

# 창업연계공학설계입문

프로젝트 명	창업연계공학설계입문
팀 명	2 조
문서 제목	4 차 과제 보고서

Version	1.5
Date	2017-11-20

팀원	서형석(조장) / 201716
	서명희 / 20171627
	송지영 / 201716
지도교수	한재일 교수
분반	3 분반

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>4 차 과 제 보 고 서</b>		
	<b>프로젝트 명</b>	장애물 회피 라인트레이싱	
	<b>팀 명</b>	2 조	
	Confidential Restricted	내용 전체 수정	2017-11-20

#### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 공학설계입문 수강 학생 중 프로젝트 “장애물 회피 라인트레이싱”를 수행하는 팀 “3 분반 2 조”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “3 분반 2 조”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

## 문서 정보 / 수정 내역


<b>Filename</b>	4 차과제-Tracing.py
<b>원안작성자</b>	서명희, 서형석, 송지영
<b>수정작업자</b>	

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2017-11-18	서명희	1.0	최초 작성	서론 작성
2017-11-18	서형석	1.1	내용 추가	수행내용 작성
2017-11-19	서명희	1.2	내용 추가	애로 및 건의사항 작성
2017-11-19	송지영	1.3	내용추가	기본 아이디어 작성
2017-11-20	서형석	1.4	내용추가	향후추진계획 작성
2017-11-20	송지영	1.5	수정	내용 전체적으로 수정

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>4 차 과 제 보 고 서</b>		
	<b>프로젝트 명</b>	장애물 회피 라인트레이싱	
	<b>팀 명</b>	2 조	
	Confidential Restricted	내용 전체 수정	2017-11-20

## 목 차

1. 서론
2. 기본 아이디어
  - 2.1 H/W 디자인 방향
  - 2.2 S/W 디자인 방향
3. 수행내용
  - 3.1 장애물 인식 및 회피
  - 3.2 라인 트레이싱 및 곡률 코스 처리
4. 향후 추진 계획
  - 4.1 향후 계획의 세부 내용
5. 애로 및 건의 사항
6. 회의록


 <b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>4 차 과 제 보 고 서</b>		
	<b>프로젝트 명</b>	장애물 회피 라인트레이싱	
	<b>팀 명</b>	2 조	
	Confidential Restricted	내용 전체 수정	2017-11-20

## 1 서론

직접 조립한 자율주행 자동차에 라즈베리 파이로 코드를 실행하여 4 차 과제를 수행한다. 4 차 과제의 목표는 적외선 IR 광 추적 센서로 검은 선을 구별하여 라인 트레이싱을 하고 초음파 센서로 자동차의 앞을 가로막는 2 개의 장애물을 각각 인식하여 돌아가서 다시 라인 트레이싱으로 복귀한다.

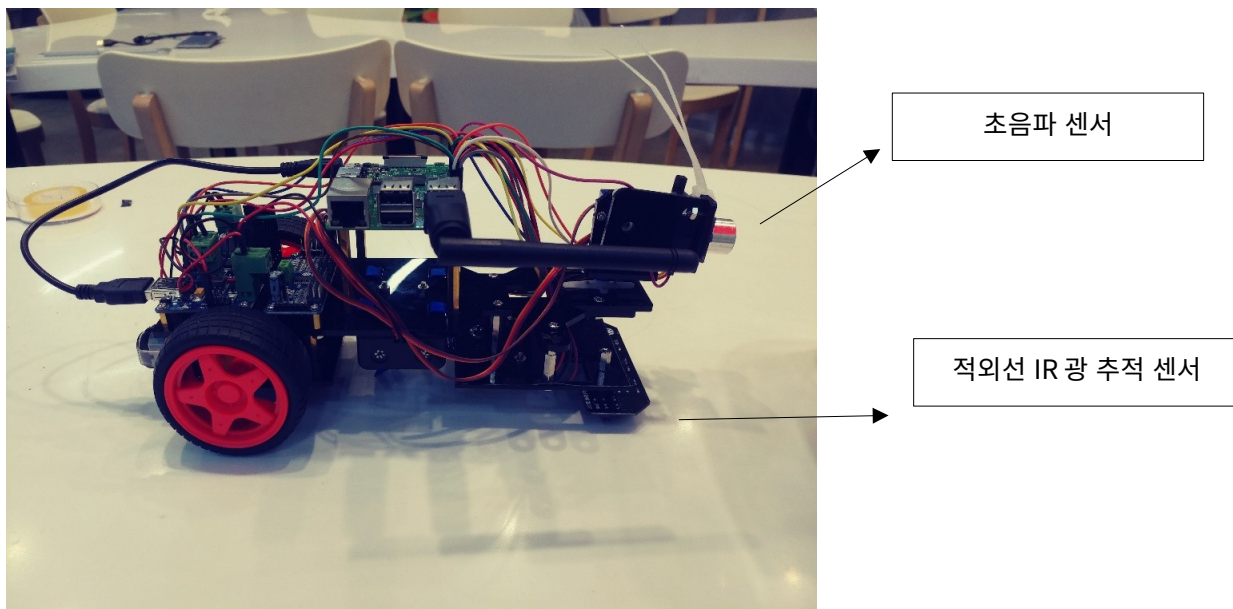
4 차 과제는 1,2 차로 나누어 평가된다. 1 차 평가는 반 시계 방향으로 라인 바깥선을 따라 라인 트레이싱을 하다가 장애물을 만나면 라인의 바깥 방향인 오른쪽으로 세 차례 턴을 하여 라인 트레이싱으로 복귀한 후 다시 주행하다가 또 하나의 장애물을 같은 방식으로 돌아가 출발했던 선으로 다시 돌아오는 것이다. 2 차 평가는 시계 방향으로 라인 안쪽 선을 따라 라인 트레이싱을 하다가 장애물을 만나면 라인의 안쪽 방향인 오른쪽으로 세 차례 턴을 하여 라인 트레이싱으로 복귀한 후 다시 주행하다가 또 하나의 장애물을 같은 방식으로 돌아가 출발했던 선으로 다시 돌아오는 것이다.

