

한이음 공모전 2018 참가신청서

작품 정보		
프로젝트명	국문	VR과 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량
	영문	wireless controlled Rc car using VR and 360° Camera
프로젝트기간	2018. 4. 24 ~ 2018. 11. 30	
작 품 명	360° Avatar Driver	
작품소개	360도 카메라에서 사용되는 스티칭 기법과 VR을 결합하여 기존 영상보다 생동감 있는 영상을 사용자에게 제공한다. 제공받은 영상을 통해 사용자는 원격으로 차량을 제어한다.	
주제영역	<input type="checkbox"/> 건강 <input type="checkbox"/> 생산성 <input checked="" type="checkbox"/> 생활 <input type="checkbox"/> 안전 <input checked="" type="checkbox"/> 엔터테인먼트 <input type="checkbox"/> 기타 ()	
타 대회참가 신청수상여부	<input type="checkbox"/> 미참가 <input checked="" type="checkbox"/> 참가신청 중 <input type="checkbox"/> 수상	
임베디드 SW 경진대회/공개 SW 개발자대회/ICT 전국 스마트 디바이스 공모전		

팀 정보					
팀 명		VICER			
팀 원	이 름	소 속	부서/학과	직위/학년	
멘 토	나준규	(주)스마트동스쿨	이러닝 사업부	대표	
지도교수					
멘 티 (참여학생)	멘티 1(팀장)	박지훈	한국외국어대학교	정보통신공학과	3학년
	멘티 2	류형오	한국외국어대학교	정보통신공학과	3학년
	멘티 3	나윤호	한국외국어대학교	정보통신공학과	3학년
	멘티 4	유한석	한국외국어대학교	정보통신공학과	3학년
	멘티 5				

본인은 「한이음 공모전 2018」의 제반규정 및 유의사항을 준수하고 제출된 서류의 모든 내용에 허위 사실이 없음을 서약합니다. 또한, 공모전 심사를 위한 평가에 성실히 응할 것이며 참가자와 관련된 정보 활용에 동의합니다.

★ 개인정보 수집·이용(개인정보보호법 제15조)

- * [수집·이용목적] 한이음 공모전 및 한이음 엑스포 행사 운영/관리, 특허출원을 위한 선행기술조사, 언론홍보 및 행사안내, 한이음 사이트 등 사업 관련 자료 공개 및 홍보자료 활용
- * [수집항목] 이름, 소속, 부서/학과, 직위/학년, 전화번호, 이메일, 프로젝트 수행내용
- * [보유·이용기간] 사업 종료 후 5년

선정된 작품은 「한이음 엑스포 2018」 행사에 반드시 전시해야 하며, 전시 불참 시 평가 및 수상에서 제외됨을 확인하고 본 공모전에 참가 신청합니다.

본 참가신청서 제출 시, 위 모든 사항에 동의한 것으로 간주합니다.

2018년 8월 6 일

[붙임] 개발보고서 1부

한이음 공모전 2018 개 발 보 고 서

2018. 9. 5

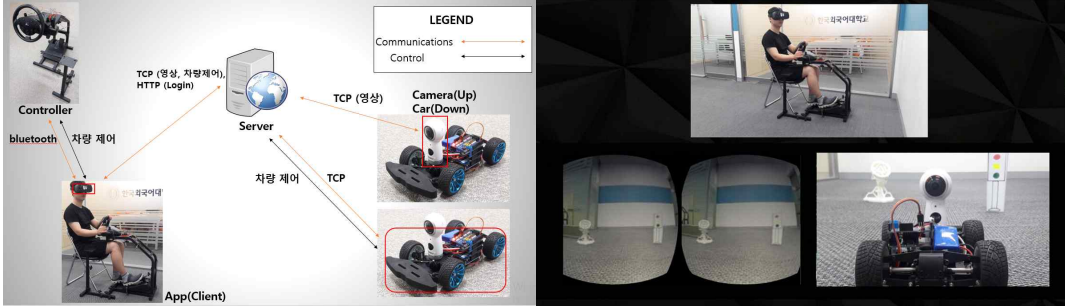
프로젝트명	국문	VR과 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량
	영문	wireless controlled Rc car using VR and 360° Camera
작 품 명	360° Avatar Driver	
신 청 자	한국외국어대학교 정보통신공학과 박지훈	

요 약 본

팀 정보

팀 명	VICER			
팀 원	이 름	소 속	부서/학과	직위/학년
멘 토	나준규	(주)스마트동스쿨	이러닝 사업부	대표
지도교수				
멘티 1(팀장)	박지훈	한국외국어대학교	정보통신공학과	3학년
멘티 2	류형오	한국외국어대학교	정보통신공학과	3학년
멘티 3	나윤희	한국외국어대학교	정보통신공학과	3학년
멘티 4	유한석	한국외국어대학교	정보통신공학과	3학년
멘티 5				



작품 정보		
프로젝트명	국문	VR과 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량
	영문	wireless controlled Rc car using VR and 360° Camera
작품명	360° Avatar Driver	
작품 소개	<ul style="list-style-type: none"> - Unity Engine을 활용하여 app device를 원격운전자에게 제공 - Samsung gear 360 SDK를 활용하여 영상 스트리밍을 구현하고 이를 gear VR를 통하여 원격운전자에게 제공 - 원격운전자는 컨트롤러(핸들, 페달, 기어)를 사용하여 실내에서 원격으로 차량을 제어 - 차량 소유주와 원격운전자에게 App 제공 - DB에 있는 차량 소유주의 일련번호를 OTP인증방법을 이용하여 원격운전자에게 제공함으로써 보안강화 	
작품 구성도		
작품의 개발배경 및 필요성	<ul style="list-style-type: none"> - 원격운전자에 대한 수요 증가 및 자율주행 한계 보완 - 원격 조정을 통해 군 지역에서 전투 및 지뢰 등으로 인한 인명피해 방지 - 후쿠시마 원전 같은 방사능 및 유해물질 발생 지역에서 작업 수행 가능 	
작품의 특징점	<ul style="list-style-type: none"> - 2D의 영상만을 제공하는 기존 무선조종 RC카와 달리 실제 화면 영상을 360도 카메라의 스티칭기법을 통하여 4D 영상을 제공하여 실제 운전하는 것과 같은 효과를 줌 - 근거리에서만 할 수 있는 기존 원격제어 제품들과 달리 거리제한 없이 지구반대편에서도 조종 가능 	
작품 기능	<ul style="list-style-type: none"> - AWS 활용 RC카와 사용자간 통신을 위한 서버를 구축 - 실시간 운전영상 송출 360도 카메라로 포착한 상황을 운전자에게 실시간으로 전달 - VR 어플리케이션 실시간 영상을 보다 현실감 있게 볼 수 있도록 VR 영상으로 제공 - 디바이스를 통한 운전제어 아두이노 가변저항센서를 이용해 핸들,엑셀,기어를 조작하고 자동차를 원격으로 제어 	
작품의 기대효과 및 활용분야	<p>기대효과: - VR을 통하여 원격제어 하므로 불필요한 준비시간을 줄일 수 있음 - 사람이 출입하기 위험한 구역에 대한 인명피해 방지</p> <p>활용분야: - 원격 제어를 통해 민간분야(대리운전, 택시 등)에 편의성을 제공 - 산업·군사 분야(원전, DMZ)의 위험성 최소화 산업·군사 분야(원전, DMZ)까지</p>	

본 문

I. 작품 개요

※ 평가항목 : 기획력 (필요성, 차별성)

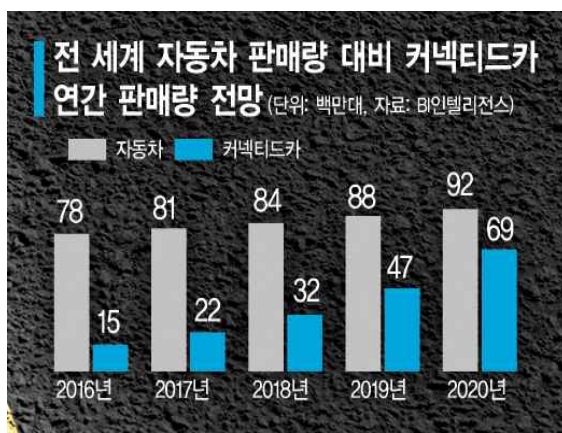
1. 작품 소개

- 사용자는 핸드폰 application을 통하여 원격운전자에게 차량의 제어에 필요한 정보를 전송함
- 원격 운전자는 서비스 사용자의 차량제어에 필요한 일련번호를 받아 application에 입력하고 자신의 컨트롤러와 360° 카메라를 연결함
- 원격 운전자는 360° 카메라를 통해 촬영된 3차원 영상을 보고 직접 컨트롤러를 제어함으로써 마치 실제로 자기가 운전하는 것처럼 느낄 수 있음

2. 작품의 개발 배경 및 필요성

○ VR 및 원격운전에 대한 수요증가

- 4차 산업혁명에 힘입어 ICT의 융합으로 차량을 이용한 원격운전 시장이 계속 확대될 것으로 예측됨
- 접하기 어려웠던 VR기기가 상용화됨에 따라 VR 하드웨어 및 콘텐츠 시장이 증가함



세계 VR 시장 현황 및 전망

(단위: 달러)



○ 자율주행의 한계 보완

- 최근 여러 회사의 자율주행 차량 테스트에서 보행자나 운전자 사망사고가 일어나는 것을 뉴스에서 많이 볼 수 있음. 이러한 자율주행 차량의 결함을 우려하는 목소리가 커져가고 있는 가운데 VR원격 제어 기술을 사용하면 이를 보완 할 수 있음

- 원격 조정을 통해 군 지역에서의 전투 및 지뢰 등으로 인한 인명피해 방지
 - 민간인은 물론 군인의 연간 지뢰 피해 현황과 지뢰 폭발사고가 계속 일어남에도 불구하고 지뢰탐지에 대한 장비가 부족함
 - 이에 인명피해를 최소화시키기 위해 순찰용 원격제어 차량을 활용하여 지뢰를 탐지할 수 있음



- 후쿠시마 원전 같은 방사능 및 유해물질 발생 지역에서 작업 수행 가능
 - 기존의 산업폐기물 현장에서 사람이 출입하여 방사능을 체크함
 - 유해물질에 대한 노출이 있는 위험구역에서 인명피해발생
 - 이런 제한적 요소들을 해결하기 위해 원격으로 제어하여 감지 할 수 잇는 차량이 필요함



< 사고지역 작업자 >



<방사능 피해자>



<원격제어용 차량>

3. 작품의 특징 및 장점

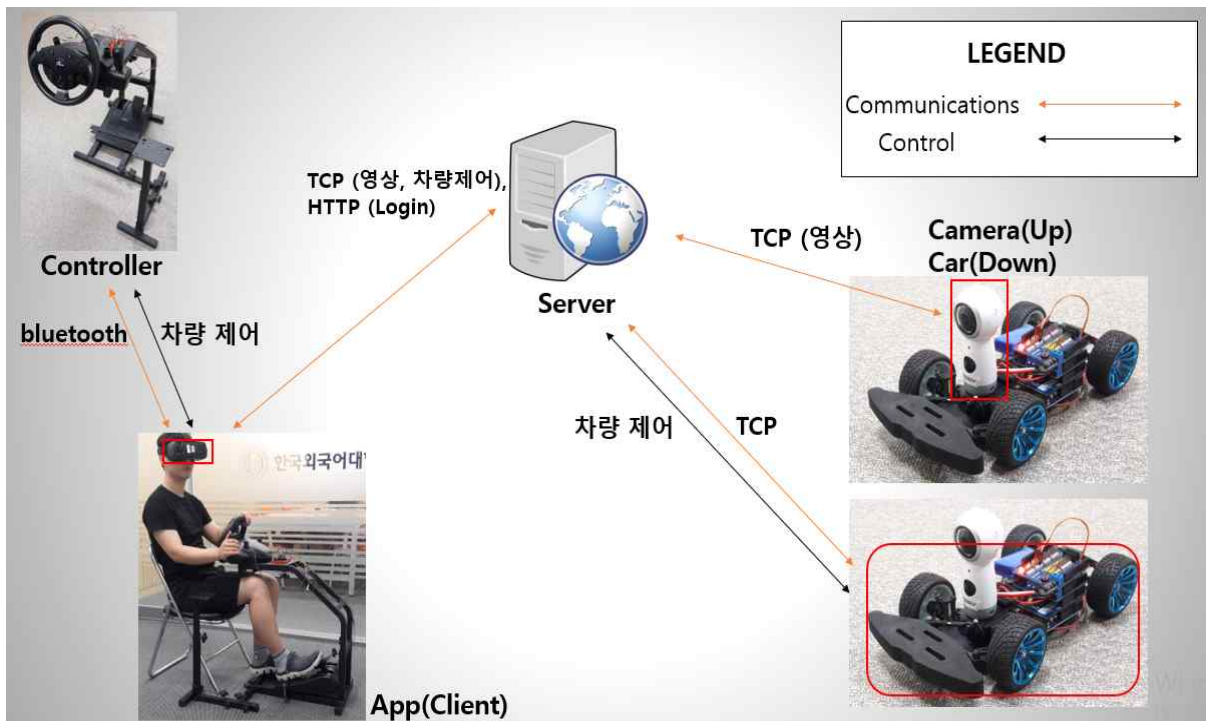
- 본 작품은 기존의 무선조종 RC카가 제공하는 2D영상의 한정된 시야를 극복하고 360도 카메라를 이용하여 4D영상을 실시간으로 송신 받아 운전자에게 사방의 상황을 제공함
- 현재 시장에 출시되어 있는 원격제어 제품(드론, 무선조종 RC카)들의 P2P 통신방식의 한계인 거리 제약을 극복하기 위해 본 작품은 소켓서버를 통한 실시간 제어를 구현함으로써 거리의 제약 없이 지구반대편에서도 이용 가능함

