

# 2020년 스마트해상물류×ICT멘토링 프로젝트 결과보고서

프로젝트명	스마트해상물류. 5G와 수상드론을 통한 울산항 안전운항 관리 솔루션 및 시뮬레이션)
-------	--

# 요 약 본

프로젝트 정보	
주제영역	<input checked="" type="checkbox"/> 지능화 <input type="checkbox"/> 자동화
기술분야	<input type="checkbox"/> IoT <input type="checkbox"/> 모바일 <input checked="" type="checkbox"/> 데스크톱 SW <input type="checkbox"/> 인공지능 <input type="checkbox"/> 보안 <input checked="" type="checkbox"/> 가상현실 <input type="checkbox"/> 빅데이터 <input type="checkbox"/> 자동제어기술 <input type="checkbox"/> 블록체인 <input type="checkbox"/> 영상처리
달성성과	<input checked="" type="checkbox"/> 특허출원 <input checked="" type="checkbox"/> 논문발표 <input type="checkbox"/> 앱등록 <input type="checkbox"/> 프로그램등록 <input type="checkbox"/> 기술이전 <input type="checkbox"/> 실용화 <input checked="" type="checkbox"/> 공모전(공모전명 2020 스마트 해상물류 공모전 ) <input type="checkbox"/> 기타( )
프로젝트명	스마트해상물류, 5G와 수상드론을 통한 울산항 안전운항 관리 솔루션 및 시뮬레이션)
프로젝트 소개	<p>해양사고에 소형 선박의 운항자들의 운항 실력 및 안전 의식이 많은 영향을 미치고 있다. 이로 인한 선박 사고를 방지하기 위해 도선사를 위한 안전교육 및 운항교육을 실시하여 항만관계자 및 도선사들의 사고로 인한 피해를 줄인다.</p>
개발배경 및 필요성	<p>우리나라의 울산항은 항 내수 면적이 1,100만 제곱미터에 달하는, 부산항 다음으로 거대한 국내 최대급의 무역항구이다. 울산항은 액체 중점 항만으로서 원유, 화학공업생산물, 석유정제품 등의 액체화물이 전체 물동량의 80%를 차지하는데, 해당 화물들의 위험성에 비해서 항구의 사고건수는 해가 지나갈수록 증가하는 추세에 있다. 이는 좁은 항로, 복잡해지는 항만환경 등의 문제에 더해 운항자의 운항부주의가 합해진 것이 주요 원인으로 꼽히고 있다. 특히나 해양사고의 원인 중 운항부주의가 37%, 사고 선박 중 500톤 미만의 선박이 64%에 달하는 것을 보아 소형 선박 운항자들의 운항 실력 및 안전의식이 적지 않은 영향을 끼치고 있음을 알 수 있다. 이에 따라 소형선박 운항자를 대상으로 안전교육 및 운항교육을 목적으로 선박 안전 운항 솔루션을 개발하는 것, 그리고 운항시의 충돌사고를 방지하기 위한 관제 솔루션을 개발하는 것이 본 프로젝트의 배경 및 목적이라 할 수 있다.</p>
프로젝트 주요기능	<p>- 단계적인 안전 교육 및 사고 방지 솔루션 제공</p> <p>단계 1) 조타기, 기어를 사용하는 가상현실 시뮬레이션</p> <p>단계 2) 조타기, 기어로 수상드론을 통해 운항 교육</p> <p>단계 3) 실제 운항 시 gps값을 이용, 선박 충돌을 감지하는 관제 시스템의 제공</p>
작품의 기대효과 및 활용분야	<p>- 실감나는 시뮬레이션을 통한 교육과 충돌방지시스템 구축을 통해 선박운항사고 발생 확률을 줄인다.</p> <p>- 울산 항만공사는 사고로 인한 선박, 물류, 인명 등의 피해 및 손해를 방지할 수 있다.</p> <p>- 울산항만을 기반으로 한 시뮬레이션 코스를 통해 도선사들의 운항 숙련도를 높인다.</p> <p>- 선선박사고로 인한 선주의 물질적인 손해를 줄이고 선원들의 인명피해 발생 확률을 줄인다.</p> <p>시뮬레이션과 충돌감지시스템이 핵심 부품을 공유해 비용을 최소화한다.</p>

# (본문) 프로젝트 결과보고서

## I. 프로젝트 개요

### 가. 프로젝트 소개

○ 해양 사고의 원인 중 다수를 차지하는 소형 선박의 운항 부주의 문제 해결을 위해 울산항만공사에 운항 교육 및 사고 방지에 대한 전체적인 솔루션을 제공한다. 높은 완성도의 솔루션을 제공하기 위하여 작년 동일 분야의 프로젝트를 실시한 팀과 미팅을 실시하고, 작년의 운항교육 솔루션에서 미흡했던 실물조종 부분의 개선, 사고 방지 관제 시스템 추가 및 운항 교육 시스템과의 연계를 완성한다.

솔루션은 운항 교육, 충돌 감지 시스템의 크게 2가지 분류로 나누어진다.

#### 1) 운항 교육

- 운항 교육은 2가지 단계로 이루어진다.

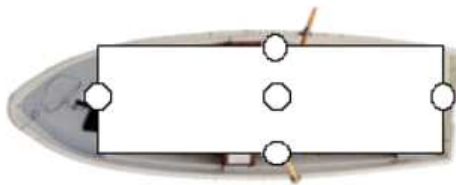
1 단계 . 울산항을 반영한 가상현실 시뮬레이션을 통한 운항 교육

2 단계 . 수상 드론을 통한 운항 교육

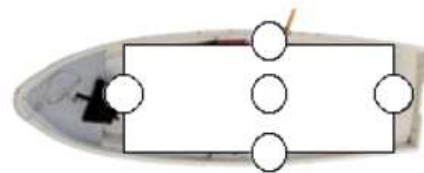
- 운항 교육을 모두 통과한 사람은 교육 자격증을 발급받고, 실제 배를 운항할 수 있는 자격을 얻는다.

#### 2) 충돌 감지 관제 시스템

- 배를 운항할 때, 배의 gps 값을 이용해 다른 배와 충돌하는지의 여부를 계산한다.



선박 A



선박 B

이해를 돕기 위한 참고 이미지입니다.