LED 가 바닥의 색을 인식하여 라인의 바깥인 하얀색일 때는 1, 라인에 있을 때인 0 을 출력하므로 5 개의 광센서의 출력 결과를 반영하여 경우마다 다른 주행 방향을 설정하는 코드를 완성해야 한다. 우리에게 주어진 5 개의 광센서도 충분히 많은 경우의 수가 나올 수 있지만 완벽한 라인 트레이싱을 위해서는 약간의 한계가 있다고 생각한다. 또한 주행 방향 뿐만 아니라 틀어진 정도에 따라 주행할 때 꺾여지는 정도까지 다르게 설정해주어야 한다. 장애물을 돌아갈 때에는 적절한 거리에서 장애물을 인식하고 원하는 각도로 최대한 정확하게 돌아야 한다. 장애물을 피하여 다시 라인으로 복귀할 때에는 라인과 자동차의 각도가 최대한 적게 하여 수월하게 들어가도록 해야 한다. 하지만 다양한 경우를 고려해서 코드를 각각 맞춰서 짤다고 해도 매번 자동차를 놓을 때 바퀴의 방향이나 무게 등의 요인에 따라 정말 미세한 오차에 따라서도 조금씩 결과가 달라진다. 그러므로 오차가 생기더라도 최대한 안정적으로 주행할 수 있도록 경우의 수를 모두 고려하여 코드를 짜는 것이 중요하다고 생각했다. 마지막으로 주행을 마치고 도착선에 닿으면 자동차가 주행을 멈추도록 해야 한다. 이 때는 광센서가 모두 00000 을 출력할 때 멈추도록 설정하는데, 장애물을 돌아오는 과정에서 라인과 자동차의 각도가 90 도에 가까운 경우에는 00000 으로 인식하고 멈출 수 있으므로 각도를 고려하여 보완해주어야 한다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>4 차 과 제 보 고 서</b>		
	<b>프로젝트 명</b>	장애물 회피 라인트레이싱	
	<b>팀 명</b>	2 조	
	Confidential Restricted	내용 전체 수정	2017-11-20

