

# 스마트폰을 이용한 지능형 모형 자동차의 원격 제어에 관한 연구

김재영, 박성근, 김동현, 전희성  
울산대학교 전기공학부  
e-mail : hsjun@ulsan.ac.kr

## A Study on Remote Control for Intelligent Model Car using Smart Phone

Jae-Young Kim, Sung-Geun Park, Dong-Hyeon Kim, Heesung Jun  
School of Electrical Engineering, University of Ulsan

### 요 약

본 연구에서는 스마트폰을 이용한 실험실용 모형 지능형 자동차의 원격 제어 시스템을 구현하였다. 기존의 임베디드 기기(지능형 모형 자동차) 원격 제어 시스템은 단순한 1:1 통신으로 조종 기기와 임베디드 기기 사이에 일정 간격을 유지해야 하는 단점이 있다. 본 시스템은 인터넷 망을 이용하여 구현하였으므로 조종 기기와 임베디드 기기와의 네트워크 통신이 가능하다면 원격 조종이 가능한 거리의 제한이 없어진다. 사설 와이파이끼리의 P2P 연결에서 공인아이피를 알아내기 위해 UDP서버 경유방식을 사용하였다. 그리고 기기 근처에 연결 가능한 무선 네트워크망이 있어야 한다는 단점을 개선하기 위해 Ad-Hoc방식을 구현하였다. 또한 일반패킷 구조를 사용하는 것 보다 직접 구현한 데이터 패킷 구조를 사용하여 이미지 및 센서 데이터를 전송한 결과 전송속도가 훨씬 개선하였다.

### 1. 서론

최근 스마트폰 시장이 급격히 성장하고 있는 추세이고, 스마트폰을 이용한 각종 응용 프로그램이 배포되고 있다. 스마트폰의 기능이 다양화되어 가고 있으며 처리속도와 무선 네트워크 통신 속도 또한 향상되고 있다. 앞으로 스마트폰을 활용한 다양한 원격 시스템과 영상처리 시스템들이 연구될 것으로 전망된다.

현재 스마트폰을 이용한 원격제어에 관한 많은 연구들이 진행 중에 있다. 스마트폰을 이용한 원격 차량 시동 시스템, 스마트 홈 관리 시스템, 컴퓨터 원격 제어 시스템 등의 원격제어 시스템은 이미 개발되어 있다. 하지만 스마트폰을 이용한 차량 원격 운전에 관한 연구는 아직 상용화된 것이 없으며 기술적 문제와 안전성 문제가 해결되지 않고 있다.

본 논문에서는 스마트폰(안드로이드폰)을 이용하여 와이파이 모듈이 설치된 지능형 모형 자동차를 원격 운전하기 위해 안정된 연결과 실시간 전송을 구현한 네트워크 시스템을 연구하였다.

원격제어 연결 네트워크는 실시간 영상획득과 원격제어를 위해 전송속도를 증가시키기 위해 P2P(Peer to Peer) 기반과 Ad-Hoc기반으로 구현하였다. 2절에서는 원격제어 연결에서 사용되는 알고리즘과 전송 방식에 대해서 설명하고, 3절에서는 원격제어를 위해 송수신되는 데이터 패킷의 구조와 이 구조의 장점에 대해 설명한다. 4절에서는 원격 제어를 할 때 모형 자동차와 스마트폰 사이의 통신 알

고리즘을 설명할 것이다. 5절에서는 홉핑칭 테스트와 실제 구현한 데이터 패킷 구조를 사용하였을 때 성능 평가를 해보았다.

### 2. 원격 제어 연결 방식

스마트폰을 모형 자동차로 연결하기 위해서는 모형자동차의 아이피 주소와 포트번호를 알아야한다. 모형자동차는 유동아이피를 사용하므로 연결을 위해서는 매개역할을 할 수 있는 서버가 필요하다.

클라이언트와 서버사이의 연결은 TCP(Transmission Control Protocol)로 이루어지고 클라이언트끼리의 연결은 UDP(User Datagram Protocol)로 이루어진다.

