

PC에서 유닛 제어를 위한 다중 스마트폰의 통신 인터페이스 연구

Study on Communication Interface of Multiple Smartphones for Unit Control in a PC

정 하 민, 김 동 헌*

(Hahmin Jung¹ and Dong Hun Kim¹)

¹Department of Electrical Engineering, Kyungnam University

Abstract: This study presents the unit control of a PC (Personal Computer) program using multiple smartphones based on ad hoc communication. A design of the data packet that multiple smartphones send to a computer, and a framework that manages and controls the unit in a computer after analyzing the data packet, are proposed. As a result, multiple users are able to control their own units using their smartphones while seeing the monitor connected to a computer. In other words, multiple users can share the same game in a computer or control their units embedded in a system using their smartphones. An experimental result shows that a racing game in a PC can be realized by the proposed communication interface, where four iPhones are used to control their units in a computer. Thus, the proposed framework can be effectively used for unit control in a PC using multiple smartphones.

Keywords: smartphone, ad hoc communication, unit control, communication interface, iPhone

I. 서론

스마트폰의 등장과 많은 용도의 어플리케이션을 통해 우리의 생활은 점차 편해지고, 윤택해지고 있다. 현재의 스마트폰은 S/W 중심으로 어플리케이션이 발달 되었지만, 스마트폰을 컨트롤러로 사용하기 위해 산업용, 군사용, 의료용 등 많은 분야에서 연구되어지고 있다[1-3]. 애플 아이폰과 삼성 갤럭시 모델과 같은 현재의 스마트폰은 매우 얇고 가볍다. 하지만 이러한 작은 크기에도 불구하고, 카메라, GPS, 블루투스, 무선통신, 가속도센서, 지자기센서등과 같은 매우 다양하고 많은 H/W 기능들을 가지고 있다. 기술이 발전할 수록 스마트폰은 더 높은 사양의 H/W 기능들을 갖게 될 것이다. 따라서 기술적인 관점에서 스마트폰은 휴대하기 가장 좋은 최고의 제어기가 될 수 있다[4].

이동로봇의 원격 무선제어를 위해서 일반적으로는 적외선 방식을 많이 사용하지만, 현재 스마트폰의 획기적인 기술의 발달로 아이폰과 같은 휴대전화기를 무선제어기로 사용하는 기술들이 연구되어지고 있다[5]. 스마트폰을 무선제어기로 사용하면 별도로 다른 제어기를 구매할 필요가 없다. 또한 아이폰과 같은 휴대전화기는 화면에 현재의 제어 상황과 같은 많은 정보를 사용자에게 제공할 수 있으므로 장점이 매우 많다.

* 책임저자(Corresponding Author)

Manuscript received January 14, 2013 / revised April 17, 2013 / accepted April 18, 2013

정하민: 경남대학교 첨단공학과(zerofull1@naver.com)

김동헌: 경남대학교 전기공학과(dhkim@kyungnam.ac.kr)

※ 본 연구는 2013학년도 경남대학교 학술연구 장려금 지원으로 이루어졌음.

저자는 스마트폰을 이용해 모니터에 있는 가상의 이동로봇을 제어하는 시뮬레이터를 연구하였다[6]. 이 연구에서 개발된 자동 운전 시스템은 이동로봇의 제어를 위해 애드혹 무선통신을 사용하였다. 차동구동 로봇을 대상으로 아이폰의 가속도 센서를 사용하여 아이폰을 자동 무선 핸들처럼 사용 하였다. [7]에서 저자는 아이폰을 원격 제어기로 사용하기 위한 세 종류의 사용자 인터페이스를 제안하였다. 본 연구는 [6]와 [7]의 확장 연구로 그림 1과 같이 한 대의 스마트폰이 아니라 여러 대의 스마트폰으로 하나의 PC와 모니터에서 무선 통신을 하며 자신의 유닛을 제어하는 인터페이스에 대하여 다룬다. 아이폰 한 대를 제어기로 사용하는 연구 사례들은 현재 유튜브(YouTube) 인터넷 사이트에서 외국의 응용사례를 찾을 수 있고[8-10], 하나의 시스템을 스마트폰 한 대로 제어하는 사례는 이미 연구되었다 [6,7,11]. 하지만 하나의 시스템에 있는 유닛(unit)을 여러 대의 아이폰으로 제어하기 위하여 데이터를 처리하는 무선 인터페이스 기술은 아직 많이 연구되어 지고 있지 않다. 유닛이란 제어기에 의해 조정되는 대상을 의미한다.

한 대의 시스템에서 다중 아이폰 사용자의 입력을 무선으로 받아서 처리할 수 있으면 다음의 사례들이 가능해진다. 예를 들어, 여러 스마트폰 사용자가 특별한 H/W 없이 스마트폰의 어플리케이션 센터에서 S/W를 다운로드 받은 후 한 대의 모니터 화면에서 여러 사람이 스마트폰으로 게임을 할 때, 혹은 여러 명이 동시에 한 대의 시스템에 있는 특정 유닛을 제어 할 때 사용할 수 있다. 이러한 이유로 본 논문에서는 한 대의 PC와 다중 스마트폰 제어기와 통신 인터페이스를 연구한다.

