
UPnP 기반 지능형 로봇 제어 시스템 구현

김성우* · 박유현** · 권순각***

Implementation of an Intelligent Robot Control System Based on UPnP

Seong-Woo Kim* · Yoo-Hyun Park** · Soon-kak Kwon***

이 논문은 2012년도 동의대학교 연구비를 지원받았음(연구과제번호: 2012AA181)

요 약

유비쿼터스 홈 서비스에 대한 기대가 높아지면서 지능형 로봇에 대해 폭넓은 관심이 요구되고 있다. 이러한 로봇은 원격 서버와 연동할 수 있는 미들웨어를 통해 다양한 서비스를 제공할 수 있다. 본 논문에서는 지능형 로봇의 원격 제어를 위한 UPnP 구조 기반의 오픈 소스 소프트웨어 프레임워크를 개발하였다. 홈 네트워크에 많이 사용되는 UPnP 구조는 명령 제어, 이벤트 전달, 웹 서비스 등 다양한 통신 방식을 제공한다. 본 논문에서는 2 종류의 실제 플랫폼에 탑재되어 성공적으로 구현된 결과를 보여준다.

ABSTRACT

With the increased demand of ubiquitous home services, intelligent robots have wide attentions. This kind of robots offer various services through middleware components which can connect with remote servers. In this paper we present an UPnP(Universal Plug and Play) based open-source software framework which makes control the robots remotely. This UPnP architecture for home networking can provide various communication methodologies like command control, eventing, presentation with web services and so on. This paper shows successful implementation results on two real platforms.

키워드

URC 로봇, UPnP 기술, 지능형 로봇, 로봇 미들웨어

Key word

ubiquitous robot companion,, universal plug and play, intelligent robot, robot middleware

* 정회원 : 동의대학교 컴퓨터소프트웨어공학과(교신저자, libero@deu.ac.kr)

접수일자 : 2012. 05. 24

** 종신회원 : 동의대학교 컴퓨터소프트웨어공학과

심사완료일자 : 2012. 07. 19

*** 정회원 : 동의대학교 컴퓨터소프트웨어공학과

I. 서 론

디지털 컴퓨터, 센서, 무선 통신 기술의 발달로 언제 어디서나 다양한 서비스를 제공하는 로봇들이 등장하고 있다. 특히, 이러한 개념의 로봇들 중 URC (Ubiquitous Robot Companion) 라는 지능형 서비스 로봇은 유무선 통신 장치를 장착하고 영상/음성인식, 위치인식과 같은 고급 기술들을 바탕으로 사용자에게 다양한 서비스를 제공할 수 있다. 예를 들면, 노인들을 따라다니며 건강을 점검한다든지 집안의 청소나 각종 센서를 제어하는 인간 친화적인 기능을 들 수 있다.

이러한 현대의 로봇들은 수 많은 하드웨어와 소프트웨어 모듈들을 포함하는 복잡한 분산 시스템으로 생각할 수 있다. 로봇 내의 센서, 구동기, 통신 장치 등의 하드웨어는 특정한 작업을 수행하기 위하여 소프트웨어 모듈들과 긴밀하게 협력해야 하며, 로봇은 사용자에게 다양한 서비스를 제공하기 위하여 원격 로봇 서버와 같은 외부 환경과의 상호작용을 할 수 있어야 한다. 따라서 이처럼 복잡한 로봇 기능들을 구현하기 위해서 로봇 소프트웨어는 응용 소프트웨어 컴포넌트들이 유기적으로 연결되고 손쉽게 결합할 수 있도록 하는 미들웨어 구조를 가져야 한다.

그동안 로봇 미들웨어에 대하여 다양한 연구가 이루어져 왔다. Miro는 독일 Ulm 대학에서 개발된 객체지향 로봇 미들웨어이다[1]. Miro는 Tao라는 CORBA 소프트웨어를 사용하여 디바이스를 추상화하고 다양한 서비스를 제공하는 클래스 프레임워크를 제공한다. 이러한 객체 지향 설계 방식을 적용함으로써 분산 로봇 제어를 위한 프로세스 및 플랫폼 상호가용성을 높이고 있다. 한편, 일본 도시바에서는 단일 자동차와 분산 센서 네트워크를 활용하는 로봇들을 개발하기 위해 Orca라는 모듈화 기반의 미들웨어 프레임워크를 개발하였다[2]. Orca는 사용자가 인터페이스와 통신 방식을 정의하도록 하여 모듈화 기반 분산 로봇 시스템을 구현할 수 있도록 하였다.