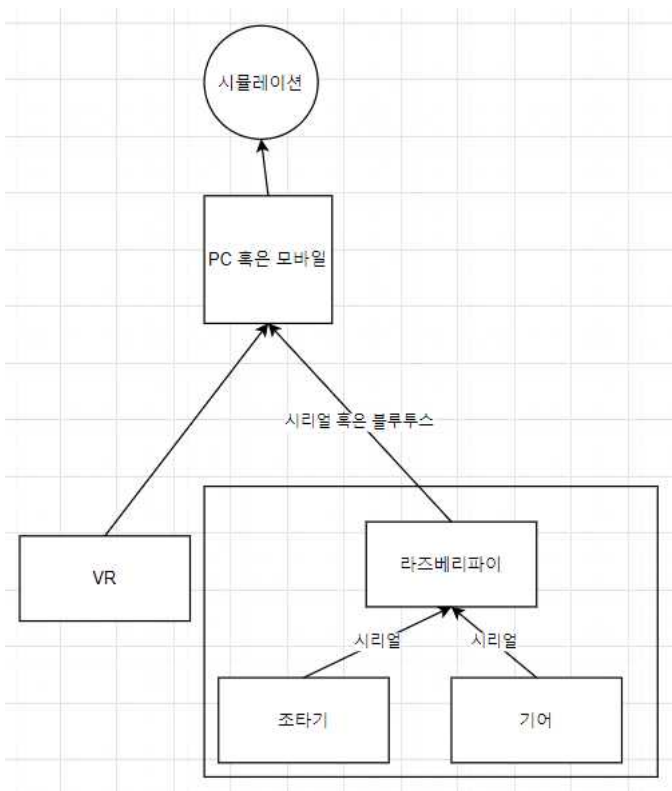
위의 그림에서 볼 때, ○에 해당하는 위치에 라즈베리파이를 이용해 gps 값을 수신 받고, 이 값을 이용해 선박 하나의 위치를 다음과 같이 사각형의 모양으로 만들어낼 수 있다. 이 때, 선박 A 와 B 의 gps 영역이 겹치지 않도록 관제 시스템을 통해 충돌을 감시하고 방지한다. 또한 운항교육에서 도선사 각자가 사용한 수상드론을 gps모듈로 사용하기 때문에 추가적인 비용 없이 시스템을 구축 가능하다.

## 나. 개발배경 및 필요성

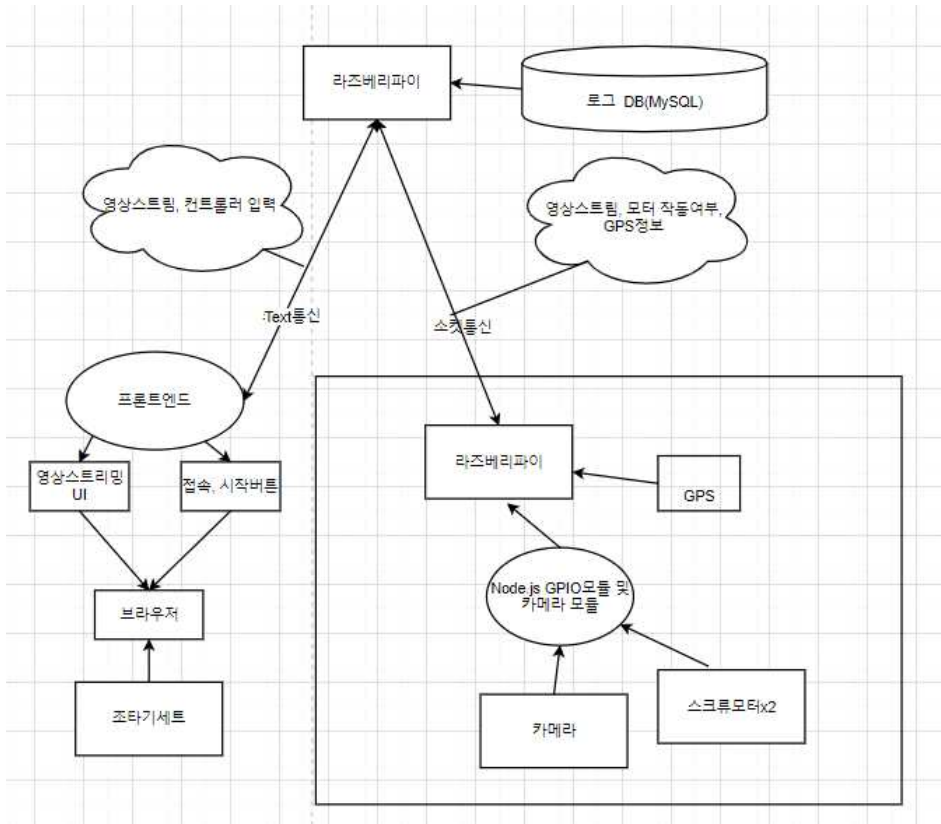
- 울산항은 액체 중점 항만으로서 액체화물이 전체 물동량의 80%를 차지하는데, 해당 화물들의 위험성에 비해서 항구의 사고건수는 해가 지나갈수록 증가하는 추세에 있다. 이는 좁은 항로, 복잡해지는 항만환경 등의 문제에 더해 운항자의 운항부주의가 합해진 것이 주요 원인으로 꼽히고 있다. 이에 따라 사고의 주요 원인이 되는 소형선박 운항자를 대상으로 안전교육 및 운항교육을 목적으로 선박 안전운항 솔루션을 개발하는 것, 그리고 운항시의 충돌사고를 방지하기 위한 관제 솔루션을 개발하는 것이 본 프로젝트의 배경 및 목적이라 할 수 있다.

