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. 10. 31 |  |
| 한국외국어대학교 | 나윤호 | 2018. 10. 31 |  |
| 한국외국어대학교 | 류형오 | 2018. 10. 31 |  |
| 한국외국어대학교 | 유한석 | 2018. 10. 31 |  |
| 한국외국어대학교 | 유정현 | 2018. 10. 31 |  |
| **검토자** | 한국외국어대학교 | 박지훈 | 2018. 10. 31 |  |
| 한국외국어대학교 | 나윤호 | 2018. 10. 31 |  |
| 한국외국어대학교 | 류형오 | 2018. 10. 31 |  |
| 한국외국어대학교 | 유한석 | 2018. 10. 31 |  |
| 한국외국어대학교 | 유정현 | 2018. 10. 31 |  |
| **사용자** |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **승인자** | 한국외국어대학교 | 홍진표 | 2018. 10. 31 |  |

**개정 이력**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **버전** | **작성자** | **개정일자** | **개정 내역** | | **승인자** | |
| 1.0 | 박지훈 | 2018.10.29 | 초안 작성 | |  | |
| 나윤호 |
| 류형오 |
| 유한석 |
| 유정현 |
| 검토자 | 박지훈 | | | | |
| 1.1 | 박지훈 | 2018.10.30 | 초안 수정 | |  | |
| 나윤호 |
| 류형오 |
| 유한석 |
| 유정현 |
| 검토자 | 박지훈 | | | | |
| 1.2 | 박지훈 | 2018.10.31 | | 최종 수정 | |  |
| 나윤호 |
| 류형오 |
| 유한석 |
| 유정현 |
| 검토자 | 박지훈 | | | | |

목 차

목차5

표 목차6

그림 목차**7**

1. 개요9

1.1 목적9

1.2 범위9

1.3 관련 문서10

1.4 용어 및 약어10

1. 연구 목표 및 내용11
   1. 연구 목표11
   2. 관련 기술12
      1. Frame Compression12
      2. Media Transmission13
2. 전체 시스템 구성도 및 필요 장비14
   1. 전체 시스템 구성도14
   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. 요구사항 및 제한사항22

4.1 인터페이스 요구사항22

4.1.1 시스템 인터페이스 기능22

4.1.2 하드웨어 인터페이스 요구사항22

4.1.3 소프트웨어 인터페이스 요구사항 23

4.1.4 네트워크 인터페이스 요구사항 23

4.2 사용자 요구사항 24

4.2.1 사용자 기능적 요구사항 24

4.2.2 사용자 비기능적 요구사항 24

4.3 시스템 구현의 제한사항 25

1. 자체 실험 방법 및 절차26
2. 향후 실제 적용 방안27
3. 기대 효과27
4. 프로젝트 세부 추진 계획 및 일정27
5. 부록27
6. 소스 코드 첨부27

표 목 차

[Table 1] 관련 문서 10

[Table 2] 약어 및 용어 10

[Table 3] 사용자 인터페이스 요구사항 22

[Table 4] 하드웨어 인터페이스 요구사항 23

[Table 5] 소프트웨어 인터페이스 요구사항23

[Table 6] 네트워크 인터페이스 요구사항23

[Table 7] 사용자 기능적 요구사항 24

[Table 8] 사용자 비기능적 요구사항 25

그 림 목 차

[Figure 1] 시스템 구성도 11

[Figure 2] 통신 흐름도 11

[Figure 3] Frame Compression 12

[Figure 4] TCP/IP Packet Structure 12

[Figure 5] UDP Packet Framing 13

[Figure 6] System Diagram 3.1 13

[Figure 7] System Diagram 3.2.1 14

[Figure 8] VR Watch Application Use Case Diagram 14

[Figure 9] Controller Use Case Diagram 15

[Figure 10] Using VR Application sequence diagram 15

[Figure 11] Control RC Car Sequence diagram 16

[Figure 12] Unity Engine 16

[Figure 13] Oracle 17

[Figure 14] Android Studio 17

[Figure 15] Arduino(Sketch) 18

[Figure 16] Arduino 18

[Figure 17] HC-06 Bluetooth Module 18

[Figure 18] Samsung 360 Camera 19

[Figure 19] RC Car19

[Figure 20] Controller Handle 20

[Figure 21] Controller Gear20

[Figure 22] Controller Pedal20

[Figure 23] Gear VR 21

1. 개요

본 장에서는 VR과 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량 시스템에 대한 요구사 항의 총괄 개요를 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 요구사항 명세서에는 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

