

실시간 운영체제를 탑재한 원격 제어 로봇 시스템

Remote Controlled Robot System using Real-Time Operating System

이 태 희*, 조 상
(TaeHee Lee and Sang Cho)

Abstract : This paper presents a robot system that combines computer network and an autonomous mobile robot where RTOS is installed. We propose a wireless communication protocol, and also implement it on the RTOS of the robot system. Main controller of the robot processes the control program as a task type in the real-time operating system. Peripheral devices are driven by the device driver functions with the dependency of the hardware. Because the client and server program was implemented to support the multi-platforms by Java SDK and Java JMF, it is easy to analyze programs, maintain system, and correct the errors in the system. End-user can control a robot with a vision showing remote sight over the Internet in real time, and the robot is moved keeping away from the obstacles by itself and command of the server received from end-user at the local client.

Keywords : real-time, robot, remote control, software library, RTOS

I. 서론

원격지에 존재하는 로봇이나 기계를 조정하는 원격 제어의 응용 분야는 사용자가 직접 작업 현장에 존재하지 않아도 이를 수행할 수 있다는 점에서 공간적, 시간적 제약을 해소하고 위험성을 배제시킬 수 있는 장점을 갖기 때문에 활용 분야의 확대와 중요성이 점차 증대되고 있다. 그러나 이러한 원격제어는 최종 제어 대상과 사용자간의 원활한 통신보장과 현장의 상황을 사용자에게 전달할 수 있는 상황 전달 문제, 보안성, 로봇의 실시간 대응 등의 문제를 갖고 있다[1-3].

공간적인 제약성을 해결하기 위해 인터넷의 도입이 활발하게 진행되고 있으며, 인터넷 기반의 원격제어는 반응속도가 느리고 보안문제를 갖는 웹 상에서의 처리보다는 독립적인 클라이언트 서버의 구조가 많이 연구되고 있다. 또한 인터넷상에서의 개발과 다양한 플랫폼에서의 구현문제를 해결할 수 있는 방안으로 Java를 이용한 사용자 인터페이스 부분과 네트워크 관련 부분의 구현이 활발하게 진행되고 있다 [2-4].

인터넷을 통한 제어에 있어서 시간지연으로 인한 직접 제어의 문제를 해결하기 위한 연구[1]에서는 경로 추종 제어기를 도입하여 사용자의 제어 입력이 지연되어 전달되는 경우에도 로봇이 이동경로에서 크게 이탈되지 않음을 제시하고 있다. 또한 Taylor[5]는 인터넷을 통해 전송되는 화상을 보고 로봇 팔을 제어하여 블록을 쌓는 시스템을 구현하였고, Chen[6-7]은 센서 기반의 이동 로봇을 인터넷을 통해 제어할 수 있는 시스템을 구현하였다.

본 논문에서는 사용자가 인터넷상의 클라이언트를 통해 로봇을 제어하는 서버에 접속하는 원격 로봇 제어 시스템의 모델을 제시하고, 서버와 로봇간의 무선 통신을 위한 무선 통신 프로토콜을 설계, 구현한다. 원격 제어되는 로봇은 소규모 RTOS를 탑재하여 실시간 응답성과 개발환경의 표준화를 지원하며, 제안된 프로토콜과 로봇의 모터 구동, 초음파 센서, LCD구동 등의 모듈을 제어하는 드라이버들을 라이브러리로 구축한다.

또한 클라이언트와 서버 프로그램은 자바(Java)를 이용하여 구현함으로써 멀티플랫폼을 지원하고, 로봇으로부터 서버에 전송되는 실시간 동영상을 JMF를 이용하여 클라이언트 측에서 재생함으로써 사용자의 로봇 모니터링 시스템을 제공한다.

II. 원격 제어 로봇 시스템

원격 제어 로봇 시스템은 크게 서버와 클라이언트 개념의 컴퓨터 네트워크와 자율이동 로봇의 구조를 가지며 제안된 시스템은 앞에서 기술한 바와 같이 서버 시스템, 클라이언트 시스템, 로봇 시스템으로 나뉘며 그림 1과 같다.

원격 제어 로봇 시스템의 서버 시스템, 클라이언트 시스템은 네트워크 기능, 보안, 데이터 베이스, 멀티플랫폼 지원 등을 위해 Java로 구현되며, 로봇 시스템은 실시간 운영체제에서 모든 제어모듈들이 드라이버에 의해 구동되고, 각 드라이버들은 태스크의 형태로 C언어에 의해 구현된다.

인터넷을 이용한 원격 제어 로봇 시스템의 경우에는 인터넷 브라우저를 기반으로 응용 프로그램이 구현되는 경우와 독립적인 클라이언트/서버 프로그램을 소켓(Socket)을 이용해서 구현하는 형태로 구분된다. 웹 브라우저를 기반으로 구현된 원격 제어 로봇 시스템의 경우는 특별한 프로그램의 설치가 요구되지 않는 장점과 브라우저라는 실행 환경의 제약으로 속도, 보안, GUI의 제약 등과 같은 단점을 갖는다.

