

## 주행용 로봇 플랫폼을 위한 임베디드 프로세서 기반 원격영상감시 시스템 구현

(Implementation of Remote Image Surveillance for Mobile Robot Platform based on Embedded Processor)

한경호\* · 윤호원

(Kyong-Ho Han · Hyo-Won Yun)

### 요 약

본 논문에서는 무선랜 환경에서 주행 로봇 플랫폼에 의한 원격 영상 감시 시스템을 제안하였다. 제안된 방법은 ARM9코어기반의 PXA255 프로세서 시스템에 소형 CMOS 카메라를 장착하고 촬영한 영상을 803.11b/g 무선 네트워크를 이용하여 전송하는 이동형 원격 영상 감시 시스템을 구성하였다. 로봇플랫폼의 주행 명령을 원격으로 전송하며, 촬영된 영상은 640×480, 320×240픽셀 등의 정지 영상을 초당 3~10프레임의 속도로 전송하였으며, 시스템을 구현하고 영상에 의한 원격감시기능을 적절히 수행할 수 있음을 실험을 통하여 확인하였다. 제안된 시스템은 리눅스 운영체제를 기반으로 구축하였다.

### Abstract

In this paper, we proposed the remote visual monitoring system on mobile robot platform. The proposed system is composed of ARM9 core PXA255 processor, micro CMOS camera and wireless network and the captured visual image is transmitted via 803.11b/g wireless LAN(WLAN) for remote visual monitoring operations. Robot platform maneuvering command is transmitted via WLAN from host and the 640×480, 320×240 pixel fixed visual image is transmitted to host at the rate of 3~10 frames per second. Experimental system is implemented on Linux OS base and tested for remote visual monitoring operation and verified the proposed objects.

Key Words : Visual Surveillance, Robot Platform, Wireless LAN, ARM9, PXA255

## 1. 서 론

최근 다양한 분야에서 다양한 형태의 로봇들이 적용되고 있다. 특히 경비 및 보안, 산업현장 등의 위험한 상황에서 인간을 대신하여 정보를 수집하여 제공하는 기능을 갖는 보안용 로봇에 대한 관심이

\* 주저자 : 단국대학교 전자전기공학전공 교수  
Tel : 031-8005-3608, Fax : 031-8005-4096  
E-mail : kyonghan@dgu.edu  
접수일자 : 2008년 10월 22일  
1차심사 : 2008년 10월 23일  
심사완료 : 2008년 11월 7일

## 주행용 로봇 플랫폼을 위한 임베디드 프로세서 기반 원격영상감시 시스템 구현

가장 많이 늘어나고 있다. 이러한 로봇을 이용한 감시 시스템의 핵심 기술은 원격으로 로봇을 이동시키고 로봇에 장착된 카메라를 통하여 영상정보를 수집하는 것이며 무선 인터넷이 급속도로 발전되면서 인터넷을 통한 로봇의 원격 감시 및 제어가 주류를 이루고 있다[1-3]. 무선 인터넷은 무선 랜(WLAN: Wireless LAN)기술이 발전함에 따라 803.11a/b/g/n 등의 규격에서 전송속도가 11[M]/54[M]/100[Mbps] 등으로 발전하여 무선 랜을 통하여 영상정보를 전송할 수 있는 대역폭이 확장되고 있다[4-5]. 본 논문에서는 무선 랜 환경에서 주행 로봇 플랫폼을 위한 원격 영상 감시 시스템을 구현하기 위하여 주행 로봇에 장착된 카메라를 이용하여 정지영상을 촬영하고 이를 무선 랜을 통하여 원격의 호스트에 전송하고, 호스트에서는 무선 랜을 통하여 로봇의 주행을 원격으로 제어하도록 하는 플랫폼을 임베디드 프로세서와 리눅스 운영체제를 이용하여 구성하였다. 제안된 내용은 시스템을 구현하고 실험으로 확인하였다.