## 다. 작품 구성도

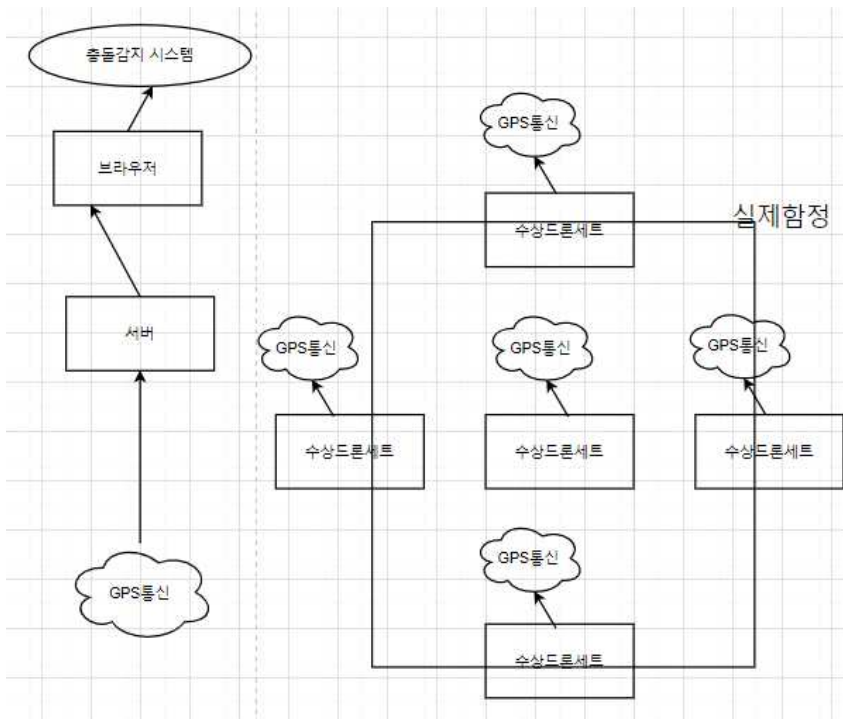
### 1) 가상 현실 시뮬레이션



### 2) 수상드론



### 3) 수상드론을 이용한 관제 시스템



#### 라. 작품의 특징 및 장점

- 기존 VTS 에서는 레이더 시스템을 사용하는데, 해당 시스템은 이중화가

되어있지 않아 정전이나 시스템 장애가 발생하면 선박 관제가 불가능하다. 이 VTS 시스템 자체를 이중화하기에는 VTS 1곳당 20억원의 예산이 필요하다. 또한, VTS 가 100% 해외의 장비들로 구축되기 때문에 시스템 개선 및 유지보수에도 아주 많은 비용이 소모된다.

- 50톤 이상 선박들은 VTS 시스템 장애 발생시 AIS ( 선박 자동 식별 장치 ) 와 유지 보수 인력으로 문제를 해결할 수 있지만, 중소형 선박은 레이더 고장 시 충돌을 방지할 수 있는 다른 대안이 없다. 하지만 해당 프로젝트에서 제작하고 있는 휴대폰의 gps값을 사용하는 관제 시스템을 사용한다면 VTS 시스템에 비해 적은 비용으로 효과적인 시스템 이중화, 보수, 대체 등이 가능하다.
- 가상현실 시뮬레이션을 제작하는데, 소프트웨어 하나만으로 운항 연습을 할 수 있으므로 기존의 시뮬레이터 시스템에 비해 접근성이 뛰어나다. 또한, 리플레이 기능을 통해 자신이 운항 연습을 했던 모습을 다시 볼 수 있어 운항 시뮬레이션 교육에 효과적이다.
- 해당 프로젝트의 수상드론은 서버를 통해 원격 조종 및 실시간 스트리밍이 가능하다. 또한, 안전 운항 교육의 목적으로도 쓰일 수 있어 범용성이 있으며, 유지 보수를 원한다면 기존 시장에 있는 드론처럼 추가적으로 센서를 달아 사용할 수 있으므로 더 적은 비용으로 큰 효과를 볼 수 있다.

## **II. 프로젝트 수행결과**

### **가. 업무분장**

번호	성명	역할	담당업무
1	김정민	멘 토	프로젝트 기획, 설계, 진행 멘토링
2	-	지도교수	-
3	김연진	팀 장	프론트엔드, 백엔드
4	김정수	팀 원2	하드웨어, 시뮬레이션
5	황준호	팀 원3	시뮬레이션, 프론트엔드
6	김성연	팀 원4	백엔드, 시뮬레이션
7	-	팀 원5	-

#### 나. 주요기능

구분	기능	설명	현재진척도(%)
S/W	가상현실 시뮬레이션	- VR glass를 이용해 가상현실로 안전한 항 연습이 가능하다.	100%
	라즈베리파이 서버	- 안드로이드, 웹, 수상드론간의 통신을 5G네트워크로 연결해주는 서버 역할	100%
	실시간 관제 시스템	- 수상드론 스트리밍 화면 및 각종 정보 를 확인하고, 현재 실제 선박 위치 및 정보를 실시간으로 전달 - 디버그 및 버그 수정 중	100%
	안드로이드	- 수상드론을 조종하고, 서버에 위치 및 영상 정보 전송	100%

구분	기능	설명	현재진척도(%)
H/W	수상드론	- 시뮬레이션 교육을 마친 후 2차 운항 교육에 사용 - 실시간 관제 시스템에서 선박의 위,경 도 정보를 전송	100%
	조타기	- 가상현실 시뮬레이션과 수상드론의 조 종에 쓰이는 컨트롤러 - 외관 조립이 진행 중	95%

#### 다. 프로젝트 개발환경

구분		상세내용
S/W 개발환경	OS	Raspberry Pi OS, Windows10
	개발환경(IDE)	Unity3D(2019.3.0f6), VisualStudioCode, Android Studio
	개발도구	Node.js(12.18.0), MySQL(8.0.20), SteamVR
	개발언어	자바스크립트, C#
	원격개발	Putty, Code-Server, WOL, RDP
H/W 구성장비	디바이스	삼성 오딧세이MR, 라즈베리파이3B+ 및 Rasbian, IPTIME공유기
	드론	라즈베리파이, 안드로이드 스마트폰
	조타기	아두이노
프로젝트 관리환경	형상관리	GitLab, Git LFS, GitKraken
	의사소통관리	GitLab issue, Slack, 카카오톡
	자료관리	구글드라이브

#### 라. 장비(기자재/재료) 활용

번호	품명	작품에서의 주요기능
1	삼성 오딧세이MR	- 가상현실 시뮬레이션 조작
2	라즈베리파이3B+	- 자체 서버 및 수상드론 메인보드
3	아두이노	- 컨트롤러 메인보드
4	갤럭시 S8	- 드론 제어 정보 전달 및 드론 관련 정보 송수신

#### 마. 프로그램 작동 동영상

○ <https://youtu.be/T6bku1yp5mo>

#### 바. 결과물 상세 이미지





## 사. 달성성과

■ 논문게재 및 포스터발표	게재(발표)자명	논문(포스터)명	게재(발표)처	게재(발표)일자
	김연진	5G 및 수상드론을 통한 울산항 안전운항 관리 솔루션 및 가상현실 시뮬레이션	정보처리학회	20.10.5
□ 앱(APP) 등록	등록자명	앱(APP)명	등록처	등록일자
	-	-	-	-
□ 프로그램 등록	등록자명	프로그램명	등록처	등록일자
	-	-	-	-
■ 특허/실용신안 출원	출원자명	특허/실용신안명	출원번호	출원일자
	팀원 명	스마트해상물류, 5G와 수상드론을 통한 울산항 안전운항 관리 솔루션 및 시뮬레이션	출원 중	출원 중
□ 기술이전	기술이전기업명	기술명	금액	이전일자
	-	-	-	-
■ 공모전	구분(교내/대외)	공모전명	수상여부(출품/수상)	상격
	대외	2020 스마트 해상물류 공모전	-	
□ 실용화	#실용화한 내용에 대한 구체적 작품설명			
□ 기타				