* 책임저자(Corresponding Author)

논문접수 : 2004. 2. 11., 채택확정 : 2004. 7. 1.

이태희 : 동우대학 컴퓨터정보과(thlee@duc.ac.kr)

조 상 : 청주대학교 컴퓨터정보공학과(sangcho@chongju.ac.kr)

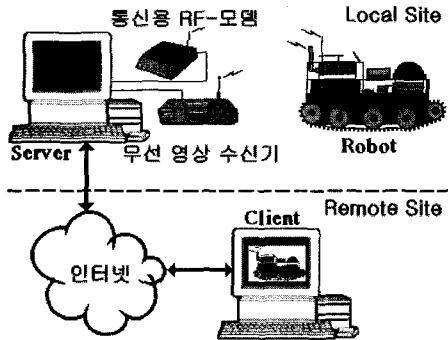


그림 1. 원격 제어 로봇 시스템의 전체 구조.
Fig. 1. Overall structure of remote controlled robot system.

독립적인 클라이언트/서버 프로그램을 이용한 원격 제어 로봇 시스템의 경우는 사용자의 컴퓨터에 클라이언트 프로그램이 설치되어야 하는 단점이 있지만, 다양한 기능의 구현과 편리한 GUI, 보안 및 접속 규제 등에 있어서 시스템 설계자와 사용자의 편의성을 높일 수 있는 장점이 있다. 본 논문에서는 독립적인 클라이언트/서버 프로그램을 통한 원격 제어 로봇 시스템을 구현하고 있으며, 원격지의 로봇과 서버간에는 무선통신을 통하여 클라이언트 측의 사용자가 요구하는 명령과 로봇의 응답 정보를 상호 교환할 수 있도록 설계, 구현하였다.

원격 제어되는 로봇에 장착된 카메라로부터 전송되는 영상 정보는 무선으로 서버의 무선 영상 수신기를 통해 영상 캡처 장비에 입력되며, 이를 JMF를 이용하여 인터넷으로 클라이언트 측에 실시간으로 전송하여 사용자에게 제공하게 된다.

JMF는 동영상 등의 미디어 파일들을 재생하기 위한 자바(Java) API로 각종 미디어 포맷을 지원하며, RTP(Real-time Transport Protocol)를 통해 실시간 미디어 스트리밍을 지원하므로 로봇에서 보낸 영상을 정지영상이나 압축영상으로 변환하여 전송하고, 다시 복원하는 방식을 개선할 수 있다.

1. 로봇 시스템

로봇 시스템은 80196 마이크로프로세서로 제어보드를 설계하였으며, 20MHz의 클럭으로 동작하고, 32KB의 ROM과 32KB의 RAM을 시스템 메모리로 사용한다. 로봇은 무한궤도를 이용하여 이동하며, 주행을 위한 DC 모터와 조향을 위한 서보 모터를 각각 1개씩 가지고 있으므로 주행속도를 PWM으로 제어하고 주행방향을 전진과 후진으로 제어하는 DC 모터 제어 모듈, 그리고 PWM을 이용하여 조향제어를 담당하는 PIC모듈이 로봇의 주행기능을 담당한다. 조향을 담당하는 PIC모듈은 16F84를 이용하여 외부의 인터럽트에 반응하여 인터럽트 요청과 함께 수신된 조향 정보에 따라 서보 모터의 회전엔 상응하는 PWM펄스를 서보 모터에 출력한다.

16문자 x 2라인의 LCD를 이용하여 메인 시스템의 주요 정보를 사용자에게 출력하며, 무선영상 전송장비의 On/Off 제어를 위해 소형 릴레이를 사용한다.

서버와 명령 및 데이터를 교환하기 위해서 RS-232 인터페이스

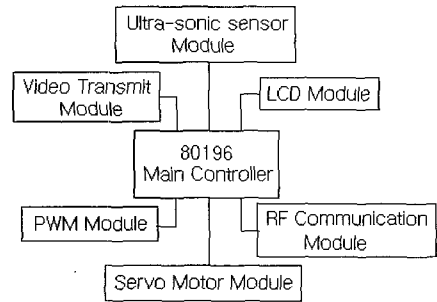


그림 2. 로봇 시스템의 제어모듈.
Fig. 2. Control modules in Robot system.

이를 장착하여 무선 RF 모듈이나 블루투스 통신 모듈을 장착한다.

장애물 검출을 위해서 초음파 모듈 SRF04를 로봇의 전방과 후방에 각각 3조씩 장착하여 메인 시스템의 장애물까지의 거리 탐지 인터럽트 요청에 대해 Cm 단위의 정보를 차례로 제공한다.