## 2. 영상 원격 감시 로봇 플랫폼 구성

그림 1은 본 논문에서 제안하는 로봇 플랫폼의 구성도이다. 구성도에서 메인보드는 임베디드 프로세서를 중심으로 플래시메모리, SRAM 등으로 구성되어 있으며, 운영체제, 영상정보 수집 기능, 네트워크 접속 기능 그리고 모터 제어 기능을 수행한다. 모터는 로봇 플랫폼을 이동시키며 모터의 전후 및 속도 제어회로에 의하여 구동된다. CMOS 카메라는 정지영상을 수집하며 메인 보드의 카메라 제어 신호에 의하여 영상이 촬영되고 무선 랜으로 전송된다. 무선 랜 부분은 802.11x 무선 프로토콜을 지원하는 무선 랜 카드이며 AP(Access Point)에 의하여 AdHoc 방식 또는 호스트와 직접 연결되는 Peer-to-Peer 방식을 사용할 수 있다[6-7].

영상정보는 정지영상 또는 동영상 포맷으로 수집할 수 있으며 수집된 영상정보는 전송을 위하여 부호화 되며 동영상의 경우 정지영상보다 부호화 과정의 복잡도가 매우 높으며 화소 수, 초당 전송 프레임 수, 화소 당 컬러비트수 등에 따라 더 빠른 처리속도

의 프로세서가 필요하게 된다. 로봇에 탑재되는 프로세서는 전원소비 및 크기 등의 제약으로 모바일용 저 전력 임베디드 프로세서가 사용되므로 일반 데스크탑 프로세서보다 처리능력이 비교적 느리므로 수집된 영상정보에 대하여 동영상보다 정지영상으로 전송하는 것이 더 적합하다.

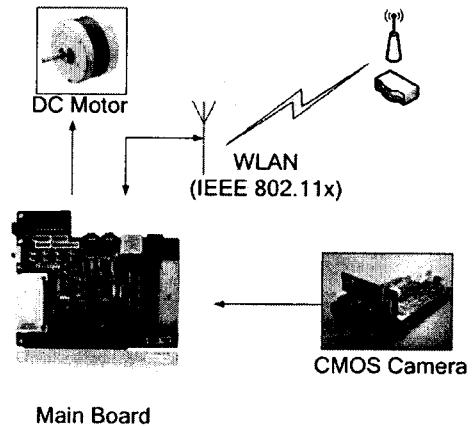


그림 1. 로봇 플랫폼 구성  
Fig. 1. Robot Platform Configuration

메인 보드는 HyBus 사의 X-Hyper255B를 사용하였으며 사용된 프로세서는 ARM9 코어를 가진 Intel사의 PXA255(XScale)로 32[bit] 400[MHz]의 클럭으로 동작한다. BootLoader, 운영체제 및 로봇 플랫폼 응용 프로그램은 메인 보드의 플래시 메모리에 저장되며, 원격 영상 감시 프로그램은 유·무선 인터넷을 통하여 PC에서 플래시 메모리로 저장할 수 있도록 하였다. 로봇 플랫폼을 움직이기 위한 구동부의 모터는 MAXON 사의 DC 모터를 사용하였고 모터 구동을 위하여 일반적인 MOSFET 소자의 H-Bridge 회로를 이용하여 플랫폼을 전후, 좌우로 이동하도록 하였다. 모터구동을 위한 신호로 ARM 프로세서의 병렬포트(GPIO)를 이용하였다. 무선 랜은 802.11a/b/g/n 등의 다양한 규격에 따라 각각 11[M]/54[M]/100[Mbps] 등의 최대 전송 대역폭을 갖는다. 무선 랜을 통하여 전송 되는 데이터는 로봇에서 PC로 전송되는 영상정보와 PC에서 로봇으로 전송되는 구동 제어 신호이며 구동 제어 신호는 전송 대역폭의 극히 적게 차지하므로 PC로 전송되는

영상정보의 전송대역폭에 따라 사용할 무선 랜의 규격을 정한다.

