

0. 작성 시 주의사항

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

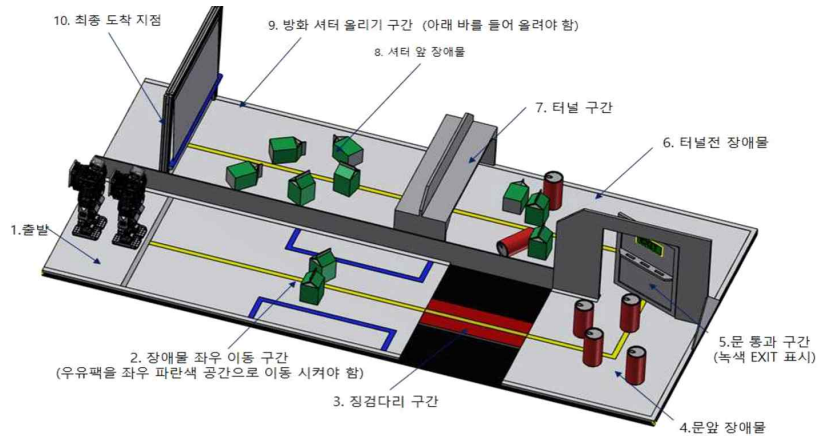
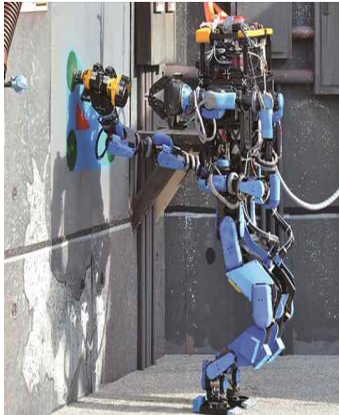
- ※ 제출 분량 : A4 용지 상세내용 포함 30 page 이내
- ※ 작성 양식 (폰트 : 맑은 고딕 / 폰트 크기 : 10pt / 자간 : 0% / 장평 : 100% / 줄 간격 : 130%)
- ※ 제출 포맷 : pdf

1. 팀 정보

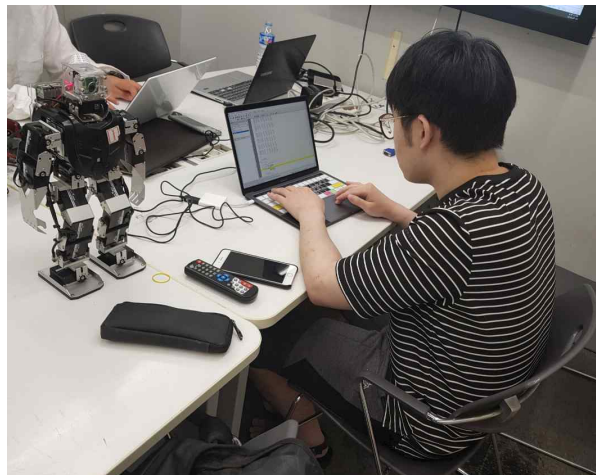
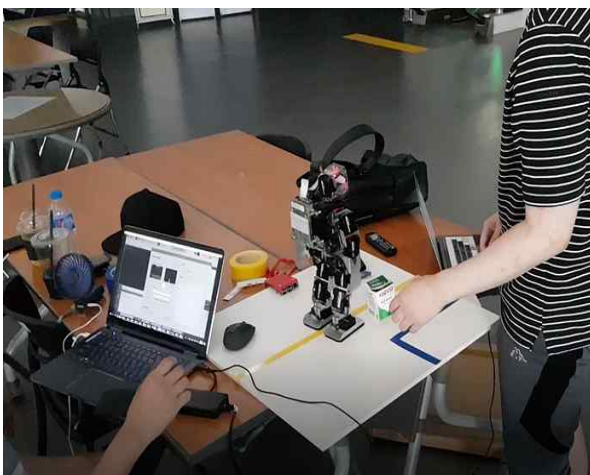
팀명	BBASAE(빠세)	팀장	박준오
팀원	신용민	팀원	오준혁
팀원	김지운	팀원	

