제16회 임베디드SW경진대회

개발일지 [주니어 임베디드SW 챌린저]

작품명 : 택배비는2000원

팀 명 : 동네닭.zip

팀번호	17014
대표소속	부천정보산업고등학교
팀원이름	김호영,이종인,윤한나
지도교사(멘토) 이름	김윤영
시연동영상(URL)	https://youtu.be/jm7WTWpTq8o

임베디드 소프트웨어 경진대회
The World Embedded Software Contest

개발일지 목차

번호	연구기간	연구내용
1	08월 18일 ~ 08월 26일	로봇 베이스 제작
		승객 하차장치 연구
		승객 하차장치 제작
2	09월 01일 ~ 09월 02일	하드웨어 개선
3	09월 10일 ~ 09월 17일	색상 저장방법
4	09월 18일 ~ 현재	이동경로 검토 및 라인트레이싱
5	09월 18일 ~ 현재	최단거리 계산

프로젝트1 (하드웨어 구성 및 안전한 로봇 착지1)	
프로젝트 기간 및 목표	
기간	08월 18일 ~ 08월 26일
목표	승객을 목표지점까지 정확하고 안정적이게 놓는다.

프로젝트 활동내용

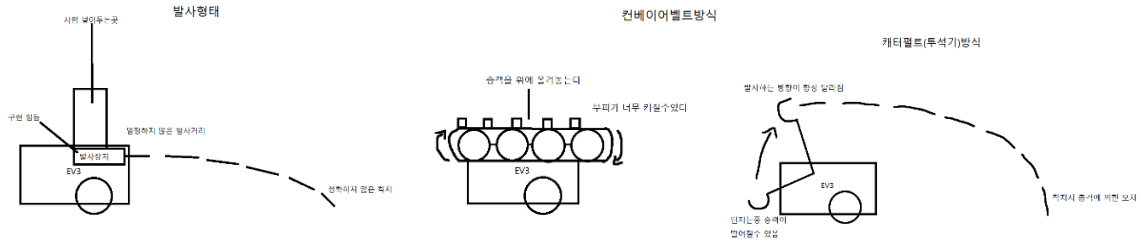
방법 :발사,컨베이어벨트,캐터펄트(투석기),미끄럼틀, 기어활용 등의 방법을 논의했다

발사의 단점 : 각도,모터상태,착지 등의 변수를 항상 고려해야 하고,
만약 변수가 생겨 수정을 해야 할 때 수정하기가 어렵다.

컨베이어벨트의 단점 :공간을 꽤나 많이 사용할 것이고, 설치하는 장소도 마땅치 않다.*

캐터펄트(투석기)의단점: 정확하게 발사각도를 계산하지 않는 한은 레고로봇이
제자리에 착지 할 것이라는 보장도 없으며 공간을 꽤나 많이 차지할 것이다.

기어활용의 단점 : 정확하고 확실한 설계가 필요하며 약간의 오차도 있어서는 안 된다.
또한 내부에 문제가 생겼을 때 수정하기도 어렵고 사용하기에 복잡하다.



이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

지도교사 확인 및 의견

승객을 정확하게 하차 시키는 하드웨어를 제작하는 것은 프로그래밍 못지 않게 어려운 작업일 수 있음.	(서명)

프로젝트1 (하드웨어 구성 및 안전한 로봇 착지2)	
프로젝트 기간 및 목표	
기간	08월 18일 ~ 08월 26일
목표	승객을 목표지점까지 정확하고 안정적이게 놓는다.

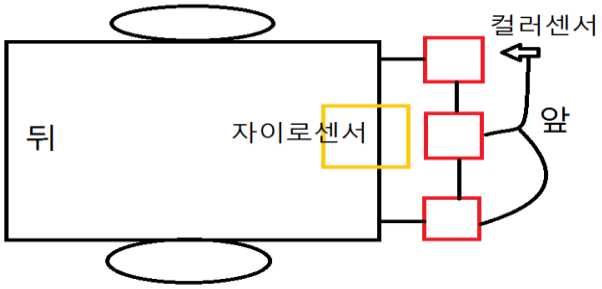
프로젝트 활동내용

우리가 선택한 방법:

우리가 선택한 방법은 미끄럼틀을 이용한 레고사람떨어뜨리기다.

