Bootloader for STM with CRC Integration

- **Author: Alfred Nagy Alfred**
- **Date: [01/03/2025]**
- **Version: [2]**

Contents

Abstract

This project presents a UDS-like protocol bootloader for the STM32F446RE microcontroller, developed using STM32CubeIDE HAL drivers. The bootloader implements 11 services inspired by the Universal Diagnostic Services (UDS) protocol, enabling secure and efficient firmware management. These services cover key functionalities such as firmware upload, download, memory operations, and system control.

To ensure data integrity during firmware updates, a Cyclic Redundancy Check (CRC) mechanism is employed, providing robust error detection, and enhancing reliability. The design is modular and scalable, making it adaptable to other STM32 devices and communication interfaces.

Development was carried out in C, using the Nucleo-32F446RE board as the target platform, with a focus on reliability, ease of integration, and adherence to industry best practices.

Key Features

1. **UDS-like Command Set:**
    - A minimalistic yet functional set of commands for managing the bootloading process.
2. **CRC Integrity Check:**
    - Ensures accurate and reliable firmware updates.
3. **Scalability**:
    - Designed with modularity, allowing easy adaptation for other STM32 families.
4. **Ease of Use:**
    - Compatible with standard communication protocols for straightforward integration.

System Overview

Workflow

The system workflow involves securing the firmware update process as follows:

1. A firmware update is hashed using **CRC** before transmission to the STM device.
2. Upon receiving the firmware update, the **System** calls cryptographic functions implemented in the **security module** to:
    - Verify its integrity by comparing the received hash with the calculated hash.
3. If the firmware passes verification, the system calls the **bootloader to check the command packet to:**
    - Check the asked service then call the proper service’s handler.

Current Implementation

- The **system** is responsible for calling cryptographic functions implemented in the **security module** integrity verification.
- This integration ensures an efficient communication mechanism while maintaining a modular design.