1. 개요

1. 작품 개요 & 개발 목표



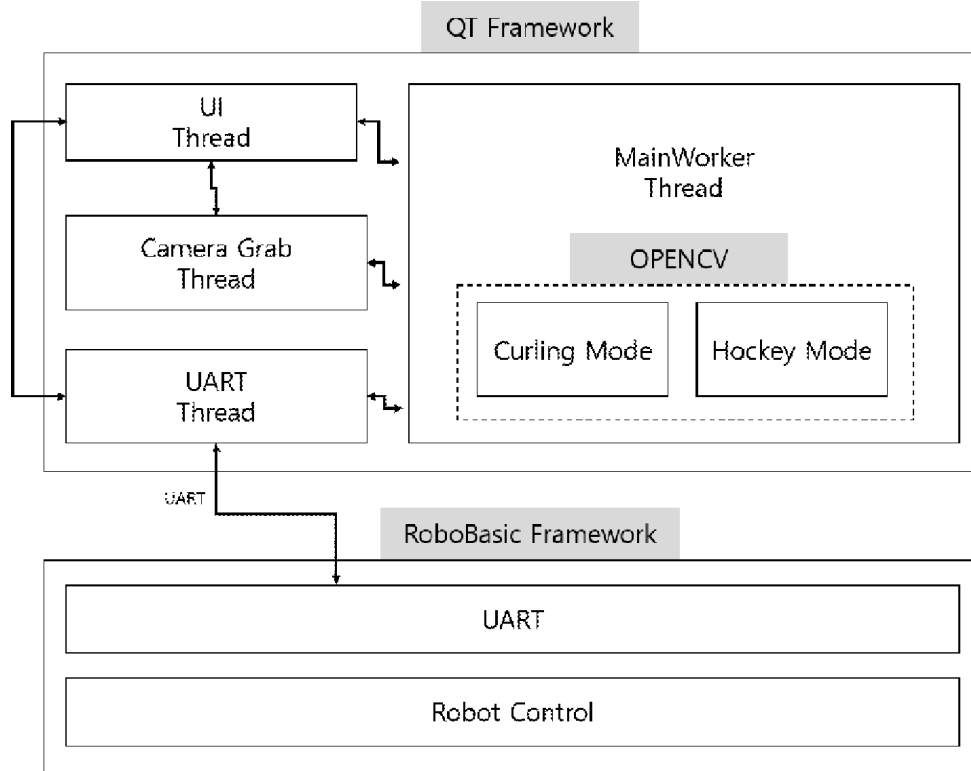
- 다르파 세계 재난로봇 경진대회(DRC)는 미국 국방성 산하 고등연구계획국(DARPA)이 주최하는, 세계 최고 성능의 재난로봇을 가리는 대회와 같이 상황이 주어진 대회로 난제를 해결해야함



- 경기를 해결하기 위해 다양한 아이디어와 기술이 활용되어 기술 개발에 긍정적인 결과를 기대

2. 개발 환경 설명

2.1. Software 구성 & Software 설계도(흐름도 및 클래스 다이어그램 등 (개발언어에 따라 선택))



- QT Framework

: 라즈베리파이는 QT Framework 기반으로 구현되어 있으며, QT Creator 툴을 이용하여 개발을 한다. QT Framework를 사용함으로써, 다양한 기능들을 쉽고 편리하게 사용할 수 있다.

- OPENCV

: OPENCV는 마이크로소프트에서 만든 영상처리 라이브러리 이다. QT에서도 OPENCV를 이용할 수 있다. 단, QT용으로 다시 빌드를 해야하기 때문에, 소스를 다운받아 QT용으로 CMAKE를 이용하여 다시 빌드하여 사용 하였다.