II. 작품 내용

※ 평가항목 : 기술력 (기능구체성, 난이도, 완성도)

1. 작품 구성도

○ 전체적인 아키텍처 구성도



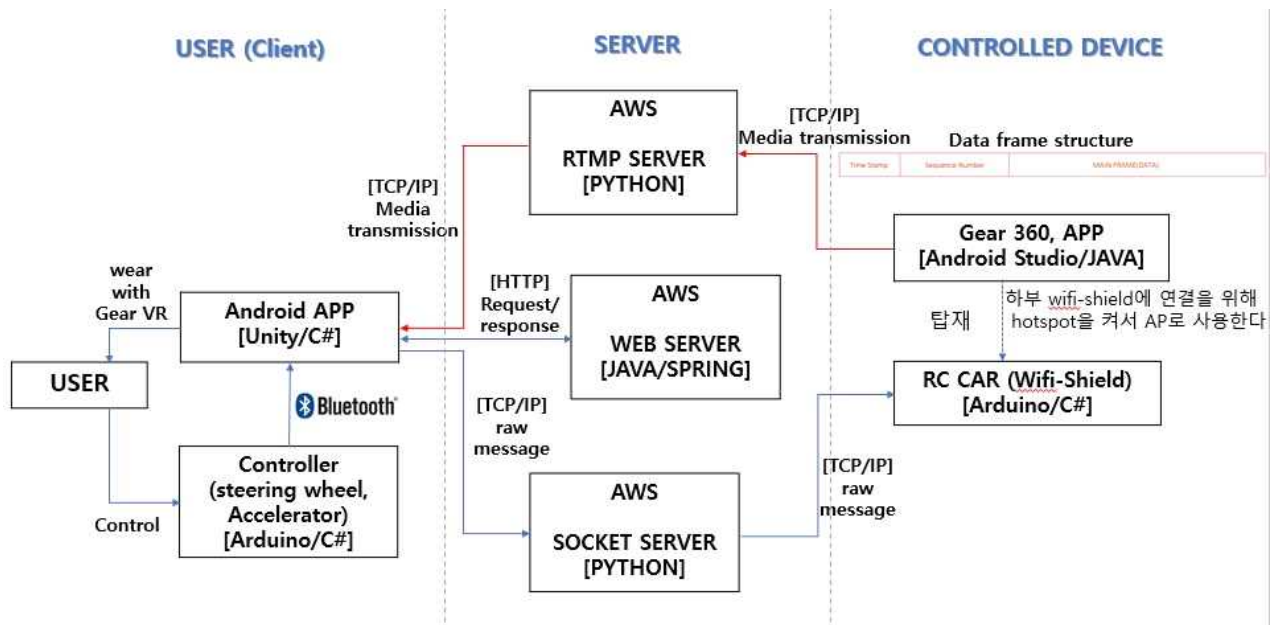
- 사용자는 App을 통해서 Controller와 블루투스로 연결한다. 핸들, 액셀, 기어를 조작하게 되면 서버를 통해서 수신된 값을 RC Car로 보내 제어를 하게 된다.
- 차량에 부착된 360도 카메라에서 촬영된 정보를 실시간으로 전송한다. 이를 App에서 수신하여 4D 영상으로 전환 된다. 만약 사용자가 옆을 돌아보게 되면 차량의 옆방향이 보이게 된다. 여기서 사용자는 실제 차량에 탑승 한 것 같은 느낌을 받게 되고 자연스러운 차량 주행이 가능해진다.

○ 실제 시연 모습 (세 사진 동시에 촬영된 사진)

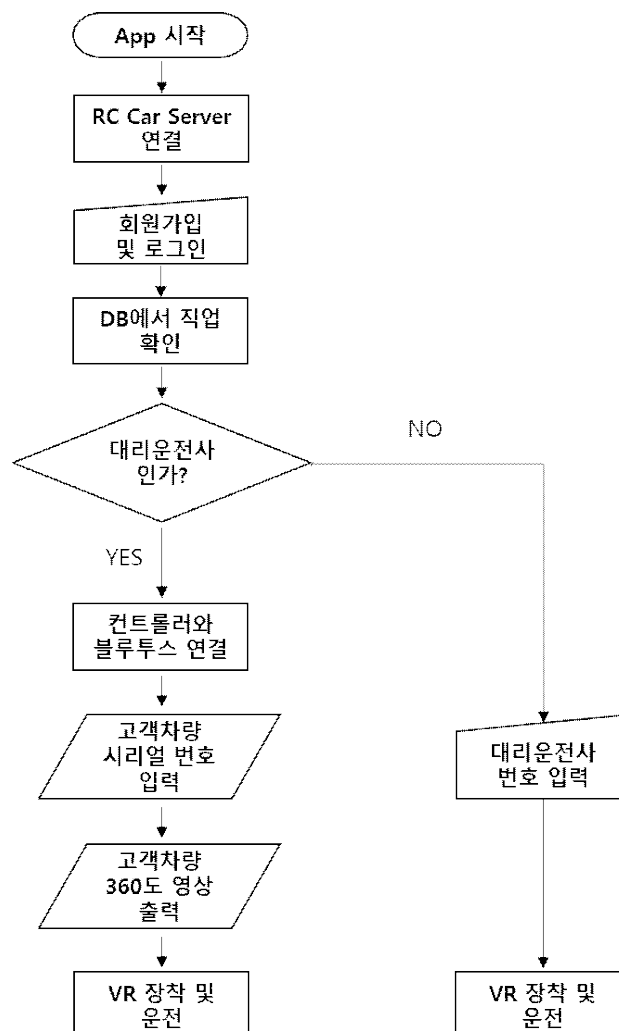


- 상단에 위치해 있는 사진은 사용자가 VR 헤드셋을 착용하고 Controller를 사용하여 주행을 하는 모습이다.
- 하단 좌측에 위치해 있는 사진은 실제 사용자가 GearVR을 통해서 보게 되는 App의 VR영상 이다.
- 하단 우측에 위치해 있는 사진은 사용자가 실제로 움직이는 차량의 모습이다.