이와 별도로 2002년부터 CORBA를 기반으로 하는 로봇 미들웨어인 RT-Middleware를 개발하였다[3]. 이것의 목적은 기능적인 로봇 부품들을 모듈화된 소프트웨어 구조로 만들고 이것들을 단순히 결합함으로써 실제 로봇을 제작할 수 있도록 하는 것이다. 한편, Willow Garage

사의 ROS (Robot Operating System)는 이기종 기계들 간의 운영체제 위에 구축된 통신 계층을 제공하는 오픈 소스 로봇 운영체제이다. 이것은 이기종의 하드웨어뿐만 아니라 다양한 컴퓨터 언어와 오픈 소스 소프트웨어 패키지들을 결합하여 소프트웨어 재사용성을 극대화하도록 설계하였다[4]. 이러한 지능형 로봇을 위한 미들웨어들은 소프트웨어 컴포넌트들을 조합하여 로봇의 기능들을 효율적으로 제어하도록 하는 반면 통신 구조가 무겁고 전체적으로 구성이 복잡하고 큰 메모리 용량을 필요로 한다.

한편, 스마트 폰, TV, 전등, 블루레이 플레이어 등의 가용용 전기기기들의 제어 및 정보전송에 많이 쓰이는 UPnP 기술은 분산 네트워크 환경에 적합하며 동적으로 재구성하기가 쉬운 구조를 가지므로 지능형 로봇에도 이를 적용하는 방안이 제안되었다[5, 6]. 본 논문에서는 이러한 UPnP 기술을 활용하여 스마트 기기를 통해 지능형 로봇을 제어하고 감시할 수 있는 새로운 로봇 제어 시스템을 제안한다. 본 시스템은 지능형 서비스 로봇을 위한 명령 실행, 센서 모니터링, 카메라 감시, 웹 서비스 등의 다양한 UPnP 서비스를 통해 로봇을 탐색하고 제어하고 모니터링할 수 있도록 설계되었다. 또한 본 논문에서 구현한 로봇 소프트웨어는 오픈 소스로 공개되어 있어 누구나 참고할 수 있으며 간단한 구조로 쉽게 적용할 수 있다.

본 논문의 구성은 다음과 같다. 먼저 UPnP 기술에 대하여 2장에서 간략하게 소개한다. 3장에서는 제안하는 UPnP 로봇 제어 시스템의 구조와 세부적인 기능에 대하여 설명하고 4장에서는 UPnP 로봇 시스템을 구현한 결과에 대하여 설명한다. 마지막으로 5장에서는 결론을 맺는다.

II. UPnP 구조

UPnP (Universal Plug and Play)는 TCP/IP, SSDP, SOAP, GENA, HTTP 및 XML과 같은 인터넷 통신 표준과 기술을 기반으로 하여 네트워크로 연결된 기기들 간에 제공하는 서비스를 발견하고 서로 통신을 가능하게 해주는 구조의 일종이다[7]. UPnP는 본래 마이크로소프트사가 제안하였는데 현재는 수백 개의 업체가 참여하

는 UPnP 포럼에서 표준을 관리하고 있다. UPnP 포럼은 1999년에 결성되었는데, 현재 가전, 컴퓨터, 홈오토메이션, 홈 시큐리티, 프린터, 네트워크 등 여러 분야의 800여 개 업체가 참여하고 있다.

UPnP 구조에서 가장 중요한 3가지 요소는 디바이스(device), 서비스(service), 컨트롤포인트(control point)이다. 디바이스는 특정한 서비스를 제공하는 장치를 말하며, 여기서 서비스는 디바이스의 특정한 기능에 대한 소프트웨어 구현이다. 반대로, 컨트롤포인트는 서비스를 요청하는 기기 또는 기능을 말한다. 어떤 물리적인 기기는 리모컨처럼 컨트롤포인트의 기능만을 가지기도 하고, TV처럼 디바이스와 서비스를 포함하기도 한다.

