제17회 임베디드SW경진대회

개발완료보고서 [지능형 휴머노이드]

0. 작성 시 주의사항

※아래의 작성 양식(제출분량, 폰트, 크기, 줄 간격 등)을 미준수 시 서류 평가의 감점요인됨

※ 제출 분량 : A4 용지 상세내용 포함 30 page 이내
※ 작성 양식 (폰트 : 맑은 고딕 / 폰트 크기 : 10pt / 자간 : 0% / 장평 : 100% / 줄 간격 : 130%)
※ 제출 포맷 : pdf

1. 팀 정보

팀명	Hands	팀장	황준영
팀원	정형원	팀원	박도현
팀원	박주영	팀원	

2. 개발완료보고서

1. 개요

1.1. 작품 개요

제안하는 작품에 대한 개요를 자세히 기술한다.

1.2. 개발 목표

개발 목표를 명확하게 제시한다.

2. 개발 환경 설명 (최대한 자세하게 기술)

2.1. Software 구성

2.2. Software 설계도(흐름도 및 클래스 다이어그램 등 (개발언어에 따라 선택))

2.3. Software 기능 (알고리즘 설명 포함)

2.4. 프로그램 사용법 (Interface)

2.5. 개발환경 (언어, Tool, 사용시스템 등)

3. 개발 프로그램 설명 (최대한 자세하게 기술)

3.1. 파일 구성

3.2. 함수별 기능

3.3. 주요 함수의 흐름도

3.4. 기술적 차별성

4. 개발 중 장애요인과 해결방안

개발 과정에서 나타났던 모든 장애 요인(Risk)들을 나열하고 이러한 장애요인들이 발생했던 경우 어떻게 해결했는지 구체적으로 제시한다.

5. 개발결과물의 차별성

개발한 결과물에 대해 타 팀과의 차별성 혹은 우수성을 서술한다.

6. 단계별 개발계획 및 실제 참여인원 및 업무 분장

참여 인원의 역할을 구체적으로 제시하고 개발 일정 계획상에서 각 참여 인력이 어떤 역할을 수행했는지, 어떤 부분을 협업하였고, 어떤 개발 절차로 개발은 진행했는지에 대해 자세히 명시한다.

1. 개요

1.1. 작품 개요

본 작품에서는 장애물 코스를 통과하기 위한 알고리즘을 구현하고 이를 휴머노이드의 임베디드 보드에 적용한다. 이를 통해 외부 서버의 도움 없이 재난 환경에서 능동적으로 파악하고 임무를 수행하는 휴머노이드의 역할을 확인한다. 본 작품에서 수행하고자 하는 내용은 다음과 같다.

1. 영상 및 센서처리 기술을 이용한 외부환경 인식
2. 장애물, 문, 밸브 등 과제를 해결할 알고리즘 고안 및 구현
3. 장애물 코스를 제작하여 테스트 진행을 통한 성능 확인

