Digtal Door Lock System

Team6
200611503 이승원
200912432 김다영
201211386 최하나
201013761 염　민

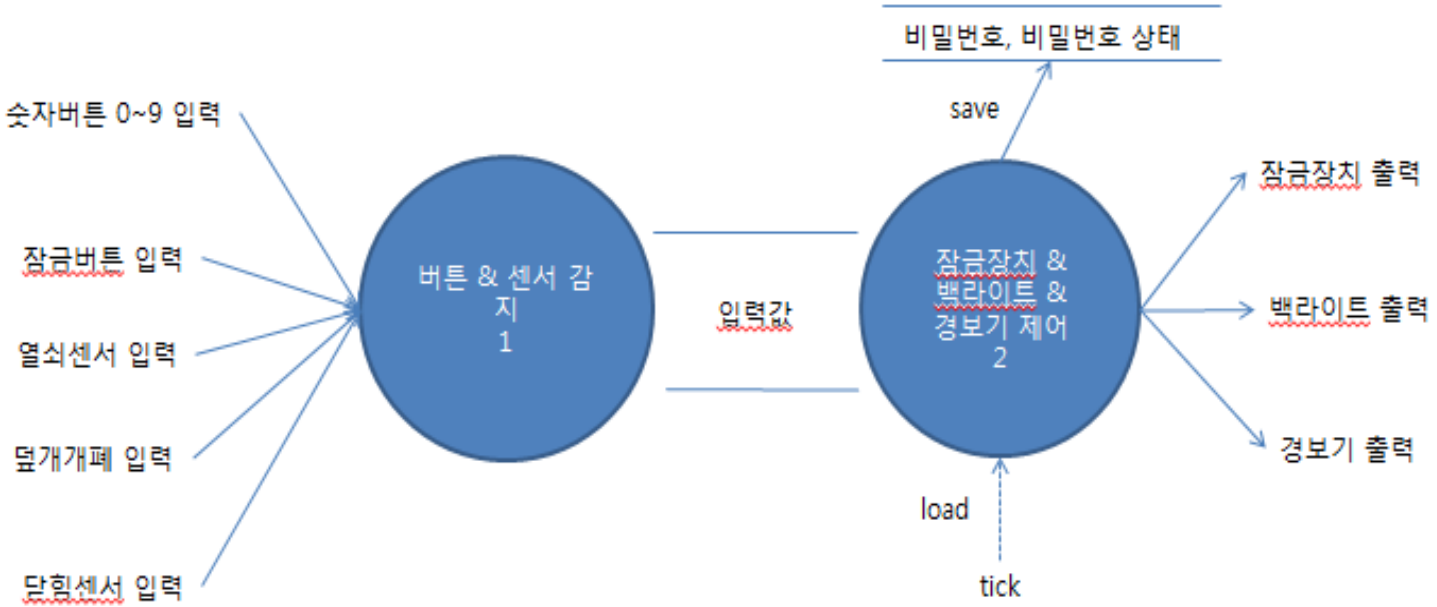
목차

- SRA
- SDS
- 구현
- 소프트웨어 공학 개론 수업을 들으며 느낀 점

- P.S 소프트웨어 설계 방법론..