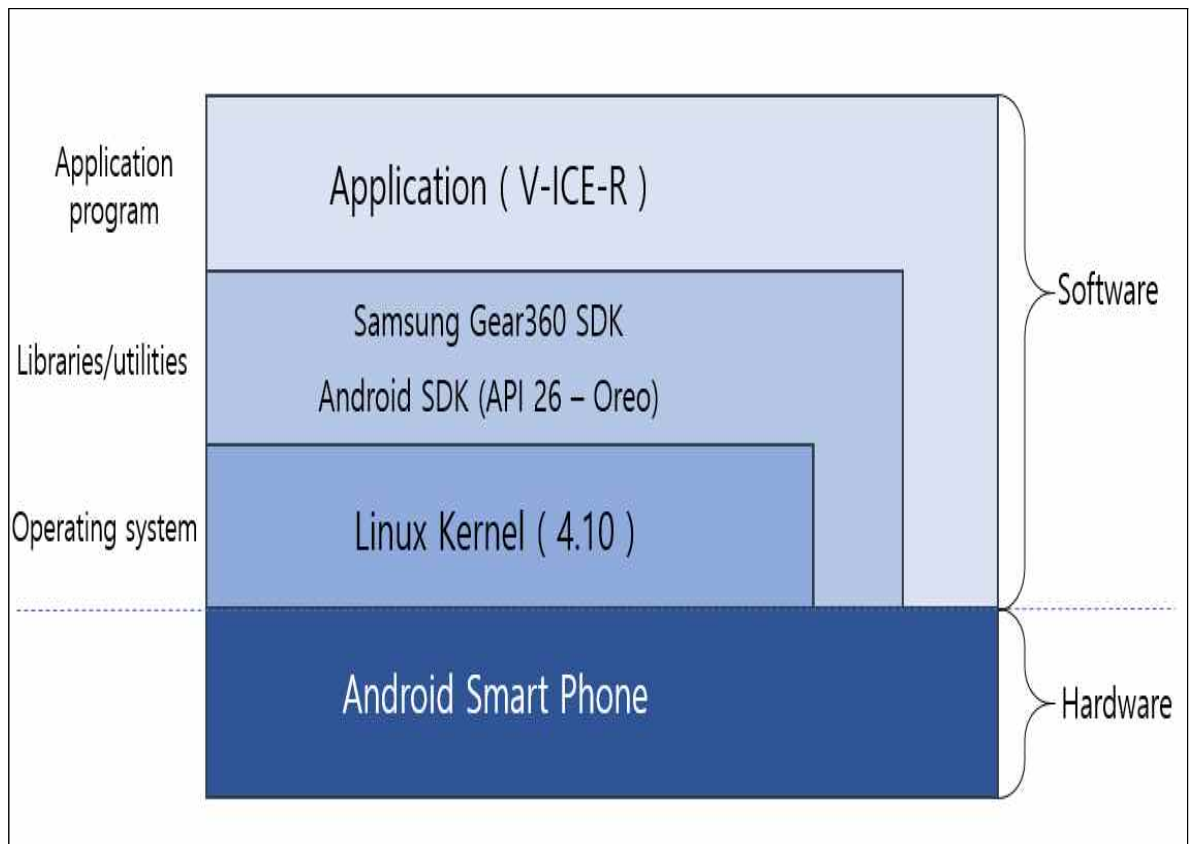
○ 통신 흐름도



○ 제품 알고리즘

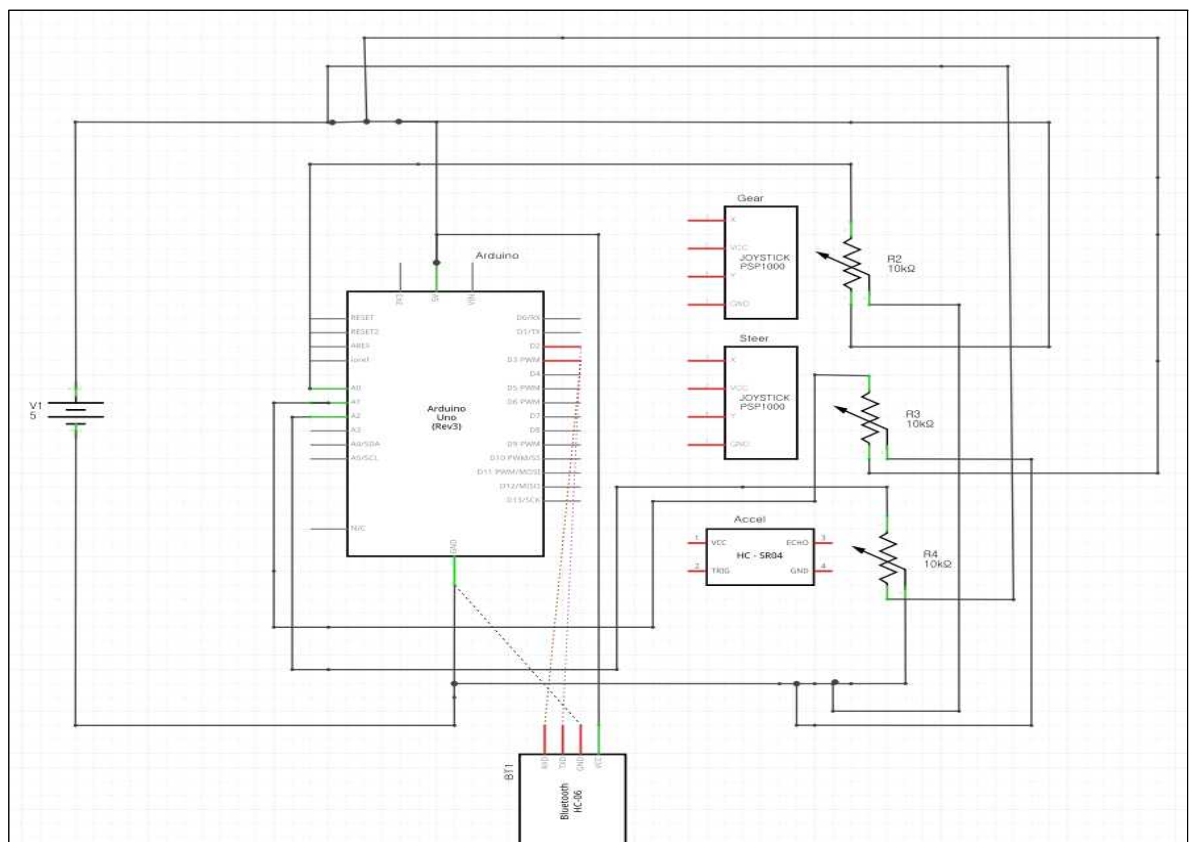


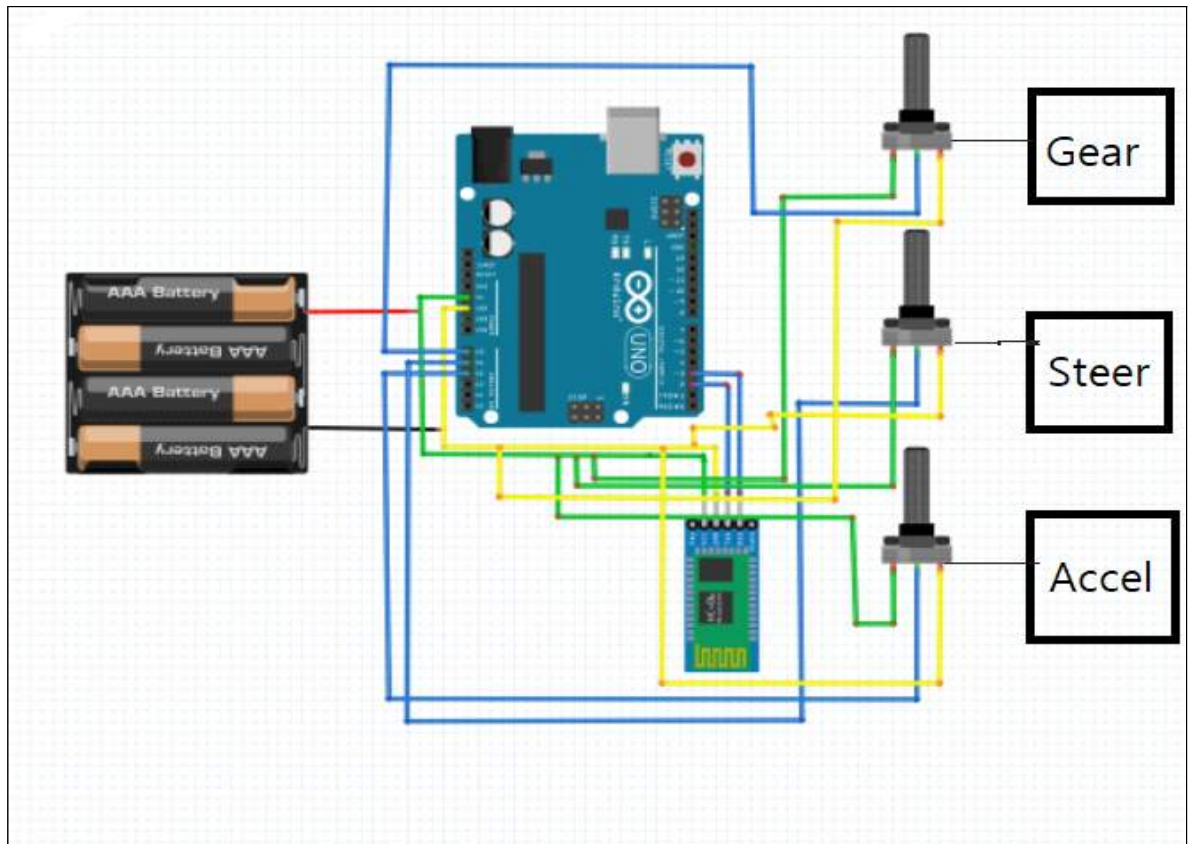
- Application Infrastructure



○ 모듈 설계서 및 회로도

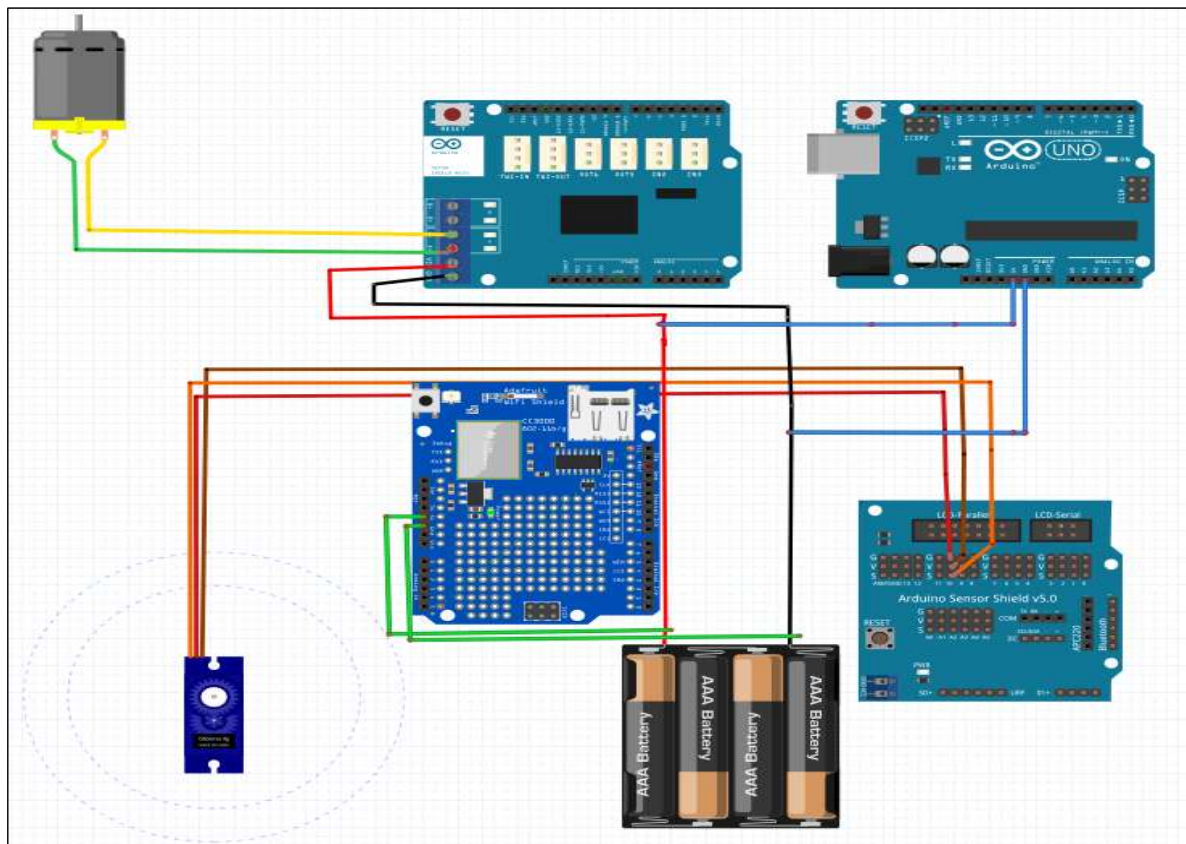
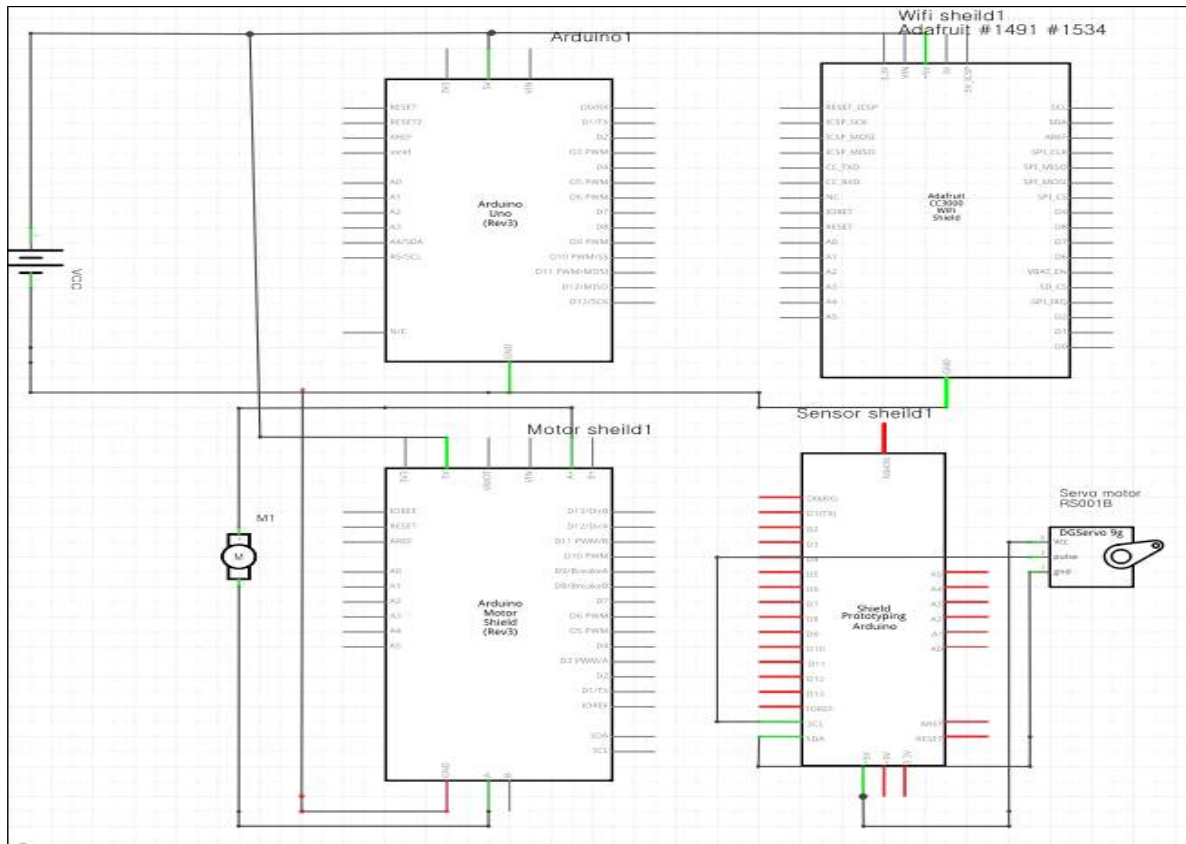
- 컨트롤러





모듈 종류	연결 핀	설명
가변저항 센서 1 (Rotary potentiometer)	Vcc	아두이노 5V의 연결
	gnd	아두이노 GND에 연결
	Signal Output	아두이노 A0에 연결
가변저항 센서 2 (Rotary potentiometer)	Vcc	아두이노 5V의 연결
	gnd	아두이노 GND에 연결
	Signal Output	아두이노 A1에 연결
가변저항기 센서 3 (Rotary potentiometer)	Vcc	아두이노 5V의 연결
	gnd	아두이노 GND에 연결
	Signal Output	아두이노 A2에 연결
블루투스 모듈 (HC-06)	vcc	아두이노 5V의 연결
	gnd	아두이노 GND에 연결
	Rx	아두이노 D2에 연결
	Tx	아두이노 D3에 연결

- RC car



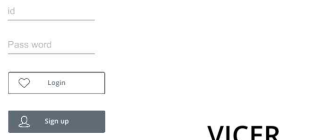




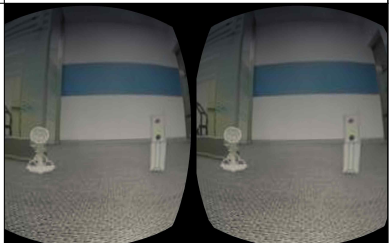
구분	장치	기능
메인 장치	아두이노 우노 (Arduino Uno)	원격 차량 조종을 위해 주변장치들 간의 통신을 제어함 블루투스 통신을 통하여 원격 차량 본체를 조종함
	배터리 (Battery)	기기들의 방전을 막기 위해 지속적으로 전력을 공급하는 배터리를 사용함
	블루투스 모듈 (HC-06)	가변저항 센서로부터 받은 문자 값을 차량과 연결된 App에 전달하기 위한 통신 인프라
주변 장치	가변저항센서 (Rotary Potentiometer)	원격 차량의 하드웨어(핸들, 기어, 페달)에 각각 부착함 센서의 회전 각도에 따라 구간을 나뉘 (0~1023) 값을 문자 값으로 변환하여 차량의 부착된 App으로 값을 전달함

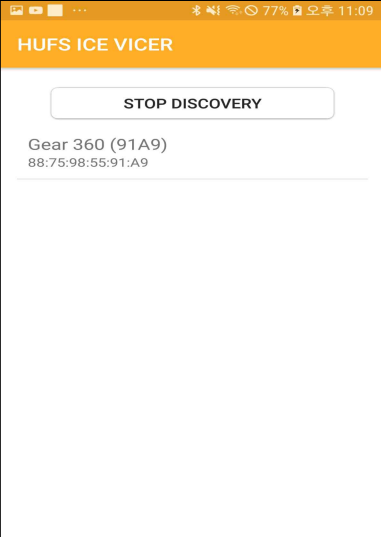
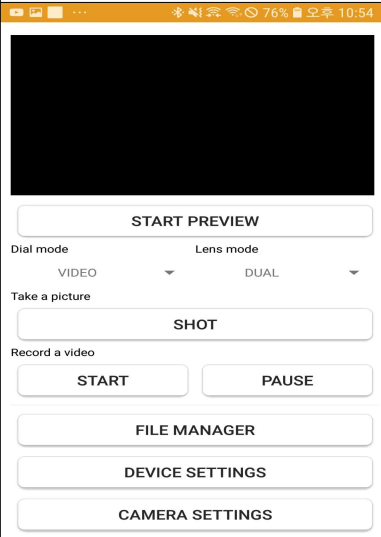

