

듀얼 बैं크 플래시가 내장된 마이크로 컨트롤러의 장점

프리스케일 듀얼 बैं크 플래시는 CPU 지연 상황을 피하고 성능을 강화함으로써 애플리케이션 설계를 간소화하고, 코드 실행 도중에도 인터럽트 서비스 루틴의 작동을 유지하며 램에 루틴을 복사할 필요성을 없애주는 간단한 아이디어이다. 이러한 모든 기능을 활용하면 코드 실행 도중 플래시 메모리 쓰기 또는 소거 작업이 필요한 최종 애플리케이션을 더 쉽게 설계 및 구현할 수 있으며, 부트로더 또는 EEPROM 에뮬레이션과 같은 애플리케이션 또한 단지 적절한 메모리 할당을 고려하고 플래시 루틴 실행 도중 통신 주변기기를 중단하는 것만으로 이러한 기능을 활용하여 효율성을 높일 수 있다.

자료 제공 : Rafael Perález / 애플리케이션 엔지니어링 관리자
프리스케일 / www.freescale.co.kr

마이크로 컨트롤러는 지난 수십 년 동안 CPU 성능, 통신 인터페이스, 아날로그-디지털 및 디지털-아날로그 주변기기, 메모리 용량, 메모리 읽기 및 쓰기 시간 등의 측면에서 기하급수적으로 발전해왔다. 특히 비휘발성 임베디드 메모리(USB 플래시 드라이브, 메모리 카드 등에 사용되는 플래시 메모리)가 내장된 마이크로 컨트롤러를 집중적으로 살펴보면, 첫 번째 디바이스인 OTP(One-time-Programmable)에서 EPROM(Electrically Programmable Read-Only memory), EEPROM(“소거 가능”이라는 특성이 추가되어 자외선을 사용하지 않고도 소거가 가능함)으로 이어졌으며, 이제 임베디드 플래시 메모리(경우에 따라 플래시 EEPROM이라고도 부름)가 가장 많이 사용되고 있다.

EEPROM과 플래시는 모두 전기적으로 소거 및 기록 가능한 메모리라는 점에서 개념상 유사하지만, 몇 가지 차이점이 존재한다. 원래 플래시는 대용량 블록으로만 기록 가능했지만, 아키텍처의 8비트, 16비트 또는 32비트 지원 여부와 짝수 주소 정렬 필요성에 따라 싱글 바이트, 워드 또는 더블 워드를 기록할 수 있다는 점에서

현재는 두 가지가 유사하며, 주된 차이점은 소거 절차이다. EEPROM은 소거 용량이 작은 반면(대부분의 경우 싱글 바이트 단위로 소거 가능) 플래시는 경우에 따라 블록 또는 페이지로 불리는 대용량 섹터 단위로 소거해야 하며, 사용하는 디바이스에 따라 섹터가 수 바이트에서 수천 바이트에 이를 수 있다.

플래시의 인기가 높은 데에는 두 가지 중요한 이유가 있으며, 하나는 섹터가 추가됨에 따라 소거 절차가 일반 EEPROM 디바이스보다 훨씬 빠르다는 점이다. EEPROM을 사용할 때는 소거 시간이 상당히 느리며, 대개 한 바이트에 밀리초 단위가 필요하다. 플래시의 소거 시간도 대략 비슷한 수준이지만, 소거 단위가 전체 섹터에 적용된다. 따라서 디바이스 프로그래밍에 소요되는 시간이 단축되고 그 결과 제조 공정에 필요한 시간 또한 줄어든다. 다른 한 가지 이유는 플래시 메모리의 제조 비용이 EEPROM보다 훨씬 저렴하므로 다량의 플래시 메모리가 내장된 MCU를 제조하는 것이 더 비용 효율적이라는 점이다.