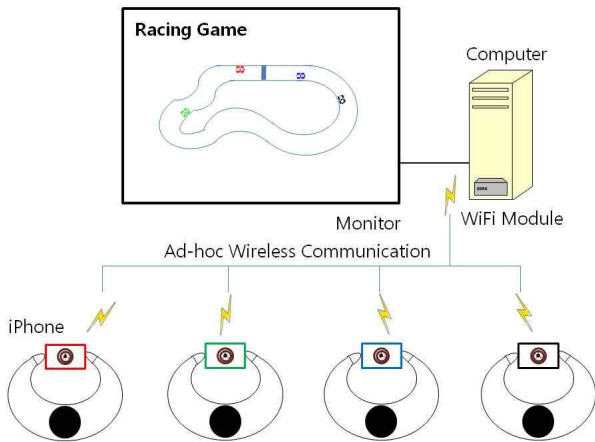


그림 1. 한 대의 PC와 다중 스마트폰과의 통신 인터페이스.
Fig. 1. Communication interface between a computer and multiple smartphones.

아이폰에 의한 제어방식은 아이폰 버튼을 사용하는 방식과 아이폰에 있는 가속도 센서를 이용하는 방법이 있다. [12]에서 저자는 실제 이동로봇을 대상으로 가속도 센서의 무선 원격제어 인터페이스 연구를 하였으며, 무선 통신 방법의 효율성을 분석하였다. 한 대의 스마트폰으로 이동로봇을 제어하는 연구로 아이폰에 내재된 가속도 센서를 이용해 아이폰을 좌우로 기울여 아이폰을 마치 자동차 핸들로 사용하였다. [12]의 연구와 제안된 연구와의 차이점은 한 대의 스마트폰이 아니라 다중의 스마트폰 제어를 사용하여 각각의 스마트폰에 해당된 시스템을 한 대의 PC로 제어하는 데 있다. 따라서 그림 1과 같은 구성이 가능해진다. 그림 1에서는 네 명의 사용자가 각각의 스마트폰을 무선원격제어기로 사용하며 한 대의 시스템(PC)에서 지원되는 경주 게임에서 자신의 유닛을 실시간으로 조정한다. 그림 1에서는 사용자의 스마트폰에 의해 조정되는 자동차가 유닛이 된다.

본 논문은 다음과 같이 구성되어 있다. II 장에서는 스마트폰에서 전송하는 패킷을 설계하고, 전송된 패킷을 PC에서 분석하여 사용하는 방법을 제안한다. III 장에서는 다중 스마트폰과 하나의 시스템과의 무선 통신을 위해 데이터를 전송하고, 수신하는 방법에 대해 제안한다. IV 장에서는 제안된 통신 인터페이스를 통하여 다중 아이폰으로 각각의 유닛을 제어하는 실험을 보여준다. 마지막으로, V 장에는 본 연구의 결론을 제시한다.

II. 인터페이스와 통신패킷 설정

본 연구에서는 애플OS를 기반으로 동작하는 아이폰을 스마트폰으로 채택하여 연구하였다. 본 장에서는 다중 아이폰으로 한 대의 PC 유닛을 무선 조정하기 위하여 본 연구에서 설계한 통신 방법과 인터페이스를 설명한다.

1. 스마트폰 인터페이스

본 논문에서 사용하는 아이폰 제어 인터페이스는 저자가 [7]에서 제안한 인터페이스 방식 중 터치모드에 기반한 인터페이스를 선택하여 사용한다.

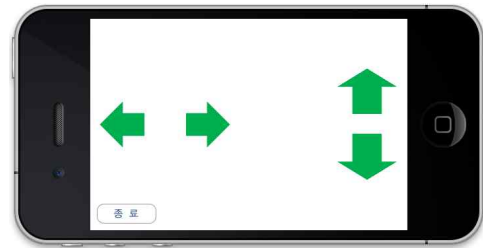


그림 2. 터치 센서를 이용한 인터페이스.
Fig. 2. Interface using touch sensor.

아이폰에서 제어 화면은 다음과 같은 방법으로 설계되어 사용자에게 의해 터치된 부분이 인식된다.

