ARM 9 임베디드 시스템에 의한 무선 제어 및 영상 감시 로봇 구현

Implementation of Wireless Control and Image Monitoring Robot using ARM 9 Embedded System

윤효원*, 한경호**
Hyo-Won Yun, Kyong-Ho Han

Abstract - This paper is dealing with how to control of a client robot's movement for instructions from a server PC and a wireless and remote control Robot that sends the server information of images for monitoring. To implement this, 802.11x WLAN with TCP/IP socket programming is used to get the driving instructions from the server PC and control movements of the robot such as a forward, backward and directions. As well as this, ARM9 cored PAX255 embedded processor and Linux OS is used for the function transmitting BMP format of 320 × 240 pixel for stopped image data.

Key Words : Embedded system, Wireless LAN, ARM processor, Image Monitoring, Linux kernel

1. 서론

로봇 분야는 최근 급속하게 발전하고 많은 분야에서 응용되고 있다. 실생활에 필요한 청소 로봇부터 군사용 로봇에 이르기까지 로봇 시장은 점점 더 커지고 있다. 그 중 고부가가치에 해당하는 휴먼 로봇, 군사용 로봇, 보안용 로봇 등은 활발하게 연구 개발되고 있는 분야이다. 휴먼 로봇은 인간의 형상을 닮은 2족 2수에 인간의 행동과 유사한 행동을 하고, 군사용 로봇은 폭탄 제거, 생화학 물질 등에 대한 오염원 판단과 농도를 측정하는 등 각종 군사작전이나 화학전에 인간의 임무를 대신 수행한다. 보안용 로봇은 국제회의, 혹은 국제 스포츠 행사 등 어떤 중요한 행사에서의 보안은 물론 각종 산업 현장이나 공장 등의 경비를 맡아 위험한 상황에서 인간을 대신하여 보다 빠르고, 정확하게 중요한 정보를 제공함으로써 인명과 재산을 보호한다. 이 로봇들은 고가 및 대형일 뿐 만 아니라 고전력을 요한다.

본 논문에서는 고부가가치의 로봇 중 보안용 로봇에 착안하여 이를 구현하고자 한다. 이 로봇의 주목적 중 하나가 감시이므로 로봇에는 카메라를 설치하고, WLAN을 이용하여 서버에서 영상을 모니터링하는 시스템을 구현한다.

2. 임베디드 시스템

임베디드 시스템이란 미리 정해진 특정 기능을 수행하기 위해 하드웨어와 소프트웨어가 내장된 전자 제어 시스템을 의미한다. 즉, 단순 회로만으로 구성된 장치가 아닌 마이크로프로세서가 내장되어 있으며, 이러한 마이크로프로세서를 운용하여 제한된 기능을 수행 및 관리하는 프로그램이 포함된 시스템을 임베디드 시스템이라고 한다.

대부분의 임베디드 시스템이 몇 년 전만 하더라도 운영체제 없이 개발되었지만, 한 조사에 의하면 60% 정도의 임베디드 시스템이 운영체제를 포함하고 있는 것으로 밝혀져 점점 더 많은 임베디드 시스템이 운영체제를 가지고 개발되고 있다는 것을 알 수 있다. 임베디드 시스템에서 운영체제를 점점 더 많이 도입하는 중요한 이유는 좀 더 좋은 성능을 가진 복잡한 하드웨어가 사용되고 예전과는 비교할 수 없는 많은 소프트웨어 기능이 요구되기 때문이다.

임베디드 시스템에서 사용하는 운영체제는 용도에 따라 크게 두 가지 분류로 나눈다. 하나는 범용 운영체제인 리눅스나 마이크로소프트 윈도우와 같이 서버나 PC에서 실행되는 응용프로그램들을 지원하고, 다른 하나는 실시간 운영체제(RTOS)와 같이 우주, 항공, 군사, 의료 장비 등과 같은 임베디드 시스템에서 엄격한 시간 제약성을 만족시켜야 하는 환경에서 사용된다. 초기에는 실시간 운영체제가 주로 이용되어 개발되었으나, 리눅스의 성능과 안정성이 나아지고 다양한 프로세서를 지원하면서 리눅스를 활용한 응용 사례가 늘어나고 있다. 또한 리눅스는 오픈 소스로 인하여 각종 주변 장치 드라이버 개발도 용이하다. 이에 본 논문의 시스템 역시 리눅스를 기반으로 개발하였다.[1]