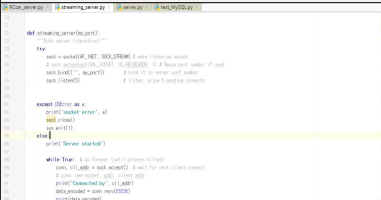
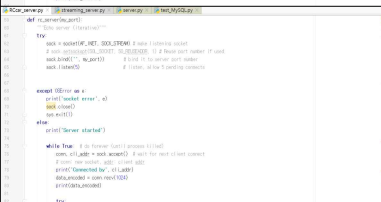
2. 작품 기능

2-1.전체기능

구분	기능	설명	현재진척도(%)
S/W	VR Application (Android App)	- 사용자에게 VR 영상을 제공 - 제어장치의 정보를 서버로 전송 - 일회용 인증번호(OTP)를 통해 보안 강화	100%
	차량용 소프트웨어 (Android app)	-360 gear와의 블루투스통신을 통해 얻은 인코딩 프레임을 디코딩, 압축하여 서버로 송신	100%
	Http Server (JAVA, SPRING)	- 사용자의 개인정보를 저장하여 로그인 시스템 구축	100%
	RC Car Socket Server (Python 3.6)	- IOT 자동차와 사용자간 통신을 위한 Server 구축	100%
	Streaming RTMP Server (Python 3.6)	- 360도 실시간 스트리밍을 위한 RTMP Server 구축	100%
	아두이노(차량)	- Socket 통신을 통해 제어 정보를 서버로부터 수신	100%
	아두이노(컨트롤러)	- bluetooth 통신을 통해 컨트롤러 정보를 app으로 송신	100%
H/W	아두이노(차량)	- 서버를 통해 수신 받은 제어 정보에 따라 작동	100%
	360도 카메라(차량 부착) 구매 H/W	- 360도 카메라로 포착한 실시간 영상을 서버로 송신	완제품
	아두이노 컨트롤러(핸들)	- 가변저항기의 값(0~1023)의 범위를 설정해 왼쪽, 오른쪽 및 페달과 같이 눌렀을 때 문자를 설정해 App을 통해 서버로 송신한다.	100%
	아두이노 컨트롤러(기어)	- 가변저항기의 값(0~1023)을 읽어 들여 후진,중립,전진을 조정할 문자를 App을 통해 송신한다.	100%
	아두이노 컨트롤러(페달)	- 가변저항기의 값(0~1023)을 읽어 들여 후진,중립,전진 값을 App을 통해 송신한다.	100%
	Gear VR 구매 H/W	- VR Application에 수신된 영상을 VR로 보기위한 H/W	완제품




2-2. S/W 주요 기능

작품	기능	설명	작품실물사진
Service App	Login	어플의 첫 화면으로써 회원가입 창으로 넘어갈 수 있으며, 아이디와 비밀번호를 입력하고 로그인 버튼을 누르게 되면, HTTP POST 통신 방식을 통하여 서버를 통해 DB의 회원 정보와 비교한 뒤 고객과 대리운전기사의 창으로 넘어가게 된다.	
	회원가입	로그인 하기 위한 계정을 생성하기 위한 창이다. HTTP POST통신 방식으로 데이터 베이스에 회원 정보를 저장한다.	
	고객이 운전자에게 인증번호 전송	서비스를 제공받는 고객이 로그인하면 넘어오는 창이다. 회원가입시 입력한 차량의 시리얼번호와 운전자의 번호를 입력하게 되면 서버로 정보를 넘겨서 운전자가 고객의 차량을 제어할 수 있는 권한을 얻게 된다.	
	대리기사의 시리얼번호 인증	운전자가 로그인하게 되면 나오는 창으로써 고객에게 받은 차량의 시리얼 넘버를 입력하게 되면 차량의 360도 카메라와 연결된다.	
	블루투스 connection	시리얼 넘버를 인증하고 난 뒤에 운전자는 차량을 제어하기 위해 자신의 컨트롤러와 차량을 연결한다. 블루투스 통신을 이용하여 운전자의 어플과 연결된 후에 socket server를 통하여 원격으로 차량을 제어 할 수 있게 된다.	
	VR 스트리밍 창	360도 카메라와, 컨트롤러를 모두 연결하여 고객의 차량을 제어할 준비가 되면 360도 카메라에서 송신하는 영상을 운전자의 application을 통하여 볼 수 있게 된다.	

차량용 수퍼트레이너 (Android app)	gear 360 detection activity	bluetooth 통신으로 페어링 되어 있는 Gear360을 찾고, 연결을 누르면 다음 Activity로 넘어간다.	
	gear 360 control activity	Gear360의 설정, 캡처, 동영상 촬영등과 함께 StartStreaming을 클릭하면 서버로 영상을 송신한다.	
Http Server (JAVA, SPRING)	회원가입, 로그인, 차량 시리얼 인증 Server	Spring framework를 사용하여 회원가입과 로그인 서비스를 구현 하였고 대리 운전기사로 로그인 했을 때에는 고객의 차량의 시리얼 번호를 입력하여 해당 고객의 차량을 운전 할 수 있도록 구현했다.	
RC Car Socket Server (Python 3.6)	RC Car Socket Server	Python3으로 개발된 TCP 서버로 Client(App)에서 전송된 차량 제어에 대한 데이터를 수신하여 이를 차량에 송신해준다.	
Streaming TCP Server (Python 3.6)	TCP Server	Python3으로 개발된 TCP서버로 차량에 부착된 카메라(Gear360)에서 촬영, 가공된 영상을 수신하여 이를 Client(App)에 전송한다.	

Arduino	차량	RC Car Server로부터 정보를 받아 RC Car가 작동할 수 있도록 구현했다.	<pre>#include <Servo.h> #include <Servo.h> #include <EEPROM.h> #include <SPI.h> #include <Phone.h> SoftwareSerial mySerial(10, 11); // RX, TX Servo myservo; char server_name[] = "52.79.181.10"; PhoneClient client; void setup() { Serial.begin(9600); Middle_Cal_Val = EEPROM.read(Middle_Cal_Addr); //EEPROM 보정값이 저장되지 않았다면 초기값인 90으로 저장한다 if ((Middle_Cal_Val == 0x00) (Middle_Cal_Val == 0xFF)) { Middle_Cal_Val = 90; EEPROM.write(Middle_Cal_Addr, Middle_Cal_Val); delay(10); } Middle_Val = Middle_Cal_Val; myservo.attach(9); myservo.write(Middle_Cal_Val); Serial.begin(9600); pinMode(D18A, OUTPUT); pinMode(PWMA, OUTPUT); no_Motor(FORWARD, 0); Phone.begin(TX_LOD_SPI PF_LOD_NET); if (client.connect(server_name, 5501)) { Serial.print("Connected to server"); client.print("C"); } else { Serial.print("connection failed"); } } //RC Car Servo Motor SoftwareSerial BTSerial(2, 3); // 2 : TX, 3 : RX String myString = ""; int temp = 510; bool a = true; bool gearccw = false; bool gearrev = false; void setup() { // put your setup code here, to run once: Serial.begin(9600); BTSerial.begin(9600); } void loop() { if (a) { BTSerial.print("S"); a = false; } int gear = analogRead(A0); int steer = analogRead(A1); int eccol = analogRead(A2); Serial.print(gear); if (gear < 325) { gearccw = true; gearreverse = false; } else if (gear > 370) { gearccw = false; gearreverse = true; } else { gearccw = false; gearreverse = false; } if (gearccw)</pre>
	컨트롤러	컨트롤러를 통하여 움직임 정보를 블루투스로 연결된 Service App으로 전달하도록 구현했다.	

2-3. H/W 주요 기능

기능/부품	설명	작품실물사진
컨트롤러 (아두이노)	가변 저항을 이용하여 오른쪽 또는 왼쪽으로 돌아갈 때 어느 쪽으로 돌아가는지 알 수 있게 하였다.	
	가변 저항을 이용하여 차량의 D(전진 모드), R(후진 모드), N(중립)을 구현하였다.	
	가변 저항을 이용하여 차량의 전진, 후진을 할 수 있도록 구현하였다.	

Gear VR	360도 영상을 VR로 볼 수 있게 해준다.	
360도 카메라	RC 카에 부착되어 360도 영상을 제공한다.	
RC카(아두이노)	wi-fi 실드를 탑재하여 무선으로 조종이 가능하다.	

3. 주요 적용 기술

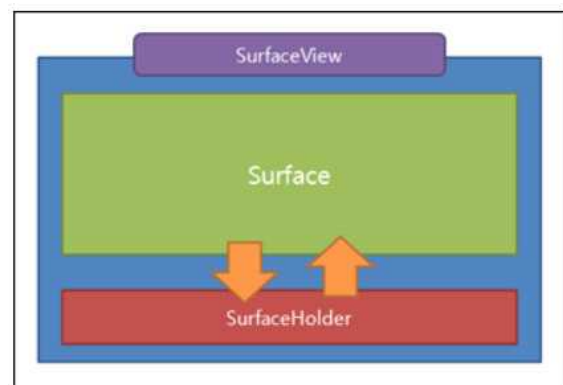
- Gear360 SDK



Gear360은 전후 2개의 195도의 큰 화각을 갖는 카메라를 통해 얻는 화상을 이어 붙이는 스티칭 (Stitching) 기술을 통해 접합 시켜줌으로, 과거에 보통 HDR 파노라마 이미지나, 환경 맵 제작을 위해서 널리 사용되었던 방법을 간소하게 구현한 것이다.



크롭볼을 활용한 촬영 방식, 출처 :
<http://www.diyphotography.net/turn-them-christmas-ornaments-to-a-360-fisheye-lens/>



[출처] 커니의 안드로이드 이야기 <http://androidhuman.tistory.com/307>

기존에는 큰 화각을 촬영하기 위해서 크롭볼을 활용하는 등의 방식이 사용되어 왔지만, 이는 이미지 손상이 심했고, 촬영 방식도 매우 까다로웠었다. 하지만 최근에 큰 화각을 촬영할 수 있는 렌즈들이 개발되면서 손쉽게 큰 화각 이미지의 촬영이 가능해졌고 이를 Software 상(SDK)으로 스티칭 해줌으로써 손쉬운 360도 촬영이 가능해 졌다.

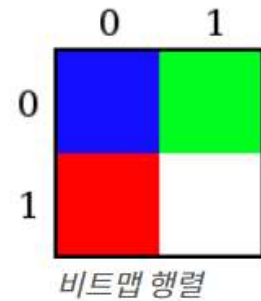
촬영후의 스티칭 처리까지 완료된 프레임을 SDK에서 제공하는 startStreaming 메서드에서 Android상의 SurfaceView로 렌더링 되기 바로 직전에 (SurfaceView 는 더블 버퍼링 방식으로 Surface로 먼저 스케치 한 후에 SurfaceView 에 그려진다) 프

레이름을 뽑아낸 후에 이를 디코딩,인코딩 해주어 변환된 Byte Array를 서버로 전송 해준다.

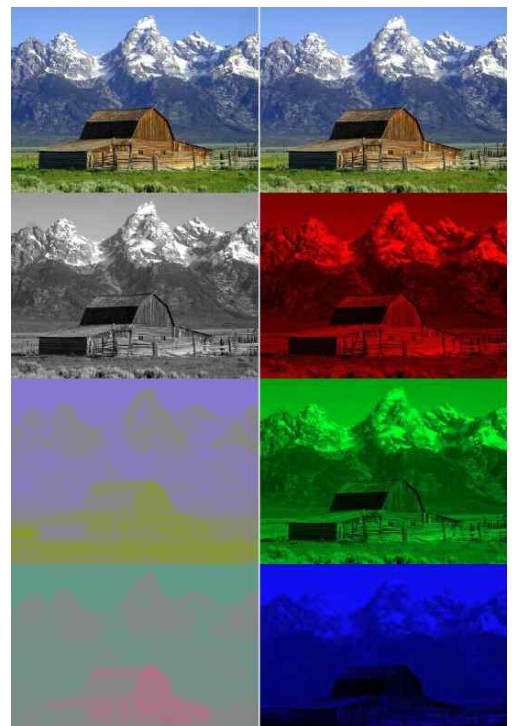
○ 영상 데이터 처리

- Bitmap

비트맵은 R,G,B 세가지 채널로 구성되는데 이들은0~255 까지 색상 정보를 담고 있는 하나의 픽셀(Pixel)이다. 즉, R,G,B 세가지 행렬이 겹쳐져서 보이는 것이 실제 사람이 접하게 되는 이미지인 것이다.



YUV는 Video Data형식으로 사용하며 TV에서 컬러페이스(컬러 서브 캐리어의 주파수와 진폭을 이용하여 2차원 컬러를 보내는 것)로 전송하는 컬러 형식이다. YUV는 비디오 시스템에서 아날로그 휘도신호와 색차신호를 말할 때 사용하는 약어로 Y는 휘도 신호, U V는 PAL방식에서 사용되는 2개의 부반송파(SubCarrier) 중심축을 뜻한다. YUV는 색과 빛이 따로 구성되어있는 형식으로 일반 RGB 형식에 비하여 작은 대역폭으로 전송이 가능한 장점이 있다.



좌측 YUV 우측 RGB

Y : 명암(밝기)

U(Cb) : 명암에서 파란색(B) 성분.

