2018. 11. 15

상세설계서

**VR과 360도 카메라를 이용한**

**실시간 스트리밍 원격제어 차량**

**(360°Avatar Driver)**

Ver. 1.2

2018.11.15

한국외국어대학교 정보통신공학과

1팀(VICER)

**문서 정보**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **구 분** | **소 속** | **성 명** | **날 짜** | **서 명** |
| **작성자** | 한국외국어대학교 | 박지훈 | 2018. 11. 15 |  |
| 한국외국어대학교 | 나윤호 | 2018. 11. 15 |  |
| 한국외국어대학교 | 류형오 | 2018. 11. 15 |  |
| 한국외국어대학교 | 유한석 | 2018. 11. 15 |  |
| 한국외국어대학교 | 유정현 | 2018. 11. 15 |  |
| **검토자** | 한국외국어대학교 | 박지훈 | 2018. 11. 15 |  |
| 한국외국어대학교 | 나윤호 | 2018. 11. 15 |  |
| 한국외국어대학교 | 류형오 | 2018. 11. 15 |  |
| 한국외국어대학교 | 유한석 | 2018. 11. 15 |  |
| 한국외국어대학교 | 유정현 | 2018. 11. 15 |  |
| **사용자** |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **승인자** | 한국외국어대학교 | 홍진표 | 2018. 11. 15 |  |

**개정 이력**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **버전** | **작성자** | **개정일자** | **개정 내역** | | **승인자** | |
| 1.0 | 박지훈 | 2018.11.09 | 초안 작성 | |  | |
| 나윤호 |
| 류형오 |
| 유한석 |
| 유정현 |
| 검토자 | 박지훈 | | | | |
| 1.1 | 박지훈 | 2018.11.14 | 초안 수정 | |  | |
| 나윤호 |
| 류형오 |
| 유한석 |
| 유정현 |
| 검토자 | 박지훈 | | | | |
| 1.2 | 박지훈 | 2018.11.15 | | 최종 수정 | |  |
| 나윤호 |
| 류형오 |
| 유한석 |
| 유정현 |
| 검토자 | 박지훈 | | | | |

목 차

목차4

표 목차6

그림 목차**7**

1. 개요8

1.1 목적8

1.2 범위8

1.3 관련 문서9

1.4 용어 및 약어9

1. 시스템 구성10
   1. 시스템 구성도10
   2. 소프트웨어10
   3. 하드웨어 11
2. 기능 설명 11
   1. 14
   2. 세부 시스템 구성도14
      1. 통신 시스템14
      2. VR Watch Application Use Case Diagram15
      3. Controller Use Case Diagram15
      4. Using VR Application sequence diagram16
      5. Control RC Car Sequence diagram16
   3. 소프트웨어17
      1. Unity Engine17
      2. Oracle17
      3. Android Studio17
      4. Arduino(Sketch) 18
   4. 하드웨어18
      1. Arduino18
      2. HC-06 블루투스 모듈18
      3. 360도 카메라19
      4. RC Car19
      5. Controller20
         1. Controller Handle20
         2. Controller Gear20
         3. Controller Pedal20
      6. 기어 VR21
3. 기능동작22

4.1 제품 알고리즘22

4.2 Sequence Diagram24

4.2.1 Video Streaming Sequence Diagram 24

4.2.2 RC Car Control Sequence Diagram 24

1. 제품 자체 시험 방법 및 절차26
2. 향후 실제 적용 방안26
3. 기대 효과26
4. 프로젝트 세부추진계획 및 세부일정27
5. 소스 코드26

표 목 차

[Table 1] 관련 문서 9

[Table 2] 약어 및 용어 9

[Table 3] 가변 저항 센서 구성 13

[Table 4] 가변 저항 센서 모듈 스펙 13

[Table 5] 블루투스 모듈 구성15

[Table 6] 블루투스 모듈의 스펙15

그 림 목 차

[Figure 1] 시스템 구성도 10

[Figure 2] 시스템 구성요소(컨트롤러)12

[Figure 3] 가변 저항 센서 모듈 12

[Figure 4] 핸들(컨트롤러)14

[Figure 5] 액셀(컨트롤러) 14

[Figure 6] 기어(컨트롤러) 14

[Figure 7] 블루투스 모듈(HC-06) 15

[Figure 8] 시스템 구성요소(RC car) 16

[Figure 9] 모터쉴드 17

[Figure 10] 센서쉴드 17

[Figure 11] Data Flow 19

[Figure 12] 영상 스트리밍 구성도 20

[Figure 13] User Interface 22

[Figure 14] Bluetooth Plugin Model1 24

[Figure 15] Bluetooth Plugin Model2 24

[Figure 16] Bluetooth Plugin Model3 25

[Figure 17] 스티칭 된 frame 26

[Figure 18] wrapping frame26

[Figure 19] 사용자가 보는 화면27

[Figure 20] Gear 360 앱 구성도 27

[Figure 21] Data Flow 29

[Figure 22] 제품 알고리즘 30

[Figure 23] Video Streaming Sequence Diagram 30

[Figure 24] RC Car Control Sequence Diagram 31

1. 개요

본 장에서는 VR과 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량시스템에 상세설계서를 제공한다. 즉, 현장감 있는 원격의 차량 운전 시스템인 VICER의 목적과 범위, 정의 사항, 참고 자료, 그리고 본 문서의 개요를 소개한다.

1.1 목적

본 프로젝트의 목적은 VR과 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량 시스템 구축을 하는데 그 목적이 있다.

본 프로젝트는 기존의 무선조종 RC카가 제공하는 2D영상의 한정된 시야를 극복하고 36 0도 카메라를 이용하여 4D영상을 실시간으로 송신 받아 운전자에게 사방의 상황을 제공 한다.

또한, 현재 시장에 출시되어 있는 원격제어 제품(드론, 무선조종 RC카)들의 P2P 통신방 식의 한계인 거리 제약을 극복하기 위해 소켓서버를 통한 실시간 제어를 구현함으로써 거리 의 제약 없이 지구반대편에서도 이용 가능하다.

프로젝트를 진행하기 위해 아래의 사항을 구체적으로 명시하고 구현할 것이다.

(1) Unity Engine을 활용하여 app device를 원격 운전자에게 제공

(2) Samsung gear 360 SDK를 활용하여 영상 스트리밍을 구현하고 원격 운전자는 gear VR를 장착하여 이를 보게되므로, 현장감 넘치는 실시간 3D 운전 영상을 제공

(3) 원격 운전자는 컨트롤러(핸들, 페달, 기어)를 사용하여 실내에서 원격으로 차량을 제어

(4) 차량 소유주와 원격 운전자에게 App 제공

(5) DB에 있는 차량 소유주의 일련번호를 OTP 인증방법을 이용하여 원격 운전자에게 제공함으로써 보안 강화

1.2 범위

본 상세 설계서에는 차량에 360도 카메라를 부착하고, 차량과 원격으로 연결된 컨트롤러를 통해 사용자가 운전을 하면, 카메라를 통해 촬영되는 운전하는 차량의 상황을 VR 기기를 통해 실시간으로 스트리밍 할 수 있는 시스템의 구현 기술에 대해 기술하고 있다. 사용자는 어플리케이션을 통해 자신의 정보와 차량 정보를 입력 후 회원 가입을 하고, 로그인을 한다. 어플리케이션은 사용자가 입력하는 차량의 시리얼 넘버를 통해 해당 차량과 컨트롤러를 연결시키고, 차량에 부착된 360도 카메라가 실시간으로 촬영하는 차량 주변의 모습을 스트리밍 해준다. 스트리밍 되는 영상은 VR기기를 통해 사용자가 확인하면서 컨트롤러를 제어하는데, 이것은 직접 차 안에서 운전을 하는 듯한 현장감을 제공하게 된다. 본 프로젝트 개발 진행에 있어 다음과 같은 범위를 둔다.

1. **어플리케이션을 이용하는 사용자 판단**: 사용자가 어플리케이션 이용을 위해 입력하는 정보들, 이를테면 이름, 전화번호, 차량 번호, 시리얼 번호 등은 요청 시, 어플리케이션과 연동된 WAS에서 DB에 접속하여 그것들을 저장하거나, 다른 정보들과 비교하거나, 이에 응답해준다.
2. **차량과 컨트롤러의 연결**: 사용자가 입력한 시리얼 번호를 토대로 하여, 어플리케이션은 블루투스를 통해 차량과 컨트롤러를 서로 연결시킨다.
3. **촬영되는 영상의 전송**: 차량에 부착된 360도 카메라에서 촬영된 정보가 영상 스트리밍을 위한 서버로 전송되면, 서버는 이 정보를 바로 안드로이드 어플리케이션 상으로 전달한다. 어플리케이션은 이 영상을 3D로 전환시킨다.
4. **영상의 현장감 전달**: 사용자가 어플리케이션 상에서 VR 버튼을 누르면, 어플리케이션은 전환시킨 3D 영상을 실시간으로 스트리밍 해준다. 사용자는 VR 기기를 장착하고, 스마트폰을 통해 스트리밍 되는 영상을 보며 운전한다.
5. **원격 차량 제어:** 사용자가 컨트롤러를 사용하면, 컨트롤러가 감지하는 정보들은 차량 제어를 위한 Control 서버로 전송된다. 이 서버는 게이트웨이 역할을 하면서, 받은 정보들을 차량으로 넘겨주고, 이 정보들은 차량의 Steer과 Wheel Motor를 동작시킴으로써 사용자는 차량을 제어하게 된다.