클라이언트끼리의 연결을 UDP로 구현한 이유는 두 가지가 있다. 첫째, UDP는 전송속도가 TCP에 비해 빠르다. 따라서 전송속도가 중요한 실시간 원격제어 시스템에서는 UDP를 사용하는 것이 안정적이다. 제어 명령의 패킷 오류는 체크섬 비트(Check Sum Bit)를 이용하여 예방할 수 있다. 둘째, 사설아이피끼리의 연결에서 TCP로는 P2P 연결을 완벽하게 할 수 없다. TCP는 연결 지향(connection-oriented)이기 때문에 접속에 사용될 소켓은 한 번 접속하게 되면 연결된 소켓과의 통신만이 가능하다 [4]. 또한 소켓을 새로 할당한다 하더라도 라우터의 종류에 따라 다음 포트번호를 할당해주는 방법이 다르다. 따라서 서버를 통해 접속하려는 소켓의 아이피와 포트번호를 정확히 알더라도 사설아이피의 소켓들은 재사용될 수



## 2.2 Ad-Hoc 방식

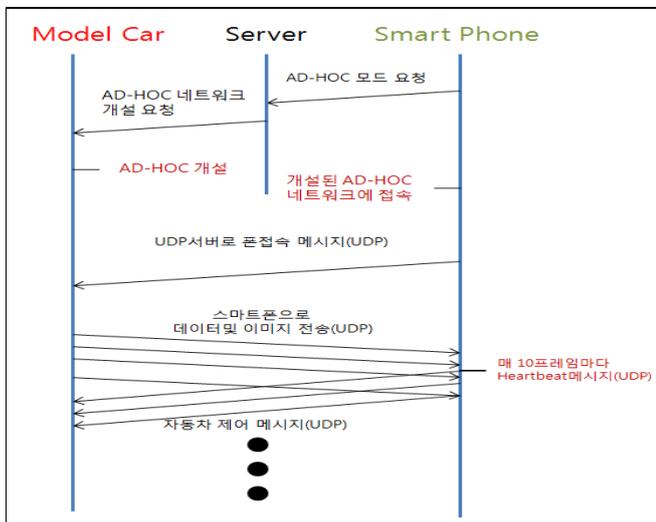
Ad-Hoc 방식은 인터넷망을 경유하지 않고 로컬 네트워크망을 구축하여 1:1로 연결할 수 있는 방식이다. 이 방식은 지능형 자동차 주변에 연결가능한 공유기가 없더라도 원격제어를 할 수 있으며, 외부 인터넷망에 관계없이 전송속도를 유지할 수 있다. 그러나 자동차에서 네트워크망을 구축하기 때문에 자동차와 스마트폰 사이에 유효한 거리가 있다는 단점이 있다. (그림 3)은 Ad-Hoc 연결 시 모형 자동차와 스마트폰 사이에 전송하는 메시지를 순서대로 나타낸 것이다.

### 2.2.1 지능형 모형 자동차

처음 지능형 자동차가 부팅될 때 디폴트로 접속 가능한 공유기를 찾아서 인터넷에 연결하게 되어있다. 그러나 연결가능한 공유기를 찾지 못하거나 서버로부터 스마트폰이 Ad-Hoc 연결 요청을 했다는 메시지를 받게 되면 자동차는 Ad-Hoc 네트워크를 구성하고 스마트폰의 접속을 기다린다. 스마트폰이 접속되면 로컬 네트워크망으로 1:1 원격제어가 가능하게 되며, 연결이 끊어지거나 일정시간동안 연결이 없을 경우 다시 서버로 접속을 시도한다.

### 2.2.2 스마트폰

서버에 Ad-Hoc접속 요청을 하고 자동차에서 구축한 로컬 네트워크로 접속한다. 그 후 자동차에 구축된 서버에 접속하고 원격제어 통신을 한다.



(그림 3) Ad-Hoc연결 순서도

## 3. 데이터 패킷 구조

서버에 접속할 때는 TCP로 통신하고 클라이언트끼리 접속할 때는 UDP로 통신한다.(2절 참조) 이 절에서는 UDP에서 영상 및 센서 데이터를 빠르게 전송하기 위한 방법을 설명한다.

UDP는 TCP와 달리 Data Stream으로 전송하지 않고 Datagram으로 패킷단위로 전송한다. 한 패킷당 보낼 수

있는 최대 데이터량은 8KB로 제한되어 있어 8KB보다 큰 데이터를 전송해야 할 경우 한 패킷에 전송할 수 없고 둘 이상의 패킷을 보내야 한다.