### 3. 소프트웨어 구성

본 논문에서 응용 소프트웨어는 ARM과 PC 두 가지 프로세서용으로 구현하였다. ARM 프로세서용 소프트웨어는 로봇 플랫폼에 탑재되며 영상정보를 수집하여 무선 랜으로 전송하고, PC로부터 전송되는 제어 신호를 받아 모터를 구동한다. PC용 소프트웨어는 원격 영상을 받아 화면에 보이고 로봇의 주행을 제어하기 위한 사용자 인터페이스 소프트웨어이다. ARM 프로세서의 응용 소프트웨어는 리눅스 기반에 C로 구현하였으며 PC의 응용 소프트웨어는 Window 기반에 Visual C++로 구현하였다[8]. 로봇 플랫폼에는 응용 소프트웨어 외에 시스템 소프트웨어로 부트로더와 운영체제가 실장되며 CMOS 카메라, GPIO 병렬포트, 무선랜카드, USB2.0 호스트 등의 하드웨어에 대한 디바이스 드라이버가 운영체제에 추가된다. 디바이스 드라이버는 리눅스 커널과 별도로 모듈로 만들고 시스템 부팅시에 하드웨어가

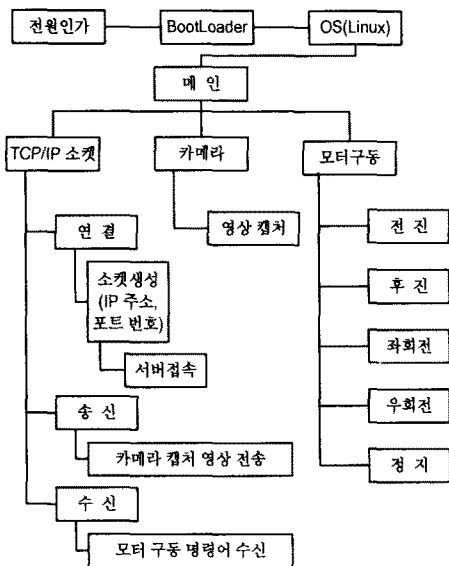


그림 2. 로봇 플랫폼 응용 소프트웨어  
Fig. 2. Robot Platform Application Software

검지될 때에 해당 디바이스 드라이버를 커널에 추가 되는 방법을 사용하여 리눅스 커널이 메모리를 차지 하는 용량이 작도록 하였다[9-11]. PC와 로봇이 무선 랜을 통하여 접속하고 데이터를 송수신하기 위한 소켓(Socket) 프로그램을 구현하여 로봇의 프로세서에 PC의 IP 주소 및 포트번호로 생성된 소켓을 통하여 모터 구동 신호와 영상 데이터를 전송한다[12].

로봇 플랫폼의 응용 프로그램의 순서도는 그림 2와 같다[13-14].

PC의 사용자 인터페이스를 위한 응용 소프트웨어는 로봇과 무선 랜으로 연결하기 위한 소켓 프로그램, 로봇 이동을 제어하는 명령어 입력기능, 로봇에서 전송된 정지영상을 화면에 표시하는 기능으로 구성되어 있으며 그림 3에서 소프트웨어의 구성을 보인다.