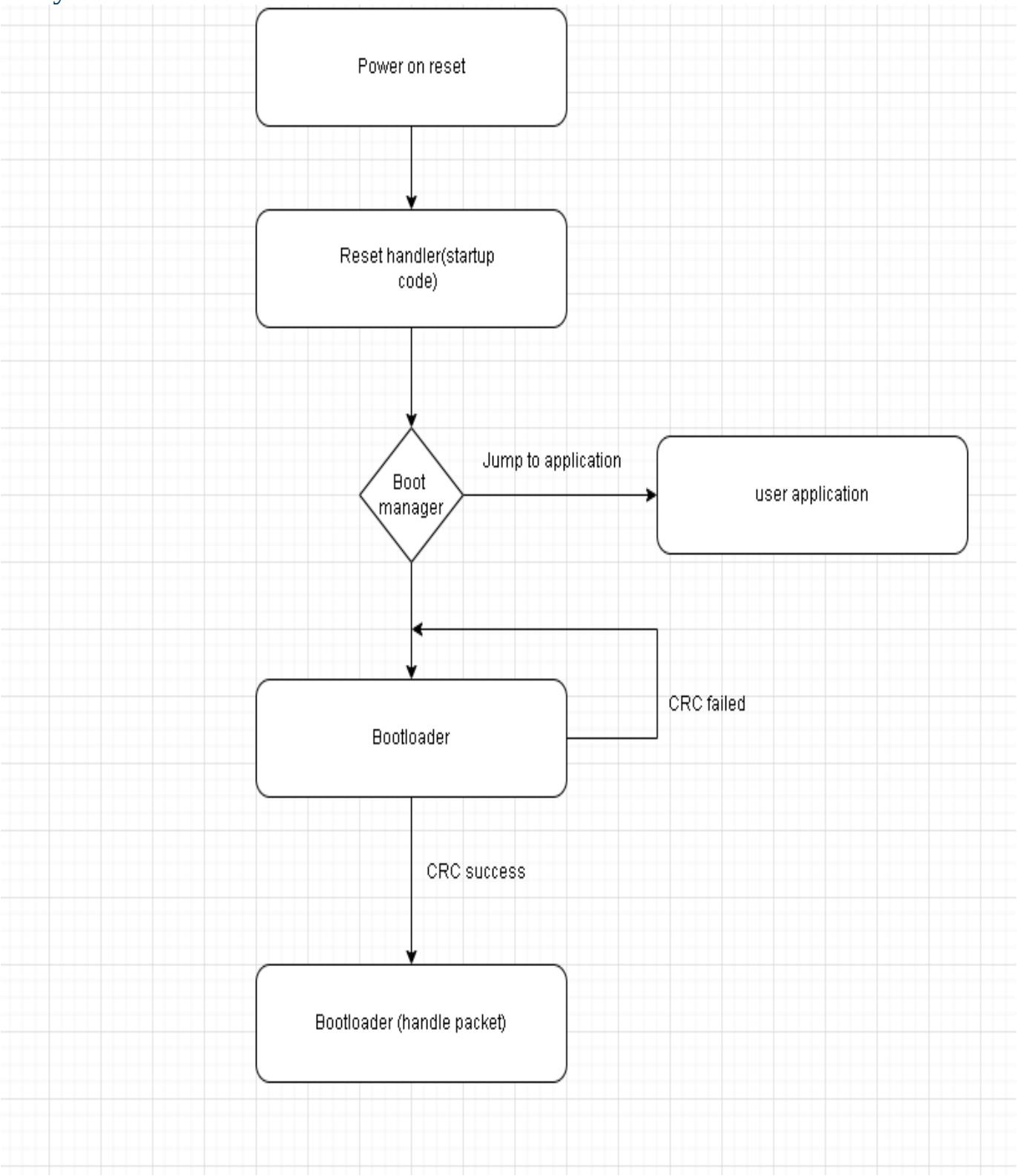
Bootloader

- Communicates with the host through **UART**.
- Interpret the command packet and provide proper handling to the supported services.

