Line Maze Race

목 차

0. 고객 요구사항

***기본 기능**

1. 미로 경로를 최적경로 알고리즘에 따라 시작에서 끝으로 이동할 수 있어야 한다. (Pass/Fail)
2. 자율이동 로봇이 주행선을 따라 주행할 수 있다. 대회에서 사용하는 경기장은 대회 당일에 공개한다. (Pass/Fail)
3. Client 모니터링 화면에는 속도, Map정보, 각종 센서 정보가 기록된다.

***조별 Race (예선전 : 각 반, 본선 : 통합)**

1. 미로 찾기 : 미로 경로는 최단 시간안에 통과한다.
2. 꼬리 잡기 : 경기장에서 속도를 경쟁하며, 꼬리를 잡히지 않은 로봇이 통과, 제일 먼저 잡힌 조부터 순위 기록한다.

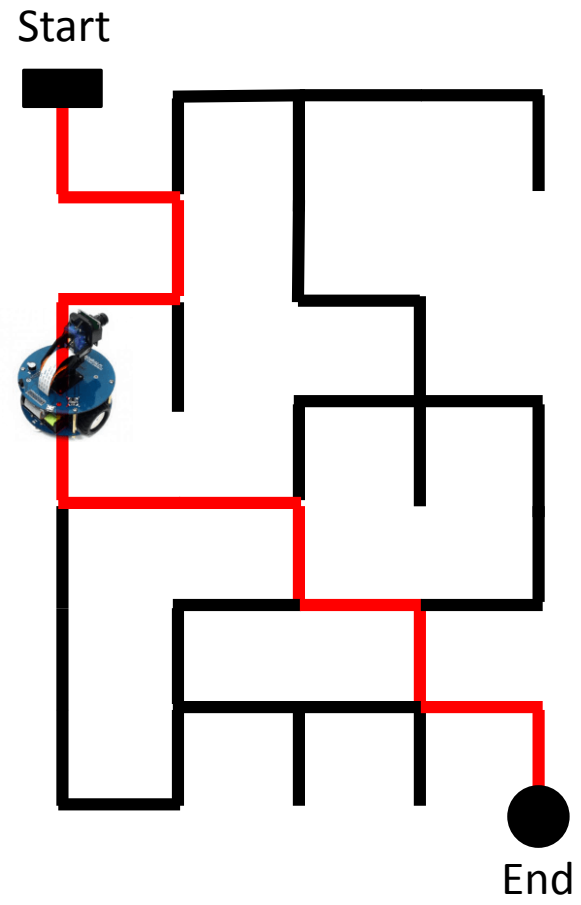
***가점 기능 (3개 중 하나 이상 반드시 구현할 것)**

1. 로봇에 부착된 카메라에서 영상을 전송하여, 경기상황을 클라이언트에서 영상으로 play할 수 있다. (난이도★ ★ ★ 15점)
2. 적외선 센서로 장애물을 인식하여, 경고음 출력 후, 피할 수 있다.(난이도★★ 10점)
3. 속도에 따라 RGB불빛이 다른 색을 반짝이고, 소리를 발생한다. .(난이도★ 5점)

1. 프로젝트 미션 1

Line Maze Robot 이란 ?

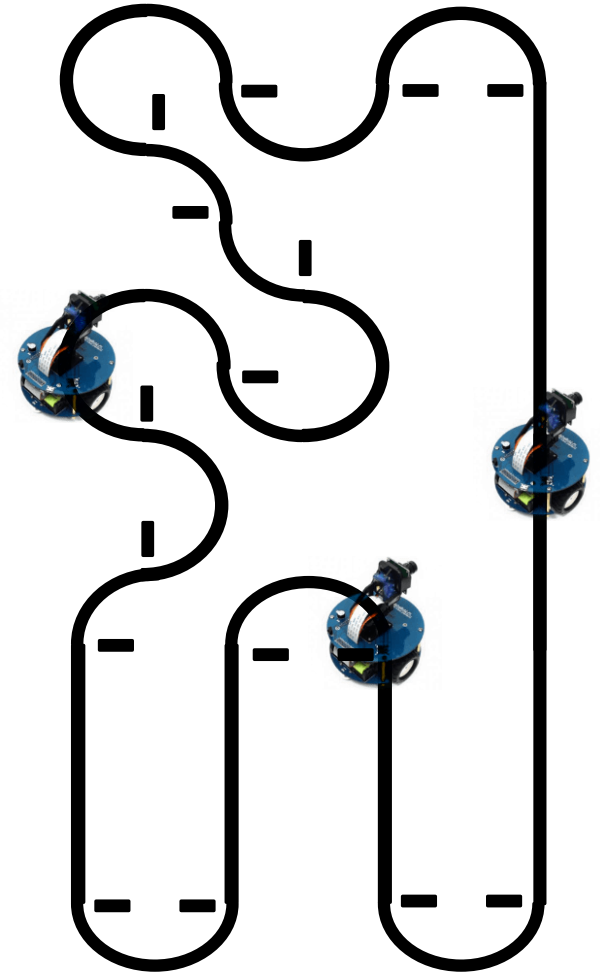
- 기존 벽으로 미로 찾기를 하였던 Micro Mouse의 또 다른 버전임
- 각 라인 미로에는 시작점과 끝점이 있음.
- 1차 주행에서 라인의 모든 경로를 로봇이 순회하며 라인정보를 기록함
- 1차 주행에서 기록된 정보를 이용하여 최단경로를 원하는 알고리즘을 이용하여 찾는다.
- 2차 주행에서 최단경로로만 주행하여 시작점에서 끝점까지 도달하면 됨
- 연습용 미로가 제공되지만 조별 대회에 사용될 미로는 대회 당일 공개함



