


# 창업연계공학설계입문

프로젝트 명	미로찾기
팀 명	갈피를 못잡조~
문서 제목	5 차 과제 보고서

Version	1.3
Date	2017/12/16

팀원	박정규(조장) / 20171624
	박태범 / 20171626
	남경태 / 2017*****
지도교수	한광수 교수
분반	5 분반

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>5 차 과제 보고서</b>		
		프로젝트 명	미로찾기	
		팀 명	갈피를 못잡죠~	
		Confidential Restricted	Version 1.3	2017/12/11


#### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 공학설계입문 수강 학생 중 프로젝트 “미로찾기”를 수행하는 팀 “갈피를 못잡죠~”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “갈피를 못잡죠~”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

## 문서 정보 / 수정 내역


<b>Filename</b>	5 차과제-프로젝트명.doc
원안작성자	박정규, 박태범, 남경태
수정작업자	박정규, 박태범, 남경태

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2017-11-29	박정규	1.0	최초 작성	최초 작성.
2017-12-6	박정규	1.1	내용 추가	코드 알고리즘 수정. 회의록 추가.
2017-12-11	박정규	1.2	결과 추가.	결과. 회의록 추가.
2017-12-16	박태	1.3	느낀점 추	느낀점 추

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>5 차 과제 보고서</b>	
		프로젝트 명	미로찾기
		팀 명	갈피를 못잡죠~
		Confidential Restricted	Version 1.3 2017/12/11

## 목 차

<b>1</b>	<b>과제 개요.....</b>	<b>4</b>
<b>2</b>	<b>요구 사항 분석.....</b>	<b>5</b>
	2.1 요구 사항.....	5
<b>3</b>	<b>프로그램 설계.....</b>	<b>6</b>
	3.1 프로그램 알고리즘.....	6
	3.2 모듈 구조.....	6
<b>4</b>	<b>프로그램 구현.....</b>	<b>7</b>
	4.1 프로그램 구현.....	7
<b>5</b>	<b>평가 결과 및 결론.....</b>	<b>8</b>


	<b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>5 차 과제 보고서</b>		
		프로젝트 명	미로찾기	
		팀 명	갈피를 못잡죠~	
		Confidential Restricted	Version 1.3	2017/12/11

## 1 과제 개요

### 미로 찾기 과제

#### 1. 미로를 탈출한다.

- 미로는 검정색 테이프의 선으로 이루어져 있으며, 구동체는 테이프의 위를 주행해야 한다.
- 미로는 직선 부분과 교차로 부분으로 이루어져 있으며, 각 교차로는 직각으로 두 선이 만나고 있다.
- 미로를 탈출하기 위해 직진, 회전 등 이동 기능과 라인을 판단할 수 있는 기능을 갖춰야 한다.
- 제한 요소로서는 배터리의 충전 상태가 가장 중요하므로, 이에 대한 해결 방안을 갖춰야 한다.

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>5 차 과제 보고서</b>		
		프로젝트 명	미로찾기	
		팀 명	갈피를 못잡죠~	
		Confidential Restricted	Version 1.3	2017/12/11

## 2 요구 사항 분석

### 2.1 요구 사항 및 기능.

미로 찾기 미션을 해결하기 위해 구동체가 필요한 기능으로는 처음에 구동체를 초기화하는 세팅 기능, 직진 회전 정지 등 구동체를 이동 시키는 이동 관련 기능, 라인의 상태를 인식하고 판단하는 인식 기능이 있다.

이 기능을 구현하기 위해 각 기능별로 하나의 모듈을 만들어서


1. 세팅 모듈
2. 이동 모듈
3. 라인 인식 모듈

이라는 세 개의 모듈을 만들려 한다.

세팅 모듈에는 모터 세팅과 센서 세팅 기능을 구현한다.

이동 모듈에는 직진, 회전, 미로 찾기 세 개의 기능을 구현한다.

라인 인식 모듈에서는 라인을 센서로 받아들여 리스트로 리턴하는 기능을 구현한다.

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>5 차 과제 보고서</b>		
		프로젝트 명	미로찾기	
		팀 명	갈피를 못잡죠~	
		Confidential Restricted	Version 1.3	2017/12/11

## 3 프로그램 설계

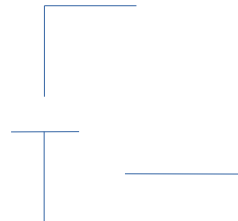
### 3.1 프로그램 알고리즘

오른손 법칙 : 미로를 찾아나가는 방법으로, 오른쪽 벽에 손을 대고 출구까지 손을 떼지 않고 나아가는 방법.

