

ARTIK 플랫폼을 활용한 IoTivity 기반 IoT 서비스 시나리오 설계 및 구현

최재완⁰ 김병선 조진성
경희대학교 컴퓨터공학과

jaewan@khu.ac.kr, ykbs0903@khu.ac.kr, chojs@khu.ac.kr

Design and Implementation of IoTivity-based IoT Service Scenario using ARTIK Platform

Jaewan Choi⁰ Byoungseon Kim Jinsung Cho
Department of Computer Science and Engineering
KyungHee University

요 약

IoT 시장 주도권 확보를 위해 많은 국내외 기업들은 IoT 플랫폼 경쟁을 진행하고 있다. 최근 삼성전자는 IoT 통합 플랫폼인 ARTIK 을 출시하였으며, ARTIK 생태계 형성을 위한 다양한 시도를 전개하고 있다. 본 논문에서는 삼성전자의 ARTIK 플랫폼을 활용하여 IoTivity 기반의 IoT 서비스 시나리오를 설계 및 구현한다. 구현된 시나리오를 기반으로 IoTivity 호환성 및 IoT 기능 동작을 검증한다.

1. 서 론

최근 IoT 시장의 주도권 확보를 위해 많은 국내외 기업들은 디바이스 플랫폼 경쟁을 진행하고 있다. 이에 대표적인 예로 삼성전자는 모듈, 클라우드, 에코 시스템까지 흡수 가능한 개방형 IoT 플랫폼인 ARTIK을 출시하였다.

삼성전자는 향후 스마트홈, 스마트빌딩, 헬스케어 등 다양한 IoT 분야에 ARTIK 생태계 형성을 목표로 모든 생산 제품에 ARTIK을 지원할 예정이다. 이를 위해 ARTIK 플랫폼은 IoT 환경에서 안정된 서비스 운영을 보장하기 위한 IoT 소프트웨어 플랫폼과의 호환성 및 다양한 IoT 기능 테스트가 필수적으로 선행되어야 한다.

한편 IoT 소프트웨어 플랫폼으로는 대표적으로 OCF(Open Connectivity Foundation)의 IoTivity가 있다. IoTivity는 디바이스간 상호 운용성 보장을 위한 오픈소스 소프트웨어 기술로써 이를 적용한 IoT 디바이스는 네트워크 프로토콜에 상관없이 다른 IoT 디바이스와 서로 연결될 수 있고, 개발자는 본 플랫폼을 통해 쉽고 빠른 IoT 제품 개발이 가능하다.

본 논문에서는 ARTIK 플랫폼을 활용한 IoTivity 기반의 IoT 서비스 시나리오를 설계 및 구현하였다. 구현된 시나리오를 기반으로 ARTIK 플랫폼의 IoTivity 소프트웨어 호환성을 확인하고 범용 IoT 기능에 대한 동작 수행을 검증한다.

목차는 다음과 같다. 2장에서는 ARTIK 플랫폼 및 IoTivity 아키텍처에 대한 분석을 수행한다. 분석 결과를 기반으로

3장에서는 IoT 서비스 시나리오를 설계하고, 구현 결과를 4장에서 다루도록 한다. 마지막으로, 5장에서는 결론을 맺는다.

2. 관련 연구

본 장에서는 IoT 서비스 시나리오 환경을 구성하기 위한 ARTIK 플랫폼 및 IoTivity 아키텍처에 대해 설명한다.

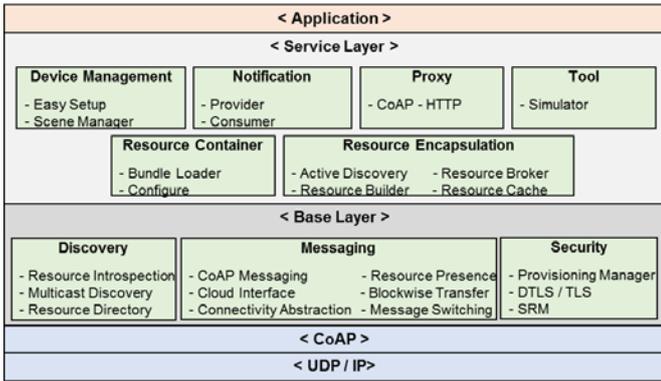
2.1 ARTIK 디바이스 플랫폼

삼성전자는 2016년, IoT 생태계 주도권을 잡기 위해 ARTIK을 출시하였다. ARTIK은 삼성전자에서 개발한 프로세서, 메모리, 통신칩, 센서 등으로 구성된 IoT 모듈로써 소프트웨어, 스토리지, 보안, 클라우드 등의 다양한 기능을 종합적으로 지원하는 통합 플랫폼이다. 이러한 ARTIK은 각종 IoT 분야에 모두 적용될 수 있도록 시리즈 형태로 개발되고 있으며,

