

ARM Coretex-M

# 펌웨어 설계



## 하드웨어 테스트 코드 분석



한국기술교육대학교  
온라인평생교육원

## 학습목표

- STM32 라이브러리 소스 코드들을 분석할 수 있다.
- 소스 코드 분석을 위한 툴을 사용할 수 있다.
- 하드웨어 자가 진단 테스트를 위한 소프트웨어를 사용할 수 있다.

## 학습내용

- STM32 라이브러리 소스 코드 분석
- 하드웨어 자가 진단 테스트하기

## STM32 라이브러리 소스 코드 분석



## STM32 라이브러리 소스 코드 분석

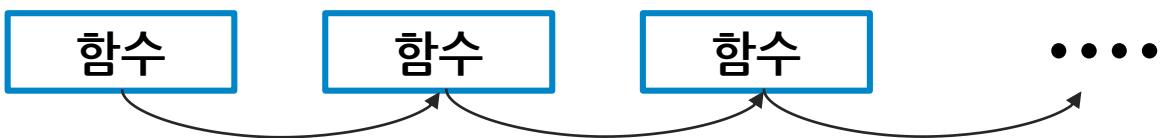
### 소스 코드 분석이란?

#### 소스 코드 분석

프로그래밍 언어로 작성된 소스 코드간의 관계를 분석하여 프로그램의 흐름을 이해하는 것

- ... 하나의 프로그램이 완성되기 위해 대부분 여러 개의 파일을 사용
- ... 여러 개의 파일간에 얹힌 소스 코드들의 관계에서 전역변수, 지역변수, 함수간의 호출등을 분석해야 함
- ... 수작업으로 분석하기에 코드가 복잡한 경우가 많기 때문에 분석 툴을 사용할 수 있음

#### 함수 호출의 반복



하나의 함수가 또 다른 서브 함수를 호출하고 다시 또 다른 서브함수를 호출하는 경우가 많은데 함수를 따라 가다 보면 기억의 한계에 부딪힘

# STM32 라이브러리 소스 코드 분석



## STM32 라이브러리 소스 코드 분석

### 소스 코드 분석 툴의 종류

- ... 소스코드 분석 툴은 다른 파일에 존재하는 전역변수나 함수 등의 추적이 용이해야 함
- ... 개발 환경에 따라 적절한 분석 툴을 선택해야 함
- ... 소스코드를 분석할 수 있는 툴로는 통합 개발 환경이라고 하는 IDE (Integrated Development Environment) 개발 환경을 들 수 있음

### 마이크로 소프트사

Visual Studio



Eclipse



- ... IDE개발환경에는 함수 호출을 추적할 있는 기능들을 대부분 내장함
- ... IDE와 같은 컴파일러는 없지만 소스 코드 분석을 위한 전용 툴인 source insight 도 있음

### Source Insight



# STM32 라이브러리 소스 코드 분석



## STM32 라이브러리 소스 코드 분석

### 소스 코드 분석 툴 설치하기



소스코드 분석툴 중 source insight를 설치할 경우  
Source Insight는 유료 프로그램으로 테스트를 위해 **trial** 모드로 진행함

**1** Source Insight 홈페이지에서 설치파일을 다운로드 한다.

➡ 다운로드 페이지 (<https://www.sourceinsight.com/download/>)