1.3 관련 문서

|  |  |
| --- | --- |
| **출판사** | **문서 제목** |
| 위키북스 | 절대강좌! 유니티 5 |
| 한빛아카데미 | 안드로이드 프로그래밍 |
| 한빛미디어 | 초보자를 위한 유니티 5 입문 : 설치에서 3D와 2D 게임까지 |

[Table 1] 관련 문서

1.4 용어 및 약어

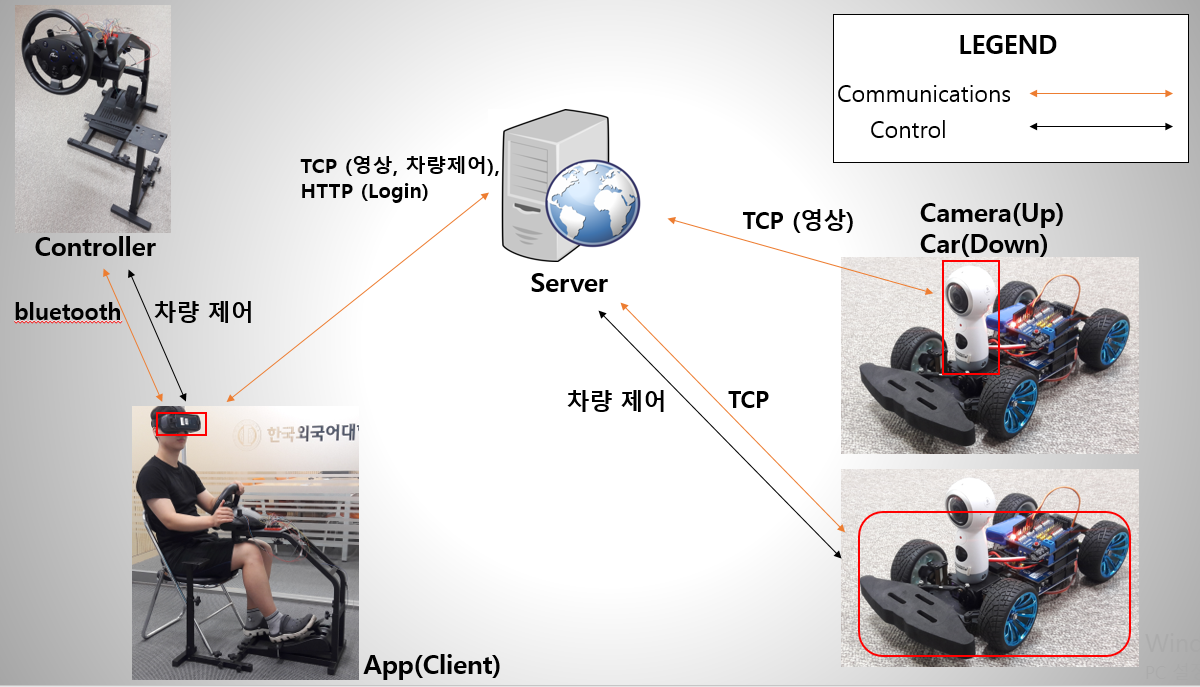
|  |  |  |
| --- | --- | --- |
| **용어 및 약어** | **풀이** | **비고** |
| WAS | Web Application Server, 웹 애플리케이션과 서버환경을 만들어 동작시키는 기능을 제공하는 소프트웨어 프레임워크, 웹 애플리 케이션 서버는 동적 서버 콘텐츠를 수행하는 것으로 일반적인 웹 서버와 구별됨 |  |
| MTU | ㅇ 어떤 데이터링크에서 하나의 프레임 또는 패킷에 담아 운반 가능한 최대 크기  - 수용 가능한 상위 계층 데이터(헤더 포함된 전체 싸이즈)의 크기 |  |
| 스티칭기법 | 영상을 이어 붙이는 기법, 앞,뒤,좌우,위,아래의 모든 모습이 나오는 360도 영상을 위한 방법 |  |
| VICER | VR(Vritural Reality)과 ICE(Information and Communication Engineering)의 합성어, 원격 차량 제어 시스템의 이름이자 팀명 |  |

[Table 2] 약어 및 용어

1. 시스템구성

2.1 시스템 구성도

VR과 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량(360°Avatar Driver)의 시스템 구성은 아래 그림과 같다.



[Figure 1] 시스템 구성도

2.2 소프트웨어

VR과 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량(360°Avatar Driver)는 영상 스트리밍 서비스와 차량제어 서비스를 제공하기 위해 아래와 같은 소프트웨어를 사용한다.

• 서버

* 영상 스트리밍과 RC Car 조작을 위한 중계 서버로 Amazon Web Service(AWS)

를 사용한다.

* 서버와 데이터베이스 간의 통신을 하기 위한 언어로 Spring을 이용하고 이러한 코드를 이해하기 위한 엔진으로 Tomcat을 사용한다.

• 클라이언트

* 360**°**카메라의 영상을 서버로 전달해주기 위해 안드로이드 스튜디오를 사용하여 어플리케이션을 설계한다.
* 서버로부터 영상을 받아 보는 것과 블루투스로 연결되어 있는 컨트롤러에서부터 전달받은 데이터를 서버로 전달해주기 위해 Unity를 이용하여 어플리케이션을 설계한다.

• 데이터베이스

* DBMS로 Oracle을 이용한다.
* MyBatis를 이용하여 서버와 연결한다.

• UI

* Unity를 이용하여 어플리케이션의 UI를 설계한다.

• 서버 메시지 전달

* 유니티의 C#을 이용하여 컨트롤러에서 생성하는 값을 서버로 전달하여 RC Car를 조작한다.
* 안드로이드 스튜디오의 Java를 이용하여 360**°**카메라의 영상 프레임을 서버로 전달하여 영상 스트리밍을 진행한다.

• 차량 제어 메시지 생성

* Arduino를 이용하여 Unity로 만들어진 어플리케이션과 블루투스 통신으로 컨트롤러에서 생성되는 메시지를 전송한다.

2.3 하드웨어

VR과 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량(360°Avatar Driver)는 360°영상을 얻기 위해 Gear360을 이용한다. Gear360에서 찍은 영상을 서버PC로 전달하고 서버PC는 영상을 제공받는 스마트폰으로 전달되어 Gear VR을 이용하여 360°영상을 제공받는다.

또한 이동하는 차량을 Arduino Uno를 이용하여 제작한다. 이 차량을 제어하기 위한 컨트롤러는 Arduino의 가변저항 센서를 사용하여 제작한다.

1. **기능 설명**

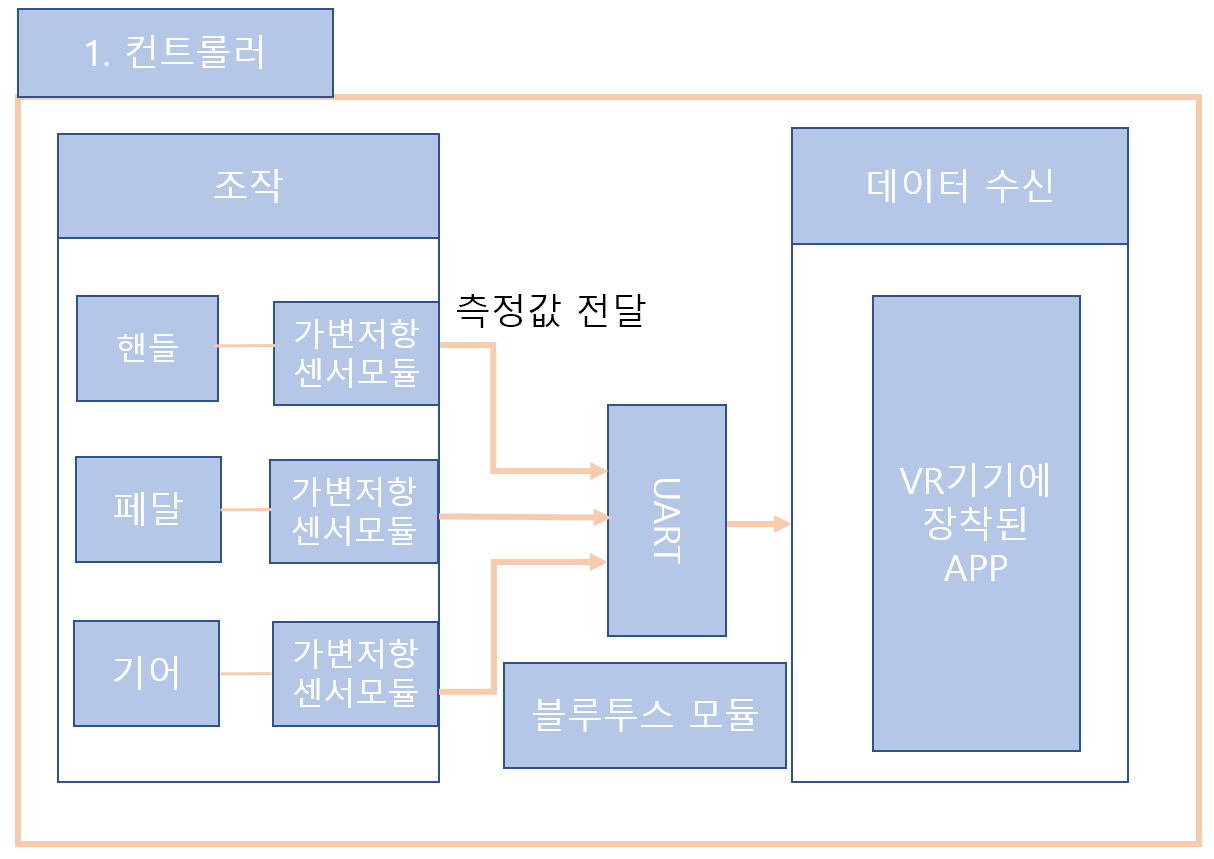
본 절에서는 요구사항 정의서에 기술한 360도 아바타 드라이버의 기능적 요구사항을 구현하기 시스템을 상세 설계한다. 전체 시스템 구성을 H/W에서부터 S/W까지 Data의 흐름에 따라 나누어 기술하였다.

