- STM32F767 정복

- https://book.naver.com/bookdb/book_detail.nhn?bid=11647852

- 이해도 85%

목차

제4장 C언어 프로그래밍

- 윤덕용교수님 책은 워낙 유명해서 익히 알고 있었고, Firmware의 레지스터 제어 방식을 제대로 공부하기 위해서 보았다. 책의 앞부분은 STM32F767 MCU의 각각의 아키텍처와 그 안의 레지스터에 대해 교과서식 + 저자의 풀어서 쓴 설명 반반으로 전개된다. 뒷부분은 각각의 Peripheral 에 대한 보드 기반의 예제를 담고 있다. 나는 이 책에서 사용하는 보드도 직접 ohm사에서 구매해서 공부했고, 레지스터를 공부하는 데 어느 정도 성공했다고 생각한다.

그러나 최근의 Firmware 개발 트렌드는 레지스터를 직접 건드리기 보다는 ST 사의 MCU의 경우 CubeMX의 GUI에 의해 레지스터 셋팅을 굉장히 편리하게 하고, HAL라이브러리를 사용해서 응용 소프트웨어를 편하게 개발하고 있는 추세이다. 레지스터를 직접 제어해야 하느냐, 아니면 HAL라이브러리를 사용해야 하느냐에 대해서는 인터넷 상의 논쟁을 몇 번 보았지만 난 HAL 라이브러리를 사용하는 것이 맞다고 생각한다. 컴퓨터 아키텍처를 직접 설계하는 사람이라면 레지스터를 잘 이해하고 있기 때문에 레지스터 제어를 통해 firmware를 하는 것이 맞지만, 우리 같은 일반 엔지니어는 컴퓨터 아키텍처를 설계해 본 일도 없을 뿐더러 개발을 빠르게 하기 위해서는 이미 검증된 HAL 라이브러리를 사용하는 것이 훨씬 유리하기 때문이다.

만약 레지스터를 직접 제어한다면 목적에 맞게 32bit나 되는 레지스터 각각을 셋팅해야 하고, 함수까지 자기가 직접 만들면서 개발을 해야한다. MCU를 직접 만드는 사람이 아니고서야 자기가 만든 함수보다는 당연히 ST사에서 제공하는 HAL 라이브러리가 신뢰성, 완성도가 있을 수 밖에 없다.

그러므로 나는 이제 Firmware 개발은 코드 자동 생성툴을 사용하고, HAL 라이브러리를 사용해서 개발을 하는 것이 맞다고 생각한다. 그러나 주의할 점은 그것에만 의존해서는 안된다는 것이다. 라이브러리를 사용하다가 문제가 터졌을 때 라이브러리 내부를 분석하며 수정할 수도 있어야 하고, 라이브러리를 제대로 이해하는 것은 해당 MCU 아키텍처에 대한 공부가 되므로 Firmware 개발을 더 잘할 수 있는 기반이 된다고 생각한다.