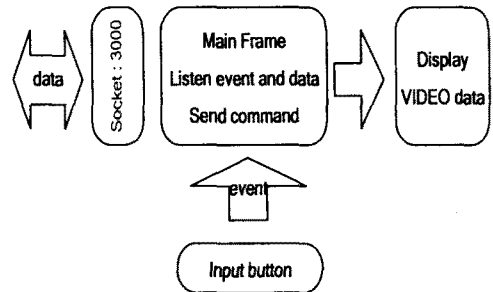


그림 3. PC 사용자 인터페이스 소프트웨어  
Fig. 3. PC User Interface Software

### 4. 원격 영상정보 처리

영상 정보는 카메라에서 촬영되고 패킷형태로 구성되어 무선 랜을 통하여 전송되며 PC에서는 이를 프레임 단위로 구분하여 화면에 표시한다. 영상에 의한 원격 감시를 위하여 초당 3~10프레임 정도의 정지영상으로 기능을 충분히 수행할 수 있고, 무선 랜 전송 속도가 10[Mbps]이상임을 고려할 때 영상 감시에 충분한 화질의 영상 전송이 가능하다. 예를 들어 영상 프레임이 320×240 화소에 한 화소당 16비트 컬러 즉, 65,536 가지의 색을 표현하는 경우에 320×240×16=1.2288[Mbps]의 데이터 대역을 사용하며 16비트의 컬러 비트를 예를 들면, R(적색) 5비트, G(녹색) 6비트, B(청색) 5비트를 할당하여 구성되

## 주행용 로봇 플랫폼을 위한 임베디드 프로세서 기반 원격영상감시 시스템 구현

는 한 프레임의 정지영상의 데이터 포맷은 그림 4와 같다.

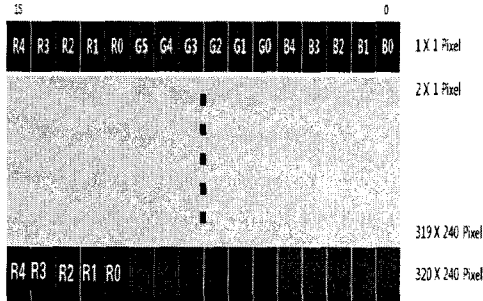


그림 4. 영상 데이터 포맷 예  
Fig. 4. Image Data Format Example

### 5. 실험 및 결과

ARM 보드를 이용하여 로봇 플랫폼을 그림 5와 같이 제작하고 실험하였다.

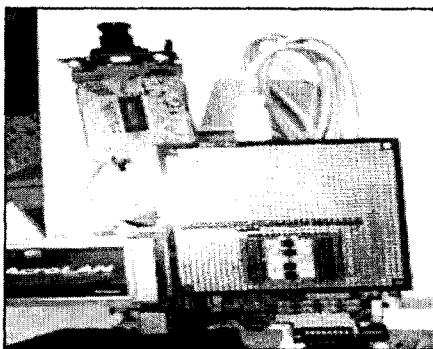


그림 5 로봇 플랫폼 실험장치  
Fig. 5. Experimental Robot Platform System

카메라는 OmniEye 사의 USB 2.0 방식의 CMOS 카메라를 사용하였으며 30만 화소의 CMOS 이미지 센서로 정지 영상을 최대 640×480 화소로 초당 최대 30프레임(Frame)의 속도로 촬영할 수 있다.

USB 카메라와 AcroLAN 사의 802.11g 무선 랜 모뎀을 메인보드에 연결하였다. 802.11g 무선 랜의 최대 전송 대역폭이 54[Mbps]임을 고려하여 이를 초과하지 않는 범위에서 화소 수는 320×240 또는 160×120를 사용하고 초당 전송 프레임 수 및 컬러비

트를 표 1과 같이 몇 가지 경우를 사용하여 실험하였으며 모두 정상적인 영상정보를 받을 수 있었다. 응용 분야에 따라 초당 전송 프레임수를 늘이거나, 화소 당 컬러해상도를 높여야 하는 경우 화소 수를 160×120로 낮추어 데이터 전송속도 및 프로세서의 처리 속도 범위 내에서 전송하였다. 화소 당 컬러비트는 9 비트, 12 비트, 24 비트의 경우 R, G, B 각각을 3 비트, 4 비트, 8 비트를 사용하였고, 16 비트의 경우 R, G, B를 각각 5 비트, 6 비트, 5 비트를 사용하였다. 영상정보 전송 대역폭에서 중요한 것은 영상 파일 형식의 전체 비트 수이며, 화소의 컬러비트 비율은 임의로 정하였다.

