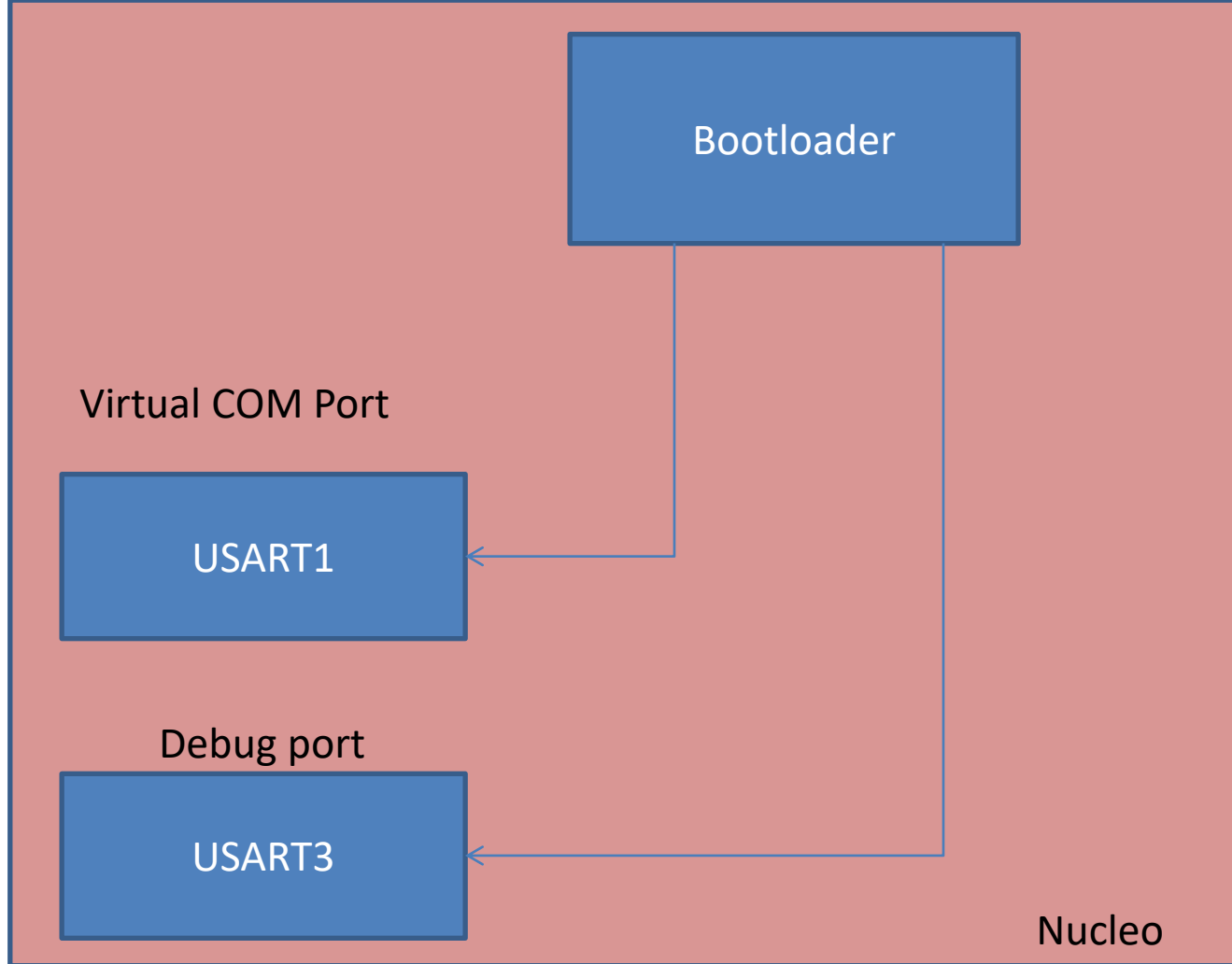
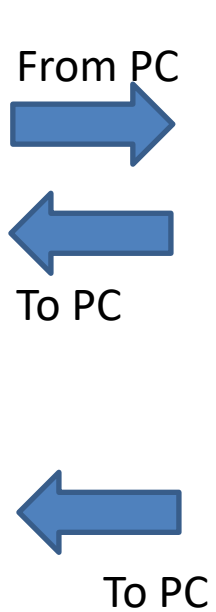
***Sector-2 to Sector-7 will be used  
for storing user application***