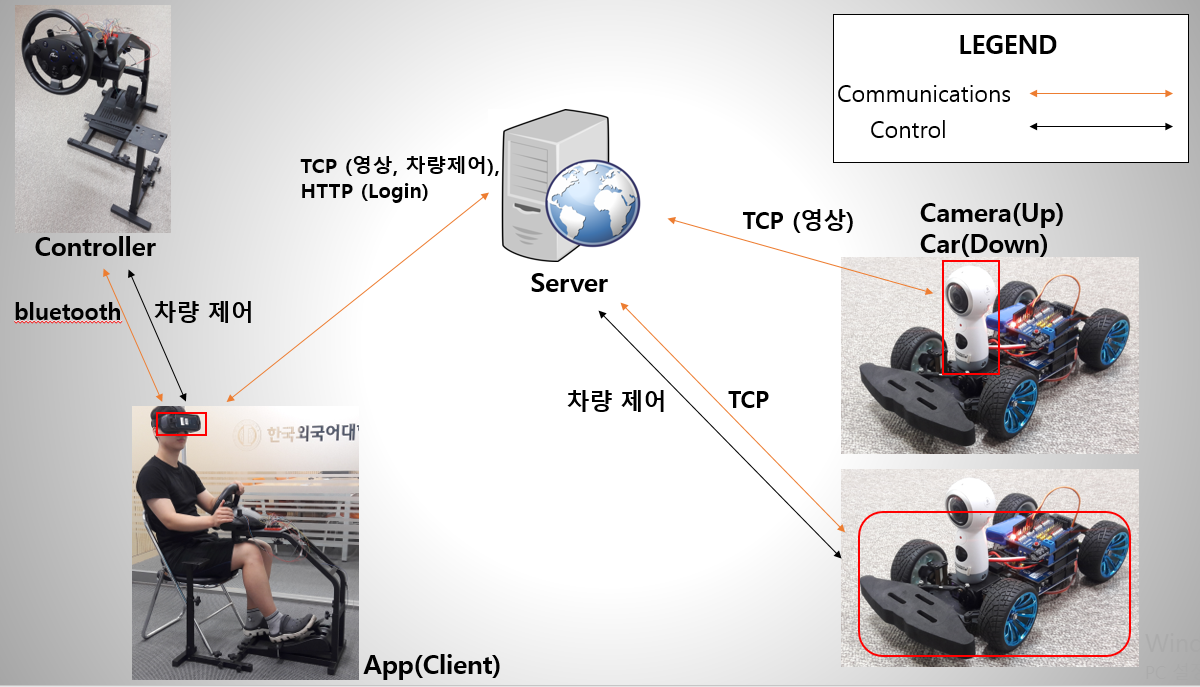
|  |  |  |
| --- | --- | --- |
| **용어 및 약어** | **풀이** | **비고** |
| WAS | Web Application Server, 웹 애플리케이션과 서버환경을 만들어 동작시키는 기능을 제공하는 소프트웨어 프레임워크, 웹 애플리 케이션 서버는 동적 서버 콘텐츠를 수행하는 것으로 일반적인 웹 서버와 구별됨 |  |
| MTU | ㅇ 어떤 데이터링크에서 하나의 프레임 또는 패킷에 담아 운반 가능한 최대 크기  - 수용 가능한 상위 계층 데이터(헤더 포함된 전체 싸이즈)의 크기 |  |
| 스티칭기법 | 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