1. 프로젝트 미션 2

Line Tracer 꼬리잡기 란 ?

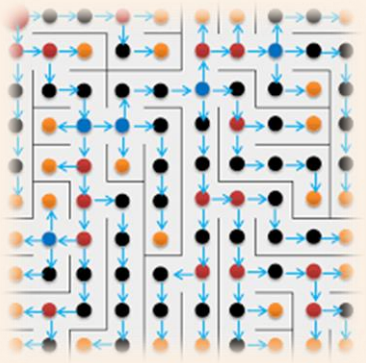
- 라인 트레이서 로봇 경기는 독립전원을 가진 자율이동 로봇이 정해진 주행선을 따라 주행하면서 속도를 경쟁하는 경기이다.
- 이런 로봇을 라인 트레이서라 하며, 주행이 안정적이고, 주행시간이 가장 짧은 트레이서 로봇을 최우수 라인 트레이서 로봇으로 한다.
- 1차 주행에서 전체 경로를 주행하며 각 커브의 방향 및 위치를 기록한다.
- 2차 주행에서는 1차 주행에서 기록된 정보를 이용하여 모터의 속도를 조절하여 주행하고 여러 조가 경쟁하여 꼬리잡기를 한다. 꼬리를 잡히지 않은 로봇이 예선을 통과한다.
- 연습용 경기장이 제공되지만 최종 조별 대회에 사용될 경기장은 대회 당일 공개함



2. 프로젝트 과목 목표

알고리즘 훈련

- 기본 과정에서 교육받은 알고리즘을 실전 로봇에 반영하여 효과 적인 알고리즘의 효율성을 체험할 수 있다.



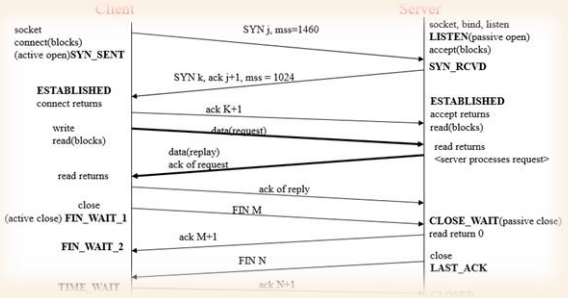
시스템 프로그램 능력 향상

- System Programming에서 교육받은 Linux System 상의 프로그램 구현 능력을 Wiring Pi 라이브러리 분석 및 활용을 통해 향상할 수 있다.

2. 프로젝트 과목 목표

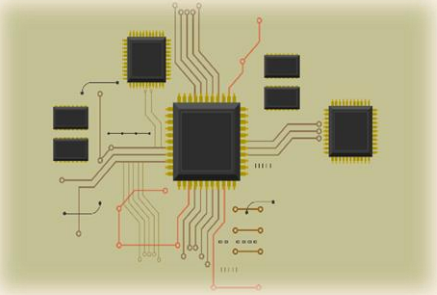
네트워크 프로그램 능력 향상

- 미로 찾기와 라인 트레이싱에 사용될 로봇을 서버로 하여 구현하고 제어 모듈을 클라이언트에 둠으로써 효율적인 네트워크 모델을 설계하는 능력을 향상 시킨다.



임베디드 프로그램 능력 향상

- 로봇의 각종 하드웨어(센서, RGB Led, Buzzer, Camera, 모터 …)를 제어하고 알고리즘에서 계산된 값을 정밀하게 모터제어에 반영하여 좀더 성능 좋은 로봇으로 성능을 향상 시킨다.