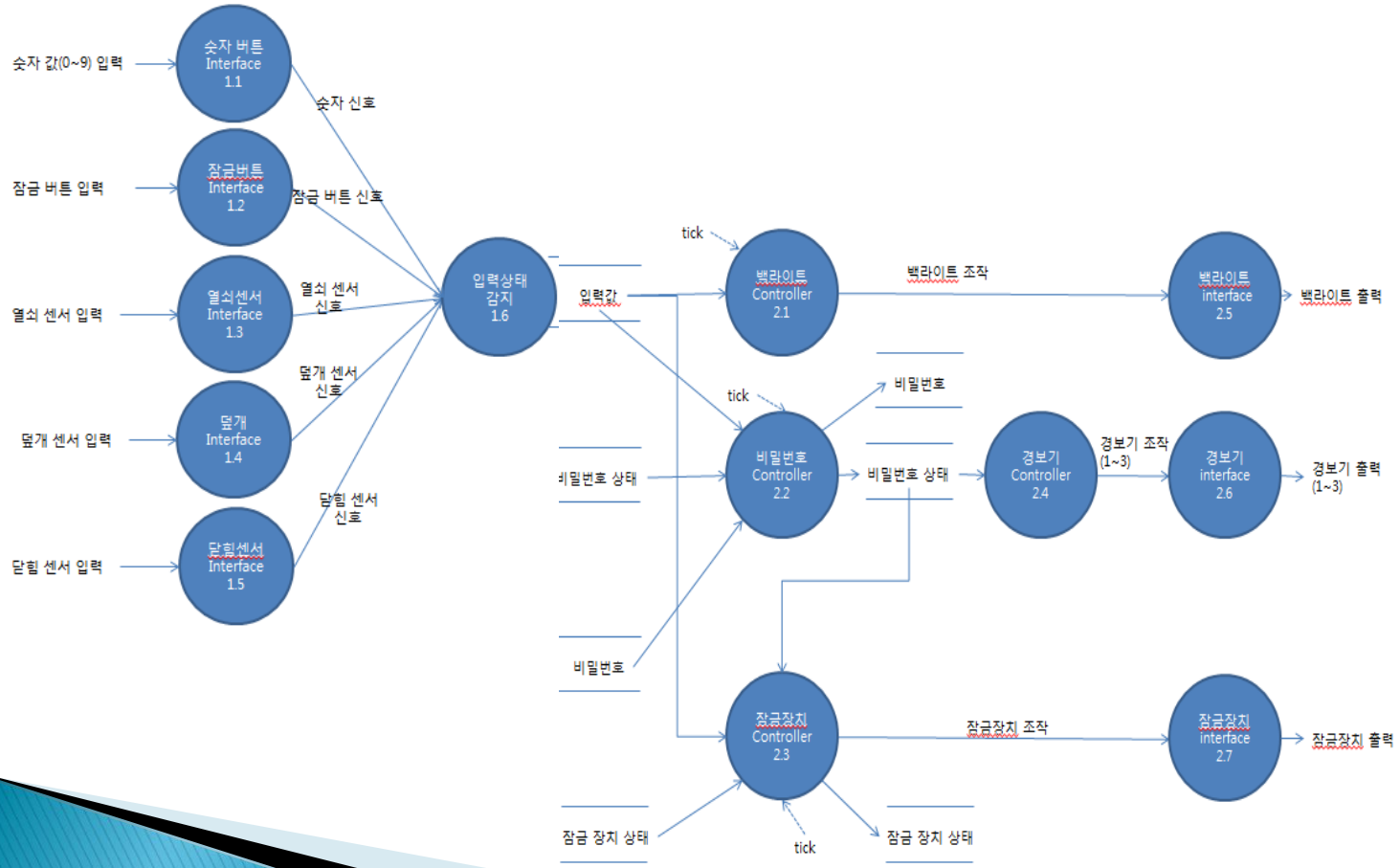
SRA

- DFD lv 1(ver.1.0)