임베디드 플래시가 내장된 마이크로 컨트롤러는 시스템 내(In-

System) 프로그래밍을 지원한다. 이는 최종 애플리케이션에 사용되는 PCB(인쇄 회로 기판)에서 마이크로 컨트롤러를 프로그래밍할 수 있음을 의미한다. 경우에 따라 프로그래밍 모드를 시작 및 종료하는 데 일부 회로를 추가해야 할 수도 있지만, 디바이스를 분리하지 않고 또는 먼저 소켓 내에서 프로그래밍이 가능하다는 점은 가치가 있으며, 이는 소프트웨어 개발이 완료되지 않았더라도 보드를 완전히 조립할 수 있음을 의미한다. 또한 최초 발표 후에 하드웨어를 전혀 변경하지 않고 소프트웨어를 업그레이드할 수 있으며, 발표 전에 완벽한 버전의 소프트웨어가 필요한 OTP 또는 ROM 기반 디바이스에 비해 제품화 기간을 단축할 수 있다.

이러한 혁신의 다음 단계는 마이크로 컨트롤러 플래시 메모리에 자체 프로그래밍 기능을 추가하여 런타임 도중에 MCU가 쓰기/소거 작업을 실행할 수 있도록 하는 것이다. 여기에는 두 가지 커다란 장점이 있으며, 첫 번째는 코드 자체에 전체 애플리케이션을 다시 프로그래밍할 수 있는 루틴을 포함시켜 주변기기(일반적으로 부트로더라 불리며, 대개 직렬 인터페이스의 주변기기가 사용됨)를 통한 원격 업데이트를 지원함으로써, 현장에 투입된 이후에 설계를 업데이트할 수 있는 유연성을 제공한다는 점이다. 두 번째는 런타임 도중에 공조 온도 또는 TV에 프로그래밍된 채널 등과 같은 비휘발성 데이터를 저장할 수 있다는 점이다.

플래시 자체 프로그래밍

이와 관련하여, 다음은 듀얼 플래시 어레이 없이 플래시를 시스템 내 및 런타임 도중에 기록하는 방법에 대한 설명이다.

런타임 도중 쓰기 및 소거 절차를 실행하려면 높은 전압을 사용하여 각각의 비트를 설정하거나 또는 지워야 한다. 플래시 बैं크에 이와 같은 고전압이 적용되는 시간 동안에는 전체 메모리 बैं크를 읽을 수 없으며, 따라서 다음과 같은 두 가지 대안을 사용하여 시스템의 작동 상태를 유지해야 한다. 하나는 다른 메모리(일반적으로 RAM)에서 플래시를 쓰기/소거하는 코드를 실행하는 것이며, 다른 하나는 플래시 작업이 실행되는 동안 CPU가 코드 실행을 지연할 수 있는 방안을 마련하는 것이다.

두 가지 방법 모두 추가적인 제약이 존재한다. 일반적으로 인터

Running from RAM		CPU Stall	
Pros	Cons	Pros	Cons
It is faster to write/erase flash when running from code already stored in RAM	If the code is in RAM all the time, the amount of memory for variables is reduced.	It is easier to implement the routine since there is no need to move the code from Flash to RAM	Code execution is totally stopped during a flash command execution.
The system can execute other instructions while Flash is being updated	When running from the stack, it is slower to make a copy of the routine every time is needed	Memory usage is more efficient: RAM is used only for variables	The system needs to be designed in a way that it can be stopped from a few micro-seconds to a few milli-seconds

표 1. 플래시를 시스템 내 및 런타임 도중에 기록하는 방법

럽트 벡터가 플래시 메모리 내에 위치하므로 플래시 명령이 실행되는 동안에는 인터럽트가 비활성화되어야 하며, 이는 해당 시간 동안 CPU가 플래시를 읽을 수 없으며 필요할 경우에도 인터럽트 벡터를 가져올 수 없기 때문이다. I2C, UART 또는 USB 등의 직렬 주변기기를 사용하는 시스템의 경우, 해당 주변기기는 초당 길