UPnP는 네트워킹을 위해 TCP/IP, UDP, HTTP와 같은 기존의 인터넷 프로토콜을 사용하고 디바이스 간에 전송되는 데이터는 XML로 표현하므로, 이기종 미디어나 제품 간 확장성이나 상호운용성이 용이하여 여러 분야의 파급효과가 크다. 그림 1은 UPnP의 프로토콜 스택을 보여준다. 여기서, SSDP(Simple Service Discovery Protocol)은 네트워크 상의 서비스를 찾기 위한 프로토콜이며, SOAP(Simple Object Access Protocol)은 제어 명령(Action)을 보내기 위한 프로토콜이며, GENA(General Event Notification Architecture)는 디바이스 상태가 변할 때 이를 다른 디바이스에 이벤트로 알려주는 프로토콜이다.

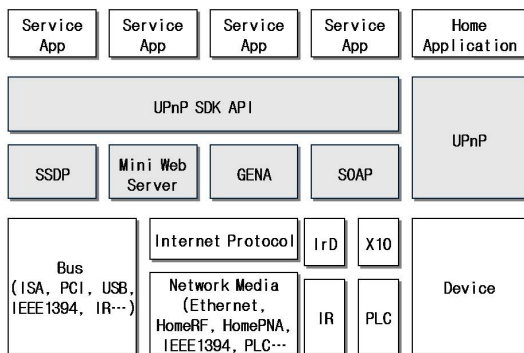


그림 1. UPnP 소프트웨어 구성
Fig. 1 UPnP Software Architecture

UPnP를 통한 네트워킹 단계는 다음과 같은 총 6단계로 나누어진다.

- 1) 자신의 네트워크 주소를 획득하는 Addressing 단계
- 2) SSDP 프로토콜을 통해 디바이스가 브로드캐스트하여 자신을 알리고 다른 디바이스와 서비스를 찾는 Discovery 단계
- 3) 컨트롤포인트가 HTTP 프로토콜을 통해 찾은 디바이스에 대한 기술문서를 XML로 가져오는 Description 단계
- 4) 컨트롤포인트가 기술문서를 파악하여 SOAP 프로토콜을 통해 원하는 디바이스 서비스 명령(Action)을 요청하는 Control 단계
- 5) 이와 별도로 컨트롤포인트가 서비스에 요청하여 서비스의 상태가 변했을 때 GENA 프로토콜을 통해 이벤트 메시지를 받도록 할 수 있는 Eventing 단계
- 6) 디바이스가 제공하는 URL을 통해 브라우저로 가져와 디바이스를 제어하거나 상태를 볼 수 있는 Presentation 단계

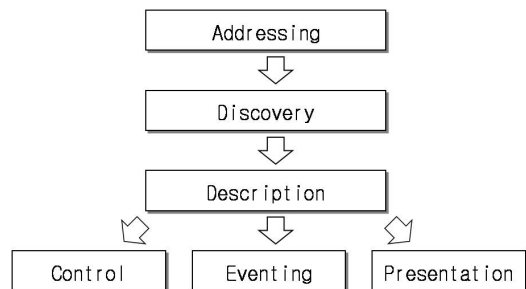


그림 2. UPnP 네트워킹 단계
Fig. 2 Steps to UPnP Networking

III. UPnP 로봇 제어 시스템

지능형 서비스 로봇은 크게 로봇 클라이언트, 로봇 서버, 그리고 이들 간의 통신과 같은 세부 시스템으로 나누어진다. 로봇은 사람과 상호작용하여 원하는 서비스를 제공하는 개체이지만 제어 서버의 통제 하에 동작하게 되고, 이들은 각각 몇 개의 서브시스템으로 구성되어 서로 연동된다.

3.1. UPnP 로봇 소프트웨어

본 논문에서 제안하는 UPnP 로봇 소프트웨어의 구성은 다음 그림과 같다.