- a) 무선 조종 인터페이스를 위한 이미지 작성
- b) 해당 이미지를 아이폰 전면부에 배치
- c) 아이폰 전면부에 터치된 점들을 모두 저장
- d) 실제 그림 좌표와 터치된 좌표들을 비교하여 터치된 점의 좌표와 그림 좌표가 일치하면 해당 그림이 터치된 것으로 인식
- e) 터치된 내용에 맞게 명령 패킷 전송

방향 터치 모드는 아이폰의 터치 센서를 이용하여 모니터 화면에 있는 유닛을 제어한다. 방향 터치 모드는 아이폰을 조이패드처럼 사용할 수 있다. 그림 2는 제안된 다중 스마트폰을 이용한 유닛 제어를 위해 사용된 터치 센서 기반의 인터페이스를 보여준다. 왼손은 좌우 화살표를 누르고, 오른손은 전진과 후진 화살표를 쉽게 누를 수 있도록 방향버튼이 배치되어 있다. 터치가 확인된 화살표는 빨간색 표시 되고 진동을 통해 유저에게 터치가 되었음을 알린다.

2. 통신 패킷 설정

실시간 제어를 위해 손실된 명령어의 값을 예측하는 방법[13]도 있지만 패킷의 크기를 줄여서 서버에서 빠른 데이터 처리를 유도하는 방법도 있다. 각각의 아이폰은 PC에 있는 자신의 유닛을 제어할 때 사용된다. 아이폰과 PC는 무선통신으로 연결된다. 무선 통신을 통해 유닛을 제어할 때 각 아이폰 데이터가 각각의 유닛에 빠르게 반영 되어야 한다. 하지만, 아이폰의 개수가 많아지면 많아질수록 한 대의 PC로 처리해야 할 무선통신 데이터와 계산시간은 점차 많아진다. 따라서, 연결된 아이폰수에 맞는 최소한의 데이터 사용이 요구된다. 본 연구에서는 애드혹 통신을 기반으로 다중 아이폰을 이용하여 하나의 모니터에 있는 각각의 이동로봇을 제어할 수 있는 사례를 다룬다. 다중 사용자가 하나의 시스템에서 무선 제어를 하기 위하여 필요한 최소의 데이터량을 설정하고, 전송된 데이터 손실과 유닛의 반응 시간을 측정한다.

본 연구는 그림 3과 같이 한대의 마스터(master)와 다중 슬레이브(slave)와의 무선 통신을 위하여 애드혹 통신을 이용한다. 아이폰은 모니터 상에서 자신에게 속해있는 한 가지 유닛을 이동시킬 수 있다. 유닛을 이동시키기 위해 아이폰에서 전송되는 명령데이터는 다음과 같이 정의 된다. PC와 연결을 위해 아이폰마다 자신의 ID가 필요하다. 그리고 아이폰에서 전송하는 명령으로 유닛의 전진, 후진, 좌측, 우



그림 3. 통신 방법.
Fig. 3. Communication method.

표 1. 패킷 정의.
Table 1. Definition of packet.

패킷의미	상위4비트	하위4비트				
		좌	우	중립	상	하
아이폰 번호	아이폰 번호	0100	1100	1000	0001	0011
		1100	1000	0011	0011	0011
		1000	1000	0011	0011	0011

즉 이동, 정지명령이 필요하다. 이러한 아이폰 ID, 전송명령의 내용을 1byte의 데이터 패킷으로 표현한다. 표 1은 아이폰에서 전송하는 데이터 패킷을 보여준다. 다수의 아이폰을 이용하기 위하여 상위 4비트는 아이폰의 ID로 설정하고, 하위 4비트는 유닛 이동 명령으로 설정한다. 좌, 우, 상, 하 버튼 어느 것도 선택되어 있지 않으면 자동으로 정지 명령이 전송된다. 정지명령은 하위 4비트가 좌우에서의 중립 1000, 상하에서의 중립 0010이 OR 연산이 되어 1010으로 구성되어 PC로 보내진다.

아이폰에서 PC로 전송되는 패킷은 표 1의 정보를 이용하여 그림 4와 같은 방법을 통해 만들 수 있다. 또한 아이폰의 패킷이 PC에 전송되면 PC에서는 그림 5와 같은 방법으로 패킷의 내용을 알 수 있다. 따라서 아이폰이 PC에 접속 되었을 때 전송되는 기본 명령은 0x1A, 0x2A, 0x3A, ..., 0xFA가 된다.

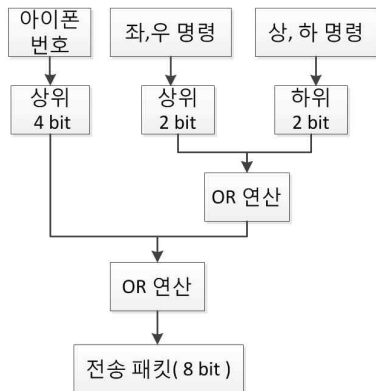


그림 4. 아이폰에서 전송되는 패킷.
Fig. 4. Transferred packet from iPhone.

