실증적SW개발프로젝트 주간보고 (6주차)

작성일: 2025/04/13 팀명: Wiper

|  |  |  |
| --- | --- | --- |
| 팀 활동  보고 | 활동일시 | 2025.04.06 ~2025.04.13 |
| 장 소 | 동아대학교 Makespace 창업실 5번방 |
| 참석자 | 박준현, 서지완, 손주석, 이준영, 최창욱 |
| 특이사항 | 없음 |
| 이번주  진행사항 | **1. 개발내용**  **센서**   * RADAR 테스트 * IMU 센서 값(Yaw/Pitch/Roll) 계산 코드 작성 * MCU 보드 초음파 센서 융합   **RC카**   * RC카 설계도 작성 및 프레임 제작   **이미지 디헤이징**   * AOD-net 실시간 영상에 적용 및 yolov5 결합 * 이미지 디헤이징 논문 분석   **MCU**   * FreeRTOS 기반 센서 태스크 멀티화 * UART 출력 전용 태스크 구조 도입 * FreeRTOS 기반 MCU 모듈화 및 Task 구조 설계   **코드 및 기능 분리 작업**  **2. 팀원별 활동내용**  **손주석(RADAR 테스트, RC카 설계)**  **1. RADAR 테스트**  - RADAR SDK 설치 및 실행  - Jetson에서 SDK 실행을 위한 SDK 분석 및 코드 수정  - CPU 차이 및 보드의 PinMAP 차이를 해결하기 위한 코드 수정을 하였으나 라이브러리 종속성 문제로 해결 실패  - Raspberry pi 3에서 SDK 실행  - SDK에서 지원되지 않는 Library 확인 및 디버깅 진행, Raspberry pi 3에서 SDK 실행 완료  - Raspberry pi 3용 SDK 내용 분석 및 전송에 필요한 데이터를 전처리 및 SPI 통신을 활성화 하여 Jetson에 전송  - RADAR의 감지 거리 및 정확성에 대한 문제 제기로 인해 초음파 센서를 이용하는 것으로 방향 수정 논의  **2. RC카 설계**  - 효율적인 부품 배치를 위한 RC카 디자인 설계  텍스트, 사무용품, 실내, 펜이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.  - 윗면  텍스트, 사무용품, 친필, 펜이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.  - 아랫면  **3. STM32 IDE 공부**  - RC카 제어 알고리즘 구현을 위해 IDE 사용법에 대한 공부 진행  - 기존 테스트 코드 학습 및 실습 진행  - FreeRTOS에서 task 핸들러를 등록해 다루는 방법 및 Main에서 FreeRTOS를 사용하는 법에 대한 공부 진행  **서지완(MCU 센서 멀티태스킹 구조 구현, IMU 정보 활용 방안 탐색 및 UART 통신 구조 개선, DHT11/22 센서 테스트, RC카 설계도(대략적) 작성 및 프레임 제작)**  **내용**  **1. IMU 센서 값(Yaw/Pitch/Roll) 계산 코드 작성**  – 가속도, 자이로 데이터를 읽어오는 함수 구현  　 – 읽어온 값을 UART 통신으로 전송하도록 구현  – YAW/Pitch/Roll 값을 계산하는 함수 구현  　 - Z축 자이로 값 적분 방식으로 Yaw 계산 　 - ±180° 범위 유지, yaw, pitch, roll 값 UART 출력 가능하도록 수정  **2. FreeRTOS 기반 센서 태스크 멀티화**  - 기존 센서 통합 태스크(StartDefaultTask)를 분리하여 각 센서 전용 태스크 생성 　 - StartMPUTask, StartDHT11Task, StartCDSTask 구현 　 - 각 태스크에 적절한 주기 (osDelay(2000) 등) 설정 　 - 타이머 정확도 및 태스크 간 간섭 없이 병렬 동작 확인  **3. UART 출력 전용 태스크 구조 도입**  - 각 센서 태스크에서 직접 HAL\_UART\_Transmit() 하지 않도록 구조 개선 　 - 센서 메시지 전송 전용 구조체 SensorMessage\_t 정의 　 - FreeRTOS 메시지 큐(osMessageQueue) 생성 및 적용 　 - UART 전용 태스크 StartUARTTask() 구현 　 - 센서 태스크 3개 모두 osMessageQueuePut() 방식으로 메시지 전달하도록 수정  **4. RC카 대략적인 설계도 작성 및 프레임 제작**  – 각 센서와 부품들의 치수를 잰 후, 설계도를 작성 　 – 전면도, 측면도, 평면도 도면을 ppt로 작성하였음. 　 텍스트, 스크린샷, 도표, 디자인이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.