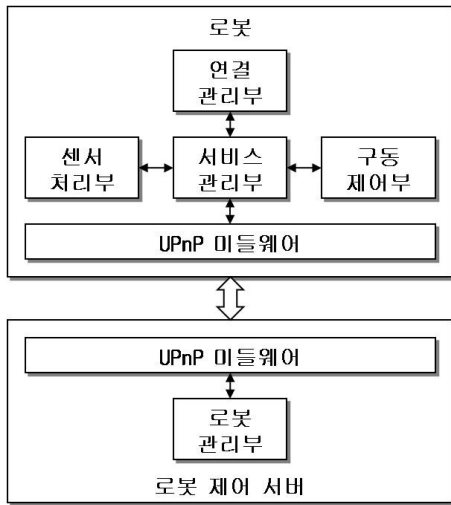


그림 3. UPnP 로봇 소프트웨어 구성
Fig. 3 UPnP Robot Software Architecture

3.1.1. 로봇 클라이언트 기술

로봇은 로봇 서비스 관리 및 제어 기술을 구현할 수 있도록 내부적으로 몇 개의 서브시스템으로 구성된다. 이것은 로봇이 수행하는 기본적인 서비스들을 관리하고 구동하는 서비스 관리부, 외부환경으로부터 정보를 획득하는 센서처리부, 또한 로봇을 동작시키는 구동 제어부, 그리고 다른 로봇 또는 서버로부터의 자료 전송을 담당하는 연결 관리부로 나누어진다.

3.1.2. 로봇 제어 서버 기술

이러한 지능형 서비스 로봇은 특정한 임무를 독자적으로 수행할 수도 있지만, 가까이 또는 멀리 떨어져 있는 원격 로봇 제어 서버와 협력하여 다양한 서비스를 제공할 수 있어야 한다. 로봇 제어 서버는 자체적으로 운영하는 서비스들과 서버에 연결된 로봇들의 정보를 관리하고 로봇에게 적절한 명령을 내리며, 반대로 로봇으로부터 오는 이벤트 및 데이터, 그리고 사용자의 요구입력을 처리할 수 있어야 한다.

3.1.3. 통신 매커니즘

로봇 서버가 원격으로 로봇을 제어하거나 반대로 로봇에서 서버 또는 다른 로봇으로 자료를 전송하기 위해서는 서로간의 통신 구조를 설계하여야 한다. 본 논문에서는 UPnP 프로토콜을 사용하여 서버 또는 로봇이 메시

지라는 자료 구조를 통해 서로 통신한다. 이러한 메시지는 전송되는 자료에 따라 다양한 형태를 지원하여야 한다. 메시지의 형태는 크기에 따라 기본적인 정수, 부동소수점 실수, boolean 등을 들 수 있고 문자열이나 배열과 같은 자료형도 가능하다. 또한 메시지는 전송 방식에 따라 다양한 종류로 구분할 수 있다. 본 논문에서 제안하는 로봇이 전송하는 메시지 종류는 다음과 같다.

표 1. UPnP 로봇의 주요 통신 메시지
Table. 1 Communication Messages for UPnP Robot

종류	동작	설명
제어 메시지	제어	서비스 호출 기능 (예, 모션 명령)
응답 메시지	응답	서비스 호출에 대한 응답
상태 메시지	이벤트처리	주기적/비주기적인 상태 전달 (예, 센서 데이터)
등시성 메시지	웹프로토콜 접근	대용량데이터 전송 (예, 카메라영상)

3.2. UPnP 로봇의 세부 기능

UPnP 로봇은 UPnP 구성 요소들인 컨트롤포인트, 디바이스, 서비스를 활용하여 구현되며, 기본적으로 디바이스와 서비스로 구성된다.

3.2.1. 로봇 명령 서비스

로봇 명령 서비스는 UPnP의 액션 호출 기능을 활용하여 구현할 수 있다. 예를 들면 로봇 서버가 “GoForward” 라는 액션 명령을 생성하여 UPnP 로봇으로 전송하면 로봇은 이러한 명령을 수행하게 된다. 이러한 기능은 ROS의 서비스 (srv) 형에 해당한다.

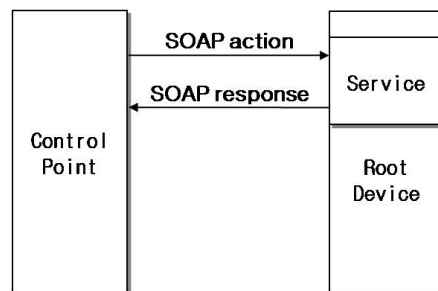


그림 4. 로봇 명령 서비스를 위한 통신 과정
Fig. 4 Communication for Robot Command Service

3.2.2. 센서 모니터링 서비스

센서 모니터링 서비스는 UPnP의 이벤트 알림 기능을 통하여 구현할 수 있다. 이벤트 알림 기능은 출판/구독 (publish/subscribe) 방식으로 서비스의 상태가 변하면 이벤트 메시지를 콜백함수에게 전달해 준다.

