



- EMBEDDED RECIPES (임베디드 레시피)
- https://book.naver.com/bookdb/book_detail.nhn?bid=6224668 (코너북에서 구매 가능)
- 이해도 70%

목차

1. Hardware 콜라주 (Collage) - 회로도 읽기

- 1 Hardware 콜라주
- 2 신호와 주파수 영역 - Spectrum Analysis
- 3 Analog 신호와 Digital 신호, 그리고 Ground
- 4 초간단 회로미론 R(저항), L(인덕터), C(캐패시터)
- 5 초간단 회로미론 응용 - 필터 (Filter)
- 6 Transistor 1%
- 7 Pull up, Pull down 그리고 Open Collector
- 8 RLC와 Transistor 感
- 9 논리회로의 확장
- 10 IC 기본
- 11 Register 년 누구냐
- 12 Clock 미란
- 13 Bus Transfer Mechanism
- 14 Timing 그리고 Spec 읽기
- 15 Memory의 선정과 XIP
- 16 RAM Memory의 물리적 동작
- 17 확장 to the CPU - How CPU works
- 18 CPU의 동작 예와 Pipe line

2. Microprocessor 아틀리에 (Atelier) - ARM을 파헤쳐 보자

- 1 Microprocessor 아틀리에
- 2 ARM inside
- 3 ARM mode와 THUMB mode
- 4 ARM 동작 Modes - 나는 어느 Mode를 써야 하나
- 5 ARM Register와 Context
- 6 ARM Exception과 Modes
- 7 ARM - Thumb PCS: Register 사용법
- 8 ARM은 Interrupt 냄새를 어떻게 맡는가
- 9 ARM SoC (System On Chip) - ARM 그렇고 말고
- 10 AMBA - SoC 안에서 IP끼리의 Bus 규격

3. Software 데쿠바쥬 (Decoupage) - Software의 정체와 만들기

- 1 Little Endian과 Big Endian
- 2 컴파일에 대한 단상
- 3 컴파일 공장 이야기
- 4 원하는 컴파일을 해보자
- 5 Preprocess (-E option)과 #include
- 6 Assembly로 만드는 방법
- 7 Library를 만들자 - 남한테 보여주기 싫어
- 8 Lib을 까보자
- 9 컴파일러 option들
- 10 변수의 scope와 그 생애 (Memory Map)
- 11 Memory Map과 Symbol 이야기
- 12 ELF format Object File의 진실
- 13 Linker를 마무리 짓자 - ELF와 fromelf까지!
- 14 Scatter Loading - Linker Description Script
- 15 MAP file 분석
- 16 Memory Map과 Linker의 만남 Locator
- 17 Makefile은 말하는 녀석일까~
- 18 컴파일을 더더더 쉽게. MACRO와 SUFFIX
- 19 조금 더 Make 테크닉들
- 20 Make option들

4. ARM 미장센 - ARM 제어의 구현

- 1 ARM Assembly를 파헤쳐 보자 ADS VS GNU
- 2 대충의 간단한 Assembly와 Reverse Engineering
- 3 ARM Thumb mode와 S 접미사
- 4 ARM과 Thumb 사이의 Veneer (베니아)
- 5 Inline Assembly와 INTLOCK() 구현
- 6 Pipeline과 Exception의 관계, 그리고 ^ 접미사
- 7 Exception Vector Table (EVT)과 각 Handler의 구현
- 8 SWI의 진실
- 9 Coprocessor Assembly
- 10 Bootloader와 Memory Budget (Mapfile) - 어떻게 변수를 초기화 할 것인가
- 11 Reset Handler에서 main (Entry point)까지
- 12 Scatter Loading과 Booup - __user_initial_stackheap

5. Software 비네팅 (Vinetting)

- 1 Context와 AAPCS
- 2 Pointer와 배열은 소녀시대와 원더걸스, 그리고 이중 포인터
- 3 struct와 typedef, 그리고 PACKED
- 4 Stack과 Heap에 관한 소고
- 5 Stack의 정제와 자세히 보기 - initialization 까지
- 6 함수가 불렀을 때 일어나는 일 - Stack 뒤지기 신포
- 7 Stack 동작의 비밀과 실제 메모리 덤프
- 8 Stack Size는 어떻게 잡는가
- 9 함수 포인터와 실행주소 변경
- 10 Linked List와 Queue