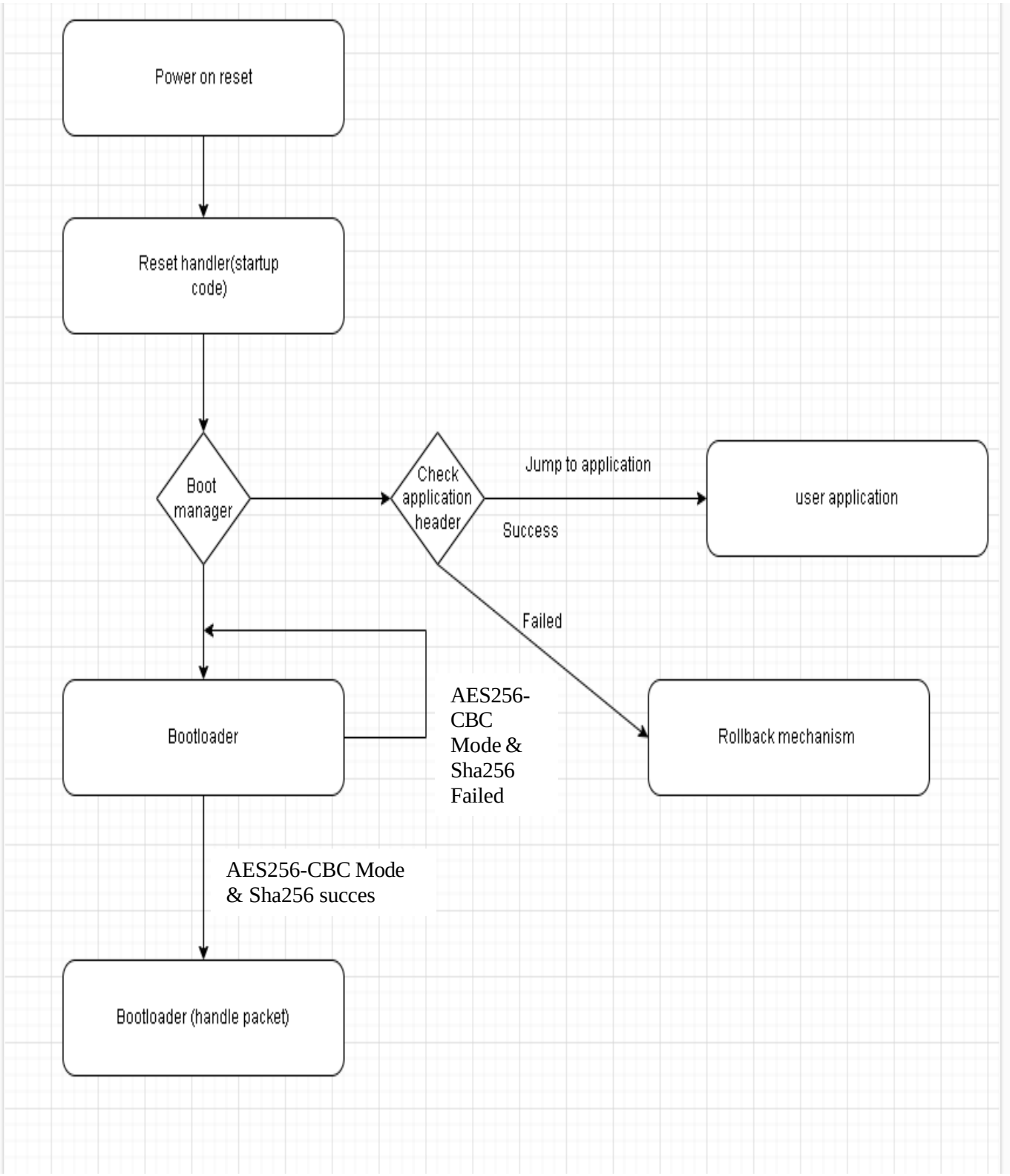
Future vision

1. **Authentication check:**
    - Adding encryption layer using AES-256 CBC mode to ensure that the authenticity of users.
2. **Leverage integrity check and use sha256.**
3. **System improvements:**
    - A header will be added before every image consists of:
        i. Start address.
        ii. Image length.
        iii. Hash such as SHA256.
        iv. Flag to specify the required operation on the image(update-remove-upload).
    - A shared **flash sector** containing the common function between **Application** and **Bootloader** to facilitate the calling and remove duplications.
    - Using the header, the bootloader will check for the application images before jumping into it.

System Architecture:

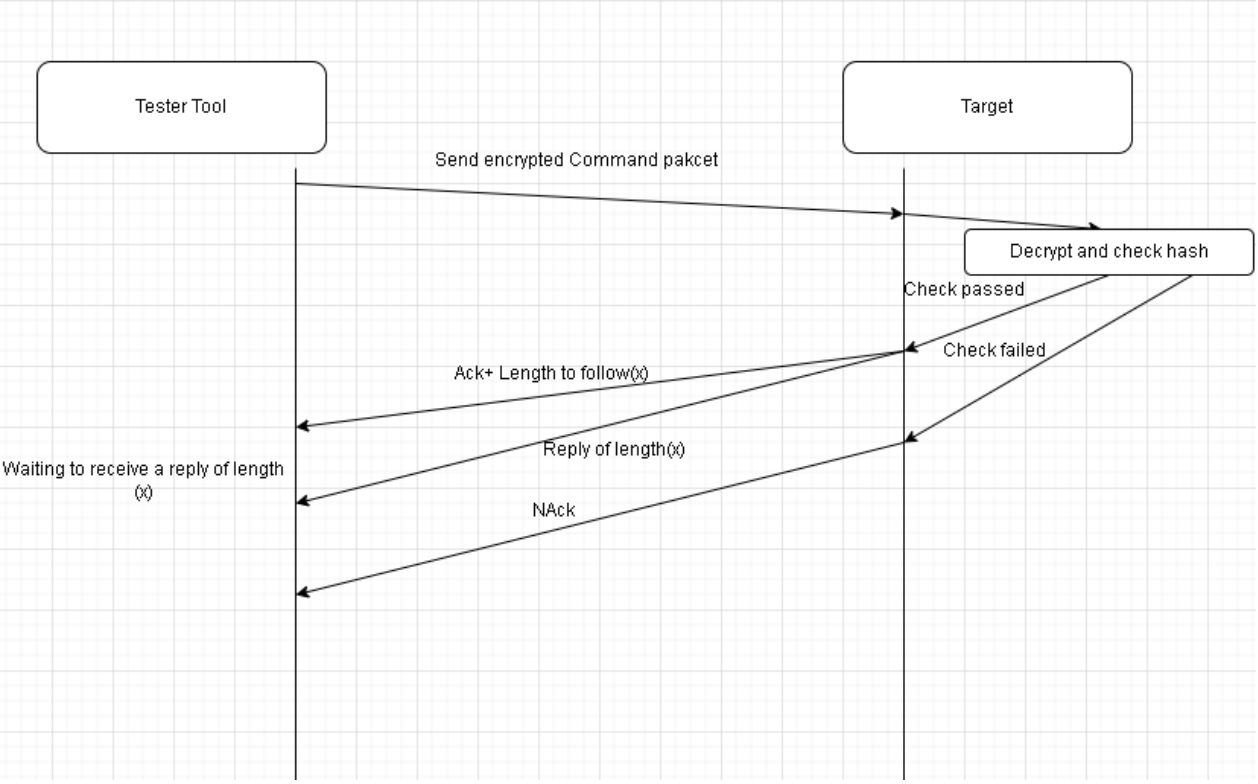


Future workflow:



Bootloader

Host-Target communication



Flash_HAL_OK	= 0x00
Flash_HAL_ERROR	= 0x01
Flash_HAL_BUSY	= 0x02
Flash_HAL_TIMEOUT	= 0x03
Flash_HAL_INV_ADDR	= 0x04

Note: The length to follow is sent unencrypted.

RDP protection levels – OPTION BYTES

The user area in the Flash memory can be protected against read operations by an entrusted code. Three read protection levels are defined:

- **Level 0: no read protection:**

  When the read protection level is set to Level 0 by writing 0xAA into the read protection option byte (RDP), all read/write operations (if no write protection is set) from/to the Flash memory are possible in all boot configurations (Flash user boot, debug or boot from RAM).

- Level 1: read protection enabled:
  It is the default read protection level after option byte erase. The read protection Level 1 is activated by writing any value(except for 0xAA and 0xCC used to set Level 0 and Level 2, respectively) into the RDP option byte. When the read protection Level 1 is set:
  - No access (read, erase, program) to Flash memory can be performed while the debug feature is connected or while booting from RAM or system memory bootloader. A bus error is generated in case of read request.
  - When booting from Flash memory, accesses (read, erase, program) to Flash memory from user code are allowed.

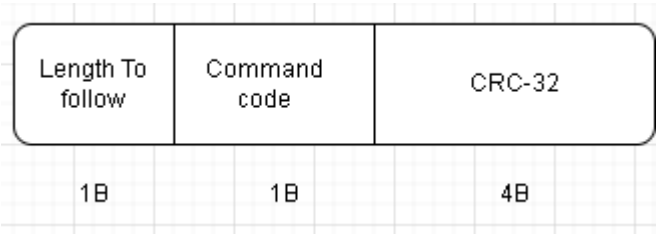
  When Level 1 is active, programming the protection option byte (RDP) to Level 0 causes the Flash memory to be mass-erased. As a result the user code area is cleared before the read protection is removed. The mass erase only erases the user code area. The other option bytes including write protections remain unchanged from before the mass-erase operation. The OTP area is not affected by mass erase and remains unchanged. Mass erase is performed only when Level 1 is active and Level 0 requested. When the protection level is increased (0->1, 1->2, 0->2) there is no mass erase.

- Level 2: debug/chip read protection disabled:
  The read protection Level 2 is activated by writing 0xCC to the RDP option byte. When the read protection Level 2 is set:
  - All protections provided by Level 1 are active.
  - Booting from RAM or system memory bootloader is no more allowed.
  - JTAG, SWV (single-wire viewer), ETM, and boundary scan are disabled.
  - User option bytes can no longer be changed. When booting from Flash memory, accesses (read, erase and program) to Flash memory from user code are allowed.

  Memory read protection Level 2 is an irreversible operation. When Level 2 is activated, the level of protection cannot be decreased to Level 0 or Level 1.

Command Packets(Services):

1- Service name: Bootloader get version:

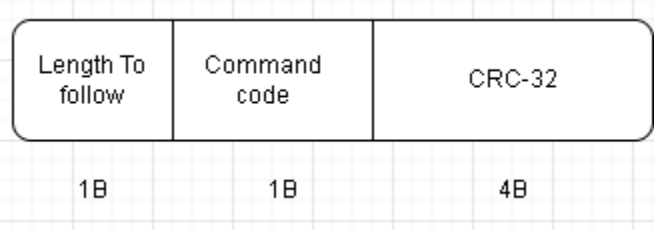


- Packet size = 6 Bytes.
- Length to follow= 5 Bytes.
- Command Code = 0x51.