2. 서버와 클라이언트 시스템

서버와 클라이언트 시스템은 사용자와 로봇을 컴퓨터를 통해 인터넷이라는 네트워크로 연결해 주며, 사용자는 클라이언트를 통해 서버와 접속하고 서버는 로봇과 교신한다. Java를 이용해서 작성되어 멀티 플랫폼을 지원할 수 있으며, 독립적인 클라이언트/서버 프로그램으로 작성되어 소켓을 통해 클라이언트와 서버 시스템간의 직접적인 정보교환이 이루어지므로 웹 브라우저를 이용해서 구현된 경우보다 GUI의 구현에 적합하고, 보안문제에 유연하게 대처할 수 있으며, 수행 속도가 향상되는 장점을 갖는다.

클라이언트 시스템은 사용자의 명령을 입력받는 부분과 서버 또는 로봇으로부터의 정보를 받아 사용자에게 출력하는 부분, 로봇으로부터 전송 받은 동영상을 재생하는 부분으로 구성된다. 클라이언트 시스템에서 로봇으로 전달하는 명령이나 서버 또는 로봇으로부터 클라이언트로 수신되는 정보들은 자체적으로 설정한 명령과 자료구조를 사용한다. 로봇이 무선으로 전송한 동영상은 서버로부터 RTP를 통해 클라이언트에서 실시간으로 수신되어 재생된다.

서버 시스템은 클라이언트 시스템의 사용자 요구에 따라 로봇을 제어하는 기능과 로봇에서 클라이언트로의 영상 및 정보를 제공하는 역할을 한다. 따라서 서버 시스템은 무선통신장치와 무선 영상 수신장비를 탑재하고 있어야 한다. 또한 접속자의 수를 제한하고, 로봇에 접근할 수 있는 사용자의 접속여부를 판단한다.

III. 실시간 운영체제와 라이브러리 구현

본 연구에서 로봇을 제어하는 시스템은 16비트 마이크로 프로세서를 이용하며, ROM영역으로 32KB 크기의 물리적인 기억 공간 제약을 갖는다. 원격제어를 담당하는 대부분의 마이크로프로세서가 갖는 기능적인 한계 때문에 내장형 시스템(Embedded System)의 기억공간은 대부분 128KB이하의 소규모인 경우가 많지만 원격 제어를 실시간으로 수행

하기 위해서는 고도의 소프트웨어 기술이나 실시간 운영체제(RTOS)를 탑재해야 한다. 또한 제어 시스템 소프트웨어는 주요 태스크의 형태로 주요 프로세스들을 구현하고, 주변 장치들은 드라이버를 통해 시스템 자원으로 사용해야 한다.

1. 실시간 운영체제

로봇의 제한적인 메모리 공간에 실시간 운영체제를 설치하기 위해서 전체 소프트웨어 시스템의 크기가 제약을 받는다. 본 논문에서는 교육 및 상업용으로 다양하게 사용되고 있는 MicroC/OS-II를 제어용 실시간 운영체제로 이용한다. MicroC/OS-II는 이식이 용이하고, 롬(ROM)에 내장할 수 있으며, 커널 사이즈 조절이 용이한, 선점형 실시간 멀티태스킹 커널로 최대 64개의 태스크를 지원한다. 실제로는 8개의 태스크를 시스템 용도로 사용하고 있으므로 응용프로그램에서는 최대 56개의 태스크를 사용할 수 있다. 단, 라운드-로빈 방식의 스케줄링을 지원하지 않으므로 모든 태스크는 상호간에 중복되지 않는 고유 우선 순위를 가져야 한다.

모든 태스크들은 자신의 스택 크기를 서로 다르게 설정하는 것으로 메모리 사용량을 조절할 수 있다. 인터럽트를 활용하면 중첩된 모든 인터럽트 처리가 끝나는 시점에서 바로 더 높은 우선 순위의 태스크로 문맥전환을 발생시킬 수 있으며, 255회까지 중첩이 가능하다.

2. 운영 태스크

로봇을 제어하기 위한 핵심적인 태스크는 로봇이동, 장애물 검출, 자체 진단, 무선통신을 담당하는 네 가지이며, 우선 순위는 장애물 검출, 무선통신, 자체 진단, 로봇 이동의 차례로 자체 진단 태스크의 우선 순위가 가장 높다. 로봇에서 제어되는 하드웨어 장치로는 DC 모터, Servo 모터, LCD (16문자 x 2라인), Video 전송 제어기, 무선 통신 장치, 초음파 거리 측정 모듈 등으로 구성된다. 각 태스크별 기능은 다음과 같다.