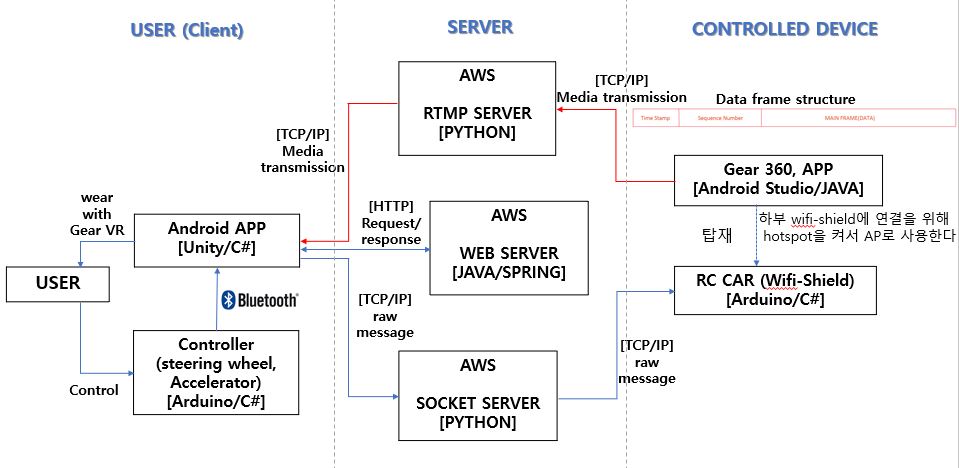
1. 시스템구성

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



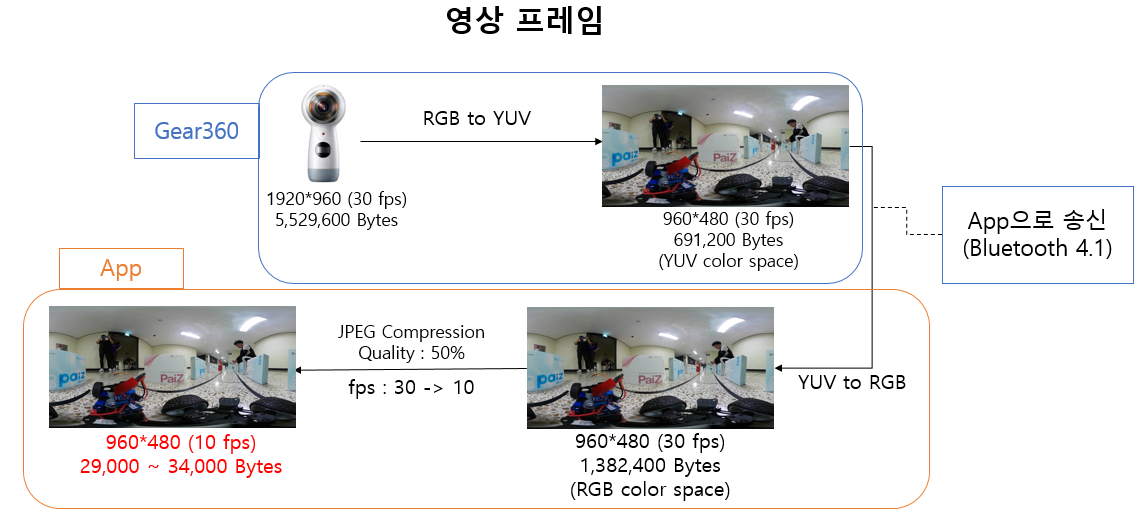
[Figure 2] 통신 흐름도

2.2 관련 기술

본 프로젝트에서 가장 큰 비중을 차지하는 기술이라 함은 영상 프레임의 압축과 통신 시스템의 구축 두가지 일 것이다. 초기 카메라로부터 오는 원시 프레임을 통신 규격에 맞게 재 framing 하여야 하고 이를 가장 optimal한 통신 프로토콜로 전송해서 영상의 delay가 가능한 한 적어야 한다.

2.2.1 Frame Compression

App에서는 1차로 YUV frame을 RGB-Bitmap으로 변환시켜주고 JPEG Compression (Quality :50%)을 통해 최종 데이터 크기를 30KB정도로 압축시켰다. (만약의 Traffic Congestion을 대비해서 fps를 30에서 10으로 낮춰주었다)



[Figure 3] Frame Compression

2.2.2 Media Transmission

(1) TCP

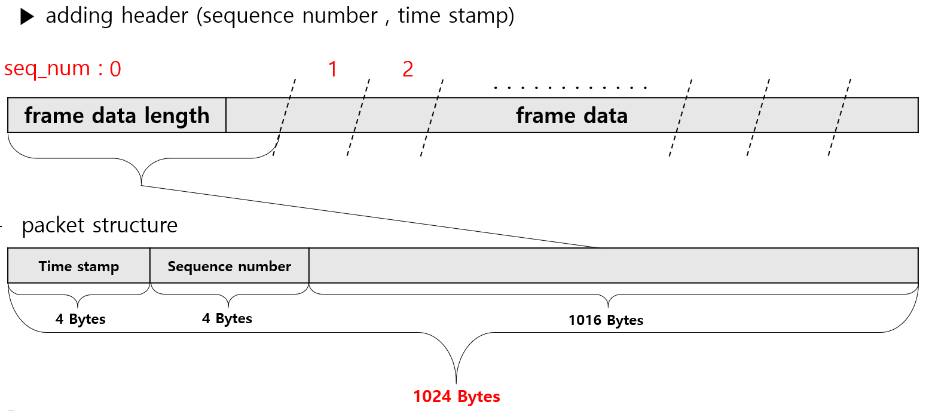
RTSP protocol을 참조하여 독자적으로 Media Transmission TCP protocol을 만들었다. Sender는 프레임의 길이를 측정하여 미리 초기화해둔 4byte크기의 byte배열에 저장 후 Receiver에게 보내준다. 이 후 Sender는 해당영상의 프레임을 pngbyte로 Encoding하여 Receiver에게 보낸다. Receiver는 미리 받을 프레임의 길이만큼의 프레임이 들어오게 되면 프레임을 인식하여 화면에 띄워 준다.

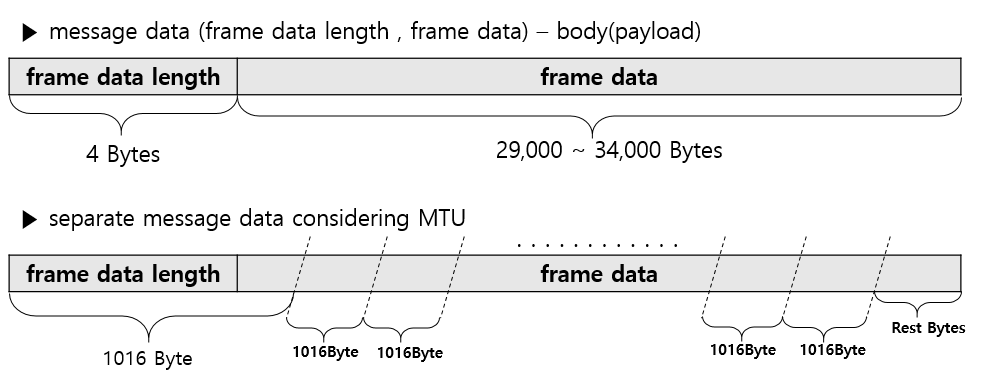
(2) UDP

본 작품에서 사용되지는 않았지만 실시간으로 packet을 송수신 하기 위해 연구했던 통신 프로토 콜이다. Ethernet MTU(1500byte)를 고려하여 packet framing 하였고 UDP Hole Punching 방식으로 구현하였다. Udp Hole Punching이란 서로 다른 NAT으로 막혀 있는 두 peer들 사이에 p2p통신을 하기위한 방법이다. 영상 Delay를 줄이기 위해 서버를 거치지 않도록 사용하였다.