Request: This service requests the bootloader`s version.

Reply: Bootloader version.

2- Service name: Bootloader get help:

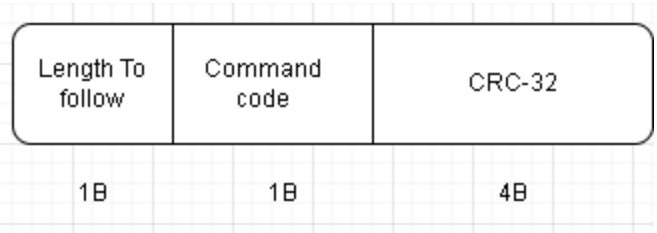


- Packet size = 6 Bytes.
- Length to follow= 5Bytes.
- Command Code = 0x52.

Request: This service asks the bootloader about it's supported services.

Reply: all supported bootloader`s services. Then, the tester tool represents them in a user- friendly interface.

3- Service name: Bootloader get chip ID:

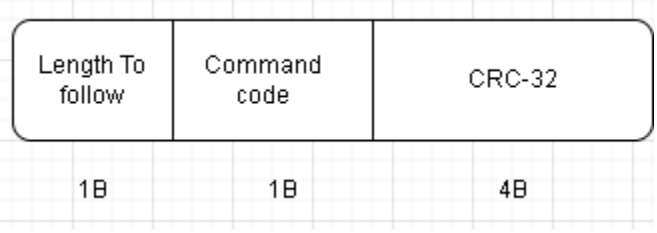


- Packet size= 6 Bytes.
- Length to follow= 5 Bytes.
- Command Code = 0x53.

Request: This service requests the ecu CHIP ID.

Reply: ECU CHIP ID

4- Service name: Bootloader get read protection (RDP-options bytes):

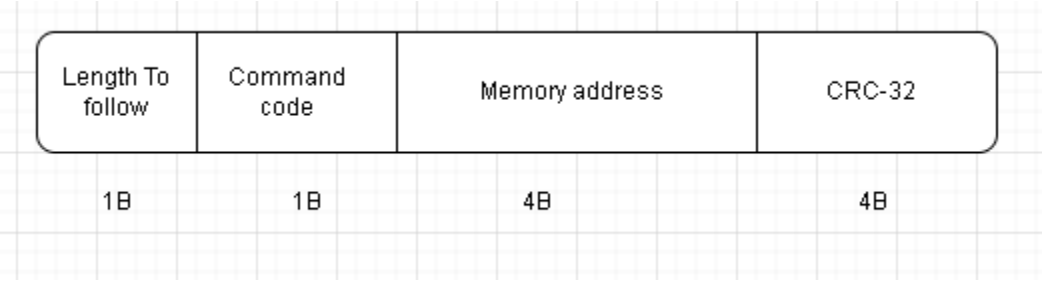


- Packet size= 6 Bytes.
- Length to follow= 5 Bytes.
- Command Code = 0x54.

Request: This service retrieves the RDP from option bytes.

Reply: Sends the activated RDP level.

5- Service name: Bootloader Go to address:



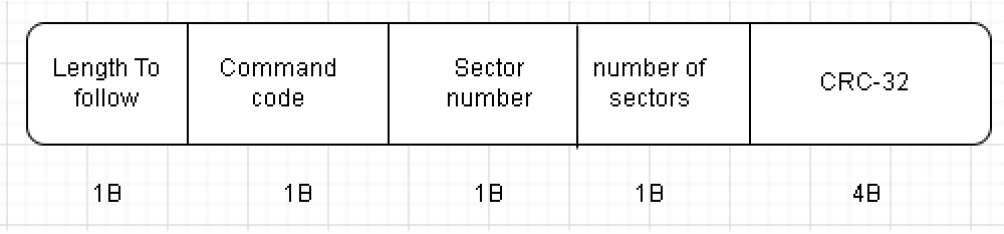
- Packet size =10 Bytes.
- Length to follow= 9 Bytes.
- Command Code = 0x55.

Request: This service sends a memory location to the bootloader to execute from this location.
Reply: Address in/valid.

Future development:

The bootloader checks the memory location content firstly.