3.1 Controller

컨트롤러(핸들,페달,기어)에서 조작한 값을 VR App으로 전달하기 위해

컨트롤러에 가변저항 센서모듈을 부착하고 측정된 가변저항 센서의 값을

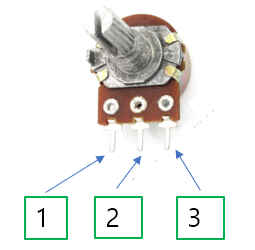
블루투스 통신을 이용해 App에 전달한다.



[Figure 2] 시스템구성요소(컨트롤러)

3.1.1 컨트롤러 조작

가변저항센서모듈을 통해 컨트롤러에서 조작된 데이터 값은(0~1023)은 문자의 형태로 변환되어 송수신간의 9,600 Baud rate로 VR기기에 장착된 앱으로 전송된다.



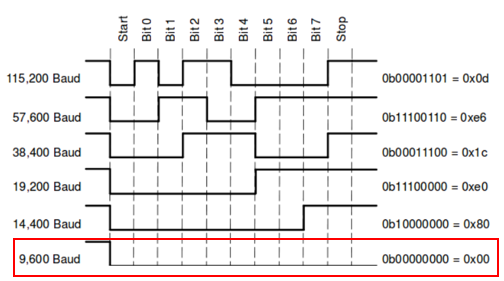
[Figure 3] 가변 저항 센서 모듈

|  |  |  |
| --- | --- | --- |
| NO | 명칭 | 비고 |
| 1 | GND | 아두이노 GND에 연결 |
| 2 | Signal Ouptput | 0~1023의 측정값을 아날로그값으로 출력 |
| 3 | Vcc | 아두이노 5V에 연결 |

[Table 3] 가변 저항 센서 구성

|  |  |
| --- | --- |
| Sensor | Potentionmeter(가변저항센서) |
| Range | 0~1023 사이의 Analog 값 |
| Operation Volt | 5V |
| Output | Analog Output |

[Table 4] 가변 저항 센서 모듈의 스펙



3.1.1.1 핸들 (좌, 우)

핸들의 부착된 가변저항센서 모듈은 핸들의 좌,우 각도를 계산하여 범위값에

대한 문자값을 설정하여

UART 포트를 통해 블루투스 통신으로 App으로 전달된다.



[Figure 4] 핸들(컨트롤러)

3.1.1.2 페달 (F, B)

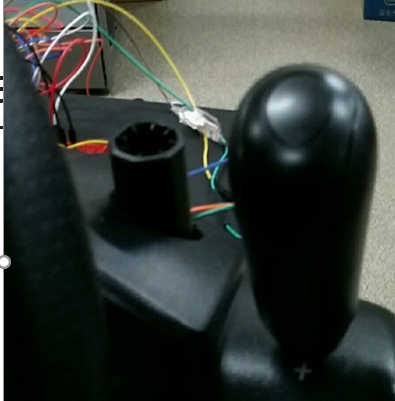
페달의 부착된 가변저항센서 모듈은 밟는 정도에 따라 계산하여 범위값에 대한 문자값을 전환되 UART 포트 통해 블루투스 통신으로 App으로 전달된다.



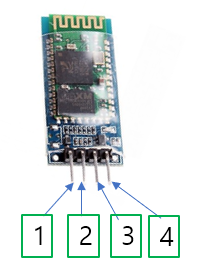
[Figure 5] 액셀(컨트롤러)

3.1.1.3 기어 (전진, 후진, 중립)

기어의 부착된 가변저항센서 모듈은 (0~1023)값에 따라 전진,후진,중립 범위를 설정하여 문자값으로 전환해 UART 포트 통해 블루투스 통신으로 App으로 전달된다.



[Figure 6] 기어(컨트롤러)



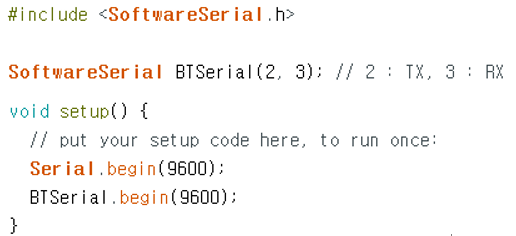
[Figure 7] 블루투스 모듈(HC-06)

|  |  |  |
| --- | --- | --- |
| No | 명칭 | 비고 |
| 1 | Rxd | 아두이노 digital 2번핀에 연결 |
| 2 | Txd | 아두이노 digital 3번 핀에 연결 |
| 3 | Vnc | 아두이노 5V에 연결 |
| 4 | Gnd | 아두이노 GND에 연결 |

[Table 5] 블루투스 모듈 구성

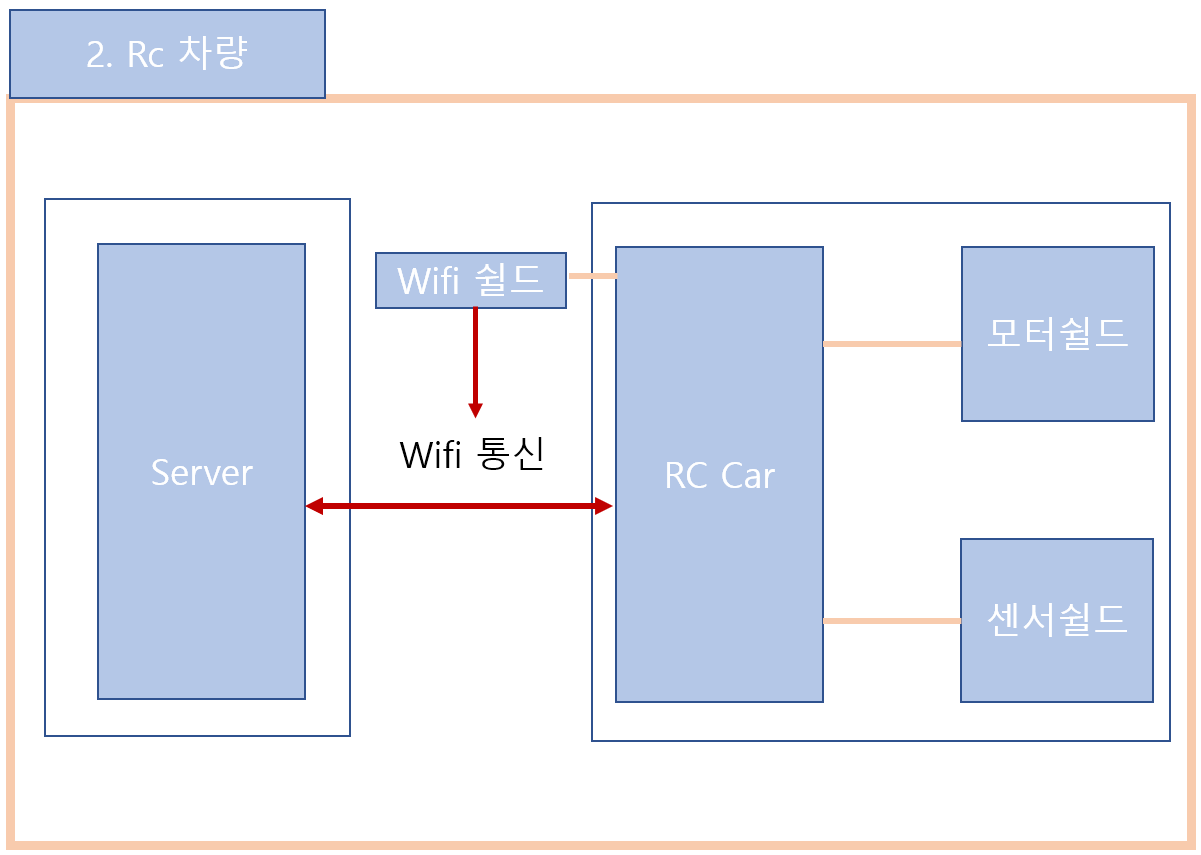
|  |  |
| --- | --- |
| 모듈명 | UART 블루투스 모듈 |
| 인터페이스 | VCC,GND,TXD,RXD,STATE,EN(KEY) |
| 소비전류 | 페어링 대기 상태 30mA, 페어링 완료 상태 10mA/20uA 이하 |
| 입력전원 | 3,3v~6v DC |
| 통신거리 | 개방공간 기준 10m |
| 초기설정값 | Bluetooth Slave/9600bps/8/1/None/pwd 1234 |
| LED상태표시 | 점멸(페어링대기), 점등(페어링 완료) |
| 크기 | 3.5cm \* 1.52cm |

[Table 6] 블루투스 모듈의 스펙



아두이노 우노(R3) 기존에 Built-in 된 시리얼 통신용 0,1번 핀 외에 다른 디지털핀으로 Serial 통신을 원활하게 해주기 위해 SofrtwareSerial 라이브러리 사용.

