

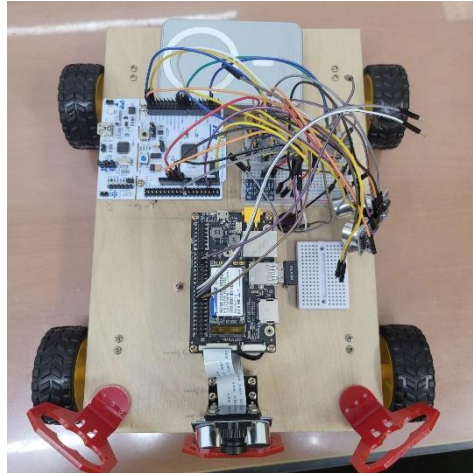
실증적SW개발프로젝트 주간보고 (7주차)

작성일: 2025/04/20 팀명: Wiper

팀 활동 보고	활 동 일 시	2025.04.14 ~2025.04.20
	장 소	동아대학교 Makerspace 창업실 5번방
	참 석 자	박준현, 서지완, 손주석, 이준영, 최창욱
	특 이 사 항	없음
이번주 진행사항	<div>1. 개발내용</div> <div>MCU<ul style="list-style-type: none">- SPI 통신 테스트- Git 세팅- 초음파 센서 핀 분리 및 freertos 변경- 모터 제어 디버깅 진행 후 블루투스 제어 완료</div> <div>Jetson<ul style="list-style-type: none">- SPI 통신 테스트- 객체의 크기 인식 및 위치 파악- Jetson orin nano 객체 인식 종류 제한 테스트</div> <div>이미지 디헤이징<ul style="list-style-type: none">- AOD-net 경량화- 디헤이징의 선택적 구현</div> <div>RC카<ul style="list-style-type: none">- 설계 및 조립- RC카 제어 데이터 설계 및 구현</div> <div>기타<ul style="list-style-type: none">- 깃허브 리팩토링- 트랙 제작</div> <div>2. 팀원별 활동내용</div> <div>손주석(팀장, RC카 제작 및 제어 알고리즘 구현)</div> <div>내용</div>	

RC카 설계 및 제작

1. RC카 부품 변경에 따른 설계 조정
 - i) RADAR 미사용 결정에 따른 초음파 센서 설계 추가
 - ii) 차체 안정성을 위한 DC모터 위치 조정
2. RC카 제작
 - i) 설계도에 따른 부품 조립 및 오차 수정
 - ii) 제작한 RC카를 기존 코드를 이용해 테스트 진행



MCU와 Jetson SPI 통신 설정

1. 핀 연결
 - i) Jetson을 Master, MCU를 Slave로 설정해 MISO, MOSI, SCK, CS핀을 연결함. 추가적으로 GND 핀도 상호간 연결
2. 수동 CS 설정 -> 실패
 - i) MCU 설정 - GPIO핀에 NSS 설정을 하지 않고 GPIO input을 설정함으로써 Jetson이 통신 유무를 제어하도록 함
 - ii) Jetson 설정 - Jetson 측에서 CS핀의 통신을 GPIO를 이용해 관리하도록 코드를 작성하고 실행함. 하지만 GPIO 설정에 대한 허용 권한이 없고 코드가 복잡해져 다른 방안을 이용하는 것으로 결정함.
3. 자동 CS 설정(NSS)
 - i) MCU 설정 - ioc의 GPIO핀에 NSS 설정을 이용함에 따라 CS핀이 자동으로 관리되도록 설정함.
 - ii) Jetson 설정 - Raspberry pi에서 SPI 통신으로 쓰던 예제와 동일하게 자동으로 CS핀이 관리되도록 함.

MCU의 모터 드라이버 오류 수정

1. 센서 전류 과다 사용으로 예상한 디버깅 진행
 - i) IMU, 초음파, Cds 센서 등을 사용하지 않고 모터드라이버와 블루투스만을 이용해 작동하는 코드를 이용해 테스트를 진행함.
 - ii) 다른 센서를 사용하지 않음에도 동일하게 동작이 되지 않는 것을 확인함.
2. 오실로스코프를 통한 전압 분석
 - i) 모터 작동 신호를 관리하는 핀인 ENA와 ENB핀을 확인함. 하지만 파형 및 전압에 문제가 없는 것을 확인함.
 - ii) IN1 ~ IN4 핀 전압을 확인함. 파형이 디지털형태가 아닌 아날로그형태가 나오는 것을 확인하였으며 전압이 의도한 바와 다르게 낮게 걸리는 것을 확인함.

3. 핀 연결 수정

- i) GND핀에 다른 센서가 너무 많이 연결되어 있어 간섭을 고려해 서지완 학생이 GND 핀을 수정함. 후진을 제외한 나머지 동작이 문제없이 작동하는 것을 확인
- ii) IN1 ~ IN4 핀 수정. 현 IN핀에 다른 인접한 핀이 없도록 만들기 위해서 핀의 번호를 수정함. IN핀을 다른 핀들과 격리함에 따라 신호 간섭이 사라져서 후진 동작이 문제없이 작동하는 것을 확인할 수 있었음.