## Ⅲ. 프로젝트 수행방법

### 가. 프로젝트 수행일정

프로젝트 기간 (한이음 사이트 기준)		2020.00.00. ~ 2020.00.00.											
구분	추진내용	프로젝트 기간											
		1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월
계획	드론 제작, 서버 구축에 필요한 사전 조사												
분석	소프트웨어 설계, 라즈베리파이 학습												
설계	소프트웨어 설계												
	하드웨어 설계												
개발	수상드론 제작												
	서버 구축, 웹 제작												
	수상드론 제어, 서버 구축 및 연동												
테스트	실제 테스트 및 오류 수정												
종료	프로젝트 최종 마무리 및 종료												

#### 나. 문제점 및 해결방안

##### ○ 프로젝트 관리 측면

- 코로나19로 인해 프로젝트를 모여서 진행할 수 있는 개발 장소를 구할 수 없었다. 그래서 학교 동아리 방을 이용하기도 하고, 서울 창업허브에 모여 개발을 진행하기도 했다. 코로나19가 심해지면서 더 이상 모일 장소가 아무 곳도 없게 되었을 때는 팀원들끼리 줌과 디스코드를 활용해 온라인으로 현황을 공유하며 개발을 진행하였다.
- 시뮬레이션의 경우 3D 가상현실 프로그램이기 때문에 개발을 위해서는 고사양의 컴퓨터가 필요했는데, 현재 거주지의 컴퓨터로는 사양이 부족하여 개발하는데 어려움이 있었고, 한 달마다 가는 본가의 컴퓨터로만 개발을 할 수 있었다. 지속적으로 개발을 하기 위해 본가의 컴퓨터에 WOL, DDNS, RDP등의 설정을 하여 외부망에서 언제든지 접근하여 원격으로 개발을 할 수 있도록 하였다.
- 시뮬레이션은 협업을 위해 20GB가 넘어가는 고용량의 데이터를 버전관리, 공유할 필요가 있었는데, 파일의 용량이 너무 커서 기본적인 Git으로는 이를 Remote에 올릴 수가 없었고, 이를 관리하기 위해 Git LFS를 적용하였다.
- 라즈베리파이의 본 서버가 제대로 작동하지 않을 경우를 대비하여 백업용

라즈베리파이 서버를 연결해야 했는데 해당 기기를 연결해야하는 공유기는 iptime이 아닌 다른 모델이어서 ddns 설정을 하는 데 어려움이 있었다. 다른 모델의 공유기로 교체해 해당 문제를 해결하였다.

#### ○ 작품 개발 측면

- 시뮬레이션의 테스트 플레이 시, 매우 낮은 프레임과 낮은 반응속도를 보여줬다. 실 사용을 하기엔 부적합하였기에 Object Pooling, Forward 렌더링 등의 최적화 기술을 적용하여 문제없는 성능을 보여주게 되었다.
- 서버의 기능 중 하나인 실시간 드론영상 전송을 구현하기 위하여 RTMP 프로토콜을 사용할 예정이었지만, 실제 테스트 시 3초 이상의 긴 지연시간으로 인해 실시간 조종에 적용하기 힘들었다. 이를 해결하기 위해 오픈소스 JSMPEG 영상전송 라이브러리를 이용해 드론에서의 영상을 적절한 길이의 작은 영상단위로 쪼갬 뒤 인코딩하여 1초정도의 지연시간으로 실시간 전송이 가능한 코드를 만들었고, 이를 적용하고 있다.
- Google Maps API를 이용해서 관제시스템의 현재 선박의 위치를 표시해주거나 관제시스템의 애니메이션 상황을 적용해 주는데, 본래 Google Map API가 그런 용도로 사용되지 않아 표현하기에 어려움이 있었다. 따라서 해당 API에서 제공해주는 함수들을 최대한으로 활용해 위/경도 위치 및 선박 사이 거리 등을 고려한 수학적 계산을 통해 일일이 알고리즘을 제작했다.
- AIS 데이터는 민간의 수준에서 모든 배의 정보를 실시간으로 받아오는 데에 한계가 있었기 때문에 비실시간의 데이터를 무료로 제공하는 웹의 API를 적용했다.
- 본래 선박 충돌 시에 연락 체계를 웹을 통해 제공해줄 생각이었으나 선박들끼리의 무선 통신을 이용한 체계가 이미 존재해 충돌 정보를 모아 파일로 제공하는 방식으로 변경하게 되었다.
- 배를 설계해 본 경험이 없어 실제 사용하는 배 형태의 드론을 만들어도 사용이 불가능했다. 무게중심을 맞추는 수가 없어서 좌우로 기울어지기도 하고, 앞뒤도 한쪽으로 기울어지기도 했다. 그래서 하단이 평평한 구조로 변경해서 운행에 문제가 없게 제작했다. 또한 폼보드로 외형과 내부를 구성하려

고 했는데, 원하는 형태로 제작하는 데 어려움이 있어 외부와 내부를 3d프린터로 뽑고, 결합하는 방식으로 진행 중에 있다.

- 모터와 배터리에 대한 이해도가 낮아 처음 산 모터의 필요전력이 너무 높아 구동이 불가능해 좀 더 낮은 출력이 낮지만 운행이 가능한 수준의 모터를 새로 구매했다.