- 독립적 4개의 Thread

: 라즈베리파이에서는 총 쓰레드가 4개가 동작한다. 첫 번째는 UI Thread로써 UI버튼 / Radio 버튼 / 영상 출력등을 담당한다. 두 번째로 Worker Thread는 주 메인 알고리즘을 처리하는 쓰레드 이다. UI Thread와 분리함으로써 UI의 멈춤 현상을 방지해 준다. 모든 알고리즘을 UI에서 처리하게 되면 해당 알고리즘이 끝날 때 까지 UI는 멈추게 된다. 세 번째는 Camera Grab Thread이다. Camera Grab Thread에서는 초당 10장의 사진을 Grab하게 된다. 마지막으로 UART Thread는 로봇의 컨트롤을 담당하는 마이컴과 시리얼 통신하는 쓰레드 이다.

- RoboBasic

: RoboBasic은 마이컴 기반의 미니로봇 사의 로봇을 쉽게 제어하기 위한 SW Framework로써, HW지식 없이 간단한 SW 기법만을 이용하여 로봇의 움직임을 구현할 수 있다.

2.2. Software 기능

- 프로세스간 통신

Raspberry Pi <-> Micom : Serial

- 로봇과 라즈베리파이간 송,수신

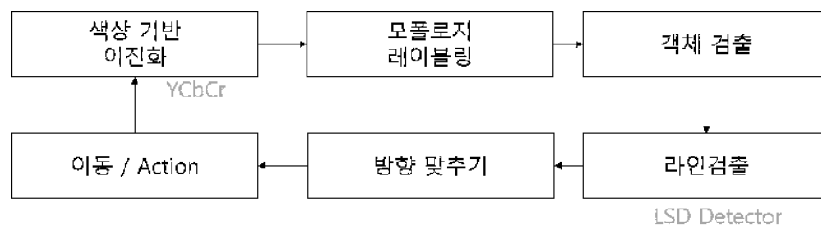
- QT 라이브러리 : QT 라이브러리를 활용하여 Uart통신 TX,RX제어

- 쓰레드간 통신

QT Framework의 SIGNAL & SLOT

- SIGNAL & SLOT

: 각각의 쓰레드 간 통신은 QT의 SIGNAL & SLOT 방식을 이용한다. 사용자 지정 SIGNAL이 발생하면, 미리 지정된 SLOT이 동작함으로써 쉽게 통신할 수 있다. 쓰레드와 쓰레드간의 통신뿐만아니라 class와 class간의 데이터 전달도 사용된다.



1). 색상 기반 이진화

: 임베디드 환경에서의 영상처리는 프로세스파워에 의해 제한적인 낮은 수준의 구현으로 동작된다. 특히, 제한된 환경과 조명에서는 색상기반 처리도 동작성이 보장되므로, 본 프로젝트에서는 색상기반의 객체 검출을 제안한다.

색상기반의 검출을 수행하기 위해서는 컬러 공간에 대한 선택이 필요하다. 일반적으로 RGB를 사용하지만 조명의 영향을 줄이기 위해 본 프로젝트에서는 YCbCr 컬러 공간을 사용하였다. Y는 밝기 정보를 Cb와 Cr은 파랑색과 빨간정도를 나타낸다. 색상 기반의 검출에서는 Cb와 Cr값을 이용하고 검정과 흰색 구분에서는 Y값을 이용하여 구분할 수 있다.

색상기반의 이진화란 일정 범위의 임계치 안에 들어오면 255(흰색)을 아니면 0으로 만든 영상을 의미한다. 예로, 파랑색을 검출하기 위해서는 Cb값이 높고 Cr값이 낮기 때문에 해당 범위에 들어오는 픽셀영역은 255를 아니면 0을 부여하여 영상을 생성한다.