먼저 로봇 서버가 이벤트 알림 기능에 대한 구독 (subscription) 요청을 하면 로봇은 이를 받아들여 주기적 또는 비주기적으로 로봇의 센서 상태를 로봇 서버로 전송해 준다. 이것은 ROS의 메시지 (msg) 형에 해당한다.

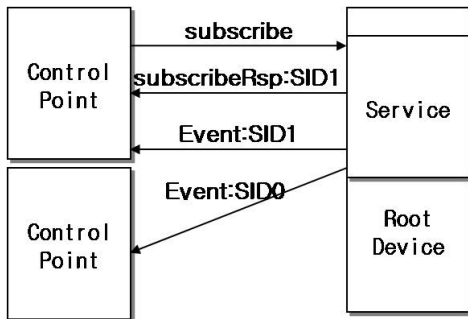


그림 5. 센서 모니터링 서비스를 위한 통신 과정
Fig. 5 Communication for Sensor Monitoring Service

3.2.3. 카메라 영상 감시 서비스

카메라 영상 감시 서비스는 UPnP의 AV 기능과 유사하게 구현할 수 있다. 카메라 영상과 같은 대용량의 미디어 데이터는 이벤트 구독/알림 기능을 통해 지속적으로 메시지를 자주 주고 받는 것은 비효율적이다.

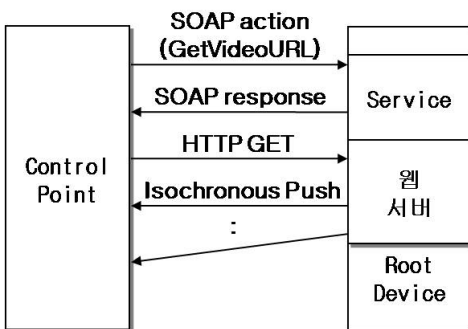


그림 6. 카메라 영상감시 서비스를 위한 통신 과정
Fig. 6 Communication for Cam Monitoring Service

따라서 UPnP AV 기능처럼 로봇 서버는 카메라 영상에 대한 미디어 콘텐츠의 웹 주소를 요청하고 제공받은 다음 이후에는 메시지 전송이 아니라 HTTP 웹 프로토콜을 통해 서버 푸시 미디어 스트리밍을 통해 영상 데이터를 전송받게 된다.

3.2.4. 웹 제어 기능

웹 제어 기능은 UPnP의 Presentation 기능을 활용하여 구현할 수 있다. 로봇 서버는 UPnP 메시지 프로토콜 대신에 로봇이 제공하는 웹 서버 기능을 이용하여 웹페이지로 접속하여 로봇 서비스를 제공받게 된다. 이것은 CGI와 Ajax 자바 스크립트의 웹 기술들을 활용하여 구현할 수 있다.

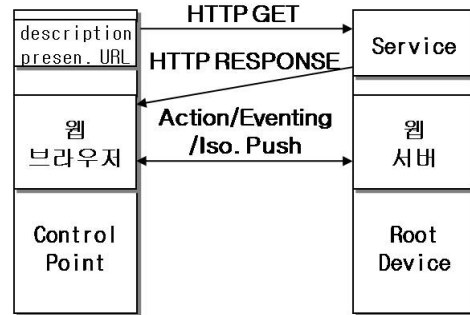


그림 7. 웹 제어 서비스를 위한 통신 과정
Fig. 7 Communication for Web Control Service

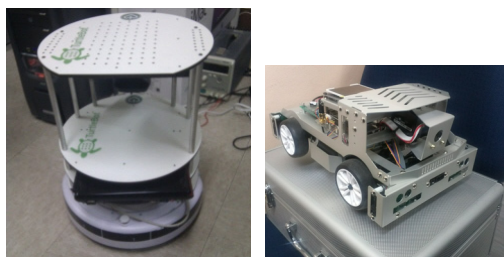
IV. 구현

본 논문에서는 다양한 UPnP 서비스를 탑재한 UPnP 로봇을 구현하고 이를 제어할 수 있는 스마트폰용 로봇 제어 소프트웨어를 구현하였다.