3. 시스템 구성

3.1 전체 시스템 구성

그림 1은 무선 제어 및 영상 감시 로봇의 전체 시스템 개념도이다 서버와 로봇 각각에 WLAN을 설치하여 데이터를 송수신하게 된다. 본 논문에서 구현된 시스템은 서버로부터 받은 명령대로 로봇이 움직이면서 설치된 CMOS 카메라의

저자 소개
* 尹孝遠: 檀國大學 電氣工學科 碩士課程
** 韓敬浩: 檀國大學 電氣工學科 敎授

영상을 전송하여 서버에서 사용자가 감시를 한다. WLAN을 이용한 네트워크를 구성할 때 중간 지점에 AP를 설치하면 서버와 로봇 간의 무선 통신 거리가 더 늘어나는 이점이 있다.

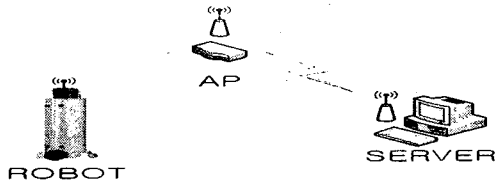

그림 1. 전체 시스템 개념도

3.2 하드웨어 구성

그림 2는 구현한 로봇 시스템의 블록도이다. 프로세서는 ARM 9 코어를 가진 Intel 사의 PXA255(XScale)로 고성능 저전력으로 향상된 멀티미디 기술 및 무선랜을 제공한다. Flash Memory는 운영체제(리눅스 커널 2.4.18) 및 응용 프로그램을 저장하고, SDRAM은 Flash Memory의 내용을 저장하여 프로세서가 실행할 수 있도록 한다. 이런 기본적인 시스템에 외부 디바이스인 WLAN, USB Host 2.0, CMOS 카메라, DC모터를 사용하였다.[2][3]

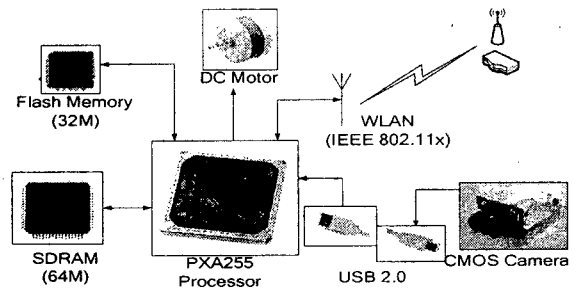

그림 2. 로봇 시스템 블록도

WLAN은 IEEE 802.11 (2.4GHz, 2Mbps, DS 방식) 규격을 표준으로 IEEE 802.11a (5GHz, 54Mbps, OFDM 방식), IEEE 802.11b (2.4GHz, 11Mbps, DS 방식), IEEE 802.11g (5GHz, 54Mbps, OFDM 방식) 등의 표준이 있으며 각 표준을 표 1에서 보이고 있다. 본 논문에서 사용한 WLAN은 PCMCIA 타입의 IEEE 802.11b 표준으로 구현하였다.[4]

표 1. WLAN 표준안 비교

Parameter	IEEE802.11a	IEEE802.11b	IEEE802.11g
주파수 대역	5GHz	2.4GHz	2.4GHz
거리	30~60m	60~100m	60~100m
최대 전송률	54Mbps	11Mbps	54Mbps
채널 접속	CSMA/CA	CSMA/CA	CSMA/CA
Duplexing	TDD	TDD	TDD
전송 기술	OFDM	DS	OFDM

*CSMA/CA : Carrier Sense Multiple Access / Collision Avoidance

*OFDM : Orthogonal Frequency Division Multiplexing

CMOS 카메라는 30만 화소의 CMOS이미지 센서를 채택하여 정지 영상 캡처 시 최대 640 × 480 픽셀에 초당 30프레임을 전송할 수 있다. 버스 방식이 USB로 USB Host 2.0 기능을 제공하는 TDI 칩을 사용하였고, DC모터 구동을 위해 GPIO를 사용하였다.

