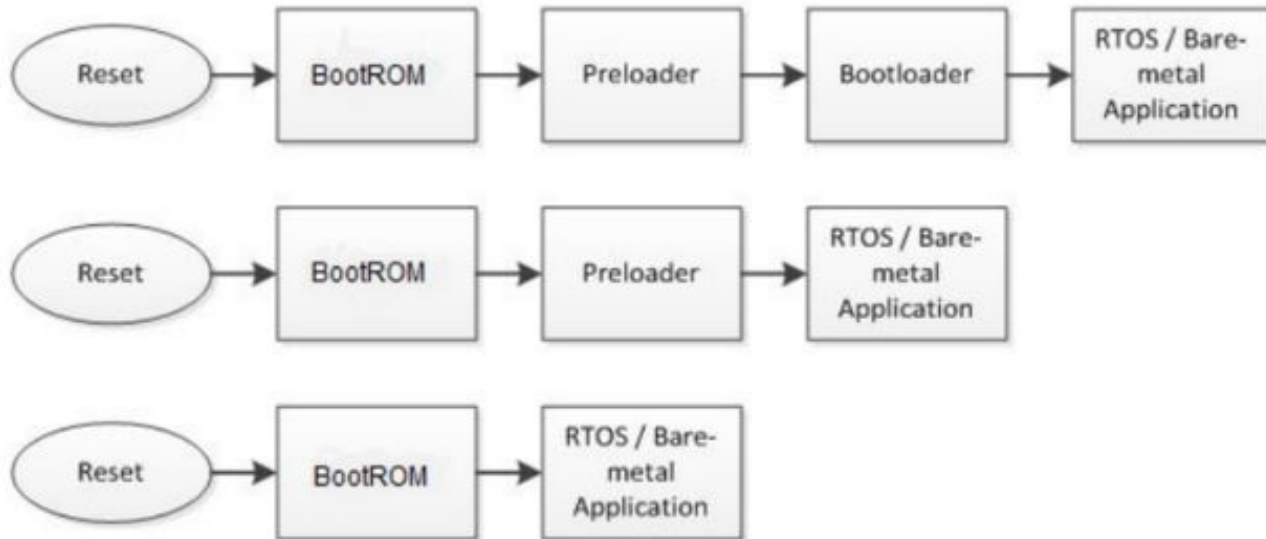

부트로더

Hancheol Cho

부트로더란?