# Host – Bootloader Communication

HOST

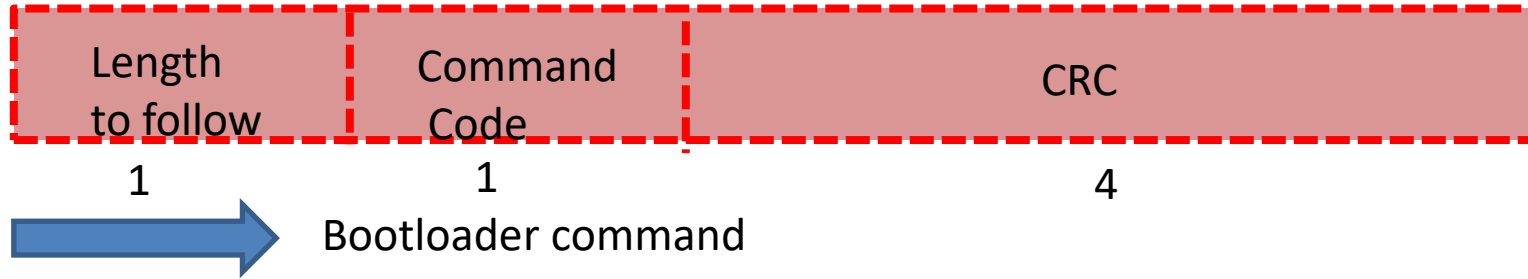
MCU  
(Bootloader)