V(Cr) : 명암에서 붉은색(R) 성분.

YUV의 종류로는 YUV420 ,YUV422 등이 있다.

현재 제품에 적용된 기술은 YUV420

형식이다.

출처 : burnleyandpendle's blog

- YCbCr

YUV와 비슷하지만 YCbCr은 디지털 방식이라는 차이가 있다. 디지털 통신에서는 약간 변형한 방식을 이용한다.

- YUV to RGB (Decoding , Encoding)

BT.601(아날로그 비디오 신호를 디지털 비디오로 인코딩하기 위한 형태)은 다음과 같은 상수를 정의 한다.

$$\begin{aligned}W_R &= 0.299 \\W_G &= 1 - W_R - W_B = 0.587 \\W_B &= 0.114 \\U_{\text{Max}} &= 0.436 \\V_{\text{Max}} &= 0.615\end{aligned}$$

Y'UV는 다음과 같이 RGB로부터 계산된다.

$$\begin{aligned}Y' &= W_R R + W_G G + W_B B = 0.299R + 0.587G + 0.114B \\U &= U_{\text{Max}} \frac{B - Y'}{1 - W_B} \approx 0.492(B - Y') \\V &= V_{\text{Max}} \frac{R - Y'}{1 - W_R} \approx 0.877(R - Y')\end{aligned}$$

Y', U 및 V의 결과 범위는 [0, 1], [- U Max , U Max] 및 [-V Max , V Max]이다. 위의 변환을 반전하면 Y'UV가 RGB로 변환된다.

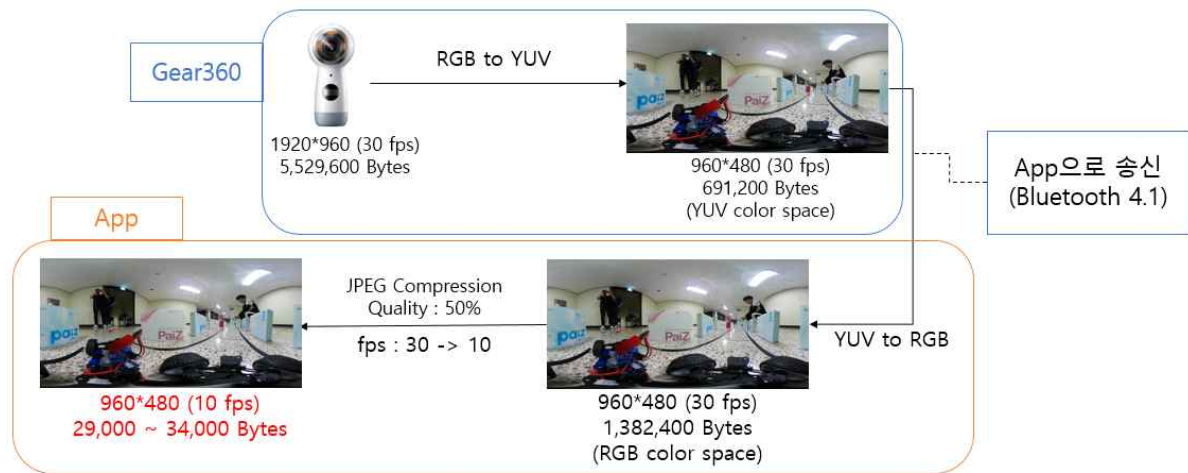
$$\begin{aligned}R &= Y' + V \frac{1 - W_R}{V_{\text{Max}}} = Y' + \frac{V}{0.877} = Y' + 1.14V \\G &= Y' - U \frac{W_B(1 - W_B)}{U_{\text{Max}} W_G} - V \frac{W_R(1 - W_R)}{V_{\text{Max}} W_G} \\&= Y' - \frac{0.232U}{0.587} - \frac{0.341V}{0.587} = Y' - 0.395U - 0.581V \\B &= Y' + U \frac{1 - W_B}{U_{\text{Max}}} = Y' + \frac{U}{0.492} = Y' + 2.033U\end{aligned}$$

이를 행렬 식으로 표현하게 되면 다음과 같은 Encoding, Decoding 식을 얻는다.

$$\begin{aligned}\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} &= \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \\ \begin{bmatrix} R \\ G \\ B \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 1.13983 \\ 1 & -0.39465 & -0.58060 \\ 1 & 2.03211 & 0 \end{bmatrix} \begin{bmatrix} Y' \\ U \\ V \end{bmatrix}\end{aligned}$$

○ 상황별 영상 데이터 크기 및 Encoding/Decoding Image Frame

영상 프레임



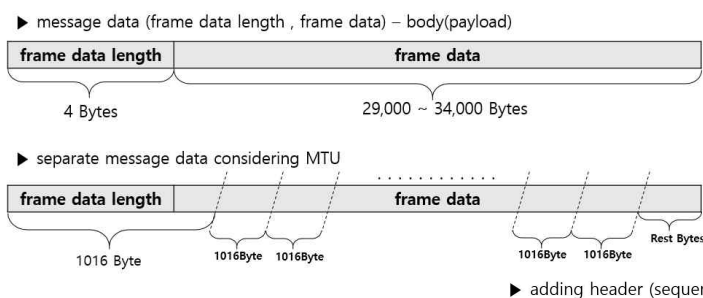
- Gear360에서는 RGB->YUV Converting을 통해 App으로 Bluetooth 송신을 하기위해 데이터 크기를 1/8로 압축시켜 주었다.
- App에서는 1차로 YUV frame을 RGB-Bitmap으로 변환 시켜주고 JPEG Commpression(Quality : 50%)을 통해 최종 데이터 크기를 30KB정도로 압축시켜 주었다. (만약의 Traffic Congestion을 대비해서 fps를 30에서 10으로 낮춰주었다)

※ 영상 송수신 기술은 UDP, TCP 두 가지의 방식으로 구현 하였다.

■ 영상 송수신 기술1(UDP)

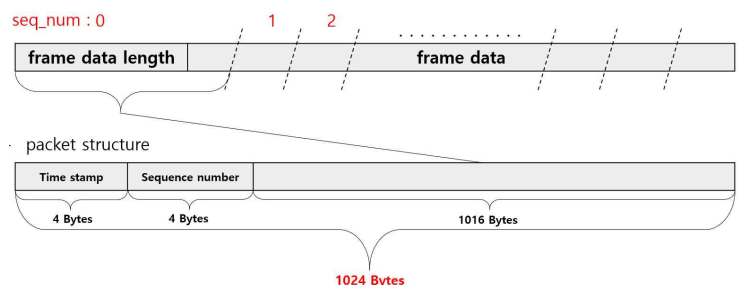
○ Packet 분할 protocol

- 위에서 확인 했듯이 최종 데이터는 30KB 정도이다.
- Ethernet의 MTU가 1500Byte인 것을 고려하여 원활한 통신을 위해 한 Packet의 사이즈를 1024Byte 로 맞춰 준다.



- 1) 수신 측에 data 길이를 파악하기위해 data length (4Byte)를 header로 붙여준다.
- 2) MTU를 고려하여 message data를 1016Byte로 잘라준다.

- 3) 수신 측에서 Packet 조합을 위해 frame 식별을 위한 'Time Stamp' 와 message data 복원을 위한 'Sequence Number' 를 header로 추가해준다.

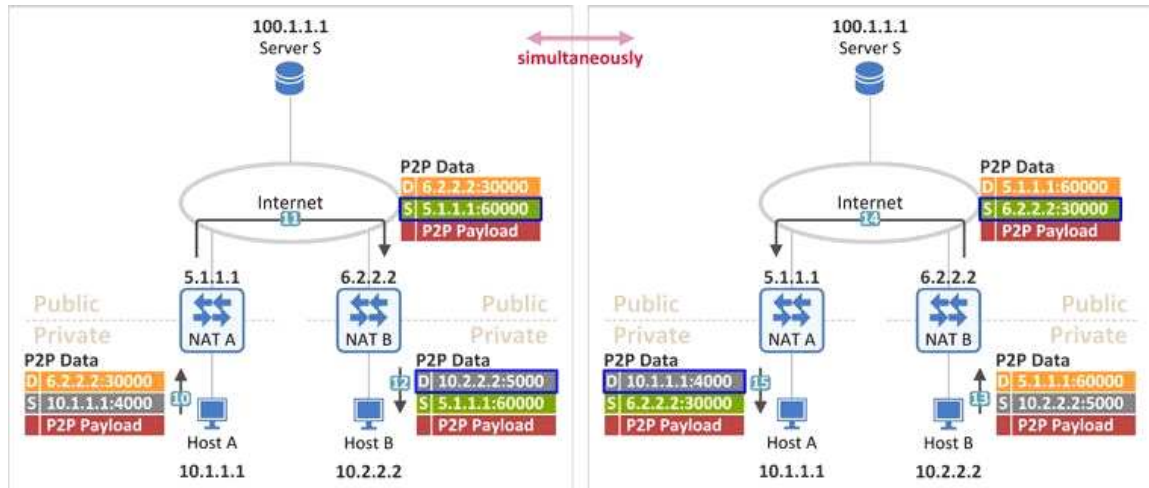


○ 영상 송·수신client간 원격 p2p연결

NAT gateway를 통과하는 UDP connection을 연결하고 유지를 하기위한 기술인 NAT Traversal 방식중에 리얼타임에 가장 적합한 UDP Hole Punching을 사용했다.

- UDP Hole Punching

서로 다른 NAT으로 막혀 있는 두peer들 사이에 p2p통신을 하기위한 방법이다. 처음에 UDP 통신용 Connection을 열기 위해 공인ip서버를 이용했다.



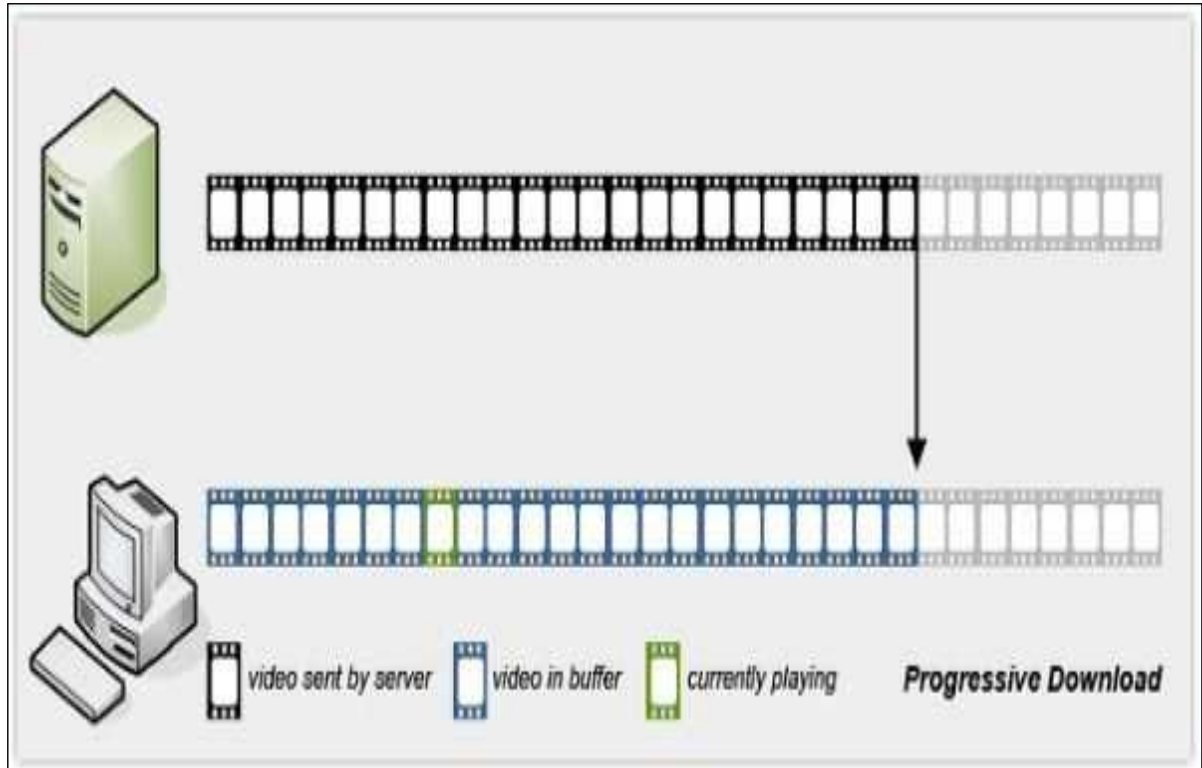
위의 그림은 서로 다른 NAT에 있는 HOST끼리 원격 P2P연결로 데이터를 주고 받는 것을 볼 수 있다.

연결 과정은 다음과 같다.

1. 공인ip서버로 각자 HOST의 private 정보와 NAT의 정보를 보내준다.
2. 서버가 양쪽 HOST의 정보를 알고 있는 상태에서 한쪽 HOST가 서버로 다른 HOST의 Connection request를 보낸다.
3. Connection request를 받은 서버는 각자의 HOST로 상대의 HOST정보를 보내 준다.
4. 각자의 HOST는 상대방의 public, private정보를 알게 된다.
5. P2P 데이터 통신을 거의 동시에 시도 할 수 있게 된다.