[Figure 4] TCP/IP Packet Structure





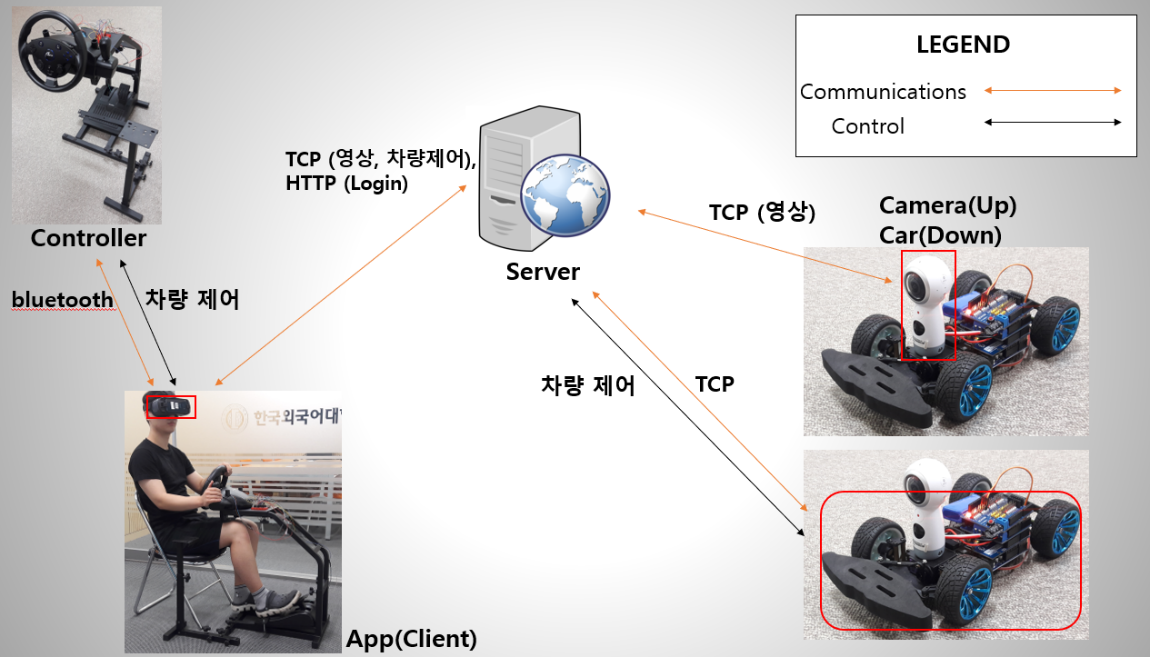
[Figure 5] UDP Packet Framing

1. 시스템 구성도 및 필요 장비

본 장에서는 서비스인 에 대한 전체 시스템 구성, Application infrastuructue 및 시스템을 구성하기 위한 개별 소프트웨어, 하드웨어를 기술한다.

전체 시스템의 구성도는 3.1과 같이 이루어져 있으며 Application infrastructure 구성 은 3.2절, 시스템을 구성하기 위한 개별 소프트웨어에 대한 소개는 3.3절, 하드웨어에 대 한 소개는 3.4절에서 다룬다.