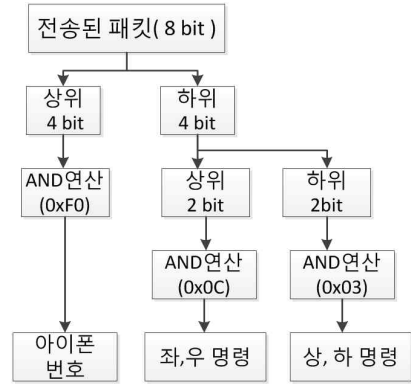


그림 5. PC에 전송되는 패킷 분석.
Fig. 5. Analysed packet on PC.

III. 다중 아이폰과 PC의 통신

다중 아이폰과 PC와의 통신을 위하여 즉, 그림 2와 같이 다중 슬래이브와 한대의 마스터 사이에 애드혹 통신을 위하여 다음과 같은 두 가지 통신방법이 고려 될 수 있다.

1. 요청과 응답에 의한 데이터 처리

N:1의 통신 방법 중 N개에서 전송되는 데이터를 동시에 정확히 처리하기 위해 요청과 응답에 의한 데이터 통신 방법을 고려한다.

그림 6은 요청과 응답에 의한 데이터 처리 방법을 보여준다. 그림 6에서 가로 축은 PC의 인터럽트 발생 횟수를 의미하고, 세로 축은 아이폰의 수를 의미한다. 여기서, 점 표시는 PC에서 아이폰으로 데이터 요청을 의미하고, X 표시는 아이폰에서 PC로 데이터를 전송 하는 것을 의미한다. 따라서 PC에 의한 데이터 요청 후 전송되는 아이폰의 데이터는 정확하게 전송된 시점에 처리된다. 하지만 유닛 하나의 이동을 위해서는 1 스텝이 필요하고 N개의 아이폰에서 전송되는 모든 데이터가 처리되기 위해서는 N+1스텝 만큼의 시간이 소요된다. 또한, 만약 통신을 위해 필요한 아이폰의 수가 많아진다면, 아이폰1에서 아이폰N 까지 순환적으로 통신을 하는 데는 시간이 많이 소요된다. 그림 6의 내용에 대해 표 2는 유닛 반응 시간을 보여준다. 표 2는 각각

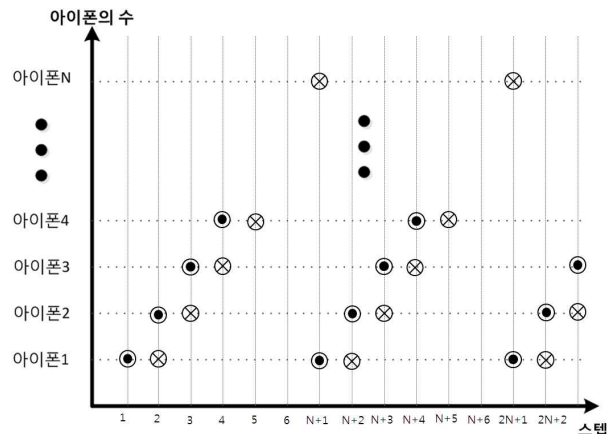


그림 6. 요청과 응답에 의한 데이터 처리.
Fig. 6. Data process by request and reply.

표 2. 그림 6의 유닛 반응 시간.

Table 2. Unit response time of Fig. 6.

아이폰	1대	2대	3대	4대
반응시간	0.5013 [sec]	0.9918 [sec]	1.2063 [sec]	1.9671 [sec]

10번씩 측정 한 후 평균을 낸 값이다. 그림 6에서 요청과 응답의 방법은 한 대의 아이폰에 대해 유닛의 반응속도가 최소 0.5013초가 소요되었다. 두 대의 아이폰에 대해 유닛의 반응속도는 PC와 아이폰1의 통신이 끝나는 시점에 아이폰2의 데이터를 요청하기 때문에 아이폰1의 유닛 반응속도는 0.9918초로 더 느려졌다. 따라서 네 대의 아이폰을 사용하게 되면 더 반응속도가 느려짐을 예측할 수 있고, 표 2의 내용과 같이 실험을 통해서 확인하였다.

더욱 빠른 유닛의 반응시간을 얻기 위해 그림 7과 같은 방법을 사용한다. 그림 7의 방법은 PC에서 다중 아이폰의 데이터를 수신 후 다중 아이폰에게 동시에 데이터를 요청하는 방법이다. 그림 7의 방법은 그림 6의 방법과 유사하지만 응답 데이터를 받은 후 곧 바로 요청하기 때문에 그림 6에서 1스텝부터 1+N스텝까지 걸리는 대기 시간의 손실 없이 각 아이폰의 유닛을 이동시킬 수 있다. 또한, 곧 바로 데이터의 요청과 획득이 가능하기 때문에 PC에 수신된 데이터의 손실이 없다. 하지만, 표 3의 내용과 같이 유닛의 반응 시간은 PC에 접속된 아이폰의 수와 관계없이 약 0.49 초 이상이 소요 되었다. 표 3은 각각 10번씩 측정 한 후 평균을 낸 값이다. 0.49초 이상의 유닛 반응 시간은 유닛을 실시간(real time)으로 자연스럽게 움직이는데 필요한 시간 보다 매우 느려 유닛 제어에 적절하지 않은 시간이다. 따라서, 유닛의 반응속도를 높이고 데이터 획득이 간단한 다른 방법이 필요하다.