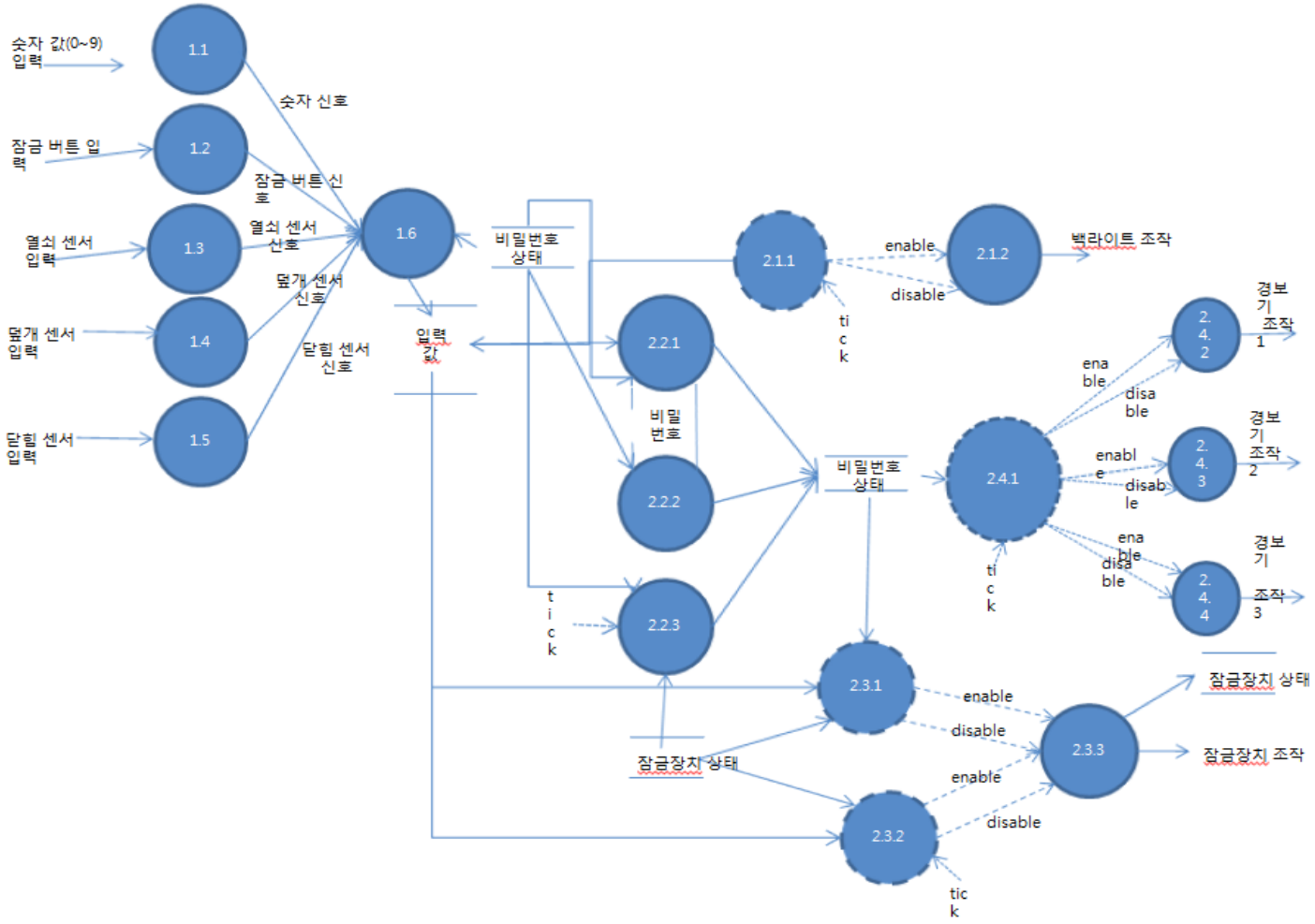
SRA

- DFD lv 2(ver3.0)