표 1 실험영상파일형식  
Table 1. Experimental Image File Formats

| 픽셀 수    | 영상 파일 형식 |         | 전송속도<br>([Mbps]) |
|---------|----------|---------|------------------|
|         | 초당프레임    | 컬러비트수   |                  |
| 320×240 | 3[fps]   | 9[bit]  | 2.07             |
| 320×240 | 3[fps]   | 12[bit] | 2.76             |
| 320×240 | 3[fps]   | 16[bit] | 3.69             |
| 160×120 | 10[fps]  | 24[bit] | 4.61             |
| 320×240 | 5[fps]   | 24[bit] | 9.22             |
| 320×240 | 3[fps]   | 24[bit] | 5.53             |

로봇 플랫폼을 이동시키는 모터 구동부를 그림 6과 같이 구성하였다. M-L과 M-R은 각각 좌측 구동 모터와 우측 구동모터이며, LF와 LB는 각각 좌측 모터의 전진과 후진 제어 신호를 의미하며 GPIO 핀 6에서 LF 신호를 발생시키고 이를 반전하여 LB 신호를 발생한다. LF 신호와 LB 신호에 따라 Q1과 Q2, Q3과 Q4가 ON/OFF를 서로 반대로 작용하며 좌측 구동 모터회전을 정 방향 또는 역 방향으로 회전시킨다.

RF와 RB는 각각 우측 모터의 전진과 후진 제어 신호를 의미하며 GPIO 핀 7에서 RF 신호를 발생시키고 이를 반전하여 RB 신호를 발생한다. RF 신호와 RB 신호에 따라 Q5와 Q6, Q7과 Q8이 서로 반대로 ON/OFF되어 우측 구동 모터회전을 정방향 또는

역방향으로 회전시킨다. 좌회전 및 우회전은 좌측 모터 와 우측 모터의 회전방향을 서로 반대로 구동한다. GPIO 핀 5 출력을 사용하는 EN 신호는 Q0를 ON/OFF하며 모터 구동회로 전체에 공급하는 전원을 제어하여 로봇의 주행을 허용 또는 정지하도록 제어한다. 표 2에서 모터제어용 GPIO신호를 보인다.

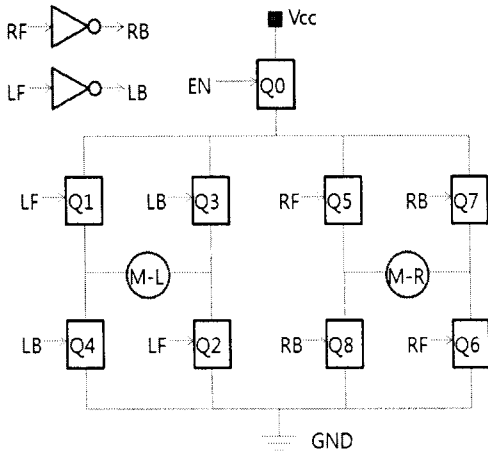


그림 6. 모터 구동부 구성  
Fig. 6. Motor Drive Configuration

표 2. 모터제어 GPIO 신호구성  
Table 2. Motor Drive Signal GPIO Configuration

| 모터제어용 GPIO 핀(출력) |                               |                               |                               |
|------------------|-------------------------------|-------------------------------|-------------------------------|
| 방향제어             | GPIO5<br>(구동회로<br>전체제어신호: EN) | GPIO6<br>(좌측모터구동제어<br>신호: LF) | GPIO7<br>(우측모터구동제어<br>신호: RF) |
| 전진               | 1                             | 1                             | 1                             |
| 후진               | 1                             | 0                             | 0                             |
| 좌회전              | 1                             | 1                             | 0                             |
| 우회전              | 1                             | 0                             | 1                             |
| 정지               | 0                             | X                             | X                             |