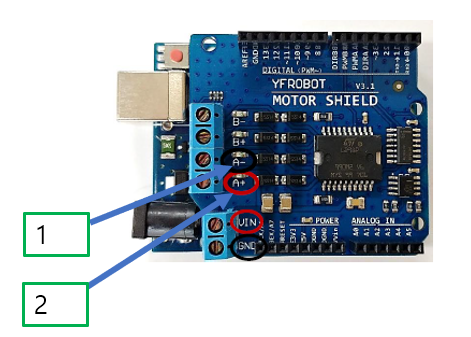
3.2 RC 차량



[Figure 8] 시스템구성요소(RC Car)

3.2.1 모터쉴드

모터쉴드 모듈은 DC모터의 +,- 단자를 채널 A의 +,-에 연결시켜 모터의 같은 물리적인 입출력을 제어한다.+

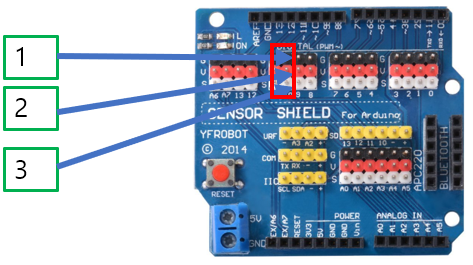


[Figure 9] 모터쉴드

3.2.2 센서쉴드

센서쉴드 모듈은 모터쉴드위에 결합하여 서브모터에 연결된 선을 아날로그 입출력

에 연결한다.



[Figure 10] 센서쉴드

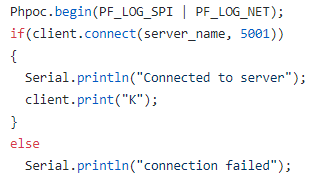
서브모터에 연결된 선을 1,2,3 에 순서대로 각각 9번핀에 s,v,g로 연결한다.

3.2.4 wifi쉴드(Phpoc)

와이파이 쉴드 Phpoc 모듈은 차량과 서버간의 와이파이 통신을 위해 구축한 인프라이다.











차량을 손쉽게 제어하기 모터의 정력과 모터의 속도를 0~255중에

110으로 설정했다.

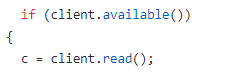
phpoc 와이파이 쉴드를 사용하기 위해 phpoc 라이브러리와 Spi 라이브러리 이용한다.

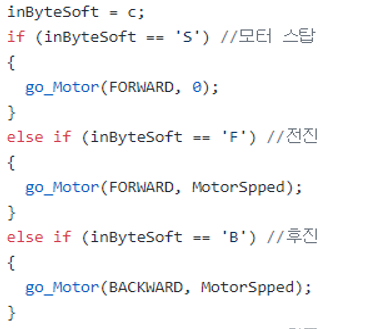
서버와 연결하기 위해 필요한 초기설정

3.2.4.1 Phpoc 와이파이쉴드를 (컨트롤러 조작한 값 서버로 송신)

Phpoc(와이파이 쉴드)로 인해 차량과 서버가 연결되었을 때

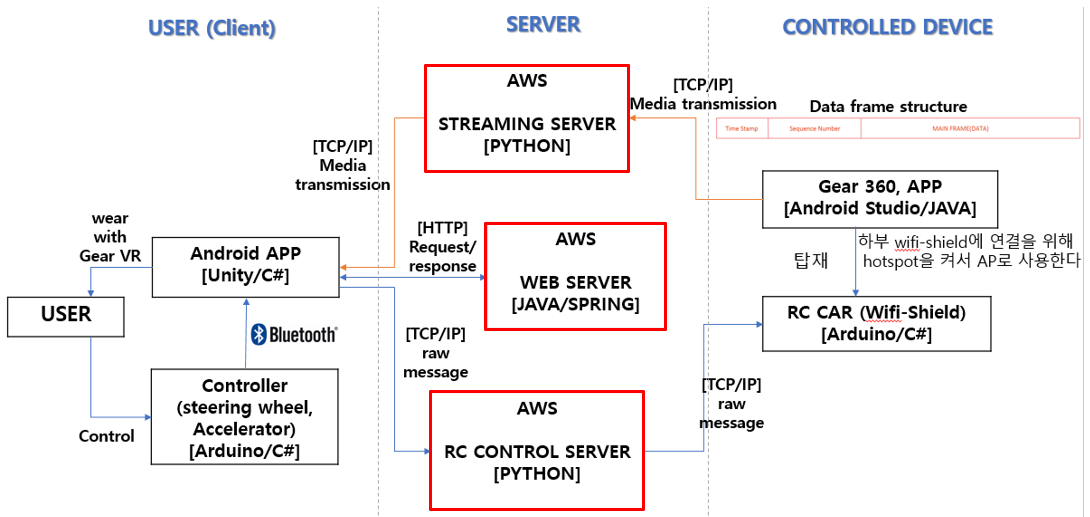
Phpoc로 차량과 서버가 연결되었을 때 컨트롤러로 조작한 값이 문자로 변환되어 아래와 같이 c변수에





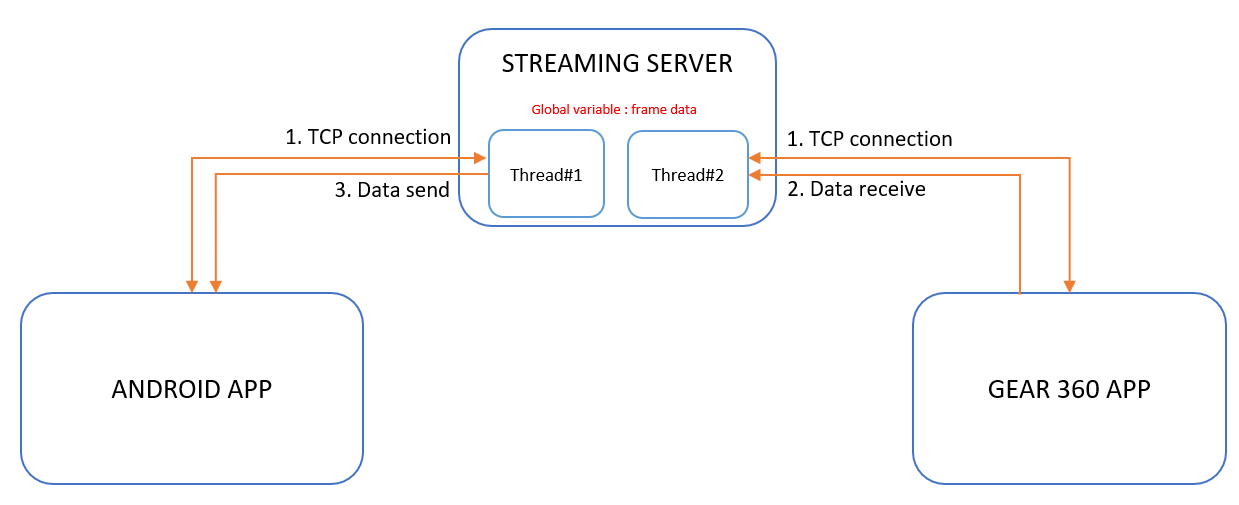
서버에서 위와 같이 Character 값을 전달하여 차량이 제어된다.

3.3 Server

 본 작품에는 촬영한 frame을 사용자에게 전달하기 위한 streaming server , 사용자가 제어한 제어정보를 차량까지 전달하기 위한 rc control server , 사용자 식별 및 차량 식별을 위한 web server 로 구성 되어 있다.

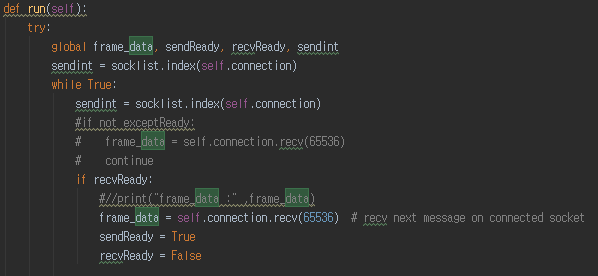
[Figure 11] Data Flow

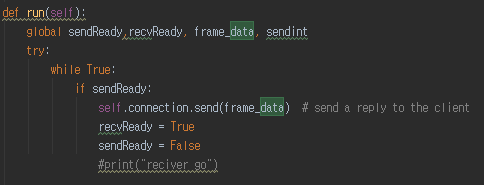
3.3.1 Streaming Server

 본 서버는 TCP Protocol을 기반으로 하는 relaying(중계) 서버로 gear 360 app 으로부터 전달받은 frame data를 어떠한 처리도 하지도 않고 사용자 앱(Android App) 으로 포워딩 해주는 역할을 담당 하고 있다.