- 어플리케이션/펌웨어가 실행되기전에 초기화하고 다운로드 기능

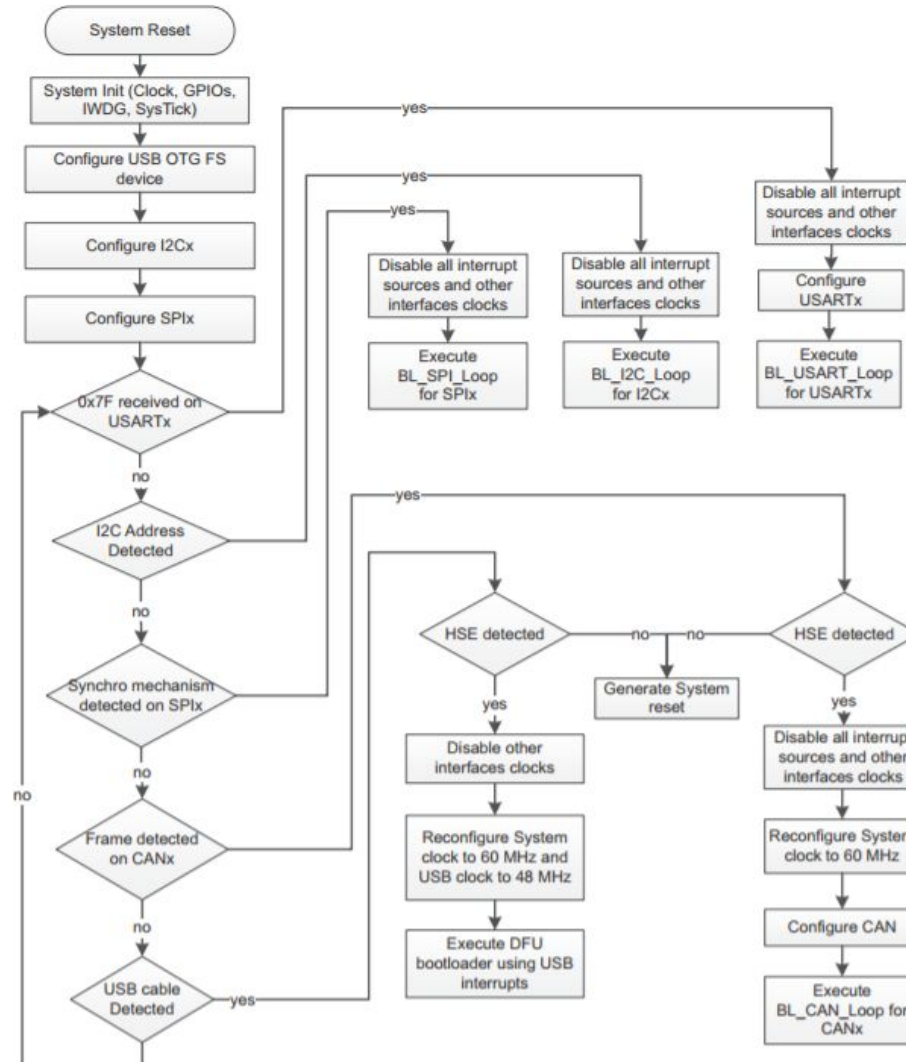


STM BOOTLOADER

- STM MCU 내부에 기본 내장되어 있는 부트로더
 - BOOT 핀 상태로 부트로더 실행 모드 선택함

Boot mode selection		Boot area
BOOT	Boot address option bytes	
0	BOOT_ADD0[15:0]	Boot address defined by user option byte BOOT_ADD0[15:0] ST programmed value: <u>Flash on ITCM at 0x0020 0000</u>
1	BOOT_ADD1[15:0]	Boot address defined by user option byte BOOT_ADD1[15:0] ST programmed value: <u>System bootloader at 0x0010 0000</u>

STM BOOTLOADER



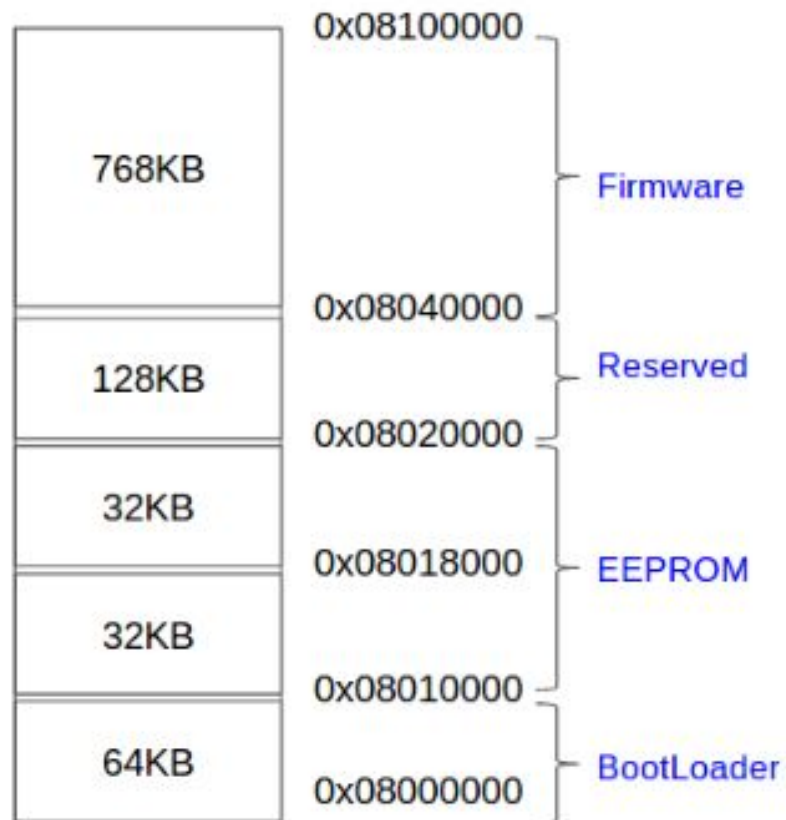
FLASH 메모리 맵

- STM32F746의 FLASH 메모리 구성

Block	Name	Bloc base address on AXIM interface	Block base address on ICTM interface	Sector size
Main memory block	Sector 0	0x0800 0000 - 0x0800 7FFF	0x0020 0000 - 0x0020 7FFF	32 KB
	Sector 1	0x0800 8000 - 0x0800 FFFF	0x0020 8000 - 0x0020 FFFF	32 KB
	Sector 2	0x0801 0000 - 0x0801 7FFF	0x0021 0000 - 0x0021 7FFF	32 KB
	Sector 3	0x0801 8000 - 0x0801 FFFF	0x0021 8000 - 0x0021 FFFF	32 KB
	Sector 4	0x0802 0000 - 0x0803 FFFF	0x0022 0000 - 0x0023 FFFF	128 KB
	Sector 5	0x0804 0000 - 0x0807 FFFF	0x0024 0000 - 0x0027 FFFF	256 KB
	Sector 6	0x0808 0000 - 0x080B FFFF	0x0028 0000 - 0x002B FFFF	256 KB
	Sector 7	0x080C 0000 - 0x080F FFFF	0x002C 0000 - 0x02F FFFF	256 KB
Information block	System memory	0x1FF0 0000 - 0x1FF0 EDBF	0x0010 0000 - 0x0010 EDBF	60 Kbytes
	OTP	0x1FF0 F000 - 0x1FF0 F41F	0x0010 F000 - 0x0010 F41F	1024 bytes
	Option bytes	0x1FFF 0000 - 0x1FFF 001F	-	32 bytes