그림 7은 PC에서 수행되는 사용자 인터페이스 화면으로, 로봇의 주행을 제어하고, 로봇에서 전송된 영상을 표시한다. 무선 랜의 접속 및 차단은 “disconnect” 버튼으로 제어하고, “TV OFF/ON” 버튼은 영상을 표시할 것인지 아닌지를 제어한다. 현

재는 영상이 표시되고 있으므로 “TV off” 버튼으로 표시되며, 이 버튼을 선택하면 영상이 표시되지 않고 “TV on” 버튼으로 표시되어 이 버튼을 선택하면 다시 영상이 표시되도록 하였다. 화면 하단의 전, 후, 좌, 우 화살표 표시는 로봇의 진행방향을 제어하며, 중앙의 사각형 버튼은 로봇을 주행 또는 정지 기능을 제어한다.

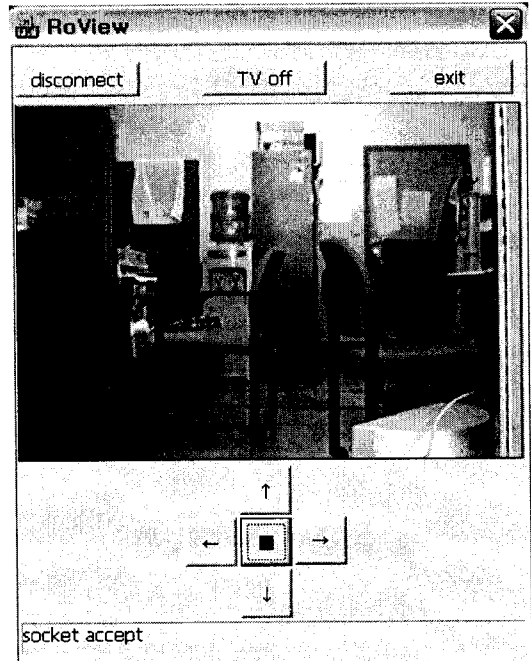


그림 7. 사용자 인터페이스 실행 화면  
Fig. 7. User Interface Operation Display

본 실험에서는 최대 54[Mbps]의 대역폭을 갖는 802.11g 무선 랜 카드를 사용하였다. 그림 8에서 PC 와 플랫폼을 무선 랜으로 접속한 상태에서 320×240의 화소수와 화소 당 9비트 컬러비트 영상을 초당 3 프레임씩 전송하는 경우에 영상 데이터를 계속 전송 하면서 로봇의 주행을 제어하는 동안 무선 데이터 접속 상태를 보이고 있다. 무선 랜의 실제 최대 연결 속도는 48[Mbps]로 나타내고 있으며, 3~11[%]의 네트워크 이용률을 보이고 있어 무선 랜 데이터 트래픽이 실험 영상데이터를 충분히 전송하고 있음을 알 수 있었다.