4.1. 하드웨어 구성

본 논문에서 설계한 UPnP 로봇 프레임워크는 다양한 로봇 플랫폼에 적용 가능하도록 설계되었다. 현재 UPnP 로봇 프레임워크는 Willow Garage 사의 Turtlebot 모바일 로봇과 하이버스사의 H-AICAR 지능형 자동차 로봇을 대상으로 구현되어 있다. Turtlebot은 일반 넷북형 노트북을 처리부, iCreate 사의 Roomba 로봇을 구동부, 그리고 마이크로소프트사의 Kinect 센서를 카메라 및 센서부

로 구성한다[9]. 반면, H-AiCAR 지능형 자동차 로봇은 PXA320 ARM 프로세서, 266MHz DDRSDRAM 128MB, Nand Flash 128MB 메모리를 탑재하고 있으며, CMOS 카메라, 적외선 센서, 전후측방 LED, DC 서보모터, 가속센서, 무선랜 등의 장치를 내장하고 있어 충돌방지, 전후 좌우 신호등 점멸, 경사감지 등의 동작을 수행할 수 있다 [10]. 다음 그림은 본 논문에서 적용한 UPnP 로봇 플랫폼들의 외양을 보여준다.



(a) (b)

그림 8. 실험에 사용된 로봇 플랫폼들

(a) 터틀봇 (b) H-AICAR

Fig. 8 Robot Platforms Used for Experiment

(a) Turtlebot (b) H-AICAR

4.2. 로봇 서버 소프트웨어

로봇에 내장된 소프트웨어로는 우분투 리눅스 또는 임베디드 리눅스와 로봇 서비스 프로그램을 탑재하였다. 로봇 서비스 프로그램을 구현하기 위해 리눅스용 오픈소스 UPnP SDK인 libupnp 라이브러리를 사용하였다. libupnp는 UPnP Device Architecture 표준 V1.0 호환 컨트롤 포인트, 디바이스, 브리지 등을 지원하는 API를 제공한다[11].

UPnP 서비스 프로그램은 총 3개의 기본 서비스로 구성되어 있는데, 로봇을 전후좌우로 움직이는 모션 기능과 로봇의 전조등, 브레이크등, 비상등 등을 켜고 끄는 LED 제어 기능을 포함한 로봇 명령 서비스, 로봇에 장착된 거리 센서 측정값을 출력하는 센서 모니터링 서비스, 로봇의 전면 카메라 영상을 실시간으로 전송하는 카메라 영상 감시 서비스이다. 또한, 기본 서비스 대신에 웹을 통해 제어할 수 있는 웹 제어 서비스도 포함하였다.

```
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <device>
    <deviceType>urn:schemas-upnp-org:device:RobotServer:1</deviceType>
    <UDN>uuid:898f9738-d930-....</UDN>
    <serviceList>
      <service>

<serviceType>urn:schemas-upnp-org:service:CommandController:1</serviceType>

<serviceId>urn:upnp-org:serviceId:CommandController</serviceId>
      <SCPDURL>/web/ccs.xml</SCPDURL>
      <controlURL>/web/ccs_control</controlURL>
      ...
      <presentationURL>/web/urobot.html</presentationURL>
      ...
    </serviceList>
  </device>
</root>
```

그림 9. UPnP 로봇 기술 XML 파일의 예

Fig. 9 An example of UPnP Robot Description XML

```
<?xml version="1.0" encoding="utf-8"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <actionList>
    <action>
      <name>GoForward</name>
      <argumentList>
        <argument>
          <name>Command</name>
          <direction>in</direction>
        </argument>
      </argumentList>
      <relatedStateVariable>A_ARG_TYPE_Command</relatedStateVariable>
    </action>
    ...
  </actionList>
  <serviceStateTable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_Command</name>
      <dataType>i4</dataType>
    </stateVariable>
    ...
  </serviceStateTable>
</scpd>
```

그림 10. UPnP 로봇 서비스 XML 파일의 예

Fig. 10 An example of UPnP Robot Service XML

그림 9와 10은 각각 UPnP 로봇의 기술 XML 파일과 서비스 XML 파일의 예를 보여준다. 기술XML 파일은 로봇 서버의 타입, UDN 번호, 서비스 목록, Presentation을 위한 URL 주소를 기록한다. 서비스 XML 파일은 서비스 액션의 이름, 인자 목록과 상태변수들의 목록을 담

고 있다.