① 자체 진단 (우선 순위 3)

전원 상태, 각 제어 모듈의 동작 상태 등을 점검하여 오류가 있는 경우는 모든 동작을 멈추고 제어용 서버에 오류상태를 전송한다. 전원의 경우는 동작이 가능하면 정상적인 동작 모드로 복귀하며 각 제어 모듈에서 오류가 발생되면 시스템 리셋(reset)이 필요하다.

② 무선 통신 (우선 순위 2)

무선 모듈을 통해 패킷 형태로 명령과 데이터를 송수

신하며 원격지의 제어명령에 따라 로봇의 각 제어 드라이버를 통해 각 장치를 구동한다.

③ 장애물 검출 (우선 순위 1)

초음파 거리 측정모듈을 이용해서 주변의 장애물 검출을 관리한다. 최대 8개의 초음파 거리 측정 모듈로부터 장애물까지의 거리정보를 8비트 해상도로 받아서 장애물 회피 및 운행제한 등의 처리를 직접 수행하고 서버에게 알리는 기능을 수행한다. 각 초음파 거리 모듈은 Cm 단위로 거리정보를 전송하며 최대 255Cm 거리까지를 감지할 수 있으며 본 실험에서는 6개의 초음파 모듈을 사용한다.

④ 로봇 이동 (우선 순위 4)

모터의 회전 방향과 회전수를 제어하며, 로봇의 좌우 회전과 관계된 서보모터의 제어도 담당한다. 무한계도에 연결된 이동거리 감지 센서로부터 좌우의 이동거리를 확인하여 지시된 방향으로의 진행이 유지되도록 자체적으로 제어한다. 무선통신과 자체진단, 장애물 검출에 따라 이동에 대한 제한과 시도가 이루어진다.

3. 드라이버 라이브러리

드라이버는 로봇의 주변 장치들을 하드웨어 의존적인 내용으로 제어하는 함수들을 가지고 태스크로부터의 요구에 대응하는 프로그램이다. 또한 태스크들이 주변 장치들에 접근하는데 있어서 자원의 공유를 효율적으로 수행시키고 프로그램의 추상화와 계층화를 가능케 한다.

드라이버 프로그램은 하드웨어 의존적인 부분이 많기 때문에 드라이버의 실행 중에는 RTOS의 태스크 점유시간이 비결정적일 수 있으므로 본 연구에서는 RTOS의 태스크 전환을 유보시키면서 드라이버 고유의 하드웨어를 제어하는 부분을 최소화하며, 해당 장치에 실질적인 제어를 취하지 않는 경우는 최대한 자원의 사용권을 독점하지 않는다.

본 논문에서 구현한 원격 제어 로봇 시스템에서는 주변 장치로 무선 통신장치, 로봇을 이동시키기 위한 구동장치, 주변 장애물과의 거리를 초음파로 측정하는 장치, 로봇에 장착된 카메라로부터 입력되는 영상을 실시간으로 무선 전송하는 장치, 16문자를 2줄까지 출력할 수 있는 LCD 장치 등이 있다.

4. 소프트웨어 구성

전체 소프트웨어의 구조는 표 1과 같이 클라이언트, 서버, 로봇 제어의 세 부분으로 구성된다.

제어용 클라이언트와 서버 프로그램은 소켓을 이용하여 TCP/IP에서의 상호 메시지 교환을 처리하며 RTP(Real-time Transport Protocol)기반의 JMF에서 제공하는 실시간 Stream 전송 함수를 이용하여 서버의 영상 캡처 장비로부터 획득된 영상을 클라이언트 측으로 전송하고 클라이언트에서 동영상으로 재생한다.

JMF API를 이용함으로써 다양한 플랫폼에서 수행이 가능하면서 다양한 멀티미디어 포맷과 스트리밍, 미디어 캡처 등의 작업을 RTP를 통해 다른 클라이언트로 전송이 가능하다.

각 소프트웨어를 구성하는 주요함수들을 살펴보면 표 2에서 볼 수 있듯이 서버와 클라이언트 모두 소켓기반에 영상정보의 전송과 수신 및 재생이 공통적이며 서버의 경우는

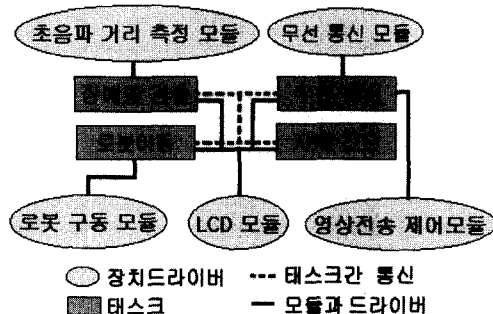


그림 3. 로봇 제어 태스크와 드라이버 구성.
Fig. 3. Control task of robot and driver structure.

클라이언트의 버튼 이벤트로부터 소켓을 통해 수신되는 제어 명령에 따라 RF 무선 모뎀을 통해 로봇에게 해당 제어 명령을 전송하는 구조임을 알 수 있다.