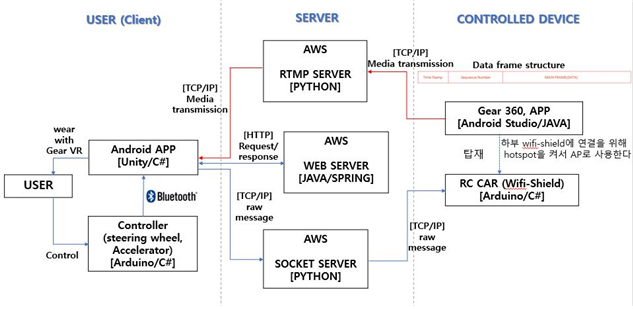
3.1 전체 시스템 구성도



[Figure 6] System Diagram 3.1

3.2 세부 시스템 구성도

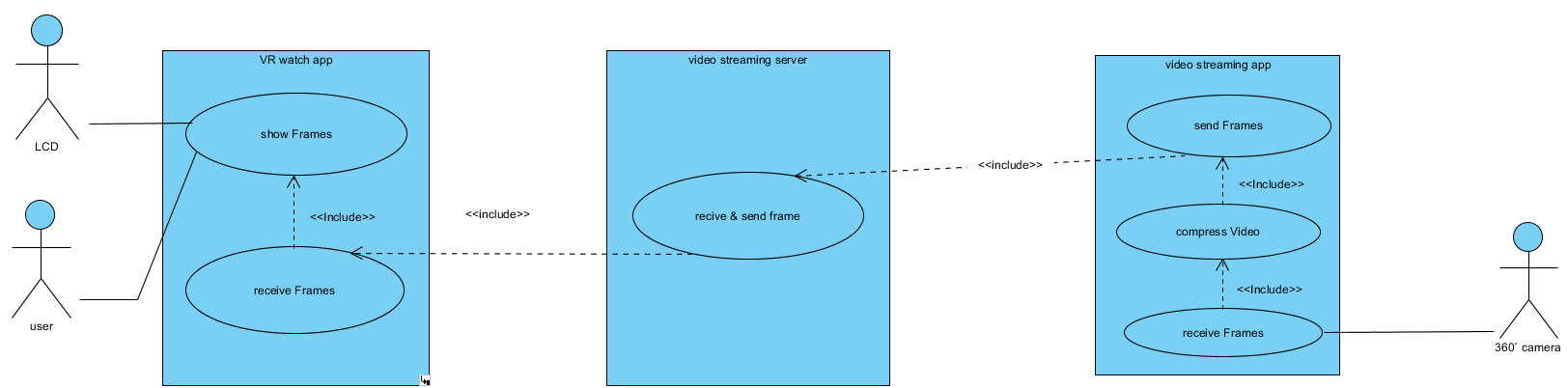
3.2.1 통신 시스템



[Figure 7] System Diagram 3.2.1

3.2.2 VR Watch Application Use Case Diagram

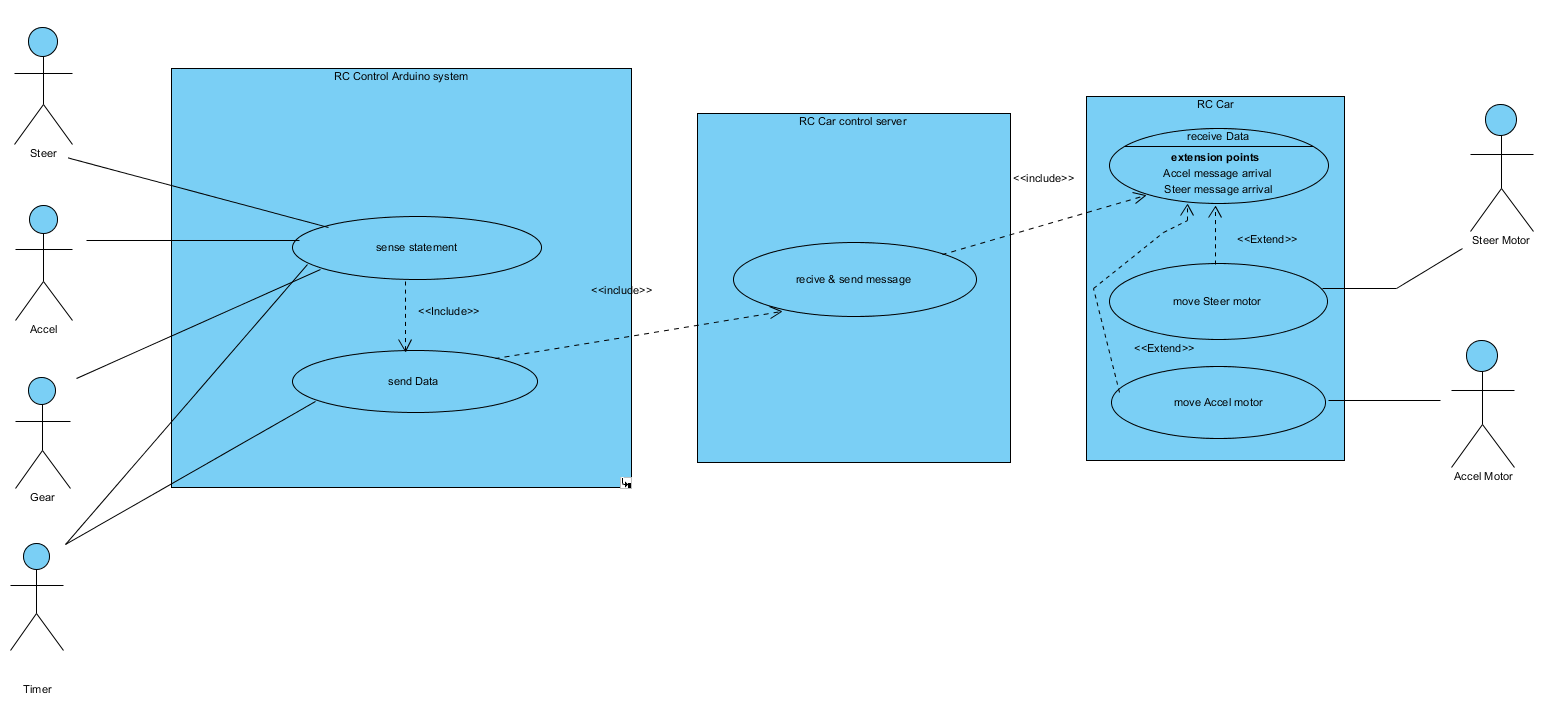
Streaming App은 360° 카메라로부터 원시 프레임을 수신 받는다. Application에서는 영상을 압축한 뒤 영상 스트리밍을 위한 서버로 보내주게 된다. 서버는 게이트웨이 역할을 하며 다시 User가 사용하는 Application으로 송신하고 User의 VR Application은 수신 받은 데이터를 Gear VR을 통하여 스크린에 출력하여 User가 볼 수 있도록 한다.



