

# 스마트폰을 이용한 로봇 어플리케이션의 OPRoS 기반 개발 방법에 대한 연구

이경수\*, 강동병\*, 지상훈\*\*, 정구민\*

\*국민대학교 전자공학과

\*\* 한국 생산 기술 연구원

e-mail : gm1004@kookmin.ac.kr

## A Robot application design using a smartphone based on OPRoS

Kyung-Su Lee\*, Dong-Byoung Kang\*, Sang-Hoon Ji\*\*, Gu-Min Jeong\*

\*Dept. of Electronics Engineering Kook-Min University

\*\*Korea Institute of Industrial Technology

### 요 약

본 논문에서는 스마트폰을 기반으로 로봇을 원격 제어하는 어플리케이션을 OPRoS를 활용한 개발 방법을 제안하고 스마트폰과 OPRoS기반의 로봇으로 구현한다. 컴포넌트 기반의 로봇용 미들웨어인 OPRoS를 활용하면 이미 개발 되었거나 공개된 컴포넌트를 활용하여 개발의 시간 및 비용을 절감할 수 있는 장점이 있다. 이미 검증된 모터, 음성, TCP/UDP, 카메라 컴포넌트를 재활용하였다. 또한 오픈로스의 통합된 개발환경을 통해 개발 시간을 더욱 단축시켰다. 실제 구현에서는 Wi-Fi기반의 TCP, UDP 컴포넌트를 활용하여 로봇과 스마트폰간 송수신되는 데이터의 목적에 따른 적절한 통신 방식을 사용하였다. 또한 이미지 전송, 음악 재생, 메시지 전송, 음성 인식 등의 기능을 활용하여 구현된 시스템을 검증 하였다.

### 1. 서론

로봇은 IT기술의 집합체이며 미래의 인간을 대체할 수단으로 각광받고 있다. 이러한 로봇 산업이 발전하면서 제조업, 농업, 의료, 개인 서비스용, 전문 서비스용 등의 다양한 분야에서 로봇이 활용되고 있다. 이러한 로봇의 광범위한 활용으로 로봇 소프트웨어의 다양한 디바이스 프로그램과 알고리즘이 필요하게 된다. 또한 지적인 서비스를 제공하기 위해서 로봇 소프트웨어는 더욱더 복잡해지고 있다[1].

이러한 현실적인 문제점을 해결하기 위해서 컴포넌트 기반의 로봇 소프트웨어 플랫폼은 매우 효과적이다. 컴포넌트 기반의 어플리케이션 개발은 이미 구현된 코드 및 프로그램을 재사용할 수 있다. 컴포넌트 기반의 코드 재사용 방법은 어플리케이션을 만들 때 개발자의 시간과 비용을 절감할 수 있는 장점이 있다. 최근 ROS, OROCOS, MSRDS, OPRoS 등의 컴포넌트 기반 로봇 SW 플랫폼이 연구되고 있다. 본 논문에서는 컴포넌트 기반 로봇 SW 플랫폼 중의 하나인 OPRoS(Open Platform for Robotic Services)을 활용 하였다. 이는 통합된 개발 환경을 제공함으로써 컴포넌트 기반의 어플리케이션 개발의 장점을 더 부각시켰다. 통합된 개발환경은 컴포넌트 개발과 어플리케이션 개발을 하나의 틀에서 가능하다. 또한 어플리케이션 개발 시 컴포넌트의 포트를 활용하여 구성되며 이는

기존 컴포넌트들을 활용할 때도 쉽게 재사용이 가능하다.

한편 스마트폰은 HW의 성능뿐만 아니라 OS의 성능 또한 계속 발전하면서 다양한 통신 모듈, 센서, 고 해상도 디스플레이 기능을 탑재하여 최근에는 로봇, 자동차 등을 제어하기 위한 인터페이스로[2]-[4]씩 각광 받고 있다.

