

---

---

# ARM 펌웨어 개발환경 개요

Hancheol Cho

---

---

MCU 공부를 어떻게 해야 하나요?

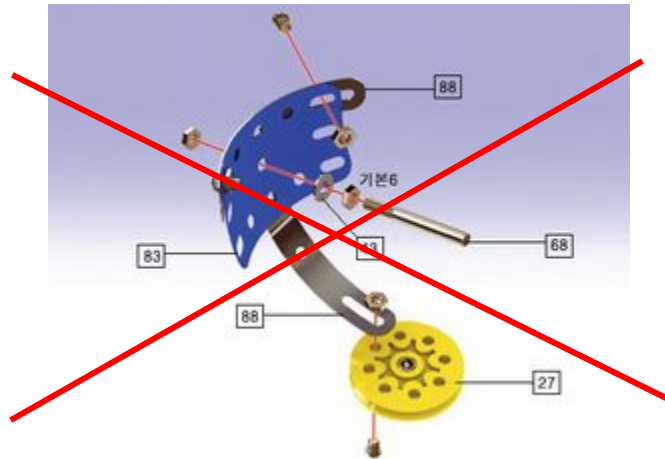


**Data Sheet**

**Reference Manual**

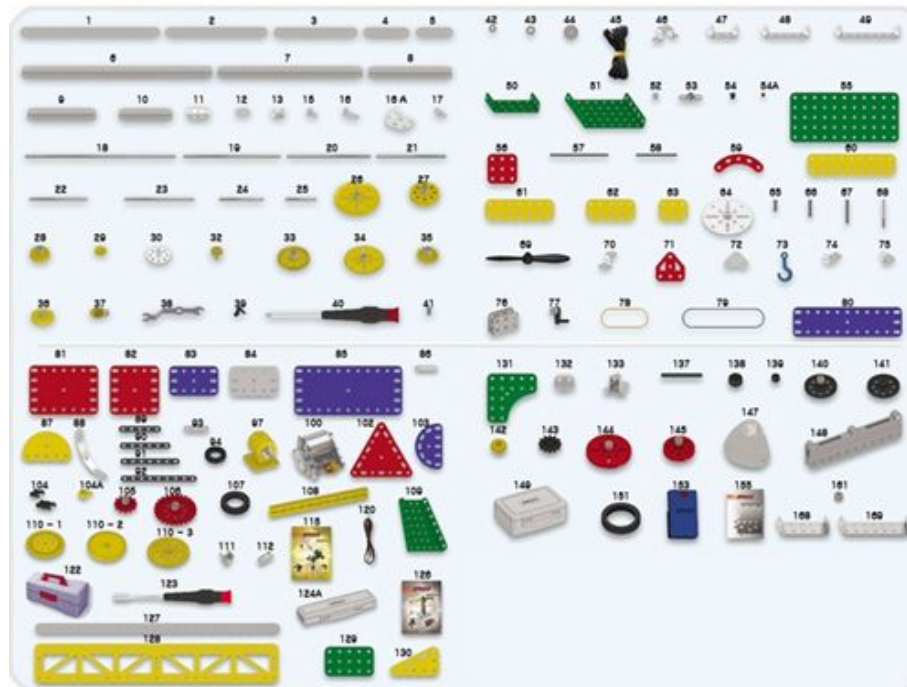
# Data Sheet가 어려운 이유

Data Sheet는 조립 메뉴얼이 아님



# Data Sheet가 어려운 이유

Data Sheet는 부품 사용법




# Data Sheet가 어려운 이유

조립 방법 및 활용은 스스로 만들어 가는 것




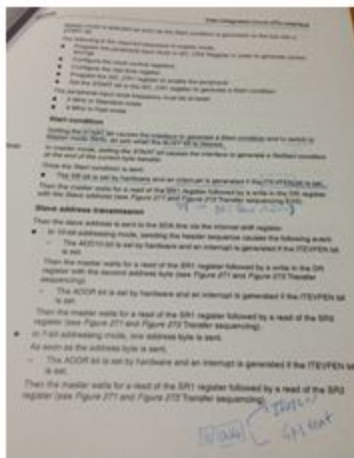
# Data Sheet Study

반복 학습하여 각 기능을 정확히 파악

 STM32F10xxx Cortex™-M3 programming manual.pdf

 STM32F103CB Reference Manual.pdf

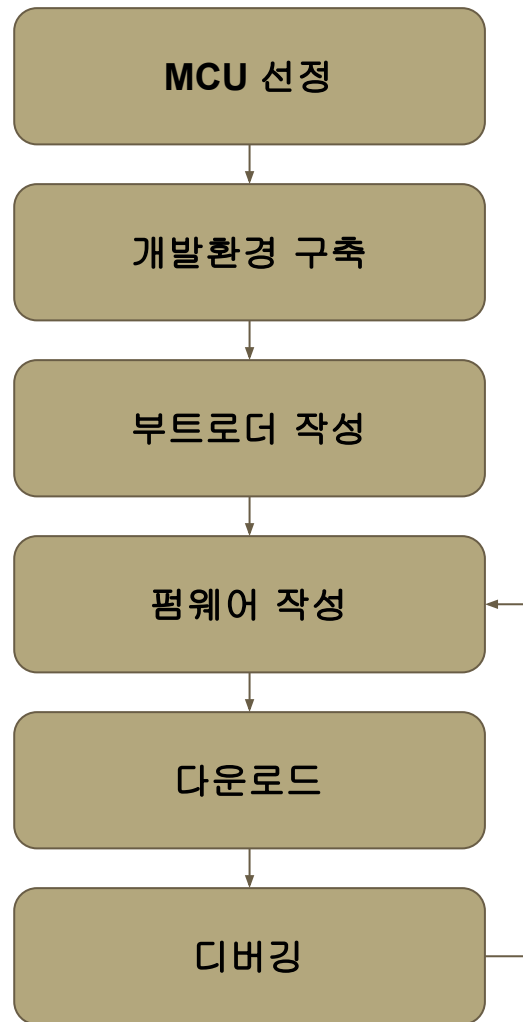
 STM32F103CB 데이터 쉬트.pdf



어떤 MCU로 공부 해야 하나요?