텍스트, 스크린샷, 도표, 디자인이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.텍스트, 스크린샷, 도표, 디자인이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.  – 설계도 기반으로 프레임 제작  – 설계도에 부품 및 볼트 구멍을 표시 후, 목공실에서 구멍을 뚫었음.  – 추가적으로 필요한 지지대를 구매하였음.  **결론**   * DHT11 센서 값 읽어오기 실패. -> DHT22로 시도했으나 실패 -> 습도 센서 없이 습도를 임의적으로 올리는 환경을 구현 시도 예정 * IMU 값 읽어오는 코드 구현 * 센서별로 값을 읽어오는 태스크 분리 및 UART 통신하는 태스크 생성 * RC카 프레임에 대한 설계도를 PPT로 제작 후, 프레임 제작   **박준현(이미지 디헤이징 실시간 영상 적용 및 테스트 yolov5 결합, 이미지 디헤이징 논문 분석)**  **내용**  – AOD-net 실시간 영상에 적용 및 yolov5 결합  – 실시간 영상에 AOD-net 적용  – opencv에서 받아온 각 프레임당 디헤이징 적용  opencv에서 받아오는 프레임은 제한된 상태  - 디헤이징한 프레임을 yolov5에 적용  – 디헤이징한 데이터를 yolov5 모델에 전달하여 객체 인식  - 디헤이징 + yolov5의 객체 인식 향상 테스트  (1)    (2)    (3)    (4)  - 사진 (1) (디헤이징 적용x, 안개 필터 x)과 사진 (3) (디헤이징 적용o, 안개 필터 x) 를 비교하였을 때 평상시에는 디헤이징을 적용하지 않는것이 객체 인식력이 높아 보임.  - 사진 (2) (디헤이징 적용x, 안개 필터 o) 과 사진 (4) (디헤이징 적용o, 안개 필터 o) 를 비교하였을 때 디헤이징을 적용한 것의 객체 인식력이 월등히 높음  - 지속적으로 사용시 메모리 사용량이 높아져 실행이 중단됨, 해결책 필요    - 이미지 디헤이징 논문 분석  – Optimized Contrast Enhancement for Real-Time Image and Video Dehazing 논문 분석  – 대비 향상과 정보 손실 최소화를 동시에 고려한 Optimized Contrast Enhancement기반 안개 제거 알고리즘을 제안  – 블록 단위 전달 값 추정 후 Edge-Preserving Filter와 Shiftable Window Scheme을 적용하여 픽셀 기반 복원 영상을 생성  - 영상 처리 시 프레임 간 밝기 변화에 따라 Temporal Coherence Cost를 적용하여 시간적 일관성을 유지  - Luminance-Only Processing, Downsampling, Partially Overlapping Sub-Blocks, SIMD및 OpenMP기반 병렬 처리 기법을 활용해 실시간 처리 속도를 확보  - Adaptive Contrast Based Real-Time Image and Video Dehazing 논문 분석  - Adaptive Contrast Optimization을 기반으로 한 전송 맵(transmission map) 추정을 통해 대비 향상과 정보 손실을 최소화하는 방식 제안  - Quad-Tree Search기반의 대기광(atmospheric light) 추정으로 안개 제거 정확도를 높이며, 밝기 변화가 큰 영역을 정밀하게 탐색  - 전송 맵 정제를 위해 Gaussian Filtering과 Guided Filtering을 연속 적용하여 에지 보존 및 노이즈 제거를 동시에 수행  - 실시간 성능 확보를 위해 CUDA Programming과 Custom GPU Kernels을 이용한 병렬처리와 영상 해상도 다운스케일링을 도입  - 영상 처리에서는 프레임 간 유사도를 반영한 Temporal Coherence Factor를 계산하여, 반복 연산을 줄이고 30fps 수준의 처리 속도를 달성  **결론**  jetson에서의 이미지 디헤이징을 결합한 yolov5 구현 성공  그러나 리소스 사용량이 많아지면 프로그램이 중지되는 현상 발생  최적화 시스템의 구현 필요  **이준영(MCU 초음파 센서 3개 융합, 이미지 디헤이징 논문 분석)**  **내용**  MCU 보드 초음파 센서 융합  – 초음파 센서 task 나누기  – task 분리 코드 작성  – 각 초음파 센서 별 pin 분배 및 세팅 완료    - 초음파 센서 사용시 간섭 확인  – 초음파 센서 두 개의 각도를 조절하면서 간섭 확인    2. 