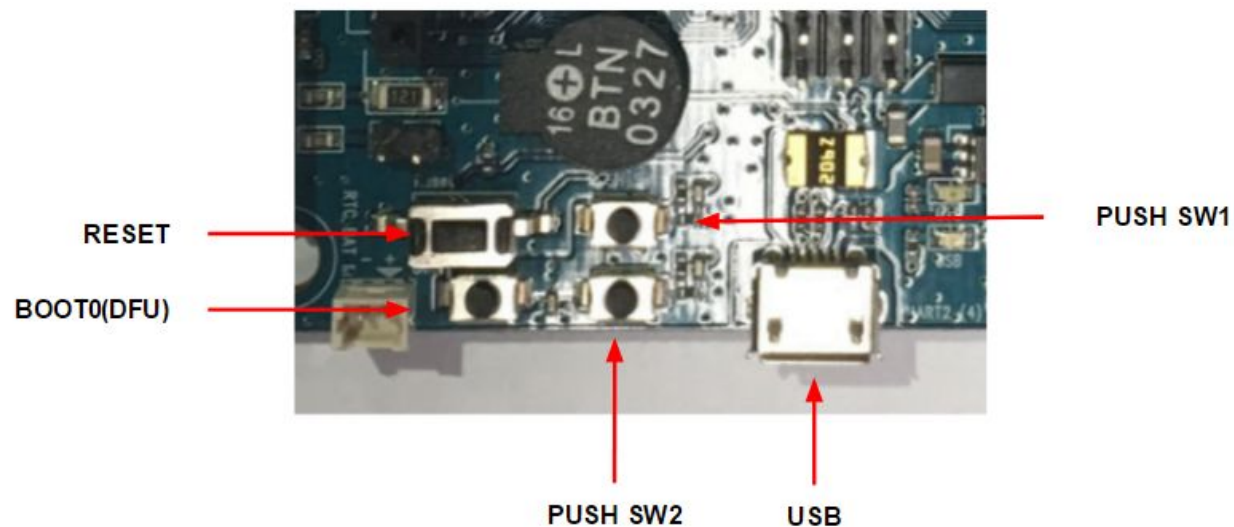
FLASH 메모리 맵

- FLASH 메모리의 영역 할당



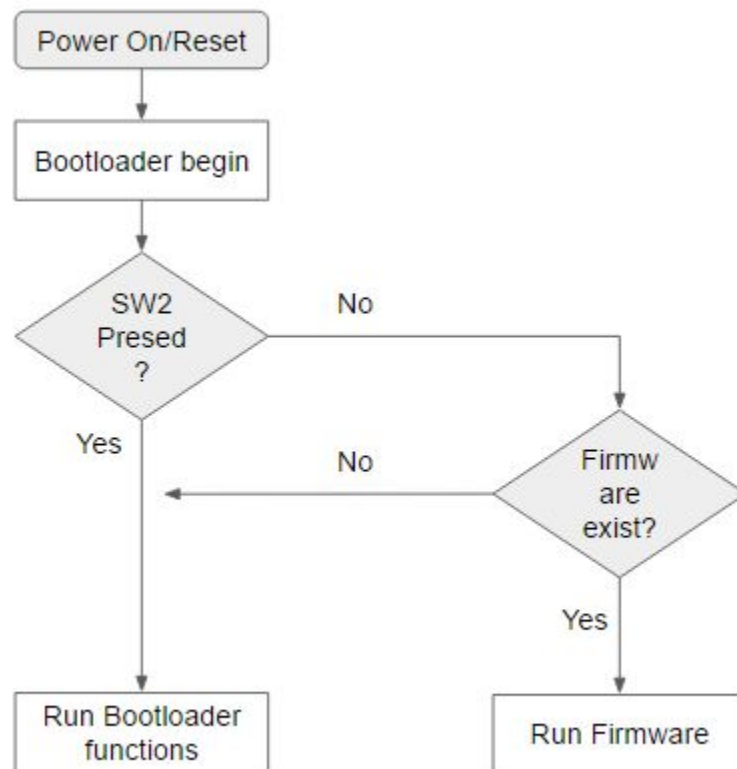
부트로더 실행

- 부트로더 실행을 위한 별도 버튼 할당



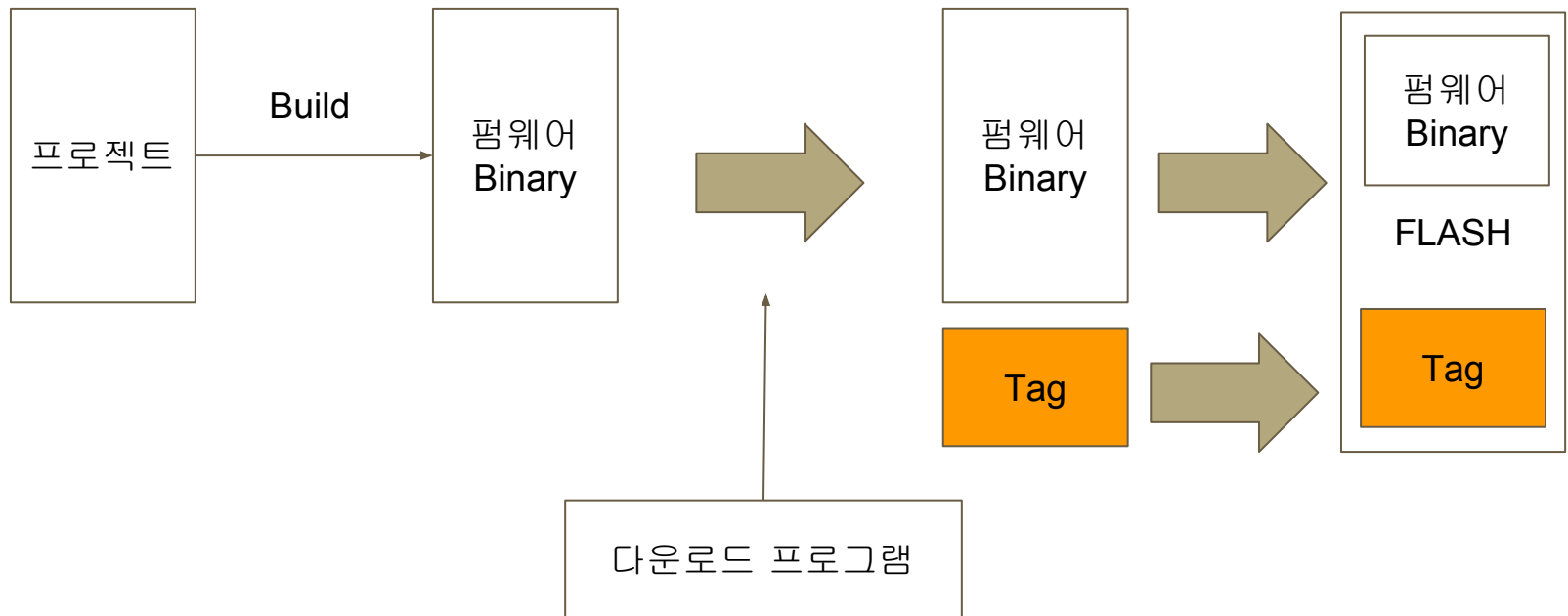
부트로더 실행

- 부팅 순서 정의



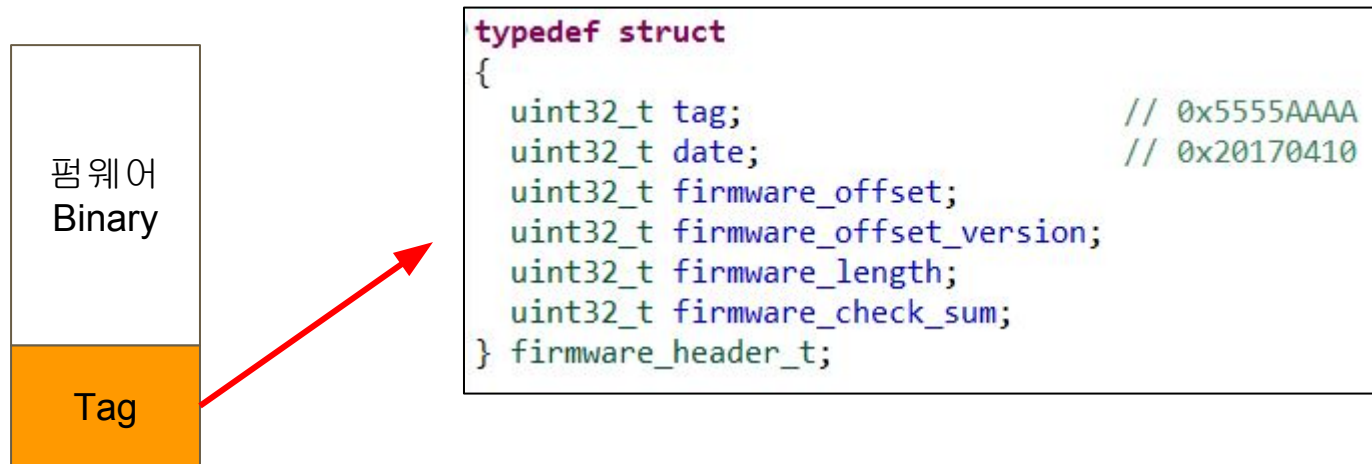
펌웨어 이미지 생성

- 프로젝트 빌드 후에 생성된 펌웨어 Binary에 대한 정보를 Tag정보로 Flash에 저장