# 개발 과정

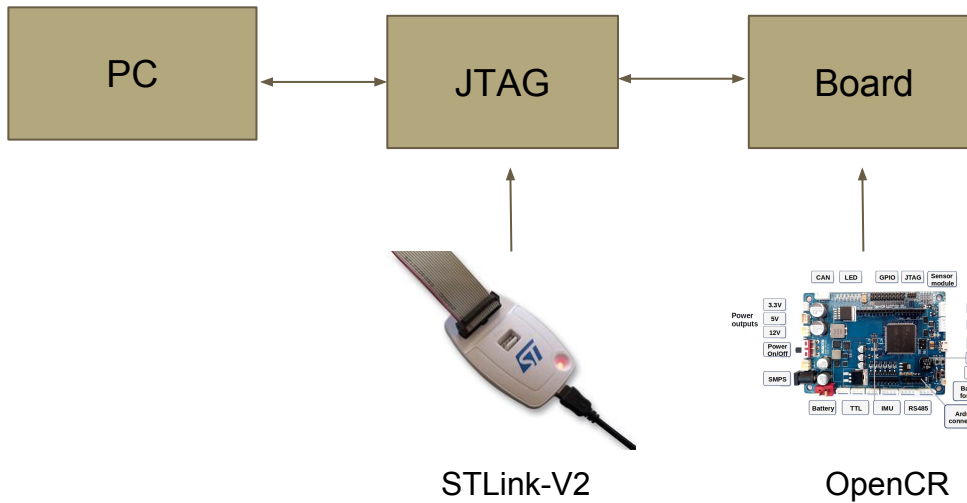




# 개발 환경 - Hardware

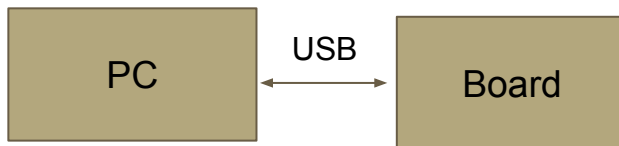
- JTAG

- JTAG을 이용한 실시간 디버깅 가능



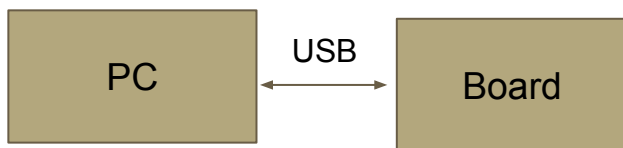
- DFU 모드

- MCU에 자체 내장된 DFU 부트로더를 이용하여 펌웨어 다운로드



# 개발 환경 - Hardware

- 부트로더
  - 직접 구현한 부트로더를 이용하여 펌웨어 다운로드



# 컴파일러 선정

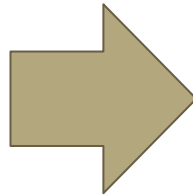
- 임베디드 프로그래밍을 위한 컴파일러 선택은 중요함

유료 컴파일러 vs 무료 컴파일러

# 컴파일러 선정

- 유료 컴파일러
  - 통합 개발 환경 제공으로 쉬운 사용
  - 문제 발생시 고객 지원 가능

비용 확보 가능



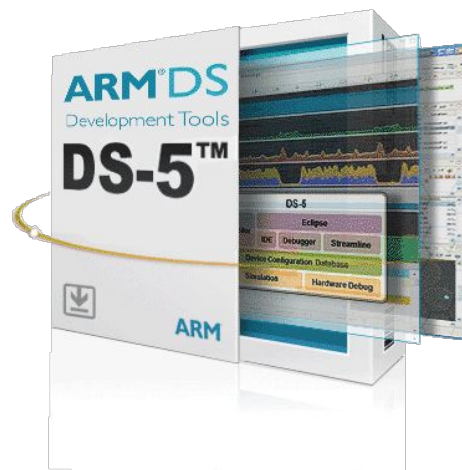
유료  
컴파일러

# 컴파일러 - 무료

- <http://www.openstm32.org/HomePage>
- <http://www.coocox.org/software/coide.php>
- <http://www.emide.org/>
- Eclipse + GCC
  - <https://eclipse.org/>
  - <https://launchpad.net/gcc-arm-embedded>