## 2 기본 아이디어

### 2.1 H/W 디자인 방향

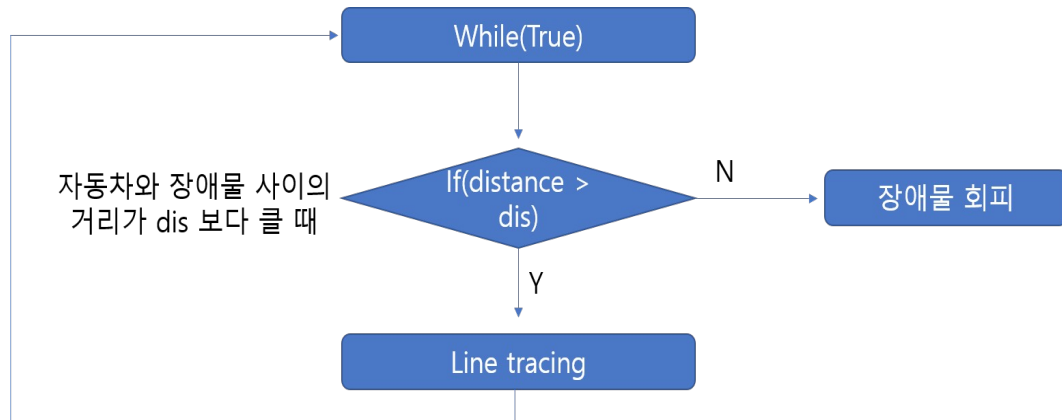


장애물을 인식하기 위해 자동차 앞부분에 초음파 센서를 달았고 라인 트레이싱을 위해 자동차 아래부분에 적외선 IR 광 추적 센서를 달았다.



적외선 IR 광 추적 센서는 바닥의 색깔을 인식하여 바닥의 색깔이 흰색일 때 불이 들어오고 검정색일 때에는 불이 들어오지 않는다. 이 센서를 이용해 들어오면 1을 리턴하고 들어오지 않으면 0을 리턴한다.

## 2.2 S/W 디자인 방향



위의 그림과 같이 자동차와 장애물 사이의 거리가 dis 보다 클 때에는 라인 트레이싱 코드를 실행하고 그렇지 않을 때에는 장애물 회피 코드를 실행하도록 소프트웨어를 디자인했다.

## 3 수행 내용

### 3.1 장애물 인식 및 회피

Raspberry Car 사이트에서 제공되는 ultraModule.py 모듈 안의 getDistance() 함수를 통해 자동차 앞의 초음파 센서에서 지속적으로 자동차와 앞에 위치한 장애물까지의 거리인 Distance 값을 받아온다. 또 자동차가 장애물과 얼마만큼의 거리에 도달해야 멈출 것인지 미리 Dis 라는 변수에 지정해 놓는다. Dis 값과 Distance 값을 비교하여 만약 Distance 가 Dis 보다 크다면 라인트레이싱을 진행하며 그렇지 않다면 장애물을 만난 것으로 인식하여 회피하게 된다.

장애물을 회피하는 방법은 go\_any.py 모듈의 go\_forward 함수와 TurnModule.py 모듈의 rightPointTurn, leftPointTurn 함수를 사용하여 만약 반시계 방향으로 출발하였다면 ‘오른쪽 포인트 턴 → 앞으로 이동 → 왼쪽 포인트 턴 → 앞으로 이동 → 왼쪽 포인트 턴 → 앞으로 이동 → 라인으로 복귀’ 하는 방법을 사용하였으며 시계 방향에서는 포인트 턴의 방향을 반전시켜 진행하였다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>4 차 과 제 보 고 서</b>		
	<b>프로젝트 명</b>	장애물 회피 라인트레이싱	
	<b>팀 명</b>	2 조	
	Confidential Restricted	내용 전체 수정	2017-11-20

## 3.2 라인트레이싱 및 곡률 코스 처리

TrackingModule.py 모듈에 5 방향 트래킹 센서의 값을 GPIO.input 함수로 받아서 5 개의 센서 값을 각각 String 형으로 변환하고 더한 값을 return 하는 getSensorResult 함수를 추가하였다. (ex '11001'이라면 Centerled와 rightlessled 가 라인 위에 위치하고 있다는 뜻이다)

5 방향 트래킹 센서 상태의 경우의 수를(ex) '11011' 담고 그 경우가 되었을 때 설정할 왼쪽과 오른쪽 바퀴의 속도, 총 3 가지 정보를 담은 2 차원 배열 result 를 만들었다. for 문을 사용하여 getSensorResult 로 얻어온 현재 트래킹 센서 값과 경우의 수(result[i][0])를 result 의 길이만큼 비교해 만약 같다면 현재 자동차의 왼쪽, 오른쪽 바퀴의 속도 값을 저장해 놓은 속도 값(result[i][1], result[i][2])으로 바꾼다. 이렇게 하면 라인트레이싱을 하게 되고, 만약 자동차가 라인 밖으로 나간다면 즉, getSensorResult 의 리턴 값이 '11111'상태가 되게 되면 자동차의 출발 방향(시계 or 반시계)에 따라 오른쪽 혹은 왼쪽 바퀴 값을 0으로 변경하여 제자리에서 돌면서 다시 라인을 검색한다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>4 차 과제 보고서</b>		
	<b>프로젝트 명</b>	장애물 회피 라인트레이싱	
	<b>팀 명</b>	2 조	
	Confidential Restricted	내용 전체 수정	2017-11-20