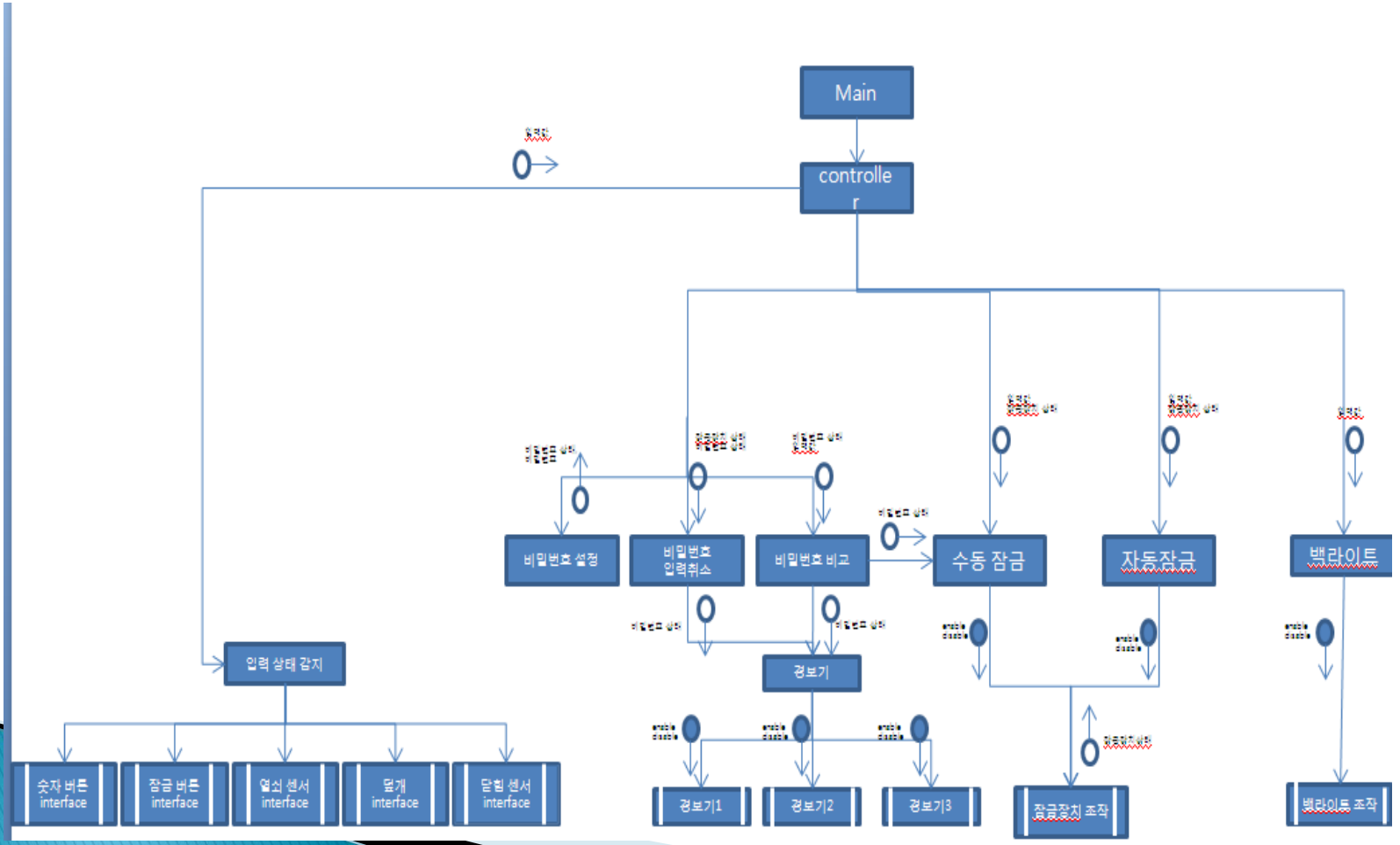
SRA

- DFD lv 3 (overall)