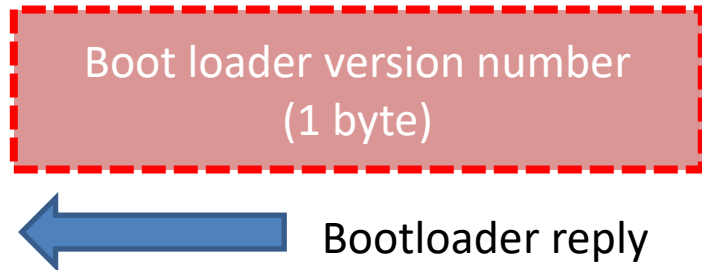
## Command Name : BL\_GET\_VER



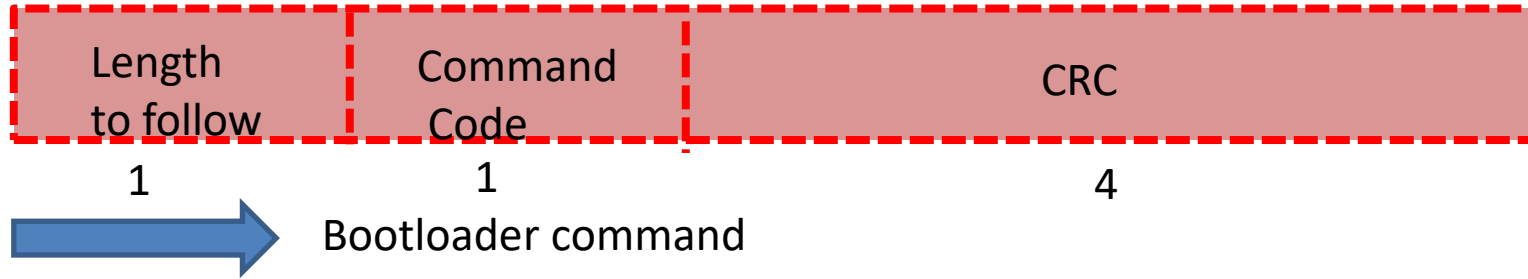
Total Bytes of the packet = 6

"Length to follow" field will contain the value : 5

**Command Code : 0x51**



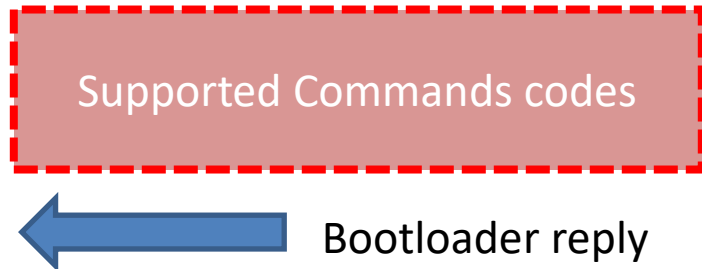
## Command Name : BL\_GET\_HELP



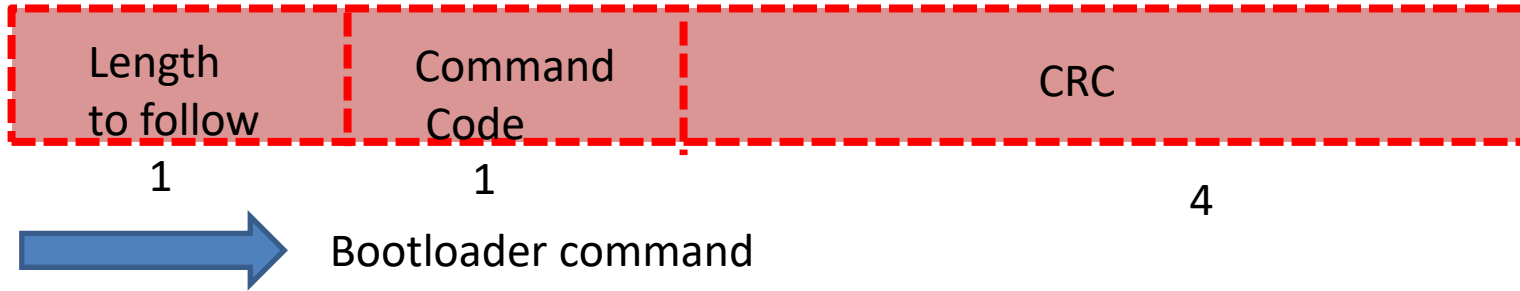
Total Bytes of the packet = 6

"Length to follow " field will contain the value : 5

**Command Code : 0x52**



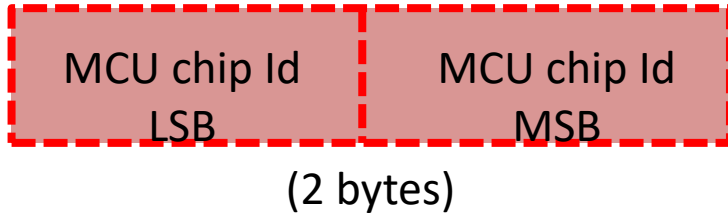
## Command Name : BL\_GET\_CID



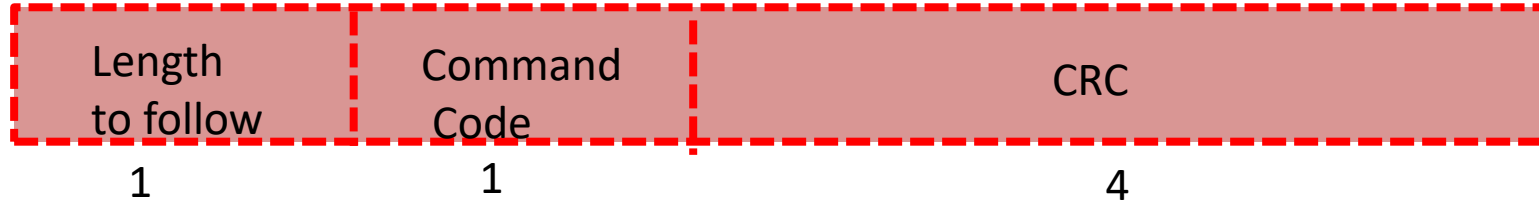
Total Bytes of the packet = 6


"Length to follow " field will contain the value : 5