제어용 컴퓨터에서 명령을 전송하는 프로그램은 그림 4와 같이 자바를 이용해서 작성되어 인터넷 기반에서 프로그램을 확장할 수 있도록 하였으며, 다중 플랫폼을 지원하기 위해 표준 인터페이스를 지원하는 자바 패키지만으로 작성되었다.

표 1. 원격 로봇 제어 시스템의 소프트웨어 구조.

Table 1. Software structure of System.

구분	주요 구성 요소	담당 기능
Server (Class)	Server	무선 통신 및 소켓, 명령 처리
	AVTransmitt	영상 획득 및 RTP를 통한 전송
	SerialReader	RS-232 수신 인터럽트 처리
	StateListener	영상 요청 처리
Client (Class)	Client	소켓을 통한 사용자 명령 전송 RTP를 통한 영상 수신 및 재생
	CommTask	무선으로 명령 수신 및 해독
Robot (Task)	DiagTask	자체 점검
	SensTask	초음파 및 이동 거리 모듈 구동
	MoveTask	로봇의 이동 제어
	CharTask	LCD 출력

표 2. 서버와 클라이언트의 주요 구성 함수.

Table 2. Main functions of Server and Client.

함수명	기능	소속
Server()	생성자	Server
readSocket()	소켓을 통한 수신	StateListner
writeSocket()	소켓을 통한 송신	StateListner
tx_Video()	영상 전송	AVTrensmitter
build_Packet()	패킷 구성	SerialReader
gen_crc_table()	CRC 참조표 생성	SerialReader
SetSerialPort()	직렬 포트 초기화	SerialReader
update_crc()	CRC 갱신	SerialReader
packet_Flow()	무선 통신 흐름제어	SerialReader
tx_data()	무선 자료 전송	SerialReader
tx_packet()	무선 패킷 전송	SerialReader

함수명	기능	소속
CapVideo()	비디오 캡처	AVTrensmitter
Start()	영상 전송 시작	AVTrensmitter
Stop()	영상 전송 정지	AVTrensmitter
CheckVideoSize()	영상 크기 검사	AVTrensmitter
SetJPEGQuality()	영상 품질 설정	AVTrensmitter
CreateProcessor()	처리기 생성	AVTrensmitter
CreateTransmit()	전송기 생성	StateListener
SerialEvent()	직렬통신 이벤트	Server
ControlUpdate()	전송영상 갱신	StateListener

(a) 서버의 주요 구성 함수

함수명	기능
Client()	생성자
initialize()	초기화
ReadSocket()	소켓을 통한 수신
start_rx()	영상 수신 시작
Button_Pressed()	입력 이벤트
find()	영상 재생기 탐색
start()	영상 재생
stop()	영상 재생 중지
isDone()	영상 재생 완료
update()	영상 정보 갱신

(b) 클라이언트의 주요 구성 함수

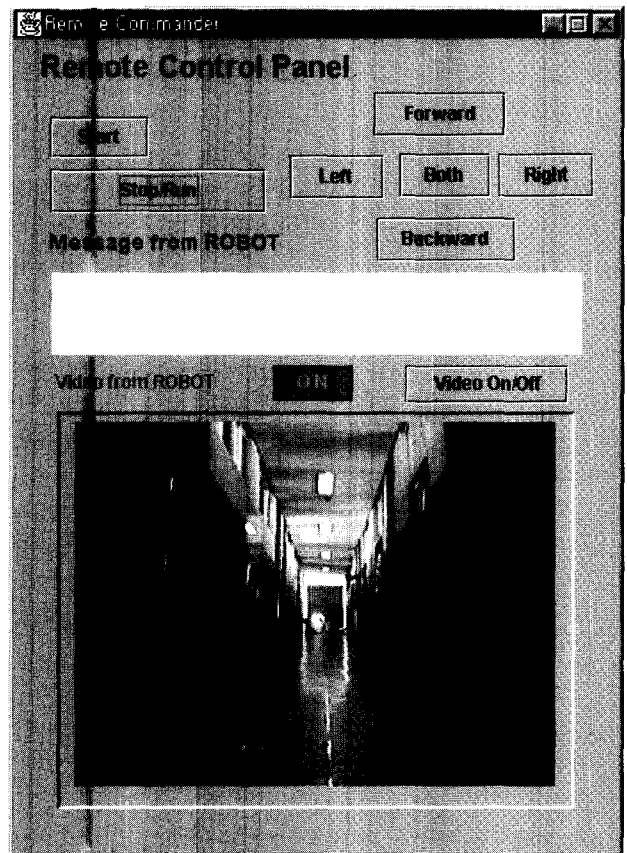


그림 4. 클라이언트의 제어 프로그램.

Fig. 4. Client control program.