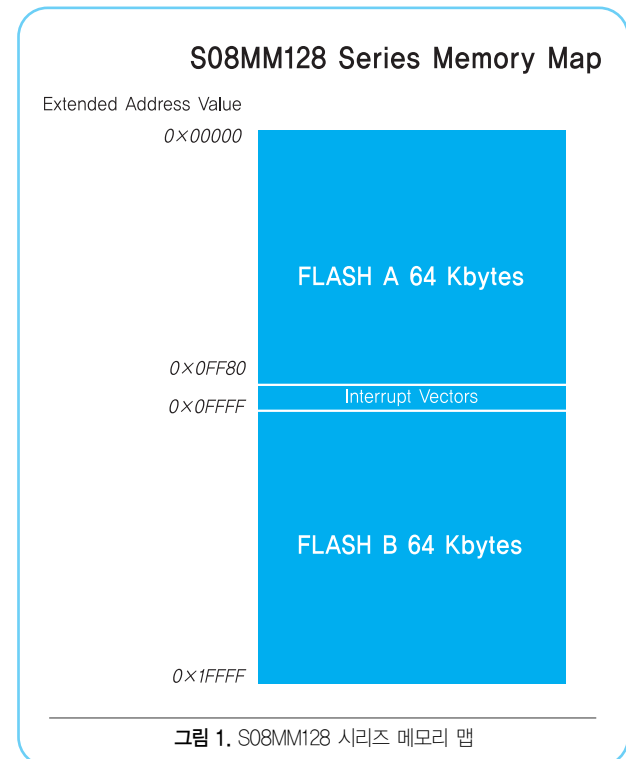


그림 1. S08MM128 시리즈 메모리 맵

로비트(UART 또는 I2C 버스)에서 초당 메가비트(예: USB) 단위로 정보를 주고 받을 수 있으므로, 몇 밀리초 동안이라도 인터럽트가 비활성화되면 그 기간 동안 많은 정보가 손실될 수 있다. 따라서 플래시가 수정되는 동안에는 이러한 직렬 통신을 중단하고, 인터럽트가 다시 활성화되면 모든 정보를 복구할 수 있도록 시스템을 설계해야 한다.

듀얼 बैं크 플래시 솔루션

듀얼 बैं크 플래시란 동일한 디바이스 내에 두 가지 다른 플래시 블록이 존재하는 솔루션을 의미한다. 지금부터는 프리스케일 MC9S08MM128 MCU를 예로 들어 설명하겠다. 이 디바이스에는 각각 64KB의 어레이 2개로 분할된 128KB의 플래시 메모리가 내장되어 있다. 이전 장에서 플래시의 쓰기 또는 소거 작업이 진행되는 동안에는 전체 플래시 블록을 읽을 수 없음을 밝힌 바 있다. 플래시 작업을 실행하기 위한 두 가지 대안 즉, CPU 지연 또는 램에서 실행하는 방법이 언급되었다. 듀얼 बैं크 플래시에도 동일한 논리가 적용되지만, 이제 서로 다른 बैं크 2개가 있으므로 플래시 A에서 플래시 B에 쓰기 또는 소거하는 코드를 실행하거나 그 반대의 작업이 가능하다.