<원본 영상>

<이진화 결과>

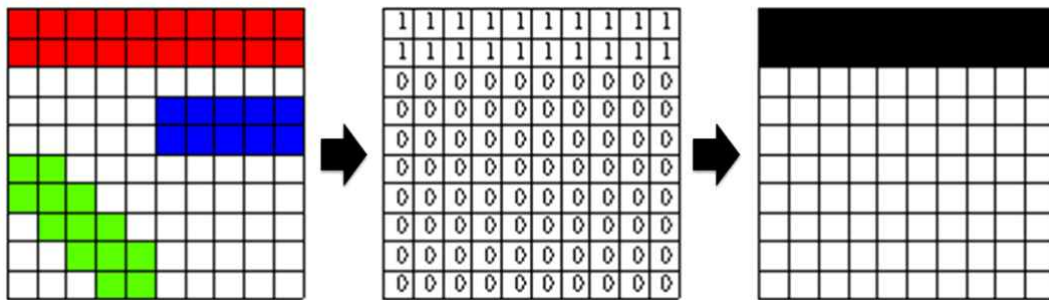
이진화된 영상은 찾고자 하는 객체의 후보영역으로 선정이되며, 찾고자 하는 객체는 영상에서 일정크기 이상으로 제일 크게 존재한다는 가정으로 구현 하였다.

2. 모폴로지(잡음제거) & 레이블링

: 모폴로지 기법은 이진화된 영상에서 가장 많이 쓰이는 잡음 제거 기법이다. 침식, 팽창, 열림, 닫힘 연산 총 4개가 있으며, 일반적으로 팽창 후 침식을 하면 구멍이 뚫린 듯한 잡음을 제거 할 수 있다. 가장 큰 영역이 찾고자 하는 영역이라는 가정이 되기 위해서는 구멍뚫린 잡음현상을 최소화 해 줘야 한다.

침식	영역에서 가장 작은 값을 픽셀 중심부의 값으로 바꾸는 최소값 필터 역할
팽창	영역을 넓혀가는 연산이며 최대값 필터의 역할
열림(opening)연산	침식 연산 -> 팽창연산을 수행하여 잡음들을 제거할 수 있음, 본 영상 처리는 닫기 연산을 적용
닫힘(closing)	연산 : 팽창 연산-> 침식연산을 수행하여 구멍을 제거할 수 있음

본 프로젝트에서는 팽창 후 침식 연산 즉, 열림 연산을 2회 씩 실시하여 잡음을 제거해 주었다. 레이블링이란, 이진영상에서 뭉쳐있는 영역끼리 Segmentation 하기 위한 기법으로 뭉쳐있는 같은영역을 같은 번호로 부여하는 연산을 의미한다.

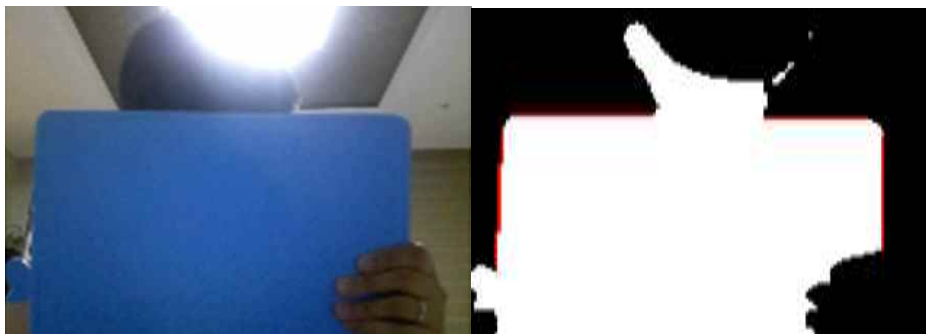


앞선 계속된 가정에서와 같이, 찾고자 하는 물체는 특정 색상을 가지고 있고, 특정색상중에 일정크기 이상 가장 큰영역이다라는 가정을 성립하기 위해 레이블링 연산은 효과적이다. 이진영상 전역을 레이블링을 한 뒤, 가장 많은 수를 가지고 있는 영역을 찾아내면 찾고자 하는 관심영역일 확률이 높다.

3. 라인 검출(각도 맞추기)

: 본 미션에서는 바닥에 색상기반의 라인이 그려져 있다. 이를 이용하여 로봇의 틀어진 각도를 보정할 수 있다. 라인 검출 방법에는 고전적으로 Hough 변환이 있으나, 실제 라인이 잘 찾아지지 않는 결과를 볼 수 있다.