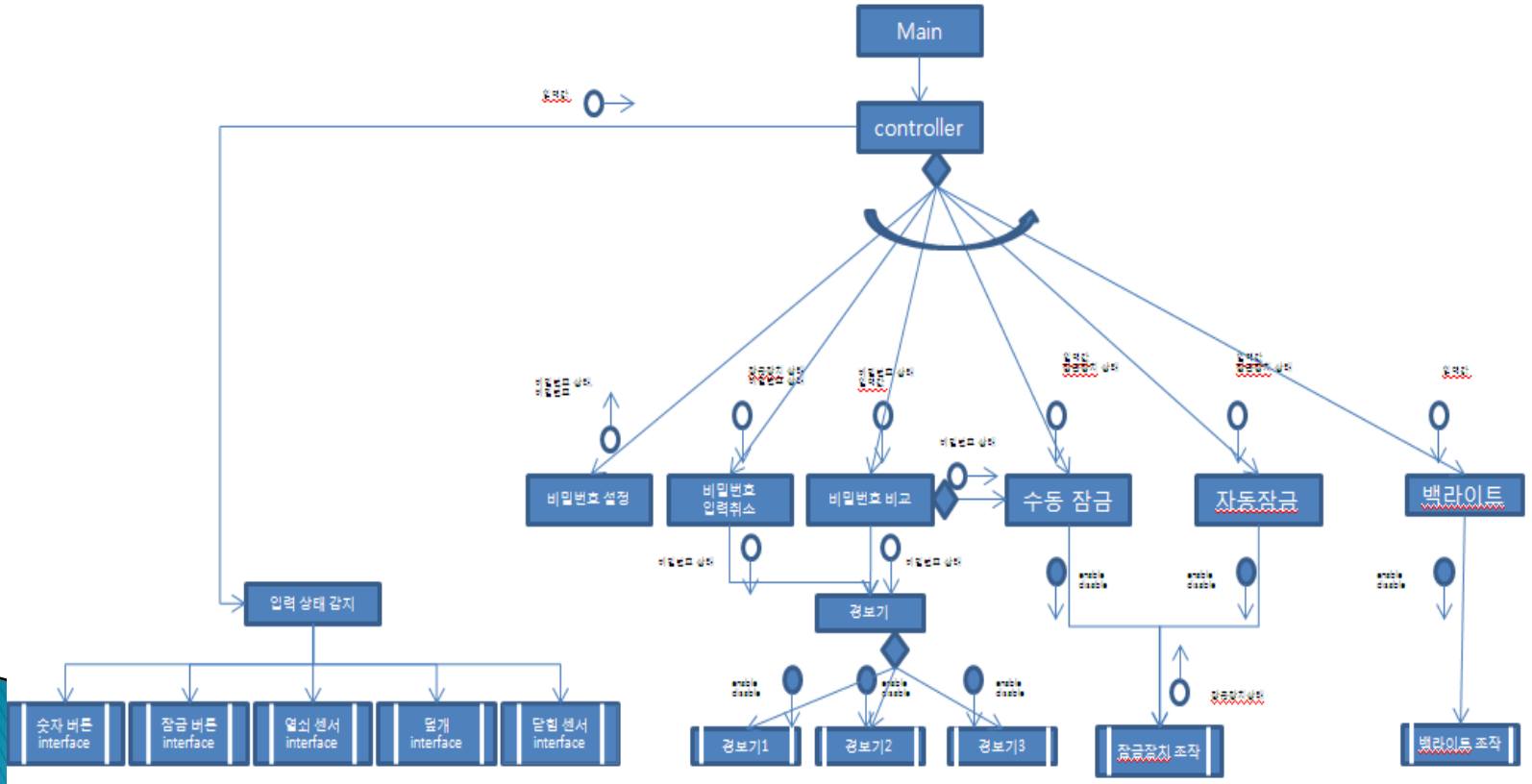
SD

- Basic


Main
controller
비밀번호 설정
비밀번호 입력취소
비밀번호 비교
수동 잠금
자동잠금
밸라이트
입력 상태 감지
경보기
숫자 버튼 interface
잠금 버튼 interface
열쇠 센서 interface
덮개 interface
닫힘 센서 interface
경보기1
경보기2
경보기3
잠금장치 조작
밸라이트 조작