 - Tag는 펌웨어 Binary의 유효성 검증용

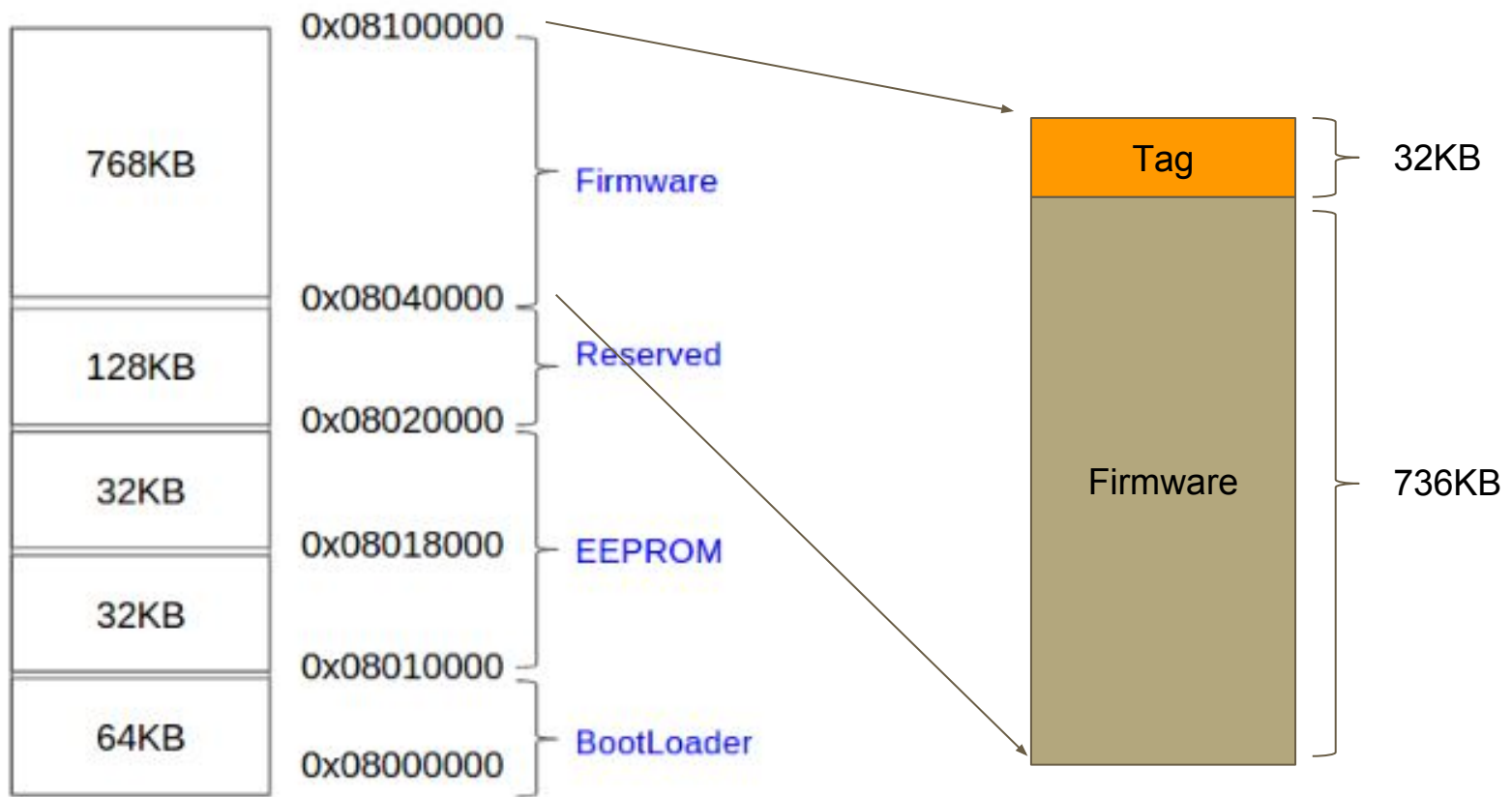


펌웨어 Tag 구성

- 펌웨어 Tag에는 펌웨어 유효성을 검증할 수 있는 데이터가 포함
 - 펌웨어 위치 및 크기와 CRC 데이터를 이용하여 검증 함



Tag를 포함한 Memory Map



메모리 정의

- hw_def.h 에 메모리 주소 정의

```
#define FLASH_FW_SIZE          ((768-32)*1024)    // 736KB
#define FLASH_FW_ADDR_START    0x08040000
#define FLASH_FW_ADDR_END      (FLASH_FW_ADDR_START + FLASH_FW_SIZE)

#define FLASH_TAG_SIZE          (32*1024)          // 32KB
#define FLASH_TAG_ADDR_START    FLASH_FW_ADDR_END
#define FLASH_TAG_ADDR_END      (FLASH_TAG_ADDR_START + FLASH_TAG_SIZE)
```