대부분의 카메라에서 영상 포맷은 YUV이며 YUV는 압축을 사용하지 않는 포맷이다. 8KB이상의 YUV이미지를 실시간 전송하기에는 속도가 너무 느리고, UDP를 사용하기 때문에 패킷이 잘못된 순서로 도착할 경우 올바른 이미지를 얻을 수 없게 된다. 따라서 YUV를 JPEG 포맷으로 압축/변환하여 보내면 변환 시간은 전송시간에 비해 매우 빠르므로 전송속도가 빨라진다.

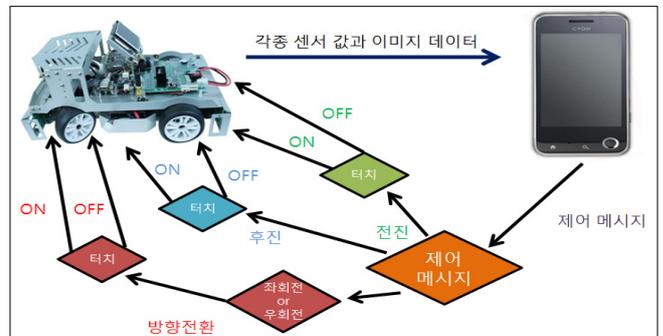
지능형 자동차에 장착된 여러 센서값들은 여러 패킷으로 보내는 것보다 한 패킷으로 보내는 것이 더 빠르다. 따라서 영상 데이터를 보낼 때 영상 데이터 버퍼에 센서값들을 붙여서 보내는 것이 효율적이다. 한 패킷의 기본 구조도는 (그림 4)와 같다.



(그림 4) 영상 및 센서 데이터 패킷의 구조

## 4. 지능형 모형 자동차의 제어

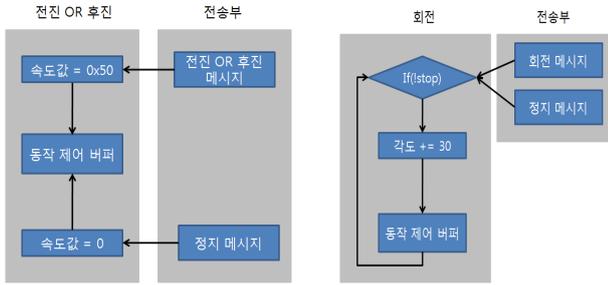
스마트폰을 이용해 자동차를 제어하려면 먼저 이미지를 받는 소켓과는 별도로 제어를 위한 또 다른 소켓을 생성하고 그 소켓을 통해 명령메시지를 보내 제어한다. 제어 명령 메시지는 전등 ON/OFF메시지와 모터제어 메시지가 있고, 자동차로부터 받는 센서값은 적외선 거리 센서와 라인센서가 있다. (그림 5)는 제어를 위해 주고받는 전송 메시지를 보여준다.



(그림 5) 스마트폰과 자동차의 통신

지능형 모형 자동차의 제어 방식은 모터버퍼에 8비트 16진수로 0x00부터 0xFF까지의 값을 입력하여 모터를 제어할 수 있다. 제어 버퍼는 전·후진 그리고 방향전환 버퍼가 있으며, 스마트폰에서 전진 메시지를 전송하면 자동차는 전진 모터제어 버퍼에 값을 입력하게 되어 전진하고 멈춤 메시지가 전송되면 자동차가 멈추게 된다. 방향전환 버퍼는 두 개가 있는데 첫번째 버퍼는 모터를 큰 각도 단위로 움직일 수 있게 하고 두번째 버퍼는 세밀한 각도로 바퀴의 회전을 좀 더 부드럽게 할 수 있다. 스마트폰에서 버튼을 누르는 순간 동작 명령 메시지가 전송되고 때는 순간 정지메시지가 전송된다. 전진과 후진의 경우 정지 메

시지가 전송되면 버퍼에 속도값을 0을 주어 정지시키고 회전시 경우 정지메시지를 받을 때까지 반복문으로 회전 각도를 조금씩 증가시키거나 감소시킨다. (그림 6)