6- Service name : Booloader flash erase:

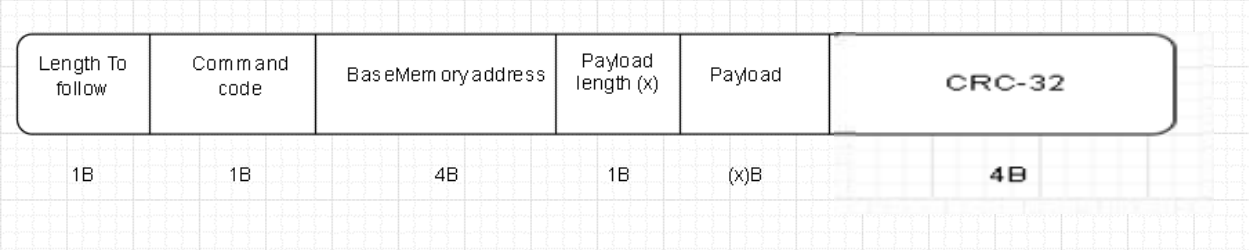


- Packet size= 8 Bytes.
- Length to follow= 7 Bytes.
- Command Code = 0x56.

Request: This erase number of sectors in flash

Reply: operation status.

7- Service name: Bootloader Memory write:

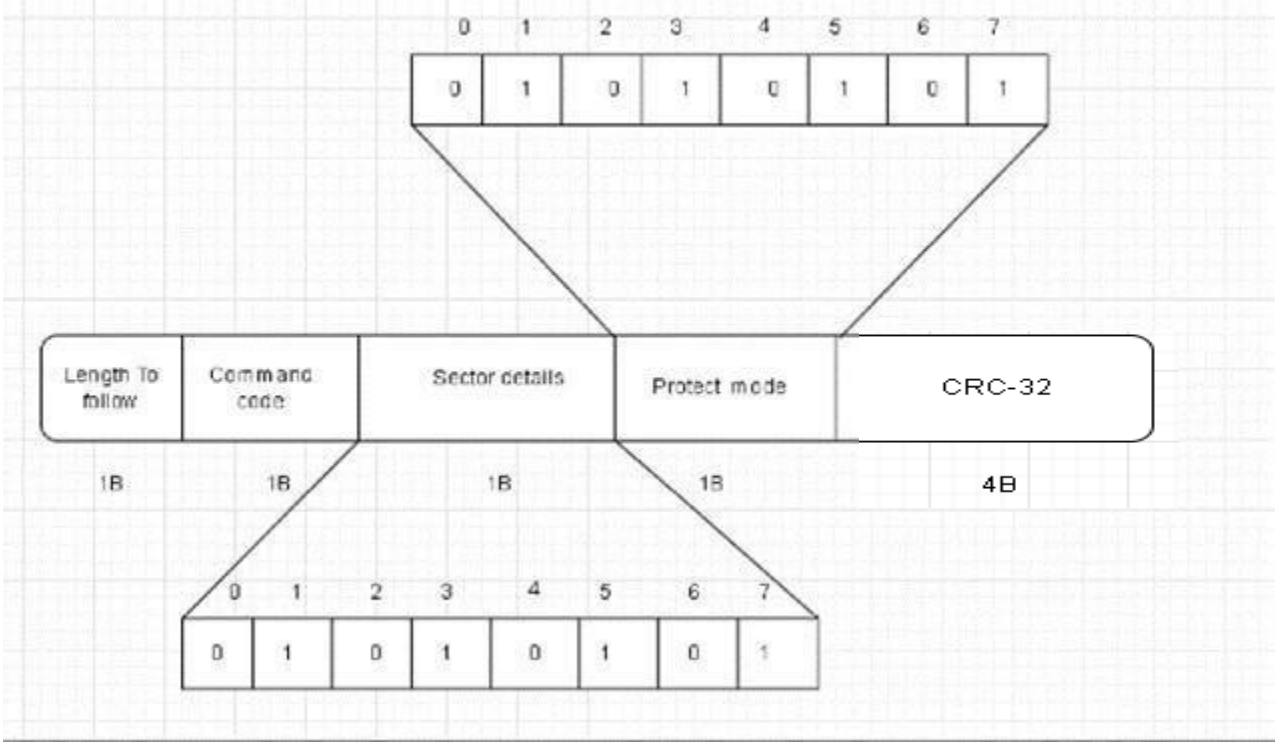


- Packet size = 11+x Bytes.
- Length to follow= 10+x Bytes.
- Command Code = 0x57.