■ 영상 송수신 기술2(TCP)

○ TCP 송수신 기술



영상 송수신 기술은 세상에 널리 알려진 streaming protocol중에 RTSP protocol을 만들려고 시도하였으나 너무 복잡하여서 독자적으로 비슷하게 TCP protocol상으로 만들었다. 방식은 위 그림에서 보이는 RTSP protocol을 참조 하였다.

찍은 영상의 프레임의 길이를 측정하여 미리 초기화해둔 4byte크기의 byte배열에 저장 후 Receiver에게 해당 프레임의 길이를 byte배열로 보낸다. 그 후 해당영상의 프레임을 pngbyte로 Encoding하여 Receiver에게 보낸다. Receiver는 처음에 받은 길이만큼의 byte가 들어오게 되면 한화면의 프레임을 인식하여 화면에 띄워주게 된다.

4. 작품 개발 환경

구분		상세내용
S/W 개발환경	OS	Windows 7
	개발환경(IDE)	Unity engine, Eclipse Oxygen, Android Studio, STS, Visual Studio, Sketch(Arduino)
	개발도구	SQL Developer, MyBatis, Maven, Gradle
	개발언어	JAVA, XML, SQL, JSP, JavaScript, JSON, C#
	기타사항	
H/W 구성장비	디바이스	Arduino, Wi-fi shield, Blue Tooth module(HC-06), RC car, Gear 360
	센서	Variable resister,
	통신	Blue tooth
	언어	C#
	기타사항	
프로젝트 관리환경	형상관리	Git Hub (Source Tree)
	의사소통관리	SNS (카카오톡), 문자, 전화, Team Viewer
	기타사항	

5. 기타 사항 [본문에서 표현되지 못한 작품의 가치(Value)] 및 제작 노력

○차별화된 성능

- 누구나 쉽게 접할 수 있게끔 스마트폰 어플리케이션의 개발로 진입장벽을 낮추었다.
- Oculus Utility(Unity)를 활용하여 VR영상 제공한다.
- Samsung gear 360 SDK를 활용한 실시간 영상 스트리밍을 구현한다.
- 기존의 웹캠이 전송하는 평면 이미지가 아니라 영상을 360도 카메라의 스티칭기법을 사용하여 3차원 이미지를 송·수신 한다.
- P2P통신을 대체한 Socket Server/Client를 통신에서의 거리 제약을 해결하였다.

○신뢰성

- 다른 통합 개발 환경으로 재현이 불가능한 VR영상을 세계에서 검증된 Unity Engine을 통해 구현하였다.
- 자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크인 Spring을 사용하여 안드로이드와 연동을 통해 데이터베이스로 접근 시 보안적인 이슈를 해결 할 수 있으며 쉽게 App과 Web간의 통신이 가능하다.

○사용성

- 무인으로 조종되는 자율 주행 자동차나 AI를 이용하면 긴박한 순간에서의 처리가 학습된 알고리즘으로 수행되기 때문에 큰 문제를 일으킬 수 있다. 사람이 접근할 수는 없지만 결정적인 순간에 사람의 생각이 개입되어야 하는 부분에 우리 제품이 사용될 수 있다.
예를 들면 군사용 목적으로 사람이 출입할 수 없는 곳에서의 사용과 원전과 반도체 공장과 같이 사람에게 큰 피해를 줄 수 있어 사람의 출입이 제한되어야 하는 상황 속에서 필요하다.

○알고리즘의 창작성

- TCP 스트리밍 서버 : 두개의 semaphore 를 활용한 spinlock을 채택하여 보다 높은 성능의 Multi threading synchronization을 구현하였다.
- 기존 Streaming Protocol인 RTSP Protocol을 참조하여 TCP Protocol통신상에서 영상 Streaming을 구현하였다.

○데이터와 프로그램의 가치

- 원활히 개발이 이루어지지 않은 스티칭된 360° 영상을 최대한 압축하여 송신해줌으로써 기존에 없는 360° 영상 스트리밍이 가능해졌다.

○작품을 통한 가치창출

- 본 프로젝트의 시작은 귀가하는 버스에서 출근하는 대리운전기사와 아들과의 통화를 우연찮게 들음으로써 시작되었다. 가장으로서 삶의 애환이 묻어 나오는 것을 느꼈다. 수척해진 모습이지만 아들을 위해 근무지로 나가시는 모습을 보니 가슴이 뭉클했다. IT를 전공하는 학생으로서 조금이나마 도움이 되고 싶어서 원격제어를 이용한 작품을 생각하게 되었다. 본 작품을 이용한다면 대리기사를 비롯한 동종업계 사람들에게 복지여건을 개선하고 자택근무를 가능하게 하는 편의성을 제공한다.

가족과 함께하는 시간을 늘림으로써 가족과의 화합, 소통의 가치를 창출한다고 생각한다. 더 나아가 본 작품의 장점인 원격제어를 통해 군사 분야와 산업분야에 대해서도 가치를 창출할 수 있다고 생각한다. 2015년 08월 4일에 일어난 모함 지뢰 사건에서도 볼 수 있듯이 사람이 직접 순찰해야 하는 곳에 원격제어를 통해 인명피해를 막을 수 있다. 또한 사람이 주기적으로 방사능 체크를 위해 출입해야 하지만 출입해서는 안 되는 지역인 후쿠시마나 체르노빌에 대해서도 인명피해를 방지할 수 있는 안전의 가치를 창출할 수 있다고 생각한다. 기술적인 면에서도 시행되지 않았던 4D 환경을 360도 카메라의 실시간 스트리밍을 통해 VR과 접목시켜 4차 산업혁명 시대에 앞서 경제적 가치를 넘어 사회적 가치를 창출 해낼 수 있다.

III. 프로젝트 수행 내용

※ 평가항목 : 수행능력 (문제해결능력, 수행충실성)

1. 멘티(참여학생) 업무분장

번호	이름	대학	학과	학년	역할	담당업무
1	박지훈	한국외국어대학교	정보통신공학과	3학년	팀장	전체 프로젝트 관리 및 개발
2	류형오	한국외국어대학교	정보통신공학과	3학년	팀원	VR Application 개발
3	유한석	한국외국어대학교	정보통신공학과	3학년	팀원	Amazon Cloud Server를 통한 Database/ Back-end 개발
4	나윤희	한국외국어대학교	정보통신공학과	3학년	팀원	Module 및 무인차량 제작
5						

2. 프로젝트 수행일정

[illegible]

3. 프로젝트 추진 과정에서의 문제점 및 해결방안

3-1. 프로젝트 관리 측면

- Gear360 에 대한 정보 부족

최대한 사용자에게 차량 제어에 대한 현실감을 주기 위해서 360도 카메라를 사용하여 영상을 송신하면 좋겠다는 아이디어가 나와서 Gear360의 존재를 알게 되었다. 하지만 아직 세간에 알려지지 않은 장비인 만큼 책이나 인터넷에서 해당 기술에 대한 정보를 확보하기는 불가능 했다(아예 존재하지 않는다). 이에 삼성 개발자와 연락 하여 두세 번의 연락 끝에 SDK와 이에 관련된 논문을 확보 했다.

- 기술에 대한 이해 부족

영상 스트리밍에 대한 기술은 세부적인 전공이 아닌 이상 학부 과정 내에서는 처음 보는 내용의 기술이 대부분 이었다. 이에 시작부터 바로 직접적인 개발에 들어가기 보다는 책, 인터넷을 통해서 해당 전문 분야에 대해 학습을 2~3개월 가량 먼저 진행을 한 후에 개발을 진행 했다.

- 작업 환경

데스크탑을 활용한 작업이 불가피 했던 상황이라 작업실을 구해서 작업을 해야 하는 상황 이었다. 하지만 올해는 관측 이래 최고의 폭염인 가운데 작업실에 에어컨이 구비되어있지 않아 실내 기온이 40도를 넘어가는 상황이 되어 팀원 모두의 사기와 체력이 저하되었다. 이에 폭염을 이겨내기 위한 물수건 얼리기, 선풍기에 캔 달기 등 실내 기온을 낮추기 위해 총력을 가한 결과 실내 기온을 3도가량 낮추었다.

3-2. 작품 개발 측면

- 클라이언트 어플 개발 초기에 안드로이드에 대한 관련 지식이 부족하여 플러그인을 통합하고 배포하는 과정에서 문제가 있었다. 안드로이드를 네이티브로 사용하고 유니티를 원본 어플로 사용하는 과정에서 설정을 명확하게 하지 않았기 때문이었는데 manifest 파일과 만들어준 플러그인들의 App이름, SDK 등등을 재 설정 해주어 문제를 해결 할 수 있었다.

- 차량의 모델이 될 RC카를 주문하는데 있어서 비용의 한계 때문에 절충적인 제품을 구매하였다. 완성도가 높은 제품은 아니어서 그런지 완성 후 사용하려 하자 충전이 안 되거나 부착해 놓은 모듈들이 연결 불량인 생겼다. 때문에

전선들을 같이 구입했던 새 전선들로 모두 교체하고, 수작업으로 보강작업을 하여 주행에 문제가 없게 만들 수 있었다.

4. 프로젝트를 통해 배우거나 느낀 점

○ 팀워크의 중요성

프로젝트를 진행하기 이전부터 HW 및 SW 파트에 대해 팀원들이 업무 분담을 철저하게 나누었다. 하지만 프로젝트 진행 중 팀원 개개인의 역량으로 해결되지 않는 문제들이 많이 발생하기 시작하였다. 팀원들 간의 소통이 필요하다고 생각하고 본인들이 하던 일을 모두 중단하고 같이 머리를 감싸며 문제를 해결해 나갔다. 문제를 해결하고 나니 자신감이 생겼고 서로에 대한 신뢰와 유대감이 더욱 강력해졌다. 팀워크란 물론 팀원들이 팀원의 위치에서 충실히 업무를 수행하는 것이지만 때로는 팀 자체 본연의 의미로 돌아가 함께 고민하는 것이라고 생각하게 되었다. 업무 효율성의 관점에서 팀워크의 중요성은 HW 및 SW 개발에 한정되어 있을 수 있지만 팀워크를 통해 얻은 교훈은 한정되어 있지 않다고 생각한다. 이번 프로젝트를 통해 실무에서 팀워크의 의미를 깨닫게 되는 계기가 되었다.

○ 자료검색의 중요성

본 프로젝트에서 우리가 개발한 작품은 기존의 널리 알려진 기술에 대해 프로젝트를 진행한 것이 아니었기 때문에 생소한 기술들이 많았다. 교과과정에서 배운 부분 이외의 것에 대해서는 자료 검색이 필요했다. 자료 검색을 통해 우리가 알았던 사실이 틀린 경우도 있었고 추가적으로 우리기술에 핵심이 되는 내용들이 무엇인지 명확하게 인지하게 되었다. 자료 검색을 통해 점점 기술에 대해 알 나가게 되니 기존에 알고 있던 지식이 확장되었고 확장된 지식을 통해 다음 번 자료 검색 시간을 단축시킬 수 있었다. 업무 개발 측면에서 효율적이었고 거듭할수록 프로젝트의 완성도를 높일 수 있었다.

○ 배경지식의 활용

학부과정 동안 전공수업을 들으면서 각 과목이 어떤 환경에서 어떻게 쓰이는지 항상 궁금증을 품어왔었다. 프로젝트를 진행하면서 우리가 들었던 전공 수업은 우리의 배경지식이 되어있었고 새로운 기술에 대한 자료 검색을 할 때 받아들이기 굉장히 수월했다. 이를 통해가지고 있던 정보를 활용하는 정보능력을 배양할 수 있었다.

IV. 작품의 기대효과 및 활용분야

※ 평가항목 : 기획력 (활용가능성)

1. 작품의 기대효과 쓰기

○ 차량 소유주 측면(고객)

- 대기시간 단축 : 대기 시간을 예측 가능하여 시간을 절약할 수 있다
- 요금의 표준화 : 정확한 수치를 통하여 표준화된 요금을 지불할 수 있다.
- 대리운전기사의 대기시간을 예측하여 시간을 절약할 수 있음

○ 대리운전기사 측면(주사용자)

- 편의성 : 직접 현장에 나가지 않고 원격으로 제어할 수 있다.
- 안전성 : 고객들과의 직접적인 마찰(신체적 접촉) 등을 줄여 안정적인 운전을 수행할 수 있다.
- 자택근무를 통하여 원격제어 하므로 업무시 불필요한 노동시간을 줄일 수 있다.

2. 작품의 활용분야

○ 첫 번째 활용 분야로는 우리 작품의 장점인 원격제어를 활용하는 것이다.

원격제어를 통해 얻을 수 있는 이점은 다음 두 가지가 있다. 첫째는 고객들과의 직접적인 마찰(신체적 접촉) 등을 줄여 안정적인 주행을 할 수 있게 해준다. 사람의 생명을 다루는 문제이기 때문에 안정성을 고려해야 한다. 둘째는 직접 근무센터에 오고 가는 불필요한 시간을 없애줌으로써 업무이외에 경제적인 손실을 막아준다. 대리운전기사의 업무뿐만 아니라 차량 렌트업체, 택시와 같이 차량을 이용하는 동종업계로 영역을 넓힐 수 있다.