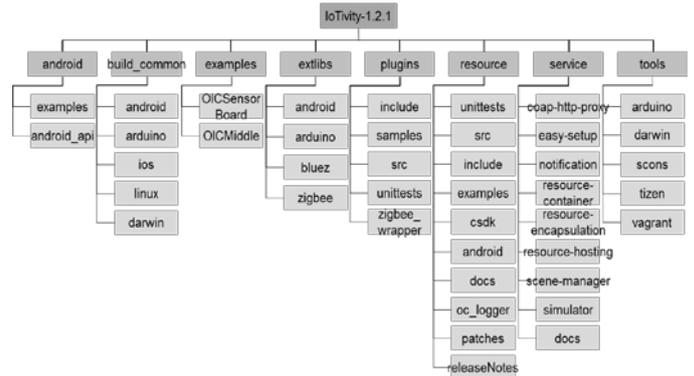
	ARTIK-0	ARTIK -5	ARTIK -7	ARTIK -10
CPU	ARM Cortex-M4	ARM Cortex-A7	ARM Coretex-A53	ARM Cortex-A15, Cortex-A7
Flash	256KB	4GB	4GB	16GB
RAM	32KB	512MB LPDDR3	1GB DDR3	2GB LPDDR3
활용예시	센서, 조명	조명 컨트롤러	스마트 홈 게이트웨이	드론, 이미지처리

[표 1] ARTIK 특징 및 활용 분야

⁰본 연구는 삼성전자의 지원으로 수행되었음.



[그림 1] IoTivity 아키텍처



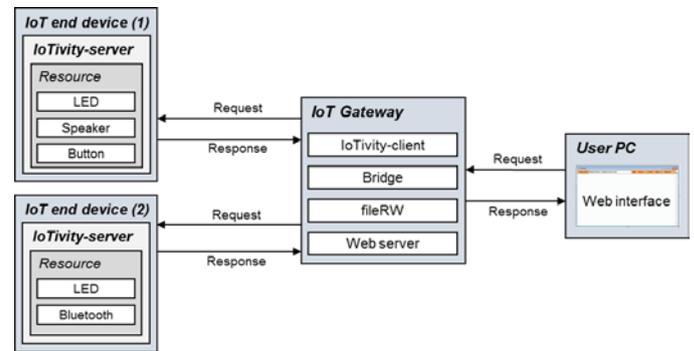
[그림 2] IoTivity-1.2.1의 디렉토리 구조

애플리케이션 별로 다양한 범위의 모듈이 존재한다. [표 1]은 ARTIK 모듈에 대한 하드웨어 특징 및 활용 분야를 나타낸다. 개발자들은 이러한 ARTIK 플랫폼을 통해 IoT 제품과 서비스를 개발하는데 시간·비용을 절감할 수 있다.

2.2 IoTivity 오픈소스 프레임워크

OCF 표준 기반의 IoTivity는 IoT 디바이스 간 상호 운용성을 보장하기 위한 오픈소스 프레임워크 기술이다. 본 기술은 IoT 환경의 모든 사물을 리소스화 하고, 각 리소스에 대한 관리·제어 기능을 제공한다. [그림 1], [그림 2]는 각각 IoTivity 1.2.1의 아키텍처 및 디렉토리 구조를 나타낸다. 아키텍처를 구성하는 각 컴포넌트에 대한 자세한 설명은 아래와 같다.