미로의 구성은 직선과 교차로로 이루어져 있다. 구동체가 직선 구간에 존재할 때에는 직진만 하고, 구동체가 교차로 구간에 들어갔을 때에는 교차로의 종류에 따라 각자 다른 회전을 한다.

1. 오른쪽 회전이 가능한 2 거리.  
구동체의 오른쪽 센서가 감지되면 무조건 우회전한다.

1.



2. 오른쪽 회전이 가능한 3 거리.  
구동체의 오른쪽 센서가 감지되면 무조건 우회전한다.

2.



3. 4 거리.  
구동체의 오른쪽 센서가 감지되면 무조건 우회전한다.

3.



4. 왼쪽 회전만 가능한 3 거리.  
왼쪽 회전이 가능하나 오른쪽 우선이므로 직진한다.

4.

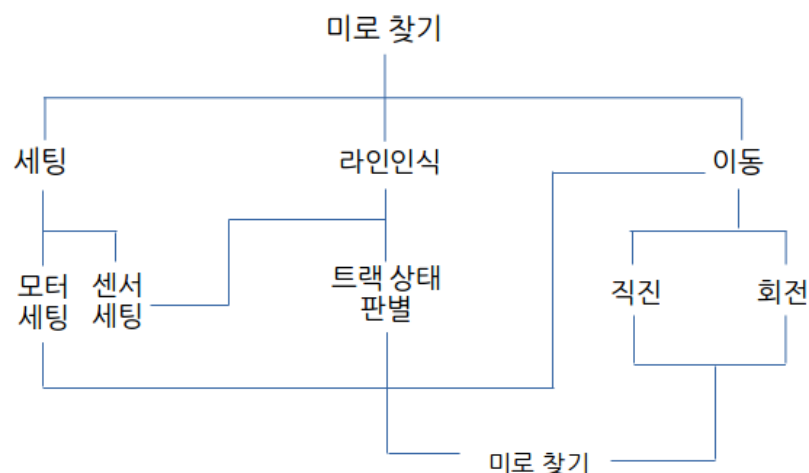



5. 왼쪽 회전만 가능한 2 거리.  
오른쪽이 아예 존재하지 않음으로 좌회전 한다.

5.



### 3.2 모듈 구조



	국민대학교 소프트웨어학부 창업연계공학설계입문	5 차 과제 보고서	
		프로젝트 명	미로찾기
		팀 명	갈피를 못잡죠~
		Confidential Restricted	Version 1.3      2017/12/11

## 4 프로그램 구현

### 4.1 프로그램 구현

```

#####
def goingtrack():
    """
    이 함수는 5 방향 센서중, 직진에 관한 센서 가운데 3개의 값을 리스트로 반들어서 돌려준다.
    :return: 5방향 센서의 상태에 관한 문자열.
    """
    L2s = GPIO.input(setting.L2)
    L1s = GPIO.input(setting.L1)
    Cs = GPIO.input(setting.C)
    R1s = GPIO.input(setting.R1)
    R2s = GPIO.input(setting.R2)

    Tracklist = [L2s, L1s, Cs, R1s, R2s]
    return Tracklist
#####

```

센서를 받아오는 goingtrack 함수의 경우, 0~4 번 까지의 센서의 정보 값을 를 위치에 따라 L2s, L1s, Cs, R1s, R2s 로 할당한 뒤, 이 리스트를 리턴함.



```
# =====
def lineTrace(speed):
    t = Tracking_sensor.goingtrack()
    print(t)
    if not t[2] and not t[4]:
        stop()
        go_any(forward, speed, speed)
        time.sleep(0.4)
        PointTurn('right', speed)
        time.sleep(0.5)
        while Tracking_sensor.goingtrack()[3]:
            continue
        stop()
    elif t[1] and not t[2] and t[3]:
        go_any(forward, speed, speed)
    elif not t[1]:
        go_any(forward, speed, speed/3)
    elif not t[3]:
        go_any(forward, speed/3, speed)
    elif t == [1,1,1,1,1]:
        stop()
        time.sleep(0.5)
        PointTurn('left', speed)
        while Tracking_sensor.goingtrack()[3]:
            continue
        stop()
    time.sleep(0.1)
```


# =====

라인 주행을 하는 lineTrace 함수는 t 라는 변수에 goingtrack 의 값을 받아 트랙의 상태를 받은 뒤, 이에 따라 상황을 판단해서 주행한다.

다음의 설명에서 ?는 해당 번호, 위치의 센서 값은 전혀 신경 쓰지 않는다는 것이다.