3.3 스프트웨어 구성

본 논문에서 구현한 소프트웨어는 크게 두 가지로 나눈다. 하나는 로봇의 소프트웨어이고, 다른 하나는 서버의 소프트웨어이다. 로봇은 운영체제 리눅스를 기반으로 시스템 C로 구현하였고, 서버는 PC기반으로 Visual C++로 구현하였다.

로봇의 소프트웨어는 BootLoader 및 운영체제 포팅, 디비이스 드라이버 구현, 응용 프로그램 구현으로 나눌 수 있다. BootLoader는 시스템의 전원을 켰을 때 가장 먼저 실행되는 프로그램으로 램 초기화, 기본적인 디바이스(시리얼, 네트워크 등) 초기화, 커널 램 복사 등의 기능이 있다. 운영체제인 리눅스 커널은 타겟 시스템의 하드웨어와 소프트웨어를 지원하는 최소한의 크기를 가지도록 구성된다. 기본적인 타겟 시스템의 개발환경을 구축하면 사용할 디바이스의 드라이버를 구현한다. 구현된 디바이스는 WLAN, USB Host 2.0, GPIO, CMOS 카메라이다. 각각의 디바이스 드라이버를 커널과 별도로 컴파일하고 필요할 때에만 커널에 포함시키도록 하는 방법을 사용하였다. 이러한 경우 시스템 부팅 시에 하드웨어를 검사 후, 해당 디바이스 드라이버를 커널에 추가하므로 자동으로 디바이스가 인식이 되도록 하였다.[5][6]

응용 프로그램은 WLAN을 이용하여 서버와 네트워크를 이루게 하는 소켓 프로그램과 카메라 영상을 캡처하는 기능(BMP format , 320 × 240 pixel) 및 모터를 구동하는 프로그램을 구현하였다. 소켓 프로그램은 서버와 접속 시, 서버 IP 및 포트번호를 입력으로 소켓을 생성하여 서버와 네트워크를 이루고, 데이터를 송수신하여 모터 구동 및 캡처된 데이터를 전송한다. 로봇의 제어 프로그램 순서도는 그림 3과 같다.

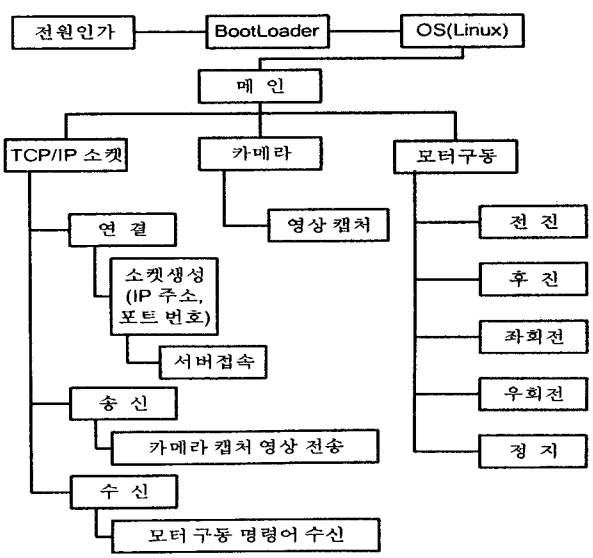

그림 3. 로봇 시스템 제어 프로그램 순서도

서버의 인터페이스는 메인 프레임 안에 명령어 입력 버튼, 정지 영상 데이터를 표시하는 부분과 소켓 프로그램으로 구성되었다. 입력 버튼으로부터 이벤트가 발생 시에 소켓에서 데이터를 송신하고, 정지 영상 데이터는 일정 시간마다 수신하여 표시를 하는 소프트웨어를 구현하였다. 그림 4는 구현한 서버 인터페이스의 전체적인 코드 구성이다.

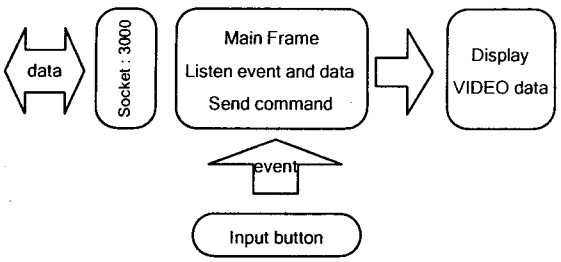

그림 4. 서버 사용자의 인터페이스 코드 구성

4. 실험 및 결과

구현된 로봇을 이용하여 서버와 네트워크를 생성하고 데이터를 송수신하는지를 실험하고 구현이 되었음을 확인하였다. 소프트웨어를 실행시키면 소켓의 연결을 기다리고, 로봇에서 연결을 요청하면 서버와 로봇은 네트워크를 이루게 되고, 서버에서 송신한 명령을 로봇이 적절히 움직이는 것을 확인하고 로봇에서 보내준 정지 영상 데이터를 서버에서 표시됨을 그림 5에서 보인다.