- **Base Layer:** 리소스를 관리, 제어하기 위한 각종 기능을 제공한다. 먼저, Discovery 컴포넌트는 IoTivity 서버 디바이스에 등록된 리소스를 탐색한다. Messaging은 CoAP 기반의 다양한 통신 기능(예., WiFi, Bluetooth 등의 통신 프로토콜 추상화, 메시지 스위칭, 대용량 메시지 전송을 위한 메시지 블록 전송)을 제공한다. 이러한 기능 구현은 IoTivity-1.2.1\resource 디렉토리에 구현되어 있으며, 이를 활용한 각종 예제 소스가 함께 포함되어 있다.
- **Service Layer:** IoT 서비스 개발 시, Base Layer 기능 기반의 다양한 확장 기능을 제공하는 여러 컴포넌트로 이루어진다. 한다. “Proxy”는 http 서버와 IoTivity 환경 간의 통신이 가능하도록 하는 도구이고, “Device Management”는 IoTivity 환경을 처음 사용하게 될 디바이스의 속성 값을 세팅하고 모바일 기기와의 IoTivity 네트워크 환경을 원활하게 연결하는 기능을 한다. “Notification”은 리소스의 상태 변화를 모니터링하여 속성 값이 변경된 경우 알림을 수신 할 수 있도록 한다. “Tool”은 가상 IoTivity 서버/클라이언트 시뮬레이션 기능이 있고, “Resource Encapsulation”은 리소스 관리를 용이하게 하는 기능을 제공한다. “Resource Container”는 IoTivity의 통신 방식을 사용하지 않는 프로토콜을 사용하는 리소스들을 IoTivity를 사용하는 클라이언트에서 제어할 수 있도록 돕는다.



[그림 3] IoT 서비스 환경 구성

3. 제안하는 시스템

[그림 3]은 ARTIK-10을 활용한 IoTivity 기반 IoT 서비스 시나리오 설계 결과를 나타낸다. 시나리오 환경은 크게 IoT 엔드 디바이스, IoT 게이트웨이, 사용자 PC로 구성된다. 각 디바이스의 핵심 기능 및 세부 동작은 아래와 같다.

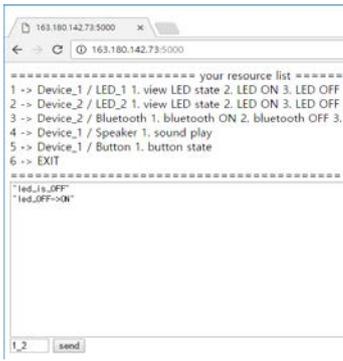
3.1 IoT 엔드 디바이스

2대의 IoT 엔드 디바이스에는 IoTivity 서버가 동작하며 IoTivity 서버는 범용 IoT 기능으로 활용될 수 있는 스피커, 버튼, 블루투스, LED를 리소스로 갖는다. 리소스 제어 요청을 처리하고, 이에 대한 응답을 반환한다.

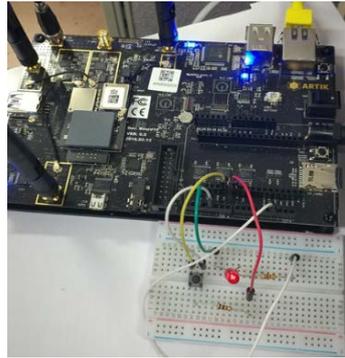
3.2 IoT 게이트웨이

사용자 PC측 리소스 제어 요청을 IoT 엔드 디바이스에 전달하고, 요청에 대한 응답을 사용자 PC로 다시 반환한다. 이러한 기능 수행을 위해 본 디바이스는 IoTivity 클라이언트, bridge, fileRW, 웹 서버 데몬을 포함하며, 각 데몬에 대한 자세한 설명은 아래와 같다.

- **웹 서버:** IoT 엔드 디바이스 리소스를 제어할 수 있는 웹 인터페이스를 제공한다. 웹 인터페이스로부터 요청이 들어오면 fileRW 데몬에 전달할 메시지를 파일로 저장하여



[그림 4-1] 웹 인터페이스



[그림 4-2] 디바이스 수행 결과

[그림 4] 웹 브라우저를 이용한 리소스 제어

메시지를 전달한다. 반대로, fileRW 데몬이 웹 서버에 메시지 전달을 위한 파일을 생성하면 웹 서버는 fileRW 데몬이 생성한 파일을 읽고 해당 메시지를 웹 브라우저에 전송한다.