핵심적인 구성은 기본 자바 SDK를 이용하였으며, GUI 부분은 Swing 컴포넌트를 이용해서 작성되었고, 직렬통신 부분은 IO 패키지 중에서 통신 API를 이용하여 작성되었으며, 로봇으로부터 무선으로 수신된 동영상은 캡처 장비에 의해 실시간으로 처리하기 위해 JMF 패키지를 이용하고 있다.

원격 로봇의 배터리 전원을 고려하여 로봇에 장착된 카메라는 제어용 컴퓨터에서 명령을 통해 전원을 차단할 수 있으며, 특정한 메시지를 전송할 수 있도록 대화창을 설치하여 서버와 클라이언트 컴퓨터간의 대화가 가능하도록 구성하였다.

4.1 서버 프로그램

서버 프로그램은 클라이언트 프로그램에서 소켓 접속 요청에 따른 응답으로 동작을 개시하여 RTP를 통해 동영상을 전송하는 전송기(Transmitter) 객체를 생성하는 것이 가장 중요한 초기화 작업이다.

클라이언트로부터 소켓 접속요청에 따른 인터넷 접속이 완료되면 로봇과 통신하기 위한 직렬포트를 초기화하고 클라이언트로부터 수신되는 사용자의 제어명령을 분석하여 해당되는 제어명령을 로봇에게 무선 프로토콜을 통해 전달한다.

클라이언트의 사용자와 메시지를 주고받을 수 있는 대화창을 제공하고 클라이언트 측에서 프로그램을 종료시키면 서버 프로그램도 모든 객체를 종료시키고 프로그램 수행을 종료한다.

4.2 클라이언트 프로그램

클라이언트 프로그램에서는 초기화 부분에서 소켓을 생성시켜 서버와의 접속을 시도하고 실패한 경우는 종료되며, 소켓을 통한 서버와의 접속이 성공적으로 이루어지면 JMF 초기화 부분을 실행시켜 프로그램에서의 동영상 구현 여부를 판단한다.

클라이언트 프로그램의 주요 초기화 부분은 인터넷을 통한 제어에 있어서 서버로 전송하는 명령은 소켓을 통해 정보를 교환하며, 동영상은 RTP를 이용하는 JMF를 이용함으로써 명령과 감시용 동영상의 전송 경로를 분리하여 구현하였다. RTP를 이용한 동영상의 제공은 로봇이 위치한 원격지의 영상을 감시하기 위한 것으로 로봇을 제어하는 것을 보조하기 위한 수단이므로 RTP의 세션이 수립되지 않아 동영상이 수신되지 않더라도 전체 프로그램은 수행된다.

TCP/IP를 이용한 소켓의 생성과 RTP세션에 관한 초기화 과정이 수행된 후에는 클라이언트의 제어 프로그램에서 버튼으로 입력되는 제어정보를 서버에 전송하거나 메시지 창을 통해 입력된 내용을 서버에 전송하는 과정만이 프로그램이 종료될 때까지 반복된다.

4.3 로봇 시스템의 제어 프로그램

로봇 시스템은 MicroC/OS-II를 RTOS로 탑재하여 통신, 센서 입력, 자체 점검, 로봇이동 등의 태스크를 통해 서버의 명령에 따라 움직일 수 있도록 구성되었다.

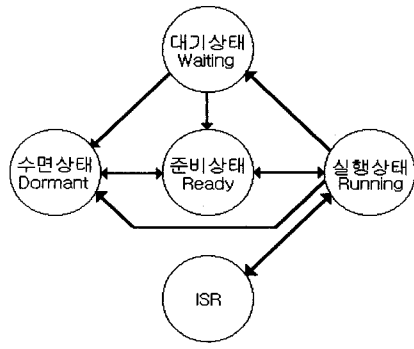


그림 5. MicroC/OS-II에서 태스크의 상태.
Fig. 5. Task status of MicroC/OS-II.

표 3. RTOS에서 수행되는 제어용 함수.

Table 3. Control functions running in RTOS.

함수명	기능
init_pwm()	PWM기능의 초기화
set_pwm()	PWM 설정
Env_init()	시스템 초기 값 설정
gen_crc()	CRC 참조표 생성
update_crc()	CRC 갱신
build_packet()	패킷의 구성
get_distance()	장애물 거리 획득
get_speed()	속도 감지
tx_data()	무선 자료 전송
tx_packet()	무선 패킷 전송
packet_Flow()	패킷 흐름 제어

MicroC/OS-II에서 각 태스크는 그림 5와 같이 수면상태(Dormant), 준비상태(Ready), 실행상태(Running), 인터럽트 서비스 루틴(ISR, Interrupt Service Routine) 상태 중 한가지 상태로 무한 루프를 수행한다.

