**8주차 업무 일지**

**결론**

라인트레이서 부착 및 구동 테스트 진행 완료

라인트레이서 기반 자율주행 알고리즘 코드 수정 및 테스트 진행 중

**서지완(팀원, 태스크 정의 파일 분리, 라인트레이서 부착 및 테스트, MCU 코드 리팩토링, 자율주행 알고리즘 구현 및 테스트 진행)**

**내용**

**1. 태스크 정의 파일 분리**

i) task\_manage.c/h 파일을 생성하여 freertos.c에 있던 tasks  
 정의를task\_manage.c/h에 분리하여 유지보수하기 편리하게 수정

ii) freertos.c에서는 tasks를 생성하여 실행만 하도록 구현

**2. RC카에 라인트레이서 부착 및 테스트**

i) RC카에 전면부 하단에 라인트레이서를 부착할 구멍을 뚫고, 지지대로 라인트레이서를 고정

전기 배선, 전자 공학, 전자제품, 회로 구성요소이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

ii) 라인트레이서를 부착 후, lineTracer.c/h 안에 라인트레이서 읽기 함수를 구현하여 테스트를 진행



**3. MCU 코드 리팩토링**

i) mpu6050.c

- 배열 기반 구조화

- ReadSensorData() 함수를 도입하여, 가속도/자이로 데이터 모두 처리

ii) bluetooth.c

* SendMotorCommand() 함수를 도입하여, 중복되는 UART 디버깅 출력 정리

**4. 자율주행 알고리즘 구현**

i) automotive.c

- LineTracerDriveDecision() 함수를 도입하여, 라인 감지 방향에 따라 Motor\_Forward, Motor\_Left, Motor\_Right 호출하도록 구현

- 복구 로직 :

(1) 1초 이내: 좌/우 회전만 시도

(2) 1~3초: 후진 후 좌/우 회전

(3) 3초 초과: 정지

ii) task\_manage.c

* StartLineTracerTask() 태스크 추가 : 라인트레이서 상태를 읽고 방향을 결정하도록 하였음

오토마톤, 전기 배선, 기계이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**계획**

* 자율주행 알고리즘 디버깅 중 라인 인식 후 실시간으로 모터 제어를 할 수 있도록 수정할 예정