1.2. 개발 목표

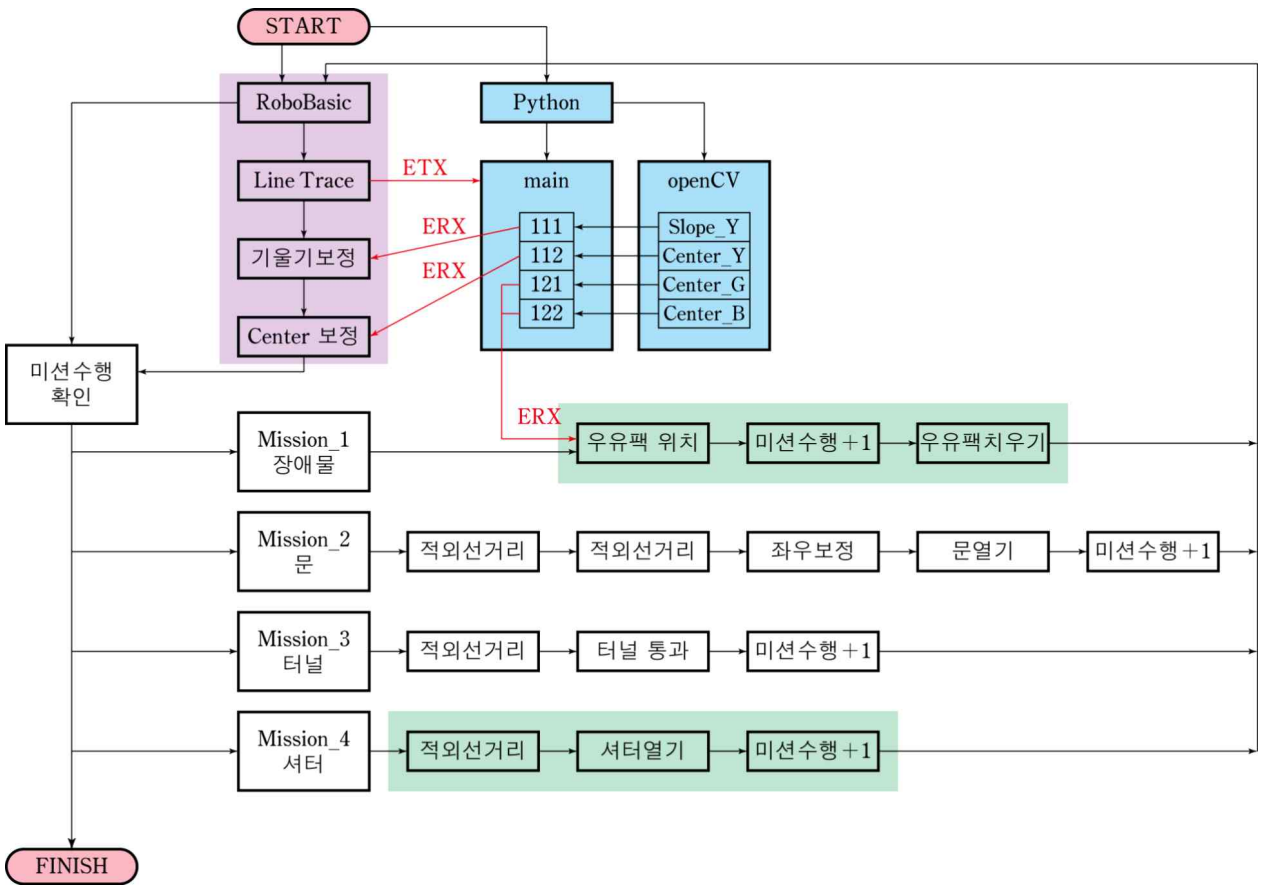
향후 재난 상황에서 실제로 활용할 수 있는 휴머노이드의 기초적인 형태를 구현해보려고 한다. 외부장치 카메라와 거리센서를 이용하여 장애물과 외부 환경을 파악하고, 이를 반영하여 요구되는 임무를 수행하는 휴머노이드를 구현하는 것이 목표이다.

2. 개발 환경 설명 (최대한 자세하게 기술)

2.1. Software 구성

소프트웨어는 크게 라즈베리파이와 로봇 제어보드 소프트웨어로 나누어져 있다. 라즈베리 파이에서는 파이썬과 openCV를 이용하여 카메라로부터 받아온 영상을 처리하고, 영상에서 주행에 필요한 데이터를 분석하여 로봇 제어보드가 이해할수 있는 형태의 데이터로 가공하여 송신한다. 로봇 제어보드는 라즈베리파이로부터 받아온 데이터를 바탕으로 현재 상황에서 로봇이 취해야할 동작을 판단하고, 로보베이직을 이용하여 각각의 모터를 제어한다.

2.2. Software 설계도(흐름도 및 클래스 다이어그램 등 (개발언어에 따라 선택))



보라색: Robobasic에서 루프 도는 부분

파란색: 파이썬

초록색: 선두로봇만 실행하는 코드

빨간색: 파이썬과 Robobasic이 통신하는 부분

2.3. Software 기능 (알고리즘 설명 포함)

Robobasic에서 필요한 자원을 라즈베리파이에서 돌고있는 Python프로그램에 요청한다. Python에서

OpenCV를 통해서 노란색, 파란색 선, 초록색 우유팩을 감지하고 선의 센터와 기울기, 우유팩의 위치 등을 반환한다. 그러면 반환받은 값을 판단하여 동작을 실행한다.

2.4. 프로그램 사용법 (Interface)