6. RTOS 팩토리 - Kernel 이야기

- 1 RTOS와 Kernel
- 2 Embedded Software는 무한루프
- 3 Task의 구조와 Signal
- 4 Task의 상태, Task는 Service단위와 ISR
- 5 Preemptive (선점형) Multitasking이란 도대체
- 6 Context Switching과 TCB - Task 상태변화
- 7 TCB - Task Control Block
- 8 Scheduler의 구현
- 9 ISR은 어떻게 구현해 - 선점형과 비선점형
- 10 선점형 Kernel에서 wait, send, clr signal의 구현과 IRQ Handler
- 11 Clock Tick ISR - Timer Service
- 12 ATOMIC - Critical Section, Mutex, Semaphore
- 13 Interrupt 와 Polling
- 14 Queue와 Inter Task Communication
- 15 DPC나 APC, 그리고 Bottom Half
- 16 Watchdog과 Sleep
- 17 Bootup중 Kernel로의 진입 - main() 함수 -
- 18 Kernel을 포팅한다는 것

7. Device Control and System Architecture

- 1 Device를 Control 한다는 것
- 2 Register를 Setting한다는 것과 Memory Mapped I/O
- 3 Memory Device를 Control 한다는 것
- 4 Bus Sizer Register와 Memory Controller - 메모리 확장구성과 CS와의 관계
- 5 특이한 Device Interface의 CS와 Address의 미용 - LCD
- 6 Shadow의 개념과 Bit operation
- 7 C의 조미료 MACRO 테크닉들
- 8 Synchronous와 Asynchronous는 머나
- 9 Wait State 이야기
- 10 PLL과 M/N:D
- 11 GPIO (Tristate Buffer)의 정제와 GPIO ISR
- 12 DMA(Direct Memory Access) - CPU몰래 영차 영차
- 13 Cache, Cash
- 14 MMU를 완전 베껴 보자 - Memory Management Unit
- 15 JTAG Interface를 Control 해 보자

8. 그래서 어쩌란 말이나 - How to debug

- 1 그래서 어쩌란 말이나 - ART of Debugging
- 2 Interrupt Lock에 의한 Watchdog Reset
- 3 Task Lock에 의한 Watchdog Reset
- 4 Data Abort가 발생했어
- 5 Stack Corruption과 Prefetch Abort - Stack 뒤지기 신포
- 6 Abort in ISR, IRQ mode에서의 Abort라...
- 7 0x0 go - 망연자실 Reset
- 8 Watchdog Reset in ISR
- 9 다양한 Exception과 Memory 불량 Symptom
- 9 Semihosting - Printf
- 10 JTAG과 ICD의 정제
- 11 Oscilloscope 사용법.
- Epilogue (에필로그)

- 임베디드 시스템에 관해 전반적인 개념을 잡을 수 있는 아주 좋은 책이다. 하드웨어 레벨부터 OS 레벨까지 전반적인 개념을 이야기하듯 서술해서 재미있게 공부했다. 대학 교과서처럼 딱딱하게 서술된 책보다는 이렇게 친절하게 서술된 책이 초보자에게는 공부하기에 더 적합하다고 생각한다.

물론 임베디드 시스템에 관한 모든 개념을 아주 자세하게 설명했다고 말하기에는 약간 부족하다고 생각합니다. 이 책 한권으로 만족하지 말고 다른 책도 같이 봐야 한다고 생각한다. 하지만 이정도 책이면 아주 훌륭한 책이라고 생각하면 소장하면서 여러 번 읽을 가치가 있다고 생각한다.

중간에 컴파일하는 과정이나 어셈블리가 나오면서 더 깊숙히 들어가는데 그런 부분은 잘 이해하지 못했다. ARM 어셈블리 부분이나 크로스 컴파일하는 과정은 다른 책을 통해 더 공부해야 할 것 같다.