[Figure 12] 영상 스트리밍 구성도

1. Android App 과 Gear360 App 에서 각각 들어오는 TCP connection 요청을 식별하여 각 Thread를 생성시켜준다.
2. Gear 360으로부터 data를 수신하는 Thread#2는 수신 받은 data를 전역 변수인 frame data에 넣어준다.
3. Frame data 의 정보를 Thread#1을 통해서 Android App으로 전송해준다.
4. 2,3 이행 시 Multi-Thread에서 공용변수를 사용하는 상황으로 mutual exclusion을 구현해 주기위해 lock 변수를 2개 설정해 주어 spinlock(busy waiting)기법으로 구현 하였다.

Receiving Thread

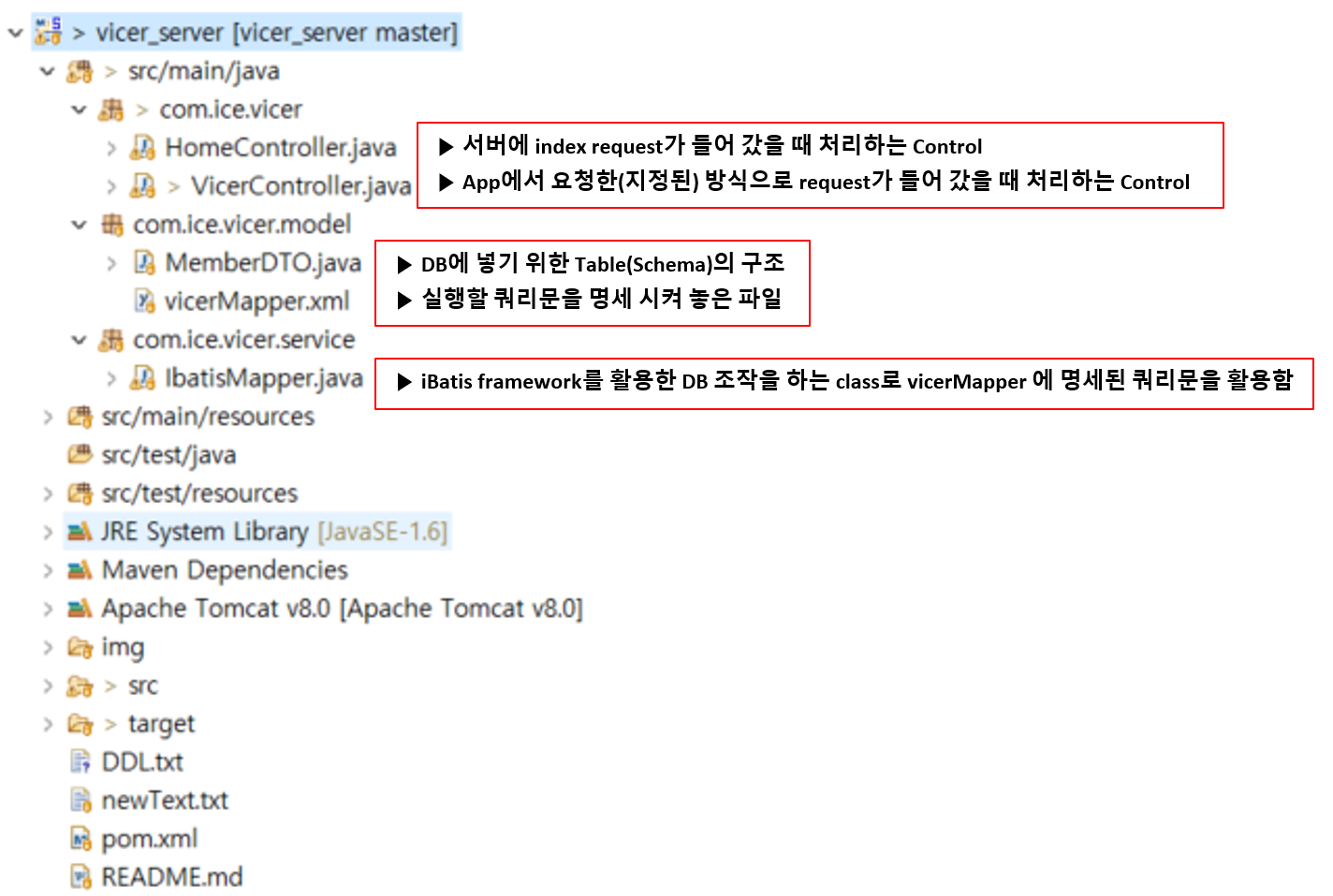
Sending Thread

3.3.2 RC Control Server

본 서버는 TCP Protocol을 기반으로 하는 relaying(중계) 서버로 사용자가 제어하는 Controller 으로 부터 전달받은 조작 정보를 어떠한 처리도 하지도 않고 RC Car으로 포워딩 해주는 역할을 담당 하고 있다.

기술적인 구현은 3.3.1의 Streaming Server와 동일 하므로 생략한다.

3.3.3 Web Server

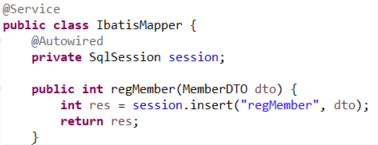
본 서버는 사용자 식별 및 제어하는 차량의 식별을 위한 back-end 웹 서버로 SPRING framework를 사용하여 제작되었다. (서비스 제공은 User Application을 통해서 제공되어 web 상의 view는 존재하지 않는다.)

1. VicerController.java



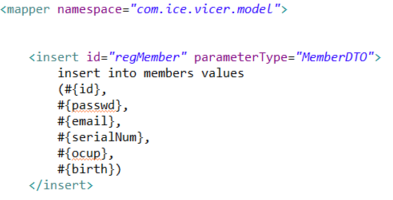
Path : /member\_req.do 로 request가 들어오게 되면 작동되는 부분으로 회원가입 등록이 정상적으로 이루어지면 res = 1 이 되어 response로 1을 보내주고 실패하게 되면 예외처리가 되어 res = 999가되어 response로 999를 보내주게 된다.

1. IbatisMapper.java



Controller에서 호출되는 부분으로 DB조작을 위한 쿼리문 실행을 위해 vicerMapper에 있는 쿼리문에 접근한다.

1. vicerMapper.xml



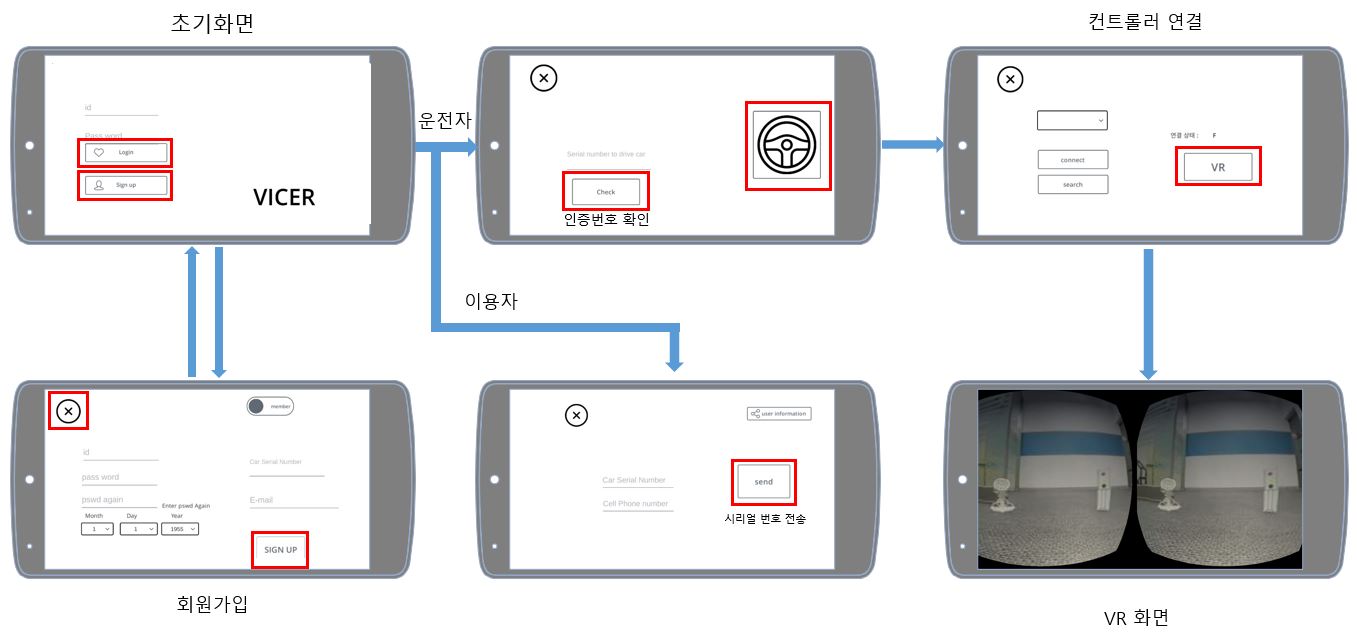
회원 가입을 할 때 불려오는 부분으로 Mapper의 일부이다.