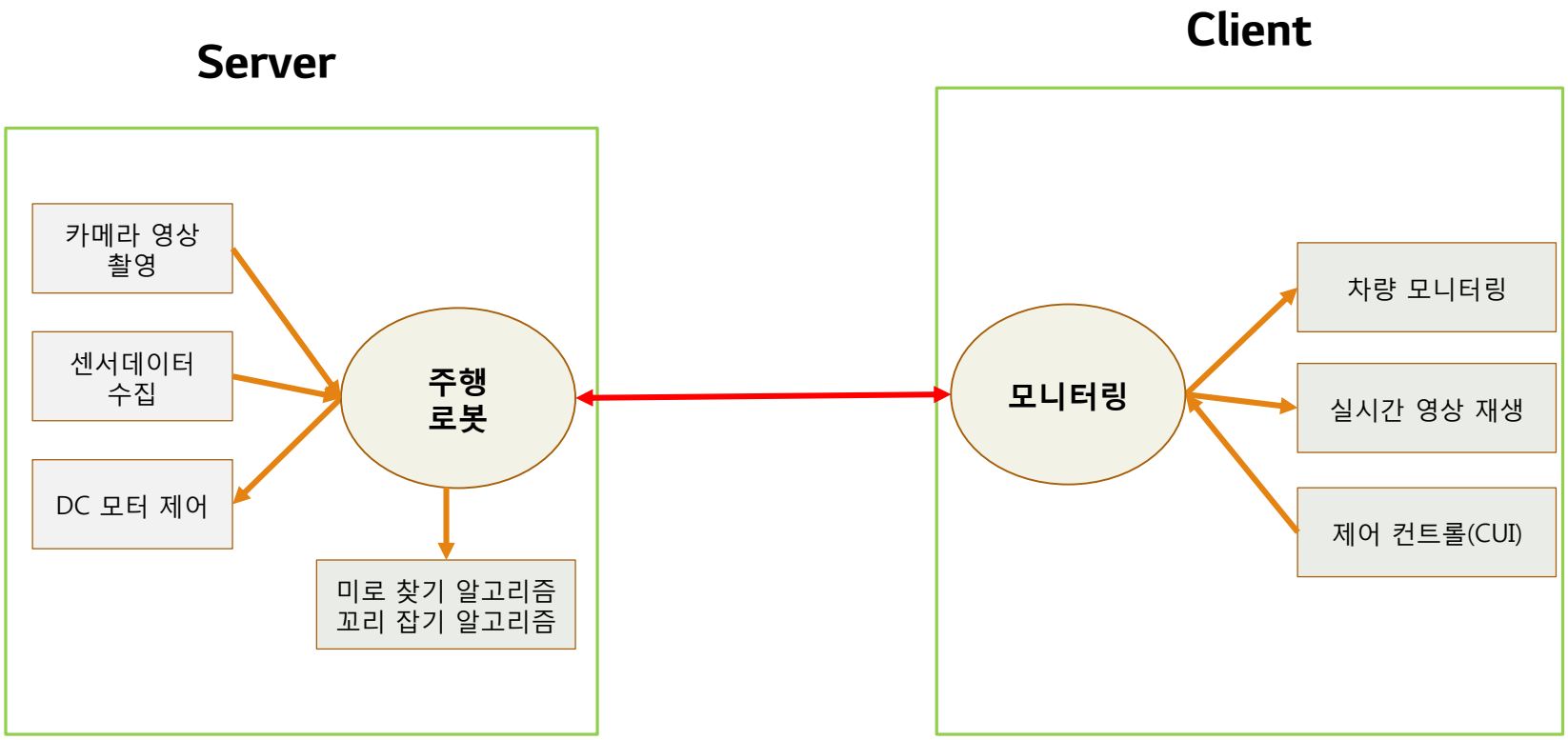
3. 목표 기술서

구 분	기능블록	규 격
Software	OS	RASPBIAN - Debian Wheezy. Linux kernel 4.9.79-v7+
	Compiler	GCC Compiler – 리눅스 커널 기반
	센서 데이터 전송	서버 프로그램 (Linux Network Programming) 센서 데이터 클라이언트 프로그램 (Linux 기반)
	Actuator 프로그램	Application (C 기반) Network (C 기반) 구동부제어 (Linux System Programming C 기반으로 개발)
Hardware	라인트레서 로봇	AlphaBot2 control interface Omni-direction wheel ITR20001/T: reflective infrared photoelectric sensor, for line tracking TB6612FNG dual H-bridge motor driver LM393 voltage comparator N20 micro gear motor reduction rate 1:30, 6V/600RPM Rubber wheels diameter 42mm, width 19mm Battery holder: supports 14500 batteries WS2812B: true color RGB LEDs