이를 해결하기 위해 LSD를 이용 하였다. LSD는 Line Segment Detector의 약자로 최근 라인 검출기로 많이 사용되고 있다.

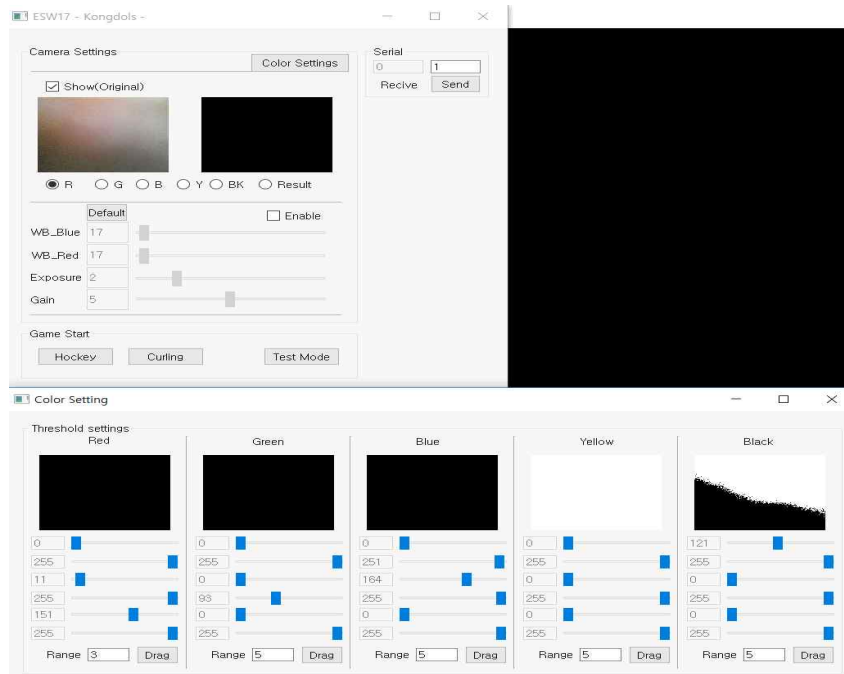


<원본 영상>

<라인 검출 결과>

틀어진 각도는 라인검출의 양끝 포인트의 기울기로 구 할 수 있다. 이론적으로는 카메라 캘리브레이션을 통해 정확히 계산해야 하지만, 실험에 의해 정답과 차이가 크지 않아 양 끝점의 기울기를 틀어진 각도로 사용하였다.

2.3. 프로그램 사용법 (Interface)



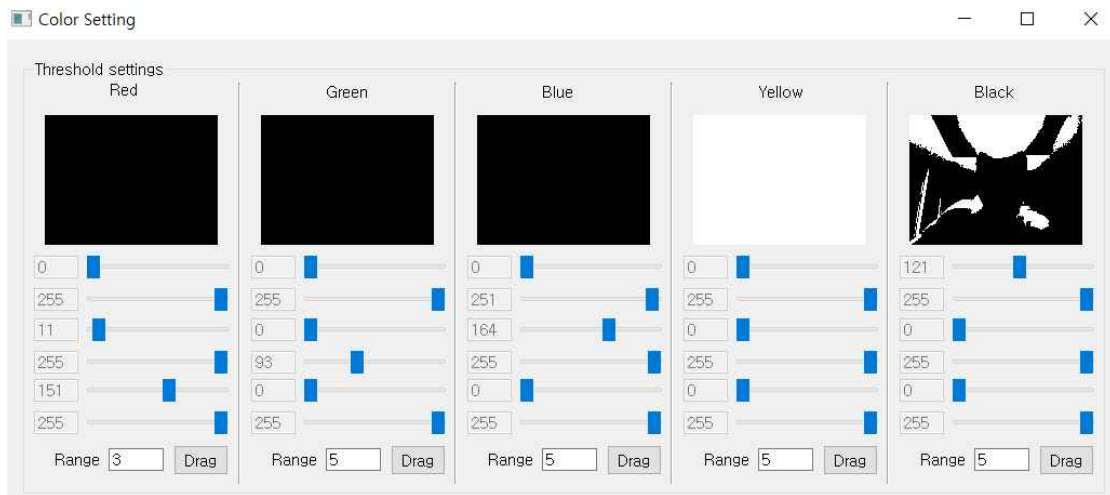
<UI>

- MainWindow

: 시리얼 명령어 송수신, 실시간 영상 출력, 결과 화면 출력, 동작 모드 설정, 카메라 파라미터 설정등을 한다.