2. 데이터를 수신과 동시에 서버에서 처리

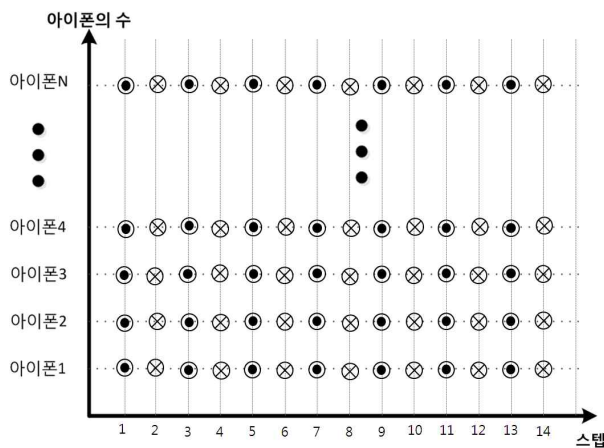


그림 7. 향상된 데이터 전송 방법.
Fig. 7. Improved data transmission method.

표 3. 그림 7의 유닛 반응 시간.

Table 3. Unit response time of Fig. 7.

아이폰	1대	2대	3대	4대
반응시간	0.4992 [sec]	0.5081 [sec]	0.4987 [sec]	0.5469 [sec]

데이터를 수신과 동시에 서버에서 처리하는 방법에서는 유닛의 반응 속도를 높이기 위해 서버는 데이터 수신을 대기(stand-by)만하고, 아이폰에게 데이터 요청은 하지 않는다. 그리고 아이폰은 매 샘플링 시간마다 사용자에게 의해 입력된 데이터를 규칙적으로 전송만 한다.

그림 8은 N개의 아이폰에서 데이터를 전송할 때 시간에 따른 PC의 수신 인터럽트를 보여준다. 여기서 X표시는 아이폰에서 PC로 전송된 데이터를 의미하고, X표시 한 개의 데이터 길이는 1byte를 의미한다. 그리고 N개의 아이폰 데이터 전송 샘플링 시간은 모두 동일하다. N개의 아이폰에서 데이터가 전송되면 PC에서는 데이터 수신에 대해 총괄적인 수신 인터럽트가 발생된다. 그림 8에서 발생한 인터럽트마다 수신된 데이터의 길이를 살펴보면 인터럽트가 발생할 때 마다 데이터 길이는 가변적이다. 예를 들어, 그림 8의 2 스텝에서 아이폰2의 데이터가 비어 있거나, 4 스텝에서 아이폰1의 데이터가 비어있는 것은 PC에서 아이폰으로부터 수신된 데이터를 한 번의 인터럽트에 의해 처리하기 때문이다. 따라서 PC에서 수신된 다중 아이폰들의 데이터들은 길이가 가변적이지만, 수신되는 데이터의 전체 패킷 수는 아이폰에서 전송한 데이터의 전체 패킷 수와 일치하기 때문에 데이터의 손실은 없다.

아이폰 네 대와 PC와의 통신방법을 그림 1과 같은 시스템에 적용시킬 때 유닛이 반응하는 시간은 표 4의 내용과 같다. 표 4는 각각 10번씩 측정 한 후 평균을 낸 값이다. 표 4에서 세로로 데이터를 주기적으로 보낼 때 사용되는 샘플링 시간을 의미한다. 데이터 통신 실험의 결과 한 대의 아이폰 명령에 대한 유닛의 반응시간은 샘플링 시간에 맞게 줄어들지만 접속한 아이폰의 수가 많아지거나 패킷이 전송되는 샘플링 시간이 길어지면, 유닛이 반응 하는 시간이 늦어짐을 알 수 있다. 예를 들어 샘플링 시간이 0.001인 경우 아이폰 한 대와 PC와의 데이터 통신은 가능하지만, 두 번째 아이폰과 PC와의 데이터 통신 시간은 0.5435초가 소요되어 유닛을 움직이는데 적절하지 않다는 것을 알 수 있다.

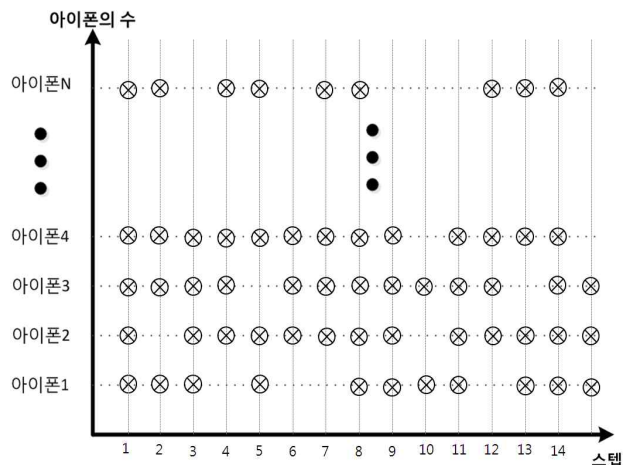


그림 8. 패킷 수신.
Fig. 8. Packet receive.