본 논문에서는 스마트폰을 기반으로 로봇을 제어하는 어플리케이션을 개발할 때 OPRoS를 활용한 방법을 제안하고 스마트폰과 OPRoS기반의 로봇으로 구현한다. OPRoS 로봇 어플리케이션은 기존에 공개된 컴포넌트들을 재사용하여 개발 시간을 단축 시켰고 각 컴포넌트들의 포트와 연결되어 어플리케이션 전체를 컨트롤하는 컨트롤 컴포넌트로 구성되었다. 스마트폰과 로봇의 통신은 Wi-Fi기반의 TCP, UDP 컴포넌트를 활용하여 데이터의 목적에 따른 적절한 통신 방식을 사용한다. 이미지 전송, 음악 재생, 메시지 전송, 음성 인식 등의 기능을 활용하여 구현된 시스템을 검증한다.

### 2. 스마트폰을 이용한 로봇제어

최근 몇 년간 스마트폰을 통한 로봇을 제어하는 연구가 활발히 이루어지고 있다. 기존 연구에 대해 살펴보면, [2]에서는 안드로이드 스마트폰을 활용하여 사용자가 자율적인 모바일 로봇을 제어하는 방법을 제안하였다. [3]에서는 스마트폰과 PC를 이용한 일 대 다 통신을 위한 비디오

회의 로봇 설계를 제안하였다. 사용자가 Wi-Fi를 통하여 스마트폰으로 로봇을 제어하는 경우 어디서든지 이를 사용할 수 있다.



(그림 1) 비디오 회의 로봇

[4]에서는 Walky 로봇을 스마트폰 화면 위에 손가락 움직임 인식을 통해 제어하는 방법을 제안하였다.



(그림 2) Walky 로봇

스마트폰은 Bluetooth, Wi-Fi, NFC 등 근거리 네트워크를 활용하여 다양한 방식으로 다양한 디바이스와 연동되어 인터페이스 역할의 연구가 진행되고 있다. 스마트폰은 SW와 HW가 다양한 방향으로 발전하고 있고 이는 스마트폰을 활용하여 어플리케이션을 즐기고 활용함에 그치지 않고 센서 및 통신 기능을 활용하여 여러 분야에 걸쳐 사용되게 될 전망이다.

### 3. OPRoS와 OPRoS 개발환경

#### 3.1 OPRoS

OPRoS는 Open Platform for Robotic Services의 약자로써 지식 경제부 산하의 현재 개발 중인 컴포넌트 기반의 로봇 SW플랫폼이다. OPRoS는 여러 종류의 로봇 시스템들의 응용 프로그램들을 동작 시킬 수 있는 개방형 SW 구조이다. 또한 OPRoS는 컴포넌트 기반의 플랫폼으로 코드의 재활용성이 높아 OPRoS기반의 시스템 개발할 때 기간의 단축과 비용 절감의 효과를 낼 수 있다.

OPRoS가 추구하는 목적은 로봇용 플랫폼과 통합된 개발환경을 제공하여 다양한 콘텐츠를 활용할 수 있도록 하는 데에 있다[5]. 통합된 개발환경은 개발자가 편리하게 로봇 어플리케이션을 개발할 수 있도록 지원한다.

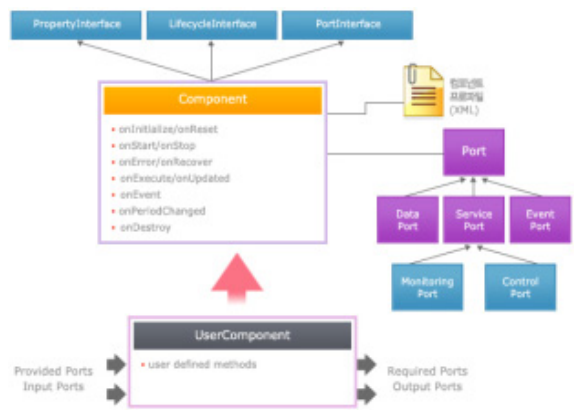
#### 3.2 OPRoS 컴포넌트

컴포넌트는 소프트웨어 개발을 블록을 쌓듯이 쉽게 할 수 있도록 하는 기술을 말한다. 즉, 소프트웨어 구성단위(module)를 미리 만든 뒤 필요한 응용 기술을 개발할 때 이 모듈을 조립하는 기술을 말한다[5]. OPRoS 컴포넌트는

외부와 통신을 위한 인터페이스를 가지고 있으며, 포트를 통해서 외부 컴포넌트와 인터페이스가 이루어진다[5].