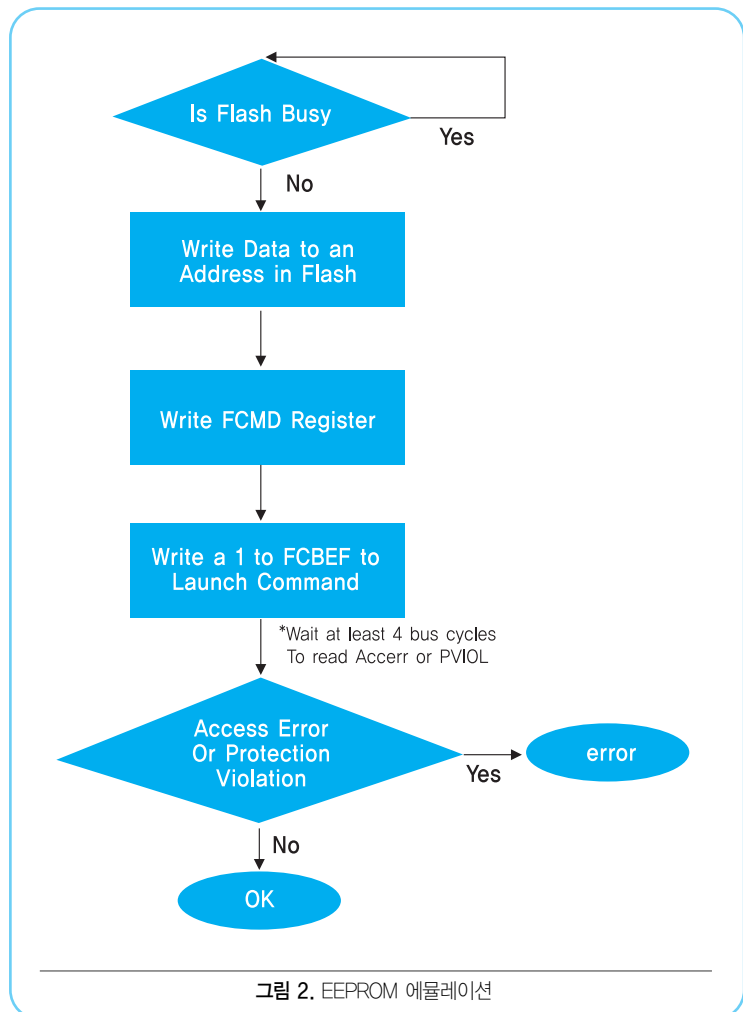
비휘발성 메모리를 사용하여 변수를 저장하는 경우, 모든 비휘발성 변수를 하나의 플래시 बैं크에 저장하는 방식 즉, 한 블록을 의사 EEPROM으로, 다른 한 블록을 코드용으로 사용하도록 듀얼 बैं크 플래시를 구성할 수 있다. 예를 들어, 모든 데이터를 플래시 B에 저장하고, 메모리를 쓰기 및 소거할 코드는 플래시 A에서 실행함으로써 램과 스택을 더 효율적으로 활용하는 방식이다.

또한 듀얼 बैं크 플래시 마이크로 컨트롤러에는 CPU 지연이 필요 없다. 메모리의 절반에만 높은 전압이 필요하며 나머지 절반은 정상적인 코드 실행을 계속할 수 있으므로 시스템을 계속 가동할 수 있다. 이는 특히 코드 블로킹(코드 실행을 계속하려면 CPU를 정지시키거

나 루프에서 이벤트가 발생하기를 기다리는 코드의 일부, 이 경우 플래시 명령이 완료되기를 기다림)을 방지하도록 설계된 애플리케이션에서 중요하다.

데이터를 플래시 B에 저장하는 데 따른 추가적인 장점은 인터럽트 벡터 테이블이 플래시 A의 일부이므로 인터럽트를 비활성화할 필요가 없다는 점이다. 이는 모든 직렬 통신, A-D 변환, 타이머 등을 인터럽트가 활성화된 상태로 계속 실행할 수 있으며, 명령 실행 도중 코드를 점프시켜 인터럽트 벡터를 가져오고 인터럽트 서비스 루틴을 실행한 다음, 플래시 작업이 완료되었으며 새로운 작업이 시작되어야 하는지 검증할 수 있음을 의미한다.

또 다른 특징으로 전체 디자인의 원격 업그레이드를 수행하는



애플리케이션에 내결함성이 추가되었다. 한 플래시 बैं크에 새로운 프로젝트 버전을 저장하고 다른 बैं크를 백업으로 보존할 수 있다. 새로운 버전이 업로드 및 검증되면 이전 버전은 소거할 수 있다. 시스템 설계 차원에서 항상 플래시 A를 사용하여 플래시 B에 새로운 버전을 기록하거나 또는 그 반대의 경우도 가능하므로, 업데이트 도중 장애가 발생하더라도 작동 가능한 버전이 손실될 가능성이 없다.