표 4. 그림 8의 유닛 반응 시간.

Table 4. Unit response time of Fig. 8.

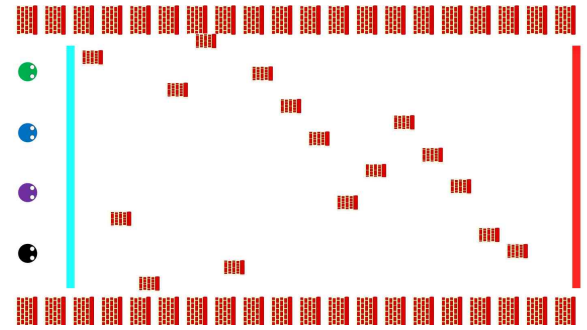
아이폰 샘플링시간	1대	2대	3대	4대
0.001 [data/sec]	0.0989 [sec]	0.5435 [sec]	0.4343 [sec]	0.8315 [sec]
0.005 [data/sec]	0.1262 [sec]	0.2847 [sec]	0.4554 [sec]	0.3397 [sec]
0.01 [data/sec]	0.0923 [sec]	0.1377 [sec]	0.3503 [sec]	0.5658 [sec]
0.05 [data/sec]	0.1675 [sec]	0.2220 [sec]	0.1737 [sec]	0.1587 [sec]
0.1 [data/sec]	0.0994 [sec]	0.1037 [sec]	0.1023 [sec]	0.1008 [sec]
0.5 [data/sec]	0.5027 [sec]	0.5027 [sec]	0.5075 [sec]	0.4992 [sec]
1 [data/sec]	1.0054 [sec]	1.0040 [sec]	0.9964 [sec]	0.9983 [sec]

실험으로 얻어진 표 4의 결과로 아이폰 두 대와 PC와의 데이터 통신, 아이폰 세 대와 PC와의 데이터 통신, 혹은 아이폰 네 대와 PC와의 데이터 통신을 위해서는 최소 샘플링 시간이 0.1초 이상이 되어야 한다. 샘플링 시간보다 유닛의 반응시간이 너무 늦게 되면, 통신할 때 시간지연이 축적되어 아이폰에서 보내는 데이터를 다음 스텝에서 처리할 수 없다. 이것은 아이폰이 최대 두 대, 세 대 혹은 네 대 일 때, 샘플링 시간이 0.05 이하로 설정되면 이러한 문제가 생긴다.

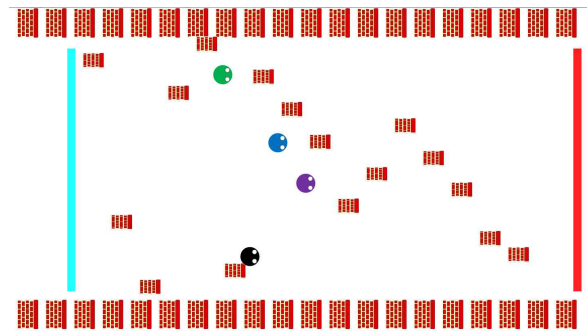
슬레이브와 마스터와의 N:1 통신을 위한 유닛제어는 신속하고 정확하게 이루어져야 한다. 네 대의 유닛 반응 시간에 대해 3.1절의 방법과 3.2절의 방법을 비교하면 3.1절의 방법은 표 3에서 유닛 반응시간이 0.5469초인 반면 3.2절의 방법은 표 4에서 아이폰의 샘플링 시간이 0.1초일 때 유닛 반응시간은 0.1008초였다. 아이폰 네 대가 모두 비슷한 반응시간을 보일 때, 샘플링 시간은 0.1초였다. 이때가 네 대의 유닛이 같은 반응 시간을 보이고, 가장 적은 샘플링 시간이 소요된 경우이다. 따라서 네 대의 아이폰을 사용하는 다음 장의 레이싱 게임 실험에서는 데이터를 수신과 동시에 서버에서 처리하는 3.2절의 방법을 사용하고, 이 때 아이폰의 샘플링 시간은 0.1초로 설정한다.