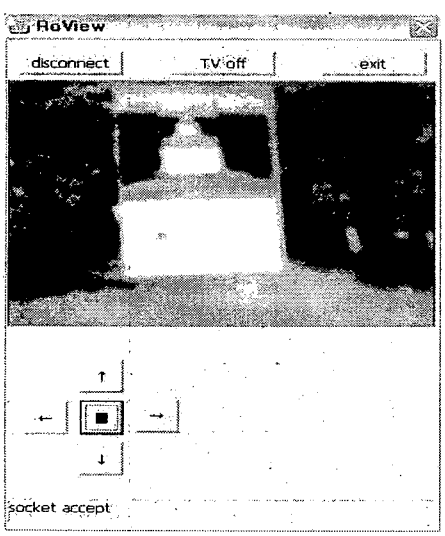

그림 5. 서버 사용자의 인터페이스

실험 결과 정지 영상 320 × 240 픽셀을 정상적으로 표현하였다. 그러나 초당 15프레임 이상 보내게 되면 화면이 끊어지는 현상이 발생하였다. 이는 WLAN IEEE 802.11b 채널을 단독으로 사용할 경우, 11Mbps의 전송률을 가지므로, 픽셀 수, 컬러 비트 수, 초당 프레임 수를 볼 때, 320 × 240 픽셀, 15비트 컬러, 초당 10프레임을 전송할 경우 11.52Mbps의 전송 속도로 안정된 화면을 표시할 수 있으며 그 관계를 표 2와 같다.[7]

표 1. 비디오 파일 형식별 전송속도

비디오 파일 형식			전송속도 (Mbps)
픽셀 수	초당프레임	컬러비트수	
640 × 480	30 fps	24 bit	221.18
320 × 240	30 fps	24 bit	55.30
160 × 120	30 fps	24 bit	13.82
320 × 240	15 fps	24 bit	27.65
320 × 240	10 fps	24 bit	18.43
320 × 240	10 fps	9 bit	6.91
320 × 240	10 fps	12 bit	9.21
320 × 240	10 fps	15 bit	11.52

5. 결론

본 논문에서는 ARM 9 코어를 내장하고 임베디드 리눅스를 기반으로 한 프로세서와 WLAN 기술을 활용하여 로봇 구동 및 영상을 감시하는 로봇을 구현하였다. 실험 결과를 분석하면 WLAN의 전송 속도에 영향이 크므로, 속도가 빠를수록 정지 영상의 끊김 현상이 없어지고, 더 높은 화질의 영상을 표시할 수 있다. 또, 임베디드 시스템을 활용하여 로봇의 소형화와 저전력 소비, 저렴한 가격으로 활용 분야가 확대될 것이다.

참 고 문 헌

[1] (주)한백전자, "임베디디 리눅스 시스템을 이용한 디바이 제어 및 통신의 이해", pp17-18, 2005년

[2] (주)하이버스, "X-Hyper255B-TKUII Developer's Manual", pp8, 2003년

[3] 김희수, "ARM 프로세서를 이용한 PMP 구현", 대한전기학회 제37회 하계학술대회 논문집, pp2138, 2006년 7월

[4] 김현동, 최상호 "802.16e 기반 시스템을 위한 시간동기화 방법에 관한 연구", 2005년도 정보 및 제어 학술대회 (CICS 05) 논문집, pp456, 2005년

[5] 유영창, "리눅스 디바이스 드라이버", 한빛미디어, pp109-132, 2004년

[6] 이동훈, "ARM 프로세서와 Linux를 이용한 마이크로 웹 서버 구현", 단국대학교 대학원 석사학위논문, pp12-18, 2002년

[7] 한경호, "ARM 9 임베디드 시스템에 의한 무선 컨텐츠 액세스 PMP 구현", 한국조명전기설비학회, 제21권 2호, pp101-103, 2007년 2월