- **bridge 데몬:** bridge 데몬은 파이프를 사용하여 IoTivity 클라이언트 데몬과 fileRW 데몬 간의 프로세스 통신을 할 수 있도록 한다. Bridge 데몬이 실행되면 자식 프로세스를 생성하고 자식 프로세스에서는 fileRW 데몬을 부모 프로세스에서는 IoTivity 클라이언트 데몬을 실행한다. 두 프로그램 간의 프로세스 통신을 위하여 두 개의 파이프를 이용하고 표준 입출력을 통해 통신할 수 있게 한다.
- **fileRW 데몬:** 웹 서버는 사용자 PC로부터 요청된 메시지를 파일에 저장한다. fileRW 데몬은 이 파일의 존재를 확인하고, 파일이 존재하면 파일에 저장된 메시지를 읽은 후 IoTivity 클라이언트 데몬으로 표준 입출력 함수를 통해 전송한다. 전송은 파이프를 이용한 프로세스 간 표준 입출력을 이용하는데 bridge 프로그램이 이를 가능하도록 파이프 역할을 한다.
- **IoTivity 클라이언트 데몬:** IoTivity 클라이언트 데몬에서는 IoT 엔드 디바이스 리소스를 탐색하고, fileRW 데몬의 리소스 제어 요청을 대기한다. 만약, fileRW 데몬으로부터 어떠한 리소스 제어 요청이 들어올 경우 경우, 요청 메시지를 분석하여 사용자가 원하는 요청이 무엇인지 판단한 후 요청을 해당 엔드 디바이스에 전달하고 이에 대한 응답을 fileRW 데몬에게 반환한다. 요청 후 응답받은 메시지를 분석한 결과에 따라 사용자에게 상황에 맞는 응답 메시지를 fileRW 데몬으로 출력한다. 출력 후 fileRW 출력 후 다시 사용자가 입력한 문자열을 기다린다.

3.3 사용자 PC

- **웹 브라우저:** 엔드 디바이스의 리소스를 제어할 수 있는 인터페이스로 웹 브라우저이다. 웹 브라우저에서 사용자는 웹 페이지에 출력된 리소스 리스트 중 제어할 리소스에 대한 리소스 번호와 기능을 입력한 후, 전송 버튼을 누른다. 이를 통해 웹 브라우저는 웹 서버에 요청을 전달하고, 이를 수신한

웹 서버는 해당 요청에 대한 처리를 진행한 후 사용자 PC에 응답을 전송한다. 응답 메시지는 웹 페이지의 Textarea에 출력하게 된다.

4. 구현

[그림 4]는 [그림 3] 설계를 구현한 시스템의 웹 인터페이스 화면이다. 두 엔드 디바이스의 IoTivity 서버에서 생성한 5가지 리소스와 각 리소스 기능이 웹 페이지 상단에 표시된다. 표시된 리소스 목록 아래는 웹 페이지에서 리소스 제어를 요청한 결과가 표시되는 Textarea, 사용자가 요청을 입력할 수 있는 InputText와 전송 버튼이 있다. 사용 예시로 ARTIK의 GPIO를 이용하여 연결된 LED를 웹 인터페이스를 통해 LED 리소스의 상태를 확인하고 제어하는 명령을 내렸다. LED가 성공적으로 점등 되었음을 확인 하였다.

이를 통해 IoTivity 프레임워크를 사용하여 ARTIK의 기능이 제대로 동작한다는 것과 ARTIK에 IoTivity를 설치하여 리소스를 등록 및 관리할 수 있다는 것이 검증되었다.

5. 결론 및 향후 계획

IoT는 이미 수년 전부터 시장 규모가 커져왔고 시장을 주도하기 위한 여러 기업이나 단체들이 경쟁을 벌이고 있다. 경쟁에 참여하고 있는 OCF의 IoTivity는 IoT 디바이스 간의 상호 운용성을 목표로 여러 플랫폼에서 활용 가능하도록 프로젝트를 진행하고 있다. 삼성에서는 IoT 서비스 개발을 위한 하드웨어 플랫폼을 통해 IoT 시장을 노리고 있으며 IoT 서비스 개발 플랫폼인 ARTIK을 공개하였다.

본 논문에서는 ARTIK 플랫폼과 IoTivity 프레임워크를 사용한 IoT 서비스 개발의 적합성을 검증하기 위해 검증 시나리오를 작성한 후 구현하였다. 검증은 범용 IoT 기능 동작을 검증하였고, 이를 통해 ARTIK 보드가 다양한 IoT 서비스를 제공하는 데 기능 및 성능이 양호할 것으로 판단된다.

참고 문헌

- [1] IoTivity, [online] Available: www.wiki.iotivity.org
- [2] ARTIK developers, [online] Available: www.developer.artik.io
- [3] 이원석, 차흥기, 전종홍. "사물인터넷 오픈소스 기술-IoTivity", 한국통신학회, 정보와 통신 열린강좌, vol. 32, no. 12, pp. 27-35, 2015.
- [4] 박수홍, "OCF 사물인터넷 오픈소스 플랫폼 표준 동향", TTA 저널, vol. 166, pp. 41-45, 2016.