SD

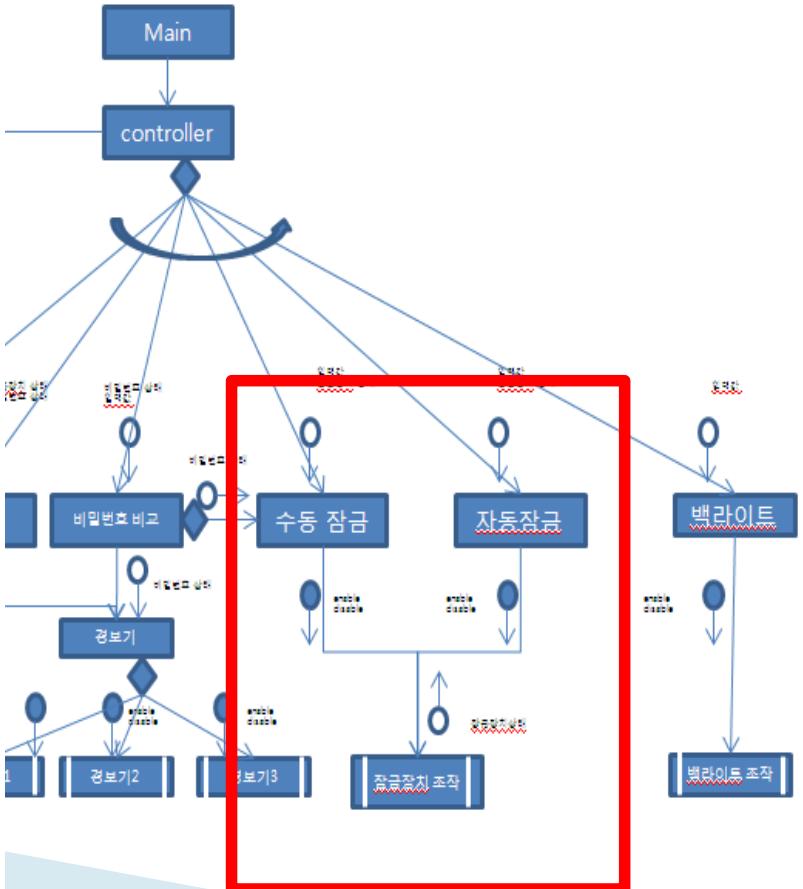
- Advanced


Main
controller
비밀번호 설정
비밀번호 입력취소
비밀번호 비교
수동 잠금
자동잠금
백라이트
입력 상태 감지
숫자 버튼 interface
잠금 버튼 interface
열쇠 센서 interface
덮개 interface
닫힘 센서 interface
경보기
경보기1
경보기2
경보기3
잠금장치 조작
백라이트 조작

```
void DDLSLockController(int inputValue)
{
    //printf("DDLSLock.c DDLSLockController()\n");
    switch(inputValue){
    case INPUT_KEY_SENSOR:
        handLock(-1);
        break;
    case INPUT_LOCK_BUTTON:
        handLock(inputValue);
        break;
    case INPUT_DOOR_SENSOR_CLOSE:
        if(password_state == STATE_PW_UNSET ||
            password_state == STATE_PW_SET_1 ||
            password_state == STATE_PW_SET_2 ||
            password_state == STATE_PW_SET_3 ){

        }
        else{
            if(lock_state == STATE_LOCK_OPEN){
                if(door_tick == -1){
                    door_tick = tick;
                }
            }
        }
        break;
    case INPUT_DOOR_SENSOR_OPEN:
        door_tick = -1;
        break;
    }
    if(password_state == STATE_PW_RIGHT){
        password_state = STATE_PW_SET;
        handLock(-1);
    }

    if(door_tick != -1){
        if(lock_state == STATE_LOCK_OPEN){
            if(tick - door_tick >= 3000){
                autoLock();
            }
        }
        else{
            door_tick = -1;
        }
    }
}
```



구현

- 화면 만드는 게 제일 어려웠음. 처음엔 번호키 디바이스를 콘솔에 그리려고 했으나 결국 콘솔에 printf로 찍게 되었다..

```
DDLSMain.c init()
키센서 = 평소           커버 = 열림
잠금버튼 = 평소         문 = 열림
num_0 = not Clicked     num_1 = not Clicked     num_2 = not Clicked
num_3 = not Clicked     num_4 = not Clicked     num_5 = not Clicked
num_6 = not Clicked     num_7 = not Clicked     num_8 = not Clicked
num_9 = not Clicked
STATE_PW_UNSET  STATE_LOCK_OPEN  비밀번호-1-1-1-1
 output
백라이트 = 켜짐 잠금장치 = 열림
alarm1 = off    alarm2 = off    alarm3 = off
--------------------------------------------------
```

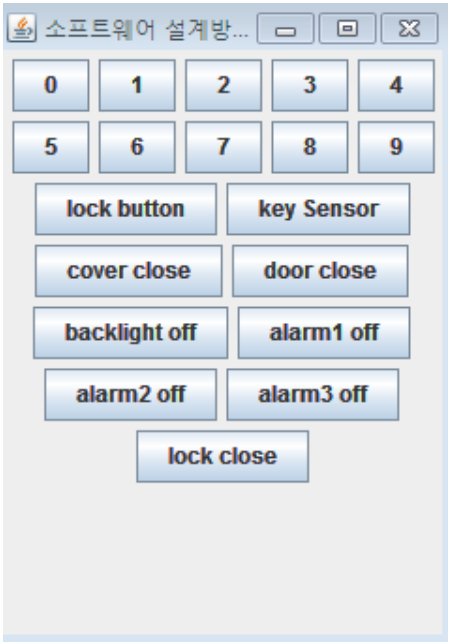
소프트웨어 공학 개론 수업을 들으며 느낀 점

- 승원 : 처음 설계한 내용 그대로 프로그래밍 하는 게 어려웠다. 하지만 무작정 프로그래밍 하는 것보다 더 효율적인 것 같다.

- 다영 : waterfall 모델은 처음이라 많이 헤맸다. 구현이 많이 아쉬웠다.

- 하나 : 개발에 설계가 중요하다는 것을 느꼈다.

P.S 소프트웨어 설계 방법론

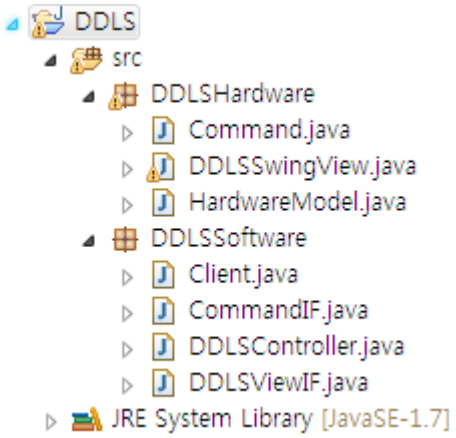


- 결국 Java로 한 번 더 구현..