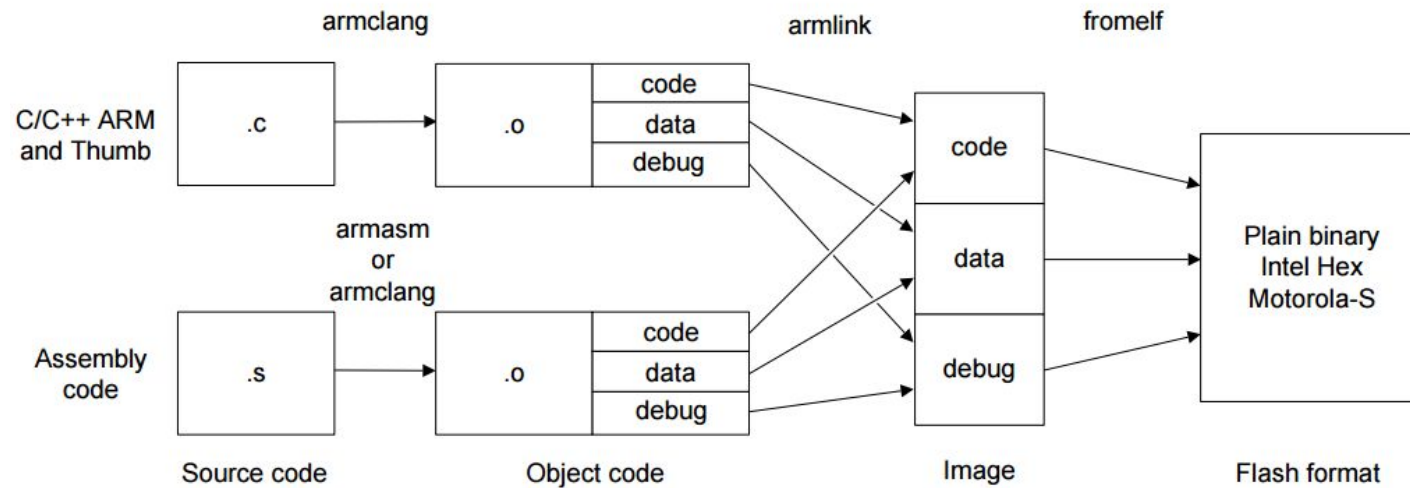


# 컴파일러 - 유료



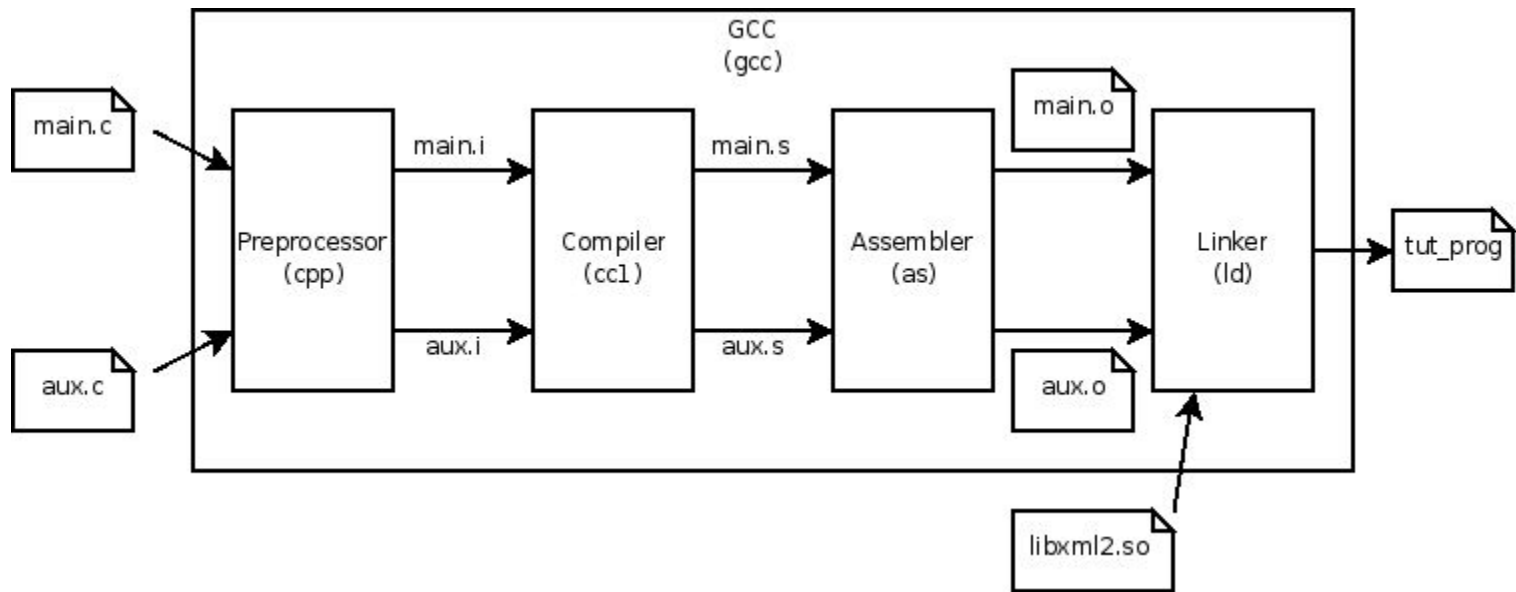
# 빌드 과정

- 소스 빌드 과정 - ARM 컴파일러



# 빌드 과정

- 소스 빌드 과정 - GCC 컴파일러

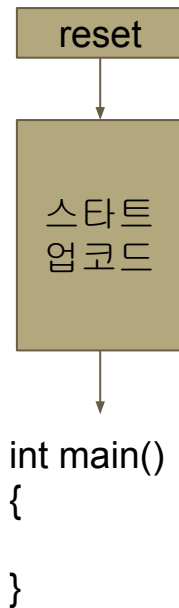




스타트업 코드 ?  
링커 스크립트 ?

# 스타트업 코드

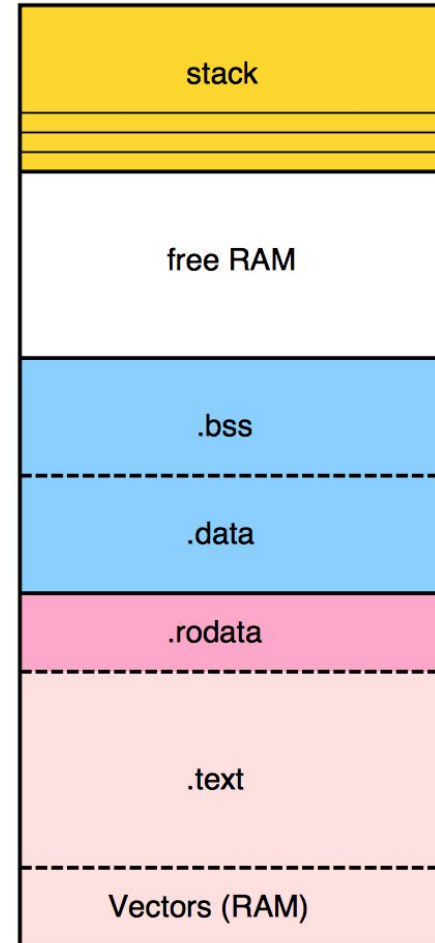
- Reset 이후에 main함수가 실행되기 전까지의 코드를 일반적으로 스타트업 코드라고 함
  - 어셈블리어로 대부분 작성되어 있으며 Cortex-M 시리즈는 C코드만으로도 작성 가능



# 메모리 섹션

## Sections:

- **.text**: Program code. Read only
- **.rodata**: constants (**const** modifier) and strings. Read only
- **.data**: Initialized global and static variables (startup value  $\neq 0$ )
- **.bss**: Uninitialized global and static variables (zero value on startup)



# 메모리 섹션