1. [? ? 0 ? 0] : 가운데 [2]번 센서와 우측 끝 [4]번 센서가 인식되었으므로, 상황은 직진 주행 중 네거리 교차로나 우회전이 가능한 3거리 교차로, 혹은 우측으로 꺾인 2거리 교차로 중 한 가지 경우이고, 이 세 경우 모두 오른손 법칙에 따라 우회전에 해당한다. 따라서 우회전을 실행한다.
2. [? 1 0 1 ?] : 정 가운데 센서만 켜진 상태, 이 상황은 구동체가 정확히 라인의 정 가운데 있는 상황이므로, 직진 주행만 하면 된다.
3. [? 0 ? ? ?] : 직진 주행 중이라는 가정 하에, 현재 왼쪽으로 조금 치우쳐진 상태이므로, 왼쪽 바퀴의 속도를 조절해 주행 방향을 보정한다.
4. [? ? ? 0 ?] : 직진 주행 중이라는 가정 하에, 현재 오른쪽으로 조금 치우쳐진 상태 이므로, 오른쪽 바퀴의 속도를 조절해 주행 방향을 보정한다.
5. [1 1 1 1 1] : 현재 자동차가 검은 라인을 읽고 있지 못하다는 뜻으로, 이 경우는 두 가지 상황에 해당한다.  
첫 번째는 자동차가 직선 코스의 끝에 다다른 막다른 길이라 U 턴을 해야 하는 상황이다. 이때 U 턴도 이 미로가 선 위를 주행하는 것이 아니라 실제 미로에서 오른손을 벽에 짚고 움직인다면, 왼쪽으로 U 턴 해야 한다.  
두 번째는 자동차가 왼쪽 라인만 있는 2거리 교차로를 건너 뛰고 직진 주행하여 흰 바닥 위에 있는 경우이다. 이 경우는 왼쪽으로 돌아야 한다.



	국민대학교 소프트웨어학부 창업연계공학설계입문	5 차 과제 보고서	
		프로젝트 명	미로찾기
		팀 명	갈피를 못잡죠~
		Confidential Restricted	Version 1.3      2017/12/11


**try:**

```
# 초기 셋팅
setting.LeftPwm.start(0)
setting.RightPwm.start(0)
while True:
    going.lineTrace(40)
```

**except KeyboardInterrupt:**

```
# the speed of left motor will be set as LOW
GPIO.output(setting.MotorLeft_PWM, GPIO.LOW)
# left motor will be stopped with function of ChangeDutyCycle(0)
setting.LeftPwm.ChangeDutyCycle(0)
# the speed of right motor will be set as LOW
GPIO.output(setting.MotorRight_PWM, GPIO.LOW)
# right motor will be stopped with function of ChangeDutyCycle(0)
setting.RightPwm.ChangeDutyCycle(0)
# GPIO pin setup has been cleared
GPIO.cleanup()
```


# =====  
메인 메서드에서는 while 문 안에 lineTrace 를 원하는 속도로 호출 함으로써, 무한 루프를 돌게 하는 것이다.

	국민대학교 소프트웨어학부 창업연계공학설계입문	5 차 과제 보고서	
		프로젝트 명	미로찾기
		팀 명	갈피를 못잡죠~
		Confidential Restricted	Version 1.3      2017/12/11

## 5 평가 및 결론


실제 구동 결과 같은 팀원보다 많은 라인이탈이 있었다. 우선 주행을 할 때 약간 한쪽으로 치우쳐서 가는 경우가 있었다. 그렇다 보니 코드를 짤 때 고려했던 경우를 벗어나 원하는 대로 움직이지 않을 때가 있었다. 또 회전을 할 때도 너무 빠르게 회전을 하여 다시 라인을 인식하지 못하고 계속 돌기도 하였다. 구동체의 속도를 줄이고 회전 사이에 stop()을 추가하여 겨우 완주는 하였지만 감점은 피할 수 없었다. 코드 자체가 한정된 경우의 수로 구동체를 작동시키는 것이었고 그 코드도 같은 팀원의 구동체에 맞추어서 짜여진 것이기 때문에 나의 구동체에 최적화를 시켰어야 되는데 그 작업이 완벽하지 않아 라인 이탈이 생긴 것 같아 아쉬움이 남았다.

이번 과제는 창업공의 마지막 과제인 만큼 이전에 구현했던 코드들을 이해하고 응용할 수 있어야 하였다. 팀원들과 다양한 방법으로 코드를 구현해보고 실험해보며 더 나은 방법을 찾아갔다. 최종적으로 미로를 모두 탈출했을 때 완벽하지는 않지만 창업공의 과제를 모두 마쳤다는 성취감을 느낄 수 있었다. 이 과제를 모두 혼자 했다면 완주도 힘들었을 것이라고 생각한다. 창업공 과제를 하며 팀프로젝트의 이점과 협동의 중요성을 많이 느끼게 되었다.