미끄럼틀을 이용한 떨구기도 높은 곳에서 떨구면 튕겨서 다른 곳으로 갈 수 있다는 것이 단점이지만, 이 방식은 그렇게 고려해야 할 것이 많지 않고, 공간을 많이 차지하는 것도 아니면서 떨어지는 부분에 벽을 설치하여 제자리에 착륙할 수 있도록 유도하는 등으로 벽을 약간만 추가하면 되기 때문에 이 방법을 선택하게 되었다.

또한 우리 동네닭.zip 조는 라인 트레이싱 및 색깔 저장을 하기 위하여 전방에 컬러센서 세 개를 달았으며, 정확한 직진을 위하여 자이로센서 하나를 컬러센서 바로 뒤에 달았다.



이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

선생님께완성본을 보여드렸더니 너무 높은 곳에서 떨어지기 때문에 승객이 착지하기 어렵고, 미끄럼틀의 방향이 안쪽을 향하고 있어서 바깥쪽으로 튕겨 나갈 수 있다고 하였다.

이러한 점을 보완하기 위해 다음 프로젝트는 미끄럼틀을 더 수정하고 다른 문제가 없는지 확인해 볼 것이다.

지도교사 확인 및 의견

아이디어는 좋으나 높은 위치에서 승객을 낙하시키기 때문에 의도된 위치에 착지할 가능성이 낮음.	(서명)

프로젝트2 (하드웨어 보완1)

프로젝트 기간 및 목표

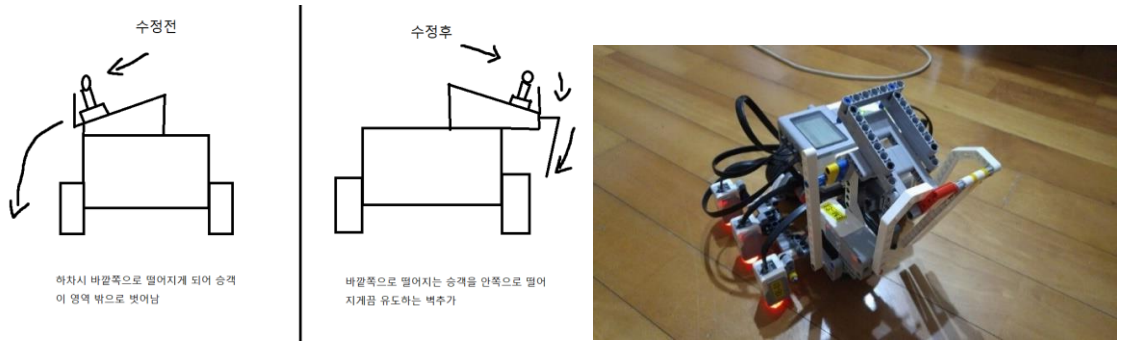
기간	09월 01일 ~ 09월 02일
목표	최대한 안정적이고 정확하게 떨어트리고 다른 단점을 보완한다.

프로젝트 활동내용

미끄럼틀을 고치면서 실험을 하다 보니 자이로센서가 붙어있는 쪽에도 문제가 있는 것 같기에 수정을 하게 되었다.

미끄럼틀 :

초반에 만든 미끄럼틀의 단점을 보완하기 위해 미끄럼틀을 원래 끼워놓은 자리의 반대편에 방향을 본체 쪽으로 전환해서 부착했다. 그리고 로봇이 떨어지는 부분에 정확하게 떨어지도록 ㄷ자 형태로 벽을 설치하여 보완하였다. 그 결과 로봇이 제자리에, 우리가 원하는 범위 내에 떨어지게 되었다.