- 두 개발 과정 비교
  - 기능 구현에는 비슷한 소스를 사용
  - 상속 관계 정의, interface를 이용하여 포함 관계를 설정하는 게 기능 구현보다 더 어려웠다…

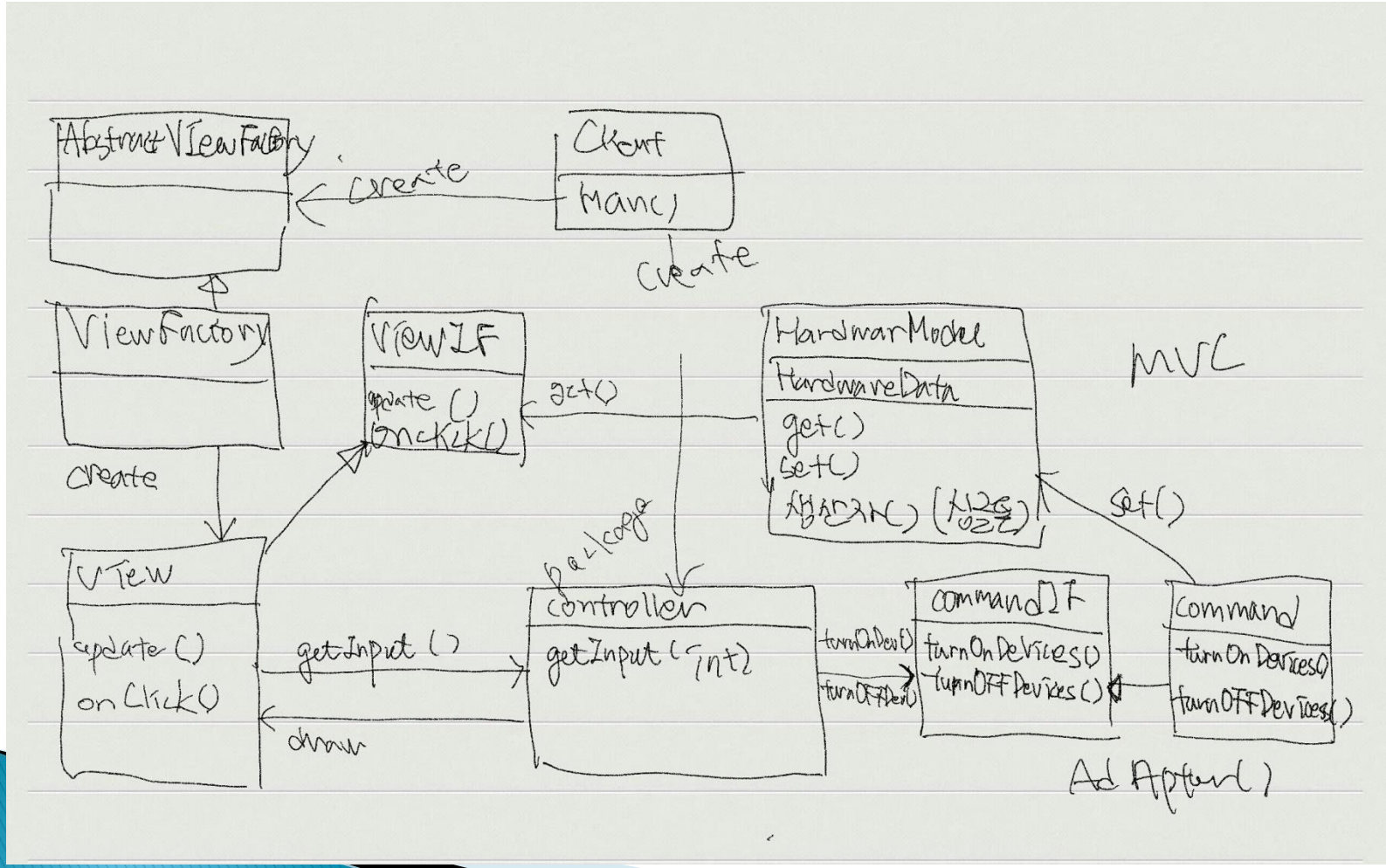
P.S 소프트웨어 설계 방법론

- 리스너 패턴을 사용, view에서 input이 들어오는 부분을 interface로 하여 controller에서 구현을 완성하도록 하였다.
- 그래서 Hardware파트는 (데이터를 저장하는 모델 빼고) 디바이스에 따라 완전이 바뀌더라도, Software파트는 변하지 않도록 하였다.