**Command Code : 0x53**



## Command Name : BL\_GET\_RDP\_STATUS



 Bootloader command

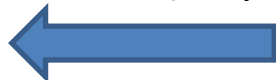
Total Bytes of the packet = 6

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

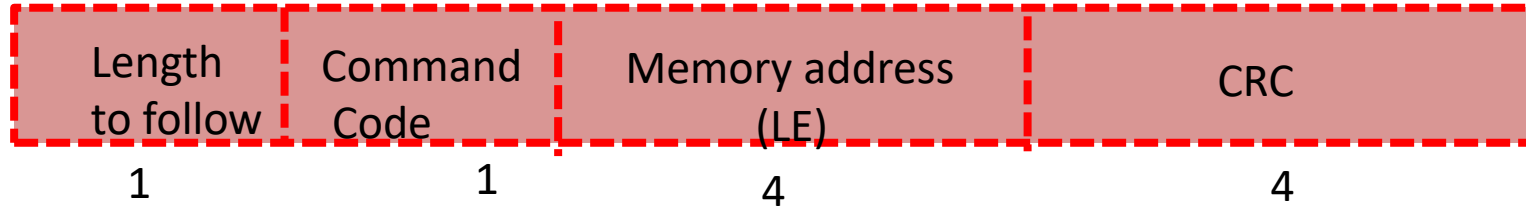
**Command Code : 0x54**



(1 bytes)

 Bootloader reply

## Command Name : BL\_GO\_TO\_ADDR



Bootloader command

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

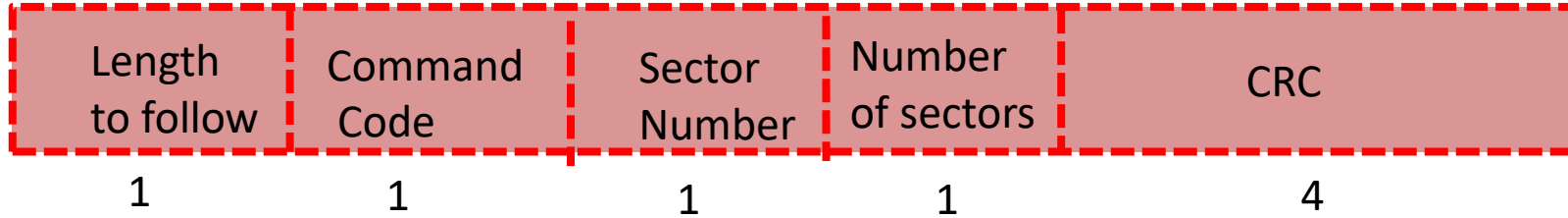


(1 byte)



Bootloader reply

## 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**

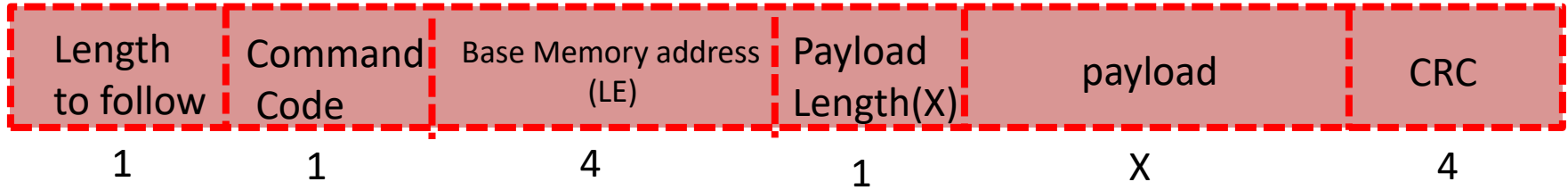


(1 byte)



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

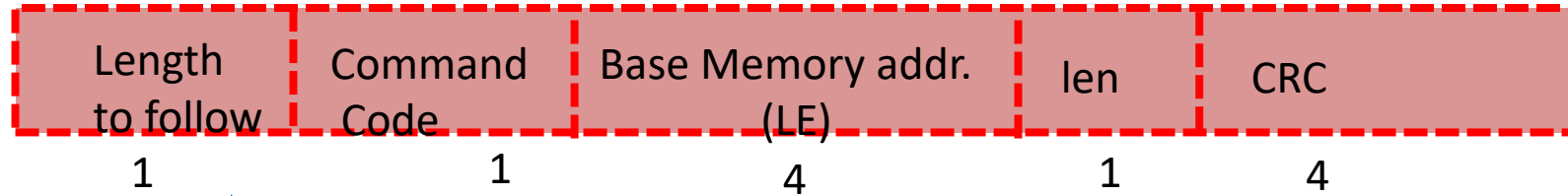



(1 byte)