#### IV. 기대효과 및 활용분야

- 울산 항만공사에 사고 방지를 위한 실시간 충돌확인 관제 솔루션을 제공하여 도선사의 운항부주의 혹은 관제 실수를 미연에 방지하도록 한다. 체계적인 통합 솔루션을 제공해 선박 사고 발생률을 낮추고, 사고로 인한 선박, 물류, 인명 등의 피해 및 손해를 방지한다.
- 실제 울산항만을 반영한 맵에서 시뮬레이션, 수상드론 운행 연습을 진행하므로 도선사는 울산항만의 전체적인 지리에 대해 익숙해 질 수 있는 기회를 가질 수 있으며 배 운항의 숙련도를 높일 수 있다. 이를 통해 항만 환경에 대한 이해도 부족 및 운항실력 부족으로 인한 사고를 줄일 수 있다.
- 운항교육 및 사고방지 솔루션의 단계적이고 유기적인 연계를 통해 두 가지 솔루션을 각각 구축한 경우에 비해 비용과 시간을 줄일 수 있다.
- 기존 VTS 레이더 시스템의 정전 및 시스템 장애 시의 문제를 해당 프로젝트의 시스템을 통해 적은 비용으로 이중화하여 문제를 예방할 수 있다. - AIS 탑재가 의무가 아닌 소형 선박들은 드론을 구비하거나 휴대폰의 어플을 사용하는 것만으로도 해당 관제 시스템 내에서 위치 확인이 가능하기 때문에 소형선박으로 인한 운항 충돌 사고를 저비용으로 효과적으로 방지할 수 있다.
- 소프트웨어 하나만으로 가상현실 시뮬레이션을 사용 가능해 접근성이 뛰어나고 리플레이 기능을 활용해 자신의 운항 연습 모습을 다시 재생 해 볼 수 있어 교육적인 목적에 아주 잘 부합한다.

- 서버를 통해 원격 조종 및 실시간 스트리밍이 가능한 수상드론을 이용해 안전 운항 교육의 목적으로도 사용하고, 추가적으로 센서를 부착한다면 다른 용도로도 충분히 사용할 수 있기 때문에 범용성이 있으며 비용이 타 드론에 비해 적게 든다.
- 복잡한 항만환경을 가지고 많은 위험화물이 오가며 사고 위험이 높아 보다 정확하고 간편한 관제 시스템을 필요로 하는 항만공사들
- 상기한 여러 항만들에서 운항을 하거나 체계적인 운항교육의 필요성을 느끼는 도선사들
- 복잡한 항만의 관제 시스템을 담당하여 선박들의 충돌사고를 미연에 방지해야하는 해상관제사 및 항만을 운영하는 직원들
- 선박 운항자들을 대상으로 가상현실 시뮬레이션을 이용해 울산항만 지역에 대한 운항 숙련도를 높인다. 자신의 운항이 미숙하다고 생각하는 경우 리플레이 기능을 이용해 잘못된 부분을 숙지할 수 있도록 한다. 해당 시뮬레이션 교육을 일정 이상의 숙련도 및 교육 횟수를 달성해야 다음 단계의 교육을 실행할 수 있도록 해서 안전한 선박 교육 체계를 제공한다.
- 시뮬레이션 교육을 마친 사용자들은 울산항만의 실제 축적을 반영한 수조에서 수상드론을 원격으로 조종해보며 2번째 단계의 교육을 받는다.
- 교육의 단계가 끝나면 수상드론은 선박의 센터에 가져가 관제시스템에서 선박의 중심에 위치하고, 정보를 송수신 한다. 선박에 상하좌우에 휴대폰을 부착하여 휴대폰 4대가 송신해주는 정보와 선박 중심에서 드론이 송신해주는 정보 총 5가지의 정보를 활용해 실시간으로 관제 시스템에 선박의 모습을 그려준다. 이 정보로 좀 더 가시성 있는 관제 서비스를 제공하고, 충돌 위험이 있을 시 웹 ui를 이용해 위험 정도를 보여준다. 또한 충돌이 일어났다면 해당 정보들을 수집해 csv 파일로 제공해 사고 이후 처리를 쉽게 할 수 있도록 도와준다.

## V. 참고자료

가. 참고 및 인용자료

- 2019 스마트 해상물류 5G를 이용한 수상드론 수행계획서 및 결과보고서

[별첨] 스마트해상물류×ICT멘토링 프로젝트 산출물

## 스마트해상물류×ICT멘토링 프로젝트 산출물

1. 가상현실 시뮬레이션 .....	16
2. 수상드론 .....	19
3. 자체 컨트롤러 .....	20
4. 안드로이드 .....	21
5. 관제 시스템 .....	21
6. 핵심 소스코드 .....	23