3.4 Application

3.4.1 User Application

운전자는 본 어플리케이션을 통하여 Controller로부터 받은 제어정보를 Control Server로 전송해주며 Streaming Server로부터 받은 영상 프레임을 GearVR을 통해 사용자가 볼 수 있도록 한다.

* 운전자 사용 흐름



[Figure 13] User Interface

1. 최초 사용자는 자신이 사용할 어플리케이션을 실행시킨다.
2. Web Server에서 조종할 RC Car에 대한 인증을 한다.
3. 사용할 Controller 와 Bluetooth Pairing한다.
4. VR 화면으로 넘어가게 되면 Gear VR에 장착하여 영상을 본다

.

* 통신 구현.

1. HTTP 통신



.

사용자가 로그인하기 위해 웹 서버를 통해 데이터베이스의 값과 비교하는 코드이다. 입력한 정보가 데이터 베이스 상에 존재한다면 1을 수신 받고, 그 외의 오류를 받게 되면 RC Car Serial 정보를 확인하는 화면으로 넘어가지 않게 된다. 회원 가입시에도 동일한 통신 방식으로 데이터베이스에 사용자 정보를 저장하게 된다.

1. Bluetooth

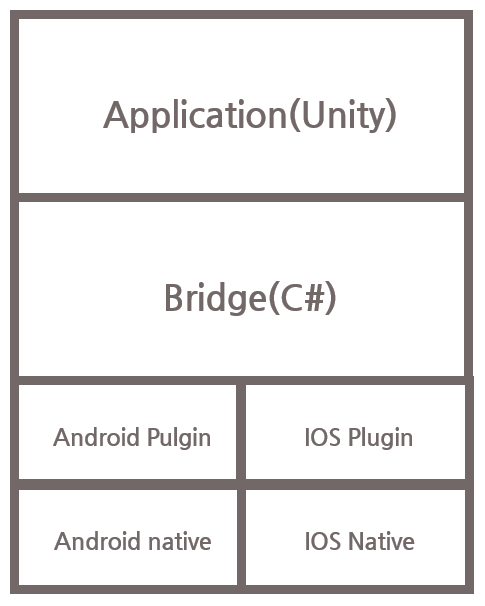
-사용자의 블루투스 사용 흐름



[Figure 14] Bluetooth Plugin Model1

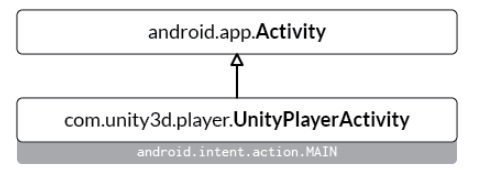
1. Search 버튼을 이용하여 현재 Pair가능한 기기의 목록을 찾고 MacAddress를 수신 받는다.
2. connect버튼을 이용하여 연결 요청을 시도하고 Pair가능하다면 VR 화면으로 넘어가는 버튼이 활성화 되며 컨트롤러로부터 받는 제어정보가 TCP 통신을 통해 전송되기 시작한다.

- 작동원리



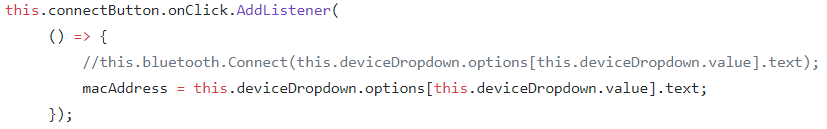
[Figure 15] Bluetooth Plugin Model2

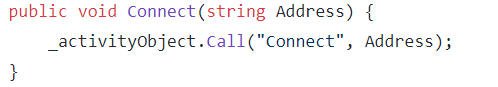
유니티 상에서 블루투스를 비롯한, 유니티에서는 제공하지 않는 안드로이드 기능을 사용하기 위해서는 Java Library를 .jar 파일로 만들어 플러그인이라는 하나의 형태로 사용하여야 한다. 본 Application에서는 블루투스 및 화면전환을 위한 기능을 추가하기 위해 플러그인으로 만들어 구현하였다.

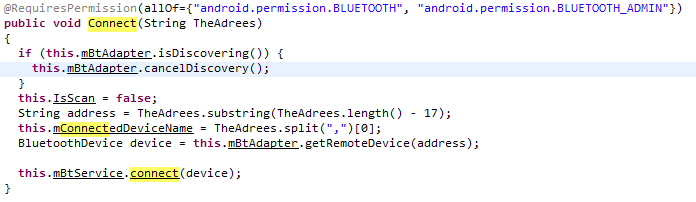


[Figure 16] Bluetooth Plugin Model3

결론적으로 유니티를 안드로이드로 빌드 시키게 되면 유니티 프로젝트가 하나의 Activity가 되어 라이브러리의 Method들을 사용하는 것처럼 작동한다.







1. Watching VR

서버로부터 수신 받는 영상 Frame은 180도 광각 렌즈로부터 촬영된 두개의 화면이 스티칭 기법을 통해 하나의 프레임으로 나오게 된다. 이를 유니티 상에서 하나의 구를 만들어 스티칭된 화면을 구의 안쪽에 랩핑 해준다. 이렇게 되면 사용자는 구의 중심에서 사방을 바라볼 때 전방향의 이미지를 볼 수 있게 된다.



[Figure 17] 스티칭 된 Frame



[Figure 18] wrapping frame

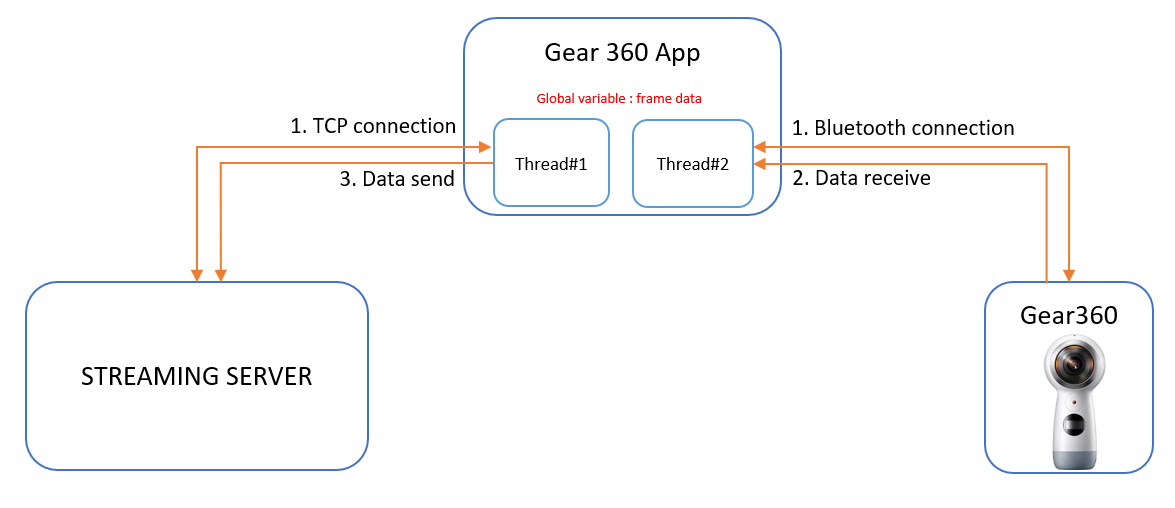
사용자는 Figure19 에서 보이듯이 구체에 씌워진 이미지 한 가운데에서 사방을 둘러 볼 수 있다.