Bootloader reply

## Command Name : BL\_MEM\_READ



 Bootloader command

Total Bytes of the packet = 11

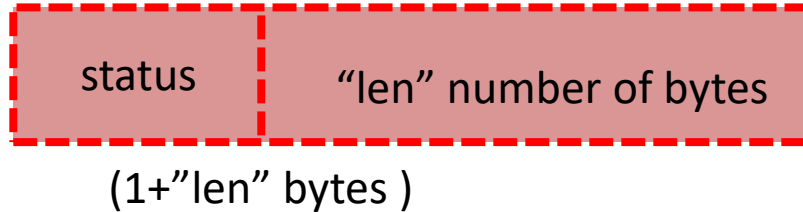
**"Length to follow"** field will contain the value : 10

**Command Code** : 0x59

**Base Memory Addr. :**

4 Byte base address from which data has to be read

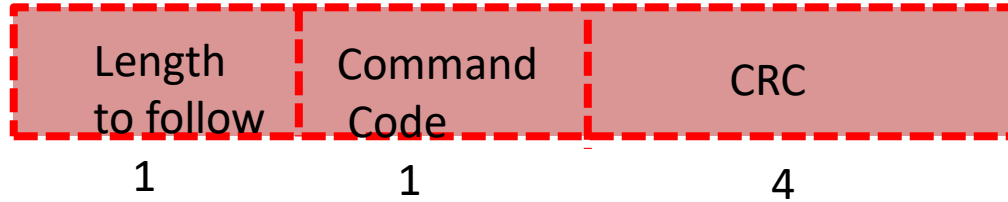
**len** : No. of bytes to read



 Bootloader reply



## Command Name : BL\_READ\_SECTOR\_STATUS



Bootloader command

Total Bytes of the packet = 6

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

**Command Code** : 0x5A

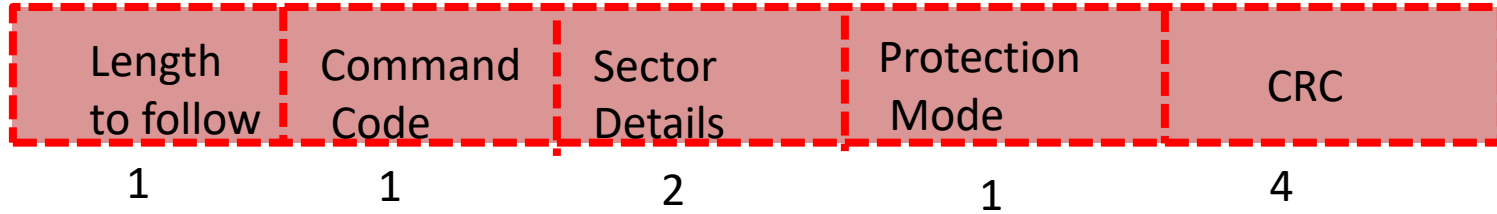


(2 bytes)



Bootloader reply

# 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<sup>th</sup> bit is sector 0)

1: protection

0: No protection

**Protection Mode :**

1(Write Protection )

2 (R/W protection)

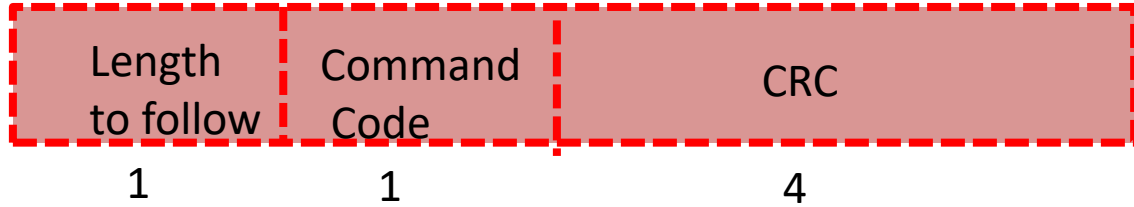


(1 bytes)



Bootloader reply

# Command Name : BL\_DIS\_R\_W\_PROTECT



Bootloader command

Total Bytes of the packet = 6

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

**Command Code : 0x5C**

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



(1 bytes)



Bootloader reply

Bootloader Code Flow Chart