4.3. 로봇 제어 소프트웨어

로봇 제어 프로그램은 안드로이드 스마트폰에 탑재하여 로봇을 제어하게 된다. 이것은 로봇과의 UPnP 통신을 위해 오픈소스 UPnP 자바 및 안드로이드 라이브러리인 **cling** 을 사용하여 구현하였다. **cling**은 UPnP 장치 구조 V1.0 표준을 구현한 코어 자바 라이브러리와 NAT 포트 매핑, A/V 미디어 서버 및 렌더러 등의 확장 기능을 구현한 지원 라이브러리로 구성되어 있다[11].

4.4. 실행 결과

본 논문에서 구현한 로봇 서버 프로그램을 로봇에 탑재하고 스마트폰에서 로봇 제어 프로그램을 실행하면 WiFi 무선랜에 연결된 UPnP 장치들이 검색되어 나타나고, 해당하는 UPnP 로봇을 선택하면 로봇 제어 화면이 나타난다. 로봇 제어 화면에서는 로봇에 특정한 명령을 내리거나 주기적인 센서 모니터링 및 실시간 영상 감시 등의 서비스를 동작시킬 수 있으며, 구글 음성인식 기능을 활용하여 명령을 내릴 수도 있다. 또한, 로봇 제어 프로그램 대신 웹 브라우저를 통해서도 로봇 서버에 접속하여 로봇을 제어할 수 있다. 다음 그림은 로봇 제어 프로그램의 실행 결과를 보여준다.

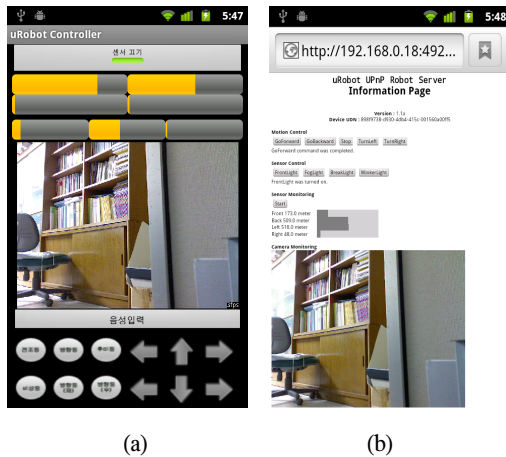


그림 11. 로봇 제어 프로그램 실행 화면
(a) UPnP 제어 화면 (b) 웹 제어 화면
Fig. 11 Screenshots for controlling robot
(a) UPnP Control (b) Web Control

다음 표는 무선 IP 공유기를 통해 로봇 서버와 제어 기간의 로봇 명령을 전달하고 응답을 수신하는 데 걸리는 응답 시간을 측정된 결과를 보여준다. 응답 시간은 매번 측정할 때마다 다른 측정값을 가지는데, 이것은 로봇 서버의 성능 및 전송 방식보다는 주로 네트워크 하위 계층과 전송 매체에 따르는 영향이 더욱 크다는 것을 의미한다.

표 2. 로봇 명령 서비스 응답 시간 (단위: 밀리초)
Table. 2 Response Time of Robot Command Service (unit: msec)

로봇 \ 방식	UPnP 제어	HTTP 웹 제어
H-AICAR	45 ~ 135	40 ~ 120
터틀봇	20 ~ 80	15 ~ 70

센서 모니터링 및 영상 감시 서비스는 주기적인 이벤트 발생 시에 전송하는 방식이므로 응답 시간 측정에는 큰 의미가 없지만, 카메라 영상 감시 서비스에서 320x240 및 640x480 트루컬러 JPG 영상을 WiFi 무선 통신을 사용해 전송할 경우, 제어 방식에 상관없이 각각 초 당 9~11 프레임 및 4~6 프레임 정도의 속도를 보여주었다.