*#소스코드 등 프로젝트 수행 중 발생한 산출물 및 달성성과 증빙 첨부*

## 1. 가상현실 시뮬레이션





# Route Select

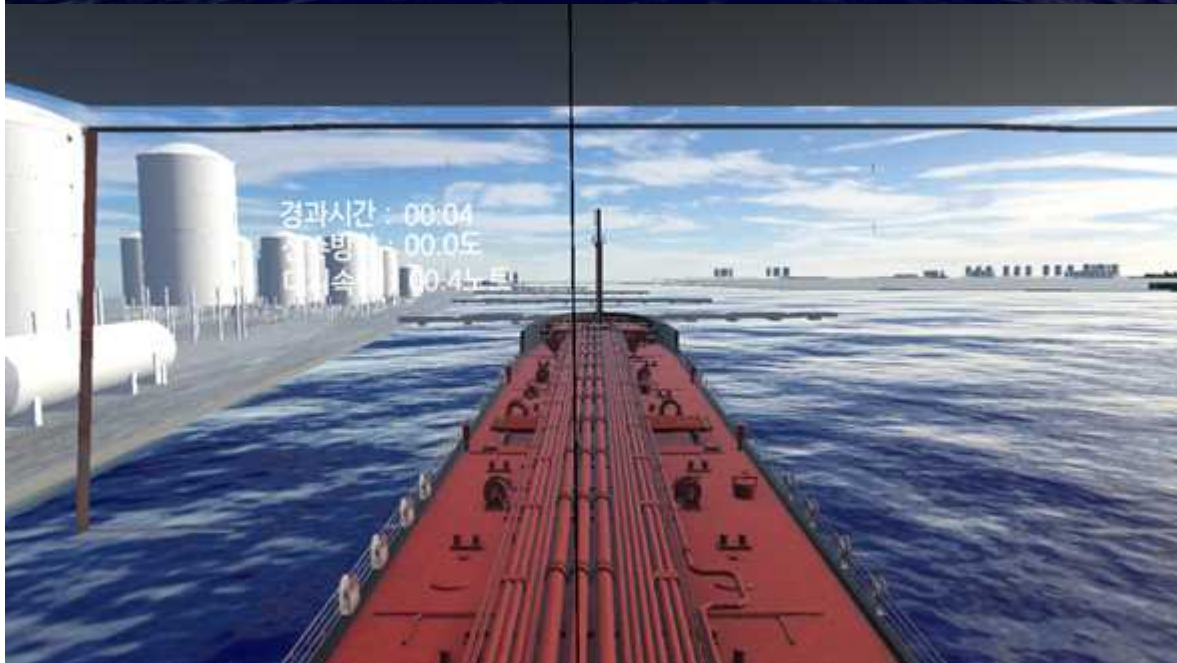
Arrival

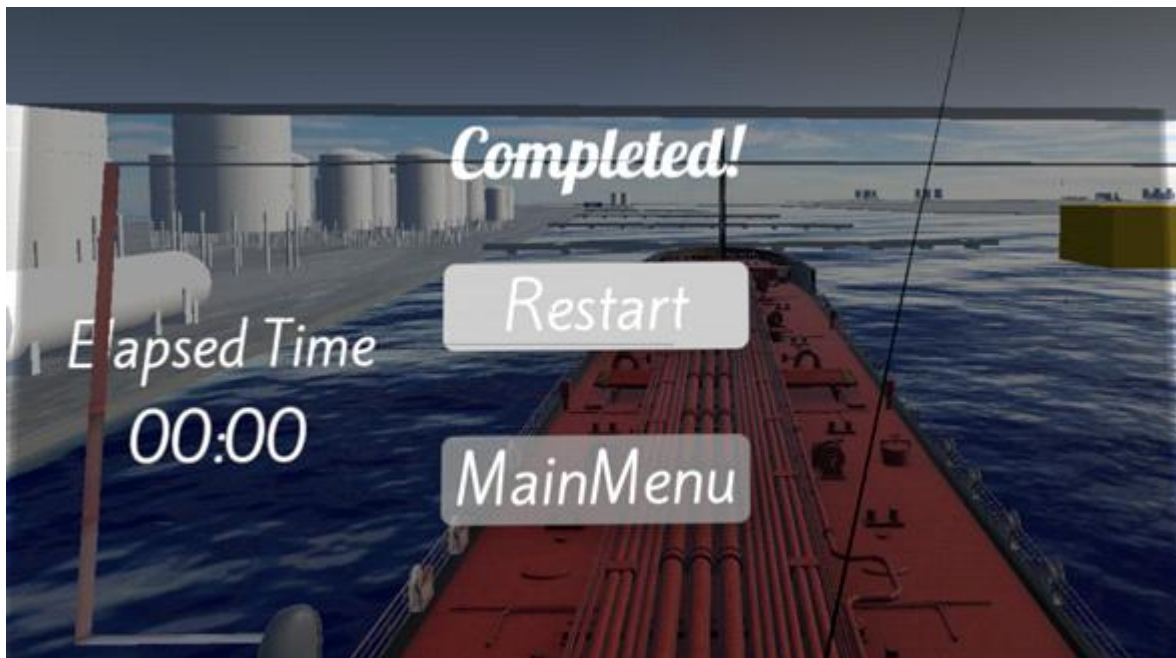
Departure

Back



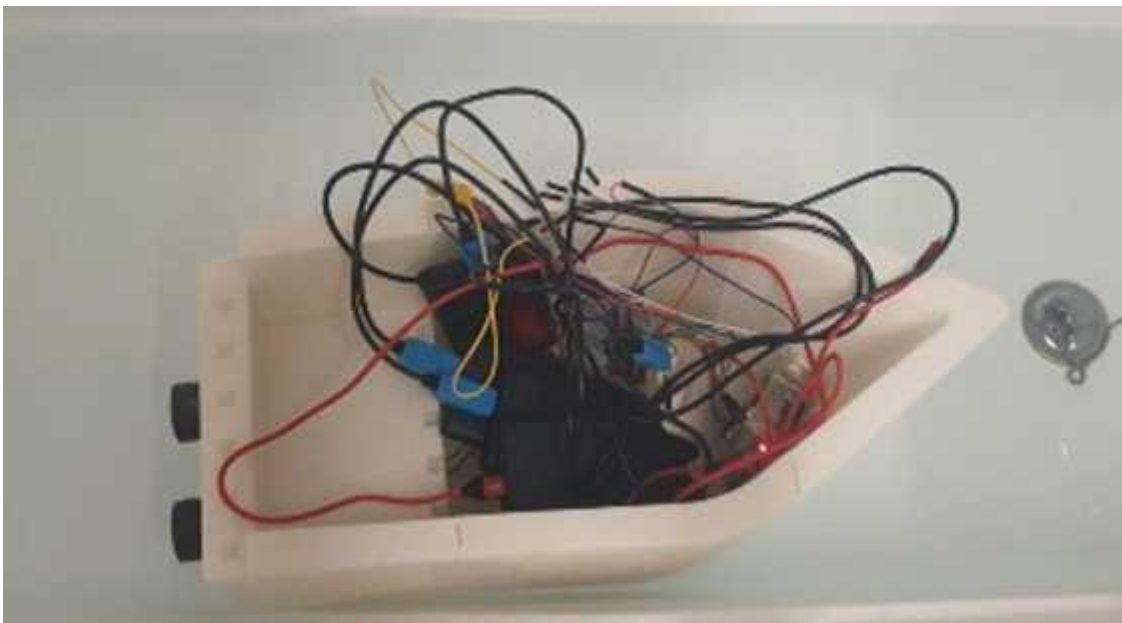
제1도선지점 --> 온산항







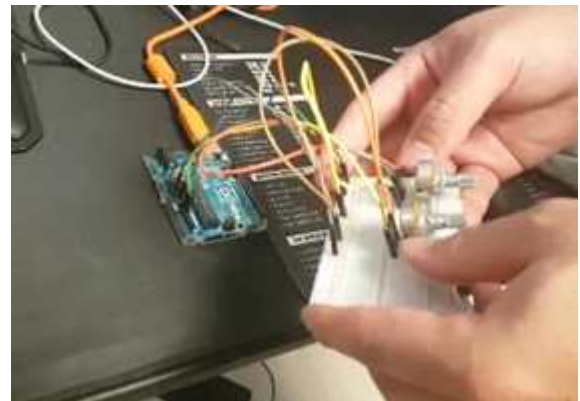
## 2. 수상드론







### 3. 자체 컨트롤러



## 4. 안드로이드



## 5. 관제 시스템





MALLANG CAU MENU

-Sorting Category-  
-Sorting Category-  
All  
Username  
time

	time	gpsX	gpsY	speed	angle
myTest	2020-07-29T14:24:20.000Z	35.4	129.3	10	10
myTest	2020-07-30T12:53:59.000Z	100	100	100	100
myTest	2020-08-01T03:21:00.000Z	1	1	1	1
test	2020-07-24T13:24:59.000Z	35.5156	129.389	1	1
test	2020-07-	35.5156	129.389	2	1