통신 프로토콜

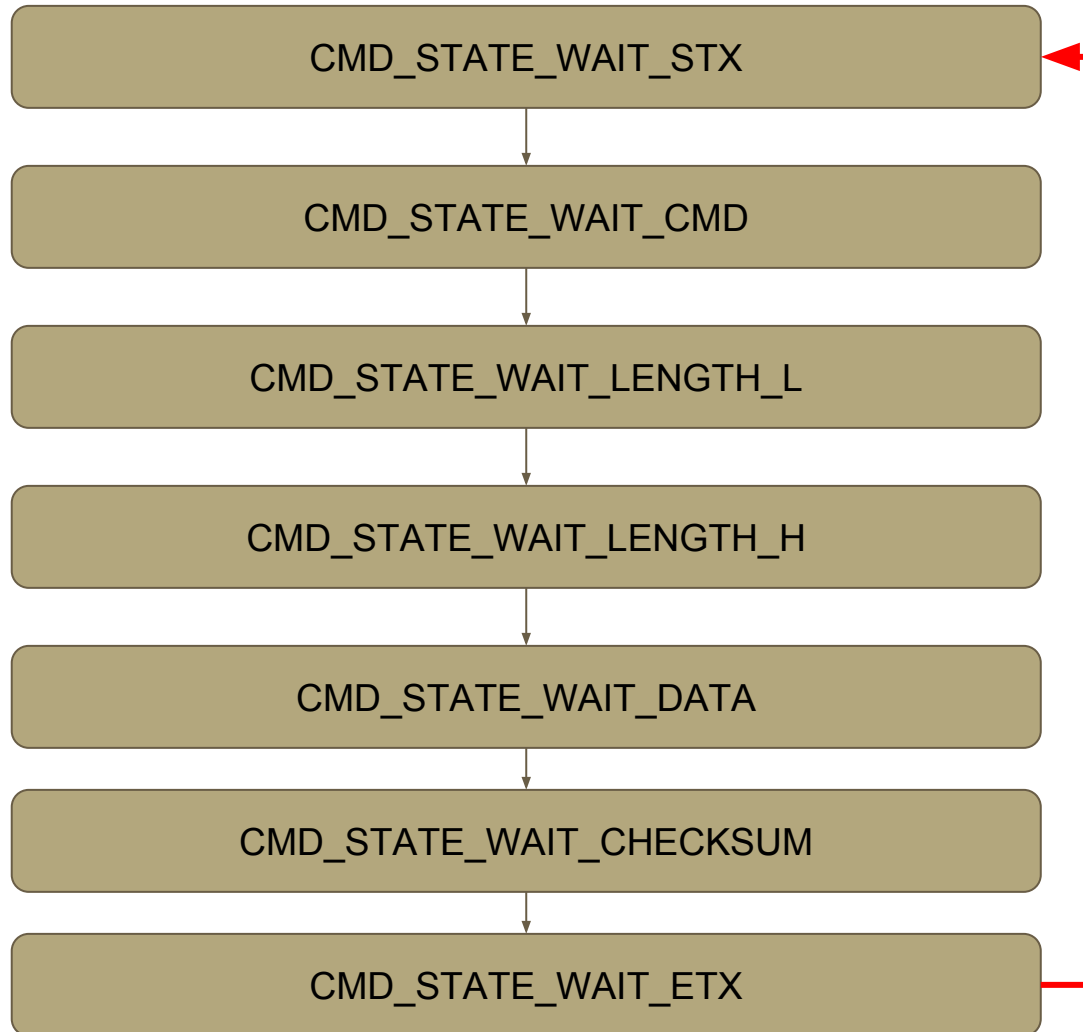
- ## ● 시리얼 통신 프로토콜 정의 - 송신 패킷

[illegible]

- ## ● 시리얼 통신 프로토콜 정의 - 수신 패킷

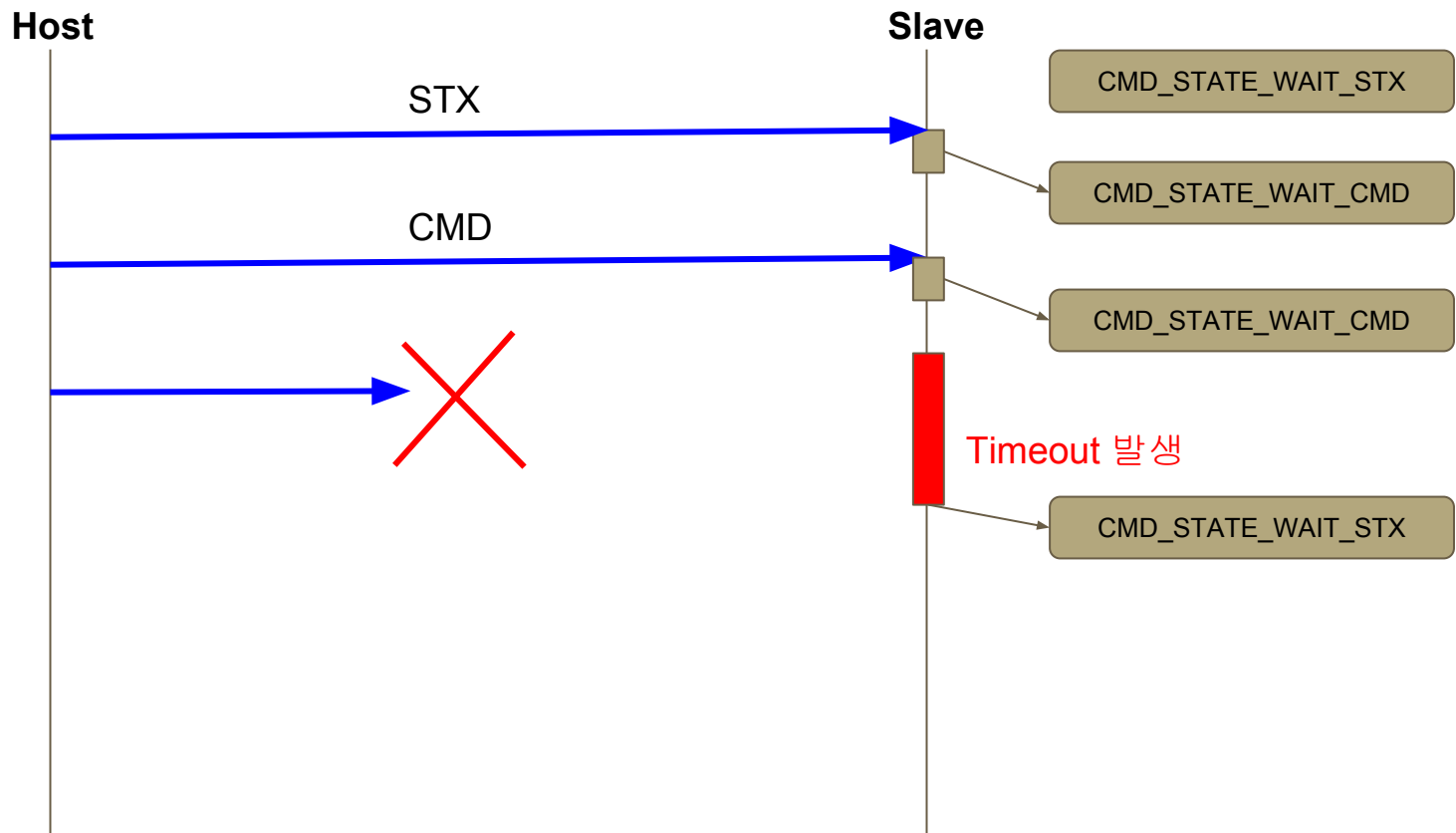
[illegible]