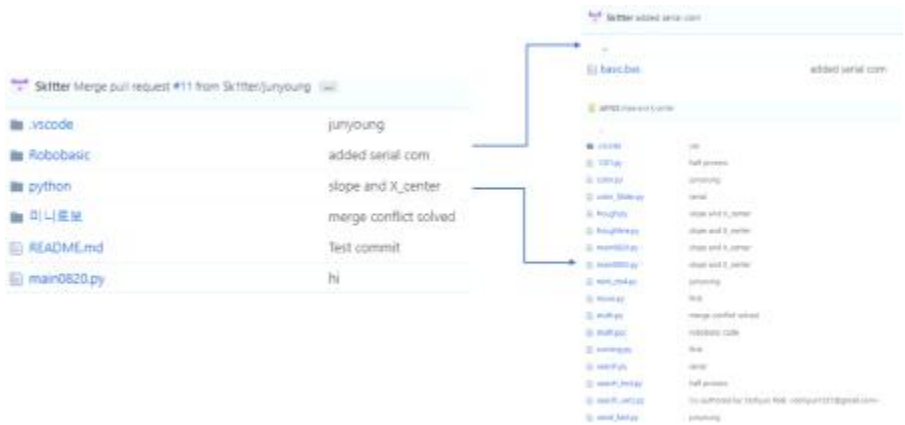
1. 미니 로봇 등 뒤 전원버튼으로 전원을 켠다
2. 라즈베리파이가 켜지면서 자동으로 파이썬이 실행된다
3. 리모콘에 1을 눌러 로보베이직 코드를 작동시킨다.

2.5. 개발환경 (언어, Tool, 사용시스템 등)

OS	WINDOWS 10, Raspbian Linux
개발환경	Pycharm, VScode, Robobasic
개발언어	Python 2.7, Robobasic
기타	OpenCV, Numpy
이슈관리	Git

3. 개발 프로그램 설명 (최대한 자세하게 기술)

3.1. 파일 구성



파일 구성 Github

3.2. 함수별 기능

3.2.1 라인 트레이싱

초록색 우유곽을 집어올릴 때 몸을 낮춰야 하는데, 다시 일어날 때 몸의 균형이 무너지는 경우가 너무 많았다. 노란 라인의 YCrCb 값을 기준으로 저장한 다음 HoughLine 알고리즘을 적용해서 직선을 검출한다. 직선이 로봇과 한가운데에서 직선으로 만나도록 로봇의 상대위치를 출력한다.

3.2.2 파란선/장애물 인식

파란선이 이루는 ㄷ자 공간을 큰 영역으로 인식하고 초록색 우유팩을 그 영역 안으로 들어서 옮기도록 한다.

3.2.3 징검다리 구간

라인 트레이싱의 오차를 최소화 하여 직진하도록 한다.

3.2.4 문 앞 장애물 인식

초록색 우유팩을 색깔로 식별한다. 확인되면 장애물 치우기 코드를 실행시킨다.

3.2.5 문 인식

적외선 센서에 무언가 인식될 경우, 첫 번째에는 문이라고 판단해서 몸을 옆으로 돌리고 게걸음으로 지나가는 코드를 실행시킨다.

3.2.6 터널 인식

적외선 센서에 무언가 인식될 경우, 두 번째에는 터널이라고 판단해서 터널을 지나가는 코드를 실행시킨다.

3.2.7 셔터 인식

초록색 우유곽을 집어올릴 때 몸을 낮춰야 하는데, 다시 일어날 때 몸의 균형이 무너지는 경우가 너무 많았다.

3.3. 주요 함수의 흐름도

3.3.1 라인 트레이싱

라인트레이싱 알고리즘은 노란색 선의 위치와 기울기를 검출하는 알고리즘 두 개로 나뉜다. 두 알고리즘 모두 노란색 라인의 YCrCb 값을 이용해 카메라 영상을 재처리하고, 이 재처리된 프레임에 HoughLine 알고리즘을 적용해 직선을 검출한다. 이다음 직선의 기울기와 가운데로부터의 offset을 계산해 로봇이 얼만큼 돌것인지, 얼만큼 옆으로 가야하는지 계산해냈다.

3.3.2 파란선/장애물 인식

파란선과 장애물 또한 라인트레이싱과 마찬가지로 YCrCb 값을 이용해 카메라 영상을 재처리 하고, HoughLine 알고리즘을 적용해 파란색 영역을 잡아냈다. 그 다음에 우유팩을 집어든 다음 파란색 영역안으로 걸어들어간 후 영역안에 들어갔다고 판단되면 우유곽을 내려놓는다. 이후 걸어왔던 만큼 다시 돌아간다.

3.3.3 징검다리 구간

징검다리 구간에는 별다른 알고리즘을 구현하지 않았다. 라인트레이싱만으로도 충분히 강력한 직진 알고리즘을 구현할 수 있었다.

3.3.4 문 앞 장애물 인식

라인트레이싱과 마찬가지로 YCrCb 값을 이용해 카메라 영상을 재처리하고, 앉아서 바닥쓸기 동작을 통해 앞의 장애물을 치운다. 만일 서서하면 무게중심이 너무 높아 넘어질 수도 있다.