3. 목표 기술서

구 분	규 격
1. 조향 및 속도 기본제어	기어 모터를 라즈베리파이 WiringPI를 이용하여 C 기반(다른 언어 안됨)으로 정밀하게 제어한다.
2. 원격 제어	WiFi 네트워크를 통하여 현재 센서 정보를 Client에게 전송한다. 모터의 좌우 속도 및 방향을 실시간 Client에게 전송하여 모니터링 한다. 차량들의 출발 및 위급 상황에서의 정지 제어를 한다. 차량의 카메라를 이용하여 촬영 된 영상을 실시간으로 Client에게 전송하고 Client는 이를 play한다.
3. 라인 트레이스 기능 구현	라인 트레이스 센서를 이용하여 라인 트레이스 기능을 C언어 기반으로 구현 한다. 라인 트레이스시 속도 및 조향은 자동으로 제어 되도록 알고리즘을 구현 한다.
4. 미로찾기 및 꼬리잡기 알고리즘 구현	차량은 서버 역할을 하고 서버에서는 미로 찾기 알고리즘과 꼬리잡기 알고리즘을 구현하여 최적화된 하드웨어 제어를 구현한다.

4. 라인 트레이서 로봇 제어



- 미로찾기 동영상
- 꼬리잡기 동영상
- 사용자메뉴얼

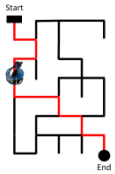
5. 시스템 구성도 (기본 구현)