[Figure 19] 사용자가 보는 화면

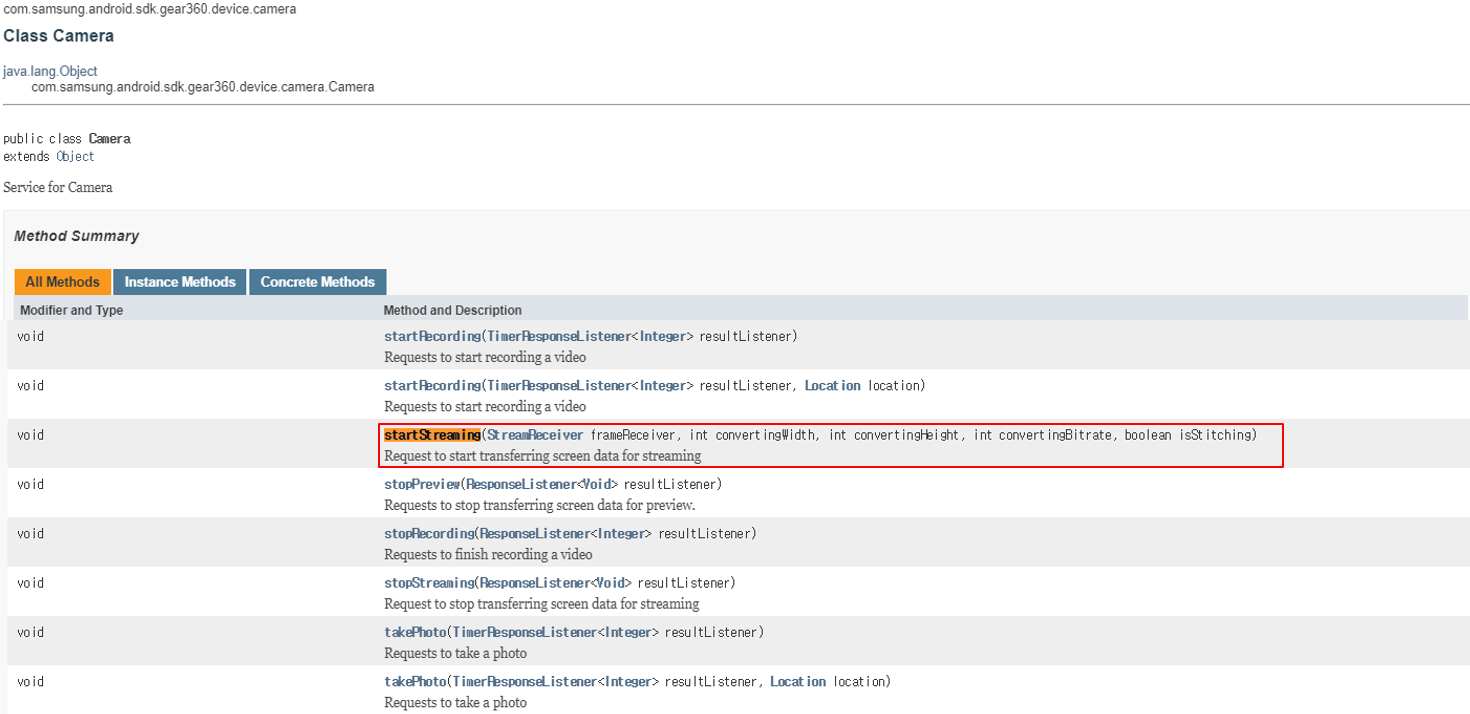
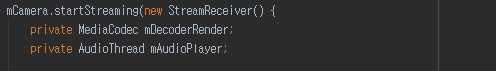
이렇게 구의 중심에서 사용자가 전방을 보는 화면을 나타낼 수 있다

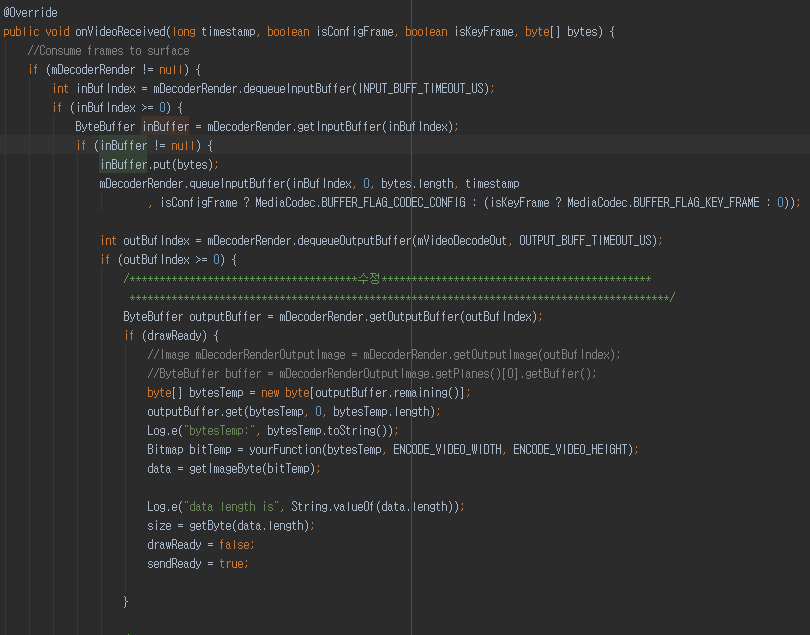
3.4.2 Gear360 App

 본 어플리케이션은 gear360 에서 촬영한 정보를 수신하여 해당 정보를 압축, 가공하여 Streaming Server 로 전송해주는 역할을 한다.

[Figure 20] Gear 360 앱 구성도

Gear360과 통신,제어 하기위해 삼성에서 제공하는 Gear360 SDK를 활용 했다.

1. 촬영한 매 순간의 값을 가져오기 위해서 한 Camera의 startStreaming 함수를 사용하기 위해 <Interface> StreamReceiver를 정의해준다.
2. 정의된 함수중 onVideoReceived 부분이 video frame이 수신 될 때 마다 실행되는 call-back 함수 이므로 해당 부분을 수정해준다.
3. onVideoReceived 함수에서 받아오는 frame data(YUV color space)를 Bitmap 형식으로 변환 시켜주고 이를 JPEG 압축을 통해 전역변수 data에 넣어준다.



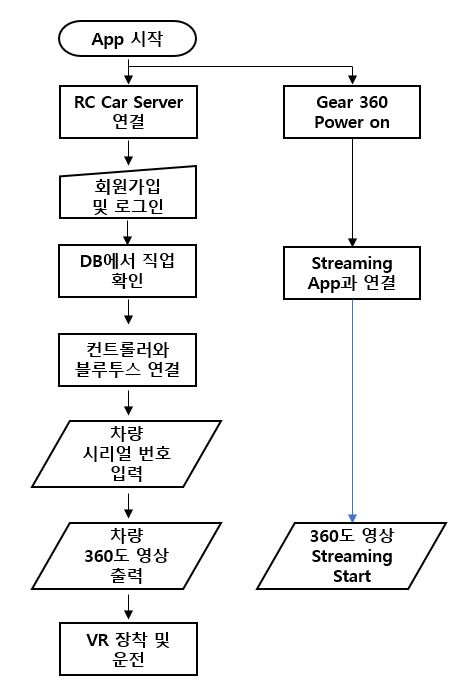
1. 전역변수 data 에 대해 Multi-Thread 에서 사용중 이므로 mutual exclusion을 구현하기 위해 3.3.1에서와 마찬가지로 spinlock기법을 이용하여 lock변수 2개를 활용하여 구현하였다.
2. 기능 동작