이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

지도교사 확인 및 의견

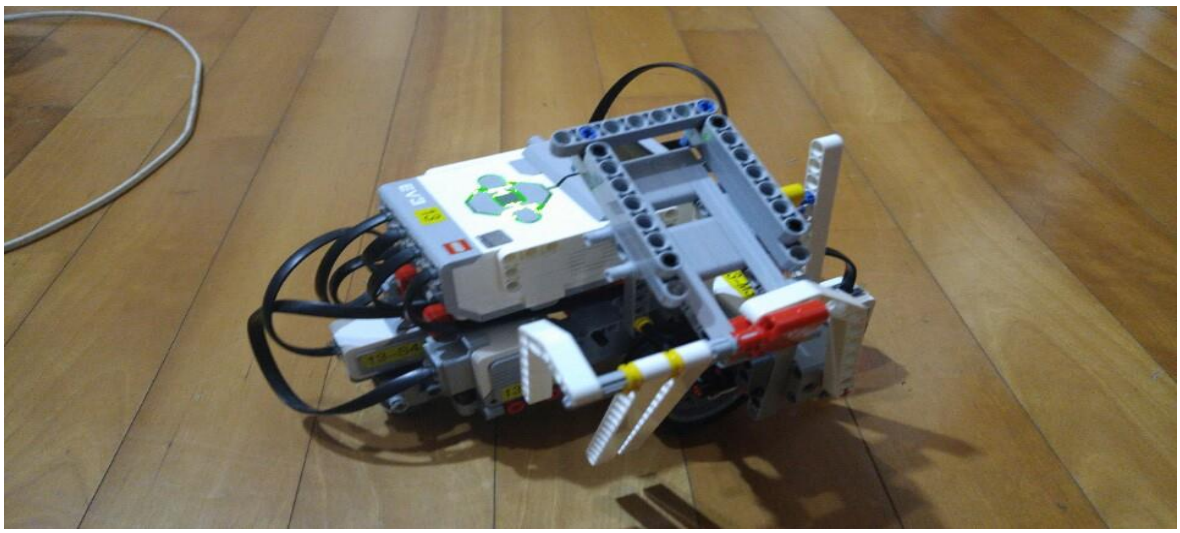
테스트 결과 승객의 하자 지점이 매우 안정적이었음. 낙하 가이드 구조물이 매우 중요한 역할을 함.	(서명)

프로젝트2 (하드웨어 보완2)	
프로젝트 기간 및 목표	
기간	09월 01일 ~ 09월 02일
목표	최대한 안정적이고 정확하게 떨어트리고 다른 단점을 보완한다.

프로젝트 활동내용

자이로센서 :

초반에는 자이로센서를 앞부분 남는 공간에 방향을 반대로 끼워서 넣었기 때문에 값을 반대로 측정하는 문제점과 문제가 생겼을 시에 수리를 하는 것에 차질이 생길 수도 있다고 판단하여 자이로센서를 기계 뒤쪽에 바르게 설치하였다.



이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

미끄럼틀의 높이는 크게 변하지 않은 것에는 반성하지만 그럼에도 로봇이 똑바로 정확하게 떨어지기 때문에 좋은 발전이라 생각한다.

하드웨어 구성은 끝났으니 다음 프로젝트는 소프트웨어에 대해 프로젝트를 진행 할 것이다.

지도교사 확인 및 의견

특이사항 없음	(서명)

프로젝트3 (색상저장 방법)

프로젝트 기간 및 목표

기간	09월 10일 ~ 09월 17일
목표	색상을 효율적으로 저장한다.