Pi Camera

Line Follower

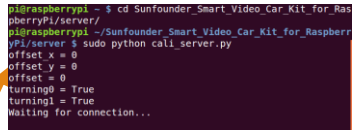
Gear Motor

Servo

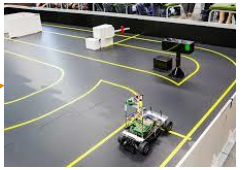


미로 찾기 알고리즘

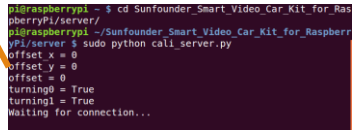
꼬리잡기 알고리즘



차량 모니터 링

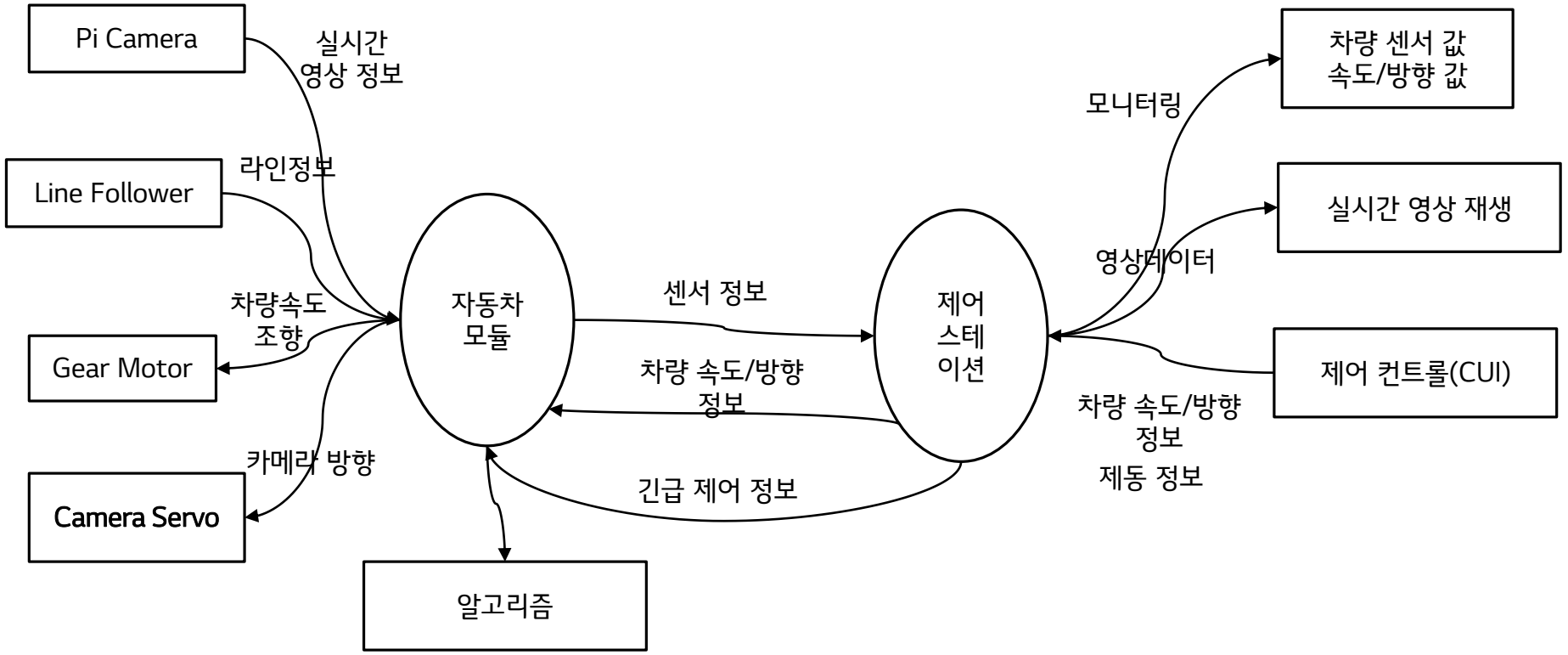


실시간 영상재생



제어 컨트롤(CUI)

6. Data Flow (기본 구현)



7. 모듈별 구현

- 로봇의 기본 구현

1) 로봇에서 촬영된 영상은 실시간으로 접속된 클라이언트에 중계하고 클라이언트는 영상을 재생해야 한다.

2) 로봇의 센서 및 속도/방향 등의 정보는 클라이언트에게 중계하여 클라이언트는 실시간 모니터링 정보를 출력한다.

- 조별 기본 라인 미로 찾기 (미션 1)

1) 1차 주행에서 전체 경로를 순회하여 최단 경로를 계산 한 다음 2차 주행 시 최단 경로로 주행하여 목적지 도달

2) 구현 완료 (Due date) : 목요일 16:00 시

3) 미로 찾기의 최단경로 알고리즘은 자유롭게 구현 한다.

4) 완료 기준 : 라인을 이탈하지 않고 최단 경로로 순회를 성공하면 통과(Pass) 그렇지 않으면 실패 (Fail)

- 조별 라인 트레이서 꼬리잡기 대회 (미션 2)

1) 경기 대진은 추첨을 통해, 3~4개의 조가 예선전을 한다.

- 예선전 : 금요일 09:00 시

2) 각 반에서 예선전을 통과한 한 조가 반별 대항전을 결승전으로 치룬다.(예선전과 동일)

- 결승전 : 금요일 10:00 시

3) 각 조의 순위는 탈락된 순서의 반대로 순위를 정한다.

4) 경로에 따른 속도 및 방향 제어 알고리즘은 자유롭게 구현 한다.

5) 기타 추가 기능은 자유롭게 구현하며, 가산 점수가 될 수 있다. 예) 상황에 따른 RGB LED 제어 , 부저 의 소리 출력 등

7. 모듈별 구현

[트랙을 만드는 방법]

자동차가 검은 선을 따라갈 수 있도록 트랙을 만들려면 다음 자료를 준비해야 한다.

*** 준비물** : 검은 테이프 롤 (검정색 선), 하드 카드 보드(트랙의 크기에 따라 다름)
또는 바닥이나 책상과 같은 평평한 표면.

1. 하드 보드에 용지를 부드럽게 펼쳐 보드 또는 평평한 표면에 붙여 넣는다.

2. 트랙 생성 규칙에 따라 테이프에 용지를 붙인다.