Request: This service sends an image of length x and specifies the base address for the image.

Reply: Write un/successful.

8- Service name: Bootloader enable read/write protection



- Packet size = 8 Bytes.
- Length to follow= 7 Bytes.
- Command Code = 0x58.

Sector details s protection mode:

- Each bit in these Bytes it`s location(number) represents the sector number to be modified.

Sector details:

- Enable protection → 1.
- No protection→ 0.

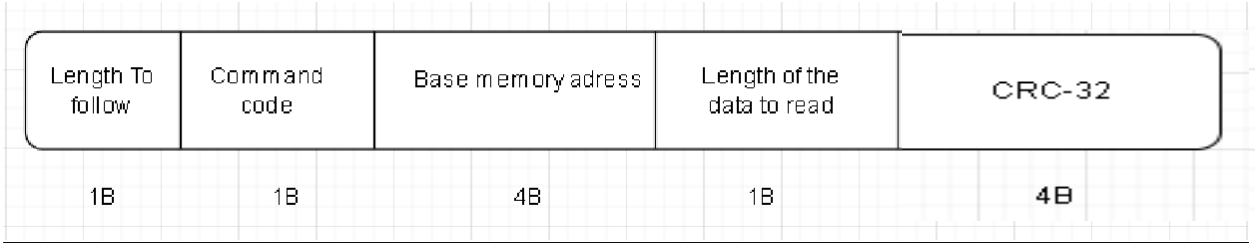
Protection mode:

- Write protection→ 1.
- (R/W) protection → 0.

Request: En/disable read/write protection using option bytes.

Reply: RDP status.

9- Service name: Bootloader Memory read:

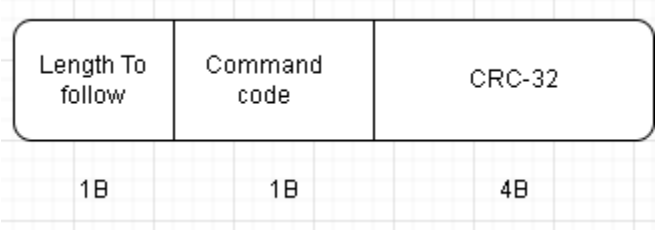


- Packet size = 11 Bytes.
- Length to follow = 10 Bytes.
- Command Code = 0x59.

Request: This service requests to read from the flash.

Reply: Flash content.

10- Service name: Bootloader read sector protection status(option bytes):

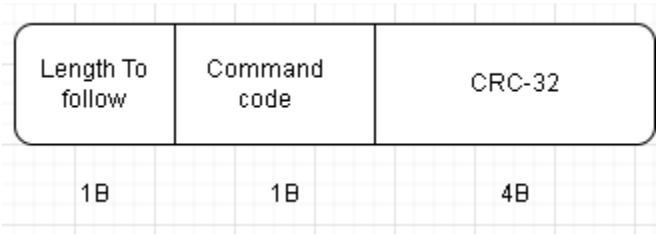


- Packet size = 6 Bytes.
- Length to follow= 5 Bytes.
- Command Code = 0x5A.

Request: This service requests to RDP protection status.

Reply: RDP protection status.

11- Service name: Bootloader read option bytes:

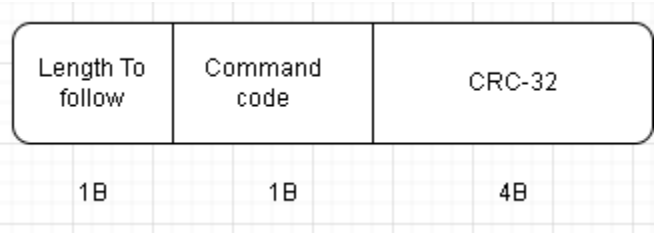


- Packet size= 6 Bytes.
- Length to follow= 5 Bytes.
- Command Code = 0x5B.

Request: This service requests to read option bytes.

Reply: option bytes.

12- Service name: Bootloader disable read/write protection:



- Packet size= 6 Bytes.
- Length to follow= 5 Bytes.
- Command Code = 0x5C.

Request: This service resets RDP protection to default state (level1).

Reply: operation status.