IV. 레이싱 게임 실험

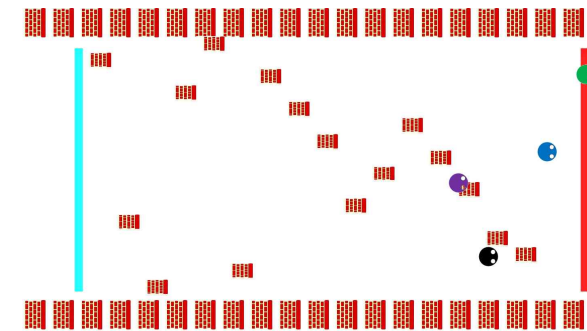
본 장에서는 본문에서 제안된 내용을 효과적으로 보여주기 위하여 PC에서 레이싱 게임 환경 시스템을 설계하여 네 대의 아이폰에 의해 각자의 유닛이 제어되는 것을 보여준다. PC와 애드혹 통신으로 연결된 아이폰의 수는 네 대이고, 하나의 모니터에서 네 개의 각자 유닛을 제어 한다. 실험으로서 레이싱 게임 환경 시스템을 보여주는 이유는 제안된 연구내용을 활용하는데 있어 가장 적합한 사례중에 하나이기 때문이다. 즉, N:1의 통신을 쉽게 적용할 수 있고, 각각의 유닛은 아이폰의 제어명령에 따라 즉각적으로 움직이는 것이 쉽게 보이기 때문이다. 제작된 게임은 Microsoft사의 Windows7 운영체제의 PC에서 개발 되었고 제작 라이브러리는 Microsoft사의 XNA 4.0이다.



(a) Start: 0 [sec].



(b) Meantime: 12.3 [sec].



(c) Finish: 25.2 [sec].

그림 9. 레이싱 게임 실험.

Fig. 9. Experiment of a racing game.

그림 9는 구현된 게임 환경을 보여준다. 네 명의 사용자가 같은 하나의 모니터 화면을 보면서, 왼쪽에 있는 자신의 컬러 유닛을 아이폰 스마트폰을 사용하여 무선 조정한다. 게임은 왼쪽에 있는 자신의 유닛을 오른쪽 결승선으로 먼저 움직이게 하는 사용자가 승리하게 된다.

그림 9에서 사용자가 무선원격제어로 모니터에 있는 자신의 컬러 유닛을 아이폰으로 제어하는 과정은 다음과 같다.

a) 모니터에 있는 현재 자신의 컬러 유닛이 장애물과 상대 컬러 유닛과의 충돌을 피해 결승선으로 빠르게 갈 수 있는 희망루트를 눈으로 확인한다.

b) 로봇이 희망루트 대로 따라갈 수 있도록 아이폰에 있는 터치 버튼을 눌러 조정한다.

c) 아이폰의 터치 버튼에서 읽은 방향 값이 아이폰과 PC와 무선 연결된 애드혹 모듈을 통해서 아이폰의 명령패킷이 PC로 전송된다.

d) PC에 있는 자신의 컬러 유닛이 게임 환경에서 명령방향대로 움직인다.

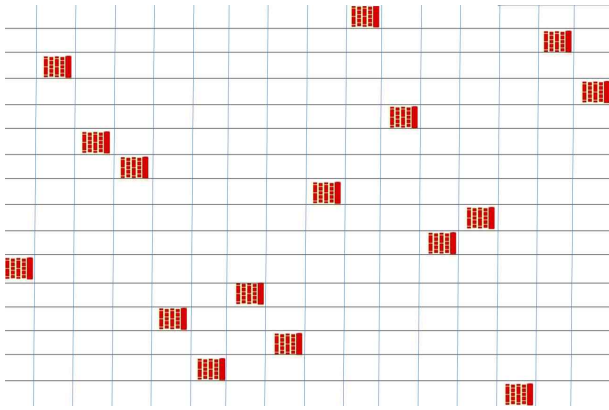


그림 10. 게임에서 장애물 생성.

Fig. 10. Making obstacles on game.

e) 모든 컬러 유닛이 결승선을 통과 할 때까지 a)부터 d)까지의 과정이 계속 반복된다.

게임을 시작하면 유닛은 왼쪽 지점에서 시작하고 오른쪽 지점에 먼저 도착하는 유닛이 승리 하게 된다. 유닛과 벽, 유닛과 유닛의 충돌이 일어나면 유닛의 이동 속력은 절반으로 줄어든다. 게임 시작시 벽들과 같은 장애물을 랜덤하게 추가하여 게임적인 요소를 가미하였다. 그림 10과 같이 벽들 장애물의 생성 위치는 배경 화면을 16X16으로 등분하여 동일 가로 선상에 한 개만 나오게 하였다. 실험에 대한 동영상은 [14]에 있다.

다수의 원격제어기로 게임콘솔(game console)에서 게임을 하기 위해서는 특정 게임콘솔과 원격제어기가 필요하다. 반면에 제안된 연구에서는 게임콘솔이 아니라 PC와 스마트폰을 사용한다는 점에서 다르다. 이것은 PC가 있는 일반 장소에서 다수의 사용자가 자신의 스마트폰을 사용해 하나의 모니터 화면을 보며 동시에 같은 게임을 할 수 있게 해준다.