**2** 설치파일을 실행하여 설치한다.

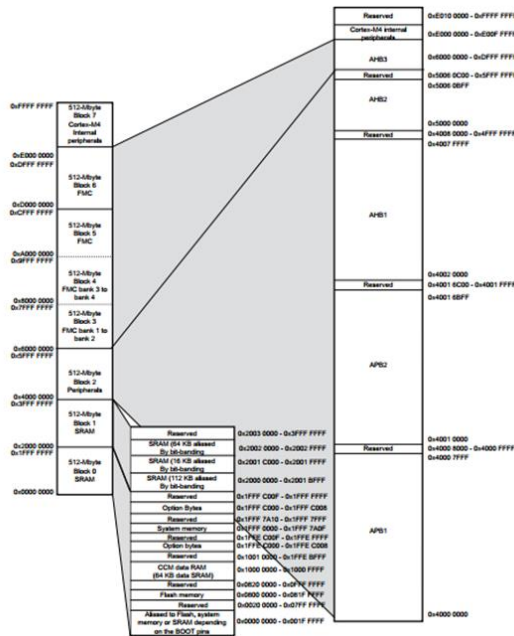
# 하드웨어 자가 진단 테스트하기



## STM32F429의 SRAM 영역

## STM32F429의 SRAM 영역

### STM32F429ZI의 메모리 맵



STM32F429는 내부에  
SRAM을 가지고 있음

## STM32F429의 메모리 맵

- ... SRAM영역을 확인할 수 있음
- ... 내부 SRAM은 총 3개의 영역으로 나뉘어져 있음

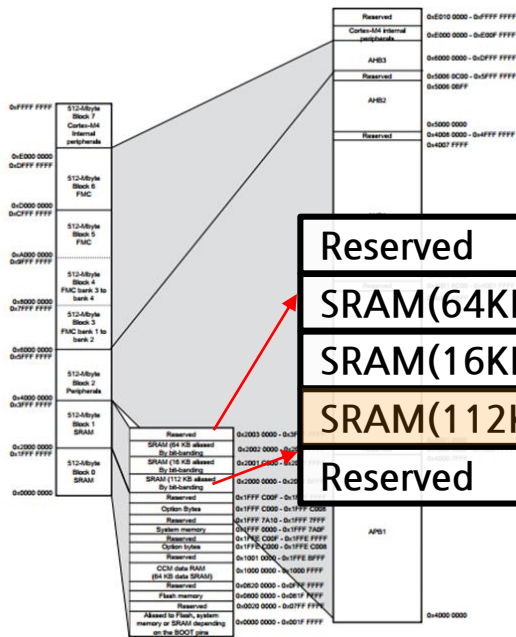
# 하드웨어 자가 진단 테스트하기



## STM32F429의 SRAM 영역

## STM32F429의 SRAM 영역

### STM32F429ZI의 메모리 맵



Reserved

SRAM(64KB aliased By bit-banding)

SRAM(16KB aliased By bit-banding)

SRAM(112KB aliased By bit-banding)

Reserved

SRAM1

... 112KB

... 0x20000000~0x2001BFFF

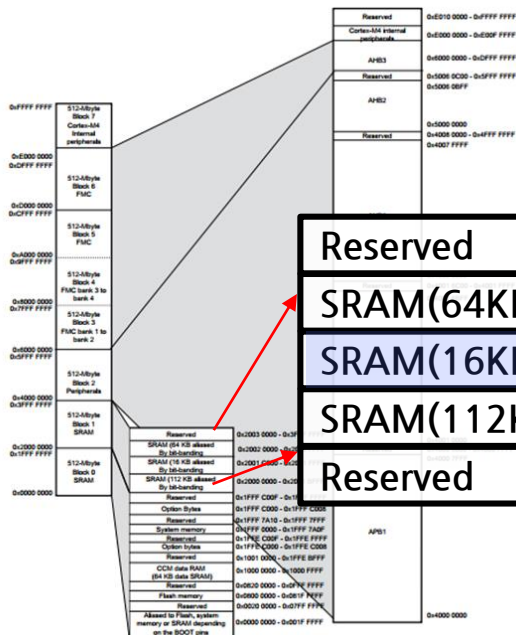
# 하드웨어 자가 진단 테스트하기



## STM32F429의 SRAM 영역

## STM32F429의 SRAM 영역

### STM32F429ZI의 메모리 맵



Reserved

SRAM(64KB aliased By bit-banding)

SRAM(16KB aliased By bit-banding)

SRAM(112KB aliased By bit-banding)