*** 트랙 생성 규칙 :**
1. 검은 선의 폭 : 약 18-30mm, 두 개의 탐침 사이의 거의 거리,
두 개의 인접하지 않은 프로브의 최소 거리 이상

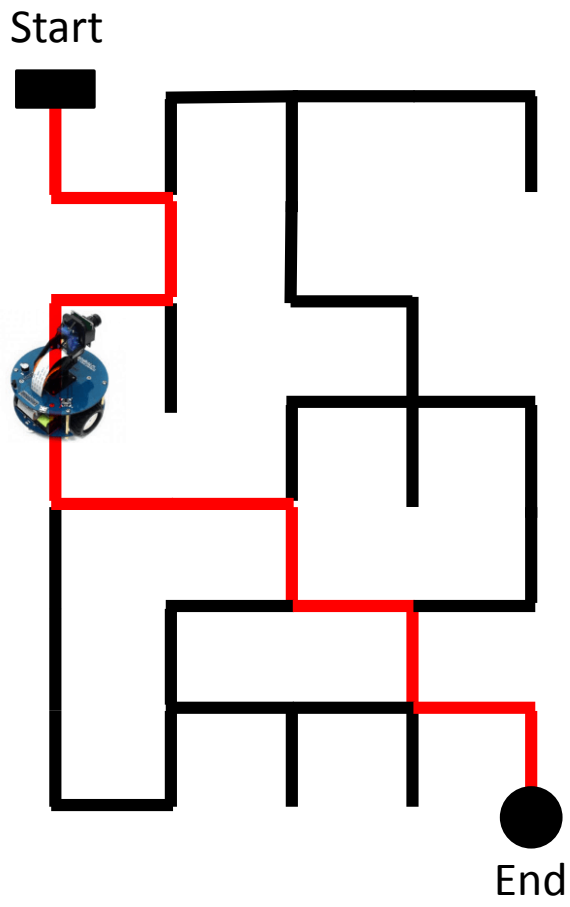
2. 2 개의 선 사이 간격 : 전체적인 단위의 폭인 125mm 이상,
동시에 두 줄을 감지 할 때 자동차가 혼란에 빠지게 한다.

3. 곡선의 반 직경 : 138mm 이상.

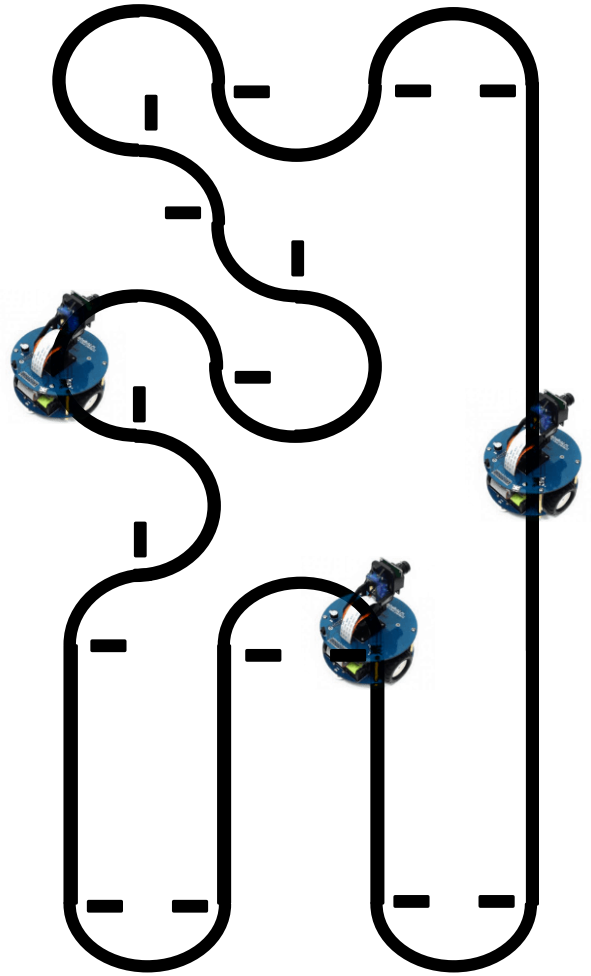
4. 곡선 표시 : 곡선의 유무를 표시하는 정보를 기록한다. (곡선의 시작과 끝에 방향에 따라 표시)

7. 모듈별 구현

[미로 찾기 예시]



[꼬리 잡기 예시]



8. 프로젝트 관리 - 유의사항

- **코드 복사 금지 (코드 유사성 검사 진행)**
- 본 프로젝트에서 제공되거나 작성되는 파일은 인터넷을 통해서 외부로 유출을 금지(모니터링 중)
- 본 프로젝트는 평가의 연장으로, 팀원들의 협업을 통해서만 진행되어야 합니다.
  모든 문제는 팀 내에서의 해결을 기본 원칙으로 합니다.
  단, 강사를 Mentor(기술전문가)로 조언을 받으실 수 있습니다.
- 본 프로젝트는 혼자 하는 프로젝트가 아닙니다. 한 명의 free-rider도 없이 모두가 참여하여 수행해야 합니다.