수면상태는 태스크가 메모리에 존재하고 있지만 멀티태스킹 커널에 등록되지 않아 실행될 수 없는 상태를 말한다. 준비상태는 언제든 실행될 수 있지만, 우선 순위가 더 높은 다른 태스크가 현재 실행되고 있는 상태이며, 태스크가 CPU를 점유해서 현재 실행되고 있는 상태가 실행상태이다. 태스크가 입출력 처리의 완료, 공유자원의 사용 가능 상태, 타이머의 만료를 기다리는 상태가 대기 상태이다. 마지막으로 인터럽트가 발생해서 CPU가 인터럽트 서비스 루틴을 수행하고 있는 상태가 ISR 상태이다.

로봇 시스템에서 사용되는 각 응용 태스크의 주요 함수들은 표 3과 같다. 제시된 함수 중에서 임계영역(Critical Section)을 사용하는 함수들은 하드웨어 자원에 직접 접근하므로 처리과정에서 실행이 중단되지 않아야 하는 부분들을 포함하기 때문에 임계영역의 전후에서 인터럽트의 금지와 사용이 가능하도록 설정해 주어야 한다.

RTOS를 구성하는 시스템 태스크는 MicroC/OS-II에서 기본 제공되는 Idle 태스크이며 응용 태스크는 자체진단 태스크, 통신 태스크, 감지 태스크, LCD 출력 태스크, 로봇이동 태스크 등으로 구성된다.

로봇의 움직임을 제어하는 로봇이동 태스크는 통신으로 수신된 명령의 수행이나 센서정보로부터 장애물까지의 안전거리 확보를 위해 수행되는 태스크로 세마포어에 의해 운영되며, 우선 순위에 의해 문맥전환이 발생하여 상위의 우선 순위를 갖는 태스크가 자원에 접근하게 된다. 로봇 시스템에서는 센서로부터의 장애물 인식에 의한 회피가 통신에 의한 사용자의 명령에 우선하며, 자체 점검 태스크가 그 다음의 우선 순위를 갖고 로봇이동 태스크가 가장 낮은 우선 순위를 갖는다. 따라서, 이동중인 로봇의 진로에 장애물이 존재하는 경우 거리에 따라 속도가 감소되며, 30cm 내에 장애물이 존재하는 경우는 로봇이 정지하게 되고 장애물이 검출되지 않는 방향으로만 진행이 가능하다.

표 4. 로봇 시스템에서의 패킷 수신.

Table 4. Packet receiving at robot system.

구 분	정상 수신 패킷 수 (9600 bps)
1차 전송	496
2차 전송	3
3차 전송	0
4차 전송	1
총 전송 회수	506
재전송율(%)	1.2

IV. 시스템 평가

로봇 시스템은 16비트 마이크로 프로세서인 인텔의 80196 프로세서를 제어기로 사용하고 메모리를 RAM과 ROM 모두 32KB 용량으로 설치하여 실시간 운영체제인 MicroC/OS-II를 설치하였다. 로봇 시스템은 무선 통신, 로봇 이동, 센서 감지, 자체 점검 등의 기능을 태스크로 구성하여 다중 처리될 수 있도록 설계 구현되었다. 이러한 시스템에서는 하드웨어 의존적인 코드들이 일부 함수로 구현되어 태스크 단계의 코드에서는 논리적인 기술만으로 프로그래머가 원하는 기능을 수행시킬 수 있다는 장점이 있다. 또한 하드웨어에 대한 접근을 운영체제의 임계 영역 설정과 세마포어의 활용을 통해 효율적으로 수행할 수 있으며 다수의 태스크들이 요구하는 지연시간을 다른 프로세서에 할당함으로써 시스템의 처리능력을 향상시킬 수 있었다.

본 논문에서 채택한 클라이언트/서버 구조를 통한 인터넷 기반의 원격 로봇 제어방식은 로봇과 사용자가 제어입력을 생성하는 클라이언트 사이에 서버를 설치함으로써 직접적인 접근을 막을 수 있기 때문에 로봇의 보안기능을 활용하는 경우보다 서버의 보안 성능에 따라 로봇에 대한 접근 보안성을 높이기 용이하다.

클라이언트와 서버간의 정보교환에 있어서 제어 명령과 로봇 측의 동영상 정보를 전송하는 인터넷 채널을 소켓용 TCP/IP와 RTP로 구분함으로써 명령 전달용 채널과 데이터 수신 채널을 분리하여 과도한 자료의 송수신 없이 제어 명령을 교환할 수 있도록 구현하였다. 따라서 기존의 연구에서 로봇의 영상을 압축하여 무선 LAN을 통해 직접 클라이언트로 전송하는 경우보다 로봇에 요구되는 시스템 요구사항을 낮출 수 있었으며 TCP/IP 프로토콜이나 FTP를 통해 동영상 정보를 전송하는 경우보다 멀티미디어 스트리밍 전송에 있어서 안정적인 네트워크 환경을 제공받을 수 있다.