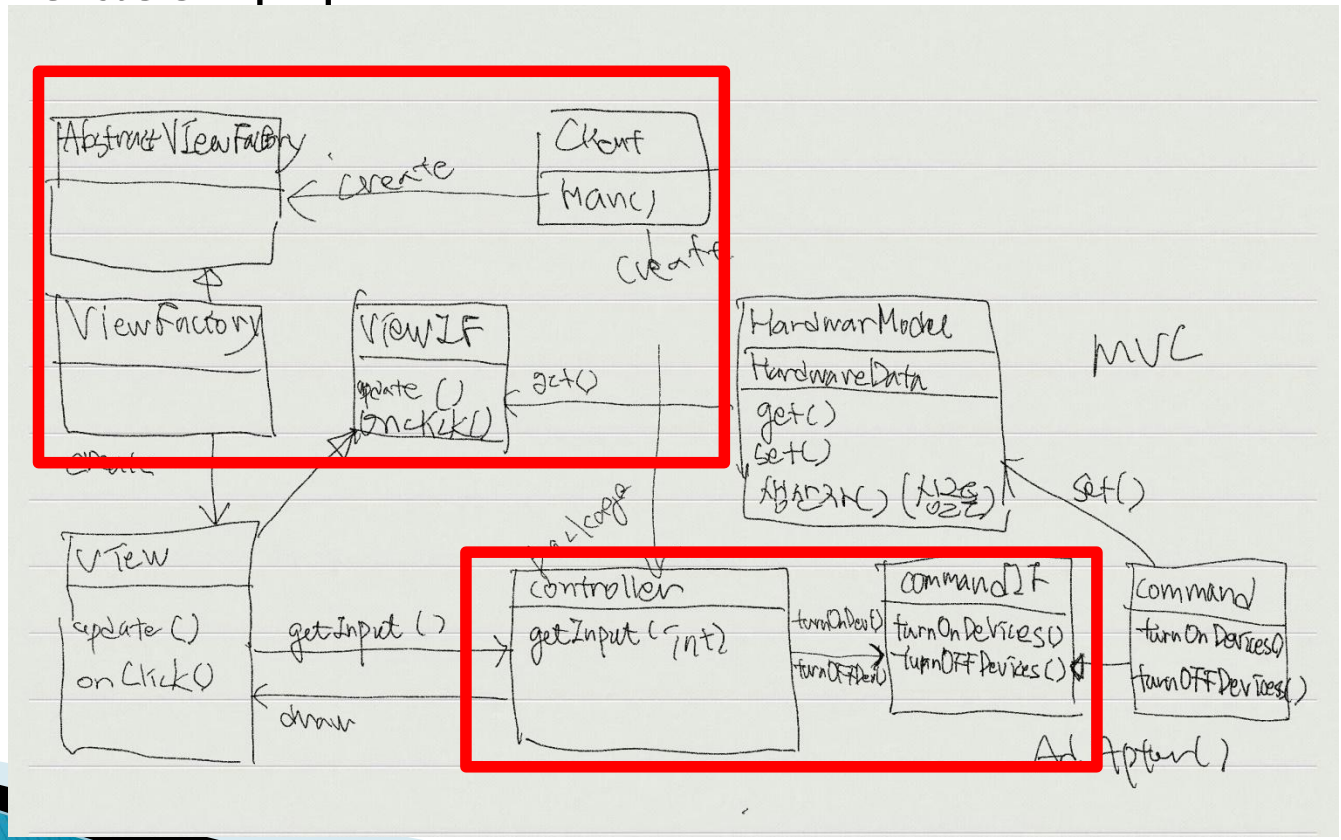
P.S 소프트웨어 설계 방법론

- 개선된 설계도



P.S 소프트웨어 설계 방법론

- 빨간 선 안쪽이 Software part. Hardware Part는 대부분 Software Part의 클래스를 상속받은 concrete class이다.



Q&A