이미지 디헤이징 논문 분석  – Dehazing & Reasoning YOLO 논문 분석  – 기존 방법 : 안개 상황에서 객체 감지 전 이미지 사전 처리 → 실시간성 떨어짐 (객체 감지 + 실시간 성능 보장되어야 함)  – 해당 문제에 대한 논문에서의 대안점 정리  – Restoration Subnet Module(RSM) 대기 산란 모델 사용  – Relation Reasoning Attention Module(RRAM) 동시 발생 관계 그래프 도입  – Adaptive Feature Fusion Module(AFFM) 위 두 개를 병합하기 위해 사용  **결론**  초음파 센서 3개 task 분리 및 거리 계산 테스트 완료  **최창욱(MCU 센서 융합, 이미지 디헤이징 논문 분석 등)**  **내용**  1. FreeRTOS 기반 MCU 모듈화 및 Task 구조 설계  - ultrasonicTask, motorTask, uartPrintTask로 역할 분리  - CMSIS-RTOS V2 기반 osThreadNew()및 osMessageQueueNew()구조 적용  – UART 전용 Task 구현  - 모든 UART 통신을 전용 Task에서 일괄 처리  - Queue를 통해 메시지를 수신하고 UART 출력 전담 처리  2. 코드 및 기능 분리 작업  – 소스 코드 모듈 분리  – motor.c, ultrasonic.c, freertos.c, uart\_task.c등으로 기능별 관리  – 모듈화로 유지보수 및 확장성 향상  – DC 모터 제어 기능 구현  – 전진, 후진, 좌회전, 우회전, 정지 등 모든 동작 구현 및 테스트 완료  – PWM 신호 제어와 방향 핀(GPIO) 설정을 통한 제어 구성  3. 주요 이슈 대응 및 해결  – 모터 미동작 이슈 해결  – PWM 설정, GPIO 모드, 타이머 설정, 회로 연결 등 ioc 설정과 하드웨어  상태 포함 전면 검토  – 단순한 코드 수정만으로는 문제 해결이 어려우며, CubeMX 세팅 및 외부  회로 점검까지 병행해야 함을 확인  – 디버깅 환경에서는 타이밍 이슈로 실제 동작과 차이가 발생할 수 있음  – 모든 기능 검증은 실행 모드(Run)에서 최종 확인 필요  **결론**  FreeRTOS 기반 구조에서 초음파 센서와 DC 모터 연동 정상 작동 확인  Task 및 Queue 기반 구조 안정화 완료  시스템의 신뢰성을 위해 ioc 설정, 코드 구조, 하드웨어 배선 점검의 병행이 필요함을 확인 | |
| 개발계획 대비 진행현황 | |  |  |  | | --- | --- | --- | | **주차** | **계획 내용** | **달성 유무** | | 1주차 | 자율주행과 관련된 정보 조사 | 달성 | | 2주차 | 자율주행에 필요한 센서 조사 | 달성 | | 3주차 | 자율주행 RC카에서 사용할 센서 선정 | 달성 | | 4주차 | RC카 주행을 위한 선행기술에 대한 추가 정보 조사 | 달성 | | 5주차 | 구매한 센서를 MCU 및 Jetson에서 개별 테스트 진행 | 지연(지연 달성) | | 6주차 | RC카 설계 진행 | **달성** | | 7주차 | 트랙 제작 진행 | 예정 | | 8주차 | 자율주행을 위한 주행 알고리즘 개발 및 구현 | “ | | 9주차 | 자율주행을 위한 주행 알고리즘 개발 및 구현 | “ | | 10주차 | 기상 문제 없이 자율주행 테스트 진행 및 디버깅 | “ | | 11주차 | 이미지 디헤이징과 센서 융합을 결합 | “ | | 12주차 | 이미지 디헤이징과 센서 융합을 결합 | “ | | 13주차 | 기상 악화 상황의 트랙에 대한 자율주행 테스트 진행 | “ | | 14주차 | 디버깅 및 성능 개선 | “ | | 15주차 | 최적화 및 자료 정리, 발표 진행 | “ | | |
| 다음주  계획 | **손주석**  RC카 부품 도착 시 부품들 조립하기  제어 알고리즘 구현을 위한 IDE 사용법 공부 및 제어 알고리즘 구현  **박준현**  실시간 디헤이징을 적용한 논문과 추가 조사를 통하여 모델의 경량화,  센서를 통해 디헤이징이 필요한 경우에 선택적으로 실행되도록 구현,  추가적으로 jetson의 리소스 사용량을 줄일 수 있는 방안 탐색  **이준영**  모터 드라이브와 초음파 센서 3개의 센서 융합 코드 테스트  **서지완**  조도 센서, HC-05, MPU6050(IMU), L298N(모터 드라이브), LED, HC-SR04(초음파)를 모두 결합하여 멀티태스킹 마무리  RC카 프레임에 각종 센서들과 보드들을 부착시키기  **최창욱**  FreeRTOS 기반 구조에서 초음파 센서와 DC 모터 연동 정상 작동 확인  Task 및 Queue 기반 구조 안정화 완료  시스템의 신뢰성을 위해 ioc 설정, 코드 구조, 하드웨어 배선 점검의 병행이 필요함을 확인 | |
| 주요 결과물 | 없음 | |