[Figure 8] VR Watch Application Use Case Diagram

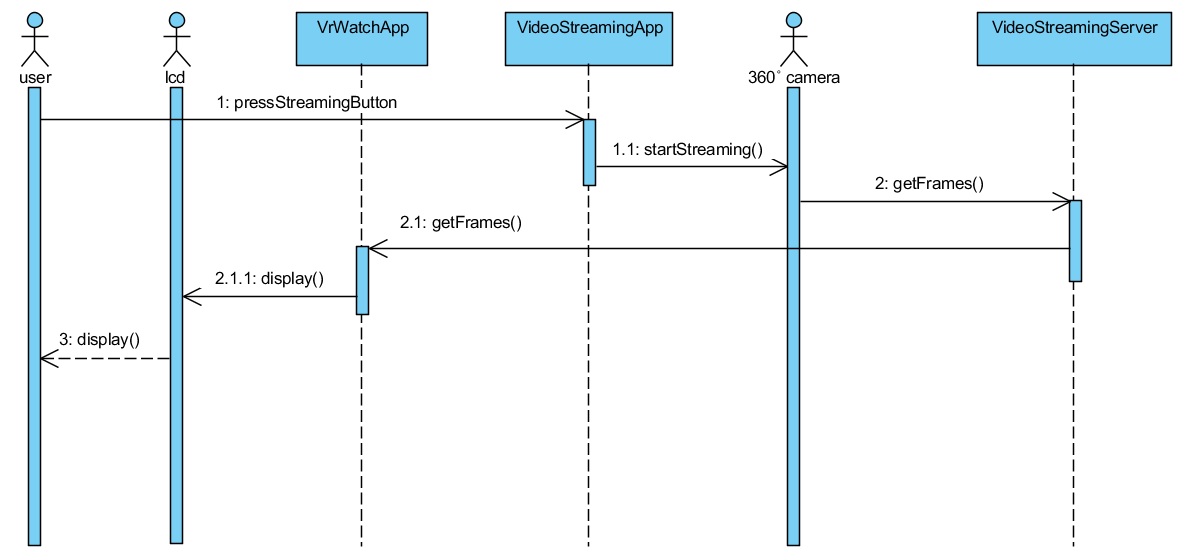
3.2.3 Controller Use Case Diagram

각 컨트롤러 장치(accel, gear, steer)는 Sensing한 State를 Control 서버로 송신한다. Control 서버는 게이트웨이 역할을 하며 RC Car로 State를 넘겨주고 RC Car에 부착된 Steer motor와 Accel motor를 조작하게 된다.



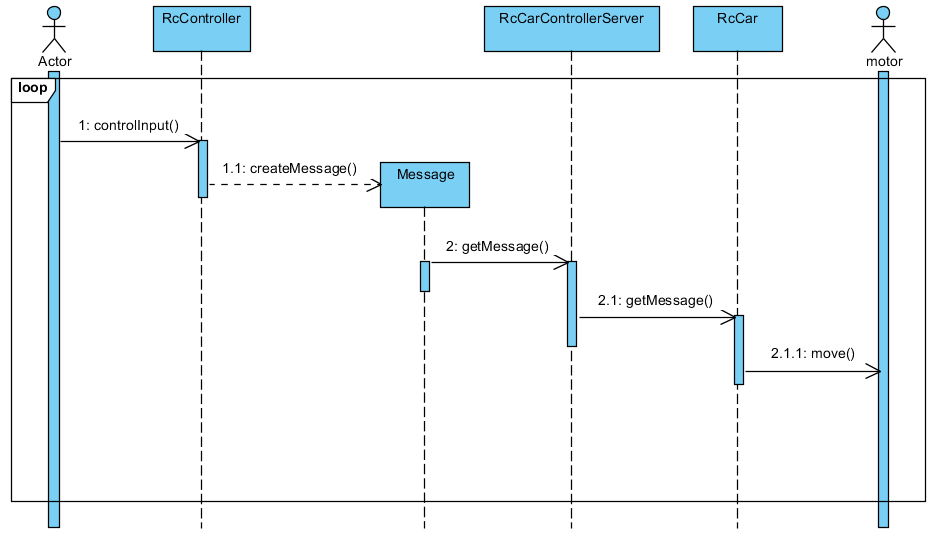
[Figure 9] Controller Use Case Diagram

3.2.4 Using VR Application sequence diagram



[Figure 10] Using VR Application sequence diagram

3.2.5 Control RC Car Sequence diagram



[Figure 11] Control RC Car Sequence diagram

3.3 소프트웨어

3.3.1 Unity Engine



[Figure 12] Unity Engine

GUI가 아주 직관적이며, 툴 사용이 서툰 사람이라도 편하게 사용하기 좋다. 다양한 플랫폼으로 빌드가 가능하다

3.3.2 Oracle



[Figure 13] Oracle

데이터베이스를 관리하며 응용 프로그램들이 데이터베이스를 공유하며 사용할 수 있는 환경을 제공하는 소프트 웨어로써 윈도우, 리눅스, 유닉스 등 다양한 운영체제에 설치를 할 수 있다. MS\_SQL , MY\_SQL보다 대량의 데이터를 처리하기 좋다.

3.3.3 Android Studio



[Figure 14] Android Studio

Android Studio는 Android 앱 개발을 위한 공식 통합 개발 환경(IDE)이며, IntellJ IDEA를 기반으로 한다. Android Studio는 Android 앱을 빌드할 때 생산성을 높여주는 다양한 기능을 제공한다.

3.3.4 Arduino(Sketch)



[Figure 15] Arduino(Sketch)

차량에서 bluetooth 통신을 통해 컨트롤러 정보를 app으로 송신 RC Car Server로부터 정보를 받아 RC Car가 작동할 수 있도록 구현했다. 컨트롤러에서는 컨트롤러를 통하여 움직임 정보를 블루투스로 연결된 Service App으로 전달하도록 구현했다.