관리자로그인 | 관리자로그아웃 | 관리자로그인 | 관리자로그아웃

MALLANG CAU MENU

속도 100

증감위험 위험

lati longi

center 29.99 29.99

up 19.99 29.99

down 19.99 29.99

left 29.99 29.99

right 29.99 29.99

Example1 Example2 Example3 vesselfinder

지도 위성

Google

© 2020 Google. All rights reserved. Terms of Service

## 6. 핵심 소스코드

### – 서버

```
yyy131, 4 days ago [3 authors (yibaDoge1 and others)]
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

var indexRouter = require('./routes/index');
var userRouter = require('./routes/user');
//var mySocketRouter = require('./routes/mysocket');
var logRouter = require('./routes/log');
//var mainRouter = require('./routes/main');

var app = express();

//-----서버설정-----
// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');
app.engine('html', require('ejs').renderFile);

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
//app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

//-----http서버설정-----
var debug = require('debug')('harbor-server:server');
var http = require('http');
```

```
/**
 * Get port from environment and store in Express.
 */
var port = normalizePort(process.env.PORT || '8000');
app.set('port', port);

/**
 * Create HTTP server.
 */
var server = http.createServer(app);

/**
 * Listen on provided port, on all network interfaces.
 */
```

```

//1000
});

//드론에서 서버로 gps정보 전달(관제서)
// gpsX, gpsY, time, location(front, back, left, right, center)
socket.on('operator gps stream', function (data) {
    console.log('operator gps stream \n' + data);
    gpsDatas[data.location] = [data.gpsX, data.gpsY];
    var isFull = true;
    for(key in gpsDatas){
        if(gpsDatas[key] == null)
            isFull = false;
    }
    if(isFull){
        //서버에서 클로드로 gps정보 전달(관제서)
        io.in('opw').emit('operator gps stream', gpsDatas);
        for(key in gpsDatas){
            gpsDatas[key] = null;
        }
    }
});

socket.on('disconnect', function (data) {
    console.log("disconnected");
    if(socket.clientType == "ctl")

```

## – 시뮬레이션

```

public class GameManager : Singleton<GameManager>
{
    public Ship player { get; private set; }
    private List<MovableEntity> entities = new List<MovableEntity>();
    [SerializeField]
    private ArrivePanel arrivePanel;
    [SerializeField]
    private InfoPanel infoPanel;
    [SerializeField]
    private OptionPanel optionPanel;
    private List<MyRoute> routes = new List<MyRoute>();
    private MyRoute currentRoute;
    private bool isStart = false;
    public float elapsedTime { get; private set; }
    public static System.DateTime startedAt = System.DateTime.MinValue;
    public GameObject map;

    // Start is called before the first frame update
    protected override void Awake()
    {
        base.Awake();
        setInstance(this);
        map = GameObject.Find("Map");
        foreach (MyRoute child in map.transform.Find("Routes").transform.GetComponentsInChildren<MyRoute>())
        {
            routes.Add(child);
        }
        foreach (MovableEntity child in map.transform.GetComponentsInChildren<MovableEntity>())
        {

```



```

    }
    foreach (MovableEntity child in map.transform.GetComponentInChildren<MovableEntity>())
    {
        entities.Add(child);
    }
    elapsedTime = 0;
}

public void Start()
{
    StartGame();
}

public void Update()
{
    entities.Add(player);
    player.startControl();
    CamManager.Instance.mainCam = player.camPos;
    CamManager.Instance.uiPos = player.uiPos;
    DWP2.WaterObjectManager.Instance.Synchronize();
    elapsedTime = 0;
    startedAt = System.DateTime.Now;
    isStart = true;
    StartCoroutine("logCoroutine");
}

public void FixedUpdate()
{
    if (isStart)
        elapsedTime += Time.fixedDeltaTime;
}

public void OnArrived()
{
    if (player == null) return;
    player.stopControl();
    stopLogging();
    infoPanel.deActive();
    optionPanel.deActive();
    arrivePanel.active();
}

public void onOption()
{
    optionPanel.active();
}

public void SetPlayer(Ship ship)
{

```

```

    public void SetPlayer(Ship ship)
    {
        player = ship;
    }

    public void doLogging()
    {
        TransformDataSet dataSet = new TransformDataSet(elapsedTime);
        foreach (MovableEntity entity in entities)
        {
            dataSet.transformDatas.Add(entity.saveTransform());
        }
        //Debug.Log(dataSet.toString());
        logger.addData(dataSet);
        // Debug.Log(logger.toString());
    }
}

```

```

public void Reset()
{
    startedAt = System.DateTime.MinValue;
    isStart = false;
    elapsedTime = 0;
    foreach(MovableEntity entity in entities)
    {
        entity.destroy();
    }
    currentRoute.destination.deActive();
}

public IEnumerator logCoroutine()
{
    Logger.initLogData(startedAt, GlobalData.shipType, GlobalData.route);
    int cooldown = Mathf.RoundToInt(Logger.interval/0.02f);
    int frame = 0;
    dotLogging();
    while (true)
    {
        frame++;
        if (frame >= cooldown)
        {
            frame = 0;
            dotLogging();
        }
        yield return new WaitForFixedUpdate();
    }
}

```

## — 관제 시스템

```

// logView.html ajax table create
function logTableCreate() {
    var sortSelect = document.getElementById("logSelcBox");
    var selectVal = sortSelect.options[sortSelect.selectedIndex].value;

    // 기본실행 : 2초마다 갱신하여 출력
    // All : 페이지 갱신없이 모든 로그 출력
    if (selectVal == 0){
        $("#tableCreation").html('');
        $.ajax({
            url: '/logviewCreate',
            dataType: 'json',
            type: 'GET',
            data: {"data" : "0"},
            success: function(result){
                successFunc(result);
            }
        });
    }

    //UserName
    if (selectVal == 1){
        $("#tableCreation").html('');
        $.ajax({
            url: '/logviewCreate',
            dataType: 'json',
            type: 'GET',
            data: {"data" : "1"},
            success: function(result){
                successFunc(result);
            }
        });
    }
}