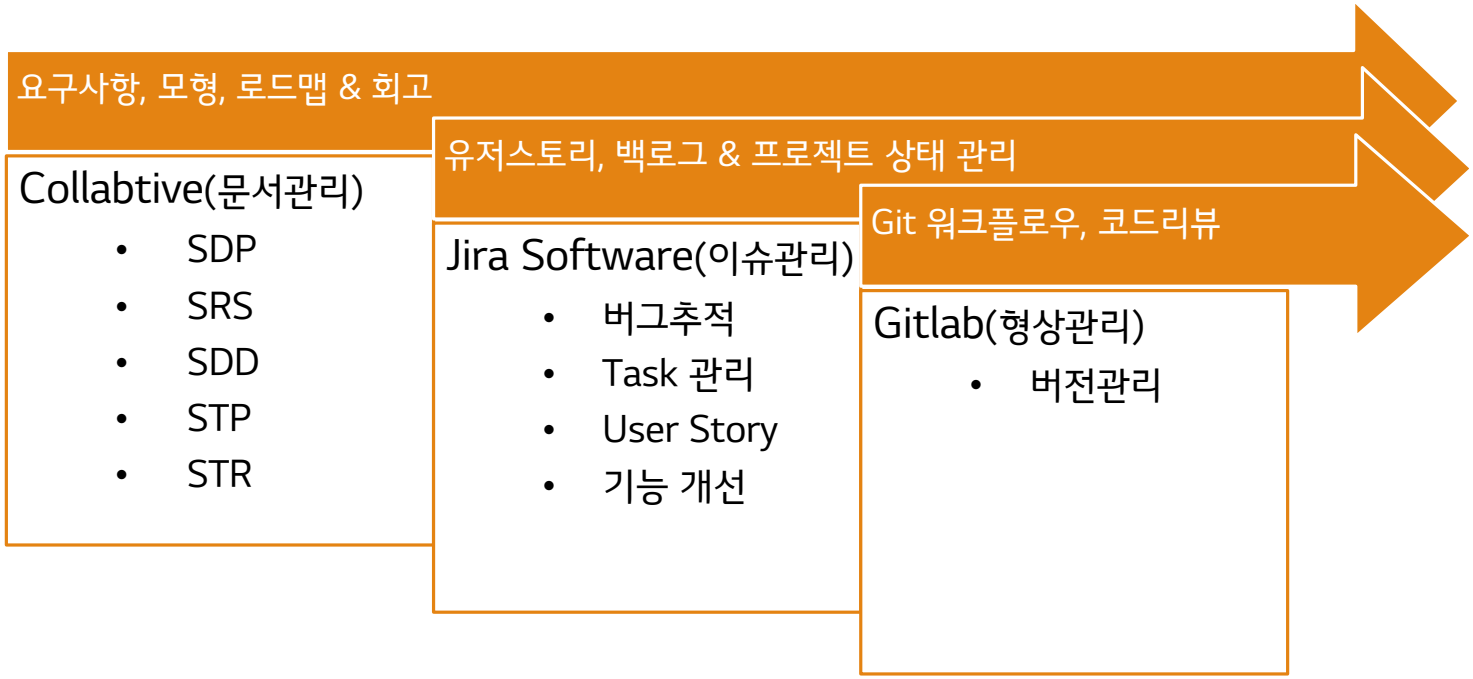
"만일 모든 사람이 같이 움직이고 있다면, 성공은 따 논 당상이다"

-헨리 포드

8. 프로젝트 관리 - 프로젝트 진행 가이드

일정		주요 내용	활동	산출물 (활용시스템)	평가기준 (비중)
04월8일(월)	개요	▪ 프로젝트 가이드 및 QnA			
SW E&P 4/15~22	계획	▪ 프로젝트 계획서 작성	➢ 모듈 별 구체적 프로젝트 계획 ➢ 업무 분장	프로젝트 계획서(Collab)	작성여부(P/F)
		• Daily Scrum Meeting	➢ Daily 리뷰	Collab	실행여부(P/F)
	설계	▪ 설계 개요 파악 ▪ 설계 모델링	➢ 최상위 아키텍처 정의 ➢ 시퀀스 다이어그램 ➢ 프로토콜 정의 ➢ DB설계 ➢ 구현일정 계획	HLD (Collab)	상호평가 (15%)
04월22일 ~25일	구현 & 테스트	▪ 구현 및 테스트 개요 ▪ Embedded Software 개발 ▪ 프로젝트 목표 시스템 통합 완료	➢ 기능별 모듈 개발 ➢ 통합 S/W 개발	TC를 포함한 소스코드	강사평가 (20%)
		▪ 단위 테스트 & 통합테스트	➢ 개발 S/W 테스팅		
04월25일 16:00		▪ 미로 찾기 미션 테스트		프로젝트 종료보고서 (ppt)	최종발표 (15%)
04월26일	최종 발표	▪ 프로젝트 종료 보고서 발표 ▪ Line Maze Race	➢ 최종 발표 자료 제출 ➢ 최종 소스코드 제출	Line Maze Race	• 기본기능 (15%) • 미로찾기 (10%) • 꼬리물기 (10%) • 옵션가점 (15%)