V. 결론

본 논문에서는 애드혹 통신을 기반으로 다중스마트폰을 사용하여 PC 프로그램의 유닛을 제어하는 연구에 대하여 다루었다. 제안된 다중스마트폰을 사용한 PC 프로그램의 유닛 제어는 최초의 연구 시도이다. 다중스마트폰과 PC와의 통신에서 유닛 제어는 신속하게 이루어져야 하고, 데이터 손실이 없어야 한다. 이러한 사양을 만족 시키는 통신 방법을 찾기 위해서 서버의 요청, 응답 방법과 서버에서 데이터 수신만하는 방법을 다루었다. 서버에서 데이터 수신 시간의 차이는 유닛의 응답 시간과 관련이 있고, 시간 측정을 통해 적절한 샘플링 시간을 찾을 수 있었다. 개발된 환경은 다중 스마트폰을 이용하여 한 대의 모니터에서 PC 게임을 수행하거나 다중 스마트폰으로 한 대의 시스템에 있는 유닛 제어를 가능하게 하였다. 실험에서는 네 대의 아이폰을 PC에서 각각의 유닛을 조정하는 원격제어기로 사용하여 제안된 방법이 효과적으로 사용될 수 있음을 보여주었다.

REFERENCES

- [1] Y. Li and L. Zeng, "Research and application of the EAN-13 barcode recognition on iphone," *International Conference on Future Information Technology and Management Engineering (FITME)*, vol. 2, pp. 92-95, Oct. 2010.
- [2] S. Chemlal, S. Colberg, M. Satin-Smith, E. Gyuricsko, T. Hubbard, M. W. Scerbo, and F. D. McKenzie, "Blood glucose individualized prediction for type 2 diabetes using iPhone application," *2011 IEEE 37th Annual Northeast Bioengineering Conference*, pp. 1-2, Apr. 2011.
- [3] H. K. Y. Chan, Z. Huiru, W. Haiying, R. Gawley, Y. Mingjing, and R. Sterritt, "Feasibility study on iPhone accelerometer for gait detection," *2011 5th International Conference on Pervasive Computing Technologies for Healthcare*, pp. 184-187, May 2011.
- [4] C. Nam, S. Gim, S. Lee, and D. Shin, "Sensor network system to operate multiple autonomous transport platform," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 18, no. 8, pp. 706-712, Aug. 2012.
- [5] B. R. Geltz, J. A. Berlier, and J. M. McCollum, "Using the iPhone and iPod touch for remote sensor control and data acquisition," *Proc. of the IEEE SoutheastCon 2010*, pp. 9-12, Mar. 2010.
- [6] D. H. Kim, "Development of a Simulator for a Mobile Robot Based on iPhone," *Journal of Korean Institute of Intelligent System (in Korean)*, vol. 23, no. 1, pp. 29-34, 2013.
- [7] H. Jung and D. H. Kim, "Study of iPhone Interface for Remote Robot Control Based on WiFi Communication," *Journal of Korean Institute of Intelligent System (in Korean)*, vol. 22, no. 5, pp. 669-674, 2012.
- [8] <http://www.youtube.com/watch?v=KaR-PHZVWMM>
- [9] <http://www.youtube.com/watch?v=E7RGBv5qjaI>
- [10] <http://www.youtube.com/watch?v=6DqeY2RLsQk>
- [11] B. R. Geltz, J. A. Berlier, and J. M. McCollum, "Using the iPhone and iPod Touch for remote sensor control and data acquisition," *Proc. of the IEEE SoutheastCon 2010*, pp. 9-12, 2010.
- [12] H. Jung, Y. Kim, and D. H. Kim, "Communication quality analysis for remote control of a differential drive robot based on iPhone interface," *Communication in Computer and Information Science*, vol. 206, pp. 278-285, Sep. 2011.
- [13] R. H. Choi, S. C. Lee, and J. Yoo, "Robust real-time wireless control platform compensating for packet loss," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 18, no. 8, pp. 768-773, Aug. 2012.
- [14] <http://www.kyungnam.ac.kr/~dhkim/>PROJECTS>VideoClips>



정 하 민

2009년 경남대 전기전자 공학부 졸업.
2011년 동 대학원 석사. 2011년~현재
경남대학교 첨단공학과 박사 과정. 관
심분야는 스웩 시스템, 지능제어&로봇,
디지털 신호처리, 영상처리.



김 동 헌

2001년 한양대학교 전기공학과 공학
박사. 2001년~2003년 미국 듀크 대학
교 연구원. 2003년~2004년 미국 보스
톤 대학교 연구원. 2004년~2005년 일
본 동경대학교 박사후 과정. 2012
년~2013년 영국 에식스대학교 방문교
수. 2005년~현재 경남대학교 전기공학과 교수. 관심분야는
스웩 시스템, 이동로봇 경로계획, 다개체 로봇, 스마트폰 응
용 로봇, 지능로봇 제어, 비선형 적응제어.