3.4 하드웨어

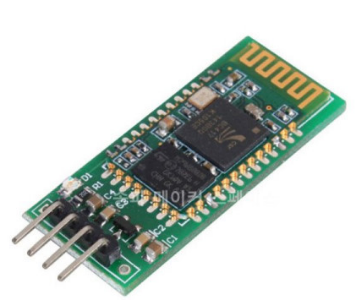
3.4.1 Arduino



[Figure 16] Arduino

오픈 소스를 지향하는 마이크로 컨트롤러를 내장한 기기 제어용 기판으로 회로기판에서 다양한 센서나 부품 등 디바이스를 연결 할 수 있다. 본 프로젝트에서는 원격 차량 조종을 위해 주변장치들 간의 통신을 제어하고 블루투스 통신을 통하여 원격 차량본체를 조종하는 용도로 사용한다.

3.4.2 HC-06 블루투스 모듈



[Figure 17] HC-06 Bluetooth Module

안드로이드 App과 아두이노간의 블루투스 통신을 위해 Arduino에 HC-06 모듈을 장착하여 블루투 스 기능을 가능하게 만드는 모듈이다. 가변저항 센서로부터 받은 문자 값을 차량과 연결된 App에 전달하기 위한 통신 인프라이다.

3.4.3 360도 카메라



[Figure 18] Samsung 360 Camera

기존 2D 영상만을 제공하는 카메라와 달리 실제 화면 영상을 360도 카메라의 스티칭기법을 활용하여 4D 영상을 제공함으로써 전방향의 상황을 전달한다.

3.4.4 RC Car



[Figure 19] RC Car

RC Car 차량에 아두이노 우노, 와이파이 쉴드, 센서 쉴드 등을 탑재하여 서버와 통신을 구축한다. Python3으로 개발된 TCP 서버로 Client(App)에서 전송된 차량 제어에 대한 데이터를 수신하여 이를 차량에 송신해준다.

3.4.5 Controller

3.4.5.1 Controller Handle



[Figure 20] Controller Handle

하드웨어를 수치적으로 계산하여 각부분에 가변 저항센서를 용하여 오른쪽 또는 왼쪽으로 돌아갈 때 어느 쪽으로 돌아가는지 알 수 있게 하였다.

3.4.5.2 Controller Gear



[Figure 21] Controller Gear

하드웨어를 수치적으로 계산하여 각부분에 가변 저항센서를 이용하여 차량의 D(전진모드), R (후진모드), N(중립)을 구현하였다.

3.4.5.3 Controller Pedal



[Figure 22] Controller Pedal

하드웨어를 수치적으로 계산하여 각부분에 가변 저항센서를 이용하여 차량의 전진, 후진을 할 수 있도록 구현하였다.

3.4.6 기어 VR



[Figure 23] Gear VR

Samsung gear 360 SDK를 활용하여 영상 스트리밍을 구현하고 이를 gear VR를 통하여 운전자가 직접 착용하여 주행중인 차량의 주변상황을 제공받는다.

1. 요구사항 및 제한사항

본 장에서는 VR과 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량 시스템의 요구사항을 인터페이스 요구사항, 시스템 기능적 요구사항, 그리고 사용자 요구사항으로 구 분하여 기술하고 본 프로젝트인 원격지에서 생동감 있게 운전을 할 수 있는 360도 카메라를 이용한 실시간 스트리밍 원격제어 차량 시스템의 제한사항과 한계점에 대해 기술한다.

4.1 인터페이스 요구사항

VR과 360도 카메라를 이용한 실시간 스트리밍 원격 제어 차량 시스템은 여러가지 인터페이스들을 고려해야 한다. 먼저 VR과 360도 카메라를 이용한 실시간 스트리밍 원격 제어 차량이란 360도 카메라로 촬영한 영상을 사용자의 어플리케이션으로 보내 VR로 장착하여 실시간으로 해당 영상을 보는 기술을 말한다. 이 시스템에 필요한 여러가지 인터페이스들을 기술한다.

4.1.1 시스템 인터페이스 기능

• 사용자 인터페이스

1. 운전대 연결
2. 차량을 운전하기 위해 운전대 조작
3. 차량의 상황을 보기위한 영상 어플리케이션을 VR에 장착
4. 차량 운전 및 종료

|  |  |
| --- | --- |
| 사용자 인터페이스 요구사항 | |
| 데이터 형식 | 텍스트, 이미지, 동영상 등 |
| 사용 가능 기능 | 기본적인 운전, 운전상황 정보 습득 |
| 사용 불가능 기능 | 기어 변속으로 인한 속도 조절 |

[Table 3] 사용자 인터페이스 요구사항

4.1.2 하드웨어 인터페이스 요구사항

|  |  |  |
| --- | --- | --- |
| 이름 | 수량 | 데이터 형식 |
| 컴퓨터 | 1대 | Byte |
| 스마트폰 | 2대 | Char, Byte |
| 컨트롤러(운전대) | 1대 | Char |
| RC Car | 1대 | Char |
| 360도 카메라 | 1대 | Byte |

[Table 4] 하드웨어 인터페이스 요구사항

차량 조절 메시지와 실시간 영상 스트리밍 프레임을 보내주기 위한 중계자 역할 서버 컴퓨터 1대, 360도 영상을 찍어주는 360도 카메라 1대, 찍은 영상을 실시간으로 보내주는 어플리케이션, 해당 영상을 받아서 보는 어플리케이션을 사용하기 위한 스마트폰 2대, 차량을 제어할 운전대 1대, 차량 1대가 필요하다.