그림 3은 OPRoS 컴포넌트의 class diagram이다. 하나의 컴포넌트는 11개의 콜백 함수를 가지며 데이터의 성격에 따라 세 가지 포트(Service Port, Data Port, Event Port)를 가진다. 각각의 포트는 모두 입력 과 출력 포트를 구성할 수 있으며 하나의 컴포넌트가 출력이면 다른 한쪽은 반드시 입력 포트를 가져야 한다. 포트들은 컴포넌트 간 원활한 상호작용을 위해 주고받는 데이터 타입을 강제 정의 하여야한다.

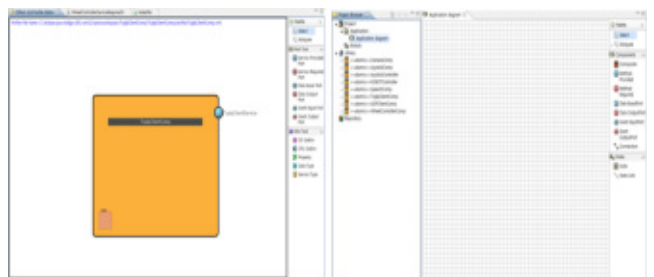
컴포넌트는 6개의 상태(Create, Ready, Active, Inactive, Error, Destroyed)를 가지며 컴포넌트의 초기 설정 값에 따라 실행 상태가 결정 된다. 실행 엔진은 컴포넌트의 상태에 따라 콜백 함수를 수행하여 컴포넌트를 동작 시킨다.



(그림 3) OPRoS 컴포넌트의 class diagram

### 3.3 OPRoS 개발 환경

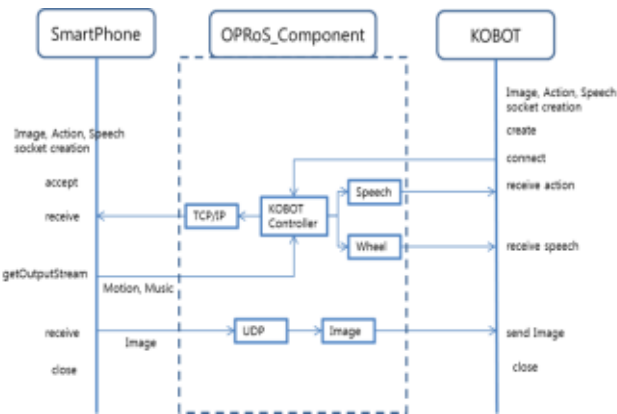
OPRoS는 통합된 개발환경을 제공한다. 통합된 개발환경은 소프트웨어의 개발 기간 단축, 유지 보수비용 절감 등의 효과를 줄 수 있고 시뮬레이션 기능을 제공하기 때문에 로봇을 직접 구매하지 않아도 개발할 수 있다는 장점이 있다. 또한 저작도구->시뮬레이션->로봇 순으로 어플리케이션을 개발하는데 체계화된 시스템을 가지고 있다. OPRoS의 개발환경은 이클립스의 플러그인 형태로 제공되며 개발 언어는 C/C++ 기반이다. 통합된 개발환경으로는 컴포넌트 저작도구, 컴포넌트 컴포우저, 테스트 개발도구, 시뮬레이터등이 있다[6]. 컴포넌트 저작도구와 컴포우저는 간단한 드래그 앤 드롭을 통해 컴포넌트의 포트 추가 및 컴포넌트들을 연결하여 어플리케이션을 구현할 수 있다.



(그림 4) OPRoS Component Editor/Composer

#### 4. 전체 시스템 구성

전체적인 시스템 구조는 그림 5와 같다. 스마트폰이 서버역할, 로봇은 클라이언트 역할을 하도록 시스템을 구성한다. 먼저 사용자가 스마트폰의 어플리케이션을 실행시켜 로봇의 접속을 기다리고 OPRoS 어플리케이션을 올린 로봇의 엔진을 실행 시키면 로봇과 스마트폰이 연결된다. 사용자가 스마트폰을 이용해 로봇의 움직임 제어, 이미지 전송, 메시지 전송 후 음성 출력, 음악 재생 등의 작업을 할 수 있다. 스마트폰과 로봇간의 네트워크는 Wi-Fi기반의 TCP, UDP 컴포넌트를 활용하여 로봇과 스마트폰간 송수신되는 데이터의 목적에 따른 적절한 통신 방식을 사용한다.