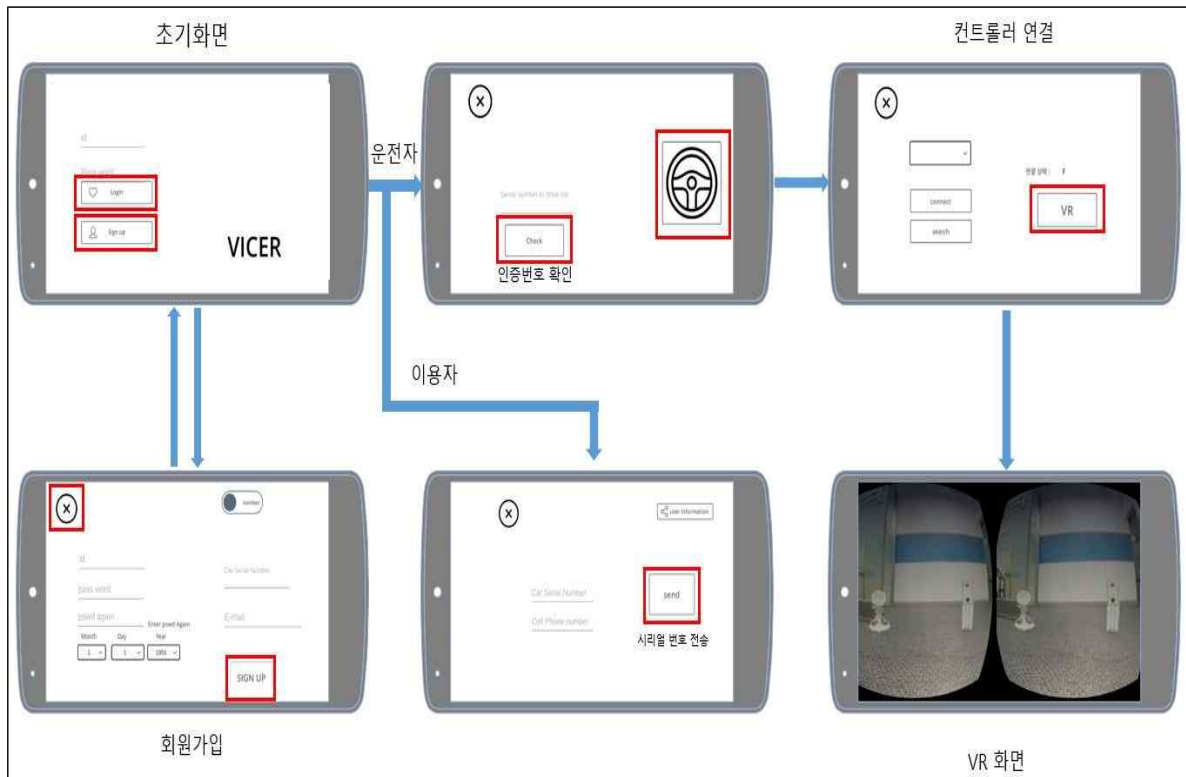
두 번째 활용 분야로는 군사 분야로 예를 들 수 있다. 현재 군대에서는 지뢰를 제거 하기위하여 병사나 간부가 직접 수색하여 제거를 하고 있다. 이러한 과정에서 예기치 못한 사고가 자주 일어난다. 여기서 사람이 직접 수색하는 것이 아닌 본 제품으로 지뢰의 위치를 파악할 수 있고 예기치 못하게 사고가 일어나도 사람이 다치는 것이 아닌 제품의 손상으로 끝나 인명피해를 없앨 수 있다.

세 번째 활용 분야로는 산업 분야로 예를 들 수 있다. 원전 사고 현장인 체르노빌이나 후쿠시마에선 방사능 수치 탐사를 위해 완전 무장을 하고 들어가도 방사능에 오염되는 경우가 많다. 이를 방지하기위해 본 제품에 방사능 수치 계산기를 달아서 해당 사고지역의 방사능 수치를 인명피해 없이 탐사할 수 있다.

V. 개발산출물

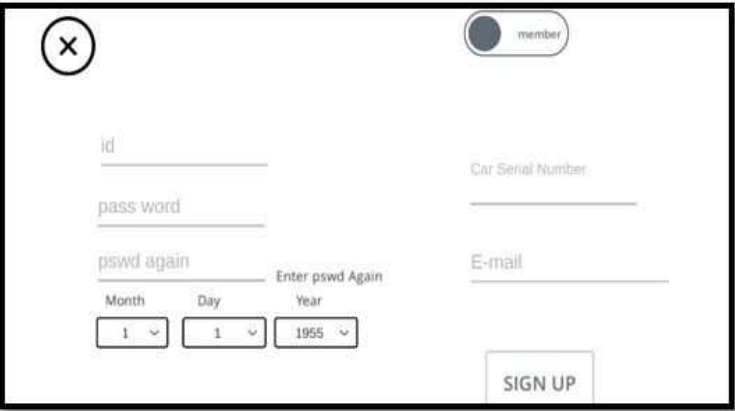
※ 평가항목 : 평가 전반에 참고

○ 메뉴 구성도




○ 화면 설계서

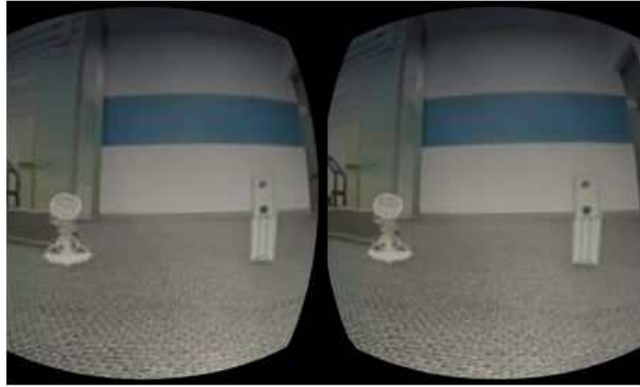
기능 명	로그인 창
기능 역할	아이디, 비밀번호 입력란 / 회원가입 창으로 넘어가기 로그인
기능설명	어플 실행 후 나오게 되는 첫 화면이다. 기존의 회원은 아이디와 비밀번호를 입력하여 각각에게 맞는 서비스 창으로 넘어갈 수 있다. 신규 회원은 회원가입 버튼을 눌러 가입하는 창으로 넘어갈 수 있다.
처리내용	로그인 아이디와 비밀번호를 입력하고 난 후 로그인을 누르게 되면 HTTP POST 방식으로 Database와 비교하여 각각의 이용자에게 맞는 화면으로 넘어가게 한다. 회원가입창 아직 계정이 없는 이용자가 회원가입을 하기위한 창으로 넘어가게 하는 버튼

	
기능 명	회원가입 창
기능 역할	이전 화면 / 회원가입 확인 / 직업 구분
기능설명	이용자가 어플을 사용하기 위해서 가입하는 창이다. 사용할 계정 정보와 자신의 직업(고객, 운전자) 에 따라 추가로 정보를 입력 할 수 있다.
처리내용	<p>■ 이전 화면 회원가입이 끝나게 되면 로그인 하기 위한 이전 페이지로 넘어갈 수 있다.</p> <p>■ 직업 선택 버튼 차량 소유주(고객)의 경우 운전 받을 차량의 시리얼 번호를 운전자에게 전송해야 한다. 따라서 회원가입 할 때 체크하여 추가로 전송 하도록 한다.</p> <p>■ 회원가입 버튼 회원가입 버튼을 누르면 입력한 정보들을 HTTP POST 방식을 통하여 서버로 전송한다.</p>

	
기능 명	이용자의 로그인 후 정보 전송 창
기능 역할	이전화면 / 운전자에게 차량의 정보를 전송
기능설명	이용자가 운전자에게 차량의 연결을 위해 입력한 번호로 시리얼 넘버를 전송한다.
처리내용	<p>■ 이전 화면 전송이 끝나게 되면 로그인 하기 위한 이전 페이지로 넘어갈 수 있다.</p> <p>■ 메시지 전송 입력한 번호로 HTTP POST 통신방식을 통해 시리얼 번호를 전송하게 된다. 성공적으로 전송이 되면 이용자는 운전자가 연결하기를 기다린다.</p>

	
기능 명	운전자의 차량 연결 창
기능 역할	이전화면 / 이용자의 차량에 연결하기위해 인증 번호를 입력한다.
기능설명	운전자는 이용자가 보낸 정보를 받아 입력한다. 인증에 성공하면 운전자는 자신의 컨트롤러와 차량을 연결할 수 있는 다음 창으로 넘어갈 수 있다.
처리내용	<p>■ 이전 화면</p> <p>전송이 끝나게 되면 로그인 하기 위한 이전 페이지로 넘어갈 수 있다.</p> <p>■ 인증 번호 확인</p> <p>인증번호 입력하고 확인 버튼을 누르면 서버로부터 현재 대기중인 차량의 목록 중에 이용자가 있는지 확인 후 컨트롤러에 연결 할 수 있게 다음 창으로 넘어갈 수 있게 한다.</p>

	
기능 명	운전자의 Application과 컨트롤러 연결
기능 역할	이전화면 / 현재 연결 가능한 컨트롤러 목록을 받아온다.
기능설명	현재 연결 가능한 컨트롤러의 목록을 확인 후 어플리케이션과 블루투스로 연결한다. 블루투스 연결이 정상적으로 된다면 VR 기어를 통해 영상을 볼 수 있는 창으로 넘어가게 된다.
처리내용	<p>■ 이전 화면</p> <p>연결이 끝나게 되면 로그인 하기 위한 이전 페이지로 넘어갈 수 있다.</p> <p>■ 블루투스 연결</p> <p>컨트롤러와 블루투스 통신으로 연결하기위해 연결 가능한 기기 목록을 불러온다. 연결에 성공하게 되면 VR 기기를 착용하는 화면으로 넘어갈 수 있다.</p>



기능 명	운전자의 VR 화면
기능 역할	운전하는 화면
기능설명	이용자의 차량과 연결하고 컨트롤러까지 연동되면 마지막으로 직접 차량을 제어하기 위해 보게 되는 화면이다.
처리내용	<p>■ VR</p> <p>360도 카메라로부터 오는 영상을 3차원 구에 덮어 씌워 고개를 어느 방향으로 돌리더라도 실제 그 공간에 있는 것 처럼 볼 수 있다.</p>

○ 블루투스 플러그인 코드

프로그램 ID	BluetoothModel.cs	프로그램 명	블루투스 연결을 위한 객체 모델	작성일	2018.09.03	Page	1/2
개요	블루투스 연결을 위해서는 블루투스 본연의 기능을 구현하고 그것을 실제로 받아줄 객체가 필요하다. 여기서는 블루투스 모델을 만들어 연결 하였다.					작성자	유한석
상세 로직				상세 로직			
<pre>private int bufferSize = 256; private char startChar = ' '; private char endChar = ' '; Socket sck; public List<string> macAddresses = null; private Queue<string> messageQueue = null; private StringBuilder rawMessage = null;</pre> <p>블루투스 연결을 할 때 필요한 변수들이다 블루투스 버퍼 사이즈는 256바이트이며 공백문자를 기준으로 문자를 구분하였다. 소켓 서버를 통해 블루투스 통신 받은 컨트롤러의 값을 서버로 전송해준다.</p>				<pre>void Awake() { this.observerList = new List<IBtObserver>(); this.macAddresses = new List<string>(); this.messageQueue = new Queue<string>(); this.rawMessage = new StringBuilder(this.bufferSize); } void Start(){ sck = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp); IPEndPoint localEndPoint = new IPEndPoint(IPAddress.Parse("52.79.181.10"), 5001); Debug.Log("connecting BTSC server..."); sck.Connect(localEndPoint); Debug.Log("connection BTSC"); string text = "R"; byte[] data = Encoding.UTF8.GetBytes(text); sck.Send(data); }</pre>			

프로그램 ID	AndroidManifest.xml	프로그램 명	블루투스 기능을 쓰기위한 플러그인	작성일	2018.09.03	Page	2/2
개요	블루투스 기능을 사용하기 위해 블루투스 플러그인을 만들어 유니티 상에서 사용하였다.					작성자	유한석
상세 로직							
<pre><?xml version="1.0" encoding="utf-8"?> <manifest package="com.hansuk12.testapp1" xmlns:android="http://schemas.android.com/apk/res/android"> <uses-permission android:name="android.permission.INTERNET"/> <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/> <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/> <uses-permission android:name="android.permission.WAKE_LOCK"/> <uses-permission android:name="android.permission.BLUETOOTH"/> <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/> <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/> <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/> <application android:icon="@drawable/app_icon" android:label="@string/app_name"> <activity android:name="com.hyeon.bluetoothPlugin.BluetoothPlugin" android:label="@string/app_name" android:configChanges="fontScale keyboard keyboardHidden locale mnc mcc navigation orientation screenLayout screenSize smallestScreenSize uiMode touchscreen"> <intent-filter> <action android:name="android.intent.action.MAIN" /> <category android:name="android.intent.category.LAUNCHER" /> </intent-filter> </activity> </application> </manifest></pre>							
				유니티상에서 안드로이드 스튜디오의 기능을 사용하려면 네이티브로 구현하여야 한다. 때문에 본 제품에서 필요한 블루투스를 manifest파일에 추가하여 플러그인으로 생성하여 유니티 프로젝트에서 사용하였다.			