프로토콜 데이터 수신 처리



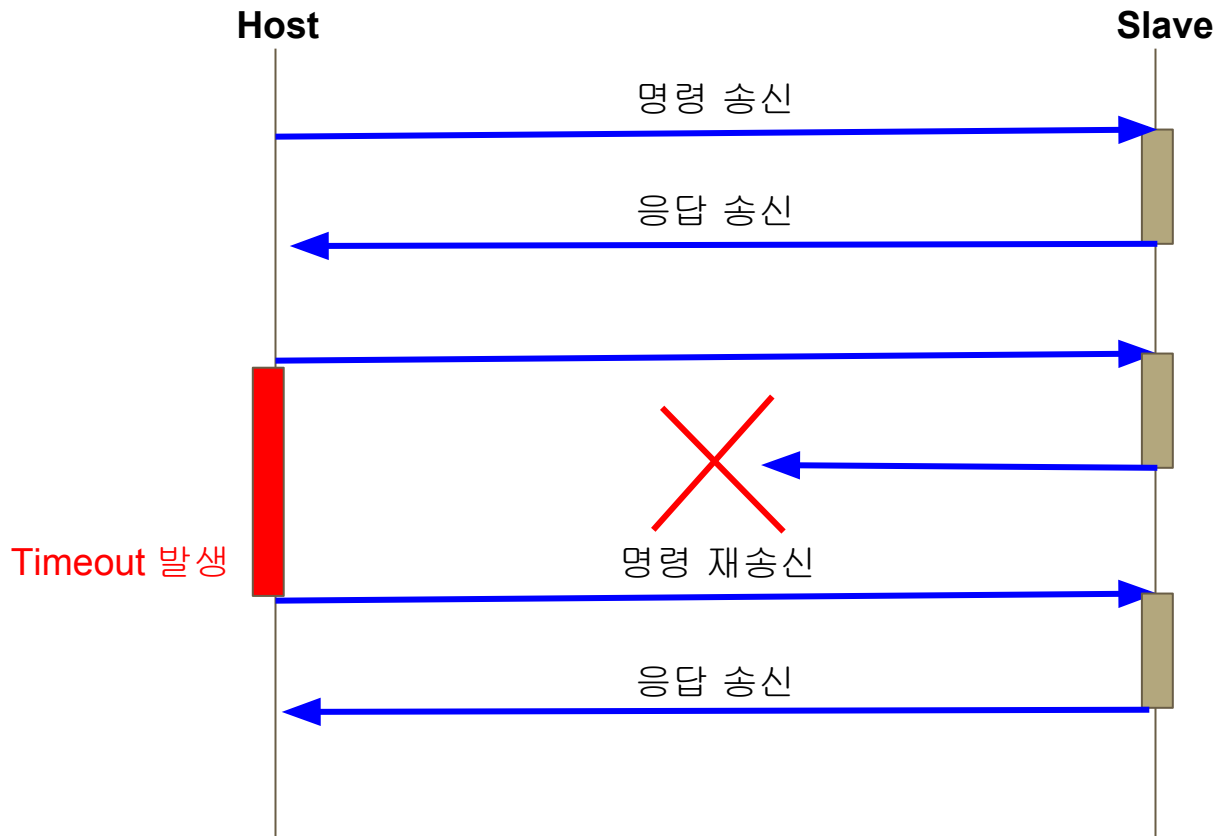
데이터 송/수신 타임아웃

- Host->Slave로 전송하는 바이트 사이의 타임아웃 발생 처리



데이터 송/수신 타임아웃

- Host->Slave로 전송하는 명령어 패킷에 대한 응답 타임아웃 발생 처리



명령어 구성

- 부트로더를 위한 명령어

```
#define BOOT_CMD_READ_VERSION      0x00
#define BOOT_CMD_READ_BOARD_NAME  0x01
#define BOOT_CMD_FLASH_ERASE       0x02
#define BOOT_CMD_FLASH_WRITE       0x03
#define BOOT_CMD_FLASH_READ        0x04
#define BOOT_CMD_FLASH_ERASE_FW    0x05
#define BOOT_CMD_FLASH_WRITE_FW    0x06
#define BOOT_CMD_FLASH_WRITE_TAG   0x07
#define BOOT_CMD_FLASH_READ_TAG    0x08
#define BOOT_CMD_FLASH_VERIFY_FW   0x09
#define BOOT_CMD_JUMP_TO_FW        0x010
```

```
static void bootCmdReadVersion(cmd_t *p_cmd);
static void bootCmdReadBoardName(cmd_t *p_cmd);
static void bootCmdFlashErase(cmd_t *p_cmd);
static void bootCmdFlashEraseFw(cmd_t *p_cmd);
static void bootCmdFlashWrite(cmd_t *p_cmd);
static void bootCmdFlashWriteFw(cmd_t *p_cmd);
static void bootCmdFlashWriteTag(cmd_t *p_cmd);

static void bootCmdJumpToFw(cmd_t *p_cmd);
static void bootCmdFlashVerifyFw(cmd_t *p_cmd);
```