(그림 5) 시스템 구조 설계

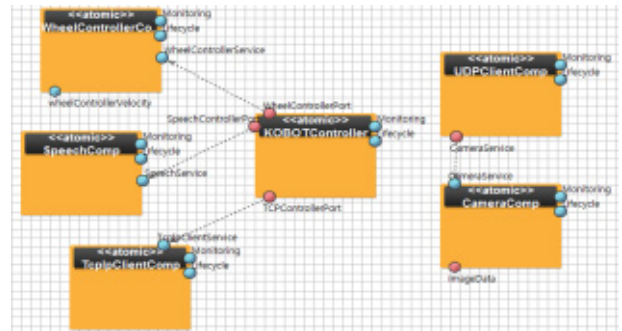
#### 5. OPRoS 원격제어 어플리케이션 구현

OPRoS 어플리케이션은 로봇에 OPRoS 엔진을 올리게 되면 쉽게 사용할 수 있다. 이 원격 제어 어플리케이션은 다른 기기를 이용한 원격제어가 가능하다. OPRoS 어플리케이션의 구성은 그림 7과 같다. Wheel, Speech, Camera, TCP/IP, UDP/IP, Controller 컴포넌트들을 이용해 어플리케이션을 구현하였다. Controller 컴포넌트는 연결될 컴포넌트의 데이터 속성에 맞게 Port를 만들고 구현할 동작에 대한 코드를 작성하여 구현 하였다. 그 이외에 컴포넌트들은 OPRoS 공식 홈페이지에 있는 기존 컴포넌트들을 재사용하였고 OPRoS API를 통해 필요한 부분만 수정하였다. OPRoS에서 제공되는 컴포넌트들은 그림 6의 홈페이지를 통해 확인할 수 있다.

Name	Desc	Source
TransportComp, v1.0.0.002	TransportComp, v1.0.0.002, Source, de	TransportComp, v1.0.0.002, Source, de
RobotTransportComp, v1.0.0.002	RobotTransportComp, v1.0.0.002, Source, de	RobotTransportComp, v1.0.0.002, Source, de
TransportClientComp, v1.0.0.002	TransportClientComp, v1.0.0.002, Source, de	TransportClientComp, v1.0.0.002, Source, de
TransportServerComp, v1.0.0.002	TransportServerComp, v1.0.0.002, Source, de	TransportServerComp, v1.0.0.002, Source, de
TransportComp, v1.0.0.002	TransportComp, v1.0.0.002, Source, de	TransportComp, v1.0.0.002, Source, de
WheelControllerComp, v1.0.0.002	WheelControllerComp, v1.0.0.002, Source, de	WheelControllerComp, v1.0.0.002, Source, de
WheelControllerClientComp, v1.0.0.002	WheelControllerClientComp, v1.0.0.002, Source, de	WheelControllerClientComp, v1.0.0.002, Source, de
WheelControllerServerComp, v1.0.0.002	WheelControllerServerComp, v1.0.0.002, Source, de	WheelControllerServerComp, v1.0.0.002, Source, de
WheelControllerComp, v1.0.0.002	WheelControllerComp, v1.0.0.002, Source, de	WheelControllerComp, v1.0.0.002, Source, de
SpeechComp, v1.0.0.002	SpeechComp, v1.0.0.002, Source, de	SpeechComp, v1.0.0.002, Source, de
SpeechClientComp, v1.0.0.002	SpeechClientComp, v1.0.0.002, Source, de	SpeechClientComp, v1.0.0.002, Source, de
SpeechServerComp, v1.0.0.002	SpeechServerComp, v1.0.0.002, Source, de	SpeechServerComp, v1.0.0.002, Source, de
SpeechComp, v1.0.0.002	SpeechComp, v1.0.0.002, Source, de	SpeechComp, v1.0.0.002, Source, de
CameraComp, v1.0.0.002	CameraComp, v1.0.0.002, Source, de	CameraComp, v1.0.0.002, Source, de
CameraClientComp, v1.0.0.002	CameraClientComp, v1.0.0.002, Source, de	CameraClientComp, v1.0.0.002, Source, de
CameraServerComp, v1.0.0.002	CameraServerComp, v1.0.0.002, Source, de	CameraServerComp, v1.0.0.002, Source, de
CameraComp, v1.0.0.002	CameraComp, v1.0.0.002, Source, de	CameraComp, v1.0.0.002, Source, de
ImageComp, v1.0.0.002	ImageComp, v1.0.0.002, Source, de	ImageComp, v1.0.0.002, Source, de
ImageClientComp, v1.0.0.002	ImageClientComp, v1.0.0.002, Source, de	ImageClientComp, v1.0.0.002, Source, de
ImageServerComp, v1.0.0.002	ImageServerComp, v1.0.0.002, Source, de	ImageServerComp, v1.0.0.002, Source, de
ImageComp, v1.0.0.002	ImageComp, v1.0.0.002, Source, de	ImageComp, v1.0.0.002, Source, de