EEPROM 에뮬레이션

플래시를 사용하여 비휘발성 정보를 저장하는 데 따른 한계 중 하나는 바이트에 쓰기 작업을 하려면 그 전에 해당 바이트가 소거 상태(모든 비트가 로직 '1'로 설정됨)여야 한다는 것이다. 이는 소거 작업이 섹터의 모든 비트를 '1'로 만들며, 쓰기 작업은 일부 또는 전체 비트를 '0'으로 변경함을 의미한다. 이에 따른 문제는 변수가 변경되어 비휘발성 백업을 실행해야 할 경우, 해당 바이트를 먼저 소거해야 하지만 플래시는 바이트 단위로 소거가 불가능하므로 전체 섹터를 소거해야 한다는 점이다.

EEPROM 에뮬레이션을 실행하는 루틴은 플래시를 사용하여 싱글 바이트 쓰기 및 소거와 관련한 EEPROM의 기능을 제공한다. 이를 구현하는 일반적인 방법은 플래시에 저장해야 하는 모든 변수가 포함된 구조를 생성하고, 이 구조에 해당 섹터가 활성 상태인지 아닌지를 나타내는 필드를 추가하는 것이다(예: 모든 데이터가 제대로 기록되었는지 검증하는 수단으로 기록되는 마지막 바이트). 플래시에서 일부 정보를 업데이트해야 하는 경우에는 항상 전체 구조가 복사된다. 이러한 비휘발성 업데이트는 일부 바이트가 변경될 때마다, 또는 애플리케이션 실행 기능의 일부로 계속 백업을 수행하는 타이머를 기준으로 실행될 수 있다.

애플리케이션의 유형에 따라 EEPROM 에뮬레이션에 필요한 플래시 양을 줄이기 위해 또는 시스템 내구성을 높이기 위해 몇 가지 변경을 적용할 수 있다. 예를 들어 하나의 플래시 섹터를 사용하는 경우, 섹터 용량을 초과하지 않는 범위 내에서 비휘발성 구조를 동일 섹터에 가능한 여러 번 기록하는 것이다(따라서 구조의 용량이 섹터 내 용량의 정확한 배수가 되도록 만드는 것이 권장되며, 일반

적으로 2배수의 용량이 사용된다). 플래시 섹터가 가득 차면 코드가 해당 섹터를 소거하고 다시 시작되어야 한다. 이 방식의 장점은 하나의 플래시 섹터를 사용한다는 점이지만, 섹터 소거 작업 중에 전원이 차단되면 모든 정보가 손실된다는 한계도 존재한다. 더불어 플래시 내구성도 2배로 소진된다.

다른 대안으로는 EEPROM 에뮬레이션에 2개의 섹터를 사용하는 방법이 있다. 먼저 새로운 섹터에 정보가 기록된 후에만 섹터가 소거되므로 플래시 내에 항상 유효한 정보 사본이 존재하게 되며, 소거 또는 쓰기 작업 도중에 절전 상태가 발생하더라도 정보가 손실되지 않지만, 비휘발성 정보를 저장하는 데 필요한 플래시의 양도 늘어난다.

어떤 방식을 사용해야 할 것인지는 애플리케이션 요구사항에 따라 결정된다.