명령어 처리 순서

- cmdReceivePacket 함수에서 통신 프로토콜 분석 후 정상 패킷 수신
- 수신된 패킷의 명령어에 따라 해당 함수를 수행함

cmdReceivePacket()

→ bootProcessCmd()

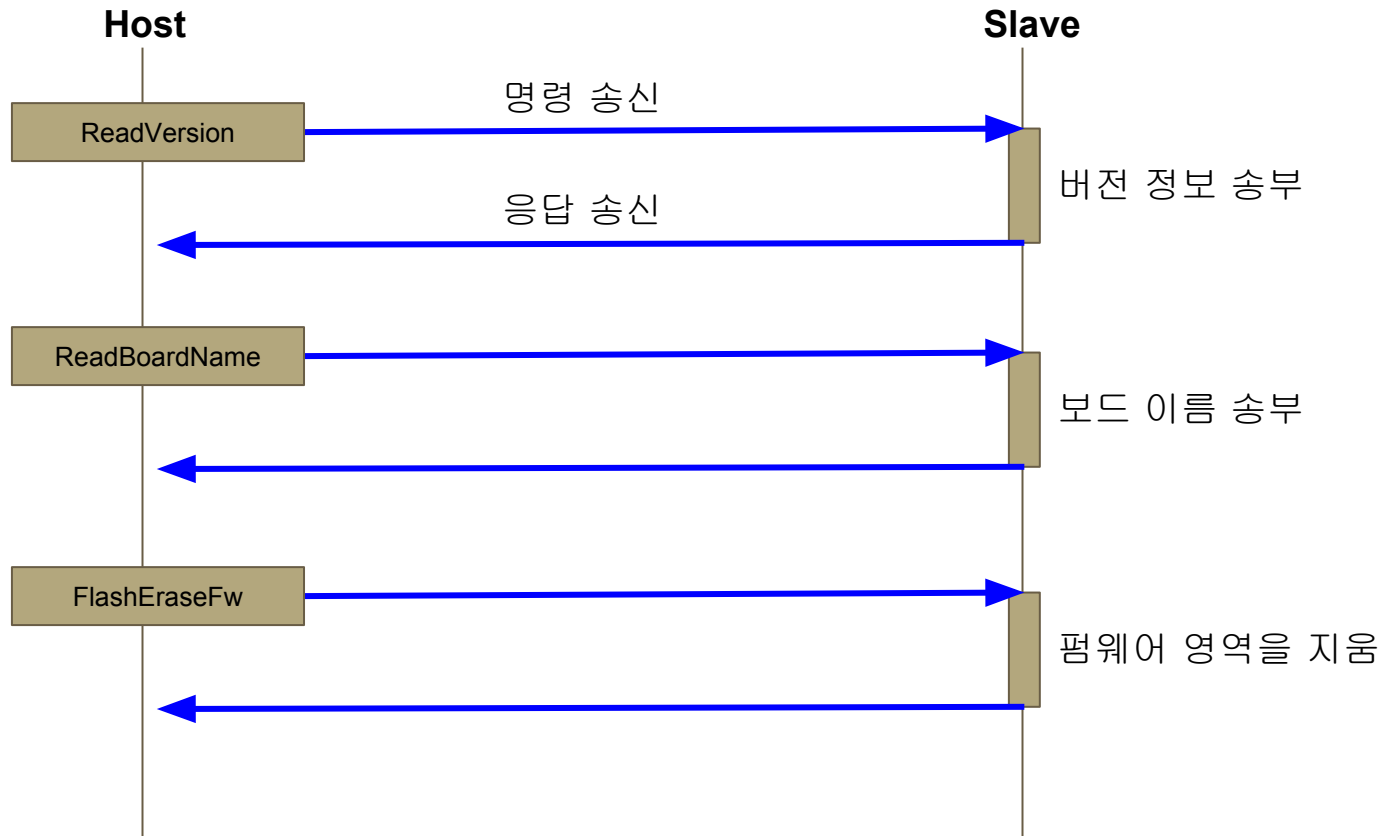
```
void bootProcessCmd(cmd_t *p_cmd)
{
    switch(p_cmd->rx_packet.cmd)
    {
        case BOOT_CMD_READ_VERSION:
            bootCmdReadVersion(p_cmd);
            break;

        case BOOT_CMD_READ_BOARD_NAME:
            bootCmdReadBoardName(p_cmd);
            break;

        case BOOT_CMD_FLASH_ERASE:
            bootCmdFlashErase(p_cmd);
            break;

        case BOOT_CMD_FLASH_WRITE:
            bootCmdFlashWrite(p_cmd);
            break;
    }
}
```

펌웨어 다운로드 순서



펌웨어 다운로드 순서