(그림 6) OPRoS 컴포넌트  
(<http://203.255.255.39/new/>)

OPRoS 컴포넌트들은 서로 Port를 통해서 연결한다. 스마트폰과의 통신을 위해 UDP와 TCP 컴포넌트를 구성하였다. UDP 컴포넌트는 Camera 컴포넌트와 Service port를 통해 연결되고 카메라의 데이터를 스마트폰으로 전송하는 역할을 한다. UDP 컴포넌트를 사용한 목적은 이미지 데이터의 빠른 전송을 통해 영상이 보이는 효과를 내기 위해서다. 그 이외에 데이터 전송은 신뢰성을 높이기 위해서 Controller 컴포넌트와 Service port를 통해 연결된 TCP 컴포넌트를 사용하였다. 통신을 할 때 연결할 IP와 Port는 UDP, TCP/IP 컴포넌트 각각의 xml 파일에 설정해 준다. Speech 컴포넌트는 Controller 컴포넌트와 Service port를 통해 연결되며 음성을 출력하는 역할을 한다. Wheel 컴포넌트 역시 컨트롤러 컴포넌트와 Service port를 통해 연결되며 바퀴를 구동 시키는 역할을 한다. 이와 같이 구현한 어플리케이션은 OPRoS 실행 엔진을 통하여 동작 시킬 수 있다.



(그림 7) OPRoS 어플리케이션구성

#### 6. 스마트폰 어플리케이션 구현

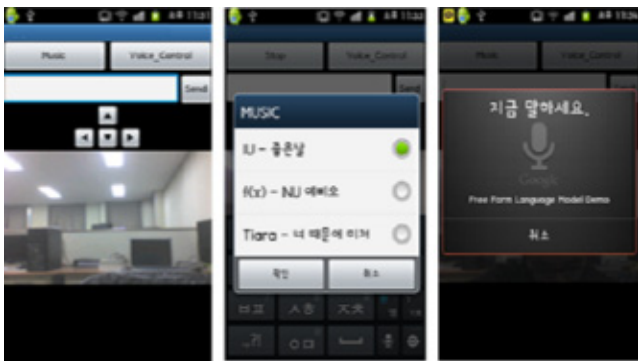
본 논문에서는 OPRoS 기반의 로봇을 안드로이드 스마트폰으로 제어하였다. 스마트폰의 화면은 액티비티 한 개로 구성되며 레이아웃의 배치는 버튼, 이미지, EditText를 이용 하였다. 로봇의 움직임 제어는 버튼의 클릭 리스너를 통해 전진, 좌회전, 우회전, 정지의 기능을 구현하였다. 스마트폰에서 송신된 이미지 데이터 출력 시 바이트 순서에 유의해야 한다. 자바는 Big Endian을 사용하고 C/C++은 Little Endian을 사용하기 때문이다. 이러한 이유로 스마트폰에서 데이터를 받을 때 바이트를 뒤집어 계산하는 함수를 구현하였다.