사례 연구: 프리스케일 S08 제품군에서 플래시 쓰기/소거 작업을 실행하는 방법

S08 제품군에서 쓰기 또는 소거 작업을 실행하는 절차도 유사하다. 쓰기, 버스트 쓰기, 소거 또는 대량 소거 작업을 개별적으로 실행하려는 경우, 첫 번째 단계는 플래시 위치에 일부 데이터를 기록하는 것이다(차이점은 명령이 소거 또는 대량 소거인 경우 기록되는 데이터에는 변화가 없다는 점이다). 그 후에는 실행될 작업과 함께 레지스터 FCMD(플래시 명령)를 기록한 다음, 플래시 상태 레지스터에 비트를 기록하여 명령을 실행하고 실행된 플래시 명령의 결과로 오류가 발생하지 않았는지 확인하는 코드가 실행되어야 한다. 싱글 플래시 बैं크 구현에서 코드는 플래시 명령 완료 플래그가 설정되기를 기다려서 정상 코드 실행 상태로 복귀해야 하지만, 듀얼 बैं크 플래시의 경우 실행된 플래시 명령의 결과로 오류가 발생되지 않았음이 확인된 후에 즉시 복귀하여 다른 부분의 코드를 실행할 수 있다. 새 명령을 실행하기 전에 이전 명령이 이미 완료되었음을 항상 확인하는 코드를 적용하여 문제 발생 가능성을 없애야 한다.

아래 텍스트 블록은 마이크로 컨트롤러에 플래시 명령을 구현하는 방법을 보여주는 코드 예제이다.

```

#define FLASH_SIZE 0x00000001
#define FLASH_PAGE_SIZE 0x00000001
#define FLASH_PAGE_COUNT (FLASH_SIZE / FLASH_PAGE_SIZE)

void main(void)
{
    unsigned char FlashAddress = 0x0000;
    unsigned char FlashData = 0x00;
    unsigned char FlashCommand = 0x00;

    (1) Flash_Init();
    {
        Flash_Init();
    }

    (2) Flash_Write();
    {
        Flash_Write();
    }

    (3) Flash_Read();
    {
        Flash_Read();
    }
}

```

코드 1. 마이크로 컨트롤러에 플래시 명령을 구현하는 방법

듀얼 बैं크 플래시 솔루션

이 절은 메인 파일에서 플래시 B 섹션에 사용할 소거 및 싱글 바이트 쓰기 루틴을 호출하는 일반적인 구현 예를 보여준다. 두 작업이 유사하므로 매크로 정의를 통해 동일한 루틴을 두 가지 목적에 사용할 수 있다. 아래에 플래시 쓰기/소거 루틴의 구현 예가 제시되어 있다.

```

#define CODE_FLASH_SIZE 0x00000001
#define CODE_FLASH_PAGE_SIZE 0x00000001
#define CODE_FLASH_PAGE_COUNT (CODE_FLASH_SIZE / CODE_FLASH_PAGE_SIZE)

void main(void)
{
    unsigned char CodeAddress = 0x0000;
    unsigned char CodeData = 0x00;
    unsigned char CodeCommand = 0x00;

    (1) Code_Init();
    {
        Code_Init();
    }

    (2) Code_Write();
    {
        Code_Write();
    }

    (3) Code_Read();
    {
        Code_Read();
    }
}

```

코드 2. 플래시 쓰기/소거 루틴의 구현 예

모든 레지스터 및 비트 이름은 프리스케일 S08 마이크로 컨트롤러 제품군에 사용되는 것과 동일하다. €

대한민국 엔지니어들을 위한 No.1 커뮤니티 온라인 매거진

온라인 마케팅 시스템

귀사의 브랜드 이미지!
테크월드의 실시간 온라인 마케팅에서 찾으십시오!



월간 전자부품의 독자들은 전자소재, 재료부터 부품, 장비, 소프트웨어에 이르기까지 매우 폭넓게 분포하고 있습니다. 이에 월간 전자부품에서는 온라인을 통한 일대일 커뮤니케이션 실현을 위해 EDM 광고 및 뉴스레터 광고를 매월 실행하고 있으며, 실시간 배너광고를 통해 귀사의 광고 시너지 효과를 극대화 하기 위해 최선을 다하고 있습니다.

실시간 배너광고

EDM 메일링 시스템

뉴스레터 광고

디지털 시대의 미디어 파트너
테크월드
www.techworld.co.kr