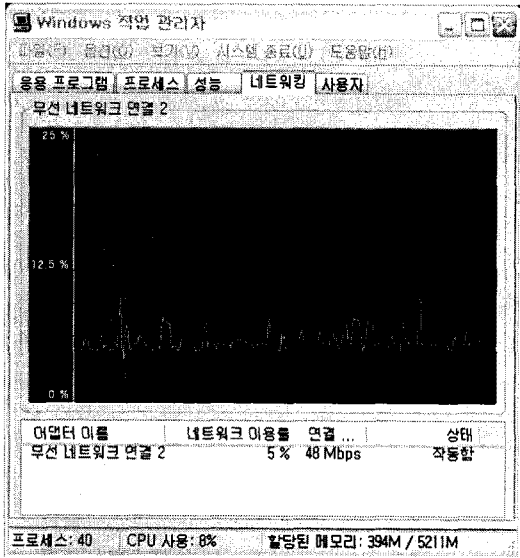


그림 8. 무선랜 데이터 트래픽  
Fig. 8. WLAN Data Traffic

실험에서는 초당 3프레임씩 전송할 때, 1/3초 마다 영상정보를 수집할 수 있으므로 일반적인 영상감시 분야에 응용함에 있어서 문제가 없었다. 응용분야에 따라 단위시간(초)당 전송 프레임 수를 더 늘여야 하는 경우와 영상 프레임 단위당 화소 수가 더 많이 필요한 경우에는 단위시간에 전송되는 영상 프레임의 데이터의 양이 증가하므로 무선 랜의 전송 대역폭을 늘여야 한다. 본 실험에서는 802.11g 무선 랜과 USB 2.0 포트를 사용하여 이론상 최대 54[Mbps]의 전송 대역폭을 허용하나, 실제 전송되는 속도는 이보다 약간 낮은 48[Mbps]이지만 일반적인 동영상 전송하는데 무리가 없었다. 그러나 그 이상의 전송속도가 필요한 경우에는 보다 넓은 전송 대역폭 100[Mbps] 이상을 지원하는 802.11n 등의 무선 랜을 사용하여야 한다. 또한 무선 랜이 연결되는 포트, 예를 들면 USB 2.0의 경우 최대 전송 속도가 480[Mbps]임을 역시 고려하여야 한다.

사용한 프로세서는 320×240 화소의 일반적인 동영상 처리를 충분히 처리할 수 있었다. 초당 프레임 수 및 프레임 당 화소 수를 늘이는데 고려할 또 다른 요소는 ARM 프로세서 등의 모바일용 임베디드 프로세서는 저 전력 등의 동작 여건으로 인하여 일반 PC용 프로세서에 비하여 연산처리 속도가 다소 낮다는

것이다. 카메라에 수집된 영상은 로봇 플랫폼의 프로세서로 소프트웨어에 의하여 한 프레임의 정지영상을 처리하고 전송한 다음에 다음 프레임 영상을 처리하기 때문에 카메라가 수집할 수 있는 최대 속도로 수집되는 영상을 모두 처리하여 전송할 수 없다. 이 경우에는 영상 처리 알고리즘의 개선 또는 고성능의 프로세서를 사용하면 개선할 수 있다.

## 6. 결 론

본 논문에서는 ARM9 코어를 내장한 PXA255 임베디드 프로세서에 리눅스를 기반으로 무선 랜을 이용하여 주행 로봇 플랫폼을 위한 원격 영상 감시 시스템을 구현하고 실험에 의하여 기능과 성능을 확인하였다. 실험용 하드웨어와 소프트웨어를 제작하고 플랫폼과 PC가 802.11g 무선 랜에 의하여 영상 정보가 플랫폼에서 PC로 전송되어 PC에서 화면으로 표시되고 PC에서 플랫폼으로 이동제어 명령을 전송하여 플랫폼의 이동을 제어하는 결과를 실험으로 확인하였다. 본 연구 결과, 320×240, 160×120 화소 수에 초당 3~10 프레임을 전송하여 모바일용 임베디드 프로세서를 사용하여도 영상 감시 기능을 충분히 구현할 수 있음을 확인하였다. 본 연구의 결과는 감시 로봇의 가장 중요한 정보 수집 기능인 영상정보를 원격에서 무선 랜으로 받아 표시하고 로봇의 이동을 제어하는 분야에 활용될 수 있으며 802.11a/n, 802.16e 등의 무선 네트워크를 추후 적용하면 거리 및 전송속도가 증가되고 고성능 임베디드 프로세서를 사용하면 본 시스템의 성능을 용이하게 개선할 수 있다[15].

본 연구는 2006학년도 단국대학교 대학연구비 지원으로 수행되었음.