끝으로 본 논문에서 구현한 로봇 서비스 및 제어 프로그램은 GPL 라이선스로 공개되어 비상업적 및 오픈소스로 자유롭게 활용할 수 있다[12].

V. 결 론

본 논문에서는 홈 네트워크 표준 기술인 UPnP를 사용하여 가정 등에서 많이 사용하는 지능형 서비스 로봇을 원격으로 제어하고 감시할 수 있는 로봇 제어 시스템을 개발하였다. 본 시스템은 오픈소스 소프트웨어로 공개하고 있고 실제 사용하는 모바일 로봇과 스마트폰 기기로 구성되어 다양한 플랫폼에 쉽게 적용할 수 있다는 점이 본 논문의 의의라 할 수 있다.

근래 외국 로봇 연구단체뿐만 아니라 국내에서도 음성인식, 영상인식 기술 등을 바탕으로 한 지능형 로봇 연구가 활발하게 이루어지고 있다. 이러한 개발 현황에 맞

추어 향후 보다 지능적인 로봇 서비스들과 동적 서비스 생성하는 기법을 개발할 계획이다.

참고문헌

- [1] H. Utz, S. Sablatng, S. Enderle, G. Kraetzschmar, "Miro-Middleware for Mobile Robot Applications," *IEEE Trans. on Robotics and Automation*, vol.18, no.4, Aug. 2002.
- [2] A. Makarenko, A. Brooks, and T. Kaupp, "Orca: Components for Robotics," *IROS*, pp. 163-168, Oct. 2006.
- [3] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W. Yoon, "RTMiddleware: Distributed Component Middleware for RT ", *IROS*, pp. 3555-3560, Aug. 2006.
- [4] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Nergler, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System", *Proceedings of the Open-Source Software workshop at ICRA*, 2009.
- [5] S. Ahn, J. Lee, K. Lim, H. Ko, Y. Kwon, and H. Kim, "Requirements to UPnP for Robot Middleware," *IROS*, Aug. 2006.
- [6] S. Ahn, K. Lim, J. Lee, H. Ko, Y. Kwon and H. Kim, "UPnP Robot Middleware for Ubiquitous Robot Control," *3rd URAI*, 2006.
- [7] UPnP 공식 포럼, <http://www.upnp.org>
- [8] Turtlebot 홈페이지, <http://www.turtlebot.com>
- [9] 하이버스 홈페이지, <http://www.hybus.net>
- [10] libupnp 홈페이지, <http://pupnp.sourceforge.net>
- [11] cling 홈페이지, <http://tealeal.org/projects/cling/>
- [12] uRobot 홈페이지, <http://code.google.com/p/urobot/>

저자소개



김성우(Seong-Woo Kim)

1991년 한국과학기술원
전기및전자공학과(공학사)
1993년 한국과학기술원
전기및전자공학과(공학석사)

1999년 한국과학기술원 전기및전자공학과(공학박사)
1999년 ~ 2002년 한국전자통신연구원 선임연구원
2002년 ~ 현재 동의대학교 컴퓨터소프트웨어공학과
부교수

※ 관심분야: 지능 로봇 제어, 센서 네트워크



박유현 (Yoo-Hyun Park)

1996년 부산대학교
전자계산학과(이학사)
1998년 부산대학교
전자계산학과(이학석사)

2008년 부산대학교 전자계산학과(이학박사)
2000년 ~ 2000년 한국국방연구원(KIDA) 연구원
2001년 ~ 2009년 한국전자통신연구원 선임연구원
2009년 ~ 현재 동의대학교 컴퓨터소프트웨어공학과
조교수

※ 관심분야: 인터넷 시스템, 클라우드 시스템, 융합IT 서비스



권순각 (Soon-kak Kwon)

1990년 경북대학교
전자공학과 (공학사)
1992년 한국과학기술원
전기및전자공학과(공학석사)

1998년 한국과학기술원 전기및전자공학과(공학박사)
1997년 ~ 1998년 한국전자통신연구원 연구원
1998년 ~ 2001년 기술보증기금 팀장
2001년 ~ 현재 동의대학교 컴퓨터소프트웨어공학과
교수

※ 관심분야: 멀티미디어신호처리, 영상통신