- Color Setting

: 관심 색상에 대한 임계치 설정을 한다.



- 모드 설정

: 테스트 모드, 각 미션에 따른 모드를 설정한다

- 디버그 프린트창

: UI에서 테스트 불가능한 부분을 디버그 메시지로 출력한다.

선택 C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe



대회 당일에는 GUI 사용을 하지 않으며, 순수 리모컨 컨트롤로만 동작 가능하도록 구현 함

2.4. 개발환경 (언어, Tool, 사용시스템 등)

언어	프레임워크	OS
C++	QT	라즈비안

3. 개발 프로그램 설명

3.1. 파일 구성

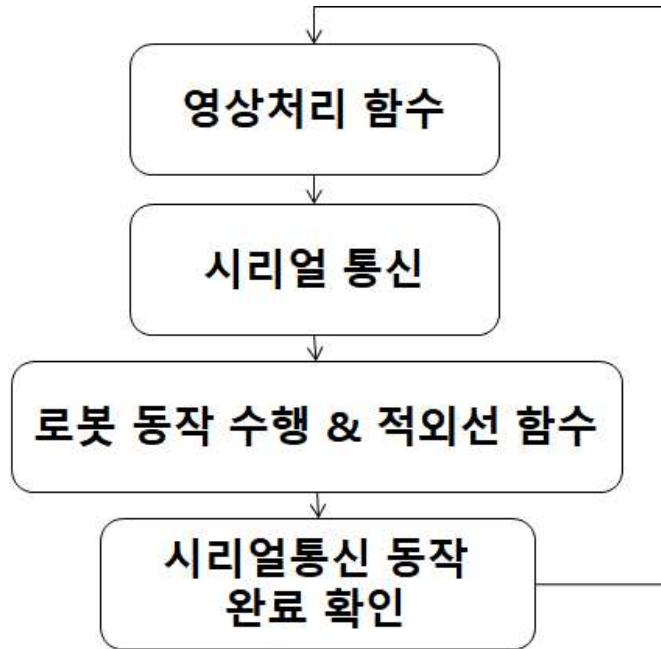


- UI관련 처리, 이미지 프로세스 파일, 시리얼 통신 등의 처리는 Thread 파일로 구성되어 있음,

3.2. 함수별 기능

영상처리 함수	라인 각도 판별 함수	라인 기울기 측정
	색상 인식 함수	색상에 따른 물체(Object) 판별 ex) 우유팩, 문
시리얼 함수	로봇 간 Uart통신	
적외선 함수	로보베이직에서 특정 적외선 거리에 따른 이벤트	