Github refactoring

- 1. 필요없는 디렉토리 삭제
- 2. MCU와 Jetson에서 각각 git clone을 통해 작업을 진행할 수 있도록 분리
- 3. 이외 Docs와 Test에 대한 디렉토리 정리

박준현(팀원, MCU 세팅, 이미지 디헤이징 논문 분석 등)

내용

AOD-net 경량화

- 1. network pruning를 사용하여 경량화
 - i) L1-norm이 낮은 weight 중에서 30%를 가지치기하도록 구현
 - ii) 가지치기 함수를 aod-net 모델에 적용
- 2. 결과
 - i) aod-net의 디헤이징 능력이 약간 감소하였음 하지만 객체 인식에는 문제 없음
 - ii) 구동 중 프로그램이 종료되지 않으며, 리소스의 사용량이 줄어든것을 확인

디헤이징의 선택적 구현

- 1. 특정 상황에 aod-net이 적용된 frame으로 객체인식이 작동 되도록 구현
 - i) 정확이 어느 환경에서, 어떠한 방법으로 특정 상황을 만들것인지 구체화 필요
 - ii) 추후 정확한 환경이 결정된다면 코드 수정이 필요할 가능성이 있어서 임시로주석 처리

객체의 크기 인식 및 위치 파악

- 1. yolov5로 인식한 객체의 크기 인식
 - i) cv 창 기준으로 좌표가 생성
 - ii) top_left, bottom_right 좌표 추출하여 크기 도출
- 2. 객체의 위치 파악
 - i) 거리 측정 함수
$$\text{Distance} = (\text{Real width} * \text{Focal Length})/(\text{Preceived Width})$$
 - ii) 함수를 기반으로 코드 작성
 - iii) 추후 제작할 트랙의 장애물 크기를 측정하고 테스트 필요

서지완(팀원, GIT 병합, RC카 조립 및 모터 제어 버그 디버깅 등)

내용

GIT 병합

1. github repo와 local repo를 remote
 - i) 기존 local에서 작업하던 MCU 프로젝트를 팀 git repository의 MCU 디렉터리와 연동하는 작업을 진행하였음.
2. markdown 파일 생성
 - i) Core/Inc, Core/Src 내의 파일들에 대한 설명 추가

sensors.c/h 파일 분리

1. dht.c/h
 - i) dht11로 센서 값을 읽고, 메시지 큐에 전송하는 코드로 분리
2. cds.c/h
 - i) cds로 센서 값을 읽고, 메시지 큐에 전송하는 코드로 분리

RC카 조립 및 모터 제어 버그 디버깅

1. RC카 프레임 제작 및 조립
 - i) dc 모터 구멍 추가 제작
 - ii) 초음파 센서 지지대 부착 및 초음파 센서 선 연장
 - iii) 모터 점퍼 선 연장
 - iv) 나머지 부품을 조립하여 RC 카 완성
2. RC카 완성 후 생긴 모터 버그 수정
 - i) RC카 조립 후 모터 회전 디버깅을 하며 정리한 표

모터 회전 방향	
앞	f
뒤	b
f, b 뒤에 s	: 느리게 돌
IN(x)	r GPIO_RESET
	s GPIO_SET

최종 테스트								
동작	IN1	IN2	IN3	IN4	원앞	원뒤	오앞	오뒤
전체 전진	r	s	r	s	fs	fs	f	f
전체 후진	s	r	s	r	b	b	b	b
왼쪽만 전진(우회전)	r	s	s	s	f	f	x	x
오른쪽만 전진(좌회전)	s	s	r	s	x	x	f	f
정지	s	s	s	s	x	x	x	x
왼쪽만 후진	s	r	s	s	b	b	x	x
오른쪽만 후진	s	s	s	r	x	x	b	b

motor.c 코드의 전진 함수를 수정하며, 모터가 제대로 동작하도록 디버깅하였음.

ii) 디버깅하며 알게 된 문제점

- 기존 좌회전의 경우, 좌측 모터 정지 & 우측 모터 전진 → 이렇게 하면, 좌회전이 가능할 것으로 예상하였음. 그러나 실제 작동해보니, 좌측 바퀴 또한 우측 바퀴와 함께 굴러가는 것을 알 수 있었음. → 생각해낸 해결 방안: 좌회전 시, 좌측 모터 후진 & 우측 모터 전진 → 이렇게 하면 좌회전 가능할 것으로 예상함.
- 모터의 출력이 줄어들음
모터 드라이브로부터 점퍼 선을 끌어올 때, 길이가 짧아서 연장하였는데, 모터의 출력이 줄어든 것처럼 보임 → 모터 전진 시, 한쪽 바퀴들만 전진하고, 다른 쪽 바퀴들은 떨리기만 함. (출력이 줄어든 것으로 추정)