(그림 6) 전진/후진과 회전 제어 블록다이어그램

### 5. 실험 및 결과

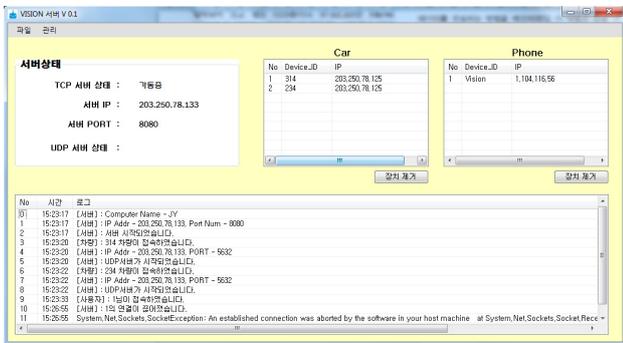
실험에서 사용된 장비는 (그림 7)에 보이는 Optimus Q(Android 2.2, 퀄컴 스냅드래곤 1Ghz CPU, 512MB 램) 와 ㈜하이버스의 지능형 모형 자동차(H-AICAR)이다. H-AICAR에는 Marvel PXA320-P CPU(Core806MHz)와 266MHz 128MB의 DDR SDRAM이 탑재되어 있다. 화면은 자동차의 좌우방향을 조종하는 터치조이패드와 전진후진 버튼으로 구성되어 있고 전진(후진) 버튼을 연속해서 누를 경우 가속이 가능하다. 반대방향 버튼을 누를 경우 감속이 된다.



(그림 7) 원격 제어 시연

#### 5.1 서버

(그림 8)은 구현한 서버의 실제 작동 모습을 보여준다. Car리스트에는 현재 접속된 차량의 목록과 차량번호를 Phone리스트에는 현재 접속중인 폰들의 목록을 보여준다.



(그림 8) 작동중인 서버

#### 5.2 데이터 패킷 구조

실험에서는 YUV422포맷을 사용하였으며 320\*240 이미

지의 경우 153600Byte이고 이는 150KB정도이다. 이 데이터를 바로 보낼 경우 총 18패킷을 반복해서 보내야 한다. 하지만 이런 경우 실시간처리가 불가능할 정도로 이미지를 느리게 받게 된다. 이러한 문제점을 해결하기 위해 YUV422를 JPEG으로 변환하여 보낸다. YUV422 포맷의 320\*240이미지를 25% Quality의 JPEG 포맷으로 변환할 경우 3000~4000Byte가 된다. 변환 시간은 전송시간보다 무시할 수 있을 정도로 빠르므로 약 18배 정도 빨라진다.

### 6. 결론

본 논문에서는 스마트폰을 이용한 조종 기기와 임베디드 기기간의 일반 네트워크 망을 경유하여 보다 빠르게 데이터를 전송하는 방법을 제안하였다. 이 방법은 조종 기기와 임베디드 기기(지능형 자동차)의 원격 조종 가능한 거리의 제한을 없앨 수 있다. P2P는 인터넷망을 경유하면서 클라이언트 간의 1:1연결로 전송속도를 최대한 보장해주었고, Ad-Hoc방식은 지능형 자동차 주변에 연결가능한 공유기가 없더라도 원격제어를 할 수 있도록 해준다.

다양한 센서를 사용하여 원격 제어를 자동화하고 영상처리를 활용하여 제어에 도움이 될 수 있도록 하는 것이 본 프로젝트의 최종목표이다.

향후 무선인터넷의 발달로 3G망이 속도가 빨라지면 자동차로 3G망을 이용하여 거리 제약과 공유기 유무에 관계없이 3G망을 통해 스마트폰으로 무선원격제어가 가능할 것이다. 또한 전송하는 이미지의 질을 높여 좀 더 선명한 화질로 원격제어가 가능할 것이다.

### 감사의 글

이 논문은 현대중공업 지원에 의한 울산대학교 전기공학부 일류화 연구비에 의하여 연구되었음

### 참고문헌

- [1] Bryan Ford, Pyda Srisuresh, Dan Kegel "Peer-to-Peer Communication Across Network Address Translators" Available : <http://www.brynosaurus.com/pub/net/p2pnat/>
- [2] Beej Jorgensen "Beej's Guide to Network Programming Using Internet Sockets" Available : [http://beej.us/guide/bgnet/output/print/bgnet\\_USLetter.pdf](http://beej.us/guide/bgnet/output/print/bgnet_USLetter.pdf)
- [3] Nadeem Abdul Hamid "Peer to Peer Programming" Available : <http://cs.berry.edu/~nhamid/p2p/>
- [4] James F. Kurose, Keith W. Ross "Computer Networking A Top-Down Approach" 5th Ed. Pearson, 2009
- [5] W. Richard Stevens, Bill Fenner, Andrew M. Rudoff "UNIX Network Programming" 3rd Ed. Addison Wesley, 2004