3.3.5 문 인식

첫 번째로 적외선 근거리 센서에 감지되면 문이라고 가정하고, 몸을 옆으로 돌린다음에 게걸음을 해서 넘어간다. 정면으로 걸어가면 뒤로 잘 넘어간다.

3.3.6 터널 인식

두 번째로 적외선 근거리 센서에 감지되면 터널이라고 가정하고, 뭄을 뒤로 젖혀서 터널 안을 기어간다.

3.3.7 셔터 인식

세 번째로 적외선 근거리 센서에 감지되면 셔터라고 가정하고, 앉은 다음 팔을 앞으로 뻗고, 일어난다. 그 다음 만세를 해서 키보다 높은 곳 까지 셔터를 올린다. 그 후 카메라로 정면을 보고, 셔터가 감지되지 않는다면, 직진한다.

3.4. 기술적 차별성

마스터 슬레이브를 이용하여 동작을 하는 동시에 영상처리와 분석을 동시에 할 수 있고, Thread를 이용하여 동시에 두가지 이상의 색을 검출 할 수 있다. 라인트레이싱을 할 때 Bounding rectangle을 이용하는게 아닌 Line detection을 이용하여 수학적인 선을 모델링 하여 분석한다. 또 영상처리를 함에 있어 HSV나 RGB가 아닌 YCrCb를 이용하여 그림자와 관계없이 정확한 색 인식을 할 수 있게 하였다.

4. 개발 중 장애요인과 해결방안

4.1 빛 반사

YCrCb 컬러 스페이스를 이용하여 빛 반사 부분을 해결하였다. 백색광의 경우에는 Y, 즉 휘도성분이 바뀐다. 기존 HSV나 RBG 방식에서는 이러한 휘도성분을 하나의 변수로 변리하지 못해서 빛 반사에 영향을 많이 받았으나, YCrCb 방식은 이를 하나의 성분으로 분리해서 영향을 줄일 수 있었다.

4.2 동작 정확성

로봇을 공중에 똑바로 든 상태로 초기화한 다음, 아주 살짝 내려놓았을 때의 대칭성을 기준으로 영점조정을 하였다. 두 로봇에 대해 개별적으로 수행하여, 최대한 많은 코드를 재활용할 수 있도록 했다.

4.3 걷기

기존에 주어진 걷기 동작은 너무 느리고, 전진종종걸음의 경우에는 바닥 재질에 따라 죄우 한쪽으로 치우치는 경향이 있다. 따라서 공중에 뜨는 동작을 최소화 하고, 최대한 빠르게 직진할 수 있도록 했다.

4.4 장애물 치우기 동작

초록색 우유곽을 집어올릴 때 몸을 낮춰야 하는데, 다시 일어날 때 몸의 균형이 무너지는 경우가 너무 많았다.

4.5 문 동작

초록색 우유곽을 집어올릴 때 몸을 낮춰야 하는데, 다시 일어날 때 몸의 균형이 무너지는 경우가 너

무 많았다. 문의 무게 때문에 앞으로 열기 힘들었다. 따라서 게걸음을 해서 좀 더 많은 무게를 지탱하면서 앞으로 걸아나갈 수 있도록 했다.

4.6 셔터 동작

초록색 우유곽을 집어올릴 때 몸을 낮춰야 하는데, 다시 일어날 때 몸의 균형이 무너지는 경우가 너무 많았다. 이 또한 일어날 때 몸의 균형이 무너

5. 개발결과물의 차별성

휴머노이드 로봇은 사람을 닮은 형태를 하고 있다. 사람과 비슷한 형태를 지니고 있음으로서 휴머노이드는 사람이 사는 환경에서 활동하기 최적화 되어있다. 재난 상황에서 휴머노이드는 사람을 위해 설계되었지만 사람이 진입하기는 위험한 곳에서 활약할 수 있다. 이 작품에서 휴머노이드의 특성을 잘 활용하여 미래 휴머노이드 활용에 대한 가능성을 열었다고 볼 수 있다.

6. 단계별 개발계획 및 실제 참여인원 및 업무 분장

단계별 개발 진행은 다음과 같다.

구분		6		7		8		9		10		11		12	
계획 / 분석	휴머노이드 영점 조절														
	실험 환경 개발														
설계	알고리즘 설계														
	동작 설계														
개발	영상처리 알고리즘 제작														
	로봇 동작 개발														
	상황 판단 알고리즘 개발														
테스트	미션 수행 테스트														
종료	대회 진행														

참여 인원 업무 분장은 다음과 같다.

성명	담당업무
황준영	영상 처리 알고리즘
박도현	동작, 알고리즘
박주영	동작, 알고리즘
정현원	통신 알고리즘