4.1.3 소프트웨어 인터페이스 요구사항

|  |  |
| --- | --- |
| 소프트웨어 인터페이스 요구사항 | |
| 구현 환경 | Python3, Arduino, Unity,  Android Studio, Spring, Java, C# |
| 인터페이스 목적 | 영상정보 제공, 차량 제어, 로그인 서비스 |
| 데이터 형식 | 텍스트, 이미지, 동영상 등 |
| 운영체제 | Ubuntu |
| 데이터 베이스 | Oracle |

[Table 5] 소프트웨어 인터페이스 요구사항

차량 조절 메시지와 실시간 영상 스트리밍 프레임을 보내주기 위한 중게자 역할 서버 컴퓨터를 Python3의 환경에서 개발, 차량과 차량을 제어하는 운전대를 Arduino 환경에서 개발, 360도 카메라에서 찍은 영상을 서버로 보내주는 것을 Android Studio 환경에서 Java로 개발, 영상을 서버로부터 받아서 보는 것을 Unity 환경에서 C#으로 개발, 로그인과 차량 시리얼 번호 인증을 Spring의 환경에서 개발하였다.

4.1.4 네트워크 인터페이스 요구사항

|  |  |
| --- | --- |
| 네트워크 인터페이스 요구사항 | |
| 메시지 형식 | 텍스트, 동영상 등 |
| 프로토콜 | TCP/IP, HTTP, 근거리 P2P |
| 사용자 인증 | 있음 |
| 제한 사항 | 네트워크 연결 없는 경우 서비스 사용 불가 |

[Table 6] 네트워크 인터페이스 요구사항

TCP/IP 프로토콜은 실시간 영상 스트리밍 기술과 차량 제어 기술 구현 시에 사용, HTTP 프로토콜은 사용자 인증 기술 구현 시에 사용, 근거리 P2P 프로토콜(블루투스)는 차량을 제어하려는 운전대 기술 구현 시에 사용하였다.

4.2 사용자 요구사항

본 장에서는 Gear360와 스마트폰 간의 실시간 3D 영상 스트리밍 기술을 활용하여 현장감을 제공하는 시스템과 이를 사용하는 서비스 객체를 위한 사용자 요구사항을 기능적 요구사항, 비기능적 요구사항으로 구분하여 기술한다.

4.2.1 사용자 기능적 요구사항

|  |  |  |
| --- | --- | --- |
| **기능** | **상세 내용** | **출처** |
| **거리제약 없는 제어** | **사용자(컨트롤러) / 차량 간 거리의 문제**  본 시스템의 사용자는 인터넷(wifi)접속이 가능한 장소에서 어플을 실행만 시킨다면 어디에서나 차량에 접속하여 제어를 할 수가 있다.  (본 프로젝트의 차량은 LTE HOTSPOT을 활용하여 네트워크에 접속 되어있기 때문에 지구상 어디라도 제어를 받을 수 있다.) |  |
| **현장감 제공** | **3D영상을 활용한 영상 스트리밍**  본 시스템의 사용자는 Gear360 카메라로 촬영한 3D영상을 수신 받고 이를 사용자가 착용한 Gear VR을 통해 제공받음으로써 타 제품과는 차별화된 현장감을 제공한다. |  |
| **회원,차량 관리 시스템** | **회원, 차량 관리를 위한 데이터베이스**  본 시스템은 사용자가 본인 식별과 소유 차량식별 및 접근을 용이하게 하기위해 back-end 웹서버를 활용하였다. 사용자는 회원가입, 로그인을 통해 일전에 등록한 차량으로 접근이 가능하다. |  |

[Table 7] 사용자 기능적 요구사항

4.2.2 사용자 비기능적 요구사항

|  |  |  |
| --- | --- | --- |
| **분류** | **상세 내용** | **출처** |
| **신뢰성** | - 다른 통합 개발 환경으로 재현이 불가능한 VR영상을 세계에서 검증된 Unity Engine을 통해 구현하였다.  - 자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크인 Spring을 사용하여 안 드로이드와 연동을 통해 데이터베이스로 접근 시 보안적인 이슈를 해결 할 수 있 으며 쉽게 App과 Web간의 통신이 가능하다. |  |
| **성능** | - 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 스티칭 기법을 사용하여 3차원 이미지를 송⋅수신 한다.  - P2P통신을 대체한 Socket Server/Client를 통신에서의 거리 제약을 해결하였다. |  |
| **지원성** | - 누구나 쉽게 접할 수 있게끔 스마트폰 어플리케이션의 개발로 진입장벽을 낮추었다. |  |

[Table 8] 사용자 비기능적 요구사항

4.3 시스템 구현의 제한사항

1. 현재 무선 통신망의 대역폭으로는 송⋅수신 하는 영상의 사이즈로 인한 지연
2. 서버로 전송하기 위해 프레임 압축으로 인한 지연

위 두가지의 이유로 인해서 본 시스템은 차량 주변에서 발생하는 시간과 사용자가 이를 영상으로 수신 받는 시간과의 차이가 0.4초 정도의 지연이 발생을 한다.

차량의 속도가 낮을 경우에는 사용자의 체감상 불편함을 느끼지 못하지만, 속도가 빠를 경우에는 사용자가 차량을 제어하는데 불편함을 느끼게 된다.

1. 자체 시험 방법 및 절차

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

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

|  |
| --- |
|  |

1. 부록
2. 소스코드 첨부