모빌리티 SW 및 임베디드 프로그래밍 40일 학습 계획

목표: 제공된 커리큘럼을 완벽히 이해하여 차량 SW 개발 면접 질문에 자신 있게 대답할 수 있도록 준비.

1주차: 기초 학습 (SW 기본, C/C++, 차량 SW 개요)

1일차:

- **SW 기본:** 미래 모빌리티 트렌드 개요 (24시간 과정).
- **C/C++:** 프로그래밍 기초: 변수, 데이터 타입, 연산자 (80시간 중점 학습).
- 학습 자료: 온라인 튜토리얼, 샘플 코드, 연습 문제.

2일차:

- **C/C++:** 제어 구조 (if, switch, 루프).
- **차량 시스템:** 차량 아키텍처 기초 (차량 및 SW 시스템 이해).
- 연습 문제: 간단한 제어 구조 프로그램 작성.

3일차:

- **C/C++:** 함수, 배열, 포인터.
- **차량 SW 공학:** 차량-소프트웨어 통합 개요.
- 연습: 함수 구현 및 포인터 사용.

4일차:

- **C/C++:** 고급 포인터, 메모리 할당, 구조체.
- **차량 SW 공학:** 차량 소프트웨어 모듈 분석.
- 연습: 동적 메모리 활용 연습.

5일차:

- **모던 C/C++:** 객체지향 프로그래밍 개념 (32시간). 클래스 및 객체 소개.
- **차량 변속 시스템:** 차량 변속 시스템에 대한 학습.

- 연습: 기본 클래스 구현.

6일차:

- **모던 C/C++:** 다형성, 상속, 캡슐화.
- **차량 변속 시스템:** 고급 변속 시스템 학습.
- 연습: 객체지향 개념을 활용한 프로젝트 구현.

7일차:

- **복습:** 프로그래밍 및 SW 기본 개념 연습.
- 모의 테스트: 코드 작성 및 SW 트렌드 설명.

2주차: 차량 인터넷 시스템 및 임베디드 기본

8일차:

- **차량 인터넷 시스템:** 통신 프로토콜 이해 (40시간).
- **임베디드 시스템:** 임베디드 소프트웨어 기초 (24시간).
- 연습: CAN 및 LIN 통신 조사.

9일차:

- **임베디드 시스템:** 간단한 임베디드 코드 작성.
- 연습: 기본 마이크로컨트롤러를 사용한 LED 깜박임 구현.

10일차:

- **차량 SW 프로세스:** 차량 소프트웨어 라이프사이클 분석 (40시간).
- **C 프로그래밍:** 임베디드 시스템에 맞춘 기초.
- 연습: 소규모 임베디드 함수 작성.

11일차:

- **차량 SW 프로세스:** 디버깅 및 테스트.

- 연습: 샘플 임베디드 코드 디버깅.

12일차:

- **SW 요구사항 분석:** 요구사항 분석 도구 학습 (40시간).
- 연습: 요구사항 분해 사례 학습.

13일차:

- **차량 보안:** 차량 사이버 보안 학습 (40시간).
- **C 프로그래밍:** 안전한 프로그래밍 기초.
- 연습: 기본 인증 함수 구현.

14일차:

- **복습:** 차량 인터넷 시스템 및 임베디드 기본 복습.
- 모의 테스트: 임베디드 코드 디버깅 시뮬레이션.

3주차: 중급 임베디드 프로그래밍

15일차:

- **임베디드 SW 개발:** 임베디드 프로그래밍 기초 (40시간).
- 연습: 간단한 임베디드 루프 개발.

16일차:

- **임베디드 SW 개발:** 기본 하드웨어 인터페이스.
- 연습: GPIO를 사용한 입출력 작업.

17일차:

- **임베디드 MCU 프로그래밍:** MCU 프로그래밍 기초 (40시간).
- 연습: STM32 보드 설정 및 간단한 프로그램 작성.

18일차:

- **임베디드 MCU 프로그래밍:** 타이머 및 인터럽트.
- 연습: 타이머 기반 LED 깜박임 구현.

19일차:

- **임베디드 리눅스:** 리눅스 시스템 기초 (32시간).
- 연습: Raspberry Pi에 리눅스 OS 설치.

20일차:

- **임베디드 리눅스:** 자동화를 위한 리눅스 스크립트 개발.
- 연습: 작업용 셸 스크립트 작성.

21일차:

- **복습:** 임베디드 개발 개념 강화.
- 모의 테스트: MCU 프로그램 개발 및 디버깅.

4주차: 고급 임베디드 프로그래밍 및 실시간 시스템

22일차:

- **실시간 OS:** 실시간 시스템 기초 (32시간).
- 연습: 기본 RTOS 작업 스케줄러 구현.

23일차:

- **실시간 OS:** 고급 실시간 시스템 개발.
- 연습: 멀티 태스킹 함수 생성.

24일차:

- **AUTOSAR 기본:** AUTOSAR 표준 소개 (32시간).
- 연습: 간단한 AUTOSAR 기반 함수 개발.

25일차:

- **AUTOSAR 기본:** 인터페이스 및 모듈 구현.
- 연습: AUTOSAR 통합 모의.

26일차:

- **소프트웨어 테스트:** 테스트 프레임워크 기초 학습 (32시간).
- 연습: 샘플 프로그램 오류 테스트.

27일차:

- **OTA 시스템:** OTA 프로그래밍 학습 (40시간).
- 연습: 간단한 OTA 업데이트 시뮬레이션.

28일차:

- **복습:** AUTOSAR, RTOS 및 테스트 개념 복습.
- 모의 테스트: 실시간 및 AUTOSAR 통합 시뮬레이션.

5주차: 최종 프로젝트 및 면접 준비

29일차:

- **프로젝트 시작:** 차량 개발용 임베디드 SW.
- 연습: 프로젝트 목표 설정 및 아키텍처 정의.

30일차:

- **프로젝트 구현:** 소프트웨어 모듈 개발.
- 연습: 초기 프로젝트 문제 디버깅.

31일차:

- **프로젝트 구현:** 하드웨어와 소프트웨어 통합.
- 연습: 모듈 간 호환성 보장.

32일차:

- **프로젝트 완료:** 프로젝트 테스트 및 문서화.
- 연습: 프로젝트 산출물 개선.

33일차:

- **면접 준비:** 차량 SW 개념 복습.
- 연습: AUTOSAR, RTOS 및 차량 보안 관련 모의 질문.

34일차:

- **면접 준비:** 코딩 면접 질문.
- 연습: 알고리즘 기반 질문 풀이.

35일차:

- **면접 준비:** 시스템 설계 질문.
- 연습: 차량 SW 설계 설명.

36일차:

- **면접 준비:** 디버깅 면접 질문.
- 연습: 샘플 코드 효율적으로 디버깅.

37일차:

- **복습:** 주요 개념 및 약점 보강.
- 모의 테스트: 전체 면접 시뮬레이션.

38일차:

- **복습:** 고급 개념 및 Q&A.
- 모의 테스트: 최종 코딩 도전 과제 풀이.

39일차:

- **최종 모의 면접:** 종합 평가.
- 피드백: 개선 사항 식별.

40일차:

- **최종 복습:** 학습 내용 되돌아보기.
- 성공 축하 및 실전 면접 준비.

이 40일 계획은 커리큘럼의 핵심 주제를 포괄적으로 다루도록 설계되었습니다. 각 주는 이론 학습과 실습을 병행하여 이해를 강화합니다. 특정 주제에 대한 자료나 추가 연습 문제가 필요하면 말씀해주세요!