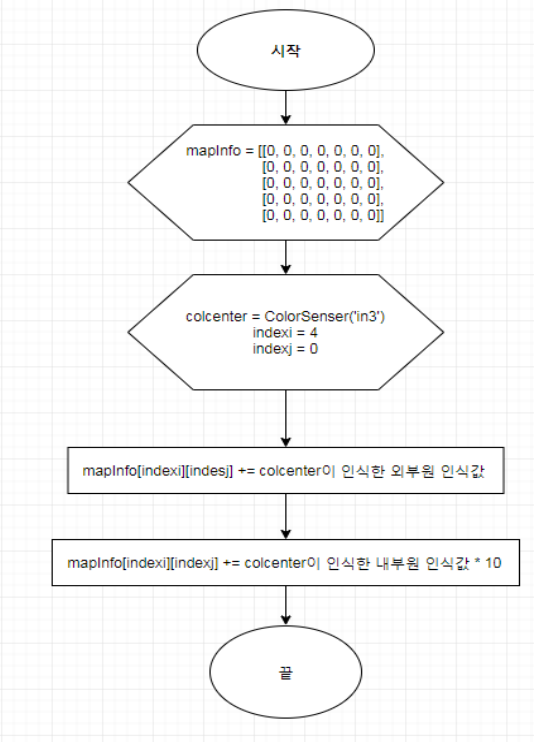
프로젝트 활동내용

▽아래와 같이 5X 7짜리 색상 저장 리스트를 만든다.

MapInfo = [[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0]]

컬러센서가 바깥 원에 있는 색깔을 인식해서 그 값을 그대로 mapInfo에 있는 리스트에 저장하고

안쪽 원에 있는 색깔은 인식한 값 * 10을 해서 그 좌표에 있는 mapInfo에 더한다.



이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

알고리즘을 최대한 간단히 표현하려고 하다 보니 세세한 부분을 넣지 못한 것에 대해 반성해야 할 것 같다. 또한 아직 이론일 뿐이라 추후 바뀔 수 있다는 것에 주의를 해야 할 것이다.

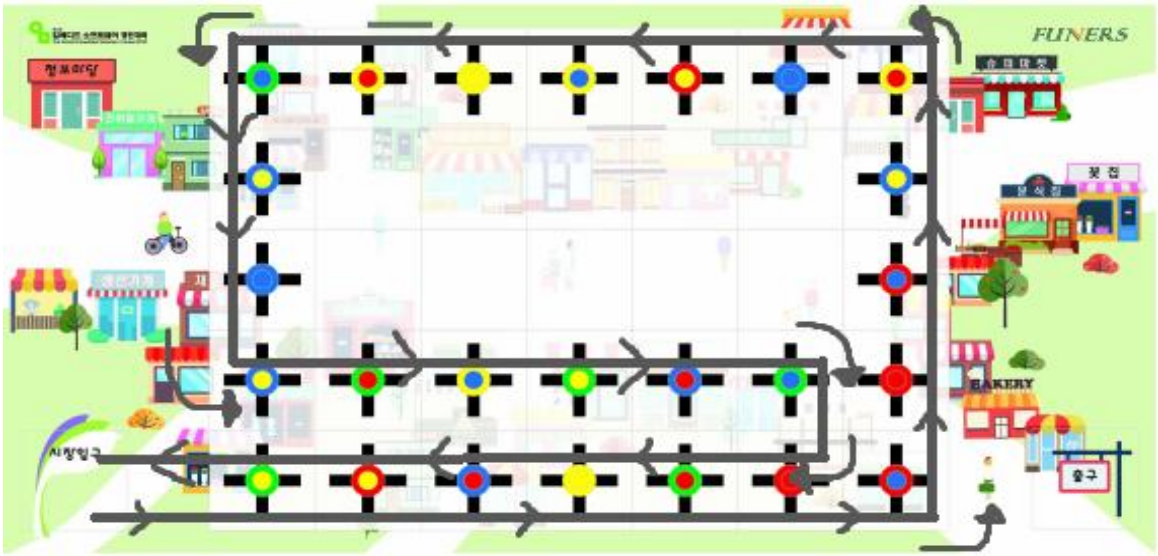
색을 저장하는 알고리즘을 만들었으니 다음에는 라인 트레이싱에 대해 프로젝트를 진행 할 것이다.