```

```

// droneSystem.html 드론 조종 소켓 통신
//TODO: 비디오 값 수신
var socket = io.connect('http://localhost:8000');
var gearVal = 0;
var angleVal = 0;

socket.on('news', function (data) {
    console.log(data.serverData);
});

// userid ??? db or websocket.id?
socket.emit('client connected',
{ clientData : '클라이언트 접속', clientType : 'ctw', userid : 'userid'});

socket.on('drone data stream', function (data) {
    $('#userid').text(data.userid);
    $('#gpsX').text(data.gpsX);
    $('#gpsY').text(data.gpsY);
    $('#speed').text(data.speed);
})

//keyboard input event
document.addEventListener('keydown', function(event){
    var controlTimeVal = new Date();
    console.log(event.keyCode);

    switch (true){
        case(event.keyCode >= 48 && event.keyCode <=51):
            //translate(gearVal, event.keyCode-48);
            gearVal = event.keyCode-48;
            $('#gearKeyInput').text(gearVal + '단');
            $('#control_gear').fadeOut(280);
            $('#control_gear').fadeIn(280);

// opSystem 소켓 통신
// TODO: Ajax , db 거치지 않고 socket 으로 화면 갱신 없는 polyline 애니메이션
var socket = io.connect('http://localhost:8000');
var areaRadius = 0;
var shipCoordinates = new Array();

var shipArea = new google.maps.Polyline({
    path: shipCoordinates,
    geodesic: false,
    strokeColor: 'FF0000',
    strokeOpacity: 1.0,
    strokeWeight: 2
});

var circle = new google.maps.Circle({
    strokeColor: 'FF0000',
    strokeOpacity: 0.8,
    strokeWeight: 1,
    fillColor: 'FF0000',
    fillOpacity: 0.2,
    center: {lat: 35.514447, lng: 129.389802}, // gpsDatas.center
    radius: areaRadius
});

socket.on('news', function (data) {
    console.log(data.serverData);
});

// userid ??? db or websocket.id?
socket.emit('client connected',
{ clientData : '클라이언트 접속', clientType : 'opw', userid : 'userid'});

```

```

function initMap() {
    var ulsan = {lat: 35.497021, lng: 129.391589};
    var map = new google.maps.Map(
        document.getElementById('map'), {zoom: 13, center: ulsan});
    //var marker = new google.maps.Marker({position: ulsan, map: map});

    socket.on('operator gps stream', function (data) {

        if (shipCoordinates.length != 0)
        {
            for(key in shipCoordinates){
                shipCoordinates[key] = null
            }
        }
        shipCoordinates = coordinate_segment(data);
        areaRadius = calcRadius(data);
        setDataQuery(data);

        circle.setMap(map);
        shipArea.setMap(map);
    })
}

// 위도 y 경도 x, 상1 우2 하3 좌4 순서
function coordinate_segment(gpsDatas) {
    var coordinates = new Array();
    coordinates[0] = new google.maps.LatLng(gpsDatas.front[1], gpsDatas.right[0]);
    coordinates[1] = new google.maps.LatLng(gpsDatas.front[1], gpsDatas.left[0]);
    coordinates[2] = new google.maps.LatLng(gpsDatas.back[1], gpsDatas.left[0]);
    coordinates[3] = new google.maps.LatLng(gpsDatas.back[1], gpsDatas.right[0]);
    coordinates[4] = new google.maps.LatLng(gpsDatas.front[1], gpsDatas.right[0]);
    return coordinates;
}

```

## – 안드로이드

```

//권한 체크
int permissionCheck = ContextCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_FINE_LOCATION);
if (permissionCheck == PackageManager.PERMISSION_GRANTED) {
    Toast.makeText(getApplicationContext(), text: "GPS 권한 있음", Toast.LENGTH_SHORT).show();
}

locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
lastKnownLocation = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
if (lastKnownLocation != null) {
    double lng = lastKnownLocation.getLongitude();
    double lat = lastKnownLocation.getLatitude();
    gpsLatitude.setText(":: " + lat);
    gpsLongitude.setText(":: " + lng);
    gpsX = String.valueOf(lat);
    gpsY = String.valueOf(lng);
}

listProviders = locationManager.getAllProviders();
if (listProviders.get(0).equals(LocationManager.GPS_PROVIDER)) {
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, minTimeMs: 5000, minDistanceM: 100, listener: this);
}

```

```
// 서버에서 안드로이드로 speed, angle, time 전송
private Emitter.Listener controlData = (args) → {
    // 시리얼 통신 -> 라즈베리파이로 speed, angle 전송
    String send;
    JSONObject receivedData = (JSONObject) args[0];
    byte[] sendByte;
    try {
        send = receivedData.getString( name: "speed");
        send = send.concat(",");
        send = send.concat(receivedData.getString( name: "angle"));
        send = send.concat(".");
        sendByte = binaryStringToByteArray(send);
        port.write(sendByte, timeout: 50);
    } catch (JSONException | IOException e) {
        e.printStackTrace();
    }
};
```

## — 하드웨어

```
# -*- coding: utf-8 -*-
import RPi.GPIO as GPIO
from time import sleep
STOP = 0
FORWARD = 1
BACKWARD = 2
CH1 = 0
CH2 = 1
OUTPUT = 1
INPUT = 0
HIGH = 1
LOW = 0
#PWM PIN
ENA = 26
ENB = 0
IN1 = 19
IN2 = 13
IN3 = 6
IN4 = 5
def setPinConfig(EN, INA, INB):
    GPIO.setup(EN, GPIO.OUT)
    GPIO.setup(INA, GPIO.OUT)
    GPIO.setup(INB, GPIO.OUT)
    pwm = GPIO.PWM(EN, 100)
    pwm.start(0)
    return pwm
```



```

def setMotorContorl(pwm, INA, INB, speed, stat):
    PWM pwm.ChangeDutyCycle(speed)
    if stat == FORWARD:
        GPIO.output(INA, HIGH)
        GPIO.output(INB, LOW)
    elif stat == BACKWORD:
        GPIO.output(INA, LOW)
        GPIO.output(INB, HIGH)
    elif stat == STOP:
        GPIO.output(INA, LOW)
        GPIO.output(INB, LOW)
def setMotor(ch, speed, stat):
    if ch == CH1:
        setMotorContorl(pwmA, IN1, IN2, speed, stat)
    else:
        setMotorContorl(pwmB, IN3, IN4, speed, stat)
GPIO.setmode(GPIO.BCM)
pwmA = setPinConfig(ENA, IN1, IN2)
pwmB = setPinConfig(ENB, IN3, IN4)
setMotor(CH1, 80, FORWARD)
setMotor(CH2, 80, FORWARD)
setMotor(CH1, 40, BACKWORD)
setMotor(CH2, 40, BACKWORD) sleep(5)
setMotor(CH1, 100, BACKWORD)
setMotor(CH2, 100, BACKWORD) sleep(5)
setMotor(CH1, 80, STOP) setMotor(CH2, 80, STOP)
GPIO.cleanup()

```