## 4 향후 추진계획

### 4.1 향후 계획의 세부 내용

#### (1) H/W

문제점 : 양쪽 모터의 출력이 일정하지 않아 실행시켰을 때 이동방향이 한 쪽으로 치우치는 상황이 발생하였다. 속도의 차이를 측정하고 프로그래밍을 통해 두 모터의 출력을 같게 조절하려고 했으나 라즈베리 파이 재부팅 때마다 출력량이 또 바뀌게 된다.

해결방안 : 출력이 비슷한 모터들을 찾아 현재의 것과 바꿔 장착한다.

문제점 : 자동차가 실행될 때 사용되는 전력의 양이 많아 건전지가 빨리 소모되어 속도가 느려지는 현상이 생겼다. 과제를 실행시킬 때마다 속도가 달라지게 되어 입력 값을 예측하기 힘들어졌다.

해결방안 : 연습용 건전지와 과제 검사용 건전지를 따로 사용하여 건전지가 최대로 충전이 되었을 때 검사 받을 수 있도록 하였다.

#### (2) S/W

문제점 : 라인 트레이싱하는 과정에서 자동차의 움직임이 부드럽지 못하고, 비틀거리며 이동하는 현상이 발견되었으며, 이로 인해 장애물을 회피하는 과정에서 매번 멈추는 각도가 달라져 테스트 결과가 바뀌게 되었다.

해결방안 : 라인 트레이싱을 할 때 더 많은 경우의 수를 고려하지 못하여 정해진 코드로만 방향을 바꾸기 위해 꺾는 과정이 반복되면서 과도하게 비틀거리는 것이므로 라인을 벗어나는 정도에 따라 꺾어야 할 각도를 순차적으로 줄여 안정적으로 주행할 수 있도록 한다

문제점 : 라인 트레이싱을 하기 위해 GPIO 로 입력되는 5 방향 트래킹 센서 값에 대한 모든 경우의 수를 직접 입력하고, 또한 경우의 수에 따라 속도를 지정해 주었는데 일일이 지정해주는 것이 비효율적이었다.

해결방안 : 팀원들과 상의하여 간편하고 효율적으로 고쳐 나가는 알고리즘을 생각해보고 수정할 것이다.



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>4 차 과제 보고서</b>		
	<b>프로젝트 명</b>	장애물 회피 라인트레이싱	
	<b>팀 명</b>	2 조	
	Confidential Restricted	내용 전체 수정	2017-11-20

## 5 애로 및 건의사항

우선 자동차의 부품 불량인 경우가 많았다. 우리 조나 다른 조원들을 보았을 때 각 조에서 부품불량이 적어도 한 번 씩은 나왔던 것 같다. 많은 양의 부품을 배부하는 과정에서 불가피한 경우일 수 있지만 생각보다 불량인 학생들이 많아서 조립할 때나 과제를 할 때 약간의 어려움을 겪었다.

또한 조립을 할 때 바퀴를 조이는 정도나 자동차에 연결된 선의 위치에 따른 무게 변화에 따라 약간의 영향에도 결과가 천차만별로 달라지기 때문에 같은 코드에도 매번 다른 결과가 나오기 때문에 불안정하고 실력을 평가하기에는 한계가 있다고 생각한다. 마지막으로 주행하는 트랙 환경은 바뀌지 않는 통제 변인이어야 하는데 트랙을 직접 만들어 실험하다 보니 어떤 트랙은 라인이 매끄럽지 못하거나 일반적 라인보다 두껍거나 아크릴 판이 들리는 등의 문제가 있었다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>4 차 과 제 보 고 서</b>		
	<b>프로젝트 명</b>	장애물 회피 라인트레이싱	
	<b>팀 명</b>	2 조	
	Confidential Restricted	내용 전체 수정	2017-11-20

## 6 회의록

<b>일 시</b>	11 월 17 일	<b>차수</b>	1 차
<b>장 소</b>	무한 상상실		
<b>참석자</b>	서명희, 서형석, 송지영		
<b>불참자</b>			
<b>안 건</b>	자동차 도로 주행 연습		
<b>회의내용</b>	자동차를 계속 트랙에 돌리면서 문제점에 대해 생각해보고 코드를 고쳐나갔다.		
<b>결과물</b>			