## References

- [1] A. S. Sekmen, Z. Bingul, V. Hombal, and S. Zein-Sabatto, "Huma-robot interaction over the internet", Proc. of the IEEE on southeastcon, pp. 224-228, 2000.
- [2] R. C. Luo and T. M. Chen, "Remote Supervisory Control of A sensor based mobile robot via internet", Proc. of

- the IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems, vol. 2, pp. 1163-1168, Sep. 1997.
- [3] R. L. Alves, R. F. Vassallo, E. O. Freire, and T. F. Bastos-Filho, "Teleoperation of a mobile robot through the internet", Proc. of the 43rd IEEE Midwest Symposium on Circuits and Systems, vol. 2, pp. 930-933, 2000.
  - [4] 김종선, 유영선, 김성호, 주영훈, "이동 로봇의 원격 주행 시스템", ICS'07 정보 및 제어 심포지움 논문집, pp. 325-327, 2007.
  - [5] 김상현, 정재영, 이동명, "Embedded Linux 기반의 로봇 설계 & 제작", (주)영진닷컴, pp. 27, 2004. 7.
  - [6] 윤성우, "TCP/IP 소켓 프로그래밍", FREELEC, pp. 21-24, 2003. 4.
  - [7] Jon C. Snader, "EFFECTIVE TCP/IP PROGRAMMING", pp. 45-53, July 2003.
  - [8] 박원성, 정기철, "ARM-9을 이용한 임베디드 리눅스 시스템" pp. 260-265, 2005. 12.
  - [9] 유영창, "리눅스 디바이스 드라이버", 한빛 미디어, pp. 62-64, 2004. 12.
  - [10] (주)한백전자, "임베디드 리눅스 시스템을 이용한 디바이스 제어 및 통신의 이해" pp. 17-18, 2005.
  - [11] ALESSANDRO RUBINI, "LINUX DEVICE DRIVERS", O'REILLY, pp. 28-29, Feb. 2000.
  - [12] 김희수, "무선랜 기반의 ARM 프로세서를 이용한 PMP 구현", 단국대학교 대학원 석사학위논문, pp. 5-9, Feb. 2007.
  - [13] Andrew N. Sloss, Dominic Symes, Chris Wright, "ARM System Developer's Guide", SciTech, pp. 2-17, Feb. 2005.
  - [14] Eric Harlow, "Developing Linux Applications with GTK+ and GDK", pp. 28-31, Feb. 1999.
  - [15] 김현동, 최상호, "802.16e 기반 시스템을 위한 시간동기화 방법에 관한 연구", 2005년도 정보 및 제어 학술대회(CICS 05) 논문집, pp456, 2005.

## ◇ 저자소개 ◇

### 한경호 (韓敬浩)

1959년 6월 25일생. 1982년 서울대학교 공과대학 전자공학과 졸업. 1984년 서울대학교 대학원 전자공학과 졸업(석사). 1992년 미국 Texas A&M University, College Station 졸업(EE PhD). 1984~1985년 삼성휴렛팩커드 연구원. 1985~1987년 한국통신 전임연구원. 1989~1992년 Texas A&M University, Unix System Administrator & Network Analyst. 1992~1993년 한국전자통신연구원 이동통신연구단 CDMA 개발 선임연구원. 1993~1995년 단국대학교 전기공학과 전임강사. 1995~1999년 단국대학교 전기공학전공 조교수. 1999~2005년 단국대학교 전기전자컴퓨터공학 전공 부교수. 2005년~현재 단국대학교 전자전기공학전공 교수. 2008년~현재 단국대학교 정보통신원장.  
관심분야 : 마이크로프로세서응용시스템, 임베디드 시스템, 지능형 로봇시스템, 지능형교통시스템(ITS), 위치기반시스템(LBS).

### 윤효원 (尹孝遠)

1981년 4월 14일생. 2005년 단국대학교 공학부 전기전자컴퓨터공학 전공 졸업. 2007년 단국대학교 대학원 전기공학과 졸업.