Reserved

## SRAM2

... 16KB

... 0x2001C000~0x2001FFFF



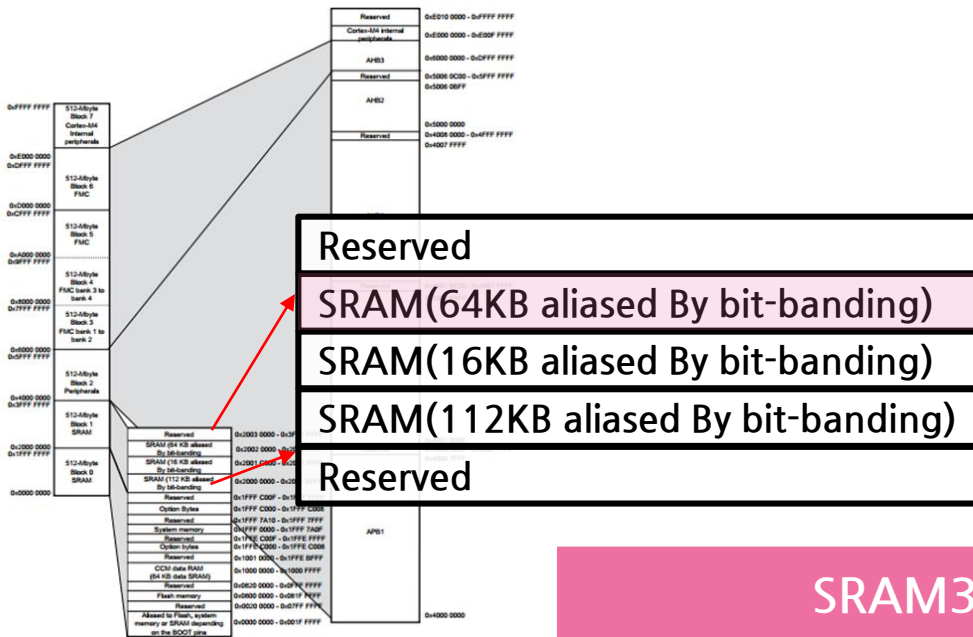
# 하드웨어 자가 진단 테스트하기



## STM32F429의 SRAM 영역

## STM32F429의 SRAM 영역

### STM32F429ZI의 메모리 맵



### SRAM3

- ... 64KB
- ... 0x20020000~0x2002FFFF



메모리는 RAM과 ROM으로 나뉘는데 이 SRAM 영역은 RAM 영역임

# 하드웨어 자가 진단 테스트하기



## STM32F429의 SRAM 영역

### STM32 라이브러리의 SRAM 영역 지정

- STM32 라이브러리에 SRAM의 3개의 영역으로 지정하는 매크로를 찾을 수 있음
- Source insight를 통해 Stm32f429xx.h 파일에 있는 3개의 SRAM영역의 매크로를 찾을 수 있음
- SRAM 영역의 시작 주소

SRAM1\_BASE

SRAM2\_BASE

SRAM3\_BASE

```

#define FLASH_BASE          0x08000000UL /*< FLASH(up to 2 MB) base address in the alias region
#define CCMDATARAM_BASE     0x10000000UL /*< CCM(core coupled memory) data RAM(64 KB) base address in the alias region
#define SRAM1_BASE          0x20000000UL /*< SRAM1(112 KB) base address in the alias region
#define SRAM2_BASE          0x2001C000UL /*< SRAM2(16 KB) base address in the alias region
#define SRAM3_BASE          0x20020000UL /*< SRAM3(64 KB) base address in the alias region
#define PERIPH_BASE         0x40000000UL /*< Peripheral base address in the alias region
#define BKPSRAM_BASE        0x40024000UL /*< Backup SRAM(4 KB) base address in the alias region
#define FMC_R_BASE          0xA0000000UL /*< FMC registers base address
#define SRAM1_BB_BASE       0x22000000UL /*< SRAM1(112 KB) base address in the bit-band region
#define SRAM2_BB_BASE       0x22380000UL /*< SRAM2(16 KB) base address in the bit-band region
#define SRAM3_BB_BASE       0x22400000UL /*< SRAM3(64 KB) base address in the bit-band region
#define PERIPH_BB_BASE      0x42000000UL /*< Peripheral base address in the bit-band region
#define BKPSRAM_BB_BASE     0x42480000UL /*< Backup SRAM(4 KB) base address in the bit-band region
#define FLASH_END           0x081FFFFFUL /*< FLASH end address
#define FLASH_OTP_BASE      0x1FFF7800UL /*< Base address of : (up to 528 Bytes) embedded FLASH OTP Area
#define FLASH_OTP_END       0x1FFF7A0FUL /*< End address of : (up to 528 Bytes) embedded FLASH OTP Area
#define CCMDATARAM_END      0x100FFFFFUL /*< CCM data RAM end address

/* Legacy defines */
#define SRAM_BASE            SRAM1_BASE
#define SRAM_BB_BASE        SRAM1_BB_BASE