음악 재생 목록 화면은 Dialog 창을 이용해 구현하였다. 음악을 선택하게 되면 로봇에서 해당 음악이 재생되는 걸 확인할 수 있다. 음성 인식 기능은 로봇의 움직임을 제어할 수 있는 몇 가지 명령어를 문자열로 저장하였다. 음성을 인식하면 유사한 문자열들이 리스트로 저장되고 이중 가장 유사한 첫 번째 인덱스에 저장된 문자열을 선택하여 미리 저장된 명령어와 비교하는 형식으로 구현하였다. 이는 구글 API를 참조하여 구현하였다.

## 7. 실험

실험 기기는 Samsung社의 SHW-M110S 와 데모 로봇으로 KOBOT을 사용하였다. 실험환경은 Wi-Fi AP에 KOBOT과 스마트폰이 접속한 상태에서 실험했다. 데모 로봇인 KOBOT은 노트북과 연결된다. 노트북은 머리, KOBOT은 몸체의 역할을 담당한다.

그림 8은 Android application의 실행 화면이다. 상단에 Music, Voice control 버튼이 있다. 그 밑으로 메시지를 보낼 수 있는 편집 창이 있고 그 밑으로는 움직임을 제어할 수 있는 방향 버튼이 있다. 마지막 부분으로 KOBOT의 카메라를 통해 전송된 이미지를 출력하는 부분이다. 그림 9는 KOBOT과 연결된 노트북에서의 OPRoS 엔진 실행 화면이다. 스마트폰에서 보낸 메시지가 KOBOT에서 음성으로 출력된다. 또한 스마트폰을 이용해 원격으로 방향을 제어할 때 마다 KOBOT의 움직임과 동시에 엔진 화면에 움직이는 방향이 출력되는 걸 확인할 수 있다.



(그림 8) Android 어플리케이션 실행화면



(그림 9) KOBOT 및 엔진실행화면

## 8. 결론

본 논문에서는 스마트폰을 기반으로 로봇을 제어하는 어플리케이션을 개발할 때 OPRoS를 활용하는 방법을 제안하고 스마트폰과 OPRoS기반의 로봇으로 구현 하였다. 또한 이미지 전송, 음악 재생, 메시지 전송, 음성인식 등의 기능을 넣어 구현된 시스템을 검증 하였다.

논문에서 구현한 OPRoS 기반의 로봇 어플리케이션은 다른 로봇에도 쉽게 적용 시킬 수 있다. 또한 컴포넌트의 재활용을 통한 로봇 어플리케이션 구현은 소프트웨어 개발 기간의 단축, 비용의 절감이라는 장점을 가질 수 있다.

향후 연구 과제로는 여러 로봇에 OPRoS 기반의 어플리케이션을 적용할 것이다. 또한 스마트폰과 연동된 로봇의

컨텐츠 요소를 첨가해 지적인 서비스를 제공할 수 있도록 구현할 것이다.

## Acknowledgements

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 육성지원 사업의 연구결과로 수행되었음 (NIPA-2012-H0301-12-2007)

## 참고문헌

- [1] Seungwoog Jung, Byoungyoul Song, Choulsoo Jang, Sunghoon kim and Michael E.Shin, Development and Execution of OPRoS Components, 11th International Conference on Control, pp 1420-1423, October 2011
- [2] Jong-Yun Yeo, Lang Bach Truong, Sung-Hun Shin, Dong-Byeong Kang, Sang-Hak Lee, An Autonomous Control of Mobile Robot using Android Smartphone and X-BOT, Korea Multimedia Society, MITA, pp 62-65 2011.7
- [3] Dong-Won Choi, Tae-Houng Song, Soon-Mook Jeong and Jae Wook Jeon, Design the Video Conferencing Robot for One-to-Many Communication using Smartphone, 11th International Conference on Control, pp 671-675, October 2011
- [4] Walky Robot  
<http://www.designinterface.jp/en/projects/Walky/>
- [5] OPRoS [www.opros.or.kr](http://www.opros.or.kr)
- [6] Choulsoo Jang, Seung-ik Lee, Seung-Woo Jung, Byoungyoul Song, Rockwon Kim, Sunghoon Kim, and Cheol-Hoon Lee, OPRoS: A New Component-Based Robot Software Platform, ETRI Journal, Volume 32, Number 5, October 2010