	<b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>5 차 과제 보고서</b>		
		프로젝트 명	미로찾기	
		팀 명	갈피를 못잡죠~	
		Confidential Restricted	Version 1.3	2017/12/11

## 6 회의록

일 시	11/29 ~ 11/30	차수	1
장 소	7 호관 무한 상상실		
참석자	박정규, 박태범, 남경태		
불참자	없음.		
안 건	프로그램의 기본 구조 구상.		
회의내용	<p>1. 프로그램의 기본 구조.  이 전 과제에서 라인 트레이싱 부분은 모두 다 그대로 채용하고, 회전이 필요한 부분을 판단하는 코드를 일부 추가하면, 정상적으로 구동할 수 있을 것 이라고 추측된다.</p> <p>2. 작업 분담.  수정하는 부분을 각자 분담하여 수정하기로 함.  이동 모듈 : 박태범  센싱 모듈 : 남경태  보고서 초안 작성 : 박정규</p> <p>3. 추후 실행 날자 결정  코드를 완성할 시간의 부족으로, 코드 실행과 실험은 다음 날로 미루기로 결정함.</p>		
결과물	<p>이 전 과제의 라인트레이싱에서는 0 번,4 번 센서를 모두 진행 보정에 사용했으나, 이번 과제에서는 교차로에서 회전을 위한 판단에 쓰여야 함으로, 이 때문에 교차로 중 어느 교차로는 회전하지만, 어느 교차로는 읽지 못하고 그냥 직진하는 문제를 보임.  코드의 뼈대부터 수정이 필요함.</p>		

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>5 차 과제 보고서</b>		
		프로젝트 명	미로찾기	
		팀 명	갈피를 못잡죠~	
		Confidential Restricted	Version 1.3	2017/12/11

<b>일 시</b>	12/4	<b>차수</b>	2
<b>장 소</b>	7 호관 자율주행 스튜디오		
<b>참석자</b>	박정규, 박태범, 남경태		
<b>불참자</b>	없음.		
<b>안 건</b>	프로그램 수정.		
<b>회의내용</b>	<p>1. 프로그램의 수정 방안.  좌우의 양쪽 끝 센서는 라인트레이싱을 위한 보정에서 제외하고, 1 번 센서와 3 번 센서가 감지 되었을 때 좌우 속도 차이를 더 크게 만들어서 보정하도록 수정.  좌우 양쪽 끝 센서는 오로지 교차로 판단을 위해서 이용되도록 수정.</p> <p>2. 프로그램 시험 구동.</p>		
<b>결과물</b>	우회전이 필요한 교차로에서는 제대로 회전하지만, 좌회전이나 U 턴을 필요로 하는 교차로에서는 그냥 직진하는 문제를 발견. 알고리즘에 대한 체계적 구현이 필요하다는 생각을 하게 되었고, 직진, 턴, 정지, 세팅을 제외한 모든 기능 초기화.		

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>창업연계공학설계입문</b>	<b>5 차 과제 보고서</b>		
		프로젝트 명	미로찾기	
		팀 명	갈피를 못잡죠~	
		Confidential Restricted	Version 1.3	2017/12/11

일 시	12/6	차수	3
장 소	7 호관 무한 상상실		
참석자	박정규, 박태범, 남경태 + 다른 분반의 김남재 (도움)		
불참자	없음.		
안 건	프로그램의 기본 구조 구상.		
회의내용	<p>1. 우수법(오른손 법칙)에 대한 기본적 생각  초기에 우수법에 대한 우리의 이해는, 그저 오른쪽 방향의 회전을 우선시 한다 정도로 이해하였음. 그에 따라 우회전의 코드는 제대로 판단하도록 하는 if 문과 실행문을 만들었음. 하지만, 좌회전에 대해서는 우회전과 반대 정도로, U 턴은 우회전과 거의 같이 구현하였고, 이를 판단하는 if 문은 실제로 거의 틀린 부분들이 많다는 것을 발견함.  우수법을 생각해본 결과, 미로를 실제 입체척 미로라고 생각했을 때, U 턴의 경우는 좌회전과 같은 방향으로 움직였고, 이를 이용하면 판단도 직선 구간, 우회전 구간을 제외하면 모두 좌회전으로 통일할 수 있다는 생각에 도달함.</p> <p>2. 회전 시간 조절 방법  다른 분반의 김남재 학우의 도움으로, 회전 시간 조절에서 while 문과 continue 를 이용해 조절할 수 있다는 것을 알게 됨.</p>		
결과물	코드를 수정함. 일부분 이탈을 제외하고 정상적으로 완주함.		