프로그램의 구현을 통해 본 연구에서 프로토콜에서의 재전송 제한을 5회로 설정하여 500회 수신한 결과, 표 4와 같이 대부분의 전송은 2차 재전송을 통해 98% 이상이 완료됨을 알 수 있으며, 최악의 4차 재전송에서도 정상적인 패킷의 전송이 수행된 결과를 통해 본 논문에서 제시된 프로토콜이 안정적으로 패킷을 전송하고 있으며 로봇의 움직임을 간접적으로 확인할 수 있다. 9600 bps의 속도에서 평균전송 시간은 89.2 ms로 수행되었다.

V. 결론

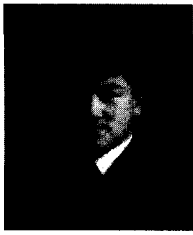
인터넷을 통한 원격 로봇의 제어에 있어서 시간적인 지연에 따라 발생할 수 있는 사용자와 로봇간의 제어의 불일치를 극복하기 위해서는 보다 빠르고 안정적인 속도의 인터넷 제공과 함께 로봇 스스로의 능동적인 대처 능력과 정밀한 사용자 명령의 전송이 보장되어야 한다.

본 논문에서는 사용자가 클라이언트에서 서버를 통해 로봇에 접근하는 원격 제어 로봇 시스템을 Java를 이용하여 웹 브라우저가 아닌 독립적인 클라이언트/서버 프로그램의 형태로 구현하였다. 인터넷을 통하여 자율 이동 로봇을 원격지에서 운용함으로써 공간적인 제약성을 탈피한 능동적인 운용 가능성을 제시하였으며, 서버와 로봇간의 안정적인 무선 통신을 위한 간단한 프로토콜을 설계하고 구현하였다. 또한 로봇에는 소규모 실시간 운영체제를 탑재하고 이에 따른 하드웨어 모듈들에 대한 제어를 위해 드라이버를 비롯한 소프트웨어 라이브러리를 구축하였다. 실험을 통해 본 연구에서 제안하고 구현한 무선통신 프로토콜의 안정적인 동작과 실시간 운영체제 도입에 따른 우수성을 확인할 수 있었다.

앞으로 소프트웨어 구조의 보다 논리적인 계층화와 하드웨어 모듈들에 대한 정형화된 모델을 설정하기 위한 후속 연구가 지속될 것이며, 로봇에 적합한 실시간 운영체제를 구축하기 위한 노력이 필요하다.

참고문헌

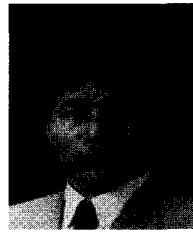
- [1] 한국현, "인터넷 기반 퍼스널 로봇 시스템의 제어 구조 설계," 석사 학위 논문, KAIST, 1999
- [2] 이동주, "웹 브라우저를 이용한 원격 로봇 제어," 석사 학위 논문, 목원대학교, 2000
- [3] 박태현, 강근택, 이원창, "인터넷을 이용한 이동 로봇의 원격 운용 시스템," *Journal of Control, Automation and System Engineering*, vol. 8, No. 3, March 2002
- [4] 조덕연, 최병욱, "임베디드 리눅스를 이용한 산업용 인버터의 웹 기반 원격 관리," 제어 자동화 시스템공학 논문지 제 9권 제 4호, pp. 340-346, Apr. 2003.
- [5] K.Taylor and B. Dalton, "Issues in Internet Telerobotics," in *Int. Conf. on Field and Service Robotics*, Dec. 1997.
- [6] T. M. Chen and R. C. Luo, "Remote Supervisory Control of An Autonomous Mobile Robot Via World Wide Web," in *Proc. IEEE Int. Symposium on Industrial Electronics*, vol. 1, pp. ss60-ss64, July 1997.
- [7] T. M. Chen and R. C. Luo, "Multisensor Based Autonomous Mobile Robot Through Internet Control," in *Proc. of Int. Conf. on Industrial Electronics, Control, and Instrumentation*, pp. 1284-1253, Aug. 1997.



이 태 희

1969년 5월19 일생. 1995년 청주대학교 전자계산학과 졸업(학사). 1997년 청주대학교 전자계산학과 졸업(석사). 2004년 청주대학교 전산정보공학과 졸업(박사). 2001년~현재 동우대학 컴퓨터그래픽과 전임강사. 관심분야 : 임베디드

시스템, 실시간 시스템.



조 상

1952년 1월 23일생. 1975년 서울대학교 물리학과 졸업(이학사). 1986년 Iowa State University 전산학과 졸업(이학석사). 1990년 University of Texas 전산학과 졸업 (이학박사). 1992년~현재 청주대학교 컴퓨터정보공학과 부교수. 관심

분야 : 실시간 시스템, 소프트웨어컴퓨팅.