4.1 Data Flow



[Figure 21] Data Flow

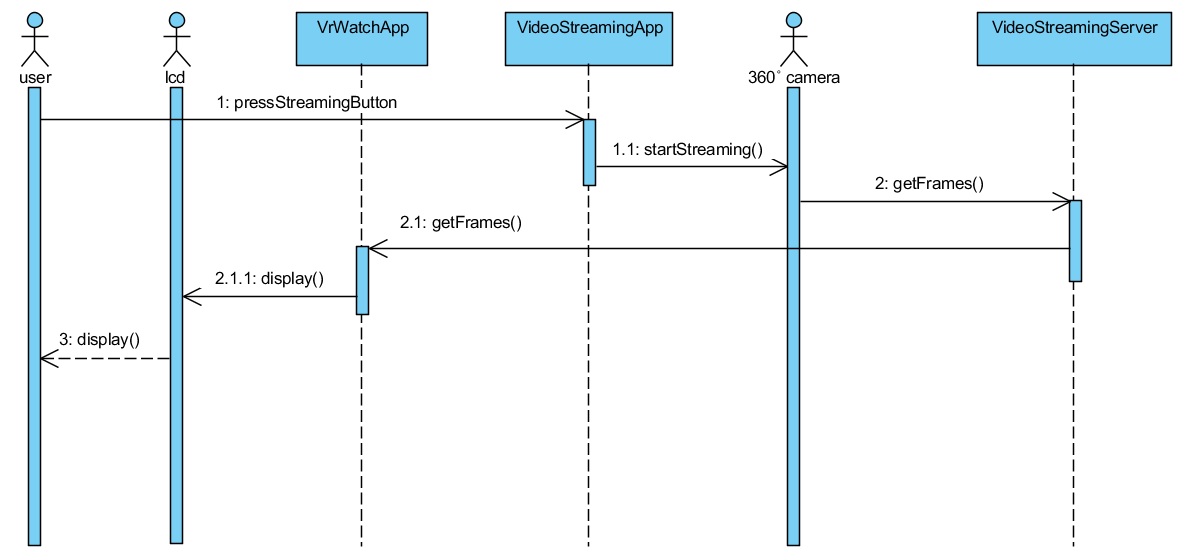
4.2 제품 알고리즘



[Figure 22] 제품 알고리즘

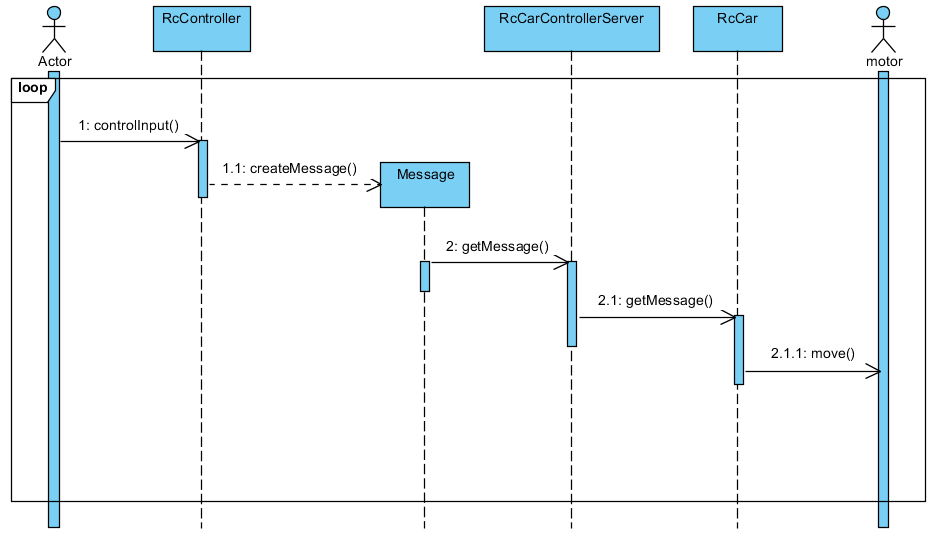
4.3 Sequence Diagram

• Video Streaming Sequence Diagram



[Figure 23] Video Streaming Sequence Diagram

• RC Car Control Sequence Diagram



[Figure 24] RC Car Control Sequence Diagram

1. 자체 시험 방법 및 절차

VICER 자체 시험은 다음과 같이 수행할 것이다. 각 단계별 시험 방법 및 절차는 아래와 같다.

1. 어플리케이션과 웹 서버 간의 시험

* 사용자가 가입을 위해 어플리케이션을 통해 입력하는 정보를 웹 서버에서 접속한 데이터베이스에 저장할 수 있는지 확인
* 데이터베이스에 이미 저장되어 있는 정보를 사용자가 입력하여 로그인 했을 때, 서버가 데이터를 조회하여 상이한 데이터가 있으면, 로그인을 거절하고, 조회한 정보랑 같은 데이터가 있으면 로그인을 처리하는 지 확인
* 로그인 상태가 계속 유지될 수 있는 지 확인
* 사용자가 로그아웃을 눌렀을 때, 로그아웃이 데이터베이스 상에서 정상적으로 처리 되는지 확인

1. 어플리케이션과 컨트롤러 간의 시험

* 블루투스 통신을 통해 어플리케이션과 컨트롤러를 서로 연결할 수 있는지 확인
* 블루투스 통신을 통해 이루어진 연결이 컨트롤러 조작 시 끊김이 생기지 않는 지 확인
* 사용자가 어플리케이션 로그아웃 시, 컨트롤러와의 연결도 제대로 끊기도록 처리되는 지 확인

1. 어플리케이션과 차량 부착 360도 카메라 간의 시험

* 어플리케이션 화면을 통해 360도 카메라가 촬영하는 영상이 실시간으로 스트리밍 되는 지 확인
* 어플리케이션에서 스트리밍되는 영상에 끊김과 지연이 발생하지 않는 지 확인

1. 어플리케이션과 VR기기 간의 시험

* VR기기를 사용하여, 어플리케이션 상의 화면을 실감나게 볼 수 있는 지 확인
* VR기기를 장착하고 보게 되는 화면이 360도 전방을 볼 수 있는지 확인

1. 컨트롤러와 RC카 간의 시험

* 컨트롤러와 RC카가 서로 잘 연결이 되어 있는지 확인
* RC카는 컨트롤러가 조작되는 대로 움직이는 지 확인

1. 향후 실제 적용 방안

6.1.

요즘 ‘키덜트(kidult: Kid와 Adult, 아이와 어른의 합성어)족’이 새로운 트렌드로 떠오르고 있다. 성인이 되었는 데도 여전히 어렸을 적의 분위기와 감성을 간직한 사람들을 일컫는다. 이들은 어린 시절에 경험했던 향수들을 여전히 잊지 못하고 그 경험들을 다시 소비하고자, RC카나 아날로그 게임 등과 같은 잡화 등을 이용한다. 키덜트족 관련 시장 규모는 2014년 5000억원대에서 해마다 20%씩 성장해 지난해 1조원을 넘어섰을 것으로 추정되고 있다. 올해는 시장 규모가 더욱 커질 것이라는 예상이다. 이와 함께, 입지를 늘려가고 있는 기술이 바로 ‘실시간 스트리밍 기술’이다. 이 기술에 대한 연구는 계속 되고 있으며, 미래에 가상현실, 증강현실 콘텐츠에서 쓰일 핵심 기술이 될 것으로 기대되고 있다.

해당 시스템은 이러한 트렌드적 가치관과 미래 유망 기술들을 기반으로 하고 있다. VR기기를 장착한 후, 자동차에 부착된 360도 카메라가 촬영하는 영상이 실시간으로 스트리밍 되는 것을 받아볼 수 있는 시스템 구조를 가지고 있는데, 이를 통해 사용자는 실제로 운전하는 듯한 현장감과 생동감을 느낄 수 있게 된다.

키덜트족은 경제적으로 여유가 있어 고가의 상품에도 돈을 아끼지 않는 성향을 가지고 있다. 또한, 소비가 본인에게 집중되는 1인가구들은 자신의 취미 생활이나 관심있는 물건에 대해 과감하게 투자하는 경향을 강하게 보인다. 이러한 사회적인 분위기는 해당 시스템이 보유한 기술들이 사용자로 하여금 소비되는 것에 어려움이 없는 환경을 만들 것이다.

6.2.

우리는 현재, 사람이 운전하지 않고도 알고리즘만으로 주행하는 ‘완전 무인 자동차’의 시대를 기다리고 있다. 국내∙외 자동차 회사들은 계속해서 자율 주행 자동차를 연구하고 있는 중이며, 이미 몇몇 회사들은 출시를 위한 테스트까지 마친 시점이다. 그들은 차를 움직이는 알고리즘이 정상적으로 작동하는지 확인하기 위해서 무인 자동차에 사람을 탑승 시켰다. 아무 일이 없었다면 다행이지만, 이로 인해 실험자의 사고가 빈번히 발생하고 있으며, 사망사고까지 일어난 상태다. ‘사람이 없는 자동차’가 과연 정상적으로 주행할 것인지 어떻게 확인할 수 있을까?

해당 시스템은 SAMSUNG Gear 360 카메라를 이용하였는데, 이는 자동차와 연결된 컨트롤러의 조작과 VR기기의 장착만으로 실제 자동차에 탑승한 듯 운전자의 시야를 실시간으로 확보해준다. 이것은 효율적인 무인 자동차의 테스트를 위한 몇 가지 이점을 가지고 있다. 기계의 작동을 확인하기 위해 사람을 이용하지 않는다는 점, 그러면서 실시간 영상 스트리밍 기술로 자동차가 주행하는 경로를 확인할 수 있다는 점, 이로 인해 잘못된 알고리즘으로 인한 사고 발생 시 재빠른 조치가 가능하다는 점, 출시 전의 테스트 뿐만 아니라 시스템 고장 확인 및 업데이트 버전의 차량 출시 시에도 단순한 연결만으로 재사용이 가능하다는 점, 운전자의 입장에서 운행을 확인함으로써 더욱 발전된 형태의 자율 주행 자동차 설계가 가능하다는 점 등이다.

사람이 없는 자동차의 출현으로 인하여, 해당 차량 시스템이 도태될 것이라고 예측한다는 것은 옳지 않다. 이것은 둘 중 누가 더 어떤 장점을 가지고 있는 지, 견주어 비교할 문제가 아니라, 자율 주행 차량을 위해서 꼭 필요한 시스템이라고 할 수 있다.

1. 기대 효과

해당 시스템을 이용하면, 운전을 하기 위해서, 자동차에 직접 탑승하지 않아도 된다. 이것이 가져오는 가장 큰 효과는 인명피해를 최소화 시킬 수 있다는 점이다. 자동차로 인한 사망 및 사고는 자신이 조심한다고 해서 일어나지 않는 것이 아니다. 다른 운전자들의 부주의를 통해서도 빈번히 일어난다. 우리는 원격으로 차량을 조정함으로써 사고로 인한 인명피해를 방지할 수 있게 된다.

또한 사용자는 실제로 차량이 있는 위치에 의존하지 않아도 된다. 사용자가 동작시키는 것은 컨트롤러 하나이다. 목적지가 어디든지 상관없다. 먼 거리이던 짧은 거리이던 간에, 사용자는 현 위치에서 원격으로 그 곳에 도착할 수 있다. 이것은 사용자에게 시간적으로 엄청난 경제적 효과를 가져다 줄 것이다.

이 시스템은 컨트롤러와 실제 자동차를 연결시킴으로써 운전이 가능하다. 연결은 시리얼 번호를 인증하는 방법으로 이루어지는데 이것은 꼭 한 대의 자동차에만 국한된다고 볼 수 없다. 적으면 한 대에서 많으면 수십대까지, 하나의 컨트롤러를 가지고 여러 대의 자동차를 운전하고 관리할 수 있다는 점에서 확장성과 용이성을 기대할 수 있다.

1. 세부 추진 계획 및 진행 상황

|  |
| --- |
|  |