○ RC Car 서버 코드

프로그램 ID	RC_Control_Server	프로그램 명	Rccar_server.py	작성일	2018. 09. 01	Page	1/3
개요	두종류의 Client 송신자(App : Controller), 수신자(Arduino : RC Car) 가 들어오면 각자 상황에 맞는 Thread를 실행 시켜서 TCP connection을 형성 시켜준다.					작성자	박지훈
상세 로직							
<pre>import sys from socket import * from threading import Thread data = (bytes) sendReady = False recvReady = True class myThread_sender(Thread): def __init__(self,connection): Thread.__init__(self) self.connection = connection self.resend = True def run(self): try: global data, sendReady, recvReady while True: if recvReady: data = self.connection.recv(1024) # recv next message on connected socket sendReady = True recvReady = False if not data: print("socket end") break # eof when the socket closed except OSError as e: # socket.error exception print('socket error:', e) except Exception as e: print('Exception:', e)</pre>				<pre> graph TD Rccar_server.py --> myThread_sender Rccar_server.py --> myThread_reciver Rccar_server.py --> Main_Tread[Main Tread] style myThread_sender stroke:#f00,stroke-width:2px </pre> <p>Global variable data : Byte[]</p>			

프로그램 ID	RC Control Server	프로그램 명	Rccar_server.py	작성일	2018. 09. 01	Page	2/3
개요	두종류의 Client 송신자(App : Controller) , 수신자(Arduino : RC Car) 가 들어오면 각자 상황에 맞는 Thread를 실행 시켜서 TCP connection을 형성 시켜준다.					작성자	박지훈

상세 로직

```

class myThread_reciver(Thread):
    def __init__(self,connection):
        Thread.__init__(self)
        self.connection = connection
        self.resend = True
    def run(self):
        try:
            global sendReady,recvReady,data
            while True:
                if sendReady:
                    self.connection.send(data) # send a reply to the client
                    recvReady = True
                    sendReady = False
                    print(data)
                    print("reciver go")

        except OSError as e: # socket.error exception
            print('socket error:', e)
        except Exception as e:
            print('Exception:', e)

def rc_server(my_port):
    """Echo server (iterative)"""
    try:
        sock = socket(AF_INET, SOCK_STREAM) # make listening socket
        # sock.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1) # Reuse port number if used
        sock.bind(('', my_port)) # bind it to server port number
        sock.listen(5) # listen, allow 5 pending connects
      
```

Rccar_server.py

Global variable
data : Byte[]

myThread_sender

myThread_reciver

Main Tread

프로그램 ID	RC_Control_Server	프로그램 명	Rccar_server.py	작성일	2018. 09. 01	Page	3/3
개요	두개의 Thread가 모두 형성이 되면 global variable(data)를 통해 수신된 값을 넣고 해당 값을 송신해준다. 이때 data의 길이가 너무 길어지게 되면 서버에서 송.수신할 때 원래 data에서 나뉜진 상태로 송.신을 하게 된다. 그러므로 들어오는 값들을 차례로 보다 빠른 처리를 위해서 Semaphore(sendReady, rcvReady)를 활용한 Spinlock 동기화(synchronization)로 구현 했다.					작성자	박지훈
상세 로직							

```
except OSError as e:
    print('socket error', e)
    sock.close()
    sys.exit(1)
else:
    print('Server started')
    while True: # do forever (until process killed)
        conn, cli_addr = sock.accept() # wait for next client connect
        # conn: new socket, addr: client addr
        print('Connected by', cli_addr)
        data_encoded = conn.recv(1024)
        print(data_encoded)
        try:
            data_decoded = data_encoded.decode("utf-8")
            print("decoded data"+ data_decoded)
            if "K" in data_decoded:
                print("in exception")
                raise ex
            else:
                print("start thread_sender")
                th = myThread_sender(conn)
                th.start()
        except Exception as ex: #exception 걸려서 오는 상황 (byte가 들어온 rcvver 상황)
            #data_encoded2 = conn.recv(1024)
            print("start thread_rcvver")
            th = myThread_rcvver(conn)
            th.start()
if __name__ == '__main__':
    rc_server(8011)
```

Rccar_Control.py

myThread_sender

myThread_rcvver

Main Tread

Global variable
data : Byte[]

○ 360° Streaming Server 코드

프로그램 ID	360Video_Streaming_server	프로그램 명	streaming_server.py	작성일	2018. 09. 01	Page	1/3
개요	두종류의 Client 송신자(Gear360 : Camera) , 수신자(App : VR) 가 들어오면 각자 상황에 맞는 Thread를 실행 시켜서 TCP connection을 형성 시켜준다.					작성자	박지훈
상세 로직							
<pre>import sys from socket import * from threading import Thread import binascii import PIL.Image import io data = (bytes) sendReady = False recvReady = True class myThread_sender(Thread): def __init__(self,connection): Thread.__init__(self) self.connection = connection self.resend = True def run(self): try: global data, sendReady, recvReady while True: if recvReady: #//print("data :",data) data = self.connection.recv(65536) # recv next message on connected socket sendReady = True recvReady = False if not data: print("socket end") break # eof when the socket closed except OSError as e: # socket.error exception print('socket error:', e) except Exception as e: print('Exception:', e)</pre>				<div><div>streaming_server.py</div><div><div>myThread_sender</div><div>myThread_rcvver</div><div>Main Tread</div></div><div>Global variable data : Byte[]</div></div>			

프로그램 ID	360Video_Streaming_server	프로그램 명	streaming_server.py	작성일	2018. 09. 01	Page	2/3
개요	두종류의 Client 송신자(Gear360 : Camera) , 수신자(App : VR) 가 들어오면 각자 상황에 맞는 Thread를 실행 시켜서 TCP connection을 형성 시켜준다.					작성자	박지훈
상세 로직							
<pre>class myThread_recver(Thread): def __init__(self,connection): Thread.__init__(self) self.connection = connection self.resend = True def run(self): try: global sendReady,rcvReady while True: if sendReady: self.connection.send(data) # send a reply to the client rcvReady = True sendReady = False #print("reciver go") except OSError as e: # socket.error exception print('socket error:', e) except Exception as e: print('Exception:', e)</pre>							

프로그램 ID	360Video_Streaming_server	프로그램 명	streaming_server.py	작성일	2018. 09. 01	Page	3/3
개요	두개의 Thread가 모두 형성이 되면 global variable(data) 를 통해 수신된 값을 넣고 해당 값을 송신해준다. 이때 data의 길이가 너무 길어지게 되면 서버에서 송.수신할 때 원래 data에서 나뉜 상태로 송.신을 하게 된다. 그러므로 들어오는 값들을 차례로 보다 빠른 처리를 위해서 Semaphore(<u>sendReady</u> , <u>rcvReady</u>)를 활용한 Spinlock 동기화(synchronization)로 구현 했다.					작성자	박지훈
상세 로직							

```
def streaming_server(my_port):
    """Echo server (iterative)"""
    try:
        sock = socket(AF_INET, SOCK_STREAM) # make listening socket
        # sock.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1) # Reuse port number if used
        sock.bind(('', my_port)) # bind it to server port number
        sock.listen(5) # listen, allow 5 pending connects
    except OSError as e:
        print('socket error', e)
        sock.close()
        sys.exit(1)
    else:
        print("Server started")
        while True: # do forever (until process killed)
            conn, cli_addr = sock.accept() # wait for next client connect
            print('Connected by', cli_addr)
            data_encoded = conn.recv(65536)
            print(data_encoded)
            try:
                data_decoded = data_encoded.decode("utf-8")
                print("decoded data"+ data_decoded)
                print("start thread reciver")
                th = myThread_recver(conn)
                th.start()
            except Exception as ex: #exception 걸려서 오는 상황 (byte가 들어온 sender 상황)
                print("start thread sender")
                th = myThread_sender(conn)
                th.start()
    if __name__ == '__main__':
        streaming_server(8010)
```


○ 로그인 Server 코드

프로그램 ID	회원 관리, 시리얼 넘버 처리, OTP 작업 등을 위한 SPRING back-end Server	프로그램 명	Vicer-Server (Maven Project)	작성일	2018. 09. 01	Page	1/2
개요	Spring Framework 는 MVC 패턴을 기반으로 하고있다. 아래의 코드는 Request가 들어오는 경우 어디로 Mapping하는지를 결정해주는 Controller 부분이다.					작성자	박지훈

상세 로직

```
import com.ice.vicer.service.*;
import com.ice.vicer.model.*;

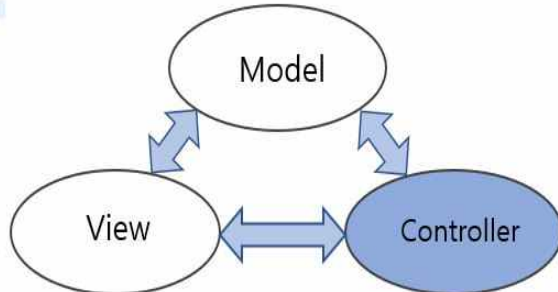
@Controller
public class VicerController {
    @Autowired
    private IMapper ibatisMapper;

    @RequestMapping(value="/member_reg.do", method=RequestMethod.GET)
    public int memberReg(HttpServletRequest arg0, MemberDTO dto) {
        int res = ibatisMapper.insertMember(dto);
        return res;
    }

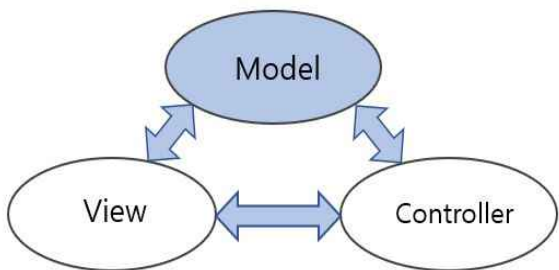
    @RequestMapping(value="/member_login.do", method=RequestMethod.GET)
    public int memberLogin(HttpServletRequest arg0, String id, String passwd) {
        int res = ibatisMapper.searchMember(id, passwd);
        return res;
    }

    @RequestMapping(value="/serial_search.do", method=RequestMethod.GET)
    public int serial_search(HttpServletRequest arg0, String serialNum) {
        int res = ibatisMapper.searchSerial(serialNum);
        return res;
    }

    @RequestMapping(value="/create_otp.do", method=RequestMethod.GET)
    public int createOTP(HttpServletRequest arg0, String serialNum) {
        int res = ibatisMapper.otp_create(serialNum);
        return res;
    }
}
```



```
graph TD
    Model((Model)) <--> View((View))
    Model <--> Controller((Controller))
    View <--> Controller
```

프로그램 ID	회원 관리, 시리얼 넘버 처리, OTP 작업 등을 위한 SPRING back-end Server	프로그램 명	Vicer-Server (Maven Project)	작성일	2018. 09. 01	Page	2/2
개요	아래의 코드는 DB상의 처리를 위한 Model 부분이다. (안드로이드 App을 통해서 사용자에게 제공되므로 View는 따로 존재하지 않는다.)					작성자	박지훈
상세 로직							
<pre>package com.ice.vicer.model; public class MemberDTO { private String id; private String passwd; private String email; private String serialNum; private String ocup; private String birth; public String getId() { return id; } public void setId(String id) { this.id = id; } public String getPasswd() { return passwd; } public void setPasswd(String passwd) { this.passwd = passwd; } public String getEmail() { return email; } public void setEmail(String email) { this.email = email; } public String getSerialNum() { return serialNum; } public void setSerialNum(String serialNum) { this.serialNum = serialNum; } }</pre>							

○ RC Car Controller 코드

프로그램 ID	RC Controller	프로그램 명	Controller.ino	작성일	2018. 09. 01	Page	1/3
개요	가변 저항으로 연결된 컨트롤러에서 실시간으로 변화하는 값(전진, 후진, 좌회전, 우회전, 기어변속)들을 블루투스로 연결된 App으로 전달.					작성자	류형오

상세 로직

```

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  BTSerial.begin(9600);
}

void loop() {

  if(a){
    BTSerial.print("S");
    a = false;
  }

  int gear = analogRead(A0);
  int steer = analogRead(A1);
  int accel = analogRead(A2);
  if (gear < 325) {
    gearaccel = true;
    gearreverse = false;
  }
  else if (gear > 370){
    gearaccel = false;
    gearreverse = true;
  }
  else{
    gearaccel = false;
    gearreverse = false;
  }
}

```

프로그램 ID	RC Controller	프로그램 명	Controller.ino	작성일	2018. 09. 01	Page	2/3
개요	가변 저항으로 연결된 컨트롤러에서 실시간으로 변화하는 값(전진, 후진, 좌회전, 우회전, 기어변속)들을 블루투스로 연결된 App으로 전달.					작성자	류형오

상세 로직

```

if (gearaccel)
{
  // case1: 전진
  if (steer - temp > 0)
  {
    if (steer - temp < 5 && steer - temp > -5) {
      if (accel < 486 && accel > 466)
      {
        BTSerial.print("S"); // 정지
      }
      else if (accel < 466)
      {
        BTSerial.print("F"); // 앞

      }
    }
  }
  else {
    if (accel < 486 && accel > 466)
    {
      BTSerial.print("R"); //핸들 위로
    }
    else
    {
      BTSerial.print("L"); //오른쪽앞
    }
  }
  delay(80);
}

```