3. RC 카 모터 제어 버그 수정 (완)

- i) 센서 전류 과다 사용 의심
 - 시도: 타 센서를 사용하지 않는 간단한 코드로 모터만 구동
 - 예상: 센서들이 과도한 전류를 소비해 모터 드라이브에 신호 전달 불가
 - 결과: 동일한 오작동 발생
 - 센서 전류 소비는 원인이 아니었음
- ii) 복잡한 회로 제거 후 단순 테스트
 - 시도: 모터 드라이브 전용 코드로 회로 간소화
 - 결과: 여전히 오작동 발생
 - 회로 복잡도가 직접 원인은 아님
- iii) 오실로스코프를 통한 신호 파형 분석
 - 시도: ENA/ENB, IN1~IN4 핀 전압 측정
 - 결과:
 - ENA, ENB 는 정상적인 PWM 출력 확인
 - IN1~IN4 는 이상한 아날로그 파형 (삼각파 등)
 - 디지털 신호가 제대로 출력되지 않음
- iv) 접지(GND) 문제 의심 및 수정
 - 시도: MCU와 모터 드라이브를 직접 GND 연결
 - 결과:
 - 후진 제외 전부 정상 동작 확인
 - GND 문제였음 → 접지 간섭/불안정 해소
- v) 핀 간 신호 간섭 제거 (물리적 거리 확보)
 - 시도:
 - IN1~IN4, ENA/ENB 핀 배선을 하나씩 건너뛰도록 재배치
 - 간섭 최소화를 위한 핀 분산
 - 결과:
 - IN2 핀 위치 조정 → 왼쪽 바퀴 정상 작동
 - IN3 핀 위치 조정 → 오른쪽 바퀴 후진 정상 작동
 - 후진 문제 해결 완료

이준영(팀원, Jetson 객체 인식 종류 제한, 초음파 센서 핀 분리 및 freertos 변경)

내용

Jetson 객체 인식 제한

1. yolov5의 객체 인식 사물의 인덱스 파악
 - i) 자율주행에 필요한 객체 인식 종류 파악 (사람, 차, 신호등 등)
 - ii) 선택한 종류에 대한 yolov5의 인덱스 파악
2. 객체 인식 코드에 적용 가능한지 테스트
 - i) 먼저 객체 인식을 진행 후 후처리로 해당 번호에 맞는 사물만 인식
 - ii) 실제 테스트로 인덱스로 사람만 처리 후 테스트

초음파 센서 핀 분리 및 freertos 변경

1. MCU 보드에서 중복되는 핀 분리
 - i) 초음파 센서에 사용 가능한 핀들 종류에서 중복된 것을 찾고, 최적의 핀 매핑

2. Freertos로 코드 변경

- i) main.c에서만 테스트 분리 후 각각 uart 출력을 하고 있음.
- ii) freertos에서도 불러오게 변경 후 uart 큐에 센서값을 넣어 출력하게 변경

최창욱(팀원, GIT 연동, 트랙 제작, 무선 통신 방법 탐색)

내용

무선 통신 방식 검토 및 구조 설계

1. 문제 인식

자율주행 테스트 중 MCU(STM32)와 노트북 간 USB 유선 연결이 차량 이동 시 선 꼬임과 연결 불안정을 유발하여 테스트 정확도 저하 가능성 발생

2. 해결 방안 탐색

ESP32를 STM32에 연결하여 WiFi Serial Bridge 방식을 **무선 통신 구조** 설계.
노트북에서는 VSCode로 서버를 열고, UART를 통해 ESP32와 STM32가 제어신호를 주고받도록 계획.

3. 진행 상황

현재까지 무선 통신 구조 설계 및 필요성 분석을 완료하였으며, 실제 구현 및 테스트는 향후 예정

Git 연동 시도

1. 목적

STM32CubeIDE와 GitHub 간의 연동을 통해 코드 버전 관리 및 팀원 협업 효율화

2. 진행 내용

- i) 원격 저장소 URL 등록, 브랜치 선택, 디렉토리 경로 지정 등 기본 연동 과정 수행
- ii) .git확장자 누락, 디렉토리 비어 있지 않음 등의 이유로 복수의 오류 발생
- iii) Eclipse 기반 UI 오류 (Unhandled event loop exception)도 확인됨

3. 결론

현재까지 Git 연동은 성공하지 못했으며, 추후 새로운 디렉토리 생성 또는 CLI 연동 방식 검토 예정

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	지연
	8주차	자율주행을 위한 주행 알고리즘 개발 및 구현	예정
	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 제어 알고리즘의 설계에 대한 구체화 진행 MCU와 Jetson 데이터 송수신에 대한 설계 구체화 및 구현 주요 설계 알고리즘을 테스트 문제가 발생하는 알고리즘 디버깅 및 테스트 진행		
	박준현 트랙 제작 완료 + 장애물 크기 측정 인식한 객체 데이터를 mcu 보드에 전송 or 회피를 위한 데이터 전달		
	서지완 자율주행 알고리즘 탐색		
	이준영 객체 인식 종류를 제한한 것으로 자원 소모를 줄이고, 파악한 값을 이용해서 자율주행에 사용 및 테스트		
	최창욱 무선 통신 환경 구축 실험, Git 연동 재시도, 트랙 기반 자율주행 테스트		

주요 결과물

없음