```

STM라이브러리의 SRAM영역 지정

# 하드웨어 자가 진단 테스트하기



## ⚙ 메모리 영역 테스트 프로그램 작성 및 테스트

### 🌈 메모리 영역 테스트 코드

- ... 완벽한 시스템이란 없기 때문에 하드웨어 테스트를 진행
- ... 하드웨어 테스트 중 가장 기본이 되는 테스트가 바로 메모리 테스트
- ... 메모리 테스트는 메모리에 여러 가지 패턴을 쓰고 그 패턴을 다시 읽어 보는 과정을 반복함으로써 메모리의 문제점을 파악함
- ... 본 학습에서는 SRAM3 영역을 테스트하는 실습 진행

# 하드웨어 자가 진단 테스트하기



## 메모리 영역 테스트 프로그램 작성 및 테스트

### 메모리 테스트 코드 작성

- 1 배열 test\_pattern 에 쓰고 싶은 패턴을 저장
- 2 SRAM3의 시작주소를 start변수에 저장하고 마지막 주소를 end 변수에 저장
- 3 패턴 0번부터 SRAM3 전 연역에 저장하고 곧바로 읽고 비교하여 패턴이 동일한지 검사

```
static uint32_t test_pattern[] = {
    0x55555555UL,
    0aaaaaaaaUL,
    0xffffffffUL,
    0x00000000UL,
};

int main(void)
{
    uint32_t *p, *start, *end;
    uint32_t pattern_num, count;

    start = SRAM3_BASE;
    end = SRAM3_BASE + 0x10000;
    pattern_num = 0;
    count = 0;
```

```
while (1)
{
    for(p = start; p < end; p++)
        *p = test_pattern[pattern_num];

    for(p = start; p < end; p++)
    {
        if(test_pattern[pattern_num] != *p)
        {
            //Memory Test Fail
            return -1;
        }
    }
    // Memory Test Success
    pattern_num = count % 4;
    count++;
}
```

## 하드웨어 자가 진단 테스트하기



### 메모리 영역 테스트 프로그램 작성 및 테스트

#### 교수님 메모리 테스트 코드 작성 실습 영상

#### 1 제시된 코드 작성을 통한 테스트 진행

## 요점노트

## 1. STM32 라이브러리 소스 코드 분석



- STM32 라이브러리 소스 코드 분석
  - 소스 코드 분석은 프로그래밍 언어로 작성된 소스 코드간의 관계를 분석하여 프로그램의 흐름을 이해하는 것
  - 여러 개의 파일간에 얹힌 소스 코드들의 관계에서 전역변수, 지역변수, 함수간의 호출 등을 분석해야 함
  - 소스코드를 분석할 수 있는 툴로는 마이크로 소프트사의 visual studio나 Eclipse, source insight 등이 있음

## 요점노트

## 2. 하드웨어 자가 진단 테스트하기



- 하드웨어 자가 진단 테스트하기
  - STM32F429의 메모리 맵을 통해 SRAM영역을 확인할 수 있음
  - STM32 라이브러리에 3개의 SRAM영역을 나타내는 매크로를 쉽게 찾을 수 있음
  - 메모리 테스트는 메모리에 여러 가지 패턴을 쓰고 그 패턴을 다시 읽어 보는 과정을 반복함으로써 메모리의 문제점을 파악