3.3. 주요 함수의 흐름도



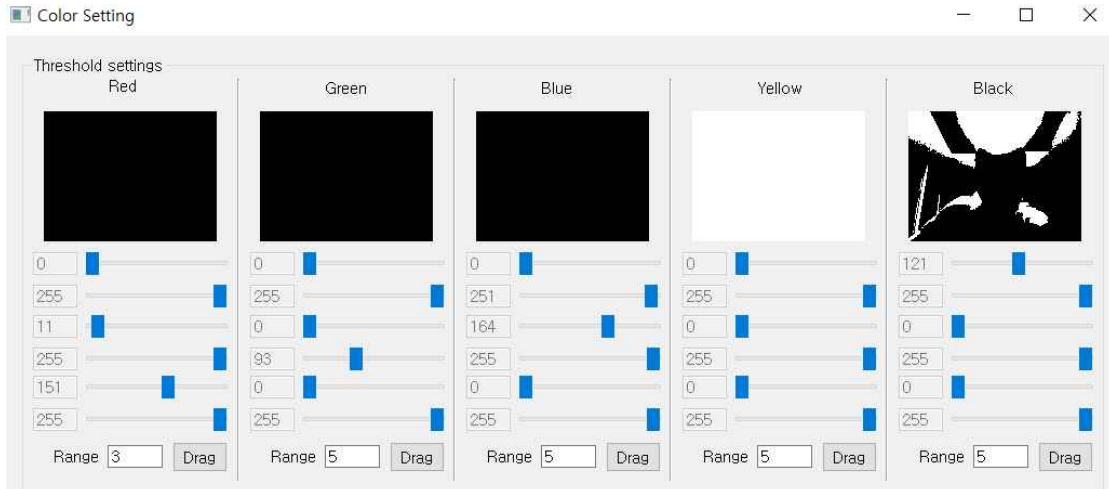
3.4. 기술적 차별성

- UI와 VNC 원격 기반 온라인 개발 환경 구축
- 멀티 쓰레드 구현
 - : 독립적 4개의 쓰레드를 통해 처리속도 향상
 - : 안정성 향상
- 빠른 처리 속도(5 FPS)
 - : 모든 영상처리시에도 5 FPS이상의 속도 측정 됨

4. 개발 중 장애요인과 해결방안

- 로봇 OS 부팅 상태 확인 및 로봇 동작 준비상태 확인 : 로봇상태를 디버그용 사운드모듈을 활용하여 확인
- 로봇 영점설정 : 일관성있는 영점 설정하기 위해서 담당자가 DB화함으로써 영점 변수 최소화
- 미션 수행중 넘어지는 이벤트 발생 : 2축 기울기 센서를 활용하여 이벤트를 확인하여 알고리즘을 다시 시작하는 알고리즘 구현
- 카메라 화이트 밸런스로 인한 카메라 색상 변하는 문제: 화이트 밸런스를 제어할 수 있는 알고리즘을 적용하여 색상기반에 영상처리 문제 해결

5. 개발결과물의 차별성



- 개발된 결과물로 주어진 시간내에 빠르게, 영상처리 개발환경을 설정하여 미션을 준비할 수 있음
-

6. 단계별 개발계획 및 실제 참여인원 및 업무 분장

구분 \ 역할	상반기	하반기
영상처리 (신용민)	- OPENCV라이브러리를 활용 - 영상처리 환경 구축 및 알고리즘 개발	- 영상처리 알고리즘 적용하여 피드백 - 추가 알고리즘 개발
로봇제어 (오준혁, 김지운)	- 로봇 제어 프로그래밍 연구 - 로봇과 라즈베리파이간 통신 구축	- 미션 알고리즘 개발 - 미션 시간을 단축위한 동작 구현
라즈베리파이 (박준오)	- 로봇과 라즈베리파이간 인프라 구축 : 유,무선 통신설정, 영상처리 설정, 소스 틀 - 라즈베리파이 영상처리 적용	- 라즈베리파이 처리 성능 향상

구분	단계별 목표	6월	7월	8월	9월	10월	11월	12월
사전 자료 조사	임베디드 시스템 이해	○ ○						
	플랫폼에 대한 센서 이해	○ ○						
	임베디드(라즈베리파이) 개발 환경 구축	○ ○						
	로보 베이직 개발 환경 학습		○					
H W	임베디드 보드 분석		○ ○					
	모터 동작에 대한 분석			○ ○				
	경기장 구축				○ ○			
S W	라즈베리파이와 로봇간의 시스템 구축		○ ○ ○					
	OPENCV 스터디	○ ○ ○						
	영상처리 구현		○ ○ ○ ○		○ ○ ○ ○			
	로봇 스터디	○ ○ ○ ○						
	로봇 동작 구현					○ ○ ○ ○		
	통신 SW구축		○ ○					
	통합과정에서 문제점 해결						○ ○	
미 션	미션 개발					○ ○ ○ ○ ○		
						○ ○ ○ ○ ○		