지도교사 확인 및 의견

각 상점의 정보는 튜플(파이썬에서 제공하는 자료형)을 이용하면 위의 방법보다 쉽게 자료를 저장할 수 있다. 또는 상점 클래스를 만들어서 저장하는 것도 생각해 볼만 하다.	(서명)

프로젝트4 (이동경로 검토 및 라인트레이싱)	
프로젝트 기간 및 목표	
기간	09월 18일 ~ 현재진행형
목표	빠르고 정확하게 모든 상점의 색을 인식하게 한다.

프로젝트 활동내용



아직 확실히 정하지는 않았지만 현재 생각한 방법은 각 조건이 되는 변수들을 선언한 다음에 조건이 모두 충족되면 자이로 센서를 이용해 직진을 하거나 90도를 회전하는 식으로 위에 그림처럼 맵을한바퀴 돌면서 색을 저장할 것이다. 코드를 작성할 때에는 각 역할을 수행하는 함수를 따로 지정해서 사용할 것으로 생각된다.

이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

아직 실험을 해보지 않은 상태라서 그냥 이론만 내세우는 것에 대해 반성을 해야 할 것 같으면서 실현 가능성이 있는 것 에 희망을 가져도 될 것 같다.

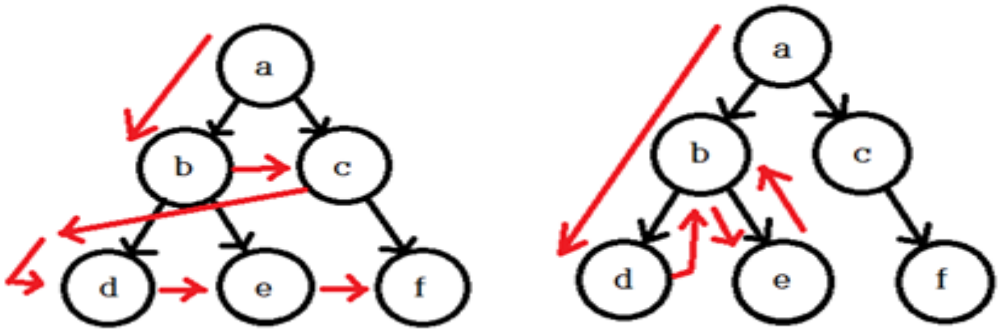
이 프로젝트와 더불어 최단거리를 계산하는 방법을 구하는 프로젝트도 같이 진행할 예정이다.

지도교사 확인 및 의견

특이사항 없음.	(서명)

프로젝트5 (최단거리 계산)	
프로젝트 기간 및 목표	
기간	09월 18일 ~ 현재진행형
목표	가장 효율적인 계산 방법을 선택한다.

프로젝트 활동내용



△너비우선△깊이우선

너비우선 방식과 깊이우선 방식을 이용하여 최단거리를 구하려 한다.

먼저 정보마당에서 받은 추천 색상과 안쪽 색상이 같은 상점을 모두 찾아낸 다음 출발지점에서 가장 가까운 지점을 먼저 선택하고 그다음 가까운 선택하는 식으로 4가지의 가게를 모두 찾은 다음 모든 경로를 계산해 본 다음 가장 짧은 거리를 구할 것이다. 물론 이때 프로젝트 3에서 만든 좌표를 이용할 것이기 때문에 시작 좌표는 (0,0)으로 하고 (1,1)~(1,4), (2,1)~(2,4)의 좌표는 피해가도록 코딩을 할 것이다.

이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

아직 가지고 있는 지식이 부족해서 정확하게 어떤 방법으로 최단거리를 구해야 하는지 정하지 못한 것을 반성하고 싶다.

만약 라인트레이싱, 최단거리 구하기 이 프로젝트들이 끝나면 다음에는 내가 구한 최단거리를 그대로 로봇이 가서 레고를 떨어트리는 방법에 대해 연구를 하고 싶다.

지도교사 확인 및 의견

그리디(Greedy) 알고리즘을 생각하고 있지만, 그리디가 최적의 해가 되기 위해서는 필요조건이 많이 필요하다. 다른 방법을 찾아보길 바람.	(서명)