8. 프로젝트 관리 - Project 관리 툴 사용

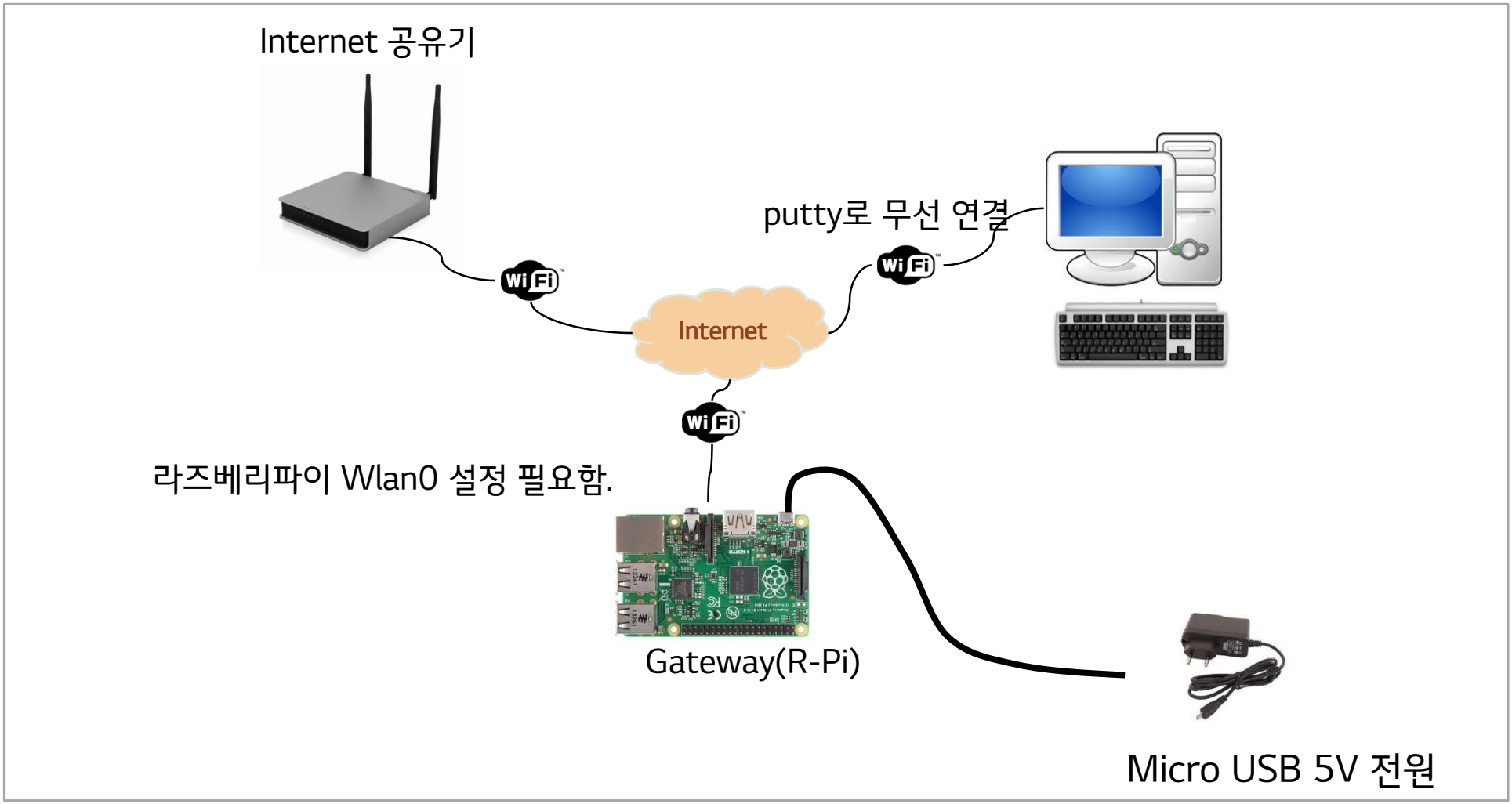


9. 프로젝트 관리 – 조 구성

➲ 조 구성 (4~5인 1조)

- 한 조는 하나의 자동차를 구현한다.
- 미로 찾기 및 라인 트레이서 꼬리잡기를 구현한다.

10. 개발환경


Internet 공유기
putty로 무선 연결
Internet
라즈베리파이 Wlan0 설정 필요함.
Gateway(R-Pi)
Micro USB 5V 전원