

# DO-178B 문서 템플릿

---

- 목표 :

- C 언어로 시시:분분:초초를 표시해주는 프로그램.
- 시는 24 시간으로 표시,
- Ctrl+C 를 입력받기 전까지는 1 초에 1 번 화면에  
시시:분분:초초 를 보여주고 다음줄로 넘어가는 프로그램

## DO-178B 문서 리스트

1. PSAC (Plan for Software Aspects of Certification) | 인증 계획 문서
2. SDP (Software Development Plan) : 개발 프로세스 설명
3. SVP (Software Verification Plan) : 테스트 계획 및 절차
4. SCMP (Software Configuration Management Plan) : 형상관리 계획
5. SQAP (Software Quality Assurance Plan) : 품질보증 계획
6. SRS (Software Requirements Specification) : 요구사항 정의서
7. SDD (Software Design Description) : 설계 명세서
8. SCI (Software Code & Integration) : 소스 코드 및 통합 설명
9. STD (Software Test Description) : 테스트 케이스 설명
10. STR (Software Test Report) : 테스트 결과 보고서

## 1. PSAC (Plan for Software Aspects of Certification) | 인증 계획 문서

- 프로젝트 명: Real-Time Clock Display
- 소프트웨어 목적: 실시간으로 24 시간 시계 표시
- Level: DO-178B Level D (교육용 예제)
- 소프트웨어 환경: Windows, GCC, C 언어
- 대상 플랫폼: 일반 Windows PC
- 인증 활동:
  - 요구사항 명세서 작성
  - 설계 문서 작성
  - 단위 테스트 수행

## 2. SDP (Software Development Plan) : 개발 프로세스 설명

- 개발 표준: ANSI C
- 코딩 규칙: MISRA-C 일부 적용
- 형상 관리: Git
- 산출물:
  - 요구사항 명세서
  - 설계 문서
  - 테스트 케이스 및 리포트

### **3. SVP (Software Verification Plan) : 테스트 계획 및 절차**

- 목표: 모든 요구사항이 코드에 정확히 구현되었는지 확인
- 테스트 종류:
  - 단위 테스트
  - 요구사항 기반 기능 테스트
- 자동화: 없음 (수동 테스트)
- Pass/Fail 기준: 출력이 요구사항에 맞는지 확인

#### 4. SCMP (Software Configuration Management Plan) : 형상관리 계획

- 형상 항목:
  - Source Code (clock.c)
  - 요구사항 명세서
  - 설계 문서
  - 테스트 케이스 및 결과
- 버전 관리: Git 사용

## 5. SQAP (Software Quality Assurance Plan) : 품질보증 계획

- QA 담당자: 교육용 실습 진행자
- 활동:
  - 코드 리뷰 (학생 상호 검토)
  - 요구사항 추적성 확인
  - 테스트 케이스 적용 여부 검토

## 6. SRS (Software Requirements Specification) : 요구사항 정의서

- REQ-01: 프로그램은 현재 시:분:초를 24 시간 기준으로 출력해야 한다.
- REQ-02: 1 초마다 새로운 시간을 한 줄씩 출력해야 한다.
- REQ-03: 사용자가 Ctrl+C 를 누르면 프로그램은 종료되어야 한다.

## 7. SDD (Software Design Description) : 설계 명세서

- 주요 모듈:
  - main(): 루프 제어 및 시간 출력
  - handle\_sigint(): SIGINT 처리
- 라이브러리 사용:
  - time.h: 시간 정보 얻기
  - signal.h: Ctrl+C 처리
  - unistd.h: sleep()

## 8. SCI (Software Code & Integration) : 소스 코드 및 통합 설명

- 코드 저장소: src/clock.c
- 컴파일 방법: gcc -o clock clock.c
- 실행 방법: ./clock

## 9. STD (Software Test Description) : 테스트 케이스 설명

Test ID	Description	Input	Expected Output
TC-01	시간 출력	실행 후 3 초 대기	3 줄의 시간 출력
TC-02	Ctrl+C 종료	실행 후 Ctrl+C	"프로그램 종료." 출력

## 10.STR (Software Test Report) : 테스트 결과 보고서

Test ID	Result	Comments
TC-01	Pass	시:분:초 출력 정상
TC-02	Pass	Ctrl+C 시 정상 종료 확인

## DO178\_ClockProject/

```
|   └── doc/  
|       └── DO178_ClockProject.doc  
└── src/  
    └── clock.c  
└── test/  
    └── test_result.txt
```

[c언어 > verilog로 변경해서 작성해보기](#)

```
#include <stdio.h>  
#include <time.h>  
#include <unistd.h>  
#include <signal.h>  
#include <stdlib.h>  
  
volatile sig_atomic_t stop = 0;  
  
void handle_sigint(int sig) {  
    stop = 1;  
}  
  
int main() {  
    signal(SIGINT, handle_sigint); // Ctrl+C 핸들링  
  
    while (!stop) {  
        time_t raw_time;  
        struct tm *time_info;  
  
        time(&raw_time);  
        time_info = localtime(&raw_time);  
  
        printf("%02d:%02d:%02d\n",
               time_info->tm_hour,  
               time_info->tm_min,  
               time_info->tm_sec);  
  
        sleep(1);  
    }  
  
    printf("프로그램 종료.\n");  
    return 0;  
}
```